

INSTITUTO TECNOLÓGICO DE SAN LUIS POTOSÍ



OPCIÓN X

Informe de Residencia Profesional

**“Brailinox, Reingeniería de Prototipo de Tecnología que
facilite el proceso de enseñanza del alfabeto Braille a
personas con ceguera y debilidad visual”**

Para obtener el título de:

INGENIERO EN MECATRÓNICA

Presenta:

Héctor Adrián Méndez López

Soledad de Graciano Sánchez, S.L.P. 6 de Marzo del 2015

Agradecimientos.

Agradezco a mis padres María Isabel López, Juan Jorge Méndez que me brindaron su amor, confianza y las oportunidades necesarias para llegar hasta este punto, agradezco también a mis tías Ma. Guadalupe López y Ma. De Lourdes Méndez que me apoyaron con cariño y consejos durante este proceso, a mis Hermanos Margarita, Cristina y Jorge que con su apoyo incondicional, palabras de aliento y la unión que nos caracteriza lograron inspirarme y darme ánimos para seguir entre el mas adverso de los panoramas; Gracias a mis Sobrinos Natalia, Kevin, Pamela, Diego, Kassandra que con su corta edad me han impulsado a luchar sin descanso y ser ejemplo de vida para ellos.

Agradezco a mis grandes amigas Andrea Betancourt Barrios y Ruth Isabel Rodriguez Vazquez por ser siempre el más fuerte de los apoyos, que, con buenos consejos y los mejores deseos lograron inspirarme en esta increíble experiencia de once meses fuera de casa, agradezco a mis amigos Manuel Ortiz, Miguel Ángel Guerrero, Gildardo Omar, Gerardo Elizalde y Jorge Camacho; gracias a ellos por ser mi segunda familia y apoyarme siempre en mi carrera.

Agradezco también al Ingeniero Manuel Haro Márquez Coordinador del Laboratorio Internacional de Software Libre por darme la oportunidad de continuar con mi investigación en el laboratorio, por ser siempre el mejor de los consejeros y ayudarme en cualquier cosa que se necesitó en la estadía en el laboratorio.

Resumen.

En el siguiente informe se presentan de manera detallada las actividades ejecutadas durante dos procesos escolares, el servicio social en donde se tuvo la concepción de la idea de Braillinox y se llegó hasta una etapa en la cual el proyecto se considerara como un prototipo maduro y tangible, en este documento también se abordará la realización de la residencia profesional con el Proyecto denominado **Braillinox, Reingeniería de Prototipo de Tecnología que Facilite el Proceso de Enseñanza del Alfabeto Braille a Personas con Ceguera y Debilidad Visual**, así como el apoyo en la realización de diferentes actividades dentro del Laboratorio de software libre del COZCYT, específicamente en el área de hardware libre orientado a dispositivos que marcan tendencia en el estudio y filosofía del Hardware y Software libre, como lo son las plataformas de desarrollo Arduino y Raspberry Pi, haciendo énfasis en este último ya que será la parte medular de la residencia profesional, debido a que un dispositivo de este tipo se incluyó desde su inicio hasta su culminación en el proyecto elaborado para residencia profesional.

Índice de Contenido.

CAPÍTULO 1	1
Introducción.	1
Justificación.	2
Objetivos.	3
Objetivo general.....	3
Objetivos específicos.	3
Problemas a Resolver.....	3
Alcances y limitaciones.....	4
Limitaciones.	4
Alcances.....	4
 CAPÍTULO 2 La Empresa.....	6
Caracterización del área en que se participó.	6
Empresa	6
Misión	7
Visión	7
Objetivos	7
Giro	8
Estructura Organizacional	8
 CAPÍTULO 3 Fundamento Teórico	10
Fundamento Teórico.....	10
Brailinox.....	10
Reingeniería del Producto.	10
Sistema Mecatrónico.....	11
Fuente Conmutada.....	12
Actuadores Lineales.....	13
Raspberry Pi Modelo B.	13
Software Libre.....	14

Hardware Libre.....	14
Sistema Operativo.....	15
GPIO.....	16
L293B.....	17
Cloruro Férrico.....	18
Tablilla Fenolica de baquelita.....	19
Buses de comunicación.....	19
Diadema digital.....	20
Software usado para automatizar el proceso.....	20
Putty.....	20
Real VNC.....	21
NOOBS.....	21
QT4.....	23
QTCreator.....	23
TightVNC.....	24
Mplayer.....	24
Pocketsphinx – CMU Sphinx.....	24
Wiringpi.....	25
GNU Bison.....	25
Perl.....	26
 CAPÍTULO 4 El Servicio Social.....	 27
Procedimiento y descripción de las actividades realizadas.....	27
Lo realizado como servicio social.....	28
Cómo se construyo la aplicación.....	52
 CAPÍTULO 5 La Residencia Profesional.....	 59
Lo realizado como Residencia profesional.....	59
 CAPÍTULO 6 Los Resultadis.....	 89
Resultados, planos, gráficas, prototipos y programas.....	89
 CAPÍTULO 7 Las Conclusiones.....	 94

Conclusiones y recomendaciones.....	94
Referencias Electrónicas.....	96
Índice de ecuaciones	101
Índice de Tablas	101
Índice de ilustraciones	101
ANEXOS	103
Anexo 1 – Características de la diadema finalmente seleccionada.	103
Anexo 2 – Cronograma de actividades.	108
Anexo 3 – Programación básica del primer script.....	109

CAPÍTULO 1

Introducción.

Todo comenzó a mediados del año pasado (junio 2013) llegó el momento de realizar el servicio social y de inmediato se pensó en el “Instituto para ciegos y débiles visuales Ezequiel Hernández Romo”. Cuando se llega a la asociación fue clara la propuesta de realizar un proyecto de ingeniería enfocado a las habilidades específicas desarrolladas en la carrera, al principio no se sabía que realizar, pero luego se pensó en construir un sistema que ayudara a un sector vulnerable de la sociedad, un sector donde no hay mucho apoyo, después de unos días de dejar las ideas claras, se propuso un proyecto. Se desarrolló un sistema de enseñanza para ciegos y débiles visuales enfocado al alfabeto Braille.

Cuando llega el momento de iniciar la residencia profesional la fase del proyecto en el servicio social ya era muy avanzada y cuando llegó la hora de buscar residencia profesional, en la ciudad de Zacatecas, en una entrevista en CGTRATEC se mencionó el prototipo, después de conocerlo, el Centro ofreció la oportunidad de seguir trabajando con ellos en la siguiente etapa del dispositivo.

Cuando es el momento para que una persona con ceguera o debilidad visual aprende a comunicarse con el mundo es cuando el problema de la lectura y la escritura hacen un fuerte conflicto, para las personas con ceguera de nacimiento y para los que por alguna circunstancia perdieron su vista en una edad avanzada, se realizó un sistema didáctico para que ellos aprendan a leer el alfabeto Braille de un modo fácil y sencillo, con juegos y retos que ayudarán a que nunca se les confundan las letras en su composición, para esto, el dispositivo está dotado de una minicomputadora que auxilia al sistema con patrones de reconocimiento de voz, procesamiento digital de señales y todo esto estará interactuando directamente con actuadores lineales acondicionados para poder manejar unas salidas de baja potencia.

El dispositivo usará técnicas de enseñanza moderna para interactuar con el individuo y darle la posibilidad de aprender el alfabeto Braille.

Y, se planteara una pregunta que se pretende en la conclusión sea respondida: ¿Es posible incorporar las nuevas tecnologías en hardware y software libre para agilizar el aprendizaje en personas con ceguera o debilidad visual?

Las experiencias contadas aquí son el reflejo de una residencia profesional llena de investigación, cosas nuevas y muchos sacrificios.

Justificación.

Para una persona con facultades normales es fácil poder interactuar con el mundo mediante la escritura y la lectura, pero, para una persona con ceguera o debilidad visual la lectura no es posible por obvias razones, en cuanto a la escritura es muy difícil ya que no existe una retroalimentación de cómo debe hacerse, ni si esta correctamente elaborada, por esto se crea el alfabeto Braille (que por definición es un lenguaje táctil), pero su aprendizaje es complicado y requiere de numerosas horas de práctica, esto no es bueno para niños en edad temprana ya que necesita concentración y mucha paciencia, en adultos que por alguna circunstancia perdieron la visión, también es difícil de aprender, ya que no se tiene la sensibilidad necesaria para identificar los patrones de cajetín (rectángulo generador, con seis puntos en relieve).

Respondiendo a esta necesidad, se cree conveniente implementar nuevas tecnologías a la enseñanza y lograr hacer más rápido el proceso de aprendizaje de este Alfabeto táctil.

Según la *International Agency for the Prevention of Blindness (IAPB)* en el mundo existen alrededor de 285 millones de personas con ceguera y/o problemas visuales severos, situando a casi el 90% de las personas con problemas visuales en países en vías de desarrollo, por esto se considera que el precio del dispositivo debe ser accesible a casi cualquier persona o institución pública.

[1]

Objetivos.

Objetivo general.

Rediseñar un prototipo a bajo costo haciendo uso de Hardware y Software libre, que aproveche la miniaturización de semiconductores a tal grado que en un espacio pequeño se pueda desarrollar una herramienta modular e integrada que ayude a las personas con ceguera y debilidad visual a aprender el alfabeto Braille de una forma divertida, novedosa y muy amigable.

Objetivos específicos.

Reconocer patrones de audio y hacer un modelo autodidacta de enseñanza Braille.

Procesar señales analógicas y convertirlas a digitales con el uso de filtros discretos.

Rediseñar la mecánica del dispositivo para hacerlo confiable, duradero y facilitar su ensamble.

Problemas a Resolver.

El Problema a Resolver es la educación, considerando que el 80% de los niños con problemas como ceguera o debilidad visual, se encuentran en países en vías de desarrollo, en los que típicamente también tienen fuertes problemas en materia de educación, se desarrollo un prototipo económico, amigable e inteligente para intentar solventar este gran problema conjunto: el analfabetismo en los niños con problemas como lo es la ceguera o una discapacidad visual, a continuación se analizarán los problemas acontecidos en la residencia profesional.

El problema más grande en la residencia profesional con el proyecto fue que no se encontraba el modo de procesar audio ya que se seleccionó como minicomputadora una Raspberry pi modelo B y esta no posee entradas analógicas, así que se optó por

una captura digital de audio y para lograr esto se propuso usar un nuevo hardware que acondicionara el audio para que la Raspberry Pi pudiera reconocerlo.

El segundo gran problema a resolver fue el rediseño del dispositivo para un fácil ensamble y una estructura ordenada para lograr un producto duradero.

El tercer problema a resolver fue la grabación correcta del audio para que el amplificador pudiera cubrir el ancho de banda que exige la voz humana para que la reproducción y la calidad del audio fueran óptimas.

El cuarto gran problema consistió en construir o disponer de un amplificador de gran ancho de banda para que el audio fuera lo mejor posible y brindara una experiencia más personal al usuario final.

Alcances y limitaciones.

Limitaciones.

- Requiere un mercado muy especial, como institutos para no videntes y débiles visuales (Aunque con la nueva reforma de inclusión educativa esto ya no será una limitante).
- El análisis de los armónicos para generar los patrones de reconocimiento de voz y el hardware y el software que se va a usar.
- Sólo aplica para personas con ceguera y debilidad visual (no aplica en enseñanza para sordo ciegos).
- No es un dispositivo que por su composición pueda costear una familia.
- Es costoso el desarrollo inicial.
- En México no se tiene una cultura sobre el Hardware el Software Libre.

Alcances.

Según planteado por la diputada Claudia Edith Anaya Mota, la reforma a la Ley de la Inclusión Educativa en la que ella esta inmiscuida, por ser la líder de la Comisión de Personas con Discapacidad; plantea, “las escuelas deberán tener los instrumentos, los equipos y las capacitaciones en las habilidades requeridas para cubrir la

necesidad de un alumno con cualidades especiales que se quiera matricular en una escuela regular”, acatando a estos lineamientos se considera importante el dispositivo para que tome un papel protagónico en la Ley de Inclusión Educativa, asegurando la inclusión al sistema de educación regular de los niños con ceguera y debilidad visual.

[2]

CAPÍTULO 2

Caracterización del área en que se participó.

El proyecto fue realizado para el Centro de Gestión y Transferencia Tecnológica del Conocimiento, ubicado en Lago Yuriria #213, Fracc. Lomas de Lago en la Ciudad de Zacatecas.

Cabe señalar que derivado de la alianza estratégica con el Laboratorio Internacional de Software Libre (LABSOL), en el interior del Consejo Zacatecano de Ciencia y Tecnología e Innovación (COZCyT) de Gobierno del Estado de Zacatecas y ubicado en Paseo de la Juventud, Jesús Reyes Heróles #504, La Encantada, C.P. 98089 en Zacatecas, diversas actividades se coordinaron y se llevaron a cabo en las instalaciones del mismo, lo que permitió que se pudiera contar con todo el acceso a información, asesoría y capacitación en áreas de desarrollo de software libre y utilizar las instalaciones con que se cuentan en el mismo.

Empresa

Centro de Gestión y Transferencia Tecnológica del Conocimiento

El Centro de Gestión y Transferencia Tecnológica del Conocimiento (CGTRATEC), es un espacio independiente y dedicado a fomentar la gestión y transferencia del conocimiento en personas y organizaciones y con un elevado compromiso social y ambiental.

Es una empresa 100% Zacatecana, desarrolladora de plataformas tecnológicas para organizaciones de todos los sectores de la sociedad (público y privado). Minimizando considerablemente el costo de desarrollo al utilizar para todos los procedimientos herramientas distribuidas con código libre, lo que permite optimizar los recursos.

Misión

Diseñar soluciones tecnológicas de vanguardia, que permitan elevar la competitividad de las organizaciones, mediante una estrategia de gestión de conocimiento, lo que permite innovar en los procesos de generación y transferencia de conocimiento con el uso de plataformas tecnológicas que garanticen la protección de los activos intangibles de nuestros clientes.

Visión

Ser uno de los principales centros de investigación y desarrollo tecnológico de Latinoamérica, especializado en soluciones cartográficas y repositorios digitales *open source*.

Objetivos

- Capacitar en todos los niveles, en la utilización y aplicación de Tecnologías de Información, en modalidad a distancia (on-line) y presencial.
- Integrar una comunidad virtual de profesionales sustentada sobre las redes sociales, que permita la interacción y el aprendizaje mutuo de los participantes.
- Desarrollar soluciones geomáticas Open Source, basadas en insumos cartográficos y bases de datos geográficas.
- Implementar repositorios digitales que permitan almacenar y administrar el conocimiento generado dentro de las organizaciones.
- Implementar y capacitar en el manejo de plataformas tecnológicas de aprendizaje a distancia (e-learning).

Giro

Las principales actividades son enfocadas en el impulso al desarrollo científico y tecnológico, relacionados con la investigación básica, investigación aplicada o desarrollo tecnológico en ciencias o ingeniería. Además del desarrollo de proyectos aplicados a resolver la problemática ambiental ocasionada por el cambio climático.

Estructura Organizacional

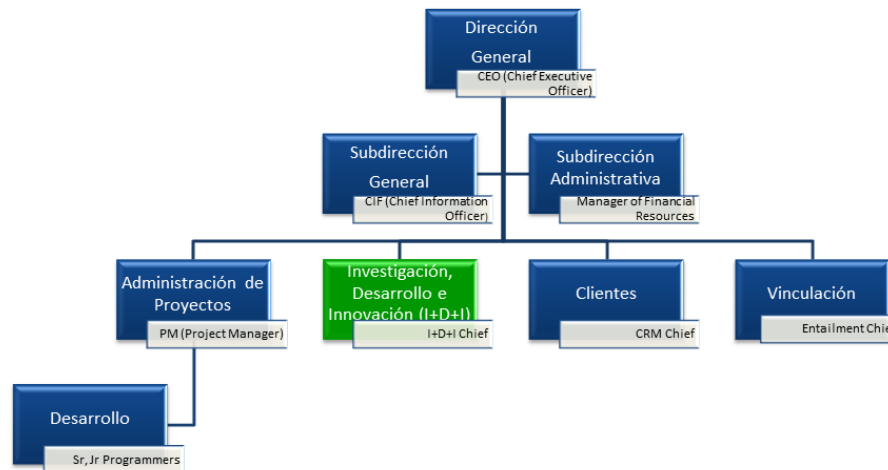


Ilustración 1. Organigrama

En este caso se desarrolló la residencia profesional en el área de proyectos de investigación, desarrollo e innovación, las siguientes son sus funciones:

Funciones del Área de Proyectos de Investigación, Desarrollo e Innovación

- I. Dirigir, analizar y evaluar las alternativas de programas y proyectos.
- II. Dirigir, supervisar y controlar el análisis de la información y justificación de las necesidades que generan un proyecto.
- III. Determinar en su caso la factibilidad de los proyectos que se solicitan.
- IV. Dirigir, coordinar y controlar el desarrollo de los proyectos.
- V. Informar sobre el avance de los proyectos ejecutivos.

VI. Controlar los expedientes de los proyectos y autorizar su distribución a las áreas que lo soliciten.

VII. Mantener una estrecha comunicación con las direcciones del Patronato, para el eficaz desempeño de sus funciones.

El área de especialización del lugar es el software libre, que complementa perfectamente a este proyecto, ya que sólo se usó software libre, pero al no ser especialista resultó una gran sorpresa saber que existen muchas más cosas que las que se conocían, por esto, estar desarrollando hardware en un lugar en el que se desarrolla software aparte de ser muy retador, resulto ser muy estimulante ya que da la oportunidad de fortalecer y aumentar los lenguajes y la lógica de programación.

CAPÍTULO 3

Fundamento Teórico.

A continuación se abordará en forma descriptiva básica cada elemento que conforma el sistema Braillinox y en algunos casos una breve explicación de cómo es importante este elemento para que funcione adecuadamente el sistema.

Como decisión de diseñador se optó por abordar primero la teoría de concepción, pasando por lo físico Electrónico-Mecánico y por ultimo todo el software usado para realizar el desarrollo en sus dos fases, es decir la de servicio social y la de Residencia profesional.

Braillinox.

Se le decidió llamar Braillinox al primer prototipo porque el dispositivo tiene como meta enseñar el alfabeto Braille del cual se desprende “Brail”, y se decidió complementar con el “inox” por dos influencias características en la investigación, primero por Linux que es el núcleo de operación que tiene la minicomputadoras que se va a usar, entonces lo lógico es que se llamara “inux” pero debido a la influencia y el uso e importancia fundamental del puerto GPIO, se decide complementar con “I” referente a “*Input*” y “O” referenciando “*Output*” por lo tanto el nombre clave del proyecto es “Braillinox”.

Reingeniería del Producto.

La reingeniería de producto realiza estudios sobre productos o diseños preexistentes con el fin de optimizar una o varias de sus características.

Como en cualquier problema de optimización, se busca maximizar o minimizar una función atendiendo a una serie de restricciones. En la práctica se busca mejorar prestaciones tales como:

- Reducir el peso.
- Aumentar la vida útil del producto.
- Disminuir el coste de producción.
- Incrementar la resistencia de un determinado componente.

Para ello se juega con otros parámetros, por ejemplo:

- Materiales.
- Geometrías.
- Tecnologías de fabricación.

Se consigue a través de un proceso de cuatro fases:

1. Rediseño conceptual.
2. Ingeniería de detalle.
3. Gestión de la solución propuesta.
4. Ensayos y certificaciones de la solución adoptada.

De este modo la ingeniería se pone al servicio de las necesidades concretas de la industria.

[3]

Sistema Mecatrónico.

Un sistema Mecatrónico es aquel sistema digital que recoge señales, las procesa y emite una respuesta por medio de algún especie de actuador final, generando movimientos o acciones sobre el sistema en el que se va a actuar: Los sistemas mecánicos están integrados con sensores, microprocesadores y controladores. Los robots, las máquinas controladas digitalmente, los vehículos guiados automáticamente, etc. Se deben considerar como sistemas Mecatrónicos.

La Mecatrónica surge de la combinación sinérgica de distintas ramas de la Ingeniería, entre las que destacan: la Mecánica de precisión, la Electrónica, La Informática y los Sistemas de Control. Su principal propósito es el análisis y diseño de productos y de procesos de manufactura automatizados.

Es decir, el proyecto con nombre clave “Brallinox” cumple con todas las especificaciones de diseño, de concepción teórica y práctica para ser un sistema Mecatrónico en toda la extensión de la palabra.

[4]

Fuente Conmutada.

Una fuente conmutada es un dispositivo electrónico que transforma energía eléctrica Mediante transistores en conmutación. Mientras que un regulador de tensión utiliza transistores polarizados en su región activa de amplificación, las fuentes conmutadas utilizan los mismos conmutándolos activamente a altas frecuencias (20-100 kHz típicamente) entre corte (abiertos) y saturación (cerrados).

La forma de onda cuadrada resultante es aplicada a transformadores con núcleo de ferrita (los núcleos de hierro no son adecuados para estas altas frecuencias porque tienen muchas pérdidas debido a corrientes de Foucault y sobre todo por las grandes pérdidas por histéresis; se tiene que recordar que una curva de saturación normal de acero cocido corresponde a un material con característica dura y alta densidad de flujo) para obtener uno o varios voltajes de salida de corriente alterna (CA) que luego son rectificadas (con diodos rápidos) y filtradas (inductores y condensadores) para obtener los voltajes de salida de corriente continua (CC).

Las ventajas de este método incluyen menor tamaño y peso del núcleo, mayor eficiencia y por lo tanto menor calentamiento. Las desventajas comparándolas con fuentes lineales es que son más complejas y generan ruido eléctrico de alta frecuencia que debe ser cuidadosamente minimizado para no causar interferencias a equipos próximos a estas fuentes.

En este caso para alimentar al dispositivo se uso una fuente de alimentación conmutada del tipo alimentación de corriente alterna (CA), y a su salida muestra una salida de corriente continua (CC), los elementos necesarios para una fuente conmutada son: rectificador, conmutador, transformador, rectificador de salida, filtro. Se decidió que era correcto conectarse a la toma de corriente del suministro eléctrico de una casa para facilitar muchos problemas de diseño, entre ellos qué tipo de baterías se usarían y en donde colocarlas, entre otros muchos otros problemas, como el peso o la misma composición de las baterías.

[5]

Actuadores Lineales.

Un actuador lineal eléctrico es un dispositivo que convierte el movimiento de rotación de un motor de corriente continua de baja tensión en movimiento lineal, es decir, los movimientos de empuje y tiro.

[6]



Ilustración 2. Actuador lineal calidad Automotriz

Raspberry Pi Modelo B.

Raspberry Pi es un ordenador de placa reducida de bajo costo, desarrollado en Reino Unido por la Fundación Raspberry Pi, con el objetivo de estimular la enseñanza de ciencias de la computación en las escuelas.

El diseño incluye un SoC (*System-on-a-chip*) *Broadcom BCM2835*, que contienen un procesador central (CPU) *ARM1176JZF-S* a 700 MHz (el *firmware* incluye unos modos “Turbo” para que el usuario pueda hacerle *overclock* de hasta 1 GHz sin perder la garantía), un procesador gráfico (GPU) *VideoCore IV*, y 512 MiB de memoria RAM. El diseño no incluye un disco duro ni unidad de estado sólido, ya que usa una tarjeta SD para el almacenamiento permanente; tampoco incluye fuente de alimentación ni carcasa.

[7]



Ilustración 3. Raspberry Pi modelo B.

Software Libre.

Es la denominación del software que respeta la libertad de todos los usuarios que adquirieron el producto y, por tanto, una vez obtenido el mismo puede ser usado, copiado, estudiado, modificado, y redistribuirlo libremente de varias formas. Según la *Free Software Foundation*, el software libre se refiere a la seguridad de los usuarios para ejecutar, copiar, distribuir, y estudiar el mismo, e incluso modificar el software y distribuirlo modificado.

[8]

Hardware Libre.

Se llama hardware libre o electrónica libre a aquellos dispositivos de hardware cuyas especificaciones y diagramas esquemáticos son de acceso público, ya sea bajo

algún tipo de pago o de forma gratuita. La filosofía del software libre es aplicable a la del hardware libre y por eso forma parte de la cultura libre. Algo que tiene en común el hardware con el software es que ambos corresponden a las partes tangibles de un sistema informático.

[9]

Sistema Operativo.

Raspbian es una distribución del sistema operativo GNU/Linux y por lo tanto, libre, esta basado en *Debian Wheezy (Debian 7.0)* para la placa computadora Raspberry Pi, orientado a la enseñanza de informática y cuyo lanzamiento inicial fue en junio de 2012.

Técnicamente el sistema operativo es una distro no oficial de Debian Wheezy armhf para el procesador (CPU) de Raspberry Pi, con soporte optimizado para cálculos en coma flotante por hardware, lo que permite dar más rendimiento en algunos casos específicos para la ALU debido a la arquitectura ARM. El port fue necesario al no haber versión Debian Wheezy armhf para la CPU ARMv6 que contiene el Raspberry PI.

La distribución usa LXDE como entorno de escritorio y Midori como navegador web. Además contiene herramientas de desarrollo como *IDE* para el lenguaje de programación Python o Scratch, y diferentes ejemplos de juegos usando los módulos Pygame.

Destaca también el menú "raspi-config" que permite configurar el sistema operativo sin tener que modificar archivos de configuración manualmente. Entre sus funciones, permite expandir la partición root para que ocupe toda la tarjeta de memoria, configurar el teclado, aplicar overclock sin comprometer la garantía, etc.

[10]

GPIO.

GPIO (*General Purpose Input/Output*, [Entradas y salidas de Propósito General],) es un pin genérico en un chip, cuyo comportamiento (incluyendo si es un pin de entrada o salida) se puede controlar (programar) por el usuario en tiempo de ejecución.

Los pines GPIO no tienen ningún propósito especial definido, y no se utilizan de forma predeterminada. La idea es que a veces para el diseño de un sistema completo que utiliza el chip podría ser útil contar con un puñado de líneas digitales de control adicionales, y tenerlas a disposición ahorra el tiempo de tener que organizar circuitos adicionales para proporcionarlos.

GPIO se utilizan en:

- chips con pocos pines: IC, SoC, integrados y hardware a la medida, dispositivos lógicos programables (por ejemplo, *FPGAs*)
- chips multifunción: gestores de energía, códecs de audio, tarjetas de vídeo.
- aplicaciones embebidas (por ejemplo, *Arduino*, *BEAGLEBONE*, *Raspberry Pi*) hacen un uso intensivo de *GPIO* para la lectura de varios sensores ambientales (IR, de vídeo, la temperatura, la orientación de 3 ejes, aceleración), y para enviar la salida a motores de corriente continua (con control avanzado y la posibilidad de incorporar *PWM*), audio, LCD o pantallas LED para el estado de un sistema.

