



UNIVERSIDAD TECNOLÓGICA ISRAEL

TRABAJO DE TITULACIÓN EN OPCIÓN AL GRADO DE:

**INGENIERA EN ELECTRÓNICA DIGITAL Y
TELECOMUNICACIONES**

TEMA:

**SISTEMA DE TRADUCCIÓN DE LENGUAJE DE SEÑAS A LENGUAJE ORAL Y
ESCRITO MEDIANTE UN GUANTE ELECTRÓNICO**

AUTOR: BYRON FERNANDO GARZÓN TIPÁN

TUTOR: MG. RENÉ ERNESTO CORTIJO LEYVA

QUITO – ECUADOR

2020

APROBACIÓN DEL TUTOR

En mi calidad de **TUTOR DEL TRABAJO DE TITULACIÓN DIRECTOR** del Proyecto: **SISTEMA DE TRADUCCIÓN DE LENGUAJE DE SEÑAS A LENGUAJE ORAL Y ESCRITO MEDIANTE UN GUANTE ELECTRÓNICO, DE LA CIUDAD DE QUITO**. Presentado por el ciudadano **BYRON FERNANDO GARZÓN TIPAN** estudiante del programa de Ingeniería en **ELECTRÓNICA DIGITAL Y TELECOMUNICACIONES** de la Universidad Tecnológica Israel considero que dicho informe investigativo reúne los requisitos y méritos suficientes para ser sometido a la revisión y evaluación respectiva por parte del Tribunal de grado que se digne para su correspondiente estudio y calificación.

Quito D.M. _____.

EL TUTOR

Ing. René Ernesto Cortijo Leyva, Mg.

DECLARACION DE AUTENTICIDAD

Yo, **BYRON FERNANDO GARZÓN TIPAN**, con cedula de identidad No 171676324-6 declaro que este trabajo de titulación que lleva por tema **“SISTEMA DE TRADUCCIÓN DE LENGUAJE DE SEÑAS A LENGUAJE ORAL Y ESCRITO MEDIANTE UN GUANTE ELECTRÓNICO, DE LA CIUDAD DE QUITO.”** es de mi autoría; el cual ha sido desarrollado considerando los métodos de investigación existentes, así como también se ha respetado los derechos intelectuales de terceros considerándose en las citas bibliográficas.

La Universidad Tecnológica Israel, puede hacer uso de los derechos correspondientes a este trabajo, según lo establecido en su reglamento y por la normatividad institucional vigente.

Quito D.M. _____.

BYRON FERNANDO GARZÓN TIPAN
C.I.: 1802946853

AGRADECIMIENTO

A mis maestros y profesores de la carrera de Electrónica Digital y Electrónica, por la ayuda y los conocimientos que nos compartieron durante el tiempo que estude en la Universidad Israel, brindando su apoyo con paciencia, que me ayudo en la vida profesional y mi vida personal.

Gracias

DEDICATORIA

A mis padres, mis hermanas e hijo por el apoyo, cariño, confianza brindada cada día, ayudándome con su amor que gracias a ellos he logrado llegar hasta aquí, este el momento para demostrarle todo mi agradecimiento y amor que siento para ellos.

A los docentes por haberme brindado la confianza y sus conocimientos gracias a todos los conocimientos adquiridos en práctica en mi vida laboral.

El autor

TABLA DE CONTENIDO

RESUMEN	XI
ABSTRACT	XII
INTRODUCCIÓN	1
ANTECEDENTES	1
PLANTEAMIENTO Y JUSTIFICACIÓN DEL PROBLEMA	3
OBJETIVOS DEL TRABAJO DE TITULACIÓN	4
<i>Objetivo general</i>	4
<i>Objetivos Específicos</i>	4
ALCANCE.....	5
LA HIPÓTESIS O IDEAS A DEFENDER EN EL PROCESO INVESTIGATIVO.....	5
DESCRIPCIÓN DE LOS CAPÍTULOS	6
CAPÍTULO 1	7
FUNDAMENTACIÓN TEÓRICA	7
1.1 LENGUAJE DE SEÑAS	7
1.2 CONDUCTOR ELÉCTRICO	10
1.3 LA RESISTENCIA ELÉCTRICA.....	10
1.4 SENSORES	11
<i>1.4.1 Sensor Flex</i>	11
1.5 MICRO PULSADORES.....	13
1.6 TARJETA ARDUINO	14
<i>1.6.1 Arduino Nano</i>	14
<i>1.6.2 Características principales</i>	15
1.7 COMUNICACIÓN INALÁMBRICA.....	16
<i>1.7.1 Bluetooth</i>	16
1.8 MÓDULO BLUETOOTH ARDUINO HC-05	17

1.8.1 Características	17
1.9 PROGRAMACIÓN ARDUINO.....	17
1.9.1 Estructura.....	19
1.9.2 Comandos básicos de programación	19
1.10 MIT APP INVENTOR	23
1.10.1 Componentes diseñador	24
1.11 ANDROID	26
1.12 CARGADOR PORTÁTIL.....	27
CAPÍTULO 2.....	28
MARCO METODOLÓGICO.....	28
2.1 TIPO DE INVESTIGACIÓN	28
2.1.1 Modalidad	28
2.1.2 Diseño de la investigación	29
2.1.3 Enfoque de la investigación	30
2.1.4 Técnica de recolección de datos	30
2.2 FASE DE DESARROLLO	30
2.2.1 Fase I. Selección de los gestos.....	30
2.2.2 Fase II. Reconocer los componentes.....	31
2.2.3 Fase III. Desarrollo del proyecto.....	31
2.2.4 Fase IV. Pruebas de funcionamiento	32
CAPÍTULO 3.....	33
PROPUESTA.....	33
3.1 IDEA DE LA PROPUESTA	33
3.2 IDEA INTEGRAL DE LA PROPUESTA	33
3.3 PLANTEAMIENTO DE UNA APLICACIÓN	34
3.4 PLAN DE TRABAJO	34

CAPÍTULO 4.....	36
IMPLEMENTACIÓN	36
4.1 FABRICACIÓN DE GUANTE ELECTRÓNICO.....	36
4.1.1 <i>Desmontaje de guante derecho</i>	36
4.1.2 <i>Fijación de componentes al guante</i>	37
4.1.3 <i>Diseño de la placa electrónica</i>	39
4.1.4 <i>Diseño de caja plástica</i>	39
4.1.5 <i>Producto</i>	40
4.2. CÁLCULOS	42
4.3 DIAGRAMA DE CONEXIONES A LA TARJETA ARDUINO NANO	43
4.3.1 <i>Conexión del sensor flex</i>	43
4.3.2 <i>Conexión del micro pulsador</i>	44
4.3.3 <i>Conexión del módulo Bluetooth</i>	44
4.3.4 <i>Conexión de componentes del guante</i>	45
4.4 CREACIÓN DE LA APLICACIÓN CON EL SOFTWARE EN LÍNEA AI2	46
4.4.1 <i>Interfaz de usuario</i>	46
4.5 PRESUPUESTO	53
CONCLUSIONES.....	54
RECOMENDACIONES.....	55
BIBLIOGRAFÍA.....	56
ANEXOS	59
ANEXO 1: DIAGRAMA ELÉCTRICO.....	60
ANEXO 2: DISEÑO.....	61
ANEXO 3: DIAGRAMA DE FLUJO DE PROGRAMACIÓN ARDUINO	63
ANEXO 4: PROGRAMACIÓN ARDUINO.....	64
ANEXO 5: EDITOR DE BLOQUES APP INVENTOR 2	71

ANEXO 6: MANUAL DE USUARIO	74
ANEXO 7: FICHA TÉCNICA	84
<i>Anexo 7.1 Arduino Nano</i>	84
<i>Anexo 7.2 HC-05</i>	89
<i>Anexo 7.3 Sensor Flex</i>	106
ANEXO 8: CRONOGRAMA DE ACTIVIDADES	108

ÍNDICE DE FIGURAS

FIGURA 1.1 NÚMEROS Y LETRAS LENGUAJE DE SEÑAS	9
FIGURA 1.2 CONDUCTOR ELÉCTRICO	10
FIGURA 1.3 RESISTENCIA ELÉCTRICA	10
FIGURA 1.4.1 SENSOR FLEX	11
FIGURA 1.4.2 COMPORTAMIENTO INTERNO	12
FIGURA 1.5 MICRO PULSADORES	13
FIGURA 1.6 TARJETA ARDUINO	14
FIGURA 1.6.1 ARDUINO NANO	15
FIGURA 1.6.2 CARACTERÍSTICAS PRINCIPALES	15
FIGURA 1.7.1 BLUETOOTH	16
FIGURA 1.8 MODULO BLUETOOTH ARDUINO HC-05	17
FIGURA 1.9 PROGRAMACIÓN DE ARDUINO	18
FIGURA 1.10 MIT APP INVENTOR 2	23
FIGURA 1.10.1 COMPONENTES DISEÑADOR	24
FIGURA 1.10.1.2 EDITOR DE BLOQUES AI2	26
FIGURA 2.2.4 FASE IV. PRUEBAS DE FUNCIONAMIENTO	32

FIGURA 4.1.1 DESMONTAJE DE GUANTE.....	37
FIGURA 4.1.2 FIJACIÓN DE COMPONENTES AL GUANTE	38
FIGURA 4.1.3 INTERFAZ DE USUARIO.....	39
FIGURA 4.1.4 CAJA PLÁSTICA	40
FIGURA 4.1.5 PRODUCTO FINAL.....	41
FIGURA 4.2 CÁLCULOS	42
FIGURA 4.3.1. CONEXIÓN DEL SENSOR FLEX	43
FIGURA 4.3.2 CONEXIÓN DE MICRO PULSADOR	44
FIGURA 4.3.3 CONEXIÓN DEL MÓDULO BLUETOOTH.....	45
FIGURA 4.3.4 CONEXIÓN DE COMPONENTES DEL GUANTE	46
FIGURA 4.4.1 INTERFAZ DE USUARIO.....	47
FIGURA 4.4.1.1 PANTALLA 1	48
FIGURA 4.4.1.2 PANTALLA 2	49
FIGURA 4.4.1.3. PANTALLA 3	51
FIGURA 4.4.1.4 PANTALLA 4	52

INDICE DE TABLA

TABLA 4.5 PRESUPUESTO	53
------------------------------------	-----------

RESUMEN

El presente proyecto tiene como fin ofrecer la posibilidad de guiar el conocimiento del lenguaje de señas que utiliza la comunidad sorda. Se conoce que esta comunidad ha sido altamente vulnerable dado que es muy reducido el número de personas que conocen el lenguaje de señas, es así que, el guante tiene como propósito traducir números y letras en este idioma. Se conoce además que gracias a la traducción y guía que ofrece la aplicación creada en el guante, las personas que se inician en este idioma podrán aprender a hacer los gestos básicos de manera interactiva.

El modelo de traducción que se ha propuesto utilizar un guante electrónico provisto de sensores de flexión y micro pulsadores dispuestos de tal manera que puedan discriminar los gestos de la mano derecha para traducir a un lenguaje escrito y oral. La programación de los signos se realiza en una tarjeta Arduino que recibe las señales que proporcionan los gestos de las manos y los traduce al lenguaje que conocemos. Además, el módulo Bluetooth instalado en el sistema hace posible la conexión inalámbrica entre el guante electrónico y un dispositivo Android.

Finalmente, se considera importante esta investigación debido a que los costos del guante electrónico son de fácil acceso que fortalece el aprendizaje del idioma de señas para las personas sordas y las personas oyentes que quieran aprender este idioma.

Palabras claves: Interfaz de usuario, guante electrónico, programación Arduino,

AI2

ABSTRACT

The purpose of this project is to offer the possibility of guiding knowledge about the sign language used by the deaf community, it is known that this community has been highly vulnerable given that the number of people who know sign language is very small, so the glove is intended to translate numbers and letters in this language. It is known that thanks to the translation and guide offered by the application created in the glove, people who start in this language will be able to learn to do the basic gestures of this language in an interactive way.

The translation model that has been proposed uses an electronic glove provided with sensors and micro buttons arranged in such a way that they can discriminate the gestures of the right hand to translate into written and oral language. The programming of the signs is based on the Arduino board that receives the signals provided by the hand gestures and translates them into a language we know. In addition, the Bluetooth module installed in the system, wireless connection between the electronic glove and the Android device is possible.

Finally, this research is considered important because the costs of the electronic glove are easily accessible, in addition to strengthening the learning of sign language for deaf people and hearing people who want to learn this language.

Keywords: User interface, electronic glove, Arduino programming, AI2

INTRODUCCIÓN

Antecedentes

El presente trabajo tiene como base las telecomunicaciones y sistemas informáticos aplicados a la producción y la sociedad han facilitado la comunicación entre los seres humanos, es así que, proveer una herramienta electrónica como la traducción de señas en cualquier dispositivo Android ayudará a la comunidad sorda a tener una comunicación efectiva con las personas oyentes.

Las telecomunicaciones hoy en día son indispensables dado a que se transmite información concreta que se encuentra en la nube, gracias al internet es posible tener acceso a esta información, con el constante crecimiento de los dispositivos móviles las aplicaciones (APP) quedan almacenadas en terminales inteligentes, de forma que su uso es rápido e inmediato así lo menciona Roca.

Realizado el proceso de investigación se evidencia que la revista Global Research and Development Journal for Engineering (GRD Journals) ha publicado un artículo llamado "SPEAKING AID FOR DEAF AND DUMB USING FLEX SENSORS". Este trabajo presenta un dispositivo tecnológico que ayuda a las personas sordomudas a comunicarse mediante un dispositivo de salida de voz. Este trabajo de investigación plantea un modelo tecnológico que fue desarrollado con MPLAB IDE y PICKIT 3 los cuales hacen que su componente flexible cumpla con las demandas de 16 combinaciones. La visualización que manejan es display LCD. Dicha investigación no aborda los requerimientos tecnológicos para continuar perfeccionando este sistema (Manjari, Monisha. Mahalaxme, Kalki, & Priya, 2018)

En el repositorio de la UNIVERSIDAD DE LAS FUERZAS ARMADAS ESPE se obtuvo como resultado el siguiente trabajo investigativo: "DISEÑO E IMPLEMENTACIÓN DE UN PAR DE GUANTES INTERPRETES DEL LENGUAJE DE SEÑALES ELEMENTALES A LENGUAJE ESCRITO MEDIANTE SOFTWARE LIBRE PARA FACILITAR EL APRENDIZAJE EN LA UNIDAD EDUCATIVA ESPECIALIZADA COTOPAXI". Desarrollado por los señores Luis Cristóbal Almeida Pozo y Paúl Antonio Viteri Villacís en el año 2016, se realizó con el fin de ayudar a interpretar a la comunidad sordomuda. Las características que manejan en este dispositivo van acorde con el proyecto planteado debido a que usan comunicación inalámbrica, cuentan con tarjetas Arduino, entre otros elementos. Cabe destacar que la interacción es transformada a un lenguaje Escrito. (Almeida & Viteri, 2016)

En el repositorio de la UNIVERSIDAD TÉCNICA DEL NORTE se obtuvo como resultado el siguiente trabajo investigativo: "DISPOSITIVO TRADUCTOR DEL LENGUAJE DE SEÑALES DE PERSONAS SORDAS A SONIDOS AUDITIVOS DE LAS LETRAS DEL ABECEDARIO." Desarrollado por los señores Lenin Carlos Encalada Monteros y Harold Ernesto Carrasco Cueva en el año 2016. Ellos crearon un guante con el fin de que una persona sorda pueda usar en su mano derecha para reconocer el abecedario por medio de una aplicación móvil en un teléfono inteligente. La presente investigación muestra el uso de una plataforma bajo software libre llamado "APP Inventor" la cual usa un teléfono Android para que el output sea por una vía Bluetooth con el fin de que la traducción pueda ser escuchada. (Encalada & Carrasco, 2015).

Estos documentos proporcionan información sobre la funcionabilidad del proyecto que se desea implementar para la comunidad que utiliza el lenguaje de señas, siendo así que, las investigaciones hechas previamente permiten hacer viable la

investigación que se plantea dentro de este tema de titulación.

Planteamiento y justificación del problema

En el Ecuador desde la creación del CONADIS en el año de 1992, institución especializada en cuestiones de discapacidad que promueve su integración, afirma que la sordera o hipoacusia presenta problemas de comunicación con el entorno social que desconoce el idioma de señas (Espinoza, 2017, pág. 37). El Telégrafo, periódico ecuatoriano, hace mención sobre la discapacidad auditiva en el Ecuador a nivel nacional la cual abarca a 55.020 personas (EL Telégrafo, 2017). Además, el mismo periódico menciona que el CONADIS ha registrado a 62.921 sordos de los cuales tan solo 11.114 están empleados (Monroy, 2018). Es así que las personas con capacidades diferentes aún mantienen pocos lazos de relación con respecto a las personas que pueden comunicarse de manera oral.

De esta manera la Lengua de Señas Ecuatoriana (LSEC) es la lengua que durante años ha desarrollado y transmitido la Comunidad Sorda del Ecuador de generación en generación, en el Art. 70 de la Ley de Discapacidades define: "La lengua de señas ecuatoriana como lengua propia y medio de comunicación de las personas con discapacidad auditiva..." (Ley de Discapacidades, 2012, p. 16). Esta lengua tiene una modalidad viso-gestual, estos movimientos de la mano y expresiones tienen las mismas propiedades que las lenguas habladas. Gracias a ella, las personas sordas de Ecuador pueden comunicarse, transmitir sus deseos e intereses, informarse, defender sus derechos y construir una identidad positiva que las hace miembros de esta comunidad.

El presente proyecto se considera importante para las personas con capacidades

especiales que presentan problemas auditivos y manejan un lenguaje de señas, es así que, se plantea implementar un dispositivo inteligente con el fin de interpretar los diferentes movimientos básicos de la mano derecha que se realizan para tener una comunicación básica con las personas que no manejan el lenguaje de señas, se enfoca en primera instancia en los jóvenes para probar su funcionalidad.

Objetivos del trabajo de titulación

Objetivo general

Desarrollar un sistema de traducción de lenguaje de señas a lenguaje oral y escrito mediante un guante electrónico.

Objetivos Específicos

- Establecer los requerimientos básicos del lenguaje de señas para la adaptación de un guante electrónico.
- Definir la lista de dispositivos electrónicos de diseño del guante.
- Implementar los elementos electrónicos capaces de generar señales eléctricas sobre un guante flexible.
- Diseñar la programación para la tarjeta Arduino necesaria para el censo de los movimientos de la mano derecha.
- Desarrollar un software capaz de integrar la comunicación entre guante electrónico, tarjeta Arduino y un dispositivo portátil.
- Realizar pruebas de validación y funcionamiento

Alcance

Para garantizar que la comunidad sorda tenga otros medios de comunicación el presente proyecto de investigación plantea un guante electrónico que por medio de la aplicación denominada TraDeLeS creado en un programa de uso libre en línea llamado APP Inventor 2 (AI2), el cual es capaz de leer los movimientos que la mano realice.

La falta de comunicación entre la comunidad sorda y oyente a limitado que haya desigualdad de oportunidades, es así que, se plantea la creación del guante electrónico en el cual se alojan varios sensores, la información comprendida entre flexión de cada dedo en la mano derecha es enviada a una tarjeta Arduino mediante la conexión interna de todos los elementos electrónicos presentes en el guante. Esta tarjeta será capaz de comunicarse inalámbricamente con un dispositivo portátil que a su vez realizará la traducción del lenguaje de señas a lenguaje oral y escrito.

La fabricación de este sistema de guante (derecho) será en primera instancia de diseño, se utilizará como prueba piloto y posteriormente puede ser usado como un sistema de traducción integral del lenguaje de signos. La traducción es de treinta símbolos entre números y letras tomados del lenguaje básico de señas.

La hipótesis o ideas a defender en el proceso investigativo

Con la creación de un guante electrónico traductor de lenguaje de señas se pretende que la comunidad de personas con habilidades especiales que utilizan únicamente lenguaje de señas puedan integrarse e interactuar con personas que desconocen de este idioma, siendo capaces de intervenir en una charla o acceder a un

mejor puesto de trabajo.

Además, este guante podría usarse como un primer medio de aprendizaje para las personas que quieran estudiar o conocer este lenguaje y puedan corroborar si las señas que usan son acordes a lo que pretenden decir.

Descripción de los capítulos

El capítulo 1 contiene la fundamentación teórica que ha enriquecido el contenido del proyecto, es aquí donde se evidencia los trabajos de mayor impacto dentro de la creación de un guante para traducir el lenguaje de señas al lenguaje oral y escrito.

El capítulo 2 detalla la metodología empleada en la investigación, se describe todas las técnicas usadas en el desarrollo del proyecto tales como la modalidad de campo con relación al tipo de investigación descriptivo, el diseño de la investigación es no experimental y el enfoque de la investigación corresponde a la parte cuantitativa.

El capítulo 3 presenta la propuesta, es aquí donde se describe cada uno de los equipos y como se interrelacionan dentro de este proyecto, así cómo; su funcionamiento y capacidad. Después, se analiza los costos y características de cada uno. Finalmente, se presenta el presupuesto requerido para la ejecución del proyecto.

Para finalizar el capítulo 4 muestra la implementación del proyecto de investigación, topologías, instalación, configuración puesta en marcha de la solución; así como también las pruebas realizadas y resultados obtenidos

CAPÍTULO 1

FUNDAMENTACIÓN TEÓRICA

En este apartado se revisa el lenguaje de señas ecuatoriano y adicional se explica detalladamente el funcionamiento del guante y sus componentes, de manera que, la traducción sea traducida al lenguaje escrito y oral. En primer lugar, definiremos qué es el lenguaje de señas y, en particular, se entiende la necesidad que tiene la comunidad sorda en el Ecuador. En segundo lugar, se plantean los diferentes dispositivos que intervienen para llevar a cabo el guante háptico, así como los sensores que permiten la correcta medida y recepción de dichos estímulos. Finalmente, se hará mención sobre el estado del espacio virtual, debido a que, por medio de sensores se receptan los diferentes movimientos, lo que conlleva a que cada uno de estos movimientos genere diferentes voltajes dando como resultado un símbolo lingüístico de la comunidad sorda.

1.1 Lenguaje de señas

La revista virtual Healthline, manifiesta que existen varios factores para para

desarrollar sordera, tales como; puede ser de nacimiento, por enfermedad, por accidentes y por circunstancias particulares. Estos inconvenientes han llevado a que se genere el lenguaje de señas. Vera, F. (2020) hace mención en su trabajo en la forma en cómo aparece este lenguaje de signos a partir de la observación, de esta forma, se convierten en un sistema de comunicación viso-gestuales que ayudan a personas sordas en su vida personal y social. Además este lenguaje tiene propiedades lingüísticas independientes al igual que los idiomas hablados, con una gramática diferente al oral y escrito, la cual se manifiesta a través de las manos y la cara.

La Lengua de Señas Ecuatoriana (LSEC) es una comunidad que ha ido creciendo de manera formal desde 1992 de acuerdo con el CONADIS, además ha sido reconocida en el Art 70 de la Ley de Discapacidades donde se señala que esta lengua de señas ecuatoriana es una lengua y a la vez es un medio de comunicación propio de las personas con discapacidad auditiva (Ley de Discapacidades, 2012, p. 16). Su modalidad visogestual corresponde a los movimientos de la mano y expresiones faciales, es así que, las personas sordas de nuestro país pueden comunicarse, construyendo una identidad positiva que las hace miembros de esa comunidad.

El abecedario dactilológico y los números que se muestran en la Fig.1.1 que se encuentra vigente en el Ecuador.

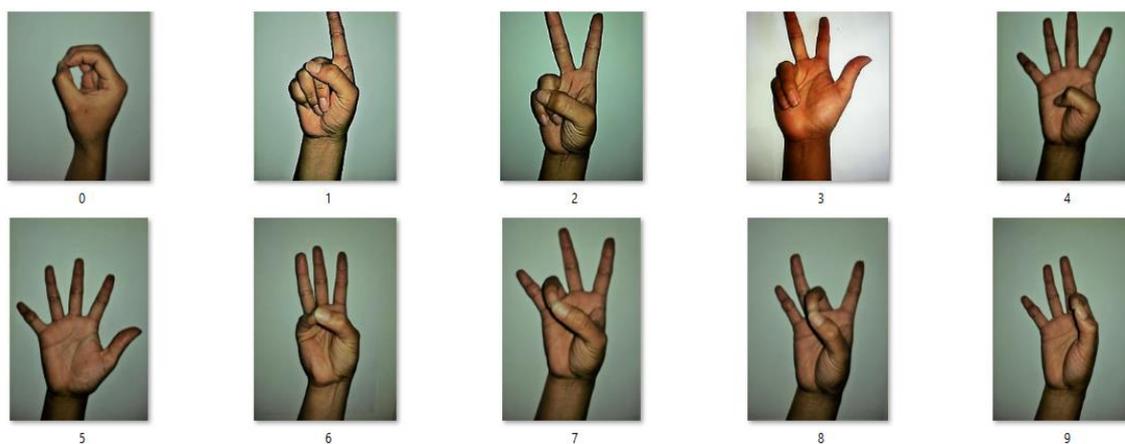




Figura 1.1 Números y Letras Lenguaje de Señas
Fuente: CONADIS (1992)

De acuerdo con el Consejo Nacional de Igualdad de Discapacidades del Ecuador (CONADIS), en Ecuador, existe alrededor de 62.921 personas que registran esta discapacidad y aproximadamente 11.114 personas tienen trabajo. Sin embargo, existe un alto número de personas que no están registradas debido a las dificultades y falta de comunicación. Por lo que se puede concluir que no hay suficiente información acerca de la realidad de las personas sordas en Ecuador.

La Constitución Política de la República del Ecuador del año 1998, declara en el Artículo 53 donde se reconoce el derecho de las personas con discapacidad, a la comunicación por medio de formas alternativas tales como el sistema Braille y otras.

1.2 Conductor eléctrico

Es un elemento que permite la circulación de corriente, y se lo puede observar en la figura 1.2, gracias a su baja resistencia eléctrica, generalmente puede ser de cobre o aluminio. Para su fácil maleabilidad para este proyecto se ha usado conductores flexibles, debido a que van a estar sometidos a movimiento constante.

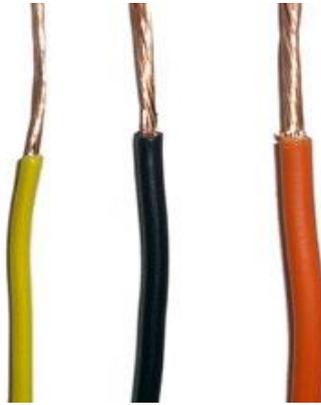


Figura 1.2 Conductor Eléctrico
Fuente: El autor

1.3 La resistencia eléctrica

Se denomina así a la oposición que presenta algún material al paso del flujo de corriente, la imagen y representación esquemática está disponible en la figura 1.3, su unidad de medida es el ohmio representado por el símbolo (Ω).

