



**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 UN ROBOT ROVER TELEDIRIGIDO,  
USANDO LA INTERFAZ BIOMÉTRICA NEUROSENSORIAL PARA CONTROL  
DE PERIFÉRICOS MEDIANTE EL SENSOR THINKGEAR DE NEUROSKY  
ELECTRÓNICOS.**

**Autora**

Sophia Victoria Álava Oyola

**Director**

Ing. Carlos Romero

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 un robot Rover teledirigido, usando la interfaz biométrica neurosensorial para control de periféricos mediante el sensor ThinkGear de NeuroSky electrónicos.", fue realizado en su totalidad por la Sra. Sophia Victoria Álava Oyola, 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. Eduardo Mendoza Merchán, Msc.

REVISOR:

---

Ing. Washington Medina Moreira, 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, SOPHIA VICTORIA ÁLAVA OYOLA**

**DECLARO QUE:**

El proyecto de grado denominado "Desarrollo e implementación de un robot Rover teledirigido, usando la interfaz biométrica neurosensorial para control de periféricos mediante el sensor ThinkGear de NeuroSky 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

---

Sophia Victoria Alava Oyola  
C.I.# 0919803445



**UNIVERSIDAD CATÓLICA  
DE SANTIAGO DE GUAYAQUIL**

**FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO**

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

### **AUTORIZACIÓN**

**Yo, SOPHIA VICTORIA ALAVA OYOLA**

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 un robot Rover teledirigido, usando la interfaz biométrica neurosensorial para control de periféricos mediante el sensor ThinkGear de NeuroSky electrónicos", cuyo contenido, ideas y criterios son de mi exclusiva responsabilidad y total autoría.

Guayaquil, Mayo de 2014

LA AUTORA

---

Sophia Victoria Alava Oyola  
C.I.# 0919803445

## AGRADECIMIENTO

Agradezco a Dios por haberme concedido la vida, y a la vida por enfrentarme primero a los momentos que pensaba eran difíciles para luego entender lo fácil que es seguir adelante.

A mis mamás Martha Oyola, Lidia Tomalá mujeres inigualables, ejemplos de amor, fuerza y orgullo que me enseñaron valores, reprendieron y guiaron con dedicación mis etapas de vida.

A mi padre Álvaro Álava por ser ejemplo de fortaleza y a la vez de serenidad ante los obstáculos encontrados.

A mi hermana que partió a un viaje sin retorno pero que dejó una estrella eterna para iluminar sus recuerdos, a mis hermanos Kevin, Gema, Alexander, Joyce y Alvarito a quienes adoro.

A mi esposo por ser ejemplo de superación, de empeño y de lucha constante de conocimiento, por ser mi amigo, mi mentor, compañero de aventuras y sobre todo por el amor que brinda en nuestro hogar.

A todos los docentes de la institución, en especial al ingeniero L. Córdova, quien nos inculco el desarrollo de proyectos en robótica y microelectrónica, así mismo a los ingenieros L. Sánchez, E. Mendoza, E. Palacios, O. Philco, W. Medina, M. Romero también por impartir sus conocimientos que formaron mi perfil profesional, un agradecimiento especial a las autoridades que hicieron posible demás logros académicos, Ing. Walter Mera PhD, Vicerrector de Investigación y Posgrado, Lcda. Elizabeth Larrea Msc, ex Vicerrectora Académica, representantes del IEEE y WIE, a los amigos verdaderos y familiares que han formado parte de mi vida les quedo eternamente muy agradecida.

## **DEDICATORIA**

Dedico este proyecto a Dios, a mis madres por haber velado por mi educación y salud, a mí padre por compartir sus experiencias y conocimientos técnicos que contribuyeron a mi formación académica, a mi hermanos que han iniciado sus estudios superiores y que deseo así mismo los culminen exitosamente.

A mí esposo Angel Yaguana y nuestro amado hijo Nicolás Jérico que son el motor de mi vida y a quienes dedico todo esfuerzo.

***“La autoeducación es, estoy convencido, el único tipo de educación que existe.” (Asimov, 1992)***

# Índice General

<b>PORTADA</b> .....	I
<b>DECLARACIÓN DE RESPONSABILIDAD</b> .....	III
<b>AUTORIZACIÓN</b> .....	IV
<b>AGRADECIMIENTO</b> .....	V
<b>DEDICATORIA</b> .....	VI
<b>RESUMEN</b> .....	XII
<b>ABSTRACT</b> .....	XIII

## **CAPITULO 1**

<b>ASPECTOS GENERALES DEL TRABAJO DE TITULACIÓN</b> .....	1
1.1. Introducción .....	1
1.2. Contribución potencial del estudio .....	2
1.3. Planteamiento del problema .....	2
1.4. Problema .....	2
1.5. Justificación .....	2
1.6. Limitaciones del estudio .....	3
1.7. Objetivos .....	3
Objetivo general .....	3
Objetivos específicos .....	3
1.8. Hipótesis .....	4
1.9. Metodología .....	4
1.9.1. Justificación de la selección del método .....	4
1.9.2. Diseño de la investigación .....	4
1.9.3. Muestra .....	5
1.9.4. Técnicas de recogida de datos .....	5
1.9.5. Técnicas e instrumentos de análisis y procesamiento de datos ...	5

## **CAPITULO 2**

<b>FUNDAMENTACIÓN TEÓRICA</b> .....	6
2.1. Antecedentes .....	6
2.2. Bases teóricas o conceptuales .....	9
2.2.1. Ondas cerebrales .....	9
2.2.2. Dispositivo ThinkGear .....	11
2.2.3. Artefacto .....	11
2.2.4. Robot y su morfología de tipo Rover .....	13

## **CAPITULO 3**

<b>PRUEBAS DE CAMPO CON EL MODULO DE CONTROL DE PERIFERICOS</b> .....	15
3.1. Análisis del dispositivo .....	15
3.2. Análisis de tramas de datos .....	16
3.2.1. Estructura del paquete de la trama .....	16
3.2.2. Estructura y definición de códigos del Data Payload .....	16
3.3. Prueba de campo .....	18
Materiales .....	18



Informe del procedimiento .....	18
3.3.1. Resultados.....	19
Análisis .....	21
Discusión .....	21
<b>CAPITULO 4</b>	
<b>PRUEBAS DE CAMPO PARA ANALISIS DE MOVILIDAD DEL ROBOT.....</b>	<b>22</b>
4.1. Análisis de cinemática del robot.....	22
4.2. Implementación de Control difuso.....	24
4.2.1. Medición de distancia del obstáculo.....	24
4.2.2. Proceso de control difuso según el modelo Takagi y Sugeno.....	25
4.2.3. Modelamiento del sistema de control difuso.....	27
4.3. Módulo de motores con driver I293d.....	30
4.3.1. Prueba de campo .....	30
4.3. Diseño de etapa de potencia.....	33
4.4.1. Componentes del circuito .....	33
4.4.2. Esquemático del circuito.....	33
4.4.3. Tarjeta de circuito ruteada en Eagle.....	34
<b>DISEÑO DE FIRMWARE DEL ROBOT .....</b>	<b>36</b>
5.1. Determinación de umbrales.....	36
5.1.1. Diseño de un boceto de GUI.....	37
5.1.2. Resultados.....	37
5.2. Prueba al boceto.....	38
5.2.1. Procedimiento.....	39
5.2.2. Análisis .....	40
5.2.3. Discusión.....	40
5.3. Parámetros del diseño del Firmware.....	40
<b>CAPITULO 6</b>	
<b>DISEÑO DEL HARDWARE DEL ROVER SMART MINDBOT.....</b>	<b>43</b>
6.1. Morfología del robot.....	43
6.1.1. Electrónica .....	44
6.1.2. Telecomunicaciones .....	45
6.1.3. Mecánica.....	46
6.1.4. Software y comunicaciones .....	49
6.2. Construcción del robot. ....	49
6.2.1. Elaboración de los PCBs.....	49
6.2.2. Construcción de la morfología Rover del robot .....	49
6.2.3. Pruebas de comunicación.....	50
6.2.4. Análisis .....	50
6.3. Pruebas de campo del robot MindBot Smart Rover.....	51
<b>CONCLUSIONES.....</b>	<b>52</b>
<b>RECOMENDACIONES.....</b>	<b>54</b>
<b>GLOSARIO .....</b>	<b>55</b>

<b>REFERENCIAS BIBLIOGRÁFICAS</b> .....	67
<b>ANEXOS</b> .....	704

## Listado de Figuras

Figura 2.1	Silla de ruedas movida con ondas del cerebro.....	7
Figura 1.2	Brazo robótico usado en la investigación INIAC. ....	7
Figura 2.3	Características de las ondas cerebrales. ....	9
Figura 2.4	Alteraciones de la Actividad de Base .....	10
Figura 2.6	Artefactos identificados en un EEG.....	12
Figura 2.7	Robot Rover Escarabajo, usado para extraer petróleo. ....	14
Figura 3.1	Escucha de paquetes de datos vía puerto serial emitidos por el módulo de control de periféricos de la Interfaz Biométrica Neurosensorial, conectados al sujeto de prueba. ....	19
Figura 4.1	Formula de MRUV. Ley de cinemática. Newton .....	23
Figura 4.2	Sensor HC-SR04 conectado al módulo Arduino UNO. ....	25
Figura 4.3	Variable lingüística y su conjunto término. ....	27
Figura 4.4	Prueba de funcionamiento del control difuso implementado para detección de obstáculos.....	30
Figura 4.5	Fotografía tomada al módulo de motores de Arduino con el driver L293 sobre un Arduino Uno, conectado a los motores de prueba.....	32
Figura 4.6	Esquema del circuito diseñado para el control de motores.....	34
Figura 4.7	Diseño de la PCB de control de motores usando Eagle CadSoft. ....	35
Figura 5.1	Boceto GUI para visualización de datos EEG usando el API proporcionados por NeuroSky. ....	37
Figura 5.2	Reproducción del boceto de Processing creado para determinar los umbrales de los estados mentales Atención, Meditación y del parpadeo de Ojos. ....	38
Figura 5.3	Sujeto de prueba en acción con la GUI desarrollada y enlazada a la IBNS.....	38
Figura 5.4	Bloque Funcional del Rover Smart MindBot. ....	41
Figura 6.1	Bloque funcional de la electrónica del robot .....	44
Figura 6.2	Atemga 328p.....	45
Figura 6.3	Módulo Bluetooth RN- 41 .....	46
Figura 6.4	Tamiya 70194 Spike Tire Set (2 tires).....	47
Figura 6.6	Pruebas de comunicación del Robot MindBot Smart Rover con la PC usando la GUI diseñada para simular su control. ....	50
Figura 6.7	Prueba de campo realizada con la IBNS y el Robot en comunicación directa entre el sujeto de prueba y la interfaz de control. ....	51
Figura A-6.1	Materiales.....	96
Figura A-2.2	Transferencia del fotolito mediante calor. ....	96

Figura A-6.3 Limpieza de residuos de papel transfer en el PCB previo el atacado químico.	96
Figura A-6.4 Ataque Químico usando Percloruro de Hierro.	97
Figura A-6.5 Limpieza del PCB con sosa caustica	97
Figura A-6.6 Perforación del PCB.	97
Figura A-7.1 Materiales para construcción del robot.	98
Figura A-7.2 Ensamblaje de cuerpo de suspensión.	98
Figura A-7.3 Ensamblaje modular de la circuitería del robot.	99
Figura A-7.4 Presentación de la construcción del robot MindBot Smart Rover	99

### Listado de Tablas.

Tabla 3.1	Análisis cualitativo de la IBNS enlazada al MCPE.	15
Tabla 3.2	Estructura del paquete de la trama.	16
Tabla 3.3	Estructura y definición de códigos del Data Payload.	16
Tabla 3.4	Valores identificados en el primer paquete recibido.	20
Tabla 3.5	Valores identificados en el segundo paquete recibido.	20
Tabla 3.6	Valores identificados en el tercer paquete recibido.	21
Tabla 4.1	Variables de entrada para lógica difusa aplicada al robot.	27
Tabla 4.2	Variables de salida para lógica difusa aplicada al robot.	28
Tabla 4.3	Reglas de control para lógica difusa aplicada al robot.	28
Tabla 4.4	Componentes del circuito de control de motores.	33
Tabla 5.1	Parámetros del experimento.	39
Tabla 5.2	Resultados recolectados de las pruebas de experimentación.	40

### Listado de Flujogramas

Flujograma 4.1	Algoritmo de cálculo de distancia promedia.	24
Flujograma 4.2	Algoritmo para Control Difuso.	29
Flujograma 5.1	Algoritmo de control final usando los procesos Ping y Fussy.	41
Flujograma 5.2	Lógica de control basada en umbrales y eventos.	42

## RESUMEN

En este trabajo, se presenta la implementación de un sistema de control de un Rover teledirigido por medio de una interfaz cerebro-computador y su módulo para control de periféricos electrónicos, permitiendo con esto interpretar la actividad bioeléctrica de sus señales cerebrales con el fin de lograr maniobrabilidad y evasión autónoma.

Se señalan conceptos referenciales sobre la neurotecnología y se realiza pruebas funcionales y de campo que resultan en la creación de esquemas y algoritmos, utilizando para ello instrumentos de investigación como la Interfaz Biométrica Neurosensorial, kit's de desarrollo ofrecidos por NeuroSky, materiales electrónicos, equipos de medición y demás artículos de un laboratorio ambientado para este desarrollo.

El robot es controlado por medio de ondas cerebrales de las cuales se identifican patrones de comportamiento que denotan estados mentales y para darle dirección se usa una secuencia de registros de desplazamiento de eventos llamados Blink-Eyes que se producen eventualmente con la presencia del Artefacto EEG "Parpadeo de ojos". Estos datos, son recibidos desde el módulo de control de periféricos en forma serial.

Para su autonomía en cuanto a sus variables cinemáticas se implementó una estrategia de control difuso, que le permite administrar la frecuencia de trabajo de los motores para desplazarse con una determinada velocidad en una dirección indicada por el piloto, esto debido a que ha sido provisto de un algoritmo de detección de objetos que le permite tener control sobre su velocidad para evitar colisiones frontales.

Es preciso mencionar que el control difuso es efectivo, cuando se calibran las variables de entrada de acuerdo a los umbrales identificados en cada persona que necesite usar el dispositivo mostrando así los resultados esperados.

Con la experiencia entregada a través de esta investigación, se pretende que su metodología de control basada en la determinación de umbrales de los estados mentales y del parpadeo de ojos, otorgue al usuario un control efectivo y personalizado del robot y que éste al tener autonomía en su velocidad de desplazamiento pueda detenerse antes de colisionar evitando accidentes.

*Palabras clave: rover, control, robot, biométrico, interfaz, difuso, prótesis, neuro tecnología.*

## ABSTRACT

In this work, is presented the implementation of a control system of a remote Rover via a brain -computer interface module and peripheral electronic control, thereby allowing for interpreting the bioelectric activity of the brain signals in order to achieve autonomous maneuverability and evasion.

Referential concepts neurotechnology are reported and functional testing and field resulting in the creation of schemes and algorithms is performed, using research tools such as Interface Biometric Sensorineural, kit's development offered by NeuroSky, electronic materials, measuring equipment and other articles of a diner for this development laboratory.

The robot is controlled by brain waves which behavior patterns that denote mental states and to give direction to identify a sequence of shift registers events called Blink-eyes that eventually results in the presence of artifact EEG is used " Blink of an eye " . These data are received from the peripheral control module serially.

For autonomy in their kinematic variables a strategy of fuzzy control, which allows you to manage the working frequency of the motors to move with a certain speed in a direction indicated by the pilot, was implemented this because it has been provided with an algorithm for detection of objects that allows you to have control over their speed to avoid frontal collisions.

It should be mentioned that the fuzzy control is effective when the input variables according to the thresholds identified in each person need to use the device and showing the expected results are calibrated.

With the experience delivered through this research, it is intended that the control methodology based on thresholding of mental states and the blinking of an eye, give the user an effective and personalized control of the robot and to have autonomy in this its speed of movement can be stopped before colliding avoiding accidents.

Keywords: rover, control, robot, biometric, interface, diffuse, prosthetics, neuro technology.

## CAPITULO 1

### ASPECTOS GENERALES DEL TRABAJO DE TITULACIÓN

#### 1.1 Introducción

Los avances tecnológicos en nuestros días ya no sólo tratan de circuitos electrónicos dirigidos con sensores programados o con comandos enviados por un ordenador hacia un objeto para que este realice una determinada acción; sino que actualmente existen desarrollados sistemas de comunicación teledirigidos por las ondas cerebrales del ser humano quien emite o recepta la información integrando circuitos electrónicos con interfaces biométricas multisensoriales.

Como referencia se resaltan algunos desarrollos de la neurociencia en el campo de la medicina; en el año de 1936 utilizaron los patrones de iris en el campo de la oftalmología; seguidas de algunas patentes relacionadas destinadas al campo de estudio y seguridad de los sentidos encontramos: Pulso Cardíaco desarrollado por la Universidad del Estado de New México, Reflexión de ondas acústicas en la cabeza , desarrollado por el Departamento de Defensa de EEUU, Firma biodinámica patente del 2004 por OMPI, etc., (Sistemas Biométricos, 2012).

Como demostración de las facultades aplicativas de esta neuro tecnología se ha encontrado grandes aportaciones en el campo de la robótica, electrónica de consumo, marketing electrónico, juegos y sobre todo para mejorar la calidad de vida de las personas con capacidades especiales.

Al demostrar por medio de este proyecto un prototipo que pruebe la efectividad de una interfaz biométrica neurosensorial permitiendo la movilidad de un robot mediante estímulos mentales se entregará a la comunidad una herramienta que motive a futuros desarrollos.

En el transcurso de los capítulos se realizan pruebas de funcionalidad y campo para los sistemas implementados, algoritmos y estrategias de control, con los cuales se obtuvo la movilidad efectiva del robot por medio de los estados mentales.

## **1.2. Contribución potencial del estudio**

Al tener un amplio conocimiento de las capacidades de manejo de un robot mediante una BCI, mediante el uso de su algoritmo y estrategias de control, se puede contribuir potencialmente con el desarrollo de aplicaciones robóticas que busquen la creación de prótesis inteligentes que permitan realizar tareas de primera necesidad.

## **1.3. Planteamiento del problema**

Actualmente existe dificultad en los estudiantes y desarrolladores para crear aplicaciones de neurotecnología debido a que implica un profundo conocimiento técnico y científico en éste campo para poder interpretar apropiadamente los datos que son registrados en los sensores que capturan las señales EEG.

## **1.4. Problema**

Analizar la efectividad de la interfaz biométrica neuro sensorial y el módulo de control de periféricos electrónicos, para generar aplicativos neuro tecnológicos con el propósito de observar los beneficios obtenidos de su implementación y su rendimiento frente a las necesidades o interés.

## **1.5. Justificación**

Este trabajo se realiza con la motivación de innovar y proporcionar conocimientos y experiencias a la sociedad, entablando un diálogo de saberes entre desarrolladores de neurotecnología y las áreas interesadas, por medio del uso de una BCI para el desarrollo de aplicaciones que hasta ahora pudieron ser creadas usando técnicas tradicionales.

En consecuencia, producto de esta investigación, se puede pensar objetivamente en generar aplicaciones que a través de neuro señales que permitan a las personas interactuar o manipular máquinas y/o artefactos electrónicos fácilmente utilizando este trabajo de investigación adaptando el módulo de control de periféricos y mejorar la calidad de vida de las personas u otorgándole la utilidad necesaria.

## **1.6. Limitaciones del estudio**

Los estudios disponibles de neurotecnología actualmente son desarrollados en países extranjeros y en idiomas ajenos que usan lenguajes técnicos muy complejos, y además teniendo en cuenta que para desarrollar una aplicación comercial se necesita un profundo conocimiento en su área tecnológica, suficiencia en recursos económicos y tiempo de desarrollo, por lo cual este estudio se limita a crear una aplicación conceptual que determinará la efectividad del control de un robot mediante el uso de una BCI.

Otra limitación la constituye las alteraciones de las ondas cerebrales de cada individuo ya que por su edad, estado físico y mental existe una degradación de la señal EEG, por lo cual es necesario establecer los umbrales de funcionamiento.

## **1.7. Objetivos**

### ***Objetivo general***

Implementar un sistema de control para teledirigir un Robot usando la BCI TGAM1 de NeuroSky y su módulo de control de periféricos electrónicos.

