

**ADECUACIÓN E IMPLEMENTACIÓN DE UN DISPOSITIVO ELECTRÓNICO QUE
SIRVA COMO GUÍA DE DESPLAZAMIENTO DE LA COMUNIDAD INVIDENTE**

JUAN CAMILO OSORIO CORRALES

**INSTITUCIÓN UNIVERSITARIA PASCUAL BRAVO
FACULTAD DE INGENIERÍA
TECNOLOGÍA ELECTRÓNICA
MEDELLÍN
2015**

**ADECUACIÓN E IMPLEMENTACIÓN DE UN DISPOSITIVO ELECTRÓNICO QUE
SIRVA COMO GUÍA DE DESPLAZAMIENTO DE LA COMUNIDAD INVIDENTE**

JUAN CAMILO OSORIO CORRALES

Trabajo de grado para optar al título de Tecnólogo Electrónico

Asesora

Vesna Srdanovic

Ingeniera Electrónica con Maestría en Telecomunicaciones

INSTITUCIÓN UNIVERSITARIA PASCUAL BRAVO

FACULTAD DE INGENIERÍA

TECNOLOGÍA ELECTRÓNICA

MEDELLÍN

2015

Contenido

Resumen.....	8
Abstract.....	9
Glosario.....	10
Introducción	11
1. Planteamiento del problema	12
1.1. Descripción.....	12
1.2. Formulación	13
2. Justificación.....	14
3. Objetivos.....	16
3.1. General.....	16
3.2. Específicos.....	16
4. Marco teórico.....	17
4.1. Los dispositivos existentes de ayuda para personas invidentes.....	17
4.1.1. El bastón blanco	17
4.1.2. Marcas táctiles	17
4.1.3. Bastón inteligente	18
4.1.4. Muñeca detectora de objetos.....	19
4.1.5. El bastón virtual	19
4.1.6. Gafas para invidentes.....	20
4.1.7. Chip que ayuda a los ciegos a ver formas y tonalidades.....	21
4.2. Sensores.....	21
4.2.1. Sensores de proximidad.....	22
4.2.2. Sensores ultrasónicos	22
4.2.3. Sensor detector de línea	24
4.3. Tecnología de Arduino	26
4.3.1. Arduino UNO.....	26
4.4. Microcontrolador atmega328p-pu	27
4.5. Buzzer.....	29
4.6. Relé	29

4.7.	Motores DC	30
4.7.1.	Motor vibrador.....	31
4.8.	Fuente de alimentación	32
4.8.1.	La batería.....	32
5.	Metodología.....	34
5.1.	Tipo de estudio	34
5.2.	Método	34
5.3.	Instrumentos de recolección de información.....	35
5.3.1.	Fuentes primarias	35
5.3.2.	Fuentes secundarias.....	35
6.	Resultados.....	36
7.	Conclusiones.....	48
8.	Recomendaciones	49
9.	Referencias bibliográficas	50
10.	Bibliografía	52
11.	Anexos	53

Lista de figuras

<i>Figura 1.</i> Bastón blanco.....	17
<i>Figura 2.</i> Marcas táctiles.....	18
<i>Figura 3.</i> Bastón inteligente.....	18
<i>Figura 4.</i> Guante háptico para ciegos.....	19
<i>Figura 5.</i> Bastón virtual.....	20
<i>Figura 6.</i> Gafas para ciegos.....	20
<i>Figura 7.</i> Chip visual.....	21
<i>Figura 8.</i> Sensor sonar.....	22
<i>Figura 9.</i> Sensor de distancia HC-SR04.....	23
<i>Figura 10.</i> Sensor óptico.....	24
<i>Figura 11.</i> Diagrama eléctrico del sensor infrarrojo.....	25
<i>Figura 12.</i> Sensor óptico CNY70.....	25
<i>Figura 13:</i> tarjeta de Arduino UNO.....	26
<i>Figura 14:</i> microcontrolador ATMEGA328P-PU.....	27
<i>Figura 15:</i> Forma física de un buzzer.....	29
<i>Figura 16:</i> Presentación esquemática y física de un relé.....	30
<i>Figura 17:</i> Motor DC y sus partes.....	31
<i>Figura 18:</i> Presentación física de un motor vibrador.....	32
<i>Figura 19:</i> Presentación física de la batería tipo secundario.....	33
<i>Figura 20:</i> esquema de sensor infrarrojo.....	36
<i>Figura 21:</i> conexión de sensor infrarrojo.....	37
<i>Figura 22:</i> conexión Arduino y sensor de proximidad.....	38
<i>Figura 23:</i> Programación en Arduino.....	39
<i>Figura 24:</i> Pruebas del proyecto con tarjeta Arduino.....	40
<i>Figura 25:</i> Conexión para cargar Bootloader	41
<i>Figura 26:</i> Arduino como ISP.....	41
<i>Figura 27:</i> Quemar Bootloader.....	42
<i>Figura 28:</i> Practica de grabar secuencia de inicio al microcontrolador.....	42
<i>Figura 29:</i> Circuito para el funcionamiento independiente del microcontrolador.....	43
<i>Figura 30:</i> Configuración de los pines del microcontrolador con la tarjeta Arduino.....	43

<i>Figura 31: Montaje de circuito y pruebas.....</i>	<i>44</i>
<i>Figura 32: Diseño para circuito impreso.....</i>	<i>44</i>
<i>Figura 33: Circuito impreso.....</i>	<i>45</i>
<i>Figura 34: Circuito impreso con los componentes.....</i>	<i>45</i>
<i>Figura 35: Pruebas con circuito antes de montar en el bastón.....</i>	<i>46</i>
<i>Figura 36: Montaje del circuito en el diseño de bastón.....</i>	<i>46</i>
<i>Figura 37: bastón listo para realizar pruebas.....</i>	<i>47</i>

Lista de anexos

Anexo A. Listado del programa del Bootloader.....	52
Anexo B. Diagrama esquemáticos del circuito electrónico.....	61
Anexo C. Fotos de pruebas	64

Resumen

En este proyecto se presenta el desarrollo de un prototipo de bastón para las personas con discapacidad visual. En la construcción de este bastón son utilizados dos tipos de sensores y microcontrolador para garantizar el desplazamiento de estas personas. Ubicando el dispositivo sobre tres líneas de diferentes colores y cuando el invidente este abandonando alguna de las líneas, el dispositivo va emitir unas vibraciones en diferente frecuencia para advertir que la persona debe moverse hacia la derecha o izquierda. Además, este bastón cuenta con sistema detector de obstáculos y un sistema de detección de cruces de las líneas o de irregularidad de terreno.

Palabras claves: bastón para invidentes, sensor IR, sensor de ultrasonido, Arduino, actuadores.

Abstract

In this project is presented the development of a prototype of the walking stick for persons with visual disabilities. In the construction of this walking stick are used two types of sensors and microcontroller to ensure the movement of these people. Placing the device on the three lines of different colors and when the blind is abandoning any of the lines, the device will emit different frequency vibrations to warn that the person should move to the right or to the left. Besides this walking stick has obstacle detection system and one system of crossings detection of the lines or of the irregular terrain.

Keywords: walking stick for the blind, IR sensor, ultrasonic sensor, Arduino, actuators.

Glosario

Actuadores: Son dispositivos mecánicos que proporcionan una fuerza para mover o actuar otro dispositivo mecánico. Dependiendo del origen de la fuerza los actuadores se clasifican en “neumáticos”, “hidráulicos” y “eléctricos”.

Bastón: Es una herramienta táctil que sirve de guía para los ciegos, ayudándoles a distinguir la superficie por la cual se están desplazando, a su vez ayuda un poco con su ubicación.

Dispositivo electrónico: Constan de una combinación de diferentes elementos o componentes organizados en circuitos, para controlar y aprovechar señales eléctricas.

Invidente: Persona que carece del sentido de la visión, ciego.

Programación: Hace referencia al efecto de programar, es decir, de organizar una secuencia de pasos ordenados a seguir para ejecutar una acción.

Sensores de ultrasonidos: son detectores de proximidad que pueden medir desde 1cm hasta varios metros, el sensor emite un sonido y cuenta el tiempo que este tarda en regresar, lo que hace es la conversión de tiempo distancia.

Sensor infrarrojo: son aquellos que constan de un transmisor y un receptor; cuando el transmisor emite una señal y esta es percibida por el receptor, entra en funcionamiento el circuito al cual está conectado el receptor.

Introducción

Las personas invidentes enfrentan a diario muchas dificultades para hacer sus desplazamientos, ellos saben a dónde quieren llegar mas no cómo llegar; teniendo en cuenta que son muchos los obstáculos por los que tienen que pasar.

Aplicando los avances tecnológicos en el campo de sensores y microcontroladores se construyó un prototipo de bastón que servirá como ayuda de desplazamiento de las personas invidentes. Son tres sensores infrarrojo que detectan el cambio de color de tres líneas marcadas en el piso; las señal que emiten cada uno de los sensores llegan a un microcontrolador que lleva en su programación los posibles eventos que puedan presentarse durante un recorrido sobre las líneas guías. En conjunto con los actuadores el sistema electrónico indica al invidente si se está saliendo de la línea a la derecha, a la izquierda o si hay un cruce o irregularidad. Este prototipo de bastón también lleva un sensor de ultrasonido que comunica al microcontrolador, cada vez que haya un obstáculo, y se genera una alarma que advierte a la persona para que se detenga.

Es de gran importancia la implementación de la tecnología para ayudar las personas con alguna discapacidad, en este caso las personas invidentes. Este prototipo ayudara a sus desplazamientos, a su ubicación y a evitar los tropezones; todo esto favorecerá a estas personas para que sean un poco más independientes

Este proyecto está enfocado en facilitar el desplazamiento de personas invidentes, específicamente en la ciudad de Medellín. Una ventaja importante es la económica, ya que este dispositivo no acarreará grandes costos, los elementos utilizados son de bajo costo y trabajan adecuadamente, además la implementación de las líneas en el piso no implica el cambio arquitectónico de la ciudad ya que solo habría que pintarlas por los andenes o en los lugares públicos donde estas personas necesitan desplazarse.

1. Planteamiento del problema

1.1.Descripción

Son muchas las personas infortunadas, que por diferentes motivos sufren de una discapacidad física, aunque son sorprendentes todas las habilidades que estas personas desarrollan y todo lo que hacen para llevar una mejor vida. La humanidad ha venido desarrollando nuevos dispositivos tecnológicos, en pro, de facilitar las actividades de estas personas. En el caso específico de las personas que carecen de visión, es sabido que ellos enfrentan un sinnúmero de dificultades para desplazarse de un lugar a otro. En la búsqueda de dispositivos que faciliten esta actividad cotidiana de ellos, el inventor argentino, José Mario Fallotico, diseñó un bastón denominado el Bastón Blanco, el cual tenía la función de guiar al invidente y contribuir en la facilidad de sus desplazamientos. Las funciones de este dispositivo son muy limitadas. La persona que utiliza este bastón no detecta las barreras físicas que se presentan en el recorrido, hasta que tiene contacto directo con el obstáculo mediante un golpe.

Con los avances tecnológicos, en varios países se han diseñado dispositivos que advierten con anterioridad cuando hay un obstáculo en frente de la persona que lo está portando. Son diferentes diseños, pero su finalidad en común es la misma, prevenir que la persona invidente tropiece con alguno de estos. Son dispositivos muy útiles y han ayudado en cierta forma a las personas que carecen de visión en sus desplazamientos, pero por ser dispositivos creados en el exterior, la comunidad invidente de Colombia no se ha visto muy beneficiada por sus altos costos.

En Colombia, y en particular la en ciudad de Medellín en los lugares más comunes se cuenta con unas marcas arquitectónicas táctiles que sirven de guía para estas personas; sus diferentes formas les indican la superficie por la cual debe caminar. El problema reside que las marcas táctiles se encuentran solo en lugares específicos y no en todos los lugares de la ciudad y para suplir esta necesidad sería necesario cambiar gran parte de la arquitectura pública.

Por lo anterior; el motivo de este proyecto es la adecuación y la implementación de un dispositivo detector de obstáculos y que además sirva como guía y mejore las condiciones de

desplazamiento de la comunidad invidente. El prototipo de este bastón para los invidentes va ser de bajo costo y accesible para la mayoría de Las personas invidentes de la ciudad.

1.2.Formulación

¿Si con aplicación de los sensores y microcontroladores se puede desarrollar un nuevo prototipo que sirva como guía y mejore las condiciones de desplazamiento de la comunidad invidente?

2. Justificación

En la actualidad se buscan nuevas alternativas tecnológicas que faciliten la vida de aquellas personas que carecen de alguna parte de sus cuerpos. En el caso específico de los invidentes existe una gran cantidad de dispositivos, que la humanidad ha creado para mejorar calidad de vida de estas personas. Lamentablemente hoy en día, a pesar de los esfuerzos de la sociedad, no se ha logrado desarrollar un dispositivo de bajo costo, accesible a todos los invidentes que habitan una ciudad, en este caso específico la ciudad de Medellín.

