

**DISEÑO Y CONSTRUCCION DE UN TECLADO NUMERICO CON
ESTIMULOS VISUALES PARA INTERFACES CEREBRO-COMPUTADOR**

**STEPHANYA CASANOVA MARROQUIN
CARLOS EDUARDO DURAN MONTOYA**

**UNIVERSIDAD DEL VALLE
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA ELECTRICA Y ELECTRÓNICA
SANTIAGO DE CALI**

**2016DISEÑO Y CONSTRUCCION DE UN TECLADO NUMERICO CON
ESTIMULOS VISUALES PARA INTERFACES CEREBRO-COMPUTADOR**

**STEPHANYA CASANOVA MARROQUIN
CARLOS EDUARDO DURAN MONTOYA**

**PROYECTO DE GRADO PRESENTADO COMO REQUISITO PARCIAL PARA
OPTAR AL TÍTULO DE INGENIERO ELECTRONICO**

**DIRECTOR
EDUARDO CAICEDO BRAVO Ph.D**

**DIRECTOR
JAIBER EVELIO CARDONA ARISTIZABAL Ms.C**

**GRUPO DE INVESTIGACIÓN
PERCEPCIÓN Y SISTEMAS INTELIGENTES**

**UNIVERSIDAD DEL VALLE
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA ELECTRICA Y ELECTRÓNICA
SANTIAGO DE CALI**

2016

NOTA DE ACEPTACIÓN

Ing. EDUARDO CAICEDO BRAVO-**Director**

Ing. JAIBER CARDONA ARISTIZABAL-**Director**

Jurado 1

Jurado 2

Santiago de Cali, 22 de marzo de 2016

DEDICATORIA

“A mis padres y a mis hermanos que me han apoyado durante cada proceso.”

Stephanya C. M.

“Dedicado a María Eufemia, Luis Carlos y Andrés Mauricio por su apoyo y comprensión. A los amigos, compañeros y profesores que aportaron con sus enseñanzas a la finalización de este camino”

Carlos D. M.

AGRADECIMIENTOS

Eduardo Caicedo Bravo, Ingeniero Electricista, profesor de la Escuela de Ingeniería Eléctrica y Electrónica en la Universidad del Valle. Director del proyecto. Por su acompañamiento en el proceso y por la gran oportunidad que nos dio con el proyecto.

Jaiber Evelio Cardona, Ingeniero Electrónico, por su apoyo incondicional y sus acertadas recomendaciones en el desarrollo del proyecto.

Al Grupo de Percepción y Sistemas Inteligentes por buscar expandir los conocimientos de su área.

A nuestros profesores y compañeros de trabajo en la Universidad del Valle que hicieron posible la confección y elaboración de esta tesis.

RESUMEN

En este documento se consignan los pasos para el diseño y construcción de un dispositivo alfa-numérico con estímulos visuales para interfaces cerebro computador (ICC). Inicialmente se presenta el marco teórico referente a las señales SSVEP, luego se hace una revisión bibliográfica de los trabajos a nivel nacional e internacional en donde exponen las características de estos sistemas de estimulación y posterior a esto, se hace una recopilación de las condiciones de diseño que se deben considerar a la hora del desarrollo.

En la segunda parte del documento se presenta el diseño e implementación del software, se comienza con la descripción del desarrollo del software, en donde se exponen los requerimientos funcionales y no funcionales de este, junto con el patrón de arquitectura aplicado. Se presentan esquemas de relación lógicos con los cuales el lector puede comprender la interacción de los módulos desarrollados. Seguido se expone el diseño y desarrollo del hardware y del prototipo final, explicando los algoritmos de programación y la escogencia de materiales y dimensiones.

En el tercer capítulo se presentan las pruebas y resultados obtenidos, aquí se describe la metodología desarrollada para la realización de las pruebas, es decir, el protocolo de pruebas en donde se establecen las condiciones ambientales y técnicas para una adecuada estimulación y obtención de resultados.

Finalmente, se presentan los resultados obtenidos y se concluye.

Palabras claves: Potenciales de estado estable evocados visualmente (SSVEP), Generación de Señales SSVEP, Interfaces cerebro computador (ICC), Teclado Numérico con Estímulos Visuales.

ABSTRACT

This document recorded the steps for the design and construction of a keypad with visual stimuli for brain computer interfaces. Initially is presented the theoretical framework concerning to SSVEP signals, then a literature review of work at national and international level where exposed the characteristics of these systems stimulation and after this, is done a collection of design conditions to consider at the time of development.

In the second part of the document the design and implementation of the software is presented. It begins with the description of software development, where the functional and nonfunctional requirements of this along with the architectural pattern applied are exposed, schemes logical connection with which the reader can understand the interaction of the developed modules are presented. Followed the design and development of hardware and final prototype is exposed, explaining programming algorithms and the selection of materials and dimensions.

In the third chapter the tests and results are presented, here the methodology developed for conducting the tests, that is, the test protocol where environmental and technical conditions for proper stimulation and getting results are set is described.

Finally, the results obtained are presented and concludes.

Keywords: Steady state visually evoked potential (SSVEP), Signal Generation SSVEP, brain computer interfaces (BCI), Numeric keypad with Visual stimuli.

CONTENIDO

	Pág.
INTRODUCCION	14
Problemática	16
Objetivos	17
Descripción del Trabajo	18
 1. MARCO TEORICO	 19
1.1 Introducción	19
1.2 Interfaz Cerebro-Computador (ICC)	19
1.2.1 Tipo de Respuestas Cerebrales empleadas en ICC	20
1.2.1.1 P300	20
1.2.1.2 SSVEP	20
1.2.1.3 ERD/ERS	21
1.2.2 Sistemas Estimuladores de SSVEP	21
1.2.2.1 Condiciones de diseño	24
1.2.2.2 Antecedentes Internacionales	24
1.2.2.3 Antecedentes Nacionales	25
1.2.3 Sistemas Basados en ICC	26
1.2.3.1 Antecedentes Internacionales	26
1.2.3.2 Antecedentes Nacionales	27
1.3 Conclusiones	27
 2. DISEÑO E IMPLEMENTACIÓN	 29
2.1 Módulo de Software: Interfaz de Usuario QFOCUS	30
2.1.1 Análisis de Requerimientos	30
2.1.2 Diseño y Arquitectura	33
2.1.3 Codificación	37
2.1.4 Pruebas y Resultados	40
2.2. Módulo de Hardware: Tarjeta de desarrollo y de Leds	42
2.2.1 Características Funcionales	42
2.2.2 Diseño	43

2.2.2.1 Tarjeta de Leds	43
2.2.2.2 Tarjeta de desarrollo	44
2.2.2.3 Resultados	45
2.2.3 Programación del Microcontrolador	46
2.2.3.1 Métodos	48
2.2.3.2 Pruebas y Resultados	50
2.3 Prototipo Físico del Teclado	53
2.3.1 Diseño del prototipo	54
2.3.2 Resultados	54
3. PRUEBAS Y RESULTADOS	56
3.1 Aplicación Cerebro-Computador	56
3.1.1 EMOTIV	56
3.1.2 Protocolo de Pruebas	57
3.1.3 Estímulos y Resultados	59
4. CONCLUSIONES	71
5. TRABAJOS FUTUROS	73
6. BIBLIOGRAFÍA	74

LISTA DE TABLAS

Tabla 1. Organización espacial de cada led en el diseño de la tarjeta	39
Tabla 2. Estímulos usados en las pruebas del teclado numérico y teclas externas	59
Tabla 3. Estimulación de 15.15Hz con onda de doble flanco ascendente	60
Tabla 4. Estimulación a 26.32Hz con doble flanco ascendente	60
Tabla 5. Estimulación de 11.90Hz con onda de doble flanco ascendente	60
Tabla 6. Estimulación a 23.81Hz con doble flanco ascendente	61
Tabla 7. Estimulación a 14.45Hz con triple flanco ascendente	61
Tabla 8. Estimulación a 14.04Hz con triple flanco ascendente	62
Tabla 9. Estimulación de 10Hz con onda de doble flanco ascendente	63
Tabla 10. Estimulación a 15.82Hz con triple flanco ascendente	63
Tabla 11. Estimulación de 7.93Hz con onda de doble flanco ascendente	64
Tabla 12. Estimulación a 18.52Hz con doble flanco ascendente	65
Tabla 13. Estimulación a 29.41Hz con doble flanco ascendente	65
Tabla 14. Estimulación a 27.78Hz con doble flanco ascendente	66
Tabla 15. Estimulación de 15.15Hz con onda cuadrada	67
Tabla 16. Estimulación de 11.90Hz con onda cuadrada	67
Tabla 17. Estimulación de 10Hz con onda cuadrada	68
Tabla 18. Estimulación de 7.93Hz con onda cuadrada	69

LISTA DE FIGURAS

Figura 1.	Efecto de la estimulación visual a diferentes frecuencias del SSVEP [@ Pastor, 2008]	22
Figura 2.	Sistema desarrollado: Módulo Software, Módulo Hardware	30
Figura 3.	Secuencia con desplazamiento de 0 ms y la misma secuencia con desplazamiento 5ms.	31
Figura 4.	Diagrama de Casos de Uso	33
Figura 5.	Diagrama de Flujo para la actividad de Procesamiento de Secuencias	34
Figura 6.	Diagrama de Flujo para la Actividad de Programación de Secuencias en el Teclado	35
Figura 7.	Patrón de Arquitectura MVC para la aplicación	36
Figura 8.	Diagrama de Paquetes	38
Figura 9.	Diagrama de Clases	39
Figura 10.	Interfaz QFocus 1.0	40
Figura 11.	Lectura de Fichero	41
Figura 12.	Modificación de Secuencias (s01, s03, s04, s06, s11, sA3, sA4) sobre la Interfaz de QFocus 1.0	41
Figura 13.	Comunicación Serial y Programación del micro	41
Figura 14.	Almacenamiento de Secuencias	41
Figura 15.	Tarjeta de leds	42
Figura 16.	Diseño general de la conexión de cada led	44
Figura 17.	Diseño de la tarjeta de desarrollo	45
Figura 18.	Tarjeta de desarrollo y Matriz de leds	45
Figura 19.	Formato general de tramas	46
Figura 20.	Formato de trama START	47
Figura 21.	Formato de trama LOAD	47

Figura 22.	Formato de trama STOP	47
Figura 23.	Diagrama de flujo del algoritmo del microcontrolador	48
Figura 24.	Jerarquía de métodos en la programación del microcontrolador	50
Figura 25.	Verificación de tiempos de secuencia	51
Figura 26.	Verificación de cumplimiento de retardos	51
Figura 27.	Medición de tiempo del cambio de estado (alto a bajo y bajo a alto)	52
Figura 28.	Tiempo real de cambio de estado igual a 276 us	52
Figura 29.	Máximo retardo entre la primera y última secuencia	53
Figura 30.	Vistas del diseño del teclado numérico y teclas externas	54
Figura 31.	Prototipo de teclado numérico y teclas externas	55
Figura 32.	EMOTIV Epoc Model 1.0	56
Figura 33.	Configuración del EMOTIV Epoc para las pruebas	57
Figura 34.	Voluntario con el EMOTIV Epoc con un estímulo visual de 15Hz	58
Figura 35.	Estímulo a 15.15Hz de doble flanco y respuesta a 15.15Hz	59
Figura 36.	Estímulo a 26.32Hz a doble flanco ascendente y respuesta a 26.32Hz	59
Figura 37.	Estímulo a 11.90Hz de doble flanco y respuesta a 11.90Hz	60
Figura 38.	Estímulo a 23.81Hz a doble flanco ascendente y respuesta a 23.81Hz	60
Figura 39.	Estímulo a 14.45Hz a triple flanco ascendente y respuesta a 14.45Hz	61
Figura 40.	Estímulo a 14.04Hz a triple flanco ascendente y respuesta a 14.04Hz	62
Figura 41.	Estímulo a 10Hz de doble flanco y respuesta a 10.10Hz	62

Figura 42.	Estímulo a 15.82Hz a triple flanco ascendente y respuesta a 15.82Hz	63
Figura 43.	Estímulo a 7.93Hz de doble flanco y respuesta a 7.93Hz	64
Figura 44.	Estímulo a 18.52Hz a doble flanco ascendente y respuesta a 18.52Hz	64
Figura 45.	Estímulo a 29.41Hz a doble flanco ascendente y respuesta a 29.41Hz	65
Figura 46.	Estímulo a 27.78Hz a doble flanco ascendente y respuesta a 27.78Hz	66
Figura 47.	Estímulo a 15.15Hz de onda cuadrada y respuesta a 15.15Hz	66
Figura 48.	Estímulo a 11.90Hz de onda cuadrada y respuesta a 11.90Hz	67
Figura 49.	Estímulo a 10Hz de onda cuadrada y respuesta a 9.99Hz	68
Figura 50.	Estímulo a 7.93Hz de onda cuadrada y respuesta a 7.93Hz	68

INTRODUCCION

La Ingeniería se define como el conjunto de conocimientos y técnicas científicas aplicadas a la creación, mejoramiento e implementación de estructuras (tanto físicas como teóricas) para la resolución de problemas. Estas problemáticas van desde la construcción de un satélite artificial hasta el desarrollo de sistemas encargados del mejoramiento de la calidad de vida de personas que sufren de movilidad disminuida. En esta última se ha generado un gran interés en el mundo académico, ya que permite la integración de diversas áreas con las que antes no se trabajaba, tales como la biología, la neurología y la Medicina, además que los resultados propuestos generan un gran beneficio para un sinnúmero de personas y abren las puertas para muchas otras investigaciones que no solo se aplican en este tipo de problemas, si no que se extienden a infinitas áreas.

Las personas que sufren de movilidad disminuida presentan dificultades de accesibilidad a tareas comunes, lo que les genera un grado de dependencia a equipos u otras personas. Entre los sistemas que buscan el mejoramiento de la calidad de vida de estas personas existen algunos que se basan en ICC. Esta tecnología permite, por ejemplo, desde controlar el movimiento de una silla de ruedas o un robot inalámbrico hasta controlar diferentes dispositivos electrónicos presentes en cualquier vivienda, tales como un celular o una alarma. Esta técnica inicialmente se creó como una solución para la comunicación con personas que tienen daños neuromusculares severos, como los producidos por esclerosis lateral amiotrófica, lesiones cerebrales o lesiones de la médula espinal [Wolpaw et al, 2002]; sin embargo, puede extenderse a otras necesidades. Las ICC captan señales del cerebro, las analizan y las traducen en comandos, transmitidos a su vez a dispositivos de salida que ejecutan las acciones deseadas.

Actualmente se trabaja con 3 tipos de señales que por su simplicidad y fácil adquisición son capturadas: ERD/ERS (*event-related desynchronization ERD and event-related synchronization ERS*), P300 y SSVEP (*Steady-States Visually Evoked Potentials*) [Pfurtscheller et al, 2010].

En el marco de un proyecto, que tiene como objetivo general realizar un aporte a la autonomía de personas con discapacidad severa usando ondas cerebrales para el control de una plataforma robótica, se requiere desarrollar un sistema programable capaz de estimular señales de tipo SSVEP, las cuales serán empleadas para el manejo de la plataforma. Este sistema programable que estimulará las señales SSVEP debe ser diseñado ergonómicamente de manera que no genere ningún efecto negativo sobre el usuario, y además su estructura debe ser fácilmente reparable.

PROBLEMÁTICA

El uso de sistemas computacionales es una actividad que realizan las personas de manera cotidiana, esta actividad involucra los sentidos del tacto, la vista y la audición. Las personas que presentan una deficiencia física, mental, intelectual o sensorial, la cual afecta la forma de interactuar y participar plenamente con igualdad en la sociedad se consideran personas en estado de discapacidad.

Las interfaces tradicionales de interacción con el computador, como el ratón o el teclado, dificultan a las personas en estado de discapacidad motora el uso de sistemas de computación. Existe un grupo de recursos tecnológicos de apoyo para facilitar la interacción de las personas en estado de discapacidad con el computador y otros sistemas a los que normalmente les es poco práctico o imposible.

Entre esos recursos, las interfaces cerebro-computador (ICC) sirven para interpretar ondas cerebrales y permiten transformarlas en acciones reales en nuestro entorno. En este caso, las acciones están enfocadas a que las personas en estado de discapacidad, con ciertas funcionalidades cerebrales, puedan usar sistemas de computación de una manera más accesible.

Existen unas señales que se producen en el cerebro al presentar un estímulo visual oscilando a cierta frecuencia, estas ondas se pueden encontrar en un electroencefalograma en un aumento de la amplitud de las componentes de la frecuencia del estímulo y sus armónicos. Estas señales se conocen como potenciales de estado estable evocados visualmente (SSVEP). Una vez identificada la señal SSVEP, la acción posterior tiene un amplio campo de aplicación, enfocados particularmente en este caso a la interacción con un sistema de cómputo.

¿Es posible tener un dispositivo entrada/salida externo que genere las ondas SSVEP e integrarlo a una ICC para facilitar la interacción con el computador para un sector de la población discapacitada? Responder este interrogante es el objetivo general de este documento.

OBJETIVOS

Objetivo General

Diseñar y construir un teclado numérico con estímulos visuales y su software de programación, para la estimulación de potenciales de estado estable evocados visualmente (SSVEP) en interfaces cerebro-computador.

Objetivos Específicos

- Identificar un sistema para la estimulación de SSVEP para su posterior desarrollo en hardware y software.
- Construir un teclado numérico con estímulos visuales programables para la estimulación de SSVEP.
- Desarrollar un software para la programación de los estímulos visuales del teclado.
- Validar el sistema generador de señales SSVEP mediante una aplicación cerebro-computador.

DESCRIPCIÓN DEL TRABAJO

Es requerido un sistema *interfaz-teclado*, capaz de generar estímulos visuales a diferentes frecuencias con la finalidad de buscar la generación de SSVEP. Las especificaciones de cada módulo se indican a continuación.

Módulo Hardware: Prototipo en forma de teclado compuesto de 16 teclas, de las cuales 12 tienen una ubicación fija en el teclado, mientras que las 4 teclas restantes son de ubicación libre, es decir, el usuario puede ubicarlas en diferentes partes de forma independiente del teclado. Cada tecla tiene asociado un *led* con el cual se genera la estimulación visual según la secuencia programada.

Se define como *Secuencia* al conjunto de 13 valores con los cuales se establece el conjunto de tiempos y parámetros con los que se estimula visualmente al usuario, estos valores son ingresados por el usuario encargado de programar el teclado. A continuación se explican los valores de las secuencias:

-*Estado Inicial:* Corresponde al estado inicial del led, es decir, si comienza en encendido o apagado.

-*Retardo:* Tiempo de inicio sobre la secuencia.

-*Des/Habilitado:* Indica si la secuencia va a estar activa o no, si no está activa, el led está apagado.

-*10 tiempos:* Cada uno entre 1 y 77 ms con los cuales se obtienen frecuencias entre 6,49 y 500,0 Hz.

Módulo Software: Interfaz de Usuario con la cual el usuario puede leer, visualizar, modificar y descargar las secuencias al teclado.

El mantenimiento y la usabilidad son también consideraciones de diseño. En el mantenimiento se considera que el prototipo debe ser de fácil reparación y modificación para el usuario final. En la usabilidad se considera que el software de programación debe ser de fácil entendimiento y uso, este requerimiento se apoya en el desarrollo de una guía de usuario en donde se indica de forma sencilla el manejo del software.

1. MARCO TEORICO

1.1 INTRODUCCIÓN

En este capítulo se presentan los conceptos teóricos necesarios para el desarrollo del SISTEMA ESTIMULADOR DE SEÑALES SSVEP. Inicialmente se habla sobre la Interfaz Cerebro-Computador, su definición, las etapas de desarrollo, los alcances actuales. Seguido, se describen los tipos de señales de respuesta que sirven para el análisis, las diferentes estrategias que se emplean para estimular las ondas cerebrales, se hace un cuadro comparativo entre éstas, y se concluye. Finalmente, se exponen algunos sistemas que emplean ICC para el control de un objetivo.

1.2 INTERFAZ CEREBRO-COMPUTADOR (ICC)

Los sistemas Interfaz Cerebro-Computador (ICC) permiten comunicar de manera directa el cerebro con el computador a través de la medición de la actividad cerebral. El Electroencefalograma (EEG) es el más usado para medir la actividad cerebral en este tipo de sistemas por su economía, portabilidad y respuesta en tiempo.

Los sistemas ICC han sido considerados como una posibilidad de comunicación para usuarios con daños neuromusculares severos como los producidos por esclerosis lateral amiotrófica, lesiones cerebrales o lesiones de la médula espinal; en este sentido, se han creado aplicaciones para escribir palabras en un ordenador, navegar en internet y realizar correos electrónicos, al igual que sistemas para el movimiento de sillas de ruedas, para el movimiento de órtesis mecánicas o en la aplicación de neuroprótesis. Así mismo, se ha trabajado en las respuestas auditivas del ICC para personas con déficit en el movimiento ocular y para la rehabilitación de personas con accidente cerebrovascular.

