Taladro Robótico

Corporación Universitaria Minuto de Dios

Tecnología Electrónica.

Juan Steven Osorio Rodríguez.
Jonathan Jesús Botia Fandiño.
Bogotá D.C. 2012

RESUMEN

Un taladro robótico, es un mecanismo capaz de moverse sobre los 3 ejes del plano en tres dimensiones (X, Y, Z), de tal forma que puede realizar acciones en el plano X, Y.

Dependiendo del posicionamiento del sistema del eje Z, se pueden realizar diferentes acciones, sin perder la precisión con la repetición de las funciones.

Se realiza con el fin de hacer un tanto más fácil y óptimo el desarrollo de circuitos impresos para estudiantes de electrónica. Se toma como base, principalmente que, para diseñar y desarrollar un circuito impreso se debe seguir una serie de pasos. Teniendo en cuenta lo anterior, por medio de investigación, se ha llegado a la deducción de que los dos últimos pasos del proceso son los que más toman tiempo, los cuales son, la perforación de la baquela y el soldado de los componentes electrónicos. Una vez analizado y entendido el problema, se tiene que en un circuito impreso, si se tratan de perforar más de 50 orificios, se va perdiendo la precisión y el proceso se vuelve tedioso y molesto. Por el contrario, si se elabora utilizando un sistema de control, con un microcontrolador pic 18F4550, por ejemplo, se desarrolla un trabajo óptimo y más preciso.

Para poder lograr esto y optimizar este proceso, se diseñara una interfaz con dos de los principales programas que nos ayudan a la simulación y diseño de circuitos impresos, estos principales programas son, Proteus y Livewire- PCB Wizard. Estos básicos programas nos ayudaran al desarrollo de nuestros circuitos impresos y aprovechando sus plataformas, Ares y PCB, las cuales nos generan un “Autoruting” del circuito, el cual nos arroja en un plano y con bastante precisión, los puntos y coordenadas en las cuales se van a ubicar cada una de las perforaciones de nuestro circuito.

Se ha investigado bastante sobre pic’s comerciales que sean capaces de brindarnos diferentes tipos de conexión para poder enviar los datos desde el programa hacia los motores de salida y así poder tener un control preciso de los motores.

Se inicio con el pensamiento de realizar una comunicación pc-pic por medio de puerto paralelo. Pero nos encontramos que en este tiempo es difícil conseguir un pc de buen rendimiento con este puerto. Ya que es el medio más viable para la precisión de nuestro trabajo que en este caso es el tema esencial y base del éxito del proyecto; Sin embargo realizamos las investigaciones necesarias pasando desde el libro mas antiguo hasta la pagina web mas actualizada con el objetivo de obtener una comunicación por medio de puerto USB.

Después de una larga búsqueda de como realizar la comunicación, encontramos una combinación entre algunos programas.

Visual Basic. Es uno de los tantos lenguajes que existen con el fin de programar aplicaciones o dispositivos compatibles con Windows. Visual no es un programa de lenguaje principiante, a pesar de lo fácil que resulta utilizarlo es un programa que lo puede hacer hasta los desarrolladores profesionales de aplicaciones.

EasyHID. Es una aplicación de microcode Studio que te permite generar código Fundamental para facilitar la conexión PICusb-Visual Basic.

Proton. Es el compilador que nos permite valga la redundancia compilar el código generado en EasyHID

Por medio de un procedimiento explicado en el cuerpo del trabajo (parte de programación) logramos que el funcionamiento de los motores sea controlado desde un programa sencillo creado en formato de visual basic, cuyo fin es que al combinar los movimientos de nuestros motores, el taladro quede en una posición exacta, definida por la necesidad que tenemos de perforar nuestro circuito de una manera menos agotadora y mas precisa.

El funcionamiento de nuestro taladro robótico en palabras comunes se define en tres movimientos.

Movimiento en la base del proyecto: La base tiene un movimiento sobre el eje z que es quien permite que el circuito se mueva hacia adelante o hacia atrás dándole un grado de alcance al taladro de 100% sobre el circuito

Movimiento de la base del taladro: Esta base tiene un movimiento sobre el eje x es quien permite que el taladro se mueva hacia la izquierda o derecha cubriendo cualquier lugar sobre el circuito

Movimiento de la columna: Este movimiento es el principal de nuestro proyecto ya que es el que permite acercar nuestro taladro al circuito y dar inicio a la perforación del mismo.

Con esto concluye el trabajo del taladro robótico y por resultado tenemos un circuito perforado de una forma profesional he impecable listo para el proceso de soldadura.

ABSTRACT

A robotic drill, is a mechanism capable of moving on the 3 axes of the plane in three dimensions (X, Y, Z), so that you can perform actions on the plane X, Y.

Depending on the positioning of the Z axis system, different actions can be performed without losing the precision with repeating functions.

It is performed in order to make easier and therefore optimum development of printed circuits for electronic students. It is taken as a basis, namely that, to design and develop a printed circuit must follow a series of steps. Given the above, by means of research, it has come to the inference that the two last steps of the process are the most time consuming, which are the baquela drilling and welding of electronic components. Once analyzed and understood the problem, you must be in a printed circuit board, if you try to drill more than 50 holes, accuracy is lost and the process becomes tedious and annoying. Conversely, if it is made using a control system with a microcontroller pic 18F4550, for example, develops a more precise optimal work.

To accomplish this and optimize this process, we will design an interface with two major programs that help us simulation and PCB design, these major programs are, Proteus and Livewire-PCB Wizard. These basic programs help us to develop our circuit boards and leveraging their platforms, Ares and PCBs, which we generate an "Autoruting" circuit, which throws us into a flat and quite accurately, points and coordinates in which are to locate each of the perforations of the circuit.

Insufficiently investigated on pic's business to be able to give us different types of connection to send the data from the program to the engine power output and thus precise control of motors.

It started with the thought of making a communication pc-pic via parallel port. But we find that at this time it is difficult to get a good performance PC with this port. Since it is the most feasible for the accuracy of our work here is the essential theme and base the success of the project, but make the necessary

inquiries from the book from oldest to most current web page in order to obtain a communication via USB port.

After a long search to realize communication, we find a combination of some programs.

Visual Basic. It is one of many languages that exist to develop applications or devices that support Windows. Visual not a beginner language program, despite how easy it is to use a program that can make up professional application developers.

EasyHID. It is a microcode Studio application that lets you generate code for easy connection Fundamental PICusb-Visual Basic.

Proton. It allows the compiler to compile pun EasyHID generated code

Through a procedure explained in the body of work (programming part) we get the engine operation is controlled from a simple program created in Visual Basic format, which aims at combining the movements of our motors, drill be in an exact position, defined by the need for us to drill our circuit in a less stressful and more accurate.

How our robotic drill common words defined in three movements.

Movement at the base of the project: The base has a motion on the z axis is who allows the circuit to move forward or backward giving a degree of scope to drill 100% on the circuit

Movement of the hole bottom this base has a movement about the axis x is the one who allows the drill to move left or right covering anywhere on the circuit

Spinal movement: This movement is the principal of our project because it is what allows us to bring the circuit drill and begin drilling the same.

This concludes the work of the robotic drill and drilled a result we have a circuit in a professional manner have impeccable ready for welding.

CONTENIDO

LISTA DE GRÁFICOS Y TABLAS

INTRODUCCIÓN

Desde el punto de vista de un estudiante de electrónica, nos hemos dado cuenta de que hay un gran problema a la hora de desarrollar circuitos impresos. El problema es que se vuelve un trabajo tedioso, aburrido e incluso muchas veces complicado el solo hecho de abrir los agujeros de las baquelas.

Muchas personas no cuentan en su casa con un motor-tool, y es un trabajo extremadamente complicado tratar de abrir los agujeros de una baquela con un taladro de percusión por ejemplo, este problema lleva a romper muchas brocas y esto, al mismo tiempo se vuelve costoso, sin mencionar que el trabajo queda con muy mala presentación.

Viendo este problema, se pensó en la solución de automatizar este proceso, para esto nos apoyamos en la robótica. Diseñando una serie de rieles y partes mecánicas, nos dimos cuenta de que podíamos lograr el manejo y control de un motor-tool por medio de motores paso a paso.

Teniendo el problema y teniendo una posible solución, empezamos el proceso de investigación, comenzando por plasmar ideas del posible diseño y la parte mecánica del robot, seguido a esto, continuamos con la investigación de la circuitería, como lo es el pic, los motores, y demás componentes electrónicos secundarios y finalmente terminamos con la investigación del diseño de software de control.

Seguido de la investigación, se crearon mediciones a escala real y luego se pasó al proceso de formación del prototipo. Una vez realizadas las medidas y realizada la primera parte del prototipo, se empezaron a plasmar las medidas sobre el material con el cual se iban a diseñar las bases del robot, seguido a esto se realizó el montaje de las partes mecánicas que se iban a utilizar para el correcto funcionamiento del mismo.

En pequeños espacios de tiempo, en los que por alguna dificultad no se podía continuar con el desarrollo del robot, se investigaba y se iba desarrollando paso a paso el software de control que manejaría desde el computador nuestros sistemas mecánicos.