GPIO puede incluir:

- Pines GPIO que pueden ser configurados para ser entrada o salida.
- Pines GPIO que pueden ser activados / desactivados.
- valores de entrada se pueden leer (por lo general alto = 1, bajo = 0)
- valores de salida de lectura / escritura.
- valores de entrada que a menudo se pueden utilizar como IRQ (típicamente para los eventos de activación)

Los periféricos GPIO varían muy ampliamente. En algunos casos son, simplemente, un grupo de pines que se puede cambiar en grupo ya sea de entrada o de salida. En otros casos, cada pin puede configurarse con flexibilidad para aceptar o entregar

diferentes voltajes lógicos, y con la potencia de accionamiento configurables. Las tensiones de entrada y salida son típicamente la tensión de alimentación del dispositivo que contiene los GPIO, y pueden ser dañados por mayores tensiones. Algunos GPIO tienen entradas tolerantes de 5 V: incluso cuando el dispositivo tiene una tensión de alimentación baja (por ejemplo, 2 V), el dispositivo puede aceptar 5 V sin daño, en el caso de la Raspberry Pi, esto no ocurre, 3.3V es el voltaje máximo que se le puede dar a una entrada y aplicar más voltaje significa un daño al SoC de la Raspi.

[11]

L293B.

El L293B es un circuito integrado que incorpora dos *drivers* denominados "puentes H". Mediante ellos es posible activar un motor de corriente continua, así como establecer su sentido de giro, esto es especialmente útil para nuestro proyecto ya que se usaran como Driver de estado sólido de potencia entre la Raspberry pi y los actuadores lineales para lograr que los datos procesados y al lanzarlos por la Raspberry pi hagan que los actuadores lineales estén arriba o abajo, según sea el caso para que el ciego o débil visual pueda aprender el Alfabeto de un modo más didáctico y operativo.

Cabe señalar que el puente H considera a 0.8 V. y valores inferiores como un cero lógico y los valores de entre 2.8 V. y 5.5 V. como lógicos 1, esto permitió no tener problemas al momento de incorporar los GPIO de la Raspi, ya que, esta da un uno lógico a los 3.3 V. quiere decir que el Hardware fue perfecto para la aplicación.

[12]

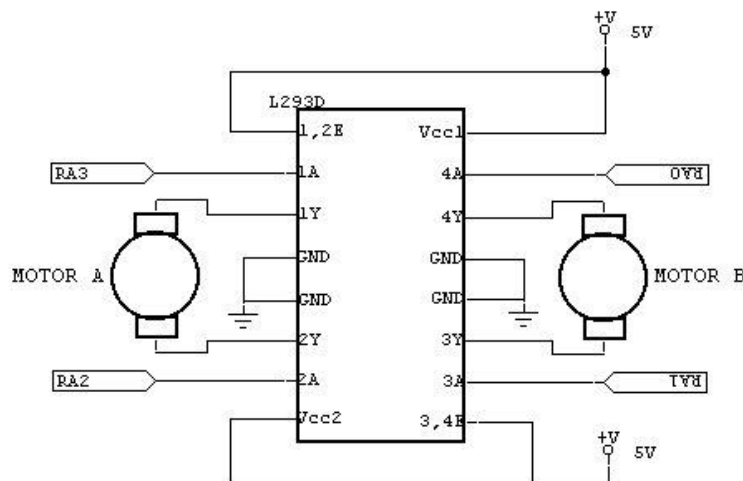
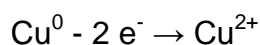
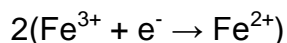
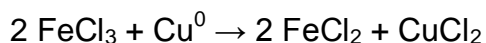


Ilustración 4. Diagrama de conexiones del L293B

Cloruro Férrico.

Es un compuesto químico y una de sus más importantes aplicaciones es en electrónica para producir placas de circuito impreso en bajas cantidades. El cloruro férrico reacciona con el cobre dando cloruro ferroso y cloruro cúprico como lo muestra la siguiente reacción rédox.



Ecuación 1. Reacción rédox del cloruro férrico con el cobre.

El procedimiento general consiste en realizar un dibujo manualmente sobre una plaqueta con una cara de cobre, mediante el uso de un rotulador permanente resistente al ataque de ácidos. También suelen aplicarse diversos métodos más efectivos reemplazando el uso del rotulador, como es el empleo de hojas de acetato (también conocidos como "Transparencias"), imprimiendo sobre este la imagen del circuito prediseñado en un ordenador. Para este método es conveniente utilizar impresoras láser o fotocopadoras, ya que contienen Tóner, y este es resistente al ataque de este ácido. Una vez impreso el circuito en el acetato, se procede a calentar el cobre de la placa (mediante el uso de una plancha o similar) para

posteriormente colocar el acetato sobre ella y adherir el Tóner a él. Para finalizar se introduce la placa en la solución de cloruro férrico.

Cloruro de hierro (III) o cloruro férrico.

Para el procedimiento se utilizan placas con capa de cobre comunes, mientras que para métodos más especializados se utilizan placas fotosensibilizadas.

De preferencia se calienta la solución a 28 °C previamente, o simultáneamente a través de un baño calefactor. La plaqueta debería flotar sobre la solución si ésta se encuentra en buen estado. Es altamente recomendable, aunque no obligatorio, utilizar una fuente de aire en el recipiente donde vaya la solución, de esta manera se oxigena y el ataque del Cloruro Férrico es más efectivo, además de ayudar a eliminar el residuo de cobre de la placa.

[13]

Tablilla Fenolica de baquelita.

En estas tablillas se coloca la circuitería necesaria para hacer funcionar todo el dispositivo, vienen en diferentes tamaños y están recubiertas con cobre de un lado, pero en el anverso están hechas de baquelita, se logra usar el camino de cobre como un camino conductor para hacer las uniones entre los diferentes dispositivos y que todo funcione en armonía, esto se logra sometiendo la tablilla fenolica a un proceso químico que se le denomina “impresión con química de oxidación”.

[14]

Buses de comunicación.

Los buses son pequeños tramos de cable muy flexible, que permiten la conexión entre grupos de nodos que deben unirse entre sí, como por ejemplo, se tuvo que enlazar la Raspberry con la placa en la que se montaron los puentes H que se desarrollo a partir de un semiconductor tipo L293B y los actuadores lineales que fueron los dispositivos de salida de nuestro sistema.

Diadema digital.

La diadema digital, es un hardware comúnmente usado para transmisión de voz por VoIP, ya que la Raspberry Pi no tiene ADC (*analog digital converter*) para entradas analógicas, es decir, no puede procesar voz directamente desde un micrófono, se debe generar un modelo de captura de Audio digital auxiliado del Hardware necesario, en este caso se seleccionó una diadema digital de la marca DYNEX modelo DX-840 que es un producto de alto desempeño pero no de tan alta familia, por eso es perfecta para nuestra aplicación, pues no es tan costosa pero tampoco está hecha de materiales de baja calidad y cuyas características totales se enlistan en los anexos con el título Anexo 1.

Sobre el Hardware usado ya hablamos, es turno de mencionar a todo el software que fue necesario en todas las fases del desarrollo.

Software usado para automatizar el proceso.

Como ya se mencionó el Sistema operativo de nuestra Raspberry Pi es Raspbian basado en Debian 7 y usa un núcleo Linux, esta es sólo la base para que el desarrollo funcione, en realidad, el Software auxiliar y que hace todo posible es un gran listado de programas usados en el desarrollo.

El software usado en la etapa del servicio social fue:

Putty.

PuTTY es un cliente SSH, Telnet, rlogin, y TCP raw con licencia libre. Disponible originalmente sólo para Windows, ahora también está disponible en varias plataformas Unix, y se está desarrollando la versión para Mac OS clásico y Mac OS X. Otras personas han contribuido con versiones no oficiales para otras plataformas, tales como Symbian para teléfonos móviles. Es software beta es escrito y mantenido principalmente por Simon Tatham, open source y licenciado bajo la Licencia MIT.

[15]

Real VNC.

Es una paquetería de software que tiene todos los elementos necesarios para visualizar escritorios remotos con una interfaz gráfica, en este caso usaremos una Edición libre/gratuita-libre, versión *open source* distribuida bajo licencia GPL; se ejecuta en diversos sabores de Unix (*Linux, Solaris, HP-UX, AIX*) y algunas versiones de *Windows* y es proporcionada por el sitio realvnc.com

[16]

NOOBS.

NOOBS se corresponde con la palabra inglesa que significa *novatos* o *principiantes* y que se emplea mucho en el argot informático. Y eso tiene mucho que ver con su acometido, puesto que permite a los *principiantes* que usan la Raspberry Pi probar diversos sistemas operativos de una manera sencilla y rápida, sin tener que volver a formatear la tarjeta SD, descargar nuevas imágenes, instalarlas, etc.

En realidad NOOBS son las siglas de *New Out Of Box Software* y se trata de un software que empaqueta diversos sistemas operativos oficiales para Raspberry Pi. Esto te facilita su puesta en marcha desde una misma SD sin realizar cambios drásticos. Entre los sistemas operativos que nos brinda NOOBS están: *Raspbian, RaspBMC, Pidora, OpenELEC, Arch Linux y RISC OS*.

Raspbian es la opción recomendada, puesto que es una distribución Linux basada en Debian Wheezy especialmente pensada para la arquitectura ARM, con un compilador cruzado desarrollado por el equipo de trabajo de Linaro y con *Pi Store* para descargar multitud de aplicaciones para la Raspberry Pi. Al estar basada en Debian, su sistema de paquetería es DEB, obviamente.

- Si Usted no está acostumbrado a usar este tipo de paquetes y usted viene de una distribución basada en RPM, quizás te interese más probar Pidora. Se trata de una versión de Fedora para microprocesadores ARM. Así que puede ser una buena alternativa a la anterior, aunque no cuenta con tanto apoyo.

- RaspBMC es una distribución que se apoya en dos pilares básicos, Debian y XBMC. Así podrás tener un Media Center en tu TV o transformar tu TV en una flamante *smartTV* con acceso a contenidos multimedia, juegos, aplicaciones e Internet.
 - La otra opción es elegir openELEC, muy parecida a RaspBMC. Se trata de otra distribución especialmente pensada para el contenido multimedia. Ya explicamos en esta web como crear un centro multimedia con XBMC, pero con estas distros es mucho más sencillo.
 - Arch Linux es una conocida distribución Linux que también puede instalarse bajo plataformas ARM. Cuenta con mucho apoyo, aunque no esté tan optimizada para el hardware de la Raspberry Pi. Es ideal si en tu equipo portátil o de sobremesa ya trabajas con Arch. Se apoya sobre el gestor de paquetes Pacman (creado por Judd Vinet para gestionar los paquetes binarios) y su sistema de actualización es *Rolling Release Model* (con actualizaciones periódicas, en vez de cambios bruscos).
 - Por último se tiene RISC OS, menos recomendable que el resto. Este sistema fue creado por *Acorn Computer*, la misma compañía que desarrolló los ARM. Por eso está especialmente creado y optimizado para esta arquitectura. Es bastante rápido, pero por el contrario no cuenta con tanto software y apoyo.
- Todos esos sistemas son los oficiales de Raspberry Pi, pero para qué elegir si puedes tenerlos todos con NOOBS... y solo con pulsar la tecla Shift (mayúsculas) durante el arranque, te permitirá elegir el sistema operativo.

Requisitos mínimos para instalar NOOBS

- Raspberry Pi Modelo B (con sus accesorios básicos necesarios).
- Tarjeta de memoria SD con al menos 4GB de espacio libre (más de 8GB recomendable).
- Un PC o Mac con lector de tarjetas desde donde instalar los sistemas en la tarjeta.
- La imagen para instalación offline de NOOBS.

[17]

QT4.

Qt es una amplia plataforma de desarrollo que incluye clases, librerías y herramientas para la producción de aplicaciones de interfaz gráfica en C++ que pueden operar en varias plataformas. Con Qt se pueden desarrollar ricas aplicaciones gráficas, incluye soporte de nuevas tecnologías como *OpenGL*, *XML*, Bases de Datos, programación para redes, internacionalización y mucho más.

Qt dispone de una amplia gama de herramientas que facilitan la creación de formularios, botones y ventanas de dialogo con el uso del ratón. Las aplicaciones creadas con Qt son muy elegantes, se ven y se operan mejor que las aplicaciones nativas.

Qt dispone de tres grandes ventajas ante las librerías de ventanas rivales:

1. Qt es completamente gratuito para aplicaciones de código abierto
2. Las herramientas, librerías y clases están disponibles para casi todas las plataformas Unix y sus derivados (como Linux, MacOS X, Solaris, etc) como también para la familia Windows, por lo que una aplicación puede ser compilada y utilizada en cualquier plataforma sin necesidad de cambiar el código y la aplicación se verá y actuará mejor que una aplicación nativa.
3. Qt tiene una extensa librería con clases y herramientas para la creación de ricas aplicaciones. Estas librerías y clases están bien documentadas, son muy fáciles de usar y tienen una gran herencia de programación orientada a objetos lo cual hace de la programación de interfaces gráficas una aventura placentera.

[18]

QTCreator.

Qt Creator es un *IDE* creado por *Trolltech* para el desarrollo de aplicaciones con las bibliotecas Qt, requiriendo su versión 4.x. Se dispone de una versión en

GNU/Linux 2.6.x, para versiones de 32 y 64 bits con Qt 4.x instalado. Además hay una versión para Linux con gcc 3.3.

[19]

TightVNC.

TightVNC es una paquetería de software libre encargada de controlar remotamente una maquina, es decir tu puedes controlar una computadora diferente a la tuya usando tu propio teclado y *mouse* auxiliado de este software, es multiplataforma, es decir, sirve en Linux, Windows y Mac.

[20]

Mplayer.

MPlayer es un reproductor multimedia multiplataforma liberado bajo la licencia GPL. Reproduce la mayoría de los archivos MPEG, VOB, AVI, OGG/OGM, MKV, VIVO, ASF/WMA/WMV, QT/MOV/MP4, FLI, RM, NuppelVideo, YUV4MPEG, FILM, RoQ, PVA, soportados por algunos códecs nativos, XAnim, y DLL's Win32. Además puede reproducir VideoCD, SVCD, DVD, 3ivx y DivX/Xvid3/4/5, es también un útil reproductor multimedia por terminal que usaremos en nuestro dispositivo.

[21]

El software usado en la etapa de residencia fue:

Pocketsphinx – CMU Sphinx.

CMU Sphinx (acortado como Sphinx), es el término general para describir un grupo de sistemas de reconocimiento de voz desarrollado en la Universidad de Carnegie Mellon. Incluye una serie de programas para reconocimiento de voz (Sphinx 2 - 4) y un entrenador modelo acústico (SphinxTrain).

En el año 2000, el grupo de Sphinx se comprometió a desarrollar varios componentes para reconocimiento de voz, incluyendo Sphinx 2 y más tarde Sphinx 3 (en 2001). Los decodificadores de voz vienen con modelos acústicos y aplicaciones de ejemplo. Los recursos disponibles incluyen además el software para el entrenamiento de modelos acústicos, la compilación de un modelo de lenguaje y un diccionario de pronunciación en dominio público llamado *cmudict*.

PocketSphinx es una versión de la Sphinx que se pueden utilizar en sistemas embebidos (por ejemplo, basado en un procesador ARM). PocketSphinx está siendo evaluado para desarrollar e incorporar características como la aritmética de coma fija y algoritmos eficientes para el cálculo de modelos mezclados. Puede ser utilizado en muchos equipos portátiles y también en teléfonos móviles.

[22]

Wiringpi.

WiringPi es una librería de estilo Arduino escrita en C y desarrollada por Gordon Henderson, esta librería da un fácil acceso a los pines de propósito general del Raspberry Pi, la librería tiene soporte para el puerto *uart*, *spi*, *i2c*, *pwm*.

[23]

GNU Bison.

GNU bison es un programa generador de analizadores sintácticos de propósito general perteneciente al proyecto GNU disponible para prácticamente todos los sistemas operativos, se usa normalmente acompañado de flex aunque los analizadores lexicos se pueden también obtener de otras formas.

Bison convierte la descripción formal de un lenguaje, escrita como una gramática libre de contexto LALR, en un programa en C, C++, o Java que realiza análisis sintáctico. Es utilizado para crear analizadores para muchos lenguajes,

desde simples calculadoras hasta lenguajes complejos. Para utilizar Bison, es necesaria experiencia con la sintaxis usada para describir gramáticas.

GNU Bison tiene compatibilidad con Yacc: todas las gramáticas bien escritas para Yacc, funcionan en Bison sin necesidad de ser modificadas. Cualquier persona que esté familiarizada con Yacc podría utilizar Bison sin problemas.

Bison fue escrito en un principio por Robert Corbett; Richard Stallman lo hizo compatible con Yacc y Wilfred Hansen de la *Carnegie Mellon University* añadió soporte para literales multicaracter y otras características.

[24]

Perl.

Perl es un lenguaje de programación diseñado por Larry Wall en 1987. Perl toma características del lenguaje C, del lenguaje interpretado bourne Shell (sh), AWK, sed, Lisp y, en un grado inferior, de muchos otros lenguajes de programación.

Estructuralmente, Perl está basado en un estilo de bloques como los del C o AWK, y fue ampliamente adoptado por su destreza en el procesado de texto y no tener ninguna de las limitaciones de los otros lenguajes de script.

[25]

CAPÍTULO 4

Procedimiento y descripción de las actividades realizadas.

Lista de materiales para todo lo que se uso en el servicio social no tiene un valor de mayoreo, todos los precios son unitarios, y la investigación hasta este punto fue desarrollada con recursos propios.

Lista de materiales.

Cantidad	Concepto	Precio total
1	Starter kit Raspberry pi modelo B.	57.2 dls
1	Cable micro USB.	3dls
1	Fuente USB 1 Amper.	7dls
3	Puentes H L293B.	60 \$
8	Metros de cable telefónico.	24 \$
26	Clemas.	52\$
1	Tablilla fenólica de 7.5x10 cm.	12\$
1	Fuente conmutada de 12 V. y 5 V. a 2 A.	9dls
1	Placa de acrílico .22x1.44 mts.	217\$
10	Escuadras de 2".	20\$
68	Tornillos 3/8 con tuerca.	52\$
6	Actuadores lineales automotrices.	150\$
1	Bus 12 líneas.	8\$
1	Cable HDMI.	150\$
4	Bases de goma para el sistema.	12\$
6	Esferas del cajetín.	20\$
3	Bases para CI.	30\$
2	Metro de soldadura 60-40 y pasta.	20\$
		1882\$

Tabla 1. Lista de productos con precio usada en Brailinox en fase de servicio social.

Lo realizado como servicio social.

Lo que se realizó en el servicio social es lo que se podría considerar un avance muy significativo, pero en materiales, estructuras, y en conceptos clave, a continuación se muestra un desarrollo de 7 meses como primera etapa de desarrollo. Lo primero en lo que se pensó fue en la automatización propia del mismo Brailin, pero no se consideró ni ético ni original, así que se optó por desarrollar un dispositivo único en su tipo, y con alcances mucho mayores de lo que Brailin había alcanzado.

Entonces se concibió la idea mientras se reparaba la camioneta, el seguro eléctrico no funcionaba y cuando se estaba cambiando, bang, la iluminación llegó, ese era el dispositivo que se necesitaba para darle vida al producto, un actuador lineal de bajo costo, compacto y con control de entrada-salida con un cambio de polaridad al energizarlo, entonces se empezó a juntar dinero poco a poco hasta juntar seis actuadores de similares características como los de la siguiente imagen.



Ilustración 5. Actuador lineal principal.

Entonces se descubrió que el desarrollo requería una elaboración de hardware sin precedentes, implicaba la construcción de dos fuentes estables, unidas por el método de tierras; también implicaba construir un buffer de estado sólido, de respuesta

rápida y que funcionara como un puente H completo, entonces de inmediato se pensó en el usual puente H integrado L298N, pero resultó no ser confiable por su costo y el espacio que ocupaba, luego se pensó en el puente H single drive basado en transistores con arreglo Darlington, pero resultó hacer difícil la integración por el gran número de semiconductores que se usarían, entonces se recordó un proyecto antiguo en el que se usó un puente *full dúplex*, de baja integración, fácil de usar y que considera un *TrueFlag* desde 2.8 Volts (especialmente útil para nuestro dispositivo ya que los *GPIO* de la *Raspberry Pi* dan salidas digitales lógicas que se consideran un uno lógico a los 3.3V.), el reto ya no era el dispositivo, el reto ahora consistió en diseñar una tablilla capaz de incluir todo en un solo bloque, de modo ordenado y sin complicaciones, el resultado fue el siguiente:

Como software para diseñar esta tablilla se consideró usar “*Electronical Workbench Multisim*”, pero al notarse que no todos los semiconductores estaban disponibles pues se buscó otra alternativa, fue ahí cuando se pensó de inmediato en un producto de la reconocida desarrolladora de software “*Labcenter Electronics*”, el producto que se decidió usar fue la plataforma para diseño y simulación electrónica “*Proteus*” y el subproducto “*ISIS*” para el diseño del circuito y asegurarnos que funcionaría perfectamente y para diseñar el circuito impreso se usó el asistente “*Ares*” que de inmediato proporcionó resultados positivos, que a continuación se describen.

Primero se diseñó en *Ares* un circuito impreso basado en tres L293B, esto implicó darle al dispositivo las capacidades para accionar tres puentes H *full duplex* en una sola tarjeta, con esto se dotó de recursos a nuestro sistema para mover a voluntad seis actuadores lineales y organizar las entradas y las salidas provenientes de la *Raspberry pi* y las salientes del bus de salida a los actuadores, incluyendo un bloque único para los voltajes en la placa como se muestra en la figura 6.

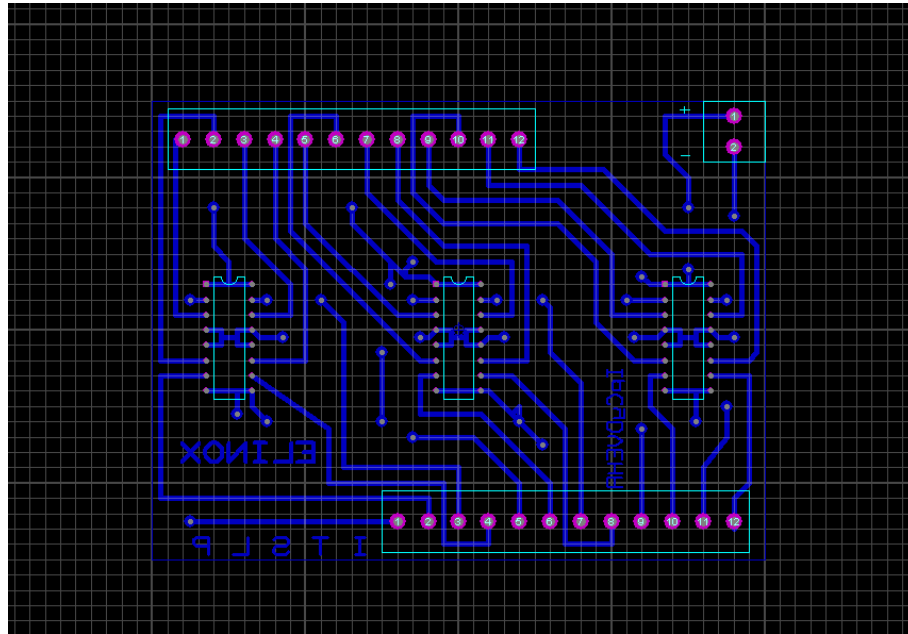


Ilustración 6. Diseño en Ares de la placa unica basada en L293b.

Al finalizar el diseño se generó un Render para tener una referencia de cómo iba a quedar la tarjeta, la realidad es muy diferente a lo ideal generado por el render, pero nos da una referencia tangible de cómo debería lucir al terminar el proceso, el Render es ilustrado en la figura 7.

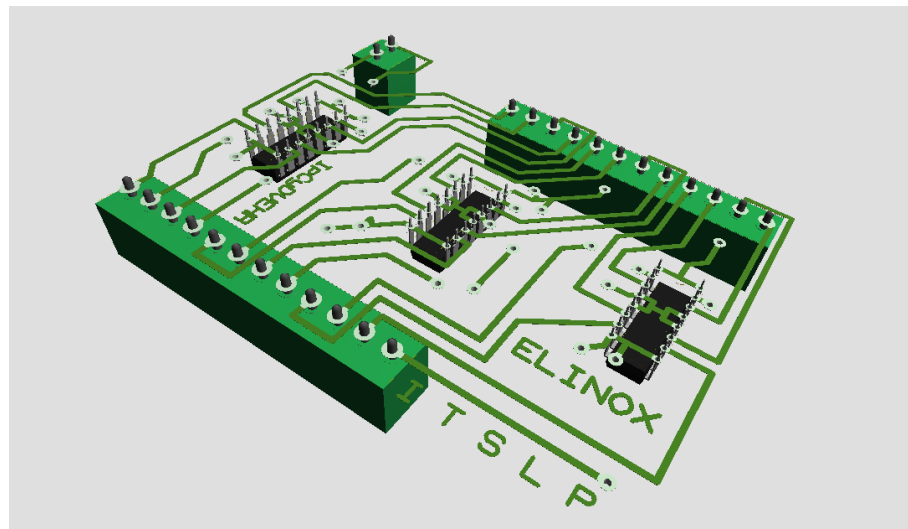


Ilustración 7. Render hecho por el asistente Ares de nuestro circuito impreso.

Cuando llego el momento de hacer la tabllilla real primero se imprimió el PDF que genera el asistente para hacer circuitos impresos Ares, el asistente generó el PDF

con la ilustración 8 que luego se plancho con un lienzo húmedo sobre la hoja impresa, hasta que la tinta quedo plasmada sobre la superficie de cobre.

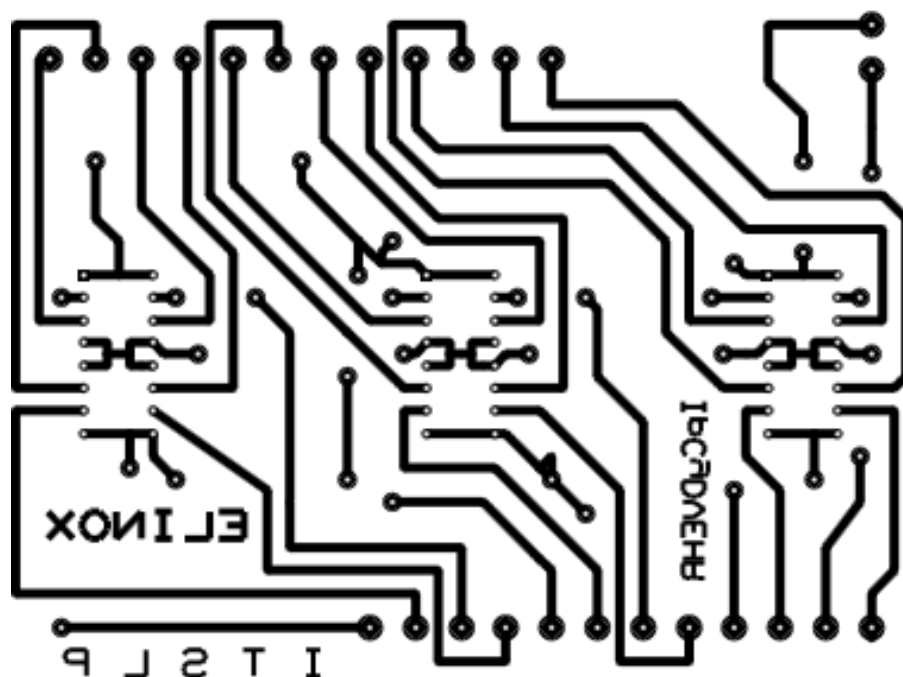


Ilustración 8. Circuito a planchar en la placa de baquelita.

