



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Escuela Técnica Superior de Ingeniería del Diseño

TRABAJO FIN DE GRADO

DOCUMENTO Nº 1: MEMORIA

VALENCIA, JULIO 2019

TÍTULO:

**DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA BCI
BASADO EN EL PARADIGMA SSVEP**

APELLIDOS/NOMBRE ESTUDIANTE: Aguado Guerrero, Alonso

DNI: 70592233-N

GRADO CURSADO: Ingeniería Electrónica Industrial y Automática

APELLIDOS/NOMBRE TUTOR: Quiles Cucarella, Eduardo

AGRADECIMIENTOS

Me gustaría aprovechar esta ocasión tan importante para agradecer a D. Eduardo Quiles Cucarella, como Director de este proyecto, la labor realizada, siempre con una actitud cercana y dispuesto a ofrecerme la ayuda necesaria para que pudiera superar los problemas que se iban planteando en el desarrollo de la aplicación. También quiero agradecer a D^a. Nayibe Chio Cho su aportación e implicación en este proyecto, mostrándome en muchas ocasiones el camino correcto para seguir adelante.

Por otro lado, me gustaría agradecer tanto a mis compañeros de clase, como de piso, sin olvidar a mis amigos de estudio, ya que sin ellos no hubiera sido posible llegar hasta aquí.

Finalmente, y no por ello menos importante, agradecer a mi familia todo el apoyo recibido, que, en los momentos más complicados ha confiado en mí y me ha inculcado los valores que hoy en día me definen.

RESUMEN

El presente proyecto está enfocado en el empleo de una interfaz cerebro-computador (BCI- Brain Computer Interface), para el control de una aplicación, utilizando el amplificador EEG Enobio de 8 canales. Este sistema utiliza las señales del electroencefalograma (EEG) generadas por el cerebro humano y permite que una persona pueda interactuar con su entorno sin necesidad de mover ningún músculo. Los sistemas BCI pueden basarse en numerosos paradigmas: P300, Motor Imagery... Pero para esta aplicación se ha elegido el paradigma SSVEP (Steady State Visual-Evoked Potentials), que se considera como una respuesta cortical continua provocada por estímulos visuales repetitivos (RVS). Es decir, los SSVEP contienen la misma frecuencia fundamental que el RVS al que esté mirando el usuario. Por tanto, este tipo de aplicaciones están destinadas principalmente a personas con problemas de movilidad reducida.

Para llevarlo a cabo, el proyecto se divide en dos partes; Por un lado, usando el entorno LabVIEW y la DAQ USB-6008, se ha realizado una fase de entrenamiento, necesaria para el funcionamiento del sistema de control posterior. Tres matrices de diodos LEDs han actuado como RVS. Se han hecho varias pruebas a diferentes sujetos de varias edades que se analizarán y veremos cómo esta fase repercute en la segunda parte del proyecto.

Por otro lado, se ha diseñado e implementado una aplicación para el control, utilizando como RVS los LEDs. De esta manera, se ha desarrollado una aplicación en LabVIEW, que, mediante la comunicación con OpenViBE, extrae la frecuencia del SSVEP, y en función de esta, actuará encendiendo otro determinado LED, previamente definido.

Palabras clave: BCI, Interfaz cerebro-computador, SSVEP, Estímulo Visual Repetitivo (RVS), Electroencefalograma (EEG), Frecuencia.

ÍNDICE GENERAL

1. INTRODUCCIÓN	11
1.1. MOTIVACIÓN	12
1.2. ANTECEDENTES	12
1.3. DESCRIPCIÓN DE LA APLICACIÓN	14
1.4. FASES DEL DESARROLLO	15
1.5. ELEMENTOS UTILIZADOS	16
2. CONCEPTOS BÁSICOS	18
2.1. ELECTROENCEFALOGRAFÍA (EEG)	18
2.2. BRAIN COMPUTER-INTERFACE (BCI)	19
2.3. BRAIN COMPUTER-INTERFACE BASED SSVEP	20
2.4. TARJETA DE ADQUISICIÓN DE DATOS (DAQ)	23
2.5. ENOBIO 8	24
3. SOFTWARE UTILIZADO	27
3.1. LABVIEW	27
3.2. OPENVIBE	29
4. PROGRAMAS PREVIOS AL DESARROLLO	33
4.1. CREACIÓN DEL ESTÍMULO VISUAL REPETITIVO	33
4.2. APLICACIÓN EN OPENVIBE	38
4.3. CONEXIÓN ENTRE LABVIEW Y OPENVIBE	51
4.3.1. APLICACIÓN EN LABVIEW PARA LA CONEXIÓN TCP	53
5. DESCRIPCIÓN DETALLADA DEL PROYECTO	57
5.1. OBJETIVO DEL PROYECTO	57
5.2. FUNCIONAMIENTO GENERAL DE LA APLICACIÓN	58
5.3. FASE DE ENTRENAMIENTO	61
5.4. ETAPA DE CONTROL	67
6. EXPERIMENTO	71
7. RESULTADOS	82
8. CONCLUSIONES	97
9. CONCLUSIONES ACADÉMICAS	99
10. REFERENCIAS	101
ANEJOS A LA MEMORIA	105
A-1. DATASHEETS	105

ÍNDICE DE FIGURAS

Figura 1. Número de publicaciones sobre BCI por año (2000 – 2018) [1]	13
Figura 2. EEG dependiendo del rango de frecuencia [6]	19
Figura 3. Localización de la corteza visual primaria [12]	20
Figura 4. Esquema general de funcionamiento de una BCI basada en el paradigma SSVEP	22
Figura 5. Tarjeta de adquisición de datos NI USB-6008 [21]	23
Figura 6. Esquema de pines de la DAQ NI USB-6008 [22]	24
Figura 7. Casco EEG Enobio 8 [23]	24
Figura 8. Posición de los 32 agujeros del casco [24]	25
Figura 9. Simbología del programa LabVIEW [25]	27
Figura 10. A la izq. el panel frontal, a la der. el diagrama de bloques de un VI	29
Figura 11. Logo OpenViBE [26]	29
Figura 12. OpenViBE Acquisition Server	30
Figura 13. OpenViBE Designer	31
Figura 14. Bloque “DAQmx Create Virtual Channel”	34
Figura 15. Estructura Time Loop de LabVIEW	34
Figura 16. Estructura Case de LabVIEW	35
Figura 17. Programación del ‘case selector’	35
Figura 18. Programación de la estructura ‘case’, a la izq. genera un ‘1’, a la der. un ‘0’ lógico	36
Figura 19. Programación de la frecuencia de la onda cuadrada	37
Figura 20. Programación del stop automático	38
Figura 21. Generador de pulsos en LabVIEW completo	38
Figura 22. Detalle del escenario OpenViBE “ssvep-bci-0-acquisition-test.xml”	39
Figura 23. Detalle del escenario OpenViBE “ssvep-bci-1-ssvep-configuration.xml”	40
Figura 24. Detalle del bloque configurable “Peripheral Settings”	40
Figura 25. Detalle del bloque configurable “Experiment Settings”	42
Figura 26. Detalle del escenario OpenViBE “ssvep-bci-2-training-acquisition.xml” ..	43
Figura 27. Detalle del bloque configurable “SSVEP Training Controller Box”	45
Figura 28. Ventana gráfica de la fase de entrenamiento SSVEP	46
Figura 29. Detalle del escenario OpenViBE “ssvep-bci-3-CSP-training.xml”	47
Figura 30. Detalle del escenario OpenViBE “ssvep-bci-4-classifier-training.xml”	48
Figura 31. Detalle del escenario OpenViBE “ssvep-bci-5-online-test-shooter.xml”	49
Figura 32. Detalle del bloque “VRPN Ship Control”	50
Figura 33. Online shooter generado por el escenario OpenViBE “ssvep-bci-5-online-test-shooter.xml”	50
Figura 34. Detalle de la configuración del bloque “Stimuli to TCP” del escenario 2... 52	
Figura 35. Programa para la transmisión vía TCP de datos binarios entre LabVIEW y otro programa externo	53
Figura 36. Interfaz de usuario programa de conexión vía TCP entre LabVIEW y otro programa externo	54
Figura 37. Diagrama de funcionamiento de la fase de entrenamiento	58
Figura 38. Diagrama de funcionamiento de la etapa de control	59

Figura 39. Interfaz gráfica del programa “Fase de entrenamiento.vi”	62
Figura 40. Detalle del escenario OpenViBE “Análisis entrenamiento.xml”	65
Figura 41. Interfaz gráfica del programa “Generador RVS.vi”	67
Figura 42. Interfaz gráfica del programa “Aplicación Control BCI – SSVEP.vi”	68
Figura 43. Configuración de los electrodos en el Enobio 8	71
Figura 44. Verificación del funcionamiento de los electrodos, NIC2	72
Figura 45. Ajustes de la pestaña “Driver Properties”	73
Figura 46. Configuración de los nombres de los canales “Change channel names”	73
Figura 47. Conexión Enobio 8 – OpenViBE	74
Figura 48. Adquisición de señales EEG en el programa de OpenViBE	74
Figura 49. Ajustes para la fase de entrenamiento	75
Figura 50. Ilustraciones de la fase de entrenamiento	76
Figura 51. Ajustes para la generación de los RVS	77
Figura 52. Ajustes para la fase de control	78
Figura 53. Sistema de control, el sujeto se fija en el RVS de 12 Hz	79
Figura 54. Sistema de control, el sujeto se fija en el RVS de 15 Hz	79
Figura 55. Sistema de control, el sujeto se fija en el RVS de 20 Hz	80
Figura 56. Análisis espectral de la fase de entrenamiento del sujeto nº 1	82
Figura 57. Análisis espectral de la fase de entrenamiento del sujeto nº 2	84
Figura 58. Análisis espectral de la fase de entrenamiento del sujeto nº 3	86
Figura 59. Análisis espectral de la fase de entrenamiento del sujeto nº 4	88
Figura 60. Análisis espectral de la fase de entrenamiento del sujeto nº 5	90
Figura 61. Análisis espectral de la fase de entrenamiento del sujeto nº 6	92
Figura 62. Análisis espectral de la fase de entrenamiento del sujeto nº 7	94

ÍNDICE DE TABLAS

Tabla 1. Resultados de la etapa de control del sujeto nº 1.....	83
Tabla 2. Resultados de la etapa de control del sujeto nº 2.....	85
Tabla 3. Resultados de la etapa de control del sujeto nº 3.....	87
Tabla 4. Resultados de la etapa de control del sujeto nº 4.....	89
Tabla 5. Resultados de la etapa de control del sujeto nº 5.....	91
Tabla 6. Resultados de la etapa de control del sujeto nº 6.....	93
Tabla 7. Resultados de la etapa de control del sujeto nº 7.....	95

1. INTRODUCCIÓN

A medida que pasan los años aumenta la necesidad de facilitar la vida del ser humano, para ello fue creada la tecnología, la cual, forma una parte fundamental en la vida cotidiana de las personas. Además, toda empresa busca utilizar la tecnología de forma eficiente para obtener la máxima rentabilidad. De ello surge la intención de automatizar y controlar los procesos. Actualmente, en cualquier vivienda o empresa podemos observar ejemplos de este control de procesos (control de climatización, control de robots en una cadena de montaje...).

El presente proyecto tiene como objetivo desarrollar una aplicación BCI (Brain-Computer Interface) o interfaz cerebro-computador basado en el paradigma SSVEP (Steady-State Visual Evoked Potential), dirigida principalmente a personas con habilidades motoras reducidas.

Aunque generalmente se concibe como una tecnología comercial para el confort y el lujo en edificios de alta gama. Esta aplicación agrega vías alternativas y flexibles al paradigma de interacción típico del usuario con su entorno doméstico, lo que representa una solución accesible y eficiente para brindar a las personas con discapacidad una interacción directa con el ambiente que aumentará de forma significativa su calidad de vida.

Para ello, se ha optado por desarrollar la mencionada aplicación BCI basada en SSVEP, la cual proporciona un mayor rendimiento de información y requiere un menor tiempo de entrenamiento al de otros sistemas BCI que utilizan otras señales cerebrales. Para obtener un SSVEP, se debe presentar un estímulo visual repetitivo (RVS – Repetitive Visual Stimuli), es decir, un punto luminoso que este parpadeando a una frecuencia. Analizando la señal EGG (electroencefalograma) procedente del cerebro podremos obtener numerosas conclusiones, tal y como veremos a lo largo de este proyecto.

1.1. MOTIVACIÓN

La principal motivación para desarrollar este proyecto es la finalización de los estudios del Grado en Ingeniería Electrónica Industrial y Automática, cursado en la Universidad Politécnica de Valencia (UPV), concretamente en la Escuela Técnica Superior de Diseño (ETSID).

Además, el tema que trata este proyecto resulta de gran interés y con el tiempo podría ser una gran apuesta de futuro en numerosas aplicaciones como una interfaz de automatización del hogar (o Domótica), en hospitales...

Este proyecto se desarrolla bajo la supervisión y guía del tutor D. Eduardo Quiles Cucarella.

1.2. ANTECEDENTES

Aunque el auge de esta tecnología se da en la última década, el origen de las tecnologías BCI surge con el descubrimiento de la naturaleza del cerebro. Por tanto, hay que remontarse al año 1875, cuando el científico Richard Caton usó un galvanómetro para observar impulsos eléctricos en la superficie del cerebro de algunos animales.

Pero no será hasta 1924, cuando Hans Berger consiguió registrar la actividad cerebral mediante electroencefalografía, siendo el primero en registrar y analizar un EEG de un cerebro humano.

Durante la década de los 70 se inician las primeras investigaciones de los novedosos dispositivos BCI en la Universidad de California Los Ángeles (UCLA). Estas investigaciones iban dirigidas principalmente a fines médicos.

Con la entrada del siglo XXI se incrementa el número de artículos de investigación sobre esta tecnología. Además se descubren numerosas aplicaciones que se pueden llevar a cabo mediante una interfaz cerebro-computador. Seguidamente, el estudio se centra también en la industria de videojuegos y se comienzan a ver las primeras BCI en el mercado.

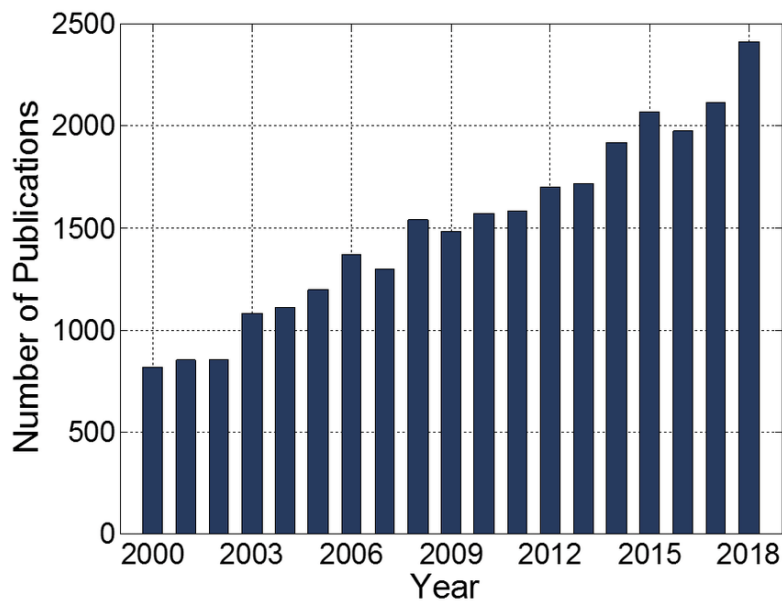


Figura 1. Número de publicaciones sobre BCI por año (2000 – 2018) [1]

En los últimos años ha despegado completamente esta tecnología, y ha aumentado considerablemente el número de investigaciones, por lo que se ha prosperado y avanzado de forma importante en el conocimiento de los sistemas BCI.

1.3. DESCRIPCIÓN DE LA APLICACIÓN

En este apartado se realiza un breve resumen más técnico de la aplicación para, posteriormente, profundizar en cada etapa de diseño.

Como se ha indicado anteriormente, la principal intención de esta aplicación es la de permitir a un usuario interactuar con su entorno, mediante una interfaz cerebro-computador (BCI), sin mayor esfuerzo que fijar la mirada en un estímulo visual repetitivo (RVS). Para ello, hemos utilizado el prototipo llamado “*Enobio 8*”, un amplificador EEG de electrodos con 8 canales que se encarga de la grabación de electroencefalogramas (EEG) procedentes del cerebro.

Como estímulo visual repetitivo (RVS) se ha optado por unas matrices de LEDs que serán las encargadas de generar trenes de pulsos a una frecuencia determinada. Esto es posible gracias al entorno *LabVIEW*, ya que se ha programado una aplicación que permite generar trenes de pulsos a una frecuencia controlada por el usuario. Además del uso de la tarjeta de adquisición de datos “*NI USB-6008*”, que permite la conexión entre el programa y la parte física.

Además, el desarrollo de una aplicación en *OpenViBE*, cuyos RVS se configuran en la pantalla del computador, permitirá realizar una fase de entrenamiento del sistema que posibilite la fase posterior de control. Durante esta fase de entrenamiento, se verá el diferente rendimiento del sistema en función del sujeto que la realice. El proyecto se centrará en los resultados con los LEDs como RVS.

Gracias a esta fase de entrenamiento, se configuraran los filtros y el clasificador que posibilitan la realización de una etapa de control, donde el usuario, al centrar a mirada en un estímulo (matriz de LEDs) habilitará el encendido de otro LED que ejercerá de actuador.

1.4. FASES DEL DESARROLLO

Para llevar a cabo el proyecto se han seguido las siguientes fases según el orden descrito a continuación:

1. Búsqueda de información sobre BCI y señales EEG a través de la lectura de artículos científicos publicados durante los últimos años.
2. Desarrollo de una aplicación para controlar la frecuencia en *LabVIEW*, a través de la generación de trenes de pulsos con la tarjeta de adquisición de datos “*NI USB-6008*”.
3. Auto-aprendizaje sobre la plataforma de software *OpenViBE* para ser capaz de entender y desarrollar las aplicaciones pertinentes.
4. Búsqueda de información sobre las conexiones entre programas externos haciendo uso del protocolo TCP.
5. Desarrollo de una fase de entrenamiento del sistema.
6. Desarrollo de una aplicación para el control del sistema BCI.
7. Realización de una fase experimental con diferentes usuarios y el posterior análisis de los resultados correspondientes.

1.5. ELEMENTOS UTILIZADOS

La realización de este proyecto ha sido posible gracias al uso de los siguientes medios, tanto hardware como software:

- **Hardware:**
 - Placa protoboard.
 - Diodos LEDs (Light-Emitting Diode) de diferentes colores.
 - Tarjeta de adquisición de datos de National Instruments *USB-6008*.
 - Casco EEG de 8 canales, “*Enobio 8*”.
 - Computador portátil Asus X541U, Intel® Core i5-7200, 15.6", de 4GB RAM, Disco Duro 500GB, Gráfica GeForce 920M 2GB y Windows 10.

- **Software:**
 - Entorno de desarrollo *LabVIEW 2018*.
 - Software *OpenViBE 2.2.0*.
 - Software de Neuroelectrics, *NIC2*.

2. CONCEPTOS BÁSICOS

Durante este apartado se explican de manera científica y profunda los conceptos fundamentales para la correcta y total comprensión del proyecto.

2.1. ELECTROENCEFALOGRAFÍA (EEG)

La electroencefalografía (EEG) es una exploración neurofisiológica que se basa en el registro de la actividad cerebral.

Existen numerosas técnicas para medir la actividad cerebral de un ser humano, como el magnetoencefalograma (MEG), la espectroscopia del infrarrojo cercano (NIRS), el electrocorticograma (ECoG), la resonancia magnética (IRMf) y la electroencefalografía (EEG). Cada técnica posee algunas ventajas y desventajas en comparación con otras. Por ejemplo, en la EEG, la resolución temporal es alta, pero la resolución espacial es baja en comparación con IRMf. Debido al bajo coste y la gran portabilidad, el EEG se ha utilizado en gran medida en aplicaciones tanto clínicas como de investigación [2][3][4][5].

Hay muchas características que se pueden extraer del electroencefalograma, por ejemplo, se pueden distinguir seis ritmos cerebrales en función de las diferencias en los rangos de frecuencia; delta (1-4 Hz), theta (4-7 Hz), alfa (8-12 Hz), mu (8-13 Hz), beta (14-30 Hz) y gamma (25-100 Hz). Los ritmos delta y theta ocurren en condiciones emocionales altas o en una etapa de sueño. El ritmo alfa se da despierto y relajado, y su oscilación tiene un patrón suave. El patrón del ritmo beta está desincronizado y se da despierto en condiciones normales. Por último, el ritmo mu puede adquirirse a partir de la corteza sensoriomotora, y el ritmo gamma de la corteza somato sensorial.

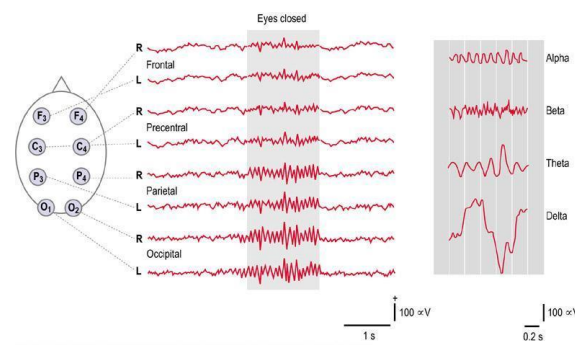


Figura 2. EEG dependiendo del rango de frecuencia [6]

2.2. BRAIN COMPUTER-INTERFACE (BCI)

Una de las aplicaciones de investigación de la EEG es en un sistema de interfaz cerebro-computador (BCI).

Una interfaz cerebro-computador (BCI) es un sistema de comunicación en el que la intención de un sujeto se transmite al mundo externo sin necesidad de involucrar las vías de salida normales de los nervios y músculos periféricos. Las BCI son especialmente relevantes para personas con habilidades motoras reducidas. Un sistema BCI puede enviar comandos distinguidos por el procesamiento de señales EEG, controlados por la actividad cerebral.

Se distinguen dos tipos de BCI según el método de obtención de la señal; **dispositivos invasivos**, en los que la medición se realiza directamente desde el cerebro del usuario, por lo que es necesario realizar una intervención quirúrgica; **dispositivos no invasivos**, que es el tratado en este proyecto, y se caracterizan por el uso de un sensor que se coloca de manera que no se realiza una intrusión sobre el cuerpo humano.

En las BCI no invasivas, la EEG se emplea comúnmente debido a su alta resolución temporal, facilidad de adquisición y rentabilidad en comparación con las otras técnicas. Los dispositivos no invasivos para el control de BCI incluyen la sincronización/desincronización (ERS/ERD), los “visual evoked potentials” (VEP), “steady-state visual evoked potentials” (SSVEP), los “slow cortical potentials” (SCP), los “P300 evoked potentials” y los ritmos mu y beta [7][8][9][10].

En este proyecto se ha utilizado un control BCI basado en el paradigma “Steady-State Visual Evoked Potentials” (SSVEP), los cuales han recibido una mayor atención en el ámbito científico ya que pueden proporcionar “*bit rates*”¹ más altos, de hasta 70 bits/min, mientras que requieren un menor tiempo de entrenamiento.

Por último, cabe mencionar que durante los últimos años se ha convertido en una tendencia para la investigación la combinación de diferentes tipos de interfaces cerebro-computador, denominadas “*BCI híbridas*”. Destaca, por ejemplo, la BCI híbrida P300-SSVEP.

2.3. BRAIN COMPUTER-INTERFACE BASED SSVEP

Diferentes estudios electrofisiológicos y neurofisiológicos han demostrado el aumento de la actividad neuronal al fijar la mirada en un estímulo. Los potenciales evocados visuales (“Visual Evoked Potentials”) son provocados por estímulos visuales repetitivos (RVS – Repetitive Visual Stimulus). Serán estos RVS los que conducen a un patrón estable (“Steady State”) de oscilaciones de voltaje en el EEG, que denominaremos SSVEP.

A lo largo del tiempo, SSVEP se ha definido de dos formas diferentes. Por un lado Ragan [11] planteó que el SSVEP es una respuesta directa en la corteza visual primaria también conocida como zona occipital.

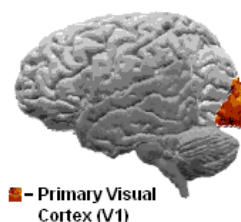


Figura 3. Localización de la corteza visual primaria [12]

¹ Bit rate (Tasa de bits): Se trata del número de bits que se transmiten por unidad de tiempo a través de un sistema de transmisión digital o entre dos dispositivos digitales. Por tanto, se conoce como la velocidad de transferencia de datos.

Sin embargo, Silberstein en [13] asumió que el SSVEP incluye respuestas corticales indirectas a través de *cortical-loops* (bucles corticales) desde la retina periférica, mientras se realiza una tarea cognitiva.

En la actualidad, aunque aún se desconoce de forma exacta el mecanismo principal del SSVEP, en general, se considera como una respuesta cortical continua provocada por estímulos visuales repetitivos (RVS) con una frecuencia constante en la retina central.

Como una forma de onda oscilatoria casi sinusoidal, el SSVEP contiene la misma frecuencia fundamental que el RVS, además de algunos armónicos de dicha frecuencia. Es decir, cuando el usuario centra su mirada en un RVS, se obtiene un SSVEP que se manifiesta como componentes oscilatorios en el EEG procedente del BCI, especialmente en las señales de la corteza visual primaria.

Los SSVEP pueden ser provocados por RVS en el rango de frecuencias de 1 a 100 Hz [14]. En la mayoría de diseños de sistemas BCI las frecuencias de estimulación están en el rango de 4 a 50 Hz. En [15] estas frecuencias se han clasificado en tres bandas: baja (1-12 Hz), media (12-30 Hz) y alta (30-60 Hz). Cada banda de frecuencia tendrá unas ventajas y unas desventajas en su uso.

En el artículo [16] se realizan varias pruebas con diferentes sujetos para determinar cuáles son las características más eficientes de los RVS para el diseño de BCI basadas en SSVEP. En cuanto a la frecuencia de los RVS, se obtuvieron las siguientes conclusiones; las mayores amplitudes de SSVEP se observaron cerca de los 10 Hz seguidas de 16-18 Hz, y la banda de alta frecuencia mostró la respuesta de menor amplitud. Aunque las bandas de baja y media frecuencia fueron las que mejor respuesta ofrecieron, tienen algunas desventajas significativas. Primero, las evaluaciones subjetivas mostraron que las frecuencias entre 5 y 25 Hz son más molestas que las altas, es decir, la fatiga visual aparece con mayor facilidad. En segundo lugar, algunos tipos de RVS, como los de inversión de flash o patrón pueden provocar ataques epilépticos, especialmente en el rango de 15 a 25 Hz [17]. Y en tercer lugar, la banda de baja frecuencia cubre el ritmo alfa (8-13 Hz) antes mencionado, que puede causar una cantidad considerable de falsos-

positivos. Todos estos inconvenientes podrían evitarse utilizando la banda de frecuencia alta, pese a obtener una peor respuesta.

En cuanto al resto de características, se concluyó que la respuesta SSVEP provocada por un LED fue mayor que la de un estímulo en la pantalla de un ordenador o la de reversión de un patrón [15]. Esto explica que los “*bit rates*” de BCI que utilizan RVS de LEDs sean más altos que los que usan otra fuente.

Además, está demostrado en [18] que diferentes colores en RVS provoca diferentes efectos sobre el SSVEP, en combinación con la frecuencia utilizada. Por ejemplo, la luz roja provocó la respuesta más fuerte cuando se moduló a 11 Hz, sin embargo, los estímulos de luz amarilla mostraron una peor respuesta.

Otro aspecto muy importante a tener en cuenta para el diseño de una BCI basada en SSVEP es la relación Señal/Ruido (SNR - Signal/Noise Ratio), calculada como la relación entre la potencia del EEG en la frecuencia de estimulación y la potencia media de las bandas de frecuencia adyacentes. Un análisis en [19] demostró que la utilización de la banda de alta frecuencia puede ser muy prometedora en este aspecto.

Los SSVEP se pueden detectar automáticamente a través de una serie de pasos de procesamiento de señales; preprocesamiento (por ejemplo, un filtro pasa banda), la detección/corrección de artefactos, la extracción de características (por ejemplo, el contenido espectral en las frecuencias de estimulación) y la clasificación de características. El rendimiento de BCI se evalúa generalmente en términos de precisión de clasificación, velocidad de clasificación y número de opciones disponibles.

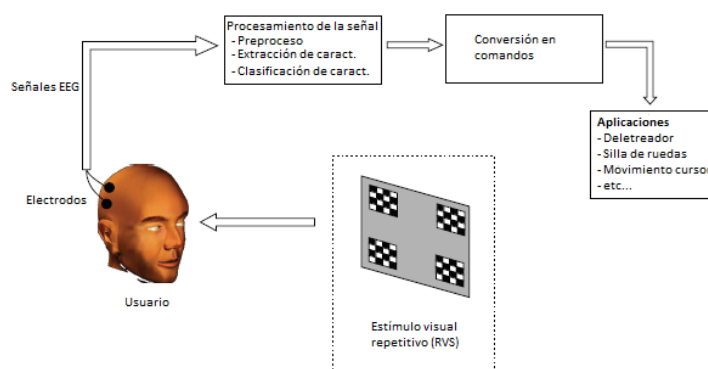


Figura 4. Esquema general de funcionamiento de una BCI basada en el paradigma SSVEP

2.4. TARJETA DE ADQUISICIÓN DE DATOS (DAQ)

Las tarjetas de adquisición de datos (hardware) actúan como la interfaz entre una computadora y las señales físicas, es decir, la información recaudada por el sensor, localizado en el interior, se pasa a la DAQ, la cual se encarga de transformar los códigos del mundo real a códigos digitales, y viceversa. Como si se tratara de un intérprete que traduce de un lenguaje a otro, con el fin de que el sistema digital (computadora, dispositivo electrónico...) sea capaz de comprender los signos del analógico.

Las tarjetas de adquisición de datos están formadas por tres partes principales: por un lado está el circuito de acondicionamiento de señales, después está el convertidor analógico-digital (ADC) y, finalmente, el bus del ordenador [20].

En este proyecto, la DAQ utilizada es la NI USB-6008, del fabricante *National Instruments*, y tiene un papel vital para el correcto funcionamiento de la aplicación.



Figura 5. Tarjeta de adquisición de datos NI USB-6008 [21]

La DAQ NI USB-6008 dispone de 32 pines divididos en 8 entradas analógicas, 2 salidas analógicas y 12 entradas/salidas digitales configurables, más los pines de GND y +V. Los canales digitales se dividen en dos puertos (P0 y P1). Si configuramos uno o más canales de un puerto como entrada o salida, todo el puerto se bloquea en ese modo.

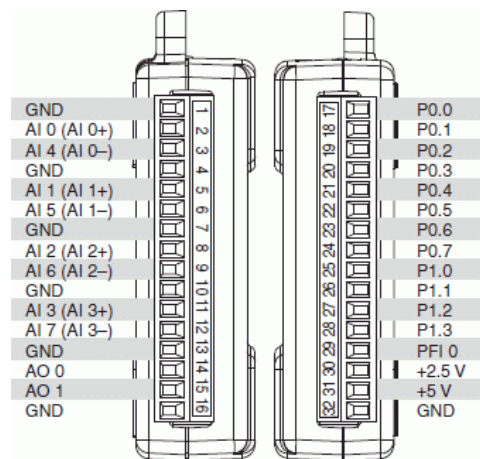


Figura 6. Esquema de pines de la DAQ NI USB-6008 [22]

Su elección para este proyecto se debe por una parte a su bajo coste, pero principalmente porque la USB-6008, al ser fabricada por National Instruments, usa el software “NI-DAQmx” que es totalmente compatible con LabVIEW.

2.5. ENOBIO 8

Para la adquisición de señales EEG se ha utilizado el casco *Enobio 8*, sistema de sensor de electrofisiología inalámbrico y portátil, fabricado por *Neuroelectrics*, para la grabación de EEG, que cuenta con 8 canales. Viene integrado con una potente e intuitiva interfaz de usuario para una fácil configuración, grabación y visualización de datos EEG de 24 bits a 500 S/s, incluyendo espectrograma y visualización 3D en tiempo real de características espectrales.



Figura 7. Casco EEG Enobio 8 [23]

El casco dispone de 32 agujeros para posicionar los electrodos (fig. 8), de los que únicamente utilizaremos 8. Este equipo permite el uso de tanto electrodos secos como húmedos. Para realizar las pruebas correspondientes a este proyecto se han empleado los electrodos secos, pero, en diferentes estudios, se ha observado que se obtienen mejores resultados de adquisición usando los electrodos con el gel conductor.

Al colocar el prototipo se coloca una pinza conductora en el lóbulo de una de las orejas del sujeto, esto se hace para tener una masa y una referencia de tensión como referencia de potencial, ya que las tensiones son medidas en modo diferencial respecto este punto. Estas señales son amplificadas y filtradas por el propio Enobio antes de ser enviadas al computador vía *Bluetooth*.

Su conexión con un computador es realmente sencilla y casi intuitiva, ya que lo hace vía *bluetooth*. Además, la empresa *Neuroelectrics* permite la instalación de un programa que sirve de gran ayuda a la hora de utilizar el prototipo, ya que permite verificar que todos los electrodos conectados están realizando contacto correctamente y se encuentran en pleno funcionamiento. El programa se denomina *NIC2*.

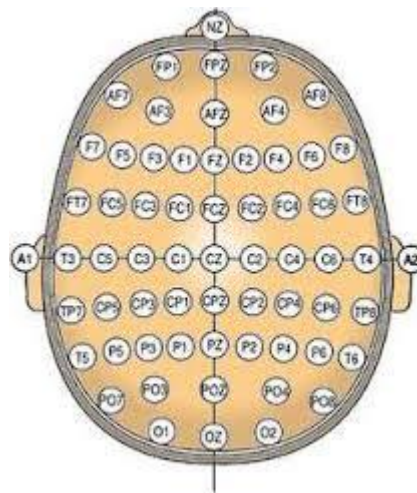


Figura 8. Posición de los 32 agujeros del casco [24]

3. SOFTWARE UTILIZADO

Es imprescindible entender de forma básica el software con el que vamos a desarrollar la aplicación para la correcta comprensión de esta. Por tanto, durante este apartado se va a realizar una breve explicación de los conceptos más importantes del software utilizado.

3.1. LABVIEW

- **¿Qué es LabVIEW?**

LabVIEW es un software de ingeniería de sistemas, desarrollado por National Instruments, que requiere pruebas, medidas y control con acceso rápido a hardware e información de datos.



Figura 9. Simbología del programa LabVIEW [25]

- **¿Por qué elegimos LabVIEW?**

Este entorno ofrece un enfoque de programación gráfica que ayuda a visualizar cada aspecto de nuestra aplicación, incluyendo configuración de hardware, datos de medidas y depuración. Esta visualización hace que sea más sencillo integrar hardware de medidas (por ejemplo una DAQ), representar una lógica compleja en el diagrama, desarrollar algoritmos de análisis de datos y diseñar interfaces de usuario personalizadas [25].

- **¿Cómo funciona LabVIEW?**

Como se ha indicado LabVIEW es una herramienta gráfica de programación, esto significa que los programas no se escriben, sino que se dibujan, facilitando su comprensión. Al tener ya pre-diseñados una gran cantidad de bloques, se le facilita al usuario la creación del proyecto (está formado por varios VI's), con lo cual en vez de estar una gran cantidad de tiempo en programar un dispositivo/bloque, se le permite invertir menos tiempo y dedicarse un poco más en la interfaz gráfica y la interacción con el usuario final. Cada VI consta de dos partes diferenciadas.

- Panel frontal: Se trata de la interfaz con el usuario, se utiliza para interactuar con este cuando el programa se está ejecutando. Los usuarios podrán observar los datos del programa actualizados en tiempo real (estado de las entradas y salidas). En esta interfaz se definen los *controles* (entradas como botones, marcadores...) e *indicadores* (salidas que pueden ser gráficas, termómetros...).

- Diagrama de bloques: Es el programa propiamente dicho, donde se define su funcionalidad, aquí se colocan iconos que realizan una determinada función y se interconectan (el código que controla el programa).