Las ayudas para el desplazamiento de los invidentes de la ciudad de Medellín son muy limitadas. La ciudad de Medellín cuenta con una guía arquitectónica para los invidentes, en cierta parte de la ciudad. La falencia de este es que estas estructuras solo están ubicadas en los lugares más comunes de Medellín, estos es, en los principales sitios. Por ende fue conveniente la creación de un dispositivo prototipo de fácil manejo y bajo costo, accesible a todos los ciudadanos invidentes de la ciudad.

La ventaja de este prototipo es que no implica modificaciones arquitectónicas en la ciudad, hecho que permite instalar las líneas de tres colores que guiaran los invidentes por los lugares de mayor importancia para ellos. Además este dispositivo advierte con anterioridad al usuario invidente sobre los obstáculos que se le van presentado por el camino, así hace que los desplazamientos de ellos dejen de ser tan complejos como lo es hoy en día.

En la actualidad si existen bastones con sensores que advierten a los usuarios de los obstáculos, pero han sido creados en el exterior, lo que ha implicado la importación de estos, haciéndolos de difícil alcance para la población invidente de la ciudad de Medellín. Por otra parte, los bastones existentes no cuentan con las líneas guías, que se presentan en este diseño, que hace que sea un prototipo inclusivo, esto es, un diseño pensado para que hasta los más pobres de la ciudad sean guiados por este dispositivo de una manera rápida y cómoda.

En conclusión, fue viable el diseño de un prototipo que detecta obstáculos y es guiado por un sistema de sensores que mejora la problemática de movilidad de la comunidad invidente porque

servirá a todas las personas ciegas de la ciudad, sin importar su condición social; además es muy práctico ya que su aplicación en la ciudad no requiere de cambios arquitectónicos, que lo hace muy económico a la hora de su implementación.

3. Objetivos

3.1.General

Diseñar un bastón que detecta obstáculos y sea guiado por un sistema de sensores que mejore la problemática de movilidad de la comunidad invidente de la ciudad de Medellín.

3.2.Específicos

Rastrear teorías, diseños e inventos en torno a los dispositivos tecnológicos que facilitan la movilidad de personas invidentes.

Seleccionar el modelo de bastón, sensores y actuadores que se adecuen al contexto específico.

Programar el microcontrolador utilizando la tarjeta de desarrollo ARDUINO.

Ensamblar y probar el funcionamiento del bastón detector de obstáculos, guiado por las líneas de tres colores.

Presentar el prototipo de bastón detector de obstáculos que facilite la movilidad de personas invidente de la ciudad de Medellín.

4. Marco teórico

4.1. Los dispositivos existentes de ayuda para personas invidentes

4.1.1. El bastón blanco

El argentino José Mario Fallotico fue el primero en crear una ayuda para las personas invidentes, el Bastón Blanco es un dispositivo que ha mejorado la forma de desplazamiento y detección de obstáculos. Quien patentó el invento fue el estadounidense George Benham. (Fallotico, 2006) En la Figura 1 se puede observar un Bastón Blanco.



Figura 1. Bastón blanco

Fuente: extraído de <http://slideplayer.es/slide/2273611/>

4.1.2. Marcas táctiles

Otra de las ayudas creadas para las personas invidentes son las marcas táctiles, estas deben de tener unas especificaciones y una buena distribución para que cumplan el objetivo de servir de guía. Son muchos los lugares que cuentan con estas ayudas, aunque muchos de estos no las tienen adecuadas de la mejor manera y por ayudar, más fácil terminan confundiendo a la persona que lo esté utilizando. Es conveniente que todo lugar este demarcado en forma adecuada por estas guías, que el invidente no tenga problemas en ir a diferentes lugares. Sin embargo lograr este objetivo es muy difícil ya que la estructura de la ciudad cada vez es más grande. En la Figura 2. se pueden observar *Marcas Táctiles* ubicadas en la ciudad de Medellín.

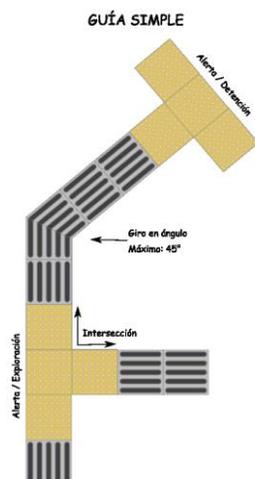


Figura 2. Marcas táctiles

Fuente: extraído de <file:///C:/Users/Personal/Downloads/CiudadInclusiva.pdf>

4.1.3. Bastón inteligente

En la Universidad Miguel Hernández de Elche (España) y con un diseño similar en la Universidad de Arkansas en Little Rock (Estados Unidos) se fabricó un bastón que detecta los obstáculos antes de estar en contacto con ellos y que le indica al individuo que lo porta, que debe cambiar su dirección de movilidad para que no fuese a tropezar con el mismo. Ha sido uno de los adelantos tecnológicos más novedosos y de gran ayuda en los últimos tiempos. En Colombia también se han creado algunos dispositivos basados en los mismos principios. (Tendencias Tecnológicas, 2013)



Figura 3. Bastón inteligente

Fuente: extraído de http://www.tendencias21.net/Nuevo-baston-inteligente-ayuda-a-los-ciegos-a-evitar-obstaculos_a20626.html

4.1.4. Muñeca detectora de objetos

Este dispositivo creado por Steve Hofer, percibe los obstáculos que se encuentran cerca de la mano de quien la está portando, cada vez que detecta algún objeto emite unas vibraciones que indican que está cerca de este (ver Figura 4). Este aparato cuenta con cuatro sensores de proximidad, un microcontrolador, una batería y dos motores de vibración. (Hofer, 2011)



Figura 4. Guante háptico para ciegos

Fuente: extraído de <http://www.cuentamealgobueno.com/2011/08/tacit-un-sonar-para-ciegos/>

4.1.5. El bastón virtual

Recientemente el desarrollo de una investigación revolucionó el uso de bastones para personas con baja visión. Un equipo de científicos de la Universidad Hebrea de Jerusalén, bajo la dirección de Dr. Amir Amedi, ha desarrollado un sistema revolucionario que probablemente va a cambiar la vida de millones de ciegos y personas con baja visión en todo el mundo. El funcionamiento de este se basa en un bastón con cámara incluida que toma fotos del entorno, donde se encuentra el invidente y las transforma en sonido; así le informa a quien lo esté portando del tamaño de los obstáculos que se encuentran en su recorrido. (Opticos Web, 2013)



Figura 5. Bastón virtual

Fuente: extraído de <http://www.opticosweb.com.ar/baston-virtual-para-no-videntes.html>

4.1.6. Gafas para invidentes

Con este dispositivo una persona invidente podrá abrir puertas, subir escalas, detectar obstáculos, entre otros, gracias al sistema de sensores que posee (ver Figura 6). Este dispositivo puede indicar a una persona a qué distancia se encuentra de los objetos, evitar golpes, pedir ayuda y ser ubicado en caso de pérdida de la persona. Este novedoso invento solo necesita estar conectado a un teléfono móvil Android y cuenta con un botón de emergencia para comunicarse con un equipo rescate o asistencia en caso de necesidad. (Diario Independiente Larazon.es, 2013)



Figura 6. Gafas para ciegos

Fuente: extraído de <http://www.larazon.es/las-gafas-para-ciegos-que-acabaran-con-el-baston-y-el-perro-guia-XF4412653#.Tt1xFBaBlkbsy7>

4.1.7. Chip que ayuda a los ciegos a ver formas y tonalidades

La Universidad de Tübingen, en Alemania, creó un chip que se coloca debajo de la parte de la retina llamada mácula, en donde se encuentra la concentración más alta de células sensibles a la luz. La técnica es apta para las enfermedades que afectan las células de varilla y las cónicas que son las que perciben la luz, la convierte en señales eléctricas y las envía al cerebro por medio del nervio óptico. Desafortunadamente no servirá para las enfermedades que afectan el nervio óptico. Este dispositivo que posee 1500 elementos sensibles a la luz, servirá de reemplazo a las células muertas y estimulará las células sanas para enviar la señal al cerebro el cual reconstruirá la imagen. (Simonetti, 2015)

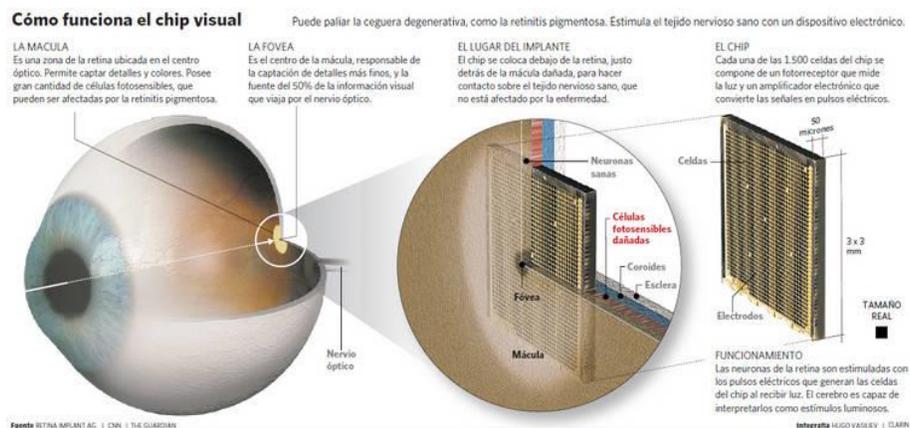


Figura 7. Chip visual

Fuente: extraído de http://www.clarin.com/salud/Crean-ayuda-ciegos-formas-tonalidades_0_372562898.html

4.2. Sensores

Como dice Serna, A. Ros, F. y Rico, J. (2010) en su libro *Guía práctica de sensores* estos son dispositivos que nos ayudan a ejercer una interacción con el ambiente, ya que estos nos facilitan información de algunas variables que pueden cambiar u organizar según la necesidad, para luego ordenar o activar un proceso. Los sensores copian la capacidad de captación de los seres humanos, por ende es muy común encontrarlos en todos los adelantos tecnológicos; debido a esta similar característica con los seres humanos existen sensores que se ven identificados con los sentidos como: vista, oído y tacto; estos reaccionan a la luz, el sonido y el contacto, entre otros.

Los sensores se han convertido en elementos muy comunes debido a la gran aplicabilidad que tienen en los electrodomésticos, automóviles, telefonía, alarmas, procesos industriales entre otros. La adaptación de sensores a los diferentes sistemas electrónicos les proporciona cierta inteligencia, ya que la información al ser procesada permite tomar decisiones rápidas y precisas al objetivo para el cual está diseñado el dispositivo electrónico.

4.2.1. Sensores de proximidad

Uno de los tipos de sensores es el de proximidad. Este tipo de sensores son muy utilizados en el área de la automatización, tienen la capacidad de detectar objetos a una determinada distancia, todo va a depender del tipo de sensor y su capacidad de alcance. (Hyde, Cuspinera, & Regué, 1997)

4.2.2. Sensores ultrasónicos

Localizan objetos sin ningún tipo de contacto físico con estos. Son permisivos para el ajuste de las distancias, ya sean mínimas o máximas; esto hace que sean muy aplicados en el proceso de automatización industrial. En la Figura 8 se puede observar que la distancia mínima es la que se encuentra después de la zona ciega. Hay sensores de salida digital y salida analógica; los digitales se comportan como un contacto es decir para apertura o cierre y los analógicos son capaces de medir con exactitud la distancia a la que se encuentra el objeto. (Martín & García, 2009)

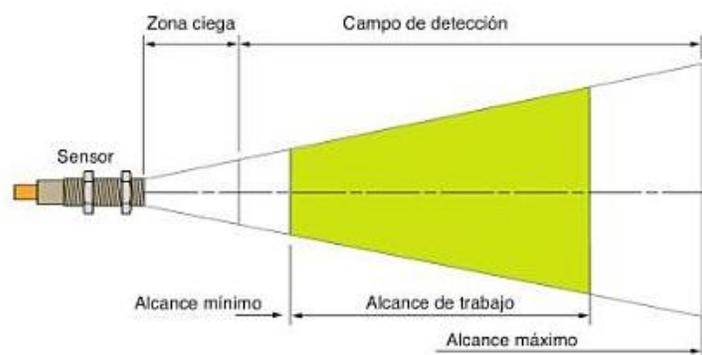


Figura 8. Sensor sonar

Fuente: extraído de libro Automatismos industriales (Martín & García, 2009)

En la Figura 9 observamos un sensor ultrasonido HC-SR04. Tiene una característica muy especial, no solo detecta los objetos que están en frente, sino que también mide la distancia a la cual se encuentra. Además su funcionamiento no lo afectan la luz o la oscuridad. Hay que tener en cuenta que la velocidad del sonido en el aire (a una temperatura de 20 °C) es de 343 m/s y por cada grado centígrado que sube la temperatura, la velocidad del sonido aumenta en 0,6 m/s.



