

ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA

SISTEMA DE LECTURA DE TEXTO E IDENTIFICACION DE IMÁGENES PARA PERSONAS NO VIDENTES

**PROYECTO PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN
ELECTRÓNICA Y TELECOMUNICACIONES**

MAGALY ISABEL RODRIGUEZ CARVAJAL

magaly.rc@hotmail.com

OSCAR MARCELINO LOPEZ REVELO

oscar.mlr@hotmail.com

DIRECTOR: Dr. ROBIN GERARDO ÁLVARES RUEDA

arobin7es@yahoo.es

Quito, Octubre 2008

DECLARACIÓN

Nosotros, Magaly Isabel Rodríguez Carvajal y Oscar Marcelino López Revelo, declaramos bajo juramento que el trabajo aquí descrito es de nuestra autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que hemos consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedemos nuestros derechos de propiedad intelectual correspondientes a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad vigente

Magaly Isabel Rodríguez Carvajal

Oscar Marcelino López Revelo

CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por los Señores Magaly Isabel Rodríguez Carvajal y Oscar Marcelino López Revelo, bajo mi supervisión.

Dr. Robin Álvarez
DIRECTOR DE PROYECTO

DEDICATORIA

Cada una de las letras de este trabajo es para ti mi Dios pues me has dirigido por el mejor camino, y me has dado la salud y sabiduría para alcanzar mis metas.

Magaly

AGRADECIMIENTO

A mi madre quien con su apoyo moral me ayudó a no abandonar la batalla contra la adversidad.

Magaly

DEDICATORIA

Amor, comprensión y apoyo son mis padres, a quienes dedico cada idea, pensamiento, ocurrencia y sensación del contenido de este fruto. Son quienes me enseñaron que con esfuerzo y perseverancia se logra lo que sea para un mañana de superación en el camino que me ha encomendado mi DIOS.

Mi ñaño y mis ñañas dulces y alegres me transmitieron su abrasador cariño como aliento en momentos perdidos supieron darme el empuje de cada verso en este fruto.

AGRADECIMIENTO

Ardiente de alegría y envolvente de amor es mi DIOS quien me escucha y me entiende. Inspirador de este fruto, miles de gracias por tenerme en este punto.

Sutil como una roza e inspiradora de universos, sabia como los poetas y luchadora como un guerrero, amiga de corazón ardiente y dulce de alegría, apoyo constante de firmes vigas y tierno corazón. Mi mamita que con sus fuertes manos me ha llevado a conquistar todo lo que en la adversidad se puede encontrar.

Fuerte y emprendedor es mi papito quien con sus sabias palabras me sabe guiar y apoyar, con su tierno corazón y sus manos de emprendedor me enseñó a afrontar lo nuevo del mundo y todo lo que pueda encontrar, gracias por tu apoyo de corazón.

Mis ñaños, mi espada y mi escudo quienes me abrazan hacia la lucha con su apoyo incondicional, mi alma y mi corazón les envuelve de abrazos con todo mi yo, por todo el apoyo de cada segundo y el aliento prestado para continuar.

CONTENIDO

RESUMEN	XIV
PRESENTACIÓN	XV
CAPÍTULO 1	1
1.1 INTRODUCCIÓN.....	1
1.2 PLANTEAMIENTO DEL PROBLEMA.....	1
Estadísticas registradas por el CONADIS	3
Instituciones que prestan atención a personas con discapacidad.....	4
1.3 OBJETIVOS DEL ESTUDIO.....	5
1.4 CONCEPTOS ASOCIADOS A LA DEFICIENCIA VISUAL Y SENTIDO TÁCTIL-KINESTÉSICO.....	6
1.4.1 DEFICIENCIA VISUAL	6
1.4.2 CLASIFICACIÓN DE LA DEFICIENCIA VISUAL.....	8
1.4.3 CAUSAS DE LA DEFICIENCIA VISUAL.....	9
1.4.4 HEREDITARIAS	9
1.4.4.1 Albinismo (carencia de pigmento).....	9
1.4.4.2 Aniridia (ausencia o atrofia del iris)	10
1.4.4.3 Atrofia del nervio óptico (degeneración nerviosa)	10
1.4.4.4 Cataratas congénitas (cristalino opaco).....	10
1.4.4.5 Coloboma (deformaciones del ojo)	11
1.4.4.6 Glaucoma congénito (lesiones por presión ocular).....	12
1.4.4.7 Miopía degenerativa (pérdida de agudeza visual).....	12
1.4.4.8 Queratocono (Córnea en forma de cono)	13
1.4.4.9 Retinitis Pigmentaria (pérdida pigmentaria retinal).....	14
1.4.4.10 DQUIRIDAS/ ACCIDENTALES	14

1.4.5	SENTIDO AUDITIVO.....	15
1.4.5.1	Aprendizaje y desarrollo auditivo	16
1.4.5.2	Sonidos del Medio.....	17
1.4.5.3	Sonidos Específicos.....	17
1.4.5.4	Discriminación de Sonidos	18
1.4.5.5	Reconocimiento Auditivo y Asociaciones	18
1.4.5.6	Interpretar Instrucciones Verbales	18
1.4.5.7	Habilidades Auditivas y Escuchar para Aprender.....	19
1.4.6	SENTIDO TÁCTIL-KINESTÉSICO.....	19
1.4.6.1	Aprendizaje táctil.....	19
1.4.6.2	Conocimiento y Atención	20
1.4.6.3	Estructura y Forma	20
1.4.6.4	Relación de las Partes al Todo.....	21
1.4.6.5	Representaciones Gráficas	21
1.4.6.6	Símbolos Braille.....	22
1.4.6.7	Sistema Braille.....	23
1.4.6.8	Lectura en Braille	24
1.5	TECNOLOGÍAS ESPECÍFICAS PARA PERSONAS CIEGAS Y DEFICIENTES VISUALES.....	25
1.5.1	DISPOSITIVOS DESTINADOS A POSIBILITAR EL ACCESO A LA INFORMÁTICA Y A INTERNET	25
a) _	Amplificadores de caracteres y gráficos en pantalla de ordenador....	25
b) _	Lectores de pantalla.....	26
c) _	Revisor de pantalla (JAWS).....	27
d) _	sintetizador para LINUX (ORCA).....	27
1.5.2	DISPOSITIVOS DESTINADOS A PERMITIR LA LECTURA DE TEXTOS EN SOPORTE TRADICIONAL	29

1.5.2.1	Programas de OCR	29
1.5.2.2	Sistemas de OCR compactos	30
1.5.2.3	Magnetófonos	30
1.5.3	DISPOSITIVOS QUE PERMITEN LA ESCRITURA Y LA IMPRESIÓN EN BRAILLE	31
1.5.3.1	Líneas braille	31
1.5.3.2	Anotadores electrónicos parlantes:	32
1.5.3.3	Impresoras braille.....	33
1.5.3.4	Un ordenador con tacto.....	34
1.5.3.5	Ratón de Computadora para Ciegos	34
1.5.3.6	Top Braille, un traductor para ciegos	36
1.6	SISTEMA A DESARROLLAR	37
1.6.1	HERRAMIENTAS A UTILIZAR	38
1.6.1.1	SOFTWARE	38
1.6.1.1.1	Descripción de PC VOZ	38
1.6.1.1.2	Descripción de VISUAL BASIC 6	38
1.6.1.1.3	Descripción de MATLAB.....	39
1.6.1.1.4	Descripción de OCR.....	39
1.6.1.1.5	Descripción de BASCOM AVR.....	40
1.6.1.1.5.1	CARACTERISTICAS	40
1.6.1.1.5.2	COMANDOS E INSTRUCCIONES:	41
1.6.1.2	HARDWARE	43
1.6.1.2.1	Descripción de ESCANER.....	43
1.6.1.2.2	DLINK WEBCAM DSB-C320	44
1.6.1.2.3	SISTEMAS MICROCONTROLADORES	45
1.6.1.2.4	AVR.....	46
1.6.1.2.4.1	Características de funcionamiento del AVR	47

CAPITULO 2	50
DISEÑO	50
2.1 DISEÑO DEL SOFTWARE DEL PROTOTIPO.....	50
2.1.1 DIAGRAMA DE BLOQUES DEL SOFTWARE.....	51
2.1.2 ENTORNO DEL SOFTWARE.....	51
2.1.3 PANTALLA DE INICIO.....	51
2.1.4 MENU DE APLICACIONES	57
2.1.5 LIBROS DIGITALES	58
2.1.6 DIGITALIZACIÓN POR MEDIO DE ESCÁNER.....	62
2.1.7 HISTORIAS EN MP3.....	66
2.1.7.1 Subrutinas utilizadas en el reproductor:	70
2.1.8 BOTÓN IMÁGENES.....	71
2.1.9 Botón LEDs.....	72
2.1.10 Botón Tacto	74
2.1.11 Botón WebCám.....	76
2.1.12 ROCESAMIENTO DE IMÁGENES	78
2.1.12.1 Procesamiento en MATLAB	82
2.2 DISEÑO DE HADWARE DEL PROTOTIPO.....	86
2.2.1 Diodo emisor de luz	86
2.2.2 BANCO DE LEDS.....	86
2.2.3 MATRIZ DE LEDS.....	87
2.2.4 FRECUENCIA DE REFRESCO O BARRIDO	87
2.2.5 TIEMPO DE PERSISTENCIA	89
2.2.6 FACTOR DE SERVICIO	90
2.2.7 EFECTOS DE UNA SEÑAL DE BARRIDO O DE PULSOS	90
2.2.8 NUMERO DE LEDs.....	91
2.2.9 FUNCIONAMIENTO DEL CONTROL.....	91
2.2.10 DISEÑO DE LA MATRIZ DE LEDS.....	93

2.2.10.1 FUENTE DE VOLTAJE.....	93
2.2.10.2 Regulación de tensión básica.....	93
2.2.10.3 COMUNICACIÓN SERIAL	93
2.2.10.4 CIRCUITO CONTROLADOR	94
2.2.10.5 Decodificadores	95
2.2.10.6 Multiplexado	97
2.2.10.6.1 Barrido de columnas.....	99
2.2.10.7 Circuito integrado ULN2803.....	100
2.2.10.8 Amplificación de Ánodos.....	101
2.2.11 MATRIZ DE DIODOS	102
2.2.11.1 Configuración de la Matriz.....	102
2.2.12 DISEÑO DEL PROTOTIPO DE MATRIZ TACTIL.....	104
2.2.12.1 DISEÑO DEL HARDWARE.....	104
2.2.12.1.1 MULTIPLEXACION POR LATCH.....	104
2.2.12.1.2 DESCRIPCION DE LOS ELECTROIMANES.....	106
CONSIDERACIONES TEORICAS.....	107
2.2.12.1.2.1 FUERZA MAGNETOMOTRIZ	107
2.2.12.1.2.2 FUERZA MAGNETIZADORA.....	108
2.2.12.1.2.3 PERMEABILIDAD	108
CONSIDERACIONES GENERALES.....	109
DISEÑO PRÁCTICO por el método de FUERZA-BRUTA.	110
2.2.12.2 DISEÑO DEL SOFTWARE.....	113
2.3 IMPLEMENTACION	116
2.3.1 CONSTRUCCIÓN DE PROTOTIPOS.....	116
2.3.2 Matriz de LEDs	116
2.3.2.1 Implementación de Placas.....	116

2.3.2.2	Placa de matriz de LEDs	116
2.3.3	Placa de Controlador de matriz de LEDs	122
2.3.4	MATRIZ TÁCTIL	127
2.3.4.1	Placa del controlador del prototipo	127
2.3.4.2	Matriz de electroimanes	133
2.3.4.2.1	Elaboración de Electroimanes.....	133
2.4	COSTOS DEL PROTOTIPO	144
CAPITULO 3		145
3.1	Pruebas, resultados y discusión.....	145
3.2	Resultados del sintetizador de voz.....	149
3.3	Resultados de la matriz de LEDs.....	150
3.4	Resultados de la matriz de tacto.....	153
CAPITULO 4		155
CONCLUSIONES Y RECOMENDACIONES		155
4.1	CONCLUSIONES.....	155
4.2	RECOMENDACIONES.....	156
BIBLIOGRAFIA		157

RESUMEN

El desarrollo del proyecto que se presenta trata del diseño y construcción de un sistema tecnológico de ayuda para personas no videntes, para el acceso y la integración al mundo cultural y laboral de las mismas. Para ello se han desarrollado componentes tanto de hardware como de software.

El hardware es un sistema sencillo de utilizar, el cual es controlable mediante un software amigable que permite al usuario interactuar con las diferentes aplicaciones como es: leer texto escaneado, libros digitales, reproducción de historias en formato MP3, el despliegue de gráficas prediseñadas y capturadas por Webcam.

El funcionamiento del sistema, esta basado fundamentalmente en el desarrollo de un software por medio del cual se controla cada una de las propiedades diseñadas y disponibles para el usuario para que por medio de este se pueda ejecutar las diferentes aplicaciones según sea el requerimiento.

Entre las opciones a escoger que se encuentran en la pantalla principal del programa son las siguientes: lectura de texto externo (revistas, libros, etc.), lectura de texto digitalizado por un escáner, lectura de libros digitales, escuchar historias, tacto de imágenes prediseñadas y tacto de imágenes capturadas por Webcam. Las cuales podrán ser seleccionadas y ejecutadas pulsando los botones izquierdo, derecho y central del ratón del computador.

Este sistema esta diseñado de tal manera que el usuario siempre estará informado en que parte de la pantalla se encuentra o que aplicación se encuentra seleccionada, este es uno de los sistemas más fáciles de acceder a la información, tanto para personas con deficiencia visual completa como con baja visión; su objetivo es sencillamente el envío de información de la computadora al usuario mediante mensajes hablados que, como veremos, suelen ser emitidos con voces total o parcialmente sintéticas aunque en algunas ocasiones se puedan emplear voces naturales grabadas. Para que de esta manera no haya problemas de orientación en el momento de utilizar alguna o todas las propiedades que se ofrece en la pantalla principal del programa.

PRESENTACIÓN

El concepto de supresión de barreras, que en pocas palabras puede definirse como la adaptación de las condiciones de trabajo, desplazamiento y vida cotidiana en general, para cubrir las necesidades de las personas con discapacidad visual, es de gran interés para empresas y ciudadanos, tanto afectados como ajenos al asunto por lo que se han preocupado por de alguna manera suplir la grave falta que supone la inexistencia de centros de rehabilitación visual basados en las nuevas tecnologías, investigando, experimentando y adaptando lo existente, para facilitar cada día más a las personas con discapacidad visual el acceso a todo tipo de información, pero en la mayoría de los casos esto termina con una llamada de atención sobre el gran obstáculo para la difusión de esta tecnología, que no es otro que su alto precio de adquisición ya que ha implicado las adaptaciones propiamente dichas y las aplicaciones de soporte para ellas.

Nosotros hemos trabajado, en el uso de la voz sintética, para facilitar a los usuarios ciegos acceso a información ya sea de texto externo, texto digitalizado u otros, pero también estamos centrados en el uso de la magnificación o ampliación de imágenes mediante la construcción de una matriz táctil para que ciegos y débiles visuales puedan acceder a información que no solo sea texto; es un paso gigantesco que abre un nuevo horizonte de posibilidades para el colectivo de discapacitados visuales tanto a nivel personal como laboral.

CAPÍTULO 1

1.1 INTRODUCCIÓN

Uno de los grandes problemas que tienen las personas consideradas no videntes ya sea porque poseen una agudeza visual inferior al diez por ciento de la óptima, discapacitadas visuales de nacimiento, pérdida de la visión de forma progresiva y también pérdida de la visión súbitamente a consecuencia de un accidente, es que sólo pueden acceder a la información escrita si previamente ha sido convertida al sistema Braille, código de lecto-escritura táctil inventado en el siglo XIX que, aún siendo el ideal para quienes no tienen vista, su aprendizaje y manipulación representan en muchos casos un proceso complejo. Por este motivo nuestro objetivo principal es el desarrollar un sistema que de alguna manera contribuya a la inserción e integración más completa de los discapacitados visuales en el mundo laboral y cultural, permitiendo el desarrollo y mejoramiento del conocimiento práctico y teórico de los ciegos y débiles visuales para que la falta de visión no represente más limitaciones de las estrictamente necesarias.

1.2 PLANTEAMIENTO DEL PROBLEMA

Las personas que padecen algún tipo de deficiencia visual, ya sea total o parcial, se encuentran en desventaja cuando pretenden acceder, y sobre todo, tratar, la información deseada y cuando de alguna manera se tratan de independizar, comunicar y capacitar, lo cual se torna un problema cuando de inserción laboral se trata. La ceguera ocasiona daños en todos los ámbitos de la vida, familiar, social, laboral o escolar, lo que obviamente repercute de forma negativa en las relaciones interpersonales.



Figura 1.1: Discapacidad visual

Para las personas no videntes la posibilidad de realizarse por medio de un trabajo es, en muchos casos, difícil. El estudio es visto, no como un medio para un futuro proyecto, sino como un fin en si mismo, una manera de llenar el presente. Y el trabajo, no como una manera de obtener ingreso y satisfacción personal, sino simplemente para mantenerse ocupado. Al llegar a esta etapa, los recursos educativos y de rehabilitación suelen agotarse. Resulta muy difícil construir un proyecto de vida, ya que las escuelas especiales, terapias, instituciones de rehabilitación contemplan la infancia, la adolescencia y no continúan apoyando la construcción de un proyecto de vida adulto es por esto que la persona queda discapacitada en una etapa posterior de su vida. La pérdida brusca o paulatina de una capacidad física es siempre una situación traumática y de crisis que, en muchos casos, es vivida como intolerable para el yo.

En cualquier área de la actividad humana que pensemos, nos daremos cuenta que un gran porcentaje de la información llega a través de la vista. Cuando la ceguera es total, el oído y el tacto pasan a ser los principales canales en la recepción de la información, por lo que el sistema braille fue y es una gran opción para poder acceder a información. No obstante el sistema braille mantiene varios inconvenientes, como por ejemplo, el excesivo volumen que ocupa cuando se imprime o el alto coste de su producción.

Estadísticas registradas por el CONADIS

El Consejo Nacional de Discapacidades, CONADIS, es un organismo autónomo de carácter público, creado en agosto de 1992, a través de la Ley 180 sobre Discapacidades.

Misión

Es un organismo público y autónomo que genera políticas e información, planifica y coordina acciones con los sectores público y privado, en el ámbito de las discapacidades y canaliza recursos nacionales e internacionales en este campo.

PROVINCIA	Discapacidad por genero		Mayores de edad	Menores de edad	Total
	M	F			
AZUAY	401	271	601	71	672
BOLIVAR	161	71	216	16	232
CAÑAR	134	59	179	14	193
CARCHI	95	70	146	19	165
CHIMBORAZO	246	136	341	41	382
COTOPAXI	236	144	341	39	380
EL ORO	476	245	643	78	721
ESMERALDAS	397	216	550	63	613
GALAPAGOS	10	6	13	3	16
GUAYAS	1773	872	2316	329	2645
IMBABURA	234	128	324	38	362
LOJA	407	269	576	100	676
LOS RIOS	314	159	417	56	473
MANABI	919	406	1168	157	1325
MORONA SANTIAGO	184	139	261	62	323
NAPO	110	80	160	30	190
ORELLANA	75	44	93	26	119
PASTAZA	67	37	86	18	104
PICHINCHA	1464	903	2130	237	2367
SUCUMBIOS	136	74	173	37	210
TUNGURAHUA	189	140	290	39	329
ZAMORA CHINCHIPE	81	46	109	18	127
TOTAL	8109	4515	11133	1491	12624

Tabla 1.1: Discapacidad visual en el Ecuador

Instituciones que prestan atención a personas con discapacidad

En el Ecuador existe un conjunto de Instituciones del sector público que prestan atención y servicios a las personas con discapacidad, como son:

- Las Comisiones Provinciales
- Los departamentos provinciales de los distintos Ministerios del Ecuador, entre otros el de Bienestar Social, el de Educación y Cultura, el de Salud y el de Trabajo.
- En el área de la Educación se tienen varias Escuelas.
- En el área de la Salud, se cuenta con distintos Hospitales que han destinado las unidades de rehabilitación para el servicio a este sector de la población.

Así mismo en el país existe un grupo de Organizaciones Privadas, que prestan servicio para las personas con discapacidad, como son:

- Las Fundaciones
- Las Corporaciones
- El Instituto Nacional del Niño y la Familia - INNFA -, con sus centros de rehabilitación en Ibarra, Quito, Portoviejo y Guayaquil, y sus Unidades del Programa de Atención Médico Solidario (PAMS)

Por otro lado, existen Organizaciones Privadas, formadas de Personas con Discapacidad y que han sido creadas por un grupo de personas con alguna discapacidad específica:

Estas son:

- La Federación Nacional de Ciegos del Ecuador - FENCE
- Organizaciones e Instituciones
- Comisión Técnica

Esto nos indica que cada vez más y más personas necesitan de un sistema eficaz y eficiente que les ayude a desarrollarse en todos los campos y quitar como principal barrera su discapacidad como tal.

1.3 OBJETIVOS DEL ESTUDIO

El presente trabajo tiene como objetivo el desarrollo de un hardware y herramientas de software que reciben las órdenes entregadas por el usuario mediante el ratón del computador, a la cual el software responde con sonidos que asemejan la voz humana, llamada voz sintetizada para que el usuario conozca el sitio donde se encuentra el sistema. Haciendo posible que los ciegos y débiles visuales lean sin necesidad de que esté escrito en Braille, desarrollando y mejorando el conocimiento práctico y teórico de los mismos en muchos ámbitos, y a la vez contribuyendo a la inserción e integración más completa de los discapacitados visuales en el mundo laboral y cultural.



Figura 1.2: Enseñanza con libros de Braille

Los programas OCR son muy importantes para las personas con discapacidad visual grave porque permiten extraer el texto de publicaciones impresas a partir de una imagen del documento digitalizada con ayuda de un escáner, pudiéndose verbalizar ese texto extraído mediante programas que lo leen con voz sintética, de esta manera se posibilita que las personas con discapacidad visual accedan a la mayoría de obras impresas y por fin sean cada uno de ellos los que elijan las lecturas que ellos deseen, y las puedan hacer de una forma práctica e inmediata, sin tener que esperar grabaciones o transcripciones.

1.4 CONCEPTOS ASOCIADOS A LA DEFICIENCIA VISUAL Y SENTIDO TÁCTIL-KINESTÉSICO

1.4.1 DEFICIENCIA VISUAL

La vista, desde el momento mismo del nacimiento, es un sentido social. Según estudios realizados, hasta los doce años la mayoría de las nociones aprendidas se captan a través de la vista, en una proporción del 83%, frente a los estímulos captados por los otros sentidos, que se reparten el 17% restante.

Los ojos, que comienzan captando tan sólo un juego de luces y sombras, activan zonas del cerebro que emiten respuestas motrices, y esta actividad sensorial-motriz es la clave del desarrollo del niño. Lo que el ojo ve, quiere tocarlo con la mano. A la primera etapa de concentración visual sigue otra de atención, y a estas dos una tercera de reconocimiento visual.

“Los términos de déficit visual, baja visión, visión residual, y otros, giran en torno a una reducción de la agudeza visual. Debido a un proceso de enfermedad que afecta a la zona ocular o cerebral.”

De este modo, el niño con déficit visual es entendido como aquel que padece de una alteración permanente en los ojos o en las vías de conducción del impulso visual. Esto conlleva a una disminución en la capacidad de visión, que, constituye un obstáculo para su desarrollo, por lo que requiere una atención a sus necesidades especiales.

Los defectos de visión son las anomalías sensoriales más frecuentes, existe un gran número de niños con defectos de visión suficientemente amplios, que necesitan una enseñanza especializada. En la intervención educativa, más importante que las limitaciones son las posibilidades que quedan para la intervención.

Para definir las minusvalías visuales, hay que tener en cuenta los siguientes aspectos:

- Agudeza visual: Es la capacidad que tiene el ojo para discriminar detalles de un objeto a cierta distancia.
- Campo Visual: Es la amplitud de campo que un sujeto puede llegar a ver. Cuanto más cerca del objeto menos campo visual.
- Debilidad de visión: Ambliopía, se conoce como ojo vago, lo provoca la ausencia de uso de ese ojo o por la miopía, es irreversible y se detecta de los 3-4 años.
- Baja visión: Cuando un sujeto necesita de determinadas ayudas ópticas para poder funcionar lo más adecuadamente como vidente.

Podemos considerar deficientes visuales a aquellos sujetos con problemas graves de visión y que para funcionar correctamente pueden necesitar de algunos de estos aspectos:

- Un programa de rehabilitación visual
- Un programa de estimulación visual
- Ayudas ópticas.

En el desarrollo y aprendizaje del sujeto con deficiencias visuales es importante tener en cuenta las siguientes variables:

- Grado de disminución de la vista: Sin resto de visión o con resto de visión útil.
- Momento de aparición de los problemas visuales: En el nacimiento, en el desarrollo o de adulto.
- Modo de aparición de la deficiencia: Gradual o súbita.
- Presencia o no de otros trastornos asociados: Neurológicos, endocrinos, metabólicos, sordera, etc.

En definitiva, la ceguera supone una discapacidad total para orientarse a sí mismo mediante la visión y la imposibilidad de leer aun con corrección óptica o magnificación de los textos escritos en tinta.

1.4.2 CLASIFICACIÓN DE LA DEFICIENCIA VISUAL

En sentido clínico, ya que la deficiencia visual puede presentar diferentes grados de variabilidad. Desde la ceguera hasta deficiencias visuales mínimas en general, se distinguen entre visión parcial, visión escasa, ceguera parcial y ceguera.

- *Visión parcial:* cuando la persona afectada muestra dificultades para percibir imágenes con uno o ambos ojos, siendo la iluminación y la distancia adecuadas, necesitando lentes u otros aparatos especiales para normalizar la visión.
- *Visión escasa:* Cuando el resto visual de la persona tan solo le permite ver objetos a escasos centímetros.
- *Ceguera parcial:* cuando el resto visual tan solo permite captar la luz, aunque sin formas, solo bultos y algunos matices de colores.
- *Ceguera:* la agudeza visual es útil cuando supera $1/3$ de la visión, de forma que el espacio comprendido entre $1/3$ y $1/10$ es lo que recibe la denominación de debilidad visual o ambliopía. Son ciegos quienes padecen ceguera o no perciben nada o apenas algo de luz; no obstante, sería conveniente distinguir entre ceguera de nacimiento y adquirida; pues el haber tenido oportunidad de percibir colores, tamaño, forma, etc. Facilita enormemente la posibilidad de autonomía y aprendizaje, lo que origina la conducta adaptativa. Cualquiera de las conductas que se describen a continuación, podrían describir una deficiencia visual:
 - Dificultad de localización de objetos a corta, media y larga distancia.
 - Dificultad de localización de objetos en movimiento.
 - Dificultad de desplazamiento.
 - Enrojecimiento de los ojos.
 - Mirada lateral.
 - Necesidad de acercamiento de los objetos que se manipulan.
 - Necesidad de mayor cantidad de luz.
 - Parpadeo y lagrimeo excesivo.

Las principales dificultades que suelen presentar los sujetos pacientes de baja visión, son: auto imagen alterada y deficiencias en el vínculo madre-hijo. Distorsión en la percepción de la realidad con integración pobre o confusa de la misma, imposibilidad de evitar comportamientos, gestos y juegos, problemas en el control del mundo que les rodea, ritmo más lento de maduración y desarrollo, trastornos en la atención e hiperactividad y necesidad de una estimulación lo más precoz posible.

1.4.3 CAUSAS DE LA DEFICIENCIA VISUAL

La deficiencia visual puede aparecer por distintos motivos, en función de la parte del proceso u órgano de la visión que se ve afectado, aunque, normalmente, las más frecuentes son las que afectan al globo ocular, destacando como más importantes las que a continuación se citan:

1.4.4 HEREDITARIAS

1.4.4.1 Albinismo (carencia de pigmento)

Ausencia congénita, tanto parcial como total, de la pigmentación normal, de forma que la piel es clara, el pelo blanco y los ojos rojos: se debe a un efecto en la síntesis de melanina.

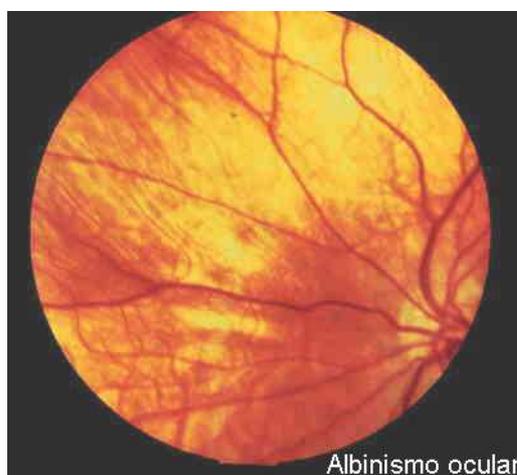


Figura 1.3: Albinismo Ocular

1.4.4.2 Aniridia (ausencia o atrofia del iris)

Es un trastorno ocular en la que están involucradas varias partes del ojo. Generalmente es bilateral (en ambos ojos) e incompleta, ya que en la mayoría de los casos, existe un iris incipiente que no ha llegado a desarrollarse.



Figura 1.4: Ausencia o atrofia del iris

La apariencia externa del ojo anirídico es una gran pupila central negra, en muchos casos, una pequeña franja coloreada.

1.4.4.3 Atrofia del nervio óptico (degeneración nerviosa)

Atrofia o muerte tisular del nervio que lleva la información desde la visión del ojo hasta el cerebro.

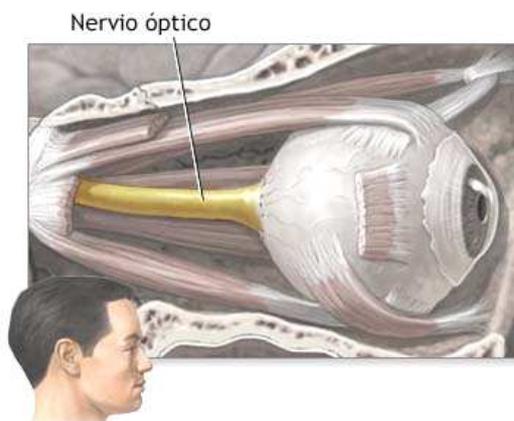


Figura 1.5: Degeneración nerviosa

1.4.4.4 Cataratas congénitas (cristalino opaco)

Es un área nublada u opaca en el cristalino del ojo. La catarata es congénita cuando dicha opacidad se presenta en el feto en algún momento durante el curso del embarazo y está presente en el momento del nacimiento.

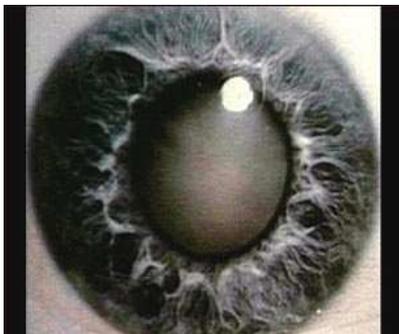
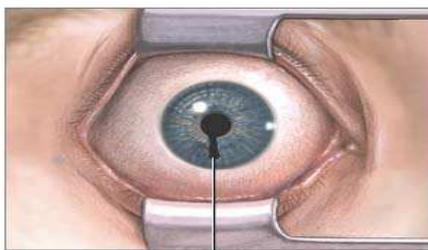


Figura 1.6: Cristalino opaco

1.4.4.5 Coloboma (deformaciones del ojo)

Es un defecto en el iris del ojo, el cual se ve como un hueco negro de profundidad variable en el borde de la pupila, dándole a esta forma irregular.



Coloboma of the iris

Figura 1.7: Deformaciones del ojo

Un tipo de coloboma es el ojo de gato, es cualquier defecto del iris que permita el paso de la luz al ojo por otra vía diferente a la pupila.

Es posible que un orificio o hendidura adicional esté presente desde el nacimiento, pero puede ser producto de algún trauma. Los colobomas también pueden aparecer en el párpado, siendo un defecto que interrumpe el borde de este.



Figura 1.8: Ojo de gato (coloboma)

1.4.4.6 Glaucoma congénito (lesiones por presión ocular)

Es un aumento de la presión intraocular, por falta del drenaje del humor acuoso, que produce lesiones en el nervio óptico con problemas en la visión y sino se corrige a tiempo produce ceguera. Si no se detecta a tiempo, este aumento de presión puede afectar al nervio óptico produciendo un deterioro progresivo del campo visual y una disminución de la visión. El sexo que más padece esta enfermedad son varones (65%), y se presenta en los primeros 3 años de vida.



Figura 1.9: Lesiones por presión ocular

1.4.4.7 Miopía degenerativa (pérdida de agudeza visual)

Las cosas se ven claras de cerca y de lejos borrosas, estos niños prefieren hacer trabajos de cerca y no exploran sus alrededores. Su error refractivo se corrige con una lente cóncava. Los niños con miopía degenerativa son miopes desde muy temprana edad y el problema se acrecienta con el paso de los años. Las lentes

pueden ayudar, pero no corrigen su visión hasta ser normal. Corren riesgo de sufrir desprendimiento de retina.

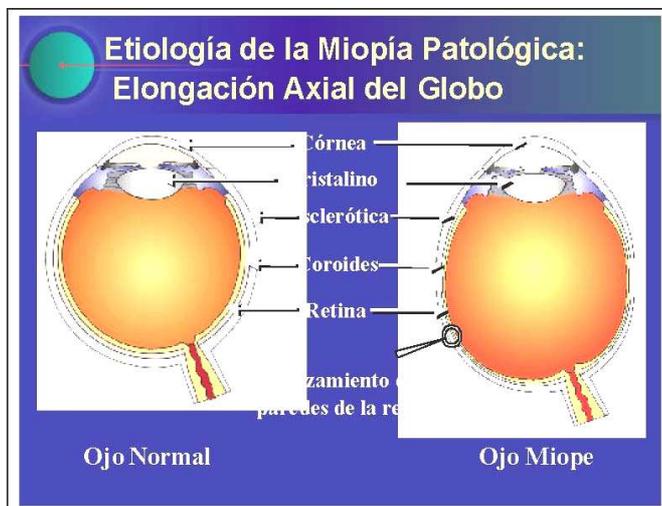


Figura 1.10: Pérdida de agudeza visual

1.4.4.8 Queratocono (Córnea en forma de cono)

El queratocono o córnea cónica es un desorden del ojo humano que raramente causa ceguera pero que puede interferir significativamente con la visión. Es una condición en la cual la forma normal redondeada de la cornea se distorsiona y desarrolla una prominencia en forma de cono. Ya que la córnea es la principal lente del ojo, la visión disminuye significativamente al deformarse de esta manera. La progresión del queratocono es generalmente lenta y puede detenerse en cualquiera de las fases: desde leve hasta severa. Si el queratocono progresa, la córnea se abomba y adelgaza tornándose irregular y algunas veces formando cicatrices.

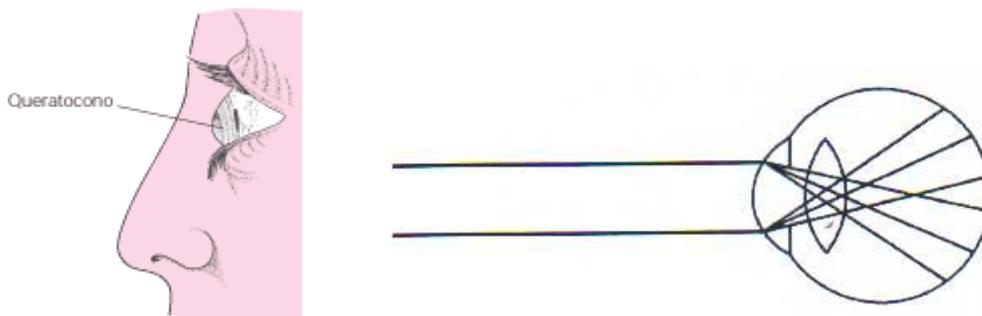


Figura 1.11: Córnea en forma de cono

1.4.4.9 Retinitis Pigmentaria (pérdida pigmentaria retinal)

La retinitis pigmentosa es el nombre dado a un grupo de desórdenes hereditarios del ojo que envuelven la retina del ojo, la capa nerviosa sensible a la luz que reviste la parte de atrás del ojo y que causan una reducción o pérdida en la habilidad visual gradual pero progresiva.



Figura 1.12 a) Visión normal



b) Con retinitis pigmentosa

1.4.4.10 DQUIRIDAS/ ACCIDENTALES

Desprendimiento de retina (lesión retinal)

El desprendimiento de retina es una separación anormal entre dos de las dos capas que constituyen la retina: el epitelio pigmentario (capa externa de la retina) y la retina sensorial (capa interna de la retina), entre las que se interpone un líquido. Si no se trata, el desprendimiento de retina lleva a la pérdida funcional total del ojo (ceguera).

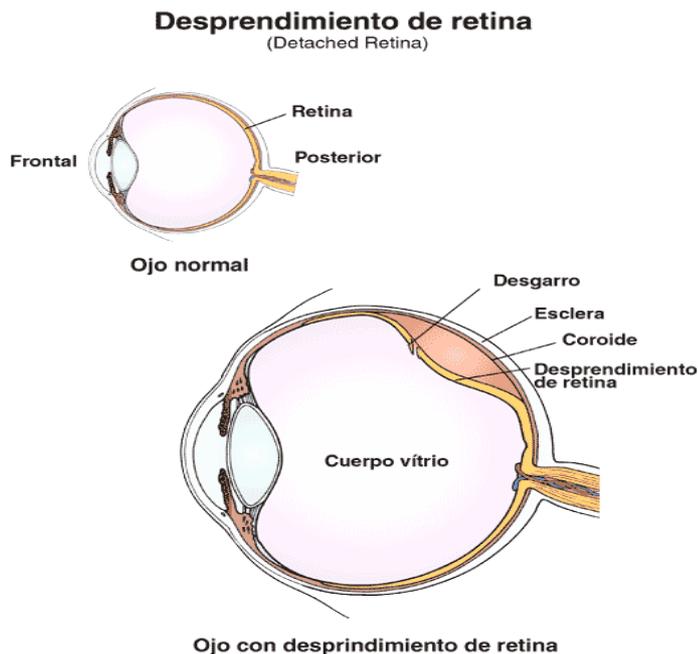


Figura 1.13: Desprendimiento de retina

1.4.5 SENTIDO AUDITIVO

El sentido auditivo funciona a través de terminaciones nerviosas que están profundamente ubicadas dentro del oído interno y rodeadas de líquidos. El estímulo a través del sentido del oído es más difícil durante los primeros meses de vida porque el área receptiva auditiva está ubicada profundamente dentro del centro del cerebro. Aunque el infante puede mostrar respuestas involuntarias al sonido, la verdadera discriminación y reconocimientos de los mismos no son posibles hasta después de varios meses de nacer. Pronto puede imitar sonidos especialmente la voz humana. Esta imitación es un proceso importante, ya que la información que llega al cerebro a través del sentido del oído forma la base para el desarrollo del futuro lenguaje y su emisión.

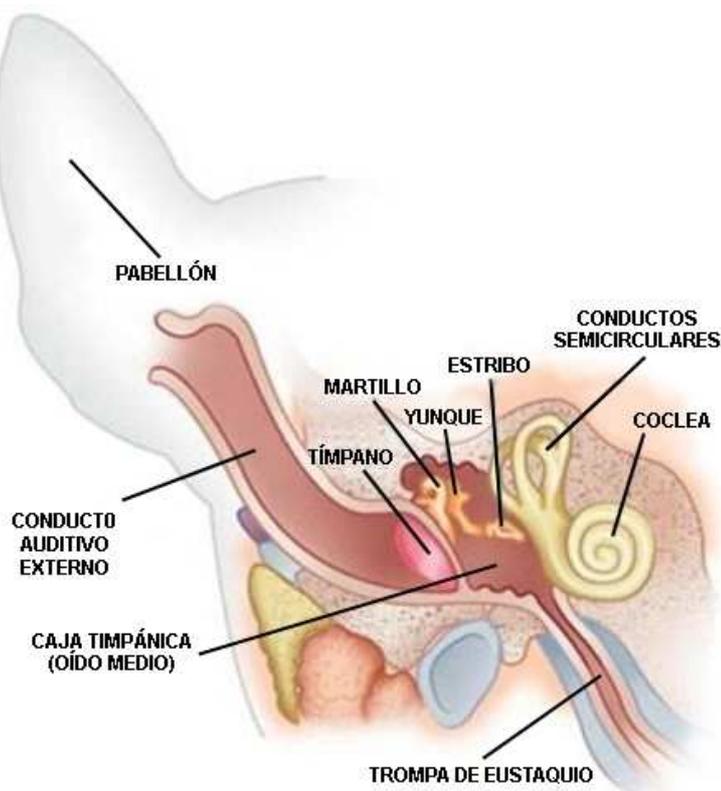


Figura 1.14: Sentido Auditivo

1.4.5.1 Aprendizaje y desarrollo auditivo

El ser humano debido a la naturaleza del sentido auditivo y a la continua presencia de sonidos que invaden el medio, tiene poco control físico sobre las sensaciones auditivas, pero sin embargo debe aprender a ejercitar un control mental con la percepción selectiva, pues la presencia de sonidos no significa necesariamente que el individuo los oye o los escucha.

Para el niño limitado visual una estimulación auditiva de sonidos sin significado puede provocar una actitud repetitiva. El aprendizaje debe dirigirse a la relación entre el lenguaje y el desarrollo auditivo, el uso del oído como un medio primario de aprendizaje, el desarrollo de eficientes habilidades para escuchar y la relación del desarrollo auditivo con el lenguaje como un instrumento para el pensamiento.

1.4.5.2 Sonidos del Medio.

Para que el niño ciego tenga conciencia de los sonidos debe tener la posibilidad de oír muchos sonidos agradables, tales como música y la voz humana. Estos sonidos crean en el infante un conocimiento inconsciente y lo unen al medio, además de traer consigo sentimientos de calidez y confort. Aun en el infante más pequeño, antes que pueda escuchar sonidos específicos, el sonido de la voz humana le provoca sentimientos de comunicación y lo une a las personas, probablemente sustituyendo las expresiones faciales y los gestos que no existen en el niño ciego total. Si en los primeros meses de vida no se expone al infante ciego a la voz humana, puede sentirse viviendo en un mundo totalmente aislado.

Una estimulación verbal continúa y apropiada en los primeros meses pueden ayudar al niño ciego a aprender a usar este contacto auditivo en forma semejante a lo que el niño con vista hace usando el movimiento de sus ojos para seguir las acciones de aquellas personas que le rodean. El colocar campanillas u otros objetos que producen ruidos agradables cerca del niño para que los pueda tocar, ayuda a estabilizar la idea que hay muchos sonidos dentro del medio. El niño al mismo tiempo que se mueve o desplaza por su ambiente debe escuchar sonidos específicos que le sean agradables.