Una vez finalizado el software desde el computador, se dejaron abiertas las posibles opciones de usar conexión por puerto paralelo o por puerto serial USB, puesto que en primera medida se había pensado en desarrollar la comunicación entre el computador y el microcontrolador por medio del puerto USB, pero tomando como consecuencia la dificultad y los diferentes protocolos que deben seguirse, se dejó abierto un “plan B”, el cual consiste en la comunicación por puerto paralelo, puesto que esta comunicación es mucho más sencilla como barata, pero al mismo tiempo es eficaz y nos brinda la precisión correcta para lo que necesitamos desarrollar de nuestro proyecto.

OBJETIVOS GENERALES

1. Diseñar y desarrollar un taladro robótico que cubra nuestras necesidades de perforación de baquela básicas.

2. Controlar motores paso a paso para el correcto funcionamiento de las piezas mecánicas del robot.

3. Programar un microcontrolador para el movimiento y funcionamiento de los motores.

4. Realizar la comunicación con la computadora a travez del puerto paralelo o serial USB.

5. Diseñar un software o interface que nos permita tener comunicación con Proteus o PCB Wizard, leer y mostrar la posición exacta de cada orificio a perforar, tener control tanto automático como manual de los movimientos de los ejes y nos muestre una interface de comunicación correcta entre la computadora y el microcontrolador.

OBJETIVOS ESPECIFICOS

1. Investigar los temas básicos para adquirir conocimientos solidos sobre el tema y poder tener bases para apoyar el proceso. Lo principales temas a investigar son:

    a. Manejo y control de motores paso a paso.

    b. Programación de un microcontrolador.

    c. Desarrollo de conexión serial USB.

    d. Funcionamiento de sistemas mecánicos de transmisión de movimiento rotacional a traslacional.

    e. Desarrollo de programas y conocimientos en programación en programas de software como c++ y Visual Basic.

2. Elaboración de estructura y partes mecánicas en acrílico o en madera.

3. Creación de prototipo a escala real, análisis y solución de posibles problemas.

4. Montaje de circuitos y terminación de proyecto probando su funcionalidad.

CAPITULO I

Marco Teórico

1. EL ROBOT:

Nuestro proyecto de investigación se centra principalmente en la parte de programación y robótica. El robot que se diseñó, es capaz de moverse en un plano cartesiano de tres dimensiones. Al ser un plano tridimensional, se toman tres ejes (X, Y y Z) siendo los ejes X y Z ejes de movimiento horizontal y el eje Y, eje de movimiento vertical. Logrando el desplazamiento de las piezas adecuadas en cada eje, se logra tener una buena precisión y un buen manejo del motor-tool, además, nos ayuda a poder desplazarnos hacia cualquier parte bien sea de nuestra baquela para circuito electrónico o de nuestro proyecto a desarrollar.

Dentro del área de robótica, encontramos infinidad de modelos de robots que se desarrollan actualmente, debido a que hoy en día el ser humano planea automatizar cualquier tarea o proceso para un mejor rendimiento y calidad. Para nuestro caso, nos permitimos desarrollar la idea de un robot taladro o taladro automatizado, el cual, como se ha mencionado anteriormente, nos permitirá realizar pequeños trabajos con una muy buena precisión.

Para poder automatizar nuestra idea y poder dar solución al problema planteado, hemos pensado en el diseño de taladro tipo taladro de árbol o taladro de columna puesto que este tipo de diseño nos genera un tanto más de fuerza sobre la lámina a perforar y al mismo tiempo nos genera mucha más estabilidad a la hora de perforar.

Básica y sencillamente, nuestro proyecto trata de un taladro de árbol, el cual tiene un movimiento traslacional de arriba abajo, subiendo y bajando el motor-tool después de abrir un orificio, consta también de movimiento traslacional horizontal, el cual le da la libertad al taladro o en

este caso motor-tool, de movilizarse de izquierda a derecha sobre la placa a perforar, y finalmente tiene una base al igual que las demás, en movimiento traslacional de adelante hacia atrás. Dadas las especificaciones anteriores se puede ver y concluir que con el sencillo movimiento de estos tres ejes, se logra cubrir cualquier parte en donde se desee perforar la placa.

El sistema de traslación se genera a través de los motores paso a paso, para poder convertir el movimiento rotacional de los motores a movimiento traslacional, se usan ejes de varilla roscada, o en otros términos un “sinfín”, poniendo en la parte inferior de cada base una tuerca la cual nos generara el movimiento traslacional deseado y por ultimo consta de dos rieles junto a cada eje que nos ayudaran a estabilizar la placa y generar un menor margen de error a la hora de la perforación.

I. ELEMENTOS MECÁNICOS

Los elementos mecánicos son todas aquellas piezas que bien se utilizan para generar movimiento o fuerza permitiendo así mismo la transmisión de estos a otros elementos formando así entre todas las piezas, un correcto y óptimo funcionamiento de nuestro mecanismo. Los principales elementos mecánicos se describen en detalle y se analizan a continuación:

a) Rieles y Bases*:*

Los rieles, son los objetos mecánicos que permiten la traslación de las principales bases, no generan movimiento, pero si lo transmiten, esto quiere decir que la fuerza que aplica el motor sobre el eje central no se tiene que ver muy afectada por la fuerza de rozamiento de los rieles, es de uso estrictamente obligatorio usar los rieles en nuestro proyecto, puesto que si no se utilizaran, las placas base se moverían

inestablemente hacia todos lados y sería un gran problema a la hora de trabajar con el proyecto.

Figura 1. Rieles

- **Varilla roscada o “sinfín”:**

La varilla roscada, o comúnmente llamada “sinfín”, es quizá una de las piezas mecánicas más importantes de nuestro proyecto además de los rieles. Los elementos roscados se usan extensamente en la fabricación de casi todos los diseños de ingeniería y mecánica. Los tornillos suministran un método relativamente rápido y fácil para mantener unidas dos partes y para ejercer una fuerza que se pueda utilizar para ajustar partes movibles.

Los elementos roscados, casi siempre llamados tornillos, tienen una extensa teoría en cuanto a mediciones, tipos y generalidades de tamaño, precisión, fuerza, entre muchas otras variables. Puesto que en nuestro proyecto no vamos a necesitar que las varillas roscadas ejerzan fuerza, ni tampoco necesitamos una precisión exacta o impecable, sencillamente vamos a enfocarnos en el tema de avance. El avance es fácil de explicar, es la distancia que avanzaría el tornillo relativo a la tuerca en una rotación. Para un tornillo de rosca sencilla el avance es igual al paso, para uno de rosca doble, el avance es el doble del paso, y así sucesivamente.

Para nuestro proyecto, utilizamos la técnica del avance, principalmente para poder convertir el movimiento rotacional de los motores en movimiento traslacional de las bases.

- **Bases de madera**:

Para las bases de madera, principalmente se hicieron los planos con las respectivas medidas con las que se iba a trabajar, seguido de esto se desarrolló un prototipo en cartón a escala real, el cual nos ayudó a ver el tamaño real de nuestro proyecto y a corregir pequeños errores de medición puesto que la madera es de un grosor más amplio que el cartón. Una vez solucionados los problemas anteriores y habiendo desarrollado los planos sobre la madera, se procedió a cortar y armar. Para el proceso de construcción y unión de las piezas de madera, se utilizaron puntillas de acero de 1/2 pulgada.

Figura 2. Bases de madera

- **Puntilla de acero, o puntilla acerada**:

Un clavo o también llamado puntilla, es una pieza delgada de metal, bien sea de acero, de aluminio o metal galvanizado. Su principal función es sujetar dos o más objetos. Las puntillas, se dividen usualmente en tres partes: la punta, astil o cuerpo y la cabeza. En una industria fabricadora de clavos, el proceso básico a utilizar a la hora de fabricar clavos o puntillas es: Moldeado de la cabeza, Alimentación de los alambres,

Apretado del alambre, Corte del alambre, Modelado de la punta y Expulsión.

Las puntillas o clavos, se clasifican también de acuerdo a su uso, lo cual nos genera una gran cantidad de modelos. Por ejemplo no todos los clavos son lisos, la cabeza de las puntillas varía según su tamaño y según la estética del trabajo que se desea realizar, por ejemplo si se va a trabajar en un área donde la estética del proyecto no es muy importante, se utilizan puntillas con cabeza, pero si la estética del proyecto es importante y se necesita de una buena presentación, entonces se usan puntillas sin cabeza, además, cabe resaltar que la cabeza de las puntillas puede variar dependiendo del material sobre el cual se va a trabajar y pueden ser de cabeza plana o redonda.

Figura 3. Puntillas o clavos

Las puntillas o clavos, usualmente no se miden por diámetro, sino por longitud, y esa se mide en pulgadas. La pulgada, es una medida de longitud métrica. Anteriormente una pulgada castellana equivalía a 23,22 milímetros. Actualmente en Estados Unidos, Panamá y otros países se usa una pulgada de 25,4 milímetros. En la figura 4, se pueden apreciar las diferentes longitudes de puntillas y su tamaño.

LARGO Pulg.	LARGO (m.m.)	LARGO (m.m.)	CLAVOS POR KILO
3/4	20	14.5	2400
1	25	14.5	1930
1 1/4	31	14.5	1400
1 1/2	38	14.5	1040
2	51	12.5	520
2 1/2	62	11.5	320
3	76	10.5	207
3 1/2	89	9	132
4	101	7	100
4 1/2	112	6	62
5	127	6	52

Figura 4. Longitudes de puntillas

b) Bujes y tuercas:

- **Bujes:**

Los bujes, son pequeñas piezas elaboradas en distintos materiales metálicos o plásticos.

