



**UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL**

FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO

TESIS DE GRADO

Previo a la obtención del título de:

**INGENIERO EN TELECOMUNICACIONES
Mención en gestión empresarial**

TEMA:

**DESARROLLO E IMPLEMENTACIÓN DE UNA INTERFAZ DE COMUNICACIÓN QUE
PERMITA LA INTERACCIÓN ENTRE UN USUARIO Y LAS SEÑALES EMITIDAS POR SUS
ONDAS CEREBRALES USANDO UN DISPOSITIVO DE EEG DE NEUROSKY PARA
CONTROLAR PERIFÉRICOS ELECTRÓNICOS.**

Autor

Angel D. Yaguana Hernandez

Director

Ing. Carlos Romero Rosero

GUAYAQUIL – ECUADOR

Mayo 2014



**UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL**

FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO

**INGENIERÍA EN TELECOMUNICACIONES CON
MENCIÓN EN GESTIÓN EMPRESARIAL**

CERTIFICACIÓN

Siguiendo las técnicas metodológicas de Investigación Científica y la implementación de NORMAS APA, CERTIFICO que el presente trabajo titulado "Desarrollo e implementación de una interfaz de comunicación que permita la interacción entre un usuario y las señales emitidas por sus ondas cerebrales usando un dispositivo de EEG de NeuroSky para controlar periféricos electrónicos.", fue realizado en su totalidad por el Sr. Angel D. Yaguana Hernández, como requerimiento parcial para la obtención del título de Ingeniero en Telecomunicaciones con Mención en Gestión Empresarial

Guayaquil, Mayo de 2014

DIRECTOR DE TESIS

Ing. Carlos Romero Rosero.

REVISOR:

Ing. Pedro Tutivén López, Msc.

REVISOR:

Ing. Judith Gálvez Soto, Msc.

DIRECTOR DE CARRERA

Ing. Armando Heras Sánchez, Msc.



**UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL**

FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO

**INGENIERÍA EN TELECOMUNICACIONES CON
MENCION EN GESTIÓN EMPRESARIAL**

DECLARACIÓN DE RESPONSABILIDAD

Yo, ANGEL DAVID YAGUANA HERNÁNDEZ

DECLARO QUE:

El proyecto de grado denominado “Desarrollo e implementación de una interfaz de comunicación que permita la interacción entre un usuario y las señales emitidas por sus ondas cerebrales usando un dispositivo de EEG de NeuroSky para controlar periféricos electrónicos”, declaro que ha sido desarrollada en base a una investigación exhaustiva, respetando derechos intelectuales de terceros conforme las citas que constan al pie de las páginas correspondientes. Consecuentemente este trabajo es de nuestra total autoría.

En virtud de esta declaración, nos responsabilizamos del contenido, veracidad y alcance científico de la tesis del Grado Académico en mención.

Guayaquil, Mayo de 2014

Angel D. Yaguana Hernández
C.I# 1719994095



**UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL**

FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO

**INGENIERIA EN TELECOMUNICACIONES CON
MENCION EN GESTIÓN EMPRESARIAL**

AUTORIZACIÓN

Yo, ANGEL DAVID YAGUANA HERNÁNDEZ

Autorizo a la Universidad Católica de Santiago de Guayaquil, la publicación, en la biblioteca de la institución la Tesis de Ingeniería titulada: "Desarrollo e implementación de una interfaz de comunicación que permita la interacción entre un usuario y las señales emitidas por sus ondas cerebrales usando un dispositivo de EEG de NeuroSky para controlar periféricos electrónicos", cuyo contenido, ideas y criterios son de mi exclusiva responsabilidad y total autoría.

Guayaquil, Mayo de 2014

EL AUTOR

Angel D. Yaguana Hernández

C.I# 1719994095

AGRADECIMIENTO

A mis padres por haberme apoyado durante toda mi vida con cada reto y meta que me he propuesto; a mi esposa por iluminarme en cada momento difícil y apoyarme en todos y cada uno de los proyectos que he trabajado y más aún por su disposición a los que vendrán; a mi hijo por haber hecho de mi vida una gran aventura de superación, llena de alegrías y hazañas.

DEDICATORIA

Dedico este proyecto primeramente a Dios quien me dio la fuerza y la salud mental suficiente para afrontar con éxito todo este proceso académico, a mi querida madre Mariana Hernández, a mi ejemplo y guía e ética y servicio, mi padre, Leonzo Angel Yaguana, a mi ejemplar hermana Diana Yaguana, a mi amada esposa Sophia Álava y a mi lindo bebé Nicolás Jericó Yaguana Álava.

Así mismo quiero expresar con énfasis mi admiración a mis mentores y maestros, Ing. Luis Córdova Rivadeneira, Msc., quien formó mi inclinación por el desarrollo de tecnología y el estudio de la microelectrónica, al Ing. Eduardo Mendoza Merchán, Msc., quien me disciplino como investigador y desarrollador electrónico, al Ing. Roberto García Vacacela, Msc., quien me guio con sabiduría en el desarrollo de esta obra, a los Ingenieros Fabián Ortiz y Diana Chávez, amigos leales e inseparables quienes siempre me han apoyado en cada reto profesional que me he propuesto alcanzar y sobre todo a mi ejemplo de superación en el campo científico e investigativo, el Ing. Walter Mera Ortiz, PhD.

Doy mis gracias y mediante este proyecto, dedico todo mi esfuerzo de compensación a todas las demás personas de mi querida universidad que hicieron posible mi proceso de formación como persona íntegra, ética y honorable, aquellos que me forjaron como investigador y docente, como profesional desarrollador de tecnología, apoyándome en todo momento y necesidad. Para todos ellos mis más sinceros agradecimientos por su apoyo y compañía.

**"Pensé que podría retomar el viejo sueño de la humanidad:
controlar cosas con el pensamiento" (Millán, 2011).**

ÍNDICE GENERAL

PORTADA.....	III
CERTIFICACIÓN	IVIV
DECLARACIÓN DE RESPONSABILIDAD.....	III
AUTORIZACIÓN.....	IV
AGRADECIMIENTO.....	V
DEDICATORIA.....	VI
ÍNDICE GENERAL.....	VIII
RESUMEN	XIV
ABSTRACT.....	XV
CAPÍTULO 1	
DESCRIPCIÓN Y ASPECTOS TEÓRICOS.....	16
1.1. Introducción.....	16
1.2. Antecedentes.....	18
1.3. Problema	19
1.4. Objetivos	19
1.4.1. Objetivo general	19
1.4.2. Objetivos específicos	20
1.5. Hipótesis.....	20
1.6. Justificación	20
1.7. Limitaciones del estudio.....	20
1.8. Técnicas y métodos empleados en la investigación	21
CAPÍTULO 2	
MARCO TEÓRICO REFERENCIAL.....	24
2.1. Marco teórico	24
2.1.1. Sistemas de adquisición de datos de neuroseñales	24
2.1.2. Métodos de medición mediante EEG	25
2.1.3. Comportamiento de la señal EEG.....	26
2.1.4. Captación de las señales EEG	29
2.1.5. Sistemas BCI y BMI basados en EEG	31
2.1.6. Modelo funcional genérico de un sistema BCI.....	33
2.1.7. El sistema internacional 10-20 EEG	34
CAPÍTULO 3	
ESTADO DEL ARTE Y ANALISIS DE LOS SISTEMAS BCI	35
3.1 Estado de Arte	35
3.2. Análisis de los antecedentes y requerimientos previos al desarrollo conceptual.....	37
3.3. Identificación de requerimientos del cliente.....	37
3.4. Parámetros de proceso de diseño electrónico del prototipo.....	38
3.5. Generación de conceptos.....	38
3.6. Descomposición funcional.....	38
3.7. Exploración Sistematizada.....	39
3.8. Obtención de materiales primordiales.....	39
3.9. Descarga de información relevante al producto	39
3.10. Descarga de Kit de herramientas de desarrollo de NeuroSky	39

3.11. Descarga e instalación de herramientas adicionales secundarias.....	40
3.12. Dispositivo de EEG ThinkGear TGAM1	40
3.13. Descripción del dispositivo ThinkGear	41

CAPITULO 4

PRUEBAS DE CAMPO CON MODULO THINKGEAR DE NEUROSKY	49
4.1. Prueba de campo al módulo TGAM1	49
4.1.1. Objetivo de la prueba	49
4.1.2. Materiales	49
4.1.3. Procedimiento.....	49
4.1.4. Resultados.....	50
4.1.5. Discusión	52

CAPITULO 5

ANALISIS DEL KDT DE NEUROSKY	53
5.1. Análisis descriptivo del KDT proporcionadas por NeuroSky.....	53
5.2. Descripción de lenguajes compatibles con la DLL proporcionada.....	53
5.2.1. Plataforma C/C++	53
5.2.2. Plataforma C# y Java.....	53
5.2.3. Plataformas y lenguajes opcionales.....	54
5.2.4. Windows Mobile, C/C++	54
5.2.5. Windows Mobile, C#.....	54
5.2.6. Symbian para Dispositivos Móviles	54
5.2.7. Unix, Linux y Mac OS.....	54
5.3. Elección de la plataforma más adecuada para el prototipo.....	55
5.4. Breve descripción de la plataforma de código abierto Processing.....	57
5.5. Bocetos y Sketchbook	58
5.6. Adición de bibliotecas, herramientas y modos de ejecución	58
5.7. Breve descripción de la plataforma de código abierto Arduino.....	58
5.7.1. Hardware Arduino	60
5.7.2. Discusión de elección del Microcontrolador	60
5.8. Análisis de las API de NeuroSky para tecnologías Atmel y C++	61
5.8.1. ThinkGearStreamParser API en C	61
5.9. Desarrollo de una GUI conceptual	62
5.9.1. Objetivo de la prueba	62
5.9.2. Materiales, requerimientos de hardware y software.....	62
5.9.3. Informe del Procedimiento seguido en la prueba de concepto.....	62
5.9.4. Resultados.....	63
5.9.5. Discusión sobre la prueba realizada	64
5.10. Prueba de concepto al API proporcionado por NeuroSky.....	64
5.10.1. Objetivo.....	64
5.10.2. Materiales	64
5.10.3. Procedimiento.....	64
5.10.4. Resultados.....	66
5.10.5. Discusión	68

CAPITULO 6

FABRICACION DEL FIRMWARE.....	69
6.1. Compatibilidad.....	69
6.2. Firmware	69
6.3. Pruebas de concepto del prototipo desarrollado.....	70
6.3.1. Procedimiento.....	70

6.3.2. Resultados.....	71
CAPITULO 7	
MODULO DE CONTROL DE PERIFÉRICOS ELECTRONICOS.....	72
7.1. Diseño del circuito electrónico	72
7.1.1. MCPE (Módulo de control de periféricos electrónicos).....	72
7.1.2. Resultado	77
7.2. Pruebas de campo del prototipo desarrollado.....	78
7.2.1. Procedimiento.....	79
7.2.2. Resultados.....	79
CONCLUSIONES.....	80
RECOMENDACIONES.....	83
GLOSARIO.....	84
REFERENCIAS BIBLIOGRAFIAS.....	99
ANEXOS	105

Índice de Figuras

Figura 2.1	Onda completa de EEG en un segundo de muestreo.....	26
Figura 2.2	Ondas Delta.....	27
Figura 2.3	Ondas Theta.....	27
Figura 2.4	Ondas Alfa.....	28
Figura 2.5	Ondas Beta.....	28
Figura 2.6	Ondas Gama.....	29
Figura 2.7	Ondas Mu. "mu rhythmic".....	29
Figura 2.8	Captación de las señales EEG.....	30
Figura 2.9	Señal Artefacto presente en un estudio EEG.....	31
Figura 2.10	ocalización de electrodos sobre la corteza cerebral.....	32
Figura 2.11	Douglas Myers, Electrical Geodesics GES-250 system.....	32
Figura 2.12	Esquema de los bloques genéricos de las BCI.....	33
Figura 2.13	Distribución de sensibilidad en electrodos de EEG.....	34
Tabla 3.1	Aplicaciones de sistemas BCI y BMI basados en EEG.....	35
Figura 3.1	Orden KDT.....	39
Figura 3.2	Email recibido de confirmación de orden de compra.....	40
Figura 3.3	Módulo ThinkGear de NeuroSky.....	41
Figura 3.4	Dibujo mecánico de la placa del dispositivo ThinkGear de NeuroSky.....	43
Figura 3.5	Vistas superior e inferior de la placa TGAM.....	43
Figura 3.6	Configuración física de funcionamiento del módulo TGAM.....	44
Figura 3.6	Esquema básico del BCI de NeuroSky.....	44
Figura 3.8	Estructura de un paquete del dispositivo TGAM1.....	45
Figura 4.1	Puerto para enlace vía Serial COM13 hacia el dispositivo IBNS.....	49
Figura 4.2	Escucha de paquetes de datos desde el dispositivo TGAM1.....	50
Figura 5.1	Plataforma de desarrollo IDE Processing.....	57
Figura 5.2	Ejemplo Blink LED en el Entorno GUI IDE Arduino.....	59
Figura 5.3	Hardware Arduino Uno.....	60
Figura 5.4	Reproducción del boceto de Processing.....	63
Figura 5.5	Boceto en Fritzing creado para diagramar las conexiones.....	65
Figura 5.6	Esquema en Fritzing creado para diagramar las conexiones.....	65
Figura 5.7	PCB en Fritzing creado para diagramar las conexiones.....	66
Figura 5.8	Escucha de paquetes de datos vía Serial COM.....	67
Figura 5.9	Reproducción del boceto de Processing.....	67
Figura 5.10	Configuración del TGDC.....	68
Figura 7.1	Diagrama electrónico del circuito 1.....	73
Figura 7.2	Diagrama electrónico del circuito 2.....	74
Figura 7.3	Diseño de la PCB del módulo de control de periféricos.....	76
Figura 7.4	IBNS y MCPE enlazados al sujeto de prueba, el autor, Angel Yaguana.....	77
Figura 7.5	Modulo Arduino cargado con el firmware desarrollado.....	78
Figura 11 A-1	Resultados del muestreo de niveles de atención en el estudio de caso 1.	120
Figura 11 A-2	Resultados del muestreo de niveles de meditación en el estudio de caso 1.	120
Figura 11 A-3	Resultados del muestreo de niveles de intensidad de parpadeo en el estudio de caso 1.....	121
Figura 12 A-1	Resultados del muestreo de niveles de atención en el estudio de caso 2.	122
Figura 12 A-2	Resultados del muestreo de niveles de meditación en el estudio de caso 2.	122

Figura 12 A-3	Resultados del muestreo de niveles de intensidad de parpadeo en el estudio de caso 2.	123
Figura 10 A-1	Laboratorio acondicionado.	124
Figura 10 A-2	Materiales de fabricación de PCBs para prototipado.	125
Figura 10 A-3	Transferencia del fotolito mediante calor en una plastificadora.	125
Figura 10 A-4	Limpieza de residuos en el PCB para garantizar un buen ataque químico.	125
Figura 10 A-5	Materiales para atacado químico.	125
Figura 10 A-6	Ataque Químico usando Percloruro de Hierro y Peróxido de Hidrogeno.	126
Figura 10 A-7	Limpieza del PCB con sosa caustica.	126
Figura 10 A-8	Perforación del PCB.	126
Figura 10 A-9	Fabricación de la serigrafía del PCB.	127
Figura 10 A-10	Construcción del PCB.	127
Figura 10 A-11	Soldadura de elementos SMD.	127
Figura 10 A-12	Carga del bootloader Arduino en el Atmega 328 y posteriormente del firmware realizado.	128
Figura 10 A-13	Producto final listo para pruebas de campo.	128
Figura 13 A-1	Resultados de exploración EEG del sujeto 1, "Adulto Joven".	130
Figura 13 A-2	Resultados de exploración EEG del sujeto 2 "Mujer con Deficiencia Cardiaca".	130
Figura 13 A-3	Resultados de exploración EEG del sujeto 3 "Hombre con Deficiencia Visual".	131
Figura 13 A-4	Resultados de exploración EEG del sujeto 3 "Mujer de 3ra Edad".	131
Figura 13 A-5	Resultados de exploración EEG del sujeto 3 "Bebe de 18 meses"	131

Índice de Tablas

Tabla 3.1	Aplicaciones de sistemas BCI y BMI basados en EEG.	35
Tabla 3.2	Estructura y definición de códigos del Data Payload.	46
Tabla 3.3	Contenido del paquete capturado.	48
Tabla 4.2	Primer paquete identificado:.....	51
Tabla 4.3	Análisis de la trama recibida del módulo TGAM1 en el terminal serial del PC..	52
Tabla 5.1	Análisis de requerimientos básicos de diferentes entornos IDE para PC.....	56
Tabla 5.2	Análisis de requerimientos básicos de diferentes entornos IDE para uCs.	56
Tabla 5.3	Características de módulos Arduino para migrar la interfaz según sus recursos y aplicaciones.....	61
Tabla 11 A-1	Datos del sujeto 1 para exploración EEG usando el TGAM1 enlazado al MCPE prototipado y al PC vía Bluetooth y puerto serial respectivamente:	120
Tabla 12 A-1	Datos del sujeto 2 para exploración EEG usando el TGAM1 enlazado al MCPE prototipado y al PC vía Bluetooth y puerto serial respectivamente:	122
Tabla 13 A-1	Parámetros del estudio de caso.	129
Tabla 13 A-2	Resultados de los estudios.....	129
Tabla 14 A-1	Resultados de la encuesta realizada al sujeto 1.....	132
Tabla 14 A-2	Resultados de la encuesta realizada al sujeto 2.....	132
Tabla 14 A-3	Resultados de la encuesta realizada al sujeto 3.....	132
Tabla 14 A-4	Resultados de la encuesta realizada al sujeto 4.....	133
Tabla 14 A-5	Resultados de la encuesta realizada al sujeto 5.....	133

RESUMEN

Actualmente existe una emergente necesidad en disponer de herramientas, sistemas y plataformas que potencien estudios de experimentación de aspectos relativos al control neuromotor o al análisis biomecánico, y sobre todo que constituya a establecer las capacidades de la neurotecnología a fin de alcanzar el desarrollo de prótesis electromecánicas controladas mediante órdenes cerebrales.

La presente tesis abarca los campos científico, tecnológico y experimental, y propone la implementación de un sistema BCI (Interfaz cerebro-computadora) que otorgue control de periféricos eléctricos a un ser humano por medio de sus estados mentales asociados a la acción biométrica del cerebro en cada uno de ellos.

Como solución a esta necesidad, se desarrolla una interfaz biométrica neurosensorial para controlar periféricos electrónicos mediante un módulo de comunicación que se enlaza al dispositivo EEG colocado a una persona, el cual leerá y procesará las señales emitidas por el cerebro logrando definir los estados mentales que éste presenta, este dispositivo tiene como característica proporcionar un conjunto de sistemas con una plataforma de desarrollo amigable y compatible con Arduino, para crear aplicaciones que ayuden a mejorar la calidad de vida de las personas, en especial de aquellas que poseen capacidades limitadas, empleando para ello tecnología de vanguardia.

Palabras clave: neuroseñales, control, biométrico, interfaz, periféricos

ABSTRACT

Currently there is an emerging need to have tools, systems and platforms that enhance experimental studies of aspects related to neuromotor control or biomechanical analysis, especially constituting establish neurotechnology capabilities to achieve the development of electromechanical prosthesis controlled by brain commands.

This thesis covers the scientific, technological and experimental fields, and proposes the implementation of a system BCI (brain computer interface) to give control of electrical peripherals to a human being through their associated mental states to biometric brain action in each one.

As a solution to this need, sensorineural develops biometric interface for controlling electronic peripherals through a communications module that EEG device placed on a person, which will read and processed signals from the brain making mental states define it binds features, this device has the feature to provide a set of systems with a friendly platform Arduino compatible development and to create applications that help improve the quality of life of people, especially those that have limited capabilities, employing forefront technology..

Keywords: neurosignals , control, biometric , interface, peripherals

CAPÍTULO 1

DESCRIPCIÓN Y ASPECTOS TEÓRICOS

1.1. Introducción.

En la actualidad, empresas e institutos de investigación en el área de desarrollo de prototipos basados en Neurotecnología, han emprendido investigaciones orientadas al uso y empleo de los sistemas BCI (Interfaz Cerebro-Computador) en busca de obtener referencias reales de la interacción entre personas en condiciones físicas especiales y los dispositivos que faciliten su situación primaria de buen vivir. En virtud de satisfacer esta necesidad, esta investigación busca proporcionar control mental sobre dispositivos electrónicos y mecánicos, mediante la interpretación de señales cerebrales que al ser acogidas en un EEG, se conviertan en un medio de comunicación que facilita a los usuarios una adaptación real con el entorno exterior mediante el uso de sus estados mentales como medio de control. (By Sean M. Montgomery, 2013)

Los equipos que permiten realizar estudios EEG traducen la actividad cerebral como producto de un reflejo a estímulos sensoriales recogidos por los cinco sentidos y además captan frecuencias que según sus patrones de comportamiento pueden permitir interpretar estados mentales como imaginación, meditación, concentración, frustración, miedo, ira, y cualquier tipo de actividad que genere pensamientos emotivos e incluso existen algunas BCI que pueden interpretar señales motoras aun cuando no se haya realizado el movimiento muscular en el cuerpo todavía.

Todas estas señales pueden ser analizadas mediante algoritmos de comparación de patrones de comportamiento y ser registradas para su estudio ya sea este con un enfoque meramente interpretativo de comportamiento cerebral, o de generación de acciones reales mediante aplicativos como por ejemplo: sillas de ruedas, prótesis, vehículos motorizados, robots, y en general todo tipo de electro mecanismos que pueden ser controlados con señales eléctricas, con el fin de asistir de manera óptima al cliente en una tarea específica.

Sin embargo, todas estas aplicaciones tienen varios requisitos para su adecuado funcionamiento siendo los recursos tecnológicos y los medios de investigación los más importantes para su adecuado desarrollo y es, precisamente en este campo donde se centra ésta investigación ya que un requerimiento esencial es garantizar que las señales recogidas por la BCI utilizada, representen de modo real los distintos patrones de comportamiento del cerebro disponibles en el transcurso de la adquisición, además de que sus resultados sean de fácil extracción.

Cabe mencionar que uno de los objetivos principales del desarrollo de aplicativos neurotecnológico, se centra en el ámbito de desarrollo de prótesis y biomecanismos debido a la creciente tasa de personas discapacitadas, por lo cual se hace necesario pensar en una forma de adaptar los diferentes sistemas BCI y BMI que existen para mejorar la calidad de vida de estas personas.

En la presente investigación, se presenta una interfaz gráfica que permite al desarrollador visualizar de forma intuitiva y muy versátil las variables que se está usando en el sistema de interpretación de señales cerebrales propio de NeuroSky, además, se demuestra mediante el uso de un hardware electrónico que se puede manipular dispositivos electrónicos por medio del uso de los umbrales de los estados mentales o tendencias de patrones ya existentes generadas a través de algoritmos de redes neuronales. De esta manera, esta propuesta se proyecta a desencadenar desarrollos tecnológicos en campos como la biomedicina, la robótica, la mecatrónica y la biomecánica, desde un punto de vista neurotecnológico y social.

Un ejemplo muy útil del uso de este módulo, es la implementación de una BCI para mover una silla de ruedas a través de varios estados mentales, pero que se debe en primera instancia lograr controlar un periférico electrónico de mínimo dos estados, para lo cual se toma como base el estudio realizado por el Ing. Roberto García en el proyecto "Investigación de Neuro-Tecnologías Innovadoras Aplicadas Comercialmente INIAC." presentado al SINDE en el año 2010 en Guayaquil- Ecuador , del cual cabe mencionar que se tuvo participación en el diseño del prototipo electrónico de control y su enlace inalámbrico a la silla de ruedas eléctricas.

1.2. Antecedentes

En un inicio la neurotecnología era un sistema que estuvo orientado a ayudar a personas que tenían algún tipo de discapacidad y les ayudaba con la movilidad. Por ejemplo, el concepto original era que a través de las ondas cerebrales se puede mover una silla de ruedas. (Tatoon., 2007)

En el año 2008 NeuroSky a través de su producto Mindset el cual está basado en el sistema BCI ThinkGear AM 1, mostró a un usuario como este podría mover objetos dentro de un juego usando señales emitidas desde su cerebro, esto provocó una reacción en cadena en varias universidades (Brown, University of Glasgow, University of Wollongong, Kaist, Dongguk University, entre otras) a que se adhieran a la investigación de esta tecnología. (Costumers., 2008). Adicionalmente se observó que hasta el año 2011 no existen registros o patentes de equipos que se hallen presentes en Universidades de Ecuador, por lo cual nuestra universidad quedo temporalmente relegada de los beneficios que pueda brindar la neurotecnología hasta el año 2012 en el que gracias a la investigación del Ing. Roberto García Vacacela, catedrático de nuestra universidad, la UCSG ya cuenta con un caso de estudio en este campo.

Este estudio, según los resultados de su proyecto de exploración en Neurotecnología INIAPCE, han expuesto la necesidad de una interfaz electrónica más accesible puesto que, su limitante principal es la interacción directa para adquisición de datos y su complejo uso de las señales digitalizadas, impidiendo que profesionales del medio puedan proyectar a corto plazo y limitado presupuesto el desarrollo de dispositivos electrónicos independientes del uso de un ordenador o de sus software demostrativos para crear aplicaciones que ayuden a mejorar la calidad de vida del usuario, incrementando exponencialmente las diferencia de nuestra comunidad en cuanto al avance tecnológico y social.

Cabe mencionar que crear un control electrónico o mecánico a través de algoritmos computacionales partiendo de señales cerebrales se concibe importante cuando se denota que los desarrollos en esta área son limitados en comparación con desarrollos vinculados con el área de medicina y de entornos virtuales como videojuegos y simuladores.

Además, es preciso señalar que el dispositivo TGAM1 comenzó a utilizarse en marketing comercial en Ecuador con el objetivo de conocer ciertas preferencias de los clientes y su reacción frente a estimulaciones sensoriales, tal como lo explica Johanna Vinuesa, Gerente Nacional de Mercadeo de Ecuacerámica en la Revista Líderes:

“Se ha aplicado el servicio de NeuroSky en dos grupos de 12 personas en Quito y 20 en Guayaquil, realizando un análisis de la cuña publicitaria que estaban pautando por los 50 años de la empresa. El principal resultado fue que la música que utilizaron no llamaba la atención a los espectadores...” (Editorial-Cuenca, 2010).

1.3. Problema

La BCI TGAM1 de NeuroSky, siendo una interfaz muy robusta, posee protocolos de comunicación muy complejos que limitan su uso en desarrollos tecnológicos, es por esto que surge la necesidad de generar una interfaz de comunicación sencilla, segura y amigable para el desarrollador, que cumpla con todas las características básicas de su sistema BCI y que mejore eficientemente el nivel de respuesta y entrega de datos al usuario.

1.4. Objetivos

El crecimiento de la neurotecnología va de la mano con el crecimiento y la evolución de las aplicaciones que surgen de las necesidades de los clientes, conforme avanzan los estudios relacionados, se hace imprescindible incursionar en el perfeccionamiento de dispositivos electrónicos que satisfagan estas necesidades para cumplir de manera eficiente con los requerimientos de los usuarios.

1.4.1. Objetivo general

Desarrollar un módulo de control de periféricos electrónicos, que mediante una BCI de NeuroSky otorgue un control sencillo y confiable de éstos a un investigador de neurotecnología.

1.4.2. Objetivos específicos

- a. Establecer el estado del arte del sistema BCI de NeuroSky.
- b. Analizar el funcionamiento del dispositivo de EEG comercializado por NeuroSky.
- c. Realizar pruebas de concepto con el Kit de desarrollo para tecnologías Atmel y C++, comercializado por NeuroSky.
- d. Elaborar un protocolo de comunicación que implementado en un firmware para un microcontrolador, permita una comunicación sencilla y seguro al desarrollador.
- e. Elaborar un módulo de control de periféricos electrónicos, que mediante el firmware implementado, permita demostrar la factibilidad de desarrollar fácilmente aplicaciones neurotecnológicas.

1.5. Hipótesis

Mediante el diseño de un protocolo de comunicación sencillo para la BCI TGAM1 de NeuroSky implementado en un módulo de control de periféricos electrónicos, se impulsará el desarrollo de aplicaciones neurotecnológicas

1.6. Justificación

Mediante el uso de esta esta propuesta, el investigador podrá aprovechar esta tecnología en proyectos encaminados a mejorar la condición de vida de las personas, en especial de aquellas que poseen capacidades limitadas, contribuyendo de forma directa en la formación de un entorno social donde todas las personas puedan tener las mismas oportunidades sin caer en estereotipos de discriminación por sus capacidades físicas.

1.7. Limitaciones del estudio

El estudio según indican sus objetivos no intenta crear aplicaciones listas para su uso, sino, más bien intenta demostrar la factibilidad de crear dichas aplicaciones a través del empleo de una interfaz de comunicación más sencilla que permita en poco tiempo y con poco esfuerzo e inversión, el desarrollo de las mismas y que además, para ser realmente útiles deben ser elaboradas junto a profesionales de la ciencia sobre la cual va a ser aplicada para su implementación.

1.8. Técnicas y métodos empleados en la investigación

La investigación parte de un análisis del estado de arte de la neurotecnología para conocer los fenómenos que se relacionan con el EEG para lograr manipular sus variables sistemáticamente y así obtener los resultados deseados mediante procesos de experimentación teniendo en cuenta las posibles aplicaciones que se pueden generar con el resultado desde un enfoque de ambiente natural.

Método

La investigación se fundamentará en un enfoque cuantitativo de tipo experimental, de observación y bibliográfica, así como la aplicación de pruebas de concepto para el diseño del prototipo, usando para esto el método lógico inductivo con inducciones científicas incompletas.

Justificación de la selección del método

Por ser una investigación en un principio analítica y que mediante experimentación cuantifica las variables observadas en sus resultados, mediante el empleo de pruebas de concepto y de campo, y que éstas parten de casos particulares elevándolos a conocimientos generales, es necesario un método que nos permita la formulación de hipótesis para una investigación demostrativa usando inducciones incompletas de tipo científicas, donde *"Los elementos del objeto de investigación no pueden ser numerados y estudiados en su totalidad, obligando al sujeto de investigación a recurrir a tomar una muestra representativa, que permita hacer generalizaciones"* (Gamboa, 2014).

Así mismo, debido a que el objetivo de esta investigación prevé realizar pruebas de concepto para generar conclusiones en las cuales se pueda referenciar el prototipo a diseñar y de las cuales se harán pruebas experimentales cualitativas y cuantitativas, es necesario usar otro método que permita *"observar el comportamiento de los modelos-*

planteados, los comportamientos de sus elementos y sus relaciones", finalmente, debido a que los datos recogidos de la descomposición del objeto principal de estudio y del análisis a cada una de estas, necesitamos usar un método que nos permita "analizar cuantitativa y cualitativamente las variables para poder predecir sus comportamientos de forma subjetiva." (Arena, 2008), para poder formular mediante inducción científica, "las conexiones necesarias del objeto de investigación y sus relaciones de causalidad. Este método se apoya en métodos de observación y la experimentación." (Gamboa, 2014)

Diseño de la investigación

Se someterá el hardware y firmware a determinadas condiciones y estímulos, para observar los efectos o reacciones que se producen, usando un estilo netamente explicativo, por cuanto nuestro propósito es demostrar la relación causa-efecto entre el hardware proporcionado por NeuroSky y la herramienta de control concebida, por lo cual el diseño será experimental, tipo causal. (Tamayo, 2000)

Muestra

Esta investigación se basará en el estilo de muestreo intencionado, (Alvira, 2003). Esto se debe a que se realizarán pruebas de campo y de concepto entre hardware y software para tomarlas como unidades de estudio, estas pruebas de concepto serán aplicadas tanto a los componentes de NeuroSky como en la solución desarrollada para demostrar el potencial de la tecnología.