Cada bloque o elemento que haga de control/indicador en el diagrama de bloques se corresponde con el elemento del panel frontal. Pero habrá numerosos bloques que no se visualizarán en el panel frontal, ya que no están para interactuar con el usuario, sino para el correcto funcionamiento del programa.

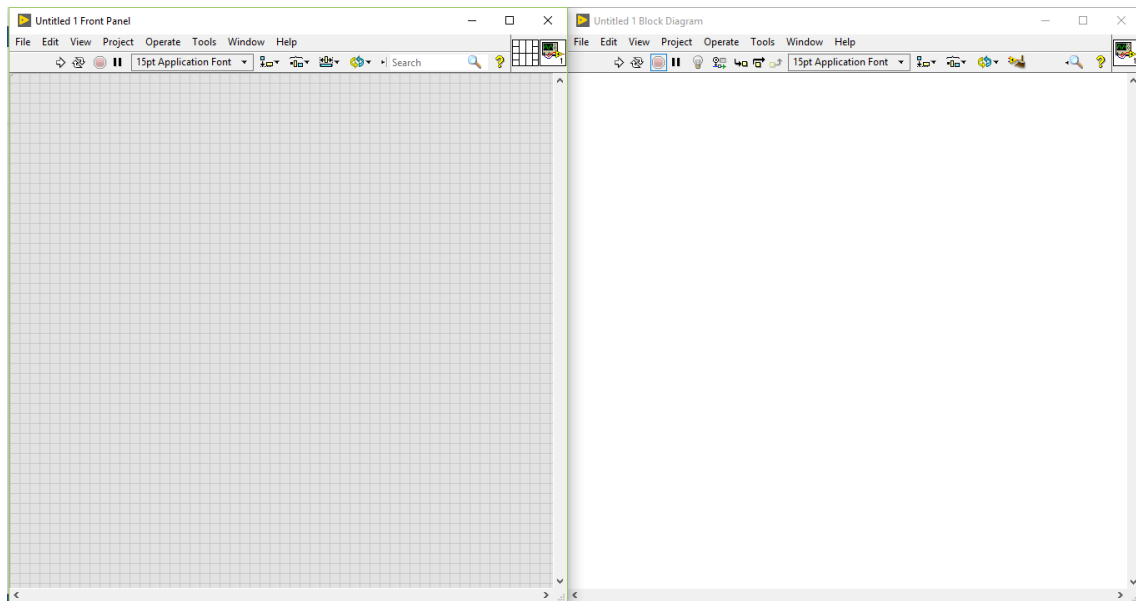


Figura 10. A la izq. el panel frontal, a la der. el diagrama de bloques de un VI

Estos conceptos permitirán entender sin ningún problema las aplicaciones desarrolladas en LabVIEW para llevar a cabo el diseño del proyecto.

3.2. OPENVIBE

OpenViBE es una plataforma de software dedicada al diseño, testeo y uso de interfaces cerebro-computador (BCI). La descarga de este programa comprende una herramienta de diseño para crear y ejecutar aplicaciones personalizadas, además de varios programas ejemplo que están listos para su uso [26].



Figura 11. Logo OpenViBE [26]

Este software trabaja de manera online por lo que permite el procesamiento en tiempo real de señales cerebrales a una elevada frecuencia. Puede emplearse para adquirir, filtrar, procesar, clasificar y visualizar las señales cerebrales en tiempo real.

Su programación se realiza en escenarios, y es muy gráfica, ya que se realiza mediante bloques como en LabVIEW, lo que hace posible una rápida programación con una curva de aprendizaje muy baja.

Al descargar e instalar OpenViBE veremos que disponemos de dos ejecutables principales: *OpenViBE Acquisition Server* y *OpenViBE Designer*.

- *OpenViBE Acquisition Server* se encarga de vincular el amplificador de señales EEG (en nuestro caso Enobio 8) y el programa principal. Permite seleccionar el puerto de conexión, el número de muestras enviadas por bloque, la frecuencia de muestreo, el número de canales (8 para nuestro sensor) y el nombre de estos para posteriormente poder reconocer cada electrodo/canal por dicho nombre, en lugar de por su índice, lo que facilita considerablemente la comprensión para el programador. Además, *OpenViBE Acquisition Server* permite la generación de señales oscilatorias, senoidales y oscilatorias, para aquellos que no dispongan de un sensor al que conectarse. Esto permitirá a cualquiera programar y ejecutar una aplicación.

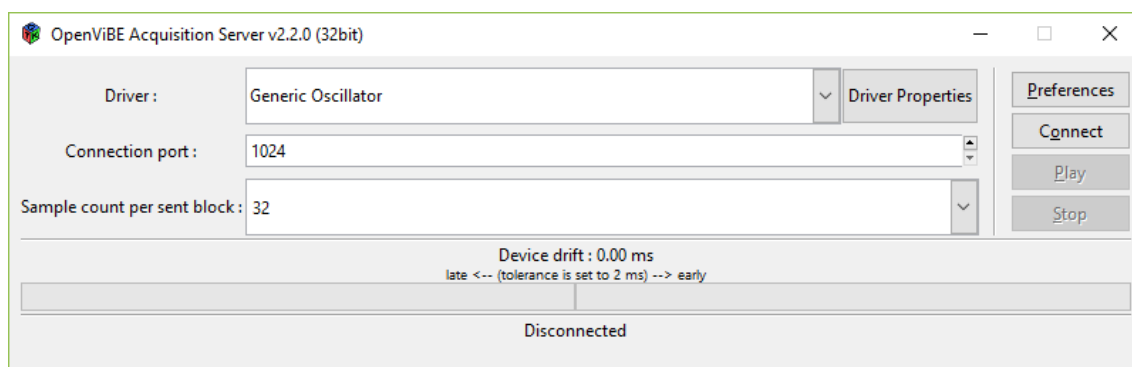


Figura 12. OpenViBE Acquisition Server

- Por otro lado, *OpenViBE Designer* es el entorno de programación, compuesto por escenarios, que a su vez están compuesto por bloques para el tratamiento de señales. Sin embargo, el propio software ya cuenta con una serie de escenarios básicos como son los *box-tutorial* en los que existen ejemplos de cuáles son las funciones de cada una de las cajas de programación. Esto permite aprender rápidamente a utilizar este programa. Además, en cuanto a interfaces cerebro-computador, el propio software cuenta con una serie de *bci-examples* en los cuales ya se encuentran programado algunos de los paradigmas más empleados en este tipo de controles, como pueden ser: *SSVEP*, *P300* o *Motor Imagery*. Estos programan nos sirven para partir como base, ya que con unas ligeras modificaciones dependiendo de los canales o el tipo de filtrado, podremos obtener el programa buscado.

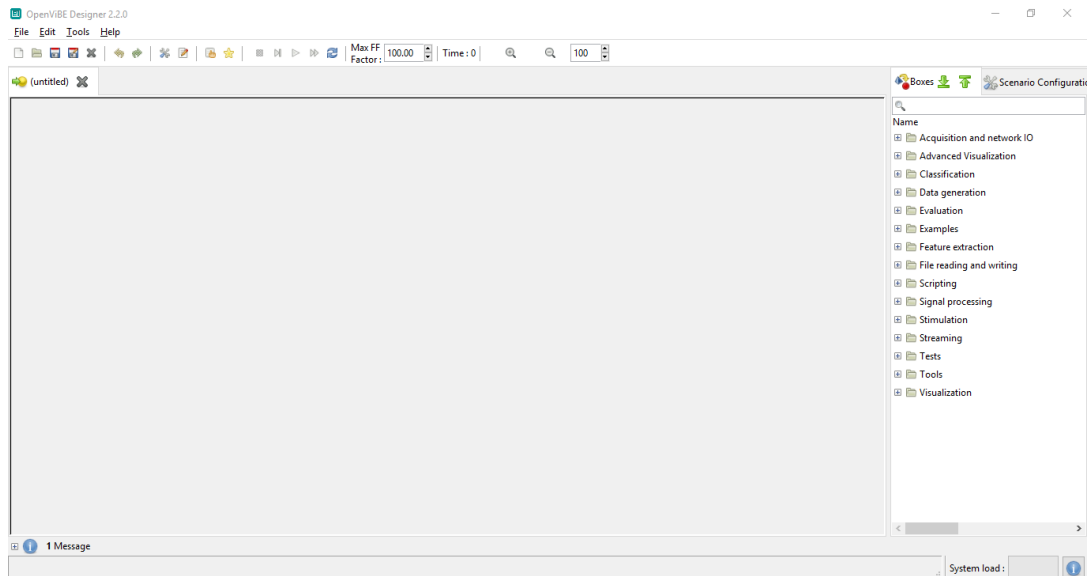


Figura 13. *OpenViBE Designer*

4. PROGRAMAS PREVIOS AL DESARROLLO

Durante este apartado se describe detalladamente los programas que han sido necesario desarrollar para lograr la aplicación final. En secciones posteriores se verá la combinación de todos ellos para llegar al sistema final.

4.1. CREACIÓN DEL ESTÍMULO VISUAL REPETITIVO

El primer paso para desarrollar el proyecto es la confección de los RVS, siguiendo las pautas previamente mencionadas. Como ya se ha indicado, los RVS serán tres matrices de tres LEDs de color rojo.

Para gestionar el control de los estímulos se ha empleado LabVIEW como programa fundamental. Es decir, se ha creado un programa capaz de generar trenes de pulsos con una frecuencia controlada por el usuario.

En esta parte destaca la función de la DAQ USB-6008, que será la encargada de transmitir dichos pulsos generados en LabVIEW a las matrices de LEDs correspondientes.

El primer paso para poder lograr este programa es la instalación del driver “*NI-DAQmx*” previamente mencionado. Este controlador nos permitirá la conexión entre la NI USB-6008 y LabVIEW, ya que es un software gestionado por el mismo fabricante: *National Instruments*.

Una vez instalado el drive nos disponemos a realizar la programación de la aplicación. Para generar los pulsos vamos a utilizar los pines digitales de entrada/salida configurables, en este caso tres y como salida, ya que se trata de generar los trenes de pulsos desde el computador para alimentar a los LEDs.

Entonces, se intuye que el primer paso será la programación de un bloque que configure el pin elegido de la DAQ como salida digital. Este bloque, del que disponemos gracias al drive previamente instalado, se denomina “*DAQmx Create Virtual Channel*” y se encargará de crear un canal virtual, configurarlo como salida digital y agregarlo a una tarea posterior.

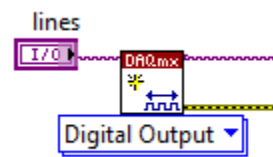


Figura 14. Bloque "DAQmx Create Virtual Channel"

Para lograr la generación de los pulsos la estructura de LabVIEW escogida es la llamada "Time Loop" o "Bucle de tiempo", cuya función es la de ejecutar uno o más subdiagramas secuencialmente cada iteración del bucle, en el periodo que se especifique.

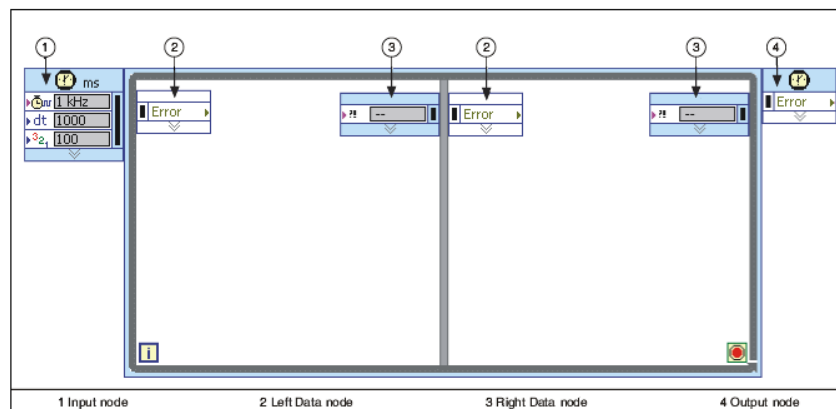


Figura 15. Estructura Time Loop de LabVIEW

Es decir, nos permite desarrollar una programación con capacidades de temporización precisa, retroalimentación sobre la ejecución del bucle, características de temporización que cambien dinámicamente o varios niveles de prioridad de ejecución. El círculo rojo de la figura 15 se corresponde con el botón de parada, y el cuadrado con la letra 'i', nos indica la iteración en la que se encuentra el bucle. Por tanto, esta estructura nos capacitará de establecer la frecuencia deseada. Tal y como veremos a continuación.

Pero antes debemos comprender la programación que irá en el interior de este bucle, la cual será la encargada de generar los pulsos. Esto se llevará a cabo con una estructura “Case”, comúnmente conocida en otros lenguajes de programación como “Switch”, en la que el programador establece un número de casos/estados, y la estructura va ejecutando el estado (compuesto por una programación determinada) que el selector de casos indique.

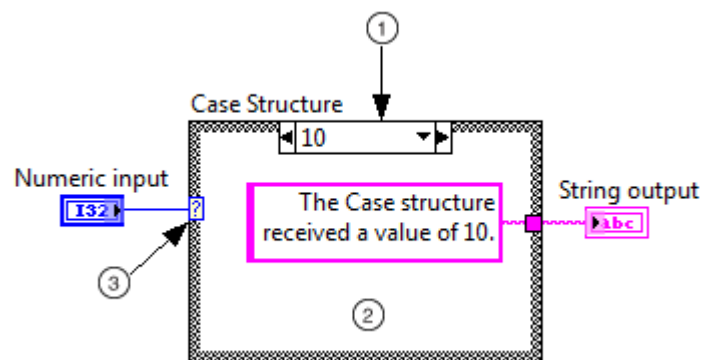


Figura 16. Estructura Case de LabVIEW

Para este programa solo nos harán falta dos casos, uno para el estado alto, y otro para el estado bajo del pulso, por tanto, dejaremos los dos estados que la estructura case nos ofrece por defecto (true, false). En uno de ellos escribiremos un ‘1’ lógico o estado alto, y en el otro un ‘0’ lógico o estado bajo, así se irá conmutando de un estado a otro para generar el correspondiente tren de pulsos.

Para cambiar de un estado a otro haremos uso de la iteración del bucle, tal y como vemos en la siguiente figura:

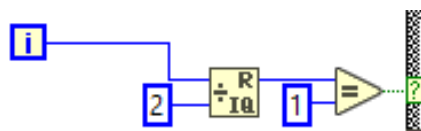


Figura 17. Programación del 'case selector'

La programación es fácil de comprender, pero para ello tenemos que conocer la función del bloque “*Quotient & Remainder Function*”. Esta función utiliza la iteración en la que nos encontramos en el bucle y realiza la siguiente operación: $i-2*\text{floor}(i/2)$; es decir, primero divide la iteración entre 2, el resultado lo redondea al menor entero más cercano (función *floor*), el valor resultante será multiplicado por 2, y se lo restamos a la iteración, para que sea comparada si es igual a 1 o no.

Por ejemplo, si estamos en la $i=1$, el resultado será igual a 1, pero si pasamos a la iteración 2, el resultado será igual a 0. En definitiva, la función de este conjunto de bloques es la de variar entre 0 y 1, lo que dará lugar a un ‘*true o false*’ si lo comparamos con 1, y así pasar la estructura *case* de un estado a otro.

La programación en el interior de la estructura *case* se basa en escribir un ‘0’ o un ‘1’ a través de las DAQ. Para ello haremos uso de la función “*DAQmx Write Digital ID Bool 1Chan 1Samp*” la cual escribe una única muestra de valores booleanos en una tarea que contiene únicamente un canal de salida digital.

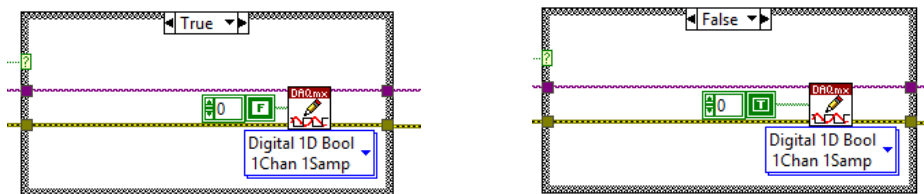


Figura 18. Programación de la estructura ‘case’, a la izq. genera un ‘1’, a la der. un ‘0’ lógico

Para poder establecer la frecuencia deseada, tendremos que pasarla a periodo, ya que el “*Time Loop*”, nos pide el periodo (en milisegundos) al que queremos que se pase de una iteración a otra. Por tanto realizaremos una serie de operaciones matemáticas para lograr pasar la frecuencia (en Hz) que establece el usuario, al periodo (en ms) que exige el bucle.

De esta manera, comenzaremos calculando el periodo a partir de la frecuencia, calculando su inversa. Posteriormente, multiplicamos dicho periodo por 1000, para pasarlo de segundos a milisegundos, y finalmente lo dividimos entre dos. Esta última división tiene una sencilla explicación, como hemos visto antes, en el interior del “*Time Loop*” solo escribimos un ‘0’ o un ‘1’ lógico, y el periodo de una onda cuadrada o tren de pulsos se corresponde a la suma del estado bajo más el estado alto. Por tanto, habrá que pasar el semiperiodo, para que la suma de uno y otro haga el periodo completo.

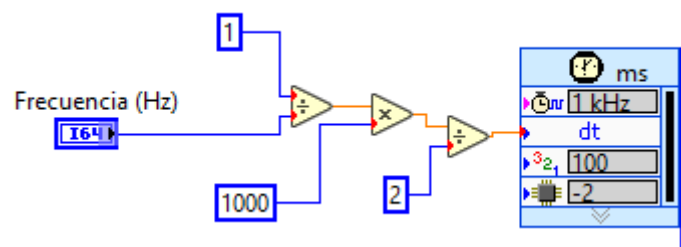


Figura 19. Programación de la frecuencia de la onda cuadrada

Hasta el momento, para detener la aplicación tendríamos que pulsar el botón de stop del bucle, previamente mencionado. Pero en aplicaciones posteriores nos interesará automatizar esta cuestión. Es decir, que el usuario determine un tiempo (en segundos) y la aplicación se detenga una vez ha discurrido el tiempo indicado.

Entonces crearemos un control al que llamaremos “*Tiempo de parpadeo*”, que el propio usuario determinará antes de poner en funcionamiento la aplicación y una vez que haya transcurrido dicho tiempo, esta se cerrará automáticamente.

La idea es pasar los segundos a número de iteraciones, para compararlas con las iteraciones transcurridas en el bucle, y cuando el bucle supere un número determinado de iteraciones, se cierre automáticamente. Lo primero es pasar la frecuencia a periodo, al igual que en el caso anterior, luego dividiremos el “*Tiempo de parpadeo*” introducido, entre el periodo obtenido y el resultado lo multiplicaremos por 2. De esta manera, obtenemos en número de iteraciones deseadas, solo nos quedará programar que cuando la iteración del bucle supere a la calculada, se pulse el botón de stop.

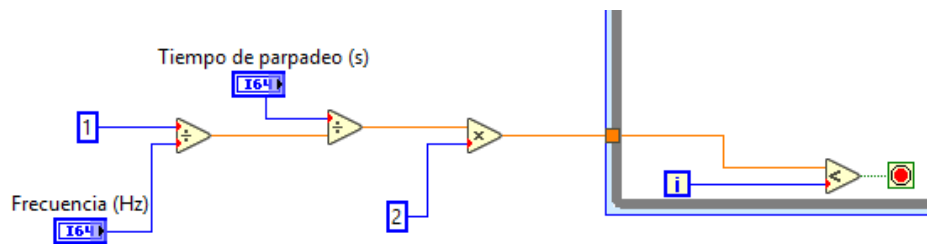


Figura 20. Programación del stop automático

Finalmente, la aplicación completa para generar trenes de pulsos queda de la siguiente forma:

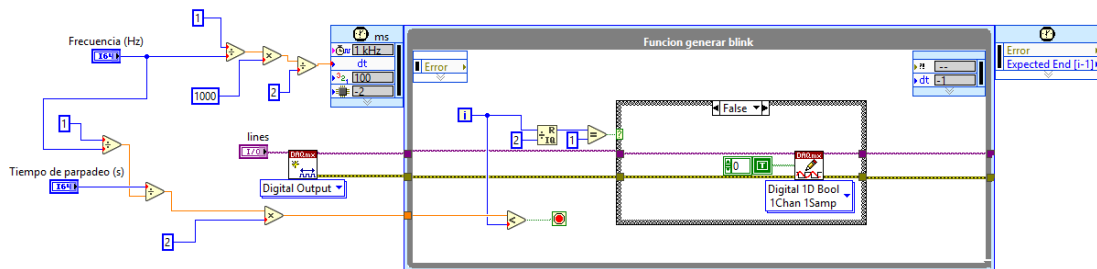


Figura 21. Generador de pulsos en LabVIEW completo

4.2. APLICACIÓN EN OPENVIBE

Como se ha descrito previamente, OpenViBE es la aplicación ideal para el análisis y tratamiento de electroencefalograma (EEG) así como para el diseño y desarrollo de interfaces cerebro-computador. Además, posee numerosos ejemplos que pueden servir de base al empezar a programar.

Dicho esto, para desarrollar esta aplicación del paradigma SSVEP se parte del ejemplo disponible en la carpeta “*bci-examples*” de OpenViBE, denominado “*ssvep*”, cuya función explicaremos a lo largo de este apartado. Este programa nos permite probar un sistema BCI basado en el paradigma SSVEP mediante una serie de pasos: 1. Configuración del experimento; 2. Adquisición de datos mediante entrenamiento; 3. Entrenamiento del clasificador; 4. Prueba online.

El ejemplo consta de seis escenarios, y en cada uno de ellos se comenta y proporciona información adicional, que facilita su completa comprensión. A continuación se comenta la labor de cada escenario:

- ***ssvep-bci-0-acquisition-test.xml*** (fig. 22). Este escenario se encarga de adquirir y mostrar las distintas señales procedentes del sensor, en este caso del Enobio 8 y les realiza un filtro pasa banda. Su principal función es la de verificar que los distintos electrodos del prototipo está realizando una correcta conexión.

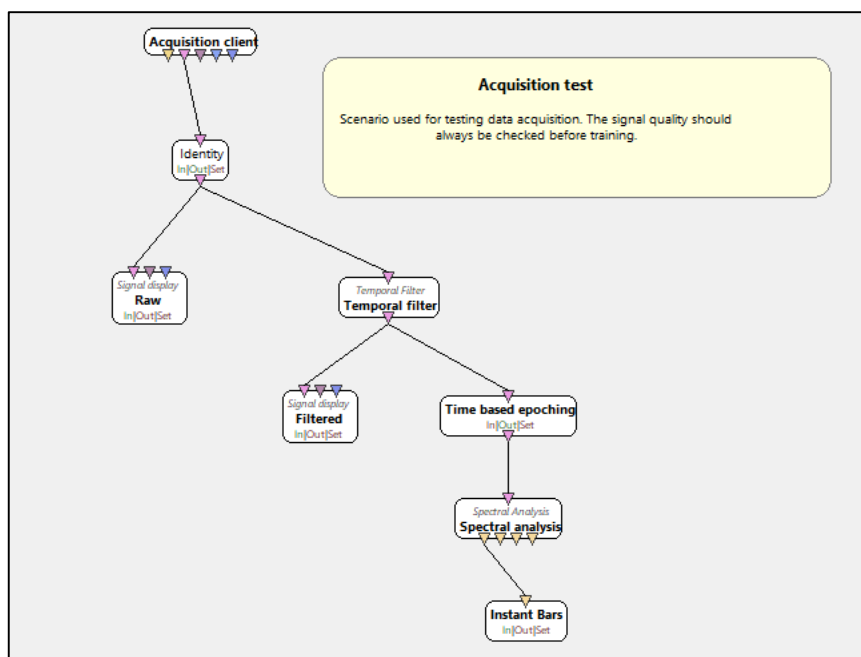


Figura 22. Detalle del escenario OpenViBE “*ssvep-bci-0-acquisition-test.xml*”

- ***ssvep-bci-1-ssvep-configuration.xml*** (fig. 23). Este escenario es el encargado de realizar el primer paso, la configuración del experimento SSVEP, es decir, de la fase de entrenamiento. Y para que la aplicación funcione debe ser ejecutado al menos una vez. Nos centramos en dos bloques, ya que contienen los parámetros de configuración:

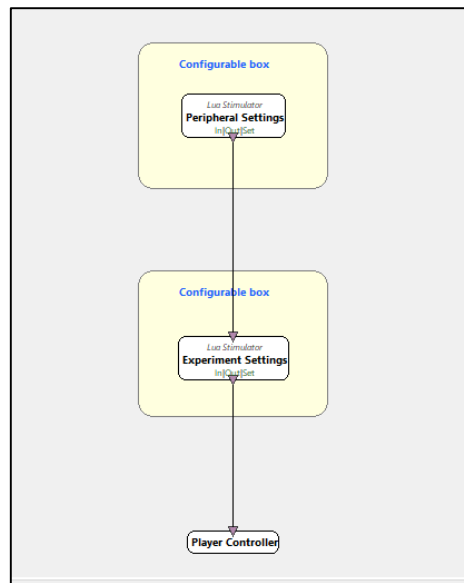


Figura 23. Detalle del escenario OpenViBE “ssvep-bci-1-ssvep-configuration.xml”

- *Peripheral Settings* (Configuración de periféricos): Primero, se debe ingresar información acerca del hardware utilizado. Solo hay una opción configurable: *Display Refresh Rate* (Frecuencia de actualización de la pantalla), el valor indicado debe coincidir con la frecuencia de actualización de la pantalla conectada al computador que ejecuta el estimulador. Este valor determinará qué frecuencias se podrán utilizar en el experimento. Se recomienda la clásica pantalla LCD estándar de 60 Hz.

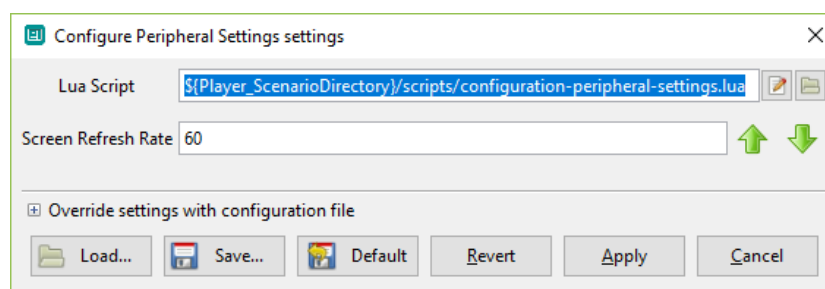


Figura 24. Detalle del bloque configurable “Peripheral Settings”

Esta característica se debe a que OpenViBE, utiliza como RVS la propia pantalla de la computadora, por lo tanto, el estímulo se representa mediante una forma que cambia de color a una frecuencia constante. Para que el SSVEP funcione, este parpadeo o cambio de color debe darse en intervalos muy precisos. Esto significa que las frecuencias que se pueden lograr dependen de la frecuencia de actualización de la pantalla. Por ejemplo, con una “*display refresh rate*” de 60 Hz, que es la que viene por defecto en el programa, se podrán visualizar frecuencias de 30 Hz, 20 Hz, 15 Hz, 12 Hz, 10 Hz y menores; en una pantalla de 50 Hz, se podrá usar 25 Hz, 16.66 Hz, 12.5 Hz, 10 Hz y frecuencias más bajas.

- *Experimental Settings* (Ajuste de experimento): Una vez se hayan configurado los periféricos pasamos a la configuración del experimento SSVEP. En este bloque hay varios ajustes disponibles por configurar:

1. *Target Light Color and Target Dark Color* (Color claro y color oscuro del objetivo). Como se ha descrito previamente, el parpadeo de los objetivos se logra cambiando entre dos colores. El usuario puede elegir cualquier color que desee. Se recomienda elegir dos colores que contrasten, como por ejemplo, **el rojo y el gris oscuro**, que son los colores predeterminados del ejemplo.

2. *Stimulation Frecuencias* (Frecuencias de estimulación): Se trata de las frecuencias de los RVS utilizadas durante el experimento. Las frecuencias se insertan como valores “float” en una lista separadas por punto y coma. El estimulador utilizará estos valores para calcular automáticamente el número de frames necesarios para representar la frecuencia determinada, de acuerdo con la frecuencia de actualización de la pantalla. Los valores que vamos a utilizar para nuestro experimento serán de **20 Hz, 15 Hz y 12 Hz**.

3. *Processing Epoch Duration* (Duración de la etapa de procesamiento): El procesamiento de la señal EEG consiste en dividirla en periodos de tiempo y luego extraer las amplitudes de las ondas cerebrales en cada una de las bandas de frecuencias estimuladas. Este parámetro determina la longitud de dichas divisiones en segundos. Cuanto mayor sea este valor, mejor será la detección, pero mayor será el retardo. En nuestra aplicación utilizaremos el valor por defecto: **0.5 s**.

4. *Processing Epoch Interval* (Intervalo de la etapa de procesamiento): La velocidad (en segundos) a la que se capturarán las divisiones de los periodos de tiempo mencionados. Un intervalo grande proporcionará un control más fluido, mientras que si disminuimos el valor de este parámetro, el control será más preciso. Utilizaremos un valor de **0.1 s**.

5. *Processing Frequency Tolerance* (Tolerancia de la frecuencia de procesamiento): Este parámetro determina el ancho de las bandas buscadas. Es decir, una tolerancia de 0.250 Hz significa que el programa considerará frecuencias de 19.75 a 20.25 Hz para determinar si se estimuló la frecuencia de 20 Hz. Un valor de **0.250 Hz** será suficiente para la aplicación.

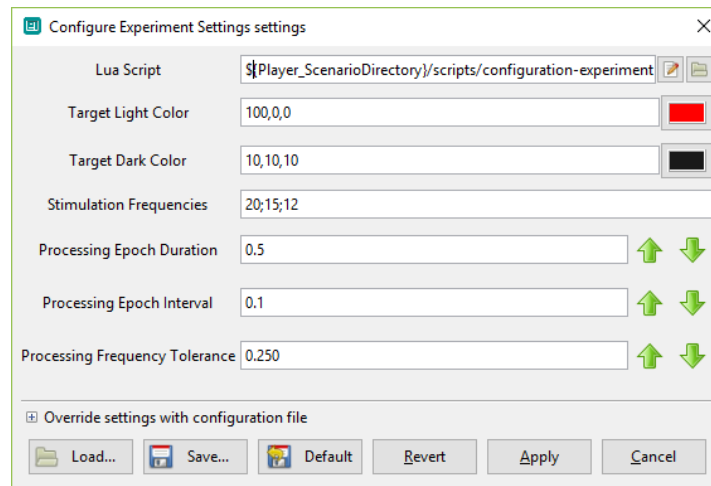


Figura 25. Detalle del bloque configurable “Experiment Settings”

Una vez se hayan configurado todos los parámetros, se ejecuta el escenario. Esto durará unos segundos y se detiene automáticamente, ya que el propio OpenViBE se encarga de crear los archivos de configuración necesarios.

▪ *ssvep-bci-2-training-acquisition.xml* (fig. 26). Después de configurar este escenario, se podrá pasar a la fase de entrenamiento. Los escenarios SSVEP usan los llamados “*Common Spatial Pattern (CSP)*”², que mediante un filtro espacial permite seleccionar las mejores características automáticamente. Esto significa que el usuario dispone de total libertad para colocar los electrodos en cualquier parte del cuero cabelludo del área occipital para obtener buenos resultados.

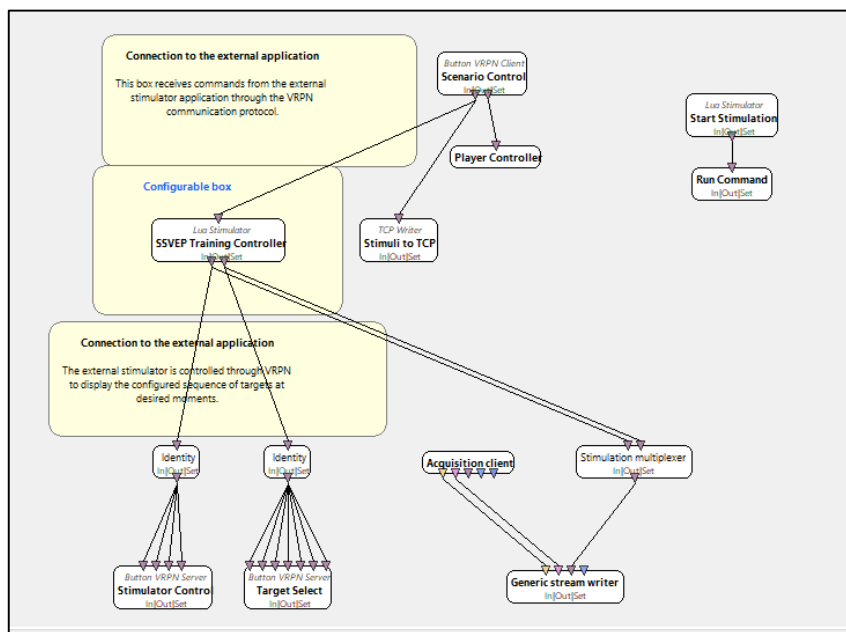


Figura 26. Detalle del escenario OpenViBE “*ssvep-bci-2-training-acquisition.xml*”

² Common Spatial Pattern (CSP): Es una técnica matemática usada en el procesamiento de señales para separar señales multivariadas en subcomponentes con distintas varianzas.

Tras ejecutar este escenario, si inicia la fase de entrenamiento y aparecerá una ventana emergente que muestra tres cuadrados de color rojo en un fondo negro, estos serán los estímulos visuales repetitivos configurados en el escenario anterior. Además, se genera un cuadrado adicional que no parpadeará en ningún momento. Este representa la no estimulación, y no lo veremos ya que es del mismo color que el fondo.

Durante la fase de entrenamiento, los periodos de estimulación (cuando cada cuadrado parpadea a su frecuencia predefinida) se repetirán junto con periodos de descanso (sin parpadeo). Antes de cada periodo de estimulación, uno de los cuadrados se marcará con una flecha amarilla. El sujeto debe enfocar su atención visual en el estímulo marcado durante todo el periodo de estimulación.

Cada casilla se marcará varias veces, según una secuencia, durante todo el entrenamiento. Al final del experimento, se generará un archivo (.ov) que contiene una señal EEG anotada. Por tanto, es importantes establecer un directorio conocido para poder trabajar posteriormente con esa señal.

En el escenario anterior hemos configurado el experimento para tres frecuencias, que se estimularán ocho veces cada una. Para configurar la fase de entrenamiento debemos abrir el bloque denominado “*SSVEP Training Controller Box*”. Los ajustes son los siguientes:

1. “*Goal sequence*”: Hay que tener en cuenta que el cuadrado 0 nunca parpadea. Por lo tanto, si hemos definido tres frecuencias de estimulación, los índices serán 0 (0 Hz), 1 (20 Hz), 2 (15 Hz), y 3 (12 Hz). Entonces se establece una secuencia que indica el estímulo que indicará la flecha para que el usuario se fije. Por ejemplo, la secuencia 0 2 3 1 indica que la flecha primero marcará el estímulo que no parpadea, luego marcará el cuadrado que parpadea a 15 Hz, después el de 12 Hz y finalmente 20 Hz. Como cada frecuencia debe repetirse ocho veces, la secuencia estará compuesta de ocho intervalos de cuatro frecuencias, es decir, treinta y dos valores. La frecuencia que aparece por defecto en el programa de ejemplo es la siguiente: **0 2 3 1 2 1 0 3 1 2 3 0 2 0 3 1 0 3 1 2 3 0 1 2 1 3 2 0 3 2 0 1**

2. “*Stimulation duration*”: Duración (en segundos) del periodo de estimulación. Valor utilizado: **7 s**

3. “*Break duration*”: Duración (en segundos) del periodo de descanso. Valor para la aplicación: **4 s**.

4. “*Flickering delay*”: Tiempo (en segundos) entre la aparición de la flecha que marca el cuadrado para la estimulación y el comienzo del parpadeo. Para la aplicación no hemos requerido este tipo de retardo, por tanto, el valor utilizado ha sido de **0 s**.

5. “*Training Target Size*”: El tamaño del RVS, se configura en proporción al tamaño de la ventana gráfica. Por lo tanto, un tamaño de 0.5; 0.1 significa que el estímulo será un rectángulo con un ancho igual a la mitad del ancho de la ventana gráfica y la altura igual a 1/10 de la altura de esta. La ventana gráfica (área utilizada para la visualización) es el cuadrado más grande que se ajusta a toda la ventana, esto significa que un estímulo con la misma relación de ancho y altura será un cuadrado. Para el experimento se han utilizado cuadrados cuyo tamaño será de **0.3; 0.3**.

