



ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO
FACULTAD DE INFORMÁTICA Y ELECTRÓNICA
INGENIERÍA ELECTRÓNICA EN CONTROL Y REDES INDUSTRIALES

**”IMPLEMENTACIÓN DE UN ROBOT CARTESIANO, PARA LA
ELABORACIÓN DE FIGURAS EN RELIEVE ENFOCADO A
PERSONAS CON DISCAPACIDAD VISUAL.”**

TRABAJO DE TITULACIÓN

TIPO: DISPOSITIVO TECNOLÓGICO

Previo a la obtención del título:

**INGENIERO EN ELECTRÓNICA, CONTROL Y REDES
INDUSTRIALES**

AUTOR: ANGEL RODRIGO MUNGABUSI SISA

TUTOR: ING. JOSÉ ENRIQUE GUERRA SALAZAR

Riobamba – Ecuador

2018

©2018, Angel Rodrigo Mungabusi Sisa

Se autoriza la reproducción total o parcial, con fines académicos, por cualquier medio o procedimiento, incluyendo a la cita bibliográfica del documento, siempre y cuando se reconozca el Derecho de Autor.

ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO
FACULTAD DE INFORMÁTICA Y ELECTRÓNICA
ESCUELA DE INGENIERÍA ELECTRÓNICA EN CONTROL Y REDES
INDUSTRIALES

El Tribunal de Trabajo de Titulación certifica que: Dispositivo tecnológico: “IMPLEMENTACIÓN DE UN ROBOT CARTESIANO, PARA LA ELABORACIÓN DE FIGURAS EN RELIEVE ENFOCADO A PERSONAS CON DISCAPACIDAD VISUAL” de responsabilidad del señor ANGEL RODRIGO MUNGABUSI SISA, ha sido minuciosamente revisado por los Miembros del Tribunal de Trabajo de Titulación, quedando autorizada su presentación.

| NOMBRE | FIRMA | FECHA |
|---|--------------|--------------|
| Dr. Julio Roberto Santillán Castillo. VICEDECANO DE LA FACULTAD DE INFORMÁTICA Y ELECTRÓNICA. | | |
| Ing. Freddy Enrique Chávez Vásquez. DIRECTOR DE LA ESCUELA DE INGENIERÍA ELECTRÓNICA EN CONTROL Y REDES INDUSTRIALES. | | |
| Ing. José Enrique Guerra Salazar. DIRECTOR DE TRABAJO DE TITULACIÓN. | | |
| Ing. Marco Antonio Viteri Barrera. MIEMBRO DEL TRIBUNAL. | | |

“Yo, **ANGEL RODRIGO MUNGABUSI SISA**, declaro ser responsable de las ideas, doctrinas y resultados expuestos en este Trabajo de titulación, y el patrimonio intelectual de la misma pertenece a la **ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO**”.

Angel Rodrigo Mungabusi Sisa

DEDICATORIA

Este trabajo de titulación el cual se ha realizado como mucho esfuerzo ha sido impulsado por el fuerza de mis familiares en especial de mi padre Prudencio Mungabusi quien ha sido el apoyo incondicional a lo largo de mi carrera de estudiante, mi Madre Esthela Sisa que con sus consejos me motivo día a día para lograr cada una de las metas que me he propuesto tomo mi cariño y respeto se ve reflejado en un agradecimiento enorme por todo el esfuerzo que tuvieron que hacer para poder culminar mi carrera. Las palabras no puede expresar todo el amor y sobre todo la motivación que significa en esta última etapa de mi carrera mi hijo Adrián Mungabusi, que con el hecho de su presencia y su vitalidad me motivaron a no desmayar en este trajinar. Este trabajo de titulación se ha convertido para mí en más que un solo trámite para poder alcanzar el título universitario comprendí que el proyecto va más allá de ayudar personas que tiene discapacidad visual sino que puede cambiar vidas de los niños y jóvenes en etapa escolar que tanto lo necesitan. Estoy seguro que con esfuerzo y perseverancia llevar este proyecto social a las manos que realmente lo requieren. Sin más un agradecimiento enorme a quienes de alguna forma influyeron sobre mí de manera positiva para ellos nada más que decir que un Dios le pague y Gracias Totales.

Angel

AGRADECIMIENTO

Mi Padre y Madre quienes con su apoyo incondicional me han llevado a culminar mi carrera profesional, a toda mi familia que sin sus palabras este logro no se hubiese cristalizado.

Quiero agradecer con el mismo ahínco a las personas con discapacidad visual que me supieron acoger para poder realizar todo el trabajo mismo que espero sea en poco tiempo de mucho provecho para ellos. Quienes fueron parte en mi vida importantes seguro no poder ponerlos en esta pequeña página pero ellos saben cuánto los aprecio y los estimo con sus palabras han dejado huella y siempre estarán en el fondo de mi corazón, es por eso que le agradezco y darles un abrazo enorme aunque sea a la distancia y este agradecimiento es para todos ustedes.

Angel

TABLA DE CONTENIDO

| | |
|--|-------------|
| RESUMEN | xxiv |
| ABSTRACT | xxv |
| INTRODUCCION | 1 |
| CAPÍTULO I | |
| 1. MARCO TEÓRICO | 4 |
| 1.1. Técnicas de elaboración de figuras en relieve | 4 |
| 1.2. Sistemas comerciales basados en control numérico computarizado (CNC) | 6 |
| <i>1.2.1. Sistemas electrónicos de elaboración de gráficos tangibles.</i> | <i>6</i> |
| 1.3. Robot cartesiano | 8 |
| 1.4. Editor de gráficos vectoriales | 9 |
| 1.5. G-code | 9 |
| <i>1.5.1.1. Marlin (Interprete de G-code)</i> | <i>10</i> |
| <i>1.5.1.2. Host</i> | <i>11</i> |
| 1.5.2. Placas de desarrollo | 11 |
| <i>1.5.2.1. Placas de circuitos modulares (SHIELD)</i> | <i>12</i> |
| 1.5.3. Motor pasó a paso (PAP) | 13 |
| 1.5.4. Servomotor | 14 |
| 1.5.5. Bombas de vacío | 15 |
| 1.5.6. Controlador de motor PAP A4988 | 15 |
| 1.5.7. Solenoide | 16 |
| 1.5.8. Sensor de contacto | 17 |
| 1.5.8. Sensor óptico reflexivo | 17 |
| 1.6. Normativa INEN 2850 | 18 |
| CAPÍTULO II | |
| 2. MARCO METODOLÓGICO | 19 |
| 2.1. Requerimientos para la implementación del Prototipo. | 19 |
| 2.2. Concepción general del sistema del prototipo | 19 |

| | | |
|---------------------|--|-----------|
| 2.3 | Diseño de la arquitectura del prototipo | 21 |
| 2.4 | Selección de los elementos para el prototipo..... | 21 |
| 2.4.1 | <i>Arduino Mega 2560.....</i> | <i>21</i> |
| 2.4.2 | <i>Sensores para el prototipo.....</i> | <i>22</i> |
| 2.4.3 | <i>Shield Ramps 1.4.....</i> | <i>24</i> |
| 2.4.4 | <i>Diseño de circuitos para el prototipo</i> | <i>25</i> |
| 2.4.5 | <i>Selección de actuadores</i> | <i>29</i> |
| 2.4.5.1 | <i>Motor de pasos Nema17.....</i> | <i>29</i> |
| 2.4.5.2 | <i>Solenoides como punzón Braille.....</i> | <i>30</i> |
| 2.4.5.3 | <i>Selección de mini compresor de aire</i> | <i>31</i> |
| 2.5 | Selección alimentación para el prototipo | 33 |
| 2.6 | Esquema de conexión para el prototipo | 34 |
| 2.7 | Software para el prototipo..... | 36 |
| 2.7.1 | <i>Requerimientos de software por el prototipo</i> | <i>36</i> |
| 2.7.2 | <i>Generación de comandos (Gcode) con Inkscape</i> | <i>37</i> |
| 2.7.3 | <i>Software para el manejo del prototipo Repetier-Host</i> | <i>39</i> |
| 2.7.4 | <i>Diagrama de flujo del intérprete de comandos Marlin del prototipo</i> | <i>40</i> |
| 2.7.5 | <i>Diagrama de flujo del proceso de impresión del prototipo</i> | <i>41</i> |
| 2.8 | Diseño mecánico para el Prototipo | 43 |
| 2.8.1 | <i>Estructura mecánica cartesiana para la implementación del prototipo</i> | <i>44</i> |
| 2.8.3 | <i>Diseño y simulación de la estructura mecánica en SOLIDWORKS 2016</i> | <i>44</i> |
| CAPÍTULO III | | |
| 3. | ANÁLISIS DE RESULTADOS DEL PROTOTIPO | 47 |
| 3.1 | Pruebas de impresión del prototipo..... | 47 |
| 3.1.1 | <i>Prueba de gráficos en relieve.....</i> | <i>47</i> |
| 3.1.2 | <i>Prueba de impresión de escritura braille</i> | <i>54</i> |
| 3.1.3 | <i>Prueba de estabilidad del prototipo</i> | <i>63</i> |
| 3.1.4 | <i>Prueba de legibilidad de escritura Braille y figuras en relieve.....</i> | <i>66</i> |

| | | |
|-------|--|----|
| 3.1.5 | <i>Prueba de comparación de figura patrón con la resultante.....</i> | 67 |
| 3.2 | Tiempo de impresión de imágenes en relieve..... | 68 |
| 3.3 | Puesta en marcha del prototipo general..... | 70 |
| 3.4 | Potencia de consumo del prototipo..... | 71 |
| 3.5 | Análisis económico del prototipo | 71 |
| | Conclusiones | 73 |
| | Recomendaciones | 74 |

BIBLIOGRAFIA

ANEXOS

ÍNDICE DE TABLAS

| | | |
|-------------------|--|----|
| Tabla 1-1: | G-code comandos..... | 10 |
| Tabla 2-1: | Modelos de bombas de vacío..... | 15 |
| Tabla 1-2: | Características del Arduino Mega 2560..... | 22 |
| Tabla 2-2: | Características principales del sensor TCRT5000L..... | 23 |
| Tabla 3-2: | Características principales del sensor de contacto..... | 24 |
| Tabla 4-2: | Principales características de la Ramps 1.4..... | 25 |
| Tabla 5-2: | Características principales del motor de pasos NEMA17..... | 30 |
| Tabla 6-2: | Características técnicas del solenoide elegido..... | 31 |
| Tabla 7-2: | Especificaciones técnicas de mini compresor de aire..... | 32 |
| Tabla 8-2: | Datos técnicos de ventosa..... | 34 |
| Tabla 1-3: | Tiempos de impresión de imágenes en relieve y texto..... | 64 |
| Tabla 2-3: | Análisis de las muestras tomadas de símbolos braille para el estudio estadístico.. | 66 |
| Tabla 3-3: | Evaluación de legibilidad..... | 67 |
| Tabla 4-3: | Tiempos de impresión de imágenes en relieve y texto..... | 68 |
| Tabla 5-3: | Potencia de consumo total de energía eléctrica..... | 71 |
| Tabla 6-3: | Costo total del prototipo..... | 72 |

ÍNDICE DE FIGURAS

| | | |
|---------------------|---|----|
| Figura 1-1: | Sistema Empastado..... | 6 |
| Figura 2-1: | Sistema de microencapsulado..... | 6 |
| Figura 3-1: | Sistema de termoformado..... | 7 |
| Figura 4-1: | BASSIC-D V5..... | 8 |
| Figura 5-1: | VISION 1624 ROUTER..... | 8 |
| Figura 6-1: | Robot Cartesiano..... | 9 |
| Figura 7-1: | G-code y su interpretación..... | 10 |
| Figura 8-1: | Modelos de placas de desarrollo Arduino para realizar robots cartesianos..... | 12 |
| Figura 9-1: | Placas de circuito modular (SHIELD)..... | 13 |
| Figura 10-1: | Motor paso a paso bipolar de la marca Polulo..... | 14 |
| Figura 11-1: | Servomotor..... | 14 |
| Figura 12-1: | Controlador de motor PAP A4988..... | 16 |
| Figura 13-1: | Solenoides lineales..... | 16 |
| Figura 14-1: | Sensores de contacto..... | 17 |
| Figura 15-1: | Sensor óptico reflexivo..... | 17 |
| Figura 16-1: | Ancho y alto de celda para símbolos braille..... | 18 |
| Figura 17-1: | Símbolo de teléfono con escritura Braille..... | 18 |
| Figura 1-2: | Concepción general del sistema de prototipo..... | 20 |
| Figura 2-2: | Diseño de la arquitectura del prototipo..... | 21 |
| Figura 3-2: | Arduino Mega 2560..... | 22 |
| Figura 4-2: | Sensor Óptico..... | 23 |
| Figura 5-2: | Sensor de contacto..... | 24 |
| Figura 6-2: | Shield Ramps 1.4..... | 24 |
| Figura 7-2: | Diagrama del circuito para la bomba de vacío y sensor óptico..... | 26 |
| Figura 8-2: | Diagrama del circuito de control de solenoide..... | 29 |
| Figura 9-2: | Motor a pasos Nema17..... | 29 |
| Figura 10-2: | Solenoides seleccionados..... | 30 |
| Figura 11-2: | Bomba de vacío seleccionada..... | 31 |
| Figura 12-2: | Ventosa modelo VC-33 A 1/8 de la marca VI-CAS..... | 32 |
| Figura 13-2: | Fuente de alimentación para el prototipo..... | 33 |
| Figura 14-2: | Conexión inicial de Ramps 1.4 sobre Arduino mega 2560..... | 34 |
| Figura 15-2: | Esquema de conexión para el prototipo..... | 35 |
| Figura 16-2: | Función añadida llamada Generate Braille Gcode..... | 36 |
| Figura 17-2: | Ventana de configuración de parámetros para la impresión..... | 38 |

| | | |
|---------------------|--|----|
| Figura 18-2: | Interfaz de Repetier-Host..... | 39 |
| Figura 19-2: | Ventana de control manual del prototipo..... | 40 |
| Figura 20-2: | Diagrama de flujo interprete de comandos Marlin..... | 41 |
| Figura 21-2: | Diagrama de flujo de la ejecución de comandos..... | 43 |
| Figura 22-2: | Estructura mecánica cartesiana pórtico horizontal..... | 42 |
| Figura 23-2: | Acople de varilla en SolidWorks 2016..... | 44 |
| Figura 24-2: | Simulación de la estructura mecánica del prototipo en SolidWorks 2016..... | 45 |
| Figura 25-2: | Impresora 3d Ultimachine Mendelmax2..... | 45 |
| Figura 1-3: | Implementación del prototipo final..... | 46 |
| Figura 2-3: | Gráfico digital y posterior impresión en relieve..... | 47 |
| Figura 3-3: | Toma de medida de ancho total entre puntos en relieve..... | 47 |
| Figura 4-3: | Medida de ancho de centro de puntos con calibrador universal..... | 49 |
| Figura 5-3: | Medida de alto de relieve con calibrador universal..... | 50 |
| Figura 6-3: | Medida de ancho de base de punto con calibrador universal..... | 52 |
| Figura 7-3: | Imagen con descripción en braille..... | 53 |
| Figura 8-3: | Símbolos Braille generado por el prototipo..... | 53 |
| Figura 9-3: | Medida de distancia de centro de punto horizontal con calibrador universal..... | 54 |
| Figura 10-3: | Medida de distancia vertical entre centros de puntos tomados por el calibrador universal..... | 55 |
| Figura 11-3: | Toma de medida de centro de puntos idénticos celdas contiguas con calibrador universal..... | 57 |
| Figura 12-3: | Medida de base de puntos toma de muestras con calibrador universal..... | 58 |
| Figura 13-3: | Medida de relieve de puntos con calibrador universal..... | 59 |
| Figura 14-3: | Ancho y alto de la celda generadora del sistema braille..... | 60 |
| Figura 15-3: | Toma de muestras del ancho del símbolo braille..... | 63 |
| Figura 16-3: | Impresión de imagen en relieve y texto braille..... | 66 |
| Figura 17-3: | Puesta en marcha del prototipo..... | 67 |
| Figura 18-3: | Sobre posición de imágenes a tinta con relieve..... | 68 |
| Figura 19-3: | Impresión de imagen en relieve y texto braille..... | 69 |
| Figura 20-3: | Puesta en marcha del prototipo..... | 70 |

ÍNDICE DE GRÁFICOS

| | | |
|----------------------|--|----|
| Gráfico 1-3: | Relación de medidas de ancho entre puntos respecto a límites superior e inferior de la normativa INEN..... | 47 |
| Gráfico 2-3: | Relación de medidas distancia entre puntos contiguos..... | 49 |
| Gráfico 3-3: | Relación de medidas respecto al mínimo del relieve regida por normativa INEN..... | 50 |
| Gráfico 4-3: | Medidas de la base de puntos respecto a la normativa INEN 2850..... | 52 |
| Gráfico 5-3: | Medidas de centro de punto horizontal respecto de límites regidos por normativa..... | 55 |
| Gráfico 6-3: | Medidas de centro de punto vertical respecto a límites regidos por normativa..... | 57 |
| Gráfico 7-3: | Medidas centro de punto celdas contiguas respecto a los límites de normativa INEN..... | 59 |
| Gráfico 8-3: | Diámetro de la base de puntos en relación a límites de la normativa INEN 2850... | 60 |
| Gráfico 9-3: | Medidas de relieve de punto con respecto al mínimo permitido por norma..... | 62 |
| Gráfico 10-3: | Resultado de mediciones de ancho de celta..... | 64 |
| Gráfico 11-3: | Resultado de medidas del alto de la celda con respecto a límites de normativa... | 65 |

ÍNDICE DE ABREVIATURAS

| | |
|---------------|---|
| CNC: | Control Numérico Computarizado |
| USB: | Bus universal en serie |
| INEN: | Instituto Ecuatoriano de Normalización |
| EE.UU: | Estados Unidos de América |
| GRBL: | Interprete de Gcode |
| GLPv3: | Licencia publica general |
| SD: | Secure digital memoria flash |
| PWM: | Modulación por ancho de pulso |
| PNG: | Portable Network Graphics gráfico de red portables |
| SVG: | Gráficos vectoriales escalables |
| JPEG: | Join Photographic Experts Group, grupo expertos en fotografía |
| DC: | Corriente directa |
| LCD: | Liquid Cristal Display |
| GND: | Conexcion a tierra |
| PSI: | Libra de fuerza por pulgada cuadrada |
| ATX: | Advanced Technology Extended |
| VAC: | Voltaje de corriente alterna |
| CAD: | Diseño asistido por computadora |

ÍNDICE DE ANEXOS

- Anexo A: Datasheet Arduino Mega**
- Anexo B: Datasheet sensor óptico TCRT5000L**
- Anexo C: Datasheet sensor de contacto**
- Anexo D: Esquema de conexión de Ramps 1.4**
- Anexo E: Datasheet del circuito integrado LM555**
- Anexo F: Datasheet del MOSFET IRFZ44N**
- Anexo G: Datasheet motor Nema17**
- Anexo H: Datos característicos de la micro compresor de vacío**
- Anexo I: Ventosa VC-33C de la empresa VI-CAS**
- Anexo J: Datasheet fuente de alimentación micro ATX**
- Anexo K: Código Fuente del plugin Braille Mungabusi**

RESUMEN

Se implementó un robot cartesiano para la elaboración de figuras en relieve, enfocado para personas con discapacidad visual, el prototipo pretende ser una alternativa de bajo costo a una impresora braille convencional. El proyecto intenta tener un impacto social positivo en docentes y padres de familia que tengan estudiantes con discapacidad visual al contar con una herramienta de ayuda para una educación personalizada. El prototipo permite realizar figuras y texto braille en hojas de papel de estándar A4 de 120 gramos, los elementos que conforman el proyecto el Host que es una computadora cargada de un programa que permite realizar el tratamiento de la imagen y simbología Braille y generar los comandos que a posterior se enviara por medio de comunicación USB hacia el robot. Un Arduino MEGA 2560 interpreta los comandos enviados por el Host y traslada todas las señales eléctricas a la parte actuante del prototipo. La generación del relieve es elaborada por un punzón, que basado en un solenoide genera la fuerza suficiente en el embolo el cual al golpear la hoja de papel para generar el relieve que conformara el elemento braille. De las pruebas realizadas se obtuvo que el consumo de potencia del prototipo implementado es de 55.65W, que representa un 28% de la potencia requerida por una impresora convencional. Se determinó que las figuras y escritura en braille elaboradas por el prototipo cumplen con la normativa INEN 2850 aplicadas a dispositivos de similares características. Por lo que se puede concluir que el prototipo implementado puede constituirse en una herramienta de ayuda para la elaboración de material didáctico braille similar a equipos comerciales. Es recomendable implementar un sistema de módulos compactos con protección plástica de alta resistencia que reduzca el peso, facilite su transportable y mejore su estética.

Palabras clave: <TECNOLOGIA DE CONTROL AUTOMATICO>, <ROBOT CARTESIANO>, <DISCAPACIDAD VISUAL>, <TEXTO BRAILLE>, <TRATAMIENTO DE IMAGENES <NORMA INEN>.

ABSTRACT

A cartesian robot was implemented to the elaboration of relief figures, focoused for people with visual disability, the prototype pretends to be a low cost alternative to one conventional braille printer. The project intents to get a positive social impact in teachers and parents who have visual disability students by having a tool for a personalized instrution. The prototype allows to create braile figures and text in on A4 standart sheets of paper of 120 grams , the elements which make up the project are the Host that is a computer which is loaded with a program that allows to do a treatment of the image and braille symbology, and generate the commands which later will be sent through the USB communication to the robot, an Arduino MEGA 2560 reads the commands that were sent by the Host and transfers all the electrical signals to the acting part of the prototype. The relief generation is elaborated using a punch, which based on a solenoid generates the enough strong in the plunger which to hit the sheet of paper to generate the relief that will conform the braille element. From the tests performed it was establish that the power consume of the implemented prototype is of 55.65 W, that represents a 28% of the power that a conventional printer needs. It was determined that, the braille shapes and writing, elaborated by the prototype conform to the INEN norm 2850 aplyed to devices with similar characteristics. Consequently it can conclude that the implemented prototype can be built as a aid tool for the elaboration od braille didactic material that is similar to commercial equipments. It is recommended to implement a compact modules system with plactic protection of high resistance that reduces the weight, facilitates its transportation and improves its aesthetics.

Key words: <AUTOMATIC CONTROL TECHNOLOGY>, <CARTESIAN ROBOT>, <VISUAL DISABILITY>, <BRAILLE TEXT>, <IMAGES TREATMENT>, <INEN NORM>.

INTRODUCCION

El desarrollo de la robótica permite realizar trabajos cada vez más complejos de ayuda para el ser humano ya no solamente en la industria, sino también como herramientas de servicio personal. Las tecnologías de código abierto (Open Source) tanto en hardware y software, permite el desarrollo e implementación de prototipos robóticos de bajo coste de aplicaciones diversas como: Impresoras 3D, Router CNC, Plotters de dibujo, robots cartesianos.

En los últimos años el avance en robótica se enfoca en la ayuda social, implementado sistemas que apoyen a personas con alguna discapacidad física, motora, sensorial, psíquica, intelectual o mental.

La discapacidad visual, quizá sea un reto de las nuevas tecnologías y poder brindar a estas personas la capacidad de llevar las experiencias del mundo a sus mentes. Los gráficos tangibles ayuda a la traducción de la información visual al relieve, es decir, se debe trasladar e interpretar los elementos visuales a un lenguaje admisible por el tacto, lo que se realiza mediante la elevación de los componentes o elementos del gráfico (Gual, Serrano, Manez, s/f, p. 2).

Las tecnologías desarrolladas a nivel mundial son diversas para poder ayudar a las personas con discapacidad visual, para poder hacer llegar la información visual al tacto, una de ellas es el prototipado rápido por impresoras 3D también están las impresoras braille y Router CNC.

JUSTIFICACIÓN TEÓRICA

El problema fundamental que enfrentan el diseño y elaboración de gráficas tangibles, son básicamente las técnicas de los procesos de producción mediante los cuales se puede confeccionar unas imágenes en relieve, se ha desarrollado una serie de sistemas que permiten realizar gráficos tangibles los más extendidos son: Estampado, termo-formado y micro-encapsulado, estas técnicas permiten realizar gráficos en relieve con materiales flexibles como el papel o laminas finas de plástico (Gual, Serrano, Mañez, s/f, p. 3).

En la ciudad de Ambato en la escuela para niños y jóvenes con discapacidad visual “Julius Doefner” se encuentran 35 estudiantes los cuales requieren de insumos en sistema braille e imágenes en relieve, luego de una entrevista se conoce que la impresora braille que poseen se encuentra dañada, y en el país no se encuentra soporte técnico, también se conoce que la única empresa que presta servicios de imprenta braille se encuentra en la ciudad de Riobamba y que requiere de tiempo de espera como de altos costos poder realizar pedidos de material didáctico para los estudiantes, esto hace difícil conseguir los materiales para poder dar a los estudiantes una enseñanza dinámica.

El trabajo realizado en esta investigación pretende implementar un sistema robótico cartesiano, para la elaboración de imágenes en relieve es una alternativa a los estudiantes de la región y el país de obtener de manera dinámica la información visual del entorno en que se rodean y también como de gráficos para la educación de las personas con discapacidad visual.

JUSTIFICACIÓN APLICATIVA

El vertiginoso avance de la tecnología permite el desarrollo de nuevos sistemas electrónicos automáticos en pro del beneficio social, con aplicaciones de ayuda para personas con algún tipo de discapacidad da a las personas la posibilidad de percibir y movilizarse en escenarios diversos en el mundo real y por ende mejorar la experiencia que tienen de su entorno.

El robot cartesiano para la elaboración de imágenes en relieve, el cual se basa en un control numérico computarizado, pretende apoyar a las personas con discapacidad visual facilita la obtención de material de apoyo en la educación de jóvenes y niños, así también una herramienta fundamental para poder percibir el entorno en el cual se desenvuelve las personas invidentes.

El sistema consta de una fase preliminar donde se realiza un tratamiento a la imagen dentro de un editor de gráficos vectoriales, donde se exporta el código G (G-code), el cual es enviado asía la tarjeta de desarrollo mediante un Host, la tarjeta interpreta los comandos mediante un firmware llamado Marlin que se encuentra alojado en la misma, esta traduce la información y las convierte en movimientos mecánicos de los motores paso a paso que están sujetas a una configuración cartesiana que permite movimientos lineales en el eje XY, misma que lleva un cabezal de impresión basado en un solenoide lineal para realizar los puntos que conforma el gráfico tangible.

OBJETIVOS

OBJETIVO GENERAL

- Implementar de un robot cartesiano, para la elaboración de figuras en relieve enfocado a personas con discapacidad visual.

OBJETIVOS ESPECÍFICOS

- Investigar como las figuras en relieve ayudan a las personas con discapacidad visual.
- Determinar los requerimientos que serán necesarias para implementar el robot cartesiano.
- Diseñar y realizar la estructura mecánica del robot cartesiano.
- Validar el correcto funcionamiento del sistema en general.

La memoria descriptiva del trabajo de investigación consta de tres capítulos conclusiones y recomendaciones. El primer capítulo aborda generalidades bibliográficas, el capítulo dos incorpora el diseño de hardware, software y mecánico de la implementación de un robot por último se tiene el tercer capítulo se da a conocer los resultados obtenidos de las diferentes pruebas al que fue sometido el prototipo adicionalmente se realiza un análisis económico del prototipo.

CAPÍTULO I

1. MARCO TEÓRICO

1.1. Técnicas de elaboración de figuras en relieve

Uno de los aspectos más importantes de las figuras en relieve, son productos que ayudan a personas con discapacidad visual a percibir el entorno que rodea de manera táctil, estas personas utilizan el tacto para poder llevar las experiencias gráficas y visuales a sus mentes. Para esto se requiere convertir la información visual o figuras en relieve, por lo tanto se debe traducir e interpretar los elementos que componen una gráfica tangible a un sistema que reconozca la persona con discapacidad visual la cual se compone de elevaciones de las componentes de un gráfico. Generalmente estos gráficos en relieve se compone por: gráficos elevados (puntos, líneas, texturas), texto (en relieve o no) y código de lectoescritura braille. Es importante tener memoria háptica para poder comprender los estímulos obtenidos mediante el tacto de las figuras en relieve ya que así se podrá seguir de modo ordenado la gráfica y obtener una mejor información del gráfico tangible (Gual,Serrano, Mañez, s/f, p. 3).

Existen varios procesos de elaboración de gráficas tangibles, entre las más extendidas están: estampado, microencapsulado y termoformado estas técnicas reproducen imágenes en relieve realizado en base a material como el papel y láminas de plástico fino.

A continuación se va a describir de manera breve las tres técnicas más empleadas para realizar gráficos tangibles.

- **Sistema Estampado (embossed)**

Consiste en la impresión de puntos braille para representar un gráfico tangible, normalmente es realizado por impresoras Braille ya que utilizan papel como material de impresión, también cabe mencionar que el papel. Este sistema es utilizado en la educación de las personas con discapacidad visual. En la Figura 1-1 representación de un cilindro realizado por estampado de puntos braille.



Figura 1-1: Sistema Empastado.

Fuente: (Gual,Serrano, Mañez, s/f, p. 4)

- **Sistema microencapsulado**

Este sistema reproduce la gráfica tangible sobre un papel microencapsulado donde las partículas de alcohol impregnadas al papel las cuales reaccionan ante la tinta y la aplicación de calor. El costo de la misma hace que solo especialistas en temas de discapacidad visual tengan acceso a dicha tecnología. En la Figura 2-1 se muestra un plano táctil microencapsulado.



Figura 2-1: Sistema de microencapsulado.

Fuente: (Gual,Serrano, Mañez, s/f, p. 5)

- **Sistema de termoformado**

Este sistema utiliza láminas de termoplástico y en base a un molde se realiza el trabajo por medio de calor y vacío para realizar las imágenes en relieve. La matriz o molde se lo realiza con técnicas manuales o también de mecanizado, este sistema es utilizado más en

la industria que para realizar figuras tangibles propiamente. En la Figura 3-1 se muestra el proceso de termoformado.

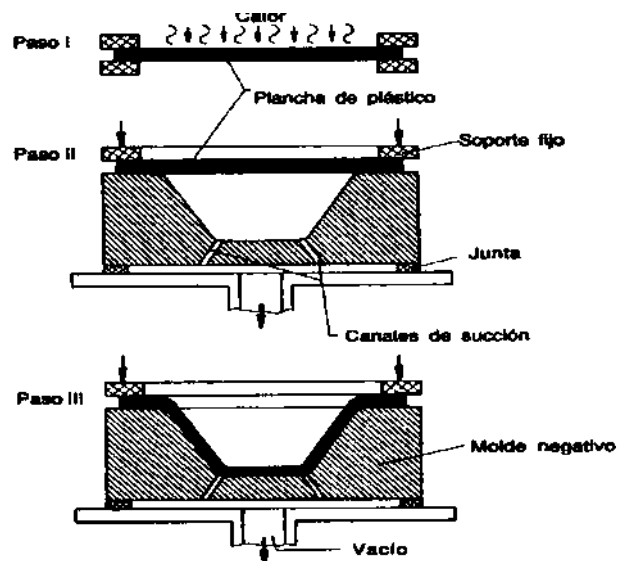


Figura 3-1: Sistema de termoformado.

Fuente: (Gual,Serrano, Mañez, s/f, p. 5)

1.2. Sistemas comerciales basados en control numérico computarizado (CNC)

Los sistemas CNC fueron diseñados con el objetivo de realizar tareas repetitivas con la mayor precisión, basadas en sistemas cartesianos, mejorar los procesos de producción, actualmente se utilizan en el prototipado rápido como son las impresoras 3D, su uso es extendido en los routers de corte, fresado, plotters de dibujo, así también en robots de manipulación de objetos y paletización.

1.2.1. Sistemas electrónicos de elaboración de gráficos tangibles.

Un dispositivo electrónico para la elaboración de imágenes en relieve, es realmente cualquier sistema electrónico que pueda realizar o interpretar el sistema braille incluido las figuras en relieve. Basados en esta premisa se tienen los siguientes sistemas que tienen la capacidad de elaborar figuras en relieve y lenguaje braille.

- INDEX BRAILLE

Empresa sueca líder en el mundo en la fabricación de impresoras Braille, fundada en los años 80, esta empresa tiene modelos variados de impresoras tanto compactas y modulares

según los requerimientos de las personas con discapacidad visual, como el BASIC-D V5 mostrado en la Figura 4-1, que es un sistema compacto de impresora braille de papel continuo.



Figura 4-1: BASSIC-D V5

Fuente: (INDEX BRAILLE, 2018)

Esta impresora braille tiene la capacidad de imprimir 100 caracteres por segundo, esto también implica poder realizar las gráficas tangibles en papel de 120 gramos a modo continuo este sistema utiliza para realizar los puntos en relieve usa un sistema mecánico por solenoide, que por la fuerza impresa en el papel deforma la misma generar así el punto relieve.

- VISION ENGRAVING & ROUTING SYSTEMS

Empresa con sede en los EE.UU, fabricantes máquinas de grabado computarizado CNC y enrutamiento, dentro de esta gama de productos destaca la CNC Router/ Engraver 1624R, que realizar aparte del fresado y enrutamiento realiza gráficos tangibles y lenguaje Braille basado en el sistema Raster, consiste en una herramienta automática de inserción de pequeñas bolas plásticas en agujeros realizados previamente por el router CNC.



Figura 5-1: VISION 1624 ROUTER

Fuente: (VISION, 2018, p. 2)

En la Figura 5-1 se puede apreciar el router CNC de la empresa VISION, tiene un sistema opcional de inserción Braille es utilizado para realizar señalética para personas con discapacidad visual. Algunas de las especificaciones del producto como: el área del trabajo máximo es de 400x600 mm, utiliza un software de propietario llamado Visión Exprés, utiliza un interface Ethernet para la comunicación con el host, también es necesario mencionar el costo 12 450 dólares americanos en EE.UU sin tener en cuenta los costos de importación al Ecuador (VISION, 2018, p. 2).

1.3. Robot cartesiano

Robot es un sistema electrónico programable capaz de realizar tareas repetitivas que puedan resultar de riesgo para las personas (ASALE, s/f). Los robots cartesianos son sistemas organizados con una configuración cartesiana como se muestra en la Figura 4-1, la cual utiliza articulaciones lineales (prismáticas) para poder operar el área de trabajo, los ejes del robot son perpendiculares entre sí, este tipo de sistemas robóticos son utilizados en aplicaciones como la impresión 3D, CNC y plotters de dibujo que basan su funcionamiento en un control numérico computarizado para realizar las tareas.

