

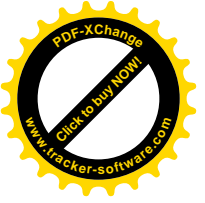
SISTEMA DE INFORMACIÓN GEOMÉTRICA DINÁMICA PARA USO POTENCIAL EN PEDAGOGÍA DE INVIDENTES



CARLOS MIGUEL AVENDAÑO LÓPEZ
JULIÁN DAVID JAIMES GÓMEZ



UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FÍSICO-MECÁNICAS
ESCUELA DE INGENIERÍAS ELÉCTRICA, ELECTRÓNICA Y DE
TELECOMUNICACIONES
BUCARAMANGA 2013



**SISTEMA DE INFORMACIÓN GEOMÉTRICA DINÁMICA PARA USO
POTENCIAL EN PEDAGOGÍA DE INIDENTES**

CARLOS MIGUEL AVENDAÑO LÓPEZ

JULIÁN DAVID JAIMES GÓMEZ

**Trabajo de grado presentado como requisito parcial para optar al título de
ingeniero electrónico**

Director

RICARDO ALZATE CASTAÑO

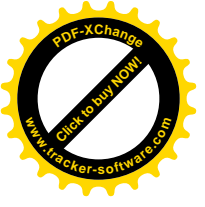
Ingeniero Electrónico, PhD

Codirector

MARTIN EDUARDO ACOSTA

Licenciado en Matemáticas, PhD

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FÍSICO-MECÁNICAS
ESCUELA DE INGENIERÍAS ELÉCTRICA, ELECTRÓNICA Y DE
TELECOMUNICACIONES
BUCARAMANGA 2013**



AGRADECIMIENTOS

En el presente trabajo de tesis de grado primeramente me gustaría agradecerle a ti Dios por bendecirme para llegar hasta donde he llegado, porque hiciste realidad este sueño anhelado.

A mis padres, Libardo Jaimes Castro y María Esther Gómez Cala, sin su ayuda incondicional y sus sabios consejos no podría dar este paso tan grande para mi futuro.

A la UNIVERSIDAD INDUSTRIAL DE SANTANDER por darme la oportunidad de estudiar y ser un profesional.

A mi director de tesis, Dr. Ricardo Alzate Castaño por su esfuerzo y dedicación, quien con sus conocimientos, su experiencia, su paciencia y su motivación ha logrado en mí que pueda terminar mis estudios con éxito.

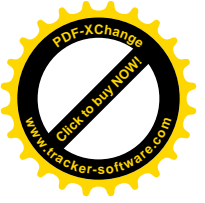
Julián David Jaimes Gómez

Quisiera agradecer como primera instancia a mis padres, Carlos A. Avendaño y Martha Mireya López, por su apoyo incondicional, sus consejos y los grandes esfuerzos que han hecho para que yo logre este anhelado sueño.

A la universidad con todo su equipo de profesores por ayudarme en mi formación profesional. A mi director de proyecto Dr. Ricardo Alzate por haberme guiado en la elaboración de este documento, por su inmensa paciencia y comprensión.

Además quisiera agradecerle a todos mis compañeros por los grandes momentos que han compartido conmigo durante estos años.

Carlos Miguel Avendaño López



DEDICATORIA

Dedico este trabajo principalmente a Dios, por haberme dado la vida y permitirme haber llegado hasta este momento tan importante de mi formación profesional. A mis padres, por su apoyo, consejos, comprensión, amor, ayuda en los momentos difíciles y por ayudarme con los recursos necesarios para poder obtener este título. Me han dado todo lo que soy como persona mis valores, mis principios, mi carácter, mi empeño, mi perseverancia, mi coraje para conseguir mis objetivos.

Julián David Jaimes Gómez

Quisiera dedicar este trabajo principalmente a mis padres como muestra de agradecimiento por todo el esfuerzo que han hecho para permitirme lograr este título. Además quisiera dedicar este logro a mi hermana Silvia Katherine Avendaño López, a quien le deseo lo mejor en la vida y los mejores éxitos en la carrera que está iniciando.

Carlos Miguel Avendaño López

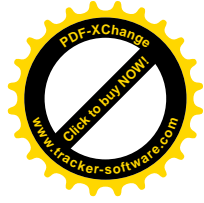
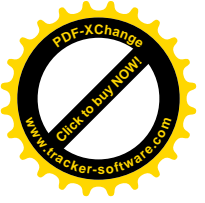
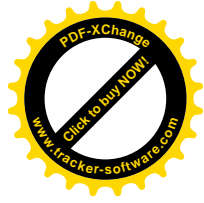
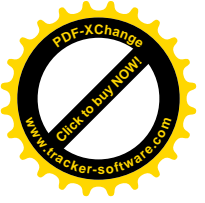


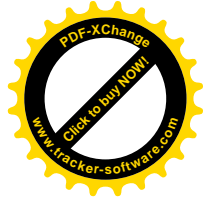
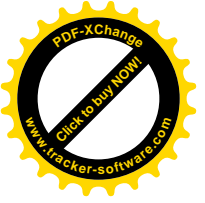
TABLA DE CONTENIDO

	pág
INTRODUCCION	17
CAPITULO 1.....	19
1.1. ANTECEDENTES Y JUSTIFICACIÓN:.....	19
1.2. PLANTEAMIENTO Y DEFINICIÓN DEL PROBLEMA:.....	21
1.3. OBJETIVOS:.....	22
1.3.1. OBJETIVO GENERAL.....	22
1.3.2. OBJETIVOS ESPECÍFICOS	23
CAPITULO 2.....	24
2.1. INTRODUCCION A LA TIFLOTECNOLOGIA	24
2.2. DISEÑO Y CONSTRUCCION DE UN SISTEMA EMULADOR DE ESTIMULACIÓN TÁCTIL DE FORMAS GEOMÉTRICAS.....	27
2.3. DISEÑO DE HARDWARE.....	28
2.3.1. BLOQUE DE CONTROL.....	29
2.3.2. ETAPA DE VISUALIZACIÓN	33
CAPITULO 3.....	40
3.1 DISEÑO DE LAS RUTINAS DE SOFTWARE.....	40
3.1.1. MODIFICACIÓN DEL CÓDIGO FUENTE DE CARMETAL.....	41
3.1.2. RUTINA DE PROGRAMACIÓN DEL MICROCONTROLADOR.....	46
CAPITULO 4.....	54
4.1. IMPLEMENTACION DEL SISTEMA	54
4.2. RESULTADOS DE LAS PRUEBAS	57
4.2.1. CÁLCULO DEL ERROR PARA LAS IMÁGENES ENVIADAS A LA MATRIZ.....	57
4.2.2. CÁLCULO DEL ERROR PARA IMÁGENES DEL DOBLE DE RESOLUCIÓN, MANTENIENDO LA RELACIÓN DE ASPECTO.....	67
5. CONCLUSIONES	70
6. OBSERVACIONES Y TRABAJO FUTURO	72
CITAS BIBLIOGRAFICAS	75
BIBLIOGRAFIA	79
ANEXOS.....	80



ÍNDICE DE TABLAS

	pág
Tabla 1. Principales características del MCU MC9S08JM60.....	30
Tabla 2. Características eléctricas de la matriz de LEDs LTP-2558AA	34
Tabla 3. Datos de la figura triángulo.	60
Tabla 4. Datos de la figura círculo.	62
Tabla 5. Datos de la figura cuadrado.....	64
Tabla 6. Datos de la figura casa.	66
Tabla 7. Datos de la figura cuadrado durante la prueba 2.....	68
Tabla 8. Datos de la figura casa durante la prueba 2.	69



INDICE DE FIGURAS

	pág
Figura 1. Nota en braille.....	24
Figura 2. Brailenote, dispositivo conversor a braille.....	25
Figura 3. Modelo general del proyecto, Blindisp.....	28
Figura 4. Etapas principales en el diseño y la construcción del hardware.....	29
Figura 5. Modelo de la PCB del MCU.....	31
Figura 6. Circuito utilizado para alimentar el MCU.....	32
Figura 7. Etapas del diseño de la PCB de la etapa de visualización.....	33
Figura 8. División de la etapa de control.....	35
Figura 9. División de la etapa de potencia.....	36
Figura 10. Conexión de los componentes de hardware en la etapa de visualización.....	38
Figura 11. Módulos que se ejecutaron durante la elaboración de las rutinas de software.....	40
Figura 12. Diagrama de bloques de la rutina de software.....	41
Figura 13. Modificación del código de CarMetal.....	43
Figura 14. Diagrama de bloques del módulo USB del MC9S08JM60.....	48
Figura 15. Transferencia de estados del dispositivo USB.....	50
Figura 16. Programadora.....	54
Figura 17. Tarjeta del Microcontrolador.....	55
Figura 18. Cara frontal de la matriz de LEDs.....	56
Figura 19. Cara posterior de la matriz de LEDs.....	56
Figura 20. Triangulo en el área de trabajo de CaRMetal.....	59
Figura 21. Imagen reducida del triángulo.....	59
Figura 22. Fotografía de la matriz de LEDs, figura triángulo.....	60
Figura 23. Círculo en el área de trabajo de CaRMetal.....	61
Figura 24. Imagen reducida del círculo.....	61
Figura 25. Fotografía de la matriz de LEDs, figura círculo.....	62
Figura 26. Cuadrado en el área de trabajo de CaRMetal.....	63
Figura 27. Imagen reducida y binarizada del cuadrado.....	63
Figura 28. Fotografía de la matriz de LEDs, figura cuadrado.....	64
Figura 29. Casa en el área de trabajo de CaRMetal.....	65
Figura 30. Imagen reducida de la casa.....	65
Figura 31. Fotografía de la matriz de LEDs, figura casa.....	66
Figura 32. Cuadrado en el área de trabajo de CaRMetal para la prueba 2.....	67

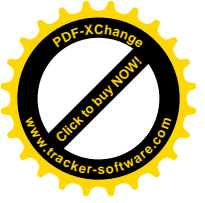
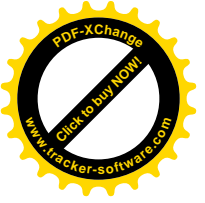
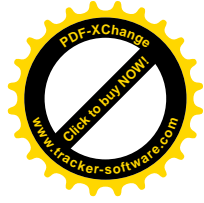
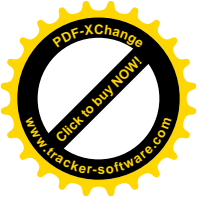


Figura 33. Imagen reducida del cuadrado para la prueba 2..... 68
Figura 34. Casa en el área de trabajo de CaRMetal para la prueba 2..... 69
Figura 35. Imagen reducida de la casa para la prueba 2..... 69



LISTA DE ANEXOS

	pág
ANEXO A.....	80
A.1. PCB del microcontrolador MC9S08JM60 en Eagle.....	80
A.2. Esquemático de la PCB del microcontrolador en Eagle.....	80
A.3. Distribución de pines del MC9S08JM60.....	81
ANEXO B.....	82
B.1. Instalación driver LibUSB.....	82
B.2. Descripción del protocolo USB.....	83



RESUMEN

Título: SISTEMA DE INFORMACIÓN GEOMÉTRICA PARA USO POTENCIAL EN PEDAGOGÍA DE INVIDENTES¹

Autores: CARLOS MIGUEL AVENDAÑO LÓPEZ, JULIÁN DAVID JAIMES GOMEZ²

Palabras Clave:

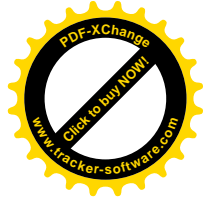
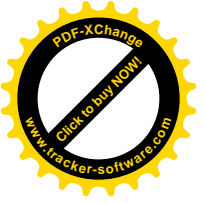
Microcontrolador, JAVA, Protocolo USB, Matriz LEDs.

Descripción:

Este proyecto de grado muestra la construcción e implementación de un sistema dinámico para la obtención de la imagen del área de trabajo de un software para uso pedagógico en geometría, para esto se utiliza un programa de software libre llamado CaRMetal, se presenta el modo como se construyó el sistema partiendo desde una idea global y dividiéndolo en subsistemas para una fácil interpretación; al igual, se muestra características principales del microcontrolador utilizado, los pasos a seguir para establecer la comunicación USB y la modificación hecha al software con el fin de que el sistema tenga un correcto funcionamiento; a su vez se muestra el procedimiento que se debe seguir para instalar el controlador, el cual es el encargado de llevar a cabo la comunicación entre el lenguaje JAVA y el sistema operativo. Como resultado se evidencia el montaje del sistema y una prueba realizada al mismo con el fin de corroborar su desempeño, de allí se puede observar que al mejorar la resolución de la matriz se reduce el error y así se formarían imágenes con mayor detalle. Como trabajo futuro se propone implementar el sistema con los diferentes tipos de tecnologías actuales que permitan convertir las señales eléctricas en estímulos táctiles, con el fin de enfocar el proyecto en la educación y el apoyo a las personas invidentes.

¹ Trabajo de grado

² Facultad de ingenierías Físico-mecánicas. Escuela de Ingenierías Eléctrica, Electrónica y de Telecomunicaciones. Director: Ricardo Alzate Castaño. Codirector: Martín Eduardo Acosta.



SUMMARY

Title: VISUAL INFORMATION SYSTEM FOR POTENTIAL USE IN PEDAGOGY OF GEOMETRY FOR BLIND PEOPLE³

AUTHORS: CARLOS MIGUEL AVENDAÑO LÓPEZ, JULIÁN DAVID JAIMES GOMEZ⁴

Keywords:

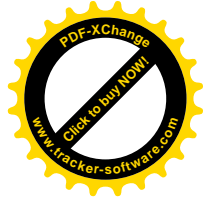
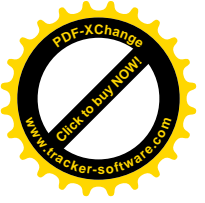
Microcontroller, JAVA, USB protocol, LED matrix.

Description:

This Project shows the construction and implementation of a dynamic system which needs the image of the workspace in software for educational purposes in geometry: CaRMetal. This software has been chosen because it is open source software that is usually used as a teaching tool. We present how the system has been built, beginning from a global system, then dividing it into little subsystems, to create an easy comprehension of the topic. Also, we indicated some main features of the microcontroller, steps we have followed to create the USB communication and how to modify CaRMetal in order to do what we need. In the same way we show how you install the driver which has the purpose of creating communication between JAVA programming language and operating system. As a result we exhibit the system assembly and a function test. One important conclusion is that the quality of the graphics which are being showed in the LED Matrix depends on the resolution of the display, therefore if it grows, the difference between the original image and its corresponding representation in the LED matrix, decreases. As a future work we propose to implement this display of LED matrix with other existing technologies which turns electric signals into tactile stimulation in order to focus this project on helping blind people.

³ Degree work.

⁴ Physico-mechanical Engineering Faculty, School of Electrical Engineering. Supervisor: Ricardo Alzate Castaño. Co-supervisor: Martín Eduardo Acosta.



GLOSARIO

BufferedImage: Es un objeto de tipo imagen que está alojada en memoria RAM y consta de dos parte el modelo de color y el mapeo de la imagen.

AffineTransform: Representación del mapeo lineal de una imagen en 2D, permite transformaciones.

GetScaleInstance: método el cual realiza una transformación de escala a una imagen.

WritableRaster: Clase que permite crear el mapeo de una imagen cuyos pixeles pueden ser modificados.

GetSample: método para obtener el valor de un pixel.

SetSample: método que permite darle un valor específico a un pixel de una imagen.

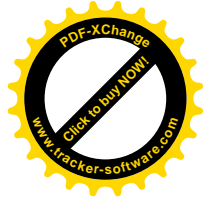
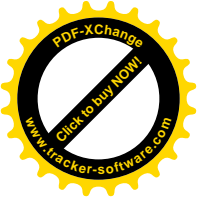
ToByteArray: método que permite convertir una matriz de bytes a un vector de bytes.

System.arraycopy: método que permite copiar un vector a otro, así como permite recortarlo.

WriteBulk: método que permite crear un envío a puntos terminales (endpoints), que son de tipo bulk.

Timer: permite crear un timer dentro del código, es decir un bucle que se ejecuta constantemente con un lapso específico.

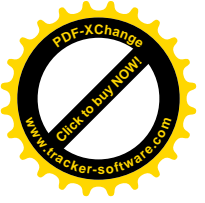
ExportGraphicBuffer: obtiene la imagen en del área de trabajo y la almacena en memoria RAM.



INTRODUCCION

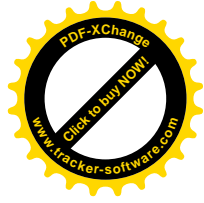
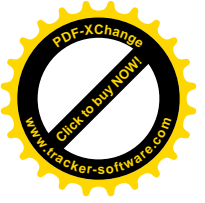
En el mundo actual los dispositivos electrónicos forman cada vez más, una parte activa de nuestra vida y un conocimiento en el manejo de estas tecnologías es un factor determinante para el desarrollo personal, social y laboral de un individuo. La mayor parte de estos dispositivos son diseñados para que tengan una buena aceptación en la mayoría de las personas, y para esto presentan un gran contenido visual e interfaces fáciles de usar para un individuo promedio. Pero para una persona con discapacidad la mayoría de estos dispositivos, en lugar de ser herramientas útiles que faciliten la vida, se convierten en otra causa de segregación. Esta situación va en contra de la concepción misma de la tecnología, debido a que ésta es pensada como el conjunto de conocimientos y técnicas que facilitan la adaptación al medio ambiente y satisfacen tanto los deseos como las necesidades de la humanidad. Por ende, se supone que las nuevas tecnologías deben ayudar a mitigar la exclusión social a la que se pueden ver sometidas regularmente las personas con necesidades especiales.

