



*"Responsabilidad con pensamiento positivo"*

# **UNIVERSIDAD TECNOLÓGICA ISRAEL**

## **TRABAJO DE TITULACIÓN**

### **CARRERA DE ELECTRÓNICA DIGITAL Y TELECOMUNICACIONES**

**TEMA:** Diseño e implementación de un dispositivo electrónico para visualizar caracteres en general de manera alterna, en la publicidad de marcas representadas en el Ecuador por la empresa UNISCAN CIA. LTDA.

**AUTOR:** Juan Carlos Vásquez Revelo

**TUTOR:** Ing. José Robles Salazar Mg.

AÑO 2014

# UNIVERSIDAD TECNOLÓGICA ISRAEL

## APROBACIÓN DEL TUTOR

En mi calidad de Tutor del Trabajo de Graduación certifico:

Que el Trabajo de Titulación **“Diseño e implementación de un dispositivo electrónico para visualizar caracteres en general de manera alterna, en la publicidad de marcas representadas en el Ecuador por la empresa UNISCAN CIA. LTDA.”**, presentado por el Sr. Juan Carlos Vásquez Revelo, estudiante de la Carrera de Electrónica y Telecomunicaciones, reúne los requisitos y méritos suficientes para ser sometido a la evaluación del Tribunal de Grado, que se designe, para su correspondiente estudio y calificación.

D. M. Quito, Agosto 2014

TUTOR

---

Ing. José Robles Salazar Mg.

## **UNIVERSIDAD TECNOLÓGICA ISRAEL**

### **AUTORÍA DEL TRABAJO DE TITULACIÓN**

El abajo firmante, en calidad de estudiante de la Carrera de Electrónica Digital y Telecomunicaciones, declara que los contenidos de este Trabajo de Titulación, requisito previo a la obtención del Grado de Ingeniería en Electrónica y Telecomunicaciones, son absolutamente originales, auténticos y de exclusiva responsabilidad legal y académica del autor.

D. M. Quito, Agosto 2014

---

Juan Carlos Vásquez Revelo

CC: 0401222690

# UNIVERSIDAD TECNOLÓGICA ISRAEL

## APROBACIÓN DEL TRIBUNAL DE GRADO

Los miembros del Tribunal de Grado, aprueban el Trabajo de Titulación de acuerdo con las disposiciones reglamentarias emitidas por la Universidad Tecnológica Israel para títulos de pregrado.

D. M. Quito, Agosto 2014

Para constancia firman:

TRIBUNAL DE GRADO

---

PRESIDENTE

---

MIEMBRO 1

---

MIEMBRO 2

## **AGRADECIMIENTO**

Mi incondicional agradecimiento a mi madre Diana María quién fue mi guía principal de crecimiento personal y profesional durante el tiempo que estuvo conmigo.

A mi Tutor Ing. José Robles Salazar. Mg, quien con su conocimiento direccionamiento y paciencia supo guiarme día a día en el desarrollo y culminación de este proyecto.

También dar gracias a mi esposa por su amor, paciencia y comprensión que con su apoyo desinteresado me dio las fuerzas suficientes para seguir adelante.

Al Ing. Mauro Bolagay que gracias a su ayuda en la institución me brindó las herramientas necesarias en el proceso de profesionalismo.

Gracias también a mis amigos, compañeros, Ingenieros, docentes y personas que dedicaron parte de su tiempo, en la ardua pero feliz tarea de cumplimiento del proyecto.

## DEDICATORIA

A mi Señor Jesús, quien me dio la fe, la constancia, la fortaleza y la salud necesaria para llegar a mi meta inmediata.

A mi madre Diana María y abuelita Mariana de Jesús, quien sé que me observan desde algún rincón del cielo, personas que desde pequeño me enseñaron a luchar y ser valiente hasta cumplir mis metas trazadas. ¡Las llevo por siempre en mi corazón!

A mis queridos hijos Deysi Anabel y Maikel Sebastián quienes supieron prestarme inconscientemente el tiempo que les pertenecía durante estos años de preparación. ¡Gracias hijos míos!

A mi esposa Lorena Uyana, quien encaminada de la mano junto a mí, estuvo constantemente todo el tiempo brindándome su amor, su cariño, su paciencia, y el ánimo total. ¡Gracias! ¡Te amo!

## ÍNDICE GENERAL

APROBACIÓN DEL TUTOR .....	II
AUTORÍA DEL TRABAJO DE TITULACIÓN .....	III
APROBACIÓN DEL TRIBUNAL DE GRADO .....	IV
AGRADECIMIENTO.....	V
DEDICATORIA.....	VI
ÍNDICE GENERAL.....	VII
ÍNDICE DE FIGURAS .....	IX
INTRODUCCIÓN .....	1
OBJETIVOS.....	2
Objetivo general.....	2
Objetivos específicos .....	2
CAPÍTULO I.....	3
FUNDAMENTACIÓN TEÓRICA.....	3
1.1    Introducción .....	3
1.2    Antecedentes del efecto POV (Persistence of Vision) .....	3
1.2.1    Primeros Juguetes Ópticos .....	3
1.2.1.1    El taumatropo o roboscopio .....	3
1.2.1.2    El Estroboscopio .....	4
1.2.1.3    El Zootropo .....	4
1.3    LEDs y Láser .....	5
1.3.1    Diodos LED: ¿Cómo funcionan? .....	5
1.3.2    Funcionamiento del diodo láser.....	6
1.3.3    Diferencias entre el diodo láser y el diodo LED .....	6
1.3.4    Láseres riesgo en su utilización efectos biológicos .....	6
1.3.5    Clasificación de los láseres .....	7
1.4    Motor.....	8
1.5    Tecnología Arduino.....	9
1.5.1    Hardware Arduino .....	9
1.5.2    Software Arduino.....	11
CAPÍTULO II .....	12
DESCRIPCIÓN DEL PROCESO INVESTIGATIVO REALIZADO .....	12
CAPÍTULO III .....	17

PRESENTACIÓN DE RESULTADOS .....	17
3.1    Introducción .....	17
3.2    Propuesta de solución del problema .....	17
3.3    Diagrama de bloques .....	17
3.3.1  Diagrama de Bloques de Hardware del dispositivo POV .....	17
3.4    Simulación y montaje de componentes en protoboard .....	20
3.4.1  Diagrama de Bloques Software del lenguaje Arduino.....	22
3.5    Desarrollo e implementación .....	26
3.6    Validación del sistema .....	31
3.6.1  Pruebas experimentales.....	31
3.6.2  Pruebas finales .....	32
3.7    Evaluación Técnica .....	33
3.8    Resultados .....	33
3.9    Análisis de Resultados.....	37
CONCLUSIONES Y RECOMENDACIONES.....	38
Conclusiones.....	38
Recomendaciones.....	38
BIBLIOGRAFÍA .....	39
ANEXOS .....	40

## ÍNDICE DE FIGURAS

Figura 1.1 El taumatropo o roboscopio.....	3
Figura 1.2 El Estroboscopio .....	4
Figura 1.3 El Zootropo.....	4
Figura 1.4 Ejemplo de Motor DC, 0 a 5000 revoluciones por minuto .....	9
Figura 1.5 Placa Arduino UNO (Frente y posterior) .....	10
Figura 1.6 Placa Arduino Leonardo (frente y posterior) .....	10
Figura 1.7 Placa Arduino Bluetooth HC-06 (frente y posterior) .....	11
Figura 1.8 Lenguaje de Referencia Arduino .....	11
Figura 2.1 Formato Entrevista. ....	14
Figura 2.2 Porcentaje de entrevistas por cada pregunta .....	16
Figura 3.1 Diagrama de bloques funcionamiento mecánico. ....	18
Figura 3.2 Fuente regulable .....	18
Figura 3.3 Etapa de movimiento.....	19
Figura 3.4 Ubicación de Leds.....	20
Figura 3.5 Imagen generada. ....	20
Figura 3.6 Simulación de componentes.....	21
Figura 3.7 Unión de terminales .....	21
Figura 3.8 Placa Arduino y componentes para la simulación.....	21
Figura 3.9 Diagrama de bloques programación Arduino.....	22
Figura 3.10 Letras mayúsculas 7x5.....	23
Figura 3.11 Giro de patrón de caracteres .....	23
Figura 3.12 Transformación a bits y array en Arduino .....	24
Figura 3.13 Descarga de software Arduino.....	24
Figura 3.14 Iniciando la aplicación en Microsoft Windows 7 Enterprise de 32 bits.....	25
Figura 3.15 Acceso a la plataforma .....	25
Figura 3.16 Declarando variables en el entorno Arduino .....	26
Figura 3.17 Desarrollo de la placa electrónica para los Leds.....	27
Figura 3.18 Montaje de Leds en la placa PCB.....	28
Figura 3.19 Identificación de los pines en Arduino Leonardo.....	28
Figura 3.20 Ubicación y montaje de los componentes utilizados. ....	29
Figura 3.21 Ubicación total con todos los componentes.....	29
Figura 3.22 Uploading Sketch. ....	30
Figura 3.23 Efectos primarios.....	30
Figura 3.24 Utilizando Smart Phone para cambiar secuencias.....	31

## ÍNDICE DE TABLAS

Tabla 1.1 Tipos de diodo .....	5
Tabla 1.2 Diferencias entre el diodo láser y el diodo LED.....	6
Tabla 1.3 Efectos biológicos en los ojos.....	7
Tabla 1.4 Clasificación del láser.....	8
Tabla 2.1 Porcentaje de respuestas .....	15
Tabla 3.1 Pruebas Realizadas .....	32
Tabla 3.2 Check List.....	33
Tabla 3.3 Reemplazo de caracteres.....	34
Tabla 3.4 Reemplazo de caracteres por imágenes .....	34
Tabla 3.5 Cambio de rotación de giro del mensaje.....	35
Tabla 3.6 Sobre montaje o distorsión de caracteres.....	35
Tabla 3.7 Cambio de secuencias .....	36
Tabla 3.8 Encendido y apagado de LEDs .....	36

## INTRODUCCIÓN

La empresa UNISCAN CIA. LTDA., se encuentra ubicada en las calles Ignacio Bossano E10-115 y Gral. Roca de la Parroquia Ñaquito (Centro-Norte) de la ciudad de Quito Ecuador, fue creada en Abril del 2004. Actualmente la empresa UNISCAN siendo una empresa ecuatoriana proveedora de servicios en el sector de la tecnología, brinda soluciones integrales hacia los clientes en el área de la educación y la tecnología.

Los medios de publicidad de la empresa UNISCAN son limitados en lo que se refiere a la presentación de las marcas que se comercializan, esto hace que los clientes en general desconozcan las distintas firmas que la empresa representa a nivel nacional; por lo que los clientes se limitan a adquirir lo que conocen y en muchos de los casos no se enteran de promociones, descuentos y/o servicios que la empresa brinda.

En Quito, las empresas por lo general utilizan medios publicitarios como vallas, gigantografías, etc. para publicitar ventas y servicios.

Se ha investigado mediante tutoriales en internet, en páginas web de países como Estados Unidos, Rusia, Colombia, Perú, Venezuela entre otros, que existen dispositivos electrónicos desarrollados mediante el efecto de persistencia de la visión en inglés Persistence Of Vision (efecto POV) que son utilizados como medio publicitario y desarrollados con PICs en lenguaje Micro C.

El dispositivo con efecto POV genera interés sobre las personas al presentar caracteres o letras, incluso palabras completas de manera alterna secuencial aparentando estar en el aire mientras gira sobre un motor o un elemento que se encuentre rodando a cierta velocidad adecuada, además de utilizar la tecnología Arduino para programar el dispositivo y crear los mismos efectos que los desarrollados con PICs.

## **OBJETIVOS**

### **Objetivo general**

Estudiar, diseñar e implementar un dispositivo electrónico con efecto POV con la finalidad de realzar la presentación de marcas que la empresa UNISCAN CIA. LTDA representa en el Ecuador.

### **Objetivos específicos**

- Estudiar la tecnología Arduino para el desarrollo del dispositivo electrónico con efecto POV.
- Diseñar un dispositivo electrónico con efecto POV usando tecnología Arduino de acuerdo a las características del proyecto.
- Implementar un dispositivo electrónico de efecto POV usando tecnología Arduino para la empresa UNISCAN CIA. LTDA.
- Validar mediante pruebas el correcto funcionamiento del dispositivo implementado.

# CAPÍTULO I

## FUNDAMENTACIÓN TEÓRICA

### 1.1 Introducción

Es necesario estudiar los conceptos que se describen a continuación para deducir el efecto de Persistencia de Visión, el cual se produce al hacer girar una columna de LEDs a cierta velocidad.

### 1.2 Antecedentes del efecto POV (Persistence of Vision)

El Efecto POV (Persistencia de la visión) fue un supuesto fenómeno visual descubierto por el inglés Peter Mark Roget que demuestra como una imagen permanece en la retina humana una décima de segundo antes de desaparecer por completo (Cambridge, 2009, pg. 131-140).

#### 1.2.1 Primeros Juguetes Ópticos

Hay muchas formas de demostrar la capacidad de retención en nuestros ojos a continuación se muestran algunos de los sistemas y juguetes más antiguos y conocidos del efecto de persistencia de visión.

##### 1.2.1.1 El taumatropo o roboscopio

Sistema que fue creado para confundir un par de imágenes, mediante un trozo de cartón de forma circular y un par de hilos retorcidos, que al estirar los hilos, el cartón gira rápidamente generando la ilusión de mezclar imágenes como en el siguiente ejemplo: De un lado se pinta una jaula y del otro lado un ave, que al girar rápidamente parecería que el ave se encuentra dentro de la jaula (Camara, 2006, pg. 1). Obsérvese figura 1.1.

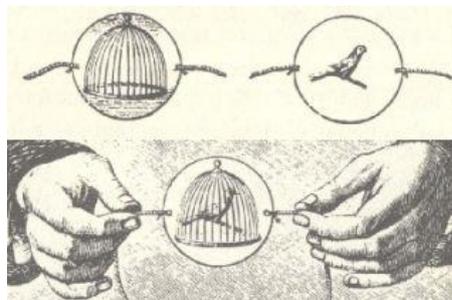


Figura 1.1 El taumatropo o roboscopio

Fuente: (Camara, 2006, pg.1)

### 1.2.1.2 El Estroboscopio

Es un sistema de destellos de luz variante que puede ser ajustado desde un segundo hasta más de 300, por ejemplo; un disco que gira rápidamente mediante este sistema parecería que el disco estuviera en reposo.

De igual forma este proceso puede ser invertido al hacer girar varias imágenes continuas y calibrar el destello de acuerdo al movimiento de cada imagen, las cuales fingirían un movimiento continuo. (Fuentes, 2009, pg. 1) Ver figura 1.2.

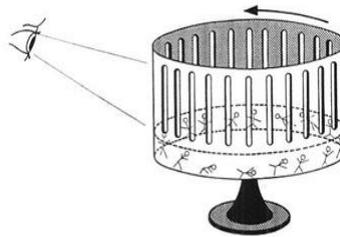


Figura 1.2 El Estroboscopio

Fuente: (Fuentes, 2009, pg.1)

### 1.2.1.3 El Zootropo

Consiste en un tambor de forma cilíndrica con ranuras en cuyo interior se coloca una tira de papel de igual tamaño que la circunferencia del tambor, en la tira de papel se dibujan las figuras a mostrar. Al hacer girar el tambor se puede observar a través de las ranuras como las figuras dibujadas adquieren movimiento. (Camara, 2006, pg. 1). Ver figura 1.3.



Figura 1.3 El Zootropo

Fuente: (Camara, 2006, pg.1)

Mediante estos primeros juguetes desarrollados se logró determinar el efecto de persistencia de la visión conocido en la actualidad como efecto POV.

El autor Guent en su libro de Persistencia Retiniana afirma que “El ojo hace que cualquier imagen vista se grabe en la retina por un corto tiempo (durante aproximadamente 0,1 segundos). Por ejemplo, al colocar un objeto frente a la vista y retirarlo rápidamente, cualquier persona que estuviera mirándolo lo seguirá viendo,

porque el cerebro retiene la impresión de la luz. Esta impresión es notoria al mirar por un momento una fuente de luz como por ejemplo, al hacer girar un fósforo se puede lograr que éste parezca formar un círculo de fuego en el aire, al igual que con un LED encendido (Guent, 2011, pg. 12).

Si ha dicho LED o a un conjunto de LEDs se los activa de manera apropiada en tiempo y forma, se obtendrá imágenes que parecerían estar flotando en el aire. En el caso de los carteles, una línea (o columna) se enciende a la vez y su traslación rápida y continua, con el encendido apropiado va formando el mensaje en forma horizontal.

### 1.3 LEDs y Láser

Los diodos emisores de luz o LEDs hoy en día se utilizan ampliamente en casi todas las áreas de la electrónica principalmente como dispositivos indicadores de pantalla, debido a que son resistentes, más fiables que las lámparas incandescentes de tipo filamento y con un consumo de energía menor a los focos normales (Boylestad, R y Nashelsky L, 2009, pg. 41-48).

La tabla 1.1, muestra las caídas de voltaje de varias clases de LEDs.

Tipo de diodo	Diferencia de potencial típica (V)	Valor de corriente para la intensidad luminosa (mA)
Rojo de bajo brillo	1.7	10
Rojo de alto brillo, alta eficiencia y baja corriente	1.9	20
Naranja y amarillo	2	10
Verde	2.1	10
Blanco brillante, verde brillante y azul	3.4	20
Azul brillante y LED especializados	4.6	20

Tabla 1.1 Tipos de diodo

Fuente: Investigador

#### 1.3.1 Diodos LED: ¿Cómo funcionan?

Dentro de una cubierta de plástico de un diodo LED hay un material semiconductor. Cuando se aplica una pequeña corriente eléctrica, emite luz, sin producir calor y con un

color definido. El color puede ser incluso invisible para el ojo humano, como los LED infrarrojos que hay en el mando a distancia del televisor (Boylestad, R y Nashelsky L, 2009, pg. 41-48).

### 1.3.2 Funcionamiento del diodo láser

Los Diodos láser (Light amplification by stimulated emission of radiation) luz amplificada por el principio de emisión estimulada, surge cuando un fotón induce a un electrón que se encuentra en un estado excitado a pasar al estado de reposo, este proceso está acompañado con la emisión de un fotón, con la misma frecuencia y fase del fotón estimulante. Esto se logra con una polarización directa de la unión, y por el otro una cavidad resonante, la cual posibilita tener una trayectoria de retroalimentación positiva facilitando que se emitan más fotones de forma estimulada y se seleccione ciertas longitudes de onda haciendo más angosto al espectro emitido. Se usan en productos de consumo y comunicaciones de banda ancha hoy en día utilizados en la fibra óptica (Malvino, 2006, pg. 195-198). La tabla 1.2 indica las diferencias entre los LEDs y láser.

### 1.3.3 Diferencias entre el diodo láser y el diodo LED

LÁSER	LED
Más rápido.	Mayor estabilidad térmica.
Potencia de salida mayor.	Menor potencia de salida, mayor tiempo de vida.
Emisión coherente de luz.	Emisión incoherente.
Construcción es más compleja.	Más económico.
Acoplado para sistemas de telecomunicaciones.	Fibras ópticas en distancias cortas de transmisión.
Modulación a altas velocidades, hasta GHz.	Velocidad de modulación hasta 200MHz.

Tabla 1.2 Diferencias entre el diodo láser y el diodo LED.

Fuente: Investigador

### 1.3.4 Láseres riesgo en su utilización efectos biológicos

Los órganos humanos que pueden resultar dañados en una exposición a radiación láser son los ojos y la piel. La gravedad de la lesión dependerá de la longitud de onda del

láser y del nivel de exposición alcanzado. En los ojos, el tipo de lesión producida varía con la longitud de onda de la radiación; observar tabla 1.3 (Robledo, 2013, pg. 567-570).