Técnicas de recogida de datos

Los datos recogidos se relacionarán basándose en técnicas de observación (Tamayo, 2000), debido a que luego de crear el ambiente para prueba, se procederá a observar y registrar los resultados obtenidos de las modificaciones a las variables analizadas, esta técnica es la adecuada para esta investigación puesto que comprende el diseño de hardware y software, en la cual se va a observar su comportamiento e interacción y por ende, sus resultados son medibles y demostrables.

Técnicas y modelos de análisis de datos

Para el modelo de análisis, los datos serán analizados en base a los resultados obtenidos en las pruebas de concepto, las cuales permitirán calificar el potencial aporte de cada elemento analizado dependiendo de los valores de las variables obtenidas en cada una de las prueba. (Tamayo, 2000)

Resultados

Los resultados de los muestreos de las variables analizadas serán tabulados para luego ser anexados al informe final, adicionalmente las muestras obtenidas con el software y hardware desarrollado para usarlo con la tecnología NeuroSky, también serán anexadas para su análisis.

CAPÍTULO 2

MARCO TEÓRICO REFERENCIAL

2.1. Marco teórico

En este segmento se relata una descripción detallada de los estudios previos el desarrollo de los sistemas BCI y BMI basados en un EEG como forma de adquisición de la actividad bioeléctrica y la forma biométrica de cuantificación y cualificación de datos recogidos mediante las interfaces neurosensoriales.

2.1.1. Sistemas de adquisición de datos de neuroseñales

Richard Caton (1842-1926), un médico de Liverpool, presentó en 1875 sus hallazgos sobre los fenómenos bioeléctricos en los hemisferios cerebrales de ratones y monos, expuestos por craniectomía. Posteriormente, Hans Berger (1873-1941) comenzó sus estudios sobre electroencefalografía en humanos, en 1920 como una técnica que le permitió estudiar la actividad cerebral para conocer mejor el diagnóstico y la localización de su enfermedad, y en muchos casos la intensidad de una posible lesión. (Berger, 1991)

De esta forma, hoy en día ya se puede orientar a un médico especialista sobre el diagnóstico y tratamiento a seguir mediante estos estudios.

En la actualidad los centros de investigación de neurociencias aplicadas y las universidades a través de sus institutos de medicina en el área de neurología han centrado en su mayoría sus investigaciones en diseñar sistemas de adquisición de datos que sean efectivos para que les permita recoger datos confiables del comportamiento de las señales bioeléctricas y todas señales de la actividad cerebral para así poder estructurar métodos y formatos que hagan más simple el manejo de las diferentes tecnologías y así la información pueda ser interpretada por medio de un procesador y genere informes adecuados al estudio para el cual fueron diseñados.

En este tipo de desarrollos los sistemas a menudo están basados en estudios que mediante un EEG y complejos algoritmos matemáticos de ubicación espacial han permitido definir las posiciones de los electrodos para extraer de manera efectiva la información de los estados mentales del cerebro en diferentes momentos de la vida de un sujeto.

Actualmente existen por definición dos interfaces que permiten una comunicación en tiempo real con el pensamiento del humano estas son los sistemas BCI y BMI. (PAULO A, 2010)

2.1.2. Métodos de medición mediante EEG

Los EEG son registros de los potenciales eléctricos generados por ondas cerebrales, que son recogidos a través de sensores que captan las señales bioeléctricas emitidas por el cerebro como una respuesta integrada de las estructuras cortico-subcorticales reflejadas en las capas más superficiales de la corteza craneal.

En el cerebro existen millones de neuronas, cada una de estas genera pequeños campos de voltaje eléctrico, el EEG es la superposición de muchas de estas señales que típicamente se hallan en el rango de 1 μ V a 100 μ V en un adulto normal y de 10 a 20mV cuando son medidos con redes de electrodos, pero para prevenir el caos en la comunicación a las diferentes entidades del cuerpo desde nuestro cerebro, este usa una sincronización llamada rhythmic syncopation a varias escalas de tiempo tal que las redes neuronales locales puedan sincronizarse una con otra y con distantes redes como una melodiosa orquesta sinfónica.

Básicamente, la idea es que cuando las neuronas entran en proceso de sinapsis para transmitir impulsos de potencial eléctrico una con otra estas lo hacen sincronizadamente y el cerebro puede generar auto recordatorios preestablecidos del camino más efectivo y armonioso con el cual los micro voltios oscilaron en momentos específicos de estimulación o actividad para así volver a repetir las mismas acciones en similares eventos.

Estos trazos de información al ser recogidos por un dispositivo de EEG nos dan la capacidad de poder clasificar estas variaciones de voltaje de acuerdo a la frecuencia con la cual oscilan en estímulos provocados hasta obtener un estado real del pensamiento instantáneo.

La mayoría de los sistemas de EEG de estos días, son digitales, y la señal amplificada es digitalizada a través de un CAD (Convertor Analógico Digital), después de haber sido pasado a través de un filtro anti -aliasing. El muestreo en un CAD se produce normalmente en 256-512 Hz en el cuero cabelludo. (Yasui, 2009)

La señal de EEG digital se almacena electrónicamente y se puede filtrar para la visualización. La configuración típica para el filtro de paso alto y un filtro de paso bajo son 0,5-1 Hz y 35-70 Hz, respectivamente. El filtro de paso alto normalmente filtra el artefacto lento, tales como señales de electrogalvánico y movimiento artefacto, mientras que el filtro de paso bajo filtra los artefactos de alta frecuencia, como las señales electromiográficas. Un filtro de muesca adicional se utiliza típicamente para eliminar artefacto causado por líneas de energía eléctrica (60 Hz en Ecuador y 50 Hz en muchos otros países).

2.1.3. Comportamiento de la señal EEG

A partir de la señal del EEG es posible diferenciar ondas clasificadas en términos de su frecuencia de oscilación y que son denominadas alfa, beta, gama, delta, y theta, (Shackman, McMenemy, Maxwell, Greischar, & Davidson, 2010). Un ejemplo de la forma de onda de un EEG sin descomponerse en términos de actividad rítmica se da en la figura a continuación:

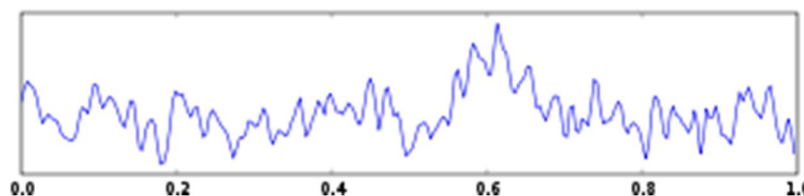


Figura 2.1 Onda completa de EEG en un segundo de muestreo.

Fuente: Elektroenzephalographie (EEG), (Heinz Penin, Walter Fröscher, 2013)

Bandas de frecuencia de actividad rítmica en el EEG

Ondas Delta

Las ondas delta tienen el rango de frecuencia de 0,1-4 Hz y son detectables en los recién nacidos y adultos mientras duermen y en la etapa non-REM del inconsciente. Están también presentes en el sueño profundo y puede ser estado de sonambulismo, de hablar estando dormido. (Steinkopff-Verlag, 2008)

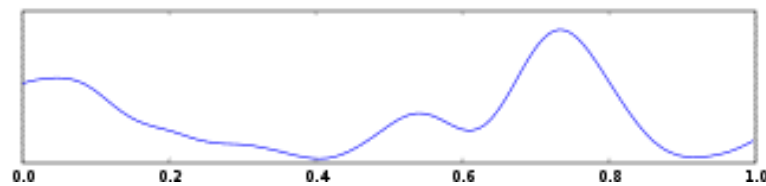


Figura 2.1 Ondas Delta.

Fuente: Elektroenzephalographie (EEG), (Heinz Penin, Walter Fröscher, 2013)

Ondas Theta

Las ondas theta tienen el rango de frecuencia de 4-7 Hz y se obtienen en momentos de creatividad, fantasía y en los sueños que implican imaginación de los niños o de los adultos. Puede ser nivel de clasificación de Ideas que incluyan creatividad asociativa, relajación profunda, y percepción reducida, excepto en personas experimentadas en técnicas de meditación (yoguis, maestros Zen, etc.)

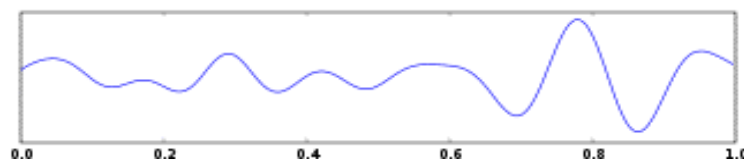


Figura 2.3 Ondas Theta.

Fuente: Elektroenzephalographie (EEG), (Heinz Penin, Walter Fröscher, 2013)

Ondas Alfa

Las ondas alfa tienen el espectro de frecuencia de 8-12 Hz y se puede medir a partir de la región occipital en una persona despierta en momentos de relajación pero no dormitación, tranquilidad y conciencia mientras los ojos están cerrados. Patológicamente es síntoma de coma. Estas ondas son el reflejo de la "MENTE SUBCONSCIENTE" la cual comprende los niveles de ausencia de estrés, relajación, creatividad, percepción, meditación. Predomina el HEMISFERIO DERECHO el cual es de particularidad intuitiva, no-verbal, y tiene mucha influencia en los sentimientos.

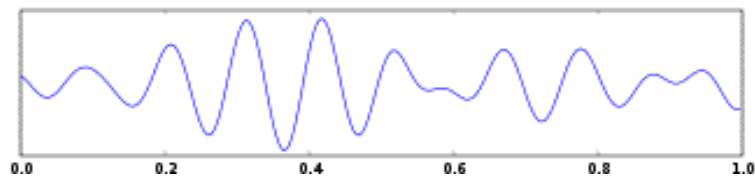


Figura 2.1 Ondas Alfa.

Fuente: Elektroenzephalographie (EEG), (Heinz Penin, Walter Fröscher, 2013)

Ondas Beta

Las ondas beta van en el rango de 12-20 Hz, los cuales son detectables en los lóbulos parietales, frontales y pueden ser medidas en el momento de SMR y relajación enfocada. Las ondas beta son el reflejo de la "MENTE CONSCIENTE" ya que recibe señales de los 5 sentidos normales Vista (+- 80%), tacto, oído, olfato, gusto, Además está presente en el hecho de desenvolverse en la vida diaria, estrés, ansiedad, aprensión, tensión, competir. Predomina el HEMISFERIO IZQUIERDO el cual es descrito por su particularidad lógica, racional, verbal.

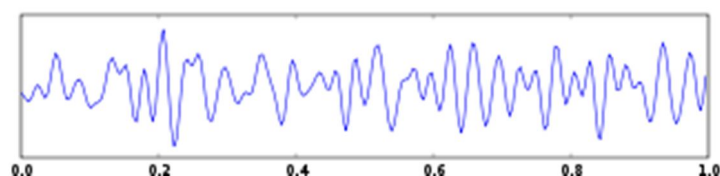


Figura 2.5 Ondas Beta.

Fuente: Elektroenzephalographie (EEG), (Heinz Penin, Walter Fröscher, 2013)

Ondas Gama

Las ondas gama pueden alcanzar valores que van desde los 30Hz hasta los 100Hz en algunos casos pero más frecuentemente son de hasta 50Hz, están presentes en periodos cortos de memoria, reconocimiento de sonidos y sensaciones de reconocimientos de objetos.

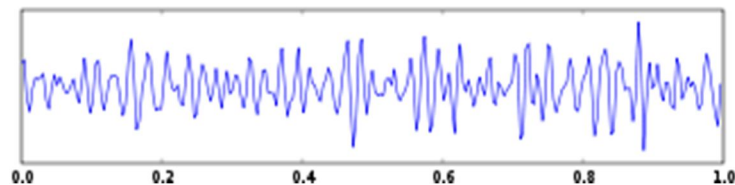


Figura 2.6 Ondas Gama.

Fuente: Elektroenzephalographie (EEG), (Heinz Penin, Walter Fröscher, 2013)

Ondas Mu

Estas ondas a pesar de estar en el mismo rango de las alfa están presentes en el resto de actividades cerebrales que implican las neuronas motoras y son las que representan el instante de una función motora.

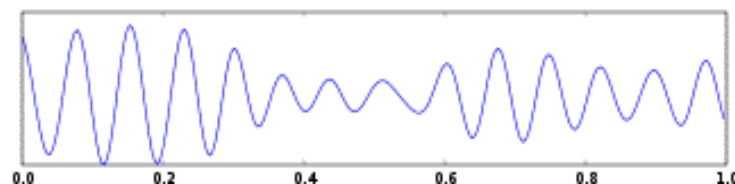


Figura 2.7 Ondas Mu. "mu rhythmic".

Fuente: Elektroenzephalographie (EEG), (Heinz Penin, Walter Fröscher, 2013)

2.1.4. Captación de las señales EEG

El sistema más básico de referencia mediante medición a través de electrodos es el mostrado en la Figura 2.8, en el cual se puede apreciar la toma de muestra de una señal sinusoidal captada por un electrodo o varios en un arreglo predispuesto, los cuales entregan esta información a un amplificador para tener una señal que se adapte a los valores de modulación digital, previamente filtrada para eliminar ruidos y artefactos.

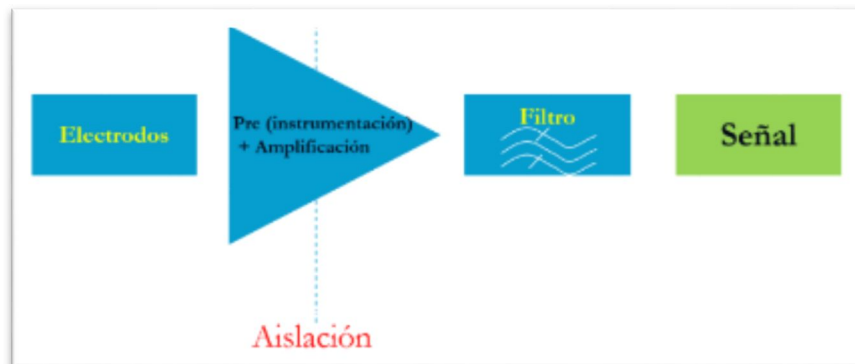


Figura 2.8 Captación de las señales EEG.

Fuente: El autor, Angel Yaguana.

El muestreo de las señales durante un EEG debido a su gran sensibilidad y su limitado rango de 1 a 100mV puede ser fácilmente contaminado por otras fuentes. Una señal de EEG que no se originan en el cerebro se denomina un artefacto. (Joyce, Gorodnitsky, & Kutas, 2004)

Artefacto

Un Artefacto en un examen EEG es una señal no deseada dentro del registro y que muchas veces es considerada como ruido. Existen dos categorías: fisiológicos y no fisiológicos.

Artefactos Fisiológicos. Son cualquier fuente en el cuerpo que tiene un dipolo eléctrico o genera un campo eléctrico. Estos incluyen el corazón, los ojos, los músculos faciales, y la lengua.

Enfermedades como la Hiperhidrosis también puede alterar la impedancia en la medida de los electrodos e interactuar con las señales involucradas y producir un artefacto (Nuñez, 1981).

Artefactos no fisiológicos. Incluyen 60 Hz de interferencia proveniente de aparatos eléctricos, artefactos quinesiológicos causados por el cuerpo o movimientos en los electrodos y artefactos mecánicos causados por el movimiento del cuerpo.

La remoción de artefactos es un factor importante en el análisis de EEG, por ello se requiere de una metodología apropiada, la cual permite identificar y eliminar la mayoría de los artefactos presentes en un estudio EEG sin omitir información. (Jung, y otros, 2000)

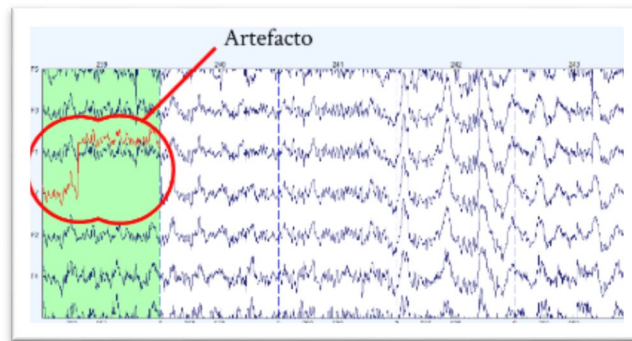


Figura 2.9 Señal Artefacto presente en un estudio EEG.

Fuente: Removiendo de un EEG el ruido y artefactos mediante la separación ciega de fuentes, (Fitzgibbon, Powers, Pope, & Clark, 2007).

2.1.5. Sistemas BCI y BMI basados en EEG

Definición

Son sistemas que permiten leer neuroseñales para mediante un específico procesamiento digital interpretar sus resultados en una computadora o máquina.

Arquitectura

Comprende básicamente un equipo de diagnóstico que permita realizar una lectura de actividad cerebral, las cuales fueron previamente amplificadas para mejorar las señales recibidas. Una vez digitalizadas estas señales, se envían a un computador donde se procesan para obtener un resultado que va desde generar una retroalimentación visual mediante una GUI a controlar un dispositivo electrónico.

Características

Esos sistemas se caracterizan por involucrar distintos saberes entre tecnologías multidisciplinarias como la biotecnología, nanotecnología, tecnologías de la información, ingeniería bioquímica, neurociencia y matemáticas aplicadas que entre otras ofrecen soluciones a problemas físicos, mentales y psicológicos.

Clasificación

- Método invasivo.

En este método, la calidad de lectura de las neuroseñales es muy buena dada su cercanía a la corteza cerebral y el medio de transmisión que usan que en este caso es la membrana cerebral, un ejemplo de ello se presenta en la siguiente figura la cual muestra la colocación de implantes para detección de áreas epileptógenas y de áreas de la corteza donde se asientan funciones motoras o de habla.

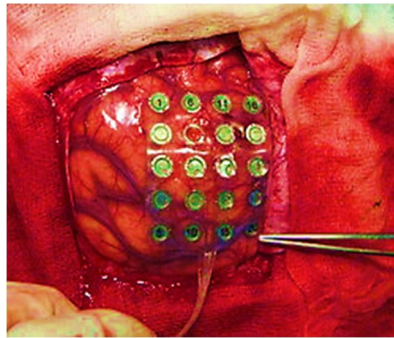


Figura 2.10 Localización de electrodos sobre la corteza cerebral.

Fuente: Detección de áreas de la corteza donde asientan funciones motoras o del habla, (IFCN, 2014)

- Método no-invasivo.

Este método no requiere de ningún tipo de cirugía para colocar algún dispositivo en el interior del cráneo, sino que más bien, permite identificar un amplio rango de actividad por sus múltiples electrodos dispuestos en un arreglo espacial.



Figura 2.11 Douglas Myers, Electrical Geodesics GES-250 system.

Fuente: The Neurocognitive Imaging lab is equipped with two EEG/ERP recording systems, (Lab, 2014).

2.1.6. Modelo funcional genérico de un sistema BCI

En este apartado se describe el principio de funcionamiento básico de las interfaces cerebro ordenador según el modelo funcional genérico de bloques al que podrían reconocer la práctica totalidad de los sistemas BCI (La Cofa, 2011). Se diferencian cuatro bloques principales:

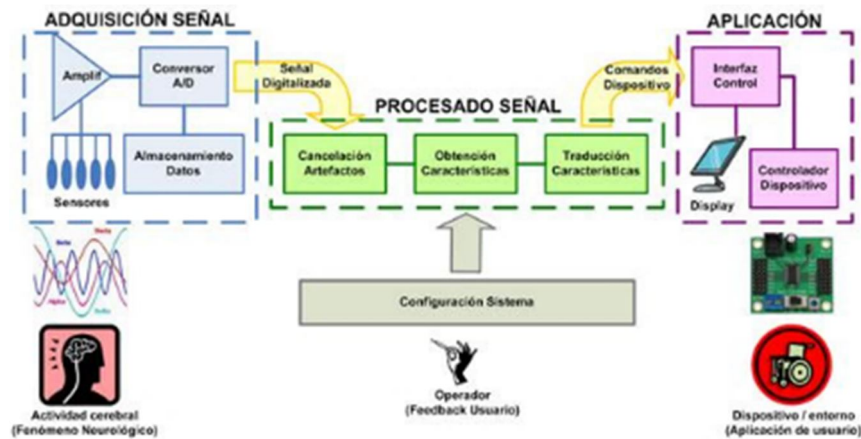


Figura 2.12 Esquema de los bloques genéricos de las BCI.

Fuente: Introducción a los sistemas Brain Computer Interface, (La Cofa, 2011).

Adquisición

En esta etapa se realiza el registro de la señal tomada por los electrodos colocados en el usuario y su adecuación para su procesamiento posterior mediante etapas de amplificación y digitalización.

Procesamiento

Es donde se recibe la señal digitalizada y se la transforma en comandos que entiende el dispositivo sobre el que el usuario está actuando. Se divide en tres etapas secuenciales:

- **Cancelación de artefactos**, elimina el ruido producido por algún otro tipo de actividad bioeléctrica que contaminan la señal de entrada.
- **Obtención de características**, traduce la señal en un vector de características en correlación con el fenómeno neurológico asociado a la señal.
- **Traducción de características**, transforma el vector de características en una señal de control adecuada al dispositivo que se pretende controlar.

Aplicación

Es aquí donde finalmente se obtiene los comandos de control y se realiza las acciones correspondientes en el dispositivo a través del controlador del mismo. Puede incorporar una GUI que proporcione feed-back al usuario.

Configuración

Aquí es donde un operador puede definir los parámetros del sistema, el cual no necesariamente debe ser una persona técnica, sino que puede ser el propio usuario del sistema o inclusive podrían ser algoritmos automáticos que ajustan el comportamiento del sistema en función del feed-back generado.

2.1.7. El sistema internacional 10-20 EEG

El sistema internacional 10-20 EEG de medidas, es usado como un estándar para ubicar posiciones específicas en un esquema de sistema de medidas EEG. El sistema original incluye solo 19 electrodos, posteriores estudios llegarían hasta los 70 electrodos formando una nube de posiciones estándar en la periferia del cráneo pero generalmente uno de los electrodos es usado como referencia para obtener con más eficiencia los datos y este electrodo está situado precisamente en el lóbulo de la oreja tal como lo indica el primer método. (PAULO A, 2010)

El sistema de localización estándar 10-20 para EEG fue publicado en Gilmore en 1994 y toma este nombre debido a las distancias que hay entre los electrodos en relación a los porcentajes de 20% y 10%. El esquema esta descrito en la siguiente figura:

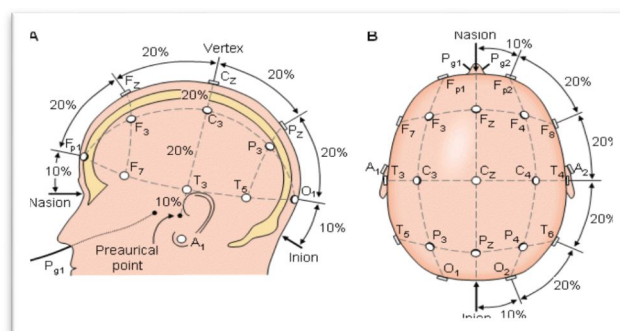


Figura 2.13 Distribución de sensibilidad en electrodos de EEG.

Fuente: (Gilmore, 1994), (Suihko V, 1993)

CAPÍTULO 3

ESTADO DEL ARTE Y ANALISIS DE LOS SISTEMAS BCI

En este capítulo se expone sistemáticamente los avances existentes acerca del tema de investigación y de los procesos más relevantes y fundamentales de la investigación, relacionando el funcionamiento y calidad de la herramienta a través de un profundo análisis.

3.1 Estado de Arte

A continuación se citan referencias de estudios y aplicaciones en las cuales se basa este proyecto para guiar la investigación a partir de hechos relacionados al tema, además, compendia un listado de avances tecnológicos que se han realizado hasta el momento y sobre los cuales se hizo el análisis de contribución potencial del estudio, destacando la importancia y aporte que se realiza con esta I+D+i.

Tabla 3.1 Aplicaciones de sistemas BCI y BMI basados en EEG.

Institución	Aplicación
Científicos de la Escuela de Medicina de Hahnemann y del Centro Médico de la Universidad de Duke	Ratas controlan un brazo dispersor de agua mediante sus señales cerebrales. Posible uso para autenticación biométrica basada en ondas cerebrales ante un estímulo.
Compañía NeuroSky	Desarrollo de sistemas BCI para fines médicos y para la industria del entretenimiento.
Ciberkinetics Neurotechnology Systems	Desarrollo de un BCI de tipo invasivo llamado BrainGate que consta de un arreglo de 100 micro electrodos de silicón colocados en la corteza motora, los cuales captan las señales cerebrales y son enviados mediante cables de fibra óptica a una computadora para generar la salida deseada. Su objetivo es restaurar el movimiento de las extremidades de los pacientes con daños en la espina dorsal y que sufren de distrofia muscular así como permitir a víctimas de enfermedades motoras cerebrales comunicarse con el medio.
Centro Wadsworth del Departamento de Salud Pública del	Desarrollo de un sistema BCI no-invasivo llamado sistema de investigación BCI2000 que busca permitir a los usuarios comunicarse. Consta de hasta 200 electrodos colocados en el cuero cabelludo los cuales registran la actividad eléctrica.

Estado de Nueva York	Mediante la onda cerebral P300 (método basado en un potencial eléctrico generado ante un estímulo) pueden conocer si el usuario tiene deseo de hablar así como conocer las palabras que busca comunicar de entre distintas palabras generadas al azar. Buscan generar una fundación no lucrativa que haga el dispositivo accesible a quienquiera que lo necesite.
Compañía Honda	Sistema BCI que utiliza un escáner de Imagen por Resonancia Magnética para obtener las señales cerebrales del usuario las cuales son enviadas mediante cables de Ethernet a una computadora que genera como salida el movimiento de una mano robótica que se posiciona de la misma forma que la del usuario. Buscan llegar a implementar en un futuro esta tecnología para mejorar la seguridad durante el manejo enviando a los demás conductores las intenciones de algún conductor.
Universidad de Stanford	Desarrollo de un sistema BCI que permita reconocer las señales que preceden al movimiento y que indican la intención y dirección del mismo para desarrollar prótesis más efectivas y un sistema que permita al usuario comunicarse con el dispositivo.
Universidad de Columbia.	Desarrollo de un sistema BCI no invasivo que permita identificar imágenes mucho más rápido que los mismos humanos, tiene como nombre C3Vision. El usuario utiliza un gorro con electrodos los cuales detectaran los picos de actividad eléctrica generados ante la presencia de algo inusual en las imágenes las cuales son clasificadas según el potencial de la actividad cerebral.
Universidad de Tecnología de Helsinki	Desarrollo de un sistema BCI que utilice electroencefalografía o magneto encefalografía para controlar un teclado virtual con sólo pensarlo.
Neural Signals	Desarrollo de un proyecto BCI invasivo que logra restaurar el habla mediante electrodos neurotróficos en el área de Broca. En la punta de estos electrodos crecen las neuronas y se leen de esta forma las señales emitidas por las mismas e identificando los fonemas adecuados a cada señal.
Compañía Emotiv	Desarrollo de sistemas BCI para su aplicación en videojuegos.
Compañía Toyota	Desarrollo de un sistema BCI para controlar una silla de ruedas con un tiempo de respuesta de 125 milisegundos.

Fuente: El Autor, Angel Yaguana.

3.2. Análisis de los antecedentes y requerimientos previos al desarrollo conceptual

Para el desarrollo de la interfaz biométrica neurosensorial propuesta, que tiene como objetivo interpretar la actividad cerebral a través de la energía de potencial bioeléctrico generado por las neuroseñales para controlar periféricos electrónicos y mecánicos, se consideró requerimientos técnicos que fueron determinados por la investigación INIAC e INIAPCE en sus resultados, los mismos que son base de desarrollo de la interfaz que consiguió mover la silla de ruedas a través de medidas de meditación y atención, con lo cual se observó la necesidad de una interfaz más robusta y con un algoritmo ajustable a más variables capaz de controlar efectivamente los dos ejes de la misma y que se adapte a los entornos de trabajo que demanden futuros desarrollos sin ahondar en conocimientos profundos, permitiendo que estudiantes con conocimiento básico de electrónica puedan usarla para generar aplicaciones útiles en corto tiempo.

3.3. Identificación de requerimientos del cliente

Este proceso se divide en dos fases de desarrollo, la primera busca generar una solución electrónica para controlar periféricos electromecánicos a través de la comprensión de la actividad cerebral. Y la segunda en poder generar una herramienta que permita desarrollar aplicaciones neurotecnologías como por ejemplo, proyectos encaminados a mejorar la condición de vida de personas con discapacidades físicas.

- Planteamiento del cliente

A continuación se describen las necesidades generadas por las investigaciones INIAC e INIAPCE.

- ✓ Estabilidad en la señal.
- ✓ Adaptabilidad a varias aplicaciones.
- ✓ Independencia en su funcionamiento, es decir que el usuario no tenga un involucramiento constante en el funcionamiento del dispositivo.
- ✓ Operatividad bajo demanda mental del usuario, es decir que funcione únicamente bajo la estimulación cerebral del usuario.
- ✓ Fácil mantenimiento.
- ✓ Tiempo de uso ajustable a vida de sus baterías.

- Generación de soluciones

Se describe las posibles soluciones a las necesidades planteadas por el cliente

- ✓ Descripción de los estados cerebrales.
- ✓ Fácil extracción de datos.
- ✓ Utilización de métodos no invasivos.
- ✓ Adaptabilidad a diferentes sistemas.
- ✓ Respuesta en tiempo considerable.
- ✓ Interfaz de usuario fácil de entender y manejar.
- ✓ Sistema de control remoto para controlar a distancia.
- ✓ Sistema de adquisición de datos de fácil extracción.

3.4. Parámetros de proceso de diseño electrónico del prototipo

Las especificaciones preliminares del diseño están pensadas en respuesta a las necesidades del cliente como su objetivo primordial, es decir, usando técnicas de QFD(Despliegue de función de calidad) asociado al modelo "Hoshin Kanri" para otorgar estratégicamente la calidad requerida usando procesos de máxima efectividad en el desarrollo del prototipo, de esta forma podemos asegurar que éste prototipo una vez testado por el cliente seguirá su proceso de mejora en función de la calidad y la reducción de costos. (Hutchins, 2008)

3.5. Generación de conceptos

Los conceptos se generan mediante dos procesos, una descomposición funcional para describir el proyecto y una exploración sistematizada con la técnica Product Breakdown Structure para describir las etapas de desarrollo del proyecto.

3.6. Descomposición funcional

Tomando en cuenta las necesidades que han sido planteadas por el cliente y el problema planteado en la investigación, se realiza una descomposición funcional de todas las partes involucradas para desarrollar metodológicamente el prototipo, así tenemos la siguiente figura. (Ver Anexo 3.1)

3.7. Exploración Sistematizada

Para detallar las etapas que componen el proyecto sistematizadamente se utilizó la técnica del PBS el cual a través de un mapa conceptual pretende lograr obtener un adecuado funcionamiento del prototipo, tal cual las demandas del cliente de acuerdo a los criterios técnicos y requerimientos establecidos mediante el QFD. (Ver Anexo 3.2)

3.8. Obtención de materiales primordiales

El dispositivo se lo puede conseguir en la tienda online Aliexpress, la cual muestra en la página referenciada el producto TGAM1. (Aliexpres, 2014), para esta investigación se usa un módulo proporcionado por proyecto INIAPCE inventariado con el numero 82229 el mismo que al término de la investigación será devuelto al Ing. Roberto García para los fines pertinentes.

3.9. Descarga de información relevante al producto

En la web de NeuroSky podemos encontrar información necesaria para comenzar a trabajar con el dispositivo, a continuación se muestra la referencia al vínculo de descarga del KDT, MDT y varias herramientas adicionales: <http://store.NeuroSky.com/>

3.10. Descarga de Kit de herramientas de desarrollo de NeuroSky

El KDT puede ser descargado de forma gratuita en la siguiente página <http://store.NeuroSky.com/products/developer-tools-2-1>.