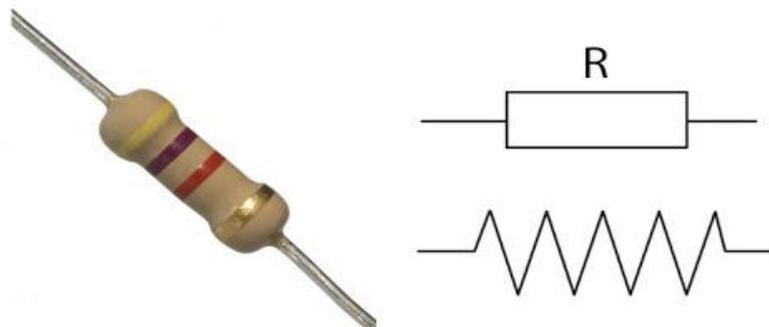


Figura 1.3 Resistencia Eléctrica
Fuente: El autor

1.4 Sensores

Son elementos sensibles a la variación de alguna magnitud física como la fuerza, luminosidad, temperatura, distancia, etc, en la industria son muy usados juntos con transductores para transformar cualquier magnitud física en alguna magnitud medible para que pueda ser aprovechada y pueda trabajar con algún otro dispositivo que interprete estas señales.

Desde hace varios años en el mercado se pueden encontrar varios tipos de sensores, los cuales permiten realizar varios proyectos con variables que no sean constantes. Para este proyecto se han tomado en cuenta los sensores de flexión por su característica principal de variación de resistividad cuando se flexionan los mismos.

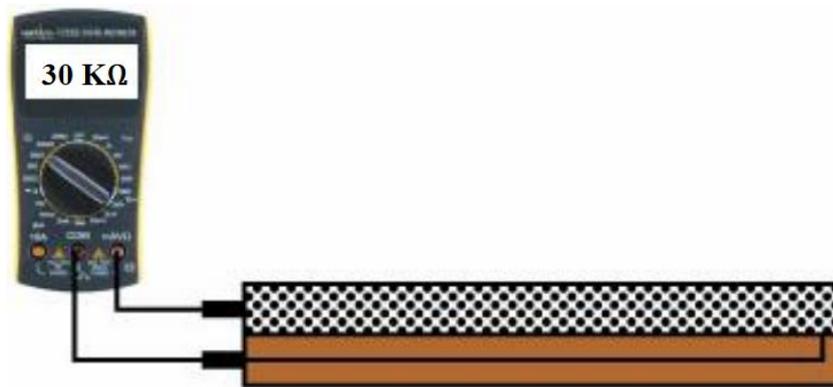
1.4.1 Sensor Flex

Los sensores flex 2.2, pueden ser apreciados en la figura 1.4.1, al ser flexionados generan un cambio en la resistencia eléctrica dentro de sus terminales. Además, son transductores pasivos, es decir necesitan alguna excitación para poder convertir un tipo de energía en otra.

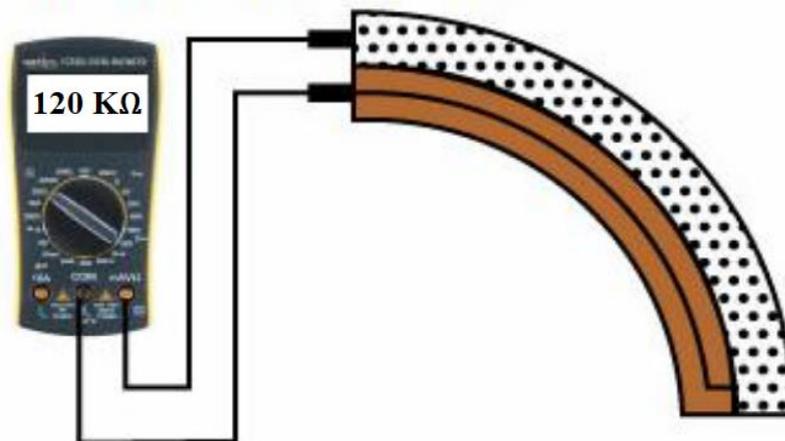


Figura 1.4.1 Sensor Flex
Fuente: LUNAGATE (2016)

El sensor es impreso con una tinta de polímero que tiene partículas conductoras embebidas en él. Cuando el sensor está recto las partículas de la tinta dan una resistencia de aproximadamente $30\text{ K}\Omega$. Cuando el sensor está doblado lejos de la línea, las partículas conductoras se encuentran muy separadas, aumentando la resistencia dando alrededor de $120\text{ K}\Omega$. Cuando el sensor se endereza de nuevo, la resistencia vuelve al valor original. Tal como se lo puede apreciar en la Figura 1.4.2.



a) Partículas conductoras juntas



b) Partículas conductoras separadas

Figura 1.4.2 Comportamiento interno: (a) Partículas conductoras juntas, (b) Partículas conductoras separadas

Fuente: El autor

La fácil adaptación de estos sensores a cualquier superficie hace que su uso sea bien aprovechado en proyectos con guantes electrónicos, ya que estos pueden ir

dispuestos sobre los 5 dedos de la mano. El valor que presente de resistencia eléctrica, dependerá de la flexión que realice cada dedo según la funcionalidad que se le quiera dar. Esta flexión en forma general varía entre valores mínimos de (0 grados) y valores máximos de (90 grados).

El rango de trabajo que presentan estas resistencias variables hace que los niveles de corrientes sean muy seguros para proteger el adecuado funcionamiento de la tarjeta Arduino a la que se ha incorporado, evitando así la incorporación de más resistencias en serie.

1.5 Micro pulsadores

Tiene con finalidad dar paso a la corriente eléctrica cuando se lo oprime y cuando se deja de oprimir este operador eléctrico interrumpe el paso de la corriente (abc, 2006), la imagen de los sensores corresponde a la figura 1.5.

Para este proyecto se ha diseñado la incorporación de estos elementos para ofrecer más precisión a la hora de interpretar un signo del lenguaje de señas, aumentando más variables que puedan determinar con más exactitud el gesto se pretende interpretar.



Figura 1.5 Micro Pulsadores
Fuente: YoRobotics

1.6 Tarjeta Arduino

Arduino es una marca de microcontroladores usada ampliamente en el campo educativo por su fácil adaptación a componentes inmersos en el campo actual de la electrónica y robótica, el símbolo de la tarjeta Arduino está representado en la figura 1.6.



Figura 1.6 Tarjeta Arduino

Fuente: Arduino

1.6.1 Arduino Nano

El proyecto de titulación utiliza una tarjeta Arduino Nano, este Arduino basado en ATmega328 de acuerdo a las especificaciones encontradas en la tienda oficial de Arduino. La placa utilizada se la puede apreciar en la figura 1.6.1. La placa Arduino es la encargada de recibir las señales digitales y analógicas de cada uno de los dispositivos conectados al guante electrónico (sensores flex y micro pulsadores), este se encarga de realizar las acciones que en él se han programado de acuerdo al valor que presenten las variables de los sensores ubicados en el guante electrónico. Este modelo puede comunicarse con un ordenador, otra placa u otros micro-controladores.

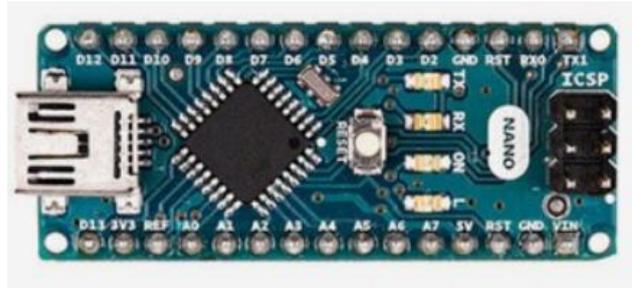


Figura 1.6.1 Arduino Nano
Fuente: Arduino

1.6.2 Características principales

Sus dimensiones son de 18 X 45 mm, requiere una fuente de alimentación de 5V, posee 13 puertos digitales de entrada o salida, 8 entradas analógicas, cristal de 16 MHz, micro controlador ATmega 328, entrada de alimentación por USB o por conector hacia algún adaptador, memoria de EEPROM de 1 KB, memoria flash de 2 KB y memoria SRAM de 2 KB

La distribución de los pines es la siguiente:

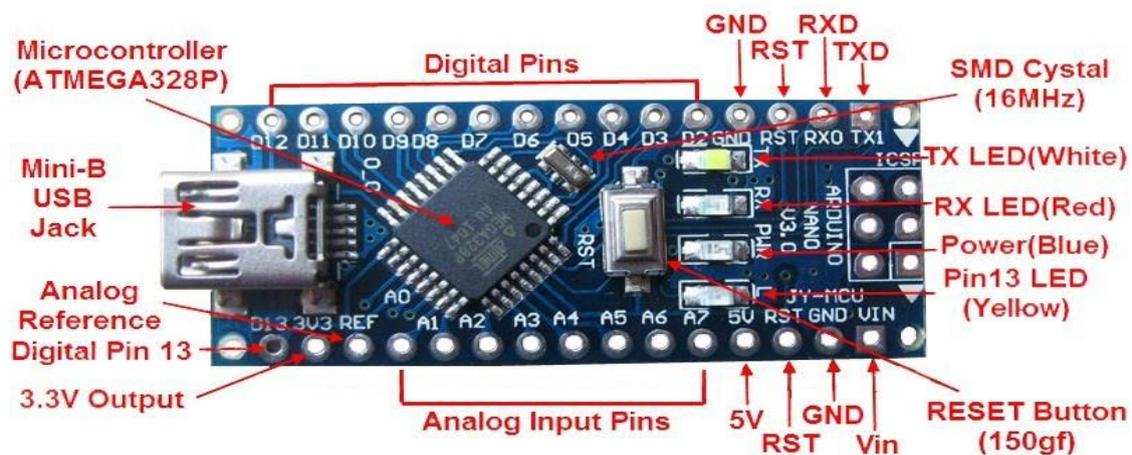


Figura 1.6.2 Características Principales
Fuente: Arduino

1.7 Comunicación Inalámbrica

En la actualidad la electrónica moderna procura que la comunicación entre emisor y receptor sea sin cables, para evitar que el usuario se encuentre atado a cables al momento de usar algún dispositivo electrónico.

Los métodos más comunes de comunicación inalámbrica son con dispositivos tecnológicos como Bluetooth, Zigbee, WIFI, entre otros, los cuales permiten la transferencia de video, voz y datos sin necesidad de cables

1.7.1 Bluetooth

Normalmente usado para transmisión de video, voz y datos en redes inalámbricas de área personal gracias a enlaces por ondas de radio frecuencia en la banda ISM de los 2.4 GHz. Permite la creación de pequeñas redes inalámbricas que posibilitan la sincronización de datos entre dispositivos personales.

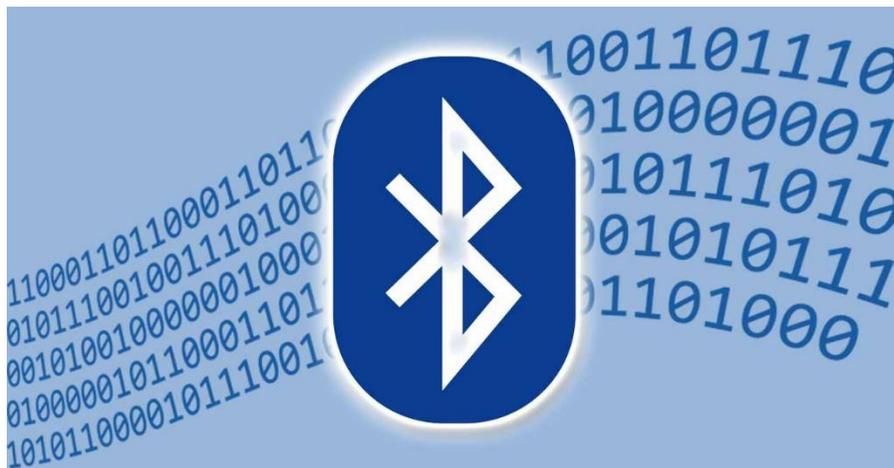


Figura 1.7.1 Bluetooth
Fuente: Adsl Zone

1.8 Módulo Bluetooth Arduino HC-05

Este módulo está fabricado para la comunicación por Bluetooth entre tarjetas Arduino y dispositivos móviles o con otro modulo Bluetooth HC-05, de forma inalámbrica.

Mediante un LED inmerso en el módulo se puede saber si esta emparejado cuando esta encendido de manera continua o si no está emparejado si el LED parpadea. El procedimiento garantiza que el módulo HC-05 encaja en modo AT comandos, y la consola va a servir para programarlo, y así evitar complicaciones.



Figura 1.8 Modulo Bluetooth Arduino HC-05
Fuente: El autor

1.8.1 Características

Posee 6 pines de conexión alimentación 5V (Vcc), tierra (Gnd), transmisión (Tx), recepción (Rx) y su alcance puede llegar a 10 metros en áreas abiertas.

1.9 Programación Arduino

Arduino tiene un software de libre acceso a todo público con funciones sencillas,

pero de gran utilidad, socializando así el uso de esta tecnología la cual traduce líneas de código en tareas de automatización, todo esto sucede con la incorporación de dispositivos electrónicos (sensores, servomotores, luces led, etc) a los diferentes modelos de tarjetas Arduino.

La programación es realizada en un computador y luego grabada en el microprocesador por medio del puerto USB incorporado en las tarjetas para facilidad de comunicación de manera muy práctica.

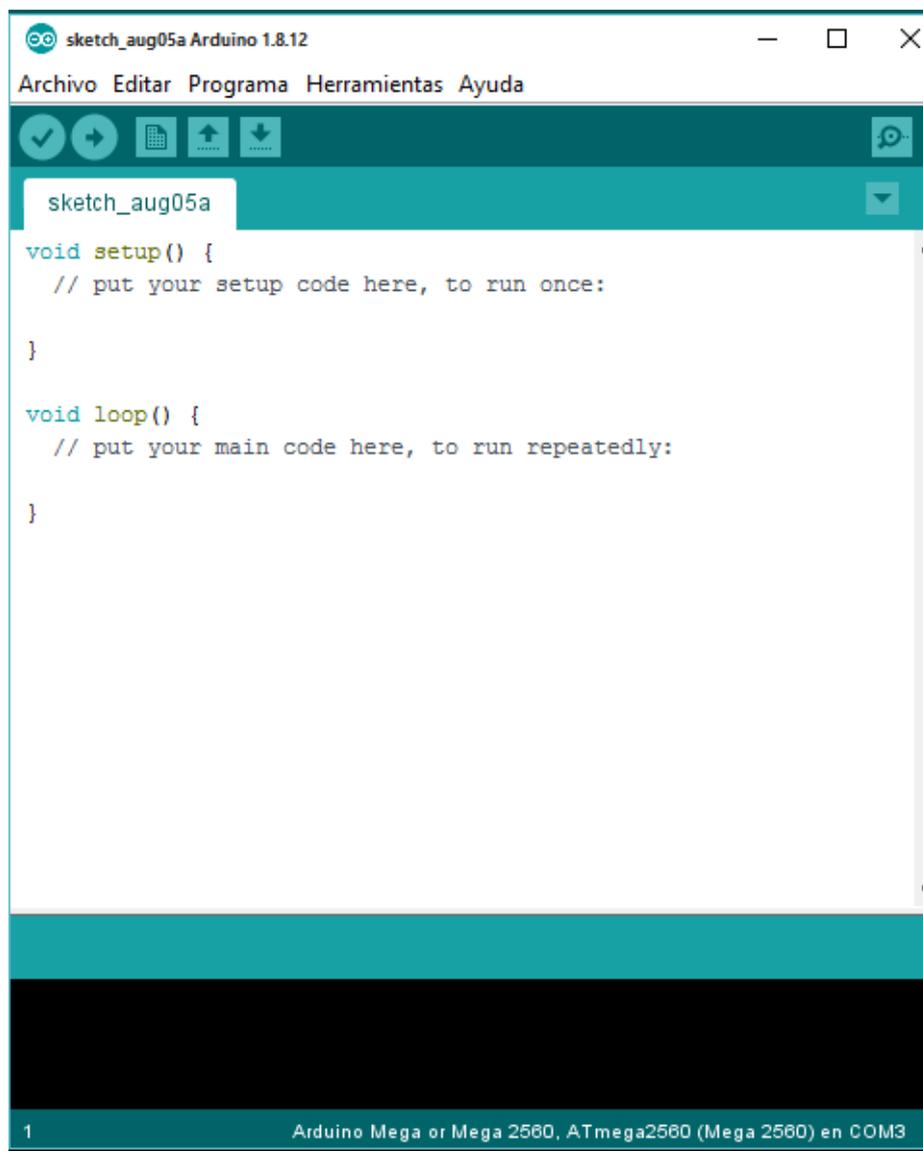


Figura 1.9 Programación de Arduino
Fuente: El autor

1.9.1 Estructura

Esta estructura está compuesta al menos de dos bloques de programación “void setup(){declaraciones}” y “void loop() { instrucciones;}”.que contienen instrucciones o declaraciones

Donde:

1.9.2 Comandos básicos de programación

void setup(){} es la parte encargada de almacenar la configuración de variables como pines de entrada y salidas, ya sean estos digitales o análogos.

```
void setup()  
{  
  estamentos  
}
```

void loop() {} es la parte que almacena la programación que se ejecutará de manera cíclica y en él se realiza la lectura de las entradas y la ejecución según la programación.

```
void loop()  
{  
  estamentos  
}
```

Únicamente con estas dos funciones el programa puede trabajar y ejecutar las instrucciones que en él se programen.

Entre llaves {}

Definen el principio y el final de un bloque de instrucciones y siempre van juntas

puesto que una llave de entrada “{“marca el inicio y una llave de salida”}” marca el final, sino está escrito así al momento de compilar mostrará errores.

Punto y coma;

Son utilizados para separar las instrucciones de programación y también instrucción de tipo “bucle for”.

Bloque de comentarios /*... */

Son un conjunto de párrafos que el programa ignora por lo tanto no ocupa espacio en la memoria del Arduino, pero nos sirve para agregar contenidos que pueden abarcar varias líneas que expliquen más detalladamente alguna parte de la programación que se realice.

Línea de comentarios //

Al igual que el bloque de comentarios son usados para agregar contenidos con información de algo en específico, pero solo lo que abarque una línea de código, es usado muy a menudo después de una instrucción a manera de recordatorio.

Variable

Es una manera de denominar algún valor numérico dentro de la programación asignando algún nombre de referencia y una dirección física según sea el tipo de variable digital o analógica (estos valores pueden variar constantemente). Una variable debe ser declarada al principio y especificar si es de entrada o salida, así mismo puede ser declarada en una serie de lugares del programa y esto determinará en qué momento se hará uso de ella.

Variable de tipo entero (Int)

Si el valor de la variable es de tipo entero usaremos el comando int, este almacena valores numéricos de hasta 16 bits sin decimales.

Ejemplo de declaraciones de variables de tipo entero:

```
int pulgar = A0;
int lecturaPulgar=0;
int indice= A1;
int lecturaIndice=0;
int pulsadorindice = 2;
int lecturaPulsadorindice=0;
```

Pin mode

Sirve para configurar el tipo de trabajo de un pin ya que este puede ser output (salida) o input (entrada), de la siguiente manera:

```
pinMode (pulgar, INPUT);
pinMode (indice, INPUT);
pinMode (medio, INPUT);
pinMode (anular, INPUT);
pinMode (menique, INPUT);
pinMode (pulsadorindice, INPUT);
pinMode (pulsadormedio, INPUT);
pinMode (pulsadoranular, INPUT);
pinMode (pulsadormenique, INPUT);
```

DigitalRead (pin)

Lee valores de los pines asignados como digitales y da como respuesta HIGH (1) o LOW (0)

AnalogRead (pin)

Lee el valor de un pin de entrada definido como analógica con un rango de valor que oscila entre 0 a 1023.

Ejemplo de programación de comandos digitalRead y analogRead

```
lecturaPulgar=analogRead(pulgar);  
lecturaIndice=analogRead(indice);  
lecturaMedio=analogRead(medio);  
lecturaAnular=analogRead(anular);  
lecturaMenique=analogRead(menique);  
lecturaPulsadorindice=digitalRead(pulsadorindice);  
lecturaPulsadormedio=digitalRead(pulsadormedio);  
lecturaPulsadoranular=digitalRead(pulsadoranular);  
lecturaPulsadormenique=digitalRead(pulsadormenique);
```

Delay

Pausa la ejecución del programa en un tiempo definido en ms descritos en la misma instrucción. Así por ejemplo 2000 equivale a 2seg.

Serial.print

Envía un carácter o número, al puerto serie de acuerdo a la previa programación que se haya realizado.

Condicional (if)

El condicional “if” es utilizado normalmente para llevar a cabo operaciones de programación que necesiten respuesta si el valor de alguna variable alcance una magnitud específica, con esto se puede ejecutar una serie de declaraciones descritos dentro de llaves.

Para programar alguna condición es necesario hacer uso de las operaciones de comparación de la siguiente manera:

X >= Y	X es mayor o igual que Y
X <= Y	X es menor o igual que Y
X == Y	X es igual que Y

Ejemplo de programación de condicionales usando if, delay, serial.print y operaciones de comparación

```
{if (lecturaMenique >= 790 && lecturaPulsadorIndice == 0 && lecturaPulsadorMedio == 0)
Serial.print("U");
delay(100);

{if (lecturaMenique <= 790 && lecturaPulsadorIndice == 0 && lecturaPulsadorMedio == 0)
Serial.print("V");
delay(100);}

{if (lecturaMenique <= 790 && lecturaPulsadorIndice == 0 && lecturaPulsadorMedio == 0)
Serial.print("W");
delay(100);}
```

1.10 MIT APP INVENTOR

APP Inventor 2 (AI2) es un Entorno de Desarrollo Integrado (IDE) la segunda generación de APP Inventor administrado por el Instituto de Massachusetts de Tecnología (MIT). AI2 basado en la nube mediante un navegador de Internet.



Figura 1.10 MIT APP INVENTOR 2

Fuente: Posada (2019)

AI2 tiene 2 partes principales: interfaz de usuario y el editor de bloques los cuales son utilizados por la persona que diseña.

1.10.1 Componentes diseñador

Esta representa a ciertas características que contienen ciertos métodos listos para utilizarse dentro de la programación Java como se puede observar en la Figura 1.10.1. El diseñador puede hacer uso de ciertos componentes dentro de estas se ha tomado a:

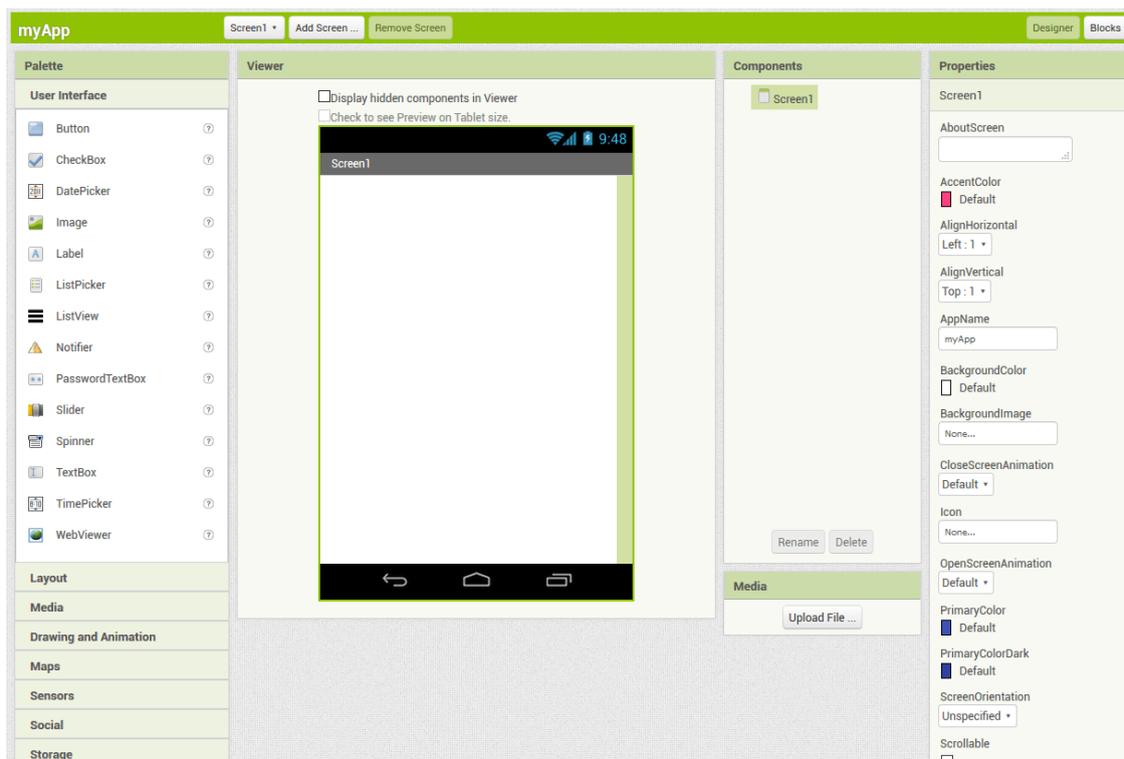


Figura 1.10.1 Componentes Diseñador
Fuente: El autor

1.10.1.1 Interfaz de usuario

Corresponde a la parte visual de la aplicación.

Diseño la presentación que se presenta en la interfaz de usuario y a la vez se puede tener opciones para manejar la aplicación, por ejemplo, los botones, casilleros, etiquetas, opciones para deslizar, Switch, campo de texto, entre otros.

Conectividad, dentro de este se utilizó la opción de cliente la cual permite la conectividad con el dispositivo Android.

1.10.1.2 Editor de bloques

Es un conjunto de bloques que tienen contenidos dentro de los cuales se ramifican las funciones, utiliza rompecabezas de bloques que están disponibles a través de una serie de códigos tal como lo indica la Figura 1.10.1.2.