Además, los sistemas ICC se han empleado en experimentos dirigidos a comprender el comportamiento del cerebro; por ejemplo, en el procesamiento subconsciente del ruido o del parpadeo de luces aparentemente imperceptibles a simple vista, en la predicción de la codificación exitosa de la memoria, y en la medición del nivel de atención.

Los sistemas ICC han permitido también el estudio de diferentes patologías como la transición del estado de enclaustramiento (LIS) y el estado de enclaustramiento total (TLIS), además de promisorios resultados en diagnósticos y tratamientos médicos como la autorregulación del SCP (*Slow Cortical Potential*) en niños con déficit de atención o hiperactividad, y el aprendizaje y desarrollo de la comunicación social en individuos autistas.

1.2.1 Tipo de Respuestas Cerebrales empleadas en ICC

Las ICC modernos, no invasivos, basados en EEG, cuentan con uno de tres tipos de señales: ERD/ERS (*event-related desynchronization ERD and event-related synchronization ERS*), P300 y SSVEP (*Steady-States Visually Evoked Potentials*).

1.2.1.1 P300

El P300 se produce como una respuesta natural e involuntaria a un estímulo externo visual, auditivo o táctil, que llama la atención por su infrecuencia o expectativa, en el cual se presenta un pico positivo cerca de los 300 ms después del estímulo y se ubica en señales EEG del córtex centroparietal. El P300 ha demostrado ser bastante estable en usuarios con enclaustramiento, apareciendo incluso después de lesiones del tronco cerebral.

1.2.1.2 SSVEP

Los potenciales de estado estacionario inducidos visualmente, o *Steady-States Visually Evoked Potentials* (SSVEP), constituyen un método síncrono que corresponde a la respuesta del córtex visual debido a la estimulación de la retina por una fuente de luz parpadeante. La amplitud del espectro del SSVEP depende de la frecuencia de la luz parpadeante; así, teniendo un arreglo de leds con luces de diferente frecuencia, el usuario puede enfocar su atención sobre el led que indique el comando que quiere

realizar. Debido a que es una respuesta natural del cerebro, este no requiere ningún tipo de entrenamiento. Las respuestas típicas son del orden de los segundos.

1.2.1.3 ERD/ERS

La sincronización y desincronización relacionada a eventos (*event-related desynchronization* ERD and *event-related synchronization* ERS) hacen referencia respectivamente al decrecimiento de las señales asociadas al movimiento y preparación para el movimiento, y al incremento producido después de un movimiento o con la relajación. Estas señales representan cambios específicos en la frecuencia por la disminución o el aumento de la energía relacionada con una disminución o aumento de la sincronización de las poblaciones neuronales subyacentes. Como característica especial pertinente para los sistemas BCI, las señales ERD y ERS pueden ser generadas con solamente la imaginación del movimiento (*motor imagery*).

1.2.2 Sistemas Estimuladores de SSVEP

Estos sistemas son los encargados de estimular visualmente las ondas cerebrales que serán procesadas y clasificadas para efectuar alguna acción deseada, por lo cual, influyen directamente en el rendimiento de un ICC basado en SSVEP. Un mal diseño de un sistema afectará la comodidad del usuario y la confianza de las señales de respuesta obtenidas.

En [Danhua, 2010] se hace una revisión de 57 artículos relacionados con ICC basadas en SSVEP en donde se identifican parámetros de diseño, tales como: *Tipo del estímulo* (luz, imagen, patrón invertido), *fuentes del estímulo repetitivo* (leds, LCD, RCL), *rango de frecuencias* de trabajo (bajas [1-12], medias [12-30] y altas [30-60]), *color del estímulo* (blanco, rojo, verde y azul) y otras para los sistemas estimuladores de señales SSVEP. El rendimiento es medido en bits/s. En este artículo se concluye que todos los estímulos visuales repetitivos tienen diversas propiedades tales como: frecuencia, color y contraste, y que estas afectan a la respuesta SSVEP.

A continuación se presentan las investigaciones que se han hecho con relación a las consideraciones de diseño: *fente de estímulo, color de estímulo, rango de frecuencias de trabajo y patrón de contraste.*

-Con relación a la fuente de estímulo (LEDs, LCD, CRT)

Stimulator selection in SSVEP-based BCI [Zhenghua, 2001] En esta publicación se investigaron las diferencias de espectro de los tres tipos de destellos y las diferencias en las SSVEPs generadas por tres fuentes de estimulación diferentes: LEDs, un monitor de CRT y una pantalla LCD. Los resultados mostraron que las diferencias SSVEP estaban fuertemente relacionados con las diferencias de espectro de frecuencia de los destellos. De acuerdo con estas diferencias, se seleccionó la pantalla de LEDs como aquella que entregaba los mejores resultados.

-Con relación al color del estímulo (blanco, verde, rojo, azul, otros)

Flashing Color on the Performance of SSVEP-based Brain-Computer Interfaces [Teng, 2012] En esta investigación se propuso identificar cuál era el color que genera menos fatiga visual y a la vez determinar cuál es el que presenta los coeficientes de CCA (*análisis de correlación canónica*) más altos. El CCA es un método estadístico multivariable que permite extraer las características frecuenciales de las señales SSVEP. Aquí se concluyó que el color blanco presenta los CCA más altos y en cuanto al color rojo, se encontró que es altamente variable con las frecuencias utilizadas, a diferencia del color verde.

-Con relación a las frecuencias utilizadas

Human Cerebral Activation During Steady-State Visual-Evoked Responses [Pastor, 2003] En esta publicación se busca relacionar qué área de la corteza visual es activada por un determinado rango de frecuencias. El rango de trabajo fue de [2 - 90] Hz. Además en este informe se indica que en 15Hz se obtienen los picos más altos de las señales SSVEP, y que esta amplitud disminuye a medida que aumenta la frecuencia. En la figura 1 se observa esta relación.

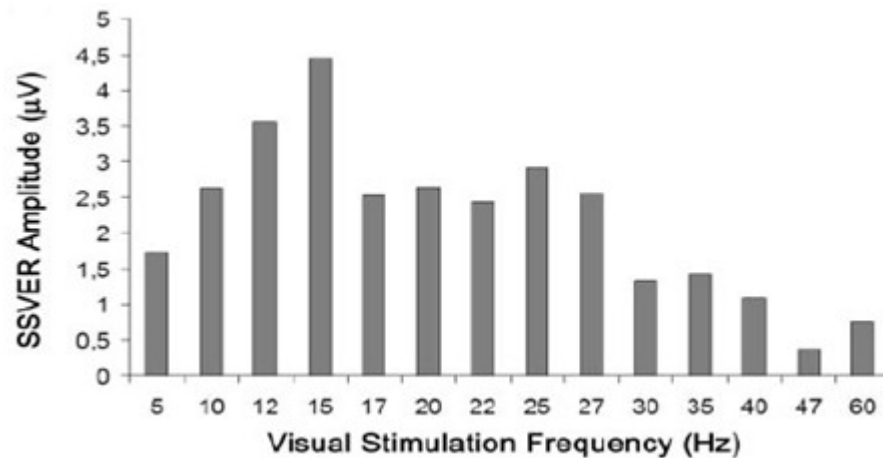


Figura 1. Efecto de la estimulación visual a diferentes frecuencias del SSVEP
[(@ Pastor, 2008)]

-Con relación al patrón empleado (imagen plana, imagen con 2 patrones inversos)

Comparison of Plain and Checkerboard Stimuli for Brain Computer Interfaces Based on Steady State Visual Evoked Potentials [Rosanne,2013] Este estudio compara la sensibilidad en el desempeño de una ICC basado en SSVEP para los estímulos planos y aquellos donde se utiliza un estímulo compuesto por 2 patrones inversos como un tablero de ajedrez. Los resultados mostraron una respuesta más fuerte en estímulos normales que aquellos que tenían el patrón del ajedrez, este resultado fue independiente de la frecuencia.

En cuanto a la distancia entre cada estímulo, la inclinación del estímulo y la diferencia mínima que debe haber entre cada frecuencia, no se encontró ninguna publicación con un estudio real de cómo se deben determinar los valores apropiados. Estos fueron puestos de manera empírica, es decir, se probaron distintos valores y se eligieron los que entregan mejores resultados. En [Han-Jeong, 2012] y [Xiaorong, 2003] la diferencia entre las frecuencias es de 0,1 y 0,2 Hz. En [Resalat, 2012] se determinó una distancia de 1 cm entre cada estímulo y los ángulos fueron de 30° y 40°, aquí también se concluyó que el ángulo no influía en la rendimiento del sistema ICC basado en SSVEP.

En la siguiente sección se describen las condiciones mínimas que se deben considerar al desarrollar un Sistema Estimulador de Señales SSVEP.

1.2.2.1 Condiciones de diseño

Entre los requerimientos que se buscan alcanzar al desarrollar un Sistema Estimulador de SSVEP están:

1. Comodidad del Usuario
2. Fácil Integración y Reparación
3. Las señales producidas presenten un error relativo porcentual inferior al 5%.

A continuación se presentan las condiciones que deben evitarse durante el desarrollo de un sistema ICC

1. *Fatiga Visual*: la cual además de producirse por trabajar con frecuencias en el rango de [5 – 25] Hz, también se genera por el color de la luz emitida [Han-Jeong, 2012] en aquellos sistemas que emplean luces como estímulos.
2. *Epilepsia*: se debe evitar de igual forma trabajar con frecuencias en el rango [15 – 25] Hz [3].
3. *Falsos Positivos*: respuestas que no corresponden a estímulos visuales, se generan en las frecuencias [8 - 13] Hz [4].

En cuanto a la construcción del sistema se busca que:

- a. las respuestas sean coherentes con los estímulos generados y no representen una composición de varias frecuencias, por lo cual, se debe considerar el espaciado entre cada estímulo para que no se interfieran entre ellos [Resalat, 2012];
- b. la inclinación y la distancia del usuario con los estímulos sea la apropiada para obtener una adecuada señal de respuesta;
- c. la reparación y el mantenimiento del sistema sean realizables de manera cómoda por el usuario;
- d. el sistema debe poderse integrar fácilmente a cualquier ICC basado en SSVEP.

A continuación se exponen trabajos producidos a nivel internacional y nacional, en donde se desarrollan sistemas generadores de estímulos SSVEP.

1.2.2.2 Antecedentes Internacionales

SSVEP-BCI Implementation for 37-40 Hz Frequency Range [Torres, 2011] El sistema que se presenta en este trabajo es muy eficiente en la detección de SSVEP en el rango de alta frecuencia (> 30 Hz). El uso de las frecuencias más altas de estímulo lleva a una estimulación más cómoda. Los mejores resultados se encontraron usando seis electrodos, pero los buenos resultados se pueden encontrar usando sólo tres canales occipitales durante la etapa de procesamiento. Este sistema automático sería útil para implementar un SSVEP - BCI en línea. Tal BCI se aplicaría para comandar un robot o incluso una silla de ruedas robótica, por ejemplo.

Development Of An SSVEP-Based BCI Spelling System Adopting A QWERTY-Style LED Keyboard [Han-Jeong, 2012] En este estudio, se introduce un nuevo sistema ortográfico mental basada en SSVEP; se adopta un teclado estilo QWERTY con 30 LEDs que parpadean a diferentes frecuencias para generar los estímulos. A través de experimentos preliminares *off-line* y *on-line*, se confirmó que SSVEPs provocados por estímulos visuales parpadeantes con una frecuencia resolución de 0,1 Hz se podrían clasificar con una alta precisión. Durante los experimentos preliminares realizados *off-line* con cinco participantes se optimizaron diversos factores que influyen en el rendimiento del sistema de ortografía mental, tales como las distancias entre las teclas adyacentes, arreglos de fuente de luz, frecuencias de estimulación, grabación de electrodos, y los ángulos visuales. Los experimentos *on-line* se llevaron a cabo con seis participantes y ayudaron a verificar la viabilidad del sistema de ortografía mental optimizado. Los resultados de los experimentos *on-line* eran un promedio de velocidad de mecanografía de 9,39 letras por minuto (lpm), con una tasa de éxito promedio de 87,58 %, que corresponde a una tasa promedio de transferencia de información de 40,72 bits por minuto, lo que demuestra el alto rendimiento del sistema de ortografía mentales desarrollado. De hecho, la velocidad media de tipificación de 9,39 lpm alcanzado en este estudio fue uno de los mejores resultados lpm entre los reportados en la literatura ICC anteriores.

A BCI-based environmental controller for the motion-disabled [Xiaorong, 2003] Este artículo presenta un controlador usando una técnica ICC basado en SSVEP. El sistema se compone de un estimulador, un procesador de señal digital, y un controlador remoto por infrarrojo. Las características atractivas de este sistema incluyen la grabación de la señal no invasiva y una alta tasa de transferencia de información. Los resultados de las pruebas han demostrado que este sistema puede distinguir al menos 48 objetivos y proporcionar una tasa de transferencia de hasta 68 b / min. El sistema se ha aplicado al control de un aparato eléctrico con éxito.

1.2.2.3 Antecedentes Nacionales

No se encontraron artículos a nivel nacional en donde se estudiaran o propusieran características o parámetros para diseñar un Sistema Estimulador de SSVEP.

1.2.3 SISTEMAS BASADOS EN ICC

A continuación se exponen trabajos producidos a nivel internacional y nacional, en donde se desarrollan sistemas que hacen uso de interfaces cerebro-computador (ICC).

1.2.3.1 Antecedentes Internacionales

An SSVEP BCI to Control a Hand Orthosis for Persons With Tetraplegia [Rupert, 2011] En este estudio, se presenta un sistema ICC para el control de una prótesis. Para el experimento se emplearon 7 sujetos sin entrenamiento previo. En los resultados se obtuvo que, seis sujetos tuvieron un buen control con un valor predictivo positivo (VPP) superior al 60%. El VPP general para todos los sujetos alcanzó el $78\% \pm 10\%$. Sin embargo, la tasa de falsos positivos fue alta, y algunos sujetos se sintieron incómodos con las luces parpadeantes. Para trabajos futuros, se espera reducir tanto la tasa de falsos positivos como las molestias producidas por las luces parpadeantes. Se espera también validar el prototipo con personas con tetraplejia.

Using a SSVEP-BCI to Command a Robotic Wheelchair [Torres, 2011] Este trabajo presenta una interfaz cerebro-ordenador (ICC) con base en los potenciales evocados de

estado estable (SSVEP). En el experimento, los voluntarios obtuvieron un éxito del 60% al 100% en la operación del ICC. Sumado a esto, los voluntarios lograron controlar una silla de ruedas a través de un ambiente exterior. Una característica adicional en el proyecto fue que el ICC incorpora una retroalimentación visual, lo que es esencial para mejorar el rendimiento de todo el sistema.

Developing Stimulus Presentation on Mobile Devices for a Truly Portable SSVEP-based BCI [Yu TE, 2013] Este estudio integra la generación del estímulo visual con el procesamiento de datos en tiempo real en un dispositivo móvil (por ejemplo, una tableta o un teléfono celular). El objetivo de este estudio es aumentar la viabilidad, la portabilidad y la ubicuidad de un ICC basados en SSVEP para el uso diario. Este estudio analizó la densidad del espectro de potencia de las señales provocadas por los estímulos visuales presentados en las ICCs móviles. Por último, este estudio realizó una prueba en línea con el sistema ICC basado en la tableta y obtuvo una velocidad de transferencia de información de un promedio de 33,87 bits / min en tres temas. Como resultado se concluyó, que la integración con el dispositivo móvil fue altamente práctica para la vida diaria.

1.2.3.2 Antecedentes Nacionales

BCI Games With Motion Capture and its Possibilities in Rehabilitation [Muñoz, 2013] En este artículo se propone una BCI para la recreación de un juego con propósitos de rehabilitación. Aquí se emplea un sistema Kinect que sirve para sensar las respuestas del usuario en tiempo real, que permite la captura tanto de los movimientos como de los gestos. Los resultados preliminares fueron obtenidos de un usuario con hemiparesia.

La precisión del software de EPOC para la clasificación de pasos dentro de la dinámica del juego fue del 64%, inferior a los niveles alcanzados en una formación anterior. Por último, se destacan algunas características como la portabilidad, la fluidez de la interacción, la calidad de la experiencia del usuario y el bajo costo del sistema desarrollado, que se convierte en una herramienta ideal para la rehabilitación de usuarios con enfermedades neuro-motoras.

Uso De Potenciales De Estado Estable Evocados Visualmente Para La Implementación De Un Sistema De Interfaz Cerebro Computador [Caicedo, Cardona, 2014] En este trabajo se presenta un primer prototipo de un BCI basado en SSVEP planteado como tecnología de apoyo a la discapacidad, contando con la ventaja de usar un equipo de relativo bajo costo y el uso de electrodos secos, reduciendo el tiempo necesario para su instalación y puesta en marcha. El sistema fue probado en un voluntario adulto de 37 años logrando una precisión del 87% en la identificación de 5 estímulos visuales con un tiempo promedio de 4.5sg para detectar un comando y una rata de transferencia de información de 20bits por minuto.

1.3 Conclusiones

Luego de la revisión bibliográfica realizada se concluyó que los siguientes parámetros se deben considerar a la hora de construir un sistema estimulador de SSVEP:

-Respecto al Estimulo: color, fuente, potencia, rango de frecuencias de trabajo.

Aquí se debe considerar con precaución las frecuencias de los estímulos, ya que algunas de estas pueden provocar ataques de epilepsia, y si se muestra que son excesivamente brillantes, pueden afectar la visión del usuario. Además, algunas frecuencias de estimulación pueden inducir fatiga.

-Estructura de Organización de los Estímulos: La separación entre cada uno de los estímulos y el número de estímulos.

Se debe considerar que al aumentar el número de estímulos se ofrece un mayor número de posibles comandos, pero puede disminuir la precisión y la velocidad de la clasificación.

-Prototipo final: La inclinación y la cantidad de estímulo que se transmite, es decir, la cantidad de luz con la que el usuario es estimulado.

2. DISEÑO E IMPLEMENTACIÓN

El proyecto de “*Diseño y Construcción de un Teclado con Estímulos Visuales para Interfaces Cerebro-Computador*” es una solución ante la necesidad de un sistema con el que se puedan generar estímulos visuales de forma sencilla y rápida. Este sistema debe permitir, junto con la lectura de las secuencias desde un fichero, la visualización, la modificación y la posterior transmisión de secuencias al teclado. Las secuencias corresponden a un conjunto de tiempos de encendido y apagado del led de la tecla, lo que se traduce en una estimulación visual para el usuario. La respuesta a esta estimulación visual, posteriormente, se captura, se procesa y se asocia a una acción, como mover hacia adelante una silla de ruedas, o hacia atrás, o realizar una llamada.

El proyecto desarrollado consiste en 2 principales módulos:

1. Módulo Software: Interfaz con la cual el usuario puede leer las secuencias desde un fichero, visualizar, modificar y transmitir las secuencias al teclado alfanumérico.
2. Módulo Hardware: Corresponde al prototipo físico del teclado con el cual se proyectan las secuencias programadas. Este se compone de 16 teclas, de las cuales 12 son fijas en este y 4 son externas, las cuales el usuario puede ubicar libremente.

En la figura 2 se observa el sistema planteado compuesto por los 2 módulos mencionados anteriormente.

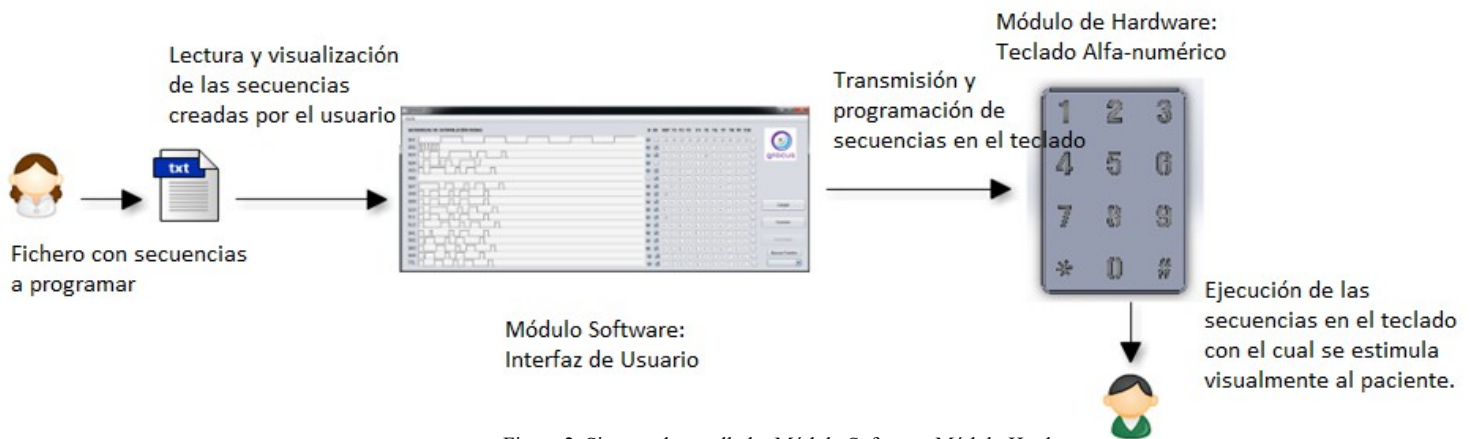


Figura 2. Sistema desarrollado: Módulo Software, Módulo Hardware

A continuación se explica, más a fondo, el desarrollo de estos módulos.

