

UNIVERSIDAD MAYOR DE SAN ANDRES
FACULTAD DE TECNOLOGIA
CARRERA DE ELECTRONICA Y TELECOMUNICACIONES



PROYECTO DE GRADO TECNICO
PRESENTADO PARA OBTENER EL GRADO DE TECNICO UNIVERSITARIO
SUPERIOR

DISEÑO DE UN SISTEMA PROTOTIPO PARA LA LOCALIZACION DE
OBJETOS PARA LAS PERSONAS DEL INSTITUTO BOLIVIANO DE LA
CEGUERRA (IBC)

POSTULANTE: CARLOS ANDRES CHUQUIMIA MENDEZ

TUTOR: MSC. LUIS RICHARD MARQUEZ GONZALES

LA PAZ - BOLIVIA

Junio, 2022

AGRADECIMIENTOS

Primeramente, agradezco a DIOS y la VIRGEN por darme, su bendición, la vida y la inteligencia para desarrollar este proyecto. También agradezco a todos mis docentes de la carrera de Electrónica y Telecomunicaciones por la formación recibida que sin ellos no sería posible esta meta. Agradezco a mi TUTOR MSC. Luis Richard Marquez Gonzales, por todo su apoyo, comprensión, confianza y sobre todo su contribución oportuna en la revisión del presente proyecto.

DEDICATORIA

Dedico este proyecto a mis padres Andres Carlos Chuquimia Endara y Betty Mendez Mercado que han sido un pilar fundamental en mi formación como profesional, por brindarme la confianza, consejos y apoyarme económicamente para lograrlo. A mis hermanos y demás familia en general por el apoyo que siempre me brindaron día a día en el transcurso de cada año de mi carrera Universitaria.

INDICE

CAPITULO I

1. ANTECEDENTES	1
1.1. PLANTEAMIENTO DEL PROBLEMA	1
1.2. OBJETIVOS.....	2
1.2.1. OBJETIVO GENERAL	2
1.2.2. OBJETIVO ESPECIFICO.....	2
1.3. JUSTIFICACION.....	2
1.4. DELIMITACION	3
1.4.1. TEMPORAL.....	3
1.4.2. ESPACIAL	3
1.5. METODOLOGIA.....	3
1.5.1. INVESTIGACION DE CAMPO.....	3
1.5.2. INVESTIGACION EXPERIMENTAL	3
1.5.3. INVESTIGACION CIENTIFICA	4
1.5.4. METODOS	4
1.5.5. TECNICAS	4

CAPITULO II

2. INGENIERIA DEL PROYECTO	5
2.1. FUNDAMENTO TEORICO.....	5
2.1.1. DISCAPACIDAD VISUAL.....	5
2.1.2. CEGUERA.....	6
2.1.3. ORIENTACION Y MOVILIDAD	7
2.1.4. TECNICAS DE ORIENTACION Y MOVILIDAD PARA INVIDENTES	7
2.1.5. HERRAMIENTAS DE NAVEGACION PARA INVIDENTES.....	8
2.1.5.1. BASTON	8
2.1.5.2. PERRO GUIA.....	9

2.1.5.3.	AYUDAS ELECTRONICAS	10
2.1.6.	TECNOLOGIAS RELACIONADAS CON LA ORIENTACION Y MOVILIDAD.....	10
2.1.6.1.	SISTEMAS DE LOCALIZACION Y POSICINAMIENTO	10
2.1.6.2.	SENSORES	11
2.1.7.	ARDUINO.....	11
2.1.7.1.	HARDWARE.....	14
2.1.7.2.	PLACAS.....	14
2.1.7.3.	PLACAS DE EXPANSIÓN (SHIELDS).....	17
2.1.7.4.	APLICACIONES.....	17
2.1.7.5.	ESQUEMAS DE CONEXIONES	18
2.1.7.5.1.	ENTRADAS Y SALIDAS.....	18
2.1.7.6.	LENGUAJE DE PROGRAMACION ARDUINO.....	18
2.1.7.7.	VENTAJAS DEL IDE ARDUINO	18
2.1.7.8.	PRIMERA EJECUCION DEL IDE ARDUINO Y CONFIGURACION INICIAL PARA EL DESARROLLO DEL PROYECTO	19
2.1.7.9.	OPCIONES DE LA INTERFAZ.....	21
2.1.7.10.	ARDUINO UNO	22
2.1.7.10.1.	CARACTERISTICAS.....	22
2.1.7.10.2.	VENTAJAS Y DESVENTAJAS DE ARDUINO UNO.....	23
2.1.7.10.3.	PARTES ESPECIFICACIONES DE LA PLACA ARDUINO UNO.....	23
2.1.7.10.4.	POTENCIA.....	25
2.1.7.10.5.	ENTRADA Y SALIDA.....	26
2.1.7.11.	SENSOR DE DISTANCIA ULTRASONICO SR04.....	27
2.1.7.11.1.	CARACTERISTICAS.....	28
2.1.7.11.2.	PINES DE CONEXIÓN.....	29
2.1.7.11.3.	FORMULA PARA MEDIR LA DISTANCIA.....	29

2.1.7.12. MODULO DE INTERFAZ TARJETA MICRO SD.....	30
2.1.7.12.1. FUNCIONAMIENTO.....	33
2.2. DESARROLLO PRACTICO / EXPERIMENTAL	34
2.2.1. DIAGRAMA DE BLOQUES.....	34
2.2.2. COMPONENTES QUEE UTILIZO PARA EL PROYECTO	35
2.2.2.1. ARDUINO UNO	35
2.2.2.2. CAPACITOR ELECTROLÍTICO	35
2.2.2.3. SENSOR ULTRASÓNICO	36
2.2.2.4. MEMORIA MICRO SD.....	36
2.2.2.5. PLAQUETA EXPERIMENTAL	36
2.2.2.6. MÓDULO LECTOR MICRO SD.....	37
2.2.3. DESARROLLO DEL PROTOTIPO.....	37
2.2.4. CODIGO ARDUINO.....	38
2.2.5. CIRCUITO FINAL	61
CAPITULO III	
3. ANALISIS DE COSTOS.....	62
3.1. COSTOS FIJOS	62
3.2. COSTOS VARIABLES.....	62
CAPITULO IV	
4. CONCLUSIONES Y RECOMENDACIONES.....	64
4.1. CONCLUSIONES	64
4.2. RECOMENDACIONES.....	64
BIBLIOGRAFIA.....	65
ANEXOS	67

INDICE DE FIGURAS

Figura 1. Personas Invidentes.....	5
Figura 2. El Glaucoma una causa de la ceguera.....	6
Figura 3. Bastón para persona no vidente.....	9
Figura 4. Perro guía o perro lazarillo.....	9
Figura 5. Diferentes tipos de Placas Arduino.....	17
Figura 6. Primer paso de ejecución de Arduino como Administrador.....	19
Figura 7. Selección de tarjeta Arduino Uno.....	20
Figura 8. Selección del Puerto Serial.....	20
Figura 9. Opciones del programa Arduino.....	21
Figura 10. Placa Arduino Uno.....	23
Figura 11. Entrada y Salida de la Placa Arduino.....	27
Figura 12. Sensor de distancia de ultrasonido HC-SR04.....	28
Figura 13. Funcionamiento del sensor HC-SR04.....	28
Figura 14. Dimensiones del Sensor HC-SR04.....	29
Figura 15. Pines de conexión del sensor HC-SR04.....	29
Figura 16. Gráfico de pulsos del sensor HC-SR04.....	30
Figura 17. Modulo lector de memoria Micro SD, modelo MOD-SD-01.....	32
Figura 18. Modulo lector de memoria Micro SD, modelo MOD-MSDC.....	32
Figura 19. Memoria Micro SD.....	33
Figura 20. Conexiones del modelo MOD-SD-01.....	33

Figura 21. Conexiones del modelo MOD-MSDC	33
Figura 22. Conexiones a la placa Arduino.	34
Figura 23. Diagrama de bloques, funcionamiento del prototipo.	34
Figura 24. Placa Arduino Uno.....	35
Figura 25. Capacitor electrolítico 100 uF x 16V.....	35
Figura 26. Sensor Ultrasónico, modelo HC-SRo4	36
Figura 27. Memoria Micro SD.....	36
Figura 28. Plaqueta experimental perforada	36
Figura 29. Modulo Micro SD.....	37
Figura 30. Conexiones del ultrasónico, modulo Micro SD y Buzzer a la placa Arduino	60

INDICE DE TABLAS

Tabla 1. Placas Arduino Uno R3, Leonardo, Diecimila y Nano	14
Tabla 2. Placas Arduino Bluetooth, lilyPad y Mini	15
Tabla 3. Placas Arduino Serial, Esplora, Mega ADK.....	16
Tabla 4. Costos fijos del proyecto	61
Tabla 5. Costos variables del proyecto	61

RESUMEN

El presente proyecto de grado técnico “DISEÑO DE UN SISTEMA PROTOTIPO PARA LA LOCALIZACION DE OBJETOS PARA LAS PERSONAS DEL INSTITUTO BOLIVIANO DE LA CEGUERA (IBC)” se basa en la integración de diferentes tecnologías estudiadas a lo largo de la carrera de Electrónica y Telecomunicaciones con la finalidad de tratar de ser un aporte con las personas que presentan discapacidad visual.

Este proyecto tiene como objetivo principal elaborar un diseño de un sistema prototipo para la localización de objetos para las personas del Instituto Boliviano de la Ceguera (IBC), indicando la distancia a la que se encuentre un obstáculo mediante un Módulo Micro SD producir una voz que le indica a cuantos centímetros está el objeto, de tal manera que este pueda evadirlo al momento de moverse.

Se utilizó herramientas como Arduino Uno, Sensor Ultrasónico y un Módulo Micro SD y software como Arduino se prosiguió con el montaje y los sensores en el prototipo para finalmente realizar las pruebas necesarias y verificar el funcionamiento del proyecto.

CAPITULO I

1. ANTECEDENTES DEL PROYECTO

En la actualidad vemos personas con discapacidad visual que pueden ser muy importante para el comienzo de una investigación o para el desarrollo de nuevos proyectos tecnológicos.

Actualmente ya existen muchos entes preocupados en la ayuda a dichas personas, proyectos dirigidos a personas con discapacidad visual ya existen escuelas de formación y enseñanza que en algunos casos usan la tecnología no se les hace nada complicado. Gracias al avance de la tecnología en software y hardware que nos permite conseguir microcontroladores, módulo ultrasónico, modulo SD, componentes electrónicos, etc.

Podemos utilizar estos recursos para diseñar un circuito de que nos permitirá ayudar a la persona no vidente y mejorar su calidad de vida.

El sistema prototipo para la localización de objetos para las personas del Instituto Boliviano de la Ceguera servirá para que una persona invidente no se golpee, la persona del instituto boliviano de la ceguera va adquirir el equipo a bajo costo.

Sin duda el uso de nuevas técnicas, dispositivos electrónicos son un eje fundamental para facilitar o al menos intentar dar una solución a su problema.

1.1. PLANTAMIENTO DE PROBLEMA

Existen organismos dedicados a ayudar a personas con algún tipo de discapacidad, como es la FENACIEBO (Federación Nacional de Ciegos del Bolivia) una organización sin fines de lucro, que impulsa el desarrollo y la integración social, cultural, educacional y económica de las personas con discapacidad visual.

Según los registros del INE (Instituto Nacional de Estadística) en La Paz hay aproximadamente 1442.0 invidentes.

Uno de los mayores inconvenientes que tienen los invidentes al momento de moverse son los obstáculos que se encuentran en la parte superior, ya que en la parte inferior se ayudan con el bastón o en ciertos casos con sus pies. Objetos como ramas de árboles, letreros, escaleras o cualquier impedimento que esté arriba de su torso y no pueda ser detectado ocasionando molestias, choques o incluso hasta que la persona se lastime es una gran dificultad desarrollada dentro de la formulación del problema.

1.2. OBJETIVOS

1.2.1. OBJETIVO GENERAL

- Diseñar un sistema prototipo para la localización de objetos para las personas del instituto boliviano de la ceguera (IBC) mediante dispositivos electrónicos con comunicación Modulo Micro SD.

1.2.2. OBJETIVO ESPECIFICO

- Diseñar un prototipo con un sensor ultrasónico incorporado en la frente para ayudar a guiarse a la persona no vidente,
- Diseñar un Software para que reproduzca las voces de alerta.
- Acoplar el sensor ultrasónico y el modulo Micro SD al Arduino Uno.
- Armar un sistema de envío y recepción de datos utilizado un Módulo Micro SD.

1.3. JUSTIFICACION

El presente proyecto propone diseñar un prototipo para el instituto boliviano de la ceguera, que cumpla con los requerimientos encuestados antes de iniciar la investigación y al mismo tiempo colaborar en ámbito social,

Con el proyecto ayudaremos a personas no videntes que tenga una comodidad con la finalidad de lograr resultados positivos.

Se busca implementar un prototipo que cumpla con los requerimientos encuestados antes de iniciar la investigación y al mismo tiempo colaborar en el ámbito social, ofertando el desarrollo en el Instituto Boliviano de la Ceguera (IBC).