Luego con la tinta impresa sobre el cobre se puso la placa de baquelita en un recipiente con cloruro férrico.

Para optimizar los resultados, la investigación arrojó que el cloruro férrico debe ser suficiente y debe de hacer que la placa flote en el recipiente, la base química no debe estar fría, preferentemente debe ser templada a 32° centígrados, y de ser posible debe de hacerse el proceso en un lugar ventilado para mayor seguridad y con la presencia del sol, por ultimo de ser posible se le debe de inyectar un poco de aire para que el proceso de oxidación fuera más rápido para lograr un resultado optimo, aún así el resultado varia un poco de lo ideal a lo real, en la ilustración 9 se muestra la placa terminada.

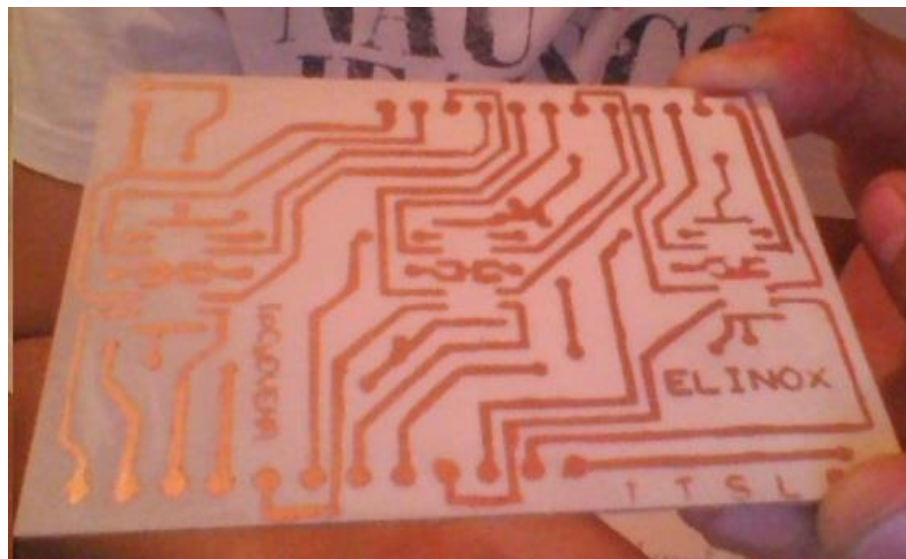


Ilustración 9. Tablilla fenolica ya impresa, limpia, lista para perforar y montar componentes.

Más tarde se abordara el uso fundamental de este desarrollo en el proyecto y porque su diseño es una implementación hecha a medida para cubrir las necesidades del sistema.

Ya con el buffer de estado sólido construido y funcionando se descubrió que se necesitaría más que una fuente de celular para darle vida al sistema, al principio, en origen, el sistema Braillinox, tendría sólo una fuente conmutada, por mucho, esta es la decisión de diseñador más fuerte que se tuvo que tomar, al final se optó por una fuente conmutada para los actuadores y el puente H y una fuente separada, clase D para la Raspberry Pi, y ambas fuentes interactúan en niveles de voltajes similares gracias al arreglo entre sus tierras, una fuente es una USB de 1A. la otra fuente es conmutada con salida dual de 5 y 12 Volts y con una corriente máxima de 2A.

El desarrollo de la fuente se basó en una aplicación vista en un portal de Internet, el portal es un renombrado sitio web sobre tendencias en la tecnología enfocadas a la electrónica de estado sólido, en el artículo se mencionó una nueva aplicación para el diseño de sistemas conmutados en Internet llamado “*Power Supply WebDesigner (PSWD)*” de la prestigiosa firma *Fairchild Semiconductor*, la aplicación es una herramienta que nos permite diseñar una fuente conmutada partiendo de una necesidad, es decir, se deben tener las características de la fuente que se pretende

diseñar e introducir los parámetros de lo que se necesita y el sistema empieza a formular preguntas sobre cómo debe hacerlo, entre ellas, los voltajes de entrada y salida, potencia máxima, cual puede ser el sobreimpulso de la fuente, cuánto tiempo puede durar el transitorio, cuanta variación puede permitir el sistema, que porcentaje de eficiencia puede tener, que dimensiones son aceptables, cuanta es la variación de rizado permisible, que semiconductores es posible incorporar, qué inductores se pueden incorporar al dispositivo, que capacitores se usaran para filtrar el rizado y amortiguar la señal y que frecuencia de conmutación debería tener la fuente, entre muchas otras características, PSWD permite tener control sobre lo que se diseñe.

[26][27]

Al final el sistema genera como salida un diagrama y una lista de materiales para construir la fuente que se diseñó, la aplicación posee también un asistente que sugiere fuentes conmutadas de diferentes fabricantes y que empatan con las características de lo que se acaba de diseñar, no es tan general pero sugiere fuentes que cumplan con las exigencias del diseño que se plantean en ese momento, en este apartado por fortuna se encontró una fuente de uso general para el sistema que es comúnmente usada para reproductores de vídeo, pero que para nuestro caso resulto ser muy económica y que se adquirió con mucha facilidad porque el número de modelo es muy usual.

La fuente a usar fue una E302002 fabricada por *RJ semiconductor* que es una empresa que le fabrica bloques de propósito general a *Daewoo*, en la imagen 10 se muestra la fuente.



Ilustración 10. Vista superior de una fuente similar a la usada en Braillinox.

Según el sistema esta es la fuente que más empata con lo que se diseñó, el diagrama de lo que se diseñó se muestra en la figura 11 y los materiales usados se listan en la tabla 2.

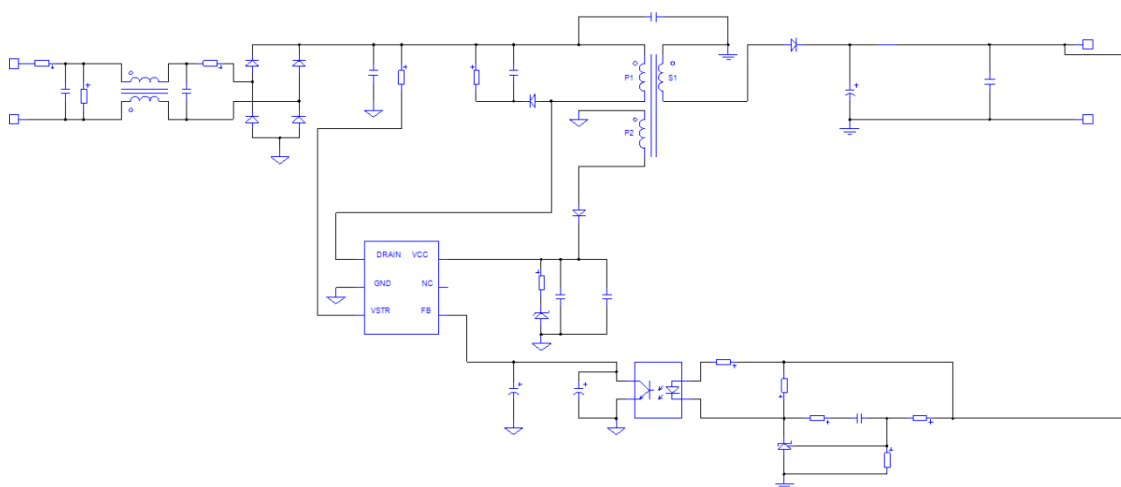


Ilustración 11. Diagrama de la fuente conmutada diseñada en PSWD.

Ref	Qty	Part Number	Manufacturer	Description
U1	1	FSL126MRT	Fairchild Semiconductor	Green-Mode Fairchild Power Switch
Cb	1	C0603C151J4GACTU	Kemet	Cap Ceramic 150pF 16VDC C0G 5% SMD 0603 Paper T/R
Cdc	1	EET-ED2W560BA	Panasonic	Cap Aluminum 56uF 450VDC 20% 22 X 25mm Snap-In 10m
Cf	1	12065C154JAT2A	AVX	Cap Ceramic 0.15uF 50VDC X7R 5% SMD 1206 Paper T/R
Cin1	1	B32613A3224J008	Epcos	CAP .22UF 250V METAL POLYPRO
Cin2	1	B32652A3154J	Epcos	CAP .15UF 250V METAL POLYPRO
Co1	3	94SH227X0010FBP	Vishay	Aluminum Organic Polymer Capacitors 220uf 10volts 20%
Co1_3	1	0603ZC104JAT2A	AVX	Cap Ceramic 0.1uF 10VDC X7R 5% SMD 0603 Paper T/R
Copto	1	12063G104ZAT2A	AVX	Multilayer Ceramic Capacitors (MLCC) - SMD/SMT 25V 0.1uF
Cps	1	564R30GAD47	Vishay	Cap Ceramic 0.0047uF 3000VDC Z5U 20% 15.7 X 5mm RDL 9.5mm Bulk
Csn	1	BFC241912702	Vishay	CAP FILM MKP .0027UF 400VDC 2%
Cvcc_bk	1	EEU-EB1H470	Panasonic	Cap Aluminum 47uF 50VDC 20% 6.3 X 15mm RDL 2.5mm 87mA 7000 hr Bulk
Cvcc_hf	1	08055C224KAT2A	AVX	Cap Ceramic 0.22uF 50VDC X7R 10% SMD 0805 Embossed T/R
D1	1	MBR1060	Fairchild Semiconductor	DIODE SCHOTTKY 10A 60V TO-220AC
Din1	1	MB6S	Fairchild Semiconductor	Diode Rectifier Bridge Single 600V 0.5A 4-Pin SOIC N T/R
Din2	1	MB6S	Fairchild Semiconductor	Diode Rectifier Bridge Single 600V 0.5A 4-Pin SOIC N T/R
Din3	1	MB6S	Fairchild Semiconductor	Diode Rectifier Bridge Single 600V 0.5A 4-Pin SOIC N T/R
Din4	1	MB6S	Fairchild Semiconductor	Diode Rectifier Bridge Single 600V 0.5A 4-Pin SOIC N T/R
Dsn	1	EGP10K	Fairchild Semiconductor	Diode Switching 800V 1A 2-Pin DO-41 T/R
Dvcc	1	BAY80	Fairchild Semiconductor	Rectifiers DO-35 GP SWITCHING
Dzvcc	1	MMSZ5250B	Fairchild Semiconductor	DIODE ZENER 20V 500MW SOD-123
Fuse	1			Resistor 1Ω
Lin	1			Inductor 20mH
NTC	1			Resistor 1Ω
Rbias	1	CRCW06031K00FKEB	Vishay	Thick Film Resistors - SMD 1/10watt 1Kohms 1%
Rd	1	RC0603FR-071K5L	Yageo	Thick Film Resistors - SMD 1.5K OHM 1%
Rdvcc	1	KTR10EZPF51R0	Rohm Semiconductor	RES 51.0 OHM 1/8W 1% 0805 SMD
Rin1	1	VR37000001504FA100	Vishay	Res Metal Glaze 1.5M Ohm 1% 1/2W ±200ppm/°C Conformal AXL Thru-Hole Ammo Pack
Rs_bot	1	CRCW06032K55FKEA	Vishay	RES 2.55K OHM 1/10W 1% 0603 SMD
Rsn	1	MBB02070C6192FC100	Vishay	Res Thin Film 61.9K Ohm 1% 3/5W ±50ppm/°C Conformal AXL Thru-Hole Ammo Pack
Rs_top	1	CRCW06032K55FKEA	Vishay	RES 2.55K OHM 1/10W 1% 0603 SMD
Rstr	1	CF 1/2 75K 5% R	Stackpole Electronics	RES 75K OHM 1/2W 5% CARBON FILM
U2	1	TL431ACL	Fairchild Semiconductor	Voltage & Current Programmable shunt, 2.5V
U3	1	FODM8801A	Fairchild Semiconductor	Optocoupler, transistor out, 3750Vrms Iso, CTR 100% @ 1mA

Tabla 2. lista de materiales usados en la fuente diseñada.

Con los elementos centrales contruidos se procede a ensamblar la estructura, la idea de los actuadores flotados fue una idea de concepción adoptada por el montaje Aéreo de semiconductores; la idea consiste en tener anclados a una base sólida el menor número de elementos y que la rigidez del sistema será proporcionada por los elementos más pequeños que integran al sistema y que al estar unidos entre sí, proporcionen un sistema estable para trabajar, mecánicamente hablando.

Cuando llegó el momento de realizar la cubierta exterior fue la oportunidad más clara de reflejar las teorías de diseño que se conocieron en la carrera, una de ellas es la teoría Europea de fabricación, al realizar un sistema modular es fácil agregar o quitar ciertas características del dispositivo sin comprometer críticamente la integridad del sistema, asimismo, si alguna parte de Braillinox llegará a fallar, sería en teoría fácil reemplazar una pieza o un panel ya que todo está unido al sistema y todo es útil al mismo, pero al construir pequeñas piezas que cumplan por sí mismas funciones específicas en el sistema, al presentarse alguna falla el error arrojado por el sistema permite determinar fácilmente que pieza es la que falla y es muy fácil sustituir por una nueva, de bajo costo y de fácil colocación, es decir, el sistema por sí mismo en

su concepción estructural permite que se cumpla uno de los objetivos que es hacer un sistema con una vida útil larga y que sea muy estable estructuralmente.

Primero se pensó en un bloque estructuralmente sólido, después, al adoptarse con tanto compromiso las teorías europeas modulares, se decidió hacer un bloque flotado con paneles fijos pero separados alrededor de 1.0 a 1.5 cm. de la base para simbolizar que el prototipo posee una estructura modular.

El sistema fue adaptando a la estructura a sus necesidades, hasta llegar al punto en el que al final la estructura en su concepción misma definió las dimensiones en base a la necesidad que se tuvo por espacio y limitaciones de software, una vez más nos damos cuenta de que lo real difiere mucho a lo ideal.

Primero se diseñó la estructura básica del sistema partiendo de las medidas naturales del sistema Braille, pero por cuestiones prácticas se tuvieron que agrandar, siguiendo la misma tendencia ordenada se centro un punto base y desde ahí se desprendió una matriz rectangular como se muestra en la figura 12 a continuación.

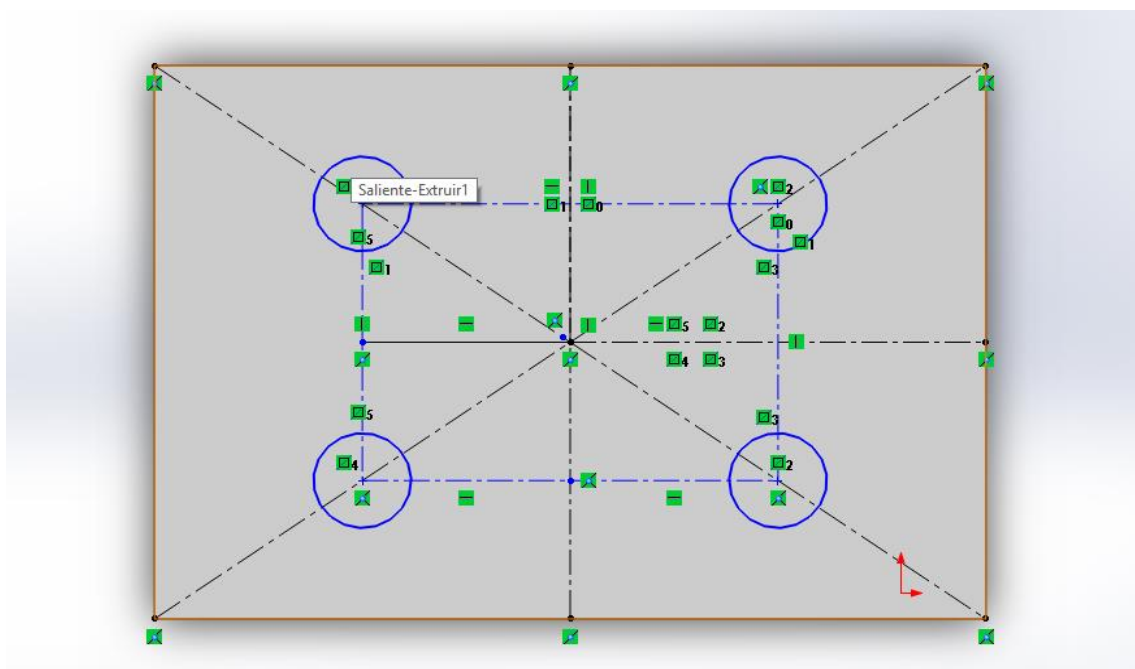


Ilustración 12. Matriz cuadrada para el rectángulo generador del alfabeto Braille.

El Software que se utilizo para generar todos los diseños mecánicos de la estructura fue SolidWorks 2013, desarrollado por la marca "Dassault Systèmes".

Desde la idea original el material ya estaba claro, se uso un acrílico de 3 mm. gracias a esto la placa desde su diseño en el software se considero de 3 mm., al eliminar el material para que los actuadores pudieran salir el resultado ya con la matriz trazada en lo idea fue optimo, tal como se muestra en la ilustración 13.

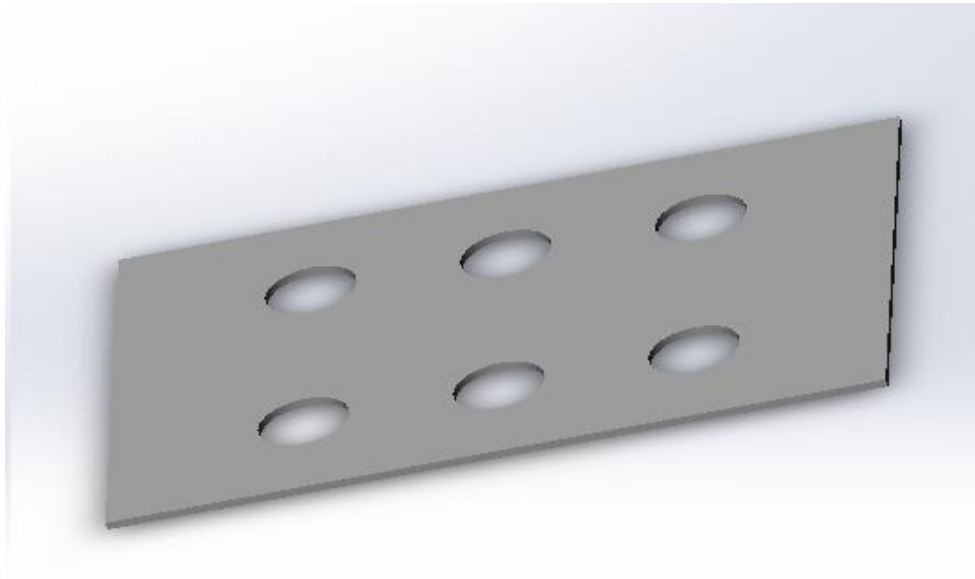


Ilustración 13. Parte superior del sistema ya terminada.

Al final considerando las medidas de 25.5 cm de largo por 17.5 cm de ancho y 18 cm de alto se logro generar una caja que estructuralmente nos de ideas de cómo debe quedar en aspectos finales generales la caja Brailinox como se muestra en la imagen 14.

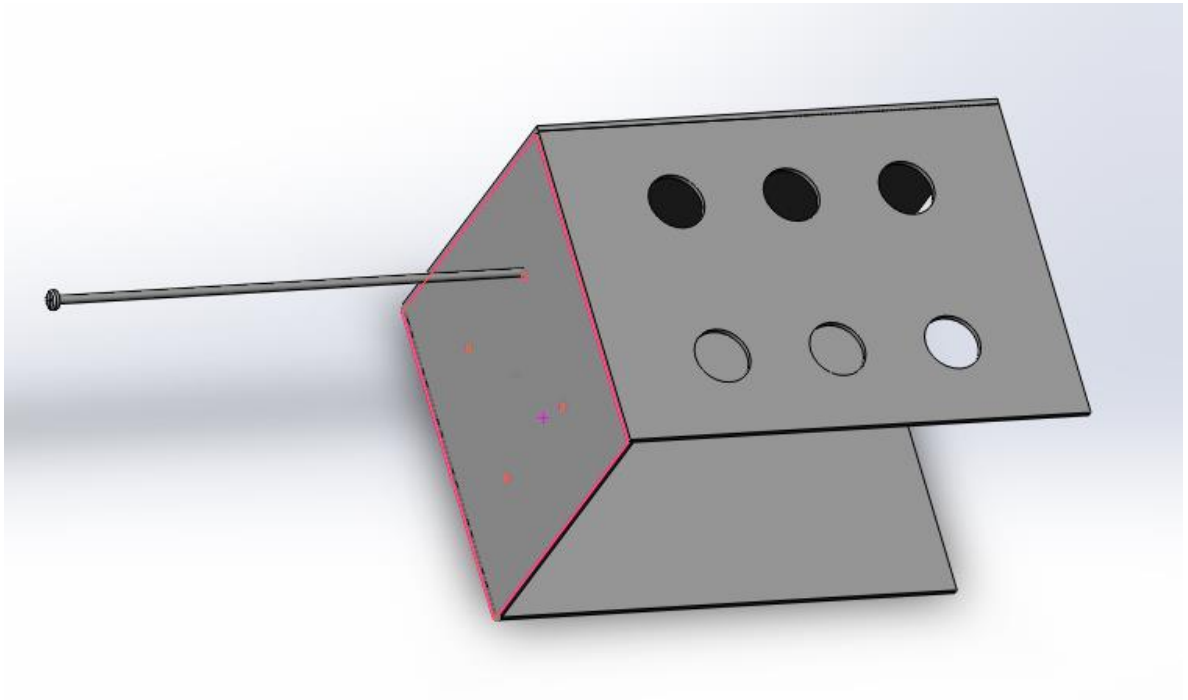


Ilustración 14. Carcasa externa del sistema Brailinox.

Ya con la estructura hecha en el simulador y los paneles medidos se procedió a cortar el acrílico, obviamente no se tuvieron problemas mayores, las dimensiones finales del producto varían un poco por la complejidad de los paneles pero aquí se muestran.

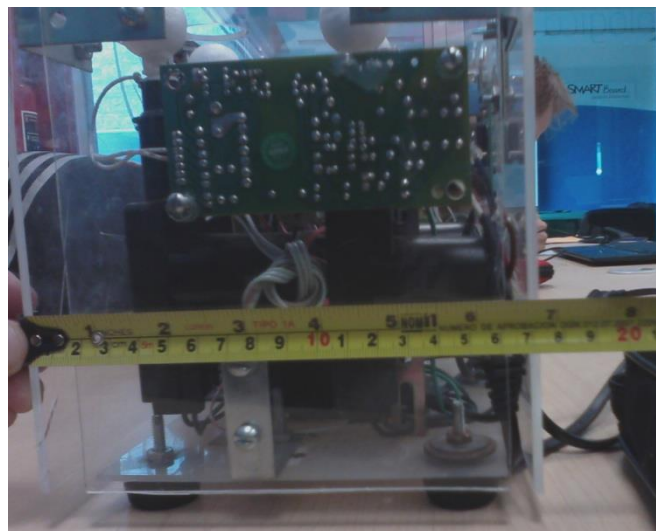


Ilustración 15. Medida final ancho 17.5 cm.



Ilustración 16. Medida final alto 18 cm.

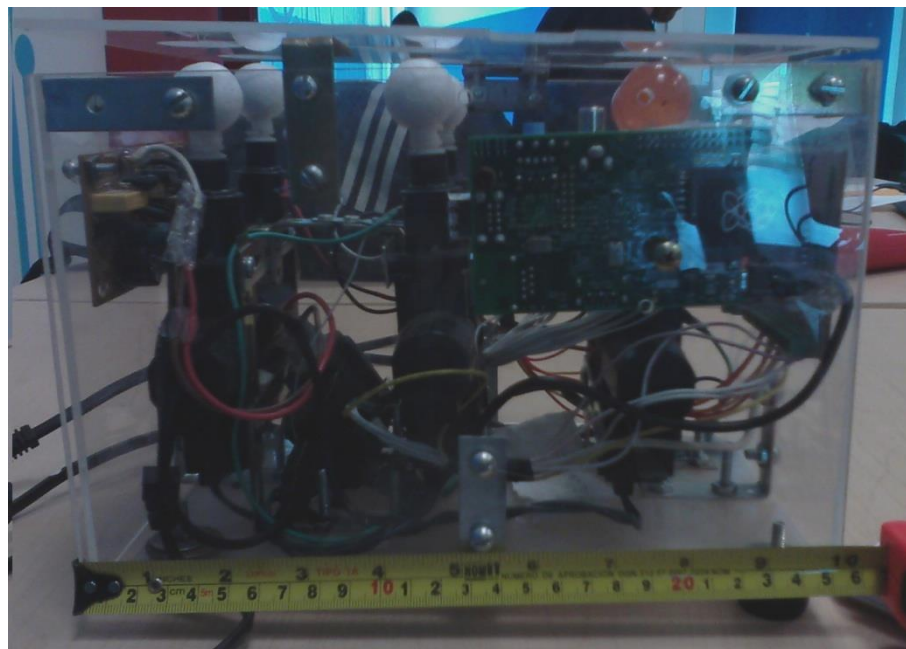


Ilustración 17. Medida final largo 25.5 cm.

Ya con la estructura definida se prosiguió a ensamblar los componentes unos con otros para lograr tener nuestro sistema armado, con palabras es un poco difícil representar ideas de conexiones anidadas y compartidas, así que se elaboró un pequeño esquema simbólico que a continuación se muestra en la figura 18.

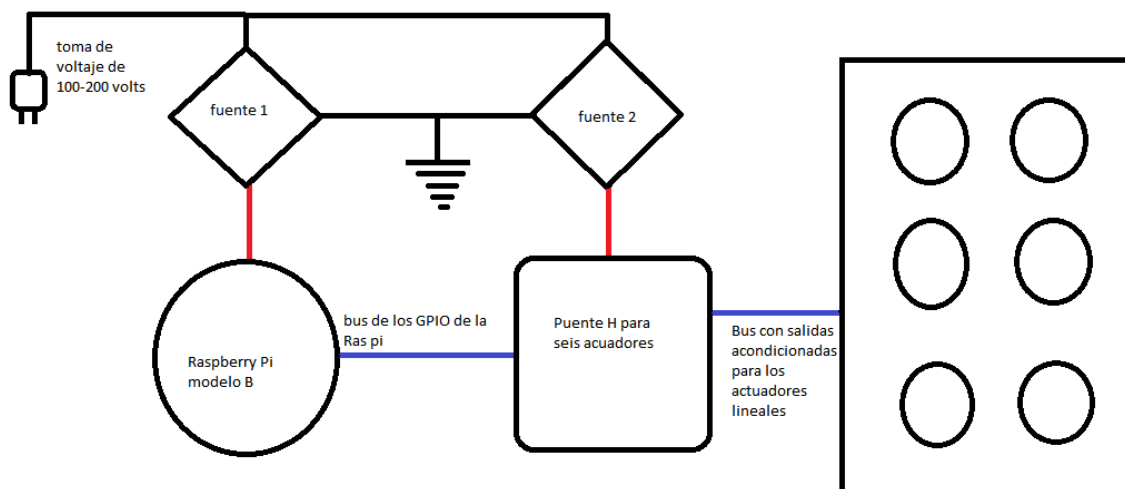


Ilustración 18. Esquema simbólico de las conexiones internas del dispositivo.