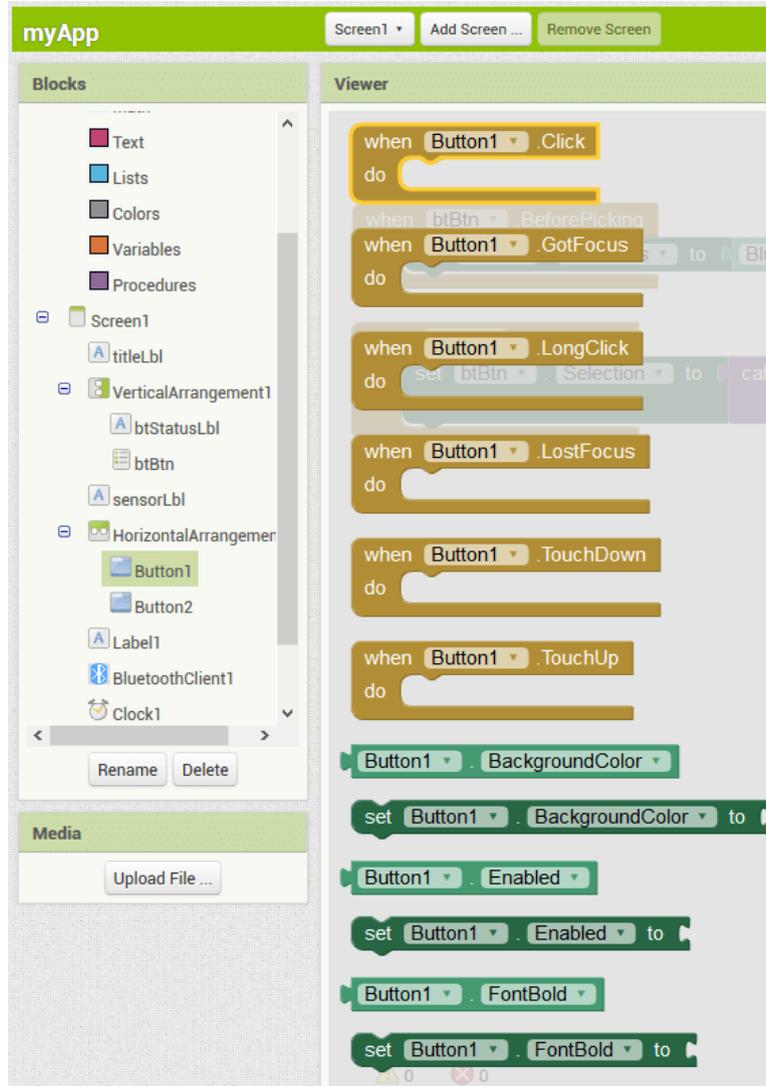


Figura 1.10.1.2 Editor de bloques AI2

Fuente: El autor

1.11 Android

Es un sistema operativo libre el cual se lo utiliza en dispositivos móviles. Las aplicaciones Android se desarrollan en el lenguaje Java es así que al utilizarlo con el AI2 no hay inconveniente dado a que los dos funcionan con el mismo lenguaje de programación.

1.12 Cargador portátil

Fuente alimentación de corriente continua Power Bank Samsung, el cual es capaz de dar voltaje al Arduino y a su vez da voltaje a los dispositivos del guante. También con batería incorporada de alta capacidad, apariencia pequeña, vida útil prolongada y rendimiento de bajo costo.

CAPÍTULO 2

MARCO METODOLÓGICO

Azuero (2019) manifiesta que el marco metodológico se encarga de reconstruir datos con el fin de descubrir los supuestos que se han planteado en el proyecto de investigación. Los objetivos que se plantean en el proyecto servirán como guía para escoger los tipos de datos que resuelvan estas interrogantes formuladas al inicio del trabajo. En otras palabras, se puede decir que es la relación sistemática para recolección, ordenamiento y análisis de la información obtenida, la cual a su vez colaborará con la interpretación de los resultados de la investigación.

2.1 Tipo de investigación

2.1.1 Modalidad

Esta investigación es de campo, y tiene como finalidad que la comunidad sorda encuentre en este primer prototipo electrónico un conductor del lenguaje de señas desde

su mano derecha.

La utilización del guante se realizó en una familia de diez integrantes dentro de los cuales cinco pertenecen a la comunidad sorda, los mismos forman parte del Instituto Nacional de Audición y Lenguaje (INAL). Debido a la contingencia sanitaria de COVID-19, no se utilizó en la población del Instituto.

2.1.2 Diseño de la investigación

El diseño de la investigación es no experimental. Sousa, V., Driessnack, M., & Costa, I. (2007) mencionan que este diseño es capaz de describir la relación que se encuentran entre las variables. Dentro del proyecto encontramos que la implementación del guante puede ayudar a interactuar a la comunidad sorda con la comunidad oyente, es así que, estos grupos de personas son capaces de aprender el lenguaje de señas por medio de dicho guante electrónico. El diseño no experimental se encarga en observar los beneficios de la creación del guante para unir a las personas.

Debido a la contingencia sanitaria COVID-19 es difícil tener acceso al INAL, de este modo, tomando las medidas de precaución se pudo recolectar datos en cierto momento haciendo uso del diseño transaccional de la investigación con la familia antes mencionada.

La implementación del guante se hará de campo dado a que los diez integrantes de la familia podrán hacer uso y con ello probar de primera mano los beneficios que puede brindar el correcto uso de este prototipo.

2.1.3 Enfoque de la investigación

El presente trabajo tendrá un enfoque cuantitativo que es el recomendado para las características y necesidades que tiene la investigación. Este enfoque ayuda a que la investigación pueda ser medida en la base al nivel de satisfacción y beneficio que tuvieron las personas que probaron el guante y a la vez genera información para mejorar el guante electrónico para futuros usos.

2.1.4 Técnica de recolección de datos

En este punto se puede hacer uso de varias técnicas para recoger información sobre el uso y funcionamiento del guante electrónico, pero por situaciones de emergencia sanitaria solamente se usará la técnica de la encuesta.

Debido a que los integrantes de la familia Arce tendrán el guante electrónico en su casa, ellos podrán aportar información sobre su funcionamiento, además que podrán brindar información sobre los beneficios y su respectivo nivel de importancia que el presente proyecto de investigación brinda.

2.2 Fase de desarrollo

2.2.1 Fase I. Selección de los gestos

El proyecto propone traducir treinta y tres gestos que se pueden realizar con la mano derecha. En visitas realizadas al INAL se pudo constatar que se puede recrear el

idioma básico con diez números, con veinte y tres letras del alfabeto, haciendo uso únicamente de una sola mano.

2.2.2 Fase II. Reconocer los componentes

Al analizar los requerimientos de los gestos recolectados se llegó a la conclusión de utilizar sensores de flexión incorporados que sean ubicados en la parte superior de los dedos. El idioma de gestos hace uso de la flexión de los diferentes dedos de la mano para interpretar un símbolo lingüístico. De manera adicional a estos sensores se añadió micro pulsadores para proporcionar más exactitud a la traducción de los gestos requeridos. Todos estos elementos son superpuestos a un guante, este está provisto de dos capas: En la primera capa se encuentran los sensores, cables y micro pulsadores. En la segunda capa se encuentra una capa que se encarga de proteger y cubrir los elementos antes mencionados.

2.2.3 Fase III. Desarrollo del proyecto

Se desmontó el guante, obteniendo de esta forma las dos capas. Sobre la capa interna del guante se colocaron los sensores, micro pulsadores y el respectivo cableado de acuerdo al diseño previo dependiendo de las necesidades que tenían que imitar los gestos de la mano.

Para protección de los sensores flex se fabricó una capa de protección con elementos flexibles, una radiografía, ajustados al tamaño de los sensores de flexión, a la vez que fueron fijados al guate con silicona e hilo de costura. La unión de los sensores y

CAPÍTULO 3

PROPUESTA

3.1 Idea de la propuesta

La idea es crear un guante mediante el uso de AI2, haciendo uso de un celular Android, capaz de traducir en el lenguaje oral y escrito de las diferentes palabras que se realice con un guante, el cual tenga los sensores programados para dichas funciones. El guante puede ser utilizado en el Instituto Nacional de Audición y Lenguaje de la Ciudad de Quito.

3.2 Idea integral de la propuesta

Para la idea integral de la propuesta se diseñará el guante de señas, el cual contiene sensores de flexión, micro pulsadores y conexión mediante cables, todo esto está conectado mediante cables hacia la tarjeta Arduino. La tarjeta Arduino a su vez tiene

incorporado un módulo Bluetooth, el cual permite realizar una comunicación inalámbrica con un dispositivo portátil Android, los cuales garantizan el correcto funcionamiento del guante.

3.3 Planteamiento de una aplicación

Para que el guante pueda hacer la lectura de la posición de los dedos es necesario conocer lo gestos que la comunidad sorda del Ecuador maneja. Una vez comprendidos cuales son estos gestos se deben agregar en la aplicación, AI2.

Para la creación de la interfaz se ha usado la AI2, esta aplicación es gratuita, creada por el Instituto de Massachusetts de Tecnología (MIT). Posterior a la elección de la aplicación se escogió al usuario de interfaz que cubra la necesidad del proyecto. La aplicación cuenta además con una guía para aprender la posición correcta de los símbolos que el lenguaje de señas tiene, para facilitar el uso que las personas puedan darle al guante.

El guante de señas diseñado cuenta con diez números, con veinte y tres letras del abecedario debido a que solo se realiza el guante derecho y dado a que dichos símbolos no cuentan con movimientos.

3.4 Plan de trabajo

Luego de tener un acercamiento con la comunidad sorda, que está representada por temas de investigación en la familia Arce, quienes viven en sector Argelia al Sureste de la ciudad de Quito, se puede constatar que para la creación del guante electrónico se debe aprender los símbolos básicos del lenguaje de señas. Como primer paso se definen

las letras del abecedario y los números, también hay que prestar particular atención a los números y letras que aparentemente tengan los mismos símbolos. Una vez escogida las treinta palabras a traducir, se debe comenzar a trabajar en las conexiones y su correspondiente programación. Finalmente se puede proceder con las pruebas para comprobar el correcto funcionamiento del guante electrónico encargado en traducir el lenguaje de señas en lenguaje escrito y oral.

CAPÍTULO 4

IMPLEMENTACIÓN

4.1 Fabricación de guante electrónico

4.1.1 Desmontaje de guante derecho

Se adquirió un par de guantes de talla L. La presente investigación solo hace uso del guante derecho dado a que es la mano con la que se va a trabajar la traducción del lenguaje de señas.

El guante a trabajar debe estar compuesto por 2 capas; una interna y otra externa: En la capa interna fue factible fijar sensores y micro pulsadores y en la capa externa a su vez tiene la finalidad cubrir los componentes electrónicos instalados para el funcionamiento del guante electrónico de esta forma se lo puede apreciar en la figura 4.1.1.



Figura 4.1.1 Desmontaje de guante
Fuente: El autor

4.1.2 Fijación de componentes al guante

Esta parte del proceso de ensamblaje es importante dado a que la correcta posición de los elementos beneficiará en a la correcta variación del voltaje que se transmite a través de los sensores al ser flexionados.

Sobre la capa interna del guante se fijó con silicona e hilo 5 micro pulsadores, cables y 5 porta sensores Flex, los cuales fueron fabricados con el material de una radiografía reciclada, todo esto de acuerdo al diseño que identificó los requerimientos de las necesidades para interpretar los gestos de la mano.

Se utilizó soldadura y estaño para fijar las conexiones entre dispositivos. Finalmente, para protección de los cables se procedió a cubrirlos con aislante termo encogible. Todos los componentes insertados en la parte interna del guante pueden ser observados en la figura 4.1.2

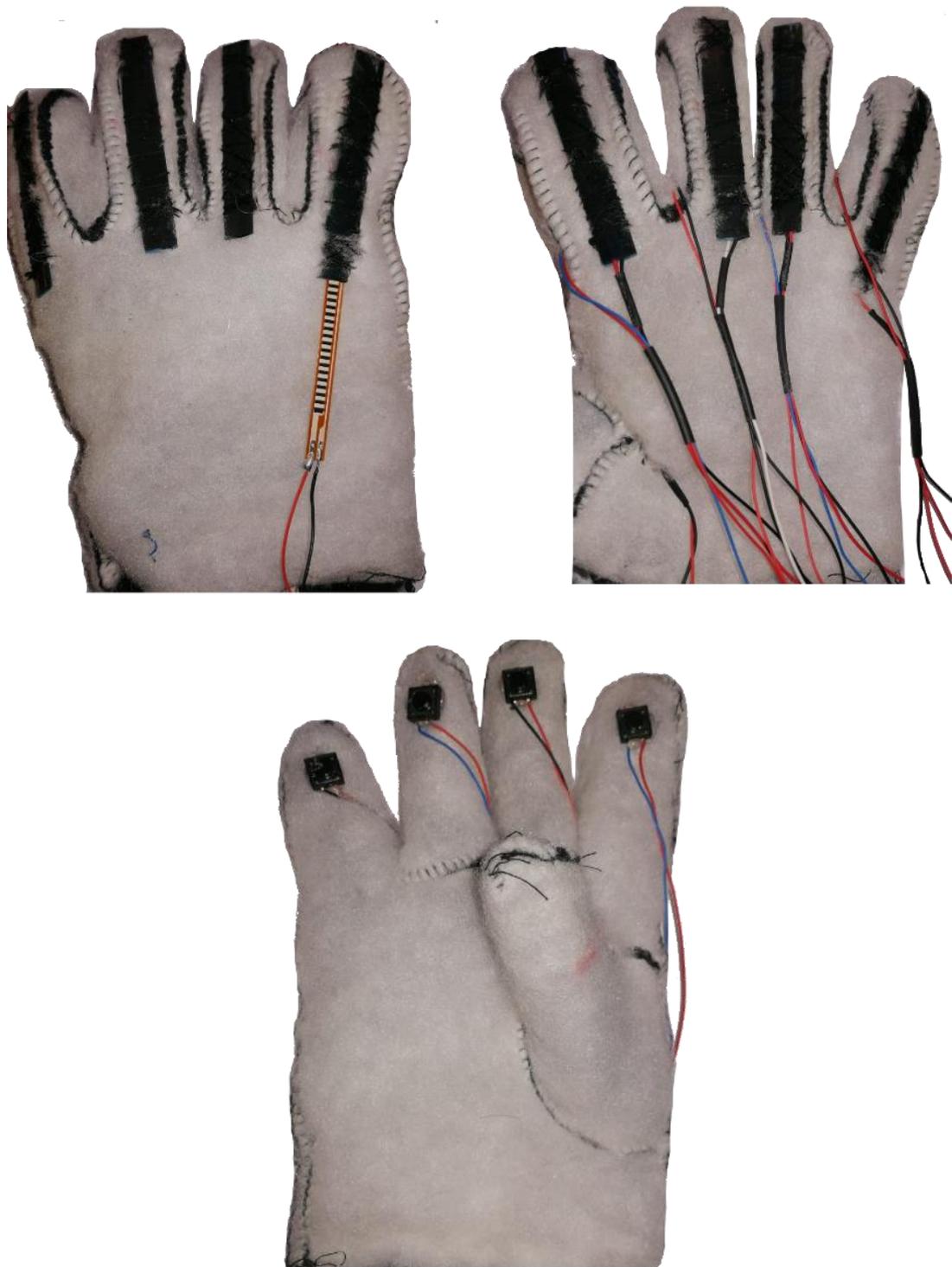


Figura 4.1.2 Fijación de componentes al guante
Fuente: El autor

4.1.3 Diseño de la placa electrónica

Para la fusión de sensores flex, micro pulsadores, Arduino y módulo Bluetooth en el guante electrónico se diseñó y fabricó una placa electrónica tal como se la presenta en la figura 4.1.3.

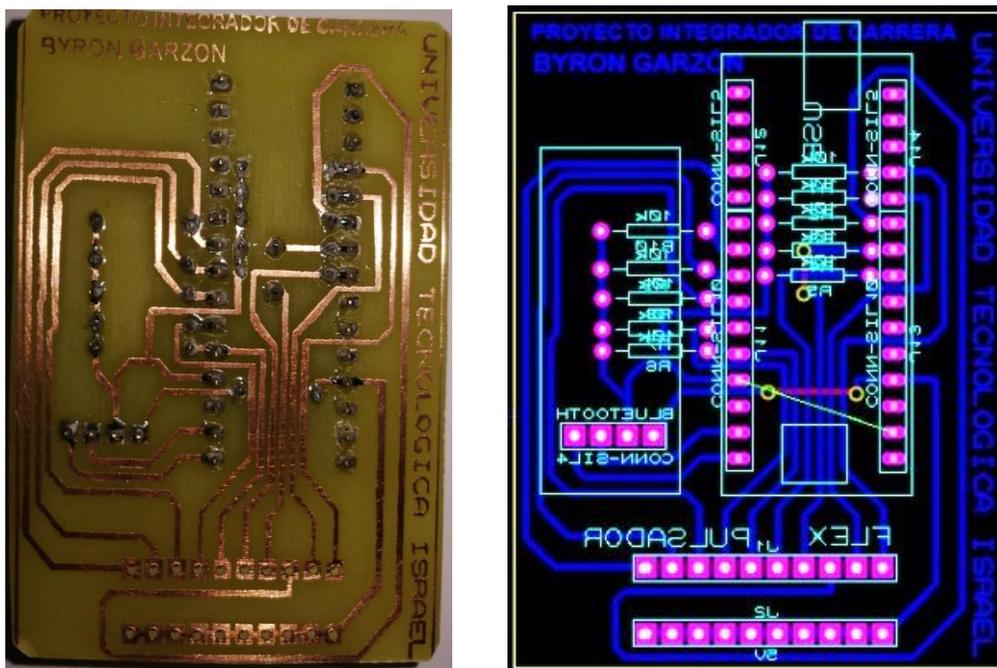


Figura 4.1.3 Interfaz de Usuario
Fuente: El autor

4.1.4 Diseño de caja plástica

Fue necesario diseñar una caja con ciertas medidas específicas. Cumple con la función de proteger la placa electrónica, la tarjeta Arduino y el módulo Bluetooth, de material plástico. Además, esta caja fue impresa en 3D para que encajen perfectamente los elementos mencionados.

Finalmente, se incorporó velcro para que se fije y remueva del guante cuando así

lo requieran, la terminación de esta parte se encuentra ilustrado en la figura 4.1.4.

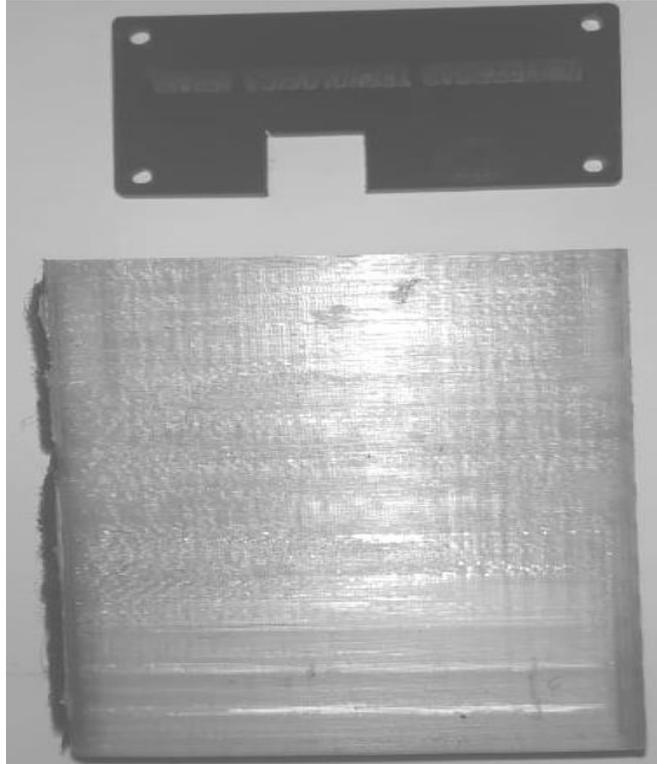


Figura 4.1.4 Caja plástica
Fuente: El autor

4.1.5 Producto

Todos los componentes mencionados anteriormente fueron montados de una manera compacta incluyendo como fuente de alimentación para los dispositivos electrónicos un cargador portátil de celulares.

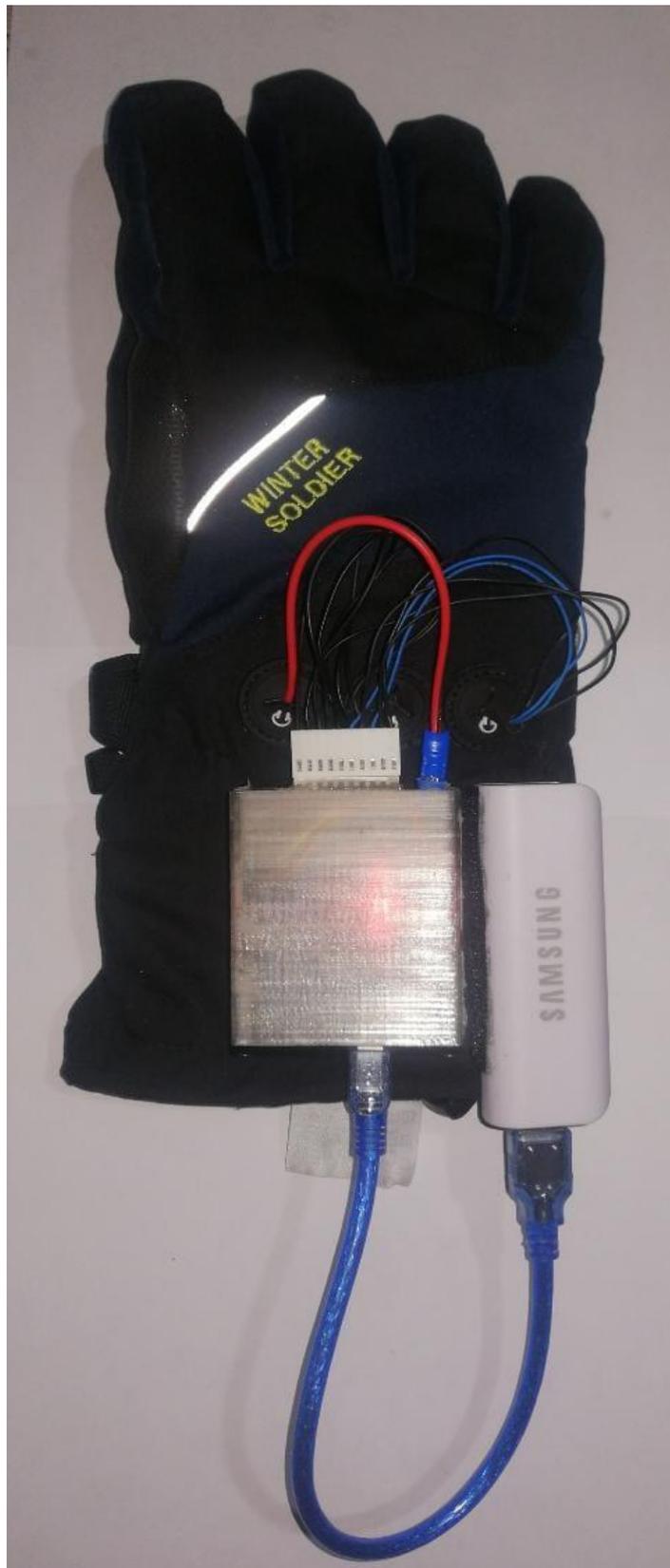


Figura 4.1.5 Producto final
Fuente: El autor

4.2. Cálculos

Consumo de corriente máxima de sensores flex

$$I = \frac{V_{in}}{R_1 + R_2} =$$

$$I_{sensor\ flex} = \frac{5[V]}{30K[\Omega] + 10K[\Omega]} = \mathbf{0,12[mA]}$$

$$I_{sensores\ flex} = 0,12[A] * 5(sensores\ flex) = \mathbf{0,6[mA]}$$

Consumo de corriente de micro pulsadores

$$I_{micro\ pulsador} = \frac{5[V]}{10K[\Omega]} = \mathbf{0,5[mA]}$$

$$I_{micro\ pulsadores} = 0,5[mA] * 5(micro\ pulsadores) = \mathbf{2,5[mA]}$$

Consumo de corriente máxima total

$$\begin{aligned} I_{total} &= I_{micro\ pulsadores} + I_{sensores\ flex} + I_{módulo\ bluetooth} + I_{arduino\ nano} \\ &= 2,5[mA] + 0,6[mA] + 50[mA] + 15[mA] = \mathbf{68,1[mA]} \end{aligned}$$

El valor de consumo eléctrico del módulo bluetooth y arduino nano fue obtenido de fichas técnicas de los dispositivos.

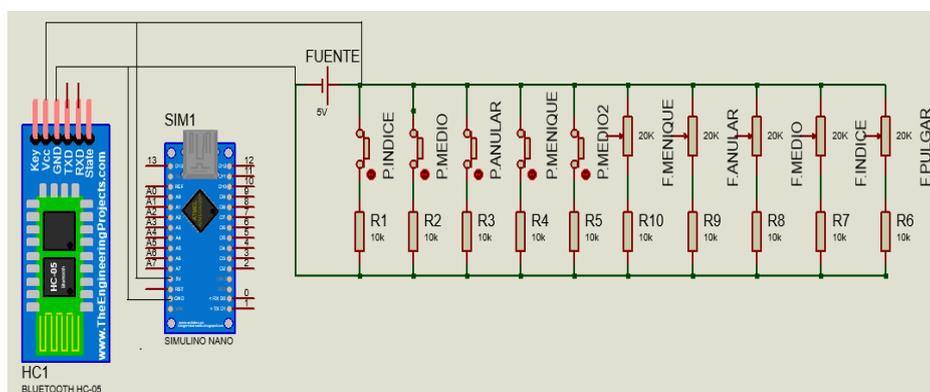


Figura 4.2 Cálculos
Fuente: El autor

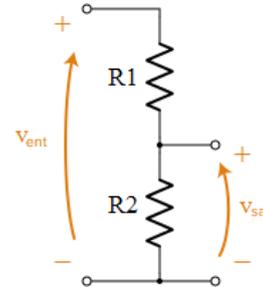
Divisor de voltaje mínimo y máximo

R1=Resistencia Variable **Resistencia Variable** (30 – 120) K[Ω]

$$V_{out} = \frac{R_2}{R_1 + R_2} * V_{in} =$$

$$V_{out_max} = \frac{10K[\Omega]}{30K[\Omega] + 10K[\Omega]} * 5[V] = 1,25[V]$$

$$V_{out_min} = \frac{10K[\Omega]}{120K[\Omega] + 10K[\Omega]} * 5[V] = 0,38[V]$$



Voltaje variable (0,38– 1,25) [V]

4.3 Diagrama de conexiones a la tarjeta Arduino Nano

4.3.1 Conexión del sensor flex

En la figura 4.3.1 se puede apreciar el diagrama de la correcta conexión del sensor flex con la tarjeta Arduino Nano.