En este proyecto se pretende desarrollar un dispositivo que sea útil para que la población invidente supere ciertas barreras en el entorno, facilitándoles la ejecución de sus actividades cotidianas. Además, la realización de este proyecto establece una herramienta con la que esta población tiene la posibilidad de sobresalir en actividades que mejorarán su autoestima e inclusión social.

1.4. DELIMITACION

1.4.1. TEMPORAL

El sistema se desarrollará entre los meses comprendidos desde julio de 2021 hasta agosto 2021, en el transcurso de este tiempo se realizará los objetivos planteados anteriormente.

1.4.2. ESPACIAL

El presente proyecto se llevará a cabo en el instituto boliviano de la ceguera (IBC), en la ciudad de La Paz Bolivia en paceños que tengan pérdida de visión que ayuden al proyecto.

1.5. METODOLOGIA

El presente estudio de investigación se enmarco dentro de los tipos de investigación detallados a continuación:

1.5.1. INVESTIGACION DE CAMPO

De acuerdo a que la investigación se desarrolló específicamente en el Instituto Boliviano de la Ceguera se mantuvo relación directa con las fuentes de información, se realizaron pruebas, encuestas y entrevistas a las personas del instituto.

1.5.2. INVESTIGACION EXPERIMENTAL

Puesto que el proyecto propone implementar y diseñar un prototipo, se realizaron pruebas durante el proceso de investigación en el instituto boliviano de la ceguera para verificar la funcionalidad.

1.5.3. INVESTIGACION CIENTIFICA

Con respecto al desarrollo del proyecto, su funcionamiento es con Módulo, componentes electrónicos y la programación diseñada en Arduino.

1.5.4. METODOS

Método de análisis y síntesis: Puesto que el prototipo para no videntes fue fraccionado en tres partes para notar observaciones en sus componentes, Arduino Uno, Modulo Micro SD, Sensor Ultrasónico, la unión de las partes forman el prototipo, el mismo que cumple la función de alertar y avisar a los invidentes la distancia de un objeto.

Método inductivo/deductivo: Se aplicó el proceso estudiando aspectos particulares de las actividades que se realizan en el instituto boliviano de la ceguera, estableciendo un soporte teórico general.

Método sistémico: Contribuyo con la evaluación de los aspectos dirigidos a modelar el objeto mediante la relación entre el proyecto y los colaboradores del mismo. Esta relación ayuda a determinar la estructuración del prototipo.

1.5.5. TECNICAS

Técnica de campo: Hace referencia todas las pruebas realizadas en la institución, validando el funcionamiento del proyecto. En este ambiente se incluyen técnicas de adquisición de información.

Técnica indirecta de información: Dado que la información recopilado fue tomada de fuentes o bibliografías existentes en el proceso de búsqueda de información.

Técnica documental: Debido al resultados adquiridos se usó la técnica documental para la presentación de un escrito formal del cumplimiento del plan de estudio.

CAPITULO II

2. INGENIERIA DE PROYECTO

2.1. FUNDAMENTO TEORICO

En esta sección se documentó la información relacionada con los conceptos y teorías que permiten contextualizar el proyecto, permitiendo entender las ideas básicas y generalidades de los elementos a tener en cuenta en el desarrollo del mismo. La investigación se realizó seleccionando los datos concernientes al tipo de discapacidad analizada y las herramientas de ayuda existentes en la actualidad que se asemejan en sus objetivos y funciones al prototipo diseñado.

2.1.1. DISCAPACIDAD VISUAL

Son personas que padecen de ceguera, una discapacidad física que consiste en la pérdida total o parcial del sentido de la vista. Existen varios tipos la parcial dependiendo del grado y tipo de pérdida de visión, como la visión reducida, el escotoma, la ceguera parcial (de un ojo) o el daltonismo.



Figura 1. Personas Invidentes

Fuente: <https://www.irfabolivia.org/personas-con-discapacidad-visual-se-benefician-con-bono-indigencia/>

2.1.2. CEGUERA

Según la OMS (2019) las principales causas de ceguera crónica son las cataratas, el glaucoma, la degeneración macular relacionada con la edad, las opacidades corneales, la retinopatía diabética, el tracoma y las afecciones oculares infantiles, como las causadas por la carencia de vitamina A.

Esta discapacidad relacionada con la edad y la debida a la diabetes no controlada están aumentando en todo el mundo, mientras que la ceguera de causa infecciosa está disminuyendo gracias a las medidas de salud pública. Tres cuartas partes de los casos son prevenibles o tratables. Las personas que sufren ceguera tienen distintas herramientas y mecanismos que les permiten mejorar su calidad de vida. El alfabeto Braille es un sistema de lectura y escritura táctil diseñado para ciegos. Los perros guías o lazarillos y los bastones también ayudan a la hora de trasladarse.

En este sentido tendríamos que destacar la existencia de instituciones que ayudan de manera notable a quienes se enfrentan a la pérdida del sentido primario de la vista.

En el 2017, la WHO (World Health Organization: Organización Mundial de la Salud) estimó que había 161 millones de personas (2.6 % de la población mundial) en el mundo con deterioro de la vista, de los cuales 124 millones (2% aproximadamente) tenía baja visión y 37 millones eran ciegos (cerca de 0.6%).

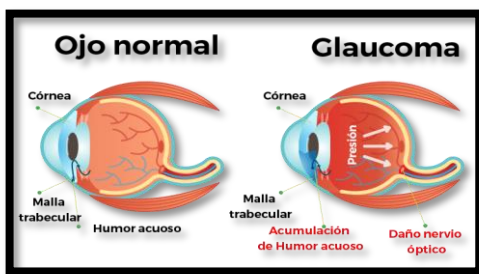


Figura 2. El Glaucoma una causa de la ceguera

Fuente: <https://tusdudasdesalud.com/vision/enfermedades-ojo/operacion-de-glaucoma/>

2.1.3. ORIENTACION Y MOVILIDAD

Se define la orientación como el proceso cognitivo que permite establecer y actualizar la posición que se ocupa en el espacio a través de la información sensorial, mientras que la movilidad, en sentido amplio, es la capacidad para desplazarse de un lugar a otro. Los especialistas en OyM (llamados en España Técnicos en Rehabilitación Básica) completan esta definición afirmando que, para que la movilidad sea adecuada, debe ser, además, independiente, segura y eficaz.

En determinados aspectos evolutivos suele haber un cierto retraso de los niños ciegos totales o deficientes visuales con respecto a sus compañeros videntes. Por ejemplo, la capacidad para establecer relaciones topológicas, como proximidad, separación u orden, se desarrolla en los videntes alrededor de los siete años y, en los invidentes, a los diez u once años. Ahora bien, si se interviene precozmente y se somete al alumno a las experiencias adecuadas, pueden paliarse las repercusiones del desfase.

Un nivel suficiente de OyM repercute en otros aspectos del desarrollo general del individuo. Así, por ejemplo, permite un adecuado repertorio conceptual, aumenta la autoestima, favorece el desarrollo psicomotor y posibilita el acceso a actividades socioculturales (desplazamiento a museos, colegios, etc.).

2.1.4. TECNICAS DE ORIENTACION Y MOVILIDAD PARA INVIDENTES

La capacidad mediante la cual un invidente establece su propia posición requiere de un proceso de aprendizaje y entrenamiento que combina capacidades mentales, intelectuales y perceptivas. Existen técnicas que ayudan al invidente a prepararse realizar estas actividades por sí solo. Para realizar un entrenamiento en orientación y movilidad se deben tener en cuenta los siguientes aspectos:

- a) **Puntos de referencia y pistas.** Objetos, sitios o características de un lugar que permiten orientarse. El invidente puede seguir pistas como sonidos y olores para identificar espacios.

- b) **Giros.** Mediante giros el invidente puede reconocer direcciones; Un cuarto de vuelta corresponde a 90 grados, media vuelta es igual a 180 grados y una vuelta completa, 360 grados.
- c) **Emplear los sentidos restantes.** La persona debe comprender toda la información que puede obtener por medio de los estímulos que recibe a través del oído, tacto y olfato.
- d) **Técnica de rastreo.** Esta técnica consiste en colocarse de lado de la pared extendiendo el brazo a la altura de la cadera, rozando la pared con el dorso de la mano, manteniendo los dedos flexionados hacia la palma, con el fin de reconocer espacios.
- e) **Técnica de encuadre.** El invidente se ubica de espalda a la pared, con los talones juntos y la punta de los pies ligeramente separados, para tomar un punto de partida para marcar una dirección hacia el objetivo en línea recta.
- f) **Técnica diagonal.** Se coloca el brazo extendido en forma diagonal sobre el cuerpo, con la palma de la mano hacia abajo, para proteger el cuerpo en caso de que la persona se tropiece con objetos a la altura de la cintura.
- g) **Técnica de cubrirse.** Se eleva el brazo a la altura de la cara y se flexiona el codo de modo que el antebrazo quede formando un ángulo de 90 grados, a una distancia de 10 centímetros. Esto permite la protección de la cara contra objetos que se encuentran a la altura de esta.

2.1.5. HERRAMIENTAS DE NAVEGACION PARA INVIDENTES

Existen instrumentos que permiten a las personas con este tipo de discapacidad a moverse por el entorno de un modo más seguro e independiente. A continuación, se describen las más destacables.

2.1.5.1. BASTON

Este elemento se utiliza para localizar obstáculos y proporcionar al individuo información sobre el entorno. Además, identifica a la persona que lo lleva como alguien

con una discapacidad visual. El común de estos bastones para este tipo de discapacidad es su color blanco.



Figura 3. Bastón para persona no vidente

Fuente: <https://es.aliexpress.com/item/32868779486.html>

La utilización del bastón largo o blanco como técnica de orientación y movilidad es la que tiene mayor utilización a nivel mundial. Además de ser económico (en comparación con los perros guías y los actuales dispositivos electrónicos de navegación), el bastón se caracteriza por una alta portabilidad y utilidad para las personas con invidencia.

2.1.5.2. PERRO GUIA

El perro guía o perro lazarillo, es un perro especialmente entrenado que apoya el normal desplazamiento de una persona con discapacidad visual. Esta técnica está orientada principalmente para la navegación exterior, ya que los perros están capacitados para viajar en diferentes medios de transporte (buses, metro, autos).



Figura 4. Perro guía o perro lazarillo.

Fuente: <https://lifestyle.fit/mas-que-fit/mascotas/razas-perro-guia-entrenamiento>

Esta es una técnica recomendada para personas adolescentes o adultas, que poseen un cierto grado de madurez y estabilidad emocional para poder hacerse cargo del animal.

2.1.5.3. AYUDAS ELECTRONICAS

Son dispositivos electrónicos que facilitan a la persona con discapacidad visual la localización e identificación de referencias y obstáculos. Esta clase de dispositivos se conocen como ETAs, por sus siglas en inglés: Electronic Travel Aids. Son formas de tecnologías de asistencia que tienen como fin apoyar y mejorar la movilidad y orientación de las personas ciegas durante la navegación en entornos cerrados o abiertos, ya sean familiares o desconocidos. En su diseño se utilizan distintas tecnologías y dispositivos electrónicos, y generalmente están enfocados a una tarea específica (esquivar obstáculos, acceder a información, etc.) Los sistemas de navegación generalmente están provistos de:

- ❖ Detección de obstáculos y peligros.
- ❖ Información de ubicación y orientación durante la navegación.
- ❖ Rutas óptimas hacia un destino deseado.

2.1.6. TECNOLOGIAS RELACIONADAS CON LA ORIENTACION Y MOVILIDAD

Los sistemas para la identificación y localización de lugares o referencias de interés, y los sistemas de orientación por posicionamiento desde satélite, se cuentan entre las herramientas que la tecnología ha proporcionado para la orientación y movilidad de las personas invidentes y deficientes visuales.

2.1.6.1. SISTEMAS DE LOCALIZACION Y POSICIONAMIENTO

Un sistema de localización se define como una combinación de tecnologías de posicionamiento que permiten la localización geográfica de unidades móviles o inmóviles. Se tienen dos tipos de sistemas, los basados en la utilización de satélites y los basados en tecnologías distintas al satélite.

Los primeros hacen uso de los satélites para posicionar un elemento en la superficie terrestre, apoyados en aparatos receptores de cálculo de posición, los cuales estiman una posición a partir de la medida de la distancia entre el objeto a localizar con un mínimo de tres satélites de ubicación conocida, mediante triangulación. Las principales aplicaciones de los sistemas de localización satelitales tienen lugar a través de los conocidos Sistemas de Navegación Global por Satélite. Estos sistemas nacieron en un marco militar con el objetivo de conocer la posición de las tropas para fines de inteligencia, sin embargo, hoy en día existen muchas aplicaciones cotidianas de estos sistemas, como los dispositivos de navegación terrestre GPS.

El sistema de posicionamiento global (GPS por sus siglas en inglés), es un sistema que permite determinar la posición las 24 horas del día, en cualquier lugar de la tierra, en cualquier condición climatológica. Un receptor GPS es un aparato electrónico pequeño, utilizado por aquellos que viajan por tierra, mar o aire, que permite recibir las señales de los satélites. Este receptor utiliza las señales de radio para calcular su posición, que es facilitada como un grupo de números y letras que corresponden a un punto sobre un mapa.