Según cifras del DANE (año 2005) en Colombia el 0.8% de la población posee alguna limitación visual, de ellos menos del 0.01% presenta instrucción académica [1]. Para el caso específico de Santander se cuentan casi 19 mil personas con limitación visual de las cuales el 12.8% se encuentran en la ciudad de Bucaramanga [2]. Análogamente, según la ley 361 de febrero de 1997 [3]: “El Estado Colombiano en sus instituciones de Educación Pública garantizará el acceso a la educación y a la capacitación en los niveles primario, secundario, profesional y técnico para las personas con limitación, quienes para ello dispondrán de una formación integral dentro del ambiente más apropiado a sus necesidades especiales”. De ahí que deban existir programas gubernamentales de apoyo para garantizar el acceso de personas discapacitadas a los servicios de educación estatales.



El desarrollo y financiamiento de estos dispositivos ha sido impulsado por diversas organizaciones, corporaciones y asociaciones que se han creado en todo el mundo y que buscan mejorar la calidad de vida de las personas con deficiencias visuales. Entre estas organizaciones encontramos a *ONCE* (Organización Nacional de Ciegos de España), la cual es una de las más desarrolladas del mundo [4], la *BPA* (*Blind's People Association*), la cual es una organización India, la *AFB* (*American Foundation for the Blind*) y *ULAC* (Unión Latinoamericana de Ciegos). Estas organizaciones buscan mejorar las oportunidades de trabajo o de educación de las personas ciegas; para esto se apoyan en la tiflotecnología, por lo cual son los principales precursores de estas tecnologías [4]. En países como España, la *ONCE* se encarga de gestionar y vigilar la entrega del material tiflotecnológico a los estudiantes y trabajadores con deficiencias visuales [21].

Además de esto, estas organizaciones son las mayores inversionistas en la investigación y el desarrollo de dispositivos que buscan ser usados en la tiflotecnología, pero siempre se han encontrado con una limitante, la cual es el material de fabricación de estos dispositivos. Estos materiales se caracterizan por tener un alto costo de adquisición y poca miniaturización; para mitigar esta limitante se están haciendo estudios y pruebas con nuevos materiales y tecnologías, entre las cuales encontramos *SAW* (*Surface Acoustic Wave*) [12,13], que utiliza vibraciones para posicionar objetos, *SMA* (*Shape Memory Alloy*) [13,14], metales inteligentes que pueden cambiar su forma con solo agregarles calor, canales de succión, los cuales hacen el análogo a los solenoides en vez de mostrar un relieve se succiona la piel permitiendo esbozar graficas simples, y dispositivos piezo-eléctricos[16], elementos que con el paso de corriente eléctrica cambian su forma.



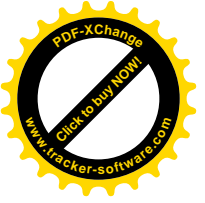
CAPITULO 1

1.1. ANTECEDENTES Y JUSTIFICACIÓN:

En las sociedades subdesarrolladas, los individuos con limitaciones físicas enfrentan serios problemas para desarrollar con normalidad sus actividades cotidianas, ocasionando en algunos casos discriminación y exclusión social. Dicha exclusión no es sólo a nivel de oportunidades laborales o de interacción social, sino también de acceso a requerimientos fundamentales como la educación, que por obvias razones requiere de una infraestructura no-conventional para este tipo de situaciones particulares.

Según cifras del DANE (año 2005) en Colombia el 0.8% de la población posee alguna limitación visual, de ellos menos del 0.01% presenta instrucción académica [1]. Para el caso específico de Santander se cuentan casi 19 mil personas con limitación visual de las cuales el 12.8% se encuentran en la ciudad de Bucaramanga [2]. Análogamente, según la ley 361 de febrero de 1997 [3]: “El Estado Colombiano en sus instituciones de Educación Pública garantizará el acceso a la educación y a la capacitación en los niveles primario, secundario, profesional y técnico para las personas con limitación, quienes para ello dispondrán de una formación integral dentro del ambiente más apropiado a sus necesidades especiales”. De ahí que deban existir programas gubernamentales de apoyo para garantizar el acceso de personas discapacitadas a los servicios de educación estatales.

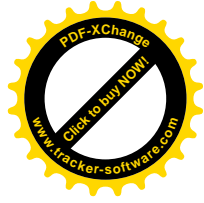
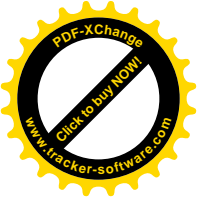
Dependiendo del tipo y grado de discapacidad que posea el individuo se utilizan diversas técnicas pedagógicas (e.g. sistema Braille para invidentes y lenguaje de señas para sordomudos). Estas discapacidades pueden deberse a factores congénitos o ser consecuencia de graves accidentes. Actualmente se desarrollan un gran número de investigaciones cuyo objetivo primordial es



mejorar la calidad de vida de los afectados por discapacidades, a partir de la creación y el uso de herramientas tecnológicas. Un ejemplo importante para esta tendencia es la llamada Tiflotecnología o tecnología de apoyo para personas invidentes [4,5], a través de la cual se han desarrollado equipos como: lectores de pantalla (que muestran información visual a través de un sintetizador de voz), magnificadores de pantalla (aumentando el tamaño de los caracteres mostrados en el computador), conversores Braille, y reconocedores de texto OCR (reconocimiento óptico de caracteres) entre otros.

Otra importante tendencia para ayudar al invidente a superar sus limitaciones es mediante la estimulación del sentido del tacto. Esto es posible debido a ciertas similitudes que existen entre los sistemas sensoriales asociados al tacto y la visión [6]. Aunque no se tenga entera certeza que con el sentido del tacto pueden percibirse todos los efectos del sistema visual, se considera factible que el cerebro esté en capacidad de procesar imágenes táctiles. Por tanto, desde el punto de vista tecnológico un problema aún por resolver corresponde a la creación de dispositivos o pantallas que estimulen el sentido del tacto, para ser posteriormente utilizados como herramientas de formación (enseñanza) para personas invidentes o con baja visión. El principio de funcionamiento para estos dispositivos ha sido una matriz de puntos (o pines) que se activan o desactivan dependiendo de la información visual correspondiente en la imagen [7,8]. Cada punto (o pin) se denomina *tactel* dado el contexto de la aplicación, y equivale a un píxel de información visual [7]. El principal limitante para este tipo de desarrollos tecnológicos, han sido las características del material empleado para transmitir los impulsos táctiles representando inconvenientes como alto costo, elevado consumo energético, poca fuerza de pines y dificultad de miniaturización [7,8].

Si bien son pocos los dispositivos táctiles que se han producido para distribución comercial, un número importante de investigaciones se realizan para desarrollar nuevos materiales que permitan disminuir costos y aumentar



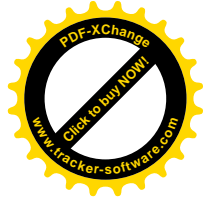
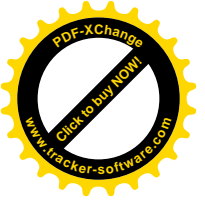
rendimientos en dispositivos de estimulación táctil. Algunos adelantos al respecto permiten enumerar tecnologías como la *SAW* (*Surface Acoustic Wave*) que utiliza vibraciones para posicionar objetos [9, 10, 11]; también existen trabajos en tecnologías alternativas como la *SMA* (*Shape Memory Array*) [12, 13, 14], canales de succión [15] y dispositivos piezo-eléctricos [16].

A nivel nacional solo se han encontrado dos referencias acerca del tema y son de la Universidad Javeriana [17,18] donde se realiza el desarrollo de interfaces hardware para transformar imágenes en estímulos táctiles. En la Universidad Industrial de Santander no se han encontrado referencias de desarrollos relacionados con herramientas tecnológicas de apoyo a invidentes.

1.2. PLANTEAMIENTO Y DEFINICIÓN DEL PROBLEMA:

Pantallas táctiles disponibles en el mercado incluyen la METEC-DMD1260 de 159x59 *tacteles* (basada en solenoides) que cuesta alrededor de \$70000 USD [7, 8, 19]. Debido al alto costo para un dispositivo táctil de alta resolución se han creado diferentes técnicas que explotan mejor las capacidades de las pantallas táctiles de baja resolución, por ejemplo a partir de celdas Braille de 6x2 *tacteles* con costos individuales de \$40 USD [19]. Otra forma de aprovechar pantallas de baja resolución es a partir de pantallas virtuales en las cuales un computador reconstruye una imagen mediante movimientos progresivos ejecutados por un usuario. El sistema VirtTouch [7] implementa esta estrategia y tiene un costo cercano a los \$10000 USD. A pesar de su utilidad, estos sistemas carecen de información táctil (roce de la piel con la superficie del dispositivo).

A partir de lo anterior, se concluye que las alternativas tecnológicas comerciales para realizar estimulación táctil en invidentes, no son asequibles para



ciudadanos de escasos recursos en economías latinoamericanas. También se nota una carencia de sistemas adaptados para compatibilidad con herramientas pedagógicas estándar en enseñanza de asignaturas de base como la geometría, entre las cuales se destacan aplicaciones de geometría dinámica como *CaRMetal*, *GeoGebra* o *CABRI*¹. Por tanto, alrededor de esta temática es válido plantearse inquietudes de investigación como las siguientes: ¿Cómo adaptar herramientas actuales para enseñanza de la geometría básica en modo tal que puedan emplearse con personas invidentes? ¿Qué tecnología de bajo costo puede emplearse para realizar estimulación táctil con fines pedagógicos en Colombia? ¿Cuáles son las características técnicas mínimas que debe tener una pantalla táctil para enseñanza de la geometría básica? ¿Qué tan viable y útil es implementar una pantalla táctil de bajo costo?

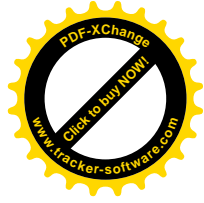
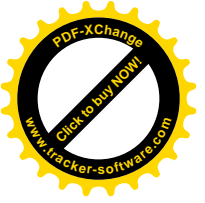
El presente proyecto de grado busca realizar aportes direccionados a la resolución de estas inquietudes, constituyendo una base para desarrollos posteriores que permitan afianzar el compromiso social de la Universidad Industrial de Santander con las comunidades vulnerables.

1.3. OBJETIVOS:

1.3.1. OBJETIVO GENERAL

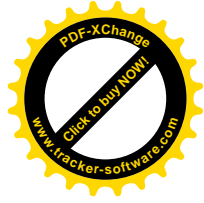
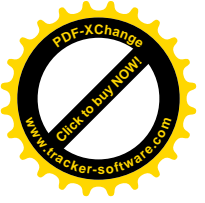
- Diseño y desarrollo de un sistema de extracción de información geométrica dinámica para uso potencial en pedagogía de invidentes mediante estimulación táctil.

¹ Que permiten realizar construcciones geométricas a partir de puntos, vectores, segmentos, rectas, secciones cónicas o funciones de las mismas, modificadas a posteriori de manera dinámica e interactiva.



1.3.2. OBJETIVOS ESPECÍFICOS

- Desarrollar una interfaz que procese y convierta la información gráfica de un PC en señales eléctricas para ser enviadas a través de un puerto USB. La información a tratar se extraerá desde la memoria RAM del PC mediante herramientas de programación de bajo nivel y luego se procesará para reducir su resolución y obtener su equivalente en escala de grises con la ayuda de una herramienta de software de más alto nivel.
- Implementar la interfaz en un *display* de LEDs de baja resolución (alrededor de 20x35 pixeles) y con un tamaño alrededor del de una hoja tamaño carta (21x28 cm).
- Desarrollar un procedimiento de pruebas para verificar la validez técnica del sistema implementado.
- Acondicionar la interfaz diseñada a herramientas de software para la enseñanza de geometría.
- Seleccionar una tecnología que permita el estímulo del tacto considerando las existentes tales como: *SMA (Shape Memory Alloy)*, *SAW (Surface Acoustic Wave)*, *MEMS (Micro-Electro-Mechanical Systems)*, canales de succión, estimulación eléctrica, estimulación térmica, actuadores termo-neumáticos, solenoides, entre otros. Teniendo en cuenta aspectos como consumo de potencia, posibilidad de miniaturización, costo del material, peso, vida útil del material y accesibilidad.



CAPITULO 2

2.1. INTRODUCCION A LA TIFLOTECNOLOGIA

Figura 1. Nota en braille



Fuente Instituto Universitario de Posgrado, España.

En el mundo actual los dispositivos electrónicos forman cada vez más, una parte activa de nuestra vida y un conocimiento en el manejo de estas tecnologías es un factor determinante para el desarrollo personal, social y laboral de un individuo. La mayor parte de estos dispositivos son diseñados para que tengan una buena aceptación en la mayoría de las personas, y para esto presentan un gran contenido visual e interfaces fáciles de usar para un individuo promedio. Pero para una persona con discapacidad la mayoría de estos dispositivos, en lugar de ser herramientas útiles que faciliten la vida, se convierten en otra causa de segregación. Esta situación va en contra de la concepción misma de la tecnología, debido a que ésta es pensada como el conjunto de conocimientos y técnicas que facilitan la adaptación al medio ambiente y satisfacen tanto los deseos como las necesidades de la humanidad. Por ende, se supone que las nuevas tecnologías deben ayudar a mitigar la

exclusión social a la que se pueden ver sometidas regularmente las personas con necesidades especiales.

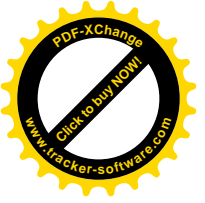
Por lo anterior, hoy en día se observa una tendencia que busca apoyar a estos pequeños grupos; y en todo el mundo se están desarrollando estrategias pedagógicas para instruir a las personas discapacitadas y poner la tecnología al servicio de estos individuos [4, 5, 7, 8, 20], entre estas estrategias encontramos la tiflotecnología, la cual basa su principio en el apoyo tecnológico a personas invidentes o con baja visión, ya sea mediante equipos adaptados o específicos [20].

Con los equipos adaptados, lo que se busca es modificar equipos ya existentes para el uso potencial, como herramientas sonoras o de tacto, permitiendo al invidente hacer uso de los mismos dispositivos que una persona promedio puede utilizar. En estos casos se remplazan los estímulos visuales [4, 5, 7, 8, 20], comúnmente utilizados en la mayoría de dispositivos y se transforman a estímulos anteriormente comentados; entre ellos encontramos los lectores de pantalla, los navegadores de internet parlante, reconocimiento de texto y los conversores a braille, entre otros. Los equipos específicos son diseñados concretamente para personas ciegas o con baja visibilidad, y les permiten gestionar su información; un ejemplo de estos equipos son los anotadores parlantes [4, 20].

Figura 2. Braillenote, dispositivo conversor a braille.

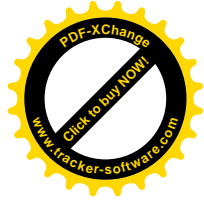
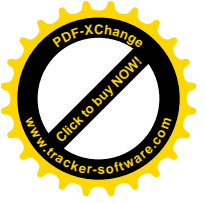


Fuente Ediversidad.



El desarrollo y financiamiento de estos dispositivos ha sido impulsado por diversas organizaciones, corporaciones y asociaciones que se han creado en todo el mundo y que buscan mejorar la calidad de vida de las personas con deficiencias visuales. Entre estas organizaciones encontramos a *ONCE* (Organización Nacional de Ciegos de España), la cual es una de las más desarrolladas del mundo [4], la *BPA* (*Blind's People Association*), la cual es una organización India, la *AFB* (*American Foundation for the Blind*) y *ULAC* (Unión Latinoamericana de Ciegos). Estas organizaciones buscan mejorar las oportunidades de trabajo o de educación de las personas ciegas; para esto se apoyan en la tiflotecnología, por lo cual son los principales precursores de estas tecnologías [4]. En países como España, la *ONCE* se encarga de gestionar y vigilar la entrega del material tiflotecnológico a los estudiantes y trabajadores con deficiencias visuales [21].

Además de esto, estas organizaciones son las mayores inversionistas en la investigación y el desarrollo de dispositivos que buscan ser usados en la tiflotecnología, pero siempre se han encontrado con una limitante, la cual es el material de fabricación de estos dispositivos. Estos materiales se caracterizan por tener un alto costo de adquisición y poca miniaturización; para mitigar esta limitante se están haciendo estudios y pruebas con nuevos materiales y tecnologías, entre las cuales encontramos *SAW* (*Surface Acoustic Wave*) [12,13], que utiliza vibraciones para posicionar objetos, *SMA* (*Shape Memory Alloy*) [13,14], metales inteligentes que pueden cambiar su forma con solo agregarles calor, canales de succión, los cuales hacen el análogo a los solenoides en vez de mostrar un relieve se succiona la piel permitiendo esbozar graficas simples, y dispositivos piezo-eléctricos[16], elementos que con el paso de corriente eléctrica cambian su forma.