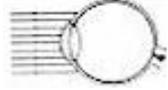
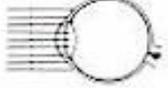
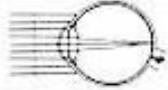
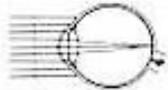
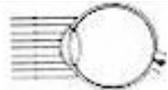
REGIÓN DEL ESPECTRO	ABSORCIÓN DEL OJO	MÁXIMA ABSORCIÓN EN	LESIÓN PRODUCIDA
UV-C y UV-B 200 a 315 nm		Córnea	Fotoqueratitis.
UV-A 315 a 400 nm		Cristalino	Catarata fotoquímica.
Visible 400 a 780 nm		Retina	Lesiones retinianas fotoquímicas y térmicas.
IR-A 780 a 1400 nm		Retina	Lesión térmica en la retina.
IR-B e IR-C 1400 a 1 mm		Córnea	Catarata térmica quemadura corneal.

Tabla 1.3 Efectos biológicos en los ojos.

Fuente: (Robledo, 2013, pg. 570)

El autor Robledo en su libro Normas Láseres y Clasificaciones afirma que “En el caso de la piel, la profundidad de penetración de un haz láser variará también con la longitud de onda” (Robledo, 2013, pg. 571).

### 1.3.5 Clasificación de los láseres

El autor Robledo en su libro Normas Láseres y Clasificaciones afirma “Los láseres no forman un sólo grupo al cual se apliquen límites de seguridad comunes, ya que los riesgos que se derivan de su uso varían notablemente debido a los amplios intervalos posibles para la longitud de onda, la potencia o energía y las características de emisión continua o de impulsos de un haz láser” (Robledo, 2013, pg. 572-574).

Se muestra de forma resumida en la tabla 1.4 la clasificación de la norma CEI-825-1984 (Comité Electrónico Internacional), que agrupa los láseres en 4 clases generales, niveles de emisión accesible máximos permitidos dentro de una clase determinada (Robledo, 2013, pg. 572-574). (Tablas I a IV de la Norma CEI-825-84).

Clase 1	Seguros en todas las condiciones; incluyendo el uso de instrumentos ópticos de visión directa.
Clase 1M	Longitud de onda ( $\lambda$ ) entre 302,5 y 400nm son seguros en condiciones de uso razonable, pero pueden ser peligrosos si se emplean instrumentos ópticos de visión directa.
Clase 2	Longitud de onda comprendido entre 400 y 700 nm. La protección ocular se consigue normalmente por las respuestas de aversión, incluido el reflejo perpetral.
Clase 2M	Longitud de onda comprendido entre 400 y 700 nm. La protección ocular se consigue normalmente por las respuestas de aversión, incluido el reflejo perpetral. Pero la visión del haz puede ser peligrosa si se usan instrumentos ópticos.
Clase 3R	Longitud de onda entre 302.5 y 106nm, cuya visión directa del haz es potencialmente peligrosa. El límite de emisión accesible es menor que 5 veces el LEA de la clase 2 en el rango 400-700nm, y cinco veces menor el LEA de la clase 1 para otras longitudes de onda.
Clase 3B	Láseres cuya visión directa es siempre peligrosa (ejemplo. Dentro de la distancia nominal de riesgo ocular).
Clase 4	Pueden producir reflexiones difusas peligrosas. Pueden causar daños sobre la piel y pueden también constituir un peligro de incendio. Su utilización precisa de extrema precaución.

Tabla 1.4 Clasificación del láser.

Fuente: Investigador

#### 1.4 Motor

Un motor es la parte sistemática de un dispositivo, ver figura 1.4, utilizando el tipo adecuado se obtiene la velocidad necesaria para que el dispositivo funcione correctamente.

Existen varios tipos, los más comunes son los siguientes:

- Motores Térmicos
- Motores de Combustión Interna
- Motores de Combustión externa.
- Motores Eléctricos



Figura 1.4 Ejemplo de Motor DC, 0 a 5000 revoluciones por minuto

Fuente: (Awesome, 2011, pg.1)

## 1.5 Tecnología Arduino

La tecnología Arduino es una plataforma de electrónica abierta para la creación de prototipos basada en software y hardware flexibles y fáciles de usar, que difiere de cierta manera a la tecnología utilizada con microcontroladores o PICs comúnmente llamados, para ello se realiza un breve repaso de esta tecnología.

### 1.5.1 Hardware Arduino

Las placas Arduino son muy versátiles para el desarrollo de varios tipos de proyectos, es por ello que se mencionan ciertos modelos de placas de las más conocidas y utilizadas en el campo electrónico tales como: Arduino UNO, módulo Bluetooth HC-06, y finalmente se opta por la placa Arduino Leonardo gracias a su capacidad de memoria.

A continuación se presentan las características y especificaciones de las placas Arduino y módulo Bluetooth a utilizar.

- Arduino UNO

Placa electrónica basada en el microcontrolador ATmega328, cuenta con 14 entradas/salidas digitales, de los cuales 6 pueden ser utilizados como salidas PWM, 6 entradas analógicas, un oscilador de cristal de 16 MHz, una conexión USB, un conector de alimentación, y un botón de reinicio, contiene todo lo necesario para apoyar al microcontrolador. Para que inicie simplemente basta conectar la placa a un ordenador con un cable USB, con un adaptador AC-DC de 7V-12V o una batería. (Oxer, 2009, pg. 1) Ver figura 1.5.



Figura 1.5 Placa Arduino UNO (Frente y posterior)

Fuente: (Oxer, 2009, pg.1)

- Arduino Leonardo

Placa electrónica basada en el ATMEGA32U4 de la misma forma cuenta con 20 entradas/salidas digitales (de los cuales 7 se pueden utilizar como salidas PWM y 12 entradas como analógicas), un 16 MHz oscilador de cristal, una conexión micro USB, un conector de alimentación, y un botón de reset. La placa se la puede poner en funcionamiento al conectar a un ordenador con un cable USB, alimentarla con un adaptador o una batería de 9V (Oxer, 2009, pg. 1).

La placa Arduino Leonardo difiere de todas anteriores en que el ATMEGA32U4 ha incorporado en la comunicación USB, eliminando la necesidad de un procesador secundario. Esto permite que el Leonardo aparezca a una computadora conectada como un ratón y el teclado, además de un puerto serie / COM virtual. (Oxer, 2009, pg. 1) Ver figura 1.6.



Figura 1.6 Placa Arduino Leonardo (frente y posterior)

Fuente: (Oxer, 2009, pg.1)

- Modulo Bluetooth HC-06

El módulo bluetooth HC-06 de Arduino es una placa pequeña, completa, de fácil uso y configuración basada en la comunicación inalámbrica, lamentablemente la poca

existencia de información sobre la placa crea un entorno un tanto complicado inicial de utilizar, sin embargo las ventajas principales del módulo HC-06 además de su tamaño pequeño son las características de transmisión y recepción que le brindan un amplio alcance debido a que se trata de un sistema local Bluetooth y su bajo consumo de corriente que posee tanto en funcionamiento como en modo de espera, otra característica interesante es que una vez emparejado con otro dispositivo mantiene en su memoria y no necesita de validación cada vez que se conecte al dispositivo (por defecto “1234”). (Peterson, 2013, pg. 1) Ver figura 1.7.



Figura 1.7 Placa Arduino Bluetooth HC-06 (frente y posterior)

Fuente: (Peterson, 2013, pg.1)

### 1.5.2 Software Arduino

La tecnología y su entorno de código abierto Arduino permiten escribir fácilmente en la plataforma Arduino y es muy simple cargar a la placa. Funciona en Windows, Mac OS X y Linux. El entorno está escrito en Java y basado en Processing, avr-gcc y otros programas también de código abierto (Banzi, 2005, pg. 1). Observar figura 1.8.

Structure	Variables	Functions
<ul style="list-style-type: none"> <li>- setup()</li> <li>- loop()</li> </ul> <p><b>Control Structures</b></p> <ul style="list-style-type: none"> <li>- if</li> <li>- if...else</li> <li>- for</li> <li>- switch case</li> <li>- while</li> <li>- do... while</li> <li>- break</li> <li>- continue</li> <li>- return</li> </ul>	<p><b>Constants</b></p> <ul style="list-style-type: none"> <li>- HIGH   LOW</li> <li>- INPUT   OUTPUT   INPUT_PULLUP</li> <li>- LED_BUILTIN</li> <li>- true   false</li> <li>- integer constants</li> <li>- floating point constants</li> </ul> <p><b>Data Types</b></p> <ul style="list-style-type: none"> <li>- void</li> <li>- boolean</li> <li>- char</li> <li>- unsigned char</li> </ul>	<p><b>Digital I/O</b></p> <ul style="list-style-type: none"> <li>- pinMode()</li> <li>- digitalWrite()</li> <li>- digitalRead()</li> </ul> <p><b>Analog I/O</b></p> <ul style="list-style-type: none"> <li>- analogReference()</li> <li>- analogRead()</li> <li>- analogWrite() - PWM</li> </ul> <p><b>Due only</b></p> <ul style="list-style-type: none"> <li>- analogReadResolution()</li> <li>- analogWriteResolution()</li> </ul>

Figura 1.8 Lenguaje de Referencia Arduino

Fuente: (Peterson, 2013, pg.1)

## CAPÍTULO II

### DESCRIPCIÓN DEL PROCESO INVESTIGATIVO REALIZADO

- **Problema Principal**

Después de haber realizado el problema investigado se determina que la empresa UNISCAN CIA. LTDA no cuenta con un medio de presentación electrónico con efecto de Persistencia de Visión de las marcas que representa en el Ecuador.

- **Idea de los Objetivos**

El objetivo general se enmarca en la necesidad de la empresa UNISCAN CIA LTDA de contar con un dispositivo electrónico como medio de presentación digital visual de las marcas que representa en el Ecuador.

El objetivo de estudiar la tecnología Arduino es porque permite la integración con nuevas herramientas tecnológicas desconocidas en el desarrollo del dispositivo de efecto POV.

El proceso de diseño es necesario porque se fusiona el uso de las tecnologías escogidas con los requerimientos de la empresa solicitante.

Es importante la implementación del dispositivo en la empresa UNISCAN. CIA. LTDA porque el cliente no espera que sólo quede como un diseño, sino que también pueda apreciar de forma física como el control del efecto POV crea el mensaje que aparece y se desplaza lentamente por todo el contorno del giro.

La validación del sistema permite asegurar el correcto funcionamiento del dispositivo.

- **Hipótesis**

De acuerdo al plan realizado, la hipótesis del proyecto es que: cuando se implemente el dispositivo electrónico de efecto POV como presentación de marcas representadas por la empresa UNISCAN, habrá un mayor conocimiento por parte del cliente de las marcas, descuentos y/o promociones que la empresa disponga.

De esta manera la variable independiente es la implementación del dispositivo de efecto POV y la variable dependiente será el mayor conocimiento del cliente externo o interno de la empresa.

- **Justificación Teórica**

La fundamentación del proyecto se basa en:

LEDs o láser que son ubicados en una única columna, y al ser encendidos de manera sincronizada y controlada crean mejores efectos de iluminación, y en el caso del proyecto, forma mensajes o palabras completas al rotar toda la columna en su propio eje.

La tecnología del software Arduino está basada en una plataforma de software libre bastante amigable para el desarrollador, el hardware de las placas Arduino UNO o Arduino Leonardo son las más accesibles del mercado, tienen memoria suficiente para almacenar caracteres y secuencias importantes dentro de la programación para crear el efecto POV, son versátiles porque son placas electrónicas que cuentan con todo lo indispensable para controlar el dispositivo, además disponen de puertos de salida/entrada necesarios para los LEDs. Sin embargo existen otro modelo de placas Arduino que tienen mucha más capacidad de memoria y que se podrían utilizar para almacenar mayor información, secuencias y/o procesos en el control programado.

Módulo Bluetooth HC-06 que tiene un alcance de aproximadamente 10 metros y es capaz de crear un puerto serial COM virtual entre el dispositivo de efecto POV y otro equipo que tenga la capacidad de conectarse vía bluetooth, para la recepción de información de comandos previamente programados en el software Arduino.

- **Metodologías utilizadas**

El desarrollo del dispositivo de efecto POV se divide en tres fases descritas a continuación:

En la primera fase del Trabajo de Titulación se utilizó el método de análisis y síntesis para recopilar toda la información necesaria para dicho trabajo, la misma que se encuentra contenida en el capítulo 1.

En la segunda fase se utilizó el método de modelación para realizar el diseño adecuado del dispositivo de efecto POV.

En la tercera fase de implementación, se utiliza el método experimental para realizar las pruebas necesarias de funcionamiento y verificación del proyecto.

- **Técnica de investigación**

Para este proyecto se utilizó la entrevista como técnica de investigación, a continuación se presenta el formato de la entrevista en la figura 2.1, y las entrevistas realizadas se puede observar en el anexo 1.

UNIVERSIDAD TECNOLÓGICA ISRAEL  
CARRERA DE ELECTRONICA Y TELECOMUNICACIONES

Entrevista al personal de la Empresa UNISCAN CIA. LTDA.

Nombre:.....

Cargo:.....

Fecha:.....

1. ¿Qué opina usted sobre la implementación del nuevo dispositivo de publicidad de efecto POV?

Bueno  Regular  Malo

Porque?.....

.....

2. ¿Cree usted que el dispositivo ayudó al cliente en la publicidad de las marcas representadas por su empresa?

Bueno  Regular  Malo

Porque?.....

.....

3. ¿Cree usted que se pueda implementar el dispositivo en las oficinas de la ciudad de Guayaquil tomando en cuenta el resultado obtenido en la matriz principal de Quito?

Si  No  Tal vez

Porque?.....

.....

4.- ¿Está usted familiarizado con el manejo del equipo? ¿Qué ventajas observa sobre el dispositivo implementado?

Si  No  Tal vez

Porque?.....

.....

5.- ¿Cree usted que el nuevo dispositivo de efecto POV mantiene una estética adecuada para la empresa?

Si  No  Tal vez

Porque?.....

.....

Figura 2.1 Formato Entrevista.

Fuente: Investigador

- **Análisis de los resultados de la entrevista**

A continuación en la tabla 2.1 se presentan los resultados de las entrevistas, de acuerdo a cada pregunta realizada.

Pregunta	Opciones de respuesta en %.					
	Bueno	Regular	Malo	Si	No	Tal vez
¿Qué opina usted sobre la implementación del nuevo dispositivo de publicidad de efecto POV?	100%	0%	0%	–	–	–
¿Cree usted que el dispositivo ayudó al cliente en la publicidad de las marcas representadas por su empresa?	95%	5%	0%	–	–	–
¿Cree usted que se pueda implementar el dispositivo en las oficinas de la ciudad de Guayaquil tomando en cuenta el resultado obtenido en la matriz principal de Quito?	–	–	–	85%	0%	15%
¿Está usted familiarizado con el manejo del equipo? ¿Qué ventajas observa sobre el dispositivo implementado?	–	–	–	50%	40%	10%
¿Cree usted que el nuevo dispositivo de efecto POV mantiene una estética adecuada para la empresa?	–	–	–	80%	0%	20%

Tabla 2.1 Porcentaje de respuestas

Fuente: Investigador

Mediante los gráficos estadísticos de la figura 2.2 se puede observar el porcentaje total de las personas entrevistadas. Por tanto se puede apreciar que la implementación del proyecto tiene un margen aceptable dentro de una empresa, sin embargo existe incertidumbre en cuanto a la manipulación del dispositivo por lo que se opta por la necesidad de agregar un manual de usuario, el mismo que se lo puede observar en anexo 2.

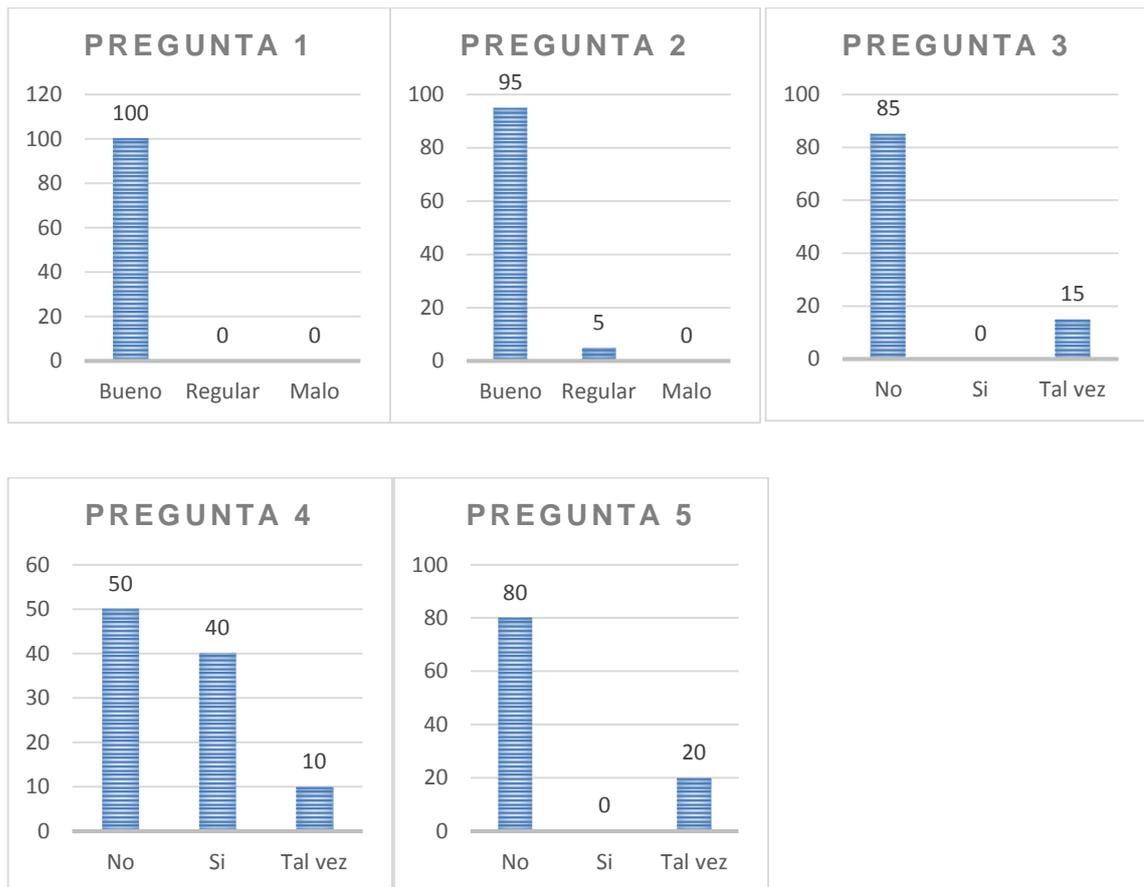


Figura 2.2 Porcentaje de entrevistas por cada pregunta

Fuente: Investigador

- **Resultados esperados del proyecto**

Se espera que una vez que se haya implementado este proyecto los clientes tengan más conocimiento de las marcas que representa la empresa en el país, y posteriormente el dispositivo sea aceptado en las diferentes empresas de la ciudad.

## **CAPÍTULO III**

### **PRESENTACIÓN DE RESULTADOS**

#### **3.1 Introducción**

En el presente capítulo se presenta la propuesta de solución al problema presentado por la empresa UNISCAN CIA. LTDA mediante los procesos de: simulación, montaje e implementación del proyecto; se incluyen diagramas de bloques y funcionamiento; presentación de la plataforma Arduino donde se realiza la respectiva programación y control de encendido de los Leds para generar el efecto POV.

#### **3.2 Propuesta de solución del problema**

Como propuesta se ha diseñado el proyecto de acuerdo a ciertos aspectos que determinan la estética del producto final, también la dimensión de sus componentes para que no tenga inconvenientes al momento de rotar y que los LEDs deben encenderse con adecuada sincronía y secuencia con la finalidad de generar una serie de patrones parecidos a toda una matriz de un display.

#### **3.3 Diagrama de bloques**

En los diagramas siguientes figura 3.1 y figura 3.9 se muestran el desarrollo y funcionamiento tanto en hardware como en software del dispositivo con efecto POV.

##### **3.3.1 Diagrama de Bloques de Hardware del dispositivo POV**

Dentro del funcionamiento del dispositivo lo más indispensable es que la estructura gire a la velocidad adecuada, puesto que al tener algún problema de lentitud o aumento de giro causa que el dimensionamiento de espacios entre caracteres y la presentación de imagen se sobreponga entre sí; cuando ocurre este inconveniente no se logra observar lo presentado. El diagrama de bloques de la figura 3.1 muestra las etapas de funcionamiento del dispositivo POV.