6. “*Training Targets’ Positions*”: Posiciones del centro de los estímulos. Hay $N + 1$ estímulos, considerando ‘N’ como el número de frecuencias. El primero es el estímulo que no parpadea. Las posiciones se indican como offsets relativos al centro de la ventana gráfica. Es decir, la posición 1; 1 será su esquina superior derecha y -1; -1 la esquina inferior izquierda. Se han configurado las siguientes posiciones: (0.0; 0.0); (0.0; 0.5); (-0.5; 0.0); (0.5; 0.0).

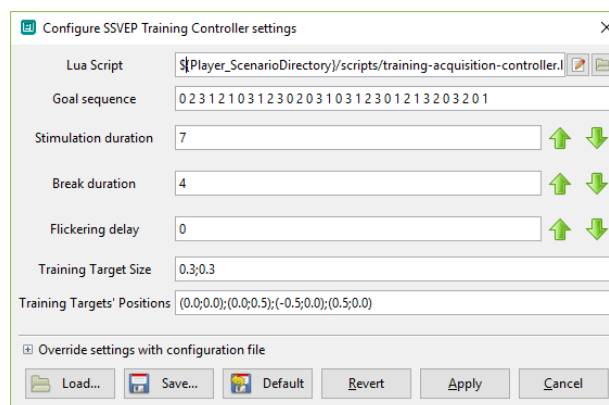


Figura 27. Detalle del bloque configurable “SSVEP Training Controller Box”

Al ejecutar este escenario se genera automáticamente la ventana gráfica, para iniciar la fase de entrenamiento pulsaremos la tecla “SPACE”, y para detenerla pulsaremos “ESC”. Esta quedará de la siguiente forma:

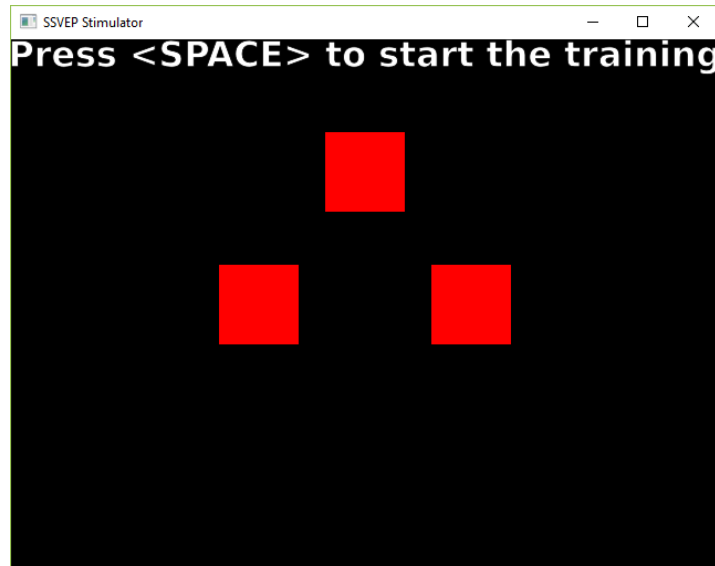


Figura 28. Ventana gráfica de la fase de entrenamiento SSVEP

Cabe destacar la inclusión del bloque “*Stimuli to TCP*” que permite la conexión entre LabVIEW y OpenViBE, tal y como se explica más adelante.

Hasta este escenario sería para realizar la primera parte de nuestro proyecto, es decir, la fase de entrenamiento.

- *ssvep-bci-3-CSP-training.xml* (fig. 29). Una vez se hayan adquirido los datos de entrenamiento, se puede usar este escenario, el cual, se encarga de generar los filtros CSP, que posteriormente serán utilizados para el reconocimiento de cada frecuencia.

El primer bloque que debemos configurar en este escenario es el llamado “*Generic stream reader*”, en el que tendremos que poner el directorio donde se encuentran los datos de la señales EEG procedentes de la fase de entrenamiento del escenario anterior (el archivo “.ov”).

Una vez ejecutemos, el escenario realizará un entrenamiento mediante los filtros “Common Spatial Pattern (CSP)” previamente definidos, para obtener las mejores características de las señales adquiridas.

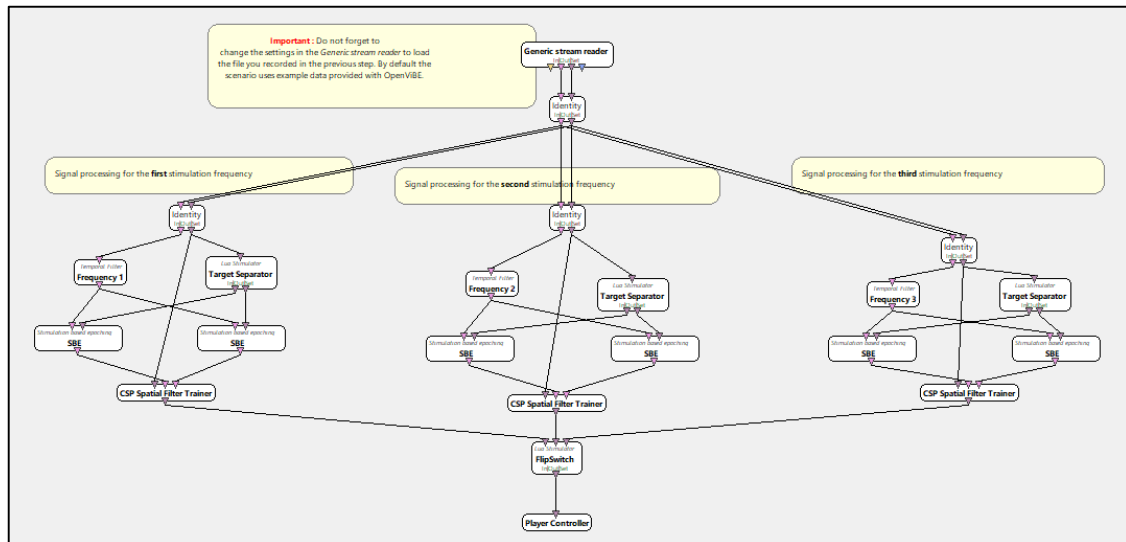


Figura 29. Detalle del escenario OpenViBE “ssvep-bci-3-CSP-training.xml”

- **ssvep-bci-4-classifier-training.xml** (fig. 30). Este escenario crea los clasificadores finales para la prueba online. Al igual que en el escenario anterior, hay que insertar el directorio donde se encuentran las señales EEG adquiridas en la fase de entrenamiento (archivo “.ov”). Cada frecuencia es tratada de forma individual, por tanto, se genera un clasificador para cada una.

Hay dos bloques que conviene comentar para la comprensión de este escenario: el bloque “*Temporal filter*” extrae una sola banda de frecuencia y es configurado por un archivo generado en el escenario 0 de configuración; “*Target Separator*”, se encarga de separar los estímulos (cuadrados/objetivos) para marcar una frecuencia de los otros, y envía un bit a las salidas correspondientes (Objetivo/ No-Objetivo).

Finalmente, el clasificador recibe los valores de amplitud para periodos de estimulación y de no-estimulación, y entrena para poder distinguir uno y otro. Se entrena un clasificador por cada frecuencia de estimulación.

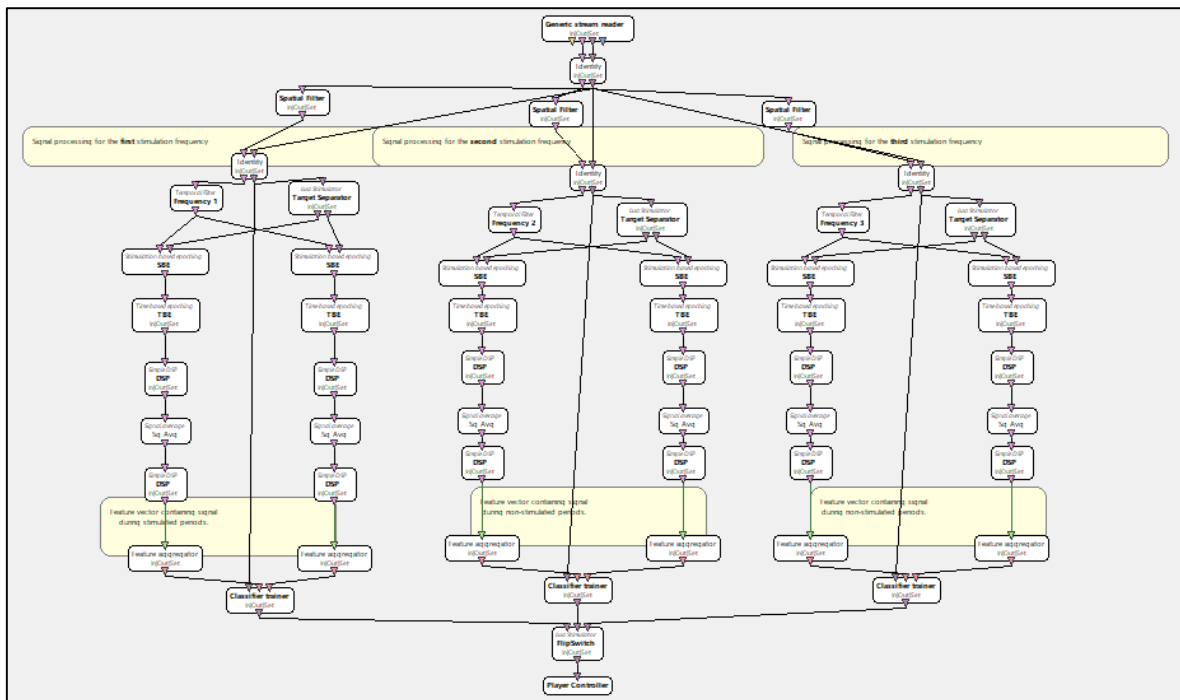


Figura 30. Detalle del escenario OpenViBE “ssvep-bci-4-classifier-training.xml”

- **ssvep-bci-5-online-test-shooter.xml.** (fig. 31). Se trata del último escenario del ejemplo, y es de gran importancia para esta aplicación, ya que forma, junto con un programa desarrollado en LabVIEW, la parte de control del proyecto.

Aunque no se utiliza toda la programación del escenario, hay una parte de reconocimiento de frecuencia que utiliza los filtros y clasificadores de cada frecuencia creados en los escenarios anteriores para averiguar la frecuencia de la señal EEG que le llega.

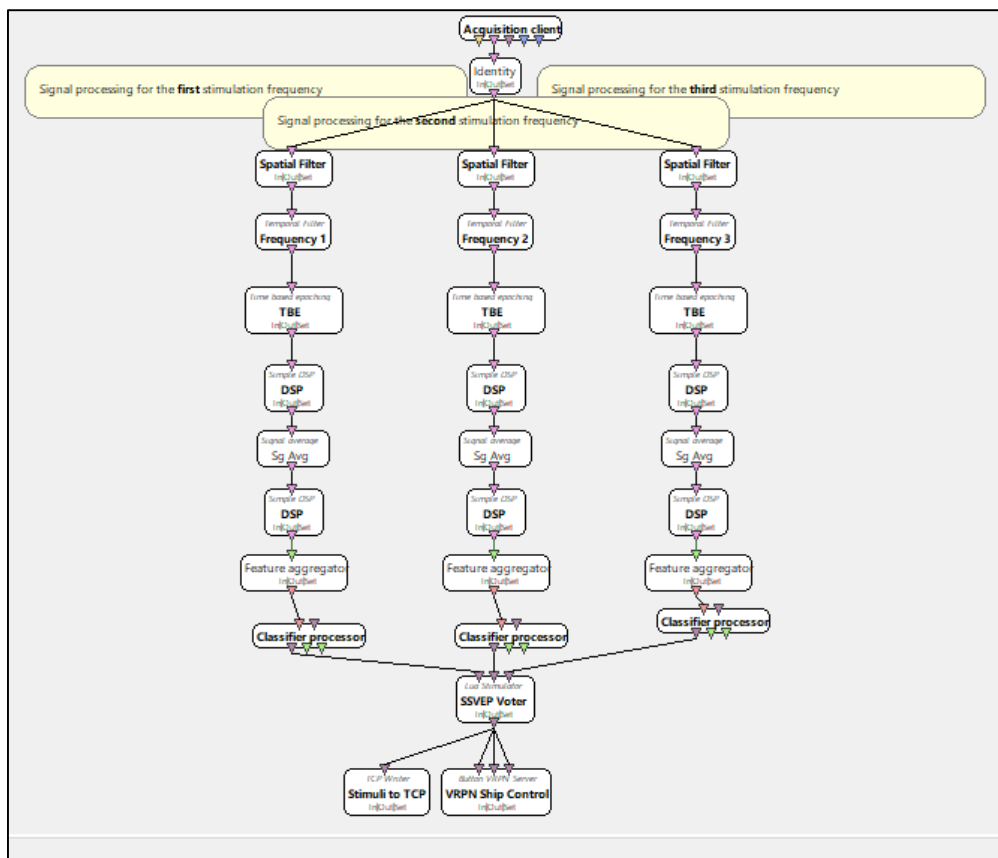


Figura 31. Detalle del escenario OpenViBE “ssvep-bci-5-online-test-shooter.xml”

El primer bloque está recibiendo la señal procedente del “*OpenViBE Acquisition Server*”, es decir, del amplificador Enobio 8. A esta señal EEG, se le aplica inicialmente los tres filtros que se han generado (uno para cada frecuencia entrenada). Para, después de un procesamiento de la señal, sea el clasificador correspondiente el que indique si la frecuencia analizada es la que ha sido entrenada. Por tanto, al bloque “*SSVEP Voter*” le llegará un 1 si se trata de la frecuencia 1, en este caso 20 Hz, un 2 si es la frecuencia 2, 15 Hz, o un 3 si es 12 Hz, que han sido las frecuencias entrenadas. Y el siguiente bloque (“*VRPN Ship Control*”) recibe la siguiente cadena de caracteres en función de la frecuencia que sea.

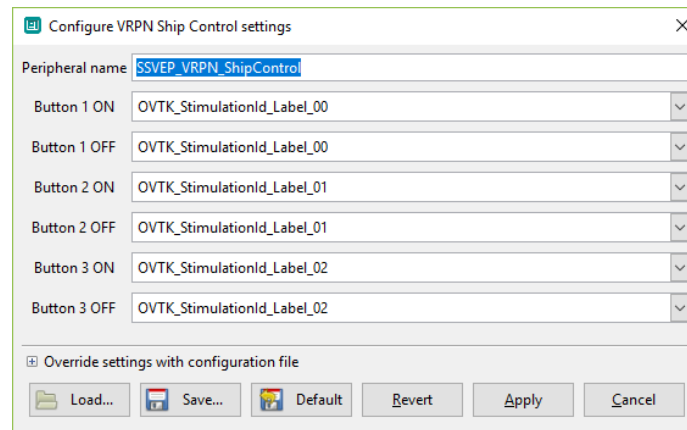


Figura 32. Detalle del bloque “VRPN Ship Control”

Esta cadena de caracteres será enviada a otro programa en LabVIEW gracias a la inclusión del bloque “*Stimuli to TCP*”, tal y como veremos en apartados posteriores.

La programación no utilizada sirve para generar un simple juego de “shooter”, como indica el propio nombre del escenario, para comprobar el funcionamiento. El usuario controla una “figura” representada por dos alas (dos cuadrados parpadeando a diferente frecuencia a los lados) y un cañón (un triángulo al frente parpadeando). El usuario, centrando la mirada en los diferentes estímulos, podrá moverse y disparar a los círculos blancos que irán apareciendo. Para moverse deberá fijar en el ala izquierda para girar la figura en sentido de las agujas del reloj, y en el ala derecha para girar en el sentido contrario; para disparar a los círculos blancos deberá fijarse en el triángulo.

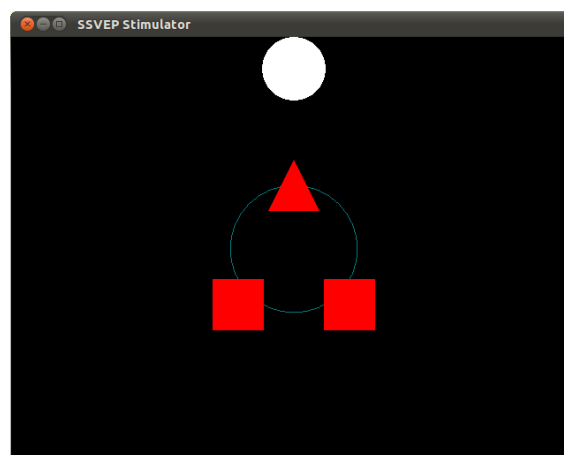


Figura 33. Online shooter generado por el escenario OpenViBE “ssvep-bci-5-online-test-shooter.xml”

4.3. CONEXIÓN ENTRE LABVIEW Y OPENVIBE

La documentación disponible de OpenViBE indica que existen varias formas de comunicarse con otros programas externos. Si nos fijamos en la carpeta de bloques “*Acquisition and network IO*” de OpenViBE disponemos de tres opciones: “*LSL Export*”, “*TCP Writer*” y “*VRPN*”. Tras recoger información acerca de las tres opciones, escogimos la conexión vía protocolo TCP³ al estar suficientemente documentada en la página web de OpenViBE [27], y fácilmente entendible, además de ser suficiente para nuestra aplicación. Para ello se siguió el tutorial disponible en la web del programa [28].

Por tanto, como se ha explicado previamente, la primera parte del proyecto consiste en comparar la fase de entrenamiento de OpenViBE, con los estímulos visuales en la pantalla del ordenador, con la matriz de LEDs, alimentada desde una aplicación en el entorno LabVIEW, así podremos analizar los diferentes resultados obtenidos.

Para ello tendremos que sincronizar ambos programas, es decir, cuando comience la fase de entrenamiento en el escenario correspondiente de OpenViBE, que automáticamente lo haga la aplicación de LabVIEW. La táctica para realizar esta sincronización es la siguiente:

Como hemos comentado en el anterior apartado cuando llegamos al escenario de la fase de entrenamiento del programa de OpenViBE y se ejecute tras configurarla, se genera la ventana gráfica con los RVS. La idea es que al pulsar la tecla “SPACE” que da inicio a la simulación, se envíe un bit (0/1) desde un programa a otro vía protocolo TCP, para que la aplicación en LabVIEW se ejecute automáticamente y se sincronice con la de OpenViBE.

³ Protocolo de Control de Transmisión (TCP): Es uno de los protocolos fundamentales en Internet.

Muchos programas dentro de una red de datos compuesta por redes de computadoras pueden usar este protocolo para crear “conexiones” entre sí, a través de las cuales puede enviarse un flujo de datos.

Con el fin de lograr emitir un bit, se ha añadido en el programa de OpenViBE, escenario “*ssvep-bci-2-training-acquisition.xml*” el bloque “*Stimuli to TCP*” (véase la fig. 26), que recibirá y enviará un pulso o ‘1’ lógico vía TCP cuando le llegue un estímulo procedente del bloque anterior “*Scenario Control*”.

Tendremos que configurar de manera correcta este bloque para que la conexión se realice correctamente. Si abrimos su configuración nos aparecen dos ajustes:

- “*Port*”: Se trata del puerto TCP que aceptará las conexiones del cliente (en este caso el cliente será la aplicación en LabVIEW). El número de puerto que se ha utilizado es el **8001**. Puede ser cualquier número, pero recuerda que debe coincidir con el insertado en el otro programa.

- “*Output format*”: Las opciones son *uint64*, *Hex* (Hexadecimal) o *String* (Cadena de caracteres). En este caso se configura en formato **hexadecimal**.

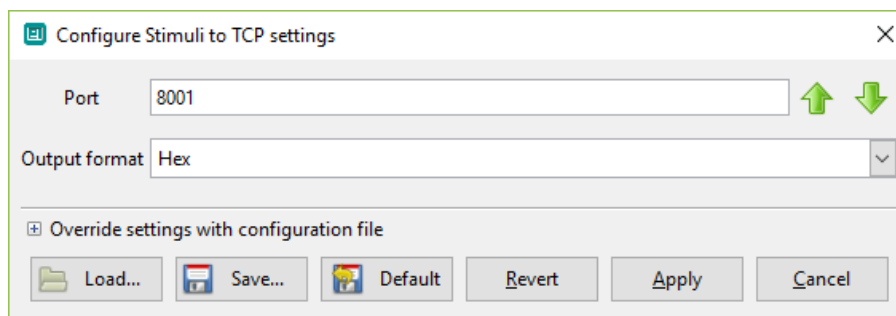


Figura 34. Detalle de la configuración del bloque “*Stimuli to TCP*” del escenario 2.

Además, para la fase de control también haremos uso de la conexión TCP. El escenario 5 de OpenViBE, encargado de reconocer y clasificar cada frecuencia enviará una cadena de caracteres a LabVIEW, para que este conozca la frecuencia de la que se trata. Por tanto, en este caso el ajuste “*Output format*” será configurado como “*String*”.

4.3.1. APLICACIÓN EN LABVIEW PARA LA CONEXIÓN TCP

La programación en LabVIEW para conseguir la conexión de ambos programas es algo más compleja. El programa que se ha desarrollado (véase en la fig. 35) sirve únicamente para la transmisión de datos binarios, es decir, con la finalidad de que OpenViBE o cualquier otro programa transmita un impulso/bit y LabVIEW lo recoja y cambie de estado una variable binaria. Cabe recalcar este asunto, porque en caso de intentar transmitir señales EEG, analógicas... habrá que desarrollar otro programa diferente.

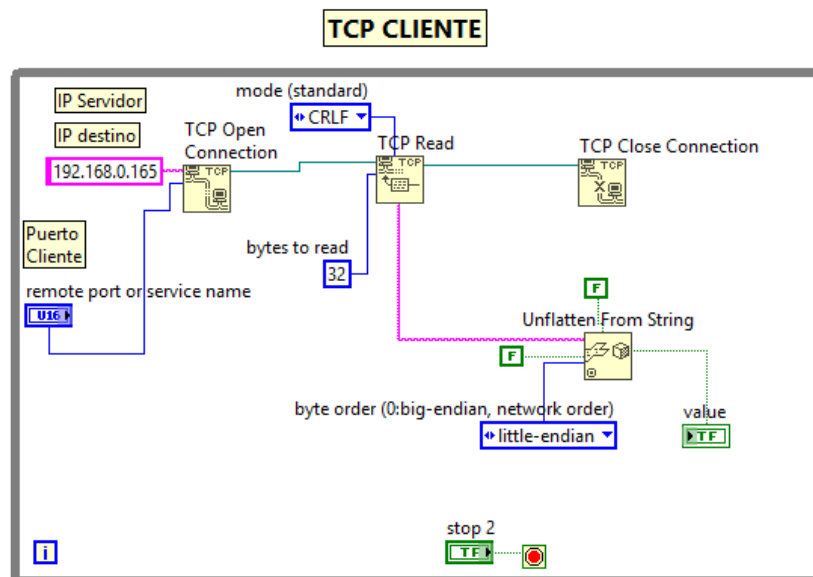


Figura 35. Programa para la transmisión vía TCP de datos binarios entre LabVIEW y otro programa externo

En esta aplicación, al ser OpenViBE el programa que envía el dato será el ‘servidor’, y como LabVIEW se encarga de recogerlo, lo llamamos ‘cliente’. Toda la programación se encuentra dentro de un bucle “while”, para que una vez ejecutado y conectado, se dedique a leer constantemente los valores enviados desde el programa externo.

Primero, colocamos el bloque “*TCP Open Connection*”, al que le tendremos que introducir dos ajustes imprescindibles para lograr la conexión: la “*IP Servidor*”, cuyo valor depende de la red a la que se encuentre conectado el computador con el que trabajamos, este número será común para ambos programas; y el “*Remote port*” que debe ser el mismo número previamente configurado en el programa de OpenViBE.

Seguidamente, pasamos al bloque “*TCP Read*”, encargado de leer y transmitir los datos procedentes del programa externo, se ha configurado para que lea 32 bytes ya que si no da error. Esto se debe a que el programa de OpenViBE envía, además del dato que nos interesa, ocho variables uint32 (número de versión del formato, frecuencia de muestreo, número de canales...), aunque después nos quedaremos solamente con el bit que nos importa. Los datos que lee este bloque los transmite en variable tipo *string* o cadena de caracteres.

El siguiente bloque, “*Unflatten From String*” convierte los datos de entrada, configurados en “*flattened data*”⁴ a código binario, y lo transmite a un indicador binario denominado “*value*”, que será el que muestre el estado del bit que envía OpenViBE para realizar la conexión vía TCP. Véase la interfaz con el usuario en la fig. 36.

El último bloque, “*TCP Close Connection*”, cierra la conexión entre ambos programas.

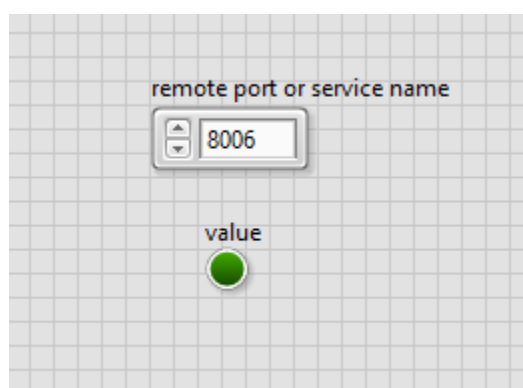


Figura 36. Interfaz de usuario programa de conexión vía TCP entre LabVIEW y otro programa externo

⁴ Flattened Data: LabVIEW convierte los datos del formato en memoria a una forma más cómoda para tratar, escribir o leer desde un archivo. Este formato LabVIEW lo denomina “flattened data”.

Como se puede apreciar, predomina la sencillez, y la comprensión es simple, el usuario inicialmente deberá introducir el número de puerto correspondiente, posteriormente, una vez ejecutemos el programa, OpenViBE se conectará con LabVIEW, y cuando envíe un impulso, este indicador digital se encenderá.

En el caso de enviar una cadena de caracteres la programación cambia ligeramente. En el bloque “*Unflatten from String*” en vez de convertir los datos a código binario los pasamos a cadena de caracteres (*string*). Entonces, en la interfaz gráfica en vez de tener un indicador digital, tendremos un indicador de string.

5. DESCRIPCIÓN DETALLADA DEL PROYECTO

A lo largo de esta sección se profundizará en el conjunto del proyecto, es decir, se combinan las aplicaciones ya desarrolladas, así veremos su función en la aplicación final que se describe a continuación.

5.1. OBJETIVO DEL PROYECTO

El objetivo principal del proyecto es el diseño eficiente y robusto de una interfaz cerebro-computadora basada en el paradigma SSVEP que permita a un sujeto interactuar con el mundo externo sin involucrar ningún músculo, únicamente centrando la mirada en un estímulo visual repetitivo.

Para ello, tendremos que encontrar la solución a varios problemas que se irán presentando a lo largo del desarrollo de la aplicación.

Primero, se programará una fase de entrenamiento necesaria para generar los filtros y el clasificador, que permitirán el correcto funcionamiento de la aplicación final.

La aplicación final estará compuesta por tres matrices de diodos LEDs de color rojo y parpadeando a una frecuencia diferente, los cuales actuarán de estímulos visuales repetitivos. El sujeto, que tenga puesto el sensor Enobio 8, tendrá que fijarse en una matriz de LEDs cualesquiera, dependiendo de la matriz en la que centre la mirada, un LED de distinto color se encenderá, haciendo de actuador.

Además se elaborarán unas tablas con los resultados de un pequeño experimento realizado con diferentes sujetos, con el que obtener nuestras propias conclusiones acerca de la robustez del diseño.

Esta aplicación es una simulación más elemental de otras mucho más complejas, como podría ser una silla de ruedas controlada por una BCI basada en SSVEP, o la domótica de una vivienda controlada por BCI... en las que el sistema de control sería similar al desarrollado en este proyecto, solo que cambiaría el número de entradas (RVS) y salidas.

5.2. FUNCIONAMIENTO GENERAL DE LA APLICACIÓN

Para lograr el funcionamiento final de la aplicación es necesaria la inclusión de una fase de entrenamiento que genere los filtros y el clasificador, los cuales se encargarán de reconocer la frecuencia del RVS a la que el usuario se está fijando.

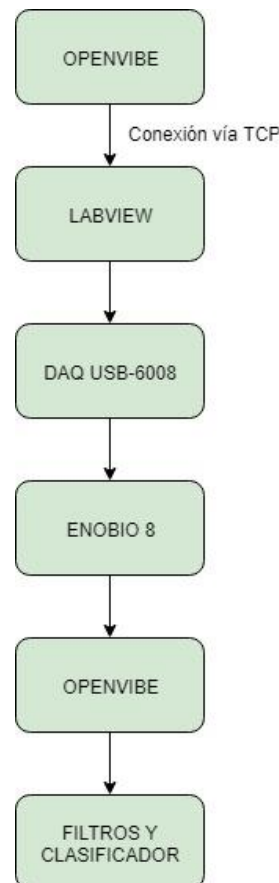


Figura 37. Diagrama de funcionamiento de la fase de entrenamiento

La fase de entrenamiento se inicia con el programa desarrollado en OpenViBE, una vez se ha verificado que el amplificador Enobio 8 está adquiriendo las señales EEG del cerebro del sujeto, ejecutamos el OpenViBE, que mandará una señal vía TCP al programa desarrollado en LabVIEW, que iniciará una fase de entrenamiento similar y totalmente sincronizada con la de OpenViBE, pero en este caso utilizando como RVS las matrices de LEDs.

Tras realizar la fase de entrenamiento, el escenario 2 del programa de OpenViBE genera un archivo .ov que registra los datos recogidos durante el entrenamiento. Este será el archivo que debemos insertar en los dos escenarios siguientes para que se generen los filtros CSP y el clasificador.

La creación de estos filtros y el clasificador permitirá realizar la fase de control en el último escenario.

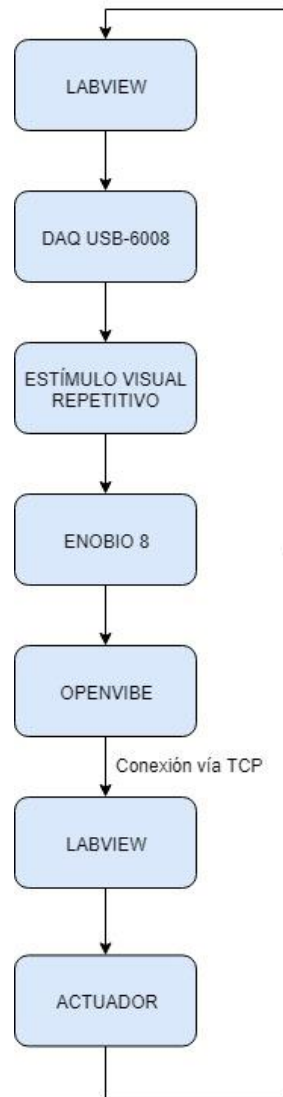


Figura 38. Diagrama de funcionamiento de la etapa de control

Una vez se han generado los filtros CSP y el clasificador de las frecuencias entrenadas podemos iniciar la etapa de control.

Primero, se forman los estímulos visuales repetitivos en las matrices de LEDs rojos, gracias a un programa en LabVIEW que permite controlar la frecuencia de cada RVS en todo momento, y que, mediante la tarjeta de adquisición de datos USB-6008, alimenta dichas matrices.

El sujeto, que ya tendrá correctamente colocado el casco, centrará la mirada en la matriz de LEDs que el desee, entonces, el último escenario del programa en OpenViBE se encargará, mediante el uso de los filtros y el clasificador previamente generados, de reconocer la frecuencia del RVS (si se trata de una de las frecuencias entrenada) y, mediante la conexión TCP enviará una determinada señal al programa desarrollado en LabVIEW. Este programa, recibirá la frecuencia reconocida, y actuará encendiendo un LED, en función de esta.

Los siguientes apartados profundizan la explicación de cada fase para lograr la comprensión correcta y completa.

5.3. FASE DE ENTRENAMIENTO

La primera parte del proyecto consiste en realizar una fase de entrenamiento similar a la programada en OpenViBE para generar los filtros espaciales y el clasificador que reconozca cada frecuencia en la etapa de control. Por tanto, se ha desarrollado un programa en LabVIEW que realiza la misma fase de entrenamiento que el OpenViBE, además para que el OpenViBE pueda recoger los resultados al mismo tiempo, se ha sincronizado ambas fases.

La explicación es la siguiente, queremos “engañar” al OpenViBE, es decir, que recoja los datos del sensor Enobio 8 mientras el sujeto lo entrena con los LEDs como RVS, en vez de en la pantalla del computador. Entonces, se entiende que ambos programas deban ejecutarse totalmente sincronizados. Por ejemplo, cuando la flecha amarilla que indica la frecuencia a la que se tiene que fijar el sujeto marque el cuadrado de 20 Hz por la pantalla, la matriz de LEDs rojos deberá parpadear a 20 Hz también. Luego, hay que configurar el programa en LabVIEW para que la secuencia de frecuencia, el tiempo de simulación y el tiempo de descanso sea similar al de OpenViBE.

Por tanto, se ha desarrollado en LabVIEW la siguiente aplicación (fig. 39), llamada “*Fase de entrenamiento.vi*”, compuesta por una máquina de estados que consta de 33 estados. Esta programación en máquina de estados se utiliza muchísimo en este entorno de programación, y se forma por una estructura “*While*”, junto una “*Case*”, añadiendo un “*Shift Register*”. Los casos de la estructura *case* serán los diferentes estados por los que pasará nuestro programa.

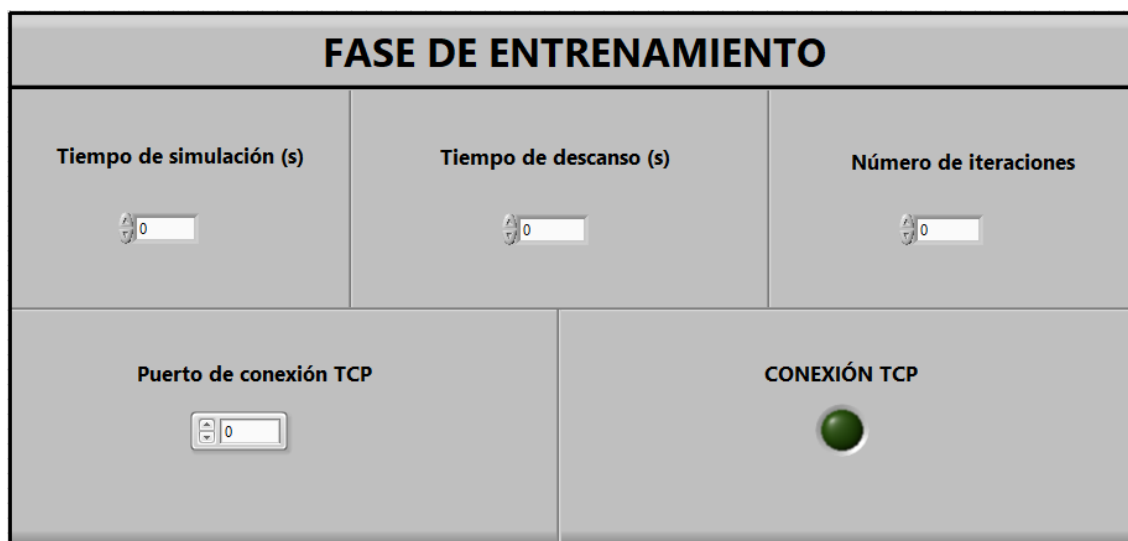


Figura 39. Interfaz gráfica del programa “Fase de entrenamiento.vi”

La interfaz con el usuario se ha simplificado al máximo para poder hacer accesible esta aplicación a cualquier tipo de persona, ya que no tiene que introducir las frecuencias y, además, se puede apreciar que el usuario dispone de total libertad para configurar esta fase, lo que siempre debe tener presente es que estos ajustes deben ser similares a los configurados en la aplicación de OpenViBE. El “*tiempo de simulación*”, es el tiempo de parpadeo de cada RVS en cada iteración, recordemos que en OpenViBE era de **7 s**. El “*tiempo de descanso*” es el tiempo que no parpadeará el RVS en cada iteración, **4 s**. El “*número de iteraciones*” marcará el tiempo total de la fase de entrenamiento. En un principio la fase está programada para las 32 iteraciones que aparecen en OpenViBE por defecto, sin embargo, puede resultar algo longeva y pesada para el usuario, ya que duraría unos seis minutos y medio. Por tanto, reduciendo el número de iteraciones de acortará dicho tiempo, y resultará más dinámico. Pero cuanto mayor sea la duración del entrenamiento más precisa será esta fase. Recordar que si reducimos las iteraciones, deberemos realizar la misma variación en el programa de OpenViBE.