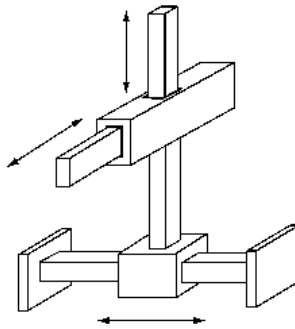


Figura 6-1: Robot Cartesiano.

Fuente: (Barrietos, Peñin, Balaguer, Aracil, 2007, p. 35)

1.4. Editor de gráficos vectoriales

Los gráficos vectoriales, imagen digital integrada por objetos geométricos independientes como pueden ser líneas y curvas, las cuales son muy utilizadas por los diseñadores para realizar ilustraciones, logos y demás productos gráficos, en la actualidad existen varios editores como: Adobe Ilustrador, CorelDraw e Inkscape, este último tiene la particularidad de ser código abierto (Open Source), donde se realiza múltiples operaciones con su variedad de extensiones, esta herramienta ha tomado auge en los últimos años con la tendencia Maker, aficionados a la tecnología y con herramientas como placas de desarrollo y demás productos realizan máquinas como la Cortadoras Laser, CNC y plotters de dibujo. El software permite la edición, modificación y conversión de gráficos vectoriales los cuales luego de un tratamiento se puede ejecutar en un Host y así poder realizar tareas como el grabado en laser, fresado y dibujo, utilizar sistemas cartesianos como herramienta.

1.5. G-code

El G-code es un lenguaje de programación que indica cómo debe realizar el trabajo las máquinas de control numérico computarizado (CNC), indica a la máquina a qué velocidad debe moverse y en qué dirección, al seguir las instrucciones cartesianas.

Los comandos G-code contiene letras y números enteros o racionales, en la Tabla 1-1 se muestra la relación letra seguido de una variación alfanumérica la cual se representa con **nnn** y su respectiva función (Autodesk, 2018).

Tabla 1-1: G-code comandos

| Letra | Función |
|---------|--|
| Gnnn | Movimiento |
| Xnnn | Posición en el eje X a mover (Horizontal) |
| Ynnn | Posición en el eje Y a mover (Vertical) |
| Znnn | Posición en el eje Z a mover (Profundidad) |
| Fnnn | Velocidad de avance cabezal de impresión |
| Snnn | Velocidad de eje |
| Tnnn | Selección de herramienta |
| Mnnn | Funciones Varias |
| I, Jnnn | Centro incremental de un arco |
| Rnnn | Radio de arco |
| P | Parámetro de comando, como el tiempo |

Realizado por: (MUNGABUSI Angel, 2018)

Fuente: (Autodesk, 2018)

En la Figura 5-1 se muestra ejemplos de G-code y su interpretación.

```
1  G0 x0 y0 z0 ; Establecer origen
2  G0 X20 ;      Mover 20 mm en el eje X
3  P200;        Realizar una pausa de 200 mili segundos
4  G0 F2000;    Establecer avance a 2000 mm/minuto
5  M0 ;         Comando parar
6  M106;       Encender Ventilador
7  G0 x0 y0 z0; Volver al origen
8
```

Figura 7-1: G-code y su interpretación.

Realizado por: (MUNGABUSI Angel, 2018)

1.5.1.1. Marlin (Interprete de G-code)

Es un firmware de código abierto para la familia RepRap de prototipos rápidos replicantes o más conocidos como impresoras 3D, es derivado de Sprinter y GRBL, la cual cuenta con una licencia GPLv3 y es gratis para todo tipo de aplicaciones.

El apogeo de este firmware es que utiliza como motor de operaciones un Arduino Mega 250 y la Ramps 1.4 placas de desarrollo populares, la misma que trabaja todo el control de la maquina como los motores a paso, sensores, ventiladores y demás instrumentación todo para la impresión 3D, la versatilidad y las comandos extendidos que posee la hace ideal para realizar todo tipo de

máquinas basadas en sistemas cartesianos (jbrazio, 2018). Para mencionar algunas de las principales características como:

- Códigos G con más de 150 comandos
- Conjunto completo de movimientos de código G que incluye líneas, curvas.
- Compatibilidad con cinemática cartesiana, Delta, SCARA y Core/ H-bot
- Interfaz de usuario
- Impresión basada en tarjeta SD de forma automática

1.5.1.2 Host

Es un dispositivo o computador que provee de servicios ya sean de transferencia de archivos, conexión remota, servicios web u otros. Para el caso se utiliza un computador donde se encuentra alojado un software el cual realizara toda las transferencia de información desde el host hacia la placa de desarrollo la cual tiene que interpretar y retornar la información del estado de la maquina en funcionamiento.

1.5.2. Placas de desarrollo

La variedad de las tarjetas de desarrollo para poder controlar sistemas electrónicos basados en control numérico computarizado son diversos como Arduino, Raspberry Pi. Las más populares basadas en el chip de ATMEL AVR Arduino, para ello se ha desarrollado firmware GRBL, el cual gestiona la parte de control de estas máquinas, las placas Raspberry Pi se descartaron por el modo el cual gestionan la multi-tarea la cual no permite interrumpir procesos en ejecución, la misma que es indispensable en el algoritmo de control del proyecto.

- ARDUINO

Es una Placa de desarrollo de código abierto de precios muy asequibles. Cabe resaltar que se puede realizar con dos modelos como son el Arduino Mega 2560 y Arduino UNO en la Figura 8-1 se muestra las placas en mención. Las cuales son las más documentadas al momento de implementar sistemas CNC (“Arduino - Introduction”, s/f).

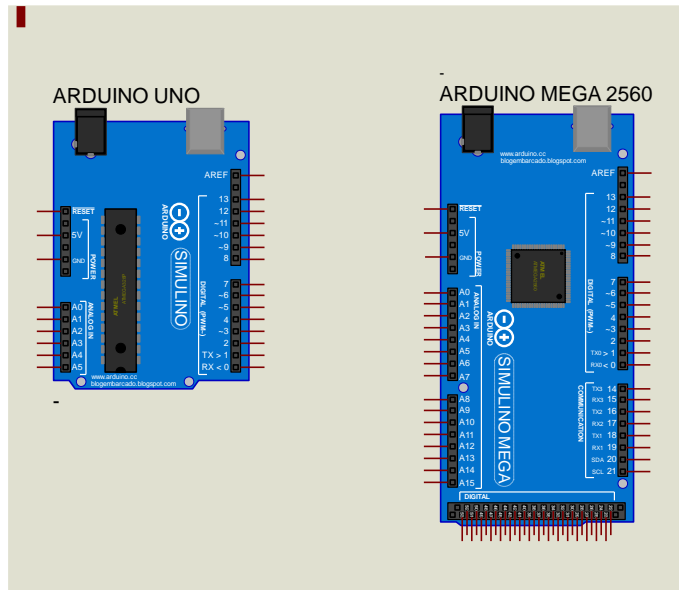


Figura 8-1: Modelos de placas de desarrollo Arduino para realizar robots cartesianos.

Realizado por: (MUNGABUSI Angel, 2018)

1.5.2.1. Placas de circuitos modulares (SHIELD)

Son pequeños circuitos los cuales van acoplados de manera modular sobre las placas de desarrollo, la placa de Arduino tanto puede ser, Arduino UNO o Arduino Mega, de estas son las más populares para la implementación de máquinas de dibujo, CNC e impresoras 3D, en la Figura 9-10 se muestra las shield, que tienen como objetivo el control de motores paso a paso, control de servomotores y control de impresión 3D de manera independiente, que fue desarrollado para realizar maquinas cumplan con los requerimientos de una CNC e Impresora 3D, con la CNC shield y Ramps 1.4, respectivamente (Kruger, 2015) (“RAMPS 1.4/es - RepRap”, s/f)

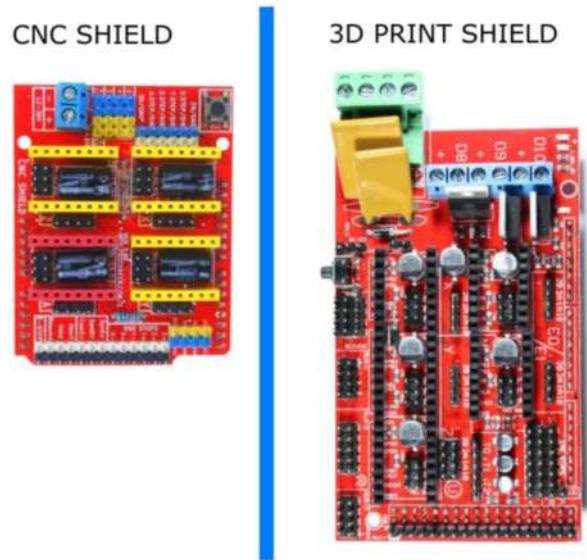


Figura 9-1: Placas de circuito modular (SHIELD)

Realizado por: (MUNGABUSI Angel, 2018)

1.5.3. Motor pasó a paso (PAP)

Los motores paso a paso son dispositivos electromecánicos, que transfieren pulsos eléctricos a un movimiento mecánico rotatorio. El mismo que es se usa generalmente en varias aplicaciones donde es necesario un control preciso de los movimientos donde un sistema de retroalimentación no procede (POLULO, 2018). Entre la variedad de aplicaciones las más destacadas son:

- Impresoras de tinta
- Maquinas CNC
- Impresora 3D de la familia RepRap
- Cortadoras Laser
- Pick and place robots
- Actuadores lineales

En la Figura 10-1 se representa motor paso a paso bipolar.



Figura 10-1: Motor paso a paso bipolar de la marca Polulo.

Fuente: (POLULO, 2018)

1.5.4. Servomotor

Es un motor de corriente continua con características de control y posicionamiento electrónico, compuesto de un sistema de engranajes que aportan la fuerza requerida que es utilizada en aplicaciones educativas en micro robótica, el circuito electrónico es el que se encarga de manejar el movimiento y el control de posicionamiento, y este es comandado por anchos de modulación de pulsos (PWM), se muestra en la Figura 11-1, se puede observar las partes que constituyen un micro servomotor (García Antony, 2016).

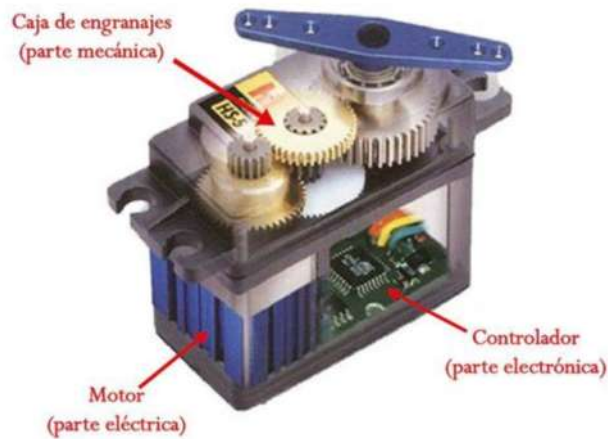


Figura 11-1: Servomotor.

Fuente: (García Antony, 2016)

Tiene como elementos principales la caja de engranajes, controlador o parte electrónica y motor la parte eléctrica.

1.5.5. Bombas de vacío

Son elementos que utiliza un mecanismo por aspiración generan el vacío de un lado y expulsión de aire por el otro, estas bombas se caracterizan por trabajar a corriente continua a 12V, en la Tabla 2-1 se muestra los diferentes tipos de bombas, tasa de flujo y presión (SparkFun, 2018).

Tabla 2-1: Modelos de bombas de vacío.

| Modelo | Tasa de Flujo | Presión | Bomba de vacío |
|-----------------|---------------|---------|--|
| D2028B | 12 L/min | 220 kpa |  |
| PROPUMPS | 5 L/min | 0.8 Mpa |  |
| Jh12-65 | 5 L/min | 65 kpa |  |

Realizado por: (MUNGABUSI Angel, 2018)

1.5.6. Controlador de motor PAP A4988

Controlador de motores pasó a paso, de micro paso con un traductor incorporado para fácil interpretación. Está diseñado para operar motores bipolares paso a paso, en pleno, medio, cuarto, octavo, dieciséis pasos, que maneja 35V a 2 Amperios de salida. Este controlador hace fácil el control ya que al tener un traductor interno, se da un pulso al pin de STEP y el motor da un paso (Allegro, 2018, p. 1). En la Figura 12-1 se muestra el controlador de motor paso a paso A4988.



Figura 12-1: Controlador de motor PAP A4988.

Fuente: (“Pololu - A4988 Stepper Motor Driver Carrier”, s/f)

1.5.7. Solenoide

Dispositivo electromecánico que tiene la particularidad de convertir la energía eléctrica en fuerza y movimiento lineal o giratorio. El movimiento que realice está determinado por la forma en la cual esta ensamblado mecánicamente y puede realizar funciones de push-pull (empujar-halar) (Uriel Méndez, 2016). Las partes que componen un solenoide son:

- Bobina (lleva la corriente)
- Un carcasa de hierro la cual compone el circuito magnético
- Embolo o polo móvil

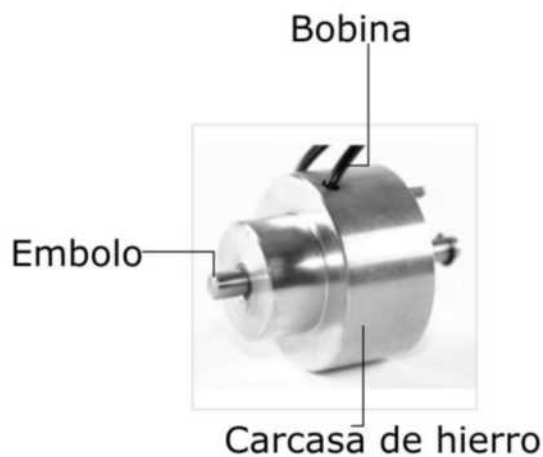


Figura 13-1: Solenoide lineal.

Realizado por: (MUNGABUSI Angel, 2018)

En la Figura 13-1 se aprecia el solenoide lineal el cual se utiliza en variedad de aplicaciones en la industria.

1.5.8 *Sensor de contacto*

Tiene la capacidad de detectar la presencia de un objeto o para el caso de los robots cartesianos utilizados como finales de carrera, los cuales indican que se ha llegado a la posición límite máximo o mínimo del área de trabajo del robot (Adrew Waugh, 2018). En la Figura 12-1.



Figura 14-1: Sensores de contacto

Realizado por: (MUNGABUSI Angel, 2018)

1.5.8 *Sensor óptico reflexivo*

Utiliza un haz de luz que es emitido por un diodo infrarrojo el cual al rebotar en un objeto este es detectado por el fototransistor y de esa manera detecta el objeto que ha pasado al frente del sensor. En el mercado existen módulos que incluyen acondicionamiento de señal para mejorar su usabilidad, también están los que solo constan el sensor individual en la Figura 15-1 se observa sensor óptico reflexivo (Geekbot Electronics, 2018).

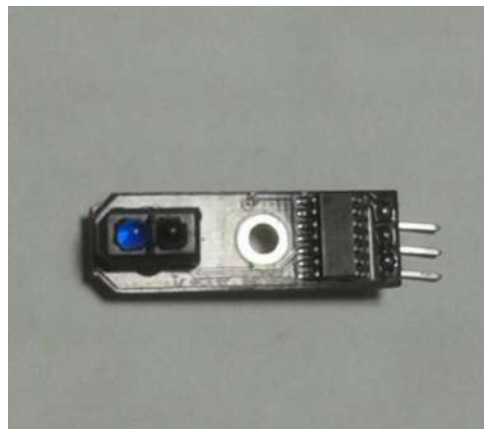


Figura 15-1: Sensor óptico reflexivo

Realizado por: (MUNGABUSI Angel, 2018)

1.6. Normativa INEN 2850

Es necesario mencionar la normativa, ya que bajo ella se rigen los estamentos para la elaboración tanto del sistema braille así también como de las figuras en relieve que en parte de sus normas mencionan lo siguiente:

El ancho de la celda generadora del braille tiene una medida que va desde 3.7mm a 4.5mm, claro está que esto depende directamente de los centros de puntos como se revisa en la normativa INEN 2850 (NTE INEN, 2014).

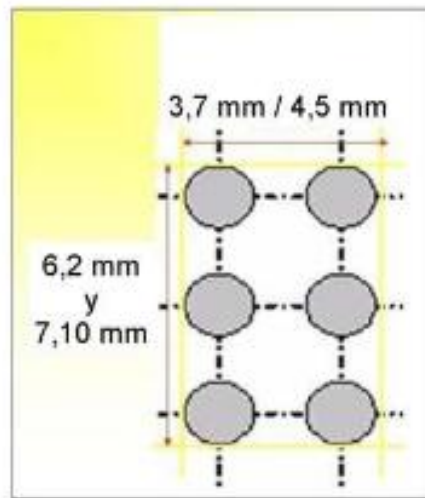


Figura 16-1: Ancho y alto de celda para símbolos braille.

Fuente: (NTE INEN, 2014)

Para las figuras en relieve se menciona que el alto debe tener mínimo 0.5 milímetros, y que toda figura debajo y alineado a la izquierda debe aparecer escrito en sistema Braille la descripción textual de la misma por ejemplo en la Figura 17-1, se muestra un teléfono y debajo su significado en Braille (NTE INEN, 2014). La normativa se encuentra en la página oficial.



Figura 17-1: Símbolo de teléfono con escritura Braille.

Fuente: (NTE INEN, 2014)

CAPÍTULO II

2. MARCO METODOLÓGICO

En el presente capítulo se procede a detallar el proceso de implementación del robot cartesiano para la elaboración de imágenes en relieve. Los requerimientos mecánicos, hardware y software, que serán seleccionados de manera sistemática en función a lo requerido.

2.1 Requerimientos para la implementación del Prototipo.

Luego de haber realizado el estudio pertinente en el capítulo anterior se determina requiere en primer lugar conocer lo siguientes parámetros para los cuales satisfacen al prototipo.

- Ser totalmente funcional y de bajo costo
- Utilizar tecnologías de código abierto.
- Fácil de manejar por el usuario.
- Cumplir con la normativa de diseño de imágenes en relieve.
- La impresión tiene que ser en hoja de papel estándar A4 120 Gramos.

2.2 Concepción general del sistema del prototipo

La manera conceptual se muestra en la Figura 1-2, el sistema general del prototipo, el cual se encuentra constituido por elementos tales como la parte estructural mecánica de sistema cartesiano. Parte de procesamiento de la imagen que incluye generación de código G y envío de datos por comunicación serial, por consecuencia estos van a la tarjeta de desarrollo donde se procesa la información y actúa con el fin de generar las imágenes en relieve.

Para poder comprender la lógica de la concepción del sistema que conforma el prototipo, se va realizar una guía como una guía de ejecución.

- Parte desde una imagen digital la cual puede ser tomada desde una fotografía o descargada de internet en formato PNG, SVG y JPEG.
- Consiguientemente se tiene que pasar al ordenador donde se realizan operaciones a las imágenes y se procede a generar el G-code.
- Al obtener el código g se carga al host desde donde se va a realizar él envío de todos los comandos que conformen el archivo.

- Pasa luego a la parte de ejecución mecánica de las instrucciones antes generadas, el robot cartesiano debe reconocer cada uno de los comandos y ejecutar desde la tarjeta de desarrollo que es de donde se controla el robot.
- Estos comandos hacen que el driver que controlan los motores se muevan por toda el área de trabajo.
- A su vez la electrónica adicional del robot, hace posible el control de los actuadores extra necesarios para poder realizar la imagen en relieve.
- Los actuadores finales como son actuador neumático y cabezal de impresión actúan de manera coordinada con el prototipo.

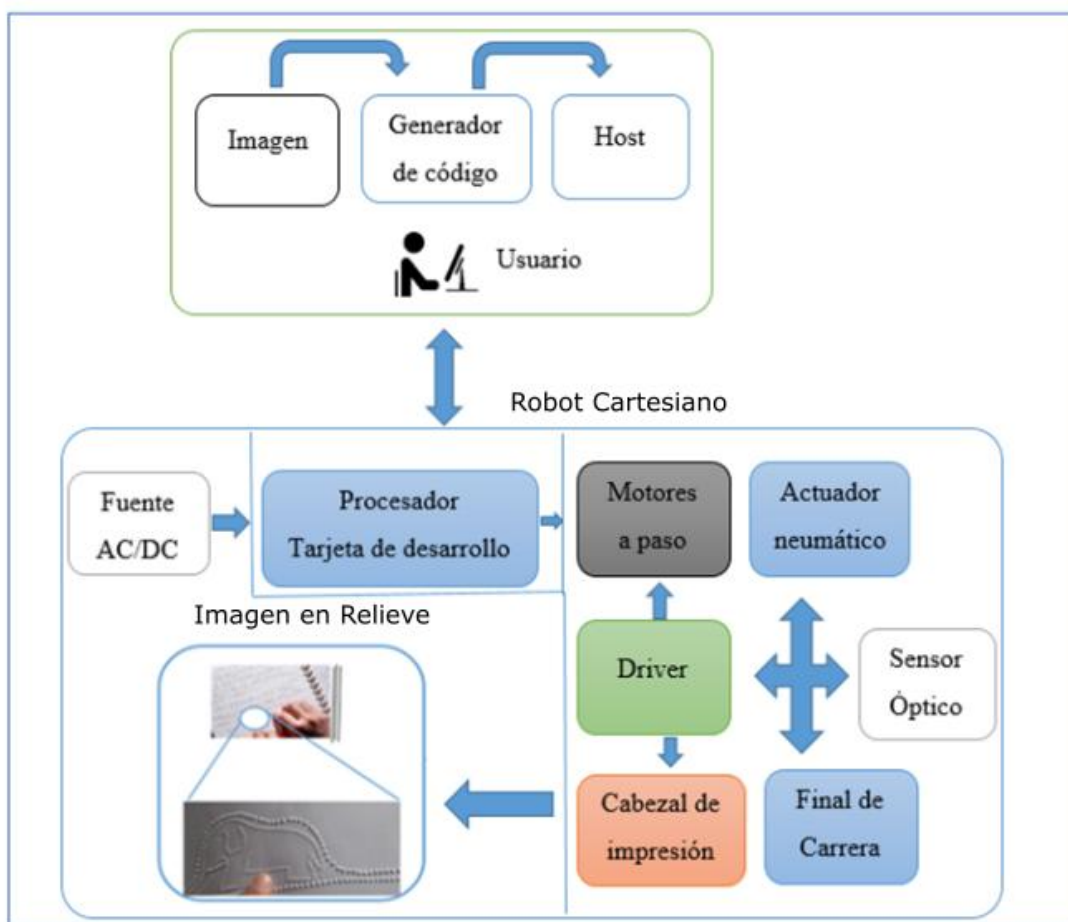


Figura 1-2: Concepción general del sistema de prototipo.

Realizado por: (MUNGABUSI Angel, 2018)

2.3 Diseño de la arquitectura del prototipo

El robot al ser un compendio de un sistema en conjunto se procede a realizar la arquitectura del mismo en la Figura 2-2, se muestra su arquitectura. Se muestran ocho bloques los cuales se observa la interconexión interna del robot, conformado por elementos como:

- Sensor óptico el cual tiene como objetivo censar si existe papel.
- Actuador neumático es utilizado para la ubicación de papel en el área de trabajo.
- Final de carrera como prevención en caso de que se salga de la trayectoria.
- Cabezal de impresión punzón con el que se realizara la deformación de papel para realizar las imágenes.
- Motores pasó a paso.

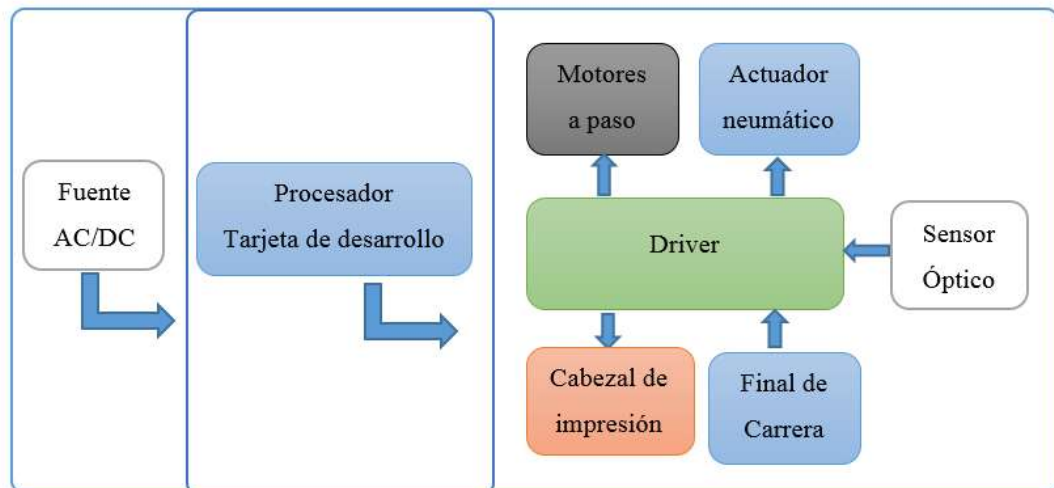


Figura 2-2: Diseño de la arquitectura del prototipo.

Realizado por: (MUNGABUSI Angel, 2018)

2.4 Selección de los elementos para el prototipo

A continuación, se especifican algunas de las particularidades de los elementos del prototipo.

2.4.1 Arduino Mega 2560

Para el prototipo se utiliza Arduino Mega 2560 debido a su versatilidad al momento de implementar proyectos robóticos con características cartesianas. Arduino Mega es el seleccionado para el proyecto por contar con mayor número de terminales de entrada y salida (E/S), ya que esto ayuda a manejar más elementos eléctricos y electrónicos que conforman el prototipo es más para

nuestro proyecto requiere de actuadores extra los cuales maneja sin inconvenientes. En la Figura 3-2 se muestra la tarjeta de desarrollo basada en Arduino Mega 2560 (“Arduino Mega 2560 R3”, 2014).

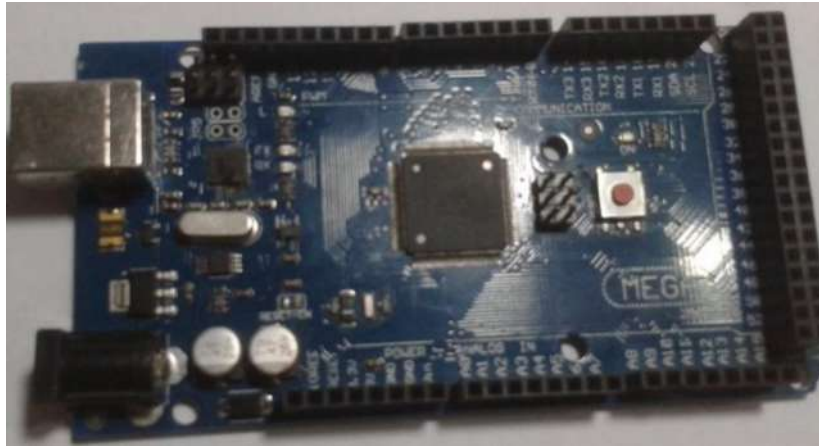


Figura 3-2: Arduino Mega 2560

Realizado por: (MUNGABUSI Angel, 2018)

Sus principales características se muestran en la Tabla 1-2. Adicionalmente se puede revisar parte las características de la tarjeta de desarrollo Arduino Mega en el Anexo A.

Tabla 1-2: Características del Arduino Mega 2560

| | |
|---------------------|---|
| Nombre del producto | Arduino Mega 2560 |
| Procesador | ATMEGA2560 |
| Voltaje operativo | 5V DC |
| Terminales E/S | 54 terminales (E/S) de ellos 14 son PWM |
| Entradas analógicas | 16 Analógicas |
| Memoria Flash | 256K |

Realizado por: (MUNGABUSI Angel, 2018)

Fuente: (“Arduino Mega 2560 R3”, 2014)

2.4.2 *Sensores para el prototipo*

A continuación se muestra los sensores utilizados para implementar el prototipo, los cuales son dos el primero es el sensor óptico que tiene como función detectar la hoja de papel y el sensor de contacto que es utilizado como final de carrera.

- **Sensor Óptico TCRT5000L**

Ideal para detectar un cambio en la superficie sobre el cual se trabaja y cuenta con un sensor óptico reflectivo infrarrojo y que combinado con un circuito integrado como comparador de voltaje hace la detección de cambio de una manera eficiente y precisa (Geekbot Electronics, 2018). En la Figura 4-2 se muestra el sensor óptico.

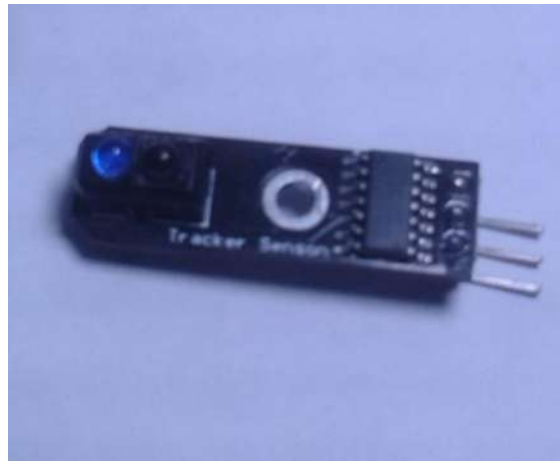


Figura 4-2: Sensor óptico reflexivo

Realizado por: (MUNGABUSI Angel, 2018)

Las características relevantes se muestran en la Tabla 2-2, y parte de sus características en el Anexo B.

Tabla 2-2: Características principales del sensor TCRT5000L

| | |
|------------------------|--------------------------|
| Nombre del Producto | Sensor Óptico |
| Voltaje de operativo | 5V DC |
| Distancia de detección | 10mm-25mm |
| Dimensiones | 32x14x10 mm |
| Peso | 3 gramos |
| Salida | Digital ["1"-“0”] lógico |

Realizado por: (MUNGABUSI Angel, 2018)

Fuente: (Geekbot Electronics, 2018)

- **Sensor de contacto (final de carrera)**

Este sensor es mayormente utilizado en el área de actuadores lineales para determinar si su elemento terminal ha llegado al final de la trayectoria y consecuentemente parar el accionamiento (Bart Meijer, 2014). El mismo que se muestra en la Figura 5-2.



Figura 5-2: Sensor de contacto

Realizado por: (MUNGABUSI Angel, 2018)

También se encuentran parte del Datasheet en el Anexo C y sus características principales en la Tabla 3-2.

Tabla 3-2: Características principales del sensor de contacto.

| | |
|------------------------|--------------------|
| Nombre del producto | Sensor de contacto |
| Voltaje operativo | 5V DC |
| Elementos Auxiliares | Luces piloto |
| Dimensiones | 10x30x0.7mm |
| Terminales de conexión | Vcc-Out-Gnd |

Realizado por: (MUNGABUSI Angel, 2018)

Fuente: (Bart Meijer, 2014)

2.4.3 *Shield Ramps 1.4*

Esta placa modular que permite el control de un driver para poder manejar un motor a pasos y su principal uso es para impresión 3D así también para robots cartesianos de propósito general. La placa es amplia tiene también características especiales como poder manejar actuadores tipo servomotor, sensores ópticos y de contacto adicionalmente se le puede acoplar un LCD y una tarjeta de memoria para poder imprimir directamente desde esa ubicación sin tener un host (Jhonny R, 2018). En la Figura 6-2 se muestra su presentación.

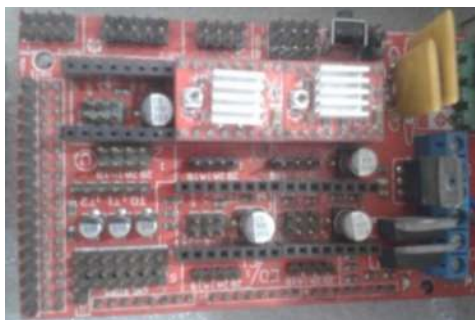


Figura 6-2: Shield Ramps 1.4

Realizado por: (MUNGABUSI Angel, 2018)

Las características técnicas principales se muestran en la Tabla 4-2 y la conexión de la placa se muestran la página oficial, la misma que está en el Anexo D.

Tabla 4-2: Principales características de la Ramps 1.4

| | |
|---------------------------|-------------------|
| Nombre del producto | Ramps 1.4 |
| Voltaje de trabajo | 12V DC |
| Placa modular de conexión | Arduino Mega 2560 |
| Dimensiones | 100x60x10.5mm |

Realizado por: (MUNGABUSI Angel, 2018)

Fuente: (Jhonny R, 2018)

2.4.4 *Diseño de circuitos para el prototipo*

A continuación se muestra los diagramas de los circuitos extra que fueron necesario, básicamente su funcionalidad radican en el manejo de la electrónica de potencia para activar los actuadores dentro de ellos están el mini compresor de aire y el solenoide que posteriormente se especificara respectivamente.

- **Circuito para el control del sensor óptico y control del mini compresor de aire.**

El mini compresor de aire necesita de un voltaje de 12V que deben ser conmutados por el Arduino con la salida de 5V para poder activar el actuador. También del mismo se deriva el circuito para detectar el papel con el sensor óptico.

Para controlar el mini compresor de aire se realiza un conmutador electrónico mediante un transistor de potencia IRFZ44N pueden revisar parte de las características técnicas en el Datasheet en el Anexo F.

Con la Ecuación 1-2 se determina la resistencia requerida para obtener una intensidad de corriente de activación o conmutación del transistor.

Ecuación 1-2: Resistencia para conmutar el transistor IRFZ44N.

$$R = \frac{V_G}{I_G}$$

Donde:

R: resistencia de conmutación que habilita el terminal Gate del transistor IRFZ44N.

V_G : Voltaje del Gate

I_G : Intensidad mínima para activar el transistor

Al conocer el voltaje de salida de la Ramps 1.4 de 5V DC, y una intensidad mínima requerida de 25 a 250 miliamperios para garantizar el funcionamiento se ha tomado el valor de 50 miliamperios al aplicar la Ecuación 1-2 como resultado R igual a 10kohms.

La resistencia R2 que tiene un valor de 100kohms se conecta hacia GND mismo que permite mantener un cero lógico y la intensidad de corriente mínima de 0.1miliamperios que impida que el transistor se active por ruido o fallas. Para las resistencias R4 y R5 son calculadas del mismo modo que R ya que el transistor Q1 tiene las mismas características que el irfz44n y realiza un trabajo idéntico.

A continuación se describe las conexiones del circuito mostrado en la Figura 7-2.

- Los conectores J1 y J2 se acoplan a los terminales del Auxiliar 3 y al terminal D9 de Ramps 1.4 respectivamente.
- Terminal J3 se conecta al mini compresor de aire en paralelo a estos se encuentra el diodo D1 rectificador así también el capacitor C1, con el objetivo de prevenir los voltajes en inversa y el ruido del motor individualmente.
- J4 es la alimentación de 12V DC para el mini compresor de aire.
- Adicionalmente a esto se tiene un diodo led (D2) que tiene la función de alertar si está en funcionamiento todo el circuito como luz piloto.
- El diodo D3 y el capacitor C2 tiene el objetivo de proteger el circuito del transistor Q2 del zumbador, que tiene como objetivo alertar la falta de papel en la bandeja.