Como se muestra en la imagen anterior el cableado es lo mas simplificado posible, de una conexión a la toma de corriente eléctrica convencional de cualquier domicilio

y se usa para alimentar las dos fuentes (ilustradas como rombos) que nos suministran los voltajes necesarios para encender la Raspberry pi y los actuadores, La minicomputadora (representada por un círculo) va conectada por un bus de datos (simbolizado con una línea azul) con el bloque de puentes H (que está gráficoado como un rectángulo redondeado) se procesan las señales de la Raspberry Pi y se les da salida a los actuadores con las señales acondicionadas para ya no tener problemas.

EL resultado final de la elaboración y las conexiones ya con todos los elementos integrados es el mostrado en las imágenes 19, 20 y 21 que es mostrado a continuación.

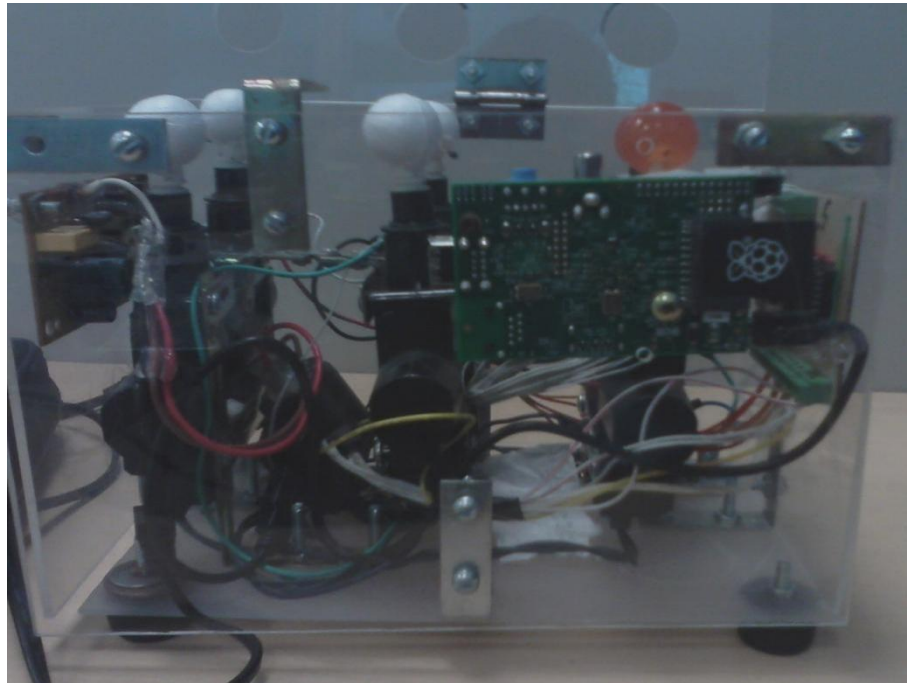


Ilustración 19. Vista lateral del sistema.

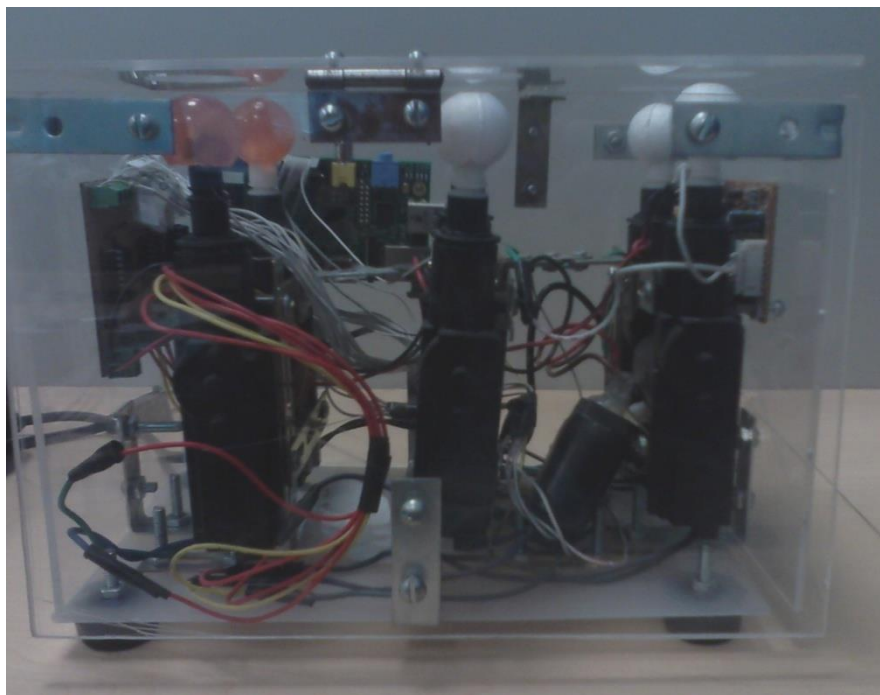


Ilustración 20. Vista lateral 2 del sistema.

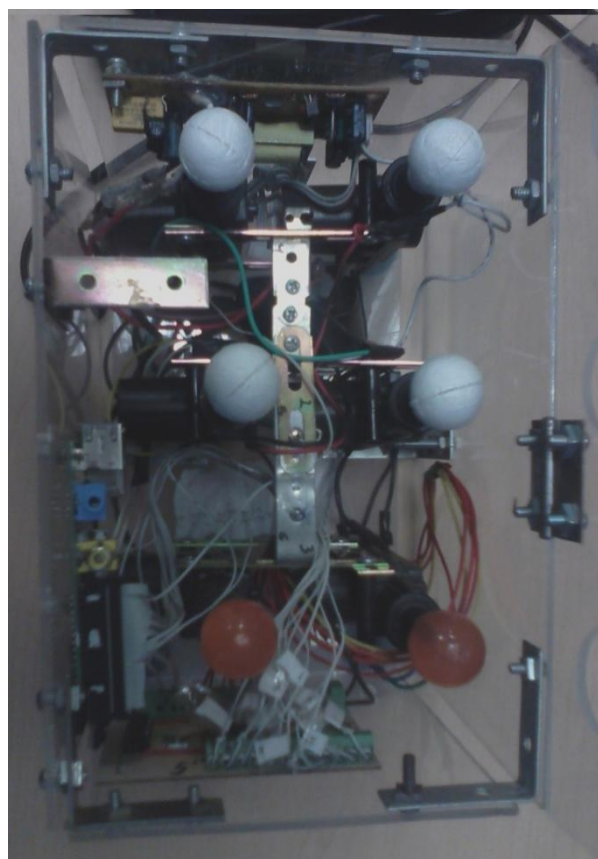


Ilustración 21. Vista superior del sistema ya terminado.

La forma en la que los buses de datos interactúan es muy inteligente y es uno de los grandes aciertos del sistema, ya que la Raspberry Pi permite por terminal manejar ceros y unos a voluntad mediante los puertos de entradas y salidas de propósito general (GPIO), simplemente se dan ceros y unos al sistema y los envía a los puentes H como señales, es decir, la verdadera importancia del acondicionamiento de señales radica en que lo que se programe va a salir en los actuadores en forma de una letra en Braille.

Cada salida de los GPIO tiene un número que debe relacionarse con la placa de los puentes H para hacer funcionar cada actuador de forma individual, cada salida del GPIO tiene un número asignado, como se muestra en la siguiente ilustración.

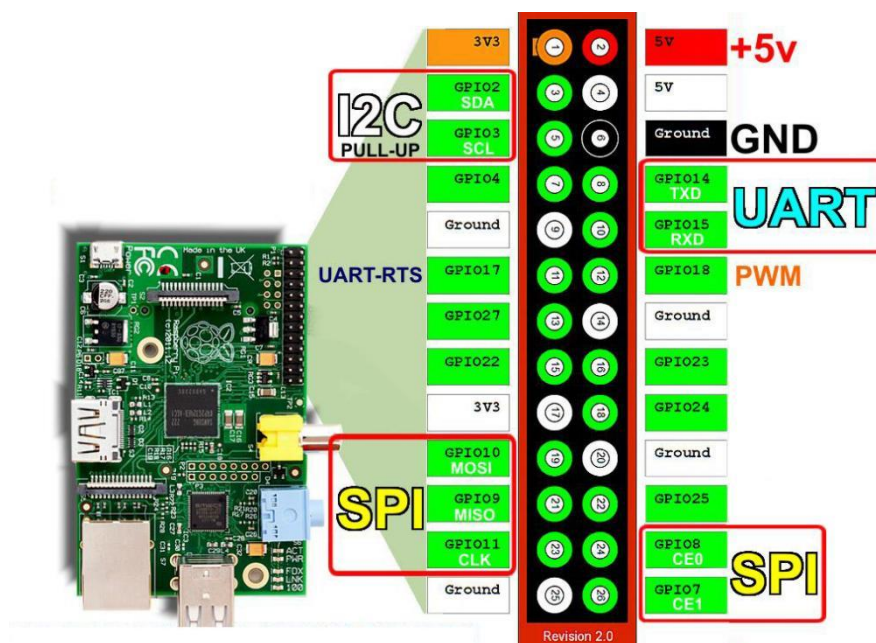


Ilustración 22. Numeración de las salidas del puerto GPIO.

[28]

A determinada salida de los GPIO le corresponde a un actuador determinado, ya sea en subida o en bajada para lograr generar cada una de las letras del alfabeto táctil.

La bibliografía siempre sugiere que el primer campo del cajetín sea el 1 considerado primero al estar en la parte superior izquierda, y el dos que sea el vertical inmediato a este y así sucesivamente.

En la siguiente ilustración se muestra lo detallado en el párrafo anterior del orden y en que actuadores se aplican que salidas de los GPIO las flechas hacia arriba de color verde son los GPIO que se deben de activar con un valor lógico en alto para hacer subir los actuadores lineales y mostrar las letras en Braille, si se activan los GPIO de las flechas grises en valores en alto se bajan los actuadores simbolizando que no existe ninguna letra que mostrar.

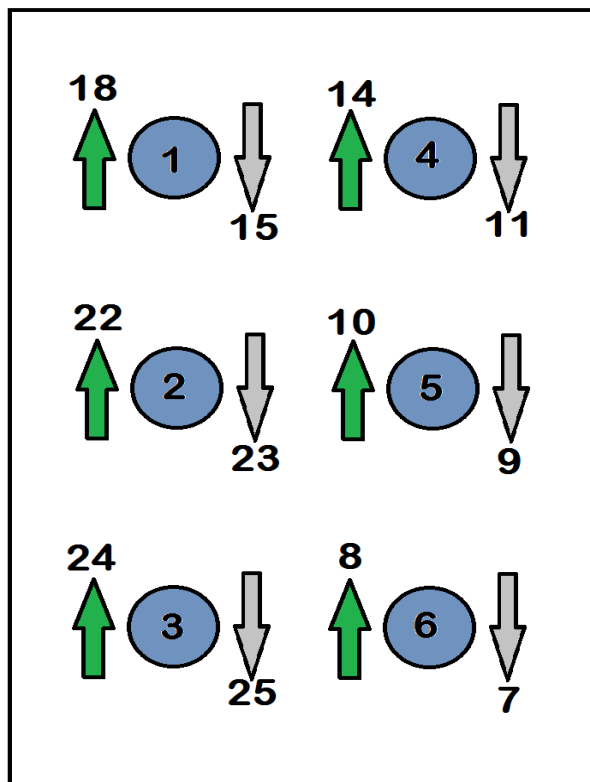


Ilustración 23. Esquema simplificado de los GPIO de subida y bajada.

Cuando el cableado quedó listo el propósito inicial del servicio social consistía en mostrar una letra programada y decirla, luego otra y decirla, es decir se deberían establecer rutinas de manejo de las letras por tiempos, al final el desarrollo mostró avances tan positivos que se decidió integrar una interfaz gráfica para hacer el desarrollo, pero tenía que ser especial.

Al principio se pensó en una salida con audio de una voz pregrabada y establecida, pero al hacerse los ajustes preliminares se determinó que esta voz no era la correcta, la intención de tomar en cuenta todos los detalles era hacer un dispositivo que no

cualquiera pudiera hacer, un dispositivo que fuera único, entonces se pensó en hacer las grabaciones por cuenta propia, el 22 de septiembre del 2013 se lanzo un concurso en la red social Twitter bajo la primicia “Dona tu voz para una maquina que enseñe Braille”, de inmediato se tuvieron 4 candidatas tangibles, tres de ellas estudiantes de comunicaciones, con experiencia en radio y en grabaciones, pero el timbre de voz debería ser especial, es decir: maternal, amigable, amable y afectuoso, de estas características en el timbre de voz la más difícil de cumplir fue la de maternal, ya que los prospectos eran jóvenes, la voz más neutral, amigable, y maternal que se encontró fue la voz de Dalila Magaña Sánchez de 21 años estudiante de la carrera de negocios internacionales en la Universidad Marista de San Luis Potosí.

La grabación no fue fácil duro cerca de tres horas en la cual se le explico el concepto de idea y se realizaron pruebas de audio con un equipo muy modesto, básicamente sólo un micrófono sencillo y una laptop equipada con el software libre Audacity.

En la siguiente tabla se muestra la lista de los audios grabados y el estado de la grabación.

NUM	PALABRA LETRA	ESTADO	REVISIO N	CAPTURAD O	LOCALIDAD	TITULO
1	A	Audio Capturado	1	SI	0H0000	AUDIO1
2	B	Audio Capturado	1	SI	0H0001	AUDIO2
3	C	Audio Capturado	1	SI	0H0002	AUDIO3
4	D	Audio Capturado	1	SI	0H0003	AUDIO4
5	E	Audio Capturado	1	SI	0H0004	AUDIO5
6	F	Audio Capturado	1	SI	0H0005	AUDIO6
7	G	Audio Capturado	1	SI	0H0006	AUDIO7
8	H	Audio Capturado	1	SI	0H0007	AUDIO8
9	I	Audio Capturado	1	SI	0H0008	AUDIO9
10	J	Audio Capturado	1	SI	0H0009	AUDIO10
11	K	Audio Capturado	1	SI	0H000A	AUDIO11
12	L	Audio Capturado	1	SI	0H000B	AUDIO12
13	M	Audio Capturado	1	SI	0H000C	AUDIO13
14	N	Audio Capturado	1	SI	0H000D	AUDIO14
15	Ñ	Audio Capturado	1	SI	0H000E	AUDIO15
16	O	Audio Capturado	1	SI	0H000F	AUDIO16

17	P	Audio Capturado	1	SI	0H0010	AUDIO17
18	Q	Audio Capturado	1	SI	0H0011	AUDIO18
19	R	Audio Capturado	1	SI	0H0012	AUDIO19
20	S	Audio Capturado	1	SI	0H0013	AUDIO20
21	T	Audio Capturado	1	SI	0H0014	AUDIO21
22	U	Audio Capturado	1	SI	0H0015	AUDIO22
23	W	Audio Capturado	1	SI	0H0016	AUDIO23
24	V	Audio Capturado	1	SI	0H0017	AUDIO24
25	X	Audio Capturado	1	SI	0H0018	AUDIO25
26	Y	Audio Capturado	1	SI	0H0019	AUDIO26
27	Z	Audio Capturado	1	SI	0H001A	AUDIO27
28	0	Audio Capturado	1	SI	0H001B	AUDIO28
29	1	Audio Capturado	1	SI	0H001C	AUDIO29
30	2	Audio Capturado	1	SI	0H001D	AUDIO30
31	3	Audio Capturado	1	SI	0H001E	AUDIO31
32	4	Audio Capturado	1	SI	0H001F	AUDIO32
33	5	Audio Capturado	1	SI	0H0020	AUDIO33
34	6	Audio Capturado	1	SI	0H0021	AUDIO34
35	7	Audio Capturado	1	SI	0H0022	AUDIO35
36	8	Audio Capturado	1	SI	0H0023	AUDIO36
37	9	Audio Capturado	1	SI	0H0024	AUDIO37
38	CORRECTO	Audio Capturado	1	SI	0H0025	AUDIO38
39	PRUEBA DE NUEVO	Audio Capturado	1	SI	0H0026	AUDIO39
40	ADIVINA LA LETRA	Audio Capturado	1	SI	0H0027	AUDIO40
41	EXCELENTE	Audio Capturado	1	SI	0H0028	AUDIO41
42	MUY BIEN	Audio Capturado	1	SI	0H0029	AUDIO42
43	EMPEZEMOS	Audio Capturado	1	SI	0H002A	AUDIO43
44	HASTA LUEGO	Audio Capturado	1	SI	0H002B	AUDIO44
45	ERROR 1	Audio Capturado	1	SI	0H002C	AUDIO45
46	ERROR 2	Audio Capturado	1	SI	0H002D	AUDIO45
47	ERROR 3	Audio Capturado	1	SI	0H002E	AUDIO45
48	ERROR 4	Audio Capturado	1	SI	0H002F	AUDIO45
49	ERROR 5	Audio Capturado	1	SI	0H0030	AUDIO45
50	ERROR 6	Audio Capturado	1	SI	0H0031	AUDIO45
55	DEMO	Audio Capturado	1	SI	0H0032	AUDIO46
56	AMOR	Audio Capturado	1	SI	0H0033	AUDIO47
57	BONITO	Audio Capturado	1	SI	0H0034	AUDIO48
58	CASA	Audio Capturado	1	SI	0H0035	AUDIO49
59	DIA	Audio Capturado	1	SI	0H0036	AUDIO50
60	ELEFANTE	Audio Capturado	1	SI	0H0037	AUDIO51

61	FELIZ	Audio Capturado	1	SI	0H0038	AUDIO52
62	GRANDE	Audio Capturado	1	SI	0H0039	AUDIO53
63	HOLA	Audio Capturado	1	SI	0H003A	AUDIO54
64	IDEA	Audio Capturado	1	SI	0H003B	AUDIO55
65	ESPACIO	Audio Capturado	1	SI	0H003C	AUDIO56
66	.	Audio Capturado	2	SI	0H003D	AUDIO58
67	,	Audio no Capturado	3	NO	0H003E	AUDIO57

Tabla 3. Lista de palabras usadas en la Ras Pi

Ya con todo listo sólo faltaba hacer la programación, pero al encender la Raspberry Pi modelo B se notó que ni siquiera poseía un sistema operativo, cuando se compró el Starter Kit de Raspberry Pi venía con una memoria de 8gb en la cual se incluía Noobs (mencionado anteriormente en el marco teórico) que se utiliza la primera vez que se va a usar el sistema.

Para que todo funcione según todas las referencias electrónicas se pensó en Raspbian como sistema operativo, así que era hora de instalar, para hacer esto se conectó el cable de la alimentación (conectando el USB a cualquier puerto de la Laptop o empleando un adaptador de 5V. para conectar el cable USB y que suministre más de 500mA). Esto iniciará directamente la Raspi, ya que carece de interruptor *ON/OFF*. Se verá que está encendida por el titileo de los LEDs.

1. Primero arrancará Noobs y seleccionamos la opción recomendada tal como se muestra en la ilustración 24 presentada en la siguiente página.

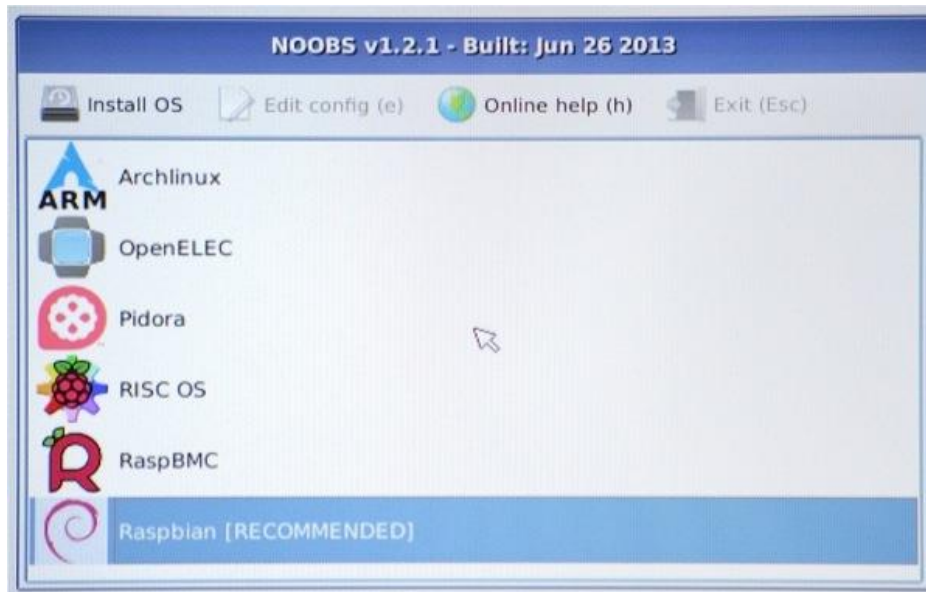


Ilustración 24. Pagina de inicio de NOOBS

2. Al seleccionar Raspbian se instala el sistema operativo y tras una instalación de unos 10 minutos, reinició.
3. En el primer arranque llevó directamente al menú de configuración. Se seleccionaron los parámetros adecuados como la zona horaria, idioma del teclado, contraseña, etc. Algunas de las opciones más interesantes vistas fueron, por ejemplo, *Expand root partition to fill SD card* para expandir la partición de la SD para que ocupe toda la tarjeta (en caso de tener una capacidad superior a 4Gb). En *Set keyboard layout* se configuro el modelo de teclado, eligiendo el *Generic* en caso de no estar muy seguros. *Change password for: 'pi'* permite cambiar la contraseña por defecto por la que se desea. Normalmente por defecto el nombre de usuario es *pi* y la contraseña *raspberry*, que debemos cambiar por una más segura. Hay que recordar que la pide dos veces para verificarla. Siguiendo con más opciones se tiene *Set locale* y *Set timezone*, para seleccionar el idioma de preferencia para el usuario final (*es_ES.UTF-8 UTF-8*) y zona horaria (*México*). También es posible configurar *Internet TCP/IP* si se va a utilizar. Para la configuración *TCP/IP* se escribe en la consola:

sudo nano /etc/network/interfaces

En el fichero editable que se abre, se borra la línea:

iface eth0 inet dhcp

Luego, al final del fichero se añaden unas líneas nuevas. Las líneas son las siguientes (sustituyendo los números de *IP* y *gateway* del ejemplo por los propios):

auto eth0

iface eth0 inet static

address 192.168.1.64

netmask 255.255.255.0

gateway 192.168.1.255

Una vez realizados los cambios, se reinicia para que al ser lanzado el controlador de Red del Kernel surtan efecto las configuraciones. Para ello se puede escribir en la consola *sudo reboot -f* y presionar *ENTER*.

Al reiniciarse, el sistema pidió el nombre de usuario y contraseña. Al introducirlos apareció el *prompt (LXterminal)* a la espera de introducir un comando. Si se escribe *startx* se inicia el modo gráfico. Ahora ya podemos movernos por nuestro nuevo Raspbian con el uso del *mouse* y del teclado.

Una vez finalizado, ya se pudo comenzar a utilizar la miniPC y a instalar las aplicaciones que se necesitaron para lograr mostrar letras en Braille desde una interfaz gráfica.

Como ya estaba la Raspberry Pi trabajando y configurada, se prosiguió a configurar un servidor VNC para hacer funcionar todo desde una Laptop, en este caso ya se tenía configurada el IP fija, así que eso no fue problema, primero en la Laptop se instaló *PuTTY* para establecer comunicación con la Raspi por línea de comandos en una terminal, sólo se tecleó la IP **192.168.1.64**, el usuario que es “pi” y la contraseña

que es “qwertyu” y se entró al sistema, ya enlazado se tuvo que lanzar el servicio VNC pero en una Raspbeery PI con Raspbian no viene instalado por defecto, así que lo instalamos simplemente tecleando en la línea de comandos de la terminal la siguiente instrucción.

```
sudo apt-get update
```

```
sudo apt-get upgrade
```

```
sudo apt-get install tightvncserver
```

Una vez instalado sólo reiniciamos con:

```
sudo reboot -f
```

Ya reiniciado se lanzó el servicio con los parámetros de cada sistema en este caso lanzamos con la siguiente configuración tecleándola en la línea de comandos de PuTTY.

```
vncserver :1 -geometry 1280x800 -depth 16 -pixelformat rgb565
```

Donde se lanzó el servicio VNC en un primer único escritorio, con una resolución de 1280 de largo por 800 de alto, con una profundidad de 16 bits y un formato estándar de colores.

Ya con estos parámetros introducidos en la Raspi, se lanzó un escritorio virtual, lo único que faltaba hacer es inicial el servicio VNC, para esto se usó el software RealVNC.

Para ver el escritorio virtual, se abre la aplicación, en la primera ventana que salió se tecleo la IP y la contraseña mencionada anteriormente y ya podemos visualizar nuestra Raspi desde alguna *Lap-top* como si fuera un simple programa.

Cuando se iba a construir la aplicación es cuando en realidad se da cuenta de que en cuanto a software viene muy limitada así que hay que instalar la nueva paquetería y actualizar el sistema, con los siguientes comandos en la terminal actualizamos al núcleo a su última versión segura y estable, actualizamos los paquetes, instalamos

una serie de librerías de compiladores de GNU basados en C, se prosiguió a instalar el soporte de QT4 e instalamos el IDE Qtcreator y un reproductor multimedia por consola con las siguientes instrucciones de terminal en su LXterminal.

sudo apt-get update

sudo apt-get upgrade

sudo apt-get install g++

sudo apt-get install qt4-dev-tools

sudo apt-get install qtcreator

sudo apt-get install mplayer

Al abrir el menú de programación ya se encuentra el icono de QT4, así que al abrir lanza un asistente para la creación de aplicaciones con entorno gráfico, en una computadora cualquiera es muy fácil generar aplicaciones en QT4, en Raspi, es un poco más difícil, ya que se deben configurar los compiladores y unas capas del Shell, pero sólo la primera vez, una vez configurado no se tendrá problema, el modo de configurar es el siguiente.

Se abre Qtcreator

En la ventana se va a *tools>>options>>build & Run>>Qtversions* se asegura que el Qmake este seleccionado.

En la ventana se va a *tools>>options>>build & Run>>Tool and chains* se añade GCC. En el *compiler Path* se añade un archivo ubicado en /usr/Bin llamado gcc. En el *Debugger* se añade un archivo ubicado en /usr/Bin llamado gdb, se guardan los cambios y se cierra, con estas acciones básicamente se cargó el compilador necesario para hacer funcionar la Raspberry pi con Qtcreator y se asegura que no dé problemas más adelante, este paso usualmente en un sistema basado en un núcleo Linux no es necesario, pero debido a que la Raspi esta diseñada sobre ARMV6 y es en sí, un sistema embebido su gcc y su gdb están desarrollados por

Linaro, que es en sí un conjunto de herramientas de desarrollo basadas para que Linux corra sin ningún problema en plataformas ARM y otras arquitecturas no específicas.

Ahora con todo en orden se va a desarrollar como se hizo el proyecto en QtCreator.

