



UNIVERSIDAD TECNOLÓGICA ISRAEL

TRABAJO DE TITULACIÓN EN OPCIÓN AL GRADO DE:

INGENIERO EN ELECTRÓNICA DIGITAL Y TELECOMUNICACIONES

TEMA: GAFAS Y BASTÓN INTELIGENTE PARA UNA PERSONA INVIDENTE

AUTOR: LENIN GIOVANNY PAUCAR ESPINOSA

TUTOR/ A: ING. RENE ERNESTO CORTIJO LEYVA Mg.

AÑO: 2018

CERTIFICACIÓN DEL TUTOR

UNIVERSIDAD TECNOLÓGICA ISRAEL

APROBACIÓN DEL TUTOR

En mi calidad de tutor del trabajo de titulación certifico:

Que el trabajo de titulación “**GAFAS Y BASTÓN INTELIGENTE PARA UNA PERSONA INVIDENTE**”, presentado por el Sr. Lenin Giovanny Paucar Espinosa, estudiante de la carrera de Electrónica Digital y Telecomunicaciones, reúne los requisitos y méritos suficientes para ser sometido a la evaluación del Tribunal de Grado, que se designe, para su correspondiente estudio y calificación.

Quito D.M Febrero del 2018

TUTOR

.....

Ing. Rene Ernesto Cortijo Leyva Mg.

AGRADECIMIENTOS

En primer lugar doy gracias a Dios, porque sin sus fuerzas no podía alcanzar lo que me propuse.

Agradezco también a toda mi familia, el cual me brindaron su apoyo incondicional. A mis padres, que siempre estuvieron a mi lado dándome su amor, corrigiendo mis fallas y siempre celebraron conmigo mis triunfos. A mi esposa por brindarme su amor durante todo este proceso y a mi hijo, por el tiempo que no he compartido y vivido junto con él.

A la Universidad Tecnológica Israel y a todos mis profesores por transmitir sus conocimientos valiosos para mi formación.

A todos mis amigos de la Universidad, quienes han compartido conmigo todos los sacrificios, logros y satisfacciones de la vida estudiantil.

Lenin Giovanni Paucar E.

DEDICATORIA

Dedico de manera especial la culminación de este proyecto a Dios por brindarme la vida, salud y fuerzas, considerando que en este lapso de tiempo he atravesado obstáculos.

A mi amado hijo Israel, que con su nacimiento me dio los ánimos para seguir adelante y así poderle brindar un mejor futuro.

A mi amada esposa Liz, que con su apoyo y amor incondicional ha sido pilar fundamental para la culminación de mi carrera, ya que llegó a mi vida cuando inicié mis estudios superiores.

A mis padres Lenin y Nancy, que con su amor me enseñaron virtudes y me animaron para que culmine con éxito mi carrera.

A mi hermana Patricia, que con su cariño y comprensión me ha ayudado a afrontar los retos.

A todos mis amigos que he encontrado en el camino y que con el tiempo hemos compartido buenos y malos momentos, gracias por formar parte de mi vida.

Lenin Geovanny Paucar E.

Tabla de contenido

RESUMEN	1
INTRODUCCIÓN	3
Antecedentes de la situación objeto de estudio	3
Planteamiento del problema	3
Formulación del problema	3
Justificación.....	4
Objetivos	4
Objetivo general:	4
Objetivos específicos:.....	4
Descripción de los capítulos.....	5
CAPÍTULO I	5
FUNDAMENTACIÓN TEÓRICA	5
1.1 Historia.....	5
1.2. Bastón	5
1.2.1. Tipos de bastones.....	5
1.3. Gafas	6
1.3.1. Armazón o Montura.....	6
1.3.2. Medidas de Monturas	6
1.4. La ceguera.....	7
1.4.1. Tipos de ceguera	7
1.4.2. Ayudas a la movilización de los no videntes.....	8
1.4.3. Gafas con GPS para ciegos.....	9
1.4.4. Traje de ayuda para invidentes	9
1.4.5. Una vara para ciegos.....	9
1.4.6. Robot para ciegos	10
1.4.7. Retina artificial	10
1.4.8. Estadísticas de la ceguera	11
1.5. Discapacidad	11
1.5.1. Clasificación de la discapacidad.....	12
1.6. Sensores	13
1.6.1. ¿Qué son los sensores?	13
1.6.2. Características generales de un sensor.....	14
a) Características Estáticas	14

b) Características Dinámicas	14
1.6.3. Clasificación de los sensores	15
1.7. Arduino	15
1.7.1. ¿Por qué Arduino?	16
1.8. Módulo RF	16
1.9. Tarjeta Micro SD	17
1.10. Módulo RTC	17
1.11. Módulo GPRS	17
1.12. El SIM900 GSM/GPRS Shield	18
1.13. Sistema global para las comunicaciones móviles (GSM)	18
1.14. Módulo detector de agua	19
CAPÍTULO II	20
PROPUESTA	20
Análisis de Requerimientos	20
2.1. Diseño	22
2.1.1. Arquitectura del Sistema prototipo	22
2.1.2. Diagrama general de bloques del sistema prototipo	23
2.1.3. Bloque del sistema de control	24
2.1.4. Diseño electrónico	24
2.1.5 Descripción de diseño de los componentes	25
2.1.5.1 Arduino Uno	25
2.1.5.2 Módulo Emisor y Receptor RF 433mhz.	27
2.1.5.3 Sensor de proximidad HC-SR04	28
2.1.5.4 Diagrama electrónico del sistema de comunicación GPRS	29
2.1.5.5 Módulo RTC DS3231	30
2.1.5.6 Módulo Reproductor Mp3	32
2.1.5.7 Sensor infrarrojo Sharp	33
2.1.5.8 Sistema de alimentación	35
2.1.5.9 Sensor de agua	35
2.1.6.0 Arduino Nano	36
CAPÍTULO III	37
IMPLEMENTACIÓN	37
3.1 Desarrollo	37

3.1.1	Diseño del circuito en Proteus (Tarjeta Madre)	37
3.1.1.1	Conexión al Arduino Uno.....	37
3.1.1.2	Conexión del Módulo RTC DS3231	37
3.1.1.3	Conexión del Transmisor FM.....	38
3.1.1.4	Conexión del Módulo RF	38
3.1.1.5	Configuración de la Hora	38
3.1.1.6	Conexión del Reproductor de voz	39
3.1.2	Diseño del circuito en Proteus (Bastón).....	39
3.1.2.1	Encendido del sistema	39
3.1.2.2	Conexión al Arduino Nano.....	40
3.1.2.3	Conexión del Transmisor RF.....	40
3.1.2.4	Conexión del sensor ultrasónico HC-SR04	41
3.1.2.5	Conexión del Sensor de Humedad.....	41
3.1.3	Diseño del circuito en Proteus (Gafas).....	42
3.1.3.1	Encendido del sistema	42
3.1.3.2	Conexión al Arduino Nano.....	42
3.1.3.3	Conexión del Transmisor RF.....	43
3.1.3.4	Conexión de los Sensores ultrasónicos HC-SR04.....	43
3.1.4	Diseño del circuito Impreso	43
3.2	Construcción	44
3.2.1	Construcción del circuito impreso e instalación de los componentes	44
3.2.2	Construcción de cables de extensión para conexión	46
3.2.3	Construcción de cajas para los circuitos de control	47
3.2.4	Utilización de un bastón.....	48
3.2.5	Utilización de una Gafas	49
3.3	Montaje	49
3.3.1	Colocación del circuito de control en la caja principal	49
3.3.2	Colocación del circuito de control en las Gafas	49
3.3.3	Colocación del circuito de control en el Bastón.....	50
3.4	Pruebas de funcionamiento	50
3.4.1	Monitoreo	50
3.4.2	Prueba de los componentes	50
3.4.3	Prueba de funcionamiento SIM 908 GPRS.....	51

3.5 Análisis de resultados	51
CONCLUSIONES	56
RECOMENDACIONES	57
CRONOGRAMA DE ACTIVIDADES	58
EVALUACION DE LA PROPUESTA	59
REFERENCIAS BIBLIOGRAFICAS	60
ANEXOS	63

LISTA DE FIGURAS

Figura 1. Bastón blanco	5
Figura 2. Gafas.....	6
Figura 3. Gafas de madera	6
Figura 4. Medidas de gafas	7
Figura 5. Falencia del sentido de vista.....	7
Figura 6. Gafas GPS para ciegos	9
Figura 7. Traje de ayuda para invidentes	9
Figura 8. Una vara para ciegos	10
Figura 9. Robots para ciegos.....	10
Figura 10. Retina artificial	11
Figura 11. Símbolo de personas con discapacidad física.....	12
Figura 12. Personas con discapacidad física.....	12
Figura 13. Persona con discapacidad sensorial.....	12
Figura 14. Persona con discapacidad sensorial.....	12
Figura 15. Persona con discapacidad psíquica.....	13
Figura 16. Símbolo de personas con discapacidad visual.....	13
Figura 17. Esquema general de un sensor.....	13
Figura 18. Esquema Arduino	16
Figura 19. Módulo Emisor y Receptor	17
Figura 20. Módulo RTC.....	17
Figura 21. Módulo GPS	18
Figura 22. Módulo Detector de Agua	19
Figura 23. Comunicación entre los dispositivos	20
Figura 24. Componentes en Gafas.	21
Figura 25. Componentes en el Bastón.	21
Figura 26. Subsistemas de Transmisión y Recepción.....	22
Figura 27. Diagrama de bloques general del sistema prototipo.....	23
Figura 28. Diagrama de bloques del sistema de control.	24
Figura 29. Diagrama electrónico del Prototipo.....	25
Figura 30. Arduino Uno.....	26
Figura 31. Módulo TX-RX 433 MHz.....	27
Figura 32. Diagrama del Arduino a los Módulos RF.	28
Figura 33. Diagrama del Arduino a los sensores ultrasónicos.....	28
Figura 34. Inicialización del programa con el Sensor HC-SR04.....	29
Figura 35. Diagrama de flujo lectura del Sensor ultrasónico.....	29
Figura 36. Esquema de conexión del módulo GPRS al Arduino Uno.....	30
Figura 37. Esquema de conexión del Arduino Uno y Módulo RTC DS3231.	31
Figura 38. Diagrama de bloques del Módulo RTC DS3231.....	31
Figura 39. Diagrama de flujo del módulo RTC DS3231.....	32
Figura 40. Esquema de conexión del Arduino Uno y Módulo Mp3.....	32
Figura 41. Diagrama de flujo del Módulo Mp3.....	33

Figura 42. Esquema de conexión del Arduino Nano y Sensor Sharp.....	33
Figura 43. Inicialización del programa con el Sensor Sharp.	34
Figura 44. Conexión y diagrama de bloques del Sensor Sharp.	34
Figura 45. Diagrama de bloques del sistema de alimentación.....	35
Figura 46. Esquema de conexión del Arduino Nano y el Sensor de agua.	35
Figura 47. Estructura del Arduino.	36
Figura 48. Conexión al Arduino Uno.	37
Figura 49. Conexión del Módulo RTC DS3231.	37
Figura 50. Conexión del Transmisor FM.....	38
Figura 51. Conexión del Módulo RX.	38
Figura 52. Configuración de la hora.	39
Figura 53. Reproductor de voz.....	39
Figura 54. Encendido del equipo.	40
Figura 55. Conexión al Arduino Nano.....	40
Figura 56. Conexión del Transmisor RF.....	41
Figura 57. Sensor HC-SR04.	41
Figura 58. Sensor de Humedad.	42
Figura 59. Encendido del equipo.	42
Figura 60. Arduino Nano.	42
Figura 61. Conexión RF.....	43
Figura 62. Conexión de Sensores HC-SR04.....	43
Figura 63. Diseño PCB circuito impreso.	44
Figura 64. Impresiones de las pistas del circuito de control.	44
Figura 65. Colocación de la impresión en la baquelita.....	45
Figura 66. Retiro de la impresión de las pistas.	45
Figura 67 Perforación de baquelita.	45
Figura 68. Soldada de elementos.	46
Figura 69. Switch de encendido para dispositivos.....	46
Figura 70. Cable ribbon de 4 hilos para conexión.	46
Figura 71. Regletas macho y hembra.....	46
Figura 72. Verificando el cable de punto a punto.....	47
Figura 73. Cables para sensores y elementos.....	47
Figura 74. Prueba de funcionamiento con el Lcd.	47
Figura 75. Madera.....	48
Figura 76. Realización de perforaciones de la caja.....	48
Figura 77. Caja terminada.....	48
Figura 78. Bastón modificado.....	49
Figura 79. Caja terminada.....	49
Figura 80. Caja principal.....	49
Figura 81. Caja en gafas.....	50
Figura 82. Caja en gafas.....	50
Figura 83. Distancia entre 70 cm a 120 cm	50
Figura 84. Monitoreo GPS por medio de SMS.....	51

LISTA DE TABLAS

Tabla 1: Características Técnicas Módulo Arduino Uno.....	26
Tabla 2: Módulo GPRS/GSM SIM900.....	29
Tabla 3: Pines especiales del Módulo Arduino.....	31
Tabla 4: Pines especiales del Módulo Mp3.....	33
Tabla 5: Pines especiales del Sensor Sharp.....	34
Tabla 6: Pines especiales del Sensor Sharp.....	36
Tabla 7: Especificación de pines del Arduino Nano.....	36
Tabla 8: Sistema de detección de obstáculos en las gafas.....	52
Tabla 9: Sistema de detección de obstáculos en el bastón (Sensor).....	52
Tabla 10: Sistema de detección de obstáculos en el bastón (Sensor Sharp).....	52
Tabla 11: Sistema de detección de obstáculos en el bastón (Sensor de Agua).....	53
Tabla 12: Sistema de control “Tarjeta Principal”.....	53
Tabla 13: Sistema de control “Hora”.....	54
Tabla 14: Sistema de control “Módulo GPRS SIM900”.....	54
Tabla 15: Actividades para presentación del proyecto.....	58
Tabla 16: Evaluación económica.....	59

RESUMEN

La presente tesis “**GAFAS Y BASTÓN INTELIGENTE PARA UNA PERSONA INVIDENTE**”, se basa en la integración de diferentes tecnologías estudiadas a lo largo de la carrera de Ingeniería Electrónica Digital y Telecomunicaciones, con la finalidad de ser un aporte a las personas que presentan discapacidad visual.

Este proyecto tiene como objetivo principal elaborar unas gafas y un bastón electrónico que de alerta a la persona no vidente, indicando la distancia a la que se encuentra un obstáculo mediante sensores ultrasónicos, de tal manera que éste pueda evadirlo al momento de moverse.

Se utilizaron herramientas como tarjetas de control, los módulos de comunicación inalámbrica, sensores, software como son: PROTEUS, ARDUINO. Se prosiguió con el montaje de las tarjetas de control para finalmente realizar pruebas y verificar el correcto funcionamiento del proyecto.

Con los resultados obtenidos se presenta un prototipo de unas gafas que se comunican vía bluetooth a unos audífonos, capaz de advertir tanto en frente como a los lados y un bastón que indica la presencia de un obstáculo en la parte frontal, de igual forma un sensor infrarrojo que ayuda a detectar los desniveles, un sensor de humedad que indica si hay presencia de agua y por último un botón que reproduce la hora y un pulsador de emergencia cuando la persona requiera de ayuda.

Palabras clave: Módulo RF, localizador GPRS, Arduino, Sensor Ultrasónico, Módulo Mp3.

ABSTRACT

The present thesis "GLASSES AND INTELLIGENT STAFF FOR AN INIDENT PERSON" is based on the integration of different technologies studied along the career of Digital Electronics Engineering and Telecommunications, with the purpose of being a contribution to the people who have visual disability.

The main objective of this project is to create glasses and an electronic baton that alerts the blind person, indicating the distance to which an obstacle is located by means of ultrasonic sensors, in such a way that it can evade it when it is mobilized.

We used tools such as control cards, wireless communication modules, sensors, software such as: PROTEUS, ARDUINO. The assembly of the control cards was continued to finally carry out tests and verify the correct operation of the project.

With the results obtained, a prototype of a pair of glasses communicating via bluetooth to hearing aids, able to warn both in front and on the sides and a stick that indicates the presence of an obstacle in the front, is presented infrared that helps detect unevenness a humidity sensor that indicates if there is water and finally a button that reproduces the time and an emergency button when the person needs help.

Keywords: RF Module, GPRS Locator, Arduino, Ultrasonic Sensor, Mp3 Module.

INTRODUCCIÓN

Antecedentes de la situación objeto de estudio

En la actualidad se encuentran sistemas electrónicos de ayuda a personas no videntes donde se transmite cierta información de los obstáculos que lo rodean, el dispositivo electrónico reconoce si el obstáculo está cerca o lejos para que la persona invidente ejecute acciones como por ejemplo detenerse o continuar caminando.

La discapacidad visual casi siempre remite nuestra mente a pensar en alguien que no ve y a sentir una consideración muy especial hacia él. Sin embargo, es muy importante aprender a reconocer que existe otro rango de discapacidad visual en el cual se encuentran ubicados niños, jóvenes y adultos con baja visión que les impide beneficiarse de la estimulación normal (visual en un 85%) y que las implicaciones de su discapacidad, sea total o no, pueden volverse relativas implementando para ellos una propuesta individual (Cabrera, 2008).

Planteamiento del problema

Desde el punto de vista de la discapacidad de la visión, se puede distinguir diferentes grados de ceguera que van desde la falta total de este sentido, por lo que es imposible adquirir información a través de la vista con un grado de disminución visual.

Según los datos estadísticos de la OMS (Organización Mundial de la Salud) en el mundo existen 253 millones con discapacidad visual total: 36 millones con ceguera y 217 millones con discapacidad visual moderada a grave.

Así mismo según la OMS en Ecuador existen 38.952 personas con discapacidad por deficiencias visuales, de los cuales tan solo el 10.1% usan ayudas técnicas (OMS, 2013).

Por otra parte la tecnología puede contribuir con un paliar esta situación pues permite que las personas con este problema se desempeñen con relativa facilidad permitiendo una mejor inclusión en la sociedad.

No obstante un elevado número de personas no tienen los recursos económicos como para aprovechar los beneficios de la tecnología para mejorar sus condiciones de vida, Ecuador no escapa a esta situación, la OMS asegura que de las personas invidentes solo el 4% puede acceder al beneficio tecnológico, ya sea un recurso propio o del estado, surge la necesidad de implementará un dispositivo de ayuda para la persona invidente. Este sistema brinda mayor seguridad y percepción de objetos.

Formulación del problema

¿Cómo diseñar y construir unas gafas con un bastón inteligente para invidentes que le proporcione una fácil movilidad en las calles utilizando dispositivos de ultrasonido?

¿Cuáles serían las ventajas y desventajas que tendrían los invidentes al usar las gafas y un bastón inteligente?

¿Cuáles serían los aportes que se harían a la ciencia desde la ingeniería de sistemas con el diseño de unas gafas y un bastón inteligente?

Justificación

Existen numerosos implementos que utilizan estas personas para desplazarse de un lado a otro, los cuales aunque han sido tradicionales pueden ser remplazados por dispositivos electrónicos que permita dar aviso o conocimiento a tiempo al usuario, sobre la presencia de algún obstáculo en el camino. Por ello, es muy importante construir un dispositivo para la detección de obstáculos de bajo costo y al alcance de cualquier persona con discapacidad visual.

Actualmente se puede desarrollar este tipo de dispositivos, por cuanto en el mercado se encuentran distintos elementos electrónicos que se adaptan con facilidad y pueden permitir el desarrollo del proyecto. Este dispositivo reemplaza parcialmente a los instrumentos de ayuda común que utilizan las personas con discapacidad visual.

Objetivos

Objetivo general:

Desarrollar unas gafas y un bastón electrónico para una persona invidente

Objetivos específicos:

- Especificar los requerimientos que serán solucionados por el prototipo y los dispositivos con los cuales se llevaran a cabo.
- Diseñar la estructura de unas gafas y un bastón.
- Construir plataforma de sensores integradas a unas gafas y un bastón controlado por Arduino para la toma de decisiones.
- Implementar el prototipo con las tecnologías seleccionadas para garantizar la comunicación entre las gafas y el bastón.
- Realizar pruebas de funcionamiento.

Descripción de los capítulos

A continuación se describe el documento y el desarrollo del proyecto que se ha dividido en cuatro capítulos diferentes.

- El capítulo 1, contiene las generalidades y los conceptos necesarios para conocer de manera clara las particularidades técnicas, actuales, de una persona invidente.
- El capítulo 2, describe netamente el diseño propuesto en el hardware y software, los criterios técnicos con las especificaciones de los elementos propuestos.
- El capítulo 3, se desarrolla la implementación del sistema y se presentan todas las pruebas realizadas para asegurar el correcto funcionamiento del sistema prototipo y que este no presente errores significativos. Adicionalmente se indica los análisis de resultados.

CAPITULO I

FUNDAMENTACIÓN TEÓRICA

1.1. Historia

El argentino José Mario Fallótico en el año de 1921 fue quien inventó el primer bastón (Figura 1) para personas con problemas visuales, vio a una persona ciega que se encontraba esperando para cruzar la calle; lo ayudó pero quedó obsesionado con una pregunta: ¿cómo hacer para distinguir a un ciego y poder ayudarlo?

La persona con discapacidad visual puede saber, a través de la información que le brinda el bastón las técnicas específicas de orientación y movilidad, la orientación del hombre está basada principalmente en su vista, cuando la pierde depende del sentido del tacto y del oído.

Con esto, un bastón es de gran importancia para la orientación y movilidad de las personas ciegas (Aragunde, 2016).



Figura 1. Bastón blanco. (Ximénez, 2015).

1.2. Bastón

Es un instrumento que permite orientar y movilizar a una persona con discapacidad visual, con el afán de desplazarse de manera segura, firmeza y con conocimiento que pueda ser apreciado como una extensión de su propio tacto y es necesario que lo utilicen (Dr. Francisco, 2010).

1.2.1. Tipos de bastones

- Bastón largo: Este bastón, se diseñó como herramienta para movilizarse, dando a conocer los obstáculos en la trayectoria¹. La longitud del bastón depende de la altura de la persona, se extiende tradicionalmente del piso al esternón del usuario.
- Bastón Kiddie: Este bastón está diseñado exclusivamente para los niños.

¹Trayectoria: Lugar geométrico de las posiciones sucesivas por las que pasa un cuerpo en su movimiento.

- Bastón de identificación: Este bastón funciona como medio de la identificación. Además este bastón es limitado, como dispositivo de la movilidad.
- Bastón de ayuda: Se diseña para ofrecer estabilidad física a un usuario con una visión estropeada.
- Bastón de movilidad: Estos bastones se fabrican especialmente de aluminio y plástico reforzado, puede venir con una variedad amplia de divisiones dependiendo de cómo prefiera el usuario (Asociación FENCE , 2010).

1.3. Gafas

Es un instrumento óptico formado por un par de lentes sujetadas a un armazón, el cual se apoya a la nariz mediante un arco y dos varillas (conocidas coloquialmente como “patillas”) que ayudan a sostenerlas en las orejas (Flores, 2012).



Figura 2. Gafas. (Ximénez, 2015).

1.3.1. Armazón o Montura

El armazón o montura es un elemento fundamental en la adaptación de lentes oftálmicos, ya que es el soporte donde se sujetan dichos lentes. Por tal motivo es fundamental una buena escogencia de la montura al momento de la adaptación ya que de eso depende entre otras cosas, una buena calidad de visión y una buena estética.



Figura 3. Gafas de madera. (Ximénez, 2015).

1.3.2. Medidas de Monturas

Son medidas estándares que se comparan entre monturas. El primer número indica el diámetro del cristal en milímetros, el segundo número indica el ancho del puente y el tercero el largo de

la patilla. Las dos últimas medidas, ancho del puente y longitud de la patilla, son las que más influyen en la comodidad de uso de las gafas.



Figura 4. Medidas de gafas. (Ximénez, 2015).