1.4.5.3 Sonidos Específicos

La respuesta a sonidos específicos probablemente no ocurre antes de los cuatro o cinco meses de edad. La respuesta puede ser una sonrisa, escuchar intencionalmente y en silencio, y más tarde tratando de imitar y vocalizar. Esta conducta sugiere que el niño está comenzando a mantener contactos con sonidos específicos y a localizar la fuente de éstos. Dar la vuelta a la cabeza en respuesta a un sonido sugiere que intenta escuchar.

La manipulación de objetos únicamente para escuchar el sonido se puede enseñar, el niño debe aprender a buscar el "sonido - juguete", debe moverse hacia la fuente sonora.

1.4.5.4 Discriminación de Sonidos

Se trata de discriminar entre sonidos familiares, voces y tonos musicales. En este momento el bebé se puede mover en relación a los sonidos que hay en la casa para encontrar la fuente de los mismos. Los padres deben permitir esta actividad y dejar que el niño explore táctilmente los sonidos que han llamado su atención y pueda así localizarlos. Este también es el momento apropiado para dar el nombre de la fuente de los sonidos, para desarrollar así el conocimiento que los sonidos vienen de diferentes cosas que están en el hogar, lo que permite al niño asociar los sonidos con las cosas que toca. Así, el niño está aprendiendo a conectar sus propias acciones y la de los otros con específicos ruidos o sonidos.

El niño disminuido visual severo puede comenzar a reconocer la gente por los diferentes tonos de sus voces, o sus pasos, y puede asociar los pasos o las voces con anticipación.

1.4.5.5 Reconocimiento Auditivo y Asociaciones

Reconocimiento de sonidos relacionados a palabras específicas y conectadas con el lenguaje. Como el niño está aprendiendo que los objetos tienen nombre, también puede aprender que los sonidos tienen específicas palabras asociadas a ellos y que sus propias acciones tienen palabras para describirlas.

Hablarle a un niño disminuido visual es la única forma que tiene de interpretar el significado de sus propias acciones, al proveerles de estímulos auditivos significativos. La estimulación auditiva de la radio, televisión, etc., sin significado da como resultado un lenguaje ecolálico o verbalización que no tiene real significado para el niño y no contribuye a su desarrollo cognitivo

1.4.5.6 Interpretar Instrucciones Verbales

Es la etapa en la cual el niño disminuido visual puede aprender a escuchar en forma selectiva, ya puede comenzar a formular su propio lenguaje para

escucharse a él mismo y a los otros. A través de un proceso auditivo de retroalimentación el adulto reconocerá discrepancias en el significado e inexactitud de las cosas que dice el niño cuando ha escuchado las palabras "sin haber visto".

1.4.5.7 Habilidades Auditivas y Escuchar para Aprender

Lograr el más alto nivel en el proceso auditivo y la eficacia para escuchar es esencial para el futuro desarrollo cognitivo. El procesamiento a través del sentido auditivo sin la habilidad perceptiva es prácticamente una tarea imposible. Adquirir información táctil mediante la lectura Braille es un proceso lento.

A medida que el niño progresa a través de sus años escolares puede comprender que ha desarrollado sus habilidades para escuchar en la medida que sus entradas auditivas se presentan a una velocidad mucho más lenta que lo que puede su cerebro procesar, es decir, su cerebro va más rápido que la percepción auditiva.

1.4.6 SENTIDO TÁCTIL-KINESTÉSICO

Se le llama sentido de la "piel". Un compromiso activo con el medio y con los objetos de él, depende del sentido táctil-kinestésico, el cual está provocado por estímulos mecánicos, térmicos y químicos. Las manos y otras partes del cuerpo pueden accionar, tomar, empujar, frotar y levantar a fin de obtener información. El uso de los músculos kinestésicamente, a través del movimiento o el manipuleo de objetos o materiales, da la más comprensiva y precisa información cuando uno no puede usar el sentido de la visión.

1.4.6.1 Aprendizaje táctil

La progresión en la secuencia del aprendizaje es semejante en todos los sentidos. En los niños disminuidos visuales, sin embargo, puede ser necesaria mayor atención a niveles específicos y un determinado tipo de aprendizaje, ya que como se ha dicho anteriormente, mucho se descansa en el sentido de la visión. Ahora

veremos cada uno de los niveles en los cuales la secuencia y naturaleza del tipo de aprendizaje del desarrollo del niño ciego se debe utilizar.

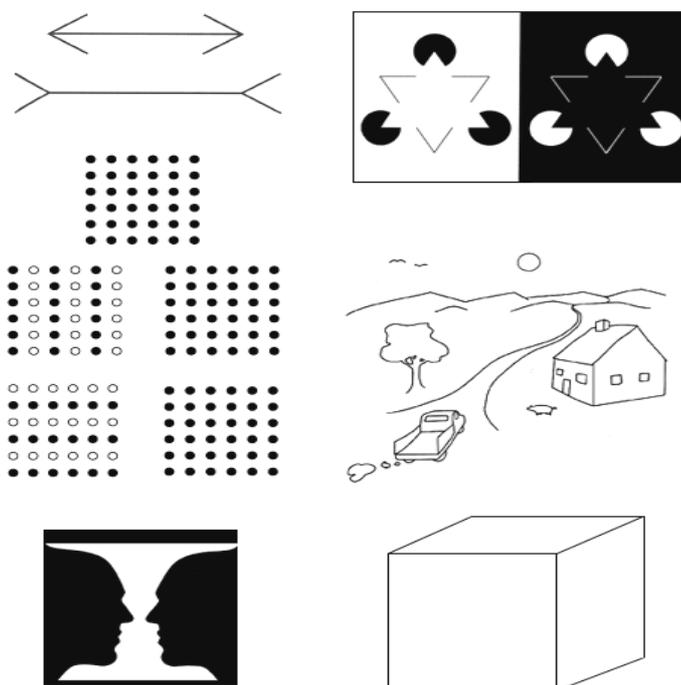


Figura 1.15: Aprendizaje Táctil

1.4.6.2 Conocimiento y Atención

El desarrollo táctil-kinestésico comienza con el conocimiento y atención, para diferenciar textura, temperatura y superficies vibratorias y materiales de variadas consistencias. Así los niños comienzan a conocer que algunos objetos son duros, otros blandos, unos ásperos y otros suaves, diferentes temperaturas, diferentes consistencias, distintas formas de actuar de las diferentes sustancias, unos objetos vibran y otros no, el niño ciego aprende que él recibe información de los objetos y al mismo tiempo es capaz de alterar y adaptar algunos objetos por el uso táctil-kinestésico mientras que no es posible modificar otros de la misma forma.

1.4.6.3 Estructura y Forma

Este nivel se relaciona con el "conocimiento de las estructuras básicas" de los objetos encontrados. Al mover sus manos a través de objetos, tomarlos y

sostenerlos de diferentes formas, objetos de diferentes tamaños, etc. los niños adquieren conocimientos acerca de los contornos y de la variedad de tamaño y peso. La interacción para la máxima información de esta etapa se logra a través de aspectos conocidos que son partes de la vida diaria del niño, tales como pastilla de jabón, tazas, platos, zapatos y medias. Cuando estos niños empiezan a discriminar entre los objetos es el momento de introducir el lenguaje que enseñe el reconocimiento de los objetos específicos. Al colocar las manos alrededor de un objeto, esto da una parte de información gruesa del mismo, pero moviendo sus manos y trazando la forma del objeto, le da una información específica y sucesiva acerca del mismo, que facilita su reconocimiento por el nombre.

1.4.6.4 Relación de las Partes al Todo

Cuando los niños ya son capaces de reconocer objetos de su vida diaria por el nombre ya están preparados para aprender las relaciones de las partes con el todo a través de objetos que pueden ser separados en partes y armados nuevamente. También un aprendizaje adicional debiera enfocarse en el uso de las manos y en la inspección manual, a través del manipuleo los niños invidentes comienzan a formar conceptos de las relaciones de las partes con el todo.

Las impresiones táctiles permiten al niño hacer acomodaciones a los nuevos elementos que tiene en sus manos y asimilar rápidamente éstas en relación a la información táctil que ya tiene.

1.4.6.5 Representaciones Gráficas

Se trata de presentar objetos en dos dimensiones.

Seleccionar esquemas estructurales simples tales como formas geométricas que pueden ser tocadas y representadas en distintas dimensiones, permite al niño gradualmente obtener sucesivas impresiones táctiles y a medida que sus dedos y músculos se mueven siguiendo distintos modelos, puede aprender a asociar el real objeto y el que está representado.

Dar una representación gráfica completa sería confuso y crearía lo que se llama "ruido táctil" (en el sentido de confusión).

1.4.6.6 Símbolos Braille

Es el nivel más alto de desarrollo táctil-kinestésico, se trata de la discriminación y reconocimiento de símbolos para que se puedan leer y escribir. El niño ciego total no sólo debe reconocer los símbolos táctilmente, sino también debe interpretar su significado en relación a otros signos Braille y al contexto del material que está leyendo. Esto provoca una gran carga en la memoria táctil-kinestésica y requiere por parte del niño tomar una inmediata decisión en relación al reconocimiento, memoria, asociación e interpretación.

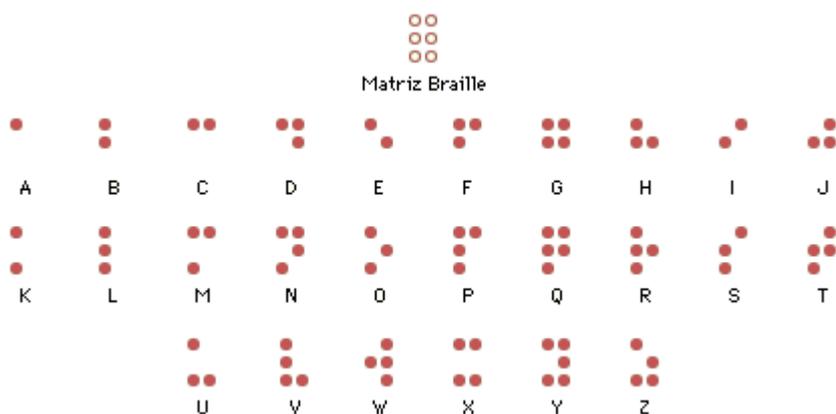


Figura 1.16: Símbolos Braille

El proceso de lectura táctil es más complejo que el de lectura visual, por los numerosos caracteres Braille, "63 combinaciones de puntos en un cajetín". Varios símbolos Braille tienen muchos usos y su interpretación depende de la relación con los otros símbolos, sus posiciones en el cajetín y la posición inicial, media y final en la palabra o en la oración. La misma letra o palabra puede tener diferentes significados según el lugar donde se encuentre dentro de una oración, de aquí que haya que tomar decisiones al leer Braille y esto requiere un alto nivel de habilidad en las funciones cognitivas.

1.4.6.7 Sistema Braille

El braille es un sistema de lectura y escritura táctil pensado para personas ciegas. Fue inventado en 1829 por el profesor francés Louis Braille, que se quedó ciego debido a un accidente durante su niñez mientras jugaba en el taller de su padre. Cuando tenía 13 años, el director de la escuela de ciegos y sordos de París, donde estudiaba el joven Braille le pidió que probara un sistema de lecto-escritura táctil inventado por un militar llamado Charles Barbier. Louis Braille, al cabo de un tiempo descubrió que el sistema era válido y lo reinventó utilizando un sistema de 8 puntos. Al cabo de unos años lo simplificó dejándole en el sistema universalmente conocido y adoptado de 6 puntos.

El braille resulta interesante por tratarse de un sistema de numeración binario que precedió a la invención de los ordenadores.



Figura 1.17: Sistema Braille en un tablón

El Braille se define como un sistema de lectoescritura táctil para ciegos, basado en la combinación de seis puntos en relieve, dispuestos en dos columnas verticales y paralelas de tres puntos cada una.

Este signo, formado por los seis puntos, se denomina signo generador o elemento universal del sistema Braille o generador Braille.

A partir de seis posiciones se pueden realizar 64 combinaciones diferentes. Braille organizó estas combinaciones en series o grupos de 10 caracteres cada uno, siguiendo unas normas muy simples y pensando en las necesidades del alfabeto francés por lo que en español existen algunas particularidades.

En Braille no existe signo para el acento ortográfico, así las vocales acentuadas tienen su propia representación. Para representar una letra o un símbolo se emplea un solo cajetín, que es un rectángulo vertical que tiene la posibilidad de albergar los seis puntos. Entre dos palabras se deja siempre un cajetín en blanco. No obstante existen símbolos en Braille sin transcripción alguna, sin representación directa en tinta, como por ejemplo el paréntesis auxiliar en Braille que se usa para evitar equívocos en funciones matemáticas, etc. Aun así existen signos en Braille que no teniendo transcripción en tinta por si mismos, modifican al signo al que preceden. Son los prefijos. Con estos se convierte cualquier letra en mayúscula, minúscula o cursiva

1.4.6.8 Lectura en Braille

La lectura en Braille no presenta excesiva dificultad respecto a la lectura en tinta. Los elementos básicos en el proceso de adquisición de la lectura son los mismos para ciegos y videntes. Por tratarse de un sistema lectoescritor que usa un código diferente al alfabético en tinta, requiere de un aprendizaje distinto.

La lectura mediante el tacto se realiza letra a letra y no a través del reconocimiento de las palabras completas, como sucede en tinta. Por ello se trata de una tarea lenta en un principio, que requiere de una gran concentración difícil de alcanzar a edades tempranas.

- La velocidad media de lectura de un ciego viene a ser de unas 100 palabras por minuto. En los niños el ciego tarda más en el adiestramiento lector que el vidente.

TECNOLOGÍAS ESPECÍFICAS PARA PERSONAS CIEGAS Y DEFICIENTES VISUALES

1.5.1 DISPOSITIVOS DESTINADOS A POSIBILITAR EL ACCESO A LA INFORMÁTICA Y A INTERNET

a)._ Amplificadores de caracteres y gráficos en pantalla de ordenador

Se trata de un tipo de herramienta que permite a las personas con resto visual funcional ver lo que aparece en la pantalla del ordenador gracias a la ampliación de las partes seleccionadas de la imagen. El producto se encarga de acceder a la información en pantalla y tratarla (modificando sus atributos de color, tamaño, forma, etc.) para devolvérsela al usuario en las condiciones de visualización elegidas por él, de forma que pueda leer cómoda y fácilmente la información de la pantalla. Este tratamiento de la imagen se realiza de forma rápida y transparente para el usuario, permitiendo además configurar el producto para realizar lecturas automáticas a la velocidad seleccionada, definir colores eligiéndolos de la paleta de Windows, y otras muchas funciones para optimizar y personalizar su rendimiento.

En la actualidad, todos los productos existentes para la magnificación de pantallas son potentes programas que trabajan utilizando las características intrínsecas del ordenador, es decir, que no se requiere la instalación de una tarjeta específica. Poseen para su manejo una interfaz fácil, y ofrecen grandes posibilidades de personalizar las opciones de visualización. Entre ellas se pueden citar la inversión de la pantalla a blanco sobre negro, las diferentes modalidades de coloración de la imagen o pantalla, el suavizado de bordes (que evita el efecto de dientes de sierra al aumentar mucho los caracteres, los cuales están realmente hechos de puntos cuadrados), la selección de distintos tipos de fuentes, de distintos tamaños, la aplicación de colores, etc.



Figura 1.18: Ampliador de caracteres

b)._ Lectores de pantalla

Se trata de herramientas informáticas que permiten el acceso al texto presente en la pantalla del ordenador por medio de su presentación en forma de voz usando una síntesis o de texto en sistema braille. Es, por tanto, la forma de comunicación entre el usuario ciego y el ordenador. El usuario escucha lo que se le presenta en la pantalla, o bien lo lee a través de alguno de los dispositivos de braille efímero (línea braille), pudiendo acceder a la información existente mediante órdenes del teclado, simplemente realizando funciones estándar de los sistemas operativos y obteniendo respuestas automáticas de los lectores de pantalla.



Figura 1.19: Lectores de Pantalla

c).- Revisor de pantalla (JAWS)

Un revisor de pantalla es una aplicación o un Software que permite a la persona invidente o baja visión saber que es lo que pasa en la pantalla del computador, entender cada uno de los procedimientos que se efectúan en él para lograr X o Y cosa, es un recurso auditivo que por medio de una voz sintetizada dice literalmente todo cuanto ocurra en el computador, es decir, la ubicación, por ejemplo si se está en el llamado escritorio, si se está en un menú, en un cuadro de diálogo, en una ventana, en una aplicación, etc. Además está en condición de brindar información adicional para que el usuario conozca como puede desplazarse en cada uno de los lugares antes mencionados.

También está en condiciones de leer un texto completo o de línea en línea, de frase en frase, de párrafo en párrafo, de palabra en palabra o de letra en letra, en office Word al igual que ocurre con las hojas de cálculo como las que se encuentran en office excel o en una presentación de powerpoint. En este momento seguramente se estarán preguntando como es posible esto si para todo en el computador se utiliza el ratón?, pues es preciso mencionar que los revisores de pantalla como el JAWS usan los comandos de Windows para enriquecerse y basan el manejo del computador en el principio de que toda acción del ratón está soportada en un comando de teclado permitiendo que todo lo que una persona vidente realiza con un simple clic, la persona invidente lo haga con una combinación de teclas por ejemplo: Para cerrar una aplicación o programa la persona vidente encausará el ratón hacia la X ubicada en la parte superior derecha de la pantalla mientras que la persona invidente usará la combinación Alt.+F4 para tal fin.

d)._ sintetizador para LINUX (ORCA)

Orca, es una tecnología de apoyo libre/open source, flexible, extensible y potente para las personas ciegas y deficientes visuales. Usando varias combinaciones de voz, braille y magnificación. Orca ayuda a proporcionar accesibilidad a las aplicaciones y toolkits que soporten AT-SPI (como por ejemplo el escritorio de

GNOME). El desarrollo de Orca ha sido liderado por el Accessibility Program Office de Sun Microsystems, Inc. con las contribuciones de muchos miembros de la comunidad.

El comando para ejecutar Orca es "orca". Puedes introducir este comando pulsando alt + f2 tras iniciar sesión, esperar un par de segundos y escribir orca más la tecla enter. Orca está diseñado para presentar la información tal como navegas por el escritorio utilizando los comandos internos de navegación del escritorio de GNOME. Estas teclas son consistentes y de uso común en la mayoría de aplicaciones de GNOME.

Además a veces querrás saber cómo controlar Orca, por ejemplo, como acceder al interfaz de configuración de Orca pulsando insertar + espacio mientras Orca se ejecuta, o cómo usar el modo de revisión plana para leer una ventana. Puedes consultar la lista de comandos de teclado de Orca o la lista de comandos de Orca en PC's portátiles. para saber con más detalle los comandos de Orca. El interfaz de configuración de Orca también incluye una solapa "Atajos de teclado" que te permite conocer las teclas disponibles en Orca.



Figura 1.20: Logotipo del sintetizador Orca

1.5.2 DISPOSITIVOS DESTINADOS A PERMITIR LA LECTURA DE TEXTOS EN SOPORTE TRADICIONAL

1.5.2.1 Programas de OCR

Son programas capaces de reconocer textos a partir de imágenes. El reconocimiento óptico de caracteres es un software que permite detectar en un papel la presencia de las formas gráficas correspondientes a letras, números y demás signos utilizados en la escritura habitual a partir de la imagen del texto que proporciona un escáner. El OCR actúa sobre la imagen codificada en forma digital. El texto reconocido queda almacenado en archivos estándar de ordenador.

Es importante tener en cuenta que para un ordenador, si no existe el programa de OCR, una imagen proveniente de un escáner es igual que una fotografía o un plano. Una foto de una página de un libro obtenida por un escáner, está codificada como fotografía, y como tal se trata en el ordenador. No hay letras, ni números, ni fotografías, sino solamente una imagen hasta que interviene el OCR (*Optical Character Recognition* o reconocimiento óptico de caracteres), que se encarga de separar las letras una a una o por grupos, dependiendo de la tecnología empleada, de tratar por otro lado los dibujos, gráficos y fotografías del papel, de comparar cada letra o grupo con modelos existentes en la memoria del ordenador, con diccionarios de cada lengua y con diferentes modelos de letras, y de obtener finalmente archivos de texto, o de texto e imagen, de tipo estándar, una vez almacenados en ficheros, los caracteres identificados pueden ser luego presentados al usuario ciego en forma ampliada en pantalla de ordenador, con síntesis de voz, por medio de una línea braille como texto braille efímero o combinando estas técnicas, dependiendo de las adaptaciones que este usuario tenga instaladas en su equipo informático.

Básicamente un programa de OCR consta de dos partes bien diferenciadas: el motor, verdadero cerebro del reconocimiento, que consta de programas y subprogramas capaces de realizar el estudio de la imagen recibida del escáner, y la interfaz, que es la parte visible del programa, encargada de la comunicación con el usuario.



Figura 1.21: Reconocimiento de Texto

1.5.2.2 Sistemas de OCR compactos

Son aparatos que permiten el reconocimiento de textos escritos en soporte de papel con salida de la información fundamentalmente a través de voz. En un mismo dispositivo compacto se integran un escáner, para capturar la imagen presentada, una placa de ordenador o cualquier otro tipo de circuitería capaz de alojar el software OCR, y la interfaz, el programa de OCR, que es la herramienta que permite reconocer entre gráficos la presencia y tipo de caracteres alfanuméricos, un teclado para el manejo de la interfaz y una salida en síntesis de voz de las palabras reconocidas.

Estos productos pueden ser una opción válida en la actualidad para personas que no tengan conocimientos informáticos y no deseen adquirirlos, como, por ejemplo, personas de edad avanzada. No obstante, siguen siendo sistemas caros y poco portátiles, tan solo justificables cuando no sea posible, por las causas que fuere, utilizar un ordenador en el que conectar el escáner y ejecutar un programa de OCR de los descritos anteriormente.

1.5.2.3 Magnetófonos

Se utilizan fundamentalmente para la escucha de los libros hablados, es decir, grabaciones sonoras de todo tipo de textos realizadas por un narrador con un sistema de grabación que permite cuadruplicar la duración de la información grabada. El selector de cada una de las cuatro pistas disponibles, así como el

control de la velocidad de reproducción o grabación, etc., los distinguen de los magnetófonos convencionales. La miniaturización y digitalización son el avance que cabe destacar en los dispositivos de grabación-reproducción.



Figura 1.22: Grabador de bobina abierta de Sony.

1.5.3 DISPOSITIVOS QUE PERMITEN LA ESCRITURA Y LA IMPRESIÓN EN BRAILLE

1.5.3.1 Líneas braille

Son dispositivos que alinean de forma mecánica una serie de elementos (que representan cada uno un carácter braille) por medio de diversos métodos electrónicos, generalmente en una sola línea de diversos anchos.

A cada elemento capaz de representar un carácter en braille se le denomina «celda braille». Un conjunto de celdas alineadas deben tener un dispositivo de control que traduzca la información recibida de una fuente, generalmente un ordenador, y que haga subir y bajar los puntos necesarios en cada celda para que, de una forma ordenada y coherente, se represente la información que se desea. Las líneas braille, por tanto, disponen de una parte de software de control y, si se conectan a ordenadores, el peso del ordenamiento del flujo de la información entre la pantalla y la línea lo lleva un software que se ejecuta en el ordenador, generalmente un lector de pantallas o un magnificador mixto.



Figura 1.23: Líneas Braille

1.5.3.2 Anotadores electrónicos parlantes:

Son máquinas portátiles para escribir en braille y para procesar información, útiles en todo tipo de situaciones en las que se requiera la escritura. Incorporan un teclado braille estándar de seis u ocho puntos (braille computerizado), tanto para introducir la información, como para configurar y enviar órdenes a los equipos. En general, poseen una síntesis de voz para poder escuchar la información, editar los textos, etc. Algunos poseen también línea braille. Disponen de puertos de comunicación para intercambiar información con otros equipos informáticos. Casi todos ellos, aparte de las funciones de escritura y edición de textos, son verdaderos PDAs para ciegos, ya que incorporan calculadoras, calendarios, alarmas, relojes, cronómetros, agendas de citas y otras funciones.



Figura 1.24: Anotador Parlante Braille.

1.5.3.3 Impresoras braille

Son periféricos que, conectados a un dispositivo informático que les envíe texto, imprimen en código braille sobre soporte de papel, plástico, u otras superficies. A pesar de que son impresoras preparadas para el sistema braille, por lo que realizan semiperforaciones con matrices de 6 u 8 puntos y separaciones entre líneas y caracteres, algunos modelos son capaces de producir gráficos en forma de imágenes en relieve, bien usando esta matriz con sus separaciones, bien logrando realizar líneas continuas de puntos con los que conformar los gráficos.



Figura 1.25: Máquina de escribir braille.

Podemos dividir las impresoras braille en los siguientes grupos:

- **Portátiles:** Diseñadas para uso personal, suelen imprimir por una sola cara y su producción es lenta.
- **Semi-industriales:** De tamaño medio, su producción es algo más grande que la de las impresoras personales. Su mecánica es más robusta, a fin de lograr mayores producciones y funcionamientos más prolongados en el tiempo. Suelen escribir a doble cara.
- **Industriales:** El tamaño es aún mayor que el de las anteriores. En ocasiones, son muebles tipo armarios roperos. Su mecánica es aún más robusta y precisan de grupos especiales de refrigeración y tensiones

elevadas de corriente, así como fuentes de alimentación estabilizadas de gran tamaño y peso. Escriben en doble cara y casi todas ellas logran realizar gráficos sin separaciones.

1.5.3.4 Un ordenador con tacto

La Agencia Nacional de Desarrollo Espacial japonesa (NASDA) ha presentado un ordenador que puede suponer una auténtica revolución para los usuarios invidentes. Es el primer prototipo de ordenador para ciegos que funciona con el tacto.



Figura 1.26: Ordenador para ciegos que funciona con el tacto

Se trata de un ordenador convencional, cuya pantalla está salpicada con 3.072 puntos que sirven para configurar las imágenes que aparecen en ésta. Permite a los ciegos ver lo que la pantalla del ordenador muestra a través de sus propias manos.

1.5.3.5 Ratón de Computadora para Ciegos

Un nuevo ratón de computadora para los no videntes pone los programas de computación y sus gráficos en la punta de sus dedos.



Figura 1.27: Ratón de Computadora para Ciegos

El Sistema de Contacto Virtual (SCV) es un ratón de computadora con tres almohadillas paralelas de contacto, cada una de las cuales está formada por 32 minúsculas "agujas" que pueden subir o bajar para crear copias táctiles de texto o imágenes en la pantalla. Las diferentes alturas de las agujas significan variaciones de color-blanco, gris claro, gris oscuro, negro.

El SCV hace rodar el cursor por la pantalla igual que el ratón de una persona vidente. A la vez, refleja en las almohadillas de contacto la imagen sobre la cual se está moviendo. Cuando lee textos, el SCV traduce tres letras de la pantalla a tres letras en relieve en las almohadillas. Las letras pueden ser en Braille o una "impresión" de los caracteres reales del alfabeto, en una fuente que pueda ser reconocida al tacto. Esta es una capacidad completamente nueva entre las ayudas para no videntes. El sistema de lectura es conocido como "Braille corriente".

Para las imágenes gráficas se activan instrucciones grabadas y un discurso generado por la computadora, lo que proporciona instrucciones de procedimiento y explicaciones que dan una información adicional acerca de la imagen que el usuario ha encontrado. La detención en cualquier parte de la imagen activa automáticamente una corta explicación. El beneficio de acceder a un entorno gráfico computarizado no sólo abre ante los ciegos una serie de ocupaciones de computación, sino también les permite crear obras de arte, conectando con programas estándar de dibujo.

Además, ocho botones arriba y a los costados del ratón permiten al usuario una interacción con la computadora, mover el cursor a posiciones específicas y enviar

órdenes. El centro de comando se encuentra en la palma de la mano del usuario, eliminando así la necesidad de memorizar incontables comandos basados en el teclado, lo que permite al ciego indicar y hacer clic, igual que los usuarios videntes de una PC.

1.5.3.6 Top Braille, un traductor para ciegos

Las personas ciegas van a tener menos dificultades a la hora de enfrentarse con textos que no estén en braille. Poder leer las etiquetas en los comercios, un folleto de publicidad o el periódico va a ser posible para los ciegos gracias a un lector portátil inventado por un ingeniero francés que es capaz de traducir el texto al braille de forma instantánea. Este aparato es casi del tamaño de un ratón de ordenador, pesa 120 gramos y cabe en la palma de una mano o en un bolsillo.

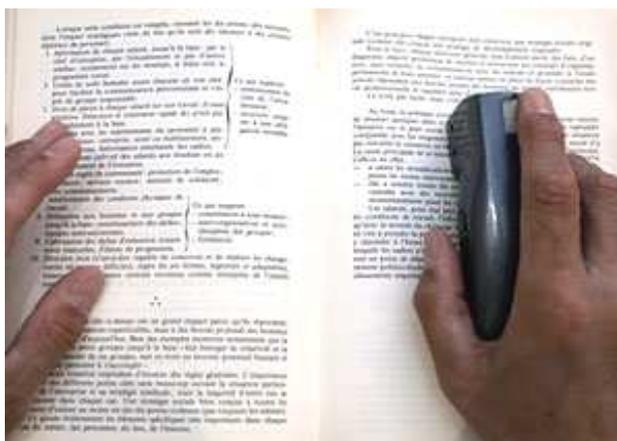


Figura 1.28: Traductor para ciegos

Raoul Parienti, el ingeniero que lo ha creado, estima que 'Top Braille', que es como se llama su invento, podría aumentar la autonomía de los 42 millones de ciegos en el mundo y de las personas con problemas de visión, a medida que el aparato se desplaza por una línea de texto, una microcámara digital escanea cada carácter, transmite las imágenes a un procesador que dirige una célula braille bajo el índice del usuario.

Las pequeñas partes puntiagudas de esta unidad bajan o suben para componer la traducción instantánea al braille de cada letra, tanto si está impresa en papel como en otro soporte, como latas de conservas o cajas de medicamentos.

1.6 SISTEMA A DESARROLLAR

La información que se presenta a continuación es una descripción de todas las herramientas que utilizamos para el diseño de nuestro prototipo. El sistema que se ha desarrollado consta de hardware y software el cual realiza algunas actividades y cada una de ellas tiene sus propias herramientas como son:

Digitalización de texto, para lo que se utiliza un sintetizador de voz, OCR, escáner, PC y sus respectivos parlantes.

Digitalización de imágenes, utilizamos la PC con su respectivo software que realizara el procesamiento de imágenes y dos módulos, el primero se encarga de mostrar las imágenes prediseñadas y las capturadas con una Webcam en un prototipo de matriz de LEDs, donde se puede observar que cada diodo encendido representa un pin de tacto que puede mostrar la diferente información; el segundo modulo es un prototipo de matriz de tacto que muestra diferentes ejemplos de imágenes que pueden ser de gran ayuda para personas con discapacidad visual.

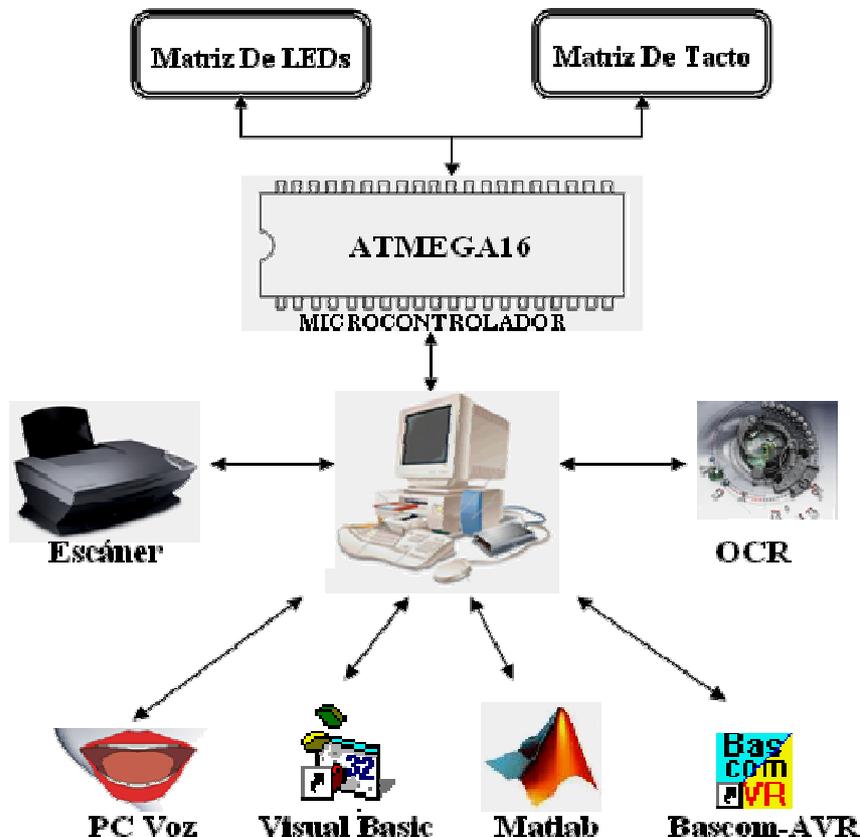


Figura 1.29: Componentes del Prototipo

1.6.1 HERRAMIENTAS A UTILIZAR

1.6.1.1 SOFTWARE

1.6.1.1.1 Descripción de PC VOZ

PCVoz es un programa lector de pantallas que lee todos los textos del programa con el que se trabaja, lee documentos, las opciones de menú, las ventanas de diálogo, etc. Requiere el uso de tarjeta de sonido y altavoces.

Esta es una herramienta amigable de fácil accesibilidad y para una gran diversidad de aplicaciones. Utiliza el poder de la voz para permitir a las personas con discapacidad visual disfrutar de las posibilidades de desarrollo personal, profesional.

1.6.1.1.2 Descripción de VISUAL BASIC 6

Visual Basic constituye un editor de código (programa donde se escribe el código fuente), un depurador (programa que corrige errores en el código fuente para que pueda ser bien compilado), un compilador (programa que traduce el código fuente a lenguaje de máquina), y un constructor de interfaz gráfica o GUI (es una forma de programar en la que no es necesario escribir el código para la parte gráfica del programa, sino que se puede hacer de forma visual).

Es un lenguaje de fácil aprendizaje tanto para programadores principiantes como expertos, es orientado a eventos, y se basa en el desarrollo de formularios que facilita el rápido desarrollo de aplicaciones gráficas. No requiere de manejo de punteros y posee un manejo muy sencillo de cadenas de caracteres. Posee varias bibliotecas para manejo de bases de datos, pudiendo conectar con cualquier base de datos a través de ODBC (Informix, DBase, Access, MySQL, SQL Server, PostgreSQL ,etc) a través de ADO.

Es utilizado principalmente para aplicaciones de gestión de empresas, debido a la rapidez con la que puede hacerse un programa que utilice una base de datos sencilla, además de la abundancia de programadores en este lenguaje.

Permite generar librerías dinámicas (DLL) ActiveX de forma nativa y Win32 (no ActiveX, sin interfaz COM) mediante una reconfiguración de su enlazador en el proceso de compilación.

1.6.1.1.3 Descripción de MATLAB

Matlab es una herramienta muy útil por la diversidad de programas que ofrece como son: la manipulación de matrices, la representación de datos y funciones, la implementación de algoritmos, la creación de interfaces de usuario (GUI) y la comunicación con programas en otros lenguajes y con otros dispositivos de hardware.

Este dispone también en la actualidad de un amplio número de programas de apoyo, denominado Toolboxes, que incrementa significativamente el número de funciones incorporadas en el programa principal. Estos Toolboxes cubren en la actualidad prácticamente casi todas las áreas principales en el mundo de la ingeniería y la simulación, destacando entre ellos el 'toolbox' de proceso de imágenes, señal, control robusto, estadística, análisis financiero, matemáticas simbólicas, redes neuronales, lógica difusa, identificación de sistemas, simulación de sistemas dinámicos, etc. es un entorno de cálculo técnico.

1.6.1.1.4 Descripción de OCR

El software de reconocimiento óptico de caracteres, abreviado habitualmente como OCR (*Optical character recognition*), extrae de una imagen los caracteres que componen un texto para almacenarlos en un formato con el cual puedan interactuar programas de edición de texto.

Mientras que en una imagen los caracteres se describen indicando cada uno de los píxeles que los forman, al convertirlos a un formato de texto (por ejemplo ASCII o Unicode), pasan a estar descritos por un solo número, por lo que se produce una reducción significativa del espacio en memoria que ocupan.

A partir de ahí el texto es reconocido como texto, de modo que se pueden buscar en él cadenas de caracteres, sacar el texto a un editor de textos, o a otras aplicaciones, etc.

1.6.1.1.5 Descripción de BASCOM AVR

El BASCOM-AVR es un compilador de BASIC para la familia AVR de ATMEL, desarrollado por la empresa Holandesa MCS Electronic. Ha sido desarrollado sobre W95/98/NT y dispone de todas las características de la familia BASCOM

1.6.1.1.5.1 CARACTERISTICAS

- BASIC estructurado con etiquetas.
- Programación estructurada con sentencias IF-THEN-ELSE-END IF, DO-LOOP, WHILE-WEND, SELECT- CASE.
- Generación de código máquina nativo en lugar de código interpretado.
- Bit, Byte, Entero, Word, Largo, y variables tipo String . (Solo con la Prof. Edición)
- Los programas compilados trabajan con todos los microprocesadores (no-MEGA) de AVR que tienen memoria interior. La Prof.-edición apoyará la serie de MEGA también. Puesto que los 1200 no tienen SRAM, no funcionará con los 1200.
- Las instrucciones y comandos de este BASIC son bastante similares a las del Visual Basic y QuickBASIC de Microsoft.
- Comandos específicos para el manejo de displays LCD, integrados I2C e integrados 1WIRE Chips, teclado de PC, teclado de matriz, recepción RC5, software UART. SPI, LCD Gráficos, envió de IR RC5 o código Sony.
- Soporta variables locales, uso de funciones, y librerías
- Emulador terminal integrado con opción de download.
- Simulador integrado por probar.
- Programador de ISP integrado (aplicación nota AVR910.ASM).

- Integrado el soporte del programador STK200 y STK300. También soporta el Electronics programme de bajo costo.
- Editor con subrayado de sentencias.
- Ayuda ON LINE en el editor.
- Soporta algunas declaraciones

1.6.1.1.5.2 COMANDOS E INSTRUCCIONES:

De estructura y condicionales

IF, THEN, ELSE, ELSEIF, END IF, DO, LOOP, WHILE, WEND, UNTIL, EXIT DO, EXIT WHILE, FOR, NEXT, TO, DOWNTO, STEP, EXIT FOR, ON .. GOTO/GOSUB, SELECT, CASE.

De entrada/salida

PRINT, INPUT, INKEY, PRINT, INPUTHEX, LCD, UPPERLINE, LOWERLINE, DISPLAY ON/OFF, CURSOR ON/OFF/BLINK/NOBLINK, HOME, LOCATE, SHIFTLCD LEFT/RIGHT, SHIFTCURSOR LEFT/RIGHT, CLS, DEFLCDCHAR, WAITKEY, INPUTBIN, PRINTBIN, OPEN, CLOSE, DEBOUNCE, SHIFTLIN, SHIFTOUT, GETATKBD, SPC

Funciones numéricas

AND, OR, XOR, INC, DEC, MOD, NOT, ABS, BCD, LOG, EXP, SQR, SIN, COS, TAN, ATN, ATN2, ASIN, ACOS, FIX, ROUND, MOD, SGN, POWER, RAD2DEG, DEG2RAD, LOG10, TANH, SINH, COSH.

12C

I2CSTART, I2CSTOP, I2CWBYTE, I2CRBYTE, I2CSEND and I2CRECEIVE.

1WIRE

1WWRITE, 1WREAD, 1WRESET, 1WIRECOUNT, 1WSEARCHFIRST, 1WSEARCHNEXT

SPI

SPIINIT, SPIIN, SPIOUT, SPIMOVE

Gestión de interrupciones

ON INT0/INT1/TIMER0/TIMER1/SERIAL, RETURN, ENABLE, DISABLE, COUNTERx, CAPTUREx, INTERRUPTS, CONFIG, START, LOAD.

Manipulación de bits

SET, RESET, ROTATE, SHIFT, BITWAIT, TOGGLE.

Variables

DIM, BIT , BYTE , INTEGER , WORD, LONG, SINGLE, STRING , DEFBIT, DEFBYTE, DEFINT, DEFWORD.

Varios

REM, ' , SWAP, END, STOP, CONST, DELAY, WAIT, WAITMS, GOTO, GOSUB, POWERDOWN, IDLE, DECLARE, CALL, SUB, END SUB, MAKEDEC, MAKEBCD, INP,OUT, ALIAS, DIM , ERASE, DATA, READ, RESTORE, INCR, DECR, PEEK, POKE, CPEEK, FUNCTION, READMAGCARD, BIN2GREY, GREY2BIN, CRC8, CRC16, CHECKSUM.

Directivas

\$INCLUDE, \$BAUD and \$CRYSTAL, \$SERIALINPUT, \$SERIALOUTPUT, \$RAMSIZE, \$RAMSTART, \$DEFAULT XRAM, \$ASM-\$END ASM, \$LCD, \$EXTERNAL, \$LIB.

Cadenas

STRING, SPACE, LEFT, RIGHT, MID, VAL, HEXVAL, LEN, STR, HEX, LTRIM, RTRIM, TRIM, LCASE, UCASE, FORMAT, FUSING, INSTR.

Y muchas otras funciones, declaraciones y directivas.

1.6.1.2 HARDWARE

1.6.1.2.1 Descripción de ESCANER



Figura 1.30: Escáner plano

Un escáner de computadora es un periférico que se utiliza para convertir, mediante el uso de la luz, imágenes impresas a formato digital.

Al obtenerse una imagen digital se puede corregir defectos, recortar un área específica de la imagen o también digitalizar texto mediante técnicas de OCR.

Datos de salida

Cabe mencionar que para el desarrollo del proyecto, el escáner se utiliza para capturar texto editable (no sólo imágenes), siempre y cuando la computadora pueda leer este texto. A este proceso se le llama OCR (Optical Character Recognition).

Al escanear se obtiene como resultado una imagen RGB no comprimida que puede transferirse a la computadora. Algunos escáneres comprimen y limpian la imagen usando algún tipo de firmware embebido. Normalmente las imágenes escaneadas se guardan con formato JPEG, TIFF, Mapa de bits y PNG dependiendo del uso que se le quiera dar a dicha imagen más tarde.