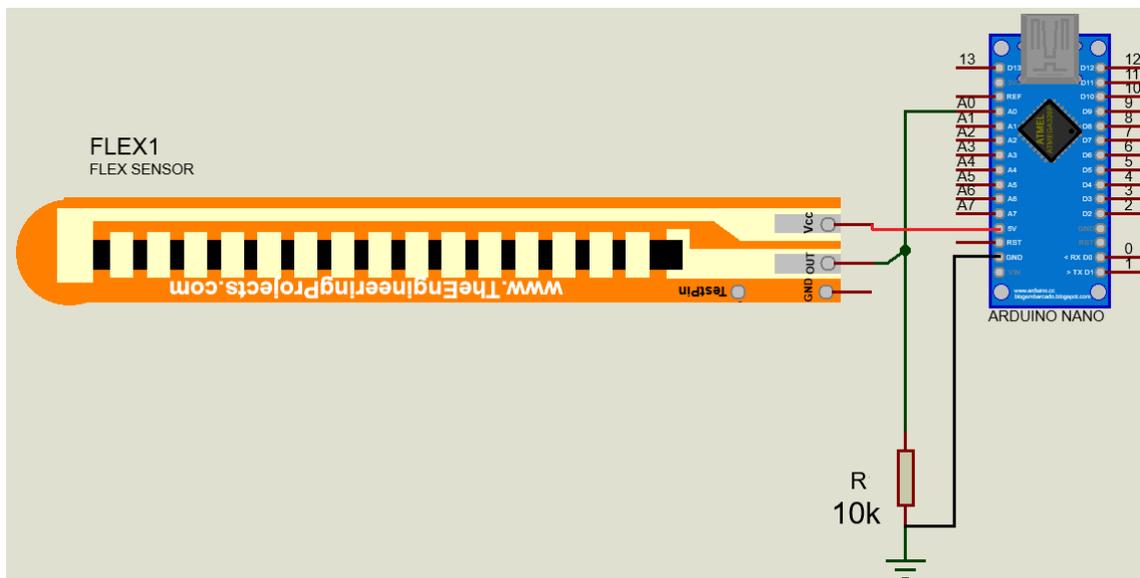


Figura 4.3.1. Conexión del sensor flex

Fuente: El autor

4.3.2 Conexión del micro pulsador

En la figura 4.3.2 se puede apreciar el diagrama de conexión del micro pulsador con la tarjeta Arduino Nano.

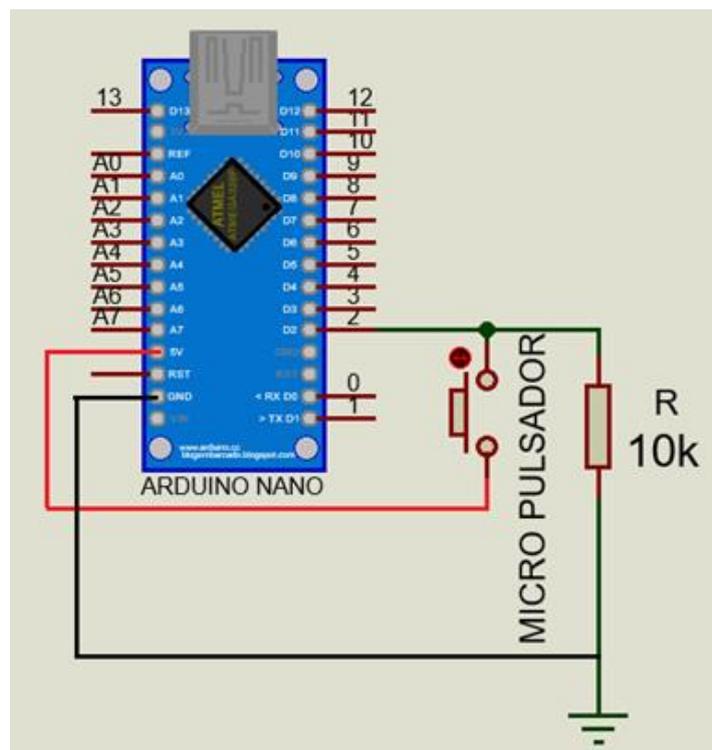


Figura 4.3.2 Conexión de micro pulsador
Fuente: El autor

4.3.3 Conexión del módulo Bluetooth

En la figura 4.3.3 se puede apreciar el diagrama de conexión del módulo Bluetooth con la tarjeta Arduino Nano.

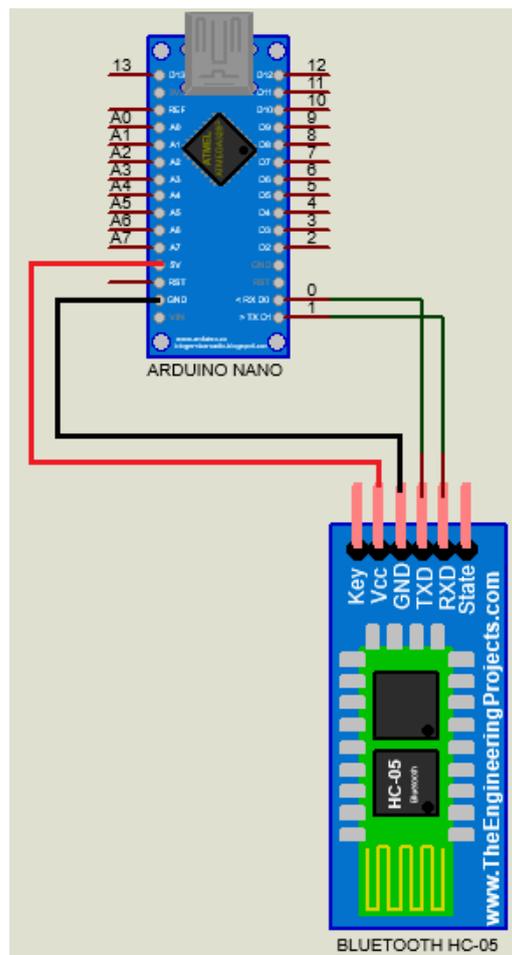


Figura 4.3.3 Conexión del módulo Bluetooth
Fuente: El autor

4.3.4 Conexión de componentes del guante

En la figura 4.3.4 se puede apreciar el diagrama de conexión de todos los dispositivos integrados en el guante electrónico con tarjeta Arduino Nano y módulo Bluetooth HC-05.

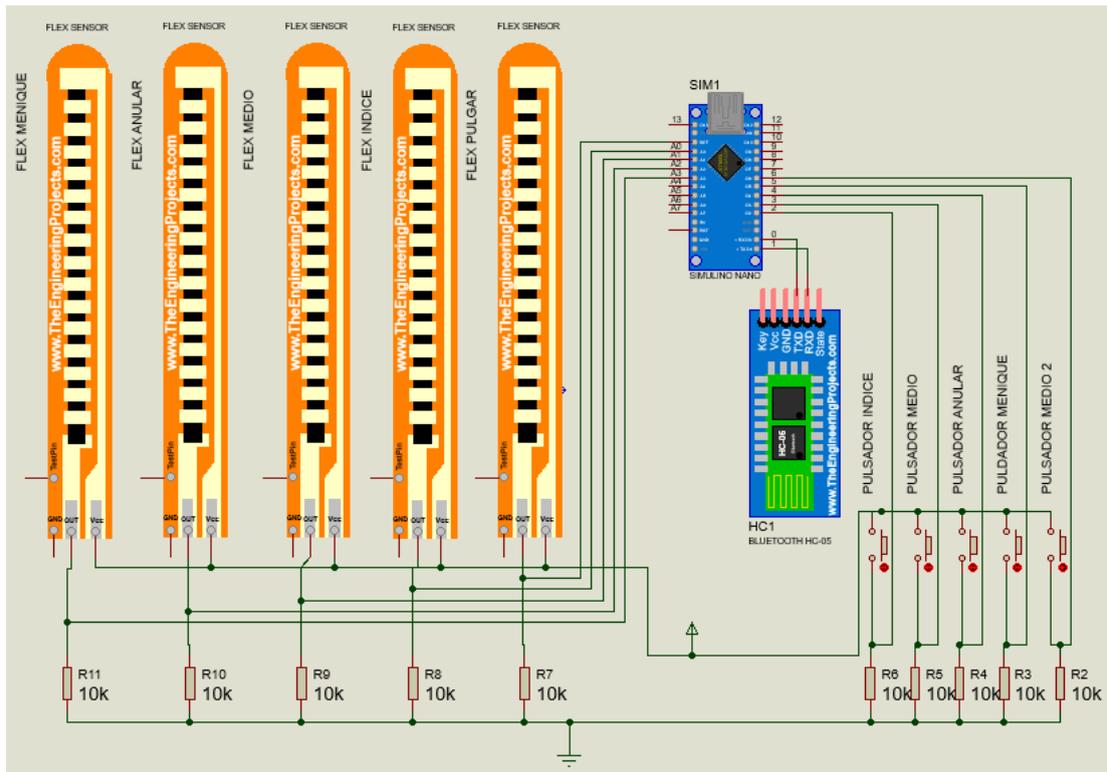


Figura 4.3.4 Conexión de componentes del guante

Fuente: El autor

4.4 Creación de la aplicación con el software en línea AI2

4.4.1 Interfaz de usuario

En concordancia con el diseño de software se ha creado un ícono, el mismo que puede ser visualizado en la figura 4.4.1, este tiene como función acceder a la aplicación y de igual manera sus siglas simbolizan “Traducción de lenguaje de señas” (TraDeLeS) este diseño fue implementado por el autor.

Por otro lado, el ícono representa una persona con problemas de comunicación, el cual ayudado por un cartel puede expresar lo que pretende decir.

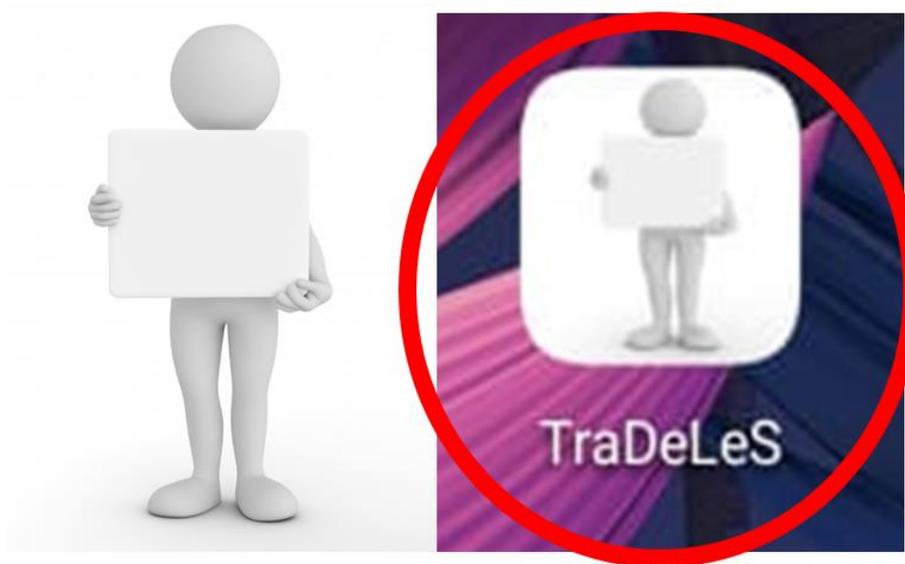


Figura 4.4.1 Interfaz de Usuario
Fuente: El autor

4.4.1.1 Pantalla 1. Menú Principal

Es la pantalla inicial de la aplicación, ofrece un mensaje de bienvenida y muestra cuatro botones táctiles con las siguientes opciones:

- Acerca de
- Aprendizaje
- Traducción
- Salir

Los principales componentes de programación utilizados del software API2 son:

- Botones
- Etiquetas
- Disposiciones de texto o imagen

La interfaz de esta pantalla y la organización de sus componentes se puede observar en la figura 4.4.1.1.

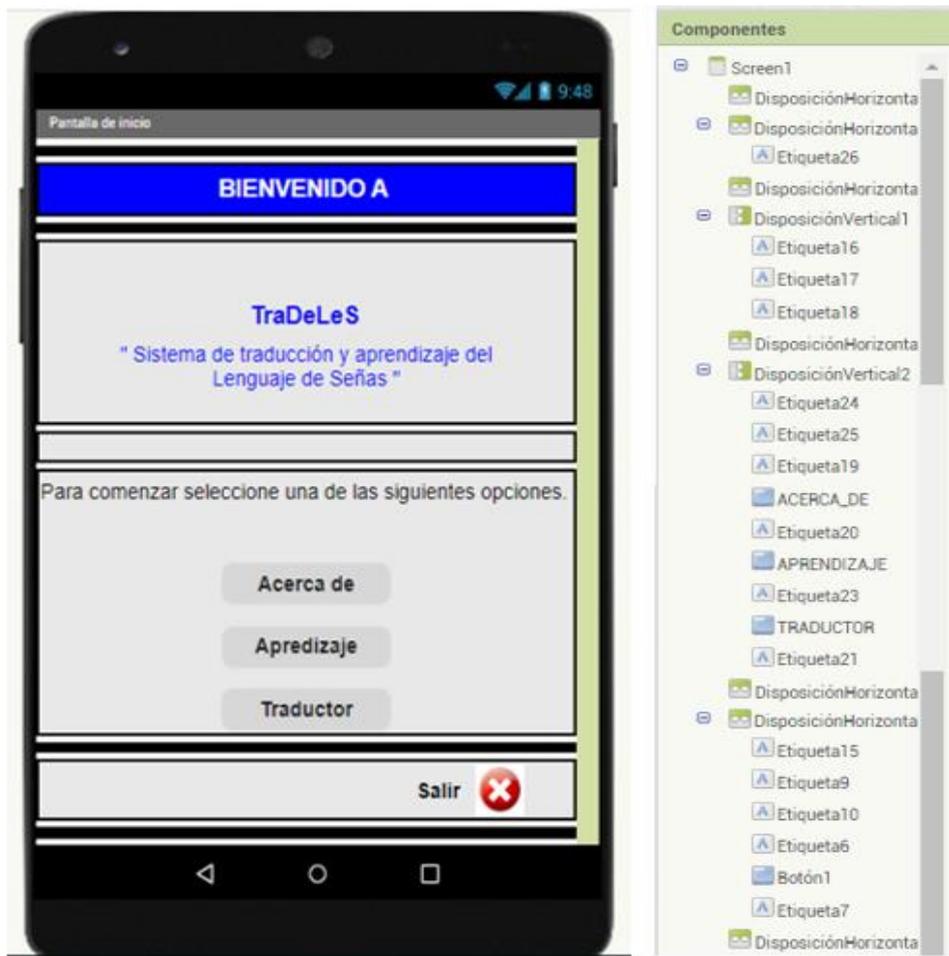


Figura 4.4.1.1 Pantalla 1

Fuente: El autor

4.4.1.2 Pantalla 2. Acerca de

Al dar clic en el botón “acerca de”, en la pantalla principal del menú, aparece la interfaz que se encarga en dar una breve información de que trata la aplicación. La figura 4.3.1.2 muestra de manera más precisa sobre la explicación de la interfaz.

Al final de esta pantalla hay dos botones que dan la opción de salir o regresar a la pantalla anterior.

Los principales componentes de programación utilizados del software API2 son:

- Botones
- Etiquetas
- Disposiciones de texto o imagen

La interfaz de esta pantalla y la organización de sus componentes se puede observar en las siguientes imágenes.

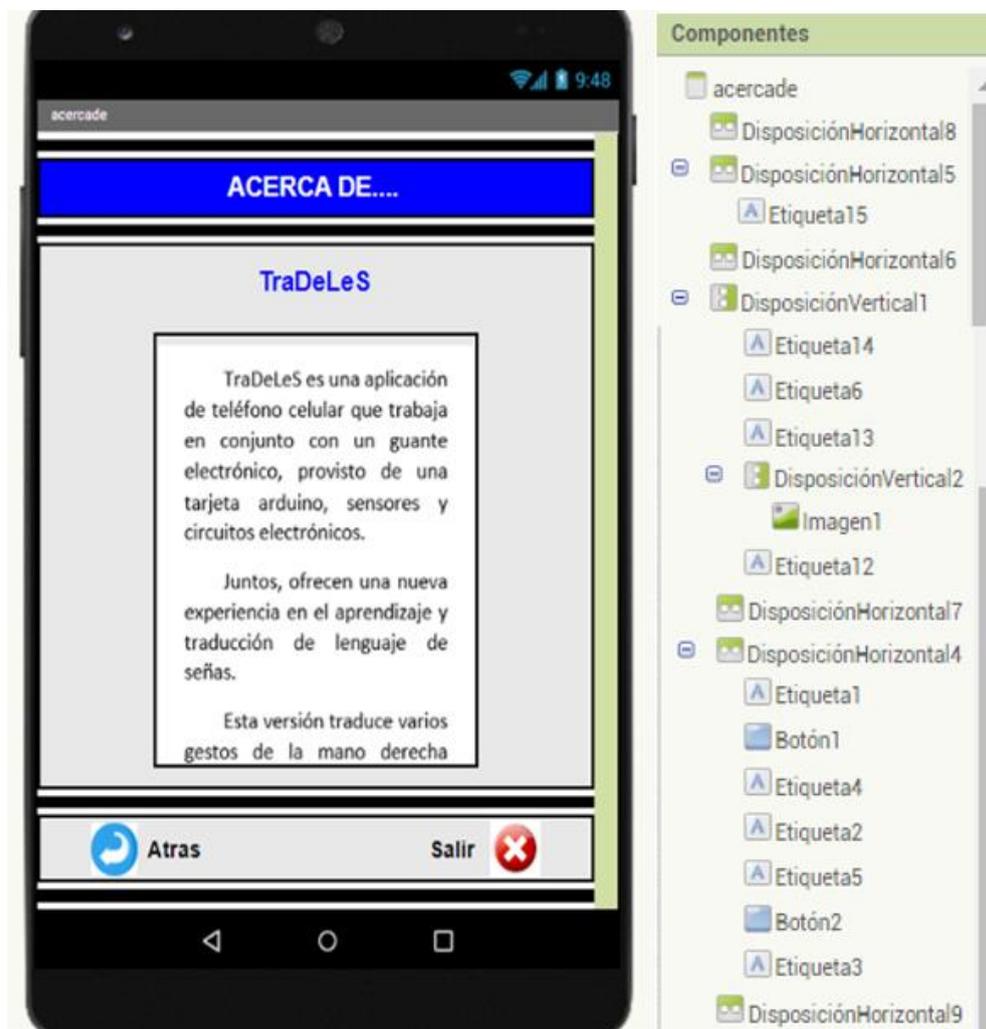


Figura 4.4.1.2 Pantalla 2
Fuente: El autor

4.4.1.3 Pantalla 3. Modo Aprendizaje

Esta pantalla corresponde al botón “modo de aprendizaje”, la cual al hacerle un clic de la pantalla principal del menú aparece una interfaz, a la vez que se puede visualizar cuatro botones principales, tales como:

- Menú
- Atrás
- Salir
- Bluetooth

Al dar clic en el botón menú aparece un submenú que permite la elección de 30 caracteres entre números y letras. Posteriormente al seleccionar el ícono de un carácter inmediatamente aparece en el lado izquierdo superior de la pantalla el símbolo a traducir y al lado derecho una ilustración del gesto en lenguaje de signos que corresponde al carácter escogido por el usuario.

Por otro lado, en esta pantalla al dar clic en la opción “Bluetooth” permite que la conexión sea realizada hacia el guante electrónico y en la parte inferior se muestra la traducción que la tarjeta Arduino ha interpretado.

Los principales componentes de programación utilizados del software AI2 son:

- Botones
- Etiquetas
- Disposiciones de texto o imagen
- Reloj
- Cliente Bluetooth

La interfaz de esta pantalla y la organización de sus componentes se puede

observar en la figura 4.4.1.3.

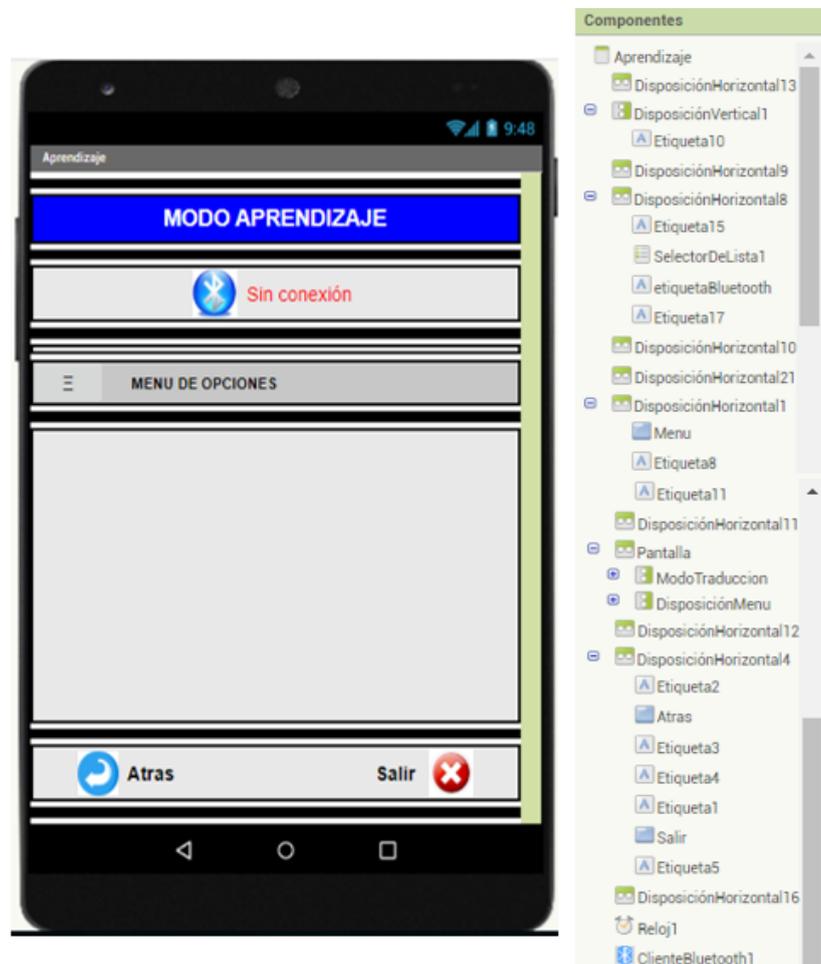


Figura 4.4.1.3. Pantalla 3

Fuente: El autor

4.4.1.4 Pantalla 4. Modo traductor

Al dar clic en el botón “modo de traductor” la pantalla principal del menú aparece una interfaz con cuatro botones principales:

- Bluetooth
- Sonido
- Atrás
- Salir

Al dar clic en el botón menú aparece un submenú que permite la elección de 30

caracteres entre números y letras. Además, al dar clic en algún carácter elegido aparece en el lado izquierdo superior de la pantalla el carácter seleccionado y en lado derecho una imagen del gesto en lenguaje de signos del carácter escogido por el usuario.

Los principales componentes de programación utilizados del software AI2 son:

- Botones
- Etiquetas
- Disposiciones de texto o imagen
- Reloj
- Cliente Bluetooth

La interfaz de esta pantalla y la organización de sus componentes se puede observar en la figura 4.4.1.4.

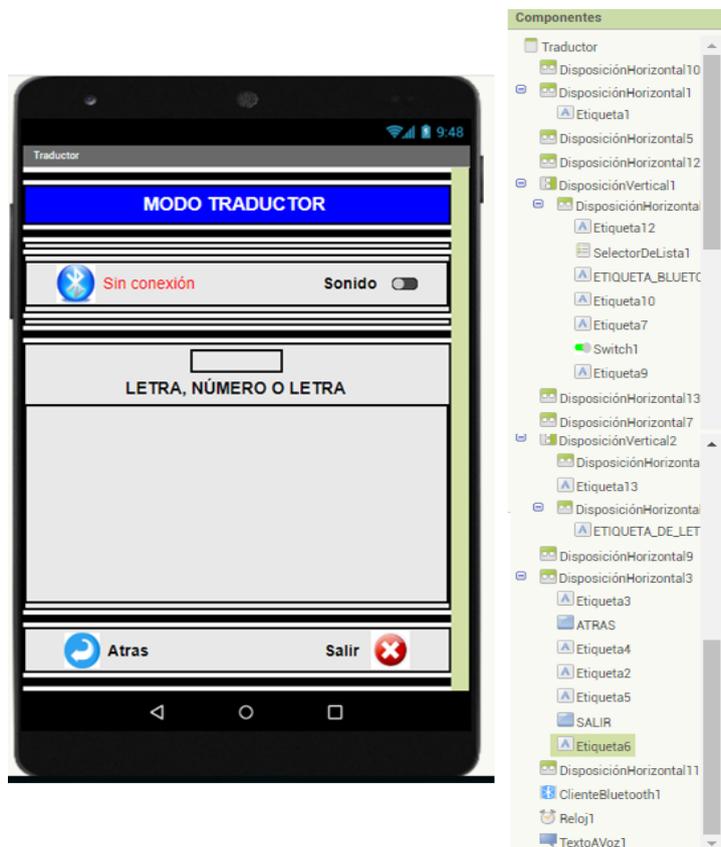


Figura 4.4.1.4 Pantalla 4
Fuente: El autor

4.5 Presupuesto

Tabla 4.5 Presupuesto

	DESCRIPCIÓN	CANTIDAD	VALOR UNIT.	VALOR TOTAL
1	Par de guantes	1	\$20,00	\$20,00
2	Sensore flex	5	\$30,00	\$150,00
3	Micro pulsadores	5	\$0,50	\$2,50
4	Cables	1	\$2,00	\$2,00
5	Estaño	1	\$1,00	\$1,00
6	Tarjeta arduino nano	1	\$10,00	\$10,00
7	Modulo Bluetooth	1	\$8,00	\$8,00
8	Placa electrónica	1	\$10,00	\$10,00
9	Protector plastico para tarjeta arduino y placa electrónica	1	\$15,00	\$15,00
10	Cargador portatil de celular	1	\$10,00	\$10,00
11	Gastos adicionales	1	\$10,00	\$10,00
12	Mano de obra construcción de guante electrónico	1	\$60,00	\$60,00
13	Mano de obra programación arduino	1	\$200,00	\$200,00
14	Mano de obra creación de aplicación	1	\$200,00	\$200,00
			TOTAL	\$698,50

Fuente: El autor

CONCLUSIONES

- Se identificó que los sensores de flexión podían ser implementados en el guante, los cuales tenían movimientos similares a los de la mano derecha. Además, fueron de fácil implementación al momento de asignarle una letra o número a los diferentes voltajes que indicaba el sensor en el AI2.
- Debido a que los materiales son de fácil acceso fue fácil discriminar que tipos de sensores de flexión se iban a utilizar en el guante, los cables para asignar la corriente, y más elementos antes descritos.
- El algoritmo de funcionamiento del dispositivo electrónico fue desarrollado en la plataforma libre de Arduino, permite establecer una interfaz amigable entre el dispositivo y el usuario.
- El dispositivo no pudo ser utilizado en la comunidad educativa debido a las normas de seguridad en contra al COVID19.
- El diseño del software se lo realizó en el APP Inventor 2, este modelo gratuito ayuda a que las aplicaciones puedan ser programadas y contiene un lenguaje de fácil lenguaje de programación, que puede ser utilizado por cualquier persona.
- Al momento de utilizar el guante en la comunidad sordo mudo fue de fácil uso para la familia Arce, a su vez los participantes pudieron corroborar que la posición de los dedos contaba con el número o letra que ellos querían mostrar.