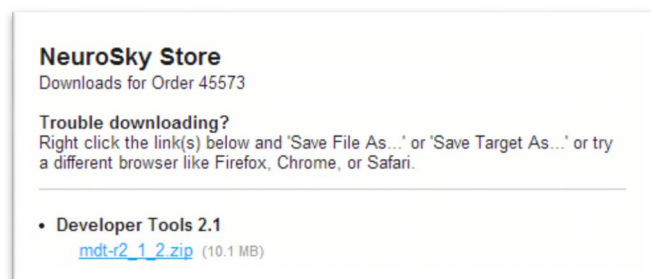


Figura 3.1 Orden KDT

Fuente: Print Screen de Google Chrome.

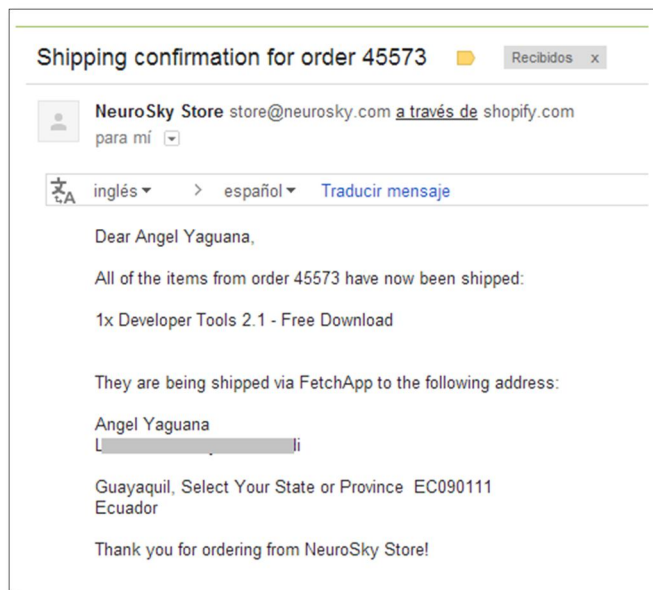


Figura 3.2 Email recibido de confirmación de orden de compra.

Fuente: Print Screen del correo electrónico del autor.

3.11. Descarga e instalación de herramientas adicionales secundarias

El Kit de herramientas de desarrollo es otro de los elementos más necesarios para el desarrollo de esta investigación ya que provee de toda la información pertinente al sensor TGAM1 y manuales para su uso óptimo. Puede ser comparado en la siguiente página: <http://store.NeuroSky.com/products/mindset-research-tools>.

Esta herramienta no está presente en el adjunto digital debido a los derechos de autor y el costo de la misma, sin embargo, se adjunta un conjunto de herramientas que suplen con igual eficiencia este Kit y de forma gratuita. Estas herramientas pueden ser halladas en el CD adjunto.

3.12. Dispositivo de EEG ThinkGear TGAM1

La información a continuación mostrada es un abstracto tomado directamente del archivo adjunto TGAM Datasheet.pdf, (Corp., N., 2009)

3.13. Descripción del dispositivo ThinkGear

El dispositivo que se contempla como medio de interfaz entre el cerebro y la computadora es el módulo ThinkGear AM de NeuroSky el mismo que cumple con el requerimiento de ser un método no invasivo al mismo tiempo de tener un grado de confiabilidad del 98% y ser además el dispositivo más usado a nivel de desarrollo de aplicaciones para EEG en varias universidades alrededor del mundo.

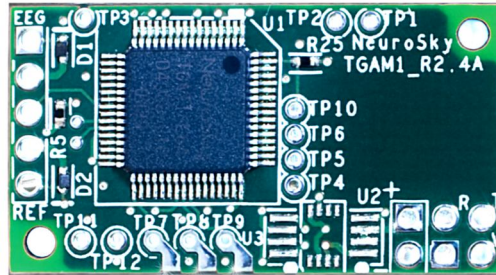


Figura 3.3 Módulo ThinkGear de NeuroSky.

Fuente: Mindset_Communications_Protocol.pdf

Antecedentes técnicos

Descripción Funcional del módulo TGAM: Módulo ASIC ThinkGear

- Se conecta directamente a un electrodo seco
- Tiene un canal de EEG con tres contactos: EEG , REF , y GND
- Su ajuste incorrecto detecta a través de la advertencia "Mala calidad de la señal " si se debe restablecer la comunicación con la cabeza del usuario
- Está provisto de la tecnología de filtrado avanzado con alta inmunidad al ruido
- Tiene bajo consumo de energía adecuado para aplicaciones impulsadas por batería portátiles
- Consumo de energía máximo 15mA a 3.3 V
- Salida de datos en bruto del EEG a 512 bits por segundo

Descripción Física.

- Entrega de una señal eléctrica generada por la onda cerebral tomada en el punto FP1, la cual recibe el nombre de RAW.

- Procesamiento de las señales y entrega de los espectros producto de las señales del EEG (Alfa, Beta, etc.)
- Entrega de los valores eSense para la obtención de Atención, meditación.
- Dato PSQ, el cual se entrega para permitir observar un análisis de calidad de la señal (se puede utilizar para detectar un mal contacto y si el dispositivo está fuera de la cabeza)
- Detección de parpadeo físico (solo mediante el TGDC).
- Dimensiones (máx.) 2.79cm x 1.52cm x 0.25cm
- Peso (máx.) 130 mg

Especificaciones técnicas de adquisición.

- 512 bits por segundo frecuencia de muestreo
- Rango de frecuencia de 3-100 Hz
- Protección ESD: Contacto de 4 kV de descarga, descarga 8 kV aire
- Consumo de energía máximo: 15mA@3.3V
- Tensión de funcionamiento 2.97 ~ 3.63V
- Salida y velocidad de transmisión.
 - ✓ 1200 baudios con eSense, ePower, (alfa, beta, etc.) y PSQ.
 - ✓ 9600 baudios con eSense, ePower, (alfa, beta, etc.) y PSQ.
 - ✓ 57600 baudios con eSense, ePower, (alfa, beta, etc.), PSQ y RAW

Electrodos, requerimientos técnicos.

- Superficie máxima de ~ 150mm².
- El material del electrodo debe ser Ag o AgCl, acero inoxidable, oro, o plata.
- El electrodo de EEG debe estar situado encima del ojo izquierdo o derecho de la frente, es decir el punto FP1.
- Los electrodos de tierra y referencia deben estar ubicados detrás de la oreja o en el lóbulo A0/A1.
- Deben tener suficiente presión para evitar el movimiento, con un mínimo de 0,8 PSI.
- Longitud de menos de 12 pulgadas, el largo, cuanto mayor es su longitud mayor es la susceptibilidad al ruido.
- Deben tener blindaje (no es necesario para el electrodo de referencia a tierra).
- El calibre del cable debe ser más delgado que AWG28.

Dibujo mecánico y espesor de la placa

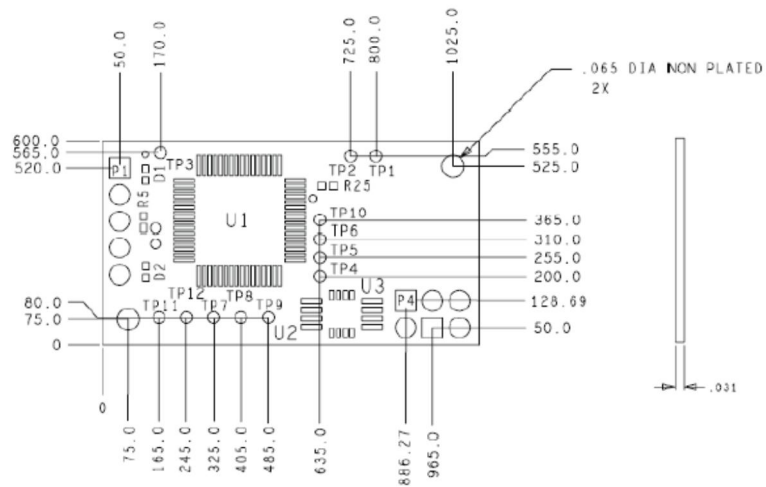


Figura 3.4 Dibujo mecánico de la placa del dispositivo ThinkGear de NeuroSky.
Fuente: Mindset_Communications_Protocol.pdf

Ubicación de conectores en la placa

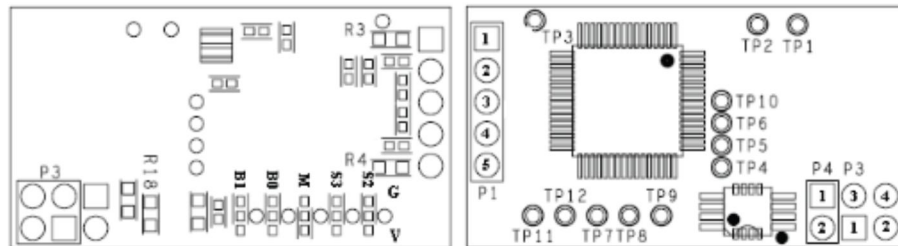


Figura 3.5 Vistas superior e inferior de la placa TGAM.
Fuente: Mindset_Communications_Protocol.pdf

PinOut del módulo TGAM1

Puerto 1. Electrodo.

- Pin1 EEG.
- Pin2 EEG blindaje.
- Pin3 Electrodo de referencia de tierra.
- Pin4 Blindaje de Electrodo de referencia.
- Pin5 Electrodo de referencia "REF".

Puerto 4. Energía.

- Pin1 VCC
- Pin2 Tierra

Puerto 3. Puerto serial UART.

- Pin1 GND
- Pin2 VCC
- Pin3 Rx
- Pin4 Tx.

Configuración física de funcionamiento del módulo TGAM

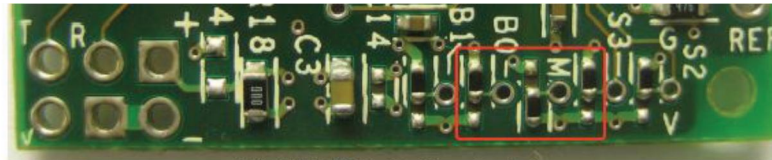


Fig. 4 TGAM1's Configuration Pads

BR1	BR0	Function
GND	GND	9600 Baud with Normal* Output Mode
GND	VCC	1200 Baud with Normal* Output Mode
VCC	GND	57.6k Baud with Normal* + Raw Output Mode
VCC	VCC	N/A

* Normal Output mode includes the following output: poor quality value, EEG Value, attention value, and meditation value.

Figura 3.6 Configuración física de funcionamiento del módulo TGAM.
Fuente: Mindset_Communications_Protocol.pdf

Ejemplo esquemático de uso en sistema BCI

Ambos electrodos, EEG y REF son conectados con cables blindados a TGAM1 y ambos blindajes son conectados a la tierra de TGAM1. La salida UART del módulo es conectada al puerto UART de un procesador.

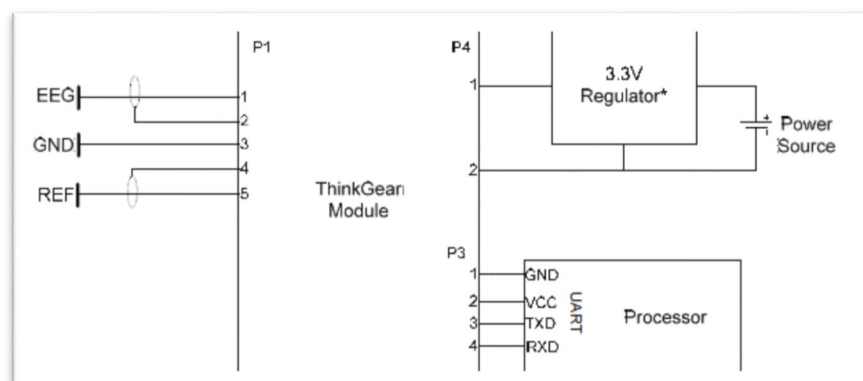


Figura 3.6 Esquema básico del BCI de NeuroSky.
Fuente: Mindset_Communications_Protocol.pdf

Descripción de tramas de datos de salida del módulo TGAM1

- Estructura del paquete de la trama

La trama es enviada vía comunicación UART, a 57600b, 8, 1, n y puede ser recibida desde cualquier dispositivo de comunicación con servicio de Puerto Serial COM. Cada paquete enviado posee la siguiente estructura:

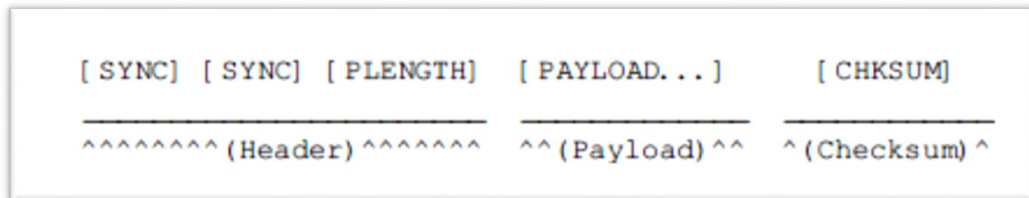
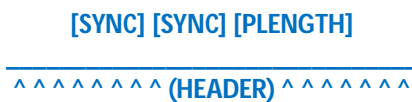


Figura 3.8 Estructura de un paquete del dispositivo TGAM1.

Fuente: Mindset_Communications_Protocol.pdf

La cabecera o "HEADER" de un paquete se compone de 3 bytes: dos sincronización [SYNC] bytes (0xAA 0xAA), seguido por un [pLength] (longitud de carga útil) de bytes:



Los dos [SYNC] bytes se utilizan para indicar el comienzo de un nuevo paquete que llega y son bytes con el valor 0xAA (decimal 170). La sincronización consiste en dos bytes de longitud, en lugar de sólo uno, para reducir la posibilidad de que [SYNC] (0xAA) bytes que ocurren dentro del Paquete podrían confundirse con el principio de un paquete. A pesar de que todavía es posible para dos datos consecutivos [SYNC] bytes a aparecer dentro de un paquete el [PLength] y [CHKSUM] combinado aseguran que es un paquete sincronizado "mis-sync'd Packet" tal que nunca será accidentalmente interpretado como un paquete válido.

El [pLength] byte indica la longitud en bytes de la sección del paquete de datos de carga útil [PAYLOAD...] y puede ser cualquier valor desde 0 hasta 169.

Cualquier valor más alto indica un error (pLength demasiado grande). Longitud completa de los paquetes siempre será [pLength] + 4.

La sección “[PAYLOAD...]” tiene un rango de hasta 169 bytes de longitud, mientras que cada uno de [SYNC], [pLength], y [CHKSUM] son un solo byte cada uno. Esto significa que un paquete válido completo es un mínimo de 4 bytes de longitud (es posible si la carga útil de datos es de cero bytes de longitud, es decir, vacío) y un máximo de 173 bytes de largo (es posible si la carga de datos es el máximo de 169 bytes de longitud).

Tabla 3.2 Estructura y definición de códigos del Data Payload.

Comando	Descripción
AA[SYNC]	Sincroniza el sensor TGAM1 con la obtención de señales de actividad cerebral.
0x02	[POOR SIGNAL] Es el código que indica que comienza a medir el valor de la señal, cuyo valor se determina por el número a continuación, el cual es un valor hexadecimal que puede según el caso variar desde 0 hasta 200, siendo un valor mayor a 0 el más sutil y mínimo indicador de contaminación por ruido, y el valor más alto “200” el cual indica que no hay contacto con la piel. La baja calidad en la señal pierde ser causada por varios motivos entre los cuales por ejemplo un mal contacto del sensor tierra o referencia con la piel del sujeto, un excesivo movimiento del sujeto en áreas cercanas al sensor, un ruído electrostático ambiental, un contacto de la persona con superficies de referencia tierra y ruidos biométricos de tipo non-EEG como por ejemplo EMG, EKG/ECG, EOG, y artefactos en la señal EEG superiores al rango determinado por el parpadeo de los ojos.
0x 04	[eSense Attention] Es el código que indica la lectura de atención de acuerdo a los parámetros establecidos por eSense los cuales pueden variar desde 0 a 100, estos indican que cualquier valor en el rango de 20 a 40 son considerados como “reducidos” niveles de atención y menores a estos son “muy reducidos”, asimismo de 40 a 60 son considerados “neutrales” y son similares a los valores “baselines” convencionalmente establecidos en EEG, mientras que valores de 60 a 80 son considerados “ligeramente elevados” y pueden ser interpretados como niveles posiblemente altos de atención en personas normales, finalmente, los valores que exceden 80 hasta 100 son considerados “elevados”, significando que son de muy altos niveles de atención.
0x 05	[eSense Meditation] Es el código que indica la lectura de atención de acuerdo a los parámetros establecidos por eSense, al igual que el anterior tiene la misma varianza y rango de clasificación pero sus valores no describen niveles sino estados mentales como “calma” o “relajación”. Hay que tener en cuenta que estos niveles de estados

	son considerados "elevados", significando que son de muy altos niveles de atención.
0x 05	[eSense Meditation] Es el código que indica la lectura de atención de acuerdo a los parámetros establecidos por eSense, al igual que el anterior tiene la misma varianza y rango de clasificación pero sus valores no describen niveles sino estados mentales como "calma" o "relajación". Hay que tener en cuenta que estos niveles de estados mentales no son niveles físicos tal que aun relajando todos los músculos del cuerpo ni implica que inmediatamente resulte en un valor de meditación alto, más aun cerrando los ojos y amenorando la frecuencia de respiración se puede obtener de manera inmediata valores de relajación que reflejan la oxigenación del cerebro y el cambio en el nivel de ondas theta.
0x 20	Este número representa el tamaño del vector, el cual está definido por [EEG Power LENGTH].
0x 81	[ASIC_EEG Powers] Es el código que indica la lectura de los distintos tipos de señales que entrega el módulo TGAM1 (Delta, Theta, Low Alpha, High Alpha, Low Beta, High Beta, Low Gamma, High Gamma), donde se lee el valor de cada una respectivamente comenzando después del numero seguido al 81.
0x83	[RAW Wave Value] Este valor es de tipo 16Bits, es decir consiste de 2 bytes y representan n único valor de onda muestreada. Este valor contiene enteros con signo en el rango de -32768 a 32767. El primer byte del valor representa el orden alto de bits de los 2 valores complementarios y el segundo representa el orden bajo de bits, la reconstrucción de estos valores se pueden visualizar en el código adjunto de la interfaz y se puede claramente apreciar su reconstrucción mediante algoritmos de "Big-Endian".

Fuente: Mindset_Communications_Protocol.pdf

Finalmente, la sección [CHKSUM] es un byte que debe ser utilizado para verificar la integridad de los datos de carga útil del paquete. Este es definido por:

La suma de todos los bytes de datos de carga útil del paquete. Obteniendo los 8 bits más bajos de la suma realizando la conjugada inversa del bit (complemento inverso) en esos 8 bits más bajos.

Un receptor que recibe un paquete debe utilizar esos 3 pasos para calcular la suma de comprobación de la carga útil de datos que se recibieron, y luego compararlo con el [CHKSUM]. Si el valor calculado es [CHKSUM] el paquete tiene una carga valida, si estos valores no coinciden, el paquete completo debe ser desechados como no válido.

Ejemplo de paquete enviado en una trama típica de datos, contenida esencialmente por los bytes [SYNC], [PLENGTH], y [CHKSUM]:

“AAA200200831800009400004200000B00006400004D00003D000007000005040D053D34”

Tabla 3.3 Contenido del paquete capturado.

byte:	valor	[CODE]
[0]	0xAA	[SYNC]
[1]	0xAA	[SYNC]
[2]	0x20	[PLENGTH]
[3]	0x02	[POOR_SIGNAL]
[4]	0x00	(0/200)
[5]	0x83	[ASIC_EEG_POWER_INT]
[6]	0x18	[VLENGTH] 24 bytes
[7]	0x00	(1/3) Inicio de Delta bytes
[8]	0x00	(2/3)
[9]	0x94	(3/3) Fin de Delta bytes
[10]	0x00	(1/3) Inicio de Theta bytes
[11]	0x00	(2/3)
[12]	0x42	(3/3) Fin de Theta bytes
[13]	0x00	(1/3) Inicio de Low-alpha bytes
[14]	0x00	(2/3)
[15]	0x0B	(3/3) Fin de Low-alpha bytes
[16]	0x00	(1/3) Inicio de High-alpha bytes
[17]	0x00	(2/3)
[18]	0x64	(3/3) Fin de High-alpha bytes
[19]	0x00	(1/3) Inicio de Low-beta bytes
[20]	0x00	(2/3)
[21]	0x4D	(3/3) Fin de Low-beta bytes
[22]	0x00	(1/3) Inicio de High-beta bytes
[23]	0x00	(2/3)
[24]	0x3D	(3/3) Fin de High-beta bytes
[25]	0x00	(1/3) Inicio de Low-gamma bytes
[26]	0x00	(2/3)
[27]	0x07	(3/3) Fin de Low-gamma bytes
[28]	0x00	(1/3) Inicio de Mid-gamma bytes
[29]	0x00	(2/3)
[30]	0x05	(3/3) Fin de Mid-gamma bytes

Fuente: Mindset_Communications_Protocol.pdf

CAPITULO 4

PRUEBAS DE CAMPO CON MODULO THINKGEAR DE NEUROSKY

En este capítulo se desarrolla una prueba conceptual de campo para demostrar el funcionamiento del módulo TGAM1 comercializado por NeuroSky.

4.1. Prueba de campo al módulo TGAM1

4.1.1. Objetivo de la prueba

Analizar las tramas de datos que envía el módulo TGAM1 e identificar los códigos del protocolo de NeuroSky.

4.1.2. Materiales

- Dispositivo BCI: IBNS con módulo TGAM1.
- PC con Windows 8.

4.1.3. Procedimiento

Se conectó el Modulo TGAM1 del dispositivo IBNS (desarrollado por el Autor para el proyecto INIAPCE) mediante el puerto serial COM 13 con una configuración 57600,b, 8, 1, n y así poder visualizar los datos mediante el Terminal serial Hércules.

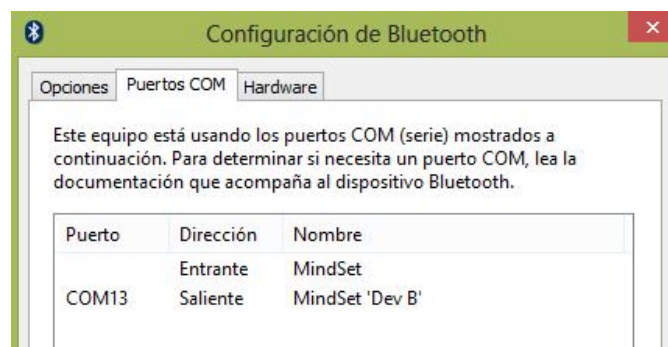


Figura 4.1 Puerto para enlace vía Serial COM13 hacia el dispositivo IBNS

Fuente: Print Screen de Ventana de Configuración de Bluetooth.

De esta prueba de campo se obtuvieron los siguientes resultados:

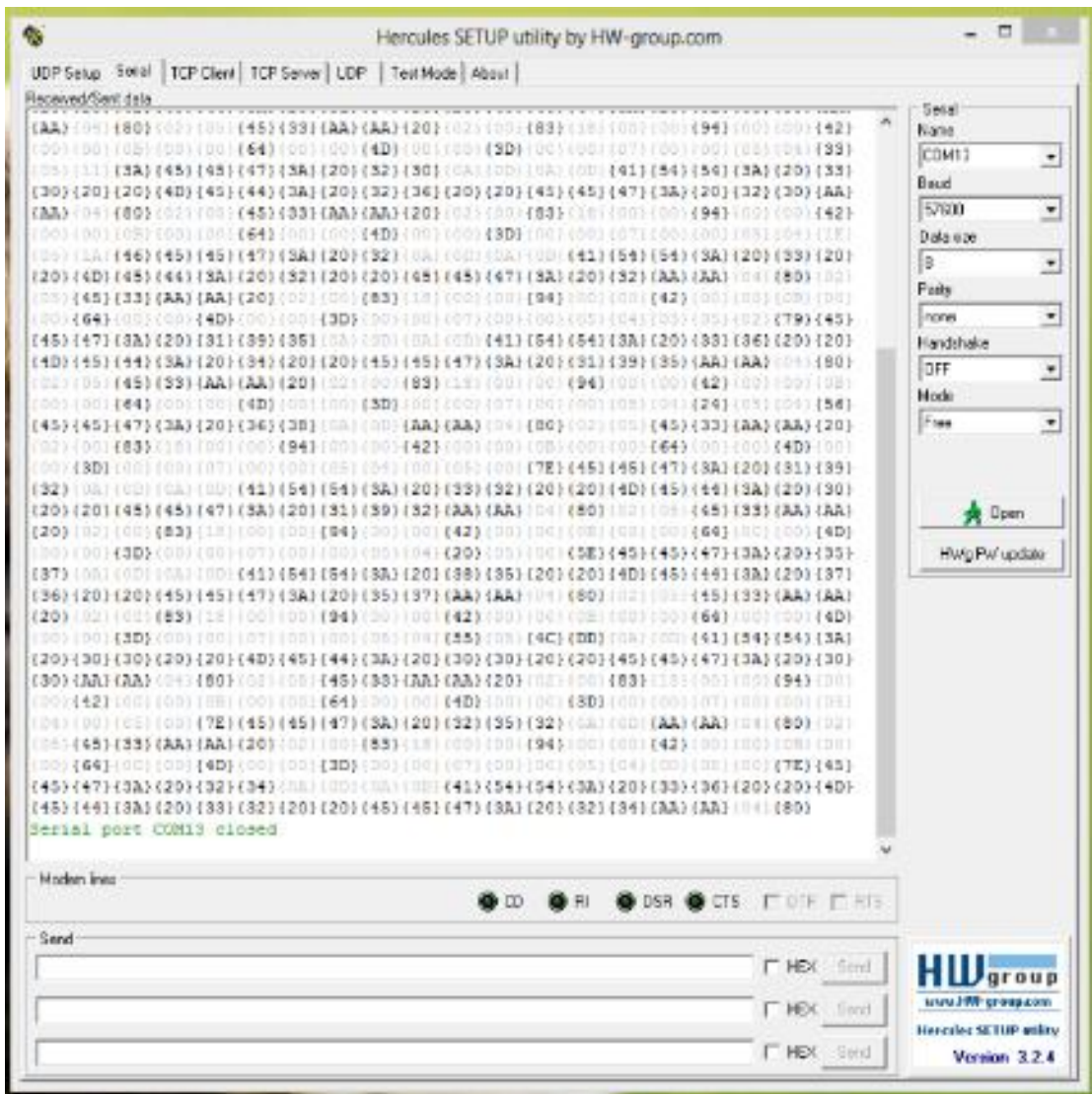


Figura 4.2 Escucha de paquetes de datos desde el dispositivo TGAM1

Fuente: Print Screen de Hércules SETUP versión 3.2.4.

4.1.4. Resultados

Como se puede apreciar el terminal serial registro una data continua de la que los valores recibidos no podían ser identificados fácilmente debido a su alta tasa de transferencia, por lo que tomando una muestra de todo el contenido del buffer se pudo identificar los siguientes patrones:

Tabla 4.1 Recorte de una sección del paquete recibido en HEX:

```

{AA}{AA}{04}{80}{02}{05}{45}{33}{AA}{AA}{20}{02}{00}{83}{18}{00}{00}{94}{00}{00}
{00}{42}{00}{00}{0B}{00}{00}{64}{00}{00}{4D}{00}{00}{3D}{00}{00}{07}{00}{00}{05}{
04}{1B}{05}{18}{4B}{45}{45}{47}{3A}{20}{39}{34}{0A}{0D}{AA}{AA}{04}{80}{02}{0
5}{45}{33}{AA}{AA}{20}{02}{00}{83}{18}{00}{00}{94}{00}{00}{42}{00}{00}{0B}{00}{
00}{64}{00}{00}{4D}{00}{00}{3D}{00}{00}{07}{00}{00}{05}{04}{00}{05}{00}{7E}{45}
{45}{47}{3A}{20}{34}{30}{0A}{0D}{0A}{0D}{41}{54}{54}{3A}{20}{36}{30}{20}{20}{4
D}{45}{44}{3A}{20}{35}{33}{20}{20}{45}{45}{47}{3A}{20}{34}{30}{AA}{AA}{04}{80}
{02}{05}{45}{33}{AA}{AA}{20}{02}{00}{83}{18}{00}{00}{94}{00}{00}{42}{00}{00}{0
B}{00}{00}{64}{00}{00}{4D}{00}{00}{3D}{00}{00}{07}{00}{00}{05}{04}{3C}{05}{35}{
0D}{45}{45}{47}{3A}{20}{35}{31}{0A}{0D}{0A}{0D}{41}{54}{54}{3A}{20}{37}{36}{2
0}{20}{4D}{45}{44}{3A}{20}{36}{38}{20}{20}{45}{45}{47}{3A}{20}{35}{31}{AA}{AA}
{04}{80}{02}{05}{45}{33}{AA}{AA}{20}{02}{00}{83}{18}{00}{00}{94}{00}{00}{42}{0
0}{00}{0B}{00}{00}{64}{00}{00}{4D}{00}{00}{3D}{00}{00}{07}{00}{00}{05}{04}{4C}{
05}{44}{EE}{45}{45}{47}{3A}{20}{32}{30}{35}{0A}{0D}{0A}{0D}{41}{54}{54}{3A}{20}
{35}{31}{20}{20}{4D}{45}{44}{3A}{20}{31}{37}{20}{20}{45}{45}{47}{3A}{20}{32}{3
0}{35}
    
```

Fuente: El Autor, Angel Yaguana.

Como se puede visualizar se han identificado 4 paquetes con valores eSense y ePower, analizamos a continuación el primero ellos y generalizamos conclusiones.

Tabla 4.2 Primer paquete identificado:

```

{AA}{AA}{20}{02}{00}{83}{18}{00}{00}{94}{00}{00}{42}{00}{00}{0B}{00}{00}{64}{
00}{00}{4D}{00}{00}{3D}{00}{00}{07}{00}{00}{05}{04}{1B}{05}{18}{4B}
    
```

Fuente: El Autor, Angel Yaguana.

Tabla 4.3 Análisis de la trama recibida del módulo TGAM1 en el terminal serial del PC.

Comando de identificación	Valor HEX recibido	Valor Decimal
[SYNC]	AA	170
[SYNC]	AA	170
[PLENGTH]	20	32
[POOR_SIGNAL]	02	2
(0/200)	00	0
[ASIC_EEG_POWER_INT]	83	131
[VLENGTH] 24 bytes	18	24
(1/3) Inicio de Delta bytes	00	0
(2/3)	00	0
(3/3) Fin de Delta bytes	94	148
(1/3) Inicio de Theta bytes	00	0
(2/3)	00	0
(3/3) Fin de Theta bytes	42	66
(1/3) Inicio de Low-alpha bytes	00	0
(2/3)	00	0
(3/3) Fin de Low-alpha bytes	0B	11
(1/3) Inicio de High-alpha bytes	00	0
(2/3)	00	0
(3/3) Fin de High-alpha bytes	64	100
(1/3) Inicio de Low-beta bytes	00	0
(2/3)	00	0
(3/3) Fin de Low-beta bytes	4D	77
(1/3) Inicio de High-beta bytes	00	0
(2/3)	00	0
(3/3) Fin de High-beta bytes	3D	61
(1/3) Inicio de Low-gamma bytes	00	0
(2/3)	00	0
(3/3) Fin de Low-gamma bytes	07	7
(1/3) Inicio de Mid-gamma bytes	00	0
(2/3)	00	0
(3/3) Fin de Mid-gamma bytes	05	5
[ATTENTION] eSense	04	4
eSense, Nivel de atención	1B	27
[MEDITATION] eSense	05	5
eSense Nivel de Meditación	18	24
[CHKSUM]	4B	75

Fuente: El Autor, Angel Yaguana.