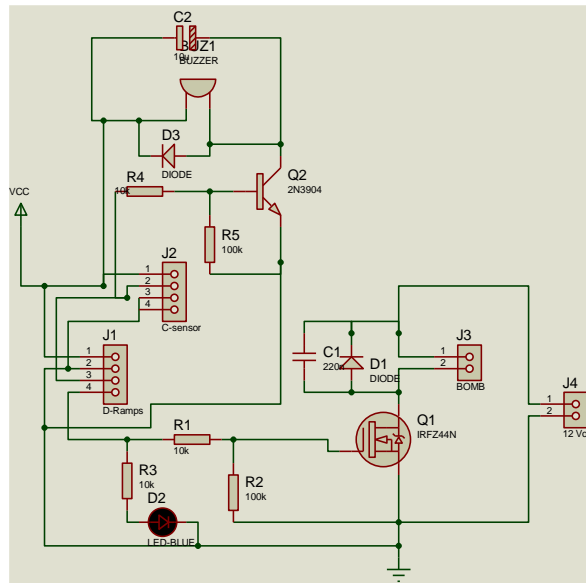


Figura 7-2: Diagrama del circuito para el mini compresor de aire y sensor óptico.

Realizado por: (MUNGABUSI Angel, 2018)

- **Circuito de potencia para el control del solenoide**

La electrónica para el control del solenoide está basado en un circuito integrado LM555, las características están en el datasheet (Anexo E), su funcionamiento radica en generar pulsos a cierta frecuencia y estos ser relevados por el MOSFET IRFZ44N características (Anexo F), el cual realiza el trabajo ON/OFF del solenoide. En la Figura 8-2 se muestra el diagrama del circuito de potencia del solenoide.

Para esto se toma en cuenta la fuerza que tiene el solenoide que es de 6 Newton la cual es la idónea para deformar el papel y generar los puntos en relieve.

Para hallar los valores de los componentes del circuito que determinan la frecuencia de conmutación del circuito LM555 en modo astable, se parte de la velocidad con la que el robot se va a mover en las trayectorias.

Velocidad del Robot en los ejes cartesianos: 1000(milímetros/Minuto)

Con la Ecuación 2-2, permite obtener los tiempos con los cuales el solenoide realiza el trabajo de troquelado para realizar el relieve en la hoja de papel(Uriel Mendez, 2016).

Ecuación 2-2: Ciclo de trabajo del solenoide.

$$DTC = \frac{on}{on + off} \times 100$$

Donde:

DTC: Ciclo de trabajo del solenoide

On: tiempo de encendido del solenoide

Off: Tiempo de apagado del solenoide

- El ciclo de trabajo se mide de manera porcentual y los límites de encendido en el 10% si se desea trabajar al voltaje máximo soportado por el solenoide y obtener la fuerza total de la misma.
- Con esto se obtiene un ciclo de trabajo de $DTC=10\%=0.1$
- Al igual que para la escritura como para los símbolos e imágenes se detalla en la norma NTE INEN 2850. Donde cita que el espacio entre centros de puntos contiguos la medida debe ser entre 2.4-2.75mm, para el caso se ha tomado el valor 2.75mm con el objetivo de tener una resolución óptima y no dañar la hoja de papel (NTE INEN, 2014).
- Con esto se ha determinado que para recorrer 2.75mm con una velocidad de 16.666 mm/seg es necesario un tiempo de 180 milisegundos, este tiempo entre dos puntos de relieve es el tiempo que el solenoide debe estar en estado de apagado.
- Off= 180 milisegundos
- Queda por remplazar valores de la Ecuación 1-2, y determinar el valor de encendido:
- On =20 milisegundos

Con esto se procede a realizar los cálculos para hallar la frecuencia del circuito oscilador cumpla con las condiciones antes mencionadas.

La ecuación 3-2, se determina el tiempo de bajo del oscilador (Floyd, 2010, p. 824).

Ecuación 3-2: Tiempo de salida está en bajo.

$$t_l = 0.694R_2C_{ex}$$

Donde:

Tl: tiempo de salida en bajo

R2: Resistencia que para el caso es RV2 en el diagrama

Cex: Es el capacitor externo del diagrama de la Figura 8-2

Ecuación 4-2: Tiempo de salida está en Alto.

$$t_h = 0.694R_1C_{ex}$$

Donde:

Th: tiempo de salida en alto

R1: Resistencia de carga en el diagrama de la Figura 8-2 como RV1

Cex: Capacitor externo.

Se determina los valores de las resistencias RV1 y RV2, se toma como valor predefinido del capacitor de 10uF (microfaradios) y realizar pruebas si se puede alcanzar el valor de ON Y OFF del solenoide calculado con anterioridad por lo tanto se iguala los valores.

ON=th=180milisegundos, OFF=tl=20milisegundos

Como resultado.

$$R2 = 27.7K\Omega$$

$$R1 = 3.08K\Omega$$

R2 y R1 resistencias para el circuito oscilador que requiere el integrado LM555. En la Figura 8.2 se muestra el circuito esquemático con los valores previamente calculados.

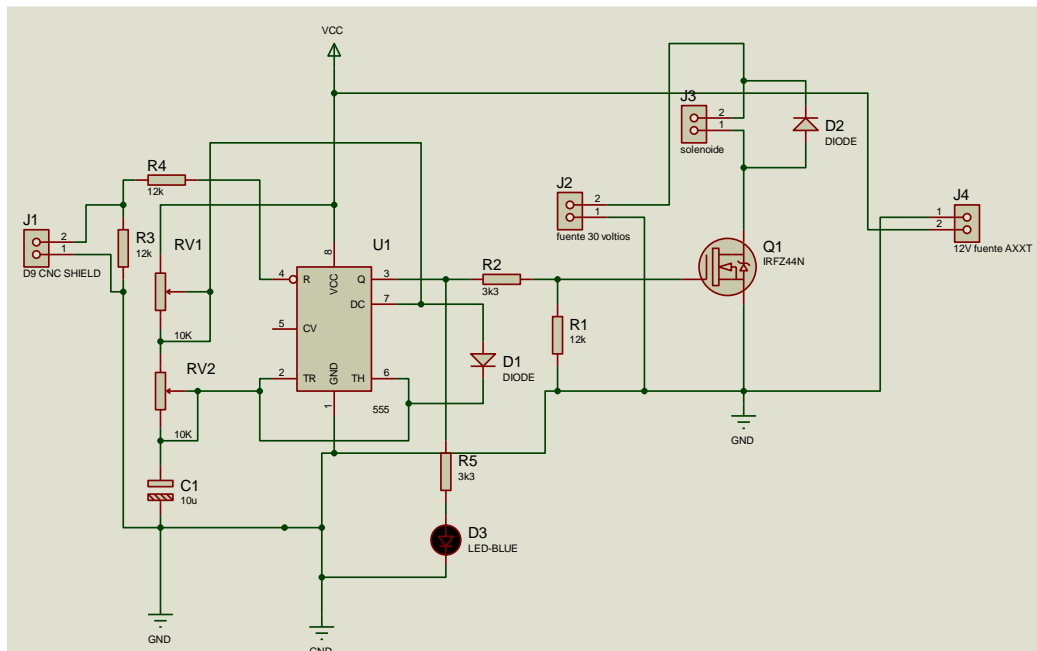


Figura 8-2: Diagrama del circuito de control de solenoide

Realizado por: (MUNGABUSI Angel, 2018)

2.4.5 Selección de actuadores

La presente sección determina los actuadores a utilizarse para el prototipo.

2.4.5.1 Motor de pasos Nema17

Para que el robot realice movimientos lineales a lo largo de los ejes se ha elegido motor de pasos por su precisión. El motor de pasos permite precisión al momento de realizar las imágenes en relieve, También dota de versatilidad, reducido tamaño y bajo costo se elige el motor NEMA17 en la Figura 9-2 se muestra su presentación.

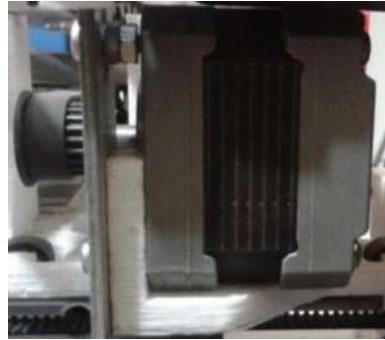


Figura 9-2: Motor a pasos Nema17.

Realizado por: (MUNGABUSI Angel, 2018)

Las características principales del motor Nema17 se muestran en la Tabla 5-2 y parte de su Datasheet en el Anexo G.

Tabla 5-2: Características principales del motor de pasos NEMA17

| | |
|---------------------|-----------------|
| Nombre del producto | Nema17 |
| Voltaje de trabajo | 12V DC |
| Peso | 0.28 Kilogramos |
| Pasos/Revolución | 200 |
| Exactitud | +/-5% |
| Clasificación IP | 40 |

Realizado por: (MUNGABUSI Angel, 2018)

Fuente: ("Stepper-Motor-Support-Document.pdf", s/f, p. 1)

2.4.5.2 Solenoide como punzón Braille.

Para realizar las imágenes en relieve es necesario realizarlos con un punzón, se realiza con un solenoide el mismo que al accionarse golpeará con la fuerza necesaria para deformar la hoja de papel y genera puntos que en conjunto forman las imágenes en relieve. En la Figura 10-2 se muestra el solenoide seleccionado (NSFCNTROLS, 2018).



Figura 10-2: Solenoide seleccionado.

Realizado por: (MUNGABUSI Angel, 2018)

Las características técnicas del solenoide en la Tabla 6-2 (PotentialLabs, 2018).

Tabla 6-2: Características técnicas del solenoide elegido

| | |
|---------------------|-----------------------------|
| Nombre del producto | Solenoide electroimán de CC |
| Modelo | JF-0630B |
| Voltaje nominal | 24V DC |
| Tipo | Pull Push |
| Corriente nominal | 300 mA |
| Fuerza y trazo | 6N / 10mm |
| Tamaño | 30x20x18mm |
| Tamaño del embolo | 7x58mm (Máx.*L) |
| Longitud del Cable | 19 cm |
| Material | Metal y partes eléctricas |
| Peso | 46g |

Realizado por: (MUNGABUSI Angel, 2018)

Fuente: (PotentialLabs, 2018)

2.4.5.3 Selección de mini compresor de aire

El prototipo debe contar con alimentación automática de papel para la elaboración de las imágenes tangibles, por esa razón se ha implementado un sistema por succión basado en una bomba de vacío y una ventosa anclada al efector final del prototipo, la misma que tiene la tarea de ubicar el papel en el área de trabajo y quitarla en la Figura 11-2, de muestra la bomba seleccionada (SparkFun, 2018).



Figura 11-2: Mini compresor de aire.

Realizado por: (MUNGABUSI Angel, 2018)

La Tabla 7-2 muestra las características principales del mini compresor de aire y en el Anexo H se encuentran el documento con algunas de las especificaciones técnicas.

Tabla 7-2: Especificaciones técnicas de mini compresor de aire.

| | |
|---------------------|-----------------------------|
| Nombre del Producto | AIRPO |
| Modelo | D2028B |
| Voltaje de trabajo | 12V DC |
| Potencia | 12W |
| Rango de flujo | 12 a 15 (Litros por minuto) |
| Rango de presión | 0 a 32 PSI |

Realizado por: (MUNGABUSI Angel, 2018)

Fuente: (SparkFun, 2018)

Ventosa

Un elemento adicional necesario para el uso del mini compresor es una ventosa la cual nos ayuda a sujetar el papel en la Figura 12-2 se muestra el modelo VC-33^a 1/8 de la empresa VI-CAS (Vi-Cas, 2018).



Figura 12-2: Ventosa de succión.

Realizado por: (MUNGABUSI Angel, 2018)

Los datos de las principales características de muestra en el Anexo I los datos técnicos se presentan en la Tabla 8-2.

Tabla 8-2: Datos técnicos de ventosa

| | |
|-------------------------|--------------------|
| Nombre del producto | Ventosa |
| Modelo | VC-33 ^a |
| Tipo de anillo de junta | Hembra 1/8 NPT |
| Material | Silicona |
| Tipo de montaje | Roscado |

Realizado por: (MUNGABUSI Angel, 2018)

Fuente: (Vi-Cas, 2018)

2.5 Selección alimentación para el prototipo

La alimentación para el prototipo se eligió en base al requerimiento tanto como del voltaje y de corriente por etapas, para la etapa de potencia del solenoide se eligió una fuente de 30 voltios requerido a una corriente de 500mA, por otra parte todo el sistema del prototipo funciona con 12 Voltios y se requiere una corriente de 2.5 se elige la fuente micro ATX (Acteck, 2018). En la Figura 13-2 se muestra su presentación.



Figura 13-2: Fuente de alimentación micro ATX.

Realizado por: (MUNGABUSI Angel, 2018)

Adicionalmente se muestra parte del Datasheet sobre la fuente micro ATX en el Anexo J y sus principales características en la Tabla 8-2.

Tabla 8-2: Características principales fuente de alimentación micro ATX

| | |
|---------------------|---------------------------|
| Nombre del producto | Fuente micro ATX |
| Potencia | 500W |
| Rango de operación | 100 VAC - 240VAC |
| Voltaje de salida | +12V, +5V, +3.3V, -12V DC |
| Dimensiones | 60*120.5*100mm |

Realizado por: (MUNGABUSI Angel, 2018)

Fuente: (Acteck, 2018)

La fuente de 30V DC para alimentar el solenoide es una fuente de computadora.

2.6 Esquema de conexión para el prototipo

Culminado el proceso de selección de todos los elementos que constituyen el prototipo.

Donde se acopla los elementos de forma gráfica y conectar a forma de bus de datos los cables y no de forma individual para evitar confusiones dentro de la conexión, también en lo posible se ha realizado las líneas de conexión con diferente color para una mejor visualización de los elementos, adicionalmente se a etiquetado los conectores de manera que su identificación y

conexión sean comprensible. Como parte central de la conexión se parte del Arduino Mega 2560, sobre ella se acopla la Ramps 1.4. Se muestra en la Figura 14-2.

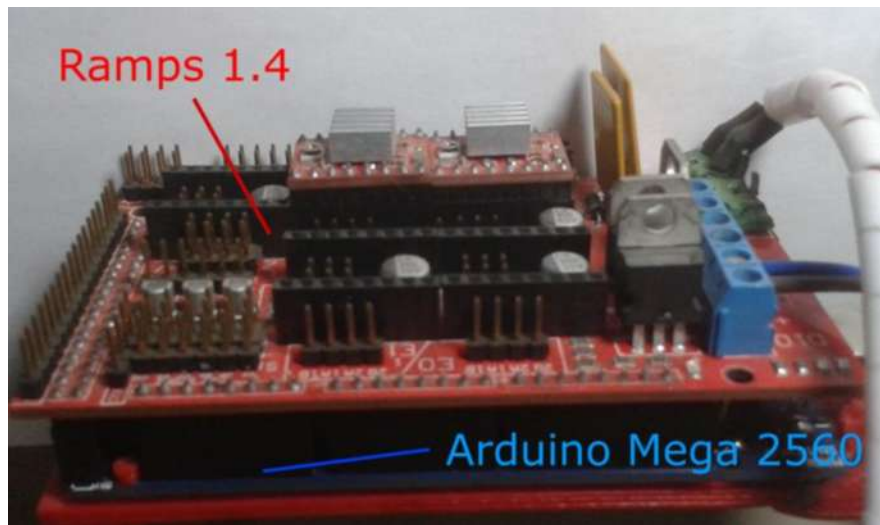


Figura 14-2: Conexión inicial de Ramps 1.4 sobre Arduino Mega 2560.

Realizado por: (MUNGABUSI Angel, 2018)

Esto de forma modular, a partir de ello se realiza la conexión con las fuentes de poder, así también los circuitos adicionales implementados para el control del mini compresor de aire, además para el control de solenoide y sensor de papel. En la Figura 15-2 se observa el esquema de conexión.

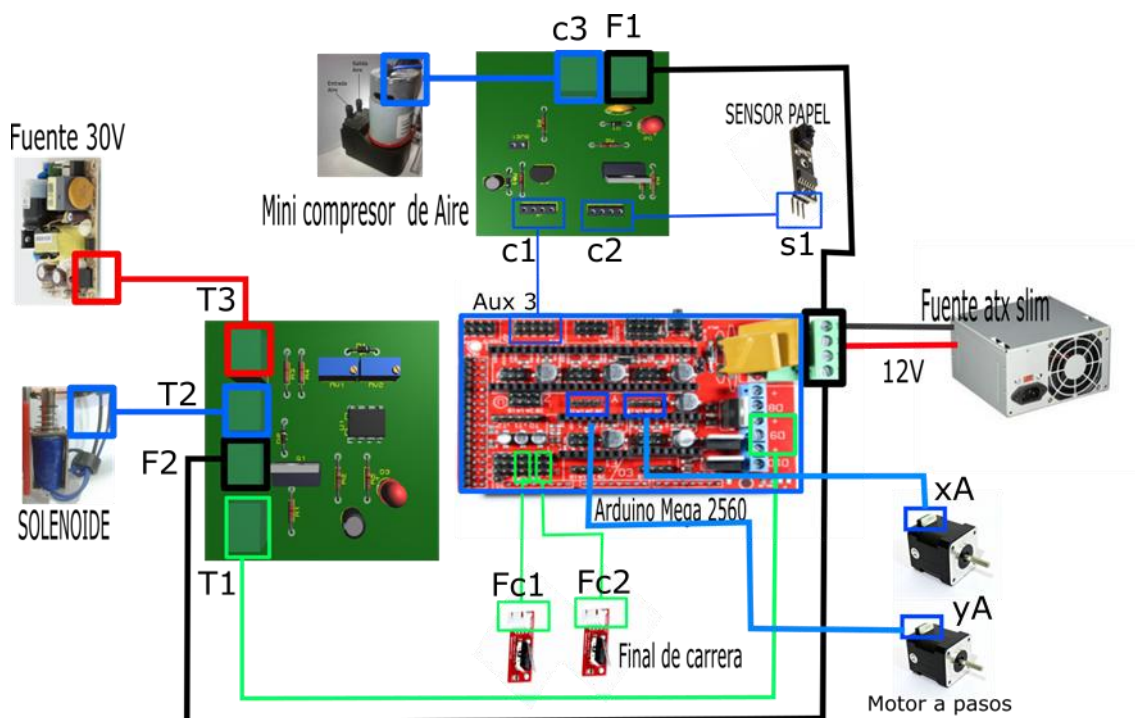


Figura 15-2: Esquema de conexión para el prototipo.

Realizado por: (MUNGABUSI Angel, 2018)

El esquema de conexión del robot se muestra a continuación.

- Arduino Mega es la parte central de las conexiones, sobre los terminales de esta se coloca de forma modular la Ramps 1.4, como se observa en la Figura 14-2.
- La alimentación es suministrada por la fuente ATX Slim, tanto al Arduino como también a los circuitos adicionales por los conectores F1 y F2 con 12V DC respectivamente.
- Desde los terminales de conexión x, y de la Ramps 1.4 se conecta con los motores a pasos mediante los terminales xA (Eje X) & yA (Eje Y).
- Los finales de carrera Fc1 y Fc2 se conectan a los terminales de la Ramps 1.4, llamados X- y Y-.
- Desde la Ramps 1.4 se conecta el terminal D10 hacia T1 que es la conexión de control para el circuito.
- Del terminal T2 se conecta con el solenoide.
- El terminal T3 se conecta con la fuente de alimentación para el solenoide de 30V DC.
- Aux 3 se conecta mediante un bus de datos hacia c1 el cual permite comunicar sobre la situación del sensor de papel y control del mini compresor de aire.
- C2 se comunica mediante un bus de datos hacia el sensor de papel.
- El terminal C3 se conecta al mini compresor de aire.

2.7 Software para el prototipo

Se basa en tecnologías de código abierto, como el Entorno de Desarrollo Integrado (IDE) de Arduino, adicionalmente se hace uso de editores de gráficos vectoriales y un host alojado en la computadora para enviar la información al prototipo con esto se dará cumplimiento con los requerimientos.

2.7.1 *Requerimientos de software por el prototipo*

- Manejo de imágenes vectoriales para posteriormente generar el Gcode
- Enviar mediante comunicación serial la información de Gcode hacia el robot para su interpretación.
- Generación de comandos complementarios para las funciones extra del robot.
- Control de movimiento del robot que tiene estructura cartesiana
- Permitir el manejo de actuadores
- Control del punzón generador de código Braille.

2.7.2 Generación de comandos (Gcode) con Inkscape

Para el proyecto se hace uso de un editor de gráficos, ya que al querer realizar imágenes en relieve estas con anterioridad requieren de un tratamiento y posterior generación de su código g para que así el prototipo pueda imprimir dicha gráfica se ha optado por el uso de Inkscape por ser un programa de código abierto y amplio uso para generación de gcode para cortadoras laser y fresadoras, con base en ello se ha procedido a incorporar una función nueva que genera las trayectorias así como el manejo de actuadores como el mini compresor de aire y el servomotor como actuador lineal (Inkscape, 2018).

El programa diseñado como una función adicional es desarrollada en lenguaje de programación Python, se ha hecho uso de ella porque es versátil y las extensiones de Inkscape están desarrolladas sobre Python se hace uso de última versión 3.7.0 que se descarga de la página oficial (Python, 2018).

De forma adicional se usa un editor de código fuente gratuito que puede manejar Python llamado Notepad++ versión 7.5 que se descarga desde la página oficial de la misma (NotepadPlusPlus, 2018).

En la Figura 16-2, en la barra de menú se dirige a extensiones dentro de ella en la pestaña Generate Braille Gcode, que es la nueva función que se añade al programa Inkscape.

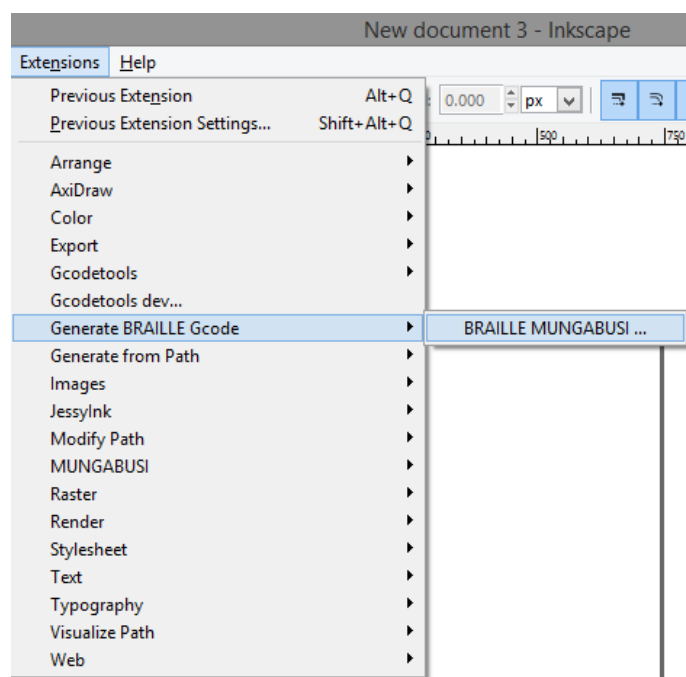


Figura 16-2: Función añadida llamada Generate Braille Gcode.

Realizado por: (MUNGABUSI Angel, 2018)

Finalmente se abre la pestaña BRAILLE MUNGABUSI, el cual abre una venta donde se puede configurar los parámetros de impresión para el prototipo, que se muestra en la Figura 17-2.

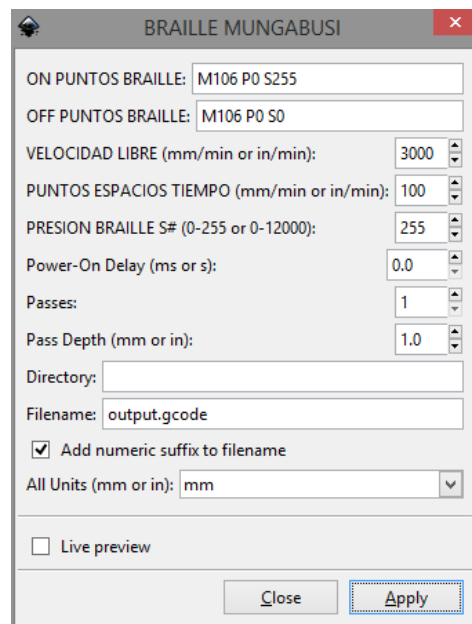


Figura 17-2: Ventana de configuración de parámetros para la impresión.

Realizado por: (MUNGABUSI Angel, 2018)

Al dar en el botón aplicar se genera los comandos código G que a posterior se interprete y genere las figuras en relieve.

Ciclo de generación de comandos para prototipo:

Los comandos se generan por la función BRAILLE MUGABUSI, siguen una secuencia la cual permite al prototipo, ubicarse en el área e imprimir las imágenes en relieve y símbolos Braille al hacer uso de comandos G y M especiales configurados para el propósito.

- Ubica al prototipo en el origen con el comando G1 X0Y0
- Apaga el mini compresor con el comando especial M42 P44 S0
- Servomotor se ubica en la posición de 0 grados con el comando especial M280 P0 S0
- Apaga el solenoide con el comando especial M106 P0 S0
- Sensor de papel espera el valor de lógico de Alto (1) con el comando M226 P42 S1
- Ubicar el prototipo en la posición G1 X75Y350
- Servomotor a 60 grados M280 P0 S60, con el comando se pretende poner la ventosa sobre la hoja de papel

- Activar el mini compresor M42 P44 S255, succiona la hoja de papel por la ventosa.
- Servomotor a 45 grados M280 P0 S45 con esto se levanta la hoja
- Ubicar el prototipo en la posición G1 X75Y0, ubica la hoja de papel en el área de trabajo
- Servomotor a 60 grados M280 P0 S60, baja la hoja de papel sujeta por la ventosa
- Apagar mini compresor M42 P44 S0 suelta la hoja de papel en el área de trabajo
- Ubica al robot en la posición G1 X0Y0 regresa al origen para iniciar con la impresión.
- Generación de código G de las trayectorias del gráfico y símbolos Braille.

Para poder revisar de una manera detallada se muestra en el Anexo K el código fuente de *BRAILLE MUNGABUSI* que se añadió a Inkscape desarrollada en lenguaje de programación Python versión 7.5.0.

2.7.3 Software para el manejo del prototipo Repetier-Host

El prototipo requiere de un host que maneje el envío de los comandos antes generados desde el editor de gráficos Inkscape *BRAILLE MUNGABUSI*, para ello se ha elegido el Repetier-Host el mismo que tiene como uso el manejo de impresoras 3D, basadas en sistemas cartesianos y otros, las funciones que añade como: Intuitivo, Interfaz amigable y configurable a las necesidades de los usuarios, por tal razón lo ha hecho ideal para el proyecto, también cabe mencionar que es de código abierto y esto facilita el acople con las tecnologías antes mencionadas y poder cumplir con los requerimientos, en la Figura 18-2 se muestra el entorno de trabajo del software Repetier-Host (Repetier, 2018)

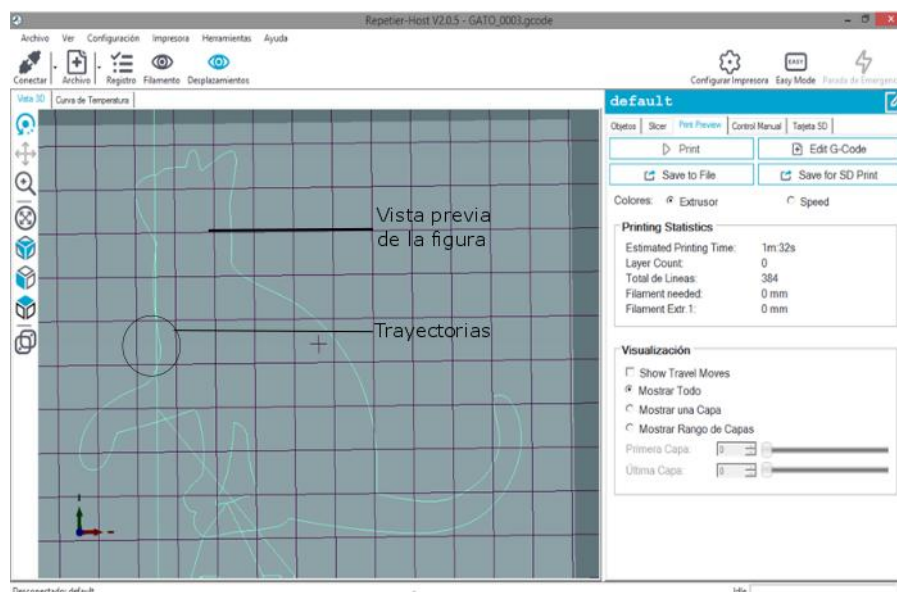


Figura 18-2: Interfaz de Repetier-Host.

Realizado por: (MUNGABUSI Angel, 2018)

Características principales

- Permite edición del Gcode
- Control manual del robot
- Permite vista previa de la figura

También posee una ventana de control manual de donde se puede ubicar al prototipo hasta la posición cero (origen) en el caso de que se pierda la energía o problemas de ubicación con el prototipo, como se muestra en la Figura 19-2.

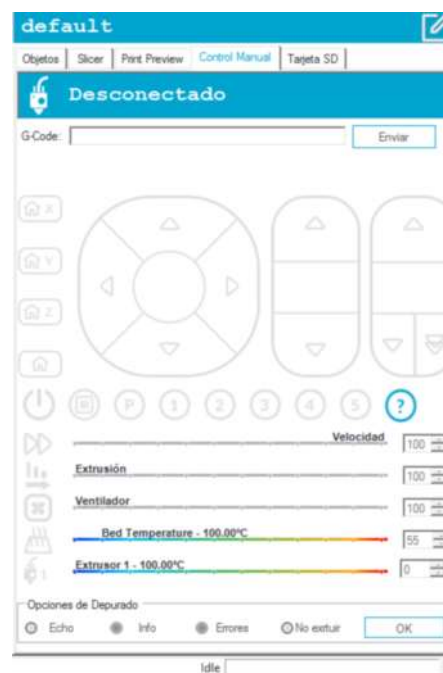


Figura 19-2: Ventana de control manual del prototipo.

Realizado por: (MUNGABUSI Angel, 2018)

2.7.4 Diagrama de flujo del intérprete de comandos Marlin del prototipo

El funcionamiento del prototipo requiere de un software para la interpretación de los comandos G y M los cuales proceden del Repetier-Host, el mismo debe comandar todas las actividades desde el procesador hacia la parte actuante del prototipo.

Se ha seleccionada el firmware Marlin, que está diseñada para impresoras 3D basadas en Arduino Mega y con la Ramps 1.4 esto facilita que otros proyectos que requieran de manejo de sistemas cartesianos los puedan utilizar sin inconvenientes, se añada funcionalidades como: Manejo de

Servomotores, Interpretación de comandos M se descarga de la página oficial (MarlinFirmware, 2018). En la Figura 20-2 se muestra el diagrama de flujo del intérprete de comandos.

Ciclo de repetición

- Inicia comunicación serial
- Verifica si esta tiene comando G o M
- Interpreta los comandos G y M
- Finaliza la comunicación

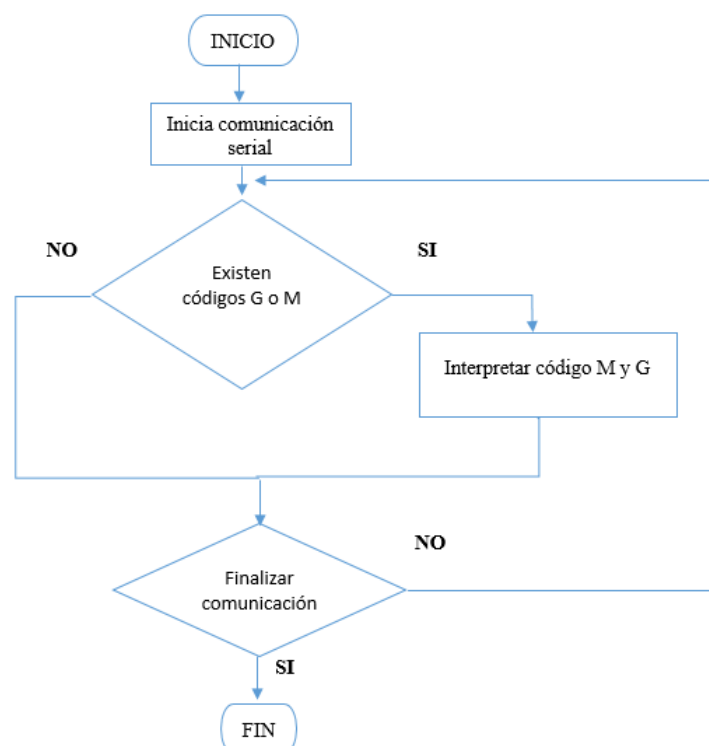


Figura 20-2: Diagrama de flujo interprete de comandos Marlin.

Realizado por: (MUNGABUSI Angel, 2018)

2.7.5 Diagrama de flujo del proceso de impresión del prototipo

El proceso de impresión toma al prototipo una sucesión de movimientos al interpretar los comandos que hacen ubicarse en posiciones específicas para ubicar el papel así también seguir las trayectorias y efectuar el golpe del punzón para realizar el relieve.

Proceso de ejecución de los comandos

- Ubicar el robot en el origen
- Apaga mini compresor para evitar que este activado por error
- Servomotor a 0 grados esto evita que el actuador choque con arrastre la herramienta.
- Punzón Braille apagado evita enclavamiento al inicio del funcionamiento.
- Verifica si existe papel en la bandeja con sensor
- Traslada al cabezal de impresión a la posición X75Y350
- Servomotor a 60 grados para sujetar la hoja de papel
- Activar mini compresor succiona la hoja de papel
- Servomotor 45 grados levanta la hoja
- Ubicar al robot a la posición X75Y0 se dirige con la hoja de papel al área de trabajo.
- Servomotor a 60 grados baja la hoja de papel
- Desactivar mini compresor suelta la hoja
- Servomotor a 45 grados se recoge el actuador lineal
- Ubicar el robot en la posición X0Y0 se traslada al origen para inicial con la impresión
- Interpreta los comandos G y M para realizar las imagines en relieve.

En la Figura 21-2 se muestra el diagrama de flujo del proceso de ejecución de los comandos.