### ***Objetivos específicos***

- a. Comprender conceptos sobre las ondas cerebrales, clasificación y alteraciones.
- b. Hacer pruebas funcionales y de campo con la interfaz biométrica enlazada al módulo ThinkGear de EEG comercializado por NeuroSky.
- c. Hacer pruebas funcionales y de campo con el módulo de motores de Arduino Uno para dar movilidad al robot.
- d. Elaborar una interfaz gráfica que permita visualizar los estados mentales de atención, meditación y parpadeo de ojos.
- e. Elaborar un algoritmo que mediante el uso de interfaz biométrica adoptada enlazada al módulo ThinkGear permita la manipulación de los movimientos del robot utilizando un microcontrolador.
- f. Elaborar los PCB's y el hardware del robot con estructura tipo Rover, realizando pruebas de comunicación con el algoritmo propuesto.



## **1.8. Hipótesis**

El uso de una aplicación que permita la posibilidad de teledirigir un robot mediante los estados mentales de una persona usando la BCI y su módulo para control de periféricos mediante el sensor ThinkGear de NeuroSky, ofrece la opción de desarrollar fácilmente aplicaciones neurotecnológicas mediante el aprendizaje experimental de las variables del sistema EEG.

## **1.9. Metodología**

El método a utilizar en el presente trabajo de investigación es el experimental con enfoque cuantitativo.

### ***1.9.1. Justificación de la selección del método***

Debido a que el proyecto es una investigación tecnológica, se propone el uso de la metodología indicada, ya que se basa en la observación, hipótesis, estudios realizados, en la medición de variables y sus efectos al ser alteradas.

El enfoque cuantitativo dimensiona al proyecto a existir con una realidad objetiva, ya que es posible cumplir con el objetivo aun modificando las variables independientes u obtener resultados sustentables para posterior estudios. Dentro de este enfoque se utiliza la lógica deductiva que va de lo general a lo particular, yendo de las teorías a los datos.

Puesto que, para la comprensión de algunos términos utilizados se realizaron pruebas funcionales y de campo, así mismo se realizó un muestro para conocer los resultados obtenidos debido a que las variables utilizadas dependen de los estado mentales.

### ***1.9.2. Diseño de la investigación***

Al ser una investigación cuantitativa se utilizó un diseño longitudinal ya que el interés del proyecto se lleva a cabo con los resultados obtenidos en los periodos de atención o meditación y parpadeo que el usuario determine, claro está que se recurrió al previo análisis y definición de umbrales.

En cuanto, también se aplica un diseño de campo ya que se obtienen resultados directamente de la realidad, demostrando en las muestras tomadas presentan diferentes resultados por los estados físicos del sujeto y el medio ambiente.

### **1.9.3. Muestra**

Se plantea usar el muestreo intencional en el cual “se selecciona los casos según el criterio de algún experto. Estas muestras son la importancia en las etapas exploratorias de la investigación, sobre todo si se utilizan a estos casos como informantes claves.” (Arena, 2008).

Por lo que, se seleccionaran intencionalmente diez muestras para el caso de detección de umbrales como componente esencial de investigación (hardware y ondas cerebrales).

### **1.9.4. Técnicas de recogida de datos**

Una vez seleccionado los casos de uso por cada componente esencial de investigación se procederá a realizar recolecciones de datos basándonos en observación y registro digital (en base a software) de los resultados con las modificaciones de las variables de entrada; con esto podemos tener la recolección apropiada para el tipo de investigación tecnológica que se propone ya que sus resultados son medibles y demostrables (Arias, 2008).

### **1.9.5. Técnicas e instrumentos de análisis y procesamiento de datos**

El análisis de datos utilizados es el descriptivo ya que se observará el comportamiento de cada muestra del estudio a través de las herramientas del procesamiento de datos.

El procesamiento de datos se representa con diagramas de flujos, tablas descriptivas, imágenes fotográficas y demostraciones reales.

## **CAPITULO 2**

### **FUNDAMENTACIÓN TEÓRICA**

#### **2.1. Antecedentes**

Los estudios en la neurotecnología estaban orientados a las personas que tenían algún tipo de discapacidad cerebral o neuronal, mediante equipos externos o implantes que permitían se comprenda de mejor manera cómo funciona el cerebro y como resultado de éstos se les ayudaba a recuperar con estímulos eléctricos el funcionamiento en específicas áreas del cerebro y con limitados grado de movilidad. (Millán, 2011)

Hoy en día existen sistemas BCI comercializados para otros fines por empresas como GTEC, Neurosky y Emotiv, (Merchán, 2008) encargados de ventas de equipos analizadores de las señales cerebrales y otras que ofrecen variedad en la industria de juegos, entornos adaptativos inteligentes, televisión interactiva, automoción, etc. (Emotiv, 2004)

Tomando como referencia los estudios realizadas por institutos o desarrolladores en esta ciencia que hayan utilizado sistemas BCI para aplicaciones móviles se encuentra el l'Ecole Polytechnique de Lausanne (EPFL) lugar donde se trabaja con enjambres de robots voladores para redes de comunicaciones y que han sido así mismo los precursores del proyecto de una silla robótica controlada con la mente, a cargo del catalán José Millán del Centro EPFL para neuroprótesis quien ha desarrollado una silla de ruedas que responde a la actividad cerebral mediante el seguimiento EEG en la cual los operadores sólo necesitan pensar en mover su mano y la silla se moverá automáticamente como si fuera parte de su cuerpo. (School of Engineering EPFL - CNBI, 2013)



Figura 2.1 Silla de ruedas movida con ondas del cerebro.  
Fuente: (School of Engineering EPFL -CNBI, 2013)

En el año 2010 la U.C.S.G. por medio del SINDE quien financió el Proyecto de Investigación Semilla “Investigación de Neuro Tecnologías Innovadoras Aplicadas Comercialmente” por sus siglas INIAC; se pudo obtener movimiento de un brazo robótico de 8 cm usando el ordenador como procesador de las señales recogidas por el EEG. (García, 2010) Manuscrito no publicado.

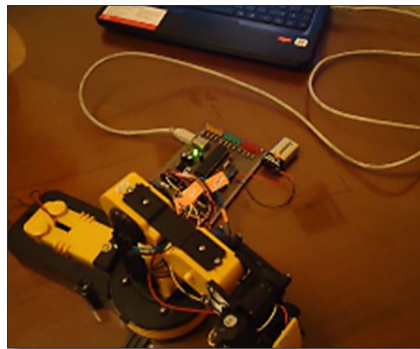


Figura 1.2 Brazo robótico usado en la investigación INIAC.  
Fuente: Proyecto de Investigación INIAC. (García, 2010)

Es preciso mencionar que las pocas implementaciones de esta tecnología en Ecuador han sido aplicadas por empresas de marketing para conocer las preferencias de los consumidores por medio de la atención hacia sus campañas.(García, 2010), con esto podemos determinar que es necesario promover a la neuro tecnología en un entorno educativo, de la cual nuestra facultad tendría ya un estudio.

En los informes generados del Ing. Roberto García en el año 2010 en los proyectos INIAC e INIAPCE se puede observar que demanda mucho tiempo y esfuerzo en cada uno de estos desarrollos y que así mismo requiere inyección de capital sustancial para cada uno de los proyectos, sin embargo, se muestra como efectivamente las técnicas y modelos de análisis de datos usadas en estas investigaciones pueden ayudarnos a implementar aplicativos o conceptos de gran ayuda a la sociedad a través de la implementación de modelos electrónicos e interfaces que reduzcan sustancialmente las limitantes de tiempo y capital necesarios..

En el presente año, el Sr. Angel Yaguana, Gerente de diseños electrónico y sistemas embebidos de la empresa APM Región Litoral, en calidad de estudiante investigador de la Facultad Técnica para el Desarrollo, de la UCSG, basándose en su experiencia de desarrollador en el proyecto INIAPCE, presentó en su trabajo de titulación de tesis un conjunto de herramientas permiten a un desarrollador profesional o estudiante realizar aplicaciones con neurotecnología a través del uso de un módulo que se puede enlazar via bluetooth fácilmente con dispositivos electrónicos para controlarlos mediante señales neuronales por medio del uso de un sensor ThinkGear el cual realiza un EEG en el punto FP1 de un paciente o en este caso de un piloto de pruebas. (Yaguana, 2014)

Usando la información brindada por este proyecto se adopta la interfaz biométrica y el módulo de control de periféricos para implementarla en una aplicación demostrativa desarrollando un Robot de morfología de tipo Rover el cual es teledirigido por medio de estímulos mentales que de acuerdo a los niveles de concentración, meditación y la detección de eventos Blink-eyes, permitirá al usuario o piloto controlar su dirección y velocidad en cuatro grados de libertad. A continuación, se procederá con una pequeña introducción de las ondas cerebrales captadas en el punto FP1 del sistema de localización estándar 10-20 para EEG y de las principales componentes de la interfaz biométrica adoptada, para generar un conocimiento referencial sobre los datos obtenidos y compararlos con los conocimientos adquiridos en el apartado anterior, finalmente se concluye con un diagnóstico generalizado en base a sus características técnicas.

## 2.2. Bases teóricas o conceptuales

### 2.2.1. Ondas cerebrales

El cerebro del ser humano está compuesto por millones de neuronas interconectadas. En su interacción los patrones se manifiestan como simples pensamientos y los estados emocionales se presentan como atención y meditación.

El promedio de pensamientos por día son de 70 mil, por cada interacción se crea una descarga eléctrica minúscula de la cual su frecuencia es medible únicamente por equipos de encefalografía ya que se percibe ondas desde 0.5 Hertz (Delta) hasta 50 Hertz (Beta), sin embargo con un estado dominante impulsado por la actividad colectiva de neuronas, pueden ser medibles en otros equipos.

Para entender las ondas cerebrales existen cuatro tipos distintos de ondas que toman su propia manera de percibir: procesos, aprendizajes, información, etc. todas estas se producen en cada momento y al crearse por un estado predominado o por voluntad son medibles. (Jem, 2013)



Figura 2.3 Características de las ondas cerebrales.  
Fuente: (Neurofeedback, 2013)

Las ondas Beta ocurren por pensamientos lógicos, con razonamiento, multitareas estrés y decisiones. Las ondas Alfa se caracterizan por la creatividad, inspiración, motivación, el sueño y relajación. Las ondas Teta presentes en el sueño tienen acceso a nuestros recuerdos y conciencia espiritual. Las ondas Delta están mayormente presentes en estado de sueño profundo o coma. (Neurofeedback, 2013)

A través del tiempo los expertos en la neurociencia han determinado en qué parte del cerebro es la responsable de actividades específicas realizadas por el cuerpo. Por ejemplo del control de nuestras extremidades se produce en la parte superior del cerebro, la visión se procesa en la parte posterior del cerebro, la memoria se desarrolla en la parte inferior de la corteza. Los estados mentales emocionales y la concentración se producen en la parte frontal del cerebro (Navarro, 2009); por esta razón se coloca el electrodo seco en esta área dominante para la correcta funcionalidad del módulo TGAM1.

### ***Alteraciones de la Actividad de Base***

Es importante tomar en cuenta que existe una variación en la actividad EEG de acuerdo a la edad, por ejemplo el cerebro del neonato es inmaduro y por el escaso entendimiento sus sentidos son incontrolados, mientras se desarrolla el crecimiento el cerebro ira moldeándose progresivamente, así mismo durante la tercera edad la actividad cerebral ira en disminución, puesto que producen también modificaciones en la actividad eléctrica del cerebro. Así mismo, en estados de estrés no es posible apreciar la actividad de las ondas alfa, por lo que es importante adoptar un estado mental de relajación durante la exploración EEG. (Granada, La Actividad Cerebral, 2005). En la figura 2.4 se muestran las fluctuaciones de las señales EEG según las edades estudiadas y las patologías más relevantes.

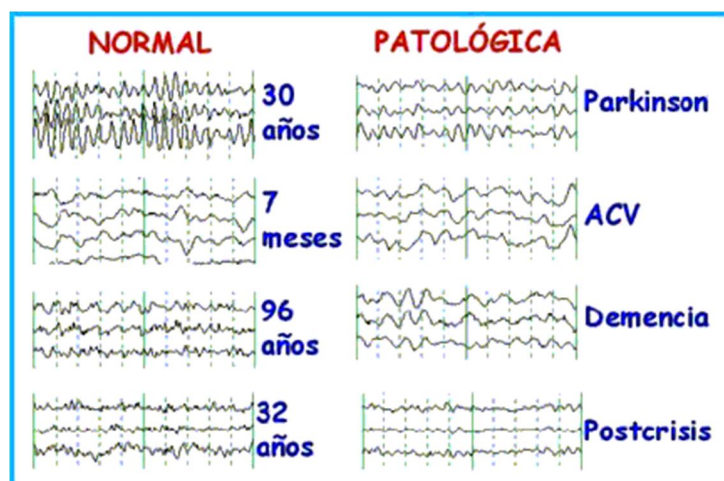


Figura 2.4 Alteraciones de la Actividad de Base  
Fuente: (Granada, Actividad Patológica, 2005)

### 2.2.2. Dispositivo ThinkGear

El dispositivo ThinkGear es el módulo comercializado por la empresa NeuroSky que permite recoger la señal EEG mediante la adquisición, amplificación, filtrado y el análisis de las señales de las ondas cerebrales para colocar esta información a disposición de las aplicaciones compatibles.

El TGAM1 procesa señales EEG y mediante un algoritmo propio de NeuroSky obtiene los valores eSense de atención y meditación, así mismo muestra mediante el dato RAW EEG los artefactos conocidos como parpadeo, usando para ello electrodos secos simples, localizados en el punto FP1 del sistema de localización estándar 10-20 para EEG, permitiendo distinguir la señal del cerebro con la del ruido para lo cual utiliza algoritmos propios codificados en la memoria del chip. La comunicación se realiza a través de una interfaz UART a niveles TTL de bajo voltaje, y opera con una fuente de 2.8 a 3.7v. (NeuroSky, Inc., 2010)

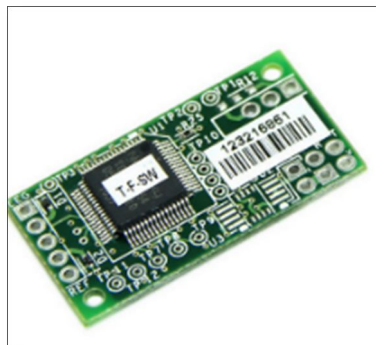


Figura 2.5 Módulo ThinkGear Asic  
Fuente: (Seedstudio, 2013)

El módulo TGAM1 viene acompañado de una distribución oficial API la cual permite mediante una DLL llamada THINKGEAR CONECTOR que se ejecuta en la PC, obtener los paquetes de datos que contienen el EEG. (Soo, 2013)

### 2.2.3. Artefacto

Un Artefacto es una señal no deseada dentro del registro EEG. Los artefactos pueden dividirse en dos categorías: fisiológicos y no fisiológicos.



Cualquier fuente en el cuerpo que tiene un dipolo eléctrico o genera un campo eléctrico es capaz de producir artefactos fisiológicos. Estos incluyen el corazón, los ojos, los músculos faciales, y la lengua. La sudoración también puede alterar la impedancia en la medida de los electrodos e interactuar y producir un artefacto. (Granada, La Actividad Cerebral, 2005)

### **Clasificación de Artefactos**

Se pueden dividir en fisiológicos y extra fisiológicos. Los primeros surgen del paciente directamente como el movimiento ocular ya sea por espasmo semifacial, estímulos de luz, las pulsaciones del ojo y movimientos de la lengua, respiración en una actividad rítmica que puede causar cambios de impedancia sobre un electrodo y los segundos se generan por fuera del paciente como los equipos o el medio ambiente. (H. García, 2009)

Estos artefactos pueden ser apreciados en la siguiente imagen:

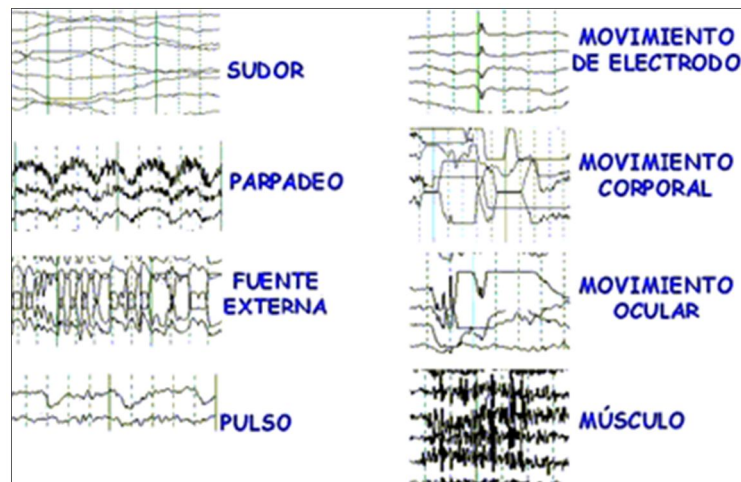


Figura 2.6 Artefactos identificados en un EEG.  
Fuente: (Granada, La Actividad Cerebral, 2005)

Cabe mencionar que hoy en día los equipos de medición han perfeccionado sus técnicas por lo que es posible realizar mediante ordenadores o sistemas modernos mejores capturas para el análisis y procesamiento de las señales EEG.

#### **2.2.4. Robot y su morfología de tipo Rover**

##### ***Introducción a la robótica***

La robótica es la ciencia que estudia el diseño y la implementación de robots, conjugando múltiples disciplinas, como la mecánica, la electrónica, la informática, la inteligencia artificial y la ingeniería de control, entre otras. (Tecnología, 2014). Para definirlo en términos generales, un robot es una máquina automática o autónoma que posee cierto grado de inteligencia, capaz de percibir su entorno y de imitar determinados comportamientos del ser humano. Los robots se utilizan para desempeñar labores riesgosas o que requieren de una fuerza, velocidad o precisión que está fuera de nuestro alcance. El término robot tiene su origen en el título de la obra teatral *Robots Universales Rossum*, escrita por el novelista y dramaturgo checo Karel Capek, en 1920. La palabra checa *robot*, que significa "trabajos forzados", fue traducida al inglés como *robot*. (Ministerio de Educación, 2012). Comprendiendo los conceptos antes nombrados se añade que el robot desarrollado es autónomo ya que dotará con capacidad de percepción y reacción ante su entorno.

##### ***Morfología Rover***

Los robots tipo Rover son llamados así por su similitud con los vehículos de la marca MG Rover diseñados para la exploración suelen, están equipados con algún tipo de función, en caso de detección de incendios, detección de un objeto, el transporte de los objetos, y otros aplicativos según la necesidad de cada proyecto y circunstancia para dar una solución a un problema en el cual el humano por sí mismo podría estar incapacitado en parte o su totalidad. Los Rover se utilizan generalmente para explorar el bosque o en el espacio ultraterrestre, incluso la exploración de la Luna, y del planeta Marte como lo hizo la NASA tomando estudios de pruebas de concepto en circunstancias similares a estos ambientes en lugares como el desierto del Sahara, los desiertos de Arizona, Nevada, Perú, Chile y en lugares con condiciones extremas como Siberia y la planicie de la Antártida. Para este proyecto se ha tomado en cuenta este diseño debido a que es el más simple en cuanto a control de movimiento, dado que suele estar equipado con un mínimo de cuatro ruedas.



Figura 2.7 Robot Rover Escarabajo, usado para extraer petróleo.  
Fuente: (Mellon, 2008)

## CAPITULO 3

### PRUEBAS DE CAMPO CON EL MODULO DE CONTROL DE PERIFERICOS

En siguiente capítulo se realizan pruebas de campo utilizando la IBNS y el módulo de control de periféricos desarrollada por el Sr. Angel Yaguana en su proyecto de fin de carrera de Ingeniería en Telecomunicaciones de la Facultad Técnica en la Universidad Católica de Santiago de Guayaquil, para realizar un análisis cualitativo, conocer la estructura del paquete de la trama, realizar las pruebas de campo mostrando sus resultados.

#### 3.1. Análisis del dispositivo

Como resultado de conectar la Interfaz Biométrica Neuro-Sensorial mediante Bluetooth al módulo para control de periféricos se obtuvo el siguiente análisis cualitativo.

Tabla 3.1 Análisis cualitativo de la IBNS enlazada al MCPE.

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

Fuente: Autor S. Álava

### 3.2. Análisis de tramas de datos

#### 3.2.1. Estructura del paquete de la trama

La trama de datos de la Interfaz Biométrica Neuro-Sensorial es enviada como un arreglo de bits serial asíncrono vía UART, a 57600b, 8, 1, n y puede ser recibida desde cualquier dispositivo de comunicación con puerto Serial. Cada paquete enviado posee la siguiente estructura:

Tabla 2.2 Estructura del paquete de la trama.

SYNC	PLENGHT	PAYLOAD								CHKSUM	END SYNC
0xAB	PL	0X01	PSQ	0X02	ATT	0X03	MED	0X04	BLINK	CHKSM	0xAC

Fuente: (Yaguana, 2014)

#### 3.2.2. Estructura y definición de códigos del Data Payload

Tabla 3.3 Estructura y definición de códigos del Data Payload.