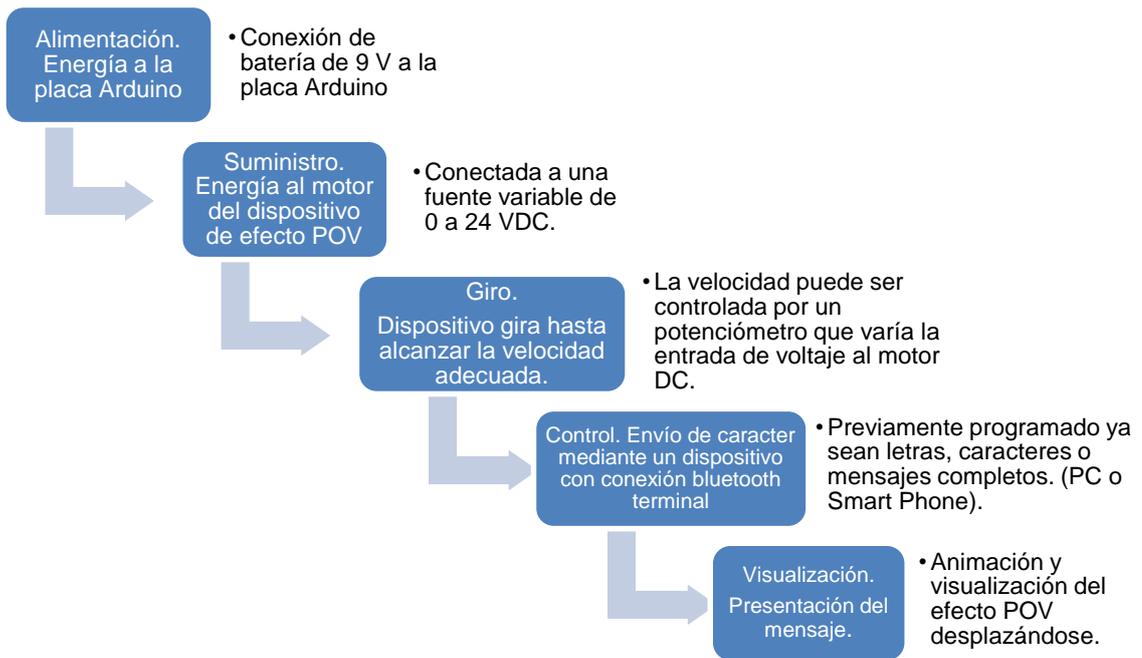


Figura 3.1 Diagrama de bloques funcionamiento mecánico.

Fuente: Investigador

Las etapas que conforman el diseño de hardware son las siguientes:

- **Alimentación de energía a la placa Arduino**

Se requiere de una batería de 9 Voltios DC para alimentar la placa Arduino, con el fin de energizar el dispositivo de efecto POV.

- **Etapas de Suministro de energía al motor DC**

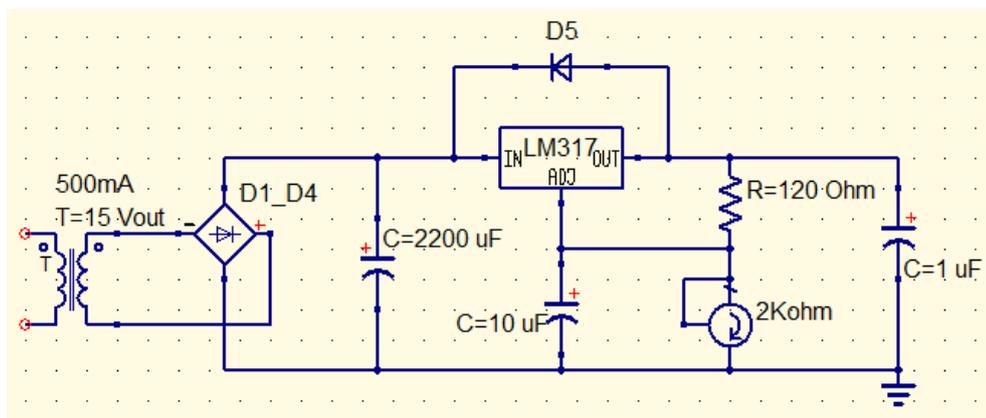


Figura 3.2 Fuente regulable

Fuente: Investigador

En la figura 3.2 la etapa la integra un transformador de voltaje de 110V AC a 15V AC el cual es rectificado por un puente de diodos, de esta manera no pueda sufrir alguna

variación que dañe a los otros componentes, luego la energía llega al capacitor que elimina los picos de la señal para que no tenga oscilaciones, pasa al regulador de tensión LM317 el mismo que hace que la señal de entrada no afecte a la tensión de salida cuando ésta aumente o disminuya, al final se encuentra un potenciómetro que hace que varíe el voltaje continuo, que posteriormente se acopla a un motor de DC y ayuda a controlar la velocidad de giro del motor.

- **Etapa de movimiento (giro de motor y componentes)**

Al realizar el giro del motor se debe tomar en cuenta las dimensiones y peso de los componentes como son: tarjeta Arduino, tarjeta de circuito impreso de leds, batería, módulo Bluetooth para que no existan vibraciones que afecten a la sincronización del encendido y apagado de los Leds. Ver figura 3.3.

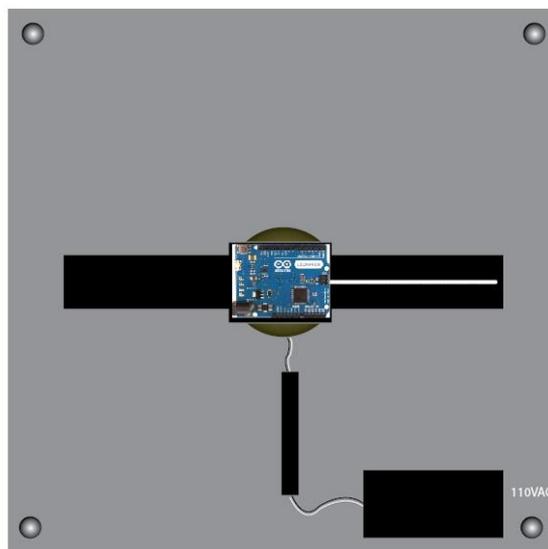


Figura 3.3 Etapa de movimiento.

Fuente: Investigador

- **Etapa de control**

En la etapa de control se encuentra la programación acoplada a la placa bluetooth HC-06 desde un computador portátil o un dispositivo Smart phone (con aplicación adecuada para el envío del dato hacia la placa Bluetooth HC-06), de esta forma se puede controlar cada mensaje a visualizar; la información de programación se encuentra en anexo 1.

- **Etapa de visualización de datos**

En la etapa de visualización de datos se ubican los LEDs en forma simétrica vertical a distancia de 1 centímetro entre LEDs. Tal y como se muestra en la figura 3.4.

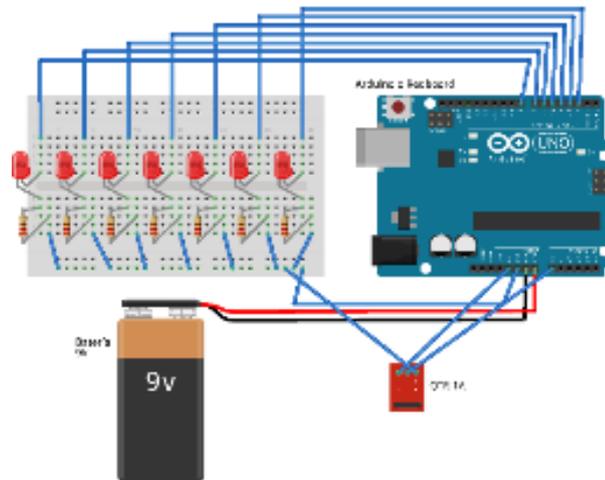


Figura 3.4 Ubicación de Leds.

Fuente: Investigador

Al rotar el motor y sus componentes crean la imagen de demostración, observe la figura 3.5.



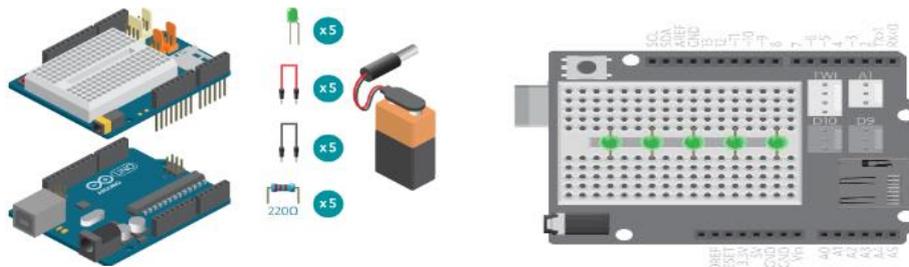
Figura 3.5 Imagen generada.

Fuente: Investigador

### 3.4 Simulación y montaje de componentes en protoboard

El gráfico de la figura 3.6 se encuentra desarrollado en el software llamado Fritzing (aplicación para graficar placas Arduino y protoboards especialmente) que muestra los componentes que se utiliza para realizar las primeras pruebas del proyecto.

- a) Componentes básicos para el montaje en el sistema Arduino, figura basada en la aplicación Fritzing, obsérvese figuras 3.6 a) y 3.6 b).



a) Gráfico de placa Arduino      b) Gráfico de protoboard básico.

Figura 3.6 Simulación de componentes.

Fuente: Investigador

- b) En el desarrollo de las conexiones de simulación se ubican los componentes en el protoboard de acuerdo a lo realizado en las pruebas iniciales de funcionamiento del dispositivo desarrollado. Obsérvese figura 3.7.

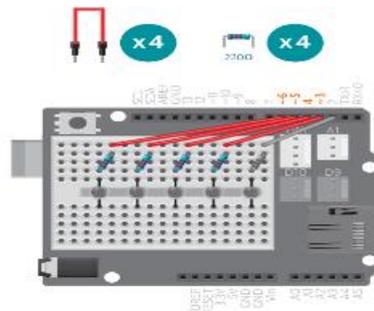


Figura 3.7 Unión de terminales

Fuente: Investigador

- c) En la figura 3.8 se muestra el gráfico de la emulación del dispositivo de efecto POV conectado con la pc del usuario, para ello se debe tener el código fuente creado en una carpeta, la cual se procede a abrir desde la plataforma Arduino y se compila directo a la placa mediante un medio guiado (cable USB 2.0).

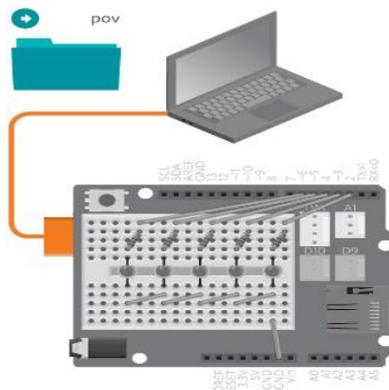


Figura 3.8 Placa Arduino y componentes para la simulación.

Fuente: Investigador

### 3.4.1 Diagrama de Bloques Software del lenguaje Arduino

Las fases del diagrama de bloques de la figura 3.9 representa el software del dispositivo del lenguaje en la plataforma Arduino.

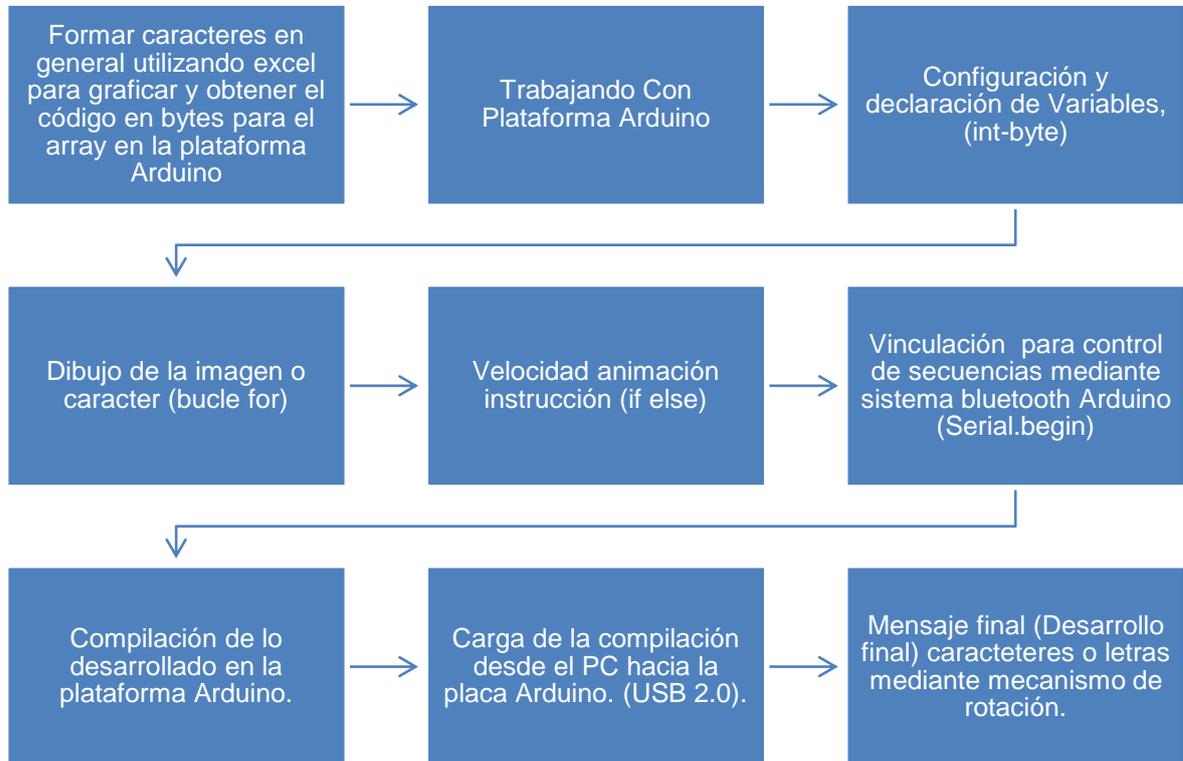


Figura 3.9 Diagrama de bloques programación Arduino.

Fuente: Investigador

Las etapas que conforman el software del dispositivo son las siguientes:

- **Formación de caracteres en Excel**

Para la creación de los patrones y formación de letras se ha realizado varias hojas de cálculo las cuales permiten crear los caracteres según como se desea apreciar en el giro, obsérvese la siguiente figura 3.10 que es un ejemplo práctico para la formación de letras mayúsculas, cada letra en una matriz de 7x5.



Figura 3.10 Letras mayúsculas 7x5.

Fuente: Investigador

En la figura 3.11 se realiza el giro de caracteres según la ubicación de la placa, en el caso del proyecto los LEDs están colocados de forma horizontal en la base de giro del motor y al utilizar este método es necesario girar la imagen para que al momento de programar en Arduino no se cometa el error de formación de caracteres.

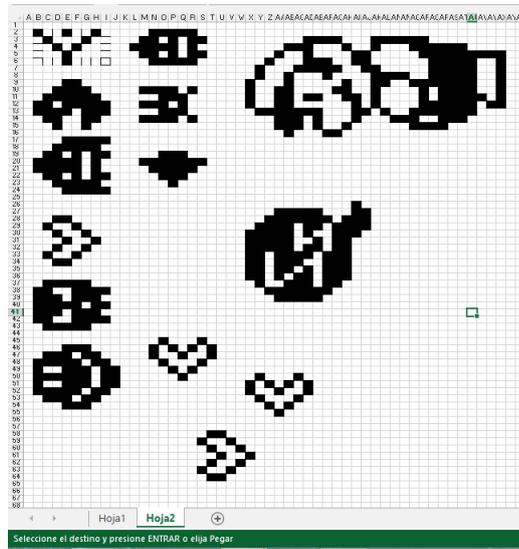


Figura 3.11 Giro de patrón de caracteres

Fuente: Investigador

Posterior a esto se transforma los caracteres o imágenes creadas a ceros y unos (bits), para que el lenguaje Arduino lo reconozca sin dificultad observe la figura 3.12.



Al realizar la descarga del software según lo indicado en la figura 3.6, se procede a la instalación de la plataforma, la cual se utiliza en todo el proceso de desarrollo de control de Leds y de compilación del lenguaje; se ha instalado el software Arduino en el sistema operativo Microsoft Windows 7 de 32 bit. Obsérvese figura 3.14.



Figura 3.14 Iniciando la aplicación en Microsoft Windows 7 Enterprise de 32 bits.

Fuente: Investigador

Este es el estado inicial de la plataforma en la cual se desarrolla el lenguaje de los distintos tipos de control para presentar el efecto POV, ver figura 3.15.

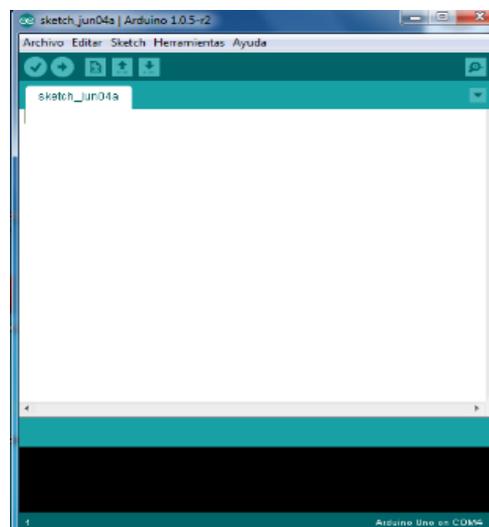


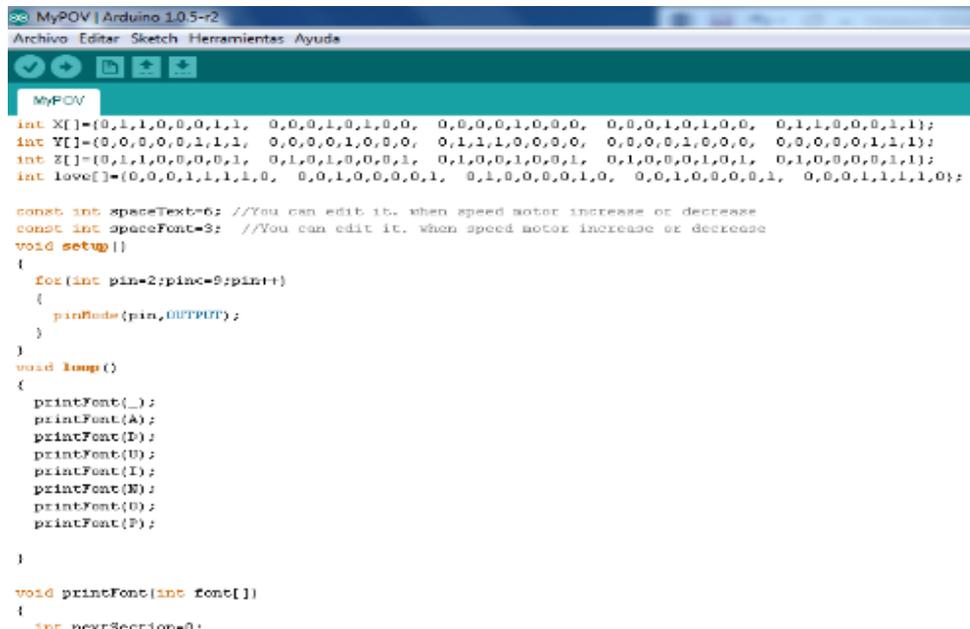
Figura 3.15 Acceso a la plataforma

Fuente: Investigador

- **Programando en la plataforma Arduino**

Al empezar a trabajar con la plataforma Arduino se inicia declarando las variables que se van a utilizar para ello se debe realizar un arreglo de bytes (array) los mismos que

contienen toda la información necesaria para el control del dispositivo de efecto POV. Se realizan las respectivas pruebas iniciales declarando variables y bucles, observar figura 3.16.



```
MyPOV | Arduino 1.0.5-r2
Archivo  Editor  Sketch  Herramientas  Ayuda

MyPOV

int X[]={0,1,1,0,0,0,1,1, 0,0,0,1,0,1,0,0, 0,0,0,0,1,0,0,0, 0,0,0,1,0,1,0,0, 0,1,1,0,0,0,1,1};
int Y[]={0,0,0,0,0,1,1,1, 0,0,0,0,1,0,0,0, 0,1,1,1,0,0,0,0, 0,0,0,0,1,0,0,0, 0,0,0,0,0,1,1,1};
int Z[]={0,1,1,0,0,0,0,1, 0,1,0,1,0,0,0,1, 0,1,0,0,1,0,0,1, 0,1,0,0,0,1,0,1, 0,1,0,0,0,1,1,1};
int love[]={0,0,0,1,1,1,1,0, 0,0,1,0,0,0,0,1, 0,1,0,0,0,0,1,0, 0,0,1,0,0,0,0,1, 0,0,0,1,1,1,1,0};

const int spaceText=6; //You can edit it. when speed motor increase or decrease
const int spaceFont=3; //You can edit it. when speed motor increase or decrease
void setup()
{
  for(int pin=2;pin<=9;pin++)
  {
    pinMode(pin,OUTPUT);
  }
}
void loop()
{
  printFont(_);
  printFont(A);
  printFont(D);
  printFont(U);
  printFont(I);
  printFont(N);
  printFont(O);
  printFont(P);
}

void printFont(int font[])
{
  int nextSection=0;
```

Figura 3.16 Declarando variables en el entorno Arduino

Fuente: Investigador

- **Desarrollo final de la programación en el lenguaje utilizando la tecnología Arduino**

En esta fase se implementa todas las librerías requeridas para el control total del dispositivo, se ha declarado las variables necesarias con sus respectivos bucles y secuencias, se ha optimizado la tecnología estándar bluetooth del módulo HC-06 para la recepción del carácter o mensaje a presentar. Toda la información de la programación puede encontrarse en el Anexo 3.