Además el usuario debe introducir el número del puerto para realizar la conexión TPC, que debe ser el mismo que el insertado en el bloque “*Stimuli to TCP*” del escenario 2 (fig. 26). El indicador digital “*Conexión TCP*” se activará cuando se inicie la fase y el OpenViBE envíe un impulso a este programa.

Como se ha comentado, la aplicación está compuesta por 33 estados. El estado 0 [29], es el inicial, y su función es la de realizar la conexión entre OpenViBE y LabVIEW, para que una vez que reciba el impulso pase al estado siguiente y se inicie la fase de entrenamiento. Se puede apreciar la inclusión de un *delay* para sincronizar ambos programas y que no empiece uno antes que otro. Por otro lado, en este estado los LEDs deberán permanecer apagados hasta que entre la señal de OpenViBE, por lo que, de momento no se generará ningún tren de pulsos (frecuencia 0 igual a 0). Además este estado inicializa las demás entradas, haciendo una puesta a 0, para que permanezcan apagados en caso de que algún RVS se hubiera quedado encendido.

Una vez se haya iniciado la fase de entrenamiento en OpenViBE (pulsando la tecla *space*), el indicador digital se encenderá, guardando un '1' lógico, esto desencadenará que el programa pase del estado 0 al 1. Los siguientes 32 estados describen la secuencia anteriormente programada en OpenViBE (recordemos 0 2 3 1 2 1 0 3 1 2 3 0 2 0 3 1 0 3 1 2 3 0 1 2 1 3 2 0 3 2 0 1, donde 0 (0 Hz), 1 (20 Hz), 2 (15 Hz), y 3 (12 Hz)), para ello hemos hecho uso del programa explicado en la sección 4.1. Como se acaba de explicar, el usuario tiene la opción de acortar el tiempo de esta fase de entrenamiento, aunque se pierda algo de precisión.

Una vez se inicie el entrenamiento, el usuario debe centrar su mirada en la matriz de LEDs que corresponda, esta empieza por el RVS situado más a la derecha e irá pasando hacia la izquierda. En cada iteración solo se encontrará parpadeando el RVS al que el usuario debe mirar, lo que facilita y simplifica la realización de esta fase de entrenamiento.

Cuando se detenga automáticamente el programa, ya habrá terminado la fase de entrenamiento, y el archivo .ov se habrá creado y guardado en el directorio indicado en el escenario 2 del programa de OpenViBE. Posteriormente, cargaremos el archivo .ov generado en el escenario 3 y lo ejecutaremos. Esto durará el mismo tiempo que la etapa de entrenamiento, y una vez se detenga automáticamente, se habrán generado los tres filtros espaciales (cada uno se corresponde con una de las frecuencias). Los archivos generados y que necesitaremos para la etapa de control se guardan con los siguientes nombres: *csp-1.cfg*; *csp-2.cfg*; *csp-3.cfg*. Cabe destacar, que a no ser que variaremos el directorio o el nombre de estos archivos se sobrescribirán los datos cada vez que ejecutemos este escenario.

De forma equitativa, cargaremos el archivo .ov de la fase de entrenamiento en el escenario 4 y lo ejecutaremos. Esto generará tres archivos: *classifier-1.cfg*; *classifier-2.cfg*; *classifier-3.cfg*. Los cuales contienen las características del clasificador para cada frecuencia. Lo que permitirá reconocer cada frecuencia en el último escenario. Al igual que en el escenario anterior, cada vez que ejecutamos, los datos anteriores se eliminan y se sobrescriben los nuevos.

Además, se ha desarrollado en OpenViBE un escenario que nos permite analizar los datos adquiridos por el entrenamiento de cada sujeto. El programa se ha llamado “*Análisis entrenamiento.xml*” (fig. 40) y muestra, mediante la inclusión de un osciloscopio, los picos de amplitud que se han generado en cada una de las frecuencias entrenadas (12, 15 y 20 Hz en nuestro caso).

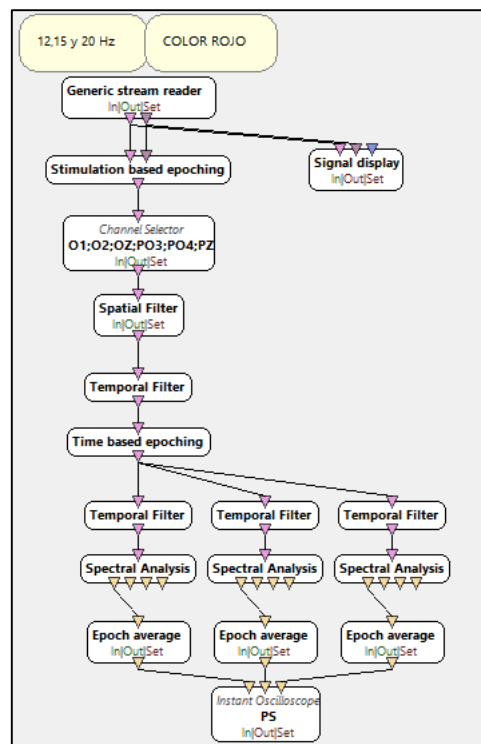


Figura 40. Detalle del escenario OpenViBE “Análisis entrenamiento.xml”

La aplicación se encuentra totalmente configurada, lo único que hay que realizar para poder obtener el análisis de los datos adquiridos es cargar el archivo ‘.ov’ creado tras el entrenamiento en el bloque “*Generic stream reader*”, al igual que en los escenarios 2 y 3 del otro programa.

Se ha programado para que solo tenga en cuenta los siguiente electodos del Enobio 8: O1; O2; Oz; PO3; PO4; Pz; mediante el bloque “*Channel selector*”. Ya que son los que están localizados en la zona occipital, y, por tanto, los que muestran de una forma más notoria dichos picos de frecuencia.

Posteriormente, la señal pasa por un filtro espacial y otro temporal. Finalmente se le aplica un análisis espectral a cada una de las frecuencias y así podremos obtener los picos de frecuencia generados por cada usuario.

Hay que entender la importancia del análisis de estos picos de frecuencia, como se ha explicado, el paradigma SSVEP se detecta por la aparición de estos picos en la frecuencia del EGG del sujeto. Por ejemplo, si una persona se está fijando en un RVS de 12 Hz, su EEG tendrá un pico más alto en 12 Hz. Bien, esto analizaremos en este programa.

El análisis de los datos del entrenamiento ofrece una idea de lo que puede ser la fase de control. Cada sujeto generará un archivo “.ov” distinto al de otro, por tanto, realizando este análisis se podrá adelantar si se prevé una fase de control correcta o complicada. Por ejemplo, si vemos en este análisis que un sujeto tiene un pico mucho más alto en 20 Hz en comparación con el que tiene en 12 Hz, la fase de control podría dar problemas y reconocer 20 Hz cuando el sujeto no este fijándose en dicho RVS. Lo ideal sería obtener tres picos de frecuencia con una amplitud lo más parecida posible.

5.4. ETAPA DE CONTROL

Una vez terminada la fase de entrenamiento y generados los filtros y clasificadores correspondientes se puede comenzar con la etapa de control. Esta etapa utiliza dos programas desarrollados en LabVIEW: *Generador RVS.vi* (fig. 41) y *Aplicación Control BCI – SSVEP.vi* (fig. 42); además del escenario 5 de la aplicación en OpenViBE anteriormente descrita.

Para realizar la etapa de control, el primer paso consiste en generar los RVS, de manera que se ha desarrollado en LabVIEW el siguiente vi: *Generador RVS.vi*, que utilizando la DAQ USB-6008, configura las matrices de LEDs con una frecuencia controlada por el usuario.

GENERADOR DEL ESTÍMULO VISUAL REPETITIVO		
RVS 1	RVS 2	RVS 3
Seleccione un pin de la DAQ: <input type="text" value="L%"/>	Seleccione un pin de la DAQ: <input type="text" value="L%"/>	Seleccione un pin de la DAQ: <input type="text" value="L%"/>
Frecuencia 1 (Hz) <input type="text" value="0"/>	Frecuencia 2 (Hz) <input type="text" value="0"/>	Frecuencia 3 (Hz) <input type="text" value="0"/>
<input type="button" value="STOP"/>	<input type="button" value="STOP"/>	<input type="button" value="STOP"/>

Figura 41. Interfaz gráfica del programa “Generador RVS.vi”

El sujeto se encarga de configurar la línea de la USB-6008 donde va conectado cada RVS, y puede controlar en todo momento la frecuencia del estímulo. Además se incluye un botón de stop individual para cada RVS, que los detendrá si es pulsado.

El diagrama de bloques del programa [30] genera tres trenes de pulsos distintos siguiendo la programación descrita en la sección 4.1., la única diferencia es que en este caso en vez de pararse el programa automáticamente, es el usuario el que lo detiene, pulsando los botones de stop.

Una vez se hayan configurado los RVS, se podrá pasar a la aplicación que realiza la fase de control: *Aplicación Control BCI – SSVEP.vi*.




ETAPA DE CONTROL		
Puerto de conexión TCP → <input type="text" value="0"/>		
ACTUADOR 1 (12 Hz)	ACTUADOR 2 (15 Hz)	ACTUADOR 3 (20 Hz)
Seleccione un pin de la DAQ para el LED 1: <input type="text" value="1"/>	Seleccione un pin de la DAQ para el LED 2: <input type="text" value="0"/>	Seleccione un pin de la DAQ para el LED 3: <input type="text" value="0"/>
Frecuencia 12 Hz 	Frecuencia 15 Hz 	Frecuencia 20 Hz 
<input type="button" value="STOP"/>		

Figura 42. Interfaz gráfica del programa “Aplicación Control BCI – SSVEP.vi”

El usuario debe insertar el mismo número de puerto que en el escenario 5 del programa en OpenViBE (fig. 31). Y deberá seleccionar los tres pines del puerto digital donde irán conectados los LEDs que hará de actuadores, no podrán utilizarse los pines configurados anteriormente como RVS.

Si analizamos la programación interna desarrollada en la ventana del diagrama de bloques de LabVIEW [31] vemos que se ha desarrollado ha desarrollado una máquina de estados con cinco casos:

1. “**Inicialización**” [31]: El primer estado realiza una función de inicialización, es decir, pone a “0” todas los actuadores. Este estado solo se realiza una vez al ejecutar el programa, al principio del todo. Sirve para apagar los LEDs en el caso de que alguno se haya quedado encendido por una mala ejecución del programa.
2. “**Reconocimiento frecuencia**” [32]: Es el estado más importante de la aplicación, y el que verdaderamente ejerce la función de control. Primero realiza una puesta a “0” de las variables binarias que indican el reconocimiento de una frecuencia. Pero lo importante de este estado es que realiza la conexión TCP con OpenViBE, una vez el escenario 5 reconozca una de las tres frecuencias entrenadas enviará una de la siguientes cadena de caracteres:

- “*OVTk_StimulationId_Label_02*” → 12 Hz

- “*OVTk_StimulationId_Label_01*” → 15 Hz

- “*OVTk_StimulationId_Label_00*” → 20 Hz

El programa la recibe y en función de la que sea, encenderá el indicador digital correspondiente indicando la frecuencia reconocida y pasará al siguiente estado. Si se trata de 12 Hz continuará con el estado “**Frecuencia 1**”, si son 15 Hz “**Frecuencia 2**”, y si son 20 Hz “**Frecuencia 3**”.

3. “**Frecuencia X**” [33]: La función de estos tres estados es encender durante cinco segundos el LED asociado a la línea de la DAQ que ha sido configurada previamente. Esto indicará que el usuario se ha fijado en la frecuencia determinada. Una vez han transcurrido cinco segundos, se apaga el LED y el programa vuelve al estado de “Reconocimiento de frecuencia”.

Para detener el programa se ha introducido un botón de stop que termina con la ejecución.

Una vez se ha explicado la función de todos los programas y se entiende el funcionamiento de cada uno podemos pasar al experimento para verificar si la aplicación funciona correctamente.

6. EXPERIMENTO

Durante este apartado se describe como realizar el experimento paso a paso y así comprobar el correcto funcionamiento y la robustez de la aplicación diseñada.

Antes de comenzar se debe establecer la posición de los electrodos en el casco Enobio 8. Como hemos señalado anteriormente este sensor cuenta con 8 canales, por tanto, serán 8 electrodos los que se han de posicionar para una correcta lectura de las señales EEG cerebrales.

Por otro lado, se ha visto que el SSVEP se considera como una respuesta directa en la corteza visual primaria también conocida como zona occipital, por tanto, la mayoría de electrodos serán colocados en esta zona. La posición de los electrodos para esta aplicación es la siguiente: **Canal 1 → Fz; Canal 2 → O1; Canal 3 → O2; Canal 4 → Oz; Canal 5 → PO3; Canal 6 → PO4; Canal 7 → Pz; Canal 8 → Cz;**

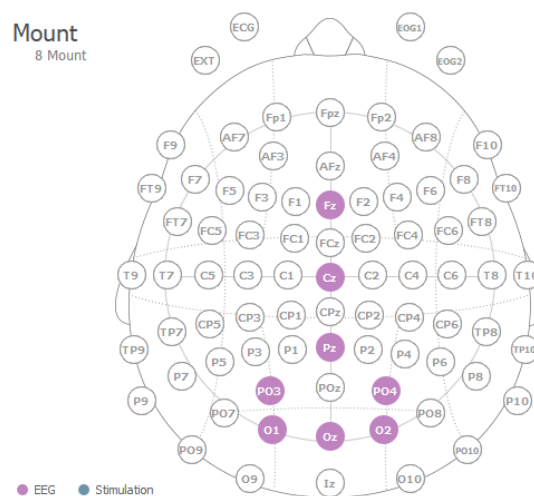


Figura 43. Configuración de los electrodos en el Enobio 8

Una vez se hayan posicionado los electrodos correctamente el sujeto podrá colocarse el sensor. Un consejo para hacerlo de forma correcta es medir la cabeza del usuario con un metro desde el centro de la frente al hueco en el que desemboca la zona occipital, esta medida de una cabeza humana suele ser entre 36 y 38 cm. Al colocárselo al sujeto el electrodo 8 (Cz) debe situarse en la mitad de esta medida.

Una vez colocado correctamente el casco, se ha de poner la pinza CMS-DRL en una de las orejas del usuario.

La conexión del Enobio 8 con el computador se realiza vía Bluetooth introduciendo la siguiente dirección MAC: **00:07:80:34:EC:15**. Cuando el sensor esté conectado al computador en el que se va a realizar la prueba abriremos el software de Neuroelectrics, *NIC2*. Este programa permite, mediante el desarrollo de un protocolo propio que contiene la configuración utilizada de los ocho canales, verificar que todos los electrodos están realizando contacto y está leyendo los diferentes EEG:



Figura 44. Verificación del funcionamiento de los electrodos, NIC2

Desde que conectamos y abrimos la aplicación pueden pasar varios segundos hasta que se verifica el correcto contacto de todos los canales. Una vez veamos que los pequeños círculos de arriba de la derecha están en color verde y que además estamos visualizando los EEG de cada canal verificamos que el sensor está realizando su labor de adquisición correctamente y podremos pasar al siguiente paso. Cabe insistir en que es muy importante que todos los electrodos hagan contacto correctamente, ya que un fallo en un solo electrodo puede derivar en el mal funcionamiento de la aplicación.

Ahora, comenzamos con los programas desarrollados. Primero conectaremos el Enobio 8 al *OpenViBE Acquisition Server*. Para ello, en la pestaña “*Driver*” se selecciona la opción “*Enobio3G*” y en “*Driver Properties*” la configuración es la siguiente:

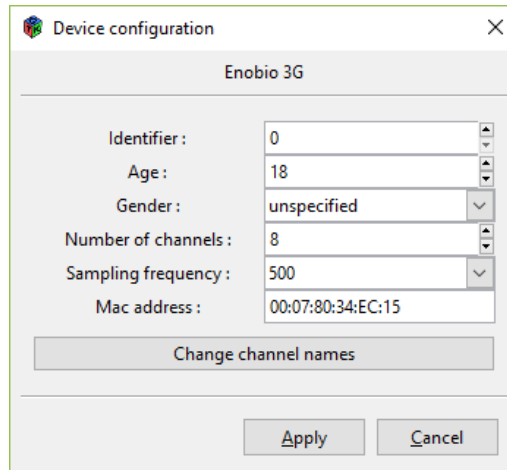


Figura 45. Ajustes de la pestaña “*Driver Properties*”

Es muy recomendable nombrar los canales para evitar futuras confusiones y saber que EEG se trata de cada canal, por tanto, en “*Change channel names*”:

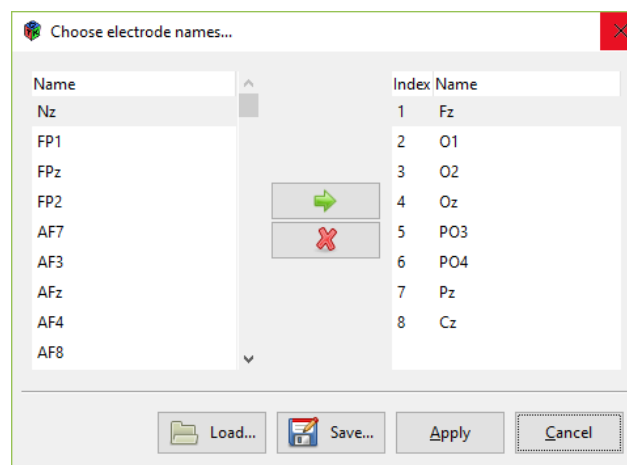


Figura 46. Configuración de los nombres de los canales “*Change channel names*”

Una vez realizados estos ajustes se realiza la conexión.

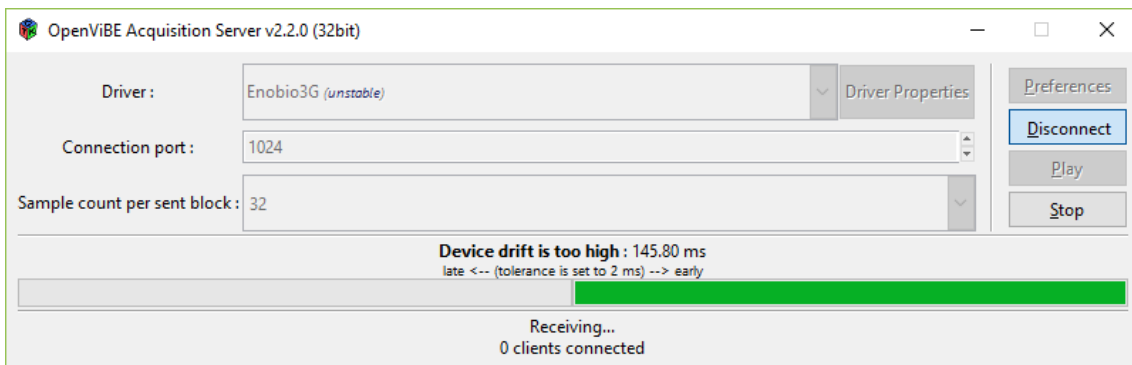


Figura 47. Conexión Enobio 8 – OpenViBE

Ya realizada la conexión, abrimos el programa desarrollado de OpenViBE. Y comienza la primera fase: el entrenamiento.

Primero, para confirmar que el *OpenViBE Designer* está recibiendo las señales EEG procedentes del Enobio 8, ejecutamos el escenario 1 de nuestro programa. Obteniendo el siguiente resultado:

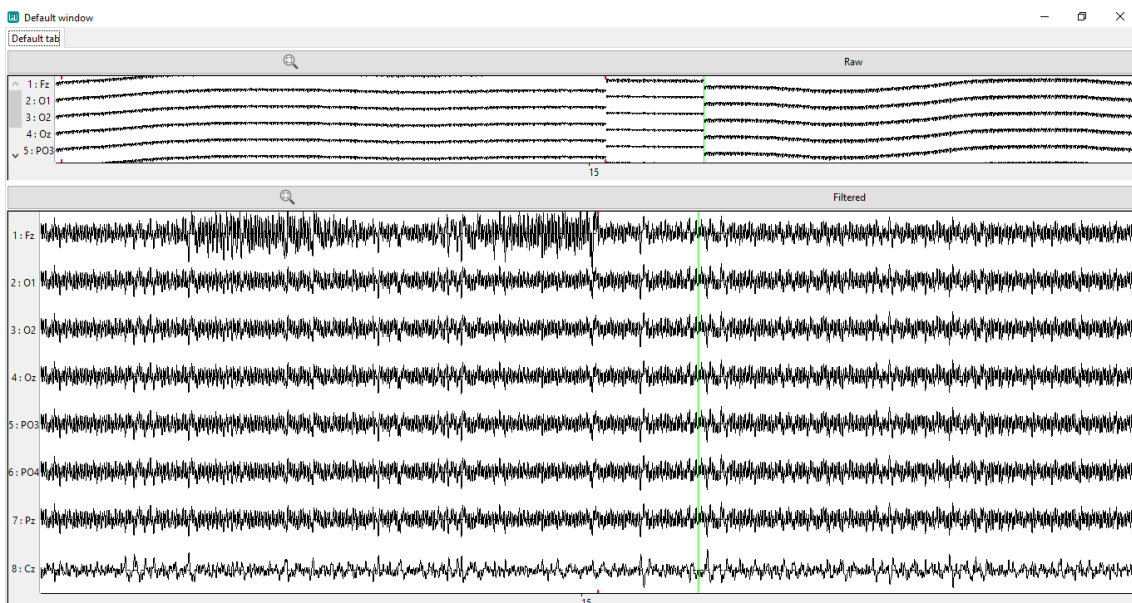


Figura 48. Adquisición de señales EEG en el programa de OpenViBE

Se puede apreciar que se está recibiendo correctamente la señal EEG de cada canal.

Pasando al siguiente escenario, con los ajustes definidos en la sección 4.2, lo ejecutamos para configurar la **fase de entrenamiento**.

Para iniciar la fase de entrenamiento, se empieza conectando las matrices de LEDs (RVS) a las líneas de la DAQ USB-6008 que seguidamente definiremos en la aplicación de LabVIEW “*Fase de entrenamiento.vi*”. Posteriormente, se configuran los ajustes de la aplicación similares a los introducidos en OpenViBE, tal y como se ha explicado en secciones anteriores (ver apartado 4.2.). Para este proyecto se ha ajustado la fase de entrenamiento de la siguiente manera:


FASE DE ENTRENAMIENTO			
Tiempo de simulación (s)	Tiempo de descanso (s)	Número de iteraciones	
<input type="text" value="7"/>	<input type="text" value="4"/>	<input type="text" value="32"/>	
Puerto de conexión TCP		CONEXIÓN TCP	
<input type="text" value="8001"/>			

Figura 49. Ajustes para la fase de entrenamiento

Antes de comenzar el entrenamiento el sujeto, ya con el sensor puesto, debe encontrarse en una posición cómoda, evitando malas posturas y distracciones. La distancia ideal entre el sujeto y los RVS es de 60 cm. Además, hay que tener en cuenta la iluminación de la sala donde se produzca la prueba, cuanto menos luz hay mejor resultados se obtendrán. Recordar también establecer el directorio donde se creará el archivo ‘.ov’, en el bloque “*Generic stream writer*” (fig. 26).

Una vez realizados todos los ajustes ejecutamos el escenario 3, una vez el programa ha generado la ventana emergente con los RVS en el computador, e importante, antes de darle a la tecla 'space', ejecutamos la aplicación de LabVIEW. Una vez el vi esté ejecutándose, se pulsa 'space' y se iniciara la fase de entrenamiento.

El entrenamiento comienza en la matriz de LEDs situada más a la derecha, parpadea durante 7 segundos a la frecuencia que le toque según la secuencia introducida (ver fig. 27), y se detiene 4 segundos de descanso. Seguidamente comienza a parpadear el RVS situado a su izquierda durante el mismo intervalo, y cuando acabe hará lo mismo el tercer estímulo. El sujeto debe ir fijando la mirada en los estímulos que parpaddeen en cada momento (van de derecha a izquierda). Se ilustran las siguientes imágenes que facilitan la comprensión.

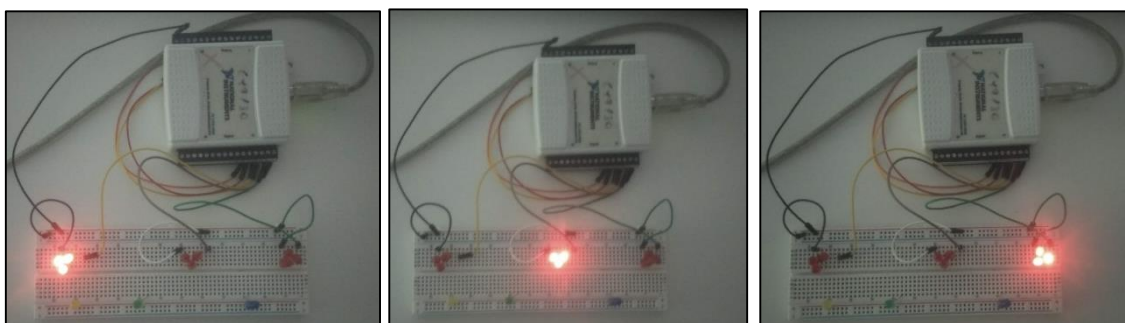


Figura 50. Ilustraciones de la fase de entrenamiento

Recordar que en algunas iteraciones la frecuencia es 0 Hz, por tanto, al RVS que le tocara encenderse permanecerá apagado, cuando transcurra el tiempo programado se pasará a la siguiente iteración cambiando de estímulo.

Una vez transcurran las 32 iteraciones, el programa se cerrará automáticamente y la fase de entrenamiento habrá terminado.

Lo importante de esta fase de entrenamiento es la creación del archivo '.ov'. Por tanto, mediante el programa "Análisis entrenamiento.xml" (fig. 40), ya se puede analizar y valorar como ha sido el entrenamiento del sujeto. Además de anticipar una hipótesis acerca del sistema de control.

Antes de pasar a la fase de control, hay que ejecutar el escenario 3 para generar en entrenar los filtros CSP, y el escenario 4 para entrenar al clasificador de cada frecuencia. Para ello debemos introducir en cada escenario el archivo ‘.ov’ generado en el entrenamiento.

Una vez generados y entrenados los filtros y clasificadores de cada frecuencia se podrá iniciar la **etapa de control**.

Primero iniciamos la aplicación de LabVIEW, “**Generador RVS.vi**”, de manera que se genere un tren de pulsos de diferente frecuencia para cada matriz de LEDs. La configuración utilizada para este experimento se detalla en la siguiente figura:

GENERADOR DEL ESTÍMULO VISUAL REPETITIVO		
RVS 1	RVS 2	RVS 3
Seleccione un pin de la DAQ: <input type="text" value="Dev1/port0/line0"/>	Seleccione un pin de la DAQ: <input type="text" value="Dev1/port0/line1"/>	Seleccione un pin de la DAQ: <input type="text" value="Dev1/port0/line2"/>
Frecuencia 1 (Hz) <input type="text" value="12"/>	Frecuencia 2 (Hz) <input type="text" value="15"/>	Frecuencia 3 (Hz) <input type="text" value="20"/>
<input type="button" value="STOP"/>	<input type="button" value="STOP"/>	<input type="button" value="STOP"/>

Figura 51. Ajustes para la generación de los RVS

De esta manera, al ejecutarlo el RVS la derecha parpadeará a una frecuencia de 12 Hz, el del centro a 15 Hz y el de más a la izquierda a 20 Hz.

Tras generar los estímulos, pasamos a configurar el vi “*Aplicación Control BCI – SSVEP.vi*”, tal y como indica la siguiente figura:




ETAPA DE CONTROL		
Puerto de conexión TCP → <input type="text" value="8002"/>		
ACTUADOR 1 (12 Hz)	ACTUADOR 2 (15 Hz)	ACTUADOR 3 (20 Hz)
Seleccione un pin de la DAQ para el LED 1: <input type="text" value="Dev1/port0/line3"/>	Seleccione un pin de la DAQ para el LED 2: <input type="text" value="Dev1/port0/line4"/>	Seleccione un pin de la DAQ para el LED 3: <input type="text" value="Dev1/port0/line5"/>
Frecuencia 12 Hz 	Frecuencia 15 Hz 	Frecuencia 20 Hz 
<input type="button" value="STOP"/>		

Figura 52. Ajustes para la fase de control

Se ha configurado para que si el sujeto fija su mirada en el RVS de 12 Hz se active el LED azul que tiene enfrente, si se trata de 15 Hz se activa el LED verde y si son 20 Hz el amarillo.

Tras este ajuste, se pasa al escenario 5 del programa de OpenViBE y, el único que cambio que hay que introducir es el número de puerto en el bloque “*Stimuli to TCP*”, que tendrá que ser 8002.

Una vez hecho este cambio ya se podrá llevar a cabo la ejecución del sistema. Previamente, se hará los mismos ajustes con el Enobio 8 que los realizados para la fase de entrenamiento (distancia sobre los RVS, comodidad...). Ejecutaremos el escenario 5 y acto seguido lo haremos con el “*Aplicación Control BCI – SSVEP.vi*”. Pueden darse las siguientes situaciones:

- 1) El sujeto centra la mirada en el RVS de 12 Hz, el LED azul se activa durante 5 s:

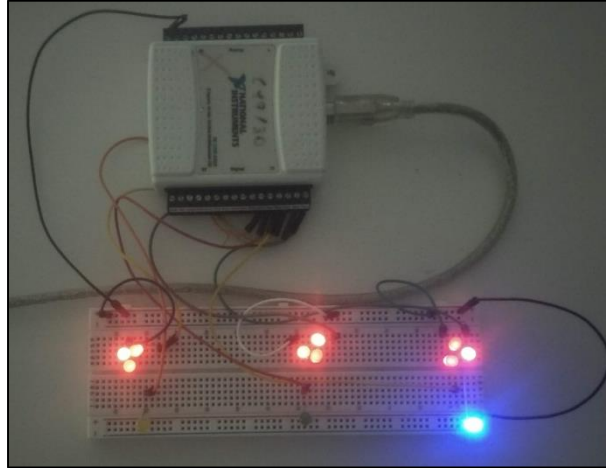


Figura 53. Sistema de control, el sujeto se fija en el RVS de 12 Hz

- 2) El sujeto se fija en el RVS de 15 Hz, el LED verde se enciende:

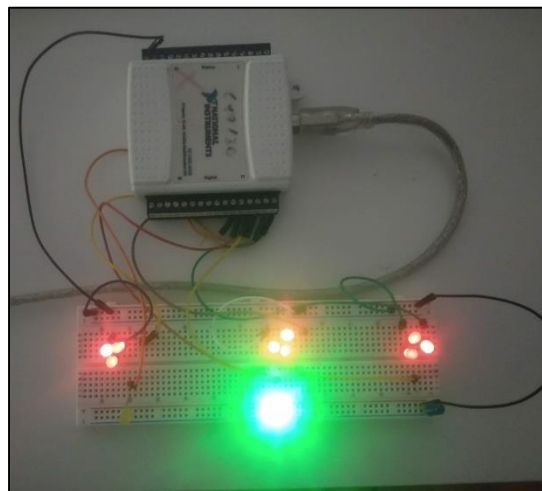


Figura 54. Sistema de control, el sujeto se fija en el RVS de 15 Hz

3) El sujeto mira al RVS de 20 Hz, el LED amarillo se activa:

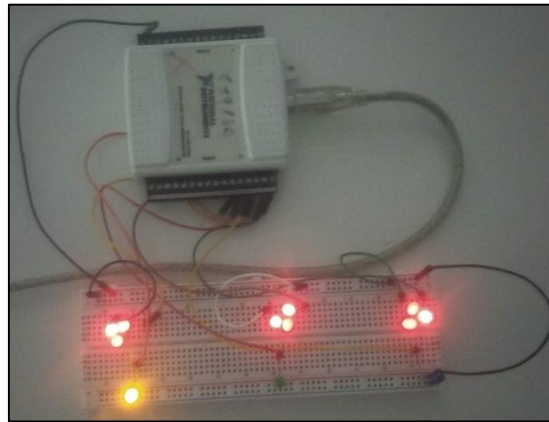


Figura 55. Sistema de control, el sujeto se fija en el RVS de 20 Hz

Otra opción que también puede darse es que el sistema no reconozca alguna frecuencia o que se equivoque y clasifique una frecuencia en otra. Esto dependerá del sujeto y de la fase de entrenamiento realizada tal y como se analiza en el siguiente apartado.

7. RESULTADOS

Durante este apartado se exponen y comentan los resultados obtenidos para siete personas diferentes tanto para la fase de entrenamiento como la de control.

Para la fase de entrenamiento se han realizado tres pruebas a cada sujeto, tras ser analizadas minuciosamente se ha incluido únicamente la que mejor resultados ha ofrecido. De esta manera, dicha prueba será comentada, para posteriormente realizar la fase de control con ese entrenamiento.

El apartado está dividido por las siete personas, es decir, nombraremos al sujeto y a continuación se expondrán y comentarán los resultados tanto en la fase de entrenamiento como la de control.

- **Sujeto nº 1** (edad, 20 años).

Tras realizar la fase de entrenamiento, introducimos el archivo ‘.ov’ generado en la aplicación “*Análisis entrenamiento.xml*” (fig. 40), ejecutamos y visualizamos el resultado en el bloque que simula un osciloscopio:

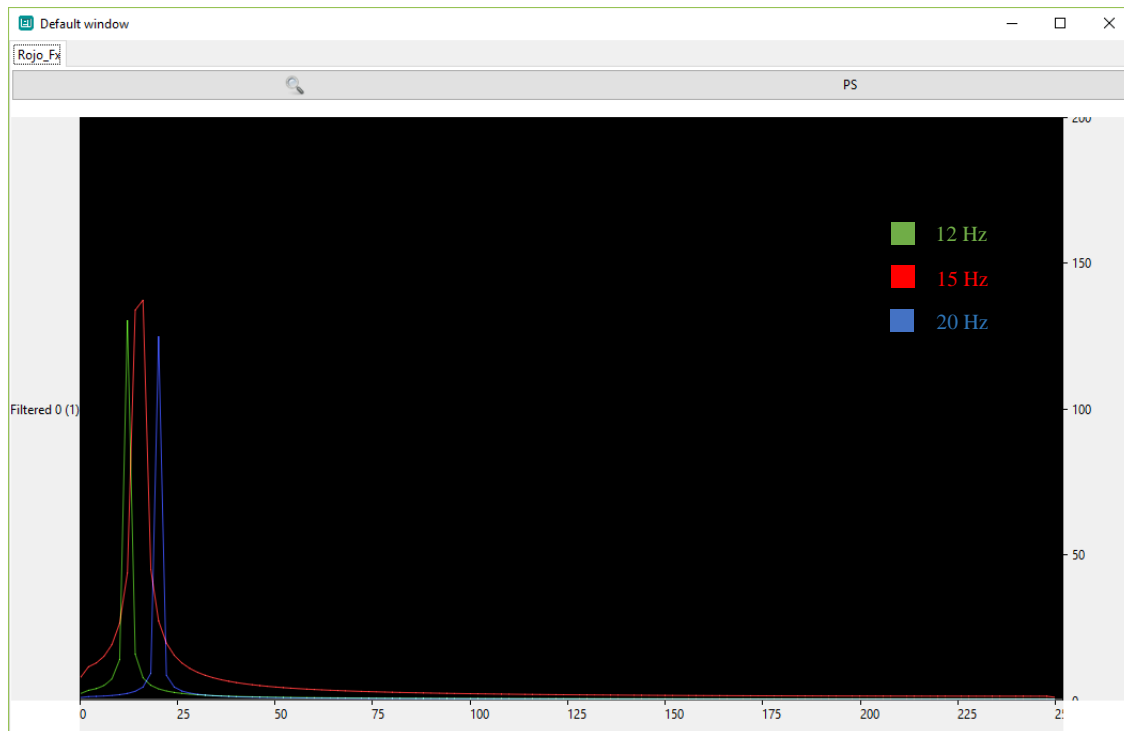


Figura 56. Análisis espectral de la fase de entrenamiento del sujeto nº 1

El resultado es bastante bueno, ya que hemos obtenido los tres picos de frecuencia en una amplitud casi equivalente. Esto nos indica que vamos a obtener un sistema de control robusto y, que debería reconocer cada frecuencia sin problemas.