Figura 9. Sensor de distancia HC-SR04

Fuente: extraído de <http://electronilab.co/tienda/sensor-de-distancia-de-ultrasonido-hc-sr04/>

Este sensor posee cuatro pines con las siguientes características:

- Pin1 Vcc: Alimentación 5V (4.5V min – 5.5V max)
- Pin2 Trig: Trigger entrada (input) del sensor
- Pin3 Echo: Echo salida (output) del Sensor
- Pin4: GND

Las principales características técnicas del sensor HC-SR04 son:

- Corriente de reposo: < 2mA
- Corriente de trabajo: 15mA
- Ángulo de medición: 30°
- Detección de 2cm a 400cm
- Frecuencia de trabajo: 40KHz

El funcionamiento comienza cuando se envía un pulso de al menos de 10us por el Pin Trigger. El sensor enviará 8 pulsos de 40KHz (ultrasónicos) y coloca su salida Echo en alto; se debe detectar este suceso e iniciar un conteo de tiempo. La salida Echo se mantendrá en alto hasta recibir el eco reflejado por el obstáculo, a lo cual el sensor pondrá su pin Echo a bajo, y se

da fin al conteo del tiempo. Se da un tiempo de aproximadamente 50ms de espera después de terminar el conteo. El mismo autor define la siguiente fórmula para calcular la distancia:

$$\text{Distancia en cm (centímetros)} = \text{Tiempo medido en us} \times 0.017 \quad (\text{Soria, 2013})$$

4.2.3. Sensor detector de línea

Su funcionamiento se basa en circuito que incorpora un sensor infrarrojo. El circuito consta de dos partes, la primera es un led emisor infrarrojo, el cual emite una luz; y la segunda es un fototransistor al cual llega la señal emitida por el led (ver Figura 10).

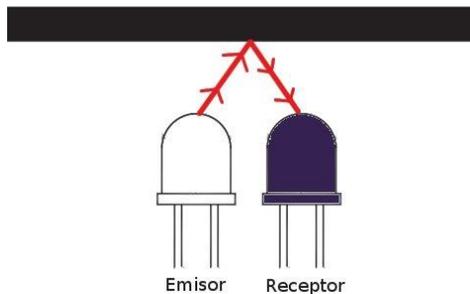


Figura 10. Sensor óptico

Fuente: extraído de. <http://www.ingeniosolido.com/blog/2011/02/sensores-infrarrojos-para-linea-negra/>

El funcionamiento del dispositivo es por reflexión, el emisor envía una señal infrarroja que si encuentra un objeto que la refleje o un color claro o blanco va a llegar por reflexión al fototransistor y este va a conducir, o si por el contrario la luz infrarroja está pegando sobre una base oscura el rayo no se va reflejar y el fototransistor no va a conducir.

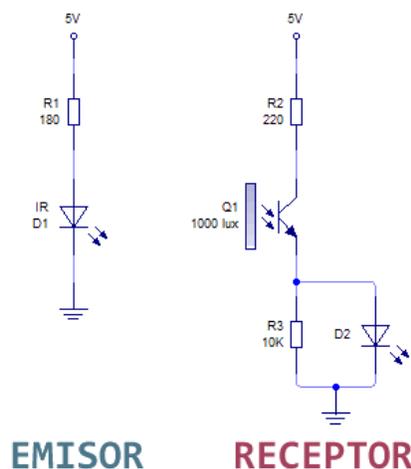
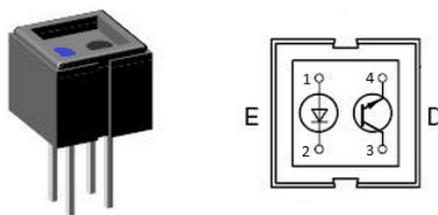


Figura 11. Diagrama eléctrico del sensor infrarrojo

Fuente: extraído de. <http://notitronicampana.blogspot.com/2011/05/emisor-y-receptor-con-un-infrarrojo.html>

En la Figura 12 se observa un sensor CNY70 compuesto por dos elementos, un led infrarrojo y un fototransistor receptor del espectro de luz emitido por el led.



Pin	Función	Descripción
1	ánodo	
2	Cátodo	
3	Colector	
4	Emisor	

Figura 12. Sensor óptico CNY70

Fuente: extraído de. http://www.trastejant.es/tutoriales/electronica/sensordeinfrarrojos_cny70.html

Cuando el led es alimentado comienza a emitir luz en un espectro, fuera de nuestro alcance visual. Esta luz si encuentra un objeto a corta distancia, puede rebotar e incidir en el receptor del sensor. Este sensor es muy preciso y nos puede dar valores analógicos, todo esto dependiendo del color en el que este emitiendo la luz, es decir este sensor nos puede ayudar a diferencia los colores.

4.3. Tecnología de Arduino

Nace de un proyecto de grado de un estudiante colombiano Hernando Barragán quien desarrollo una placa llamada WIRING; luego un grupo de profesores italianos tomaron esta placa, la hicieron más barata y le pusieron el nombre de ARDUINO; fue concebido en el año 2005. Es una plataforma de hardware abierto, su lenguaje de programación es Processing que es muy similar al lenguaje C++; la tarjeta posee entradas y salidas analógicas y digitales. Este dispositivo interconecta el mundo físico con el virtual o el mundo analógico con el digital. (proyecto arduino).

4.3.1. Arduino UNO

Es una placa electrónica que posee un microcontrolador ATmega328. La placa Arduino UNO tiene 14 entradas o salidas digitales, de los cuales 6 de ellos pueden ser utilizados como PWM; 6 entradas analógicas y un oscilador de 16MHz. Es decir esta placa posee los elementos necesarios para la programación del microcontrolador y para utilizar los resultados de la misma



Figura 13: tarjeta de Arduino UNO.

Fuente: extraído de <http://forum.hobbycomponents.com/viewtopic.php?f=64&t=1741>

Las principales características de la placa Arduino UNO son:

- Voltaje de funcionamiento: 5V
- Voltaje de entrada: 7V - 12V
- Corriente DC de 3.3V: 50 mA
- Memoria Flash: 32 KB (ATmega328)
- SRAM: 2 KB (ATmega328)
- EEPROM: 1 KB (ATmega328)

4.4. Microcontrolador atmega328p-pu

Un microcontrolador es un circuito programable capaz de ejecutar las ordenas grabadas en su memoria, como dice Rossano, V. en su libro *Electrónica y Microcontroladores PIC*. Los Microcontroladores cada vez son más utilizados en muchas áreas de la ciencia, gracias a sus grandes ventajas, como la capacidad de almacenamiento de datos, su pequeña estructura y es mucho más fácil y económico crear un circuito con un microcontrolador que utilizar compuertas, registros, contadores, entre otros. Es por esto que el aprendizaje en el manejo de los microcontroladores es de gran importancia para los profesionales en electrónica.

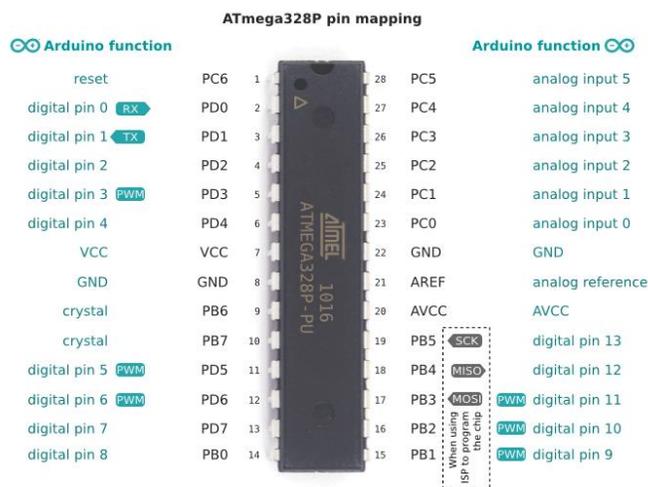


Figura 14: microcontrolador ATMEGA328P-PU

Fuente: extraído de http://www.chicoree.fr/w/Arduino_sur_ATmega328P

Un microcontrolador es muy similar a un microcomputador, posee una unidad central de procesamiento (CPU), entradas / salidas y una memoria; todo esto en un solo componente electrónico llamado circuito integrado.

Los microcontroladores disponen de un oscilador que genera unos pulsos que organizan las operaciones que se ejecutan internamente; en la gran mayoría de estos se utiliza un cristal de cuarzo, estos generan gran estabilidad en la frecuencia. La velocidad con que se ejecutan las operaciones y la respuesta de este va relacionada dependiendo de la frecuencia del oscilador. Al igual que un microcomputador la CPU es el cerebro de los microcontroladores, esta posee una programación, recibe las operaciones a realizar las analiza para después ser ejecutadas y convertirse, así, en una parte importante de un proceso de automatización. (valdes perez & Pallas Areny, 2007)

En secuencia con lo anterior uno de los elementos más importantes en los microcontroladores es la memoria de programa esta puede ser de tres tipos: memoria ROM, memoria PROM y memoria REEPROM. Además existen otros tipos de microcontroladores que vienen sin memoria, esta se le puede adatar externamente.

Memoria ROM es la memoria programada desde su fabricación, los programas grabados en microcontrolador van a ser utilizados para cumplir una función específica.

Memoria PROM viene sin programación, este se programa de acuerdo a la utilidad que se le vaya a dar y luego de ser programada no se puede modificar.

Memoria REEPROM tiene gran ventaja que después de haber efectuado la programación, se puede modificar a través de unos rayos ultravioleta.

4.5. Buzzer

Es un elemento capaz de transformar la electricidad en sonido. La parte central de un buzzer piezoeléctrico es un disco, consiste en una placa cerámica con una capa metálica; si el disco es controlado por un circuito oscilador externo se dice que es un transductor piezoeléctrico, o si por el contrario el circuito oscilado está incluido en él, se le llama zumbador piezoeléctrico.

Estos elementos son de gran utilidad en el diseño de alarmas, o controles acústicos de estrecho rango de frecuencia como los aparatos domésticos o de medicina. (Opiron Electronics, 2013)



Figura 15: Forma física de un buzzer

Fuente: extraído de <http://www.hobbyelectronica.nl/product/buzzer-5v>

4.6. Relé

Es un elemento electromecánico que permite la conmutación de un circuito de media o alta potencia por medio de un circuito electrónico de baja potencia. Una de sus ventajas y por lo cual se usa bastante en electrónica es que la línea eléctrica de conmutación está completamente aislada de la parte electrónica que controla el relé. Es activado o desactivado magnéticamente según sea la conexión.

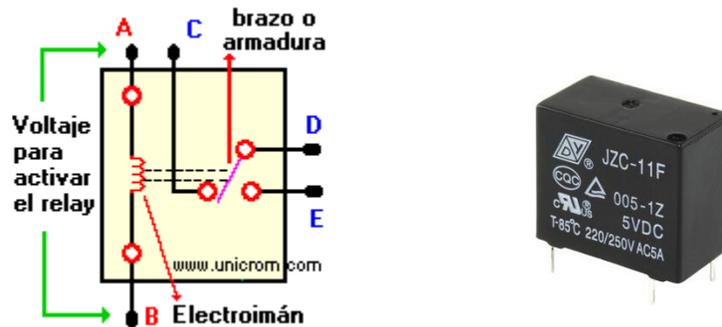


Figura 16: Presentación esquemática y física de un relé.

Fuente: extraído de http://unicrom.com/Tut_relay.asp

Si el electroimán está energizado, atrae el brazo y conecta los puntos C y D; si el electroimán es desenergizado, conecta los puntos D y E (ver Figura 16). De esta manera podemos activar un circuito y la misma vez desactivar otro.

Unas de las características del relé son:

- la cantidad y posición de los contactos
- la potencia de conmutación
- el voltaje de trabajo de la bobina
- la corriente de la bobina (o resistencia)

Estas características determinan generalmente el tamaño del relé. Cuanto más sea cantidad de contactos y la potencia que estos pueden conmutar, mayor será el tamaño de relé. (Inventable.en)

4.7.Motores DC

Un motor es un elemento capaz de transformar la corriente eléctrica en energía mecánica, útil al hombre; en este caso específico es la corriente continua. Los motores de corriente continua son muy similares a un dinamo o generador solo que la función de estos es contraria, el generador lo que hace es tomar la energía mecánica y transformarla en energía eléctrica. (Roldan, 2014)

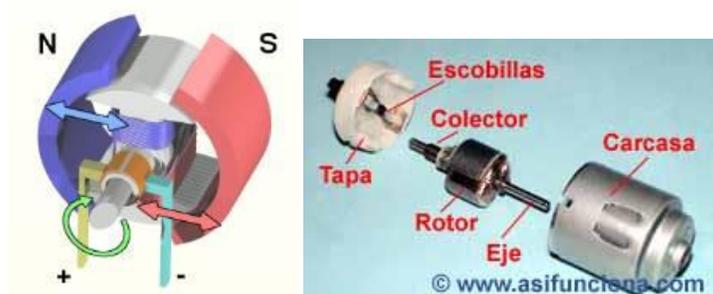


Figura 17: Motor DC y sus partes

Fuente: extraído de http://www.asifunciona.com/electrotecnia/af_motor_cd/af_motor_cd_5.htm

Un tipo de motores DC son los motores IP que utilizan el campo magnético permanente de imanes y campos magnéticos producidos por la corriente de excitación que fluye a través del rotor. Su funcionamiento está basado en la repulsión que ejercen los campos magnéticos que se encuentran en los imanes y en el rotor. Cuando la corriente eléctrica pasa a través del electroimán giratorio o rotor se genera un campo electromagnético que interactúa con el campo magnético del imán permanente; si su polo coincide se genera una repulsión o rechazo que sirve de torque para que el rotor sobrepase la inercia y empiece a girar. El sentido del giro va a depender de la conexión del motor con la batería. (García Álvarez, 2012)

4.7.1. Motor vibrador

El motor lleva acoplado a su eje una rueda excéntrica, que tiene una masa considerable respecto a la del motor propiamente dicho. Al ser alimentado a su tensión de funcionamiento, el motor gira a alta velocidad y debido a la masa excéntrica del volante, genera una oscilación de esta misma frecuencia. La vibración del motor provoca un aviso claramente perceptible por la persona que lleva cualquier tipo de aparato fabricado con este motor-vibrador (ya sea en el bolsillo, en el cinturón, en la mano, etc.).