Primero en la ventana principal se crea un nuevo proyecto y se inicia un asistente *new file or project*, y lanza la aplicación un cuadro de dialogo en el cual se selecciona en *Aplicacion*, *QT gui Aplicacion* y el asistente lanza otra ventana en la cual es introducido el nombre del proyecto, en ese caso se llama pruebaRoot, ya que al principio tuvimos problemas para darles valores de súper administrador a la Raspi, problema que tiene solución al inicial sesión con la instrucción por terminal.

sudo -i

Luego a los formularios que aparecen a continuación se le dice a todo que si ya que al cargar los archivos del compilador y del *Debugger* todo se auto configuró, finalmente, se selecciona la ubicación de los archivos y se inicia con la programación.

[29]

Cómo se construyo la aplicación.

Primero se elaboró una interfaz en la cual se programaron sólo seis letras, de la A a la F, un botón de arranque, un botón de salida y un botón para dar opciones para elegir otra letra, a continuación se muestra cómo se relacionaron los objetos para elaborar las rutinas.

Para el botón de arranque.

```
system("sudo echo 23 >/sys/class/gpio/export");  
  
system("sudo chmod 777 -R /sys/class/gpio/gpio23 ");  
  
system("sudo echo out >/sys/class/gpio/gpio23/direction");  
  
system("sudo echo 25 >/sys/class/gpio/export");  
  
system("sudo chmod 777 -R /sys/class/gpio/gpio25 ");
```

```
system("sudo echo out >/sys/class/gpio/gpio25/direction");

    system("sudo echo 7 >/sys/class/gpio/export");

    system("sudo chmod 777 -R /sys/class/gpio/gpio7 ");

system("sudo echo out >/sys/class/gpio/gpio7/direction");

    system("sudo echo 8 >/sys/class/gpio/export");

    system("sudo chmod 777 -R /sys/class/gpio/gpio8 ");

system("sudo echo out >/sys/class/gpio/gpio8/direction");

    system("sudo echo 9 >/sys/class/gpio/export");

    system("sudo chmod 777 -R /sys/class/gpio/gpio9 ");

system("sudo echo out >/sys/class/gpio/gpio9/direction");

    system("sudo echo 10 >/sys/class/gpio/export");

    system("sudo chmod 777 -R /sys/class/gpio/gpio10 ");

system("sudo echo out >/sys/class/gpio/gpio10/direction");

    system("sudo echo 11 >/sys/class/gpio/export");

    system("sudo chmod 777 -R /sys/class/gpio/gpio11 ");

system("sudo echo out >/sys/class/gpio/gpio11/direction");

    system("sudo echo 14 >/sys/class/gpio/export");

    system("sudo chmod 777 -R /sys/class/gpio/gpio14 ");

system("sudo echo out >/sys/class/gpio/gpio14/direction");

    system("sudo echo 15 >/sys/class/gpio/export");

    system("sudo chmod 777 -R /sys/class/gpio/gpio15 ");
```

```
system("sudo echo out >/sys/class/gpio/gpio15/direction");  
  
    system("sudo echo 18 >/sys/class/gpio/export");  
  
    system("sudo chmod 777 -R /sys/class/gpio/gpio18 ");  
  
system("sudo echo out >/sys/class/gpio/gpio18/direction");  
  
    system("sudo echo 22 >/sys/class/gpio/export");  
  
    system("sudo chmod 777 -R /sys/class/gpio/gpio22 ");  
  
system("sudo echo out >/sys/class/gpio/gpio22/direction");  
  
    system("sudo echo 24 >/sys/class/gpio/export");  
  
    system("sudo chmod 777 -R /sys/class/gpio/gpio24 ");  
  
system("sudo echo out >/sys/class/gpio/gpio24/direction");
```

Para el botón de salida.

```
    system("sudo echo 7 >/sys/class/gpio/unexport");  
  
    system("sudo echo 8 >/sys/class/gpio/unexport");  
  
    system("sudo echo 9 >/sys/class/gpio/unexport");  
  
system("sudo echo 10 >/sys/class/gpio/unexport");  
  
system("sudo echo 11 >/sys/class/gpio/unexport");  
  
system("sudo echo 14 >/sys/class/gpio/unexport");  
  
system("sudo echo 15 >/sys/class/gpio/unexport");  
  
system("sudo echo 18 >/sys/class/gpio/unexport");  
  
system("sudo echo 22 >/sys/class/gpio/unexport");  
  
system("sudo echo 23 >/sys/class/gpio/unexport");
```

```
system("sudo echo 24 >/sys/class/gpio/unexport");  
system("sudo echo 25 >/sys/class/gpio/unexport");  
system("mplayer /root/audiobraillinux/audio44.ogg");
```

Para el botón de dar opciones de elegir otra letra.

```
system("echo 1 >/sys/class/gpio/gpio7/value");  
system("echo 1 >/sys/class/gpio/gpio9/value");  
system("echo 1 >/sys/class/gpio/gpio11/value");  
system("echo 1 >/sys/class/gpio/gpio15/value");  
system("echo 1 >/sys/class/gpio/gpio23/value");  
system("echo 1 >/sys/class/gpio/gpio25/value");
```

La letra “a”

```
system("mplayer /root/audiobraillinux/audio1.ogg");  
system("echo 1 >/sys/class/gpio/gpio18/value");  
system("mplayer /root/audiobraillinux/audio47.ogg");
```

La letra “b”

```
system("mplayer /root/audiobraillinux/audio2.ogg");  
system("echo 1 >/sys/class/gpio/gpio18/value");  
system("echo 1 >/sys/class/gpio/gpio22/value");  
system("mplayer /root/audiobraillinux/audio48.ogg");
```

La letra “c”

```
system("mplayer /root/audiobraillinux/audio3.ogg");  
    system("echo 1 >/sys/class/gpio/gpio18/value");  
    system("echo 1 >/sys/class/gpio/gpio14/value");  
system("mplayer /root/audiobraillinux/audio49.ogg");
```

La letra “d”

```
system("mplayer /root/audiobraillinux/audio4.ogg");  
    system("echo 1 >/sys/class/gpio/gpio18/value");  
    system("echo 1 >/sys/class/gpio/gpio14/value");  
    system("echo 1 >/sys/class/gpio/gpio10/value");  
system("mplayer /root/audiobraillinux/audio50.ogg");
```

La letra “e”

```
system("mplayer /root/audiobraillinux/audio5.ogg");  
    system("echo 1 >/sys/class/gpio/gpio18/value");  
    system("echo 1 >/sys/class/gpio/gpio10/value");  
system("mplayer /root/audiobraillinux/audio51.ogg");
```

La letra “f”

```
system("mplayer /root/audiobraillinux/audio6.ogg");  
    system("echo 1 >/sys/class/gpio/gpio22/value");  
    system("echo 1 >/sys/class/gpio/gpio14/value");  
    system("echo 1 >/sys/class/gpio/gpio18/value");
```

```
system("mplayer /root/audiobraillinux/audio52.ogg");
```

Entonces se recuerda la imagen 23, en la cual cada salida de letra corresponde a un GPIO, entonces se nota qué cada salida del GPIO numerada corresponde a los elementos necesarios para generar una letra, ese es el verdadero truco en el dispositivo, así que con poco código se podrá lograr elaborar cada letra o cada combinación que se necesite.

Una interfaz gráfica muy humilde y sólo unas pocas letras en la pantalla, es lo que seis meses de investigación y desarrollo lograron hacer, la siguiente figura ejemplifica todo lo que se logró en todo el servicio social.

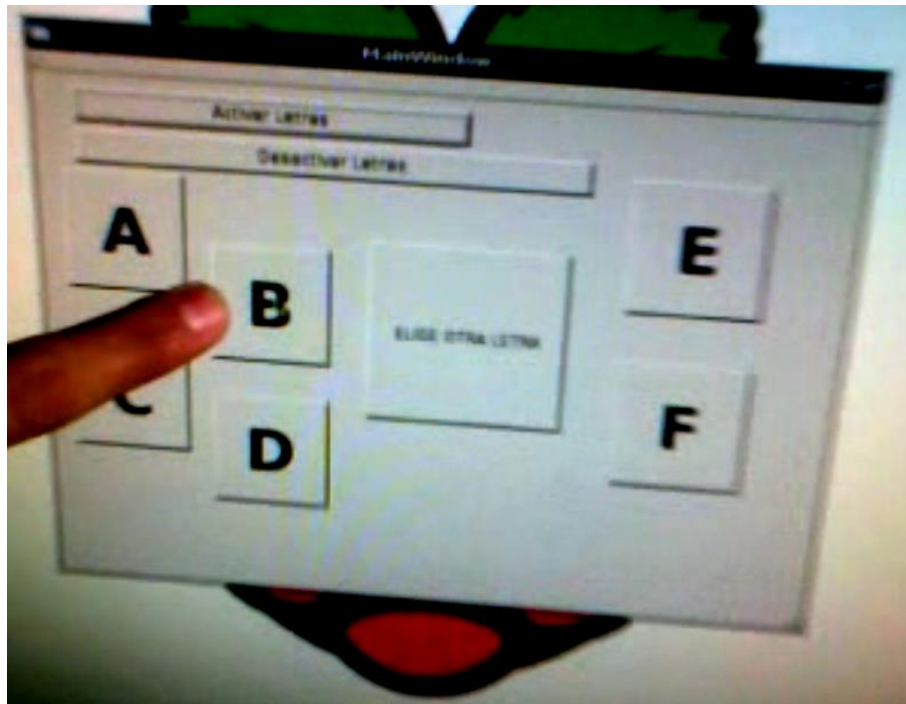


Ilustración 25. Interfaz finalmente entregada como servicio social.

El proyecto se finalizó para la etapa de servicio social oficialmente el 10 de enero del 2014, con una reunión en la escuela para ciegos y débiles visuales, propiedad de Instituto para Ciegos AC, en la cual las instancias más altas de la asociación conocieron el proyecto y decidieron apoyar el desarrollo y brindaron su apoyo para hacer pruebas en la escuela para personas con ceguera y debilidad visual.

CAPÍTULO 5

Lo realizado como Residencia profesional.

Inició oficialmente un 14 de febrero del 2014 en el centro de gestión y transferencia tecnológica del conocimiento, en una entrevista para nuevos residentes en áreas y proyectos de investigación se habló de Brailinox, la respuesta fue inmediata y se empezó la residencia profesional muy rápido, se trabajó desde casa cerca de un mes, hasta que oficialmente, por petición de la Diputada estatal Claudia Anaya Mota, se turno el proyecto en el COZCYT, y, al mismo tiempo, gracias a la alianza entre el Laboratorio internacional de software libre y CGRATEC se pudo realizar la residencia profesional en su LABSOL.

La idea de rediseño fue muy específica, llegar a los alcances de hardware y software del Raspberry Pi, mediante implementaciones de diseño en hardware y software.

Lograr procesar audio de una manera elegante y sencilla pero a la vez funcional con el propósito de hacer un desarrollo rápido, a prueba de fallas y muy inteligente.

El primero objetivo tangible fue hacer un rediseño de la mecánica del dispositivo, en lugar de tener una estructura flotada, cambiar el modelo y tener la opción de una estructura única lineal que atravesara los actuadores por sus barrenos establecidos y así lograr un perfecto ajuste en el sistema, hacer esto implicaba construir un tornillo de 26 cm. de largo y con un ancho de 3 mm, encontrarlo no fue posible, había un proveedor que manejaba tornillos a medida, el hecho fue que eran de grado quirúrgico y pidió unas pruebas pero desafortunadamente no llegaron a tiempo para cubrir el diseño en el periodo de la residencia profesional.

El segundo gran reto de diseño implicaba una decisión de diseñador, una de esas decisiones que hacen grandes diferencias, el reto era construir un sistema empaquetado en Brailinox, que ofreciera a la persona con ceguera o debilidad visual

la sensación de confort y trato personal, muchas veces no se pone atención a los detalles, en este caso, se deben cuidar mucho.

Al ser un producto enfocado a ciegos y débiles visuales se debe considerar que el producto debe tener ciertas características específicas ya que el mercado es muy específico, por ejemplo, la voz debía cumplir ciertas características que ya se cubrieron en las grabaciones, aunque, la misma física de los semiconductores de estado sólido limita la experiencia que el usuario pudiera recibir, para complementar esto se propuso el reto de desarrollar un sistema de audio de calidad, es decir, un amplificador con un ancho de banda de 100 kHz, y una bocina que respondiera al audio para darle al usuario final la sensación de personalidad y maternidad que él necesita para tener confianza y aprender en un estado muy cómodo.

Primero se decidió buscar un amplificador de bajo costo, pero de buena calidad, que tuviera buena relación de consumo-potencia y que cumpliera la característica principal que es poder reproducir sonido de alta calidad en un gran ancho de banda sin perder fidelidad ni sobrecalentarse, se usó una configuración de amplificador específica para voz, con un ancho de banda cercano a los 100kHz (95kHz) estuvo muy sobrado en calidad, pero no es un integrado de alto desempeño, sólo tiene 2.5 Watts como potencia máxima, pero fue suficiente para la aplicación. A continuación se muestran imágenes del diseño y su puesta en marcha en la vida real.

El siguiente diagrama es decir, en la ilustración 26, se ejemplifica de un mejor modo como entra la señal y como después de ser muestreada por el *bypass* de alta frecuencia nos permite responder a casi cualquier ancho de banda, en este caso el muestreo más alto que se usará es 44.00 Khz así que, en teoría, el amplificador podría ser concebido como un modelo de caja negra en el cual introducimos una señal audio ciega y al salir de la caja negra pudiera ser casi idea la salida, para no tener pérdidas, que implica que, lo mismo que entra es lo mismo que sale, pero amplificando la ganancia pero sin aumentar significativamente el error.

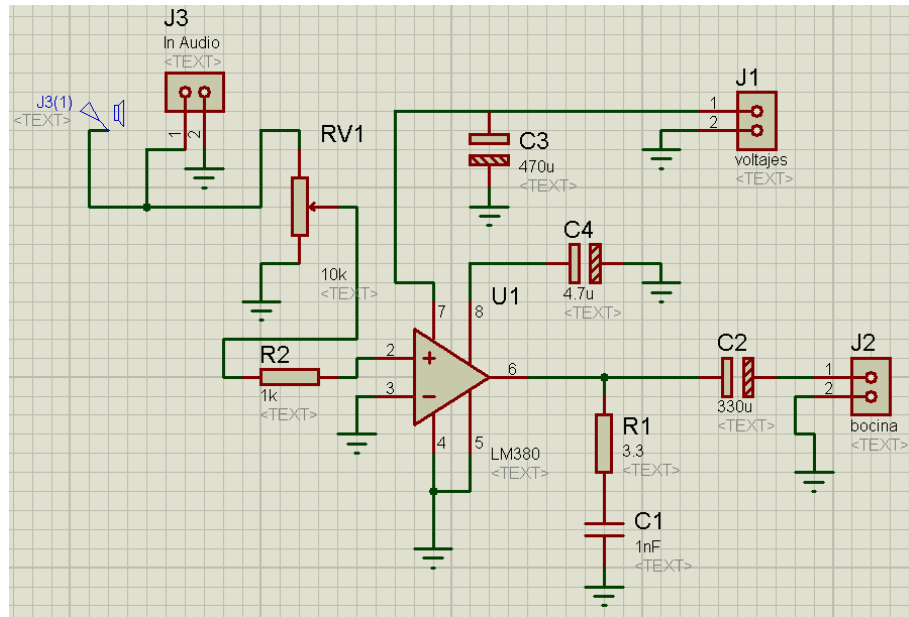


Ilustración 26. Diagrama de conexiones

Luego que se analizó la entrada con el osciloscopio virtual, se mide la respuesta a las oscilaciones y la distorsión. Con las pruebas hechas se notó que el error que se obtiene a la salida es bajo, el siguiente paso fue hacer el circuito en tablilla de prácticas (*protoboard*), usando semiconductores de calidad comercial se ensambló todo y al ver que el resultado fue tan bueno como en la simulación, se hacen pruebas de audio.

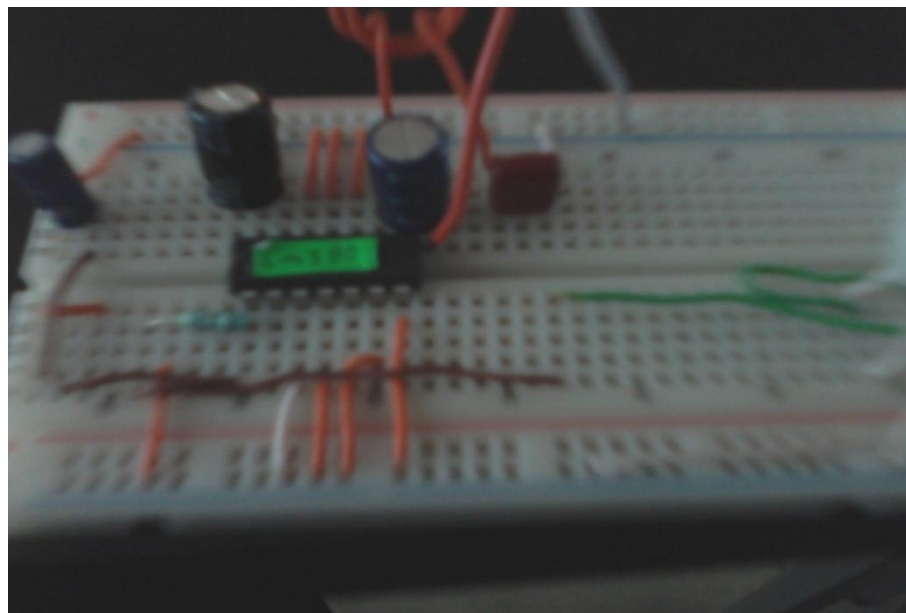


Ilustración 27. LM380 en la tablilla de prácticas.

Al final se le agrego un Plug de 3.5 mm. para enviar audio desde la lap-top y un potenciómetro para aumentar el error y ver que tanto se distorsionaba el sonido.

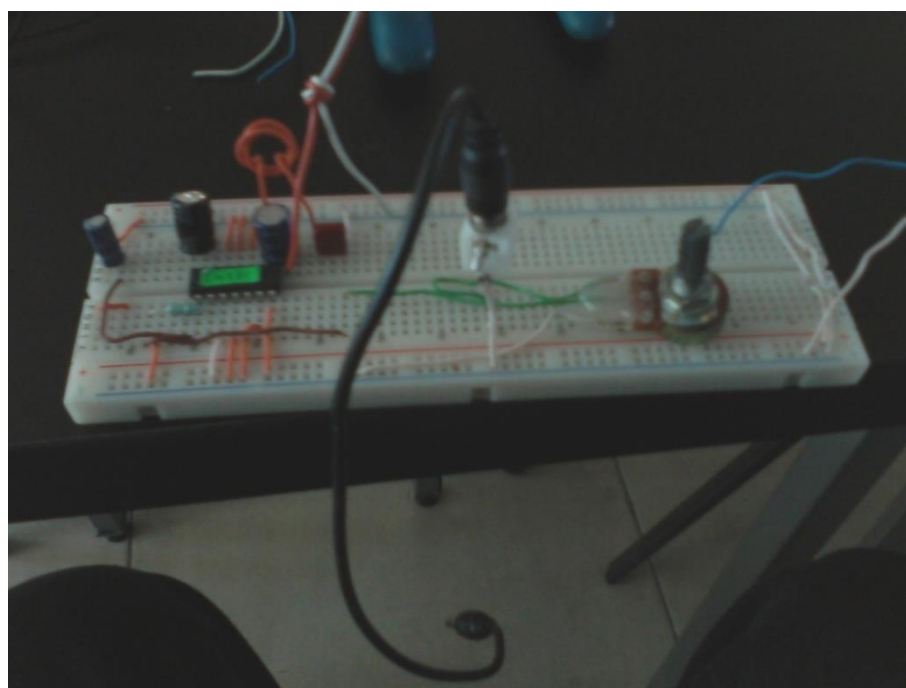


Ilustración 28. Circuito terminado.

Afortunadamente la respuesta del diagrama fue óptima y todo funcionó según los diseños, la oscilación del *bypass* fue buena y con una respuesta increíble, entonces

proseguimos a diseñar una tarjeta en la cual la planta ya no se moviera y el sistema fuera más comercial, predecible y confiable.

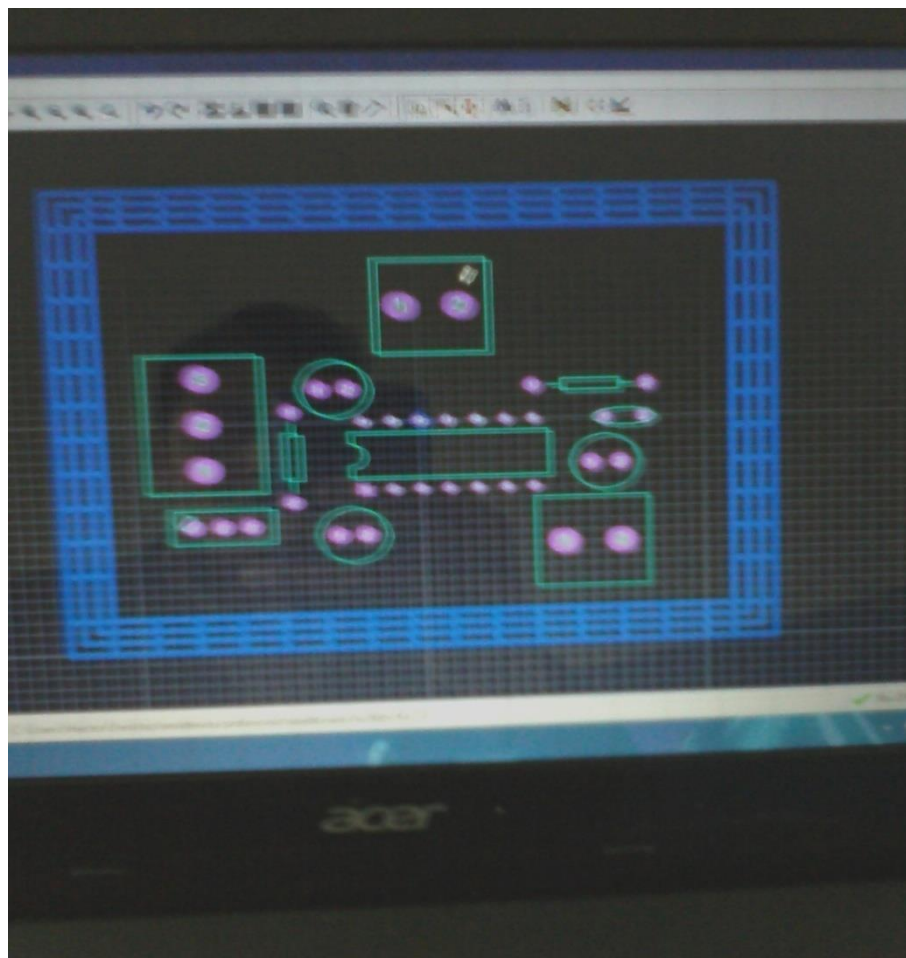


Ilustración 29. Diseño en PCB y ordenado de los componentes.

En la imagen 29 se muestra la jaula de aislamiento que se usó para el circuito, usando el método de anulación de corrientes parásitas de Faraday, se logró neutralizar el ruido proveniente del sistema y la centralización del LM380n-14 permite una correcta distribución de los componentes y permite el aislamiento de las tierras para eliminar el ruido blanco producido por los ruidos del *buffer* del audio.

Al mismo tiempo se dimensiona el tamaño del amplificador, teniendo una cobertura total de componentes en 4 pulgadas cuadradas.

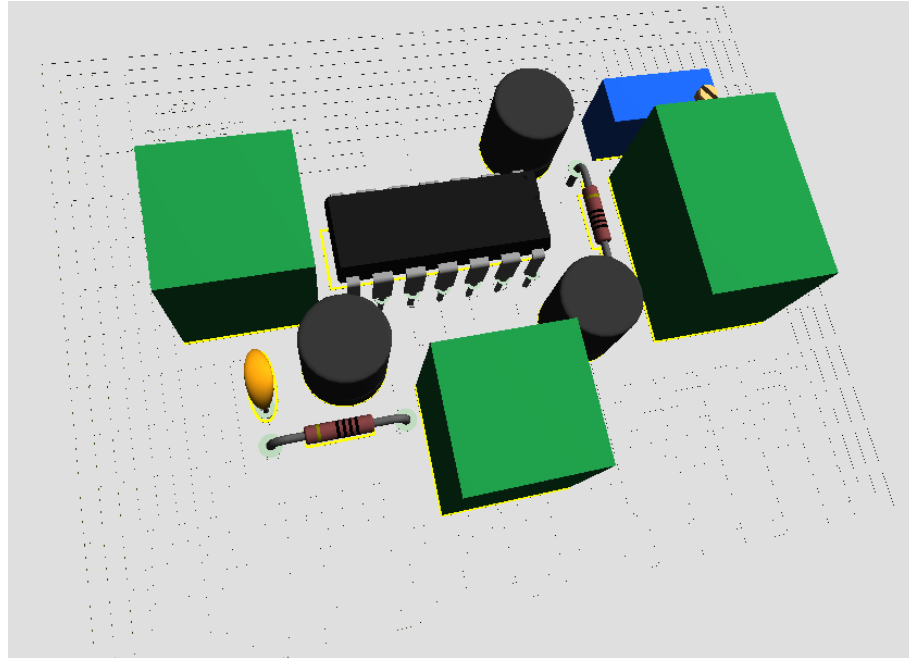


Ilustración 30. Render de la vista superior.

Este es el diseño total del amplificador ya realizado y tomando en cuenta su vista superior, cabe destacar que el uso de los bloques verdes son entradas y salidas, el bloque del amplificador esta al centro en negro y el resto de los componentes son semiconductores de bajo costo y de uso estratégico.