2.1.6.2. SENSORES

Un sensor es un dispositivo que convierte magnitudes físicas en valores medibles de dicha magnitud. Para las actividades de orientación y movilidad, los sensores de proximidad son los más empleados en tecnología de asistencia para este tipo de discapacidad. Existen diversos tipos de sensores de proximidad: Sensores inductivos, sensores capacitivos, sensores ultrasónicos, sensores ópticos y sensores magnéticos.

2.1.7. ARDUINO

Arduino es una compañía de hardware libre, y comunidad tecnológica, que diseña y manufactura placas de desarrollo de hardware y software compuesta respectivamente por circuitos impresos que integran un microcontrolador, y un entorno de desarrollo (IDE) en donde se programa cada placa. Arduino se enfoca en acercar y facilitar el uso de la electrónica y programación de sistemas embebidos en proyectos multidisciplinarios. Toda

la plataforma, tanto para sus componentes de hardware como de software son liberados bajo licencia de código abierto que permite libertad de acceso a los mismos.

El hardware consiste en una placa de circuito impreso con un microcontrolador, usualmente Atmel AVR, puertos digitales y analógicos de entrada/salida, los cuales pueden conectarse a placas de expansión (shields) que amplían las características de funcionamiento de la placa Arduino. Asimismo, posee un puerto de conexión USB desde donde se puede alimentar la placa y establecer comunicación serial con el computador.

Por otro lado, el software consiste en un entorno de desarrollo (IDE) basado en el entorno de Processing y lenguaje de programación basado en Wiring, así como en el cargador de arranque (bootloader) que es ejecutado en la placa. El microcontrolador de la placa se programa a través de un computador, haciendo uso de comunicación serial mediante un convertidor de niveles RS-232 a TTL serial.

La primera placa Arduino fue introducida en el 2005, ofreciendo un bajo costo y facilidad de uso para novatos y profesionales buscando desarrollar proyectos interactivos con su entorno mediante el uso de actuadores y sensores. A partir de octubre de 2012, se incorporaron nuevos modelos de placas de desarrollo que hacen uso de microcontroladores Cortex M3, ARM de 32 bits, que coexisten con los originales modelos que integran microcontroladores AVR de 8 bits. ARM y AVR no son plataformas compatibles en cuanto a su arquitectura y por lo tanto su set de instrucciones, pero se pueden programar y compilar bajo el IDE predeterminado de Arduino sin ningún cambio.

Las placas Arduino están disponibles de forma ensambladas o en forma de Kits "Hazlo tú mismo" (por sus siglas en inglés <<DIY>>). Los esquemáticos de diseño del Hardware están disponibles bajo licencia Libre, permitiendo a cualquier persona crear su propia placa Arduino sin necesidad de comprar una prefabricada. Adafruit Industries estimó a mediados del año 2011 que alrededor de 300,000 placas arduinos habían sido producidas comercialmente, y en el año 2013 estimó que alrededor de 700.000 placas oficiales de la empresa Arduino se puede utilizar para desarrollar objetos interactivos autónomos o puede ser conectado a software tal como Adobe Flash, Processing, Max/MSP, Pure Data, etc.

Una tendencia tecnológica es utilizar Arduino como tarjeta de adquisición de datos desarrollando interfaces en software como JAVA, Visual Basic y LabVIEW. Las placas se pueden montar a mano o adquirirse. El entorno de desarrollo integrado libre se puede descargar gratuitamente.

Hay muchos otros microcontroladores y plataformas con microcontroladores disponibles para la computación física. Parallax Basic Stamp, BX-24 de Netmedia, Phidgets, Handyboard del MIT, y muchos otros ofrecen funcionalidades similares.

Todas estas herramientas organizan el complicado trabajo de programar un Microcontrolador en paquetes fáciles de usar. Arduino, además de simplificar el proceso de trabajar con microcontroladores, ofrece algunas ventajas respecto a otros sistemas a profesores, estudiantes y amateurs:

- a) **Asequible.** Las placas Arduino son más asequibles comparadas con otras plataformas de microcontroladores. La versión más cara de un módulo de Arduino puede ser montada a mano, e incluso ya montada cuesta bastante menos de 50\$us. Lo único que “vale” en la placa son los componentes, ya que no debemos pagar el costo de la licencia de su creador, por el hecho de ser libre.
- b) **Multi-Plataforma.** El software de Arduino funciona en los sistemas operativos Windows, Macintosh OSX y Linux. La mayoría de los entornos para microcontroladores están limitados a Windows.
- c) **Entorno de programación simple y directo.** El entorno de programación de Arduino es fácil de usar para principiantes y lo suficientemente flexible para los usuarios avanzados. Pensando en los profesores, Arduino está basado en el entorno de programación de Processing con lo que el estudiante que aprenda a programar en este entorno se sentirá familiarizado con el entorno de desarrollo Arduino.
- d) **Software ampliable y de código abierto.** El software Arduino está publicado bajo una licencia libre y preparado para ser ampliado por programadores experimentados. El lenguaje puede ampliarse a través de librerías de C++, y si se está interesado en profundizar en los detalles técnicos, se puede dar el salto a la

programación en el lenguaje AVR C en el que está basado. De igual modo se puede añadir directamente código en AVR C en tus programas si así lo deseas.

- e) **Hardware ampliable y de Código abierto.** Arduino está basado en los microcontroladores ATMEGA168, ATMEGA328 y ATMEGA1280. Los planos de los módulos están publicados bajo licencia Creative Commons, por lo que diseñadores de circuitos con experiencia pueden hacer su propia versión del módulo, ampliándolo u optimizándolo. Incluso usuarios relativamente inexpertos pueden construir la versión para placa de desarrollo para entender cómo funciona y ahorrar algo de dinero

2.1.7.1. HARDWARE

Los modelos de Arduino se categorizan en: placas de desarrollo, placas de expansión (shields), kits, accesorios e impresoras 3d.

2.1.7.2. PLACAS

Arduino Nano, Arduino Uno, Arduino Leonardo, Arduino Due, Arduino Yún, Arduino Tre (En Desarrollo), Arduino Zero, Arduino Micro, Arduino Esplora, Arduino Mega ADK, Arduino Ethernet, Arduino Mega 2560, Arduino Robot, Arduino Mini, LilyPad Arduino Simple, LilyPad Arduino SimpleSnap, LilyPad Arduino, LilyPad Arduino USB, Arduino Pro Mini, Arduino Fio, Arduino Pro, Arduino MKR1000/Genuino MKR1000, Arduino MICRO/Genuino MICRO, Arduino 101/Genuino 101, Arduino Gemma.

Tabla 1. Placas Arduino Uno R3, Leonardo, Diecimila y Nano

	Arduino UNO R3: es un módulo que se basa en el ATMEGA328. Tiene 14 entradas/salidas, de las cuales 6 se pueden usar como PWM, 6 salidas analógicas, un cristal de 16 MHz, una conexión USB, un Jack de alimentación, un botón de reset y un "header" ICSP. El Arduino UNO R3 ya no lleva el FTDI para la conexión USB, pero usa el ATMEGA16U2.
---	--





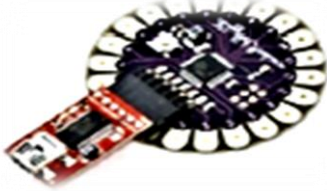
	<p>Arduino Leonardo: Se basa en el micro ATMEGA32u4. Cuenta con 20 entrada/salidas, de las cuales 7 pueden ser salidas PWM y 12 entradas analógicas. Como oscilador tiene un cristal de 16MHz.</p>
	<p>Arduino Diecimila: Esta es la placa Arduino más popular. Se conecta al ordenador con un cable estándar USB y contiene todo lo que necesitas para programar y usar la placa. Puede ser ampliada con variedad de dispositivos y/o escudos (shields) con características específicas.</p>
	<p>Arduino Nano: Una placa compacta diseñada para uso como tabla de pruebas, el Nano se conecta al ordenador usando un cable USB Mini-B.</p>

Tabla 2. Placas Arduino Bluetooth, LilyPad y Mini.

	<p>Arduino Bluetooth: Arduino BT contiene un módulo bluetooth que permite comunicación y programación sin cables. Es compatible con los dispositivos Arduino.</p>
	<p>Arduino LilyPad: Diseñada para aplicaciones listas para llevar, esta placa puede ser conectada en fábrica, y un estilo sublime.</p>





	<p>Arduino Mini: Esta es la placa más pequeña de Arduino. Trabaja bien en tabla de pruebas o para aplicaciones en las que prima el espacio. Se conecta al ordenador usando el cable Mini USB.</p>
---	---

Tabla 3. Placas Arduino Serial, Esplora, Mega ADK

	<p>Arduino Serial: Es una placa básica que RS232 como un interfaz con el ordenador para programación y comunicación. Esta placa es fácil de ensamblar incluso como ejercicio de aprendizaje.</p>
	<p>Arduino Esplora: Es un controlador listo para usarse. Te permite explotar las infinitas posibilidades que tiene en el mundo de sensores y se puede comenzar a trabajar con una palanca de mando tipo joystick y con los sensores de luz y temperatura.</p>
	<p>Arduino Mega ADK: Esta placa esta basada en el Arduino MEGA 2560 y dispone de un puerto USB Host incorporado. Arduino MEGA ADK es una placa basada en microcontrolador ATMEGA2560. Dispone de un puerto USB host interfaz para conectarlo con dispositivos basados en Android basados en el chip MAX3421e. Dispone de 54 pines I/O (14 de los cuales con PWM), 16 entradas analógicas, 4 puertos UARTs (por hardware), un cristal de 16 MHz, un conector USB, un conector de alimentación Jack, pines para ICSP y botón de RESET.</p>

2.1.7.3. PLACAS DE EXPANSION (SHIELDS)

Arduino GSM Shield, Arduino Ethernet Shield, Arduino WiFi Shield, Arduino Wireless SD Shield, Arduino USB Host Shield, Arduino Motor Shield, Arduino Wireless Proto Shield, Arduino Proto Shield.

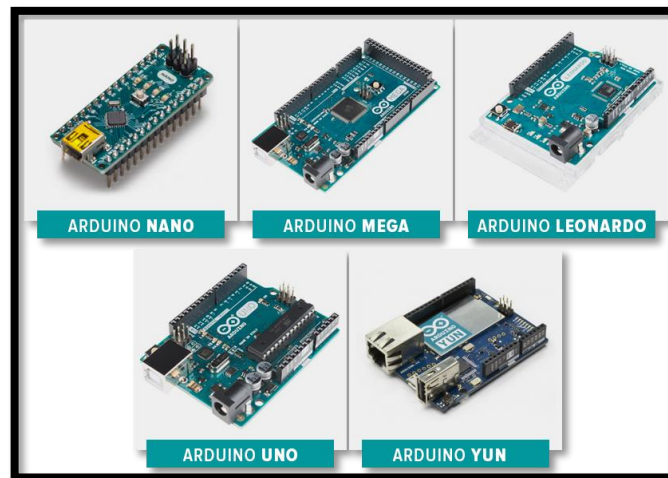


Figura 5. Diferentes tipos de Placas Arduino.

Fuente: <https://arduino.cl/que-es-arduino/>

2.1.7.4. APLICACIONES

La plataforma Arduino ha sido usada como base en diversas aplicaciones electrónicas:

- a) **Xoscillo:** Osciloscopio de código abierto.
- b) **Arduinome:** Un dispositivo controlador MIDI.
- c) **OBduino:** un económetro que usa una interfaz de diagnóstico a bordo que se halla en los automóviles modernos.
- d) **SCA-ino:** Sistema de cómputo automotriz capaz de monitorear sensores como el TPS, el MAP y el O2S y controlar actuadores automotrices como la bobina de ignición, la válvula IAC y aceleradores electrónicos.
- e) **Humane Reader:** dispositivo electrónico de bajo coste con salida de señal de TV que puede manejar una biblioteca de 5000 títulos en una tarjeta microSD.

- f) **The Humane PC:** equipo que usa un módulo Arduino para emular un computador personal, con un monitor de televisión y un teclado para computadora.
- g) **Ardupilot:** software y hardware de aeronaves no tripuladas.
- h) **ArduinoPhone:** un teléfono móvil construido sobre un módulo Arduino.
- i) Máquinas de control numérico por computadora (CNC).
- j) **Open Theremín Uno:** Versión digital de hardware libre del instrumento Theremín.

2.1.7.5. ESQUEMAS DE CONEXIONES

2.1.7.5.1. ENTRADAS Y SALIDAS

Poniendo de ejemplo al módulo Diecimila, este consta de 14 entradas digitales configurables como entradas y/o salidas que operan a 5 voltios. Cada contacto puede proporcionar o recibir como máximo 40 mA. Los contactos 3, 5, 6, 9, 10 y 11 pueden proporcionar una salida PWM (Pulse Width Modulation). Si se conecta cualquier cosa a los contactos 0 y 1, eso interferirá con la comunicación USB.

Diecimila también tiene 6 entradas analógicas que proporcionan una resolución de 10 bits. Por defecto, aceptan de 0 hasta 5 voltios, aunque es posible cambiar el nivel más alto, utilizando el contacto Aref y algún código de bajo nivel.

