CONSTRUCCIÓN DE PROTOTIPO ROBOT INDUSTRIAL CARTESIANO X-Y, PARA EL TRAZADO SOBRE LÁMINAS DE METAL

AUTORES:

CHRISTIAN SANTODOMINGO CEBALLOS

DANIEL ALBERTO ALZATE QUINTERO

UNIVERSIDAD TECNOLÓGICA DE PEREIRA

FACULTAD DE INGENIERÍAS ELÉCTRICA, ELECTRÓNICA, FÍSICA Y CIENCIAS DE LA COMPUTACIÓN

PROGRAMA DE INGENIERÍA ELECTRÓNICA

PEREIRA

2014

CONSTRUCCIÓN DE PROTOTIPO ROBOT INDUSTRIAL CARTESIANO X-Y, PARA EL TRAZADO SOBRE LÁMINAS DE METAL

AUTORES:

CHRISTIAN SANTODOMINGO CEBALLOS

DANIEL ALBERTO ALZATE QUINTERO

PROYECTO DE GRADO PRESENTADO COMO REQUISITO PARCIAL PARA

OPTAR AL TÍTULO DE

INGENIERO ELECTRÓNICO

DIRECTOR:

INGENIERO ELECTRICISTA VICTOR DANIEL CORREA RAMÍREZ

UNIVERSIDAD TECNOLÓGICA DE PEREIRA

FACULTAD DE INGENIERÍAS ELÉCTRICA, ELECTRÓNICA, FÍSICA Y CIENCIAS DE LA COMPUTACIÓN

PROGRAMA DE INGENIERÍA ELECTRÓNICA

PEREIRA

2014

Nota de aceptación

_____

_____

_____

_____

_____

_____

Firma del presidente del jurado

_____

Firma del jurado

_____

Firma del jurado

Pereira 25 de Julio del 2014

DEDICATORIA

A mi madre Cecilia y mi tía Amparo porque con su esfuerzo y dedicación me han ayudado a salir adelante en la vida y me han enseñado la gran importancia que tiene la educación tanto en la vida como en el crecimiento a nivel personal.

Christian Santodomingo Ceballos.

A mi madre Miryam por ayudarme en todos los sentidos, por su esfuerzo, apoyo y dedicación; por el anhelo que ha tenido siempre de ver que sus hijos progresen y sean mejores personas cada día.

A mi hermana Maira por servirme como ejemplo de vida con su valentía, su perseverancia y su entrega; por los sacrificios realizados que nos ayudaron para salir adelante.

Daniel Alberto Alzate Quintero.

AGRADECIMIENTOS

A la Universidad Tecnológica de Pereira, por brindarnos el espacio y el recurso humano necesario para nosotros poder adquirir los conocimientos científicos y tecnológicos que hoy nos sirven para el desarrollo de este proyecto.

Un agradecimiento muy especial a todos aquellos docentes y compañeros que nos compartieron toda su experiencia y conocimiento así como también nos brindaron un ambiente de estudio adecuado.

A TecnoParque Colombia Nodo Pereira, por servir de guía y de apoyo en la elaboración de este proyecto con sus recursos tecnológicos e intelectuales, los cuales ayudaron a que el desarrollo de este proyecto se realizará de una manera más rápida y eficiente.

TABLA DE CONTENIDO

CAPÍTULO 1. INTRODUCCIÓN 9
1.1 DEFINICIÓN DEL PROBLEMA 10
1.1.1 PLANTEAMIENTO 10
1.1.2 FORMULACIÓN 10
1.1.3 SISTEMATIZACIÓN 10
1.2 JUSTIFICACIÓN 11
1.3 OBJETIVOS 12
1.3.1 OBJETIVO GENERAL 12
1.3.2 OBJETIVOS ESPECÍFICOS 13
1.4 MARCO HISTÓRICO 13
1.4.1 PRIMERA Y SEGUNDA GENERACIÓN 13
1.4.2 TERCERA GENERACIÓN 13
1.4.3 TENDENCIAS FUTURAS 14
1.4.4 CLASIFICACIÓN 14
1.4.5 ESTRUCTURA MECÁNICA 14
1.4.6 APLICACIONES 16
1.4.7 NUEVAS ESTRUCTURAS DE ROBOTS MANIPULADORES 17
CAPÍTULO 2. ESTRUCTURA MECÁNICA DEL PROTOTIPO 18
2.1 DESARROLLO MECÁNICO DEL PROTOTIPO 19
2.1.1 ÁREA DE TRABAJO DEL PROTOTIPO 19
2.1.2 SOPORTE DE GUÍAS PARA EL DESPLAZAMIENTO EN LOS EJES X,Y. 20
2.1.3 GUÍAS PARA DESPLAZAMIENTO X, Y 21
2.1.4 ÁNGULOS DE FIJACIÓN 23
2.1.5 RODAMIENTOS LINEALES 23
2.1.6 CHUMACERAS PARA LOS RODAMIENTOS LINEALES 24
2.1.7 SISTEMA DE MOVIMIENTO 25
2.1.8 DISPOSITIVO ACTUADOR FINAL 28

CAPÍTULO 3. ETAPA DE CONTROL DE POSICIONAMIENTO X-Y 29
3.1 HARDWARE 34
3.1.1 MICROCONTROLADOR 35
3.1.2 DRIVERS 36
3.1.3 CIRCUITO NOT 37
3.1.4 MOTORES PASO A PASO 38
3.1.5 FINALES DE CARRERA 39
3.1.6 FUENTE DE ALIMENTACIÓN 40
3.2 SOFTWARE 41
3.2.1 CÓDIGOS G (G-CODE) 42
3.2.2 TXAPUCNC_TX 42
3.2.3 TXAPUCNC_RX 43
3.2.4 INKSCAPE 44
3.2.5 GCODETOOLS 44
CAPÍTULO 4. IMPLEMENTACIÓN DE LA INTERFAZ GRÁFICA 46
4.1 IMPLEMENTACIÓN INKSCAPE 46
4.2 IMPLEMENTACIÓN TXAPUCNC_TX 51
4.2.1 VENTANA DE CONTROL 52
4.2.2 CONSOLA DE SALIDA 53
4.2.3 EDITOR DE CÓDIGOS G 53
4.2.4 SIMULADOR DE CÓDIGOS G 54
CAPÍTULO 5. PRUEBAS DE FUNCIONAMIENTO DEL PROTOTIPO 56
5.1 PRUEBA DE FUNCIONAMIENTO CON LÍNEAS RECTAS 57
5.2 PRUEBA DE FUNCIONAMIENTO LÍNEAS DIAGONALES 59
5.3 PRUEBA DE FUNCIONAMIENTO LÍNEAS CURVAS 60
CAPÍTULO 6. CONCLUSIONES Y RECOMENDACIONES 62

LISTA DE TABLAS

Tabla 1. Señales de control motor paso a paso. 37
Tabla 2. Señales de control resumidas motor paso a paso. 37
Tabla 3. Comandos G reconocidos por TxapuCNC. 42
Tabla 4. Botones ejecución de códigos G. 54
Tabla 5. Ficha técnica prototipo. 56

INDICE DE FIGURAS

Figura 1. Países con mayor densidad de robots industriales. 12
Figura 2. Estructura de un brazo robótico. 15
Figura 3. Tipos de configuración de los robots industriales. 16
Figura 4. Robot manipulador redundante. 17
Figura 5. Configuración robot tipo pórtico. 18
Figura 6. Área de trabajo del prototipo. 19
Figura 7. Soporte guías eje X. 20
Figura 8. Soporte guías eje Y. 21
Figura 9. Guías eje X. 22
Figura 10. Guías eje Y. 22
Figura 11. Ángulos de acero AISI A 36. 23
Figura 12. Rodamiento lineal. 24
Figura 13. Chumaceras rodamiento lineal. 24
Figura 14. Motor paso a paso. 25
Figura 15. Rueda dentada. 26
Figura 16. Correa dentada. 27
Figura 17. Acople motor rueda dentada. 27
Figura 18. Actuador final. 28
Figura 19. Sistema de control a lazo abierto. 29
Figura 20. Micro-controlador Arduino Mega 2560. 35
Figura 21. Drivers L298N para motores DC y PaP. 36
Figura 22. Circuito con integrado NOT. 38
Figura 23. Esquema motor paso a paso bipolar. 39
Figura 24. Interruptor final de carrera. 40
Figura 25. Fuente de alimentación. 41
Figura 26. TxapuCNC_TX 43
Figura 27. Inicio Inkscape. 46
Figura 28. Crear y editar objetos de texto. 47
Figura 29. Trayecto. 48
Figura 30. Puntos de orientación. 49
Figura 31. Biblioteca de herramientas. 50
Figura 32. Trayecto a Gcode. 51
Figura 33. Ventana de control. 52
Figura 34. Consola de salida. 53
Figura 35. Editor códigos G. 54
Figura 36. Simulador códigos G. 55
Figura 37. Vista previa letra “T”. 57

Figura 38. Trazado letra “T” mediante prototipo.....58
Figura 39. Resultado obtenido trazo de rectas. 58
Figura 40. Vista previa letra “M”.....59
Figura 41. Resultado obtenido trazo de diagonal.....60
Figura 42. Vista previa letra “C”. 60
Figura 43. Resultado obtenido trazo de curva. 61

CAPÍTULO 1. INTRODUCCIÓN

En la actualidad la tecnología robótica está avanzando muy rápidamente; los sistemas robóticos están siendo desarrollados cada vez más con mayor autonomía, lo cual significa un reto a involucrarse en este auge tecnológico. En el ámbito industrial nacional, específicamente en las pequeñas y medianas empresas, se crea la necesidad de incentivar a la adquisición de estas tecnologías, puesto que actualmente esto va ligado a la competitividad; una manera de que se den a conocer las ventajas de la utilización de robots en la industria, de una forma rápida y económica, es con la implementación de robots cartesianos los cuales son utilizados comúnmente para: el apilamiento de estructuras, tareas de soldadura, pintura, trazado, corte y perforación de placas metálicas etc.

En el presente trabajo se mostrará la información necesaria para la construcción de un prototipo robot de coordenadas cartesianas X-Y, así como también la información correspondiente al software utilizado para el control del prototipo.

1.1 DEFINICIÓN DEL PROBLEMA

1.1.1 PLANTEAMIENTO

En una visita técnica realizada a la microempresa INDUSTRIAS CM, ubicada en la ciudad de Cartago departamento del Valle, se conocieron los procesos que se utilizan para la producción de closets, cocinas, archiveros y gabinetes metálicos. Estos procesos consisten en:

- Recepción y almacenamiento del material
- Trazado sobre lámina de metal
- Corte
- Doblado
- Soldadura
- Pulido
- Pintura
- Horno

El proceso en el que se realiza el trazado sobre las láminas, es primordial para pasar a los procesos de corte y doblado; el proceso de trazado sobre lámina de metal actualmente se hace de forma manual, al ser un proceso crítico en la producción es necesario garantizar una muy buena precisión, mayor velocidad y minimizar los errores humanos.

1.1.2 FORMULACIÓN

¿Qué tipo de mecanismo se podría desarrollar para que efectuara el proceso de trazado sobre láminas de metal?

1.1.3 SISTEMATIZACIÓN

¿Cuál debería ser el paso inicial para la construcción de un prototipo de robot industrial cartesiano X-Y, para el trazado sobre láminas de metal?

¿Cuáles son las especificaciones técnicas que mejor se adecuan en cuanto a la forma y funcionamiento del prototipo de robot cartesiano X-Y, para el trazado sobre láminas de metal en la empresa?

¿Cómo se ayudaría a disminuir el impacto ambiental en la construcción de un prototipo de robot industrial cartesiano X-Y, para el trazado sobre láminas de metal?

¿Cómo se puede llegar a tener un buen funcionamiento previamente definido para el prototipo de robot cartesiano X-Y, para el trazado sobre láminas de metal?

¿Cuál sería la mejor forma de que un operario común pueda utilizar sin ninguna dificultad el prototipo de robot cartesiano X-Y, para el trazado sobre láminas de metal?

¿De qué forma se podría evaluar el desempeño óptimo del prototipo de robot cartesiano X-Y, para el trazado sobre láminas de metal?

1.2 JUSTIFICACIÓN

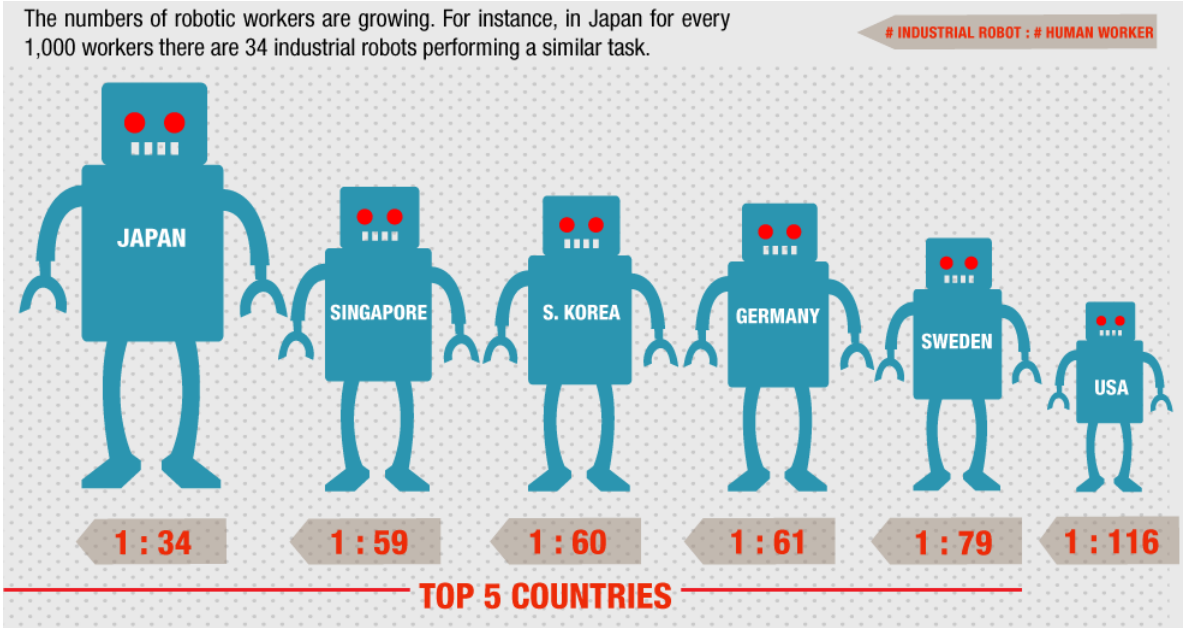
El uso de robots en la industria ha venido experimentando un crecimiento a nivel global en los últimos años, debido a las ventajas que estos ofrecen, entre ellas: mayor precisión, menor cantidad de tiempo utilizado en los procesos de producción, pueden realizar trabajos en los cuales los seres humanos estarían arriesgando su integridad física, trabajar una gran cantidad de horas seguidas; esto se ve reflejado en mejoras en la calidad de los productos y en tiempos de entrega más bajos, lo que se traduce a una ventaja significativa en el mercado.

En Colombia la mayoría de robots industriales son utilizados por las grandes empresas, entre ellas la que más destaca es la industria automotriz, pero las pequeñas y medianas empresas no pueden contar fácilmente con esta tecnología debido a los altos costos que esto conlleva; esto no solo sucede en Colombia, la mayoría de países Latinoamericanos presentan una muy baja implementación en el uso de robótica comparado con las grandes potencias.

Para dimensionar la desventaja que se tiene frente a otros países, se muestra un estudio de IEEE spectrum “The Rise of the Machines” presentado en Diciembre del 2008, el cual presenta los seis (6) países con mayor densidad de robots en el mundo; los datos muestran la cantidad de robots en comparación con la cantidad de trabajadores del mismo sector industrial. Estos datos se pueden ver en la figura 1.