2.1 Módulo Software: Interfaz de Usuario

La interfaz es el componente con el cual el usuario carga un conjunto de secuencias definidas para su visualización, modificación y posterior transmisión al dispositivo encargado de proyectar las secuencias. Como se mencionó anteriormente estas secuencias, corresponden a un conjunto de tiempos que se traducen en estados de encendido y apagado del led. Los tiempos son establecidos según el usuario.

A continuación se presentan las *etapas del desarrollo de software*.

2.1.1 Análisis de Requerimientos

El análisis de requerimientos comprende las restricciones de diseño con las cuales se realizó el proyecto. A continuación se describen las 2 principales actividades y las tareas asociadas a estas:

- Procesamiento de las secuencias extraídas desde un fichero miSecuencia.txt*
- Programación de las secuencias en el teclado*

La primera actividad cubre las tareas de *lectura, visualización, actualización y almacenamiento* de las secuencias; En la primera tarea se debe hacer una validación de los datos leídos, es decir, que se debe verificar que cumplan con los rangos y formatos de valores propuestos en el proyecto.

Las **secuencias** contienen los siguientes datos:

Valor Inicial: Corresponde al estado inicial de la secuencia, es decir, si comienza en bajo o en alto (apagado o encendido). El valor es discreto y se encuentra entre $[0, 1]$.

Desplazamiento: Valor en ms en el rango $[0, t]$, donde t corresponde a los tiempos de la secuencia] que indica el tiempo en el cual se debe activar la secuencia. Este representa un tiempo de activación desplazado (ver Figura 3).

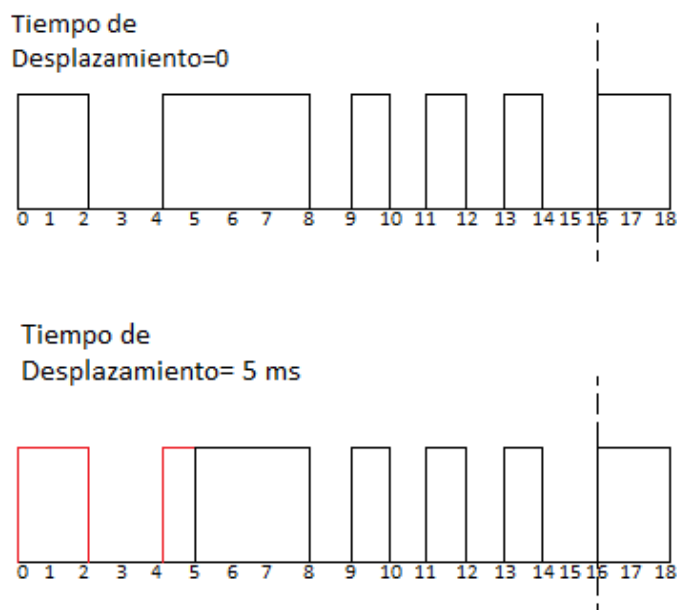


Figura 3. Secuencia con desplazamiento de 0 ms y la misma secuencia con desplazamiento 5ms.

En la figura 3 se observa que la secuencia con tiempo de desplazamiento igual a 0 ms muestra la secuencia normal desde el principio, mientras que la segunda secuencia con tiempo de desplazamiento de 5 ms, se activa a los 5 ms y luego continúa igual que la anterior.

Se debe mencionar que estas consideraciones en las secuencias son por solicitud del director de tesis, ya que el proyecto hace parte de un macro proyecto titulado PLATAFORMA ROBÓTICA PARA LA MOVILIDAD ASISTIDA DE PERSONAS CON DISCAPACIDAD entre la Universidad del Valle y la Universidad del Quindío [Caicedo, Cardona, 2014].

Des/Habilitado: Corresponde a la activación de la tecla. Si está activada entonces la tecla muestra la secuencia programada, si no, está apagada. El valor es discreto y se encuentra entre [0, 1].

Tiempos: Una secuencia de números en ms indicando cuanto tiempo debe mantenerse el estado de la luz encendido y apagado. Cada tiempo está entre 1 y 77 ms con los cuales se obtienen frecuencias entre 6,49 y 50,00 Hz.

El requerimiento de *Visualización* consiste en mostrar un ciclo completo de la secuencia, en la cual el usuario pueda observar, analizar y comparar las secuencias a programar. Es una ayuda visual con la que el usuario puede verificar las secuencias construidas.

La *actualización* consiste en permitir que el usuario modifique las secuencias leídas. Esto a la vez actualizará la visualización de las mismas.

El *almacenamiento* consiste en generar un nuevo archivo a partir de la secuencia modificada u original.

La segunda actividad consiste en *transmitir las tramas de las secuencias al dispositivo* por medio de una comunicación *usb*. Además de la restricción en la comunicación se deben procesar los datos de las secuencias, de tal forma que sean comprensibles para el microcontrolador. Para los valores de estado inicial y habilitación, la información se transmite al microcontrolador como [0,1]. En el caso del parámetro de desplazamiento para cada secuencia, este es recalculado por medio de la Ec.1, esta establece que el valor transmitido corresponde al módulo de la suma total de los tiempos de la secuencia, es decir:

$$\text{Tiempo de Desplazamiento Transmitido} = (\text{Tiempo de Desplazamiento}) \% \quad [\text{Ec. 1}]$$

Para los valores de los tiempos se hace una conversión a ASCII, de los valores superiores a 10.

A continuación se presenta el diagrama de casos de uso (Figura 4) asociado a las funcionalidades del sistema, cabe mencionar que este diagrama corresponde a los requerimientos funcionales solicitados por el usuario.

En los siguientes apartados se explicará cómo se desarrolló el software para darle respuesta a estos requerimientos.

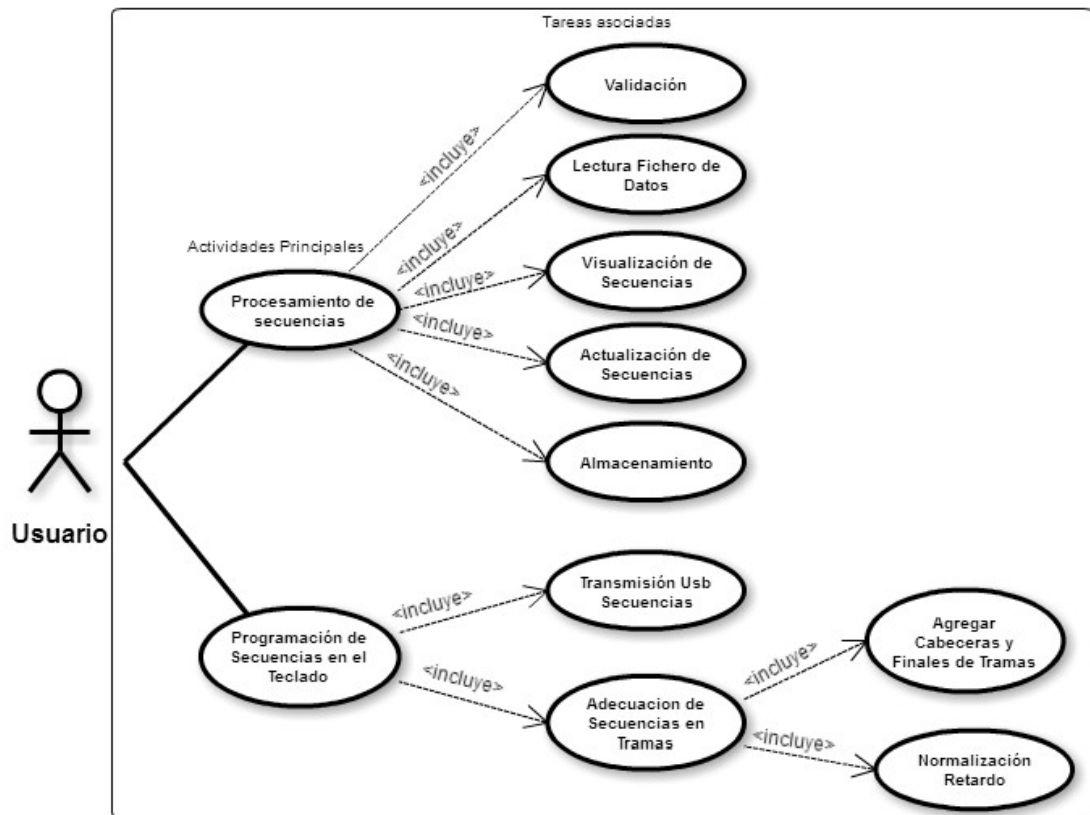


Figura 4. Diagrama de Casos de Uso

2.1.2 Diseño y Arquitectura

Inicialmente para realizar el diseño del software se deben establecer las relaciones de dependencia entre cada una de las funcionalidades anteriormente expuestas, esto con el propósito de estructurar los objetos y las relaciones que se construirán en el software. En las figuras 5 y 6 se observa cómo se integran las funcionalidades principales con las tareas asociadas, y además, se muestra cómo fluye la información entre cada funcionalidad.

En la figura 5 se presenta el diagrama de flujo para la actividad de Procesamiento de Secuencias, en esta se expone el orden con el cual la información de las secuencias es procesada, así:

1. Se lee fichero
2. Se validan los datos: rango, tipo, cantidad de datos.
3. Se grafican las secuencias: se dibuja en la interfaz la secuencia leída.
4. Si hay una actualización se redibuja la secuencia en la interfaz.
5. El usuario puede almacenar esta secuencia.

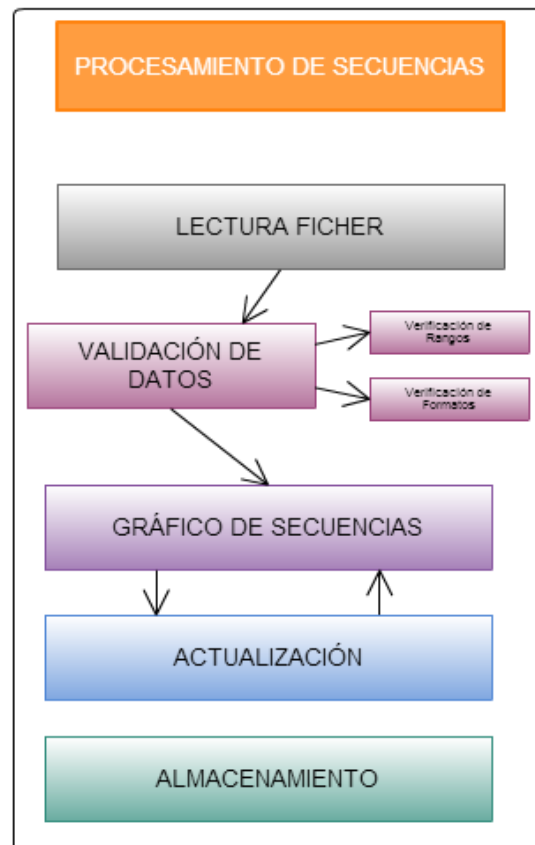


Figura 5. Diagrama de Flujo para la actividad de Procesamiento de Secuencias

En la figura 6 se muestra el diagrama de flujo para la actividad de Programación de las secuencias en el Teclado. Aquí se observa que los datos antes de ser transmitidos deben ser adecuados a la trama establecida en el microcontrolador. El retardo o desplazamiento debe ser un valor normalizado que no puede superar el tiempo máximo de la secuencia; para esto se calcula el módulo y el resultado es el tiempo que se toma como retardo (Ec. 1). Además del acondicionamiento de los datos, se realiza una copia de las secuencias transmitidas, esto con el fin de que el usuario conozca cual fue la última configuración programada.

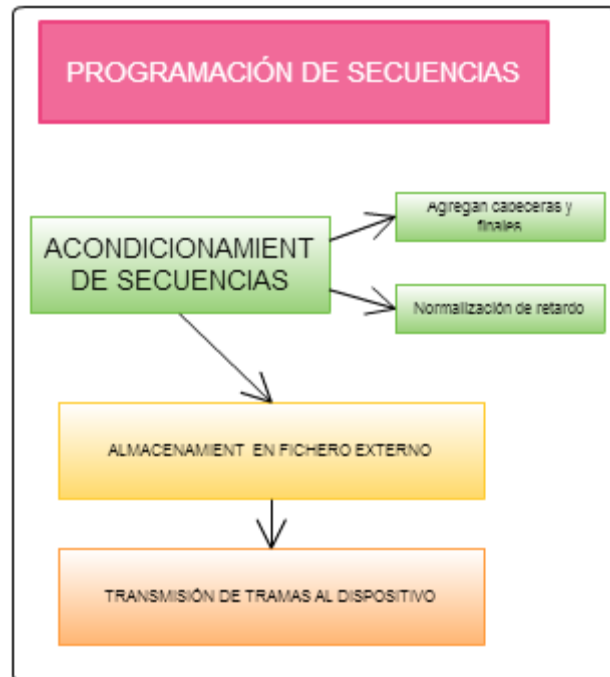


Figura 6. Diagrama de Flujo para la Actividad de Programación de Secuencias en el Teclado

Luego de haber establecido las relaciones entre las funcionalidades, se selecciona el patrón de arquitectura bajo el cual se desarrolla la aplicación. Para el proyecto se determinó que el patrón de arquitectura *Modelo- Vista- Controlador (MVC)* cubre de forma robusta los requerimientos de software. El patrón de arquitectura MVC separa los [datos](#) y la [lógica de negocio](#) de la [interfaz de usuario](#) y del módulo encargado de gestionar los eventos y las comunicaciones.

Los componentes propios de la arquitectura MVC se definen de la siguiente manera:

Modelo: Es la representación de la información con la cual el sistema opera, por lo tanto gestiona todos los accesos a dicha información, tanto consultas como actualizaciones, implementando también los privilegios de acceso que se hayan descrito en las especificaciones de la aplicación. Envía a la Vista aquella parte de la información que en cada momento se le solicita para que sea mostrada. Las peticiones de acceso o manipulación de información llegan al Modelo a través del Controlador.

Controlador: Responde a eventos e invoca peticiones al Modelo cuando se hace alguna solicitud sobre la información. También puede enviar comandos a su Vista asociada si se

solicita un cambio en la forma en que se presenta el Modelo, por tanto se puede afirmar que el Controlador hace de intermediario entre la Vista y el Modelo.

Vista: Presenta el Modelo, es decir la información y lógica del negocio en un formato adecuado para interactuar.

La arquitectura de la aplicación se presenta en la figura 7, junto con sus componentes.

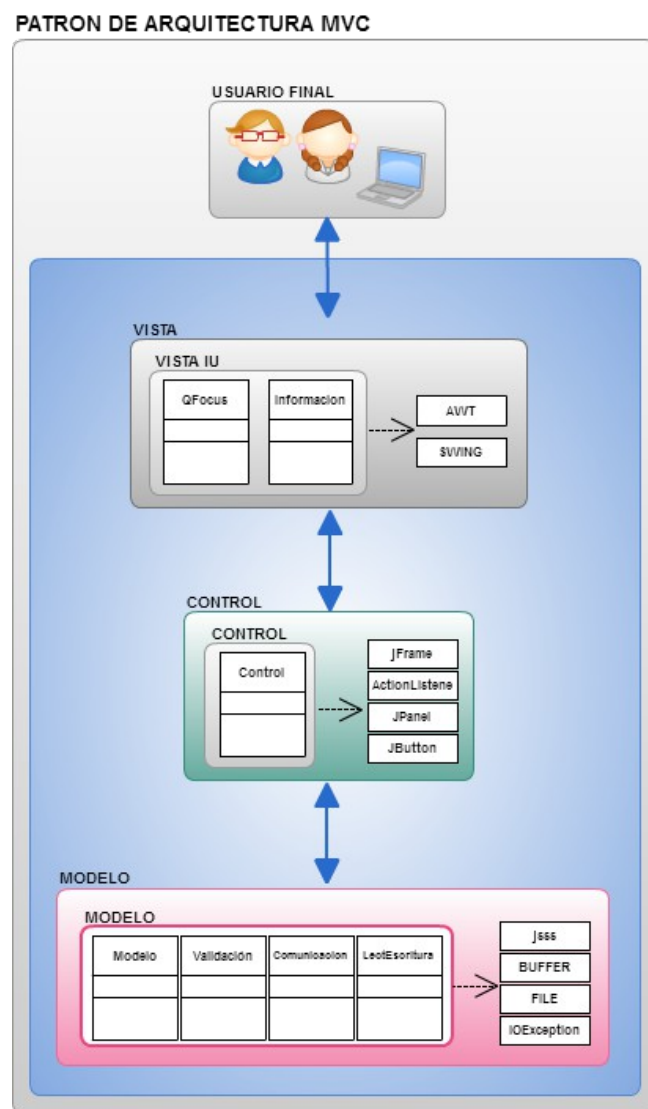


Figura 7. Patrón de Arquitectura MVC para la aplicación

La vista de la arquitectura en la Figura 7 permite observar como viaja la información de las secuencias a través de los módulos vista, controlador y modelo de la aplicación. La interacción comienza por el usuario en la ventana QFocus, en donde genera una

solicitud, ya sea descargar el manual de usuario, cargar un fichero de secuencias o transmitir la secuencia al teclado, luego, esta solicitud es enviada al controlador, el cual la re-direcciona hacia el modelo para que este la resuelva. Finalmente, la respuesta es enviada por el mismo camino hacia la Interfaz para que el usuario pueda visualizar la respuesta dada por el sistema.

2.1.3 Codificación

Consideración de la programación:

- La aplicación fue construida bajo la plataforma de desarrollo Java SE, con JDK versión 8.0_25.
- Se empleó la librería jssc v2.8 para la comunicación serial.
- Se seleccionó como **QFOCUS 1.0** el nombre la aplicación desarrollada.

En la figura 8 se presenta el diagrama de paquetes de la aplicación, en donde se pueden observar las agrupaciones lógicas y las dependencias entre estas. Esta distribución lógica de funciones sigue estando bajo el patrón MVC.

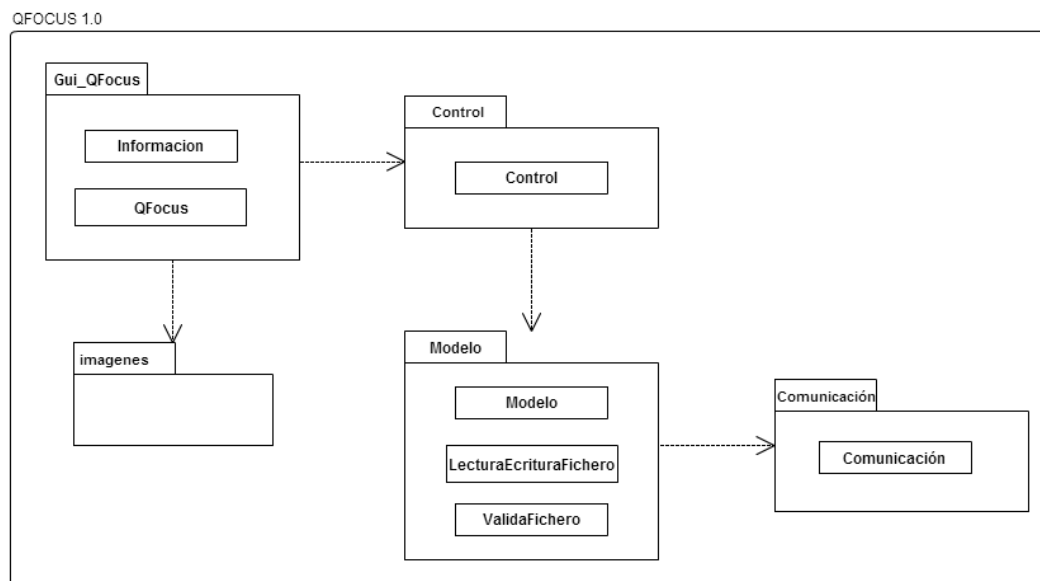


Figura 8. Diagrama de Paquetes

A continuación se da una breve explicación de cada uno de los paquetes y sus clases.

Paquete Vista: Directorio que almacena las clases relacionadas con la interfaz de Usuario.

Clase QFocus: Clase donde se construye la vista principal de la aplicación.

Clase Información: Clase secundaria donde se construye la ventana Acerca de...

Paquete Control: Directorio encargado de la escucha de las solicitudes del usuario y del envío de estas al Modelo.

Clase Control: Clase encargada de implementar los listener de la clase QFocus y de hacer las invocaciones a los métodos adecuados en el paquete Modelo.

Paquete Modelo: Directorio encargado de las tareas de procesamiento de secuencias y transmisión de datos.

Clase Modelo: Clase encargada realizar la creación de objetos e invocar los métodos adecuados en las clases respectivas, según la solicitud enviada por el control.

Clase Validación: Clase encargada de la validación de los datos leídos, actualizados, almacenados y transmitidos.

Clase LectoEscritura: Clase encargada de la lectura y escritura de ficheros.

Paquete Comunicación: Directorio encargado de la transmisión de las tramas al dispositivo por medio de comunicación serial.