RECOMENDACIONES

- Se debe tomar en cuenta que el envío y recepción de información debe ser entre 4 y 10 metros de distancia del Bluetooth, ya que a mayor distancia se perderá la conectividad.
- Al momento de que los símbolos sean representados con los dedos se debe mantener en esa posición por 2 segundos para que el dispositivo no confunda la orden.
- Debido a la emergencia sanitaria la comunidad sorda que se encontraba en la unidad educativa no pudo seguir con el uso del guante. Es así que se debió implementar con una familia que correspondía a dicha unidad educativa.
- Para los proyectos de este estilo se recomienda realiza un análisis físico y flexibilidad de cables a la vez que se solicita implementar un guante de talla mediana para asegurar que más de una persona pueda probar su funcionamiento.

BIBLIOGRAFÍA

- Acosta, C. (2011). El puente entre dos mundos: Intérpretes de la lengua de señas. Quito.
- Almeida, C., & Viteri, P. (2016 de Mayo de 2016). Repositorio Digital. Recuperado el Mayo de 2019, de <https://repositorio.espe.edu.ec/bitstream/21000/12451/I/T-ESPEL-MEC-0078.pdf>
- Azuero, Á. E. A. (2019). Significatividad del marco metodológico en el desarrollo de proyectos de investigación. Revista Arbitrada Interdisciplinaria Koinonía, 4(8), 110-127.
- Barrantes Echavarría, R. (1999). Investigación: un camino al conocimiento: un enfoque cualitativo y cuantitativo.
- Broches, Pedro, Java 2 Micro Edición: Soporte Bluetooth - 1er Tomo, Edición Electrónica Madrid - España 2004
- Carlos, G. (27 de 07 de 2020). AZ adslzone. Recuperado el 29 de 07 de 2020, de <https://www.adslzone.net/esenciales/windows-10/bluetooth-configurar-activar-conectar/>
- Chacón, E. (2013). Desarrollo de una interfaz para el reconocimiento automático del lenguaje de signos. Sangolquí: Universidad de las Fuerzas Armadas ESPE.
- El Telégrafo. (28 de Septiembre de 2017). Recuperado el Mayo de 2019, de La discapacidad auditiva afecta a 360 millones de personas en el mundo: <https://www.eltelegrafo.com.ec/noticias/sociedad/4/la-discapacidad-auditiva-afecta-a-360-millones-de-personas-en-el-mundo>
- Encalada, L., & Carrasco, H. (11 de Febrero de 2015). Repositorio Digital. Recuperado el Mayo de 2019, de http://repositorio.utn.edu.ec/bitstream/123456789/3765/1/04MEC_054_TESIS.pdf
- Espinosa, P., & Hernán , P. (2013). Diseño y construcción de un guante protipo electrónico capaz de traducir el lenguaje de señas de una persona sordo muda a

lenguaje de letras. Cuenca: Universidad Politécnica Salesiana.

Espinoza, E. (Mayo de 2017). PROYECTO DE LA LEY ORGÁNICA DE MOVILIDAD HUMANA, Y LOS DERECHOS HUMANOS DE LAS PERSONAS CON DISCAPACIDAD EN MIGRACIÓN. Recuperado el Mayo de 2019, de Repositorio Digital: [http://repositorio.iaen.edu.ec/bitstream/24000/4771/1/TESINA Espinosa.pdf](http://repositorio.iaen.edu.ec/bitstream/24000/4771/1/TESINA_Espinosa.pdf) Patricio

FENASEC. (2012). Glosario Básico de Lenguaje de Señas Ecuatoriana. Quito

Gálvez, Segio, Java a Tope: J2Me (Java 2 Micro Edición) - 1era Edición, Edición Electrónica, España - Málaga, 2003

INISTERIO DE RELACIONES LABORALES, Código de Trabajo, Quito 16 de diciembre del 2005

Manjari, S., Monisha, V., Mahalaxme, K., Kalki, G., & Priya, B. (Marzo de 2018). GDR JOURNALS. Recuperado el Mayo de 2019, de https://www.academia.edu/36768483/Speaking_Aid_for_Deaf_and_Dumb_using_Flex_Sensors

Mares Careño, J., Corona Ramirez, L. G., & Abarca Jimenez, G. S. (2014). Sensores y actuadores "Aplicaciones con Arduino". Mexico Distrito Federal: Grupo Editorial Patria.

Monroy, A. (23 de Septiembre de 2018). Con la lengua de señas, todos están incluidos. Recuperado el Noviembre de 2020 de El Telégrafo: <https://www.eltelegrafo.com.ec/noticias/sociedad/6/lengua-senas-inclusion>

Posada Prieto, F. (04 de 2019). MIT App Inventor 2. Recuperado el 29 de Abril de 2020, de <https://intef.es/wp-content/uploads/2019/03/MIT-App-Inventor-2.pdf>

Prado, E. P. (31 de 07 de 2016). LUNAGATE. Recuperado el 27 de Junio de 2020, de <http://www.lunagate.net/2016/07/tutorial-aprender-usar-un-sensor->

flex_31.html#.Xys9rihKjIU

- S/A. (04 de 06 de 2017). Arduino para todos. Recuperado el 25 de Junio de 2020, de <http://arduparatodos.blogspot.com/2017/06/modulo-bluetooth-hc-06-con-arduino-y.html>
- S/A. (17 de 09 de 2017). Arduino y la conexión BlueTooth. Recuperado el 20 de Julio de 2020, de <http://www.bolanosdj.com.ar/MOVIL/ARDUINO2/ConfigHC05.pdf>
- Sousa, V., Driessnack, M., & Costa, I. (2007). Revisión de diseños de investigación resaltantes para enfermería. Parte 1: Diseños de investigación cuantitativa. *Rev latino-am enfermagem*, 15(3), 1-6.
- Taufiq, M., Amalia, A. V., Parmin, P., & Leviana, A. (2016). Design of science mobile learning of eclipse phenomena with conservation insight android-based app inventor 2. *Jurnal Pendidikan IPA Indonesia*, 5(2), 291-298.
- Vera, F. (2020). "Las lenguas signadas como objeto de estudio lingüístico". *Revista de Estudios de Lenguas de Signos REVLES*, 2: 76-92
- Wihidayat, E. S. (2017). Pengembangan aplikasi android menggunakan integrated development environment (IDE) App Inventor-2. *Eduatic-Scientific Journal of Informatics Education*, 4(1).

ANEXOS

ANEXO 1: DIAGRAMA ELÉCTRICO

ANEXO 2: DISEÑO

ANEXO 3: PROGRAMACIÓN ARDUINO

ANEXO 4: DIAGRAMA DE FLUJO DE PROGRAMACIÓN ARDUINO

ANEXO 5: EDITOR DE BLOQUES APP INVENTOR 2

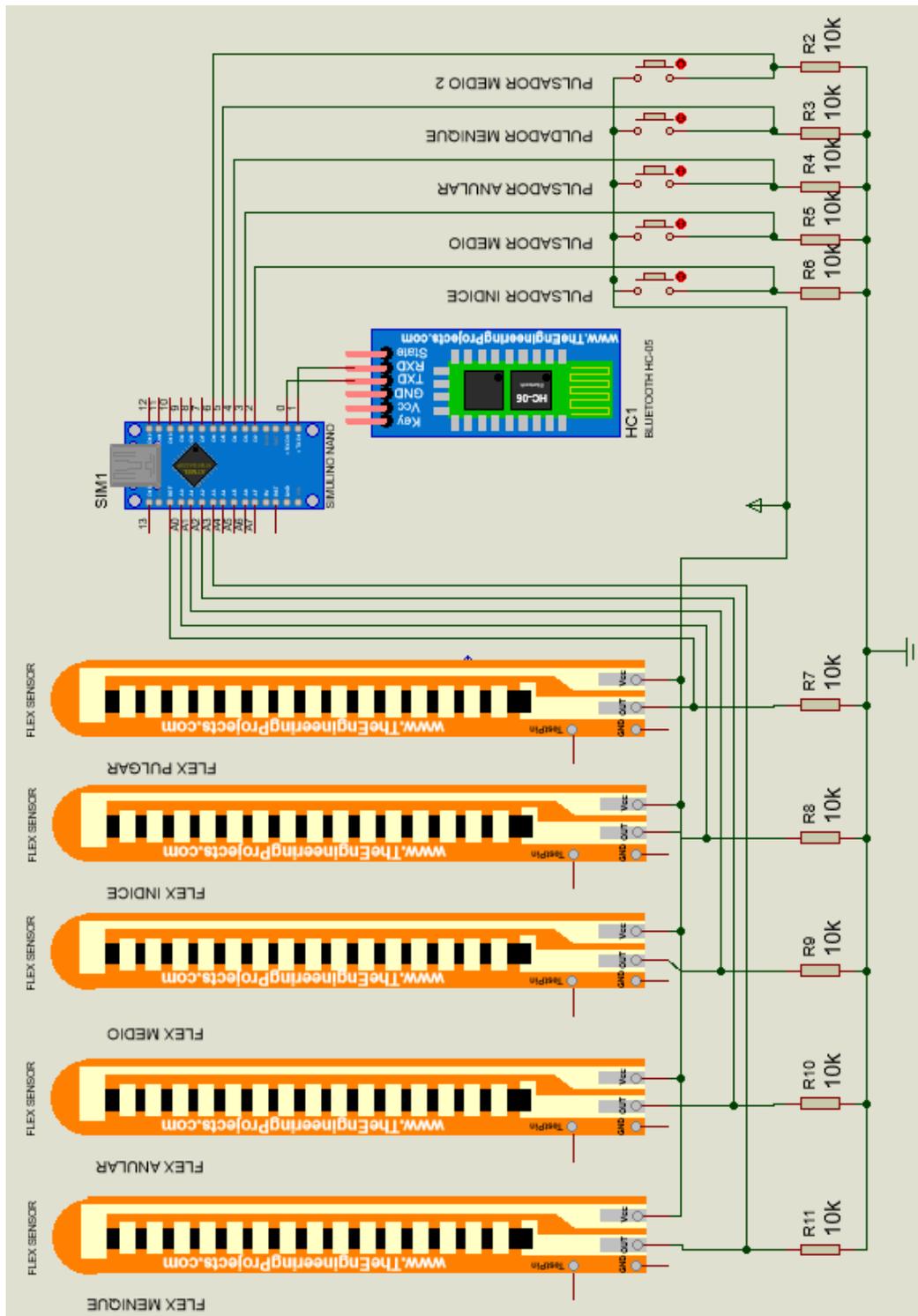
ANEXO 6: MANUAL DE USUARIO

ANEXO 7: FICHA TÉCNICA

ANEXO 7.1 Arduino Nano

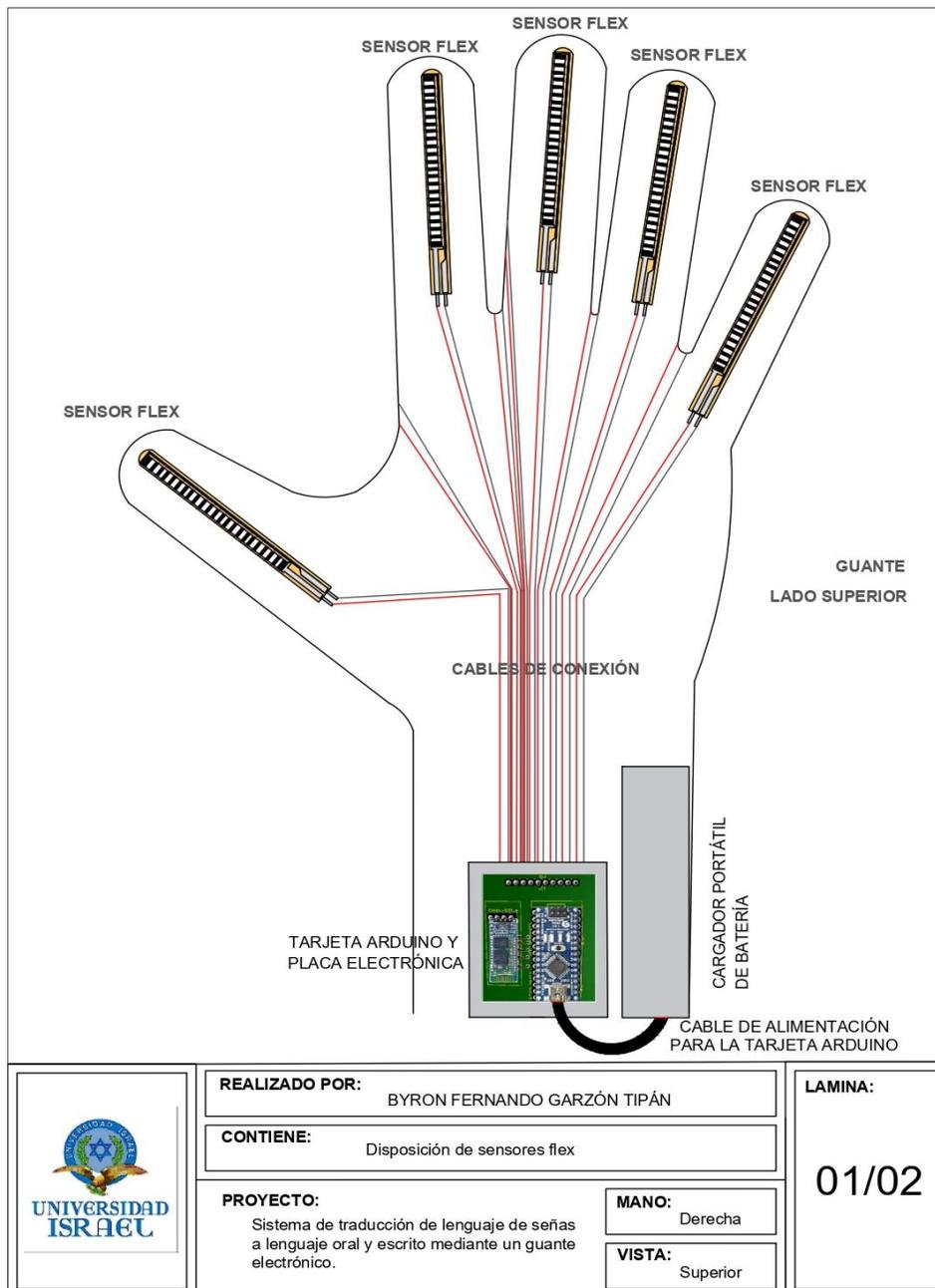
ANEXO 7.2 HC-05

Anexo 1: Diagrama eléctrico

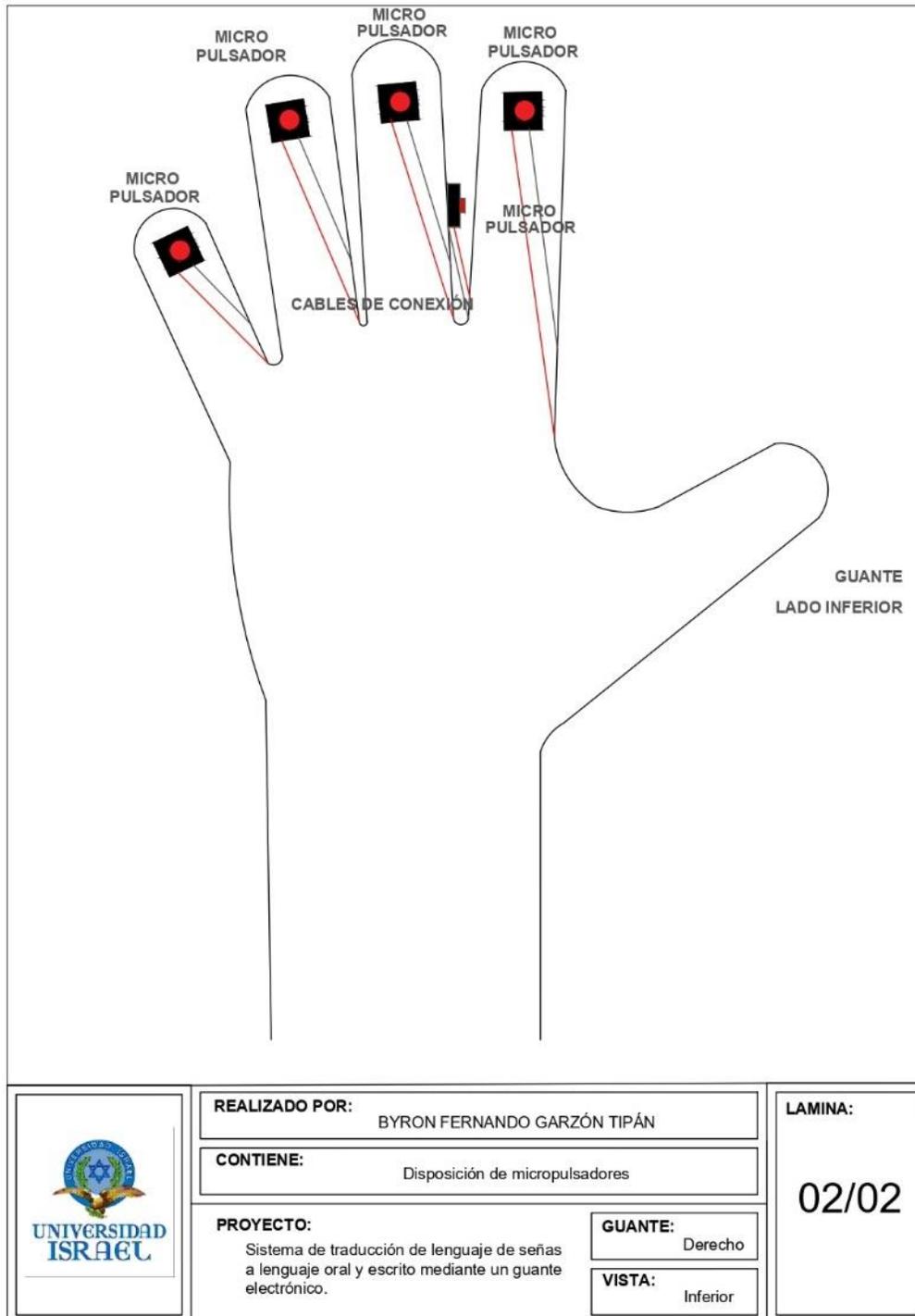


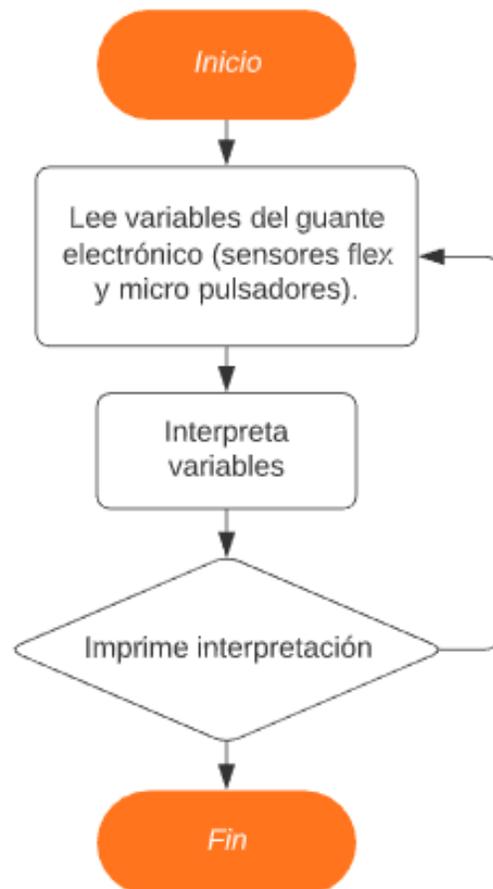
Anexo 2: Diseño

Disposición superior de sensores flex sobre dedos de guante derecho



Disposición inferior de micro pulsadores en dedos de guante derecho



Anexo 3: Diagrama de flujo de programación Arduino

Anexo 4: Programación Arduino

Declaración de variables

Programacion_Tradeles §

```
int pulgar = A0;
int lecturaPulgar=0;
int indice= A1;
int lecturaIndice=0;
int pulsadorindice = 2;
int lecturaPulsadorindice=0;
int medio = A2;
int lecturaMedio=0;
int pulsadormedio = 3;
int lecturaPulsadormedio=0;
int anular = A3;
int lecturaAnular=0;
int pulsadoranular = 4;
int lecturaPulsadoranular=0;
int menique = A4;
int lecturaMenique=0;
int pulsadormenique = 5;
int lecturaPulsadormenique=0;
int pulsadormedio2 = 6;
int lecturaPulsadormedio2=0;
```

Declaración de entradas

Programacion_Tradeles §

```
void setup()
{
  Serial.begin(9600);
  pinMode(pulgar, INPUT);
  pinMode(indice, INPUT);
  pinMode(medio, INPUT);
  pinMode(anular, INPUT);
  pinMode(menique, INPUT);
  pinMode(pulsadorindice, INPUT);
  pinMode(pulsadormedio, INPUT);
  pinMode(pulsadoranular, INPUT);
  pinMode(pulsadormenique, INPUT);
  pinMode(pulsadormedio2, INPUT);
```

Lectura de Variables

Programacion_Tradeles §

```
void loop()
{
  lecturaPulgar=analogRead(pulgar);
  lecturaIndice=analogRead(indice);
  lecturaMedio=analogRead(medio);
  lecturaAnular=analogRead(anular);
  lecturaMenique=analogRead(menique);
  lecturaPulsadorindice=digitalRead(pulsadorindice);
  lecturaPulsadormedio=digitalRead(pulsadormedio);
  lecturaPulsadoranular=digitalRead(pulsadoranular);
  lecturaPulsadormenique=digitalRead(pulsadormenique);
  lecturaPulsadormedio2=digitalRead(pulsadormedio2);
}
```

Programación de Variables

Programacion_Tradeles §

// NÚMEROS

```
{if(lecturaPulgar<=140&&lecturaIndice>=250&&lecturaMedio<=135&&lecturaAnular<=
230&&lecturaMenique<=175&&lecturaPulsadorindice==0&&lecturaPulsadormedio==0&
&lecturaPulsadoranular==0&&lecturaPulsadormenique==0&&lecturaPulsadormedio2==0)
Serial.print("1");
delay(400);}

{if(lecturaPulgar<=140&&lecturaIndice>=250&&lecturaMedio>=250&&lecturaAnular<=
220&&lecturaMenique<=220&&lecturaPulsadorindice==0&&lecturaPulsadormedio==0&
&lecturaPulsadoranular==0&&lecturaPulsadormenique==0&&lecturaPulsadormedio2==0)
Serial.print("2 / V");
delay(400);}

{if(lecturaPulgar>=250&&lecturaIndice>=250&&lecturaMedio>=250&&lecturaAnular<=
240&&lecturaMenique<=240&&lecturaPulsadorindice==0&&lecturaPulsadormedio==0&
&lecturaPulsadoranular==0&&lecturaPulsadormenique==0&&lecturaPulsadormedio2==0)
Serial.print("3");
delay(400);}
```

```

{if(lecturaPulgar<=150&&lecturaIndice>=250&&lecturaMedio>=250&&lecturaAnular>=
250&&lecturaMenique>=250&&lecturaPulsadorindice==0&&lecturaPulsadormedio==0&
&lecturaPulsadoranular==0&&lecturaPulsadormenique==0&&lecturaPulsadormedio2==0)
Serial.print("4");
delay(400);}

{if(lecturaPulgar>=240&&lecturaIndice>=250&&lecturaMedio>=250&&lecturaAnular>=
250&&lecturaMenique>=250&&lecturaPulsadorindice==0&&lecturaPulsadormedio==0&
&lecturaPulsadoranular==0&&lecturaPulsadormenique==0&&lecturaPulsadormedio2==0)
Serial.print("5");
delay(400);}

{if(lecturaPulgar<=2300&&lecturaIndice>=250&&lecturaMedio>=250&&lecturaAnular>
=250&&lecturaMenique<=260&&lecturaPulsadorindice==0&&lecturaPulsadormedio==0
&&lecturaPulsadoranular==0&&lecturaPulsadormenique==1&&lecturaPulsadormedio2==
0)
Serial.print("6");
delay(400);}

{if(lecturaPulgar<=270&&lecturaIndice>=250&&lecturaMedio>=250&&lecturaAnular<=
300&&lecturaMenique>=250&&lecturaPulsadorindice==0&&lecturaPulsadormedio==0&
&lecturaPulsadoranular==1&&lecturaPulsadormenique==0&&lecturaPulsadormedio2==0)
Serial.print("7");
delay(400);}

{if(lecturaPulgar<=300&&lecturaIndice>=250&&lecturaMedio<=250&&lecturaAnular>=
250&&lecturaMenique>=250&&lecturaPulsadorindice==0&&lecturaPulsadormedio==1&
&lecturaPulsadoranular==0&&lecturaPulsadormenique==0&&lecturaPulsadormedio2==0)
Serial.print("8");
delay(400);}

{if(lecturaPulgar<=300&&lecturaIndice<=270&&lecturaMedio>=240&&lecturaAnular>=
250&&lecturaMenique>=250&&lecturaPulsadorindice==1&&lecturaPulsadormedio==0&