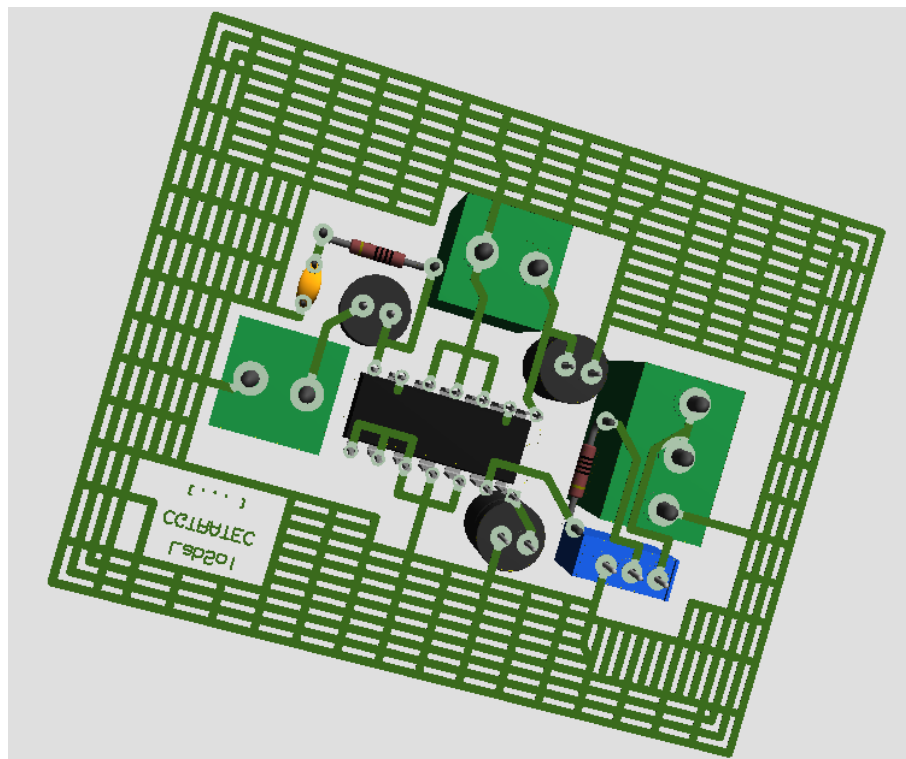


Ilustración 31. Renderizado de la vista inferior del amplificador

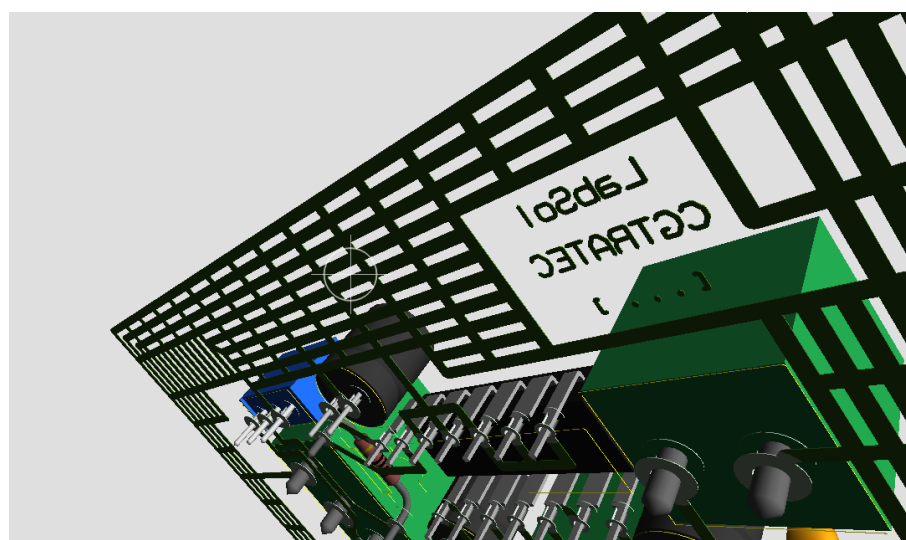


Ilustración 32. Vistas de LabSol y CGTRATEC

En la imagen 32 se aprecian las ilustraciones inferiores del Render ya con LABSOL y la jaula de protección correctamente enlazada a tierra.

Como en la tablilla de prácticas todo funcionó era hora de hacerlo real, así que el diseño ya generado se imprimió del PDF para plancharlo sobre el cobre y este fue el resultado obtenido.

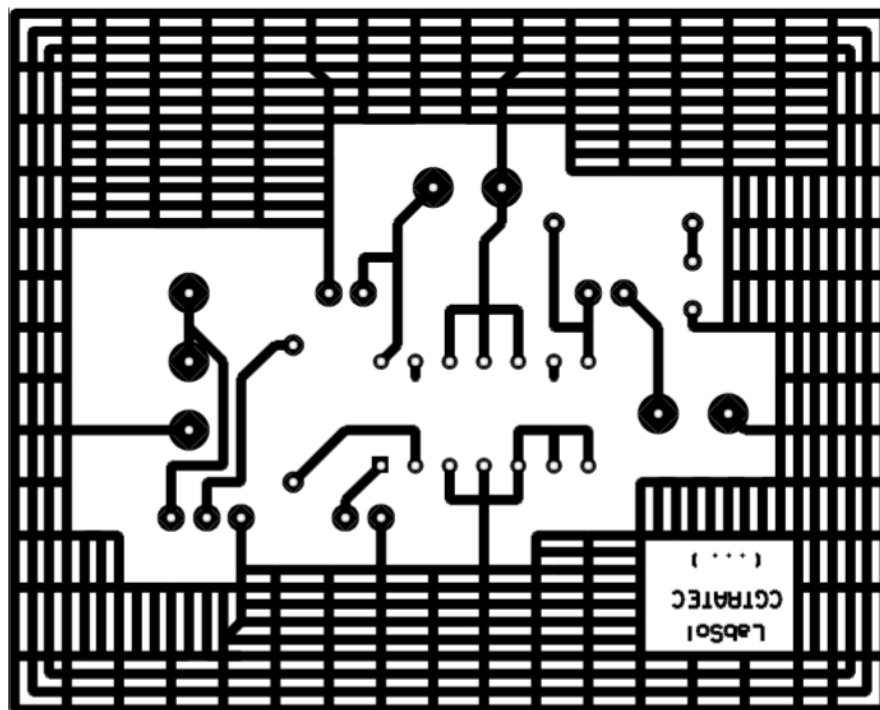


Ilustración 33. Diseño antes de ser planchado sobre el cobre.



Ilustración 34. Diseño impreso sobre el cobre

En la ilustración 34 se observa la placa de baquelita con cobre ya impresa y correctamente trabajada para ser introducida al Cloruro Ferrico, la ilustración de la imagen siguiente describe como en el jardín de COZCYT LABSOL se desarrolló toda la química necesaria para deshacerse del cobre que no se requiere para la circuitería.



Ilustración 35. Tablilla en el cloruro férrico.

Como ya se había visto antes, es recomendable hacer este proceso en un día muy soleado, y se dejó que el cloruro se templara un poco con el sol pues la bibliografía recomienda no usar la química totalmente fría.

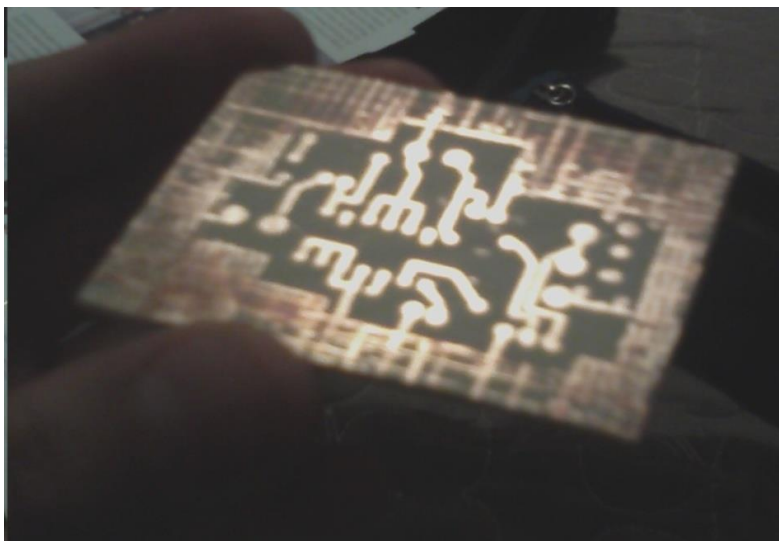


Ilustración 36. Tablilla ya impresa y recién limpiada.

En la ilustración 36 se aprecia la placa ya cortada, sin excedente del cobre ni tinta, en este punto ya sólo faltaba prepararla y dejarla lista para el proceso de barrenado.

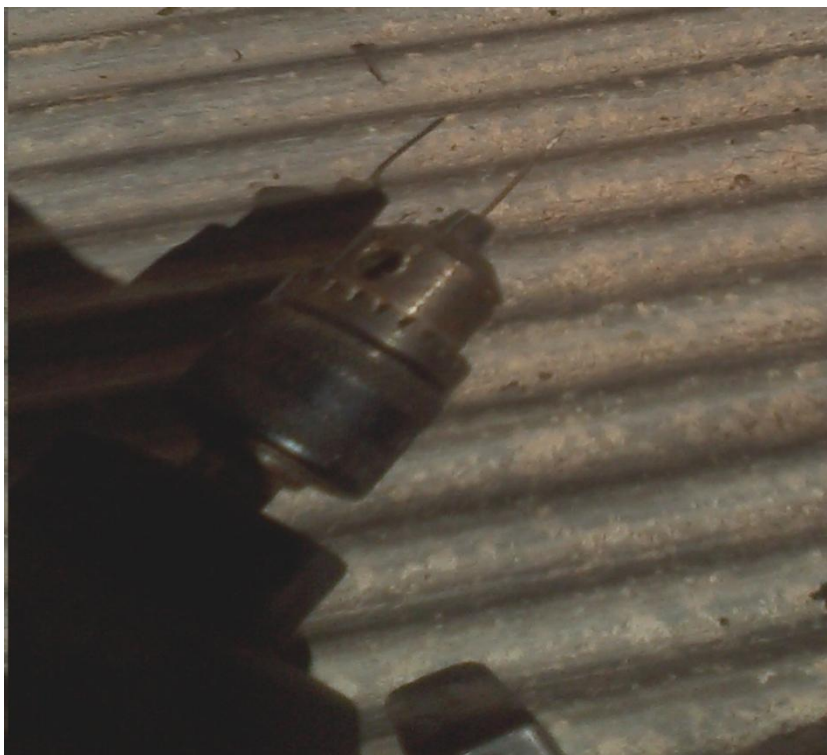


Ilustración 37. Taladro con el que se barrenó la tablilla

Como no se dispuso del equipo adecuado para hacer un barrenado limpio como un Driller y un cortador de 1/64" pues se improvisó en un taladro una aguja de costura nctr-06 que es usada para máquinas de coser y se barrenó la tarjeta con los resultados correctos.



Ilustración 38. Tablilla ya barrenada.

Aquí se muestra la tarjeta ya barrenada y lista para aplicar en sellador y la soldadura, cabe notar que este proceso fue particularmente largo por la complejidad del amplificador, es decir, influyen muchos factores que determinan si algo es fácil o no lo es por su composición, en este caso había muchos factores que determinaron que la elaboración del amplificador tomara mucho tiempo entre ellos: que el circuito es lineal en Dip-8, que todos los semiconductores son de diferente clase, que la jaula resulto ser un gran reto de diseño, y que la decisión de hacer un diseño centralizado situó a todos los componentes de modo asimétrico distribuidos alrededor de la parte central de la placa.

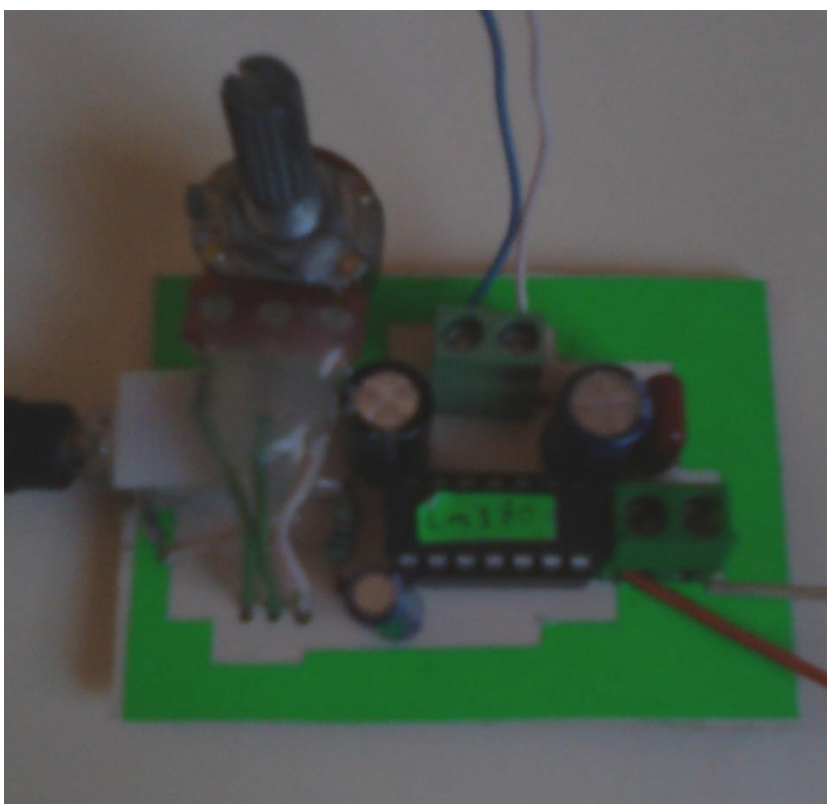


Ilustración 39. Amplificador terminado vista superior.

Esta es la vista superior final del amplificador ya funcionando con sus entradas, salidas, alimentaciones y control de volumen, pero sobre todo hay que hacer resaltar una característica, es de un tamaño un tanto reducido pues para este diseño conviene tener un amplificador reducido en dimensiones.

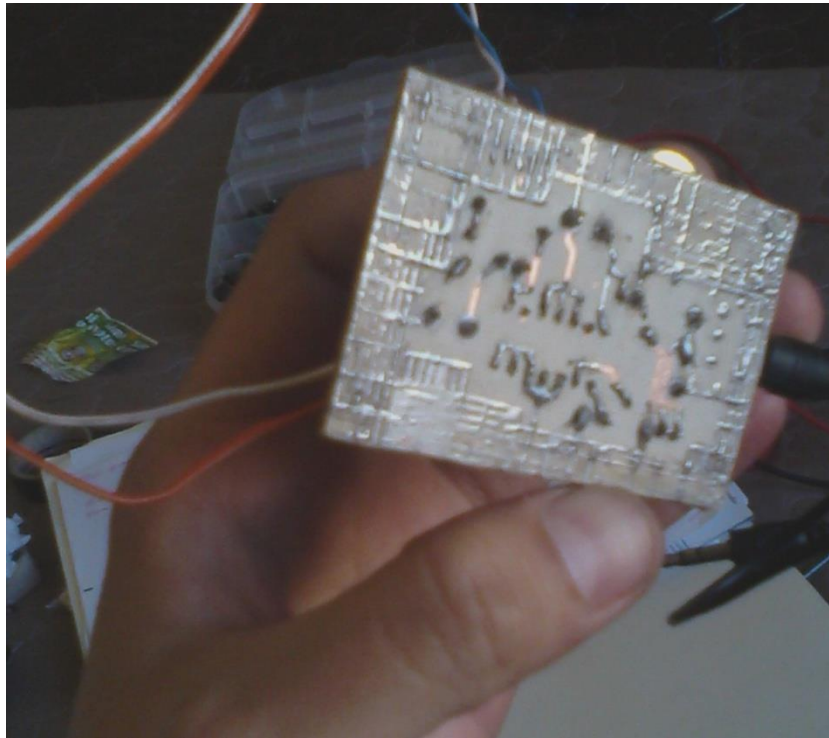


Ilustración 40. Amplificador terminado vista inferior

En la ilustración 40 se ve la vista inferior del diseño totalmente aterrizado en sus extremos y es muy confiable, el ruido es poco y sus corrientes parásitas no se manifiestan de modo significativo en el audio.

Ya con el amplificador diseñado se prosiguió a elegir la bocina necesaria para hacer funcionar el dispositivo, debería de ser una bocina con una impedancia de 8 Ohms, de tamaño muy reducido y que soportara más de 3 Watts, pero sobre todo que el bocina respondiera correctamente al ancho de banda, es decir, que la bobina en el cono responda a las frecuencias especificadas, lo ideal es que respondiera de 100Hz. a 120 kHz, lamentablemente ninguna bocina comercial responde a rangos entre 500Hz - 4 kHz y las bocinas especializadas en acústica sólo responden de 200 Hz a 8 kHz es decir que no importaba que el amplificador tuviera un ancho de banda increíblemente grande pues al final la bocina nunca podría responder a la velocidad de respuesta que tiene el amplificador.

Al finalizar el desarrollo del amplificador con la bocina que se tenía y que no es de alta respuesta, se notó que el sistema diseñado era perfecto para la inclusión de

nuevas características mediante la teoría modular, al conectar un plug o un bus ya se tenían nuevas características en el dispositivo sin modificar muchas de las características del mismo, al final se hicieron las pruebas en un entorno real, pero, al ser una bocina de pobre respuesta, los resultados no fueron los mejores, a continuación una imagen de las pruebas reales.



Ilustración 41. Pruebas finales con el sistema Braillinox

Al final se decidió integrar una diadema Dynex, para integrar un audio de alta fidelidad con un alto baudaje y una captura y salida digital del audio, y es con lo que finalmente se realizó la programación y el diseño.

Se decidió mejorar un poco la programación inicial simplemente introduciendo todas las letras posibles, entre ellas la “á” la “ñ” y muchas de las letras necesarias para que los individuos tengan un correcto proceso y pueda familiarizarse con cada una de las posibles letras a mostrar usando sólo una cajetín, a continuación se muestra la interfaz gráfica con todas las letras y que será de utilidad cuando necesitemos para generar el *script* que realicemos en un lenguaje de medio nivel.

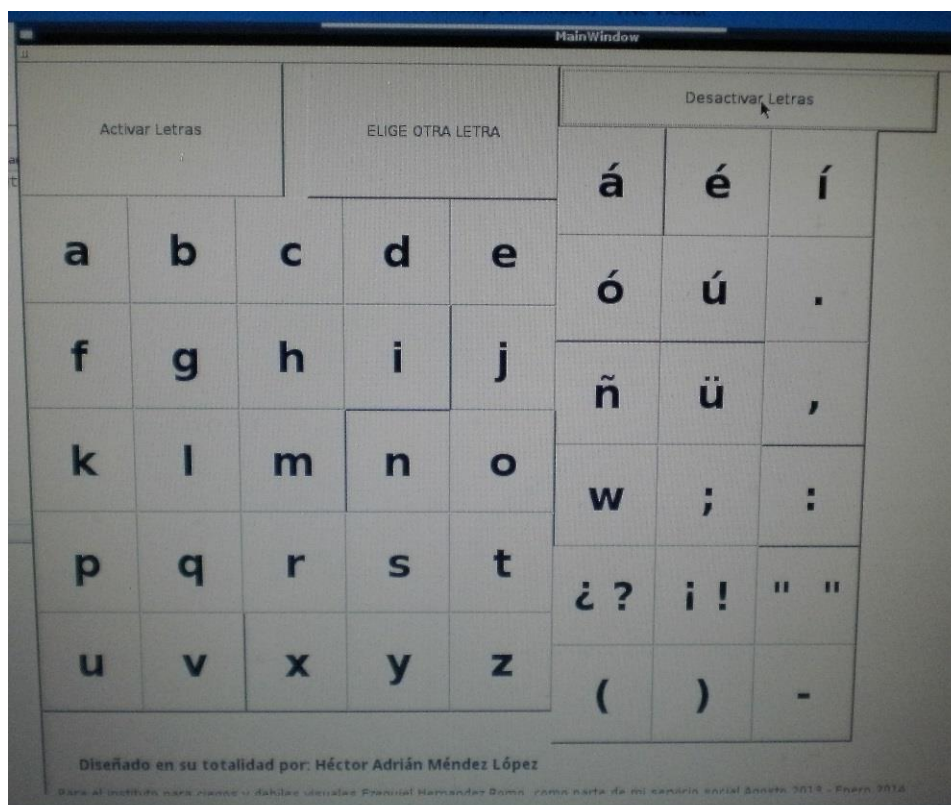


Ilustración 42. Interfaz gráfica de Braillinox a nivel Residencia profesional.

El código de esta interfaz es mucho más extenso, tanto que sólo el simple control de los GPIO de todos los botones tiene una longitud de 438 líneas y cuyo código se muestra en el Anexo 3.

Ya con el código de todas las letras que se pueden mostrar nos resultara más fácil construir el *script* para la programación.

Con todo junto, se prosiguió a establecer un pseudocódigo de todo lo que se debe hacer para que el sistema pueda procesar audio. Los pseudopasos son mostrados en la siguiente ilustración.

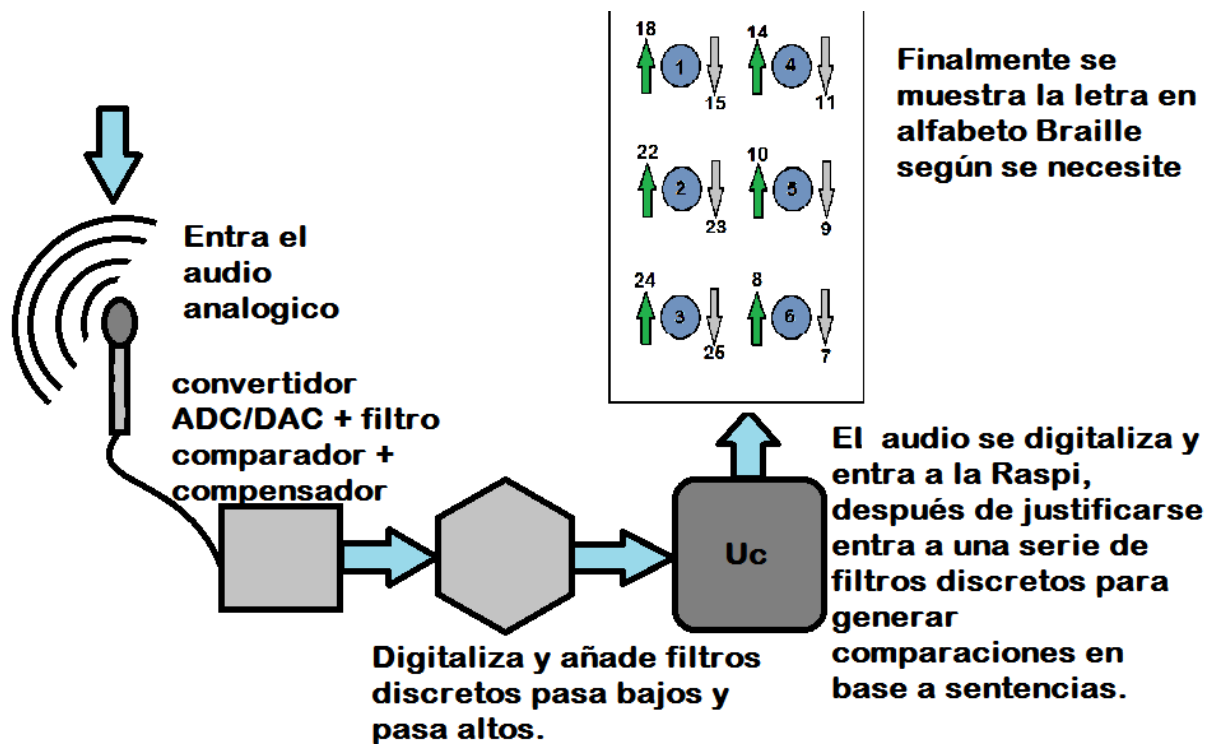


Ilustración 43. Proceso de adquisición de audio.

Finalmente esta es la parte medular del desarrollo, mediante la adquisición de audio analógico generar una plataforma que mediante el procesamiento de señales discretas logre generar una letra pero en un alfabeto táctil como lo es el Braille, esto se logró realizando una instalación de paquetería muy avanzada y se muestra lo requerido en los siguientes pasos enlistados en orden de ejecución.

1. Primero se instala Wring PI y se deben hacer scripts con leds, esto sólo se usará como prueba en caso de que funcione el dispositivo correctamente.
2. Luego, por defecto la Raspberry pi considera como salida de audio uno a la *analog mixer* en la salida hembra de 3.5 mm y algunas veces en audio digital por el canal 1 del HDMI, entonces se tiene que configurar el micrófono en el canal 0, para que sea detectado automáticamente al momento de iniciar el sistema y ya no se tenga que configurarlo nuevamente, esto se logra mediante las instrucciones en la LXterminal:


```
sudo nano /etc/modprobe.d/alsa-base.conf
```

Modificando la linea:

```
# Keep snd-usb-audio from being loaded as first soundcard
```

```
options snd-usb-audio index=-2
```

Y cambiar.

```
index=-2 a index=0
```

Después.

```
sudo alsa force-reload
```

Esto hará que se cargue por defecto el dispositivo *USB* en el sistema como audio de canal cero.

Con el comando

```
cat /proc/asound/cards
```

Se verá el microfono tiene que quedar en el canal 0

[30]

3. Para suministrar a la Raspberry Pi de todo el soporte de software necesario es un requisito instalar GNU/Bison y libasound2-dev, mediante la siguiente línea de texto en la terminal.

Esto se debe hacer porque Pocketsphinx 0.7 y 0.8 no funciona con *PulseAudio*, sólo funciona con *Alsamixer*, entonces se instala libasound2-dev para darle el soporte necesario a *Alsamixer* para funcionar.

```
apt-get install bison
```

```
apt-get install libasound2-dev
```

[31]

4- Ahora se prosigue a Instalar Sphinxbase y Pocketsphinx ya que son software necesario para el reconocimiento de voz y que auxiliaran a los filtros digitales y la síntesis de voz, como ya se mencionó en el marco teórico Pocketsphinx. Es un conjunto de paquetería de software que auxiliará el proceso de reconocimiento de voz en microistemas con arquitectura ARM, por eso Pocketsphinx fue la opción obvia al momento de intentar procesar audio en la Raspi.

Primero se obtiene por terminal el código fuente de BaseSphinx.

```
wget http://downloads.sourceforge.net/project/cmusphinx/sphinxbase/0.8/sphinxbase-0.8.tar.gz
```

Se descomprime.

```
tar -xvf sphinxbase-0.8.tar.gz
```

Se va a su directorio.

```
cd sphinxbase-0.8
```

Se configuró con el GCC disponible.

```
./configure --enable-fixed
```

Se compiló.

```
Make
```

Y, finalmente se instala el software.

```
sudo make install
```

Ahora para instalar Pocket Sphinx se siguen los mismos pasos recién ejecutados

```
wget http://sourceforge.net/projects/cmusphinx/files/pocketsphinx/0.8/pocketsphinx-0.8.tar.gz
```

```
tar -xvf pocketsphinx-0.8.tar.gz
```

```
cd pocketsphinx-0.8
```

```
./configure --enable-fixed
```

```
make clean
```

```
make
```

```
sudo make install
```

[32]

5- Ya con la paquetería de software correctamente instalada, es todo lo que se consideró en ese momento necesario para hacer el reconocimiento de voz, entonces se tuvieron que aplicar ciertas configuraciones extra como los son Exportar la LD_Library, la ventaja de que sea un sistema basado en Debian radica en que cualquier problema presentado básicamente ya le ha pasado a otro usuario, debido a todo el soporte que se le puede ofrecer al usuario final, hoy en día es muy fácil usar Debian gracias a la comunidad que cada día se encarga de engrandecer los sistemas operativos libres.

En este caso una librería genera muchos conflictos, así que es mejor buscar el soporte del software, gracias a esto la comunidad logró ofrecer una solución viable en dos pasos y son los siguientes.

```
export LD_LIBRARY_PATH=/usr/local/lib
```

```
export PKG_CONFIG_PATH=/usr/local/lib/pkgconfig
```

[33]

6- Como todo principio en un lenguaje para que exista comunicación, ambas partes deben tener en común un mismo idioma, desde el fundamento de los conceptos se sabe que un idioma es un sistema

signos que utilizan dos partes para comunicarse, en este caso, se sabe que se puede definir como idioma un lenguaje que entiendan las dos partes de la maquina entonces se debe crear un diccionario con las palabras necesarias para que la comunicación sea correcta, el proceso para hacer esto es, enlistar en un documento de texto las palabras que deseamos reconocer, cada palabra ocupará un renglón, nuestro lenguaje fue tan extenso como 11 palabras, unas más sencillas que otras, en este caso es muy difícil generar las sentencias del lenguaje de forma manual, así que, se sigue el siguiente procedimiento:

Primero se genera un archivo de texto (extensión .txt), en él se capturaron las palabras que se quisieron que el sistema reconozca, ejemplo, a, b, c, Hola, Adiós, y cada palabra será separada por un salto de linea con retorno de carro como se ejemplifica a continuación.