4.1.5. Discusión

Como se pudo observar, la densidad de los datos enviados por el módulo hace difícil el reconocimiento de los valores eSense y ePower sin una herramienta apropiada y como es de suponerse se requiere de un conocimiento profundo del funcionamiento de sus tramas de paquetes para poder identificarlos de manera óptima, esto sin la ayuda de una aplicación de PC o en un uC es imposible de realizar en tiempo real.

CAPITULO 5

ANALISIS DEL KDT DE NEUROSKY

En este capítulo, se analiza conceptualmente las herramientas proporcionadas por NeuroSky para desarrollar aplicaciones BCI tanto para PC como para uC y poder adquirir la suficiente destreza para emplearlos adecuadamente.

5.1. Análisis descriptivo del KDT proporcionadas por NeuroSky

El conjunto de herramientas de desarrollo KDT (Kit Development Tools) de NeuroSky, otorga al desarrollador los recursos necesarios para crear y publicar aplicaciones y juegos capaces de tomar ventaja de la existente BCI ThinkGear, asimismo de los drivers y de los códigos de ejemplo que describen como desarrollar aplicativos en algunas plataformas de software de PC, celulares y microcontroladores.

5.2. Descripción de lenguajes compatibles con la DLL proporcionada

Los lenguajes directamente soportados son C, C++, C#, Java y J2ME. Cualquiera de estos está ampliamente documentado para comenzar a crear aplicativos usando el poder de la mente de un sujeto. (Corp., N., 2009).

5.2.1. Plataforma C/C++

En el caso del PC, la interfaz primaria para desarrollo de aplicaciones es la comunicación a través de la DLL y el API de ThinkGear EM Communication, la misma que hace extremadamente sencillo su uso para cualquier aplicación Win32 y su comunicación con el módulo. (Corp., N., 2009)

5.2.2. Plataforma C# y Java

Los paquetes para C # y Java (a través de JNI) se proporcionan, de manera que las aplicaciones escritas en esos idiomas también tienen acceso directo a las funciones proporcionadas por el archivo DLL de Win32 se ha descrito anteriormente para C/C++.

5.2.3. Plataformas y lenguajes opcionales

Además de los paquetes antes mencionados, es posible para prácticamente cualquier otra plataforma y lenguaje llamar a las funciones de la DLL Win32 para poder comunicarse con ThinkGear AM a través de su API de comunicación.

5.2.4. Windows Mobile, C/C++

Se proporciona un archivo DLL compilado para procesadores ARM para aplicaciones específicas para la plataforma Windows Mobile. La API es exactamente la misma que la API de Win32 que se ha descrito anteriormente, de manera que exportar entre plataformas es fácil.

5.2.5. Windows Mobile, C#

También se proporciona un C # contenedor delgado para la versión de Windows Mobile de la DLL, prácticamente el mismo previsto en Win32. A diferencia de la plataforma Win32, Windows Mobile no es generalmente soportado en Java JNI, por lo que no se proporciona una capa Java.

5.2.6. Symbian para Dispositivos Móviles

El MDT incluye un conjunto de clases Java para permitir que las aplicaciones Java escritas para que la plataforma Symbian para dispositivos móviles puedan comunicarse fácilmente con el TGAM1.

5.2.7. Unix, Linux y Mac OS

Plataformas UNIXTM, LinuxTM, y Mac OS X incluyen varias herramientas incorporadas en línea de comandos para la comunicación con el TGAM1.

La documentación incluida en el MDT proporciona ciertos ejemplos sobre cómo utilizar estas herramientas.

El código fuente para el análisis de los datos del TGAM1 se incluye como parte del KDT, el código fuente está escrito en ANSI C y está totalmente documentado, por lo que es posible el desarrollo de aplicativos en prácticamente cualquier plataforma, así mismo usando lenguajes de bajo nivel en el caso de microprocesadores y micro controladores para poder comunicarse y obtener Información del módulo TGAM1.

5.3. Elección de la plataforma más adecuada para el prototipo

La elección de una plataforma para soportar un desarrollo electrónico es un tema muy complejo, ya que involucra dos temas muy difíciles de abordar, por una parte el tema tecnológico el cual será la mejor combinación de elementos para que nuestra plataforma funcione correctamente y tenga la performance adecuada, y por otro lado el económico ¿cuánto me constara esa solución?, es aquí donde se produce un cruce entre la decisión tecnológica y la económica donde el objetivo es encontrar un punto medio entre una solución óptima desde el punto de vista tecnológico y a un costo que pueda ser absorbido por el desarrollador que en este caso es el estudiante.

Es por este motivo que es de suma urgencia la realización de un estudio que pudiera dar algunos lineamientos para la toma de decisión sobre la plataforma más adecuada en cada caso para los estudiantes de las carreras de Ingeniería de la Facultad Técnica. Por ahora, a falta de éste estudio, en el presente proyecto se evalúan las plataformas convencionalmente conocidas como C y C++ que fueron las asignadas en la materia de programación de nuestro pensum, analizando una combinación entre software licenciado y Open Source según la necesidad del proyecto y el mínimo cumplimiento con los estándares y tendencias mundiales para generar un diagnostico en base a sus herramientas, características y necesidades, tomando en cuenta los entornos de desarrollo sugeridos por NeuroSky mencionados anteriormente al inicio de este capítulo.

Tabla 5.1 Análisis de requerimientos básicos de diferentes entornos IDE para PC

Requerimiento	Microsoft Visual Studio	IDE Eclipse C++	IDE Processing	IDE Java NetBeans	IDE Python
Plataforma Open Source.		X	X	X	X
Soporte de funciones estándar C y algunas de C++.	X	X	X		
Fácil creación de librerías en C, Clases objetos y métodos en C++.	X	X	X		
Servicio de enlace al puerto de comunicación "Serial".	X	X	X	X	X
Manipulación de puertos de comunicación en alto y bajo nivel.		X	X		X
Sintaxis de programación similar a C/C++.	X	X	X		
Soporte amplio de bibliotecas Open Source.	X	X	X	X	X
Instalación del IDE en diferentes entornos.		X	X		X
Espacio en disco reducido y portable			X		X

Fuente: El Autor, Angel Yaguana.

Tabla 5.2 Análisis de requerimientos básicos de diferentes entornos IDE para uCs.

Requerimiento	Arduino/ Processing	PicC CCS	Mplab	Keil	IAR
Plataforma Open Source.	X	X	X	X	X
Soporte de funciones estándar C y algunas de C++.	X	X	X	X	X
Fácil creación de librerías en C, Clases objeto y métodos en C++.	X	X	X	X	X
Servicio de enlace al puerto de comunicación "Serial".	X	X	X	X	X
Manipulación de puertos de comunicación en alto y bajo nivel.	X	X	X	X	X
Sintaxis de programación similar a C/C++.	X	X	X	X	X
Soporte amplio de bibliotecas Open Source	X				
Instalación del IDE en diferentes entornos.	X	X	X	X	X
Espacio en disco reducido y portable	X				

Fuente: El Autor, Angel Yaguana

Comprobando los resultados obtenidos del análisis de las plataformas de desarrollo para PC y para Microcontroladores y teniendo en cuenta su similitud, portabilidad, estructura, características y soporte podemos apreciar que el software de desarrollo Open Source IDE Processing, desarrollado por el MIT, cuenta con todas las características requeridas sobre todo tomando en cuenta su portabilidad ya que no necesita ser instalado, sino más bien ejecutado desde cualquier ubicación local.

Arduino por otro lado es un entorno basado en Processing y Wiring que mantiene sus librerías en modo Open Source y sus shields en modo Open Hardware para que cualquier persona pueda acceder a ellos fácilmente con la finalidad de poder contribuir al crecimiento global en cuanto a conocimiento científico.

5.4. Breve descripción de la plataforma de código abierto Processing

Processing Development Environment (PDE). El entorno de desarrollo de Processing, consiste en un simple editor de texto para la escritura de código, un área de mensajes, una consola de texto, fichas para la gestión de archivos, una barra de herramientas con botones para las acciones comunes, y una serie de menús. Las opciones de los menús cambian de un modo a otro. El modo por defecto de Java se documenta aquí.

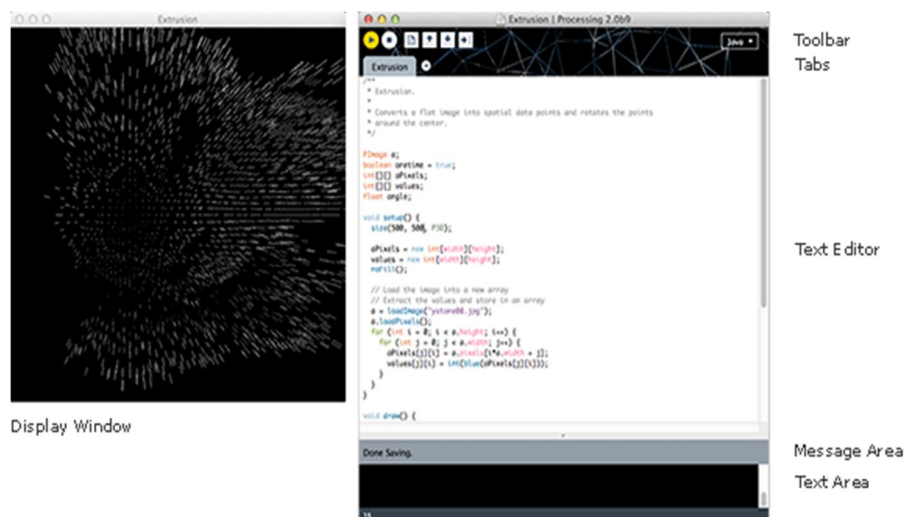


Figura 5.1 Plataforma de desarrollo IDE Processing.

Fuente: Print Screen de página de Processing.org.

5.5. Bocetos y Sketchbook

Todos los proyectos de Processing se llaman bocetos. Cada dibujo tiene su propia carpeta. El archivo principal de cada dibujo tiene el mismo nombre que la carpeta y se encuentra en el interior. Por ejemplo, si el dibujo se llama "Sketch_123", la carpeta para el boceto se llamará "Sketch_123" y el archivo principal se llama "Sketch_123.pde". La extensión de archivo PDE es un acrónimo para el entorno de desarrollo de Processing.

Una carpeta de croquis a veces contiene otras carpetas para archivos multimedia y otros códigos. Cuando se añade una fuente o imagen a un boceto seleccionando "Add File..." en el menú Sketch, se crea una carpeta "data". Los archivos también pueden ser añadidos a su esquema de procesamiento arrastrándolos en el editor de texto. Imagen y Música archivos arrastrados a la ventana de la aplicación se añadirán automáticamente a la carpeta "data" del dibujo actual. Todas las imágenes, fuentes, sonidos y otros archivos de datos cargados en el dibujo deben estar en esta carpeta.

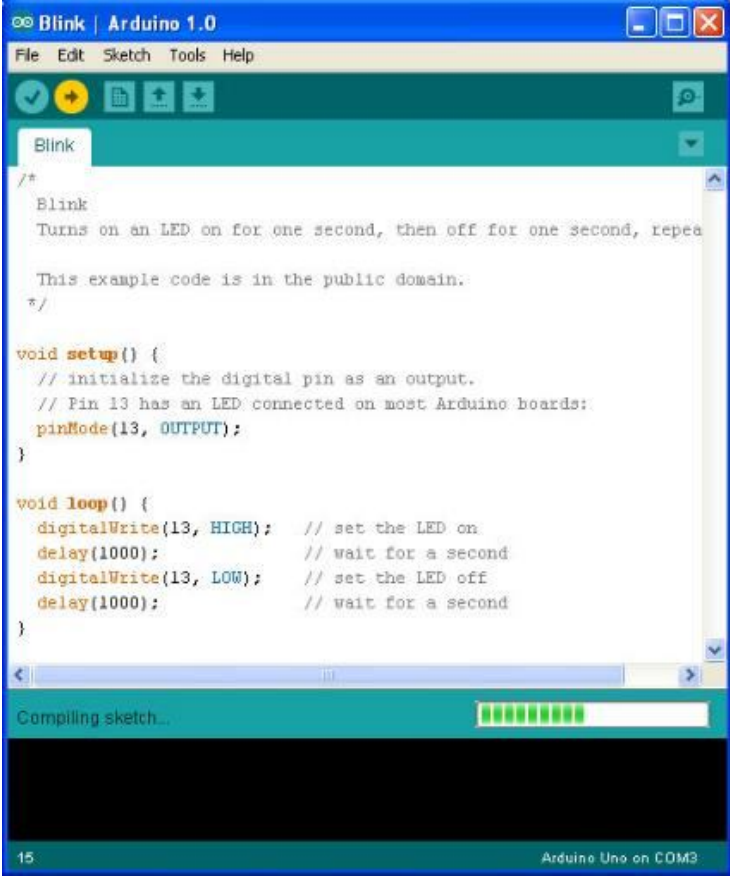
5.6. Adición de bibliotecas, herramientas y modos de ejecución

Processing 2.0 incluye un conjunto de nuevas características para hacer más fácil la instalación, actualizar y eliminar bibliotecas, herramientas y modos. Se puede simplemente agregar bibliotecas mediante la opción "Add Library...", herramientas seleccionando "Add Tool..." y modos seleccionando "Add modo...". No todas las bibliotecas disponibles se han convertido a aparecer en "Add Library...". Si una biblioteca no está ahí, tendrá que ser instalado manualmente.

5.7. Breve descripción de la plataforma de código abierto Arduino

Arduino es una plataforma electrónica Open-Source orientada a facilitar el desarrollo y creación de prototipos en proyectos multidisciplinarios.

Su entorno GUI permite desarrollar fácilmente aplicaciones basadas en la fabricación de firmwares para específicos microcontroladores Atmel y ARMs, tal como se puede apreciar en la figura 5.2 en la cual se prende y apaga un diodo Led a 1hz de frecuencia.



```
Arduino 1.0
File Edit Sketch Tools Help
Blink
/*
 * Blink
 * Turns on an LED on for one second, then off for one second, repea
 *
 * This example code is in the public domain.
 */

void setup() {
  // initialize the digital pin as an output.
  // Pin 13 has an LED connected on most Arduino boards:
  pinMode(13, OUTPUT);
}

void loop() {
  digitalWrite(13, HIGH); // set the LED on
  delay(1000);           // wait for a second
  digitalWrite(13, LOW); // set the LED off
  delay(1000);           // wait for a second
}

Compiling sketch...
15 Arduino Uno on COM3
```

Figura 5.2 Ejemplo Blink LED en el Entorno GUI IDE Arduino.

Fuente: Print Screen de IDE Arduino versión 1.0

Lenguaje de programación Arduino

Los microcontroladores con firmware de Arduino se programan mediante el uso de un lenguaje programación de alto nivel derivado de Processing, pero, también es posible utilizar otros lenguajes de programación debido a que Arduino usa un Bootloader para cargar los firmwares mediante el protocolo STK500, el cual soporta el formato serie de forma nativa.

5.7.1. Hardware Arduino

El hardware de Arduino, es un diseño modular y único en su arreglo de puertos que consiste en una placa base seccionada en un puerto de poder, un puerto de 6bits de entradas analógicas, un puerto de 8bits digital y uno de 6 bits de los cuales 3 son PWM y los demás son GPIO. Todos estos, conectados a un microcontrolador Atmel o ARM que por su sencillez y bajo coste que permiten el desarrollo de múltiples diseños. (Arduino, 2014)



Figura 5.3 Hardware Arduino Uno.
Fuente: Modulo Arduino, (Arduino, 2014)

5.7.2. Discusión de elección del Microcontrolador

En cuanto al microcontrolador que se puede usar para este proyecto podemos dar tolerancia según el espacio de memoria del programa ROM, el espacio de memoria RAM, el consumo de voltaje y su frecuencia de trabajo, es decir, podríamos estipular 2 tipos de microcontroladores con bootloader Arduino para este proyecto como base de desarrollo teniendo en cuenta que es una herramienta proyectada a ser un recurso de fácil acceso a los estudiantes para desarrollar aplicaciones embebidas que requieran el uso del módulo ThinkGear y que puede variar su tamaño de código según la aplicación a ser desarrollada. A continuación una breve comparación de los tipos de microcontroladores compatibles en su aspecto más básico:

Tabla 5.3 Características de módulos Arduino para migrar la interfaz según sus recursos y aplicaciones

Arduino	ATmega328	ATmega2560
Voltaje operativo	5 V	5 V
Voltaje de entrada recomendado	7 - 12 V	7 - 12 V
Voltaje de entrada límite	6 - 20 V	6 - 20 V
GPIO entrada y salida digital	14 - 6 PWM	54 - 14 PWM
PI de entrada analógica	6	16
Intensidad de corriente	40 mA	40 mA
Memoria Flash	32KB	128KB
Bootloader	2KB	4KB
EEPROM	1 KB	4 KB

Fuente: Módulos Arduino UNO y MEGA, (Arduino, 2014).

5.8. Análisis de las API de NeuroSky para tecnologías Atmel y C++

NeuroSky en el documento Mindset communications protocol, proporciona un ejemplo de código en C que puede ser tomado como referencia lógica en cualquier plataforma de desarrollo para capturar la trama de datos que el módulo TGAM1 envía. (Ver [Anexo 5.1](#)).

5.8.1. ThinkGearStreamParser API en C

El API ThinkGearStreamParser es una biblioteca que implementa el procedimiento de análisis descrito anteriormente y los resúmenes en dos funciones simples, por lo que el programador no tiene que preocuparse por análisis de paquetes y DataRows en absoluto. Todo lo que queda es para el programador para obtener los bytes del flujo de datos, meterlos en el analizador, y luego definir lo que la aplicación hace con los valores Value [] bytes de cada DataRow que se recibe y analiza.

El código fuente para el API ThinkGearStreamParser se proporciona como parte del MDT, y consta de un .h encabezado y un .c Fuente. Se implementa en puro ANSI C para la máxima portabilidad para todas las plataformas (incluyendo microprocesadores).

El uso de la API consta de 3 pasos:

- Definición de un administrador de datos (callback) que se encarga de actuar sobre los valores de datos, ya que son recibidos y analizados.
- Iniciar una estructura ThinkGearStreamParser llamando a la función THINKGEAR_initParser ().
- A medida que se recibe cada byte de la secuencia de datos, el programa pasa a la función THINKGEAR_parseByte (). Esta función llamará automáticamente a la función de manejador de datos definida en 1 cada vez que un valor de datos es analizado.

La referencia ThinkGearStreamParser.h, que es la documentación de la API que proporciona NeuroSky. (Ver Anexo 5.2)

5.9. Desarrollo de una GUI conceptual

A continuación se fabrica una GUI para demostrar el funcionamiento del módulo TGAM1 usando los Api proporcionadas por NeuroSky y el enlace al DLL del ThinkGear Connector.

5.9.1. Objetivo de la prueba

Determinar los umbrales de cada una de las variables eSense usando el API de comunicación propio de NeuroSky en las librerías de Processing.

5.9.2. Materiales, requerimientos de hardware y software

- Processing IDE 2.1 o superior.
- Librerías de Processing :
- NeuroSky y org.json (enlace al DLL de TGDC disponible en Library Manager desde Processing).
- ControlP5 (librería grafica disponible en Library Manager desde Processing).
- Arduinoscope (librería de PSD con enlace a Arduino, disponible en Library Manager desde Processing).
- ThinkGear Connector
- Bluetooth Clase 2, con servicio de Puerto Serial COM instalado en el PC.
- Módulo IBNS, encendido pareado al PC y totalmente cargado. Código de emparejamiento: 2912.

5.9.3. Informe del Procedimiento seguido en la prueba de concepto

- Se instaló el TGDC en una PC con Windows 8.

- Se emparejo la IBNS que contiene el módulo TGAM1 con el PC mediante el módulo Bluetooth, usando el código de emparejamiento 2912.
- Se descargó las librerías y documentación necesaria para realizar una rápida interfaz gráfica de usuario en Processing y así poder demostrar el funcionamiento de las API de NeuroSky visualizando las variables enviadas en sus paquetes de datos en tiempo real.
- Se creó el boceto IBNS_EEG_VISUALIZER, (ver Anexo 5.3).

5.9.4. Resultados

Una vez realizadas las líneas de programa necesarias procedemos a dar PLAY al boceto y tenemos el siguiente resultado:

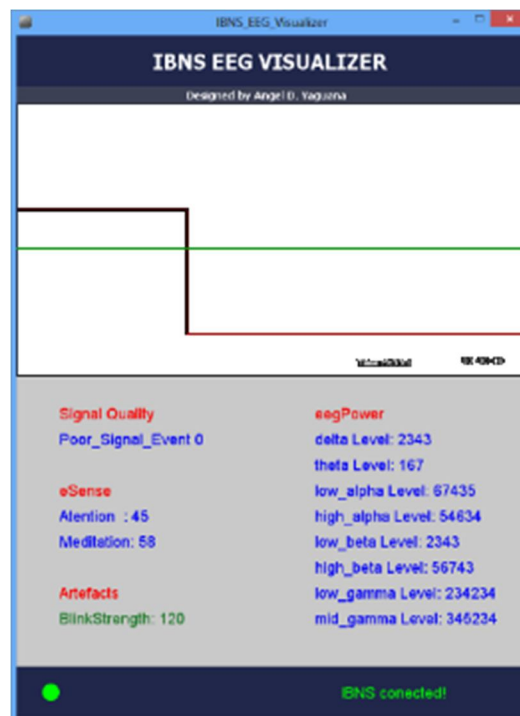


Figura 5.4 Reproducción del boceto de Processing.

Fuente: Print Screen de IDE Processing versión 2.1

Como se puede apreciar, el código implementado permite extraer del paquete de datos enviado por el módulo TGAM1 la información necesaria y real, apoyándonos en la DLL del TGDC, haciendo de éste un sencillo procedimiento sin la necesidad de ahondar en procesos de depuración de datos que serían necesarios si la captura de éstos fuese de manera directa.

5.9.5. Discusión sobre la prueba realizada

Una vez elaborada la GUI conceptual para demostrar el funcionamiento del módulo TGAM1 usando las API proporcionadas por NeuroSky y el enlace al DLL del ThinkGear Connector, podemos concluir que efectivamente los datos que son enviados usando el protocolo del módulo TGAM1 son mucho más sencillos de visualizar si están en un lenguaje comprensible y entendible, siendo así prácticamente vistos en tiempo real sin mayores esfuerzos.

5.10. Prueba de concepto al API proporcionado por NeuroSky

En la siguiente prueba utilizamos como guía el documento “Interfacing the Mindset with Arduino” como guía de procedimiento para diagnosticar el proceso de captura de datos mediante el servicio BT Serial COM en la recepción de paquetes enviados desde el módulo TGAM1 usando un módulo Arduino UNO.

5.10.1. Objetivo

Diagnosticar el funcionamiento de la captura de datos mediante el servicio BT Serial COM en la recepción de paquetes enviados desde el módulo TGAM1 usando un módulo Arduino UNO.

5.10.2. Materiales

- Módulo IBNS, totalmente cargado. Código de emparejamiento: 2912
- Arduino UNO o similar.
- Módulo BlueSMiRF con chip RN41 o similar en modo Maestro.
- Arduino IDE

5.10.3. Procedimiento

- Obtenidos todos los materiales procedemos con la configuración de cada uno de los módulos requeridos.
- Iniciamos el IDE Arduino, este puede ser descargado desde www.Arduino.cc/es/downloads.
- En el IDE Arduino: Archivo, Nuevo, Save As: SET_BT_Smirf.ino.

- En el BlueSMiRf revisamos que tipo de chip Bluetooth tenemos, en este caso es el RN-41D, este nos permite configurar la comunicación en modo Maestro de tal manera que se conecte automáticamente al dispositivo IBNS cuando este sea encendido, para ello realizamos un pequeño programa de configuración siguiendo el diagrama de las siguientes conexiones:

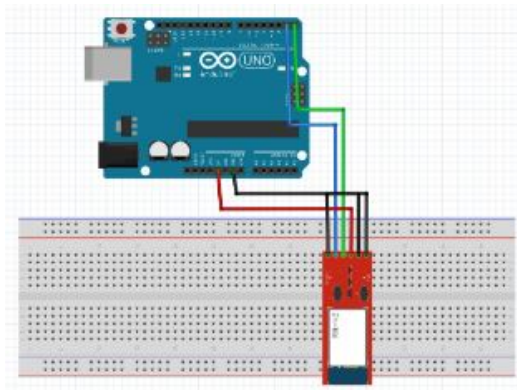


Figura 5.5 Boceto en Fritzing creado para diagramar las conexiones.

Fuente: Print Screen de IDE Fritzing versión 0.8.7

Para esta etapa de desarrollo como se puede apreciar en la figura anterior se usa el IDE educativo Fritzing V0.8.7 beta, el cual es un entorno muy intuitivo para generar diagramas de conexión con elementos básicos y módulos como por ejemplo Arduino, mbed, Basic Stamp, Picaxe, Snootlab, Atlas Scientific, Sparfun, etc. Este sencillo entorno permite generar esquemáticos y placas prototipo al mismo tiempo en que uno diagrama el boceto del protoboard tal como se muestra en las siguientes figuras respectivamente:

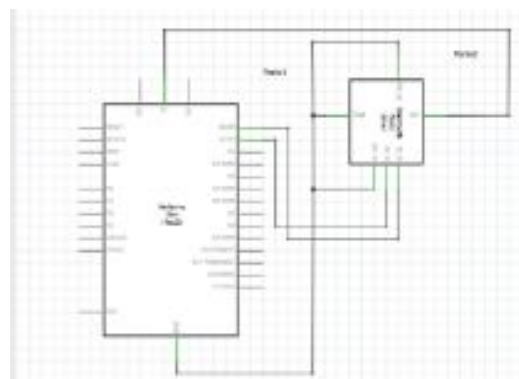


Figura 5.6 Esquema en Fritzing creado para diagramar las conexiones.

Fuente: Print Screen de IDE Fritzing versión 0.8.7

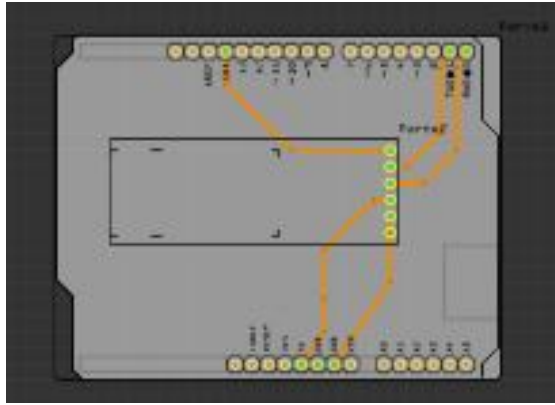


Figura 5.7 PCB en Fritzing creado para diagramar las conexiones.

Fuente: Print Screen de IDE Fritzing versión 0.8.7

El PCB generado por Fritzing no necesariamente debemos realizarlo pero si es de una gran guía para poder realizar un PCB más profesional que nos permita una conexión directa con el módulo Arduino a manera de un Shield Arduino.

Una vez realizado esto, procedemos a realizar un programa en el IDE Arduino para poder configurar de manera limpia y directa el módulo Bluetooth.

Se crea a continuación el boceto SET_BT_Smirf en IDE Arduino. (Ver Anexo 5.4)

Hecho esto ya podemos enlazarnos con nuestro dispositivo IBNS vía Bluetooth pero para poder leer los paquetes de datos enviados por el módulo TGAM1 aun debemos crear un firmware conceptual que nos permita captar las variables deseadas de entre todos los bytes enviados dentro del paquete de la trama.

Se crea a continuación el boceto Test_IBNS.ino en el IDE Arduino. (Ver [Anexo 5.5](#))

5.10.4. Resultados

A continuación se muestran las tramas recogidas por el módulo Arduino usando el módulo BT_Smirf para enlace de comunicación Bluetooth con Servicio de Puerto Serial al dispositivo IBNS mediante el módulo TGAM1.

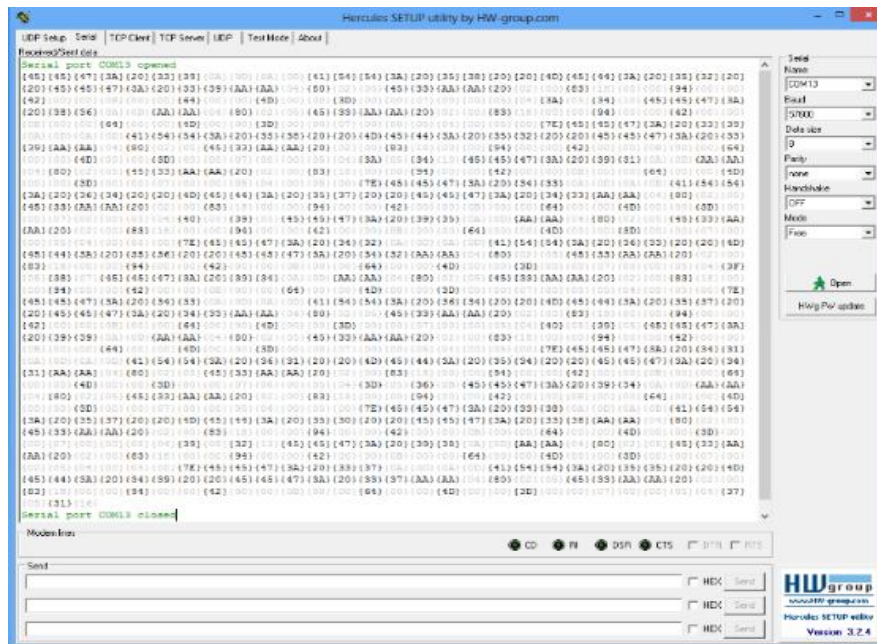


Figura 5.8 Escucha de paquetes de datos vía Serial COM.

Fuente: Print Screen de Hércules SETUP versión 3.2.4.

Como se puede apreciar el microcontrolador Atmega328 registro una data continua en la cual los valores recibidos no pueden ser identificados fácilmente debido a su alta tasa de transferencia, por lo que se procedemos a utilizar la interfaz GUI para poder visualizarlos de manera más sencilla.



Figura 5.9 Reproducción del boceto de Processing.

Fuente: Print Screen de IDE Processing versión 2.1

Básicamente el único cambio que se ha realizado para que el TGDC pueda capturar las tramas de los datos y validarlas desde el microcontrolador Arduino 328 es cambiar en su configuración el puerto al cual está vinculado como se muestra en la figura siguiente.

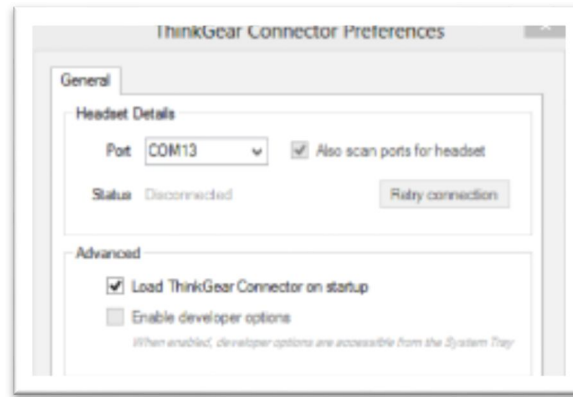


Figura 5.10 Configuración del TGDC.

Fuente: Print Screen de ThinkGear Connector Preferences versión 1.5.3829.28009

5.10.5. Discusión

Las Interfaz de programación de aplicaciones proporcionadas por NeuroSky para trabajar con su módulo TGAM1 mediante el DLL TGDC han demostrado ser muy útiles para poder recoger los datos ya sea mediante el Bluetooth de la PC o mediante un uC que sirva de puente de comunicación el cual podrá a futuro usar estos datos para crear una aplicación embebida.

CAPITULO 6

FABRICACION DEL FIRMWARE

6.1. Compatibilidad

En el presente capítulo se aplicaran los conocimientos adquiridos en el manejo de las Api de NeuroSky y de la experiencia obtenida en la prueba de concepto del uso de un uC Atmega328 para la creación de un firmware que siendo compatible con cualquier bootloader Arduino pueda migrarse a cualquier tipo de microcontrolador soportado por esta plataforma, los cuales se enumeran a continuación:

Tabla 6.1 Microcontroladores compatibles con el firmware a fabricarse.