Su poco peso, su reducido tamaño y el bajo consumo de energía que necesita para funcionar, lo hacen idóneo para toda clase de aplicaciones portátiles, especialmente como avisador silencioso personal, avisos de seguridad en ambientes altamente ruidosos, avisos y señales para

personas con dificultades, tanto auditivas como de visión y en general, todas aquellas aplicaciones que requieran privacidad o allí donde se exige silencio. (Colorado, 2009)



Figura 18: Presentación física de un motor vibrador

Fuente: extraído de <http://www.carrod.mx/>

4.8. Fuente de alimentación

Como dice Pallas, R. (2006) en su libro *Instrumentos Electrónicos Básicos* para que los dispositivos o circuito electrónicos funcionen correctamente, se le debe suministrar una energía eléctrica de corriente y tensión.

4.8.1. La batería

La batería es un elemento capaz de almacenar energía química y luego liberarla como energía eléctrica y así hacerla útil a las necesidades del ser humano. Es la fuente de corriente continua (DC) más común; consiste en la combinación de varias celdas siendo estas la fuente fundamental de energía eléctrica desarrollada por medio de una reacción química. Hay dos tipos de baterías. Tipo primario: es la batería no recargable, esta utiliza un ánodo de zinc pulverizado (+), un electrolito de hidróxido de potasio y un cátodo de dióxido de manganeso y carbón (-). Su desventaja es que es desechable. Tipo secundario: son baterías recargables, la reacción química de la celda secundaria puede ser invertida para restaurar su capacidad. Cuando se aplica una carga a las terminales de la batería se da una transferencia de electrones desde el electrodo de plomo al de peróxido de plomo a través de la carga. Esta transferencia de electrodos se da hasta que la batería este descargada por completo. (Boylestad, 2004)

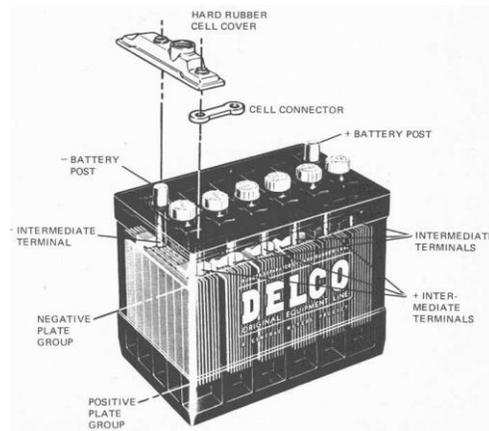


Figura 19: Presentación física de la batería tipo secundario

Fuente: extraído de <http://automecanico.com/auto2001/bat4.jpg>

5. Metodología

5.1. Tipo de estudio

Es una investigación experimental que nos permite la aplicabilidad de los conocimientos adquiridos en el transcurso de la formación en tecnología electrónica y profundizar sobre estos, de una manera clara y organizada para que sea esta el apoyo fundamental para la realización de presente trabajo de grado.

La investigación fue cualitativa ya que se buscó la mejora de un dispositivo de apoyo para las personas invidentes, basado en los diferentes prototipos existentes y toda su información, teniendo en cuenta sus ventajas, desventajas y características. También se analizó la información de los componentes electrónicos utilizados, los tipos y características, apoyados en las teorías, proyectos e investigaciones existentes, con el fin de obtener un nuevo prototipo.

5.2.Método

En base de observación y deducción de las necesidades de las personas invidentes proviene la propuesta de este proyecto de desarrollar un dispositivo que mejora las condiciones de desplazamiento de esta comunidad.

En la etapa inicial es fundamental la búsqueda de información acerca de proyectos realizados en el campo de avances tecnológicos para las personas invidentes y los dispositivos utilizados, sus funciones, ventajas y desventajas y tomar esta información como referente para la realización de este proyecto.

La etapa posterior es orientada hacia el análisis de fichas técnicas de componentes; realización de un diseño del circuito, prueba de funcionamiento de los sensores infrarrojo y de proximidad, para proceder a la programación del microcontrolador.

Después del diseño de sistema electrónico hacer el montaje en protoboard, las pruebas experimentales, análisis de los problemas presentados y respectivas correcciones.

Una vez terminadas las pruebas experimentadas elaborar el plano esquemático, diseño y fabricación de la tarjeta electrónica, montaje de componentes y realización de pruebas finales.

Como parte final en el desarrollo de este prototipo se hace la prueba de confiabilidad del sistema en el ambiente de aplicación real y elaboración de informe final.

5.3.Instrumentos de recolección de información

5.3.1. Fuentes primarias

Se realizo la recopilación de información de libros, sitios web, tesis de grado de estudiantes de varias universidades, revistas universitarias y periódicos nacionales o internacionales on-line.

5.3.2. Fuentes secundarias.

Otra parte de la información recopilada es de profesionales en electrónica, ingenieros, docentes, compañeros de estudio y de trabajo

6. Resultados

Inicialmente se realizó un análisis de los componentes necesarios para el circuito que detecta el cambio de color, tomando como referente un proyecto con un carro seguidor de línea. Además teniendo en cuenta que son varios los circuitos y variables a controlar se llega a la conclusión de utilizar la tecnología Arduino.

El CNY70 es un sensor infrarrojo que está compuesto internamente por dos elementos que son los que detectan el cambio de color. En primer lugar encontramos un diodo emisor, que es alimentado por su ánodo con 5V y su otro extremo está conectado a tierra por medio de una resistencia. Su función es enviar constantemente un haz de luz hacia la superficie donde este proyectando. En segundo lugar encontramos un fototransistor, alimentado en su colector por 5V y en el emisor una resistencia a tierra; a su vez será este el nodo donde tomaremos la señal que vamos a llevar hacia la placa de Arduino.

El circuito a utilizar es el de la Figura 20; se puede observar el sensor infrarrojo CNY70 con las conexiones necesarias para que cuando sea proyectada la luz IR del transmisor sobre diferentes colores (blanco, negro o rojo) arroje diferentes niveles de voltaje en salida, dependiendo de los colores de la superficie.

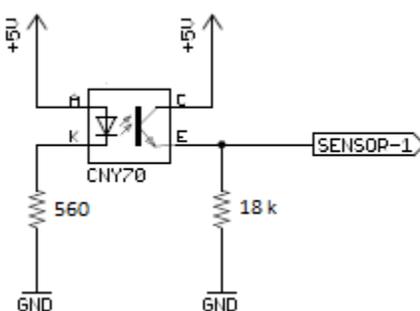


Figura 20: esquema de sensor infrarrojo
Fuente: Imagen propia del autor

Cuando el led emisor proyecta su luz sobre una superficie clara o blanca esta se refleja en la base del foto transistor, haciendo que este conduzca dando una tensión de voltaje en el emisor. A su vez cuando el led emisor proyecta su luz sobre una superficie oscura o de color negro esta no se va a reflejar y no vamos a tener tensión de voltaje en el emisor del fototransistor. Todos los colores pueden reflejar cierta cantidad de luz, y dependiendo de cuanta sea esta, el emisor del foto transistor percibirá cambios de tensión, estos cambios son los que se utilizan para la identificación de colores a través de este sensor.

Las salidas de los sensores se conectan a las entradas análogas de la tarjeta Arduino y con base a la diferencia de potencial, que nos dan los colores blanco, negro y rojo, se realiza su programación. La tarjeta Arduino debe programarse inicialmente con tres entradas análogas que son a las que va llegar la señal de los sensores infrarrojo, se hace la programación para poder ver en el monitor serial de Arduino los valores dados por cada uno de los sensores en los diferentes colores (ver Figura 21).



Figura 21: conexión de sensor infrarrojo

Fuente: Imagen propia del autor

Para dar cumplimiento a otro de nuestros objetivos del proyecto, como lo es la detección de los obstáculos es necesario tener un sensor de proximidad, para este caso se utiliza el HC-SR04, un sensor ultrasónico que necesita un complemento como la tecnología Arduino, ya que será necesario que uno de sus pines llegue un pulso con determinado tiempo.

Este sensor posee cuatro pines: una alimentación de 5V, tierra, trig y echo. Para realizar la medición debemos enviar un pulso de 10us por el pin trig, luego el sensor con su oscilador

piezoeléctrico, empieza a generar pulsos a 40khz, para poder realizar la localización de eco. Cuando se encuentra un obstáculo, por el pin de echo se devuelve un pulso, que corresponde al doble de la distancia, tanto de ida como de regreso. Es decir damos el pulso por el trig y esperamos la respuesta por el echo (ver Figura 22).

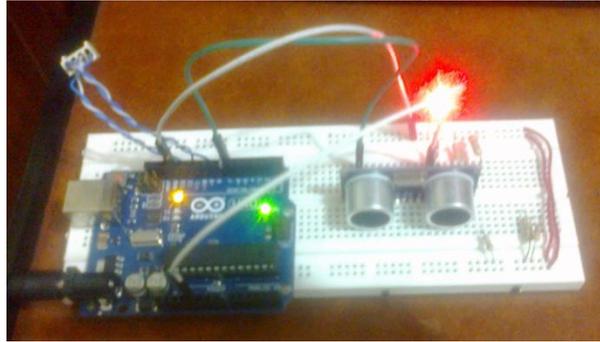


Figura 22: conexión Arduino y sensor de proximidad
Fuente: Imagen propia del autor

Para calcular la distancia utilizamos la siguiente formula.

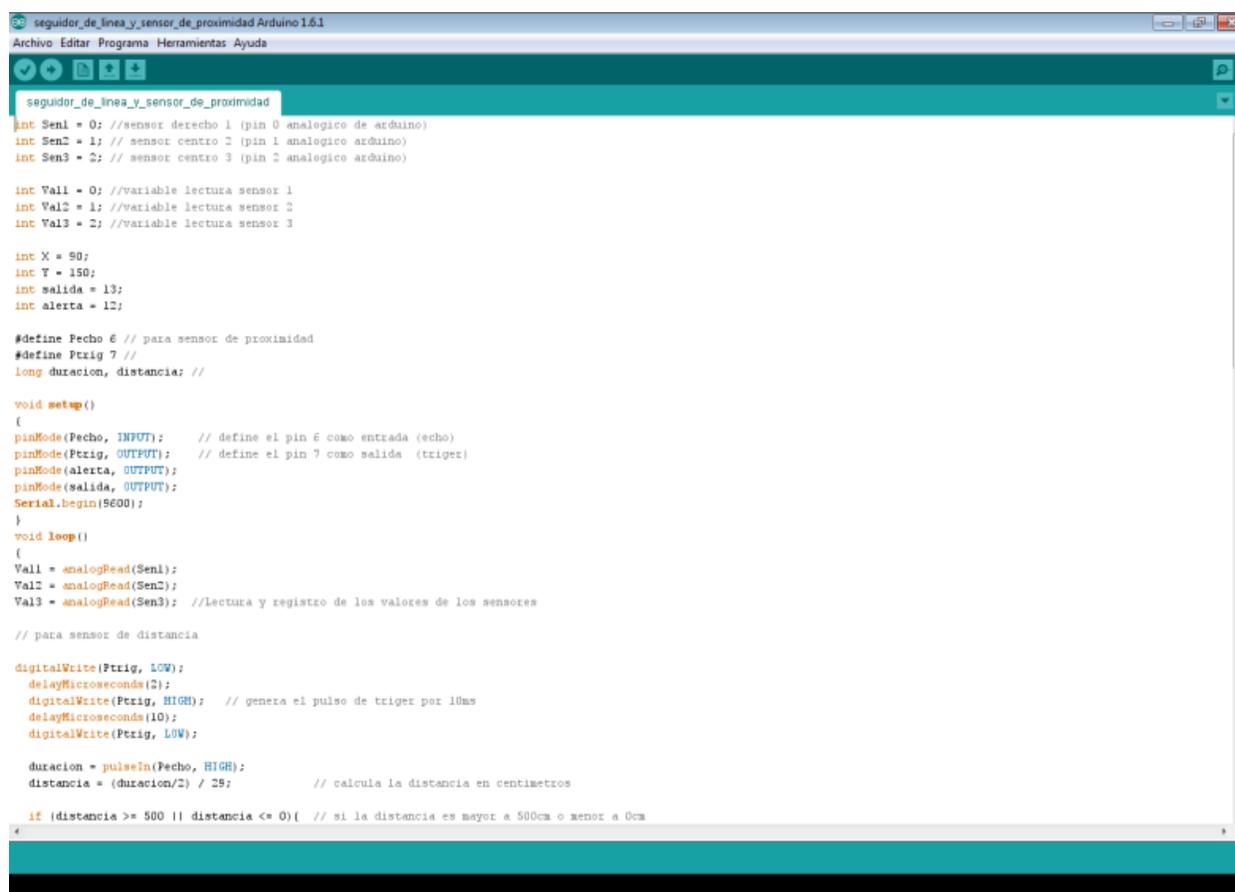
$$distancia = \frac{\frac{duracion}{2}}{29}$$

Se divide la duración por 2 ya que ese tiempo es el que tarda el pulso de ida y regreso y sobre 29 porque se trabaja con la formula de la velocidad del sonido en cm.