2.1.7.6. LENGUAJE DE PROGRAMACION ARDUINO

Arduino está basado en C y soporta todas las funciones del estándar C y algunas de C++.

El entorno de programación IDE programa la tarjeta Arduino desde un ambiente gráfico que permite para fines didácticos avanzar con mayor velocidad.

2.1.7.7. VENTAJAS DEL IDE ARDUINO

Tiene un lenguaje simple, basado en C/C++.

- Permite desde un primer contacto estar programando directamente al hardware.
- Es un proyecto open-source, por lo que debido a su precio podemos probar y experimentar sobre la misma tarjeta.

- Tiene una comunidad de desarrollo alrededor que permite un acceso a referencias, ejemplos, proyectos de gran ayuda.

2.1.7.8. PRIMERA EJECUCION DEL IDE ARDUINO Y CONFIGURACION INICIAL PARA EL DESARROLLO DEL PROYECTO

Tras instalar el controlador de Arduino, ahora realizaremos una primera ejecución para configurar el IDE de desarrollo, pulsaremos con el botón derecho del ratón sobre el ejecutable "arduino.exe" de la carpeta descomprimida anteriormente y seleccionaremos "Ejecutar como administrador" (no es necesario, pero así evitamos posibles problemas de limitaciones de Windows 7):

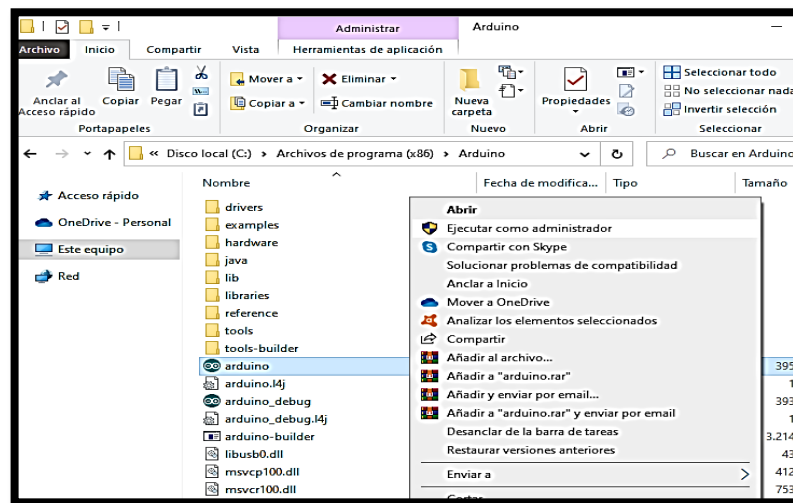


Figura 6. Primer paso de ejecución de Arduino como Administrador

Fuente: Propia

En el IDE de desarrollo de Arduino, en primer lugar, seleccionaremos el tipo de dispositivo (Tarjeta). Para ello pulsaremos en el menú "Herramientas" - "Tarjeta" y seleccionaremos "Arduino Nano":

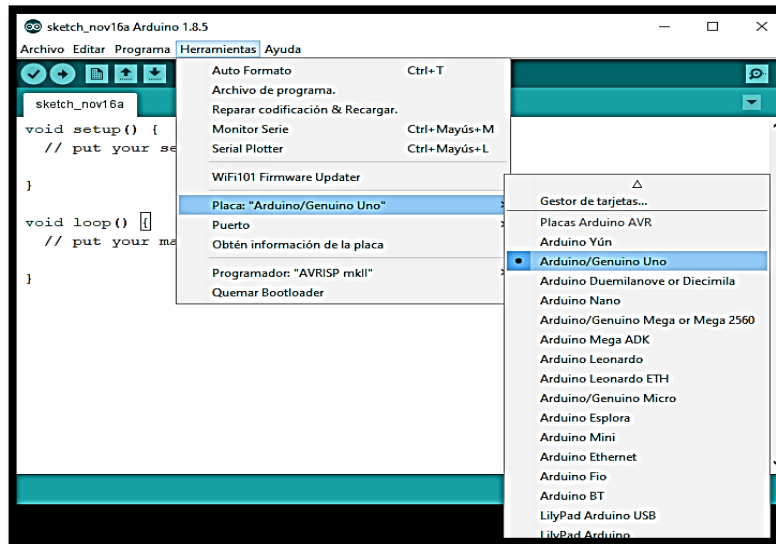


Figura 7. Selección de tarjeta Arduino Uno

Fuente: Propia

Seleccionaremos también el puerto serie asignado al controlador de Arduino (en nuestro caso COM1), para ello accederemos al menú "Herramientas"- "Puerto Serial" - "COM1":

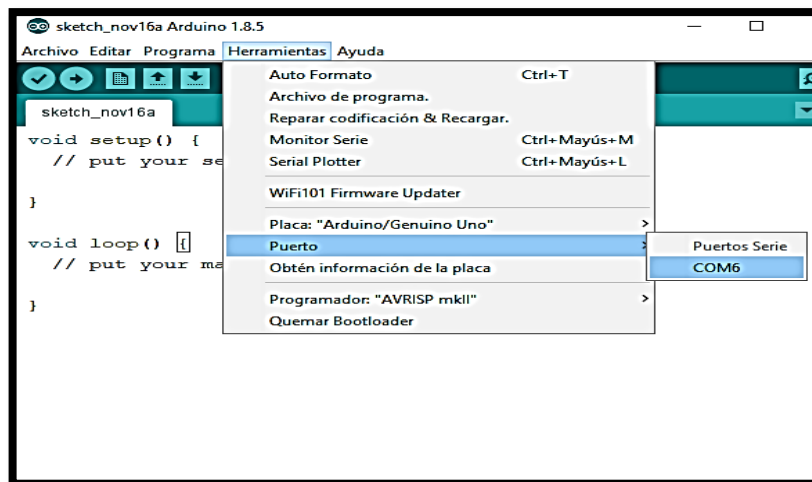


Figura 8. Selección del Puerto Serial

Fuente: Propia

2.1.7.9. OPCIONES DE LA INTERFAZ



Verificar/Compilar. Revisa el código, identificando los errores



Crea un nuevo proyecto



Presenta un menú para abrir proyectos previamente guardados y también muestra los proyectos de ejemplos.



Guarda el proyecto actual dentro de la carpeta sketch/. Se puede guardarlo también con un nombre distinto por el menú Archivo → Guardar como



Descarga el programa compilado desde el PC hasta la tarjeta Arduino



Realiza el monitoreo del puerto serial, visualiza la data enviada desde la tarjeta Arduino.

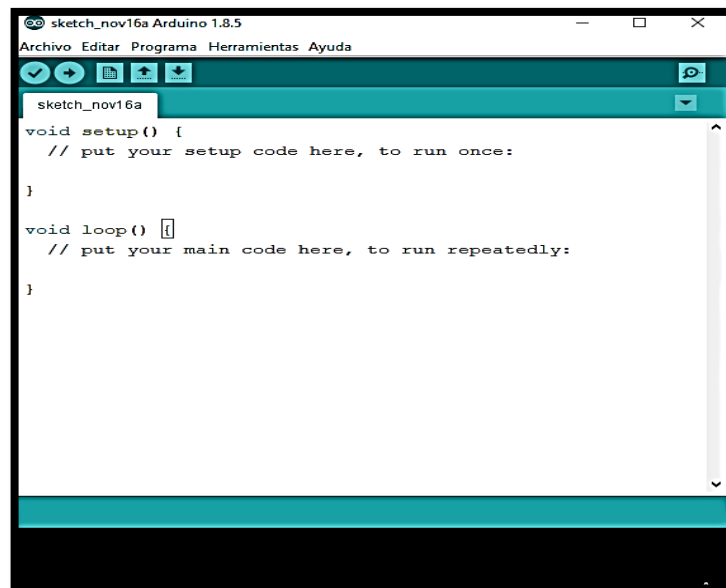


Figura 9. Opciones del programa Arduino

Fuente: Propia

2.1.7.10. ARDUINO UNO

Esencialmente la placa Arduino Uno es una placa electrónica basada en el chip de Atmel ATmega328. Tiene 14 pines digitales de entrada / salida, es el Arduino Pinout de los cuales 6 los puede utilizar como salidas PWM, 6 entradas analógicas, un oscilador de cristal de 16 MHz, una conexión USB, un conector de alimentación, una cabecera ICSP y un botón de reset. El software de la placa incluye un controlador USB que puede simular un ratón, un teclado y el puerto serie.

Esta placa tiene todo lo necesario para apoyar el microcontrolador basta con conectarlo a un ordenador con un cable USB o con un adaptador.

El Arduino Uno se diferencia de todas las placas anteriores en que no utiliza el FTDI USB a serie driver chip. En lugar de ello, cuenta con el Atmega8U2 programado como convertidor de USB a serie.

2.1.7.10.1. CARACTERISTICAS

- Microcontrolador: **ATmega328**
- Voltaje de operación: **5V**
- Voltaje de entrada (recomendado): **7-12V**
- Voltaje de entrada (límites): **6-20V**
- Pines de E/S digitales: **14 (de los cuales 6 proporcionan salida PWM)**
- Pines de entrada analógica: **6**
- Corriente DC por pin de E/S: **40 mA**
- Corriente DC para 3.3V Pin: **50 mA**
- Memoria Flash: **32 KB de los cuales 0,5 KB utilizados por el bootloader**
- SRAM: **2 KB (ATmega328)**
- EEPROM: **1 KB (ATmega328)**
- Velocidad de reloj: **16 MHz**

2.1.7.10.2. VENTAJAS Y DESVENTAJAS DE ARDUINO UNO

Es muy bueno para llevar a cabo un proyecto específico que puedas tener en mente, especialmente si puedes encontrar el escudo adecuado, pero no te enseña mucho sobre los microcontroladores y el AVR en particular. Para obtener resultados rápidos, es genial pero no te ayudará a aprender microelectrónica o programación.

a) Ventajas

- No se requiere mucho conocimiento para empezar.
- Un coste bastante bajo, dependiendo de los shields que necesites.
- Hay muchos bocetos y escudos disponibles.
- No se necesita un programador externo o una fuente de energía.

b) Desventajas

- No se entiende el microcontrolador AVR.
- Los programas y escudos pueden ser difíciles de modificar.
- No se incluye un depurador para comprobar los guiones.
- No tienes experiencia en C o en herramientas de desarrollo profesional

2.1.7.10.3. PARTES ESPECIFICACIONES DE LA PLACA ARDUINO UNO

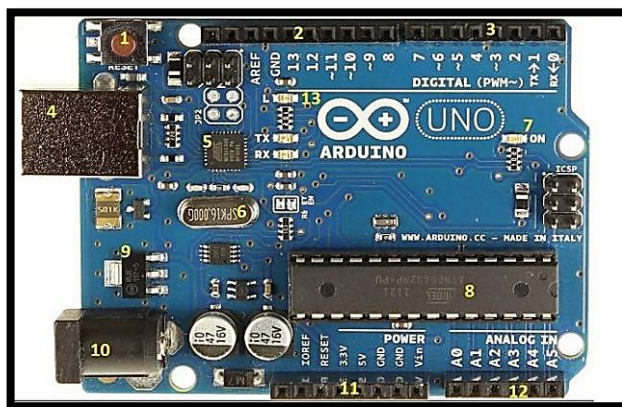


Figura 10. Placa Arduino Uno

Fuente: <https://www.hwlibre.com/arduino-uno/>

Nº 1. Botón de reset. Sirve para inicializar nuevamente el programa cargado en el microcontrolador de la placa. Cuando deje de responder el Arduino Uno es el botón de encendido o apagado para que vuelva a restablecerse.

Nº 2 y 3. Pines o puertos de entrada y salida, son los pines donde conectar los sensores, componentes y actuadores que necesiten de señales digitales.

Nº 4. Puerto USB. Utilizado tanto para conectar con un ordenador y transferir o cargar los programas al microcontrolador como para dar electricidad al Arduino. También se usa como puerto de transferencia serie a la placa, tanto para transmisión como para recepción de datos.

Nº 5. Chip de interface USB, es el encargado de controlar la comunicación con el puerto USB.

Nº 6. Reloj oscilador. Es el elemento que hace que el Arduino vaya ejecutando las instrucciones. Es el encargado de marcar el ritmo al cual se debe ejecutar cada instrucción del programa.

Nº 7. Led de encendido. Es un pequeño LED que se ilumina cuando la placa está correctamente alimentada.

Nº 8. Microcontrolador. Este es el cerebro de cualquier placa Arduino. Es el procesador que se encarga de ejecutar las instrucciones de los programas.

Nº 9. Regulador de tensión. Este sirve para controlar la cantidad de electricidad que se envía a los pines, con lo que asegura que no se estropee lo que conectemos a dichos pines.

Nº 10. Puerto de corriente continua. Este puerto es el que se usa para darle electricidad a la placa si no se usa alimentación USB.

Nº 11. Zócalo de tensión. Aquí estarán los pines con los que alimentaremos nuestro circuito.

Nº 12. Entradas analógicas. Zócalo con distintos pines de entrada analógica que permiten leer entradas analógicas.