Modelo	uC	Digital I/O	PWM	Flash
Due	AT91SAM3X8E	54	12	512Kb
Leonardo	ATmega32U4	20	7	32Kb
Uno - R3	ATmega328	14	6	32Kb
Mega 2560	ATmega2560	54	14	256Kb
Mini 05	ATmega328	14	8	32Kb
Pro Mini 3.3V/8MHz	ATmega328	14	6	32Kb
Pro Micro	ATmega32U4	12	5	32Kb

Fuente: Arduino, compatibilidad y modulos disponibles, (Arduino, 2014).

6.2. Firmware

A continuación se realiza la construcción del firmware usando el API de referencia de NeuroSky como resultado del trabajo de investigación que será el vínculo amigable para que un estudiante o inclusive un desarrollador profesional pueda generar aplicaciones embebidas que permitan interactuar con la interfaz IBNS y manejar periféricos electrónicos mediante estímulos mentales emocionales y artefactos. (Ver Anexo 6.1)

6.3.Pruebas de concepto del prototipo desarrollado

Una vez creado el firmware base con el cual podemos identificar la carga útil de cada uno de los paquetes enviados por el módulo TGAM1 durante un procedimiento EEG al sujeto de prueba, podemos discriminar la información para tomar de ésta solo lo que necesitamos.

6.3.1. Procedimiento

Se procede entonces a implementar un filtro para captar el parpadeo de los ojos del sujeto de prueba mediante la detección de un pico alto y su lapso de duración hasta el siguiente pico. Esta porción de código es implementada debido a que en las pruebas que se hizo con el DLL del TGDC se podía captar la intensidad del parpadeo.

Sin embargo, en todos los análisis de los datos que se hicieron a lo largo de la investigación no se pudo obtener un identificador 0x16 en ningún byte y es debido a que NeuroSky señala expresamente en el documento de Protocolo de comunicación Mindset.

“Note: Tis data value is currently only available via the TGCD and TGC APIs. It is not directly available as output from any current ThinkGear hardware. For TGCD, see the TG_DATA_BLINK_STRENGTH data type for use with the TG_GetValueStatus () and TG_GetValue () functions.” (NeuroSky, Inc., 2010)

Por lo tanto, se implementan las siguientes líneas en el anterior código para identificar un evento del tipo BLINK_STRENGTH. (Ver [Anexo 6.2](#)). Realizado esta mejora al firmware podemos proceder a realizar una prueba conceptual para determinar los umbrales de estados mentales eSense y del artefacto BLINK_STRENGTH.

Para esta etapa del diseño necesitamos compilar el firmware, cargarlo en el Atmega 328 y realizar un EEG mientras se toma nota de los datos visualizados en el puerto serial del PC al cual está conectado el módulo prototipado en el Arduino Uno.

Estos datos para su estudio, si es el caso, se adjuntan por completo en la versión digital de la presente investigación con los nombres Anexo 6.3 y Anexo 6.4 debido a su magna densidad, sin embargo, los resultados concretos de las dos observaciones al estudio EEG se los puede encontrar con su respectiva descripción para el primer sujeto y para el segundo adjuntos al final de este trabajo (ver Anexo 6.3 y Anexo 6.4)

6.3.2. Resultados

El firmware desarrollado es capaz de identificar los principales parámetros que un desarrollador necesita para realizar su aplicación usando neurotecnología con la posibilidad de identificar sus propios umbrales o hacer que estos sean calibrados dependiendo del sujeto que vaya a usar esta interfaz, como por ejemplo se pudo determinar que en este EEG los umbrales de Atención, Meditación y Blink_strength rodean los valores de 83, 74 y 200ms respectivamente en el primer sujeto.

Sin embargo, los mismos valores se ven muy diferentes en el segundo sujeto debido a su condición cardíaca que causa artefactos de alto potencial y provoca inestabilidad en la variable Blink_strenght (ver Anexo 6.4), como resultado de su exploración EEG se obtuvo sus umbrales de atención en 95, meditación en 82, y fuerza de parpadeo en 246ms . Cabe mencionar que para hacer de esta información una estadística confiable el parámetro Poor Quality Signal debió ser siempre "0".

CAPITULO 7

MODULO DE CONTROL DE PERIFÉRICOS ELECTRONICOS

En este capítulo se describe el diseño del circuito basado en el módulo Arduino Uno en el cual se hicieron las pruebas en la investigación en curso y del cual se obtuvieron resultados muy favorables en cuanto a tiempo de diseño, costo y técnica.

7.1. Diseño del circuito electrónico

El diseño electrónico se ha realizado en forma modular por la facilidad de implementación y compatibilidad que este posee y el objetivo de este proyecto el cual es servir de herramienta de diseño, por lo cual, el diseño de la PCB también tendrá una apariencia de Shield de Arduino con la única diferencia que será el módulo que lleva en si el Atmega 328 con el firmware diseñado.

7.1.1. MCPE (Módulo de control de periféricos electrónicos)

Este módulo posera un microcontrolador Atmega328 con bootloader Arduino Uno R3, consta de 2 reguladores para 5V y 3.3V, apariencia de Shield, leds indicadores, botón de reset y un Plug de entrada de voltaje.

Diagrama electrónico

A continuación se presenta el diseño electrónico creado basándose en la plataforma Arduino y siguiendo las estrictas normas de su licencia GNU Creative Commons las cuales señalan que el diseño puede ser replicado únicamente para comprensión y fabricación de prototipos mas no en forma industrial o con fines de lucro sin previa autorización de la empresa desarrolladora de Arduino. (Arduino, 2014).

Por tanto, el diseño presente en la Figura 7.1, es de autoría propia ya que únicamente utiliza el concepto “diseño modular” para ser compatible con los Shield que circulan en el mercado diseñados para la plataforma Arduino, y el Atmega 328-PU del cual se toma sus referencias de condicionamiento electrónico desde su hoja de datos, pero con el bootloader Optiboot previamente cargado para poder usar el microcontrolador en un módulo Arduino Uno.

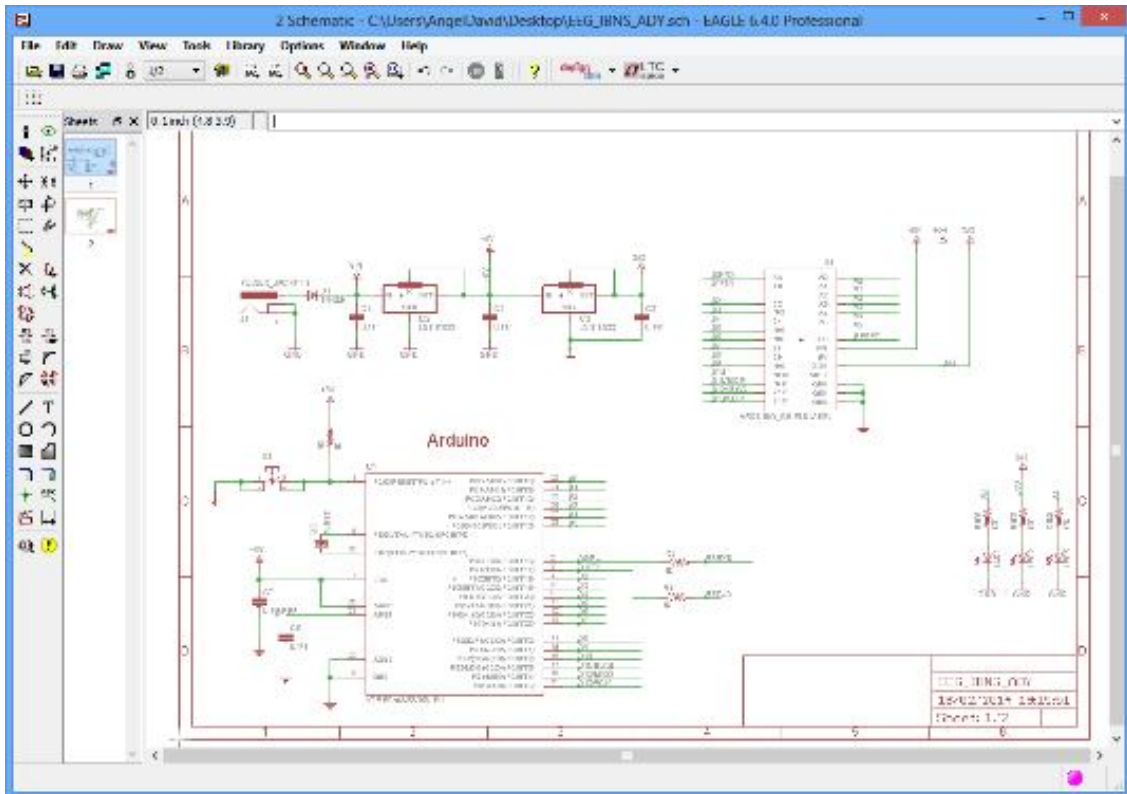


Figura 7.1 Diagrama electrónico del circuito 1.

Fuente: Print Screen de EAGLE 6.4.0 Professional

En el mismo diagrama electrónico existe una segunda hoja de diseño, esta parte comprende el conector serial DB9 para comunicarse por medio de señales CMOS a un PC u otro dispositivo. Al mismo tiempo este circuito permite cargar el programa compilado mediante el protocolo STK500 del bootloader Arduino en caso de quererse modificarlo o simplemente recargarlo en otro microcontrolador Atmega328.

Este circuito no será necesario de ser el caso en que se utilice el módulo de control de periféricos en una aplicación específicas sino únicamente solo si se desea mantener una comunicación UART con el Atmega328 o para actualizar su Firmware.

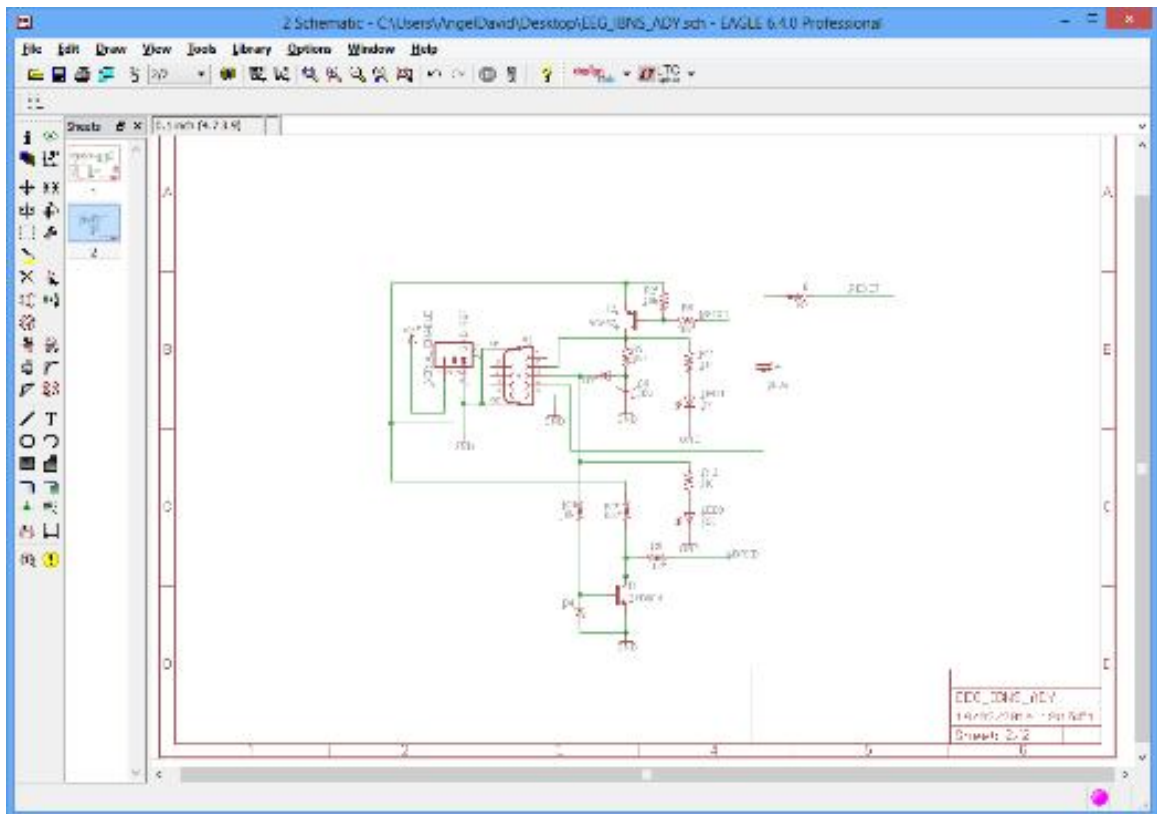


Figura 7.2 Diagrama electrónico del circuito 2.

Fuente: Print Screen de EAGLE CadSoft 6.4.0 Professional

Ambos archivos están disponibles en el CD adjunto al proyecto junto con toda la información pertinente a cada integrado y sus hojas de datos técnicos.

Adjunto se encuentra el archivo EEG_IBNS_ADY.txt el cual compendia una lista completa de todos los elementos empleados en la fabricación y ensamblado de este circuito. Este archivo fue generado mediante un aplicativo de Eagle llamado BOM.ulp el cual crea una lista automatizada de materiales que han sido usados y que son directamente vinculados desde las respectivas librerías de elementos de Eagle.

Tabla 7.1 Listado de materiales:

Part	Value	Device	Package	Description
Blink		Led3mm	Led3mm	Led
C1	0.1uf	Cap1206	1206	Capacitor
C2	0.1uf	Cap1206	1206	Capacitor
C3	0.1uf	Cap1206	1206	Capacitor
C4	100n	Cap1206	1206	Capacitor
C6	0.1uf	Cap1206	1206	Capacitor
C7	0.1uf	Cap1206	1206	Capacitor
C9	10u	Cap_Pol1206	Eia3216	Capacitor Polarized
D1	1n4004	1n4004	Do41-10	Diode
D2	1n4148	Diodesma	Smadiode	Diode
D4	1n4148	Diodesma	Smadiode	Diode
Ic1	328-Pu	Atmega 328-Pu	Dil28-3	
Ic2	Ld1117	Ld1117s33	Sot223	Voltage Regulator Smd 3,3v
Ic3	Ld1117	Ld1117s33	Sot223	Voltage Regulator Smd 3,3v
J1	Jack_pth	Power_Jackpth	Power_Jack	Power Jack
Jp1	Select	M03pth	1x03	Header 3
On	Led	Led3mm	Led3mm	Led
Psq	Led	Led3mm	Led3mm	Led
Q3	16mhz	Crystalhc49us	Hc49us	Crystals
R1	10k	R-U _s _R1206	R1206	Resistor, American Symbol
R2	10k	R-U _s _R1206	R1206	Resistor, American Symbol
R3	1k	Resistor1206	1206	Resistor
R4	1k	Resistor1206	1206	Resistor
R5	10k	R-U _s _R1206	R1206	Resistor, American Symbol
R6	1k	R-U _s _R1206	R1206	Resistor, American Symbol
R7	0	Resistor1206	1206	Resistor
R8	1k	Resistor1206	1206	Resistor
R9	4k7	R-U _s _R1206	R1206	Resistor, American Symbol
R10	10k	R-U _s _R1206	R1206	Resistor, American Symbol
R11	1k	R-U _s _R1206	R1206	Resistor, American Symbol
R12	1k	R-U _s _R1206	R1206	Resistor, American Symbol
R103	470	R-U _s _R1206	R1206@1	Resistor, American Symbol
R106	470	R-U _s _R1206	R1206@1	Resistor, American Symbol
R109	470	R-U _s _R1206	R1206@1	Resistor, American Symbol
Reset	Buttom	Tac_Switchpth	Tactile-Pth	Momentary Switch
Rx	Led0	Led3mm	Led3mm	Led
Com	Serial	F09hp	F09hp	Conector Db9 Hembra Pcb
T1	2n3904	Transistor_Npn	Sot23-3	Transistor Npn
T2	Bc557	Mmbt 3906	Sot23-Bec	Pnp Transistor
Tx	Led4	Led3mm	Led3mm	Led
U1	Arduino_Shield	Arduino_Shield	Shield	

Fuente: Bill of material, ULP generado por Eagle Cad Soft.

Diseño de la PCB

Una vez obtenido el diseño del circuito se procede a diseñar el PCB en el mismo software usando la herramienta "Generate to Board".

Para el diseño y ruteado del PCB a realizarse se ha tomado en cuenta las normas estándares de diseño IPC-A-610D SP la cual especifica información sobre montajes y IPC-2221A la cual es el estándar genérico de diseño de tarjetas de circuitos impresos, así mismo se ha tomado en cuenta los siguientes requerimientos técnicos en base al diseño modular del circuito original de Arduino:

- Orificios compatibles con el módulo Arduino Uno para ensamble modular vertical.
- Dimensiones físicas con un máximo de espacio de 10cm x 7cm para poder usar un diseño modular con los shields de Arduino Uno.
- Diseño del PCB con una sola capa.
- No utilización de puentes, diseño estético del circuito.

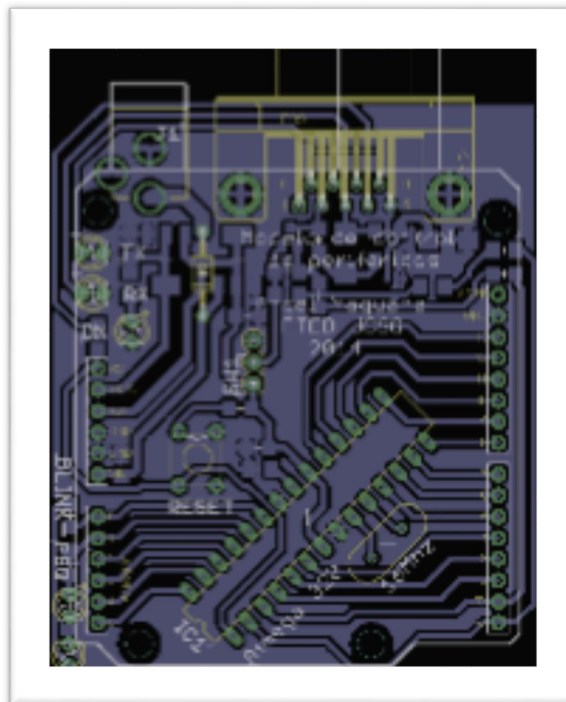


Figura 7.3 Diseño de la PCB del módulo de control de periféricos.

Fuente: Print Screen de EAGLE 6.4.0 Professional

Fabricación de los PCBs

El prototipado y fabricación de los PCBs se realiza utilizando materiales comunes de laboratorio de prototipado para poder hacer de pruebas de campo de su funcionalidad, previo al envío de la solicitud de fabricación profesional de las mismas a la empresa de fabricación de circuitos impresos APM Litoral en la ciudad de Guayaquil. (Ver Anexo 7.1)

7.1.2. Resultado

En la siguiente figura se puede apreciar sobre la mesa de trabajo el MCPE ya armado y con el Firmware cargado en el Atmega 328, además, se puede observar que el indicador PSQ está encendido ya que el sensor TGAM1 está procesando las señales cerebrales en el punto FP1 y enviándolas mediante Bluetooth hacia el dispositivo RN-41



Figura 7.4 IBNS y MCPE enlazados al sujeto de prueba, el autor, Angel Yaguana.

Fuente: Fotografía digital del autor, Angel Yaguana

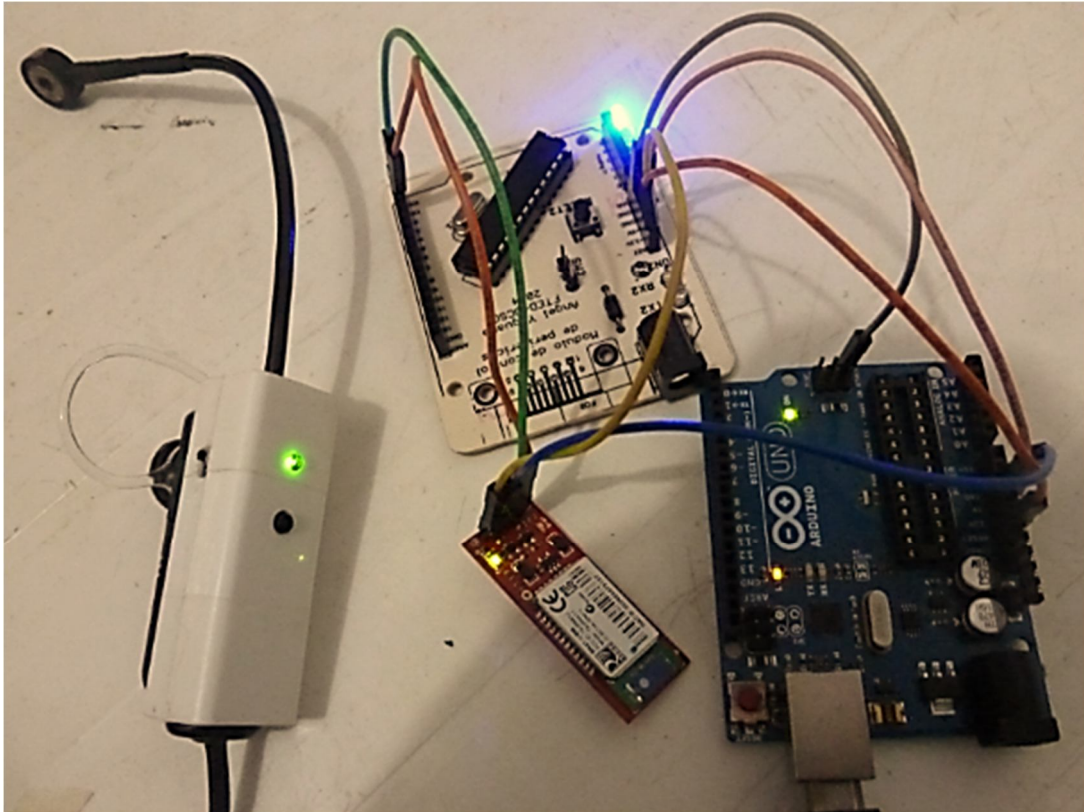


Figura 7.5 Modulo Arduino cargado con el firmware desarrollado.

Fuente: Fotografía digital del MCPE y la IBNS, Angel Yaguana

7.2. Pruebas de campo del prototipo desarrollado

Teniendo en cuenta el tipo de investigación y el método utilizado en el cual la prueba de campo a realizarse contempla un estudio de caso sobre el análisis de la solución planteada al cliente, se tiene en cuenta que está “prohibido revelar información personal confidencial identificable concerniente a los individuos participantes en la investigación” (Estándar 4.07 del Código de Ética de la APA, Utilización de Información Confidencial con Propósitos Didácticos o de Otra Naturaleza). (American Psychological Association, 2010), por lo cual, los resultados de la prueba en cuestión, se adjuntan sin categorización ni identificación de individuos, sin embargo, se aclara su condición física y mental durante el estudio de su caso en particular. Esta información se presenta en los anexos 13 y 14, esto es, estudio de exploración EEG y resultados de la estimulación Hombre Vs. Hardware.

7.2.1. Procedimiento

Para el siguiente estudio de caso, se dispone del dispositivo IBNS conectado al módulo de control de periféricos, y de un ambiente subjetivo pero real. Se le pide al sujeto de estudio que “simplemente trate de encender el LED azul”. Cada una de las pruebas tiene un tiempo de 3 minutos de duración y durante la cual los sujetos harán el ejercicio solicitado mientras un PC recoge en un archivo tabulado el examen EEG en el punto FP1 para corroborar su actividad cerebral en relación con lo sucedido físicamente.

7.2.2. Resultados

En el Anexo 7.2 se presentan los resultados correspondientes a la exploración EEG completa del sujeto de estudio y en el Anexo 7.3 se presenta los resultados del estudio del sujeto como cliente de frente al planteamiento en cuanto a necesidades generadas por las investigaciones INIAC e INIAPCE y generación de soluciones, respectivamente.

Adicionalmente, en el Anexo 15, se detalla un manual de uso y configuración del módulo de control de periféricos diseñado para su utilización en posteriores investigaciones o en el desarrollo de nuevas aplicaciones.

CONCLUSIONES

1. Producto del levantamiento de la situación actual de los procesos tecnológicos de exploración EEG, realizado mediante la investigación del estado de arte de esta tecnología, se ha logrado un nivel de conocimiento que evidencia que la clave para el desarrollo de aplicaciones en neurotecnología es el uso adecuado de la BCI de NeuroSky y la disponibilidad de herramientas que puedan facilitar su uso de forma generalizada y eficaz, aun cuando en la actualidad su alcance sea muy limitado.
2. Las pruebas de campo realizadas a las herramientas de desarrollo proporcionadas por NeuroSky, revelaron una gran dificultad en cuanto al análisis de los paquetes de datos enviados por el módulo TGAM1, causando un dificultoso reconocimiento de los valores de las variables del EEG sin el uso de una herramienta adecuada para gestionarlos y decodificarlos, además, se requirió un conocimiento muy técnico de sus características funcionales para poder identificarlos eficazmente.
3. En base a la evidencia resultante de la GUI desarrollada, se observó que efectivamente los datos que son enviados usando el protocolo del módulo TGAM1 son mucho más sencillos de visualizar si están en un lenguaje comprensible y entendible, siendo así prácticamente vistos en tiempo real sin mayores esfuerzos, de igual manera, también se evidencio que las API pueden ser usadas para capturar los paquetes desde un uC a través de un enlace Bluetooth, lo cual posibilita la creación de una aplicación embebida independiente de un PC.

4. La fabricación del firmware implementado usando los conceptos adquiridos del análisis de las API, incorporo el uso de un complejo algoritmo de captura y decodificación de las tramas del TGAM1, de las cuales se dispuso a entregar como resultado solo tres variables, dos de eSense y una de presencia de parpadeo, facilitando al receptor su manipulación desde un terminal serial de manera mucho más sencilla y comprensible. En este aspecto, el aporte de este algoritmo como puente entre la BCI y el receptor final, hace que sea más amigable la comunicación con el módulo TGAM1 durante una exploración EEG, otorgando una interfaz de interacción intuitiva para poder generar proyectos de implementación en los campos tecnológicos aplicativos que lo requieran sin mayores esfuerzos.

5. Previo la implementación de una aplicación neurotecnología usando esta interfaz como herramienta de desarrollo, es necesario comprender con claridad el comportamiento de las tres variables principales eSense, ePower y Blink-Eyes para poder con éxito manipular su comportamiento y predecirlo mediante un algoritmo que permita interactuar con el usuario de la manera más efectiva y fácil, para lo cual, es importante tener en cuenta que de todas éstas, la variable Blink es la única que nos permite cambiar de estados sin la necesidad de implementar una red neuronal para decodificar lo que el sujeto está tratando de hacer y así fijar secuencialmente mediante registros de desplazamiento los comportamientos que se quiera obtener.

6. El diseño modular del hardware permiten fácilmente crear aplicaciones compatibles con la plataforma Arduino y Processing que pueden ser orientadas a mejorar la condición de vida de las personas con capacidades físicas especiales como lo planteo el proyecto INIAPCE, además, esto influye directamente en la contribución a la formación de un entorno social donde todas las personas puedan tener las mismas oportunidades sin caer en estereotipos de discriminación por sus capacidades físicas.

7. Los cinco estudios de caso enfocados a receptar la opinión del cliente sobre la calidad y acceso al dispositivo diseñado, dejan un precedente a disposición del lector que evidencian la importancia de la calibración de los umbrales de sensibilidad en éstas variables previo el uso del dispositivo en sujetos de prueba distintos.

RECOMENDACIONES

Es ineludible contemplar la posibilidad de desarrollar un sistema que permita mediante el módulo de control de periféricos, la interacción con dispositivos que sirvan de ayuda a personas con capacidades limitadas, promoviendo una cultura de desarrollo para mejorar la calidad de vida de este grupo social, permitiéndoles además, participar de actividades que antes solo podían soñar.

Es por esto que es imperativo crear una aplicación orientada a la robótica educativa que se convierta en una base de I+D+i, para que mediante su contribución del algoritmo de manipulación y los grados de libertad que ésta disponga se pueda generar prototipos biomecánicos basados en su funcionalidad y a partir de estos construir equipos de control operables de alto rendimiento.

GLOSARIO

&: Símbolo de unión. Alternativa gráfica a la conjunción copulativa Y. Utilizado ad hoc si hay necesidad de citar o referenciar más de un autor.

A: El amperio o ampere (símbolo A), es la unidad de intensidad de corriente eléctrica.

Address: (dirección) En Internet dicese de la serie de caracteres, numéricos o alfanuméricos, que identifican un determinado recurso de forma única y permiten acceder a él. En la red existen varios tipos de dirección de uso común: "dirección de correo electrónico" (email address); "IP" (dirección internet); y "dirección hardware" o "dirección MAC" (hardware or MAC address).

Ag: Símbolo químico de la plata.

AgCl: El cloruro de plata es un compuesto químico bien conocido por su baja solubilidad en agua. Al ser sometido a luz o calor, el cloruro de plata reacciona y produce plata y cloro gaseoso. El cloruro de plata se encuentra en el mineral clorargirita.

Algoritmo: Procedimiento por medio del cual se resuelve cierta clase de problemas. Es la representación gráfica de una sucesión lógica de operaciones o pasos que conducen a la solución de un problema o a la producción de un bien o a la prestación de un servicio.

ANSI C: Es un estándar publicado por el Instituto Nacional Estadounidense de Estándares (ANSI), para el lenguaje de programación C. Se recomienda a los desarrolladores de software en C que cumplan con los requisitos descritos en el documento para facilitar así la portabilidad del código.

API: Application Program Interface, es un compendio de librerías que permiten realizar una aplicación.

Archivo de Texto: Archivo que utiliza solamente caracteres del estándar ASCII y por lo tanto que puede ser enviado por correo electrónico sin ningún tipo de modificación. Vea Binario, UUENCODE, UUDECODE.

Archivo electrónico: En computación se refiere a cualquier documento electrónico que contiene información que puede ser leída, vista u oída o que pueda tratarse de un programa informático.

Arduinoscope: librería de PSD con enlace a Arduino, disponible en Library Manager desde Processing.

ARM:Arquitectura Advanced Risch Machine, es una familia de microprocesadores producidos por la empresa ARM Holdings.

Artefacto EEG: Son señales en los EEG de origen no-cerebral.

ASCII: (American Standard Code for Information Interchange). Es de facto el estándar del World Wide Web para el código utilizado por computadoras para representar todas las letras (mayúsculas, minúsculas, letras latinas, números, signos de puntuación, etc.). El código estándar ASCII es de 128 letras representadas por un dígito binario de 7 posiciones (7 bits), de 0000000 a 1111111.

Ataque químico: Proceso químico por el cual un ácido actúa sobre otro elemento químico o un objeto material se diferente densidad y propiedad.

Atmega: Son microcontroladores AVR grandes con 4 a 256 kB de memoria flash programable, encapsulados de 28 a 100 pines, conjunto de instrucciones extendido (multiplicación y direccionamiento de programas mayores) y amplio conjunto de periféricos.

Atmel: Es una compañía de semiconductores, fundada en 1984. Su línea de productos incluye microcontroladores (incluyendo derivados del 8051, el AT91SAM basados en ARM, y sus arquitecturas propias AVR y AVR32), dispositivos de radiofrecuencia, memorias EEPROM y Flash, ASICs, WiMAX, y muchas otras. También tiene capacidad de ofrecer soluciones del tipo system on chip (SoC).

ATxmega: Son procesadores muy potentes con 16 a 384 kB de memoria flash programable, encapsulados de 44, 64 y 100 pines (A4, A3, A1), capacidad de DMA, eventos, criptografía y amplio conjunto de periféricos con DACs.

Authentication: (autenticación) Verificación de la identidad de una persona o de un proceso para acceder a un recurso o poder realizar determinada actividad. También se aplica a la verificación de identidad de origen de un mensaje encriptado en una trama.