Clase Comunicación: Clase encargada de la transmisión de las tramas.

Para más detalle sobre la estructura jerárquica de las clases diríjase a la documentación del código.

Finalmente, en la figura 9 se observa el diagrama de clases de la aplicación.

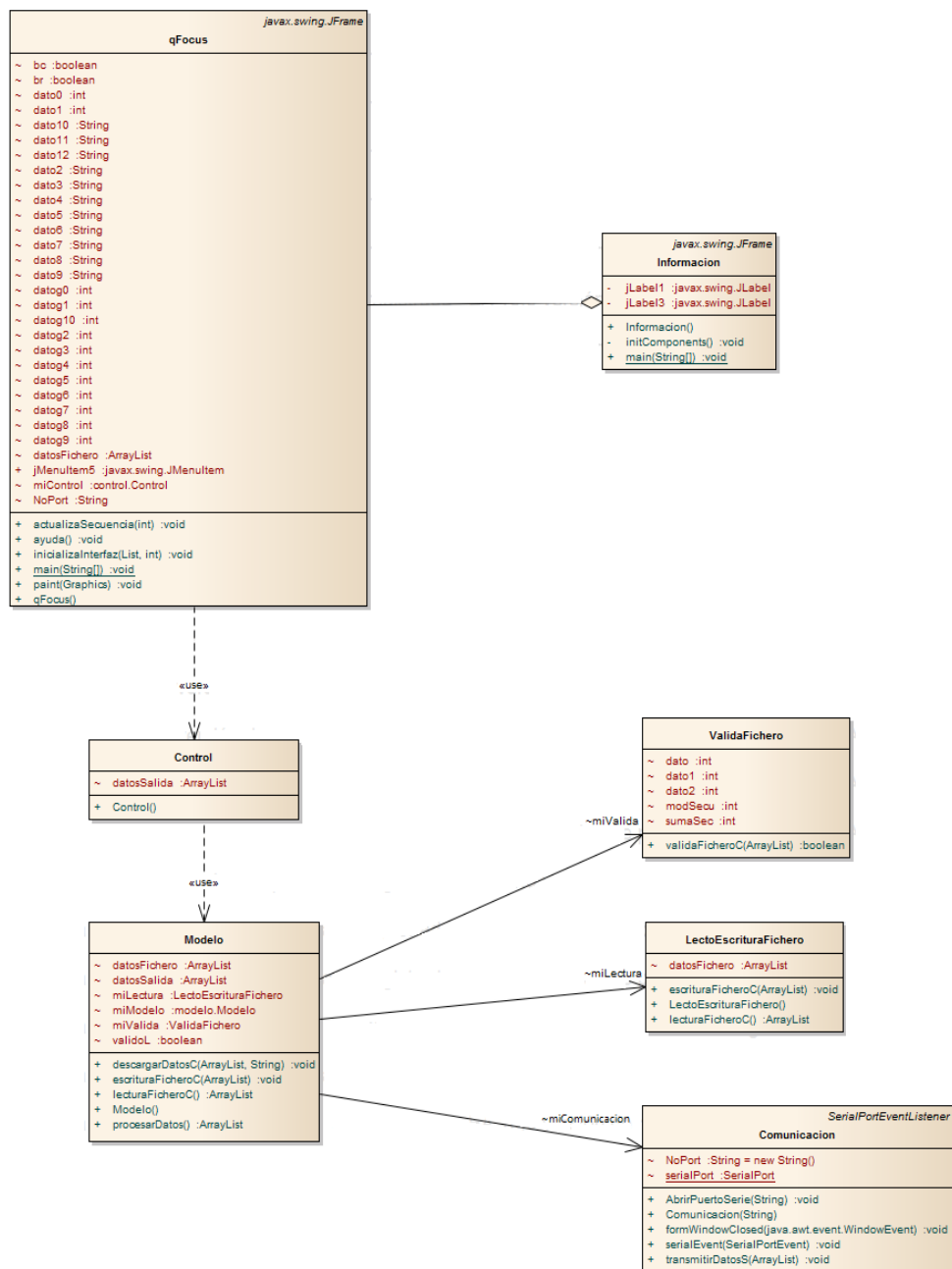


Figura 9. Diagrama de Clases

Por último, **QFocus** es distribuida bajo licencia GLP v3. El repositorio de la aplicación puede ser encontrado en: <https://github.com/savacano28/QFocus.git>

2.1.4 Pruebas y Resultados

En la figura 10 se presenta la interfaz desarrollada para la aplicación **QFOCUS 1.0**. En esta se pueden observar 3 divisiones principales: *Control*, *Visualización* y *Actualización*, las cuales dan cumplimiento a los requerimientos del proyecto.

El área de *Control* está compuesta por las tareas de carga, almacenamiento y transmisión de Tramas. *Visualización* comprende el gráfico de un ciclo de las secuencias. Y *Actualización*, cubre los campos en donde se pueden editar las secuencias cargadas desde el fichero.

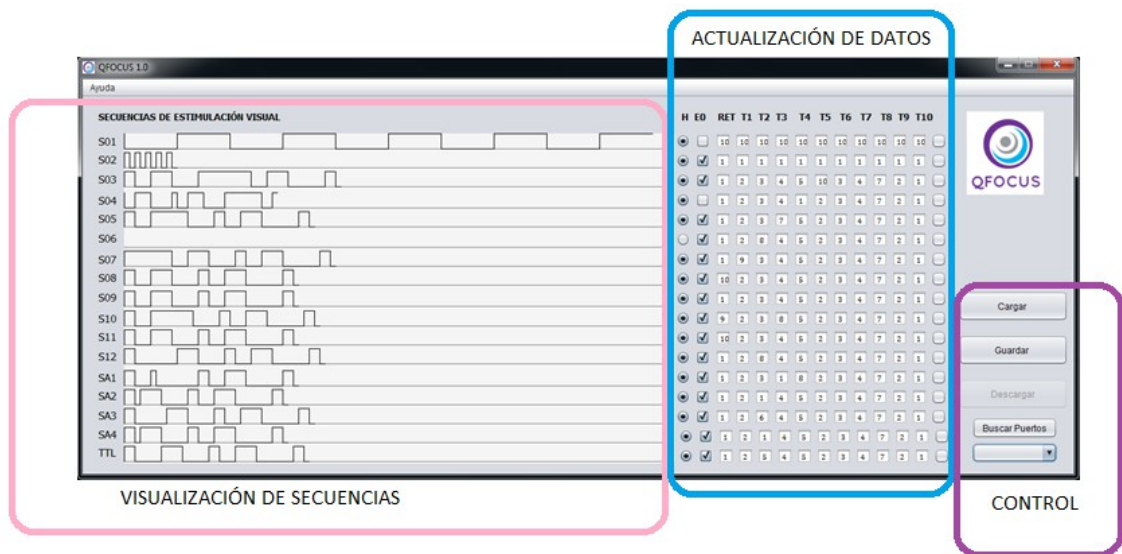


Figura 10. Interfaz **QFocus 1.0**

En las figuras 11 hasta la 14 se observan las pruebas que se realizaron al software **QFOCUS 1.0**. En la figura 11 se presenta la carga de un archivo de frecuencias, aquí se observa el fichero con las secuencias a cargar y la Interfaz QFocus 1.0 con las secuencias ya cargadas y visualizadas. En la figura 12 se presenta la actualización de una secuencia sobre la interfaz, en esta se observa además que la visualización de la interfaz también es actualizada. En la figura 13 se observa que la interfaz reconoce el dispositivo para establecer la comunicación serial. Finalmente, en la figura 14 se observa que la interfaz permite el almacenamiento de las secuencias modificadas.

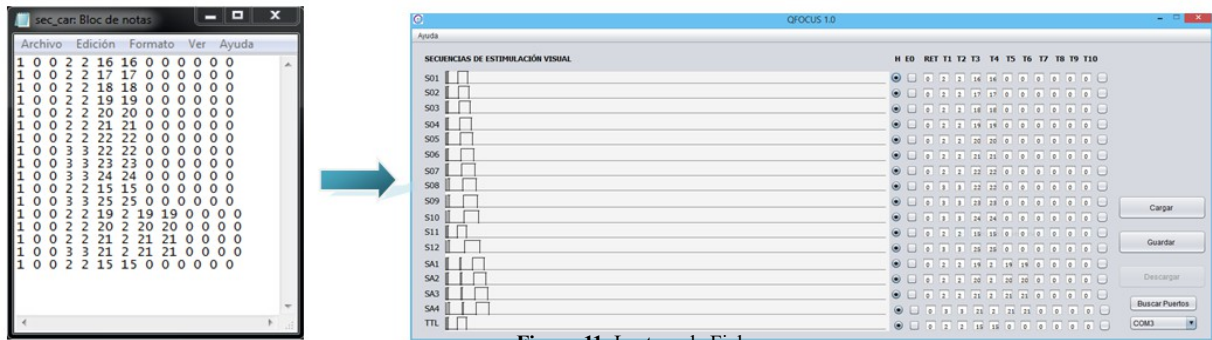


Figura 11. Lectura de Fichero

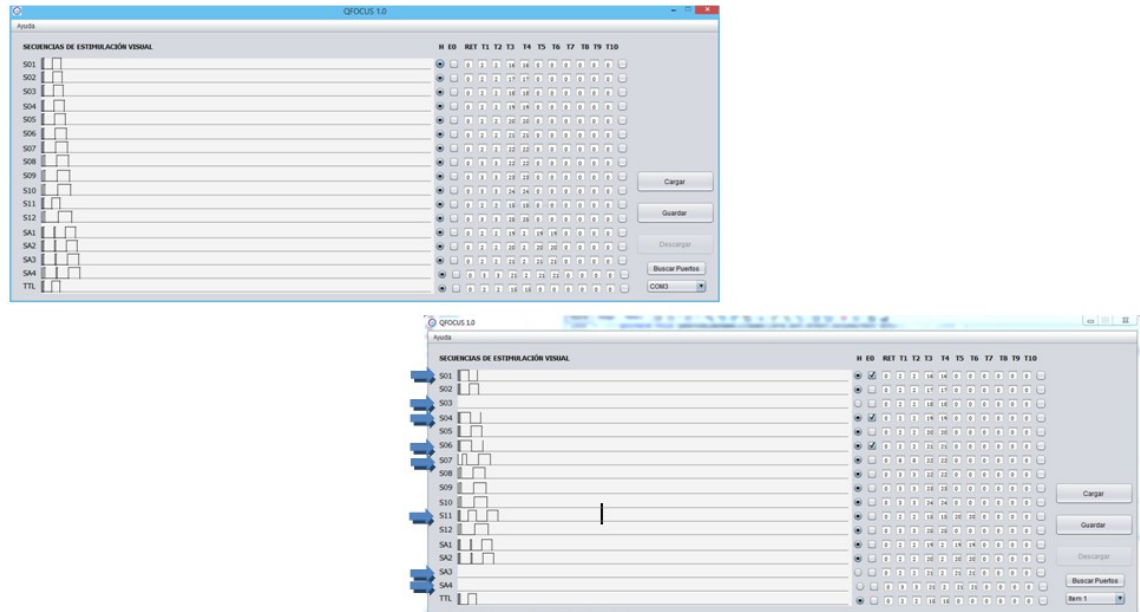


Figura 12. Modificación de Secuencias (s01, s03, s04, s06, s11, sA3, sA4) sobre la Interfaz de QFocus 1.0

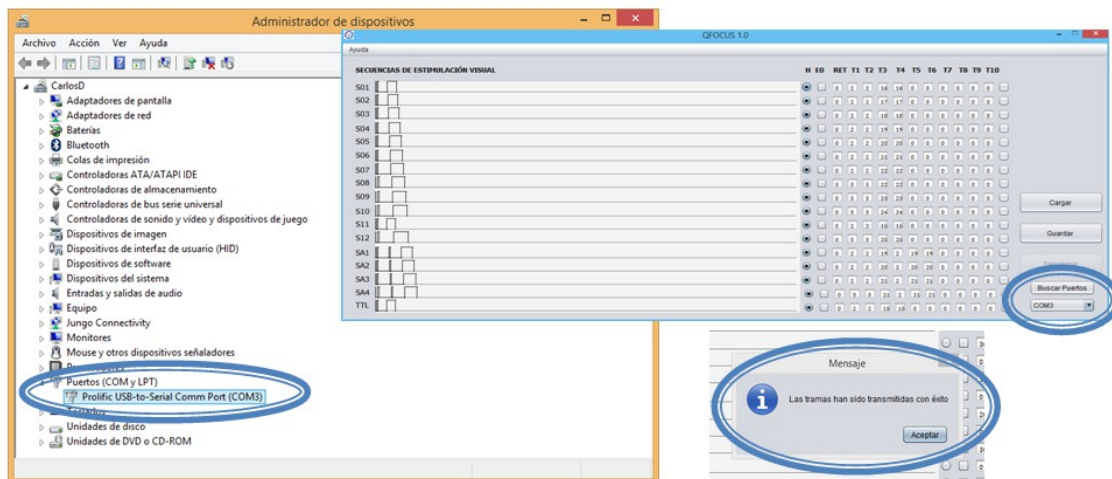
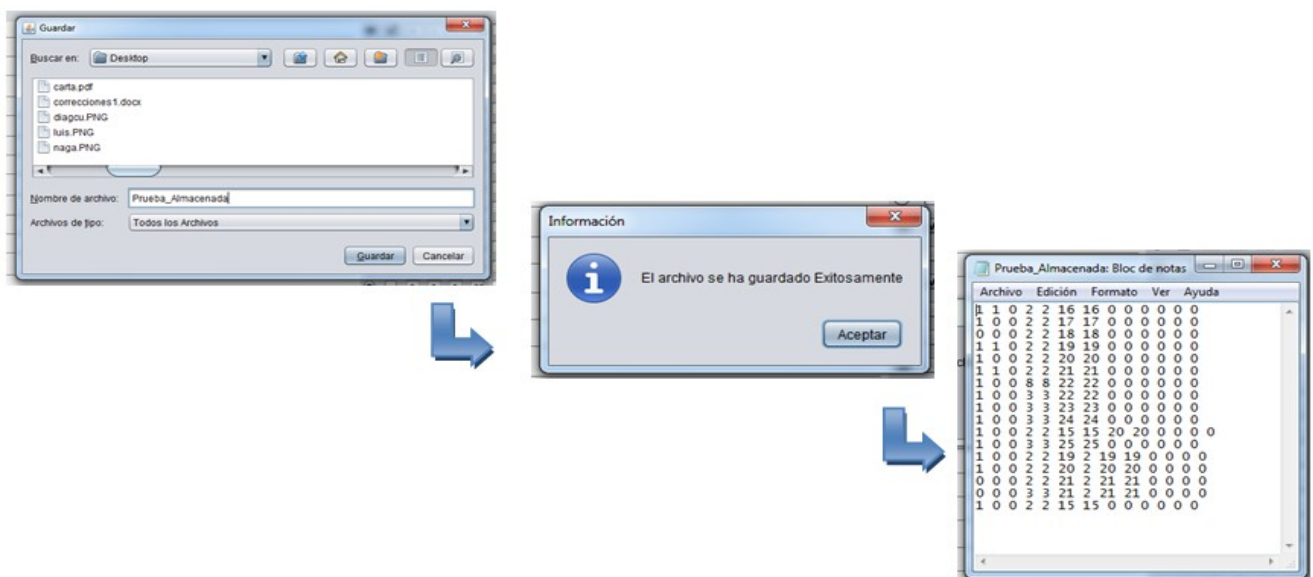


Figura 13. Comunicación Serial y Programación del micro

Figura 14. Almacenamiento de Secuencias



Por último, la documentación del software incluye:

- Guía de Usuario (ver documento Anexo)
- Documentación del código

2.2 Módulo Hardware: Tarjeta de desarrollo y Matriz de leds

Esta etapa del proyecto corresponde al desarrollo del hardware que permite comunicar la interfaz de usuario **QFOCUS 1.0** con la tarjeta de leds, la cual estimulará visualmente a un usuario para la posterior verificación de la generación de ondas SSVEP.

2.2.1 Características Funcionales

El módulo debe operar en dos modos:

- Modo Programación:* En este modo, se debe interactuar con la interfaz de usuario para programar a gusto las secuencias, incluyendo tiempos de desplazamiento, estados iniciales y habilitación/inhabilitación de cada uno de los estímulos. Dentro de las consideraciones de diseño, se establece que la conexión debe ser usb.
- Modo Automático:* El teclado debe mostrar de forma automática el último conjunto de secuencias cargadas, es decir, éste debe reproducir las secuencias sin necesidad de estar conectado a un computador. Cada conjunto de secuencias programadas en el teclado deben ser almacenadas para su posterior ejecución.

2.2.2 Diseño

Son necesarios dos diseños de tarjetas, una con doce leds fijos y cuatro móviles y una tarjeta de desarrollo, la cual se comunica con la interfaz de usuario y con la matriz de leds para reproducir las secuencias.

2.2.2.1 Tarjeta de Leds

Según los requerimientos especificados, son necesarios 16 estímulos visuales, 12 fijos en un teclado numérico (del 0 al 9 incluyendo los caracteres * y #) y los otros 4 estímulos no fijos (con los caracteres A, B, C y D), de forma tal, que el usuario pueda ubicarlos libremente según su parecer.

De acuerdo al diseño de [Han-Jeong, 2012], se debe tener una distancia de 2cm entre teclas adyacentes y cada tecla debe ser de 2 cm x 2 cm. Además se debe usar una luz blanca. Por lo tanto, la tarjeta de leds debe cumplir con dimensiones de 15x11 cm, con leds en las posiciones (largo, ancho y en centímetros) como se observa en la tabla 1.

Tabla 1. Organización espacial de cada led en el diseño de la tarjeta

Carácter de Led	Largo(cm)	Ancho(cm)
1	1.5	1.5
2	1.5	5.5
3	1.5	9.5
4	5.5	1.5
5	5.5	5.5
6	5.5	9.5
7	9.5	1.5
8	9.5	5.5
9	9.5	9.5
*	13.5	1.5
0	13.5	5.5
#	13.5	9.5

Para las cuatro teclas individuales se realizó un diseño simple con una tarjeta de 3.5x3.5 cm con el led en el centro de cada una. En la figura 15 se observa el diseño de la tarjeta de led y en la figura 16 se observa la conexión realizada para cada led.

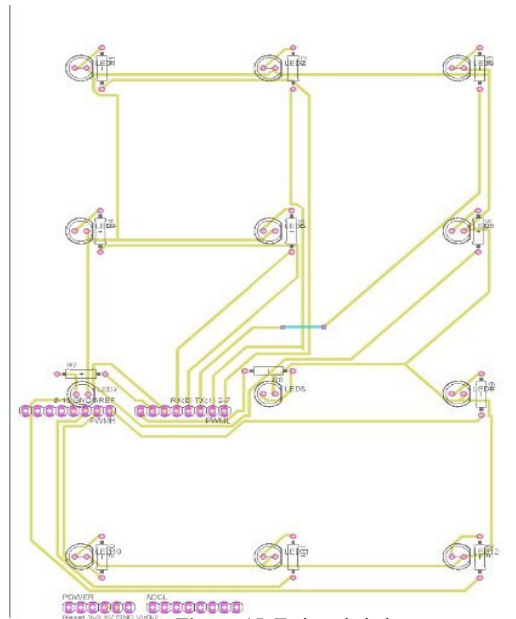


Figura 15. Tarjeta de leds

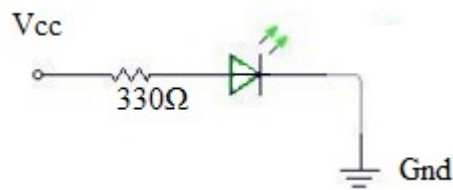


Figura 16. Diseño general de la conexión de cada led

2.2.2.2 Tarjeta de Desarrollo

Las consideraciones de diseño para esta tarjeta son:

- 16 salidas digitales, correspondientes a los 16 estímulos programables
- 1 una salida TTL, señal de control
- Mínimo de 512 MB de EEPROM, para el almacenamiento de las 16 secuencias.
- Mínimo 1KB de FLASH,
- Comunicación serial, para la comunicación con la interfaz de Programación **QFOCUS**

1.0

Con 32 salidas digitales distribuidas en sus 4 puertos, una memoria FLASH de 64 Kbytes, una EEPROM de 2 Kbytes, un timer counter de 16 bits y otros dos timers de 8 bits, USART serial programable y la posibilidad de conectarse a un oscilador externo de

hasta 20MHz, el ATMEGA644 cumple con los requisitos mencionados para la tarjeta de desarrollo.

En la figura 17 se observa el diseño de la tarjeta de desarrollo.

