

UNIVERSIDAD MAYOR DE SAN ANDRÉS
FACULTAD DE TECNOLOGÍA
CARRERA DE ELECTRÓNICA Y TELECOMUNICACIONES



Dispositivo electrónico de comunicación para estudiantes con
discapacidad intelectual en la ciudad de La Paz.

Trabajo de Aplicación – Examen de Grado presentado para obtener el Grado de
Licenciatura

POR: Gabriel Angel Quispe Yujra

TUTOR: Lic. Luis Richard Márquez Gonzales

LA PAZ – BOLIVIA

Mayo, 2023

RESUMEN

El presente trabajo alude a las problemáticas que presentan las poblaciones con Necesidades Educativas Específicas, en cuanto al desarrollo de las competencias comunicativas. Se considera relevante el contexto en el que está inmerso el sujeto, el cual carece de reconocimiento psicosocial, y por lo tanto requiere de un medio que le facilite el uso de sus propios códigos de comunicación, para poder ser comprendido por el resto de la sociedad.

Para abordar esta problemática se suscita la posibilidad de trabajar la comunicación aumentativa y alternativa, que se interesa en aportar soluciones a las personas que no pueden comunicarse debido a una deficiencia de tipo corporal y motor.

Por ello, es relevante el uso de la tecnología, donde esta ayudara a superar las alteraciones marcadas en la comunicación de las personas que presentan discapacidad mental, mediante un intercomunicador que mejorara esas deficiencias presentadas en el proceso comunicativo, para iniciar procesos de enseñanza-aprendizaje, promoviendo la calidad de vida relacionada con la integración e inclusión del sujeto en la sociedad.

Dentro del contexto áulico existe la falta de espacios designados al desarrollo de actividades de la vida diaria en las aulas, los espacios son reducidos para el desarrollo curricular existe una carencia de personal de apoyo en el aula, también existe un evidente retraso en las áreas de desarrollo de cada niño por la falta de asistencia de servicios especializados, carencia de materiales didácticos y mobiliario específico para las necesidades de cada estudiante y la dificultad en la comunicación por parte de los estudiantes con su contexto.

ÍNDICE

CAPITULO I.....	6
1.1. Planteamiento del problema.....	7
1.2. Justificación del trabajo.....	7
1.2.1. Justificación Tecnológica.....	7
1.2.2. Justificación Económica.....	8
1.2.3. Justificación Social.....	8
1.3. Objetivos.....	9
1.3.1. Objetivo General.....	9
1.3.2. Objetivos Específicos.....	9
CAPITULO II.....	10
2.1. Fundamentación Teórica.....	11
2.1.1. Tipos de discapacidad.....	11
2.1.1.1. Discapacidad Física.....	11
2.1.1.2. Discapacidad Sensorial.....	11
2.1.1.3. Discapacidad Psíquica.....	12
2.1.1.4. Discapacidad Intelectual o mental.....	12
2.1.2. Grados de afectación según el DSM-IV.....	12
2.1.2.1. Discapacidad cognitiva leve CI 50-55 a 70.....	12
2.1.2.2. Discapacidad cognitiva moderada CI 35-40 a 50-55.....	13
2.1.2.3. Discapacidad cognitiva grave CI 20-25 a 35-40.....	13
2.1.2.4. Discapacidad cognitiva profunda CI 20-25.....	13
2.1.2.5. Discapacidad cognitiva de gravedad no especificada.....	14
2.1.3. Comunicación con personas con discapacidad.....	14
2.1.3.1. Sistemas alternativos y aumentativos de comunicación (SAACs)...	14
2.1.3.2. Tipos de Tecnología de ayuda para la comunicación Aumentativa y Alternativa.....	16
2.1.4. Arduino.....	27
2.1.4.1. Breve Historia.....	27
2.1.4.2. Concepto.....	28
2.1.4.3. Ventajas.....	28
2.1.4.4. Hardware.....	29

2.1.4.5.	Arquitectura Básica	31
2.1.4.6.	Arduino Uno	37
CAPITULO III	44
3.1.	Desarrollo del Trabajo	45
3.1.1.	Diagrama en bloques	45
3.1.2.	Teclado Matricial	46
3.1.2.1.	Funcionamiento	47
3.1.2.2.	Conexiones	49
3.1.3.	Diseño y elaboración del esquema de audio	50
3.1.3.1.	Creación y conversión de Archivos	51
3.1.3.2.	Preparación de la tarjeta Micro SD	55
3.1.4.	Descripción del circuito	58
3.1.5.	Elaboración del programa	66
3.1.5.1.	Codificación	66
CAPITULO IV	72
4.1.	Conclusiones	73
5.1.	BIBLIOGRAFIA Y ANEXOS	74
5.1.1.	Bibliografía.....	74
5.1.2.	ANEXOS	75

INDICE DE FIGURAS

Figura 1	Sistema de Signos Manuales	17
Figura 2	Sistema de Signos Tangibles	18
Figura 3	Sistema Pictográfico	18
Figura 4	Sistema Bliss.....	19
Figura 5	Tríptico.....	20
Figura 6	Cuaderno Personalizado de Comunicación	20
Figura 7	Dispositivo Big – Mack	21
Figura 8	Libro Electrónico de Comunicación	21
Figura 9	Dispositivo Chat Box.....	23
Figura 10	Dispositivo LightWraiter	23
Figura 11	Dispositivo Side – Kick	24
Figura 12	Dispositivo Alphatalker	24
Figura 13	Dispositivo Delta Talker	25
Figura 14	Dispositivo Springboard	25
Figura 15	Dispositivo Pathfinder	26

Figura 16 Diagrama en bloques de una placa Arduino	30
Figura 17 Diagrama de Arquitectura AVR	31
Figura 18 Mapa de memoria AVR	35
Figura 19 Modelos Placas Arduino	37
Figura 20 Arduino Uno	37
Figura 21 Pines Placa Arduino	38
Figura 22 Microcontrolador Atmega 328.....	42
Figura 23 Entorno del programa Arduino	43
Figura 24 Diagrama en bloques del prototipo.....	45
Figura 25 Pulsadores en forma Matricial en el simulador Proteus	47
Figura 26 Construcción del Teclado Matricial.....	47
Figura 27 Ubicación librería Keypad.....	49
Figura 28 Conexión Teclado Matricial a Arduino.....	49
Figura 29 Pruebas del Circuito Intercomunicador	50
Figura 30 Ingreso al programa Loquendo.....	51
Figura 31 Entorno del programa Loquendo	51
Figura 32 Grabación de palabras en Loquendo.....	52
Figura 33 Página de inicio Online – Convert.....	53
Figura 34 Entorno de Online - Convert para convertir archivos .wav	53
Figura 35 Formato de conversión Online – Convert.....	54
Figura 36 Conversión del archivo .wav en Online – Convert.....	54
Figura 37 Descarga del archivo .wav en Online – Convert	55
Figura 38 Formateo de la Tarjeta MicroSD	56
Figura 39 Formato completo de la Tarjeta MicroSD	56
Figura 40 Archivos .wav con el formato requerido para Arduino.....	57
Figura 41 Estructura del Módulo lector de tarjetas MicroSD	58
Figura 42 Configuración Maestro-Eslavo entre Arduino y la tarjeta MicroSD	59
Figura 43 Conexión Modulo lector MicroSD con Arduino	60
Figura 44 Distribución de pines Tarjeta MicroSD	61
Figura 45 Modulo lector de Tarjetas MicroSD	61
Figura 46 Circuito Amplificador de Audio	62
Figura 47 Circuito Amplificador.....	63
Figura 48 Esquema de Pines del LM386	63
Figura 49 Conexión a la placa Arduino Uno del Teclado Matricial.....	65
Figura 50 Diagrama completo Circuito electrónico de Comunicación.....	66
Figura 51 Estructura del código Arduino	66
Figura 52 Dispositivo Electrónico de Comunicación Vista Frontal.....	75
Figura 53 Dispositivo Electrónico de Comunicación Vista Lateral Izquierda.....	75
Figura 54 Dispositivo Electrónico de Comunicación Vista Trasera	76
Figura 55 Dispositivo Electrónico de Comunicación Vista Lateral Derecha	76
Figura 56 Dispositivo Electrónico de Comunicación Vista Panorámica	77

CAPITULO I
ANTECEDENTES DEL PROYECTO

1.1. PLANTEAMIENTO DEL PROBLEMA

La fundación “Arco Iris” se encuentra en la Zona Villa Copacabana Av. Burgaleta #265, esta fundación es una organización sin fines de lucro cuyo principal desafío es responder a las necesidades de los niños, niñas y jóvenes huérfanos con alguna discapacidad especial dotándoles de techo, alimentación, salud, vestimenta, educación, capacitación técnica y cubriendo la necesidad de cariño y apoyo integral para su superación personal.

Dentro del contexto áulico existe la falta de espacios designados al desarrollo de actividades de la vida diaria en las aulas, los espacios son reducidos para el desarrollo curricular, existe una carencia de personal de apoyo en el aula, también existe un evidente retraso en las áreas de desarrollo de cada niño por la falta de asistencia de servicios especializados, carencia de materiales didácticos y mobiliario específico para las necesidades de cada estudiante, y la dificultad en la comunicación por parte de los estudiantes con su contexto.

Este proyecto se enfoca en esta problemática: la dificultad que presenta el estudiante de comunicarse con su contexto. Esto provoca que el estudiante no pueda interactuar con su comunidad a causa de barreras comunicativas, presentando limitación en la expresión de: ideas, pensamientos y sentimientos, y que repercute en el proceso enseñanza-aprendizaje. Por ello se propone el uso de la tecnología para romper estas barreras y limitaciones pudiendo así mejorar la calidad de vida de esta población, diseñando un intercomunicador electrónico que ayude a estos, a afrontar día a día la vida cotidiana donde mejore su aprendizaje y a la vez puedan comunicarse.

1.2. JUSTIFICACIÓN DEL TRABAJO

1.2.1. Justificación Tecnológica

El uso y desarrollo de la tecnología siempre se desarrolló en brindar mayor comodidad y facilidades al usuario, si bien en otros países se desarrollan día a día nuevas tecnologías que ayuden a personas con discapacidades para mejorar su nivel de vida, en nuestro país vemos que pocas veces se concentra en beneficiar a esta población, y que a la vez deberían ser el mayor centro de nuestra atención ya que son las más incomprendidas por la falta de información de cómo tratarlos y comunicarnos con ellos, los limitamos y restringimos de

tantos avances que podrían tener si podríamos aumentar e incentivar su enseñanza-aprendizaje por medio de apoyos tecnológicos. Este es el objetivo de nuestro proyecto ya que al implementar este tipo de tecnología podremos conseguir este avance comunicativo e incentivar a las personas con discapacidad intelectual, incluyéndolos en nuestra sociedad pero incluirlos realmente para ya no verlos como aquellos o ellos sino en verdad verlos como un nosotros frenando así la creciente discriminación que está latente dentro de nuestra sociedad, brindándoles mayores oportunidades usando de nuestro conocimiento tecnológico para eliminar estas barreras.

1.2.2. Justificación Económica

Al implementarse el proyecto dentro del aula, desplazara el uso de materiales didácticos menos sofisticados, esto generara la disminución de presupuesto en la adquisición de estos, si bien solo se construirá un dispositivo electrónico de comunicación alternativa-aumentativa para estudiantes con discapacidad intelectual, a largo plazo viendo los resultados que genere este, abre la posibilidad de realizar varios dispositivos que lleguen a cubrir los requerimientos de la Fundación “Arco Iris”, con la producción que se hagan de estos dispositivos se puede llegar a abaratar costos y con la adecuada financiación este proyecto puede ser implementado en muchas más instituciones, donde ya no se necesitaría importar este tipo de tecnología que claramente tendría un costo mayor, sino se usaría nuestros propios recursos y tecnologías para adaptar estas a los requerimientos de este tipo de población.

1.2.3. Justificación Social

Este proyecto tendrá beneficios futuros, ya que mientras se implemente a los estudiantes a una corta edad mayor beneficio obtendrán, pues en el transcurso de los años su capacidad intelectual será estimulada y su aprendizaje también será mayor, abriéndoles varias puertas en un futuro, ya que la captación de la realidad y su comunicación con la sociedad tendrá un aumento significativo, beneficiándolos a ellos y también a su familiares ya que permitirá que estos puedan crear un vínculo más estrecho con su hijos, entender sus necesidades y sus requerimientos y así poder unir a más y más familias, de esta manera fortalecer la comunicación e interrelación del estudiante con su contexto inmediato.

1.3. OBJETIVOS

1.3.1. Objetivo General

Diseñar un prototipo de un dispositivo electrónico de comunicación alternativa-aumentativa para estudiantes con discapacidad intelectual leve a grave en la fundación “Arco Iris” de la ciudad de La Paz.

1.3.2. Objetivos Específicos

Para alcanzar el objetivo general del presente proyecto se plantearon los siguientes objetivos específicos:

- Seleccionar el sistema de símbolos de comunicación que se adapte a las necesidades de los estudiantes con discapacidad intelectual de la fundación “Arco Iris”.
- Diseñar un teclado electrónico de acuerdo con los símbolos establecidos para la efectiva comunicación de los estudiantes dentro de la fundación “Arco iris”.
- Elaborar el código fuente que se grabará en el microcontrolador destinado a la funcionalidad del dispositivo electrónico de comunicación aumentativa - alternativa.
- Construir el prototipo del dispositivo electrónico de comunicación alternativa - aumentativa que cubra las necesidades de los estudiantes con discapacidad intelectual en la fundación “Arcoíris”.

CAPITULO II
MARCO TEORICO

2.1. FUNDAMENTACIÓN TEÓRICA

2.1.1. Tipos de discapacidad

Las Personas con Discapacidad, son aquellas personas con deficiencias físicas, mentales, intelectuales y/o sensoriales a largo plazo o permanentes, que al interactuar con diversas barreras puedan impedir su participación plena y efectiva en la sociedad, en igualdad de condiciones con las demás. (Legislativo, 2018)

Existen los siguientes tipos de discapacidad: física, psíquica, sensorial e intelectual o mental.

Cada uno de los tipos puede manifestarse en distintos grados y una persona puede tener varios tipos de discapacidades simultáneamente, con lo que existe un amplio abanico de personas con discapacidad.

2.1.1.1. Discapacidad Física

La diversidad funcional motora se puede definir como la disminución o ausencia de las funciones motoras o físicas (ausencia de una mano, pierna, pie, entre otros), disminuyendo su desenvolvimiento normal diario. (Wikipedia, Tipos de Discapacidad, 2022)

Las Personas con Discapacidad Física – Motora, son las personas con deficiencias anatómicas y neuro-músculo funcionales causantes de limitaciones en el movimiento. (Legislativo, 2018)

2.1.1.2. Discapacidad Sensorial

La discapacidad sensorial corresponde a las personas con deficiencias visuales, a los sordos y a quienes presentan problemas en la comunicación y el lenguaje. Existen distintos parámetros para distinguir los distintos grados de discapacidad sensorial. (Wikipedia, Tipos de discapacidad, 2022)

Las personas con Discapacidad Visual son las personas con deficiencias anatómicas y/o funcionales, causantes de ceguera y baja visión. (Legislativo, 2018)

Las personas con Discapacidad Auditiva son las Personas con pérdida y/o limitación auditiva en menor o mayor grado. A través del sentido de la visión, estructura su experiencia e integración con el medio. Se enfrenta cotidianamente con barreras de comunicación que impiden en cierta medida su acceso y participación en la sociedad en igualdad de condiciones que sus pares oyentes. (Legislativo, 2018)

2.1.1.3. Discapacidad Psíquica

Se considera que una persona tiene discapacidad psíquica cuando presenta "trastornos en el comportamiento adaptativo, previsiblemente permanentes". (Wikipedia, Tipos de discapacidad, 2022)

Las personas con Discapacidad Mental o Psíquica, son personas que debido a causas biológicas, psicodinámicas o ambientales son afectadas por alteraciones de los procesos cognitivos, lógicos, volitivos, afectivos o psicosociales que se traducen en trastornos del razonamiento, de la personalidad, del comportamiento, del juicio y comprensión de la realidad, que los dificultan adaptarse a ella y a sus particulares condiciones de vida, además de impedirles el desarrollo armónico de relaciones familiares, laborales y sociales, sin tener conciencia de la enfermedad psíquica. (Legislativo, 2018)

2.1.1.4. Discapacidad Intelectual o mental

El funcionamiento intelectual hace referencia a un nivel de inteligencia inferior a la media. La concepción de inteligencia que se maneja a este respecto es la propuesta por Gardner, quien habla de la existencia más que de una capacidad general, de una estructura múltiple con sistemas cerebrales semiautónomos, pero que, a su vez, pueden interactuar entre sí.

En cuanto al uso de un coeficiente de inteligencia, de uso común, para que se pueda hablar de discapacidad intelectual debe estar por debajo de 70 y producir problemas adaptativos. (Wikipedia, Tipos de discapacidad, 2022)

2.1.2. Grados de afectación según el DSM-IV

2.1.2.1. Discapacidad cognitiva leve CI 50-55 a 70

Se denomina así a las personas que transitan la "etapa educable" son alrededor del 85% de las personas afectadas por el trastorno. Suelen desarrollar habilidades sociales y de comunicación durante los años preescolares (0-5 años), tienen insuficiencias mínimas en las áreas sensorias motoras y con frecuencia no se diferencian de otros niños sin discapacidad cognitiva hasta edades posteriores. Adquieren habilidades sociales y laborales adecuadas para una autonomía mínima, pero pueden necesitar supervisión, orientación y asistencia. (Wikipedia, Discapacidad Intelectual, 2023)

2.1.2.2. Discapacidad cognitiva moderada CI 35-40 a 50-55

La discapacidad cognitiva moderada equivale aproximadamente a la categoría pedagógica de adiestrable. Este grupo constituye alrededor del 10% de toda la población con discapacidad cognitiva. Adquieren habilidades de comunicación durante los primeros años de la niñez. Adquieren una formación laboral y, con supervisión moderada, pueden adquirir destrezas para su propio cuidado personal. También pueden beneficiarse de adiestramiento en habilidades sociales y laborales, pero es improbable que progresen más allá de un segundo nivel en materias escolares. En su mayoría son capaces de realizar trabajos no cualificados o semicualificados, siempre con supervisión, en talleres protegidos o en el mercado general del trabajo. (Wikipedia, Discapacidad Intelectual, 2023)

2.1.2.3. Discapacidad cognitiva grave CI 20-25 a 35-40

Incluye el 3-4 % de los individuos con discapacidad cognitiva. Durante los primeros años de la niñez la adquisición de un lenguaje comunicativo es escasa o nula. Durante la edad escolar pueden aprender a hablar y pueden ser adiestrados en habilidades elementales de cuidado personal. Se benefician sólo limitadamente de la enseñanza de materias pre académicas como la familiaridad con el alfabeto y el cálculo simple. Los adultos pueden ser capaces de realizar tareas simples estrechamente supervisadas en instituciones. (Wikipedia, Discapacidad Intelectual, 2023)

2.1.2.4. Discapacidad cognitiva profunda CI 20-25

Incluye aproximadamente en el 1% - 2% de las personas con discapacidad cognitiva. La mayoría de los individuos con este diagnóstico presentan una enfermedad neurológica identificada que explica su discapacidad cognitiva. Durante los primeros años desarrollan considerables alteraciones del funcionamiento sensorio motor. Puede predecirse un desarrollo óptimo en un ambiente altamente estructurado con ayudas y supervisión constantes, así como con una relación individualizada con el educador. El desarrollo motor y las habilidades para la comunicación y el cuidado personal pueden mejorar si se le somete a un adiestramiento adecuado. Algunos de ellos llegan a realizar tareas simples en instituciones protegidas y estrechamente supervisados. (Wikipedia, Discapacidad Intelectual, 2023)

2.1.2.5. Discapacidad cognitiva de gravedad no especificada

Se utiliza cuando existe claridad sobre la discapacidad, pero no es posible verificar mediante los test.