AVR: Son una familia de microcontroladores RISC del fabricante estadounidense Atmel.

Baudios: (baud) es una unidad de medida, usada en telecomunicaciones, que representa el número de símbolos por segundo en un medio de transmisión digital. 1 Cada símbolo puede codificar 1 o más bits dependiendo del esquema demodulación.

BCI: Sigla en inglés de Brain Computes Interface, es una interface que enlaza el cerebro con la PC.

BMI: Sigla en inglés de Brain Machine Interface, interfaz que enlaza el cerebro con una máquina. Ej.: Robot

Big-Endian: El término inglés endianness ("extremidad") designa el formato en el que se almacenan los datos de más de un byte en un ordenador.

BinHex: (BINary HEXadecimal) Método para convertir archivos no ASCII o binarios al formato de siete bits ASCII. Este método es utilizado principalmente por computadora Macintosh. Esto es necesario porque el correo en Internet solo puede utilizar el ASCII. En 7 bits

Biométrica: (del griego bios vida y metron medida) es el estudio de métodos automáticos para el reconocimiento único de humanos basados en uno o más rasgos conductuales o rasgos físicos intrínsecos.

Bit: (Binary Digit). Unidad mínima de almacenamiento de la información. Su valor puede ser 0 o 1 o verdadero o falso.

Blindaje: El término blindaje se refiere a barreras físicas de protección

Bluetooth: es una especificación industrial para Redes Inalámbricas de Área Personal (WPAN) que posibilita la transmisión de voz y datos entre diferentes dispositivos mediante un enlace por radiofrecuencia en la banda ISM de los 2,4 GHz.

Bootloader: Un gestor de arranque (bootloader) es un programa sencillo que no tiene la totalidad de las funcionalidades de un sistema operativo, y que está diseñado exclusivamente para preparar todo lo que necesita el sistema operativo para funcionar.

Byte: Conjunto de 8 bits. Suele representar un valor asignado a un carácter.

C#: Es un lenguaje de programación orientado a objetos desarrollado y estandarizado por Microsoft como parte de su plataforma .NET, que después fue aprobado como un estándar por la ECMA (ECMA-334) e ISO (ISO/IEC 23270).

C/C++: Es un lenguaje de programación diseñado a mediados de los años 1980 por Bjarne Stroustrup. La intención de su creación fue el extender al exitoso lenguaje de programación C con mecanismos que permitan la manipulación de objetos. En ese sentido, desde el punto de vista de los lenguajes orientados a objetos, el C++ es un lenguaje híbrido.

Callback: Administrador de datos, se encarga de actuar sobre los valores de datos, ya que son recibidos y analizados.

CAD: Un conversor, (o convertidor) de señal analógica a digital, (o también CAD de "Conversor Analógico Digital", o ADC del inglés "Analog-to-Digital Converter") es un dispositivo electrónico capaz de convertir una señal analógica de voltaje en una señal digital con un valor binario. Se utiliza en equipos electrónicos como computadora, grabadores de sonido y de vídeo, y equipos de telecomunicaciones. La señal analógica, que varía de forma continua en el tiempo, se conecta a la entrada del dispositivo y se somete a un muestreo a una velocidad fija, obteniéndose así una señal digital a la salida del mismo.

Campo de aplicación: Se define en función del tema de estudio de investigación, u objeto de la asistencia técnica y/o capacitación, o elemento muestreado en cuestión, en relación a su aplicación en una actividad socioeconómica determinada.

CMOS: (Complementary metal-oxide-semiconductor) Es una de las familias lógicas empleadas en la fabricación de circuitos integrados. Su principal característica consiste en la utilización conjunta de transistores de tipo pMOS y tipo nMOS configurados de tal forma que, en estado de reposo, el consumo de energía es únicamente el debido a las corrientes parásitas, colocado obviamente en la placa base.

COM: Puerto serie de algunos ordenadores que usa el protocolo de comunicación RS-232.

ControlP5: librería grafica de Processing

Córtex: En neurociencias, es el manto de tejido nervioso que cubre la superficie de los hemisferios cerebrales;

Cortex-M3: Es un tipo de procesador ARM miembro de la familia de los Cortex™-M, estos son procesadores de 32 bits líderes en la industria para aplicaciones en tiempo real altamente deterministas, desarrollados específicamente para permitir desarrollar plataformas de bajo costo, de alto rendimiento y para una amplia gama de dispositivos, incluyendo los microcontroladores, sistemas de carrocería de automóvil, sistemas de control industrial, redes inalámbricas y sensores.

Corteza cerebral: Es el manto de tejido nervioso que cubre la superficie de los hemisferios cerebrales.

CHKSUM: Es una suma de verificación, en telecomunicación e informática, es una función hash que tiene como propósito principal detectar cambios accidentales en una secuencia de datos para proteger la integridad de estos, verificando que no haya discrepancias entre los valores obtenidos al hacer una comprobación inicial y otra final tras la transmisión.

Data Payload: Es la carga útil o real de una transmisión de datos.

DataRow: Es una clase de C++ que representa una fila de datos en un DataTable.

Desarrollo tecnológico: Desarrollo de nuevos productos o nuevos procesos. La producción de conocimientos potencialmente aplicables a una solución tecnológica cuyo desarrollo alcanza una escala de laboratorio, o equivalente. La construcción de prototipos y ensayos a escala piloto.

Diagrama electrónico: También conocido como un esquema eléctrico o esquemático es una representación pictórica de un circuito eléctrico.

DLL: Es una biblioteca de enlace dinámico sigla en inglés de dynamic-link library.

EAGLE CadSoft: (Easily Applicable Graphical Layout Editor) Es un programa de diseño de diagramas y PCBs con auto enrutador. Famoso alrededor del mundo de los proyectos electrónicos DiY, debido a que muchas versiones de este programa tienen una licencia Freeware y gran cantidad de bibliotecas de componentes alrededor de la red.

ECG: El electrocardiograma es la representación gráfica de la actividad eléctrica del corazón, que se obtiene con un electrocardiógrafo en forma de cinta continua.

EEG: Electroencefalografía es una exploración neurofisiológica que se basa en el registro de la actividad bioeléctrica cerebral en condiciones basales de reposo, en vigilia o sueño, y durante diversas activaciones (habitualmente hiperpnea y estimulación luminosa intermitente) mediante un equipo de electroencefalografía (producto sanitario).

EEPROM: (Electrically Erasable Programmable Read-Only Memory) Es un tipo de memoria ROM que puede ser programada, borrada y reprogramada eléctricamente.

Electrical Geodesics GES-250 system: Es un sistema de localización de electrodos para EEG con una disposición de 128 canales alrededor de la cabeza de una persona.

Electrodos: Es un conductor eléctrico utilizado para hacer contacto con una parte no metálica de un circuito, por ejemplo un semiconductor, un electrolito, el vacío (en una válvula termoiónica), un gas (en una lámpara de neón), etc.

EMG: La electromiografía es una técnica para la evaluación y registro de la actividad eléctrica producida por los músculos esqueléticos.

EOG: El electro oculograma es un examen que consiste en colocar pequeños electrodos cerca de los músculos de los ojos para medir el movimiento de éstos.

eSense y ePower: Relación de medidas que hacen referencia a los estados mentales y los valores de las frecuencias del cerebro humano según la empresa NeuroSky.

Estado del arte: Se entiende como "estado del arte", "estado de la técnica" o "estado de la cuestión", son todos aquellos desarrollos de última tecnología en los cuales se basa la investigación tecnológica.

Firmware: Es un bloque de instrucciones de máquina para propósitos específicos, grabado en una memoria, normalmente de lectura/escritura (ROM, EEPROM, flash, etc.), que establece la lógica de más bajo nivel que controla los circuitos electrónicos de un dispositivo de cualquier tipo. Está fuertemente integrado con la electrónica del dispositivo siendo el software que tiene directa interacción con el hardware: es el encargado de controlarlo para ejecutar correctamente las instrucciones externas. En resumen, un firmware es un software que maneja físicamente al hardware.

Fotolito: Es un paso intermedio en el complejo proceso de impresión sobre película o soporte transparente.

Fritzing: Es un programa de automatización de diseño electrónico libre que busca ayudar a diseñadores y artistas para que puedan pasar de prototipos (usando, por ejemplo, placas de pruebas) a productos finales.

Gilmore: Ciudad estadounidense del Estado de Arkansas

GND: Hace referencia al punto donde potencial de voltaje es "0V".

GPIO: (General Purpose Input/Output, Entrada/Salida de Propósito General) es un pin genérico en un chip, cuyo comportamiento (incluyendo si es un pin de entrada o salida) se puede controlar (programar) por el usuario en tiempo de ejecución.

GUI: (Interfaz Gráfica de Usuario) Componente de una aplicación informática que visualiza el usuario y a través de la cual opera con ella. Está formada por ventanas, botones, menús e iconos, entre otros elementos: Ver también: "icon", "interface", "WUI".

Hardware: Se refiere a todas las partes tangibles de un sistema informático o electrónico; sus componentes son: eléctricos, electrónicos, electromecánicos y mecánicos.

Header: En informática, Header es la parte inicial de un programa.

HEX: Son las siglas de referencia al sistema Hexadecimal el cual es un sistema de numeración en base 16 usada en informática y programación.

Hz: El hercio, hertzio o hertz (símbolo Hz), es la unidad de frecuencia del Sistema Internacional de Unidades.

I+D: La I+D es tan sólo una de esas actividades y puede llevarse a cabo en diferentes fases del proceso innovador, no sólo como la fuente original de ideas creadoras sino también como una forma de resolver los problemas que puedan surgir en cualquier etapa hasta su culminación. Además de la I+D, a menudo se pueden distinguir en el proceso de innovación otros seis campos de actividades innovadoras (Stead, 1976; OCDE, 1992c):

IBNS: Interfaz Biométrica Neuro-Sensorial, término acuñado por Angel Yaguana, 2014. Es un dispositivo que usa la tecnología de NeuroSky para realizar un EEG en el punto FP1, cuenta con servicio de transición de datos vía puerto serial mediante Bluetooth 2.0 y Audio Stream. Fue diseñado para ser usado en la investigación INIAPCE y actualmente su propiedad intelectual está registrada en la ciudad de Guayaquil, Ecuador.

IDE Arduino: Es una plataforma de hardware libre, basada en una placa con un microcontrolador y un entorno de desarrollo, diseñada para facilitar el uso de la electrónica en proyectos multidisciplinarios

IDE Processing: Es un lenguaje de programación y entorno de desarrollo integrado de código abierto basado en Java, de fácil utilización, y que sirve como medio para la enseñanza y producción de proyectos multimedia e interactivos de diseño digital.

IDE Eclipse: Es un programa informático compuesto por un conjunto de herramientas de programación de código abierto multiplataforma para desarrollar lo que el proyecto llama "Aplicaciones de Cliente Enriquecido", opuesto a las aplicaciones "Cliente-liviano" basadas en navegadores.

IDE Java NetBeans: Es un entorno de desarrollo integrado libre, hecho principalmente para el lenguaje de programación Java.

IDE Python: Es un lenguaje de programación interpretado cuya filosofía hace hincapié en una sintaxis muy limpia y que favorezca un código legible. Se trata de un lenguaje de programación multiparadigma, ya que soporta orientación a objetos, programación imperativa y, en menor medida, programación funcional.

IDE: (Integrated Development Environment) Entorno de desarrollo integrado, es un programa informático compuesto por un conjunto de herramientas de programación.

INIAC: Investigación de Neuro-Tecnologías Innovadoras Aplicadas Comercialmente

INIAPCE: Investigación de Neuro tecnologías Innovadoras Aplicadas a Personas con Capacidades Especiales

Innovaciones tecnológicas: Abarcan los nuevos productos y procesos, así como las modificaciones tecnológicas importantes en productos y procesos. Una innovación se ha

realizado en el momento en que se ha introducido en el mercado (innovación de producto) o se ha utilizado en un proceso de fabricación (innovación de proceso). Las innovaciones hacen intervenir todo tipo de actividades científicas, tecnológicas, de organización, financieras y comerciales.

Interfaz: En informática se utiliza para nombrar a la conexión física y funcional entre dos sistemas o dispositivos de cualquier tipo dando una comunicación entre distintos niveles.

Investigación y desarrollo (I+D): La investigación y el desarrollo experimental (I+D) comprenden el trabajo creativo llevado a cabo de forma sistemática para incrementar el volumen de los conocimientos humanos, culturales y sociales y el uso de esos conocimientos para derivar nuevas aplicaciones. Comprende tres actividades: investigación básica, investigación aplicada y desarrollo experimental.

IPC-2221A: Norma de estándar genérico de diseño de tarjetas de circuitos impresos

IPC-A-610D SP: Norma de estándar genérico de diseño que trata acerca de montajes

Java: Es un lenguaje de programación de propósito general, concurrente, orientado a objetos y basado en clases que fue diseñado específicamente para tener tan pocas dependencias de implementación como fuera posible.

JNI: Java Native Interface (JNI) es un framework de programación que permite que un programa escrito en Java ejecutado en la máquina virtual java (JVM) pueda interactuar con programas escritos en otros lenguajes como C, C++ y ensamblador.

kB: Un kilobyte, es una unidad de almacenamiento de información cuyo símbolo es el kB (con la 'k' en minúsculas) y equivale a 10³ (mil) bytes.

KDT: (Kit Development Tools) Es un compendio de herramientas de desarrollo que ofrece un sistema en concreto o en conjunto con un lenguaje o un entorno de desarrollo específico llamado KDE.

Link: Es un indicador de texto o una imagen que sirve como enlace a otro documento.

Linux: Es un núcleo libre de sistema operativo basado en Unix

Lóbulo: Es una parte de la corteza cerebral que subdivide el cerebro según sus funciones.

Mac OS: (Machintosh Operating System) es el nombre del sistema operativo creado por Apple para su línea de computadoras Macintosh.

MC: Modulo de Control.

MCPE: Modulo de Control de Periféricos Electrónicos, término acuñado por Angel Yaguana, 2014.

MDT: (Mindset Development Tools) Es un compendio de herramientas para desarrollar aplicaciones usando el dispositivo Mindset de NeuroSky.

Microsoft Visual Studio: Es un IDE para sistemas operativos Windows. Soporta múltiples lenguajes de programación tales como C++, C#, Visual Basic .NET, F#, Java, Python, Ruby, PHP; al igual que entornos de desarrollo web como ASP.NET MVC, Django, et., a lo cual sumarle las nuevas capacidades online bajo Windows Azure en forma del editor Monaco.

Mindset: Es un dispositivo creado por la empresa NeuroSky para realizar un EEG desde el punto FP1 de la corteza craneal y que usa la tecnología de procesamiento del chip TGAM1.

MIT: (Massachusetts Institute of Technology) Instituto Tecnológico de Massachusetts, es una universidad privada localizada en Cambridge, Massachusetts (Estados Unidos). Además, es la escuela de ingeniería es reconocida como la mejor en Estados Unidos y en el mundo por U.S. News & World Report. El MIT ha sido catalogado mundialmente como el mejor en tecnología por Times Higher Education-QS World University Rankings.

Modelo: Conjunto de variables relacionadas entre sí e interactuantes, que en bloque dinámico conducen a obtener un resultado predeterminado o a solucionar un problema.

Módulo: Es una parte repetitiva, autónoma e intercambiable de un diseño modular

Neurociencias: Las neurociencias son un conjunto de disciplinas científicas que estudian la estructura y la función, el desarrollo de la bioquímica, la farmacología, y la patología del sistema nervioso y de cómo sus diferentes elementos interactúan, dando lugar a las bases biológicas de la conducta.

Neuroseñales: Son señales eléctricas emitidas por el cerebro que se han obtenido mediante un estudio EEG o EMG.

NeuroSky: Es una compañía presidida por Stanley Yang en San José (California), Estados Unidos. La tecnología que se desarrolla en sus instalaciones es un ejemplo de interfaz cerebro computadora. Esta empresa hace uso de la electroencefalografía al hacer uso de electrodos situados sobre la cabeza con el fin de captar impulsos eléctricos cerebrales que se generan con el pensamiento.

Neurotecnología: Es un conjunto de herramientas que sirven para analizar e influir sobre el sistema nervioso del ser humano, especialmente sobre el cerebro. Estas tecnologías incluyen simulaciones de modelos neurales, computadores biológicos, aparatos para

interconectar el cerebro con sistemas electrónicos y aparatos para medir y analizar la actividad cerebral.

Non-REM: Es un periodo comprendido en un EEG como "sin movimiento muscular en los ojos".

Ondas Cerebrales: Es la actividad eléctrica producida por el cerebro. Estas ondas pueden ser detectadas mediante el electroencefalógrafo y se clasifican en: ondas delta (1 a 3 Hz), ondas theta (3,5 a 7,5 Hz), ondas alpha (8 a 13 Hz), ondas beta (12 a 30 Hz), ondas gamma (25 a 100 Hz).

Open Hardware: Se llama hardware libre o electrónica libre a aquellos dispositivos de hardware cuyas especificaciones y diagramas esquemáticos son de acceso público, ya sea bajo algún tipo de pago o de forma gratuita.

Open Source: Código abierto es la expresión con la que se conoce al software distribuido y desarrollado libremente. Se focaliza más en los beneficios prácticos (acceso al código fuente) que en cuestiones éticas o de libertad que tanto se destacan en el software libre.

PAYLOAD: Carga útil en la computación (a veces referido como los datos reales o cuerpo) es la carga de una transmisión de datos. Es la parte de la transmisión de datos que es el propósito fundamental de la transmisión, a la exclusión de la información enviada con él (tales como cabeceras o metadatos, a veces referido como datos generales) únicamente para facilitar la entrega.

PBS: (Product Breakdown Structure) es una herramienta para analizar, documentar y comunicar los resultados de un proyecto, y forma parte de la planificación de productos de base técnica.

PCB: Es una placa de circuito impreso en la que se apoya mecánicamente y se conecta eléctricamente componentes electrónicos usando conductores pistas, almohadillas y otras características grabados desde hojas de cobre laminadas sobre un sustrato no conductor.

PDE: (Processing Development Environment), Entorno de desarrollo de Processing

Percloruro de Hierro: Es un compuesto químico utilizado a escala industrial perteneciente al grupo de los haluros metálicos, cuya fórmula es FeCl_3 . También se le denomina percloruro férrico.

Periféricos: En informática y electrónica, se denomina periféricos a los aparatos y/o dispositivos auxiliares e independientes conectados a la unidad principal del sistema implementado ya sea este una CPU o un microprocesador. El término periférico implica a menudo el concepto de "adicional pero no esencial"

Peróxido de Hidrogeno: (H₂O₂), también conocido como agua oxigenada, dioxigen o dioxidano, es un compuesto químico con características de un líquido altamente polar, fuertemente enlazado con el hidrógeno tal como el agua, que por lo general se presenta como un líquido ligeramente más viscoso que ésta. Es conocido por ser un poderoso oxidante.

PinOut: Es un término anglosajón que, en traducción libre, significa patillaje, o más correctamente asignación de patillaje. Es usado en electrónica para determinar la función de cada pin en un circuito integrado, o bien en un dispositivo electrónico discreto. En informática, para describir cómo un conector es cableado. Cada pata del conector tiene un propósito que se describe brevemente en el pinout.

Print Screen: Es una captura de pantalla de una imagen tomada por una computadora o un teléfono inteligente para capturar los elementos vistos en el monitor u otro dispositivo de salida visual. Generalmente es una imagen digital tomada por el sistema operativo o aplicaciones siendo ejecutadas en la computadora o teléfono, pero puede ser también una captura hecha por un dispositivo externo como una cámara o algún otro dispositivo interceptando la salida de video de la computadora.

Protección ESD: Es la protección a la descarga electrostática la cual es un fenómeno electrostático que hace que circule una corriente eléctrica repentina y momentáneamente entre dos objetos de distinto potencial eléctrico;

Prototipo: Se puede referir a cualquier tipo de máquina en pruebas, o un objeto diseñado para una demostración de cualquier tipo. Un prototipo en software es un modelo del comportamiento del sistema que puede ser usado para entenderlo completamente o ciertos aspectos de él y así clarificar los requerimientos... Un prototipo es una representación de un sistema, aunque no es un sistema completo, posee las características del sistema final o parte de ellas.

Prueba de concepto: Es una implementación, a menudo resumida o incompleta, de un método o de una idea, realizada con el propósito de verificar que el concepto o teoría en cuestión es susceptible de ser explotada de una manera útil. Prototipo de Prueba de Concepto es un término que sirve para denominar a un circuito construido siguiendo una guías similares a las de un prototipo convencional, pero en el que la intención no es construir una versión inicial o temprana de un diseño de producción, sino únicamente demostrar que un determinado circuito o una nueva técnica de producción es ciertamente posible.

PSI: La libra-fuerza por pulgada cuadrada, más conocida como psi (del inglés pounds-force per square inch) es una unidad de presión en el sistema anglosajón de unidades.

QFD: (Despliegue de función de calidad) es un método de gestión de calidad¹ basado en transformar las demandas del usuario en la calidad del diseño, implementar las funciones que aporten más calidad, e implementar métodos para lograr calidad del diseño en subsistemas y componentes, y en última instancia a los elementos específicos del proceso de fabricación.

RAM: (random-access memory) Es una memoria de acceso aleatorio se utiliza como memoria de trabajo para el sistema operativo, los programas y la mayoría del software.

Ref.: Es un punto de referencia usado para hacer un diferencial de potencial en contraposición con otro punto.

REM: (Rapid eye movement) o 'movimiento ocular rápido, es la fase del sueño durante la que suceden los sueños más intensos).

Rhythmic syncopation: Un ritmo sincopado es un ritmo inesperado que se desvía de los latidos normales regularmente espaciados en un pulso. El ritmo se considera sincopado cuando se acentúa la medida no acentuada.

RN-41: Rovin Networks #41, Modulo Bluetooth comercial.

ROM: (read-only memory), es un medio de almacenamiento utilizado en ordenadores y dispositivos electrónicos, que permite sólo la lectura de la información y no su escritura, independientemente de la presencia o no de una fuente de energía.

Running head: En APA, es el encabezado principal del documento realizado.

Rx: Pin de recepción de datos seriales en un puerto UART

Serigrafía: Es una técnica de impresión empleada en el método de reproducción de documentos e imágenes sobre cualquier material, y consiste en transferir una tinta a través de una malla tensada en un marco, el paso de la tinta se bloquea en las áreas donde no habrá imagen mediante una emulsión o barniz, quedando libre la zona donde pasará la tinta.

Shields Arduino: Las "Shields" son placas que pueden ser conectadas encima de la placa Arduino extendiendo sus capacidades. Las diferentes "shields" siguen la misma filosofía que el conjunto original: son fáciles de montar, y baratas de producir.

Sintaxis de programación: Es la forma en la que se describe las combinaciones posibles de los símbolos que forman un programa sintácticamente correcto. El significado que se le da a una combinación de símbolos es manejado por su semántica (ya sea formal o como parte

del código duro de la referencia de implementación). Dado que la mayoría de los lenguajes son textuales, este artículo trata de la sintaxis textual.

Sketch: Un sketch es el nombre que usa Arduino para un programa. Es la unidad de código que se sube y ejecuta en la placa Arduino.

Software: (componentes lógicos, programas, software) Programas o elementos lógicos que hacen funcionar un ordenador o una red, o que se ejecutan en ellos, en contraposición con los componentes físicos del ordenador o la red. Ver también: "hardware", "malware".

Sosa caustica: El hidróxido de sodio (NaOH) o hidróxido sódico, también conocido como soda cáustica o sosa cáustica, es un hidróxido cáustico usado en la industria (principalmente como una base química) en la fabricación de papel, tejidos, y detergentes.

SPD: Personas en situación de discapacidad

SRAM: Static Random Access Memory (SRAM), o Memoria Estática de Acceso Aleatorio es un tipo de memoria basada en semiconductores que a diferencia de la memoria DRAM, es capaz de mantener los datos, mientras esté alimentada, sin necesidad de circuito de refresco. Sin embargo, si son memorias volátiles, es decir que pierden la información si se les interrumpe la alimentación eléctrica.

Symbian: Es un sistema operativo propiedad de Nokia, y que en el pasado fue producto de la alianza de varias empresas de telefonía móvil, entre las que se encontraban Nokia, Sony Mobile Communications, Psion, Samsung, Siemens, Arima, Benq, Fujitsu, Lenovo, LG, Motorola, Mitsubishi Electric, Panasonic, Sharp, etc. Sus orígenes provenían de su antepasado EPOC32, utilizado en PDA's y Handhelds de PSION.

SYNC: Es un comando de sincronización que permite identificar el inicio de la transición de datos en una interfaz de comunicación.

TGAM1: (ThinkGear Advanced Machine v.1) Es un dispositivo diseñado por la empresa NeuroSky para diseñar aplicaciones orientadas a Neurociencia o Neurotecnología mediante un EEG.

TGC: (ThinkGear Connector) Es un software que usa un DLL llamado TGDC por el cual se enlazan los datos de un dispositivo que usa un TGAM1 con algún IDE para ser mostrados en una aplicación de usuario de tipo GUI.

TGCD: (ThinkGear Device Connector) Es un DLL diseñado por NeuroSky para administrar los datos transmitidos por un dispositivo TGAM1 y que trabaja conjuntamente con un API de un IDE para gestionar las variables del sistema de adquisición previa su publicación en una GUI.

ThinkGear: Idéntico a TGAM1, Es un dispositivo diseñado por la empresa NeuroSky para diseñar aplicaciones orientadas a Neurociencia o Neurotecnología mediante un EEG.

TTL: (transistor-transistor logic) Es una familia lógica o lo que es lo mismo, una tecnología de construcción de circuitos electrónicos digitales.

Tx: Pin de transmisión de datos seriales en un puerto UART

UART: (Universal Asynchronous Receiver-Transmitter) Es un protocolo de comunicación que controla los puertos y dispositivos serie. Se encuentra integrado en la placa base o en la tarjeta adaptadora de un dispositivo.

uC: Microcontrolador

UCSG: Universidad Católica de Santiago de Guayaquil

Unix: Sistema operativo especializado en capacidades de multiusuario y multitarea. Fue la base inicial de Internet. Entre sus características más importantes se encuentran: Redireccionamiento de Entrada/Salida Alta portabilidad al estar escrito en lenguaje C, lo que lo hace independiente del hardware Interface simple e interactivo con el usuario.

URL: Localizador Uniforme de recursos (Uniform Resorce Locator). Sistema de direccionamiento estándar para archivos y funciones de Internet, especialmente en el Word Wide Web.

V: El voltio, o volt, por símbolo V, es la unidad derivada del Sistema Internacional para el potencial eléctrico, la fuerza electromotriz y la tensión eléctrica.

Virtual: (virtual) Algo que tiene existencia aparente y no real. [DRAE]. Es un término de frecuente utilización en el mundo de las Tecnologías de la Información y de las Comunicaciones para designar dispositivos o funciones simulados. "virtual circuit", "Virtual Private Network".

Win32: Es la interfaz de programación de aplicaciones de Windows, cuyo nombre en inglés es Windows API y es un conjunto de funciones residentes en bibliotecas (generalmente dinámicas, también llamadas DLL por sus siglas en inglés, término usado para referirse a éstas en Windows) que permiten que una aplicación corra bajo un determinado sistema operativo.

Windows Mobile: es un sistema operativo móvil compacto desarrollado por Microsoft, y diseñado para su uso en teléfonos inteligentes (Smartphones) y otros dispositivos móviles.

Wiring: Es una plataforma de creación de prototipos de electrónica compuesto por un lenguaje de programación, un entorno de desarrollo integrado (IDE) y un microcontrolador de una sola tarjeta.

Yoguis: Un yogui es un practicante de yoga; esta designación se reserva generalmente para practicantes avanzados de yoga.

Zen: Es una escuela del budismo mahāyāna que apartándose del conocimiento teórico o intelectual, el zen busca la experiencia de la sabiduría más allá del discurso racional.

REFERENCIAS BIBLIOGRAFICAS

- Aliexpress. (2014). *Tienda Online de Asia*. Obtenido de <http://www.aliexpress.com/item/brainwave-sensor-brain-control-toys-mind-games-bio-sensor-NeuroSky-ThinkGear-am-TGAM-geek-BCI-bio/1361377769.html>
- Alvira, F. (2003). *El análisis de la realidad social. Métodos y técnicas de investigación*. Madrid: Alianza Universidad Textos.
- American Psychological Association. (2010). Manual de publicaciones, tercera edición. *Manual de publicaciones de la American Psychological Association, tercera edición, 6º*, 16. (M. G. Frías., Trad.) Mexico DF, Mexico: El Manual Moderno S. A. de C. V. doi:978-1-4338-0561-5
- Arduino. (2014). *Home page*. Recuperado el 16 de 01 de 2014, de <http://Arduino.cc/es/uploads/Main/>
- Arena, S. Z. (2008). *Metodología de la Investigación*. Recuperado el 24 de 10 de 2013, de <http://metodologiadelainvestigacion.forumcommunity.net/?t=14325119>.
- Berger, H. (1991). *La Elektrenkephalogramm del hombre*. Frankfurt am Main : Gerhard Muehlau pmi-Verlag.
- By Sean M. Montgomery, I. M. (13 de 2 de 2013). Track your body's signals and brain waves and use them to control things. *Make(26)*. Recuperado el 13 de 2 de 2014, de <http://hackaday.com/>: <http://archive.makezine.com/26/primer/>
- Corp., N. (2009). *NeuroSky Home Page*. Recuperado el 2 de 12 de 2013, de www.neurosky.com
- Costumers. (2008). *Mindball Home Page*. Recuperado el 1 de 12 de 2013, de <http://www.mindball.se/customers.html>
- Editorial-Cuenca. (2010). Tecnología. *Revista Lideres*,, 6.

- Fitzgibbon, S. P., Powers, D. M., Pope, K. J., & Clark, C. R. (2007). Removal of EEG noise and artifact using blind source separation. *Journal of Clinical Neurophysiology*, 3(24), 232-243. doi:10.1097/WNP.0b013e3180556926
- Gamboa, R. (2014). *METODOS DE LA INVENTIGACION*. Recuperado el 26 de Marzo de 2014, de <http://ricardogamboa.weebly.com/meacutetodos-de-la-investigacioacuten.html>
- García Vacacela, R. (2013). *Investigación de Neuro Tecnologías Innovadoras Aplicadas Comercialmente*. Universidad Católica de Santiago de Guayaquil, Sistema de Investigación y Desarrollo. Guayaquil: Sinda-UCSG.
- Gilmore, J. C. (1994). *Neurophysiol.* (J. Clin., Ed.) American Electroencephalographic Society guidelines in electroencephalography, evoked potentials, and polysomnography.
- Heinz Penin, Walter Fröscher. (06 de 06 de 2013). <http://www.izepilepsie.de/home/showdoc,id,392,aid,650.html>. (S. A. eV, Editor)
- Hutchins, D. (2008). Hoshin Kanri - The Strategic Approach to Continuous Improvement . En H. Kanri, *The Strategic Approach to Continuous Improvement* . Gower publishing.
- IFCN, C. I. (2014). *TECNICAS NEUROFISIOLOGICAS Y CASOS CLINICOS*. Obtenido de http://www.neuromonitoreo.8m.net/catalog_2.html
- Joyce, C. A., Gorodnitsky, I. F., & Kutas, M. (2004). Automatic removal of eye movement and blink artifacts from EEG data using blind component separation. *Psychophysiology*, 2(41), 313-25. doi:10.1111/j.1469-8986.2003.00141.x
- Jung, T.-P., Makeig, S., Humphries, C., Lee, T.-W., McKeown, M. J., Iragui, V., & Sejnowski, T. J. (2000). Removing electroencephalographic artifacts by blind source separation. *Psychophysiology*, 2(37), 163-78. doi:10.1016/S1388-2457(00)00386-2
- La Cofa. (9 de Diciembre de 2011). <http://www.lacofa.es/index.php/general/introduccion-a-los-sistemas-brain-computer-interface>. Recuperado el 15 de Noviembre de 2013, de <http://www.lacofa.es/>
- Lab, N. I. (2014). *ERP/EEG*. Obtenido de <http://www.ncilab.ca/erpeeg-2>

- Millán, J. (2011). *Curiosidad y trabajo en equipo*. (L. M. ARIZA, Ed.) Lausana, Suiza: El País.
- NeuroSky, Inc. (28 de Junio de 2010). ASIC_EEG_POWER. *MindSet Communications*, 8. Recuperado el 30 de Noviembre de 2013, de <http://neurosky.com/>
- Nuñez, P. (1981). *Electric Fields of the Brain: The Neurophysics of EEG*,. New York: Oxford University Press.
- PAULO A, H. S. (2010). TECNICAS DE CLASIFICACION PARA NEUROSEÑALES. *Scientia Et Technica*, XVII(46), 153-156. doi:0122-1701
- Shackman, A. J., McMenamin, B. W., Maxwell, J. S., Greischar, L. L., & Davidson, R. J. (2010). Identifying robust and sensitive frequency bands for interrogating neural oscillations. *NeuroImage*, 4(51), 13319-33. doi:10.1016/j.neuroimage.2010.03.037
- Steinkopff-Verlag, H. (2008). El manual de AASM para la puntuación de sueño y eventos asociados. Reglas, la tecnología y las especificaciones técnicas. *Academia Americana de Medicina del Sueño (eds.)*.
- Suihko V, M. J. (1993). Distribution of sensitivity of electric leads in an inhomogeneous spherical head model. Tampere Univ. Techn. Ragnar Granit Inst.
- Tamayo, M. T. (2000). El Proceso de la Investigación. Limusa Noriega Editores.
- Tatoon. (12 de 12 de 2007). *Sega crea juguetes controlados mentalmente*. Recuperado el 5 de 12 de 2013, de <http://news.segarcadia.com/2007/12/sega-crea-juguetes-controlados.html>
- Yasui, Y. (2009). A Brain signal measurement and data procesing tecnique for daily life applications. *Journal of Physiological and Anthropology*, 6.