1.6.1.2.2 DLINK WEBCAM DSB-C320

La Webcam DSB-C320 (Figura 1.30) es una Cámara de video Digital USB, y está diseñada para ser utilizada conectada a la computadora. La conexión es plug & play, simplemente se debe conectar a algún puerto USB disponible, permite realizar sus propios vídeos digitales, gracias a su tecnología en hardware y software incorporada en este producto. Esta cámara tiene un precio de 28 dólares incluido el IVA pero no incluye costos de envío. A continuación presentaremos sus principales características:



Figura 1.31: Dlink Webcam DSB-C320

Características Principales:

- Resolución: 1024 x 768 píxeles
- Transferencia máxima de 30 frames por segundo.
- Dimensiones: 61mm(W)*65.3mm(H)*60.2mm(D)
- Usa tecnología CMOS para el sensor de luz.
- Conexión plug-play (puerto USB 2.0).
- Se usa para varias aplicaciones como: video conferencia, video seguridad, servicio remoto para consumidores.
- No necesita tarjetas de vídeo ni hardware adicional, la cámara se conecta a su computadora a través del puerto USB.

Requerimientos del Sistema:

- Esta cámara no tiene mayores requerimientos del sistema, ya que es muy simple y sencilla de manejar.
- 64Mb de memoria RAM
- Pentium II 450 MHz Interfaz USB.
- Microsoft Windows 98 Second Edition, Microsoft Windows 2000, Microsoft Windows Millennium Edition, Microsoft Windows XP

1.6.1.2.3 SISTEMAS MICROCONTROLADORES

Un microcontrolador es un ordenador completo. Contiene grandes cantidades de memoria, es capaz de comunicarse con dispositivos de entrada o dispositivos de salida, contiene todos los elementos del microprocesador básico además de otras funciones especializadas.

Un microcontrolador esta formado por lo general de los siguientes componentes:

- Procesador o UCP (Unidad Central de Proceso).
- Memoria RAM para contener los datos.
- Memoria para el programa tipo ROM/PROM/EPROM.
- Líneas de E/S para comunicarse con el exterior.
- Diversos módulos para el control de periféricos (temporizadores).
- Puertas Serie y Paralelo, (CAD: Conversores Analógico/Digital, CDA: Conversores Digital/Analógico, etc.).
- Generador de impulsos de reloj que sincronizan el funcionamiento de todo el sistema.

El utilizar un microcontrolador significa tener algunos beneficios entre los que más se destacan son los que hacen del proyecto en el que se lo esta empleando un trabajo con funcionalidades extras lo cual se refleja en algunas mejoras considerables como son:

Aumento de prestaciones lo que significa controlar a cada elemento con el que trabaja el microcontrolador.

Es mucho más confiable debido a que en un solo dispositivo se puede tener tantas funciones como si tuviéramos todos los elementos que lo representarían en caso de no usarlo y ese aumento de elementos podría significar un problema en cuanto a eficiencia se refiere, no solo porque hay más riesgo de cometer errores sino que también el aumentar elementos significaría aumentar el tamaño físico del proyecto y por ende aumento de mano de obra. El utilizar un microcontrolador significa también tener mayor flexibilidad porque las características de control están programadas por lo que su modificación sólo necesita cambios en el programa de instrucciones.

1.6.1.2.4 AVR

El AVR tiene 32 registros de 8 bits. Algunas instrucciones sólo operan en un subconjunto de estos registros. El conjunto de los 32 registros, los registros de entrada/salida y la memoria de datos conforman un espacio de direcciones unificado, al cual se accede a través de operaciones de carga/almacenamiento. A diferencia de los microcontroladores PIC, el stack se ubica en este espacio de memoria unificado, y no está limitado a un tamaño fijo.

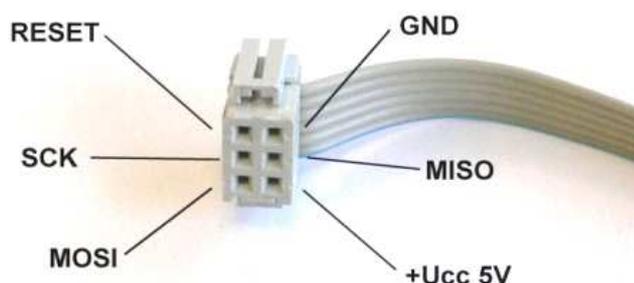


Figura 1.32: Descripción de pines del cable ISPof,

El AVR fue diseñado desde un comienzo para la ejecución eficiente de código C compilado. Como este lenguaje utiliza punteros para el manejo de variables en memoria, los tres últimos pares de registros internos del procesador, son usados como punteros de 16 bits al espacio de memoria externa, bajo los nombres X, Y y Z. Hacer que todo el banco superior de 16 registros de 8 bits tenga un

comportamiento alterno como un banco de 8 registros de 16 bits, complicaría mucho el diseño, alterando su característica principal de simplicidad.

1.6.1.2.4.1 Características de funcionamiento del AVR

- Los registros punteros X, Y y Z tienen capacidades de direccionamiento diferentes entre sí.
- Los registros 0 al 15 tienen diferentes capacidades de direccionamiento que los registros 16 al 31.
- Las registros de I/O 0 al 31 tienen distintas características que las posiciones 32 al 63.
- La instrucción CLR afecta los 'flag', mientras que la instrucción SER no lo hace, a pesar de que parecen ser instrucciones complementarias (dejar todos los bits en 1, y dejar todos los bits en 0 respectivamente).

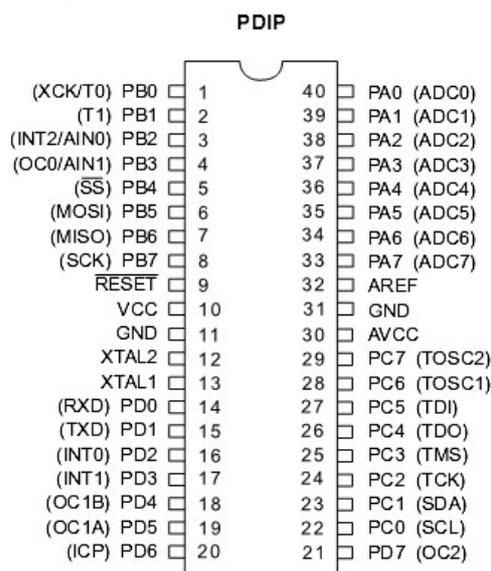


Figura 1.33: ATMEGA16

Dentro de los microcontroladores AVR, existen tres tipos de AVR: AVR, megaAVR y tinyAVR. A continuación se presentan las características del AVR ATmega16:

- Arquitectura:

Set de 130 instrucciones.

Ciclo de reloj de ejecución más simple.

32 x 8 registros de trabajo de propósito general.

Operación completamente estática.

Velocidad efectiva sobre los 16 MIPS a 16MHz.

- Memorias de programa y de datos no volátil.

Flash programable en sistema de 8K Bytes.

Soporta 10,000 ciclos de escritura/borrado.

EEPROM de 512 Bytes. Soporta 100,000 ciclos de escritura/borrado.

SRAM interna de 1K Byte.

Bits de bloqueo para seguridad del código.

- Dispositivos Periféricos

Dos temporizadores/contadores de 8-bit con pre-escala por separado.

Un temporizador/contador de 16-bit con pre-escala por separado.

Contador en Tiempo Real con oscilador separado.

Tres canales PWM (Modulación por Ancho de Pulso, en inglés)

Conversor ADC de 8 canales en encapsulado TQFP y MLF: *Seis canales de 10-bit de precisión. Dos canales de 8-bit de precisión.*

Conversos ADC de 6 canales en encapsulado PDIP: *Cuatro canales de 10-bit de precisión. Dos canales de 8-bit de precisión.*

Interfaz serial *Two-wire Serial Interfáce* orientado a Byte.

USART Serial programable. Interfaz Serial SPI Maestro/Esclavo

Comparador analógico.

- Características especiales:

Oscilador RC interno ajustable.

Fuentes de interrupción Externas e Internas

- I/O y Encapsulados

23 Líneas I/O Programables

PDIP: 28 pines, TQFP: 32 pines MLF: 32 pads.

- Voltajes de operación.

2.7-5.5V(ATmega8L)

4.5-5.5V(ATmega8)

- Grados de velocidad:

0 - 8 MHz (ATmegaSL)

0-16 MHz (ATmega8)

- Consumo de Potencia a 4 Mhz, 3V, 25°C

Activa: 3.6 mA

Modo Idle: 1.0 mA

Modo Power-down: 0.5 u A

CAPITULO 2

DISEÑO

2.1 DISEÑO DEL SOFTWARE DEL PROTOTIPO

En este capítulo se detalla el diseño e implementación del software del prototipo propuesto. El programa se encuentra diseñado básicamente como control principal, utilizando el lenguaje de programación de Visual Basic. Se utilizan los diferentes componentes de programas adicionales como el sintetizador de PC Voz, el programa de reconocimiento de texto ABBYY FineReader 5.0 Sprint, el software adicional del escáner, los componentes para reproducción de archivos mp3 y Matlab para el procesamiento de imágenes.

Se presenta las imágenes del programa que muestran la estructura y fácil funcionamiento del mismo desplegando una interfaz amigable para la persona que va a utilizarlo, además el programa esta completamente automatizado, en lo posible para simplificar a un mínimo de pasos el manejo para la persona no vidente.

De igual manera se explica el funcionamiento de los métodos y algoritmos en los que se basan los sistemas para lograr el reconocimiento de texto digitalizado, para luego pasarlo al sintetizador de voz el mismo que permite la reproducción de libros simulando voz humana por medio de la cual se da paso a la lectura de libros digitales y reproducción de historias en mp3. También se expone la explicación de los métodos utilizados en el procesamiento de imágenes para después ser mostrados en los módulos prototipos creados para presentar las imágenes prediseñadas o las capturadas por Webcam.

2.1.1 DIAGRAMA DE BLOQUES DEL SOFTWARE

El siguiente diagrama muestra como se encuentra estructurado el software del proyecto en su totalidad, cada herramienta del menú tiene un vínculo que llevara a otra ventana donde podemos ejecutar una aplicación determina.

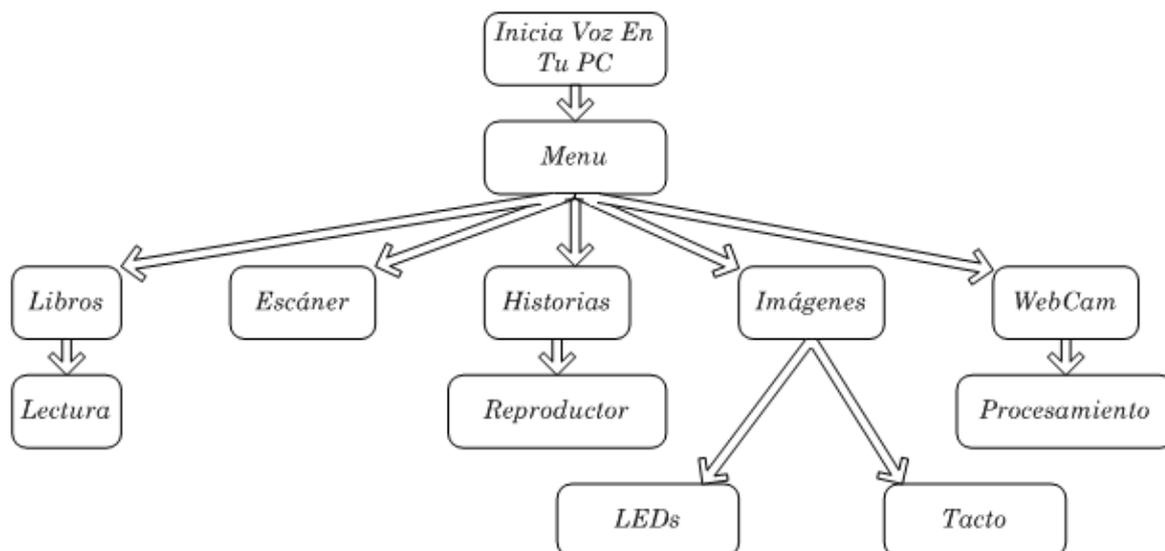


Figura 2.1: Diagrama de bloques del software

2.1.2 ENTORNO DEL SOFTWARE

Cuando la maquina se haya encendido por completo y el entorno del sistema operativo de Windows se encuentre totalmente ejecutado, la primera pantalla que se obtiene es la figura 2.2, que inicia el camino a una exploración de herramientas de ayuda para personas con discapacidad visual, abriendo el paso a mucha información y aplicaciones que un ordenador les puede ofrecer.

2.1.3 PANTALLA DE INICIO

Las pantallas son diseñadas en forma que la persona no vidente pueda manejarlo lo mas sencillamente posible, es por esta razón que su diseño tiene en su mayor parte una automatización de todos los controles.



Figura 2.2: Menú de inicio de Voz en tu PC

Cuando se inicia el programa se ejecutan los componentes que lo conforman, como DirectSS1 el mismo que esta incluido en la lista de los controles de Visual Basic al instalar el sintetizador de voz. Para escuchar el texto de cada botón, se utiliza el ratón del computador.

Funciones del ratón

Cada vez que se hace un clic con el ratón con cualquiera de los botones del mismo, sin importar si este se haya deslizado o no, se ejecutan varios comandos que permite escuchar la voz sintética que ofrece el PCVoz para guiar y ayudar al usuario a orientarse.

Comandos utilizados:

Cuando se inicia el programa, también se inicia un contador que será el que permitirá recorrer por toda las aplicaciones, es decir cuando se hace un clic con el

botón izquierdo del ratón el contador crecerá de uno en uno hasta un límite que será el número de aplicaciones presentes que se tiene en la pantalla.

Cuando se hace clic con el botón derecho del ratón, el contador decrecerá del valor en el que se encuentre actualmente, lo cual permitirá ir a la aplicación anteriormente escuchada y si se desea ejecutar lo que se escuchó con la voz sintetizada al realizar clic, solo hay que presionar el botón central del ratón. Es decir los botones izquierdo y derecho del ratón tienen la función de permitir la navegación sobre todas las aplicaciones disponibles en este proyecto y dependiendo de lo que el usuario desee realizar después de haber escuchado todas las opciones, él podrá ejecutar la aplicación requerida con el botón central del ratón.

La estructura del programa tiene como base principal subrutinas que dependiendo del valor del contador se ejecutarán las líneas de comando que a este le correspondan.

Iniciamos con la variable del contador donde se almacenará el aumento o decremento según el caso, de los números para ejecutar la aplicación correspondiente.

```
Dim contador As String
```

Subrutina de ayuda que ejecuta el comando de sintetizador de voz

```
Private Sub Cmd_Ayuda()  
    DirectSS1.Speak "Usa las teclas del ratón para navegar y el centro para ejecutar"  
End Sub
```

Subrutina para cerrar el programa

```
Private Sub Cmd_Cerrar()  
    End  
End Sub
```

Subrutina para ingresar al menú de aplicaciones, cuando se ingrese a este menú se presenta una nueva pantalla que muestra las diferentes aplicaciones que este tiene y se sierra la anterior.

```
Private Sub Cmd_Menu()
    Menu.Show
    Inicio.Hide
End Sub
```

```
Private Sub Form_Load()
    contador = 0
End Sub
```

El siguiente evento es el que da vida a este programa ya que recepta las pulsaciones del ratón y dependiendo de cuales fuesen estas se ejecuta una aplicación.

```
Private Sub Form_MouseDown(Button As Integer, Shift As Integer, X As Single, Y As Single)
```

Identifica si se pulso el botón izquierdo del ratón y se inicia el contador en forma ascendente enviando el resultado a una caja de texto.

```
    If Button = 1 Then
        contador = contador + 1
        Text1.Text = contador
    End If
```

Identifica si se pulso el botón derecho del ratón y el contador decrece leyendo el dato que se encuentre en la caja de texto.

```
    If Button = 2 Then
        contador = contador - 1
        Text1.Text = contador
    End If
```

Identifica si se pulso el botón central del ratón y ejecuta la opción que va ha depender de lo que se encuentre en la caja de texto. Si es el caso que en la caja de texto se encuentre el número 1 entonces se ejecutara la subrutina ayuda.

```

If Button = 4 Then
  If Text1.Text = 1 Then
    Cmd_Ayuda
  End If
  If Text1.Text = 2 Then
    Cmd_Menu
  End If
  If Text1.Text = 3 Then
    Cmd_Cerrar
  End If
End If

```

Controlamos los valores del contador, cuando este debe incrementar o decrecer el valor actual de la caja de texto.

```

If contador < 1 Then
  Text1.Text = 3
  contador = 3
End If
If contador > 3 Then
  Text1.Text = 1
  contador = 1
End If
End Sub

```

El siguiente evento de la caja de texto realiza un escaneo del valor que se encuentra en la caja de texto para luego ser ejecutado, este método lo utilizamos para que el usuario pueda navegar por el menú escuchando que opciones tiene el programa y cual de ellas desea utilizar.

```

Private Sub Text1_Change()

  COMAN = UCase(Text1.Text)
  **** PARA EL BOTON AYUDA****
  If InStr(COMAN, "1") = 0 Then

```

```

Else
AYUDA = "1"
End If
If AYUDA = "1" Then
AYUDA = " "
DirectSS1.Speak "Sobre boton ayuda"
End If

**** PARA EL BOTON MENU ****
If InStr(COMAN, "2") = 0 Then
Else
AMenu = "1"
End If
If AMenu = "1" Then
AMenu = " "
DirectSS1.Speak "Sobre boton menú"
End If

**** PARA EL BOTON CERRAR****
If InStr(COMAN, "3") = 0 Then
Else
CERRAR = "1"
End If
If CERRAR = "1" Then
CERRAR = " "
DirectSS1.Speak "Sobre boton cerrar"
End If
End Sub

```

Este método es utilizado para todas las aplicaciones del programa tanto su forma de navegación como de ejecución.

Las imágenes que corresponden a cada ventana nos muestran las aplicaciones del programa las mismas que están representadas mediante botones, estas imágenes tienen el objetivo de mostrar de que manera y en que orden están desarrolladas las funciones del programa pero esto solo tiene sentido para las personas videntes que deseen explorar lo que aquí se propone y porque no desarrollar herramientas que mejoren nuestra propuesta. Para una mejor comprensión detallaremos el contenido de cada uno de ellos que esta expresado en líneas de comandos como una subrutina.

Botón ayuda, es quien da las instrucciones de cómo utilizar el programa de una forma eficiente.

Botón menú, da el acceso al menú de aplicaciones los cuales son procesamientos que se han propuesto en el proyecto, asiendo clic con el botón central del ratón despliega las diferentes opciones que ofrece el programa al usuario del proyecto.

2.1.4 MENU DE APLICACIONES

En esta ventana encontramos las diferentes aplicaciones que tiene el proyecto con las mismas funciones de ayuda descritas para la figura anterior, cada uno de estos botones tiene la particularidad de ingresar a un siguiente menú que realiza el procesamiento adecuado de acuerdo a la aplicación correspondiente.



Figura 2.3: Menú de aplicaciones

Botón libros, pasa a un siguiente menú el cual permite escoger un libro para ser escuchado con el sintetizador de voz y un menú de reproducción de avance del texto que se encuentra ejecutando en ese instante.

2.1.5 LIBROS DIGITALES

En esta ventana tenemos los títulos de diferentes libros digitales, en nuestro caso se encuentran en formato RTF que puede ser editado por Microsoft Word u otro programa que utilice este tipo de documentos. Se utilizo este formato porque el formato RTF se puede leer en numerosos programas y plataformas. Aunque muchas aplicaciones pueden importar archivos DOC de Microsoft Word, algunos programas más antiguos son incapaces de importar o convertir versiones más recientes de Word.

Esto no sucede con el formato RTF. Todos los programas de procesamiento de textos de mayor difusión pueden leer archivos RTF del sistema, los archivos RTF son fáciles de usar. Para guardar un documento en formato RTF, seleccione la opción Guardar Como del menú Archivo y luego RTF.

Si se desea escuchar algún libro, haciendo clic, se ejecutan algunas instrucciones que cargan el texto adecuado al componente *RichTextBox* quien nos da la facilidad de cambiar el color de determinada porción de un texto, el tipo de fuente, mostrar imágenes, cargar archivos de texto enriquecido (archivos RTF), leer y guardar archivos mediante una serie de métodos que posee el control, tanto RTF como TXT, buscar cadenas de texto dentro del control con un método propio, y muchas otras opciones. También permite automatizar el avance del texto a leer, escuchándolo por medio del sintetizador de voz.

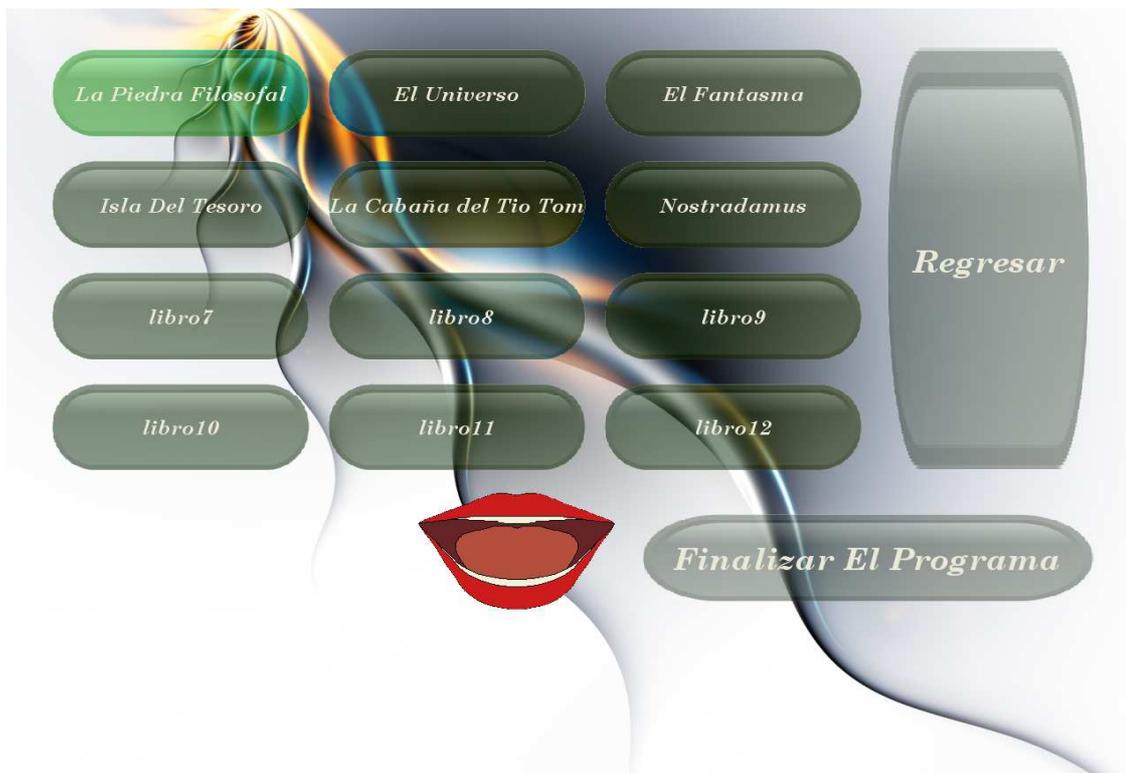


Figura 2.4: Menú de libros digitales

Comandos utilizados

Cuando el usuario selecciona el libro escogido por alguno de los métodos descritos anteriormente se ejecutara la siguiente codificación.

```
Private Sub Cmd_Lib1()
    Lectura.Show
    Libros.Hide
    Lectura.Text2.Text = "libro1"
End Sub
```

El comando *Lectura.Show* abre una nueva ventana, *Libros.Hide* cierra la actual ventana de los libros y *Lectura.Text2.Text = "libro1"* hace un pequeño escrito en la caja de texto dos, de la ventana lectura (figura 2.4) para cargar automáticamente el libro adecuado al editor *RichTextBox*. En la forma *Lectura* tenemos la codificación de carga del texto.

```
Private Sub Text2_Change()
```

```

If Text2.Text = "libro1" Then
    RichTextBox1.LoadFile ("D:\Oscar\Proyecto\visualBasic\Ejemplos\VozPC\LIBROS\HARRY POTTER Y
LA PIEDRA FILOSOFAL.RTF")
End If

If Text2.Text = "libro2" Then
    RichTextBox1.LoadFile ("D:\Proyecto\visualBasic\Ejemplos\VozPC\LIBROS\EI universo en una cáscara
de nuez.rtf")
End If
.. etc, etc
End Sub

```

Con el comando comparación *If* determinamos que libro fue seleccionado para su carga respectiva desde el archivo donde se encuentra guardado y pasarlo al editor de *RichTextBox* que nos facilita la reproducción, para la carga del archivo utilizamos el comando de lectura de archivos *RichTextBox1.LoadFile ("archivo.rtf")* donde debe estar el nombre adecuado y dirección del archivo a ser usado.

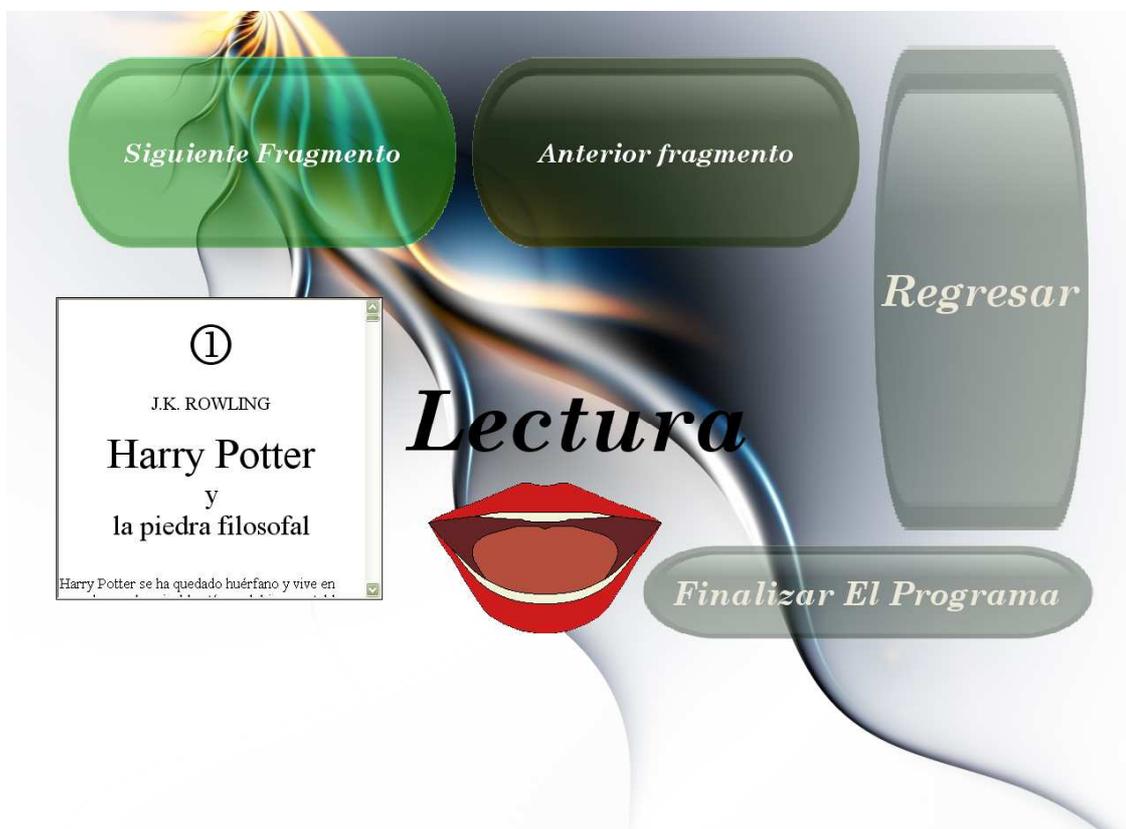


Figura 2.5: Lectura de libros digitales

Para iniciar la lectura el usuario debe presionar el botón siguiente fragmento y se ejecuta la siguiente línea de comando.

```
Private Sub Cmd_adelante()
    RichTextBox1.SetFocus
    SendKeys "{RIGHT}", True

    **** Selecciona asta encontrar los signos ****
    RichTextBox1.Span ".?!:", True, True
    **** Amplía la selección para incluir signos de puntuación.

    RichTextBox1.SelLength = RichTextBox1.SelLength + 1
    DirectSS1.Speak (RichTextBox1.SelText)
End Sub
```

Con el comando *SendKeys "{RIGHT}", True* hacemos que el cursor se mueva hacia la derecha para seleccionar el siguiente fragmento hasta encontrar un signo de puntuación con el comando *RichTextBox1.Span ".?!:", True, True* y luego seleccionar también el signo de puntuación para que el método tenga un uso repetitivo cada vez que se presione el botón *Siguiente Fragmento*, y por ultimo se lea con el sintetizador de voz el fragmento utilizando el comando *DirectSS1.Speak (RichTextBox1.SelText)* permitiendo escuchar el texto seleccionado.

Si el usuario desea repetir el fragmento anterior usamos las siguientes líneas que como las anteriores solo tiene el único cambio en el comando *SendKeys "{LEFT}", True* para retroceder al párrafo anteriormente leído.

```
Private Sub Cmd_atras()
    RichTextBox1.SetFocus
    SendKeys "{LEFT}", True

    **** Selecciona asta encontrar los signos ****
    RichTextBox1.Span ".?!:", True, True

    **** Amplía la selección para incluir signos de puntuación.
    RichTextBox1.SelLength = RichTextBox1.SelLength - 1
    DirectSS1.Speak (RichTextBox1.SelText)

End Sub
```

Botón Escáner, es una aplicación de automatización de procesamiento de texto externo que se encuentra en revistas, libros, comercio etc. Para luego reproducirlo con el sintetizador de voz.

2.1.6 DIGITALIZACIÓN POR MEDIO DE ESCÁNER

Cuando se desea utilizar el procesamiento del escáner se utiliza varios comandos para realizar la automatización completa de todo el proceso que el escáner realiza para el reconocimiento y digitalización de texto que por lo general se ejecuta de forma manual, en esta parte del programa se cumple con el objetivo buscado que el usuario transforme información impresa a voz con tan solo hacer un clic.

Desde la ventana de menú principal haciendo clic en el *botón escáner* pasamos a otra ventana (figura2.5) que ejecuta la automatización del procesamiento de texto, igual que las anteriores ventanas tenemos la ayuda correspondiente para el usuario y lo que sucede al presionar cualquiera de los siguientes botones.



Figura 2.6: Digitalización y lectura de texto

Cuando se presiona el *botón escáner* se ejecuta la siguiente codificación que por medio de la utilización de temporizadores se procesa paso a paso el control del sistema de digitalización y de reconocimiento de caracteres utilizando el OCR comercial.

Para iniciar el procesamiento se necesita de variables que sean ejecutadas en diferentes tiempos.

```
Dim id As Variant
Dim word As Object
```

Cuando se haga clic en el botón de *escanear* se carga el programa de digitalización del escáner con la ejecución de las líneas *id = Shell("C:\Archivos de programa\Lexmark X1100 Series\lxbkaiox.exe", 1)* que son las encargadas de encontrar el programa y ejecutarlo.

```
Private Sub Cmd_escaner()
    Text2.Text = " "
    id = Shell("C:\Archivos de programa\Lexmark X1100 Series\lxbkaiox.exe", 1)
    AppActivate id
    Timer0.Enabled = True
End Sub
```

Luego de ser ejecutado el programa, este se lo mantiene activo con el comando *AppActivate id* hasta cuando sea cerrado. Luego de esta línea se activa un primer temporizador *Timer0.Enabled = True*.

```
Private Sub Timer0_Timer()
    AppActivate id
    SendKeys "{ENTER}", True 'vista preliminar de lo escaneado
    Timer0.Enabled = False
    Timer1.Enabled = True
End Sub
```

Ejecutado el temporizado *Timer0* se mantiene activo el programa del escáner con el comando *AppActivate id* en el que se ejecuta un ENTER con el comando *SendKeys "{ENTER}", True* que se refiere a la inicialización del procesamiento de

digitalización del texto que se encuentre en el escáner, se termina el temporizador desactivándolo con *Timer0.Enabled = False* y activando un nuevo temporizador que continuara con el siguiente paso del procesamiento *Timer1.Enabled = True*.

```
Private Sub Timer1_Timer()
    SendKeys "{TAB 21}", True
    Timer1.Enabled = False
    Timer2.Enabled = True
End Sub
```

El temporizador uno realiza la tarea de hacer 21 TABs los cuales son necesarios para llegar hasta la opción que nos interesa de reconocimiento de texto dentro del menú del software comercial que se encuentra en el escáner utilizado y luego se desactiva el temporizador uno y se activa el nuevo temporizador dos.

```
Private Sub Timer2_Timer()
    SendKeys "{enter}", True 'ejecuta OCR
    Timer2.Enabled = False
    Timer3.Enabled = True
End Sub
```

El temporizador dos realiza la tarea de ejecutar el OCR el cual se encarga de que se realice el reconocimiento de cada uno de los caracteres que conforman el texto a ser analizado y que eventualmente será reproducido con el sintetizador de voz.

```
Private Sub Timer3_Timer()
    AppActivate id
    SendKeys "{TAB}", True 'foco en enviar (al Word)
    AppActivate id
    SendKeys "{ }", True ' , True 'presionamos barra espaciadora q ejecuta la opción enviar(anterior)
    DirectSS1.Speak "Espere un momento, procesando texto"
    Timer3.Enabled = False
    Timer4.Enabled = True
End Sub
```

En el temporizador tres se envía el texto del OCR a Microsoft Word para tenerlo en texto editable, esto se realiza moviendo el foco con el tabulador y usando el comando de automatización *SendKeys "{ }", True ' , True* que es presionar la barra espaciadora para ejecutar la instrucción de envío de texto a Word.

Utilizamos la barra espaciadora porque después de realizar diferentes pruebas con todas las teclas que dispone el teclado y un sin número de combinaciones entre ellas se llegó a la conclusión que es la única tecla aparte del botón derecho del ratón que nos permite realizar la función deseada.

```
Private Sub Timer4_Timer()
    SendKeys "^e", True
    SendKeys "^c", True
    Timer4.Enabled = False
    Timer5.Enabled = True
End Sub
```

Cuando tenemos listo el texto en Word, se activa el temporizador cuatro que realiza la selección de todo el texto *SendKeys "^e"*, trae, luego copia al portapapeles el texto seleccionado *SendKeys "^c"*, *True* terminando el tiempo de su ejecución.

```
Private Sub Timer5_Timer()
    SendKeys "%{F4}", True
    SendKeys "%{F4}", True
    Text2.SetFocus
    Timer5.Enabled = False
    Timer6.Enabled = True
End Sub
```

Después de tener el texto en el portapapeles y con la activación del quinto temporizador, Microsoft Word será cerrado dejando el foco de ejecución en la caja de texto dos. Cerramos Word con la finalidad de no recargar al programa con cada digitalización, ya que al ejecutar una digitalización se estaría acumulando cada vez más editores de texto, por lo tanto cada que se ejecute el botón escanear será como una nueva digitalización.

```
Private Sub Timer6_Timer()
    Text2.SetFocus
    SendKeys "^V", True
    Timer6.Enabled = False
    Timer7.Enabled = True
End Sub
```

Con el foco en la caja de texto se procede a pegar el texto que quedo en el portapapeles ejecutando el temporizador seis que con el comando *Text2.SetFocus* mantenemos el foco en la caja de texto y con *SendKeys "^(\V)", True* pegamos el texto escaneado.

```
Private Sub Timer7_Timer()
    DirectSS1.Speak (Text2.Text)
    Timer7.Enabled = False
End Sub
```

Ya con el texto en el sitio adecuado procedemos a la lectura de este, ejecutando el último temporizador siete que reproduce el texto con el comando *DirectSS1.Speak (Text2.Text)*.

Botón Historias, el botón historias nos lleva a un menú que permitirá escoger la historia que el usuario desee, la misma que previamente estará en formato mp3 y que mediante el reproductor que esta realizado de manera que sea de fácil uso y comprensión, las historias serán escuchadas.

2.1.7 HISTORIAS EN MP3

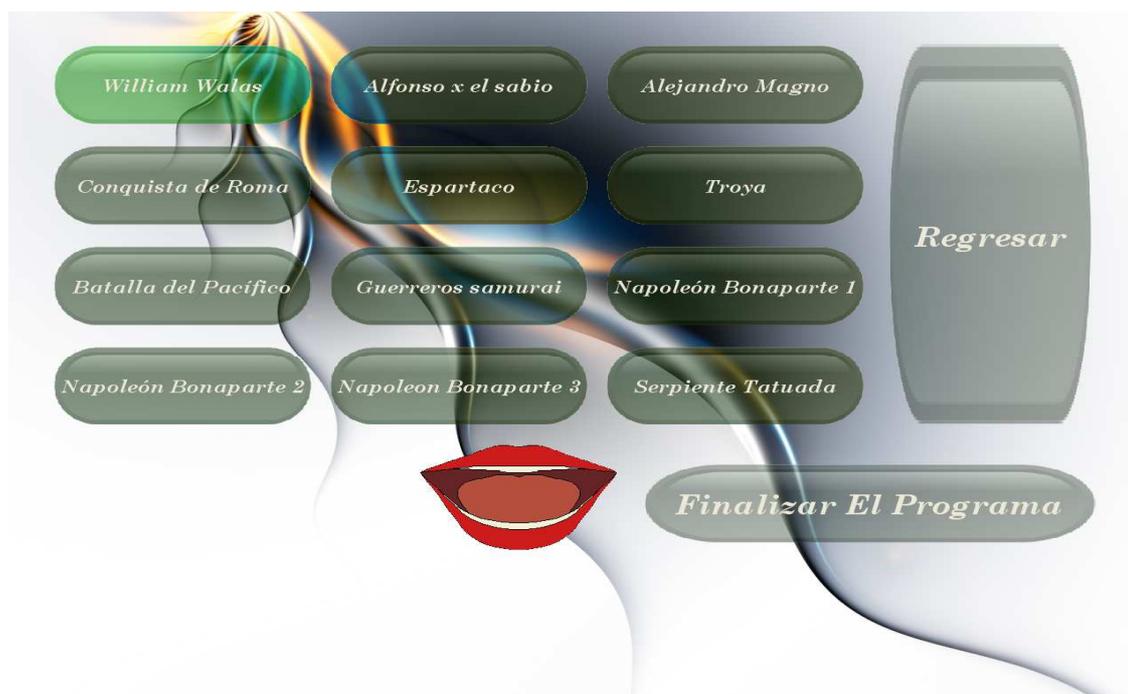


Figura 2.7: Menú de historias en MP3

Encontrándonos en la pantalla de historias MP3 el usuario podrá navegar por los botones que mencionaran los títulos de las historias que están disponibles para escucharlas, si deseamos cambiar a otra historia se presiona la tecla izquierda o derecha del ratón y escucharemos el título de la siguiente historia.

Con el tema escogido, presionamos el botón central del ratón y se abrirá otra ventana que por la ejecución de algunos comandos nos hará escuchar la reproducción de la historia.

La nueva ventana es abierta por el comando *Reproductor.Show* que luego coloca el tema seleccionado en la caja de texto *Text1* para que sea cargado y reproducido activando la opción reproducir de la figura 2.9 permitiendo escuchar el tema seleccionado.

```
Private Sub Cmd1()

    Reproductor.Show
    Reproductor.Text1.Text = "tema1"
    Reproductor.CmdPlay.SetFocus
End Sub
```

```
Private Sub Cmd2_Click()

    Reproductor.Show
    Reproductor.Text1.Text = "tema2"
    Reproductor.CmdPlay.SetFocus
End Sub
```

```
..
Etc, etc
```

Los comandos anteriores nos muestran como se realiza el proceso de cargar el tema al reproductor para poder ser escuchado, cada vez que se realice active el botón reproducir se procesara el tema seleccionado el cual es llamado por una subrutina que reproduce la historia correspondiente, con un salto de ejecución cargando la historia desde el archivo donde se encuentre guardada.

```
Private Sub CmdPlay()

    If Text1.Text = "tema1" Then
        historia1
    End If
```

```

    If Text1.Text = "tema2" Then
    historia2
    End If
    ...
    Etc.
End Sub

```

Después de que se ha seleccionado el tema correcto se carga la historia llamándola desde su respectiva carpeta donde se encuentran los archivos de formato MP3 que se los ejecuta con las siguientes instrucciones.

```

Private Sub historia1()

    WILLIAM_WALLACE = App.Path & "\HISTORIAS\WILLIAM_WALLACE.mp3" ' Ruta del archivo formato
    MP3
    MMControl1.DeviceType = MP3 ' Indica el tipo de sonido ( "WAVEAUDIO" )
    MMControl1.FileName = WILLIAM_WALLACE
    Call Reproducir

End Sub

```

```

Private Sub historia2()

    ALFONSO_X_EL_SABIO = App.Path & "\HISTORIAS\ALFONSO_X_EL_SABIO.mp3" ' Ruta del
    archivo formato MP3
    MMControl1.DeviceType = MP3 ' Indica el tipo de sonido ( "WAVEAUDIO" )
    MMControl1.FileName = ALFONSO_X_EL_SABIO
    Call Reproducir
    ...
    Etc. etc

End Sub

```

Los archivos son cargados al reproductor llamando a la subrutina “*Reproducir*” por medio del comando *Call* que carga el archivo y puede ser escuchado, de la misma manera son utilizadas las subrutinas para pausar o detener la reproducción.

Para que la reproducción sea posible utilizamos el componente de visual Basic *Microsoft Multimedia Control 6.0* (figura 2.8) que realiza la reproducción de archivos de sonido en diferentes formatos como MP3, WAV, etc.