### 3.5 Desarrollo e implementación

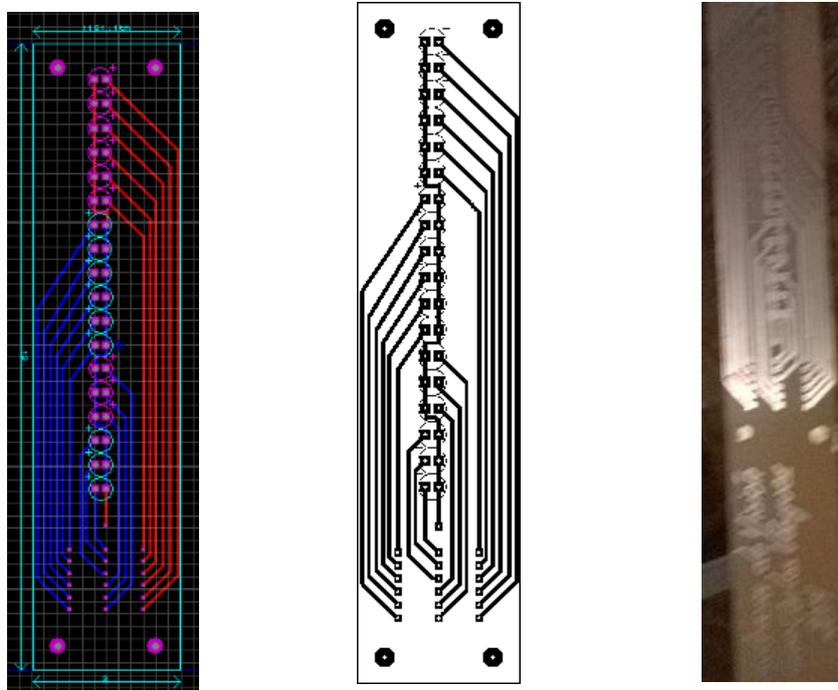
Los materiales utilizados para la implementación dispositivo son los que se muestran en la tabla 3.1.

Descripción	Cantidad
Motor de DC 24V	1
Arduino Leonardo	1
LEDs	18
Elementos para la fuente Variable de 0 V a 24 V.	1
Batería de 9 V	1
Cables y conectores	1 kit
Cable USB 2.0	1
Módulo Bluetooth HC-06	1
Estructura Acrílico	1
Base horizontal (madera)	1
Base para motor	1

Tabla 3.1 Materiales Utilizados

Fuente: Investigador

a) Desarrollo de la placa donde se ubican los Leds o Láser. Ver figura 3.17.

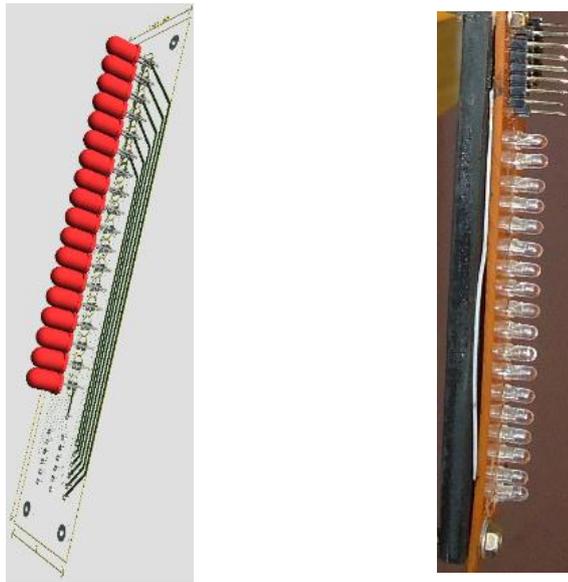


a) Desarrollo en Ares    b) Circuito impreso    c) PCB final.

Figura 3.17 Desarrollo de la placa electrónica para los Leds.

Fuente: Investigador

a) Montaje de Leds y conectores. Ver figura 3.18.



a) Visualización 3D

b) Visualización física

Figura 3.18 Montaje de Leds en la placa PCB.

Fuente: Investigador

a) Determinación de los pines en la placa Arduino para conectar con los respectivos Leds. Los pines cubren las entradas establecidas de los componentes. Ver figura 3.19.

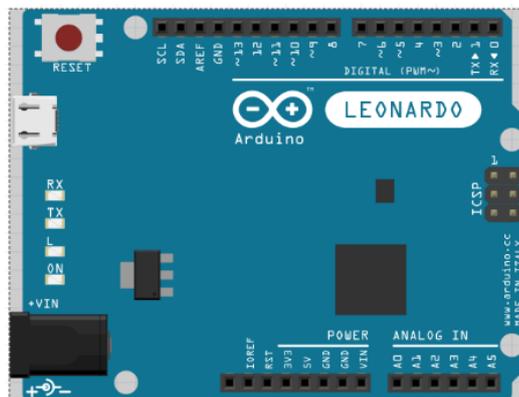


Figura 3.19 Identificación de los pines en Arduino Leonardo

Fuente: Investigador

- De los pines 2 al 13 digitales para la salida de los Leds
- De los pines A0-A5 salida analógica configurada para salida digital de acuerdo a lo diseñado.
- Entrada Vcc
- Entrada Gnd

- Pin 0 y 1 (Rx/Tx) para el módulo Bluetooth.
- b) Ubicación de los componentes en la base de madera, tarjeta Arduino, batería de 9 voltios (que es la encargada de energizar la placa Arduino para su funcionamiento), placa con LEDs y modulo bluetooth, debidamente distribuido su peso, en caso de tener una variación en el equilibrio causa grandes vibraciones al girar. Ver figura 3.20.

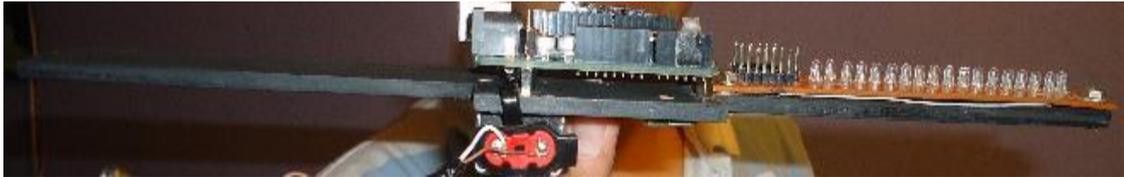


Figura 3.20 Ubicación y montaje de los componentes utilizados.

Fuente: Investigador

- c) Montaje en la estructura del motor debidamente ubicada y sujeta en el soporte acrílico rígido, junto a este soporte se encuentra la fuente variable de voltaje (caja plástica negra) Ver figura 3.21.

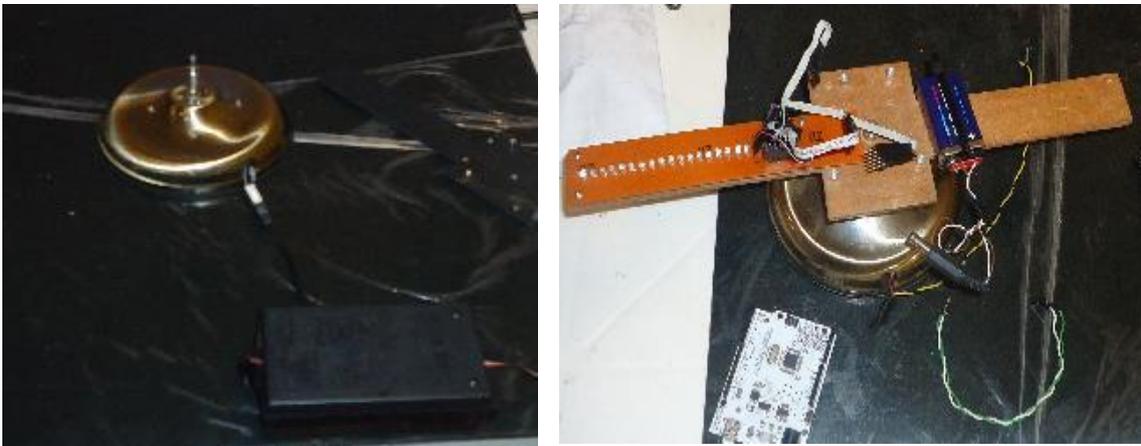


Figura 3.21 Ubicación total con todos los componentes.

Fuente: Investigador

- d) Compilación y carga de la programación desarrollada en Arduino, de acuerdo a la programación final desarrollada. Ver figura 3.22.



Figura 3.22 Uploading Sketch.

Fuente: Investigador

e) Efectos iniciales de prueba del dispositivo de efecto POV. Ver figura 3.23.



Figura 3.23 Efectos primarios

Fuente: Investigador

- f) Uso de Smart phone para enviar la primera secuencia de presentación del mensaje a visualizar. Ver figura 3.24.

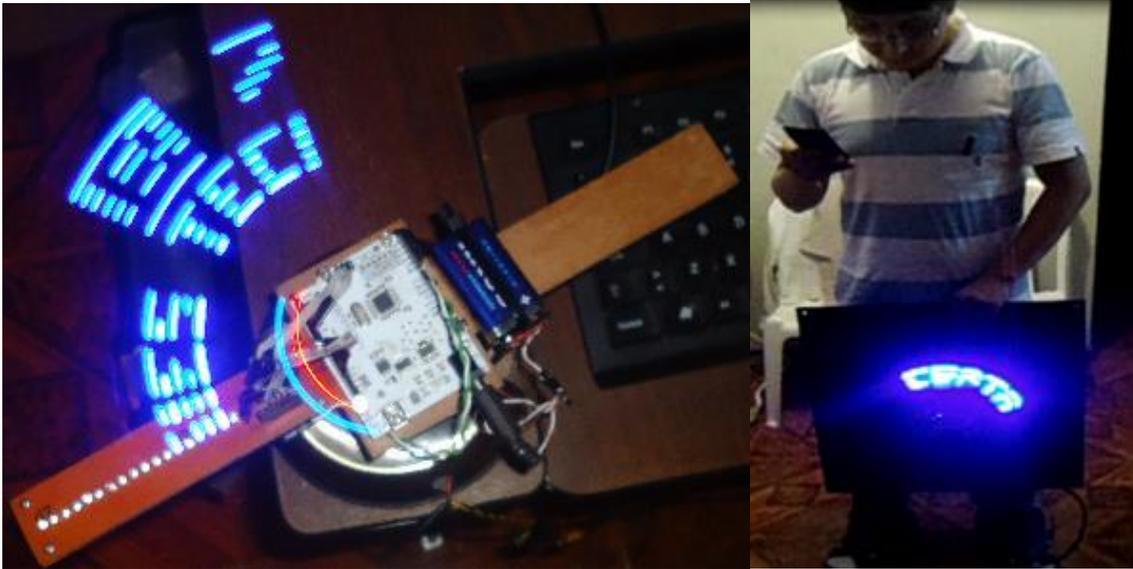


Figura 3.24 Utilizando Smart Phone para cambiar secuencias.

Fuente: Investigador

### 3.6 Validación del sistema

Se realizaron pruebas simulando la presentación real del dispositivo, para ello se ubicaron todos los componentes de acuerdo al diseño establecido en un protoboard y en estructuras giratorias creadas de forma espontánea.

#### 3.6.1 Pruebas experimentales

De acuerdo a las pruebas experimentales realizadas, se obtiene que la estructura del dispositivo debe ser la adecuada y equilibrada, para que no genere vibraciones y esto altere la visualización de las imágenes mostradas, a continuación se observa la tabla 3.1 que indica que tipos de pruebas se realizó durante todo el proceso de desarrollo del proyecto.

PRUEBAS REALIZADAS						
#	Motor adecuado	Sistemas de Giro	Material de Estructuras	Placas Arduino	Control con Arduino	Arrays en bits
1	Motor de ventilador de CPU	Vertical	Metal	Arduino UNO	En esta fase se realizó aproximadamente 95 tipos de pruebas para lograr obtener el control, la secuencia, la comunicación Bluetooth y finalmente la sincronización correcta del dispositivo de efecto POV.	Fast LCD
2	Motor de AC	Esférico	Acrílico	Arduino Nano		GLCD font Creator
3	Motor de 9 Voltios	Horizontal	Madera	Arduino mini		Bitmap2LCD
4	Motor de 12 Voltios			Arduino Leonardo		Aplicación compilada en Java
5	Motor de 24 Voltios 0-3000rpm					Hoja de cálculo

Tabla 3.1 Pruebas Realizadas

Fuente: Investigador

### 3.6.2 Pruebas finales

Cuando se logró corregir las variantes se procedió con la implementación en la estructura real para llevar a cabo la puesta en marcha del proyecto y realizar las pruebas finales tales como:

Prueba 1: Reemplazar caracteres o palabras de acuerdo a cada tamaño de secuencia (mensaje).

Prueba 2: Agregar imágenes creadas en la programación, ejemplo corazón, rayo, comecocos, calavera, cara feliz entre otras.

Prueba 3: Cambio de rotación de giro de palabra de izquierda a derecha o viceversa.

Prueba 4: Mejoramiento o distorsión (sobre montaje) de la imagen de acuerdo al ajuste de velocidad del motor.

Prueba 5: Cambio de secuencias establecidas mediante Smart Phone o PC con soporte hyperterminal para bluetooth.

Prueba 6: Encendido y apagado de secuencias al presionar los botones programados para realizar la tarea.

### 3.7 Evaluación Técnica

Se llevó a cabo el siguiente CHECK LIST del funcionamiento del dispositivo, observar tabla 3.2.

Procedimiento	Funcionamiento		
	Correcto	Incorrecto	Observaciones
Reemplazo de caracteres mediante plataforma Arduino.	X		Se utiliza cable USB para compilar el o los cambios de los caracteres.
Reemplazo de imágenes desarrolladas.	X		Se utiliza cable USB para compilar el o los cambios de las imágenes creadas.
Variación de giro de Izquierda a derecha y viceversa.	X		Se realiza mediante ajuste de velocidad del motor.
Distorsión o mejoramiento de caracteres o imágenes por variación de velocidad.	X		Se realiza mediante ajuste de velocidad del motor.
Cambio de secuencias	X		Se observa correctamente, se utiliza dispositivo Smart Phone o PC con soporte hyperterminal para bluetooth.
Encendido y Apagado de LEDs	X		Se realiza la tarea mediante teclado programado en software Arduino.

Tabla 3.2 Check List

Fuente: Investigador

### 3.8 Resultados

**Prueba 1: Reemplazo de caracteres o palabras de acuerdo a cada tamaño de secuencia.**

La tabla 3.3 muestra los resultados de las pruebas del reemplazo de caracteres en el dispositivo de efecto POV, para tal efecto se hicieron 7 cambios en las secuencias.

Secuencia	Palabra	Reemplazo por caracteres	Número de Pruebas realizadas	Observaciones
1	UNIVERSIDAD ISRAEL	ELECTRONICA _PURA_	15	Sin novedad
2	JUAN C VÁSQUEZ	AUCAS CAMPEON	12	Sin novedad
3	BIENVENIDOS	__ARDUINO__	14	Sin novedad
4	EFEECTO POV	BLUETOOTH_	16	Sin novedad
5	UNISCAN CIA LTDA	ECUADOR MI__PAIS	15	Sin novedad
6	ZEBRA	SUGAR	11	Sin novedad
7	HONEYWELL MOTOROLA	_GRACIAS_ _AMIGOS_	14	Sin novedad

Tabla 3.3 Reemplazo de caracteres

Fuente: Investigador

### Prueba 2: Adición de imágenes creadas en la programación.

La tabla 3.4 muestra los resultados de las pruebas de funcionamiento del reemplazo por imágenes programadas en el software del dispositivo de efecto POV. Se agregaron 7 imágenes de acuerdo a lo que permite cada secuencia.

Secuencia	Caracteres	Reemplazo por imágenes	Número de Pruebas realizadas	Observaciones
1	UNIVERSIDAD ISRAEL	CORAZON FLECHA	10	Sin novedad
2	JUAN C VÁSQUEZ	CALAVERA PACMAN	13	Sin novedad
3	BIENVENIDOS	CARA FELIZ	12	Sin novedad
4	EFEECTO POV	RAYO	12	Sin novedad
5	UNISCAN CIA LTDA	LUNA	14	Sin novedad
6	ZEBRA	SLEEP	11	Sin novedad
7	HONEYWELL MOTOROLA	ONDAS	13	Sin novedad

Tabla 3.4 Reemplazo de caracteres por imágenes

Fuente: Investigador

**Prueba 3: Cambio de rotación de giro de palabra de izquierda a derecha o viceversa.**

La tabla 3.5 muestra los resultados de las pruebas de funcionamiento del cambio de giro de la palabra o caracteres del dispositivo de efecto POV.

<b>Secuencia</b>	<b>Caracteres</b>	<b>Giro</b>	<b>Número de Pruebas realizadas</b>	<b>Observaciones</b>
1	UNIVERSIDAD ISRAEL	Sentido horario a sentido anti horario	12	Sin novedad
2	JUAN C VÁSQUEZ	Sentido anti horario a sentido horario	11	Sin novedad
3	BIENVENIDOS	Sentido horario a sentido anti horario	11	Sin novedad
4	EFFECTO POV	Sentido horario a sentido anti horario	13	Sin novedad
5	UNISCAN CIA LTDA	Sentido anti horario a sentido horario	14	Sin novedad
6	ZEBRA	Sentido horario a sentido anti horario	15	Sin novedad
7	HONEYWELL MOTOROLA	Sentido anti horario a sentido horario	12	Sin novedad

Tabla 3.5 Cambio de rotación de giro del mensaje

Fuente: Investigador

**Prueba 4: Nitidez de la imagen de acuerdo al ajuste de velocidad del motor.**

La tabla 3.6 muestra los resultados de las pruebas de funcionamiento del mejoramiento o distorsión de caracteres en el dispositivo de efecto POV.

<b>Velocidad del motor</b>	<b>Secuencia</b>	<b>Efecto</b>	<b>Número de Pruebas realizadas</b>	<b>Observaciones</b>
Aumento	UNIVERSIDAD ISRAEL	Sobre posición de caracteres.	15	Sin novedad
Disminución	JUAN C VÁSQUEZ	Repetición y distorsión.	15	Sin novedad

Tabla 3.6 Sobre montaje o distorsión de caracteres

Fuente: Investigador

**Prueba 5: Cambio de secuencias mediante Smart Phone o PC con soporte hyperterminal para bluetooth.**

La tabla 3.7 muestra los resultados de las pruebas de funcionamiento del cambio de secuencias del dispositivo de efecto POV, cuando se presiona uno de los botones del teclado alfanumérico del Smart Phone según lo programado en el software.