BIBLIOGRAFÍA

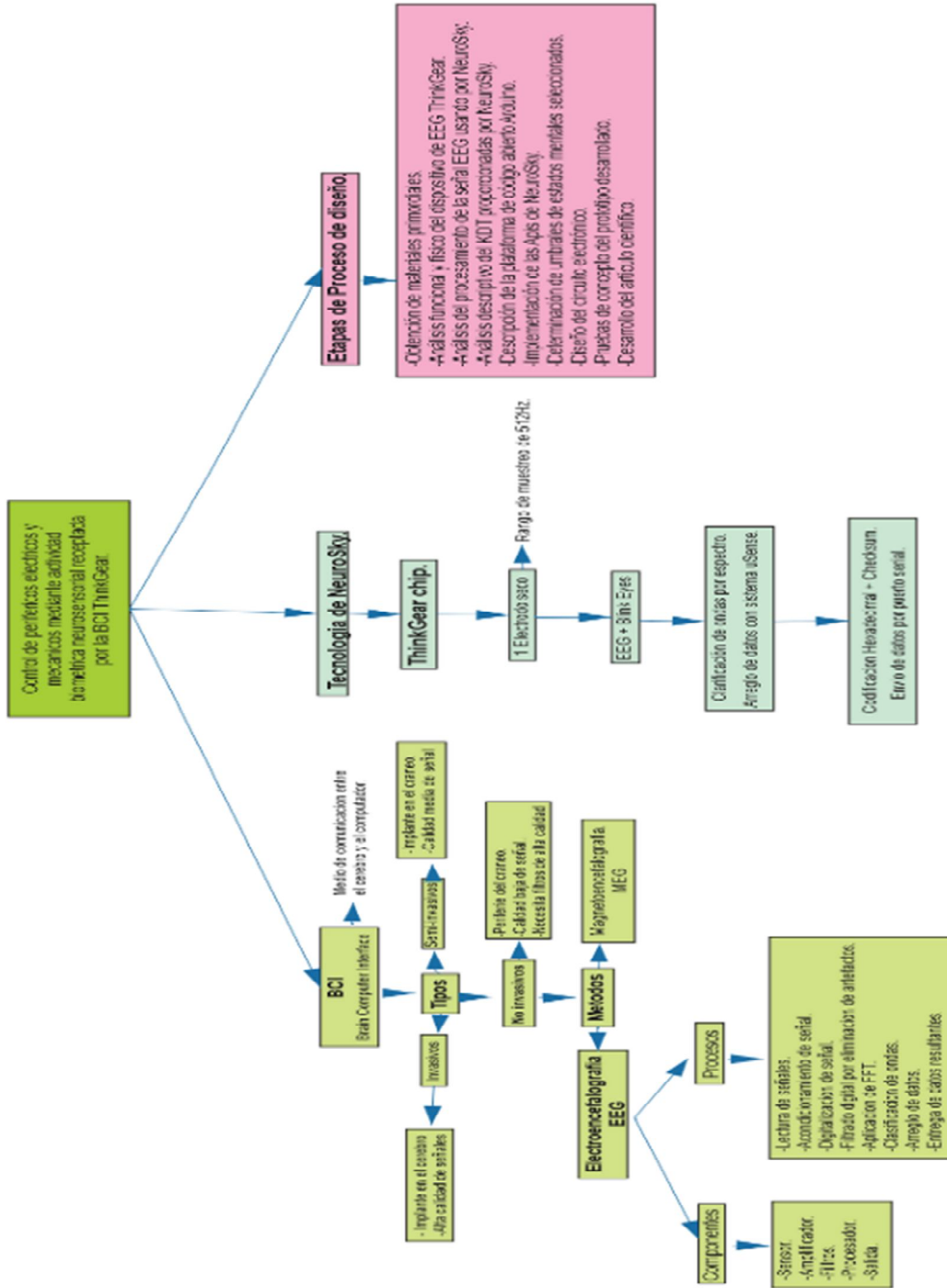
- Aliexpress. (2014). *Tienda Online de Asia*. Obtenido de <http://www.aliexpress.com/item/brainwave-sensor-brain-control-toys-mind-games-bio-sensor-NeuroSky-ThinkGear-am-TGAM-geek-BCI-bio/1361377769.html>
- Alvira, F. (2003). *El análisis de la realidad social. Métodos y técnicas de investigación*. Madrid: Alianza Universidad Textos.
- American Psychological Association. (2010). Manual de publicaciones, tercera edición. *Manual de publicaciones de la American Psychological Association, tercera edición, 6º*, 16. (M. G. Frías., Trad.) Mexico DF, Mexico: El Manual Moderno S. A. de C. V. doi:978-1-4338-0561-5
- Arduino. (2014). *Home page*. Recuperado el 16 de 01 de 2014, de <http://Arduino.cc/es/uploads/Main/>
- Arena, S. Z. (2008). *Metodología de la Investigación*. Recuperado el 24 de 10 de 2013, de <http://metodologiadelainvestigacion.forumcommunity.net/?t=14325119>.
- Berger, H. (1991). *La Elektrenkephalogramm del hombre*. Frankfurt am Main : Gerhard Muehlau pmi-Verlag.
- By Sean M. Montgomery, I. M. (13 de 2 de 2013). Track your body's signals and brain waves and use them to control things. *Make(26)*. Recuperado el 13 de 2 de 2014, de <http://hackaday.com/>: <http://archive.makezine.com/26/primer/>
- Corp., N. (2009). *NeuroSky Home Page*. Recuperado el 2 de 12 de 2013, de www.neurosky.com
- Costumers. (2008). *Mindball Home Page*. Recuperado el 1 de 12 de 2013, de <http://www.mindball.se/customers.html>
- Editorial-Cuenca. (2010). Tecnología. *Revista Lideres*,, 6.

- Fitzgibbon, S. P., Powers, D. M., Pope, K. J., & Clark, C. R. (2007). Removal of EEG noise and artifact using blind source separation. *Journal of Clinical Neurophysiology*, 3(24), 232-243. doi:10.1097/WNP.0b013e3180556926
- Gamboa, R. (2014). *METODOS DE LA INVENTIGACION*. Recuperado el 26 de Marzo de 2014, de <http://ricardogamboa.weebly.com/meacutetodos-de-la-investigacioacuten.html>
- García Vacacela, R. (2013). *Investigación de Neuro Tecnologías Innovadoras Aplicadas Comercialmente*. Universidad Católica de Santiago de Guayaquil, Sistema de Investigación y Desarrollo. Guayaquil: Sinde-UCSG.
- Gilmore, J. C. (1994). *Neurophysiol.* (J. Clin., Ed.) American Electroencephalographic Society guidelines in electroencephalography, evoked potentials, and polysomnography.
- Heinz Penin, Walter Fröscher. (06 de 06 de 2013). <http://www.izepilepsie.de/home/showdoc,id,392,aid,650.html>. (S. A. eV, Editor)
- Hutchins, D. (2008). Hoshin Kanri - The Strategic Approach to Continuous Improvement . En H. Kanri, *The Strategic Approach to Continuous Improvement* . Gower publishing.
- IFCN, C. I. (2014). *TECNICAS NEUROFISIOLOGICAS Y CASOS CLINICOS*. Obtenido de http://www.neuromonitoreo.8m.net/catalog_2.html
- Joyce, C. A., Gorodnitsky, I. F., & Kutas, M. (2004). Automatic removal of eye movement and blink artifacts from EEG data using blind component separation. *Psychophysiology*, 2(41), 313-25. doi:10.1111/j.1469-8986.2003.00141.x
- Jung, T.-P., Makeig, S., Humphries, C., Lee, T.-W., McKeown, M. J., Iragui, V., & Sejnowski, T. J. (2000). Removing electroencephalographic artifacts by blind source separation. *Psychophysiology*, 2(37), 163-78. doi:10.1016/S1388-2457(00)00386-2
- La Cofa. (9 de Diciembre de 2011). <http://www.lacofa.es/index.php/general/introduccion-a-los-sistemas-brain-computer-interface>. Recuperado el 15 de Noviembre de 2013, de <http://www.lacofa.es/>
- Lab, N. I. (2014). *ERP/EEG*. Obtenido de <http://www.ncilab.ca/erpeeg-2>

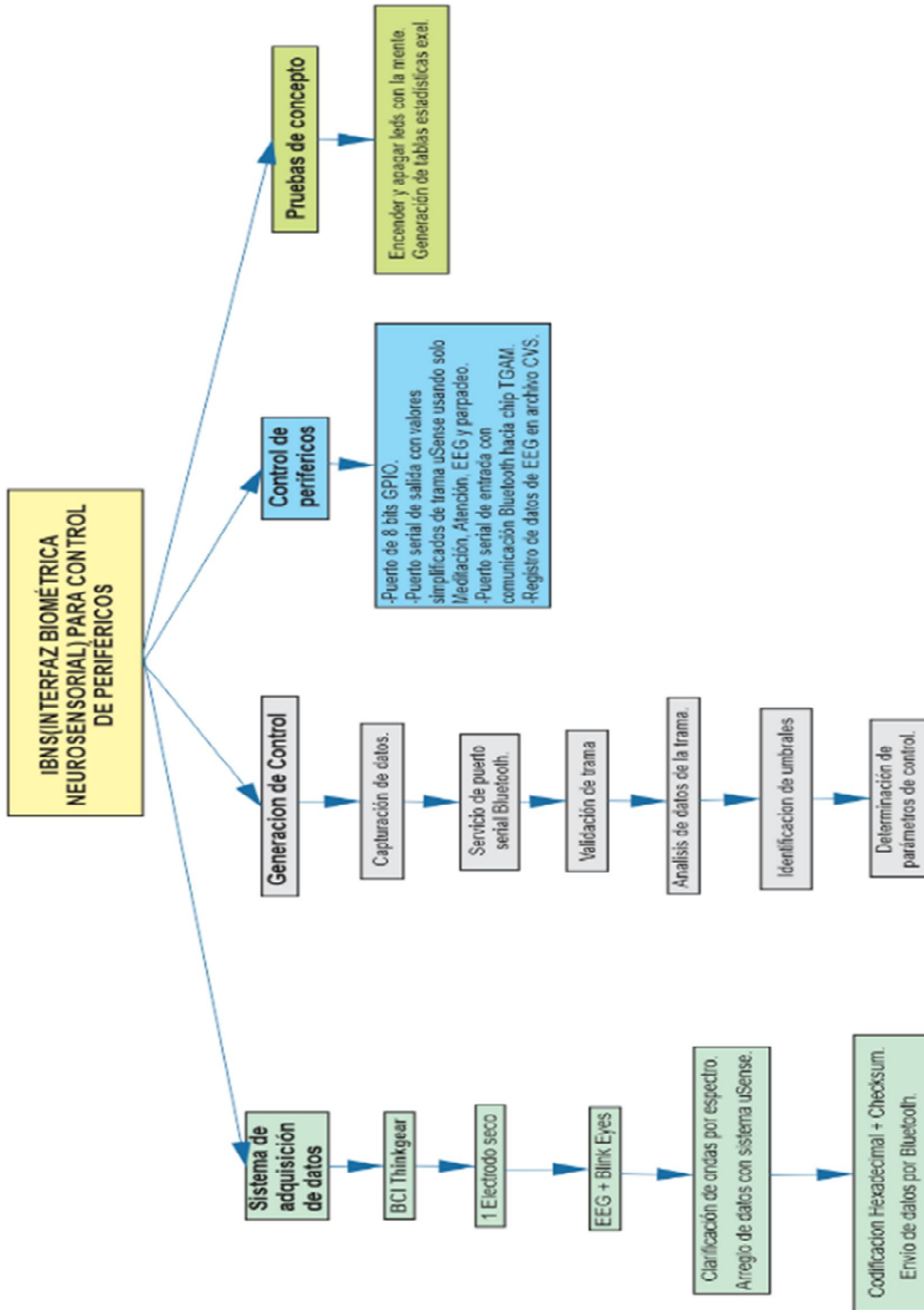
- Millán, J. (2011). *Curiosidad y trabajo en equipo*. (L. M. ARIZA, Ed.) Lausana, Suiza: El País.
- NeuroSky, Inc. (28 de Junio de 2010). ASIC_EEG_POWER. *MindSet Communications*, 8. Recuperado el 30 de Noviembre de 2013, de <http://neurosky.com/>
- Nuñez, P. (1981). *Electric Fields of the Brain: The Neurophysics of EEG*,. New York: Oxford University Press.
- PAULO A, H. S. (2010). TECNICAS DE CLASIFICACION PARA NEUROSEÑALES. *Scientia Et Technica*, XVII(46), 153-156. doi:0122-1701
- Shackman, A. J., McMenamin, B. W., Maxwell, J. S., Greischar, L. L., & Davidson, R. J. (2010). Identifying robust and sensitive frequency bands for interrogating neural oscillations. *NeuroImage*, 4(51), 13319-33. doi:10.1016/j.neuroimage.2010.03.037
- Steinkopff-Verlag, H. (2008). El manual de AASM para la puntuación de sueño y eventos asociados. Reglas, la tecnología y las especificaciones técnicas. *Academia Americana de Medicina del Sueño (eds.)*.
- Suihko V, M. J. (1993). Distribution of sensitivity of electric leads in an inhomogeneous spherical head model. Tampere Univ. Techn. Ragnar Granit Inst.
- Tamayo, M. T. (2000). El Proceso de la Investigación. Limusa Noriega Editores.
- Tatoon. (12 de 12 de 2007). *Sega crea juguetes controlados mentalmente*. Recuperado el 5 de 12 de 2013, de <http://news.segarcadia.com/2007/12/sega-crea-juguetes-controlados.html>
- Yasui, Y. (2009). A Brain signal measurement and data procesing tecnique for daily life applications. *Journal of Physiological and Anthropology*, 6.

ANEXOS

ANEXO 3.1



ANEXO 3.2



ANEXO 5.1

Ejemplo de captura de un paquete en lenguaje C.

```
#include <stdio.h>
#define SYNC 0xAA
#define EXCODE 0x55
int parsePayload( unsigned char *payload, unsigned char pLength ) {
    unsigned char bytesParsed = 0;
    unsigned char code;
    unsigned char length;
    unsigned char extendedCodeLevel;
    int i;

    while( bytesParsed < pLength ) {
        extendedCodeLevel = 0;
        while( payload[bytesParsed] == EXCODE ) {
            extendedCodeLevel++;
            bytesParsed++;
        }
        code = payload[bytesParsed++];
        if( code & 0x80 ) length = payload[bytesParsed++];
        else length = 1;
        printf( "EXCODE level: %d CODE: 0x%02X length: %d\n",
            extendedCodeLevel, code, length );
        printf( "Data value(s):" );
        for( i=0; i<length; i++ ) {
            printf( " %02X", payload[bytesParsed+i] & 0xFF );
            printf( "\n" );
            bytesParsed += length;
        }
        return( 0 );
    }

    int main( int argc, char **argv ) {
        int checksum;
        unsigned char payload[256];
        unsigned char pLength;
        unsigned char c;
        unsigned char i;
        FILE *stream = 0;
        stream = fopen( "COM4", "r" );
        while( 1 ) {
            fread( &c, 1, 1, stream );
            if( c != SYNC ) continue;
            fread( &c, 1, 1, stream );
            if( c != SYNC ) continue;
            while( true ) {
                fread( &pLength, 1, 1, stream );
                if( pLength ~ 170 ) break;
            }
            if( pLength > 169 ) continue;
            fread( payload, 1, pLength, stream );
            checksum = 0;
            for( i=0; i<pLength; i++ ) checksum += payload[i];
            checksum &= 0xFF;
            checksum = ~checksum & 0xFF;
            fread( &c, 1, 1, stream );
            if( c != checksum ) continue;
            parsePayload( payload, pLength );
        }
        return(0);
    }
}
```

ANEXO 5.2

ThinkGearStreamParser API en C.

```
#define PARSER_BATTERY_CODE 0x01
#define PARSER_POOR_SIGNAL_CODE 0x02
#define PARSER_ATTENTION_CODE 0x04
#define PARSER_MEDITATION_CODE 0x05
#define PARSER_RAW_CODE 0x80

THINKGEAR_initParser()
Int THINKGEAR_initParser( ThinkGearStreamParser *parser, unsigned char parserType,
void (*handleDataValueFunc)(
unsigned char extendedCodeLevel,
unsigned char code, unsigned char numBytes,
const unsigned char *value, void *customData),
void *customData );

THINKGEAR_parseByte()
Int THINKGEAR_parseByte( ThinkGearStreamParser *parser, unsigned char byte );
```

ANEXO 1.3

Creación del boceto IBNS_EEG_VISUALIZER.

```
//Librerías importadas
import Arduinoscope.*;
import NeuroSky.*;
import org.json.*;
import controlP5.*;

ControlP5 cp5;
ControlTimer c;
Textlabel t;
Oscilloscope scope;
ThinkGearSocket neuroSocket;

//Configuración del boceto
void setup() {
ThinkGearSocket neuroSocket = new ThinkGearSocket(this);
try {
neuroSocket.start();
}
catch (Exception e) {
println("Esta encendido su dispositivo");
}

size(x, y);
frameRate(30);
cp5 = new ControlP5(this);
c = new ControlTimer();
t = new Textlabel(cp5, "--", 520, 377);
c.setSpeedOfTime(1);
smooth();
tint(255, 126);
int[] dimensions = {x, 200};
int[] position = {0, 150};
scope = new Oscilloscope(this, position, dimensions);
scope.setLine_color(color(255, 0, 0));
scope.setBounds_color(color(0, 255, 0));
}

//Generacion de la GUI mediante la funcion draw
void draw(){
background(0); //RGB, Claridad
text("IBNS EEG VISUALIZER", 160, 40); //texto
text("Designed by Angel D. Yaguana", 200, 75); //texto

t.setValue(c.toString());
t.draw(this);
t.setPosition(520, 377);
int osc_blinkStrength = ((show_blinkStrength*2)+512);
smooth();
scope.addData(osc_blinkStrength);
```

```

scope.draw();
scope.drawBounds();

int s = second(); // Values from 0 - 59
int m = minute(); // Values from 0 - 59
int h = hour(); // Values from 0 - 23
text("Time :"+h+":"+m+":"+s, 400, 387); //texto

```

//La biblioteca eSense-EM ™ traduce las señales de EEG en diferentes niveles del estado mental, por ejemplo, la atención y la meditación. Estos "algoritmos" mentales deben ser probados en una muestra de población de alto y bajo diferentes condiciones ambientales, para trabajar con un amplio espectro de lugares individuales. Los resultados de la biblioteca eSense-EM se pasan a través de la secuencia de salida de la ThinkGear-EM a la plataforma final.

```

text("Poor_Signal_Event "+show_poorSignalEvent, x_pos, y_pos);
text("eSense", x_pos, y_pos);
text("Attention : "+show_attention, x_pos, y_pos);
text("Meditation: "+show_meditation, x_pos, y_pos);
text("Artefacts", x_pos, y_pos);
text("BlinkStrength: " + show_blinkStrength, x_pos, y_pos);
text("eegPower", x_pos+300, y_pos); //eegPower
text("delta Level: " + show_delta, x_pos+300, y_pos);
text("theta Level: " + show_theta, x_pos+300, y_pos);
text("low_alpha Level: " + show_low_alpha, x_pos+300, y_pos);
text("high_alpha Level: " + show_high_alpha, x_pos+300, y_pos);
text("low_beta Level: " + show_low_beta, x_pos+300, y_pos);
text("high_beta Level: " + show_high_beta, x_pos+300, y_pos);
text("low_gamma Level: " + show_low_gamma, x_pos+300, y_pos);
text("mid_gamma Level: " + show_mid_gamma, x_pos+300, y_pos);

if (show_poorSignalEvent==200) {
  text(" No contact with the skin, please ", x-300, y-30); //texto
  text(" reconnect sensor IBNS now!", x-300, y-15); //texto
  fill(255, 0, 0); //blanco RGB, Claridad letra
  ellipse(40, y-30, 20, 20);
}
if (show_poorSignalEvent>20 && show_poorSignalEvent<200) {
  text(" Low quality signal", x-200, y-30); //texto
  text(" check your sensor!", x-200, y-15); //texto
  fill(255, 155, 0); //blanco RGB, Claridad letra
  ellipse(40, y-30, 20, 20);
}
if (show_poorSignalEvent==0) {
  fill(0, 255, 0); //blanco RGB, Claridad letra
  text("IBNS_MindBot conected!", x-220, y-23);
  fill(0, 255, 0); //blanco RGB, Claridad letra
  ellipse(40, y-30, 20, 20);
}
} // Fin de GUI

```

//Funciones y métodos para extraer datos desde el API de NeuroSky usando el TGCD.

```
void poorSignalEvent(int sig) {
    show_poorSignalEvent = sig;
}

void attentionEvent(int attentionLevel) {
    show_attention = attentionLevel;
}

void meditationEvent(int meditationLevel) {
    show_meditation = meditationLevel;
}

void blinkEvent(int blinkStrength) {
    show_blinkStrength = blinkStrength;
}

void eegEvent(int delta, int theta, int low_alpha, int high_alpha, int low_beta, int high_beta, int low_gamma, int
mid_gamma) {
    show_delta=delta;
    show_theta=theta;
    show_low_alpha=low_alpha;
    show_high_alpha=high_alpha;
    show_low_beta=low_beta;
    show_high_beta=high_beta;
    show_low_gamma=low_gamma;
    show_mid_gamma=mid_gamma;
}

//Funcion Stop, usada para parar la extracción de datos desde el TGCD usando el API de NeuroSky.
void stop() {
    neuroSocket.stop();
    output.close(); // Finishes the file
    exit(); // Stops the program
    super.stop();
}
```

ANEXO 5.4

Creación del boceto SET_BT_Smirf

```
#define LED 13//led indicador
#define FACTORYRESETBAUD 57600//velocidad de TGAM1
#define DEFAULTBAUD 115200//Velocidad por default del BT_Smirf

char str[3];
char passkey[5] = "2912";//pss_key del Bluetooth de la IBNS
boolean success = false;
char mac[13] = "0006664F918F";//Direccion MAC del RN-41 en el modulo BT_Smirf
int failOuts[10] = {3,4,5,6,7,8,9,10,11,12};

void setup()
{
  //Inicializamos los pines
  pinMode(LED, OUTPUT);
  pinMode(BLUESMIRFON, OUTPUT);
  for (int i=0; i<10; i++) {
    pinMode(failOuts[i], OUTPUT);
  }
  // Reseamos el módulo Smirf a su configuración de fabrica
  while (!success) {
    RunBlueSmirfSetup(true);
  }
  success = false;
  // Configuramos la MAC adres remota a la que debe conectarse.
  RunBlueSmirfSetup(false);
}
```

En el siguiente paso se solicita la confirmación de configuración del módulo BT_Smirf y si es correcta nos comunica este suceso con un parpadeo del led.

```
void loop() {
  if(success) {
    digitalWrite(LED,LOW);
    delay(1000);
    digitalWrite(LED,HIGH);
    delay(1000);
  }
}

void RunBlueSmirfSetup(boolean factoryReset) {

  //Inicializa el Puerto serial del Atmega 328.
  if (factoryReset) {
    Serial.begin(FACTORYRESETBAUD);
  } else {
    Serial.begin(DEFAULTBAUD);
  }

  //Envia caracteres de sincronizacion al modulo para ponerlo en modo de programacion
  Serial.print('$');
  Serial.print('$');
  Serial.print('$');
  delay(100);
  Serial.flush();

  /Resetea el modulo a configuraciones de fabrica
  if (factoryReset) {
    Serial.print('$');
    Serial.print('F');
  }
}
```



```
Serial.print(',');
Serial.print('1');
Serial.print('\r');
```

```
while(Serial.available() < 3);
str[0] = (char)Serial.read();
str[1] = (char)Serial.read();
str[2] = (char)Serial.read();
if(str[0] == 'A' && str[1] == 'O' && str[2] == 'K') {
  success = true;
} else {
  success = false;
  digitalWrite(failOuts[0],HIGH);
}
delay(100);
Serial.flush();
} else {
```

```
//Configura la velocidad del UART a la que debe trabajar
```

```
Serial.print('S');
Serial.print('U');
Serial.print(',');
Serial.print('5');
Serial.print('7');
Serial.print('\r');
```

```
while(Serial.available() < 3);
str[0] = (char)Serial.read();
str[1] = (char)Serial.read();
str[2] = (char)Serial.read();
if(str[0] == 'A' && str[1] == 'O' && str[2] == 'K') {
  success = true;
} else {
  success = false;
  digitalWrite(failOuts[1],HIGH);
}
delay(100);
Serial.flush();
```

```
//Configura la dirección MAC remota a la que debe conectarse
```

```
Serial.print('S');
Serial.print('R');
Serial.print(',');
for(int i = 0; i < 12; i++) {
  Serial.print(mac[i]);
}
Serial.print('\r');
```

```
while(Serial.available() < 3);
str[0] = (char)Serial.read();
str[1] = (char)Serial.read();
str[2] = (char)Serial.read();
if(str[0] == 'A' && str[1] == 'O' && str[2] == 'K') {
  success = true;
} else {
  success = false;
  digitalWrite(failOuts[2],HIGH);
}
delay(100);
Serial.flush();
```

```
//Configura el PASS_KEY que posee la IBNS para poder emparejarse.
```

```
Serial.print('S');
Serial.print('P');
```

```

Serial.print(',');
for(int i = 0; i < 4; i++) {
  Serial.print(passkey[i]);
}
Serial.print('\r');

while(Serial.available() < 3);
str[0] = (char)Serial.read();
str[1] = (char)Serial.read();
str[2] = (char)Serial.read();
if(str[0] == 'A' && str[1] == 'O' && str[2] == 'K') {
  success = true;
} else {
  success = false;
  digitalWrite(failOuts[3],HIGH);
}
delay(100);
Serial.flush();

//Configura el modo de trabajo: Modo Maestro
Serial.print('S');
Serial.print('M');
Serial.print(',');
Serial.print('3');
Serial.print('\r');

while(Serial.available() < 3);
str[0] = (char)Serial.read();
str[1] = (char)Serial.read();
str[2] = (char)Serial.read();
if(str[0] == 'A' && str[1] == 'O' && str[2] == 'K') {
  success = true;
} else {
  success = false;
  digitalWrite(failOuts[4],HIGH);
}
delay(100);
Serial.flush();

delay(100);

//Envía caracteres de petición de salida del modo de programación del BT_Smirf
}
Serial.print('-');
Serial.print('-');
Serial.print('-');
Serial.print('\r');
}

```

ANEXO 5.5

Construcción del boceto Test_IBNS.ino en el IDE Arduino

```
#define BAUDRATE 57600
```

```
void setup()
{
  Serial.begin(BAUDRATE);
}
```

//En el siguiente paso se solicita la confirmación de recepción de datos en el buffer serial y en caso de ser un dato valido se presenta el valor en el puerto serial de salida hacia el PC para ser tomado por el TGDC mediante la GUI creada anteriormente o simplemente mediante un visor de datos de puerto serial a través del módulo Arduino Uno.

```
void loop() {
  ReadOneByte ();
}
```

//Creamos la función que nos permiten leer cada uno de los bytes recibidos en el puerto serial para poder visualizarlos mediante un terminal serial en el PC.

```
void ReadOneByte() {
  int ByteRead;
  while(!Serial.available());
  ByteRead = Serial.read();
  Serial.print((char)ByteRead);
}
```

ANEXO 6.1

Construcción del Firmware del módulo de control de periféricos.

```
Definiciones Globales
#define LED_blink A0
#define LED_PSQ A1
#define INT_LED A2
#define BAUDRATE 57600
```

```
#include <Arduino.h>
#include <TimerOne.h>
```

4.1.2 Variables globales para el programa.

```
//Variables de chequeo
byte generatedChecksum = 0;
byte checksum = 0;
int payloadLength = 0;
byte payloadData[64] = {0};
```

```
//Constantes de salida
int PARSER_SYNC_BYTE = 0xAA;
int PARSER_EXCODE_BYTE = 0x55;
int PARSER_POS = 0x01;
int PARSER_ATT = 0x02;
int PARSER_MED = 0x03;
int PARSER_BLK = 0x04;
int PARSER LENGHT = 0x0C;
int PARSER_SYNC_START = 0xAB;
int PARSER_SYNC_END = 0xAC;
```

```
//Variables de identificacion para IBNS
int poorQuality = 200; //0-255
int attention = 0; //0-100
int meditation = 0; //0-100
int BLINK_STRENGTH = 0; //0-255 solo aparece cuando hay un BLINK_STRENGTH
```

```
//Variables para almacenar valores ePower leidos
unsigned int delta;
unsigned int theta;
unsigned int alpha1;
unsigned int alpha2;
unsigned int beta1;
unsigned int beta2;
unsigned int gamma1;
unsigned int gamma2;
```

```
//variables del sistema
unsigned long lastReceivedPacket = micros();
boolean bigPacket = false;
int ledState = LOW;
boolean bit_blink = 0;
int CHSM=0;
```

4.1.3 Declaración de funciones del programa.

```
void ReadData();
unsigned long GetMicrosDelay(unsigned long t0);
byte ReadOneByte();
void loop();
void setup();
```