En conclusión, para entrar a competir en un mercado global, la industria nacional debería tener la infraestructura necesaria para mejorar la calidad y confiabilidad de sus productos, una forma de contribuir es facilitando la adquisición de estas tecnologías a las pequeñas y medianas empresas para que exploren y conozcan las ventajas que se pueden obtener con el uso de robots en la industria y que no sólo las grandes empresas estén en la capacidad de hacer uso de esta tecnología.

Figura 1. Países con mayor densidad de robots industriales.



GUIZZO, E. The Rise of the Machines. Spectrum, IEEE. Volume: 45 Issue: 12 p.88. Dec. 2008.

1.3 OBJETIVOS

1.3.1 OBJETIVO GENERAL

Construir un prototipo de un robot industrial cartesiano X-Y, para el trazado sobre láminas de metal.

1.3.2 OBJETIVOS ESPECÍFICOS

- Desarrollar un sistema mecánico (prototipo) para el trazado sobre láminas de metal.
- Seleccionar el tipo de control de posicionamiento adecuado que permita un buen desempeño.
- Implementar una interfaz humano-máquina que permita un uso sencillo y rápido del prototipo.
- Verificar el correcto funcionamiento del prototipo.

1.4 MARCO HISTÓRICO

La introducción de los microprocesadores en los años 70 ha hecho posible que la tecnología de los robots haya sufrido grandes avances. La fusión de la electrónica y la mecánica ha hecho posible el robot actual, hasta tal punto que los japoneses han acuñado el término mecatrónica para describirla. El año 1980 fue llamado primer año de la era robótica porque la producción de robots industriales aumentó ese año un 80% respecto al año anterior. En la actualidad los robots son utilizados cada vez más en la industria, pues algunos de ellos pueden realizar el mismo trabajo que un ser humano de forma más rápida, eficiente y hacerlo en ambientes que serían demasiado nocivos para las personas, por estas razones se prevé un aumento en la demanda en la tecnología robótica para los próximos años.

1.4.1 PRIMERA Y SEGUNDA GENERACIÓN

"La primera generación de robots era reprogramable, de tipo brazo y sólo podían memorizar movimientos repetitivos, asistidos por sensores internos que les ayudaban a realizar sus movimientos con precisión. La segunda generación coincide con finales de los 70 y ya poseen sensores externos que proporcionan al robot información del exterior. Pueden tomar limitadas decisiones, reaccionar ante el entorno de trabajo y se les conoce como robots adaptativos.

1.4.2 TERCERA GENERACIÓN

"La tercera generación emplea la inteligencia artificial, hace uso de microprocesadores avanzados y es capaz de realizar razonamientos lógicos e incluso aprender".[1]

[1] RUIZ de GARIBAY PASCUAL, Jonathan. Robótica: Estado del arte. p. 11. Universidad de Deusto. Sep. 2007

1.4.3 TENDENCIAS FUTURAS

"*Durante muchos años, los robots han sido considerados útiles sólo si se empleaban como manipuladores industriales, mientras que en la actualidad los roles de éstos han dado un giro tremendo. Estos nuevos robots móviles pueden realizar tareas en entornos diferentes y se les conoce como robots de servicio.*

Proporcionan muchas funciones de utilidad y se emplean para el ocio, educación, medicina, etc. La creciente utilización de los robots muestra el papel tan importante que pueden desempeñar en el futuro".[2]

1.4.4 CLASIFICACIÓN

Resulta tremendamente difícil realizar una clasificación de los tipos de robots existentes, debido a la multitud de parámetros sobre los que se puede realizar dicha clasificación. De manera general, y basándose en su morfología, los robots se suelen dividir en los siguientes tipos:[3]

- Robot industrial o manipulador
- Robot móvil
- Robot androide o humanoide
- Robot zoomórfico

1.4.4.1 Robots industriales

Los robots manipuladores o robots industriales (conocidos así porque inicialmente fueron usados masivamente en la industria) fueron los encargados de inaugurar la era de los robots en los años 60, con la herencia adquirida de los primeros tele-operadores. Por ello, es el área de la robótica donde la investigación está más avanzada.[4]

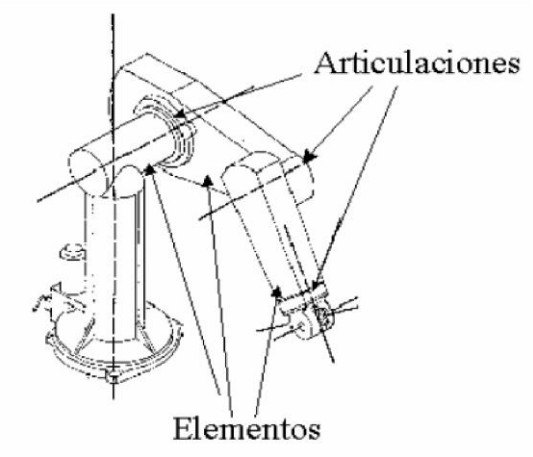
1.4.5 ESTRUCTURA MECÁNICA

Formalmente, se denomina base al punto de apoyo del robot, generalmente sujeto de forma fija al suelo. Los elementos o eslabones van unidos por medio de diferentes articulaciones, que permiten un movimiento relativo entre cada dos eslabones consecutivos. En la parte final, se sitúa el elemento terminal o efector

[2] Ibid, p. 12
[3] Ibíd., p. 15
[4] Ibid., p. 18

final, que son los encargados de interaccionar directamente con el entorno del robot.[5]

Figura 2. Estructura de un brazo robótico.



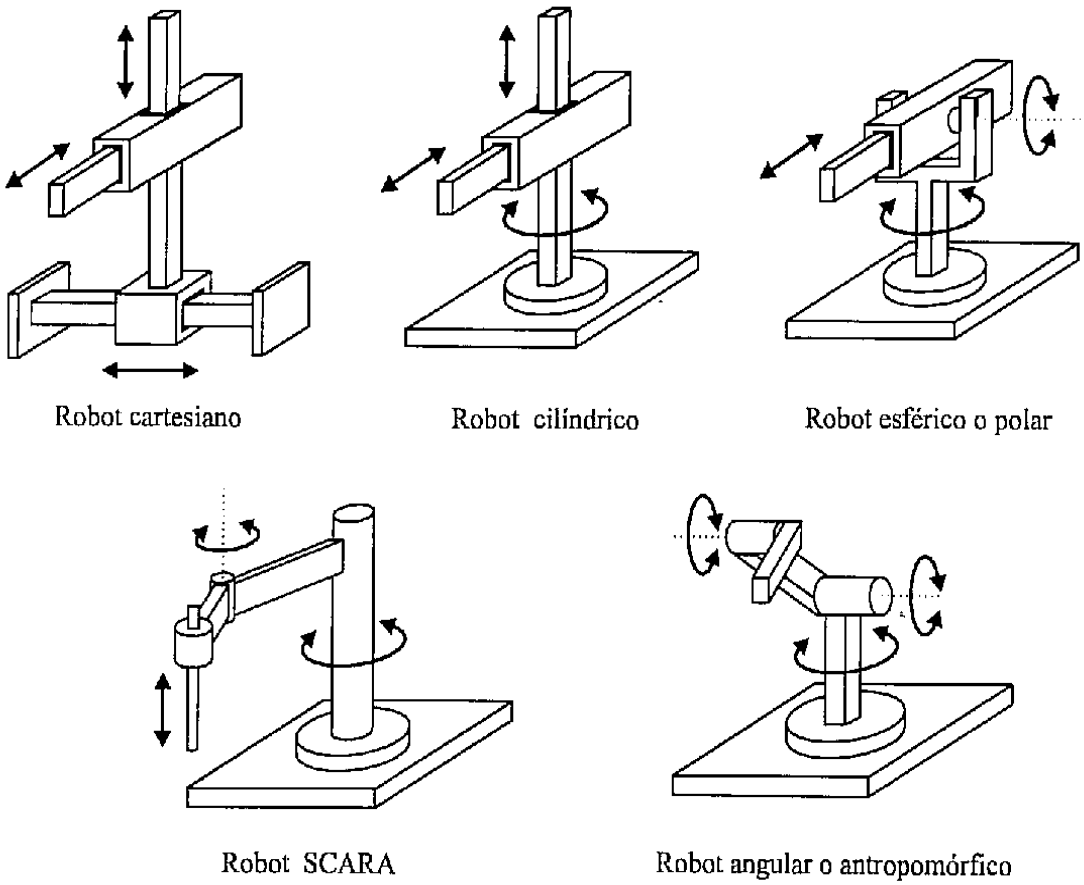
El movimiento de cada articulación puede ser de desplazamiento, de giro o de una combinación de ambos. De este modo son posibles los 5 tipos diferentes de articulaciones, con sus diferentes grados de libertad.[6]

"Los grados de libertad son el número de movimientos independientes que puede realizar cada articulación con respecto a la anterior. El número de grados de libertad de un robot viene dado por la suma de los grados de libertad de cada una de sus articulaciones.

El empleo de diferentes combinaciones de articulaciones en un robot, da lugar a diferentes configuraciones con características a tener en cuenta tanto en el diseño y construcción del robot como en su aplicación".[7]

[5] Ibid., p. 19
[6] Ibid., p. 20
[7] Ibid., p. 21

Figura 3. Tipos de configuración de los robots industriales.



BARRIENTOS, Antonio, PEÑIN, Luis Felipe, BALAGUER, Carlos y ARACIL Rafael. Fundamentos de Robótica, Madrid: McGRAW HILL, 1997. 327 p.

1.4.6 APLICACIONES

"Los robots manipuladores tienen su principal foco de trabajo en la industria, automatizando los procesos de producción o almacenaje. Generalmente no trabajan de forma independiente sino en conjunto con otras máquinas herramientas formando células de trabajo. Se enumeran a continuación algunos ejemplos:

- *Operaciones de procesamiento, como soldadura, pintura, etc. Este tipo de robots son muy comunes en la industria de la automoción.*
- *Operaciones de ensamblaje, donde el trabajo repetitivo facilita el uso de este tipo de robots.*
- *Operaciones de empaque (en tarimas o pallets), agilizando el proceso y manejando grandes pesos.*

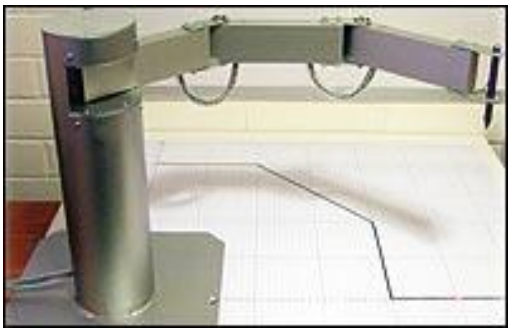
- *Otro tipo de operaciones como pueden ser remachados, estampados, corte por chorro de agua, sistemas de medición, etc"*.[8]

1.4.7 NUEVAS ESTRUCTURAS DE ROBOTS MANIPULADORES

1.4.7.1 Robots manipuladores redundantes

Básicamente se componen de un gran número de eslabones, los cuales además tienen la cualidad de ser iguales y repetitivos. Se denominan también robots de tipo serpiente. Estos robots tienen la capacidad de introducirse por espacios poco estructurados debido a su flexibilidad.[9]

Figura 4. Robot manipulador redundante.



Disponible en internet: http://www.emb.cl/electroindustria/articulo.mvc?xid=1579&edi=83

[8] Ibíd., p. 21
[9] Ibíd., p. 24

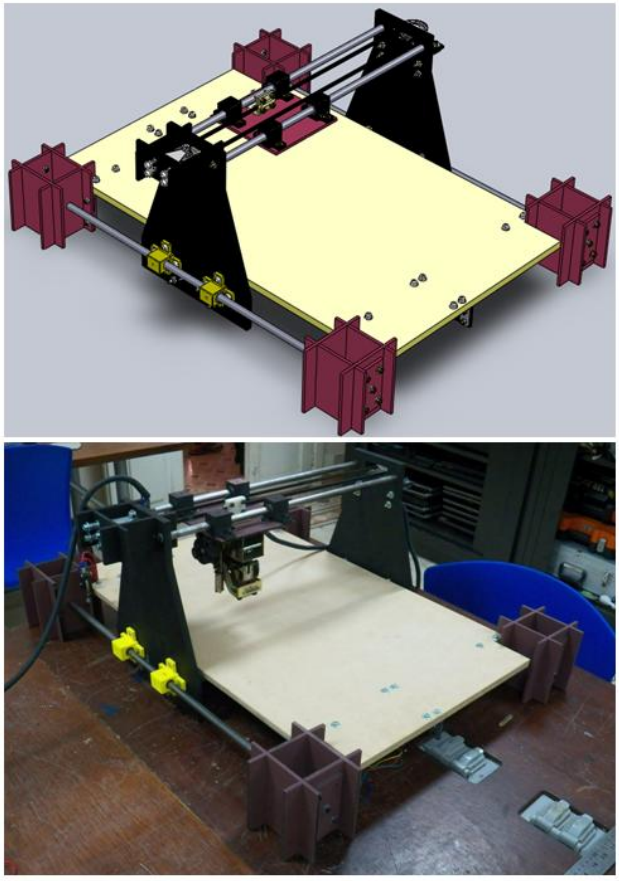
CAPÍTULO 2. ESTRUCTURA MECÁNICA DEL PROTOTIPO

Con base en el requerimiento planteado para la función del prototipo, se plantea el desarrollo de un sistema basado en un robot cartesiano X,Y,Z. Dicho sistema busca realizar el trazado sobre placas metálicas con un tamaño de 35 cm por 26 cm aproximadamente.

El sistema se plantea mediante tres ejes coordenados, a través de los cuales se lleva a cabo el desplazamiento de un determinado componente. Luego de haberse considerado diversas opciones para tal fin, se procedió a realizar pruebas preliminares mediante un sistema de eje-rodamiento lineal, el cual brindó un desempeño adecuado para esta aplicación.

Se procede entonces a plantear dos sistemas básicos de desplazamiento mediante rodamientos lineales (ejes X y Y) y un tercer sistema mediante engranajes y un motorreductor DC para el desplazamiento del tercer eje (eje Z), según se observa en la figura 5.

Figura 5. Configuración robot tipo pórtico.



2.1 DESARROLLO MECÁNICO DEL PROTOTIPO

El desarrollo mecánico del prototipo se realizó desde varias áreas de trabajo. En un principio se plantea un sistema base (mediante el desarrollo de un estado del arte y una vigilancia tecnológica), el cual arrojó que un sistema basado en un robot tipo pórtico puede suplir la necesidad inicial. Posterior a ello se realiza un modelo base (borrador), el cual basados en el requerimiento planteado, arroja un modelo preliminar.

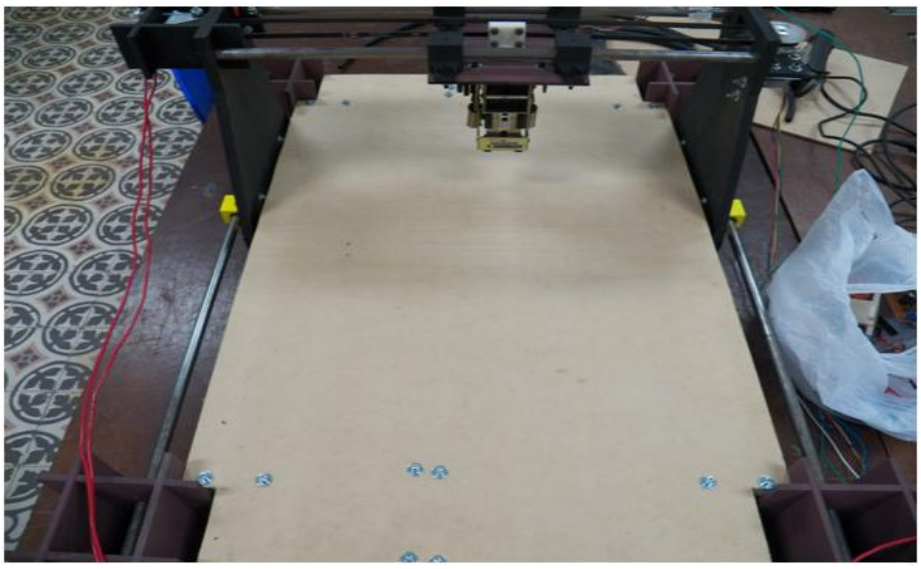
En un proceso posterior se lleva a cabo el desarrollo de aprendizaje en el manejo del software SolidWorks, para posteriormente realizar un modelo tridimensional del sistema completo (figura 5).