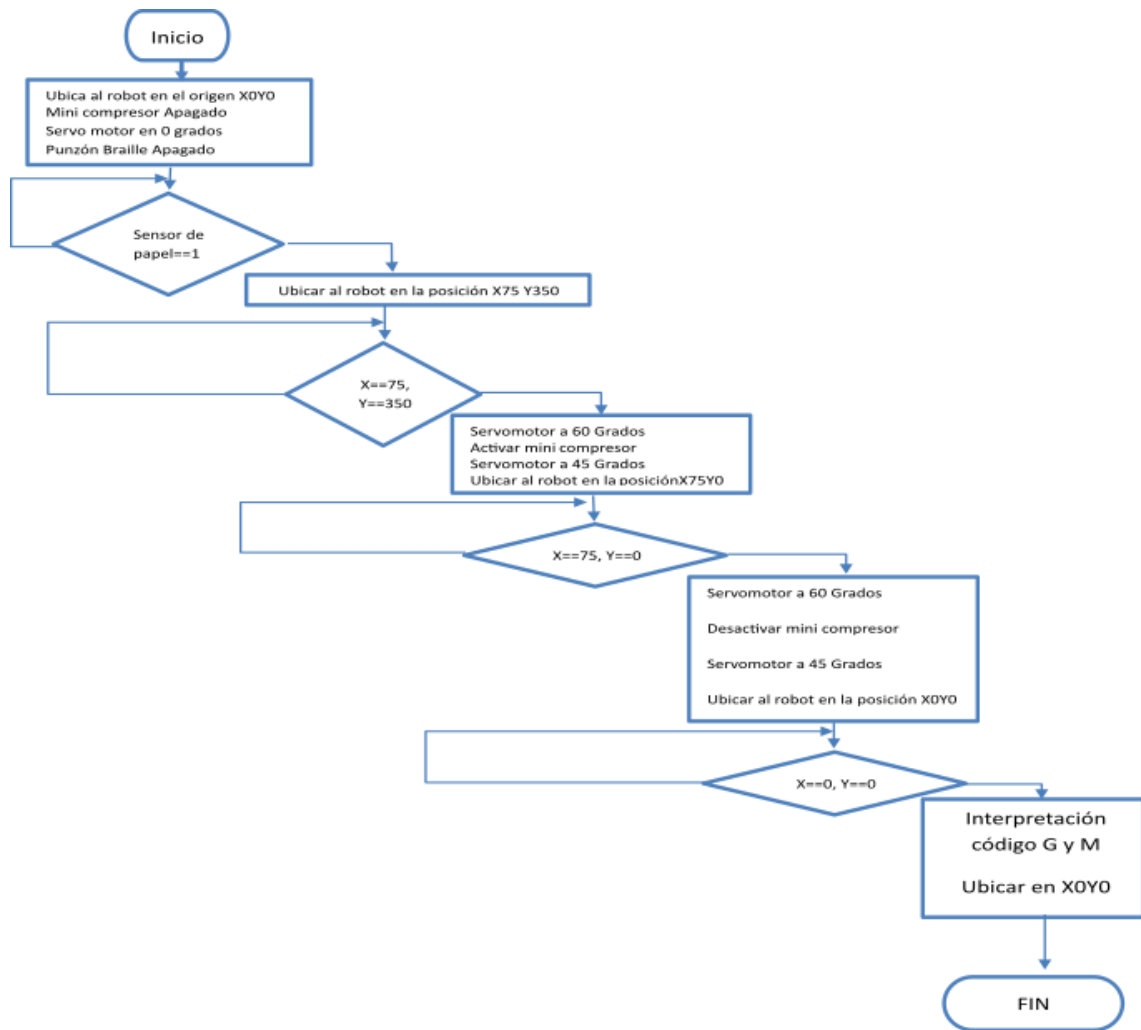


Figura 21-2: Diagrama de flujo de la ejecución de comandos.

Realizado por: (MUNGABUSI Angel, 2018)

2.8 Diseño mecánico para el Prototipo

Antes de elaborar la parte mecánica del prototipo se toma en consideración el área de trabajo donde la maquina tiene que desempeñar sus funciones para ello se realiza la selección de papel.

- Selección de papel

Para la elaboración de las imágenes en relieve se requiere de hojas de papel ya que es el material utilizado de manera común y más asequible en el mercado, cabe mencionar el papel debe tener un espesor de 120 gramos/m², el mismo que permite su durabilidad por tiempo de uso, también se debe tomar en cuenta que el papel seleccionado será en formato A4 ya que es una medida estándar a nivel mundial, permite determinar que el área de trabajo tiene de medidas 210x297 milímetros.

2.8.1 Estructura mecánica cartesiana para la implementación del prototipo

El sistema XY también se lo denomina pórtico horizontal, está compuesto por módulos individuales que mueve sus ejes de manera independiente, para generar un movimiento en 2D, esto permite al prototipo tener gran rigidez mecánica y estructura robusta para la aplicación el sistema es accionado por correa dentada ya que esto permite carreras muy largas a grandes velocidades a esto se añade que puede transportar cargas pesadas (FESTO, 2018).

En la Figura 22-2, se muestra el sistema pórtico horizontal.

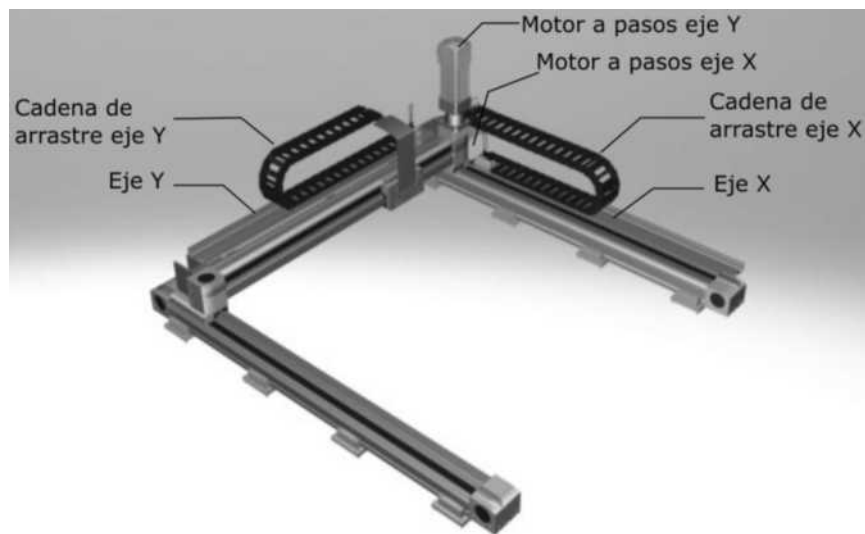


Figura 22-2: Estructura mecánica cartesiana pórtico horizontal.

Realizado por: (MUNGABUSI Angel, 2018)

2.8.3 Diseño y simulación de la estructura mecánica en SOLIDWORKS 2016

El diseño asistido por computadora (CAD), permite elaborar esquemas 2D y 3D del modelado mecánico, para el prototipo se ha optado por diseñar la estructura y realizar la impresión 3D de las piezas con esto se pretende un mejor acople del sistema mecánico de manera precisa y obtener resultados de acuerdo a los requerimientos necesarios del prototipo.

Utilizado una herramienta de software CAD llamada SolidWorks 2016, se caracteriza por su entorno de fácil manejo y su popularidad en el área académica (SOLIDWORKS, 2018).

En la Figura 23-2 se muestra el acople de varillas para el prototipo.

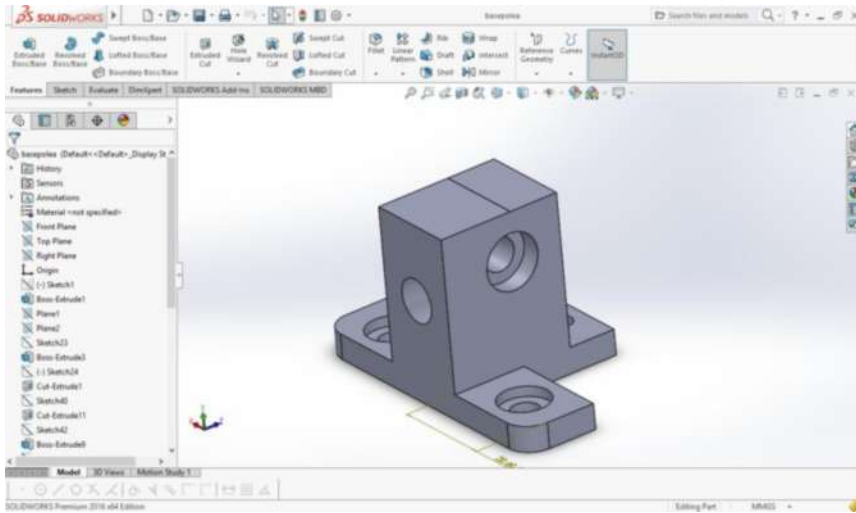


Figura 23-2: Acople de varilla en SolidWorks 2016.

Realizado por: (MUNGABUSI Angel, 2018)

- **Componentes mecánicos adicionales**

Para poder implementar la mecánica del prototipo se procedió a adquirir elementos adicionales como:

- Polea dentada GT2 (NaylampMecatronics, 2018a).
- Correa dentada GT2 (NaylampMecatronics, 2018a).
- Rodamientos lineales SC8UU (NaylampMecatronics, 2018b).
- Varilla lisa de hierro diámetro 8mm
- Cadena de arrastre

En la Figura 24-2 se aprecia el modelo 3D de la simulación mecánica del prototipo.

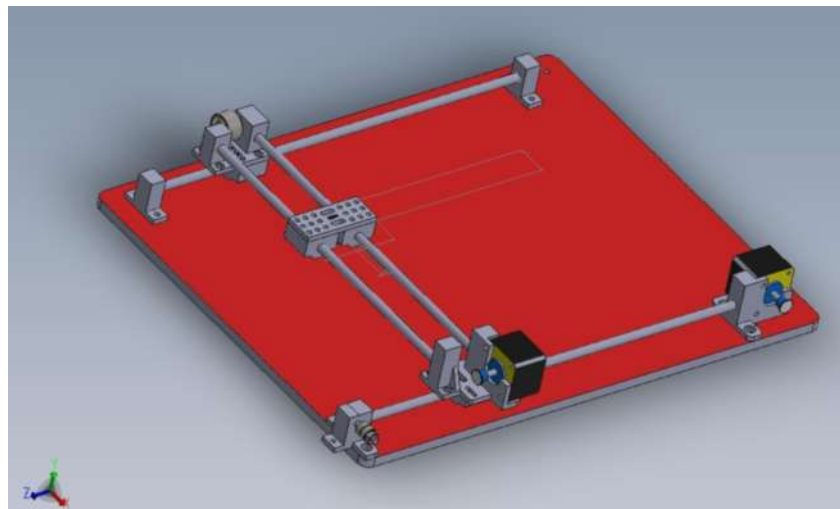


Figura 24-2: Simulación de la estructura mecánica del prototipo en SolidWorks 2016.

Realizado por: (MUNGABUSI Angel, 2018)

La impresión de las 7 piezas mecánicas que conforman la parte de sujeción de los ejes y motores del prototipo se realizaron en la impresora 3D de la marca ULTIMACHINE modelo Mendelmax2 en la Figura 25-2 se muestra su presentación.

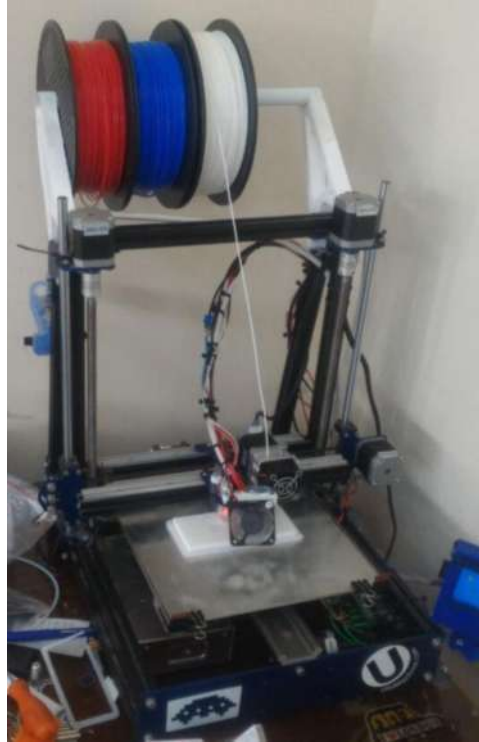


Figura 25-2: Impresora 3d Ultimachine Mendelmax2.

Realizado por: (MUNGABUSI Angel, 2018)

CAPÍTULO III

3. ANALISIS DE RESULTADOS DEL PROTOTIPO

En capítulo actual se compone de los resultados obtenidos tanto en funcionalidad y su impresión de imágenes tangibles y texto braille que cumplan con la normativa INEN 2850. Adicionalmente de presenta el análisis de costo, en la Figura 1-3 se muestra el prototipo implementado.



Figura 1-3: Implementación del prototipo final.

Realizado por: (MUNGABUSI Angel, 2018)

3.1 Pruebas de impresión del prototipo

En esta sección se presenta las pruebas de impresión tanto de imágenes en relieve así también como su texto braille que describe cada figura.

3.1.1 Prueba de gráficos en relieve

El sistema utilizado para realizar las imágenes en relieve se basa en el sistema de grabado o empastado esto deforma la hoja de papel para generar el relieve tanto de puntos para los gráficos así también para el texto braille.

Estos puntos y relieve están regidos por la normativa INEN 2850, Anexo L donde se especifica las características que debe poseer tanto una impresión en relieve como la escritura en Braille.

Para realizar las pruebas se va a tomar 20 muestras con calibrador universal para evaluar cada parámetro de gráficos y de símbolos Braille luego se procederá a comparar estos datos con la normativa INEN 2850 con esto se pretende conocer si el prototipo cumple con los requerimientos establecidos por la normativa para la elaboración de figuras en relieve y símbolos Braille para invidentes. El calibrador universal puede tomar medidas aproximadas teniendo en cuenta el error humano también la deformación de la hoja de papel al momento de realizar la medición. En la Figura 2-3 se muestra la imagen digital y su respectiva representación en figura en relieve (NTE INEN, 2014).

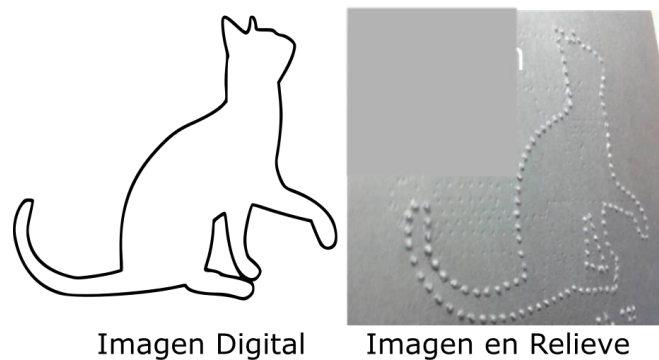


Figura 2-3: Gráfico digital y posterior impresión en relieve.

Realizado por: (MUNGABUSI Angel, 2018)

- **Prueba de distancia de ancho total entre puntos en relieve**

Las medidas tomadas con un calibrador universal se muestran en la Figura 3-3.

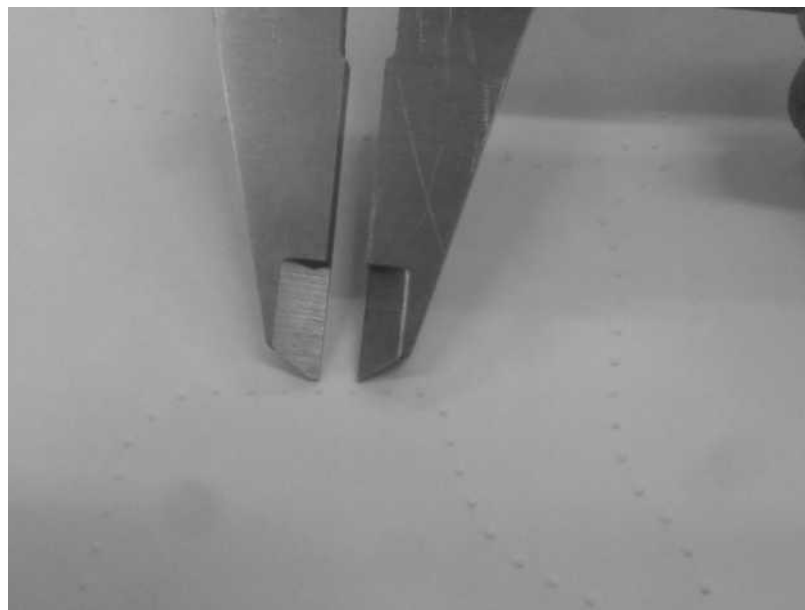


Figura 3-3: Toma de medida de ancho total entre puntos en relieve.

Realizado por: (MUNGABUSI Angel, 2018)

Se tomaron las muestras las cuales se fueron comparando con la normativa INEN 2850 y su representación se muestra en el Gráfico 1-3

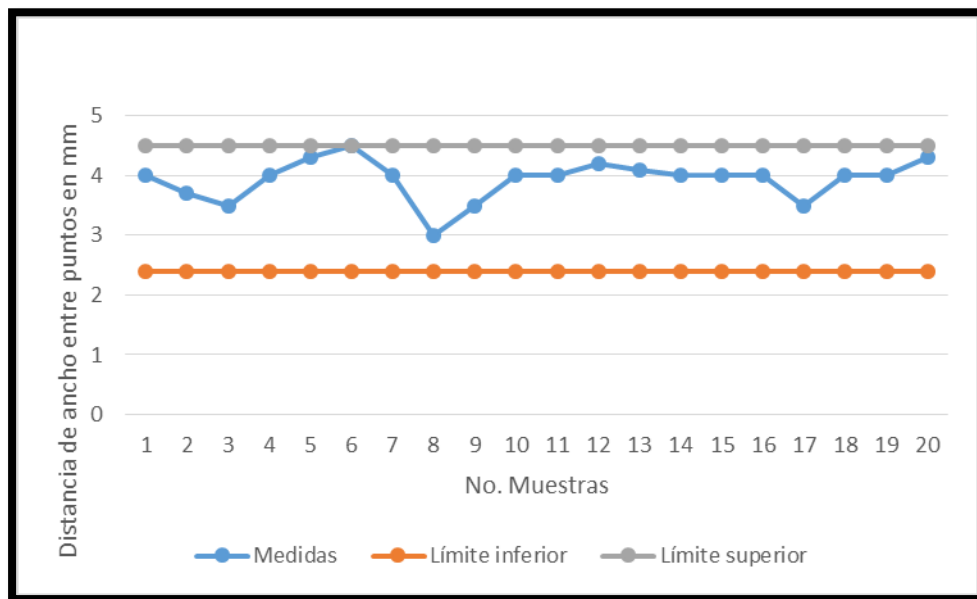


Gráfico 1-3: Relación de medidas de ancho entre puntos respecto a límites superior e inferior de la normativa.

Realizado por: (MUNGABUSI Angel, 2018)

El Gráfico 1-3 muestra las medidas de ancho entre puntos tomados de la Figura 3-3 y representado en paralelo con los límites por normativa INEN 2850, que ratifica la correcta elaboración de dichos puntos y que el prototipo cumple con los requerimientos establecidos por normativa.

- Prueba de distancia entre centros de puntos contiguos

Permite delimitar los espacios entre puntos contiguos para una mayor resolución de las imágenes, Figura 4-3 muestra la toma de medidas para su posterior análisis.

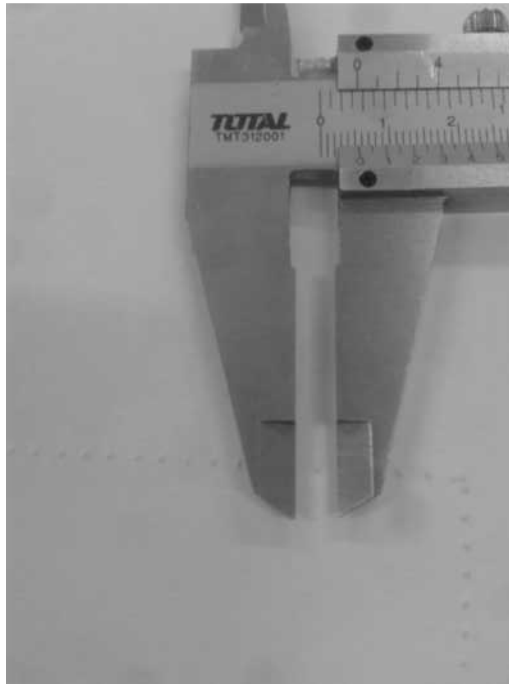


Figura 4-3: Medida de ancho de centro de puntos con calibrador universal.

Realizado por: (MUNGABUSI Angel, 2018)

Gráfico 2-3 representa las muestras tomadas y relacionadas en un gráfico de control a los límites superior e inferior de la normativa INEN 2850 para la distancia entre puntos contiguos.

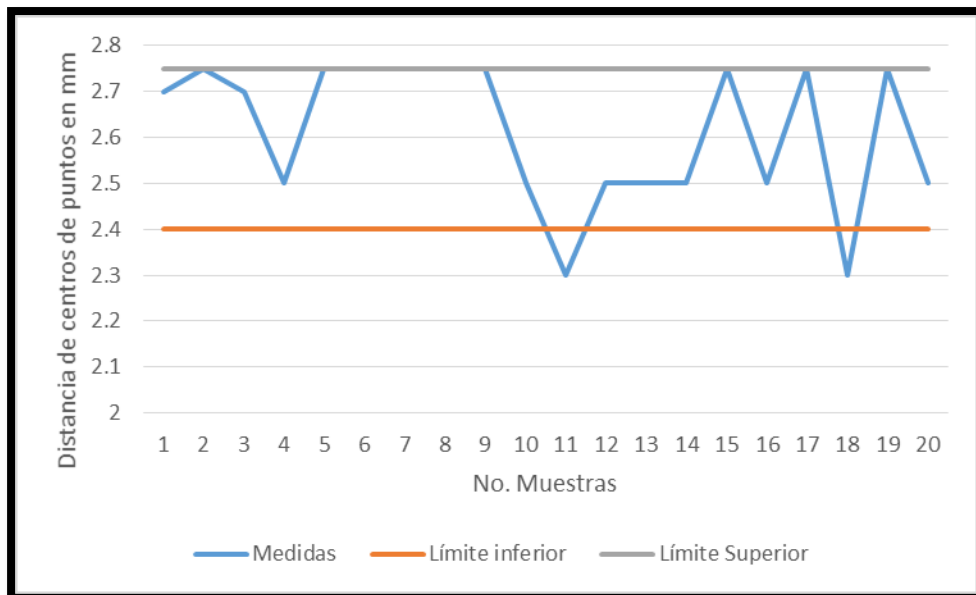


Gráfico 2-3: Relación de medidas distancia entre puntos contiguos.

Realizado por: (MUNGABUSI Angel, 2018)

En el Gráfico 2-3 se observa que dos muestras que representan el 10% del total están por debajo del límite de distancia de centro de punto, esto no representa ningún inconveniente en la gráfica tangible ya que esto incrementa la resolución, es por el hecho que las imágenes en relieve pueden ser representadas por una serie de puntos o una línea en relieve continua.

- **Prueba de relieve de los puntos**

El relieve de las imágenes es determinante ya que el tacto tiene que percibir estos y poder interpretar, es por esa razón que según la normativa INEN 2850 la altura mínima del relieve es de 0.5 mm, se ha realizado mediciones con el calibrador universal mostrado en la Figura 5-3.

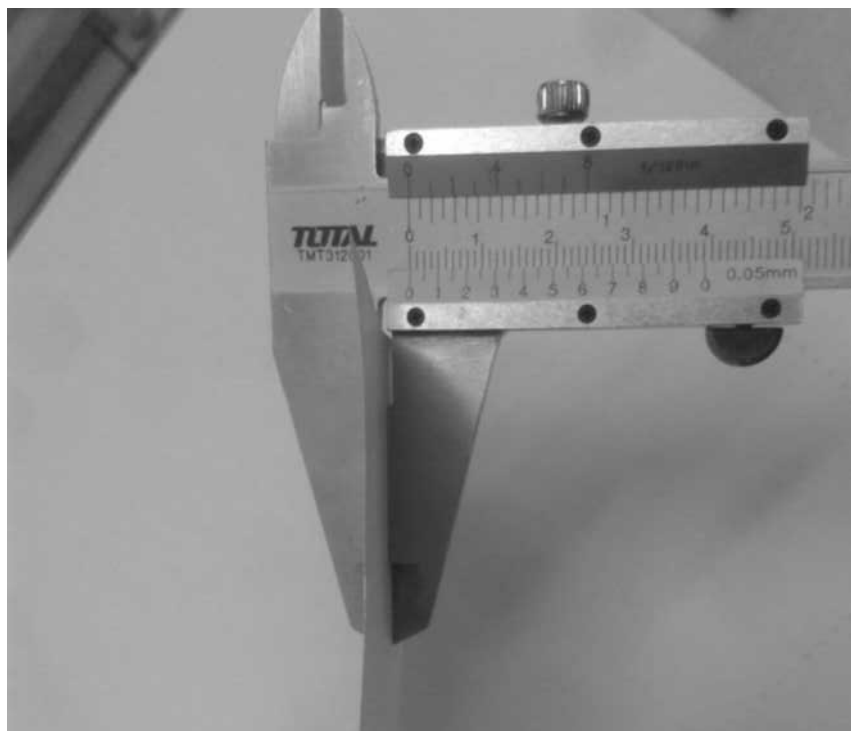


Figura 5-3: Medida de alto de relieve con calibrador universal

Realizado por: (MUNGABUSI Angel, 2018)

En la Gráfica 3-3 se muestran las 20 medidas del alto de relieve en paralelo con el mínimo permitido por normativa INEN 2850. .

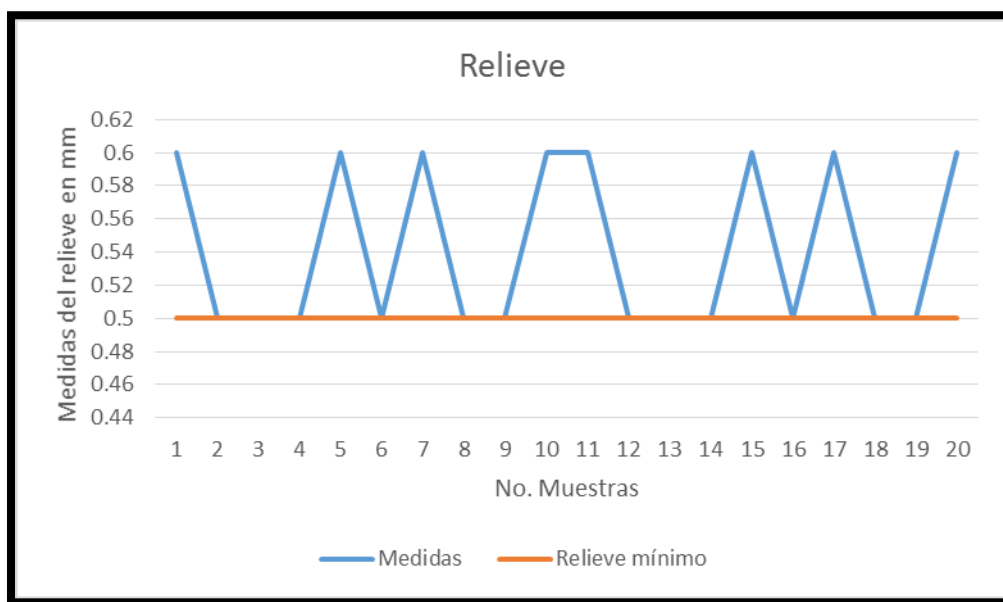


Gráfico 3-3: Relación de medidas respecto al mínimo del relieve regida por normativa INEN 2850.

Realizado por: (MUNGABUSI Angel, 2018)

Mediante el Gráfico 3-3 se puede ratificar el correcto funcionamiento del prototipo al momento de elaborar el alto relieve ya que está por encima de la altura mínima, esto permite tener mayor sensibilidad al tacto al instante de ser percibido por el invidente.

- **Prueba de diámetro de la base de los puntos**

Depende de la base de impresión donde el solenoide que se usa como punzón braille golpea y deforma la hoja de papel, para ello también se procedió a limar la parte del émbolo del solenoide y obtener la dimensión adecuada como punzón braille, al realizar las pruebas respectivas se determinó que el ángulo de ataque debe tener 70 grados, esto permite que el punzón no se trabe con la hoja de papel. En la Figura 6-3 se muestra la medida tomada para su posterior análisis.



Figura 6-3: Medida de ancho de base de punto con calibrador universal.
 Realizado por: (MUNGABUSI Angel, 2018)

En el Gráfico 4-3 se ha tomado muestra y representado en una gráfica de líneas.

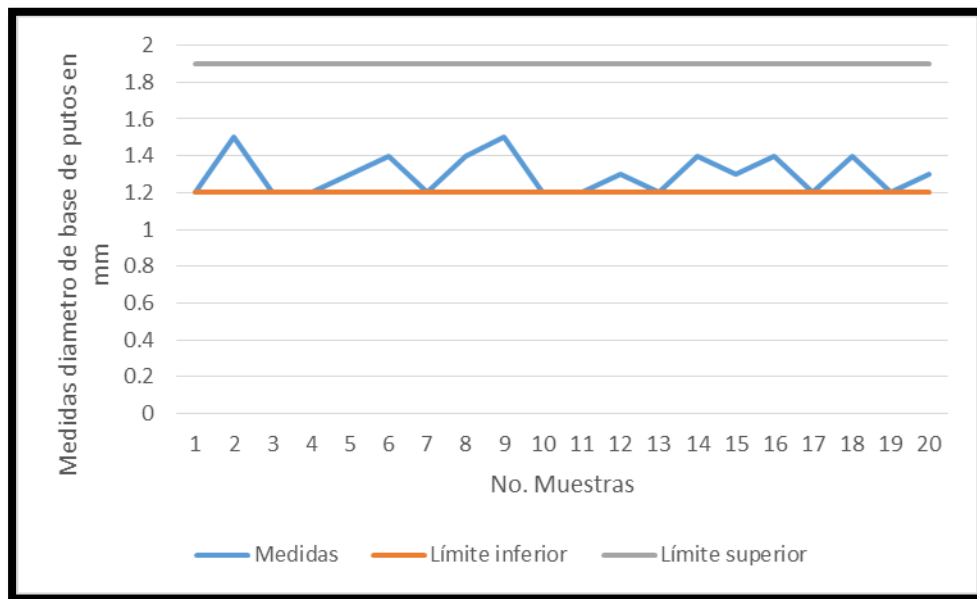


Gráfico 4-3: Medidas de la base de puntos respecto a la normativa INEN 2850 en límite máximo y mínimo.

Realizado por: (MUNGABUSI Angel, 2018)

Las medidas tomadas y representadas en comparación a los límites por normativa en el Gráfico 4-3, fueron realizadas con una superficie de impresión de madera esto permite la permanencia del diámetro de la base de punto respecto al límite inferior por normativa.

3.1.2 Prueba de impresión de escritura braille

En la normativa INEN 2850 se menciona que los símbolos o para el caso de estudio figuras en relieve deben tener escrito su en braille su significado, en la Figura 7-3 se muestra la figura de un perro y en la parte inferior el respectivo simbología en braille.

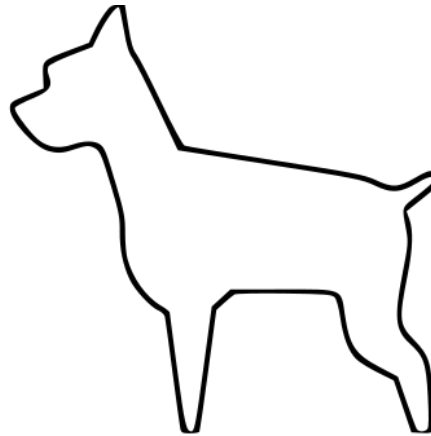


Figura 7-3: Imagen con descripción en braille.

Realizado por: (MUNGABUSI Angel, 2018)

El Sistema braille es indispensable para los invidentes ya que describe la imagen y de esa manera pueden obtener la información y generar una imagen mental en este caso la Figura 7-3 de lo que es un perro, el prototipo puede realizar este lenguaje la normativa así lo requiere, basado en esto se procede a realizar las diferentes pruebas. En la Figura 8-3 se muestra los símbolos Braille generados por el prototipo.



Figura 8-3: Símbolos Braille generado por el prototipo.

Realizado por: (MUNGABUSI Angel, 2018)

- **Prueba distancia horizontal entre centros de punto continuo**

Las pruebas para el lenguaje Braille se realiza igual que las pruebas de gráficos tangibles se toman 20 muestras con el calibrador universal y su posterior análisis. La medida para la distancia horizontal de centro de punto continuo se muestra en la Figura 9-3.



Figura 9-3: Medida de distancia de centro de punto horizontal con calibrador universal.

Realizado por: (MUNGABUSI Angel, 2018)

Gráfico 8-3 se muestra la distancia horizontal entre centros de puntos medidos respecto a los límites por normativa INEN 2850 puesta en paralelo.

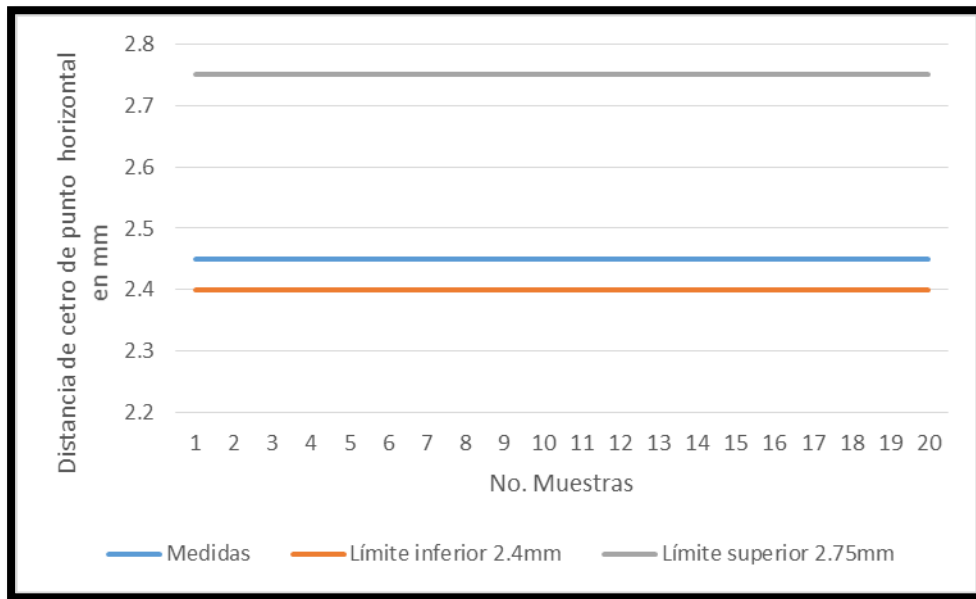


Gráfico 5-3: Medidas de centro de punto horizontal respecto de límites regidos por normativa.