Aunque bien se sabe por experiencia que de la teoría a la práctica pueden darse errores, por tanto, hemos realizado varias pruebas para comprobar el funcionamiento de la fase de control.

Estas pruebas consisten en indicar al usuario en la frecuencia en la que se debe fijar para comprobar si el sistema reconoce la frecuencia correcta, se han realizado para doce frecuencias. La siguiente tabla resume los resultados para el primer sujeto:

SUJETO N° 1		
Frecuencia de estimulación (Hz)	Frecuencia reconocida (Hz)	Evaluación
12	12	ACIERTO
15	15	ACIERTO
20	20	ACIERTO
12	12	ACIERTO
12	12	ACIERTO
15	15	ACIERTO
15	15	ACIERTO
20	15	FALLO
20	20	ACIERTO
15	15	ACIERTO
20	20	ACIERTO
12	12	ACIERTO

Tabla 1. Resultados de la etapa de control del sujeto n° 1

Se puede observar que se han obtenido unos resultados bastante buenos, con un solo fallo, tal y como se adelantaba tras realizar la fase de entrenamiento.

Por último, para concluir el análisis de los resultados, se va a calcular la precisión del sistema, mediante la siguiente fórmula:

$$\text{Precisión (\%)} = \frac{\text{Acertos}}{\text{Acertos} + \text{Fallos}} * 100 = \frac{11}{11 + 1} * 100 = \mathbf{91.67 \%}$$

- **Sujeto nº 2** (edad, 21 años).

La fase de entrenamiento de este sujeto ha derivado en el siguiente resultado:

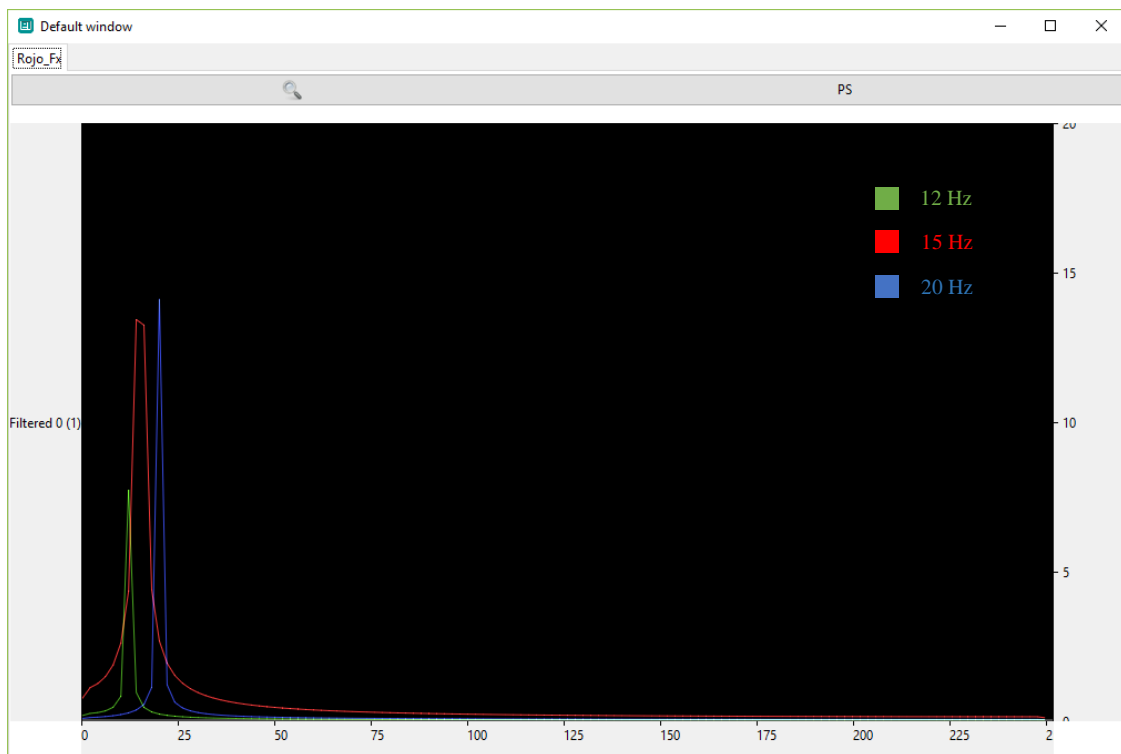


Figura 57. Análisis espectral de la fase de entrenamiento del sujeto nº 2

Este sujeto ha obtenido una respuesta mucho peor a la realizada para el sujeto número 1. En primer lugar, la amplitud de los picos es severamente menor a la anterior (fíjense en que está entre 5 y 15, cuando la anterior sobrepasaba el 100). Esto acarreará en una mayor dificultad de reconocimiento para el sistema de control.

Por otro lado, se han obtenido dos picos de frecuencia mayores en 15 y 25 Hz, en comparación con los 12 Hz. El sistema de control será propenso a fallar con las frecuencias de 12 Hz.

Los resultados del sistema de control han sido los siguientes:

SUJETO N° 2		
Frecuencia de estimulación (Hz)	Frecuencia reconocida (Hz)	Evaluación
12	15	FALLO
15	15	ACIERTO
20	20	ACIERTO
12	---	FALLO
12	12	ACIERTO
15	15	ACIERTO
15	15	ACIERTO
20	20	ACIERTO
20	20	ACIERTO
15	15	ACIERTO
20	20	ACIERTO
12	15	FALLO

Tabla 2. Resultados de la etapa de control del sujeto n° 2

Tal y como se ha previsto, este sistema tiene problemas a la hora de reconocer la frecuencia de 12 Hz.

$$\text{Precisión (\%)} = \frac{\text{Acertos}}{\text{Acertos} + \text{Fallos}} * 100 = \frac{9}{9 + 3} * 100 = 75 \%$$

- **Sujeto n° 3** (edad, 22 años).

El resultado de la fase de entrenamiento del sujeto n° 3 es el siguiente:

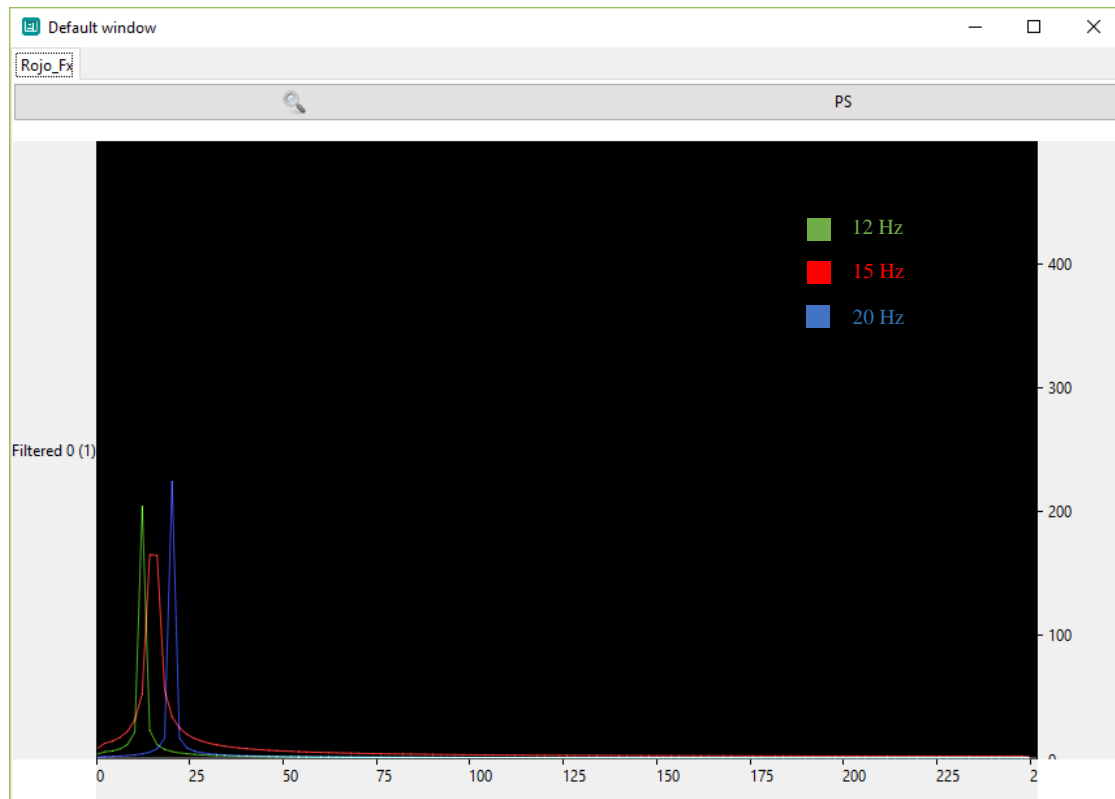


Figura 58. Análisis espectral de la fase de entrenamiento del sujeto n° 3

Se ha obtenido un mejor resultado al anterior, ya que la amplitud de los picos de frecuencia es mucho mayor, lo que facilitará el reconocimiento de cada frecuencia. El único problema que puede haber, es que el pico de 15 Hz es algo menor que el resto.

SUJETO N° 3		
Frecuencia de estimulación (Hz)	Frecuencia reconocida (Hz)	Evaluación
12	12	ACIERTO
15	15	ACIERTO
20	20	ACIERTO
12	12	ACIERTO
12	12	ACIERTO
15	20	FALLO
15	15	ACIERTO
20	20	ACIERTO
20	20	ACIERTO
15	15	ACIERTO
20	20	ACIERTO
12	12	ACIERTO

Tabla 3. Resultados de la etapa de control del sujeto n° 3

El sistema solo ha fallado una vez, por tanto, se confirma la hipótesis realizada tras observar los resultados del entrenamiento.

$$\textit{Precisión} (\%) = \frac{\textit{Aciertos}}{\textit{Aciertos} + \textit{Fallos}} * 100 = \frac{11}{11 + 1} * 100 = \mathbf{91.67 \%}$$

- **Sujeto n° 4** (edad, 55 años).

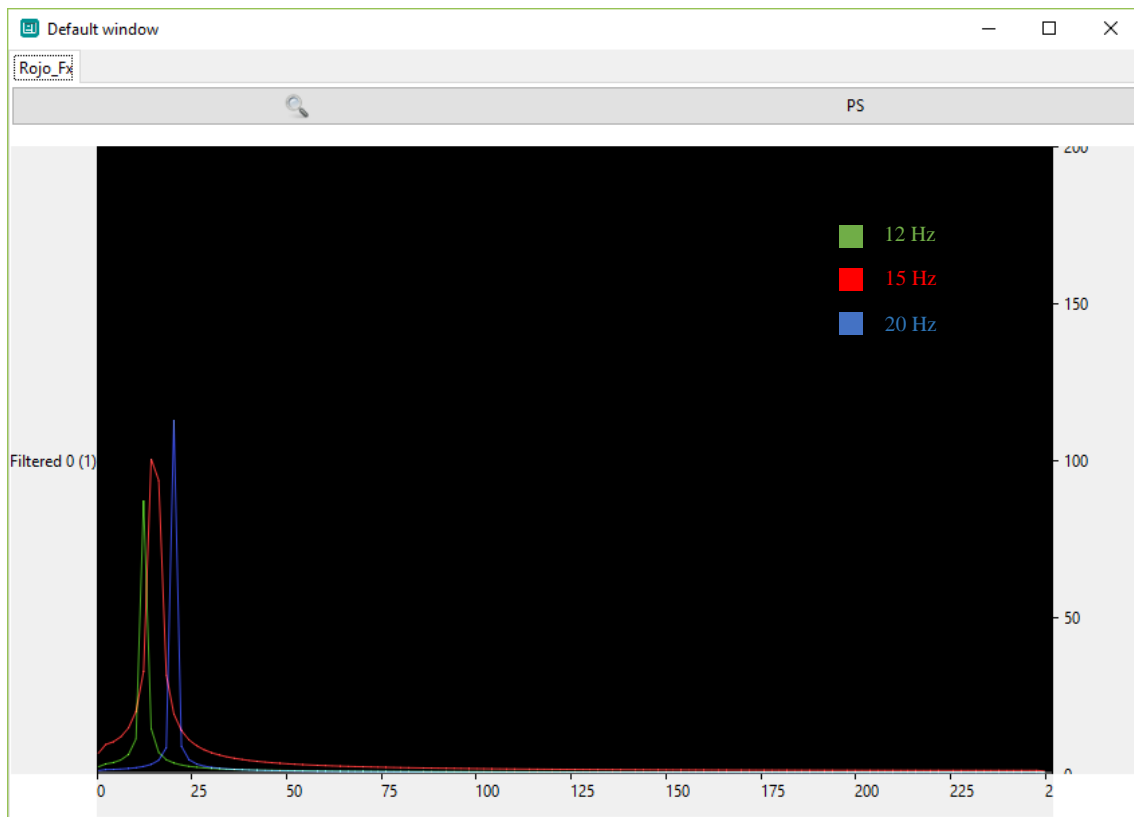


Figura 59. Análisis espectral de la fase de entrenamiento del sujeto n° 4

La respuesta no es tan buena como la del sujeto anterior, ya que la amplitud es algo menor, pero no existe mucha diferencia entre los tres picos de frecuencia.

SUJETO N° 4		
Frecuencia de estimulación (Hz)	Frecuencia reconocida (Hz)	Evaluación
12	12	ACIERTO
15	15	ACIERTO
20	20	ACIERTO
12	---	FALLO
12	12	ACIERTO
15	15	ACIERTO
15	15	ACIERTO
20	20	ACIERTO
20	20	ACIERTO
15	20	ACIERTO
20	20	FALLO
12	12	ACIERTO

Tabla 4. Resultados de la etapa de control del sujeto n° 4

$$\textit{Precisión} (\%) = \frac{\textit{Aciertos}}{\textit{Aciertos} + \textit{Fallos}} * 100 = \frac{10}{10 + 2} * 100 = \mathbf{83.33 \%}$$

- **Sujeto n° 5** (edad, 19 años).

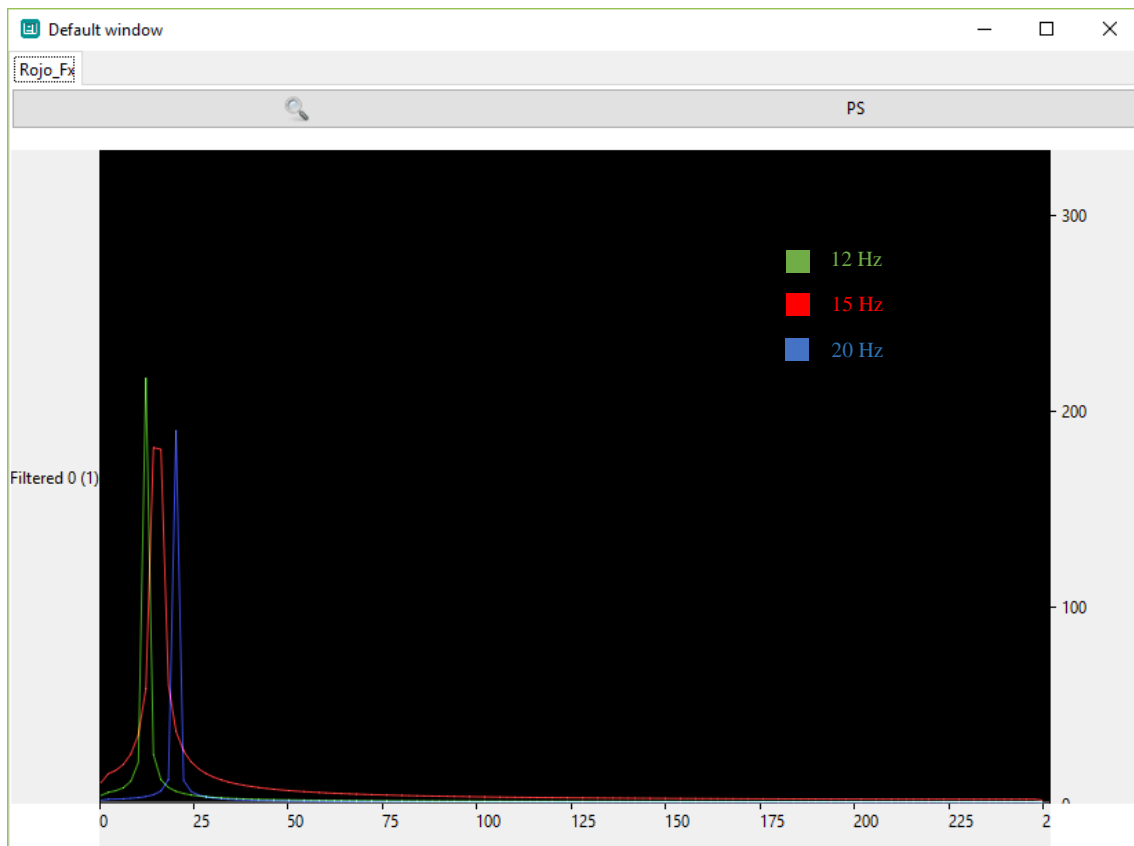


Figura 60. Análisis espectral de la fase de entrenamiento del sujeto n° 5

Este sujeto presenta una gran respuesta, debido a la amplitud de los picos y a la poca diferencia que hay entre ellos.

SUJETO N° 5		
Frecuencia de estimulación (Hz)	Frecuencia reconocida (Hz)	Evaluación
12	12	ACIERTO
15	15	ACIERTO
20	20	ACIERTO
12	12	ACIERTO
12	12	ACIERTO
15	15	ACIERTO
15	15	ACIERTO
20	20	ACIERTO
20	20	ACIERTO
15	15	ACIERTO
20	20	ACIERTO
12	12	ACIERTO

Tabla 5. Resultados de la etapa de control del sujeto n° 5

Para el sujeto n° 5 se han obtenido los mejores resultados, ya que todas las frecuencias reconocidas por el sistema coincidían con las de los RVS.

$$\textit{Precisión} (\%) = \frac{\textit{Aciertos}}{\textit{Aciertos} + \textit{Fallos}} * 100 = \frac{12}{12 + 0} * 100 = \mathbf{100 \%}$$

Y así se refleja con una precisión de 100%.

- **Sujeto n° 6** (edad, 21 años).

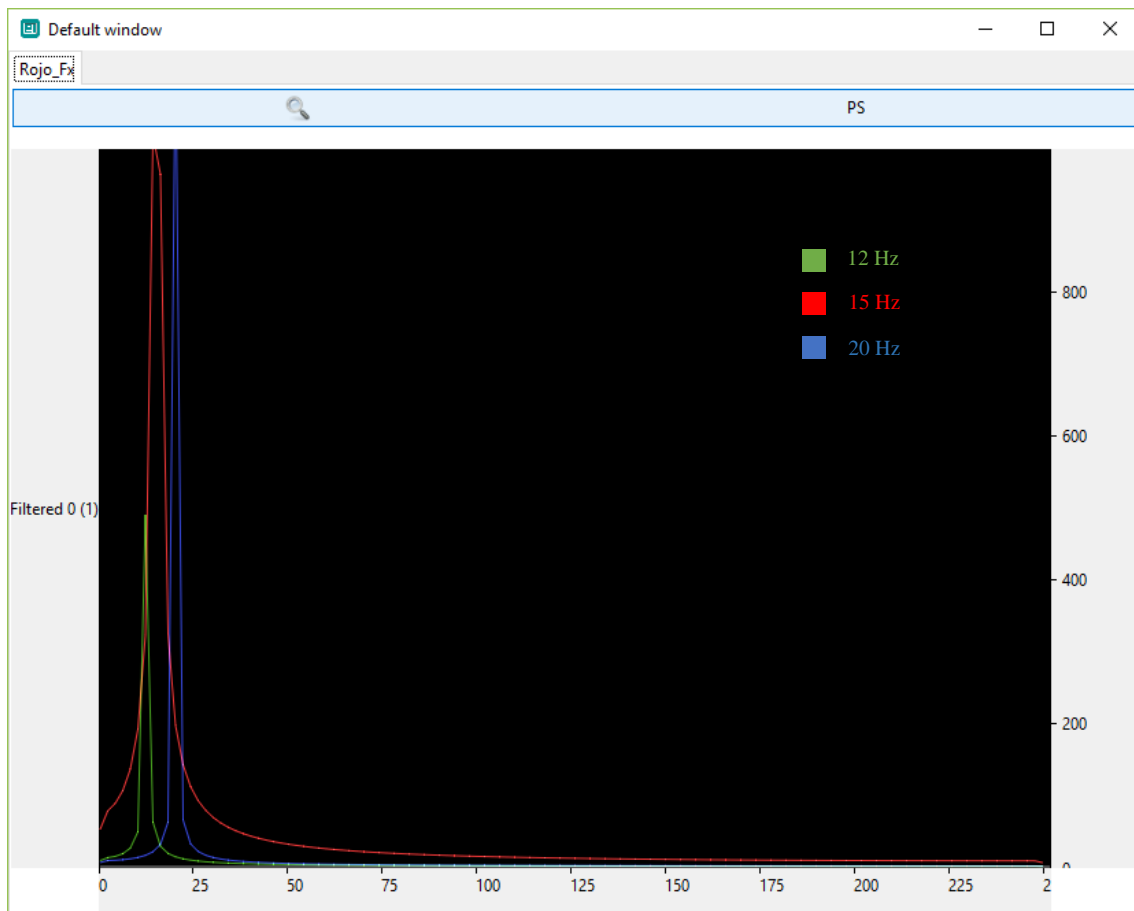


Figura 61. Análisis espectral de la fase de entrenamiento del sujeto n° 6

En cuanto al valor de la amplitud, con este sujeto hemos obtenido los mejores resultados, ya que los picos de 15 y 20 Hz superan el valor de 800. Pero el gran problema es que el pico de frecuencia de 12 Hz es muy inferior al de los otros, lo que generará problemas para reconocer esta frecuencia.

SUJETO N° 6		
Frecuencia de estimulación (Hz)	Frecuencia reconocida (Hz)	Evaluación
12	15	FALLO
15	15	ACIERTO
20	20	ACIERTO
12	15	FALLO
12	12	ACIERTO
15	15	ACIERTO
15	15	ACIERTO
20	20	ACIERTO
20	20	ACIERTO
15	15	ACIERTO
20	20	ACIERTO
12	15	FALLO

Tabla 6. Resultados de la etapa de control del sujeto n° 6

Como se ha adelantado, el sistema tiene varios problemas a la hora de reconocer la frecuencia de 12 Hz. Por este motivo la precisión disminuye:

$$\textit{Precisión} (\%) = \frac{\textit{Acertos}}{\textit{Acertos} + \textit{Fallos}} * 100 = \frac{9}{9 + 3} * 100 = 75 \%$$

- **Sujeto n° 7** (edad, 49 años).

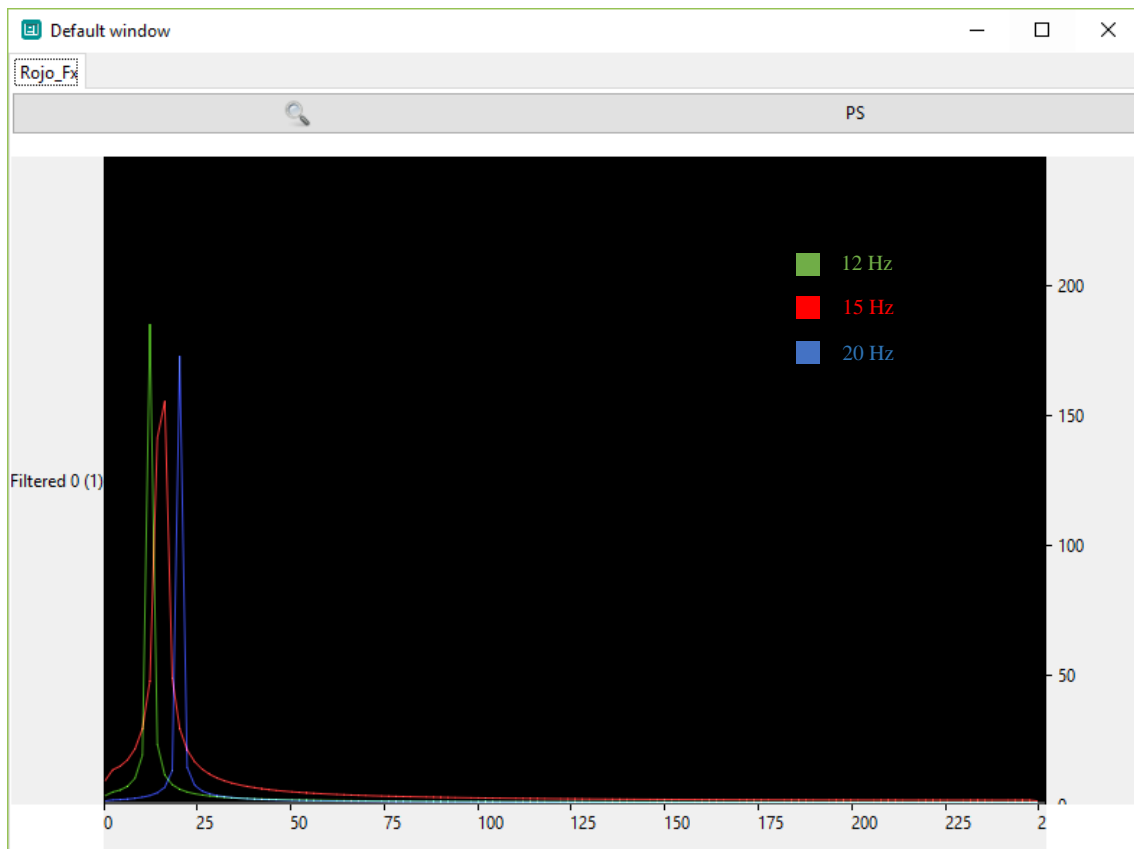


Figura 62. Análisis espectral de la fase de entrenamiento del sujeto n° 7

La amplitud de los picos no es tan grande como la del caso anterior, pero en este caso la variación de uno a otro no es muy significativa.

SUJETO N° 7		
Frecuencia de estimulación (Hz)	Frecuencia reconocida (Hz)	Evaluación
12	12	ACIERTO
15	15	ACIERTO
20	20	ACIERTO
12	12	ACIERTO
12	12	ACIERTO
15	---	FALLO
15	15	ACIERTO
20	20	ACIERTO
20	20	ACIERTO
15	15	ACIERTO
20	20	ACIERTO
12	12	ACIERTO

Tabla 7. Resultados de la etapa de control del sujeto n° 7

$$\text{Precisión (\%)} = \frac{\text{Aciertos}}{\text{Aciertos} + \text{Fallos}} * 100 = \frac{11}{11 + 1} * 100 = \mathbf{91.67 \%}$$

En conclusión, se puede apreciar que la respuesta que obtenemos depende en parte del sujeto en cuestión, ya que las pruebas han sido realizadas en unas condiciones similares para todos ellos.

Se aprecia cómo se obtienen resultados desiguales, a veces mejores y otras veces peores, pero generalmente el sistema tiene un buen rendimiento, ya que en todos los experimentos la tasa de aciertos es mucho mayor a la de fallos.

8. CONCLUSIONES

Como conclusiones extraídas de la realización de este proyecto se puede decir que las interfaces cerebro-computador, BCI, constituyen una tecnología muy interesante y que puede ser de gran ayuda para el ser humano. Si bien es cierto que aún les queda un largo camino de desarrollo para que sean introducidas en la vida cotidiana.

En concreto, los sistemas BCI basados en el paradigma SSVEP serían de gran ayuda para personas con problemas de discapacidad motora. En este proyecto se ha diseñado una aplicación aparentemente simple, pero con esta base se pueden realizar aplicaciones mucho más complejas que faciliten la vida a este tipo de personas en su día a día.

Por ejemplo, se podría diseñar una silla de ruedas controlada por un sistema BCI basado en SSVEP. Los RVS podrían ser cuatro flechas parpadeando a diferente frecuencia, una para adelante, dos hacia la izquierda/derecha y una última hacia atrás. Otra opción sería desarrollar una aplicación para escribir o hablar, si configuramos los RVS como las letras del abecedario, el usuario podría ir escribiendo las letras solo fijándose en ellas.

También se podría utilizar este tipo de sistemas en la domotización de una vivienda. Es decir, controlar aspectos como la iluminación, climatización, riego... únicamente con la mirada de un estímulo.

En definitiva, la realización de este proyecto hace ver la gran utilidad de estos sistemas BCI y la cantidad de aplicaciones que se pueden lograr mediante un diseño adecuado.

9. CONCLUSIONES ACADÉMICAS

Gracias al estudio y realización de este proyecto se han adquirido numerosos conocimientos académicos previamente desconocidos.

Primero, se ha aprendido y profundizado en el desarrollo de sistemas interfaz-computador, BCI, principalmente basadas en el paradigma SSVEP, además de observar su gran utilidad y aplicación.

Luego, se han adquirido conocimientos acerca del tratamiento de las señales EEG del cerebro humano, lo que suponía un campo desconocido antes de comenzar el proyecto.

Se ha avanzado de manera muy notoria e importante en la programación en el entorno LabVIEW, visto ligeramente en asignaturas como “Control de Sistemas Mecatrónicos” a lo largo del grado. Además el uso de una tarjeta de adquisición de datos ha hecho que se afiancen los conceptos previamente vistos en otras asignaturas como “Técnicas de Control”.

Se ha descubierto un nuevo software con una utilidad amplísima en el campo de las interfaces cerebro-computador: OpenViBE. Un programa totalmente desconocido antes de comenzar este proyecto, y en el que ahora se puede decir que se poseen unos conocimientos básicos para empezar a programar.

En conclusión, cabe mencionar que gracias a la ejecución de este proyecto se ha aprendido a trabajar con una planificación establecida, es decir, avanzando poco a poco cada día y descomponiendo tareas que puedan resultar complejas en varias más simples.

10. REFERENCIAS

- [1] Web Researchgate. Obtenido de https://www.researchgate.net/figure/Number-of-publications-on-BCI-over-the-past-18-years-obtained-via-Science-Direct-search_fig1_333448203
- [2] Wolpaw, J, Birbaumer, N, Mcfarland, D, Pfurtscheller, G, & Vaughan, T. Brain-computer interfaces for communication and control. *Clinical neurophysiology* (2002). , 113-767.
- [3] Weiskopf, N, Veit, R, Erb, M, Mathiak, K, Grodd, W, Goebel, R, & Birbaumer, N. Physiological self-regulation of regional brain activity using real-time functional magnetic resonance imaging (fMRI): methodology and exemplary data. *Neuroimage* (2003). , 19-577.
- [4] Waldert, S, Preissl, H, Demandt, E, Braun, C, Birbaumer, N, Aertsen, A, & Mehring, C. Hand movement direction decoded from MEG and EEG. *The Journal of neuroscience* (2008). , 28-1000.
- [5] Coyle, S, Ward, T, Markham, C, & Mcdarby, G. On the suitability of near-infrared (NIR) systems for next-generation brain-computer interfaces. *Physiological Measurement* (2004).
- [6] Web de Neuropsicología - Sincrolab. Obtenido de <https://blog.sincrolab.es/2017/03/20/que-es-y-que-podemos-medir-con-el-electroencefalograma-eeg/>
- [7] Pfurtscheller, G. and Lopes Da Silva FH. Event-related EEG/MEG synchronization and desynchronization: basic principles. *Clinical Neurophysiology* (1999). , 110(11), 1842-1857.
- [8] Allison, B, Faller, J, & Neuper, C. H. BCIs that use steady state visual evoked potentials or slow cortical potentials. *Brain-Computer Interfaces: Principles and Practice*. Wolpaw and E. W. Wolpaw, Eds. Oxford University Press; (2012).

[9] Vidal, J. J. Toward direct brain-computer communication. *Annual Review of Biophysics and Bioengineering* (1973). , 2-157.

[10] Vidal, J. J. Real-time detection of brain events in EEG. *Proceedings of the IEEE* (1977). , 65(5), 633-64.

[11] Regan, D. Steady-state evoked potentials. *J Opt Soc Am* (1977).

[12] Wikipedia (2019) Corteza Visual. Obtenido de https://es.wikipedia.org/wiki/Corteza_visual

[13] Silberstein, R. B. Steady state visually evoked potential (SSVEP) topography in a graded working memory task. *International journal of psychophysiology* (2001).

[14] C. S. Herrmann, "Human EEG responses to 1–100 Hz flicker: resonance phenomena in visual cortex and their potential correlation to cognitive phenomena," *Experimental Brain Research*, vol. 137, no. 3-4, pp. 346–353, 2001.

[15] D. Regan, *Human Brain Electrophysiology: Evoked Potentials and Evoked Magnetic Fields in Science and Medicine*, Elsevier, Amsterdam, The Netherlands, 1989.

[16] Danhua Zhu, "A Survey of Stimulation Methods Used in SSVEP-Based BCIs". *Review Article*, 2010.

[17] R. S. Fisher, G. Harding, G. Erba, G. L. Barkley, and A. Wilkins, "Photic- and pattern-induced seizures: a review for the epilepsy foundation of america working group," *Epilepsia*, vol. 46, no. 9, pp. 1426–1441, 2005.

[18] D. Regan, "An effect of stimulus colour on average steady-state potentials evoked in man," *Nature*, vol. 210, no. 5040, pp. 1056–1057, 1966.

[19] Y. Wang, R. Wang, X. Gao, and S. Gao, "Brain-computer interface based on the high-frequency steady-state visual evoked potential," in *Proceedings of the 1st International Conference on Neural Interface and Control*, pp. 37–39, May 2005.

[20] JMIndustrial, Technology S.A. de C.V. Obtenido de <https://www.jmi.com.mx/tarjetas-de-adquisicion-de-datos>

[21] Web de National Instruments. Obtenido de <http://sine.ni.com/psp/app/doc/p/id/psp-117/lang/es>

[22] Obtenido de <https://electronicaalvarosjm2.wordpress.com/ni-daq-6008/>

[23] Web de Neuroelectrics. Obtenido de <https://www.neuroelectrics.com/products/enobio/>

[24] Obtenido de <https://www.pardell.es/electroencefalografo.html>

[25] Web National Instruments. Obtenido de <http://www.ni.com/es-es/shop/labview.html>

[26] Wikipedia (2019) OpenViBE. Obtenido de <https://en.wikipedia.org/wiki/OpenVibe>

[27] OpenViBE web. Obtenido de <http://openvibe.inria.fr/>

[28] OpenViBE TCP Writer Tutorial. Obtenido de http://openvibe.inria.fr/documentation/2.2.0/Doc_BoxAlgorithm_TCPWriter.html

[29] Documento nº 2 – Planos. Plano nº 1: Fase de entrenamiento.vi – Estado 0.

[30] Documento nº 2 – Planos. Plano nº 6: Generador RVS.vi.

[31] Documento nº 2 – Planos. Plano nº 7: Aplicación Control.vi – Estado “Inicialización”.

[32] Documento nº 2 – Planos. Plano nº 8: Aplicación Control.vi – Estado “Reconocimiento frecuencia”.

[33] Documento nº 2 – Planos. Plano nº 9: Aplicación Control.vi – Estado “Frecuencia 1”

ANEJOS A LA MEMORIA

A-1. DATASHEETS

A continuación se incluyen los datasheets del hardware utilizado: Enobio 8; DAQ USB-6008; Diodos LEDS.

enobio^{NE®}

Neuroelectrics User Manual
Enobio



Europe Office

Avinguda Tibidabo, 47 bis
08035 Barcelona, Spain
Tel. +34 93 254 03 66
Fax. +34 93 212 64 45

US Office

210 Broadway, Suite 201
Cambridge, MA 02139, USA
Tel. +1 617 682 0770

Email: info@neuroelectrics.com

www.neuroelectrics.com

Copyright© by Neuroelectrics®. All rights reserved.