En el ámbito industrial, el buje es el elemento de una maquina donde se apoya y/o gira un eje. En nuestro caso, el buje es la pieza mecánica que se desliza sobre los rieles la cual nos ayuda a mantener la fuerza de fricción en un rango mínimo evitando al mismo tiempo el rápido desgaste de los rieles. Los bujes irán adaptados a las bases movibles de madera de nuestro robot, a través de los bujes pasan los rieles de varilla cromada o acerada los cuales nos ayudan a estabilizar el sistema de movimiento de las bases.

Los tipos de bujes más utilizados son, de bronce, acerados, bipartidos, cónicos, tipo barril y existen unos llamados tipo “eliminator” de venom.

Figura 5. Bujes

- **Tuercas**:

La tuerca, es una pieza metálica con un orificio en el centro, similar a un buje pero presenta una rosca en el centro que se utiliza para acoplar a un tornillo de forma fija o deslizable. La tuerca es una pieza mecánica porque ayuda a mantener la fuerza que generan dos piezas desmontables, pero en el caso de nuestro robot, la tuerca cumple un papel fundamental, ya que la tuerca es la encargada de convertir el movimiento de rotación en movimiento de traslación.

La tuerca siempre sea cual sea el caso, va a actuar con un tornillo que en nuestro caso es una varilla roscada. Esto nos lleva a buscar una tuerca apropiada para nuestra varilla roscada, y para ello se tiene en cuenta primordialmente el tipo de rosca.

Para la utilización y el correcto funcionamiento de una tuerca, se debe fijar que la rosca debe ser la misma que la rosca del tornillo. Las diferentes consideraciones que se deben tener en cuenta al buscar una tuerca para un tornillo son, tener en cuenta si la rosca es derecha o izquierda, sencilla o múltiple, métrica o cuadrada o truncada o redonda.

Las tuercas, tienes varias características que nos ayudan a identificarlas, como el número de caras, el grosor, el diámetro y el tipo de rosca.

El número de caras de las tuercas suele ser 6 (tuerca hexagonal) ó 4 (tuerca cuadrada) como se puede apreciar en la figura 6. Sobre estos modelos básicos se pueden introducir diversas variaciones que imprimen a la tuerca características especiales (ciega, con reborde, ranurada...).

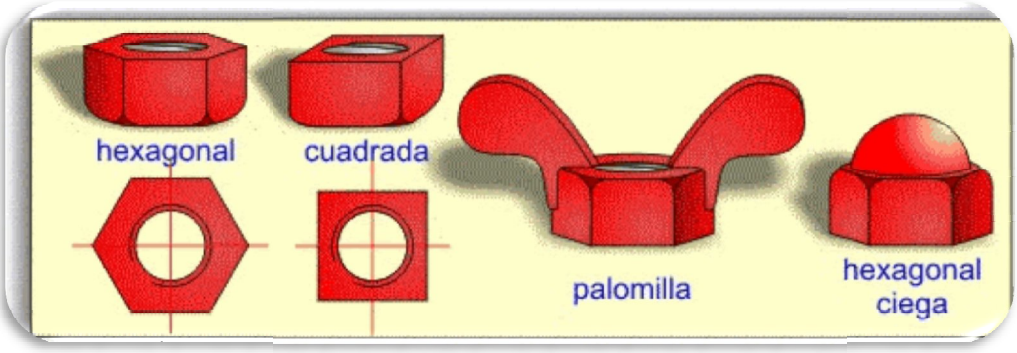


Figura 6. Tipos de tuercas

- Un modelo de tuerca muy empleado es la palomilla, que contiene dos planos salientes para facilitar el giro de la tuerca empleando solamente las manos.

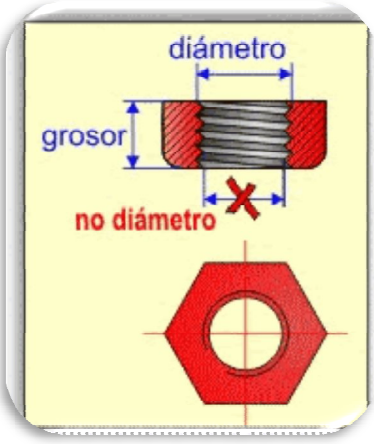


Figura 7. Tipos de roscas

- El grosor es la longitud de la tuerca.
- El diámetro hace referencia al diámetro del tornillo que encaja en ella. Este diámetro no es el del agujero, sino el que aparece entre los fondos de la rosca.
- El tipo de rosca se refiere al perfil de la rosca (que está normalizado) junto con el diámetro del tornillo que encaja en ella.

c) Rodamientos:

Un rodamiento es una pieza mecánica que reduce la fricción entre un eje y las piezas conectadas a este por medio de rodadura, la cual sirve de apoyo y suaviza el desplazamiento.

Figura 8. Rodamientos

Los rodamientos, principalmente usan esferas o rodillos que usan como elemento rotativo. Constan de dos aros metálicos, generalmente de acero o aluminio, en el medio de estos dos aros se encuentran las esferas, las cuales van sobre un anillo de plástico para que no pierdan su ubicación y así puedan generar un movimiento óptimo y puedan hacer la actividad para la cual fueron diseñadas.

A los rodamientos se les aplica una grasa preferiblemente anti-fricción, esto para que las piezas móviles del rodamiento no se calienten generando desgaste de las mismas. Por ultimo a los rodamientos se les

coloca un anillo de plástico, comúnmente llamado tapa, para evitar que el polvo o las partículas de metal, plástico o piedra caigan dentro de las esferas generando un mal funcionamiento y un desgaste casi inmediato de las mismas.

II. ELEMENTOS ELECTROMECÁNICOS:

a) Motores paso a paso:

- **Motor paso a paso (*Mitsumi M35SP-11HPK*):**

SPECIFICATIONS

Items	M35SP-11NK	
Rated Voltage	DC 12V	DC 24V
Working Voltage	DC 10.8~13.2V	DC 21.6~26.4V
Rated Current/Phase	200mA(PEAK)	300mA(PEAK)
No. of Phase	2 Phase	2 Phase
Coil DC Resistance	25Ω/phase±7%	25Ω/phase±7%
Step Angle	3.75˚/step	3.75˚/step
Excitation Method	2-2 Phase excitation(Bipolar driving)	
Insulation Class	Class E insulation	Class E insulation
Holding Torque	14.7mN·m	20mN·m
Pull-out Torque	8.9mN·m/1200pps	12.1mN·m/1800pps
Pull-in Torque	10.8mN·m/400pps	13.8mN·m/600pps
Max. Pull-out Pulse Rate	1900pps	3300pps
Max. Pull-in Pulse Rate	920pps	1250pps

Figura 9. Motor paso a paso Mitsumi M35SP-11HPK y especificaciones.

El motor paso a paso Mitsumi M35SP-11HPK, es un motor de torque medio y de bajo consumo. Está diseñado principalmente para usarse en escáneres y multifuncionales, además que es un motor de excelente precisión. Entre las características más importantes, encontramos que este motor trabaja en un rango de 12v a 24v DC, a pesar de que puede empezar a trabajar con un voltaje mínimo de 10.8v DC y un voltaje máximo de 26.4v DC. Es un motor que tiene un ángulo de 3.75º por paso, esto quiere decir que en 24 pasos nos genera una rotación de 90º.

En la figura 10, podemos observar la curva característica de trabajo de este motor.