Realizado por: (MUNGABUSI Angel, 2018)

La Gráfica 5-3, ratifica la exactitud y precisión de la elaboración de centro de punto horizontal de un símbolo braille con una efectividad del 100%.

- **Prueba distancia vertical entre centros de puntos**

La Figura 10-3, muestra la toma de medidas de la distancia vertical de puntos con calibrador universal.

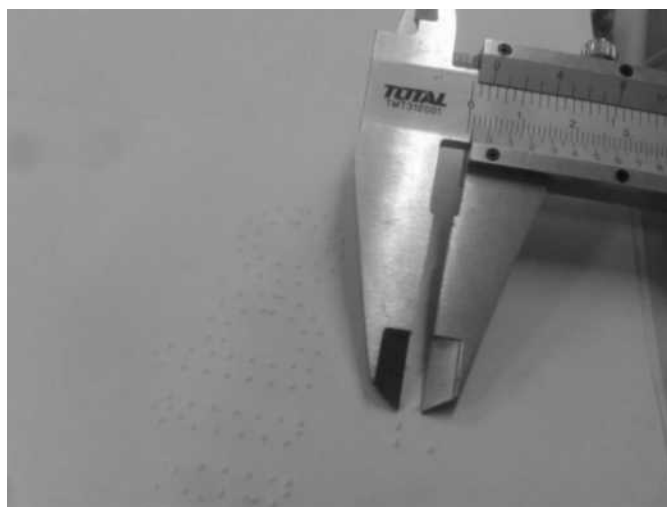


Figura 10-3: Medida de distancia vertical entre centros de puntos tomados por el calibrador universal.

Realizado por: (MUNGABUSI Angel, 2018)

Gráfico 6-3 representa una comparación de las muestras tomadas respecto a los límites de la normativa INEN 2850.

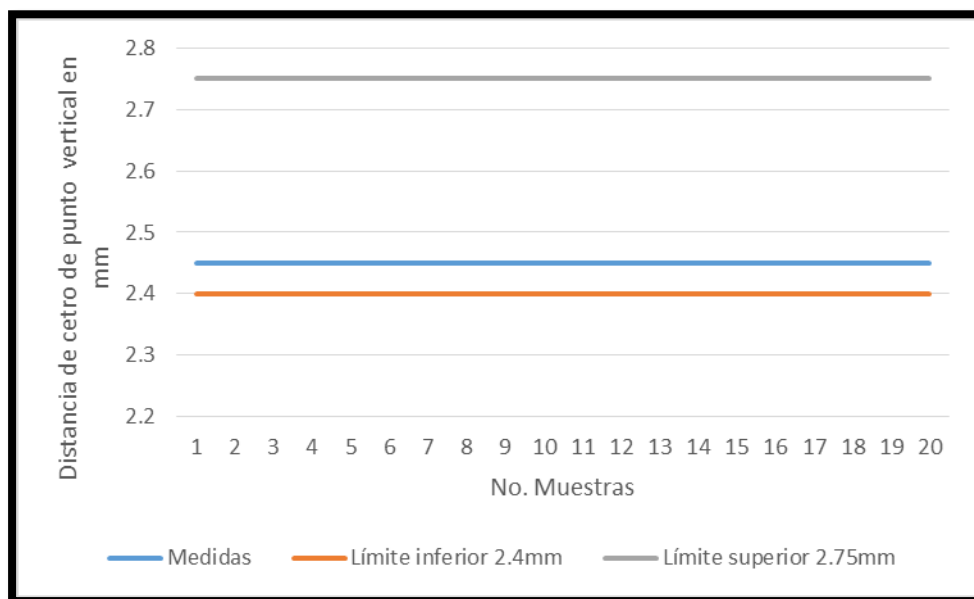


Gráfico 6-3: Medidas de centro de punto vertical respecto a límites regidos por normativa.

Realizado por: (MUNGABUSI Angel, 2018)

En la Gráfico 6-3, se denota la medida obtenida es de 2.45mm y es la constante mediante todas las mediciones realizadas con esto se comprueba que la escritura braille realizada por el prototipo cumple con los requerimientos de elaboración de escritura braille.

- **Prueba distancia entre los centros de puntos idénticos de celdas contiguas**

Esta medida es determinante ya que el sentido táctil es inferior al visual es por esa razón que cada celda del sistema braille debe estar separado con la misma distancia, ya que de no ser así se puede confundir las letras del lenguaje Braille. La toma de muestras con el calibrador universal se observa en la Figura 11-3.

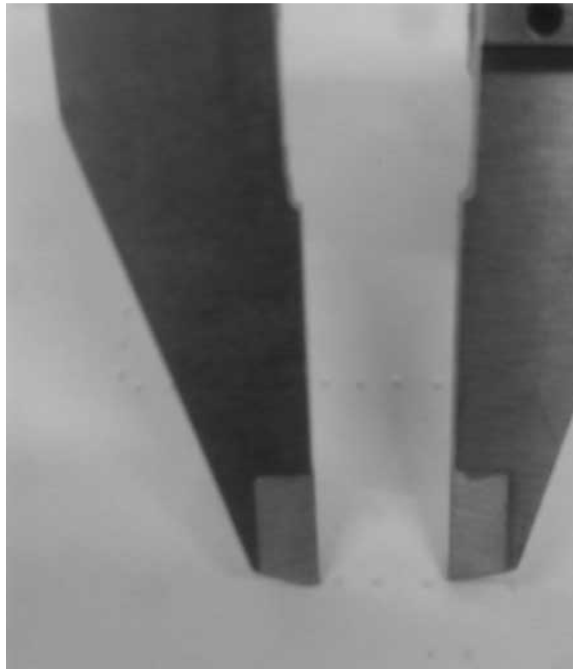


Figura 11-3: Toma de medida de centro de puntos idénticos celdas contiguas con calibrador universal.

Realizado por: (MUNGABUSI Angel, 2018)

En el Gráfico 7-3 se muestra las medidas realizadas con respecto a la normativa la cual mantiene un margen tanto en el límite superior e inferior para el caso de seis milímetros y seis punto nueve respectivamente.

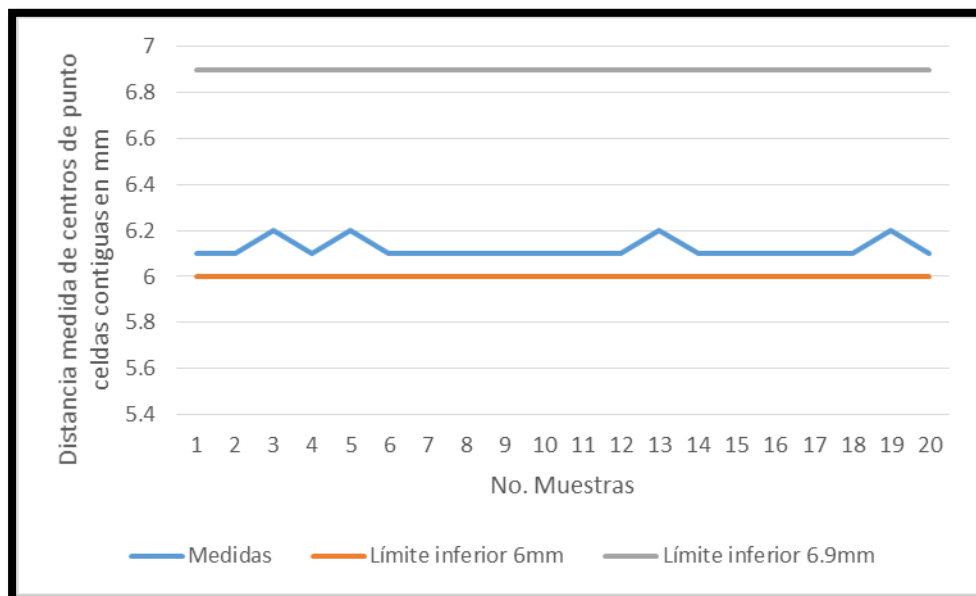


Gráfico 7-3: Medidas centro de punto celdas contiguas respecto a los límites de normativa INEN 2850.

Realizado por: (MUNGABUSI Angel, 2018)

Las distancias medidas de los centros de puntos de celdas contiguas mostradas en el Gráfico 7-3 respecto de la normativa tiene una variabilidad con una diferencia de 0.1 milímetros respecto del límite inferior.

- **Prueba diámetro de la base de puntos**

Valor entre 1.2mm a 1.9mm por la normativa INEN 2850. En la Figura 12-3 se muestra la toma de medidas de la base de puntos.



Figura 12-3: Medida de base de puntos toma de muestras con calibrador universal.

Realizado por: (MUNGABUSI Angel, 2018)

En la Gráfica 8-3 se muestra las medidas tomadas de forma que se pueda comparar con los límites superior e inferior de la normativa.

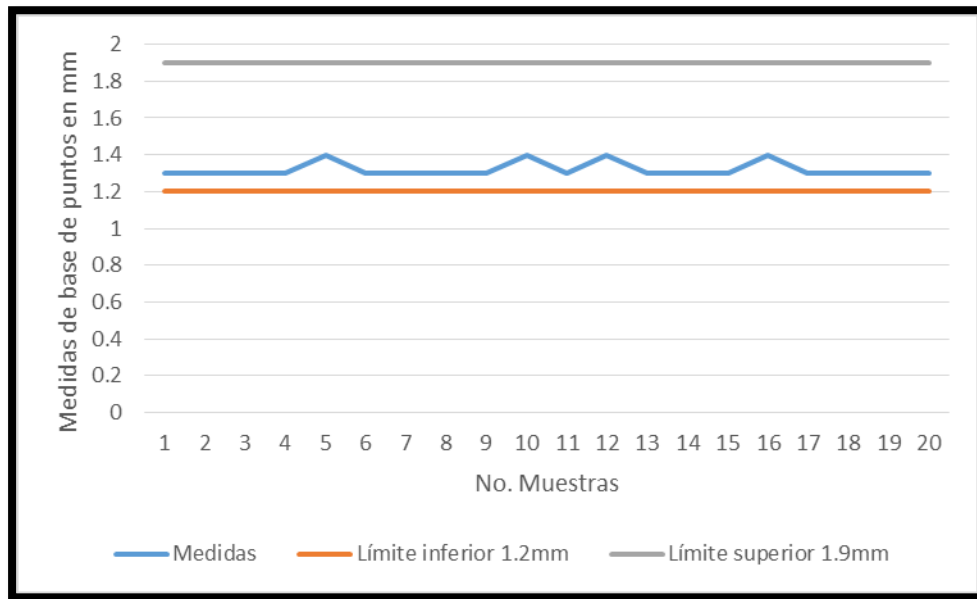


Gráfico 8-3: Diámetro de la base de puntos en relación a límites de la normativa INEN 2850.

Realizado por: (MUNGABUSI Angel, 2018)

El Gráfico 8-3 muestra las medidas tomadas una tendencia al límite inferior de 1.2 milímetros con una diferencia de 0.1 mm con esto se ratifica que la generación de la base de puntos se encuentra dentro de los rangos especificados por normativa.

- **Prueba relieve de los puntos.**

La prueba consiste en tomar 20 muestras con un calibrador universal su presentación se observa en la Figura13-3.

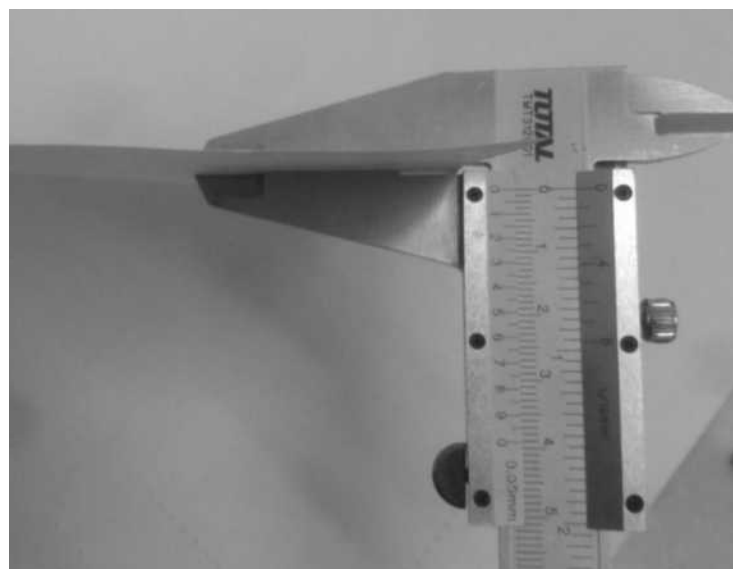


Figura 13-3: Medida de relieve de puntos con calibrador universal.

Realizado por: (MUNGABUSI Angel, 2018)

En la Gráfico 9-3 la comparativa en base al límite inferior del alto relieve.

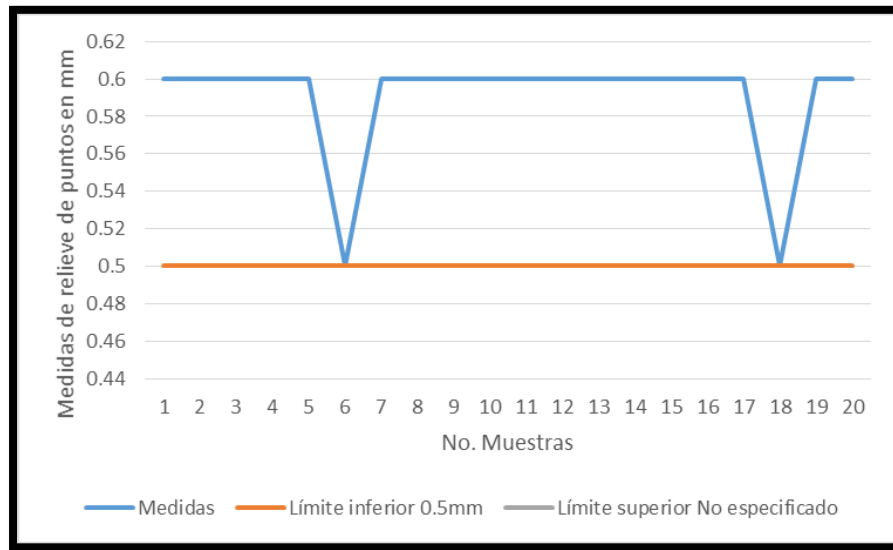


Gráfico 9-3: Medidas de relieve de punto con respecto al mínimo permitido por norma.

Realizado por: (MUNGABUSI Angel, 2018)

Gráfico 9-3 representa de manera clara que el relieve generado por el prototipo es superior a la medida mínima permitida por normativa con esto se concluye que cumple con los requerimientos.

- Prueba Ancho y Alto

En la Figura 14-3 se aprecia la toma de muestras para el ancho del símbolo braille.



Figura 14-3: Toma de muestras del ancho del símbolo braille.

Realizado por: (MUNGABUSI Angel, 2018)

Gráfico 10-3 donde se muestra el resultado de las mediciones y representadas en paralelo con los límites por normativa INEN 2850.

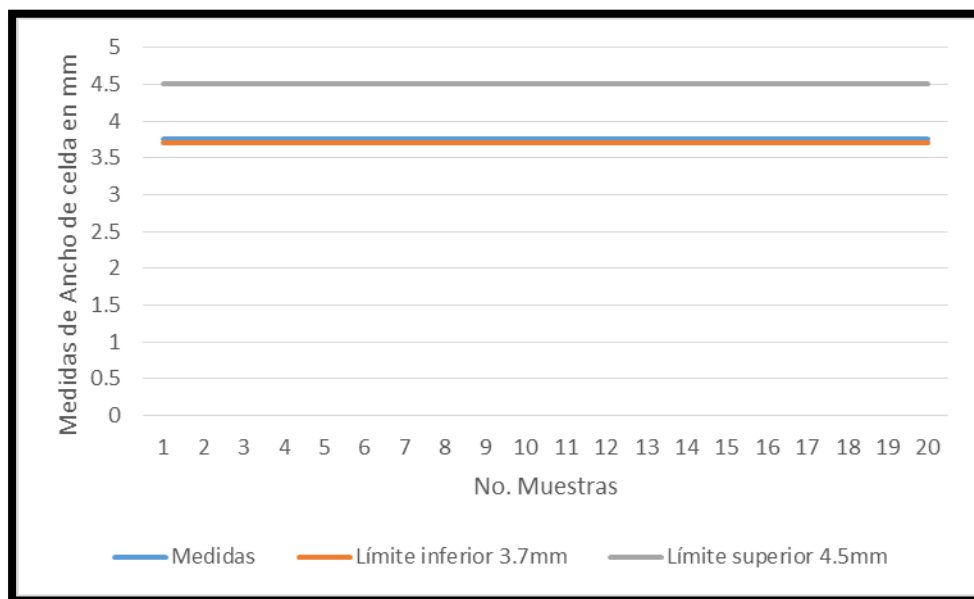


Gráfico 10-3: Resultado de mediciones de ancho de celda.

Realizado por: (MUNGABUSI Angel, 2018)

El Gráfico 10-3 muestra dos líneas horizontales que básicamente se solapan esto entre los datos medidos y el límite inferior de ancho de celda con esto se concluye que las pruebas anteriores ratifican efectividad en la elaboración de escritura braille.

- **Prueba del alto de celda**

Figura 15-3 muestra la toma de medida del alto de la celda.

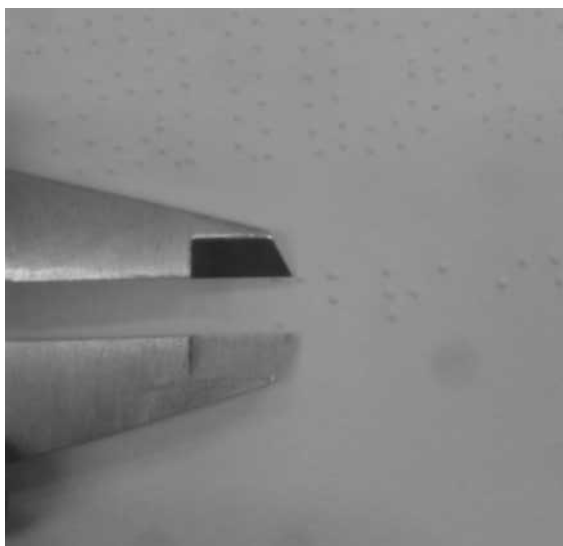


Figura 15-3: Toma de muestras del alto del símbolo braille.

Realizado por: (MUNGABUSI Angel, 2018)

El Gráfico 11-3 muestra el resultado de las medidas del alto.

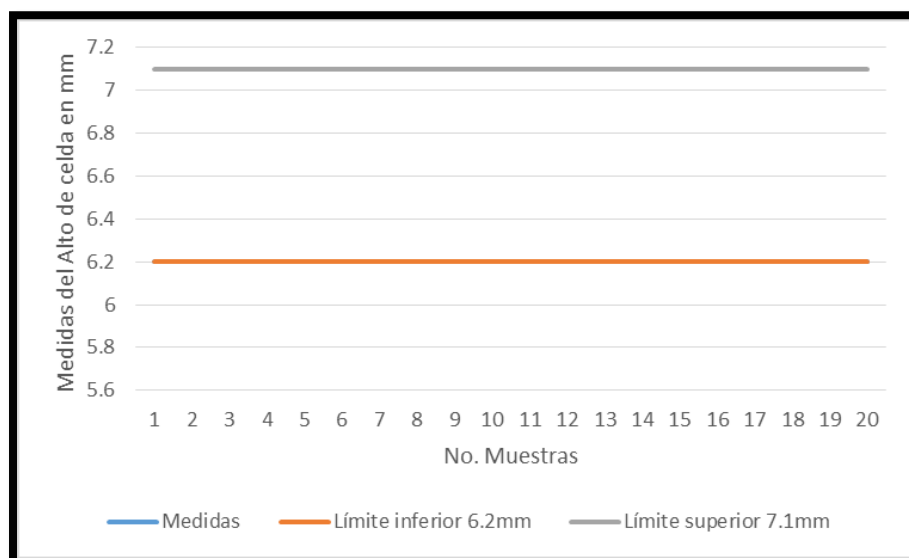


Gráfico 11-3: Resultado de medidas del alto de la celda con respecto a límites de normativa.

Realizado por: (MUNGABUSI Angel, 2018)

El Gráfico 11-3 muestra que para el caso las medidas tomadas se sobrepone con el límite inferior permitido por la normativa esto a un tamaño de letra de 20 puntos que ratifica la precisión y exactitud de la elaboración de texto Braille.

3.1.3 Prueba de estabilidad del prototipo

Se imprime nueve veces una imagen y texto braille luego se procede a tomar medidas de un punto y la distancia entre puntos. Se procede a determinar el coeficiente de variación porcentual, considerando que a menor sea este valor mayor homogeneidad en los valores de la variable medida con esto se prueba la repetitividad precisión y exactitud del prototipo (Walpole, 2012, p. 15).

- X: Distancia respecto del borde lateral hacia el punto generado.
- Y: Distancia respecto del borde inferior hacia el punto generado.
- A: Distancia idénticas entre celdas contiguas.
- B: Diámetro de la base de punto.
- C: centro de punto horizontal
- D: centro de punto vertical
- E: Alto de la celda
- F: Ancho de la celda
- R: Relieve del punto.

En la Figura 16-3 se aprecia toma de medidas para la figura en relieve.

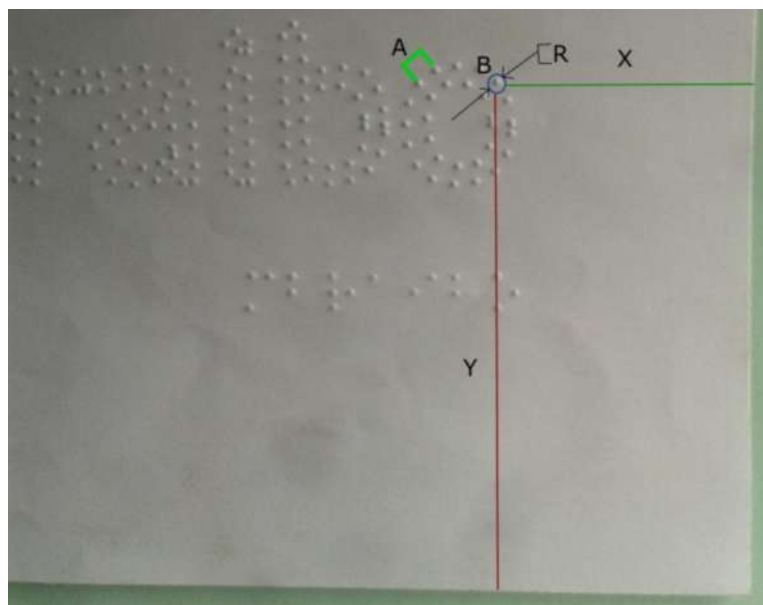


Figura 16-3: Toma de medidas de parámetros para figura.

Realizado por: (MUNGABUSI Angel, 2018)

La Tabla 1-3 muestra el resultado de la toma de medidas de los parámetros indicados y resultados del análisis estadístico.

Tabla 1-3: Análisis de las muestras tomadas de las figuras en relieve para el estudio estadístico.

| N° | X(mm) | Y(mm) | A(mm) | B(mm) | R(mm) |
|----------------------------|--------------|--------------|--------------|--------------|--------------|
| 1 | 42 | 85 | 2.7 | 1.2 | 0.6 |
| 2 | 42 | 84.5 | 2.5 | 1.2 | 0.5 |
| 3 | 41 | 83 | 2.6 | 1.2 | 0.5 |
| 4 | 42 | 80 | 2.7 | 1.1 | 0.6 |
| 5 | 41.5 | 85 | 2.5 | 1.2 | 0.5 |
| 6 | 41 | 84.5 | 2.5 | 1.2 | 0.5 |
| 7 | 42 | 84 | 2.6 | 1.2 | 0.6 |
| 8 | 43 | 82 | 2.7 | 1.2 | 0.6 |
| 9 | 42 | 83 | 2.6 | 1.2 | 0.6 |
| Valor medio | 41.833 | 83.444 | 2.600 | 1.189 | 0.556 |
| Desviación Estandar | 0.612 | 1.648 | 0.087 | 0.033 | 0.053 |
| Coef de variación % | 1.464 | 1.975 | 3.331 | 2.804 | 9.487 |

Realizado por: (MUNGABUSI Angel, 2018)

El análisis de la Tabla 1-3, se puede observar que el coeficiente de variación porcentual va de 1.47% a 9.48%, lo que representa que existe igualdad entre los valores del conjunto de datos

analizado. Se puede concluir que en el proceso de impresión de las imágenes, el prototipo implementado es estable.

En la Figura 17-3, se muestra los parámetros adicionales que se deben medir para evaluar la estabilidad en la elaboración de texto braille como son:

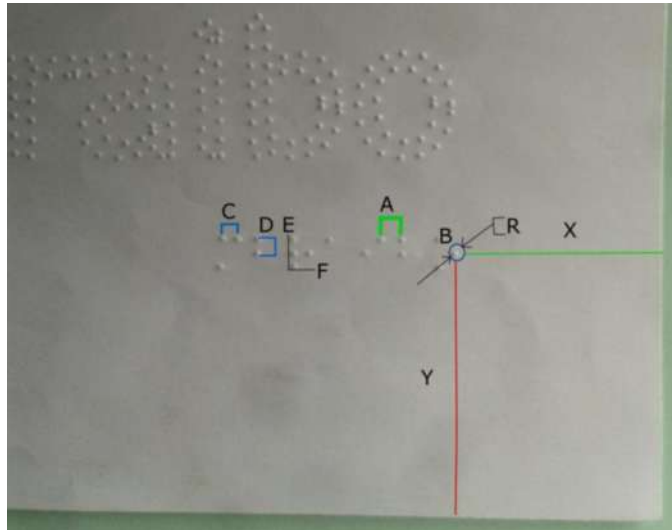


Figura 17-3: Toma de medidas de parámetros para simbología braille.

Realizado por: (MUNGABUSI Angel, 2018)

En la Tabla 2-3 muestra la toma de medidas de los parámetros en milímetros indicados en la Figura 17-3 y resultados del análisis estadístico para determinar variabilidad en la repetitividad del proceso de impresión.

Tabla 2-3: Análisis de las muestras tomadas de símbolos braille para el estudio estadístico.

| N° | X(mm) | Y(mm) | A(mm) | B(mm) | C(mm) | D(mm) | E(mm) | F(mm) | R(mm) |
|----------------------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
| 1 | 39 | 50 | 4 | 1.2 | 2.5 | 2.4 | 6.2 | 4.1 | 0.6 |
| 2 | 39 | 49.5 | 4 | 1.3 | 2.5 | 2.45 | 6.2 | 4.1 | 0.6 |
| 3 | 40 | 48 | 4 | 1.2 | 2.5 | 2.4 | 6.3 | 4.1 | 0.6 |
| 4 | 38.5 | 49 | 4 | 1.2 | 2.5 | 2.4 | 6.2 | 4.1 | 0.6 |
| 5 | 39 | 50 | 4 | 1.3 | 2.5 | 2.45 | 6.2 | 4.1 | 0.6 |
| 6 | 40 | 50.5 | 4 | 1.2 | 2.5 | 2.4 | 6.2 | 4.1 | 0.6 |
| 7 | 39.5 | 49 | 4 | 1.2 | 2.5 | 2.4 | 6.2 | 4.1 | 0.6 |
| 8 | 39 | 48 | 4 | 1.3 | 2.5 | 2.45 | 6.2 | 4.1 | 0.6 |
| 9 | 40 | 49 | 4 | 1.3 | 2.5 | 2.45 | 6.2 | 4.1 | 0.6 |
| Valor medio | 39.333 | 49.222 | 4.000 | 1.244 | 2.500 | 2.422 | 6.211 | 4.100 | 0.600 |
| Desviación Estandar | 0.559 | 0.870 | 0.000 | 0.053 | 0.000 | 0.026 | 0.033 | 0.000 | 0.000 |
| Coef de variación % | 1.421 | 1.768 | 0.000 | 4.235 | 0.000 | 1.088 | 0.537 | 0.000 | 0.000 |

Realizado por: (MUNGABUSI Angel, 2018)

El análisis de la Tabla 2-3, se puede que el coeficiente de variación porcentual el 0% a 4.23% obtenido, representa que poca variabilidad entre los valores del conjunto de datos analizado como conclusión que al repetir el proceso de impresión de escritura Braille se puede decir que el prototipo es estable ante la repetitividad del proceso.

3.1.4 Prueba de legibilidad de escritura Braille y figuras en relieve del prototipo vs impresora braille.

El propósito de la prueba es validar si la escritura braille y la figura en relieve elaborada por el prototipo son legibles a la percepción táctil. Se procede a realizar tres impresiones de escritura Braille y figura en relieve por el prototipo e impresora braille, posteriormente se evalúa a cinco personas con discapacidad visual si pueden o no identificar claramente el documento, luego se tabula y se halla el porcentaje personas que pudieron identificar la escritura Braille y figura en relieve elaborada por el prototipo versus la impresora braille.

La Tabla 3-3 representa las personas evaluadas para determinar la legibilidad de la escritura braille y las figuras en relieve elaboradas por el prototipo versus una impresora braille.

Donde:

F1P, F2P, F3P: Figura relieve elaborado por el prototipo

F1I, F2I, F3I: Figura relieve elaborado por impresora braille

E1P, E2P, E3P: Escritura Braille elaborado por el Prototipo

E1P, E2P, E3P: Escritura Braille elaborado por impresora braille

Tabla 3-3: Evaluación de legibilidad

| Figura y escritura | 1 | 2 | 3 | 4 | 5 | TOTAL % legible por Figura y Escritura braille | |
|------------------------------------|------|------|-----|------|-----|---|-----------|
| F1P | 1 | 1 | 1 | 1 | 1 | 100 | |
| F2P | 1 | 1 | 1 | 1 | 1 | 100 | |
| F3P | 1 | 1 | 1 | 1 | 1 | 100 | |
| Total % legible FP | | | | | | 100 | |
| F1I | 1 | 1 | 1 | 1 | 1 | 100 | |
| F2I | 1 | 0 | 1 | 1 | 1 | 80 | |
| F3I | 1 | 1 | 1 | 1 | 1 | 100 | |
| Total % legible FI | | | | | | 93.3 | |
| E1P | 1 | 1 | 1 | 1 | 1 | 100 | |
| E2P | 1 | 1 | 1 | 1 | 1 | 100 | |
| E3P | 1 | 1 | 1 | 1 | 1 | 100 | |
| Total % legible EP | | | | | | 100 | |
| E1I | 1 | 1 | 1 | 1 | 1 | 100 | |
| E2I | 1 | 1 | 1 | 1 | 1 | 100 | |
| E3P | 1 | 1 | 1 | 1 | 1 | 100 | |
| Total % legible FI | | | | | | 100 | |
| TOTAL % legible por persona | 91.6 | 91.6 | 100 | 91.6 | 100 | 94.96 | 95 |

Realizado por: (MUNGABUSI Angel, 2018)

De la prueba se puede concluir que el 100% de las personas con discapacidad visual evaluadas pudieron comprender la escritura braille y la figura en relieve elaborada por el prototipo, en contra posición del 93 - 100% con la impresora braille respectivamente. También se puede decir que para evaluar las figuras en relieve se requiere de agudizar la percepción táctil y conocimiento adquirido.

3.1.5 Prueba de comparación de figura patrón con la resultante

Consiste en imprimir a tinta en una hoja de papel una figura, luego imprimir la misma figura con el prototipo y evaluar si las líneas en relieve, así como la de tinta se sobreponen de esta manera se verifica la exactitud y precisión de la impresión de figuras en relieve. En la Figura 18-3 se muestra el método de evaluación de la prueba.

Se puede concluir que las figuras en relieve como la de impresa a tinta se solapan en una sola imagen con ello se puede decir que el prototipo realiza con exactitud y precisión la impresión.

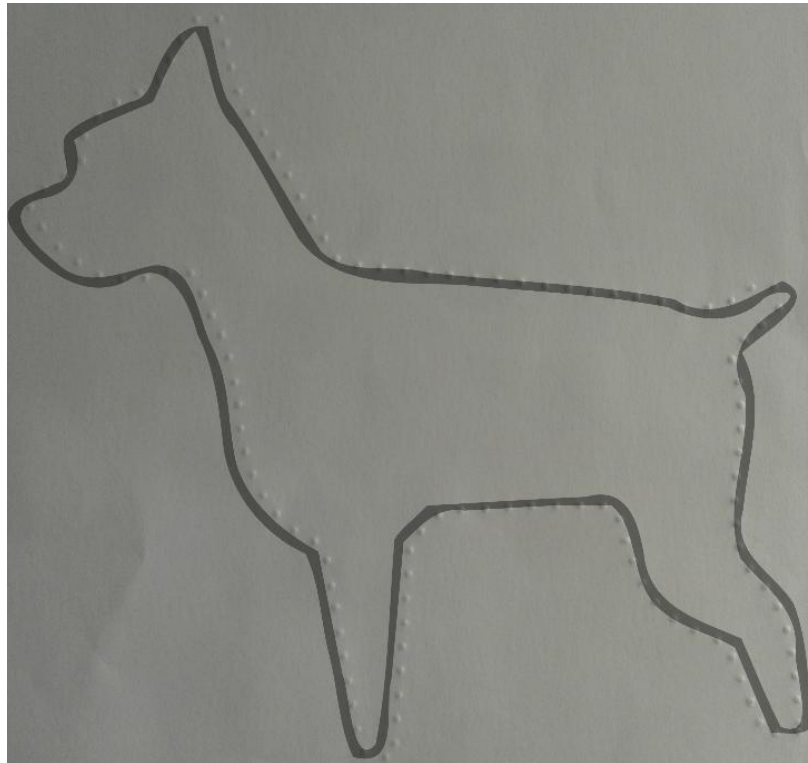


Figura 18-3: Sobre posición de imágenes a tinta con relieve

Realizado por: (MUNGABUSI Angel, 2018)

3.2 Tiempo de impresión de imágenes en relieve

A diferencia de algunas impresoras comerciales que tiene como matriz de generación los puntos conformados para toda la plantilla de la hoja, el prototipo utiliza de forma punto flotante esto quiere decir que se puede realizar punto en relieve en cualquier parte de la hoja de papel.

El tiempo es determinado por la suma procesos iniciales que realiza el prototipo antes de comenzar con la ejecución de la impresión. Se a toma el tiempo de cada ejecución es por ello que se tiene la siguiente Tabla 4-3 donde se muestra los valores de tiempo estimado con tiempo real.