Figura 2.8: Componente *Microsoft Multimedia Control 6.0*

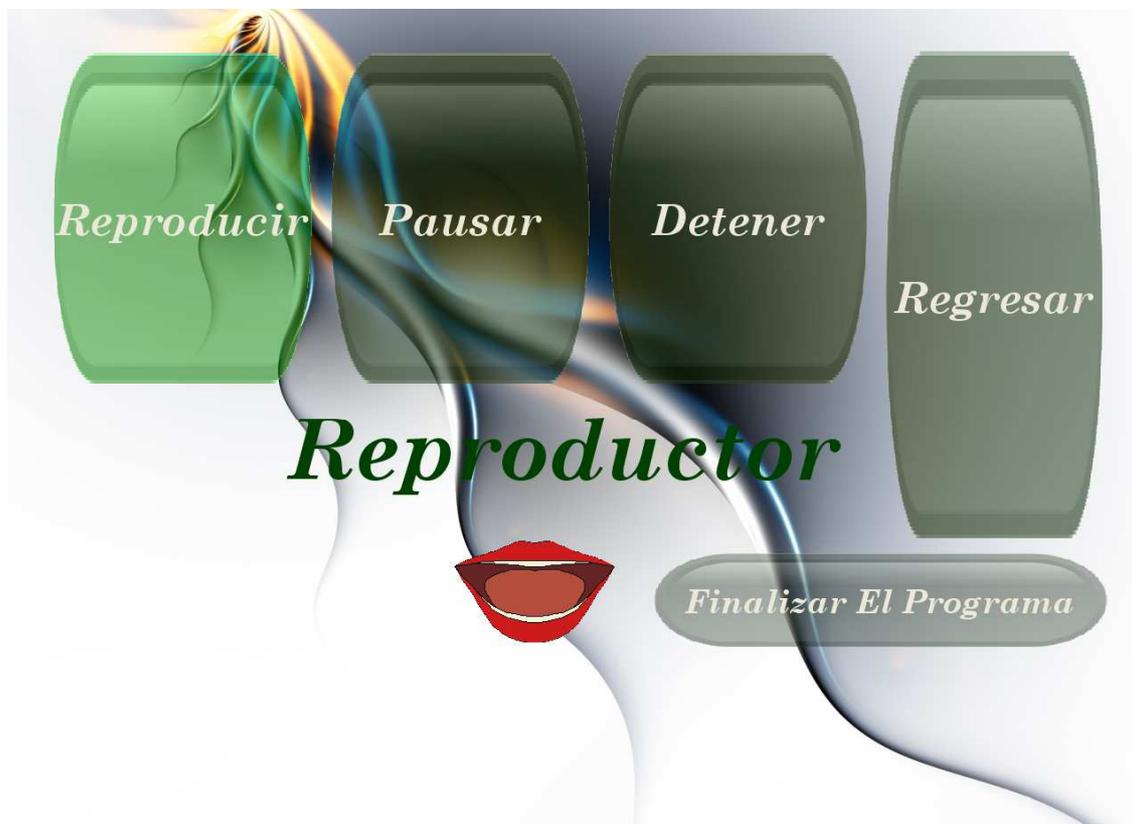


Figura 2.9: Reproductor MP3

Los diferentes botones del reproductor tienen la utilidad de llamar a su operación por medio de subrutinas que ejecutan los comandos por medio del componente *Microsoft Multimedia Control 6.0* que realiza la ejecución de reproducir, pausar y detener el tema elegido, y luego haciendo clic con el ratón se ejecutan comandos diferentes que pasan a otras opciones del programa.

En el caso de que se este reproduciendo un tema, el usuario podrá pausarlo o reproducirlo nuevamente después de una pausa haciendo clic con el ratón se podrá ejecutar estas características propias de un reproductor y que como tal fueron tomadas en cuenta y por ende también diseñadas en este proyecto.

```
Private Sub CmdPause()
```

```
    Pausar
```

```

    CmdPlay.SetFocus
End Sub

```

```

Private Sub CmdStop()
    Detener
End Sub

```

2.1.7.1 Subrutinas utilizadas en el reproductor:

Los siguientes comandos son los que ejecutan el componente de reproducción de archivos de audio, que realiza las diferentes operaciones de reproducir, pausar o detener la reproducción.

```

Private Sub Pausar()

    If MMControl1.Mode <> mciModePlay Then
        Exit Sub
    End If
    MMControl1.Command = "pause" ' Indica el Comando ( pausar sonido )
    Do
        DoEvents
    Loop Until MMControl1.Mode = mciModePause
    CmdPlay.Enabled = True
End Sub

```

```

Private Sub Reproducir()
    If MMControl1.Mode = mciModeNotOpen Then
        MMControl1.Command = "open" ' Indica el Comando Abrir o reproducir
    End If

    If MMControl1.Mode = mciModePause Then
        MMControl1.Command = "pause"
    Else
        MMControl1.Command = "play"
    End If
    Do
        DoEvents
    Loop Until MMControl1.Mode = mciModePlay
End Sub

```

```

Private Sub Detener()
    ' Comando para detener el sonido o cerrarlo

```

```
MMControl1.Command = "stop"  
MMControl1.Command = "close"
```

```
Do  
    DoEvents  
    Loop Until MMControl1.Mode = mciModeNotOpen  
    CmdPlay.Enabled = True  
    MMControl1.Visible = False  
End Sub
```

2.1.8 BOTÓN IMÁGENES

Este botón es el acceso a dos tipos de imágenes prediseñadas: para el prototipo de imágenes en matriz de LEDs y en matriz de tacto. Al hacer clic en el botón imágenes tenemos acceso a la siguiente pantalla donde podemos escoger para la demostración de los prototipos diseñados, sea en imágenes de luz de LEDs o en tacto.

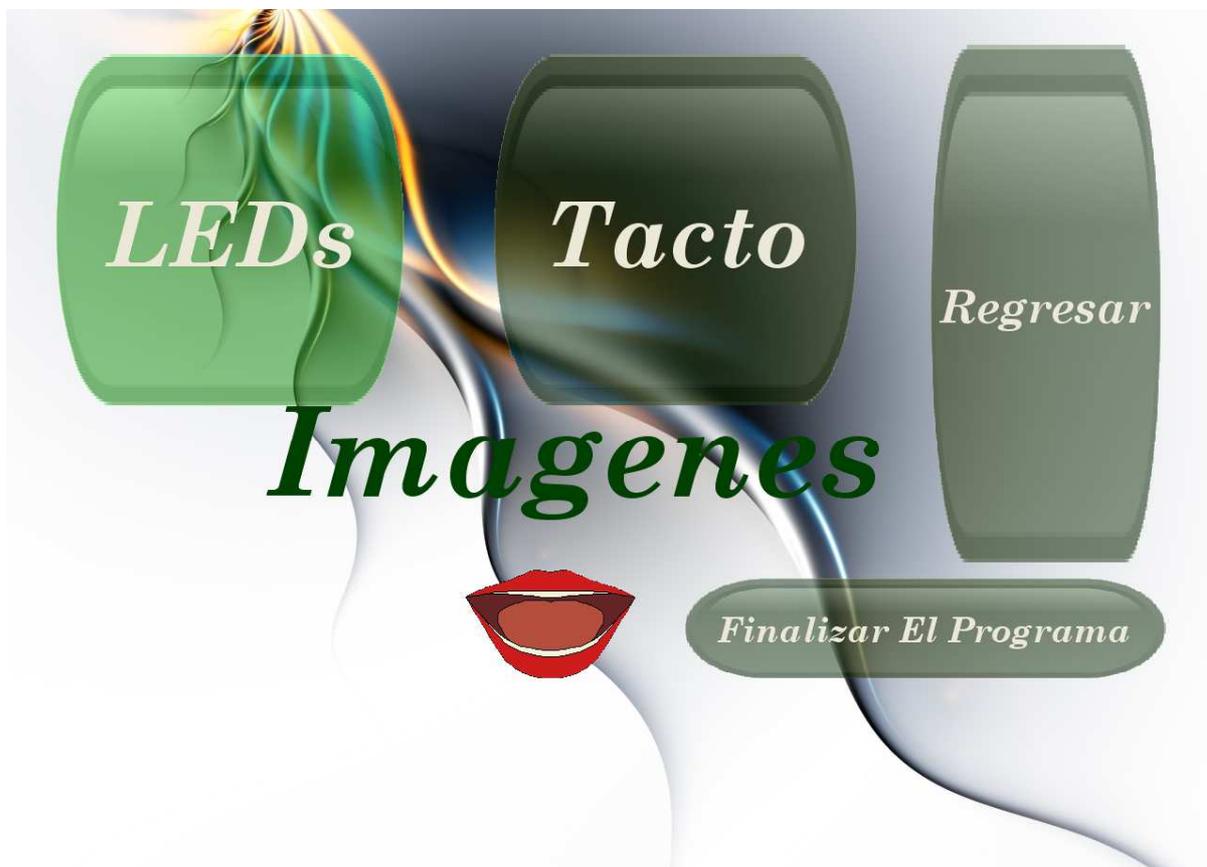


Figura 2.10: Menú de imágenes prediseñadas

2.1.9 Botón LEDs

Al escoger este botón nos permitirá entrar a una nueva ventana que contiene una variedad de imágenes prediseñadas que la matriz puede soportar debido a la limitación en cuanto a número de píxeles se refiere, mostraremos imágenes que puedan ser su forma apreciable y entendible.



Figura 2.11: Imágenes prediseñadas

La figura 2.11 muestra los nombres de las diferentes imágenes que se ha diseñado para muestra del prototipo, la resolución que tomamos es de imágenes que sean apreciables a la vista y muestren el funcionamiento correcto del proyecto diseñado.

Las figuras diseñadas tienen una resolución de 24 filas por 32 columnas, es decir 768 píxeles, cada píxel esta representado con un diodo emisor de luz, que para nuestro proyecto se lo remplazara por un pin de plástico que hará la función de un píxel de tacto de una imagen en alto relieve formada por pines.

En este caso utilizamos comunicación serial entre el computador y el microcontrolador para enviar los datos de la imagen y sacarla a la vista que en nuestro caso seria una imagen que se pueda reconocer al tacto.

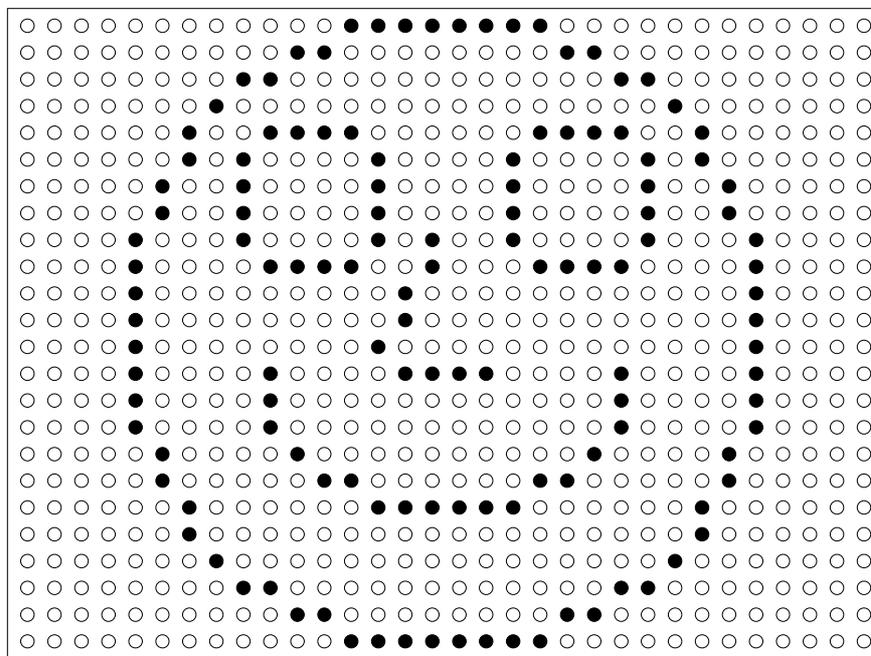


Figura 2.12: Imagen carita feliz

Para mostrar el grafico 2.12 se ha enviado 768 datos, conformados solo por ceros y unos que comprenden en apagar o encender un LED respectivamente. Estos datos inician un barrido de derecha a izquierda y de arriba hacia abajo tomando 12 filas simultaneas, pasando columna por columna hasta tomar la forma deseada, se toma este número de filas por columna por la múltiplexación de la matriz optada para su diseño que es de 4 submatrices de 12 x 16 para tener un mejor desempeño de trabajo y disminuir los costos de diseño ya que el propósito del proyecto no es mostrar imágenes en luz sino en una matriz táctil que se desarrolla mas adelante.

Para que exista una comunicación serial debemos utilizar uno de los componentes de Visual Basic que realiza la comunicación con periféricos externos como en nuestro caso una matriz. El componente utilizado por nosotros es *Microsoft Comm Control 6.0* que inicia su funcionamiento abriendo el puerto con el comando `MSComm1.PortOpen = True`, para iniciar la comunicación y luego

con el comando *MSComm1.Output*, se realiza la transmisión de datos hacia el microcontrolador que estará esperando información para ser procesada y pasar a encender los respectivos puntos.

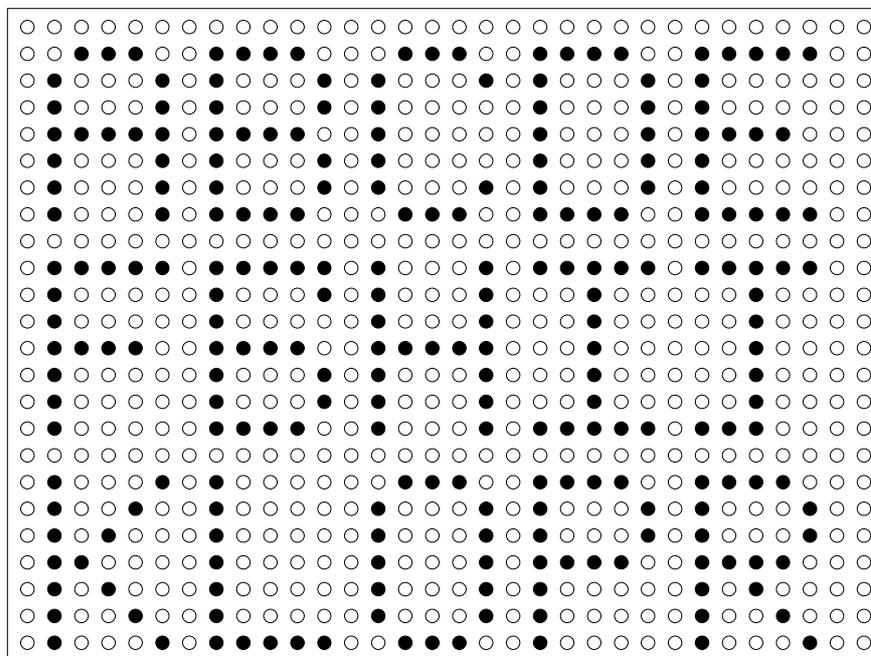


Figura 2.13: Imagen de texto

La imagen anterior muestra uno de los resultados esperados de este proyecto y la utilidad del mismo en cuanto a enseñanza y aprendizaje de personas con discapacidad visual como puede ser el conocimiento de la forma de las letras por medio del tacto, además pueden ser imágenes de diferentes objetos.

2.1.10 Botón Tacto

En esta ventana mostramos las imágenes que podemos reconocer por medio del tacto, utilizando un prototipo de matriz táctil de dimensión de 8 x 8 píxeles, es decir que tenemos 64 pines donde podemos formar imágenes de una resolución aceptable de tal manera que puedan ser identificadas sin ninguna confusión.

Se tomo esta resolución debido a que nuestro trabajo es un prototipo por lo que no es necesaria una gran resolución ya que solo es de modo demostrativo lo cual significa que puede ser ensamblado en mayor cantidad de píxeles para mostrar imágenes grandes y con mayor complejidad en cuanto a forma se refiere, como

mapas o rostros. La programación para este prototipo es la misma que para la matriz de LEDs es decir que enviamos los datos de la misma manera que en la ventana de imágenes para matriz de LEDs.

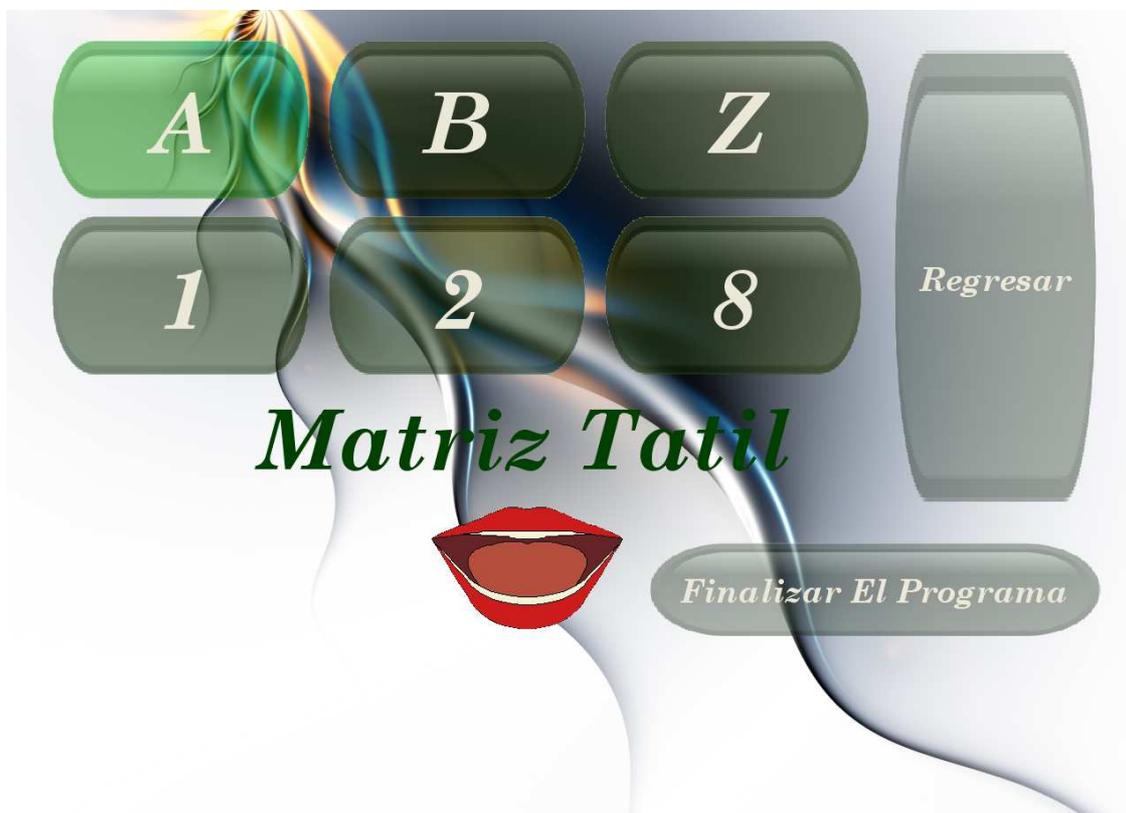


Figura 2.14: Imágenes para Tacto

Las imágenes prediseñadas para tacto tienen una resolución de 8 x 8 píxeles, resultando de esto un tamaño de matriz que físicamente cabe en la palma de la mano para una mejor identificación mediante el sentido del tacto. Estas características son aceptables desde el punto de vista demostrativo debido que nos permite presentar lo más esencial como son: números, letras y también objetos de dirección como flechas y símbolos que se pueden representar en una matriz pequeña.

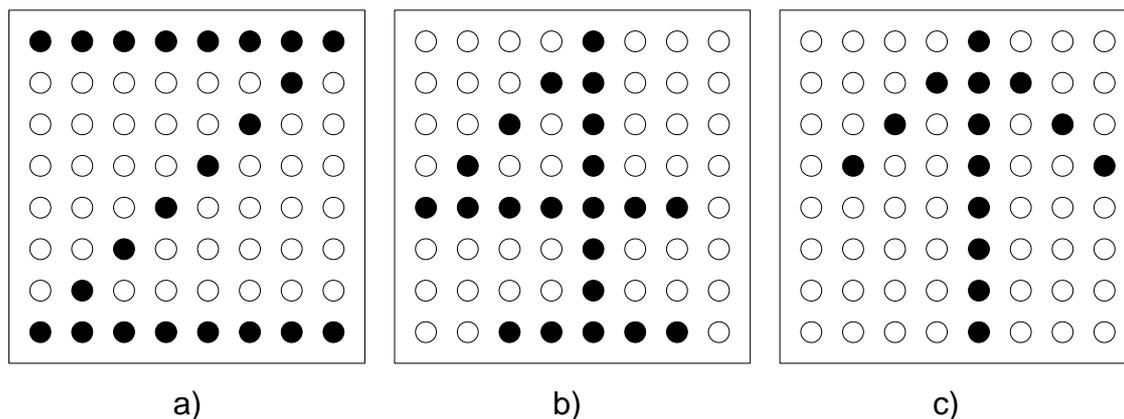


Figura 2.15: Imágenes para Tacto

2.1.11 Botón WebCám

Esta aplicación es utilizada con la ayuda de un laso a otro programa de procesamiento de imágenes como MATLAB que también es utilizado en nuestro proyecto. Este programa es utilizado junto con Visual Basic para realizar una aplicación de capturar imágenes por medio de una Webcam procesarlas en MATLAB en forma binaria, guardar los datos que necesitamos en un archivo de extensión TXT , llamar a este archivo desde visual Basic con la ayuda de algunos componentes y enviar estos datos por comunicación serial al microprocesador y observar la gráfica tomada por la Webcam con una menor resolución y con la idea de que pueda ser de forma táctil para personas no videntes.

En la figura 2.16 observamos la distribución de procesamiento que se realiza, en el recuadro de la izquierda se mostrara la figura que se ha tomado por la Webcam y en la derecha la imagen procesada a un número menor de píxeles, en nuestro caso a la resolución de la matriz de LEDs diseñada de 24 x 32 píxeles.

El procesamiento es automático en su totalidad, cuando inicia la ejecución del programa se le guiara al usuario que debe esperar un tiempo predeterminado, para luego ser advertido por medio de la voz sintetizada que el procesamiento a finalizado correctamente y puede tocar la imagen final en la matriz, en nuestro caso esto se debería realizar en la matriz táctil que por ser a nivel demostrativo, el prototipo tiene baja resolución y es de 8 x 8 y una imagen tomada por la Webcam

no es notable con estas dimensiones, para lo cual se optó por hacer también una matriz de 24x32 pero basada en LEDs, no se realizó una matriz de pines de mayor dimensión debido a factores de costo y sobre todo debido a que la mayoría de los elementos usados en el módulo de matriz táctil fueron realizados manualmente porque todo fue producto de una larga investigación basada en experimentación continua probando cada elemento que conformaría la mecánica de la matriz táctil. Las dimensiones de la matriz que realizamos en el proyecto son en base a pruebas con la Webcam obteniendo la menor resolución posible que tiene por defecto la cámara y procesando la imagen por medio de un algoritmo para bajar a menor número de píxeles su resolución y pueda ser construida como prototipo.

Una mejor calidad de representación de las imágenes es con una mayor resolución en número de píxeles, con mayor resolución los gráficos pueden ser más detallados como con más rasgos en los rostros o en la captura de imágenes que contienen objetos con más detalles en cuanto a su forma, pueden ser más visibles a una escala determinada.

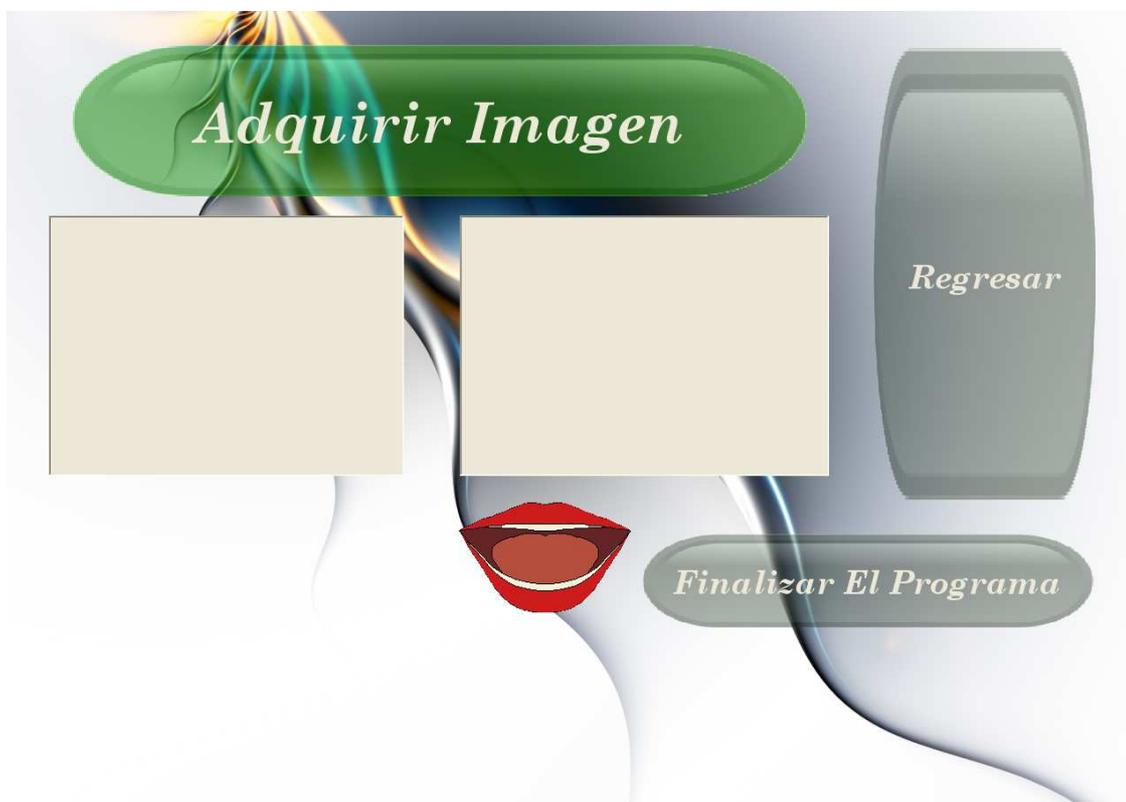


Figura 2.16 Menú para procesamiento de imágenes

2.1.12 PROCESAMIENTO DE IMÁGENES

Capturamos las imágenes utilizando una Webcam que nos da una mínima resolución de 160 x 120 píxeles, dato que encontramos en las especificaciones de la cámara, la misma que tiene inconvenientes al existir bastante luz en el ambiente en el que se desarrolla la práctica, comparada con otras cámaras, es en su mayoría utilizada para uso por Internet sin mayores características de video, en nuestro caso la tomamos por su bajo costo, en el mercado hay a disposición Webcams de gran calidad en cuanto a resolución se refiere pero con costo elevado.

El procesamiento sigue una serie de pasos que se ejecutan automáticamente siguiendo sus respectivos comandos en un tiempo determinado para lograr el objetivo deseado. En el siguiente diagrama se muestra el procesamiento para obtener la imagen a una menor escala y enviarla al periférico final.

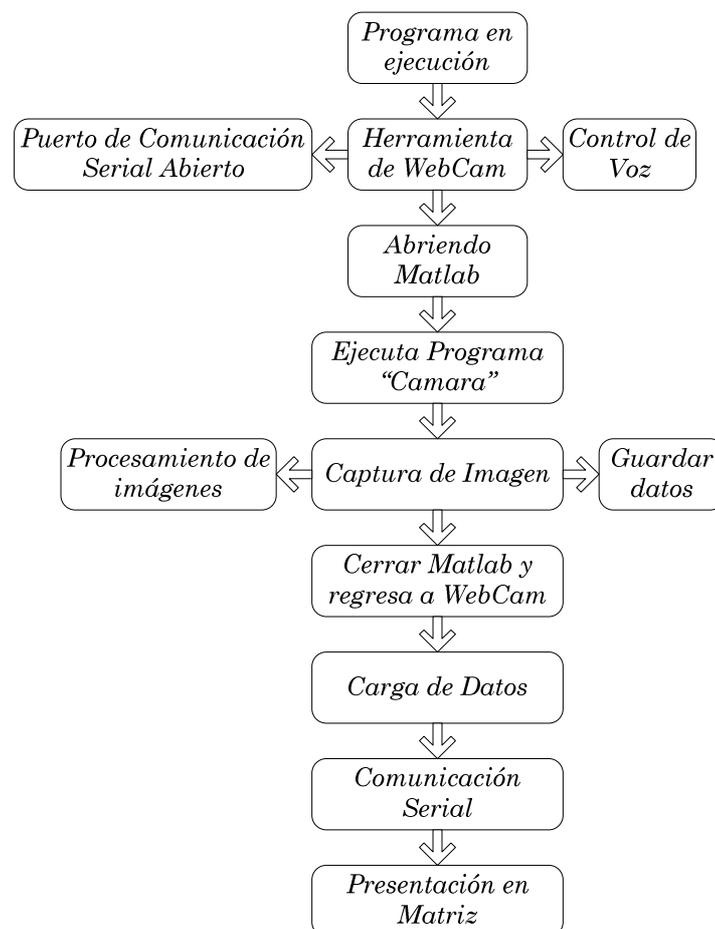


Figura 2.17: Captura y procesamiento de imágenes

Cuando se ejecuta la herramienta de Webcam se inicia abriendo el puerto de comunicación serial, si se presiona el botón *adquirir Imagen* se abre el programa de procesamiento de imágenes MATLAB con su comando *id = Shell("C:\MATLAB7\bin\win32\MATLAB.exe", 6)* que realiza el llamado de este, desde el disco duro donde se encuentra guardado.

```
Private Sub Cmd_WebCam()

    ' **** Abrimos Matlab
    id = Shell("C:\MATLAB7\bin\win32\MATLAB.exe", 6)

    ' **** Activado Matlab
    AppActivate id
    'Abilito el Timer para capturar los datos de Matlab
    Timer1.Enabled = True
End Sub
```

El comando *AppActivate id*, permite que permanezca activo el programa ejecutado, para realizar las diferentes actividades, *id* es la variable con la que se le ha nombrado al programa, además habilitamos un *Timer* para ejecutar más instrucciones.

```
Private Sub Timer1_Timer()
    'Escribe en el matlab "camara"
    SendKeys "camara", True
    SendKeys "{enter}"
    Timer1.Enabled = False
    Timer2.Enabled = True
End Sub
```

El *Timer1* con la instrucción *SendKeys "camara"*, escribe en MATLAB automáticamente el nombre del programa que realizara el procesamiento de la imagen capturada para ejecutarlo y realizar el procesamiento.

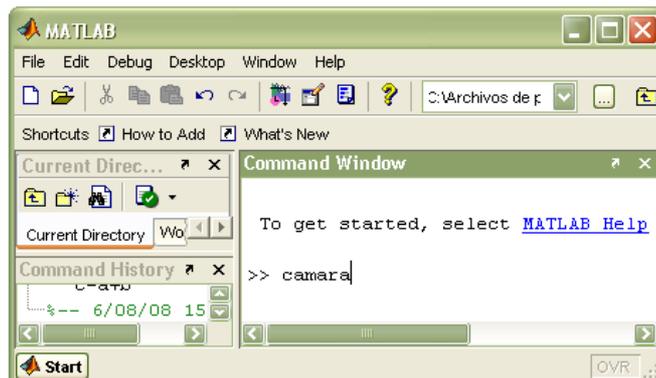


Figura 2.18: Matlab llamando programa

```
Private Sub Timer2_Timer()
    AppActivate id
    ' **** Cerramos Matlab
    SendKeys "exit", True
    SendKeys "{enter}"

    ' **** Recopilamos datos
    Timer2.Enabled = False
    Timer3.Enabled = True
End Sub
```

El siguiente *Timer2* cierra MATLAB después de guardar la información que necesitamos en Visual Basic para enviar al prototipo realizado.

```
Private Sub Timer3_Timer()
    '**** Recopilación de datos para transmitir
    RichTextBox1.LoadFile ("D:\Proyecto\visualBasic\Ejemplos\Voz En Pc\camara.txt")
    Timer3.Enabled = False
    Timer4.Enabled = True
End Sub
```

El *Timer3* realiza la carga de los datos que se obtuvo en el programa de procesamiento de imágenes utilizando el componente *Microsoft Rich Textbox Control 6.0* que tiene comandos de lectura de archivos solamente dándole la dirección de donde se encuentra guardado el archivo específico, en nuestro caso *camara.txt* que contiene los datos separados por comas, para obtener solo los datos, utilizamos el comando *Replase* que se lo realiza en el *Timer4*

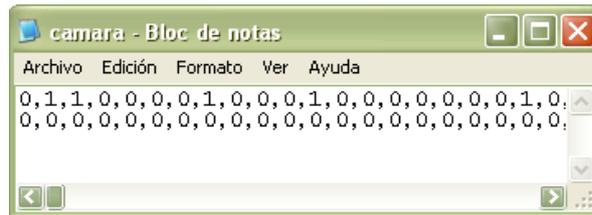


Figura 2.19: Archivo que guarda la información de puntos

```
Private Sub Timer4_Timer()
    remplaso = RichTextBox1.Text
    remplaso = Replace(remplaso, ",", "")
    Text3.Text = remplaso
    Timer4.Enabled = False
    Timer6.Enabled = True
End Sub
```

El siguiente *Timer6* realiza la carga de las imágenes en un cuadro *Picture* para visualizar la imagen original y la procesada para poder comparar las diferencias que existe en estas cuando se disminuye notablemente el número de píxeles utilizados.

```
Private Sub Timer6_Timer()
    Set x = LoadPicture("D:\Proyecto\visualBasic\Ejemplos\VozPc\foto.bmp")
    Set Picture1.Picture = x
    Set Y = LoadPicture("D:\Proyecto\visualBasic\Ejemplos\VozPc\imagenbin.bmp")
    Set Picture2.Picture = Y
    Timer6.Enabled = False
    Timer7.Enabled = True
End Sub
```



a)



b)

Figura 2.20: Archivo que guarda la información de puntos

La figura 2.20a muestra la imagen original tomada por la Webcam a una resolución de 120 x 160 píxeles y la figura 2.20b muestra una imagen binaria que ha sido filtrada y pasada a una resolución de 24 x 32 píxeles

```
Private Sub Timer7_Timer()
    MSComm1.Output = Text3.Text & vbCr
    Timer7.Enabled = False
End Sub
```

El último *Timer7* realiza la comunicación serial con el prototipo de matriz de diodos, enviando la información correspondiente para formar la imagen que se capturo con la Webcam.

2.1.12.1 Procesamiento en MATLAB

Utilizamos este programa conjuntamente con Visual Basic para obtener la imagen capturada por la Webcam, controlando automáticamente sus herramientas y llevándonos los datos que necesitamos a un componente de visual Basic.

Programamos un archivo.m que le llamamos “*cámara*” donde colocamos la programación que realizara el procesamiento de imágenes capturadas.

Las primeras líneas realizan la operación de limpiar todas las variables y cerrar todas las aplicaciones que se encuentren abiertas en MATLAB, esto se realiza para tener un proceso limpio en cuanto a ejecución se refiere.

```
clear all
close all

%   *** LECTURA DE LA CAMARA ***
%   *** Definición del tipo de cámara y su resolución
```

Tomamos la variable *vid* para asignarle con este nombre a la cámara con sus respectivas especificaciones de resolución, luego del número de imágenes que se cargarán para realizar la captura, solamente escoge una de ellas y será la que

presente mejores características y buena resolución para luego guardarla con la variable asignada que para nuestro caso es foto.bmp.

```

vid = videoinput('winvideo',1,'RGB24_160x120');
preview(vid)           % Abre la pantalla de video en windows
pause(1)
closepreview(vid);     % Cierra la pantalla de video
pause(1)               % Pausa de 1 s para tomar buena imagen
imagen = getsnapshot(vid); %Guarda un cuadro del video

foto= imresize(imagen,[240 320],'nearest');
imshow(foto);

saveas(gcf,'foto.bmp') %Guarda la imagen con nombre 'foto.bmp'

I=rgb2gray(imagen);    % Escala de Grises

BWS=edge(I,'canny');   % Determina los bordes de la imagen

Luego de tener la imagen de una forma binaria determinada, solo los bordes
pasaran por un filtro para borrar puntos innecesarios, para continuar con la
minimizaci3n de resoluci3n.

%***** comprimiendo imagen ****

[filas, colu]=size(BWS); % Tama1o original de la imagen

Filas=24;                % N1mero de filas y columnas a convertir
Columnas=32;
IMN={};
n=5;                    % Factores de cuadros a ser reemplazados
y=1;

for i=1:Filas
    for j=1:Columnas
        a=(i-1)*n+1;
        b=n*i;
        c=(j-1)*n+1;
        d=n*j;
    
```

```

    IMN{y}=BWS(a:b,c:d);
    A=sum(IMN{y});
    B=sum(A');
    C=n*n/6;

    if B>C
        NUEVA(i,j)=1;
    else
        NUEVA(i,j)=0;
    end
    y=y+1;
end
end
imagenes=[NUEVA];

% ***** datos para guardar

dato={};
datoin={};
clear dato;
for k=1:12

    %***** primeros datos superiores de la matriz *****

    dato{k}=imagenes(k:k,1:8);
    q=k+12;
    dato{q}=imagenes(k:k,9:16);
    r=k+24;
    dato{r}=imagenes(k:k,17:24);
    s=k+36;
    dato{s}=imagenes(k:k,25:32);

    %***** segundos datos inferiores de la matriz *****
    % **** iniciamos desde el lado derecho de la matriz **
    % **** Ordenamos los datos para ser guardados ***

    z=k+12;
    t=k+48;
    datoin{t}=imagenes(z:z,25:32);

```

```
dato{t}=fliplr(datoin{t});

u=k+60;
datoin{u}=imagenes(z:z,17:24);
dato{u}=fliplr(datoin{u});

v=k+72;
datoin{v}=imagenes(z:z,9:16);
dato{v}=fliplr(datoin{v});

w=k+84;
datoin{w}=imagenes(z:z,1:8);
dato{w}=fliplr(datoin{w});
end

%***** Guardamos los datos en el archivo camara.txt *****
dlmwrite('camara.txt', dato, 'precision', '%.0f', 'newline', 'pc')

% ***** Guardamos la imagen q se esta procesando

L= imresize(imagenes,[240 320],'nearest');
figure
imshow(L);

saveas(figure(2),'imagenbin.bmp')

% Cerramos las imagines que se encuentran en pantalla

close(figure(1));
close(figure(2));
```

2.2 DISEÑO DE HADWARE DEL PROTOTIPO

2.2.1 Diodo emisor de luz

Diodo emisor de luz, también conocido como LED (*Light-Emitting Diode*) es un dispositivo semiconductor (diodo) que emite luz incoherente de espectro reducido cuando se polariza de forma directa la unión PN del mismo y circula por él una corriente eléctrica.

Para obtener una buena intensidad luminosa debe escogerse bien la corriente que atraviesa el LED; para ello, hay que tener en cuenta que el voltaje de operación va desde 1,8 hasta 3,8 voltios aproximadamente (lo que está relacionado con el material de fabricación y el color de la luz que emite) y la gama de intensidades que debe circular por él varía según su aplicación. Valores típicos de corriente directa de polarización de un LED están comprendidos entre los 10 y los 40 mA.

2.2.2 BANCO DE LEDS

Es una colección de LEDs conectados con un extremo en común, donde la resistencia limitadora puede o no ser común: Cuando el número de LEDs es elevado, el acceso a los mismos suele multiplexarse para reducir el pinout y el consumo de potencia. Cuando un banco de LEDs se multiplexa en todo momento sólo un LED está encendido por lo que la información que muestra debe refrescarse periódicamente. Ver Fig. 3.1

Un caso particular de banco de LEDs son los displays de 7 segmentos, que constan de 7 LEDs para los segmentos y uno o dos LEDs adicionales para los puntos decimales. De la misma forma los displays de 7 segmentos se suelen agrupar en bancos de displays con acceso multiplexado: a) LEDs pertenecientes al mismo display, comparten cátodo (lógica directa); y, b) LEDs en la misma posición en displays diferentes comparten ánodo (lógica inversa.)

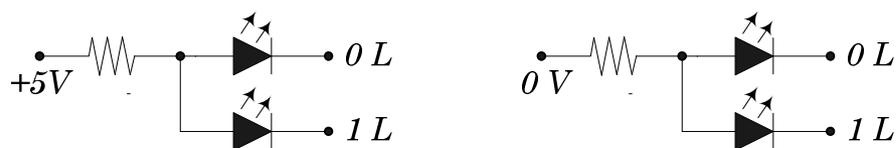


Figura 2.21: a) ánodo común (lógica inversa), b) cátodo común (lógica directa)

2.2.3 MATRIZ DE LEDS

Es un conjunto de LEDs dispuestos en filas y columnas con acceso multiplexado. Todo LED comparte un terminal con todos sus vecinos de fila y el otro terminal con todos sus vecinos de columna.

Existen dos tipos de matrices:

- Matrices con filas en cátodo común y columnas en ánodo común.
- Matrices con filas en ánodo común y columnas en cátodo común.

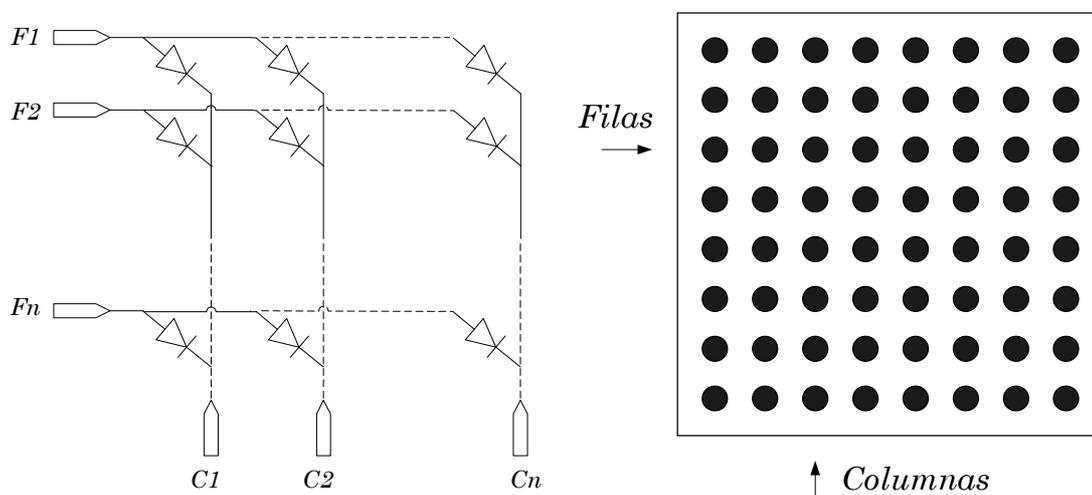


Figura 2.22: Matriz de LEDs (filas ánodo común, columnas cátodo común)

2.2.4 FRECUENCIA DE REFRESCO O BARRIDO

Es la frecuencia a la que un LED se selecciona. El ojo humano detecta como parpadeo frecuencias inferiores a 50 Hz. (periodo 20 ms). El barrido es un efecto

visual que permite visualizar imágenes sin que el ojo humano se percate que éstas se encuentran en continua formación, las imágenes formadas pueden encontrarse estáticas o en movimiento y esto formara un proceso cíclico de activación. El dato y la columna deben habilitarse en el mismo intervalo, de lo contrario no habría sincronismo, y el resultado sería colocar datos en columnas erradas. Cada intervalo de tiempo de activación es el mismo por lo cual se entiende que el proceso de activación es periódico.

En efecto, el inverso del periodo es la frecuencia, por tanto una señal periódica con cierta frecuencia genera el proceso de barrido. Esta frecuencia se denomina frecuencia de barrido. Por tanto si se requiere barrer el carácter "A" a una frecuencia de barrido de 600 Hz, su período es 1.666 ms; esto significa que cada columna se activa en este intervalo de tiempo, o dicho de otra forma, en 1 segundo se activan 600 columnas, pero el carácter se forma con tan solo 5 columnas, de lo que se deduce:

$$\frac{600Hz}{5columnas} = 120 \frac{Hz}{columna}$$

Eq 2.1 Frecuencia de refresco o barrido

Cada columna se activa en un segundo 120 veces, o el dato es refrescado 120 veces en la misma columna de LEDs, el ojo humano no puede apreciar este número de refresco por lo que se visualiza simplemente un dato encendido permanentemente. Si la frecuencia de barrido disminuye a valores menores de 50Hz, el ojo humano visualiza leves parpadeos, si la frecuencia disminuye aun más este parpadeo se presenta más acentuado, y desde luego la visualización del carácter se distorsiona.