0xAB	Sincroniza el módulo de control de periféricos con el dispositivo acoplado para toma de datos mediante puerto serial.
PL	[PLENGTH] Este byte indica la longitud en bytes de la sección de paquetes PAYLOAD los cuales son los datos de carga útil y que pueden variar desde 0 hasta 100, cualquier error más alto indica un error en la carga útil. Este dato fue diseñado con la finalidad de poder adaptar el decodificador a las variaciones futuras que pueda tener la IBNS.
0x01	[POOR QUALITY SIGNAL] PSQ. 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 puede 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 de 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 02	[eSense Attention] Es el código que indica la lectura de atención (ATT) 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 03	[eSense Meditation] Es el código que indica la lectura de meditación (MED) de acuerdo a los parámetros establecidos por eSense, al igual que el anterior tiene la misma variación 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 memorando 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.
0x04	Es el código que identifica el comienzo del evento parpadeo (BLINK).
CHKSM	Con este byte se identifica que los valores obtenidos son correctos haciendo una sumatoria entre los eventos eSense (ATT+MED)+ Blink.
0xAC	Es el código que identifica el final de la trama.

Fuente: (Yaguana, 2014)

### 3.3. Prueba de campo

Para determinar si factible el uso de la interfaz biométrica neurosensorial conectada al módulo de control de periféricos adoptado se realizará la siguiente prueba de campo con materiales adecuados y se presentara los resultados obtenidos de una muestra de 3 tramas.

#### Objetivo de la prueba

Visualizar las tramas de comunicación que envía la interfaz conectada en el punto FP.1 mediante el módulo TGAM1 e identificar los caracteres del protocolo de comunicación especificados por el proveedor.

#### *Materiales*

- Módulo TGAM1.
- Software Hércules versión 3.2.4
- Puerto COM conectado al módulo de control de periféricos de la interfaz.
- Computadora

#### *Informe del procedimiento*

Se conectó la interfaz mediante cable serial DB9 con conversión de señales CMOS a TTL y se instaló el driver de puerto serial para crear un puerto serial virtual en una PC con Windows 8, al mismo que se configuró en 57600b, 8, 1, n, para visualizar los datos que entrega al terminal serial Hércules del cual se consiguieron los siguientes registros:

```
{3D}{04}{00}{69}{AC}{0A}{AB}{0C}{01}{00}{02}{3D}{03}{40}{04}{00}{7D}{AC}{0A}{AB}{0C}{01}{00}{02}{1E}{03}{45}{04}{00}{63}{AC}{0A}{AB}{0C}{01}{00}{02}{29}{03}{4A}{04}{00}{73}{AC}{0A}{AB}{0C}{01}{00}{02}{33}{03}{53}{04}{00}{86}{AC}{0A}{AB}{0C}{01}{00}{02}{33}{03}{53}{04}{01}{87}{AC}{0A}{AB}{0C}
```

Los mismos que se pueden visualizar a continuación:

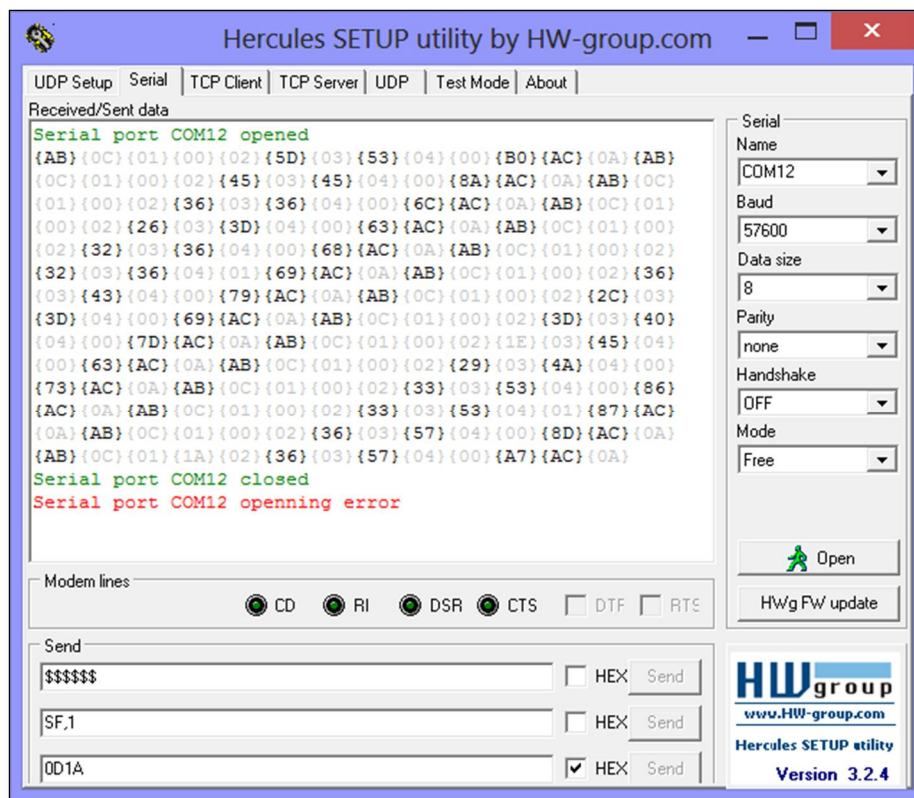


Figura 3.1 Escucha de paquetes de datos vía puerto serial emitidos por el módulo de control de periféricos de la Interfaz Biométrica Neurosensorial, conectados al sujeto de prueba.

Fuente: Impresión de pantalla del programa Hércules SETUP versión 3.2.4.

### 3.3.1. Resultados

El examen EEG, registró una data captada desde el MCPE de la cual los valores recibidos permiten efectivamente visualizar los datos de PSQ, Atención, Meditación y parpadeo tal cual lo indica el proveedor, demostrando la efectividad de su funcionamiento y al mismo tiempo proporcionando una trama limpia, segura y fácil de manejar.

#### *Inspección de la trama*

De la lectura recibida en el paso anterior se pueden visualizar los siguientes datos:



{AB}{OC}{01}{00}{02}{5D}{03}{53}{04}{00}{B0}{AC}{0A}{AB}{OC}{01}{00}{02}{45}{03}{45}{04}{00}{8A}{AC}{0A}{AB}{OC}{01}{00}{02}{36}{03}{36}{04}{00}{6C}{AC}{0A}{AB}{OC}{01}{00}{02}{26}{03}{3D}{04}{00}{63}{AC}{0A}{AB}{OC}{01}{00}{02}{32}{03}{36}{04}{00}{68}{AC}{0A}{AB}{OC}{01}{00}{02}{32}{03}{36}{04}{01}{69}{AC}{0A}{AB}{OC}{01}{00}{02}{36}{03}{43}{04}{00}{79}{AC}{0A}{AB}{OC}{01}{00}{02}{2C}{03}{3D}{04}{00}{69}{AC}{0A}{AB}{OC}{01}{00}{02}{3D}{03}{40}{04}{00}{7D}{AC}{0A}{AB}{OC}{01}{00}{02}{1E}{03}{45}{04}{00}{63}{AC}{0A}{AB}{OC}{01}{00}{02}{29}{03}{4A}{04}{00}{73}{AC}{0A}{AB}{OC}{01}{00}{02}{33}{03}{53}{04}{00}{86}{AC}{0A}{AB}{OC}{01}{00}{02}{33}{03}{53}{04}{01}{87}{AC}{0A}{AB}{OC}{01}{00}{02}{36}{03}{57}{04}{00}{8D}{AC}{0A}{AB}{OC}{01}{1A}{02}{36}{03}{57}{04}{00}{A7}{AC}{0A}

Se han identificado varios paquetes con valores PSQ, eSense y Blink, se analiza a continuación tres de ellos y se generaliza conclusiones.

Tabla 3.4 Valores identificados en el primer paquete recibido

byte:	valor	[CODE]
[ 0]	0xAB	[SYNC]
[ 1]	0x0C	[PLENGTH]
[ 2]	0x01	[POOR_SIGNAL]
[ 3]	0x00	(0/200) Calidad afectada= 0
[ 4]	0x02	[ATTENTION] eSense
[ 5]	0x5D	eSense, Nivel de atencion:93
[ 6]	0x03	[MEDITATION] eSense
[ 7]	0x53	eSense Nivel de Meditacion: 83
[ 8]	0x04	[BLINK STRENGTH] eSense artefact
[ 9]	0x00	(0/1) Evento Parpadeo=0
[10]	0xB0	[CHKSUM] =176
[11]	0xAC	[END SYNC]

Fuente: Autor S. Álava

Tabla 3.5 Valores identificados en el segundo paquete recibido

byte:	valor	[CODE]
[ 0]	0xAB	[SYNC]
[ 1]	0x0C	[PLENGTH]
[ 2]	0x01	[POOR_SIGNAL]
[ 3]	0x00	(0/200) Calidad afectada = 0
[ 4]	0x02	[ATTENTION] eSense
[ 5]	0x36	eSense, Nivel de atencion:54
[ 6]	0x03	[MEDITATION] eSense
[ 7]	0x36	eSense Nivel de Meditacion: 54
[ 8]	0x04	[BLINK STRENGTH] eSense artefact
[ 9]	0x00	(0/1) Evento Parpadeo=0
[10]	0x6C	[CHKSUM] =108
[11]	0xAC	[END SYNC]

Fuente: Autor S. Álava

Tabla 3.6 Valores identificados en el tercer paquete recibido

byte:	valor	[CODE]
[ 0]	0xAB	[SYNC]
[ 1]	0x0C	[PLENGTH]
[ 2]	0x01	[POOR_SIGNAL]
[ 3]	0x00	(0/200) Calidad afectada = 0
[ 4]	0x02	[ATTENTION] eSense
[ 5]	0x32	eSense, Nivel de atencion:50
[ 6]	0x03	[MEDITATION] eSense
[ 7]	0x36	eSense Nivel de Meditacion: 54
[ 8]	0x04	[BLINK STRENGTH] eSense artefact
[ 9]	0x01	(0/1) Evento Parpadeo=1
[10]	0x69	[CHKSUM] =105
[11]	0xAC	[END SYNC]

Fuente: Autor S. Álava

### ***Análisis***

Como se pudo observar, las tramas de datos enviadas por la interfaz permiten visualizar de manera sencilla los valores Hexadecimales de PSQ, eSense y Blink sin la necesidad de utilizar un complejo algoritmo de decodificación como lo era antes de la implementación de esta útil herramienta, permitiendo que un individuo con conocimientos generales del sistema hexadecimal de codificación pueda en tiempo real identificar los paquetes recibidos y su carga útil, más aun cuando los últimos dos bytes permiten al software Hércules separar la carga útil de los paquetes enviados mediante el uso de los bytes 0x0D o 0x0A.

### ***Discusión***

Diseñar una aplicación sencilla basada en la GUI IBNS\_EEG\_VISUALIZER realizada en el IDE Processing para poder capturar directamente los datos recibidos por el módulo TGAM1 en tiempo real y compararlos con los datos entregados por el módulo de control de periféricos, elaborando un entorno similar a un juego el cual sirva de incentivo a futuras generaciones de estudiantes para incursionar en el desarrollo de aplicaciones basadas en neuro-tecnología.

## **CAPITULO 4**

### **PRUEBAS DE CAMPO PARA ANALISIS DE MOVILIDAD DEL ROBOT**

En el presente capítulo se realiza pruebas de campo al módulo de motores del entorno Arduino.

#### **4.1. Análisis de cinemática del robot**

Esta área de desarrollo se centra en el estudio del control del movimiento con respecto a un sistema de referencia en un eje horizontal, es decir sobre una superficie sobre la cual se va a desplazar. Comprendiéndose de dos etapas importantes:

- La descripción analítica del movimiento espacial en función del tiempo, para la cual se desarrolla una lógica que le permita tomar decisiones de manera autónoma en cuanto a su velocidad y aceleración en una dirección y distancia determinada.
- Las relaciones de localización del extremo del tramo a recorrerse por el robot.

En cuanto a la descripción analítica del movimiento es claramente un problema cinemático del tipo MRUV (movimiento rectilíneo uniformemente variado) ya que se debe determinar la posición y de acuerdo a la orientación del extremo final del robot con respecto a un sistema de coordenadas de referencia de desplazamiento para generar los parámetros cinemáticos de desplazamiento del robot.

El problema de su movilidad es entonces un problema cinemático inverso debido a que el robot debe determinar la configuración que debe adoptar para avanzar a una posición del extremo de orientación conocido.

Este problema debe entonces ser tratado con un análisis de estrategias de control para determinar cuál es la forma más apropiada en base a sus variables intervinientes y solucionarlo de manera efectiva.

Una de las estrategias de control más eficaces es el control PID, este control nos permite estipular un punto de equilibrio de una variable que en este caso podría ser la velocidad a la que el robot avanza y mediante un control de PWM (modulación por ancho de pulsos) rotar los motores con la frecuencia que se ajuste a este punto de equilibrio, sin embargo, este no es el modelo que debemos usar, puesto que el robot que pretendemos manipular no tiene variables del tipo lineales en cuanto a distancia de avance sino que más bien debe reaccionar autónomamente de acuerdo con el valor de distancia que debe recorrer y la velocidad a la cual debe desplazarse para recorrerla en el menor tiempo sin llegar a colisionar con el extremo del sistema de referencia o el objeto que se interponga en su ruta. Matemáticamente las ecuaciones cinemáticas definen una función que relaciona el espacio Vs. velocidad, definida por:

$$a = (V_f - V_o)/t$$

Figura 4.1. Formula de MRUV. Ley de cinemática. Newton

Donde  $V_o$  es la velocidad que tiene el robot en el momento de tomar la medición de la distancia a recorrer y  $V_f$  es la velocidad que debe adquirir para llegar en el menor tiempo posible,  $a$  es la aceleración producida y  $t$  el tiempo de la variable que se va a calcular.

Teniendo esto en cuenta parecería algo muy complicado de hacer que el robot tenga la capacidad de procesar todos estos datos y ejecutar la acción necesaria sin un razonamiento humano de por medio, sin embargo, la estrategia de implementar un sistema de control difuso basado en el comportamiento de la variable "distancia" es la opción más apropiada y sencilla para poder lograrlo efectivamente.

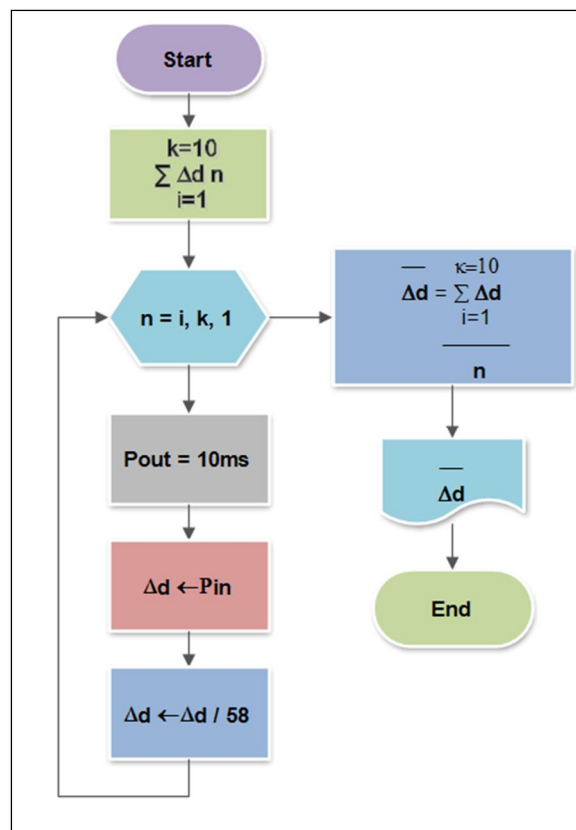
Para implementar esta estrategia se dispone que el robot posea un sensor de distancia que le permita recoger estos datos de forma constante mientras se desplaza y así tomar la decisión más oportuna en cuanto a velocidad de desplazamiento.

Siendo así, se propone el uso de un sensor ultrasónico, más conocido como PING en el ámbito de la robótica, este sensor tiene la capacidad de simular un sonar a través de un algoritmo que le permite medir el tiempo que tarda el sensor en recibir el eco de esta señal transmitida para determinar la distancia a la que se encuentra el obstáculo.

## 4.2. Implementación de Control difuso

### 4.2.1. Medición de distancia del obstáculo

Para esta etapa de desarrollo del proyecto, se utiliza sensores ultrasónicos para que detecten los obstáculos en frente del robot y determinen la distancia a la cual se encuentra mediante una rutina de censado promediado entre 10 muestras realizadas estando en estado detenido.



Flujograma 4.1 Algoritmo de cálculo de distancia promedio.

Fuente: Autor S. Álava

Esta lógica se muestra en la implementación siguiente, realizada en el IDE Arduino usando un sensor ultrasónico HC-SR04, tal cual se lo muestra en la siguiente figura.

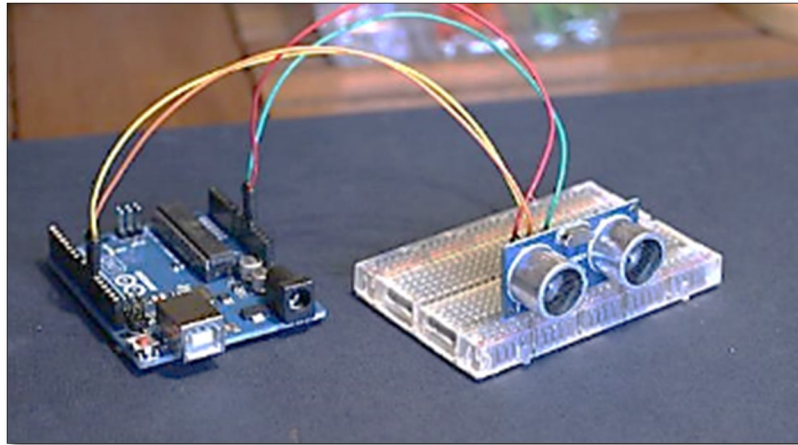


Figura 4.2 Sensor HC-SR04 conectado al módulo Arduino UNO.  
Fuente: Autor S. Álava

Implementación del algoritmo de manejo del sensor HC-SR04 en el IDE Arduino. (Ver [Anexo 4.1](#))

A continuación se realiza una breve investigación sobre el control difuso previo la implementación del algoritmo en el módulo Arduino, teniendo en cuenta que el sensor ultrasónico envía estos datos a la memoria RAM del microcontrolador para ser usadas por el procesador del control difuso. Se propone que el sistema de control planteado conste de una variable de entrada que represente la distancia del objetivo y de una salida que represente la frecuencia de trabajo de los motores, la cual se traduce en velocidad. Cada entrada y salida tiene cinco funciones de membresía. En esta etapa se describe el diseño del sistema de control orientado a la tarea “alalogwrite (Puerto, PWM)”.

#### **4.2.2. Proceso de control difuso según el modelo Takagi y Sugeno**

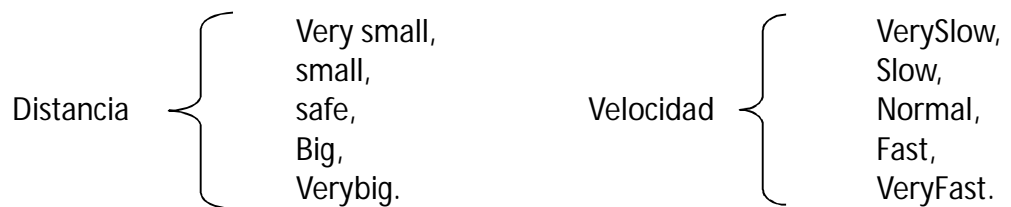
Para el sistema de control difuso se utiliza una estrategia de control aplicando de manera similar los modelos difusos de Takagi y Sugeno. Tales se caracterizan por tener reglas de la forma, en donde cada una relaciona un conjunto difuso definido previamente por antecedentes de la regla en función a las variables de entrada estableciendo un orden (Takagi y Sugeno, 1995).

Los conceptos fundamentales de la lógica difusa son cuatro:

1.- Conjuntos difusos: es el conjunto con fronteras suaves o uniformes, que proyecta valores de membresía.

2.-Variables lingüísticas: en el conjunto difuso se describen las variables cualitativas y cuantitativamente, que es la asociación de términos lingüísticos este expreso conocimiento y concepto; mientras que la función de membresía procesa el dato numérico de entrada.

Por ejemplo, para este proyecto se interpretaran las variables lingüísticas con sus conjuntos de la siguiente forma:



3.-Distribución de posibilidad: son las limitaciones dadas a las variables lingüísticas dentro de un conjunto difuso.

4.-Reglas difusas: IF Them- si entonces, se emplea para describir una proyección funcional, mientras que las reglas de implicación difusa generaliza una relación de implicación entre formulas lógicas de variables y términos lingüísticos.