Dependiendo del nivel de gravedad y de la discapacidad cognitiva, el sujeto se puede educar y capacitar para que aprenda a vivir en la sociedad; puede dominar ciertas habilidades de lectura global (señalización de tránsito, por ejemplo: "STOP"), puede trasladarse a lugares desconocidos o familiares, puede aprender un oficio y trabajar en él, siempre y cuando la sociedad le dé la oportunidad de hacerlo. (Wikipedia, Discapacidad Intelectual, 2023)

2.1.3. Comunicación con personas con discapacidad

2.1.3.1. Sistemas alternativos y aumentativos de comunicación (SAACs)

2.1.3.1.1. Definición

Los Sistemas Aumentativos y Alternativos de Comunicación (SAAC) son un conjunto de recursos, sistemas o estrategias dirigidos a facilitar la comprensión y la expresión del lenguaje de personas que tienen dificultades en la adquisición del habla y/o en la escritura, y su finalidad es la de aumentar el habla o sustituirla. (Abadín, 2012)

Los SAAC son también ayudas y medios que se utilizan para favorecer a las personas que presentan dificultades en el ámbito de la comunicación, de forma que puedan expresar sus deseos, intercambiar conocimientos, opiniones e, incluso, expresar su propia personalidad de manera mucho más eficiente e inteligible para los demás, enriqueciendo así su campo de experiencia. (Abadín, 2012)

Por lo tanto, el objetivo primordial y global de la comunicación aumentativa y alternativa es instaurar y ampliar los canales de comunicación social de las personas con alteración o imposibilidad del habla, apoyando así a una mejor calidad de vida. (Abadín, 2012)

Si se consigue aumentar las posibilidades de autonomía de la persona no oral y proporcionar más fluidez a sus actos comunicativos, con la utilización de nuevas estrategias, se habrá dado un paso muy importante en la mejora de su autoestima. Para conseguir estos objetivos, el momento de la introducción estratégica de comunicación aumentativa, debe ser cuanto antes. Desde el momento en que se detectan dificultades

para el desarrollo del lenguaje oral, se debe acudir al especialista para recibir orientaciones y diseñar un plan de actuación con la finalidad de aumentar la comunicación. (Abadín, 2012)

Hoy en día, no se puede sostener la hipótesis de quienes afirman que, usando los sistemas y estrategias de comunicación aumentativa, se impide el desarrollo en el lenguaje oral o del habla, ya que el trabajo día a día con estas personas avala lo contrario: cuando la persona tiene cubiertas sus necesidades básicas comunicativas, mejora la calidad y aumenta las emisiones verbales. Además, se desarrollan habilidades comunicativas, se aprende y habitúa a la participación social, se mejora la comprensión del entorno en el que se vive formándose su personalidad de una forma global. (Abadín, 2012)

2.1.3.1.2. Sistemas Alternativos

Van dirigidos hacia aquellas personas que no tienen lenguaje oral y que es imposible que se dé a corto o largo plazo, o que se considera que el esfuerzo necesario para que el lenguaje se dé no es rentable y se necesita encontrar un sistema para que el sujeto se comunique. (Wikipedia, Comunicación con personas con discapacidad, 2022)

2.1.3.1.3. Sistemas Aumentativos

Son aquellos que han sido diseñados para incrementar el habla. No suprime la verbalización ni el lenguaje oral pero no es suficiente para establecer una comunicación satisfactoria, se usa con personas sordas. (Wikipedia, Comunicación con personas con discapacidad, 2022)

2.1.3.1.4. Tipos de SAACs

Existen dos grupos claramente diferenciados: los sistemas con ayuda y los sistemas sin ayuda. Se clasifican dependiendo de si hacen uso de soportes o no, y la elección de un tipo de SAAC u otro se verá reflejado dependiendo de la necesidad de la persona. (Wikipedia, Comunicación con personas con discapacidad, 2022)

2.1.3.1.4.1. SAACs con ayuda

Son mecanismos que hacen uso de soportes externos a la persona para poder facilitarle la comunicación con su entorno. En general, estos sistemas incluyen pictogramas, ortografía o escritura en soportes individualizados. (Wikipedia, Comunicación con personas con discapacidad, 2022)

- El sistema Bliss

- El sistema SPC
- El sistema PEC

2.1.3.1.4.2. SAACs sin ayuda

Son mecanismos mediante los cuales las personas con alguna deficiencia o carencia lingüística pueden mejorar su comunicación sin hacer uso de apoyos externos a la persona. Están estrechamente ligados a la educación y reeducación de las personas con sordera profunda prelocutiva con el objetivo principal de desarrollar la lengua oral y, posteriormente, la lengua escrita. (Wikipedia, Comunicación con personas con discapacidad, 2022)

Se destacan:

- La lengua de señas
- El sistema bimodal
- El alfabeto dactilológico
- La palabra complementada

2.1.3.2. Tipos de Tecnología de ayuda para la comunicación Aumentativa y Alternativa

2.1.3.2.1. Sistemas de signos

2.1.3.2.1.1. Sistemas de signos manuales

También denominado lenguaje de signos consiste en realizar determinados signos con las manos para comunicarse con los demás. Uno de los inconvenientes del lenguaje de signos es que precisa que el interlocutor lo conozca, por lo que los familiares y compañeros deben aprenderlo, para que el lenguaje de los signos tenga la utilidad y funcionalidad real que se pretende. Este sistema alternativo de comunicación utilizado en personas que sufren trastornos que inciden en la expresión oral y/o en la audición (deficientes auditivos) y no presentan deficiencias motoras que les impidan el movimiento voluntario de las manos. (Belloch, 2023)

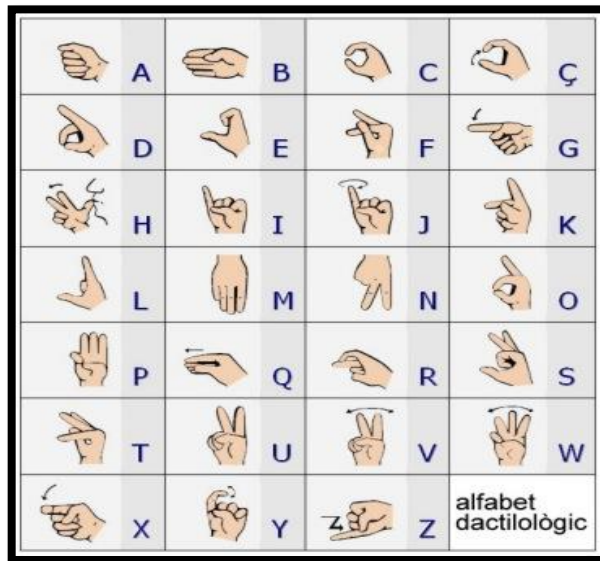


Figura 1 Sistema de Signos Manuales

Fuente: <http://mari6986.blogspot.com/p/dactilologia.html>

2.1.3.2.1.2.Sistemas de signos tangibles

En este modo de comunicación se utilizan diferentes objetos con los que se expresa una información relacionada con el objeto, por ejemplo, se puede mostrar un balón para indicar que se desea jugar a la pelota. Los objetos utilizados suelen ser pequeños, miniaturas o fichas tipo Premack (con el contorno indicando el objeto relacionado con la información). El sistema de signos tangibles es utilizado en personas con retraso mental grave o autistas que se muestran reacios a que alguien les toque las manos. Además, para estas personas el sistema de signos tangibles presenta la ventaja de ser más permanente que los sistemas de signos manuales, ya que estos desaparecen el poco tiempo de realizarse y el sujeto necesita realizar un proceso de la información rápido, lo que en algunas ocasiones resulta complicado para personas con retraso mental grave. Sin embargo, siempre que sea posible, es conveniente combinar ambos sistemas de signos, para que el usuario tenga un mayor vocabulario. (Belloch, 2023)



Figura 2 Sistema de Signos Tangibles

Fuente: <http://deinfantilydiversidad.blogspot.com/2010/02/sistemas-alternativos-y-aumentativos-de.html>

2.1.3.2.1.3. Sistemas de signos gráficos

Existen dos sistemas de signos gráficos que tienen una mayor implantación, y resultados más positivos en su utilización: (Belloch, 2023)

En los sistemas pictográficos, como el sistema SPC (Sistema Pictográfico de Comunicación), la comunicación se realiza a través de un conjunto de gráficos o dibujos lineales e icónicos, que presentan un vocabulario limitado, con el que se comunica la persona.

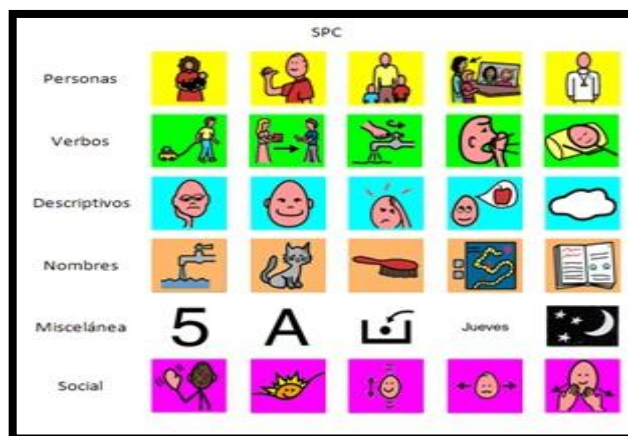


Figura 3 Sistema Pictográfico

Fuente: <https://www.uv.es/bellohc/images/SPCcolores.jpg>

Los sistemas logográficos, más cercanos a la escritura, utilizan dibujos lineales esquemáticos, pictográficos, ideográficos o arbitrarios, resultando más complejo su aprendizaje, por lo que sólo pueden ser utilizados con sujetos que dispongan de un nivel intelectual. Una de las grandes ventajas que presenta si se compara con el sistema pictográfico es que el vocabulario que se puede comunicar no es limitado, ya que la combinación de estos signos puede dar lugar a palabras nuevas, algunas de ellas resultado de la creación del sujeto. El sistema logográfico más utilizado es el sistema Bliss.

Personas (amarillo)	yo ↓	chica ♀	padre ♂	hombro ↓ ↘	niños ↓ 2
Acciones (verde)	ser ⊙	leer ⊙ □	pensar ⌒	aprender ⊙ I	competir →
Sentimientos (azul)	aburrido ∩	triste ♥ ↓	enojado H	feliz ∩ ↑	clase ⊙ →
Cosas (naranja)	plato ∩	pie △ ↓	lago x ~	comedor □ ⊙	pez ∩
Ideas (blanco)	abajo ↓	ocho ⊙	color ⊙	noche ☾	escribe x
Relaciones (rosa/morado)	qué cosa ? □	quién [?]	no -!	parte de ÷	dónde ?

Figura 4 Sistema Bliss

Fuente: <https://www.espaciologopedico.com/revista/articulo/138/sistema-de-simbolos-bliss.html>

2.1.3.2.2. Soportes o ayudas básicas

Son instrumentos sencillos, de fácil fabricación y bajo costo. Genéricamente los conocemos como tableros de comunicación. Los hay de muy diversos tipos:

- **Trípticos:** superficies duras divididas en tres cuerpos, lo que permite su plegado para facilitar el transporte.



Figura 5 Tríptico

Fuente: <https://i.calameoassets.com/201114031055-969625078d25ce1ff0dfa8523ba9dce0/large.jpg>

- **Cuadernos personalizados:** De muy diversa índole. Suelen constar de un conjunto de hojas plastificadas que contienen los símbolos. Para su uso presentan la dificultad de pasar de página de forma autónoma, para lo cual se puede habilitar un código de colores, a modo de solapas, que el usuario debe señalar.



Figura 6 Cuaderno Personalizado de Comunicación

Fuente: https://www.aulawabisabi.com/wp-content/uploads/2019/07/IMG_1640-1.jpg

2.1.3.2.3. Ayudas de Baja Tecnología

-Big-mack: Funciones de Pulsador y grabador, reproductor de mensajes. El usuario activará el mensaje, previamente grabado, mediante un simple pulsado sobre la amplia base.



Figura 7 Dispositivo Big – Mack

Fuente: <https://tecnoaccesible.net/catalogo/big-step-step-levels>

-Libros electrónicos: Se trata de sencillos libretos con una o dos caras en las que se disponen un número variable de casillas que ofrecen la posibilidad de introducir un pictograma por cada una de ellas y grabar un solo mensaje oral.



Figura 8 Libro Electrónico de Comunicación

Fuente: <https://encrypted->

[tbn0.gstatic.com/images?q=tbn:ANd9GcROhu3OiOgIya3o800CAcdIAPw0coThlGI00N9Krao2B9rZ_QXuN5u9b8-FewPMc7F-usM&usqp=CAU](https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcROhu3OiOgIya3o800CAcdIAPw0coThlGI00N9Krao2B9rZ_QXuN5u9b8-FewPMc7F-usM&usqp=CAU)

2.1.3.2.4. Ayudas de Alta Tecnología

2.1.3.2.4.1. Comunicadores Electrónicos

Estos dispositivos electrónicos son creados específicamente para la comunicación y generalmente de fácil portabilidad, con los que el usuario puede con cierta facilidad producir mensajes. Los comunicadores pueden incorporar símbolos SPC, BLISS, Minspeak o los símbolos alfanuméricos. Los comunicadores aportan interactividad y sonido a los mensajes, pudiendo llevar voz digitalizada o sintetizada.

El uso de comunicadores puede resultar más complicado por lo que requieren un proceso de aprendizaje más complejo, comenzando por un número de símbolos pequeño y cercano al entorno natural del sujeto y paulatinamente ir aumentando el número de iconos y complejidad del sistema. Los comunicadores con símbolos alfanuméricos pueden incorporar además de la síntesis de voz, recursos como las abreviaturas, frases predefinidas clasificadas por campos semánticos y el texto predictivo para agilizar el proceso de comunicación.

Algunos ejemplos de comunicadores son:

-Chat box: Comunicador portátil ideado para la iniciación al lenguaje Minspeak, aunque soporta prácticamente cualquiera de los sistemas existentes. Consta de 16 casillas que soportan tanto el pictograma como la grabación de mensajes.

Una pequeña luz situada en la esquina superior izquierda de cada casilla nos indica donde están almacenados los mensajes. Esto hace muy fácil el poder localizar los mensajes y utilizar las secuencias de iconos correctamente. El usuario puede acceder a los mensajes mediante un conmutador externo de cualquier tipo, utilizando un sistema de barrido. Puede trabajar en cuatro niveles diferentes, (hasta 272 mensajes en cada uno de ellos).



Figura 9 Dispositivo Chat Box

Fuente: <https://docplayer.es/docs-images/27/11323406/images/23-0.jpg>

-Lightwriter: Es un dispositivo ligero y portátil de comunicación que utiliza la síntesis de texto a voz como sistema de comunicación. Por tanto, es necesaria la lecto-escritura para poder utilizarlo. Diseñado para personas con capacidades comunicativas ilimitadas, ya que la escritura permite expresar cualquier necesidad, deseo, pensamiento, opinión, etc.



Figura 10 Dispositivo LightWriter

Fuente: <http://www.info-dca.com/uploads/userfiles/images/lightwriter.jpg>

-Side-Kick: Dispone de un teclado de 24 casillas que pueden trabajar en 4 niveles diferentes, lo que permite desde tener almacenada una gran cantidad de vocabulario de diferentes usuarios a la vez (de gran utilidad para los centros), hasta disponer del mismo vocabulario en diferentes idiomas para un mismo usuario. El acceso al SideKick se puede

hacer mediante la activación directa de las teclas, o bien a través de un sistema de barrido controlado por uno o dos pulsadores, incorporando un sistema de predicción de iconos.



Figura 11 Dispositivo Side – Kick

Fuente: <http://3.bp.blogspot.com/-oCoDcYherf0/Vpfr-w5ZLUI/AAAAAAAAACAo/Z1cmT8AUFV4/s1600/scaa3.png>

-Alphatalker: Puede utilizarse además de comunicador, como herramienta de valoración o como herramienta para el desarrollo de diferentes capacidades (asociación física y de conceptos, predicción, lateralidad, etc.). Desarrolla las estrategias de palabras (MAP) de Minspeak a través de 32 casillas. Posibilidad de barrido y voz sintetizada.



Figura 12 Dispositivo Alphatalker

Fuente: <http://1.bp.blogspot.com/-IGKq8gegtLk/TfklltQyA3I/AAAAAAAAAFY/tRHCXqEQtns/s1600/scaa.jpg>

-Delta talker: Comunicador diseñado para el trabajo con Minspeak, con síntesis de voz digitalizada, además de disponer del lenguaje Minspeak (con 128 casillas).

Para los usuarios se convierte en un importante compañero para el aprendizaje en varios campos (lenguaje, categorización, lecto-escritura y autocorrección entre otras) y a los profesionales les ofrece un apoyo más en la enseñanza (evaluación de progresos del alumno, corrección, motivación, reflejos, lateralidad, causa-efecto, etc.).



Figura 13 Dispositivo Delta Talker

Fuente: <http://www.tecnum.net/delta.jpg>

-Springboard (trampolín): Comunicador portátil con sistema de síntesis de voz para el desarrollo del sistema Minspeak y SPC. Posee pantalla táctil, selector por infrarrojos y barrido, dispone de 8, 15 y 32 casillas. Crea y guarda la valoración inicial del sujeto, así como los protocolos de entrenamiento. Posee un juego introductorio de aprendizaje de uso. Ofrece, además selección predictiva de los iconos, importación de iconos y batería con autonomía para 8 horas de funcionamiento.