Tabla 4-3: Tiempos de impresión de imágenes en relieve y texto.

| Parámetro | Tiempo de impresión estimado (h:m:s) | Tiempo de impresión real (h:m:s) | Diferencia de tiempo por impresión (%) |
|---------------------------------|---|---|---|
| Impresión de imagen (120x120mm) | 00:01:00 | 00:00:45 | -25% |
| Impresión por carácter Braille | 00:00:05 | 00:00:03 | -40% |
| Ubicación de hoja de papel | 00:01:00 | 00:00:40 | -33% |
| Posicionamiento del servomotor | 00:00:01 | 00:00:005 | -65% |
| TOTAL | 00:02:06 | 00:01:27 | -40.75% |

Realizado por: (MUNGABUSI Angel, 2018)

En la Tabla 4-3 muestra el tiempo real de impresión por el prototipo es menor en un 40.75% en promedio respecto al tiempo estimado. En la Figura 19-3 se muestra la impresión de gráfico y símbolo braille. Parte de esto se puede calcular que para imprimir una hoja completa solo de texto braille tendría un tiempo estimado de 30 minutos con 48 segundos, tienen en cuenta que una hoja tiene 24 filas de 29 caracteres cada una para completar la hoja pero el tiempo real es de 18 minutos.

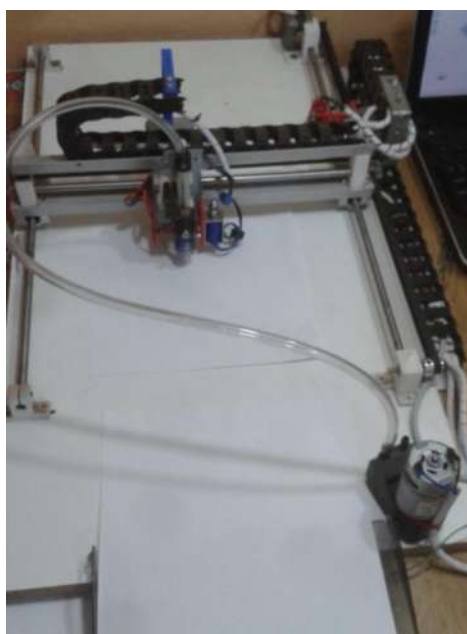


Figura 19-3: Impresión de imagen en relieve y texto braille.

Realizado por: (MUNGABUSI Angel, 2018)

3.3 Puesta en marcha del prototipo general

La prueba se realizó ya con el prototipo implementado y funcional donde se realizó la calibración del punto inicial X0Y0, también prueba de comunicación serial y actuadores como resultado ningún fallo en la misma.

Adicionalmente se probó el algoritmo el cual parte desde el editor de gráficos vectoriales Inkscape, este tiene el objetivo de posicionar la hoja de papel en el área de trabajo de manera correcta, ubicarla y quitarla al termina la impresión de las imágenes en relieve y su respectivo texto.

También se ajustó de mejor manera la correa dentada GT2, esta tendía con correr en banda con el motor de pasos, esto provocaba que perdiera pasos y la imagen se distorsionara por completo y el prototipo perdiera su posicionamiento, con esto se solucionó este problema mecánico.

Se diseñó la guía de manguera en 3D y su posterior impresión, que ayude de soporte para el tubo que va desde el mini compresor de aire hasta la ventosa ya que este producía error al posicionar la hoja de papel en el área de trabajo una tendencia hacia la izquierda respecto al prototipo.

Con ello se cumple con los requerimientos propuestos anteriormente en la Figura 20-3 se muestra el prototipo.

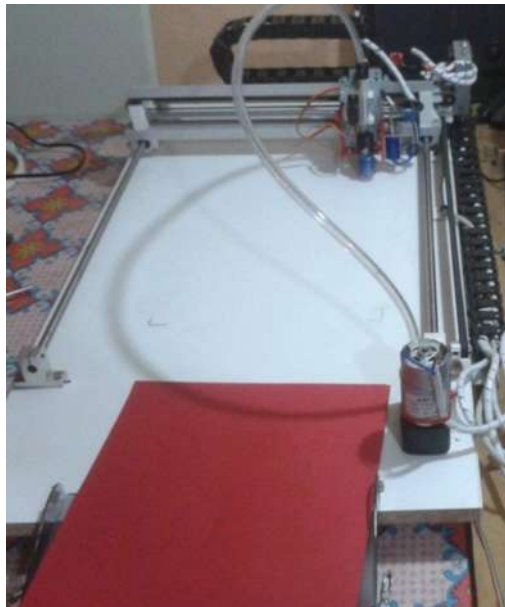


Figura 20-3: Puesta en marcha del prototipo.

Realizado por: (MUNGABUSI Angel, 2018)

Para la prueba de funcionamiento se procedió realizar las conexiones por los diferentes terminales DB9 hacia la parte móvil del prototipo que son los actuadores, continua con el proceso se realizó la conexión hacia la parte de comunicación por cable USB que comunica al Arduino con la PC, esto hace titilar un diodo led dentro del Arduino que corrobora el correcto funcionamiento,

también se verifico que ningún actuador este en modo encendido al momento de encender la fuente de alimentación.

3.4 Potencia de consumo del prototipo

Permite determinar la potencia de consumo total del prototipo y poder compararla con impresoras de tinta convencional con esto se pretende identificar donde se puede mejorar y hacer uso de menos energía para mover el mecanismo en futuras actualizaciones. En la Tabla 5-3 se muestra la potencia de consumo.

Tabla 5-3: Potencia de consumo total de energía eléctrica.

| Potencia de consumo del Prototipo | | | | |
|--|----------------------------------|--------------------|----------------------|---------------------|
| | Componentes | Voltaje (V) | Corriente (I) | Potencia (W) |
| PROTOTIPO | Arduino Mega 2560 | 5 | 0.04 | 0.2 |
| | Ramps 1.4 | 12 | 0.2 | 2.4 |
| | Motores a pasos | 12 | 1 | 12 |
| | Mini compresor de vacío | 12 | 1 | 12 |
| | Servomotor MG90S | 5 | 0.1 | 0.5 |
| | Solenoides | 30 | 0.3 | 9 |
| | Finales de carrera | 5 | 0.01 | 0.05 |
| | Placa de circuitos extra | 12 | 1.5 | 18 |
| | Sensor óptico | 5 | 0.3 | 1.5 |
| | POTENCIA DE CONSUMO TOTAL | | | |

Realizado por : (MUNGABUSI Angel, 2018)

El consumo de potencia del prototipo es de 55.65W durante la impresión, que es el 28% de la potencia total utilizada por una impresora convencional de tinta.

3.5 Análisis económico del prototipo

La Tabla 6-3 muestra los costos por cada elemento que constituye el prototipo, cabe mencionar que el mini compresor de aire se realizó una importación desde Estados Unidos que incremento el costo del prototipo.

Los costos de impresión 3D son variables dependen básicamente el lugar donde se las realice y el material para el prototipo se utilizó PLA, es por ello que se tiene estos valores mostrados a continuación.

Tabla 6-3: Costo total del prototipo.

| Prototipo | Componentes | Cantidad | Costo |
|------------------|--|-----------------|--------------|
| PROTOTIPO | Arduino Mega 2560 | 1 | \$20 |
| | Ramps 1.4 | 1 | \$15 |
| | Driver A4988 | 2 | \$10 |
| | Motores a Paso | 2 | \$30 |
| | Mini compresor de vacío | 1 | \$45 |
| | Servomotor MG90S | 2 | \$14 |
| | Solenoides | 1 | \$9 |
| | Finales de carrera | 2 | \$6 |
| | Fuente de 30V | 1 | \$15 |
| | Fuente MICROATX | 1 | \$20 |
| | Placa de circuitos extra | 1 | \$15 |
| | Impresión 3D | 7(Piezas) | \$50 |
| | Tornillos | 40 | \$6.50 |
| | Componentes Electromecánicos adicionales | | \$125 |
| | COSTO TOTAL DEL PROTOTIPO | | |

Realizado por: (MUNGABUSI Angel, 2018)

El costo total del prototipo es de 380.50 dólares americanos al comparar con una impresora comercial la cual tiene un costo elevado de miles de dólares americanos con el prototipo se ha reducido el costo en un 90%, esto hace asequible y poder contar con una impresora Braille personal.

Conclusiones

- Se implementó un robot cartesiano, para la elaboración de figuras en relieve enfocado a personas con discapacidad visual basado en la normativa INEN 2850, trabaja con papel estándar A4, con un tamaño de letra de 20 puntos.
- El prototipo implementado utiliza el 60% del tiempo real de impresión de figuras y escritura estimado por el Host, por lo que se puede concluir que el tiempo empleado en el proceso de impresión es óptimo.
- De las pruebas realizadas se determinó que la potencia consumida por el prototipo es de 55.65W durante en una tarea de impresión y es el 28% de la potencia total requerida por una impresora de tinta comercial lo que concluye que el equipo desarrollado tiene características de consumo similar con equipos comerciales de iguales propósitos.
- Al analizar los costos de implementación del prototipo implementado y compararlo con el de equipos comerciales de similares características se determinó que este representa apenas el 10% de inversión, lo que se puede concluir que el equipo implementado es de bajo costo, aun si duplicara el valor al incluir el rubro de mano de obra y costos para mejorar su estética
- De las pruebas realizadas para evaluar la estabilidad del prototipo implementado, aplicado a los valores relacionados con la normativa INEN 2850 para generar el símbolo braille y al emplear el coeficiente de correlación lineal de Pearson, se obtuvo valores menores al 10%, valor aceptado dentro de los rangos que delimitan la estabilidad de un sistema.
- De las pruebas de impresión realizadas para evaluar el cumplimiento de la normativa INEN 2850 se obtuvo un 100% de apego a los estándares establecidos, por lo que se puede concluir que el prototipo implementado cumple con los requerimientos establecidos al principio de la investigación.

Recomendaciones

- Incorporar módulos de impresión de papel continuo, aviso por altavoz e impresión por red.
- Realizar un estudio del prototipo para mejorar los tiempos de impresión.
- Mejorar el mecanismo de elaboración del relieve con ello bajar los niveles de ruido generados por el solenoide.
- Desarrollar un programa que permita el manejo de figuras y escritura braille con control directo del prototipo.

BIBLIOGRAFÍA

Acteck. Fuente de Poder Acteck Blazar de 500W para Gabinete Micro-ATX.[En línea]. [Consulta: 12 octubre2018]. Disponible en:

<https://pcel.com/Acteck-AF-B500-Slim-79195>

Adrew Waugh. What is a Limit Switch? [En línea].2018. [Consulta 9 de octubre de 2018]. Disponible en:

<https://library.automationdirect.com/what-is-a-limit-switch/>

Allegro. A4988.pdf.[En línea]. [Consulta: 20 de febrero de 2018]. Disponible en: https://www.pololu.com/file/download/A4988.pdf?file_id=0J450

Arduino - Introduction. (s/f).[En línea]. [Consulta:9 de octubre de 2018]. Disponible en: <https://www.arduino.cc/en/Guide/Introduction>

Arduino Mega 2560 R3.[En línea].2018. [Consulta 13 de junio de 2018].Disponible en: <http://arduino.cl/arduino-mega-2560/>

ASALE, R.-. (s/f). diccionario. [En línea]. [Consulta: 3 de febrero de 2018]. Disponible en: <http://dle.rae.es/?id=WYRlhzm>

Autodesk. Empezando con G-Code | Programación CNC | Autodesk. [En línea]. [Consulta: 12 de octubre de 2018]. Disponible en: <https://www.autodesk.com/industry/manufacturing/resources/manufacturing-engineer/g-code>

Barrietos, Peñin, Balaguer,Aracil, A., Luis, Carlos, Rafael. (2007). *Fundamentos_de_robotica.pdf* (Segunda).

Bart Meijer. Mechanical endstop PCB v1.1 | 3D printing experts. [En línea]. [Consulta: 12 de octubre de 2018]. Disponible en: https://reprapworld.com/products/electronics/endstops/mechanical_endstop_pcb_v1_1/

FESTO. FLAECHEPORTALE_ES. [En línea]. [Consulta: 30 de mayo de 2018]. Disponible en: https://www.festo.com/cat/es_es/data/doc_es/PDF/ES/FLAECHEPORTALE_ES.PDF

Floyd, T. L. *Dispositivos electrónicos*. 8. México: 2008. Pearson Educación. ISBN: 978-970-26-1193-6. p799-845.

García Antony. ¿Qué es y cómo funciona un servomotor? [En línea]. [2016]. [Consulta: 28 de mayo de 2018]. Disponible en:
<http://panamahitek.com/que-es-y-como-funciona-un-servomotor/>

Geekbot Electronics. TCRT5000 Sensor Óptico Reflectivo. [En línea]. [Consulta: 9 de octubre de 2018]. Disponible en:
<http://www.geekbotelectronics.com/producto/tcrt5000-sensor-optico-reflectivo/>

Gual,Serrano, Mañez. Obtención de gráficos tangibles mediante técnicas de Prototipado Rápido: el volumen como elemento compositivo de diseño — Web de la ONCE [GOBERNAMENTAL]. [En línea]. [Consulta: 31 de enero de 2018]. Disponible en:
http://www.once.es/new/servicios-especializados-en-discapacidad-visual/publicaciones-sobre-discapacidad-visual/nueva-estructura-revista-integracion/copy_of_numeros-publicados/numero-65/obtencion-de-graficos-tangibles-mediante-tecnicas-de-prototipado-rapido-el-volumen-como-elemento-compositivo-de-disenio/?searchterm=graficos%20tangibles

INDEX BRAILLE. Basic-D V5: light-weight braille embosser. [En línea]. [Consulta: 20 de febrero de 2018]. Disponible en:
<https://www.indexbraille.com/en-us/braille-embossers/basic-d-v5?c=EUR>

Inkscape. Acerca de | Inkscape. [En línea]. [Consulta: 16 de octubre de 2018]. Disponible en:
<https://inkscape.org/es/acerca-de/>

jbrazio. Introduction. [En línea]. [Consulta: 3 de febrero de 2018]. Disponible en:
<http://marlinfw.org/docs/basics/introduction.html>

Jhonny R. RAMPS 1.4/es - RepRap. [En línea]. [Consulta: 27 de junio de 2018]. Disponible en:
https://reprap.org/wiki/RAMPS_1.4/es

Kruger, B. Arduino CNC Shield – 100% GRBL Compatible. [En línea]. [2015]. [Consulta: 9 de octubre de 2018]. Disponible en:
<https://blog.protoneer.co.nz/arduino-cnc-shield/>

MarlinFirmware. Home. [En línea]. [Consulta: 18 de octubre de 2018]. Disponible en:

<http://marlinfw.org/>

Mungabusi, Angel, R. Tesis MUNGABUSI. [2018]

NaylampMechatronics. Polea dentada GT2-20T-5. [En línea]. [Consulta: 18 de octubre de 2018].

Disponible en:

<https://naylampmechatronics.com/impresion-3d-cnc/116-polea-dentada-gt2-20t-5.html>

NaylampMechatronics. Rodamiento Lineal con Soporte - SC8UU. [En línea]. [Consulta: 18 de octubre de 2018]. Disponible en:

<https://naylampmechatronics.com/impresion-3d-cnc/220-rodamiento-lineal-con-soporte-sc8uu.html>

NotepadPlusPlus. Bloc de notas ++ Inicio. [En línea]. [Consulta: 18 de octubre de 2018].

Disponible en:

<https://notepad-plus-plus.org/>

NSFCONTROLS. Los solenoides para cada aplicación incluyen bloqueo, enganche, liberación de retención. *Soluciones OEM incluyendo seguridad.* [En línea]. [Consulta: 12 de octubre de 2018]. Disponible en:

<http://www.nsfcontrols.co.uk/our-products/solenoids/>

NTE INEN. NTE INEN 2850 - Servicio Ecuatoriano de Normalización. [En línea]. [2014]. [Consulta: 13 de junio de 2018]. Disponible en:

<http://studylib.es/doc/8038414/n-te-inen-2850---servicio-ecuadoriano-de-normalizaci%C3%B3n>

Pololu - A4988. Stepper Motor Driver Carrier. [En línea]. [Consulta: 20 de febrero de 2018].

Disponible en:

<https://www.pololu.com/product/1182>

POLULO, R. & E. Pololu - Stepper Motor: Bipolar, 200 Steps/Rev, 28×32mm, 3.8V, 0.67 A/Phase. [En línea]. [Consulta: 20 de febrero de 2018]. Disponible en:

<https://www.pololu.com/product/1205>

PotentialLabs. JF-0630B DC24V 6N Force 10mm Stroke Linear Motion Solenoid Electromagnet. [En línea]. [Consulta: 30 de mayo de 2018]. Disponible en:

<https://potentiallabs.com/cart/buy-jf0630b-dc24v-6n-force-10mm-stroke-linear-motion-solenoid-electromagnet>

Python. Welcome to Python.org. [En línea]. [Consulta: 8 de octubre de 2018]. Disponible en: <https://www.python.org/>

Repetier. Home. [En línea]. [Consulta: 30 de enero de 2018]. Disponible en: <https://www.repetier.com/>

SOLIDWORKS. 3D CAD | SOLIDWORKS 2018. [En línea]. [Consulta: 16 de octubre de 2018]. Disponible en: <https://www.solidworks.com/es/category/3d-cad>

SparkFun. Vacuum Pump - 12V - ROB-10398 - SparkFun Electronics. [En línea]. [Consulta: 27 de junio de 2018]. Disponible en: <https://www.sparkfun.com/products/10398>

Stepper-Motor-Support-Document. [En línea]. [Consulta: 9 de octubre 2018]. Disponible en: <http://www.pbcllinear.com/Download/DataSheet/Stepper-Motor-Support-Document.pdf>

Uriel Mendez. Solenoides - Actuadores mecánicos. [En línea]. [2016]. [Consulta 9 de octubre de 2018]. Disponible en: <https://www.330ohms.com/blogs/blog/113042692-solenoides-actuadores-mecanicos>

Vi-Cas. Descripción del producto | Vi-Cas Manufacturing Company. [En línea]. [Consulta: 12 de octubre de 2018]. Disponible en: <https://vi-cas.com/>

VISION, E. & R. S. 1624R-Router.[pdf]. [En línea]. [Consulta: 20 de febrero de 2018]. Disponible en: <http://www.visionengravers.com/images/MachinePg/1624R-Router.pdf>

Walpole, R. E. *Probabilidad y estadística para ingeniería y ciencias.*[9]. México: Pearson 2012. ISBN: 978-607-32-1417-9. p816-819.

ANEXOS

Anexo A: Datasheet Arduino Mega



Product Overview

The Arduino Mega 2560 is a microcontroller board based on the ATmega2560 (datasheet). It has 54 digital input/output pins (of which 14 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. The Mega is compatible with most shields designed for the Arduino Duemilanove or Diecimila.

Index

| | |
|---|--------|
| Technical Specifications | Page 2 |
| How to use Arduino Programming Environment, Basic Tutorials | Page 6 |
| Terms & Conditions | Page 7 |
| Environmental Policies half sqm of green via Impatto Zero® | Page 7 |



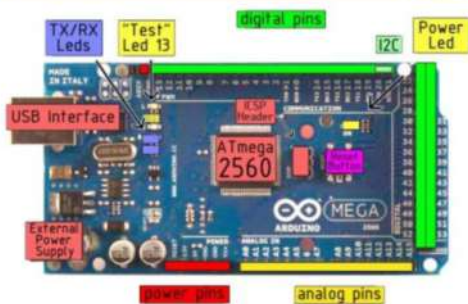
Technical Specification

EAGLE files: [arduino-mega2560-reference-design.sch](#), [Schematic: arduino-mega2560-schematic.pdf](#)

Summary

| | |
|-----------------------------|---|
| Microcontroller | ATmega2560 |
| Operating Voltage | 5V |
| Input Voltage (recommended) | 7-12V |
| Input Voltage (limits) | 6-20V |
| Digital I/O Pins | 54 (of which 14 provide PWM output) |
| Analog Input Pins | 16 |
| DC Current per I/O Pin | 40 mA |
| DC Current for 3.3V Pin | 50 mA |
| Flash Memory | 256 KB of which 8 KB used by bootloader |
| SRAM | 8 KB |
| EEPROM | 4 KB |
| Clock Speed | 16 MHz |

the board



Power

The Arduino Mega2560 can be powered via the USB connection or with an external power supply. The power source is selected automatically. External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector.

The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

The Mega2560 differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the ATmega8U2 programmed as a USB-to-serial converter.

The power pins are as follows:

- VIN:** The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
- 5V:** The regulated power supply used to power the microcontroller and other components on the board. This can come either from VIN via an on-board regulator, or be supplied by USB or another regulated 5V supply.
- 3V3:** A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
- GND:** Ground pins.

Memory

The ATmega2560 has 256 KB of flash memory for storing code (of which 8 KB is used for the bootloader), 8 KB of SRAM and 4 KB of EEPROM (which can be read and written with the [EEPROM library](#)).

Input and Output

Each of the 54 digital pins on the Mega can be used as an input or output, using [pinMode\(\)](#), [digitalWrite\(\)](#) and [digitalRead\(\)](#) functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:

- Serial:** 0 (RX) and 1 (TX); Serial 1: 19 (RX) and 18 (TX); Serial 2: 17 (RX) and 16 (TX); Serial 3: 15 (RX) and 14 (TX). Used to receive (RX) and transmit (TX) TTL serial data. Pins 0 and 1 are also connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip.
- External Interrupts:** 2 (interrupt 0), 3 (interrupt 1), 18 (interrupt 6), 19 (interrupt 4), 20 (interrupt 2), and 21 (interrupt 2). These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the [attachInterrupt\(\)](#) function for details.
- PWM:** 6 to 13. Provide 8-bit PWM output with the [analogWrite\(\)](#) function.
- SPI:** 8 (MISO), 51 (MOSI), 52 (SS), 53 (SS). These pins support SPI communication, which, although provided by the underlying hardware, is not currently included in the Arduino language. The SPI pins are also broken out on the ICSP header, which is physically compatible with the Duemilanove and Diecimila.
- LED:** 13. There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.
- IC:** 20 (SDA) and 21 (SCL). Support I²C (TWI) communication using the [Wire library](#) (documentation on the [Wiring website](#)). Note that these pins are not in the same location as the IC pins on the Duemilanove.

The Mega2560 has 16 analog inputs, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though it is possible to change the upper end of their range using the [AREF](#) pin and [analogReference\(\)](#) function.

There are a couple of other pins on the board:

- AREF:** Reference voltage for the analog inputs. Used with [analogReference\(\)](#).
- Reset:** Bring this low LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.



Communication

The Arduino Mega2560 has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega2560 provides four hardware UARTs for TTL (5V) serial communication. An ATmega8U2 on the board channels one of these over USB and provides a virtual com port to software on the computer (Windows machines will need a .inf file, but OSX and Linux machines will recognize the board as a COM port automatically). The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the board. The RX and TX LEDs on the board will flash when data is being transmitted via the ATmega8U2 chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

A [SoftwareSerial library](#) allows for serial communication on any of the Mega's digital pins.

The ATmega2560 also supports I²C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I²C bus; see the [documentation on the Wiring website](#) for details. To use the SPI communication, please see the ATmega2560 datasheet.

Programming

The Arduino Mega2560 can be programmed with the Arduino software ([download](#)). For details, see the [reference](#) and [tutorials](#).

The ATmega2560 on the Arduino Mega comes preburned with a [bootloader](#) that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol ([reference](#), [C header files](#)).

You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header; see [these instructions](#) for details.



Anexo B: Datasheet sensor óptico TCRT5000L

Reflective Optical Sensor with Transistor Output



FEATURES

- Package type: leaded
- Detector type: phototransistor
- Dimensions (L x W x H in mm): 10.2 x 5.8 x 7
- Peak operating distance: 2.5 mm
- Operating range within > 20 % relative collector current: 0.2 mm to 15 mm
- Typical output current under test: $I_C = 1$ mA
- Daylight blocking filter
- Emitter wavelength: 950 nm
- Lead (Pb)-free soldering released
- Compliant to RoHS directive 2002/95/EC and in accordance to WEEE 2002/96/EC



DESCRIPTION

The TCRT5000 and TCRT5000L are reflective sensors which include an infrared emitter and phototransistor in a leaded package which blocks visible light. The package includes two mounting clips. TCRT5000L is the long lead version.

APPLICATIONS

- Position sensor for shaft encoder
- Detection of reflective material such as paper, IBM cards, magnetic tapes etc.
- Limit switch for mechanical motions in VCR
- General purpose - wherever the space is limited

PRODUCT SUMMARY

| PART NUMBER | DISTANCE FOR MAXIMUM CTR _{REL} (1) (mm) | DISTANCE RANGE FOR RELATIVE I _C > 20 % (mm) | TYPICAL OUTPUT CURRENT UNDER TEST (1) (mA) | DAYLIGHT BLOCKING FILTER INTEGRATED |
|-------------|--|--|--|-------------------------------------|
| TCRT5000 | 2.5 | 0.2 to 15 | 1 | Yes |
| TCRT5000L | 2.5 | 0.2 to 15 | 1 | Yes |

Notes
(1) CTR: current transfer ratio, I_C/I_E
(2) Conditions list in table basic characteristics/sensors

ORDERING INFORMATION

| ORDERING CODE | PACKAGING | VOLUME (1) | REMARKS |
|---------------|-----------|---------------------------|--------------------|
| TCRT5000 | Tube | MOQ 4500 pcs, 50 pcs/tube | 3.5 mm lead length |
| TCRT5000L | Tube | MOQ 2400 pcs, 48 pcs/tube | 15 mm lead length |

Notes
(1) MOQ: minimum order quantity

ABSOLUTE MAXIMUM RATINGS (1)

| PARAMETER | TEST CONDITION | SYMBOL | VALUE | UNIT |
|---------------------------|--------------------------|-----------------------|-------------|------|
| INPUT (EMITTER) | | | | |
| Reverse voltage | | V _R | 5 | V |
| Forward current | | I _F | 60 | mA |
| Forward surge current | I _F ≤ 10 μs | I _{F(surge)} | 3 | A |
| Power dissipation | T _{amb} ≤ 25 °C | P _D | 100 | mW |
| Junction temperature | | T _J | 100 | °C |
| OUTPUT (DETECTOR) | | | | |
| Reverse voltage | | V _{RC} | 70 | V |
| Emitter collector voltage | | V _{EC} | 5 | V |
| Collector current | | I _C | 100 | mA |
| Power dissipation | T _{amb} ≤ 25 °C | P _D | 100 | mW |
| Junction temperature | | T _J | 100 | °C |
| Total power dissipation | T _{amb} ≤ 25 °C | P _T | 200 | mW |
| Ambient temperature range | | T _{amb} | -25 to +85 | °C |
| Storage temperature range | | T _{stg} | -25 to +150 | °C |
| Soldering temperature | 2 mm from case, t ≤ 10 s | T _{sold} | 260 | °C |

Notes



ABSOLUTE MAXIMUM RATINGS (1)

| PARAMETER | TEST CONDITION | SYMBOL | VALUE | UNIT |
|---------------------------|--------------------------|-------------------|-------------|------|
| OUTPUT (DETECTOR) | | | | |
| Collector emitter voltage | | V _{CE} | 70 | V |
| Emitter collector voltage | | V _{EC} | 5 | V |
| Collector current | | I _C | 100 | mA |
| Power dissipation | T _{amb} ≤ 25 °C | P _D | 100 | mW |
| Junction temperature | | T _J | 100 | °C |
| SENSOR | | | | |
| Total power dissipation | T _{amb} ≤ 25 °C | P _T | 200 | mW |
| Ambient temperature range | | T _{amb} | -25 to +85 | °C |
| Storage temperature range | | T _{stg} | -25 to +150 | °C |
| Soldering temperature | 2 mm from case, t ≤ 10 s | T _{sold} | 260 | °C |

Notes
(1) T_{amb} = 25 °C, unless otherwise specified

ABSOLUTE MAXIMUM RATINGS

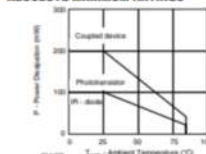


Fig. 1 - Power Dissipation Limit vs. Ambient Temperature

BASIC CHARACTERISTICS (1)

| PARAMETER | TEST CONDITION | SYMBOL | MIN. | TYP. | MAX. | UNIT |
|--------------------------------------|--|----------------------|------|------|------|-------|
| INPUT (EMITTER) | | | | | | |
| Forward voltage | I _F = 60 mA | V _F | 1.20 | 1.5 | 1.7 | V |
| Junction capacitance | V _R = 0 V, f = 1 MHz | C _J | 17 | | | pF |
| Radiant intensity | I _F = 60 mA, I _E = 20 mA | I _r | 21 | | | mW/sr |
| Peak wavelength | I _F = 100 mA | λ _p | 940 | | | nm |
| Virtual source diameter | Method: 63° enclosed angle | φ | 2.1 | | | mm |
| OUTPUT (DETECTOR) | | | | | | |
| Collector emitter voltage | I _C = 1 mA | V _{CE} | 70 | | | V |
| Emitter collector voltage | I _C = 100 μA | V _{EC} | 7 | | | V |
| Collector dark current | V _{CE} = 20 V, I _E = 0 A, E = 0 lx | I _{CE} | 10 | 200 | | nA |
| SENSOR | | | | | | |
| Collector current | V _{CE} = 5 V, I _E = 10 mA, D = 12 mm | I _C (0.1) | 0.5 | 1 | 2.1 | mA |
| Collector emitter saturation voltage | I _C = 10 mA, I _E = 0.1 mA, D = 12 mm | V _{CE(sat)} | 0.4 | | | V |

Notes
(1) T_{amb} = 25 °C, unless otherwise specified

(2) See Figure 3

(3) Test surface: mirror (Mn, SnPb or Au, Hoyer, Part No. 340000)

www.vishay.com

For technical questions, contact technical.support@vishay.com

Document Number: 88760
Rev. 1.7, 17-Aug-08

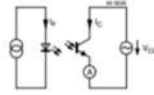


Fig. 2 - Test Circuit

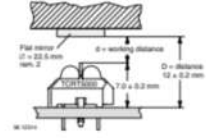


Fig. 3 - Test Circuit

BASIC CHARACTERISTICS

T_{amb} = 25 °C, unless otherwise specified

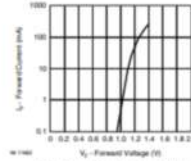


Fig. 4 - Forward Current vs. Forward Voltage

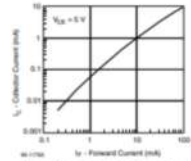


Fig. 6 - Collector Current vs. Forward Current

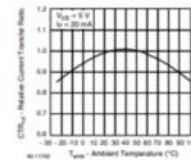


Fig. 5 - Relative Current Transfer Ratio vs. Ambient Temperature

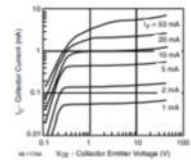


Fig. 7 - Collector Emitter Saturation Voltage vs. Collector Current

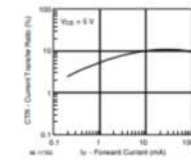


Fig. 8 - Current Transfer Ratio vs. Forward Current

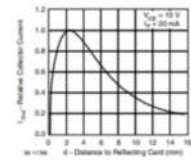
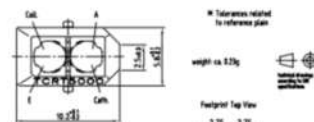
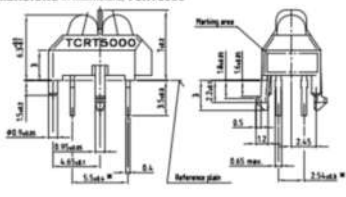


Fig. 9 - Relative Collector Current vs. Distance

PACKAGE DIMENSIONS in millimeters, TCRT5000



Drawing No. 4100-10N(01)-4
Issue 4, 10A(02)

www.vishay.com

Weight: ca. 120g

Footprint layout

Anexo C: Datasheet sensor de contacto

Mechanical Endstop v1.1



Author: Bart Meijer
 Date: 5th of December 2014
 Document version: 1.0



Mechanical Endstop Datasheet - Reprapworld.com 1

TECHNICAL SPECIFICATION

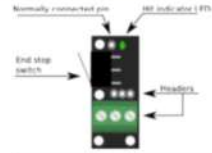
| | |
|------------------------------|-------|
| LED | Green |
| Power | 5V |
| Signal pin V (no hit) | 5V |
| Signal pin V (hit) | 0V |

MAJOR FEATURES

- Compatible with Megatronics v3.0, Minitrronics v1.1 and RAMPs
- Small design that integrates easily into your printer
- High quality 2-layer PCB
- Long durable end stop

Mechanical Endstop Datasheet - Reprapworld.com 4

CONNECTORS



| Name | Description |
|---------|---|
| Headers | Headers (3 pin and screw terminal) to connect the end stop to the RepRap electronics. 1. Signal 2. GND 3. 5V |
| NC | Normally connected pin (GND if endstop not hit, floating if hit) |

Mechanical Endstop Datasheet - Reprapworld.com 5

BOARD DIMENSIONS

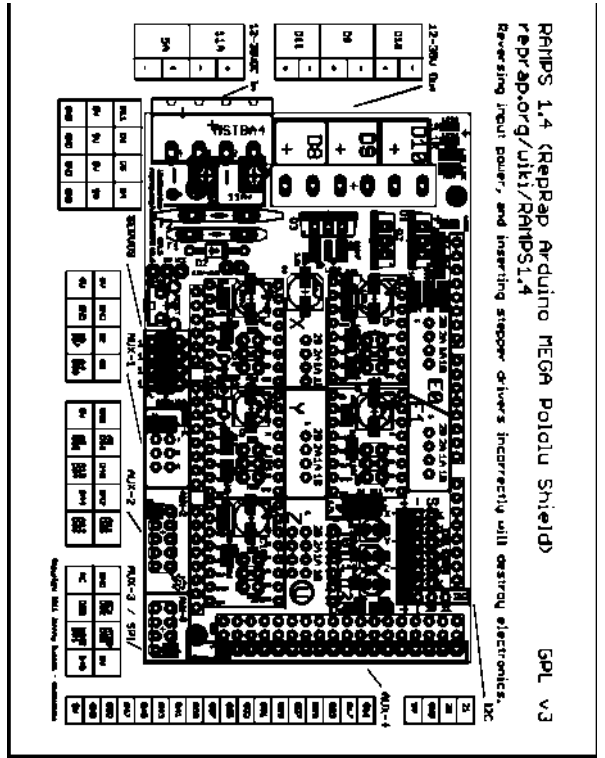
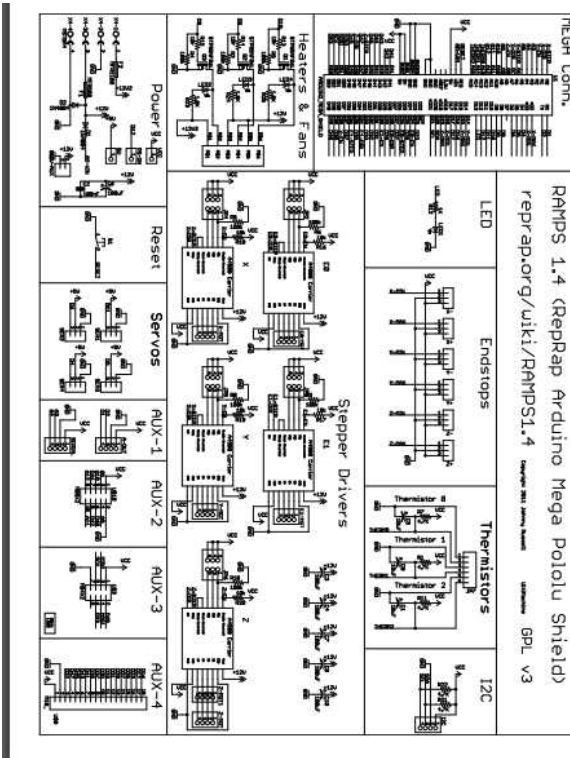
Dimensions: 16mmx40mmx15mm

List of M3 holes (measured from the bottom left):
 2.5, 2.5
 2.5, 17.5
 2.5, 37.5
 13.5, 2.5

Mechanical Endstop Datasheet - Reprapworld.com 6

Anexo D: Esquema de conexión de Ramps

1.4



Anexo E: Datasheet del circuito integrado LM555.