IF como antecedente describe una condición y THEN es consecuente, Por ejemplo lo empleamos en la siguiente línea del algoritmo. (Duarte, 1998)

IF distance = very small THEN speed = very slow

De la línea anterior se comprende que la variable "distance" distancia si es "very small" muy pequeña entonces "speed" su velocidad será "very slow" muy lenta.

### 4.2.3. Modelamiento del sistema de control difuso

#### Variables de entrada.

Después del proceso de montaje del sensor ultrasónico en un protoboard para realizar pruebas, se analizan las características y el comportamiento del sistema, sometiendo el sistema a la detección de obstáculos a distancias previamente conocidas y se observa las respuestas de los sensores, de los cuales según los resultados se plantea para cada una de las variables de entrada (ubicación y distancia), cinco funciones de membresía.

En la siguiente figuras se grafica la variable lingüística “distancia”  $D(x)$  con sus membresías no lineales.

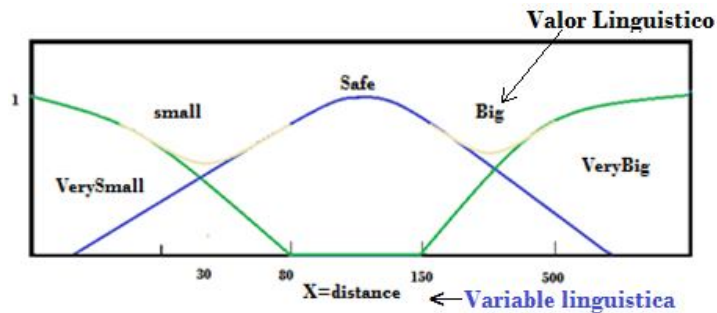


Figura 4.3 Variable lingüística y su conjunto término.  
Fuente: Autor S. Álava

En la siguiente tabla se describe los parámetros para cada variable de entrada, por ejemplo se comprende que la membresía “very small”, descrita como distancia muy corta, se encuentra configurada con una distancia de detección del obstáculo de 5 a 30 cm, así también se describe cada membresía a continuación:

Tabla 4.1 Variables de entrada para lógica difusa aplicada al robot.

Distancia	Configuración (cm)	Descripción
Very Small	(5, 15, 15, 30)	Distancia muy corta
small	(20, 40, 40, 80)	Distancia corta
safe	(70, 100, 100, 200)	Distancia segura
big	(150, 300, 300, 500)	Distancia lejana
verybig	(400, 1000, 1000, 2900)	Distancia muy lejana

Fuente: Autor S. Álava



### **Variables de salida**

Como resultado del análisis del comportamiento del robot en la labor de desplazamiento vrs. evasión de obstáculos, se construyen funciones de membresía para las variables de salida.

Tabla 4.2 Variables de salida para lógica difusa aplicada al robot.

Velocidad	Configuración (PWM)	Descripción
VerySlow	(8, 12, 18, 24)	Velocidad muy lenta
Slow	(24, 64, 96, 112)	Velocidad lenta
Normal	(112, 128, 144, 160)	Velocidad normal
Fast	(160, 192, 208, 224)	Velocidad alta
VeryFast	(224, 240, 250, 255)	Velocidad muy alta

Fuente: Autor S. Álava

### **Reglas de control**

Una vez modelado el sistema de salida se establecen las reglas de control para la navegación del robot, detección de obstáculos y prevención de colisiones:

Tabla 4.33 Reglas de control para lógica difusa aplicada al robot.

IF	THEM
VerySmall	VerySlow
small	Slow
safe	Normal
big	Fast
verybig	VeryFast

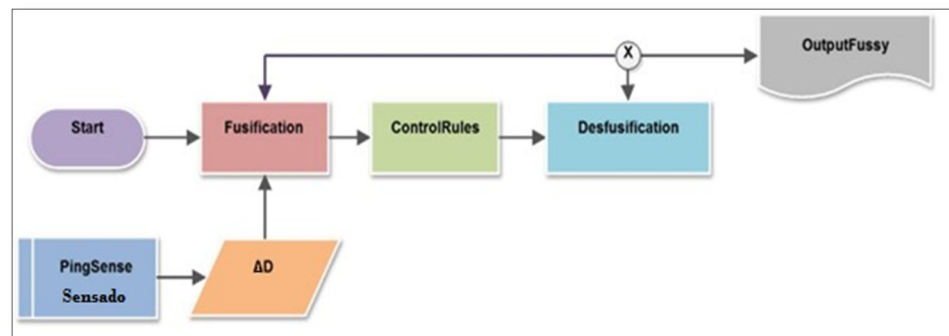
Fuente: Autor S. Álava

En el siguiente algoritmo se describen las etapas de control difuso:

**Fusificación:** Dadas las funciones definidas a las variables lingüísticas de entrada "Distance" es posible reconocer sus conjuntos difusos definidos.

**Reglas de control:** Permitieron establecer las inherencias y limitaciones para cada membresía.

**Desfufificación:** Muestras los resultados de la variable lingüística de salida “ Velocidad” a partir de la variable de entrada “Distance” a través de las reglas de control.



Flujograma 4.2 Algoritmo para Control Difuso.  
Fuente: Autor S. Álava

Una vez estipuladas con claridad las decisiones que debe tomar el robot de forma autónoma de acuerdo al espacio que debe recorrer sin colisionar con obstáculos, entonces se procede a implementar un algoritmo en el IDE Arduino, usando para ello la librería de control difuso disponible en su web.(Ver [Anexo 4.2](#))

### ***Análisis del control difuso***

Desarrollado el control difuso e implementado en el módulo Arduino Uno usando un sensor ultrasónico HC-SR04 se puede determinar que el sensor comprueba la distancia a la cual se encuentra en un determinado momento inclusive estando en movimiento respecto al objeto referenciado y genera las variables de entrada necesarias para que el control difuso implementado realice su trabajo de cómputo y genere una salida en relación a las reglas de control estipuladas. Podemos entonces concluir que ahora el robot ya cuenta con un sistema autónomo de toma de decisiones, que le permite determinar la velocidad a la que debe avanzar de acuerdo a la distancia a la que ha detectado un obstáculo y evitar una coalición con este si está demasiado cerca.

Esta velocidad del robot se traduce en PWM es decir, lo que realmente hará el control difuso es generar una determinada cantidad de pulsos entre un rango de 8 a 255 para enviar este dato a un driver que supla de voltaje y corriente a los motores y estos generen el movimiento en las cuatro ruedas del robot.

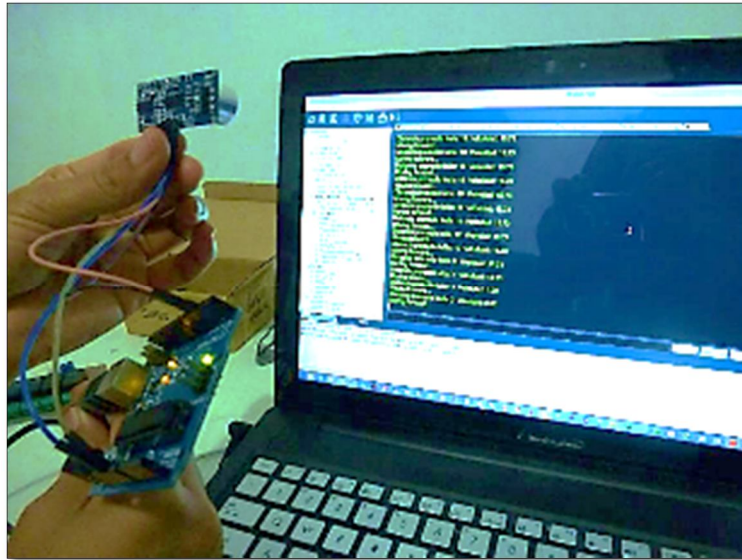


Figura 4.4 Prueba de funcionamiento del control difuso implementado para detección de obstáculos.  
Fuente: Autor S. Álava

### **4.3. Módulo de motores con driver L293d**

En esta etapa del diseño del robot, se considera la utilización de un módulo de control de motores mediante un driver L293 que proporcione la cantidad de voltaje y corriente necesaria para regular la velocidad de giro de cada uno de los 4 motores mediante el control de momento alto (encendido) y el momento bajo (apagado) tal que no se desaproveche la energía que se le suministra. Se procede a continuación a evaluar mediante una prueba de concepto la funcionalidad del módulo de motores de Arduino en contraste de sus características técnicas con el costo de inversión.

#### **4.3.1. Prueba de campo**

El módulo Arduino Motor Shield es originalmente proporcionado por la empresa de diseño electrónico y tienda digital "Adafruit".

Este es un escudo muy útil con componentes estándares que tiene como driver el integrado L293D y que usa 2 registros de desplazamiento 74HC595 para expandir los 3 pines de entrada desde el módulo Arduino a 8 pines y así controlar la dirección de los motores. Las salidas de potencia del L293D están conectadas directamente a las señales PWM generadas por el uC Atmega328.

Para obtener mayor información del código se recomienda visitar la página <http://playground.arduino.cc/Main/AdafruitMotorShield> (Arduino, 2014)

### ***Características***

1. 2 conectores para servos.
2. 4 puentes H: cada chip L293D provee 0.6A por puente (1.2A de pico)
3. Alimentación de motores de 4.5VDC a 36VDC.
4. Control bi-direccional de hasta 4 motores DC, con selección de velocidad individual de 8 bits (resolución de aproximadamente 0.5%).
5. Hasta dos stepper (unipolar o bipolar), de simple bobina o doble bobina.
6. Resistores de pull-down para mantener las salidas desactivadas durante el reset.
7. Bornes robustos para conectar los motores.
8. Conexiones separadas de alimentación de potencia y lógica.
9. Compatible con Arduino Mega 2560, UNO y UNO SMD.

### ***Objetivo***

Evaluar el desempeño de este módulo tipo Shield en contraste de sus características técnicas con el costo de inversión, tal que se pueda acoger esta herramienta para implementar un sistema de control mediante su driver de motores con la capacidad de potenciar la señal PWM generada por el microcontrolador.

### ***Materiales***

- Motor Shield para Arduino de Adafruit.
- IDE Arduino.
- 2 Motores e 1A máximo de consumo a 9V.
- Modulo Arduino UNO.

### **Procedimiento**

Se procede a implementar un código para manejo de los motores basado en el algoritmo original del ejemplo de uso del Shield. (Ver Anexo 4.3)

### **Resultados**

Como resultado de esta prueba se pudo observar que el manejo de los motores en cuanto a dirección y momento es relativamente muy sencillo pero al mismo tiempo lamentablemente el módulo de motores ocupa demasiados pines del Atmega328 por lo que no es útil para nuestro desarrollo.

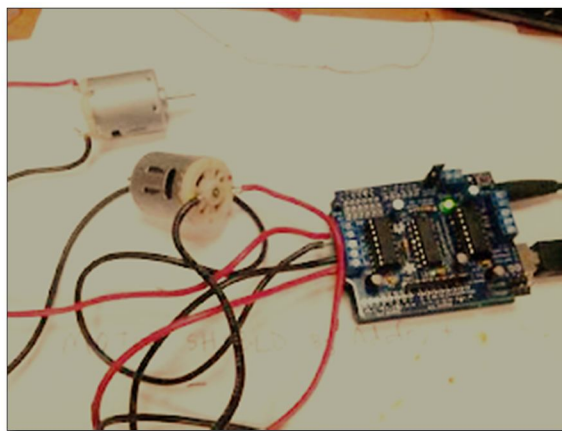


Figura 4.5 Fotografía tomada al módulo de motores de Arduino con el driver L293 sobre un Arduino Uno, conectado a los motores de prueba.  
Fuente: Autor S. Álava

### **Análisis**

El módulo de Motores con el driver L293D no es apto para nuestro proyecto debido a que usa 6 pines de PWM, 4 de manejo de registro y 2 de referencia de voltaje de nuestro módulo, dejándonos apenas 4 bits para el manejo de salidas.

### **Discusión**

Tomando en cuenta la conclusión de la prueba, es recomendable que se elabore un circuito propio para manejar los motores basándose en el circuito de conexiones estándar de un L293D ya que siendo el driver lo único que vamos a utilizar del módulo de motores no es útil para este proyecto.

### 4.3. Diseño de etapa de potencia

#### 4.4.1. Componentes del circuito

En la siguiente tabla se enuncian los elementos utilizados para el circuito de control de motores exportado de la función BOM de Eagle.

Tabla 4.4 Componentes del circuito de control de motores

Part	Value	Device	Package	Description
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	Cap. Polarized
IC1	ATMEGA328-PU	ATMEGA328-PU	DIL28-3	
J1	POWER_JACKPTH	PTH	POWER_JACK	Power Jack
JP1		M03PTH	1X03	Header 3
Q3	16Mhz	CRYSTALHC49US	HC49US	Crystals
R1	10k	R-US_R1206	R1206	Resistor
R2	10k	R-US_R1206	R1206	Resistor
R3	1k	RESISTOR1206	1206	Resistor
R4	1k	RESISTOR1206	1206	Resistor
R5	10k	R-US_R1206	R1206	Resistor

Fuente: Autor S. Álava

#### 4.4.2. Esquemático del circuito

A continuación se detalla los requerimientos del diseño esquemático del circuito que se ha elaborado para controlar la dirección y la cantidad de momento en cada uno de los motores.

1. Control de potencia mediante 1 driver cada 2 motores,
2. Suministro de batería externa para los motores.
3. Suministro de energía desde el modulo Arduino para los driver L293D
4. Uso de elementos SMD para disminuir el tamaño físico del circuito impreso.

Como resultado de estos requerimientos obtenemos el siguiente esquemático:

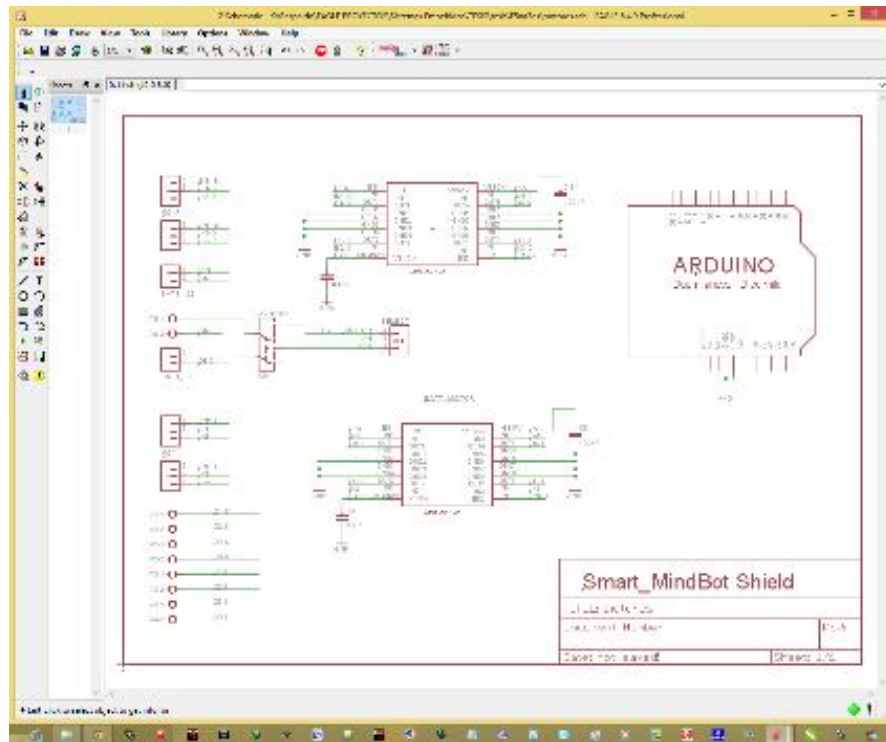


Figura 4.6 Esquema del circuito diseñado para el control de motores.

Fuente: Impresión de pantalla Eagle 6.4.0 Professional.

#### 4.4.3. Tarjeta de circuito ruteada en Eagle

Para esta etapa del diseño se detalla a continuación los requerimientos del diseño electrónico del circuito impreso que se ha elaborado para controlar la dirección y la cantidad de momento en cada uno de los motores.

1. Orificios compatibles con el módulo Arduino Uno y Mega para colocación modular vertical.
2. Dimensiones físicas con un máximo de espacio de 10cm x 7cm para poder usar un diseño modular con los shields Arduino Uno y Mega.
3. Colocación de los elementos de control de cada driver organizada en 2 secciones.
4. Borneras de entrada y salida en los límites periféricos de la placa.
5. Diseño del PCB con una sola capa.
6. Dejar un espacio en el centro para colocar las baterías.
7. Dejar el espacio suficiente para colocar los motores sin que interfieran estos con el circuito.
8. Usar las normas estándar de diseño electrónico indicadas por la norma IPC-A-610D SP sobre montajes y la norma IPC-2221A la cual es el estándar genérico de diseño de tarjetas de circuitos impresos.

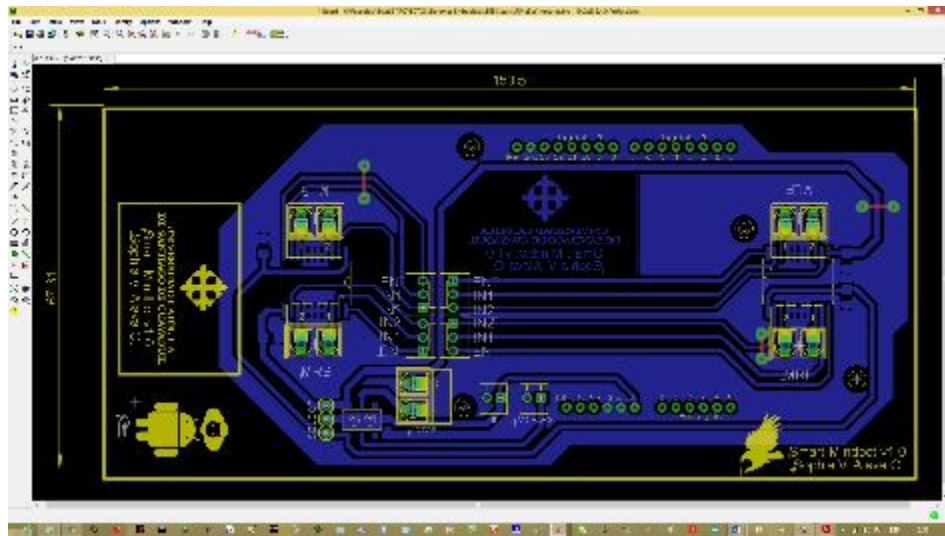


Figura 4.7 Diseño de la PCB de control de motores usando Eagle CadSoft.  
Fuente: Impresión de pantalla Eagle 6.4.0 Professional



## **CAPITULO 5**

### **DISEÑO DE FIRMWARE DEL ROBOT**

En este capítulo se procede a la construcción de un firmware para el microcontrolador Atmega328 que se enlace a la Interfaz Biométrica Neuro-Sensorial mediante su módulo de control de periféricos usando para ello el puerto UART y extraiga de los paquetes de datos la información necesaria para que el robot pueda ejecutar los movimientos ordenados por el piloto.

Para lograr este objetivo debemos primero establecer los umbrales de las variables que vamos a utilizar de tal manera que su rango efectivo actual nos permita calcular de acuerdo a la necesidad del cliente o piloto el mínimo y el máximo de cada una como un valor pertinente y que genere una probabilidad de aceptación en cada orden con un margen de error menor al 10%, tomando en cuenta que habrán variables que no se incluirán como interferencias de dispositivos cercanos y variables ambientales.

#### **5.1. Determinación de umbrales**

Establecer los umbrales de los estados mentales es el punto más importante del presente proyecto ya que con estas variables el robot podrá determinar en qué dirección y cuánto debe avanzar, dejando al control difuso la tarea de decidir a qué velocidad se desplazará. En el proyecto de diseño del dispositivo Interfaz Biométrica Neuro-Sensorial y del módulo de control de periféricos se muestra que mediante una GUI elaborada en el IDE Processing es muy sencillo llegar a determinar estos umbrales ya que se pueden visualizar las medidas de cada una de las variables originarias del TGAM1 en tiempo real.

Por lo que basados en este ejemplo se realiza a continuación un boceto de GUI que nos ayude a interpretar de manera sencilla cuales son los umbrales que debemos fijar para calibrar las variables de entrada del proceso de selección de movimiento del robot de manera que simule su comportamiento como si fuera el robot quien procese la acción pedida.



Figura 5.1 Boceto GUI para visualización de datos EEG usando el API proporcionados por NeuroSky.  
Fuente: (Yaguana, 2014).

### 5.1.1. Diseño de un boceto de GUI

En el apartado Anexo 5.1 se presenta la construcción de una Interfaz Gráfica de Usuario para permitirnos poder definir con claridad y en base a pruebas realizadas en un ambiente real los umbrales que describieron las variables eSense y Blink durante la prueba. (Ver Anexo 5.1).