Figura 14 Dispositivo Springboard

Fuente: https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcRjk6lf-gP0Q8DOtqqJmjp1bf_qS0-bR6QnP1Na9E1R-QbahZzrdLLywOS7LWrTM3DweCo&usqp=CAU

-Pathfinder: Comunicador portátil con sistema de síntesis de voz para el desarrollo del sistema Minispeak, con una tecnología muy avanzada. Puede accederse directamente vía teclado, accionado de barrido, despliegue de pantalla táctil, a través de la cabeza del infrarrojo incorporado y prácticamente con cualquier sistema de acceso, mediante los accesorios de adaptación, ajustándose los parámetros para crear el acceso más eficaz al dispositivo. Constituye también un valioso instrumento para el control del entorno del sujeto, así como posibilidades de ocio y esparcimiento (juegos, música...).



Figura 15 Dispositivo Pathfinder

Fuente: <https://userscontent2.emaze.com/images/e86b492b-6557-436a-9aba-89b06f030961/8baf6bdc-35e8-47ce-80a5-861dc83ec537.jpg>

2.1.3.2.4.2. Tableros de comunicación en soporte informático

El uso de tableros de comunicación en soporte informático requiere en algunos casos el uso de pulsadores o conmutadores y sistemas de "barrido automático" (Belloch, 2023)

Los tableros de comunicación electrónicos pueden incorporar voz, que puede ser:

Voz digitalizada: es la voz previamente grabada para integrarla en el comunicador. Cada una de las casillas del comunicador, que generalmente son una palabra o gráfico, al ser pulsada activa el correspondiente archivo de sonido, emitiéndose la palabra deseada. La voz digitalizada es de gran calidad, pero en contraposición, presenta el inconveniente de presentar un vocabulario limitado, esto es, solo lo previamente grabado.

Voz sintetizada: es aquella que crea el ordenador a partir de algoritmos de programación. Las casillas de un comunicador con voz sintetizada representan el alfabeto (aunque

algunas pueden presentar palabras o incluso frases), y el usuario, irá pulsando las diferentes casillas hasta formar los mensajes que desee enviar, los cuales se escucharán al ser seleccionados. La ventaja de la voz sintetizada, sobre la digitalizada, es que no presenta ningún límite en el vocabulario y frases que se pueden producir, sin embargo, presenta el inconveniente de tener un sonido de menor calidad, dando la sensación de voz metalizada. El ordenador y las capacidades que presenta para el tratamiento de la información permiten dotar al tablero de comunicación de amplias potencialidades:

- Bases de datos con los sistemas de signos necesarios para la comunicación lo cual amplía las posibilidades comunicativas, además de permitir personalizar los tableros a los usuarios y realizar una gestión de estos.
- Interactividad, al igual que en los comunicadores, el tablero de comunicación puede presentar diferentes niveles que vayan respondiendo en función de las elecciones de signos que realice el usuario.
- El sonido que se emite al seleccionar las casillas y permiten la formación de mensajes, esta potencialidad también la tienen los comunicadores.

Para seleccionar el dispositivo o tablero de comunicación más adecuado para un usuario deberemos tener en cuenta: la portabilidad, la capacidad del sujeto para manipularlo o señalar sus símbolos, la capacidad cognitiva y de discriminación visual del usuario. El uso de los tableros de comunicación requiere un aprendizaje por parte del usuario, que atenderá principalmente al entorno natural y próximo, y tendrá en cuenta las necesidades del usuario, su edad, capacidad cognitiva y comunicativa.

2.1.4. Arduino

2.1.4.1. Breve Historia

Arduino se inició en el año 2006 como un proyecto para estudiantes en el Instituto IVREA (Italia). En ese tiempo, los estudiantes usaban el microcontrolador BASIC Stamp, cuyo coste era de 100 dólares estadounidenses, lo que se consideraba demasiado costoso para ellos. El nombre del proyecto viene del nombre del (Bar del Rey Arduino) donde Massimo Banzi pasaba algunas horas. El rey Arduino fue rey de Italia entre los años 1002 y 1014. En la creación de este proyecto contribuyó el estudiante colombiano Hernando Barragán, quien desarrolló la tarjeta electrónica Wiring, el lenguaje de programación y la plataforma de

desarrollo. Una vez concluida dicha plataforma, los investigadores trabajaron para hacerlo más ligero, más económico y disponible para la comunidad de código abierto (hardware y código abierto). El instituto finalmente cerró sus puertas, así que los investigadores, entre ellos el español David Cuartielles, promovieron la idea. Banzi afirmaría años más tarde, que el proyecto nunca surgió como una idea de negocio, sino como una necesidad de subsistir ante el inminente cierre del Instituto de diseño Interactivo IVREA. Es decir, que, al crear un producto de hardware abierto, éste no podría ser embargado. (Wikipedia, Arduino, 2023)

2.1.4.2. Concepto

Arduino es una plataforma de hardware de código abierto, basada en una sencilla placa de circuito impreso que contiene un microcontrolador de la marca “ATMEL” que cuenta con entradas y salidas, analógicas y digitales, en un entorno de desarrollo que está basado en el lenguaje de programación processing. El dispositivo conecta el mundo físico con el mundo virtual, o el mundo analógico con el digital controlando, sensores, alarmas, sistemas de luces, motores, sistemas comunicaciones y actuadores físicos. (Prezi, 2023)

Arduino también simplifica el proceso de trabajo con microcontroladores, ya que está fabricada de tal manera que viene “preensamblada” y lista con los controladores necesarios para poder operar con ella una vez que la saquemos de su caja, ofreciendo una ventaja muy grande para profesores, estudiantes y aficionados interesados en el desarrollo de tecnologías. Las posibilidades de realizar proyectos basados en esta plataforma tienen como límite la imaginación de quien opera esta herramienta. (ARDUINO, 2023)

2.1.4.3. Ventajas

Accesibilidad. - Las placas Arduino son más accesibles comparadas con otras plataformas de microcontroladores. La versión más cara de un módulo de Arduino puede ser montada a mano, e incluso ya montada cuesta bastante menos.

Multi-Plataforma. -El software de Arduino funciona en los sistemas operativos Windows, Macintosh OSX y Linux. La mayoría de los entornos para microcontroladores están limitados a Windows.

Entorno de programación simple y directo. - El entorno de programación de Arduino es fácil de usar para principiantes y lo suficientemente flexible para los usuarios

avanzados. Pensando en los profesores, Arduino está basado en el entorno de programación de Processing con lo que el usuario aprenderá a programar y se familiarizará con el entorno de desarrollo Arduino.

Software ampliable y de código abierto. - El software Arduino está publicado bajo una licencia libre y preparado para ser ampliado por programadores experimentados. El lenguaje puede ampliarse a través de librerías de C++, y modificarlo a través del lenguaje de programación AVR C en el que está diseñado.

Hardware ampliable y de Código abierto. - Arduino está basado en los microcontroladores ATMEGA 168, ATMEGA 328 y ATMEGA 1280. Los planos de los módulos están publicados bajo licencia Creative Commons, por lo que diseñadores de circuitos con experiencia pueden hacer su propia versión del módulo, ampliándolo u optimizándolo. (ARDUINO, 2023)

2.1.4.4. Hardware

Arduino en su diseño de hardware es una placa electrónica que se puede adquirir ensamblada o construirla directamente porque se encuentran los planos electrónicos y la licencia del producto en el internet.

Las placas han ido evolucionando como su software, al inicio las primeras placas utilizaban un chip FTDI “FT232RL” para comunicarse por puerto USB al computador y un procesador para ser programado, luego se utilizó un microcontrolador especial para cumplir esta función como en el caso de Arduino “uno”, que tenían un micro para ser programado y otro para la comunicación, en la actualidad se usa un único microcontrolador que se compromete en llevar a cabo la comunicación y sobre el que también se descargan las instrucciones a ejecutar.

Arduino está constituido en el hardware por un microcontrolador principal llamado Atmel AVR de 8 bits (que es programable con un lenguaje de alto nivel), presente en la mayoría de los modelos de Arduino, encargado de realizar los procesos lógicos y matemáticos dentro de la placa, además de controlar y gestionar los recursos de cada uno de los componentes externos conectados a la misma. Consta además de una amplia variedad de sensores eléctricos que se pueden acoplar al Arduino, como cámaras VGA, sensores de

sonido, seguidores de línea, botones de control de sensores, e incluso, otras placas de microcontroladores (mejor conocidos como Shields), que pueden adaptarse fácilmente gracias a que Arduino cuenta con entradas de pines analógicos y digitales para integrar estos componentes sin necesidad de alterar el diseño original de esta placa. Estos a su vez son controlados junto con el procesador primario por otros componentes de menor jerarquía, pero de igual importancia y prioridad, como el Atmega 168, Atmega 328, Atmega 1280 y el Atmega 8, que son lo más utilizados debido a sus bajos precios y gran flexibilidad para construir diversidad de diseños. Además, Arduino cuenta con la ventaja de tener entre sus elementos principales puertos seriales de entrada /salida (input/output), lo que le permite conectarse por medio de un cable USB a una computadora para poder trabajar con ella desde nivel software, ya que es dónde se le darán las “ordenes” que ejecutarán cada uno de los componentes conectados a la placa, e incluso, para operar como un dispositivo más (dependiendo de la configuración que hayamos establecido y para que se quiere utilizar). Además, Arduino para operar necesita de una fuente de alimentación externa, ya que, por desgracia no cuenta con una propia, por lo que también se encuentra incorporada una entrada para conectar un cable con entrada similar al USB, donde será conectado a un otro dispositivo que tenga entrada USB, o hasta en el mismo dispositivo.

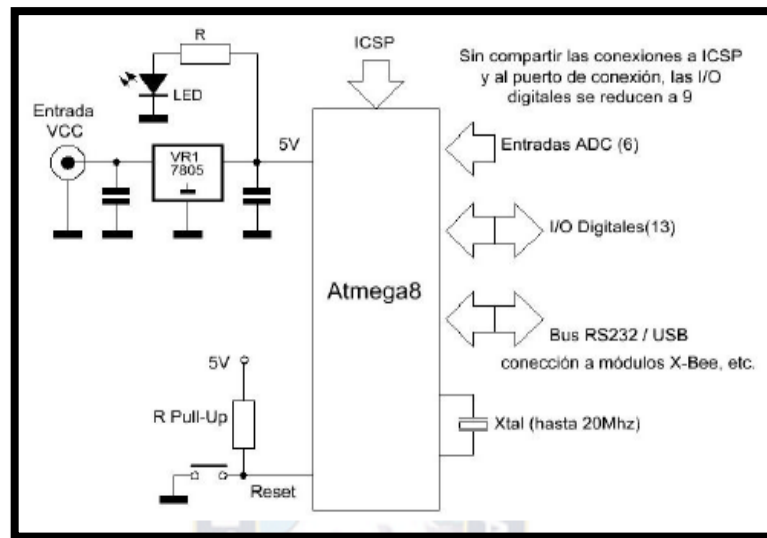


Figura 16 Diagrama en bloques de una placa Arduino

Fuente: www.neoteo.com

2.1.4.5. Arquitectura Básica

2.1.4.5.1 Atmel

Las tarjetas Arduino son placas que contienen un microcontrolador de la marca Atmel denominada AVR® 8-Bit RISC, esta línea de microcontroladores está formada por varios grupos, entre los cuales se encuentra Atmega. La diferencia entre miembros de una misma familia radica en que para cada una de ellas puede variar el número y tipo de periféricos que incluyen la cantidad de memoria de programa y de datos.

Los AVR son una familia de microcontroladores RISC en sus siglas significa (Reduced Instruction Set Computer) su diseño está basado en la arquitectura Harvard que se refiere a que la unidad central de proceso tiene por separado la memoria de programa y la memoria de datos. Los comienzos de la arquitectura AVR fue pensada por los estudiantes Alf Egil Bogen y Vegard Wollan en el Norwegian Institute of Technology, y posteriormente refinada y desarrollada en ATMEL en 1995.

La característica que tienen en común las familias pertenecientes a la línea AVR® 8-Bit RISC es que están basadas en la arquitectura AVR, mostrada en la imagen a continuación:

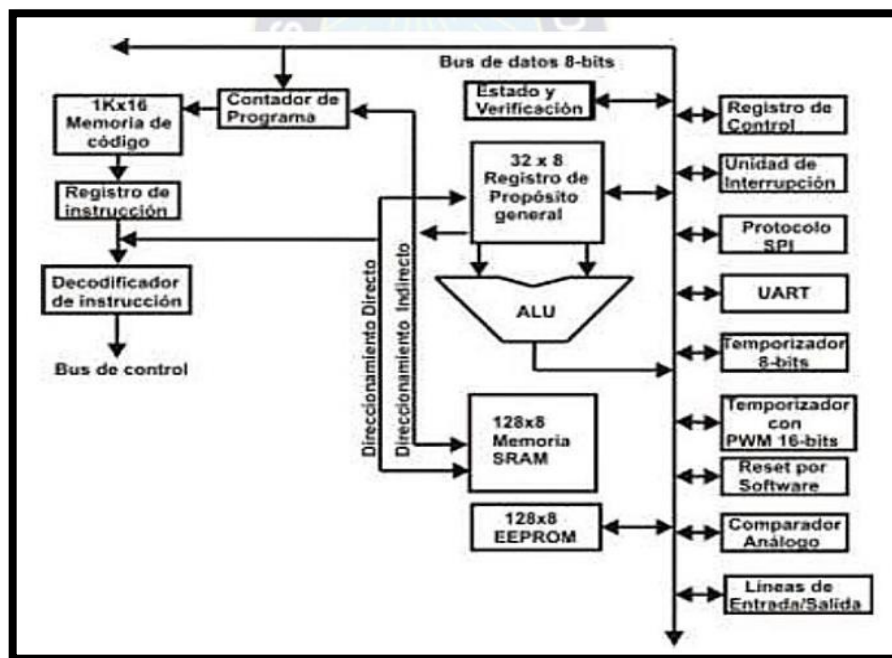


Figura 17 Diagrama de Arquitectura AVR

Fuente: <http://www.sc.ehu.es>

La Arquitectura AVR está formada por:

- Un banco de 32 registros de 8 bits de propósitos generales, sobre los que se realizan la totalidad de las operaciones de cálculo.
- Una ALU (acrónimo de Arithmetic Logic Unit, que significa unidad aritmética lógica) de 8 bits, este circuito digital permite realizar el cálculo aritmético y lógico con los datos almacenados en el banco de registros de propósitos generales.
- Una SRAM (Static Random Access Memory, que significa Memoria estática de acceso aleatorio o RAM estática) de 128x8 bits, para almacenamiento de datos, la tecnología de esta memoria RAM está basada en semiconductores, capaz de mantener los datos, mientras siga alimentada.
- Una EEPROM (Electrically Erasable Programmable Read-Only Memory, que significa ROM programable y borrable eléctricamente) interna de 128x8, para el almacenamiento de datos que deban mantenerse, aun cuando no este con alimentación del microcontrolador.
- Una memoria Flash EPROM de 1Kx16 bits para el almacenamiento de las instrucciones del programa. Esta memoria dispone de un bus dedicado, distinto del de datos, por el que transitan las instrucciones hacia el decodificador de instrucciones, encargado de su ejecución. Además, permite la lectura y escritura de múltiples posiciones de memoria en la misma operación, gracias a esto la tecnología flash permite velocidades de funcionamiento superiores.
- Un puerto serie asíncrono o UART (Universal Asynchronous Receiver-Transmitter, que significa Transmisor-Receptor Asíncrono Universal) es el dispositivo que controla los puertos y dispositivos serie. Se encuentra integrado en la placa base o en la tarjeta adaptadora del dispositivo. Su principal objetivo es convertir los datos recibidos en forma paralela, a forma serial, con el fin de comunicarse con otro sistema externo. También realiza el proceso inverso.
- Un puerto serie síncrono SPI (es un acrónimo para referirse al protocolo de comunicación serial-Serial Peripheral Interface), trabaja en modo full duplex para recibir y transmitir información, permitiendo que dos dispositivos pueden

comunicarse entre sí al mismo tiempo utilizando canales diferentes o líneas diferentes en el mismo cable. Al ser un protocolo síncrono el sistema cuenta con una línea adicional a la de datos encargada de llevar el proceso de sincronismo.

- Un contador /temporizador de 8 bits. Puede funcionar como temporizador o como contador de 0 a 255. En modo temporizador el módulo Timer0 se incrementa con cada ciclo de instrucción. En el modo contador el módulo Timer0 se incrementa en cada flanco (ascendente o descendente).
- Un contador /temporizador de 16 bits con funciones de PWM, generación de salida, control, etc. Donde la señal PWM (pulse-width modulation, que significa modulación por ancho de pulsos), es una técnica que logra producir el efecto de una señal analógica sobre una carga, a partir de la variación de la frecuencia y ciclo de trabajo de una señal digital. El ciclo de trabajo describe la cantidad de tiempo que la señal está en un estado lógico alto, como un porcentaje del tiempo total que éste toma para completar un ciclo completo. La frecuencia determina que tan rápido se completa un ciclo, y por consiguiente que tan rápido se cambia entre los estados lógicos alto y bajo. Al cambiar una señal del estado alto a bajo a una tasa lo suficientemente rápida y con un cierto ciclo de trabajo, la salida parecerá comportarse como una señal analógica constante cuando esta está siendo aplicada a algún dispositivo. En los microcontroladores AVR las señales PWM son generados por los temporizadores.
- Un WATCHDOG o perro guardián que vigila el correcto funcionamiento del programa, ya que es un mecanismo de seguridad que provoca un reset del sistema en caso de que éste se haya bloqueado.
- Un comparador analógico que sirve para: 1) Comparar dos señales (una de las cuales generalmente es una tensión de referencia). 2) Determinar cuál de ellas es mayor o menor.
- 15 líneas de entrada/salida. Una línea indistintamente sirve para enviar la señal o recibirla, dependiendo de lo que haya conectado al principio y al final de esta.
- Una unidad de generación de interrupciones.

El Núcleo del AVR cuenta con 32 registros en forma general, todos estos registros son conectados directamente a la unidad lógico-aritmética (ALU) proporcionando que dos registros puedan ser localizados en una sola instrucción (en un ciclo de reloj), y que el resultado de la operación sea almacenado en uno de estos registros, lo que permite una mejor eficiencia y un mejor rendimiento al comparar con otros microprocesadores convencionales. El núcleo AVR cuenta con memoria de programa tipo flash, lo que implica que no se necesita una memoria externa para contener el código del programa. La memoria de programa puede ser programada de dos modos: en modo serial y en modo paralelo. En la mayoría de los dispositivos que incluyen un núcleo AVR, la memoria de datos está constituida por una memoria EEPROM y una memoria RAM (SRAM).

Los procesadores AVR cuentan con algunos modos de operación, como el modo “Sleep y Powerdown”, para ahorrar energía cuando el procesador no esté trabajando, además los procesadores AVR en su estructura realizan interrupciones internas y externas. Contiene un reset por software (watchdog timer) con oscilador independiente, el cual es utilizado para su recuperación cuando suceda un problema de software, o también puede ser utilizado en algunas otras aplicaciones.