El barrido en una matriz de LEDs $m \times n$, donde m son el número de filas y n el de las columnas, puede ser:

Por filas: cuando el barrido se realiza de arriba hacia abajo o de abajo hacia arriba, en la posición frontal de la matriz $m \times n$; cada fila debe activarse en un intervalo de tiempo o periodo de barrido.

Por columnas: cuando el barrido se realiza desde izquierda a derecha o derecha a izquierda, en la posición frontal de la matriz, en este caso cada columna debe activarse en un periodo de barrido.

Por filas y columnas: cuando el barrido se realiza conjuntamente por filas y por columnas. En este caso se activa una columna y una fila determinada en un periodo de barrido, en otras palabras es la activación de un punto de la matriz en este periodo. Si no hay perfecto sincronismo de barrido de filas y columnas, se producirá error de activación de los puntos que deben seguir la secuencia, en tal caso el punto que debe recibir el dato lo recibirá otro y no se encenderá, produciendo a nivel global una distorsión del símbolo o de la figura que se quiere mostrar.

2.2.5 TIEMPO DE PERSISTENCIA

Tiempo durante el cual un LED, una columna o una fila de LEDs están seleccionados es decir es el tiempo de barrido de un punto de matriz, de una columna o fila de la misma; conforme disminuye la persistencia disminuye la luminosidad del semiconductor.

La frecuencia de refresco o barrido y la persistencia están relacionadas según el grado de multiplexación de acuerdo a la siguiente expresión:

Tiempo de persistencia x elementos a refrescar = periodo de refresco

Si el número de elementos a refrescar es igual a 1 LED, ó 1 columna / fila de LEDs, entonces el tiempo de persistencia es igual al tiempo o de barrido, por tanto siempre y cuando se cumpla esta condición, el tiempo de persistencia será el periodo de barrido medido en milisegundos, microsegundos. El caso contrario es tener varios elementos a refrescarse; por ejemplo, en una matriz 7×5 cada

columna se activa con un tiempo de persistencia: a (ms), usando la relación anterior se tiene que el periodo de refresco de la matriz es: $5a$ (ms).

2.2.6 FACTOR DE SERVICIO

El factor de servicio o relación de trabajo se define como el cociente entre el tiempo que el LED permanece encendido y el tiempo que permanece apagado. Por tanto, si se tiene una matriz de 7×5 , y se barre por columnas, cada una de las columnas se activará durante un periodo de refresco, y permanecerán durante cuatro periodos apagados, su factor de servicio es 0.25 o expresado en porcentaje 25%; pero si la matriz de LEDs tiene un número considerable de columnas el tiempo de una columna permanecerá inactiva es aproximadamente igual a la suma del periodo de activación de la columna más el tiempo de apagado y se tiene la siguiente expresión:

$$F_S = \frac{t_{ENCENDIDO}}{t_{APAGADO}} \cong \frac{T_{BARRIDO}}{T_{BARRIDO} \cdot N} = \frac{1}{N} = \frac{100}{N} \%$$

Eq 2.2 Factor de servicio

El tiempo de apagado de una columna es aproximadamente en valor al periodo de refresco de la matriz expresado como el periodo de barrido de una columna multiplicado por el número de columnas de la matriz: N , relación especificada en Eq. 2.1 En conclusión, en la expresión anterior, el factor de servicio expresado como porcentaje es $100/N \%$, siendo N el número de columnas de una matriz de LEDs $M \times N$; esta expresión se aplica en filas como: $100/M\%$, siendo M el número de filas de la matriz $M \times N$.

2.2.7 EFECTOS DE UNA SEÑAL DE BARRIDO O DE PULSOS

La relación entre la intensidad luminosa y la corriente que circula por el LED es fuertemente a lineal, si se duplica la corriente no se duplicará la intensidad de la luz. Este fenómeno se atenúa a altas corrientes donde se produce un efecto de

saturación, y donde el aumento de temperatura en el LED hace decrecer la eficiencia. Por eso, si se compara la intensidad luminosa de un LED excitado con una corriente continua de 20 mA con otro excitado con una corriente pulsada de 40 mA y una relación de trabajo del 50%, el segundo LED se verá más luminoso.

Aunque pareciera que relaciones de trabajo muy pequeñas con picos de corriente muy altas es lo más del cual la vida útil del LED empieza a disminuir. La elevación de la temperatura del LED también disminuye su vida, y es típico que en un incremento de 40°C en el semiconductor reduzca su vida útil cerca de 10 veces.

2.2.8 NUMERO DE LEDs

Definir el número de LEDs que van a conformar la matriz, para lo cual se debe considerar además del tipo de encapsulado, el ángulo de visibilidad y la intensidad luminosa de los mismos, la manera en que se van a manejar los datos que se desean mostrar y el nivel de resolución, es necesario considerar con qué valor de intensidad luminosa se va a trabajar, por lo cual se recomienda no exceder las 800 candelas para el rojo. El tener un gran número de LEDs permitiría una muy buena resolución, pero a la par afectaría la rentabilidad del proyecto, debido a que estos dispositivos actualmente tienen un elevado costo. Por estos motivos para el diseño de la matriz del prototipo se van a utilizar 768

2.2.9 FUNCIONAMIENTO DEL CONTROL

Los circuitos digitales existentes manejan información orientada al byte. El circuito integrado 74LS373, el cual es un latch transparente compuesto de 8 flip-flops tipo D, es el que se va a encargar de la captura de los datos. De acuerdo con el número de líneas de control en las columnas de la matriz (64), serán necesario 8 latches (2 para cada submatriz), mientras que el circuito encargado de la habilitación de cada uno de los anteriores, será el decoder 74LS138, el cual es un decodificador de 3 a 8; estos dos circuitos se pueden conectar directamente, ya que las salidas del decoder al igual que la habilitación de los latches son activadas en 0 lógico.

Para manejar un elemento de una matriz, es necesario tener un control tanto de la fila como también de la columna a la que pertenece dicho elemento. Debido a la multiplexación de los datos y por las características de los circuitos integrados a utilizar, simultáneamente se encenderán 8 LEDs de una misma columna, por lo que se hace necesario utilizar para cada columna un transistor que conmute la cantidad de corriente necesaria, debido a que los requerimientos de corrientes son mayores por el hecho de alimentar varios LEDs a la vez. Para obtener una intensidad luminosa aceptable dentro del rango recomendado, es necesario que por cada LED circule una corriente de 20 mA; por lo tanto, como fuente de corriente constante para las filas (y columnas) se utilizará el circuito ULN2803, el cual proporciona hasta 500 mA de salida. Los transistores a utilizar son del tipo PNP.

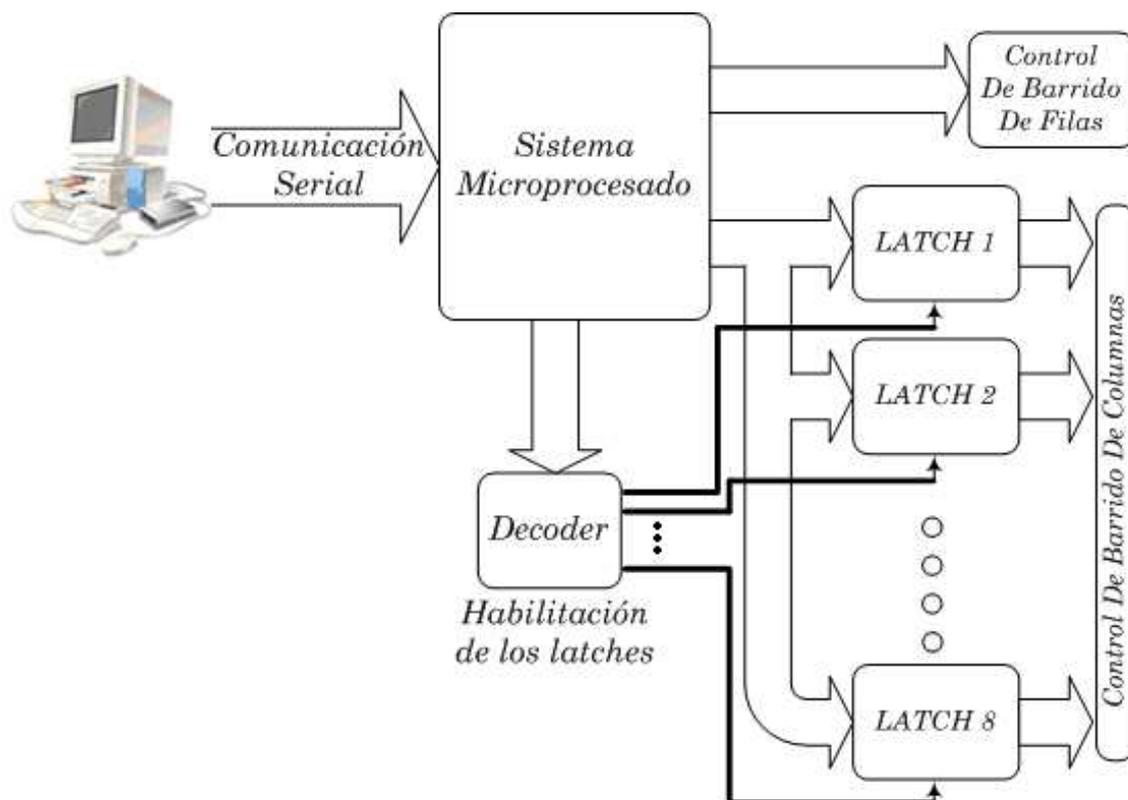


Figura.2.23 Sistema del prototipo en bloques con todas las consideraciones anteriores

2.2.10 DISEÑO DE LA MATRIZ DE LEDS

2.2.10.1 FUENTE DE VOLTAJE

Cuando se trata de circuitos electrónicos digitales que tengan integrados de la familia TTL, se requiere de una fuente regulada de voltaje de 5 voltios. Una fuente regulada entrega en su salida un voltaje constante, independiente de las variaciones en la línea de alimentación y en la carga.

2.2.10.2 Regulación de tensión básica

La aplicación mas usada en circuitos utilizando los 78XX es el de la figura 2.24 la tensión de salida depende del circuito integrado utilizado y la corriente máxima para cualquier C.I. de esa serie es de 1 A. El capacitor C_1 , filtra la tensión del rectificador, mientras que el capacitor C_2 desacopla la alimentación.

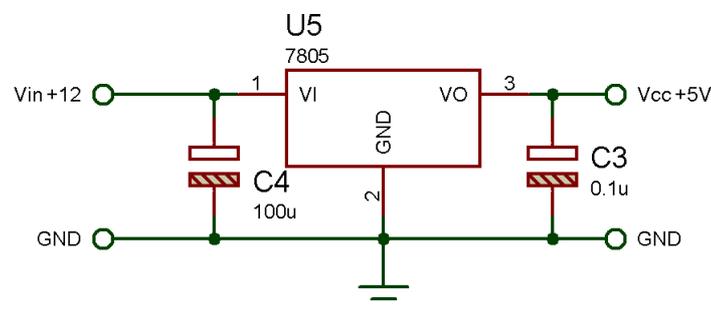


Figura 2.24: Regulación de tensión básica

2.2.10.3 COMUNICACIÓN SERIAL

El MAX232 dispone internamente de 4 conversores de niveles TTL al bus standard rs232 y viceversa, para comunicación serie como los usados en los ordenadores y que ahora están en desuso, el Com1 y Com2.

El circuito integrado lleva internamente 2 conversores de nivel de TTL a rs232 y otros 2 de rs232 a TTL con lo que en total podremos manejar 4 señales del puerto serie del PC, por lo general las más usadas son; TX, RX, RTS, CTS, estas dos

últimas son las usadas para el protocolo handshaking pero no es imprescindible su uso. Para que el max232 funcione correctamente debemos poner unos condensadores externos, todo esto lo podemos ver en la siguiente figura en la que solo se han cableado las líneas TX y RX que son las más usadas para casi cualquier aplicación.

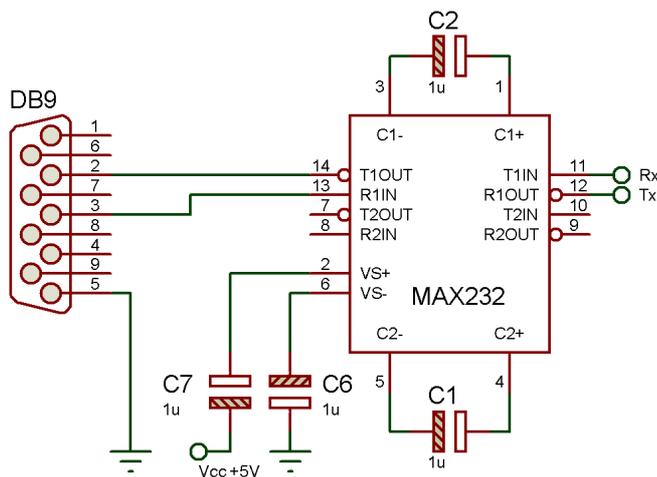


Figura 2.25: Circuito de comunicación serial

Este integrado es usado para comunicar un microcontrolador o sistema digital con un PC o sistema basado en el bus serie rs232.

2.2.10.4 CIRCUITO CONTROLADOR

Este es el *cerebro* de nuestro prototipo. Será el encargado de gestionar el encendido de cada LED mediante órdenes enviadas a las columnas mediante el flipflop 74LS373 y a las filas mediante tips 127.

Como una fila tendrá muchos LEDs y existe la posibilidad que en algún momento puedan estar todos encendidos, no podemos conectarlas directamente a pines de E/S del AVR, porque la corriente que demandarían haría que el puerto del microcontrolador se dañe. Para evitar esto, utilizaremos en medio un transistor capaz de manejar la corriente requerida.

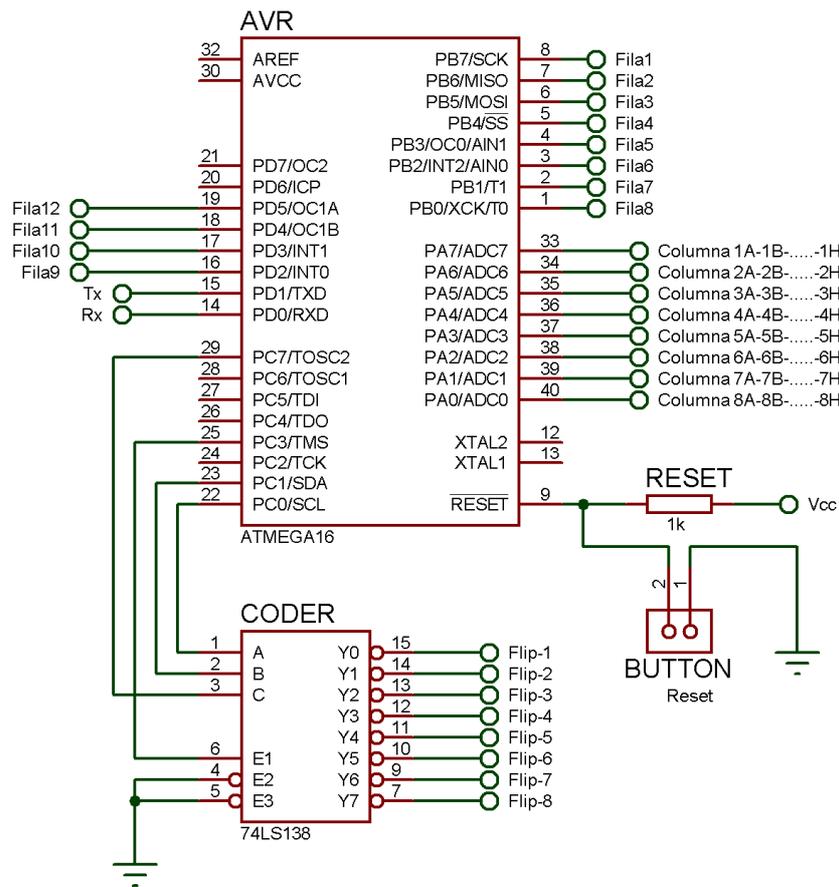


Figura 2.26: circuito controlador.

Analicemos el circuito de la figura anterior. El centro de todo es el microcontrolador ATMEGA 16, que tiene su pin de *RESET* conectado a un pulsador y un resistor de 1K. Este pulsador permite reiniciar el sistema cuando lo necesitemos. Esta siendo usado un circuito de reloj interno, basado en un cristal de 1 GHZ. Esto le permite al AVR ejecutar un millón de instrucciones por segundo, más que suficientes para este proyecto.

2.2.10.5 Decodificadores

Son circuitos que tienen por entrada una cantidad en binario y tienen tantas salidas como posibles combinaciones haya con esas entradas.

Decodificador de 3 a 8:

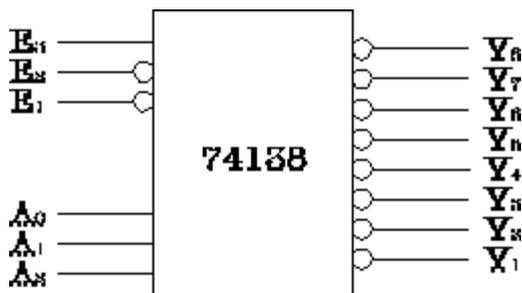


Figura 2.27: Chip 74138 decodificador de 3 a 8.

GL G1 G2	Selección	Salidas
	C B A	Y0 Y1 Y2 Y3 Y4 Y5 Y6 Y7
X X 1	X X X	1 1 1 1 1 1 1 1
X 0 X	X X X	1 1 1 1 1 1 1 1
0 1 0	0 0 0	0 1 1 1 1 1 1 1
0 1 0	0 0 1	1 0 1 1 1 1 1 1
0 1 0	0 1 0	1 1 0 1 1 1 1 1
0 1 0	0 1 1	1 1 1 0 1 1 1 1
0 1 0	1 0 0	1 1 1 1 0 1 1 1
0 1 0	1 0 1	1 1 1 1 1 0 1 1
0 1 0	1 1 0	1 1 1 1 1 1 0 1
0 1 0	1 1 1	1 1 1 1 1 1 1 0
1 1 0	X X X	Todos a nivel alto

Figura 2.28: tabla de verdad del 74138

A, B, y C..... Entradas de selección

Y0 a Y7..... Salidas

G2A' Entrada habilitadora

G2B' “ “

G1..... ” “

Vcc y GND..... Alimentación

Cuando se selecciona una salida se le envía un 0, por lo tanto, si se quiere tener un 1 en la salida se habrá que poner una puerta NOT. En nuestro caso el flipflop toma como información de encendido el 0L por lo cual no es necesario una compuerta inversora en el pin OE del 74LS373

2.2.10.6 Multiplexado

Cuando se trabaja con matrices, generalmente se utiliza la multiplexación como técnica de optimización de potencia y reducción de la circuitería, ya que al multiplexar la información, los LEDs reciben pulsos de corriente durante pequeños intervalos de tiempo. El método de controlador dinámico o llamado también refresco por multiplexación en el tiempo permita habilitar secuencialmente cada una de las filas y/o columnas de una matriz y actualizar sus datos continuamente, aún cuando éstos no cambie; es decir, se realiza en una matriz a un proceso de barrido que puede ser por filas, por columnas o por filas y columnas. El ojo humano detecta como parpadeo frecuencias inferiores a 50 Hz (periodo 20 milisegundos) y como señales continuas, frecuencias por encima de ese valor. La frecuencia de refresco o de barrido es la frecuencia a la que se selecciona un LED.

En una matriz de $m \times n$ píxeles donde m es el número de columnas y n el número de filas, la frecuencia de barrido es aquella a la cual un determinado LED vuelve a ser seleccionado luego de haber realizado el mismo proceso en el resto de elementos de toda la matriz el valor considerado para el diseño del prototipo es 85 Hz.

El período de tiempo durante el cual un LED se habilita para desplegar un dato, se le conoce como ciclo de trabajo, el cual se denota con la letra d y se obtiene de la relación entre el tiempo T necesario para actualizar toda la matriz y el tiempo t_e en que un LED esta encendido como se aprecia en la figura 2.29.

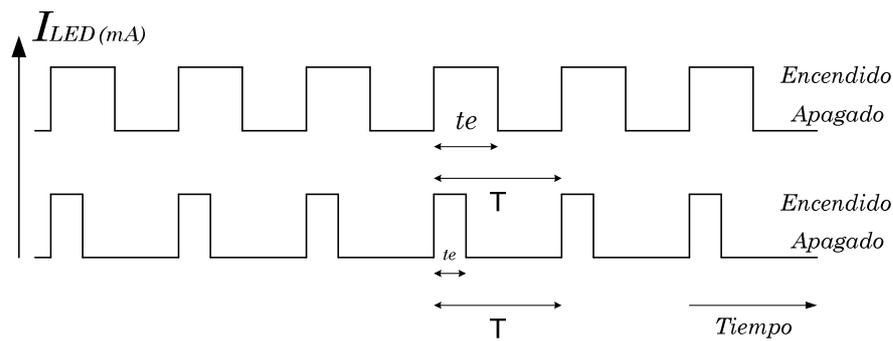


Figura 2.29: El pulso superior producirá un mayor brillo que el inferior

Conforme disminuye el ciclo de trabajo, disminuye la luminosidad y viceversa debido a que la corriente efectiva circula a través de un LED determinando el nivel de brillo del mismo. De acuerdo a los criterios de diseño previos, para la matriz se necesitarán 24 líneas de control para filas y 32 para líneas para control de columnas; por lo tanto, es más apropiado aplicar barrido por filas que por columnas debido al menor número de líneas en las filas.

Para administrar de mejor manera la matriz del prototipo, es necesario dividirla en matrices más pequeñas o submatrices; en donde se podría tener las siguientes opciones:

2 submatrices de 24 filas por 16 columnas de LEDs.

2 submatrices de 12 filas por 32 columnas de LEDs.

4 submatrices de 12 filas por 16 columnas de LEDs.

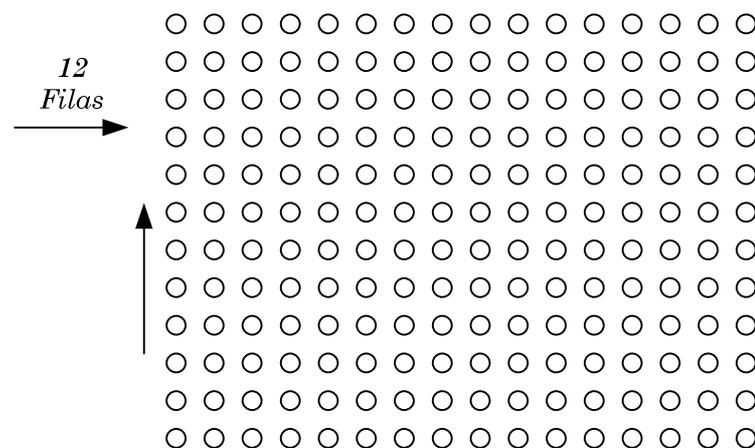


Figura 2.30 submatriz de 12 filas por 16 filas columnas

La primera opción tiene mayor número de filas por lo que se descarta su aplicación al igual que el segundo caso, debido a similar situación pero en columnas, en comparación con el tercer caso, el cual optimizará la circuitería y flexibilizará la programación del microprocesador al manejar 16 líneas en lugar de 32. De acuerdo con la forma física que debe tener la matriz, ésta se dividirá en cuadrantes de manera que a cada uno de ellos corresponda una submatriz de 12 filas por 16 columnas, como se indica en la figura 2.30.

2.2.10.6.1 Barrido de columnas

Activamos cada columna de la matriz realizando un barrido por cada una de ellas, cada vez que se activa la salida del coder para el flip-flop correspondiente. Amplificamos este pulso de corriente con los ULN 2803 que entregan a la columna una corriente amplificada para dar mayor luminosidad al diodo LED que se desea encender.

Este barrido se lo realiza con ayuda del coder y los LATCH que guardan por instantes mínimos un dato para cada columna como se muestra en el siguiente grafico que esta diseñado para el barrido de las primeras 32 columnas, para nuestro caso utilizamos dos de estos circuitos.

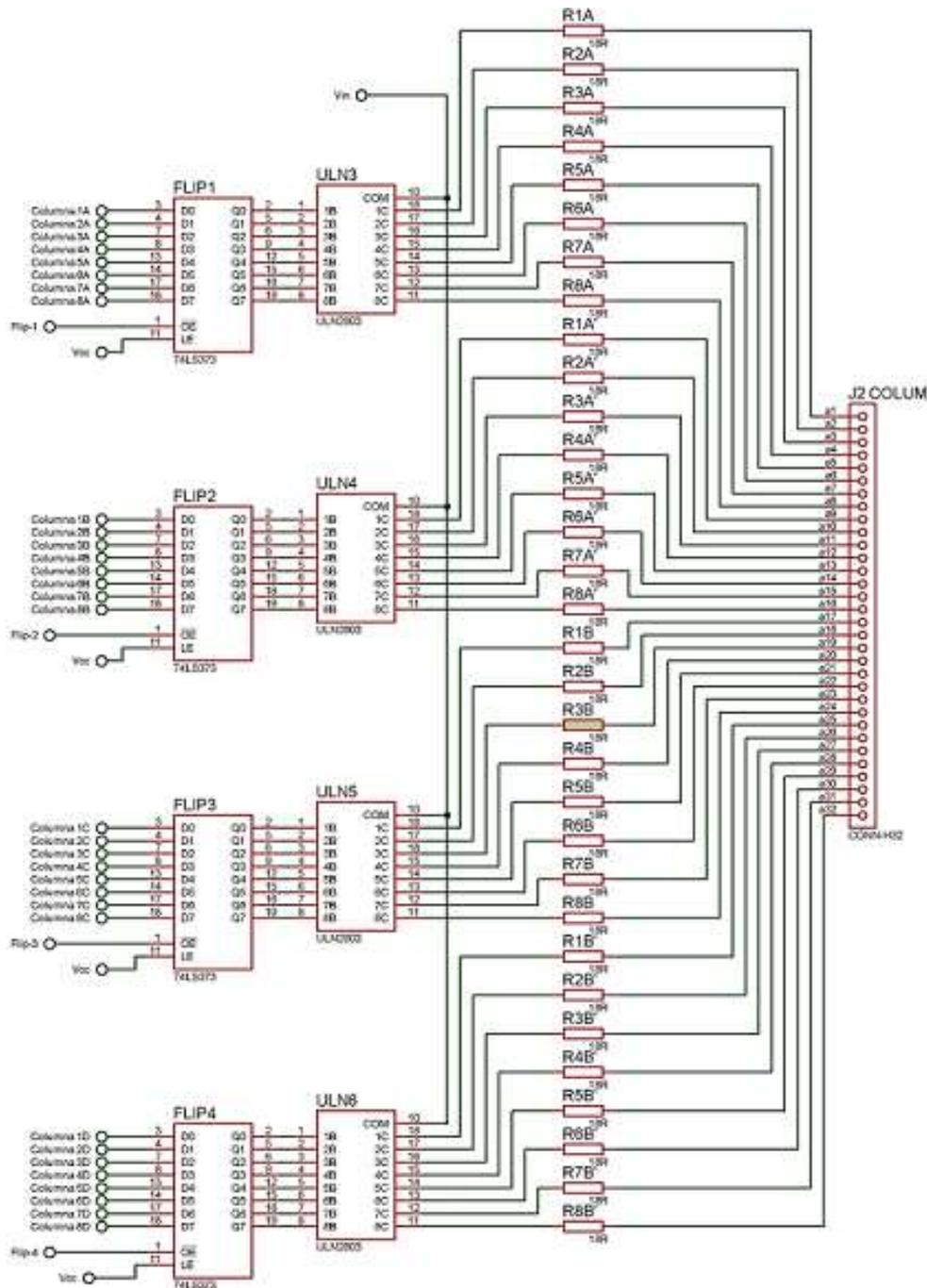


Figura 2.31: Barrido de las columnas

2.2.10.7 Circuito integrado ULN2803

Este circuito recoge las señales procedentes del puerto paralelo y aporta la potencia necesaria para poder hacer funcionar la matriz de LEDs. Tenemos ocho circuitos ULN2803 todos tienen la función de transmisión de datos. Internamente los circuitos están formados por transistores conectados de forma que a la salida

tenemos una etapa amplificadora. Las salidas de este circuito están unidas a un banco de resistencias para proteger a los LEDs.

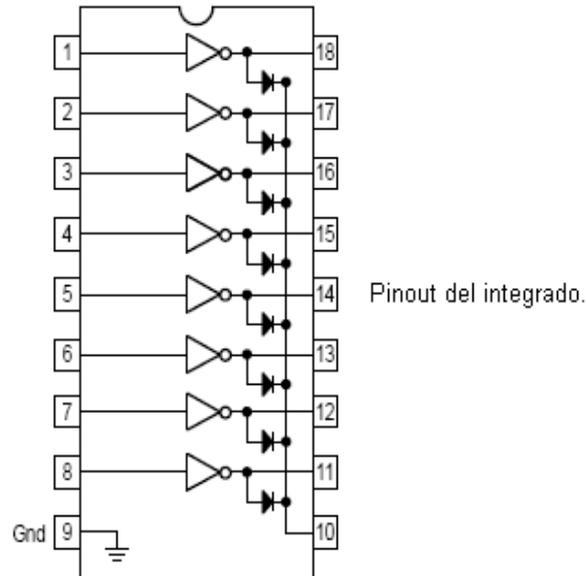


Figura 2.32: ULN2803

2.2.10.8 Amplificación de Ánodos

Se realiza la amplificación con tips 127 realizando barrido de ánodos de las cuatro submatrices que conforman la matriz de LEDs. Se realiza la amplificación de corriente con la ayuda de tips PNP que dan un pulso positivo de amplificación para activar los ánodos de cada fila de las submatrices.

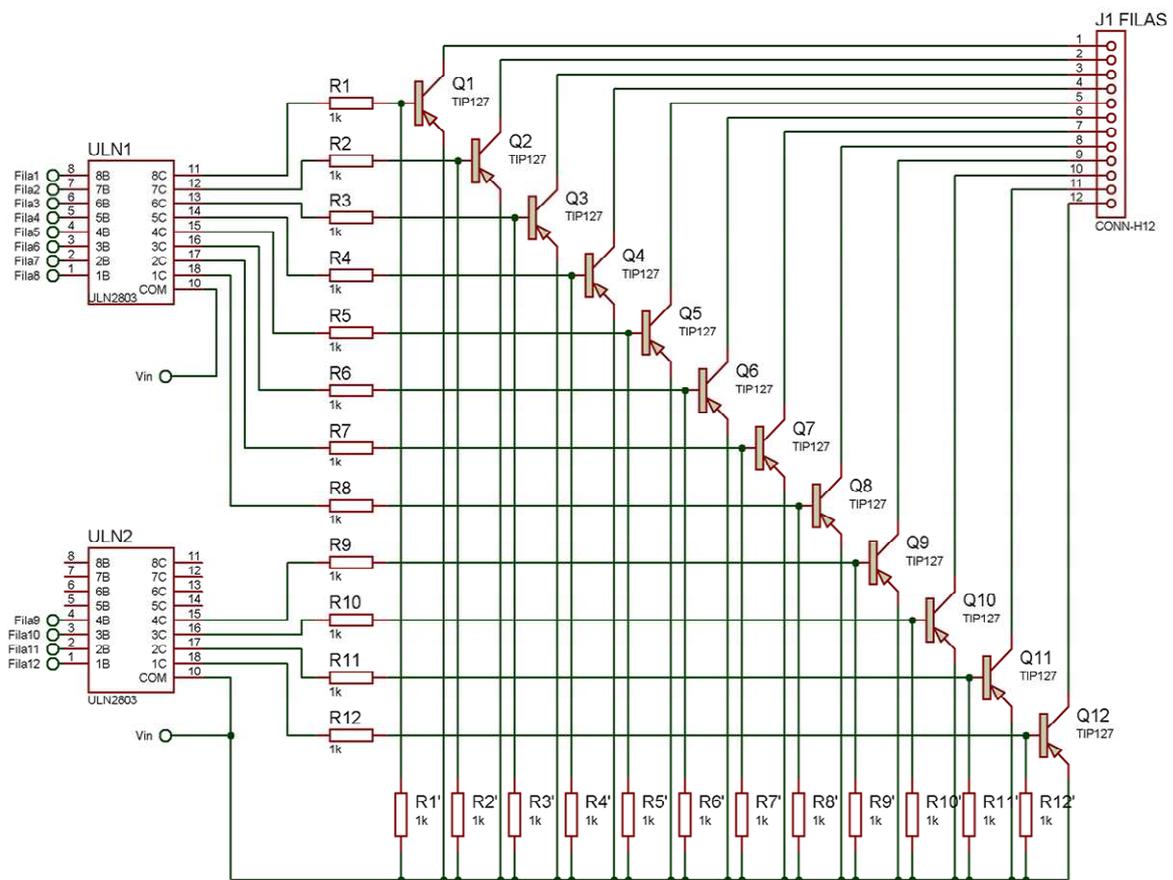


Figura 2.33: Amplificación con tips 127

2.2.11 MATRIZ DE DIODOS

2.2.11.1 Configuración de la Matriz

La matriz a diseñar, tiene una configuración de sus filas en ánodo común y columnas en cátodo común.

Se divide la matriz completa en cuatro submatrices para obtener menor consumo de potencias realizando múltiplexación tomando en cuenta el valor de la corriente amplificada por los ULN 2803 que en barrido de columnas entrega una corriente de 700mA para encender los 12 diodos que se encuentran en cada columna, esto dependería de la imagen a realizarse en el prototipo ya que algunas de ellas tomaría de cada columna un número menor de diodos encendidos.

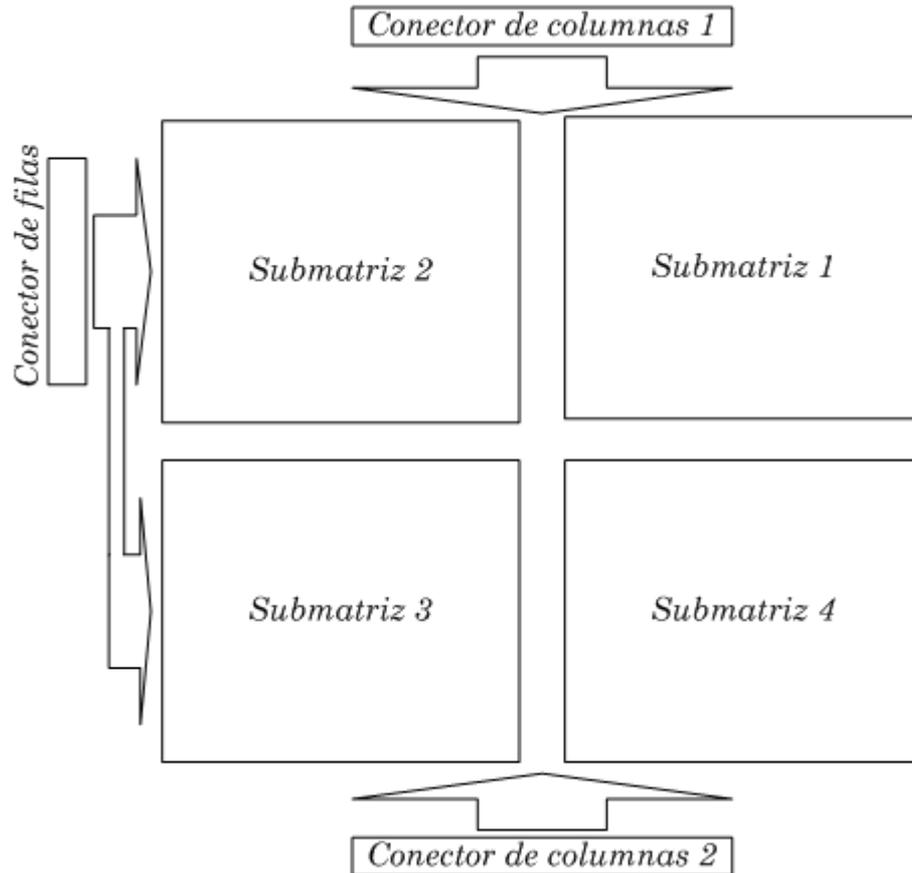


Figura 2.34: Configuración de la matriz de LEDs

2.2.11.2 Algoritmo

Al arrancar el programa, se inicializa primeramente el puerto serial del microcontrolador, de manera que en cualquier instante se pueda establecer la comunicación con el computador personal a una velocidad de 1200 bps mediante una interrupción serial, a continuación, el programa entra a operar en modo predeterminado, en el cual se tiene una secuencia de figuras letras y números. Cuando una interrupción serial ocurre, se detiene el proceso anterior que está ejecutando y mediante el software para configuración desde el PC, se envían al microcontrolador los valores del tiempo de duración para cada señal y también del modo de operación, estos valores reemplazarán a los valores por defecto, y una vez que cesa la interrupción, el programa volverá a ejecutarse con la nueva configuración, con la que operará hasta que no se vuelva a configurar el prototipo.

En el proceso de multiplexación, el tiempo durante el cual un LED de una columna deben mostrar un dato, se controla mediante una base de tiempo la cual determina la duración del almacenamiento de los datos en los latches y por consiguiente, el ritmo de la multiplexación.

2.2.12 DISEÑO DEL PROTOTIPO DE MATRIZ TACTIL

2.2.12.1 DISEÑO DEL HARDWARE

2.2.12.1.1 MULTIPLEXACION POR LATCH

Al utilizar cientos de LEDs necesitaríamos tener un microcontrolador que tenga como mínimo ese número de pines de salida (y por supuesto, no existe).

El secreto está en el multiplexado. Esta técnica permite utilizar unos pocos pines de E/S del microcontrolador para manejar una serie de circuitos integrados que se encarguen de excitar los LEDs. Hay varias maneras, y muchos modelos diferentes de circuitos para hacer esto.

Pueden usarse un tipo de integrado digital llamado *"LATCH"*, que básicamente es una memoria en la que escribimos un valor, y lo mantiene en sus salidas hasta que nosotros lo indiquemos. De esta manera, usando varios latches podríamos encender los LEDs por turnos, rápidamente para que no se note el parpadeo, y de esa manera formar una palabra o grafico. Otra forma es utilizar un registro de desplazamiento el mismo que funciona de la misma manera en que funciona una cola de gente que espera para entrar en un cine. En un registro de desplazamiento, en lugar de personas tenemos "0" y "1". Lo bueno de esto es que para "meter" datos ("0"s y "1"s) en el registro de desplazamiento solo hacen falta tres pines del microcontrolador, independientemente de lo largo que sea.

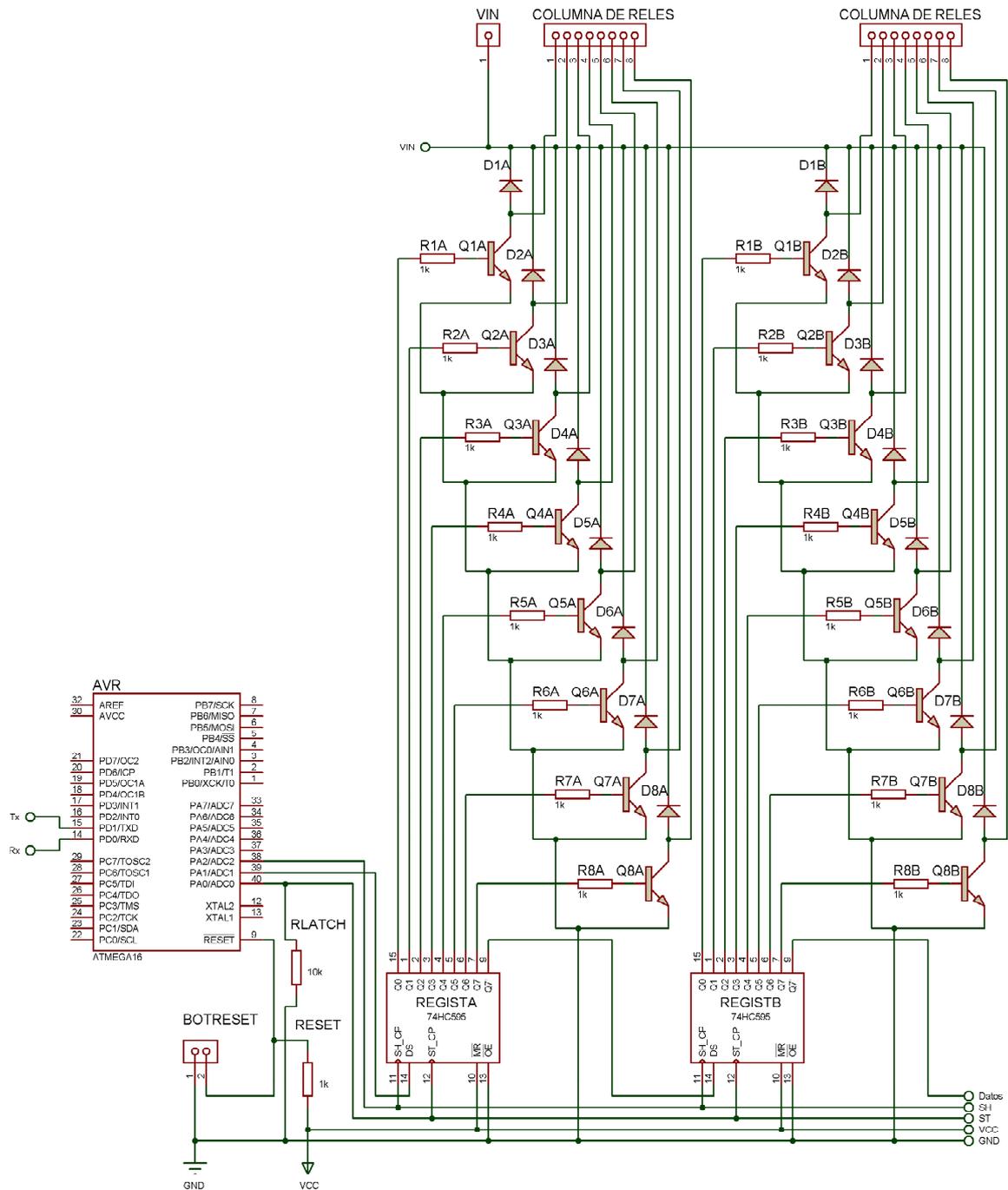


Figura 2.35: Multiplexación por LATCH

El diseño del prototipo de tacto es realizado con LATCH que utiliza únicamente 3 salidas del AVR y pueden ser varios de estos para aumentar el tamaño, en nuestro caso utilizamos el 74HC595 que transforma los datos seriales a una forma paralela y guardar esos datos hasta un determinado tiempo el cual es designado.