2.1.7.10.4. POTENCIA

El Arduino Uno puede ser alimentado a través de la conexión USB o con una fuente de alimentación externa. La fuente de alimentación se selecciona automáticamente. La alimentación externa (no USB) puede provenir de un adaptador de CA a CC (de pared) o de una batería. El adaptador se puede conectar conectando un enchufe positivo central de 2,1 mm en el conector de alimentación de la placa. Los cables de una batería se pueden insertar en las clavijas Gnd y Vin del conector POWER.

La placa puede funcionar con una alimentación externa de 6 a 20 voltios. Sin embargo, si se suministra con menos de 7V, la clavija de 5V puede suministrar menos de cinco voltios y la placa puede ser inestable. Si se utilizan más de 12V, el regulador de voltaje puede sobrecalentarse y dañar la placa. El rango recomendado es de 7 a 12 voltios. Los pines de potencia son los siguientes:

- a) **VIN.** El voltaje de entrada a la placa Arduino cuando está usando una fuente de alimentación externa (a diferencia de los 5 voltios de la conexión USB u otra fuente de alimentación regulada). Puede suministrar tensión a través de esta clavija o, si lo hace a través de la toma de corriente, acceder a ella a través de esta clavija.
- b) **5V.** Esta clavija emite un 5V regulado desde el regulador de la tarjeta. La tarjeta puede alimentarse ya sea desde el conector de alimentación de CC (7 – 12 V), el conector USB (5 V) o la clavija VIN de la tarjeta (7-12 V). La alimentación de tensión a través de las clavijas de 5V o 3,3V puentea el regulador y puede dañar la placa. No lo aconsejamos.
- c) **3V3.** Una alimentación de 3,3 voltios generada por el regulador de a bordo. El consumo máximo de corriente es de 50 mA.
- d) **GND.** Pins de tierra.

2.1.7.10.5. ENTRADA Y SALIDA

Cada uno de los 14 pines digitales de la Uno puede utilizarse como entrada o salida, utilizando las funciones `pinMode()`, `digitalWrite()` y `digitalRead()`. Funcionan a 5 voltios. Cada clavija puede proporcionar o recibir un máximo de 40 mA y tiene una resistencia pull-up interna (desconectada por defecto) de 20-50 kOhms. Además, algunos pines tienen funciones especializadas:

- a) **Serial:** 0 (RX) y 1 (TX). Se utiliza para recibir (RX) y transmitir (TX) datos en serie TTL. Estos pines están conectados a los pines correspondientes del chip Serial ATmega8U2 USB-to-TTL.
- b) **Interrupciones externas:** 2 y 3. Estos pines pueden configurarse para activar una interrupción en un valor bajo, un flanco ascendente o descendente, o un cambio de valor. Vea la función `attachInterrupt()` para más detalles.
- c) **PWM:** 3, 5, 6, 9, 10 y 11. Proporciona salida PWM de 8 bits con la función `analogWrite()`.
- d) **SPI:** 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Estos pines soportan la comunicación SPI utilizando la biblioteca SPI.
- e) **LED:** 13. Hay un LED incorporado conectado al pin 13 digital. Cuando la clavija es de valor ALTO, el LED se enciende, cuando la clavija es BAJA, se apaga.

La Uno tiene 6 entradas analógicas, etiquetadas de A0 a A5, cada una de las cuales proporciona 10 bits de resolución (es decir, 1024 valores diferentes). Por defecto miden de tierra a 5 voltios, aunque es posible cambiar el extremo superior de su rango usando el pin AREF y la función `analogReference()`. Además, algunos pines tienen funcionalidad especializada:

- **TWI:** Pin A4 o SDA y pin A5 o SCL. Soporta la comunicación TWI usando la biblioteca Wire.

Hay un par de alfileres más en el tablero:

- **AREF.** Tensión de referencia para las entradas analógicas. Se utiliza con `analogReference()`.

- **RESET.** Lleve esta línea a un nivel BAJO para reiniciar el microcontrolador. Típicamente se usa para añadir un botón de reinicio a los escudos que bloquean el que está en la placa.

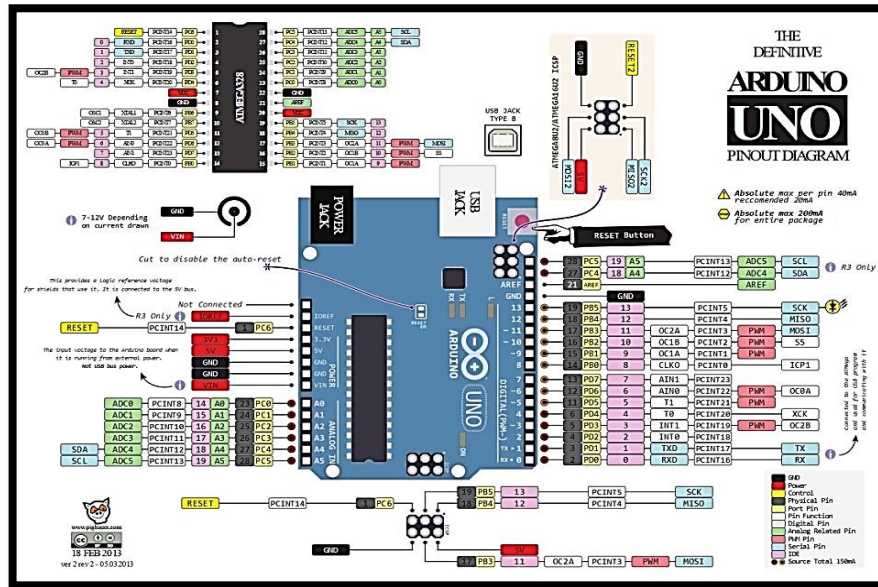


Figura 11. Entrada y Salida de la Placa Arduino

Fuente: <https://zaragozmakerspace.com/index.php/lessons/cursos-arduino-robotica-entradas-y-salidas/>

2.1.7.11. SENSOR DE DISTANCIA ULTRASONICO SR04

Es un sensor de distancias por ultrasonidos capaz de detectar objetos y calcular la distancia a la que se encuentra en un rango de 2 a 400 cm. El sensor funciona por ultrasonidos y contiene toda la electrónica encargada de hacer la medición. Su uso es tan sencillo como enviar el pulso de arranque y medir la anchura del pulso de retorno. De muy pequeño tamaño, el HC-SR04 se destaca por su bajo consumo, gran precisión y bajo precio por lo que está reemplazando a los sensores polaroid en los robots más recientes.

Los ciclos repetitivos de programación dentro del controlador permiten obtener un dato mucho más preciso de longitud con unidad de media que es el centímetro regulado por el Sistema Internacional de Medidas. Se alimenta de 5 voltios de corriente continua, la señal

de control es de nivel TTL. El consumo máximo de energía es de 20mA concluyendo un promedio de 100mW de potencia absorbida.



Figura 12. Sensor de distancia de ultrasonido HC-SR04

Fuente: <https://www.hwlibre.com/hc-sr04/>

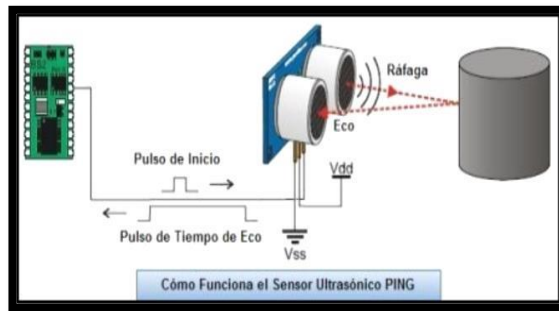


Figura 13. Funcionamiento del sensor HC-SR04

Fuente: <https://www.infootec.net/sensor-ultrasonidos-hc-sr04/>

2.1.7.11.1. CARACTERISTICAS

- ❖ Dimensiones del circuito: 43 x 20 x 15 mm.
- ❖ Tensión de alimentación: 5 Vcc.
- ❖ Frecuencia de trabajo: 40 KHz.
- ❖ Rango máximo: 4.0 m.
- ❖ Rango mínimo: 2.0 cm.
- ❖ Duración mínima del pulso de disparo (nivel TTL): 10 μ S.
- ❖ Duración del pulso eco de salida (nivel TTL): 100-25000 μ S.
- ❖ Tiempo mínimo de espera entre una medida y el inicio de otro 20 Ms.

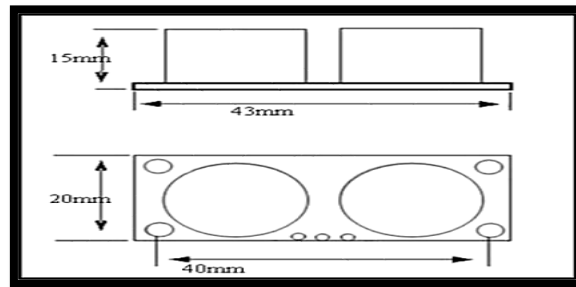


Figura 14. Dimensiones del Sensor HC-SR04

Fuente: <https://proyectoarduino.com/sensor-de-ultrasonidos-medir-distancia-con-arduino/>

2.1.7.11.2. PINES DE CONEXIÓN

- VCC
- Trig (Disparo del ultrasonido)
- Echo (Recepción del ultrasonido)
- GND



Figura 15. Pines de conexión del sensor HC-SR04

Fuente: <https://proyectedeinstrumentacion.wordpress.com/tag/hc-sr04/>

2.1.7.11.3. FORMULA PARA MEDIR LA DISTANCIA

Aprovechando que la velocidad de dicho ultrasonido en el aire es de valor 340 m/s, o 0,034 cm/microseg (ya que trabajaremos con centímetros y microsegundos). Para calcular la distancia, recordaremos que $v=d/t$ (definición de velocidad: distancia recorrida en un determinado tiempo).

De la fórmula anterior despejamos d , obteniendo $d=v \cdot t$, siendo v la constante anteriormente citada y t el valor devuelto por el sensor.

También habrá que dividir el resultado entre 2 dado que el tiempo recibido es el tiempo de ida y vuelta.

$$\text{Distancia} = \{(\text{Tiempo entre Trig y el Echo}) * (\text{V.Sonido } 340 \text{ m/s})\} / 2$$

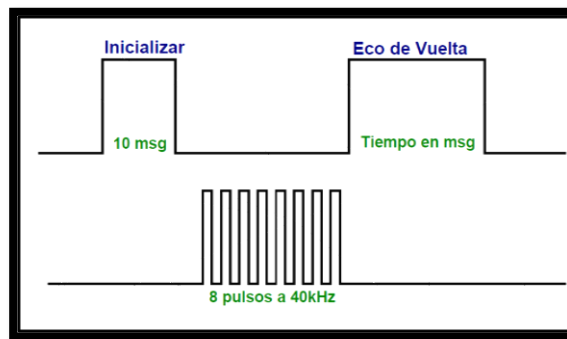


Figura 16. Gráfico de pulsos del sensor HC-SR04

Fuente: <https://ardubasic.wordpress.com/tag/hc-sr04/>

2.1.7.12. MODULO DE INTERFAZ TARJETA MICRO SD

Un lector SD es un dispositivo que permite emplear como almacenamiento una tarjeta SD, que podemos incorporar en nuestros proyectos de electrónica y Arduino.

Las tarjetas SD y micro SD se han convertido en un estándar, desplazando a otros medios de almacenamiento de datos debido a su gran capacidad y pequeño tamaño. Por este motivo han sido integradas en una gran cantidad de dispositivos, siendo en la actualidad componentes frecuentes en ordenadores, tablets y smartphones, entre otros.

Dentro del mundo de Arduino, es posible encontrar lectores de bajo coste tanto para tarjetas SD como micro SD. Los primeros en aparecer fueron los lectores SD y posteriormente los micro SD. Por tanto, en general, los módulos con micro SD son modelos más modernos que los de SD.

En ambos tipos de lectores, la lectura puede realizarse a través de bus SPI. Aunque pueden disponer de otros interfaces, como bus I2C o UART, normalmente es preferible emplear SPI por su alta tasa de transferencia.

Respecto a las tarjetas empleadas, podemos emplear tarjetas SD o SDSC (Standard Capacity) o SDHC (High Capacity), pero no SDXC (Extended Capacity). Deberá estar formateada en sistema de archivos FAT16 o FAT32.

La tensión de alimentación es de 3.3V, pero en la mayoría de los módulos se incorpora la electrónica necesaria para conectarlo de forma sencilla a Arduino, lo que frecuentemente incluye un regulador de voltaje que permite alimentar directamente a 5V.

Emplear una tarjeta SD o micro SD en con Arduino tiene la ventaja de proporcionar una memoria casi ilimitada para nuestros proyectos. Además, es no volátil (es decir, resiste cuando se elimina la alimentación), y puede ser extraída y conectada a un ordenador con facilidad.

La gran desventaja es que supone una importante carga de trabajo para Arduino. Sólo el programa ocupará el 40% de la memoria Flash, y casi el 50% de la memoria dinámica. El uso del procesador también es exigente.

En general, manejar una tarjeta micro SD o SD puede considerarse en al límite de la capacidad de Arduino como procesador. Aunque es posible emplearlas en proyectos sencillos, en proyectos reales deberíamos plantearnos emplear otra opción como un Raspberry Pi.