<b>Botón presionado teclado</b>	<b>Secuencia mostrada</b>	<b>Número de pruebas realizadas</b>	<b>Observaciones</b>
1	UNIVERSIDAD ISRAEL	12	Sin novedad
2	JUAN C VÁSQUEZ	11	Sin novedad
3	BIENVENIDOS	11	Sin novedad
4	EFEECTO POV	13	Sin novedad
5	UNISCAN CIA LTDA	14	Sin novedad
6	ZEBRA	15	Sin novedad
7	HONEYWELL MOTOROLA	12	Sin novedad

Tabla 3.7 Cambio de secuencias

Fuente: Investigador

**Prueba 6: Encendido y apagado de secuencias al presionar los botones programados para realizar la tarea.**

La tabla 3.8 muestra los resultados de las pruebas de funcionamiento del control de encendido y apagado de LEDs, cuando se presionan los botones del teclado alfanumérico del Smart Phone según lo programado en el software.

<b>Botón Presionado Teclado</b>	<b>Acción Mostrada</b>	<b>Número de Pruebas realizadas</b>	<b>Observaciones</b>
1	Enciende secuencia	14	Sin novedad
0	Apaga secuencia	14	Sin novedad
2	Enciende secuencia	12	Sin novedad
0	Apaga secuencia	12	Sin novedad
3	Enciende secuencia	10	Sin novedad
0	Apaga secuencia	10	Sin novedad

Tabla 3.8 Encendido y apagado de LEDs

Fuente: Investigador

### **3.9 Análisis de Resultados**

**Análisis de resultados de la Prueba 1:** En las pruebas de la tabla 3.3 se realizó el reemplazo de caracteres con éxito, hay que tomar en cuenta que es necesario reemplazar los caracteres iniciales por otros de la misma dimensión.

**Análisis de resultados de la Prueba 2:** En las pruebas de la tabla 3.4 se realizó el reemplazo de caracteres por imágenes, de igual manera se necesita cubrir todos los espacios de cada palabra mostrada.

**Análisis de resultados de la Prueba 3:** En las pruebas de la tabla 3.5 se realizó el cambio del sentido de giro de los caracteres de cada secuencia, es necesario variar (aumentar o disminuir) una mínima cantidad de velocidad en el motor para crear el efecto.

**Análisis de resultados de la Prueba 4:** En las pruebas de la tabla 3.6 se realizó la sobre posición o distorsión entre caracteres del dispositivo, variando al máximo la velocidad del motor se sobrepone la imagen y al variar al mínimo la velocidad de giro del motor la imagen se repite y distorsiona.

**Análisis de resultados de la Prueba 5:** En las pruebas de la tabla 3.7 se realizó el cambio de secuencias mediante un dispositivo Smart Phone con capacidad de soportar conexión hyperterminal bluetooth, al presionar los botones del Smart Phone de acuerdo a lo programado cambia la secuencia sin ningún inconveniente.

**Análisis de resultados de la Prueba 6:** En las pruebas de la tabla 3.8 se realizó el encendido y apagado de secuencias mediante un PC con capacidad de soportar conexión hyperterminal bluetooth, al presionar los botones del teclado del PC de acuerdo a lo programado enciende o apaga los LEDs.

## CONCLUSIONES Y RECOMENDACIONES

### Conclusiones

- Al realizar el proyecto mediante la plataforma Arduino, se puede comprobar que esta tecnología es muy versátil al momento de desarrollar y trabajar con el lenguaje proporcionado en sus librerías.
- Con el estudio e investigación realizada en el presente proyecto sobre el efecto POV se ha logrado desarrollar un dispositivo bastante novedoso, interesante y divertido de formar caracteres, palabras, efectos etc. Literalmente en el aire.
- Al verificar la programación desarrollada en el entorno Arduino se menciona que existe la facilidad de reemplazar cualquier carácter o mensaje por otro que se desee ubicar.
- Aunque el módulo bluetooth HC-06 no tiene suficiente respaldo escrito, se logró comprobar que es un dispositivo bastante fácil y estable para utilizar con PC o Smart Phone, puesto que convierte un puerto serial a bluetooth sin complicaciones.

### Recomendaciones

- En caso que se requiera ubicar más caracteres o mensajes en el dispositivo se sugiere reemplazar la placa Arduino utilizada con otra de mayor capacidad.
- Si se desea modificar la estética del dispositivo por ejemplo el color de la estructura se propone reemplazarla con el color más adecuado para el cliente.
- Se sugiere realizar una limpieza periódica en los LEDs para que no afecte la nitidez de la presentación de la imagen.
- Verificar que el motor se encuentre regularmente con lubricación para su giro correcto.

## BIBLIOGRAFÍA

- Cambridge, University. (2009). Roget, Peter: Explanation of an optical deception in the appearance of the spokes of a wheel when seen through vertical apertures, Londres 1825: Reedición por Press ISBN 978-1-108-00008-6.
- Ghent (2011), Museum for the History of Sciences, Ghent Scientists: Joseph Plateau, Francia: Recuperado 2 de Octubre del 2011.
- Boylestad, Robert L y Nashelsky L. (2009). Teoría de Circuitos: Diodos Emisores de Luz. (pg. 41-48). México: Prentice Hall.
- Malvino Albert P. (2006). Principios de electrónica: Diodos LED. (pg. 195-198). Madrid: McGraw-Hill interamericana.
- Robledo, Hilario. (2013). Libro Láser IV: Normas Láser y clasificaciones. (pg. 548-562). España: Académica española.
- Hurtado, Nieves. (2006). Motores Eléctricos: Unidad 11. (pg. 289-299). USA: McGraw-Hill.
- Bermúdez, Vicente. (2006). Tecnología Energética: Aspectos técnicos de la electricidad. (pg. 139-140). España: Servicio de Publicaciones.
- Zemansky, Sears y Young Freedman. (2009). Física Universitaria: Rotación de cuerpos Rígidos. (pg. 285-299). México: Pears Education.
- Camara, Sergi. (2006). Historia del Dibujo Animado. Disponible en: <http://dibujoanimado.blogspot.com>
- Fuentes, Alexander. (2009). El estroboscopio: ¿Qué es el estroboscopio? Disponible en: <http://estrober.blogspot.com/2009/03/el-estroboscopio.html>
- Griño, Nicolau. (2009). Historia del Cine NIC. Disponible en: <http://www.cinenic.es/historia.html>
- Banzi, Massimo. (2009). Getting Started with Arduino. Disponible en: <http://arduino.cc/en/Main/Products?from=Main.Hardware>.
- Oser, Jonathan. (2009). Practical Arduino: Arduino Uno. Disponible en: <http://arduino.cc/en/Main/arduinoBoardUno>.
- Peterson, Josh. (2013). Módulo Bluetooth HC-06 ideal Arduino. Disponible en: <http://electgpl.blogspot.com/2013/01/modulo-bluetooth-hc-06.html>

# **ANEXOS**

# **Anexo 1:**

Entrevistas realizadas

UNIVERSIDAD TECNOLÓGICA ISRAEL  
CARRERA DE ELECTRONICA Y TELECOMUNICACIONES

Entrevista al personal de la Empresa UNISCAN CIA. LTDA.

Nombre: Milton Navone

Cargo: Dpto. Crédito y Cobranza

Fecha: 29/07/2014

1. ¿Qué opina usted sobre la implementación del nuevo dispositivo de publicidad de efecto POV?

Bueno  Regular  Malo

Porque? Ayuda a visualizar las marcas que la empresa representa, de una manera atractiva

2. ¿Cree usted que el dispositivo ayudó al cliente en la publicidad de las marcas representadas por su empresa?

Bueno  Regular  Malo

Porque? Atrajo la atención del cliente

3.- ¿Cree usted que se pueda implementar el dispositivo en las oficinas de la ciudad de Guayaquil tomando en cuenta el resultado obtenido en la matriz principal de Quito?

Si  No  Tal vez

Porque? Es un producto novedoso y atractivo

4.- ¿Está usted familiarizado con el manejo del equipo? ¿Qué ventajas observa sobre el dispositivo implementado?

Si  No  Tal vez

Porque? El Sr. Juan C. Vasquez me indicó su  
familiaridad.

5.- ¿Cree usted que el nuevo dispositivo de efecto POV mantiene una estética adecuada para la empresa?

Si  No  Tal vez

Porque? Tiene un modo normal

UNIVERSIDAD TECNOLÓGICA ISRAEL  
CARRERA DE ELECTRONICA Y TELECOMUNICACIONES

Entrevista al personal de la Empresa UNISCAN CIA. LTDA.

Nombre: Betsy Katherine Pabon Puga

Cargo: Recepción

Fecha: 29/07/2014

1. ¿Qué opina usted sobre la implementación del nuevo dispositivo de publicidad de efecto POV.?

Bueno  Regular  Malo

Porque? Una nueva estrategia de publicidad ayuda en todo el mercado nacional e internacional siempre y cuando no se gaste más de lo que se gana.

2. ¿Cree usted que el dispositivo ayudó al cliente en la publicidad de las marcas representadas por su empresa?

Bueno  Regular  Malo

Porque? Algo nuevo y novedoso siempre ayuda a causar impacto en los clientes, las personas o clientes se familiarizan más con las marcas y de cierta manera nos ayuda a crecer tecnológicamente.

3. ¿Cree usted que se pueda implementar el dispositivo en las oficinas de la ciudad de Guayaquil tomando en cuenta el resultado obtenido en la matriz principal de Quito?

Si  No  Tal vez

Porque? Hay que tomar en cuenta que un nuevo producto tiene que estar un buen tiempo en el mercado para comprobar su eficacia, arriesgarnos a implementar en Guayaquil sería dentro de un buen tiempo ya que en la sucursal no tenemos tantas clientes como en la matriz.

4.- ¿Está usted familiarizado con el manejo del equipo? ¿Qué ventajas observa sobre el dispositivo implementado?

Si  No  Tal vez

Porque? Es de fácil manejo, no se necesita de mucha ciencia para entenderlo y por parte de mantenimientos a algún inconveniente se encargan las personas del departamento técnico.

5.- ¿Cree usted que el nuevo dispositivo de efecto POV mantiene una estética adecuada para la empresa?

Si  No  Tal vez

Porque? En la actualidad el marketing y la publicidad se maneja con buena imagen, la carta de presentación hacia un cliente es la primera imagen, esto favorece mucho a la imagen corporativa de esta empresa como de cualquier otra.

UNIVERSIDAD TECNOLÓGICA ISRAEL  
CARRERA DE ELECTRONICA Y TELECOMUNICACIONES

Entrevista al personal de la Empresa UNISCAN CIA. LTDA.

Nombre: Heiko Navarro

Cargo: Jefe de Bodega

Fecha: 20 de Julio 2014

1. ¿Qué opina usted sobre la implementación del nuevo dispositivo de publicidad de efecto POV?

Bueno  Regular  Malo

Porque? Ya que es una forma innovadora de hacer publicidad

2. ¿Cree usted que el dispositivo ayudó al cliente en la publicidad de las marcas representadas por su empresa?

Bueno  Regular  Malo

Porque? Claro ya que solo (brevemente) tiempo de hacer publicidad es bastante después a se hace la publicidad Regular

3. ¿Cree usted que se pueda implementar el dispositivo en las oficinas de la ciudad de Guayaquil tomando en cuenta el resultado obtenido en la matriz principal de Quito?

Si  No  Tal vez

Porque? Si por que se puede exponer la marca a la cual se hace publicidad y la forma de hacer publicidad

4.- ¿Está usted familiarizado con el manejo del equipo? ¿Qué ventajas observa sobre el dispositivo implementado?

Si  No  Tal vez

Porque? Es un trabajo Hemático y una forma muy diferente de hacer publicidad

5.- ¿Cree usted que el nuevo dispositivo de efecto POV mantiene una estética adecuada para la empresa?

Si  No  Tal vez

Porque? Si es que se quisiera Hemático y se colocara en cualquier lugar

UNIVERSIDAD TECNOLÓGICA ISRAEL  
CARRERA DE ELECTRONICA Y TELECOMUNICACIONES

Entrevista al personal de la Empresa UNISCAN CIA. LTDA.

Nombre: ANNY JHOANA VAGO

Cargo: DEPARTAMENTO TECNICO

Fecha: 24/07/2014

1. ¿Qué opina usted sobre la implementación del nuevo dispositivo de publicidad de efecto POV.?

Bueno  Regular  Malo

Porque? YA Q' DADO A MENUDA LA PUBLICIDAD  
ES UN SENTIDO DIFERENTE A LOS CONVENCIONALES.

2. ¿Cree usted que el dispositivo ayudó al cliente en la publicidad de las marcas representadas por su empresa?

Bueno  Regular  Malo

Porque? YA Q' EL CLIENTE VIO DE UNA FORMA  
DISTINTA LA PUBLICIDAD

3.- ¿Cree usted que se pueda implementar el dispositivo en las oficinas de la ciudad de Guayaquil tomando en cuenta el resultado obtenido en la matriz principal de Quito?

Si  No  Tal vez

Porque? PODE MEJOR LOS VENTOS, YA Q' MEJORA  
LA VISIBILIDAD DEL PRODUCTO

4.- ¿Está usted familiarizado con el manejo del equipo? ¿Qué ventajas observa sobre el dispositivo implementado?

Si:  No:  Tal vez:

Porque? MAYOR VISIÓN Y FORTALECIMIENTO.

5.- ¿Cree usted que el nuevo dispositivo de efecto POV mantiene una estética adecuada para la empresa?

Si:  No:  Tal vez:

Porque? PUEDE MEJORAR CON ALGUNOS AJUSTES ESTÉTICOS MÁS DETALLADOS.

UNIVERSIDAD TECNOLÓGICA ISRAEL  
CARRERA DE ELECTRONICA Y TELECOMUNICACIONES

Entrevista al personal de la Empresa UNISCAN CIA. LTDA.

Nombre: JENNIFER BETANCOURT

Cargo: CONTADOR

Fecha: 29/06/2014

1. ¿Qué opina usted sobre la implementación del nuevo dispositivo de publicidad de efecto POV?

Bueno

Regular

Malo

Porque? Es una herramienta nueva y novedosa para la empresa

2. ¿Cree usted que el dispositivo ayudó al cliente en la publicidad de las marcas representadas por su empresa?

Bueno

Regular

Malo

Porque? Ya q. llamo la atención del cliente quien solicita productos de esas marcas.

3.- ¿Cree usted que se pueda implementar el dispositivo en las oficinas de la ciudad de Guayaquil tomando en cuenta el resultado obtenido en la matriz principal de Quito?

Si

No

Tal vez

Porque? Esto nos ayudaría a subir el nivel de ventas en Guayaquil

4.- ¿Está usted familiarizado con el manejo del equipo? ¿Qué ventajas observa sobre el dispositivo implementado?

Si  No  Tal vez

Porque? NO estoy a cargo de eso

5.- ¿Cree usted que el nuevo dispositivo de efecto POV mantiene una estética adecuada para la empresa?

Si  No  Tal vez

Porque? Me parece q' el modelo es adecuado

# **Anexo 2:**

Manual de Usuario



# Manual de Usuario

## Efecto POV

### IMPORTANTE

Por favor lea detalladamente este manual de usuario antes de conectar y operar el dispositivo.

Su nuevo dispositivo de efecto POV es una adición a su hogar u oficina que proporcionará comodidad y rendimiento por muchos años. Este manual de usuario le proporciona instrucciones completas para instalar y operar su dispositivo.

## **ADVERTENCIA**



Para reducir el riesgo de incendios o descargas eléctricas, no exponga el dispositivo a la lluvia o a la humedad.

Para evitar una descarga eléctrica, no abra la unidad; solicite asistencia técnica únicamente a personal capacitado.

El cable de alimentación de CA sólo debe ser reemplazado con soporte técnico especializado.

No exponga la batería, ni los aparatos que tengan pilas instaladas, a fuentes de calor excesivo, tales como la luz solar, el fuego o similares.

Para prevenir lesiones, este aparato debe estar sujeto de forma segura al piso o a la pared según las instrucciones de instalación.

## **PRECAUCION**

Para uso interno únicamente.



El uso de instrumentos ópticos con este producto aumenta el riesgo de sufrir daños oculares. La prolongación de visión fija al producto es perjudicial para los ojos. Solicite reparaciones únicamente a personal calificado.

## **NOTA**



Este dispositivo se ha sometido a pruebas y se ha determinado que el uso adecuado prolonga la vida útil del dispositivo. Sin embargo no está permitido reproducir distribuir, traducir o transmitir ninguna parte de esta guía de ningún modo o por ningún medio, ya sea electrónico o mecánico, incluyendo fotocopias, grabaciones o almacenamiento en cualquier sistema de almacenamiento y recuperación de información, sin el permiso previo.

El contenido puede diferir del producto final o del software proporcionado por el proveedor de servicios o transmisores, se encuentra sujeto a cambio sin previo aviso. Para acceder a la última versión del manual, solicite al email siguiente: [jslasociados@hotmail.com](mailto:jslasociados@hotmail.com)

## Índice del Manual de Usuario del dispositivo de EFECTO POV

1. Precauciones de Seguridad .....	3
1.1 Ubicación del dispositivo .....	3
1.2 Encendido del dispositivo y funcionamiento. ....	3
1.3. Variar la velocidad del motor. ....	4
1.4 Enviando primer secuencia de mensajes desde el dispositivo bluetooth (Pc o Smart Phone) .....	4
1.5 Uso de dispositivo bluetooth .....	4
1.5 ¿Cómo conectar un computador con el dispositivo de Efecto POV? .....	5
2. Introducción.....	7
3. Solución a problemas más comunes. ....	8
4. E-mail y teléfonos de soporte técnico. ....	9

## 1. Precauciones de Seguridad

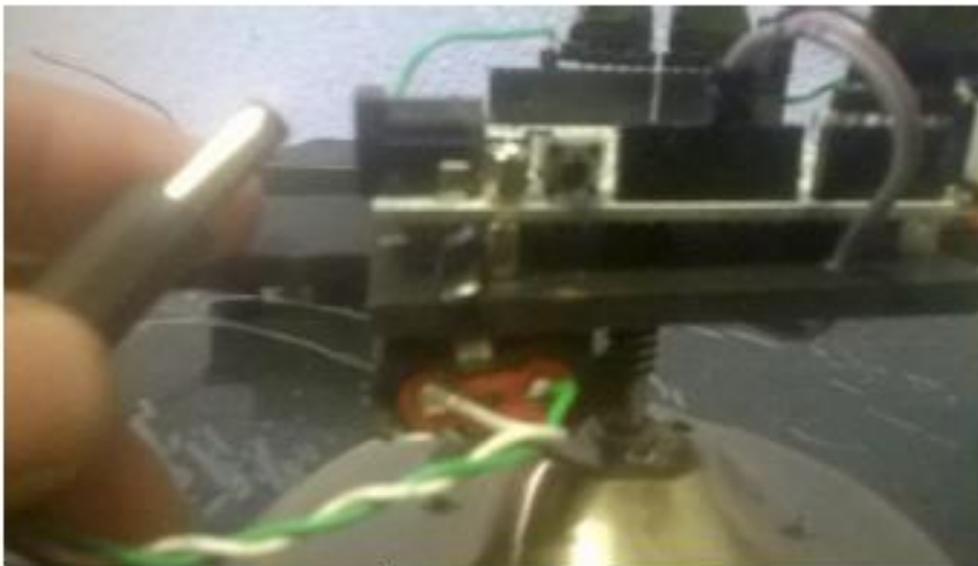
### 1.1 Ubicación del dispositivo

Para prevenir daños al dispositivo y a cosas cercanas a este se debe tener en cuenta las siguientes recomendaciones.

- Coloque el dispositivo en un lugar con ventilación adecuada para evitar el recalentamiento del mismo.
- No coloque el dispositivo sobre una superficie blanda, las vibraciones que genera el dispositivo puede hacer que caiga y se estropee.
- No coloque el dispositivo cerca de fuentes de calor, ni en lugares sujetos a la luz directa del sol, polvo excesivo o golpes mecánicos
- No ubique el dispositivo en exteriores.
- No instale el dispositivo en posición inclinada. Está diseñado para utilizarse solo en posición horizontal o vertical.
- Verificar que el cable de conexión no quede obstaculizando el dispositivo, esto causará un mal funcionamiento del dispositivo.