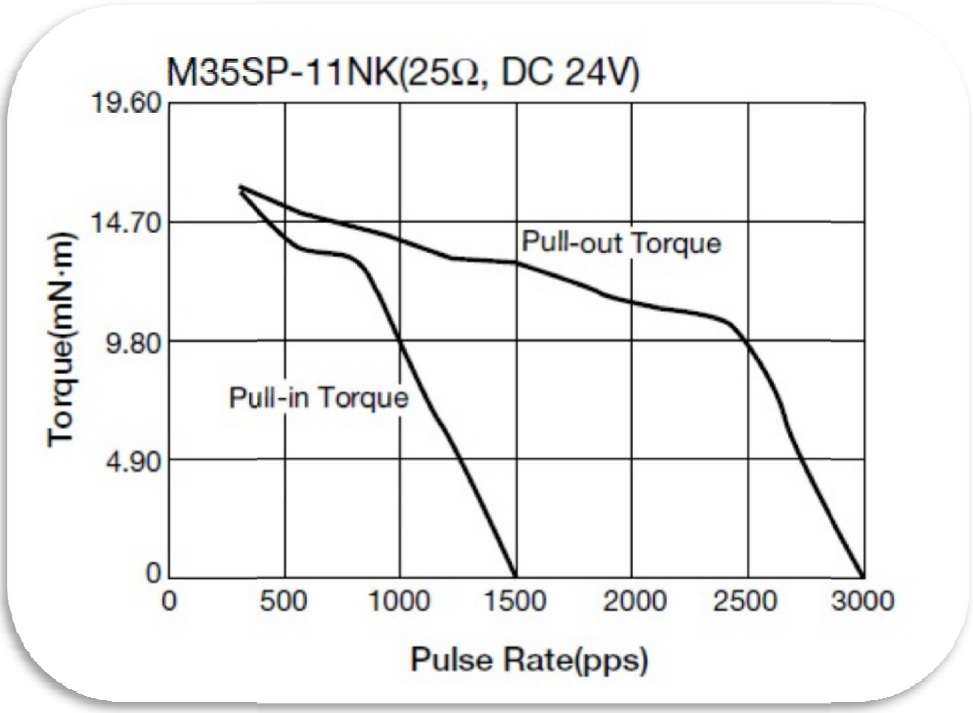


Figura 10. Curva de trabajo motor PAP M35SP-11HPK

- **Motor paso a paso (*Mitsumi M35SP-10N*):**

SPECIFICATIONS

Item	M35SP-9	M35SP-10N
Rated Voltage	DC 24V	
Working Voltage	DC 21.6~26.4V	
Rated Current/Phase	517mA	
No. of Phase	4 Phase	
Coil DC Resistance	50Ω/phase±7%	
Step Angle	7.5°/step	3.75°/step
Excitation Method	2-2 Phase excitation(Unipolar driving)	
Insulation Class	Class E insulation	
Holding Torque	51.9mN·m	26.5mN·m
Pull-out Torque	35.8mN·m/200pps	13.7mN·m/400pps
Pull-in Torque	35.3mN·m/200pps	13.2mN·m/400pps
Max. Pull-out Pulse Rate	970pps	880pps
Max. Pull-in Pulse Rate	740pps	770pps

Figura 11. Motor paso a paso Mitsumi 35SP-10N y especificaciones.

A diferencia del motor anterior, el motor M35SP-10N, es un motor paso a paso de alto torque y de bajo consumo ademas de ser de tamaño compacto. Generalmente se usa en dispositivos como impresoras, multifuncionales, fotocopiadoras, fax, entre otros. Entre las principales caracteristicas de este motor, podemos encontrar que trabaja en un voltaje de 24v DC, siendo el voltaje minimo 24.6v DC y el voltaje maximo 26.4v DC. Al igual que el motor anterior, este motor tiene un angulo de

3.75º por paso, igualmente generandonos un angulo de rotacion de 90º en 24 pasos.

En la figura 12, se puede apreciar la curva caracteristica que aplica para este motor.

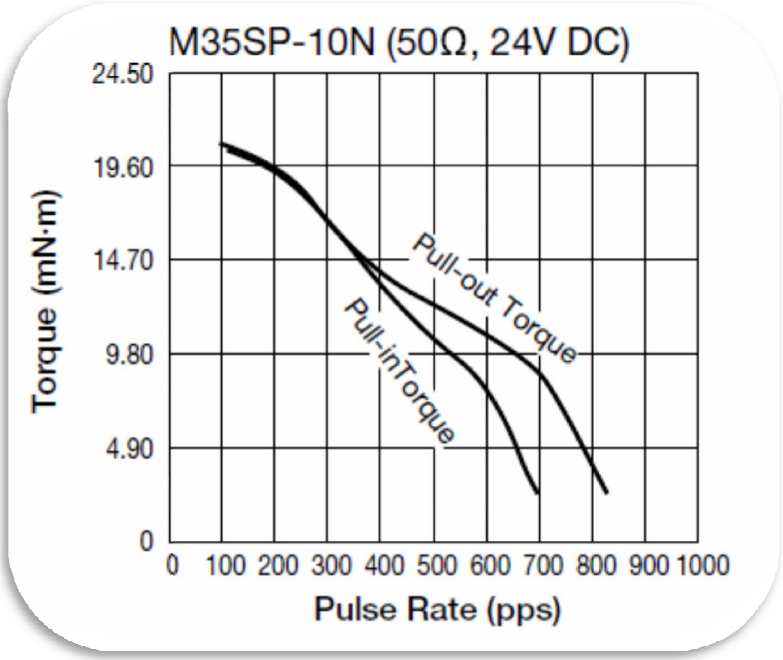


Figura 12. Curva de trabajo motor PAP M35SP-10N.

- **Control de motores:**

Los motores tanto paso a paso como los DC, necesitan tener un control tanto de prendido y apagado, así como de velocidad.

a. **Control de motores PAP:**

Los motores paso a paso como dice su nombre, son motores que dan un paso a la vez cuando un pulso es aplicado. Este paso

puede variar dependiendo de la cantidad de dientes que el fabricante diseñe. Estos dientes tienen ángulos
desde el 0,72 ° hasta los 90° brindando diferentes precisiones según la necesidad. Como se puede observar en la tabla 1, dependiendo del ángulo se tiene el número de pasos para dar una vuelta completa.

Tabla 1: Ángulos y cantidad de pasos	
Angulo	**Cantidad de pasos**
0,72°	500
1,8°	200
3,75°	96
7,5°	**48**
15°	24
90°	4

Tabla 1. Ángulos y cantidad de pasos.

En la siguiente figura, figura 13, se puede observar el rotor y el estator. Tanto el rotor como el estator en un motor paso a paso tienen la misma cantidad de dientes los que dependen del número de pasos que se necesite, es decir, entre menor ángulo mayor cantidad de dientes se debe fabricar, y entre más dientes más cantidad de pasos y mayor precisión se va a obtener.

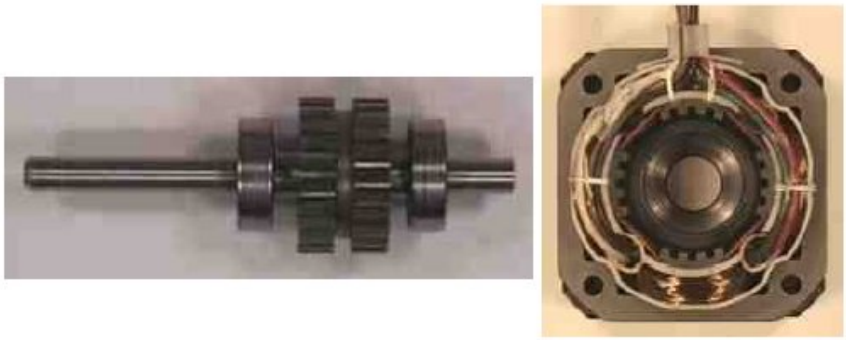


Figura 13. Rotor y estator de un motor paso a paso.

Entre los motores paso a paso, se pueden encontrar diferentes variaciones, tales como los motores paso a paso bipolares y unipolares. La diferencia entre estos dos motores es la cantidad de hilos que se encuentran. Como se observa en la figura 14, el motor paso a paso bipolar tiene 4 hilos, mientras que el unipolar presenta 6 hilos. En este tipo de motores dependiendo del fabricante podemos encontrar 5 o 6 hilos ya que el común como se observa en la figura, puede venir tanto unido como sueltos, es por aquella razón que encontramos más o menos hilos.

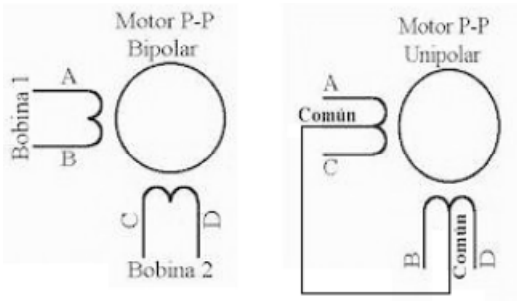


Figura 14. Formación de hilos motores paso a paso bipolares y unipolares.

Para controlar un motor paso a paso, necesariamente se debe usar pulsos para activar cada bobina.

Para la mayoría de casos, los motores paso a paso son motores de baja corriente, sin embargo, si el control se lo va a realizar por medio de un microprocesador, se debe usar un puente H, ya que se debe modificar el voltaje y se debe usar corriente continua para activar cada pulso en la bobina.

Como se puede ver en la figura 15 mediante el uso de un microprocesador, este envía el pulso de activación el cual activa la salida del puente H el cual activa la bobina conectada.

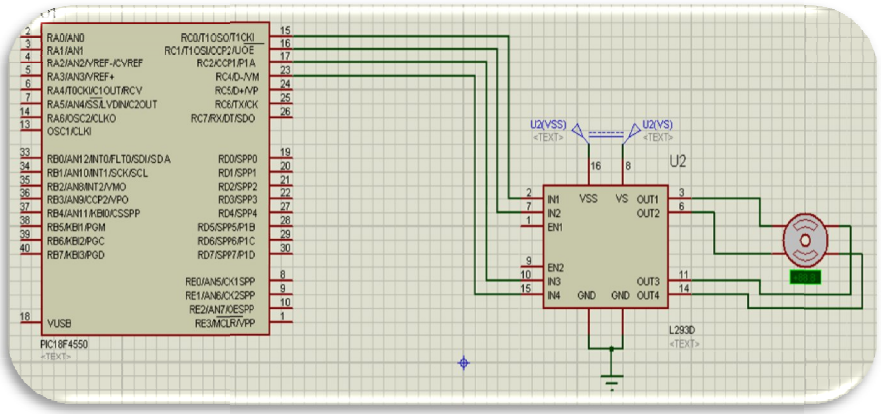


Figura 15. Conexión motor paso a paso bipolar.