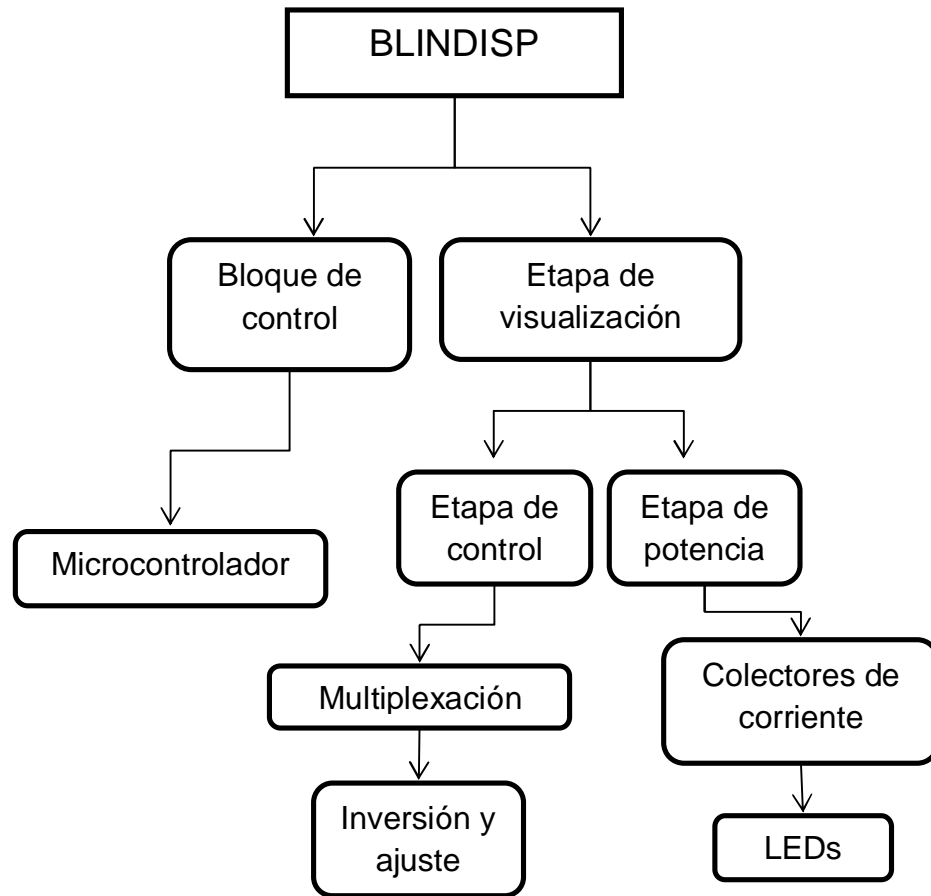


2.2. DISEÑO Y CONSTRUCCION DE UN SISTEMA EMULADOR DE ESTIMULACIÓN TÁCTIL DE FORMAS GEOMÉTRICAS

Con este proyecto se desarrolló un sistema que obtenga la información visual del área de trabajo de un software de geometría dinámica y la envíe a través del puerto USB a un dispositivo visual de baja resolución (32x25 pixeles). Para realizar esto es evidente que se necesitan construir algunos dispositivos de hardware y elaborar ciertas rutinas de software. Los dispositivos de hardware se deben encargar de recibir los datos vía USB, almacenarlos y mostrarlos de forma correcta; mientras que las rutinas de software se encargan de dirigir la comunicación USB entre el PC y el microcontrolador y, a la vez de controlar el envío de los datos del microcontrolador a un dispositivo de visualización o matriz de LEDs.

Primero que todo, para la correcta obtención y visualización de datos es necesario crear un módulo, el cual se pueda conectar al computador y que sea reconocido como un dispositivo USB externo, asimismo se debe implementar un módulo que se encargue de almacenar la información enviada desde el computador y ordenarla como una matriz (ya que la información se transmite de forma serial, los datos se deben organizar para formar una matriz). Para realizar esto se necesitaría un dispositivo con muchos pines y/o puertos de salida, ya que la resolución de la matriz propuesta es de 800 pixeles, por ende el modulo debería tener 800 pines de salida. Para resolver esta problemática y utilizar menos potencia se ha propuesto una multiplexación en el tiempo, para poder utilizar técnica se deben agregar dos módulos al sistema, ahora se requiere un módulo de multiplexación, y un módulo o dispositivo que convierta las señales eléctricas a señales visuales.

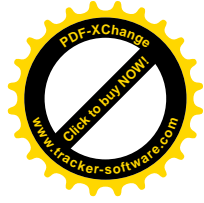
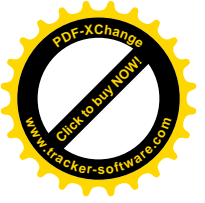
Figura 3. Modelo general del proyecto, Blindisp.



Fuente Autores.

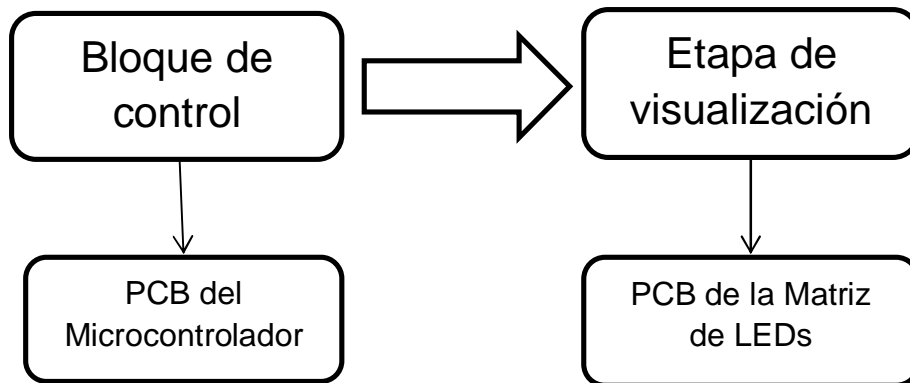
2.3. DISEÑO DE HARDWARE

En esta sección se explicará el diseño y la construcción del hardware utilizado en el proyecto. La implementación de un hardware específico se hizo necesaria debido a que se requería de un medio que permitiera visualizar los datos (matriz de LEDs) y con esto poder corroborar los resultados obtenidos durante cada etapa del proyecto. Para tener una mejor organización y manejo de las ideas. La descripción del proceso de construcción del hardware se dividió en dos



grandes bloques, los cuales son mostrados en la Figura 4 mediante un diagrama de bloques.

Figura 4. Etapas principales en el diseño y la construcción del hardware.



Fuente Autores.

2.3.1. Bloque de Control

El principal componente del bloque de control es un microcontrolador (MCU), el cual se encarga de recibir y organizar la información enviada por el PC a través del puerto USB, para posteriormente enviar estos datos a la matriz de LEDs por medio de sus pines de entrada y/o salida (pines I/O). El microcontrolador usado en este proyecto es el MC9S08JM60 de 64 pines y es fabricado por FREESCALE Semiconductors™. Las características generales de este MCU se pueden ver en la Tabla 1, y se pueden verificar en la respectiva hoja de datos [22].

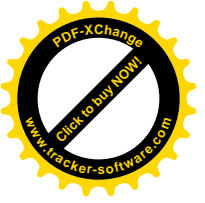


Tabla 1. Principales características del MCU MC9S08JM60.

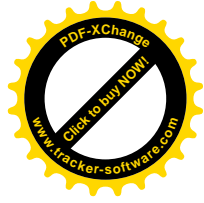
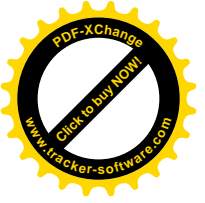
Característica	
Empaquetado	64LQFP
Voltaje de operación	2.7 [V] - 5.5 [V]
Memoria Flash	60912 [kB]
Memoria RAM	4096 [kB]
USB RAM	256 [kB]
ACMP (Comparador analógico)	Soportado
ADC (Conversor analógico - digital)	12 canales
I2C	Soportado
IRQ (Petición de interrupción)	Soportado
KBI (Interrupciones por teclado)	Soportado, 8 módulos
SCI (Comunicación serial)	Soportado, 2 módulos
SPI	Soportado, 2 módulos
TPM (Temporizador)	Soportado, 2 módulos
USB	Soportado
Pines I/O	51
MCG (Generador de reloj multipropósito)	Soportado

Fuente Hoja de datos MCU MC9S08JM60.

Los principales argumentos para la elección de este dispositivo son los siguientes:

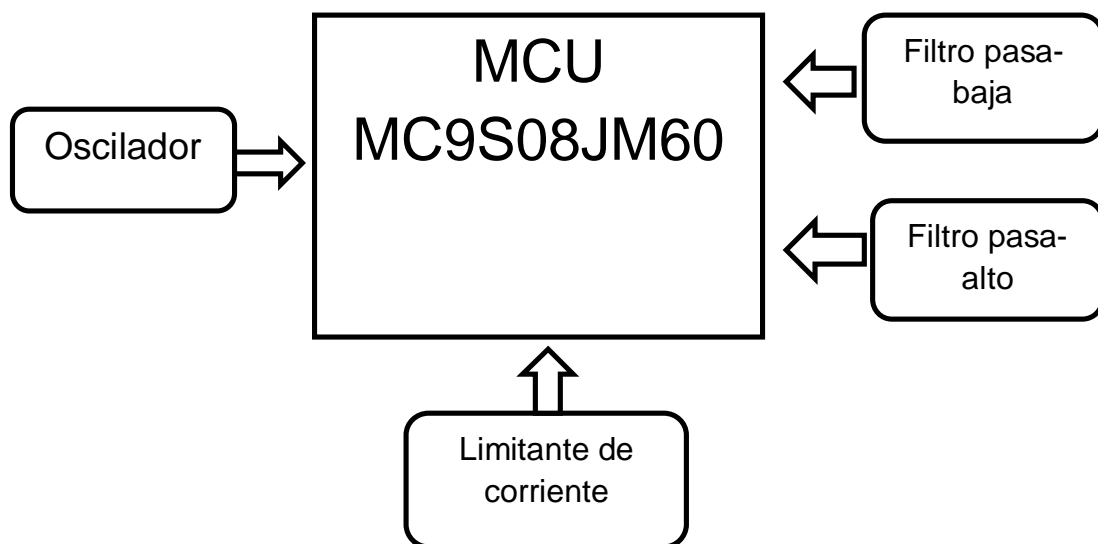
- Fácil adquisición, ya que este microcontrolador se puede obtener como una muestra gratuita o se puede conseguir en el mercado colombiano a precios no superiores a los \$8 USD.
- Amplia capacidad de memoria RAM, posee 4 [kB] de memoria RAM.
- Posee un módulo USB, que permite crear un dispositivo USB de forma rápida y configurarlo según nuestras necesidades.
- Al mismo tiempo posee 51 pines de propósito general, que se hacen necesarios para la transmisión de los datos a la matriz de LEDs.

Para el diseño y creación de esta etapa se utilizó EAGLE en su versión libre (software gratuito), un software CAD (Diseño Asistido por Computador, por sus



siglas en inglés) que se utiliza para diseñar circuitos impresos (*PCBs*). En nuestro proyecto quisimos que el MCU se alimentará directamente desde el puerto USB, por lo que hemos diseñado una PCB que cuenta con los componentes mostrados en la Figura 5, basándonos en el circuito de conexión de la Figura 6, el cual es suministrado por FREESCALE Semiconductors™ en la nota de aplicación AN3560 [23]. Nos basamos en este diagrama de conexión con el fin de garantizar la estabilidad y la correcta alimentación del microcontrolador.

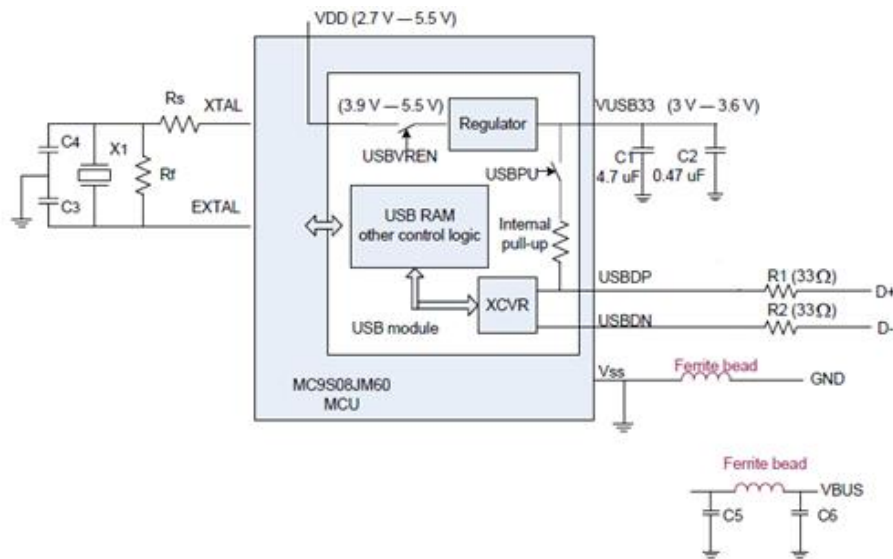
Figura 5. Modelo de la PCB del MCU.



Fuente Autores.

A continuación se enseña el circuito de alimentación de microcontrolador, este muestra detalladamente los diferentes dispositivos utilizados para realizar los módulos anteriormente descritos.

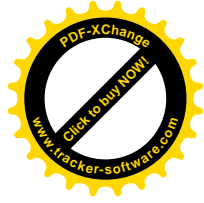
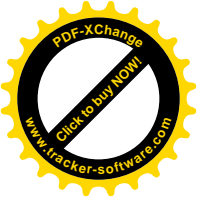
Figura 6. Circuito utilizado para alimentar el MCU.



Fuente AN3560, FREESCALE Semiconductors™.

La Figura anterior muestra de manera esquemática como van conectados los diferentes elementos al microcontrolador. A la izquierda se ve un cristal llamado X1 el cual es de 12 [MHz]; con una combinación de resistencias y condensadores que forman un filtro, el cual es un sistema con una frecuencia de resonancia igual a la del cristal, con el fin obtener una señal más estable; la resistencia R_s no se tuvo en cuenta en el diseño, es decir su valor es 0 [Ω], R_f es de 1 [M Ω], y los capacitores de 22 [pF].

El módulo USB necesita un reloj de referencia de 48 [MHz] y un reloj del bus de 24 [MHz]. Por tanto, se configura el MCG (Generador de Reloj Multipropósito) del microcontrolador para que a través de un PLL (lazo de seguimiento de fase) obtenga la frecuencia de referencia de 48 [MHz] y luego a través de un divisor de frecuencia obtenga el reloj del bus de 24 [MHz]. Lo anterior se realiza debido a la presencia de un cristal maestro de 12 [MHz].



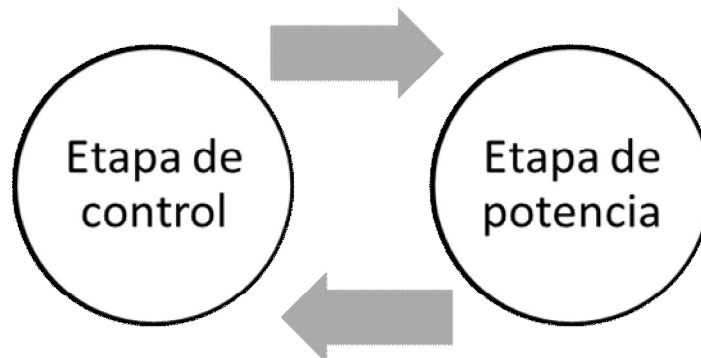
En el diseño, los condensadores C1 y C2 se hacen necesarios para la eliminación del rizado que puede presentar la línea de alimentación del bus USB, así como el núcleo de ferrita (en inglés, *ferrite bead*) y los condensadores C5 y C6 se utilizan para eliminar ruido de alta frecuencia que se puede presentar; las resistencias R1 y R2 que van en los pines de datos D+ y D-, son usadas para limitar la corriente que entrega el puerto USB a la hora transmitir datos.

En el Anexo A se muestra el diseño de la tarjeta de circuito impreso (*PCB*) que se realizó en el programa *EAGLE*.

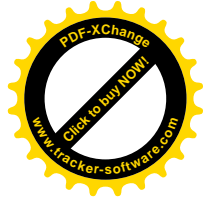
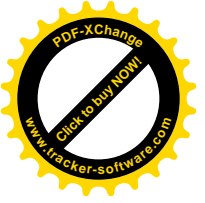
2.3.2. Etapa de visualización

La etapa de visualización hace referencia a la matriz de LEDs y todo el diseño que se hizo para que esta funcione de la forma deseada, lo cual es fundamental para poder observar los datos enviados desde el PC, es decir, es la que se encarga de convertir la información eléctrica en estímulos visuales. Esta etapa, al ser más compleja que la anterior, se divide en dos partes, una de control y otra de potencia.

Figura 7. Etapas del diseño de la PCB de la etapa de visualización.



Fuente Autores.



La matriz de LEDs que se construyó tiene una resolución de 32 x 25 pixeles y está conformada por 20 matrices individuales de 8 x 5 LEDs. La referencia de las matrices de LEDs individuales que se utilizaron es LTP-2558AA, las principales características de esta matriz de LEDs se pueden observar en la Tabla 2, datos que se pueden verificar en la hoja de datos del fabricante [24].

Tabla 2. Características eléctricas de la matriz de LEDs LTP-2558AA

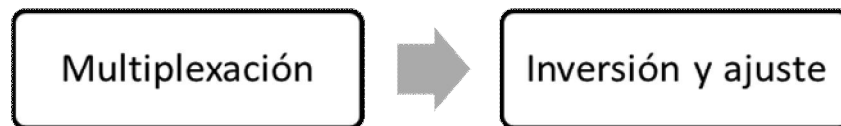
Característica	
Caída de voltaje en polarización directa	2.5 [V], si se polariza con 60 [mA]
Máxima disipación de potencia por punto	75 [mW]
Máximo pico de corriente de polarización por punto	100 [mA]
Máxima corriente de polarización por punto	25 [mA]
Máximo voltaje de reversa por punto	5 [V]

Fuente Hoja de datos LITE-ON LTP-2558AA.

La matriz de LEDs está diseñada para trabajar por multiplexación en el tiempo con el fin de reducir el consumo de energía; pero este sistema tiene la desventaja de que reduce el brillo promedio con que encienden los LEDs, por lo tanto se busca que la corriente que pase por cada punto de la matriz de LEDs sea cercana al valor máximo que estos soportan, y por esto se seleccionaron resistencias de polarización con un valor de 33 [Ω]. En el proceso de multiplexación lo que se hace es habilitar secuencialmente cada fila de la matriz de LEDs y en el lapso en que cada fila este habilitada encender solo los LEDs (las columnas de cada fila) que corresponden a puntos de la imagen que se quiere mostrar. Se hace necesario entonces hacer un barrido vertical de la matriz que habilite y deshabilite cada fila a una velocidad lo suficientemente alta como para que nuestro ojo pueda ser engañado, y no se perciba el momento en el que se enciende una u otra fila.

La etapa de control es la encargada de hacer el barrido vertical de los pixeles de la imagen que se desea mostrar. Para un mejor entendimiento, se dividió esta etapa en dos bloques que se muestran en la Figura 8.

Figura 8. División de la etapa de control.



Fuente Autores.