La mayoría de las instrucciones son de 16 bits (2 bytes) de longitud en su código de operación y toman una localidad de la memoria de programa.

Sistema de memoria

El mapa de memoria para la familia de microcontroladores AVR, como la mayoría de los microcontroladores está formado por una memoria de código y una memoria de datos.

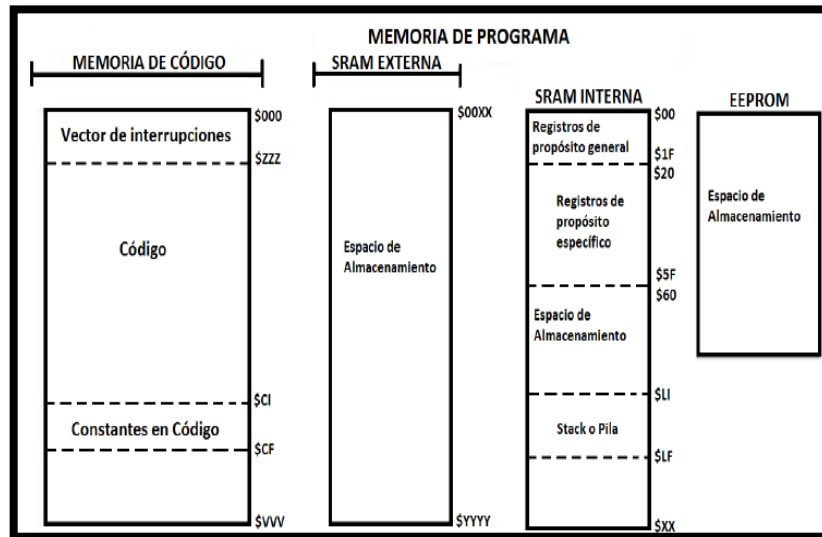


Figura 18 Mapa de memoria AVR

Fuente: El autor

Memoria de Código o Programa

La memoria de código en un microcontrolador AVR dentro la subfamilia MEGA puede contener entre los 4 kbytes y 256 kbytes y está formada en localidades de 16 bits, es direccionada por el contador de programa (PC, Program Counter) y tiene por función principal albergar las instrucciones para realizar una tarea específica. Como funciones alternativas, en el espacio de memoria contiene al vector de interrupciones y en este se pueden declarar constantes. Algunos microcontroladores AVR tienen un espacio que permite el auto-programado.

Memoria de datos

La memoria de datos tiene por función principal contener un espacio de almacenamiento temporal. La memoria de datos agrupa a tres bloques: SRAM interna, SRAM externa y EEPROM. Actualmente en el mercado está disponible el set de instrucciones AVR en diferentes dispositivos que comparten el mismo núcleo, pero tienen distintos Periféricos y cantidades de RAM y ROM. La familia de Tiny AVR ATtiny 11 de microcontrolador contiene 1KB de memoria flash y sin RAM (sólo los 32 registros), y 8 pines, hasta el microcontrolador de la familia Mega AVR ATmega 2560 con 256KB de memoria flash,

8KB de memoria RAM, 4KB de memoria EEPROM, conversor análogo digital de 10 bits y 16 canales, temporizadores, comparador analógico, JTAG, etc.

Los microcontroladores AVR tienen dos etapas (cargar y ejecutar), que les permite ejecutar la mayoría en un ciclo de reloj, lo que los hace relativamente rápidos entre los microcontroladores de 8 bit.

Clasificación de los microcontroladores Atmel

ATXMEGA: Son procesadores muy potentes con 16 a 384 kB de memoria flash programable, encapsulados de 44, 64 y 100 pines, capacidad de DMA (direct memory access, que significa acceso directo a memoria) que permite a cierto tipo de componentes acceder a la memoria del sistema para leer o escribir independientemente de la unidad central de procesamiento (CPU) principal, eventos, criptografía y amplio conjunto de periféricos con DACs (Digital to Analogue Converter), donde este conversor es capaz de transformar la señal digital en analógica.

ATMEGA: Los microcontroladores AVR contienen 4 a 256 kB de memoria flash programable, encapsulados de 28 a 100 pines, conjunto de instrucciones extendido (multiplicación y direccionamiento de programas mayores) y amplio conjunto de periféricos.

ATTINY: son pequeños microcontroladores AVR con 0,5 a 8 kB de memoria flash programable, encapsulados de 6 a 20 pines y un limitado set de periféricos.

AT90USB: Atmega integrado con controlador USB

AT90CAN: Atmega con controlador de bus CAN

AT90S: tipos obsoletos, los AVR clásicos

2.1.4.5.2. Modelos de tarjetas Arduino

En la comunidad Arduino existen una gran variedad de tarjetas Arduino creadas oficialmente y no oficialmente creadas por terceros, pero con características similares.

Las placas Arduino responden a las diferentes extensiones que se han venido realizando de las primeras placas de acuerdo con las demandas específicas de los usuarios y las investigaciones ya que son de tecnología abierta. La función de una u otra placa dependerá

del tamaño del proyecto a realizar. Las placas Arduino más comunes utilizadas en el campo de la tecnología se muestran en la imagen siguiente.

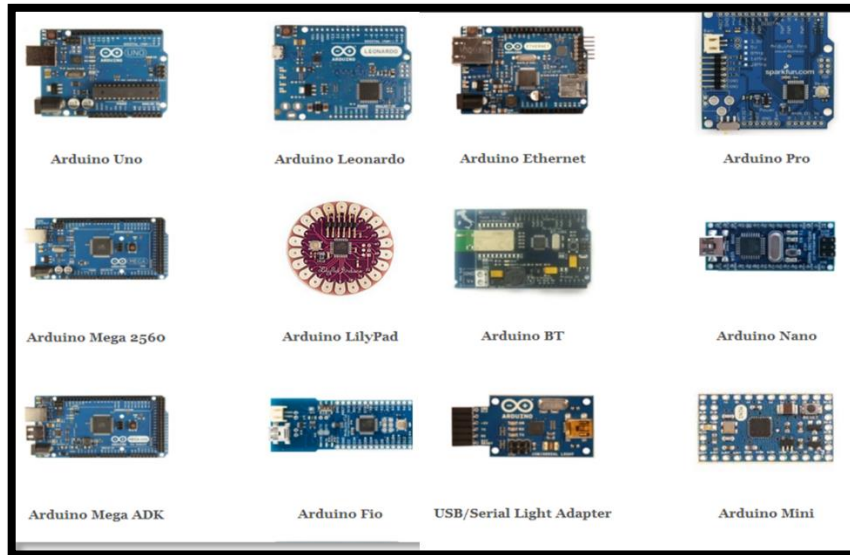


Figura 19 Modelos Placas Arduino

Fuente: [https://2.bp.blogspot.com/-](https://2.bp.blogspot.com/-KPPj6kznJzI/VCWqmqR_D6I/AAAAAAAAABV4/1qag9_ihgw/s1600/arduinos.png)

[KPPj6kznJzI/VCWqmqR_D6I/AAAAAAAAABV4/1qag9_ihgw/s1600/arduinos.png](https://2.bp.blogspot.com/-KPPj6kznJzI/VCWqmqR_D6I/AAAAAAAAABV4/1qag9_ihgw/s1600/arduinos.png)

2.1.4.6. Arduino Uno

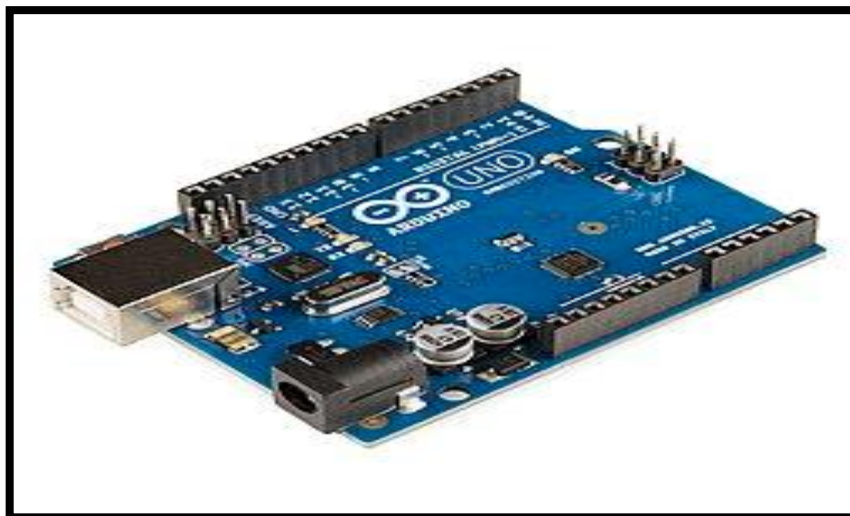


Figura 20 Arduino Uno

Fuente: https://upload.wikimedia.org/wikipedia/commons/thumb/3/38/Arduino_Uno_-_R3.jpg/375px-Arduino_Uno_-_R3.jpg

2.1.4.6.1. Hardware

Arduino Uno es la plataforma más extendida y la primera que salió al mercado. Se basa en un microcontrolador Atmel ATmega328 de 8 bits a 16Mhz que funciona a 5v. 32KB son correspondientes a la memoria flash (0,5KB reservados para el bootloader-gestor de arranque), 2KB de SRAM y 1KB de EEPROM. En cuanto a memoria es una de las placas más limitadas, pero no por ello resulta insuficiente para casi todos los proyectos que rondan la red. Las salidas pueden trabajar a voltajes superiores, de entre 6 y 20v, pero se recomienda una tensión de trabajo de entre 7 y 12v. Contiene 14 pines digitales, 6 de ellos se pueden emplear como PWM. En cuanto a pines analógicos se cuenta con hasta 6. Estos pines pueden trabajar con intensidades de corriente de hasta 40mA. También cuenta con un cristal de 16 MHZ oscilador, una conexión USB, un conector de alimentación, una cabecera ICSP (In Circuit Serial Programming) que sirve para programar el BootLoader del Microcontrolador ATmega y así poder cargar los programas creados en el IDE directamente en el microcontrolador sin tener que necesitar Programadores externos, y el botón de reinicio. El diseño ha hecho posible que el microcontrolador se pueda conectar por medio de un cable USB al ordenador o el poder con un adaptador AC-DC o batería para empezar.

Descripción de la placa Arduino

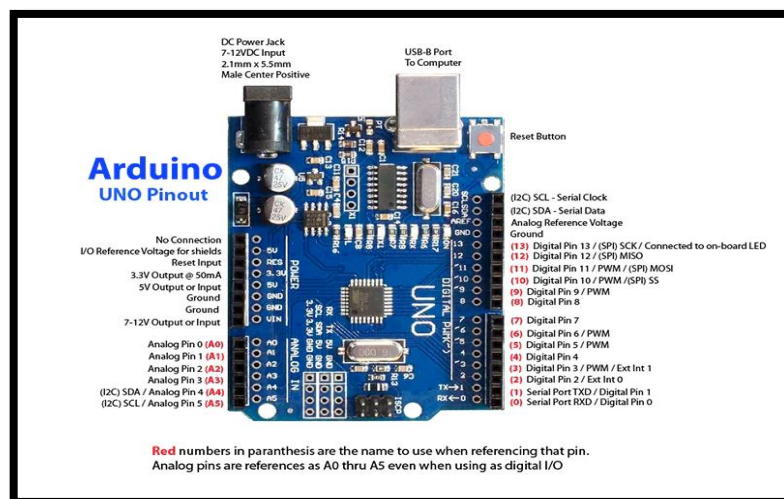


Figura 21 Pines Placa Arduino

Fuente: <https://tienda.sawers.com.bo/image/cache/catalog/00014-1-500x500.jpg>

- 1. Conector USB:** proporciona la comunicación para la programación y la toma de datos, también provee una fuente de 5VDC para alimentar al Arduino, pero de baja corriente por lo que no sirve para alimentar motores de gran potencia.
- 2. Regulador de voltaje de 5V:** Se encarga de convertir el voltaje ingresado por el plug 3, en un voltaje de 5V regulado necesario para el funcionamiento de la placa y para alimentar circuitos externos.
- 3. Plug de conexión para fuente de alimentación externa:** Es el voltaje que se suministra que debe ser directo y estar entre 6V y 18V o hasta 20V, generalmente se debe de tener cuidado de que el terminal del centro del plug quede conectado a positivo ya que algunos adaptadores traen la opción de intercambiar la polaridad de los cables.
- 4. Puerto de conexiones:** Es constituido por 6 pines de conexión con las funciones de RESET que permite resetear el microcontrolador al enviarle un cero lógico. Pin 3.3 V provee de una fuente de 3.3 VDC para conectar dispositivos externos como en la protoboard, por ejemplo. Pin 5V es una fuente de 5 VDC para conectar dispositivos externos. Dos pines GND que permite la salida de cero voltios para dispositivos externos. Pin Vin, este pin está conectado con el positivo del plug 3 por lo que se usa para conectar la alimentación de la placa con una fuente externa de entre 6 y 12 VDC en lugar del plug 3 o la alimentación por el puerto USB.
- 5. Puertos de entradas análogas:** lugar donde se conectan las salidas de los sensores análogos. Estos pines solo funcionan como entradas recibiendo voltajes entre cero y cinco voltios directos.
- 6. Microcontrolador Atmega 328:** Implementado con los Arduino uno en la versión SMD del Arduino uno R2, se usa el mismo microcontrolador, pero en montaje superficial, en este caso las únicas ventajas son la reducción del peso y ganar un poco de espacio.
- 7. Botón reset:** Permite resetear el microcontrolador haciendo que reinicie el programa.
- 8. Pines de programación ICSP:** son usados para programar microcontroladores en protoboard o sobre circuitos impresos sin tener que retirarlos de su sitio.
- 9. Led ON:** enciende cuando el Arduino está encendido.

10. Leds de recepción y transmisión: se encienden cuando la tarjeta se comunica con el PC. El Tx indica transmisión de datos y el Rx recepción.

11. Puertos de conexiones de pines de entradas o salidas digitales: La configuración como entrada o salida debe ser incluida en el programa. Cuando se usa la terminal serial es conveniente no utilizar los pines cero (Rx) y uno (Tx). Los pines 3, 5 y 6 están precedidos por el símbolo ~, lo que indica que permiten su uso como salidas controladas por ancho de pulso PWM.

12. Puerto de conexiones 5 entradas o salidas adicionales: Las salidas 9, 10 y 11 permiten control por ancho de pulso; la salida 13 es un poco diferente pues tiene conectada una resistencia en serie, lo que permite conectar un led directamente entre ella y tierra. Finalmente hay una salida a tierra GND y un pin AREF que permite ser empleado como referencia para las entradas análogas.

13. Led pin 13: indica el estado en que se encuentra.

14. Pines de programación ICSP: son usados para programar microcontroladores en protoboard o sobre circuitos impresos sin tener que retirarlos de su sitio.

15. Chip de comunicación: Permite la conversión de serial a USB.

Microcontrolador Atmega 328

El microcontrolador ATMEGA 328 es un circuito integrado que contiene las partes funcionales de una Pc, como CPU, memorias (RAM) para datos, memorias (ROM, PROM, EPROM) para escribir el programa, pines de entrada y salida para la comunicación con el mundo exterior y algunos periféricos (comunicación serial, temporizador, convertidor A/D, etc.)

Se conoce que ATMEL fabrica los microcontroladores de la familia AVR, por lo que el microcontrolador es un chip que contiene memoria flash reprogramable. Las principales características de microcontroladores ATMEL son sus 28 pines y su estructura interna cuentan con 16k bytes de In-flash, sistema programable con lectura y escritura, 512 bytes de EEPROM, 1k bytes de SRAM, 23 líneas para propósito general I/O, 32 registros para propósito de trabajo general, temporizador y contadores.

El modo de la energía-abajo guarda el contenido del registro, pero se congela el oscilador, desactivando todas las funciones del chip de otra hasta el siguiente reinicio. En el modo de ahorro de energía, el contador asíncrono sigue funcionando, lo que permite al usuario mantener una base de temporizador, mientras que el resto del dispositivo está durmiendo. Esto permite muy rápida puesta en marcha con el bajo consumo de energía.

Características físicas de los periféricos

- Interface serie SPI maestro/esclavo.
- ADC de 10 bit y 8 canales.
- 8 canales de terminación simple
- 7 canales diferenciales
- 2 canales diferenciales con ganancia programable a 1x, 10x o 200x.
- 2 canales de PWM de 8 bit.
- Modulador de comparación de salida.
- 6 canales PWM con resolución programable desde 2 a 16 bits.
- Interface serie de dos hilos orientada a byte.
- Dos timer/contadores de 8 bits con prescaler separado y modo comparación.
- Dos timer/contadores extendidos de 16 bits con prescaler separado, modo comparación y modo de captura.
- Comparador analógico On-Chip
- Contador en tiempo real con oscilador separado
- Doble USART serie programable

Voltajes de funcionamiento

- 2.7- 5.5V (ATMEGA 328L)
- 4.5- 5.5V (ATMEGA 328)

Niveles de velocidad

- 0 – 8 MHZ (ATMEGA 328L)
- 0 – 16 MHZ (ATMEGA 328L)



Figura 22 Microcontrolador Atmega 328

Fuente: https://upload.wikimedia.org/wikipedia/commons/thumb/9/9b/ATMEGA328P-PU_%282%29.jpg/330px-ATMEGA328P-PU_%282%29.jpg

2.1.4.6.2. Software

Como se había mencionado, Arduino, no sólo son componentes eléctricos ni una placa de circuitos, sino que, además, también es una plataforma que combina esto con un lenguaje de programación que sirve para controlar los distintos sensores que se encuentran conectados a la placa, por medio de instrucciones y parámetros que establecemos al conectar la placa a un ordenador. Este lenguaje que opera dentro de Arduino se llama Wiring, basado en la plataforma Processing y primordialmente en el lenguaje de programación C/C++, que se ha vuelto popular a tal grado de ser el más preferido para enseñar programación a alumnos de nivel superior que estudian computación y robótica, gracias que es muy fácil de aprender y brinda soporte para cualquier necesidad de computación. De este lenguaje derivan otros más que son muy utilizados en el ámbito de Ingeniería y desarrollo, como C#, Java, BASIC, Php, Python, JavaScript, Perl, entre otros más; por lo tanto, Arduino soporta varios lenguajes de programación de alto nivel derivados de C, haciendo de esto una ventaja para los diseñadores que trabajan en varios o en un sólo entorno de desarrollo de programación.

Para poder trabajar desde el nivel de programación del procesador, debe descargarse el software que incluye las librerías necesarias para poder utilizar el lenguaje de manera

completa. Otra ventaja es que este software puede descargarse desde el sitio web oficial de Arduino, ya que opera bajo licencia libre y está disponible a todo público.