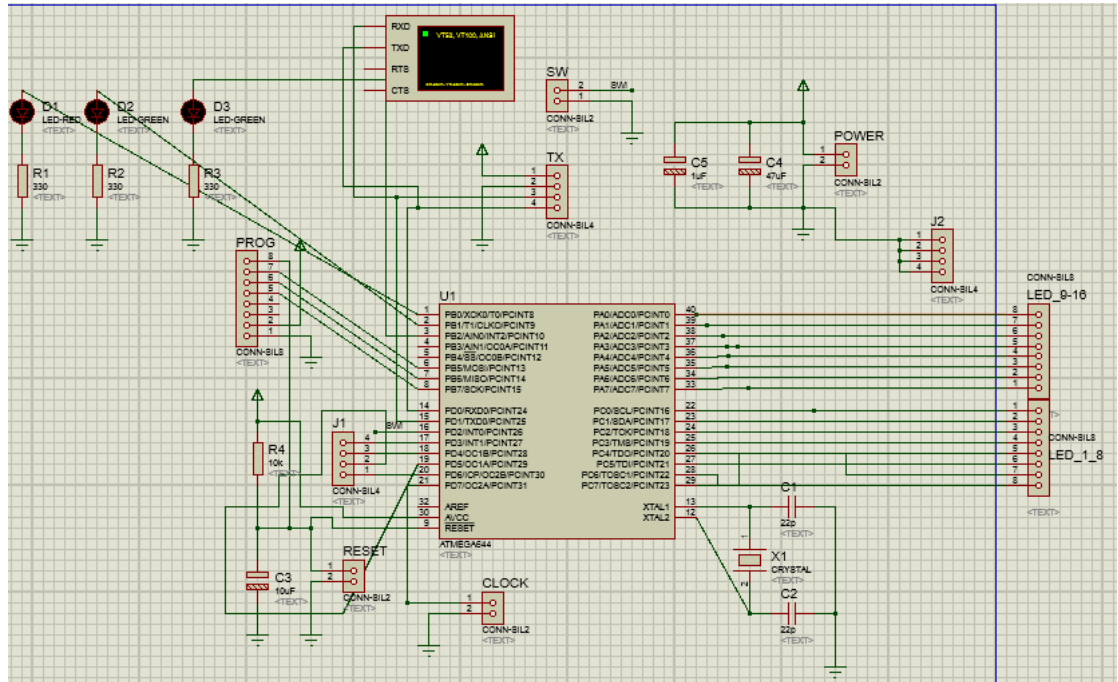


Figura 17. Diseño de la tarjeta de desarrollo

2.2.2.3 Resultados

En la figura 18 se observa la vista posterior y anterior de las 2 tarjetas construidas para el proyecto.

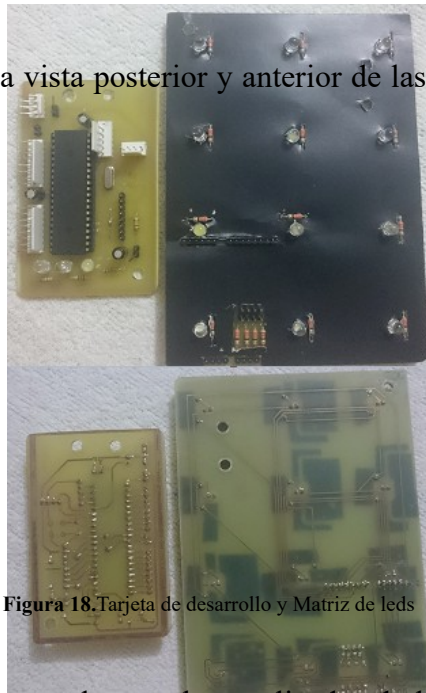


Figura 18. Tarjeta de desarrollo y Matriz de leds

En el apartado 2.2.3 se muestran las pruebas realizadas de la integración de la interfaz de *QFOCUS 1.0* con las 2 tarjetas.

2.2.3 Programación del Microcontrolador

El programa contenido en el ATMEGA644 tiene como función principal comunicarse con la interfaz de usuario, tarea que realiza mediante comunicación serial, usando un convertidor USB-serial (PL2303) y con diferentes tramas para indicar la tarea a cumplir.

Son tres tramas que se usan en el programa: LOAD para cargar los datos y condiciones iniciales de cada una de las 17 salidas, START para iniciar la aplicación (se deben haber cargado previamente las 17 tramas LOAD) y STOP para detener el proceso de ejecución.

Otra tarea del programa del microcontrolador es guardar y ejecutar todas las secuencias en caso de desconexión/conexión. Esto implica calcular retardos, estados iniciales y habilitación de teclas, recorrer arreglos de tiempos, determinar el tipo de trama de comunicación con la interfaz de usuario y atender a interrupciones externas, ya sea de timer counter o de USART.

En esta sección se especificará cada trama usada (START, LOAD y STOP) con el fin de no ser repetitivos en las siguientes secciones. Cada trama tiene 20 datos en formato ascii repartidos de la siguiente manera (Figura 19):

Figura 19. Formato general de tramas

Donde, los caracteres 'I' y 'D' son identificadores de inicio de trama. 'H' es el indicador de que la trama es activo o no, es decir, si está habilitada/deshabilitada, los valores que toma son: 0-Deshabilitada y 1-Habilitada. Tr es el tiempo de retardo o desplazamiento en la activación de la trama (*para más detalle diríjase a la sección 2.1.1 Análisis de Requerimientos*). 'INI' corresponde al estado inicial (0 ó 1), es decir, si la secuencia comienza en bajo o en alto. Los valores de 'R1' a 'R4' indican la operación que debe realizar el micro: *start*, *load*, *stop*. De 'T1' a 'T10' corresponden a los 10 datos de tiempos de las secuencias que fueron cargadas a la interfaz de **QFOCUS 1.0**. 'TN' es el

número de datos antes de un cero entre T1 y T10 y ‘~’ es el indicador de final de trama, sin embargo, no es un dato que se guarde pues la trama seria de 21 ítems.

A continuación se explican las operaciones que realiza el microcontrolador:

-*START*: Su característica principal es que en la posición 3 se encuentre una ‘A’ y en los campos R1 y R2 una ‘S’ y ‘T’ respectivamente, el resto de campos con ‘x’ no importa el valor que tengan. Su estructura se ve en la figura 20.

Figura 20. Formato de trama START

-*LOAD*: Los campos son los mismos que posee la trama general. Se leen todos los datos y se guardan todos a excepción de la ‘~’. Los campos R1 y R2 siempre van a contener respectivamente ‘L’ y ‘D’ como referencia a la palabra LOAD y los dos campos siguientes R3 y R4 corresponden al número de led y puede ir desde ‘0’‘1’ hasta ‘1’‘7’. En la figura 21 se ve el formato de la trama LOAD.

Figura 21. Formato de trama LOAD

-*STOP*: La posición 3 siempre debe tener ‘A’ y en R1 y R2 deben estar respectivamente los símbolos ‘O’ y ‘P’. Su estructura se observa en la figura 22 y los campos con ‘x’, su valor no tiene importancia.

Figura 22. Formato de trama STOP

En la figura 23, se observa el diagrama de flujo del algoritmo empleado para realizar las tareas anteriormente descritas.

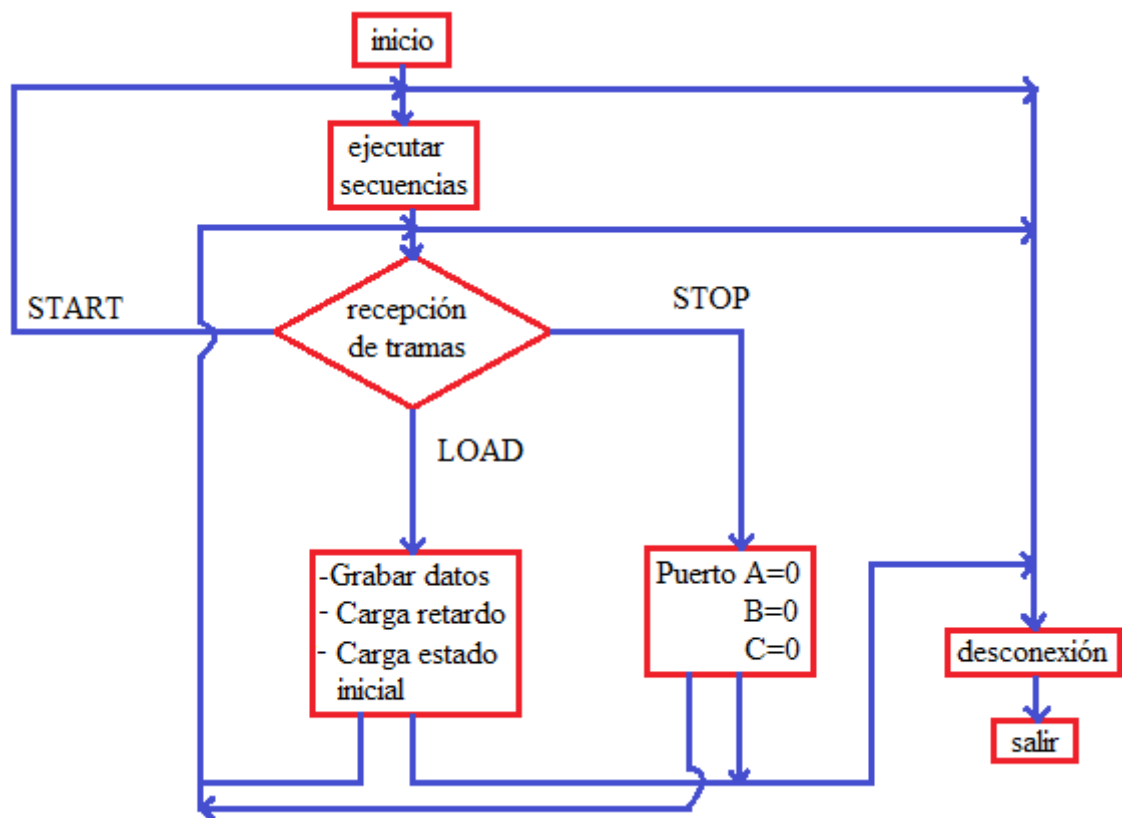


Figura 23. Diagrama de flujo del algoritmo del microcontrolador

2.2.3.1 Métodos

En esta sección se explica de manera cronológica cada uno de los métodos; luego en la figura 24 se observará la jerarquía de cada método durante la ejecución del programa. Son en total 8 métodos o rutinas que se emplearon en la programación. A continuación se explican cada uno de los métodos desarrollados:

-interrupt [USART_RXC] void usart_rx_isr(void): recepción de datos; cada vez que un dato sea enviado desde la interfaz de usuario, se atenderá con prioridad y se verificará la validez de la trama. Con un buffer de 20 datos en total, se verifica que los dos primeros datos recibidos sean 'I' y 'D', así como que el último sea 'F'. Si la trama es válida se guardará lo que haya en el buffer y si no se ignorará y se esperará a una nueva comunicación entre interfaz y microcontrolador.

-interrupt [TIM1_OVF] void timer1_ovf_isr(void): ejecución de secuencias. Una vez se da el START o hay una conexión con datos previamente cargados, se habilita la interrupción por desborde del timer counter 1 (16 bits), la cual ocurre cada milisegundo y lo que hace es actualizar el estado de cada led habilitado, revisando la posición actual y una vez terminada una secuencia se encarga de volverla a iniciar.

-void carga_out(char sele): se ejecuta siempre después de load_retardo(); el cual arroja el verdadero estado inicial de cada secuencia. Además, carga_out() está acompañado de una selección '0' después de LOAD o un '1' después de START. Después de LOAD lo que hace es invertir la salida para que se note un cambio de estado y después de START activa las salidas.

-void eeprom_data_load(): carga de eeprom a ram los 10 tiempos de cada una de las 17 secuencias, los estados iniciales, los tiempos de retardo, la habilitación o no y el número de datos validos (datos que no son cero).

-void carga_ser(): carga de ram a ram los 10 tiempos de cada una de las 17 secuencias. Los 17 arreglos nuevos son los que se recorrerán en la ejecución de la interrupción del timer1.

-void load_retardo(): Es el encargado de calcular el retardo o desplazamiento de cada salida. En este proyecto el retardo es entre 0 y la suma de todos los tiempos de la secuencia menos 1 milisegundo y corresponde a los últimos milisegundos de esa secuencia. Esta rutina arroja como resultado una posición en el array de tiempos de la secuencia y el estado en el que iniciarán a ejecutarse.

- void evalua(): tiene como tarea identificar y ejecutar la orden que se le manda desde la interfaz de usuario. Lee los 20 datos que llegan del buffer e identifica si se trata de LOAD, START o STOP. Cuando identifica LOAD guarda todos los datos ya conocidos (tiempos de ejecución, estado inicial, habilitación, retardo, etc.). Cuando es START empieza la ejecución con la interrupción del TC1 (timer counter 1) y cuando es STOP las salidas de los puertos A, B y C los manda a estado bajo (0).

-void main(void): contiene la inicialización de puertos, timers, usart, interrupciones externas, la ejecución del START (por si viene de una desconexión de energía, arrancar con los datos previamente guardados) y la ejecución de un ciclo infinito que lee el buffer para la posterior ejecución de evalua().

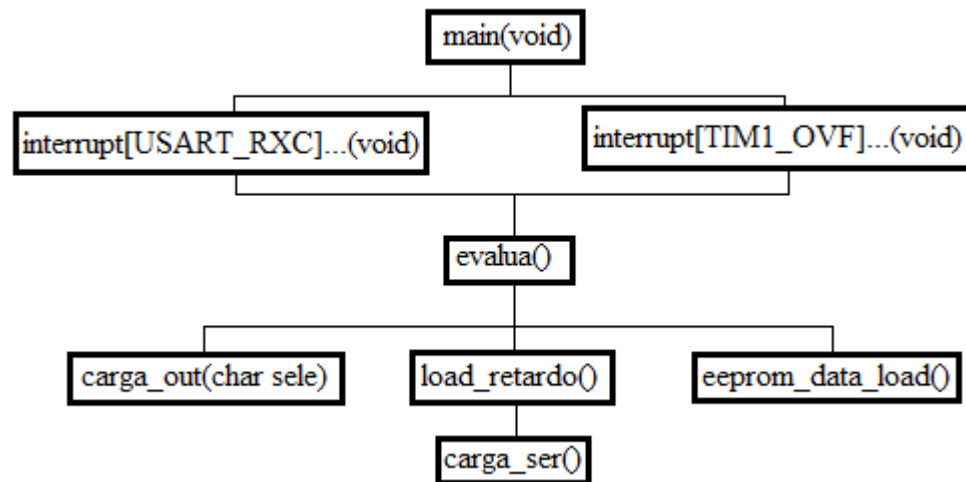


Figura 24. Jerarquía de métodos en la programación del microcontrolador

2.2.3.2 Pruebas y Resultados

Terminada la programación del microcontrolador, el paso siguiente es verificar en un osciloscopio las funciones programadas. Para esto se hicieron las siguientes pruebas.

-Cumplimiento de los tiempos programados en las secuencias: se programaron varias secuencias con el fin de comprobar que se cumplen los tiempos. En este ejemplo se documentó la secuencia [5 10 15 20] ms. Puede verse en la figura 25.

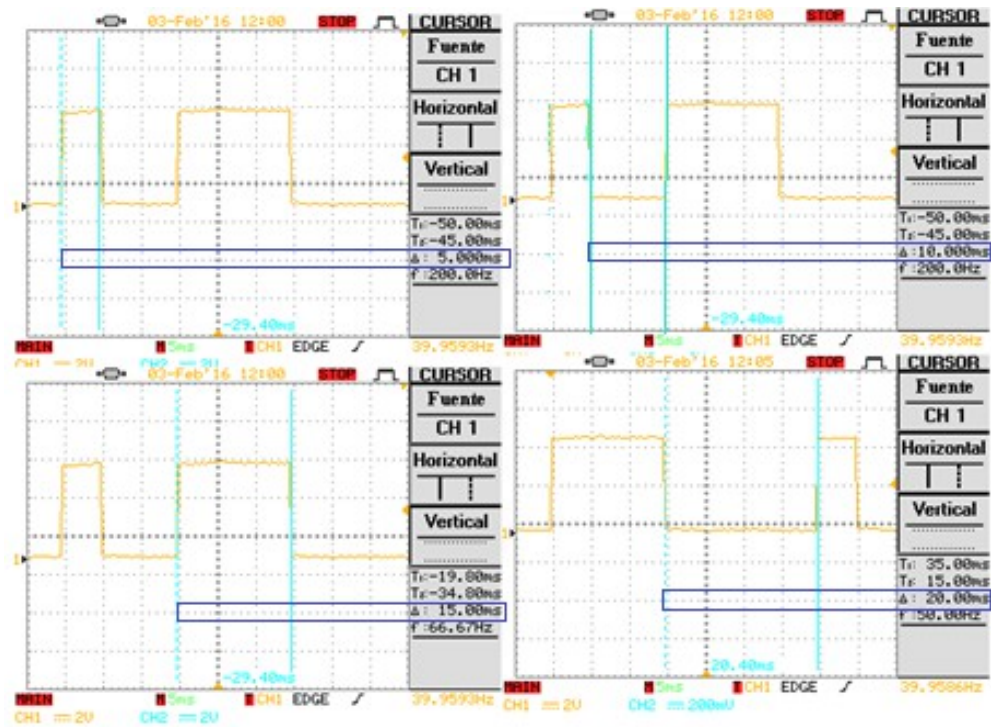


Figura 25. Verificación de tiempos de secuencia [5 10 15 20]

-Verificar los retardos: para esta prueba se programaron dos secuencias iguales con 10ms de retardo una de otra. La prueba de su cumplimiento está en la figura 26.

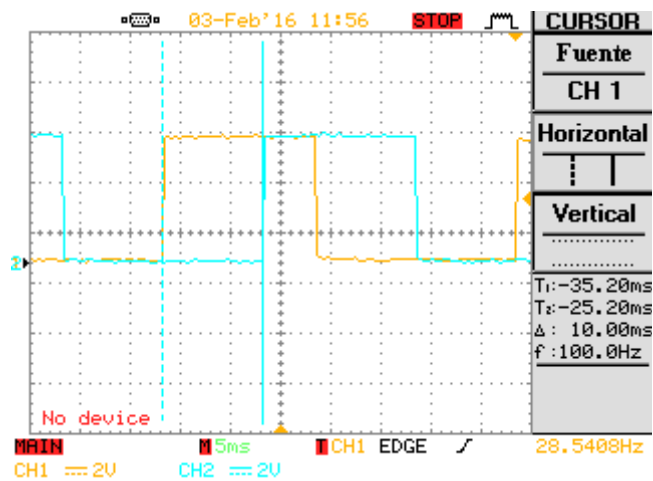


Figura 26. Verificación de cumplimiento de retardos

-Medir los tiempos de cambio de estado: Se mide en una secuencia cuánto tarda en cambiar de estado, tanto de bajo a alto como al revés. En la figura 27 se observan estos 2 tiempos.

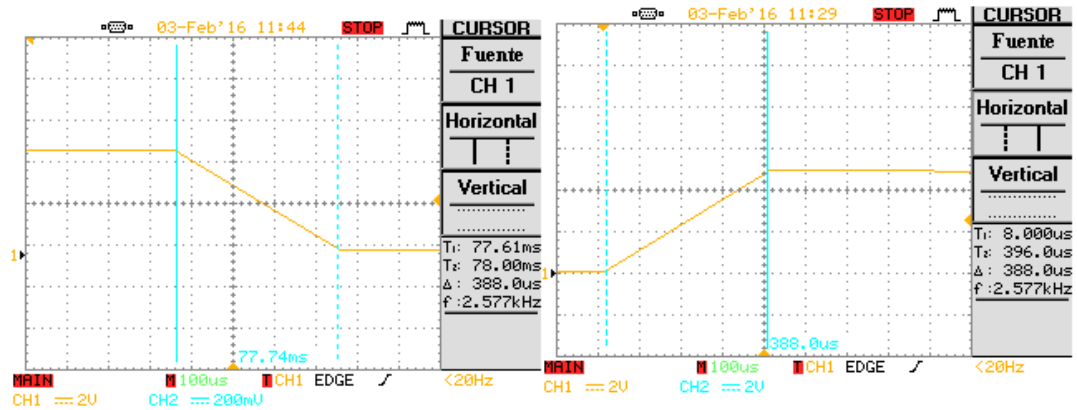


Figura 27. Medición de tiempo del cambio de estado (alto a bajo y bajo a alto)

Se observa que tanto en el cambio alto-bajo como bajo-alto el tiempo es igual a 388 us, sin embargo, en el datasheet del ATmega 64A (pág. 326, tabla “DC Characteristics”) se observa que con la alimentación de 5V, el estado ALTO de salida se toma a partir de 4,2V hacia arriba y el estado BAJO de 0,7V hacia abajo. Por lo tanto en la figura 28 se observa que el cambio de estado se realiza en **276 us**.