TEXAS INSTRUMENTS LM555
Product Folder | Sample & Buy | Technical Documents | Tools & Software | Support & Community
LM555 - FEBRUARY 2008 - REVISED JANUARY 2014

LM555 Timer

1 Features

- Direct Replacement for SE555NE/555
- Timing from Microseconds through Hours
- Operates in Both Astable and Monostable Modes
- Adjustable Duty Cycle
- Output Can Source or Sink 200 mA
- Output and Supply TTL Compatible
- Temperature Stability Better than 0.005% per °C
- Normally On and Normally Off Output
- Available in 8-pin VSSOP Package

2 Applications

- Precision Timing
- Pulse Generation
- Sequential Timing
- Time Delay Generation
- Pulse Width Modulation
- Pulse Position Modulation
- Linear Ramp Generator

3 Description

The LM555 is a highly stable device for generating accurate time delays or oscillation. Additional terminals are provided for triggering or resetting if desired. In the time delay mode of operation, the time is precisely controlled by one external resistor and capacitor. For a stable operation as an oscillator, the free running frequency and duty cycle are accurately controlled with two external resistors and one capacitor. The circuit may be triggered and reset on falling waveforms, and the output circuit can source or sink up to 200 mA or drive TTL circuits.

| Device Information(1) | | |
|-----------------------|-----------|-------------------|
| PART NUMBER | PACKAGE | BODY SIZE (MM) |
| LM555 | SOIC (8) | 4.90 mm × 3.81 mm |
| | PDP (8) | 3.81 mm × 4.25 mm |
| | VSSOP (8) | 5.00 mm × 3.00 mm |

(1) For all available packages, see the orderable address at the end of the datasheet.

Schematic Diagram

An IMPORTANT NOTICE at the end of this data sheet addresses availability, warranty, changes, use in safety-critical applications, intellectual property matters and other important disclosures. PRODUCTION DATA.

LM555 INSTRUMENTS
Product Folder | Sample & Buy | Technical Documents | Tools & Software | Support & Community
LM555 - FEBRUARY 2008 - REVISED JANUARY 2014

Table of Contents

| | | | |
|--------------------------------------|---|---|----|
| 1 Features | 1 | 7.3 Feature Description | 1 |
| 2 Applications | 1 | 7.4 Device Functional Modes | 1 |
| 3 Description | 1 | 8 Application and Implementation | 11 |
| 4 Revision History | 2 | 8.1 Application Information | 11 |
| 5 Pin Configuration and Functions | 3 | 8.2 Typical Application | 11 |
| 6 Specifications | 4 | 9 Power Supply Recommendations | 11 |
| 6.1 Absolute Maximum Ratings | 4 | 10 Layout | 11 |
| 6.2 ESD Ratings | 4 | 10.1 Layout Guidelines | 11 |
| 6.3 Recommended Operating Conditions | 4 | 10.2 Layout Example | 11 |
| 6.4 Thermal Information | 4 | 11 Device and Documentation Support | 11 |
| 6.5 Electrical Characteristics | 5 | 11.1 Trademarks | 11 |
| 6.6 Typical Characteristics | 6 | 11.2 Electrostatic Discharge Caution | 11 |
| 6.7 Detailed Description | 8 | 11.3 Glossary | 11 |
| 7 Detailed Description | 8 | 12 Mechanical, Packaging, and Orderable Information | 11 |
| 7.1 Overview | 8 | | |
| 7.2 Functional Block Diagram | 8 | | |

4 Revision History

| Changes from Revision C (March 2013) to Revision D | Page |
|--|------|
| • Added Pin Configuration and Functions section, ESD Ratings table, Feature Description section, Device Functional Modes, Application and Implementation section, Power Supply Recommendations section, Layout section, Device and Documentation Support section, and Mechanical, Packaging, and Orderable Information section | 1 |

| Changes from Revision B (March 2013) to Revision C | Page |
|--|------|
| • Changed layout of National Data Sheet to TI format | 1 |

2 Submit Documentation Feedback | Product Folder Link: LM555 | Copyright © 2008–2014, Texas Instruments Incorporated

TEXAS INSTRUMENTS LM555
Product Folder | Sample & Buy | Technical Documents | Tools & Software | Support & Community
LM555 - FEBRUARY 2008 - REVISED JANUARY 2014

5 Pin Configuration and Functions

8-Pin SOIC Package
 8-Pin PDP SOIC and VSSOP
 Top View

| Pin # | Symbol | IO | DESCRIPTION |
|-------|-----------------|----|---|
| 1 | CONTROL VOLTAGE | I | Controls the threshold and trigger levels. It determines the pulse width at the output. A minimum voltage applied to this pin can only be used to minimize the output duty cycle. |
| 2 | TRIGGER | I | Open collector output which discharges a capacitor between intervals (it pulses with output) if trigger has been high to low which resets the timer. 2/3 of the supply voltage. |
| 3 | OUTPUT | O | Driver reference voltage. |
| 4 | OUTPUT | O | Output source impedance. |
| 5 | RESET | I | Negative pulse applied to this pin will disable or reset the timer. When not used for reset, a minimum 0.1 μs pulse of 0.1 V to 0.5 V is recommended. |
| 6 | THRESHOLD | I | Controls the voltage applied to the internal timer. A reference voltage of 2/3 VCC. The amplitude of voltage applied to this terminal is measured in the test mode of the SOIC package. |
| 7 | TRIGGER | I | Resistor for threshold of the flip-flop and to reset. The output of the timer depends on the amplitude of the external trigger pulse applied to this pin. |
| 8 | VCC | I | Supply voltage with respect to GND. |

LM555 INSTRUMENTS
Product Folder | Sample & Buy | Technical Documents | Tools & Software | Support & Community
LM555 - FEBRUARY 2008 - REVISED JANUARY 2014

6 Specifications

6.1 Absolute Maximum Ratings

over operating free-air temperature range (unless otherwise noted)(1)(2)

| Parameter | LM555CM, LM555CN(1) | MIN | MAX | UNIT |
|---------------------------------------|---|------|-----|------|
| Power Dissipation(3) | LM555CM, LM555CN(1) | 1180 | mW | |
| | LM555CMM | 613 | mW | |
| Soldering Information | PDP Package | 260 | °C | |
| | Small Outline Packages (SOIC and VSSOP) | 215 | °C | |
| Storage temperature, T _{stg} | | -55 | 150 | °C |

(1) Stresses beyond those listed under Absolute Maximum Ratings may cause permanent damage to the device. These are stress ratings only, which do not imply functional operation of the device at these or any other conditions beyond those indicated under Recommended Operating Conditions. Exposure to absolute maximum rated conditions for extended periods may affect device reliability.
 (2) If Military/Aerospace specified devices are required, please contact the TI Sales Office/Distributors for availability and specifications.
 (3) For operating at elevated temperatures the device must be derated below 25°C based on a 100°C maximum junction temperature and a thermal resistance of 100°C/W (PDP), 170°C/W (SOIC-8), and 204°C/W (VSSOP) junction to ambient.

6.2 ESD Ratings

| Parameter | VALUE | UNIT |
|--|-------|------|
| V _{ESD} Electrostatic discharge (Human body model (HBM), per ANSI/ESDA/JEDEC JS-001(1)) | ±5000 | V |

(1) JEDEC document JEP155 states that 500-V HBM allows safe manufacturing with a standard ESD control process.
 (2) The ESD information listed is for the SOIC package.

6.3 Recommended Operating Conditions

over operating free-air temperature range (unless otherwise noted)

| Parameter | MIN | MAX | UNIT |
|--|-----|-----|------|
| Supply Voltage | 1.8 | 5 | V |
| Temperature, T _a | 0 | 70 | °C |
| Operating junction temperature, T _j | | 70 | °C |

6.4 Thermal Information

| THERMAL METRIC(1) | LM555 | | | UNIT |
|--|-------|------------|-------|------|
| | PDP | SOIC & PDP | VSSOP | |
| R _{θ(jc)} Junction-to-core thermal resistance | 100 | 170 | 204 | °C/W |

(1) For more information about traditional and new thermal metrics, see the IC Package Thermal Metrics application report, SPRA988.

4 Submit Documentation Feedback | Product Folder Link: LM555 | Copyright © 2008–2014, Texas Instruments Incorporated

Anexo F: Datasheet del MOSFET IRFZ44N

International Rectifier
IRFZ44N
 HEXFET® Power MOSFET

PD - 94953

- Advanced Process Technology
- Ultra Low On-Resistance
- Dynamic dv/dt Rating
- 175°C Operating Temperature
- Fast Switching
- Fully Avalanche Rated

Description
 Advanced HEXFET® Power MOSFETs from International Rectifier utilize advanced processing techniques to achieve extremely low on-resistance per silicon area. This benefit, combined with the fast switching speed and ruggedized device design that HEXFET power MOSFETs are well known for, provides that designers with an extremely efficient and reliable device for use in a wide variety of applications.

The TO-220 package is universally preferred for all commercial-industrial applications of power dissipation levels to approximately 60 watts. The low thermal resistance and low package cost of the TO-220 contribute to its wide acceptance throughout the industry.

Absolute Maximum Ratings

| Parameter | Max. | Units |
|--------------------------------------|----------------------|-------|
| I_D @ $T_J = 25^\circ\text{C}$ | 49 | A |
| I_{DM} @ $T_J = 100^\circ\text{C}$ | 160 | A |
| I_{SM} | 160 | A |
| I_{RM} @ $T_J = 25^\circ\text{C}$ | 30 | W |
| V_{GS} | 17.5 | V |
| V_{DS} | 100 | V |
| V_{GS} | 17.5 | V |
| V_{DS} | 100 | V |
| T_J | 175 | °C |
| T_{stg} | -55 to 175 | °C |
| T_{amb} | 100 (1.0W from Case) | °C |
| Storage Temperature Range | -55 to 175 | °C |
| Switching Surge, 0.1µs at 100ns | 10.0µs (1.0W) | |

Thermal Resistance

| Parameter | Typ. | Max. | Units |
|------------------|------|------|-------|
| $R_{\theta(jc)}$ | — | 1.8 | °C/W |
| $R_{\theta(jc)}$ | — | 3.0 | °C/W |
| $R_{\theta(jc)}$ | — | 62 | °C/W |

www.irf.com

International Rectifier
IRFZ44N

Fig 1. Typical Output Characteristics

Fig 2. Typical Output Characteristics

Fig 3. Typical Transfer Characteristics

Fig 4. Normalized On-Resistance vs. Temperature

www.irf.com

International Rectifier
IRFZ44N

Electrical Characteristics @ $T_J = 25^\circ\text{C}$ (unless otherwise specified)

| Parameter | Min. | Typ. | Max. | Units | Conditions |
|---------------|------|------|------|-------|--|
| $V_{DS(BO)}$ | — | — | 100 | V | $V_{GS} = 0\text{V}, I_D = 250\mu\text{A}$ |
| $V_{GS(th)}$ | — | 1.0 | 1.5 | V | Reference to 25°C, $I_D = 1\text{mA}$ |
| $V_{GS(on)}$ | — | 17.5 | 17.5 | V | $V_{DS} = 10\text{V}, I_D = 25\text{A}, R_{\theta(jc)} = 0$ |
| $V_{GS(off)}$ | — | 4.0 | — | V | $V_{DS} = V_{GS}, I_D = 250\mu\text{A}$ |
| $r_{DS(on)}$ | — | 25 | — | mΩ | $V_{GS} = 20\text{V}, I_D = 25\text{A}$ |
| $r_{DS(on)}$ | — | 280 | — | mΩ | $V_{GS} = 44\text{V}, V_{DS} = 0\text{V}, I_D = 100^\circ\text{C}$ |
| $r_{GS(on)}$ | — | 100 | — | mΩ | $V_{DS} = 20\text{V}$ |
| $r_{GS(off)}$ | — | 100 | — | mΩ | $V_{GS} = 20\text{V}$ |
| $t_{d(on)}$ | — | 83 | — | ns | $I_D = 25\text{A}$ |
| $t_{d(off)}$ | — | 14 | — | ns | $V_{GS} = 44\text{V}$ |
| $t_{d(10)}$ | — | 25 | — | ns | $V_{GS} = 10\text{V}$, See Fig. 5 and 13 |
| t_{tr} | — | 12 | — | ns | $V_{GS} = 20\text{V}$ |
| t_{sw} | — | 60 | — | ns | $R_{\theta(jc)} = 100$ |
| t_{off} | — | 84 | — | ns | $R_{\theta(jc)} = 100$ |
| t_{p} | — | 48 | — | ns | $V_{GS} = 10\text{V}$, See Fig. 10 & 11 |
| I_{D1} | — | 4.8 | — | A | Ballast load, $I_{D2} \leq 20\text{mA}$ |
| I_{D2} | — | 7.5 | — | mA | Ballast load, $I_{D1} \leq 20\text{mA}$ |
| C_{iss} | — | 1470 | — | pF | Gate to source and center of the contact |
| C_{oss} | — | 300 | — | pF | $V_{GS} = 20\text{V}$ |
| C_{rds} | — | 88 | — | pF | $V_{GS} = 20\text{V}$ |
| C_{iss} | — | 630 | — | pF | $f = 1.0\text{MHz}$, See Fig. 5 |
| E_{AS} | — | 630 | — | mJ | $I_{D1} = 25\text{A}, I_2 = 0.4\text{A}$ |

Source-Drain Ratings and Characteristics

| Parameter | Min. | Typ. | Max. | Units | Conditions |
|-------------|------|------|------|-------|---|
| I_D | — | — | 49 | A | MOSFET symbol showing the integral reverse bias junction diode |
| I_{SM} | — | — | 160 | A | Peak Forward Voltage |
| V_{GS} | — | — | 17.5 | V | $T_J = 25^\circ\text{C}, I_D = 25\text{A}, V_{DS} = 0\text{V}$ |
| t_{rr} | — | — | 63 | ns | $T_J = 25^\circ\text{C}, I_D = 25\text{A}$ |
| C_{iss} | — | — | 1470 | pF | Ballast load |
| $t_{d(10)}$ | — | — | 25 | ns | Reverse turn-on time to negligible turn-on is determined by I_{D1}/I_{D2} |

Notes:
 ① Repetitive rating; pulse width limited by rise junction temperature. (See Fig. 11)
 ② Rating: $T_J = 25^\circ\text{C}, t_r = 0.1\text{ms}$
 ③ $R_{\theta(jc)} = 25\text{K}, I_{D1} = 25\text{A}$, (See Figure 12)
 ④ $I_{D1} \leq 25\text{A}, I_{D2} \leq 250\text{mA}, V_{GS} \leq V_{GS(BO)}$
 ⑤ Pulse width $\leq 400\mu\text{s}$, duty cycle $\leq 2\%$
 ⑥ This is a typical value at device destruction and represents operation outside rated limits.
 ⑦ This is a calculated value limited to $T_J = 175^\circ\text{C}$

www.irf.com

International Rectifier
IRFZ44N

Fig 5. Typical Capacitance vs. Drain-to-Source Voltage

Fig 6. Typical Gate Charge vs. Gate-to-Source Voltage

Fig 7. Typical Source-Drain Diode Forward Voltage

Fig 8. Maximum Safe Operating Area

www.irf.com

Anexo G: Datasheet motor Nema17

PBC LINEAR
A PACIFIC BEARING CO.

Stepper Motor NEMA 17

This document describes mechanical and electrical specifications for PBC Linear stepper motors, including standard, hollow, and extended shaft variations.



Standard shaft motor shown.

| | |
|----------------------------|--------------------------------|
| Phases | 2 |
| Steps/Revolution | 200 |
| Step Accuracy | ±5% |
| SHUT Load | 30,000 Hours at 1000 RPM |
| Rated Torque | 25 N (5.6 lbs.) Push |
| Rated Torque | 65 N (15 lbs.) Pull |
| Radial IP Rating | 29 N (6.5 lbs.) At Flat Center |
| Approximate Operating Temp | 40 |
| Insulation Class | B, 100°C |
| Insulation Resistance | 100 Meg/Ohms |

| Description | Length | Mounting Bracket | Mounting Torque | Mounting Torque | Mounting Torque | Mounting Torque | Mounting Torque | Mounting Torque | Mounting Torque | Motor Weight |
|-------------|-----------------|------------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|--------------|
| Blank | 1.7" Min | None | Min. 10 N·m | 10 N·m | 10 N·m | 10 N·m | 10 N·m | 10 N·m | 10 N·m | 16 N |
| Single | 20.8mm (0.821") | 2 | 0.46 | 0.46 | 1.06 | 2.2 | 15 | 21 | 27 | 0.02 |
| Double | 40.3mm (1.601") | 2 | 0.65 | 0.65 | 1.2 | 2.5 | 25 | 35 | 30 | 0.06 |
| Triple | 60.8mm (2.411") | 2 | 0.85 | 0.85 | 1.6 | 3.0 | 30 | 42 | 35 | 0.11 |

*All standard motors have plug connector. Consult factory for other options.



Standard shaft dimensions shown. All other dimensions apply to hollow and extended shaft options.



Dimensions: mm (in)
4 Lead Connector: PBC Part#200480
(Consult factory for optional motor connectors)

LINEAR MOTION SOLUTIONS | www.pbclinear.com

Anexo H: Datos característicos de la micro compresor de vacío.

AIR PUMP SERIES **FOREVER**



D2028 Pump Specifications

Free Flow Range.....12-15LPM
 Vacuum Range.....0-16"Hg
 Pressure Range.....0-32PSI
 Standard Motor Voltage Options.....12VDC
 Power.....12W
 Motor Construction Option.....Iron Core-Oil Bearing
 Diaphragm Material Option.....EPDM
 Operating Temperature Range.....32°-120°F (0°-50°)

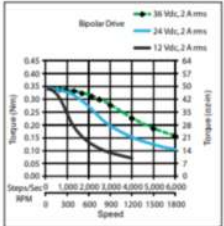


www.pbclinear.com

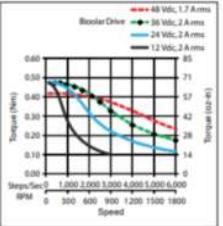
PBC LINEAR
A PACIFIC BEARING CO.

NEMA 17 Stepper Motor

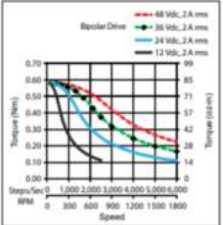
*Performance curves apply to continuous duty cycles. Consult factory for intermittent cycles or other voltages.



Single Stack

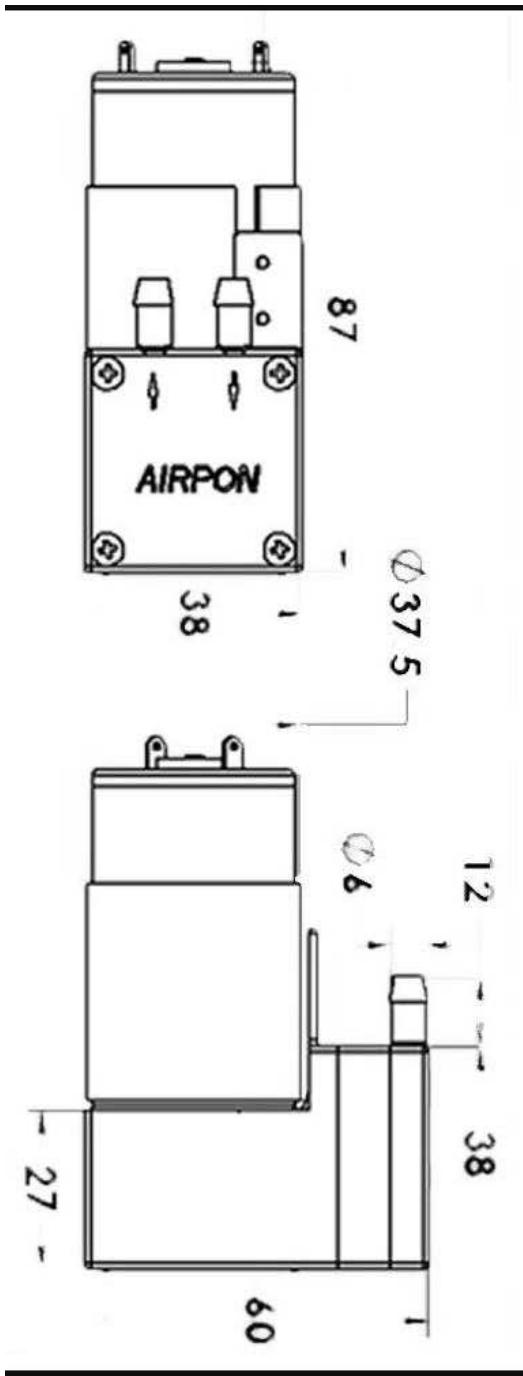


Double Stack



Triple Stack

www.pbclinear.com | LINEAR MOTION SOLUTIONS



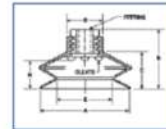
Anexo I: Ventosa VC-33C de la empresa VI-CAS.



Vi-Cas Manufacturing Company
 8407 Monroe Avenue Cincinnati, OH 45236
 Phone: 513-791-7741
 Fax: 513-791-6484
 Website: www.vi-cas.com

ITEM # VC-33A 1/8 NPT, ROUND VACUUM CUPS - STYLE J

Threaded type vacuum cups for uneven placement applications.




Specifications

| | |
|---------------------------------------|--------------------|
| Lip Diameter | 1.42 in 36.1 mm |
| Overall Height | 2.05 in 51.6 mm |
| Mounting Type | Threaded |
| Suggested Use | Uneven Surface |
| Bellows Type | Yes |
| Number of Convolutions (Bellows Type) | 5 |
| Thread Size | 1/8 NPT FEMALE |
| Cleats | NO |

6/20/11 | Page 17

| | |
|-------------------------------------|---|
| Style J - Dimension C | 1.32 in 33.5 mm |
| Style J - Dimension D | 0.68 in 17.3 mm |
| Style J - Dimension E | 0.71 in 18.0 mm |
| Style J - Dimension H | 1.28 in 32.5 mm |
| Select Material (Vinyl is standard) | Vinyl Urethane Silicone Oil Resistant Vinyl FDA Vinyl FDA Silicone |
| Select Thread Size | 1/8 NPT FEMALE |

Anexo J: Datasheet fuente de alimentación micro ATX.


Work Move Live Link Play Equip
www.acteck.com

Fuente de Poder
Fuente 500W, Micro ATX, 20+4 pines, 2 SATA
Power Supply 500W, Micro ATX, 20+4 pines, 2 SATA

Especificaciones Comerciales

- Más poder, menos espacio
- Para gabinetes compactos (Micro ATX)
- 500watts, 2 S-ATA (energía)
- Eficiente sistema de ventilación

Especificación principal

- Micro ATX
- Bajo nivel de ruido
- Sistema de ventilación
- Buena distribución del cableado
- Descripción de conectores: (1) 20+4 pines, (1) 12V 4 pines, (2) SATA (energía), (1) floppy, (2) molex.

Especificaciones de usabilidad


Commercial specifications

- More power, less space
- For slim PC cases (Micro ATX)
- 500w, 2 S-ATA (power)
- Efficient cooling system

Main feature

- Micro ATX
- Low noise level
- Ventilation System
- Efficient cable distribution
- Connector description: (1) 20+4 pins, (1) 12V 4 pins, (2) SATA (power), (1) floppy, (2) molex.

Usability specifications




Work Move Live Link Play Equip
www.acteck.com

Especificaciones:

| | |
|------------------------------|--|
| Código: | GAFU-029 |
| Modelo: | AF-B700 |
| Descripción: | Micro ATX 500w/24P/2 S-ATA |
| Nombre: | Fuente de Poder |
| Garantía: | 1 Año |
| Tamaño: | 6"12.5"10 cm |
| Peso Neto: | Peso Neto: 832 gr |
| Factor: | Micro ATX |
| Voltaje de entrada: | 100 VAC - 240 VAC |
| Corriente de Entrada: | 115 VAC/10A Max. 230 VAC/ 5A Max |
| Gama de Frecuencia: | 47-63 Hz |
| Conectores: | 1x 20+4Pines/1x12V 4 Pines/2x SATA/2x molex/1 Floppy |
| Rango de Operación: | 100 VAC - 240VAC |
| Consumo de Energía: | 500W +12V +5V +3.3V -12V +5VSB 24A 26A 0.5A 20A 2A |
| País de origen : | China |
| NOM: | Requiere |
| Contenido de empaque: | 1 Fuente de Poder 1 Cable de corriente 1 Manual de usuario |




Work Move Live Link Play Equip
www.acteck.com

Característica del producto

Una fuente con máximo poder en mínimo espacio. La fuente de poder Blazar fue especialmente creada para abastecer a gabinetes compactos, como el Bern, Berlin y Kendal de Acteck, con lo que podrás conformar tu equipo de cómputo ideal en menos espacio y con mayor estilo.

La fuente Blazar es compatible con tarjetas Micro ATX y tiene todos los conectores que necesitas para ensamblar una PC poderosa y completa.

Su sistema de ventilación mantiene una temperatura óptima para aumentar la estabilidad del equipo.

Features

Supply maximum power on minimal space. The Blazar power supply unit was specially created to work with slim PC cases, such as the Bern, Berlin and Kendal of Acteck; perfect to bring together your ideal computer on less space and with great style.

The Blazar power supply is compatible with Micro ATX motherboards and has every connector you need to assemble a powerful and complete PC.

Its cooling system maintains an optimal temperature to enhance the stability of the equipment.




Work Move Live Link Play Equip
www.acteck.com

Especificaciones:

| | |
|----------------------------|--|
| Code epicor: | GAFU-029 |
| Model: | AF-B700 |
| Description: | Micro ATX 500w/24P/2 S-ATA |
| Name: | Power Supply |
| Warranty: | 1 year |
| Dimensions: | 6"12.5"10 cm |
| Net Weight: | Net Weight: 832 gr |
| Factor: | Micro ATX |
| Input Voltage: | 100 VAC - 240 VAC |
| Input Current: | 115 VAC/10A Max. 230 VAC/ 5A Max |
| Frequency Range: | 47-63 Hz |
| Connectors: | 1x 20+4Pines/1x12V 4 Pines/2x SATA/2x molex/1 Floppy |
| Operating Range: | 100 VAC - 240VAC |
| Power consumption: | 500W +12V +5V +3.3V -12V +5VSB 24A 26A 0.5A 20A 2A |
| Country of origin : | China |
| NOM: | Required |
| Package contents: | 1 Power Supply 1 Cable 1 User manual |



Anexo K: Código Fuente del plugin Braille Mungabusi.

```
#!/usr/bin/env python
"""
```

Modified by ANGEL RODRIGO MUNGABUSI SISA
based on Jay Johnson 2015, J Tech Photonics, Inc.,
jtechphotonics.com

based on Adam Polak 2014, polakiumengineering.org

based on Copyright (C) 2009 Nick Drobchenko, nick@cnc-club.ru

based on gcode.py (C) 2007 hugomatic...

based on addnodes.py (C) 2005,2007 Aaron Spike,
aaron@ekips.org

based on dots.py (C) 2005 Aaron Spike, aaron@ekips.org

based on interp.py (C) 2005 Aaron Spike, aaron@ekips.org

based on bezmisc.py (C) 2005 Aaron Spike, aaron@ekips.org

based on cubicsuperpath.py (C) 2005 Aaron Spike,
aaron@ekips.org

This program is free software; you can redistribute it and/or
modify

it under the terms of the GNU General Public License as
published by

the Free Software Foundation; either version 2 of the License,
or

(at your option) any later version.

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the
implied warranty of

MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the

GNU General Public License for more details.

You should have received a copy of the GNU General Public
License

along with this program; if not, write to the Free Software
Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA
02111-1307 USA

```
"""
```

```
import inkex, simplestyle, simplepath
```

```
import cubicsuperpath, simpletransform, bezmisc
```

```
import os
```

```
import math
```

```
import bezmisc
```

```
import re
```

```
import copy
```

```
import sys
```

```
import time
```

```
import cmath
```

```
import numpy
```

```
import codecs
```

```
import random
```

```
import gettext
```

```
_ = gettext.gettext
```

```
### Check if inkex has errormsg (0.46 version doesnot have  
one.) Could be removed later.
```

```
if "errmsg" not in dir(inkex):
```

```
    inkex.errormsg = lambda msg:  
    sys.stderr.write((unicode(msg) + "\n").encode("UTF-8"))
```

```
def
```

```
bezierslopeatt(((bx0,by0),(bx1,by1),(bx2,by2),(bx3,by3)),t):
```

```
ax,ay,bx,by,cx,cy,x0,y0=bezmisc.bezierparameterize(((bx0,  
by0),(bx1,by1),(bx2,by2),(bx3,by3)))
```

```
dx=3*ax*(t**2)+2*bx*t+cx
```

```
dy=3*ay*(t**2)+2*by*t+cy
```

```
if dx==dy==0 :
```

```
    dx = 6*ax*t+2*bx
```

```
    dy = 6*ay*t+2*by
```

```
if dx==dy==0 :
```

```
    dx = 6*ax
```

```
    dy = 6*ay
```

```
if dx==dy==0 :
```

```
    print_("Slope error x = %s*t^3+%s*t^2+%s*t+%s,  
y = %s*t^3+%s*t^2+%s*t+%s, t = %s, dx==dy==0" %  
(ax,bx,cx,dx,ay,by,cy,dy,t))
```

```
    print_(((bx0,by0),(bx1,by1),(bx2,by2),(bx3,by3)))
```

```
    dx, dy = 1, 1
```

```
    return dx,dy
```

```
bezmisc.bezierslopeatt = bezierslopeatt
```

```
def ireplace(self,old,new,count=0):
```

```
    pattern = re.compile(re.escape(old),re.I)
```

```
    return re.sub(pattern,new,self,count)
```

```
#####
```

```
#####
```

```
###
```

```
### Styles and additional parameters
```

```
###
```

```
#####
#####

math.pi2 = math.pi*2
straight_tolerance = 0.0001
straight_distance_tolerance = 0.0001
engraving_tolerance = 0.0001
loft_lengths_tolerance = 0.0000001
options = {
defaults = {
'header': ""
G90
Adriansito
mungaBUSI
"",
'footer': ""G1 X0 Y0
G4 P100
""
}

intersection_recursion_depth = 10
intersection_tolerance = 0.00001

styles = {
  "loft_style" : {
    'main curve': simplestyle.formatStyle({ 'stroke':
'#88f', 'fill': 'none', 'stroke-width': '1', 'marker-
end': 'url(#Arrow2Mend)' }),
    },
  "biarc_style" : {
    'biarc0': simplestyle.formatStyle({ 'stroke': '#88f',
'fill': 'none', "marker-end": "url(#DrawCurveMarker)",
'stroke-width': '1' }),
    'biarc1': simplestyle.formatStyle({ 'stroke': '#8f8',
'fill': 'none', "marker-end": "url(#DrawCurveMarker)",
'stroke-width': '1' }),
    'line': simplestyle.formatStyle({ 'stroke': '#f88',
'fill': 'none', "marker-end": "url(#DrawCurveMarker)",
'stroke-width': '1' }),
    'area': simplestyle.formatStyle({ 'stroke': '#777',
'fill': 'none', "marker-end": "url(#DrawCurveMarker)",
'stroke-width': '0.1' }),
    },
  "biarc_style_dark" : {
    'biarc0': simplestyle.formatStyle({ 'stroke': '#33a',
'fill': 'none', "marker-end": "url(#DrawCurveMarker)",
'stroke-width': '1' }),
    'biarc1': simplestyle.formatStyle({ 'stroke': '#3a3',
'fill': 'none', "marker-end": "url(#DrawCurveMarker)",
'stroke-width': '1' }),

```

```

'line': simplestyle.formatStyle({ 'stroke': '#a33',
'fill': 'none', "marker-end": "url(#DrawCurveMarker)",
'stroke-width': '1' }),
    'area': simplestyle.formatStyle({ 'stroke': '#222',
'fill': 'none', "marker-end": "url(#DrawCurveMarker)",
'stroke-width': '0.3' }),
    },
  "biarc_style_dark_area" : {
    'biarc0': simplestyle.formatStyle({ 'stroke': '#33a',
'fill': 'none', "marker-end": "url(#DrawCurveMarker)",
'stroke-width': '0.1' }),
    'biarc1': simplestyle.formatStyle({ 'stroke': '#3a3',
'fill': 'none', "marker-end": "url(#DrawCurveMarker)",
'stroke-width': '0.1' }),
    'line': simplestyle.formatStyle({ 'stroke': '#a33',
'fill': 'none', "marker-end": "url(#DrawCurveMarker)",
'stroke-width': '0.1' }),
    'area': simplestyle.formatStyle({ 'stroke': '#222',
'fill': 'none', "marker-end": "url(#DrawCurveMarker)",
'stroke-width': '0.3' }),
    },
  "biarc_style_i" : {
    'biarc0': simplestyle.formatStyle({ 'stroke': '#880',
'fill': 'none', "marker-end": "url(#DrawCurveMarker)",
'stroke-width': '1' }),
    'biarc1': simplestyle.formatStyle({ 'stroke': '#808',
'fill': 'none', "marker-end": "url(#DrawCurveMarker)",
'stroke-width': '1' }),
    'line': simplestyle.formatStyle({ 'stroke': '#088',
'fill': 'none', "marker-end": "url(#DrawCurveMarker)",
'stroke-width': '1' }),
    'area': simplestyle.formatStyle({ 'stroke': '#999',
'fill': 'none', "marker-end": "url(#DrawCurveMarker)",
'stroke-width': '0.3' }),
    },
  "biarc_style_dark_i" : {
    'biarc0': simplestyle.formatStyle({ 'stroke': '#dd5',
'fill': 'none', "marker-end": "url(#DrawCurveMarker)",
'stroke-width': '1' }),
    'biarc1': simplestyle.formatStyle({ 'stroke': '#d5d',
'fill': 'none', "marker-end": "url(#DrawCurveMarker)",
'stroke-width': '1' }),
    'line': simplestyle.formatStyle({ 'stroke': '#5dd',
'fill': 'none', "marker-end": "url(#DrawCurveMarker)",
'stroke-width': '1' }),
    'area': simplestyle.formatStyle({ 'stroke': '#aaa',
'fill': 'none', "marker-end": "url(#DrawCurveMarker)",
'stroke-width': '0.3' }),
    },
  "biarc_style_lathe_feed" : {