En términos generales, se usan varios materiales, como por ejemplo MDF, acero AISI 1045, termo-polímero ABS Plus, acrílico, entre otros. El prototipo cuenta con una serie de componentes globales, los cuales se ilustran más detalladamente en los numerales siguientes.

2.1.1 ÁREA DE TRABAJO DEL PROTOTIPO

Para el área de trabajo del prototipo se obtuvo una placa de 463x740x12mm de material MDF, de la cual se usa como área de trabajo efectivo 26cm por 35cm aproximadamente, esta placa se encarga de soportar el material al cual posteriormente se le realizará el trazado.

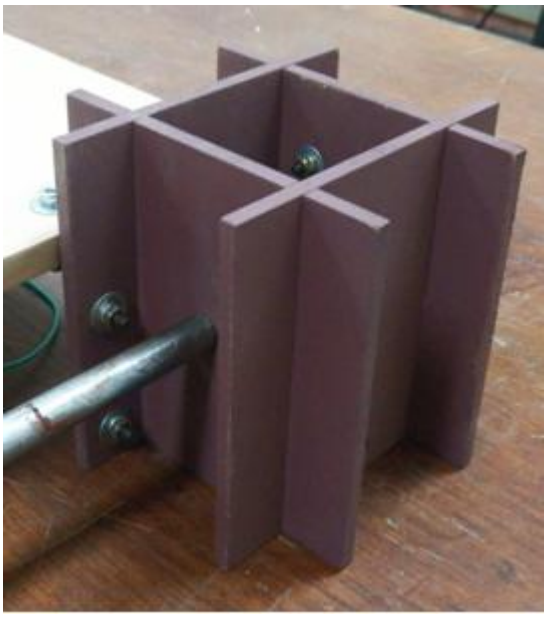
Figura 6. Área de trabajo del prototipo.



2.1.2 SOPORTE DE GUÍAS PARA EL DESPLAZAMIENTO EN LOS EJES X,Y

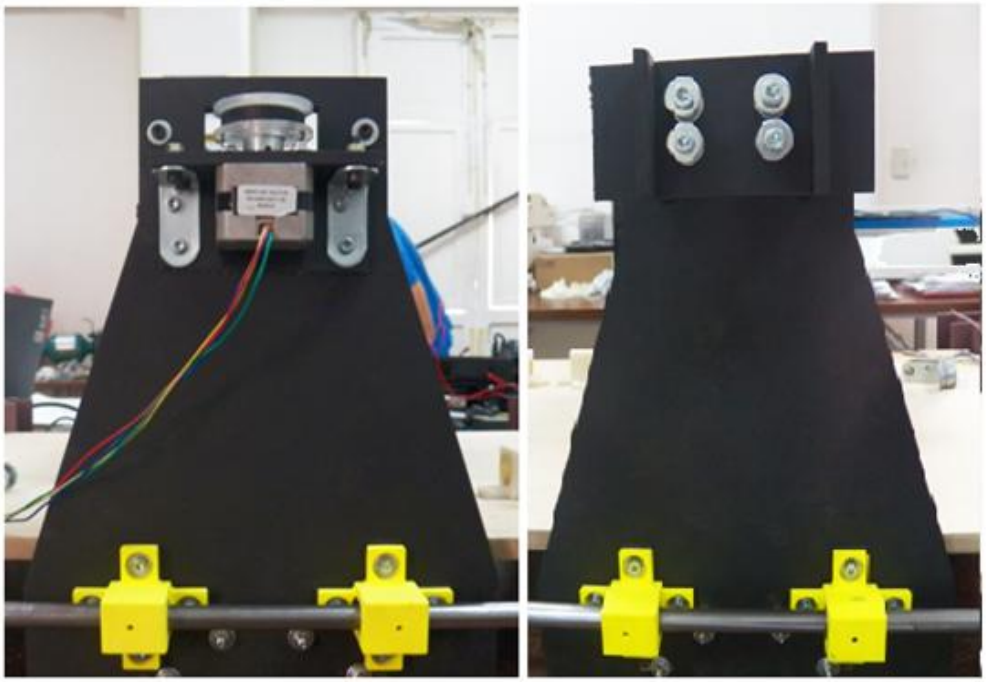
El soporte para las guías eje X consiste en un armazón modular de cuatro (4) partes de 12mm por 12mm, realizados en fibra de media densidad (MDF), los cuales tienen como finalidad de dar apoyo a los ejes que definirán el recorrido del prototipo. Los cortes fueron realizados con el cortador láser LaserPro C180.

Figura 7. Soporte guías eje X.



Para el eje Y se implementó un diseño modular más complejo, el cual permite dar el soporte tanto a los ejes como a las chumaceras de los rodamientos para posteriormente se pueda fijar el motor y el tensor necesarios para darle al actuador el movimiento en el sentido del eje Y (véase figura 8)

Figura 8. Soporte guías eje Y.



2.1.3 GUÍAS PARA DESPLAZAMIENTO X, Y

Para la fabricación de las guías se adquirió una barra de acero AISI 1045 de 2.20m de largo y 12mm de diámetro, la cual posteriormente fue cortada en cuatro (4) partes las cuales fueron maquinadas con la ayuda de un torno. En los extremos de cada una de las barras se hicieron roscados para tornillo milimétrico de 4 mm de diámetro con el propósito de ajustarlos a los soportes de fibra de media densidad (MDF) y así conformar las guías para los ejes del prototipo (véase figura 9)

Figura 9. Guías eje X.

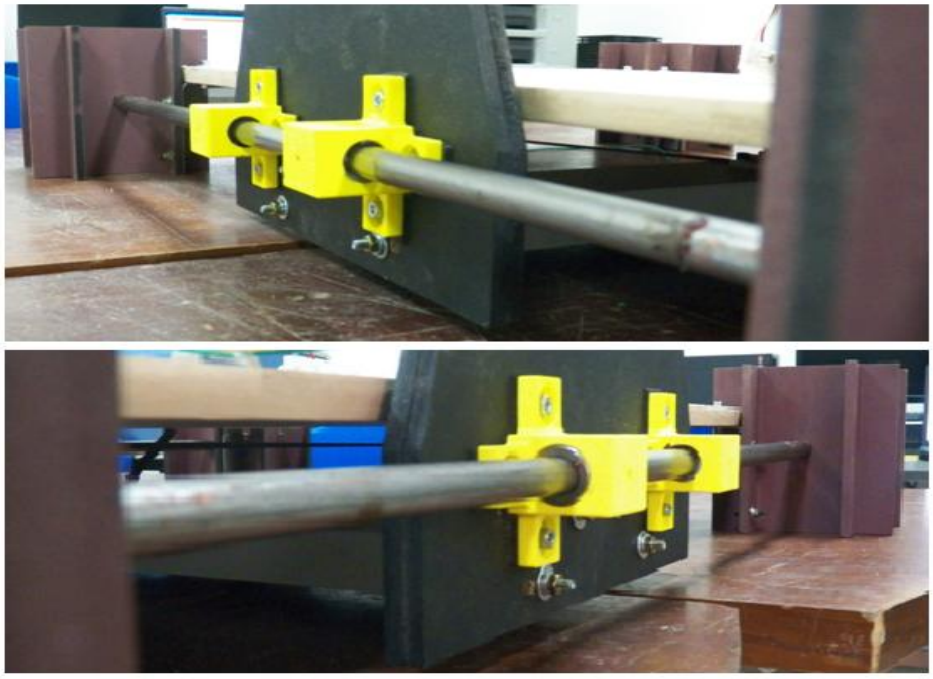
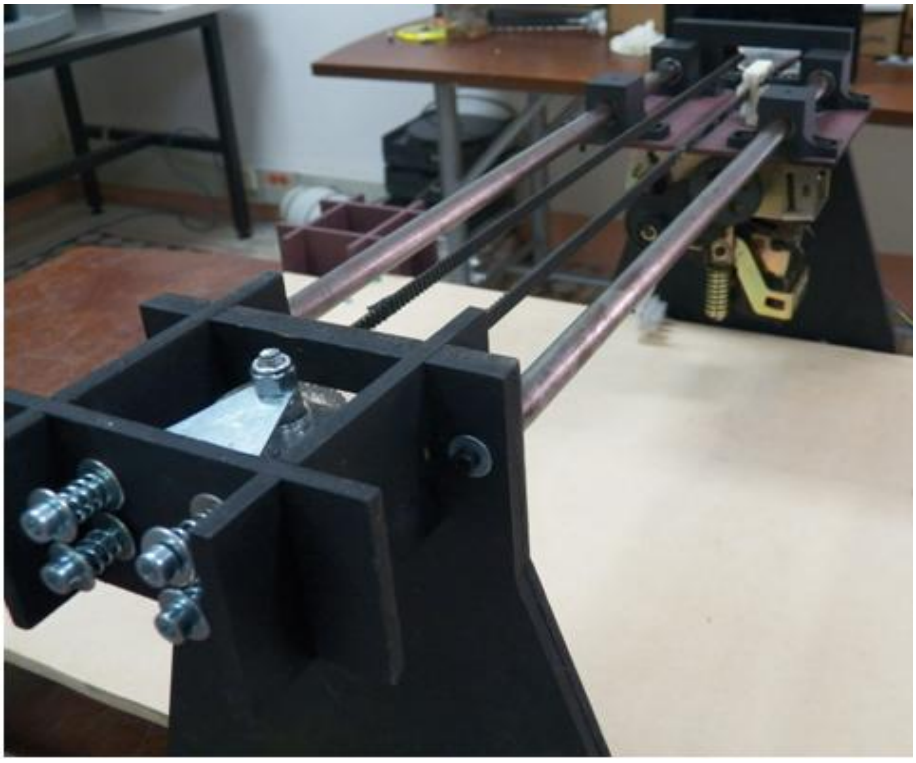


Figura 10. Guías eje Y.



2.1.4 ÁNGULOS DE FIJACIÓN

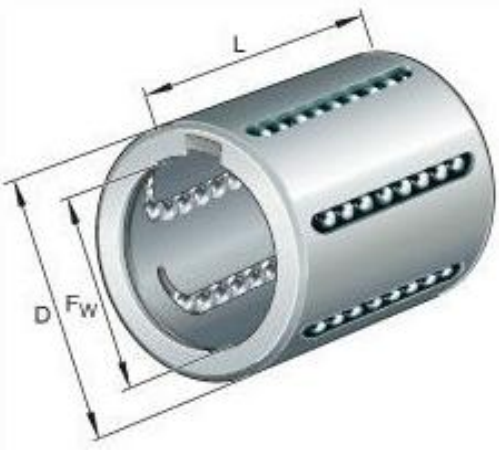
Se adquirieron diez (10) ángulos de acero AISI A36, cuatro (4) de ellos para fijar los soportes del eje X a la superficie de trabajo y a su vez darle más soporte estructural al prototipo; puesto que las cuatro (4) esquinas quedan unidas entre sí por medio de un cuerpo rígido. Los seis (6) ángulos restantes se utilizaron de la siguiente manera: cuatro (4) para soportar cada motor en las superficies X e Y respectivamente, por último dos (2) fueron utilizados para soportar el tensor de forma perpendicular a la superficie del eje X (véase figura 11)

Figura 11. Ángulos de acero AISI A 36.

2.1.5 RODAMIENTOS LINEALES

Se adquirieron ocho (8) rodamientos lineales de referencia KH1228 de 28mm de largo (L), 12mm diametro interno (Fw) y 17mm diámetro externo (D), los cuales se usaron de a dos (2) por cada eje de acero con el fin de dar movilidad al prototipo en los ejes X e Y, según se muestra en la figura 12

Figura 12. Rodamiento lineal.

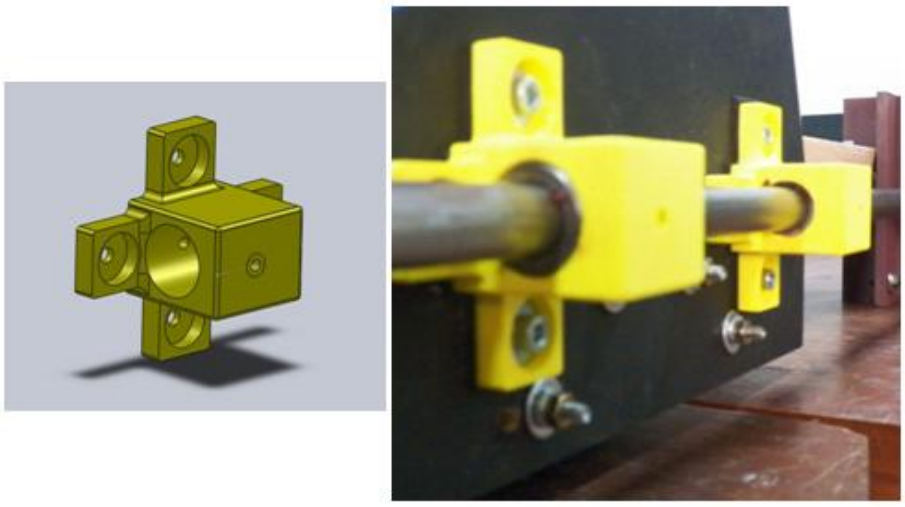


Disponible en internet: http://i00.i.aliimg.com/img/pb/717/516/461/461516717_358.jpg

2.1.6 CHUMACERAS PARA LOS RODAMIENTOS LINEALES

Debido a que las chumaceras para los rodamientos lineales no se encuentran en el mercado, se realizó el diseño de estas utilizando el programa CAD Solid Works, con el objetivo de realizar su posterior proceso de prototipado en una impresora 3D (disponible en Tecnoparque durante el acompañamiento) en un material estructuralmente adecuado para tal fin (ABS Plus), para determinar su adecuada resistencia estructural se procedió a realizar simulaciones de carga, las cuales fueron idóneas para la aplicación que se requiere.

Figura 13. Chumaceras rodamiento lineal.



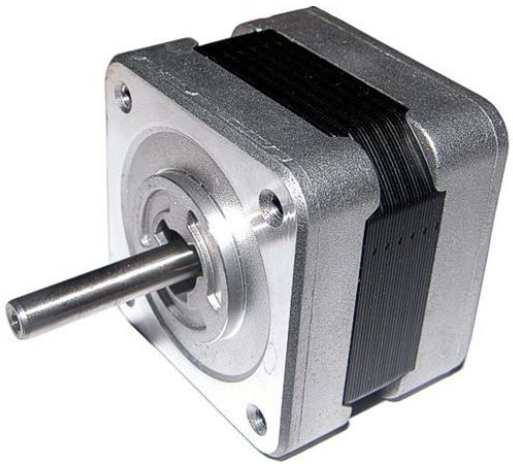
2.1.7 SISTEMA DE MOVIMIENTO

El sistema de movimiento para cada eje (X y Y), está compuesto por un (1) motor paso a paso, dos (2) ruedas dentadas y una (1) correa dentada de referencia 60S3M681 la cual tiene un ancho de 6mm y un paso de 3mm.

2.1.7.1 Motores paso a paso

Para generar el movimiento en los ejes X e Y del prototipo, se hizo la adquisición de dos (2) motores paso a paso los cuales se han ubicado y adaptado de modo tal que permitan realizar los movimientos lineales deseados. Los motores utilizados en el prototipo son motores paso a paso bipolares de 1.8° de precisión, con 2.34kg-cm de torque máximo, un peso de 0.2kg y una corriente sin carga de 300mA.

Figura 14. Motor paso a paso.