### 5.1.2. Resultados

Una vez que está construido el código del boceto IBNS\_Mindbot\_Simulator procedemos a ejecutarlo y exportarlo como archivo .bat para su ejecución desde n PC con Windows 8.



Figura 5.2 Reproducción del boceto de Processing creado para determinar los umbrales de los estados mentales Atención, Meditación y del parpadeo de Ojos.  
Fuente: Impresión de pantalla IDE Processing versión 2.1

## 5.2. Prueba al boceto



Figura 5.3 Sujeto de prueba en acción con la GUI desarrollada y enlazada a la IBNS.  
Fuente: Autor S. Álava.

### 5.2.1. Procedimiento

La siguiente tabla describe el procedimiento que se tomó para visualizar las señales EEG que son presentadas en el boceto elaborado por medio de la interfaz biométrica Neuro-Sensorial.

Tabla 5.1 Parámetros del experimento.

Parámetro	Descripción y características
Sensibilidad.	1/Umbral.
Probabilidad de permanencia	50%.
Motivación	El objetivo de pensar en la movilidad del robot que se va a construir.
Tiempo	El tiempo estimado de duración y en general sobre la totalidad del proceso al que se somete el sujeto Autor S. Álava es de aproximadamente 100 minutos, en el cual se realizaran 10 pruebas cada 10 minutos, mientras se observa la GUI y se piensa en la movilidad del Robot.
Sujeto	El sujeto experimental es una mujer de 26 años, sin ningún tipo de deficiencia mental conocida pero si discapacidad cardiaca del 50% según su identificación del CONADIS.
Procedimiento	Siendo que es un experimento del cual el sujeto es el propio desarrollador que lo propone, el objetivo del experimento está por demás decir que es conocido claramente por el sujeto, sin embargo, puede tener un porcentaje de probabilidades de error ya que es subjetivo, es decir, no se tiene un estímulo físico para utilizarlo como referencia sino que más bien el experimento está basado en los estados mentales que el sujeto refleja al pensar en la movilidad del robot.

Fuente: Autor S. Álava

### 5.2.2. Análisis

La siguiente tabla muestra los resultados obtenidos por las variables de atención, meditación y parpadeo.

Tabla 5.2 Resultados recolectados de las pruebas de experimentación

Variables	Umbral Mínimo	Umbral Máximo	Normalización estándar eSense	Rango de aceptación generados por la investigación referencial
Atención	68	96	60	60
Meditación	73	100	60	80
Intensidad de parpadeo en ms	235	412	-	150-350

Fuente: Autor S. Álava

### 5.2.3. Discusión

Los valores obtenidos mediante la GUI muestran la similitud de los rangos normalizados con los generados por la investigación referenciada en cuanto a sus recomendaciones técnicas salvo la variable de Blink, se presume que esta variable es la que más es afectada por la condición cardiaca del sujeto, sin embargo está dentro del rango propuesto y genera más del 50% de probabilidades de acertar con la generación de un Evento Blink cada vez que se realice un parpadeo.

Por lo cual, es tomado como umbral de referencia el mismo dato, solo en caso de que las probabilidades en las pruebas de campo hayan disminuido será modificado el parámetro de tiempo en el algoritmo principal del módulo de control de periféricos.

### 5.3. Parámetros del diseño del Firmware

Una vez determinados los umbrales podemos proceder a construir un algoritmo que nos permita manipular el robot para lo cual se tomará en cuenta los resultados que arrojó la investigación abordada en el anterior capítulo sobre movilidad, dirección y evasión de obstáculos, así mismo, se utilizará la misma lógica desarrollada en el boceto GUI del simulador para construir las funciones de pertenencia del robot.

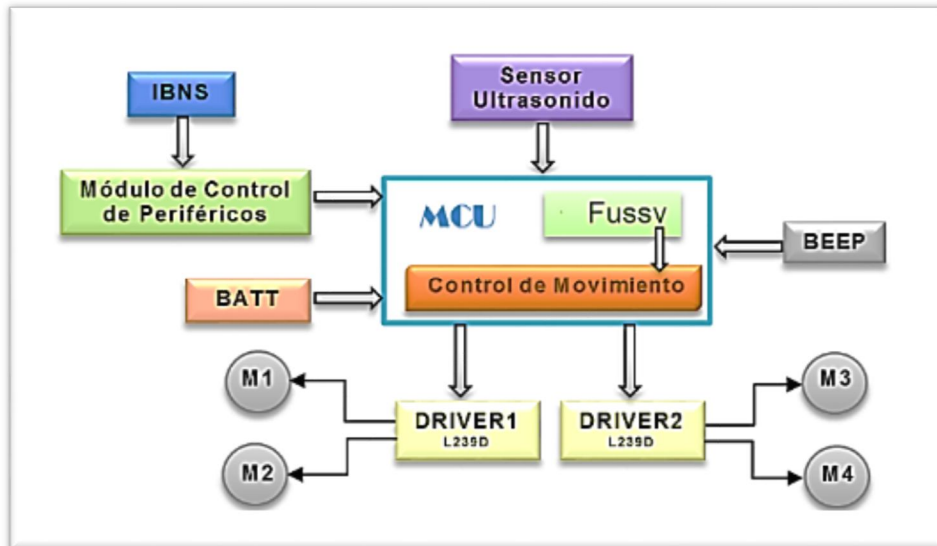
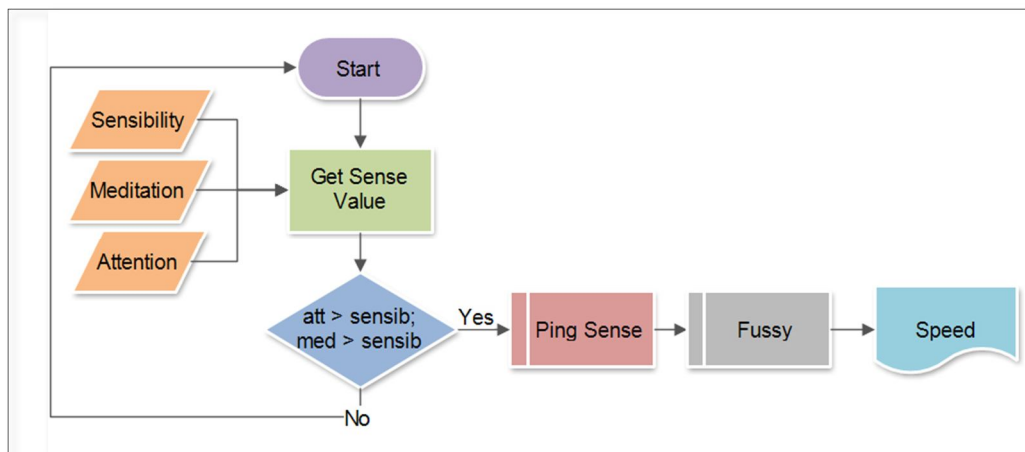
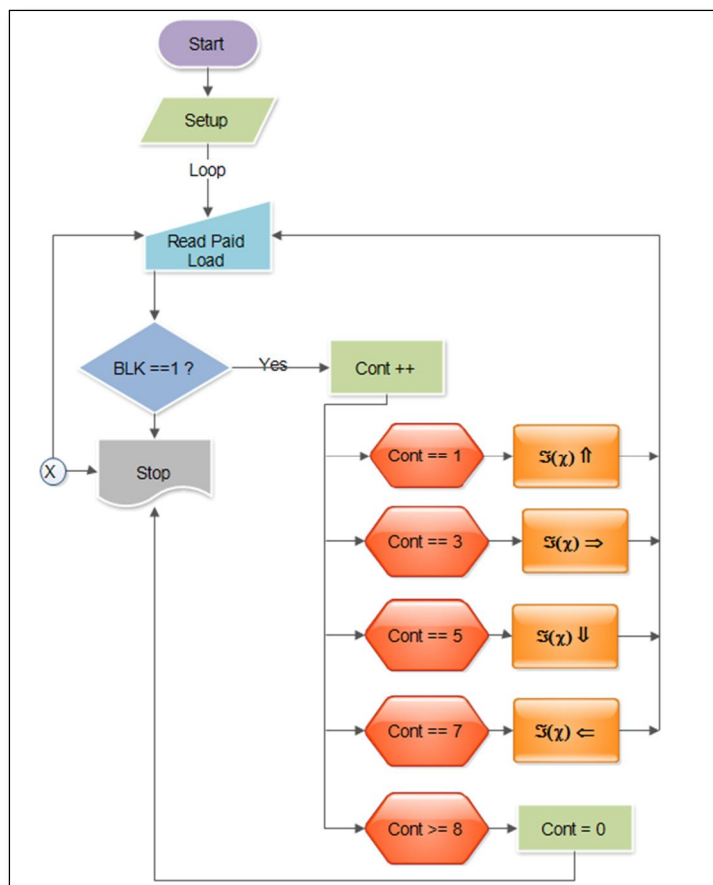


Figura 5.4 Bloque Funcional del Rover Smart MindBot.  
Fuente: Autor S. Álava

A continuación procedemos a la construcción del firmware del robot en el IDE Arduino, al cual de ahora en adelante se lo referenciará como Smart MindBot dada su lógica de funcionamiento y su control por medio de estados mentales desde el sujeto piloto. (Ver [Anexo 5.2](#)). Este código se referenció en el siguiente algoritmo diseñado para el control del robot mediante el evento Blink, y es parcialmente el mismo que se usó para diseñar la lógica de control en el entorno de simulación realizado en Processing.



Flujograma 5.1 Algoritmo de control final usando los procesos Ping y Fussy.  
Fuente: Autor S. Álava.



Flujograma 5.2 Lógica de control basada en umbrales y eventos.  
Fuente: Autor S. Álava.

Como se puede observar la lógica es muy simple, hay un contador que de acuerdo al número de parpadeos determina la dirección que debe tomar el robot y cada 2 parpadeos con la suficiente intensidad para sobrepasar el umbral fijado en el módulo de control de periféricos es un llamado a la función "stop", es decir, mientras el robot no detecte un parpadeo el seguirá en el estado presente, ya sea que este detenido o en movimiento, pero, si está en movimiento, se comenzará a mover únicamente si el umbral fijado ya sea de meditación como en este caso, es superado. Una vez ya en movimiento el robot, calculará la distancia a la cual hay un obstáculo y generará autónomamente la cantidad de PWM necesaria para avanzar sin llegar a colisionar.

Con este diseño del firmware y la lógica que le hará operar al piloto el robot se concluye el presente capítulo y se procede a la elaboración física del robot, la cual será abordada en el siguiente capítulo teniendo en cuenta los avances investigativos que se han generado hasta el momento dentro de este proyecto.

## **CAPITULO 6**

### **DISEÑO DEL HARDWARE DEL ROVER SMART MINDBOT**

En este último capítulo de esta investigación se realiza una breve introducción de las áreas competentes en el diseño en general.

#### **6.1. Morfología del robot**

Existen varios tipos de estructuras y morfologías que cumplen con su tarea respectiva de acuerdo al entorno para las cuales fueron diseñadas, cada una de estas tiene marcadas características físicas que determinan la utilidad y flexibilidad del robot dentro de las limitantes del diseño mecánico y la capacidad de su nivel de control.

De entre todas éstas, las características que más favorecen al desarrollo de esta investigación son las de un robot de morfología tipo Rover debido a su capacidad de movilidad, el diseño basado en una plataforma de ruedas y su estabilidad la cual permite mantener fijo el centro de gravedad del vehículo en triangulación con los puntos que tocan la superficie de referencia, es decir, el terreno sobre el que se desplaza. Sin embargo, las ruedas sólo pueden utilizarse sobre terrenos relativamente sólidos y lisos, si por lo contrario se quiere usar en terrenos rugosos se debe considerar el tamaño de las ruedas y su labrado tal cual sucede con los automóviles y los vehículos pesados de construcción.

En este caso se pretende usar el modelo de cuatro ruedas y tracción diferencial para restringir la capacidad de movimiento a ordenes mentales del usuario que no requieran un gran esfuerzo ya que será un prototipo en el marco conceptual de las posibles aplicaciones que se pueden generar a partir del grado de factibilidad en proyectos con similar o mayor dificultad obtenidos con este sujeto de investigación.



El arreglo más familiar de la morfología Rover es como se mencionó el utilizado por los vehículos de transporte el cual contempla cuatro ruedas colocadas en las esquinas de un rectángulo. La mayoría de estos vehículos tiene movilidad limitada debido a que su tracción se ejerce en dos pares de ruedas mediante distribución diferencial de torque y necesitan avanzar una gran cantidad de terreno para poder girar o dar vuelta. Además de estas limitaciones, se requiere de un sistema de suspensión para asegurar que las ruedas estén en contacto continuo con la superficie todo el tiempo o al menos la mayor cantidad posible caso contrario pierde fuerza la tracción y es posible que el vehículo quede eventualmente detenido o deshabilitado por las condiciones físicas del terreno.

Estas características, ventajas y desventajas antes mencionadas pueden ser evaluadas con mayor facilidad si el robot es de morfología Rover.

### 6.1.1. *Electrónica*

El siguiente esquema muestra la electrónica del robot a nivel de bloques funcionales.

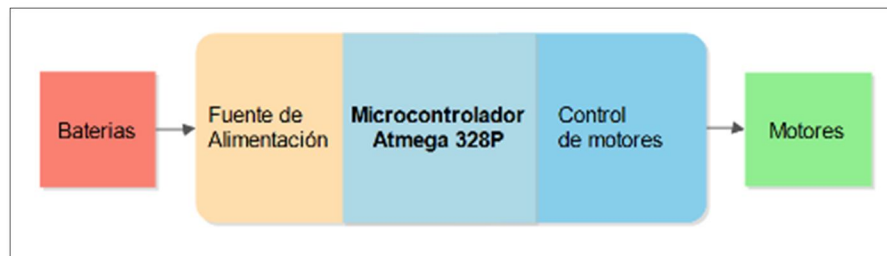


Figura 6.1 Bloque funcional de la electrónica del robot

Fuente: Autor S. Álava

#### **Descripciones de las partes del bloque funcional:**

Fuente de alimentación de control. Su función será regular la tensión de entrada de la batería, adaptándola a los diferentes niveles de tensión de los elementos de control del robot. La fuente deberá ser capaz de entregar una tensión de trabajo estable y fiable en todo momento al alimentar dispositivos delicados, tales como el microcontrolador, circuitos digitales, operacionales y sensores.

Microcontrolador. Es el cerebro de todo el sistema, su función es controlar los periféricos conectados a él. Por lo cual se utiliza el microcontrolador ATMEGA328-P caracterizado por su procesador de 8-Bit en 28 pines

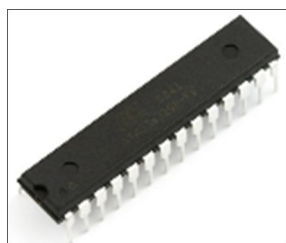


Figura 6.2    Atemga 328p  
Fuente:        (Atmel, 2014)

Baterías. Dotaran de una fuente de energía al robot de tensión nominal de 8.4 Voltios, y capacidad: 200 mAh, de manera que omitamos cableado extendido para energizarlo. A nivel de corriente, el usar una batería con más o menos amperaje por hora supondrá una autonomía mayor o menor.

Control de motores. Si bien se contempla que el microcontrolador, es el encargado del control, necesitamos un módulo que sea capaz de gobernar las tensiones y corrientes consumidas por los motores.

Motores. Básicamente son los componentes electro-mecánicos que harán moverse al robot. En este caso se utiliza un arreglo de 4 motores, agrupados en pares paralelos.

### **6.1.2. Telecomunicaciones**

Se compone de dos módulos detallados a continuación:

**Módulo de comunicación del robot:** Es el encargado de establecer los enlaces con la Interfaz Biométrica Neuro-Sensorial del sujeto que hará las veces de piloto, ya sea de forma cableada o inalámbrica.

En cuanto al sistema inalámbrico se ha optado por un módulo transceptor de radiofrecuencia a 2.4GHz, con modulación FSK y velocidad de transmisión de 57600 baudios utilizando para ello un Bluetooth Smirf que es el mismo módulo que nos provee la Interfaz Biométrica Neuro-Sensorial con el módulo de control de periféricos. Para el enlace utilizamos el protocolo RS232, con 8 bits de datos, 1 bit de stop, sin paridad y a una velocidad de 57600 baudios, y la configuración del modelo Maestro Esclavo del Bluetooth RN- 41.



Figura 6.3 Módulo Bluetooth RN- 41  
Fuente: (Pololu, RN-41 Bluetooth Module, 2014)

**Módulo de comunicación del ordenador:** Las funciones del módulo de comunicación del robot son las mismas, lo único que varía es que se trata del otro extremo del enlace. Los niveles de tensión, velocidades de transmisión, frecuencias para la comunicación inalámbrica, etc., deberán ser igualmente compatibles con la del módulo del robot quien usará protocolos específicos para esta función, los mismos que serán elegidos de acuerdo a su funcionalidad y flexibilidad, más aun si no lográsemos adaptarnos a uno de ellos, se investigaría la creación de un propio protocolo de comunicación.

### 6.1.3. *Mecánica*

Existe multitud de materiales que se pueden utilizar en la construcción del robot, ya sea para la estructura del chasis que ha de soportar los motores, placas, baterías, etc.; como para las ruedas y la superficie en contacto con el suelo. Entre los materiales más usuales se encuentran: Aluminio, Acero, Madera, Plástico, Fibras, etc.

### ***Elección de materiales para los elementos del robot:***

**Chasis:** para el chasis se estudió el optar por una estructura a base de aluminio. Los soportes laterales de aluminio se encargarán de la sujeción de las ruedas con el conjunto motor + reductora.

**Ruedas:** para el prototipo las ruedas se harán en primera instancia en material caucho negro ya que se pretende realizar únicamente una prueba de campo. 0

En consecuencia, se adquiere dos juegos de neumáticos Tamiya que incluyen dos ruedas diseñadas para superficies ásperas y desiguales con diámetro de 65 mm y 26 mm de anchura, compatible con la caja de engranajes utilizados.



Figura 6.4 Tamiya 70194 Spike Tire Set (2 tires)

Fuente: (Pololu, Tamiya 70194 Spike Tire Set (2 tires) , 2014)

Dimensionado de la base del robot: para realizar el dimensionado de los elementos electro-mecánicos que conforman la plataforma en movimiento del robot, debemos establecer las siguientes especificaciones:

- Velocidad del robot: Se puede calcular la velocidad del eje de salida con la velocidad actual por la relación de transmisión utilizada 344:1.
- Motores y reductoras: por simplicidad, decidiremos buscar el conjunto motor + reductora del mismo fabricante, ya que nos evitaremos tener que fabricar nuestros propios engranajes y adaptar ejes de distinto diámetro.

- Características del motor + reductora: por tratarse de un robot móvil autónomo, hay ciertas restricciones que debemos tener en cuenta y que nos acotaran la elección del conjunto. Estas limitaciones son:
  - Debe poder ser alimentado desde una batería
  - La circuitería de control deberá ser relativamente sencilla
  - Disponer de buena relación dimensiones – rendimiento/respuesta

### **Elección del material:**

Se adquiere dos conjuntos de doble motor de Tamiya con reductora de corriente continua, es una unidad compacta con trenes de engranaje, con relación de transmisión de 344:1.



Figura 6.5 Dual motor Gearbox de Tamiya  
Fuente: (Pololu, Tamiya 70168 Double Gearbox Kit , 2014)

Diseño de la base: antes de construir la base del robot habrá que realizar los planos de los elementos por separado. Tras el ensamblado de todos los elementos individuales en el diseño final, podremos tener una visión global del acabado para hacernos una idea de la geometría.

#### **6.1.4. Software y comunicaciones**

Antes de establecer qué protocolo o conjunto de ellos vamos a utilizar en las distintas comunicaciones de nuestro robot, hay que determinar qué componentes se van a comunicar, requerimientos de velocidad, optimización, etc.

A continuación se enumeran los elementos que llevan a cabo procesos comunicativos ya sea en el PC o el robot.

1. Control del robot: dentro del enlace robot-PC distinguiremos entre la comunicación de control de movimiento y monitorización de sensores.
2. Protocolos de comunicaciones disponibles por Tecnologías: dentro de las tecnologías comerciales actuales que pueden adaptarse a nuestras necesidades; teniendo en cuenta que debe ser posible una transmisión inalámbrica, podemos encontrar los enlaces por radiofrecuencia y los enlaces por infrarrojos.

## **6.2. Construcción del robot**

### **6.2.1. Elaboración de los PCBs**

La elaboración de los PCBs se procede a realizar con materiales de laboratorio para poder hacer de pruebas de su funcionalidad previa la solicitud de fabricación de tipo profesional de las mismas a la empresa de fabricación de circuitos impresos APM S.A.