- ***Bloque de multiplexación.***

El bloque de Multiplexación está compuesto por 3 demultiplexores o decodificares, cada uno cuenta con 3 bits de entrada y 8 de salida disminuyendo la cantidad de puertos del microcontrolador a utilizar, además poseen 3 habilitadores, de los cuales dos se dejan fijos y el restante se utiliza para ir habilitando cada decodificador cuando sea necesario. Para ello se ha elegido el decodificador SN74LVC138A de Texas Instruments [26] y los criterios son los siguientes:

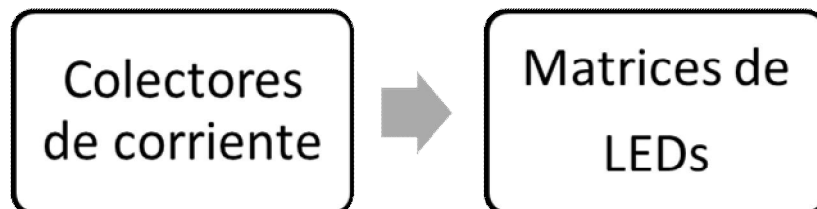
- Fácil adquisición y bajo costo, ya que se pueden obtener como muestras gratis de TI o se pueden comprar a valores cercanos a los \$0.5 USD mediante distribuidores como Digi-Key Corporation.
- Rango de operación adecuado, ya que los niveles de tensión con los que trabajan sus valores lógicos son compatibles con los del microcontrolador.

- **Bloque de inversión y ajuste**

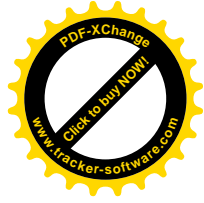
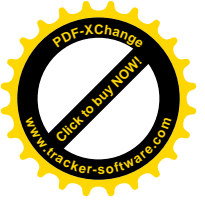
El bloque de inversión y ajuste está conformado por 2 circuitos integrados de inversores lógicos, cada uno de ellos cuenta con 16 inversores lógicos individuales; los cuales nos permiten cumplir las condiciones de operación de los transistores, que serán explicados más adelante en la etapa de potencia, además este bloque se hizo necesario debido a que el demultiplexor seleccionado posee salidas lógicas activas en bajo, lo cual no es deseado. Durante el diseño se eligió el inversor lógico SN74AHCT16240 [25] de Texas Instruments, los motivos principales son iguales que los anteriormente propuestos para los demultiplexores, agregando que estos inversores entregan una tensión estable en sus salidas lógicas, lo cual permite mantener los transistores en el estado de operación deseado, estos dispositivos mantienen una salida en alto mínimo $V_{OH\ min} = 3.8\ [V]$ y una salida en bajo máxima $V_{OL\ max} = 0.1\ [V]$.

Debido a la gran corriente que necesitan los LEDs para obtener un brillo perceptible a plena luz del día, se vio la necesidad de poner una etapa de potencia en la PCB de la matriz de LEDs. Dicha etapa está conformada por los transistores (que funcionan como colectores de corriente) y por las matrices de LEDs individuales.

Figura 9. División de la etapa de potencia.



Fuente Autores.



- **Colectores de corriente**

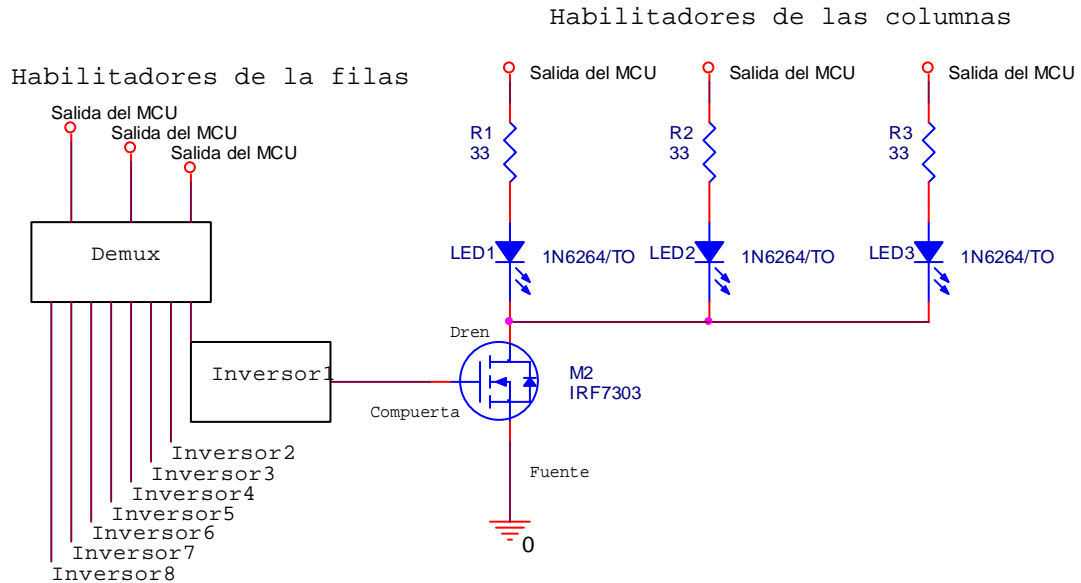
La etapa de colectores de corriente consta de 13 arreglos de transistores IRF7303. Cada uno posee 2 transistores MOSFET tipo N, los cuales están conectados como fuente común, estos transistores son de potencia y soportan hasta 5.3 [A]. El diseño del circuito se hizo buscando que los transistores trabajen como interruptores, es decir en las regiones de corte y trío. Para lograr que los transistores trabajen en corte, solo se debe hacer que la tensión entre la compuerta y la fuente (V_{GS}), no supere la tensión umbral de los transistores, la cual es de 1 [V]; esto se logra enviando una señal digital de cero a la compuerta del transistor. El transistor trabaja en la región de trío cuando se cumplen dos condiciones, primero que V_{GS} sea mayor que 1 [V] y segundo que la tensión entre la compuerta y el drenador del transistor (V_{GD}) sea mayor que la tensión umbral, es decir, que V_{GD} sea mayor que 1 [V]; esto se logra enviando un uno lógico que equivale aproximadamente a 4.5 [V].

- **Matrices de LEDs**

Los LEDs son los dispositivos que convierten las señales eléctricas enviadas a la matriz en señales visuales. Para su correcto funcionamiento se hace necesario colocar unas resistencias en la entrada de cada fila de LEDs, estas resistencias actúan como limitantes de corriente para prevenir posibles daños en los componentes. Asimismo la intensidad lumínica de los LEDs es directamente proporcional a la corriente que circula por ellos, luego se escoge una resistencia de 33 [Ω].

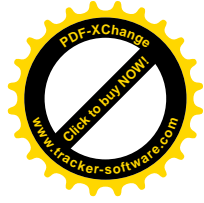
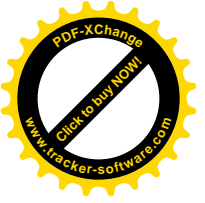
En la Figura 10 se muestra la forma en que se realizaron las conexiones de los diferentes componentes.

Figura 10. Conexión de los componentes de hardware en la etapa de visualización.



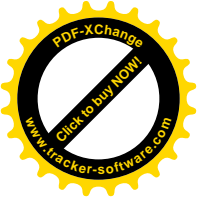
Fuente Autores

Analizando esta figura, se puede tener un mejor entendimiento y una idea más generalizada de la forma en que funciona el hardware que se implementó. En la parte izquierda de la figura se puede observar un demultiplexor, el cual recibe como entradas 3 pines de salida del microcontrolador y entrega 8 salidas, cada una de las cuales va a un inversor, y la salida de cada inversor va a un transistor. La parte descrita anteriormente corresponde al manejo de las filas de la imagen que se quiere mostrar. Como se mencionó anteriormente, la resolución total de la matriz va a ser de 32x25 pixeles, por lo cual se van a tener 25 filas, por esto se infiere que para el manejo de todas las filas se requiere de 25 transistores (los cuales están en los 13 arreglos de transistores IRF7303); 25 inversores, los cuales están en los dos circuitos integrados SN74AHCT16240 (cada uno tiene 16 inversores); y 25 entradas lógicas que lleguen a los inversores, 24 de estas entradas provienen de los 3 demultiplexores SN74LVC138A y una es entregada directamente por el microcontrolador. Como



se puede ver, con este sistema se están ahorrando pines del MCU, porque en lugar de utilizar 25 se usan 9 para controlar los demultiplexores (3 para cada uno) y uno que habilita directamente la última fila; por lo que solo 10 pines del microcontrolador son necesarios para la manipulación de las filas.

En la parte derecha de la Figura 10, se pueden observar una serie de resistencias y LEDs conectados en serie, y cada uno de estos se conecta al drenador del transistor. Cada arreglo de una resistencia conectada con un LED corresponde a un punto o columna de la fila que se está habilitando. En la Figura 10 solo se muestran 3 pero en el circuito real hay 32 de estas conexiones resistencia-LED que equivalen a las 32 columnas de la imagen y como se muestra en la figura, todos estos arreglos van al drenador de un mismo transistor, el cual se encarga de recolectar la corriente proveniente de cada LED. Cada resistencia se encuentra a su vez conectada a un pin de salida del microcontrolador, el cual se encarga de habilitar o deshabilitar cada columna o punto de la fila, de acuerdo a la imagen que se quiera mostrar.



CAPITULO 3

3.1 DISEÑO DE LAS RUTINAS DE SOFTWARE

Para la realización de este proyecto se requirió elaborar ciertas rutinas de software que se encargan de obtener la información visual de área de trabajo del programa CaRMetal, procesar dicha información, enviarla al microcontrolador a través del puerto USB y por último, enviar dicha información del microcontrolador a la matriz de LEDs. La elaboración de estas rutinas de software se llevó a cabo en dos módulos independientes que al final se acoplaron, los cuales se muestran en la Figura 11.

Figura 11. Módulos que se ejecutaron durante la elaboración de las rutinas de software.



Fuente Autores.

Para una fácil interpretación de la rutina de software partimos de las entradas y las salidas del proceso, para esto nos apoyamos en la siguiente imagen.

Figura 12. Diagrama de bloques de la rutina de software



Fuente Autores.

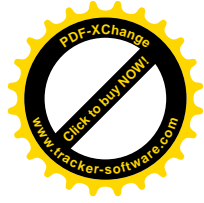
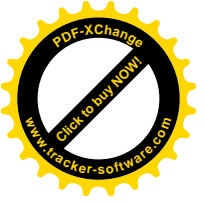
A continuación, denominamos “bloque de procesamiento” a todo el conjunto de modificaciones y tratamientos que se realizan a la imagen, así como el procedimiento que se utiliza para que la imagen pueda ser enviada desde el computador y pueda ser visualizada en la matriz de LEDs.

Dentro del bloque de procesamiento encontramos los siguientes pasos a seguir:

- Modificación del código fuente de CaRMetal.
- Rutina de programación del microcontrolador.

3.1.1. Modificación del código fuente de CaRMetal

CaRMetal [27] es un programa de geometría interactiva desarrollado por Eric Hakenholz, el cual hereda el motor C.a.R, este último es un potente algoritmo que sirve para manejar objetos y elaborar construcciones geométricas. CaRMetal es un programa de código libre y presenta una interfaz gráfica simple para un fácil entendimiento del usuario, este programa está escrito en el lenguaje de programación *JAVA* y es una herramienta muy potente a la hora de ser utilizada como instrumento didáctico en la enseñanza de la geometría.



CaRMetal tiene un código libre guiado con la licencia de uso GPL v3, esta nos permite modificar el código fuente de cualquier software o usar parte de él, indicando y dándole crédito a los creadores de este código.

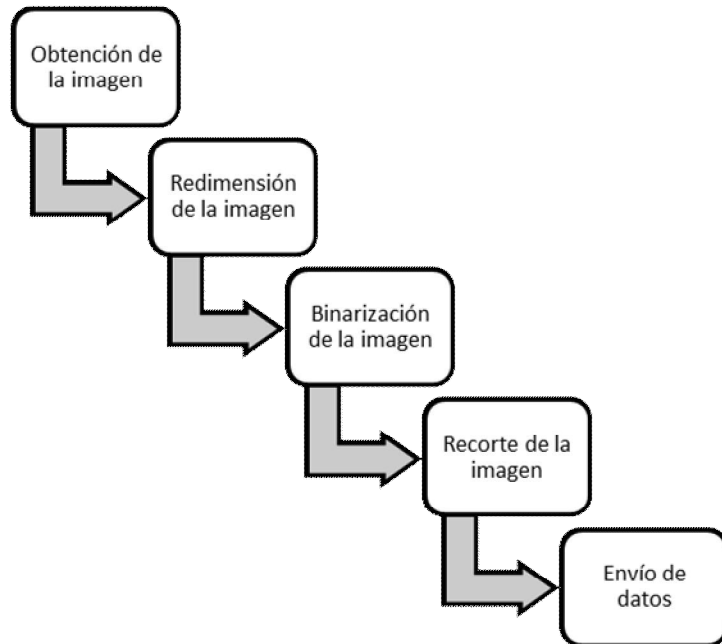
La principal razón para escoger este programa fue la libertad de su código y el fácil acceso al mismo, al igual que su buena interfaz gráfica permite una fácil adaptación del usuario a la utilización de este.

Para realizar la compilación y modificación del código, se utiliza el programa NETBEANS, entorno de desarrollo integrado (IDE, por sus siglas en inglés) que permite modificar y ejecutar códigos planteados en lenguaje *JAVA*. Se escogió este programa debido a que el código fuente de CaRMetal está escrito en lenguaje *JAVA*, además presenta una interfaz de fácil interpretación.

Dentro del paquete llamado *ERIC* encontramos las clases que utiliza el código para manejar la barra de herramientas y las imágenes (esto se debe a que el programa basa su funcionamiento en objetos tipo imágenes). Las clases (o librerías, en otros lenguajes de programación) que han sido modificadas son: *FileTools.Java*, el cual contiene los métodos que se utilizan para modificar los gráficos, como grosor de las líneas, entre otros y *JGeneralMenuBar.java*, donde se encuentran las funciones de todos los botones en la barra de menús.

Para este proyecto se agregó a la clase *JGeneralMenuBar.java* un menú llamado *Devices* con las opciones *connect* y *disconnect*. Para poder observar estas nuevas opciones en el idioma español, se editó la clase *JZirkelProperties_es.properties*, que contiene la traducción al español de los diferentes menús y opciones que se visualizan en el programa CaRMetal.

Figura 13. Modificación del código de CaRMetal.

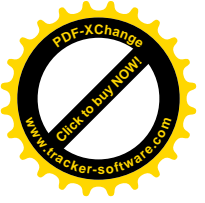


Fuente Autores.

El procedimiento con el cual se modificó el código de CaRMetal se puede esbozar en base a la Figura 13 y se explica detalladamente a continuación:

- ***Obtención de la imagen***

Al momento de obtener la imagen del área de trabajo se analizó el método que se ejecuta en el programa cuando se exporta el área de trabajo como una imagen con extensiones *png*, *eps*, *svg* o como un archivo *pdf*. Se encontró que la clase que contenía el método utilizado es *FileTools.java* y se determinó que el método *exportGraphicFile* era el que se ejecutaba la acción de capturar la imagen del área de trabajo. Una vez entendido este método, se procedió a crear un método denominado *exportGraphicBuffer*, en el cual se realizan los



mismos pasos iniciales del método original, pero al final en lugar de convertir los datos visuales a un formato de archivo conocido, lo que se hace es almacenar esta información en un objeto tipo *BufferedImage*, que es una imagen almacenada en la memoria RAM del sistema.

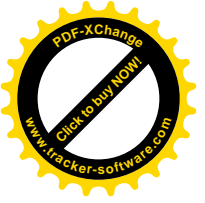
Después de haber obtenido la imagen se crea un método llamado *processImage* el cual contendrá la rutina para el procesamiento de la imagen.

- ***Redimensión de la imagen***

Para la redimensión se declaran los límites de la nueva imagen, los cuales están determinados por la matriz de LEDs, de esta forma tenemos como límites ancho = 32 unidades y alto = 25 unidades. Luego teniendo en cuenta que la imagen es de tipo *BufferedImage* se crea una imagen homóloga con las nuevas dimensiones y de tipo *TYPE_INT_RGB*. Con el método *AffineTransform.getScaleInstance* se obtiene un escalamiento de la imagen original.

- ***Binarización de la imagen***

Con el fin de binarizar la imagen se crea una segunda con las mismas características que la anterior pero con sus datos en binario. Para esto se obtiene y define el *Raster* de las dos imágenes, se declara un umbral de 200 (Como es sabido la imagen en RGB tiene pixeles cuyos valores varían entre 0 a 255). Luego se hace un barrido de la imagen y se umbraliza con dicho valor, volviendo cero (0) los pixeles que son menores al valor umbral y uno (1) a los pixeles con valores mayores al umbral. A continuación se procede a grabar la imagen con formato *bmp*.



- **Recorte de la imagen**

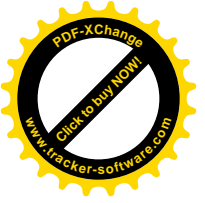
Una vez obtenida la imagen en formato *bmp* se procede a recortarla. Primero que todo se convierte la imagen a un arreglo de bytes con el método *toByteArray*. Analizando este arreglo de bytes se encuentra que los primeros 62 bytes hacen referencia al formato y por ende no nos interesan, los 100 bytes restantes tienen la información de la imagen pero para que sean de utilidad hay q invertir este arreglo de bytes. Luego de invertir el arreglo se utiliza el método *System.arraycopy*, el cual recorta el arreglo de manera tal que solo quedan los 100 bytes que contienen la información de la imagen.

Para poder enviar la imagen primero se debe hacer la instalación del *driver* que va a permitir la comunicación entre el lenguaje en *JAVA* y el puerto USB, debido a que *JAVA* no soporta transferencia de datos por USB directamente; este driver cumple la función de *JNI* (*Java Native Interface*, por sus siglas en ingles), que permite que un programa escrito en *JAVA* pueda interactuar con programas escritos en otros lenguajes. Este acoplamiento se ve de suma necesidad, ya que el programa trabajado en cuestión basa su código en *JAVA* y el sistema operativo permite acceder a sus puertos mediante códigos escritos en lenguaje *C*.