Disponible en internet:
http://www.inti.edu.sv/compu2013/hadatbord/revista/backgrounds/_electronic/motorpsp.JPG

2.1.7.2 Correas y ruedas dentadas

Para lograr la conversión de movimiento rotatorio del motor a un movimiento lineal se utilizaron unas correas dentadas, provenientes de unas antiguas fotocopiadoras (debido a la dificultad para conseguir correas nuevas de un paso pequeño y una longitud como la que se requiere en el proyecto), a partir de éstas se diseñaron las ruedas dentadas, con la ayuda del programa CAD Solidworks. Teniendo como referentes el ancho y el paso de la correa (6mm y 3mm respectivamente) de la cual se disponía para poder dimensionar el ancho de la rueda y la separación del dentado respectivamente, posteriormente se obtuvo una vista frontal de la rueda dentada diseñada en Solidworks en formato (.DWG), esta extensión es reconocible por el programa Corel Draw en el cual se realizaron ajustes, con el diseño terminado se logró fabricar la rueda dentada en material acrílico, esto se hizo con la ayuda de una impresora láser (el formato reconocido por la impresora láser para realizar el proceso debe ser .cdr).

Para poder unir las ruedas dentadas a los ejes de los motores se tuvieron que maquinar en un torno dos (2) acoples de aluminio con prisioneros a 90° de 1/8 de pulgada.

Figura 15. Rueda dentada.

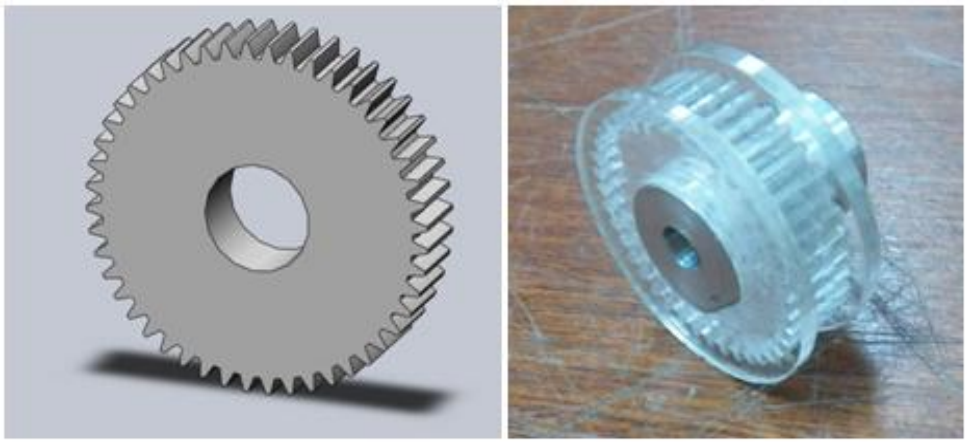


Figura 16. Correa dentada.

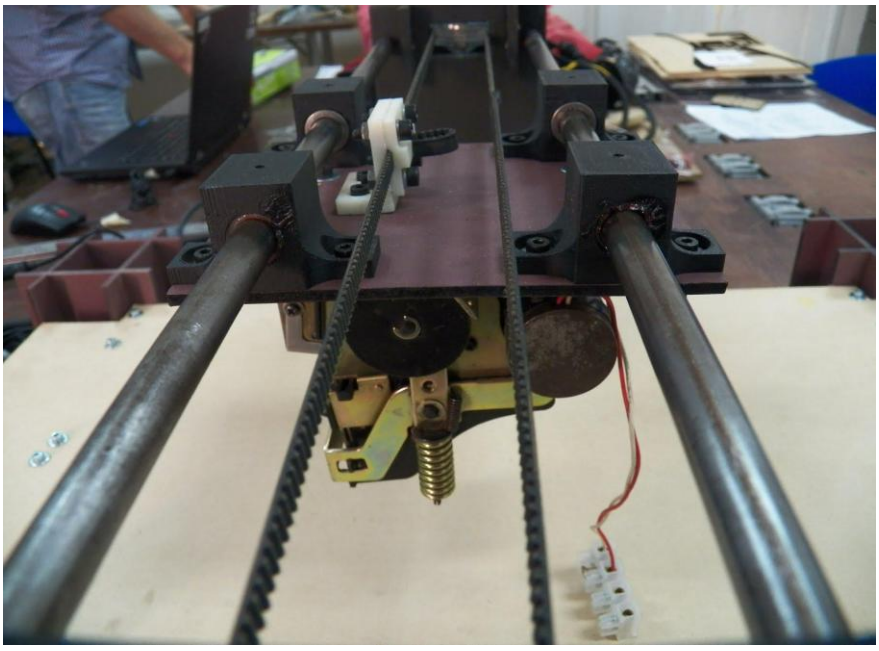
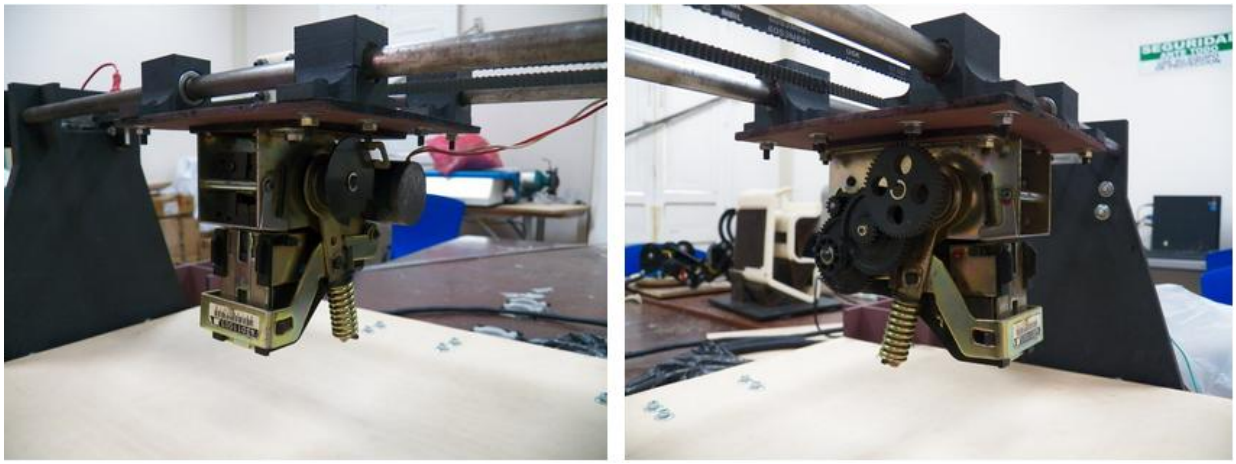


Figura 17. Acople motor rueda dentada.

2.1.8 DISPOSITIVO ACTUADOR FINAL

Para el accionamiento del elemento final encargado de realizar las trazas según la trayectoria programada, se utilizó un (1) dispositivo actuador de una antigua máquina codificadora térmica, la cual se utilizaba para poner etiquetas; este dispositivo se compone de: una estructura metálica, un (1) motor DC de 12V, dos (2) resortes y cuatro (4) engranes. Su movimiento se limita simplemente a subir o bajar el elemento final según sea necesario.

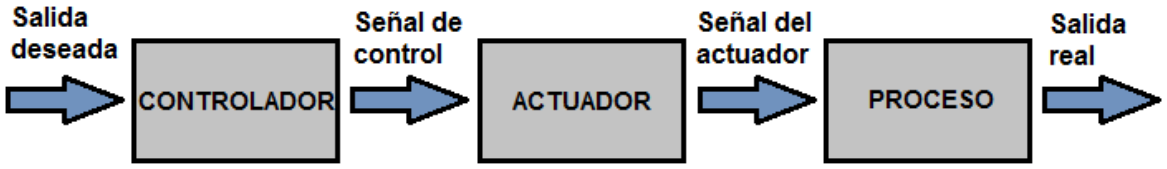
Figura 18. Actuador final.



CAPÍTULO 3. ETAPA DE CONTROL DE POSICIONAMIENTO X-Y

El control de posicionamiento que se implementó para el prototipo del robot cartesiano X-Y es un sistema de control a lazo abierto que funciona de la siguiente manera. Los comandos de posicionamiento son enviados vía comunicación serial, desde un programa desarrollado en gambas3 (TxapuCNC_TX) que corre sobre el sistema operativo Linux, hacia un controlador Arduino Mega 2560 el cual tiene cargado un programa (TxapuCNC_RX) el cual se subdivide en 4 partes las cuales se encargan de interpretar los comandos de posicionamiento para enviar las señales de control, en este caso, hacia tres(3) drivers L298N los cuales proveen potencia a cada uno de los motores de cada uno de los ejes; los motores utilizados son: dos motores paso a paso bipolares para el movimiento en los ejes X y Y, un motor DC 12V para el accionamiento del dispositivo actuador final en eje Z.

Figura 19. Sistema de control a lazo abierto.



Lo que se pretende es controlar el número de pasos que debe dar cada motor en cada eje y cuando activar el dispositivo final, con el propósito de lograr el trazado de una trayectoria deseada.

A continuación se detallarán las 4 partes en las que se subdivide el programa TxapuCNC_RX, para lograr comprender el modo de funcionamiento del prototipo a nivel electrónico.

_**init :** su principal función es la de permitir designar cuantos pasos del motor paso a paso son necesarios para una traslación lineal de 1mm, para que de esta manera el programa pueda reconocer mediante el conteo de pasos qué distancia ha recorrido, para así poder mandar las señales de “ok” cuando el prototipo ha alcanzado su destino según las medidas previamente especificadas en el diseño de la trayectoria. El programa también brinda la posibilidad de trabajar las unidades en pulgadas a preferencia del usuario. A continuación se muestra un

segmento de código en el cual se asignan estos valores mencionados anteriormente.

```
#define X_STEPS_PER_INCH 24.5
#define X_STEPS_PER_MM   1
```

Esto se realiza para cada eje de traslación del prototipo. En esta parte también se permite definir la velocidad máxima a la cual operaran los ejes del prototipo.

```
//our maximum feedrates
#define FAST_XY_FEEDRATE 2400
#define FAST_Z_FEEDRATE  2400
```

Los valores anteriormente mostrados para la velocidad son empíricos, se asignan simplemente observando el comportamiento de los motores, a medida que se varía este valor.

Por último, también nos permite definir los pines que se desean implementar en el arduino para el envío de las señales hacia los drivers que controlan a los motores, además los pines que reciben la señal de final de carrera y el pin de enable para desactivar los motores, cuando estos no están en funcionamiento, para así ahorrar energía y evitar sobrecalentamientos de la máquina.

```
//cartesian bot pins
#define X_STEP_PIN 8
#define X_DIR_PIN 9
#define X_MIN_PIN 4
#define X_MAX_PIN 2
#define X_ENABLE_PIN 15
```

El código mostrado anteriormente debe ser emulado para cada eje de traslación del prototipo.

Process_string: La función principal de este subcódigo es la de reconocer qué tipo de instrucción fue recibida por el arduino según los códigos G, y de acuerdo a estos enviar constantemente las coordenadas de posicionamiento y las señales de inicio y parada.

```
//did we get a gcode?
```

```
if (
        has_command('G', instruction, size) ||
        has_command('X', instruction, size) ||
        has_command('Y', instruction, size) ||
        has_command('Z', instruction, size)
)
```

Como se puede apreciar, en el segmento de código anterior se implementa una función en el código llamada has_command la cual se encarga de reconocer una variable tipo “char” (carácter) dentro de un arreglo.

Stepper_control: Una vez definido el tipo de movimiento que se va a realizar, esta parte del código es la encargada de enviar los pulsos eléctricos a los drivers mediante el arduino. Acá se configuran los pines como entradas o salidas digitales, según sea la necesidad.

```
void do_step(byte pinA, byte pinB, byte dir)
{
    switch (dir << 2 | digitalRead(pinA) << 1 | digitalRead(pinB)) {
      case 0: /* 0 00 -> 10 */
      case 5: /* 1 01 -> 11 */
        digitalWrite(pinA, HIGH);
        break;
      case 1: /* 0 01 -> 00 */
      case 7: /* 1 11 -> 10 */
        digitalWrite(pinB, LOW);
        break;
      case 2: /* 0 10 -> 11 */
      case 4: /* 1 00 -> 01 */
        digitalWrite(pinB, HIGH);
        break;
      case 3: /* 0 11 -> 01 */
      case 6: /* 1 10 -> 00 */
        digitalWrite(pinA, LOW);
        break;
    }
      delayMicroseconds(5);
}
```

Como se puede apreciar, en el código anterior se muestra una serie de casos para los cuales se pondrán en alto o en bajo ciertos pines designados previamente en el subcódigo Init, según sea el caso y la dirección del movimiento.

```
pinMode(X_STEP_PIN, OUTPUT);
pinMode(X_DIR_PIN, OUTPUT);
pinMode(X_ENABLE_PIN, OUTPUT);
pinMode(X_MIN_PIN, INPUT);
pinMode(X_MAX_PIN, INPUT);
```

El código mostrado anteriormente muestra cómo designar pines específicos del arduino como entradas o salidas; este proceso se realiza para cada eje.

TxapuCNC_RX01: Este es el código principal del arduino, es el punto de inicio donde empieza a correr el programa. Desde este código se hace llamado a los otros subcódigos. Además es el encargado de definir la velocidad de transferencia de datos y también de estar testeando constantemente el puerto USB del arduino, en busca de algún comando tipo char.

```
void setup()
{
	//Do startup stuff here
	Serial.begin(19200);
	Serial.println("start");

	//other initialization.
	init_process_string();
	init_steppers();

}
```

En las líneas del código anterior podemos apreciar cómo se define la velocidad de transferencia de datos en 19200 y el llamado a inicializar los otros subprocesos del código.

```
void loop()
{
	char c;

	//keep it hot!


	//read in characters if we got them.
	if (Serial.available() > 0)
	{
		c = Serial.read();
		no_data = 0;

		//newlines are ends of commands.
		if (c != '\n')
		{
			palabra[serial_count] = c;
			serial_count++;
		}
	}
	//mark no data.
	else
	{
		no_data++;
		delayMicroseconds(100);
	}

	//if theres a pause or we got a real command, do it
	if (serial_count && (c == '\n' || no_data > 100))
	{
		//process our command!
		process_string(palabra, serial_count);

		//clear command.
		init_process_string();
	}

	//no data?  turn off steppers
	if (no_data > 1000)
```

```
        disable_steppers();
}
```

Lo anterior es una muestra de cómo se lee el puerto en busca de la existencia de algún código, en caso de transcurrir cierto tiempo sin reconocer alguno, el arduino desactiva los motores para ahorrar energía.

3.1 HARDWARE

El hardware que se utilizó en el sistema de control de posicionamiento del prototipo robot cartesiano X-Y consta de:

Un (1) computador para la ejecución de los programas necesarios en la implementación de la interfaz gráfica y para el envío de la señal de salida deseada.

Un (1) microcontrolador Arduino Mega 2560 (figura 20) encargado de recibir la señales enviadas desde el computador para luego procesarlas y así generar las señales de control.

Tres (3) módulos controladores de motores DC y PaP (paso a paso) con chip L298N los cuales se encargan de controlar el sentido de giro y el suministro de potencia de los motores (figura 21).

Un (1) circuito NOT para suministrar las 4 señales de control de los motores paso a paso a partir de las dos señales de control provenientes del arduino (figura 22).

Dos (2) motores paso a paso bipolares de 2.3Kg-cm para generar el movimiento en los ejes X y Y (figura 23).

Cinco (5) finales de carrera para identificar la posición de inicio y evitar daños mecánicos del prototipo.

Un (1) elemento actuador final el cual posee un (1) motor DC de 12V

Una (1) fuente de alimentación de 12V y 5 Amp.