Manufacturer:

Neuroelectrics Barcelona SLU
Avinguda Tibidabo 47, bis
08035 Barcelona, Spain
Telephone: + 34 93 254 03 66

Code: UM003E
Version: 2.6
Date: 2019.04.04

Brand:

Enobio

Models:

Enobio 8

Enobio 20

Enobio 32

The manufacturer should be contacted:

- for assistance, if needed, in setting up, using or maintaining the Enobio system;
- to report an unexpected operation of events that result from the usage of the device.



Input: 5W, 4.25 - 7 V

About the Enobio User Manual

The Enobio User Manual belongs to Part I of the Neuroelectronics User Manual.

The Neuroelectronics User Manual includes three parts:

- ▶ Part I: Enobio User Manual
- ▶ Part II: Electrode User Manual
- ▶ Part III: NIC User Manual

Before your first use of the Enobio system, read the three parts of the Neuroelectronics User Manual. The Enobio User Manual does not discard the need of reading the Electrode and NIC parts.

The PDF version of all parts of the Neuroelectronics User Manual can be found under the Documentation section of Neuroelectronics webpage:

www.neuroelectronics.com/documentation/

Change of Record

Issue	Date	Changes made
1.0	2012.02.14	First version
2.0	2016.02.18	Neuroelectrics User Manual divided into three parts: (1) Enobio / Starstim, (2) Electrode and (3) NIC.
2.1	2017.03.10	SGS Update
2.2	2017.07.25	Product content update
2.3	2017.08.07	CE 0120 & symbols update
2.4	2018.07.16	Product content & safety symbols update
2.5	2019.03.20	EMC Compliance update
2.6	2019.04.04	NG Geltrodes introduced to the kit in the place of Geltrodes

Table of Contents

About the Enobio User Manual	4	IV. The Enobio System	17
Change of Record	5	IV.1 Technical Specifications	18
I. Use of Enobio	7	IV.2 Contents of the Enobio Package	19
I.1 Electroencephalography (EEG)	8	IV.3 Necbox: Neuroelectrics Control Box	25
I.2 Intended Use	8	IV.4 Assembling the Necbox and the electrode cables	27
I.3 Device Operation Environment	9	IV.5 Necbox battery	28
I.4 Applied Parts	9	IV.6 Disassembling the Device	29
I.5 Applications	10	IV.7 Cleaning Instructions of the Enobio Kit	29
I.6 Conditions of Use	11	V. Symbols Used	30
II. Quality and Regulatory Information	12	VI. Error Messages	31
II.1 Quality Management System	13	VII. Electromagnetic Compatibility (EMC) Information	32
II.2 Medical Device Regulations	13		
III. Safety Information	14		
III.1 Safety Warnings	15		

I. Use of Enobio

Enobio is a wireless and portable electrophysiology sensor system for the recording of the electrical activity of the human brain.

Enobio is a modern EEG device:

- ▶ Available with 8, 20 or 32 channels
- ▶ Wireless operating system
- ▶ Ideal for out-of-the-lab applications
- ▶ Offline data storage possible with microSD card
- ▶ Comfortable and precise head cap, with different head sizes available
- ▶ Flexible electrode placement with 39 possible positions based on the 10-10 system
- ▶ Capable of monitoring EOG, ECG and EMG
- ▶ Ease of use despite of the complexity of the technology



Enobio 8



Enobio 20



Enobio 32

I.1

Electroencephalography (EEG)

Electroencephalography is a non-invasive neuroimaging technique that records the electrical activity of the brain. Voltage fluctuations, that result from the ionic current within the brain neurons, are detected by electrodes placed on the scalp. The 10-20 international system defines 21 scalp positions to place the EEG electrodes. Additionally, the 10-10 system is obtained by adding electrodes in the intermediate positions. Both locations and nomenclature of the electrodes are standardized by the International Federation of Clinical Neurophysiology. The EEG measurements can be either bipolar or unipolar. The former method measures the potential difference between pairs of electrodes, while the latter measures the electrode potential compared to a reference.

The reference might be either the signal from one electrode or the average of the measurement from two or more electrodes.

The EEG measurements are frequently analyzed with spectral methods to define frequency bands, also known as brain waves. The five most common bands are delta (0 - 4 Hz), theta (4 - 8 Hz), alpha (8 - 13 Hz), beta (13 - 30 Hz), and gamma (30 - 50 Hz). These waveforms provide crucial information regarding brain function which is widely used to diagnose Epilepsy, sleep disorders, Coma or cerebral death. The EEG differs from other neuroimaging techniques due to its high temporal resolution. However, it is often combined with magnetic resonance imaging (MRI) or computed tomography (CT) for the diagnosis of tumors, stroke and other brain disorders whose diagnosis also requires high-spatial resolution.

I.2

Intended Use

ENOBIO WF is an EEG portable monitoring device of 8, 20, 32 channels intended for use in a clinical environment, hospital, research center or home healthcare environment. The applications considered are medical and non-medical such as: Brain to computer interface (BCI), Basic EEG research, Medical applications developments, neuromodulation, biometry and user affective state. ENOBIO use must be controlled by specialized medical personnel able to guarantee the correct recording.

It is intended to acquire, store, transmit and display electrophysiological signals in wireless mode as an aid in diagnostics. The system digitizes analogue EEG signals collected by a cap with electrodes, amplifies them, and uses Wi-Fi wireless connectivity to transmit the EEG data to a dedicated host computer with the software.

I.2 Intended Use

This device is intended to be used by trained healthcare personnel.

I.3 Device Operation Environment

During montage preparation phase

The patient must be situated in a comfortable position which facilitates the electrode and device placement of the required montage positions.

During Quality Check and Recording Phase

Although the specific position of the patient, an operator or other people do not influence the quality of the recording, both patient and operator must minimize sudden movements or touching elements of the headset to maximize signal quality.

I.4 Applied Parts

Electrodes are the components applied directly on the patient's scalp.

For detailed instructions regarding the use of the all electrodes applicable with Enobio, consult the Electrode User Manual. It can be also found at www.neuroelectrics.com.

I.5 Applications

Brain Computer Interface (BCI)

In order to build real world BCI applications, it is essential that the subject be comfortable and behaving naturally in his typical environment. Enobio was designed with these constraints in mind, thus it may be used as a platform for the development of BCI applications away from the lab. Possible BCI paradigms include Motor Imagery ERD, SSVEP and P300 among others. Currently, Enobio is being used by many researchers and it is compatible with BCI software systems such as BCI2000, OpenVibe, PureData and Processing.

Medical application development

Enobio is the ideal platform for when large amounts of data need to be collected from several subjects in natural environments in order to develop medical applications.

It may be used as often as necessary to quickly and reliably acquire medical data. Data integrity, versatility and ease of use are the keys to the development of medical applications. NIC software and the Nube Cloud Service provide the means to achieve them efficiently. Clinical applications include sleep monitoring and Epilepsy.

Basic electrophysiological research

Enobio offers an advantage in any scenario in which electrophysiological data is difficult to obtain in a natural way. Spontaneous EEG or ERP's on the go are possible with Enobio due to the absence of long wires.

Neuromodulation

EEG neuromodulation is the process of modifying brain state via feedback-based training. By visualizing the brain activity, the user can learn to modify it. Recent work provides evidence for the use of neuromodulatory therapy in

the treatment of a variety of disorders. This can be achieved by combining Enobio with the Neurosurfer - the Neuroelectrics neurofeedback software tool. Neurosurfer integrates EEG data, brain stimulation, and an immersive virtual reality game, all into one platform.

User affective state

User emotional state can be derived from physiological signals like EEG and ECG. Enobio provides a platform for research and development in affective computing and smart systems.

I.6

Conditions of Use

Enobio must be used with normal temperature, humidity and pressure conditions:

- ▶ Temperature Range: +5 to 40 °C
- ▶ Humidity: 15 - 93 %
- ▶ Atmospheric Pressure:
700 - 1.000 hPa

The device must be stored inside the box between uses, in the following environmental conditions:

- ▶ Temperature Range: -25 to +65 °C
- ▶ Humidity: 15 - 93 %

This equipment needs to be installed and put into service in accordance to the information provided in this user manual.

II. Quality and Regulatory Information

II.1

Quality Management System

Neuroelectrics is an ISO 13485 and ISO 9001 certified company. Thus, our medical devices are designed and manufactured following the corresponding ISO quality management systems.

Neuroelectrics complies with Quality System Regulation 21 CFR 820.

II.2

Medical Device Regulations

Neuroelectrics Enobio is a class IIa device according to the classification in the Council Directive 93/42/EEC for medical devices.

Enobio conforms to the following Standards and Directives:

USA

Clearance by the FDA, regulation 510(k) (K162681)

Europe

Directive 93/42/EEC for medical devices

Canada

Canadian Medical Device Regulations SOR/98-282

III. Safety Information



















Enobio conforms to the following standards:

- ▶ EN 60601-1:2006/A1:2013
- ▶ UNE-EN 60601-1-2:2008
- ▶ UNE-EN 60601-2-26:2004
- ▶ IEC 60601-1-11:2010
- ▶ EN 980:2008
- ▶ EN 1041:2008
- ▶ EN ISO 14971:2012
- ▶ IEC 60601-1-6:2010

The device cannot be turned on while it is connected to the main power supply through the power adaptor. The isolation of the main power supply is achieved by disconnecting the power adaptor.

III.1

Safety Warnings

-  The result of the recordings must be analyzed by a doctor or a specialist. No self medication should be done based on the results.
-  The device must never be opened or damaged.
-  The battery can only be replaced by authorized personnel.
-  Before using, please check that the device is not damaged and the packaging has not been affected by transport or storage.
-  Before using the device, please inform the prescribing clinician or operator of the presence of any pacemakers, intracranial electrodes, implanted defibrillators, cranial pathologies (e.g. holes, plaques), implantable neurostimulators, deep brain stimulators, or any other prosthesis'. In these cases, the use of the device could become unsafe.
-  In the case of malfunction, immediately contact the manufacturer or the distributor.
-  The device is not protected against excessive moisture or immersion in liquid. In the case of the device becoming wet or damp, do not use it and immediately contact the manufacturer.
-  Do not touch the device while EEG monitoring is on.
-  Always unplug the USB power supply from the device prior to connecting electrodes to the subject. The device will not work when the battery is charging.
-  Never use the device or install the electrodes on the head of the patient while connected to the power network.
-  Do not switch the device on or off when it is assembled and placed on the subject's scalp.
-  The device must be used only with Ag/AgCl electrodes recommended by the manufacturer.
-  The device is not protected against other high frequency devices. To avoid risks place the CMS/DRL as far as possible from the electrodes of the high frequency device.
-  The device is wireless and can be affected by other RF devices.
-  The device needs special EMC precautions. It needs to be used according to the EMC information at the end of the user manual.
-  The EMC emissions and immunity have been tested using the 10-wire and 12-wire 34 cm cables provided with the system.
-  The device can only be used on healthy skin without wounds.
-  The device shall be connected to the charger in a way that makes it easy to disconnect.

-
- ⚠ The use of cables or electrodes other than the ones delivered with the product might produce higher EMC emissions and less EMC immunity.
 - ⚠ The device cannot be used in the MRI room.
 - ⚠ The device is not provided sterile and should not be sterilized.
 - ⚠ The device does not need installation, maintenance or calibration.
 - ⚠ The device and the accessories should be regularly checked by the user.
 - ⚠ If the user wants to use the device in combination with another device connected to the patient, the user should contact Neuroelectrics to check the correct simultaneous use.
 - ⚠ The modification of the device is not allowed.
 - ⚠ If the device has not been used

during a long period of time, the user should check visually that there is no battery leakage.

- ⚠ The device cannot be used beside or put under other equipment. If such usage is needed, check the normal configuration.
- ⚠ The electrodes and wires or any conductive part cannot touch any other conductive part of any other device including the ground.
- ⚠ The cap is intended to be on the patient for less than 24 hours.
- ⚠ Enobio should not be used in an MRI room or close to CT, diathermy, RFID and electromagnetic security systems such as metal detectors. Some of these RF emitters (e.g. RFID) may not be visible and the device can potentially be exposed to fields from these RF emitters without the user's awareness. If any electromagnetic interference is encountered,

the user will be informed by the software with the "Packet Loss" alarm, and will indicate the user to stop the recording and move to a place without interference. In any case, the electromagnetic interference does not involve any risks to the patient, as Enobio is a non-invasive recording device that does not modify or interact with the person's brain.

- ⚠ Keep out of reach from children and anyone else who might swallow electrodes, or cause injury to themselves.
- ⚠ Keep out of reach from children and anyone else who might strangle themselves with the cables of the devices.
- ⚠ The result of the recordings is not displayed in legal units or other units within the meaning of Directive 80/181/ECC. Therefore the device is not considered to have a measuring function.

IV. The Enobio System

This chapter describes the Enobio system. First, it lists the technical specifications of Enobio. Then, the components included in the Enobio package are listed and described. For each item, the product code, the product name, a picture and a short description of its function are listed. Lastly, the Neuroelectrics Control Box (Necbox) which is the core and the control unit of Enobio is described.

For further information regarding the use of the electrodes, please consult the Electrode User Manual. Additionally, to learn how to pair your device with the computer, read the NIC User Manual. The NIC User Manual explains the steps needed to correctly perform an EEG monitoring session.

IV.1

Technical Specifications

EEG functionality

- ▶ Number of channels: 8, 20 or 32
- ▶ Sampling rate: 500 SPS
- ▶ Bandwidth: 0 to 125 Hz (DC coupled)
- ▶ Resolution: 24 bits - 0.05 μ V
- ▶ Measurement noise: < 1 μ V RMS

Other Technical Specifications

- ▶ Accelerometer: 3-axis
- ▶ Communication:
Wi-Fi iIEEE 802.11 g or USB
(Only available for use with USB Isolator accessory cable)
- ▶ Output: EDF+ (16 bits), ASCII data files or TCP/IP raw data streaming
- ▶ OS compatibility: Windows (Vista / 7 / 8 /10) and MAC OS X Snow Leopard

- ▶ Battery operating time:

		All Channels EEG
Wi-Fi	Enobio 32	5 h 15 min
	Enobio 20	5 h 20 min
	Enobio 8	6 h 23 min
USB	Enobio 32	19 h 0 min
	Enobio 20	19 h 0 min
	Enobio 8	23 h 35 min

*SD card recording will increase these values slightly

Minimum Computer Requirements

- ▶ Operating System: Windows Vista or MAC OS X Snow Leopard
- ▶ Processor: 1.6 GHz
- ▶ RAM: 2 GB
- ▶ Wi-Fi, or USB 2.0

Wireless Information

Enobio is a wireless device. The Necbox connects through Wi-Fi to the Neuroelectrics Instrument Controller (NIC) software running on a computer. The EEG data is streamed through the Wi-Fi band, which has an operating distance range of 10 meters or less.

Wireless Specifications

- ▶ Wi-Fi iIEEE 802.11 g
- ▶ Operating frequency band: 2412 ~ 2472 MHz
- ▶ Transmitting power
Max. 16 ~+17.6 dBm
- ▶ Qualifications: CE, FCC, IC, Japan and South Korea
- ▶ Data rate: 921 kbps

IV.2





Contents of the Enobio Package





The Enobio package contains all the components required to perform an EEG monitoring session, and some additional items that may be useful during your experiments. In the following pages, you may find a description for each item. Please note that the necbox code, the number of electrodes, and the electrode




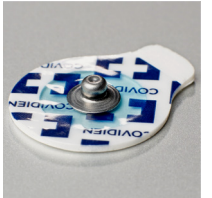
cables vary among the three models: Enobio 8 (EN8), Enobio 20 (EN20), and Enobio 32 (EN32).




Quantity	Code	Name
1	NE010WF NE011WF NE009WF	Enobio 8/20/32 Necbox
1	NE055/NE055W	USB Power Adapter
1	NE013a NE013b NE013c	EU / US / UK Power Supply Plug
1	NE014	Curved Syringe
1	NE015	USB Stick with Manuals & NIC
1	NE016b	Electrode Gel 250g
1	NE017	10 Electrode Cable
0 / 1 / 1	NE018a	10 Electrode Cable EN 20/EN32
0 / 1 / 1	NE018b	12 Electrode Cable EN20/EN32
0 / 0 / 1	NE018c	12 Electrode Cable EN32

Quantity	Code	Name
1	NE019-M	Neoprene Headcap M (54cm)
8 / 20 / 32	NE032	Electrode: NG Geltrode
50 / 50 / 50	NE025	Electrode: Stickrode
1	NE027	Electrode: Earclip
1	NE172	USB Extension Cable
1	NE031b	USB Wi-Fi Dongle

Item	Name / Description	Code
	<p>Enobio Necbox (8ch/20ch/32ch)</p> <ul style="list-style-type: none"> ▶ The Enobio Neuroelectrics Control Box (Necbox) is the core of the Enobio system. ▶ The Necbox is battery operated and it is wirelessly paired with the computer using the NIC software. ▶ The Necbox battery should never be charging during an EEG recording. 	<p>NE010WF NE011WF NE009WF</p>
	<p>USB Power Adapter & Power Supply Plug</p> <ul style="list-style-type: none"> ▶ The USB power adapter is used to charge the Necbox battery. ▶ The type of the power supply plug (EU/US/UK) included in the kit depends on the country of the customer. ▶ Minimum isolation 4000V, 2x MOPP ▶ Medical device EMC (IEC 60601-1, IEC 60601-2) 	<p>NE055/ NE055W</p>
	<p>Curved Syringe</p> <ul style="list-style-type: none"> ▶ The curved syringe is used to inject electrode gel in the electrodes. ▶ Wash and clean it after each use. 	<p>NE014</p>
	<p>USB Stick with Manuals & NIC SW</p> <ul style="list-style-type: none"> ▶ The USB stick contains the PDF version of the three parts of the Neuroelectrics User Manual, and the NIC software. ▶ Both items can be also found at www.neuroelectrics.com. 	<p>NE015</p>

Item	Name / Description	Code
	<p>Electrode Gel 250 g</p> <ul style="list-style-type: none"> ▶ The electrode gel is a highly conductive and water soluble gel. It must be applied on the contact surface, between the electrode and the scalp, in order to decrease the impedance and improve the signal quality. 	NE016b
	<p>10 Electrode Cable</p> <ul style="list-style-type: none"> ▶ The 10 electrode cable contains 8 channels, numbered from 1 to 8, for EEG monitoring, and two reference channels labeled with CMS & DRL. It is included with Enobio 8, Enobio 20 and Enobio 32. It allows the channels to be freely assigned to any position of the head cap. 	NE017
	<p>10 Electrode Cable EN20/EN32</p> <ul style="list-style-type: none"> ▶ This cable is included with Enobio 20 and Enobio 32, and is used together with NE018b. ▶ Its 10 cable connections are labeled with the positions: ▶ P7, P4, Cz, Pz, P3, P8, O1, O2, CMS & DRL 	NE018a
	<p>12 Electrode Cable EN20/EN32</p> <ul style="list-style-type: none"> ▶ This cable is included with Enobio 20 and Enobio 32, and is used together with NE018a. ▶ Its 12 cable connections are labeled with the positions: ▶ F7, T7, Fp1, F3, C3, Fz, Fp2, F4, C4, F8, T8, and EXT (EN20) or Oz (EN32). 	NE018b

Item	Name / Description	Code
	<p>12 Electrode Cable EN32</p> <ul style="list-style-type: none"> ▶ This cable is included only with Enobio 32, and it is used together with NE018a and NEO18b. ▶ Its 12 cable connections are labeled with the positions: ▶ PO3, AF3, FC5, FC1, CP5, CP1, CP2, CP6, AF4, FC2, FC6, PO4 	NE018c
	<p>Neoprene Head cap M (54cm)</p> <ul style="list-style-type: none"> ▶ The Neoprene Cap is a comfortable solution to precisely place the electrodes on the scalp based on the 10-10 system. It provides 39 possible electrode positions, but extra positions can be added using the neoprene punch tool (not included). The cap provided is medium sized, but other sizes are also available. 	NE019-M
	<p>NG Geltrode</p> <ul style="list-style-type: none"> ▶ Based on the next generation (NG) assembling system, the NG Geltrode is a EEG electrode that allows the access to the scalp through the headcap. It can be used wet with gel, or dry with solidgel technology. <p>Read the Electrode User Manual before using the NG Geltrode.</p>	NE032
	<p>Sticktrode</p> <ul style="list-style-type: none"> ▶ The sticktrode is a pre-gelled adhesive electrode used for the CMS & DRL reference channels. It can be also used to monitor ECG or EOG. It does not require the application of electrode gel. <p>Read the Electrode User Manual before using the sticktrode.</p>	NE025

Item	Name / Description	Code
	<p>Earclip</p> <ul style="list-style-type: none"> ▶ The Earclip is an easy-to-use alternative to the sticktrode. It is a dual reference electrode because it is used to connect the two reference channels, CMS and DRL, to the same earlobe. The application of electrode gel is highly recommended. <p>Read the Electrode User Manual before using the earclip.</p>	NE027
	<p>USB Dongle or USB Wi-Fi Dongle</p> <ul style="list-style-type: none"> ▶ The USB Dongle is used to provide a Wi-Fi port for computers that do not have an incorporated port. The wireless communication between the Necbox and the computer is through Wi-Fi. The USB Dongle must not be used with Mac OS computers. 	NE031/ NE031b
	<p>USB Cable & Isolator</p> <ul style="list-style-type: none"> ▶ The USB Cable & Isolator can be used to transmit EEG data between the device and the computer. This should always be used with the extender cable. Note that this cable does not charge the device. 	NE164/ NE172

In order to make your Enobio experience more complete, you can add accessories to your kit.

In our catalog and on our website, you can find:

- ▶ Different sizes of the neoprene head cap:
XL, L, M, S, Kids (K) and Kids Small (KS)
- ▶ The neoprene punch tool:
Customize your own neoprene cap!
- ▶ Mouse Head cap Cover:
Provide kids a pleasant EEG experience
- ▶ Dry electrodes for EEG monitoring:
For fast and gel-free EEG experiments
- ▶ Solid gel technology for EEG monitoring:
For clean EEG experiments

These items are available upon request.
Please contact our sales team to learn more.

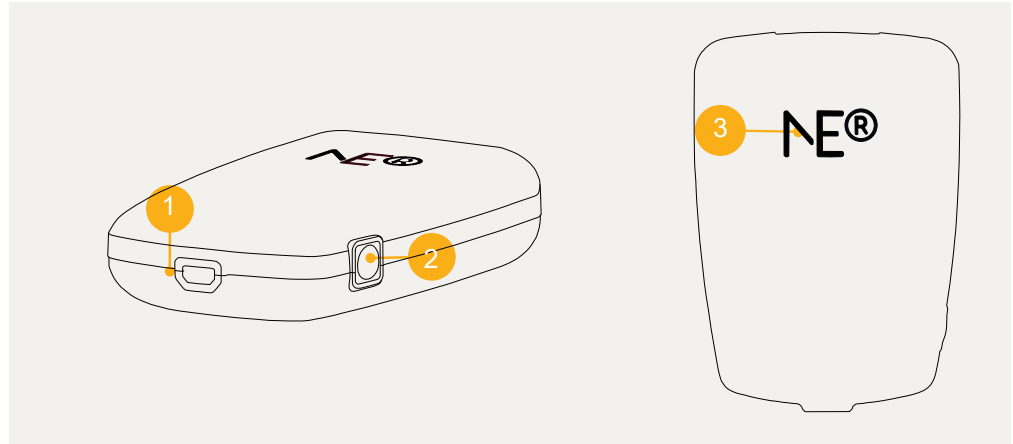


Mouse Head cap Cover

IV.3

Necbox: Neuroelectrics Control Box

The Necbox is the core and the control unit of Enobio. The Necbox is a battery operated device. It weighs 85 g and its dimensions are 89 mm x 61.1 mm x 23.8 mm. The following diagrams describe the details of the Necbox.



1. Charging LED

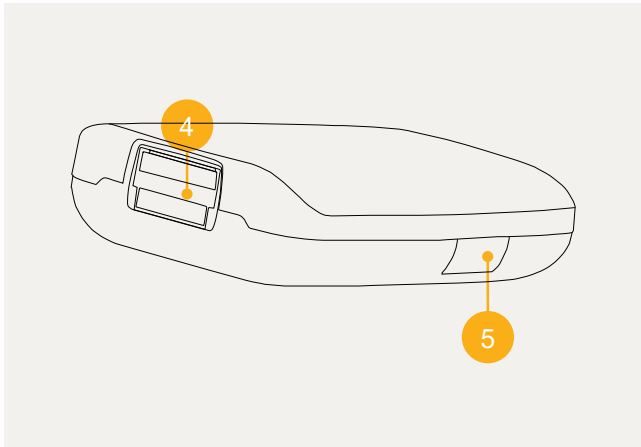
- Off: The charger is not connected.
- Yellow light: The charger is connected and the device is charging.
- Green light: The charger is connected and the device is charged.

2 ON/OFF Push-Button

- On single push, switches on the device while off.
- On 2s hold, switches off the device while on.

3 Operation LED

- Off: The device is off.
- Continuous light: The device is functioning correctly in standard operational mode.
- Blinking with 1s period: The device is functioning correctly in "holter" mode.
- Blinking with 250ms period: The device lost connection during protocol execution and became nonoperational. To continue, it needs to be switched off and on again.
- Blinking with 200ms period 16 times: The device cannot start in "holter" mode because of a problem with the SD card.

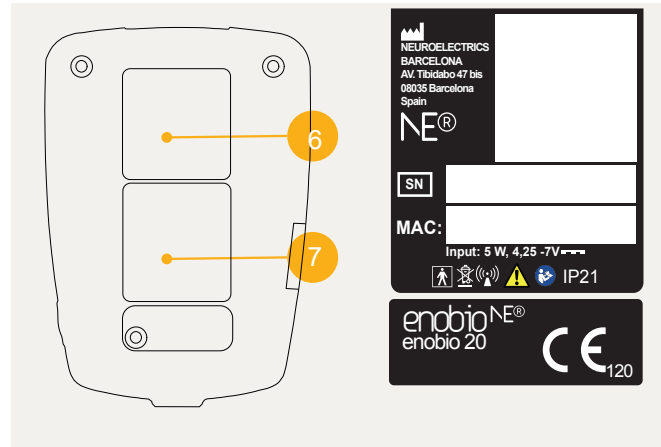


4 Pin connector slots

The number of slots depends on the number of the channels of the Enobio (see next page).

5 MicroSD card slot

Slot for microSD card (Card not included) for online data storage in the “holter” mode.



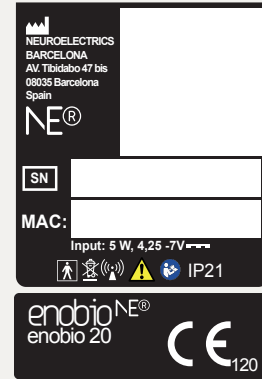
6 Velcro

To attach the Necbox to the neoprene cap.

7 Technical Specifications labels

Serial Number (SN), with the EYYYYMMDD format, where YYYY, MM and DD are the manufacturing year, month and day, respectively.

MAC address of the device.



IV.4

Assembling the Necbox and the electrode cables

First, the Necbox is attached to the neoprene cap using the velcro. Secondly, the Necbox should be connected to the electrode cable(s). The cables are inserted in the pin connector slots of the Necbox as described in the diagram below:

▶ **Enobio 8**

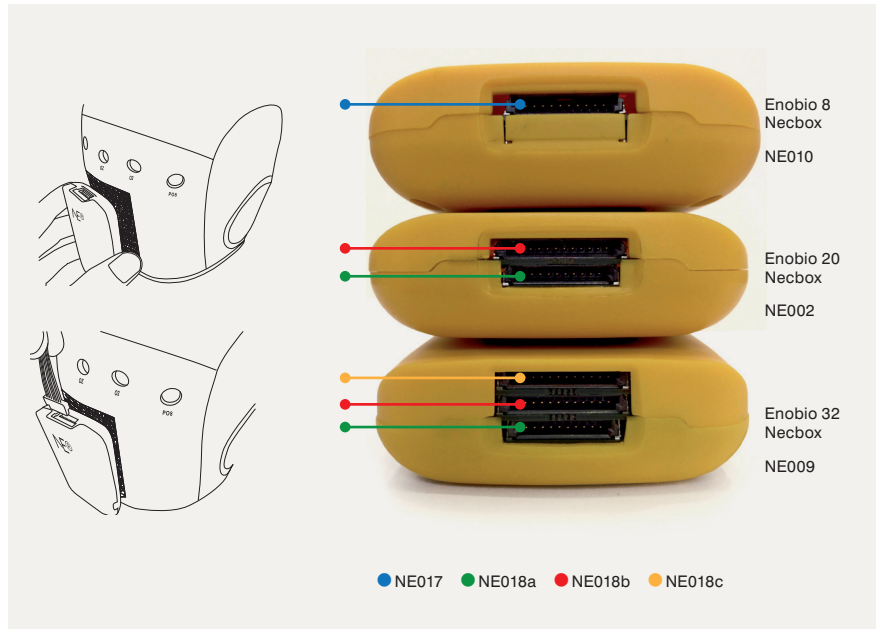
Only electrode cable ● NE017, should be connected to the Necbox using the only available slot (blue).

▶ **Enobio 20**

The pair of cables ● NE018a and ● NE018b that must be connected to the green and red slots, respectively.

▶ **Enobio 32**

Similarly to Enobio 20, but the cable ● NE018c is added and connected to the orange slot.



Enobio 20 and Enobio 32 may be alternatively used as an Enobio 8. In this case, the cable ● NE017 is the only cable that should be used, and it must be connected to the green slot while the remaining slots must be left empty.

IV.5

Necbox battery

The battery can only be charged when the device is at the OFF position.

The battery charger connects to the Necbox. The battery charger connects to the Necbox through the Micro-HDMI connector located at the bottom part of the Necbox. To charge the battery, the following specifications need to be met:

- ▶ Nominal output: 5V (4.25V - 7V)
- ▶ Current output: 1A
- ▶ Battery charger: must comply according to Standard IEC-EN 60601-1
- ▶ The battery state of charge is measured by NIC when the device is powered ON and paired with the computer.
- ▶ The battery should not be over discharged when the device is not used for a long time. It should be periodically charged instead.

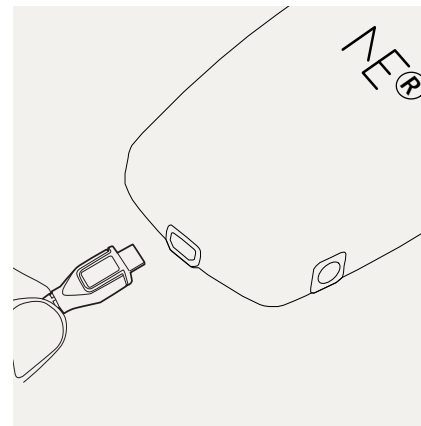
- ▶ Overdischarging may cause loss of cell performance and/or damage to battery function.
- ▶ Expected life cycle: > 500 cycles > 70% of initial capacity
- ▶ Charging with higher voltage than specified may damage the cell.
- ▶ The usual time to charge a battery from the cut-off voltage to the maximum capacity is around 2 hours, but it depends on each (battery life and memory is a function of time).
- ▶ The device can be connected to any Class 2 electrical installation.
- ▶ Device will not operate when charging.
- ▶ Only use the charger that came with the device to charge the battery.

Operating Temperature

- ▶ Charging: 0°C to 40°C
- ▶ Discharging: 0°C to 40°C

Electrical Specifications for charging:

- ▶ Voltage nominal input: 5 V DC
- ▶ Voltage input min/max: 4.25 V - 7 V
- ▶ Power input: 5 W



IV.6

Disassembly of the Device

First, disconnect all the electrode cables from the electrodes.

Then, take off the head cap and disconnect the electrode cables from the Necbox device.

Once the cables are disconnected, switch off the Necbox device by holding the ON/OFF push button for 2 seconds.

IV.7

Cleaning Instructions of the Enobio Kit

Necbox & Electrode Cable

The Enobio Necbox should be cleaned using a dry paper towel after each use.

Neoprene Head cap











The Neoprene Head cap should be cleaned and disinfected as it follows:







- ▶ Rinse off the gel with warm tap water and ivory soap
- ▶ Dry the cap carefully using a paper towel
- ▶ Spray the cap with disinfectant and let it sit for 10 minutes, or use disinfectant wet wipes
- ▶ Rinse the cap thoroughly
- ▶ Hang up the cap to dry

Electrodes

The cleaning instructions for the electrodes can be found in the Electrodes User Manual.

V. Symbols Used

Symbol	Description
	Refer to the manual/booklet
	General warning sign
	IEC 60417-5010 Push ON/OFF button UNE-EN 60601-1=2008
	ISO 7000-2498 Serial Number according to UNE-EN 980
	Device manufacturer symbol according to UNE-EN 980
	ISO 7000-2606 do not use device if product or packaging have been damaged symbol according to UNE-EN 980
	Do not throw Enobio in generic waste symbol. WARNING! When you want throw away the device, NEVER throw it in the trash, but go to the RECYCLABLE POINT or the nearest waste collection..
	CE marked device certified by the Notified Body n. 0120 of the European Community.
	ISO 60417-5140 Non-Ionizing Electromagnetic radiation.
	ISO 7000-0632 Transport and storage temperature conditions

Symbol	Description
	ISO 7000-2620 Transport and storage humidity conditions
	ISO 7000-2621 Transport and storage atmospheric pressure conditions
	Transport package shall be kept away from rain and in dry condition.
	Transport package shall not be exposed to sunlight.
	IEC 60417-5333 BF Type applicable part according to UNE-EN 6061
IP 21	Medical device is protected from objects not greater than 12 mm in diameter and protected from dripping water
	Direct Current symbol

VI. Error Messages

The following messages might appear during normal operation:

Error message	Cause	Actions
Connection lost	The computer cannot communicate with the device.	Check that the device is switched on, that the device has battery, that the computer is communicating properly, and the device is close to the computer.

VII. Electromagnetic Compatibility (EMC) Information

For Professional Use

The Enobio is suitable for use in the specified electromagnetic environment. The customer and/or user of the Enobio should ensure that is used in an electromagnetic environment as described below:

Emissions Test	Compliance	Electromagnetic Environment Guidance
RF Emissions CISPR 11:2015 + A1:2016	Group 1	The Enobio uses RF energy only for its internal function. Therefore, its RF emissions are very low and are not likely to cause any interference in nearby electronic equipment.
RF Emissions CISPR 11:2015 + A1:2016	Class B	The Enobio is suitable for use in all establishments, including domestic establishments and those directly connected to the public low-voltage power supply network that supplies buildings used for domestic purposes.
Harmonic Emissions UNE-EN 61000-3-2:2014	Class A	
Voltage fluctuations/flicker emissions UNE-EN 61000-3-2:2013	Complies	

The Enobio image intensifier is suitable for use in the specified electromagnetic environment. The customer and/or the user of Enobio image intensifier should ensure that it is used in an electromagnetic environment as described below:

Immunity Test	IEC 60601-1-2 Test Level	Compliance Level
Electrostatic discharge (ESD) UNE-EN 61000-4-2:2010	+ - 8 kV contact	+ - 8 kV contact
	+ - 15 kV air	+ - 15 kV air
Electrical fast transient/burst UNE-EN 61000-4-4:2013	2 kV for power supply lines	2 kV for power supply lines
	1 kV for input/output lines	N/A
Surge UNE-EN 61000-4-5:2015	1 kV differential mode	1 kV differential mode
	2 kV common mode	N/A
Voltage dips, short interruptions and voltage variations on power supply input lines UNE-EN 61000-4-11:2005	0% U; 0.5 cycles at 0°, 45°, 90°, 135°, 180°, 225°, 270° and 315°	0% U; 0.5 cycles at 0°, 45°, 90°, 135°, 180°, 225°, 270° and 315°
	0% U; 1 cycle	0% U; 1 cycle
	70% U; 25 cycles single phase at 0°	70% U; 25 cycles single phase at 0°
	0% U; 250 cycles	0% U; 250 cycles
Power frequency (50/60 Hz) magnetic field IEC 61000-4-8:2011	30 A/m	30 A/m

Note: U is the A/C's main voltage prior to application of the test level.