Se procede a realizar la elaboración del PCB, esta información se ha adjuntado en anexos a manera de instrucción del proceso de fabricación de los PCB en un laboratorio electrónico pero usando materiales de fácil adquisición (Ver [Anexo 6.1](#)).

### **6.2.2. Construcción de la morfología Rover del robot**

Posteriormente, una vez concluido el proceso de fabricación, se puede ya utilizar las PCB e incluirlas en la estructura modular del robot para poder continuar con las pruebas de campo pertinentes. (Ver Anexo 6.2).

### 6.2.3. Pruebas de comunicación

Una vez realizado con éxito la construcción del Rover, se procedió a la carga del bootloader del Atmega 328 para poder programarlo mediante la plataforma Arduino, realizado esto se obtuvieron los siguientes resultados.

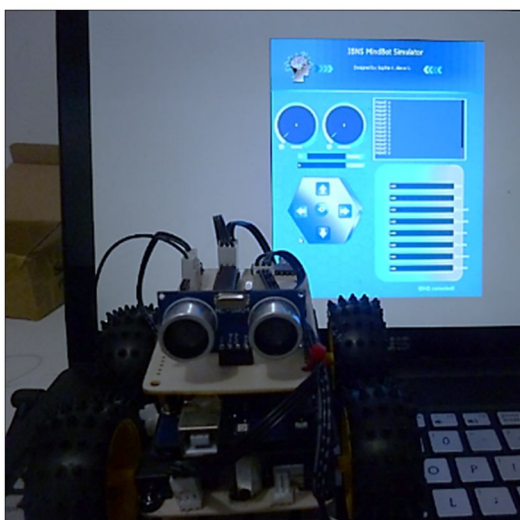


Figura 6.6 Pruebas de comunicación del Robot MindBot Smart Rover con la PC usando la GUI diseñada para simular su control.

Fuente: Autor S. Álava

### 6.2.4. Análisis

La comunicación con el robot se realizó sin mayor complicación mostrando una alta tasa de efectividad superior a la esperada gracias a la ayuda de la IBNS y del módulo de control de periféricos los cuales proporcionaron una gran ayuda para la elaboración de este proyecto.

Cabe resaltar que el robot pudo realizar todas las acciones que el sujeto "Piloto" le solicita mediante la secuencia de registros que se le envía por medio de la GUI de simulación y control, pero, debido a la frecuencia cardiaca del piloto la cual es levemente alterada por la condición médica, debió ser modificado el evento Blink según mostró las pruebas acerca de la obtención de los umbrales, los parámetros que se necesitaría para que funcione efectivamente, lo cual permitió observar un incremento de efectividad en el manejo del robot.

Sin embargo, el filtrado que se realiza a la señal en el módulo de control de periféricos debe ser bien comprendido para eventualmente poder cambiar los valores de los parámetros de configuración del control de periféricos ya que sin este conocimiento sería prácticamente jugar al azar en cuanto al control usando el evento Blink Eyes y su intensidad.

### 6.3. Pruebas de campo del robot MindBot Smart Rover

Debido a la condición que implica hacer una prueba de campo y los factores que pueden afectar al robot durante la prueba, su ejecución debe ser en un ambiente controlado de tal forma que se pueda cuantificar adecuadamente el entendimiento de todo el entorno de la realiza en la cual se ejecuta, siendo así, se propone los siguientes ambientes controlados:

1. Desplazamiento en una superficie plana para eliminar la variable de irregularidad del terreno.
2. Para sujeto de demostración: desplazamiento sobre una superficie irregular con la misma estructura sin usar suspensión para amortiguar momentos de fuerza en sus ejes de movimiento



Figura 6.7 Prueba de campo realizada con la IBNS y el Robot en comunicación directa entre el sujeto de prueba y la interfaz de control.

Fuente: Autor S. Álava



## CONCLUSIONES

1. Se pudo constatar que las pruebas funcionales y de campo fueron de mucha ayuda para establecer los parámetros de implementación del robot, a modo que cada una de estos proyectó resultados con ventajas y desventajas claras del funcionamiento de cada dispositivo probado.
2. Por ejemplo al hacer pruebas con el módulo de motores Arduino se pudo descartar su utilidad ya que ocupaba muchos pines por lo cual no proporcionaba las características técnicas adecuadas que se buscaba pero fue posible adaptar su esquema y crear un módulo de control de motores que se ajuste a las necesidades de manipulación del robot.
3. Por otro lado, el firmware diseñado, debido a su lógica de control y las estrategias implementadas puede servir como base de diseño de una endoprótesis inteligente que otorgue movilidad a una persona mediante sus estados mentales o inclusive serviría de referencia en la creación de equipos de movilidad y transporte como por ejemplo sillas de ruedas que prácticamente usarían el mismo algoritmo sin alteraciones mayores.
4. La estrategia de control implementada, permitió trabajar fácilmente con sistemas de variables de entrada y salida de acuerdo a la adaptación del medio en el que el robot se desempeña, los cuales sin el uso del control difuso generarían un modelo de bastante complejidad debido a sus variables no lineales y no podrían ser tratados mediante técnicas de control clásico.
5. Para la elaboración del prototipo de hardware de morfología Rover se utilizó el diseño modular de los PCB's electrónicos como estructura mecánica de su carrocería de tal manera que el cuerpo de suspensión se apoya en el diseño del PCB de la etapa de potencia de los motores, haciendo su conexión más sencilla sin abundar en términos de dinámica y reacción de fuerzas, como hubiese sido utilizando una suspensión elástica o con amortiguación y puesto que se necesita probar la movilidad del robot según su diseño del algoritmo, más no su capacidad de desplazamiento sobre superficies irregulares.

6. Así mismo, la lógica diseñada para usar como evento de control de cambio de dirección del robot usando la intensidad del parpadeo de los ojos del sujeto, evidencian también un alto desempeño de este algoritmo ya que solo hay una variable considera que permite el desplazamiento de bits y esto permite que no solo se pueda hacer un control efectivo de cinco estados como en este proyecto sino de cuantos uno requiera, claro está que en cuantos más haya más tiempo habrá entre cada ciclo de elección.
7. La comunicación con el robot se realizó finalmente con una mejora no contemplada debido a la capacidad de modificación de los umbrales citados para el manejo del robot, de acuerdo a las condiciones físicas y emocionales del cada piloto, es decir se dio al robot la capacidad de adaptarse a los parámetros de sensibilidad distintivos y únicos de cada persona.

## **RECOMENDACIONES**

Se recomienda utilizar la trama de datos de la BCI ya que es de fácil comprensión.

Al conectar la BCI al computador para visualizar los datos que entrega al terminal serial mediante hércules configurar en hexadecimal para comprar los registros.

Se recomienda el uso de la estrategia de control Fussy empleada en este proyecto para facilitar el manejo de variables recogidas por los sensores en desarrollos de equipos y prótesis biomecánicas que impliquen complejas lecturas biométricas.

En el proceso de atacado químicos a la placas PCb's utilizar la debida protección ya que es una sustancia nocivo

Utilizar robot ya fabricados de doble tracción si es complejo el proceso realizado

Es necesario implementar un plan de gestión, que permita promover una cultura de I+D para mejorar la calidad de vida de las personas mediante desarrollos e implementaciones tecnológicas que incursionen en campos relacionados con interfaces biométricas, generando soluciones a usuarios con capacidades limitadas, haciendo uso de recursos como el ofrecido mediante este proyecto y que así se cree una sociedad del conocimiento más avanzada en nuestra institución.

## GLOSARIO

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

**Address:** Es la dirección de referencia o identificación del objeto, conformada por caracteres alfanúmericos que representan el fabricante, proveedor e identificación del recurso único, como Email Address, Mac Address, etc.

**Algoritmo:** Conjunto de reglas específicas para la solución de un problema en un número finito de pasos.

**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.

**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) o Código Estándar para el Intercambio de la Información conformado por 95 caracteres donde se incluyen símbolos, letras y números.

**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).

**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.

**Base:** Es la plataforma o estructura en la cual el brazo del robot está sujeto.

**Bat:** Un tipo batch es un archivo de texto que contiene comandos a ejecutar en un intérprete de comandos propio de DOS o OS/2. Cuando es iniciado, un programa shell típicamente command.com o cmd.exe lo lee y ejecuta, generalmente línea a línea. De este modo, se emplea para ejecutar series de comandos automáticamente. El hecho de que funcione sólo para MS-DOS lo hace muy limitado. Su extensión es .bat o .cmd.

**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.

**BCM:** Sigla en inglés de Brain Computer Machine, interfaz que enlaza el cerebro con una máquina. Ej.: Robot

**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.

**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.

**Brazo robótico:** Es el conjunto interconectado de eslabones y articulaciones que constituyen a un manipulador el cual soporta o mueve una muñeca, consiste en un sistema de articulaciones mecánicas y sensores de posición usados en el sistema de control de bucle cerrado.

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

**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.

**Cinemática:** Es el estudio del movimiento sin importar las fuerzas y momentos que lo producen

**Comando:** Son aquellas palabras reservadas para la sintaxis de un lenguaje de programación o de comunicación.

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

**CONADIS:** El Consejo Nacional de Discapacidades. Es un organismo autónomo de carácter público, creado en agosto de 1992, a través de la Ley 180 sobre Discapacidades.

**Control:** Es el proceso de regulación manual o automática sobre una variable o sistema de variables conforme a lo que se desea.

**Controlador:** Dispositivo con el que cuenta el robot para el manejo de circuitos encargados del movimiento eléctrico. Generalmente basado en un microcomputador, que percibe las señales de los sensores de posición y envía comandos a la fuente de potencia controlada.

**ControlIP5:** librería grafica de Processing

**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.

**CNBI:** Center for Neuroprosthetics and Institute of Bioengineering.

**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.

**DB9:** El conector DB9 (originalmente DE-9) es un conector analógico de 9 clavijas de la familia de conectores D-Subminiature (D-Sub o Sub-D). El conector DB9 se utiliza principalmente para conexiones en serie, ya que permite una transmisión asíncrona de datos según lo establecido en la norma RS-232 (RS-232C).

**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:** Es un esquema con las partes electrónicas de los elementos que forman un circuito o presentación.

**DOI:** (Digital Object Identifier) es la identificación de objetos en el entorno digital que, a diferencia de la URL, una vez es asignada, por ejemplo, a un artículo, no cambia a través del tiempo.

**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.

**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).

**Eje:** Cada una de las dos o más líneas que sirven para determinar la posición de los puntos de espacio.

**Electrodos:** En términos de física y química es todo conductor eléctrico que permite el paso e ingreso de la corriente.

**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.

**Et al.:** Locución latina para "Y el resto"

**EPFL:** l'Ecole Polytechnique de Lausanne, institución dedicada al estudio de las Neurociencias.

**Firmware:** Es el conjunto de instrucciones primarias creadas para determinar la funcionalidad estándar de un sistema.

**Fluxograma:** Gráfica que muestra el flujo y número de operaciones secuenciales de un proceso o procedimiento para generar un bien o un servicio. Pertenece a la ingeniería de sistemas y también se le conoce como algoritmo, lógica o diagrama de flujo. La ingeniería industrial emplea otro diagrama conocido como de "proceso, recorrido u hoja de ruta" con una simbología diferente a la de sistemas. Al igual que el primero, se aplica al diseño de procesos y procedimientos.

**FSK:** Modulación digital por desplazamiento de frecuencia, o en inglés Frequency Shift Keying, técnica en donde la señal modulada enviada por flujos de uno y ceros es alterada por niveles de voltaje discretos.

**FP1:** Es un punto localizado en la frente de una persona y que es parte del sistema de localización de referencia de electrodos denominado 10-20 por la huella de cobertura que ofrece cada uno en un examen típico de EEG.

**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.

**Grado de libertad:** Son los posibles movimientos básicos independientes, ya sean giratorios o de desplazamiento, que el robot puede realizar.



**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:** (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

**Interfaz:** En términos informáticos es un medio de comunicación, conexión física o funcional entre sistemas o dispositivos.

**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

**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.

**Led:** Light Emitting Diode, (Diodo emisor de luz)

**Lóbulo:** Cada parte redondeada que a manera de ondas sobresalen de algo. Aplicado a la anatomía es la parte blanda de la corteza terrestre ubicada en el inferior de la oreja.

**mAh:** Medida de carga eléctrica Miliamperio-Hora.

**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.

**MindBot:** Término acuñado en el año 2011 por el Dr. Timothy E. Laird durante su proyecto de graduación de Clemson University en el proyecto "EEG/EMG Controlled Robot via Wireless Neural Headset" en la cual un rover de Parallax de cuatro ejes es

controlado mediante el dispositivo Emotiv "EPOC headset" conectado a un PC el cual decodifica las neuroseñales de su piloto y envía comandos para que este se mueva.

**Modelo:** En términos informáticos se relaciona al conjunto de variables que conducen un resultado.

**Módulo:** Es el conjunto de objetos que cumplen una determinada función.

**MRUV:** El movimiento rectilíneo uniformemente acelerado (MRUA), también conocido como movimiento rectilíneo uniformemente variado (MRUV), es aquel en el que un móvil se desplaza sobre una trayectoria recta estando sometido a una aceleración constante.

**NASA:** La Administración Nacional de la Aeronáutica y del Espacio, más conocida como NASA (por sus siglas en inglés: National Aeronautics and Space Administration), es la agencia del gobierno estadounidense responsable del programa espacial civil, además de la investigación aeronáutica y astronáutica en ese país.

**Neurociencias:** Es la ciencia que estudia el sistema nervioso del ser humano y animal.

**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.

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

**OMPI: Organización Mundial de la Propiedad Intelectual,** es un organismo especializado del Sistema de Naciones Unidas, creado en 1967 con la firma del Convenio de Estocolmo. La OMPI está dedicada a fomentar el uso y la protección de las obras del intelecto humano.

**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:** Expresión utilizada para libre distribución de la información establecida.

**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.

**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

**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"

**PID:** Estrategia de control que se implementa mediante una variable con un punto de referencia estático y constante al que debe llegar mediante un proceso llamado "Proporcional-integrativo-derivativo".

**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 patica del conector tiene un propósito que se describe brevemente en el pinout.

**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:** Término utilizado para implementaciones o investigaciones resumidas con propósito de demostrar o cuestionar de manera más útil los conceptos conocidos. El diccionario Oxford cita su primer uso del periódico Los Angeles Times, 1967.

**PSQ:** (Poor Quality Signal) En informática, es un indicador de la calidad de la señal de un periférico o de un puerto analógico mientras se realiza el muestreo y adquisición de los datos.

**PWM:** (pulse-width modulation) Es la modulación por ancho de pulsos de una señal o fuente de energía. Es una técnica en la que se modifica el ciclo de trabajo de una señal periódica (una senoidal o una cuadrada, por ejemplo), ya sea para transmitir información a través de un canal de comunicaciones o para controlar la cantidad de energía que se envía a una carga.

**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).

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

**ROM:** (read-only memory), memoria de almacenamiento de solo lectura..

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

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

**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.

**SMD:** (Surface Mount Technology) es el método de construcción de dispositivos electrónicos más utilizado actualmente. Se usa tanto para componentes activos como pasivos, y se basa en el montaje de los mismos sobre la superficie del circuito **impreso**.

**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".

**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.

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

**Trayectoria:** Está formada de puntos que realiza el robot o pasa a través de una operación dependiendo de la programación. Es una relación temporal de posición velocidad y aceleración para cada uno de los grados de libertad del robot.

**TTL:** (transistor-transistor logic) Es una familia lógica o lo que es lo mismo, una tecnología de construcción de circuitos electrónicos digitales, en la cual sus niveles de voltaje van de 0 a 5 voltios.

**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

**Usuario:** Llámeme así a toda presentación de un objeto, persona, o conjunto de recursos a los que se tiene acceso.

**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".

## REFERENCIAS BIBLIOGRÁFICAS

Arduino. (2014). *Adafruit Motor Shield*. Recuperado el 10 de Enero de 2014, de <http://playground.arduino.cc/Main/AdafruitMotorShield>

Arena, S. Z. (2008). *Metodología de la Investigación*. Recuperado el 24 de 10 de 2013, de <http://metodologiadelainvestigacion.forumcommunity.net/?t=14325119>.

Atmel. (2014). *ATmega328P*. Recuperado el 14 de Marzo de 2014, de <http://www.atmel.com/devices/atmega328p.aspx>

Duarte, O. (5 de Enero de 1998). *Sistemas de lógica difusa*. Recuperado el 5 de Mayo de 2014, de [http://www.ing.unal.edu.co/~ogduarte/Htextos/maestro/node20\\_mn.html](http://www.ing.unal.edu.co/~ogduarte/Htextos/maestro/node20_mn.html)

Emotiv. (2004). *emotiv*. Recuperado el 2 de Febrero de 2014, de <http://www.emotiv.com/about/>

Garcia, R. (2010). *Investigación de Neuro Tecnologías Innovadoras Aplicadas Comercialmente*. Universidad Católica de Santiago de Guayaquil, Sistema de Investigación y Desarrollo. Guayaquil: Sinde-UCSG. Recuperado el 25 de Enero de 2014

Granada. (2005). *Actividad Patológica*. Recuperado el 20 de Marzo de 2014, de <http://neurofisiologiagranada.com/eeg/eeg-activpatologica.htm>

Granada. (2005). *La Actividad Cerebral*. Recuperado el 20 de Marzo de 2014, de <http://neurofisiologiagranada.com/eeg/eeg-actividadcerebral.htm>



- H. García, S. A. (2009). Detección y Clasificación de Artefactos. *XIV SIMPOSIO DE TRATAMIENTO DE SEÑALES, IMAGENES Y VISIÓN ARTIFICIAL*, 8. Recuperado el 15 de Enero de 2014, de [http://www.uniquindio.edu.co/uniquindio/revistadyp/backup20121214/archivo/edicion\\_4/Articulos/Articulo%20Stsiva%20final.pdf](http://www.uniquindio.edu.co/uniquindio/revistadyp/backup20121214/archivo/edicion_4/Articulos/Articulo%20Stsiva%20final.pdf)
- Merchán, D. G. (15 de Diciembre de 2008). *Introducción a los sistemas Brain Computer Interface*. Recuperado el 1 de Marzo de 2014, de <http://www.lacofa.es/index.php/general/introduccion-a-los-sistemas-brain-computer-interface>
- Millán, J. d. (6 de Febrero de 2011). Curiosidad y trabajo en equipo. (L. M. Ariza, Ed.) *REPORTAJE: JOSÉ DEL ROCÍO MILLÁN*. Recuperado el 25 de Marzo de 2014, de [http://elpais.com/diario/2011/02/06/eps/1296977214\\_850215.html](http://elpais.com/diario/2011/02/06/eps/1296977214_850215.html)
- Ministerio de Educación, P. d. (2012). Introducción a la Robótica. *Robótica*, 36. Recuperado el 14 de Marzo de 2014
- Navarro, A. (25 de Septiembre de 2009). *El cerebro emocional*. Recuperado el 4 de Mayo de 2014, de <http://www.asociacioneducar.com/monografias/navarro.pdf>
- Neurofeedback. (2013). *Neurofeedback Internacional S.L.* Recuperado el 1 de mayo de 2014, de <http://www.neurofeedbackinternacional.com/php/cerebro.php>
- NeuroSky, Inc. (28 de Junio de 2010). ASIC\_EEG\_POWER. *MindSet Communications*, 8. Recuperado el 30 de Noviembre de 2013, de <http://neurosky.com/>
- Pololu. (2014). *RN-41 Bluetooth Module*. Recuperado el Marzo de 2014, de <http://www.pololu.com/product/2160>

- Pololu. (2014). *Tamiya 70168 Double Gearbox Kit* . Recuperado el Marzo de 2014, de <http://www.pololu.com/product/1114/>
- Pololu. (2014). *Tamiya 70194 Spike Tire Set (2 tires)* . Recuperado el Marzo de 2014, de <http://www.pololu.com/product/1687>
- School of Engineering EPFL -CNBI, P. J. (23 de Junio de 2013). *Chair in Non-Invasive Brain-Machine Interface*. Recuperado el 30 de Abril de 2014, de <http://cnbi.epfl.ch/>
- Seedstudio. (2013). *ThinkGear AM - Brainwave Sensor*. Recuperado el 26 de Marzo de 2014, de <http://www.seedstudio.com/depot/ThinkGear-AM-Brainwave-Sensor-p-1441.html>
- Sistemas Biométricos. (8 de Octubre de 2012). *sistemasbiométricos*. Recuperado el 20 de Enero de 2014, de [www.sistemasbiometricos.cl/web/tag/patentes-biometria/](http://www.sistemasbiometricos.cl/web/tag/patentes-biometria/)
- Soo, K. (18 de Agosto de 2013). *ThinkGear Connector*. Recuperado el 24 de Marzo de 2014, de [http://developer.neurosky.com/docs/doku.php?id=thinkgear\\_connector\\_tgc](http://developer.neurosky.com/docs/doku.php?id=thinkgear_connector_tgc)
- Tecnología, I. U. (2014). *Qué es la Robótica?* Recuperado el 4 de Abril de 2014, de <http://www.cibor.com.ar/>
- Yaguana, A. D. (2014). Desarrollo e Implementación de una interfaz de comunicacion que permita la interacción entre un usuario y las señales emitidas por sus ondas cerebrales usando un dispositivo EEG de NeuroSky para controlar periféricos electrónicos. Guayaquil: UCSG. Recuperado el 28 de Enero de 2014

## **BIBLIOGRAFÍA**

Arduino. (2014). *Adafruit Motor Shield*. Recuperado el 10 de Enero de 2014, de <http://playground.arduino.cc/Main/AdafruitMotorShield>

Arena, S. Z. (2008). *Metodología de la Investigación*. Recuperado el 24 de 10 de 2013, de <http://metodologiadelainvestigacion.forumcommunity.net/?t=14325119>.