En los proyectos caseros se emplean tarjetas SD y micro SD, principalmente, en proyectos de tipo datalogger, es decir, para mantener el registro de las mediciones de un sensor. También pueden ser empleadas, por ejemplo, para llevar movimientos o rutas de robots precalculados.

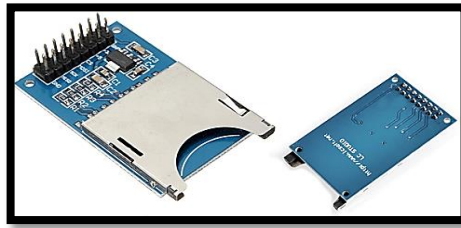


Figura 17. Modulo lector de memoria Micro SD, modelo MOD-SD-01

Fuente: <https://www.luisllamas.es/tarjeta-micro-sd-arduino/>

Dado que los lectores de tarjetas micro SD son economicos, modernos, y compactos, normalmente preferiremos los lectores micro SD a los SD. El único motivo real para comprar un lector SD frente a un micro SD es aprovechar alguna tarjeta que tengamos disponible.

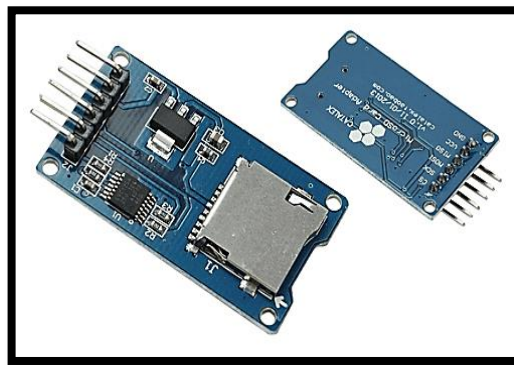


Figura 18. Modulo lector de memoria Micro SD, modelo MOD-MSDC

Fuente: <https://www.luisllamas.es/tarjeta-micro-sd-arduino/>

Respecto a la tarjeta, es sabido que han tenido un gran descenso en sus precios. En la actualidad, una tarjeta SD o micro SD de 1GB tiene un precio muy reducido, y es capacidad más que suficiente para la mayoría de proyectos. Incluso es frecuente que podamos reciclar antiguas tarjetas que hemos sustituido por otras de mayor capacidad.



Figura 19. Memoria Micro SD

Fuente: <https://www.luisllamas.es/tarjeta-micro-sd-arduino/>

2.1.7.12.1. FUNCIONAMIENTO

La conexión es sencilla y similar tanto para lectores SD como Micro SD. Simplemente alimentamos el módulo desde Arduino mediante 5V y Gnd. Por otro lado, conectamos los pines del bus SPI a los correspondientes de Arduino. La conexión del lector SD sería la siguiente:

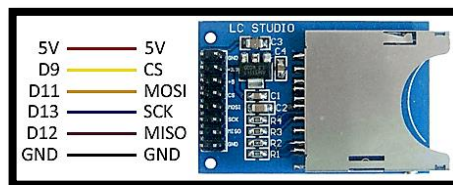


Figura 20. Conexiones del modelo MOD-SD-01

Fuente: <https://www.luisllamas.es/tarjeta-micro-sd-arduino/>

Similar a la de un lector micro SD, que sería la siguiente.

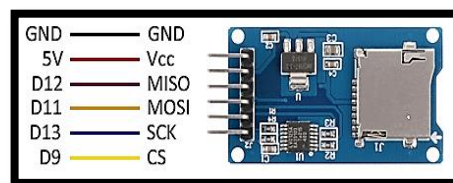


Figura 21. Conexiones del modelo MOD-MSDC

Fuente: <https://www.luisllamas.es/tarjeta-micro-sd-arduino/>

En ambos casos la conexión, vista desde el lado de Arduino, es la misma, y quedaría así.

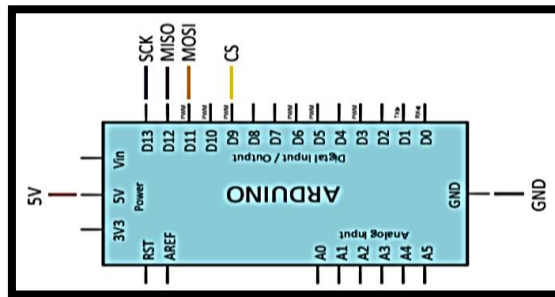


Figura 22. Conexiones a la placa Arduino.

Fuente: <https://www.luisllamas.es/tarjeta-micro-sd-arduino/>

2.2. DESARROLLO PRACTICO / EXPERIMENTAL

El proyecto que se realizo es un prototipo que ayudara a personas no videntes. Tiene la operación de que ni bien se prende el Arduino nos va a generar un audio de voz, que nos dirá con una voz a la distancia determinada por el sensor ultrasónico.

2.2.1. DIAGRAMA DE BLOQUES

El siguiente diagrama de bloques muestra los pasos para el funcionamiento del prototipo de lentes electrónicos a desarrollar.



Figura 23. Diagrama de bloques, funcionamiento del prototipo.

Fuente: Propia

2.2.2. COMPONENTES QUE SE UTILIZO PARA EL PROYECTO

2.2.2.1. ARDUINO UNO

En esta placa subiremos el código necesario para el control de todo el proyecto. Además se va a encargar de alimentar todo el proyecto con 5 voltios.



Figura 24. Placa Arduino Uno.

Fuente: https://www.researchgate.net/figure/Arduino-UNO-Las-principales-caracteristicas-de-esta-tarjeta-es-que-cuenta-con-seis_fig3_305477991

2.2.2.2. CAPACITOR ELECTROLITICO

Va conectado entre el pin 9 del arduino y el parlante, tiene polaridad, así que debemos tener en cuenta a la hora del montaje. El valor recomendable está entre 10 a 100 uF x 16 voltios.

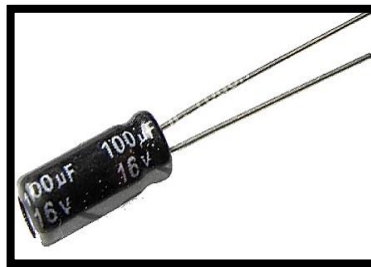


Figura 25. Capacitor electrolítico 100 uF x 16V.

Fuente: <https://www.electronicaplugandplay.com/componentes-pasivos/product/282-condensador-electrolitico-100uf-16v>

2.2.2.3. SENSOR ULTRASONICO

El modelo es un HC-SR04, un módulo muy común en proyectos caseros.

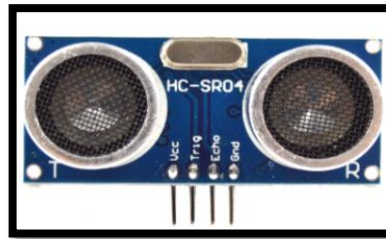


Figura 26. Sensor Ultrasónico, modelo HC-SR04

Fuente: <https://www.indiamart.com/proddetail/hc-sr04-ultrasonic-module-19296814948.html>

2.2.2.4. MEMORIA MICRO SD

Aquí guardaremos los archivos de audio en formato wav después de la conversión de los archivos, para adecuarla a nuestro proyecto.



Figura 27. Memoria Micro SD

Fuente: <https://defconshop.com/almacenamientotarjetas-de-memoria/7066-micro-sd-samsung-8gb-class-4-adaptador-sd-8806085581067.html>

2.2.2.5. PLAQUETA EXPERIMENTAL

Este tipo de placa es ideal para hacer proyectos nuevos, donde podemos ensayar un circuito antes de llevarlo a un circuito impreso de mucho mejor calidad.



Figura 28. Plaqueta experimental perforada

Fuente: <https://listado.mercadolibre.com.ar/plaqueta-experimental#!messageGeolocation>

2.2.2.6. MODULO LECTOR MICRO SD

Este módulo nos permite hacer una interface escribir o leer archivos, en el proyecto desarrollado, hace de interface de lectura de los archivos de audio, con el arduino y la tarjeta micro sd.

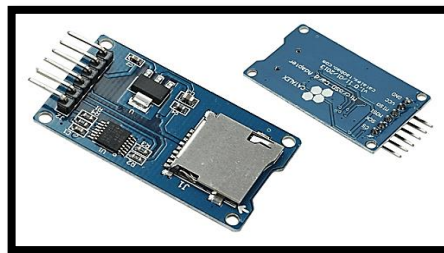


Figura 29. Modulo Micro SD

Fuente: <https://www.ecrobotics.com.bo/producto/modulo-micro-sd-tf-spi-sdio/>

2.2.3. DESARROLLO DEL PROTOTIPO

Todo el circuito funciona con 5 voltios, para obtener esta tensión conectamos el arduino a un cable USB. El módulo HC-SR04, es un sensor ultrasónico, que genera un sonido ultrasónico durante un tiempo breve (alrededor de 10uS) aplicando un pulso de ese tiempo en el pin «trigger».

Este sonido ultrasónico rebota contra un objeto y es recibido por el pin «eco», el tiempo de demora, entre esa transmisión y recepción nos va a determinar la distancia a la cual se encuentra un objeto como puede ser una pared. El pin trigger va conectado al pin número 8 del arduino, mientras que el pin eco va al pin 7.

El modo de operación es el modo voz, donde ni bien se prenda la placa Arduino nos va a reproducir un audio con la distancia a la cual está el objeto.

Para poder leer los archivos de audio necesitaremos un módulo lector de tarjeta micro SD, que nos permite leer los archivos almacenados en la tarjeta micro SD, estos archivos eASONICO

```
long duracion;
```

```
long distancia;
```

```
int Trigger = 8; // TRIG AL PIN 8
```

```
int eco = 7; // ECHO AL PIN 7
```

```
int conteo = 0;
```

```
int cent = 0;
```

```
void setup()
```

```
{
```

```
Serial.begin (9600);
```

```
// SdPlay.seSDCSPin(10); // Habilitar SÓLO si su tarjeta SD no está en el pin número
```



```

    SPlay.int(SSDA_MODE_HALFRATE      |      SSDA_MODE_MONO      |
SSDA_MODE_AUTOWORKER);

    pinMode(9, OUTPUT); // AUDIO EN PIN 9 (COMO SALIDA)

    pinMode(Trigger, OUTPUT); // TRIG COMO SALIDA

    pinMode(eco, INPUT); // ECHO COMO ENTRADA

    delay (500); // TIEMPO DE 5 SEG.

}

void loop(void)

{

    digitalWrite(Trigger, LOW);

    delayMicrosconds(4);

    digitalWrite(Trigger, HIGH);// Con el pin (Trigger) enviaremos una señal de
activación al sensor, lo activaremos durante 10 microsegundos y lo desactivaremos

    delayMicrosconds(10);

    digitalWrite(Trigger, LOW);

    duracion = pulseIn(eco, HIGH);

    cent = duracion/58.2; // la distancia en centimetros basado en la velocidad del
sonido.