```

```

&lecturaPulsadoranular==0&&lecturaPulsadormenique==0&&lecturaPulsadormedio2==0)
Serial.print("9");
delay(400);}

{if(lecturaPulgar<=300&&lecturaIndice<=300&&lecturaMedio<=300&&lecturaAnular<=
330&&lecturaMenique<=310&&lecturaPulsadorindice==1&&lecturaPulsadormedio==0&
&lecturaPulsadoranular==0&&lecturaPulsadormenique==0&&lecturaPulsadormedio2==1)
Serial.print("0 / O");
delay(400);}

//LETRAS

{if(lecturaPulgar>=215&&lecturaIndice<=200&&lecturaMedio<=160&&lecturaAnular<=
250&&lecturaMenique<=180&&lecturaPulsadorindice==0&&lecturaPulsadormedio==0&
&lecturaPulsadoranular==0&&lecturaPulsadormenique==1&&lecturaPulsadormedio2==0)
Serial.print("A");
delay(400);}

{if(lecturaPulgar<=160&&lecturaIndice>=250&&lecturaMedio>=250&&lecturaAnular>=
250&&lecturaMenique>=250&&lecturaPulsadorindice==0&&lecturaPulsadormedio==0&
&lecturaPulsadoranular==0&&lecturaPulsadormenique==0&&lecturaPulsadormedio2==1)
Serial.print("B");
delay(400);}

{if(lecturaPulgar<=260&&lecturaIndice<=280&&lecturaMedio<=290&&lecturaAnular<=
320&&lecturaMenique<=300&&lecturaPulsadorindice==0&&lecturaPulsadormedio==0&
&lecturaPulsadoranular==0&&lecturaPulsadormenique==0&&lecturaPulsadormedio2==1)
Serial.print("C");
delay(400);}

{if(lecturaPulgar<=300&&lecturaIndice>=240&&lecturaMedio<=250&&lecturaAnular<=
300&&lecturaMenique<=270&&lecturaPulsadorindice==0&&lecturaPulsadormedio==1&
&lecturaPulsadoranular==0&&lecturaPulsadormenique==0&&lecturaPulsadormedio2==0)

```

```

Serial.print("D");
delay(400);}
{if(lecturaPulgar<=120&&lecturaIndice<=160&&lecturaMedio<=160&&lecturaAnular<=
250&&lecturaMenique<=170&&lecturaPulsadorindice==0&&lecturaPulsadormedio==0&
&lecturaPulsadoranular==0&&lecturaPulsadormenique==0&&lecturaPulsadormedio2==0)
Serial.print("E");
delay(400);}
{if(lecturaPulgar<=250&&lecturaIndice<=260&&lecturaMedio>=240&&lecturaAnular>=
250&&lecturaMenique>=220&&lecturaPulsadorindice==0&&lecturaPulsadormedio==0&
&lecturaPulsadoranular==0&&lecturaPulsadormenique==0&&lecturaPulsadormedio2==0)
Serial.print("F");
delay(400);}
{if(lecturaPulgar>=215&&lecturaIndice>=220&&lecturaMedio<=170&&lecturaAnular<=
250&&lecturaMenique<=170&&lecturaPulsadorindice==0&&lecturaPulsadormedio==0&
&lecturaPulsadoranular==0&&lecturaPulsadormenique==1&&lecturaPulsadormedio2==0)
Serial.print("G");
delay(400);}
{if(lecturaPulgar<=150&&lecturaIndice<=150&&lecturaMedio<=150&&lecturaAnular<=
270&&lecturaMenique>=240&&lecturaPulsadorindice==0&&lecturaPulsadormedio==0&
&lecturaPulsadoranular==0&&lecturaPulsadormenique==0&&lecturaPulsadormedio2==0)
Serial.print("I");
delay(400);}
{if(lecturaPulgar>=250&&lecturaIndice>=240&&lecturaMedio<=150&&lecturaAnular<=
250&&lecturaMenique<=180&&lecturaPulsadorindice==0&&lecturaPulsadormedio==0&
&lecturaPulsadoranular==0&&lecturaPulsadormenique==0&&lecturaPulsadormedio2==0)
Serial.print("L");
delay(400);}
{if(lecturaPulgar<=160&&lecturaIndice<=210&&lecturaMedio<=210&&lecturaAnular<=

```

```

300&&lecturaMenique<=230&&lecturaPulsadorindice==0&&lecturaPulsadormedio==1&
&lecturaPulsadoranular==1&&lecturaPulsadormenique==0&&lecturaPulsadormedio2==0)
Serial.print("M");
delay(400);}

{if(lecturaPulgar<=205&&lecturaIndice<=200&&lecturaMedio<=240&&lecturaAnular<=
250&&lecturaMenique<=220&&lecturaPulsadorindice==0&&lecturaPulsadormedio==1&
&lecturaPulsadoranular==0&&lecturaPulsadormenique==0&&lecturaPulsadormedio2==0)
Serial.print("N");
delay(400);}

{if(lecturaPulgar<=170&&lecturaIndice>=250&&lecturaMedio>=210&&lecturaAnular<=
210&&lecturaMenique<=190&&lecturaPulsadorindice==0&&lecturaPulsadormedio==1&
&lecturaPulsadoranular==0&&lecturaPulsadormenique==0&&lecturaPulsadormedio2==0)
Serial.print("R");
delay(400);}

{if(lecturaPulgar<=160&&lecturaIndice<=160&&lecturaMedio<=160&&lecturaAnular<=
260&&lecturaMenique<=170&&lecturaPulsadorindice==1&&lecturaPulsadormedio==1&
&lecturaPulsadoranular==1&&lecturaPulsadormenique==1&&lecturaPulsadormedio2==0)
Serial.print("S");
delay(400);}

{if(lecturaPulgar>=250&&lecturaIndice<=250&&lecturaMedio>=250&&lecturaAnular>=
260&&lecturaMenique>=250&&lecturaPulsadorindice==0&&lecturaPulsadormedio==0&
&lecturaPulsadoranular==0&&lecturaPulsadormenique==0&&lecturaPulsadormedio2==0)
Serial.print("T");
delay(400);}

{if(lecturaPulgar<=160&&lecturaIndice>=250&&lecturaMedio>=250&&lecturaAnular<=
250&&lecturaMenique<=220&&lecturaPulsadorindice==0&&lecturaPulsadormedio==0&
&lecturaPulsadoranular==0&&lecturaPulsadormenique==0&&lecturaPulsadormedio2==1)
Serial.print("U");

```

```

delay(400);}
{if(lecturaPulgar<=170&&lecturaIndice>=240&&lecturaMedio>=260&&lecturaAnular>=
260&&lecturaMenique<=170&&lecturaPulsadorindice==0&&lecturaPulsadormedio==0&
&lecturaPulsadoranular==0&&lecturaPulsadormenique==0&&lecturaPulsadormedio2==0)
Serial.print("W");
delay(400);}
{if(lecturaPulgar<=210&&lecturaIndice>=200&&lecturaMedio<=150&&lecturaAnular<=
240&&lecturaMenique<=220&&lecturaPulsadorindice==0&&lecturaPulsadormedio==1&
&lecturaPulsadoranular==0&&lecturaPulsadormenique==0&&lecturaPulsadormedio2==0)
Serial.print("X");
delay(400);}
{if(lecturaPulgar>=240&&lecturaIndice<=230&&lecturaMedio<=190&&lecturaAnular<=
300&&lecturaMenique>=240&&lecturaPulsadorindice==0&&lecturaPulsadormedio==0&
&lecturaPulsadoranular==0&&lecturaPulsadormenique==0&&lecturaPulsadormedio2==1)
Serial.print("Y");
delay(400);}
}

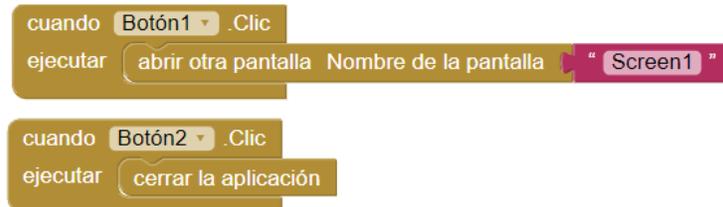
```

Anexo 5: Editor de bloques APP inventor 2

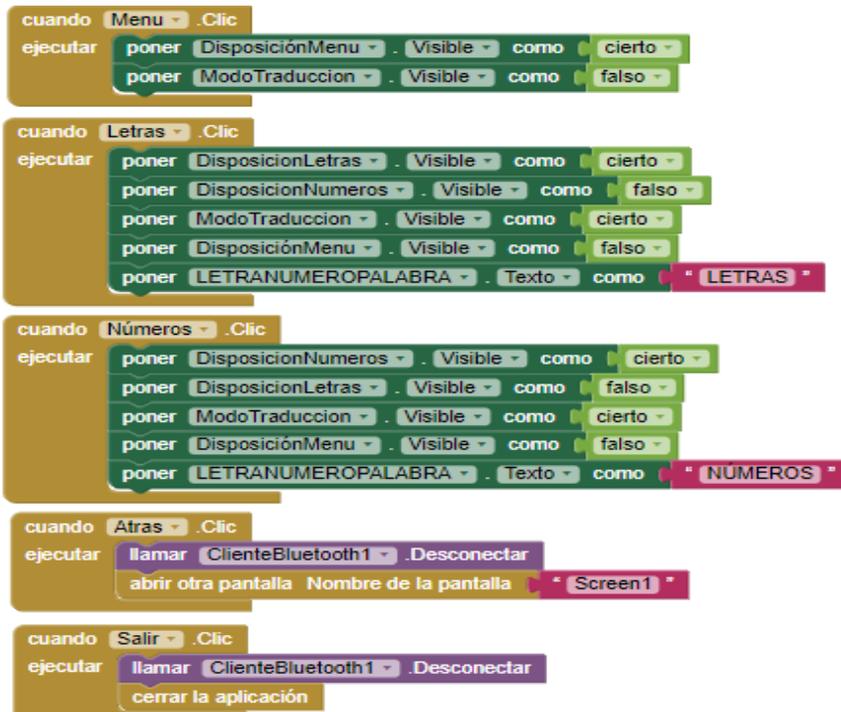
Pantalla 1. Menú Principal

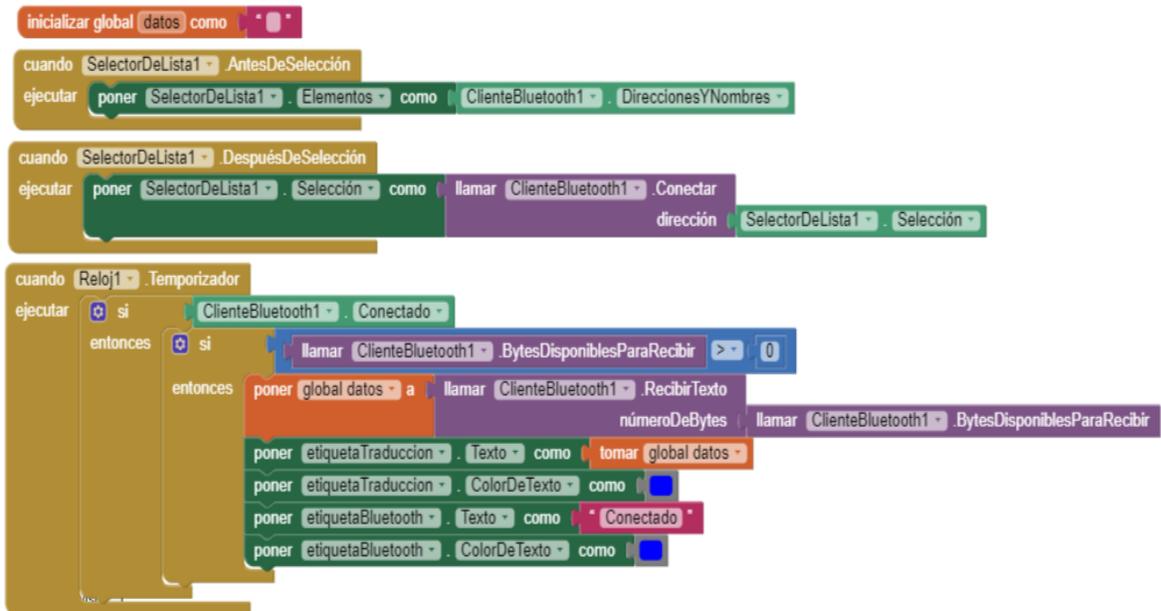


Pantalla 2. Acerca de



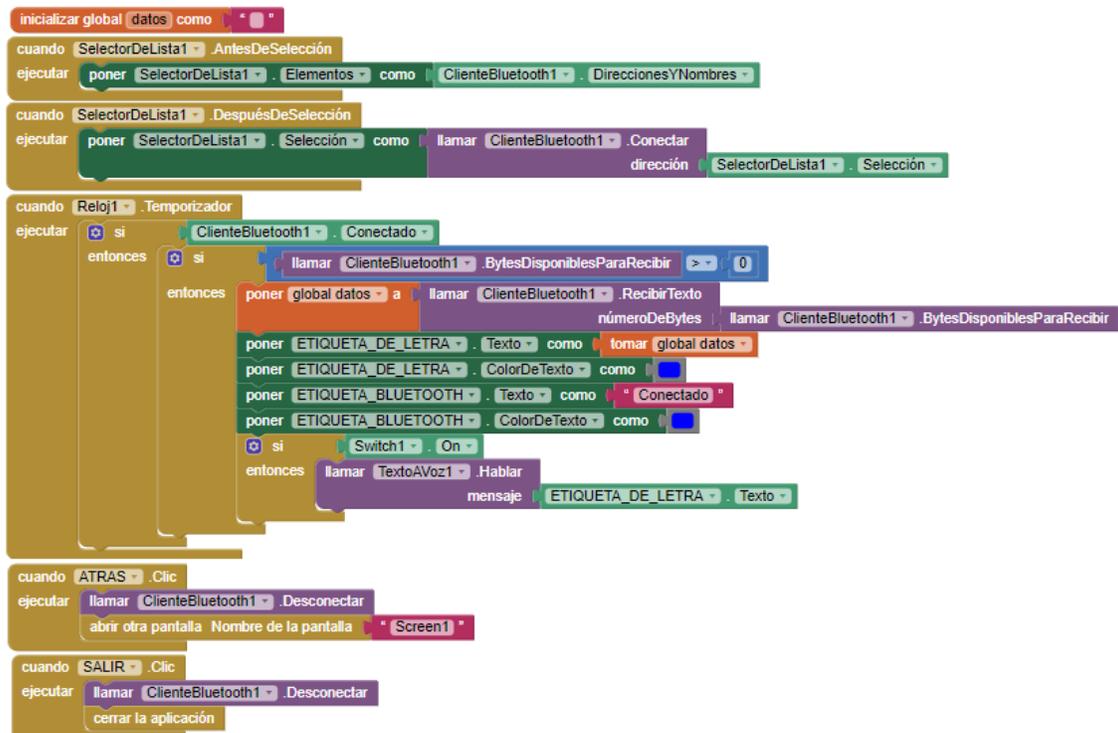
Pantalla 3. Modo Aprendizaje







Pantalla 4. Modo Traductor



Anexo 6: Manual de Usuario

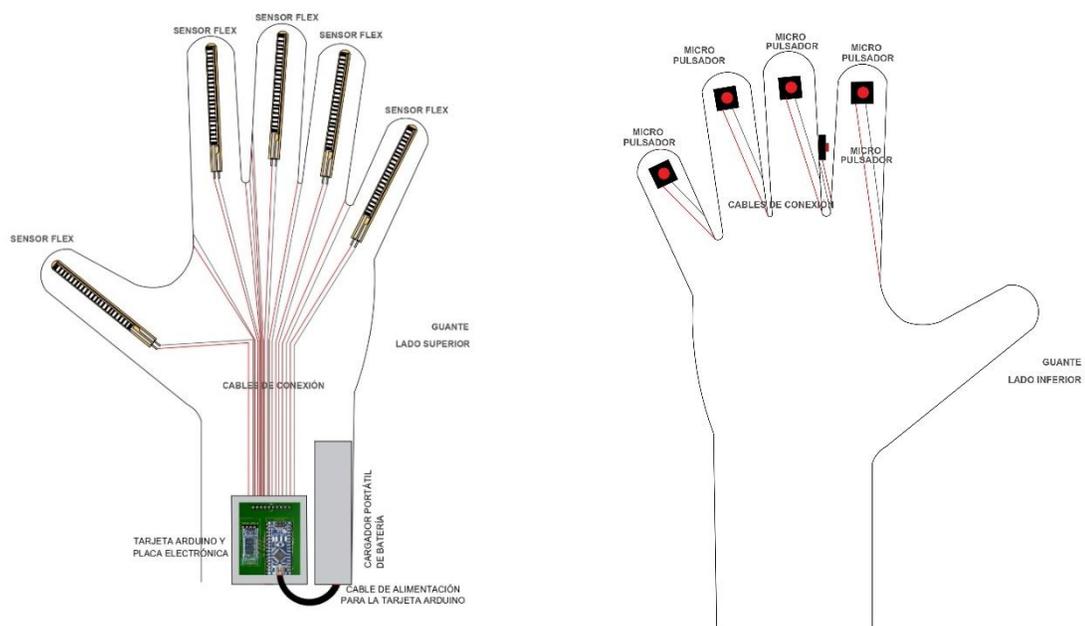


SISTEMA DE TRADUCCIÓN DE LENGUAJE DE SEÑAS A LENGUAJE ORAL Y ESCRITO MEDIANTE UN GUANTE ELECTRÓNICO.

Manual de Usuario

Versión: 1.0

2020/08



DESCRIPCIÓN DEL SISTEMA

1.1 Objeto

Ofrecer al usuario una nueva experiencia en el aprendizaje y traducción de lenguaje de señas, con ayuda de tecnología en este caso con una aplicación para dispositivos android y dispositivos electrónicos dispuestos sobre un guante.

1.2 Alcance

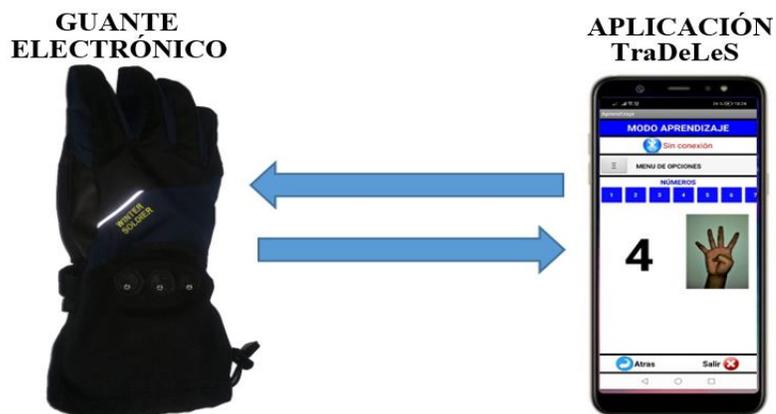
Esta versión 1.0 traduce varios gestos de la mano derecha entre letras y números que en conjunto con el guante electrónico, son capaces de traducir 30 caracteres entre números y letras del abecedario.

1.3 Funcionalidad

Esta aplicación ofrece las opciones de traducir y aprender el lenguaje de señas, la aplicación sirve como soporte para las personas que estén iniciándose en el idioma de lenguaje de señas, brindando la posibilidad de observar las guías fotográficas de signos y poder realizar traducciones de los gestos al mismo tiempo.

2 MAPA DEL SISTEMA

2.1 Modelo Lógico



3 NAVEGACION

PANTALLA INICIAL

En la pantalla tiene las siguientes opciones: Acerca de, Modo Aprendizaje, Modo Traductor.



ACERCA DE MODO APRENDIZAJE

Al dar clic en esta opción se despliega una pantalla, la misma que proporciona una breve información sobre la aplicación.



Al dar clic en esta opción se despliega una pantalla que proporciona un menú para elegir entre números y letras.



Se debe pulsar únicamente una vez para que aparezca un submenú en la pantalla



Al dar clic en la opción Números se desplegará un menú en el mismo que podrá elegir entre los números del 0 al 9 con su respectiva en el lenguaje de signos.





Al dar clic en la opción Letras se desplegará un menú en el mismo que podrá elegir entre las letras del a, b, c, d, e, f, g, i, l, m, n, o, r, s, t, u, v, w, x, y. con su respectiva en el lenguaje de signos.





MODO TRADUCTOR

Al dar clic en esta opción se despliega una pantalla en la que aparecerá la traducción del signo que se esté interpretando en el momento, previa conexión por Bluetooth con el guante electrónico.



LETRA, NÚMERO O LETRA

Para escuchar la traducción se debe realizar clic en el botón de sonido en la pantalla, después de esto debería cambiar de color a verde y empezar a escucharse la traducción.



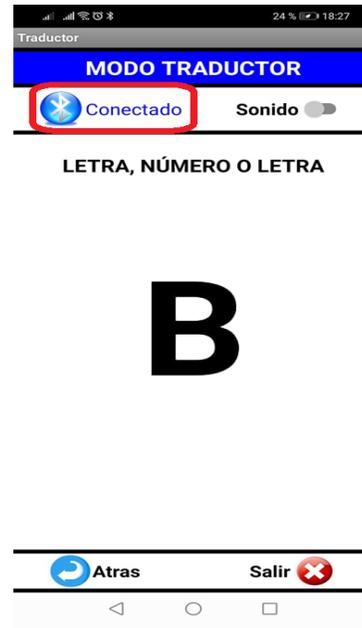
LETRA, NÚMERO O LETRA

CONEXIÓN BLUETOOTH

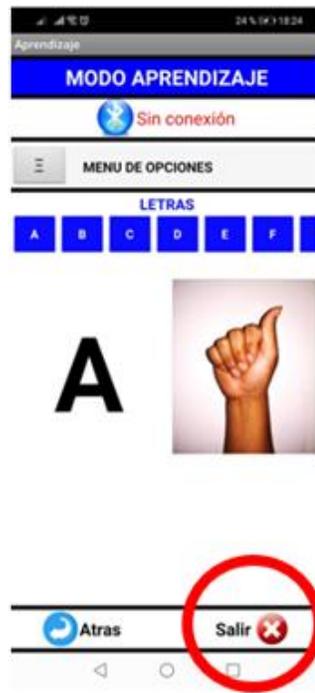
Encender el Bluetooth de su dispositivo android portátil para la conexión con el guante electrónico y escoger el dispositivo HC-05 como se muestra en la imagen.



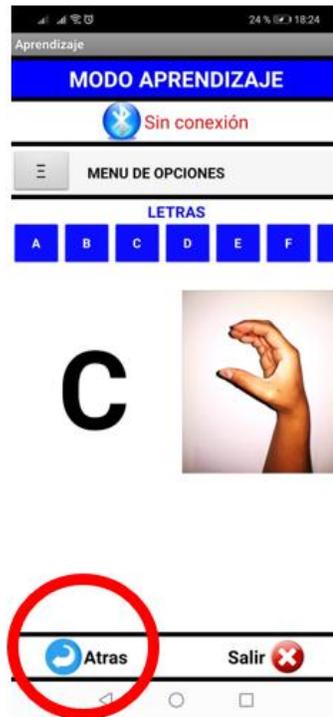
Posterior a esta acción el guante electrónico se encontrara conectado a su dispositivo android portátil y podrá interactuar en las pantallas de traducción y aprendizaje.



Para salir pulse el botón de salir ubicado en la parte inferior derecha de la pantalla



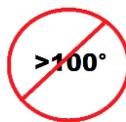
Para regresar pulse el atrás r ubicado en la parte inferior izquierda de la pantalla



**AVISO DE
SEGURIDAD**



No sumergir en agua



No flexionar los dedos del guante más de 100°



La fuente de alimentación del dispositivo únicamente debe de ser de 5 Voltios de corriente continua.



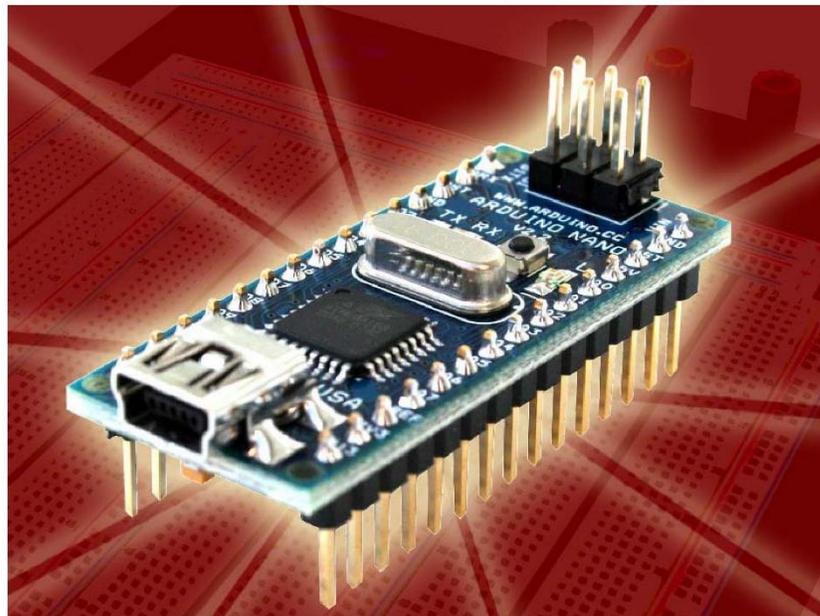
No exponer al fuego.