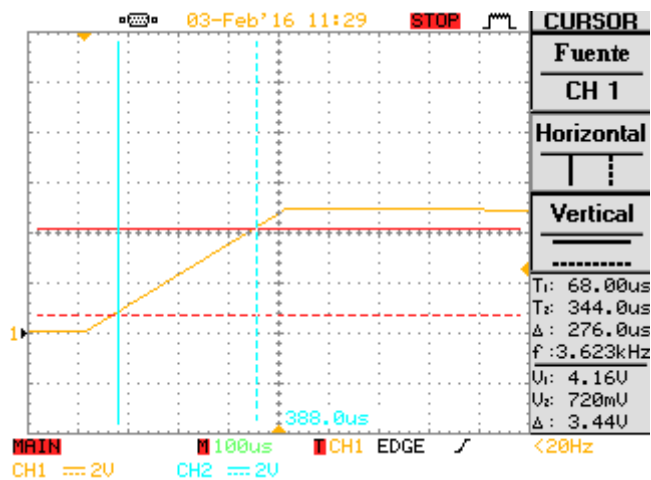


Figura 28. Tiempo real de cambio de estado igual a 276 us

-Medir el máximo retardo entre la primera secuencia y la número diecisiete: es el tiempo total que demora en actualizar todas las secuencias el microcontrolador. Para probar un caso extremo se programaron todas las 17 secuencias con 78 ms (máximo tiempo) y se midió el inicio de la primera y el inicio de la última, figura 29. Puede observarse que la medición es de **800 us**, un valor muy pequeño a comparación de los 78 ms que dura la secuencia.

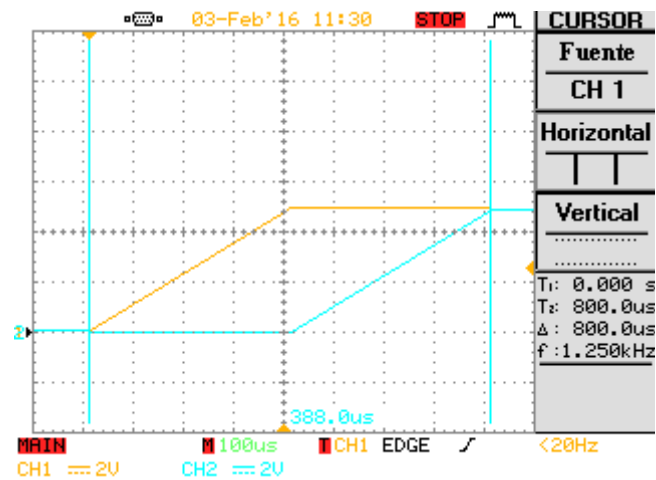


Figura 29. Máximo retardo entre la primera y última secuencia

De las pruebas realizadas a la tarjeta de desarrollo se puede concluir que:

-El rango de frecuencias de trabajo del sistema (6.41Hz y 50Hz) es un rango adecuado para la generación de ondas SSVEP, donde es conocido que entre 5 y 8 Hz son las frecuencias bajas, entre 10 y 15 Hz son frecuencias medias y entre 20 y 50 Hz son frecuencias altas en las cuales los estímulos generan las ondas SSVEP.

-El tiempo de retardo entre la primera y última secuencia (800 us), y el tiempo real de cambio de estado (276 us), no afectan las frecuencias programadas en el teclado puesto que son valores muy bajos, en comparación con la escala de tiempos con la que se trabaja (78 ms).

-Con las pruebas realizadas se concluye, además, que la estimulación se hace de manera correcta, y por tanto, el éxito de las pruebas depende de la capacidad del usuario para generar las ondas SSVEP y la aplicación usada para leer estas ondas.

2.3 Prototipo Físico del Teclado

En esta sección se describen las consideraciones de diseño que se tuvieron para la construcción del prototipo físico del teclado.

2.3.1 Diseño del prototipo de teclado

El diseño está basado en la construcción del teclado que se expone en [Han-Jeong, 2012]. Las características propuestas para el prototipo son:

- Teclas de 2x2 cm.
- Separación entre teclas adyacentes de 2cm.
- Cada tecla de color negro con la abertura correspondiente al carácter.

Las consideraciones de diseño que se tuvieron en cuenta fueron:

- Simplicidad: por sus líneas rectas ligeramente suavizadas y fácil manejo de las piezas.
- Estética: por el diseño formal de los signos numéricos y alfabéticos.
- Sensibilidad: se adapta a diferentes ángulos visuales

El material seleccionado fue poliamida, un material que posee alta resistencia térmica y estabilidad dimensional. En la figura 30 se observa el diseño preliminar del prototipo de teclado. Se pueden observar, además, las diferentes vistas del teclado como de las teclas individuales.

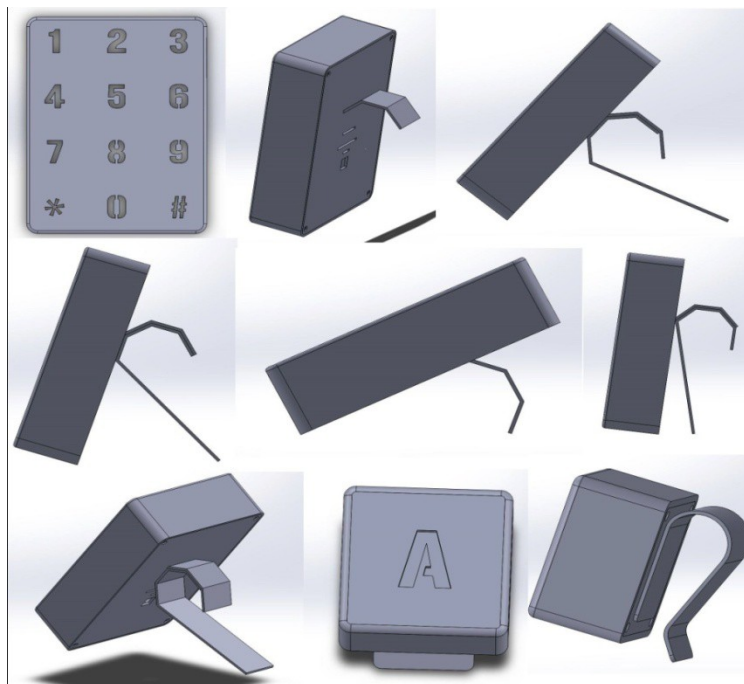


Figura 30. Vistas del diseño del teclado numérico y teclas externas

2.3.2 Resultados

En la figura 31 se observa el resultado de la impresión 3D del prototipo de teclado y de las teclas externas.

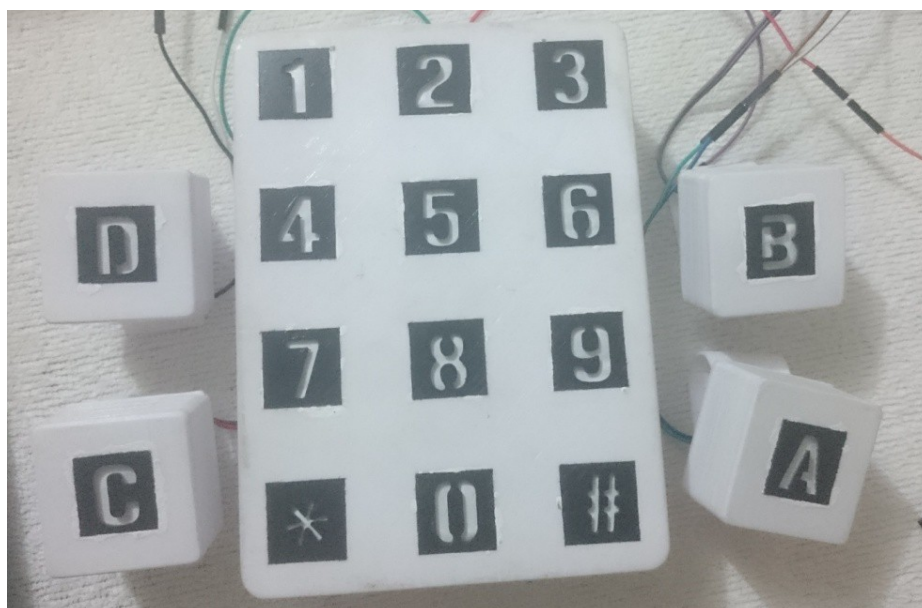


Figura 31. Prototipo de teclado numérico y teclas externas

3. PRUEBAS

3.1 Aplicación Cerebro-Computador

Esta aplicación permite observar los SSVEP generados en un usuario ante los estímulos programados con el sistema propuesto. En esta se programan las 16 teclas y se captura la respuesta cerebral del usuario, validando así el sistema desarrollado.

A continuación se describe la herramienta empleada para la captura de las respuestas, el protocolo de pruebas implementado y los resultados obtenidos.

3.1.1 EMOTIV

Para comprobar la generación de ondas SSVEP por parte de la integración de módulos interfaz de usuario y tarjeta de desarrollo, fue necesario el uso de una aplicación cerebro-computador que permitiera leer las señales cerebrales de un voluntario, la aplicación usada fue EMOTIV Epoc ®.

EMOTIV es un sistema que permite obtener un electroencefalograma de alta resolución, con 14 canales más dos referencias, con un sistema propietario de comunicación entre el EMOTIV y el computador basado en bluetooth de bajo consumo.

En la figura 32 se ve el EMOTIV Epoc, el sistema usado para las pruebas en este proyecto.



Figura 32. EMOTIV Epoc Model 1.0

3.1.2 Protocolo de pruebas

A continuación se establecen los pasos que se siguieron para la validación de la aplicación.

1. Seleccionar un voluntario ^[1] sin problemas mentales conocidos, y ajustar en su cabeza el EMOTIV Epoc. En la imagen 33 se observa la configuración usada en el voluntario y la interfaz de usuario del EMOTIV Epoc.

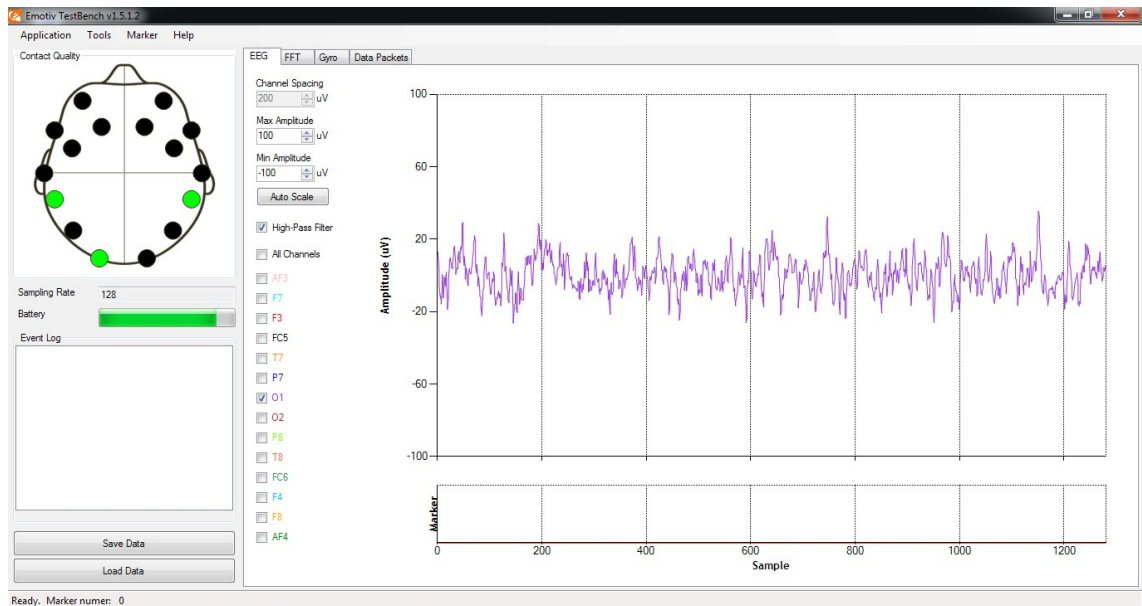


Figura 33. Configuración del EMOTIV Epoc para las pruebas

2. Construir un conjunto de secuencias con las cuales se garantice que se obtendrán SSVEP, es decir, secuencias que tengan frecuencias en el rango de 20 a 50 Hz. En la tabla II se observan las secuencias que se usaron para la aplicación. En las figuras 35 a 42 se observa en la parte izquierda la señal de estímulo usada para cada prueba y en lado derecho la frecuencia obtenida con el EMOTIV Epoc.

[1] En la validación del sistema solo se requiere un voluntario para las pruebas, ya que el propósito del proyecto consiste en construir un dispositivo programable que permita generar las SSVEP de forma sencilla, es decir, la validación del sistema radica en mostrar que el sistema cumple es fiable y fácil de usar.

3. En un ambiente con luz tenue y durante 30 segundos, el voluntario debe observar el estímulo visual evitando movimientos, después de los 30 segundos el voluntario descansa durante 5 s, y posteriormente se repite el procedimiento, una vez por cada uno de los estímulos.

4. Verificar los datos en los electrodos para identificar el que presenta mejores resultados en las frecuencias de interés, usualmente O1 para derechos y O2 para personas zurdas. Y usando MatLab, observar que la frecuencia de estímulo presente picos sobresalientes respecto a las demás frecuencias que no están siendo estimuladas.

En la figura 34 se observa al voluntario que realizó las pruebas con el sistema EMOTIV Epoc, al cual se le aplica un estímulo de 15.15Hz.



Figura 34. Voluntario con el EMOTIV Epoc con un estímulo visual de 15Hz

3.1.3 Pruebas y Resultados

Para las pruebas realizadas se usaron los datos consignados en la tabla 2. En las figuras 35 hasta 50 se observa en el lado izquierdo: la onda usada para estimular (imagen tomada con INSTEK GDS2102-2062) y en el lado derecho la frecuencia a la que se presenta la respuesta cerebral capturada por el EMOTIV Epoc. Cabe mencionar que estas pruebas se realizaron con todas las teclas programadas, es decir, tanto las teclas fijas (1 hasta #) como las móviles (A, B, C y D).

Tabla 2. Estímulos usados en las pruebas del teclado numérico y teclas externas

Tecla	Secuencia	Periodo (ms)	Frecuencia (Hz)
1	33 33	66	15.15
2	2 2 17 17	38	26.32
3	42 42	84	11.90
4	2 2 19 19	42	23.81
5	2 2 21 2 21 21	69	14.45
6	3 3 21 2 21 21	71	14.04
7	50 50	100	10.00
8	2 19 2 19 19	63	15.82
9	63 63	126	7.93
*	3 3 24 24	54	18.52
0	2 2 15 15	34	29.41
#	2 2 16 16	36	27.78
A	3 3 30 30	66	15.15
B	4 4 38 38	84	11.90
C	5 5 45 45	100	10.00
D	6 6 57 57	126	7.93

A continuación el estímulo y la respuesta de la **Tecla #1 de 15.15 Hz.**

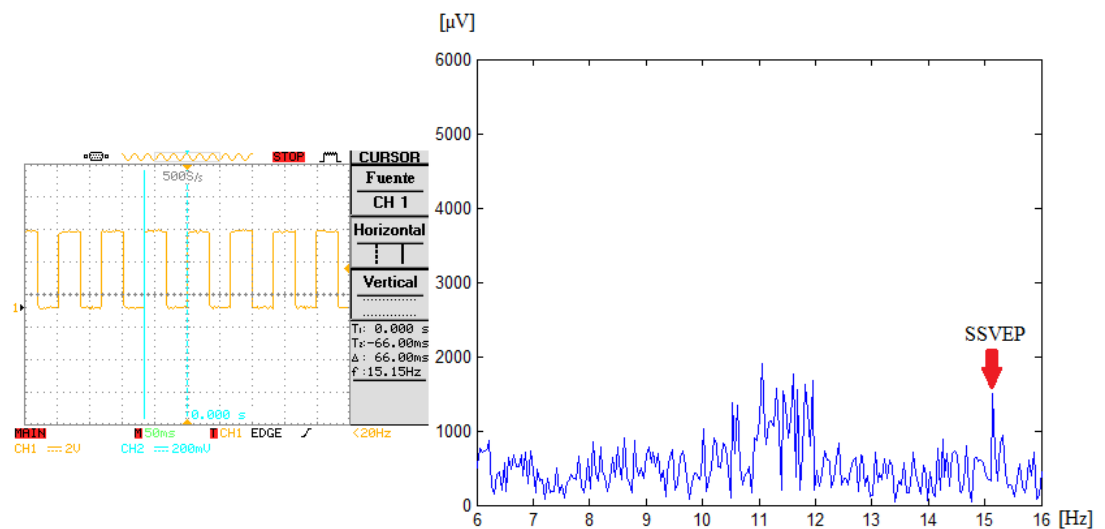


Figura 35. Estímulo a 15.15Hz de onda cuadrada y respuesta a 15.15Hz

Tabla 3. Estimulación de 15.15Hz con onda cuadrada

Secuencia (ms)	33 33
Período deseado (ms)	66
Frecuencia deseada (Hz)	15.15
Período obtenido (ms)	66
Frecuencia obtenida (Hz)	15.15
Frecuencia SSVEP obtenida (Hz)	15.15

Además de observar el pico correspondiente a la frecuencia estimulada (15.15Hz), se observa que entre 11 y 12 Hz hay un pico de altura comparable al de 15.15Hz, este es debido al ruido generado por el movimiento del voluntario.

Tecla #2: 26.32 Hz

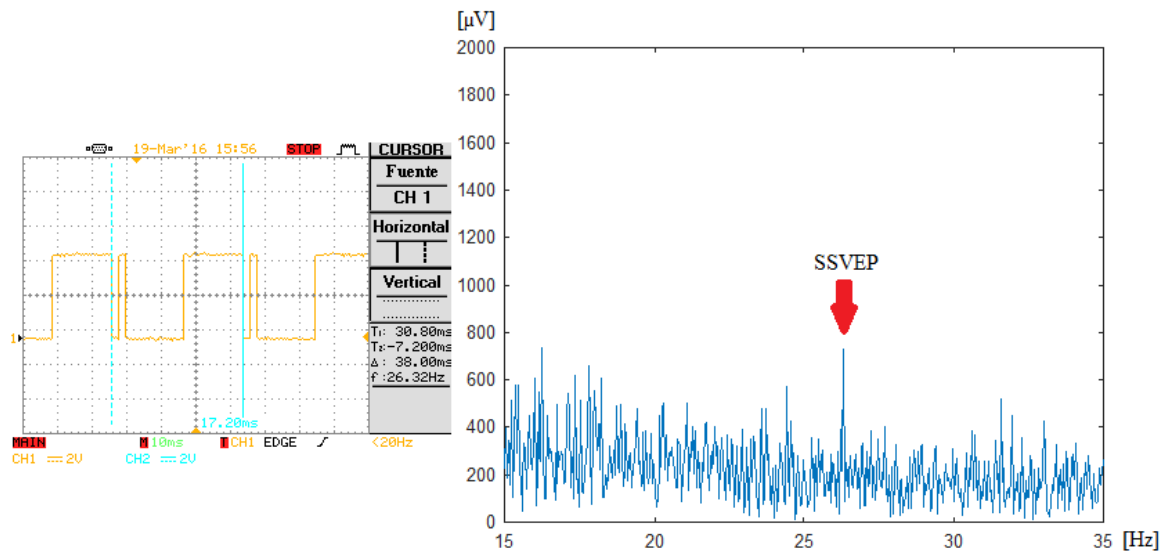


Figura 36. Estimulo a 26.32Hz a doble flanco ascendente y respuesta a 26.32Hz

Tabla 4. Estimulación a 26.32Hz con doble flanco ascendente

Secuencia (ms)	2 2 17 17
Período deseado (ms)	38
Frecuencia deseada (Hz)	26.32
Período obtenido (ms)	38
Frecuencia obtenida (Hz)	26.32
Frecuencia SSVEP obtenida (Hz)	26.32

Prueba realizada en la tecla '2' del teclado numérico, con respuesta esperada y obtenida en 26.32Hz.

Tecla #3: 11.90 Hz

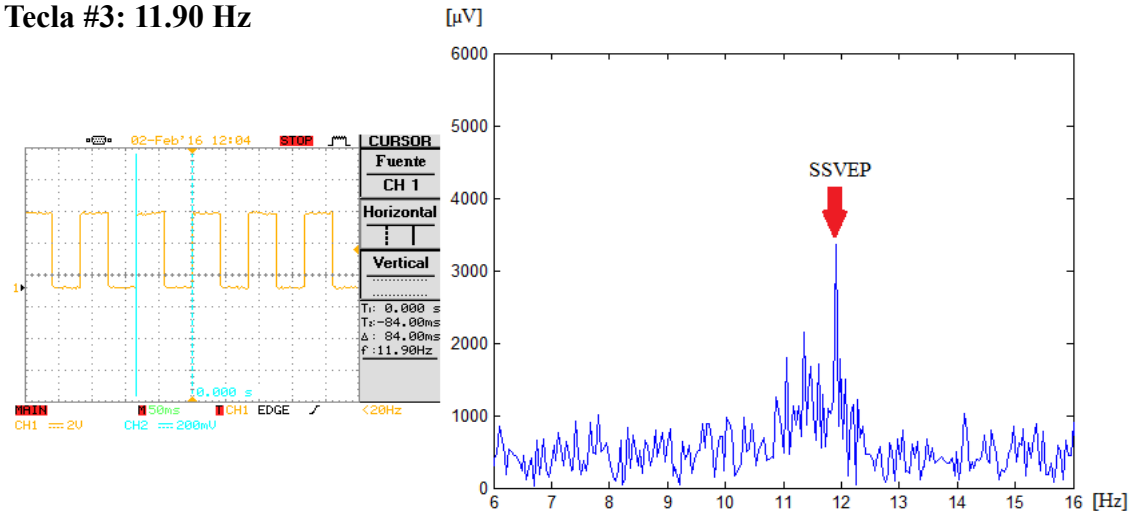


Figura 37. Estímulo a 11.90Hz de onda cuadrada y respuesta a 11.90Hz

Tabla 5. Estimulación de 11.90Hz con onda cuadrada

Secuencia (ms)	42 42
Período deseado (ms)	84
Frecuencia deseada (Hz)	11.90
Período obtenido (ms)	84
Frecuencia obtenida (Hz)	11.90
Frecuencia SSVEP obtenida (Hz)	11.90

En la tabla 5 se observa que la respuesta obtenida con el EMOTIV Epoc corresponde efectivamente a la frecuencia estimulada.