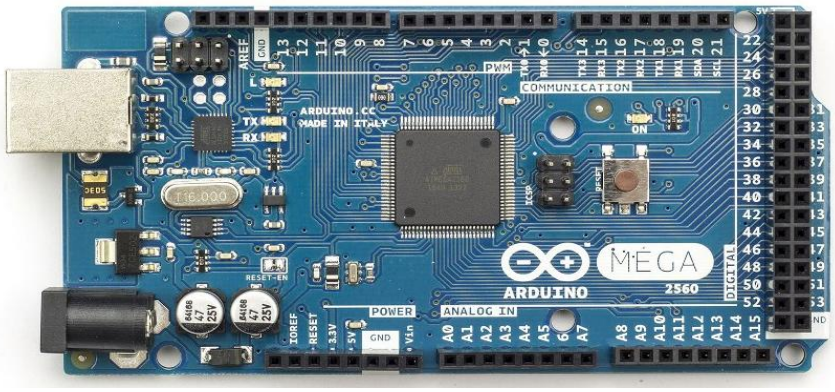
3.1.1 MICROCONTROLADOR

Como dispositivo controlador para el prototipo robot cartesiano X-Y se utilizó un Arduino Mega 2560, Este microcontrolador es desarrollado por Arduino y posee una licencia de tipo open-source, por lo tanto, existe software y hardware ya desarrollado que se puede usar libremente.

El microcontrolador Arduino Mega 2560 se compone de las siguientes partes:

- 54 pines de entradas/salidas digitales (15 pueden ser utilizados como PWM)
- 16 entradas análogas
- 4 UARTs (puertos serie)
- 1 oscilador de 16MHz
- 1 conector USB
- 1 conector de potencia

Figura 20. Micro-controlador Arduino Mega 2560.



Disponible en internet: http://arduino.cc/en/uploads/Main/ArduinoMega2560_R3_Fronte.jpg

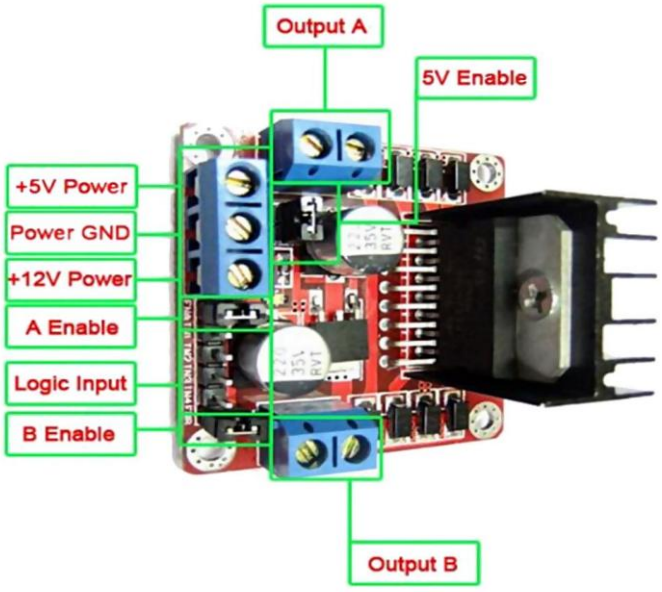
El Micro-controlador Arduino Mega 2560 recibe la señal de salida deseada desde el computador a través de un cable USB, utilizando comunicación serial, una vez recibe esta señal la procesa usando un Sketch llamado TxapuCNC_RX el cual fue cargado previamente, este se encarga de interpretar la información proveniente

del computador y con respecto a esta enviar señales de control para los motores y recibir señales de los finales de carrera, esto lo hace a través de sus puertos previamente configurados como entradas o salidas.

3.1.2 DRIVERS

Para los drivers se implementaron 3 módulos controladores de motores DC y PaP (paso a paso) con chip L298N este módulo posee muy buenas características como controlador de motores DC y motores paso a paso, entre ellas, baja pérdida de potencia por calentamiento y baja interferencia, también es posible usar su regulador integrado 78M05 para no tener que utilizar 2 fuentes de alimentación. El L298N es un controlador dual de puente completo para motores de alto voltaje y alta demanda de corriente, acepta niveles lógicos TTL.

Figura 21. Drivers L298N para motores DC y PaP.



Disponible en internet: http://www.tecnofilo.es/tienda/motores/47-l298n-motor-driver.html

3.1.3 CIRCUITO NOT

Se tuvo la necesidad de diseñar y construir un circuito con un integrado NOT debido a que el sketch de arduino fue creado para usar drivers que solo implementan dos (2) señales de control (Step y Dir) mientras que los drivers de los cuales se disponía necesitan de cuatro (4) señales de control; se realizó un análisis de la secuencia de pulsos necesaria para mover un motor paso a paso, se comprendió que las dos señales faltantes son las mismas señales (Step y Dir) pero negadas, entonces de este modo, con un circuito integrado NOT, se logró obtener las señales necesarias para poder controlar los motores con los drivers que se tenían a disposición.

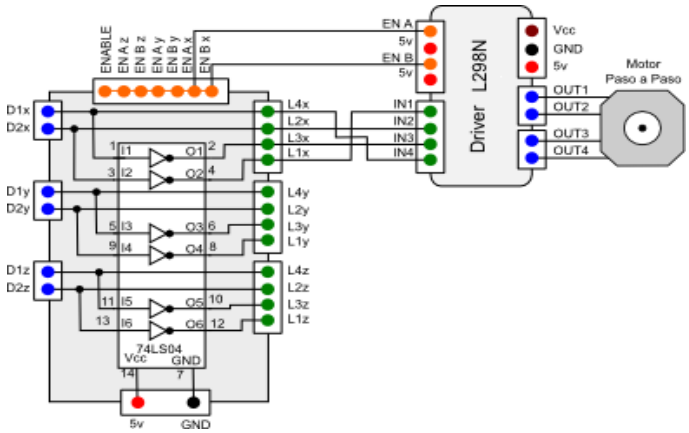
Tabla 1. Señales de control motor paso a paso.

N° DE PASOS	**1**	**2**	**3**	**4**
dev1	1	1	0	0
dev2	0	0	1	1
dev3	1	0	0	1
dev4	0	1	1	0

Tabla 2. Señales de control resumidas motor paso a paso.

N° DE PASOS	**1**	**2**	**3**	**4**
dev1=-dev2	1	1	0	0
dev3=-dev4	0	0	1	1

Figura 22. Circuito con integrado NOT.



Disponible en internet: http://txapuzas.blogspot.com/2009/12/interface-de-driver-chino-para-txapucnc.html

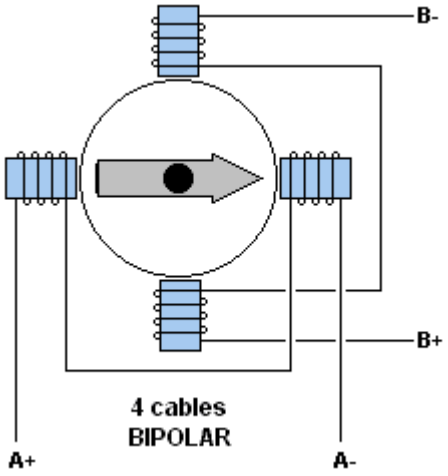
3.1.4 MOTORES PASO A PASO

Para el movimiento de los ejes se utilizaron dos (2) motores bipolares MERCURY de pasos, con referencia SM-42BYG011-25 los cuales poseen las siguientes características:

- 4 cables (2 fases).
- Paso 1.8°.
- Torque máximo 2.34Kg-cm.
- Peso 0.2Kg.
- Voltaje 12V.
- Corriente sin carga 300 mA.
- Torque Stall 2.34 Kg-cm.

Los motores paso a paso giran en un ángulo determinado en cada maniobra y se quedan parados en dicha posición hasta que se haga un cambio en la tensión de sus bobinas, esto los hace ideales para el posicionamiento preciso de mecanismos y de pequeñas masas. Los motores paso a paso bipolares están compuestos por dos (2) bobinas, el movimiento se consigue cambiando la dirección de la corriente en cada bobina.

Figura 23. Esquema motor paso a paso bipolar.

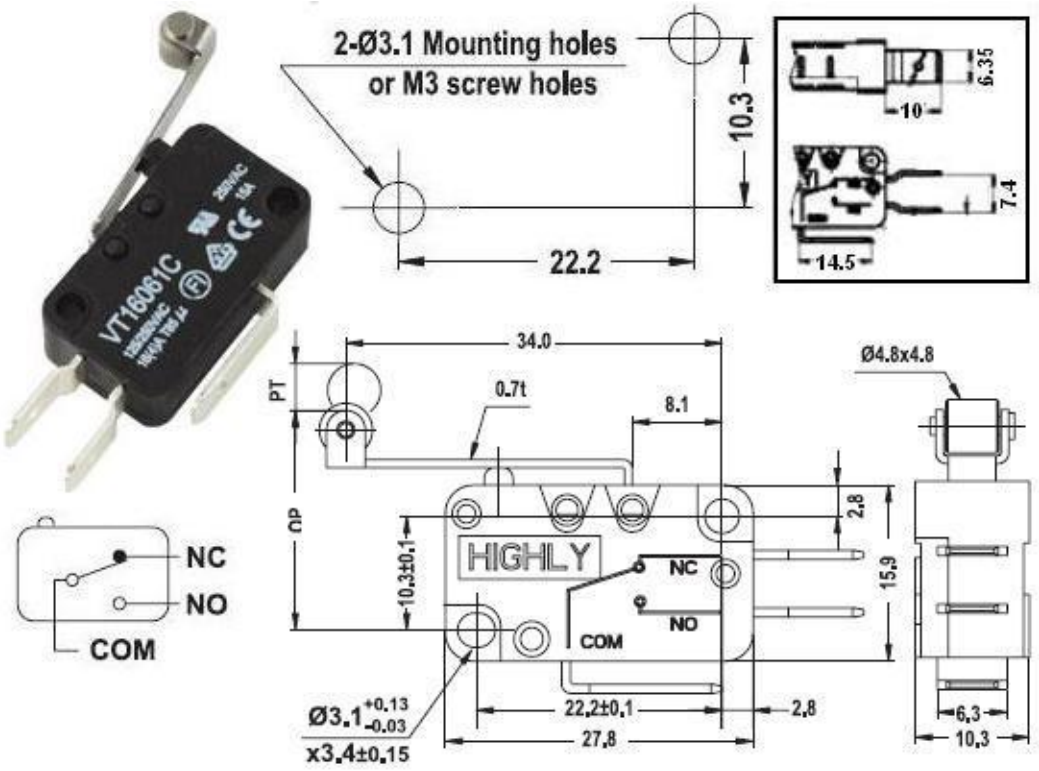


Disponible en internet: http://unelectronica.260mb.com/?p=496

3.1.5 FINALES DE CARRERA

Los finales de carrera están ubicados estratégicamente en los inicios y finales de cada eje, excepto en el eje Z donde solo se está utilizando un (1) final de carrera con esto es suficiente para verificar cuando se ha llegado a los límites; los finales de carrera que se utilizaron fueron interruptores de palanca corta y rueda de 16A, los cuales cuentan con un contacto normalmente cerrado y otro normalmente abierto, para este proyecto se utilizaron los contactos normalmente cerrados. Las señales de los finales de carrera van a las siguientes entradas del arduino: X_MIN al pin 4, X_MAX al pin 2, Y_MIN al pin 3, Y_MAX al pin 5 y Z_MIN al pin 7.

Figura 24. Interruptor final de carrera.



Disponible en internet: http://www.shoptronica.com/images/Interr.final%20de%20carrera%2016A_palanca%20larga-rueda.jpg

3.1.6 FUENTE DE ALIMENTACIÓN

La fuente de alimentación que se utilizó para este proyecto es una MLI029-1000 de 12V 5A 60W su alimentación puede ser de 100-220V, tiene una vida útil de 50.000 horas y su grado de protección es IP55, es decir, es resistente al polvo y a pequeños chorros de agua.

Figura 25. Fuente de alimentación.

Disponible en internet: http://www.microluz.es/images/P/Fuente-de-alimentacion-para-LED-12v-60W-MLI029-1000.jpg

3.2 SOFTWARE

Para el sistema de control de posicionamiento X-Y del prototipo se utilizó un software llamado TxapuCNC, es un software que permite la edición, el envío y la interpretación de códigos G (Gcode), este software consta de dos (2) partes, la primera es el TxapuCNC_TX desarrollado en gambas el cual corre en plataforma Linux y la segunda es TxapuCNC_RX un programa desarrollado en Sketch, lenguaje de programación para la plataforma arduino, este se encarga de la interpretación y envío de las señales de control hacia los motores paso a paso a través de los drivers de cada motor. Para que la máquina realice los trazados deseados por el usuario se implementó un programa editor de gráficos llamado Inkscape con el plugin Gcodetools el cual prepara y convierte trayectos desde Inkscape a Gcode usando interpolación circular.

3.2.1 CÓDIGOS G (G-CODE)

Es el lenguaje de programación más usado en control numérico (CNC) . Este forma parte de la ingeniería asistida por computadora (CAE) y es usado principalmente en automatización. En general, Gcode es un lenguaje de programación mediante el cual una persona puede decir a una máquina que ejecute instrucciones sobre a donde moverse, qué tan rápido moverse y que trayectoria seguir.

Tabla 3. Comandos G reconocidos por TxapuCNC.

COMANDO	EJEMPLO	DESCRIPCIÓN	TX	RX
G0	G0 X10	Movimiento lineal rápido	Si	Si
G1,G01	G1 X10 Y15 Z0 [F100]	Movimiento lineal controlado (Avance: 100)	Si	Si
G2,G02	G02 X60 Y30 I30 J-10 F02	Movimiento curvo (sentido horario) controlado	Si	Si
G3,G03	G03 X60 Y30 I10 J20	Movimiento curvo (antihorario) controlado	Si	Si
G4,G04	G4 P200	Pausa con retardo (Retardo: 200ms)	Si	Si
G20	G20	Definir unidades en pulgadas	Si	Si
G21	G21	Definir unidades en milímetros	Si	Si
G28	G28	Ir a origen	Si	Si
G30	G30 X10 Y20 Z30	Ir a origen a través de un punto	Si	Si
G90	G90	Definir coordenadas absolutas	Si	Si
G91	G91	Definir coordenadas relativas	Si	Si
G92	G92	Definir punto actual como origen	Si	Si
M0	M0	Paro (Pausa programada)	Si	No
M3,M03	M3	Marcha del cabezal	Si	Si
M5,M05	M5	Paro del cabezal	Si	Si

Disponible en internet: http://txapuzas.blogspot.com/2009/12/txapu-cnc-software.html

3.2.2 TXAPUCNC_TX

Este programa está desarrollado en gambas el cual se debe ejecutar en el sistema operativo Linux, la interfaz del programa TxapuCNC_TX está compuesta por un

control para mover de forma manual el robot X-Y, una consola de salida que muestra mensajes informativos sobre el estado del sistema, un editor de códigos G y un simulador de códigos G, ver figura 27. El objetivo principal de este programa es el envió de códigos G via puerto USB por comunicación serial hacia el procesador.

Figura 26. TxapuCNC_TX

Disponible en internet: https://lh4.googleusercontent.com/-Jufhjm9UH-4/TWpqXjArjOI/AAAAAAAAHgo/C-hA-98Ilzk/s1600/Layout.png

3.2.3 TXAPUCNC_RX

Este programa está desarrollado en el lenguaje de programación Sketch para la plataforma arduino. Este programa se encarga de leer los comandos que se le envían desde el TxapuCNC_TX, los interpreta y en consecuencia manda las

órdenes oportunas a los drivers para controlar los motores. Cuando el microcontrolador termina de procesar un comando de posicionamiento, TxapuCNC_RX envía un mensaje de “ok” al computador para que TxapuCNC_TX continúe y pueda enviar el siguiente comando.

3.2.4 INKSCAPE

Inkscape es un software libre editor de gráficos vectoriales, con capacidades similares a Illustrator, Freehand o CorelDraw. las características soportadas incluyen: formas, trazos, texto, marcadores, clones, mezclas de canales alfa, transformaciones, gradientes, patrones y agrupamientos. También soporta meta-datos Creative Commons, edición de nodos, capas, operaciones complejas con trazos, vectorización de gráficos, texto en trazos, alineación de textos, edición de XML directo y mucho mas, se pueden importar formatos como Postscript, JPEG, PNG, y TIFF y exporta PNG así como muchos formatos basados en vectores.