1.4. La ceguera

La baja visión es una limitación de la capacidad visual que afecta el desenvolvimiento de la persona que ejecuta alguna tarea. La ceguera es la ausencia de la percepción de la luz, una persona no vidente tiene una falencia parcial o total del sentido de la vista. La adaptación al medio físico así como social depende de cada individuo, el no vidente parcial puede ser de nacimiento pero reconoce la luz como una sombra, es decir su vista capta sólo el ambiente que está iluminado pero no conoce formas ni colores definidas (Cambre, 2013).

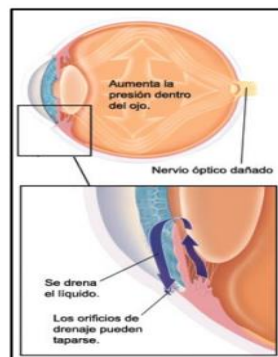


Figura 5. Falencia del sentido de vista. (Cambre, 2013).

1.4.1. Tipos de ceguera

La ceguera consiste en la incapacidad física para ver, es decir, existe algún fallo en el sentido de la vista que impide la visión.

Existen distintos tipos de ceguera dependiendo de la causa, el grado y el tipo de pérdida de visión, además de otros tipos de anomalías que dan lugar a alteraciones en la percepción, como es el caso del daltonismo, los tipos de ceguera se pueden clasificar según la afectación en:

- **Ceguera Parcial**, es cuando la persona tiene una visión muy baja o no tiene la suficiente capacidad de tener una buena visión y se ve obligada a usar anteojos.
- **Ceguera Total o Completa**, es cuando la persona no ve ni siente absolutamente nada, ni la luz, ni su reflejo.

Otro tipo de clasificación que se realiza habitualmente es la realizada a partir de la causa inicial de la ceguera, es decir, según qué estructura dañada sea la que impide la visión. Según esta clasificación, las estructuras que al dañarse producen ceguera son:

- El cristalino, como en las cataratas.
- La córnea, como en la opacidad de córnea.
- La retina, como en la degeneración macular.
- El nervio óptico, como en el glaucoma.
- El cerebro, como en la ceguera cortical.

Los daños en estas estructuras del sentido de la vista se producen con la edad y están influidas por la herencia genética, a excepción del cerebro, que suele dañarse como consecuencia de un impacto en la zona occipital del cráneo durante un accidente, siendo ésta la causa más común de ceguera parcial en menores de treinta años (UltraMed, 2013).

1.4.2. Ayudas a la movilización de los no videntes

La mayoría de personas no videntes no tienen impedimentos para caminar, pero al carecer del sentido de la vista sienten inseguridad y temor al moverse, afecta a su habilidad para establecer relación con los objetos y su capacidad de representación mental del espacio físico. Las condiciones de accesibilidad en los establecimientos públicos y/o privados es un gran inconveniente para la movilización.

Cuando pierden el sentido de la vista se intenta aprovechar la información que pueden brindar los otros sentidos. El sentido auditivo, proporciona información sobre distancia, orientación, dirección, tamaño y peligro.

El sentido del tacto, permite sensaciones de variaciones de presión, temperatura y con adecuada estimulación brindará sensaciones de orientación y memoria muscular. El sentido del olfato, brinda información sobre distancia, orientación y diferenciación.

1.4.3. Gafas con GPS para ciegos

Gafas que integran un completo servicio de GPS integrado, el cual se muestra en una pequeña pantalla al interior del lente, la cual le brindará cientos de datos a cada persona no vidente que lleve puesto estos protectores.



Figura 6. Gafas GPS para ciegos. (Ounae, 2010).

1.4.4. Traje de ayuda para invidentes

Es un traje con una serie de sensores que detectan los objetos del entorno. Esta información se transmite mediante vibraciones en la superficie del traje, que el usuario recibe e interpreta. Algo similar, según la página del proyecto, al sistema que utilizan las antenas de los insectos o los bigotes de los gatos.

Este traje va dirigido a invidentes, además a aquellos que trabajan en entornos peligrosos detecta obstáculos para conductores (en este caso los sensores cubrirían la superficie del vehículo) (RunaTech, 2014)



Figura 7. Traje de ayuda para invidentes. (RunaTech, 2014).

1.4.5. Una vara para ciegos

Cuando la ciencia hace aportes a enfermedades o padecimientos de algún tipo es una doble celebración en este caso la tecnología ayuda a los ciegos a moverse con más facilidad y evitando los obstáculos.

Diseñado por WooJin Han esta vara para ciegos hace cierta referencia al clásico bastón blanco, sólo que con un poco de tecnología que permite detectar los obstáculos sin necesidad de golpearlos.

El tubo tiene sensores incorporados que miden la distancia de los objetos y avisan por medio de vibraciones al portador de su cercanía, para los evite.



Figura 8. Una vara para ciegos. (Impresionante, 2005).

1.4.6. Robot para ciegos

Maggie es un robot diseñado para la asistencia a personas con discapacidad visual. Según sus constructores, se trata del primer prototipo de estas características desarrollado en Europa, dotado a una conexión a internet, cámaras de televisión y tecnología láser.

El robot, un humanoide de 1.4 m de altura, es capaz de hablar, bailar, reconocer voces, informar de las últimas noticias y del tiempo, distinguir los medicamentos, e incluso, es capaz de sentir cosquillas. Posee, además, una pantalla táctil desde la que se le pueden dar órdenes (Cienciamisterio, 2007).



Figura 9. Robots para ciegos. (Cienciamisterio, 2007).

1.4.7. Retina artificial

Tiene un chip que se coloca en la retina del no vidente. El microprocesador mide dos milímetros de diámetro (más pequeño que la cabeza de un alfiler) y contiene unas cinco mil pequeñas células solares, cada una están unidas a electrodos.

La idea es que cuando el chip se encuentra en retina la luz llega (a través del ojo) active los electrodos estimulando los nervios ópticos, luego transporta al cerebro una señal. Algunos especialistas critican el método porque consideran que las células difícilmente generen energía eléctrica.

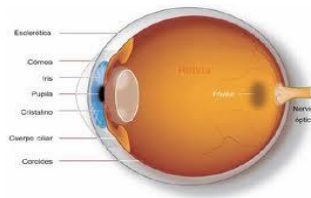


Figura 10. Retina artificial. (Muycomputer, 2009).

1.4.8. Estadísticas de la ceguera

En 2013, la Asamblea Mundial de la Salud aprobó el Plan de acción para la prevención de la ceguera y la discapacidad visual, 2014-2019, a modo de hoja de ruta para los Estados Miembros, la Secretaría de la OMS y los asociados internacionales, con el objetivo de lograr una reducción mensurable de la discapacidad visual evitable de un 25% de aquí a 2019.

Alrededor de un 65% de las personas con discapacidad visual son mayores de 50 años, si bien este grupo de edad apenas representa un 20% de la población mundial. Con el creciente envejecimiento de la población en muchos países, irá aumentando también el número de personas que corren el riesgo de padecer discapacidades visuales asociadas a la edad (OMS, 2013).

En el Ecuador, las personas con discapacidad visual es del 18,84%, el sistema de valoración de la discapacidad establece normas para la evaluación de las consecuencias de la enfermedad como la restricción o ausencia de la capacidad para realizar actividades de la vida diaria.

1.5. Discapacidad

La discapacidad es cualquier limitación o impedimento de la capacidad de realizar una actividad en la forma o dentro del margen que considera normal al ser humano. Se caracteriza por excesos o faltas en el desempeño de una actividad rutinaria normal, los cuales pueden ser temporales o permanentes, reversibles o surgir como consecuencia directa de la deficiencia o como una respuesta del propio individuo, sobre todo la psicológica, a deficiencias físicas, sensoriales o de otro tipo (Quintero, 2015).



Figura 11. Símbolo de personas con discapacidad física. (Quintero, 2015).

1.5.1. Clasificación de la discapacidad

Discapacidad física: La discapacidad física es la simbolización que cuenta con las alteraciones más frecuentes, las cuales son consecuencias de poliomielitis, lesión medular (parapléjico o cuadripléjico) y amputaciones.



Figura 12. Personas con discapacidad física. (Quintero, 2015).

Discapacidad sensorial: La discapacidad sensorial percibe a las personas con deficiencias visuales, con una muestra de problemas en la comunicación y el lenguaje.



Figura 13. Persona con discapacidad sensorial. (Lopez, 2015).

Discapacidad intelectual: La discapacidad intelectual abarca una cadena de enfermedades como son: los trastornos, el retraso mental, el Síndrome Down y la parálisis cerebral.



Figura 14. Persona con discapacidad sensorial. (Lopez, 2015).

Discapacidad psíquica: La discapacidad psíquica es cuando una persona sufre alteración neurológica y trastorno cerebral (Transtornos, 2016).



Figura 15. Persona con discapacidad psíquica. (Transtornos, 2016).

Discapacidad Visual: Se define como la variación del sistema visual que trae como resultado problemas en el desarrollo de actividades que requieran el uso de la visión (Berdahl, 2016).



Figura 16. Símbolo de personas con discapacidad visual. (Berdahl, 2016).

1.6. Sensores

1.6.1. ¿Qué son los sensores?

Son dispositivos capaces de detectar una variable física y transformarla en una variable eléctrica que sea capaz de cuantificar² y manipular.

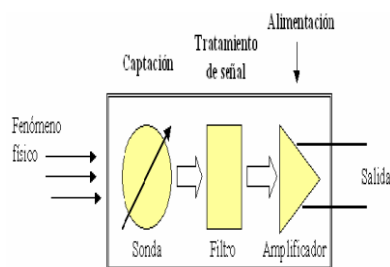


Figura 17. Esquema general de un sensor. (Prometec, 2015)

² Cuantificar: convertir una sucesión de muestras de amplitud continua en una sucesión de valores discretos preestablecidos según el código utilizado.

1.6.2. Características generales de un sensor

a) Características Estáticas

En las características estáticas tenemos los siguientes aspectos:

- **Rango de medida:** Es el conjunto de valores que puede tomar la señal de entrada, alcanzadas entre el máximo y el mínimo detectados por el sensor con una tolerancia de error aceptable.
- **Resolución:** Muestra la capacidad del sensor para discernir entre valores muy próximos de la variable de entrada. Indica qué diferenciación de la señal de entrada produce una variación detectable en la señal de salida.
- **Precisión:** Define la variación máxima entre la salida real obtenida y la salida teórica dada como patrón para el sensor.
- **Repetitividad:** Indica la máxima variación entre valores de salida obtenidos al medir varias veces la misma entrada con el mismo sensor y en idénticas condiciones ambientales.
- **Sensibilidad:** Indica la mayor o menor variación de señal de salida por la unidad de la magnitud de entrada. Mayor sea la variación de la señal de salida producida por una variación en la señal de entrada, el sensor es más sensible.
- **Ruido:** Es cualquier perturbación aleatoria del propio sistema de medida que afecta la señal que se quiere medir.

b) Características Dinámicas

En las características dinámicas tenemos los siguientes aspectos:

- **Velocidad de respuesta:** Mide la capacidad del sensor para que la señal de salida siga sin retraso las variaciones de la señal de entrada.
- **Respuesta en frecuencia:** Mide la capacidad del sensor para seguir las variaciones de la señal de entrada a medida que aumenta la frecuencia, generalmente los sensores convencionales presentan una respuesta del tipo pasa bajos.
- **Estabilidad:** Indica la desviación en la salida del sensor con respecto del valor teórico dado, al variar parámetros exteriores distintos al que se quiere medir (condiciones ambientales, alimentación, etc.) (José, 2008).

1.6.3. Clasificación de los sensores

Sensores de desplazamiento: Es un sensor sin rozamiento, ya que al ser inductivo, la variación es por campo magnético. Este desplazamiento debidamente calibrado, proporciona una relación entre la inducción y la distancia recorrida. Entre estos se tiene:

- Sensores resistivos
- Sensores inductivos y capacitivos
- Sensores ópticos y basados en el Efecto Hall

Sensores de presión: Son elementos que transforman la magnitud física de presión³ o fuerza por unidad de superficie en otra magnitud eléctrica que será la que emplearemos en los equipos de automatización o adquisición estándar. Los rangos de medida son muy amplios, desde una milésima de bar hasta los miles de bar.

Sensores de temperatura: Consisten en una fina película de resistencia variable con la temperatura (RTD) y están calibrados por láser para una mayor precisión e intercambiabilidad. Las salidas lineales son estables y rápidas. Entre estos se tiene:

- Termistores.
- Termocuplas.
- Diodos.

Sensores de velocidad: Están hechos con una bobina móvil fuera de un imán estacionario, el principio de operación es el mismo. Otro tipo de transductor de velocidad consiste en un acelerómetro con un integrador electrónico incluido.

Esta unidad se llama un Velómetro y es en todo aspecto superior al sensor de velocidad sísmico clásico. Entre estos se tiene:

- Tacómetros.
- Codificadores (Eduardo, 2011)

1.7. Arduino

Arduino es un prototipo electrónico de código abierto (open-source) basada en hardware, software flexible con una facilidad de utilizar. Arduino puede sentir el entorno mediante la recepción de entradas desde una variedad de sensores, puede afectar a su alrededor mediante el control de luces, motores. El microcontrolador de la placa se programa usando el Arduino

³ Presión: Magnitud física que mide la proyección de la fuerza en dirección perpendicular por unidad de superficies, y sirve para caracterizar cómo se aplica una determinada fuerza resultante sobre una línea.

Programming Language (basado en Wiring) y el Arduino Development Environment (basado en Processing).

1.7.1. ¿Por qué Arduino?

Hay muchos otros microcontroladores y plataformas microcontroladoras disponibles para computación físicas. Todas estas herramientas toman los desordenados detalles de la programación de microcontrolador y la encierran en un paquete fácil de usar. Arduino simplifica el proceso de trabajo con microcontroladores, pero ofrece algunas ventajas:

- **Barato:** Son económicas y baratas comparadas con otras plataformas microcontroladoras. La versión menos cara del módulo Arduino puede ser ensamblada a mano.
- **Multiplataforma:** El software de Arduino se ejecuta en sistemas operativos Windows, Macintosh OSX y GNU/Linux. La mayoría de los sistemas microcontroladores están limitados a Windows.
- **Entorno de programación simple y clara:** El entorno de programación de Arduino es fácil de usar para principiantes y estarán familiarizados con el aspecto y la imagen de Arduino.
- **Código abierto y software extensible:** El software Arduino está publicado como herramientas de código abierto, disponible para extensión por programadores experimentados. El lenguaje puede ser expandido mediante librerías C++, y la gente que quiera entender los detalles técnicos pueden hacer el salto desde Arduino a la programación en lenguaje.
- **Código abierto y hardware extensible:** El Arduino está basado en microcontroladores ATMEGA8 y ATMEGA168 de Atmel. Los planos para los módulos están publicados bajo licencia Creative Commons, por lo que diseñadores experimentados de circuitos pueden hacer su propia versión del módulo, extendiéndolo y mejorándolo.



Figura 18. Esquema Arduino. (Arduino)

1.8. Módulo RF

Estos módulos establecen una comunicación inalámbrica unidireccional, esto quiere decir que manda una señal y el otro la recibe. Con esto se puede realizar proyectos con robots, vehículos pequeños o la transmisión de datos el cual comanda una buena comunicación.

En tiempos atrás la posibilidad económica era construcción y calibración de circuitos para esto se debe dedicar tiempo y así cumplir los objetivos planteados.

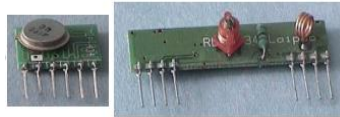


Figura 19. Módulo Emisor y Receptor. (Prometec, 2015)

1.9. Tarjeta Micro SD

Se estudia la tarjeta micro SD con una capacidad de 16 GB, en ella se guarda el sistema operativo de tarjeta Arduino Uno, es de importancia asignar el sistema operativo para poder actualizar las librerías de la tarjeta y tener una conexión óptima hacia los demás dispositivos. (Sandisk, 2012)

1.10. Módulo RTC

El Módulo RTC (Reloj en Tiempo Real) lleva un registro del tiempo que ha sido transcurrido. Con este módulo se pueden realizar diferentes tipos de proyectos como alarmas, estación de sensores y sondas de registro con esto se lleva un estricto conteo sobre el tiempo que está transcurriendo.



Figura 20. Módulo RTC. (Prometec, 2015)

1.11. Módulo GPRS

Si bien existen varios modelos de módems o módulos que permiten el envío de información a través de la red GPRS, los más fáciles de conseguir en el mercado local son: el módem

Multitech GSM/GPRS MTCBA-GF2 y el SIM900 GSM/GPRS Shield. SIM900 GSM/GPRS Shield.

1.12. El SIM900 GSM/GPRS Shield

Es totalmente compatible con la mayoría de dispositivos Arduino, ya que se lo utiliza para desarrollar proyectos o aplicaciones como por ejemplo: control remoto, estaciones meteorológicas remotas, sistemas de rastreo de automóviles. Muestra el aspecto físico del SIM900 GSM/GPRS Shield. (Simbaña, 2015)



Figura 21. Módulo GPS. (Prometec, 2015)

Después de analizar las especificaciones técnicas y características de los dos dispositivos mencionados anteriormente, se selecciona el SIM900 GSM/GPRS Shield debido a los siguientes factores:

- Se puede tener una mejor presentación estética del sistema ya que se coloca directamente encima del dispositivo Arduino Mega 2560, mientras que con el módem Multitech GSM/GPRS es necesario utilizar cables para poder conectarse al dispositivo Arduino Uno.
- Presenta menor costo con respecto al Módem Multitech GSM/GPRS MTCBA-GF2.

1.13. Sistema global para las comunicaciones móviles (GSM)

Se define la Red del Sistema Global de Telefonía GSM como aquel servicio portador constituido por todos los medios de transmisión y conmutación necesarios que permiten enlazar a voluntad dos equipos terminales móviles mediante un canal digital que se establece específicamente para la comunicación y que desaparece una vez que se ha completado la misma. (Rodríguez, 2016)

El Sistema Global para Comunicaciones Móviles, originalmente desarrollado como estándar europeo para la telefonía móvil digital, se ha convertido en el sistema móvil de uso más

difundido en el mundo. Opera en las frecuencias de 900 y 1800 MHz en Europa, Asia, Australia con una frecuencia de 1900 MHz en Norteamérica y Latinoamérica. (Rodríguez, 2016)

1.14. Módulo detector de agua

Este sensor de bajo costo cuenta con una serie de alambres en paralelo expuestos con el fin de determinar la presencia de agua en valores de salida de conversión de señales analógicas y digitales. Debido a su fácil programación es ideal para proyectos relacionados con cualquier versión de Arduino, microcontroladores PIC y Raspberry. Puede obtener nuevos datos cada 2 segundos, es decir no pueden realizarse mediciones con un intervalo menor. La identificación del sensor se da mediante gotas de agua, el sensor de detección es de rendimiento total con una conversión analógica, el valor de salida se aplica a su función personalizada. Es de bajo consumo de energía y alta sensibilidad. El rendimiento se mejora con un controlador Arduino y con un protector de relés.

El sensor de nivel de agua es fácil de usar, rentable de alto nivel / sensor de reconocimiento de caída, que se obtiene mediante una serie de cables en paralelo expuestos tramas / volumen de agua con el fin de determinar el nivel del agua. Fácil de completar el agua a la conversión de señal analógica y los valores analógicos de salida se puede leer directamente al Arduino proporcionando señales de alarmas.

Especificaciones técnicas

- Tipo de Sensor: Analógico
- Tensión de trabajo: DC3 - 5V
- Corriente de funcionamiento: <20mA
- Área de detección: 40 mm x 16 mm
- Temperatura de funcionamiento: 10 grados a - 30 grados
- Humedad: 10% - 90% sin condensación

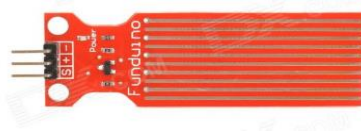


Figura 22. Módulo Detector de Agua. (Prometec, 2015)

CAPITULO II

PROPUESTA

Análisis de Requerimientos

A continuación se describen las características que presenta la el Arduino uno y el Arduino Nano, para poder usar en el desarrollo del proyecto entre los cuales están: periféricos, pines de entrada y salida, capacidad de almacenamiento, así como las características de los otros componentes los cuales son GPRS, Módulo RF, Módulo reproductor Mp3, Sensores de distancia, Sensor infrarrojo, Módulo RTC DS3231, Módulo detector de agua.

En el diseño del sistema de detección de obstáculos se utiliza una tarjeta de control denominada Arduino Uno que está basada en software libre, facilita el diseño del prototipo y el acople de los distintos dispositivos, se procederá a realizar el estudio de los parámetros necesarios para que el dispositivo de módulos de transmisión con recepción tenga la habilidad de “conexión el cual reciba datos”, de manera que sea posible generar una red interna como se indica en la Figura 23.

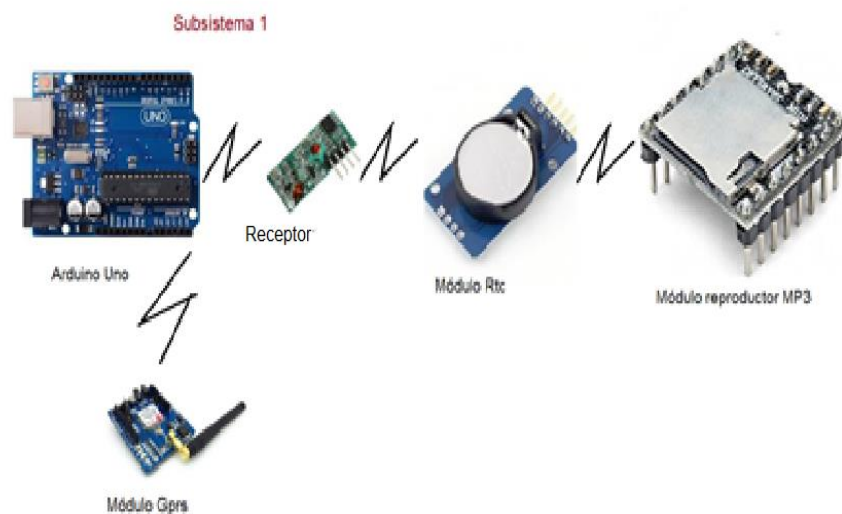


Figura 23. Comunicación entre los dispositivos.

El subsistema 1, está constituido por la tarjeta Arduino Uno, Transmisor RF, Módulo RTC DS3231, reproductor Mp3 y el Módulo GPRS que permite la comunicación inalámbrica, acceso a la hora y adicionalmente la comunicación por bluetooth, este transmite la información al subsistema 2 que está constituida por unas gafas con sensores y esta a su vez envía datos a la tarjeta de control.

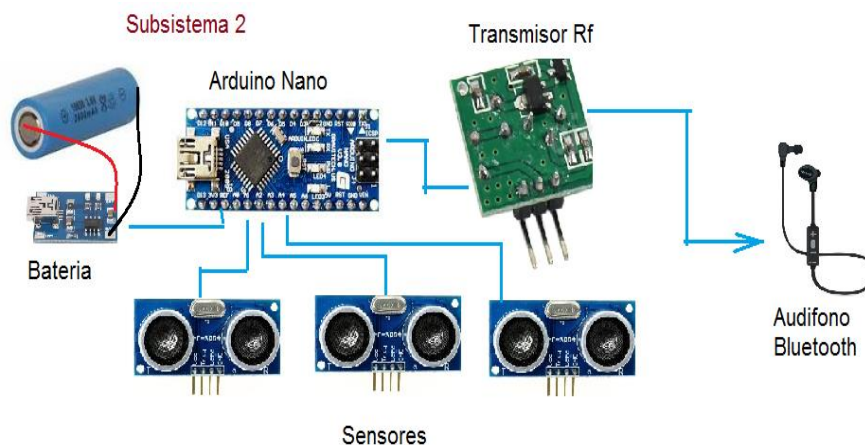


Figura 24. Componentes en Gafas.