Atmel. (2014). *ATmega328P*. Recuperado el 14 de Marzo de 2014, de <http://www.atmel.com/devices/atmega328p.aspx>

Duarte, O. (5 de Enero de 1998). *Sistemas de lógica difusa*. Recuperado el 5 de Mayo de 2014, de [http://www.ing.unal.edu.co/~ogduarte/Htextos/maestro/node20\\_mn.html](http://www.ing.unal.edu.co/~ogduarte/Htextos/maestro/node20_mn.html)

Emotiv. (2004). *emotiv*. Recuperado el 2 de Febrero de 2014, de <http://www.emotiv.com/about/>

Garcia, R. (2010). *Investigación de Neuro Tecnologías Innovadoras Aplicadas Comercialmente*. Universidad Católica de Santiago de Guayaquil, Sistema de Investigación y Desarrollo. Guayaquil: Sinde-UCSG. Recuperado el 25 de Enero de 2014

Granada. (2005). *Actividad Patológica*. Recuperado el 20 de Marzo de 2014, de <http://neurofisiologiagranada.com/eeg/eeg-activpatologica.htm>

Granada. (2005). *La Actividad Cerebral*. Recuperado el 20 de Marzo de 2014, de <http://neurofisiologiagranada.com/eeg/eeg-actividadcerebral.htm>

- H. García, S. A. (2009). Detección y Clasificación de Artefactos. *XIV SIMPOSIO DE TRATAMIENTO DE SEÑALES, IMAGENES Y VISIÓN ARTIFICIAL*, 8. Recuperado el 15 de Enero de 2014 , de [http://www.uniquindio.edu.co/uniquindio/revistadyp/backup20121214/archivo/edicion\\_4/Articulos/Articulo%20Stsiva%20final.pdf](http://www.uniquindio.edu.co/uniquindio/revistadyp/backup20121214/archivo/edicion_4/Articulos/Articulo%20Stsiva%20final.pdf)
- Merchán, D. G. (15 de Diciembre de 2008). *Introducción a los sistemas Brain Computer Interface*. Recuperado el 1 de Marzo de 2014, de <http://www.lacofa.es/index.php/general/introduccion-a-los-sistemas-brain-computer-interface>
- Millán, J. d. (6 de Febrero de 2011). Curiosidad y trabajo en equipo. (L. M. Ariza, Ed.) *REPORTAJE: JOSÉ DEL ROCÍO MILLÁN*. Recuperado el 25 de Marzo de 2014 , de [http://elpais.com/diario/2011/02/06/eps/1296977214\\_850215.html](http://elpais.com/diario/2011/02/06/eps/1296977214_850215.html)
- Ministerio de Educación, P. d. (2012). Introducción a la Robótica. *Robótica*, 36. Recuperado el 14 de Marzo de 2014
- Navarro, A. (25 de Septiembre de 2009). *El cerebro emocional*. Recuperado el 4 de Mayo de 2014, de <http://www.asociacioneducar.com/monografias/navarro.pdf>
- Neurofeedback. (2013). *Neurofeedback Internacional S.L.* Recuperado el 1 de mayo de 2014, de <http://www.neurofeedbackinternacional.com/php/cerebro.php>
- NeuroSky, Inc. (28 de Junio de 2010). ASIC\_EEG\_POWER. *MindSet Communications*, 8. Recuperado el 30 de Noviembre de 2013, de <http://neurosky.com/>
- Pololu. (2014). *RN-41 Bluetooth Module*. Recuperado el Marzo de 2014, de <http://www.pololu.com/product/2160>

- Pololu. (2014). *Tamiya 70168 Double Gearbox Kit* . Recuperado el Marzo de 2014, de <http://www.pololu.com/product/114/>
- Pololu. (2014). *Tamiya 70194 Spike Tire Set (2 tires)* . Recuperado el Marzo de 2014, de <http://www.pololu.com/product/1687>
- School of Engineering EPFL -CNBI, P. J. (23 de Junio de 2013). *Chair in Non-Invasive Brain-Machine Interface*. Recuperado el 30 de Abril de 2014, de <http://cnbi.epfl.ch/>
- Seedstudio. (2013). *ThinkGear AM - Brainwave Sensor*. Recuperado el 26 de Marzo de 2014, de <http://www.seedstudio.com/depot/ThinkGear-AM-Brainwave-Sensor-p-1441.html>
- Sistemas Biométricos. (8 de Octubre de 2012). *sistemasbiométricos*. Recuperado el 20 de Enero de 2014, de [www.sistemasbiometricos.cl/web/tag/patentes-biometria/](http://www.sistemasbiometricos.cl/web/tag/patentes-biometria/)
- Soo, K. (18 de Agosto de 2013). *ThinkGear Connector*. Recuperado el 24 de Marzo de 2014, de [http://developer.neurosky.com/docs/doku.php?id=thinkgear\\_connector\\_tgc](http://developer.neurosky.com/docs/doku.php?id=thinkgear_connector_tgc)
- Tecnología, I. U. (2014). *Qué es la Robótica?* Recuperado el 4 de Abril de 2014, de <http://www.cibor.com.ar/>
- Yaguana, A. D. (2014). Desarrollo e Implementación de una interfaz de comunicacion que permita la interacción entre un usuario y las señales emitidas por sus ondas cerebrales usando un dispositivo EEG de NeuroSky para controlar periféricos electrónicos. Guayaquil: UCSG. Recuperado el 28 de Enero de 2014

## ANEXOS

### ANEXO 4.1

#### Implementación del algoritmo de manejo del sensor HC-SR04 en el IDE Arduino.

```
const int numOfReadings = 10;
int readings[numOfReadings];
int arrayIndex = 0;
int total = 0;
int averageDistance = 0;
int echoPin = 3;
int initPin = 4;
unsigned long pulseTime = 0;
unsigned long dist = 0;

void setup() {
  pinMode(LED, OUTPUT); // inicia pin led como salida
  pinMode(echoPin, INPUT); // inicia pin echo pin como entrada
  pinMode(initPin, OUTPUT); // inicia pin init como salida

  for (int thisReading = 0; thisReading < numOfReadings; thisReading++) {
    readings[thisReading] = 0;
    Serial.begin(57600);
  }

  int ping_sense() {

    digitalWrite(initPin, HIGH); // envia pulso de 10 microsegundos
    delayMicroseconds(10); // espera 10 microsegundos antes de encender
    digitalWrite(initPin, LOW); // espera envia de pulso

    pulseTime = pulseIn(echoPin, HIGH);
    dist = pulseTime / 58;
    total = total - readings[arrayIndex];
    readings[arrayIndex] = dist;
    total = total + readings[arrayIndex];
    arrayIndex = arrayIndex + 1;

    if (arrayIndex >= numOfReadings) {
      arrayIndex = 0;
    }
    averageDistance = total / numOfReadings;
    return averageDistance;
  }
}
```

## ANEXO 4.2

**Implementación de código de lógica difusa para detección de obstáculos en modo autónomo del robot.**

```
#include <FuzzyRule.h>
#include <FuzzyComposition.h>
#include <Fuzzy.h>
#include <FuzzyRuleConsequent.h>
#include <FuzzyOutput.h>
#include <FuzzyInput.h>
#include <FuzzyIO.h>
#include <FuzzySet.h>
#include <FuzzyRuleAntecedent.h>
```

*Declaracion del objeto Fussy:*

```
Fuzzy* fuzzy = new Fuzzy();
float Output_fussy;
float Output_fussy_Computed;
```

*Configuracion de variables de entrada para el modelo Fussy:*

```
FuzzyInput* distance = new FuzzyInput(1);
```

```
FuzzySet* verylitle = new FuzzySet(5, 15, 15, 30);
distance->addFuzzySet(verylitle);
FuzzySet* litle = new FuzzySet(20, 40, 40, 80);
distance->addFuzzySet(litle);
FuzzySet* safe = new FuzzySet(70, 100, 100, 200);
distance->addFuzzySet(safe);
FuzzySet* big = new FuzzySet(150, 300, 300, 500);
distance->addFuzzySet(big);
FuzzySet* verybig = new FuzzySet(400, 1000, 1000, 2900);
distance->addFuzzySet(verybig);
```

```
fuzzy->addFuzzyInput(distance);
```

*Configuracion de variables de salida para el modelo Fussy:*

```

FuzzyOutput* velocity = new FuzzyOutput(1);
FuzzySet* vslow = new FuzzySet(8, 12, 18, 24);
velocity->addFuzzySet(vslow);
FuzzySet* slow = new FuzzySet(24, 64, 96, 112);
velocity->addFuzzySet(slow);
FuzzySet* normal = new FuzzySet(112, 128, 144, 160);
velocity->addFuzzySet(normal);
FuzzySet* fast = new FuzzySet(160, 192, 208, 224);
velocity->addFuzzySet(fast);
FuzzySet* veryfast = new FuzzySet(224, 240, 250, 255);
velocity->addFuzzySet(veryfast);

```

```
fuzzy->addFuzzyOutput(velocity);
```

*Establecimiento de Reglas de Control Fussy:*

```

//IF distance = very small THEN speed = very slow
FuzzyRuleAntecedent* ifDistanceVerysmall = new FuzzyRuleAntecedent(verylitle);
ifDistanceVerysmall->joinSingle(verylitle);
FuzzyRuleConsequent* thenVelocityVeryslow = new FuzzyRuleConsequent();
thenVelocityVeryslow->addOutput(vslow);
FuzzyRule* fuzzyRule01 = new FuzzyRule(1, ifDistanceVerysmall, thenVelocityVeryslow);
fuzzy->addFuzzyRule(fuzzyRule01);

```

```

//IF distance = slighty close THEN speed = slow bwds
FuzzyRuleAntecedent* ifDistanceSmall = new FuzzyRuleAntecedent();
ifDistanceSmall->joinSingle(litle);
FuzzyRuleConsequent* thenVelocitySlow = new FuzzyRuleConsequent();
thenVelocitySlow->addOutput(slow);
FuzzyRule* fuzzyRule02 = new FuzzyRule(2, ifDistanceSmall, thenVelocitySlow);
fuzzy->addFuzzyRule(fuzzyRule02);

```

```

//IF distance = center THEN speed = zero.
FuzzyRuleAntecedent* ifDistanceSafe = new FuzzyRuleAntecedent();
ifDistanceSafe->joinSingle(safe);
FuzzyRuleConsequent* thenVelocityNormal = new FuzzyRuleConsequent();
thenVelocityNormal->addOutput(normal);
FuzzyRule* fuzzyRule03 = new FuzzyRule(3, ifDistanceSafe, thenVelocityNormal);
fuzzy->addFuzzyRule(fuzzyRule03); // Adicionando o FuzzyRule ao objeto Fuzzy

```



```

//IF distance = slightly far THEN speed = slow fwd
FuzzyRuleAntecedent* ifDistanceBig = new FuzzyRuleAntecedent();
ifDistanceBig->joinSingle(big);
FuzzyRuleConsequent* thenVelocityFast = new FuzzyRuleConsequent();
thenVelocityFast->addOutput(fast);
FuzzyRule* fuzzyRule04 = new FuzzyRule(4, ifDistanceBig, thenVelocityFast);
fuzzy->addFuzzyRule(fuzzyRule04);

//IF distance = very far THEN speed = fast fwd
FuzzyRuleAntecedent* ifDistanceVerybig = new FuzzyRuleAntecedent();
ifDistanceVerybig->joinSingle(verybig);
FuzzyRuleConsequent* thenVelocityVeryfast = new FuzzyRuleConsequent();
thenVelocityVeryfast->addOutput(veryfast);
FuzzyRule* fuzzyRule05 = new FuzzyRule(5, ifDistanceVerybig,
thenVelocityVeryfast);
fuzzy->addFuzzyRule(fuzzyRule05);

```

Finalmente, se crea una función de llamado al proceso de fusificación, defusificación y computo de las variables de entrada para la toma de decisión sobre la variable de salida según las reglas de control implementadas.

```

float contro_fussy(int imput_var) {
    fuzzy->setInput(1, imput_var);
    fuzzy->fuzzify();
    Output_fussy = fuzzy->defuzzify(1);
    Serial.print("Distancia calculada:");
    Serial.print(imput_var);
    Serial.print("Velocidad:");
    Serial.println(Output_fussy);
    Output_fussy_Computed = Output_fussy;
    return Output_fussy_Computed;
}

```

### **ANEXO 4.3**

#### **Implementación de código para demostración de capacidades del módulo de motores Arduino Motor Shield diseñado por Adafruit.**

El código puede ser descargado en el siguiente enlace o descargado del objeto vinculado:

<http://playground.arduino.cc/Main/AdafruitMotorShield>



AdafruitMotorShield.i  
no

## ANEXO 5.1

Diseño de un boceto de GUI basada en el boceto IBNS\_EEG\_VISUALIZER usando el IDE Processing.

*Declaracion de librerías a usar.*

```
import neurosky.*;
import org.json.*;
import controlP5.*;
```

*Declaracion de métodos y objetos.*

```
ControlP5 cp5;
ControlTimer C;
Textlabel T;
ThinkGearSocket neuroSocket;
Slider abc;
Textarea myTextarea;
Println console;
```

*Declaracion de variables globales.*

```
int c = 0;
PrintWriter output;
String fname = "C:/Users/Public/Documents/IBNS_MindBot_log";
PImage img;
PFont font;
PImage test;
int rawArraySize = 512;
int index = 0;
int y_pos=230;
int x_pos=100;
int show_attention=0;
int show_meditation=0;
int show_poorSignalEvent=0;
int[] show_raw= new int[512];
int show_blinkStrength=0;
int show_delta=0;
int show_theta=0;
```

```

int show_low_alpha=0;
int show_high_alpha=0;
int show_low_beta=0;
int show_high_beta=0;
int show_low_gamma=0;
int show_mid_gamma=0;
int x=720;//720//610//320
int y=960;//960//850//240
int Attention = 0;
int Meditation = 0;
int blinkStrength=0;
int delta=0;
int theta=0;
int low_alpha=0;
int high_alpha=0;
int low_beta=0;
int high_beta=0;
int low_gamma=0;
int mid_gamma=0;
int Sensibility=40;
int cont_blink=0;
color eye_blink = color(#1A3A5A);
int var=0;
PImage img_UP;
PImage img_DOWN;
PImage img_LEFT;
PImage img_RIGHT;
PImage img_STOP;

```

Construccion de función Setup().

```

void setup() {

    test = loadImage("backcore13.png");
    cp5 = new ControlP5(this);

    //Consola
    cp5.enableShortcuts();
    myTextarea = cp5.addTextarea("txt").setPosition(350, 200).setSize(300,
200).setFont(createFont("",

```

```
12)).setLineHeight(14).setColor(color(255)).setColorBackground(#325476).setCol  
orForeground(color(255, 100));  
console = cp5.addConsole(myTextarea);
```

```
ThinkGearSocket neuroSocket = new ThinkGearSocket(this);
```

```
try {  
  neuroSocket.start();  
}  
catch (Exception e) {  
  println("Please, turn On the IBNS");  
}
```

```
size(x, y);  
smooth();  
noStroke();
```

```
output = createWriter(fname+".csv");
```

```
C = new ControlTimer();  
T = new Textlabel(cp5, "--", 520, 377);  
C.setSpeedOfTime(1);
```

```
cp5.addKnob("Attention").setPosition(22,  
221).setRadius(65).setScrollSensitivity(0.001).setMin(0).setMax(100).setDisplayS  
tyle(Controller.ARC);  
cp5.addKnob("Meditation").setPosition(182,  
221).setRadius(65).setScrollSensitivity(0.001).setMin(0).setMax(100).setDisplayS  
tyle(Controller.ARC);//.setColorForeground(color(255)).setColorBackground(color  
(#230E76)).setColorActive(color(#861410));  
cp5.addSlider("Delta").setPosition(350, 440).setSize(300, 20).setRange(0,  
10000000).setNumberOfTickMarks(50).setSliderMode(Slider.FIX);  
cp5.addSlider("Theta").setPosition(350, 480).setSize(300, 20).setRange(0,  
3000000).setNumberOfTickMarks(50).setSliderMode(Slider.FIX);  
cp5.addSlider("Low_alpha").setPosition(350, 520).setSize(300, 20).setRange(0,  
300000).setNumberOfTickMarks(50).setSliderMode(Slider.FIX);  
cp5.addSlider("High_alpha").setPosition(350, 560).setSize(300, 20).setRange(0,  
300000).setNumberOfTickMarks(50).setSliderMode(Slider.FIX);
```

```
cp5.addButton("blink_mouse")
  .setPosition(145, 545)
  .setImages(loadImage("blink_click_done.png"), loadImage("blink_click.png"),
loadImage("blink.png"))
  .updateSize();
```

```
cp5.addSlider("blinkStrength")
  .setRange(0, 255)
  .setPosition(90, 410)
  .setSize(160, 20);
```

```
img_UP = loadImage("Up.png");
img_DOWN = loadImage("down.png");
img_LEFT = loadImage("Left.png");
img_RIGHT = loadImage("Right.png");
img_STOP = loadImage("stop.png");
}
```

*Construcción de la GUI mediante la función draw()*

```
void draw() {
  background(0); //RGB, Claridad
  noStroke();

  image(test, 0, 0);
  fill(255, 255, 255); //blanco RGB, Claridad letra
  font = createFont("Tahoma", 24);
  textFont(font);
  text("IBNS MindBot Simulator", 260, 50); //texto

  fill(255, 255, 255); //blanco RGB, Claridad letra
  font = createFont("Tahoma", 14);
  textFont(font);
  text("Designed by Sophia V. Alava O.", 280, 100); //texto

  fill(255, 255, 255); //blanco RGB, Claridad letra
  font = createFont("Tahoma", 14);
  textFont(font);

  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+":", y-50, 230); //texto

Attention = show_attention;
cp5.getController("Attention").setValue(Attention);
Meditation = show_meditation;
cp5.getController("Meditation").setValue(Meditation);
blinkStrength = show_blinkStrength;
cp5.getController("blinkStrength").setValue(blinkStrength);

delta = show_delta;
cp5.getController("Delta").setValue(delta);
theta = show_theta;
cp5.getController("Theta").setValue(theta);
low_alpha = show_low_alpha;
cp5.getController("Low_alpha").setValue(low_alpha);
high_alpha = show_high_alpha;
cp5.getController("High_alpha").setValue(high_alpha);
low_beta = show_low_beta;
cp5.getController("Low_beta").setValue(low_beta);
high_beta = show_high_beta;
cp5.getController("High_beta").setValue(high_beta);
low_gamma = show_low_gamma;
cp5.getController("Low_gamma").setValue(low_gamma);
mid_gamma = show_mid_gamma;
cp5.getController("Mid_gamma").setValue(mid_gamma);

if (show_blinkStrength>Sensibility) {
  cont_blink=cont_blink+1;
}

switch(cont_blink) {
case 1:
  UP();
  show_blinkStrength=0;
  break;
case 3:
  RIGHT();

```

```

show_blinkStrength=0;
break;
case 5:
DOWN();
show_blinkStrength=0;
break;
case 7:
LEFT();
show_blinkStrength=0;
break;
case 8:
cont_blink=0;
show_blinkStrength=0;
break;
case 0:
case 2:
case 4:
case 6:
STOP();
show_blinkStrength=0;
break;
}

fill(#E3DE3E);//blanco RGB, Claridad letra
font = createFont("Calibri-Bold-48", 18);//tamaño normal
textFont(font);
if (show_poorSignalEvent==200) {
text(" No contact with the skin, please ", x-300, y-30);//texto
text(" reconnect IBNS now!", x-300, y-15);//texto
fill(255, 0, 0);//blanco RGB, Claridad letra
noStroke();//no borde
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 IBNS!", x-200, y-15);//texto
fill(255, 155, 0);//blanco RGB, Claridad letra
noStroke();//no borde
ellipse(40, y-30, 20, 20);
}
}