```

```
distancia = duracion * 10 / 292 / 2;

delay(60);

Serial.println(distancia);//Imprime ese valor en el puerto serial

if(distancia==60){

  SPlay.setFile("60.wav");//con setFile vamos a cargar el archivo que queremos
reproducir

  SPlay.play();// play reproducimos

  delay(500);

}

if(distancia==59){

  SPlay.setFile("59.wav");//con setFile vamos a cargar el archivo que queremos
reproducir

  SPlay.play();// play reproducimos

  delay(500);

}

if(distancia==58){

  SPlay.setFile("58.wav");//con setFile vamos a cargar el archivo que queremos
reproducir
```

```
SdPlay.play(); // play reproducimos

delay(500);

}

if(distancia==57){

SPlay.setFile("57.wav");//con setFile vamos a cargar el archivo que queremos
reproducir

SPlay.play(); // play reproducimos

delay(500);

}

if(distancia==56){

SdPlay.setFile("56.wav");//con setFile vamos a cargar el archivo que queremos
reproducir

SdPlay.play(); // play reproducimos

delay(500);

}

if(distancia==55){

SdPlay.setFile("55.wav");//con setFile vamos a cargar el archivo que queremos
reproducir
```

```
SdPlay.play(); // play reproducimos

delay(500);

}

if(distancia==54){

SdPlay.setFile("54.wav");//con setFile vamos a cargar el archivo que queremos
reproducir

SdPlay.play(); // play reproducimos

delay(500);

}

if(distancia==53){

SdPlay.setFile("53.wav");//con setFile vamos a cargar el archivo que queremos
reproducir

SdPlay.play(); // play reproducimos

delay(500);

}

if(distancia==52){

SdPlay.setFile("52.wav");//con setFile vamos a cargar el archivo que queremos
reproducir
```

```
SPlay.play(); // play reproducimos

delay(500);

}

if(distancia==51){

SPlay.setFile("51.wav");//con setFile vamos a cargar el archivo que queremos
reproducir

SdPlay.play(); // play reproducimos

delay(500);

}

if(distancia==50){

SdPlay.setFile("50.wav");//con setFile vamos a cargar el archivo que queremos
reproducir

SPlay.play(); // play reproducimos

delay(500);

}

if(distancia==49){

SdPlay.setFile("49.wav");//con setFile vamos a cargar el archivo que queremos
reproducir
```

```
SdPlay.play(); // play reproducimos

delay(500);

}

if(distancia==48){

SdPlay.setFile("48.wav");//con setFile vamos a cargar el archivo que queremos
reproducir

SdPlay.play(); // play reproducimos

delay(500);

}

if(distancia==47){

SdPlay.setFile("47.wav");//con setFile vamos a cargar el archivo que queremos
reproducir

SdPlay.play(); // play reproducimos

delay(500);

}

if(distancia==46){

SdPlay.setFile("46.wav");//con setFile vamos a cargar el archivo que queremos
reproducir
```

```
SPlay.play(); // play reproducimos

delay(500);

}

if(distancia==45){

SPlay.setFile("45.wav");//con setFile vamos a cargar el archivo que queremos
reproducir

SdPlay.play(); // play reproducimos

delay(500);

}

if(distancia==44){

SdPlay.setFile("44.wav");//con setFile vamos a cargar el archivo que queremos
reproducir

SdPlay.play(); // play reproducimos

delay(500);

}

if(distancia==43){

SPlay.setFile("43.wav");//con setFile vamos a cargar el archivo que queremos
reproducir
```

```
SdPlay.play(); // play reproducimos

delay(500);

}

if(distancia==42){

SdPlay.setFile("42.wav");//con setFile vamos a cargar el archivo que queremos
reproducir

SPlay.play(); // play reproducimos

delay(500);

}

if(distancia==41){

SdPlay.setFile("41.wav");//con setFile vamos a cargar el archivo que queremos
reproducir

SdPlay.play(); // play reproducimos

delay(500);

}

if(distancia==40){

SdPlay.setFile("40.wav");//con setFile vamos a cargar el archivo que queremos
reproducir
```



```
SdPlay.play(); // play reproducimos

delay(500);

}

if(distancia==39){

SdPlay.setFile("39.wav");//con setFile vamos a cargar el archivo que queremos
reproducir

SdPlay.play(); // play reproducimos

delay(500);

}

if(distancia==38){

SdPlay.setFile("38.wav");//con setFile vamos a cargar el archivo que queremos
reproducir

SdPlay.play(); // play reproducimos

delay(500);

}

if(distancia==37){

SdPlay.setFile("37.wav");//con setFile vamos a cargar el archivo que queremos
reproducir
```

```
SdPlay.play(); // play reproducimos

delay(500);

}

if(distancia==36){

SdPlay.setFile("36.wav");//con setFile vamos a cargar el archivo que queremos
reproducir

SdPlay.play(); // play reproducimos

delay(500);

}

if(distancia==35){

SdPlay.setFile("35.wav");//con setFile vamos a cargar el archivo que queremos
reproducir

SdPlay.play(); // play reproducimos

delay(500);

}

if(distancia==34){

SdPlay.setFile("34.wav");//con setFile vamos a cargar el archivo que queremos
reproducir
```

```
SdPlay.play(); // play reproducimos

delay(500);

}

if(distancia==33){

SdPlay.setFile("33.wav");//con setFile vamos a cargar el archivo que queremos
reproducir

SdPlay.play(); // play reproducimos

delay(500);

}

if(distancia==32){

SdPlay.setFile("32.wav");//con setFile vamos a cargar el archivo que queremos
reproducir

SdPlay.play(); // play reproducimos

delay(500);

}

if(distancia==31){

SdPlay.setFile("31.wav");//con setFile vamos a cargar el archivo que queremos
reproducir
```

```
SdPlay.play(); // play reproducimos

delay(500);

}

if(distancia==30){

SdPlay.setFile("30.wav");//con setFile vamos a cargar el archivo que queremos
reproducir

SdPlay.play(); // play reproducimos

delay(500);

}

if(distancia==29){

SdPlay.setFile("29.wav");//con setFile vamos a cargar el archivo que queremos
reproducir

SdPlay.play(); // play reproducimos

delay(500);

}

if(distancia==28){

SdPlay.setFile("28.wav");//con setFile vamos a cargar el archivo que queremos
reproducir
```

```
SdPlay.play(); // play reproducimos

delay(500);

}

if(distancia==27){

SdPlay.setFile("27.wav");//con setFile vamos a cargar el archivo que queremos
reproducir

SdPlay.play(); // play reproducimos

delay(500);

}

if(distancia==26){

SdPlay.setFile("26.wav");//con setFile vamos a cargar el archivo que queremos
reproducir

SdPlay.play(); // play reproducimos

delay(500);

}

if(distancia==25){

SdPlay.setFile("25.wav");//con setFile vamos a cargar el archivo que queremos
reproducir
```

```
SdPlay.play(); // play reproducimos

delay(500);

}

if(distancia==24){

SdPlay.setFile("24.wav");//con setFile vamos a cargar el archivo que queremos
reproducir

SdPlay.play(); // play reproducimos

delay(500);

}

if(distancia==23){

SdPlay.setFile("23.wav");//con setFile vamos a cargar el archivo que queremos
reproducir

SdPlay.play(); // play reproducimos

delay(500);

}

if(distancia==22){

SdPlay.setFile("22.wav");//con setFile vamos a cargar el archivo que queremos
reproducir
```

```
SdPlay.play(); // play reproducimos

delay(500);

}

if(distancia==21){

SdPlay.setFile("21.wav");//con setFile vamos a cargar el archivo que queremos
reproducir

SdPlay.play(); // play reproducimos

delay(500);

}

if(distancia==20){

SdPlay.setFile("20.wav");//con setFile vamos a cargar el archivo que queremos
reproducir

SdPlay.play(); // play reproducimos

delay(500);

}

if(distancia==19){

SdPlay.setFile("19.wav");//con setFile vamos a cargar el archivo que queremos
reproducir
```

```
SdPlay.play(); // play reproducimos

delay(700);

}

if(distancia==18){

SdPlay.setFile("18.wav");//con setFile vamos a cargar el archivo que queremos
reproducir

SdPlay.play(); // play reproducimos

delay(700);

}

if(distancia==17){

SdPlay.setFile("17.wav");//con setFile vamos a cargar el archivo que queremos
reproducir

SdPlay.play(); // play reproducimos

delay(800);

}

if(distancia==16){

SdPlay.setFile("16.wav");//con setFile vamos a cargar el archivo que queremos
reproducir
```



```
SdPlay.play(); // play reproducimos

delay(700);

}

if(distancia==15){

SdPlay.setFile("15.wav");//con setFile vamos a cargar el archivo que queremos
reproducir

SdPlay.play(); // play reproducimos

delay(500);

}

if(distancia==14){

SdPlay.setFile("14.wav");//con setFile vamos a cargar el archivo que queremos
reproducir

SdPlay.play(); // play reproducimos

delay(500);

}

if(distancia==13){

SdPlay.setFile("13.wav");//con setFile vamos a cargar el archivo que queremos
reproducir
```

```
SdPlay.play(); // play reproducimos

delay(500);

}

if(distancia==12){

SdPlay.setFile("12.wav");//con setFile vamos a cargar el archivo que queremos
reproducir

SdPlay.play(); // play reproducimos

delay(500);

}

if(distancia==11){

SdPlay.setFile("11.wav");//con setFile vamos a cargar el archivo que queremos
reproducir

SdPlay.play(); // play reproducimos

delay(500);

}

if(distancia==10){

SdPlay.setFile("10.wav");//con setFile vamos a cargar el archivo que queremos
reproducir
```

```
SdPlay.play(); // play reproducimos

delay(500);

}

if(distancia==9){

SdPlay.setFile("9.wav");//con setFile vamos a cargar el archivo que queremos
reproducir

SdPlay.play(); // play reproducimos

delay(500);

}

if(distancia==8){

SdPlay.setFile("8.wav");//con setFile vamos a cargar el archivo que queremos
reproducir

SdPlay.play(); // play reproducimos

delay(500);

}

if(distancia==7){

SdPlay.setFile("7.wav");//con setFile vamos a cargar el archivo que queremos
reproducir
```

```

SdPlay.play(); // play reproducimos

delay(500);

}

if(distancia==6){

SdPlay.setFile("6.wav");//con setFile vamos a cargar el archivo que queremos
reproducir

SdPlay.play(); // play reproducimos

delay(500);

}

if(distancia==5){

SdPlay.setFile("5.wav");//con setFile vamos a cargar el archivo que queremos
reproducir

SdPlay.play(); // play reproducimos

delay(500);

}

if(distancia==4){

SdPlay.setFile("4.wav");//con setFile vamos a cargar el archivo que queremos
reproducir

```

```
SdPlay.play(); // play reproducimos

delay(500);

}

if(distancia==3){

SdPlay.setFile("3.wav");//con setFile vamos a cargar el archivo que queremos
reproducir

SdPlay.play(); // play reproducimos

delay(500);

}

if(distancia==2){

SdPlay.setFile("2.wav");//con setFile vamos a cargar el archivo que queremos
reproducir

SdPlay.play(); // play reproducimos

delay(500);

}

if(distancia==1){

SdPlay.setFile("1.wav");//con setFile vamos a cargar el archivo que queremos
reproducir
```

```
SdPlay.play(); // play reproducimos
```

```
delay(500);
```

```
}
```

```
}
```

2.2.4. CIRCUITO FINAL

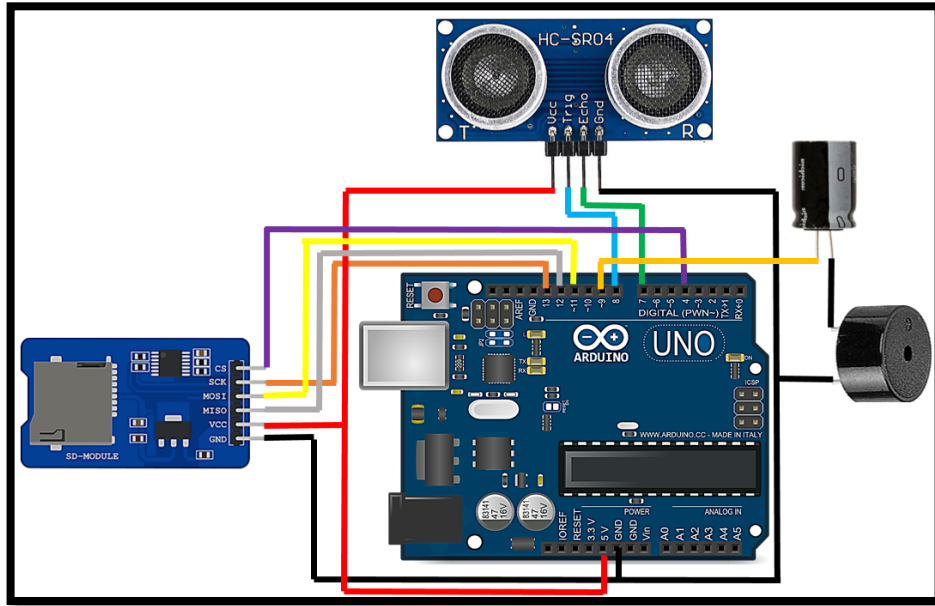


Figura 30. Conexiones del ultrasónico, módulo Micro SD y Buzzer a la placa Arduino

Fuente: Propia

CAPITULO III

3. ANALISIS DE COSTOS

3.1. COSTOS FIJOS

El proyecto de grado técnico cuenta con los siguientes costos fijos vale la pena hacer notar que los costos de estos componentes no tienen un efecto directo en cuanto al costo final del proyecto, la cual se utilizó un VR BOX como prototipo de lentes.

Tabla 4. Costos fijos del proyecto

DESCRIPCION	COSTOS Bs.
VR BOX	60
TOTAL	60

3.2. COSTOS VARIABLES

Debido que los componentes electrónicos sufren constantes variaciones en su precio final; el costo total del proyecto presentado puede llegar a tener ligeros cambios con el pasar de los días. A continuación, se detalla el costo final de cada uno de los componentes al momento de ser adquiridos tanto de proveedores locales como de proveedores en el interior del país.

Tabla 5. Costos variables del proyecto

DESCRIPCION	COSTO Bs.
Arduino Uno	55
Sensor Ultrasonico HC-SR04	15
Modulo Lector Micro SD MOD-MSDC	14
Memoria Micro SD 8Gb	25
Plaqueta experimental perforada	5

Capacitor electrolítico 100 uf x 16V	0.50
Buzzer	5
TOTAL	119.50

CAPITULO IV

4. CONCLUSIONES Y RECOMENDACIONES

4.1. CONCLUSIONES

- Este proyecto de prototipo de gafas con sensores es eficiente al momento de ayudar a guiarse a las personas no videntes porque en la parte superior de su cuerpo no tienen como hacerlo y están más propensos a sufrir golpes.
- Se logró acoplar un sistema de envío y recepción de datos mediante módulos de comunicación.
- Se consiguió interconectar y manejar los diferentes entornos que se usaron para la realización del proyecto, tanto la parte del Módulo Micro SD tuvo resultados positivos porque mediante las voces de alerta que reproduce sin duda advierte de un peligro a la persona con discapacidad visual.
- El programa Arduino actualmente son muy utilizados por estudiantes en el país, ya sea de forma académica o en lo profesional, por tanto, se concluye que este proyecto refuerza el entusiasmo de trabajar con nuevas tecnologías para brindar ayuda al ámbito social.