El subsistema 2, que se muestra en la Figura 24 está constituido por elementos: un Arduino nano, un transmisor de RF, una batería con carga USB y 3 sensores ultrasónicos que ayudará a la persona invidente a realizar una toma de decisión si el obstáculo se presenta en la parte frontal, lateral derecha, lateral izquierda dando paso a escuchar alarmas mediante unos audífonos con bluetooth.

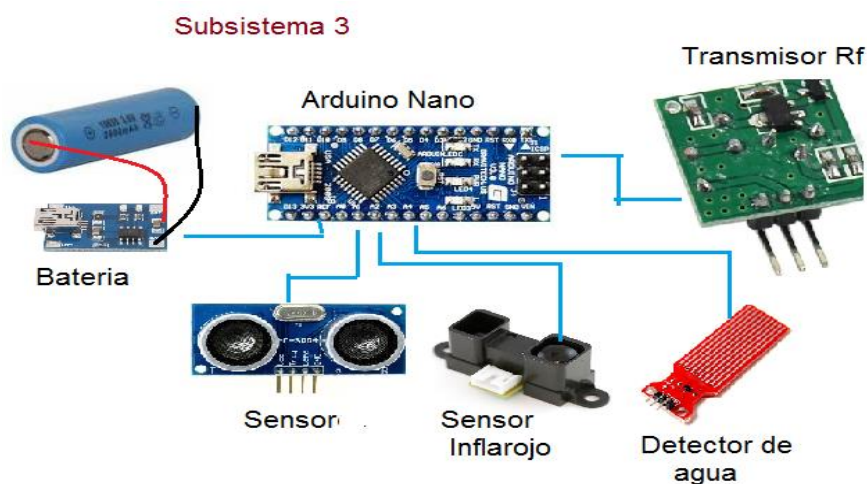


Figura 25. Componentes en el Bastón.

El subsistema 3, que se muestra en la Figura 25 está constituido por elementos como: un Arduino Nano, un Transmisor de RF, una batería con carga USB, un sensor ultrasónicos, sensor infrarrojo, un módulo detector de agua, este sistema detecta obstáculos en la parte inferior del bastón, detecta humedad y desniveles, estas alarmas serán reproducidas mediante unos audífonos con bluetooth.

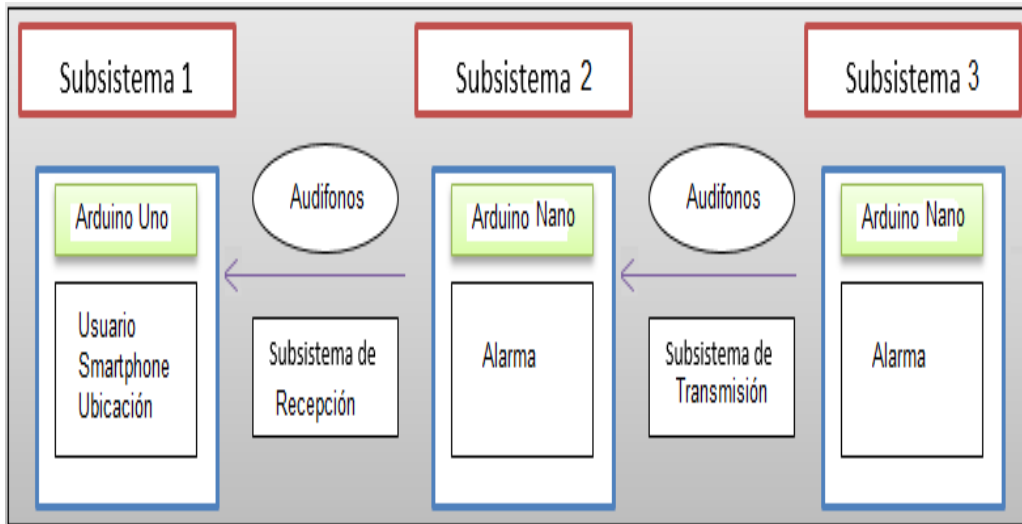


Figura 26. Subsistemas de Transmisión y Recepción.

A continuación se describe el diagrama del primer bloque que consta de un Arduino Uno, esta se encargada de recibir la señal entre las gafas y el bastón de igual forma se obtiene una ubicación mediante GPRS.

El segundo bloque envía la información a través de la localización de los sensores lo cual crea un aviso a la persona invidente.

El tercer bloque transmite información a la tarjeta de control con alertas y oídas por medio de unos audífonos. La Figura 26, muestra los bloques que forman parte de los subsistemas de transmisión y recepción, con cada uno de sus respectivos módulos.

2.1. Diseño

En esta sección se describe el diseño del sistema prototipo. Se realiza la instalación y la descripción del software, adicionalmente se hace un análisis de todas las conexiones diagramas cálculos del hardware que se utiliza en el proyecto.

2.1.1. Arquitectura del Sistema prototipo

Esta etapa define el esquema por el cual los elementos funcionales del dispositivo se adecúan de manera que los mecanismos físicos que ponen en práctica las aplicaciones del producto se desempeñen adecuadamente. En primera instancia se estableció el esquema descriptivo del dispositivo, para después plasmar esa noción en un entorno visual que dé la primera aproximación del aspecto físico que tienen las gafas y el bastón.

2.1.2. Diagrama general de bloques del sistema prototipo

En la Figura 27, se muestra un diagrama de bloques general del funcionamiento del prototipo de las gafas y bastón inteligente.

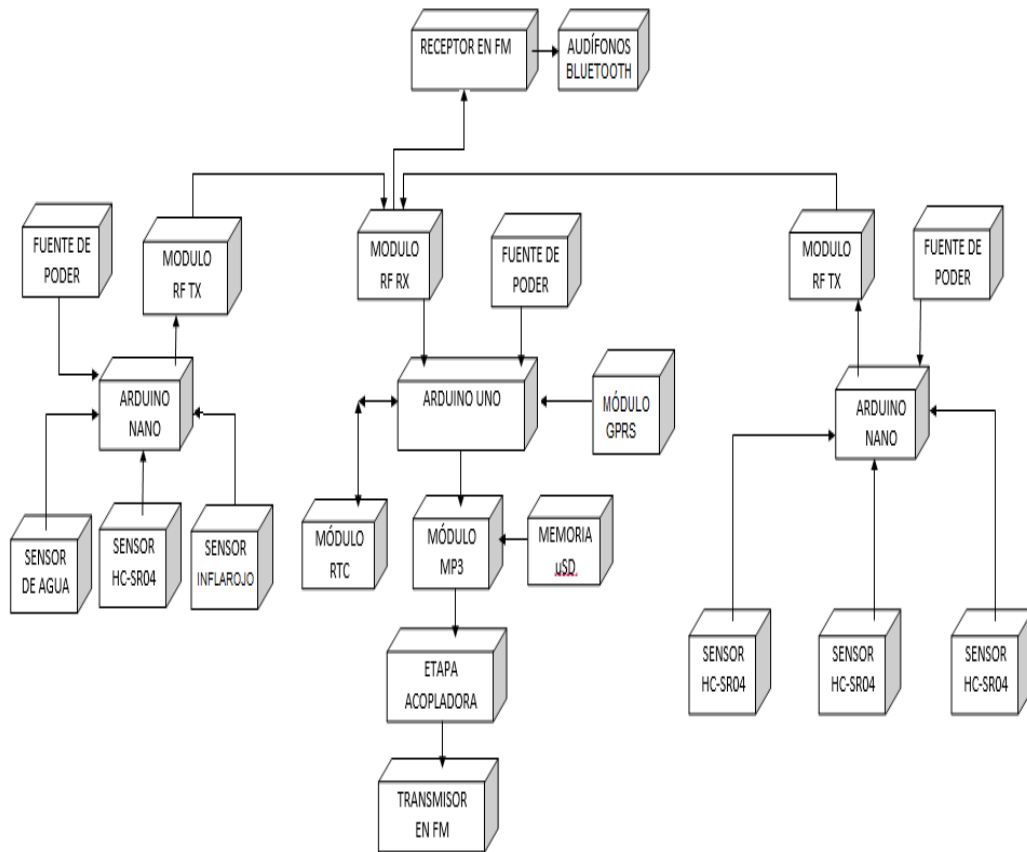


Figura 27. Diagrama de bloques general del sistema prototipo.

El diagrama de bloques de la Figura 27, muestra los dispositivos usados para el sistema prototipo, en esta parte del proyecto se distribuye en bloques para representar una función específica en el prototipo a desarrollar, como se muestra a continuación:

1. El Sistema de Comunicación está encargado en transmitir y recibir toda la información por parte de los Módulos RF.

2. El Sistema de Control procesa toda la información y al mismo tiempo envía las alarmas establecidas a los audífonos para que así el usuario tenga precaución con los obstáculos.

3. El Sistema de localización GRPS recibe las señales provenientes de la tarjeta principal que se halla en la cintura de la persona invidente teniendo un botón de emergencia que envía un mensaje de texto a la persona que le está rastreando.

4. El Sistema de Seguridad permite ejecutar las diferentes acciones cuenta con sensores que se encargan de detectar obstáculos ya sea en las gafas o en el bastón dando señales de alarmas.

5. El Sistema de Alimentación permite suministrar el voltaje y corriente necesarios para que el prototipo entre en funcionamiento normal. Si se descarga la batería optamos por cargarla por medio de un conector USB para los tres sistemas.

2.1.3. Bloque del sistema de control

Este sistema de control o tarjeta principal que está compuesta por el Arduino Uno es el encargado de gobernar todo el prototipo, recibiendo y enviando las órdenes que tiene que realizar cada uno de los bloques, además en éste se encuentran interconectados todos los módulos usados para el desarrollo del proyecto, tanto para controlar la detección de los obstáculos, localización GPRS, como para enviar / recibir solicitudes y respuestas; en resumen, el bloque del sistema de control toma las decisiones que serán ejecutadas por los demás bloques.

La Figura 28, muestra el diagrama de bloques de la etapa de control que constituye el cerebro del prototipo. Es el que controla las acciones de los módulos que están interconectados, dependiendo de las decisiones que se tome.

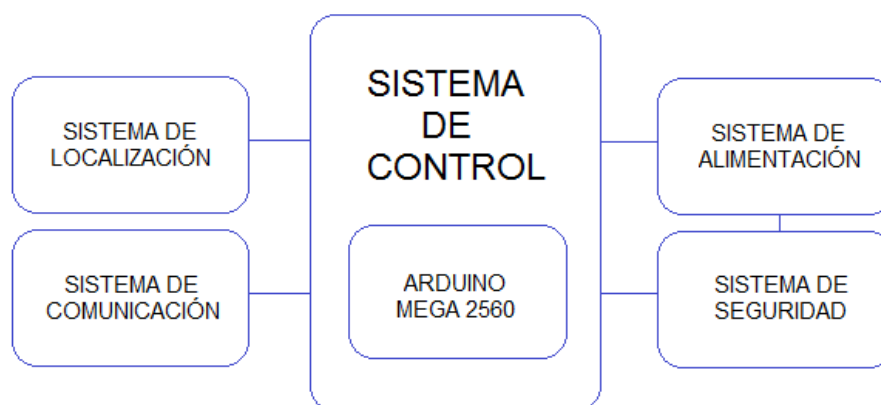


Figura 28. Diagrama de bloques del sistema de control.

2.1.4. Diseño electrónico

La Figura 29, muestra el Hardware y Software del sistema con su estructura, tomando en cuenta la interconexión y comunicación entre los elementos físicos que conforman el sistema prototipo. Cabe recalcar que se utilizaron etapas de potencia para un excelente control de sus partes móviles.

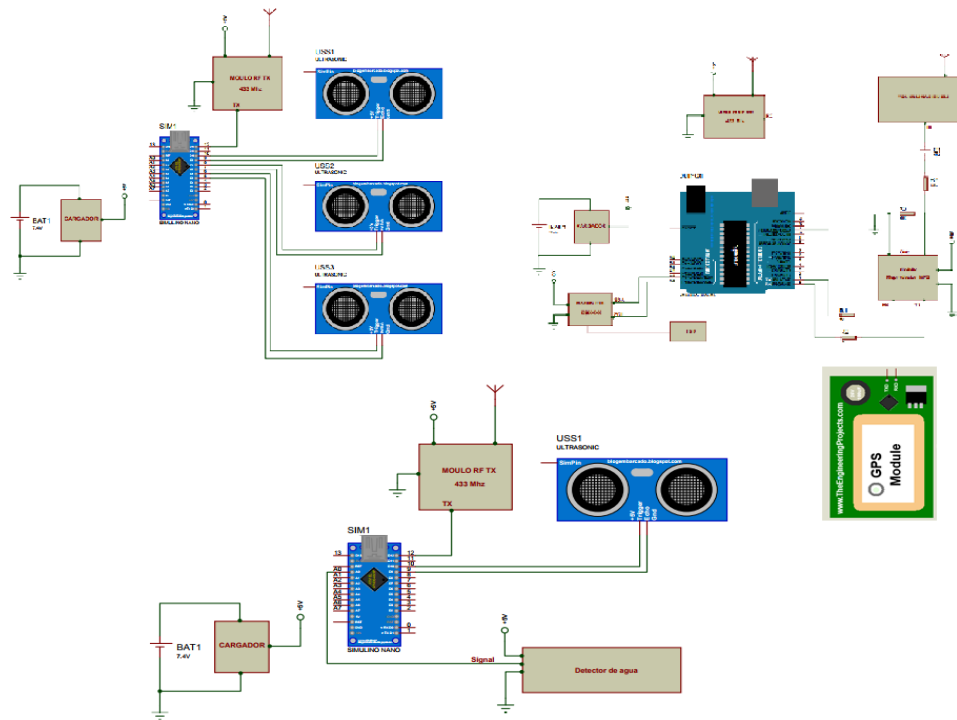


Figura 29. Diagrama electrónico del Prototipo.

En la Figura 29, se muestra el esquema propuesto para la conexión de los tres circuitos de control con el modo de conexión entre la placa principal.

2.1.5 Descripción de diseño de los componentes

En esta sección se va a diseñar las principales funciones que controlan los dispositivos y su funcionamiento.

2.1.5.1 Arduino Uno

Arduino es una plataforma electrónica de hardware libre basada en una placa con un microcontrolador de la familia Atmel AVR. El dispositivo tiene la capacidad de leer información de temperatura, presión, humedad, entre otros, a través de sus pines de entrada, para así poder controlar elementos en el entorno que le rodean (luces, motores, actuadores etc).

El microcontrolador en la placa Arduino se programa mediante el lenguaje de programación Arduino (basado en Wiring) en el entorno de desarrollo Arduino (basado en Processing). Finalmente se puede mencionar que el software es totalmente gratuito disponible en cualquier página web de Arduino. (Arduino, Arduino Mega 2560, 2014).

En el mercado existen variedades de placas Arduino, cuya elección depende de la necesidad del proyecto que se quiere realizar. La figura 29 muestra al Arduino Mega 2560, cuyas principales características son las siguientes: (Simbaña, 2015)

- Posee 54 pines digitales que funcionan como entradas/salidas.
- Cuenta con 16 entradas analógicas.
- Su cristal oscilador es de 16 MHz.
- Se puede conectar a la computadora a través de una conexión USB (Universal Serial Bus).
- Posee un botón de reset y una entrada para la alimentación de la placa.

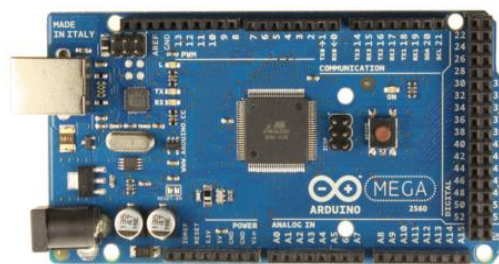


Figura 30. Arduino Uno. (Store, 2017).

Características técnicas

En la Tabla 1, se detallan las características que tiene el Módulo Arduino Mega 2560 que se va a utilizar.

Tabla 1

Características Técnicas Módulo Arduino Uno.

Microcontrolador	ATMega168
Voltaje de operación	5V
Voltaje de entrada (Recomendado)	7V a 12V
Voltaje de entrada (Límite)	6V a 20V
Pines para entrada - salida digital	54
Pines de salidas	14 se usan como salida de PWM
Corriente continua para las entradas	40 mA
Salida de alimentación a 3.3V	50 mA
Memoria Flash	256 KB (el bootloades ocupa 8 KB)
SRAM	8 KB
EEPROM	1 KB
Frecuencia de reloj	16 MHz

Nota. Fuente: (Tinajero, 2014)

2.1.5.2 Módulo Emisor y Receptor RF 433mhz.

Los Módulos RF 433 MHz funcionan como un enlace de datos simplex, transmiten en un solo sentido. Aun así resultan extremadamente útiles en aplicaciones sencillas que no requieren una comunicación bidireccional. **Conexiones**

En el caso del emisor (XY-FST):

- ATAD (Data) Pin digital 3 (Se puede cambiar por otro código en el programa)
- Vcc—>+5V
- Gnd—>Gnd

Para lograr mayor alcance se alimenta el módulo hasta 12V. Para esto se debe compartir la tierra (referencia a 0V) con el Arduino.

Para el receptor (XY-RF-5V)

- Gnd—>Gnd
- DATA—>Pin digital 3 (Se puede cambiar por otro código en el programa)
- DATA—>Este pin lo dejaremos sin conectar, solo usaremos un pin de datos (Podemos usar cualquiera de los dos).
- Vcc—>+5V

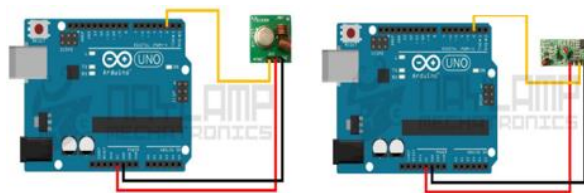


Figura 31. Módulo TX-RX 433 MHz. (Patagoniatec).

Una vez que se enciende el prototipo el Arduino lee todos los caracteres que se han grabado en la memoria, dando paso a una secuencia de datos que fue programada para enviar y procesar la información.

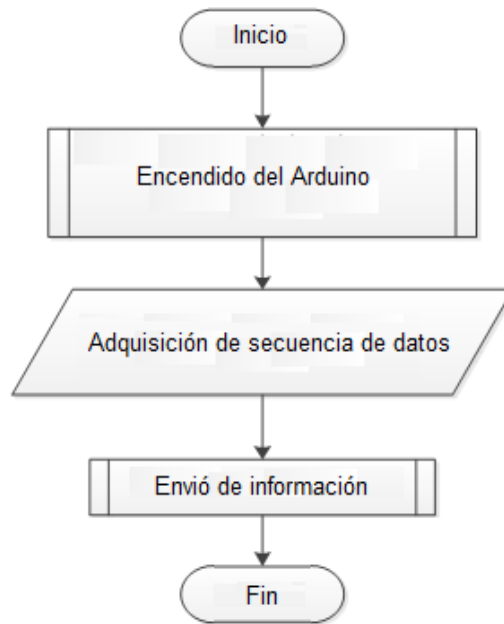


Figura 32. Diagrama del Arduino a los Módulos RF.

2.1.5.3 Sensor de proximidad HC-SR04

El Sensor ultrasónico HC-SR04 mide y detecta la distancia de un objeto, el nivel de rango es de 2 cm a un máximo de 400 cm es verdaderamente compatible con el Arduino, tomar en cuenta que este sensor función en una superficie lisa perpendicular a la propagación del sensor.

En la Figura 33, se realiza la conexión entre el Arduino, el sensor HC-SR04 para la ejecución del proyecto a tratar, cada vez que el sensor detecta un obstáculo le refleja una alarma de aviso.

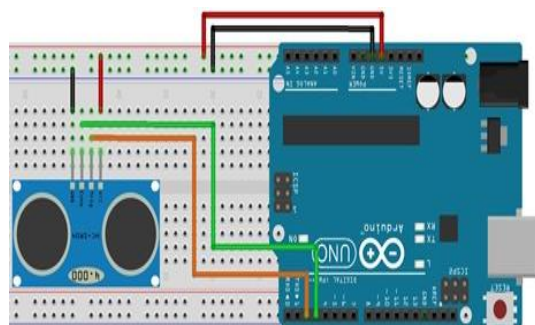


Figura 33. Diagrama del Arduino a los sensores ultrasónicos. (Patagoniatec).

En la Figura 34, se configuran los pines del Arduino con la comunicación a 9800 baudios. El bucle Void Loop() envía un pulso de 10us al Tigger del sensor ultrasónico. Luego recibe el pulso del sensor en el pin Echo, la función pulseIn (pin, Value) ayuda a medir el pulso, la

variable t es el tiempo en que llega el eco del ultrasónico y luego el sensor calcula la distancia del obstáculo.

```
const int Trigger = 2; //Pin digital 2 para el Trigger del sensor
const int Echo = 3; //Pin digital 3 para el echo del sensor

void setup() {
  Serial.begin(9600); //inicializamos la comunicación
  pinMode(Trigger, OUTPUT); //pin como salida
  pinMode(Echo, INPUT); //pin como entrada
  digitalWrite(Trigger, LOW); //Inicializamos el pin con 0
}
```

```
digitalWrite(Trigger, HIGH);
delayMicroseconds(10); //Enviamos un pulso de 10us
digitalWrite(Trigger, LOW);
```

```
t = pulseIn(Echo, HIGH); //obtenemos el ancho del pulso
```

Figura 34. Inicialización del programa con el Sensor HC-SR04.

En la Figura 35, muestra el diagrama de flujo de lectura del Sensor ultrasónico.

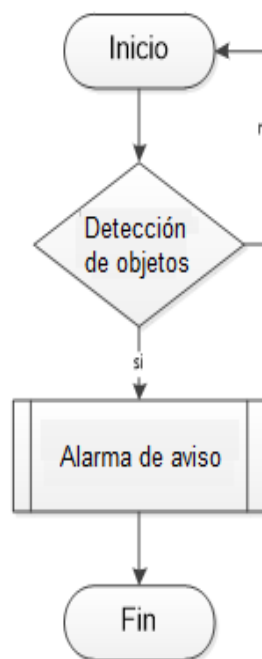


Figura 35. Diagrama de flujo lectura del Sensor ultrasónico.

2.1.5.4 Diagrama electrónico del sistema de comunicación GPRS

Se diseña el sistema de comunicación GSM con el fin de enviar mensajes de texto SMS al Smartphone, la conexión del módulo se lo realiza mediante la comunicación serial UART, a través de los pines de transmisión y recepción.

Tabla 2

Voltaje de operación	9V~20V
Comunicación	UART
Bandas de Frecuencia	850/900/1800/1900MHz
Multi-GPRS	Slot clase 10/8
Estación móvil GPRS	Clase B
Fase GSM	2/2+
Clase 1	1W a 1800/1900MHz
Clase 4	2W a 850/900MHz
Servicio de mensajería (cortos)	Envío de pequeñas cantidades de datos a través de la red (ASCII o primas hexadecimal)
Pila embebida TCP/UDP	Carga de datos a un servidor web
Puerto	Serie (Libre selección)
Altavoz y tomas de auriculares	Si
Bajo consumo de energía	1.5mA (modo dormir)
Temperatura de operación	-40°C~+85°C
Soporta	SuperCap

Nota. Fuente: (Tinajero, 2014)

En la Figura 36, se muestra los pines de conexión usados en la placa Arduino Uno se conecta a los pines de Transmisión y Recepción del módulo de comunicación GPRS SIM900, el pin usado es el 7 para transmisión que va conectado al Arduino Uno para la recepción, este sistema es el encargado de enviar la ubicación por medio de un SMS.

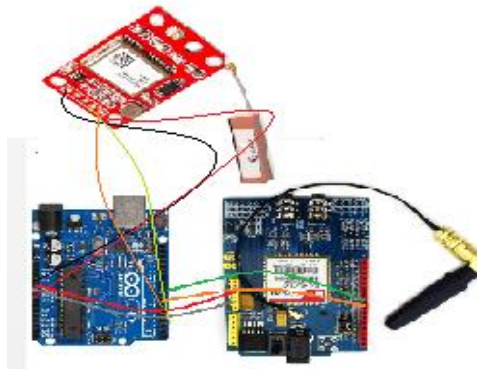


Figura 36. Esquema de conexión del módulo GPRS al Arduino Uno.