En las salidas de los LATCH utilizamos amplificación de corriente para activar electroimanes que necesitan aproximadamente 2A para trabajar en la condición que deseamos. Para esto utilizamos amplificadores de corriente como el Tip122 que junto a un diodo permite proteger a la bobina del electroimán de corrientes que podrían dañar el bobinado.

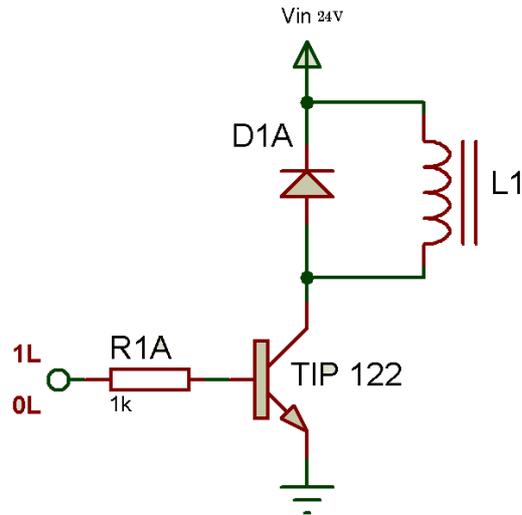


Figura 2.36: Amplificación para el electroimán

Activamos la bobina cuando el LATCH envía en su salida un 1L de lo contrario el Tip no realiza la amplificación.

2.2.12.1.2 DESCRIPCION DE LOS ELECTROIMANES

Un electroimán consiste en un núcleo de hierro rodeado de una bobina, que se imanta a voluntad cuando hacemos pasar una corriente eléctrica, y se desimanta en el momento que interrumpimos esta corriente (Figura 2.37).

El tamaño del núcleo de los electroimanes que utilizamos en el prototipo llega a la mitad de la longitud de las bobinas, se utiliza este criterio para utilizar la otra mitad que queda de espacio en la bobina como núcleo móvil para obtener de esta manera el mecanismo que necesitamos de halar un objeto.

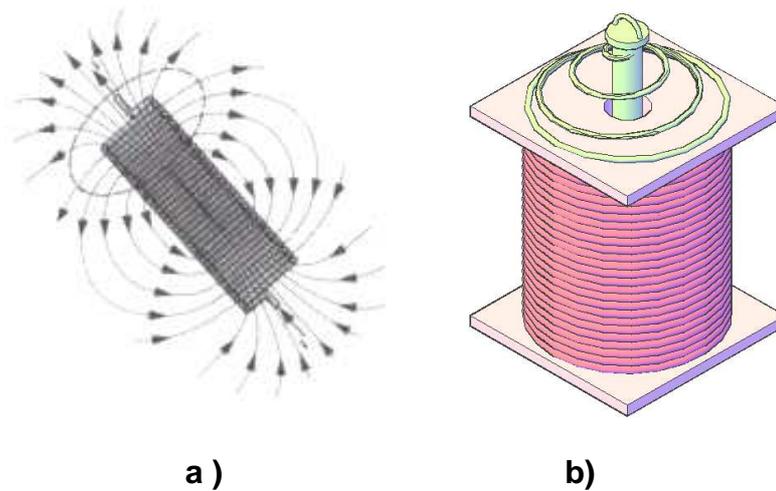


Figura 2.37: a) Campo magnético de un electroimán b) Electroimán a Diseñar

La fuerza con la que atrae un electroimán a una pieza de hierro móvil (armadura) a través del aire o entrehierro se puede calcular por medio de diferentes métodos.

Esta es una aproximación práctica al diseño de electroimanes, se reduce a un mínimo la teoría, asumiendo conocimientos básicos y teóricos en circuitos magnéticos, por esta razón no se tiene en cuenta el flujo disperso en el hierro y en las cercanías del entrehierro.

CONSIDERACIONES TEORICAS

Bobina de N espiras, por la que circula una corriente de I amperes. Fig 2.37a. Circuito magnético típico, con núcleo de hierro.

2.2.12.1.2.1 FUERZA MAGNETOMOTRIZ

En la Eq 2.3, el flujo magnético es creado por N espiras de alambre, por las que circula una corriente de I amperes, produciendo entonces una fuerza magnetomotriz dada por:

$$F = \frac{(N \times I)}{0.796}$$

Eq 2.3 Fuerza Magnetomotriz

Donde:

F: fuerza magnetomotriz en ampere.espira

N: numero de espiras

I: corriente en amperes

2.2.12.1.2.2 FUERZA MAGNETIZADORA

La fuerza magnetizadora se define como la fuerza magnetomotriz por unidad de longitud del camino.

$$H = \frac{F}{L}$$

Eq 2.4 Fuerza Magnetizadora

Donde:

H: fuerza magnetizadora en LENZ

F: fuerza magnetomotriz en amper.espira

L: longitud del camino cm.

2.2.12.1.2.3 PERMEABILIDAD

La permeabilidad (Mu) esta definida por la relación:

$$Mu = \frac{B}{H}$$

Eq 2.5 Permeabilidad

Donde:

Mu: Permeabilidad

B: Densidad de flujo, en GAUSS

H: Fuerza magnetizadora en Lenz

En el aire, H es numéricamente igual a la densidad de flujo B.

La permeabilidad es la equivalente de la conductividad en los circuitos eléctricos.

La permeabilidad en el hierro no es constante, ya que varía con el flujo.

La relación entre B, H y Mu se muestra en las curvas de B-H del hierro

CONSIDERACIONES GENERALES

Para un tratamiento simple en forma teórica, del circuito de la fig. 2.37b, se deben hacer una serie de presunciones:

- Se asume que en el circuito magnético, toda la fuerza magnetomotriz es generada para vencer la "resistencia" del entrehierro.
- Las líneas de flujo están confinadas enteramente dentro del hierro en toda la extensión del camino del hierro.
- Que las líneas de flujo están distribuidas uniformemente en la sección cruzada del hierro.
- Que la sección de hierro es constante a lo largo de todo el circuito magnético.
- Que las dimensiones del entrehierro son pequeñas comparadas con la sección de los polos.
- Que la sección del entrehierro es la misma que la del hierro.

DISEÑO PRÁCTICO por el método de FUERZA-BRUTA.

Problema:

Encontrar las NI (amper.espira) necesarias para generar una determinada densidad de flujo (B) en un determinado entrehierro (L).

Solución:

En el circuito magnético de la FIG 2.37b se puede asumir que en la región del entrehierro, la inducción magnética es uniforme, luego tenemos:

$$H = B$$

Luego, asumimos también que toda la fuerza magnetomotriz generada es Necesaria para vencer la resistencia del entrehierro:

$$H = \frac{F}{L}$$

$$H = B$$

Luego:
$$B = \frac{F}{L}$$

Finalmente:
$$F = B \times L \quad (1)$$

Recordando:
$$F = \frac{(N \times I)}{0.796} \quad (2)$$

Luego hacemos $(1) = (2)$

$$B \times L = \frac{(N \times I)}{0.796}$$

Finalmente $N \times I = 0.796 \times B \times L$ (3)

Par nuestro diseño tenemos los siguientes datos que dimensionados con la experimentación y el tipo de cable para bobinado conseguido tenemos:

$$L = 2\text{cm}$$

$$N = 4000 \text{ espiras}$$

$$I = 100\text{mA}$$

$$F = \frac{(N \times I)}{0.796} = B \times L$$

$$F = \frac{(4000 \times 100\text{mA})}{0.796} = B \times 2\text{cm}$$

$$F = 502.51256 \text{ ampere.espira} = B * 2\text{cm}$$

$$B = 502.51256 \text{ ampere.espira} / 2 \text{ cm}$$

$$B = 251.26528 \text{ ampere.espira} / \text{cm}$$

Por lo tanto tenemos:

Fuerza magnetomotriz de 502.51256 ampere.espira

Densidad de Flujo de 251.26528 ampere.espira / cm

NOTA: Este método denominado de FUERZA-BRUTA no involucra las dimensiones del circuito magnético, con la excepción de la longitud del entrehierro.

Sin embargo, como una primera aproximación da buenos resultados dentro del +/- 10 %.

El Cable usado para el prototipo es del numero 27, escogimos este numero por la corriente mínima que deseamos utilizar y el numero bastante grande de vueltas que hay que enrollar para obtener los resultados previstos.

AWG	Diam. mm	Area mm2	AWG	Diam. mm	Area mm2
1	7.35	42.40	16	1.29	1.31
2	6.54	33.60	17	1.15	1.04
3	5.86	27.00	18	1.024	0.823
4	5.19	21.20	19	0.912	0.653
5	4.62	16.80	20	0.812	0.519
6	4.11	13.30	21	0.723	0.412
7	3.67	10.60	22	0.644	0.325
8	3.26	8.35	23	0.573	0.259
9	2.91	6.62	24	0.511	0.205
10	2.59	5.27	25	0.455	0.163
11	2.30	4.15	26	0.405	0.128
12	2.05	3.31	27	0.361	0.102
13	1.83	2.63	28	0.321	0.0804
14	1.63	2.08	29	0.286	0.0646
15	1.45	1.65	30	0.255	0.0503

La matriz de electroimanes se desarrollo de una forma experimental, buscando una fuerza en los electroimanes que atraigan un núcleo móvil con la suficiente fuerza que al tocarlos estos no se separen fácilmente y así lograr que fuesen fijos al tratarlos en la práctica.

2.2.12.2 DISEÑO DEL SOFTWARE

La programación del microcontrolador AVR se la realizo con ayuda de un compilador de BASIC para la familia AVR de ATMEL, BASCOM que con la ayuda de sus instrucciones realizamos el siguiente procedimiento.

Cuando el usuario hace un clic en la opción del programa diseñado para poder tocar las imágenes se realiza el siguiente flujo de datos.

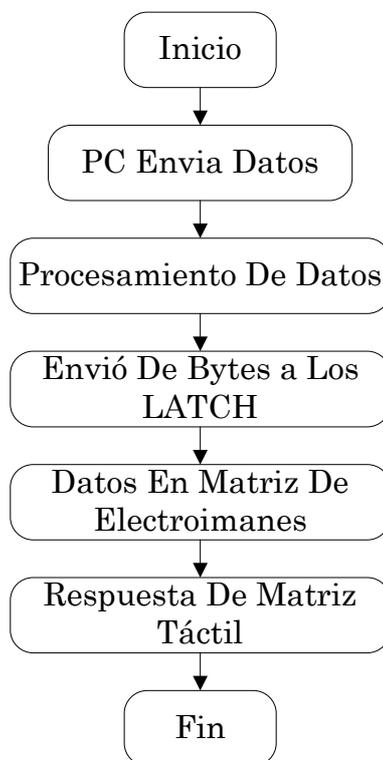


Figura 2.38: Comunicación de Datos

El código fuente de la programación que realizamos se detalla a continuación.

```

' *** Tipo de ATMEGA a utilizar
$regfile = "m16def.dat"

' *** Cristal y velocidad de comunicación
$crystal = 1000000
$baud = 4800

' *** configurando puerto de salida
  
```

Config Porta = Output

' *** Nombrando pines de salida

Datos Alias Porta.1

Reloj Alias Porta.2

Latches Alias Porta.0

' *** variables

Dim L(8) As Byte

Dim B(8) As Byte

Dim Dat1 As String * 8

Dim Dat2 As String * 8

Dim Dat3 As String * 8

Dim Dat4 As String * 8

Dim Dat5 As String * 8

Dim Dat6 As String * 8

Dim Dat7 As String * 8

Dim Dat8 As String * 8

Dim A As Byte

Dim C As Byte

' *** Inicio del programa

Inicio:

Waitms 100

' *** envia al PC listo para iniciar

Print "iniciar"

' *** todos los LATCH en cero

B(1) = &B00000000

B(2) = &B00000000

B(3) = &B00000000

B(4) = &B00000000

B(5) = &B00000000

B(6) = &B00000000

B(7) = &B00000000

B(8) = &B00000000

For C = 1 To 8

' *** Se inicia comunicacion serial a los LATCH

```
Shiftout Datos , Reloj , B(c) , Option [ , 0 , 9 ]
Pulseout Latches, 0, 1
```

```
Next C
```

```
Wait 1
```

```
' ***** Ingreso de datos del PC *****
```

```
Inputbin Dat1, Dat2, Dat3, Dat4, Dat5, Dat6, Dat7, Dat8
```

```
L(1) = Binval(Dat1)
```

```
L(2) = Binval(Dat2)
```

```
L(3) = Binval(Dat3)
```

```
L(4) = Binval(Dat4)
```

```
L(5) = Binval(Dat5)
```

```
L(6) = Binval(Dat6)
```

```
L(7) = Binval(Dat7)
```

```
L(8) = Binval(Dat8)
```

```
For A = 1 To 8
```

```
' *** Paso de datos a los LATCH
```

```
Shiftout Datos , Reloj , L(a) , Option [ , 0 , 9 ]
```

```
Pulseout Latches, 0, 1
```

```
Waitms 10
```

```
Next A
```

```
' *** se envia al PC fin
```

```
Print "fin"
```

```
End
```

Esta codificación es compilada con la ayuda del programa Bascom y luego pasados los datos de este al Atmega16 con la ayuda del programador de AVR progisp en el que configuramos la utilización de cristal interno que posee el chip.

2.3 IMPLEMENTACION

2.3.1 CONSTRUCCIÓN DE PROTOTIPOS

Después de realizar varias investigaciones basadas en la experimentación continúa de cada uno de los elementos que conforman el prototipo de lectura de texto con sintetizador de voz, matriz de diodos y matriz táctil se elaboro de forma manual el ensamblaje y construcción de los componentes de estos prototipos por los mismos autores del proyecto.

2.3.2 Matriz de LEDs

En este capitulo mostraremos la elaboración de placas y chasis que conforma en su totalidad la construcción de la matriz.

El prototipo de gráficos en diodos, se elaboro en base a la experiencia en programación de microcontroladores AVR, en el conocimiento de sistemas digitales y en la elaboración de placas para montar los circuitos diseñados, en el anterior capitulo se resuelve en su totalidad el diseño de este prototipo.

2.3.2.1 Implementación de Placas

El prototipo consta de dos placas, una para conformar la matriz de LEDs y otra para implementar su controlador, las dos se unen por tres espadines que conectan el barrido de filas comunes y los otros dos conectores que alimentan el barrido de columnas hacia los cátodos de la matriz

2.3.2.2 Placa de matriz de LEDs

Esta placa esta diseñada para mantener en una sola de ellas las cuatro submatrices que se dividieron para realizar multiplexación, además se la diseño para no utilizar cables o bus de datos, sino para que por medio de espadines se conecte a la placa que controla todo su funcionamiento.

Se utilizó varias herramientas para su diseño y construcción como es: el programa *Proteus* para diseñar las pistas que conectan cada elemento, realización de transferencia de imagen de papel fotográfico a la placa de baquelita, realización de agujeros en la placa con sus respectivos taladros y luego de realizar estos procedimientos se procedió a la corrosión del cobre para dejar la placa con sus respectivas pistas a utilizar.

Para que la placa de cobre no se oxide rápidamente se añadió una delgada capa de laca transparente o como en los casos profesionales se usa una capa verde de laca. Para evitar que al soldar los elementos, se dañe las pistas con el calor porque el estaño no se adhiere a la placa.

La matriz de LEDs está diseñada con placas de doble cara de cobre, esto para evitar la acumulación de puentes entre pistas y obtener un mejor trabajo de implementación.

En la siguiente figura se muestra como está el diseño de pistas en el lado posterior de la placa, como también la unión de todos los ánodos de los diodos para su respectivo barrido realizado por el controlador de este circuito.

El gráfico se muestra como se miran las pistas desde una perspectiva frontal de la matriz, ya que si se mira en la placa este estaría realizado un *MIRROR* para su construcción.

Por facilidad de explicación del prototipo, se tomó las imágenes a un tamaño de menor escala, los tamaños originales se presentan en los anexos del proyecto.

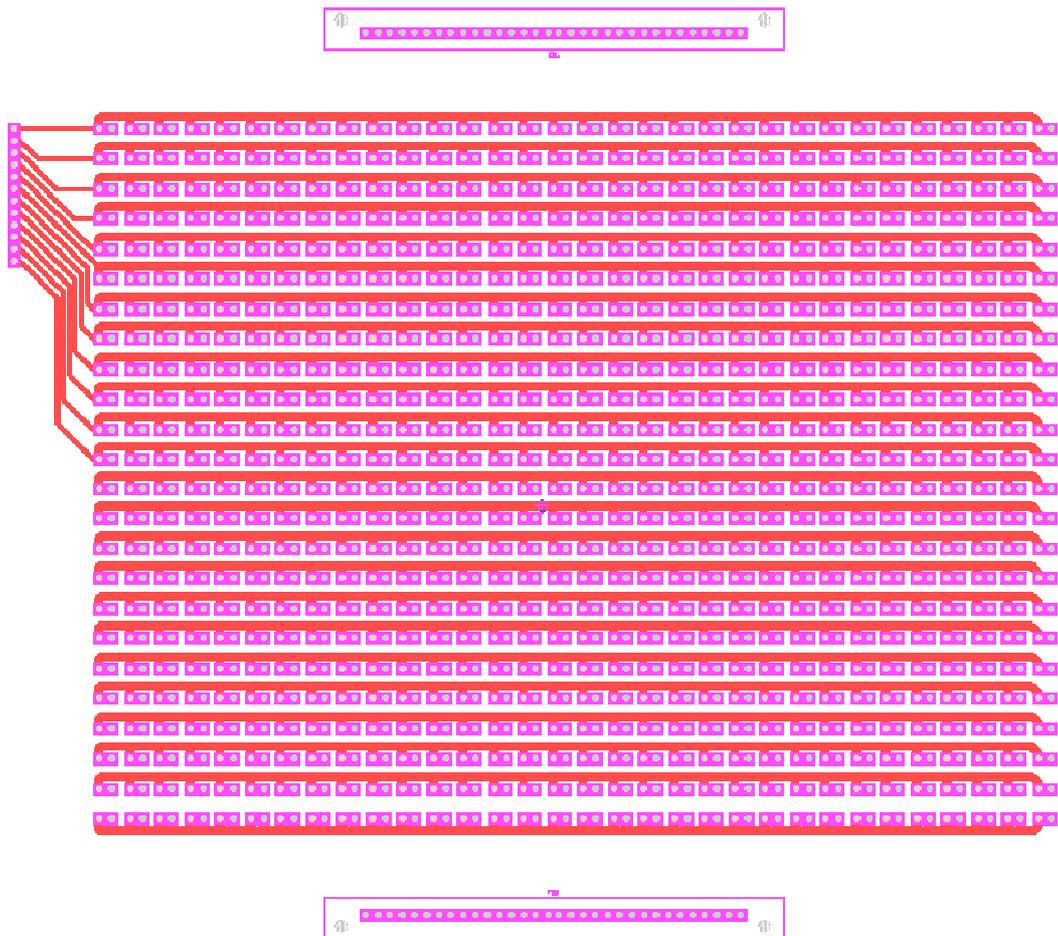


Figura 2.39: Placa de LEDs posterior

En la figura 2.40 se muestra el diseño de la parte frontal de la matriz, es decir el sitio donde se colocan los elementos que constituyen el prototipo. Además realizan el barrido de los ánodos con su espadín de 12 filas y el barrido de columnas con sus 2 espadines de 32 pines para el barrido de cátodos.

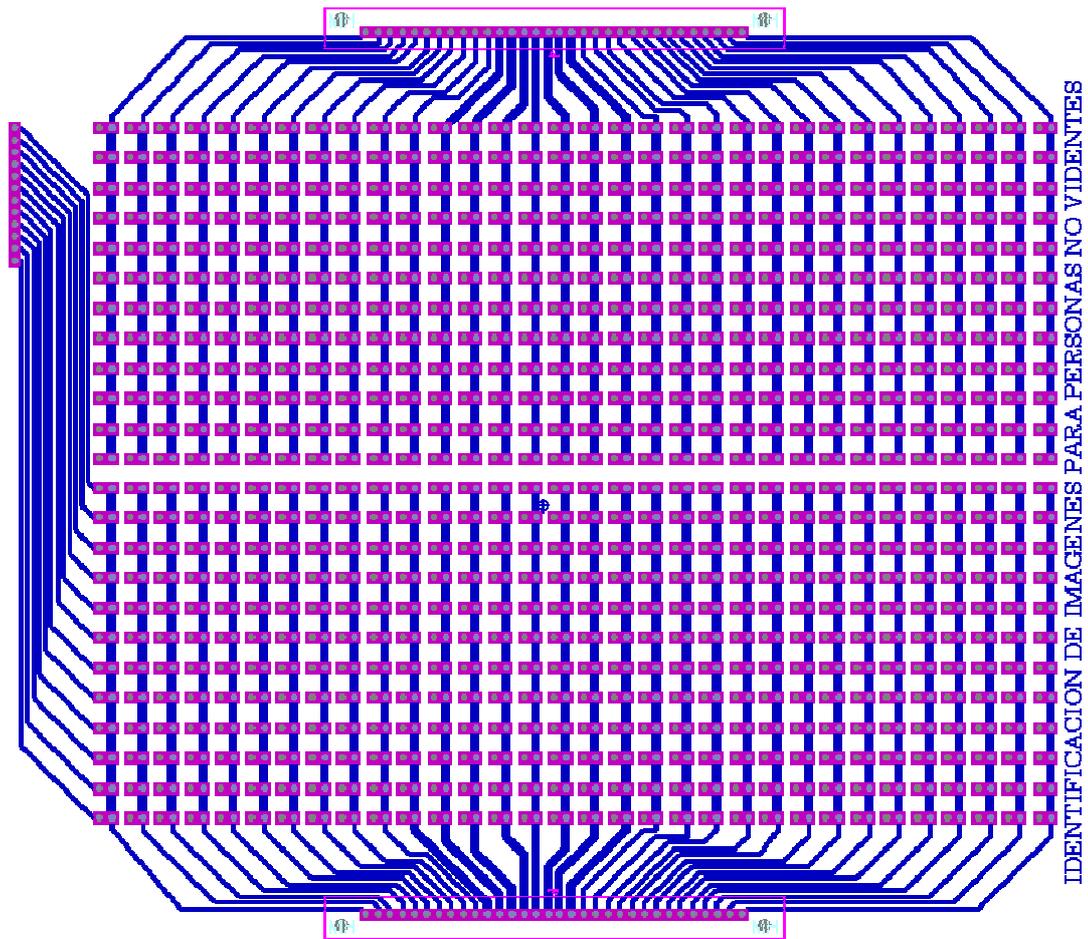


Figura 2.40: Frente de placa de LEDs

Para tener una forma más apreciable de la distribución de los elementos, se muestra el siguiente gráfico donde se aprecia el posicionamiento de cada uno de ellos.

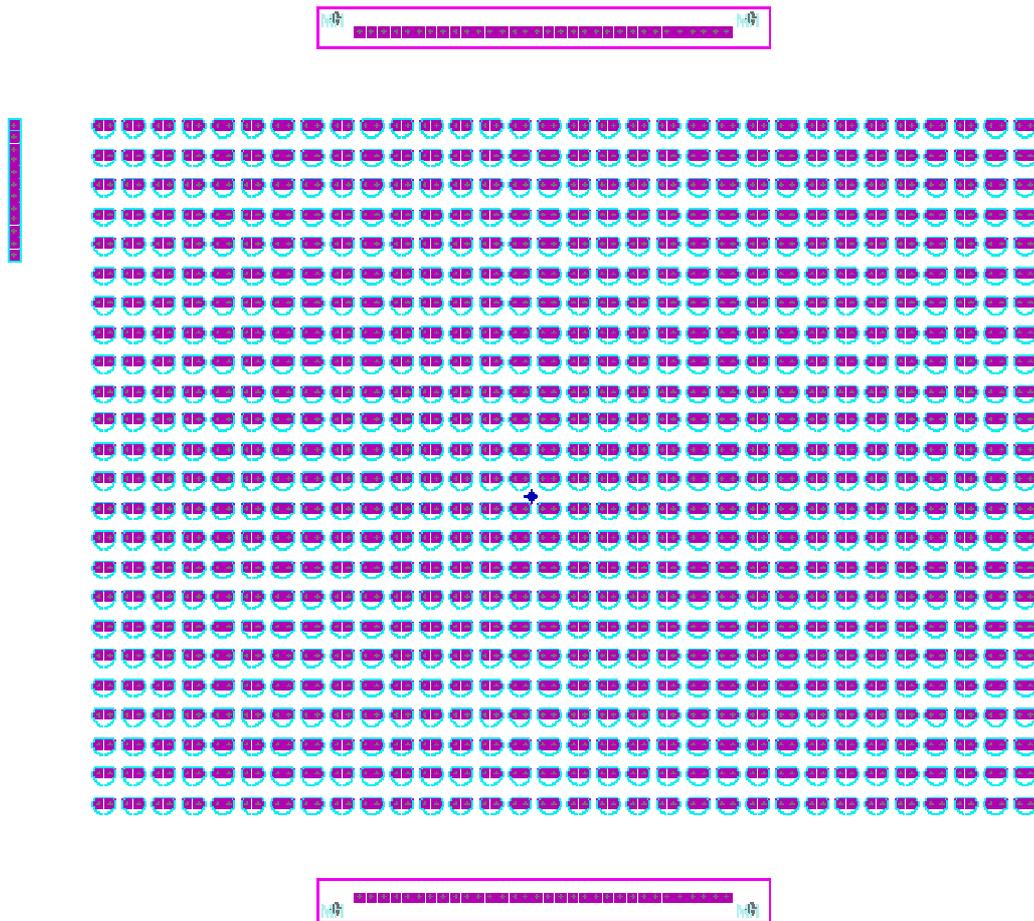


Figura 2.41: Distribución de elementos

El diseño en su totalidad con las dos caras de la placa y la distribución de los elementos se muestra a continuación, de esta manera quedaría el diseño completo de la placa de matriz de LEDs

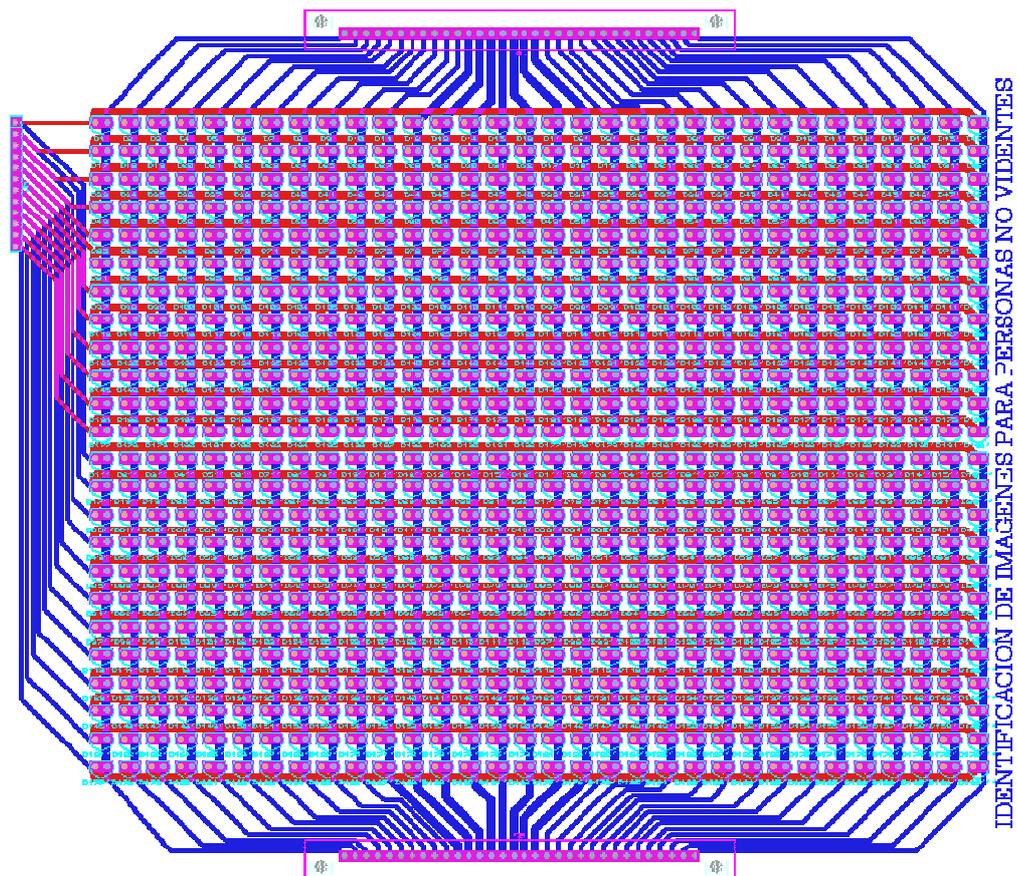


Figura 2.42: Distribución completa

Después de haber realizado el diseño completo, se procedió a la construcción misma de la matriz de LEDs, al igual que el diseño la construcción de la matriz se la realizo de forma manual y sin ninguna ayuda profesional, aplicando conocimientos de elaboraron de circuitos impresos.



Figura 2.43: Matriz de LEDs

2.3.3 Placa de Controlador de matriz de LEDs

Su diseño e implementación fue más complejo que el circuito anterior, ya que tenemos más elementos que implementar, para el diseño tenemos que tomar las mismas distancias de los conectores de espadines que la placa anterior, para que se pueda ensamblar una placa sobre otra, es decir la placa donde se encuentra el banco de LEDs sobre la placa que forma el control, además se tomo en cuenta el tamaño de los elementos para que la anterior placa no interfiera sobre esta y por ende haya inconveniente alguno.

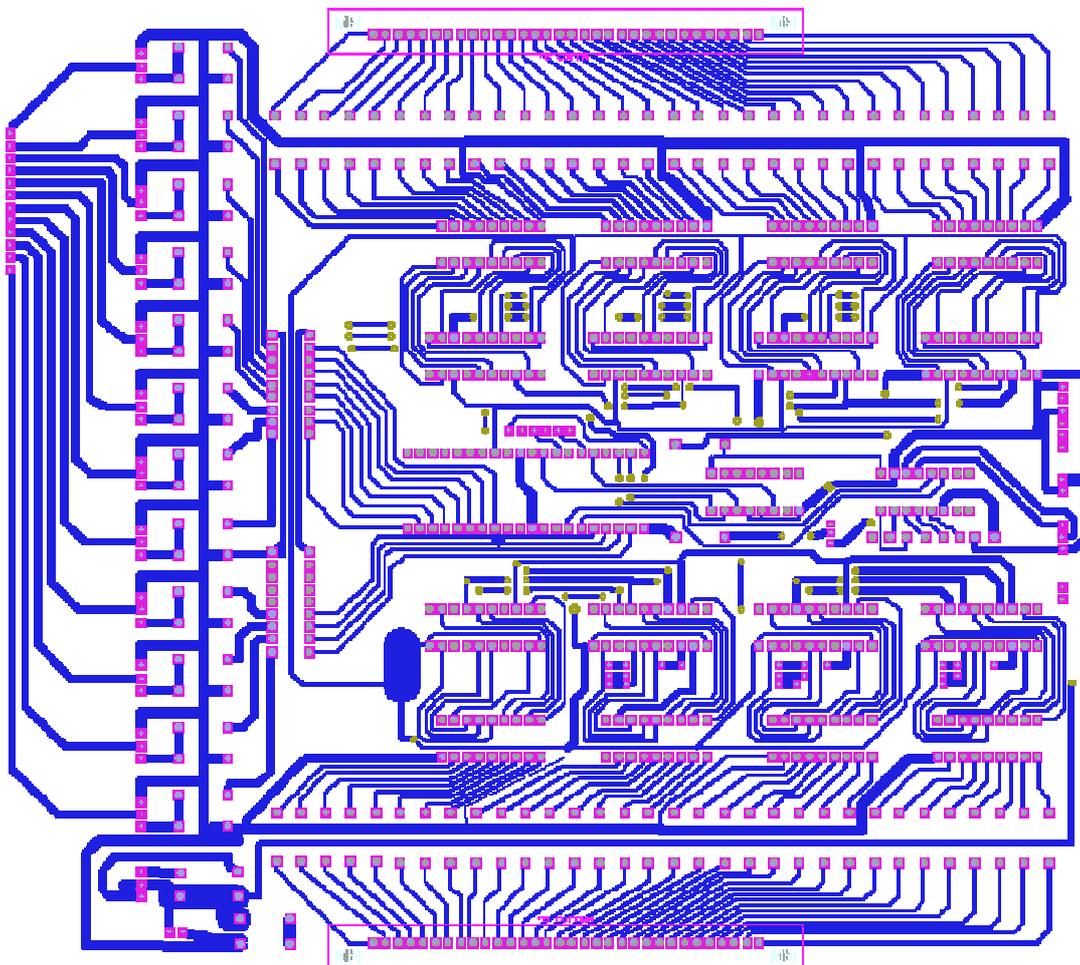


Figura 2.44: Parte posterior del controlador

Para la parte frontal del diseño se trato de optimizar la cantidad de pistas que unen a los elementos para mayor facilidad de soldadura. Como en el anterior

circuito, aquí también se muestra la figura como se miraría si la persona se encuentra frente del prototipo.

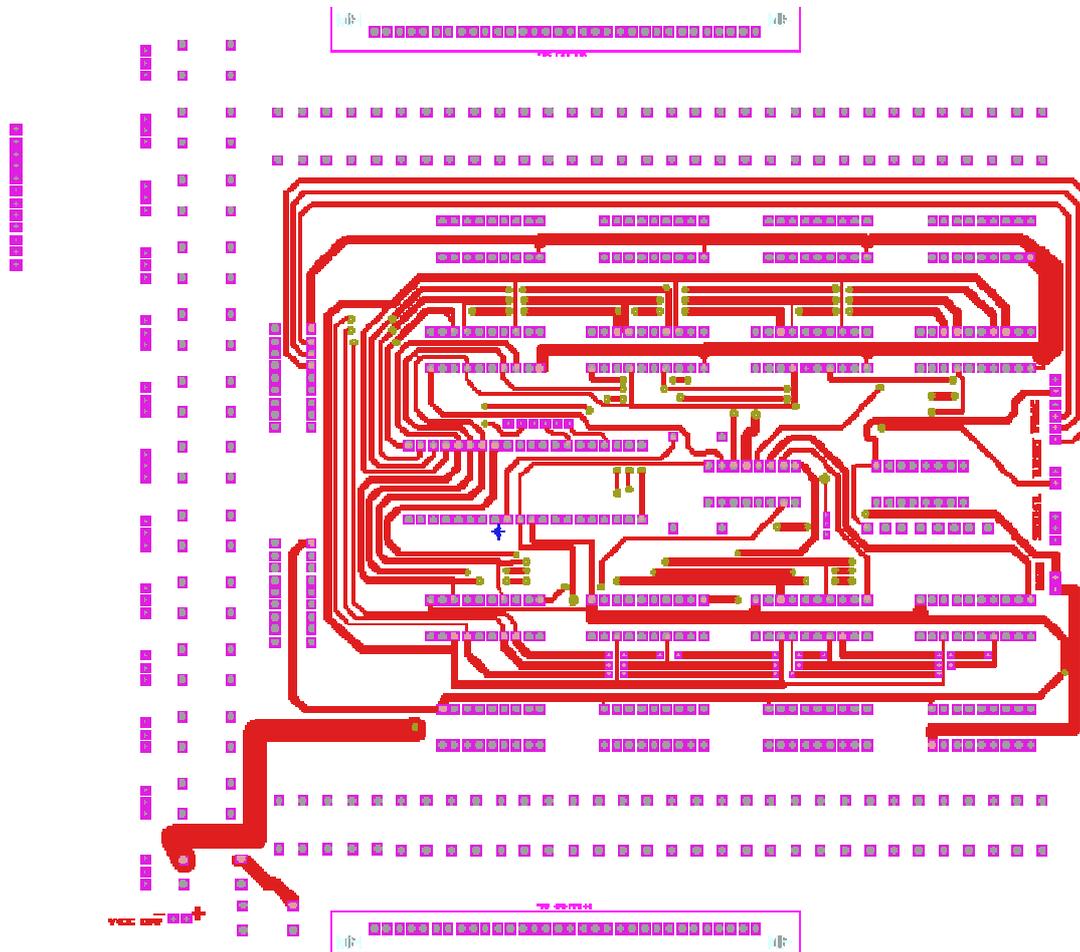


Figura 2.45: Parte frontal del controlador

En el diseño se tomó en cuenta el sitio de los conectores de alimentación, de recepción y envío de datos del circuito para mayor facilidad en la construcción del mueble que protege al prototipo y sobre todo para facilitar al usuario el manejo del prototipo.

El siguiente gráfico muestra la distribución de elementos que se acomodaron a respectivos requerimientos como los conectores para acoplar la placa de LEDs, los conectores de alimentación, datos y control del respectivo prototipo.

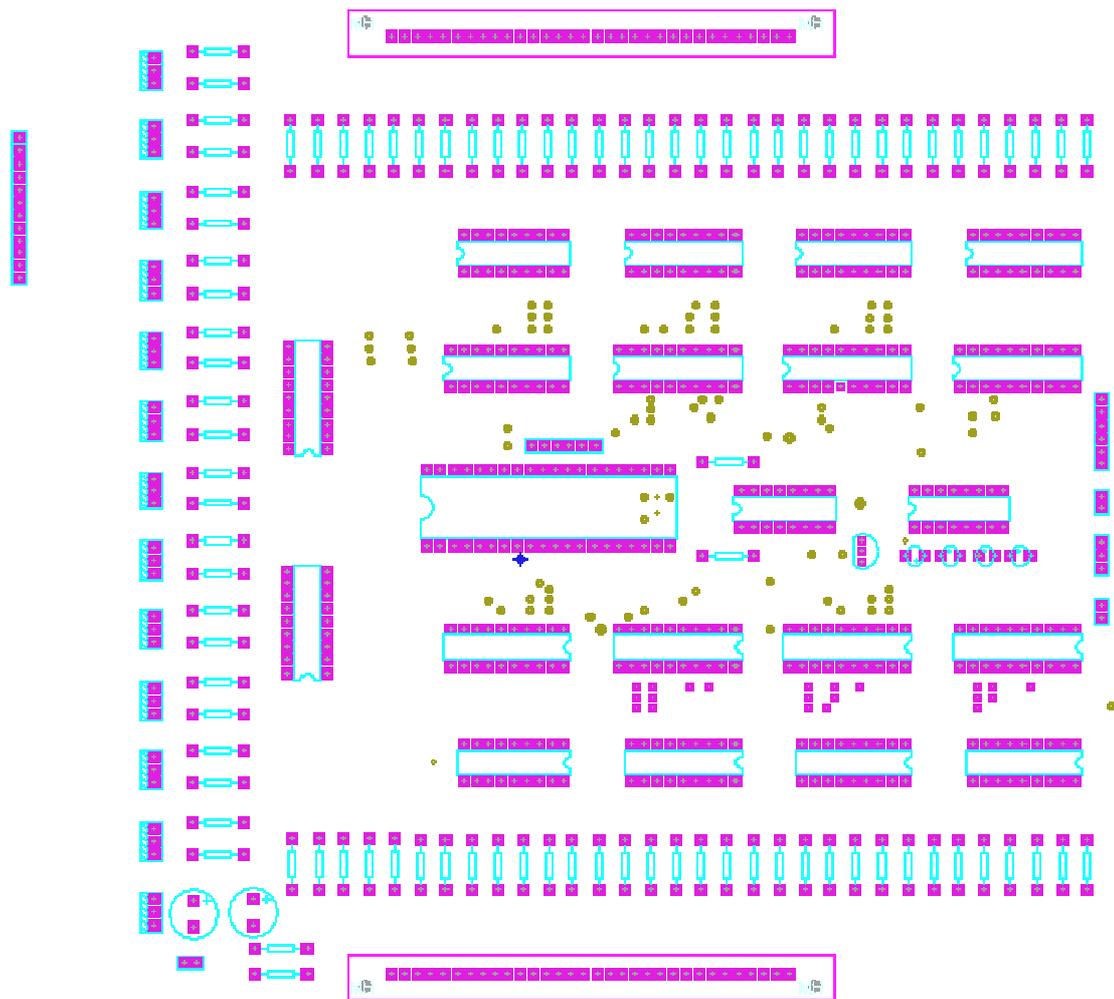


Figura 2.46: Distribución de elementos

El siguiente gráfico muestra el diseño del control del prototipo en su totalidad para información de su diseño y distribución de elementos.

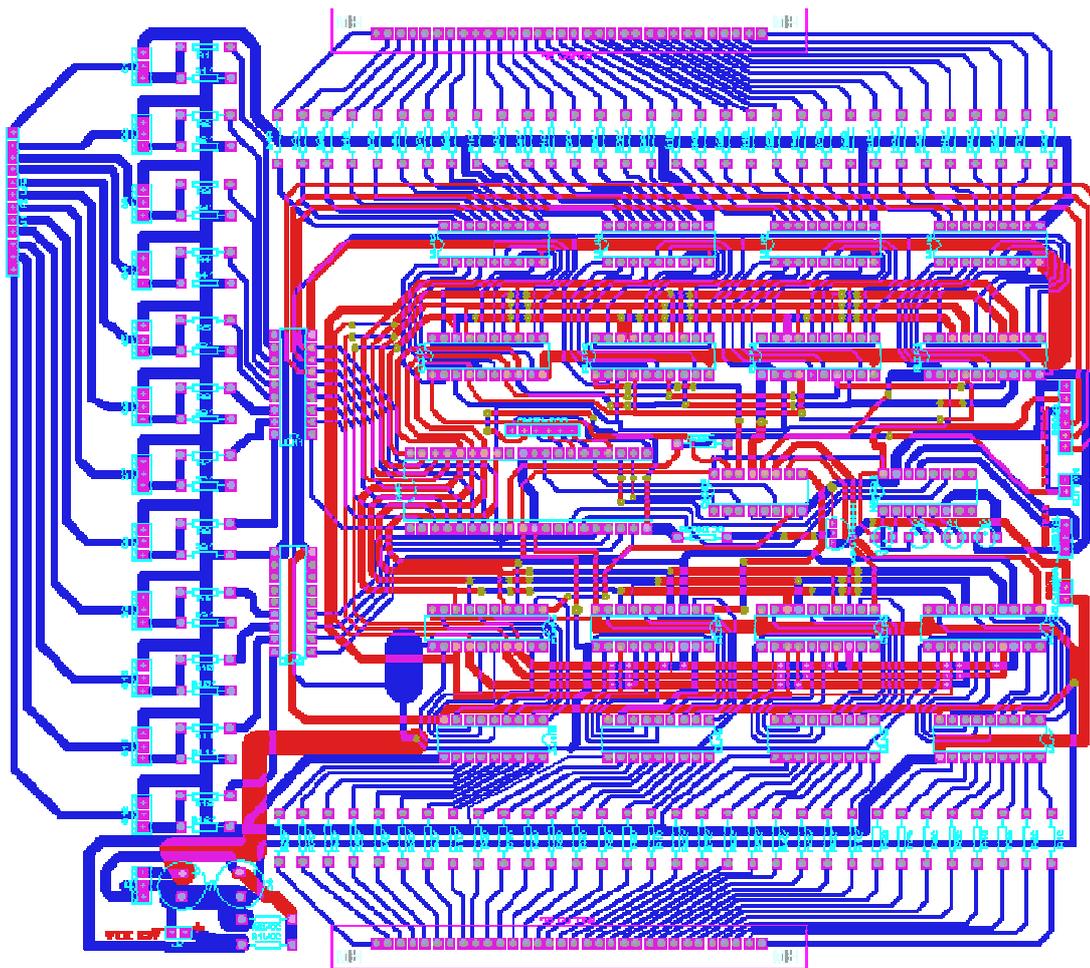


Figura 2.47: Diseño del controlador

Toda la circuitería que conforma el prototipo tiene su espacio para poderlo sujetar al mueble por medio de tornillos o pernos. El siguiente gráfico es la representación del resultado del diseño e implementación completa del control de la matriz.