3.2.5 GCODETOOLS

Es un plug-in para Inkscape multiplataforma y está escrito en Phyton, el cual prepara y convierte trayectos desde Inkscape a Gcode, este programa posee múltiples características para múltiples finalidades a continuación se nombran las más relevantes para este proyecto:

3.2.5.1 Exportar a Gcode

- Exporta trayectos a Gcode
- Utiliza interpolación circular (mediante aproximación biarco) o lineal
- Subdivisión automática del trayecto para llegar a la tolerancia definida
- Procesamiento de múltiples herramientas
- Exportación de Gcode en forma paramétrica y de forma plana
- Personalización de encabezados y pies de página automáticos
- Selección de las unidades
- Procesamiento Multi-paso
- Sufijo numerador automático en los archivos generados para evitar la sobre escritura

3.2.5.2 Procesado de Zonas

- Generación de trayectos de zona
- Los trayectos de la zona se pueden modificar

3.2.5.3 Grabado

- Generación de la trayectoria en función de la forma de la herramienta
- Definición de diferentes formas de herramienta

3.2.5.4 Biblioteca de herramientas

- Definición de parámetros de la herramienta (diámetro, feed-avance, el paso de profundidad, avance de penetración, Gcode modificable antes y después de cada trayecto, forma de las fresas)
- Las herramientas pueden ser gestionadas mediante procedimientos estándar de Inkscape (copiar, eliminar, asignar a una capa diferente)
- Procesado para múltiples herramientas

3.2.5.5 Orientación del sistema

- Escala a lo largo de cualquiera de los ejes
- Giro en el plano X-Y
- Desplazamiento a lo largo de cualquiera de los ejes
- Transformación de acuerdo a puntos arbitrarios

3.2.5.6 Post-procesador

Puede crear post-procesadores escribiendo los comandos o seleccionar alguno definido de la lista de post-procesadores por defecto

- Escalado y desplazamiento del Gcode
- Comandos de reasignación Gcode
- Parametrización Gcode
- Redondeo de los valores de coma flotante a la precisión especificada

CAPÍTULO 4. IMPLEMENTACIÓN DE LA INTERFAZ GRÁFICA

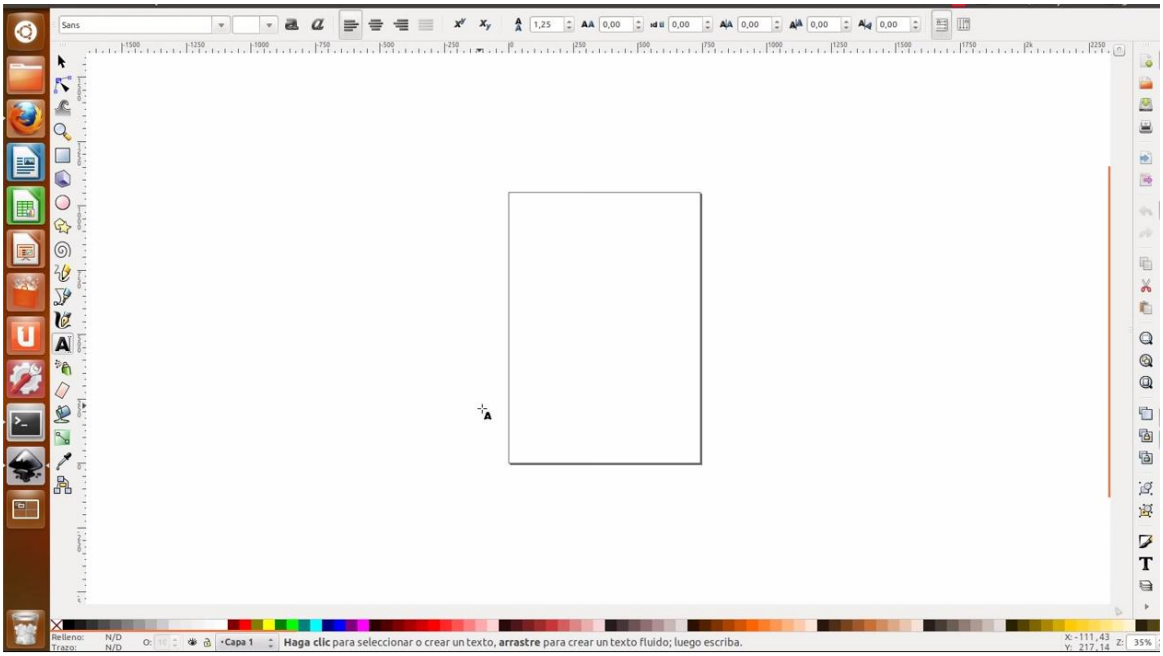
La interfaz gráfica de este proyecto consta de dos partes una es Inkscape el cual permite al usuario crear un diseño digital de los trazados que posteriormente se realizarán con el prototipo y la segunda es TxapuCNC_TX el cual se implementó con el fin de proporcionar al usuario un entorno visual sencillo que le permitiera la comunicación con el sistema de control del prototipo. Ambos programas corren bajo plataforma Linux, por este motivo fue necesaria la utilización de un computador al cual se le instaló el sistema operativo Ubuntu 13.10 x32 bits.

4.1 IMPLEMENTACIÓN INKSCAPE

Inkscape es un software de edición gráfica el cual por medio de un plug-in llamado Gcodetools permite traducir los diseños gráficos a “códigos G” los cuales son necesarios para realizar el control del prototipo. Ya que la necesidad que demanda este proyecto a nivel de diseño es algo muy básico debido a que se trata de trazos en 2D la implementación se hace bastante sencilla.

Al comenzar a abrir el Inkscape se podrá apreciar el inicio del programa tal como se ve en la figura 27.

Figura 27. Inicio Inkscape.



En este caso se realizará la transformación del contorno de una letra del alfabeto a “códigos G” para esto se utilizó la herramienta “crear y editar objetos de texto” tal como se aprecia en la figura 28.

Figura 28. Crear y editar objetos de texto.

En la figura anterior se puede apreciar que el objeto se orienta en la parte inferior izquierda del lienzo esto se debe a que esta posición coincide con la posición de inicio de el prototipo CNC lo cual hace que el trazado sea más eficaz en cuestión de tiempo.

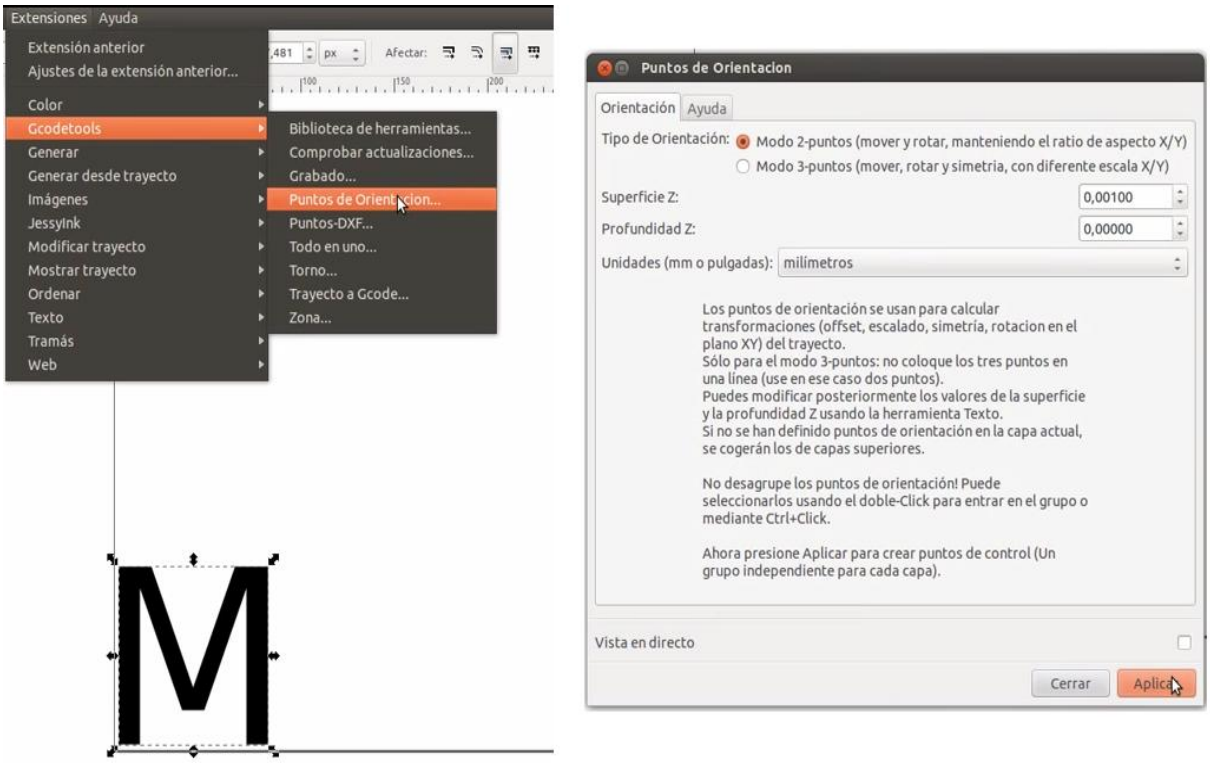
Ahora se debe convertir la letra la cual es un objeto tipo texto a un trayecto con el fin de hacerla reconocible para el Gcodetools esto se realiza por medio de una herramienta llamada “Objeto a trayecto” y posteriormente se debe convertir a un desvío dinámico con la herramienta “desvío dinámico” ambas ubicadas en el menú principal de Inkscape en la pestaña con el nombre de “Trayecto” tal como se aprecia en la figura 29.

Figura 29. Trayecto.

En la figura anterior se aprecia que ambas herramientas tienen accesos rápidos.

A continuación se debe dar al diseño los puntos de orientación los cuales son proporcionados por el Plug-in Gcodetools el cual se encuentra en la pestaña de extensiones en el menú principal una vez allí se le da click en aplicar tal como se aprecia en la figura 30.

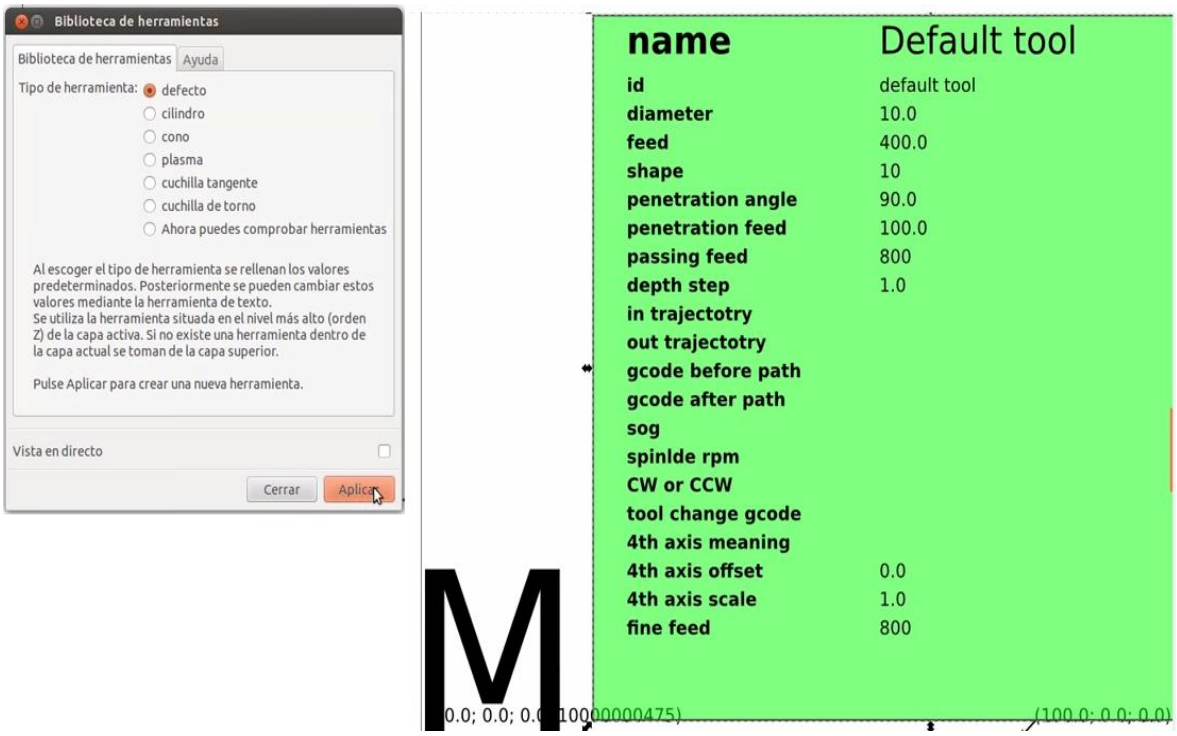
Figura 30. Puntos de orientación.



Nótese que la pestaña de “Superficie Z” tiene un valor asignado de 0.001 el cual por defecto viene en 1 esto con el fin de evitar posibles errores intrínsecos al código de recepción de datos.

Una vez el diseño está orientado se retorna a la pestaña de “Extensiones” - “Gcodetools” - “biblioteca de herramientas" allí se pueden observar varias opciones para distintas herramientas que se podrían implementar, para este caso se selecciona la herramienta por defecto (default) (figura 31).

Figura 31. Biblioteca de herramientas.



Una vez seleccionado el tipo de herramienta aparecerá un cuadro verde con las propiedades de la herramienta las cuales se pueden editar con la herramienta de “crear y editar objetos de texto” conforme a las necesidades que se tengan.

Una vez se hayan realizado todos los pasos anteriores lo siguiente es generar el Gcode del diseño, para ello se debe ingresar a la pestaña “Extensiones”- “Gcodetools”- “Trayecto a Gcode” -”Preferencias” allí se selecciona el directorio de destino en el cual se desea que quede guardado el archivo y el nombre que se le asigne a dicho archivo (figura 32).

Figura 32. Trayecto a Gcode.

Es importante que al momento de dar click en aplicar el usuario se encuentre situado en la pestaña “Trayecto a Gcode” para evitar un error generado por el Inkscape.

4.2 IMPLEMENTACIÓN TXAPUCNC_TX

TxapuCNC_TX es un programa utilizado para el control y monitoreo del prototipo de robot cartesiano X-Y, este programa está desarrollado originalmente para el control de una fresadora casera, pero teniendo en cuenta que basicamente el funcionamiento es el mismo, fue posible utilizarlo para este proyecto. La versión que se está usando de este programa para este proyecto es TxapuCNC_TX10c 1.0.1 el cual está desarrollado en Gambas3.

TxapuCNC_TX establece una comunicación serial con el controlador arduino a través del puerto USB del computador, para que esto funcione sin ningún inconveniente es necesario tener instalado un componente adicional del Gambas3 llamado gambas3-gb-net.

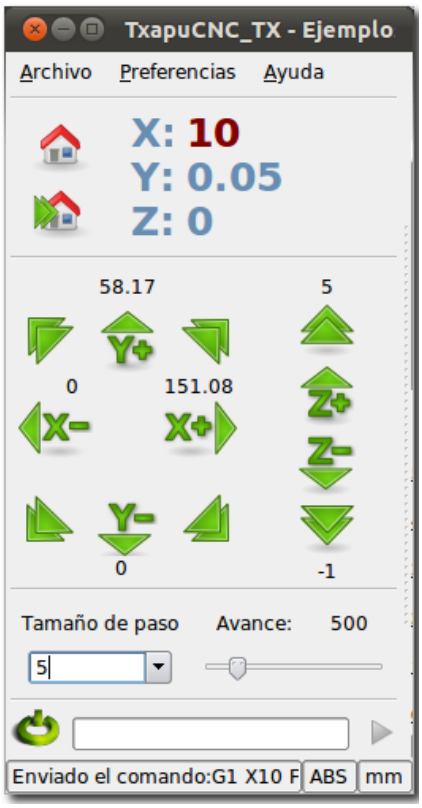
Excluyendo el menú principal, se puede dividir TxapuCNC_TX en cuatro componentes:

- Un control para mover directamente el prototipo.
- Una consola de salida en la cual se pueden visualizar mensajes informativos sobre el estado del sistema.
- Un editor de códigos G.
- Un simulador de códigos G.