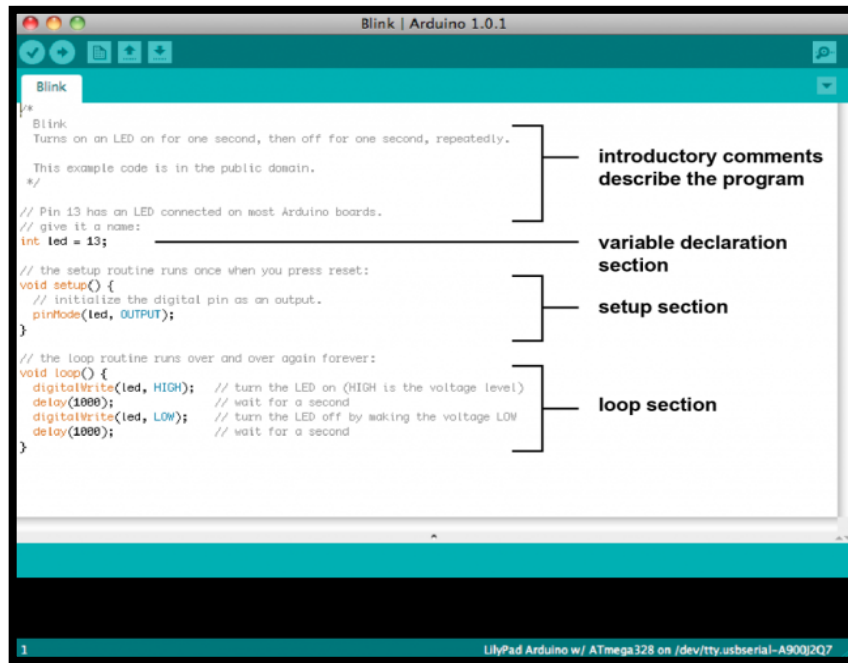


Figura 23 Entorno del programa Arduino

Fuente: <https://aprendiendoarduino.files.wordpress.com/2014/11/435ee-arduino-sketch-blink-anno-1024x844.png?w=625&h=515&zoom=2>

CAPITULO III
INGENIERIA DEL PROYECTO

3.1. DESARROLLO DEL TRABAJO

3.1.1. Diagrama en bloques

El proyecto está constituido por un teclado, el cual es el medio de acceso al dispositivo, junto con dos módulos; el primer módulo es la placa en la cual está ubicada el microcontrolador, el cual recibe y ejecuta las instrucciones dadas desde el tablero que está disponible para el usuario.

Para la realización de este proyecto se vio por conveniente la utilización de la plataforma Arduino Uno, el segundo modulo es un circuito electrónico que está constituida por un shield donde los circuitos instalados ayudan a que Arduino Uno logre reproducir la información que se encuentra en una tarjeta MicroSD, esta tarjeta contiene grabaciones de los mensajes según el dominio asignado por medio del teclado, además se integra otro modulo formado por un amplificador de audio que posteriormente será conectado a un parlante para la salida de audio.

Se puede apreciar que gracias a la diversidad de la plataforma Arduino Uno utilizada podemos simplemente conectar una fuente de 9 [v] a la placa para alimentarla y desechar por ende la construcción de una fuente externa que requeriría mayor trabajo y presupuesto. Se ha dividido en fases de forma gráfica como indica la figura 24, con la finalidad de obtener una idea clara de cómo funciona el circuito para el dispositivo electrónico de comunicación alternativa-aumentativa:

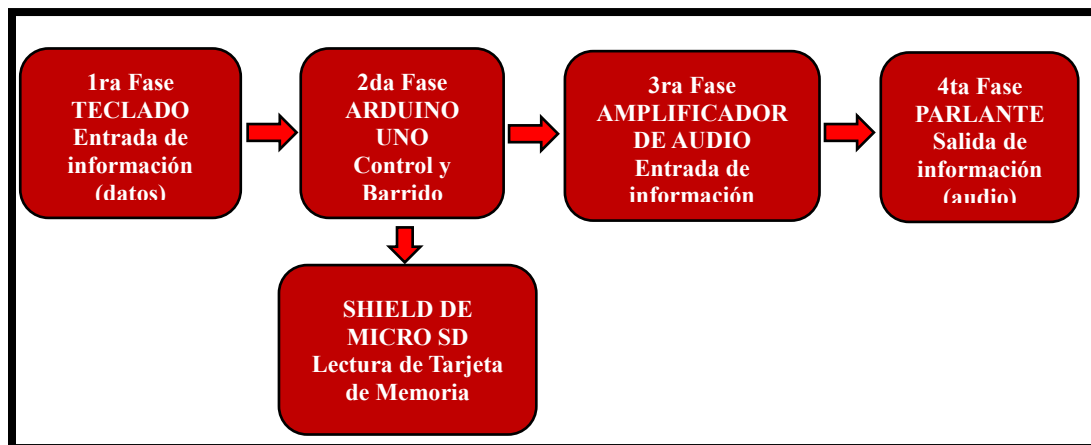


Figura 24 Diagrama en bloques del prototipo

Fuente: El autor

Inicia con la primera fase que representa el teclado, con una matriz de 3*3, es decir la combinación de columnas (X1,...X3) y filas (Y1,..Y3) que forman dicha matriz, a través de él ingresa la información al Arduino Uno que gracias al lenguaje de programación grabado en esta plataforma reconoce los pulsos que manda el teclado y logra establecer que archivo de audio se reproducirá considerando que este archivo necesita un formato de grabación particular que se detallara más adelante, refiriéndonos al hardware, para poder leer el archivo de sonido, se vio la necesidad de instalar un circuito que ayudara al Arduino a leer la información de una tarjeta MicroSD ya que recién realizado esto se podría definir que archivo sería reproducido. Con esto pasa a la tercera fase que es el amplificador de audio que es un circuito diseñado para que Arduino logre como su nombre lo indica amplificar la señal y filtrar esta, logrado esto finalmente se podrá escuchar el archivo de sonido gracias a la instalación en la última fase de un parlante.

El Microcontrolador Atmega 328 forma la parte lógica de la plataforma Arduino Uno. Sus funciones específicas son controlar el tipo de actividad que está realizando el tablero y reproducir el mensaje. Dentro de las funciones existen limitantes que deben ser controladas por el microcontrolador como son:

- Identificar la ubicación relativa del pulso que recibe dentro del programa.
- Eliminar el efecto rebote.
- Controlar el estado de la memoria de almacenamiento de los mensajes.
- Así como la ubicación lógica del mensaje dentro de la memoria MicroSD.

3.1.2. Teclado Matricial

Al conectar un pulsador en la plataforma arduino, requerimos de una entrada digital. Si el proyecto planteado requiere de varios pulsadores, vamos a tener el problema de que los pines digitales se van a terminar y no se podrá conectar más dispositivos al Arduino.

Es por esto por lo que se utiliza un arreglo de pulsadores en forma matricial, permitiéndonos así, utilizar la menor cantidad de pines en la plataforma Arduino Uno.

El proyecto planteado requiere de 9 pulsadores, pero solo utilizaríamos 6 pines (3 filas y 3 columnas) en lugar de 9, que sería ideal para poder conectarnos a otros dispositivos de la plataforma Arduino.

Esta disposición se puede apreciar en la siguiente imagen.

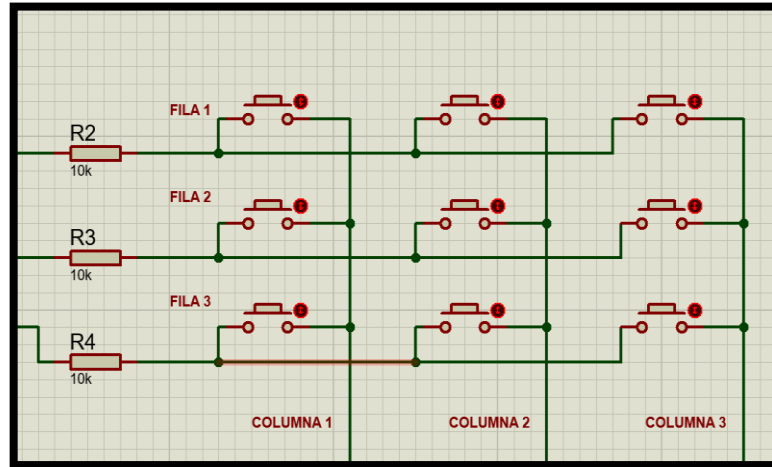


Figura 25 Pulsadores en forma Matricial en el simulador Proteus

Fuente: El autor

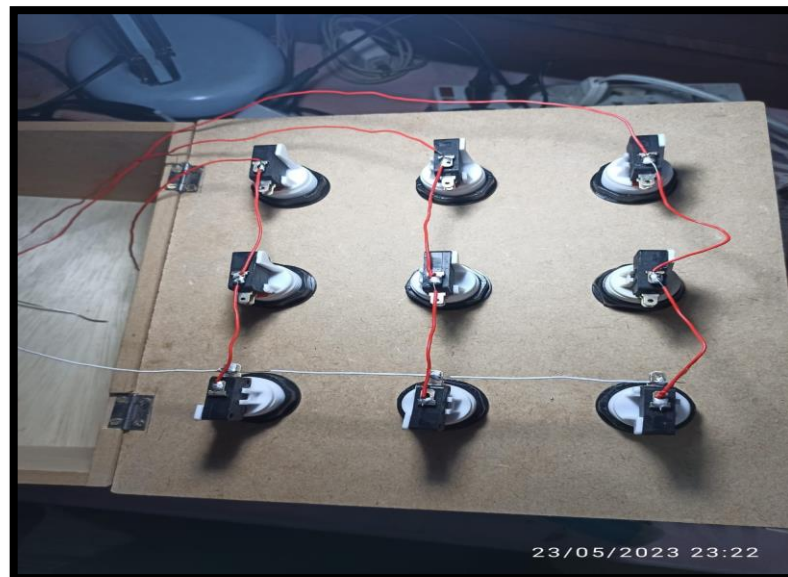


Figura 26 Construcción del Teclado Matricial

Fuente: El autor

3.1.2.1. Funcionamiento

El sistema en el que se basa el funcionamiento de un teclado matricial es el multiplexado. Configuramos tres pines del microprocesador como entradas y otros tres como salidas. Conectamos las entradas a las columnas y las salidas a las filas (o viceversa porque el

funcionamiento sería el mismo). Empezamos con las tres salidas en nivel bajo. Activamos la salida de la primera fila y leemos las tres entradas.

Cada una de ellas nos informará del estado de los tres pulsadores de esa fila. La entrada 1 el pulsador de la columna 1, la entrada 2 el pulsador de la columna 2 y así sucesivamente. A continuación, desactivamos la salida 1 y activamos la 2. En este momento en las cinco entradas podemos leer el estado de los pulsadores situados en la segunda fila. Haremos lo mismo con las salidas 2 y 3 para obtener el estado de los pulsadores situados en las filas 2 y 3.

De una manera mucho más sencilla lo podríamos resumir este procedimiento en los siguientes pasos:

1. Primero se energiza la primera fila, después se evalúa si hay señal en alguna de las columnas, si aparece señal, significa que una tecla de la primera fila y correspondiente a la columna detectada, se presionó.
2. Después se pasa a energizar la segunda fila y desconectar la primera, se evalúa si hay señal en alguna de las columnas, si aparece señal, significa que una tecla de la segunda fila y correspondiente a la columna detectada, se presionó.
3. Se hace el mismo procedimiento para las filas 3 y 4.
4. Volvemos a comenzar con la fila 1 y el ciclo se repite infinitamente.

Para hacer esto posible también se debe hacer algo de programación extra, teniendo así un código un poco más complejo. Pero esto no es muy importante ya que Arduino tiene varias librerías externas para controlar estos dispositivos.

La librería que utilizaremos se llama Keypad.h y las descargaremos en este link <http://playground.arduino.cc/code/Keypad>

Descargaremos el fichero comprimido zip al ordenador y lo desempaquetamos. Con esta operación obtendremos una carpeta llamada keypad. Esta carpeta la copiaremos en la carpeta 'libraries' del directorio donde se encuentra instalado el compilador Arduino en el equipo. En nuestro caso la ruta de acceso es: C: Archivos de programa /Arduino/libraries.

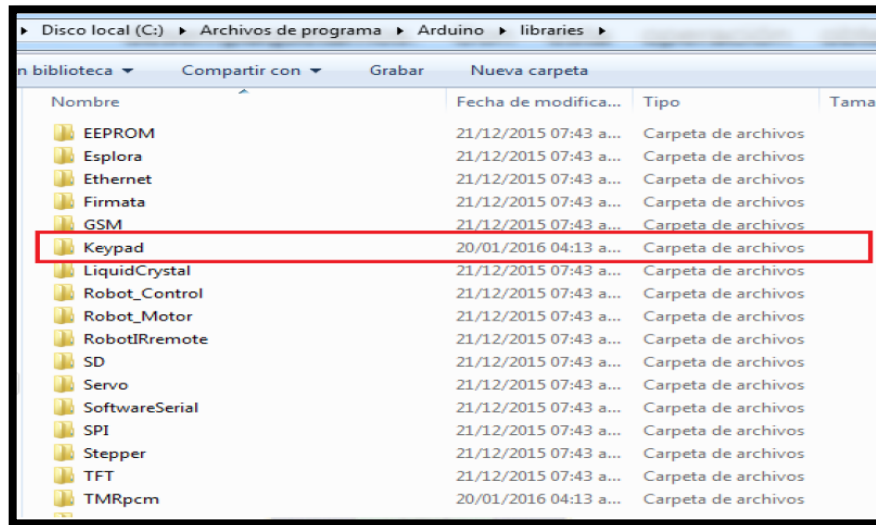


Figura 27 Ubicación librería Keypad

Fuente: El autor

Después de descargar e instalar la librería procedemos a conectar el circuito y ya podremos codificar el programa que se utilizara.

3.1.2.2. Conexiones

En la siguiente imagen se muestra una manera sencilla de como conectar el teclado matricial al Arduino

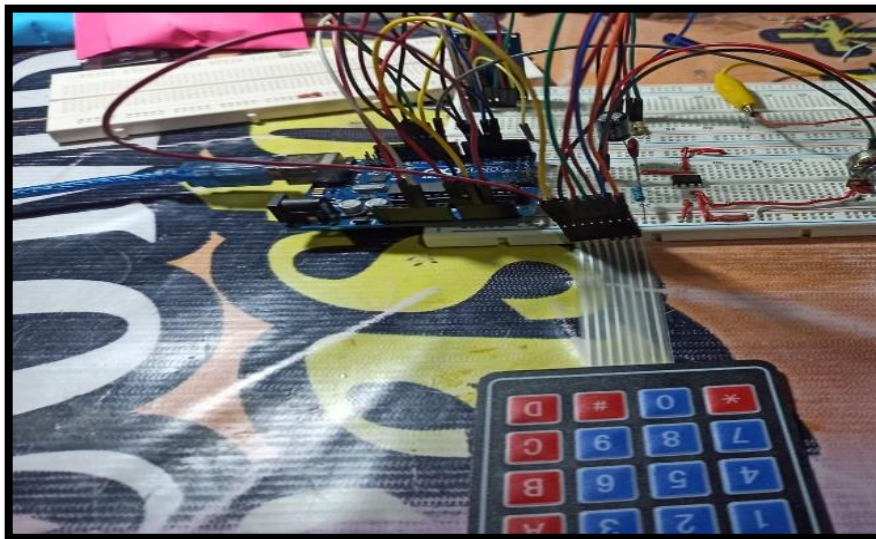


Figura 28 Conexión Teclado Matricial a Arduino

Fuente: El autor

La imagen anterior nos muestra una conexión sencilla de un teclado matricial 4*4 con Arduino Uno.

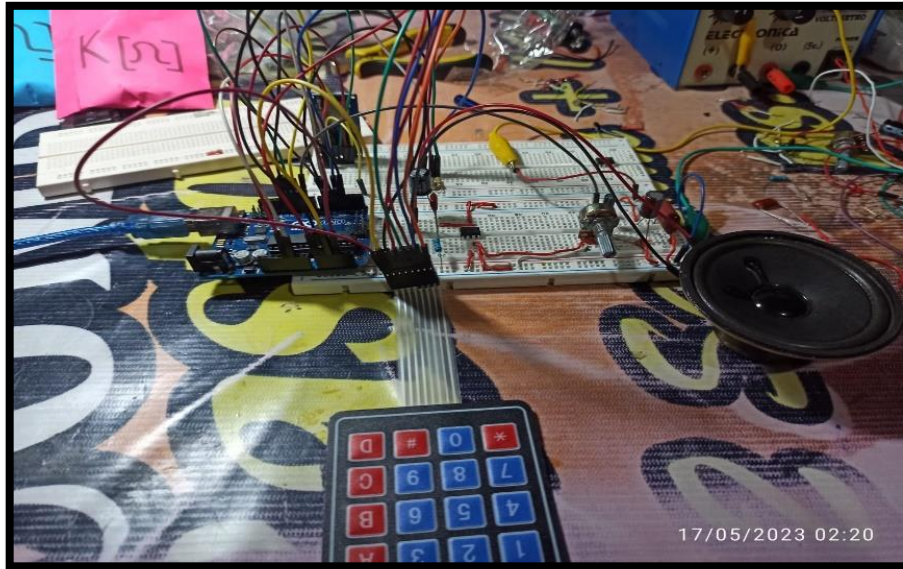


Figura 29 Pruebas del Circuito Intercomunicador

Fuente: El autor

Luego de haber finalizado las pruebas del teclado y haber obtenido resultados satisfactorios recién se pudo redireccionar estos datos al teclado 3*3.

Las primeras 3 líneas del teclado matricial que corresponden a las filas las conectamos del pin D3 al pin D5 del Arduino Uno respectivamente, las 3 líneas restantes del teclado matricial que corresponden a las columnas las conectamos del pin 14 al pin 16 respectivamente (A0-A2) entradas analógicas. Los pines 0 y 1 del Arduino que corresponden al puerto serial, los dejamos libres para poder visualizar en la terminal el carácter que corresponde a cada botón presionado.

3.1.3. Diseño y elaboración del esquema de audio

La placa Arduino uno da muchas satisfacciones, la documentación que tiene y sus librerías le hacen totalmente manejable para distintos montajes, es camaleónico. Su secreto es la documentación de su microcontrolador y la facilidad de programación de su entorno, y quizás sea esa la base de su éxito.

En el proyecto planteado se realizó un reproductor de audio, para que Arduino Uno pueda reproducir la información dada desde una tarjeta MicroSD. Pero primero debemos crear los archivos de audio que necesitamos.