De igual forma para un motor paso a paso unipolar, se conecta éste al puente H y al Pic. La única variante para este motor es que los hilos comunes se conectan a la fuente de energía.

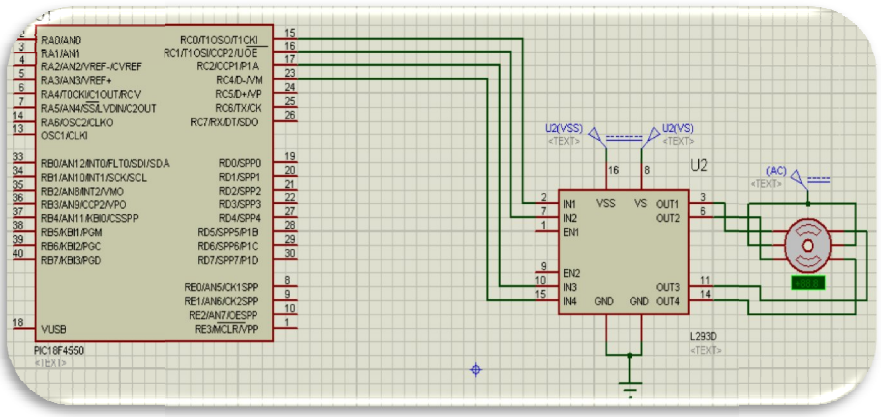


Figura 16. Conexión motor paso a paso unipolar.

Existen secuencias para controlar este tipo de motores, ya que se debe activar una bobina, luego la siguiente y así sucesivamente para poder mover el rotor del motor.

Entre las secuencias que se puede generar existe:

· Wave drive

· Full Step

· Half Step

La secuencia Wave drive, es la que activa una bobina a la vez, y así sucesivamente con la siguiente. Como se puede apreciar en la tabla 2, podemos generar una tabla de verdad, la que nos indica que bobina tiene que estar activada para generar la secuencia. De esta forma podemos programar en el Pic, que puerto tiene que activar y que puerto debe desactivar. De la misma forma poniendo la secuencia al revés se obtiene el giro contrario al programado.

Tabla 2: Tabla de verdad secuencia wave drive.

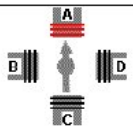
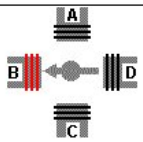
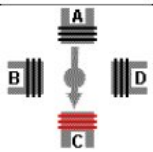
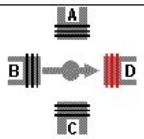
Paso	Bobina A	Bobina B	Bobina C	Bobina D	
1	1	0	0	0	
2	0	1	0	0	
3	0	0	1	0	
4	0	0	0	1	

Tabla 2. Tabla de verdad secuencia wave drive.

La secuencia Full Step, es la secuencia más común, que se genera, ya que se activan dos bobinas a la vez generando más torque. Este tipo de secuencia es la recomendada por el fabricante. Como se observa en la tabla 3, de igual forma se observa la tabla de verdad, para la secuencia Full step.

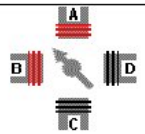
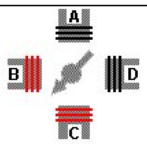
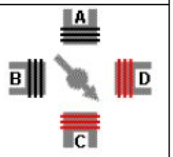
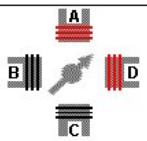
Tabla 3: Tabla de verdad secuencia Full Step.					
Paso	**Bobina A**	**Bobina B**	**Bobina C**	**Bobina D**	
1	1	1	0	0	
2	0	1	1	0	
3	0	0	1	1	
4	1	0	0	1	

Tabla 3. Tabla de verdad secuencia Full Step.

La secuencia Half Step, es una combinación de las dos secuencias anteriores, haciendo que el pulso que se genera camine solo medio paso.

Como se observó previamente, el full step hace que el rotor camine un paso, pero entre cada bobina activada y el wave drive camina entre cada bobina activada.

Si unimos los dos, es decir, primero wave drive y luego full step, hacemos que el rotor gire medio paso haciendo que en vez de tener 4 tiempos, se hacen 8. Como se puede observar en la tabla 4, es la tabla de verdad para una secuencia Half step.

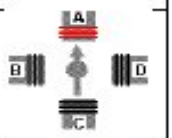
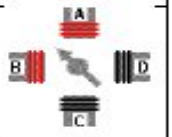
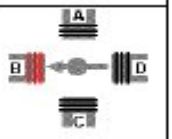
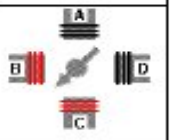
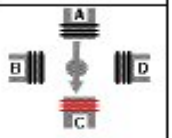
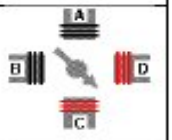
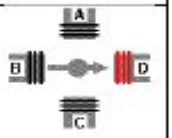
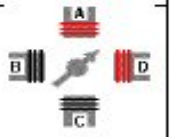
Tabla 4: Secuencia Half Step.					
Paso	**Bobina A**	**Bobina B**	**Bobina C**	**Bobina D**	
1	1	0	0	0	
2	1	1	0	0	
3	0	1	0	0	
4	0	1	1	0	
5	0	0	1	0	
6	0	0	1	1	
7	0	0	0	1	
8	1	0	0	1	

Tabla 4. Secuencia Half Step.

La velocidad de un motor paso a paso, depende específicamente del tiempo en que se demora en cambiar de un estado a otro, generalmente este tiempo para un giro continuo está entre los 3 y 50 ms. Esto se puede configurar fácilmente en la programación de nuestro microcontrolador, usando la opción **PAUSE XXX**, donde XXX es la cantidad de milisegundos.

- **MOTOR-TOOL:**

Para nuestro trabajo, como herramienta electromecánica adicional, se va a utilizar el manejo de un motor-tool, el cual nos generará las perforaciones en la placa a perforar.

El motor-tool a utilizar es de velocidad directa, no dispone de variaciones de velocidad como la mayoria de estas herramientas, ofrece una buena potencia y un torque óptimo. Sus dimensiones proporcionan una zona de agarre de 360º para comodidad y precisión. Tiene como medidas entre un rango de 11cm a 14cm de largo y no mas de 6 cm de diametro, esto nos ayuda bastante a la hora de ensamblar nuestro proyecto, puesto que al ser una herramienta de tamaño considerable, podemos hacer una buena y fácil implementacion de este componente en nuestro proyecto.

III. ELEMENTOS ELECTRÓNICOS:

- **Micrococntroladores (Pic 18f2550) y (Pic 18f4550):**

Durante el proceso de diseño y desarrollo de nuestro proyecto, utilizamos principalmente dos microcontroladores, el PIC 18f2550 y el PIC 18f4550, siendo estos dos de los principales y mas comunes microcontroladores fabricados por la empresa de Microchip, preferimos utilizar estos ya que su funcionamento es muy sencillo y fueron dos de los primeros tipos de microcontroladores creados por la empresa de Microchip con interface USB.

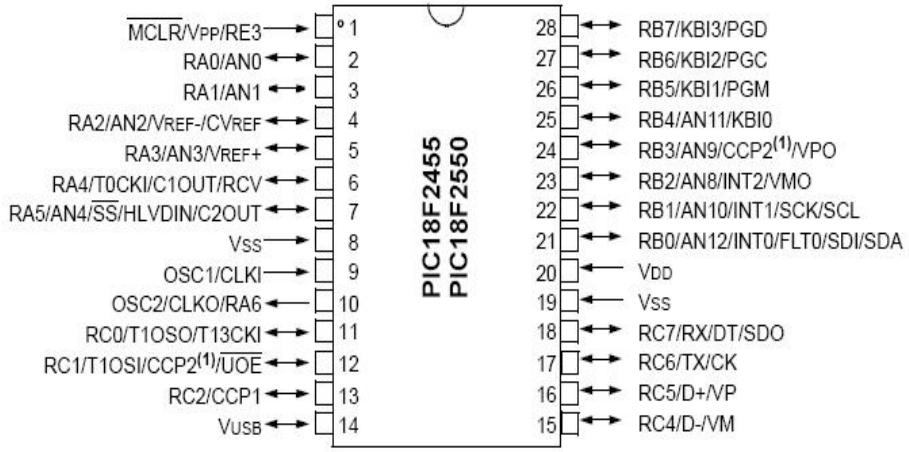


Figura 17. PIC 18F2550.

La serie 18F de los microprocesadores, es la primera serie que salió con conexión USB, fabricado por la compañía Microchip1. Esta serie nos permite utilizar de la misma forma que la serie 16F.