The Enobio is suitable for use in the specified electromagnetic environment. The customer and/or the user of Enobio should ensure that it is used in an electromagnetic environment as described below:

Immunity Test	IEC 60601-1-2 Test Level	Compliance Level	Electromagnetic Environment Guidance
Conducted RF IEC 61000-4-6:2014	3 Vrms 0.15 MHz - 80 MHz 6 Vrms in ISM bands between 0,15 MHz and 80 MHz 80% AM at 1 kHz	3 Vrms 0.15 MHz - 80 MHz 6 Vrms in ISM bands between 0,15 MHz and 80 MHz 80% AM at 1 kHz	Portable and mobile RF communications equipment should be used no closer to any part of the Enobio, including cables, than the recommended separation distance calculated from the equation appropriate for the frequency of the transmitter Recommended Separation Distance d=1,2 P d=1,2 P 80 MHz to 800 MHz d=2,3 P 800 MHz to 2,5 GHz where P is the maximum output power rating of the transmitter in watts (W) according to the transmitter manufacturer and d is the recommended separation distance in metres (m). Field strengths from fixed RF transmitters, as determined by an electromagnetic site survey (a) , should be less than the compliance level in each frequency range (b) . Interference may occur in the vicinity of equipment marked with the following symbol:
Radiated RF IEC-EN 61000-4-3:2007 + A1:2008 + A2:2011	3V/m 80 MHz - 2.7 GHz	3V/m 80 MHz - 2.7 GHz	
Proximity Fields IEC 61000-4-3:2007+ A1:2008 + A2:2011	27V/m 380 MHz - 390 MHz, 1.8W 28V/m 430 MHz - 479 MHz, 2W 9V/m 705 MHz - 787 MHz, 0.2W 28V/m 800 MHz - 960 MHz, 2W 28V/m 1700 MHz - 1990 MHz, 2W 28V/m 2400 MHz - 2570 MHz, 2W 9V/m 5100 MHz - 5800 MHz, 0.2W	27V/m 380 MHz - 390 MHz, 1.8W 28V/m 430 MHz - 479 MHz, 2W 9V/m 705 MHz - 787 MHz, 0.2W 28V/m 800 MHz - 960 MHz, 2W 28V/m 1700 MHz - 1990 MHz, 2W 28V/m 2400 MHz - 2570 MHz, 2W 9V/m 5100 MHz - 5800 MHz, 0.2W	

(a) Field strengths from fixed transmitters, such as base stations for radio (cellular/cordless) telephones and land mobile radios, amateur radio, AM and FM radio broadcast, and TV broadcast cannot be predicted theoretically with accuracy. To assess the electromagnetic environment due to fixed RF transmitters, an electromagnetic site survey should be considered. If the measured field strength outside the shielded location in which the Enobio is used exceeds the applicable RF compliance level above, the Enobio should be

observed to verify normal operation. If abnormal performance is observed, additional measures may be necessary, such as re-orienting or relocating the Enobio. **(b)** Over the frequency range 150 kHz to 80 MHz, field strengths should be less than 3 V/m.

NOTE 1: These guidelines may not apply in all situations. Electromagnetic propagation is affected by absorption and reflection from structures, objects and people.

The Enobio is intended to be used in a controlled radiated RF environment. The user of the Enobio can help to prevent electromagnetic interferences keeping a minimum distance between the portable RF device (transmitter) and the Enobio as recommended below, according to the maximum output power of the communications device.

Rated Maximum Output Power of Transmitter watts	Separation distance metres		
	150 kHz to 80 MHz $d=1,2 P$	150 kHz to 800 MHz $d=1,2P$	800 MHz to 2,5 GHz $d=2,3 P$
0.01	0.12	0.12	0.23
0.1	0.38	0.38	0.73
1	1.2	1.2	2.3
10	3.8	3.8	7.3
100	12	12	23

For transmitters rated at a maximum output power not listed above, the separation distance can be estimated using the equation in the corresponding column, where P is the maximum output power rating of the transmitter in watts (W) according to the transmitter manufacturer.

NOTE: These guidelines may not apply in all situations. Electromagnetic propagation is affected by absorption and reflection from structures, objects, and people.

NI USB-6008/6009

Bus-Powered Multifunction DAQ USB Device

Français	Deutsch	日本語	한국어	简体中文
ni.com/manuals				

The National Instruments USB-6008/6009 devices provide eight single-ended analog input (AI) channels, two analog output (AO) channels, 12 DIO channels, and a 32-bit counter with a full-speed USB interface. This user guide describes how to use these devices.

For specifications, refer to the *NI USB-6008 Device Specifications* and the *NI USB-6009 Device Specifications* available at ni.com/manuals.

The following table compares the NI USB-6008 and NI USB-6009 devices.

Table 1. NI USB-6008 and NI USB-6009 Comparison

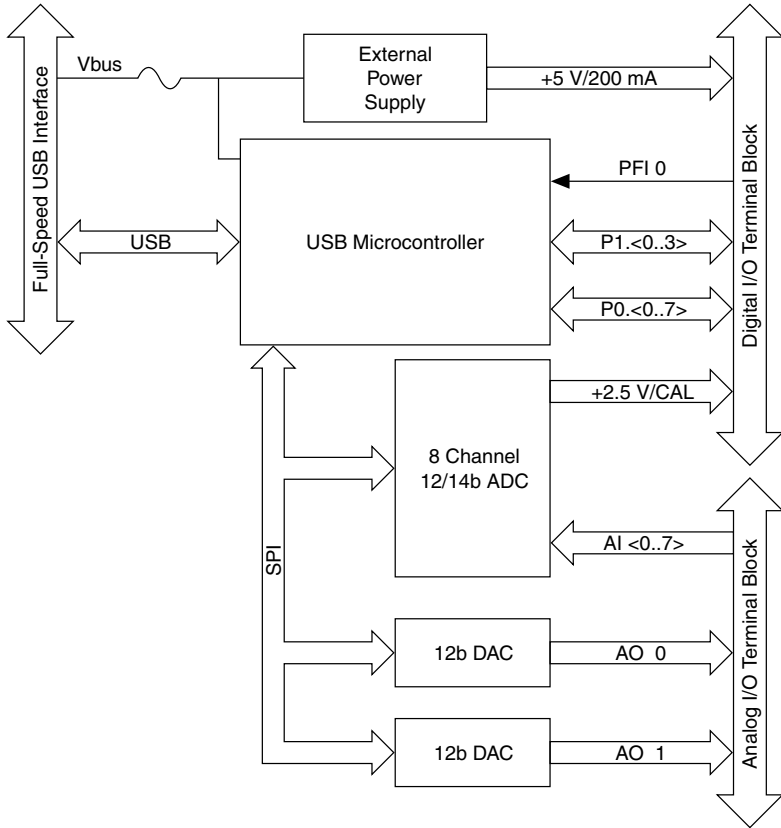
Feature	NI USB-6008	NI USB-6009
AI resolution	12 bits differential, 11 bits single-ended	14 bits differential, 13 bits single-ended
Maximum AI sample rate, single channel ¹	10 kS/s	48 kS/s
Maximum AI sample rate, multiple channels (aggregate) ¹	10 kS/s	48 kS/s
DIO configuration	Open collector ²	Each channel individually programmable as open collector or active drive ²

The following figure shows key functional components of the NI USB-6008/6009.

¹ System-dependent.

² This document uses NI-DAQmx naming conventions. Open-drain is called open collector and push-pull is called active drive.

Figure 1. NI USB-6008/6009 Block Diagram



Safety Guidelines

The following section contains important safety information that you must follow when installing and using the NI USB-6008/6009.



Caution Operate the NI USB-6008/6009 only as described in these operating instructions. Misuse of the device can result in a hazard. You can compromise the safety protection built into the device if the device is damaged in any way. If the device is damaged, contact National Instruments for repair.



Caution Do not substitute parts or modify the device except as described in this user guide. Use the device only with the accessories specified in the installation instructions.



Caution Do not operate the device in an explosive atmosphere or where there may be flammable gases or fumes. If you must operate the device in such an environment, it must be in a suitably rated enclosure.

Electromagnetic Compatibility Guidelines

This product was tested and complies with the regulatory requirements and limits for electromagnetic compatibility (EMC) stated in the product specifications. These requirements and limits provide reasonable protection against harmful interference when the product is operated in the intended operational electromagnetic environment.

This product is intended for use in industrial locations. However, harmful interference may occur in some installations, when the product is connected to a peripheral device or test object, or if the product is used in residential or commercial areas. To minimize interference with radio and television reception and prevent unacceptable performance degradation, install and use this product in strict accordance with the instructions in the product documentation.

Furthermore, any changes or modifications to the product not expressly approved by National Instruments could void your authority to operate it under your local regulatory rules.



Caution To ensure the specified EMC performance, operate this product only with shielded cables and accessories.



Caution This product may become more sensitive to electromagnetic disturbances in the operational environment when test leads are attached or when the product is connected to a test object.



Caution Emissions that exceed the regulatory requirements may occur when this product is connected to a test object.



Note Changes or modifications to the product not expressly approved by National Instruments could void your authority to operate the product under your local regulatory rules.

Unpacking the Kit



Caution To prevent electrostatic discharge (ESD) from damaging the device, ground yourself using a grounding strap or by holding a grounded object, such as your computer chassis.

1. Touch the antistatic package to a metal part of the computer chassis.
2. Remove the device from the package and inspect the device for loose components or any other sign of damage.



Caution Never touch the exposed pins of connectors.



Note Do not install a device if it appears damaged in any way.

3. Unpack any other items and documentation from the kit.

Store the device in the antistatic package when the device is not in use.

Setting Up the NI USB-6008/6009

Complete the following steps to get started with the NI USB-6008/6009.



Note For information about non-Windows operating support, refer to the *Getting Started with NI-DAQmx Base* document available from ni.com/manuals.

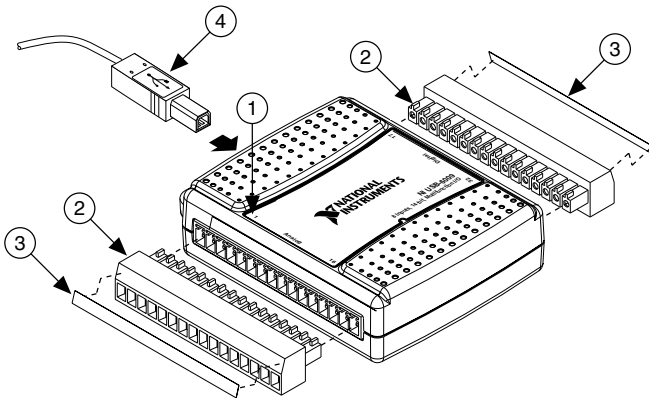
1. Install the application software (if applicable), as described in the installation instructions that accompany your software.
2. Install NI-DAQmx³.



Note The NI-DAQmx software is included on the disk shipped with your kit and is available for download at ni.com/support. The documentation for NI-DAQmx is available after installation from **Start»All Programs»National Instruments»NI-DAQmx**. Other NI documentation is available from ni.com/manuals.

3. Install the 16-position screw terminal plugs by inserting them into the connector jacks, as shown in the following figure.

Figure 2. Signal Label Application Diagram



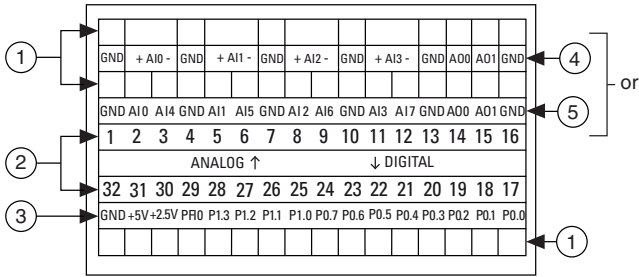
- | | |
|--|-----------------|
| 1. Overlay Label with Pin Orientation Guides | 3. Signal Label |
| 2. Screw Terminal Connector Plug | 4. USB Cable |

4. Affix the provided signal labels to the screw terminal connector plugs. You can choose labels with pin numbers, signal names, or blank labels, as shown in the following figure.

³ NI USB-6008/6009 devices are supported by NI-DAQmx 7.5 and later.

Choose one of the labels, align the correct label with the terminals printed on the top panel of your device and apply the label, as shown in the previous figure.

Figure 3. NI USB-6008/6009 Signal Labels



- | | |
|------------------------------|--|
| 1. User-Defined Custom Label | 4. Analog Input Differential Signal Name Label |
| 2. Terminal Number Label | 5. Analog Input Single-Ended Signal Name Label |
| 3. Digital I/O Label | |



Note After you label the screw terminal connector plugs, you must only insert them into the matching connector jack, as indicated by the overlay label on the device.

- Plug one end of the USB cable into the NI USB-6008/6009 and the other end into an available USB port on the computer.
- Double-click the **NI MAX** icon on the desktop to open Measurement & Automation Explorer (MAX).
- Expand **My System»Devices and Interfaces** and verify that the NI USB-6008/6009 is listed. If your device does not appear, press <F5> to refresh the view in MAX. If your device is still not recognized, refer to ni.com/support/daqmx for troubleshooting information.
- Self-test your device in MAX by right-clicking **NI USB-6008** or **NI USB-6009** and selecting **Self-Test**. Self-test performs a brief test to determine successful device installation. When the self-test finishes, a message indicates successful verification or if an error occurred. If an error occurs, refer to ni.com/support/daqmx.



Caution To ensure the specified EMC performance, operate this product only with shielded cables and accessories.

- Connect the wires (16 AWG to 28 AWG) of a shielded, multiconductor cable to screw terminals by stripping 6.35 mm (0.25 in) of insulation, inserting the wires into the screw terminals, and securely tightening the screws with the flathead screwdriver to a torque of 0.22 N · m to 0.25 N · m (2.0 lb · in. to 2.2 lb · in.). Refer to the [Pinout and Signal Descriptions](#) section for an image of the NI USB-6008/6009 pinout.

If using a shielded cable, connect the cable shield to a nearby GND terminal.



Note For information about sensors, go to ni.com/sensors. For information about IEEE 1451.4 TEDS smart sensors, go to ni.com/teds.

- Run a Test Panel in MAX by right-clicking **NI USB-6008** or **NI USB-6009** and selecting **Test Panels**.

Click **Start** to test the device functions, or **Help** for operating instructions. Click **Close** to exit the test panel.

Using the NI USB-6008/6009 in an Application

You can use the DAQ Assistant through many NI application software programs to configure virtual and measurement channels. The following table lists DAQ Assistant tutorial locations for NI applications.

Table 2. DAQ Assistant Tutorial Locations

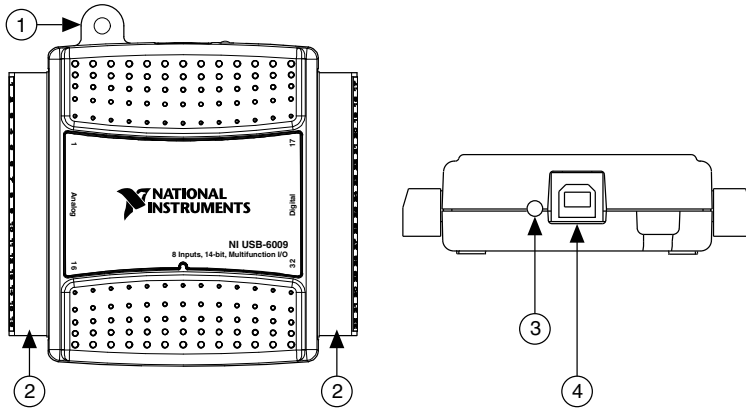
NI Application	Tutorial Location
LabVIEW	Go to Help»LabVIEW Help . Next, go to Getting Started with LabVIEW»Getting Started with DAQ»Taking an NI-DAQmx Measurement in LabVIEW .
LabWindows™/CVI™	Go to Help»Contents . Next, go to Using LabWindows/CVI»Data Acquisition»Taking an NI-DAQmx Measurement in LabWindows/CVI .
Measurement Studio	Go to NI Measurement Studio Help»Getting Started with the Measurement Studio Class Libraries»Measurement Studio Walkthroughs»Walkthrough: Creating a Measurement Studio NI-DAQmx Application .
LabVIEW SignalExpress	Go to Help»Taking an NI-DAQmx Measurement in SignalExpress .

Refer to the [Where to Go from Here](#) section for information about programming examples for NI-DAQmx and NI-DAQmx Base.

Features

The NI USB-6008/6009 features a USB connector, USB cable strain relief, two screw terminal connector plugs for I/O, and an LED indicator, as shown in the following figure.

Figure 4. NI USB-6008/6009 Top and Back Views

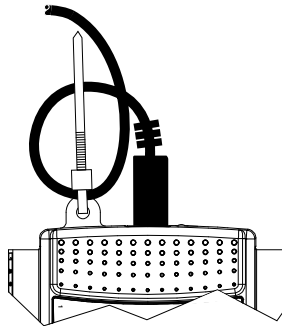


- | | |
|----------------------------------|------------------|
| 1. USB Cable Strain Relief | 3. LED Indicator |
| 2. Screw Terminal Connector Plug | 4. USB Connector |

USB Connector and USB Cable Strain Relief

The NI USB-6008/6009 features a USB connector for full-speed USB interface. You can provide strain relief for the USB cable by threading a zip tie through the USB cable strain relief ring and tightening around a looped USB cable, as shown in the following figure.

Figure 5. NI USB-6008/6009 Strain Relief



LED Indicator

The NI USB-6008/6009 device has a green LED indicator that indicates device status, as listed in the following table. When the device is connected to a USB port, the LED blinks steadily to indicate that the device is initialized and is receiving power from the connection.

Table 3. LED State/Device Status

LED State	Device Status
Not lit	Device not connected or in suspend
On, not blinking	Device connected but not initialized, or the computer is in standby mode. In order for the device to be recognized, the device must be connected to a computer that has NI-DAQmx installed on it.
Single-blink	Operating normally

Screw Terminal Connector Plugs

The NI USB-6008/6009 ships with one detachable screw terminal connector plug for analog signals and one detachable screw terminal connector plug for digital signals. These screw terminal connectors provide 16 connections that use 16 AWG to 28 AWG wire. Refer to step 4 of the [Setting Up the NI USB-6008/6009](#) section for information about selecting labels for the screw terminal connector plugs. Refer to the [Pinout and Signal Descriptions](#) section for the device pinout and signal descriptions.

You can order additional connectors and labels for your device. Refer to the [Cables and Accessories](#) section for ordering information.

Firmware

The firmware on the NI USB-6008/6009 refreshes whenever the device is connected to a computer with NI-DAQmx. NI-DAQmx automatically uploads the compatible firmware version to the device. The firmware version may be upgraded when new versions of NI-DAQmx release.

Cables and Accessories

The following table contains information about cables and accessories available for the NI USB-6008/6009. For a complete list of accessories and ordering information, refer to the pricing section of the NI USB-6008 or NI USB-6009 product page at ni.com.

Table 4. NI USB-6008/6009 Cables and Accessories

Accessory	Part Number	Description
USB-6008/6009 Accessory Kit	779371-01	Four additional screw-terminal connectors, connector labels, and a screwdriver
USB-6000 Series Prototyping Accessory	779511-01	Unshielded breadboarding accessory for custom-defined signal conditioning and prototyping. You can use up to two accessories per device.
Hi-Speed USB Cable	184125-01, 184125-02	1 m and 2 m lengths



Caution For compliance with Electromagnetic Compatibility (EMC) requirements, this product must be operated with shielded cables and accessories. If unshielded cables or accessories are used, the EMC specifications are no longer guaranteed unless all unshielded cables and/or accessories are installed in a shielded enclosure with properly designed and shielded input/output ports.

Pinout and Signal Descriptions

The following figure shows the pinout of the NI USB-6008/6009. Analog input signal names are listed as single-ended analog input name, AI x , and then differential analog input name, (AI x +/-). Refer to the following table for a detailed description of each signal.

Figure 6. NI USB-6008/6009 Pinout

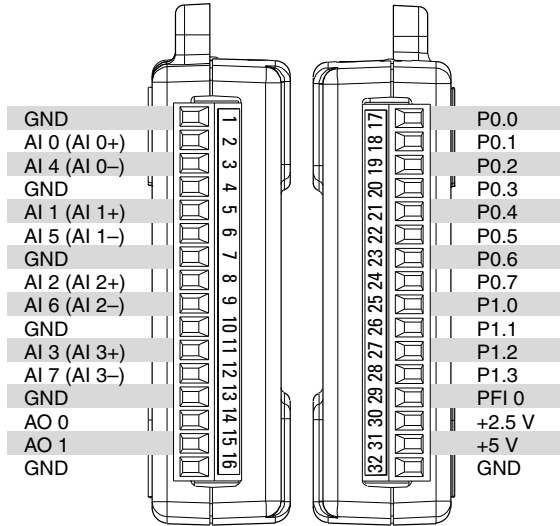


Table 5. Signal Descriptions

Signal Name	Reference	Direction	Description
GND	—	—	Ground —The reference point for the single-ended analog input measurements, analog output voltages, digital signals, +5 VDC supply, and +2.5 VDC at the I/O connector, and the bias current return point for differential mode measurements.
AI <0..7>	Varies	Input	Analog Input Channels 0 to 7 —For single-ended measurements, each signal is an analog input voltage channel. For differential measurements, AI 0 and AI 4 are the positive and negative inputs of differential analog input channel 0. The following signal pairs also form differential input channels: AI <1, 5>, AI <2, 6>, and AI <3, 7>. Refer to the <i>Analog Input</i> section for more information.

Table 5. Signal Descriptions (Continued)

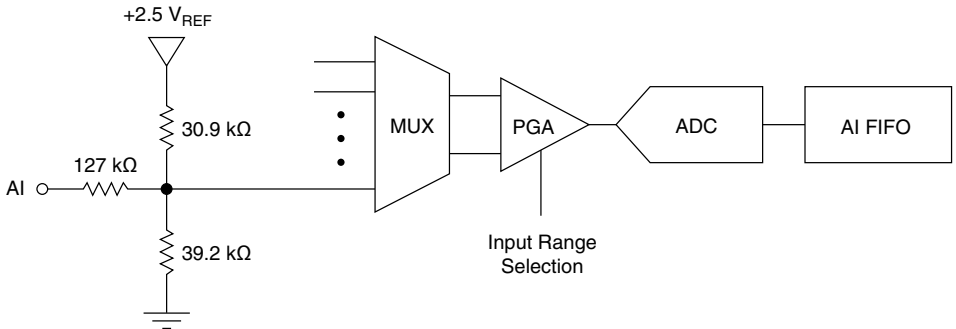
Signal Name	Reference	Direction	Description
AO <0, 1>	GND	Output	Analog Output Channels 0 and 1 —Supplies the voltage output of AO channel 0 or AO channel 1. Refer to the <i>Analog Output</i> section for more information.
P0.<0..7>	GND	Input or Output	Port 0 Digital I/O Channels 0 to 7 —You can individually configure each signal as an input or output. Refer to the <i>Digital I/O</i> section for more information.
P1.<0..3>	GND	Input or Output	Port 1 Digital I/O Channels 0 to 3 —You can individually configure each signal as an input or output. Refer to the <i>Digital I/O</i> section for more information.
PFI 0	GND	Input	PFI 0 —This pin is configurable as either a digital trigger or an event counter input. Refer to the <i>PFI 0</i> section for more information.
+2.5 V	GND	Output	+2.5 V External Reference —Provides a reference for wrap-back testing. Refer to the <i>+2.5 V External Reference</i> section for more information.
+5 V	GND	Output	+5 V Power Source —Provides +5 V power up to 200 mA. Refer to the <i>+5 V Power Source</i> section for more information.

Analog Input

The NI USB-6008/6009 has eight analog input channels that you can use for four differential analog input measurements or eight single-ended analog input measurements.

The following figure shows the analog input circuitry of the NI USB-6008/6009.

Figure 7. NI USB-6008/6009 Analog Input Circuitry



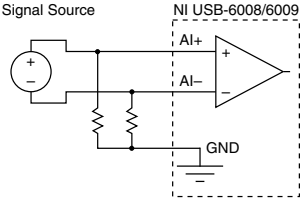
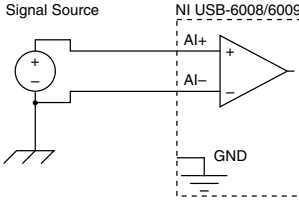
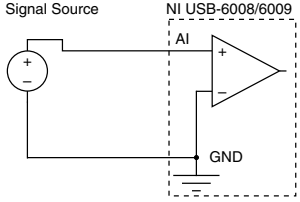
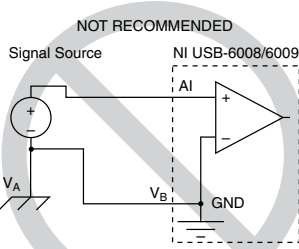
The main blocks featured in the NI USB-6008/6009 analog input circuitry are as follows:

- **MUX**—The NI USB-6008/6009 has one analog-to-digital converter (ADC). The multiplexer (MUX) routes one AI channel at a time to the PGA.
- **PGA**—The programmable-gain amplifier provides input gains of 1, 2, 4, 5, 8, 10, 16, or 20 when configured for differential measurements and gain of 1 when configured for single-ended measurements. The PGA gain is automatically calculated based on the voltage range selected in the measurement application.
- **ADC**—The analog-to-digital converter (ADC) digitizes the AI signal by converting the analog voltage into digital code.
- **AI FIFO**—The NI USB-6008/6009 can perform both single and multiple analog-to-digital conversions of a fixed or infinite number of samples. A first-in-first-out (FIFO) buffer holds data during AI acquisitions to ensure that no data is lost.

Analog Input Modes and Signal Sources

You can configure the AI channels on the NI USB-6008/6009 to take differential or referenced single-ended (RSE) measurements. The following table summarizes the recommended analog input mode(s) for floating signal sources and ground-references signal sources. Refer to the table in the [Pinout and Signal Descriptions](#) section for more information about I/O configurations for single-ended or differential measurements.

Table 6. Analog Input Configurations

Analog Input Mode	Floating Signal Sources (Not Connected to Building Ground)	Ground-Referenced Signal Sources
Examples	<ul style="list-style-type: none"> • Ungrounded thermocouples • Signal conditioning with isolated outputs • Battery devices 	Plug-in instruments with non-isolated outputs
Differential (DIFF)		
Referenced Single-Ended (RSE)		<p style="text-align: center;">NOT RECOMMENDED</p>  <p style="text-align: center;">Ground-loop potential ($V_A - V_B$) are added to measured signal.</p>

Floating Signal Sources

A floating signal source is not connected to the building ground system, but has an isolated ground-reference point. Some examples of floating signal sources are outputs of transformers, thermocouples, battery-powered devices, optical isolators, and isolation amplifiers. An instrument or device that has an isolated output is a floating signal source.

Refer to *Field Wiring and Noise Considerations for Analog Signals* for more information. To access this document, go to ni.com/info and enter the Info Code `rdfwn3`.

When to Use Differential Connections with Floating Signal Sources

Use DIFF input connections for any channel that meets any of the following conditions:

- Your application requires input ranges other than ± 10 V.
- The input signal is low level and requires greater accuracy.

- The leads connecting the signal to the device are greater than 3 m (10 ft).
- The input signal requires a separate ground-reference point or return signal.
- The signal leads travel through noisy environments.
- Two analog channels, AI+ and AI-, are available for the signal.

DIFF signal connections reduce noise pickup and increase common-mode noise rejection. DIFF signal connections also allow input signals to float within the working voltage of the device.

Refer to the [Taking Differential Measurements](#) section for more information about differential connections.

When to Use Referenced Single-Ended (RSE) Connections with Floating Signal Sources

Only use RSE input connections if the input signal meets all of the following conditions:

- The input signal can share a common reference point, GND, with other signals that use RSE.
- Your application permits the use of the ± 10 V input range.
- The leads connecting the signal to the device are less than 3 m (10 ft).

DIFF input connections are recommended for greater signal integrity for any input signal that does not meet the preceding conditions.

In the single-ended modes, more electrostatic and magnetic noise couples into the signal connections than in DIFF configurations. The coupling is the result of differences in the signal path. Magnetic coupling is proportional to the area between the two signal conductors. Electrical coupling is a function of how much the electric field differs between the two conductors.

With this type of connection, the PGA rejects both the common-mode noise in the signal and the ground potential difference between the signal source and the device ground.

Refer to the [Taking Referenced Single-Ended Measurements](#) section for more information about RSE connections.

Ground-Referenced Signal Sources

A ground-referenced signal source is a signal source connected to the building system ground. It is already connected to a common ground point with respect to the device, assuming that the computer is plugged into the same power system as the source. Non-isolated outputs of instruments and devices that plug into the building power system fall into this category.

The difference in ground potential between two instruments connected to the same building power system is typically between 1 mV and 100 mV, but the difference can be much higher if power distribution circuits are improperly connected. If a grounded signal source is incorrectly measured, this difference can appear as measurement error. Follow the connection instructions

for grounded signal sources to eliminate this ground potential difference from the measured signal.

Refer to *Field Wiring and Noise Considerations for Analog Signals* for more information. To access this document, go to ni.com/info and enter the Info Code `rdfwn3`.

When to Use Differential Connections with Ground-Reference Signal Sources

Use DIFF input connections for any channel that meets any of the following conditions:

- Your application requires input ranges other than ± 10 V.
- The input signal is low level and requires greater accuracy.
- The leads connecting the signal to the device are greater than 3 m (10 ft).
- The input signal requires a separate ground-reference point or return signal.
- The signal leads travel through noisy environments.
- Two analog channels, AI+ and AI-, are available for the signal.

DIFF signal connections reduce noise pickup and increase common-mode noise rejection. DIFF signal connections also allow input signals to float within the working voltage of the device.

Refer to the [Taking Differential Measurements](#) section for more information about differential connections.

When to Use Referenced Single-Ended (RSE) Connections with Ground-Referenced Signal Sources

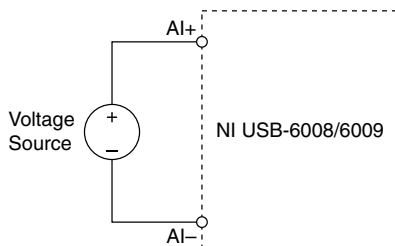
Do not use RSE connections with ground-referenced signal sources. Use differential connections instead.

As shown in the table in the [Analog Input Modes and Signal Sources](#) section, there can be a potential difference between GND and the ground of the sensor. In RSE mode, this ground loop causes measurement errors.

Taking Differential Measurements

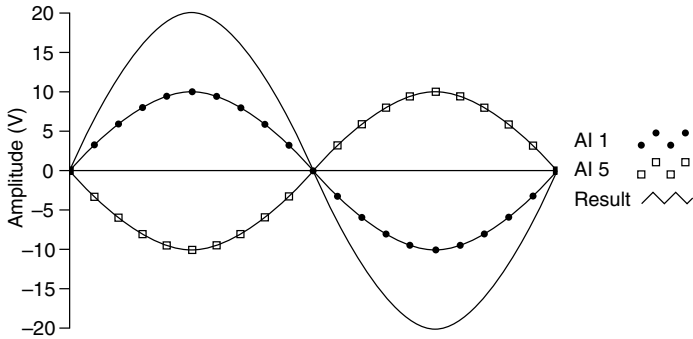
For differential signals, connect the positive lead of the signal to the AI+ terminal, and the negative lead to the AI- terminal.

Figure 8. Connecting a Differential Voltage Signal



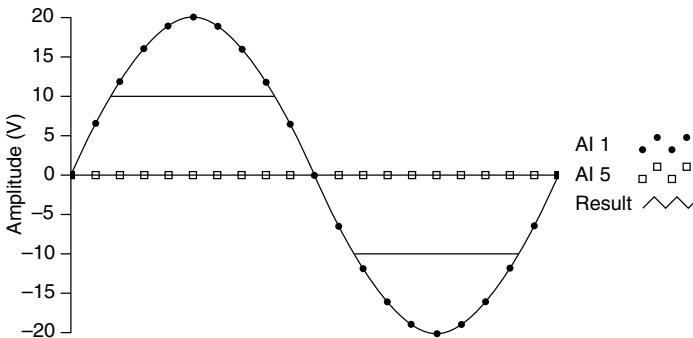
The differential input mode can measure ± 20 V signals in the ± 20 V range. However, the maximum voltage on any one pin is ± 10 V with respect to GND. For example, if AI 1 is $+10$ V and AI 5 is -10 V, then the measurement returned from the device is $+20$ V.

Figure 9. Example of a Differential 20 V Measurement



Connecting a signal greater than ± 10 V on either pin results in a clipped output.

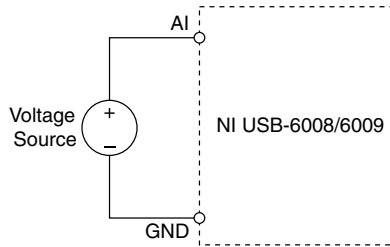
Figure 10. Exceeding ± 10 V on AI Returns Clipped Output



Taking Referenced Single-Ended Measurements

To connect referenced single-ended (RSE) voltage signals to the NI USB-6008/6009, connect the positive voltage signal to an AI terminal, and the ground signal to a GND terminal, as shown in the following figure.

Figure 11. Connecting a Referenced Single-Ended Voltage Signal



When no signals are connected to the analog input terminal, the internal resistor divider may cause the terminal to float to approximately 1.4 V when the analog input terminal is configured as RSE. This behavior is normal and does not affect the measurement when a signal is connected.

Digital Trigger

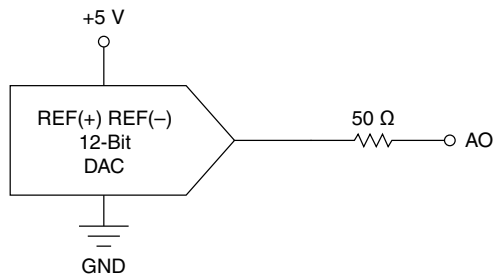
You can configure PFI 0 as a digital trigger input for analog input tasks. Refer to the [Using PFI 0 as a Digital Trigger](#) section for more information.

Analog Output

The NI USB-6008/6009 has two independent analog output channels that can generate outputs from 0 V to 5 V. All updates of analog output channels are software-timed. GND is the ground-reference signal for the analog output channels.

The following figure shows the circuitry of one analog output channel on the NI USB-6008/6009.

Figure 12. Circuitry of One Analog Output Channel

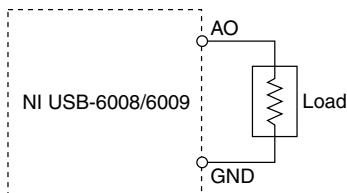


The main block featured in the NI USB-6008/6009 analog output circuitry is the digital-to-analog converter (DAC), which converts digital codes to analog voltages. There is one DAC for each analog output line.

Connecting Analog Output Loads

To connect loads to the NI USB-6008/6009, connect the positive lead of the load to the AO terminal, and connect the ground of the load to a GND terminal, as shown in the following figure.

Figure 13. Connecting a Load



Minimizing Glitches on the Output Signal

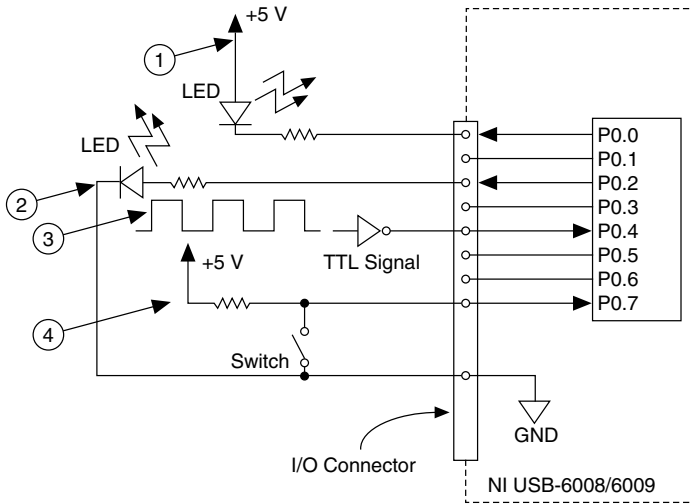
When you use a DAC to generate a waveform, you may observe glitches in the output signal. These glitches are normal; when a DAC switches from one voltage to another, it produces glitches due to released charges. The largest glitches occur when the most significant bit of the DAC code changes. You can build a lowpass deglitching filter to remove some of these glitches, depending on the frequency and nature of the output signal. For more information about minimizing glitches, refer to the KnowledgeBase document, *Reducing Glitches on the Analog Output of MIO DAQ Devices*. To access this document, go to ni.com/info and enter the Info Code `exszek`.