Velocidad de sonido = 1/29 cm/us

Para la programación se debe definir el pin trig como salida y el pin echo como entrada de la tarjeta Arduino; se utiliza el comando *serial.begin* para observar en el monitor serial la distancia a la que se encuentra el objeto u obstáculo; esto para cuando se quiere saber la distancia, o también se programa de manera que cuando un obstáculo se encuentre a determinada distancia, se active una alarma o se proceda a realizar una acción. Esta función esta implementada en el presente prototipo.

Después de tener identificados los valores arrojados por cada sensor infrarrojo cuando se encuentran sobre los tres colores a utilizar, y el sensor de proximidad funcionando adecuadamente, se procede hacer la programación. Tener en cuenta las variables *Int* que son de tipo entero y las *Long* para cuando se trabaja con valores decimales. Identificamos los pines de entrada análogos y digitales; en este caso análogos son los pines donde están conectados los sensores infrarrojos y digital es el pin echo del sensor de proximidad. Definimos los pines de salida: en este caso el pin Trig del sensor de proximidad, un pin para el relé y otro para el buzzer (ver Figura 23).



```

seguidor_de_linea_y_sensor_de_proximidad Arduino 1.6.1
Archivo Editar Programa Herramientas Ayuda

seguidor_de_linea_y_sensor_de_proximidad

int Sen1 = 0; //sensor derecho 1 (pin 0 analogico de arduino)
int Sen2 = 1; // sensor centro 2 (pin 1 analogico arduino)
int Sen3 = 2; // sensor centro 3 (pin 2 analogico arduino)

int Val1 = 0; //variable lectura sensor 1
int Val2 = 1; //variable lectura sensor 2
int Val3 = 2; //variable lectura sensor 3

int X = 90;
int Y = 150;
int salida = 13;
int alerta = 12;

#define Pecho 6 // para sensor de proximidad
#define Ptrig 7 //
long duracion, distancia; //

void setup()
{
  pinMode(Pecho, INPUT); // define el pin 6 como entrada (echo)
  pinMode(Ptrig, OUTPUT); // define el pin 7 como salida (trigger)
  pinMode(alerta, OUTPUT);
  pinMode(salida, OUTPUT);
  Serial.begin(9600);
}

void loop()
{
  Val1 = analogRead(Sen1);
  Val2 = analogRead(Sen2);
  Val3 = analogRead(Sen3); //Lectura y registro de los valores de los sensores

  // para sensor de distancia
  digitalWrite(Ptrig, LOW);
  delayMicroseconds(2);
  digitalWrite(Ptrig, HIGH); // genera el pulso de trigger por 10ms
  delayMicroseconds(10);
  digitalWrite(Ptrig, LOW);

  duracion = pulseIn(Pecho, HIGH);
  distancia = (duracion/2) / 29; // calcula la distancia en centimetros

  if (distancia >= 500 || distancia <= 0){ // si la distancia es mayor a 500cm o menor a 0cm

```

Figura 23: Programación en Arduino

Fuente: Imagen propia del autor

Teniendo toda la programación en la placa de Arduino y las conexiones de los sensores y actuadores, se procede hacer las pruebas verificando que todo esté funcionando adecuadamente y que responda como se espera.

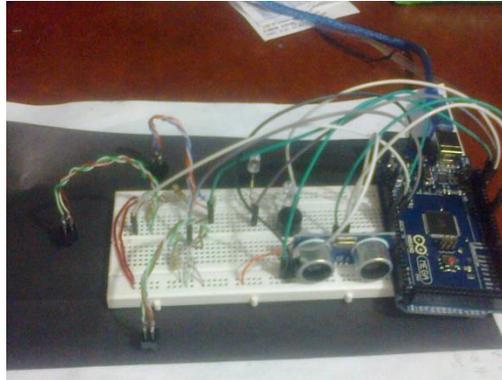


Figura 24: Pruebas del proyecto con tarjeta Arduino

Fuente: Imagen propia del autor

Luego de realizar las pruebas y obtener el sistema electrónico funcionando dentro de los parámetros estipulados de este prototipo, se considera necesaria la extracción del programa de la tarjeta Arduino a un microcontrolador (ver Figura 24). Así se obtiene un dispositivo independiente; no se tendría que adquirir una tarjeta Arduino para soportar el funcionamiento de lo que se quiera realizar. En esta manera se rebaja el costo del prototipo.

Para el siguiente proceso se utilizara la tarjeta Arduino UNO (ver Figura 13), ya que esta tiene incorporado el microcontrolador ATMEGA328P-PU, que una vez tenga cargado el programa simplemente es extraerlo de la tarjeta e implementarlo en el sistema electrónico diseñado y realizar determinadas conexiones. Pero antes se debe tener otro microcontrolador para dejar en la tarjeta y que esta siga funcionando normalmente. Todo microcontrolador para que funcione en la tarjeta Arduino tiene que tener cargado previamente el *bootloader* o secuencia de inicio; a continuación se encuentra el procedimiento para hacerlo.

En primer lugar debemos cargar al Arduino el programa para que este se convierta en un ISP. El programa se encuentra en el Anexo A.

Luego de tener el programa cargado procedemos a realizar las siguientes conexiones, entre la tarjeta Arduino y el nuevo microcontrolador (ver Figura 25).

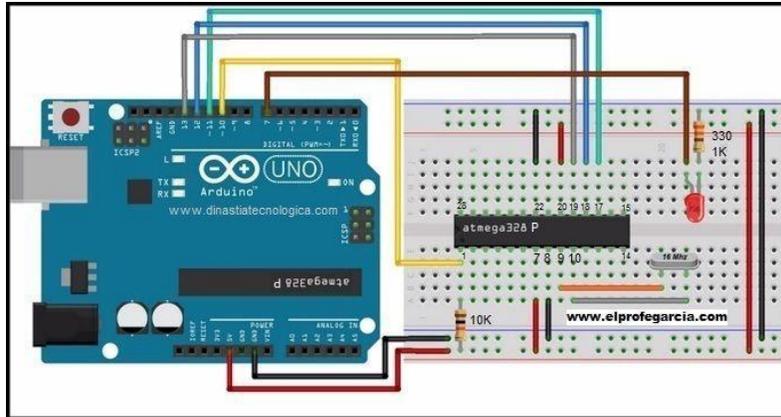


Figura 25: Conexión para cargar Bootloader

Fuente: extraído de <https://www.dropbox.com/s/0q7ej0ev1m3q388/Plano%20arduino%20isp%20bootloader.jpg>

Teniendo lista la conexión conectamos la tarjeta al PC, vamos al programa de Arduino, entramos a Herramientas – Programador – Arduino as ISP (ver Figura 26).

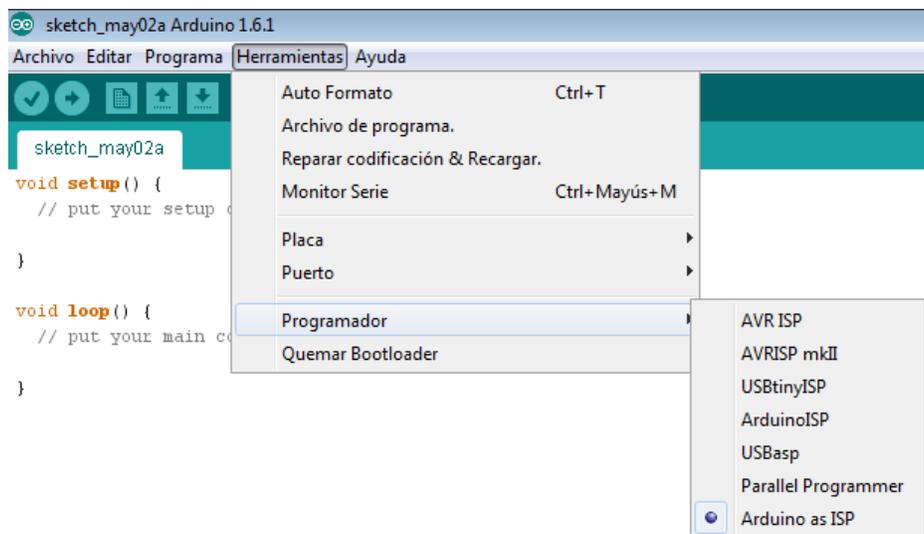


Figura 26: Arduino como ISP

Fuente: Imagen propia del autor

Posteriormente se regresa a la opción de Herramientas – Quemar Bootloader o en algunos casos aparecerá grabar secuencia de inicio (ver Figura 27).

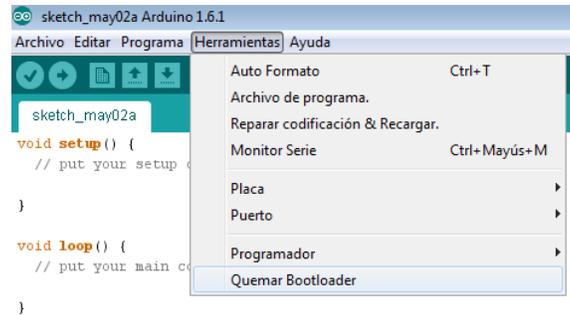


Figura 27: Quemar Bootloader

Fuente: Imagen propia del autor

Una vez damos clic en Quemar Bootloader, observaremos que empieza a parpadear los indicadores de transmisión y recepción (TX y RX) de la tarjeta y se encenderá el led que está conectado al pin 7, cuando este se apague quiere decir que el microcontrolador ya tiene el programa de inicio grabado y que está listo para ser utilizado en la tarjeta (ver Figura 28).

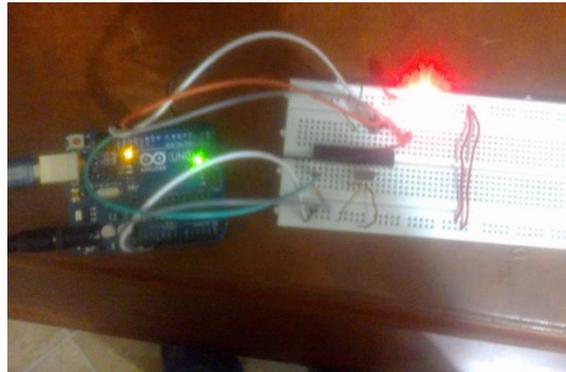


Figura 28: Practica de grabar secuencia de inicio al microcontrolador

Fuente: Imagen propia del autor

Ahora se tienen dos microcontroladores que sirven para grabar el programa que se ha creado para el proyecto, cualquiera de los anteriores va a funcionar correctamente. Una vez cargado el programa se retira el microcontrolador de la tarjeta de Arduino y se pone en una protoboard haciendo inicialmente las siguientes conexiones para que este funcione adecuadamente.

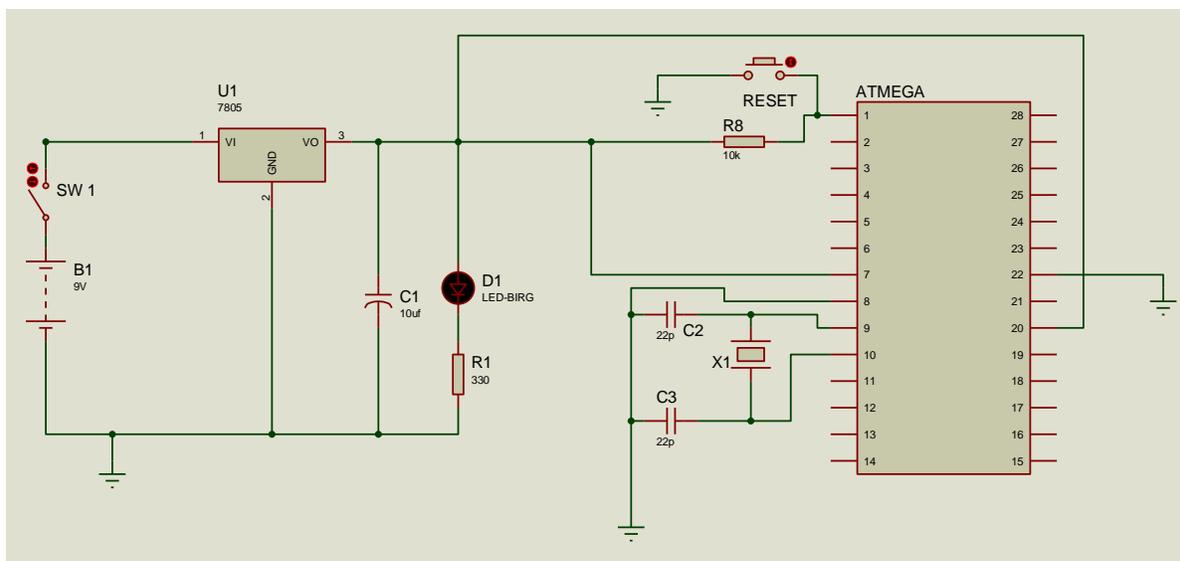


Figura 29: Circuito para el funcionamiento independiente del microcontrolador
 Fuente: Elaboración propia proteus

Como se puede observar en la Figura 29 es necesaria la inclusión de una fuente de alimentación y un circuito regulador de voltaje de 9V a 5V, que es el voltaje a utilizar para la alimentación del microcontrolador por los pines 7 y 20. Los pines 22 y 8 estarán conectados a tierra; entre los pines 9 y 10 se colocara el oscilador o cristal de 16MHz y el pin numero 1 estará conectado a 5V por medio de una resistencia de 10k. Ahora se debe hacer el reemplazo de los pines de la tarjeta con los pines del microcontrolador basados en la Figura30.