Tecla #4: 26.32 Hz

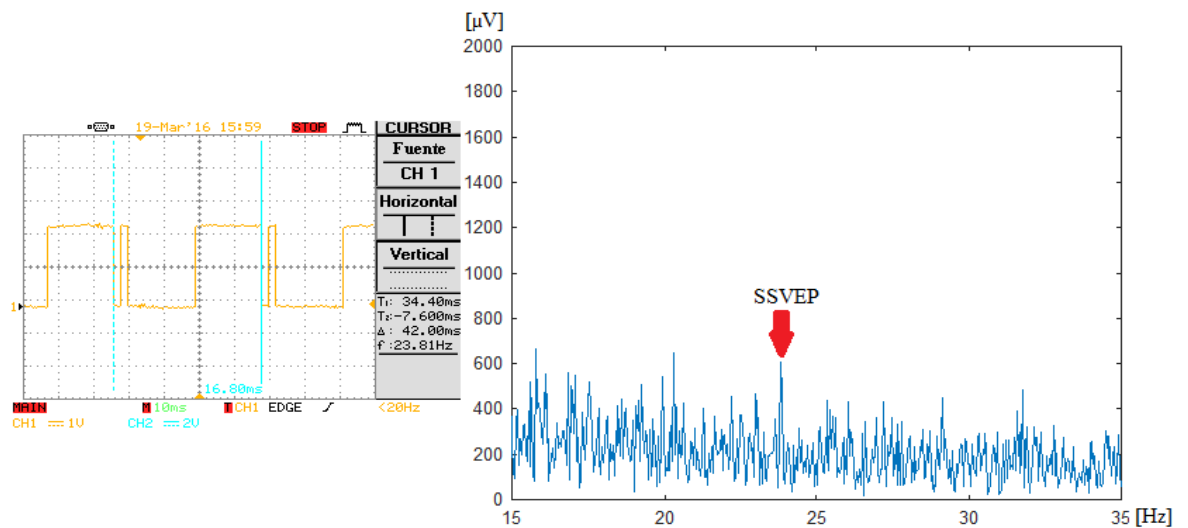


Figura 38. Estímulo a 23.81Hz a doble flanco ascendente y respuesta a 23.81Hz

Tabla 6. Estimulación a 23.81Hz con doble flanco ascendente

Secuencia (ms)	2 2 19 19
Período deseado (ms)	42
Frecuencia deseada (Hz)	26.32
Período obtenido (ms)	42
Frecuencia obtenida (Hz)	23.81
Frecuencia SSVEP obtenida (Hz)	23.81

Prueba realizada en la tecla '4' del teclado, con respuesta esperada y obtenida en 23.81Hz.

Tecla #5: 14.45 Hz

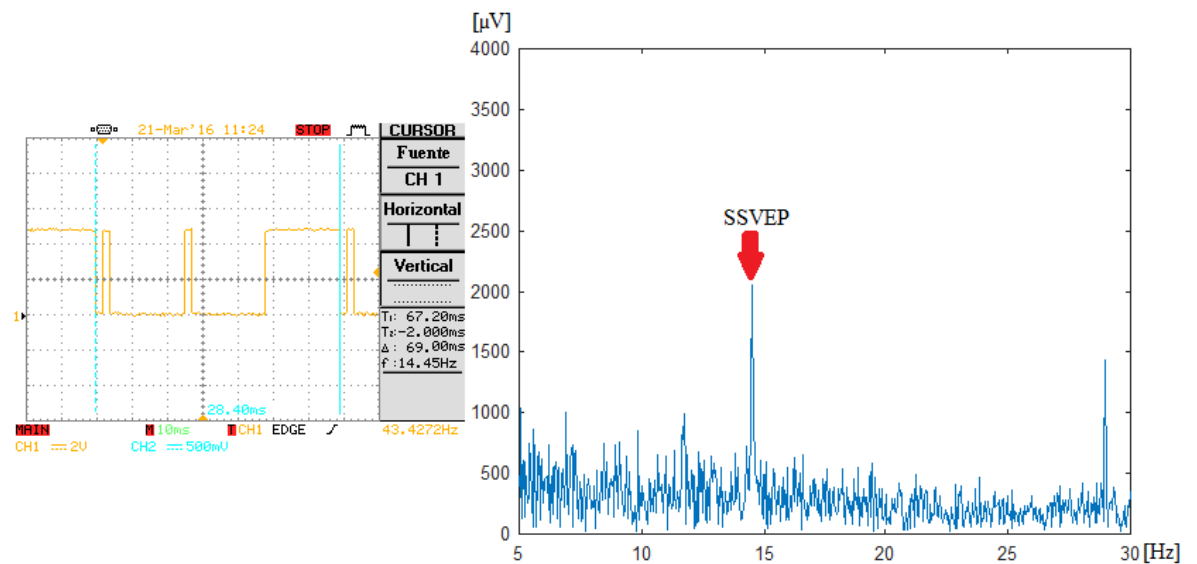


Figura 39. Estímulo a 14.45Hz a triple flanco ascendente y respuesta a 14.45Hz

Tabla 7. Estimulación a 14.45Hz con triple flanco ascendente

Secuencia (ms)	2 2 21 2 21 21
Período deseado (ms)	69
Frecuencia deseada (Hz)	14.45
Período obtenido (ms)	69
Frecuencia obtenida (Hz)	14.45
Frecuencia SSVEP obtenida (Hz)	14.45

Prueba realizada en la tecla '5' del teclado, con respuesta esperada y obtenida en 14.45Hz.

Tecla #6: 14.04 Hz

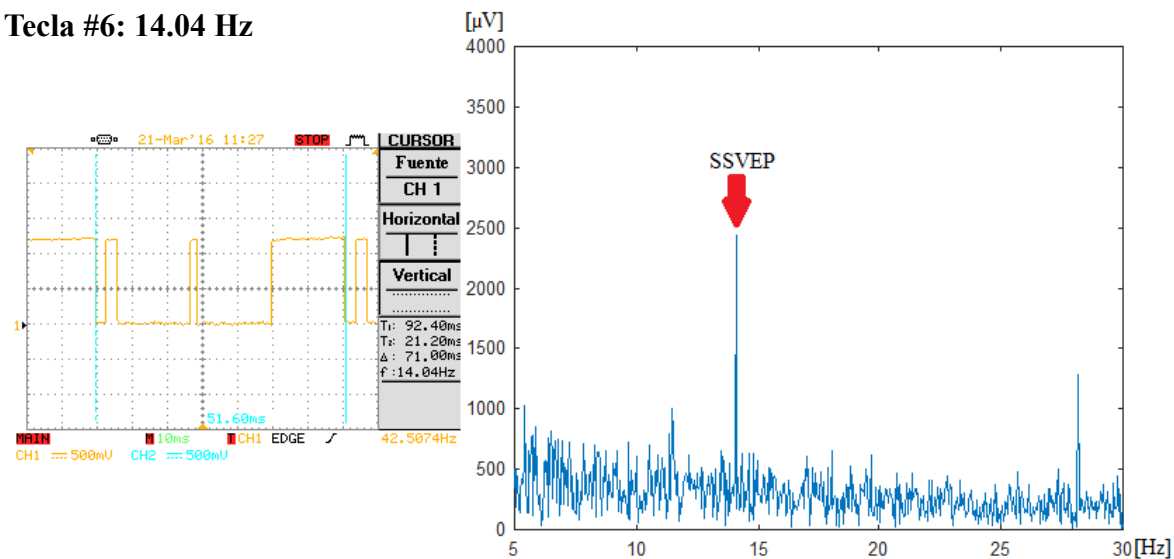


Figura 40. Estimulo a 14.04Hz a triple flanco ascendente y respuesta a 14.04Hz

Tabla 8. Estimulación a 14.04Hz con triple flanco ascendente

Secuencia (ms)	3 3 21 2 21 21
Período deseado (ms)	71
Frecuencia deseada (Hz)	14.04
Período obtenido (ms)	71
Frecuencia obtenida (Hz)	14.04
Frecuencia SSVEP obtenida (Hz)	14.04

Prueba realizada en la tecla '6' del teclado numérico, con respuesta esperada y obtenida en 14.04Hz.

Tecla #7: 10.00 Hz

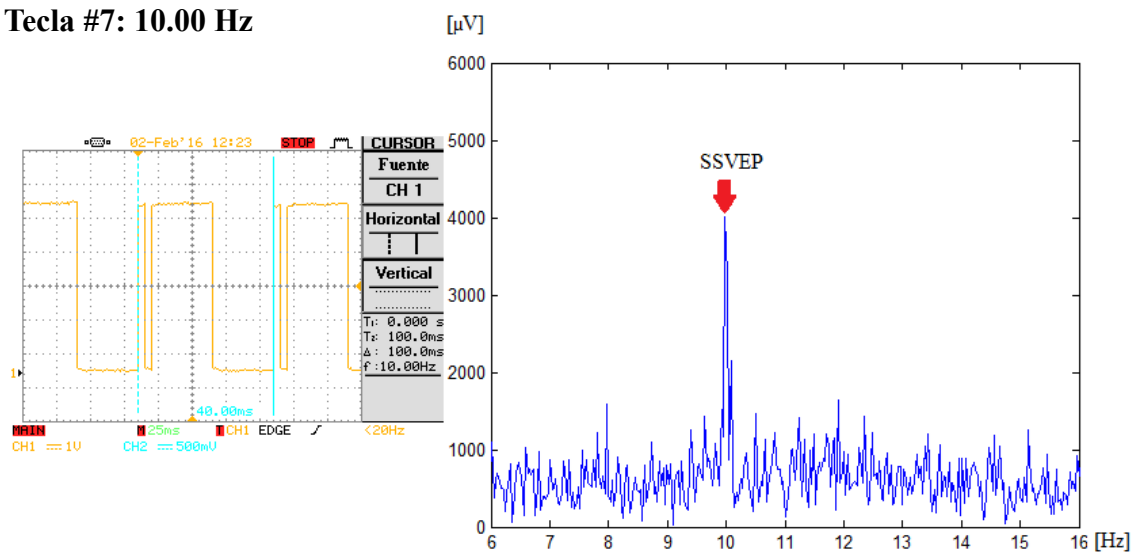


Figura 41. Estimulo a 10Hz de doble flanco y respuesta a 10.10Hz

Tabla 9. Estimulación de 10Hz con onda de doble flanco ascendente

Secuencia (ms)	5 5 45 45
Período deseado (ms)	100
Frecuencia deseada (Hz)	10
Período obtenido (ms)	100
Frecuencia obtenida (Hz)	10
Frecuencia SSVEP obtenida (Hz)	9.99

Prueba realizada en la tecla ‘7’ del teclado numérico, con respuesta esperada y obtenida en 10.00 Hz.

Tecla #8: 15.82 Hz

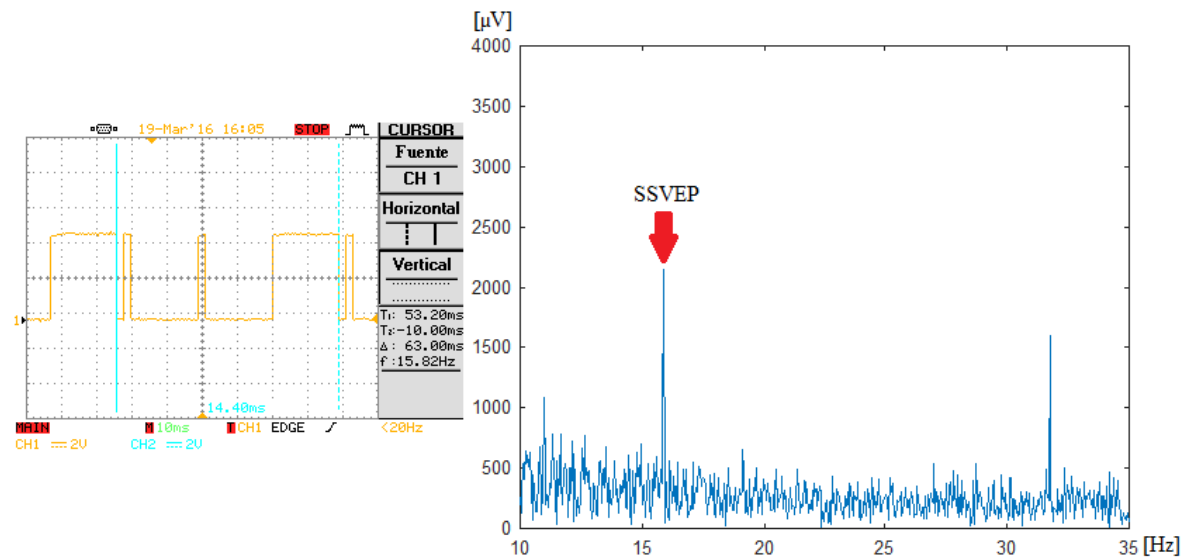


Figura 42. Estímulo a 15.82Hz a triple flanco ascendente y respuesta a 15.82Hz

Tabla 10. Estimulación a 15.82Hz con triple flanco ascendente

Secuencia (ms)	2 19 2 19 19
Período deseado (ms)	63
Frecuencia deseada (Hz)	15.82
Período obtenido (ms)	63
Frecuencia obtenida (Hz)	15.82
Frecuencia SSVEP obtenida (Hz)	15.82

Prueba realizada en la tecla ‘8’ del teclado numérico, con respuesta esperada y obtenida en 15.82Hz.

Tecla #9: 7.93 Hz

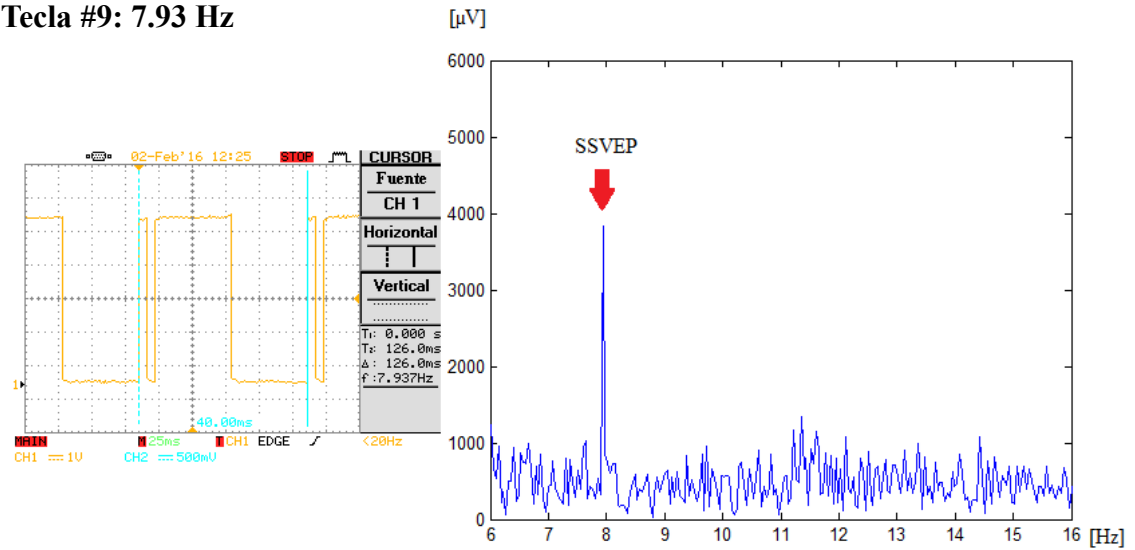


Figura 43. Estimulo a 7.93Hz de doble flanco y respuesta a 7.93Hz

Tabla 11. Estimulación de 7.93Hz con onda de doble flanco ascendente

Secuencia (ms)	6 6 57 57
Período deseado (ms)	126
Frecuencia deseada (Hz)	7.93
Período obtenido (ms)	126
Frecuencia obtenida (Hz)	7.93
Frecuencia SSVEP obtenida (Hz)	7.93

Prueba realizada en la tecla '9' del teclado numérico, con respuesta esperada y obtenida en 7.93 Hz.

Tecla *: 18.52 Hz

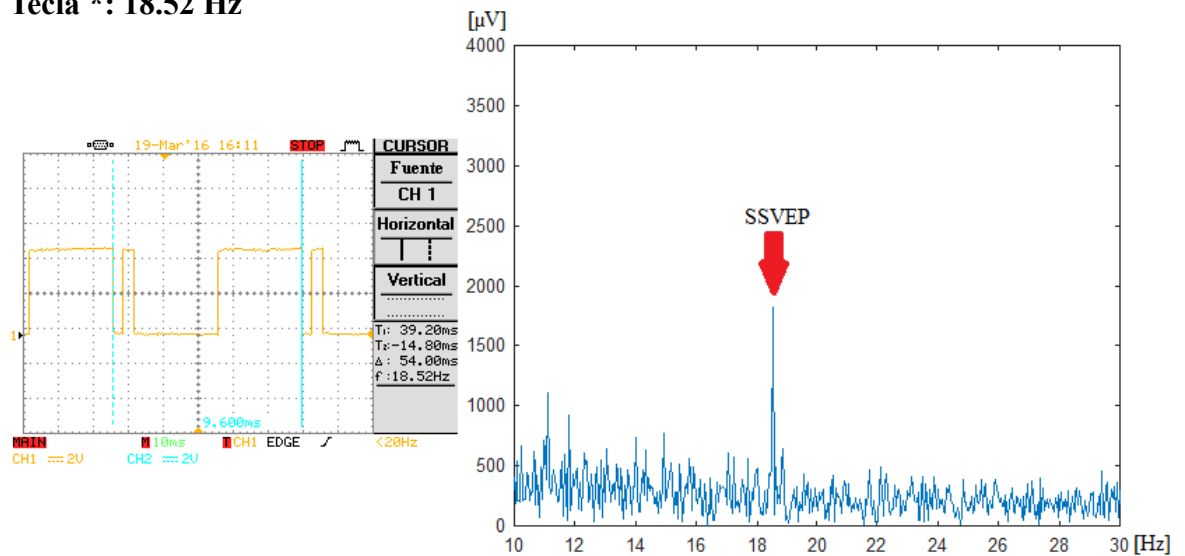


Figura 44. Estimulo a 18.52Hz a doble flanco ascendente y respuesta a 18.52Hz

Tabla 12. Estimulación a 18.52Hz con doble flanco ascendente

Secuencia (ms)	3 3 24 24
Período deseado (ms)	54
Frecuencia deseada (Hz)	18.52
Período obtenido (ms)	54
Frecuencia obtenida (Hz)	18.52
Frecuencia SSVEP obtenida (Hz)	18.52

Prueba realizada en la tecla ‘*’ del teclado, con respuesta esperada y obtenida en 18.52Hz.

Tecla #0: 29.41Hz

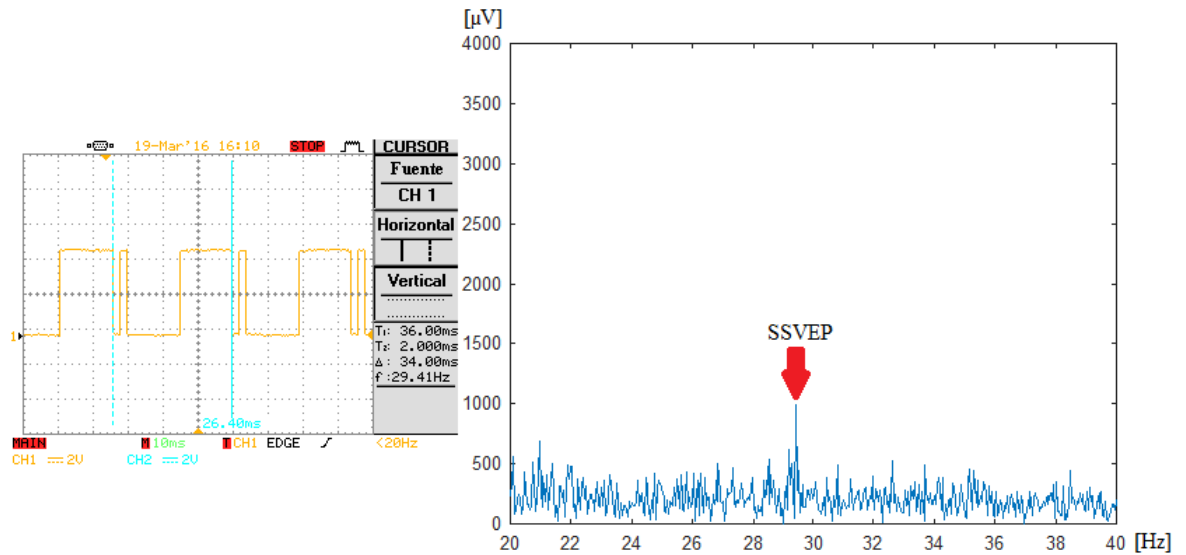


Figura 45. Estímulo a 29.41Hz a doble flanco ascendente y respuesta a 29.41Hz

Tabla 13. Estimulación a 29.41Hz con doble flanco ascendente

Secuencia (ms)	2 2 15 15
Período deseado (ms)	34
Frecuencia deseada (Hz)	29.41
Período obtenido (ms)	34
Frecuencia obtenida (Hz)	29.41
Frecuencia SSVEP obtenida (Hz)	29.41

Prueba realizada en la tecla ‘0’ del teclado, con respuesta esperada y obtenida en 29.41Hz.