### 1.2 Encendido del dispositivo y funcionamiento.

- Conecte la placa Arduino del dispositivo a la batería de 9 voltios, la batería se encuentra en la parte inferior de la guía con Leds.



- Conecte el cable de suministro de energía a una toma de corriente que se encuentre con conexión a tierra. Y del otro extremo conéctelo a la fuente de alimentación variable.





### 1.3. Variar la velocidad del motor.

Ahora el dispositivo ya se encuentra girando y podemos regular la velocidad del mismo con el variador de velocidad (Potenciómetro) ubicado en una esquina de la estructura, esto se lo debe realizar para alcanzar la visualización adecuada del mensaje.



### 1.4 Enviando primer secuencia de mensajes desde el dispositivo bluetooth (Pc o Smart Phone)

Para ello es necesario ingresar desde la misma aplicación de Arduino, conexión hyperterminal, o en caso de los Smart Phones desde la aplicación que soporte la aplicación terminal de Bluetooth.

*Nota: Las secuencias son programadas de acuerdo a lo solicitado por el cliente y acopladas al teclado del equipo conforme lo requiera. (Sólo personal calificado para el cambio de configuración de secuencias o los mensajes finales). Para realizar por primera vez el enlace con el dispositivo de Efecto POV se debe seguir los pasos 1.5.*

### 1.5 Uso de dispositivo bluetooth

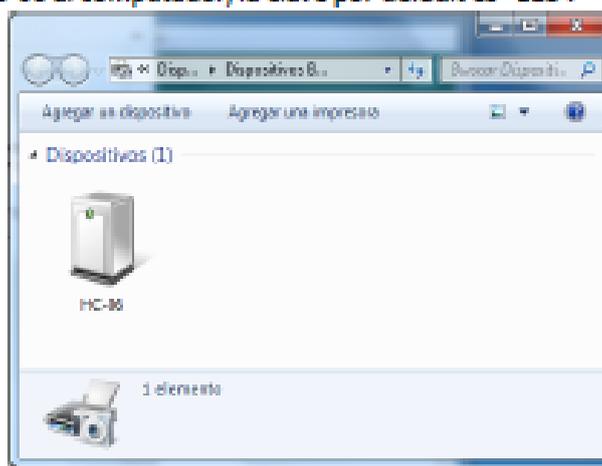
Disponer de un dispositivo portátil o de escritorio con conectividad bluetooth, en caso de Smart Phone debe soportar la aplicación hyperterminal para la transmisión de datos desde y hacia el dispositivo efecto POV con bluetooth integrado.



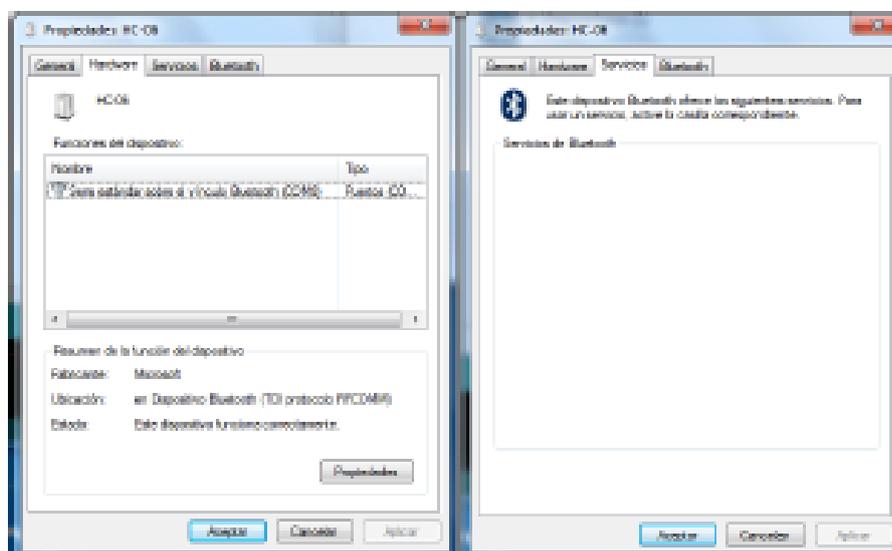
*Nota: El bluetooth del dispositivo de efecto POV se encuentra permanentemente encendido. Gracias a la memoria de almacenamiento del módulo Bluetooth HC-06 se informa que solo requiere de una vez de la configuración del enlace. Para las siguientes conexiones basta con encender el bluetooth del dispositivo (PC) y presionar el teclado de acuerdo a la secuencia programada.*

### 1.5 ¿Cómo conectar un computador con el dispositivo de Efecto POV?

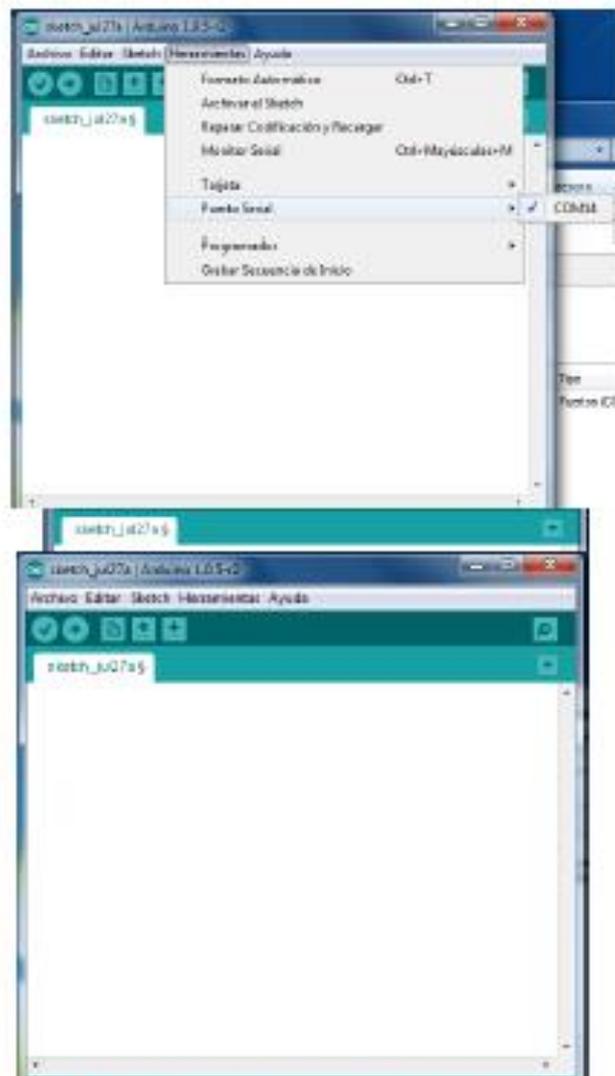
- Instalar el programa Arduino en el computador disponible en el siguiente enlace <http://arduino.cc/en/pmwiki.php?n=main/software>, disponible para diferentes sistemas operativos.
- Encender el bluetooth del computador y realizar una búsqueda del dispositivo que por default se encuentra con el nombre HC-06.
- Agregar el dispositivo HC-06 al computador, la clave por default es "1234"



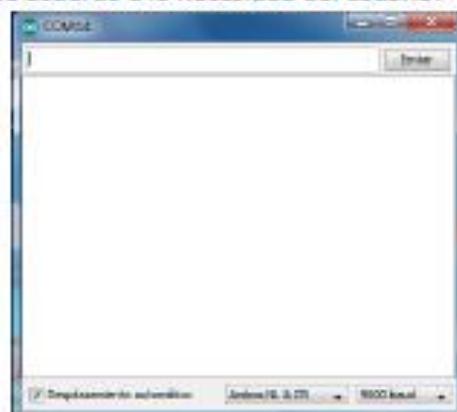
- Verificar el puerto serial virtual creado por el computador, para el caso de Sistema Operativo Windows 7 hacer click en inicio, dispositivos e impresoras, doble click en el dispositivo agregado, aparecerá una nueva ventana y se debe escoger la pestaña de servicios.



- Una vez verificado el puerto virtual serial COM Abrir el software Arduino.
- Dirigirse a la opción de Herramientas y desplegar la opción de puerto serial, escoger la opción de conexión de nuestro puerto COM.



- Realizado el proceso anterior dirigirse al icono de Monitor Serial.
- Aparecerá la línea de comandos para ingresar los datos programados en la placa Arduino. En esta sección se puede controlar la secuencia de acuerdo a la necesidad del usuario.



Ejemplo: Al presionar el numero 1 mostrará el primer mensaje en el dispositivo de efecto POV.

## 2. Introducción

### 2.1 Descripción de partes y accesorios.

#### a) Etapa de alimentación del circuito.



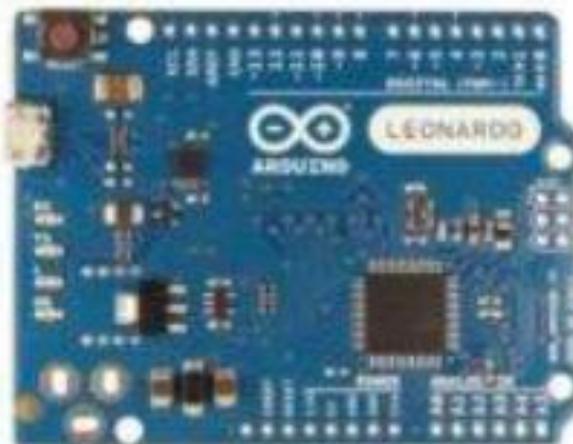
Tiene por finalidad controlar la velocidad del motor, se conecta directamente a la toma de corriente de 110/120 VAC. Está ubicado en una caja plástica.

#### b) Conector para el adaptador y batería.



La batería suministra el voltaje suficiente para el funcionamiento de la placa Arduino y módulo bluetooth.

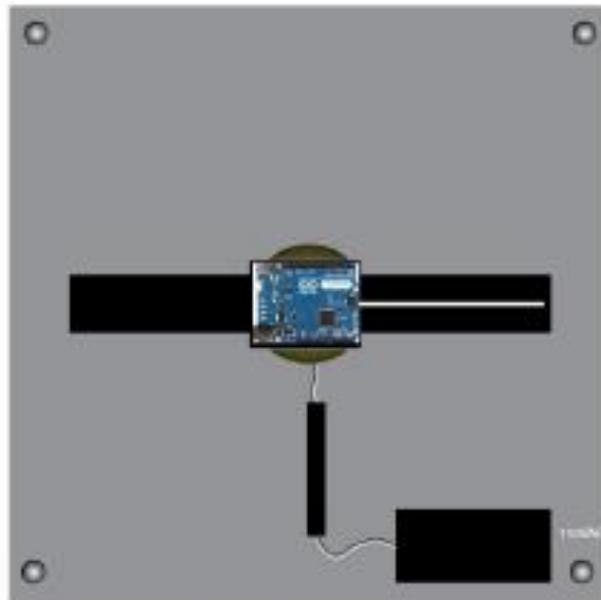
#### c) Placa electrónica.



La placa electrónica contiene todos los elementos como el microcontrolador, entradas/salidas y demás componentes que hacen de la placa una excelente herramienta.

#### d) Estructura total.

Cuando la placa y sus componentes se encuentran ubicados correctamente se observara de la siguiente forma.



e) Luces secuenciales.

Al girar los componentes de la estructura y enviar el mensaje desde el módulo Bluetooth se mostrara un mensaje como el de la siguiente figura.



El dispositivo estará completamente armado en su totalidad simplemente es necesario colocarlo en un lugar adecuado, conectar primero la batería de 9 volts a la placa Arduino, y posteriormente conectar la etapa de alimentación del motor a la toma de corriente.

### 3. Solución a problemas más comunes.

Errores más comunes	Solución
No gira el dispositivo efecto POV.	Verificar el cable de poder y el voltaje en la toma de corriente
No enciende los Leds.	a) La placa Arduino no está debidamente conectada. b) No se enlazó adecuadamente el modulo bluetooth con el dispositivo.
Mensajes no se muestran de forma normal	Verificar que la velocidad del motor gire correctamente con el potenciómetro.
No enciende todo el sistema	Comuníquese con soporte técnico.
Enlace de bluetooth no enciende.	Comuníquese con soporte técnico.

#### 4. E-mail y teléfonos de soporte técnico.

Consulte el e-mail [jslasociados@hotmail.com](mailto:jslasociados@hotmail.com) para la versión más reciente del Manual del Usuario, información adicional, descargas y servicios relacionados con su dispositivo.

Si necesita comunicarse con el Centro de Atención del Cliente o Soporte Técnico llamar al número 0995201001/3204057.

# **Anexo 3:**

Desarrollo total de la programación sistema  
Arduino para el efecto POV.

```

const int p0 = 40;
const int p1 = 2;
const int p2 = 3;
const int p3 = 4;
const int p4 = 5;
const int p5 = 6;
const int p6 = 7;
const int p7 = 8;
const int p8 = 9;
const int p9 = 10;
const int p10 = 11;
const int p11 = 12;
const int p12 = 13;
const int p13 = A0;
const int p14 = A1;
const int p15 = A2;
const int p16 = A3;
const int p17 = A4;
const int p18 = A5;

int fila1[]={p0,p1,p2,p3,p4,p5,p6,p7,p8};
int fila2[]={p7,p8,p9,p10,p11,p12,p13,p14};
int fila3[]={p11,p12,p13,p14,p15,p16,p17,p18};

int _[]={0,0,0,0,0,0,0,0, 0,0,0,0,0,0,0,0, 0,0,0,0,0,0,0,0, 0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0};
int A[]={0,1,1,1,1,1,1,1,0, 0,0,0,1,0,0,0,1, 0,0,0,1,0,0,0,1, 0,0,0,1,0,0,0,1,
0,1,1,1,1,1,1,0};
int B[]={0,1,1,1,1,1,1,1, 0,1,0,0,1,0,0,1, 0,1,0,0,1,0,0,1, 0,1,0,0,1,0,0,1,
0,0,1,1,0,1,1,0};
int C[]={0,0,1,1,1,1,1,1,0, 0,1,0,0,0,0,0,1, 0,1,0,0,0,0,0,1, 0,1,0,0,0,0,0,1,
0,0,1,0,0,0,1,0};
int D[]={0,1,1,1,1,1,1,1, 0,1,0,0,0,0,0,1, 0,1,0,0,0,0,0,1, 0,1,0,0,0,0,0,1,
0,0,1,1,1,1,1,0};
int E[]={0,1,1,1,1,1,1,1, 0,1,0,0,1,0,0,1, 0,1,0,0,1,0,0,1, 0,1,0,0,1,0,0,1,
0,1,0,0,0,0,0,1};
int F[]={0,1,1,1,1,1,1,1, 0,0,0,0,1,0,0,1, 0,0,0,0,1,0,0,1, 0,0,0,0,1,0,0,1,
0,0,0,0,0,0,0,1};
int G[]={0,0,1,1,1,1,1,1,0, 0,1,0,0,0,0,0,1, 0,1,0,0,0,0,0,1, 0,1,0,0,1,0,0,1,
0,1,1,1,1,0,1,0};
int H[]={0,1,1,1,1,1,1,1, 0,0,0,0,1,0,0,0, 0,0,0,0,1,0,0,0, 0,0,0,0,1,0,0,0,
0,1,1,1,1,1,1,1};
int I[]={0,0,0,0,0,0,0,0, 0,1,0,0,0,0,0,1, 0,1,1,1,1,1,1,1, 0,1,0,0,0,0,0,1,
0,0,0,0,0,0,0,0};
int J[]={0,0,1,0,0,0,0,0, 0,1,0,0,0,0,0,0, 0,1,0,0,0,0,0,1, 0,0,1,1,1,1,1,1,
0,0,0,0,0,0,0,1};
int K[]={0,1,1,1,1,1,1,1, 0,0,0,0,1,0,0,0, 0,0,0,1,0,1,0,0, 0,0,1,0,0,0,1,0,
0,1,0,0,0,0,0,1};
int L[]={0,1,1,1,1,1,1,1, 0,1,0,0,0,0,0,0, 0,1,0,0,0,0,0,0, 0,1,0,0,0,0,0,0,
0,1,0,0,0,0,0,0};
int M[]={0,1,1,1,1,1,1,1, 0,0,0,0,0,0,1,0, 0,0,0,0,1,1,0,0, 0,0,0,0,0,0,1,0,
0,1,1,1,1,1,1,1};
int N[]={0,1,1,1,1,1,1,1, 0,0,0,0,0,1,0,0, 0,0,0,0,1,0,0,0, 0,0,0,1,0,0,0,0,
0,1,1,1,1,1,1,1};
int O[]={0,0,1,1,1,1,1,1,0, 0,1,0,0,0,0,0,1, 0,1,0,0,0,0,0,1, 0,1,0,0,0,0,0,1,
0,0,1,1,1,1,1,0};
int P[]={0,1,1,1,1,1,1,1, 0,0,0,0,1,0,0,1, 0,0,0,0,1,0,0,1, 0,0,0,0,1,0,0,1,
0,0,0,0,0,1,1,0};
int Q[]={0,0,1,1,1,1,1,1,0, 0,1,0,0,0,0,0,1, 0,1,0,1,0,0,0,1, 0,0,1,0,0,0,0,1,
0,1,0,1,1,1,1,0};
int R[]={0,1,1,1,1,1,1,1, 0,0,0,0,1,0,0,1, 0,0,0,1,1,0,0,1, 0,0,1,0,1,0,0,1,
0,1,0,0,0,1,1,0};
int S[]={0,0,1,0,0,1,1,1,0, 0,1,0,0,1,0,0,1, 0,1,0,0,1,0,0,1, 0,1,0,0,1,0,0,1,
0,0,1,1,0,0,1,0};
int T[]={0,0,0,0,0,0,0,1, 0,0,0,0,0,0,0,1, 0,1,1,1,1,1,1,1, 0,0,0,0,0,0,0,1,
0,0,0,0,0,0,0,1};
int U[]={0,0,1,1,1,1,1,1, 0,1,0,0,0,0,0,0, 0,1,0,0,0,0,0,0, 0,1,0,0,0,0,0,0,
0,0,1,1,1,1,1,1};
int V[]={0,0,0,0,1,1,1,1, 0,0,1,1,0,0,0,0, 0,1,0,0,0,0,0,0, 0,0,1,1,0,0,0,0,
0,0,0,0,1,1,1,1};
int W[]={0,0,1,1,1,1,1,1, 0,1,0,0,0,0,0,0, 0,0,1,1,1,0,0,0, 0,1,0,0,0,0,0,0,
0,0,1,1,1,1,1,1};
int X[]={0,1,1,0,0,0,1,1, 0,0,0,1,0,1,0,0, 0,0,0,0,1,0,0,0, 0,0,0,1,0,1,0,0,
0,1,1,0,0,0,1,1};
int Y[]={0,0,0,0,0,1,1,1, 0,0,0,0,1,0,0,0, 0,1,1,1,0,0,0,0, 0,0,0,0,1,0,0,0,
0,0,0,0,0,1,1,1};