3.1.3.1. Creación y conversión de Archivos

Para crear los archivos de audio usamos el programa Loquendo, donde escribimos la palabra que queremos convertir en audio, seleccionamos la velocidad, el tipo de voz que queremos escuchar (mujer, hombre), la velocidad con lo que es leído la palabra escrita y direccionamos la grabación a una carpeta.

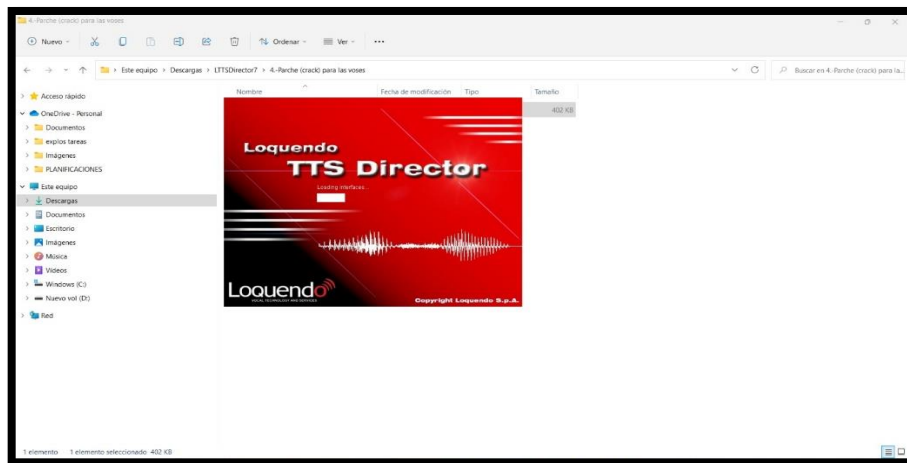


Figura 30 Ingreso al programa Loquendo

Fuente: El autor



Figura 31 Entorno del programa Loquendo

Fuente: El autor

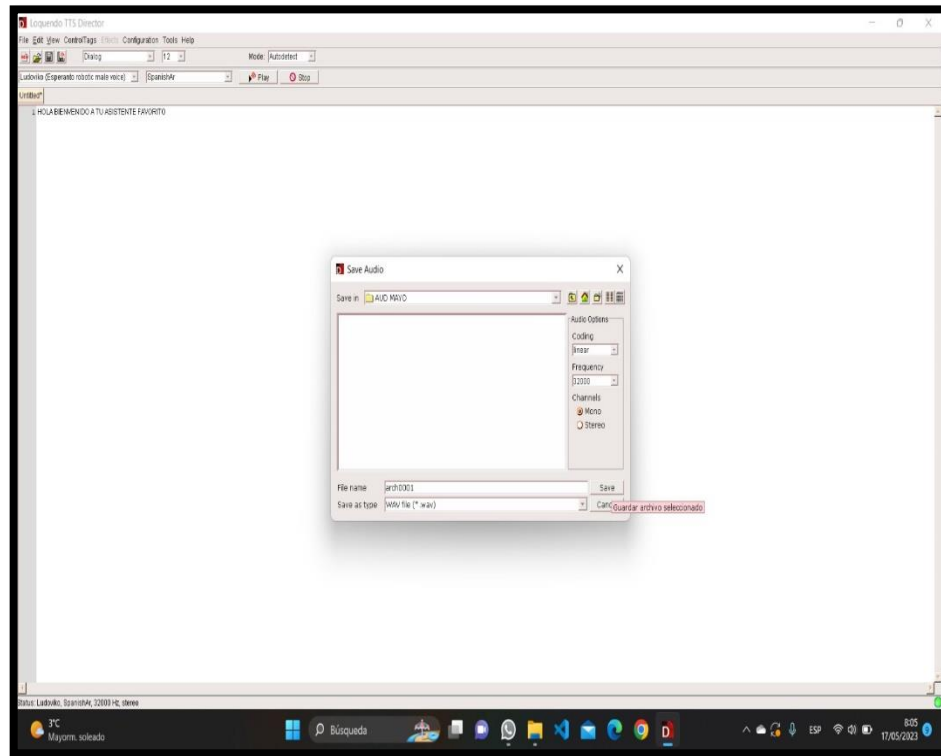


Figura 32 Grabación de palabras en Loquendo

Fuente: El autor

Finalizado este proceso y ya grabadas las 9 palabras que se utilizarán más un saludo inicial, debemos grabarlas a una tarjeta MicroSD, pero antes debemos considerar que Arduino Uno requiere que estos archivos tengan un formato especial, estos deben tener un formato de 8 bits y una frecuencia de 11 [KHz]. Lastimosamente el programa Loquendo utilizado no nos brinda estas características de formato por lo que una vez obtenidos los archivos de audio debemos convertir estos.

Por ello se vio la necesidad de utilizar otro programa similar, en este caso se optó por un convertidor online para realizar la conversión de estos archivos extremadamente pequeños, el cual es ilustrado en las imágenes a continuación:

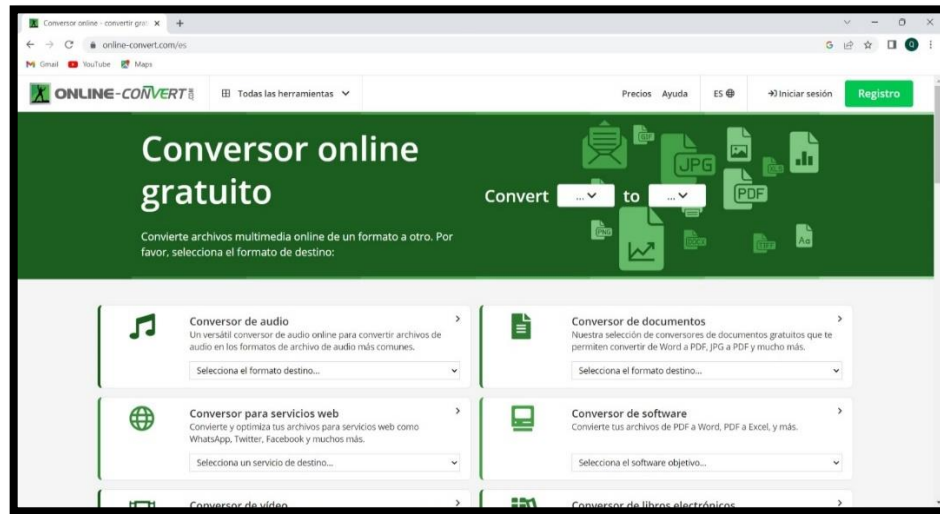


Figura 33 *Página de inicio Online – Convert*

Fuente: El autor

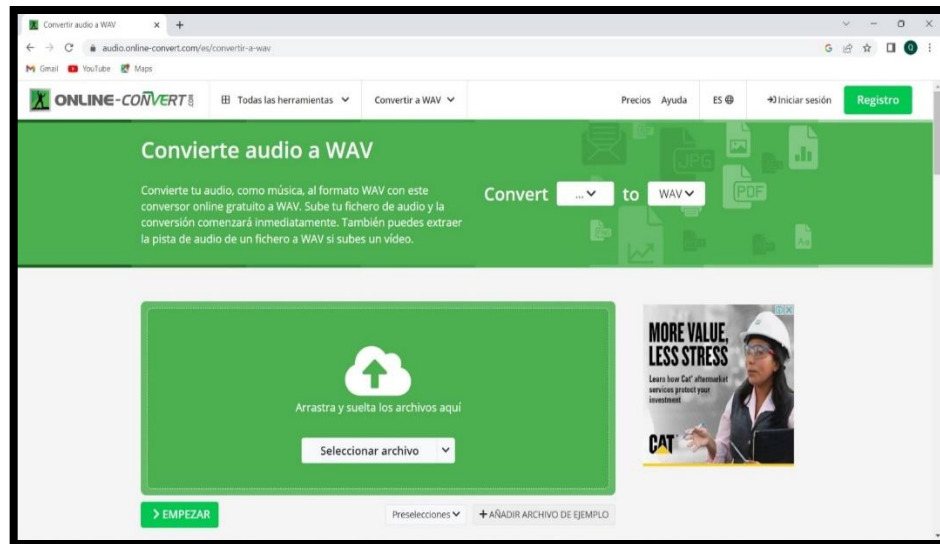


Figura 34 *Entorno de Online - Convert para convertir archivos .wav*

Fuente: El autor

Para realizar la conversión de los archivos de audio usando un conversor online primero accedemos a la página web de este conversor que es: [http:// www.audio.online-convert.com/es/convertir-a-wav](http://www.audio.online-convert.com/es/convertir-a-wav), una vez ingresado a esta página se obtendrá un entorno igual al mostrado en la imagen anterior, se debe hacer un click en el botón que dice Seleccionar archivo, al realizar esto podremos direccionar el archivo de audio deseado

desde las carpetas donde se guardaron los archivos de audio creados con Loquendo al programa conversor online, se debe esperar algunos minutos hasta que la página web cargue el archivo seleccionado al programa, una vez cargado este se deben elegir algunos parámetros de conversión para el archivo de audio seleccionado. Estos parámetros requeridos son mostrados en la imagen siguiente:

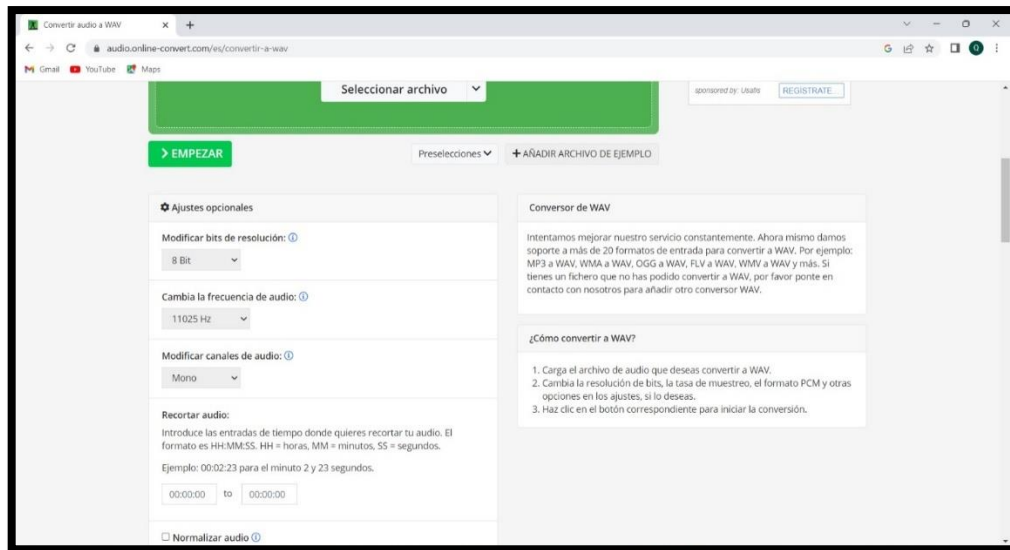


Figura 35 Formato de conversión Online – Convert

Fuente: El autor

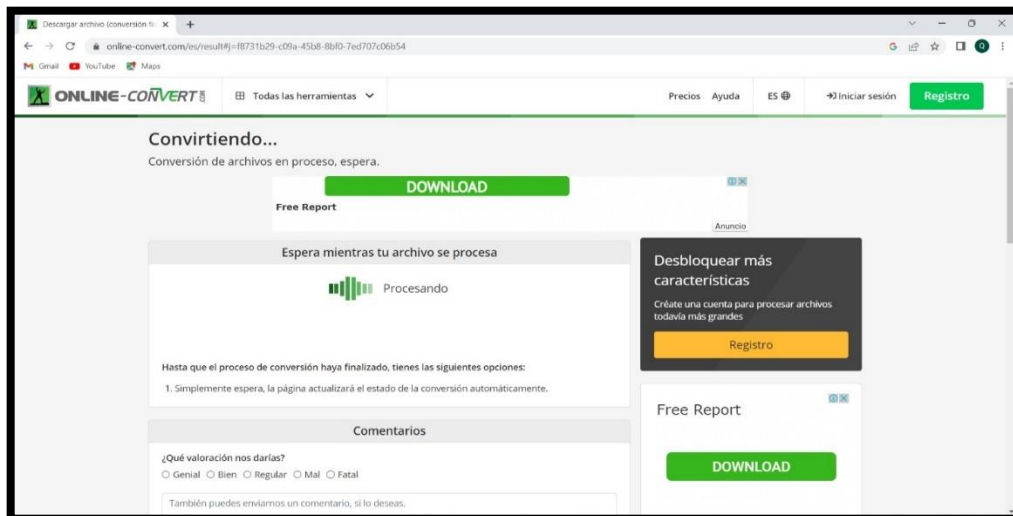


Figura 36 Conversión del archivo .wav en Online – Convert

Fuente: El autor

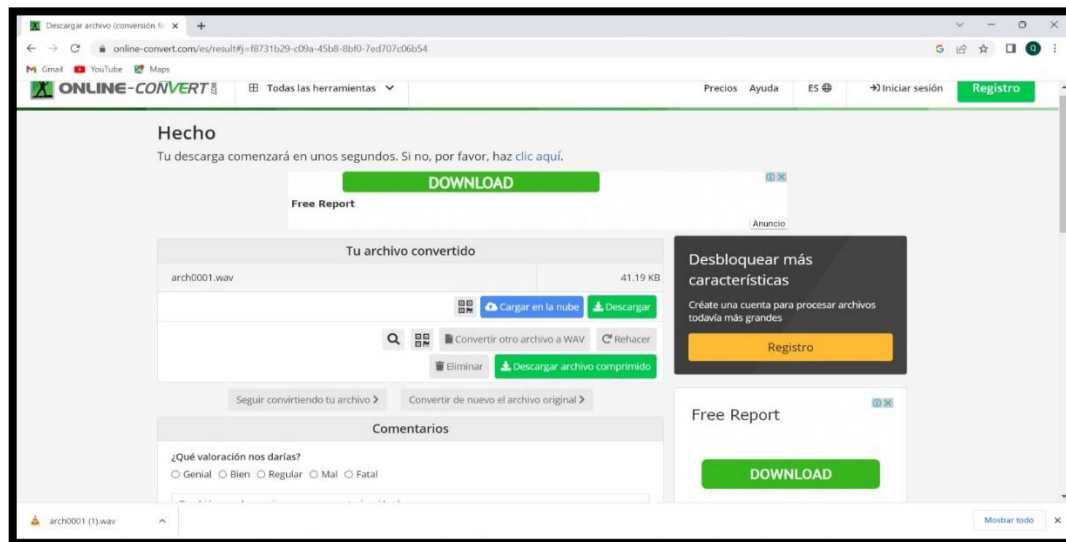


Figura 37 Descarga del archivo .wav en Online – Convert

Fuente: El autor

Se debe seleccionar dentro de los ajustes opcionales que nos muestra la página web los siguientes parámetros de conversión:

- Modificar bits de resolución: 8Bit
- Modificar la tasa de muestreo: 11025 Hz
- Modificar canales de audio: mono

Una vez realizado esta selección, se da un click al botón de Convertir archivo, se espera unos minutos y automáticamente el archivo de audio que ya cuenta con el formato requerido se empieza a descargar y luego se aloja en la carpeta de Descargas de la computadora. Una vez terminado este proceso y finalizado la conversión de los archivos de audio wav uno a uno, debemos formatear la tarjeta μ SD que utilizaremos.

3.1.3.2. Preparación de la tarjeta Micro SD

Al adquirir una tarjeta MicroSD también se debe adquirir un lector USB para tarjeta MicroSD, pues así se podrá conectar la tarjeta a la computadora, una vez conectada la tarjeta por el puerto USB se debe preparar la tarjeta, se ingresa a la carpeta Equipo del ordenador, donde claramente se verá la tarjeta en la pantalla como un dispositivo extraíble, se selecciona la tarjeta se da un click derecho al mouse y se selecciona la opción Formatear..., una vez realizado esto aparecerá una ventana, donde se podrá seleccionar el

tipo de sistema de archivos en este caso será FAT (Predeterminado), se podrá cambiar la etiqueta del volumen donde se designó a la tarjeta Micro SD con el nombre de ARDUINO y se podrá elegir opciones de formato que para realizar un formato optimo no se seleccionó ninguna opción.

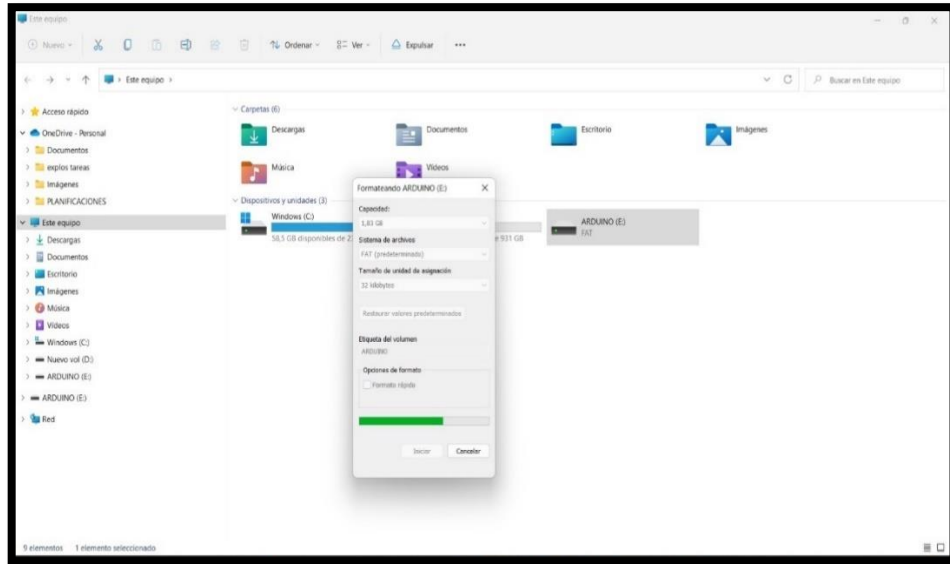


Figura 38 Formateo de la Tarjeta MicroSD

Fuente: El autor

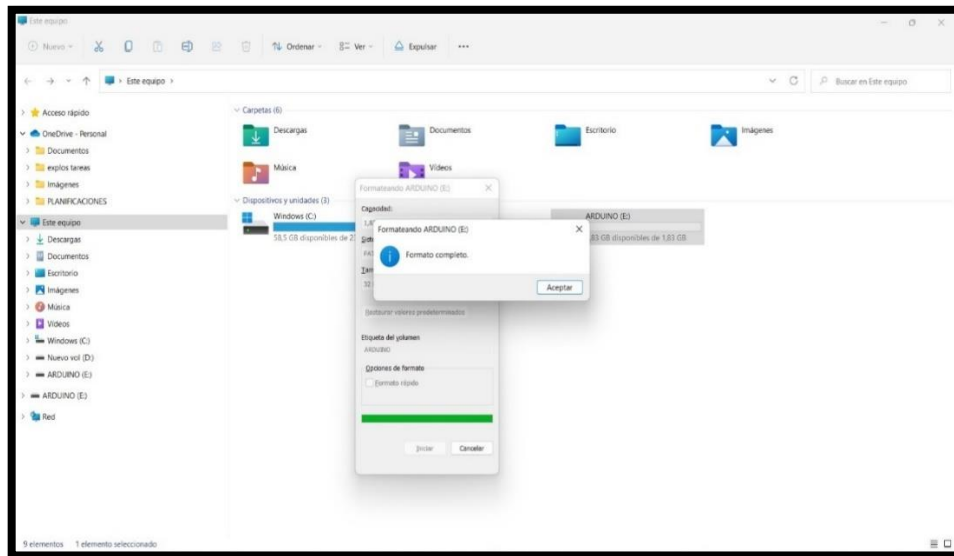


Figura 39 Formato completo de la Tarjeta MicroSD

Fuente: El autor

Se debe considerar que la tarjeta MicroSD no debe tener una capacidad superior a 2GB (no sería una MicroSD sino una MicroSDHC) ya que corremos el riesgo de que nuestra plataforma Arduino no la detecte.