4.2. RECOMENDACIONES

- Se recomienda tener cuidado con el uso del prototipo porque contiene piezas electrónicas que pueden averiarse si existe una inadecuada manipulación.
- Es importante mantener el prototipo en ambientes secos donde no esté expuesto a mojarse.
- Al momento de colocarse el prototipo se recomienda la ayuda de otra persona o familiar para asegurar que la caja estuche se encuentre bien sostenido, de igual manera con las gafas que estén bien situadas.
- Es importante recargar la fuente de alimentación cada vez que se termina de utilizar el prototipo, así nos aseguramos que tiene la suficiente carga para la próxima vez que vaya a ser usado.
- Se recomienda realizar con regularidad un mantenimiento preventivo al prototipo de la manera más cuidadosa posible para evitar daños físicos en el sistema.

BIBLIOGRAFIA

- ❖ Bermúdez Roldán, Á. C., Calero de Lara, A., Camacho, F. D., Padilla Camacho, M. (2012). *Gafas para invidentes*.
http://www.fpbollullos.es/pdf/GAFAS_PARA_INVIDENTES.pdf
- ❖ Peralta Mosquera, M. A., Urmendiz Terreros, J. V. (2014). *Sistema De Asistencia Y Guía Para Personas Invidentes*
<https://red.uao.edu.co/bitstream/10614/6868/1/T05005.pdf>
- ❖ Guillen Peñarreta, J. P., Vizhñay Aguilar, C. F. (2016). *Gafas especiales para detección de obstáculos con sistema de ubicación en caso de emergencia y ayuda de reconocimiento de billetes para personas con discapacidad visual*
<https://dspace.ups.edu.ec/bitstream/123456789/12295/1/UPS-CT006438.pdf>
- ❖ Alcantara Rivas, M. (2016). *Prototipo de asistencia para la movilidad de ciegos a través de ecolocacion*
<https://tesis.ipn.mx/bitstream/handle/123456789/21628/tesis.pdf>
- ❖ Yanchatuña Aguayo, L. A. (2016). *Vision artificial por alertas de voz y movimiento para personas con discapacidad visual en la biblioteca de no videntes de la Universidad Técnica de Ambato*
https://bitstream/123456789/20348/1/Tesis_t1108ec.pdf
- ❖ Paton Valentín, P. (2018). *Sistema de detección de obstáculos aéreos para invidentes DIY*
https://rua.ua.es/dspace/bitstream/10045/77550/1/Sistema_de_deteccion_de_obstaculos_aereos_para_invident_PATON_VALENTIN_PEDRO.pdf
- ❖ Potosi Anrango, D. E. (2019). *Diseño e implementación de un dispositivo electrónico medidor de distancias y detector de colores para ayudar a las personas no videntes mediante comando de voz*
<https://bibdigital.epn.edu.ec/bitstream/15000/20177/1/CD%209632.pdf>

- ❖ Ontiveros Paredes, S. D., Rojas Balbuena, D., Martínez Paredes, J. (2014). *Diseño y construcción de un bastón blanco electrónico para personas invidentes*
<https://www.redalyc.org/pdf/614/61448035002.pdf>
- ❖ Llamas, L. (2016). *Leer y escribir en una tarjeta SD o Micro SD con Arduino*
<https://www.luisllamas.es/tarjeta-micro-sd-arduino/>
- ❖ Todo lo que necesitas saber sobre discapacidad visual.
<https://www.incluyeme.com/todo-lo-que-necesitas-saber-sobre-discapacidad-visual/>
- ❖ INSTITUTO BOLIVIANO DE LA CEGUERA (IBC).
<http://www.ibc.gob.bo/estadisticas.php>
- ❖ Proyecto Arduino (Sensor de ultrasonidos).
<https://proyectoarduino.com/sensor-de-ultrasonidos-medir-distancia-con-arduino/>
- ❖ Arduino Uno, partes, componentes, para que sirve y donde comprar.
<https://descubrearduino.com/arduino-uno/>

ANEXOS

Componentes que se utilizara en el proyecto.

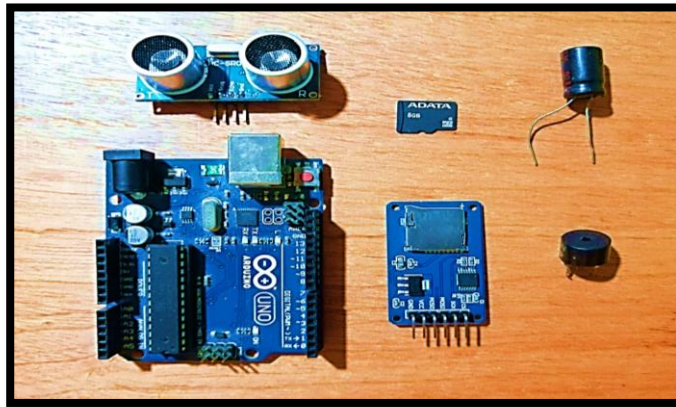


Figura: Arduino Uno, Sensor Ultrasónico, Modulo Micro SD, Memoria Micro SD, Capacitor Electrolitico, Buzzer

Cables, batería, plaqueta experimental perforada y conector de batería que se utilizara para conectar al Arduino Uno y alimentar con la batería de 9V.

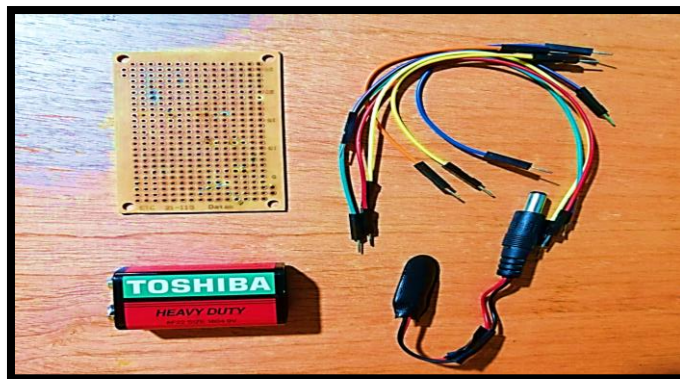


Figura: Plaqueta experimental perforada, Bateria 9V, Conector de Bateria 9V a Jack DC, Cables

Conectamos el Modulo Micro SD al Arduino Uno con los cables en los pines CS-PIN4, SCK-PIN13, MOSI-PIN11, MISO-PIN12

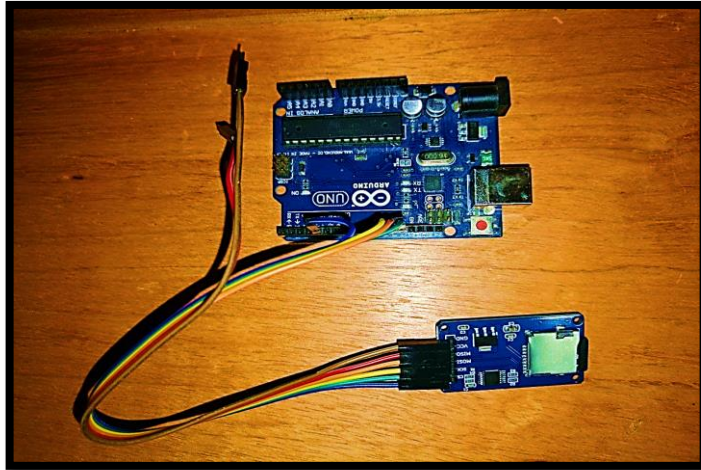


Figura: Arduino conectado al Modulo Micro SD

Seguidamente conectamos con los cables el Sensor Ultrasónico en los pines TRIG-PIN8, ECHO-PIN7.

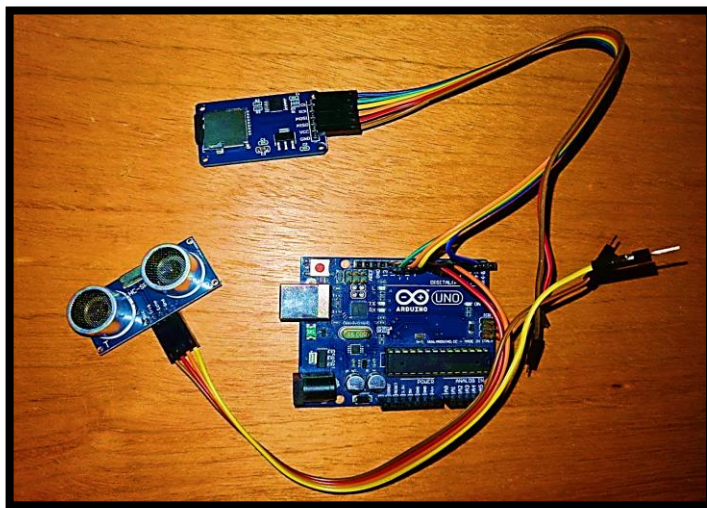


Figura: Arduino conectado al Micro SD y al Sensor Ultrasónico.

Luego conectamos el Capacitor Electrolitico con el Buzzer para transmitir el audio en el PIN9.

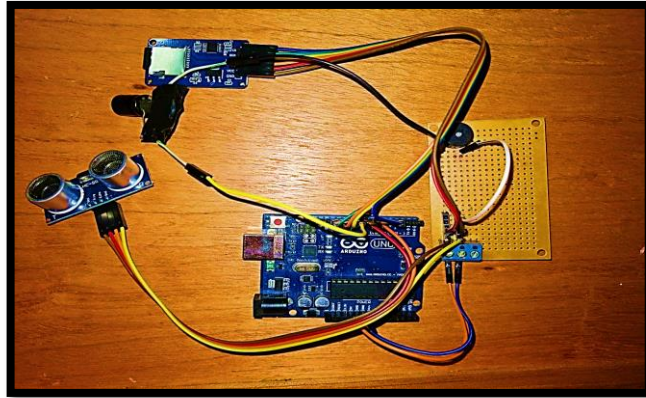


Figura: Arduino conectado al Micro SD, Sensor Ultrasónico y al Capacitor electrolítico 100 uF x 16V.

A continuación, alimentaremos con una Bateria de 9V a la placa Arduino Uno para que funcione el Modulo Micro SD, Sensor Ultrasonico y el Buzzer.

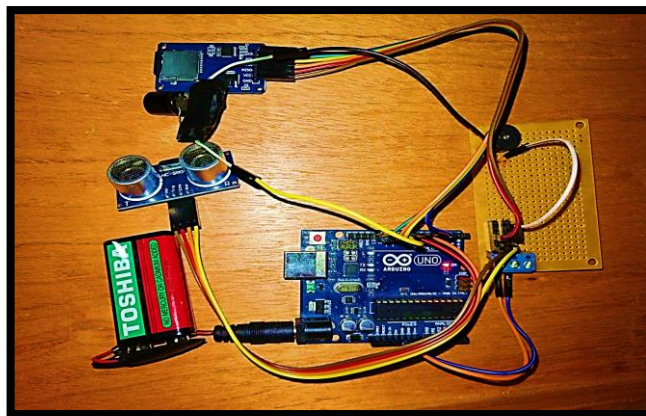


Figura: Arduino conectado al Micro SD, Sensor Ultrasónico y al Capacitor electrolítico 100 uF x 16V, alimentado con una Bateria de 9V.

Finalmente introducimos el Arduino Uno y todos los componentes en el prototipo.



Figura: Componentes dentro del Prototipo

Podemos observar que prototipo está en perfecto funcionamiento.



Figura: Prototipo finalizado visto de frente.

Haciendo pruebas del prototipo para que no tenga alguna falla al momento de utilizar.



Figura: Prototipo visto de costado.

Prototipo listo para que utilice una persona invidente.

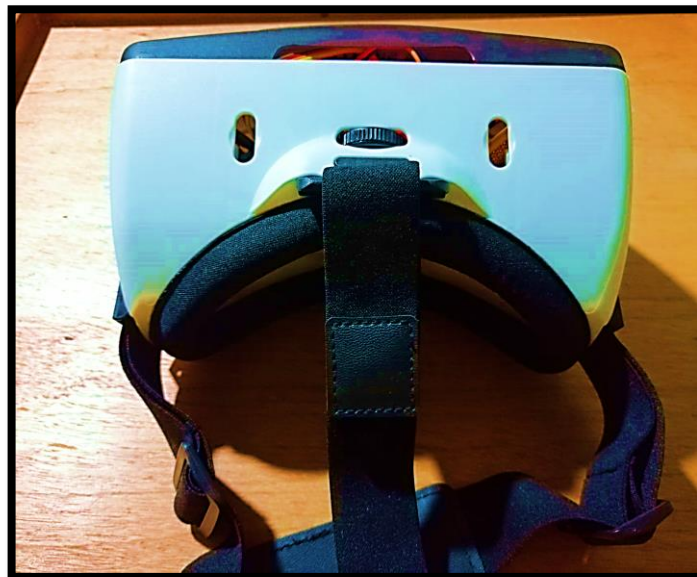


Figura: Prototipo visto de atrás.