A

B

C

Hola

Adiós

Una vez generado el archivo de texto se sube a la página “<http://www.speech.cs.cmu.edu/tools/lmtool-new.html>”, la aplicación no tarda mucho en devolver un empaquetado con los archivos necesarios para generar un lenguaje, en este caso todo el empaquetado tuvo un peso menor a 2kB así que es muy ligero, ya con el lenguaje construido se prosigue a ejecutarlo.

[34]

```
pocketsphinx_continuous -lm /home/pi/TAR3483/3483.lm -dict  
/home/pi/Descargas/TAR3483/3483.dic -hmm /home/pi/pocketsphinx-
```

```
0.8/model/hmm/en_US/hub4wsj_sc_8k | perl -ne 'system("/bin/bash
/usr/local/bin/prueba_sphinx.sh \"$_\"")'
```

Ahora se describirá brevemente la función de cada instrucción, primero al definir un lenguaje hay que detallar un diccionario por eso en la sentencia se ejecuta la instrucción “-dict”, que es un diccionario de empaquetado genérico con soporte para el idioma inglés, luego “-lm” para describir un modelo matemático de lenguaje, y esto se debe relacionar con el diccionario de Pocketsphinx, para esto se usa “-hmm”, finalmente a todo eso se añade un script, el script con la sintaxis siguiente.

```
| perl -ne 'system("/bin/bash /la dirección de tu script.sh \"$_\"")'
```

El script está muy sencillo, es básicamente lo que se había hecho en el programa antiguo, pero ahora directamente en la terminal, con un pequeño plus, se utilizará menos energía para hacer funcionar todos los actuadores lineales, ya que con el tiempo se tuvo más experiencia para poder manejar los tiempos de retardo en la terminal de Linux.

El modo en el que se programa el script es en instrucciones por terminal y el script tiene una estructura de casos, cuando un caso se introduce y ese es igualado a lo introducido por la terminal en forma de línea de texto proveniente del audio, al compararse genera una acción en caso de que algo pase.

Esta es la estructura básica del script.

```
#!/bin/bash
echo se recibe $@
echo
se recibe $@ >> /tmp/prueba_reconocimiento.log
for n in $(echo $@);
do
case $n in
HOLA) #HOLA
```

```
sudo echo 23 >/sys/class/gpio/export
sudo chmod 777 -R /sys/class/gpio/gpio23
sudo echo out >/sys/class/gpio/gpio23/direction
sudo echo 25 >/sys/class/gpio/export
sudo chmod 777 -R /sys/class/gpio/gpio25
sudo echo out >/sys/class/gpio/gpio25/direction
sudo echo 7 >/sys/class/gpio/export
sudo chmod 777 -R /sys/class/gpio/gpio7
sudo echo out >/sys/class/gpio/gpio7/direction
sudo echo 8 >/sys/class/gpio/export
sudo chmod 777 -R /sys/class/gpio/gpio8
sudo echo out >/sys/class/gpio/gpio8/direction
sudo echo 9 >/sys/class/gpio/export
sudo chmod 777 -R /sys/class/gpio/gpio9
sudo echo out >/sys/class/gpio/gpio9/direction
sudo echo 10 >/sys/class/gpio/export
sudo chmod 777 -R /sys/class/gpio/gpio10
sudo echo out >/sys/class/gpio/gpio10/direction
sudo echo 11 >/sys/class/gpio/export
sudo chmod 777 -R /sys/class/gpio/gpio11
sudo echo out >/sys/class/gpio/gpio11/direction
sudo echo 14 >/sys/class/gpio/export
sudo chmod 777 -R /sys/class/gpio/gpio14
sudo echo out >/sys/class/gpio/gpio14/direction
sudo echo 15 >/sys/class/gpio/export
sudo chmod 777 -R /sys/class/gpio/gpio15
sudo echo out >/sys/class/gpio/gpio15/direction
```

```

sudo echo 18 >/sys/class/gpio/export
sudo chmod 777 -R /sys/class/gpio/gpio18
sudo echo out >/sys/class/gpio/gpio18/direction
sudo echo 22 >/sys/class/gpio/export
sudo chmod 777 -R /sys/class/gpio/gpio22
sudo echo out >/sys/class/gpio/gpio22/direction
sudo echo 24 >/sys/class/gpio/export
sudo chmod 777 -R /sys/class/gpio/gpio24
sudo echo out >/sys/class/gpio/gpio24/direction
;;

```

A) #A

```

sudo mplayer /root/audiobraillinux/audio1.ogg
sudo echo 1 >/sys/class/gpio/gpio18/value
mplayer /root/audiobraillinux/audio47.ogg
echo 0 >/sys/class/gpio/gpio18/value
sleep 5s
echo 1 >/sys/class/gpio/gpio7/value
echo 1 >/sys/class/gpio/gpio9/value
echo 1 >/sys/class/gpio/gpio11/value
echo 1 >/sys/class/gpio/gpio15/value
echo 1 >/sys/class/gpio/gpio23/value
echo 1 >/sys/class/gpio/gpio25/value
sleep 1s
echo 0 >/sys/class/gpio/gpio7/value
echo 0 >/sys/class/gpio/gpio9/value
echo 0 >/sys/class/gpio/gpio11/value
echo 0 >/sys/class/gpio/gpio15/value

```

echo 0 >/sys/class/gpio/gpio23/value

echo 0 >/sys/class/gpio/gpio25/value

;;

B) #B

mplayer /root/audiobraillinux/audio2.ogg

echo 1 >/sys/class/gpio/gpio18/value

echo 1 >/sys/class/gpio/gpio22/value

mplayer /root/audiobraillinux/audio48.ogg

echo 0 >/sys/class/gpio/gpio18/value

echo 0 >/sys/class/gpio/gpio22/value

sleep 5s

echo 1 >/sys/class/gpio/gpio7/value

echo 1 >/sys/class/gpio/gpio9/value

echo 1 >/sys/class/gpio/gpio11/value

echo 1 >/sys/class/gpio/gpio15/value

echo 1 >/sys/class/gpio/gpio23/value

echo 1 >/sys/class/gpio/gpio25/value

sleep 1s

echo 0 >/sys/class/gpio/gpio7/value

echo 0 >/sys/class/gpio/gpio9/value

echo 0 >/sys/class/gpio/gpio11/value

echo 0 >/sys/class/gpio/gpio15/value

echo 0 >/sys/class/gpio/gpio23/value

echo 0 >/sys/class/gpio/gpio25/value

;;

C) #C

mplayer /root/audiobraillinux/audio3.ogg


```

echo 1 >/sys/class/gpio/gpio18/value
echo 1 >/sys/class/gpio/gpio14/value
mplayer /root/audiobraillinux/audio49.ogg
echo 0 >/sys/class/gpio/gpio18/value
echo 0 >/sys/class/gpio/gpio14/value
sleep 5s
echo 1 >/sys/class/gpio/gpio7/value
echo 1 >/sys/class/gpio/gpio9/value
echo 1 >/sys/class/gpio/gpio11/value
echo 1 >/sys/class/gpio/gpio15/value
echo 1 >/sys/class/gpio/gpio23/value
echo 1 >/sys/class/gpio/gpio25/value
sleep 1s
echo 0 >/sys/class/gpio/gpio7/value
echo 0 >/sys/class/gpio/gpio9/value
echo 0 >/sys/class/gpio/gpio11/value
echo 0 >/sys/class/gpio/gpio15/value
echo 0 >/sys/class/gpio/gpio23/value
echo 0 >/sys/class/gpio/gpio25/value
;;

```

D) #D

```

mplayer /root/audiobraillinux/audio4.ogg
echo 1 >/sys/class/gpio/gpio18/value
echo 1 >/sys/class/gpio/gpio14/value
echo 1 >/sys/class/gpio/gpio10/value
mplayer /root/audiobraillinux/audio50.ogg
echo 0 >/sys/class/gpio/gpio18/value

```

```

echo 0 >/sys/class/gpio/gpio14/value
echo 0 >/sys/class/gpio/gpio10/value
sleep 5s
echo 1 >/sys/class/gpio/gpio7/value
echo 1 >/sys/class/gpio/gpio9/value
echo 1 >/sys/class/gpio/gpio11/value
echo 1 >/sys/class/gpio/gpio15/value
echo 1 >/sys/class/gpio/gpio23/value
echo 1 >/sys/class/gpio/gpio25/value
sleep 1s
echo 0 >/sys/class/gpio/gpio7/value
echo 0 >/sys/class/gpio/gpio9/value
echo 0 >/sys/class/gpio/gpio11/value
echo 0 >/sys/class/gpio/gpio15/value
echo 0 >/sys/class/gpio/gpio23/value
echo 0 >/sys/class/gpio/gpio25/value
;;

```

E) #E

```

mplayer /root/audiobraillinux/audio5.ogg
echo 1 >/sys/class/gpio/gpio18/value
echo 1 >/sys/class/gpio/gpio10/value
mplayer /root/audiobraillinux/audio51.ogg
echo 0 >/sys/class/gpio/gpio18/value
echo 0 >/sys/class/gpio/gpio10/value
sleep 5s
echo 1 >/sys/class/gpio/gpio7/value
echo 1 >/sys/class/gpio/gpio9/value

```

```
echo 1 >/sys/class/gpio/gpio11/value
echo 1 >/sys/class/gpio/gpio15/value
echo 1 >/sys/class/gpio/gpio23/value
echo 1 >/sys/class/gpio/gpio25/value
sleep 1s
echo 0 >/sys/class/gpio/gpio7/value
echo 0 >/sys/class/gpio/gpio9/value
echo 0 >/sys/class/gpio/gpio11/value
echo 0 >/sys/class/gpio/gpio15/value
echo 0 >/sys/class/gpio/gpio23/value
echo 0 >/sys/class/gpio/gpio25/value
;;
```

F) #F

```
mplayer /root/audiobraillinux/audio6.ogg
echo 1 >/sys/class/gpio/gpio22/value
echo 1 >/sys/class/gpio/gpio14/value
echo 1 >/sys/class/gpio/gpio18/value
mplayer /root/audiobraillinux/audio52.ogg
echo 0 >/sys/class/gpio/gpio22/value
echo 0 >/sys/class/gpio/gpio14/value
echo 0 >/sys/class/gpio/gpio18/value
sleep 5s
echo 1 >/sys/class/gpio/gpio7/value
echo 1 >/sys/class/gpio/gpio9/value
echo 1 >/sys/class/gpio/gpio11/value
echo 1 >/sys/class/gpio/gpio15/value
echo 1 >/sys/class/gpio/gpio23/value
```

```

echo 1 >/sys/class/gpio/gpio25/value
sleep 1s
echo 0 >/sys/class/gpio/gpio7/value
echo 0 >/sys/class/gpio/gpio9/value
echo 0 >/sys/class/gpio/gpio11/value
echo 0 >/sys/class/gpio/gpio15/value
echo 0 >/sys/class/gpio/gpio23/value
echo 0 >/sys/class/gpio/gpio25/value
;;

```

G) #G

```

mplayer /root/audiobraillinux/audio7.ogg
echo 1 >/sys/class/gpio/gpio18/value
echo 1 >/sys/class/gpio/gpio14/value
echo 1 >/sys/class/gpio/gpio22/value
echo 1 >/sys/class/gpio/gpio10/value
mplayer /root/audiobraillinux/audio53.ogg
echo 0 >/sys/class/gpio/gpio18/value
echo 0 >/sys/class/gpio/gpio14/value
echo 0 >/sys/class/gpio/gpio22/value
echo 0 >/sys/class/gpio/gpio10/value
sleep 5s
echo 1 >/sys/class/gpio/gpio7/value
echo 1 >/sys/class/gpio/gpio9/value
echo 1 >/sys/class/gpio/gpio11/value
echo 1 >/sys/class/gpio/gpio15/value
echo 1 >/sys/class/gpio/gpio23/value
echo 1 >/sys/class/gpio/gpio25/value

```

```

sleep 1s

echo 0 >/sys/class/gpio/gpio7/value
echo 0 >/sys/class/gpio/gpio9/value
echo 0 >/sys/class/gpio/gpio11/value
echo 0 >/sys/class/gpio/gpio15/value
echo 0 >/sys/class/gpio/gpio23/value
echo 0 >/sys/class/gpio/gpio25/value

;;

H) #H

mplayer /root/audiobraillinux/audio8.ogg
echo 1 >/sys/class/gpio/gpio18/value
echo 1 >/sys/class/gpio/gpio10/value
echo 1 >/sys/class/gpio/gpio22/value
mplayer /root/audiobraillinux/audio54.ogg
echo 0 >/sys/class/gpio/gpio18/value
echo 0 >/sys/class/gpio/gpio10/value
echo 0 >/sys/class/gpio/gpio22/value

sleep 5s

echo 1 >/sys/class/gpio/gpio7/value
echo 1 >/sys/class/gpio/gpio9/value
echo 1 >/sys/class/gpio/gpio11/value
echo 1 >/sys/class/gpio/gpio15/value
echo 1 >/sys/class/gpio/gpio23/value
echo 1 >/sys/class/gpio/gpio25/value

sleep 1s

echo 0 >/sys/class/gpio/gpio7/value
echo 0 >/sys/class/gpio/gpio9/value

```

```

echo 0 >/sys/class/gpio/gpio11/value
echo 0 >/sys/class/gpio/gpio15/value
echo 0 >/sys/class/gpio/gpio23/value
echo 0 >/sys/class/gpio/gpio25/value
;;
ADIOS) #ADIOS
sudo echo 7 >/sys/class/gpio/unexport
sudo echo 8 >/sys/class/gpio/unexport
sudo echo 9 >/sys/class/gpio/unexport
sudo echo 10 >/sys/class/gpio/unexport
sudo echo 11 >/sys/class/gpio/unexport
sudo echo 14 >/sys/class/gpio/unexport
sudo echo 15 >/sys/class/gpio/unexport
sudo echo 18 >/sys/class/gpio/unexport
sudo echo 22 >/sys/class/gpio/unexport
sudo echo 23 >/sys/class/gpio/unexport
sudo echo 24 >/sys/class/gpio/unexport
sudo echo 25 >/sys/class/gpio/unexport
mplayer /root/audiobraillinux/audio44.ogg
;;
*)
echo comando $n no fuera de lenguaje
;;
esac
done

```

Este script es muy básico pero muestra su utilidad, y tiene la ventaja de que al ser sólo instrucciones por terminal es muy rápida y confiable la ejecución.

CAPÍTULO 6

Resultados, planos, gráficas, prototipos y programas

El archivo xxxx.dic genera todo el soporte de lenguaje posible para la descripción fonética.

```
A      AH
A(2)   EY
ADIOS  AA D IY OW S
B      B IY
C      S IY
D      D IY
E      IY
F      EH F
G      JH IY
H      EY CH
HOLA   HH OW L AH
```

Finalmente el archivo xxxx.lm que describe el modelo del lenguaje es.

```
\data\
ngram 1=14
ngram 2=23
```

ngram 3=12

\1-grams:

-0.7911 </s> -0.3010

-0.7911 <s> -0.2244

-1.8325 A -0.2244

-1.8325 ADIOS -0.2244

-1.8325 B -0.2244

-1.8325 C -0.2244

-1.8325 D -0.2244

-1.8325 E -0.2244

-1.8325 F -0.2244

-1.8325 G -0.2244

-1.8325 H -0.2244

-1.8325 HOLA -0.2244

\2-grams:

-1.3424 <s> A 0.0000

-1.3424 <s> ADIOS 0.0000

-1.3424 <s> B 0.0000

-1.3424 <s> C 0.0000

-1.3424 <s> D 0.0000

-1.3424 <s> E 0.0000

-1.3424 <s> F 0.0000

-1.3424 <s> G 0.0000

-1.3424 <s> H 0.0000

-1.3424 <s> HOLA 0.0000

-0.3010 A </s> -0.3010

-0.3010 ADIOS </s> -0.3010

-0.3010 B </s> -0.3010

-0.3010 C </s> -0.3010

-0.3010 D </s> -0.3010

-0.3010 E </s> -0.3010

-0.3010 F </s> -0.3010

-0.3010 G </s> -0.3010

-0.3010 H </s> -0.3010

-0.3010 HOLA </s> -0.3010

\3-grams:

-0.3010 <s> A </s>

-0.3010 <s> ADIOS </s>

-0.3010 <s> B </s>

-0.3010 <s> C </s>

-0.3010 <s> D </s>

-0.3010 <s> E </s>

-0.3010 <s> F </s>

-0.3010 <s> G </s>

-0.3010 <s> H </s>

-0.3010 <s> HOLA </s>

\end\

Aquí, en la imagen 44 se ilustra cómo quedó la configuración de audio del sistema.

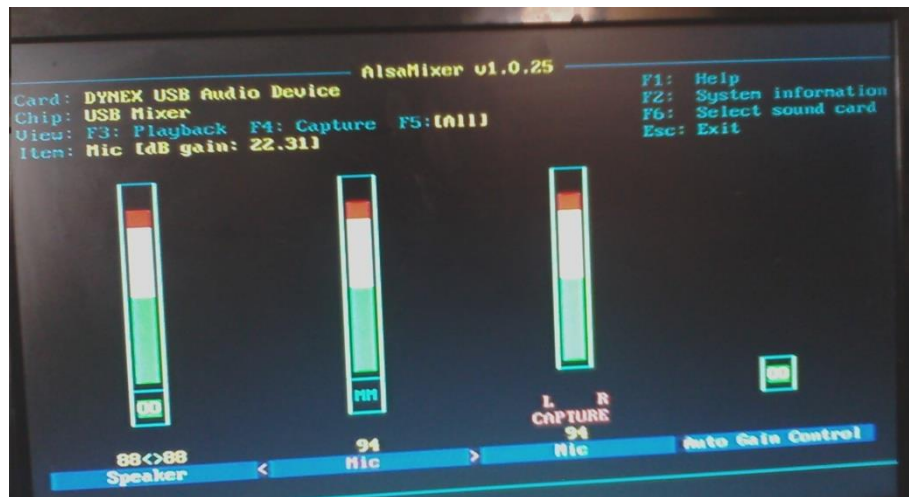


Ilustración 44. Se muestra como quedo la diadema USB DYNEX como predeterminada en el sistema.

Finalmente en la ilustración 45 se muestra el sistema ya integrado a la diadema y en uso.



Ilustración 45. Braillinox con la diadema DYNEX DX-840.

CAPÍTULO 7

Conclusiones y recomendaciones.

En general la residencia fue un proceso increíble, se dieron cursos sobre Arduino y se dieron pláticas sobre los problemas profesionales al desarrollar Brailinox y como se solucionaron.

El problema más grande por mucho es que el sistema “*overrunaba*” y no podía identificar ninguna letra, por esto, se tuvo que instalar mucha paquetería y al final se decidió instalar una versión inferior de Pocketsphinx, es decir, Pocketsphinx-0.7, otro detalle notable es que se usó el diccionario (-dict) en inglés de Pocketsphinx ya que lamentablemente el modelo en español está en desarrollo y el Alfa que está disponible es tan inestable en la arquitectura ARM que compromete de forma significativa a la Raspberry Pi.

Hay varios aspectos que se tuvieron que trabajar sobre todo con el ajuste del micrófono ya que con cualquier cosa reaccionaba y aventaba basura, entonces se trabajó con los ajustes en las ganancias y ya por fin se pudo trabajar.

El resultado no permitió generar un programa integral de enseñanza ya que se necesita un modelo de diccionario en español para poder hacer factible esta meta, pero el trabajo se queda abierto en una posterior investigación darle una conclusión a este proyecto, que ya en general se encuentra en una etapa madura del desarrollo.

Recomiendo en lo personal ampliamente a CGTRATEC y su alianza con el LABSOL para hacer residencias profesionales y estancias, ya que, el conocimiento es libre, en un entorno inteligente y ecológico con personas que siempre están dispuestas a ayudar y tienen toda la disponibilidad para resolver las inquietudes que se cuestionen, además de que al ser un laboratorio propiedad del COZCYT pues tiene alianzas con muchas instancias de gobierno, entonces es muy posible que los proyectos que se

desarrollen en las instalaciones de COZCYT-LABSOL se implementen en una instancia estatal.

Finalmente, respondiendo a la pregunta que se planteó en la introducción. ¿Es posible incorporar nuevas tecnologías en materia de Hardware y de Software libre para agilizar el aprendizaje en personas con ceguera o debilidad visual?

La respuesta es contundente: SI, pero el desarrollo de este tipo de tecnologías no es barato ni tampoco algo que enseñen en un salón de clases, por lo cual los proyectos deben estar muy cuidados y preferentemente elaborados por grupos de especialistas enfocados a áreas específicas, los procesos, claro, como en toda nueva tecnología, van a ser largos pero nuestro papel como parte de la sociedad del conocimiento es retribuir un poco de lo mucho que la sociedad nos ha dado y que mejor modo de hacerlo, que ayudando a un sector vulnerable de la sociedad.

Por eso mismo se pretende dejar esta investigación abierta a las nuevas plataformas de hardware y software de desarrollo embebido, así como a las mentes jóvenes y entusiastas capaces de hacer posible algo que hace unos años, sólo parecería un sueño.

Referencias Electrónicas.

[1] “Estadísticas sobre personas con ceguera y débilidad visual” (n. d.) Extraída el 2/V/2014 desde <http://www.iapb.org/advocacy/who-action-plan/focus-challenges>

[2] “Reforma de ley al protocolo de inclusión educativa a nivel federal” (n. d.) Extraída el 22/V/ 2014 desde http://www.sep.gob.mx/work/models/sep1/Resource/558c2c24-0b12-4676-ad90-8ab78086b184/ley_general_inclusion_personas_discapacidad.htm

[3] “Extensa explicación sobre la Reingeniería de producto” (n. d.) extraída el 15/V/2014 desde <http://ciclolimite.com/2011/08/05/%C2%BFque-es-la-reingenieria-de-producto/>

[4] “Qué elementos debe tener un sistema para considerarse un sistema Mecatrónico” (n. d.) Extraída el 6/V/ 2014 desde <http://sistemamecatronico.blogspot.mx/>

[5] “Fuente conmutada” (n. d.) Extraída el 6/III/2014 desde http://es.wikipedia.org/wiki/Fuente_conmutada

[6] “Qué es un actuador Lineal eléctrico” (n. d.) Extraída el 6/IV/2014 desde <http://www.linak.es/about/?id3=4283>

[7] “Todas las características de Raspberry PI” (n. d.) Extraída el 2/III/ 2014 desde <http://thepihut.com/pages/the-raspberry-pi>

[8] "Filosofías y uso del Software Libre" (n. d.) Extraída el 10/II/ 2014 desde <http://www.fsf.org/about/>

[9] "Todo sobre el Hardware Libre" (n. d.) Extraída el 12/III/2014 desde http://p2pfoundation.net/Free_Hardware_Design

[10] "Raspbian, el sistema operativo libre para Raspberry pi" (n. d.) Extraída el 4/III/2014 desde <http://www.raspbian.org/RaspbianInstaller>

[11] "Qué es un GPIO" (n. d.) Extraída el 13/III/2014 desde <http://www.raspberrypi.org/using-the-gpio/>

[12] "Las propiedades del circuito integrado L293B" (n. d.) Extraída el 13/III/2014 desde, [en línea] dirección URL:<http://prezi.com/8aznbsvshbv3/circuito-integrado-l293b-doble-puente-h/>

[13] "Información sobre el Cloruro férrico" (n. d.) Extraída el 21/III/2014 desde [http://es.wikipedia.org/wiki/Cloruro_de_hierro_\(III\)](http://es.wikipedia.org/wiki/Cloruro_de_hierro_(III))

[14] "Instrumentación Electrónica parte sobre Placas Fenolicas" (n. d.) Extraída el 22/III/2014 desde <http://instrumentacion-electronica.globered.com/categoria.asp?idcat=43>

[15] "Cliente SSH, telnet, serial PuTTY" (n. d.) Extraída el 31/III/2014 desde <http://es.wikipedia.org/wiki/PuTTY>

[16] "Cliente VNC, Real VNC y configuraciones básicas" (n. d.) Extraída el 7/IV/2014 desde <http://es.wikipedia.org/wiki/RealVNC>

[17] "NOOBS, el gestor de arranque de sistemas operativos en Raspberry Pi" (n. d.) Extraída el 13/V/2014 desde <http://comohacer.eu/noobs-multi-boot-en-tu-raspberry-pi/>

[18] " Programación con Qt4, software diseñado por Intel " (n. d.) Extraída el 9/IV/2014 desde http://es.wikibooks.org/wiki/Programaci%C3%B3n_con_Qt4

[19] "IDE QT creator" (n. d.) Extraída el 12/IV/2014 desde http://es.wikipedia.org/wiki/Qt_Creator

[20] "TightVNC client-server" (n. d.) Extraída el 2/IV/2014 desde <http://www.tightvnc.com/>

[21] "MPlayer, reproductor de audio y video por consola" (n. d.) Extraída el 14/V/2014 desde <http://es.wikipedia.org/wiki/MPlayer>

[22] "CMU Sphinx" (n. d.) Extraída el 28/IV/2014 desde cmusphinx.sourceforge.net/

[23] "Wiringpi y su importancia para los GPIO" (n. d.) Extraída el 28/IV/2014 desde <https://projects.drogon.net/raspberry-pi/wiringpi/>

[24] "GNU BISON" (n. d.) Extraída el 26/III/2014 desde http://es.wikipedia.org/wiki/GNU_Bison

[25] "Lenguaje de programación PERL" (n. d.) Extraída el 1/V/2014 desde <http://es.wikipedia.org/wiki/Perl>

[26] "Fairchild semiconductors Asistente para el diseño de fuentes conmutadas partiendo de una necesidad" (n. d.) Extraída el 8/V/2014 desde <http://www.fairchildsemi.com/support/design-tools/>

[27] "Artículo sobre el diseño de fuentes conmutadas" (n. d.) Extraída el 9/V/2014 desde <http://www.neoteo.com/disena-tus-fuentes-conmutadas-en-linea/>

[28] "Introducción al manejo de los GPIO de la RasPi" (n. d.) Extraída el 12/V/2014 desde <https://raspberrypiwonderland.wordpress.com/tag/gpio/>

[29] "Crear aplicaciones con QtCreator y configurar el depurador y los compiladores" (n. d.) Extraída el 13/V/2014 desde <http://colab-raspberry.blogspot.mx/2013/11/raspberry-pi-qt-creator-controlando.html>

[30] “Página web que encarga de documentar los bugs de debian, en esta caso la usaremos para configurar la tarjeta de sonido de nuestra Raspberry Pi con Raspbian” (n. d.) Extraída el 1/III/2014 desde <http://www.esdebian.org/wiki/configurar-tarjeta-audio>

[31] “Narrativa de experiencias en el reconocimiento de voz en la Raspberry pi” (n. d.) Extraída el 14/V/2014 desde <https://sites.google.com/site/observing/Home/speech-recognition-with-the-raspberry-pi>