Una vez formateada la tarjeta Micro SD se abre la carpeta donde fueron guardados los archivos de audio con el formato requerido, se selecciona todos los archivos se da un click derecho y se elige la opción copiar, luego se abre nuevamente la carpeta Equipo, se da dos clicks a dispositivo extraíble que es la tarjeta MicroSD nombrada con el nombre ARDUINO se da un click derecho con el mouse y se elige la opción pegar, se debe copiar los archivos de audio a la tarjeta sin meterlos en ninguna carpeta, directamente en la raíz de la MicroSD, además los archivos deben ser nombrados como se muestra en la siguiente imagen ya que así estarán direccionados en el código del programa planteado.

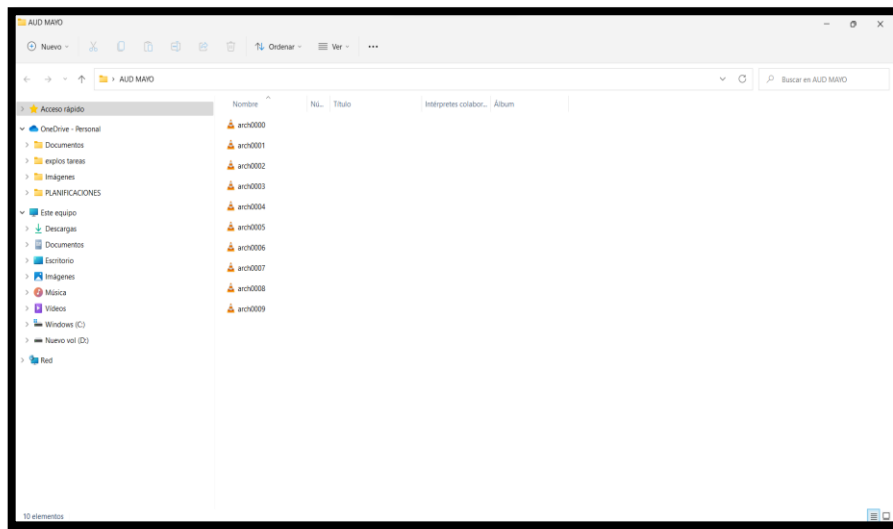


Figura 40 Archivos .wav con el formato requerido para Arduino

Fuente: El autor

El archivo de audio que contiene el saludo inicial será caratulado como arch0000, el archivo de audio que será direccionado para la primera tecla será denominado como arch0001, y así sucesivamente con el resto de los archivos de audio hasta llegar al último archivo que será denominado como arch0009, es importante mantener estos nombres para cada archivo de audio, pues en el programa que se utilizó para el proyecto se utilizó esta denominación para direccionar los archivos de audio.

Una vez finalizado este proceso nos concentraremos en el desarrollo del reproductor de audio, pues ahí introduciremos nuestra tarjeta MicroSD.

3.1.4. Descripción del circuito

Para una mayor comprensión del circuito empleado para realizar el dispositivo electrónico de comunicación alternativa-aumentativa se dividió la simulación anteriormente mostrada en tres etapas así podremos ver estas más detalladamente y podremos realizar una mejor explicación de estas.

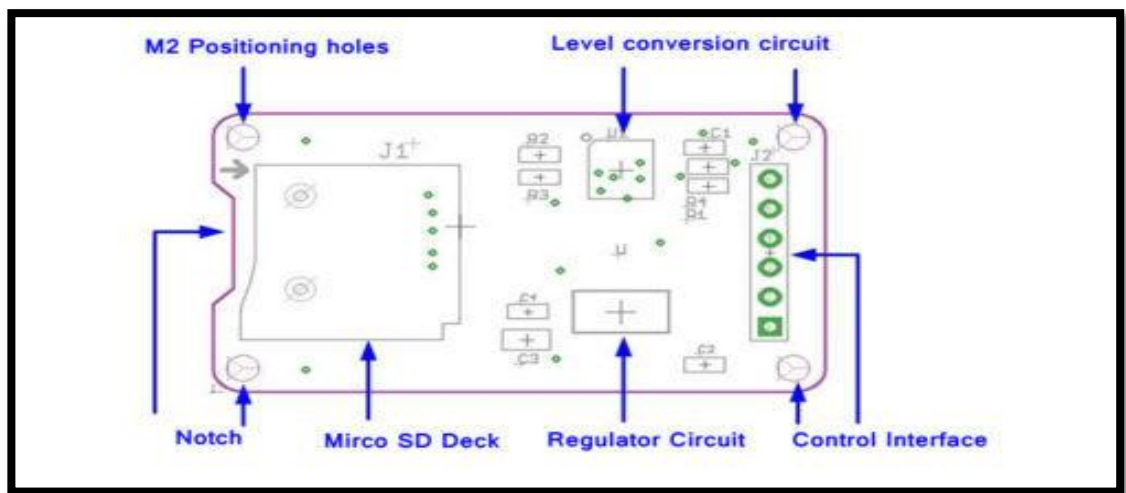


Figura 41 Estructura del Módulo lector de tarjetas MicroSD

Fuente: [https://4.bp.blogspot.com/-yBXqg7dx0tI/Wxv-](https://4.bp.blogspot.com/-yBXqg7dx0tI/Wxv-NY4qUjI/AAAAAAAAA_Y/M6UjT9sDriQ6o4x3aaqMUmYVVAgR61hjJgCEwYBhgL/s280/MicroSD%2BModule.png)

[NY4qUjI/AAAAAAAAA_Y/M6UjT9sDriQ6o4x3aaqMUmYVVAgR61hjJgCEwYBhgL/s280/MicroSD%2BModule.png](https://4.bp.blogspot.com/-yBXqg7dx0tI/Wxv-NY4qUjI/AAAAAAAAA_Y/M6UjT9sDriQ6o4x3aaqMUmYVVAgR61hjJgCEwYBhgL/s280/MicroSD%2BModule.png)

Para poder explicar esta sección, se debe conocer cómo se comunica una tarjeta SD, esta tarjeta posee dos protocolos de comunicación que son:

- Bus SD: Utiliza transferencias paralelas de 4 bits.
- SPI (Serial Peripheral Interface): Interfaz de Periféricos Serie.

Este último es el que nos interesa estudiar ya que se comunicara la tarjeta SD con Arduino, utilizando el protocolo de comunicación SPI.

El Bus SPI es un estándar de comunicaciones usado principalmente para la transferencia de información entre circuitos integrados en equipos electrónicos. Sirve para controlar

casi cualquier dispositivo electrónico digital que acepte un flujo de bits serie regulado por un reloj.

Este protocolo funciona con una configuración Master-Slave (maestro-esclavo), donde en nuestro caso el Arduino será el maestro, y la tarjeta SD será el esclavo.

Veamos:

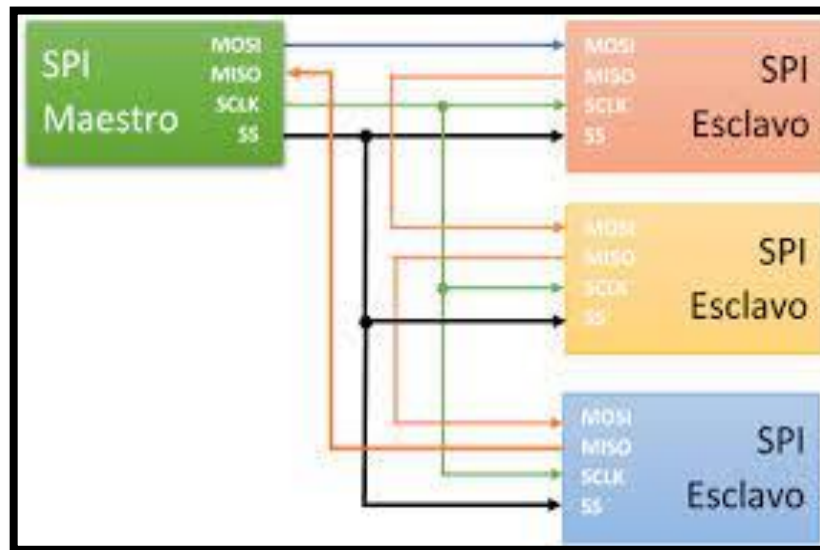


Figura 42 Configuración Maestro-Esclavo entre Arduino y la tarjeta MicroSD

Fuente:

https://upload.wikimedia.org/wikipedia/commons/thumb/f/fc/SPI_three_slaves.svg/1200px-

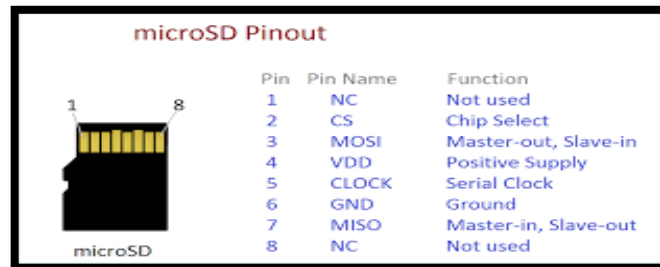
[SPI_three_slaves.svg.png](#)

Donde podemos apreciar que:

- CLK (Línea de reloj): Es la señal de reloj.
- MOSI (Master Out-Slave In): Línea por donde el maestro envía y el esclavo recibe.
- MISO (Master In-Slave Out): Línea por donde el maestro recibe y el esclavo
- CS (Chip Select) / SS (Slave Selector): Conecta o desconecta la operación del dispositivo con el que comunicamos. Permite la comunicación de varios esclavos a un mismo maestro, multiplexando la señal de reloj.

El maestro configura la velocidad de transmisión a la máxima permitida por el esclavo, y seguidamente pone a nivel bajo la señal “Selector Slave” para indicarle que va a

Dentro de la tarjeta MicroSD que se inserta en el modulo lector de MicroSD tenemos la siguiente distribución de pines:



Pin	Pin Name	Function
1	NC	Not used
2	CS	Chip Select
3	MOSI	Master-out, Slave-in
4	VDD	Positive Supply
5	CLOCK	Serial Clock
6	GND	Ground
7	MISO	Master-in, Slave-out
8	NC	Not used

Figura 44 Distribución de pines Tarjeta MicroSD

Fuente: <http://informmundo.blogspot.com/2020/06/los-pines-de-un-microsd-electronica.html>

Debemos considerar que las tarjetas MicroSD funcionan a un voltaje comprendido entre 2,7 V y 3,3 V máximo. Por lo que no podemos conectar la tarjeta directamente al Arduino, ya que los pines digitales funcionan a 5 V y quemaríamos la MicroSD.

El módulo lector de tarjetas MicroSD proporcionado por la empresa Arduino nos permite la comunicación SPI de forma directa con nuestra placa de Arduino Uno, sin necesidad de un regulador de tensión porque ya viene incluido dentro del módulo.

El módulo descrito anteriormente posee las siguientes características:

- Voltaje de alimentación: 3,3V o 5V
- Regulador de tensión incorporado 3,3V
- Consumo < 200mA
- Interfaz: SPI
- Dimensiones: 24mmx 42mm x 12mm



Figura 45 Modulo lector de Tarjetas MicroSD

Fuente: https://leantec.es/wp-content/uploads/2018/02/p_3_4_9_9_3499-MODULO-MICRO-SD-LECTOR-TARJETA-ARDUINO.jpg

Dada esta conexión lo único que se tiene que realizar es la habilitación de la librería SD de Arduino dentro de nuestra programación ya que esta librería permite la lectura y escritura en tarjetas SD, además que está incluida dentro de Arduino, la librería tiene la extensión <SD.h> pues dentro de la programación la forma de incluir esta librería al programa es con:

```
#include <SD.h>
```

La librería soporta tarjetas SD formateadas en FAT 32 y de hasta 2GB de capacidad.

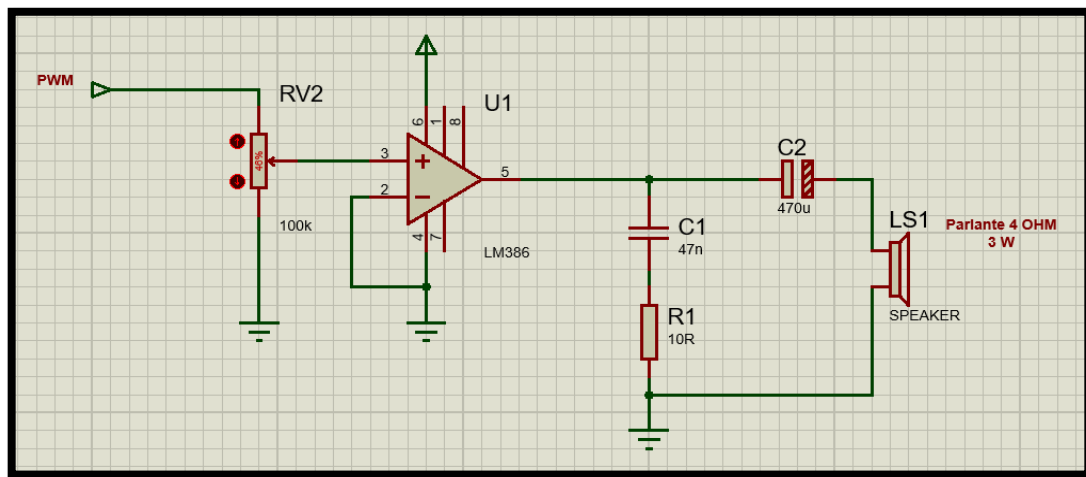


Figura 46 Circuito Amplificador de Audio

Fuente: El autor

En esta sección del circuito vemos que la señal que sale del pin 9 de la placa electrónica de Arduino es una señal PWM de 8 bits que ingresa al pin 3 del amplificador operacional LM386, debido a que Arduino Uno no puede producir una autentica señal analógica se debe emplear un “truco” que consiste en activar una salida digital durante un tiempo y mantenerla apagada durante el resto. El promedio de la tensión de salida a lo largo del tiempo, será igual al valor analógico deseado.

El LM386 es un amplificador de potencia diseñado para el empleo en usos de consumo de voltaje bajos. Existen muchas configuraciones distintas para este amplificador de audio, en este caso se implementará un amplificador cuya ganancia es igual a 20 (26 dB) que es el valor que se consigue al dejar libres los terminales 1 y 8, como lo muestra la siguiente imagen.

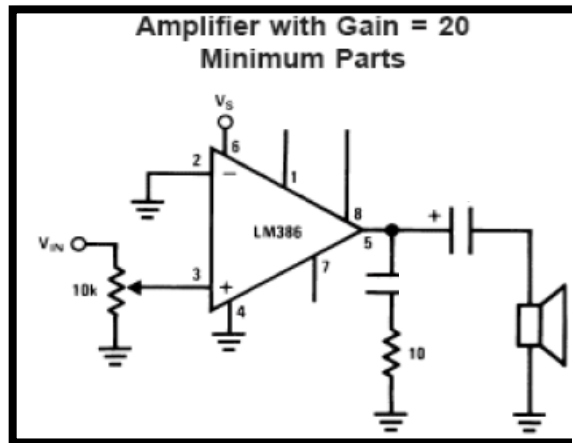


Figura 47 Circuito Amplificador

Fuente: <http://kagiva.galeon.com/3ds/tutores/opam.html>

Gain Control o Control de ganancia, son manejados por los pines 1 y 8. Cuando los pines 1 y 8 están abiertos como en el proyecto planteado la ganancia es de 20 (26 dB), si un capacitor es puesto desde el pin 1 al 8 la ganancia será superior a 200 (46 dB), si un resistor es conectado en serie con un capacitor la ganancia tendrá un rango de 20 a 200.

Un hecho común para el buen funcionamiento de los amplificadores operacionales es la necesaria alimentación simétrica, es decir una alimentación +V_s y -V_s, lo que permite una ganancia balanceada respecto de la señal.

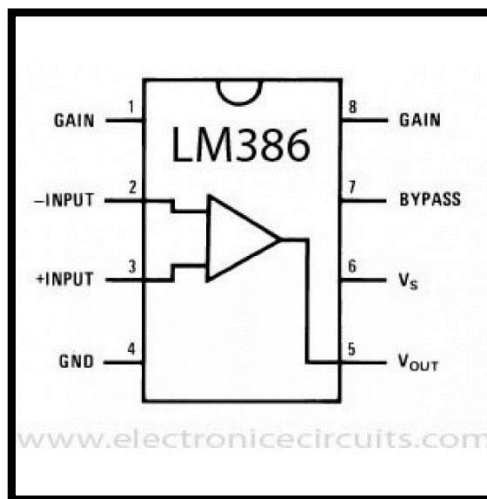


Figura 48 Esquema de Pines del LM386

Fuente: https://viasatelital.com/proyectos_electronicos/images/proyectos0022b.jpg

Para comprender este esquema se debe tener en cuenta varias cosas:

- Los pines del LM386 están numerados (como los de cualquier otro integrado), siendo el pin 1 el que se encuentra a la izquierda de la marca que se puede ver en la cápsula del integrado (normalmente un semicírculo o un pequeño punto) y el último pin (el 8 en este caso) es el de su derecha.
- V_s es la alimentación del amplificador (de entre 4V y 12V) y V_{in} es la señal que se quiere amplificar (la palabra grabada en el MicroSD), es decir, es donde irá el polo positivo, el polo negativo debe ir a tierra.
- Los “+” que se ve al lado del símbolo de algunos condensadores indican que se trata de condensadores polarizados se los debe conectar con el polo positivo en la dirección en la que se encuentra dicha marca.

Como ya se observó el circuito en esta etapa tiene como elemento principal un circuito integrado LM386, en el cual se han agrupado todos los componentes necesarios para conformar una etapa de potencia de audio. La señal de entrada pasa a través de un potenciómetro el cual hace la función de control de volumen ya que permite el paso de mayor o menor voltaje hacia la entrada del amplificador integrado (pin3). La salida amplificada sale por el pin 5 del LM386 y pasa a través del capacitor de desacople antes de llegar al parlante por donde saldrá el audio.