```



```
if (show_poorSignalEvent==0) {
  fill(0, 255, 0); //blanco RGB, Claridad letra
  text("IBNS conected!", x-220, y-23);
  fill(0, 255, 0); //blanco RGB, Claridad letra
  noStroke(); //no borde
  ellipse(40, y-30, 20, 20);
}
```

```
output.println(
show_poorSignalEvent+", //poorSignal
+show_attention+", //attention
+show_meditation+", //meditation
+show_blinkStrength+", //blinkStrength
); // Write the coordinate to the file
}
```

*Construccion de funciones para control de Eventos.*

```
public void controlEvent(ControlEvent theEvent) {
  //println(theEvent.getController().getName());
}
```

```
public void UP() {
  cont_blink=1;
  println("UP: "+show_attention);
}
```

```
public void DOWN() {
  cont_blink=5;
  println("DOWN: "+show_attention);
}
```

```
public void LEFT() {
  cont_blink=7;
  println("LEFT: "+show_attention);
}
```

```
public void RIGHT() {
  cont_blink=3;
  println("RIGHT: "+show_attention);
}
```

```
public void STOP() {  
    println("STOP");  
}
```

*Construccion de Eventos generados por el TGDC, basados en el código original del Sr. Angel Yaguana en su boceto IBNS\_EEG\_VISUALIZER usando las Api de conexión al modulo TGAM1 de NeuroSky.*

```
public void blink_mouse() {  
    int residuo = var % 2;  
    if (residuo==0)  
        show_blinkStrength=Sensibility+10;  
    var=var+1;  
}
```

```
void poorSignalEvent(int sig) {  
    println("SignalEvent: "+sig);  
    show_poorSignalEvent = sig;  
}
```

```
void attentionEvent(int attentionLevel) {  
    println("Attention Level: " + attentionLevel);  
    show_attention = attentionLevel;  
}
```

```
void meditationEvent(int meditationLevel) {  
    println("Meditation Level: " + meditationLevel);  
    show_meditation = meditationLevel;  
}
```

```
void blinkEvent(int blinkStrengthLevel) {  
  
    println("blinkStrength: " + blinkStrengthLevel);  
    show_blinkStrength = blinkStrengthLevel;  
}
```

```
void eegEvent(int delta, int theta, int low_alpha, int high_alpha, int low_beta, int  
high_beta, int low_gamma, int mid_gamma) {  
    println("Delta Level: " + delta);
```

```
println("Theta Level: " + theta);
println("Low_alpha Level: " + low_alpha);
println("High_alpha Level: " + high_alpha);
println("Low_beta Level: " + low_beta);
println("High_beta Level: " + high_beta);
println("Low_gamma Level: " + low_gamma);
println("Mid_gamma Level: " + 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;
}

void rawEvent(int[] raw) {
}

void stop() {
    println("Saved as "+fname);
    neuroSocket.stop();
    output.close(); // Finishes the file
    exit(); // Stops the program
    super.stop();
}
```

## **ANEXO 3.2**

### **Firmware del robot.**

*Boseto Firmware.ino en IDE Arduino*

*Declaracion de librerías externas.*

```
#include <FuzzyRule.h>
#include <FuzzyComposition.h>
#include <Fuzzy.h>
#include <FuzzyRuleConsequent.h>
#include <FuzzyOutput.h>
#include <FuzzyInput.h>
#include <FuzzyIO.h>
#include <FuzzySet.h>
#include <FuzzyRuleAntecedent.h>
```

*Declaracion de variables para IBNS y el modulo de control de perifericos.*

```
static unsigned char inputString[256];
boolean stringComplete = false;
byte generatedChecksum;
byte poorQuality = 200; //0-255
byte attention = 0; //0-100
byte meditation = 0; //0-100
boolean BLINK_STRENGTH = 0; //0-1
int length;
int PARSER_PQS = 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;
int cont_blink = 0;
boolean newData = false;
byte sensibiliy = 10;
```

*Declaracion de variables para control difuso.*

```
Fuzzy* fuzzy = new Fuzzy();//Declaring a new fuzzy object.
float Output_fussy;
```

```
float Output_fussy_Computed;
```

*Declaracion de variables para manipulacion del driver de motores L293D*

```
int in1 = 7;  
int in2 = 6;  
int en1 = 5;//PWM  
int en2 = 10;//PWM  
int in3 = 8;  
int in4 = 9;
```

*Declaracion de variables para el sensor Ultrasonico HC-SR04*

*Declaracion de variables:*

```
const int numOfReadings = 10;  
int readings[numOfReadings];  
int arrayIndex = 0;  
int total = 0;  
int averageDistance = 0;  
int echoPin = 3;  
int initPin = 4;  
unsigned long pulseTime = 0;  
unsigned long dist = 0;
```

*Configuracion del modulo Arduino Uno con Atmega328*

```
void setup() {  
  pinMode(LED, OUTPUT);// set init pin as output  
  pinMode(echoPin, INPUT);// set echo pin as input  
  pinMode(initPin, OUTPUT);// set init pin as output  
  
  for (int thisReading = 0; thisReading < numOfReadings; thisReading++) {  
    readings[thisReading] = 0;  
  
    FuzzyInput* distance = new FuzzyInput(1);  
  
    FuzzySet* verylitle = new FuzzySet(5, 15, 15, 30);  
    distance->addFuzzySet(verylitle);  
    FuzzySet* litle = new FuzzySet(20, 40, 40, 80);  
    distance->addFuzzySet(litle);
```

```
FuzzySet* safe = new FuzzySet(70, 100, 100, 200);
distance->addFuzzySet(safe);
FuzzySet* big = new FuzzySet(150, 300, 300, 500);
distance->addFuzzySet(big);
FuzzySet* verybig = new FuzzySet(400, 1000, 1000, 2900);
distance->addFuzzySet(verybig);
```

```
fuzzy->addFuzzyInput(distance);
```

*Configuración de variables de salida para el modelo Fussy:*

```
FuzzyOutput* velocity = new FuzzyOutput(1);
FuzzySet* vslow = new FuzzySet(8, 12, 18, 24);
velocity->addFuzzySet(vslow);
FuzzySet* slow = new FuzzySet(24, 64, 96, 112);
velocity->addFuzzySet(slow);
FuzzySet* normal = new FuzzySet(112, 128, 144, 160);
velocity->addFuzzySet(normal);
FuzzySet* fast = new FuzzySet(160, 192, 208, 224);
velocity->addFuzzySet(fast);
FuzzySet* veryfast = new FuzzySet(224, 240, 250, 255);
velocity->addFuzzySet(veryfast);
```

```
fuzzy->addFuzzyOutput(velocity);
```

*Establecimiento de Reglas de Control Fussy:*

```
//IF distance = very small THEN speed = very slow
FuzzyRuleAntecedent* ifDistanceVerysmall = new FuzzyRuleAntecedent();
ifDistanceVerysmall->joinSingle(verylitle);
FuzzyRuleConsequent* thenVelocityVeryslow = new FuzzyRuleConsequent();
thenVelocityVeryslow->addOutput(vslow);
FuzzyRule* fuzzyRule01 = new FuzzyRule(1, ifDistanceVerysmall, thenVelocityVeryslow);
fuzzy->addFuzzyRule(fuzzyRule01);
```

```
//IF distance = slighty close THEN speed = slow bwds
FuzzyRuleAntecedent* ifDistanceSmall = new FuzzyRuleAntecedent();
ifDistanceSmall->joinSingle(litle);
FuzzyRuleConsequent* thenVelocitySlow = new FuzzyRuleConsequent();
thenVelocitySlow->addOutput(slow);
FuzzyRule* fuzzyRule02 = new FuzzyRule(2, ifDistanceSmall, thenVelocitySlow);
fuzzy->addFuzzyRule(fuzzyRule02);
```

```
digitalWrite(initPin, HIGH); // send 10 microsecond pulse
delayMicroseconds(10); // wait 10 microseconds before turning off
digitalWrite(initPin, LOW); // stop sending the pulse
```

```
pulseTime = pulseIn(echoPin, HIGH);
dist = pulseTime / 58;
total = total - readings[arrayIndex];
readings[arrayIndex] = dist;
total = total + readings[arrayIndex];
arrayIndex = arrayIndex + 1;

if (arrayIndex >= numOfReadings) {
  arrayIndex = 0;
}
averageDistance = total / numOfReadings;
return averageDistance;
}
```

*Función de llamado al proceso de fusificación, defusificación y computo de las variables de entrada para la toma de decisión sobre la variable de salida según las reglas de control implementadas.*

```
float contro_fussy(int imput_var) {
  fuzzy->setInput(1, imput_var);
  fuzzy->fuzzify();
  Output_fussy = fuzzy->defuzzify(1);
  Serial.print("Distancia calculada:");
  Serial.print(imput_var);
  Serial.print("Velocidad:");
  Serial.println(Output_fussy);
  Output_fussy_Computed = Output_fussy;
  return Output_fussy_Computed;
}
```

*Construcción de la función de validación de paquetes enviados desde el módulo de control de periféricos.*

```
void ReadData_from_IBNS() {
  if (ReadOneByte() == 171) {
    length = ReadOneByte() - 2;
  }
}
```

```

for (int i = 0; i < length; i++) {
    inputString[i] = ReadOneByte();
}
if (ReadOneByte() == 0x0A)
    stringComplete = true;
else
    stringComplete = false;
if (stringComplete) {
    generatedChecksum = inputString[8]; //poorQuality+attention+meditation+bit_blink;
    if (generatedChecksum < 201) {
        for (int i = 0; i < length; i++) {
            switch (inputString[i]) {
                case 0x01:
                    i++;
                    poorQuality = inputString[i];
                    break;
                case 0x02:
                    i++;
                    attention = inputString[i];
                    PID_compute(attention);
                    break;
                case 0x03:
                    i++;
                    meditation = inputString[i];
                    break;
                case 0x04:
                    i++;
                    BLINK_STRENGTH = inputString[i];
                    i++;
                    break;
                case 0xAC:
                    newData = true;
                    stringComplete = false;
                    //i++;
                    break;
                default:
                    break;
            } // switch
        } // for loop
        if (poorQuality == 0)

```



```

    digitalWrite(LED, HIGH);
else
    digitalWrite(LED, LOW);
Serial.print("PoorQuality: ");
Serial.print(poorQuality, DEC);
Serial.print(" Attention: ");
Serial.print(attention, DEC);
Serial.print(" Meditation: ");
Serial.print(meditation, DEC);
Serial.print(" BlinkStrenght: ");
Serial.print(BLINK_STRENGTH, DEC);
Serial.print("\n");
} //end checksum
} //end PARSER_SYNC_END
} // end if read 0xAB byte
stringComplete = false;
}

```

*Funcion de control de movimiento y dirección del robot.*

```

void move_mindbot(byte dir, int att, int mdt) {
  if (dir == 0)
  {
    Serial.println("In position..stoped");
    analogWrite(en1, 255);
    digitalWrite(in1, HIGH);
    digitalWrite(in2, HIGH);
    delay(500);
    analogWrite(en1, 0);
    digitalWrite(in1, HIGH);
    digitalWrite(in2, HIGH);
  }
  else if (dir == 1 && att > sensibiliy)
  {
    ping_sense();
    contro_fussy(averageDistance);
    Serial.println("Moving forward...");
    analogWrite(en1, Output_fussy_Computed);
  }
}

```

```

digitalWrite(in1, HIGH);
digitalWrite(in2, LOW);
analogWrite(en2, Output_fussy_Computed);
digitalWrite(in3, HIGH);
digitalWrite(in4, LOW);
}
else if (dir == 2 && mdt > sensibiliy)
{
byte mdt_computed_right = 100 - mdt;
PID_compute(mdt_computed_right);
Serial.println("Moving right...");
analogWrite(en1, Output_PID_Computed); //Fussy
digitalWrite(in1, HIGH);
digitalWrite(in2, LOW);
analogWrite(en2, Output_PID_Computed);
digitalWrite(in3, LOW);
digitalWrite(in4, HIGH);
}
else if (dir == 3 && att > sensibiliy)
{
ping_sense();
contro_fussy(averageDistance);
Serial.println("Moving backward...");
analogWrite(en1, Output_fussy_Computed);
digitalWrite(in1, LOW);
digitalWrite(in2, HIGH);
analogWrite(en2, Output_fussy_Computed);
digitalWrite(in3, LOW);
digitalWrite(in4, HIGH);
}
else if (dir == 4 && mdt > sensibiliy)
{
byte mdt_computed_left = 100 - mdt;
PID_compute(mdt_computed_left);
Serial.println("Moving left...");
analogWrite(en2, Output_PID_Computed); //Fussy
digitalWrite(in1, LOW);
digitalWrite(in2, HIGH);
analogWrite(en2, Output_PID_Computed);
digitalWrite(in3, HIGH);
}

```

```

    digitalWrite(in4, LOW);
}
else
{
    Serial.println("Unknown input.");
}
delay(doopler);
}

```

*Una vez contruidas todas las funciones necesarias de control, se debe implementar una mas que haga el llamado principal a éstas. La función loop es quizás la mas importante de todo este trabajo, incluso más que la de interpretación de las señales neuro sensoriales y que las de control difuso que le dan la capacidad de tomar desiciones autónomas al robot, lo cual ya es un buen inicio para dotarlo de Inteligencia Artificial, esto se debe a que en esta función se etructura una lógica que le permite decidir al robot en que dirección debe ir y es la misma que se utilizó en el boceto de la GUI donde se realizaron las pruebas preliminares para la determinación de los umbrales con los cuales esta función debe trabajar.*

*Funcion loop.*

```

void loop() {
//move_mindbot(1, Output_fussy_Computed, Output_PID_Computed);
//Reading char from IBNS for periferical controller

ReadData_from_IBNS();
if (newData) {
    newData = false;

    if (BLINK_STRENGTH) {
        cont_blink = cont_blink + 1;
    }
    switch (cont_blink) {
        case 0:
        case 2:
        case 4:
        case 6:
            move_mindbot(0, attention, meditation);
            BLINK_STRENGTH = 0;

```

```
    break;
case 1:
    move_mindbot(1, attention, meditation);
    BLINK_STRENGTH = 0;
    break;
case 3:
    move_mindbot(2, attention, meditation);
    BLINK_STRENGTH = 0;
    break;
case 5:
    move_mindbot(3, attention, meditation);
    BLINK_STRENGTH = 0;
    break;
case 7:
    move_mindbot(4, attention, meditation);
    BLINK_STRENGTH = 0;
    break;
case 8:
    cont_blink = 0;
    BLINK_STRENGTH = 0;
    break;
//default:
}
}
}
```

## ANEXO 4.1

**Fabricación de los circuitos impresos en el laboratorio mediante un proceso artesanal, previo pedido de fabricación profesional.**



Figura A-6.1 Materiales.



Figura A-2.2 Transferencia del fotolito mediante calor.



Figura A-6.3 Limpieza de residuos de papel transfer en el PCB previo el ataque químico.



Figura A-6.4 Ataque Químico usando Percloruro de Hierro.

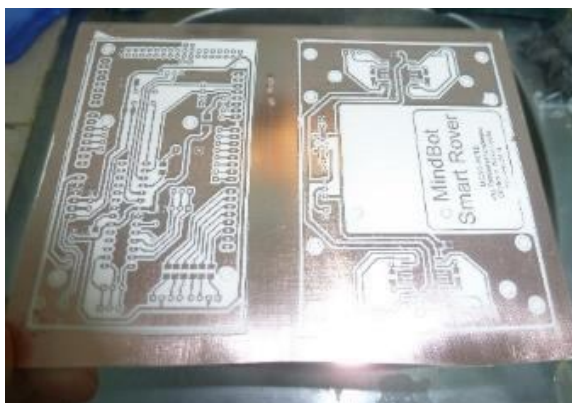


Figura A-6.5 Limpieza del PCB con sosa caustica



Figura A-6.6 Perforación del PCB.

## ANEXO 6.2

### Construcción del robot de morfología tipo Rover.



Figura A-7.1 Materiales para construcción del robot.

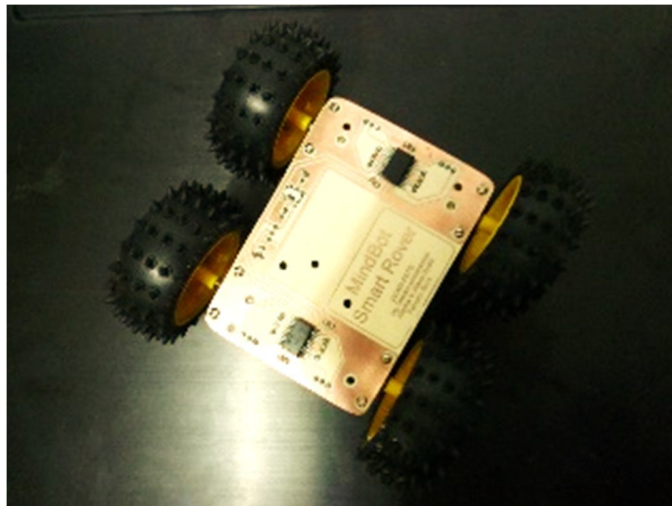


Figura A-7.2 Ensamblaje de cuerpo de suspensión.

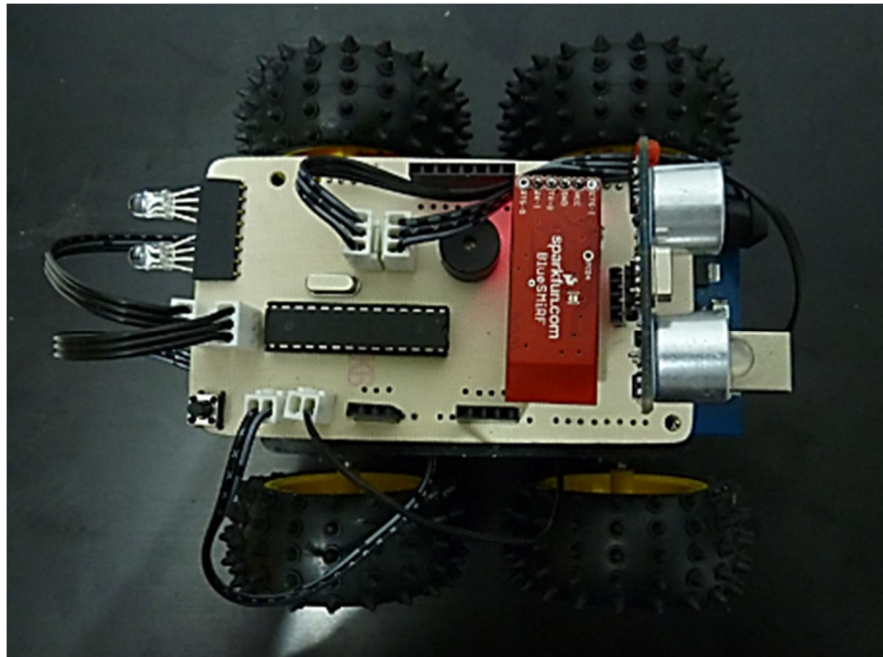


Figura A-7.3 Ensamblaje modular de la circuitería del robot.

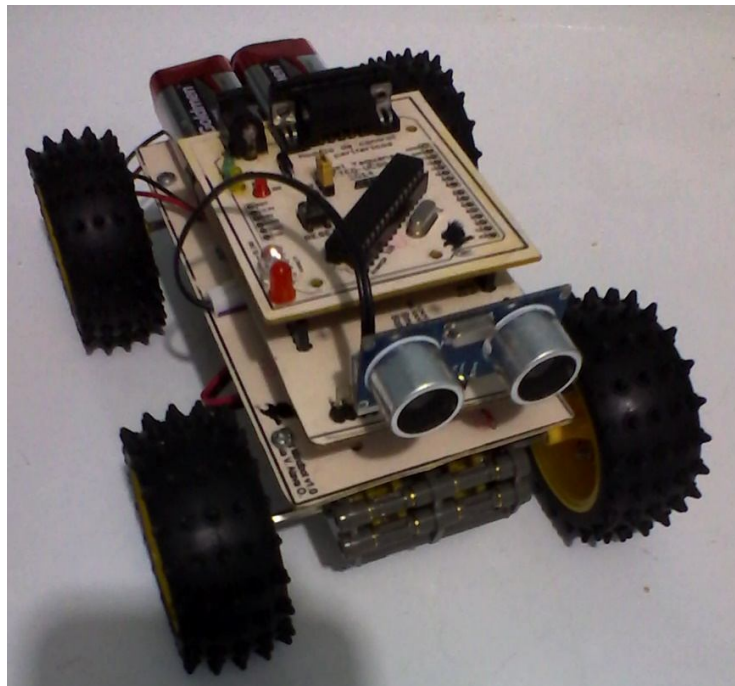


Figura A-7.4 Presentación de la construcción del robot MindBot Smart Rover