2.1.5.5 Módulo RTC DS3231

Este módulo funciona en el formato de 24 horas o 12 horas con indicador AM / PM e incluye dos alarmas programables y una salida de onda cuadrada. Los datos se transfieren en serie a través de un bus bidireccional **I2C**.

La Figura 37, muestra este módulo que es de tiempo real compatible con la placa Arduino. Al trabajar desde 5.5V, incorpora una pila que mantiene la fecha y hora, aunque esté en ausencia de corriente eléctrica.

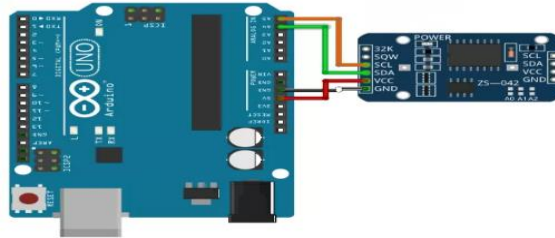


Figura 37. Esquema de conexión del Arduino Uno y Módulo RTC DS3231.

Características técnicas

En la Tabla 3, se detallan los pines de conexión del Arduino Uno y el Módulo RTC DS3231.

Tabla 3

Pines especiales del Módulo Arduino.

CONEXIÓN	
MÓDULO RTC DS3231	ARDUINO UNO
SCL	A5
SDA	A4
Vcc	5V
Gnd	Gnd

Nota. Fuente: (Tinajero, 2014)

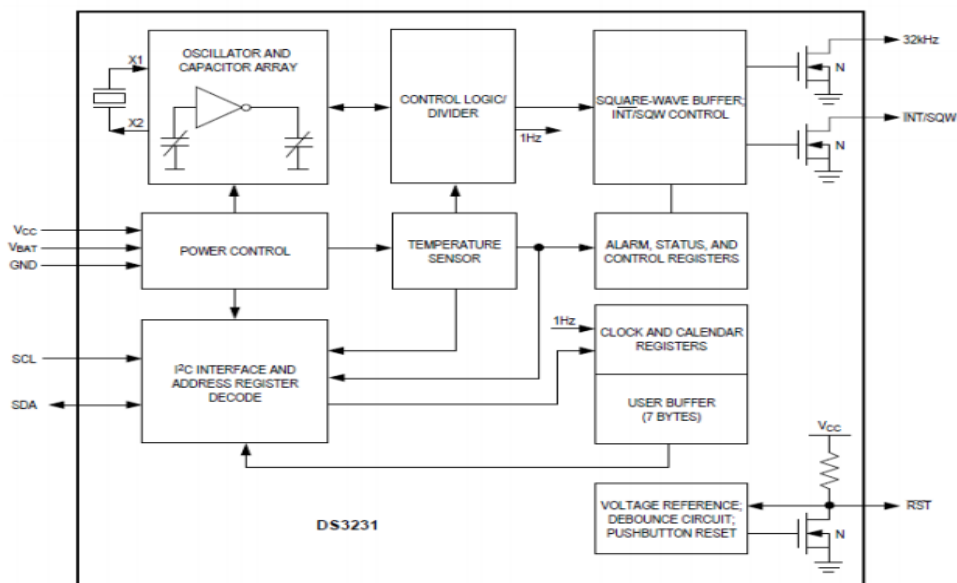


Figura 38. Diagrama de bloques del Módulo RTC DS3231. (Tinajero, 2014).

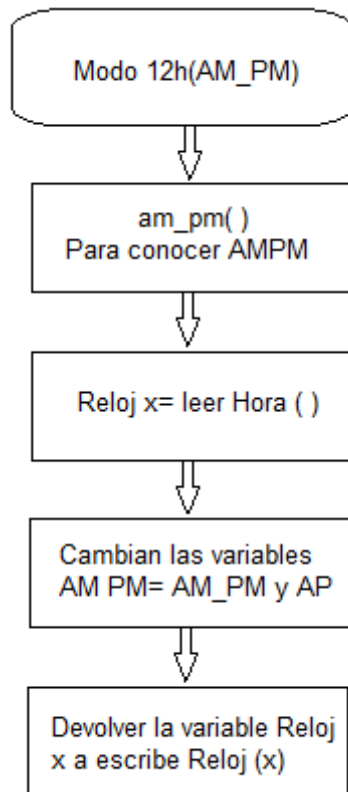


Figura 39. Diagrama de flujo del módulo RTC DS3231.

2.1.5.6 Módulo Reproductor Mp3

Es un mini reproductor de audio, que consta de una tarjeta lector micro SD con capacidad de 32GB, dispone de 6 niveles con una señal de ruido de 85db. Este módulo recibe comandos por un puerto en serie escuchado en unos audífonos con bluetooth.

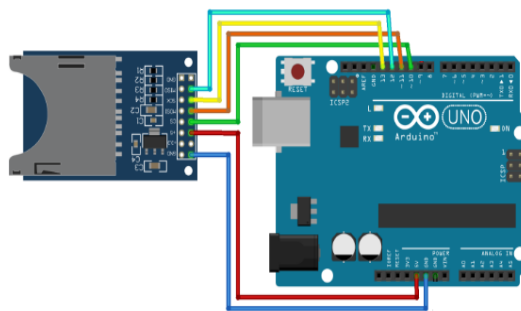


Figura 40. Esquema de conexión del Arduino Uno y Módulo Mp3. (Patagoniatec).

Características técnicas

En la Tabla 4, se detallan los pines de conexión del Arduino Uno y el Módulo Reproductor Mp3.

Tabla 4

Pines especiales del Módulo Mp3.

	MÓDULO MP3	ARDUINO UNO
Vcc	16 (VDD)	3,3V
Gnd	8 (Gnd)	Gnd
RESET	1 (RESET)	D2
CLOCK	7 (P04)	D3
DATA	10 (P05)	D4
BUSY	15 (P06)	D5

Nota. Fuente: (Tinajero, 2014)

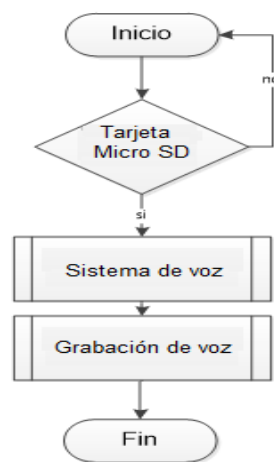


Figura 41. Diagrama de flujo del Módulo Mp3.

2.1.5.7 Sensor infrarrojo Sharp

El Sensor infrarrojo Sharp emite un haz de luz constante que al topar con un obstáculo detecta cual es la distancia entre ese objeto y el sensor, el rango de medición o detección de este sensor va desde los 10 hasta los 80 cm, es preciso ya que cuenta con una salida de datos analógicos. La conexión del sensor y el Arduino requiere de 3 conexiones que 2 son GND y VCC.

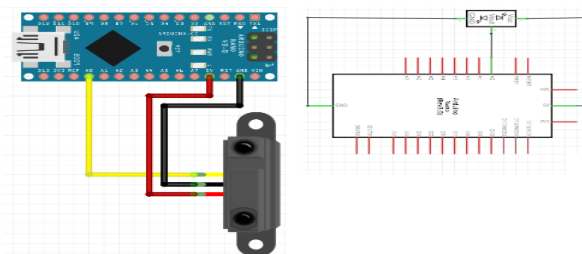


Figura 42. Esquema de conexión del Arduino Nano y Sensor Sharp. (Patagoniatec)

Una vez programado el código se abre el monitor serie y se mostrará la lectura del sensor, se debe colocar un objeto frente a él a diferentes distancias y se visualiza el dato que envía el monitor serie.

```

1 // Conexiones
2 int sensor_distancia_1 = A0;
3
4 // Inicialización
5 int lectura = 0;
6
7 void setup() {
8   Serial.begin(9600);
9   Serial.println("Iniciando lectura de sensor");
10 }
11
12 // Programa
13 void loop() {
14   lectura = analogRead(sensor_distancia_1);
15   Serial.println(lectura);
16   delay(1000);
17 }

```

Figura 43. Inicialización del programa con el Sensor Sharp.

Características técnicas

En la Tabla 5, se detallan los pines de conexión del Arduino Nano y el Sensor Sharp.

Tabla 5

Pines especiales del Sensor Sharp.

CONEXIÓN		
PIN	SENSOR SHARP	ARDUINO NANO
1	V _o	A0
2	Gnd	Gnd
3	V _{cc}	5V

Nota. Fuente: (Tinajero, 2014)

En la figura 44, muestra los 3 pines de conexión del sensor: alimentación, tierra y transmisión/recepción de data (comunicación con el procesador principal); además del diagrama de bloques del sensor.

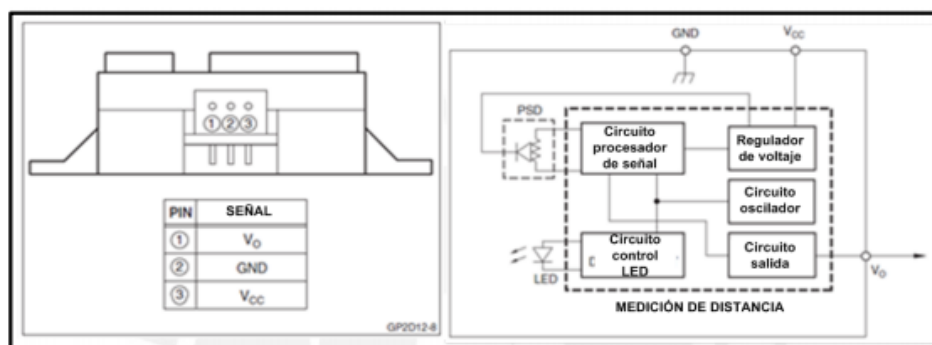


Figura 44. Conexión y diagrama de bloques del Sensor Sharp. (Domínguez, 2015)

2.1.5.8 Sistema de alimentación

Para el sistema de alimentación del prototipo se utiliza la conexión de las baterías en paralelo el cual está encargado de suministrar el voltaje necesario a cada uno de los dispositivos y módulos que conforman la totalidad del sistema prototipo. Se encuentra distribuido en varias etapas, las cuales cumplen una funcionalidad específica. Una característica especial es aquella que permite al prototipo continuar el funcionamiento normal para la función de GRPS ya que este dispositivo necesita una alimentación independiente de cada una de las tarjetas.

En la Figura 45, se observar el diagrama del sistema de alimentación con cada uno de los bloques que lo conforman.

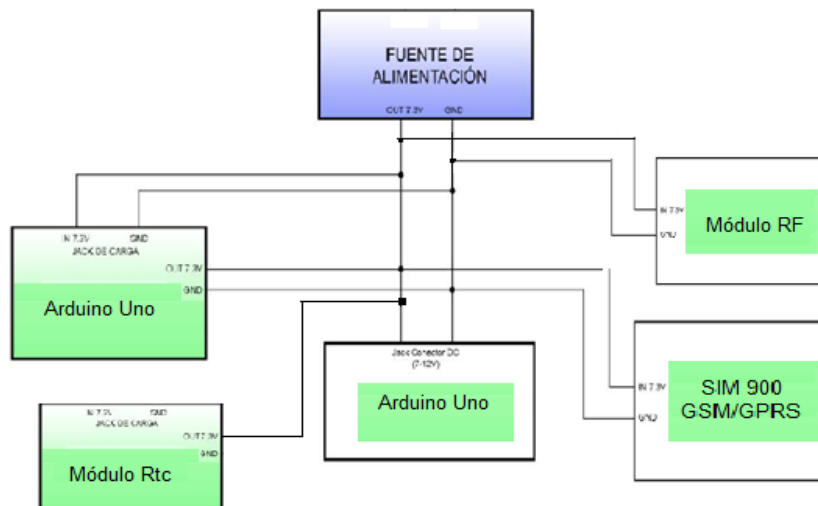


Figura 45. Diagrama de bloques del sistema de alimentación.

2.1.5.9 Sensor de agua

El sensor se caracteriza por sus niveles de tensión de salida, se conecta a una entrada digital de nuestra placa Arduino Nano compatible. Su corriente es menor de 20mA y su voltaje de salida no es superior de 4,2V.

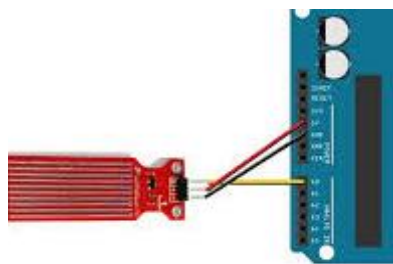


Figura 46. Esquema de conexión del Arduino Nano y el Sensor de agua.

Características técnicas

En la Tabla 6, se detallan los pines de conexión del Sensor Sharp.

Tabla 6:

Pines especiales del Sensor Sharp.

CONEXIÓN		
PIN	SENSOR DE AGUA	ARDUINO NANO
1	S	A0
2	+	5V
3	-	GND

Nota. Fuente: (Tinajero, 2014)

2.1.6.0 Arduino Nano

Al alimentar el Arduino a través del Mini USB, el CH340 proporciona una salida de 3.3V en el pin 16 de la placa. Por ende, cuando se conecta a una fuente externa (no USB), los 3.3V no se encuentran disponibles.

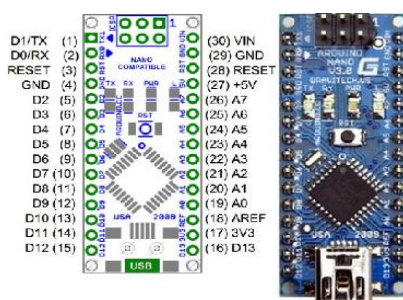


Figura 47. Estructura del Arduino. (Domínguez, 2015)

Tabla 7

Especificación de pines del Arduino Nano.

Pin No.	Name	Type	Description
1-2, 5-16	D0-D13	I/O	Digital input/output port 0 to 13
3, 28	RESET	Input	Reset (active low)
4, 29	GND	PWR	Supply ground
17	3V3	Output	+3.3V output (from FTDI)
18	AREF	Input	ADC reference
19-26	A7-A0	Input	Analog input channel 0 to 7
27	+5V	Output or Input	+5V output (from on-board regulator) or +5V (input from external power supply)
30	VIN	PWR	Supply voltage

Nota. Fuente: (Tinajero, 2014)

CAPÍTULO III

IMPLEMENTACIÓN

3.1 Desarrollo

En esta sección se detalla todo el desarrollo del proyecto así como también los pasos que se siguieron para poder instalar y ejecutar los distintos materiales utilizados.

3.1.1 Diseño del circuito en Proteus (Tarjeta Madre)

3.1.1.1 Conexión al Arduino Uno

En la Figura 48, se describe la conexión del Arduino Uno, que utiliza los pines 0, 1, 2, 11, SDA (línea de datos), SCL (línea de reloj), Vin (voltaje de entrada) que es de 5V y por último Gnd que es a tierra (referencia a 0V).

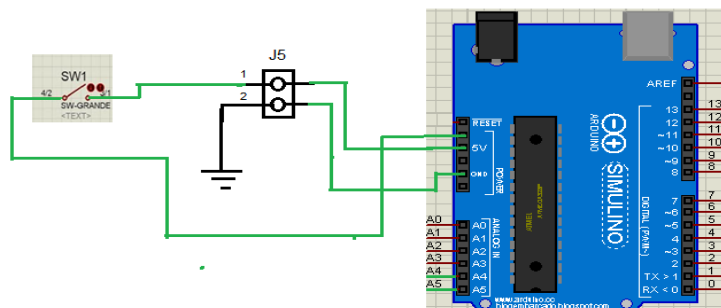


Figura 48. Conexión al Arduino Uno.

3.1.1.2 Conexión del Módulo RTC DS3231

En la Figura 49, se muestra la conexión del módulo RTC DS3231 que es de 6 pines y solo se trabaja con 4 pines. El pin 6 se conecta a tierra, el pin 5 se conecta a los 5V del Arduino Uno, el pin 3 SCL y 4 SDA es para la comunicación serial que tiene con el Arduino.

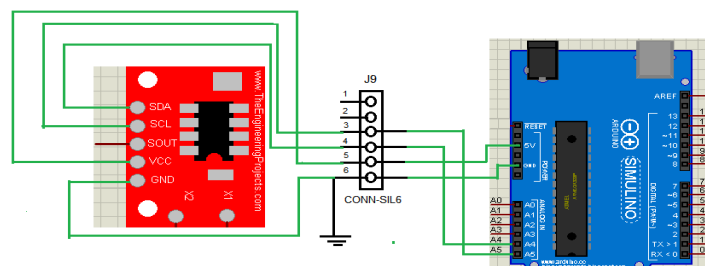


Figura 49. Conexión del Módulo RTC DS3231.

3.1.1.3 Conexión del Transmisor FM

En la Figura 50, se muestra la conexión del transmisor FM, sirve para vincular dispositivos externos cuando se necesite de 5V.

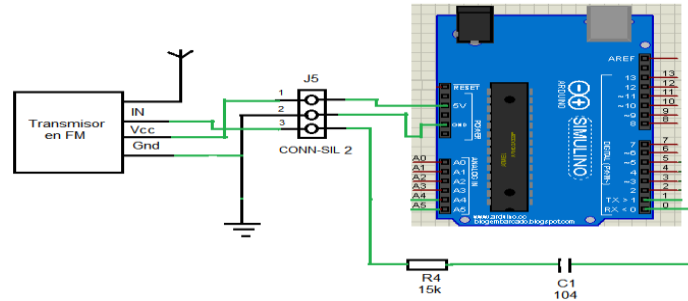


Figura 50. Conexión del Transmisor FM.

3.1.1.4 Conexión del Módulo RF

En la Figura 51, se muestra la conexión del módulo RF. El pin 1 va a tierra, los pines 2 y 3 es la salida de datos del módulo que se conecta directamente al pin 11 del Arduino Uno. Se trabaja con el pin 11 por que la librería está hecha para que ingrese y envíe datos; el módulo receptor siempre va a recibir señales de radio frecuencia de 433MHz.

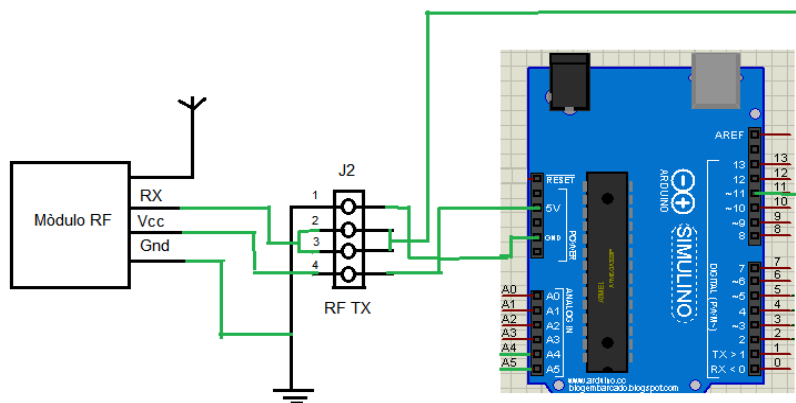


Figura 51. Conexión del Módulo RX.

3.1.1.5 Configuración de la Hora

En la Figura 52, se muestra una configuración de entrada para el Arduino. Se tiene 2 tipos de valores, un alto de 5V cuando no está presionado y una baja de 0V a tierra. El pin 1 está conectado al pin 2 del Arduino Uno, la resistencia de 10kΩ polariza la entrada del Arduino y el pin 2 se coloca a tierra, al presionar el switch nos indica la hora actual.

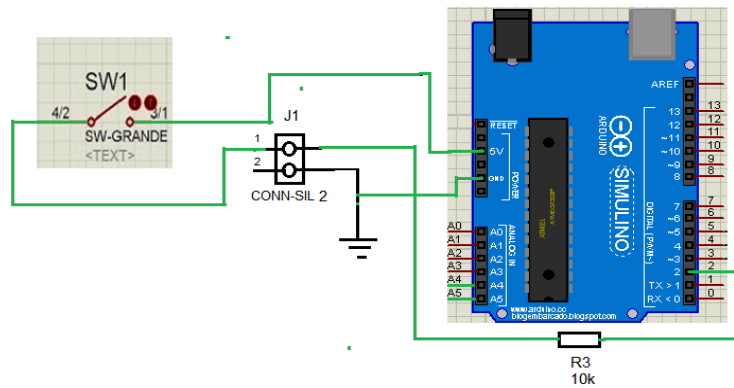


Figura 52. Configuración de la hora.

3.1.1.6 Conexión del Reproductor de voz

En la Figura 53, se muestra la conexión del Reproductor de voz, el pin 1 está conectado a 5V, el pin 2 y 3 están polarizados por dos resistencias de 1k, el pin 1 y 0 es de comunicación serial por hardware donde recibe y envía datos.

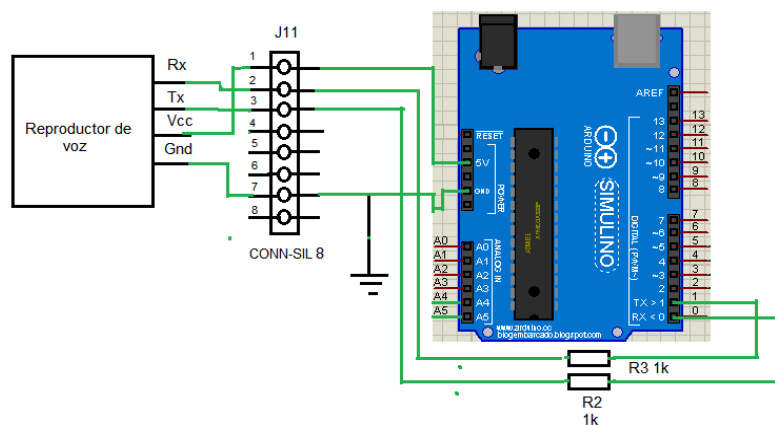


Figura 53. Reproductor de voz.

3.1.2 Diseño del circuito en Proteus (Bastón)

3.1.2.1 Encendido del sistema

En la Figura 54, se muestra el encendido del sistema, el pin 1 se encuentra conectado a 5V del Arduino Nano y el pin 2 a tierra.

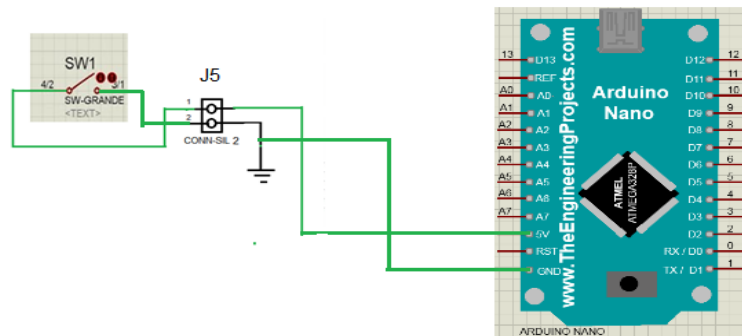


Figura 54. Encendido del equipo mediante un switich.

3.1.2.2 Conexión al Arduino Nano

En la Figura 55, se muestra el Arduino Nano. Es alimentado por medio del cable USB, con una fuente externa no regulada de 6-20V, o con una fuente externa regulada de 5V. La fuente de alimentación es seleccionada automáticamente con mayor tensión.

El chip FTDI FT232RL que posee el Nano solo es energizado si la placa está siendo alimentada usando el cable USB. Cuando se utiliza una fuente externa (no USB), la salida de 3.3V (la cual es proporcionada por el chip FTDI) no está disponible, los pines 1 y 0 parpadearán si los pines digitales 0 o 1 están a nivel alto.