Ya que el pin 1 y 8 están libres tendremos una ganancia o factor de amplificación del circuito mínima igual a 20. Debemos considerar además que este circuito puede ser alimentado con una tensión de 9 o 12 VDC (pin6).

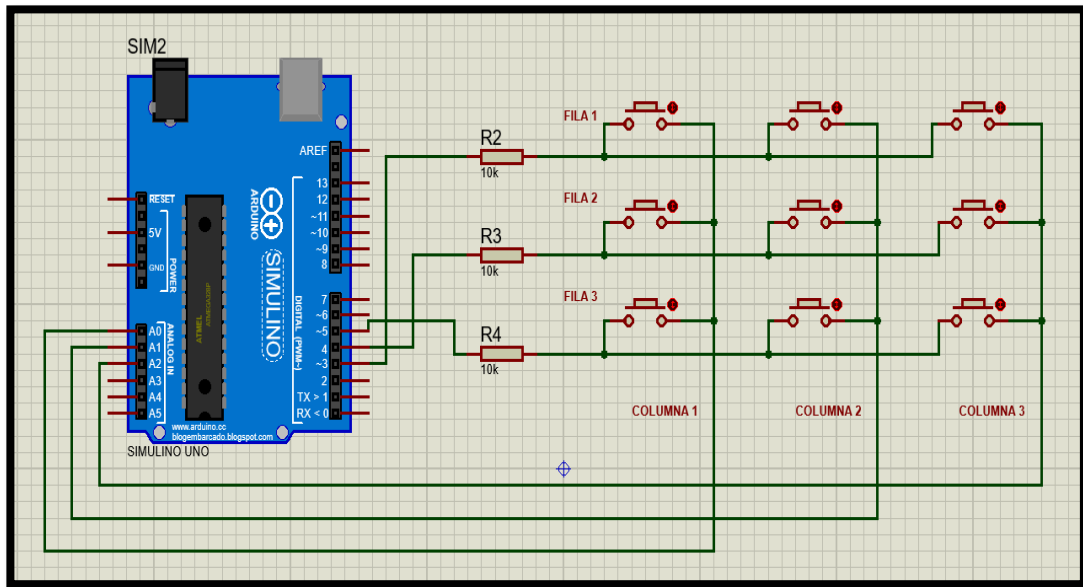


Figura 49 Conexión a la placa Arduino Uno del Teclado Matricial

Fuente: El autor

Esta etapa es mucho más sencilla en su explicación ya que simplemente se refleja a que pines de Arduino Uno fueron conectados las salidas del teclado matricial, al ser nuestro teclado una matriz 3*3 (3 columnas y 3 filas), vemos en la imagen anterior claramente que las columnas fueron conectadas a los pines A0 hasta el A2 respectivamente que son las entradas analógicas de Arduino Uno, y las filas fueron conectadas a los pines D3 hasta la D5 respectivamente junto con un conjunto de resistencias para dar protección al teclado matricial.

Una vez analizada las tres etapas de nuestro dispositivo electrónico de comunicación Alternativa – Aumentativa procedemos a mostrar un diagrama del circuito completo que se realizara en el presente proyecto.

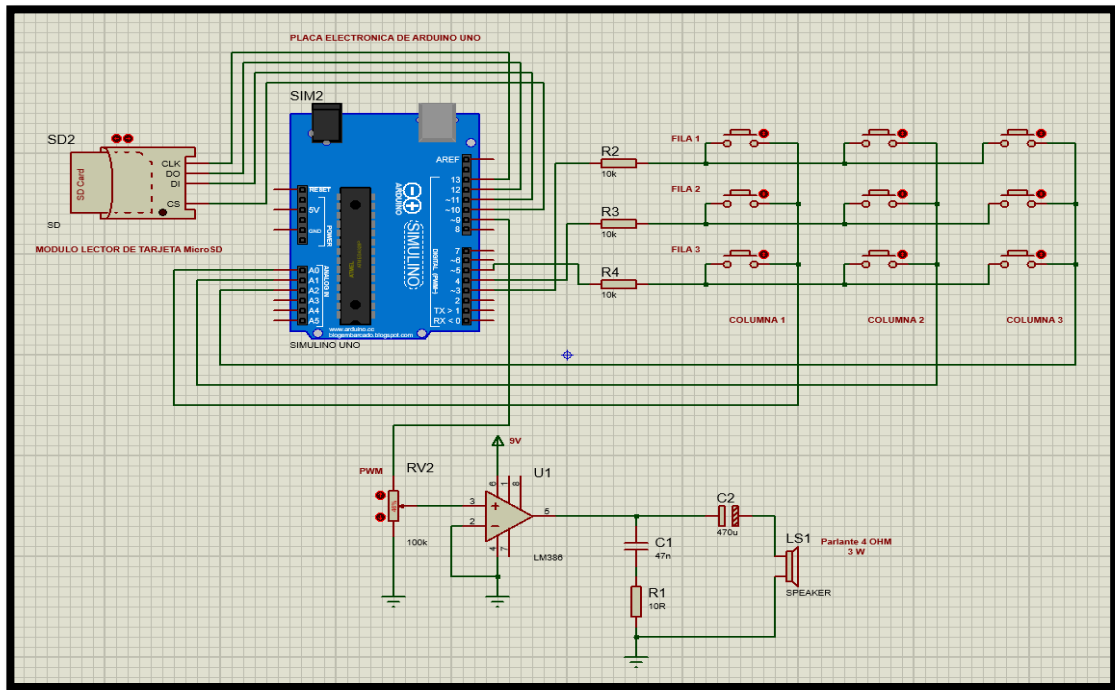


Figura 50 Diagrama completo Circuito electrónico de Comunicación

Fuente: El autor

3.1.5. Elaboración del programa

3.1.5.1. Codificación

Procedemos a la codificación del programa a utilizar, para esto es necesario conocer la estructura que debe tener el código en Arduino, esta estructura se muestra a continuación:

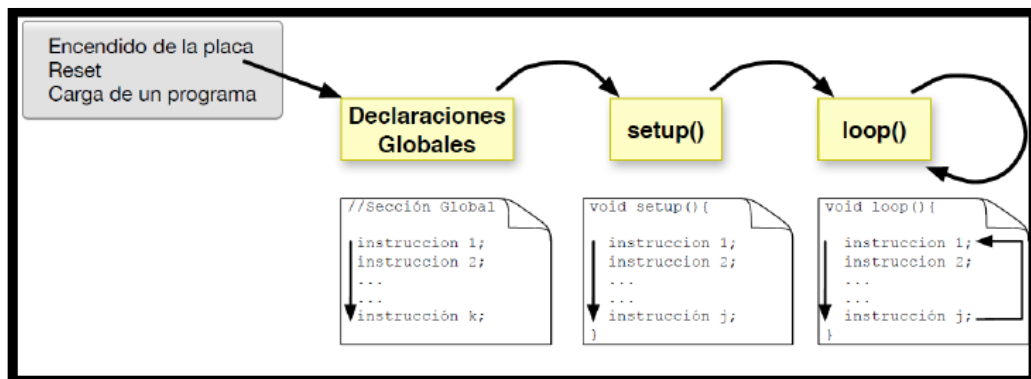


Figura 51 Estructura del código Arduino

Fuente: Programación en C para Arduino por Gabriel Astudillo Muñoz

En la primera sección del programa se realizarán las declaraciones globales, donde se deben poner las bibliotecas específicas que se utilizaran y las variables globales que se utilizaran. Este bloque se ejecuta una sola vez y bajo cualquier de los siguientes eventos:

- Encendido de la placa.
- Después de un reset.
- Después de cargar un programa desde el computador.

A continuación, mostramos esta primera sección de la programación utilizada en el proyecto planteado:

```
#include<Keypad.h> // llama a la librería del teclado matricial.
#include<SD.h> // llama a la librería para leer la tarjeta uSD.
#define chipSelect 10 // pin para seleccionar tarjeta uSD.
#include<TMRpcm.h> // llama a la librería para reproducir archivos wav.
TMRpcm tmrpcm; // crea un objeto para usar en este esquema
char tecla_pres;
const byte FILAS = 3; // 3 filas por
const byte COLUM = 3; // 3 columnas.
char teclas [FILAS][COLUM] = { // aquí pondremos la disposición de los caracteres tal cual
está en nuestro teclado.
{'1','2','3'},
{'4','5','6'},
{'7','8','9'}
};
byte rowPins [FILAS] = {3,4,5}; // pines conectados a las filas.
byte colPins [COLUM] = {14,15,16}; // pines conectados a las columnas.
Keypad Teclado = Keypad(makeKeymap(teclas), rowPins, colPins, FILAS, COLUM);
//inicializa el teclado.
```

En esta sección vemos que `#include`, hace referencia a una instrucción al preprocesador que está presente en la gran mayoría de lenguajes de alto y medio nivel, de forma genérica se usa para adicionar un archivo al código. El orden de las directivas `#include` no importa cuando todos los identificadores del programa son únicos.

La directiva `#define`, como su nombre lo indica define una constante, es una palabra clave del pre-procesador. Técnicamente se podría decir que no forma parte del lenguaje C. Antes que el compilador vea el código, el pre-procesador ya ha reemplazado la macro `chipSelect` por su definición – en nuestro caso el 10. Es decir, en todos los sitios donde aparece el identificador `chipSelect`, el compilador elige el pin 10 como si nosotros mismos hubiéramos escrito pin 10 en estos sitios. Es posible que el compilador ni siquiera pueda decirnos qué exactamente está mal si una compilación fracasa, porque no sabe nada del identificador `chipSelect`. Usando `#define` no estamos creando una constante, en realidad, se comporta igual que si se hubiese escrito el valor a mano.

Cuando definimos constantes mediante `const`, este método permite definir el tipo y el valor de la constante, en este caso un único valor es creado y en todas las partes en que sea usado será tratado como una referencia. El compilador puede comprobar el tipo de las constantes con `const` que las hace mejor que los `#define`. El tipo de variable `char` almacena un carácter utilizado en código ASCII números entre `[-128,127]`, ocupa un Byte de memoria (8 bits) y almacena un valor de carácter. El tipo de datos `byte` almacena números enteros de 8 bits sin puntos decimales, rango entre `[0,255]`, es preferible su uso desde el punto de vista de la coherencia del código.

La segunda sección inicia con la función `setup` se ejecuta después de la sección anterior y por una sola vez. Se utiliza para configurar el hardware que se utilizará. Se emplea para iniciar variables, establecer el estado de los pines, inicializar las librerías, etc. esta función se ejecutará una única vez después de que se conecte la placa Arduino a la fuente de alimentación, o cuando se pulse el botón de reinicio de la placa.

```
voidsetup () // void declara la función. Indica que no retorna un valor.
{
tmrpcm.speakerPin = 9; // Pin por donde sale audio
Serial.begin(9600); // Para la comunicación serial, se inicia el puerto serie
if (!SD.begin(chipSelect)) // Ve si la tarjeta esta presente y puede ser inicializada.
{
Serial.println("fallo SD!");
return; // no hace nada más, si no está presente la tarjeta o no es inicializada
```

```

}
tmrpcm.play("arch0000.wav"); // Se reproduce el archivo determinado.
Serial.println("SD correcto!");
}

```

Se pudo notar en esta sección el uso de Serial.begin y de Serial.println, por ello vemos necesario una breve explicación sobre estos, cuando hablamos de una comunicación serie esta se utiliza para la comunicación entre la placa Arduino y un ordenador u otro dispositivo. Todas las placas Arduino tienen al menos un puerto serie Serial. Se comunica a través de los pines digitales 0(Rx) y 1(TX), así como con el ordenador mediante USB. Por lo tanto, si se utiliza estas funciones, no se puede usar los pines 0 y 1 como entrada o salida digital.³⁶ Serial.begin, abre un puerto serie y establece la velocidad de datos en bits por segundo (baudios) para la transmisión de datos en serie, la velocidad típica para comunicación con el ordenador es de 9600 aunque se pueden soportar otras velocidades. Serial.println, imprime datos al puerto serie como texto ASCII seguido de un retorno de carro y un carácter de avance de línea automáticos. Este comando toma la misma forma que Serial.print(), pero es más fácil para leer datos en el Monitor Serial.

En la tercera sección se encuentra la función loop que ejecuta consecutivamente el código asignado de forma “perpetua”, permitiéndole al programa variar y responder. Es usada para controlar de forma activa la placa Arduino.

```

void loop() { // void declara la función.
  char tecla_pres = Teclado.getKey(); // La tecla presionada se asigna a la variable
  if (tecla_pres != NO_KEY) // se evalúa si se presionó una tecla.
  {
    switch (tecla_pres)
    {
      case '1':
        tmrpcm.play("arch0001.wav");
        Serial.println("Tecla 01");
        break;

```

```
case '2':
tmrpcm.play("arch0002.wav");
Serial.println("Tecla 02");
break;
case '3':
tmrpcm.play("arch0003.wav");
Serial.println("Tecla 03");
break;
case '4':
tmrpcm.play("arch0004.wav");
Serial.println("Tecla 04");
break;
case '5':
tmrpcm.play("arch0005.wav");
Serial.println("Tecla 05");
break;
case '6':
tmrpcm.play("arch0006.wav");
Serial.println("Tecla 06");
break;
case '7':
tmrpcm.play("arch0007.wav");
Serial.println("Tecla 07");
break;
case '8':
tmrpcm.play("arch0008.wav");
Serial.println("Tecla 08");
break;
case '9':
```

```
tmrpcm.play("arch0009.wav");  
Serial.println("Tecla 09");  
break;  
}  
}  
}
```

Vemos que en esta sección se utiliza la sentencia `if` que comprueba si cierta condición ha sido alcanzada y ejecuta todas las sentencias dentro de las llaves si la declaración es cierta. Si es falsa el programa ignora la sentencia.

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

La palabra clave `break` sale de la sentencia `switch`, y es usada típicamente al final de cada `case`. Sin esta sentencia `break`, la estructura `switch` continuaría ejecutándose hasta encontrar un `break` o hasta llegar al final de la sentencia `switch`.

Las llaves `{ }` definen el comienzo y el final de bloques de función y bloques de declaraciones como `voidloop()` y sentencias `for` e `if`. Las llaves deben estar balanceadas (a una llave de apertura `{` debe seguirle una llave de cierre `}`). Las llaves no balanceadas provocan errores de compilación. El entorno Arduino incluye una práctica característica para chequear el balance de llaves. Sólo selecciona una llave y su compañera lógica aparecerá resaltada.

Un punto y coma (`;`) debe usarse al final de cada declaración y separa los elementos del programa. También se usa para separar los elementos de un bucle `for`. Olvidar un punto y coma al final de una declaración producirá un error de compilación.

CAPITULO IV
CONCLUSIONES

4.1. CONCLUSIONES

Se pudo seleccionar el sistema de símbolos de comunicación adecuado que exprese palabras como “Divertido”, “Frio”, “Silencio”, “Que”, “Hablar”, “Beber”, “Leer”, “Comer” y “Quiero” que pueden expresar las necesidades de los estudiantes con discapacidad intelectual.

Se diseño un teclado electrónico de acuerdo con los símbolos establecidos para la efectiva comunicación de los estudiantes con sus maestros.

Se elaboro el código fuente que se grabará en el microcontrolador destinado a la funcionalidad del dispositivo electrónico de comunicación aumentativa - alternativa.

Se construyo un prototipo de un dispositivo electrónico de comunicación alternativa-aumentativa para estudiantes con discapacidad intelectual, que rompa las barreras comunicativas que presentan las personas con discapacidad y así contar con un ambiente favorable de enseñanza-aprendizaje.

Este proyecto tal como fue diseñado no es limitante ya que vemos que la tarjeta Micro SD puede ser fácilmente grabado por otras frases y los pictogramas ser cambiados, para así poder seguir avanzando en un proceso de enseñanza cada vez más desarrollado e ir avanzando según los requerimientos del alumno.

Podemos decir que el proyecto planteado abre las puertas a muchas más aplicaciones que se podría realizar con la plataforma Arduino Uno en favor de las personas con discapacidad, este trabajo solo es un precedente no debemos ponerle límites a nuestra imaginación.

5.1. BIBLIOGRAFIA Y ANEXOS

5.1.1. BIBLIOGRAFÍA

Abadín, A. (2012). Comunicación aumentativa y alternativa. Guía de referencia. *cedid.es*, 14.

ARDUINO. (16 de Mayo de 2023). Obtenido de ARDUINO : TECNOLOGIA PARA TODOS: <https://arduinodhtics.weebly.com/iquestqueacute-es.html>

Belloch, C. (2023). TECNOLOGÍAS DE AYUDA SISTEMAS ALTERNATIVOS DE COMUNICACION. 10.

Legislativo, P. (01 de Agosto de 2018). *Ley N° 223/2012. Ley General Para Personas con Discapacidad*. Obtenido de UNESCO: https://siteal.iiep.unesco.org/sites/default/files/sit_accion_files/bo_0268.pdf

Prezi. (25 de Abril de 2023). Obtenido de Uso y aplicaciones con arduino: https://prezi.com/ih2_mpb3-hre/arduino-es-una-plataforma-de-hardware-de-codigo-abierto-bas/

Wikipedia. (09 de Mayo de 2022). Obtenido de Comunicación con personas con discapacidad: https://es.wikipedia.org/wiki/Comunicaci%C3%B3n_con_personas_con_discapacidad

Wikipedia. (06 de Diciembre de 2022). *Tipos de discapacidad*. Obtenido de http://es.wikipedia.org/wiki/Tipos_de_discapacidad

Wikipedia. (09 de Mayo de 2023). Obtenido de Discapacidad Intelectual: [https://es.wikipedia.org/wiki/Discapacidad_intelectual#:~:text=Discapacidad%20intelectual%20leve%20\(CI%2050,personas%20afectadas%20por%20el%20trastorno.](https://es.wikipedia.org/wiki/Discapacidad_intelectual#:~:text=Discapacidad%20intelectual%20leve%20(CI%2050,personas%20afectadas%20por%20el%20trastorno.)

Wikipedia. (29 de Marzo de 2023). Obtenido de Arduino: <https://es.wikipedia.org/wiki/Arduino>

5.1.2. ANEXOS



Figura 52 Dispositivo Electrónico de Comunicación Vista Frontal

Fuente: El autor



Figura 53 Dispositivo Electrónico de Comunicación Vista Lateral Izquierda

Fuente: El autor



Figura 54 Dispositivo Electrónico de Comunicación Vista Trasera

Fuente: El autor



Figura 55 Dispositivo Electrónico de Comunicación Vista Lateral Derecha

Fuente: El autor



Figura 56 Dispositivo Electrónico de Comunicación Vista Panorámica

Fuente: El autor