	(PCINT14/RESET) PC6	1	28	PC5 (ADC5/SCL/PCINT13)	AIN5
RX - D0	(PCINT16/RXD) PD0	2	27	PC4 (ADC4/SDA/PCINT12)	AIN4
TX - D1	(PCINT17/TXD) PD1	3	26	PC3 (ADC3/PCINT11)	AIN3
D2	(PCINT18/INT0) PD2	4	25	PC2 (ADC2/PCINT10)	AIN3
PWM3	(PCINT19/OC2B/INT1) PD3	5	24	PC1 (ADC1/PCINT9)	AIN1
D4	(PCINT20/XCK/T0) PD4	6	23	PC0 (ADC0/PCINT8)	AIN0
	VCC	7	22	GND	
	GND	8	21	AREF	
	(PCINT6/XTAL1/TOSC1) PB6	9	20	AVCC	
	(PCINT7/XTAL2/TOSC2) PB7	10	19	PB5 (SCK/PCINT5)	D13
PWM5	(PCINT21/OC0B/T1) PD5	11	18	PB4 (MISO/PCINT4)	D12
PWM6	(PCINT22/OC0A/AIN0) PD6	12	17	PB3 (MOSI/OC2A/PCINT3)	PWM11
D7	(PCINT23/AIN1) PD7	13	16	PB2 (SS/OC1B/PCINT2)	PWM10
D8	(PCINT0/CLKO/ICP1) PB0	14	15	PB1 (OC1A/PCINT1)	PWM9

Figura 30: Configuración de los pines del microcontrolador con respecto a la tarjeta Arduino.
 Fuente: extraído de <https://www.dropbox.com/s/ffhvhehx28d8tt6/Atmel%20328P.jpg>

Se procede hacer el montaje que se tenía en la tarjeta Arduino, en la protoboard, con el microcontrolador que ya tiene cargado el programa.



Figura 31: Montaje de circuito y pruebas.

Fuente: Imagen propia del autor

Se hace la adaptación del buzzer y del relé. Se debe analizar las fichas técnicas de estos dos elementos a utilizar principalmente que la corriente que consumen no sea mayor a la que puede dar el microcontrolador, y si es así hay que hacer la implementación de transistores.

Se procede a la realización de un plano esquemático (Ver Anexo B) y el circuito PCB para la consecución del circuito impreso.

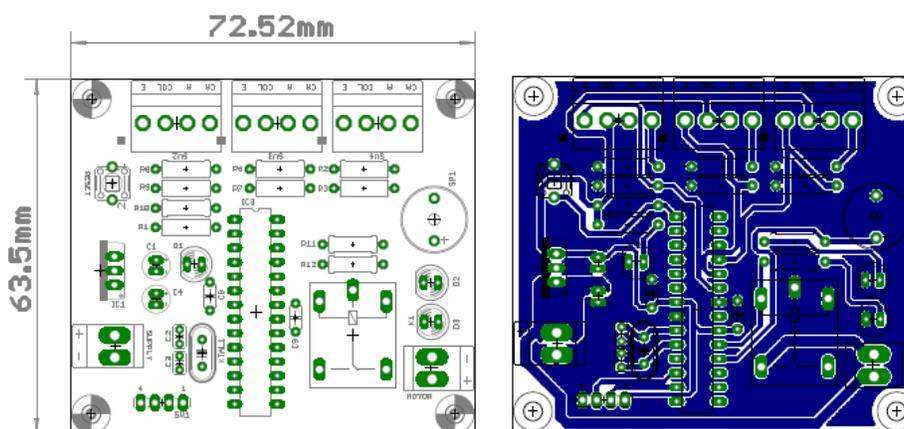


Figura 32: Diseño para circuito impreso

Fuente: Imagen propia del autor

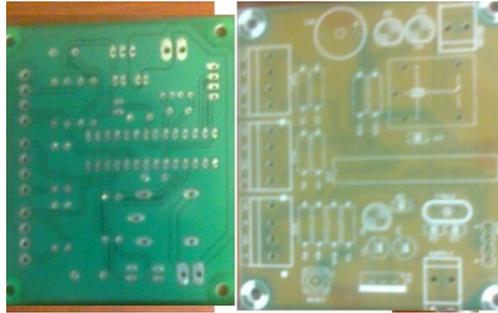


Figura 33: Circuito impreso
Fuente: Imagen propia del autor

Ya con el circuito impreso procedemos al montaje de los componentes (ver Figura 33 y 34).

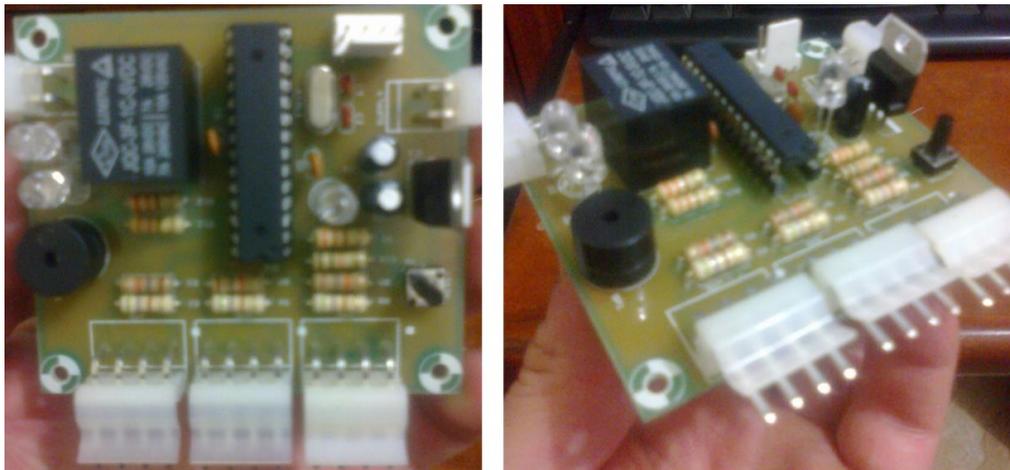


Figura 34: Circuito impreso con los componentes
Fuente: Imagen propia del autor

Teniendo nuestra palca con todos los componentes instalados, se procede a la conexión de la fuente y a realizar las pruebas, antes de ser instalado en el bastón (ver Figura 35).

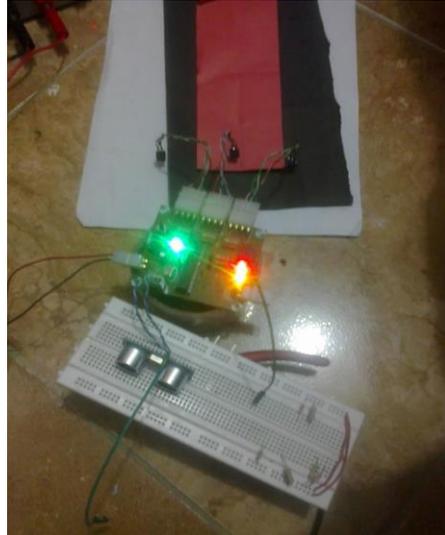


Figura 35: Pruebas con circuito antes de montar en el bastón

Fuente: Imagen propia del autor

Se procede al diseño de una caja en acrílico que nos sirva como soporte para nuestro circuito impreso y luego se adapta el bastón. Se debe tener en cuenta la ubicación de la batería, que los sensores queden a la distancia requerida del piso, y ubicar el motor vibrador en la parte superior del bastón (ver Figura 36).



Figura 36: Montaje del circuito en el diseño de bastón

Fuente: Imagen propia del autor

Se toma la decisión de conseguir una base metálica para que el bastón sea más estable, ya que como muestras la Figura 36 se tenían problemas de inestabilidad.

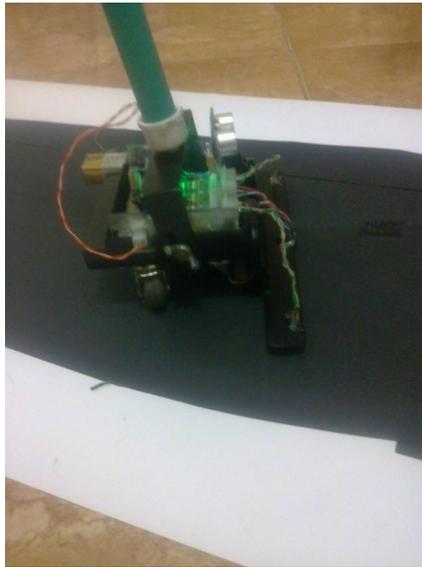


Figura 37: Bastón listo para realizar pruebas.

Fuente: Imagen propia del autor

En la Figura 37 se observa el bastón en fase de pruebas de funcionamiento (ver Anexo C).

7. Conclusiones

Se logra el desarrollo de un prototipo de bastón para persona invidentes, que detecta obstáculos que se encuentran a 30 cm de distancia; también envía dos tipos de alarma por medio de vibraciones en dos ocasiones cuando la persona está abandonando la línea a la derecha o a la izquierda. Además cuenta con una alerta adicional para cuando se está a punto de llegar a alguna irregularidad del terreno o desvío de camino.

Se realizaron pruebas a través de un camino demarcado por las líneas y las alertas funcionan correctamente, cumpliendo con los objetivos planteados (ver Anexo C).

En el transcurso del proyecto, se investigó y se aprendió acerca de la tecnología Arduino, su función y programación. Un punto importante fue el extraer la programación de la tarjeta y llevarla a un microcontrolador, para obtener un circuito independiente, logrando que este funcionara correctamente, generando un ahorro ya que no se tendrá que adquirir una nueva tarjeta.

Hubo inconvenientes en el momento del montaje de los elementos en el circuito impreso, el relé no era el adecuado pero se solucionó realizando nuevas perforaciones y adecuándolo correctamente. Además después de realizar esta mejora el circuito aun no funcionaba; se pasó a realizar pruebas haciendo varias mediciones en diferentes partes del circuito, hasta que se encontró que el cristal de 16 MHz se había dañado. Luego de solucionar este problema el circuito empezó a funcionar correctamente.

En general este prototipo se deja en condiciones de funcionamiento previsto inicialmente, los sensores de proximidad y de ultrasonido responden adecuadamente; los actuadores de buzzer y de vibración mecánica también generan respuesta esperada. La prueba final se hizo en la IU Pascual Bravo simulando un camino con línea de tres colores, de forma irregular y con obstáculos (ver Anexo C).

8. Recomendaciones

Durante el desarrollo del proyecto se encontró que aparte de las funciones que este cumple se le pueden implementar otras que hagan de este prototipo una herramienta más sofisticada para las personas que padezca una discapacidad visual.

Los sensores infrarrojo son sensibles a los colores a donde estos estén reflejando, se podría utilizar esta características para activar o modificar las alarmas de manera que indiquen muchas más cosas. Es decir implementar nuevos colores que avisen acerca de más fenómenos que se encuentran en los recorridos.

Sería de gran ayuda implementar un GPS que indique en que parte específica se encuentra el dispositivo o bastón, así su familia o parientes sabrán la ubicación de la persona que lo porta, teniendo en cuenta que se está hablando de un individuo que sufre una discapacidad.

Lograr una comunicación entre el microcontrolador y un teléfono celular ayudara a la persona que porta el dispositivo cuando este en problemas o se sienta perdida.

Se recomienda la utilización de otro tipo de sensores infrarrojo ya que el CNY70 es un sensor de corto alcance su distancia de operación es de 5mm, una irregularidad podría hacer que este no funcione correctamente. Es decir un sensor que cumpla con las mismas funciones pero que su distancia de operación sea mayor.

Todas estas mejoras se dejan como propuestas para futuros proyectos de investigación.