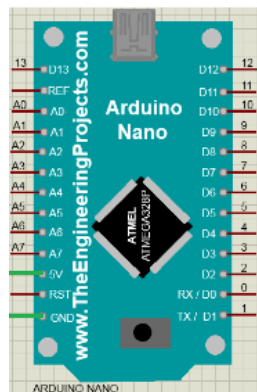


Figura 55. Conexión al Arduino Nano. (Patagoniatec)

3.1.2.3 Conexión del Transmisor RF

En la figura 56, se detalla la conexión del transmisor RF. Primero se alimentan el módulo conectando Vcc y Gnd, respectivamente a 5V y Gnd en Arduino. Se emplea la librería Virtual Wire, que funciona con cualquier pin digital. Por tanto, simplemente conectamos los pines de DATA a una salida digital cualquiera.

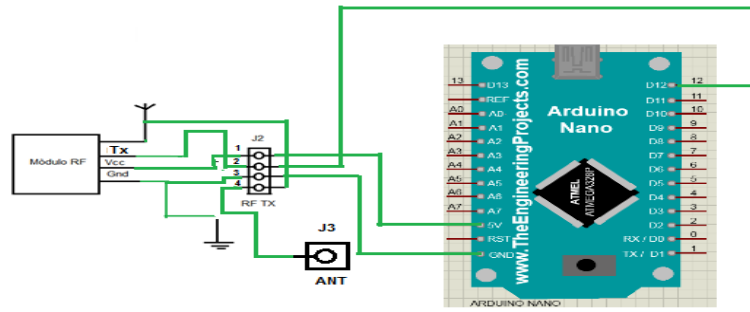


Figura 56. Conexión del Transmisor RF.

3.1.2.4 Conexión del sensor ultrasónico HC-SR04

La Figura 57, muestra la conexión del sensor ultrasónico. El pin 1 se conecta a 5V, los pines 2 y 3 realizan el disparo y recepción de alguna señal y esta a su vez está conectada al pin 9 y 8 del Arduino Nano.

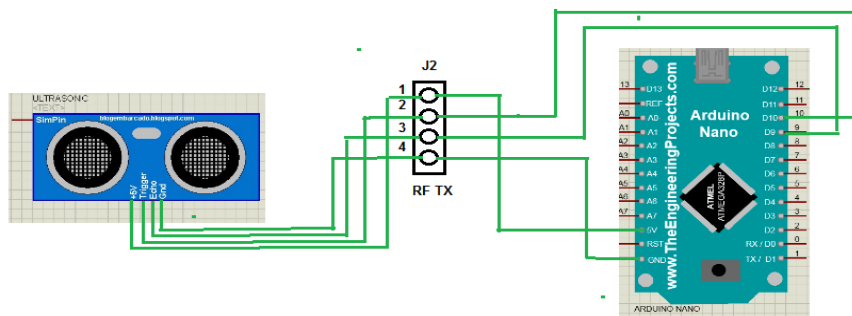


Figura 57. Sensor HC-SR04.

3.1.2.5 Conexión del Sensor de Humedad

En la Figura 58, se muestra la conexión del sensor de humedad. Se conecta tres pines, la tensión de alimentación Vcc que debe ser aproximadamente 5V, el pin de GND y luego los otros dos pines que se utilizará, si es analógico o digital.

- Vcc se conecta a 5V del Arduino.
- GND debe estar conectado al Gnd de Arduino.
- A0 es el pin de datos analógico.
- D0 es el pin de datos digital.

Estas características se dan a la cantidad de agua para la Conversión, la plasticidad basada en los valores analógicos de salida del sensor, bajo consumo de energía, alta sensibilidad, puede directamente conectado a un microprocesador u otro circuito lógico.

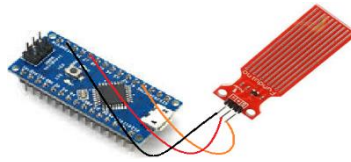


Figura 58. Sensor de Humedad.

3.1.3 Diseño del circuito en Proteus (Gafas)

3.1.3.1 Encendido del sistema

En la Figura 59, muestra el encendido del sistema para las gafas, el pin 1 se conecta a 5V del Arduino Uno y el pin 2 a tierra (referencia a 0V).

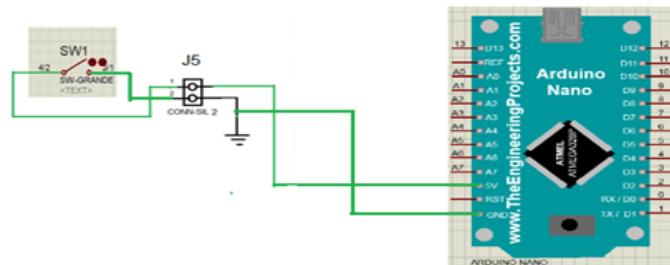


Figura 59. Encendido del equipo mediante un switich.

3.1.3.2 Conexión al Arduino Nano

La Figura 60, muestra la placa Arduino Nano que se basa en ATmega328. Esta carece de jack de alimentación DC y funciona con un cable Mini-B USB en lugar de uno estándar.

Las características de entrada / salida son cada uno de los 14 pines digitales del Nano pueden ser usados como entrada o salida, utilizando las funciones `pinMode()`, `digitalWrite()`, `digitalRead()`. Operan a 5 voltios. Cada pin provee o recibe un máximo de 40mA y posee una resistencia de pull-up (desconectada por defecto) de 20 a 50 KOhms.

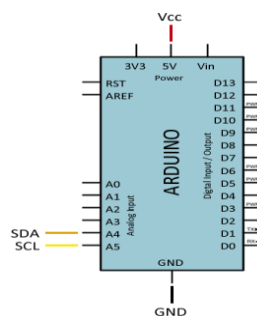


Figura 60. Arduino Nano.

3.1.3.3 Conexión del Transmisor RF

En la figura 61, se detalla la conexión del Transmisor RF. El pin 1 se conecta a 5V, el pin 2 va conectado al pin 12 que es el transmisor del Arduino Nano, el pin 3 va a tierra y el pin 4 conectado a una antena.

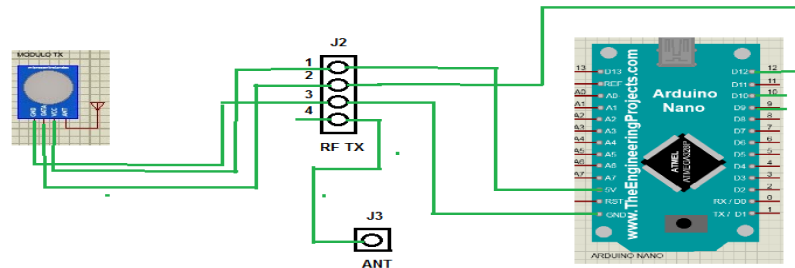


Figura 61. Conexión RF.

3.1.3.4 Conexión de los Sensores ultrasónicos HC-SR04

La Figura 61, muestra la conexión de los sensores ultrasónicos HC-SR04 tiene el pin de alimentación Vcc a +5V y masa 0V o Gnd dos pines que controlan el disparo y el Trig y el echo recibido por la onda (Echo). Para generar una secuencia de disparo debemos activar la señal Trig durante al menos 10 μ s (poner a +5V) cuando se desactiva (poner a 0V) luego se espera a que la señal Echo se active, mide el tiempo transcurrido entre generar el disparo.

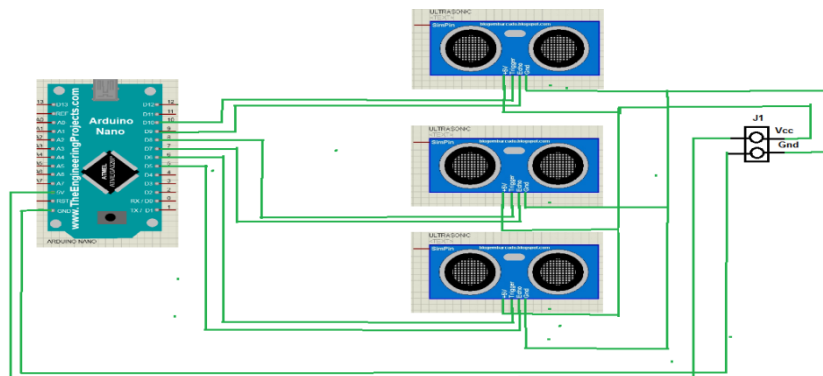


Figura 62. Conexión de Sensores HC-SR04.

3.1.4 Diseño del circuito Impreso

Ya que se realizó el circuito de control en Proteus se procede a realizar en ARES para la impresión, luego a quemar en baquelita y por último a realizar las respectivas perforaciones para colocar los elementos.

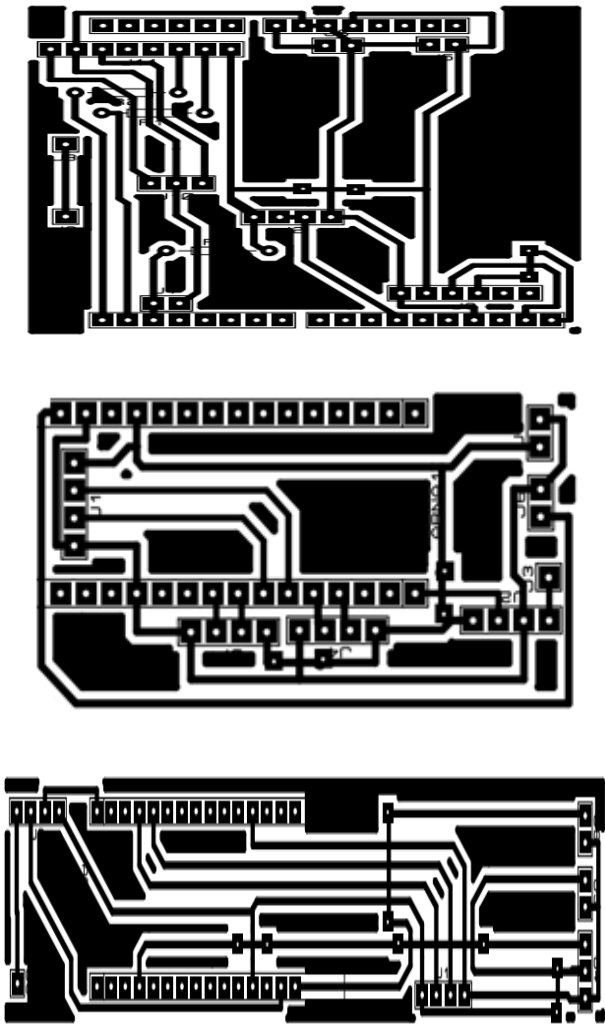


Figura 63. Diseño PCB circuito impreso.

3.2 Construcción

3.2.1 Construcción del circuito impreso e instalación de los componentes

Primeramente se realizan las pistas para luego pasar a la baquelita, se imprime el circuito que se obtiene con el programa ARES (Figura 64).

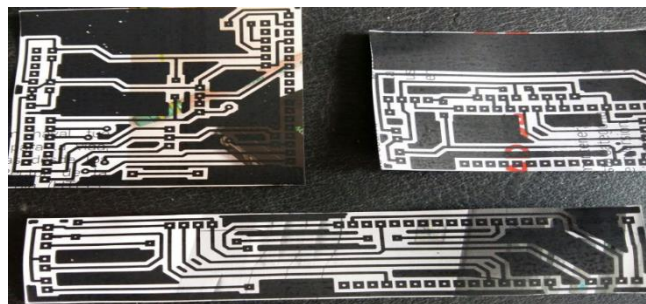


Figura 64. Impresiones de las pistas del circuito de control.

Luego que estén impresas las pistas se procede a imprimir en la baquelita (Figura 65).

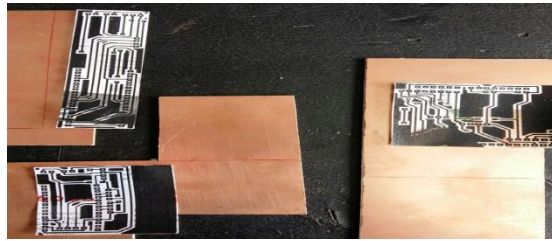


Figura 65. Colocación de la impresión en la baquelita.

Rápidamente se procede a retirar la impresión de la baquelita para decapar con el ácido (Figura 66).



Figura 66. Retiro de la impresión de las pistas.

Ahora se procede a realizar las respectivas perforaciones para la colocación de los elementos (Figura 67).

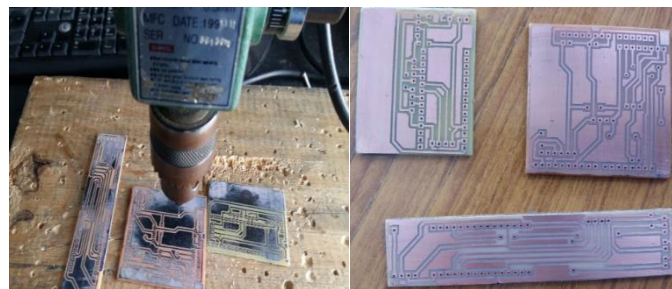


Figura 67. Perforación de baquelita.

Ya que están realizadas las perforaciones correspondientes, procedemos a colocar y soldar los elementos respectivos (Figura 68).

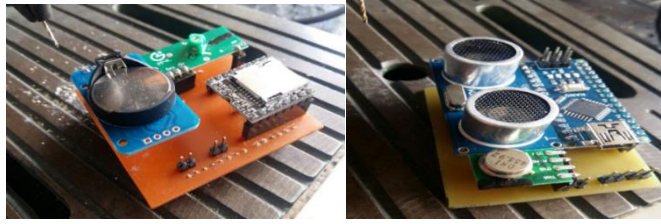


Figura 68. Soldada de elementos.

Ahora se realizó la instalación de un switch a una batería de 5V que será la fuente de alimentación para los 3 dispositivos (Figura 69).

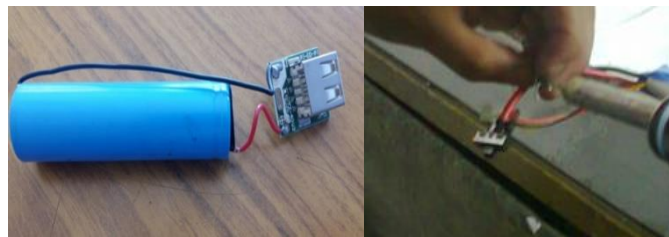


Figura 69. Switch de encendido para dispositivos.

3.2.2 Construcción de cables de extensión para conexión

Se utiliza el cable para hacer las respectivas conexiones desde la baquelita a todos los dispositivos especialmente a los sensores (Figura 70).



Figura 70. Cable ribbon de 4 hilos para conexión.

Luego de haber obtenido el cable que se va a utilizar se procede a utilizar unas regletas macho y hembra (Figura 71).



Figura 71. Regletas macho y hembra.

Una vez soldado el cable con su respectiva regleta que procede a verificar con un multímetro si hay continuidad de punto a punto indicando que está correctamente soldado (Figura 72).

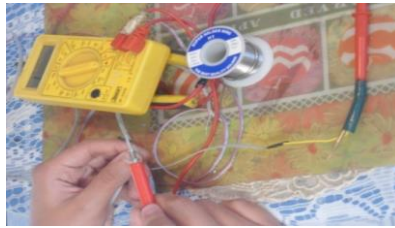


Figura 72. Verificando el cable de punto a punto.

Una vez que están probados los cables, están listos para su respectivo funcionamiento (Figura 73).

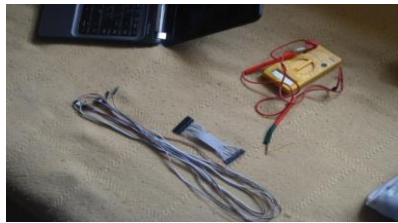


Figura 73. Cables para sensores y elementos.

Luego se realiza una prueba con los terminales de la baquelita al sensor HC-SR04 y un parlante (Figura 74).

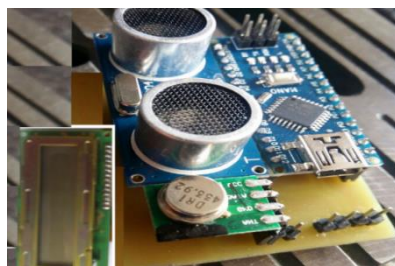


Figura 74. Prueba de funcionamiento con el Lcd.

3.2.3 Construcción de cajas para los circuitos de control

Para la construcción de la caja de la tarjeta madre se coloca el circuito de control y una batería (Figura 75).



Figura 75. Madera.

Se procede a realizar las perforaciones para colocar el enchufe de carga de la batería, el switch para el encendido y del cableado hasta el circuito de control para luego proceder a pulir los residuos (Figura 76).



Figura 76. Realización de perforaciones de la caja.

Y se tiene las cajas completamente terminadas (Figura 77).



Figura 77. Caja terminada.

3.2.4 Utilización de un bastón

Se adquirió un bastón y se realizaron las respectivas modificaciones para el ensamblaje de todos los componentes que van a ser utilizados.



Figura 78. Bastón modificado.

3.2.5 Utilización de una Gafas

Y por último se construyeron unas gafas de madera que fue modificada para el ensamblaje de los componentes.



Figura 79. Gafas terminada.

3.3 Montaje

3.3.1 Colocación del circuito de control en la caja principal

Se procede a colocar en la caja el sistema de control con la batería, el Arduino Uno, Módulo RTC DS3231, Transmisor, Módulo Bluetooth, el reproductor de audio y el módulo GPS. (Figura 80).



Figura 80. Caja principal.

3.3.2 Colocación del circuito de control en las Gafas

Se procede a colocar en la caja el sistema de control, el switch de encendido, la batería, el Arduino Nano, Transmisor y los sensores HC-SR04. (Figura 81).



Figura 81. Caja en gafas.

3.3.3 Colocación del circuito de control en el Bastón

Se procede a colocar en una parte del bastón el sistema de control con el switch de encendido, la batería, el Arduino Nano, Transmisor, sensor HC-SR04, sensor infrarrojo y sensor de agua (Figura 82).



Figura 82. Colocación del circuito en el bastón.

3.4 Pruebas de funcionamiento

3.4.1 Monitoreo

Las gafas para no videntes pueden detectar obstáculos que se encuentran ubicados desde el torso hacia la parte superior. La programación de los sensores cuenta con 2 velocidades que va incrementando al momento de acercarse más al obstáculo, en la Figura 83 se puede apreciar que la persona se encuentra a una distancia de 70 a 120 cm del obstáculo.

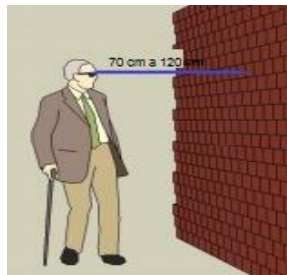


Figura 83. Distancia entre 70 cm a 120 cm

3.4.2 Prueba de los componentes

Primero el sistema prototipo de la tarjeta principal, las gafas, el bastón se enciende por medio de un switch, se emparejan los 3 dispositivos dando una alarma de activación cuando se presenta un obstáculo.

3.4.3 Prueba de funcionamiento SIM 908 GPRS

Al instante en que la persona invidente pulsa el botón de emergencia el sistema prototipo envía un SMS con la ubicación geográfica donde se encuentre que se puede ser monitoreado a través del Smartphone, en la Figura 84 se observa el mensaje enviado como un link para poder visualizar con la aplicación de GOOGLE MAPS.

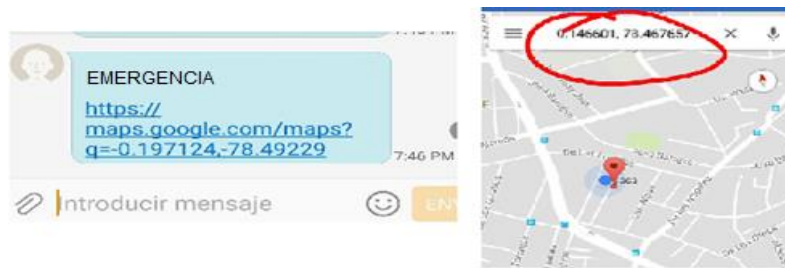


Figura 84. Monitoreo GPS por medio de SMS.

3.5 Análisis de resultados

La realización de este proyecto ha permitido estudiar más a fondo el sistema de proximidad y las aplicaciones que este tiene y que podemos aplicarlas e implementarlas en el área de electrónica. Las pruebas realizadas con los sensores colocados en las gafas y el bastón se las ejecutaron de varias formas:

- Colocar el sensor en la parte inferior del bastón no resultó favorable ya que los sensores detectaban al piso como obstáculo (peligro).
- Colocar el sensor por la mitad del bastón resultó favorable ya que el sensor no detectó el piso como obstáculo.
- Con esto podemos decir que los sensores de proximidad se convierten en los ojos auxiliares de la persona invidente, ya que por medio de estos dispositivos van a activar una alarma que indica que el peligro está cerca.
- Se comprobó las gafas y el bastón electrónico en diferentes lugares (casa, parques, calles, corredores) demostrando de esta manera que es una herramienta confiable para la persona invidente al momento que este circulado por cualquier lugar.

Tabla 8*Sistema de detección de obstáculos en las gafas.*

Número		1		
Prueba		Detección de obstáculos de los 3 sensores		
No.	Acción	Resultado	Intentos	Éxito
1	Detecta Obstáculo localizado mayor a 70 cm y 120 cm en la parte frontal, lateral derecha y lateral izquierda cuando está en movimiento.	Detecta los receptores de los 3 sensores ultrasónicos enviando un mensaje de voz grabada y escuchada en unos audífonos alertando a la persona invidente para que no choque con algún obstáculo.	1	Si
Conclusión:		El prototipo funcionó de la mejor manera.		
Observación:		Para que funcione correctamente, se coloca las gafas en posición correcta.		

Tabla 9*Sistema de detección de obstáculos en el bastón (Sensor)*

Número		1		
Prueba		Detección de obstáculos del sensor		
No.	Acción	Resultado	Intentos	Éxito
1	Detecta Obstáculo localizado mayor a 50 cm y 1 m en la parte frontal cuando está en movimiento.	Detecta el receptor del sensor ultrasónico enviando un mensaje de voz grabada y escuchada en unos audífonos alertando a la persona invidente.	1	Si
Conclusión:		El prototipo funcionó de la mejor manera.		
Observación:		Para que funcione correctamente, se debe colocar el bastón en la posición correcta.		

Tabla 10*Sistema de detección de obstáculos en el bastón (Sensor Sharp)*

Número		1		
Prueba		Detección de obstáculos del Sensor Sharp		
No.	Acción	Resultado	Intentos	Éxito
1	Detecta desniveles localizado a 25 cm (Gradas) y 1,50 m (Huecos) en la parte inferior cuando está en movimiento.	Detecta el receptor del sensor infrarrojo enviando un mensaje de voz grabada y escuchada en unos audífonos alertando a la persona invidente.	1	Si
Conclusión:		El prototipo funcionó de la mejor manera.		
Observación:		Para que funcione correctamente, se coloca el bastón en la posición correcta.		

Tabla 11*Sistema de detección de obstáculos en el bastón (Sensor de Agua)*

Número		1		
Prueba		Detección de obstáculos del Sensor Agua		
No.	Acción	Resultado	Intentos	Éxito
1	Detecta presencia de agua en la parte inferior cuando está en movimiento.	Detecta el sensor de agua enviando un mensaje de voz grabada y escuchada en unos audífonos alertando la presencia de suelo mojado.	1	Si
Conclusión:		El prototipo funcionó de la mejor manera.		
Observación:		Para que funcione correctamente, se debe colocar el bastón en la posición correcta.		

Tabla 12*Sistema de control "Tarjeta Principal"*

Número		1		
Prueba		Sistema de control "Tarjeta Principal"		
No.	Acción	Resultado	Intentos	Éxito
1	Al activarse los componentes de las gafas y del bastón envía una señal al Arduino Uno que comanda todo el sistema.	Esta señal se procede por medio del software y reproduzca la pista grabada dependiendo a la distancia que se encuentre el obstáculo.	1	Si
Conclusión:		El sistema de control funcionó de la mejor manera.		
Observación:		Para que funcione correctamente, se debe programar bien en el software y las grabaciones establecidas.		