```

```

int Z[]={0,1,1,0,0,0,0,1, 0,1,0,1,0,0,0,1, 0,1,0,0,1,0,0,1, 0,1,0,0,0,1,0,1,
0,1,0,0,0,0,1,1};

int a[]={0,1,1,1,1,1,0,0, 0,1,0,0,0,1,0,0, 0,1,0,0,0,1,0,0, 0,1,1,1,1,1,0,0,
0,1,0,0,0,0,0,0};
int b[]={0,1,1,1,1,1,1,1, 0,1,0,0,0,1,0,0, 0,1,0,0,0,1,0,0, 0,1,0,0,0,1,0,0,
0,0,1,1,1,0,0,0};
int c[]={0,0,1,1,1,0,0,0, 0,1,0,0,0,1,0,0, 0,1,0,0,0,1,0,0, 0,1,0,0,0,1,0,0,
0,1,0,0,0,1,0,0};
int d[]={0,0,1,1,1,0,0,0, 0,1,0,0,0,1,0,0, 0,1,0,0,0,1,0,0, 0,1,0,0,0,1,0,0,
0,1,1,1,1,1,1,1};
int e[]={0,0,1,1,1,0,0,0, 0,1,0,1,0,1,0,0, 0,1,0,1,0,1,0,0, 0,1,0,1,0,1,0,0,
0,0,0,0,1,0,0,0};
int f[]={0,1,1,1,1,1,1,0, 0,0,0,0,1,0,0,1, 0,0,0,0,1,0,0,1, 0,0,0,0,1,0,0,1,
0,0,0,0,0,0,1,0};
int g[]={0,0,0,0,0,1,1,0, 0,0,1,0,1,0,0,1, 0,1,0,0,1,0,0,1, 0,1,0,0,1,0,0,1,
0,0,1,1,1,1,1,0};
int h[]={0,1,1,1,1,1,1,1, 0,0,0,0,1,0,0,0, 0,0,0,0,1,0,0,0, 0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0};
int i[]={0,0,0,0,0,0,0,0, 0,0,0,0,0,0,0,0, 0,1,1,1,1,1,0,1, 0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0};
int j[]={0,1,0,0,0,0,0,0, 0,1,0,0,0,0,0,0, 0,1,1,1,1,1,1,0, 0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0};
int k[]={0,1,1,1,1,1,1,1, 0,0,0,0,1,1,0,0, 0,0,0,1,1,1,0,0, 0,0,1,0,0,0,1,1,
0,1,0,0,0,0,0,0};
int l[]={0,1,0,0,0,0,0,0, 0,1,0,0,0,0,0,0, 0,1,1,1,1,1,1,0, 0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0};
int m[]={0,1,1,1,1,1,0,0, 0,0,0,0,0,1,0,0, 0,1,1,1,1,0,0,0, 0,0,0,0,0,1,0,0,
0,1,1,1,1,1,0,0};
int n[]={0,1,1,1,1,0,0,0, 0,0,0,0,0,1,0,0, 0,0,0,0,0,1,0,0, 0,0,0,0,0,1,0,0,
0,1,1,1,1,0,0,0};
int o[]={0,0,1,1,1,0,0,0, 0,1,0,0,0,1,0,0, 0,1,0,0,0,1,0,0, 0,1,0,0,0,1,0,0,
0,0,1,1,1,0,0,0};
int p[]={0,1,1,1,1,1,0,0, 0,0,0,1,0,0,1,0, 0,0,0,1,0,0,1,0, 0,0,0,1,0,0,1,0,
0,0,0,0,1,1,0,0};
int q[]={0,0,0,0,1,1,0,0, 0,0,0,1,0,0,1,0, 0,0,0,1,0,0,1,0, 0,0,0,1,0,0,1,0,
0,1,1,1,1,1,0,0};
int r[]={0,1,1,1,1,1,0,0, 0,0,0,0,1,0,0,0, 0,0,0,0,0,1,0,0, 0,0,0,0,0,1,0,0,
0,0,0,0,0,0,0,0};
int s[]={0,1,0,0,1,0,0,0, 0,1,0,1,0,1,0,0, 0,1,0,1,0,1,0,0, 0,1,0,1,0,1,0,0,
0,0,1,0,0,1,0,0};
int t[]={0,0,0,0,0,1,0,0, 0,0,0,0,0,1,0,0, 0,1,1,1,1,1,1,1, 0,1,0,0,0,1,0,0,
0,1,0,0,0,1,0,0};
int u[]={0,0,1,1,1,1,0,0, 0,1,0,0,0,0,0,0, 0,1,0,0,0,0,0,0, 0,1,0,0,0,0,0,0,
0,0,1,1,1,1,0,0};
int v[]={0,0,0,1,1,1,0,0, 0,0,1,0,0,0,0,0, 0,1,0,0,0,0,0,0, 0,0,1,0,0,0,0,0,
0,0,0,1,1,1,0,0};
int w[]={0,0,1,1,1,1,0,0, 0,1,0,0,0,0,0,0, 0,0,1,1,0,0,0,0, 0,1,0,0,0,0,0,0,
0,0,1,1,1,1,0,0};
int x[]={0,1,0,0,0,1,0,0, 0,0,1,0,1,0,0,0, 0,0,0,1,0,0,0,0, 0,0,1,0,1,0,0,0,
0,1,0,0,0,1,0,0};
int y[]={0,0,0,0,0,0,1,1,0, 0,0,0,0,1,0,0,0, 0,1,1,1,1,0,0,0, 0,0,0,0,1,0,0,0,
0,0,0,0,0,1,1,0};
int z[]={0,1,0,0,0,1,0,0, 0,1,1,0,0,1,0,0, 0,1,0,1,0,1,0,0, 0,1,0,0,1,1,0,0,
0,1,0,0,0,1,0,0};

int lov[]={0,0,0,1,1,1,1,0, 0,0,1,0,0,0,0,1, 0,1,0,0,0,0,1,0, 0,0,1,0,0,0,0,1,
0,0,0,1,1,1,1,0};
int ene[]={0,0,1,1,1,1,1,0, 0,1,1,0,1,0,1,1, 1,1,1,1,1,0,1,0, 0,1,1,0,1,0,1,1,
0,0,1,1,1,1,1,0};
int nav[]={0,1,0,0,1,0,0,1, 0,0,1,0,0,0,1,0, 0,0,0,1,0,1,0,0, 0,0,0,0,1,0,0,0,
0,0,0,0,1,0,0,0};
int nava[]={0,0,1,1,1,0,0,0, 0,1,0,1,0,1,0,0, 0,0,0,1,1,0,0,0, 0,1,0,1,0,1,0,0,
0,0,1,1,1,0,0,0};
int navb[]={0,0,0,1,0,0,0,0, 0,1,1,0,1,1,0,0, 0,0,0,1,0,0,0,0, 0,1,1,0,1,1,0,0,
0,0,0,1,0,0,0,0};
int navc[]={1,0,0,1,0,1,0,0, 1,1,1,0,0,0,0,0, 0,0,0,0,1,0,0,0, 1,1,1,0,0,0,0,0,
1,0,0,1,0,1,0,0};
int navd[]={1,1,0,1,1,0,0,0, 0,1,1,1,0,1,0,0, 1,1,1,1,1,1,0,0, 0,1,1,1,0,1,0,0,
1,1,0,1,1,0,0,0};
int nave[]={1,0,1,1,0,0,0,0, 1,1,0,1,1,1,0,0, 0,1,1,0,0,0,0,0, 1,1,0,1,1,1,0,0,
1,0,1,1,0,0,0,0};
int navf[]={0,0,0,0,1,0,0,0, 0,1,0,1,0,1,0,0, 0,0,1,0,0,1,0,0, 0,1,0,1,0,1,0,0,
0,0,0,0,1,0,0,0};
int navg[]={1,0,0,0,1,1,0,0, 1,1,1,1,0,0,0,0, 0,0,0,0,1,0,0,0, 1,1,1,1,0,0,0,0,
1,0,0,0,1,1,0,0};

```

```

int navh[]={1,1,1,0,1,0,0,0, 1,0,1,1,0,1,0,0, 1,0,1,1,1,1,0,0, 1,0,1,1,0,1,0,0,
1,1,1,0,1,0,0,0};
int navi[]={0,1,1,1,1,1,0,0, 0,1,0,1,1,1,0,0, 1,0,1,1,1,1,0,0, 0,1,0,1,1,1,0,0,
0,1,1,1,1,0,0,0};
int navj[]={0,0,1,1,0,1,0,0, 0,1,1,1,1,0,0,0, 0,1,0,1,1,0,0,0, 0,1,1,1,1,0,0,0,
0,0,1,1,0,1,0,0};
/*int navk[]={
int navl[]={
int navm[]={
int navn[]={*/

const int Espacio=1000;
const int Ancho=1000;
//voltaje del motor 3.67Voltios
//sube=sentido reloj
//baja=regresa
int sec0=1450;//universidad oki mic
int sec1=18;//efecto oki mil
int sec2=4000;//nombre mic
int sec3=5200;//unis oki mic
int sec4=20;//bienven okimil
int sec5=82;//zebra okimil
int sec6=9300;//bematech okimic
int sec7=1000;//honeywell mic

int saltar=0;
int sigue=1000;
int dato=0;

void setup()
{
Serial1.begin(9600);
for(int t=0;t<8;t++)
{
pinMode(filal[t],OUTPUT);
pinMode(filal2[t],OUTPUT);
pinMode(filal3[t],OUTPUT);
}
}
void loop()
{
while (Serial1.available()){
dato=Serial1.read();
if (dato=='0'){
for (int s=0; s<1; s++){
saltar++;
if (saltar>=1){
saltar=0;}}}

if (dato=='1'){
for (int s=0; s<2; s++){
saltar++;
if (saltar>=2){
saltar=1;}}}

if (dato=='2'){
for (int s=0; s<3; s++){
saltar++;
if (saltar>=3){
saltar=2;}}}

if (dato=='3'){
for (int s=0; s<4; s++){
saltar++;
if (saltar>=4){
saltar=3;}}}

if (dato=='4'){
for (int s=0; s<5; s++){
saltar++;
if (saltar>=5){
saltar=4;}}}

if (dato=='5'){
for (int s=0; s<6; s++){
saltar++;

```

```

if (saltar>=6){
    saltar=5;}}

if (dato=='6'){
    for (int s=0; s<7; s++){
        saltar++;
    }
    if (saltar>=7){
        saltar=6;}}

if (dato=='7'){
    for (int s=0; s<8; s++){
        saltar++;
    }
    if (saltar>=8){
        saltar=7;}}

if (dato=='8'){
    for (int s=0; s<9; s++){
        saltar++;
    }
    if (saltar>=9){
        saltar=8;}}

if (dato=='9'){
    for (int s=0; s<10; s++){
        saltar++;
    }
    if (saltar>=10){
        saltar=9;}}
}

if (saltar==1){
    delayMicroseconds(sec0);
    muestre3(U);
    muestre3(N);
    muestre3(I);
    muestre3(V);
    muestre3(E);
    muestre3(R);
    muestre3(S);
    muestre3(I);
    muestre3(D);
    muestre3(A);
    muestre3(D);
    delayMicroseconds(sec0);
    muestre2(I);
    muestre2(S);
    muestre2(R);
    muestre2(A);
    muestre2(E);
    muestre2(L);
    delayMicroseconds(sec0);
}

if (saltar==2){
    delay(sec1);
    muestre2(E);
    muestre2(F);
    muestre2(E);
    muestre2(C);
    muestre2(T);
    muestre2(O);
    delay(sec1);
    muestre1(P);
    muestre1(O);
    muestre1(V);
    delay(sec1);
}

if (saltar==3){
    delayMicroseconds(sec2);
    muestre3(V);
    muestre3(A);
    muestre3(S);
    muestre3(Q);
    muestre3(U);
    muestre3(E);
    muestre3(Z);
    delayMicroseconds(sec2);
    delayMicroseconds(sec2);
    muestre2(J);
    muestre2(U);
    muestre2(A);
}

```

```

    muestre2(N);
    muestre2(C);
    delayMicroseconds(sec2);
}
if (saltar==4){
    delayMicroseconds(sec3);
    muestre3(U);
    muestre3(N);
    muestre3(I);
    muestre3(S);
    muestre3(C);
    muestre3(A);
    muestre3(N);
    delayMicroseconds(sec3);
    muestre2(C);
    muestre2(I);
    muestre2(A);
    delayMicroseconds(sec3);
    muestre1(L);
    muestre1(T);
    muestre1(D);
    muestre1(A);
    delayMicroseconds(sec3);
}
if (saltar==5){
    delay(sec4);
    muestre2(B);
    muestre2(I);
    muestre2(E);
    muestre2(N);
    muestre2(V);
    muestre2(E);
    muestre2(N);
    muestre2(I);
    muestre2(D);
    muestre2(O);
    muestre2(S);
    delay(sec4);
}
if (saltar==6){
    delay(sec5);
    muestre2(Z);
    muestre2(E);
    muestre2(B);
    muestre2(R);
    muestre2(A);
}
if (saltar==7){
    delayMicroseconds(sec6);
    muestre3(B);
    muestre3(E);
    muestre3(M);
    muestre3(A);
    muestre3(T);
    muestre3(E);
    muestre3(C);
    muestre3(H);
    delayMicroseconds(sec6);
    muestre1(E);
    muestre1(P);
    muestre1(S);
    muestre1(O);
    muestre1(N);
    delayMicroseconds(sec6);
}
if (saltar==8){
    delayMicroseconds(sec7);
    muestre3(M);
    muestre3(O);
    muestre3(T);
    muestre3(O);
    muestre3(R);
    muestre3(O);
    muestre3(L);
    muestre3(A);
    delayMicroseconds(sec7);}
if (saltar==9){

```

```

    muestre2(lov);
    }
}
void muestrer1(int font[]) //primera fila
{
    int primera=0;
    for(int j=0; j<5; j++)
    {
        for (int i=0; i<8; i++)
        {
            digitalWrite(filal[i], font[i+(primera*8)]);
        }
        delayMicroseconds(Ancho);
        primera++;
    }
    for (int i=0; i<8; i++)
    {
        digitalWrite(filal[i],0);
    }
    delayMicroseconds(Espacio);
}

void muestre2(int font[]) //segunda fila
{
    int segunda=0;
    for(int j=0; j<5; j++)
    {
        for (int i=0; i<8; i++)
        {
            digitalWrite(fila2[i], font[i+(segunda*8)]);
        }
        delayMicroseconds(Ancho);
        segunda++;
    }
    for (int i=0; i<8; i++)
    {
        digitalWrite(fila2[i],0);
    }
    delayMicroseconds(Espacio);
}

void muestre3 (int fuente[]) //tercera fila
{
    int tercera=0;
    for(int j=0; j<5; j++)
    {
        for (int i=0; i<8; i++)
        {
            digitalWrite(fila3[i], fuente[i+(tercera*8)]);
        }
        delayMicroseconds(Ancho);
        tercera++;
    }
    for (int i=0; i<8; i++)
    {
        digitalWrite(fila3[i],0);
    }
    delayMicroseconds(Espacio);
}

```

# **Anexo 4:**

Proforma de materiales finales utilizados  
en el proyecto.

Listado general de materiales:

<b>Descripción</b>	<b>Cantidad</b>	<b>Precio Unitario \$</b>	<b>Precio total \$</b>
Motor de DC 24V	1	15.00	15.00
Arduino Leonardo	1	28.00	28.00
LEDs	18	0.15	2.70
Elementos para la fuente Variable de 0 V a 24 V.	1	12.50	12.50
Batería de 9 V	1	3.50	3.50
Cables y conectores	1 kit	3.50	3.50
Cable USB 2.0	1	3.00	3.00
Módulo Bluetooth HC-06	1	20.00	20.00
Estructura Acrílico	1	35.00	35.00
Base horizontal (madera)	1	3.00	3.00
Base para motor	1	12.00	12.00
		<b>Subtotal</b>	<b>138.20</b>
		<b>IVA 12%</b>	<b>16.58</b>
		<b>Total</b>	<b>154.78</b>

# **Anexo 5:**

Datasheet Bluetooth HC-06

The following figure 1 is a picture of HC-06 and its main PINs. Figure 2 is a picture of HC-05 and its main PINs. Figure 3 is a comparative picture with one coin. Figure 4 is their package size information. When user designs the circuit, you can visit the website of Guangzhou HC Information Technology Co., Ltd. ([www.wavesen.com](http://www.wavesen.com)) to download the package library of protle version.



Figure 1 HC-06

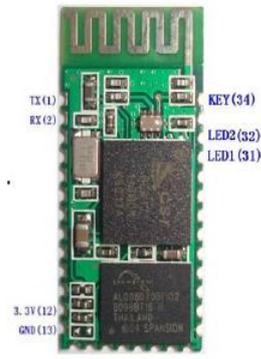


Figure 2 HC-05

This chapter will introduce the using method of HC-06 in detail. User can test the module according to this chapter when he or she uses the module at the first time.

PINs description:

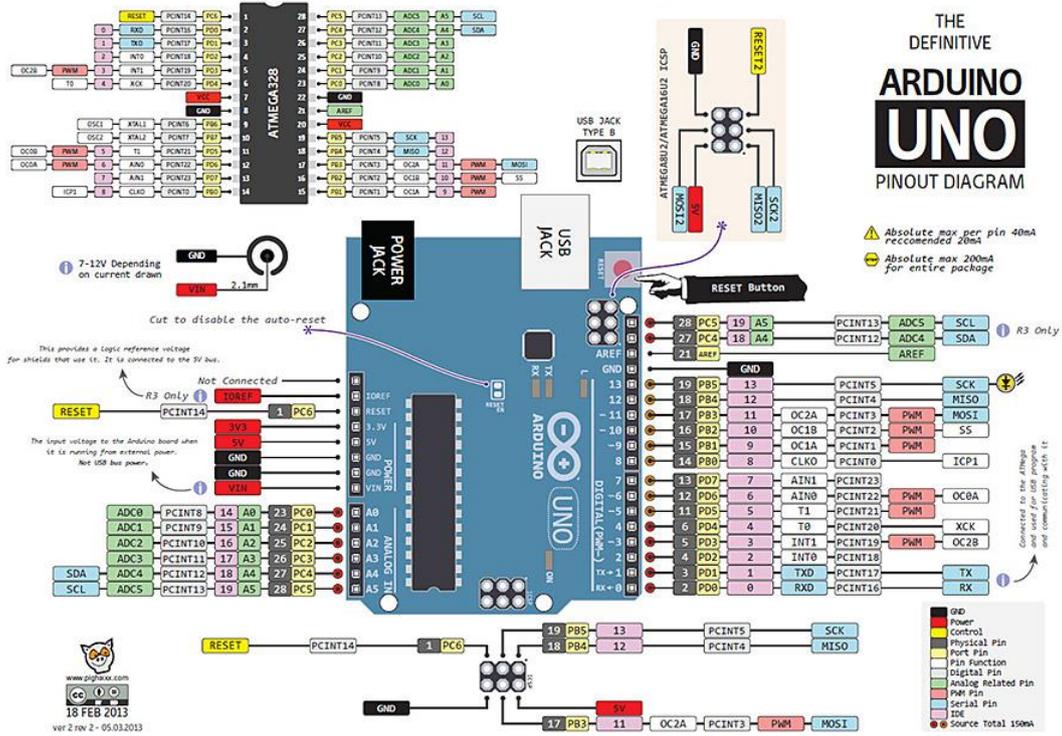
PIN1	UART_TXD , TTL/CMOS level, UART Data output
PIN2	UART_RXD, TTL/COMS level, s UART Data input
PIN11	RESET, the reset PIN of module, inputting low level can reset the module, when the module is in using, this PIN can connect to air.
PIN12	VCC, voltage supply for logic, the standard voltage is 3.3V, and can work at 3.0-4.2V
PIN13	GND
PIN22	GND

PIN24	LED, working mode indicator Slave device: Before paired, this PIN outputs the period of 102ms square wave. After paired, this PIN outputs high level. Master device: On the condition of having no memory of pairing with a slave device, this PIN outputs the period of 110ms square wave. On the condition of having the memory of pairing with a slave device, this PIN outputs the period of 750ms square wave. After paired, this PIN outputs high level.
PIN26	For master device, this PIN is used for emptying information about pairing. After emptying, master device will search slaver randomly, then remember the address of the new got slave device. In the next power on, master device will only search this address.