9. Referencias bibliográficas

- 5Hertz electronica*. (04 de 2014). Obtenido de <http://5hertz.com/tutoriales/?p=659>
- Boylestad, R. L. (2004). *Introducción al análisis de circuitos*. España: Pearson Educación.
- Colorado, J. J. (2009). *Sistema de aviso para personas con discapacidad*.
- Diario Independiente Larazon.es*. (20 de 11 de 2013). Recuperado el 27 de 04 de 2014, de <http://www.larazon.es/las-gafas-para-ciegos-que-acabaran-con-el-baston-y-el-perro-guia-XF4412653#.Ttt1xFBaBlkbsy7>
- Fallotico, J. M. (2006). *Baston blanco*.
- funlam, E. (2014). *Azul naranja palabra digital*.
- Garcia Alvarez, J. A. (3 de 2012). *Asi funciona*. Obtenido de http://www.asifunciona.com/electrotecnia/af_motor_cd/af_motor_cd_5.htm
- Garcia, P. (2014). *El profe Garcia*.
- Hernández, M. I. (25 de noviembre de 2012). *catalogo para discapacidad visual*. Recuperado el 13 de marzo de 2014, de http://leecolima.net/2014/diccionario1.dll.php?&mulieres1=a_col_especial_motriz&di=507&sec_ins=
- Hofer, S. (29 de 08 de 2011). *cuentame algo bueno*. Recuperado el 27 de 04 de 2014, de <http://www.cuentamealobueno.com/2011/08/tacit-un-sonar-para-ciegos/>
- Hyde, J., Cuspinera, A., & Regué, J. (1997). *Control electroneumetico y electronico*. España: marcombo.
- Inventable.en*. (s.f.). Obtenido de <http://www.inventable.eu/introduccion-a-los-reles/>
- Martín, J. C., & García, P. (2009). *Automatismos industriales*. España: Editex.
- Montoya García, J. J., Naranjo Contreras, K. T., Tobón Rivera, M., & Visbal Fernández, D. E. (03 de 2009). Obtenido de <http://server-die.alc.upv.es/asignaturas/PAEEES/2008-09/Sensor%20Infrarrojo%20-%20Grupo%20Naranja.pdf>
- Muy Interesante*. (2 de 10 de 2000). Obtenido de <http://www.muyinteresante.es/tecnologia/preguntas-respuestas/icomofunciona-el-baston-para-ciegos>
- Opticos Web*. (2013). Obtenido de <http://www.opticosweb.com.ar/baston-virtual-para-no-videntes.html>
- Rincon Gil, E. (7 de 2 de 2011). *Ingenio Solido*. Obtenido de <http://www.ingeniosolido.com/blog/2011/02/sensores-infrarrojos-para-linea-negra/>
- Roldan, J. (2014). *Motores de corriente continua*. España: Ediciones paraninfo.

Simonetti, S. S. (10 de 11 de 2015). *clarin*. Recuperado el 5 de 5 de 2014, de clarin:

http://www.clarin.com/salud/Crean-ayuda-ciegos-formas-tonalidades_0_372562898.html

Soria, K. (17 de 9 de 2013). *BKAR ELECTRONICA*. Obtenido de

<http://bkargado.blogspot.com/2013/09/todosobrehc-sr04.html>

Tendencias Tecnologicas. (2013). Obtenido de http://www.tendencias21.net/Nuevo-baston-inteligente-ayuda-a-los-ciegos-a-evitar-obstaculos_a20626.html

10. Bibliografía

Hyde, J., Cuspinera, A., & Regué, J. (1997). *Control electroneumetico y electronico*. España: marcombo.

Warren, J. D., Adams, J., & Molle, H. (2011). *Arduino Robotics*. Apress.

Zabala, G. Robótica. Usershop

11. Anexos

Anexo A. Listado del programa del Bootloader

Código fuente del programa para hacer Bootloader del microcontrolador ATM...

```

/ ArduinoISP version 04m3
// Copyright (c) 2008-2011 Randall Bohn
// If you require a license, see
//   http://www.opensource.org/licenses/bsd-license.php
//
// This sketch turns the Arduino into a AVRISP
// using the following arduino pins:
//
// pin name:      not-mega:          mega(1280 and 2560)
// slave reset: 10:                  53
// MOSI:          11:                  51
// MISO:          12:                  50
// SCK:           13:                  52
//
// Put an LED (with resistor) on the following pins:
// 9: Heartbeat - shows the programmer is running
// 8: Error      - Lights up if something goes wrong (use red if that makes sense)
// 7: Programming - In communication with the slave
//
// 23 July 2011 Randall Bohn
// -Address Arduino issue 509 :: Portability of ArduinoISP
// http://code.google.com/p/arduino/issues/detail?id=509
//
// October 2010 by Randall Bohn
// - Write to EEPROM > 256 bytes
// - Better use of LEDs:
// -- Flash LED_PMODE on each flash commit
// -- Flash LED_PMODE while writing EEPROM (both give visual feedback of writing
progress)
// - Light LED_ERR whenever we hit a STK_NOSYNC. Turn it off when back in sync.
// - Use pins_arduino.h (should also work on Arduino Mega)
//
// October 2009 by David A. Mellis
// - Added support for the read signature command

```

```

//
// February 2009 by Randall Bohn
// - Added support for writing to EEPROM (what took so long?)
// Windows users should consider WinAVR's avrdude instead of the
// avrdude included with Arduino software.
//
// January 2008 by Randall Bohn
// - Thanks to Amplificar for helping me with the STK500 protocol
// - The AVRISP/STK500 (mk I) protocol is used in the arduino bootloader
// - The SPI functions herein were developed for the AVR910_ARD programmer
// - More information at http://code.google.com/p/mega-isp

#include "pins_arduino.h"
#define RESET      SS

#define LED_HB     9
#define LED_ERR    8
#define LED_PMODE  7
#define PROG_FLICKER true

#define HWVER 2
#define SWMAJ 1
#define SWMIN 18

// STK Definitions
#define STK_OK      0x10
#define STK_FAILED  0x11
#define STK_UNKNOWN 0x12
#define STK_INSYNC  0x14
#define STK_NOSYNC  0x15
#define CRC_EOP     0x20 //ok it is a space...

void pulse(int pin, int times);

void setup() {
  Serial.begin(19200);
  pinMode(LED_PMODE, OUTPUT);
  pulse(LED_PMODE, 2);
  pinMode(LED_ERR, OUTPUT);
  pulse(LED_ERR, 2);
  pinMode(LED_HB, OUTPUT);
  pulse(LED_HB, 2);
}

```

```

}

int error=0;
int pmode=0;
// address for reading and writing, set by 'U' command
int here;
uint8_t buff[256]; // global block storage

#define beget16(addr) (*addr * 256 + *(addr+1) )
typedef struct param {
    uint8_t devicecode;
    uint8_t revision;
    uint8_t progtype;
    uint8_t parmode;
    uint8_t polling;
    uint8_t selftimed;
    uint8_t lockbytes;
    uint8_t fusebytes;
    int flashpoll;
    int eeprompoll;
    int pagesize;
    int eepromsize;
    int flashsize;
}
parameter;

parameter param;

// this provides a heartbeat on pin 9, so you can tell the software is running.
uint8_t hbval=128;
int8_t hbdelta=8;
void heartbeat() {
    if (hbval > 192) hbdelta = -hbdelta;
    if (hbval < 32) hbdelta = -hbdelta;
    hbval += hbdelta;
    analogWrite(LED_HB, hbval);
    delay(20);
}

void loop(void) {
    // is pmode active?

```

```
if (pmode) digitalWrite(LED_PMODE, HIGH);
else digitalWrite(LED_PMODE, LOW);
// is there an error?
if (error) digitalWrite(LED_ERR, HIGH);
else digitalWrite(LED_ERR, LOW);

// light the heartbeat LED
heartbeat();
if (Serial.available()) {
    avrisp();
}
}

uint8_t getch() {
    while(!Serial.available());
    return Serial.read();
}

void fill(int n) {
    for (int x = 0; x < n; x++) {
        buff[x] = getch();
    }
}

#define PTIME 30
void pulse(int pin, int times) {
    do {
        digitalWrite(pin, HIGH);
        delay(PTIME);
        digitalWrite(pin, LOW);
        delay(PTIME);
    }
    while (times--);
}

void prog_lamp(int state) {
    if (PROG_FLICKER)
        digitalWrite(LED_PMODE, state);
}

void spi_init() {
    uint8_t x;
    SPCR = 0x53;
```

```
x=SPSR;
x=SPDR;
}

void spi_wait() {
    do {
    }
    while (!(SPSR & (1 << SPIF)));
}

uint8_t spi_send(uint8_t b) {
    uint8_t reply;
    SPDR=b;
    spi_wait();
    reply = SPDR;
    return reply;
}

uint8_t spi_transaction(uint8_t a, uint8_t b, uint8_t c, uint8_t d) {
    uint8_t n;
    spi_send(a);
    n=spi_send(b);
    //if (n != a) error = -1;
    n=spi_send(c);
    return spi_send(d);
}

void empty_reply() {
    if (CRC_EOP == getch()) {
        Serial.print((char)STK_INSYNC);
        Serial.print((char)STK_OK);
    }
    else {
        error++;
        Serial.print((char)STK_NOSYNC);
    }
}

void breply(uint8_t b) {
    if (CRC_EOP == getch()) {
        Serial.print((char)STK_INSYNC);
        Serial.print((char)b);
    }
}
```

```
        Serial.print((char)STK_OK);
    }
    else {
        error++;
        Serial.print((char)STK_NOSYNC);
    }
}

void get_version(uint8_t c) {
    switch(c) {
        case 0x80:
            breply(HWVER);
            break;
        case 0x81:
            breply(SWMAJ);
            break;
        case 0x82:
            breply(SWMIN);
            break;
        case 0x93:
            breply('S'); // serial programmer
            break;
        default:
            breply(0);
    }
}

void set_parameters() {
    // call this after reading paramter packet into buff[]
    param.devicecode = buff[0];
    param.revision   = buff[1];
    param.progtype   = buff[2];
    param.parmode    = buff[3];
    param.polling    = buff[4];
    param.selftimed  = buff[5];
    param.lockbytes  = buff[6];
    param.fusebytes  = buff[7];
    param.flashpoll  = buff[8];
    // ignore buff[9] (= buff[8])
    // following are 16 bits (big endian)
    param.eeprompoll = beget16(&buff[10]);
    param.pagesize  = beget16(&buff[12]);
}
```

```

param.eepromsize = beget16(&buff[14]);

// 32 bits flashsize (big endian)
param.flashsize = buff[16] * 0x01000000
    + buff[17] * 0x00010000
    + buff[18] * 0x00000100
    + buff[19];

}

void start_pmode() {
    spi_init();
    // following delays may not work on all targets...
    pinMode(RESET, OUTPUT);
    digitalWrite(RESET, HIGH);
    pinMode(SCK, OUTPUT);
    digitalWrite(SCK, LOW);
    delay(50);
    digitalWrite(RESET, LOW);
    delay(50);
    pinMode(MISO, INPUT);
    pinMode(MOSI, OUTPUT);
    spi_transaction(0xAC, 0x53, 0x00, 0x00);
    pmode = 1;
}

void end_pmode() {
    pinMode(MISO, INPUT);
    pinMode(MOSI, INPUT);
    pinMode(SCK, INPUT);
    pinMode(RESET, INPUT);
    pmode = 0;
}

void universal() {
    int w;
    uint8_t ch;

    fill(4);
    ch = spi_transaction(buff[0], buff[1], buff[2], buff[3]);
    breply(ch);
}

```

```

void flash(uint8_t hilo, int addr, uint8_t data) {
    spi_transaction(0x40+8*hilo,
        addr>>8 & 0xFF,
        addr & 0xFF,
        data);
}

void commit(int addr) {
    if (PROG_FLICKER) prog_lamp(LOW);
    spi_transaction(0x4C, (addr >> 8) & 0xFF, addr & 0xFF, 0);
    if (PROG_FLICKER) {
        delay(PTIME);
        prog_lamp(HIGH);
    }
}

//#define _current_page(x) (here & 0xFFFFE0)
int current_page(int addr) {
    if (param.pagesize == 32) return here & 0xFFFFFFFF0;
    if (param.pagesize == 64) return here & 0xFFFFFFE0;
    if (param.pagesize == 128) return here & 0xFFFFFFC0;
    if (param.pagesize == 256) return here & 0xFFFFF80;
    return here;
}

void write_flash(int length) {
    fill(length);
    if (CRC_EOP == getch()) {
        Serial.print((char) STK_INSYNC);
        Serial.print((char) write_flash_pages(length));
    }
    else {
        error++;
        Serial.print((char) STK_NOSYNC);
    }
}

uint8_t write_flash_pages(int length) {
    int x = 0;
    int page = current_page(here);
    while (x < length) {

```

```

    if (page != current_page(here)) {
        commit(page);
        page = current_page(here);
    }
    flash(LOW, here, buff[x++]);
    flash(HIGH, here, buff[x++]);
    here++;
}

commit(page);

return STK_OK;
}

#define EECHUNK (32)
uint8_t write_eeprom(int length) {
    // here is a word address, get the byte address
    int start = here * 2;
    int remaining = length;
    if (length > param.eepromsize) {
        error++;
        return STK_FAILED;
    }
    while (remaining > EECHUNK) {
        write_eeprom_chunk(start, EECHUNK);
        start += EECHUNK;
        remaining -= EECHUNK;
    }
    write_eeprom_chunk(start, remaining);
    return STK_OK;
}
// write (length) bytes, (start) is a byte address
uint8_t write_eeprom_chunk(int start, int length) {
    // this writes byte-by-byte,
    // page writing may be faster (4 bytes at a time)
    fill(length);
    prog_lamp(LOW);
    for (int x = 0; x < length; x++) {
        int addr = start+x;
        spi_transaction(0xC0, (addr>>8) & 0xFF, addr & 0xFF, buff[x]);
        delay(45);
    }
}