Tabla 13*Sistema de control "Hora"*

Número		1		
Prueba		Módulo RTC DS3431 "Hora"		
No.	Acción	Resultado	Intentos	Éxito
1	Al encender el botón de emergencia el módulo se activa.	Este módulo reproduce la hora en cualquier momento.	1	Si
Conclusión:		El módulo de la hora funcionó de la mejor manera.		
Observación:		Para que funcione correctamente, se debe activar el botón de emergencia.		

Tabla 14*Sistema de control "Módulo GPRS SIM900"*

Número		1		
Prueba		Módulo GPRS SIM900		
No.	Acción	Resultado	Intentos	Éxito
1	Al encender el botón nos envía un SMS a un teléfono móvil con la dirección que se encuentra la persona invidente.	Este módulo nos ayuda a ubicar la dirección correcta de la persona que va a utilizar el prototipo.	1	Si
Conclusión:		El módulo de ubicación funcionó de la mejor manera.		
Observación:		Para que funcione correctamente, se conectar la antena de la mejor manera.		

Mediante estas tablas podemos ver algunas pruebas realizadas con las gafas y el bastón electrónico, con lo que logramos asegurar que los sensores y componentes poseen una buena respuesta y utilizar sin ningún problema para la persona invidente se va a convertir en una herramienta para mayor seguridad.

CONCLUSIONES

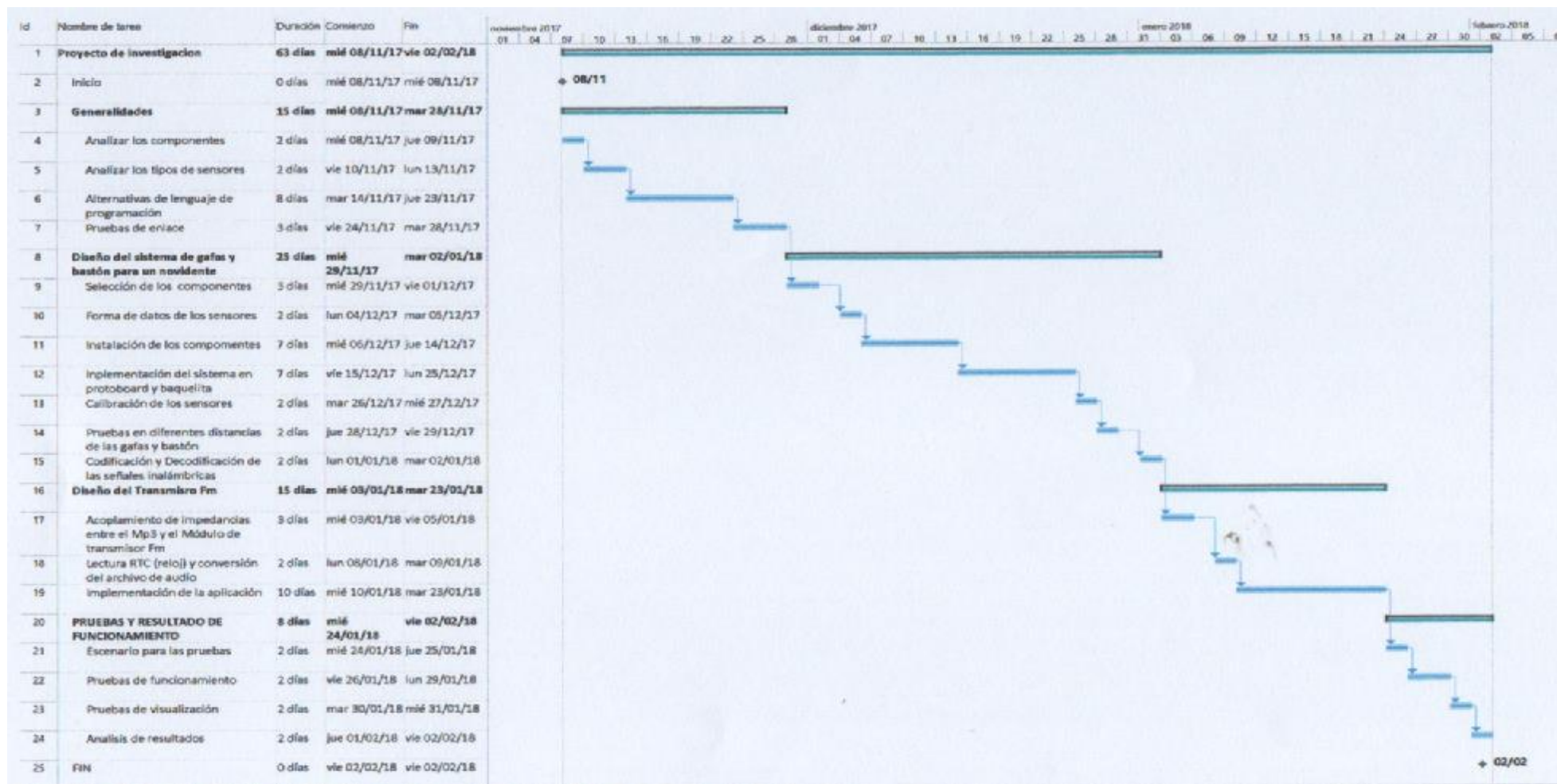
- El requerimiento del dispositivo de detección de objeto brinda una gran libertad en desplazamiento de personas con este tipo de discapacidad.
- Para la estructura del prototipo electrónico adaptable a las gafas y un bastón, se lograron cumplir los objetivos trazados al inicio de este proyecto investigativo, pero además, se lograron alcanzar logros importantes como la utilización de Bluetooth, para ingresar parámetros del usuario; descripción de zonas de seguridad.
- Se consiguió interconectar y manejar los diferentes entornos que se usaron para la realización del proyecto, tanto la parte electrónica con la del software.
- La programación para la detección de obstáculos por medio del sensor, resulto satisfactorio. Las partes utilizadas para la fabricación del prototipo fueron materiales que encontramos disponibles a la mano, tener la capacidad de importar algunos componentes o buscar diferente maneras de fabricarlos abaratarían los costos y mejorarían los puntos críticos de estabilidad del proyecto.
- El implementar tecnologías de distintas formas a las personas que sufren de ceguera, es muy delicado, porque ellos esperan más que un producto que le sea de ayuda. Para ellos el aprender a convivir a diario con su padecimiento, es de vital importancia.
- El dispositivo ha sido correctamente probado en condiciones reales, donde demostró tener un funcionamiento congruente con los objetivos planteados ya que detecto de manera satisfactoria los obstáculos próximos en el andar de la persona calles y caminos de una población.

RECOMENDACIONES

- Para el desarrollo de la estructura física del bastón y las gafas se debe investigar tipos de sensores y elementos a colocarse que no tengan ninguna interferencia entre ellos.
- Se debe efectuar versiones y modelos de carcasa futuras con la intervención de personas con discapacidad visual para recibir recomendaciones de mejoras
- La combinación de gafas y bastón proporciona un mejor aviso a posibles peligros que se encuentra en el trayecto.
- Analizar la posibilidad de aumentar el rango de mediciones para que la persona con discapacidad visual tenga un mayor conocimiento del lugar donde se encuentre, usando otro tipo de sensores que mejoren el alcance.
- La combinación de las gafas y el bastón proporciona un mejor aviso a posibles peligros que se encuentra en el trayecto.

CRONOGRAMA DE ACTIVIDADES

Tabla 15: Actividades para presentación del proyecto.



EVALUACION DE LA PROPUESTA

La evaluación económica en la implementación de las gafas y el bastón electrónico, fue financiada en su totalidad por el investigador.

Tabla 16

Evaluación económica

Ítem	Rubro de Gastos	Cantidad	Valor Unitario	Valor Total
1	Arduino Uno	2	\$ 13,00	\$ 26,00
2	Arduino Nano	2	\$ 11,00	\$ 22,00
3	Audífono Bluetooth	1	\$ 20,00	\$ 20,00
4	Cable	1	\$ 7,00	\$ 7,00
5	Conectores macho y hembra	2	\$ 3,00	\$ 6,00
6	Bastón Metálico	1	\$ 22,00	\$ 22,00
7	Cajas de Madera	3	\$ 12,00	\$ 36,00
8	Fotocopias	3	\$ 10,00	\$ 30,00
9	Gafas de Madera	1	\$ 50,00	\$ 50,00
10	Módulo Bluetooth	1	\$ 50,00	\$ 50,00
11	Módulo de reproductor Mp3	1	\$ 9,00	\$ 9,00
12	Módulo GSM/GPRS SIM 900	1	\$ 45,00	\$ 45,00
13	Módulo RTC DS 3231	1	\$ 8,50	\$ 8,50
14	Resistencias	2	\$ 0,05	\$ 0,10
15	Sensor de Humedad	1	\$ 8,00	\$ 8,00
16	Sensor Sharp	1	\$ 14,00	\$ 14,00
17	Sensores HC-SR04	4	\$ 6,00	\$ 24,00
18	Switch	4	\$ 0,90	\$ 3,60
19	Tarjeta de memoria 4G	1	\$ 9,00	\$ 9,00
20	Transmisores de RF	3	\$ 7,00	\$ 21,00
Sub total				\$ 411,20
Precio con IVA (12%)				\$ 49,344
Total:				\$ 460,54

REFERENCIAS BIBLIOGRÁFICAS

Aragunde, I. (25 de 05 de 2016). bastón. Accesibilidad.

Arduino. (27 de Junio de 2014). Arduino Mega 2560. Recuperado el 20 de Julio de 2017, de <http://arduino.cc/en/Main/ArduinoBoardMega2560>

Arduino. (s.f.). Arduino.

Arduino. (s.f.). Electrónica. Electrónica.

ARDUINO MEGA 2560. (2017). Obtenido de ARDUINO: <https://www.arduino.cc/en/Main/arduinoBoardMega2560>

Berdahl, J. (2016). Diagnostico y Tratamiento. All About Vision.

Cambre, A. (12 de 09 de 2013). Ana María. Recuperado el 04 de 07 de 2014, de <http://cambreanamaria.blogspot.com/2013/09/pedagogia-especial-frente-las.html>

Cienciamisterio. (4 de 2007). <http://www.cienciamisterio.com>. Recuperado el 09 de 07 de 2014, de <http://www.cienciamisterio.com/noticienab07.html>

CONADIS. (2013). <http://www.conadis.gob.ec>. Recuperado el 09 de 07 de 2014, de <http://www.conadis.gob.ec/investigacion04.htm#2004>

Conadis. (05 de 2013). <http://www.consejodiscapacidades.gob.ec>. Recuperado el 04 de 07 de 2014, de http://www.consejodiscapacidades.gob.ec/wp-content/uploads/downloads/2014/03/conadis_registro_nacional_discapacidades.pdf

Crespo, E. (25 de Septiembre de 2016). Aprendiendo Arduino. Recuperado el 27 de Julio de 2017, de <https://aprendiendoarduino.wordpress.com/category/software/>

Domínguez, M. (13 de 9 de 2015). Sharp.

Elcajondeardu. (20 de Septiembre de 2015). Guía básica de utilización de sensor de sonido con Arduino . Recuperado el 20 de Julio de 2017, de <http://elcajondeardu.blogspot.com/2015/09/tutorial-sensor-de-sonido.html>

Encrypted. (2006). <https://encrypted-tbn0.gstatic.com>. Recuperado el 04 de 07 de 2014, de https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcSp1K-X3LZ6TCsbcT6tTCd798_WneezMZQsBOi6A5b_tyJPblrT

Galarza Jacome. (2008). bibdigital.epn.edu.ec. Recuperado el 04 de 07 de 2014, de bibdigital.epn.edu.ec

Impresionante. (2005). <http://www.impresionante.net>. Recuperado el 09 de 07 de 2014, de <http://www.impresionante.net/general/una-vara-para-ciegos/>

Itead. (12 de Febrero de 2017). SIM900 GSM/GPRS SHIELD ICOSAT PREVIEW. Recuperado el 5 de Julio de 2017, de <https://www.itead.cc/blog/sim900-gprs-shield-icosat-preview>

Lopez, I. (2015). Revista Medica Clinica Los Andes. CLC, 42-51.

Oftalmo Ver. (2016). Obtenido de <http://oftalmover.co/>

OMS. (10 de 2013). <http://www.who.int/mediacentre/factsheets/fs282/es/>. Recuperado el 04 de 07 de 2014

Ounae. (2010). <http://ounae.com>. Recuperado el 09 de 07 de 2014, de <http://ounae.com/gafas-con-laser-para-personas-invidentes/>

Patagoniatec. (s.f.). Electronica. Electronica.

Picmanianet. (2010). <http://picmania.garcia-cuervo.net>. Recuperado el 04 de 07 de 2014, de http://picmania.garcia-cuervo.net/recursos/redpictutorials/sensores/sensores_de_distancias_con_ultrasonidos.pdf

Prometec. (2015). Electrónica. Electrónica.

Quintero, L. (2015). Discapacidad Visual.

Rodríguez, C. (2016). Diseño e implementación de un sistema de control de acceso domiciliaria vía SMS por celular. Sangolqui - Ecuador.

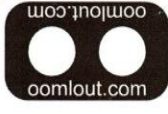
RunaTech. (05 de 2014). <http://www.fayerwayer.com>. Recuperado el 09 de 07 de 2014, de <http://www.fayerwayer.com/2014/05/runatech-el-traje-para-invidentes-desarrollado-en-ecuador/>

Sandisk, D. S. (2012). Micro SD Data Sheet. N/E: APR.


Sharp. (s.f.). Sensor.

- Store. (30 de Abril de 2017). ARDUINO MEGA 2560 REV3. Recuperado el 30 de Julio de 2017, de <https://store.arduino.cc/usa/arduino-mega-2560-rev3>
- Tecnologia-informatica. (07 de Julio de 2014). Control remoto con Arduiono. Recuperado el 30 de Julio de 2017, de <http://www.tecnologia-informatica.es/control-remoto-con-arduiono/>
- Teslabem. (3 de Marzo de 2017). Módulo sensor de sonido con micrófono electret 10mm. Recuperado el 20 de Julio de 2017, de <http://teslabem.com/modulo-sensor-de-sonido-con-microfono-electret.html>
- Tinajero, J. L. (2014). 108T0115 Evaluación de la eficiencia de los controladores Arduino Mega y Seimens Logo 23ORC en procesos industriales. Obtenido de Repositorio Institucional de la Escuela Superior Politécnica de Chimborazo: <dspace.espoch.edu.ec/bitstream/123456789/3805/1/108T0115.pdf>
- Transtornos. (2016). *OMS*.
- UltraMed. (12 de 12 de 2013). <https://es-es.facebook.com/UltraMedPrestacionesMedicas>. Recuperado el 04 de 07 de 2014, de <https://es-es.facebook.com/UltraMedPrestacionesMedicas/posts/455940384443755>
- Universidad de Girona. (2009). <http://www.webconsultas.com>. Recuperado el 04 de 07 de 2014, de <http://www.webconsultas.com/cataratas/cataratas-455>
- Universidad Maimonides. (1 de 10 de 2009). <http://weblog.maimonides.edu>. Recuperado el 09 de 07 de 2014, de <http://weblog.maimonides.edu/gerontologia2007/2009/10/>
- Webantics. (4 de Febrero de 2017). Huawei E3533 21.6MBPS HSPA+ 3G USB Modem. Recuperado el 26 de Julio de 2017, de <https://www.webantics.com/huawei-e3533-21mbps-hspa-3g-usb-modem>
- Ximénez, P. (30 de Julio de 2015). La Gualateca. La Gualateca. Recuperado el 7 de Septiembre de 2016, de Aula de Prensa: <https://auladeprensa.wordpress.com/2012/07/30/jose-maria-fallotico-inventor-del-baston-blanco-para-ciegos/>


DATASHEET DEL SENSOR INFLARROJO




Sensing Distance (SHARP GP2Y0A21YK)



The Pieces



IR Distance Sensor
(Sharp GP2Y0A21Y)
x1



Cable
(JST 3 pole)
x1

The Theory & Code

Infra Red Distance Sensor
An analog IR distance sensor is a really neat component. Simply connect +5v and ground and a voltage proportional to the distance between the sensor and an object in front of it will be returned (ranging from 0.4 volts at 80 cm to 3 volts at 10 cm). It really couldn't be simpler to use.

Converting to Distance
The voltage returned is not linear (see graph), however it can be converted to a distance using some simple maths. There will be some variation between models but here are a couple of equations that work for us (results +-~5%)

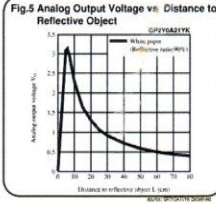


Fig.5 Analog Output Voltage vs. Distance to Reflective Object

Equations

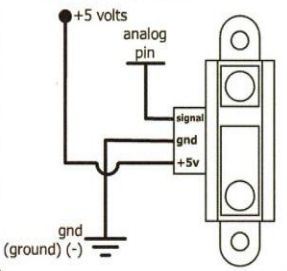
From 10 bit A/D Reading to centimeters (5v supply)
 $distance = 12343.85 * (10bit\ reading)^{-1.15}$

From Voltage Reading to Centimeters
 $distance = 27.86 (voltage\ reading)^{-1.15}$

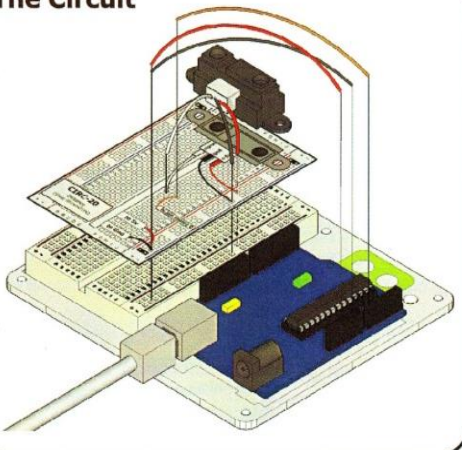
Arduino Code (5v)
 $float\ distance = 12343.85 * pow(analogRead(sensorPin), -1.15)$

Technical Details
 :: Full Datasheet: <http://tinyurl.com/yh7chku> ::

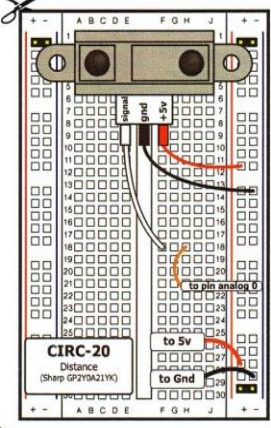
The Schematic



The Circuit



The Layout Sheet



.. **Instructions:** print out, cut out, get making ..
 .. for more details visit: <http://tinyurl.com/y9bjsjp> ..

FEATURES

- Digital Output
- LED Pulse Cycle Duration: 32 ms
- Range: 10 to 80 cm
- Typical response time: 39 ms
- Typical start up delay: 44 ms
- Average Current Consumption: 30 mA
- Detection Area Diameter @ 80 cm: 12 cm

DESCRIPTION

The GP2Y0A21YK is a wide-angle distance measuring sensor.

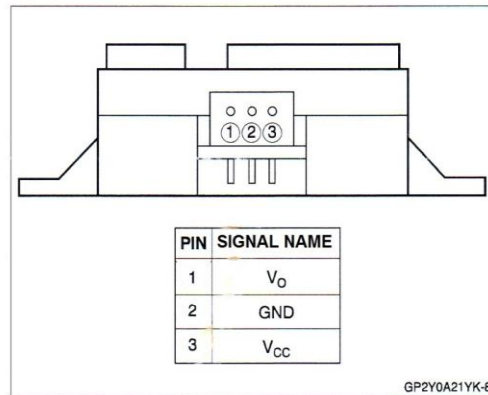


Figure 1. Pinout

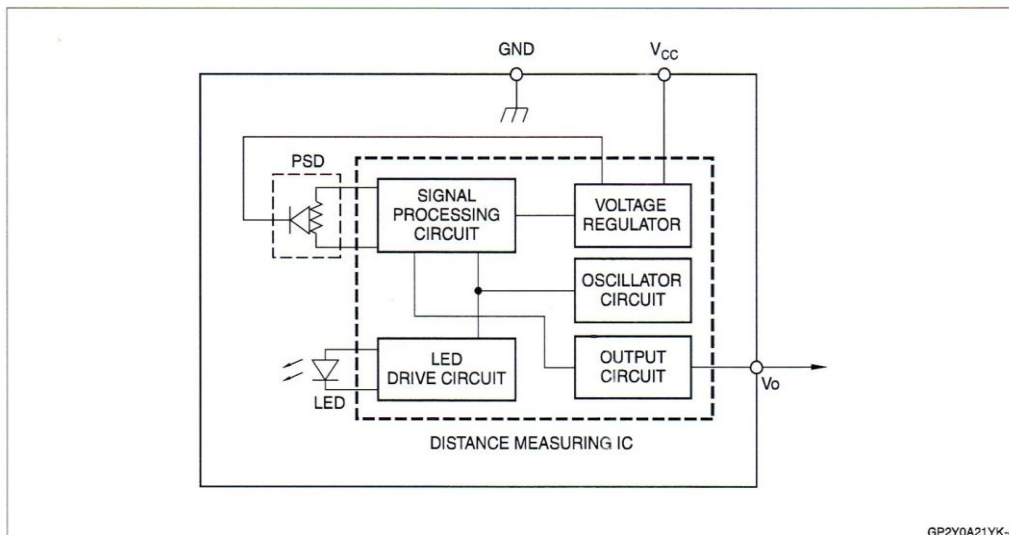


Figure 2. Block Diagram

ELECTRICAL SPECIFICATIONS

Absolute Maximum Ratings

Ta = 25°C, V_{CC} = 5 VDC

PARAMETER	SYMBOL	RATING	UNIT
Supply Voltage	V _{CC}	-0.3 to +7	V
Output Terminal Voltage	V _O	-0.3 to (V _{CC} + 0.3)	V
Operating Temperature	T _{opr}	-10 to +60	°C
Storage Temperature	T _{stg}	-40 to +70	°C

Operating Supply Voltage

PARAMETER	SYMBOL	RATING	UNIT
Operating Supply Voltage	V _{CC}	4.5 to 5.5	V

Electro-optical Characteristics

Ta = 25°C, V_{CC} = 5 VDC

PARAMETER	SYMBOL	CONDITIONS	MIN.	TYP.	MAX.	UNIT	NOTES
Measuring Distance Range	ΔL		10	-	80	cm	1, 2
Output Terminal Voltage	V _O	L = 80 cm	0.25	0.4	0.55	V	1, 2
Output Voltage Difference	ΔV _O	Output change at ΔL (80 cm - 10 cm)	1.65	1.9	2.15	V	1, 2
Average Supply Current	I _{CC}	L = 80 cm	-	30	40	mA	1, 2

NOTES:

- Using reflective object: white paper (made by Kodak Co. Ltd.) gray cards R-27, white face, reflective ratio: 90%
- L = Distance to reflective object.