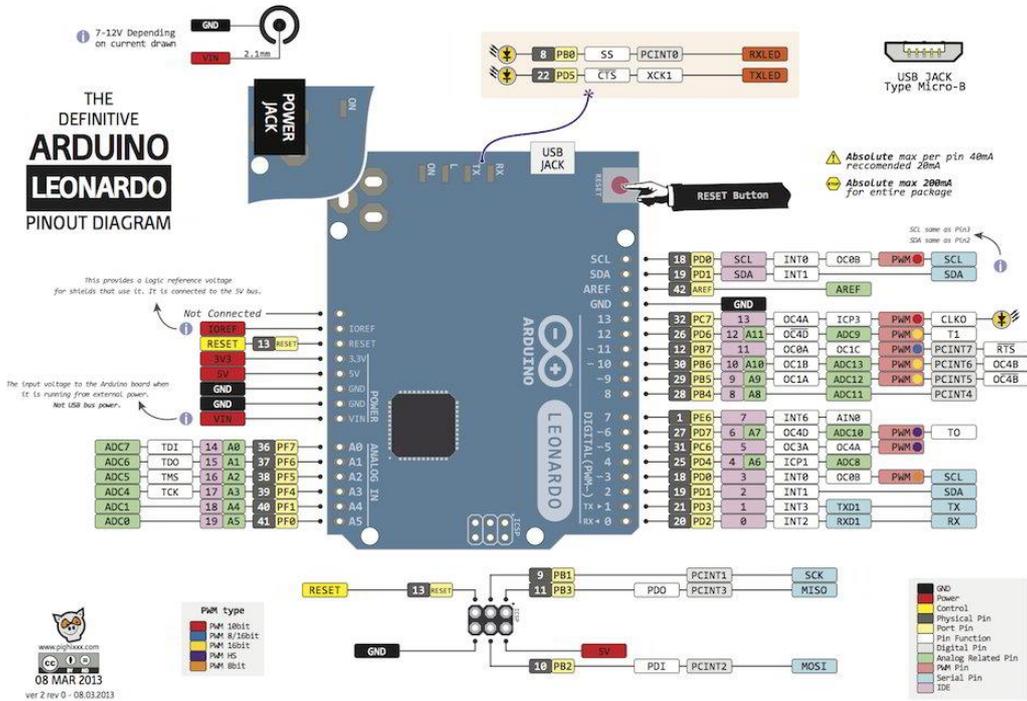
# **Anexo 6:**

Diagrama de pines de las placas:

- Arduino UNO



- Arduino Leonardo



# **Anexo 7:**

Guía de Lenguaje Arduino utilizado en el  
proyecto

## ESTRUCTURA

Setup().- La función setup() se establece cuando se inicia un programa -sketch. Se emplea para iniciar variables, establecer el estado de los pins, inicializar librerías, etc. Esta función se ejecutará una única vez después de que se conecte la placa Arduino a la fuente de alimentación, o cuando se pulse el botón de reinicio de la placa. Ejemplo:

```
int buttonPin = 3;
void setup()
{
  Serial.begin(9600);
  pinMode(buttonPin, INPUT);
}
void loop()
{
  // ...
}
loop()
```

Estructuras de control.- if (condicional) y == (igual a), != (distinto de), <(menor que), > (mayor que) (operadores de comparación) ; <= (menor o igual que); >= (mayor o igual que).

if, el cual puede ser usado en conjunto con uno o más operadores de comparación, comprueba si cierta condición se cumple, por ejemplo, si un input posee un valor mayor a cierto número. El formato para una comprobación if es el siguiente:

```
if (algunaVariable > 50)
{
  // hacer algo aquí.
}
```

// todos los ejemplos son correctos.

if/else.- permite mayor control sobre el flujo del código que la declaración if básica, por permitir agrupar múltiples comprobaciones. Por ejemplo, un input analógico podría ser comprobado, y tomarse una acción si el valor del inputes menor a 500, y, otra acción se ser igual o mayor a 500. El código se vería así:

```
if (pinCincoInput < 500)
{
  // acción A
}
else
{
  // acción B
}
```

Else.- puede proceder a una comprobación if, de esta forma, se pueden realizar múltiples comprobaciones en una misma estructura de condiciones. Cada comprobación procederá a la siguiente, sólo cuando su propio resultado sea FALSE. Cuando el resultado sea TRUE, su bloque de código contenido, será ejecutado, y el programa esquivará las siguientes comprobaciones hasta el final de la estructura de comprobaciones. Si ninguna comprobación devuelve valor TRUE, el else será ejecutado, y de no haber ninguno declarado, simplemente no sucede nada.

Entonces un bloque else if puede ser usado con o sin else al final. La cantidad de declaraciones else if, y sus ramificaciones son ilimitadas.

```
if (pinCincoInput < 500)
{
  // ejecutar A
}
else if (pinCincoInput >= 1000)
{
  // ejecutar B
}
else
{
  // ejecutar C
}
```

Declaracion FOR.- La declaración for es usada para repetir un bloque encerrado entre llaves. Un incremento de un contador es usado, normalmente, para aumentar

y terminar con el bucle. La estructura for es muy útil para la mayoría de las operaciones repetitivas, y habitualmente se usa para operaciones con vectores, para operar sobre conjuntos de datos/pines

El bucle for tiene tres partes o argumentos en su inicialización:

```
for (initialization; condition; increment) {  
  //función(es);  
}
```

```
for(int x = 0; x < 100; x++){  
    println(x); // prints 0 to 99  
}
```

Ejemplo:

```
// Variar la intensidad de un LED usando un salida PWM  
int PWMpin = 10; // En el pin 10 hay un LED en serie con una resistencia de 470  
ohmios  
void setup()  
{  
  // no es necesario nada aquí  
}  
void loop()  
{  
  for (int i=0; i <= 255; i++){  
    analogWrite(PWMpin, i);  
    delay(10);  
  }  
}
```

Sentencia switch / case.- Como las sentencias if, switch...case controla el flujo de programas permitiendo a los programadores especificar diferentes códigos que deberían ser ejecutados en función de varias condiciones. En particular, una sentencia switch compara el valor de una variable con el valor especificado en las sentencias case. Cuando se encuentra una sentencia case cuyo valor coincide con dicha variable, el código de esa sentencia se ejecuta.

La palabra clave break sale de la sentencia switch, y es usada típicamente al final de cada case. Si una sentencia break, la sentencia switch continuaría ejecutando las siguientes expresiones ("falling-through") hasta encontrar un break, o hasta llegar al final de la sentencia switch.

Ejemplo

```
switch (var) {  
  case 1:  
    //hacer algo cuando sea igual a 1  
    break;  
  case 2:  
    //hacer algo cuando sea igual a 2  
    break;  
  default:  
    // si nada coincide, ejecuta el "default"  
    // el "default" es opcional  
}
```

Bucles while.- Los bucles while se ejecutan continuamente, hasta que la expresión de dentro del paréntesis, (), pasa a ser falsa. Algo debe modificar la variable comprobada, el bucle while nunca terminará. Lo que modifique la variable puede estar en el código, como una variable que se incrementa, o ser una condición externa, como el valor que da un sensor.

Ejemplo:

```
var = 0;  
while(var < 200){
```

```

// haz algo repetitivo 200 veces
var++;
}

```

do - while.- El bucle "do" trabaja de la misma manera que el bucle "while", con la excepci3n de que la condici3n se comprueba al final del bucle, por lo que este bucle se ejecuta "siempre" al menos una vez.

```

do
{
// bloque de instrucciones
} while (condici3n);

```

Ejemplo:

```

do
{
delay(50); // espera a que los sensores se estabilicen
x = readSensors(); // comprueba los sensores
} while (x < 100); //si se cumple la condici3n se repite el bucle
break

```

break es usado para salir de los bucles do, for, o while, pasando por alto la condici3n normal del bucle. Es usado tambi3n para salir de una estructura de control switch.

Ejemplo

```

for (x = 0; x < 255; x ++)
{
digitalWrite(PWMPin, x);
sens = analogRead(sensorPin);
if (sens > threshold){ // bail out on sensor detect
x = 0;
break; // sale del bucle for.
}
delay(50);
}

```

Comentarios.- Los comentarios son l3neas en el programa para aclarar a t3 o a otros sobre el funcionamiento del programa. Estas l3neas son ignoradas por el compilador y no se exportan al procesador. No ocupan por tanto espacio en el Chip Atmega

El 3nico prop3sito de los comentarios es que entiendas o entiendan (o recuerdes) cual es el funcionamiento de tu programa. Existen dos maneras distintas de marcar una l3nea como comentario:

Ejemplo

```

x = 5; // Esto es una l3nea simple de comentario. Todo lo que va despu3s de
la doble barra es un comentario
// Hasta el final de la l3nea

```

```

/* Esto es un comentario multil3nea - 3salo para comentar bloques enteros de
c3digo

```

```

if (gwb == 0){ // Una l3nea de comentario sencilla puede usarse dentro de un
comentario multil3nea
x = 3; // pero no otro comentario multil3nea- esto no es v3lido */
}
// No olvides cerrar el comentario multil3nea
*/

```

Operaciones aritm3ticas.- = operador de asignaci3n (un solo s3mbolo de "igual")  
Guarda el valor en la derecha del s3mbolo "=" dentro de la variable a la izquierda del s3mbolo "=".

El signo de igualdad "=" en el lenguaje de programaci3n C se llama el operador de asignaci3n. Tiene un significado diferente que en la clase de 3lgebra en el que se indica una ecuaci3n o igualdad. El operador de asignaci3n le dice al microcontrolador que eval3e cualquier valor o expresi3n en el lado derecho del signo igual, y lo almacene en la variable a la izquierda del signo igual.

Ejemplo:

```

int sensVal; // declara una variable int llamada sensVal
sensVal = analogRead(0); // guarda el valor (digitalizado) del voltaje de
entrada del pin anal3gico 0 en SensVal
Suma, Resta, Multiplicaci3n y Divisi3n

```

## Descripción

Estos operadores devuelven la suma, diferencia, producto o cociente (respectivamente) de los dos operandos. La operación se lleva a cabo utilizando el tipo de datos de los operandos, por lo que, por ejemplo,  $9 / 4$  resulta 2 desde 9 y 4 que son enteros int. Esto también significa que la operación puede desbordarse si el resultado es mayor que el que se puede almacenar en el tipo de datos (por ejemplo, la suma de 1 a un int con el valor de 32.767 resulta -32.768). Si los operandos son de tipos diferentes, se utiliza el tipo del "más grande" para el cálculo.

Si uno de los números (operandos) es del tipo float o del tipo double, se usará coma flotante para el cálculo.

## Ejemplos

```
y = y + 3;
x = x - 7;
i = j * 6;
r = r / 5;
```

## Sintaxis

```
result = value1 + value2;
result = value1 - value2;
result = value1 * value2;
result = value1 / value2;
```

## Parámetros:

value1: cualquier variable o constante

value2: cualquier variable o constante

## Sugerencias de programación:

Debes saber que las integer constants por defecto son int, así que algunos cálculos con constantes pueden provocar desbordamiento (p.e.  $60 * 1000$  devolverá un resultado negativo)

Elige los tamaños de variable que sean suficientemente grandes como para alojar el resultado de tus calculos.

Debes saber en que punto tu variable se desbordará en su máximo, y que esto también ocurre en su mínimo. p.e.  $(0 - 1)$  o también  $(0 - -32768)$

Para cálculos matemáticos que requieren fracciones, usa variables float, pero ten en cuenta los inconvenientes: gran tamaño, velocidades bajas de cálculo

Usa el operador de conversión (casting). Por ejemplo: `(int)myFloat` para convertir el tipo de una variable en el momento.

## % (módulo)

### Descripción

Calcula el resto de la división entre dos enteros. Es útil para mantener una variable dentro de un rango particular (por ejemplo el tamaño de un array)

### Ejemplo

```
x = 7 % 5;    // x ahora contiene 2
x = 9 % 5;    // x ahora contiene 4
x = 5 % 5;    // x ahora contiene 0
x = 4 % 5;    // x ahora contiene 4
```

### Código de ejemplo

```
/* actualiza un valor en el array cada vez que se pasa por el bucle */
int valores[10];
int i = 0;
void setup() {}
void loop()
{
    valores[i] = analogRead(0);
    i = (i + 1) % 10;    // el operador módulo prevalece sobre la variable
}
OPERACIONES COMPUESTAS += , -= , *= , /=
```

## Descripción

Realiza una operación matemática con una variables con respecto a otra variable o una constante. El operador += (y los demás operadores) son una forma simplificada de la sintaxis completa, tal y como se muestra más abajo.

## Sintaxis

```
x += y;    // equivalente a la expresión x = x + y;
x -= y;    // equivalente a la expresión x = x - y;
x *= y;    // equivalente a la expresión x = x * y;
x /= y;    // equivalente a la expresión x = x / y;
```

Ejemplos

```
x = 2;
x += 4;      // x ahora es 6
x -= 3;      // x ahora es 3
x *= 10;     // x ahora es 30
x /= 2;      // x ahora es 15
```

HIGH.- El significado de HIGH (en referencia a un pin) depende de si el pin está configurado como entrada (INPUT) o como salida (OUTPUT). Cuando un pin se configura como entrada (INPUT) usando pinMode, y se lee con digitalRead, el microcontrolador nos retornará HIGH si en el pin hay 3 voltios o más.

Un pin puede ser configurado como entrada (INPUT) usando pinMode, y después establecerlo a HIGH con digitalWrite, esto conectará el pin a 5 Voltios a través de una resistencia interna de 20K, resistencia pull-up, la cual establecerá el pin al estado de lectura HIGH a menos que la conectemos a una señal LOW a través de un circuito externo.

Cuando un pin se configura como salida (OUTPUT) con pinMode, y se establece a HIGH con digitalWrite, el pin tiene 5V. En este estado puede usarse como fuente de corriente, e.j. Luz y LED que se conectan a través de resistencias en serie a masa (tierra), o a otro pin configurado como salida y establecido a LOW.

LOW.-El significado de LOW difiere también según esté configurado como entrada (INPUT) o como salida (OUTPUT). Cuando un pin está configurado como entrada (INPUT) con pinMode, y se lee con digitalRead, el microcontrolador retornará LOW si el voltaje presente en el pin es de 2V o menor.

Cuando un pin es configurado como salida (OUTPUT) con pinMode, y establecido LOW con digitalWrite, el pin tiene 0 voltios. En este estado puede meter corriente, e.j. Luz y LED que se conectan a través de resistencias en serie a +5 voltios, o a otro pin configurado como salida, y establecido a HIGH.

Las que definen los pines digitales, INPUT y OUTPUT

Los pines digitales pueden ser usados como entrada (INPUT) o como salida (OUTPUT). Cambiando un pin de INPUT a OUTPUT con pinMode() el comportamiento eléctrico del pin cambia drásticamente.

digitalWrite(pin, valor)

Ejemplo

```
int ledPin = 13;           // LED conectado al pin digital 13

void setup()
{
  pinMode(ledPin, OUTPUT); // establece el pin digital como salida
}

void loop()
{
  digitalWrite(ledPin, HIGH); // enciende el LED
  delay(1000);                // espera por un segundo
  digitalWrite(ledPin, LOW);  // apaga el LED
  delay(1000);                // espera por un segundo
}
```

Establece el pin número 13 a HIGH, hace un retraso con la duración de un segundo, y regresa el pin a LOW.

Nota

Los pines analogicos pueden ser usados como pines digitales con los números 14 (entrada analogica número 0) hasta 19 (entrada analogica número 5).

digitalRead()

Descripción

Lee el valor de un pin digital especificado, HIGH o LOW.

Sintaxis

digitalRead(pin)

Parámetros

pin: el número de pin digital que quieres leer (int)

Devuelve

HIGH o LOW

Ejemplo

```

int ledPin = 13; // LED conectado al pin digital número 13
int inPin = 7;   // botón (pushbutton) conectado al pin digital número 7
int val = 0;     // variable donde se almacena el valor leído

void setup()
{
  pinMode(ledPin, OUTPUT);    // establece el pin digital número 13 como salida
  pinMode(inPin, INPUT);     // establece el pin digital número 7 como entrada
}

void loop()
{
  val = digitalRead(inPin); // leer el pin de entrada
  digitalWrite(ledPin, val); // establece el LED al valor del botón
}

```

### **E/S Analógica**

`analogWrite()`.- Escribe un valor analógico (PWM) en un pin. Puede ser usado para controlar la luminosidad de un LED o la velocidad de un motor. Después de llamar a la función `analogWrite()`, el pin generará una onda cuadrada estable con el ciclo de trabajo especificado hasta que se vuelva a llamar a la función `analogWrite()` (o una llamada a las funciones `digitalRead()` o `digitalWrite()` en el mismo pin). La frecuencia de la señal PWM será de aproximadamente 490 Hz. En la mayoría de las placas Arduino (aquellas con el ATmega168 o ATmega328), se podrá generar señales PWM en los pines 3, 5, 6, 9, 10, y 11. En la placa Arduino Mega, se puede llevar a cabo con los pines desde el 2 hasta el pin 13. Las placas Arduino más antiguas que posean el chip ATmega8 solo podrán usar la función `analogWrite()` con los pines 9, 10 y 11. No hace falta configurar el pin como salida para poder usar la función `analogWrite()`. La función `analogWrite` no tienen ninguna relación con los pines de entrada analógicos ni con la función `analogRead`.

Sintaxis

```
analogWrite(pin, valor)
```

### **TIEMPO**

#### **micros()**

Descripción

Devuelve el número de microsegundos desde que la placa Arduino comenzó a correr el programa. Este número eventualmente volverá a 0 (overflow), luego de aproximadamente 70 minutos. A 16 MHz, en las placas Arduino (por ejemplo, Duemilanove ó Nano), éstas función tiene una resolución de cuatro milisegundos (por ejemplo, el valor devuelto es siempre múltiplo de 4). A 8 MHz, en placas Arduino (como LilyPad), tiene una resolución de 8 microsegundos.

Ejemplo

```
unsigned long time;
```

```

void setup(){
  Serial.begin(9600);
}
void loop(){
  Serial.print("Tiempo: ");
  time = micros();
  //Imprime el tiempo desde el comienzo del programa
  Serial.println(time);
  // espera un segundo para no enviar datos una cantidad exagerada de datos.
  delay(1000);
}

```

#### **millis()**

Descripción

Devuelve el tiempo en milisegundos transcurridos desde que se arranco la placa Arduino con el programa actual. Este número de desbordará (volverá a cero), después de aproximadamente 50 días.

Parámetros

Ninguno

Devuelve

Tiempo en milisegundos desde que el programa se inició (long sin signo (unsigned long))

Ejemplo

```
unsigned long time;
```

```
void setup(){
  Serial.begin(9600);
}
void loop(){
  Serial.print("Tiempo: ");
  time = millis();
  //Imprime el tiempo desde que se inició el programa
  Serial.println(time);
  // espera un segundo para no enviar demasiados datos
  delay(1000);
}
```

### **delay()**

Descripción

Pausa el programa por un tiempo determinado (en milisegundos) especificado por un parámetro. Hay 1000 milisegundos en un segundo.

Ejemplo

```
int ledPin = 13; // LED conectado al pin digital 13.
```

```
void setup()
{
  pinMode(ledPin, OUTPUT); // declara que el pin digital se va a usar como
  salida
}
void loop()
{
  digitalWrite(ledPin, HIGH); // enciende el LED
  delay(1000); // espera durante un segundo
  digitalWrite(ledPin, LOW); // apaga el LED
  delay(1000); // espera durante un segundo
}
```

### **delayMicroseconds()**

Descripción

Detiene brevemente el programa por la cantidad en tiempo (en microsegundos) especificada como parámetro. Existen mil microsegundos en un milisegundo, y un millon de microsegundos en un segundo.

Actualmente, el valor más grande producirá un retraso exacto es 16383. Esto puede cambiar en una futura versión de Arduino. Para retrazos más largos que algunos miles de microsegundos, deberías usar delay() .

Ejemplo

```
int outPin = 8; // selecciona el pin 8

void setup()
{
  pinMode(outPin, OUTPUT); // establece el pin digital como salida
}

void loop()
{
  digitalWrite(outPin, HIGH); // establece en encendido el pin
  delayMicroseconds(50); // espera por 50 microsegundos
  digitalWrite(outPin, LOW); // establece en apagado el ping
  delayMicroseconds(50); // espera por 50 microsegundos
}
```

# **Anexo 8:**

Documento entrega de la implementación

UNIVERSIDAD TECNOLÓGICA ISRAEL

Presente.-

De mi consideración:

Por medio de la presente queremos comunicar que el Señor Juan Carlos Vásquez Revelo, con C.I 0401222690 ha venido realizando su PROYECTO INTEGRADOR DE CARRERA "Diseño e implementación de un dispositivo electrónico para visualizar caracteres en general de manera alterna, en la publicidad de marcas representadas en el Ecuador por la empresa UNISCAN CIA. LTDA."; en donde se le ha proporcionado toda la información y ayuda necesaria para el desarrollo de su proyecto.

Atentamente;



**UNISCAN Cía. Ltda.**  
RUC: 1791937384001

José Luis Trujillo

Jefe Departamento Técnico UNISCAN

Contacto: (+593) 02 3330597 / Cel: (+593) 984423373

Email: [jtrujillo@uniscan.com.ec](mailto:jtrujillo@uniscan.com.ec)