Funciones implementadas para manejo de datos recibidos.

```
//Lectura de un Byte
```

```

byte ReadOneByte() {
    int ByteRead;
    while (!Serial.available ()); //funcion que deshabilita todo si no hay nada que recibir!
    ByteRead = Serial.read();
    //Serial.print((char)ByteRead); // echo the same byte out the USB serial (for debug purposes)
    return ByteRead;
}

//Lectura y procesado del dato enviado desde modulo TGAM1
void ReadData() {
    static unsigned char payloadData[256];
    byte generatedChecksum;
    byte checksum;
    byte vLength;
    int payloadLength;
    int powerLength = 3;
    int k;
    bool eSensePacket = false;
    bool meterPacket = false;
    bool wavePacket = false;
    long wave = 0;
    long time;

    //Buscamos el byte de sincronización 0xAA
    if (ReadOneByte() == PARSE_SYNC_BYTE) {
        if (ReadOneByte() == PARSE_SYNC_BYTE) {

            //Filtro de antirebote de dato
            do {
                payloadLength = ReadOneByte();
            }
            while (payloadLength == 170);
            if (payloadLength > 170) {
                return;
            }
        }

        for (int i = 0; i < payloadLength; i++) {
            payloadData[i] = ReadOneByte();
            generatedChecksum += payloadData[i];
        }

        checksum = ReadOneByte();
        generatedChecksum = 255 - generatedChecksum;

        if (checksum == generatedChecksum ) {
            for (int i = 0; i < payloadLength; i++) {
                switch (payloadData[i]) {
                    case 0x02:
                        i++;
                        poorQuality = payloadData[i]; //Dato de Calidad de señal
                        eSensePacket = true;
                        break;
                    case 0x04:
                        i++;
                        attention = payloadData[i]; //Dato de aSense Atencion
                        eSensePacket = true;
                        break;
                    case 0x05:
                        i++;
                        meditation = payloadData[i]; //Dato de aSense Meditacion
                        eSensePacket = true;
                        break;
                    case 0x16:
                        i++;
                }
            }
        }
    }
}

```

```
    eSensePacket = false;
  } // Checksum Ok
  else {
  }
  } // end if read 0xAA byte
} // end if read 0xAA byte
} // ReadData
```

4.1.4 Funciones principales del boceto del programa diseñado.

```
void setup() {
  pinMode(LED_blink, OUTPUT);
  pinMode(LED_PSQ, OUTPUT);
  Serial.begin(BAUDRATE);
  Timer1.initialize(5000); //longest pulse
}

void loop() {
  ReadData(); //LECTURA DE DATO
}
```

ANEXO 6.2

Función de filtrado y detección de parpadeo de ojos mediante un Artefacto.

```
case 0x80: //Recogida de datos de muestreo de señales EEG reales en FP1
    wave = payloadData[i+2] * 256 + payloadData[i+3];
    time = Timer1.read();
    if (wave > 32767) wave -= 65535;
    if (wave>300 && time == 0) Timer1.start();
    else if (wave>300 && time > 100 && time < 350) {
        if (poorQuality==0) {
            bit_blink = 1;
            CHSM=poorQuality+attention+meditation+bit_blink;
            Serial.write(PARSER_SYNC_START);//0xAB
            Serial.write(PARSER_LENGTH);//0X0C
            Serial.write(PARSER_PQS);//0X01
            Serial.write(poorQuality);//poorQuality
            Serial.write(PARSER_ATT);//0X02
            Serial.write(attention);//attention
            Serial.write(PARSER_MED);//0X02
            Serial.write(meditation);//meditation
            Serial.write(PARSER_BLK);//0X02
            Serial.write(bit_blink);//bit_blink
            Serial.write(CHSM);//CHECKSUM
            Serial.write(PARSER_SYNC_END);//0X02
            Serial.print("\n");
            if (ledState == LOW) ledState = HIGH;
            else ledState = LOW;
            digitalWrite(LED_blink, ledState); delay(400);
        }
        bit_blink = 0; Timer1.stop();Timer1.restart();
    }
    else if (time < 100 || time > 350) {
        Timer1.stop();
        Timer1.restart();
    }
    i = i + 3;
    wavePacket = true;
    break;
```

ANEXO 6.3

Estudio de caso del sujeto 1:

Tabla 11 A-1 Datos del sujeto 1 para exploración EEG usando el TGAM1 enlazado al MCPE prototipado y al PC vía Bluetooth y puerto serial respectivamente:

Datos	Valoración
Edad	28 años.
Sexo	Masculino.
Discapacidad Física	Ninguna
Discapacidad Intelectual	Ninguna
Frecuencia cardiaca	73
Actividad	Reposo

Resultados de la exploración:

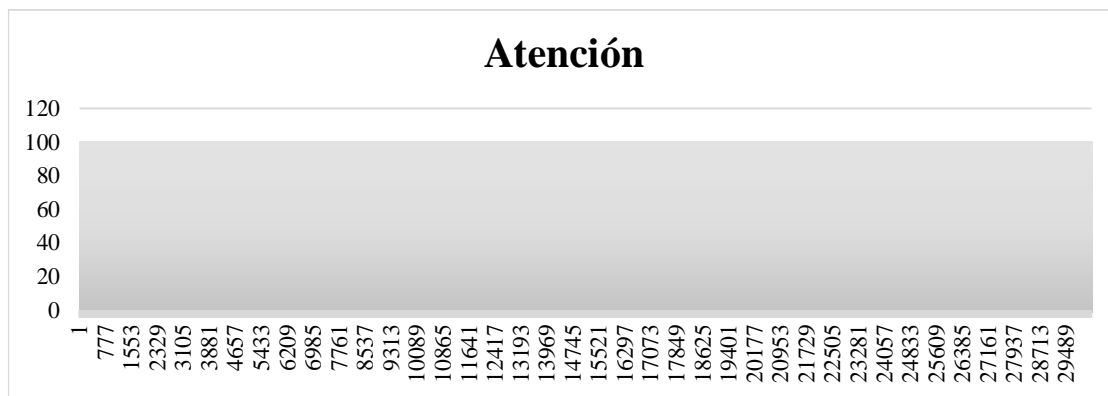


Figura 11 A-1 Resultados del muestreo de niveles de atención en el estudio de caso 1.

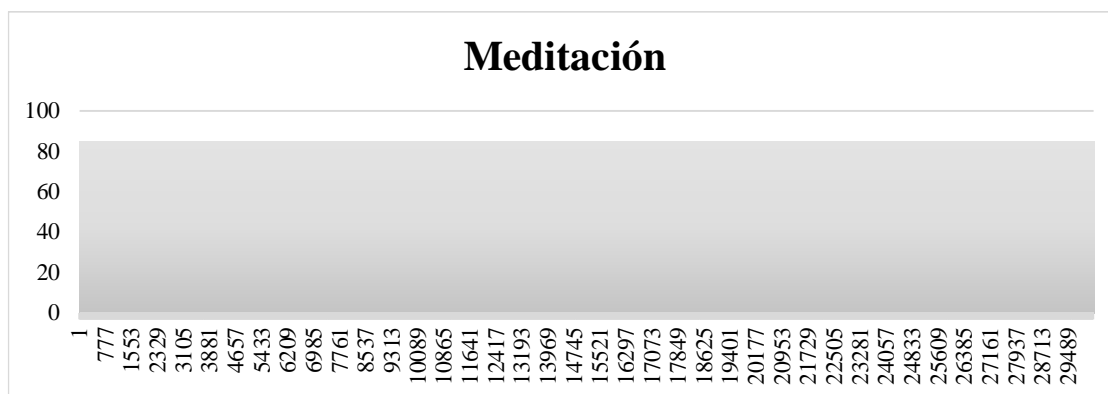


Figura 11 A-2 Resultados del muestreo de niveles de meditación en el estudio de caso 1.



Figura 11 A-3 Resultados del muestreo de niveles de intensidad de parpadeo en el estudio de caso 1.

ANEXO 6.4

Estudio de caso del sujeto 2:

Tabla 12 A-1 Datos del sujeto 2 para exploración EEG usando el TGAM1 enlazado al MCPE prototipado y al PC vía Bluetooth y puerto serial respectivamente:

Datos	Valoración
Edad	26 años.
Sexo	Femenino.
Discapacidad Física	Cardiaca, 50% CONADIS
Discapacidad Intelectual	Ninguna
Frecuencia cardiaca	132
Actividad	Reposo

Resultados de la exploración:

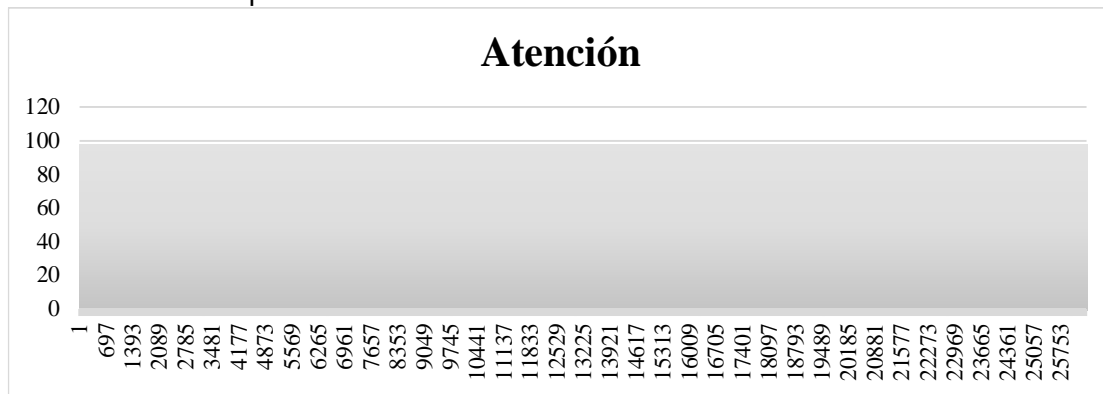


Figura 12 A-1 Resultados del muestreo de niveles de atención en el estudio de caso 2.

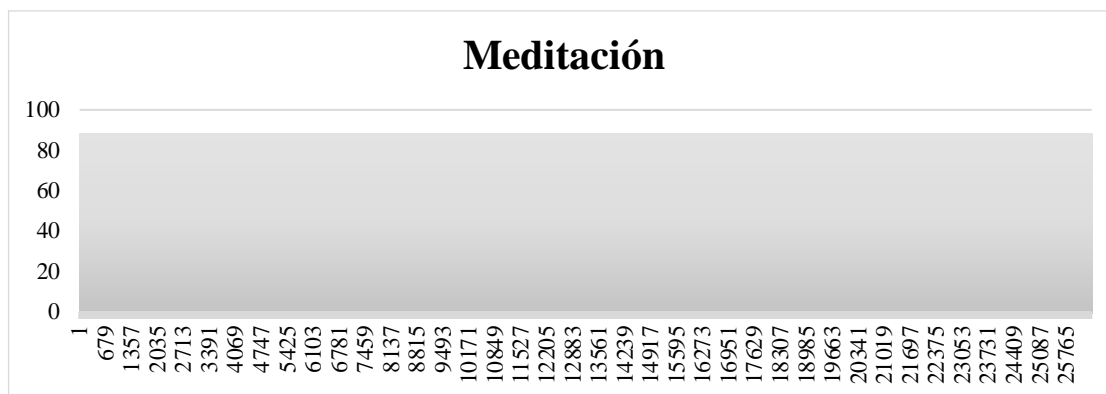


Figura 12 A-2 Resultados del muestreo de niveles de meditación en el estudio de caso 2.

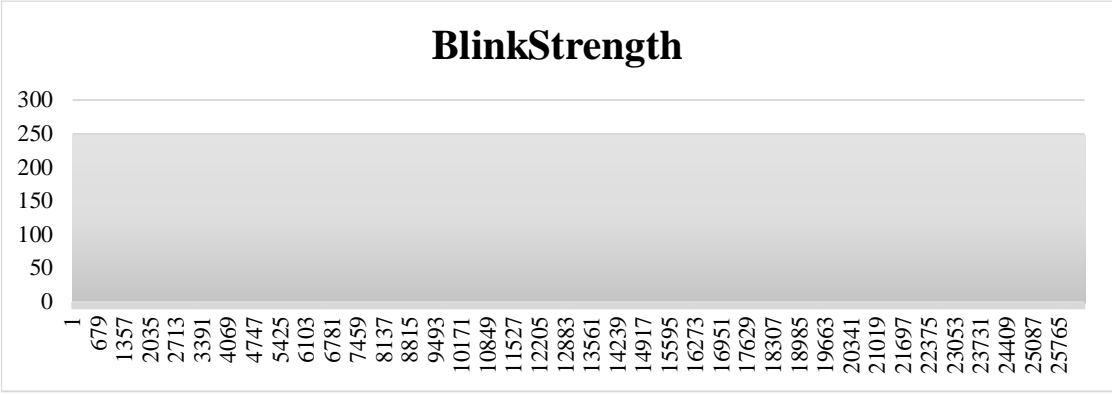


Figura 12 A-3 Resultados del muestreo de niveles de intensidad de parpadeo en el estudio de caso 2.

ANEXO 7.1

Prototipado de PCB's de laboratorio mediante transferencia de tinta laser en placas de cobre por medio de calor para ataque químico.

A continuación, se presenta la fabricación del PCB usando materiales básicos de prototipado por transferencia de calor.



Figura 10 A-1 Laboratorio acondicionado.



Figura 10 A-2 Materiales de fabricación de PCBs para prototipado.



Figura 10 A-3 Transferencia del fotolito mediante calor en una plastificadora.



Figura 10 A-4 Limpieza de residuos en el PCB para garantizar un buen ataque químico.



Figura 10 A-5 Materiales para atacado químico



Figura 10 A-6 Ataque Químico usando Percloruro de Hierro y Peróxido de Hidrogeno.

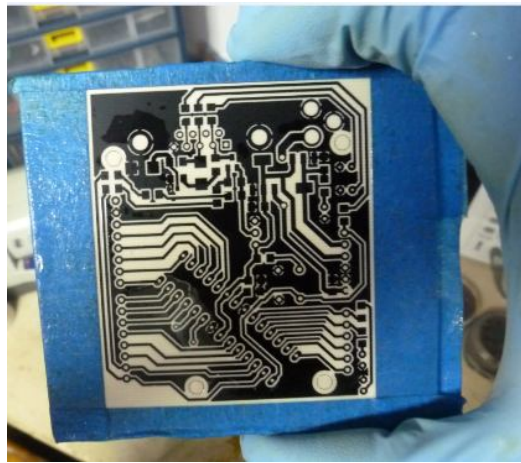


Figura 10 A-7 Limpieza del PCB con sosa caustica



Figura 10 A-8 Perforación del PCB.

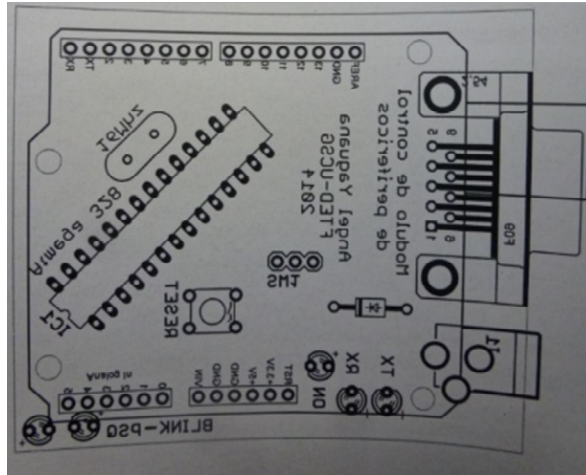


Figura 10 A-9 Fabricación de la serigrafía del PCB

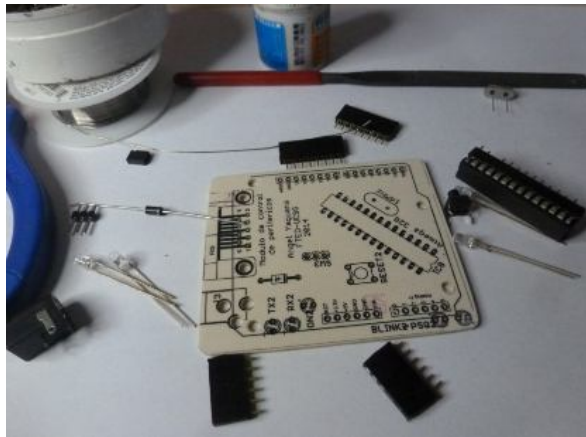


Figura 10 A-10 Construcción del PCB.

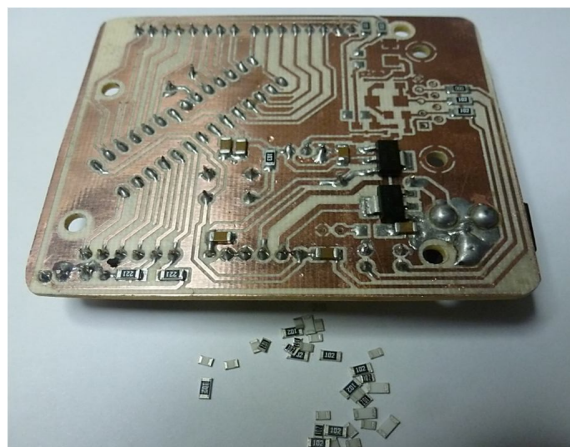


Figura 10 A-11 Soldadura de elementos SMD

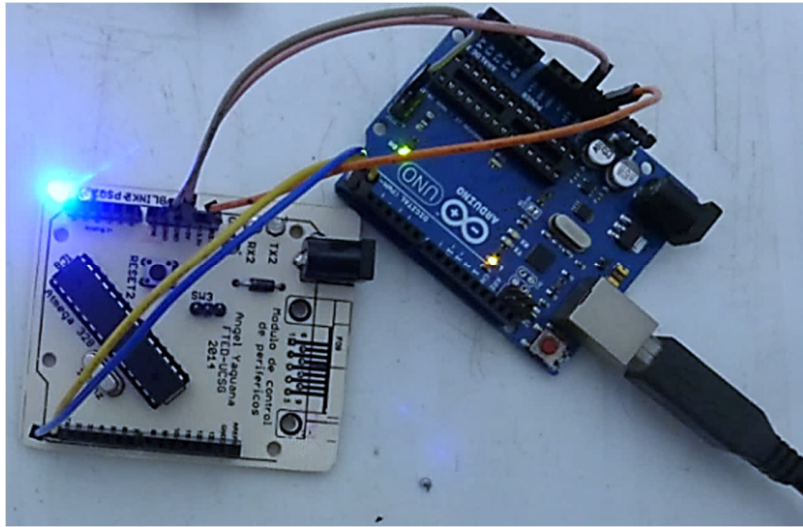


Figura 10 A-12 Carga del bootloader Arduino en el Atmega 328 y posteriormente del firmware realizado

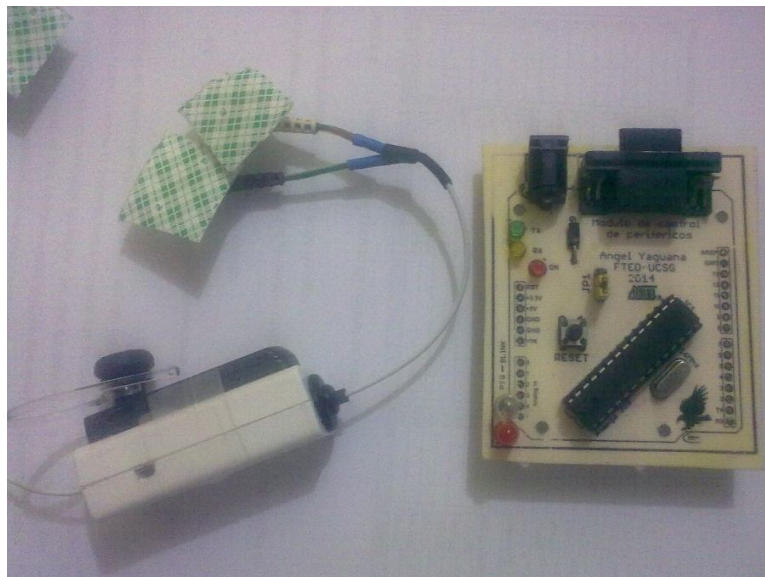


Figura 10 A-13 Producto final listo para pruebas de campo.

ANEXO 7.2

A continuación se presenta los detalles del estudio de caso realizado a 5 sujetos de distinta edad y género para observar el comportamiento de las variables eSense durante su exploración EEG usando el firmware desarrollado en el capítulo cuatro.

Tabla 13 A-1 Parámetros del estudio de caso.

Parámetro	Descripción y características
Sensibilidad.	1/Umbral.
Probabilidad de permanencia	50%.
Motivación	Encender el LED de color azul del MCP.
Tiempo	180 segundos
Sujeto	Los sujetos fueron escogidos por sus características físicas y fisiológicas para obtener variedad en los resultados y poder observar el comportamiento de las variables eSense de una manera experimental más diversa.
Procedimiento	Siendo un experimento del cual los sujetos no tienen conocimiento de cómo funciona el dispositivo, sino que, solo conocen las cualidades que determinan "que puede hacer", cada uno es instruido subjetivamente en que debe concentrarse para encender el LED de color Azul y se le da la opción de tener un estímulo físico para utilizarlo como referencia ya que el experimento está orientado a obtener los estados mentales que el sujeto refleja en esta circunstancia. Cabe mencionar que el último sujeto es un caso de estudio particular ya que es muy joven para comprender lo que realiza, sin embargo, es un caso exploratorio que vale la pena observar.

Tabla 13 A-2 Resultados de los estudios

Sujeto	Edad	Sexo	Discapacidad	Record (seg)	Umbrales		
					Atención	Meditación	Blink Strenght
1	28	Masculino	Ninguna	85	76	78	95
2	26	Femenino	Cardiaca 50%	9	86	78	60
3	54	Masculino	Visual 40%	82	100	72	150
4	85	Femenino	Ninguna	74	62	81	83
5	1 1/2	Masculino	Ninguna	146	92	83	98

Resultados de exploración EEG durante el estudio de los casos a los sujetos de prueba:

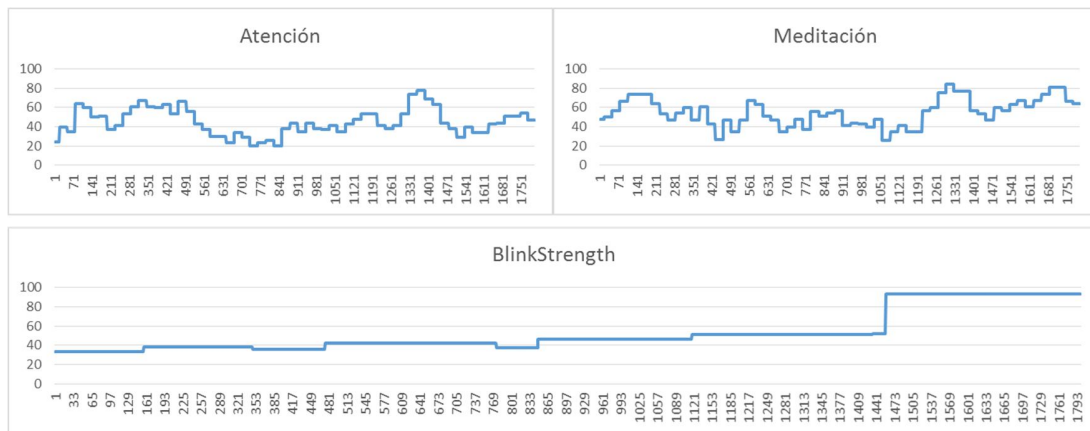


Figura 13 A-1 Resultados de exploración EEG del sujeto 1, "Adulto Joven".

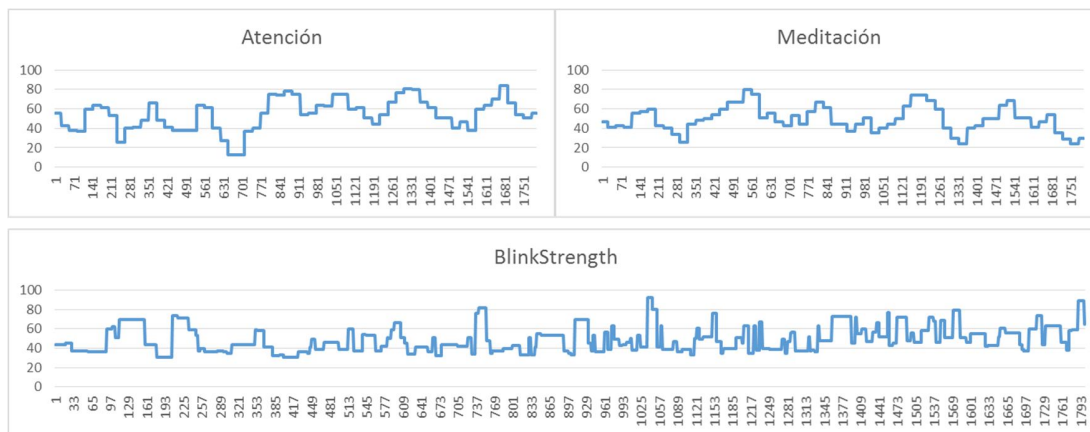


Figura 13 A-2 Resultados de exploración EEG del sujeto 2 "Mujer con Deficiencia Cardíaca".

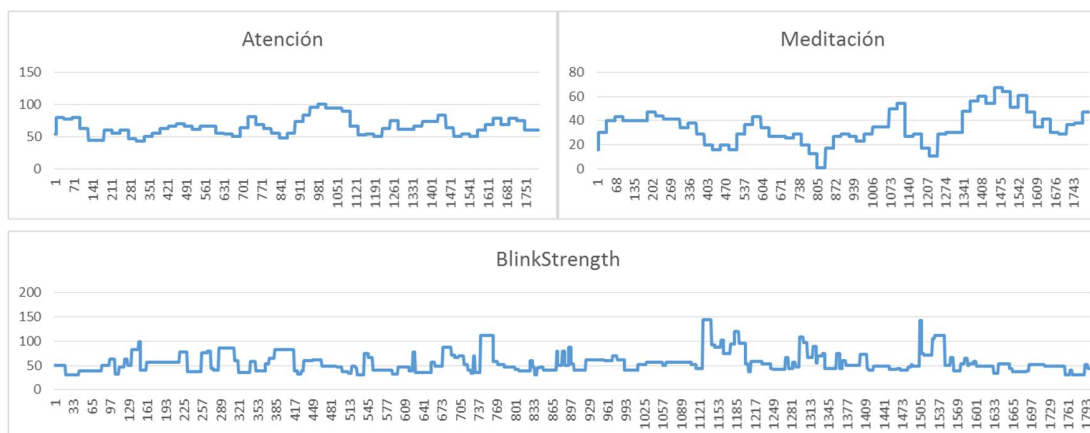


Figura 13 A-3 Resultados de exploración EEG del sujeto 3 "Hombre con Deficiencia Visual".

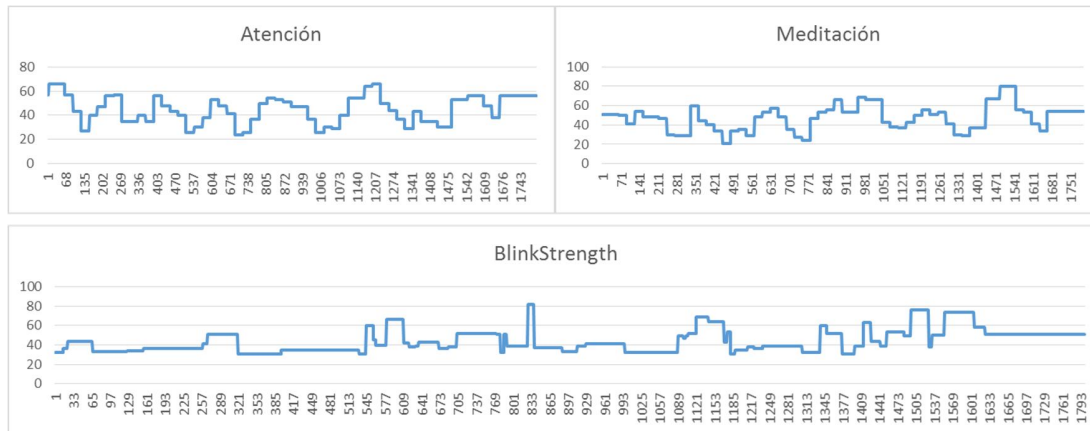


Figura 13 A-4 Resultados de exploración EEG del sujeto 3 "Mujer de 3ra Edad".

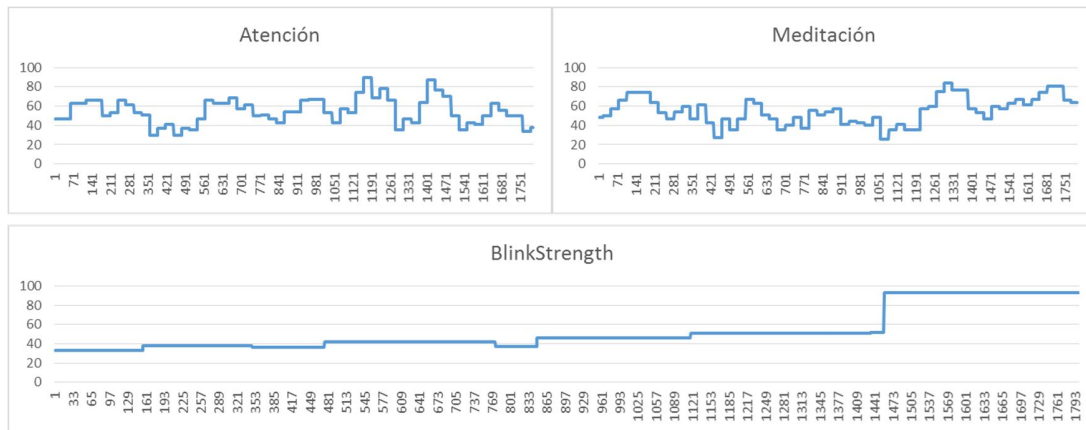


Figura 13 A-5 Resultados de exploración EEG del sujeto 3 "Bebe de 18 meses".

ANEXO 7.3

Resultados del análisis de calidad planteado a los clientes después de la prueba realizada con el MCP enlazado al TGAM1 mediante la IBNS.

Tabla 14 A-1 Resultados de la encuesta realizada al sujeto 1.

Descripción de solución presentada	Aprobación		
	Bajo	Medio	Alto
Interpretación de estados mentales			X
Fácil extracción de datos			X
Utilización de métodos no invasivos			X
Adaptabilidad a diferentes sistemas			X
Respuesta en tiempo considerable			X
Interfaz de usuario fácil de entender y manejar			X
Sistema de control remoto para controlar a distancia			X
Sistema de adquisición de datos de fácil extracción			X

Tabla 14 A-2 Resultados de la encuesta realizada al sujeto 2.

Solución presentada	Puntaje Aprobación		
	Bajo	Medio	Alto
Interpretación de los estados cerebrales			X
Fácil extracción de datos			X
Utilización de métodos no invasivos			X
Adaptabilidad a diferentes sistemas			X
Respuesta en tiempo considerable			X
Interfaz de usuario fácil de entender y manejar			X
Sistema de control remoto para controlar a distancia			X
Sistema de adquisición de datos de fácil extracción			X

Tabla 14 A-3 Resultados de la encuesta realizada al sujeto 3.

Solución presentada	Puntaje Aprobación		
	Bajo	Medio	Alto
Interpretación de los estados cerebrales			X
Fácil extracción de datos			X
Utilización de métodos no invasivos			X
Adaptabilidad a diferentes sistemas			X
Respuesta en tiempo considerable			X

Tabla 14 A-4 Resultados de la encuesta realizada al sujeto 4.

Solución presentada	Puntaje Aprobación		
	Bajo	Medio	Alto
Interpretación de los estados cerebrales			X
Fácil extracción de datos			X
Utilización de métodos no invasivos			X
Adaptabilidad a diferentes sistemas			X
Respuesta en tiempo considerable			X
Interfaz de usuario fácil de entender y manejar			X
Sistema de control remoto para controlar a distancia			X
Sistema de adquisición de datos de fácil extracción			X

Tabla 14 A-5 Resultados de la encuesta realizada al sujeto 5.

Solución presentada	Puntaje Aprobación		
	Bajo	Medio	Alto
Interpretación de los estados cerebrales		X	
Fácil extracción de datos			X
Utilización de métodos no invasivos			X
Adaptabilidad a diferentes sistemas			X
Respuesta en tiempo considerable			X
Interfaz de usuario fácil de entender y manejar		X	
Sistema de control remoto para controlar a distancia			X
Sistema de adquisición de datos de fácil extracción			X

Cabe indicar que el "sujeto 5" es un bebe de 18 meses y no puede dar datos sobre la prueba ya que aún no es consciente de sus actos, sin embargo, su madre lo ha hecho en su lugar