4.2.1 VENTANA DE CONTROL

Desde esta ventana se pueden enviar comandos independientes al controlador (Arduino). Algunos de estos comandos disponen de un botón propio.

Figura 33. Ventana de control.

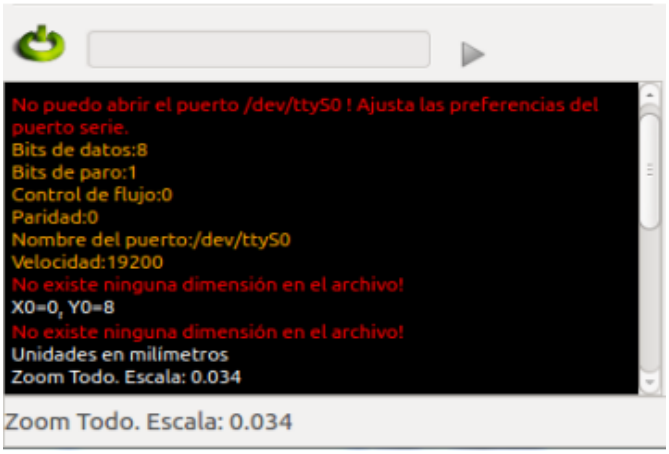


Los indicadores de posición situados en la parte superior, no indican la posición real mientras el prototipo se está moviendo, solamente cuando ha llegado a la posición de destino. Mientras se mueve el prototipo, se muestran los puntos de destino, y éstos parpadean indicando que se está moviendo. Al ser el controlador (Arduino) el encargado de mover los motores, TxapuCNC_TX no sabe la posición actual del prototipo, solamente cuando recibe la orden “ok” TxapuCNC_TX conoce que el prototipo ha llegado a la posición especificada.

4.2.2 CONSOLA DE SALIDA

En la consola de salida se pueden observar los comandos que se estén ejecutando en el momento, así como información de la comunicación serial y también se pueden visualizar mensajes de error, los cuales se muestran en color rojo.

Figura 34. Consola de salida.



4.2.3 EDITOR DE CÓDIGOS G

El editor es configurable desde el menú/Preferencias/Editor. Tiene una barra de herramientas con los comandos más usuales en este tipo de programas que permite realizar cualquier cambio deseado antes de la ejecución de los códigos. Hay que destacar los últimos controles, que son los específicos para la ejecución de los códigos G en el prototipo.

Figura 35. Editor códigos G.

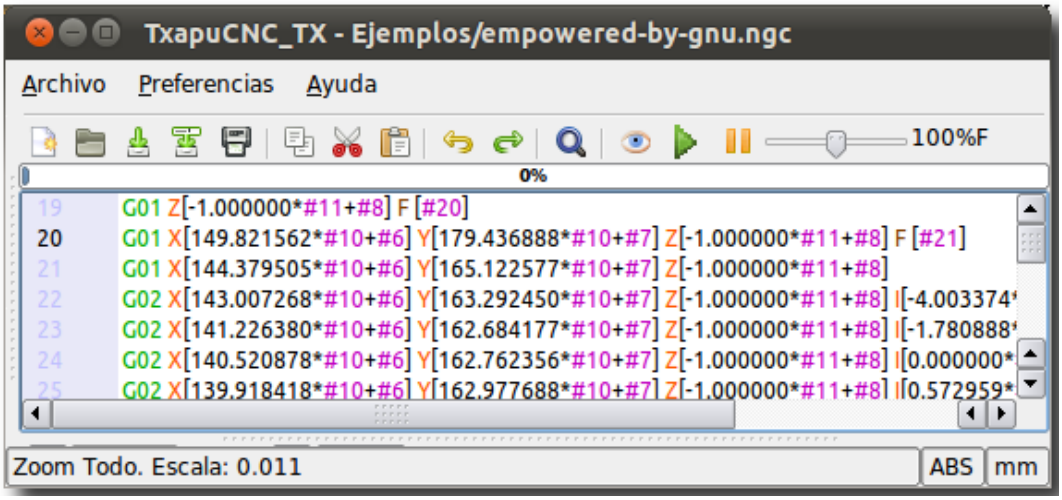


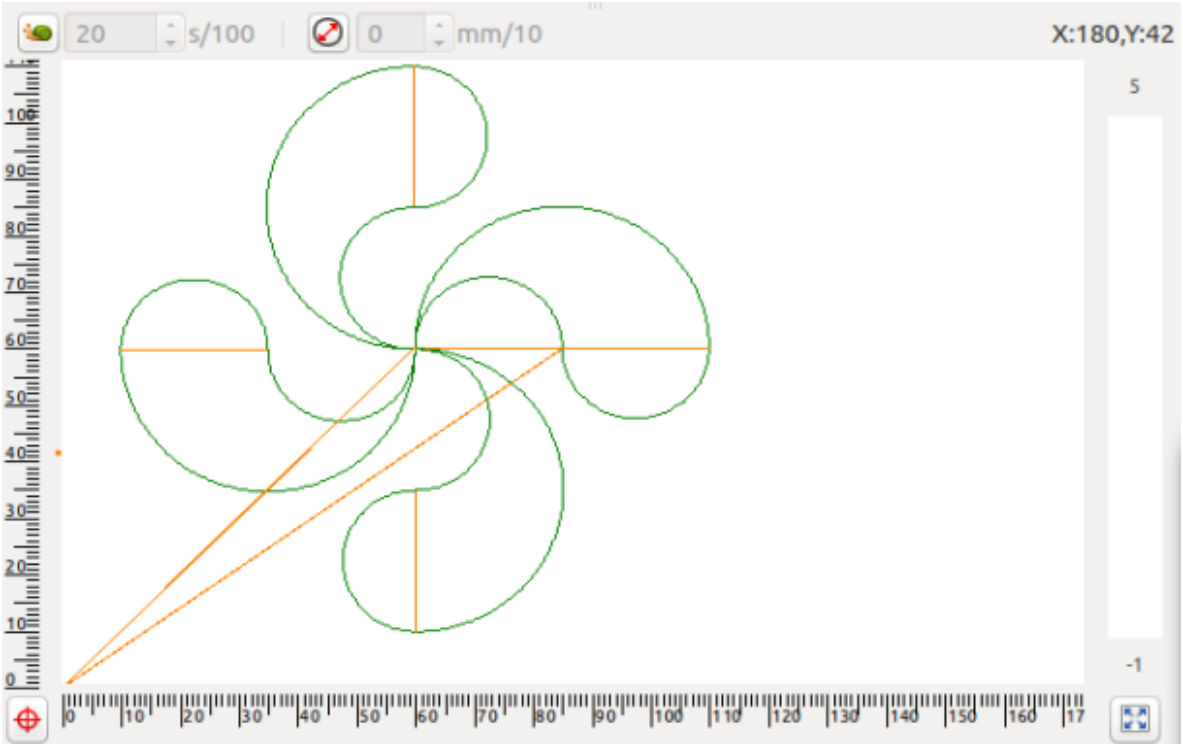
Tabla 4. Botones ejecución de códigos G.

BOTÓN	DESCRIPCIÓN
	Visualiza en el simulador el programa cargado en el editor. Este proceso se ejecuta automáticamente al cargar un programa G-Code.
	Ejecuta el programa en el prototipo , es decir, envía los comandos al Arduino y simultáneamente lo visualiza en el simulador.
	El pulsador de parada, detiene tanto la simulación como el envío de datos al controlador Arduino.

4.2.4 SIMULADOR DE CÓDIGOS G

Esta ventana muestra una visualización del archivo cargado en el editor, una vez se cargue el archivo la simulación se ejecutará automáticamente, para ejecutarla nuevamente es necesario apretar el botón de simulación situado en la barra de herramientas del editor. La simulación puede ser instantánea o pausada para ver la ejecución de cada instrucción.

Figura 36. Simulador códigos G.



En el modo de pausa se activa al apretar el botón con la imagen de un caracol, al activarse se puede modificar el tiempo de espera entre la visualización de cada paso expresado en centésimas de segundo.

Si se quiere que la simulación ocupe toda la ventana se puede apretar el botón situado en la parte inferior derecha de la pantalla. En la parte inferior izquierda del editor existe otro botón que coloca la simulación en el origen (no modifica la escala de dibujo).

En la parte derecha de la ventana del simulador vemos una representación de la herramienta del actuador final y su posición en el eje Z. Los números situados encima y debajo de la imagen, son los valores máximos y mínimos del eje Z, encontrados en el archivo cargado en el editor.

CAPÍTULO 5. PRUEBAS DE FUNCIONAMIENTO DEL PROTOTIPO

Para las pruebas de funcionamiento se decidió examinar el prototipo con tres tipos de trazos diferentes que implementarán rectas diagonales y curvas para observar y evaluar el comportamiento del prototipo. A continuación se mostrara la ficha técnica del prototipo.

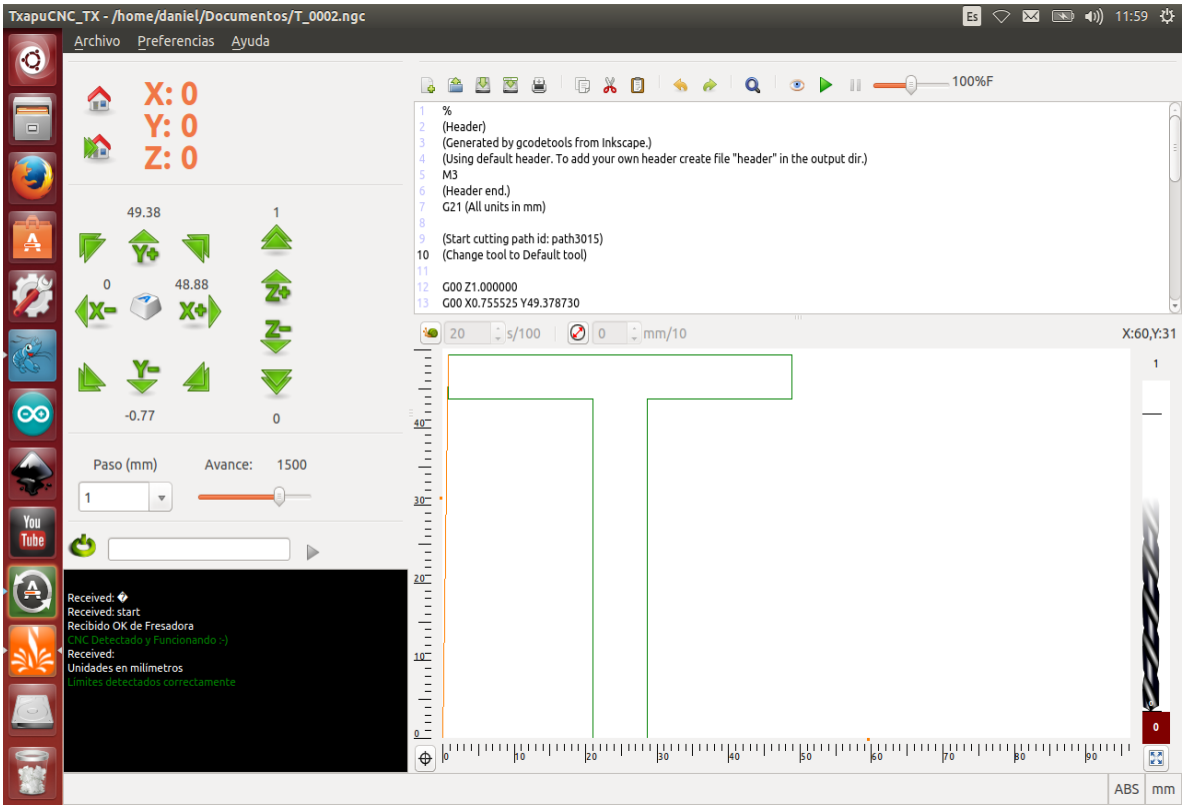
Tabla 5. Ficha técnica prototipo.

ÁREA ÚTIL DE TRAZADO	30mm x 30mm
TAMAÑO DE LA MAQUINA	800mm x 655mm
POTENCIA EN EJE X y Y	Motor paso a paso de 2.3Kg x cm a 0.33A
MATERIAL DE FABRICACIÓN	Acero AISI 1045, Polimero ABS plus, Madera MDF, Neopreno.
RESOLUCIÓN EN POSICIONAMIENTO	Aproximadamente 1mm
RANGO DE VELOCIDAD	38,3 mmxseg aproximadamente.
SISTEMA DE GUÍA X y Y	Guías lineales en acero AISI 1045 con rodamientos lineales KH1228.
SISTEMA DE ACCIONAMIENTO	Sistema de transmision engranaje-correa-tensor.
CONTROLES DE USUARIO	Software de control TX_apuCNC basado en PC con interfaz USB.
DRIVERS MOTOR	Drivers con circuito integrado L298n.
CÓDIGO	Gcode.
ENTRADA DE ENERGÍA	110V 60Hz
REQUISITOS DEL COMPUTADOR DE MANDO	Sistema operativo Linux Procesador de 1GB de RAM y 30GB de disco duro.
PESO	Aproximadamente 23Kg.

5.1 PRUEBA DE FUNCIONAMIENTO CON LÍNEAS RECTAS

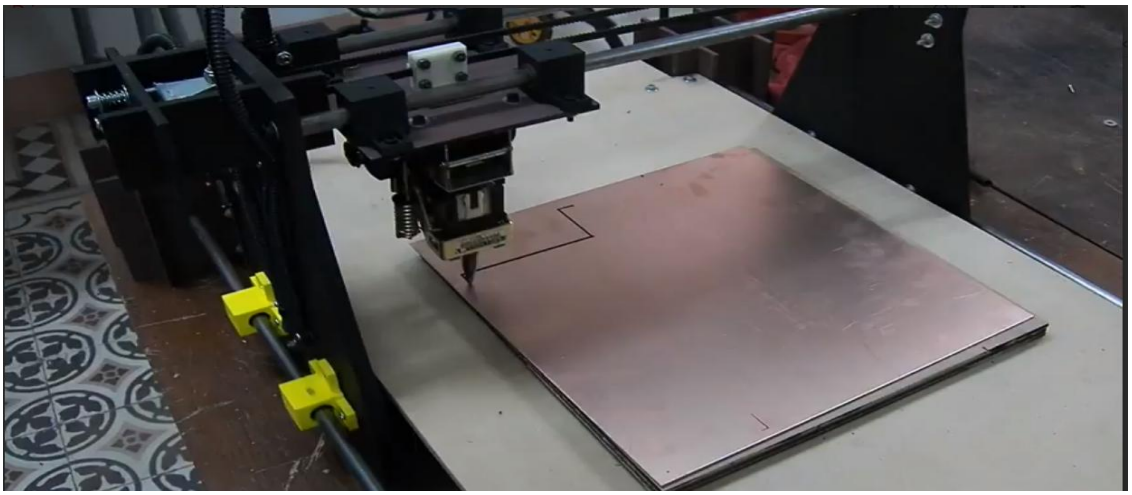
La primera prueba se realizó haciendo el trazo de una letra “T” la cual está compuesta solo por líneas rectas tal como se aprecia en la figura 37.

Figura 37. Vista previa letra “T”.



En la imagen anterior se puede apreciar la interfaz gráfica TxapuCNC_TX la cual brinda una simulación previa del trazo que posteriormente será realizado por el prototipo (figura 38).

Figura 38. Trazado letra “T” mediante prototipo.



En la figura 38 se puede apreciar que el trazo se realiza sobre una baquelita de cobre, aunque no es el material para el cual se planteó originalmente el desarrollo del prototipo, este permite dar una buena aproximación de cómo se comportará con las láminas de acero y a su vez adicionalmente muestra su versatilidad para ser implementado en otras aplicaciones como por ejemplo el trazado de pistas para los PCB.