Tecla #: 27.78 Hz

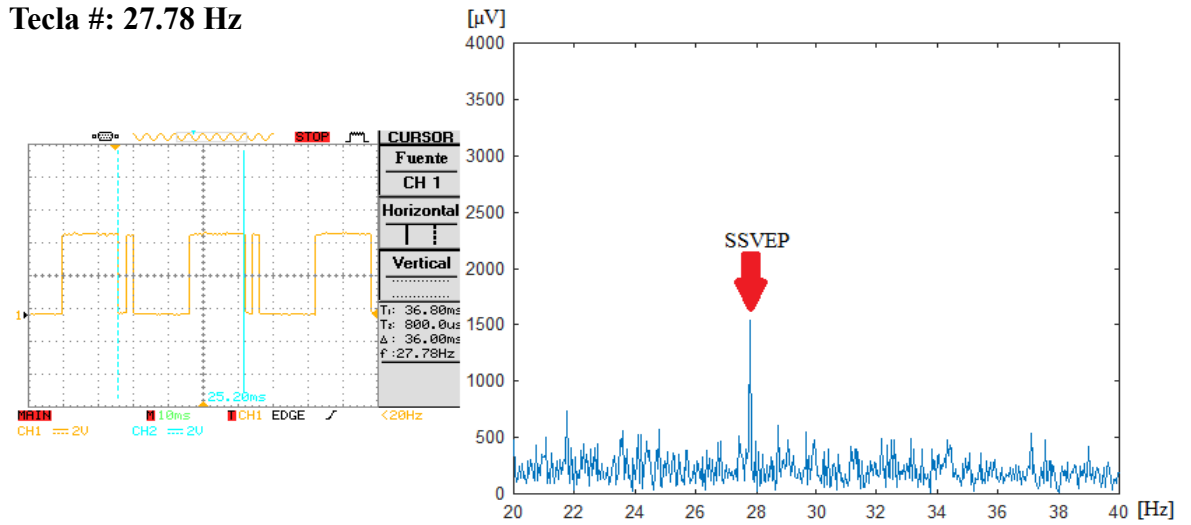


Figura 46. Estimulo a 27.78Hz a doble flanco ascendente y respuesta a 27.78Hz

Tabla 14. Estimulación a 27.78Hz con doble flanco ascendente

Secuencia (ms)	2 2 16 16
Período deseado (ms)	36
Frecuencia deseada (Hz)	27.78
Período obtenido (ms)	36
Frecuencia obtenida (Hz)	27.78
Frecuencia SSVEP obtenida (Hz)	27.78

Prueba realizada en la tecla '#' del teclado, con respuesta esperada y obtenida en 27.78Hz.

Tecla A: 15.15 Hz

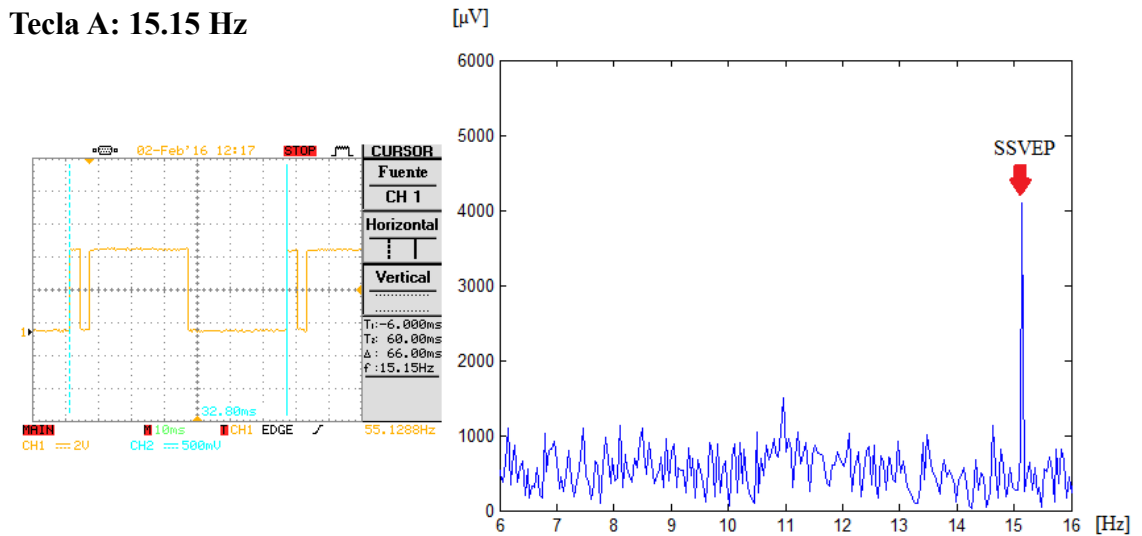


Figura 47. Estimulo a 15.15Hz de doble flanco y respuesta a 15.15Hz

Tabla 15. Estimulación de 15.15Hz con onda de doble flanco ascendente

Secuencia (ms)	3 3 30 30
Período deseado (ms)	66
Frecuencia deseada (Hz)	15.15
Período obtenido (ms)	66
Frecuencia obtenida (Hz)	15.15
Frecuencia SSVEP obtenida (Hz)	15.15

Al igual que en las pruebas anteriores, se obtuvo un pico en la frecuencia estimulada.

Tecla B: 11.90 Hz

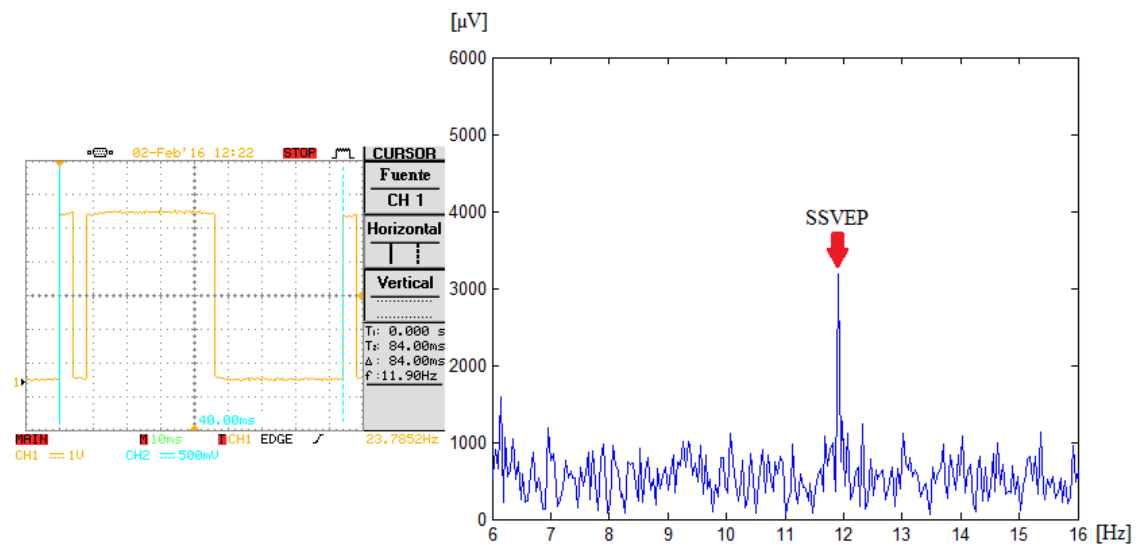


Figura 48. Estímulo a 11.90Hz de doble flanco y respuesta a 11.90Hz

Tabla 16. Estimulación de 11.90Hz con onda de doble flanco ascendente

Secuencia (ms)	4 4 38 38
Período deseado (ms)	84
Frecuencia deseada (Hz)	11.90
Período obtenido (ms)	84
Frecuencia obtenida (Hz)	11.90
Frecuencia SSVEP obtenida (Hz)	11.90

Tecla C: 10.10 Hz

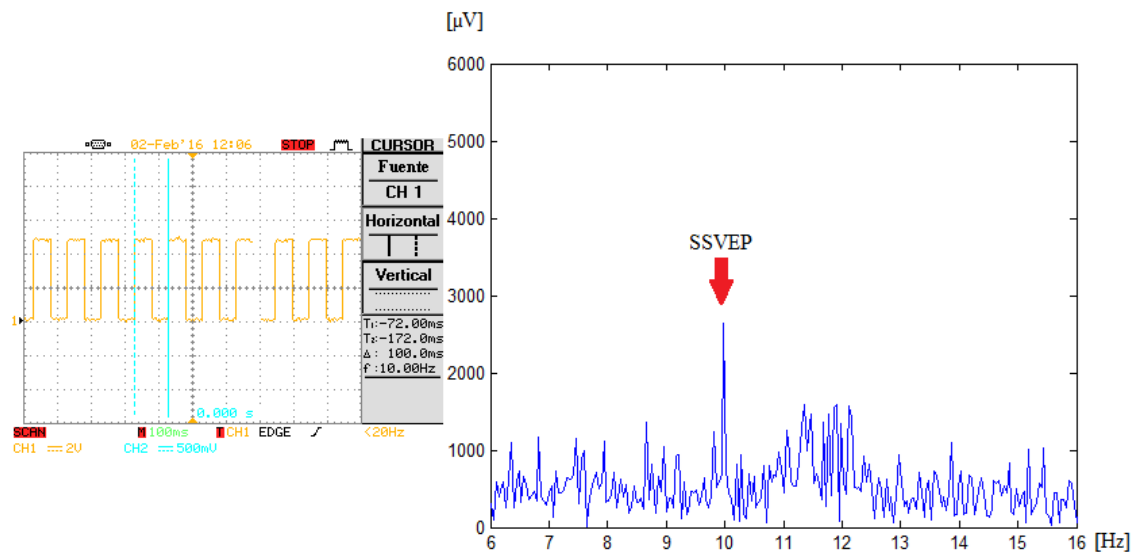


Figura 49. Estímulo a 10Hz de onda cuadrada y respuesta a 9.99Hz

Tabla 17. Estimulación de 10Hz con onda cuadrada

Tecla D: 7.93 Hz

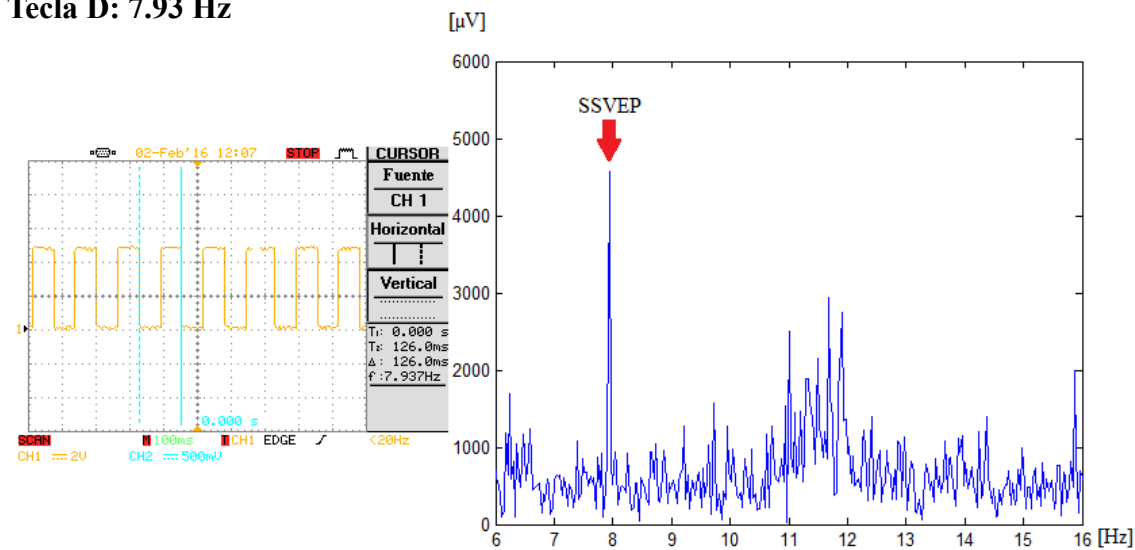


Figura 50. Estímulo a 7.93Hz de onda cuadrada y respuesta a 7.93Hz

Tabla 18. Estimulación de 7.93Hz con onda cuadrada

Secuencia (ms)	63 63
Período deseado (ms)	126
Frecuencia deseada (Hz)	7.93
Período obtenido (ms)	126
Frecuencia obtenida (Hz)	7.93
Frecuencia SSVEP obtenida (Hz)	7.93

Dado que el proyecto consiste en diseñar un sistema con el que se puedan generar los estímulos visuales, la validación de este bastaría con observar que el sistema si es capaz de generar estímulos en el rango de trabajo adecuado (20-50 Hz). Según los resultados obtenidos se observa un error relativo porcentual de 0% en la generación de la frecuencia propuesta de trabajo. En las figuras 35 hasta la 50 se exponen las ondas generadas por el sistema, aquí se pueden ver las ondas medidas en el osciloscopio INSTEK GDS2102-2062, y de esta forma confirmar el funcionamiento del proyecto.

4. CONCLUSIONES

-Se diseñó un prototipo generador de señales SSVEP configurado como un teclado alfa numérico que cuenta con 16 teclas con frecuencias de luces programables, 12 de estas teclas son fijas en el teclado, mientras que las 4 teclas restantes pueden ser ubicadas de forma libre, se tiene también una salida TTL adicional programable que puede servir como guía para comparar con otras señales o reloj de todo el sistema. Para robustecer el diseño y garantizar la generación de SSVEP se tuvieron en cuenta parámetros como el estímulo (color, potencia, rango de frecuencias), su estructura y organización (separación y cantidad) y la inclinación de la fuente [Danhua, 2010].

- El software desarrollado ***QFOCUS 1.0*** presenta funcionalidades como la posibilidad de programar todas o algunas (según desee el usuario) de las 16 teclas y una salida TTL hasta con 10 tiempos entre 1 y 77 ms, con retardos o sin ellos, con la posibilidad de escoger el estado inicial o habilitación de teclas y una gráfica de las secuencias programadas. También permite cargar archivos de texto con la programación de las teclas. Cuenta con ventajas como la facilidad de uso y la libertad de cambios para futuros trabajos y realizado en un lenguaje multiplataforma que garantiza su portabilidad entre diferentes sistemas operativos.

- El hardware elaborado garantiza el cumplimiento de los tiempos de las posibles 17 secuencias simultáneas, también el almacenamiento tanto en ram como en eeprom de cada uno de los datos, la comunicación con la interfaz y la rapidez de operación en la ejecución de cada instrucción. Entre sus ventajas está el funcionamiento a través de comunicación USB (mediante una conversión usb-serial), la operación sin necesidad la interfaz (ejecuta las ultimas secuencias guardadas) una vez tenga 5V de alimentación, la posibilidad de cambios para futuros trabajos y la portabilidad entre sistemas operativos con la instalación del driver pl2303x correspondiente.

-De las pruebas realizadas se logro validar que el sistema es capaz de generar frecuencias en el rango de trabajo deseado (20-50 Hz) con error relativo nulo.

-En conjunto, el prototipo de sistema generador de SSVEP es portable, de fácil manejo para el usuario y multiplataforma. Las pruebas mostraron respuesta a la frecuencia del estímulo, por tanto se concluye que el sistema generador de SSVEP aquí documentado, puede ser usado para una aplicación BCI con el fin de usar sus 16 salidas con estímulo visual más una salida TTL para ámbitos como la investigación (generar diferentes ondas programables y observar la respuesta en un voluntario) o la aplicación (ejecutar determinadas tareas a partir del estímulo observado).

-En cuanto a las fuentes de error en el sistema, se observó que a pesar de que estas existen, tales como las cifras significativas usadas en cada una de las mediciones, el error que se induce en los instrumentos de medición (Emotiv Epoc, osciloscopio, etc.), el tiempo de retardo generado por la atención de interrupciones por parte de cada una de las 17 teclas y el tiempo de cambio de estado en el microcontrolador, la suma de estos no es significativa debido al poco error generado en las pruebas.

5. TRABAJOS FUTUROS

Respecto al modo de programación se proponen dos trabajos:

1. Programación de las secuencias desde una aplicación en Android, usando un teléfono celular o una Tablet y adecuando un módulo bluetooth a la tarjeta. Estas propuestas son útiles, ya que no se dependería de un computador para programar las secuencias y tampoco de un puerto serial de comunicación.
2. Programación remota desde una página web con servidor: Con esta opción no se dependería de la instalación de librerías y paquetes adicionales para que funcione. La idea sería implementar una solución web similar a la interfaz con su respectiva validación de datos que sea leída por una aplicación cliente-servidor y envíe los datos por el puerto serial.

Respecto a la funcionalidad se propone:

1. Una aplicación ICC, la cual después de programar todas las secuencias, sea capaz de identificar que tecla se está visualizando y realizar una tarea predeterminada (abrir una página web, realizar una llamada, activar un electrodoméstico, etc.). Con el fin de ayudar en tareas cotidianas a alguna persona con discapacidad. Este es un proceso viable, debido a que las pruebas muestran que el sistema conjunto funciona en la generación de SSVEP y con exactitud.

6. BIBLIOGRAFÍA

[Amiri, 2013] Setare Amiri¹, Ahmed Rabbi¹, Leila Azinfar¹, “A Review of P300, SSVEP, and Hybrid P300/SSVEP Brain-Computer Interface System”, 2013.

[Caicedo, 2013] Eduardo Caicedo Bravo, Jaiber Evlio Cardona “Plataforma Robótica Para La Movilidad Asistida De Personas Con Discapacidad”, Proyecto De Investigación, Universidad Del Valle, Paginas 4-6, 2013.

[Danhua, 2010] [Danhua Zhu](#), [Jordi Bieger](#), [Gary Garcia Molina](#), “A Survey of Stimulation Methods Used in SSVEP-Based BCIs”, Computational Intelligence and Neuroscience Volume 2010, 2010.

[Han-Jeong, 2012] [Han-Jeong Hwang](#), [Young-Jin Jung](#), “Development Of An SSVEP-Based BCI Spelling System Adopting A QWERTY-Style LED Keyboard”, [Revista Neuroscience Methods](#), vol. 208, pp. 59-65, 2012.

[Hwang, 2013] Han-Jeong Hwang, Jeong-Hwan Lim, Jun-Hak Lee, Chang-Hwan Im, “Implementation Of Mental Spelling System Based On Steady-State Visual Evoked Potential (Ssvep)”, International Winter Worksh Op On Brain-Computer Interface (Bci), 2013.

[Kimura, 2013] Yosuke Kimura, Toshihisa Tanaka, “Ssvep-Based Brain-Computer Interfaces Using Fsk-Modulated Visual Stimuli”, IEEE Transactions On Biomedical Engineering, Vol. 60, No. 10, October 2013.

[Luzheng, 2013] Luzheng Bi, Yun Li, Kejie, And Xin-An Fan, “A New Ssvep Brain-Computer Interface Based On A Head Up Display”, Proceedings ICME, International Conference On Complex Medical Engineering, 2013.

[Nawrocka, 2013] Nawrocka Aghata, Karolina Holewa, “Brain - Computer Interface Based On Steady - State Visual Evoked Potentials (SSVEP)”, 14th International Carpathian Control Conference (ICCC), 2013.

[Pfurtscheller, 2010] Pfurtscheller G., Allison B. Z., Bauernfeind G., Brunner C., “The hybrid BCI. Front”, Neurosci. Vol 12, pp. 54-57, 2010.

[Resalat, 2012] [S. N. Resalat](#), [V. Saba](#); [F. Afdideh](#), “High-speed SSVEP-based BCI: study of various frequency pairs and inter-sources distances”, Physiological Measurement, vol. 27, pp. 61-71, 2012.

[Wolpaw, 2002] Wolpaw J. R., Birbaumer N., McFarland D.J., Pfurtscheller G., and Vaughan T. M., “Brain-computer interfaces for communication and control”, Clinical Neurophysiology, vol 113, pp. 767–791, 2002.

[[Xiaorong](#), 2003] [Xiaorong Gao](#), [Dingfeng Xu](#), [Ming Cheng](#) ; [Shangkai Gao](#), “A BCI-based environmental controller for the motion-disabled”, [IEEE Transactions on Neural Systems and Rehabilitation Engineering](#), Vol. 11, pp. , 2003.