```

```

        'biarc0': simplestyle.formatStyle({ 'stroke': '#07f',
'fill': 'none', "marker-end": "url(#DrawCurveMarker)",
'stroke-width': '.4' }),
        'biarc1': simplestyle.formatStyle({ 'stroke': '#0f7',
'fill': 'none', "marker-end": "url(#DrawCurveMarker)",
'stroke-width': '.4' }),
        'line': simplestyle.formatStyle({ 'stroke': '#f44',
'fill': 'none', "marker-end": "url(#DrawCurveMarker)",
'stroke-width': '.4' }),
        'area': simplestyle.formatStyle({ 'stroke': '#aaa',
'fill': 'none', "marker-end": "url(#DrawCurveMarker)",
'stroke-width': '0.3' }),
    },
    "biarc_style_lathe_passing feed" : {
        'biarc0': simplestyle.formatStyle({ 'stroke': '#07f',
'fill': 'none', "marker-end": "url(#DrawCurveMarker)",
'stroke-width': '.4' }),
        'biarc1': simplestyle.formatStyle({ 'stroke': '#0f7',
'fill': 'none', "marker-end": "url(#DrawCurveMarker)",
'stroke-width': '.4' }),
        'line': simplestyle.formatStyle({ 'stroke': '#f44',
'fill': 'none', "marker-end": "url(#DrawCurveMarker)",
'stroke-width': '.4' }),
        'area': simplestyle.formatStyle({ 'stroke': '#aaa',
'fill': 'none', "marker-end": "url(#DrawCurveMarker)",
'stroke-width': '0.3' }),
    },
    "biarc_style_lathe_fine feed" : {
        'biarc0': simplestyle.formatStyle({ 'stroke': '#7f0',
'fill': 'none', "marker-end": "url(#DrawCurveMarker)",
'stroke-width': '.4' }),
        'biarc1': simplestyle.formatStyle({ 'stroke': '#f70',
'fill': 'none', "marker-end": "url(#DrawCurveMarker)",
'stroke-width': '.4' }),
        'line': simplestyle.formatStyle({ 'stroke': '#744',
'fill': 'none', "marker-end": "url(#DrawCurveMarker)",
'stroke-width': '.4' }),
        'area': simplestyle.formatStyle({ 'stroke': '#aaa',
'fill': 'none', "marker-end": "url(#DrawCurveMarker)",
'stroke-width': '0.3' }),
    },
    "area artefact": simplestyle.formatStyle({ 'stroke':
'##ff0000', 'fill': '#ffff00', 'stroke-width': '1' }),
    "area artefact arrow": simplestyle.formatStyle({
'stroke': '#ff0000', 'fill': '#ffff00', 'stroke-width': '1' }),
    "dxf_points": simplestyle.formatStyle({ "stroke":
"##ff0000", "fill": "##ff0000" }),
}

```

```

#####
#####
###    Cubic Super Path additional functions
#####
#####

def csp_simple_bound(csp):
    minx,miny,maxx,maxy = None,None,None,None
    for subpath in csp:
        for sp in subpath :
            for p in sp:
                minx = min(minx,p[0]) if minx!=None else p[0]
                miny = min(miny,p[1]) if miny!=None else p[1]
                maxx = max(maxx,p[0]) if maxx!=None else p[0]
                maxy = max(maxy,p[1]) if maxy!=None else p[1]
    return minx,miny,maxx,maxy

def csp_segment_to_bez(sp1,sp2) :
    return sp1[1:]+sp2[2:]

def bound_to_bound_distance(sp1,sp2,sp3,sp4) :
    min_dist = 1e100
    max_dist = 0
    points1 = csp_segment_to_bez(sp1,sp2)
    points2 = csp_segment_to_bez(sp3,sp4)
    for i in range(4) :
        for j in range(4) :
            min_, max_ =
line_to_line_min_max_distance_2(points1[i-1], points1[i],
points2[j-1], points2[j])
            min_dist = min(min_dist,min_)
            max_dist = max(max_dist,max_)
            print("bound_to_bound", min_dist, max_dist)
    return min_dist, max_dist

def csp_to_point_distance(csp, p, dist_bounds = [0,1e100],
tolerance=.01) :
    min_dist = [1e100,0,0,0]
    for j in range(len(csp)) :
        for i in range(1,len(csp[j])) :
            d = csp_seg_to_point_distance(csp[j][i-
1],csp[j][i],p,sample_points = 5, tolerance = .01)
            if d[0] < dist_bounds[0] :
#                draw_pointer( list(csp_at_t(subpath[dist[2]-
1],subpath[dist[2]],dist[3]))
#
#                    +list(csp_at_t(csp[dist[4]][dist[5]-
1],csp[dist[4]][dist[5]],dist[6]),"red","line", comment =
math.sqrt(dist[0]))
                return [d[0],j,i,d[1]]

```

```

else :
    if d[0] < min_dist[0] : min_dist = [d[0],j,i,d[1]]
return min_dist

def csp_seg_to_point_distance(sp1,sp2,p,sample_points = 5,
tolerance = .01) :
    ax,ay,bx,by,cx,cy,dx,dy = csp_parameterize(sp1,sp2)
    dx, dy = dx-p[0], dy-p[1]
    if sample_points < 2 : sample_points = 2
    d = min( [(p[0]-sp1[1][0])**2 + (p[1]-sp1[1][1])**2,0.],
[(p[0]-sp2[1][0])**2 + (p[1]-sp2[1][1])**2,1.] )
    for k in range(sample_points) :
        t = float(k)/(sample_points-1)
        i = 0
        while i==0 or abs(f)>0.000001 and i<20 :
            t2,t3 = t**2,t**3
            f = (ax*t3+bx*t2+cx*t+dx)*(3*ax*t2+2*bx*t+cx) +
(ay*t3+by*t2+cy*t+dy)*(3*ay*t2+2*by*t+cy)
            df = (6*ax*t+2*bx)*(ax*t3+bx*t2+cx*t+dx) +
(3*ax*t2+2*bx*t+cx)**2 +
(6*ay*t+2*by)*(ay*t3+by*t2+cy*t+dy) +
(3*ay*t2+2*by*t+cy)**2
            if df!=0 :
                t = t - f/df
            else :
                break
            i += 1
        if 0<=t<=1 :
            p1 = csp_at_t(sp1,sp2,t)
            d1 = (p1[0]-p[0])**2 + (p1[1]-p[1])**2
            if d1 < d[0] :
                d = [d1,t]
    return d

def csp_seg_to_csp_seg_distance(sp1,sp2,sp3,sp4,
dist_bounds = [0,1e100], sample_points = 5, tolerance=.01) :
    # check the ending points first
    dist =
csp_seg_to_point_distance(sp1,sp2,sp3[1],sample_points,
tolerance)
    dist += [0.]
    if dist[0] <= dist_bounds[0] : return dist
    d =
csp_seg_to_point_distance(sp1,sp2,sp4[1],sample_points,
tolerance)
    if d[0]<dist[0] :
        dist = d+[1.]
    if dist[0] <= dist_bounds[0] : return dist

d =
csp_seg_to_point_distance(sp3,sp4,sp1[1],sample_points,
tolerance)
    if d[0]<dist[0] :
        dist = [d[0],0.,d[1]]
    if dist[0] <= dist_bounds[0] : return dist
    d =
csp_seg_to_point_distance(sp3,sp4,sp2[1],sample_points,
tolerance)
    if d[0]<dist[0] :
        dist = [d[0],1.,d[1]]
    if dist[0] <= dist_bounds[0] : return dist
    sample_points -= 2
    if sample_points < 1 : sample_points = 1
    ax1,ay1,bx1,by1,cx1,cy1,dx1,dy1 =
csp_parameterize(sp1,sp2)
    ax2,ay2,bx2,by2,cx2,cy2,dx2,dy2 =
csp_parameterize(sp3,sp4)
    # try to find closes points using Newtons method
    for k in range(sample_points) :
        for j in range(sample_points) :
            t1,t2 = float(k+1)/(sample_points+1),
float(j)/(sample_points+1)
            t12, t13, t22, t23 = t1*t1, t1*t1*t1, t2*t2, t2*t2*t2
            i = 0
            F1, F2, F = [0,0], [[0,0],[0,0]], 1e100
            x,y = ax1*t13+bx1*t12+cx1*t1+dx1 -
(ax2*t23+bx2*t22+cx2*t2+dx2),
ay1*t13+by1*t12+cy1*t1+dy1 -
(ay2*t23+by2*t22+cy2*t2+dy2)
            while i<2 or abs(F-Flast)>tolerance and i<30 :
                #draw_pointer(csp_at_t(sp1,sp2,t1))
                f1x = 3*ax1*t12+2*bx1*t1+cx1
                f1y = 3*ay1*t12+2*by1*t1+cy1
                f2x = 3*ax2*t22+2*bx2*t2+cx2
                f2y = 3*ay2*t22+2*by2*t2+cy2
                F1[0] = 2*f1x*x + 2*f1y*y
                F1[1] = -2*f2x*x - 2*f2y*y
                F2[0][0] = 2*(6*ax1*t1+2*bx1)*x + 2*f1x*f1x +
2*(6*ay1*t1+2*by1)*y +2*f1y*f1y
                F2[0][1] = -2*f1x*f2x - 2*f1y*f2y
                F2[1][0] = -2*f2x*f1x - 2*f2y*f1y
                F2[1][1] = -2*(6*ax2*t2+2*bx2)*x + 2*f2x*f2x -
2*(6*ay2*t2+2*by2)*y + 2*f2y*f2y
                F2 = inv_2x2(F2)
            if F2!=None :
                t1 -= ( F2[0][0]*F1[0] + F2[0][1]*F1[1] )
                t2 -= ( F2[1][0]*F1[0] + F2[1][1]*F1[1] )
                t12, t13, t22, t23 = t1*t1, t1*t1*t1, t2*t2,
t2*t2*t2

```

```

x,y      = ax1*t13+bx1*t12+cx1*t1+dx1 -
(ax2*t23+bx2*t22+cx2*t2+dx2),
ay1*t13+by1*t12+cy1*t1+dy1         -
(ay2*t23+by2*t22+cy2*t2+dy2)
    Flast = F
    F = x*x+y*y
else :
    break
    i += 1
if F < dist[0] and 0<=t1<=1 and 0<=t2<=1:
    dist = [F,t1,t2]
    if dist[0] <= dist_bounds[0] :
        return dist
return dist

def csp_to_csp_distance(csp1,csp2, dist_bounds = [0,1e100],
tolerance=.01) :
    dist = [1e100,0,0,0,0,0]
    for i1 in range(len(csp1)) :
        for j1 in range(1,len(csp1[i1])) :
            for i2 in range(len(csp2)) :
                for j2 in range(1,len(csp2[i2])) :
                    d =
csp_seg_bound_to_csp_seg_bound_max_min_distance(csp1
[i1][j1-1],csp1[i1][j1],csp2[i2][j2-1],csp2[i2][j2])
                    if d[0] >= dist_bounds[1] : continue
                    if d[1] < dist_bounds[0] : return
[d[1],i1,j1,1,i2,j2,1]
                    d = csp_seg_to_csp_seg_distance(csp1[i1][j1-
1],csp1[i1][j1],csp2[i2][j2-1],csp2[i2][j2], dist_bounds,
tolerance=tolerance)
                    if d[0] < dist[0] :
                        dist = [d[0], i1,j1,d[1], i2,j2,d[2]]
                    if dist[0] <= dist_bounds[0] :
                        return dist
                    if dist[0] >= dist_bounds[1] :
                        return dist
    return dist
# draw_pointer( list(csp_at_t(csp1[dist[1]][dist[2]-
1],csp1[dist[1]][dist[2]],dist[3]))
# + list(csp_at_t(csp2[dist[4]][dist[5]-
1],csp2[dist[4]][dist[5]],dist[6])), "#507","line")

def csp_split(sp1,sp2,t=.5) :
    [x1,y1],[x2,y2],[x3,y3],[x4,y4] = sp1[1], sp1[2], sp2[0],
sp2[1]
    x12 = x1+(x2-x1)*t
    y12 = y1+(y2-y1)*t
    x23 = x2+(x3-x2)*t
    y23 = y2+(y3-y2)*t
    x34 = x3+(x4-x3)*t
    y34 = y3+(y4-y3)*t
    x1223 = x12+(x23-x12)*t
    y1223 = y12+(y23-y12)*t
    x2334 = x23+(x34-x23)*t
    y2334 = y23+(y34-y23)*t
    x = x1223+(x2334-x1223)*t
    y = y1223+(y2334-y1223)*t
    return [sp1[0],sp1[1],[x12,y12]],
[[x1223,y1223],[x,y],[x2334,y2334]],
[[x34,y34],sp2[1],sp2[2]]

def csp_true_bounds(csp) :
    # Finds minx,miny,maxx,maxy of the csp and return their
(x,y,i,j,t)
    minx = [float("-inf"), 0, 0, 0]
    maxx = [float("-inf"), 0, 0, 0]
    miny = [float("-inf"), 0, 0, 0]
    maxy = [float("-inf"), 0, 0, 0]
    for i in range(len(csp)):
        for j in range(1,len(csp[i])):
            ax,ay,bx,by,cx,cy,x0,y0 =
bezmisc.bezierparameterize((csp[i][j-1][1],csp[i][j-
1][2],csp[i][j][0],csp[i][j][1]))
            roots = cubic_solver(0, 3*ax, 2*bx, cx) + [0,1]
            for root in roots :
                if type(root) is complex and abs(root.imag)<1e-10:
                    root = root.real
                if type(root) is not complex and 0<=root<=1:
                    y = ay*(root**3)+by*(root**2)+cy*root+y0
                    x = ax*(root**3)+bx*(root**2)+cx*root+x0
                    maxx = max([x,y,i,j,root],maxx)
                    minx = min([x,y,i,j,root],minx)

            roots = cubic_solver(0, 3*ay, 2*by, cy) + [0,1]
            for root in roots :
                if type(root) is complex and root.imag==0:
                    root = root.real
                if type(root) is not complex and 0<=root<=1:
                    y = ay*(root**3)+by*(root**2)+cy*root+y0
                    x = ax*(root**3)+bx*(root**2)+cx*root+x0
                    maxy = max([y,x,i,j,root],maxy)
                    miny = min([y,x,i,j,root],miny)
            maxy[0],maxy[1] = maxy[1],maxy[0]
            miny[0],miny[1] = miny[1],miny[0]
    return minx,miny,maxx,maxy

```

```
#####
#####

#####
#####

###
###   Generate Gcode
###   Generates Gcode on given curve.
###
###   Crve defenitmon [start point, type =
{'arc','line','move','end'}, arc center, arc angle, end point,
[zstart, zend]]
###
#####
#####

def generate_gcode(self, curve, layer, depth):
    tool = self.tools
    print_("Tool in g-code generator: " + str(tool))
    def c(c):
        c = [c[i] if i<len(c) else None for i in range(6)]
        if c[5] == 0 : c[5]=None
        s = [" X", " Y", " Z", " I", " J", " K"]
        r = ""
        for i in range(6):
            if c[i]!=None:
                r += s[i] + ("%f" % (round(c[i],4))).rstrip('0')
        return r

    def calculate_angle(a, current_a):
        return min(
            [abs(a-current_a%math.pi2+math.pi2),
a+current_a-current_a%math.pi2+math.pi2],
            [abs(a-current_a%math.pi2-math.pi2),
a+current_a-current_a%math.pi2-math.pi2],
            [abs(a-current_a%math.pi2),
a+current_a-current_a%math.pi2])[1]
        if len(curve)==0 : return ""

    try :
        self.last_used_tool == None
    except :
        self.last_used_tool = None
    print_("working on curve")
    print_("Curve: " + str(curve))
    g = ""

    lg, f = 'G00', "F%f"%tool['penetration feed']
    penetration_feed = "F%s"%tool['penetration feed']
    current_a = 0
    for i in range(1,len(curve)):
```

```
#   Creating Gcode for curve between s=curve[i-1] and
si=curve[i] start at s[0] end at s[4]=si[0]
    s, si = curve[i-1], curve[i]
    feed = f if lg not in ['G01','G02','G03'] else ""
    if s[1] == 'move':
        g += "G1 " + c(si[0]) + "\n" + tool['gcode before
path'] + "\n"
        lg = 'G00'
    elif s[1] == 'end':
        g += tool['gcode after path'] + "\n"
        lg = 'G00'
    elif s[1] == 'line':
        if lg=="G00": g += "G1 " + feed + "\n"
        g += "G1 " + c(si[0]) + "\n"
        lg = 'G01'
    elif s[1] == 'arc':
        r = [(s[2][0]-s[0][0]), (s[2][1]-s[0][1])]
        if lg=="G00": g += "G1 " + feed + "\n"
        if (r[0]**2 + r[1]**2)>.1:
            r1, r2 = (P(s[0])-P(s[2])), (P(si[0])-P(s[2]))
            if abs(r1.mag()-r2.mag()) < 0.001 :
                g += ("G2" if s[3]<0 else "G3") + c(si[0]+[
None, (s[2][0]-s[0][0]),(s[2][1]-s[0][1]) ]) + "\n"
            else:
                r = (r1.mag()+r2.mag())/2
                g += ("G2" if s[3]<0 else "G3") + c(si[0]) + "
R%f" % (r) + "\n"
                lg = 'G02'
            else:
                g += "G1 " + c(si[0]) + " " + feed + "\n"
                lg = 'G01'
    if si[1] == 'end':
        g += tool['gcode after path'] + "\n"
    return g

def get_transforms(self,g):
    root = self.document.getroot()
    trans = []
    while (g!=root):
        if 'transform' in g.keys():
            t = g.get('transform')
            t = simpletransform.parseTransform(t)
            trans =
simpletransform.composeTransform(t,trans) if trans != [] else
t

        print_(trans)
        g=g.getparent()
    return trans
```



```

print_(self.transform_matrix_reverse)

###self.Zauto_scale[layer] = math.sqrt(
(self.transform_matrix[layer][0][0]**2 +
self.transform_matrix[layer][1][1]**2)/2 )
### Zautoscale is obsolete
self.Zauto_scale[layer] = 1
print_("Z automatic scale = %s (computed according
orientation points)" % self.Zauto_scale[layer])

x,y = source_point[0], source_point[1]
if not reverse :
    t = self.transform_matrix[layer]
else :
    t = self.transform_matrix_reverse[layer]
return [t[0][0]*x+t[0][1]*y+t[0][2],
t[1][0]*x+t[1][1]*y+t[1][2]]

def transform_csp(self, csp_, layer, reverse = False):
    csp = [ [csp_[i][j][0][:],csp_[i][j][1][:],csp_[i][j][2][:]]
for j in range(len(csp_[i])) ] for i in range(len(csp_)) ]
    for i in xrange(len(csp)):
        for j in xrange(len(csp[i])):
            for k in xrange(len(csp[i][j])):
                csp[i][j][k] = self.transform(csp[i][j][k],layer,
reverse)
    return csp

#####
#####
### Errors handling function, notes are just printed into
Logfile,
### warnings are printed into log file and warning
message is displayed but
### extension continues working, errors causes log and
execution is halted
### Notes, warnings adn errors could be assigned to space
or comma or dot
### seeparated strings (case is ignore).
#####
#####
def error(self, s, type_= "Warning"):
    notes = "Note "
    warnings = ""
        Warning tools_warning
        bad_orientation_points_in_some_layers
        more_than_one_orientation_point_groups
        more_than_one_tool
        orientation_have_not_been_defined

```

```

tool_have_not_been_defined
selection_does_not_contain_paths

selection_does_not_contain_paths_will_take_all
selection_is_empty_will_comupe_drawing

selection_contains_objects_that_are_not_paths
"""
errors = ""
    Error
    wrong_orientation_points
    area_tools_diameter_error
    no_tool_error
    active_layer_already_has_tool
    active_layer_already_has_orientation_points
"""
if type_.lower() in re.split("[\s\n,\.]+", errors.lower()) :
    print_(s)
    inkex.errormsg(s+"\n")
    sys.exit()
elif type_.lower() in re.split("[\s\n,\.]+",
warnings.lower()) :
    print_(s)
    if not self.options.suppress_all_messages :
        inkex.errormsg(s+"\n")
elif type_.lower() in re.split("[\s\n,\.]+", notes.lower()) :
    print_(s)
else :
    print_(s)
    inkex.errormsg(s)
    sys.exit()

#####
#####
### Get defs from svg
#####
#####
def get_defs(self) :
    self.defs = { }
def recursive(g) :
    for i in g:
        if i.tag == inkex.addNS("defs", "svg") :
            for j in i:
                self.defs[j.get("id")] = i
        if i.tag ==inkex.addNS("g", 'svg') :
            recursive(i)
recursive(self.document.getroot())

```



```
#####
#####
###
###   Get Gcodetools info from the svg
###
#####
#####

def get_info(self):
    self.selected_paths = {}
    self.paths = {}
    self.orientation_points = {}
    self.layers = [self.document.getroot()]
    self.Zcoordinates = {}
    self.transform_matrix = {}
    self.transform_matrix_reverse = {}
    self.Zauto_scale = {}

def recursive_search(g, layer, selected=False):
    items = g.getchildren()
    items.reverse()
    for i in items:
        if selected:
            self.selected[i.get("id")] = i
            if i.tag == inkex.addNS("g", 'svg') and
i.get(inkex.addNS('groupmode', 'inkscape')) == 'layer':
                self.layers += [i]
                recursive_search(i, i)
            elif i.get('gcodetools') == "Gcodetools orientation
group" :
                points = self.get_orientation_points(i)
                if points != None :
                    self.orientation_points[layer] =
self.orientation_points[layer]+[points[:]] if layer in
self.orientation_points else [points[:]]
                    print_("Found orientation points in '%s' layer:
%s" % (layer.get(inkex.addNS('label', 'inkscape')), points))
                    else :
                        self.error(_("Warning! Found bad orientation
points in '%s' layer. Resulting Gcode could be corrupt!") %
layer.get(inkex.addNS('label', 'inkscape')),
"bad_orientation_points_in_some_layers")
                elif i.tag == inkex.addNS('path', 'svg'):
                    if "gcodetools" not in i.keys() :
                        self.paths[layer] = self.paths[layer] + [i] if
layer in self.paths else [i]
                    if i.get("id") in self.selected :
                        self.selected_paths[layer] =
self.selected_paths[layer] + [i] if layer in self.selected_paths
else [i]
                    elif i.tag == inkex.addNS("g", 'svg'):
```

```
recursive_search(i, layer, (i.get("id") in
self.selected) )
elif i.get("id") in self.selected :
    self.error(_("This extension works with Paths
and Dynamic Offsets and groups of them only! All other
objects will be ignored!\nSolution 1: press Path->Object to
path or Shift+Ctrl+C.\nSolution 2: Path->Dynamic offset or
Ctrl+J.\nSolution 3: export all contours to PostScript level 2
(File->Save As->.ps) and File->Import this
file."), "selection_contains_objects_that_are_not_paths")

recursive_search(self.document.getroot(), self.document.getr
oot())

def get_orientation_points(self, g):
    items = g.getchildren()
    items.reverse()
    p2, p3 = [], []
    p = None
    for i in items:
        if i.tag == inkex.addNS("g", 'svg') and
i.get("gcodetools") == "Gcodetools orientation point (2
points)":
            p2 += [i]
            if i.tag == inkex.addNS("g", 'svg') and
i.get("gcodetools") == "Gcodetools orientation point (3
points)":
                p3 += [i]
            if len(p2)==2 : p=p2
            elif len(p3)==3 : p=p3
            if p==None : return None
            points = []
            for i in p :
                point = [[], []]
                for node in i :
                    if node.get('gcodetools') == "Gcodetools
orientation point arrow":
                        point[0] =
self.apply_transforms(node, cubicsuperpath.parsePath(node.
get("d")))[0][0][1]
                    if node.get('gcodetools') == "Gcodetools
orientation point text":
                        r = re.match(r'(?)\s*(\s*(-
?\s*\d*(?:,|\.)*\d*)\s*;\s*(-?\s*\d*(?:,|\.)*\d*)\s*;\s*(-
?\s*\d*(?:,|\.)*\d*)\s*)\s*', node.text)
                        point[1] =
[float(r.group(1)), float(r.group(2)), float(r.group(3))]
                    if point[0]!=[] and point[1]!=[]: points += [point]
```

```

        if len(points)==len(p2)==2 or len(points)==len(p3)==3
: return points
        else : return None

#####
#####
###
###   dxfpoints
###
#####
#####

def dxfpoints(self):
    if self.selected_paths == {}:
        self.error(_("Noting is selected. Please select
something to convert to drill point (dxfpoint) or clear point
sign."), "warning")
    for layer in self.layers :
        if layer in self.selected_paths :
            for path in self.selected_paths[layer]:
                if self.options.dxfpoints_action == 'replace':
                    path.set("dxfpoint", "1")
                    r = re.match("^\s*\.\s*(\S+)", path.get("d"))
                    if r!=None:
                        print_(_("got path=", r.group(1)))
                        path.set("d", "m      %s      2.9375,-
6.343750000001      0.8125,1.90625      6.843748640396,-
6.84374864039      0,0      0.6875,0.6875      -6.84375,6.84375
1.90625,0.812500000001 z" % r.group(1))
                        path.set("style", styles["dxf_points"])

                if self.options.dxfpoints_action == 'save':
                    path.set("dxfpoint", "1")

                if self.options.dxfpoints_action == 'clear' and
path.get("dxfpoint") == "1":
                    path.set("dxfpoint", "0")

#####
#####
###
###   Laser
###
#####
#####

def laser(self) :

def get_boundaries(points):
    minx,miny,maxx,maxy=None,None,None,None
    out=[[[],[],[],[]]
    for p in points:
        if minx==p[0]:

```

```

        out[0]+=p]
        if minx==None or p[0]<minx:
            minx=p[0]
            out[0]=[p]

        if miny==p[1]:
            out[1]+=p]
        if miny==None or p[1]<miny:
            miny=p[1]
            out[1]=[p]

        if maxx==p[0]:
            out[2]+=p]
        if maxx==None or p[0]>maxx:
            maxx=p[0]
            out[2]=[p]

        if maxy==p[1]:
            out[3]+=p]
        if maxy==None or p[1]>maxy:
            maxy=p[1]
            out[3]=[p]
    return out

def remove_duplicates(points):
    i=0
    out=[]
    for p in points:
        for j in xrange(i,len(points)):
            if p==points[j]: points[j]=[None,None]
            if p!=points[j]: out+=p]
    i+=1
    return(out)

def get_way_len(points):
    l=0
    for i in xrange(1,len(points)):
        l+=math.sqrt((points[i][0]-points[i-1][0])**2 +
(points[i][1]-points[i-1][1])**2)
    return l

def sort_dxfpoints(points):
    points=remove_duplicates(points)

    ways=[
        # l=0, d=1, r=2, u=3
        [3,0], # ul
        [3,2], # ur

```

```

[1,0], # dl
[1,2], # dr
[0,3], # lu
[0,1], # ld
[2,3], # ru
[2,1], # rd
]

minimal_way=[]
minimal_len=None
minimal_way_type=None
for w in ways:
    tpoints=points[:]
    cw=[]
    for j in xrange(0,len(points)):

p=get_boundaries(get_boundaries(tpoints)[w[0]])[w[1]]
    tpoints.remove(p[0])
    cw+=p
    curlen = get_way_len(cw)
    if minimal_len==None or curlen < minimal_len:
        minimal_len=curlen
        minimal_way=cw
        minimal_way_type=w

return minimal_way

if self.selected_paths == {} :
    paths=self.paths
    self.error(_("No paths are selected! Trying to work on
all available paths."),"warning")
else :
    paths = self.selected_paths

self.check_dir()
gcode = ""

biarc_group = inkex.etree.SubElement(
self.selected_paths.keys()[0] if
len(self.selected_paths.keys())>0 else self.layers[0],
inkex.addNS('g','svg') )

print_("self.layers=",self.layers)
print_("paths=",paths)
for layer in self.layers :
    if layer in paths :
        print_(("layer",layer))
        p = []
        dxfpnts = []
        for path in paths[layer] :
            print_(str(layer))
            if "d" not in path.keys() :
                self.error(_("Warning: One or more paths dont
have 'd' parameter, try to Ungroup (Ctrl+Shift+G) and Object
to Path
(Ctrl+Shift+C)!"),"selection_contains_objects_that_are_not
_paths")
                continue
            csp = cubicsuperpath.parsePath(path.get("d"))
            csp = self.apply_transforms(path, csp)
            if path.get("dxfpnt") == "1":
                tmp_curve=self.transform_csp(csp, layer)
                x=tmp_curve[0][0][0][0]
                y=tmp_curve[0][0][0][1]
                print_(_("got dxfpnt (scaled) at (%f,%f)" %
(x,y))
                dxfpnts += [[x,y]]
            else:
                p += csp
        dxfpnts=sort_dxfpnts(dxfpnts)
        curve = self.parse_curve(p, layer)
        self.draw_curve(curve, layer, biarc_group)
        gcode += self.generate_gcode(curve, layer, 0)

self.export_gcode(gcode)

#####
#####
###
### Orientation
###
#####
#####

def orientation(self, layer=None) :
    print_("entering orientations")
    if layer == None :
        layer = self.current_layer if self.current_layer is not
None else self.document.getroot()
    if layer in self.orientation_points:
        self.error(_("Active layer already has orientation
points! Remove them or select another
layer!"),"active_layer_already_has_orientation_points")

    orientation_group = inkex.etree.SubElement(layer,
inkex.addNS('g','svg'), {"gcodetools":"Gcodetools
orientation group"})

    # translate == ['0', '-917.7043']
    if layer.get("transform") != None :
        translate =
layer.get("transform").replace("translate(", "").replace(",
", "").split(",")
    else :

```

```

        translate = [0,0]

        # doc height in pixels (38 mm == 134.64566px)
        doc_height = self.unittouu(self.document.getroot().xpath('@height',
namespaces=inkex.NSS)[0])

        if self.document.getroot().get('height') == "100%" :
            doc_height = 1052.3622047
            print_("Overriding height from 100 percents to %s"
% doc_height)

        print_("Document height: " + str(doc_height));

        if self.options.unit == "G21 (All units in mm)" :
            points = [[0.,0.,0.],[100.,0.,0.],[0.,100.,0.]]
            orientation_scale = 3.5433070660
            print_("orientation_scale < 0 ==> switching to mm
units=%0.10f"%orientation_scale )
        elif self.options.unit == "G20 (All units in inches)" :
            points = [[0.,0.,0.],[5.,0.,0.],[0.,5.,0.]]
            orientation_scale = 90
            print_("orientation_scale < 0 ==> switching to
inches units=%0.10f"%orientation_scale )

        points = points[:2]

        print_("(using orientation
scale",orientation_scale,"i=",points))
        for i in points :
            # X == Correct!
            # si == x,y coordinate in px
            # si have correct coordinates
            # if layer have any tranform it will be in translate so
lets add that
            si = [i[0]*orientation_scale,
(i[1]*orientation_scale)+float(translate[1])]
            g = inkex.etree.SubElement(orientation_group,
inkex.addNS('g','svg'), {'gcodetools': "Gcodetools orientation
point (2 points)"})
            inkex.etree.SubElement(g,
inkex.addNS('path','svg'),
            {
                'style': "stroke:none;fill:#000000;",
                'd': 'm %s,%s 2.9375,-6.343750000001
0.8125,1.90625 6.843748640396,-6.84374864039 0,0
0.6875,0.6875 -6.84375,6.84375 1.90625,0.812500000001 z'
% (si[0], -si[1]+doc_height),
                'gcodetools': "Gcodetools orientation point
arrow"
            })

```

```

        t = inkex.etree.SubElement(g,
inkex.addNS('text','svg'),
        {
            'style': "font-size:10px;font-style:normal;font-
variant:normal;font-weight:normal;font-
stretch:normal;fill:#000000;fill-opacity:1;stroke:none;",
            inkex.addNS("space","xml"):"preserve",
            'x': str(si[0]+10),
            'y': str(-si[1]-10+doc_height),
            'gcodetools': "Gcodetools orientation point text"
        })
        t.text = "(%s; %s; %s)" % (i[0],i[1],i[2])

#####
#####
###
###   Effect modifiko de "gcode before path": ("G4 P0 \n"
+ self.options.laser_command + " S" +
str(int(self.options.laser_power)) + "\nG4 P" +
self.options.power_delay),
###   modifiko linea a lo siguiete "gcode before path": ("G4
P0 \n" + self.options.laser_command + "\nG4 P" +
self.options.power_delay),
###   Main function of Gcodetools class
###
#####
#####
def effect(self) :
    global options
    options = self.options
    options.self = self
    options.doc_root = self.document.getroot()
    # define print_ function
    global print_
    if self.options.log_create_log :
        try :
            if os.path.isfile(self.options.log_filename) :
os.remove(self.options.log_filename)
            f = open(self.options.log_filename,"a")
            f.write("Gcodetools log file.\nStarted at
%s.\n%s\n" % (time.strftime("%d.%m.%Y
%H:%M:%S"),options.log_filename))
            f.write("%s tab is active.\n" %
self.options.active_tab)
            f.close()
        except :
            print_ = lambda *x : None
            else : print_ = lambda *x : None
            self.get_info()
            if self.orientation_points == {} :

```

```
        self.error_("Orientation points have not been
defined! A default set of orientation points has been
automatically added."),"warning")
        self.orientation( self.layers[min(0,len(self.layers)-1]
)
        self.get_info()

self.tools = {
    "name": "Laser Engraver",
    "id": "Laser Engraver",
    "penetration feed": self.options.laser_speed,
    "feed": self.options.laser_speed,
    "gcode before path": ("G4 P100 \n" +
self.options.laser_command + "\n" + "\nG4 P" +
self.options.power_delay),
    "gcode after path": ("G4 P100 \n" +
self.options.laser_off_command + "\n"+"G4 P100" + "\n" +
"G1 F" + self.options.travel_speed),
    }

    self.get_info()
    self.laser()

e = laser_gcode()
e.affect()
```