Se encontró un controlador que permite realizar todos los tipos de envío USB y acoplar cualquier dispositivo al driver por medio de su generador de archivos de tipo *INF*, además tiene compatibilidad con Windows 7 para plataformas tanto de 32 bits como de 64 bits. El conjunto de controladores se encuentra con el nombre de *Libusb-win32-wrapper* [28].

- **Envío de datos**

Mediante el anterior driver se realiza el envío de datos, el método *writeBulk* es el encargado de hacer este procedimiento, para esto hay que suministrar al método: la dirección del punto terminal al cual se quiere enviar; los datos a



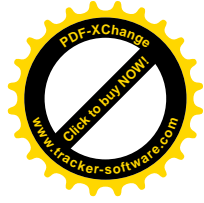
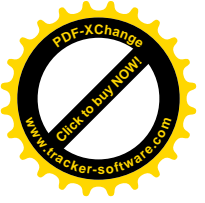
enviar; el tamaño de los mismos (este está limitado por el tamaño del punto terminal que se defina a la hora de programar el *MCU*) y el *timeout*, el cual es el tiempo milisegundos en el que el programa está actualizando el punto terminal para saber si está disponible para enviar información.

En la clase *JGeneralMenuBar.java* se crea la rutina que se ejecuta cuando se presiona el botón de conectar en el menú *Devices*. Se empieza creando un *timer* (o contador) con un tiempo de 20 milisegundos y se hace llamado al método *exportGraphicBuffer* el cual es el encargado de ejecutar todos los métodos anteriormente descritos. Seguidamente se abre el dispositivo que se encargará de recibir los datos, es decir, se habilita para que esté atento al envío o recepción de datos mediante puerto USB. Luego, en el botón de desconectar se detiene el *timer* y se cierra el dispositivo.

3.1.2. Rutina de programación del microcontrolador

La rutina que se implementó para programar el microcontrolador se encarga de inicializarlo, configurar el módulo USB, manejar la transferencia de datos por USB, almacenar los datos recibidos y enviarlos a la matriz de LEDs. Para realizar este módulo se utilizó el software *Codewarrior Development Studio* ofrecido por FREESCALE Semiconductor™; se usó específicamente la edición especial, revisión 10.2, de este software, la cual es totalmente funcional y se puede descargar gratuitamente, aunque presenta algunas restricciones en el tamaño del código para la compilación.

En el diseño de hardware se explicó que se utilizó el microcontrolador MC9S08JM60 de FREESCALE Semiconductor™, entre otras razones porque contiene un módulo USB configurable. Decidimos transferir los datos por USB debido a que es un protocolo robusto y ampliamente utilizado en los dispositivos



actuales. Para una mejor comprensión de la rutina con la que se programó el microcontrolador refiérase al anexo B.

3.1.2.1. Módulo USB del microcontrolador MC9S08JM60

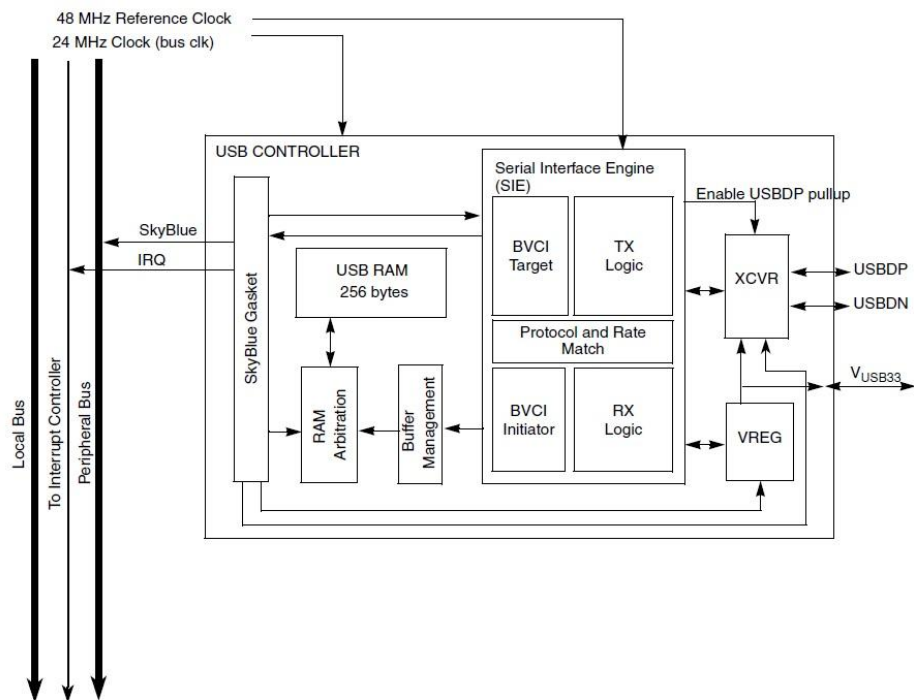
Una vez que hemos entendido las generalidades del protocolo USB, podemos adentrarnos a estudiar las características que tiene la unidad USB que nos ofrece el microcontrolador seleccionado. El módulo USB del microcontrolador MC9S08JM60 presenta las siguientes características:

- Permite configurar un dispositivo USB 2.0 *full-speed* (12 [Mbps])
- Posee un transceptor USB dedicado interno que se encarga del envío de los paquetes.
- Incluye un regulador interno de 3.3 [V], y permite seleccionar mediante software si se desea usar éste o un regulador externo.
- Tiene una resistencia de *pull-up* USB DP incorporada, necesaria para conectar o desconectar el dispositivo USB del bus.
- Soporta transferencias de control, mediante interrupciones, isócronas y transferencia de datos en masa (tipo *bulk*).
- Dispone de siete puntos terminales (*endpoints*).
- Soporta la configuración del punto terminal 0 como bidireccional y de 6 puntos terminales unidireccionales adicionales.
- Tiene la capacidad de utilizar los puntos terminales 5 y 6 para ofrecer la capacidad de doble búfer.
- Tiene 256 bytes de memoria RAM dedicados al manejo de datos USB.

La Figura 14, muestra el diagrama de bloques de los diferentes componentes que se incluyen en el módulo USB del microcontrolador. Para una información

más detallada sobre el módulo USB del MC9S08JM60 se puede consultar la respectiva hoja de datos [22]; y para tener una guía sobre cómo se debe configurar el *MCU* para que este módulo trabaje de acuerdo a nuestro propósito, se debe consultar la nota de aplicación AN3560 [23] ofrecida por FREESCALE Semiconductor®.

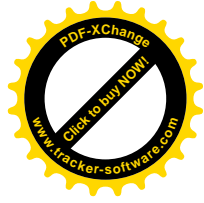
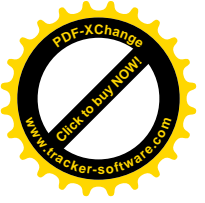
Figura 14. Diagrama de bloques del módulo USB del MC9S08JM60



Fuente AN3560, FREESCALE Semiconductors.

3.1.2.2. Rutina de software implementada en el microcontrolador

Como se ha mencionado el microcontrolador es el encargado de recibir los datos del puerto USB, almacenarlos, reorganizarlos y enviarlos a la pantalla de LEDs para que se puedan visualizar correctamente. Para llevar a cabo esto, el microcontrolador debe ser instruido con una rutina de software que le indique



qué secuencia ejecutar en determinado momento del proceso. FREESCALE Semiconductor ® a través del paquete “USB-MINI Stack for S08JM Family” suministra la rutina de software básica para programar el MC9S08JM60 como un dispositivo USB, además de algunas rutinas de ejemplo para configurar el microcontrolador como un mouse y como un conversor de *SPI*, *I2C* y serial a USB.

Para llevar a cabo este proyecto se utilizó la rutina básica que configura el microcontrolador como un dispositivo USB. La explicación del código implementado en el microcontrolador se dividirá en varias etapas para un mejor entendimiento. La secuencia de ejecución de estas etapas es la siguiente:

- Inicialización del microcontrolador.
 - Inicialización y sostenimiento de la comunicación USB.
 - Recepción y organización de datos.
 - Envío de datos a puertos.
-
- ***Inicialización del microcontrolador***

Esta rutina se encarga de cargar los módulos básicos del microcontrolador necesarios para su correcto funcionamiento. Además de esto, en esta parte del código se configura el generador de reloj (*MCG*) del *MCU* para que a partir de un cristal externo de 12 MHz se pueda generar un reloj de 48 MHz utilizado como referencia en la comunicación USB, y a partir de éste se obtiene un reloj de 24 [MHz], el cual será el reloj de trabajo del microcontrolador y el reloj del bus del protocolo USB. El reloj de 48 [MHz] se obtiene al usar lazo de seguimiento de fase configurado por software, y el reloj de 24 [MHz] se produce al aplicar un divisor de frecuencia, configurado por software también, al reloj de 48 [MHz] obtenido.

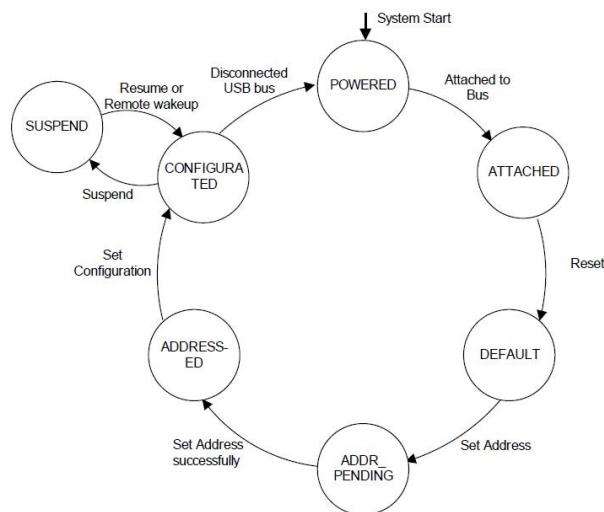
- **Inicialización y sostenimiento de la comunicación USB**

En esta sección del código se utilizó una máquina de estados que describe los diferentes estados en que se puede encontrar el dispositivo USB al estar conectado al computador anfitrión. Los estados de esta máquina son:

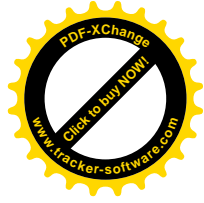
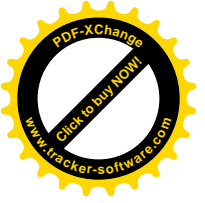
- *Powered*: El dispositivo USB no está ligado al bus USB, pero el microcontrolador está siendo alimentado.
- *Attached*: El dispositivo USB está ligado al bus USB, pero no se ha enumerado en la computadora anfitriona.
- *Default*: El dispositivo USB es reiniciado por el PC anfitrión.
- *Addressed*: El PC anfitrión ha enumerado el dispositivo USB satisfactoriamente.
- *Configured*: El dispositivo USB se ha configurado correctamente en el host.
- *Suspend*: El dispositivo USB entra en modo de suspensión.

En la Figura 15 se muestra la secuencia que ejecuta la máquina de estados.

Figura 15. Transferencia de estados del dispositivo USB



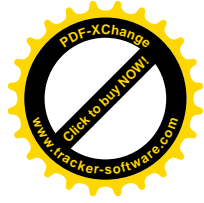
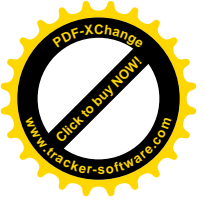
Fuente AN3560, FREESCALE Semiconductors™.



Inmediatamente el dispositivo USB es conectado al puerto USB, el dispositivo es establecido en el estado *Powered*. Después el dispositivo es ligado al bus USB y el estado cambia a *Attached*. Posteriormente, el *host* reinicia el dispositivo para que su estado cambie a *Default* y luego inicia el proceso de enumeración. Durante este proceso el dispositivo USB entrega el descriptor de dispositivo, el *host* le asigna una dirección y el estado del dispositivo cambia a *Addressed*. Luego el *host* solicita al dispositivo el descriptor de configuración, el descriptor de interfaz y el descriptor de punto terminal. El PC anfitrión instala el *driver* correspondiente y el estado del dispositivo cambia a *Configured*. Una vez el dispositivo está configurado, si el *host* detecta que no hay actividad en el bus durante 3[ms], el dispositivo entra al estado *Suspended* y continúa en este estado hasta que haya una nueva transferencia de datos. Toda la transmisión de los descriptores se realiza utilizando el punto terminal 0 (cero) del microcontrolador, mediante transmisiones de control.

- **Recepción y organización de datos**

Una vez el microcontrolador se encuentra conectado y configurado en el bus USB, éste queda atento a la llegada de cualquier paquete desde el *host* (computador anfitrión). La imagen es enviada por el *host* en dos fragmentos, uno de 64 bytes y otro de 36 bytes, donde cada bit es un punto de la imagen; por esto, el código implementado realiza el envío de los datos a través del punto terminal 1 y del punto terminal 3. A través del descriptor de punto terminal, los puntos terminales 1 y 3 son configurados de 64 bytes y ambos se configuran como de entrada. Con el fin de detectar alguna solicitud de transferencia de datos del *host*, se utiliza un bucle que revisa constantemente los puntos terminales mencionados anteriormente. Cuando una solicitud de transferencia de datos es detectada, se ejecuta una instrucción que toma los datos del punto terminal y los copia a un búfer temporal, luego de esto se toman los datos de



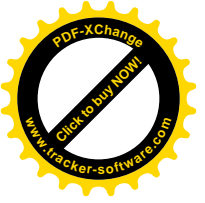
éste búfer y se almacenan ordenadamente en un vector de 100 posiciones, donde cada posición utiliza un byte de memoria.

- ***Envío de datos a puertos***

Una vez se tienen los datos almacenados en un vector, se utiliza una función que los toma ordenadamente y los envía a los puertos correspondientes. Para entender mejor como se deben enviar los datos y así lograr una correcta visualización de la imagen, debemos recordar que en la etapa el diseño del hardware se planteó que la matriz de LEDs iba a funcionar mediante multiplexación por tiempo. Por esta razón, la información visual contenida en el vector creado en la etapa de recepción no se puede enviar completa. En lugar de esto se envían fragmentos de 4 bytes, los cuales equivalen a cada fila de la pantalla y se realiza un barrido fila por fila a una frecuencia tal que el ojo no note en qué momento se apaga una fila y se enciende la otra. Para lograr enviar el fragmento correcto, en el código se utiliza un contador que almacena la posición de los últimos cuatros bytes enviados y se incrementa cada vez que se van a enviar los siguientes cuatro bytes (se cambia de fila).

Otro problema que surge a la hora de enviar los datos a los puertos que controlan los pines de la matriz, es que los datos vienen en paquetes de 4 bytes y a cada pin se debe enviar un bit de dicho fragmento, por lo que estos paquetes se deben descomponer en cada uno de sus bits y luego de esto se pueden enviar estos bits individuales al puerto correspondiente. La obtención de los bits individuales de cada paquete de 4 bytes se logra con los siguientes pasos:

1. Se divide el paquete en cada uno de sus bytes, es decir se divide la información de una fila en 4 fragmentos.
2. Se crean máscaras para cada bit, es decir vectores de 8 bits (1 byte), que estén conformadas por ceros y que solo tengan un uno en la posición del bit que se desea obtener.



3. Se aplica la operación lógica *AND* entre cada máscara que se creó y el byte que se está trabajando.
4. El resultado de cada una de las operaciones lógicas *AND* aplicadas es un vector de 8 bits (1 byte) y cada uno éstos resultados se almacenan en un vector de 8 bytes. Este arreglo está conformado por bytes que solo pueden tener 2 tipos de datos, 0x00 o 0x01, en hexadecimal. Donde cada byte de este arreglo corresponde a un bit del byte original al cual se le aplicaron las máscaras.
5. Se toma cada byte del vector de 8 bytes creado en el paso anterior y se verifica si contiene el dato 0x00 o el dato 0x01, de tal forma que cuando contiene el dato 0x01 se envía un 0 lógico al puerto correspondiente a ese bit en la matriz y, si contiene el dato 0x00 se envía un 1 lógico a dicho puerto.
6. Se toma el siguiente byte del arreglo de 4 bytes original que se tomó, es decir el siguiente fragmento de la fila y se aplica le aplican los pasos 2, 3, 4 y 5.
7. Se toman los siguientes 4 bytes, es decir la siguiente fila de la imagen y se aplican los pasos desde el 1 hasta el 6.

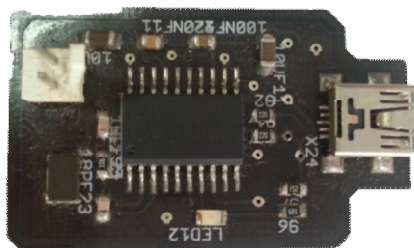
CAPITULO 4

En este capítulo se muestra la implementación hecha para cumplir con los objetivos propuestos, así como un método de prueba que se le realizó al sistema con el fin de corroborar su correcto funcionamiento. Se utilizan fotografías como único medio de dejar evidencia del método de prueba ya que la función de la matriz es convertir en estímulos visuales la información enviada desde el computador.

4.1. IMPLEMENTACION DEL SISTEMA

En esta sección se mostrará la implementación final del sistema, con cada uno de los componentes que se diseñaron y utilizaron durante cada una de las etapas. En la Figura 16 se muestra la programadora, circuito que permite la correcta programación del microcontrolador, por medio de este circuito se le puede ingresar toda la rutina mostrada anteriormente. En la imagen se pueden evidenciar los diferentes componentes necesarios para la programación.

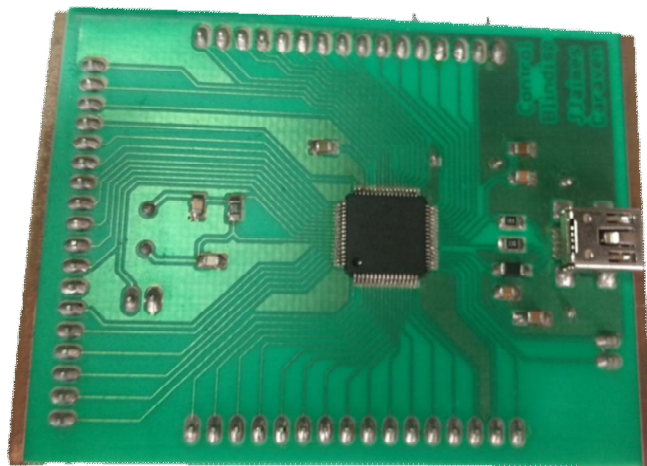
Figura 16. Programadora



Fuente Autores.