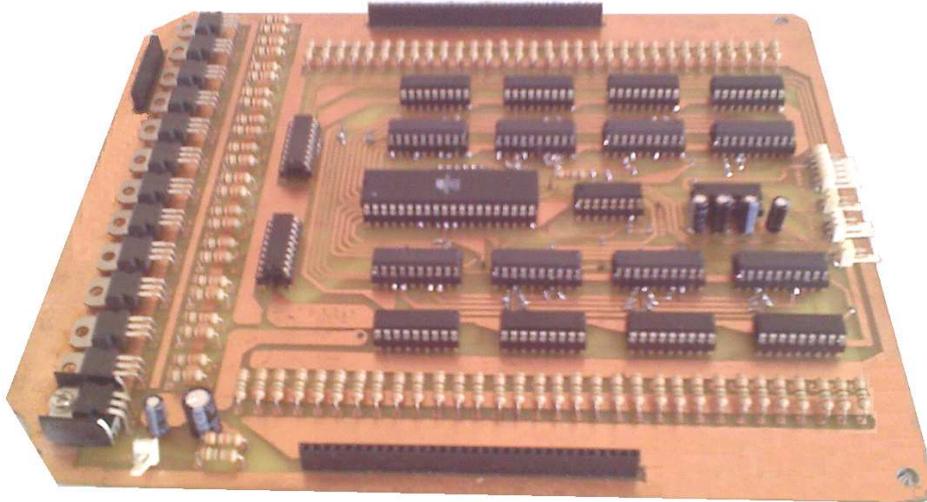


Figura 2.48: Controlador del Prototipo

Por optimización y con el fin de que se ocupe menor espacio y sobre todo por facilidad de llevar información de barrido a la matriz de LEDs se le diseño para que se colocara encima del controlador de la matriz como.

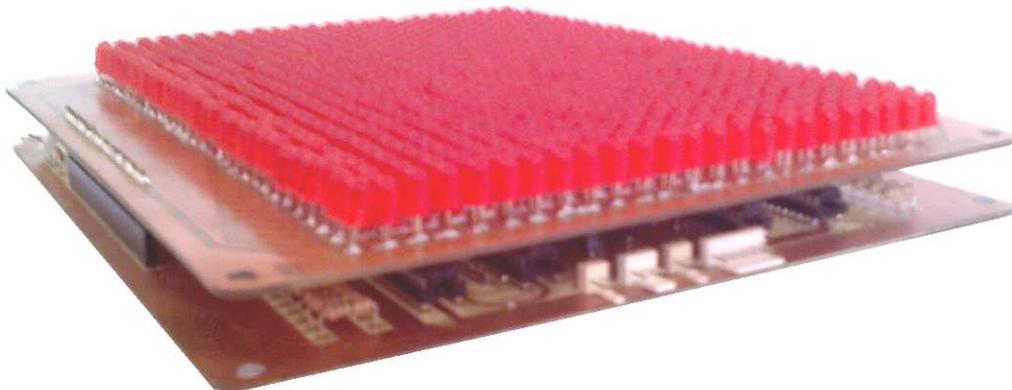


Figura 2.49: Placas del Prototipo Simplificado

Se opto por solamente dos controles del prototipo una para encendido de este y el otro para realizar el reset el mismo que permitirá realizar una nueva captura de información. Además de esto se simplifica el uso para personas con discapacidad visual ya que por medio del tacto solo tendrán que manejar los dos controles: los conectores de alimentación y de datos que se colocaron en la parte posterior del mueble tomando en cuenta la ergonomía que preste este proyecto a los usuarios que va dirigido el desarrollo del mismo, para un manejo mas agradable y cómodo.



Figura 2.50: Prototipo de matriz de LEDs

2.3.4 MATRIZ TÁCTIL

El diseño e implementación de este prototipo no solo implica circuitos y programas ahora también es necesario implementar y diseñar lo que se relaciona con electromecánica ya que incluye además de su circuitería una forma mecánica que es la que permite el movimiento de pines parte fundamental del proyecto que aquí se presenta para obtener los resultados adecuados, como en este caso es el poder llevar al tacto de alguna forma posible una imagen que se encuentra formada por pines.

2.3.4.1 Placa del controlador del prototipo

Para este diseño se tuvo que cambiar el método de multiplexación ya que el anterior utilizado para la matriz de LEDs necesita de cierta velocidad para obtener un resultado que nos permita mirar toda una imagen sin parpadeo o encendido y apagado de LEDs. En cambio para esta matriz con esa velocidad o menor no se obtiene el resultado ya que se activa el campo magnético de los electroimanes y necesita de un tiempo promedio de alrededor de 300ms para que pueda mover el objeto que realiza el mecanismo de tacto. Es por eso que decidimos utilizar LATCH que mantienen el dato del tiempo que deseamos y desactivarlo igual en un tiempo determinado.

Igual que el anterior prototipo se necesitó de la utilización de placas a doble cara ya que por la gran cantidad de elementos que se necesita implementar es más óptimo usar este tipo de placa antes que realizar un sin número de puentes que es más probable que nos lleven a cometer errores de implementación, en los dos lados de la placa lleva conectado una buena cantidad de elementos.

Utilizamos multiplexación por LATCH y lo repetimos ocho veces ya que es una matriz de 8 x 8 utilizando como amplificación Tip122 para la respectiva activación de cada electroimán.

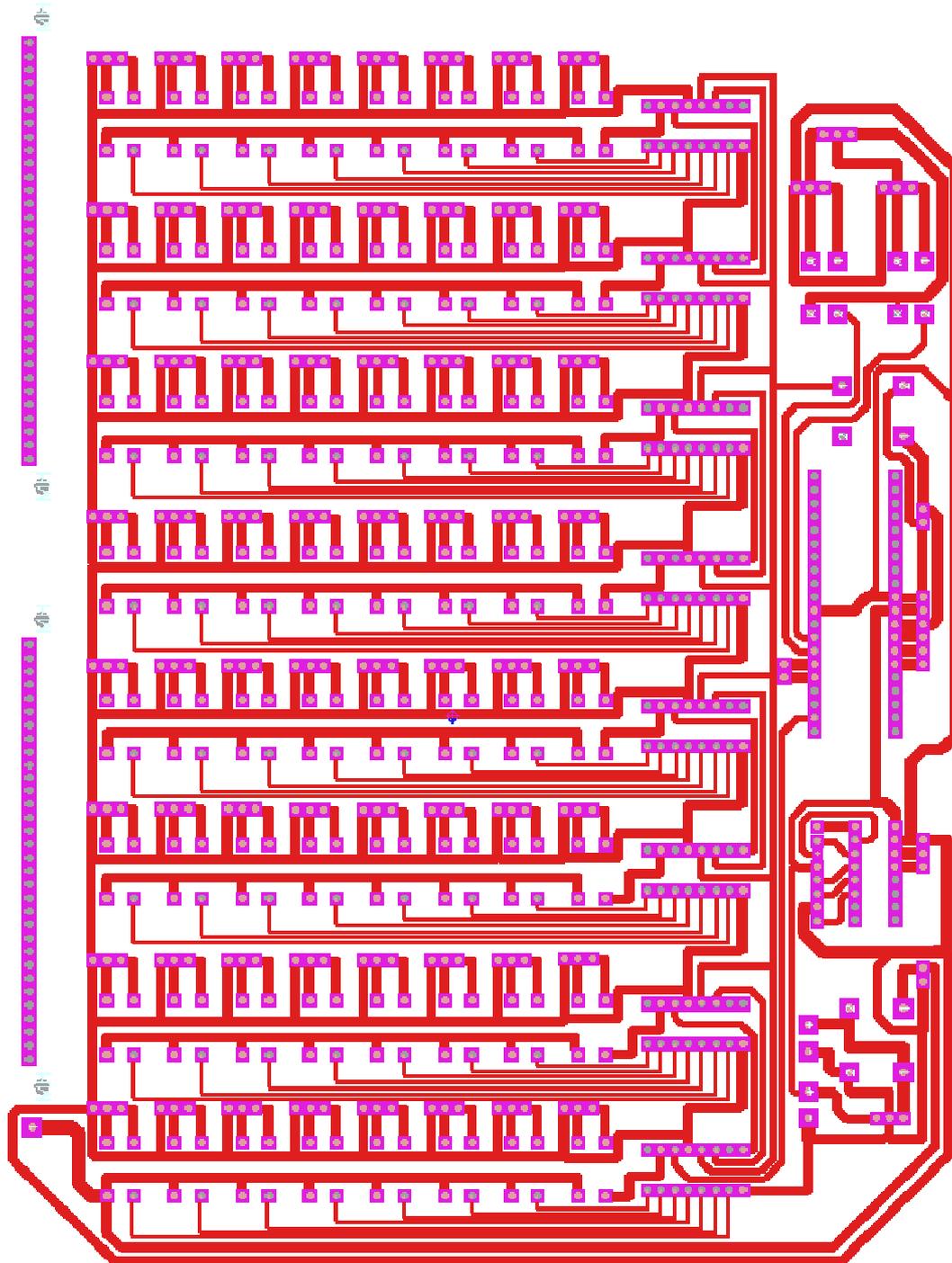


Figura 2.51: Placa posterior del control

El controlador de este prototipo es mayor que los otros en su tamaño por la cantidad de elementos que lo conforman, entre ellos están los elementos de amplificación de mayor potencia necesarios para el funcionamiento de los electroimanes que necesitan para su activación aproximadamente corrientes de 2A.

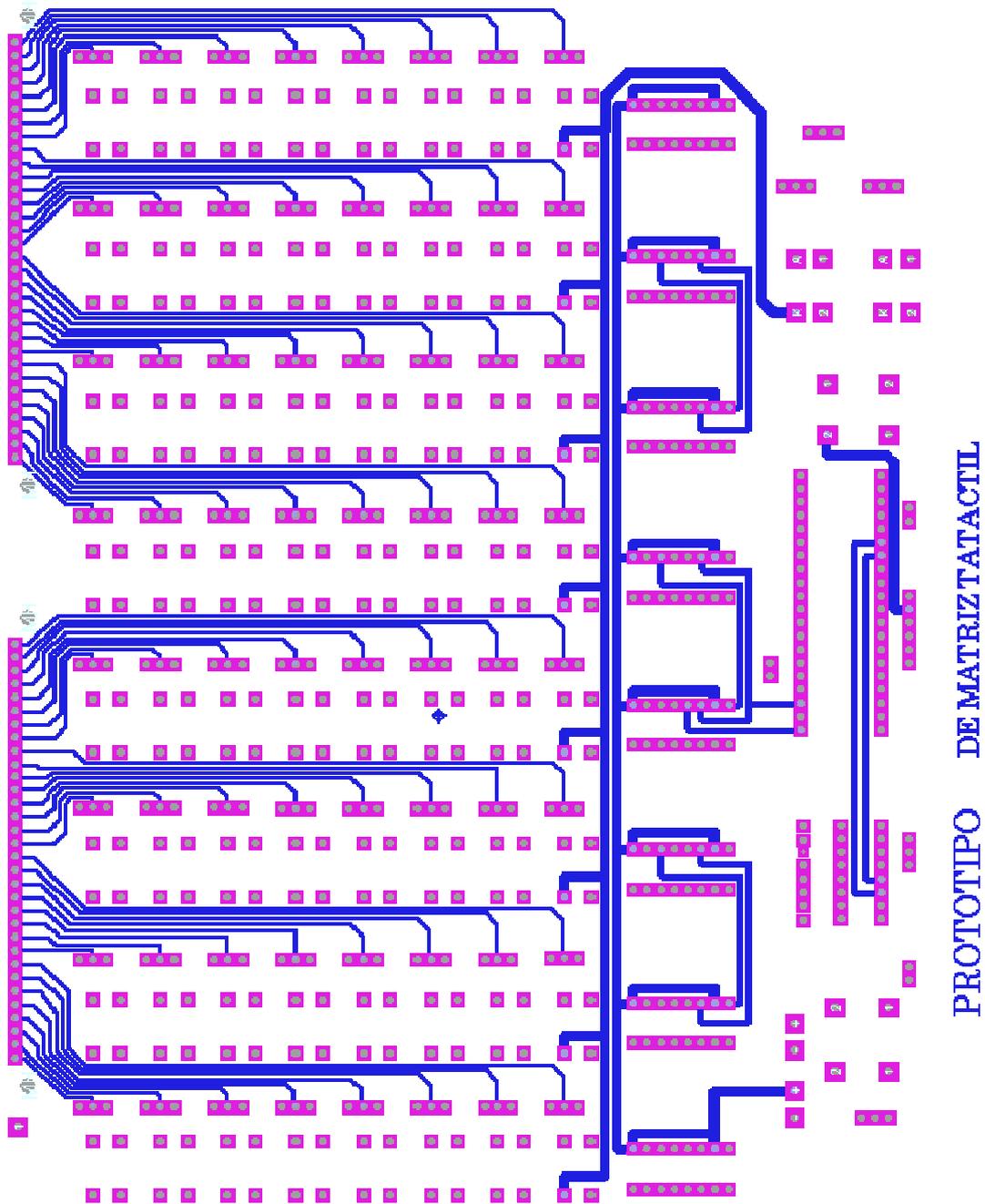


Figura 2.52: Placa frontal del controlador

Para obtener una mejor distribución y comprensión de los componentes que conforman el prototipo se colocaron los elementos en el mismo sitio que en el circuito esquemático, obteniendo una mejor distribución para colocar las salidas de estos que alimentan a la matriz de electroimanes.

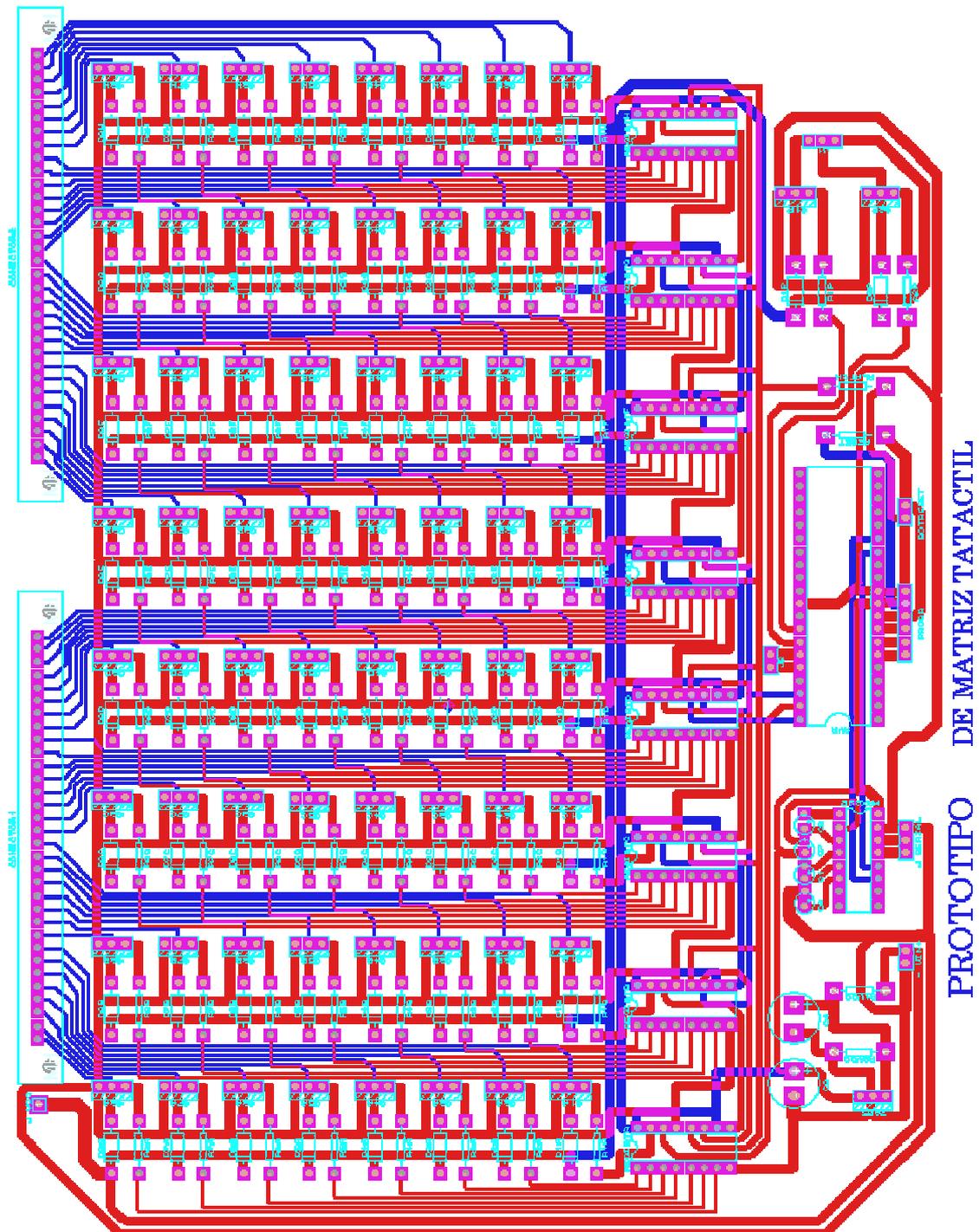


Figura 2.54: Diseño del control táctil.

La placa del control tiene un mayor tamaño que la matriz de electroimanes que explicaremos mas adelante, para que la matriz táctil sea mas pequeña se puede adoptar un sistema mecánico industrial lo cual puede llevar también a nuevos

prototipos que abarquen mayor número de píxeles sin importar el tamaño del controlador que lo conforme.

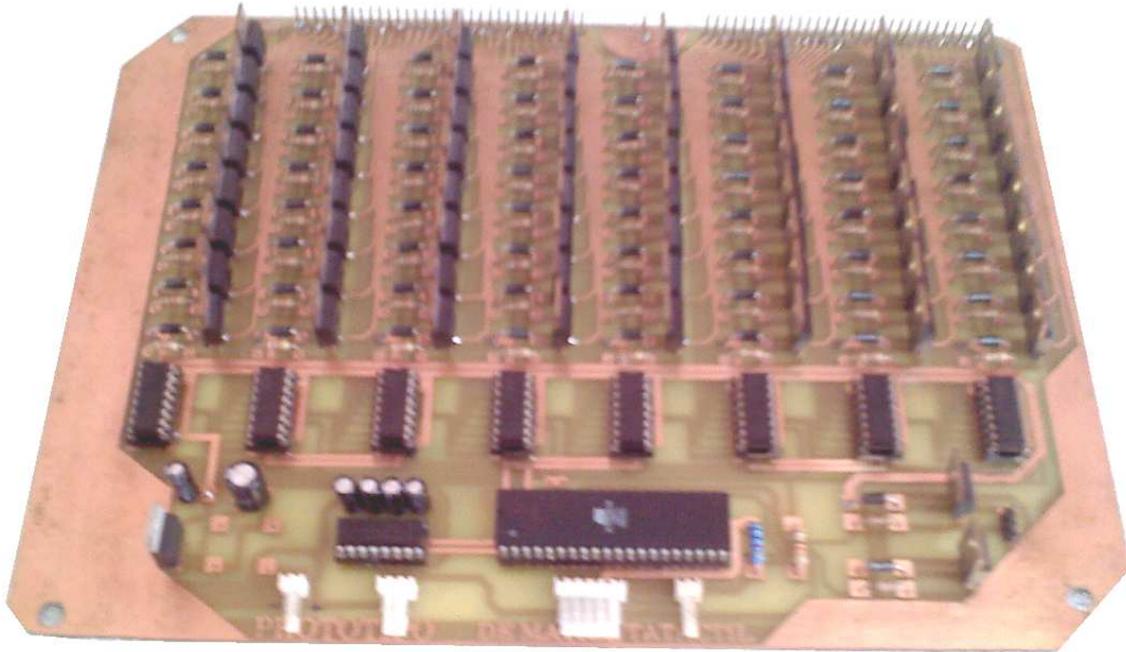


Figura 2.55: Control táctil

2.3.4.2 Matriz de electroimanes

La construcción de este prototipo se desarrollo después de una extensa búsqueda e investigación con el afán de encontrar un método aceptable para accionar un pin, mecanismo que permita que al utilizar el tacto se perciba con los nervios sensitivos de la mano formas y figuras. Su desarrollo se lo realizo poniendo en práctica conocimientos de elaboración de electroimanes y experimentando una y otra vez hasta conseguir los resultados buscados.

2.3.4.2.1 Elaboración de Electroimanes

Los electroimanes que utilizamos en este prototipo no fueron encontrados en nuestro mercado local, además sus costos en el exterior son excesivos, estas razones no llevaron a la elaboración manual de cada uno de ellos con los elementos que tenemos a disposición y podemos encontrar en nuestro medio.

Para su elaboración utilizamos tubos de plástico y placas de plástico que en nuestro caso fueron tubos de porta globos y como placas de plástico los CDs que ya no utilizamos.

Estos elementos los recortamos a las medidas que van de acuerdo al calculo del campo magnético que necesitamos y del calibre del cable que tenemos, estos los ensamblamos y los unimos con una pega resistente que en nuestro caso se uso brujita. Realizamos 64 porta bobinas porque nuestro diseño esta realizado para una matriz prototipo de 8 x 8, todo este trabajo se complemento utilizando las herramientas adecuadas como taladro, estilete, tijera y regla.

Las dimensiones del siguiente grafico esta dada en milímetros

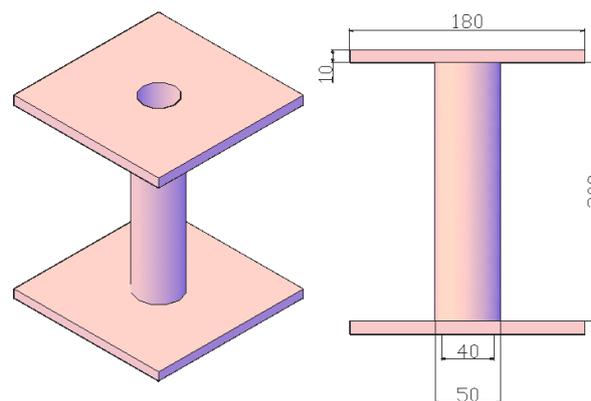


Figura 2.56: Porta Bobinas

El bobinado de los electroimanes se realizo con una maquina elaborada por nosotros con un motor de corriente continua de doce voltios adaptado con una polea para obtener mayor fuerza, el bobinado se podría realizar sin el uso de alguna máquina que también lo realizamos pero el bobinado no quedaba uniforme y eso de alguna manera nos puede influir en el desempeño del funcionamiento de la bobina lo que nos llevo a reflexionar que lo que nos puede costar en repetir bobinas, o sea tiempo y dinero, mejor lo invertimos en una pequeña pero muy útil máquina que nos dio como resultado bobinas que en forma y funcionamiento son de mejor calidad que las que empezamos elaborando solo envolviendo con

nuestras manos. Como resultado de lo descrito anteriormente fue realizar el bobinado de las espiras con una mejor estructura como se presenta en la figura 2.57.

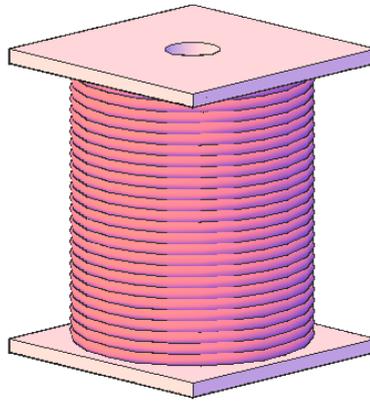


Figura 2.57: Electroimán

El núcleo para los electroimanes se diseñó en base al funcionamiento de la cabeza de la impresora matricial Epson LX 300, que constituye de un solo núcleo que llega a una distancia media de la longitud total de la bobina, esto para realizar una mayor fuerza de atracción a otro núcleo móvil, el mismo criterio es para todas las bobinas.

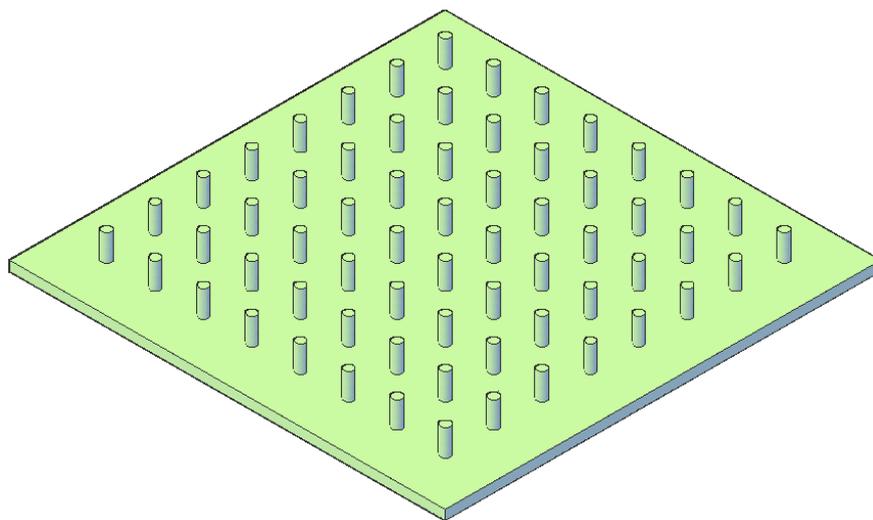


Figura 2.58: Núcleo de todos los electroimanes

Con el núcleo en común, para todos los electroimanes se obtiene una mayor fuerza de atracción que se ha observado solo por medio de experimentación más

no matemáticamente ya que no se ha encontrado algún método que resuelva este criterio en expresiones físicas.

En el prototipo tomamos una plancha metálica donde se encuentran sujetos todos los núcleos, en nuestro caso son pernos recortados a la medida adecuada y sujetos a la placa metálica para lograr el funcionamiento deseado.

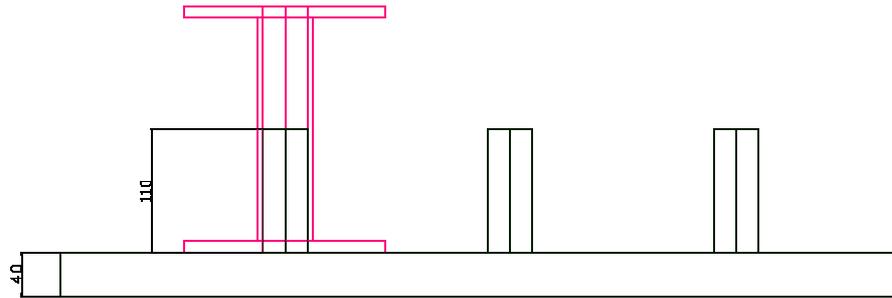


Figura 2.59: Método de Medio Núcleo para el Electroimán

Para que el mecanismo funcione de una forma adecuada colocamos un resorte de compresión helicoidal cónico en el núcleo móvil, para que en el momento de ser accionado el electroimán este regrese a su estado original de reposo.

Un resorte de compresión helicoidal cónico son usados para resistir la aplicación de fuerza de compresión o almacenar energía en forma de empuje.

El resorte es construido de alambre solidó del cable UTP que frecuentemente utilizamos en el protoboard para obtener una sensibilidad baja en el empuje al ser halado por el campo magnético accionado y luego para el retorno del núcleo móvil. En nuestro caso se diseño los resortes adecuándolos a nuestra necesidad por medio de varias pruebas.

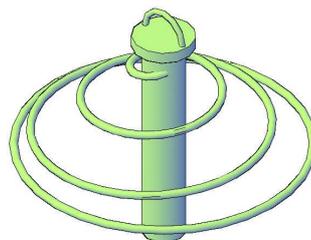


Figura 2.60: Resorte adaptado el núcleo móvil

El diseño de este prototipo de electroimán se realizó en base a la experiencia adquirida tras la búsqueda del método adecuado para minimizar la placa superior final de la matriz de pines, ya que con este método electromecánico podemos añadir una cuerda a la parte superior del núcleo móvil para halar el objeto que se mirara y se podrá tocar en la parte superior de la matriz

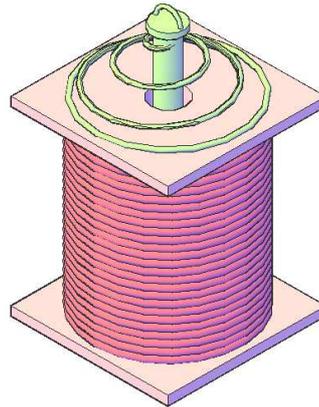


Figura 2.61: Electroimán utilizado

Cuando el electroimán se encuentra en reposo deja una pequeña distancia entre el núcleo móvil y el núcleo fijo, esta distancia la aprovechamos cuando el electroimán es accionado el núcleo móvil es atraído al núcleo fijo por medio del campo magnético y pegado a este por razones de diferentes cargas de campos que se atraen. Es en este instante cuando aprovechamos para obtener el resultado de halar un objeto que sobresalga del electroimán.

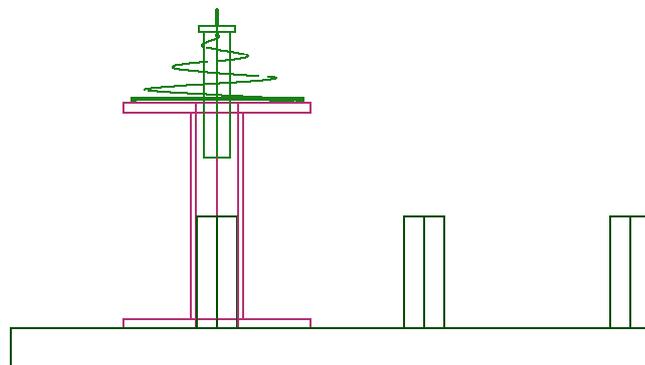


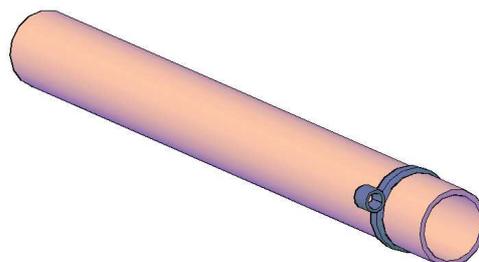
Figura 2.62: Visualización de distancias entre núcleos

La matriz de electroimanes es desarrollada en su totalidad en base al funcionamiento de las cabezas matriciales de las impresoras como ya mencionamos anteriormente, pero eso solo nos llevo a tener ideas de cómo acoplar a nuestros requerimientos, llevándonos a incluir bandas de datos como se muestra en la siguiente figura.



Figura 2.63: Matriz de electroimanes

Para completar el mecanismo que necesitamos de halar un objeto, diseñamos pequeños tubos que contiene una delgada barra la cual será guiada a la superficie de este tubo cuando el electroimán sea accionado. El tubo tiene pegado un pequeño anillo por el que será guiada la cuerda que halará la barra que saldrá a la superficie de la matriz mostrándose como un píxel de matriz táctil.



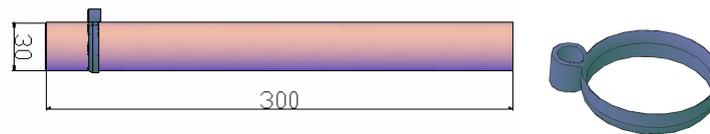


Figura 2.64: Guías para la barra táctil

La barra tiene sujeto a un extremo una pequeña bola para que la persona pueda sentirla con mayor rapidez y determinar que es un píxel de tacto, y en el otro extremo un pequeño agujero de donde se sujeta la cuerda que hala a esta barra. Los tubos se sujetan a una plancha donde están ordenados.

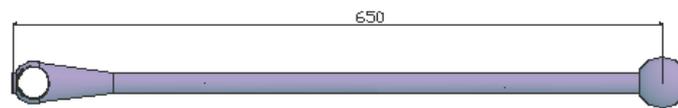


Figura 2.65: Barra táctil

Cuando el prototipo esta en funcionamiento y el electroimán se encuentra en reposo la bolita se encuentra por debajo de la plancha de tacto.

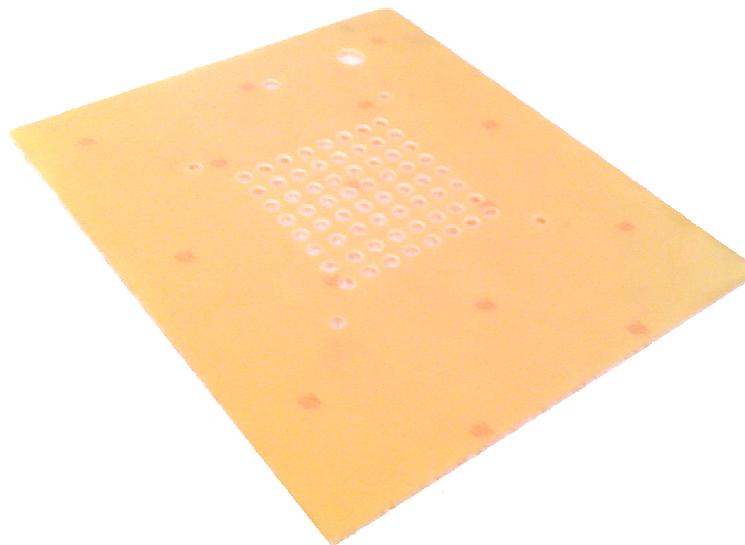


Figura 2.66: Plancha de tacto

Si el electroimán es activado, este atrae al núcleo móvil y este hala la cuerda que subirá la barra que tiene la pequeña esfera que la persona podrá tocar y sentir.

Todos los puntos que se encuentren por encima de la plancha táctil serán un píxel que conforman una imagen de un objeto.

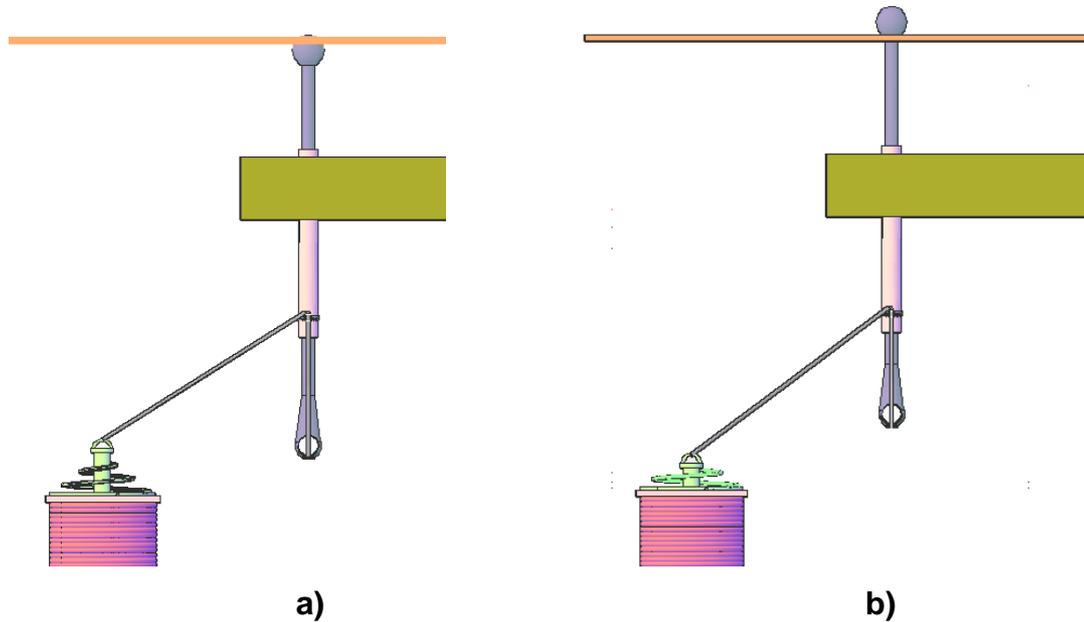


Figura 2.67: Electroimán a) en reposo y b) activado

Todos los elementos descritos anteriormente son sujetos a un chasis para mejorar la distribución de los elementos y placas que forman parte de la matriz de pines, la utilización del chasis de alguna manera nos facilita si fuera el caso de realizar cambios en el control del prototipo y en la matriz de electroimanes, es decir tanto en lo eléctrico como en lo mecánico.



Figura 2.68: Chasis del prototipo

El chasis esta diseñado para sostener los componentes del prototipo por pisos, esto es en la parte inferior del chasis va el controlador, luego la matriz de electroimanes, luego esta el sujetador de tubos para las guías de las barras y por ultimo donde sujetara la plancha táctil.

Se ha seguido este modelo para obtener algunas ventajas como:

- El orden de los niveles llegan a un ultimo que presenta el resultado final
- La manipulación y construcción del prototipo es mucho mas sencilla
- En el caso de existir daños en alguna de las placas, podemos retirar totalmente la placa y realizar el cambio correspondiente.
- El atado de los pines de tacto es mucho más sencillo con todo el chasis descubierto.



Figura 2.69: placas en sus respectivos niveles

Con la sujeción de la matriz de electroimanes en el chasis del prototipo ya podemos atar los pines que sobresaldrán en la placa superior mostrando letras, números o figuras.

Cada pin de tacto será conectado a un electroimán por medio de una cuerda delgada, en nuestro caso utilizamos hilo coser, que esta unido de la cabeza del resorte sujeto al núcleo que ingresa al electroimán, pasando por un anillo sujeto en el tubo que guía al pin de tacto y por ultimo se asegura la delgada cuerda al extremo del pin, realizando de esta manera el diseño mecánico.

En funcionamiento del electroimán atrae al núcleo móvil pegándose este al núcleo en reposo, esto hace que se mueva el pin de tacto por medio de la cuerda atada a este. Cuando una persona toque la placa en la que sobresale los pines, estos no se hundirán de la superficie ya que el campo electromagnético generado por los electroimanes es lo suficiente mente fuerte para que los núcleos móvil y estático permanezcan unidos.

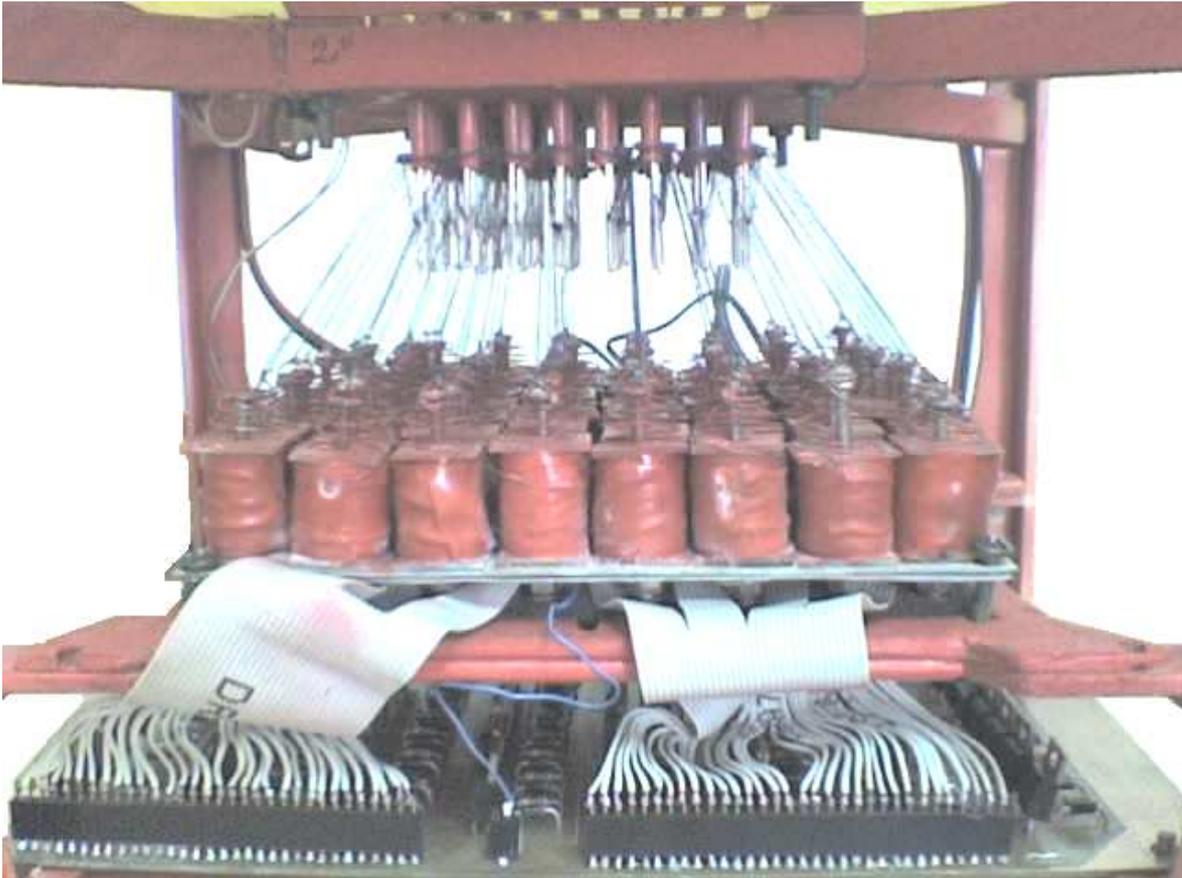


Figura 2.70: Atado de los pines al electroimán

Se tomo esta forma de atar el hilo de los pines a los electroimanes, con la idea de poder minimizar el tamaño de la matriz táctil que sobresale en la superficie, haciendo que el tamaño del controlador sea indiferente al resultado en el tamaño de la matriz táctil y de esta manera poder incluir un mayor número de pines y aumentar el tamaño y resolución de imágenes mas complejas.