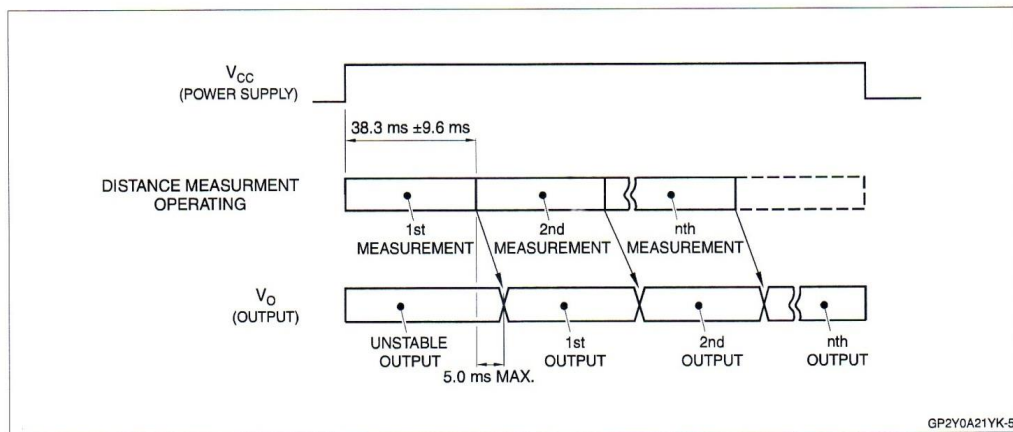
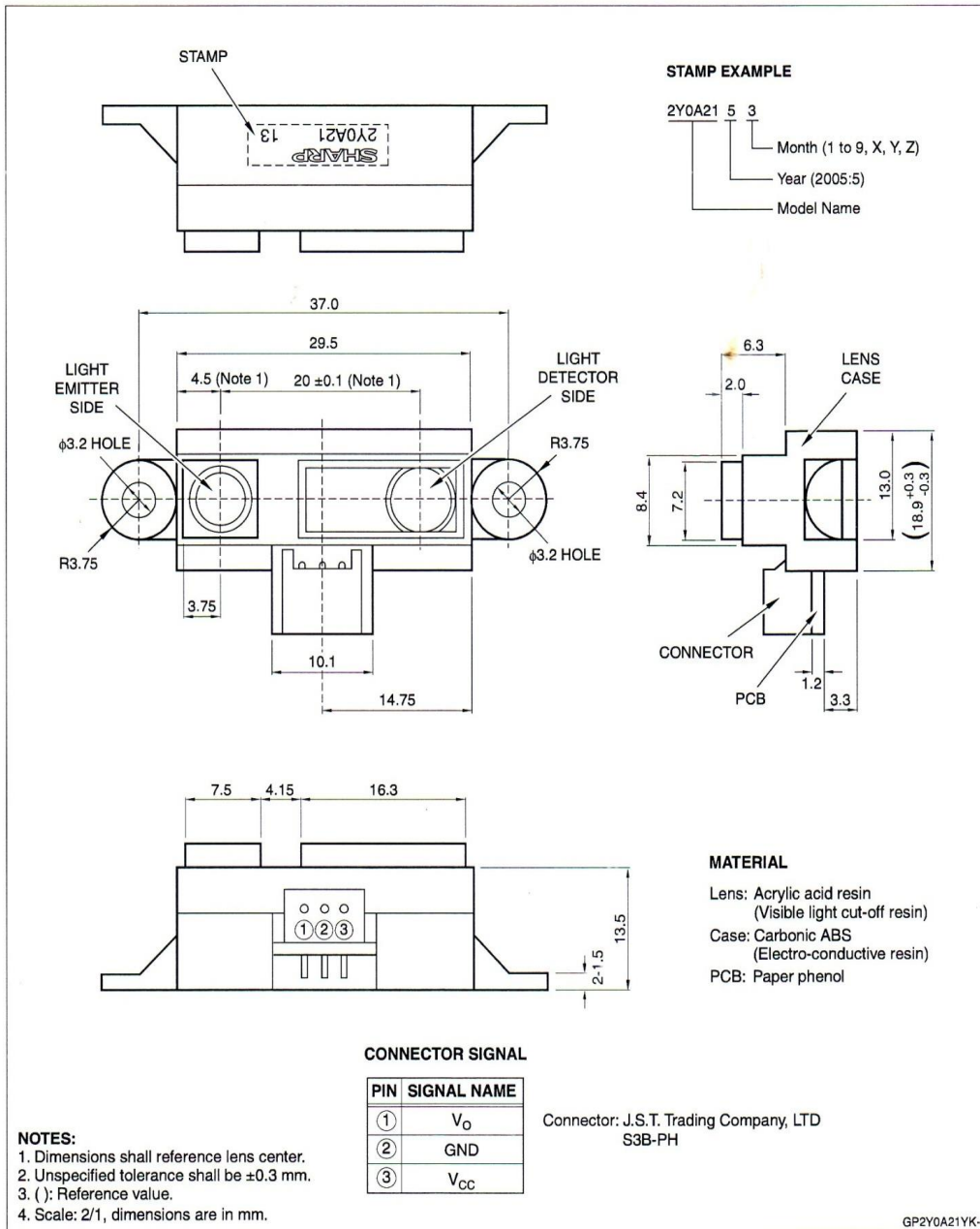


Figure 3. Timing Diagram

PACKAGE SPECIFICATIONS





Arduino Uno SMD



Overview

The Arduino Uno SMD is a version of the [Arduino Uno](#), but uses an surface mount version of the Atmega328P instead of the through-hole version. This version was made in response to a shortage in supply of the through-hole Atmega328P. The board is based on the ATmega328 ([datasheet](#)). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.

The Uno differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega8U2 programmed as a USB-to-serial converter. "Uno" means one in Italian and is named to mark the upcoming release of Arduino 1.0. The Uno and version 1.0 will be the reference versions of Arduino, moving forward. The Uno is the latest in a series of USB Arduino boards, and the reference model for the Arduino platform; for a comparison with previous versions, see the [index of Arduino boards](#).

Summary

Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328) of which 0.5 KB used by bootloader

SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Clock Speed	16 MHz

Schematic & Reference Design

EAGLE files: [arduino-uno-smd-reference-design.zip](#)

Schematic: [arduino-uno-smd-schematic.pdf](#)

Power

The Arduino Uno can be powered via the USB connection or with an external power supply. The power source is selected automatically.

External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector.

The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

The power pins are as follows:

- **VIN.** The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
- **5V.** The regulated power supply used to power the microcontroller and other components on the board. This can come either from VIN via an on-board regulator, or be supplied by USB or another regulated 5V supply.
- **3V3.** A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
- **GND.** Ground pins.

Memory

The ATmega328 has 32 KB (with 0.5 KB used for the bootloader). It also has 2 KB of SRAM and 1 KB of EEPROM (which can be read and written with the [EEPROM library](#)).

Input and Output

Each of the 14 digital pins on the Uno can be used as an input or output, using [pinMode\(\)](#), [digitalWrite\(\)](#), and [digitalRead\(\)](#) functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:

- **Serial: 0 (RX) and 1 (TX).** Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip.

- **PWM: 3, 5, 6, 9, 10, and 11.** Provide 8-bit PWM output with the `analogWrite()` function.
- **SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK).** These pins support SPI communication using the `SPI` library.
- **LED: 13.** There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.

The Uno has 6 analog inputs, labeled A0 through A5, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though it is possible to change the upper end of their range using the AREF pin and the `analogReference()` function. Additionally, some pins have specialized functionality:

- **I²C: A4 (SDA) and A5 (SCL).** Support I²C (TWI) communication using the `Wire` library.

There are a couple of other pins on the board:

- **AREF.** Reference voltage (0 to 5V only) for the analog inputs. Used with `analogReference()`.
- **Reset.** Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

See also the [mapping between Arduino pins and ATmega328 ports](#).

Communication

The Arduino Uno has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega328 provides UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). An ATmega8U2 on the board channels this serial communication over USB and appears as a virtual com port to software on the computer. The '8U2 firmware uses the standard USB COM drivers, and no external driver is needed. However, *on Windows, a .inf file is required*. The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board. The RX and TX LEDs on the board will flash when data is being transmitted via the USB-to-serial chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

A `SoftwareSerial` library allows for serial communication on any of the Uno's digital pins. The ATmega328 also supports I2C (TWI) and SPI communication. The Arduino software includes a `Wire` library to simplify use of the I2C bus; see the [documentation](#) for details. For SPI communication, use the `SPI` library.

Programming

The Arduino Uno can be programmed with the Arduino software ([download](#)). Select "Arduino Uno" from the Tools > Board menu (according to the microcontroller on your board). For details, see the [reference](#) and [tutorials](#).

The ATmega328 on the Arduino Uno comes preburned with a [bootloader](#) that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol ([reference](#), [C header files](#)). You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header; see [these instructions](#) for details. The ATmega8U2 firmware source code is available . The ATmega8U2 is loaded with a DFU bootloader. The Uno SMD has a pulldown resistor tying the HWB pin to ground, so all that's needed to enter DFU mode is to briefly short enough to short pins 5 and 6 of the 8U2 icsp connector. This will connect the 8U2 reset pin to ground. You can then use [Atmel's FLIP software](#) (Windows) or the [DFU programmer](#) (Mac OS X and Linux) to load a new firmware. Or you can use the ISP header with an external programmer (overwriting the DFU bootloader). See [this user-contributed tutorial](#) for more information.

Automatic (Software) Reset

Rather than requiring a physical press of the reset button before an upload, the Arduino Uno is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the ATmega8U2 is connected to the reset line of the ATmega328 via a 100 nanofarad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino software uses this capability to allow you to upload code by simply pressing the upload button in the Arduino environment. This means that the bootloader can have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload.

This setup has other implications. When the Uno is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the bootloader is running on the Uno. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened. If a sketch running on the board receives one-time configuration or other data when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data.

The Uno contains a trace that can be cut to disable the auto-reset. The pads on either side of the trace can be soldered together to re-enable it. It's labeled "RESET-EN". You may also be able to disable the auto-reset by connecting a 110 ohm resistor from 5V to the reset line; see [this forum thread](#) for details.

USB Overcurrent Protection

The Arduino Uno has a resettable polyfuse that protects your computer's USB ports from shorts and overcurrent. Although most computers provide their own internal protection, the fuse provides an extra layer of protection. If more than 500 mA is applied to the USB port, the fuse will automatically break the connection until the short or overload is removed.

Physical Characteristics

The maximum length and width of the Uno PCB are 2.7 and 2.1 inches respectively, with the USB connector and power jack extending beyond the former dimension. Four screw holes allow the board to be attached to a surface or case. Note that the distance between digital pins 7 and 8 is 160 mil (0.16"), not an even multiple of the 100 mil spacing of the other pins.

Ultrasonic Ranging Module HC - SR04

Product features:

Ultrasonic ranging module HC - SR04 provides 2cm - 400cm non-contact measurement function, the ranging accuracy can reach to 3mm. The modules includes ultrasonic transmitters, receiver and control circuit. The basic principle of work:

- (1) Using IO trigger for at least 10us high level signal,
- (2) The Module automatically sends eight 40 kHz and detect whether there is a pulse signal back.
- (3) IF the signal back, through high level , time of high output IO duration is the time from sending ultrasonic to returning.

Test distance = (high level time×velocity of sound (340M/S) / 2,

Wire connecting direct as following:

- 5V Supply
- Trigger Pulse Input
- Echo Pulse Output
- 0V Ground

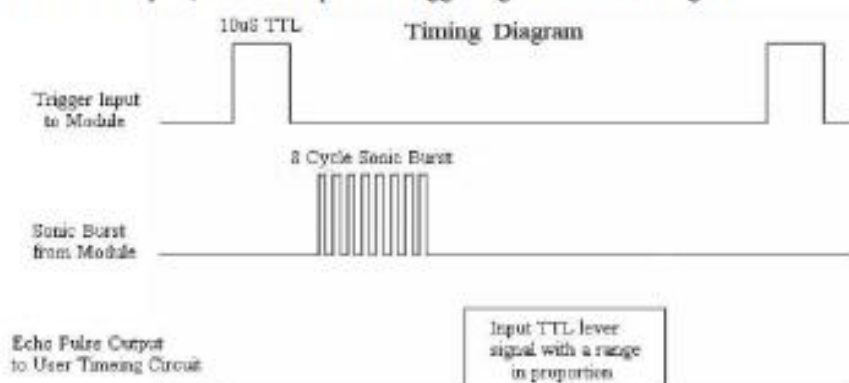
Electric Parameter

Working Voltage	DC 5 V
Working Current	15mA
Working Frequency	40Hz
Max Range	4m
Min Range	2cm
MeasuringAngle	15 degree
Trigger Input Signal	10uS TTL pulse
Echo Output Signal	Input TTL. lever signal and the range in proportion
Dimension	45*20*15mm



Timing diagram

The Timing diagram is shown below. You only need to supply a short 10 μ S pulse to the trigger input to start the ranging, and then the module will send out an 8 cycle burst of ultrasound at 40 kHz and raise its echo. The Echo is a distance object that is pulse width and the range in proportion. You can calculate the range through the time interval between sending trigger signal and receiving echo signal. Formula: $\mu\text{S} / 58 = \text{centimeters}$ or $\mu\text{S} / 148 = \text{inch}$; or: the range = high level time * velocity (340M/S) / 2; we suggest to use over 60ms measurement cycle, in order to prevent trigger signal to the echo signal.



SIM900 GSM/GPRS Module



The SIM900 is a complete Quad-band GSM/GPRS solution in a SMT module which can be embedded in the customer applications.

Featuring an industry-standard interface, the SIM900 delivers GSM/GPRS 850/900/1800/1900MHz performance for voice, SMS, Data, and Fax in a small form factor and with low power consumption. With a tiny configuration of 24mm x 24mm x 3 mm, SIM900 can fit almost all the space requirements in your M2M application, especially for slim and compact demand of design.

- SIM900 is designed with a very powerful single-chip processor integrating AMR926EJ-S core
- Quad - band GSM/GPRS module with a size of 24mmx24mmx3mm
- SMT type suit for customer application
- An embedded Powerful TCP/IP protocol stack
- Based upon mature and field-proven platform, backed up by our support service, from definition to design and production

SIM900

The GSM/GPRS Module for M2M applications

General features

- Quad-Band 850/ 900/ 1800/ 1900 MHz
- GPRS multi-slot class 10/8
- GPRS mobile station class B
- Compliant to GSM phase 2/2+
 - Class 4 (2 W @850/ 900 MHz)
 - Class 1 (1 W @ 1800/1900MHz)
- Dimensions: 24* 24 * 3 mm
- Weight: 3.4g
- Control via AT commands (GSM 07.07 ,07.05 and SIMCOM enhanced AT Commands)
- SIM application toolkit
- Supply voltage range 3.4 ... 4.5 V
- Low power consumption
- Operation temperature: -30 °C to +80 °C

Specifications for fax

- Group 3, class 1

Specifications for data

- GPRS class 10: max. 85.6 kbps (downlink)
- PBCCH support
- Coding schemes CS 1, 2, 3, 4
- CSD up to 14.4 kbps
- USSD
- Non transparent mode
- PPP-stack

Specifications for SMS via GSM Pin Assignment

/ GPRS

- Point-to-point MO and MT
- SMS cell broadcast
- Text and PDU mode

Drivers

- MUX Driver

Specifications for voice

- Tricodec
 - Half rate (HR)
 - Full rate (FR)
 - Enhanced Full rate (EFR)

- Hands-free operation (Echo suppression)
- AMR
 - Half Rate(HR)
 - Full Rate(FR)

Interfaces

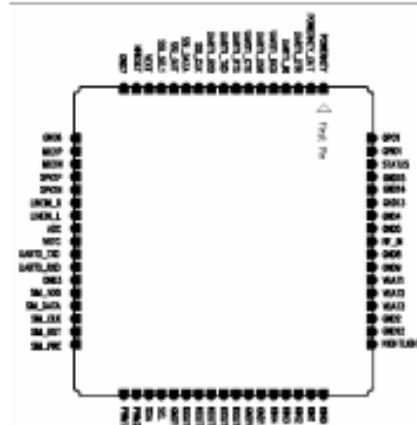
- Interface to external SIM 3V/ 1.8V
- analog audio interface
- RTC backup
- SPI interface
- Serial interface
- Antenna pad
- I2C
- GPIO
- PWM
- ADC

Compatibility

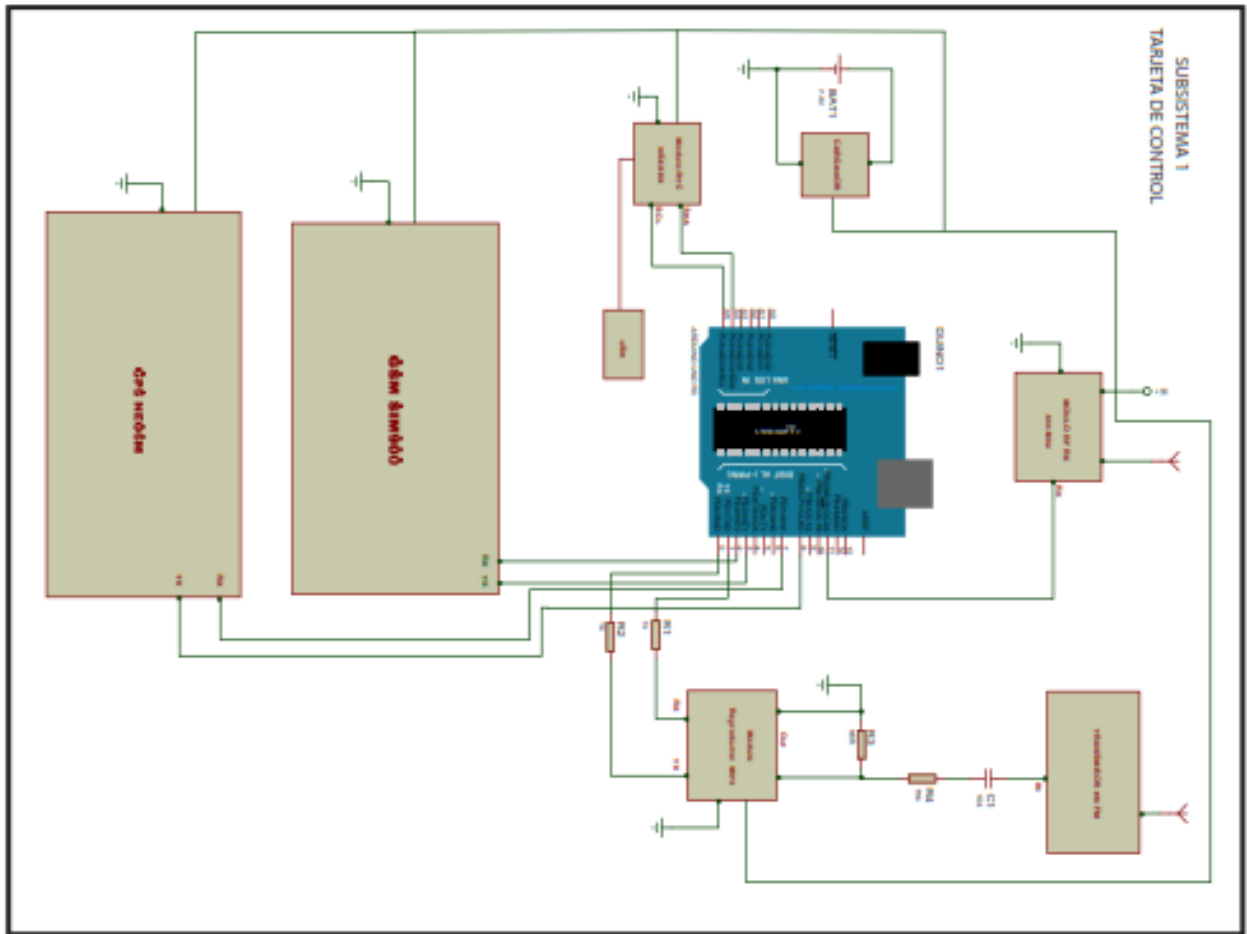
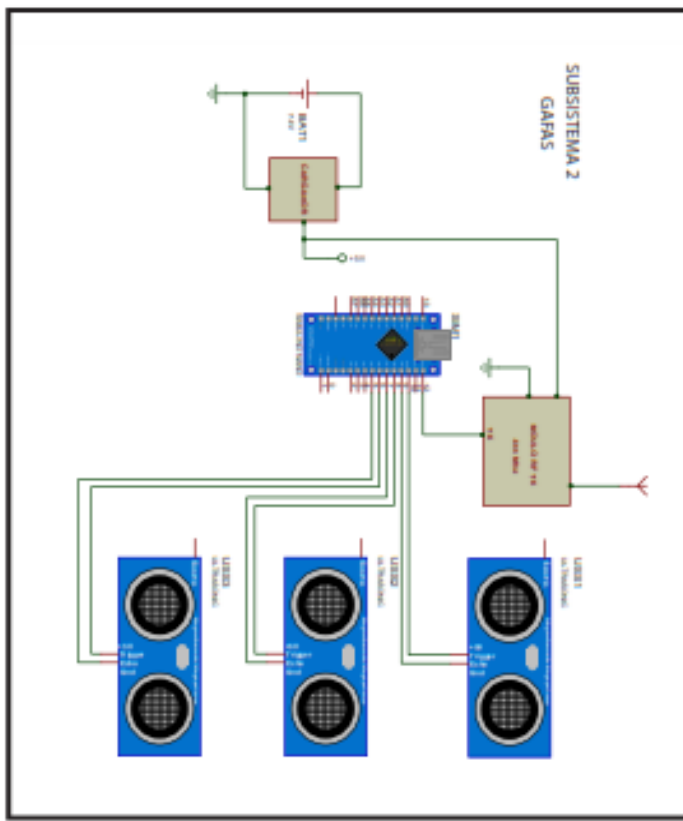
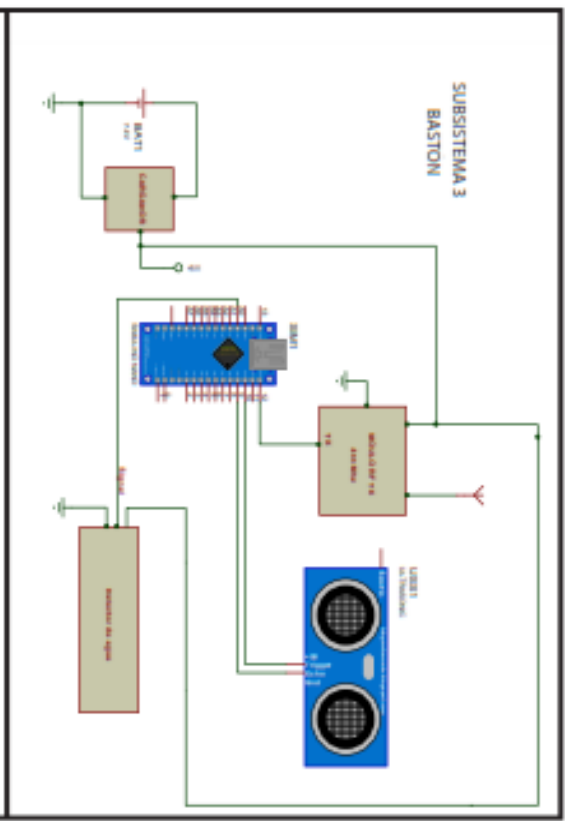
- AT cellular command interface

Approvals (in planning)