En la Figura 17 se muestra el montaje final de la *PCB* del microcontrolador con todos los componentes descritos anteriormente, como ya se ha dicho este es el encargado de recibir la imagen que se quiere mostrar a través del puerto USB, ordenarla y enviarla de forma correcta a la matriz de LEDs para que pueda ser visualizada. En la Figura 17 se pueden observar los diferentes componentes para la correcta alimentación del dispositivo, como son: los filtros y el oscilador, entre otros.

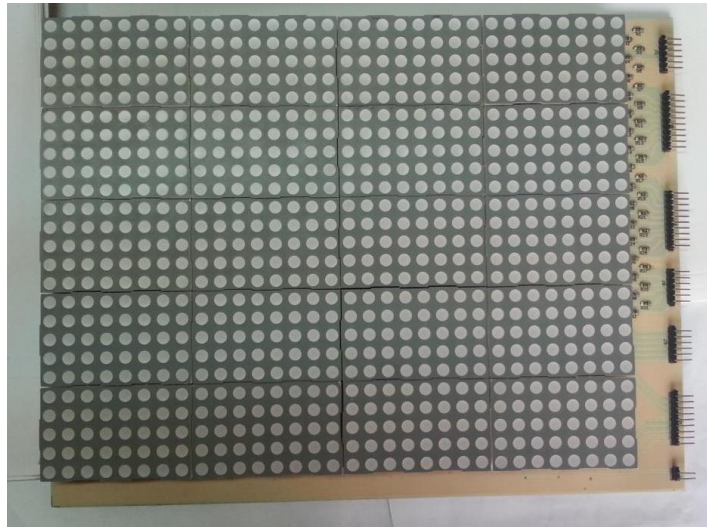
Figura 17. Tarjeta del Microcontrolador



Fuente Autores.

Siguiendo el orden predispuesto se muestra en la Figura 18 la cara frontal de la *PCB* de la matriz de LEDs. Como se puede ver en la figura, en esta cara de la *PCB* solo se encuentran las matrices de LEDs. Esto es debido a que en el momento del diseño se evitó poner componentes electrónicos en esta misma cara para tener una correcta visualización de los datos.

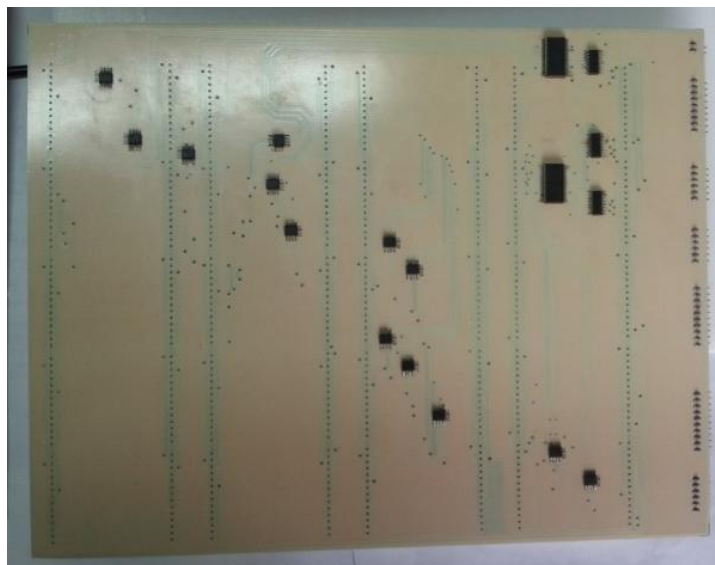
Figura 18. Cara frontal de la matriz de LEDs



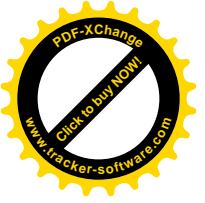
Fuente Autores.

En la Figura 19 se muestra la cara posterior de la *PCB* de la matriz de LEDs. Aquí se pueden observar los componentes encargados de la multiplexación, la inversión y la etapa de potencia del sistema.

Figura 19. Cara posterior de la matriz de LEDs



Fuente Autores.



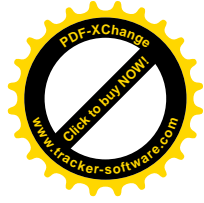
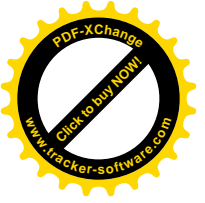
4.2. RESULTADOS DE LAS PRUEBAS

Para comprobar el correcto funcionamiento y actualización en el tiempo del sistema, se propone un método de prueba que consiste en hacer 4 figuras geométricas (3 figuras básicas y 1 compuesta). Las figuras geométricas básicas creadas fueron un círculo, un triángulo y un cuadrado; mientras que la figura compuesta utilizada se asemeja a una casa. El procedimiento que se empleó durante esta etapa de verificación fue el siguiente:

1. Se crea la figura mediante la cual se va a verificar el funcionamiento del sistema.
2. Se obtiene la imagen original del área de trabajo del programa y se almacena en el disco duro como una imagen con formato *png*.
3. Se obtiene la imagen procesada, esta es la imagen final que se va a enviar desde el computador después de haber sido reducida y binarizada. Esta imagen es almacenada en el disco duro como un archivo con formato *bmp*.
4. Por último se obtiene la imagen en la matriz de LEDs, esto se hace mediante fotografías debido a que es el único recurso que se tiene para dejar evidencia de la imagen visualizada en la matriz.

4.2.1. Cálculo del error para las imágenes enviadas a la matriz

Durante la realización del proyecto se evidenció que la limitante para formar unas figuras geométricas de buena calidad es la resolución de la matriz de LEDs. Es por esto que se hace necesario poder medir un error entre la resolución de la imagen original y de la imagen procesada. Para tener un buen cálculo del error se propone el siguiente método:



- Se realiza un pequeño código en lenguaje *JAVA* que se ejecuta al seleccionar la opción Conectar en el menú *Devices*, previamente creado en el programa CaRMetal. La función del código es contar los píxeles que forman la figura geométrica de la imagen original, es decir la del área de trabajo y, al mismo tiempo cuenta los píxeles que forman la figura geométrica después del procesamiento de la imagen. Después de esto, se calcula la razón entre el número de píxeles que forman la figura geométrica en la imagen original y el número total de píxeles que forman esta imagen, tomando a esta razón como el valor teórico. De la misma forma, se calcula la razón entre el número de píxeles que forman la figura geométrica en la imagen reducida y el número total de píxeles que forman esta imagen y, se toma esta razón como el valor experimental. Por último, se aplica la fórmula del método experimental para obtener el error.

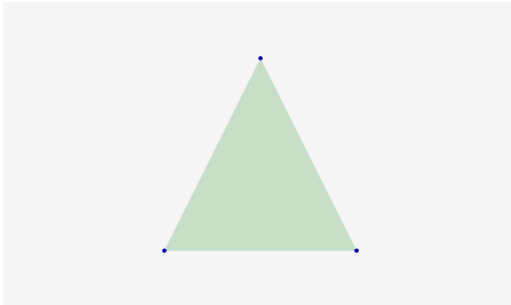
$$Error = \frac{valor\ teórico - valor\ experimental}{valor\ teórico} * 100$$

Durante el proceso de verificación se analizaron varias imágenes, a las cuales se les aplicó el método explicado anteriormente para el cálculo del error. Los resultados principales se presentan a continuación.

- **Triángulo**

Se realizó la figura de un triángulo en el programa CaRMetal y en la Figura 20 se muestra la imagen que se obtuvo del área de trabajo.

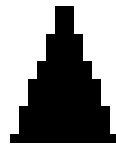
Figura 20. Triangulo en el área de trabajo de CaRMetal.



Fuente Autores.

Después de esto en la Figura 21, se muestra la imagen reducida, es decir la imagen procesada lista para ser enviada a la matriz.

Figura 21. Imagen reducida del triángulo.



Fuente Autores.

Por último, en la Figura 22 se muestra la fotografía de la matriz de LEDs con la figura de triángulo.

Figura 22. Fotografía de la matriz de LEDs, figura triángulo.



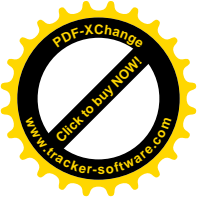
Fuente Autores.

Con el código antes mencionado se obtuvieron los siguientes datos:

Tabla 3. Datos de la figura triángulo.

número de pixeles imagen original	81649
número de pixeles totales original	678216
número de pixeles imagen reducida	94
número de pixeles totales reducida	800

Fuente Autores.

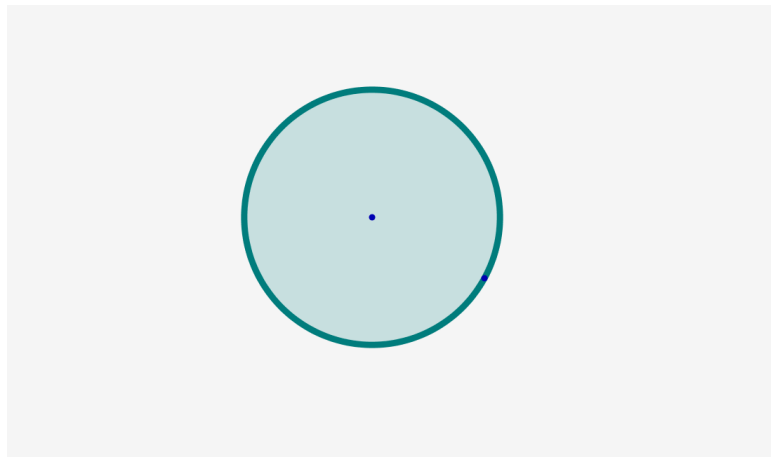


Con estos datos, aplicando la fórmula se obtiene un error de 2.40 %

Círculo

Se realizó la figura de un círculo en el programa CaRMetal y en la Figura 23 se muestra la imagen que se obtuvo del área de trabajo.

Figura 23. Círculo en el área de trabajo de CaRMetal.



Fuente Autores.

En la Figura 24, se muestra la imagen del círculo reducida.

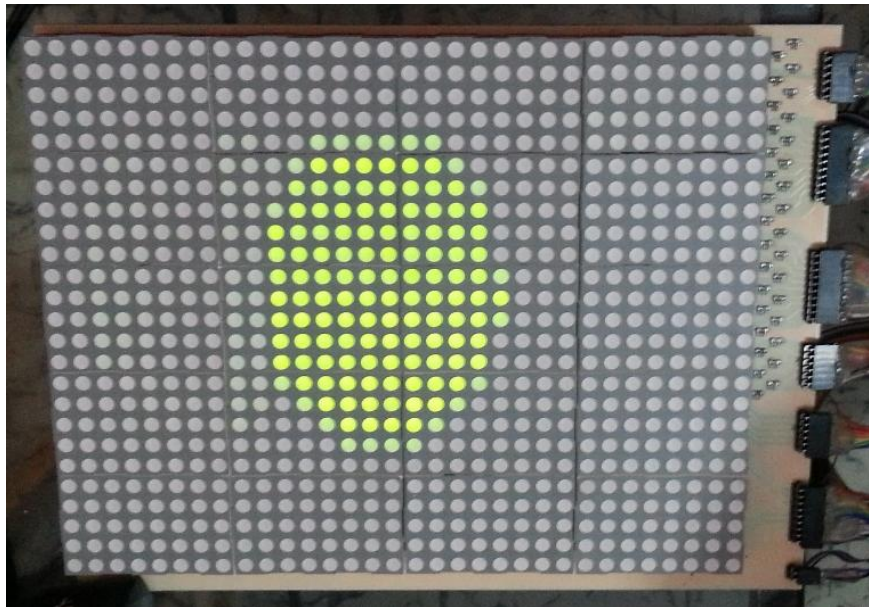
Figura 24. Imagen reducida del círculo



Fuente Autores.

En la Figura 25 se muestra la fotografía de la matriz de LEDs con la figura del círculo.

Figura 25. Fotografía de la matriz de LEDs, figura círculo.



Fuente. Autores.

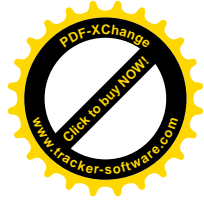
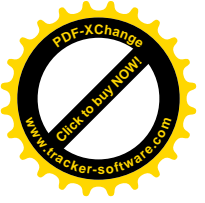
Mediante el código implementado para calcular el error se obtuvieron los siguientes datos:

Tabla 4. Datos de la figura círculo.

número de pixeles imagen original	68878
número de pixeles totales original	678216
número de pixeles imagen reducida	79
número de pixeles totales reducida	800

Fuente Autores.

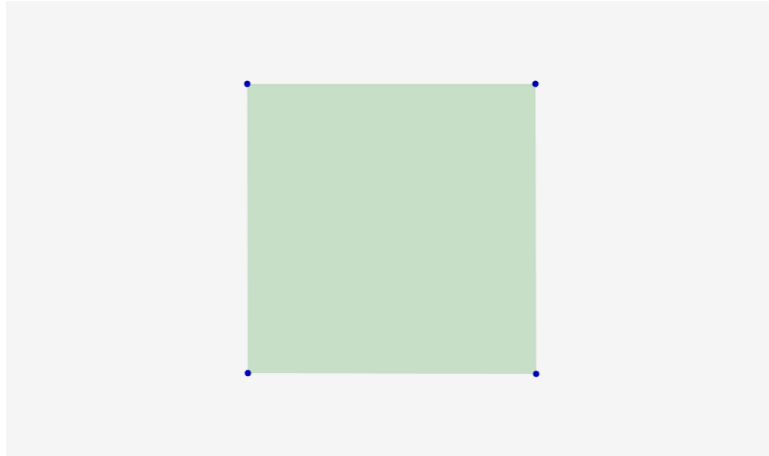
Con estos datos, aplicando la fórmula se obtuvo un error de 2.76%



- **Cuadrado**

Se realizó la figura de un cuadrado en el programa CaRMetal y en la Figura 26 se muestra la imagen que se obtuvo del área de trabajo.

Figura 26. Cuadrado en el área de trabajo de CaRMetal.



Fuente. Autores.

En la Figura 27, se muestra la imagen del cuadrado reducida y binarizada.

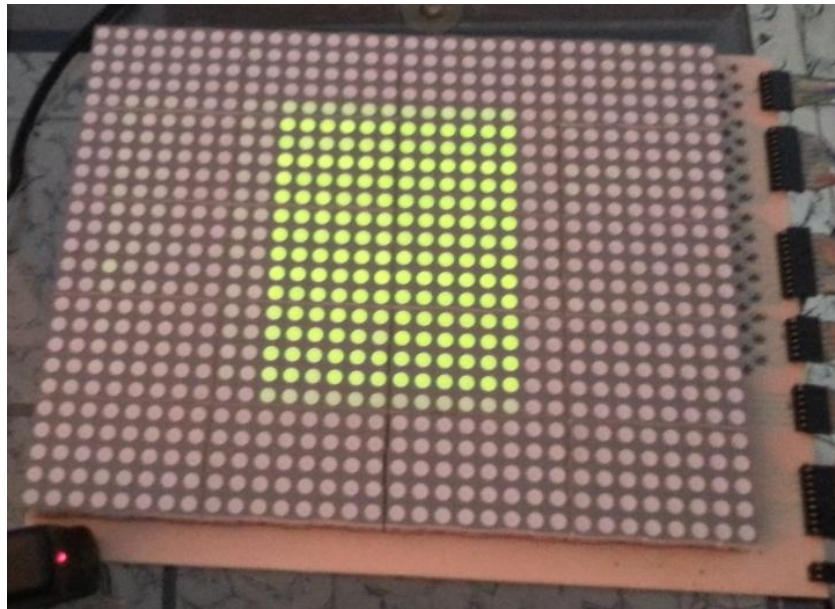
Figura 27. Imagen reducida y binarizada del cuadrado.



Fuente. Autores.

En la Figura 28 se muestra la fotografía de la matriz de LEDs con la figura del cuadrado.

Figura 28. Fotografía de la matriz de LEDs, figura cuadrado.



Fuente. Autores.

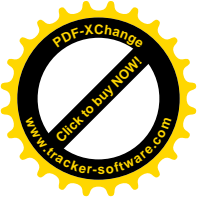
Los datos arrojados por el código utilizado para calcular el error fueron los siguientes:

Tabla 5. Datos de la figura cuadrado.

número de pixeles imagen original	168036
número de pixeles totales original	678216
número de pixeles imagen reducida	204
número de pixeles totales reducida	800

Fuente Autores.

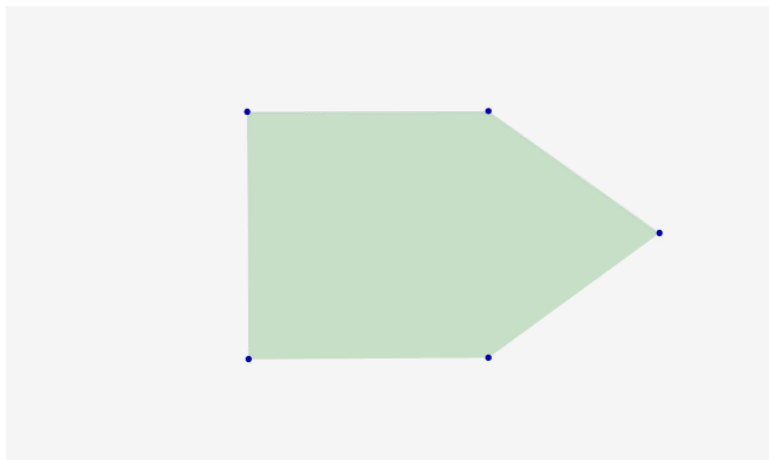
Con estos datos, aplicando la fórmula se obtuvo un error de 2.92%



- **Casa**

Se realizó la figura de una casa en el programa CaRMetal y en la Figura 29 se muestra la imagen que se obtuvo del área de trabajo.