40-Pin PDIP

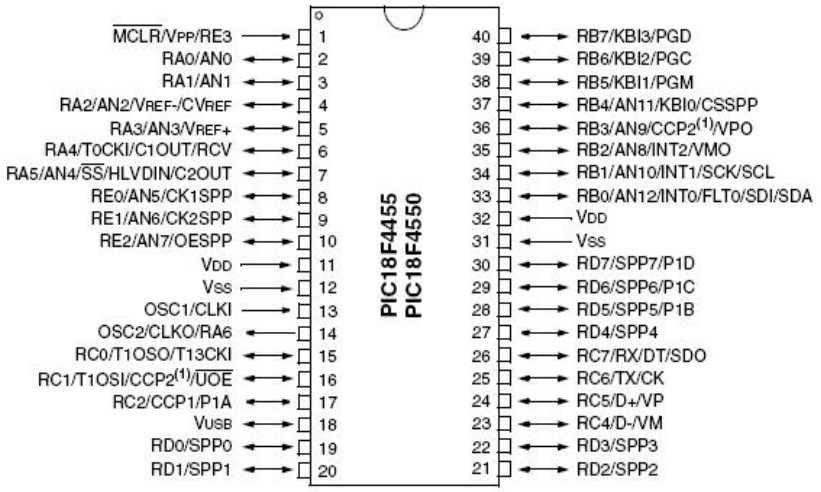


Figura 18. PIC 18F4550.

Dentro de las características principales de estos microprocesadores, encontramos que:

· Conexión de alta velocidad USB 2.0 (12Mbs)

· 48 MHz de oscilación

· Entradas análogas con comparadores

· 27 y 35 entradas y salidas digitales respectivamente

· Bajo consumo de energía

En nuestro caso solo se van a usar entradas y salidas digitales de los puertos B, C, y D en el caso del PIC 18F4550. Estos microprocesadores nos permite programar tanto en “assembler” como en Basic, de esta forma se puede controlar las entradas y las salidas que se van a utilizar.

Ademas del microcontrolador, se necesitan varios componentes externos para que este funcione de la manera mas adecuada, dentro de estos componentes se encuentra el cristal de cuarzo, resistencias, capacitores entre otros.

- **Puente H (L293B):**

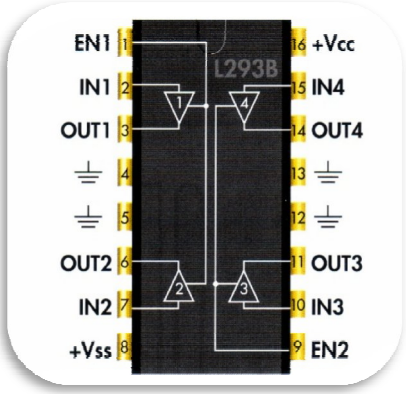


Figura 19. Puente H (L293B)

El L293D más conocido como puente H, es un circuito integrado, el cual nos permite manejar cargas diferentes con las que se está trabajando, esto nos permite controlar pequeños motores, relés, solenoides y demás elementos eléctricos que manejen un voltaje diferente al voltaje lógico.

En nuestro caso, el puente H, nos va permitir manejar un voltaje de entre 12 a 24 VDC para los motores, mientras que el PIC va a manejar un voltaje lógico de 5VDC.

El puente H puede manejar corrientes de hasta 1Amp., lo que nos permite manejar motores, DC, Paso a Paso y motores tipo servo.

El puente H, es muy útil ya que gracias al control TTL3, nos permite que una señal lógica active la entrada haciendo que esa salida sea activada con el voltaje diferente

del canal 8; esto nos permite controlar cambio de giro de motores y generar los pasos deseados para un motor paso a paso.

IV. PRODUCCIÓN DE PLACAS PCB:

La producción de placas PCB es un proceso de cinco pasos los cuales se componen de:

- Generación y simulación del circuito en el software.
- Impresión o marcado del circuito en la placa.
- Uso del ácido para retirar el cobre en exceso.
- Perforación de los orificios para inserción de los elementos electrónicos.
- Soldadura de los elementos.

Estos cinco pasos mostrados anteriormente, son pasos que van en cadena, es decir, que primero se debe realizar uno a uno para continuar con el siguiente proceso en la generación de una baquela, y además debe ser realizado con paciencia.; esto se debe a que poco a poco y con experiencia se va aprendiendo como realizarlas mejor.

- **Diseño y simulación del circuito en software:**

El primer paso para comenzar con la generación de un circuito impreso, es la simulación en un software, esto nos ayudara a detectar posibles errores a la hora de hacer el diseño y asi poder evitar futuros problemas en el funcionamiento de nuestro circuito a desarrollar.

Existen programas, tales como Orcad Pspice, Livewire - PCB Wizzard, Proteus, Eagle, entre otros.

Estos programas cuentan con una amplia gama de componentes electrónicos que podemos utilizar para poder simular gran mayoria de circuitos.

En nuestro caso los dos principales programas en los cuales nos centraremos y sobre los cuales vamos a trabajar, son el Proteus y el PCB wizard.

En la figura 20 se puede ver la simulacion de un circuito sencilo en proteus.

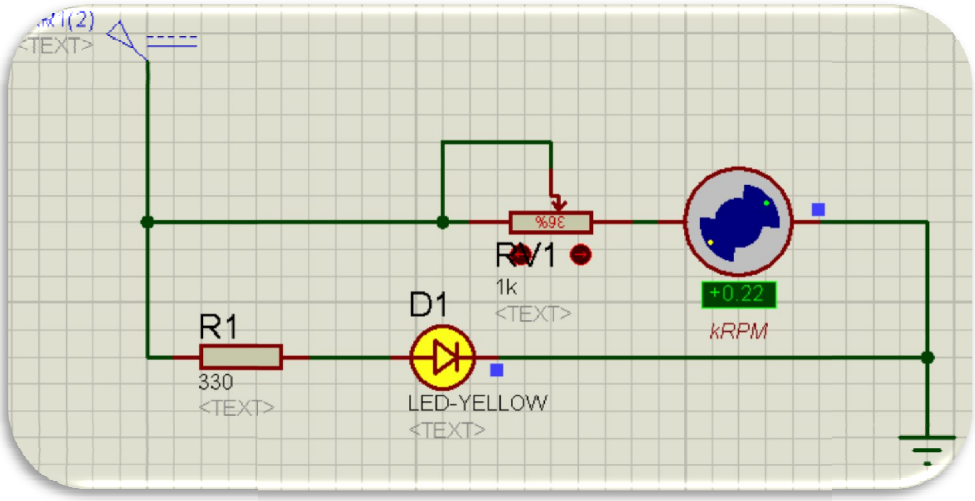


Figura 20. Simulación de circuito en proteus.

Después de haber realizado la simulación, el paquete de Proteus ofrece Ares, el cual es un programa complementario para la generación de circuitos impresos el cual tiene dos opciones:

La primera, es realizar el circuito a mano, es decir utilizar los componentes y unirlos al gusto del usuario poniendo la posición y la forma de unión de los caminos.

La segunda opción es utilizar el “autoruting” el cual es un componente de este programa que automáticamente genera los caminos y la posición de los elementos.

Como se observa en la figura 21, se ha realizado con “autoruting”.

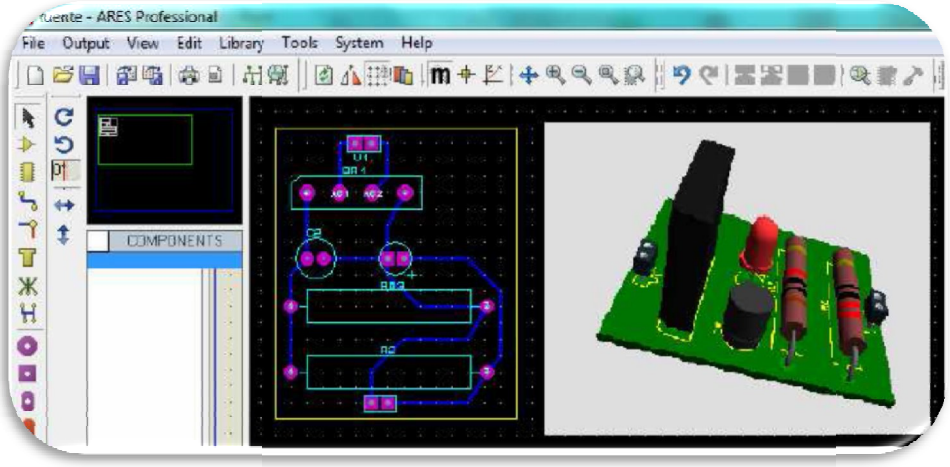


Figura 21. Circuito impreso generado por Ares en autoruting.

Al igual que el Proteus, el Livewire - PCB Wizzard también tiene la opción de simular como se observa en la figura 22, observándose el funcionamiento de un circuito simulado en Livewire - PCB Wizzard.

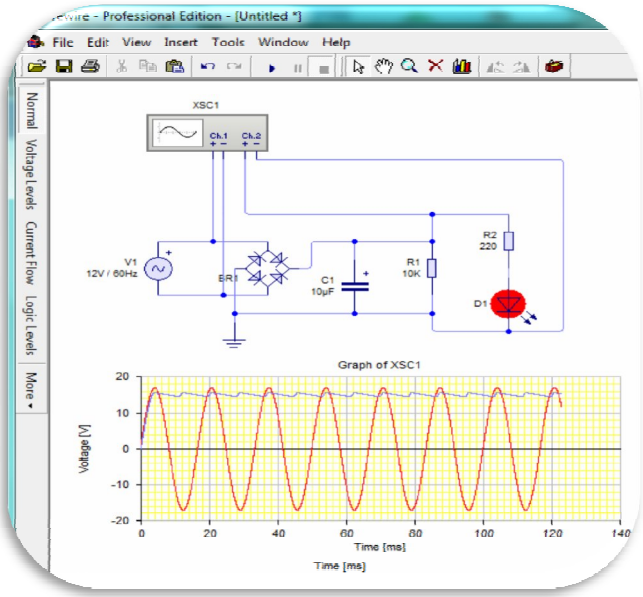


Figura 22. Simulacion de circuito en PCB wizard.