- CE
- FCC
- ROHS
- PTCRB
- GCF
- AT&T
- IC
- TA



More about SIM900 module, Please contact: Tel:+86 21 32523300



PROGRAMACIÓN DE LAS GAFAS

```
/*
EMISOR
Envía los mensajes "Apagar" y "Encender" en intervalos de un segundo
Escrito por Nano en beneficio de los seres humanos
www.robologs.net
*/
#include <VirtualWire.h>
long distancia;//Variable para almacenar el valor de la distancia
long tiempo;
int dato;
void setup()
{
//Iniciamos el Serial y la comunicación por radio
Serial.begin(9600);
Serial.write("Sistema encendido\n");
vw_setup(2000);
pinMode(9, OUTPUT); /*activación del pin 9 como salida: para el pulso ultrasónico*/
pinMode(8, INPUT); /*activación del pin 8 como entrada: tiempo del rebote del ultrasonido*/
pinMode(7, OUTPUT); /*activación del pin 7 como salida: para el pulso ultrasónico*/
pinMode(6, INPUT); /*activación del pin 6 como entrada: tiempo del rebote del ultrasonido*/
pinMode(5, OUTPUT); /*activación del pin 5 como salida: para el pulso ultrasónico*/
pinMode(4, INPUT); /*activación del pin 4 como entrada: tiempo del rebote del ultrasonido*/
}
void loop()
{
medir1();
analisis1();
medir2();
analisis2();
medir3();
analisis3();
}
//Función para enviar el mensaje
void send (char *message)
{
vw_send((uint8_t *)message, strlen(message)); //Envía el mensaje
vw_wait_tx(); //Espera hasta que se haya acabado de transmitir todo
Serial.println(message); //Muestra el mensaje por Serial
}
void medir1(){
digitalWrite(5,LOW); /* Por cuestión de estabilización del sensor*/
delayMicroseconds(5);
digitalWrite(5, HIGH); /* envío del pulso ultrasónico*/
delayMicroseconds(10);//pausa de 10 microsegundos
tiempo=pulseIn(4, HIGH); /* Función para medir la longitud del pulso entrante. Mide el tiempo
que transcurrido entre el envío del pulso ultrasónico y cuando el sensor recibe el rebote, es
decir: desde que el pin 8 empieza a recibir el rebote, HIGH, hasta que deja de hacerlo, LOW,
```



```

la longitud del pulso entrante*/distancia= int(0.017*tiempo); /*fórmula para calcular la
distancia obteniendo un valor entero*/
/*Monitorización en centímetros por el monitor serial*/
Serial.println("Distancia ");//Imprime en pantalla la palabra "Distancia"
Serial.println(distancia);//Imprime el valor de distancia en centímetros
Serial.println(" cm");
delay(100);//Pausa de 1 segundo antes de volver a tomar otro dato de lectura
}
void medir2(){
digitalWrite(7,LOW); /* Por cuestión de estabilización del sensor*/
delayMicroseconds(5);
digitalWrite(7, HIGH); /* envío del pulso ultrasónico*/
delayMicroseconds(10);//pausa de 10 microsegundos
tiempo=pulseIn(6, HIGH); /* Función para medir la longitud del pulso entrante. Mide el tiempo
que transcurrido entre el envío del pulso ultrasónico y cuando el sensor recibe el rebote, es
decir: desde que el pin 8 empieza a recibir el rebote, HIGH, hasta que deja de hacerlo, LOW,
la longitud del pulso entrante*/ distancia= int(0.017*tiempo); /*fórmula para calcular la
distancia obteniendo un valor entero*/
/*Monitorización en centímetros por el monitor serial*/
Serial.println("Distancia ");//Imprime en pantalla la palabra "Distancia"
Serial.println(distancia);//Imprime el valor de distancia en centímetros
Serial.println(" cm");
delay(100);//Pausa de 1 segundo antes de volver a tomar otro dato de lectura
}
void medir3(){
digitalWrite(9,LOW); /* Por cuestión de estabilización del sensor*/
delayMicroseconds(5);
digitalWrite(9, HIGH); /* envío del pulso ultrasónico*/
delayMicroseconds(10);//pausa de 10 microsegundos
tiempo=pulseIn(8, HIGH); /* Función para medir la longitud del pulso entrante. Mide el tiempo
que transcurrido entre el envío del pulso ultrasónico y cuando el sensor recibe el rebote, es
decir: desde que el pin 8 empieza a recibir el rebote, HIGH, hasta que deja de hacerlo, LOW,
la longitud del pulso entrante*/ distancia= int(0.017*tiempo); /*fórmula para calcular la
distancia obteniendo un valor entero*/
/*Monitorización en centímetros por el monitor serial*/
Serial.println("Distancia ");//Imprime en pantalla la palabra "Distancia"
Serial.println(distancia);//Imprime el valor de distancia en centímetros
Serial.println(" cm");
delay(100);//Pausa de 1 segundo antes de volver a tomar otro dato de lectura
}
void analisis1(){
if (distancia <= 30){
send("codigoe");
Serial.println("codigoe");
pausa();
}
if (distancia > 30 && distancia <= 70){
send("codigof");
Serial.println("codigof");
}
}

```

```

pausa();
}
if (distancia > 70 && distancia <= 120){
send("codigog");
Serial.println("codigog");
pausa();
}
}
void analisis2(){
if (distancia <= 30){
send("codigoh");
Serial.println("codigoh");
pausa();
}
if (distancia > 30 && distancia <= 70){
send("codigoi");
Serial.println("codigoi");
pausa();
}
if (distancia > 70 && distancia <= 120){
send("codigoj");
Serial.println("codigoj");
pausa();
}
}
void analisis3(){
if (distancia <= 30){
send("codigok");
Serial.println("codigok");
pausa();
}
if (distancia > 30 && distancia <= 70){
send("codigol");
Serial.println("codigol");
pausa();
}
if (distancia > 70 && distancia <= 120){
send("codigom");
Serial.println("codigom");
pausa();
}
}
void pausa(){
delay(4000);
}

```

PROGRAMACIÓN DEL BASTÓN

```

/*
EMISOR

```

```

Envía los mensajes "Apagar" y "Encender" en intervalos de un segundo
Escrito por Nano en beneficio de los seres humanos
www.robotlogs.net
*/
#include <VirtualWire.h>
long distancia;//Variable para almacenar el valor de la distancia
long tiempo;
int dato;
void setup()
{
//Iniciamos el Serial y la comunicación por radio
Serial.begin(9600);
Serial.write("Sistema encendido\n");
vw_setup(2000);
pinMode(9, OUTPUT); /*activación del pin 9 como salida: para el pulso ultrasónico*/
pinMode(8, INPUT); /*activación del pin 8 como entrada: tiempo del rebote del ultrasonido*/
}
void loop()
{
medir();
analisis();
agua();
}
//Función para enviar el mensaje
void send (char *message)
{
vw_send((uint8_t *)message, strlen(message)); //Envía el mensaje
vw_wait_tx(); //Espera hasta que se haya acabado de transmitir todo
Serial.println(message); //Muestra el mensaje por Serial
}
void medir(){
digitalWrite(9,LOW); /* Por cuestión de estabilización del sensor*/
delayMicroseconds(5);
digitalWrite(9, HIGH); /* envío del pulso ultrasónico*/
delayMicroseconds(10);//pausa de 10 microsegundos
tiempo=pulseIn(8, HIGH); /* Función para medir la longitud del pulso entrante. Mide el tiempo
que transcurre entre el envío del pulso ultrasónico y cuando el sensor recibe el rebote, es
decir: desde que el pin 8 empieza a recibir el rebote, HIGH, hasta que deja de hacerlo, LOW,
la longitud del pulso entrante*/distancia= int(0.017*tiempo); /*fórmula para calcular la
distancia
obteniendo un valor entero*/
/*Monitorización en centímetros por el monitor serial*/
//Serial.println("Distancia ");//Imprime en pantalla la palabra "Distancia"
//Serial.println(distancia);//Imprime el valor de distancia en centímetros
//Serial.println(" cm");
delay(100);//Pausa de 1 segundo antes de volver a tomar otro dato de lectura
}

void analisis(){

```

```
if (distancia <= 30){
send("codigoa");
Serial.println("codigoa");
pausa();
}
if (distancia >30 && distancia <= 70){
send("codigob");
Serial.println("codigob");
pausa();
}
if (distancia > 70 && distancia <= 120 ){
send("codigoc");
Serial.println("codigoc");
pausa();
}
}
void agua(){
dato = analogRead(0);//leer el valor del sensor de A0
if (dato > 512){
send("codigod");
Serial.println("codigod");
pausa();
}
}
void pausa(){
delay (3000);
}
```

PROGRAMACIÓN DE LA TARJETA MADRE (CINTURÓN)

```
/*
RECEPTOR
Apaga el LED 13 si recibe el mensaje "Apagar"
Enciende el LED 13 si recibe el mensaje "Encender"
Escrito por Nano en beneficio de los seres humanos
www.robologs.net
*/
#include <VirtualWire.h>
#include <SoftwareSerial.h>
#include <DFPlayer_Mini_Mp3.h>
#include <Wire.h>
#include "RTCLib.h"
RTC_DS1307 RTC;
//Creamos un mensaje
//La constante VW_MAX_MESSAGE_LEN viene definida en la librería
byte message[VW_MAX_MESSAGE_LEN];
byte messageLength = VW_MAX_MESSAGE_LEN;
int suiche =2;
int dato;
void setup()
{
pinMode(13, OUTPUT); //Configuramos el pin 13
pinMode(suiche, INPUT); //Configuramos el pin 2
Serial.begin(9600); //Iniciamos el Serial
Serial.println("Iniciando...");
vw_setup(2000);
vw_rx_start();
mp3_set_serial (Serial); //set Serial for DFPlayer-mini mp3 module
delay(1); // delay 1ms to set volume
mp3_set_volume (25);
Wire.begin(); // Establece la velocidad de datos del bus I2C
RTC.begin(); // Establece la velocidad de datos del RTC
}
void loop()
{
if (vw_get_message(message, &messageLength))
{
if(comparar("codigoa") == 0){
digitalWrite(13, HIGH);
//Serial.write("Obstaculo bastón 10cm\n");
mp3_play (61);
pausa();
}
else if(comparar("codigob") == 0)
{
digitalWrite(13,LOW);
//Serial.write("Obstaculo bastón 30cm\n");
mp3_play (62);
}
}
}
```

```

pausa();
}
else if(comparar("codigoc") == 0)
{
digitalWrite(13,LOW);
//Serial.write("Obstaculo bastón 70cm\n");
mp3_play (63);
pausa();
}
else if(comparar("codigod") == 0)
{
digitalWrite(13,LOW);
//Serial.write("Se detectó agua\n");
mp3_play (64);
pausa();
}
else if(comparar("codigoe") == 0)
{
digitalWrite(13,LOW);
//Serial.write("Obstaculo en gafas delante 10 cm\n");
mp3_play (65);
pausa();
}
else if(comparar("codigof") == 0)
{
digitalWrite(13,LOW);
//Serial.write("Obstaculo en gafas delante 30 cm\n");
mp3_play (66);
pausa();
}
else if(comparar("codigog") == 0)
{
digitalWrite(13,LOW);
//Serial.write("Obstaculo en gafas delante 70 cm\n");
mp3_play (67);
pausa();
}
else if(comparar("codigoh") == 0)
{
digitalWrite(13,LOW);
//Serial.write("Obstáculo en gafas izquierda 10 cm\n");
mp3_play (68);
pausa();
}
else if(comparar("codigoi") == 0)
{
digitalWrite(13,LOW);
//Serial.write("Obstáculo en gafas izquierda 30 cm\n");
mp3_play (69);
pausa();
}

```

```

}
digitalWrite(13,LOW);
//Serial.write("Obstáculo en gafas izquierda 70 cm\n");
mp3_play (70);
pausa();
}
else if(comparar("codigok") == 0)
{
digitalWrite(13,LOW);
//Serial.write("Obstáculo en gafas derecha 10 cm\n");
mp3_play (71);
pausa();
}
else if(comparar("codigol") == 0)
{
digitalWrite(13,LOW);
//Serial.write("Obstáculo en gafas derecha 30 cm\n");
mp3_play (72);
pausa();
}
else if(comparar("codigom") == 0)
{
digitalWrite(13,LOW);
//Serial.write("Obstáculo en gafas derecha 70 cm\n");
mp3_play (73);
pausa();
}
}
dato = digitalRead(suiche); //almacenamos en dato el valor
if (dato == LOW){ //si dato es alto
//Serial.write("La hora es :\n");
mp3_play (74);
delay(1500);
analishora();
analisisminuto();
pausa();
}
}
char comparar(char* cadena) {
//Esta función compara el string cadena con el mensaje recibido.
//Si son iguales, devuelve 1. Si no, devuelve 0.
for(int i = 0; i<messageLength; i++)
{
if(message[i] != cadena[i])
{
return 1;
}
}
return 0;
}

```

```

void pausa(){
delay(3000);
}
void analisishora(){
DateTime now = RTC.now();
if(now.hour() == 0){
mp3_play (0);
pausatiempo();
mp3_play (75); //decir HORAS
pausatiempo();
}
if(now.hour() == 1){
mp3_play (1);
pausatiempo();
mp3_play (76); //decir HORA
pausatiempo();
}
if(now.hour() == 2){
mp3_play (2);
pausatiempo();
mp3_play (75); //decir HORAS
pausatiempo();
}
if(now.hour() == 3){
mp3_play (3);
pausatiempo();
mp3_play (75); //decir HORAS
pausatiempo();
}
if(now.hour() == 4){
mp3_play (4);
pausatiempo();
mp3_play (75); //decir HORAS
pausatiempo();
}
if(now.hour() == 5){
mp3_play (5);
pausatiempo();
mp3_play (75); //decir HORAS
pausatiempo();
}
if(now.hour() == 6){
mp3_play (6);
pausatiempo();
mp3_play (75); //decir HORAS
pausatiempo();
}
if(now.hour() == 7){
mp3_play (7);
pausatiempo();
}

```



```

mp3_play (75); //decir HORAS
pausatiempo();
}
if(now.hour() == 8){
mp3_play (8);
pausatiempo();
mp3_play (75); //decir HORAS
pausatiempo();
}
if(now.hour() == 9){
mp3_play (9);
pausatiempo();
mp3_play (75); //decir HORAS
pausatiempo();
}
if(now.hour() == 10){
mp3_play (10);
pausatiempo();
mp3_play (75); //decir HORAS
pausatiempo();
}
if(now.hour() == 11){
mp3_play (11);
pausatiempo();
mp3_play (75); //decir HORAS
pausatiempo();
}
if(now.hour() == 12){
mp3_play (12);
pausatiempo();
mp3_play (75); //decir HORAS
pausatiempo();
}
if(now.hour() == 13){
mp3_play (13);
pausatiempo();
mp3_play (75); //decir HORAS
pausatiempo();
}
if(now.hour() == 14){
mp3_play (14);
pausatiempo();
mp3_play (75); //decir HORAS
pausatiempo();
}
if(now.hour() == 15){
mp3_play (15);
pausatiempo();
mp3_play (75); //decir HORAS
pausatiempo();
}

```

```

}
if(now.hour() == 16){
mp3_play (16);
pausatiempo();
mp3_play (75); //decir HORAS
pausatiempo();
}
if(now.hour() == 17){
mp3_play (17);
pausatiempo();
mp3_play (75); //decir HORAS
pausatiempo();
}
if(now.hour() == 18){
mp3_play (18);
pausatiempo();
mp3_play (75); //decir HORAS
pausatiempo();
}
if(now.hour() == 19){
mp3_play (19);
pausatiempo();
mp3_play (75); //decir HORAS
pausatiempo();
}
if(now.hour() == 20){
mp3_play (20);
pausatiempo();
mp3_play (75); //decir HORAS
pausatiempo();
}
if(now.hour() == 21){
mp3_play (21);
pausatiempo();
mp3_play (75); //decir HORAS
pausatiempo();
}
if(now.hour() == 22){
mp3_play (22);
pausatiempo();
mp3_play (75); //decir HORAS
pausatiempo();
}
if(now.hour() == 23){
mp3_play (23);
pausatiempo();
mp3_play (75); //decir HORAS
pausatiempo();
}
}
}

```

```

void pausatiempo(){
delay(1500);
}
void analisisminuto(){
DateTime now = RTC.now();
if(now.minute() == 0){
mp3_play (0);
pausatiempo();
mp3_play (78); //decir minutos
pausatiempo();
}

if(now.minute() == 1){
mp3_play (1);
pausatiempo();
mp3_play (77); //decir minuto
pausatiempo();
}
if(now.minute() == 2){
mp3_play (2);
pausatiempo();
mp3_play (78); //decir HORAS
pausatiempo();
}
if(now.minute() == 3){
mp3_play (3);
pausatiempo();
mp3_play (78); //decir HORAS
pausatiempo();
}
if(now.minute() == 4){
mp3_play (4);
pausatiempo();
mp3_play (78); //decir HORAS
pausatiempo();
}
if(now.minute() == 5){
mp3_play (5);
pausatiempo();
mp3_play (78); //decir HORAS
pausatiempo();
}
if(now.minute() == 6){
mp3_play (6);
pausatiempo();
mp3_play (78); //decir HORAS
pausatiempo();
}
if(now.minute() == 7){
mp3_play (7);

```

```

pausatiempo();
mp3_play (78); //decir HORAS
pausatiempo();
}
if(now.minute() == 8){
mp3_play (8);
pausatiempo();
mp3_play (78); //decir HORAS
pausatiempo();
}
if(now.hour() == 9){
mp3_play (9);
pausatiempo();
mp3_play (78); //decir HORAS
pausatiempo();
}
if(now.minute() == 10){
mp3_play (10);
pausatiempo();
mp3_play (78); //decir HORAS
pausatiempo();
}
if(now.minute() == 11){
mp3_play (11);
pausatiempo();
mp3_play (78); //decir HORAS
pausatiempo();
}
if(now.minute() == 12){
mp3_play (12);
pausatiempo();
mp3_play (78); //decir HORAS
pausatiempo();
}
if(now.minute() == 13){
mp3_play (13);
pausatiempo();
mp3_play (78); //decir HORAS
pausatiempo();
}
if(now.minute() == 14){
mp3_play (14);
pausatiempo();
mp3_play (78); //decir HORAS
pausatiempo();
}
if(now.minute() == 15){
mp3_play (15);
pausatiempo();
mp3_play (78); //decir HORAS

```

```

pausatiempo();
}
if(now.minute() == 16){
mp3_play (16);
pausatiempo();
mp3_play (78); //decir HORAS
pausatiempo();
}
if(now.minute() == 17){
mp3_play (17);
pausatiempo();
mp3_play (78); //decir HORAS
pausatiempo();
}
if(now.minute() == 18){
mp3_play (18);
pausatiempo();
mp3_play (78); //decir HORAS
pausatiempo();
}
if(now.minute() == 19){
mp3_play (19);
pausatiempo();
mp3_play (78); //decir HORAS
pausatiempo();
}
if(now.minute() == 20){
mp3_play (20);
pausatiempo();
mp3_play (78); //decir HORAS
pausatiempo();
}
if(now.minute() == 21){
mp3_play (21);
pausatiempo();
mp3_play (78); //decir HORAS
pausatiempo();
}
if(now.minute() == 22){
mp3_play (22);
pausatiempo();
mp3_play (78); //decir HORAS
pausatiempo();
}
if(now.minute() == 23){
mp3_play (23);
pausatiempo();
mp3_play (78); //decir HORAS
pausatiempo();
}
}

```

```

if(now.minute() == 24){
mp3_play (24);
pausatiempo();
mp3_play (78); //decir HORAS
pausatiempo();
}
if(now.minute() == 25){
mp3_play (25);
pausatiempo();
mp3_play (78); //decir HORAS
pausatiempo();
}
if(now.minute() == 26){
mp3_play (26);
pausatiempo();
mp3_play (78); //decir HORAS
pausatiempo();
}
if(now.minute() == 27){
mp3_play (27);
pausatiempo();
mp3_play (78); //decir HORAS
pausatiempo();
}
if(now.minute() == 28){
mp3_play (28);
pausatiempo();
mp3_play (78); //decir HORAS
pausatiempo();
}
if(now.hour() == 29){
mp3_play (29);
pausatiempo();
mp3_play (78); //decir HORAS
pausatiempo();
}
if(now.minute() == 30){
mp3_play (30);
pausatiempo();
mp3_play (78); //decir HORAS
pausatiempo();
}
if(now.minute() == 31){
mp3_play (31);
pausatiempo();
mp3_play (77); //decir minuto
pausatiempo();
}
if(now.minute() == 32){
mp3_play (32);

```

```

pausatiempo();
mp3_play (78); //decir HORAS
pausatiempo();
}
if(now.minute() == 33){
mp3_play (33);
pausatiempo();
mp3_play (78); //decir HORAS
pausatiempo();
}
if(now.minute() == 34){
mp3_play (34);
pausatiempo();
mp3_play (78); //decir HORAS
pausatiempo();
}
if(now.minute() == 35){
mp3_play (35);
pausatiempo();
mp3_play (78); //decir HORAS
pausatiempo();
}
if(now.minute() == 36){
mp3_play (36);
pausatiempo();
mp3_play (78); //decir HORAS
pausatiempo();
}
if(now.minute() == 37){
mp3_play (37);
pausatiempo();
mp3_play (78); //decir HORAS
pausatiempo();
}
if(now.minute() == 38){
mp3_play (38);
pausatiempo();
mp3_play (78); //decir HORAS
pausatiempo();
}
if(now.hour() == 39){
mp3_play (39);
pausatiempo();
mp3_play (78); //decir HORAS
pausatiempo();
}
if(now.minute() == 40){
mp3_play (40);
pausatiempo();
mp3_play (78); //decir HORAS

```

```

pausatiempo();
}
if(now.minute() == 41){
mp3_play (41);
pausatiempo();
mp3_play (77); //decir minuto
pausatiempo();
}
if(now.minute() == 42){
mp3_play (42);
pausatiempo();
mp3_play (78); //decir HORAS
pausatiempo();
}
if(now.minute() == 43){
mp3_play (43);
pausatiempo();
mp3_play (78); //decir HORAS
pausatiempo();
}
if(now.minute() == 44){
mp3_play (44);
pausatiempo();
mp3_play (78); //decir HORAS
pausatiempo();
}
if(now.minute() == 45){
mp3_play (45);
pausatiempo();
mp3_play (78); //decir HORAS
pausatiempo();
}
if(now.minute() == 46){
mp3_play (46);
pausatiempo();
mp3_play (78); //decir HORAS
pausatiempo();
}
if(now.minute() == 47){
mp3_play (47);
pausatiempo();
mp3_play (78); //decir HORAS
pausatiempo();
}
if(now.minute() == 48){
mp3_play (48);
pausatiempo();
mp3_play (78); //decir HORAS
pausatiempo();
}
}

```



```

if(now.hour() == 49){
mp3_play (49);
pausatiempo();
mp3_play (78); //decir HORAS
pausatiempo();
}
if(now.minute() == 50){
mp3_play (50);
pausatiempo();
mp3_play (78); //decir HORAS
pausatiempo();
}
if(now.minute() == 51){
mp3_play (51);
pausatiempo();
mp3_play (77); //decir minuto
pausatiempo();
}
if(now.minute() == 52){
mp3_play (52);
pausatiempo();
mp3_play (78); //decir HORAS
pausatiempo();
}

if(now.minute() == 53){
mp3_play (53);
pausatiempo();
mp3_play (78); //decir HORAS
pausatiempo();
}
if(now.minute() == 54){
mp3_play (54);
pausatiempo();
mp3_play (78); //decir HORAS
pausatiempo();
}
if(now.minute() == 55){
mp3_play (55);
pausatiempo();
mp3_play (78); //decir HORAS
pausatiempo();
}
if(now.minute() == 56){
mp3_play (56);
pausatiempo();
mp3_play (78); //decir HORAS
pausatiempo();
}
if(now.minute() == 57){

```

```
mp3_play (57);
pausatiempo();
mp3_play (78); //decir HORAS
pausatiempo();
}
if(now.minute() == 58){
mp3_play (58);
pausatiempo();
mp3_play (78); //decir HORAS
pausatiempo();
}
if(now.minute() == 59){
mp3_play (59);
pausatiempo();
mp3_play (78); //decir HORAS
pausatiempo();
}
if(now.minute() == 60){
mp3_play (60);
pausatiempo();
mp3_play (78); //decir HORAS
pausatiempo();
}
}
```

DECLARACION Y AUTORIZACION

Yo, LENIN GIOVANNY PAUCAR ESPINOSA, CI 1720335205 autor/a del trabajo de graduación:

Gafas y Bastón inteligente para una persona invidente, previo a la obtención del título de **Ingeniería en Electrónica Digital Y telecomunicaciones** en la UNIVERSIDAD TECNOLÓGICA ISRAEL.

1.- Declaro tener pleno conocimiento de la obligación que tienen las instituciones de educación superior, de conformidad con el Artículo 144 de la Ley Orgánica de Educación Superior, de difundir el respectivo trabajo de graduación para que sea integrado al Sistema Nacional de Información de la Educación Superior del Ecuador para su difusión pública respetando los derechos de autor.

2.- Autorizo a la SENESCYT a tener una copia del referido trabajo de graduación, con el propósito de generar un repositorio que democratice la información, respetando las políticas de propiedad intelectual vigentes.

Quito, Marzo del 2018

Atentamente.

Lenin Giovanni Paucar Espinosa.

C.I. 1720335205