Figura 29. Casa en el área de trabajo de CaRMetal.



Fuente. Autores.

En la Figura 30, se muestra la imagen del cuadrado reducida.

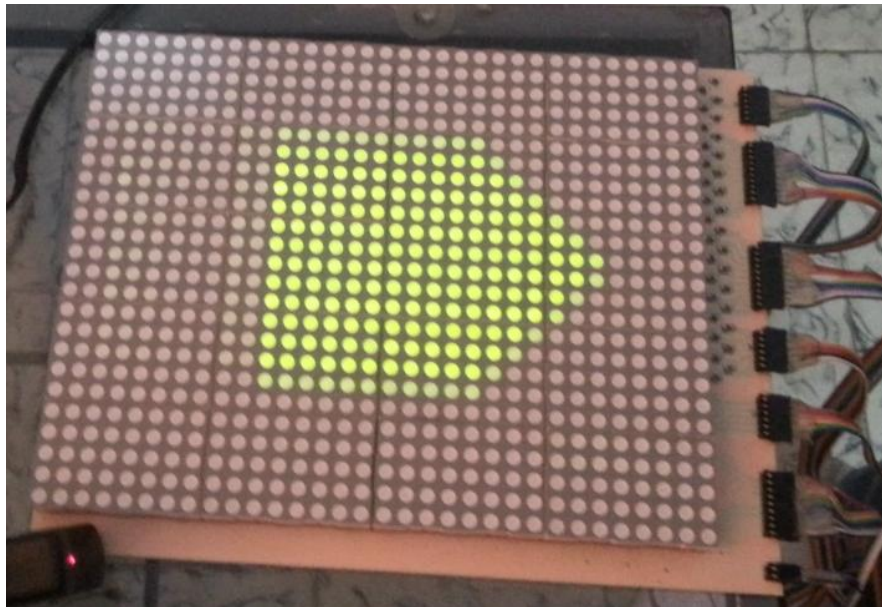
Figura 30. Imagen reducida de la casa.



Fuente. Autores.

En la Figura 31 se muestra la fotografía de la matriz de LEDs con la figura de la casa.

Figura 31. Fotografía de la matriz de LEDs, figura casa.



Fuente. Autores.

Los datos arrojados por el código utilizado para calcular el error fueron los siguientes:

Tabla 6. Datos de la figura casa.

número de pixeles imagen original	156832
número de pixeles totales original	678216
número de pixeles imagen reducida	179
número de pixeles totales reducida	800

Fuente Autores.

Con estos datos, aplicando la fórmula, se obtuvo un error de 3.24%

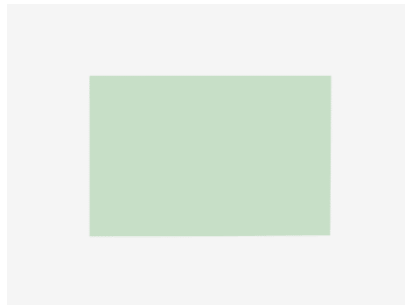
4.2.2. Cálculo del error para imágenes del doble de resolución, manteniendo la relación de aspecto

Con el fin de corroborar nuestra observación sobre la causa del error que se presenta en la imagen que se muestra en la matriz de LEDs, se realizó el mismo procedimiento de la sección pero aumentado la resolución de la imagen de salida a 64 unidades de ancho por 38 unidades de alto (para conservar la relación de aspecto), obteniendo los resultados que a continuación se presentan. Se aplicó esta prueba solo a las imágenes que presentaron el mayor coeficiente de error durante la prueba anterior, como lo son el cuadrado y la casa.

- **Cuadrado**

Se realizó la figura de un cuadrado en el programa CaRMetal y en la Figura 32 se muestra la imagen que se obtuvo del área de trabajo.

Figura 32. Cuadrado en el área de trabajo de CaRMetal para la prueba 2.



Fuente. Autores.

En la Figura 33, se muestra la imagen del cuadrado reducida.

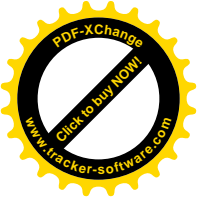


Figura 33. Imagen reducida del cuadrado para la prueba 2.



Fuente. Autores.

Los datos arrojados por el código utilizado para calcular el error fueron los siguientes:

Tabla 7. Datos de la figura cuadrado durante la prueba 2.

número de pixeles imagen original	158794
número de pixeles totales original	678216
número de pixeles imagen reducida	560
número de pixeles totales reducida	2432

Fuente Autores.

Con estos datos, aplicando la fórmula, se obtuvo un error de 1.65%

- **Casa**

Se realizó la figura de una casa en el programa CaRMetal y en la Figura 35 se muestra la imagen que se obtuvo del área de trabajo.

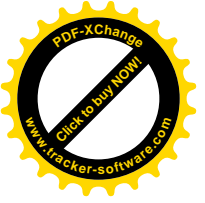
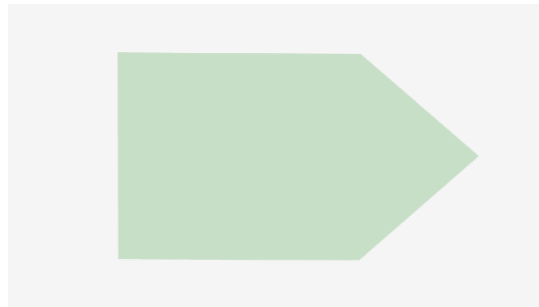


Figura 34. Casa en el área de trabajo de CaRMetal para la prueba 2.



Fuente. Autores.

En la Figura 36, se muestra la imagen del cuadrado reducida.

Figura 35. Imagen reducida de la casa para la prueba 2.



Fuente. Autores.

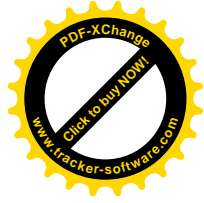
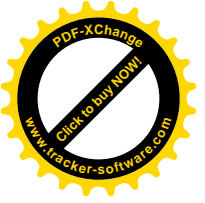
Los datos arrojados por el código utilizado para calcular el error fueron los siguientes:

Tabla 8. Datos de la figura casa durante la prueba 2.

número de pixeles imagen original	254986
número de pixeles totales original	678216
número de pixeles imagen reducida	921
número de pixeles totales reducida	2432

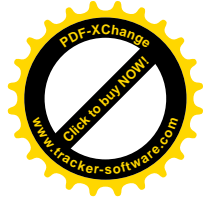
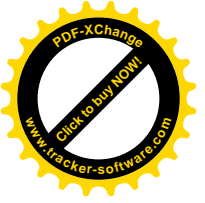
Fuente Autores.

Con estos datos, aplicando la fórmula, se obtuvo un error de 0.73%

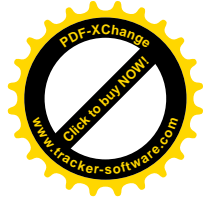
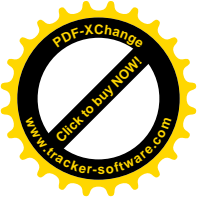


5. CONCLUSIONES

- Se diseñó un método de prueba en el cual se cuantificaron las diferencias entre la imagen original y la imagen procesada. Esta diferencia se toma como el coeficiente de error entre las dos imágenes. Luego de realizar varias pruebas se concluye que al aumentar la resolución de la matriz se disminuye el coeficiente de error, mejorando así el nivel de detalle de la imagen.
- Se implementa un sistema en el cual los datos de la imagen procesada son enviados desde el microcontrolador a la matriz de LEDs multiplexados en el tiempo, logrando así un menor consumo de potencia.
- Se logró mejorar la polarización de los transistores a través de la etapa de inversión, ya que los inversores utilizados presentan un rango de operación en sus niveles lógicos de salida adecuados para el correcto funcionamiento de los transistores.
- Se verifica que el sistema implementado presenta una actualización dinámica de la imagen, sin presentar latencias o retardos significativos entre los cambios realizados en el computador y la visualización en la matriz de LEDs.
- Se concluye que el modo de transferencia de datos en masa (Bulk) del protocolo USB es apropiado para el envío de imágenes de baja resolución en tiempo real, pues este modo presenta una buena velocidad de transferencia y tiene métodos para verificar la integridad de los datos.



- Se logró configurar el reconocimiento de un dispositivo externo y comunicación mediante protocolo USB a través de manipulación de clases en lenguaje *JAVA*.



6. OBSERVACIONES Y TRABAJO FUTURO

En la elaboración del proyecto se tuvo en cuenta que al momento de extraer la imagen tendría que ser desde la memoria RAM, ya que esta permite un fácil acceso por métodos ya prediseñados y no presenta latencias o retardos como otros dispositivos (e.g. disco duro).

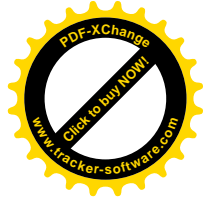
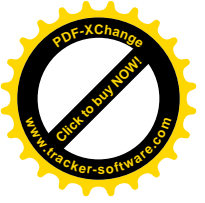
Otra razón a tener en cuenta es el problema de las rutas, ya que en diferentes computadores los nombres de las rutas cambian y esto genera un error de concordancia entre la ruta utilizada por el código y las disponibles en el computador.

Se observó que para tener un mejor nivel de detalle de la imagen, la resolución de la matriz debe aumentar. Esto conlleva a una serie de problemas, entre los cuales está aumentar el tamaño de la multiplexación, ya que se necesitaría realizarla tanto en filas como en columnas, utilizar un microcontrolador con más puertos de salida y, se necesita una *PCB* más grande y con mayor cantidad de dispositivos lógicos para su implementación.

De la realización del proyecto se evidencia la importancia de temas relacionados con la programación orientada a objetos, como herramienta fundamental en contextos prácticos de aplicación de la integración de sistemas de hardware + software, necesarios en la formación de un ingeniero actual.

- Como trabajo futuro se propone:

Reemplazar la matriz de LEDs por una matriz de materiales electromecánicos que tenga como función recrear en alto relieve la imagen que se está mostrando en la matriz de LEDs. Según las pruebas que se han realizado, esta

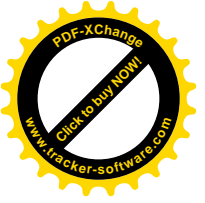


matriz debe presentar una resolución mínima 64x38 pines y cada pin debe tener una separación mínima de 1,5mm, ya que esta es la resolución espacial mínima de los receptores táctiles presentes en las puntas de nuestros dedos.

Para la construcción de la matriz se proponen dos materiales: micro-actuadores basados en solenoides o micro-actuadores basados en Aleaciones con Memoria de Forma, SMA por sus siglas en inglés (*Shape Memory Alloys*). A continuación se describen las principales características de estos materiales.

Los micro actuadores basados en solenoides [17] han sido los más utilizados y por ende los que más han sido investigados y mejorados desde su aparición. Actualmente, estos micro-actuadores presentan las siguientes características:

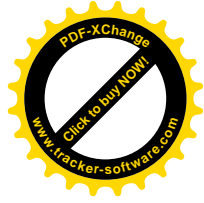
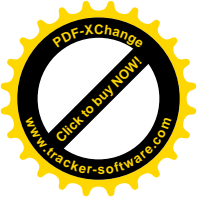
- El tamaño se ha reducido considerablemente llegando a poder construirse matrices con una separación de 1,5 [mm] entre pines, donde cada pin tiene un diámetro de tan solo 0,5 [mm], y con una altura de unos 2.5 [mm].
- El desplazamiento vertical que se logra con este sistema varía de acuerdo al tamaño del solenoide, por ejemplo, para el caso de los micro-actuadores mencionados anteriormente solo se logran desplazamientos mayores a los 0.2 [mm].
- El tiempo de respuesta de los micro-actuadores de este tipo más pequeños que se fabrican es del orden de los milisegundos.
- La fuerza de subida que entregan los micro-actuadores basados en solenoides es de hasta 60[mN].
- Se reducido considerablemente el consumo de potencia debido a que se ha implementado un sistema donde se han acoplado un solenoide, un imán permanente y un resorte elástico.
- El costo de fabricación de estos micro-actuadores es algo elevado, debido principalmente al grado de precisión y la dificultad que conlleva fabricar un dispositivo de este tamaño.



Los micro-actuadores basados en SMA [19] son una nueva alternativa que ha motivado a muchas investigaciones en los últimos años, y que debido a esto, ha alcanzado un alto grado de desarrollo a pesar de ser una tecnología relativamente nueva. En la actualidad, el micro-actuador basado en esta tecnología más pequeño que se ha desarrollado presenta las siguientes características:

- Tiene un diámetro de 1.5 [mm] y una longitud de 45 [mm]
- Puede entregar un desplazamiento vertical de hasta 3 [mm]
- El tiempo de respuesta de subida del pin varía de acuerdo a la corriente que se le suministre, por ejemplo, para 500 [mA] es de 0.26 segundos y para 200 [mA] es de 1.46 segundos. El tiempo que tarda el pin en volver a su posición original depende en gran medida al sistema enfriamiento que se utilice, por ejemplo, si se enfría mediante corrientes de aire suministrados por un ventilador se puede tener una latencia en el tiempo de respuesta de 70 [ms].
- La fuerza de subida que entrega este micro-actuador es de unos 120 [mN].
- El consumo de potencia por pin es inversamente proporcional al tiempo de respuesta, por lo que de acuerdo a los requerimientos del diseño se podría utilizar la relación más óptima entre consumo de potencia y tiempo de respuesta. Además de la potencia consumida por cada pin, se debe tener en cuenta la potencia consumida por el sistema enfriamiento a implementar.
- Presenta problemas de histéresis que se deben tener en cuenta a la hora del diseño de un sistema basado en estos micro-actuadores.
- El costo de fabricación en el laboratorio de cada pin basado en esta tecnología es de unos 2 USD.

Consideramos que para el diseño de la matriz con pines electromecánicos se deben utilizar micro-actuadores basados en solenoides.



CITAS BIBLIOGRAFICAS

[1] DEPARTAMENTO ADMINISTRATIVO NACIONAL DE ESTADÍSTICA (DANE). Información estadística sobre discapacidad nacional, Junio 2010. [Citado en 2011-08-01]. Disponible en línea: http://www.dane.gov.co/files/investigaciones/discapacidad/Total_nacional.xls

[2] DEPARTAMENTO ADMINISTRATIVO NACIONAL DE ESTADÍSTICA (DANE). Información estadística sobre discapacidad en Santander, Junio 2010. [Citado en 2011-07-15]. Disponible en línea: <http://www.dane.gov.co/files/investigaciones/discapacidad/SANTANDER.xls>

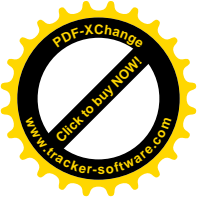
[3] COLOMBIA. CONGRESO DE LA REPÚBLICA. Ley 361. (Febrero, 1997). [Citado en 2011-07-23]. Disponible en línea: http://www.secretariassenado.gov.co/senado/basedoc/ley/1997/ley_0361_1997.html

[4] DOMÉNECH RIERA, Xavier. Historia de la Tiflotecnología en España. Enero 2010. [En línea]. Disponible en línea: <http://www.nosolousabilidad.com/articulos/tiflotecnologia.htm>

[5] FERNÁNDEZ, GARCÍA, Juan Rafael. Accesibilidad en Linux: TIFLOTECNOLOGÍA. En: Linux Magazine, Junio 2005, vol. 7, p. 82-86. [Citada en 2011-07-16]. Disponible en línea: <http://www.linux-magazine.es/issue/07/Educacion.pdf>

[6] KANDEL, Eric. Principles of Neural Science. Cuarta edición: McGraw-Hill/Appleton & Lange, 2000

[7] VIDAL VERDÚ, Fernando. Un equipo de la Universidad de Málaga desarrolla pantallas de ordenador táctiles para ciegos. Abril 2003. [Citada en 2011-07-18]. Disponible en línea: http://www.tendencias21.net/Un-equipo-de-la-Universidad-de-Malaga-desarrolla-pantallas-de-ordenador-tactiles-para-ciegos_a143.html



[8] UNIVERSIDAD DE MÁLAGA. Display táctil refrescable para gráficos y texto basado en actuadores termoneumáticos. Solicitud de patente España ES 2 204 273 A1, Abril 16, 2004. [Citada en 2011-07-18]. Disponible en línea: http://www.espatentes.com/pdf/2204273_a1.pdf

[9] TAKASAKI, Masaya, TAKAAKI, Nara y TAKESHI, Mizuno. A Progressive SAW Tactile Display On a PC Mouse Button. Int. Workshop on Robot and Human Interactive Communication, Berlin, Germany, Sept. 25-27, 2002.

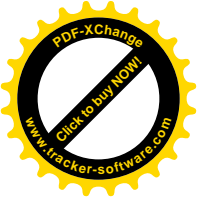
[10] TAKASAKI, Masaya, KOTANI, Hiroyuki y TAKESHI, Mizuno. Reproduction of Tactile Sensation Using SAW Tactile Display, Department of Mechanical Engineering, Saitama University, Japan, 2005.

[11] TAKAAKI, Nara, TAKASAKI, Masaya y MAEDA, Taro. Surface Acoustic Wave Tactile Display. En: *IEEE Computer graphics*, November/December 2001, Vol. 21, no. 6, pp. 56-63,.

[12] HOWE, Robert, KONTARINIS, Dimitrios y PEINE, William. Shape Memory Alloy Actuator Controller Design for Tactile Displays. 34th Conference on Decision and Control, New Orleans, LA, December 2002.

[13] CHAVES, Daniel, PEIXOTO, LIMA, Ana, VIEIRA, Fatima y DE ARAUJO, Carlos, Microactuator of SMA for Braille Display System. Int. Workshop on Medical Measurements and Applications, Cetraro, Italy, 2009.

[14] TAYLOR, P., MOSER, A. y CREED, A. The Design and Control of a Tactile Display Based on Shape Memory Alloys. Dept. of Electronic Engineering, The University of Hull, UK, 2002.