Al igual que el proteus, el PCB wizard cuenta con una herramienta llamada Livewire, la cual nos ayuda a generar el circuito a imprimir en la baquela. Tambien, cuenta con las dos modalidades de uso, enrutamiento automatico o enrutamiento manual, el uso de cada uno, depende de los gustos del diseñador.

En la figura 23 se puede observar el diseño de circuito listo para imprimir en Livewire.

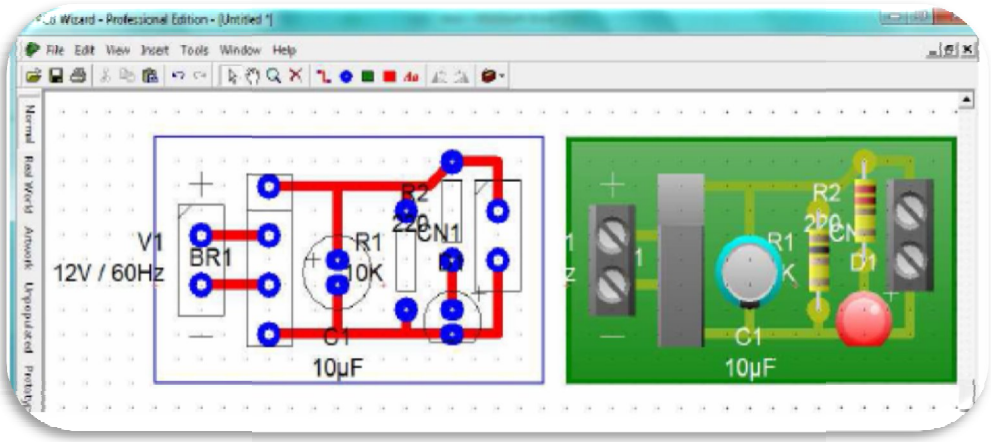


Figura 23. Diseño de circuito para imprimir en Livewire.

- **Impresión o marcado del circuito en la baquela:**

La impresión y la adhesión, es la segunda etapa para la fabricación de una placa PCB. Esta etapa está dividida en dos acciones: la de imprimir en el papel transferencia y la de planchar el papel transferencia. Estas dos acciones están unidas debido a la manipulación del papel transferencia.

El papel transferencia, es un papel de transferencia pres-n-peel12 el cual se lo puede conseguir en cualquier tienda electrónica. Este papel es un papel del tipo glossy, es decir, que tienen una capa de barniz lo que le permite que tenga aquel brillo característico.

Para realizar la impresión dependiendo del programa, se deben tomar en cuenta algunas consideraciones; en el caso del Proteus como se observa en la figura 24, podemos ver los puntos en lo que hay que poner atención para que la impresión salga correcta.

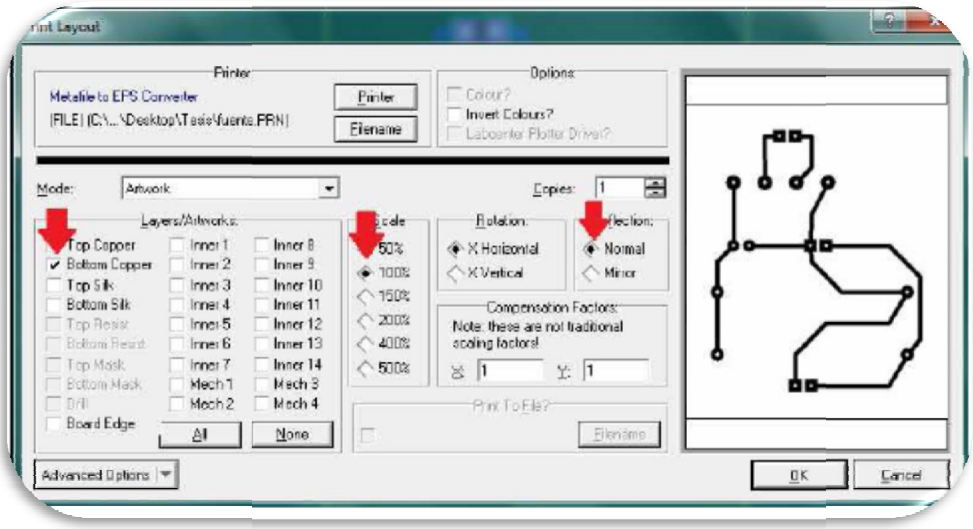


Figura 24. Caracteristicas de impresión proteus.

La primera flecha, de izquierda a derecha, nos indica que se va a imprimir, en éste caso el BOTTON COOPPER que es el área donde se ha realizado los caminos y los PAD´s (punto en el cual se va a realizar el orificio y se va a soldar). La segunda flecha nos indica la escala, por lo general es al 100%. Este porcentaje nos indica el tamaño de la placa en la vida real. Y la tercera flecha nos indica si queremos la impresión normal o “mirror” (espejo).

La impresión normal nos imprime tal cual se ha diseñado el circuito, y la opción mirror nos imprime como se lo hubiese volteado horizontalmente al circuito.

En nuestro caso debido a que se va a utilizar una placa de fibra de vidrio de un solo lado de cobre se realiza la impresión utilizando la opción normal, lo que nos va a permitir tener los caminos y los PAD's en la parte de cobre y en la que no tiene, tener los elementos.

Al igual que el Proteus, el Livewire - PCB Wizzard posee la misma opción para observar cómo quedan los caminos y los PAD's. Como se observa en la figura 25, la flecha nos indica que haciendo clic en el botón de "Artwork", se puede fácilmente observar cómo va a quedar la impresión.

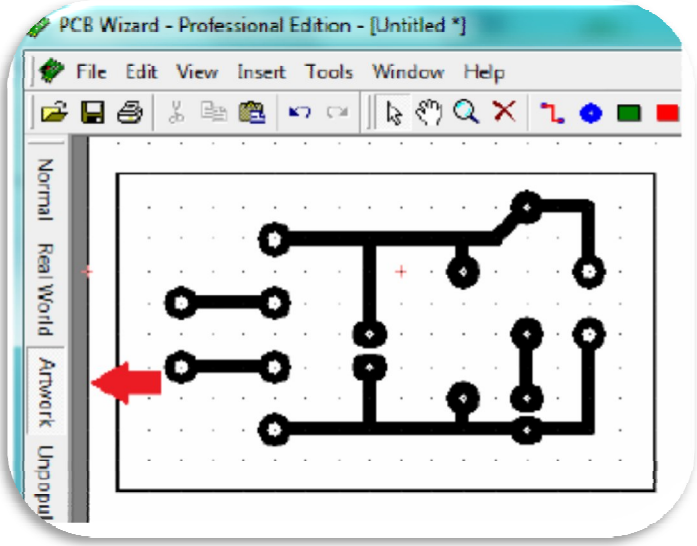


Figura 25. Opcion para ver los caminos en PCB wizard.

En el caso del Livewire - PCB Wizzard - PCB Wizzard, al momento de querer realizar la impresión, este no tiene un ambiente amigable con el cual uno pueda interactuar como el Proteus.

Una vez observado la figura de cómo va a ser la impresión solo se hace clic en el botón print (imprimir), de esta forma conseguimos lo mismo que estamos observando en el papel transferencia.

Un detalle muy importante al momento de realizar la impresión, es realizarla en una impresora laser, ya que el barniz del papel transferencia que por efectos químicos se une con el polvo del tóner, permitiendo que fácilmente los caminos y los PAD’s queden levemente levantados.

Una vez que tenemos nuestra impresión, se puede proseguir a cortar la placa. Se debe adicionar 4mm más a los bordes impresos del circuito impreso. Una vez cortado se prosigue con la transferencia térmica, èste proceso consiste mediante calor hacer que los caminos impresos en la hoja sean pegados a la placa. Antes de proseguir es recomendable limpiar el cobre con una esponja de acero, ya que se deben quitar los rayones y el óxido creado en la superficie.

Una vez limpia la placa, se debe manipular por los bordes, ya que la grasa de los dedos genera óxido en la superficie de cobre; se debe colocar sobre una mesa firme la placa y sobre ésta el papel transferencia con los caminos hacia el cobre.

Usando una plancha doméstica a su mayor potencia (máxima temperatura), se debe poner sobre la placa por 30 segundos haciendo máxima presión sobre ésta para que los caminos se transfieran bien hacia la placa.

Después del tiempo indicado, usando una franela, movemos la placa, y comenzamos a aplicar presión uniformemente frotándolo de un lado a otro hasta que se enfríe.

Una vez terminado el proceso se retira el papel transferencia, observando que los caminos y PAD’s quedaron correctamente transferidos a la placa.

- **Aplicación del ácido para retirar el cobre:**

Cuando los caminos han sido transferidos a la placa, se debe retirar el exceso de cobre para que solo los caminos protegidos por el barniz y la tinta queden. Para esta etapa, se necesitan algunos materiales extras tales como un envase plástico o de vidrio ,un par de fundas de cloruro férrico, los cuales venden en cualquier tienda electrónica, también un palillo de pinchos para poder mover y alzar la placa.