[32] “Artículo sobre la instalación, compilación y configuración básica del software de soporte para el reconocimiento de audio” (n. d.) Extraída el 16/IV/2014 desde <http://codebangers.com/?p=685>

[33] “Tutorial sobre el uso y consideraciones especiales al usar CMUSphinx” (n. d.) Extraída el 18/V/2014 desde <http://cmusphinx.sourceforge.net/wiki/tutorialpocketsphinx>

[34] “Utilidad de Imtool para generar los diccionarios para Pocketsphinx-0.XX (sólo modelos en inglés)” (n. d.) Extraída el 22/V/2014 desde <http://www.speech.cs.cmu.edu/tools/Imtool-new.html>

Índice de ecuaciones

Ecuación 1 . Reacción rédox del cloruro férrico con el cobre.	18
--	----

Índice de Tablas

Tabla 1 . Lista de productos con precio usada en Braillinox en fase de servicio social.	27
Tabla 2 . lista de materiales usados en la fuente diseñada.....	35
Tabla 3 . Lista de palabras usadas en la Ras Pi.....	47

Índice de ilustraciones

Ilustración 1. Organigrama.....	8
Ilustración 2. Actuador lineal calidad Automotriz.....	13
Ilustración 3. Raspberry Pi modelo B.	14
Ilustración 4. Diagrama de conexiones del L293B.....	18
Ilustración 5. Actuador lineal principal.	28
Ilustración 6. Diseño en Ares de la placa unica basada en L293b.	30
Ilustración 7. Render hecho por el asistente Ares de nuestro circuito impreso.	30
Ilustración 8. Circuito a planchar en la placa de baquelita.	31
Ilustración 9. Tablilla fenolica ya impresa, limpia, lista para perforar y montar componentes.	32
Ilustración 10. Vista superior de una fuente similar a la usada en Braillinox.	34
Ilustración 11. Diagrama de la fuente conmutada diseñada en PSWD.	34
Ilustración 12. Matriz cuadrada para el rectángulo generador del alfabeto Braille.	36
Ilustración 13. Parte superior del sistema ya terminada.	37
Ilustración 14. Carcasa externa del sistema Braillinox.	38
Ilustración 15. Medida final ancho 17.5 cm.	38
Ilustración 16. Medida final alto 18 cm.	39
Ilustración 17. Medida final largo 25.5 cm.	40
Ilustración 18. Esquema simbólico de las conexiones internas del dispositivo.	40
Ilustración 19. Vista lateral del sistema.	41
Ilustración 20. Vista lateral 2 del sistema.....	42
Ilustración 21. Vista superior del sistema ya terminado.....	42
Ilustración 22. Numeración del las salidas del puerto GPIO.....	43
Ilustración 23. Esquema simplificado de los GPIO de subida y bajada.	44
Ilustración 24. Pagina de inicio de NOOBS.....	48
Ilustración 25. Interfaz finalmente entregada como servicio social.....	57
Ilustración 26. Diagrama de conexiones.....	61
Ilustración 27. LM380 en la tablilla de prácticas.....	62
Ilustración 28. Circuito terminado.	62
Ilustración 29. Diseño en PCB y ordenado de los componentes.	63
Ilustración 30. Render de la vista superior.	64
Ilustración 31. Renderizado de la vista inferior del amplificador.....	65
Ilustración 32. Vistas de LabSol y CGTRATEC.....	65
Ilustración 33. Diseño antes de ser planchado sobre el cobre.	66

Ilustración 34. Diseño impreso sobre el cobre.....	67
Ilustración 35. Tablilla en el cloruro férrico.	68
Ilustración 36. Tablilla ya impresa y recién limpiada.	68
Ilustración 37. Taladro con el que se barrenó la tablilla	69
Ilustración 38. Tablilla ya barrenada.....	69
Ilustración 39. Amplificador terminado vista superior.	70
Ilustración 40. Amplificador terminado vista inferior	71
Ilustración 41. Pruebas finales con el sistema Braillinox.....	72
Ilustración 42. Interfaz gráfica de Braillinox a nivel Residencia profesional.....	73
Ilustración 43. Proceso de adquisición de audio.....	74
Ilustración 44. Se muestra como quedo la diadema USB DYNEX como predeterminada en el sistema.	92
Ilustración 45. Braillinox con la diadema DYNEX DX-840.....	93

ANEXOS

Anexo 1 – Características de la diadema finalmente seleccionada.

USB device details C-Media Electronics Inc. DYNEX USB Audio Device

- Dispositivo compuesto USB

File: \\??\\USB#VID_0D8C&PID_0139#5&dc861b&0&1#{a5dcbf10-6530-11d2-901f-00c04fb951ed}

Driver key name: {36fc9e60-c465-11cf-8056-444553540000}\\0034

Device instance ID: USB\\VID_0D8C&PID_0139\\5&DC861B&0&1

Device hardware ID: USB\\VID_0D8C&PID_0139&REV_0100

Device service: usbccgp

Device friendly name:

Device location: Port_#0001.Hub_#0001

Device physical object name: \\Device\\USBPDO-6

Device descriptor

USB: 0x0110

Device class: 0x00

Device subclass: 0x00

Device protocol: 0x00

Max packet size: 0x08 (8)

Vendor: 0x0D8C (C-Media Electronics, Inc.)

Product ID: 0x0139

Device ID: 0x0100

Manufacturer: 0x01 (C-Media Electronics Inc.)

Product: 0x02 (DYNEX USB Audio Device)

Serial number ID: 0x00

Number of configurations: 1

Connection status: Device connected

Current config value: 0x01

Device bus speed: Full (USB 1.1, 12 Mbit/s)

Device address: 0x02

Open pipes: 1

Pipe information

Endpoint address: 0x87, Input

Transfer type: Interrupt

Max packet size: 0x0004 (4)

Interval: 0x02 (2)

Configuration descriptor

Total length: 0x00FD

Number of interfaces: 4

Configuration value: 0x01

Configuration ID: 0x00

Attributes: 0x80 (Bus powered)

Max power: 0x32 (100 Ma)

Interface descriptor

Interface number: 0x00

Alternate setting: 0x00

Num of endpoints: 0

Interface class: 0x01 (Audio)

Interface subclass: 0x01 (Control Device)

Interface protocol ID: 0x00

Interface ID: 0x00

Class-Specific AC Interface Header descriptor

Length: 10

Descriptor type: 0x24

Descriptor subtype: 0x01

ADC: 0x0100

Total length: 100

In collection: 2

Interfaces: 1 - 0x01, 2 - 0x02

Input Terminal descriptor

Length: 12

Descriptor type: 0x24

Descriptor subtype: 0x02

Terminal ID: 0x01

Terminal type: 0x0101

Assoc terminal: 0x00

Channels #: 2

Channel config: 0x0003

Channel names: 0

Terminal: 0

Input Terminal descriptor

Length: 12

Descriptor type: 0x24

Descriptor subtype: 0x02

Terminal ID: 0x02

Terminal type: 0x0201

Assoc terminal: 0x00

Channels #: 1

Channel config: 0x0001

Channel names: 0

Terminal: 0

Output Terminal descriptor

Length: 9

Descriptor type: 0x24

Descriptor subtype: 0x03

Terminal ID: 0x06

Terminal type: 0x0301

Assoc terminal: 0x00

Source ID: 9

Terminal: 0

Output Terminal descriptor

Length: 9

Descriptor type: 0x24

Descriptor subtype: 0x03

Terminal ID: 0x07

Terminal type: 0x0101

Assoc terminal: 0x00

Source ID: 8

Terminal: 0

Selector Unit descriptor

Length: 7

Descriptor type: 0x24

Descriptor subtype: 0x05

Anexo 2 – Cronograma de actividades.

Actividades		Febrero				Marzo				Abril				Mayo				Junio	
		S1	S2	S3	S4	S1	S2	S3	S4	S1	S2	S3	S4	S1	S2	S3	S4	S1	S2
Elaboración de anteproyecto	P																		
	R																		
Rediseño de mecanismos	P																		
	R																		
Mejoramiento de Facilidad de ensamble	P																		
	R																		
Diseño de amplificadores	P																		
	R																		
Pruebas de reducción de materiales	P																		
	R																		
Trabajo de software de captura de audio	P																		
	R																		
Captura de audio	P																		
	R																		
Gestión de facilidad de uso	P																		
	R																		
Trabajo con <u>GPIO</u>	P																		
	R																		
Elaboración de bases de datos	P																		
	R																		
Pruebas finales	P																		
	R																		
Análisis costo beneficio	P																		
	R																		
Pruebas con usuarios	P																		
	R																		
Elaboración de documentación	P																		
	R																		

Anexo 3 – Programación básica del primer script

```
#include "mainwindow.h"

#include "ui_mainwindow.h"
MainWindow::MainWindow(QWidget *parent) :
    QMainWindow(parent),
    ui(new Ui::MainWindow)
{
    ui->setupUi(this);
}

MainWindow::~MainWindow()
{
    delete ui;
}

void MainWindow::on_pushButtonstart_clicked()
{
    system("sudo echo 23 >/sys/class/gpio/export");
    system("sudo chmod 777 -R /sys/class/gpio/gpio23 ");
    system("sudo echo out >/sys/class/gpio/gpio23/direction");
    system("sudo echo 25 >/sys/class/gpio/export");
    system("sudo chmod 777 -R /sys/class/gpio/gpio25 ");
    system("sudo echo out >/sys/class/gpio/gpio25/direction");
    system("sudo echo 7 >/sys/class/gpio/export");
    system("sudo chmod 777 -R /sys/class/gpio/gpio7 ");
    system("sudo echo out >/sys/class/gpio/gpio7/direction");
    system("sudo echo 8 >/sys/class/gpio/export");
    system("sudo chmod 777 -R /sys/class/gpio/gpio8 ");
    system("sudo echo out >/sys/class/gpio/gpio8/direction");
    system("sudo echo 9 >/sys/class/gpio/export");
    system("sudo chmod 777 -R /sys/class/gpio/gpio9 ");
    system("sudo echo out >/sys/class/gpio/gpio9/direction");
```

```

        system("sudo echo 10 >/sys/class/gpio/export");
        system("sudo chmod 777 -R /sys/class/gpio/gpio10 ");
        system("sudo echo out >/sys/class/gpio/gpio10/direction");
        system("sudo echo 11 >/sys/class/gpio/export");
        system("sudo chmod 777 -R /sys/class/gpio/gpio11 ");
        system("sudo echo out >/sys/class/gpio/gpio11/direction");
        system("sudo echo 14 >/sys/class/gpio/export");
        system("sudo chmod 777 -R /sys/class/gpio/gpio14 ");
        system("sudo echo out >/sys/class/gpio/gpio14/direction");
        system("sudo echo 15 >/sys/class/gpio/export");
        system("sudo chmod 777 -R /sys/class/gpio/gpio15 ");
        system("sudo echo out >/sys/class/gpio/gpio15/direction");
        system("sudo echo 18 >/sys/class/gpio/export");
        system("sudo chmod 777 -R /sys/class/gpio/gpio18 ");
        system("sudo echo out >/sys/class/gpio/gpio18/direction");
        system("sudo echo 22 >/sys/class/gpio/export");
        system("sudo chmod 777 -R /sys/class/gpio/gpio22 ");
        system("sudo echo out >/sys/class/gpio/gpio22/direction");
        system("sudo echo 24 >/sys/class/gpio/export");
        system("sudo chmod 777 -R /sys/class/gpio/gpio24 ");
        system("sudo echo out >/sys/class/gpio/gpio24/direction");
    }

    void MainWindow::on_pushButtonstop_clicked()
    {
        system("sudo echo 7 >/sys/class/gpio/unexport");
        system("sudo echo 8 >/sys/class/gpio/unexport");
        system("sudo echo 9 >/sys/class/gpio/unexport");
        system("sudo echo 10 >/sys/class/gpio/unexport");
        system("sudo echo 11 >/sys/class/gpio/unexport");
        system("sudo echo 14 >/sys/class/gpio/unexport");
        system("sudo echo 15 >/sys/class/gpio/unexport");
    }

```

```

system("sudo echo 18 >/sys/class/gpio/unexport");
system("sudo echo 22 >/sys/class/gpio/unexport");
system("sudo echo 23 >/sys/class/gpio/unexport");
system("sudo echo 24 >/sys/class/gpio/unexport");
system("sudo echo 25 >/sys/class/gpio/unexport");
system("mplayer /root/audiobraillinox/audio44.ogg");
    }

    void MainWindow::on_pushButtonA_pressed()
    {
system("mplayer /root/audiobraillinox/audio1.ogg");
        system("echo 1 >/sys/class/gpio/gpio18/value");
system("mplayer /root/audiobraillinox/audio47.ogg");
    }

    void MainWindow::on_pushButtonB_pressed()
    {
system("mplayer /root/audiobraillinox/audio2.ogg");
        system("echo 1 >/sys/class/gpio/gpio18/value");
        system("echo 1 >/sys/class/gpio/gpio22/value");
system("mplayer /root/audiobraillinox/audio48.ogg");
    }

    void MainWindow::on_pushButtonC_pressed()
    {
system("mplayer /root/audiobraillinox/audio3.ogg");
        system("echo 1 >/sys/class/gpio/gpio18/value");
        system("echo 1 >/sys/class/gpio/gpio14/value");
system("mplayer /root/audiobraillinox/audio49.ogg");
    }

    void MainWindow::on_pushButtonD_pressed()
    {
system("mplayer /root/audiobraillinox/audio4.ogg");
        system("echo 1 >/sys/class/gpio/gpio18/value");

```

```

        system("echo 1 >/sys/class/gpio/gpio14/value");
        system("echo 1 >/sys/class/gpio/gpio10/value");
        system("mplayer /root/audiobraillinox/audio50.ogg");
    }

    void MainWindow::on_pushButtonE_pressed()
    {
        system("mplayer /root/audiobraillinox/audio5.ogg");
        system("echo 1 >/sys/class/gpio/gpio18/value");
        system("echo 1 >/sys/class/gpio/gpio10/value");
        system("mplayer /root/audiobraillinox/audio51.ogg");
    }

    void MainWindow::on_pushButtonF_pressed()
    {
        system("mplayer /root/audiobraillinox/audio6.ogg");
        system("echo 1 >/sys/class/gpio/gpio22/value");
        system("echo 1 >/sys/class/gpio/gpio14/value");
        system("echo 1 >/sys/class/gpio/gpio18/value");
        system("mplayer /root/audiobraillinox/audio52.ogg");
    }

void MainWindow::on_pushButtonsolenoideabajo_pressed()
    {
        system("echo 1 >/sys/class/gpio/gpio7/value");
        system("echo 1 >/sys/class/gpio/gpio9/value");
        system("echo 1 >/sys/class/gpio/gpio11/value");
        system("echo 1 >/sys/class/gpio/gpio15/value");
        system("echo 1 >/sys/class/gpio/gpio23/value");
        system("echo 1 >/sys/class/gpio/gpio25/value");
    }

    void MainWindow::on_pushButtonG_pressed()
    {
        system("mplayer /root/audiobraillinox/audio7.ogg");
    }

```

```

        system("echo 1 >/sys/class/gpio/gpio18/value");
        system("echo 1 >/sys/class/gpio/gpio14/value");
        system("echo 1 >/sys/class/gpio/gpio22/value");
        system("echo 1 >/sys/class/gpio/gpio10/value");
        system("mplayer /root/audiobraillinox/audio53.ogg");
    }

    void MainWindow::on_pushButtonH_pressed()
    {
        system("mplayer /root/audiobraillinox/audio8.ogg");
        system("echo 1 >/sys/class/gpio/gpio18/value");
        system("echo 1 >/sys/class/gpio/gpio10/value");
        system("echo 1 >/sys/class/gpio/gpio22/value");
        system("mplayer /root/audiobraillinox/audio54.ogg");
    }

    void MainWindow::on_pushButtonI_pressed()
    {
        system("mplayer /root/audiobraillinox/audio9.ogg");
        system("echo 1 >/sys/class/gpio/gpio14/value");
        system("echo 1 >/sys/class/gpio/gpio22/value");
        system("mplayer /root/audiobraillinox/audio55.ogg");
    }

    void MainWindow::on_pushButtonJ_pressed()
    {
        system("mplayer /root/audiobraillinox/audio10.ogg");
        system("echo 1 >/sys/class/gpio/gpio14/value");
        system("echo 1 >/sys/class/gpio/gpio10/value");
        system("echo 1 >/sys/class/gpio/gpio22/value");
    }

    void MainWindow::on_pushButtonK_pressed()
    {
        system("mplayer /root/audiobraillinox/audio11.ogg");
    }

```

```

        system("echo 1 >/sys/class/gpio/gpio18/value");
        system("echo 1 >/sys/class/gpio/gpio24/value");
    }

    void MainWindow::on_pushButtonL_pressed()
    {
        system("mplayer /root/audiobraillinox/audio12.ogg");
        system("echo 1 >/sys/class/gpio/gpio18/value");
        system("echo 1 >/sys/class/gpio/gpio22/value");
        system("echo 1 >/sys/class/gpio/gpio24/value");
    }

    void MainWindow::on_pushButtonM_pressed()
    {
        system("mplayer /root/audiobraillinox/audio13.ogg");
        system("echo 1 >/sys/class/gpio/gpio18/value");
        system("echo 1 >/sys/class/gpio/gpio14/value");
        system("echo 1 >/sys/class/gpio/gpio24/value");
    }

    void MainWindow::on_pushButtonN_pressed()
    {
        system("mplayer /root/audiobraillinox/audio14.ogg");
        system("echo 1 >/sys/class/gpio/gpio18/value");
        system("echo 1 >/sys/class/gpio/gpio14/value");
        system("echo 1 >/sys/class/gpio/gpio24/value");
        system("echo 1 >/sys/class/gpio/gpio10/value");
    }

    void MainWindow::on_pushButtonO_pressed()
    {
        system("mplayer /root/audiobraillinox/audio16.ogg");
        system("echo 1 >/sys/class/gpio/gpio18/value");
        system("echo 1 >/sys/class/gpio/gpio10/value");
        system("echo 1 >/sys/class/gpio/gpio24/value");
    }

```



```

    }

    void MainWindow::on_pushButtonP_pressed()
    {
system("mplayer /root/audiobraillinux/audio17.ogg");
        system("echo 1 >/sys/class/gpio/gpio18/value");
        system("echo 1 >/sys/class/gpio/gpio22/value");
        system("echo 1 >/sys/class/gpio/gpio24/value");
        system("echo 1 >/sys/class/gpio/gpio14/value");
    }

    void MainWindow::on_pushButtonQ_pressed()
    {
system("mplayer /root/audiobraillinux/audio18.ogg");
        system("echo 1 >/sys/class/gpio/gpio18/value");
        system("echo 1 >/sys/class/gpio/gpio14/value");
        system("echo 1 >/sys/class/gpio/gpio24/value");
        system("echo 1 >/sys/class/gpio/gpio10/value");
        system("echo 1 >/sys/class/gpio/gpio22/value");
    }

    void MainWindow::on_pushButtonR_pressed()
    {
system("mplayer /root/audiobraillinux/audio19.ogg");
        system("echo 1 >/sys/class/gpio/gpio18/value");
        system("echo 1 >/sys/class/gpio/gpio22/value");
        system("echo 1 >/sys/class/gpio/gpio24/value");
        system("echo 1 >/sys/class/gpio/gpio10/value");
    }

    void MainWindow::on_pushButtonS_pressed()
    {
system("mplayer /root/audiobraillinux/audio20.ogg");
        system("echo 1 >/sys/class/gpio/gpio14/value");
        system("echo 1 >/sys/class/gpio/gpio22/value");

```

```

        system("echo 1 >/sys/class/gpio/gpio24/value");
    }

    void MainWindow::on_pushButtonT_pressed()
    {
system("mplayer /root/audiobraillinox/audio21.ogg");
        system("echo 1 >/sys/class/gpio/gpio14/value");
        system("echo 1 >/sys/class/gpio/gpio22/value");
        system("echo 1 >/sys/class/gpio/gpio24/value");
        system("echo 1 >/sys/class/gpio/gpio10/value");
    }

    void MainWindow::on_pushButtonU_pressed()
    {
system("mplayer /root/audiobraillinox/audio22.ogg");
        system("echo 1 >/sys/class/gpio/gpio18/value");
        system("echo 1 >/sys/class/gpio/gpio8/value");
        system("echo 1 >/sys/class/gpio/gpio24/value");
    }

    void MainWindow::on_pushButtonV_pressed()
    {
system("mplayer /root/audiobraillinox/audio24.ogg");
        system("echo 1 >/sys/class/gpio/gpio18/value");
        system("echo 1 >/sys/class/gpio/gpio22/value");
        system("echo 1 >/sys/class/gpio/gpio24/value");
        system("echo 1 >/sys/class/gpio/gpio8/value");
    }

    void MainWindow::on_pushButtonX_pressed()
    {
system("mplayer /root/audiobraillinox/audio25.ogg");
        system("echo 1 >/sys/class/gpio/gpio18/value");
        system("echo 1 >/sys/class/gpio/gpio8/value");
        system("echo 1 >/sys/class/gpio/gpio24/value");
    }

```

```

        system("echo 1 >/sys/class/gpio/gpio14/value");
    }

    void MainWindow::on_pushButtonY_pressed()
    {
system("mplayer /root/audiobraillinox/audio26.ogg");
        system("echo 1 >/sys/class/gpio/gpio18/value");
        system("echo 1 >/sys/class/gpio/gpio8/value");
        system("echo 1 >/sys/class/gpio/gpio14/value");
        system("echo 1 >/sys/class/gpio/gpio10/value");
        system("echo 1 >/sys/class/gpio/gpio24/value");
    }

    void MainWindow::on_pushButtonZ_pressed()
    {
system("mplayer /root/audiobraillinox/audio27.ogg");
        system("echo 1 >/sys/class/gpio/gpio18/value");
        system("echo 1 >/sys/class/gpio/gpio10/value");
        system("echo 1 >/sys/class/gpio/gpio24/value");
        system("echo 1 >/sys/class/gpio/gpio8/value");
    }

void MainWindow::on_pushButtonacentoa_pressed()
    {
        system("echo 1 >/sys/class/gpio/gpio18/value");
        system("echo 1 >/sys/class/gpio/gpio22/value");
        system("echo 1 >/sys/class/gpio/gpio24/value");
        system("echo 1 >/sys/class/gpio/gpio10/value");
        system("echo 1 >/sys/class/gpio/gpio8/value");
    }

void MainWindow::on_pushButtonacentoae_pressed()
    {
        system("echo 1 >/sys/class/gpio/gpio14/value");
        system("echo 1 >/sys/class/gpio/gpio22/value");
    }

```

```

        system("echo 1 >/sys/class/gpio/gpio24/value");
        system("echo 1 >/sys/class/gpio/gpio8/value");
    }

void MainWindow::on_pushButtonacentoi_pressed()
{
    system("echo 1 >/sys/class/gpio/gpio14/value");
    system("echo 1 >/sys/class/gpio/gpio24/value");
}

void MainWindow::on_pushButtonacentoo_pressed()
{
    system("echo 1 >/sys/class/gpio/gpio14/value");
    system("echo 1 >/sys/class/gpio/gpio8/value");
    system("echo 1 >/sys/class/gpio/gpio24/value");
}

void MainWindow::on_pushButtonacentou_pressed()
{
    system("echo 1 >/sys/class/gpio/gpio14/value");
    system("echo 1 >/sys/class/gpio/gpio22/value");
    system("echo 1 >/sys/class/gpio/gpio24/value");
    system("echo 1 >/sys/class/gpio/gpio10/value");
    system("echo 1 >/sys/class/gpio/gpio8/value");
}

void MainWindow::on_pushButtonenenhe_pressed()
{
    system("echo 1 >/sys/class/gpio/gpio18/value");
    system("echo 1 >/sys/class/gpio/gpio22/value");
    system("echo 1 >/sys/class/gpio/gpio14/value");
    system("echo 1 >/sys/class/gpio/gpio10/value");
    system("echo 1 >/sys/class/gpio/gpio8/value");
    system("mplayer /root/audiobraillinux/audio15.ogg");
}

```

```

void MainWindow::on_pushButtondieru_pressed()
{
    system("echo 1 >/sys/class/gpio/gpio22/value");
    system("echo 1 >/sys/class/gpio/gpio8/value");
    system("echo 1 >/sys/class/gpio/gpio10/value");
    system("echo 1 >/sys/class/gpio/gpio18/value");
}

void MainWindow::on_pushButtonw_pressed()
{
    system("echo 1 >/sys/class/gpio/gpio22/value");
    system("echo 1 >/sys/class/gpio/gpio8/value");
    system("echo 1 >/sys/class/gpio/gpio10/value");
    system("echo 1 >/sys/class/gpio/gpio14/value");
    system("mplayer /root/audiobraillinox/audio23.ogg");
}

void MainWindow::on_pushButtonpunto_pressed()
{
    system("echo 1 >/sys/class/gpio/gpio24/value");
    system("mplayer /root/audiobraillinox/audio58.ogg");
}

void MainWindow::on_pushButtoncoma_pressed()
{
    system("echo 1 >/sys/class/gpio/gpio22/value");
    system("mplayer /root/audiobraillinox/audio57.ogg");
}

void MainWindow::on_pushButtonpuntoycoma_pressed()
{
    system("echo 1 >/sys/class/gpio/gpio22/value");
    system("echo 1 >/sys/class/gpio/gpio24/value");
}

void MainWindow::on_pushButtondospuntos_pressed()

```

```

        {
            system("echo 1 >/sys/class/gpio/gpio22/value");
            system("echo 1 >/sys/class/gpio/gpio10/value");
        }
void MainWindow::on_pushButtonabrirycerrarinterrogacion_pressed()
    {
        system("echo 1 >/sys/class/gpio/gpio22/value");
        system("echo 1 >/sys/class/gpio/gpio8/value");
    }
void MainWindow::on_pushButtonabrircerraradmiracion_pressed()
    {
        system("echo 1 >/sys/class/gpio/gpio22/value");
        system("echo 1 >/sys/class/gpio/gpio24/value");
        system("echo 1 >/sys/class/gpio/gpio10/value");
    }
void MainWindow::on_pushButtonabrircerrarcomillas_pressed()
    {
        system("echo 1 >/sys/class/gpio/gpio22/value");
        system("echo 1 >/sys/class/gpio/gpio24/value");
        system("echo 1 >/sys/class/gpio/gpio8/value");
    }
void MainWindow::on_pushButtonabrirparent_pressed()
    {
        system("echo 1 >/sys/class/gpio/gpio22/value");
        system("echo 1 >/sys/class/gpio/gpio18/value");
        system("echo 1 >/sys/class/gpio/gpio8/value");
    }
void MainWindow::on_pushButtoncerrarparent_pressed()
    {
        system("echo 1 >/sys/class/gpio/gpio10/value");
        system("echo 1 >/sys/class/gpio/gpio24/value");
    }

```

```
        system("echo 1 >/sys/class/gpio/gpio14/value");  
    }  
void MainWindow::on_pushButtonguioncorto_pressed()  
{  
    system("echo 1 >/sys/class/gpio/gpio24/value");  
    system("echo 1 >/sys/class/gpio/gpio8/value");  
}
```

Fin. [...]