[15] MAKINO, Yasutoshi, ASAMURA, Naoya y SHINODA, Hiroyuki. Multi Primitive Tactile Display Based on Suction Pressure Control. 12th International Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems, 2004.

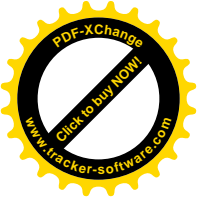
[16] OHKA M., SAWAMOTO Y., KAWABE, Y. y ABDULLAH, C. A Basic Study on Two-axial Piezoelectric Actuator for Tactile Display Using Illusions. International Conference on Advanced Intelligent Mechatronics, Montreal, Canada, 2010.

[17] OSPINA, José Alejandro. Imágenes Táctiles. Trabajo de grado Ingeniero electrónico. Bogotá DC: Pontificia Universidad Javeriana. Facultad de ingeniería, 2005.

[18] CÉTARES, Alfonso, CORTES, Carlos y SILVA, Luis. Sistema de Enseñanza del Código Braille para Niños con Limitaciones Visuales. Trabajo de grado Ingeniero electrónico. Bogotá DC: Pontificia Universidad Javeriana. Facultad de ingeniería, 2005.

[19] VELÁZQUEZ, Ramiro, PISSALOUX, Edwige y SZEWCZYK, Jérôme. Miniature Shape Memory Alloy Actuator for Tactile Binary Information Display. Laboratoire de Robotique de Paris, Fontenay aux Roses, France, CNRS FRE 2507, 2004. [Citada en 2011-07-22], Disponible en línea: <http://www.isir.upmc.fr/UserFiles/File/Jerome%20Szewczyk/velazquez.pdf>

[20] MORALES TORRES, Manuela, BERROCAL ARJONA, Manuel. Tiflotecnología y Material Tiflotécnico. I Congreso Virtual INTEREDVISUAL sobre Intervención Educativa y Discapacidad Visual, 2003. [Citada en 2013-03-10]. Disponible en línea:



http://www.juntadeandalucia.es/averroes/caidv/interedvisual/icv/tiflotecnologia_y_material_tiflotecnico_mym.pdf

[21] MINGUITO GARCÍA, Carlos, SERRANO MARUGÁN, Isabel. Intervención Específica con Alumnos con Discapacidad Visual: Criterios Diagnósticos y de Escolarización. Elementos de Acceso al Curriculum. Encuentro Anual de Equipos de Orientación Educativa, Madrid, España, noviembre de 2010. [Citada en 2013-03-10]. Disponible en línea: http://www.madrid.org/dat_capital/upe/impresos_pdf/Visuales_010.pdf

[22] FREESCALE Semiconductors™. Hoja de datos de Microcontrolador MC9S08JM60, 2009. [Citada en 2013-03-18]. Disponible en: <http://www.freescale.com/>

[23] LIU, Derek. USB Device Development with the MC9S08JM60. FREESCALE Semiconductor™, 2009. [Citada en 2013-03-19]. Disponible en: <http://www.freescale.com/>

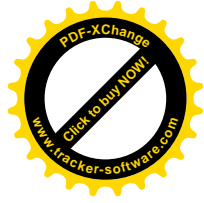
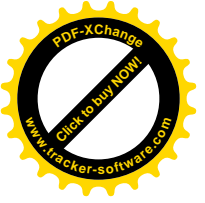
[24] LITE-ON, LED Specialist, Hoja de datos de la Matriz de LEDs, LTP-2558AA, DOT MATRIX DISPLAYS. [Citada en 2013-02-24] Disponible en: <http://doc.chipfind.ru/liteon/ltp2558aa.htm>

[25] TEXAS INSTRUMENTS, SN74AHCT16240, BUFFERS/DRIVERS, TEXAS INSTRUMENTS WIDEBUS™ FAMILY, 2000. [Citada en 2013-03-07]. Disponible en: <http://www.ti.com/>

[26] TEXAS INSTRUMENTS, SN74LV138A, DECODERS/DEMULTIPLEXERS, 2005. [Citada en 2013-03-05]. Disponible en: <http://www.ti.com/>

[27] CaRMetal, Página de inicio [Citada en 2013-02-18]. Disponible en: http://db-maths.nuxit.net/CaRMetal/index_es.html

[28] LibUsb, Página de inicio [Citada en 2013-02-18]. Disponible en: <http://www.libusb.org/>



BIBLIOGRAFIA

CÉTARES, Alfonso, CORTES, Carlos y SILVA, Luis. Sistema de Enseñanza del Código Braille para Niños con Limitaciones Visuales. Trabajo de grado Ingeniero electrónico. Bogotá DC: Pontificia Universidad Javeriana. Facultad de ingeniería, 2005.

KANDEL, Eric. Principles of Neural Science. Cuarta edición: McGraw-Hill/Appleton & Lange, 2000

LIU, Derek. USB Device Development with the MC9S08JM60. FREESCALE Semiconductor™, 2009. [Citada en 2013-03-19]. Disponible en: <http://www.freescale.com/>

MINGUITO GARCÍA, Carlos, SERRANO MARUGÁN, Isabel. Intervención Específica con Alumnos con Discapacidad Visual: Criterios Diagnósticos y de Escolarización. Elementos de Acceso al Curriculum. Encuentro Anual de Equipos de Orientación Educativa, Madrid, España, noviembre de 2010. [Citada en 2013-03-10]. Disponible en línea: http://www.madrid.org/dat_capital/upe/impresos_pdf/Visuales_010.pdf

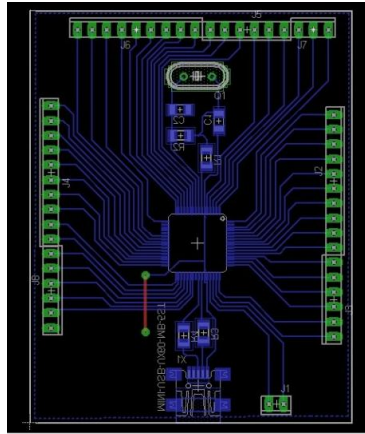
TAKASAKI, Masaya, TAKA AKI, Nara y TAKESHI, Mizuno. A Progressive SAW Tactile Display On a PC Mouse Button. Int. Workshop on Robot and Human Interactive Communication, Berlin, Germany, Sept. 25-27, 2002.

VIDAL VERDÚ, Fernando. Un equipo de la Universidad de Málaga desarrolla pantallas de ordenador táctiles para ciegos. Abril 2003. [Citada en 2011-07-18]. Disponible en línea: http://www.tendencias21.net/Un-equipo-de-la-Universidad-de-Malaga-desarrolla-pantallas-de-ordenador-tactiles-para-ciegos_a143.html

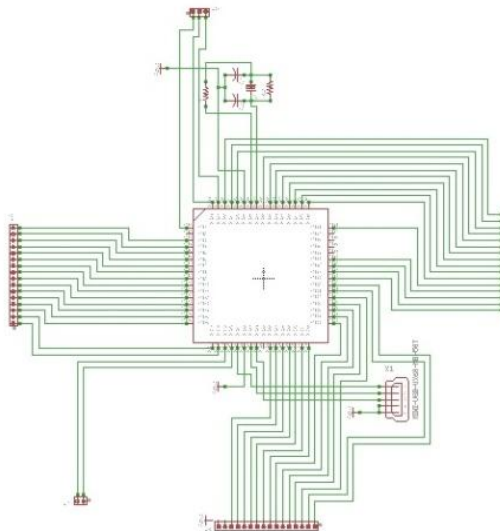
ANEXOS

ANEXO A

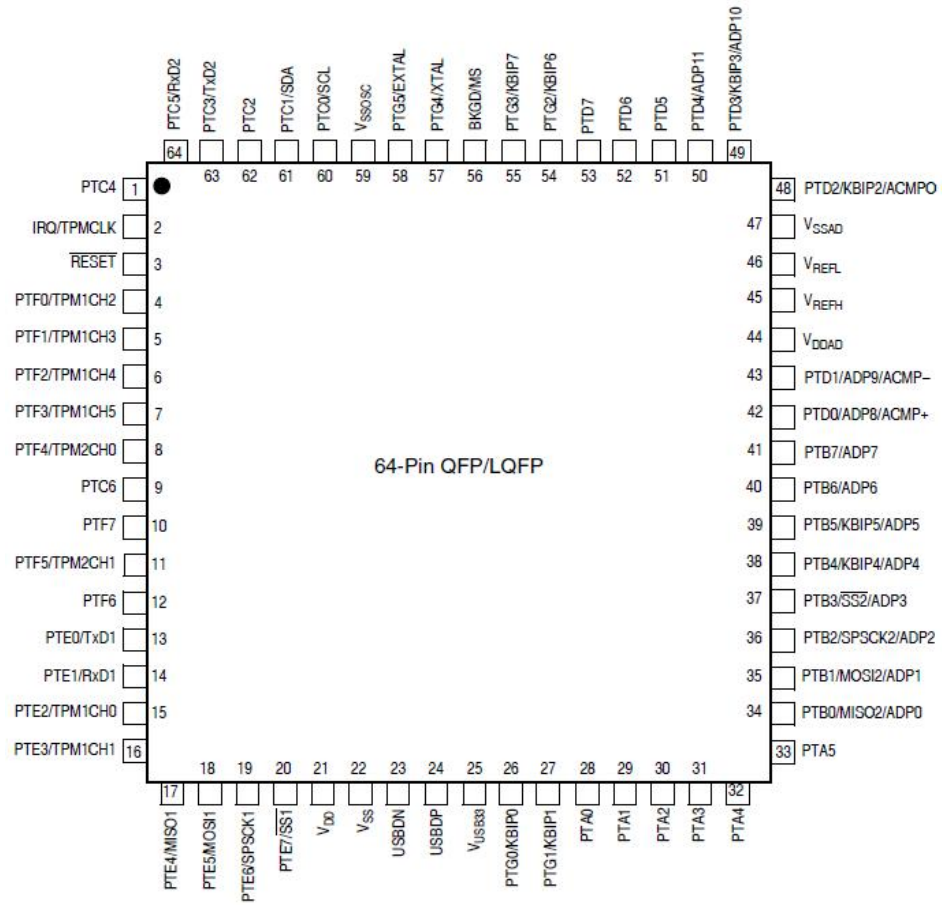
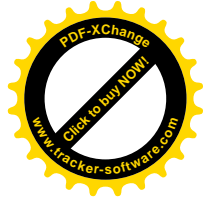
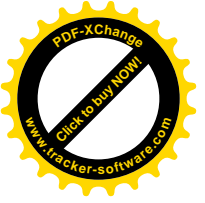
A.1. PCB del microcontrolador MC9S08JM60 en Eagle

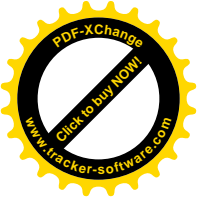


A.2. Esquemático de la PCB del microcontrolador en Eagle



A.3. Distribución de pines del MC9S08JM60



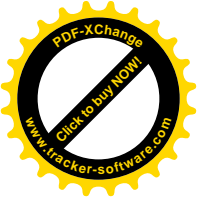


ANEXO B

B.1. Instalación driver LibUSB

Una vez descargados los archivos fuente se procede a instalar los controladores. Una vez instalados se asegura que los siguientes archivos estén en su correcta ubicación: *libusb0.dll* debe estar en `Windows\System32\`; *libusb0.sys* debe estar en `Windows\System32\drivers\` y *Libusbjava.dll* debe estar en `Windows\System32\`. Una vez estén estos archivos en sus correctas ubicaciones se procede a crear el archivo *INF* para el dispositivo el cual se quiere interactuar con *JAVA*, en nuestro caso el micro controlador. Para esto se utiliza un archivo ejecutable que se puede encontrar dentro de las librerías descargadas del programa (*inf-wizard.exe*) después de haber instalado el dispositivo para que trabaje con la librería de *JAVA*. Ahora se le hace una prueba para corroborar si la instalación fue exitosa y es la siguiente: se ejecuta el archivo llamado *testlibusb-win.exe*; si la instalación fue correcta allí deberá aparecer toda la información correspondiente al dispositivo. Cabe aclarar que se debe tener en cuenta que la arquitectura del sistema operativo concuerde con la de los controladores a instalar y esto se obtiene descargando la versión de 32 bits o 64 bits dependiendo la arquitectura del sistema.

Como paso final se debe agregar la librería de Java a las librerías nativas de *JAVA* las cuales son llamadas por el programa *NETBEANS*. La librería se llama *ch.ntb.usb-0.5.9.jar*. Aquí están contenidos todos los comandos o métodos necesarios para poder implementar el envío de información a puertos USB desde un código en *JAVA*.



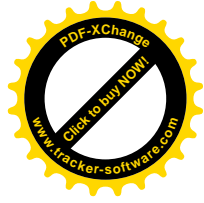
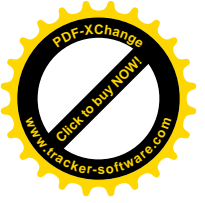
B.2. Descripción del protocolo USB

El protocolo USB (Bus Serial Universal, por sus siglas en inglés), fue creado en el año 1996 por siete empresas: IBM, Northern Telecom, Compaq, Intel, Microsoft, Digital Equipment Corporation y NEC. Los principales objetivos que se querían cumplir con este estándar eran permitir la conexión de periféricos en una sola interfaz de socket estandarizada, mejorar la capacidad de “plug and play” y suministrar a través del mismo puerto energía a dispositivos de bajo consumo. Desde su aparición en el mercado se han desarrollado cuatro versiones de este estándar:

- USB 1.0 o *Low-speed*, con tasas de transferencia de hasta 1.5 Mbps (192 KB/s). Se usa en teclados, ratones y joysticks.
- USB 1.1, *Full-speed*, con tasas de transferencia de hasta 12 Mbps (1.5 MB/s).
- USB 2.0, *High-speed*, con tasas de transferencia de hasta 480 Mbps (60 MB/s).
- USB 3.0, con tasa de transferencia de hasta 4.8 Gbps (600 MB/s).

Dentro del estándar USB existen algunos términos especiales que es necesario definir para tener una mejor comprensión de la forma como funciona este protocolo. A continuación se definirán estos términos:

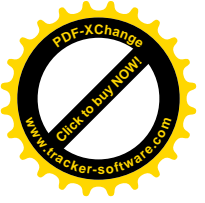
- Punto terminal: este término es usado para describir un punto o búfer donde los datos entran o dejan un sistema USB. Un punto terminal tipo *IN* (punto terminal de entrada) es un creador de datos, y un punto terminal *OUT* (punto terminal de salida) es un consumidor de datos. Se debe tener en cuenta que la dirección de los datos es relativa al PC anfitrión (*host*).



- Interfaz: Una conexión del mundo real típica puede tener muchos puntos terminales de entrada y/o salida para implementar un esquema de distribución de datos confiable. Esta colección de datos es denominada una *interfaz* y está directamente relacionada con una conexión del mundo real. El sistema operativo que tenga instalado el PC al que se conecte el dispositivo debe tener un controlador de software que corresponde a cada interfaz.
- *Pipe*: Dentro del sistema operativo se usa el término *pipe* para describir la conexión lógica entre un controlador de software en el PC anfitrión y la interfaz en el dispositivo USB.
- Configuración: Una colección de interfaces es llamada una *configuración*, y solo una configuración puede estar activa a la vez. Una configuración define los atributos y las características de un modelo específico.

El protocolo USB transmite los datos a través de paquetes, existen paquetes especiales denominados descriptores, que contienen instrucciones para configurar el dispositivo. Estos descriptores se envían a través del punto terminal 0 (cero) el cual se encarga transmitir todos los datos de configuración. Existen cuatro tipos de descriptores:

- Descriptor de dispositivo: Contiene información básica del dispositivo.
- Descriptor de configuración: Representan el estado del dispositivo (activo, suspendido, inicialización) y su configuración básica (potencia, número de interfaces, etc).
- Descriptor de interfaz: Contiene el identificador de interfaz, número de puntos terminales para cada interfaz, etc. Un dispositivo USB puede consistir en varios sub-dispositivos lógicos (cada uno tendrá una interfaz). Es necesario un driver por cada interfaz.
- Descriptor de punto terminal: Describe los puntos terminales distintos del cero.

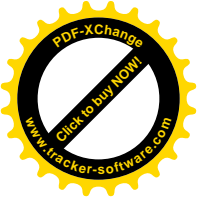


Además, el protocolo USB maneja cuatro tipos de transferencia de datos y la selección del tipo de transferencia que se quiere utilizar se hace mediante el descriptor del punto terminal. Los modos de transferencia que maneja el estándar USB son los siguientes:

- De control: Los puntos terminales de control son usados para controlar el dispositivo USB asincrónicamente. Todos los dispositivos deben tener un punto terminal 0 (cero), que debe ser de control, y que es usado para inicializar el periférico al conectarlo.
- De interrupción: El host solicita periódicamente pequeños paquetes de datos de tamaño fijo al dispositivo. El dispositivo debe esperar a ser atendido. Este tipo de transferencia se usa normalmente para ratones y teclados.
- *Bulk*: Con este tipo de transferencia se pueden transmitir grandes cantidades de datos. Con las transferencias de modo *bulk* se asegura que los datos siempre llegan a su destino. Usualmente se usa para almacenamiento de datos o impresoras.
- Isócrona: Con este modo se transfieren grandes cantidades de datos. Con este tipo de transferencia no es seguro que los datos lleguen a su destino. Se usa frecuentemente en aplicaciones de tiempo real como video o audio.

Cuando se conecta un dispositivo USB a una PC se produce una secuencia denominada *Proceso de Enumeración*, el cual consiste en que el host le solicita al dispositivo que se presente y le diga cuáles son sus parámetros, tales como:

- *Vendor ID* y *Product ID*, que se encuentran en el descriptor de dispositivo.
- Consumo de energía expresada en unidades de carga.
- Clase del producto.
- Número y tipos de puntos terminales.



- Tipo de transferencia.

El proceso de enumeración es inicializado por el *host* cuando detecta que un nuevo dispositivo ha sido conectado al bus. El *host* le asigna una dirección al dispositivo que se ha conectado y habilita su configuración, a través del controlador adecuado, permitiendo la transferencia de datos sobre el bus.