```

```

prog_lamp(HIGH);
return STK_OK;
}

void program_page() {
    char result = (char) STK_FAILED;
    int length = 256 * getch();
    length += getch();
    char memtype = getch();
    // flash memory @here, (length) bytes
    if (memtype == 'F') {
        write_flash(length);
        return;
    }
    if (memtype == 'E') {
        result = (char)write_eeeprom(length);
        if (CRC_EOP == getch()) {
            Serial.print((char) STK_INSYNC);
            Serial.print(result);
        }
        else {
            error++;
            Serial.print((char) STK_NOSYNC);
        }
        return;
    }
    Serial.print((char)STK_FAILED);
    return;
}

uint8_t flash_read(uint8_t hilo, int addr) {
    return spi_transaction(0x20 + hilo * 8,
        (addr >> 8) & 0xFF,
        addr & 0xFF,
        0);
}

char flash_read_page(int length) {
    for (int x = 0; x < length; x+=2) {
        uint8_t low = flash_read(LOW, here);
        Serial.print((char) low);
        uint8_t high = flash_read(HIGH, here);

```

```

        Serial.print((char) high);
        here++;
    }
    return STK_OK;
}

char eeprom_read_page(int length) {
    // here again we have a word address
    int start = here * 2;
    for (int x = 0; x < length; x++) {
        int addr = start + x;
        uint8_t ee = spi_transaction(0xA0, (addr >> 8) & 0xFF, addr & 0xFF, 0xFF);
        Serial.print((char) ee);
    }
    return STK_OK;
}

void read_page() {
    char result = (char)STK_FAILED;
    int length = 256 * getch();
    length += getch();
    char memtype = getch();
    if (CRC_EOP != getch()) {
        error++;
        Serial.print((char) STK_NOSYNC);
        return;
    }
    Serial.print((char) STK_INSYNC);
    if (memtype == 'F') result = flash_read_page(length);
    if (memtype == 'E') result = eeprom_read_page(length);
    Serial.print(result);
    return;
}

void read_signature() {
    if (CRC_EOP != getch()) {
        error++;
        Serial.print((char) STK_NOSYNC);
        return;
    }
    Serial.print((char) STK_INSYNC);
    uint8_t high = spi_transaction(0x30, 0x00, 0x00, 0x00);

```

```

Serial.print((char) high);
uint8_t middle = spi_transaction(0x30, 0x00, 0x01, 0x00);
Serial.print((char) middle);
uint8_t low = spi_transaction(0x30, 0x00, 0x02, 0x00);
Serial.print((char) low);
Serial.print((char) STK_OK);
}
////////////////////////////////////
////////////////////////////////////

////////////////////////////////////
////////////////////////////////////

int avrisp() {
    uint8_t data, low, high;
    uint8_t ch = getch();
    switch (ch) {
    case '0': // signon
        error = 0;
        empty_reply();
        break;
    case '1':
        if (getch() == CRC_EOP) {
            Serial.print((char) STK_INSYNC);
            Serial.print("AVR ISP");
            Serial.print((char) STK_OK);
        }
        break;
    case 'A':
        get_version(getch());
        break;
    case 'B':
        fill(20);
        set_parameters();
        empty_reply();
        break;
    case 'E': // extended parameters - ignore for now
        fill(5);
        empty_reply();
        break;
    case 'P':

```

```
start_pmode();
empty_reply();
break;
case 'U': // set address (word)
    here = getch();
    here += 256 * getch();
    empty_reply();
    break;

case 0x60: //STK_PROG_FLASH
    low = getch();
    high = getch();
    empty_reply();
    break;
case 0x61: //STK_PROG_DATA
    data = getch();
    empty_reply();
    break;

case 0x64: //STK_PROG_PAGE
    program_page();
    break;

case 0x74: //STK_READ_PAGE 't'
    read_page();
    break;

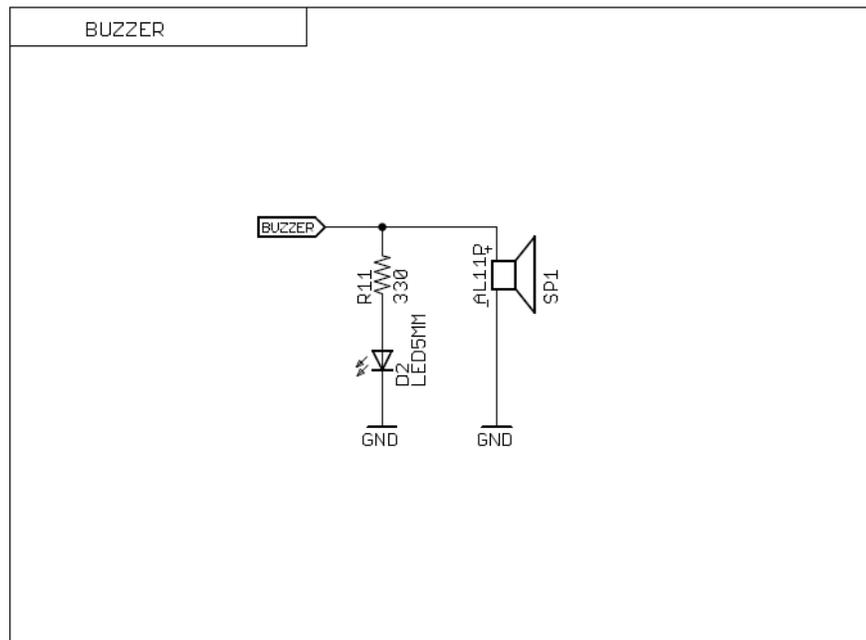
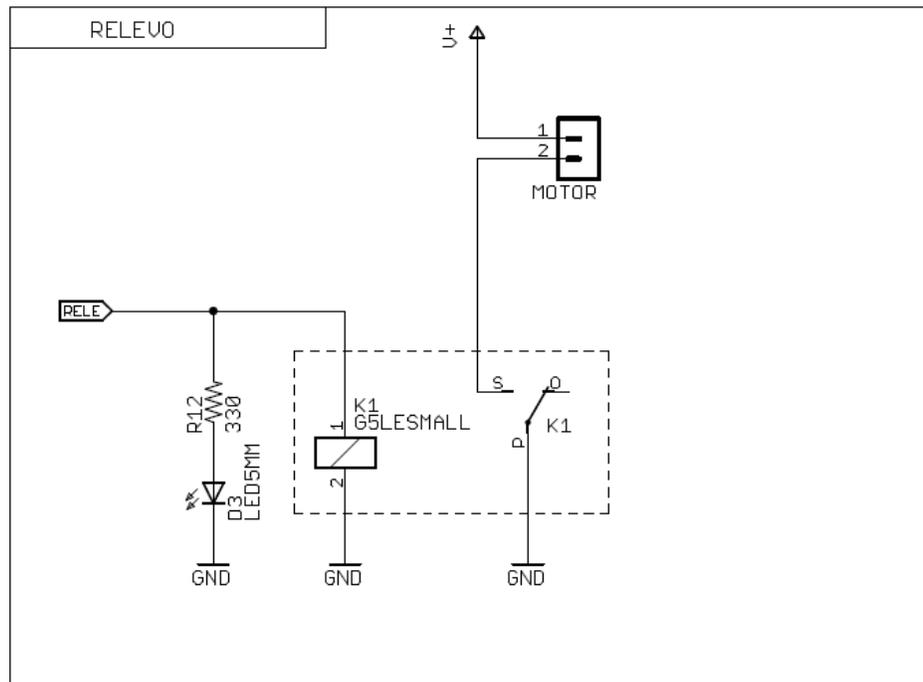
case 'V': //0x56
    universal();
    break;
case 'Q': //0x51
    error=0;
    end_pmode();
    empty_reply();
    break;

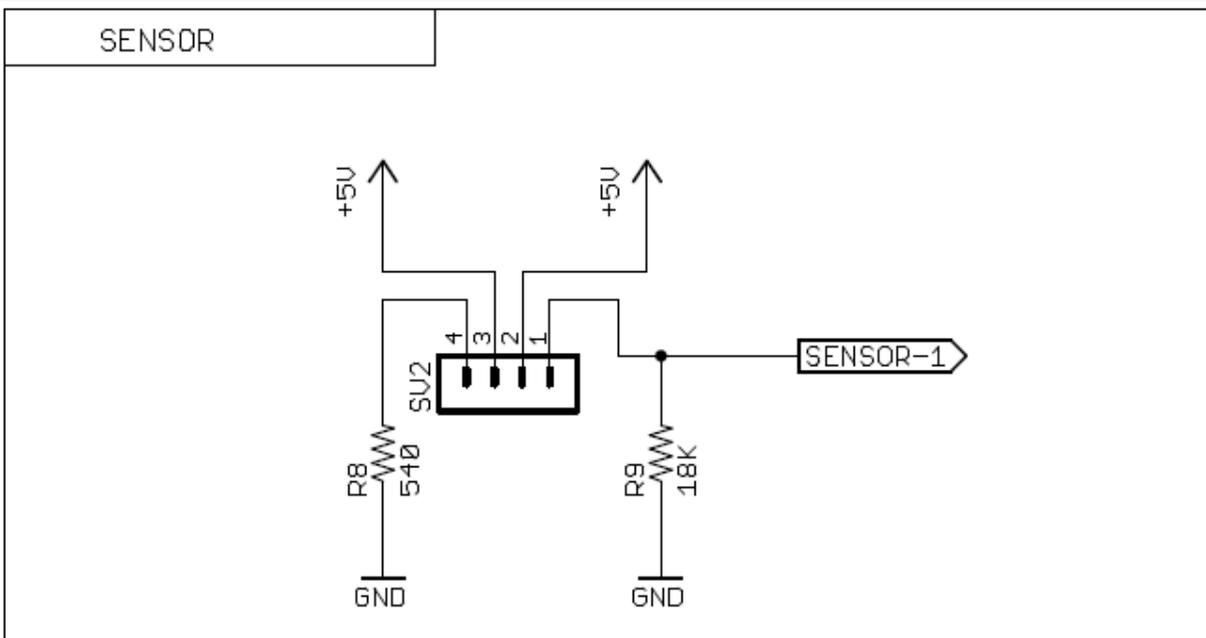
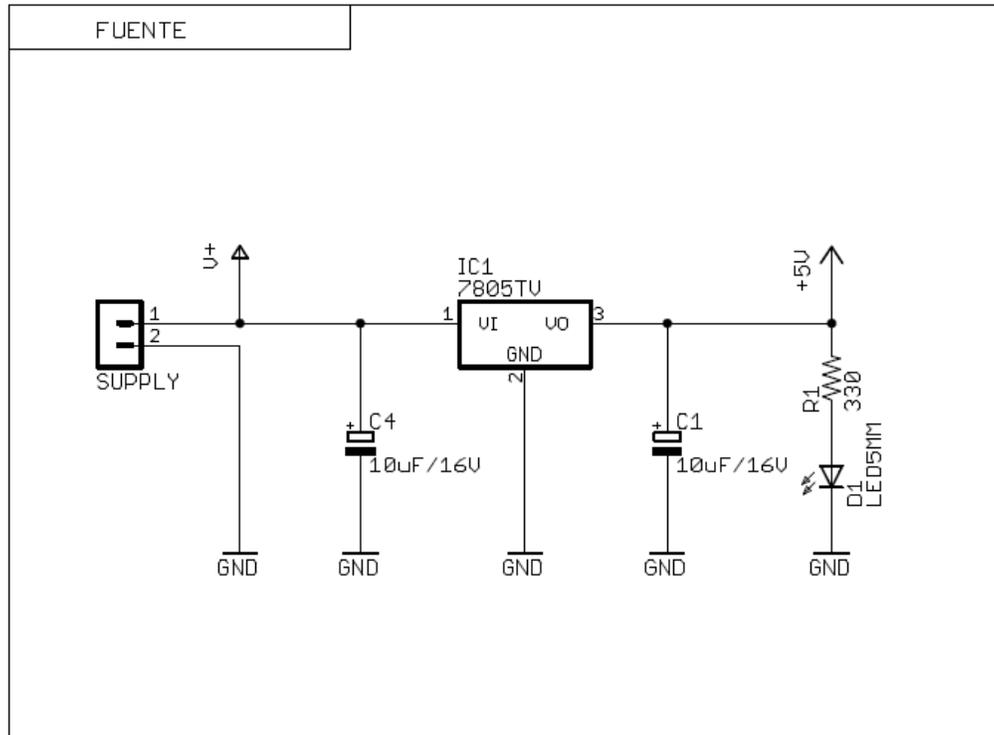
case 0x75: //STK_READ_SIGN 'u'
    read_signature();
    break;

// expecting a command, not CRC_EOP
// this is how we can get back in sync
```

```
case CRC_EOP:
    error++;
    Serial.print((char) STK_NOSYNC);
    break;

    // anything else we will return STK_UNKNOWN
default:
    error++;
    if (CRC_EOP == getch())
        Serial.print((char) STK_UNKNOWN);
    else
        Serial.print((char) STK_NOSYNC);
}
}(Garcia, 2014)
```

Anexo B. Diagrama esquemáticos del circuito electrónico



Anexo C. Fotos de pruebas.