Digital I/O

The NI USB-6008/6009 has 12 digital lines on two ports, Port 0 has eight lines, P0.<0..7>, and Port 1 has four lines, P1.<0..3>. GND is the ground-reference signal for the digital I/O ports. You can individually program all lines as inputs or outputs.

The following figure shows P0.<0..7> connected to example signals configured as digital inputs and digital outputs. You can configure P1.<0..3> similarly.

Figure 14. Example of Connecting a Load



1. P0.0 configured as an open collector digital output driving an LED
2. P0.2 configured as an active drive digital output driving an LED
3. P0.4 configured as a digital input receiving a TTL signal from a gated inverter
4. P0.7 configured as a digital input receiving a 0 V or 5 V signal from a switch



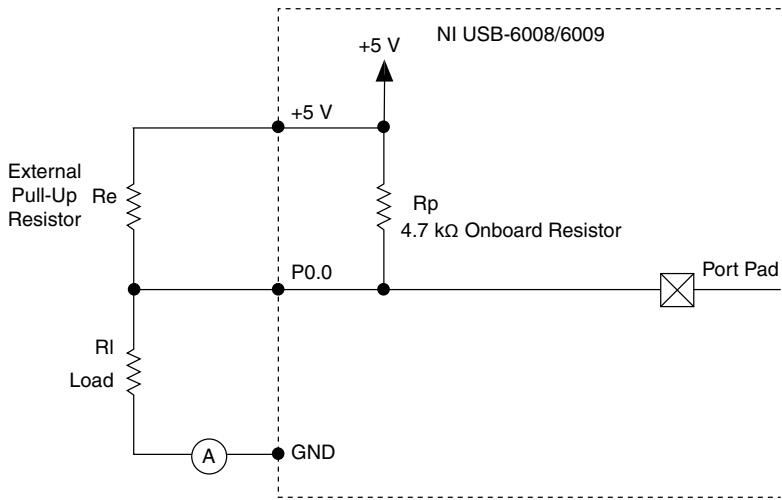
Caution Exceeding the maximum input voltage ratings or maximum output ratings can damage the device and the computer. National Instruments is not liable for any damage resulting from such signal connections. Refer to the *NI USB-6008 Device Specifications* or *NI USB-6009 Device Specifications* for more information.

Source/Sink Information

The default configuration of the NI USB-6008/6009 digital I/O ports is open collector, allowing 5 V operation, with an onboard 4.7 k Ω pull-up resistor. An external user-provided pull-up resistor can be added to increase the source current drive up to a 8.5 mA limit per line, as shown in the following figure.⁴

⁴ This document uses NI-DAQmx naming conventions. Open-drain is called open collector and push-pull is called active drive.

Figure 15. Example of Connecting an External User-Provided Resistor



The NI USB-6009 ports can also be configured as active drive using the NI-DAQmx API, allowing 3.3 V operation with a source/sink current limit of ± 8.5 mA. For more information about how to set the DIO configuration, refer to the KnowledgeBase document, *How Do I Configure My NI Device to be Open-Drain (Open Collector) or Push-Pull (Active Drive)?*. To access this document, go to ni.com/info and enter the Info Code `ex52sp`.

Determining the Value of the User-Provided Pull-Up Resistor

Complete the following steps to determine the value of the user-provided pull-up resistor:

1. Place an ammeter in series with the load.
2. Place a variable resistor between the digital output line and the +5 V supply.
3. Set P0.0 to high.
4. Adjust the variable resistor until the ammeter current reads as the intended current. The intended current must be less than 8.5 mA.
5. Remove the ammeter and variable resistor from your circuit.
6. Measure the resistance of the variable resistor. The measured resistance is the ideal value of the pull-up resistor.
7. Select a static resistor value for your pull-up resistor that is greater than or equal to the ideal resistance.
8. Reconnect the load circuit and the pull-up resistor.

I/O Protection

To protect the NI USB-6008/6009 against overvoltage, undervoltage, and overcurrent conditions, as well as ESD events, you should avoid these fault conditions by using the following guidelines:

- If you configure a DIO line as an output, do not connect it to any external signal source, ground signal, or power supply.
- If you configure a DIO line as an output, understand the current requirements of the load connected to these signals. Do not exceed the specified current output limits of the DAQ device.
National Instruments has several signal conditioning solutions for digital applications requiring high current drive.
- If you configure a DIO line as an input, do not drive the line with voltages outside of its normal operating range. The DIO lines have a smaller operating range than the AI signals.
- Treat the DAQ device as you would treat any static-sensitive device. Always properly ground yourself and the equipment when handling the DAQ device or connecting to it.

Power-On States

At system startup and reset, the hardware sets all DIO lines to high-impedance inputs. The DAQ device does not drive the signal high or low. Each line has a weak pull-up resistor connected to it.

Static DIO

Each of the NI USB-6008/6009 DIO lines can be used as a static DI or DO line. You can use static DIO lines to monitor or control digital signals. All samples of static DI lines and updates of DO lines are software-timed.

PFI 0

PFI 0 is configurable as either a digital trigger input or an event counter input.

Using PFI 0 as a Digital Trigger

When an analog input task is defined, you can configure PFI 0 as a digital trigger input. When the digital trigger is enabled, the AI task waits for a rising or falling edge on PFI 0 before starting the acquisition. To use AI Start Trigger (ai/StartTrigger) with a digital source, specify PFI 0 as the source and select a rising or falling edge.

Using PFI 0 as an Event Counter

You can configure PFI 0 as a source for counting digital edges. In this mode, falling-edge events are counted using a 32-bit counter. For more information about event timing

requirements, refer to the *NI USB-6008 Device Specifications* and *NI USB-6009 Device Specifications* documents available on ni.com/manuals.

External Reference and Power Source

The NI USB-6008/6009 creates an external reference and supplies a power source. All voltages are relative to ground (GND).

+2.5 V External Reference

The NI USB-6008/6009 creates a high-purity reference voltage supply for the ADC using a multi-state regulator, amplifier, and filter circuit. You can use the resulting +2.5 V reference voltage as a signal for self-test.

+5 V Power Source

The NI USB-6008/6009 supplies a 5 V, 200 mA output. You can use this source to power external components.



Note When the device is in USB suspend, the output is disabled.

Where to Go from Here

This section lists where you can find example programs for the NI USB-6008/6009 and relevant documentation.

Example Programs

NI-DAQmx and NI-DAQmx Base software include example programs to help you get started programming with the NI USB-6008/6009. Modify example code and save it in an application, or use examples to develop a new application, or add example code to an existing application.

NI-DAQmx

To locate NI software examples, go to ni.com/info and enter the Info Code `daqmxexp`.

To run examples without the device installed, use an NI-DAQmx simulated device. For more information, in Measurement & Automation Explorer (MAX), select **Help»Help Topics»NI-DAQmx»MAX Help for NI-DAQmx** and search for simulated devices.

NI-DAQmx Base

NI-DAQmx Base examples are accessible from **Start»All Programs»National Instruments»NI-DAQmx Base Examples**.

Related Documentation

Each application software package and driver includes information about writing applications for taking measurements and controlling measurement devices. The following references to documents assume you have NI-DAQmx 9.3 or later, and where applicable, version 8.5 or later of the NI application software.

NI USB-6008/6009

The *NI USB-6008/6009 Quick Start* included with the NI USB-6008/6009 describes how to install NI-DAQmx software, install the device, and confirm that your device is operating properly.

The *NI USB-6008 Device Specifications* and *NI USB-6009 Device Specifications* contain device specifications and are available at ni.com/manuals.

NI-DAQmx

The *NI-DAQmx Readme* is available from the Windows Start menu and lists which application software, devices, and ADEs are supported by this version of NI-DAQmx.

The *NI-DAQmx Help* is available from the Windows Start menu and contains API overviews, general information about measurement concepts, key NI-DAQmx concepts, and common applications that are applicable to all programming environments.

NI-DAQmx Base (Linux/Mac OS X/LabVIEW PDA 8.x)

The *Getting Started with NI-DAQmx Base* document describes how to install the NI-DAQmx Base software, the NI-DAQmx Base-supported DAQ device, and how to confirm that the device is operating properly on your Windows, Linux, or Mac machine. In Windows, select **Start»All Programs»National Instruments»NI-DAQmx Base»Documentation»Getting Started with NI-DAQmx Base**.

The *NI-DAQmx Base Readme* lists devices supported in different versions of NI-DAQmx Base. In Windows, select **Start»All Programs»National Instruments»NI-DAQmx Base»DAQmx Base Readme**.

The *NI-DAQmx Base VI Reference Help* contains VI reference and general information about measurement concepts. In LabVIEW, select **Help»NI-DAQmx Base VI Reference Help**.

The *NI-DAQmx Base C Function Reference Help* contains C reference and general information about measurement concepts. In Windows, select **Start»All Programs»National Instruments»NI-DAQmx Base»Documentation»C Function Reference Help**.



Note All NI-DAQmx Base documentation for Linux is installed at `/usr/local/natinst/nidaqmxbase/documentation`.



Note All NI-DAQmx Base documentation for Mac OS X is installed at / Applications/National Instruments/NI-DAQmx Base/documentation.

LabVIEW

Refer to ni.com/gettingstarted for more information about LabVIEW.

Use the *LabVIEW Help*, available by selecting **Help»LabVIEW Help** in LabVIEW, to access information about LabVIEW programming concepts, step-by-step instructions for using LabVIEW, and reference information about LabVIEW VIs, functions, palettes, menus, and tools. Refer to the following locations on the **Contents** tab of the *LabVIEW Help* for information about NI-DAQmx:

- **VI and Function Reference»Measurement I/O VIs and Functions»DAQmx - Data Acquisition VIs and Functions**—Describes the LabVIEW NI-DAQmx VIs and functions.
- **Property and Method Reference»NI-DAQmx Properties**—Contains the property reference.
- **Taking Measurements**—Contains the conceptual and how-to information you need to acquire and analyze measurement data in LabVIEW, including common measurements, measurement fundamentals, NI-DAQmx key concepts, and device considerations.

LabWindows/CVI

The **Data Acquisition** section of the *LabWindows/CVI Help* contains *Taking an NI-DAQmx Measurement in LabWindows/CVI*, which includes step-by-step instructions for creating a measurement task using the DAQ Assistant. In LabWindows/CVI, select **Help»Contents**, and then select **Using LabWindows/CVI»Data Acquisition**. This book also contains information about accessing detailed information through the *NI-DAQmx Help*.

The **NI-DAQmx Library** section of the *LabWindows/CVI Help* contains API overviews and function reference for NI-DAQmx. Select **Library Reference»NI-DAQmx Library** in the *LabWindows/CVI Help*.

Measurement Studio

If you program your NI-DAQmx-supported device in Measurement Studio using Visual C# or Visual Basic .NET, you can interactively create channels and tasks by launching the DAQ Assistant from MAX or from within Visual Studio. You can use Measurement Studio to generate the configuration code based on your task or channel. Refer to the *DAQ Assistant Help* for additional information about generating code.

The *NI Measurement Studio Help* is fully integrated with the Microsoft Visual Studio help. To view this help file in Visual Studio, select **Measurement Studio»NI Measurement Studio**

Help. For information related to developing with NI-DAQmx, refer to the following topics within the *NI Measurement Studio Help*:

- For step-by-step instructions on how to create an NI-DAQmx application using the Measurement Studio Application Wizard and the DAQ Assistant, refer to *Walkthrough: Creating a Measurement Studio NI-DAQmx Application*.
- For help with NI-DAQmx methods and properties, refer to *NationalInstruments.DAQmx Namespace* and *NationalInstruments.DAQmx.ComponentModel Namespace*.
- For conceptual help with NI-DAQmx, refer to *Using the Measurement Studio NI-DAQmx .NET Library* and *Developing with Measurement Studio NI-DAQmx*.
- For general help with programming in Measurement Studio, refer to *Getting Started with the Measurement Studio Class Libraries*.

To create an application in Visual Basic .NET or Visual C#, follow these general steps:

1. In Visual Studio, select **File»New»Project** to launch the New Project dialog box.
2. In the Project types pane, expand the **Visual Basic** or **Visual C#** node, depending on which language you want to create the project in, and select **Measurement Studio**.
3. Choose a project type. You add DAQ tasks as a part of this step.

ANSI C without NI Application Software

The *NI-DAQmx Help* contains API overviews and general information about measurement concepts. Select **Start»All Programs»National Instruments»NI-DAQmx»NI-DAQmx Help**.

The *NI-DAQmx C Reference Help* describes the NI-DAQmx Library functions, which you can use with National Instruments data acquisition devices to develop instrumentation, acquisition, and control applications. Select **Start»All Programs»National Instruments»NI-DAQmx»Text-Based Code Support»NI-DAQmx C Reference Help**.

.NET Languages without NI Application Software

With the Microsoft .NET Framework, you can use NI-DAQmx to create applications using Visual C# and Visual Basic .NET without Measurement Studio. Refer to the *NI-DAQmx Readme* for specific versions supported.

Training Courses

If you need more help getting started developing an application with NI products, NI offers training courses. To enroll in a course or obtain a detailed course outline, refer to ni.com/training.

Technical Support on the Web

For additional support, refer to ni.com/support or ni.com/examples.



Note You can download these documents at ni.com/manuals.

Many DAQ specifications and user guides/manuals are available as PDFs. You must have Adobe Reader 7.0 or later (PDF 1.6 or later) installed to view the PDFs. Refer to the Adobe Systems Incorporated website at www.adobe.com to download Adobe Reader. Refer to the National Instruments Product Manuals Library at ni.com/manuals for updated documentation resources.

Worldwide Support and Services

The National Instruments website is your complete resource for technical support. At ni.com/support, you have access to everything from troubleshooting and application development self-help resources to email and phone assistance from NI Application Engineers.

Visit ni.com/services for NI Factory Installation Services, repairs, extended warranty, and other services.

Visit ni.com/register to register your National Instruments product. Product registration facilitates technical support and ensures that you receive important information updates from NI.

A Declaration of Conformity (DoC) is our claim of compliance with the Council of the European Communities using the manufacturer's declaration of conformity. This system affords the user protection for electromagnetic compatibility (EMC) and product safety. You can obtain the DoC for your product by visiting ni.com/certification. If your product supports calibration, you can obtain the calibration certificate for your product at ni.com/calibration.

National Instruments corporate headquarters is located at 11500 North Mopac Expressway, Austin, Texas, 78759-3504. National Instruments also has offices located around the world. For telephone support in the United States, create your service request at ni.com/support or dial 1 866 ASK MYNI (275 6964). For telephone support outside the United States, visit the *Worldwide Offices* section of ni.com/niglobal to access the branch office websites, which provide up-to-date contact information, support phone numbers, email addresses, and current events.

Refer to the *NI Trademarks and Logo Guidelines* at ni.com/trademarks for information on National Instruments trademarks. Other product and company names mentioned herein are trademarks or trade names of their respective companies. For patents covering National Instruments products/technology, refer to the appropriate location: **Help»Patents** in your software, the `patents.txt` file on your media, or the *National Instruments Patent Notice* at ni.com/patents. You can find information about end-user license agreements (EULAs) and third-party legal notices in the readme file for your NI product. Refer to the *Export Compliance Information* at ni.com/legal/export-compliance for the National Instruments global trade compliance policy and how to obtain relevant HTS codes, ECCNs, and other import/export data. NI MAKES NO EXPRESS OR IMPLIED WARRANTIES AS TO THE ACCURACY OF THE INFORMATION CONTAINED HEREIN AND SHALL NOT BE LIABLE FOR ANY ERRORS. U.S. Government Customers: The data contained in this manual was developed at private expense and is subject to the applicable limited rights and restricted data rights as set forth in FAR 52.227-14, DFAR 252.227-7014, and DFAR 252.227-7015.

© 2004—2015 National Instruments. All rights reserved.

371303N-01 Jul15



ATTENTION
OBSERVE PRECAUTIONS
FOR HANDLING
ELECTROSTATIC
DISCHARGE
SENSITIVE
DEVICES

Part Number: L-154A4SURKQBDZGC

Hyper Red
Blue
Green

Features

- Uniform light output.
- Low power consumption.
- Long life-solid state reliability.
- RoHS compliant.

Description

The Hyper Red source color devices are made with Al-GaN on GaAs substrate Light Emitting Diode.

The Blue source color devices are made with InGaN Light Emitting Diode.

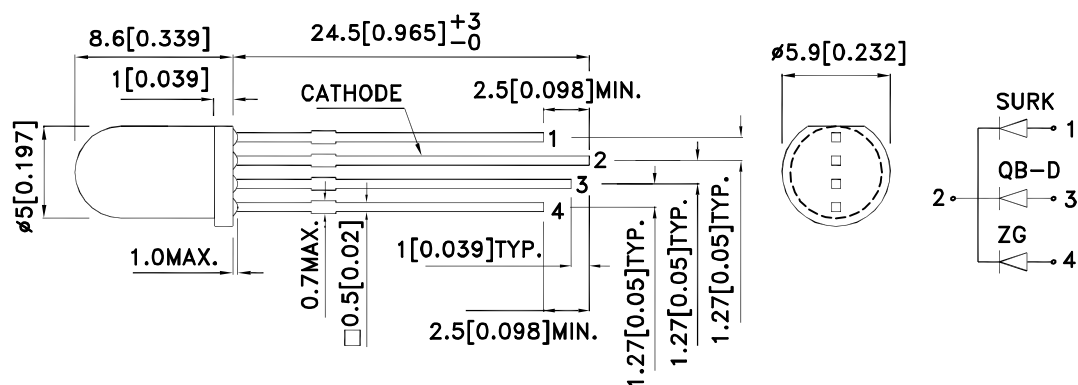
The Green source color devices are made with InGaN on Sapphire Light Emitting Diode.

Static electricity and surge damage the LEDs.

It is recommended to use a wrist band or anti-electrostatic glove when handling the LEDs.

All devices, equipment and machinery must be electrically grounded.

Package Dimensions



Notes:

1. All dimensions are in millimeters (inches).
2. Tolerance is $\pm 0.25(0.01)$ unless otherwise noted.
3. Lead spacing is measured where the leads emerge from the package.
4. The specifications, characteristics and technical data described in the datasheet are subject to change without prior notice.



Selection Guide

Part No.	Dice	Lens Type	Iv (mcd) [2] @ 20mA		Viewing Angle [1]
			Min.	Typ.	2θ1/2
L-154A4SURKQBDZGC	Hyper Red (AlGaInP)	Water Clear	600	1200	50°
	Blue (InGaN)		450	800	
	Green (InGaN)		1000	1700	

Notes:

1. θ1/2 is the angle from optical centerline where the luminous intensity is 1/2 of the optical peak value.
2. Luminous intensity/ luminous Flux: +/-15%.

Electrical / Optical Characteristics at TA=25°C

Symbol	Parameter	Device	Typ.	Max.	Units	Test Conditions
λpeak	Peak Wavelength	Hyper Red Blue Green	650 468 515		nm	IF=20mA
λD [1]	Dominant Wavelength	Hyper Red Blue Green	630 470 525		nm	IF=20mA
Δλ1/2	Spectral Line Half-width	Hyper Red Blue Green	28 25 30		nm	IF=20mA
C	Capacitance	Hyper Red Blue Green	35 100 45		pF	VF=0V;f=1MHz
VF [2]	Forward Voltage	Hyper Red Blue Green	1.95 3.3 3.3	2.5 4 4.1	V	IF=20mA
IR	Reverse Current	Hyper Red Blue Green		10 50 50	uA	VR=5V

Notes:

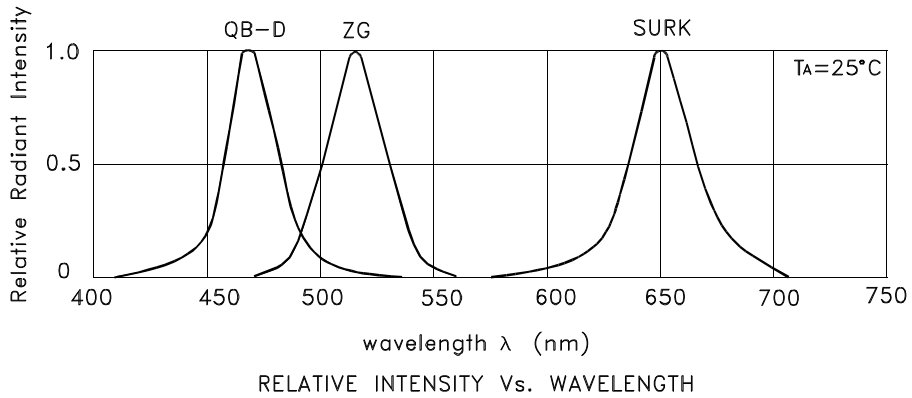
1. Wavelength: +/-1nm.
2. Forward Voltage: +/-0.1V.

Absolute Maximum Ratings at TA=25°C

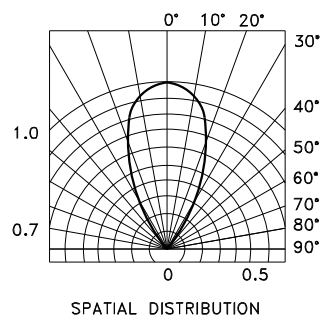
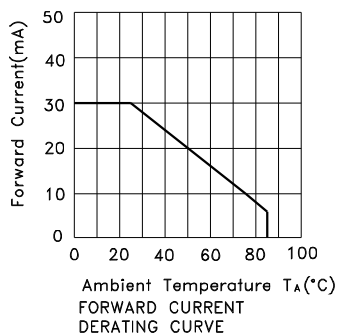
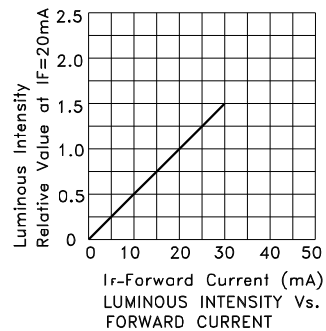
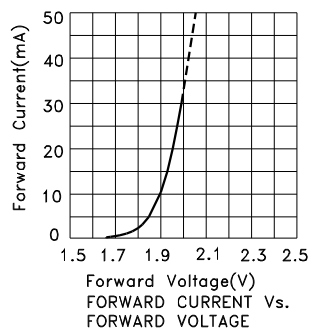
Parameter	Hyper Red	Blue	Green	Units
Power dissipation	75	120	102.5	mW
DC Forward Current	30	30	25	mA
Peak Forward Current [1]	185	150	150	mA
Reverse Voltage	5			V
Operating/Storage Temperature	-40°C To +85°C			
Lead Solder Temperature [2]	260°C For 3 Seconds			
Lead Solder Temperature [3]	260°C For 5 Seconds			

Notes:

1. 1/10 Duty Cycle, 0.1ms Pulse Width.
2. 2mm below package base.
3. 5mm below package base.

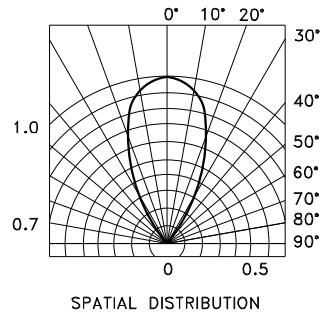
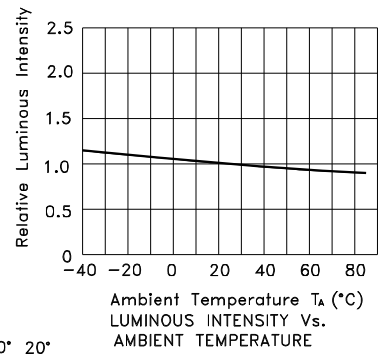
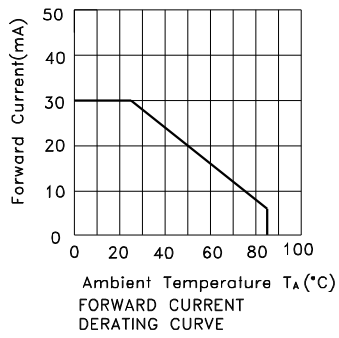
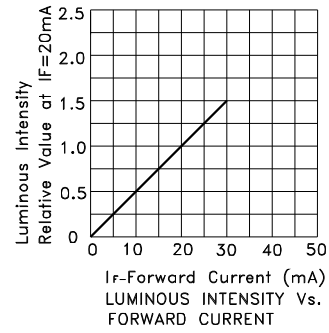
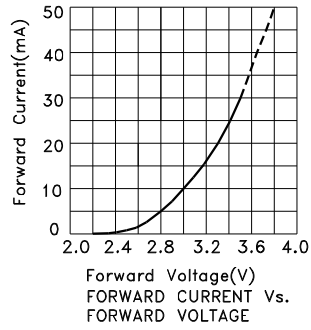


L-154A4SURKQBDZGC Hyper Red

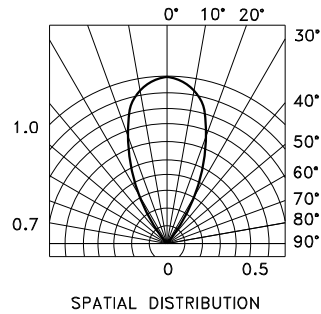
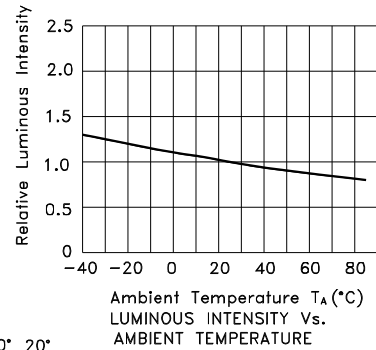
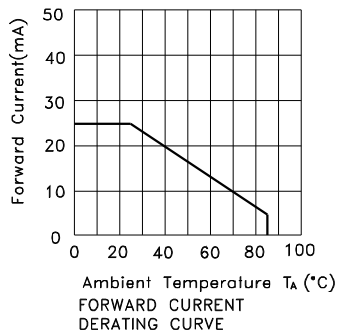
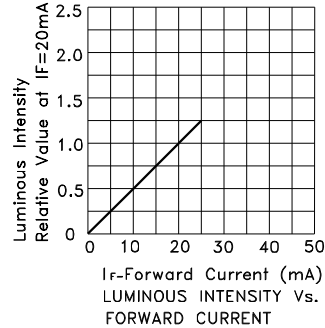
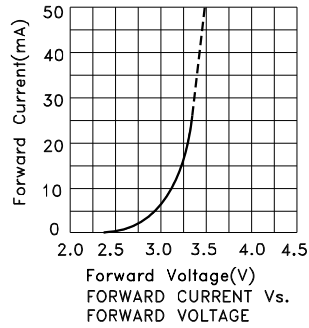


Kingbright

Blue

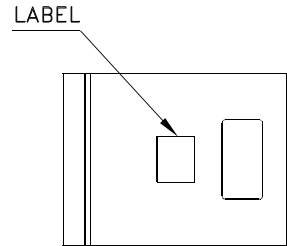
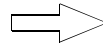
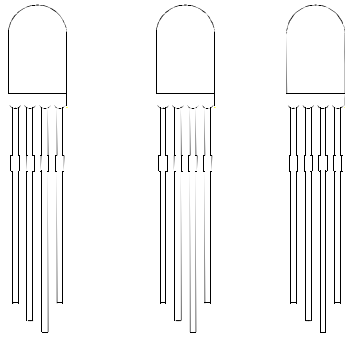


Green



PACKING & LABEL SPECIFICATIONS

L-154A4SURKQBDZGC

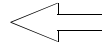


500PCS / BAG

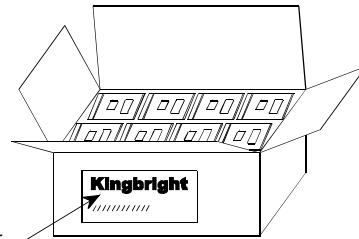


20K / 9# BOX


OUTSIDE LABEL



OUTSIDE LABEL

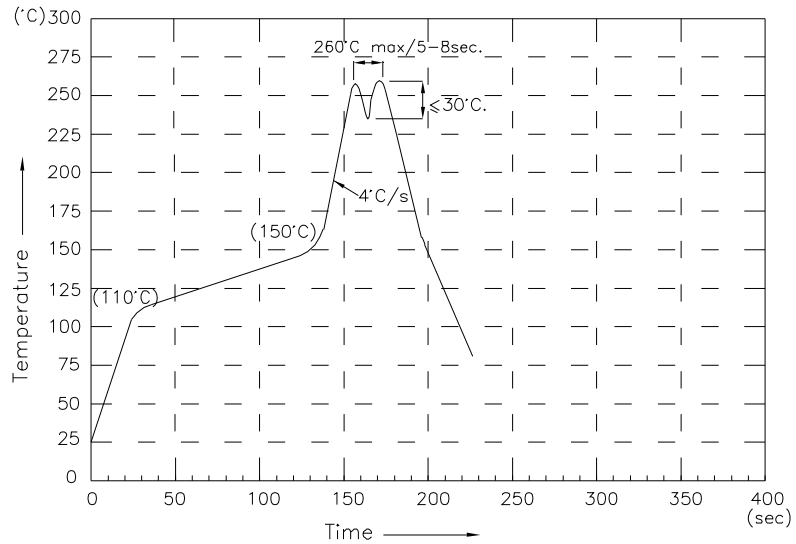


10K / 5# BOX

<h1>Kingbright</h1>	
P/NO: L-154A4xxx	
QTY: 500 pcs	Q.C. Q C XX XX XXXX PASSED
S/N: XXXX	
CODE: XXX	
LOT NO:	
 xxxxxxxxxxxxxxxxxxxxxxxx	
RoHS Compliant	

L-154A4SURKQBDZGC

Wave Soldering Profile For Lead-free Through-hole LED.



NOTES:

- 1.Recommend the wave temperature $245^{\circ}\text{C}\sim 260^{\circ}\text{C}$.The maximum soldering temperature should be less than 260°C .
- 2.Do not apply stress on epoxy resins when temperature is over 85 degree $^{\circ}\text{C}$.
- 3.The soldering profile apply to the lead free soldering (Sn/Cu/Ag alloy).
- 4.No more than once.

LED MOUNTING METHOD

1. The lead pitch of the LED must match the pitch of the mounting holes on the PCB during component placement. Lead-forming may be required to insure the lead pitch matches the hole pitch. Refer to the figure below for proper lead forming procedures.

(Fig. 1)

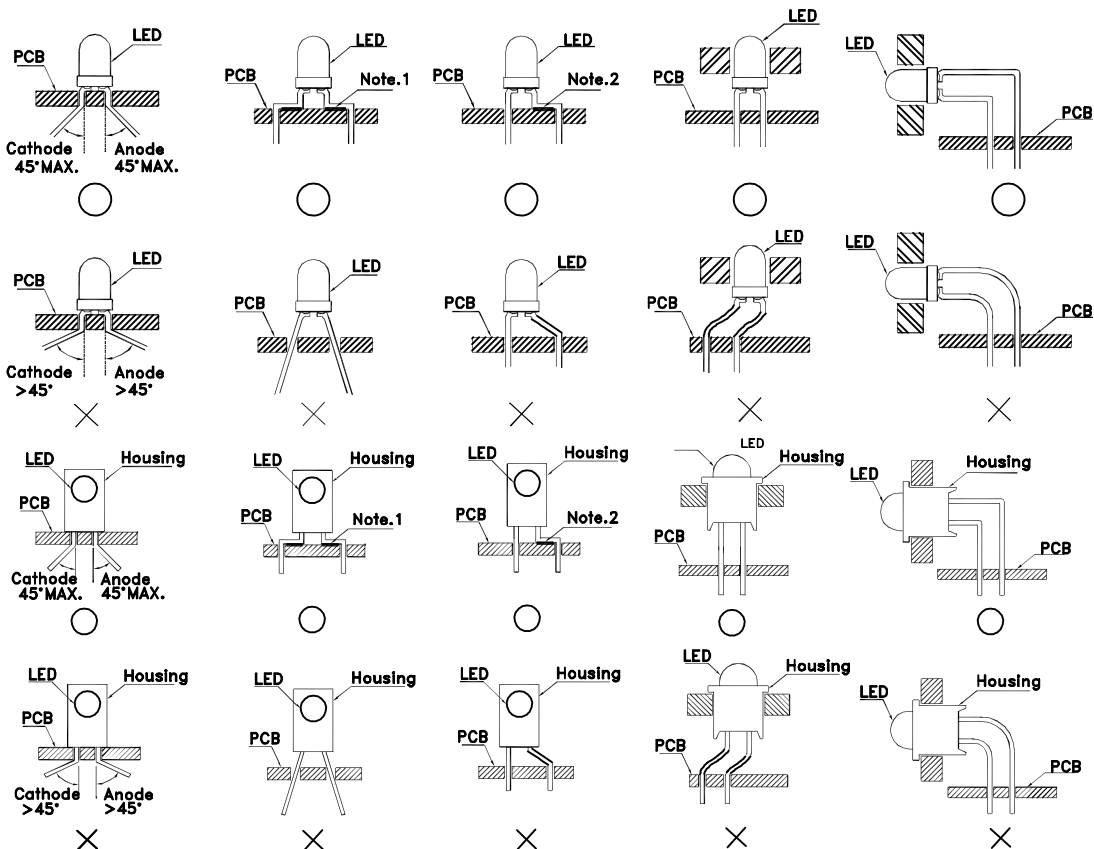


Fig.1

“O” Correct mounting method “X” Incorrect mounting method
 Note 1-2 : Do not route PCB trace in the contact area between the leadframe and the PCB to prevent short-circuits.

2. When soldering wire to the LED, use individual heat-shrink tubing to insulate the exposed leads to prevent accidental contact short-circuit.
 (Fig. 2)

3. Use stand-offs (Fig. 3) or spacers (Fig. 4) to securely position the LED above the PCB.

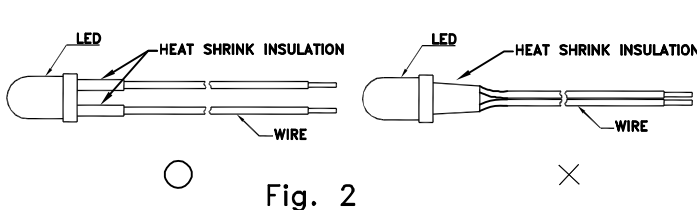


Fig. 2

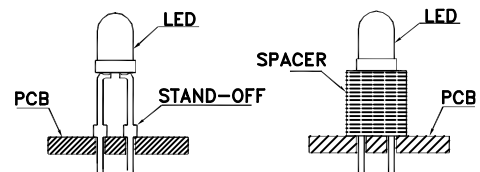


Fig. 3

Fig. 4

LEAD FORMING PROCEDURES

1. Maintain a minimum of 2mm clearance between the base of the LED lens and the first lead bend. (Fig. 5 and 6)

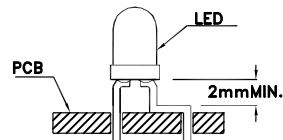


Fig. 5

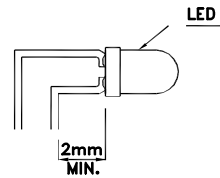


Fig. 6

2. Lead forming or bending must be performed before soldering, never during or after Soldering.
3. Do not stress the LED lens during lead-forming in order to fractures in the lens epoxy and damage the internal structures.
4. During lead forming, use tools or jigs to hold the leads securely so that the bending force will not be transmitted to the LED lens and its internal structures. Do not perform lead forming once the component has been mounted onto the PCB. (Fig. 7)
5. Do not bend the leads more than twice. (Fig. 8)
6. After soldering or other high-temperature assembly, allow the LED to cool down to 50°C before applying outside force (Fig. 9). In general, avoid placing excess force on the LED to avoid damage. For any questions please consult with Kingbright representative for proper handling procedures.

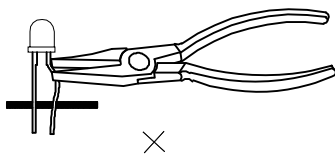


Fig. 7

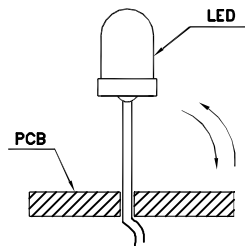


Fig. 8

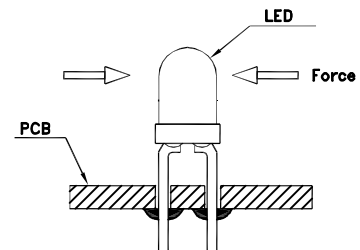


Fig. 9

7. No stress shall be applied on the LED during soldering to prevent damage.