Anexo 7: Ficha técnica

Anexo 7.1 Arduino Nano

Arduino Nano (V2.3)

User Manual



Released under the Creative Commons Attribution Share-Alike 2.5 License

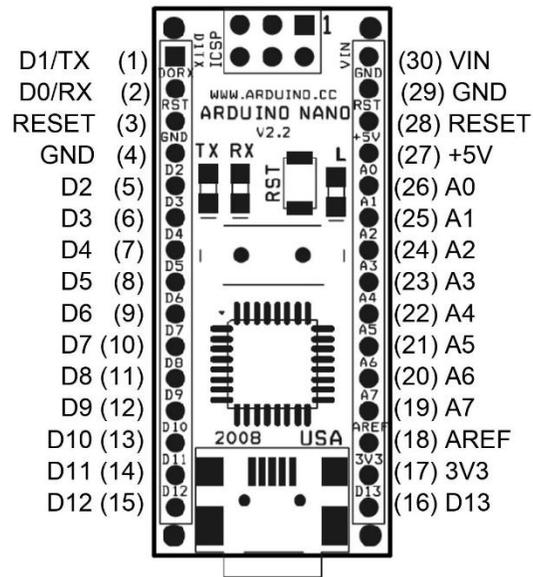
<http://creativecommons.org/licenses/by-sa/2.5/>

More information:

www.arduino.cc

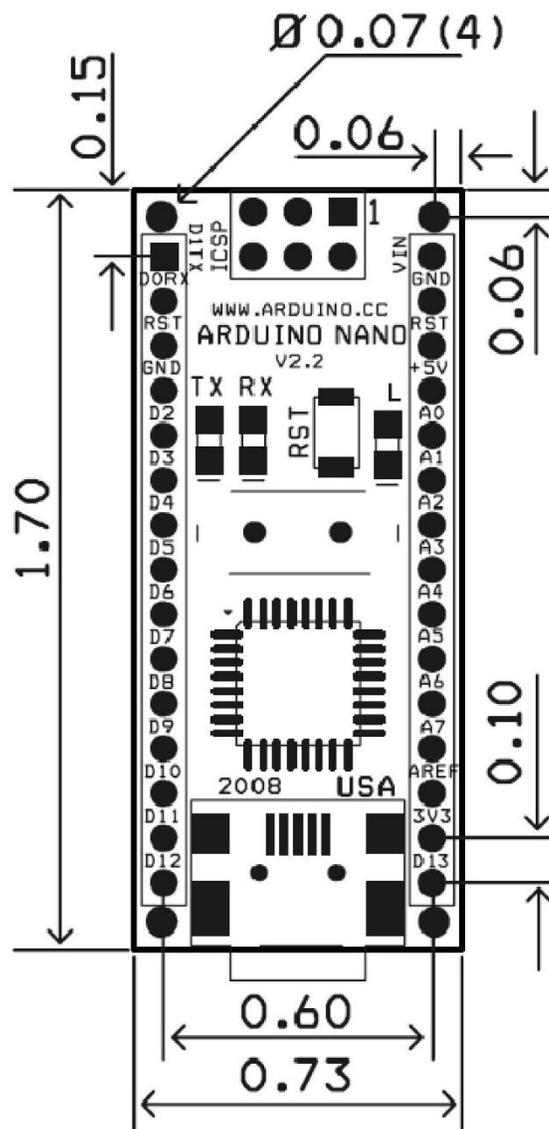
Rev. 2.3

Arduino Nano Pin Layout



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

Arduino Nano Mechanical Drawing



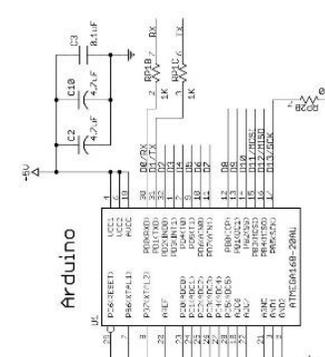
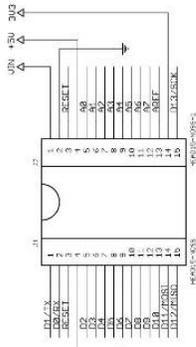
ALL DIMENSIONS ARE IN INCHES

Arduino Nano Bill of Material

Item Number	Qty.	Ref. Dest.	Description	Mfg. P/N	MFG	Vendor P/N	Vendor
1	5	C1,C3,C4,C7,C9	Capacitor, 0.1uF 50V 10% Ceramic X7R 0805	C0805C104K5RACTU	Kemet	80-C0805C104K5R	Mouser
2	3	C2,C8,C10	Capacitor, 4.7uF 10V 10% Tantalum Case A	T491A475K010AT	Kemet	80-T491A475K010	Mouser
3	2	C5,C6	Capacitor, 18pF 50V 5% Ceramic NOP/COG 0805	C0805C180J5GACTU	Kemet	80-C0805C180J5G	Mouser
4	1	D1	Diode, Schottky 0.5A 20V	MBR0520LT1G	ONSemi	863-MBR0520LT1G	Mouser
5	1	J1,J2	Headers, 36PS 1 Row	68000-136HLF	FCI	649-68000-136HLF	Mouser
6	1	J4	Connector, Mini-B Recept Rt. Angle	67503-1020	Molex	538-67503-1020	Mouser
7	1	J5	Headers, 72PS 2 Rows	67996-272HLF	FCI	649-67996-272HLF	Mouser
8	1	LD1	LED, Super Bright RED 100mcd 640nm 120degree 0805	APT2012SRCPRV	Kingbright	604-APT2012SRCPRV	Mouser
9	1	LD2	LED, Super Bright GREEN 50mcd 570nm 110degree 0805	APHCM2012CGCK-F01	Kingbright	604-APHCM2012CGCK	Mouser
10	1	LD3	LED, Super Bright ORANGE 160mcd 601nm 110degree 0805	APHCM2012SECK-F01	Kingbright	04-APHCM2012SECK	Mouser
11	1	LD4	LED, Super Bright BLUE 80mcd 470nm 110degree 0805	LTST-C170TBKT	Lite-On Inc	160-1579-1-ND	Digikey
12	1	R1	Resistor Pack, 1K +/-5% 62.5mW 4RES SMD	YC164-JR-071KL	Yageo	YC164J-1.0KCT-ND	Digikey
13	1	R2	Resistor Pack, 680 +/-5% 62.5mW 4RES SMD	YC164-JR-07680RL	Yageo	YC164J-680CT-ND	Digikey
14	1	SW1	Switch, Momentary Tact SPST 150gf 3.0x2.5mm	B3U-1000P	Omron	SW1020CT-ND	Digikey
15	1	U1	IC, Microcontroller RISC 16KB Flash, 0.5KB EEPROM, 23 I/O Pins	ATmega168-20AU	Atmel	556-ATMEGA168-20AU	Mouser
16	1	U2	IC, USB to SERIAL UART 28 Pins SSOP	FT232RL	FTDI	895-FT232RL	Mouser
17	1	U3	IC, Voltage regulator 5V, 500mA SOT-223	UA78M05CDCYRG3	TI	595-UA78M05CDCYRG3	Mouser
18	1	Y1	Cystal, 16MHz +/-20ppm HC-49/US Low Profile	ABL-16.000MHZ-B2	Abrakon	815-ABL-16-B2	Mouser

Arduino Nano Schematic

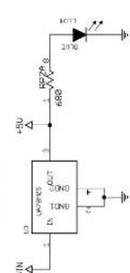
Copyright 2008 under the Creative Commons Attribution Share-Alike 2.5 License
<http://creativecommons.org/licenses/by-sa/2.5/>



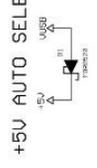
+5V AREF OPTION



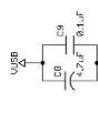
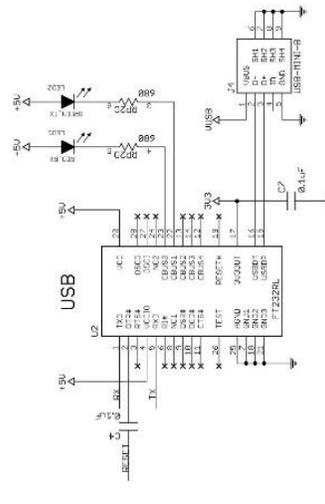
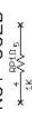
+5V REG



+5V AUTO SELECTOR



NOT USED



v2.3 - Modify FT232RL to Use +5V	
TITLE:	Arduino Nano
Document Number:	
Date:	6/26/2008 9:35:54 PM
REV:	2.3
Sheet:	1/1

Anexo 7.2 HC-05

EL MÓDULO BLUETOOTH HC-05

Arduino y la conexión BlueTooth

Version: 07-09-17

OBJETIVOS

- Presentar el módulo HC-05.
- Describir las ventajas y diferencias con el módulo HC-06.
- Presentar más posibilidades con los comandos AT.
- Presentar un montaje y programa que nos permite entrar directamente en modo de configuración de comandos AT.

MATERIAL REQUERIDO.

	<p>Arduino UNO o equivalente.</p>
	<p>Algunos cables de protoboard, preferiblemente Dupont macho/hembra.</p>
	<p>Un módulo BlueTooth HC-05, tiene 6 pines.</p>

EL MÓDULO BLUETOOTH HC-05

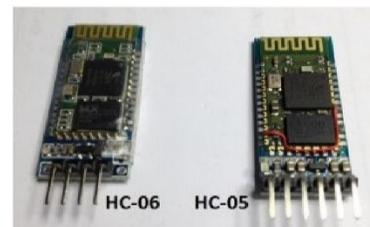
En la sesión anterior presentamos algunos conceptos básicos sobre las redes Bluetooth y las funciones que se suelen realizar con ellos. Presentamos el concepto de Master y Slave y configuramos el modulo HC-06 que solo podía trabajar en el modo Slave.

En esta sesión, vamos a utilizar su hermano mayor el modulo HC-05, que puede configurarse tanto como Master que como Slave, y que además dispone de bastante más parámetros de configuración y capacidades de interrogación.

El aspecto externo es bastante similar y la mejor manera de diferenciarlos es por los pines el soporte, 4 para el HC-06 y 6 pines para el HC-05.

No hay diferencias hardware entre ambos modelos, pero sí que hay diferencias muy importantes en el firmware (La programación interna del módulo).

Además, mientras que el HC-06 entra en modo de programación en cuanto lo enciendes y mientras no haya nadie conectado por Bluetooth, el HC-05 es ligeramente más complicado de colocar en modo comandos y requiere una cierta manera de arrancado, concretamente requiere que el pin KEY, (Que no estaba conectado el caso del HC-06) este en HIGH cuando encendemos el modulo.



He visto bastante información he Internet que recomendaba unos ciertos procesos para arrancar el modulo que siempre me han parecido confusos y tienden a despistar a los que se acercan al tema por primera vez.

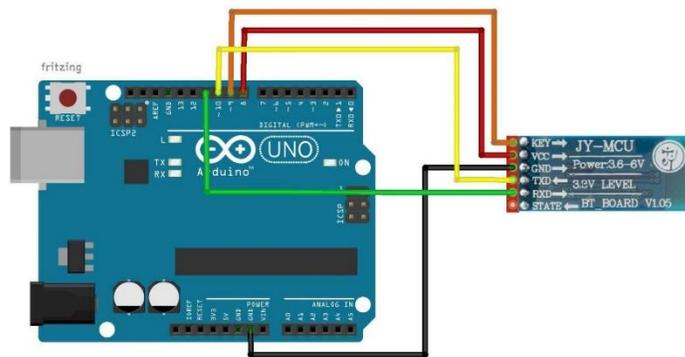
Por eso hemos decidido implementar este procedimiento en un Sketch de Arduino, para que podamos correrlo cuando necesitemos programar el módulo.

El procedimiento normal con estos módulos, suele ser conectarlos, ver la configuración y reprogramarlos con nuestras preferencias. Después mantendrá la programación hasta que decidamos cambiarla.

Pero parece que el proceso da bastante guerra, así que vamos a ver si podemos ayudar.

CONEXIÓN DEL BLUETOOTH HC-05 AL ARDUINO

La conexión es muy sencilla aunque requiere algún cable más que el modulo HC-06, y algún cambio:



En primer lugar, para que el HC-05 entre en modo comandos AT, requiere que cuando se enciende el modulo, el pin KEY este HIGH. Por eso hemos conectado la tensión Vcc del módulo BlueTooth al pin 8 de nuestro Arduino.

El consumo del módulo es mínimo y nuestro Arduino es capaz de alimentarlo sin problemas, por eso el modulo se encenderá cuando pongamos HIGH en el pin 9. Esto nos permitirá poner en HIGH el pin digital 8, al iniciar nuestro programa y después levantar el pin 8, de este modo cuando arranque entrara sin más en el modo de comandos AT.

El resto de los pines se conectan de forma similar a lo que hicimos en la sesión anterior. Txd y Rxd se deben conectar cruzados con los pines de comunicación de Arduino, que usaremos mediante la librería software Serial.

El pin State refleja, supuestamente, la situación en la que se encuentra el modulo y por ahora no vamos a utilizarlo.

PROGRAMA DE CONTROL

NO OLVIDE VER ENSAYOS REALIZADOS POR EL DOCENTE MAS ADELANTE

Así pues, no tenemos nada nuevo en el programa excepto que alimentaremos el modulo desde el pin digital 8, para forzarle a entrar en el modo comandos AT. Nuestro programa quedaría así:

```
#include <SoftwareSerial.h>

SoftwareSerial BT1(10, 11); // RX | TX

void setup()

{ pinMode(8, OUTPUT);      // Al poner en HIGH forzaremos el modo AT

  pinMode(9, OUTPUT);      // cuando se alimente de aqui
```

```

digitalWrite(9, HIGH);

delay (500) ;           // Espera antes de encender el modulo

Serial.begin(9600);

Serial.println("Levantando el modulo HC-06");

digitalWrite (8, HIGH); //Enciende el modulo

Serial.println("Esperando comandos AT:");

BT1.begin(57600);

}

void loop()

{ if (BT1.available())

    Serial.write(BT1.read());

    if (Serial.available())

        BT1.write(Serial.read());

}

```

Con este procedimiento garantizamos que el modulo HC-05 entra solo en modo AT comandos, y que nuestra consola nos va a servir para programarlo, y confío en que nos evite todas las complicaciones.

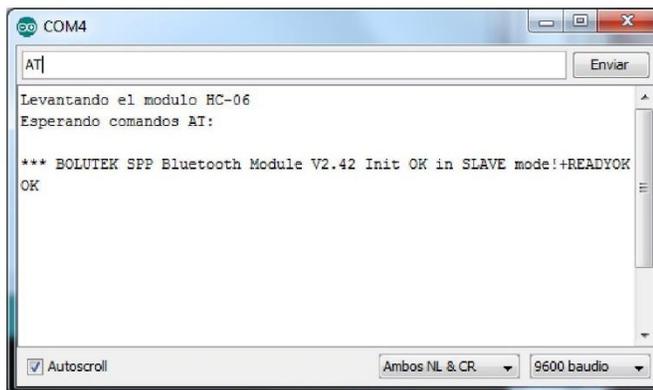
Mi HC-05 me envía a la consola un mensaje de Status en cuanto arranca en el modo AT, si es vuestro caso deberíais ver un mensaje en la consola. Si no es así, o veis signos raros, suele ser indicador de que la velocidad de comunicación entre Arduino y el modulo es incorrecta.

Id probando velocidades hasta que veáis algún mensaje correcto. Enviar un AT siempre debería recibir una respuesta de OK.

PROGRAMANDO EL HC-05 CON COMANDOS AT

(NO DEJAR DE VER ENSAYOS DEL DOCENTE, LOS MODULOS PUEDEN DIFERIR)

Confío en que si hemos seguido los pasos anteriores cuando abras la consola deberías recibir un mensaje del módulo directamente, y si no, basta con que escribáis AT Intro, recibiréis un OK como muestra la captura de pantalla:



El mensaje que obtengáis no tiene por qué ser el mismo e incluso puede que no tengáis ningún mensaje en absoluto, pero es importante que os aseguréis de que tenéis seleccionado Ambos NL&CR para terminar las líneas.

El modulo HC-6, finaliza los comandos por tiempo y por eso en la sesión anterior usamos GetLine para conseguir líneas completas. Pero el HC-05, no hace esto sino que espera que la línea acabe educadamente en \n y entonces ejecuta el comando.

- *Aparentemente el número de modificaciones de software con la que se venden estos módulos es bastante elevada y por ello no es fácil garantizar nada con ellos.*
- *Aparentemente todos los módulos que he probado aceptan ordenes similares, pero no todos los módulos aceptan todas las ordenes, (Supongo que dependiendo del Firmware).*

Cuando menos si escribís en mayúsculas AT e intro, deberías recibir una respuesta de OK en la consola. Y si es así podemos seguir viendo más comandos.

AT+VERSION, Requiere la versión del Firmware. En la captura de arriba muestra el resultado

AT+NAME, Requiere que nos informe del nombre que tiene asignado el modulo. Debería devolvernos un mensaje del tipo +NAME=HC-05, indicando que se llama HC-05.

- *El modulo HC-06 se podía renombrar exactamente así, pero no podíamos preguntarle qué nombre tenía asignado. Solo cambiarlo.*

AT+NAMEXXXX, programa el nombre que queremos presentar cuando alguien nos localice:

AT+NAMECharly

AT+BAUD, nos permite solicitar la velocidad a la que está programado el modulo para hablar con Arduino, y AT+BAUDX, Fija la velocidad de comunicación entre el modulo y la consola de acuerdo a la siguiente tabla:

1 configura	1200bps
2 configura	2400bps
3 configura	4800bps
4 configura	9600bps (Default)
5 configura	19200bps
6 configura	38400bps
7 configura	57600bps
8 configura	115200bps

Ejemplo: AT+BAUD7 configura la comunicación a 57600 baudios

Si enviamos AT+BAUD, la respuesta es de la misma forma: BAUD=7 para indicar 9600.

AT+PIN, Solicita el PIN actual y en la consola veris: PIN=1234 o similar.

AT+PINXXXX, configura el número de identificación personal, que se requerirá para establecer la vinculación

AT+PIN4516, establece 4516 como PIN.

- Recordad que el PIN es el número de identificación personal, que usaremos al conectamos al módulo, pues exigirá conocer la contraseña.
- El PIN es de 4 dígitos siempre

Básicamente estos son los mismos comandos que aceptaba el modulo HC-06 pero el HC-05 acepta bastante más.

AT+ROLE Nos informa de si está configurado como Maestro 1, o como esclavo 0.

ROLE=0 (Esclavo)

AT+ROLE1 Configura el modulo como Master.

AT+ROLE0 Configura el modulo como Slave.

- *He visto muchos ejemplos por internet que indican que se debe usar un = para asignar valores, por ejemplo AT+ROLE=1, o que para pedir información de la velocidad hay que hacer AT+BAUD?, También he visto que varias páginas aseguran que para cambiar el PIN hay que hacer AT+PSSWD.*
- *Yo no he sido capaz de que nada de eso me funcione así, y sin embargo en mi modulo funciona la sintaxis que os escribo en los ejemplos. Tengo que suponer que hay diferentes Firmwares que modifican la sintaxis porque no puedo creer que haya tantas nomenclaturas erróneas corriendo por la red.*
- *Así que tendréis que probar cual es la que os funciona a vosotros.*

No tendría demasiado sentido revisar aquí todos los posibles comandos AT, y por eso he preferido mostrar media docena de ellos, los más típicos, para que podáis experimentar y configurar el modulo.

Aquí tenéis una lista con algunos comandos a los que más o menos he encontrado sentido (Porque hay montones que no tengo ni idea para que sirvan).

AT COMMAND LISTING	
COMMAND	FUNCTION
AT	Test UART Connection
AT+RESET	Reset Device
AT+VERSION	Query firmware version
AT+ORGL	Restore settings to Factory Defaults
AT+ADDR	Query Device Bluetooth Address
AT+NAME	Query/Set Device Name
AT+RNAME	Query Remote Bluetooth Device's
AT+ROLE	Query/Set Device Role
AT+CLASS	Query/Set Class of Device CoD
AT+IAC	Query/Set Inquire Access Code
AT+INQM	Query/Set Inquire Access Mode
AT+PSWDAT+PIN	Query/Set Pairing Passkey
AT+UART	Query/Set UART parameter
AT+CMODE	Query/Set Connection Mode
AT+BIND	Query/Set Binding Bluetooth Address
AT+POLAR	Query/Set LED Output Polarity
AT+PIO	Set/Reset a User I/O pin

Cuando hayamos programado el modulo como deseemos, podemos retirar la conexión del pin rotulado como KEY y el HC-05 está listo para trabajar normalmente sin aceptar ya comandos AT. En el caso ensayado dice EN

ATENCIÓN:

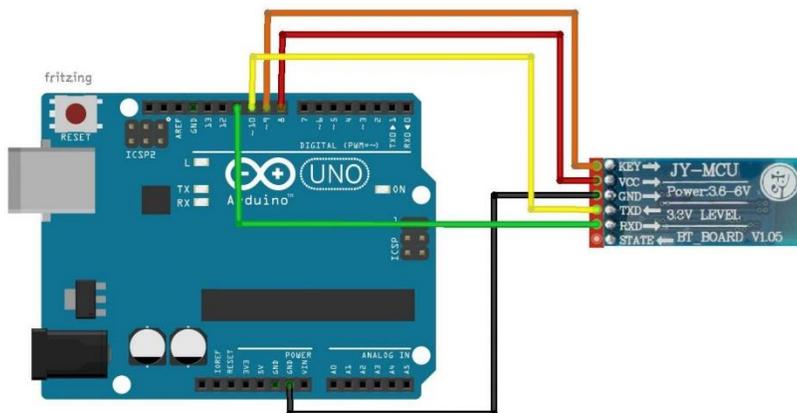
Es importante comprender, que en cuanto conectemos algún dispositivo a nuestro modulo Bluetooth HC-05, la luz roja dejará de parpadear y automáticamente saldremos del modo de comandos AT para entrar en el de transporte de información, es decir, cualquier cosa que le enviemos, incluyendo comandos AT, se consideraran texto plano que se reenviará al otro extremos de la conexión Bluetooth.

Recordar que solo podemos enviar comandos AT mientras no estemos conectados

RESUMEN DE LA SESIÓN

- Hemos presentado el modulo BlueTooth HC-05.
- Hemos visto las diferencias con el módulo HC-06 y que a diferencia de este, se puede configurar como maestro o como esclavo.
- También hemos visto que acepta bastantes mas ordenes en comandos AT.
- Hemos presentado un montaje que fuerza a entrar en el modo AT al módulo cuando arranca. Una vez programado podemos soltar el pin KEY y el modulo estará listo para el servicio.

ENSAYOS REALIZADOS POR EL DOCENTE Circuito empleado



Este es el programa que funciona

```
#include <SoftwareSerial.h>

SoftwareSerial BT1(10, 11); // RX | TX para no usar los pines especificos RX TX
void setup()
{ pinMode(8, OUTPUT); // Al poner en HIGH forzaremos el modo AT
  pinMode(9, OUTPUT); // cuando se alimente de aqui
  digitalWrite(9, HIGH);
  delay (500); // Espera antes de encender el modulo
  Serial.begin(9600);
  Serial.println("Levantando el modulo HC-05");
  digitalWrite (8, HIGH); //Enciende el modulo
  Serial.println("Esperando comandos AT:");
  // BT1.begin(57600); Original en programa
```


AT+ROLE Nos informa de si está configurado como Maestro 1, o como esclavo 0.	+ROLE:0 OK
AT+PSWD? Check PIN code AT+PSWD=<Param> Cambiar el código PIN Ej: AT+PSWD=1963	+PSWD:1234 OK OK
AT+PSWD?	+PSWD:1963 OK

TERMINADA LA CONFIGURACION, AHORA VEREMOS COMO CONECTARNOS CON ESTOS MODULOS

PERO ANTES TENGAMOS EN CUENTA LO SIGUIENTE

El módulo que se ensayo admite conectar VCC a 5V (Power 3,6v a 6v), por lo tanto es posible conectar a los 5 v de Arduino (que es lo que se uso) e incluso se probó que funciona con la salida de 3,3 del Arduino.

Con esto tenemos resuelto el “encender” el módulo, ahora lo que nos hace falta es poder comunicarnos con él.

Para ello conectaremos dos pines digitales del Arduino (podemos utilizar el RX y TX que tenemos en los pines 0 y 1 respectivamente) con los pines de TX y RX del

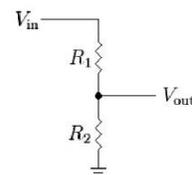


módulo, pero aquí nos encontramos con el problema de que nuestro Arduino está trabando a 5V y el módulo bluetooth en cuanto a señales trabaja a 3.3V (LEVEL 3,3v), y, aunque hay gente que los ha conectado directamente y asegura que le funciona perfectamente yo prefiero no arriesgarme y hacer una adaptación de niveles, dado que es muy simple.

En primer lugar tenemos la salida del módulo (TX, pin 1) en ella tendremos un nivel de 0 para el nivel lógico 0 y aproximadamente 3.3V para el nivel 1. Como según las características del ATmega328P el V_{ih} es $0.6V_{cc}$, y nuestro V_{cc} es de 5V, tendremos que V_{ih} es aproximadamente de 3V y por tanto **es perfectamente válido conectar el TX del HC-05 con el RX del Arduino.**

Algo totalmente diferente es la salida del Arduino hacia la entrada (RX, pin 2) del módulo. En el datasheet del HC-05 no he visto que indique que las entradas sean tolerantes a 5V, y por otro lado **en el datasheet del ATmega328P indica que V_{oh} es, como mínimo, de 4.2V**, por lo que, por precaución, **deberíamos adaptar los niveles**. Hay muchos métodos para hacer la adaptación de niveles, aunque, para mí, en este caso la solución más fácil es realizar un divisor resistivo, de forma que solo le entreguemos una parte de la tensión que esté dentro de sus márgenes de tolerancia al módulo bluetooth.

Para poder utilizar un divisor resistivo con señales lo primero que tenemos que tener en cuenta es a quien vamos a conectar la salida del divisor, ya que si tiene una impedancia en el mismo orden de magnitud (o inferior) que la resistencia de salida el paralelo de ambas hará que los cálculos que hayamos realizado estén mal. Para este paso nos iremos al datasheet del módulo y vemos que la entrada RX es de tipo CMOS con, esto significa que tendrá una



impedancia de entrada alta.

Llegados a este punto está claro queremos realizar un divisor de tensión que tenga una entrada de 5V y una salida de 3.3V, con una impedancia de salida que no sea muy alta. La disposición sera así:

La fórmula a utilizar es: $V_o = V_i * R_2 / (R_1 + R_2)$

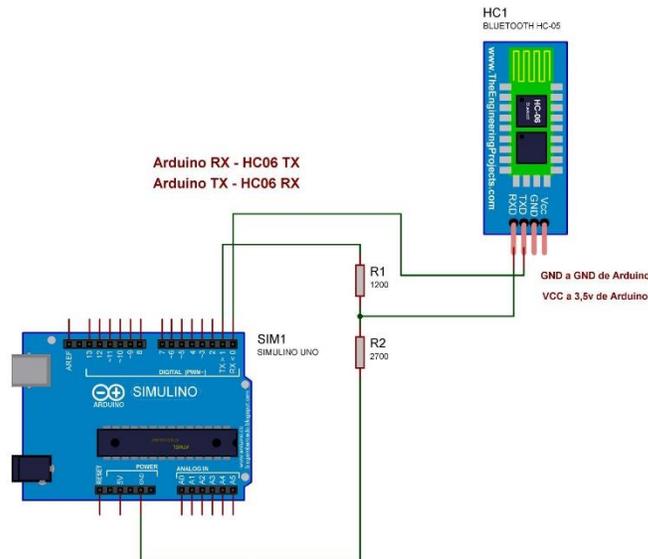
Cuando no es muy relevante la resistencia que hay que poner en una conexión uno de los valores típicos es 10KΩ, por lo que empezaremos por ese valor como resistencia de salida del divisor y vemos si nos sale algo razonable.