2.4 COSTOS DEL PROTOTIPO

<i>Cantidad</i>	<i>Elementos</i>	<i>costo Individual</i>	<i>total</i>
2	Atmega16	7,5	15
8	ULN2803	0,55	4,4
2	MAX232	1,8	3,6
1	74LS138	0,6	0,6
8	74LS373	0,45	3,6
8	74HC595	0,75	6
12	TIP127	0,75	9
64	TIP122	0,5	32
64	Diodo 1N4007	0,09	5,76
1	Disipador	0,55	0,55
2	DB9	0,35	0,7
10	Espadines	0,87	8,7
148	Resistencias 1/2 W	0,015	2,22
10	Capacitores Electroliticos 10uF	0,1	1
2	Capacitores Electroliticos 100uF	0,25	0,5
768	Diodos LED	0,07	53,76
2	Socalos 20 pines	0,9	1,8
29	Socalos	0,47	13,63
4	Estaño	2,85	11,4
3	Baquelita 20X30	9,5	28,5
12	Cloruro	0,6	7,2
6	Hojas De Transferencia	1	6
4	Pulsador	0,25	1
1	Cable bobinado	90	90
1	Clavos 2"	1,3	1,3
2	Cable	0,5	1
10	Tabla Triplex	0,8	8
1	Vidrio	1,6	1,6
1	Marcador Permanente	1,8	1,8
10	pega	0,25	2,5
2	Laminas Hierro	0,5	1
70	Pernos	0,1	7
2	Cinta	0,7	1,4
5	Plastico	0,5	2,5
1	Tubos	0,8	0,8
3	Agujas	2,5	7,5
1	Hilo	0,5	0,5
1	Bolitas	0,5	0,5
1	Espray	3	3
1	Cautin	3,9	3,9
4	Brocas	0,5	2
1	Programador	20	20
1	Fuente	24	24
	Total:		248,03

CAPITULO 3

3.1 Pruebas, resultados y discusión

Las pruebas realizadas con el prototipo funcional nos dieron resultados satisfactorios, ya que se tomo en cuenta como una clasificación importante para el uso del dispositivo la naturaleza de la limitación visual, es por este motivo que se realizo no solo un prototipo si no tres: el software desarrollado con voz sintética y OCR que por sus características de funcionamiento es de gran utilidad a personas limitadas completamente de la vista o con una limitación visual menor al 100% pero con sentido auditivo en buenas condiciones, el segundo prototipo, una matriz táctil que puede ser utilizada por personas con limitación visual completa como por personas con limitación visual parcial y que no necesariamente deberán tener el sentido auditivo en perfectas condiciones.

Esto ha permitido que no se encuentre mayor dificultad de aprendizaje para personas invidentes de nacimiento a pesar de que no poseen ninguna representación del mundo y mucho menos con las personas que adquirieron la discapacidad a lo largo de sus vidas quienes pueden asociar con mayor facilidad en el caso de reconocimiento de imágenes.

Hemos llegado a lo que significa para nosotros la parte más importante de este proyecto y lo consideramos de esta manera sin dejar a un lado todo cuanto ha intervenido para llegar hasta aquí, es decir todo el camino recorrido para obtener nuestro prototipo funcional esto implica todo lo que fue necesario investigar y estudiar para ver realizado nuestro proyecto. Pero una vez que nuestro proyecto esta funcional completamente, nos da la satisfacción de haber alcanzado los objetivos planteados pero el sentir de que el proyecto cumple con lo que nos hemos propuesto esta incompleto ya que falta la validación del mismo y que mejor manera de llevar el proyecto que aquí se presenta a los usuarios para quienes

van dirigidos este trabajo para lo cual gracias a la colaboración del COLEGIO FISCOMISIONAL CATOLICO MIXTO LEONARDO PONCE POZO " Integrado para no Videntes " ubicado en la Av. Florida 15-51 y Occidental, pero sobre todo gracias a la colaboración del joven Juan Gabriel Bolaños Benítez quien es estudiante de esta prestigiosa institución, el mismo que tiene deficiencia visual completa y adquirida desde los 2 años de edad y quien nos permitió tener una respuesta real de la utilidad del proyecto para continuar o empezar a adquirir conocimientos con las herramientas que dispone el mismo.

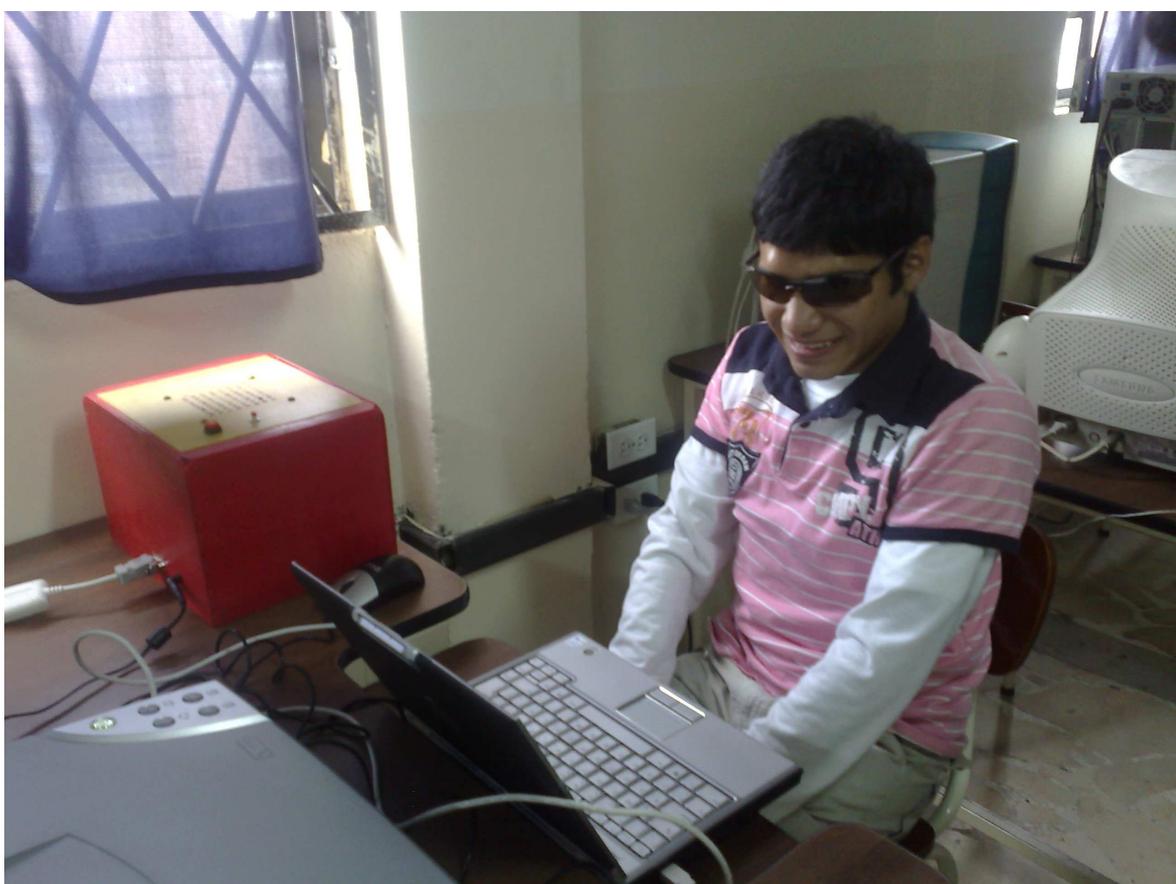


Figura 3.1: Prueba del proyecto

Con la asistencia del joven Juan Bolaños se realizó la validación de nuestro trabajo llevando a conclusiones que fueron expresadas por el mencionado estudiante y también a las que hemos llegado nosotros después de tan agradable experiencia.

El proyecto es de gran ayuda para personas no videntes ya sea con la utilización del teclado como con la utilización del ratón ya que las dos opciones tienen sus beneficios individuales pero con el mismo fin.



Figura 3.2: Prueba del proyecto

Con el uso del teclado es una muy buena alternativa debido a que estudiantes específicamente del colegio al que pertenece nuestro asistente Juan Bolaños tiene un conocimiento de ubicación y orientación de donde está cada tecla como si se tratara de una persona vidente, con la diferencia de que para ejecutar opciones se tienen que memorizar un sin número de comandos y es ahí donde radicaría en algo la dificultad aunque esto no es una barrera nos comentaba Juan Bolaños, esto es superado y da mejores prestaciones nuestro proyecto al utilizar únicamente el ratón de la computadora y solo haciendo clic y escuchando la opción que desea ejecutar.

Nos pudimos dar cuenta que nuestro trabajo es de fácil manipulación y no fue necesario una instrucción avanzada para que el usuario haga de este proyecto una herramienta que él la pueda dominar de forma rápida y precisa.



Figura 3.3: Prueba del proyecto

La matriz táctil permite ver la realidad desde otra perspectiva saliendo de los márgenes de braille que por ser este un sistema estándar que se lo conoce a nivel mundial y que la mayoría por no decir todas las personas discapacitadas lo utilizan, se llegó a pensar por un señor profesor de la institución a la que pertenecen Juan Bolaños que podría llegar a confundir al estudiante, lo que fue corregido por el mismo Juan Bolaños quien sin antes decirle lo que hacía la matriz táctil nos mencionó que desearía poder saber formas y gráficas, y fue entonces que le comentamos que eso también realiza nuestro proyecto y él muy entusiasmado de saber la forma real de las letras y algunas gráficas que tiene nuestra matriz ejecuto esta parte del programa y le pareció algo que se sale de los límites de braille donde no le puede conocer la forma real de la letra y que al

contrario de lo que pensó el profesor a Juan Bolaños le pareció algo que va a ayudar a ampliar los horizontes de los conocimientos.



Figura 3.4: Prueba del proyecto

3.2 Resultados del sintetizador de voz

El sintetizador de voz es una de las partes importantes del proyecto que aquí se presenta del cual se realizaron pruebas que como resultado de las mismas se llegó a la conclusión que guarda relación con los objetivos que se deseaba alcanzar debido a que cumple con las siguientes características:

- Reproduce texto externo es decir, texto impreso ya sea de revistas, libros, etc.
- Reproduce texto en formato mp3.
- Reproduce texto digitalizado.

- Todas las características antes mencionadas son ejecutadas de forma automática.

3.3 Resultados de la matriz de LEDs

La matriz de LEDs es un prototipo de utilidad para personas con deficiencia visual baja y para potencializar su utilidad se requiere de diodos de alta luminosidad que en este proyecto no han sido utilizados debido a factores costo.

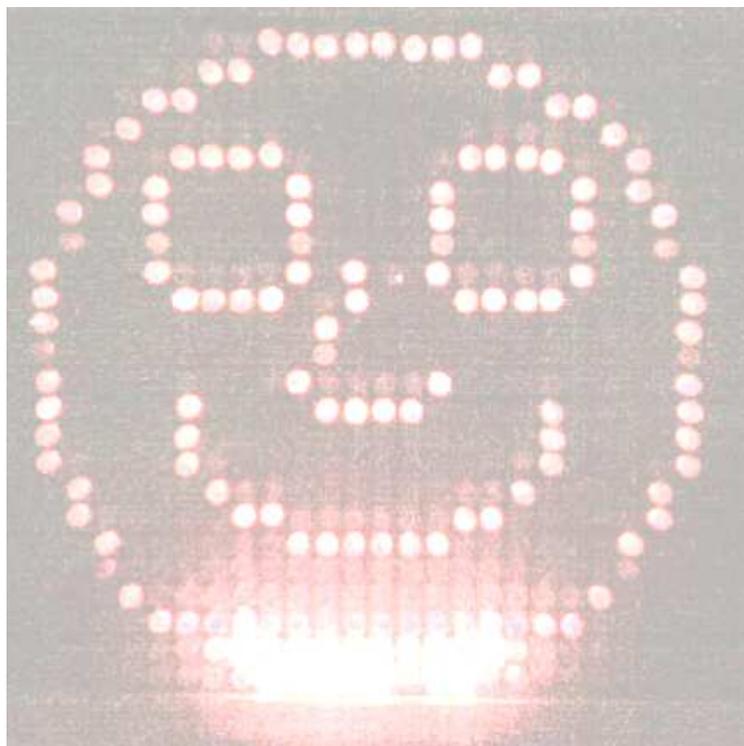


Figura 3.5: Grafica de carita feliz representada mediante matriz de LEDs

En la figura anterior se despliega una de las graficas que fueron diseñadas para la matriz de LEDs la cual por efectos de luz en la fotografía no se aprecia de la misma manera que cuando se tiene la matriz físicamente al alcance.

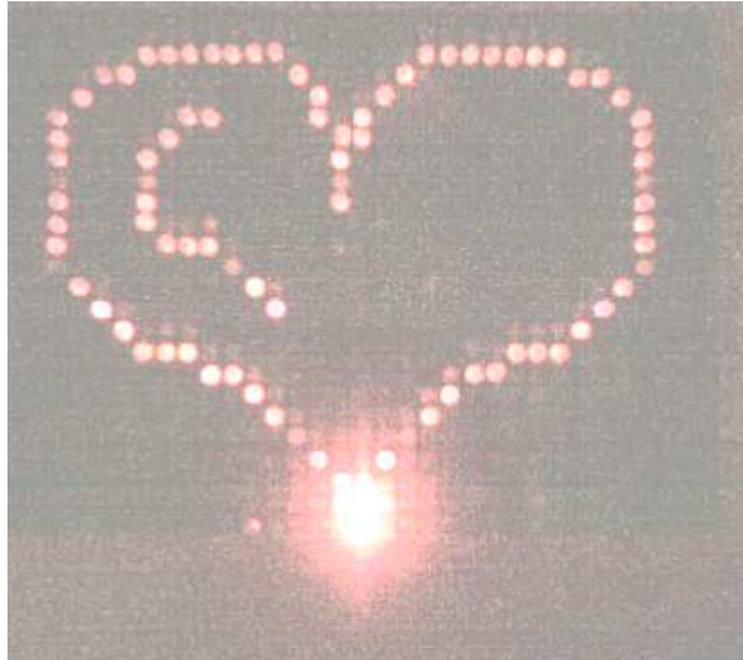


Figura 3.6: Grafica de corazón representado mediante matriz de LEDs

Esta es otra presentación de las gráficas diseñadas para la matriz de LEDs, se ha tomado como base graficas sencillas tanto de elaborar como de desplegar en la matriz debido al alcance que nos da la resolución que hemos escogido de 24X32 filas y columnas respectivamente. Se escogió esta resolución ya que esta matriz es de tipo demostrativo.

Debido a que uno de los propósitos de la matriz es con relación al aprendizaje se presenta en las siguientes figuras lo que es el abecedario y los números naturales y con el fin de que cada uno de estos elementos se figure de una manera entendible para el usuario se tomo la resolución mínima necesaria que es de 7X5 filas y columnas respectivamente



Figura 3.7: Letras del abecedario representados mediante matriz de LEDs

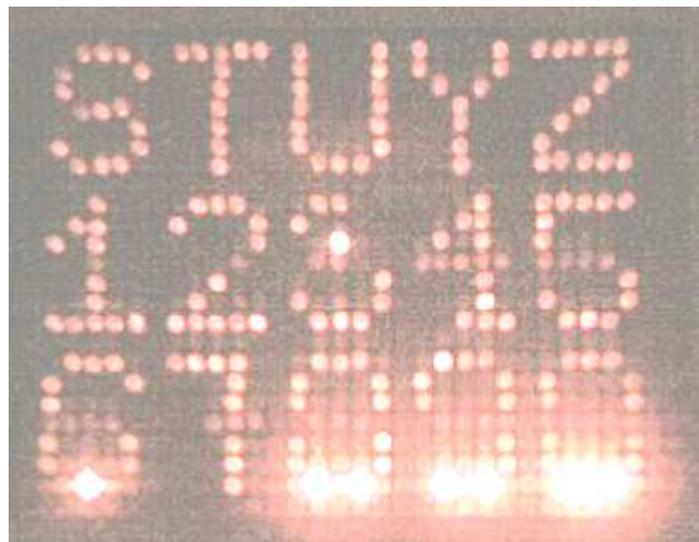


Figura 3.8: Letras y números representados mediante matriz de LEDs

En cada una de las figuras anteriores desplegadas en la matriz de LEDs podemos observar que no todos los LEDs tienen la misma luminosidad ya que son productos que comercialmente se encuentran de esa manera, es decir al comprar los 768 LEDs que utilizamos en la matriz unos vinieron con mayor luminosidad que otros y esto se noto al ser usados todos los LEDs en un solo conjunto, para solucionar de alguna manera este inconveniente que afectaba para una clara visibilidad de las graficas nosotros decidimos probar uno a uno los LEDs para

poder seleccionarlos y colocarlos en un solo tramo para tratar de unificar y ver un resultado uniforme.

3.4 Resultados de la matriz de tacto

En la matriz de tacto se representa números y letras uno a uno de forma individual, se ha optado esta forma de representarlos porque la matriz de tacto es un prototipo de lo que se podría hacer en dimensiones grandes y con fines más específicos, la resolución que se utiliza es la suficiente para representar y sobre todo que se figure de manera entendible cada letra o número que el usuario desee.

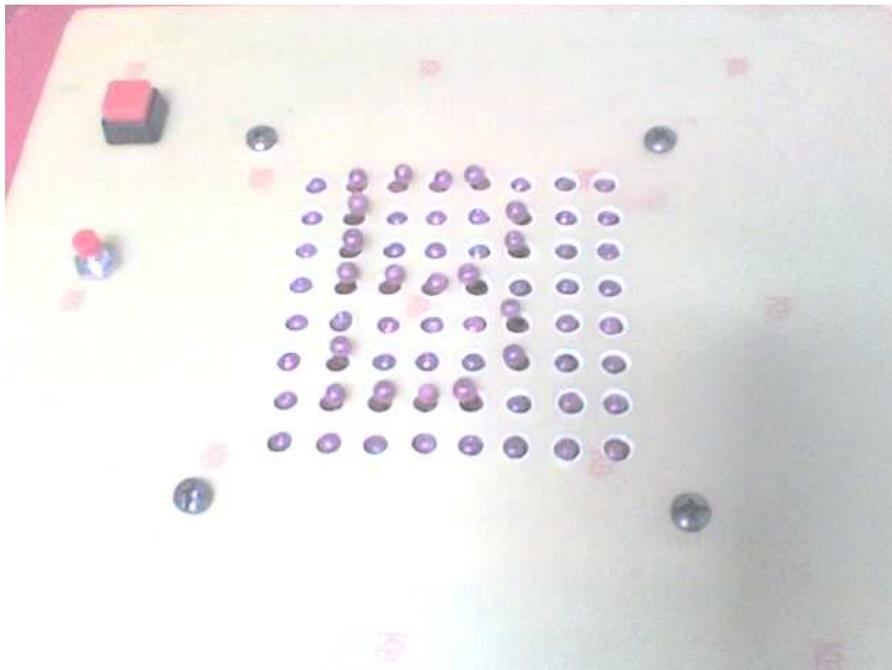


Figura 3.9: Representación de la letra B mediante matriz de pines



Figura 3.10: Representación del número 1 mediante matriz de pines

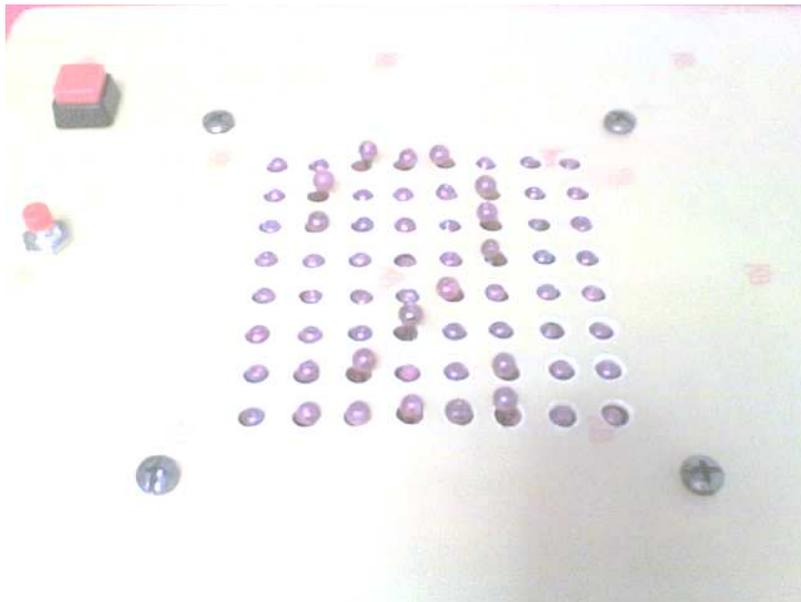


Figura 3.11: Representación del número 2 mediante matriz de pines

CAPITULO 4

CONCLUSIONES Y RECOMENDACIONES

4.1 CONCLUSIONES

- Este proyecto es un avance hacia la independencia y desarrollo en muchos campos de las personas con limitaciones visuales, porque se ha desarrollado no solo una ayuda para tener acceso a información externa sino dos: por medio de una matriz táctil, voz sintética y OCR
- Queda trazado el camino hacia la eliminación de barreras permitiendo que personas con limitaciones visuales accedan de una manera más equitativa a los beneficios que significa tener todo tipo de información a su alcance sin importar si esta o no en Braille.
- Se construyó un prototipo completamente funcional.
- En la actualidad la automatización de un procedimiento, está basada en un sistema computacional, el cual cuenta con herramientas necesarias para realizar diferentes tipos de tareas, estas herramienta son accesibles y asequibles.
- Con el desarrollo de nuestro proyecto, se diseño, construyo, ensamble circuitos impresos y materiales electromecánicos, además se ha logrado automatizar un proceso a través de la digitalización y posterior tratamiento de imágenes capturadas a través de una Webcam,. Por lo tanto, se está colocando en práctica los conocimientos en diversas áreas tales como Electrónica, Máquinas Eléctricas, Electromecánica, Programación y Procesamiento Digital de Imágenes, etc.

4.2 RECOMENDACIONES

- Entre algunas de las recomendaciones principales de este proyecto, es la adecuada iluminación que debe existir para una eficaz captura y digitalización de la imagen.
- Al igual que la comunicación RS - 232 asíncrona, la comunicación USB debe mantener la referencia entre la PC y el microcontrolador caso contrario la PC no registra o reconoce al dispositivo.
- Para futuros avances en el proyecto es recomendable que se hagan nuevas investigaciones que permitan encontrar un diseño que optimice todas las características del prototipo para de esta manera además de prestar una mejor ayuda para personas con limitaciones visuales, cubra con factores de diseño industrial.
- El prototipo debe evolucionar aumentando el número de electroimanes por área cuadrada ya que esto dará mejores resultados en cuanto a resolución se trata, es decir se podrá identificar imágenes con un grado mayor en detalles y rasgos que forman la misma.

BIBLIOGRAFIA

- Documentación del Toolbox de Image Processing de Matlab
www.mathworks.com
- Castleman, K.R.: "Digital Image Processing", Prentice-Hall, Englewood Cliffs, New Jersey 07632, 1996.
- Gonzalez, Rafael C. and Woods, Richard E. "Digital Image Processing", *Prentice Hall*: Upper Saddle River, N.J., 2002.
- RCA-Radiotron Designer's Handbook, 4 th. edition, 1953
- Circuitos Magneticos y Transformadores- EE Staff del M.I.T. , 1965
Editorial Reverte, España.
- Vademecum de radio y electricidad, Ing. E. N. packmann, 1971
Editorial Arbo, Argentina.

Paginas Web:

- http://www.revistasaludocular.org/journal/01_01.htm
- <http://fundacionhomero.blogdiario.com/>
- <http://www.ugr.es/~iramirez/Defvisual.doc>
- <http://www.sidar.org/publica/press/recortes/25ciegos.htm>
- <http://www.artistas.org.ar/notas/discapacidadvisual.htm>
- <http://www.sidar.org/publica/press/recortes/25ciegos.htm>
- <http://www.grupofundosa.es>
- <http://www.amiresmadrid.com/imagenes/miopia.jpg>
- <http://www.webpersonal.net/felinia/oido.JPG>
- <http://www.manolo.net>
- http://es.wikipedia.org/wiki/Magnet%C3%B3fono_de_bobina_abierta
- <http://bp3.blogger.com>
- <http://cidat.once.es/appimg/cidat/02090100184.jpg>

- http://es.wikipedia.org/wiki/Imagen:Braille_Writer.jpg
- <http://www.elmundo.es/navegante/2001/04/04/laimagen/986378672.html>
- <http://blogs.europapress.es/tecnologia/archive/2007/07/12/top-braille-un-traductor-para-ciegos.aspx>
- <http://www.tomaszt.com/electrotecnia/magnetismo/figuras/fig10-19.jpg>
- <http://www.geocities.com/CapeCanaveral/2404>
- <http://www.ladiscapacidad.com/tecnologiaydiscapacidad/peroqueeseljaws/peroqueeseljaws.php>
- <http://wiki.tiflolinux.org/wiki/Orca>

ANEXO 1

SOFTWARE DEL MICROCONTROLADOR PARA MATRIZ DE LEDS

```
$regfile = "m16def.dat"
```

```
$crystal = 1000000
```

```
$baud = 4800
```

```
' *** Configuración de puertos
```

```
Config Porta = Output
```

```
Config Portb = Output
```

```
Config Portc = Output
```

```
Config Portd = Output
```

```
Const Temp = 0.01
```

```
' *** Variables
```

```
Dim Dat1 As String * 8
```

```
Dim Dat2 As String * 8
```

```
Dim Dat3 As String * 8
```

```
Dim Dat4 As String * 8
```

```
Dim Dat5 As String * 8
```

```
Dim Dat6 As String * 8
```

```
Dim Dat7 As String * 8
```

```
Dim Dat8 As String * 8
```

```
Dim Dat9 As String * 8
```

```
Dim Dat10 As String * 8
```

```
Dim Dat11 As String * 8
```

```
Dim Dat12 As String * 8
```

```
Dim Dat13 As String * 8
```

```
Dim Dat14 As String * 8
```

```
Dim Dat15 As String * 8
```

```
Dim Dat16 As String * 8
```

```
Dim Dat17 As String * 8
```

```
Dim Dat18 As String * 8
```

```
Dim Dat19 As String * 8
```

```
Dim Dat20 As String * 8
```

```
Dim Dat21 As String * 8
```

```
Dim Dat22 As String * 8
```

```
Dim Dat23 As String * 8
```

```
Dim Dat24 As String * 8
```

```
Dim Dat25 As String * 8
```

```
Dim Dat26 As String * 8
```

```
Dim Dat27 As String * 8
```

```
Dim Dat28 As String * 8
```

```
Dim Dat29 As String * 8
```

```
Dim Dat30 As String * 8
```

Dim Dat31 As String * 8
Dim Dat32 As String * 8
Dim Dat33 As String * 8
Dim Dat34 As String * 8
Dim Dat35 As String * 8
Dim Dat36 As String * 8
Dim Dat37 As String * 8
Dim Dat38 As String * 8
Dim Dat39 As String * 8
Dim Dat40 As String * 8
Dim Dat41 As String * 8
Dim Dat42 As String * 8
Dim Dat43 As String * 8
Dim Dat44 As String * 8
Dim Dat45 As String * 8
Dim Dat46 As String * 8
Dim Dat47 As String * 8
Dim Dat48 As String * 8
Dim Dat49 As String * 8
Dim Dat50 As String * 8
Dim Dat51 As String * 8
Dim Dat52 As String * 8
Dim Dat53 As String * 8
Dim Dat54 As String * 8
Dim Dat55 As String * 8
Dim Dat56 As String * 8
Dim Dat57 As String * 8
Dim Dat58 As String * 8
Dim Dat59 As String * 8
Dim Dat60 As String * 8
Dim Dat61 As String * 8
Dim Dat62 As String * 8
Dim Dat63 As String * 8
Dim Dat64 As String * 8
Dim Dat65 As String * 8
Dim Dat66 As String * 8
Dim Dat67 As String * 8
Dim Dat68 As String * 8
Dim Dat69 As String * 8
Dim Dat70 As String * 8
Dim Dat71 As String * 8
Dim Dat72 As String * 8
Dim Dat73 As String * 8
Dim Dat74 As String * 8

```

Dim Dat75 As String * 8
Dim Dat76 As String * 8
Dim Dat77 As String * 8
Dim Dat78 As String * 8
Dim Dat79 As String * 8
Dim Dat80 As String * 8
Dim Dat81 As String * 8
Dim Dat82 As String * 8
Dim Dat83 As String * 8
Dim Dat84 As String * 8
Dim Dat85 As String * 8
Dim Dat86 As String * 8
Dim Dat87 As String * 8
Dim Dat88 As String * 8
Dim Dat89 As String * 8
Dim Dat90 As String * 8
Dim Dat91 As String * 8
Dim Dat92 As String * 8
Dim Dat93 As String * 8
Dim Dat94 As String * 8
Dim Dat95 As String * 8
Dim Dat96 As String * 8
Main:

```

```

! ***** Ingreso de datos *****
Inputbin Dat1, Dat2, Dat3, Dat4, Dat5, Dat6, Dat7, Dat8, Dat9, Dat10
Inputbin Dat11, Dat12, Dat13, Dat14, Dat15, Dat16, Dat17, Dat18, Dat19, Dat20
Inputbin Dat21, Dat22, Dat23, Dat24, Dat25, Dat26, Dat27, Dat28, Dat29, Dat30
Inputbin Dat31, Dat32, Dat33, Dat34, Dat35, Dat36, Dat37, Dat38, Dat39, Dat40
Inputbin Dat41, Dat42, Dat43, Dat44, Dat45, Dat46, Dat47, Dat48, Dat49, Dat50
Inputbin Dat51, Dat52, Dat53, Dat54, Dat55, Dat56, Dat57, Dat58, Dat59, Dat60
Inputbin Dat61, Dat62, Dat63, Dat64, Dat65, Dat66, Dat67, Dat68, Dat69, Dat70
Inputbin Dat71, Dat72, Dat73, Dat74, Dat75, Dat76, Dat77, Dat78, Dat79, Dat80
Inputbin Dat81, Dat82, Dat83, Dat84, Dat85, Dat86, Dat87, Dat88, Dat89, Dat90
Inputbin Dat91, Dat92, Dat93, Dat94, Dat95, Dat96

! ***** los datos han sido ingresados *****
Print "datos"

!***** GRAFICA DELA MATRIZ *****

Do

! ***** Primera submatriz de 12x8 flip flop 1000 valor = 0 *****

```

Porta = Binval(dat1) : Portb = &B10000000 : Portc = &B00001000

Waitms Temp

Porta = Binval(dat2) : Portb = &B01000000 : Portc = &B00001000

Waitms Temp

Porta = Binval(dat3) : Portb = &B00100000 : Portc = &B00001000

Waitms Temp

Porta = Binval(dat4) : Portb = &B00010000 : Portc = &B00001000

Waitms Temp

Porta = Binval(dat5) : Portb = &B00001000 : Portc = &B00001000

Waitms Temp

Porta = Binval(dat6) : Portb = &B00000100 : Portc = &B00001000

Waitms Temp

Porta = Binval(dat7) : Portb = &B00000010 : Portc = &B00001000

Waitms Temp

Porta = Binval(dat8) : Portb = &B00000001 : Portc = &B00001000

Waitms Temp

Porta = Binval(dat9) : Portd = &B00100000 : Portc = &B00001000

Waitms Temp

Porta = Binval(dat10) : Portd = &B00010000 : Portc = &B00001000

Waitms Temp

Porta = Binval(dat11) : Portd = &B00001000 : Portc = &B00001000

Waitms Temp

Porta = Binval(dat12) : Portd = &B00000100 : Portc = &B00001000

Waitms Temp

' ***** Segunda submatriz de 12x8 flip flop 1001 valor = 1 *****

Porta = Binval(dat13) : Portb = &B10000000 : Portc = &B00001001

Waitms Temp

Porta = Binval(dat14) : Portb = &B01000000 : Portc = &B00001001

Waitms Temp

Porta = Binval(dat15) : Portb = &B00100000 : Portc = &B00001001

Waitms Temp

Porta = Binval(dat16) : Portb = &B00010000 : Portc = &B00001001

Waitms Temp

Porta = Binval(dat17) : Portb = &B00001000 : Portc = &B00001001

Waitms Temp

Porta = Binval(dat18) : Portb = &B00000100 : Portc = &B00001001

Waitms Temp

Porta = Binval(dat19) : Portb = &B00000010 : Portc = &B00001001

Waitms Temp

Porta = Binval(dat20) : Portb = &B00000001 : Portc = &B00001001

Waitms Temp

Porta = Bival(dat21) : Portd = &B00100000 : Portc = &B00001001

Waitms Temp

Porta = Bival(dat22) : Portd = &B00010000 : Portc = &B00001001

Waitms Temp

Porta = Bival(dat23) : Portd = &B00001000 : Portc = &B00001001

Waitms Temp

Porta = Bival(dat24) : Portd = &B00000100 : Portc = &B00001001

Waitms Temp

' ***** Tercera submatriz de 12x8 flip flop 1010 valor = 2 *****

Porta = Bival(dat25) : Portb = &B10000000 : Portc = &B00001010

Waitms Temp

Porta = Bival(dat26) : Portb = &B01000000 : Portc = &B00001010

Waitms Temp

Porta = Bival(dat27) : Portb = &B00100000 : Portc = &B00001010

Waitms Temp

Porta = Bival(dat28) : Portb = &B00010000 : Portc = &B00001010

Waitms Temp

Porta = Bival(dat29) : Portb = &B00001000 : Portc = &B00001010

Waitms Temp

Porta = Bival(dat30) : Portb = &B00000100 : Portc = &B00001010

Waitms Temp

Porta = Bival(dat31) : Portb = &B00000010 : Portc = &B00001010

Waitms Temp

Porta = Bival(dat32) : Portb = &B00000001 : Portc = &B00001010

Waitms Temp

Porta = Bival(dat33) : Portd = &B00100000 : Portc = &B00001010

Waitms Temp

Porta = Bival(dat34) : Portd = &B00010000 : Portc = &B00001010

Waitms Temp

Porta = Bival(dat35) : Portd = &B00001000 : Portc = &B00001010

Waitms Temp

Porta = Bival(dat36) : Portd = &B00000100 : Portc = &B00001010

Waitms Temp

' ***** Cuarta submatriz de 12x8 flip flop 1011 valor = 3 *****

Porta = Bival(dat37) : Portb = &B10000000 : Portc = &B00001011

Waitms Temp

Porta = Bival(dat38) : Portb = &B01000000 : Portc = &B00001011

Waitms Temp

Porta = Bival(dat39) : Portb = &B00100000 : Portc = &B00001011

Waitms Temp

Porta = Binval(dat40) : Portb = &B00010000 : Portc = &B00001011
 Waitms Temp
 Porta = Binval(dat41) : Portb = &B00001000 : Portc = &B00001011
 Waitms Temp
 Porta = Binval(dat42) : Portb = &B00000100 : Portc = &B00001011
 Waitms Temp
 Porta = Binval(dat43) : Portb = &B00000010 : Portc = &B00001011
 Waitms Temp
 Porta = Binval(dat44) : Portb = &B00000001 : Portc = &B00001011
 Waitms Temp
 Porta = Binval(dat45) : Portd = &B00100000 : Portc = &B00001011
 Waitms Temp
 Porta = Binval(dat46) : Portd = &B00010000 : Portc = &B00001011
 Waitms Temp
 Porta = Binval(dat47) : Portd = &B00001000 : Portc = &B00001011
 Waitms Temp
 Porta = Binval(dat48) : Portd = &B00000100 : Portc = &B00001011
 Waitms Temp

' ***** Quinta submatriz de 12x8 flip flop 10001000 valor = 4 *****'

Porta = Binval(dat49) : Portb = &B10000000 : Portc = &B10001000
 Waitms Temp
 Porta = Binval(dat50) : Portb = &B01000000 : Portc = &B10001000
 Waitms Temp
 Porta = Binval(dat51) : Portb = &B00100000 : Portc = &B10001000
 Waitms Temp
 Porta = Binval(dat52) : Portb = &B00010000 : Portc = &B10001000
 Waitms Temp
 Porta = Binval(dat53) : Portb = &B00001000 : Portc = &B10001000
 Waitms Temp
 Porta = Binval(dat54) : Portb = &B00000100 : Portc = &B10001000
 Waitms Temp
 Porta = Binval(dat55) : Portb = &B00000010 : Portc = &B10001000
 Waitms Temp
 Porta = Binval(dat56) : Portb = &B00000001 : Portc = &B10001000
 Waitms Temp
 Porta = Binval(dat57) : Portd = &B00100000 : Portc = &B10001000
 Waitms Temp
 Porta = Binval(dat58) : Portd = &B00010000 : Portc = &B10001000
 Waitms Temp
 Porta = Binval(dat59) : Portd = &B00001000 : Portc = &B10001000
 Waitms Temp

Porta = Binval(dat60) : Portd = &B00000100 : Portc = &B10001000
 Waitms Temp

' ***** Sexta submatriz de 12x8 flip flop 10001001 valor = 5 *****

Porta = Binval(dat61) : Portb = &B10000000 : Portc = &B10001001
 Waitms Temp

Porta = Binval(dat62) : Portb = &B01000000 : Portc = &B10001001
 Waitms Temp

Porta = Binval(dat63) : Portb = &B00100000 : Portc = &B10001001
 Waitms Temp

Porta = Binval(dat64) : Portb = &B00010000 : Portc = &B10001001
 Waitms Temp

Porta = Binval(dat65) : Portb = &B00001000 : Portc = &B10001001
 Waitms Temp

Porta = Binval(dat66) : Portb = &B00000100 : Portc = &B10001001
 Waitms Temp

Porta = Binval(dat67) : Portb = &B00000010 : Portc = &B10001001
 Waitms Temp

Porta = Binval(dat68) : Portb = &B00000001 : Portc = &B10001001
 Waitms Temp

Porta = Binval(dat69) : Portd = &B00100000 : Portc = &B10001001
 Waitms Temp

Porta = Binval(dat70) : Portd = &B00010000 : Portc = &B10001001
 Waitms Temp

Porta = Binval(dat71) : Portd = &B00001000 : Portc = &B10001001
 Waitms Temp

Porta = Binval(dat72) : Portd = &B00000100 : Portc = &B10001001
 Waitms Temp

' ***** Septima submatriz de 12x8 flip flop 10001010 valor = 6 *****

Porta = Binval(dat73) : Portb = &B10000000 : Portc = &B10001010
 Waitms Temp

Porta = Binval(dat74) : Portb = &B01000000 : Portc = &B10001010
 Waitms Temp

Porta = Binval(dat75) : Portb = &B00100000 : Portc = &B10001010
 Waitms Temp

Porta = Binval(dat76) : Portb = &B00010000 : Portc = &B10001010
 Waitms Temp

Porta = Binval(dat77) : Portb = &B00001000 : Portc = &B10001010
 Waitms Temp

Porta = Binval(dat78) : Portb = &B00000100 : Portc = &B10001010
 Waitms Temp

Porta = Binval(dat79) : Portb = &B00000010 : Portc = &B10001010

Waitms Temp

Porta = Binval(dat80) : Portb = &B00000001 : Portc = &B10001010

Waitms Temp

Porta = Binval(dat81) : Portd = &B00100000 : Portc = &B10001010

Waitms Temp

Porta = Binval(dat82) : Portd = &B00010000 : Portc = &B10001010

Waitms Temp

Porta = Binval(dat83) : Portd = &B00001000 : Portc = &B10001010

Waitms Temp

Porta = Binval(dat84) : Portd = &B00000100 : Portc = &B10001010

Waitms Temp

' ***** Octava submatriz de 12x8 flip flop 10001010 valor = 7 *****

Porta = Binval(dat85) : Portb = &B10000000 : Portc = &B10001011

Waitms Temp

Porta = Binval(dat86) : Portb = &B01000000 : Portc = &B10001011

Waitms Temp

Porta = Binval(dat87) : Portb = &B00100000 : Portc = &B10001011

Waitms Temp

Porta = Binval(dat88) : Portb = &B00010000 : Portc = &B10001011

Waitms Temp

Porta = Binval(dat89) : Portb = &B00001000 : Portc = &B10001011

Waitms Temp

Porta = Binval(dat90) : Portb = &B00000100 : Portc = &B10001011

Waitms Temp

Porta = Binval(dat91) : Portb = &B00000010 : Portc = &B10001011

Waitms Temp

Porta = Binval(dat92) : Portb = &B00000001 : Portc = &B10001011

Waitms Temp

Porta = Binval(dat93) : Portd = &B00100000 : Portc = &B10001011

Waitms Temp

Porta = Binval(dat94) : Portd = &B00010000 : Portc = &B10001011

Waitms Temp

Porta = Binval(dat95) : Portd = &B00001000 : Portc = &B10001011

Waitms Temp

Porta = Binval(dat96) : Portd = &B00000100 : Portc = &B10001011

Waitms Temp

Loop

GoTo Main

End

ANEXO 2

SOFTWARE DEL MICROCONTROLADOR PARA MATRIZ TACTIL

```
$regfile = "m16def.dat"
$crystal = 1000000
$baud = 4800

' *** configurando puerto de salida
Config Porta = Output

' *** nombrando pines de salida
Datos Alias Porta.1
Reloj Alias Porta.2
Latches Alias Porta.0

' *** variables
Dim L(8) As Byte
Dim B(8) As Byte
Dim Dat1 As String * 8
Dim Dat2 As String * 8
Dim Dat3 As String * 8
Dim Dat4 As String * 8
Dim Dat5 As String * 8
Dim Dat6 As String * 8
Dim Dat7 As String * 8
Dim Dat8 As String * 8
Dim A As Byte
Dim C As Byte

' *** Inicio del programa

Inicio:

Waitms 100
' *** envia al PC listo para iniciar

Print "iniciar"

' *** todos los LATCH en cero

B(1) = &B00000000
B(2) = &B00000000
B(3) = &B00000000
B(4) = &B00000000
B(5) = &B00000000
```

B(6) = &B00000000

B(7) = &B00000000

B(8) = &B00000000

For C = 1 To 8

' *** Se inicia comunicacion serial

Shiftout Datos , Reloj , B(c) , Option [, 0 , 9]

Pulseout Latchs, 0, 1

Next C

Wait 1

' ***** Ingreso de datos del PC *****

Inputbin Dat1, Dat2, Dat3, Dat4, Dat5, Dat6, Dat7, Dat8

L(1) = Binval(Dat1)

L(2) = Binval(Dat2)

L(3) = Binval(Dat3)

L(4) = Binval(Dat4)

L(5) = Binval(Dat5)

L(6) = Binval(Dat6)

L(7) = Binval(Dat7)

L(8) = Binval(Dat8)

For A = 1 To 8

' *** Paso de datos a los LATCH

Shiftout Datos , Reloj , L(a) , Option [, 0 , 9]

Pulseout Latchs, 0, 1

Waitms 10

Next A

' *** se envia al PC fin

Print "fin"

End