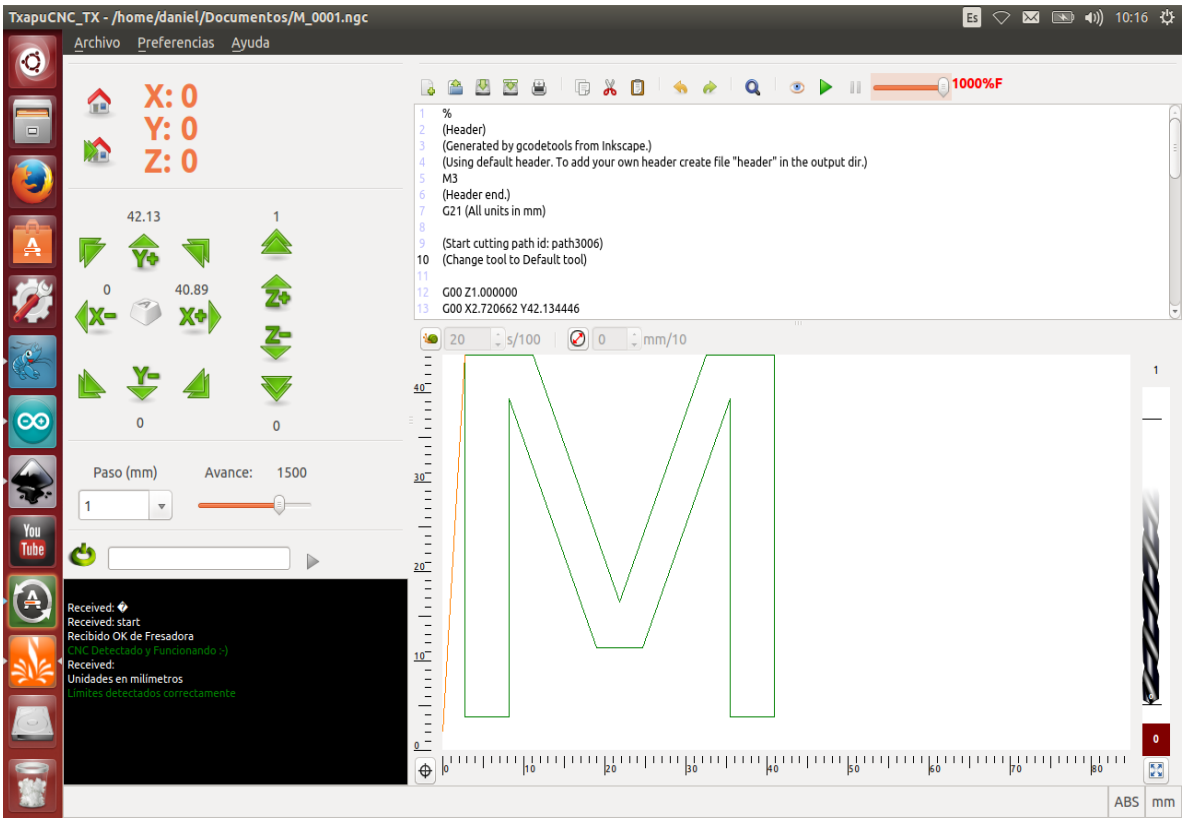
Figura 39. Resultado obtenido trazo de rectas.

Como se observa en la figura anterior para trazos compuestos sólo por líneas rectas el prototipo responde de una manera muy eficaz brindando al usuario un muy buen resultado con respecto al diseño digital.

5.2 PRUEBA DE FUNCIONAMIENTO LÍNEAS DIAGONALES

Para la prueba con líneas diagonales se procedió a realizar el trazo de una letra “M” (figura 40) la cual está compuesta por dos secciones rectas y dos diagonales y se analizó la respuesta del prototipo.

Figura 40. Vista previa letra “M”.



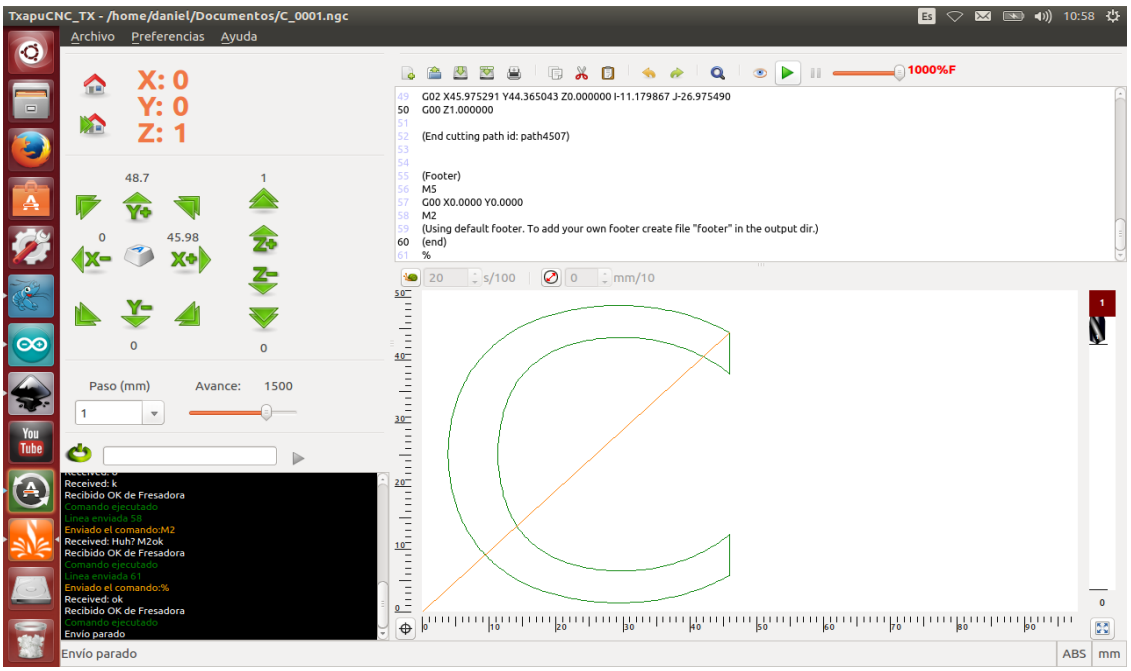
En la figura 41 se puede ver como el trazo realizado por el prototipo difiere ligeramente con el diseño digital esto debido a que las diagonales son trazos más complejos que las rectas puesto que su realización se lleva a cabo alternando rápidamente el movimiento de los motores del eje X y Y siendo así más vulnerable a posibles errores tanto de la parte electrónica como en la mecánica, aún así el resultado es aceptable para la implementación deseada.

Figura 41. Resultado obtenido trazo de diagonal.

5.3 PRUEBA DE FUNCIONAMIENTO LÍNEAS CURVAS

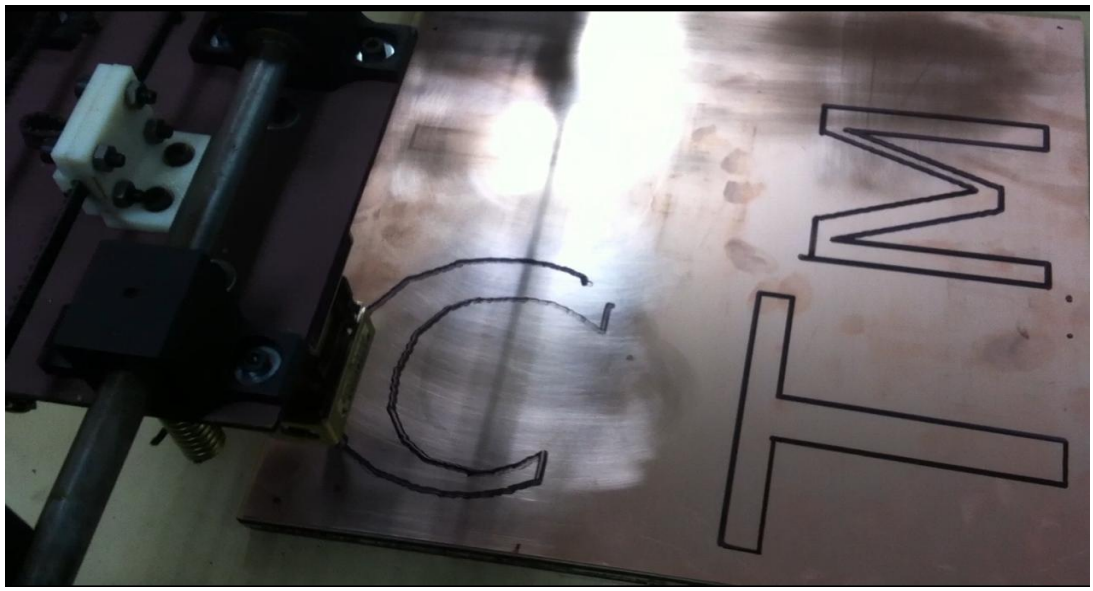
Para evaluar el comportamiento del prototipo en cuanto al trazo de curvas se realizó el diseño de la letra “C” mediante el inkscape tal como se ve en la figura 42.

Figura 42. Vista previa letra “C”.



Como se puede apreciar en la figura 43 el resultado obtenido en comparación con el diseño digital difieren bastante en cuanto a la precisión en el trazo, esto se debe a que con el sistema de desplazamiento que se selecciono utilizando correas dentadas no se cumplieron las expectativas que se tenían, el sistema desplazamiento con correas dentadas hace que el prototipo sea más veloz, pero sacrifica un poco la precisión en el trazado debido a que la correa tiene cierta elasticidad lo cual genera movimientos indeseados aun después que los motores han parado totalmente, estos pequeños movimientos se convierten en una traza muy diferente a la esperada.

Figura 43. Resultado obtenido trazo de curva.



CAPÍTULO 6. CONCLUSIONES Y RECOMENDACIONES

- Para implementaciones que requieran de alta precisión de trazado es mucho más recomendable usar sistemas de posicionamiento lineal accionados por tornillos en vez de por correas dentadas.

- Se logro construir un prototipo de robot industrial cartesiano X,Y para el trazado sobre laminas de metal que ofrece una buena funcionalidad y es adecuado para ser exhibido.

- Existen distintos programas para control CNC mucho más profesionales y con más herramientas como el Mach3, que a pesar de no ser un software gratuito, brinda una versión de prueba que permite códigos G hasta con una extensión de 500 líneas además ahorraría tener que desarrollar un código intérprete de códigos G ya que este lo hace mediante el mismo ordenador donde corre el software de interfaz de usuario.

- Es muy factible llegar a las pequeñas y medianas empresas con máquinas a precios razonables que les permitan realizar sus procesos de manera más eficaz.

- La implementación de este prototipo a escala real permitiría reducir significativamente los tiempos de la etapa de trazado de la empresa Industrias CM y reasignar a los trabajadores en otras etapas de la cadena de producción aumentando así su capacidad de oferta.

- Sí se requiere realizar piezas mecánicas para el prototipo, puesto que estas no se encuentran comercialmente, es muy importante disponer de herramientas que brinden una alta precisión de maquinado.

- Antes de realizar cualquier montaje o maquinar piezas para el prototipo es muy importante desarrollar un ensamblaje virtual de la máquina, mediante un software CAD, que permita analizar y observar posibles errores.

- La implementación ideal de este prototipo es el trazado de trayectorias en línea recta, pues lo hace con un alto grado de exactitud y a una gran velocidad.

ANEXO A. GLOSARIO

- **Actuador:** Dispositivo capaz de transformar energía eléctrica en la activación de un proceso con la finalidad de generar un efecto sobre un proceso automatizado [5].

- **Arduino:** Plataforma de hardware libre, basada en una placa con un microcontrolador y un entorno de desarrollo, diseñada para facilitar el uso de la electrónica en proyectos multidisciplinares [2].

- **CAD:** Diseño asistido por computadora CAD (Computer Aided Design) [9].

- **CAE:** Ingeniería asistida por computadora CAE (Computer Aided Engineering) [9].

- **Chumacera:** Pieza con una muesca en la cual descansa y gira cualquier eje de maquinaria [11].

- **CNC:** Control numérico computarizado (computer numerical control) [5].

- **Comunicación serial:** Protocolo común para la comunicación entre dispositivos incluido de manera estándar en cualquier computadora [13].

- **Controlador:** Tiene como objetivo elaborar la señal de control que permita que la variable controlada corresponda a la señal deseada [7].

- **GAMBAS:** Lenguaje de programación libre derivado de Basic [12].

- **Gráficos Vectoriales:** Imagen digital formada por objetos geométricos independientes como segmentos polígonos arcos etc. cada uno de ellos definidos por distintos atributos matemáticos [6].

- **HMI:** Interfaz de usuario (Human Machine Interface) [3].

- **Interpolación lineal y circular:** Forma en que se describen las trayectorias rectas y curvas en las operaciones de mecanizado [6].

- **L298N:** Controlador dual de puente completo para motores de alto voltaje y alta demanda de corriente, diseñado para aceptar niveles lógicos TTL [10].

- **Maquinar:** El maquinado es un proceso que se basa en remover por medio de una herramienta de corte el exceso de material de manera tal que la pieza terminada sea la deseada [11].

- **MDF:** Tablero de fibra de densidad media (medium density fibreboard) [10].

- **Niveles lógicos TTL:** Hace referencia a los niveles establecidos para esta familia lógica los cuales definen el rango de tensión para los estados de bajo comprendido entre 0.0V - 0.08V y alto entre 2.4V-Vcc [10].

- **Prototipo:** En este caso hace referencia a cualquier tipo de máquina en pruebas el cual permite testear el objeto antes de que entre en producción [11].

- **Robot cartesiano:** Robot industrial cuyos ejes principales de control son lineales [3].

- **Sistema mecánico:** Sistema constituido por elementos que tienen como función específica transformar o transmitir el movimiento desde las fuentes que lo generan [5].

- **Sistema de control:** Tipo de sistema dedicado a obtener la salida deseada de un sistema o proceso [7].

- **Sketch:** Es el Lenguaje de programación utilizado por la plataforma arduino [8].

- **Tensor:** Es el elemento encargado de la distribución de tensiones y esfuerzos internos producidos por el sistema piñón-correa del prototipo [11].

- **Termo-polímero ABS Plus:** Es el tipo de plástico que utiliza la prototipadora 3D que se implementó en el desarrollo del prototipo, que a temperaturas altas se vuelve deformable, se derrite cuando se calienta y se endurece en un estado de transición vítrea cuando se enfría lo suficiente.

- **Trazado:** Hace referencia al recorrido, dirección de un camino, canal o trayectoria [4].

BIBLIOGRAFÍA

[1] GUIZZO, E. The Rise of the Machines. Spectrum, IEEE. Volume: 45 Issue: 12 p.88. Dec. 2008.

[2] http://arduino.cc/en/

[3] RUIZ de GARIBAY PASCUAL, Jonathan. Robótica: Estado del arte. p. 11. Universidad de Deusto. Sep. 2007

[4] http://www.wordreference.com/definicion/trazado

[5] BARRIENTOS, Antonio, PEÑIN, Luis Felipe, BALAGUER, Carlos y ARACIL Rafael. Fundamentos de Robótica, Madrid: McGRAW HILL, 1997. 327 p.

[6] http://wwwprof.uniandes.edu.co/~gprieto/classes/compufis/interpolacion.pdf

[7] OGATA, Katsuhiko. Ingeniería de control moderna. Pearson Educación. 2003. 965 p.

[8] http://arduino.cc/en/Tutorial/Sketch

[9] GROOVER. CAD/CAM: Computer-Aided Design and Manufacturing. Pearson Education. 2006. 511 p.

[10] http://hispavila.com/3ds/atmega/hpuente.html

[11] MOTT, Robert L. SALDAÑA, Sergio. Diseño de elementos de máquinas. Pearson Educación. 2006. 872 p.

[12] http://gambas.sourceforge.net/en/main.html

[13] TORRENTE, Oscar. Arduino : curso práctico de formación. C Libros. 2013. 588 p