Si tomamos como valores $R_2 = 10K$, $V_i = 5V$, $V_o = 3.3V$ y despejamos R_1 , tendremos:

$$R_1 = (V_i / V_o - 1) * R_2$$

Donde R_1 nos da un valor 5151.515Ω de forma que llevándolo a valores estándar podremos poner una resistencia de 4K7Ω o de 5K6Ω, en el primer caso tendremos una salida de 3.4V y en el segundo de 3.2V. Y eso es todo lo necesario para conectar el módulo, otro día veremos como comunicarnos con él.

Nota: Se ha probado con $R_2 = 2,7K$ y $R_1 = 1,2K$ dándonos una tensión aproximada de de 3,3 v



La conexión sería la siguiente entre el módulo Bluetooth HC 05 (valido también para el HC-06) y la placa Arduino:

Arduino 5v – HC05 VCC
 Arduino GND – HC05 GND
 Arduino TX – HC05 RX
 Arduino RX – HC05 TX

Primer Ejemplo: El programa que se uso y funcionó es el siguiente: (PrimerBlue)

```
void setup()
{
  Serial.begin(9600); //Iniciar el serial
  pinMode(5, OUTPUT); //Establecer el pin 13 como salida
}

void loop()
{
```

```

if(Serial.available()>=1)
{
  char entrada = Serial.read(); //Leer un caracter

  if(entrada == 'h' or entrada == 'H') //Si es 'H', encender el LED
  {
    digitalWrite(5, HIGH);
    Serial.println("LED encendido");
  }

  else if(entrada == 'l' or entrada == 'L') //Si es 'L', apagar el LED
  {
    digitalWrite(5, LOW);
    Serial.println("LED apagado");
  }

  else if(entrada == 'i' or entrada == 'I') //Si es 'I', mostrar un mensaje de ayuda
  {
    Serial.println("Comandos:n (i) - abrir esta listan (h)- encender ledn (l) - apagar ledn");
  }
}
}
}
}

```

Recuerde desconectar momentáneamente el modulo Bluetooth cuando va a cargar el programa en el Arduino

Creación de la APP en Android para conectarse a Arduino por medio del modulo Bluetooth PrimerBlue



Vamos a programar una aplicación para Android con MIT App Inventor 2, un aplicativo web que nos permitirá construir aplicaciones rápidamente sin tener que programar (concepto relativo, ya que hay que saber algo de programación). Controlaremos un LED, construyéndonos una interface a medida.

El programa MIT App Inventor 2 es muy simple. Hay dos ventanas: una en la que se crea el diseño de la aplicación (arrastrando elementos de un menú tales como botones, barras y colocándolos en la pantalla del móvil) y la segunda para programar mediante bloques.

Asumiendo que ya tiene experiencia previa con el MIT App Inventor 2 (*de lo contrario hay muchos tutoriales en Internet*), los objetos que necesita serian los siguientes:

- List Picker
- Button1
- Button2
- Button3
- Bluetooth Client (de Palette->Connectivity)

La pantalla de la APP quedaría como se ve en la figura.



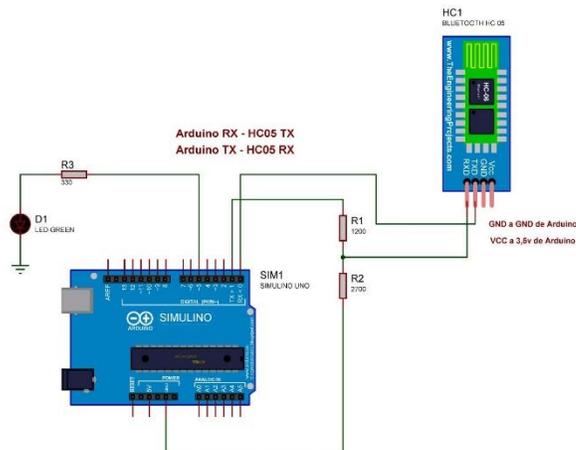
La estructura de bloques asi:



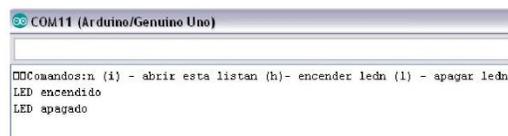
Luego de cargar la aplicación en nuestro telefono o tablet, alimentamos nuestro circuito que se detalla en la figura siguiente:

El LED del modulo Bluetooth parpadeara rápidamente, indicando que espera conexión entrante.

Encendemos el Bluetooth del movil, ejecutamos la APP, buscamos con el primer botón las conexiones cercanas, seleccionamos la de nuestro modulo (Es importante recordar el nombre que le dimos en la configuración del mismo). El LED parpadeara pero a mas baja velocidad.



Cada vez que presionemos algunos de los botones, podemos observar en el Monitor Serial.



Segundo ejemplo **SegunBlue**

Se desea realizar un nuevo programa para Arduino y para el Movil que permita encender el LED pero mediante un código específico. Por lo cual reformularemos ambos.

SegunBlue

```

void setup()
{
  Serial.begin(9600); //Iniciar el serial
  pinMode(5, OUTPUT); //Establecer el pin 5 como salida
}

void loop()
{
  if(Serial.available()>=1)
  {

    //Delay para favorecer la lectura de caracteres

    delay(22);

    //Se crea una variable que servirá como buffer
    String bufferString = "";

    /*
    * Se le indica a Arduino que mientras haya datos
    * disponibles para ser leídos en el puerto serie
    * se mantenga concatenando los caracteres en la
    * variable bufferString
    */

    while (Serial.available()>0) {
      bufferString += (char)Serial.read();
    }

    long entrada = bufferString.toInt(); //Se carga lo leído en la variable entrada

    Serial.println(entrada); //Muestro lo que entre

    if(entrada == 12345) //Si es 'codigo', encender el LED
    {
      digitalWrite(5, HIGH);
      Serial.println("LED encendido");
      Serial.println(entrada);
    }

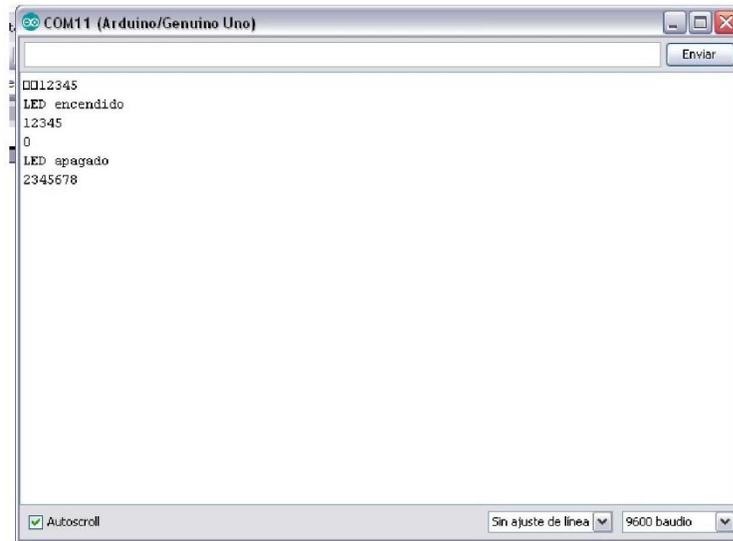
    else if(entrada == 0) //Si es '0', apagar el LED
    {
      digitalWrite(5, LOW);
      Serial.println("LED apagado");
    }

  }
}

```

Recuerde desconectar momentáneamente el modulo Bluetooth cuando va a cargar el programa en el Arduino

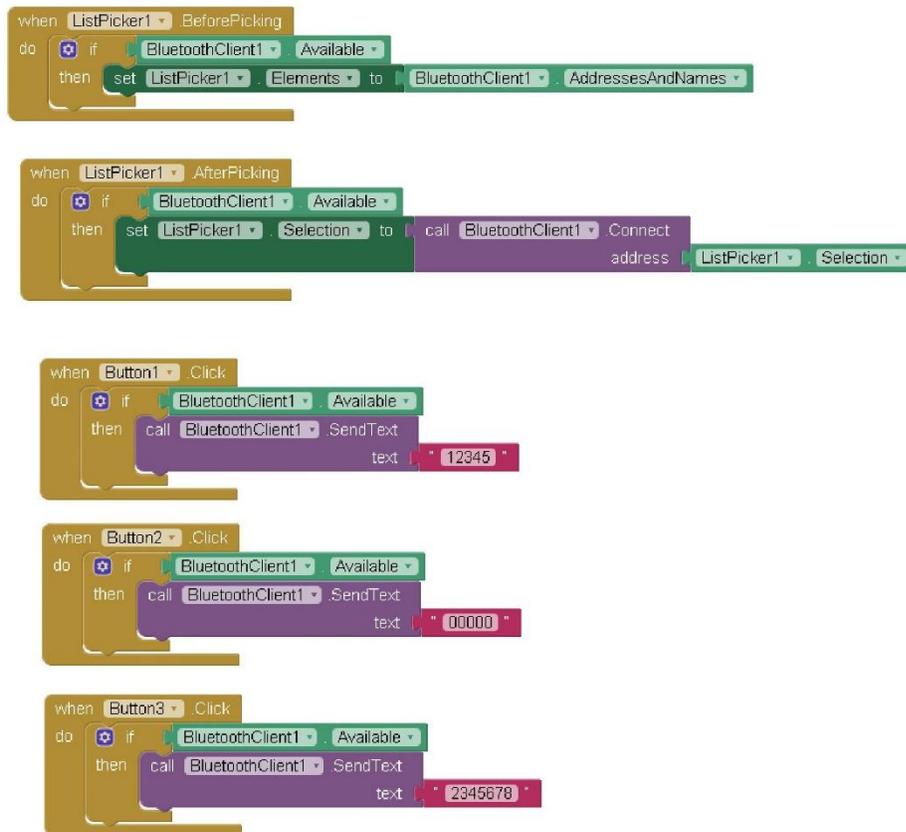
Este programa permite hacer el seguimiento por el Monitor Serie de Arduino:



Creación de la APP en Android para conectarse a Arduino por medio del modulo Bluetooth [SegunBlue](#)

Envía un código que reconoce Arduino y enciende LED. Se podría utilizar para activar otros mecanismos e inclusive ampliar a varios. Un botón envía código (12345) que Arduino reconoce y enciende LED, otro código (00000) apaga LED y dado que el programa utilizado en Arduino contempla el seguimiento por Monitor Seria se plantea el envío de otro código que no ejecuta ninguna tarea (2345678).





Tercer ejemplo **TercerBlue**

Se desea realizar una nueva versión del programa para Arduino y para el Movil que incluya las funciones del **SegunBlue**, pero que tambien incluya ordenes verbales, que se podran dar luego de agitar el movil (uso del acelerometro).

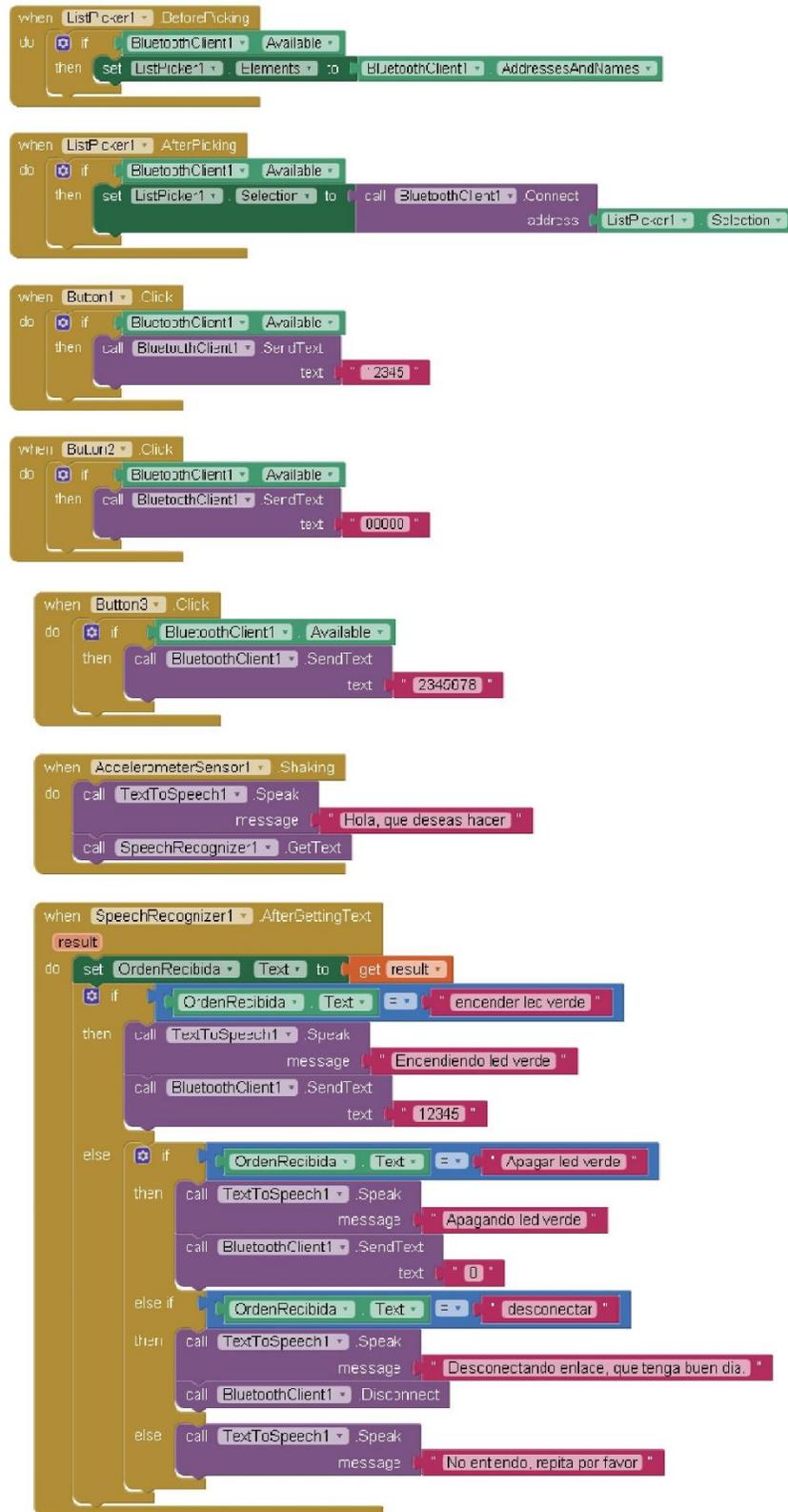
Las órdenes son:

- **Apagar LED verde**
- **Encender LED verde**
- **Desconectar (cierra el enlace Bluetooth)**

Tener en cuenta que para el reconocimiento de voz y el uso del sintetizador de voz es necesario una conexión a Internet, de lo contrario no se podran usar dichas funciones.

El programa para Arduino sigue siendo el anterior, solo cambia la APP del movil.





Cambiar el uso de los Pines RX y TX por otros Pines

A través de la librería SoftwareSerial se pueden cambiar los pines RX y TX a otros pines para establecer la comunicación con el Modulo Bluetooth ya que utilizaremos el puerto serie RX TX para comunicación con la PC, ya sea para cargar o depurar el programa o comunicarse via PC con Arduino. Esta librería viene con el IDE de Arduino, solo debemo incluirla en nuestro programa.

```
//A través de la librería SoftwareSerial
//se pueden cambiar los pines RX y TX a otros pines
//para establecer la comunicación con el Modulo Bluetooth
//ya que utilizaremos el puerto serie RX TX para comunicacion
//con la PC, ya sea para cargar o depurar el programa o comunicarse
//via PC con Arduino

#include <SoftwareSerial.h>
SoftwareSerial BT(2,3); // Cambio RX | TX para conectar Modulo Bluetooth
                        //en pines 2 y 3 (yo elijo llamarlo BT)

long bps=9600; // Como comodidad para definir la velocidad de la comunicacion

void setup()
{
  Serial.begin(bps); //Iniciar el serial para Monitor serial
  BT.begin(bps); //Iniciar serial para Modulo BT

  pinMode(5, OUTPUT); //Establecer el pin 5 como salida
}

void loop()
{
  if(BT.available()>=1) // Me refiero a la comunicacion con Modulo BT
  {

    //Delay para favorecer la lectura de caracteres

    delay(22);

    //Se crea una variable que servirá como buffer
    String bufferString = "";

    /*
     * Se le indica a Arduino que mientras haya datos
     * disponibles para ser leídos en el puerto serie
     * se mantenga concatenando los caracteres en la
     * variable bufferString
     */

    while (BT.available()>0) {
      bufferString += (char)BT.read();
    }

    long entrada = bufferString.toInt(); //Se carga lo leído en la variable entrada
```

```

Serial.println(entrada);//Muestro contenido de variable entrada

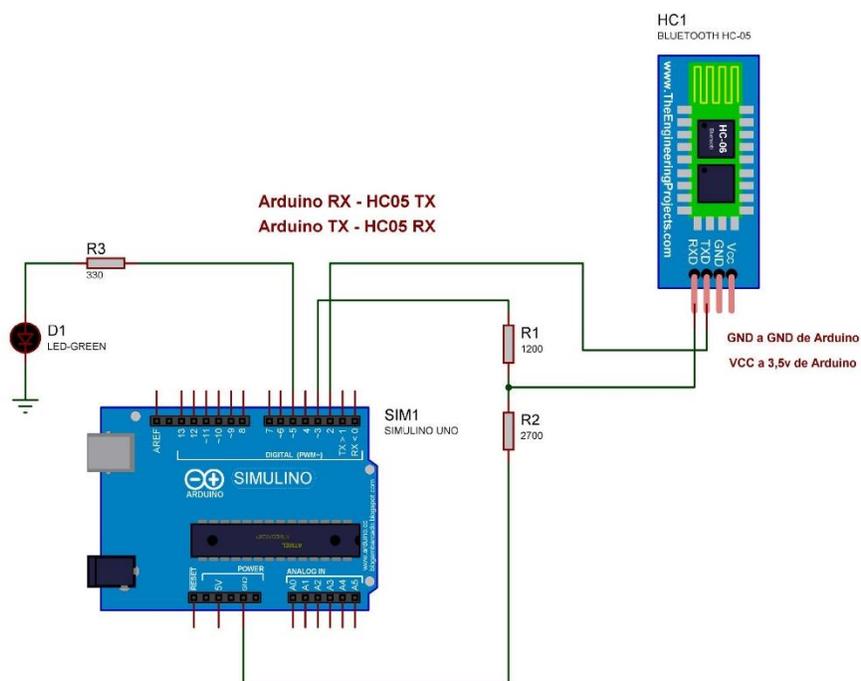
if(entrada == 12345) //Si es 'codigo', encender el LED
{
  digitalWrite(5, HIGH);
  Serial.println("LED encendido");
  Serial.println(entrada);
}

else if(entrada == 0) //Si es 'L', apagar el LED
{
  digitalWrite(5, LOW);
  Serial.println("LED apagado");
}

}
}

```

Circuito con los cambios de pines.



Anexo 7.3 Sensor Flex



FLEX SENSOR FS

Special Edition Length

Features

- Angle Displacement Measurement
- Bends and Flexes physically with motion device
- Possible Uses
 - Robotics
 - Gaming (Virtual Motion)
 - Medical Devices
 - Computer Peripherals
 - Musical Instruments
 - Physical Therapy
- Simple Construction
- Low Profile

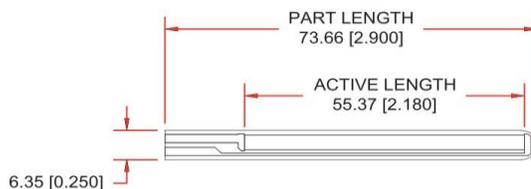
Mechanical Specifications

- Life Cycle: >1 million
- Height: ≤0.43mm (0.017")
- Temperature Range: -35°C to +80°C

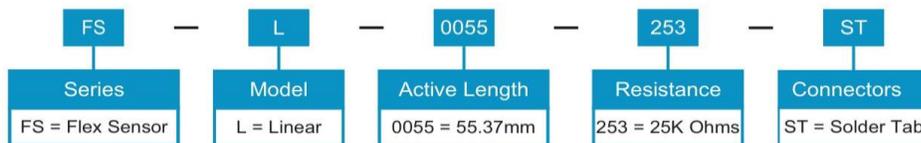
Electrical Specifications

- Flat Resistance: 25K Ohms
- Resistance Tolerance: ±30%
- Bend Resistance Range: 45K to 125K Ohms (depending on bend radius)
- Power Rating : 0.50 Watts continuous. 1 Watt Peak

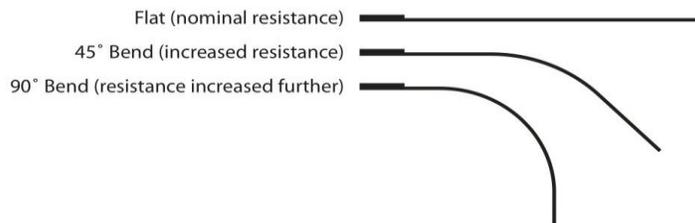
Dimensional Diagram - Stock Flex Sensor



How to Order - Stock Flex Sensor

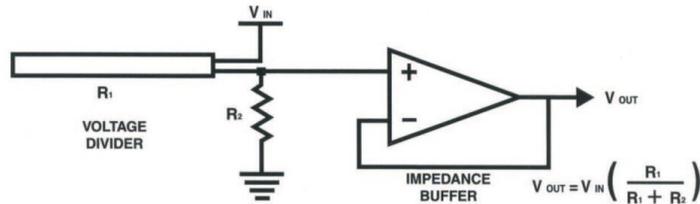


How It Works



Schematics

BASIC FLEX SENSOR CIRCUIT:

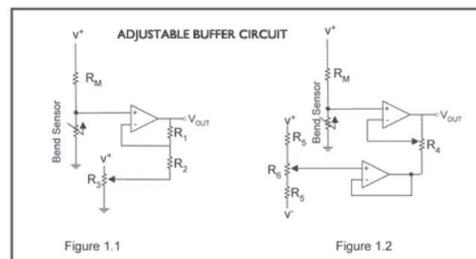


Following are notes from the ITP Flex Sensor Workshop

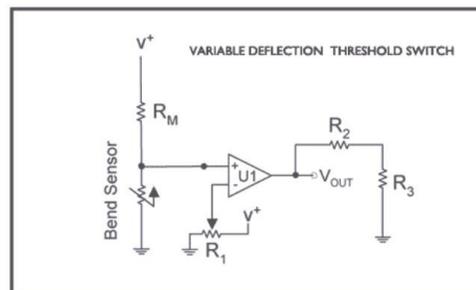
"The impedance buffer in the [Basic Flex Sensor Circuit] (above) is a single sided operational amplifier, used with these sensors because the low bias current of the op amp reduces error due to source impedance of the flex sensor as voltage divider. Suggested op amps are the LM358 or LM324."

"You can also test your flex sensor using the simplest circuit, and skip the op amp."

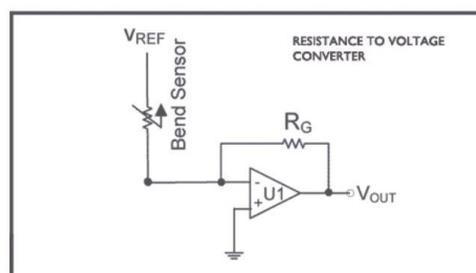
"Adjustable Buffer - a potentiometer can be added to the circuit to adjust the sensitivity range."



"Variable Deflection Threshold Switch - an op amp is used and outputs either high or low depending on the voltage of the inverting input. In this way you can use the flex sensor as a switch without going through a microcontroller."



"Resistance to Voltage Converter - use the sensor as the input of a resistance to voltage converter using a dual sided supply op-amp. A negative reference voltage will give a positive output. Should be used in situations when you want output at a low degree of bending."



Anexo 8: Cronograma de actividades

Nombre de tarea	Duración	Comienzo	Fin	Predecesoras	Semestre 1, 2019												Semestre 2, 2019												Semestre 1, 2020											
					A	M	J	J	A	S	O	N	D	E	F	M	A	M	J	J	A	S	O	N	D	E	F	M	A	M	J	J	A							
▲ PIC	282 días	lun 15/4/19	mié 19/8/20																																					
Aprobación del proyecto	37 días	lun 15/4/19	mar 18/6/19																																					
▲ Establecer el requerimiento del Lenguaje de Señas	180 días	mar 18/6/19	mar 28/4/20	2																																				
Consultar acerca	45 días	mar 18/6/19	mié 4/9/19																																					
Realizar entrevi	45 días	mié 4/9/19	jue 21/11/19	4																																				
Definir los carac	45 días	jue 21/11/19	vie 7/2/20	5																																				
Relacionar los c	45 días	lun 10/2/20	mar 28/4/20	6																																				
▲ Definir los elementos para el diseño del guante	15 días	mar 28/4/20	vie 22/5/20	3																																				
Investigar sobre	3 días	mar 28/4/20	vie 1/5/20																																					
Establecer la lis	3 días	lun 4/5/20	jue 7/5/20	9																																				
Adquirir los con	3 días	jue 7/5/20	mié 13/5/20	10																																				
Diseño del guar	3 días	mié 13/5/20	mar 19/5/20	11																																				
Realizar diagan	3 días	mar 19/5/20	vie 22/5/20	12																																				
▲ Programación de tarjeta Arduino	14 días	lun 25/5/20	mié 17/6/20	8																																				
Realizar prueba	3 días	lun 25/5/20	jue 28/5/20																																					
Configurar la pl	3 días	jue 28/5/20	mié 3/6/20	15																																				
Asignar signific	4 días	mié 3/6/20	mié 10/6/20	16																																				
Pruebas de veri	4 días	mié 10/6/20	mié 17/6/20	17																																				
▲ Creación de la aplicación para dispositivo Android	10 días	mié 17/6/20	vie 3/7/20	14																																				
Diseño de interfaz de usuario	3 días	mié 17/6/20	mar 23/6/20																																					
Elaboración de diagrama de bloques	3 días	mar 23/6/20	vie 26/6/20	20																																				
Pruebas de verificación y funcionamiento	4 días	lun 29/6/20	vie 3/7/20	21																																				
▲ Elaborar trabajo escrito	26 días	lun 6/7/20	mié 19/8/20	19																																				
Elaboración Cap	4 días	lun 6/7/20	vie 10/7/20																																					
Elaborar Capítul	7 días	lun 13/7/20	jue 23/7/20	24																																				
Elaborar Capítul	5 días	jue 23/7/20	vie 31/7/20	25																																				
Elaboración de	4 días	lun 3/8/20	vie 7/8/20	26																																				
Correcciones	6 días	lun 10/8/20	mié 19/8/20	27																																				