Primero se hierven dos vasos de agua, una vez que el agua está hirviendo se pone en el recipiente de plástico o de vidrio, no se recomienda usar un envase metálico ya que el ácido que se va a usar corroe el metal, haciendo que el envase usado no sirva.

Una vez vertido el agua en el envase, se deja caer la placa y con mucho cuidado se va arrojando poco a poco el ácido.

En esta etapa del proceso de debe tener mucho cuidado, ya que la reacción del agua con el ácido, es de acción agresiva. De igual forma se lo debe realizar en un lugar con buena ventilación y con ropa preferiblemente vieja, ya que el ácido salta y mancha.

Una vez que el ácido ha hecho contacto con el agua, se lo comienza a mover de lado a lado, formando un oleaje. Este nos permite acelerar el proceso, ya que el ácido en constante movimiento remueve de forma más fácil el exceso de cobre.

Debido a que es un proceso que puede tardar un poco y el ácido es obscuro, mediante el uso del palo de pincho se lo usa para sacar la placa y ver que tanto se ha avanzado en el proceso.

Una vez que se haya retirado todo el exceso de cobre, se lo debe lavar con abundante agua hasta que la placa quede limpia, y se prosigue a secarla con una toalla.

Cuando la placa está seca, se puede observar que los caminos y los PAD’s tienen un color negro producto de sumersión en el ácido. Cuando se secan bien los caminos éstos toman un color blanco, para retirar estas impurezas, se tienen dos formas para limpiar.

La primera utilizando acetona, tiñer o cualquier disolvente con un poco de algodón. El efecto secundario de este tipo de limpieza, si bien limpia los caminos y nos permite ver el cobre, deja un color negrizo a la placa dejándola sucia y poco presentable.

La segunda forma de limpiar es utilizar la misma esponja metálica remojada en agua de ésta forma, conseguimos quitar el color y pulir los caminos entregando una placa nítida y en perfecto estado.

- **Perforación de los orificios:**

La perforación de los orificios, es la etapa en la que se hace el espacio, generalmente redondo para que el terminal metálico pase entre en la placa, permitiendo que ésta sea aislada de los demás elementos.

Existen varios tipos de taladros, los cuales nos pueden ayudar en est etapa, entre estos tenemos:

· Taladros normales

· Taladros miniaturas o moto-tool

· Taladro industrial con pedestal

Para los dos primeros tipos de taladro, previamente se debe utilizar una puntilla y un martillo para fijar el punto donde se va a realizar el orificio. De la misma forma si no se desea usar el martillo, con un cuchillo o una navaja, se puede presionar y realizar una pequeña hendidura, esto nos sirve para poder fijar la broca evitando que se mueva y fallar la perforación.

Una vez realizada las hendiduras en las que se van a realizar los orificios, usando una broca de 1mm de diámetro, se realizan los huecos que sean necesarios.

- **Soldadura de los elementos:**

La soldadura de los elementos, es la unión del terminal del dispositivo eléctrico con el camino de cobre mediante el calentamiento del estaño. De esta forma conseguimos que el dispositivo quede pegado a la placa de tal forma que la corriente que pasa por los caminos al ser el estaño un elemento conductor, pase hacia el elemento eléctrico.

Una vez hechos los orificios, y con todos los elementos que son parte de la placa, se procede desde los más bajos a los más altos, por ejemplo, primero van las resistencias, los diodos y demás dispositivos que irían acostados. Luego irían lo zócalos, borneras, y para finalizar los dispositivos más altos por ejemplo los capacitores, de esta forma podemos ir apoyando en la mesa sin perder el equilibrio de la placa.

Una vez que todos los dispositivos están en su sitio coloca colocados con la polaridad correcta, en el caso de los capacitores electrolíticos y los leds, se procede a soldar.

La mejor técnica de soldadura, se basa en tener limpia la punta y muy caliente el cautín, de esta forma vamos a derretir el estaño más fácilmente. Con la punta limpia y caliente se toca el camino de cobre y la base de la pata, de esta forma se va a calentar rápidamente y el estaño se va a derretir. Cuando este haya sido aplicado alrededor de la pata, se suelta y en pocos segundos se seca el estaño.

Una vez soldadas todas las puntas se prosigue a cortar las patas con un alicate, haciendo que quede todo a un mismo nivel.

V. Conexión USB:

La comunicación USB (Universal Serial Bus) entre el microcontrolador y la computadora, es una técnica que se la viene usando desde que la serie 18F de microprocesadores salió al mercado. Esto se debe a que se pueden manipular velocidades de 1.2 Mbps. con el USB 1.0 y 480 Mbps. con el USB 2.0. También teniendo en cuenta que la mayoría de computadoras que se encuentran hoy en el mercado carecen tanto de puertos seriales como puertos paralelos lo que nos permitía antes realizar la comunicación.

Entre las muchas formas que existen para comunicar con USB, existe el EASY HID, el cual es un pequeño programa hecho por la compañía MECANIQUE, el cual nos genera dos plataformas de desarrollo. Entre estas podemos encontrar para los microprocesadores, el Pic Basic Pro y Proton que nos permite programar los Pic´s de la serie 18F, tales como 18F2455, 18F2550, 18F4455 y 18F4550, y para la computadora nos genera el código en Borland Delphi, Visual C++ y Visual Basic.

Este programa también nos entrega lo que son los drivers para que reconozca la computadora al Pic el momento de la primera conexión.

Cuando uno inicia el programa, la primera pantalla que nos aparece, como en la figura 26, ésta nos permite cargar el nombre del producto que se está desarrollando.

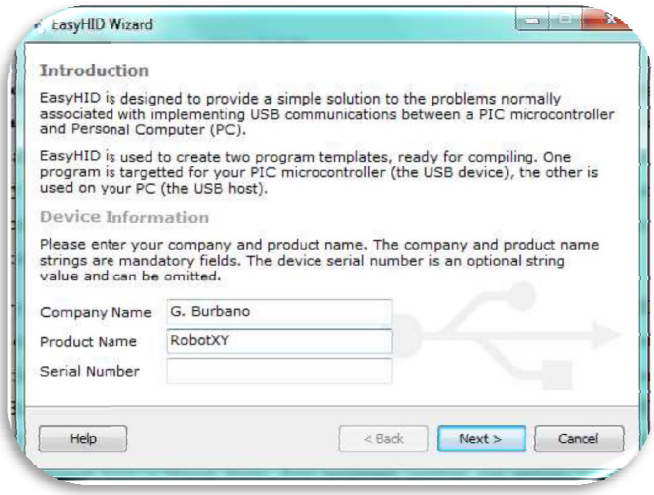


Figura 26. Primera pantalla de Easy Hid.

La siguiente página nos entrega lo que son el VENDOR ID y el PRODUCT ID, los cuales son números asignados por el organismo que regula la autenticidad de los productos USB donde se puede comprar por U$S 4.000 una membresía anual con números otorgados exclusivamente al desarrollo. Al ser un proyecto sin fin de lucro, se puede mantener los mismos datos, para realizar la aplicación.

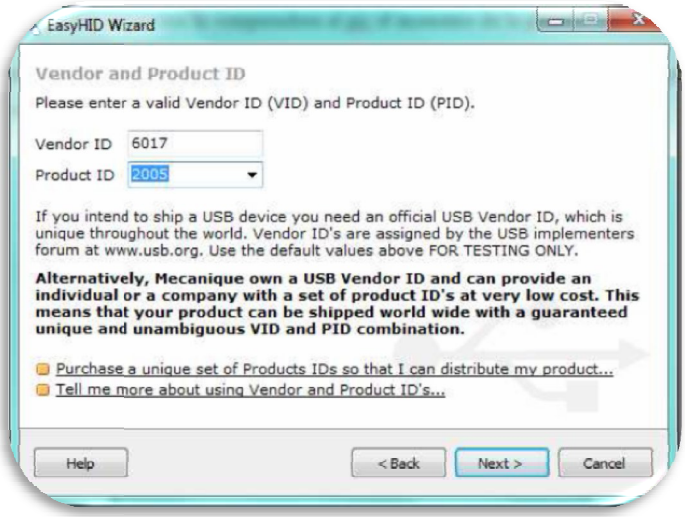


Figura 27. Pantalla de Vendor y Product ID.

La tercera etapa de este programa, como se muestra en la figura 28, se observa las diferentes opciones del USB. Son casillas donde uno puede cambiar las diferentes opciones. En Polling input y polling output como se observa son datos de tiempo. El polling input es el intervalo usado por la computadora para pedir datos del Pic, mientras que el polling output es el intervalo usado para enviar datos al Pic. Como se observa, es un tiempo de 10 ms, lo que se demora en recibir y enviar datos. El bus power, es la cantidad de corriente que maneja el puerto USB, èste tiene 50 miliamperios lo que generalmente maneja el USB de las computadoras. En Buffer input y en Buffer output, es la cantidad de bytes que se van a usar para enviar y recibir datos hacia la computadora, y se puede generar hasta 64 bytes de comunicación. La recomendación del fabricante como se observa, es que si no está seguro de la modificación de los datos, se debe trabajar con lo pre-establecido.

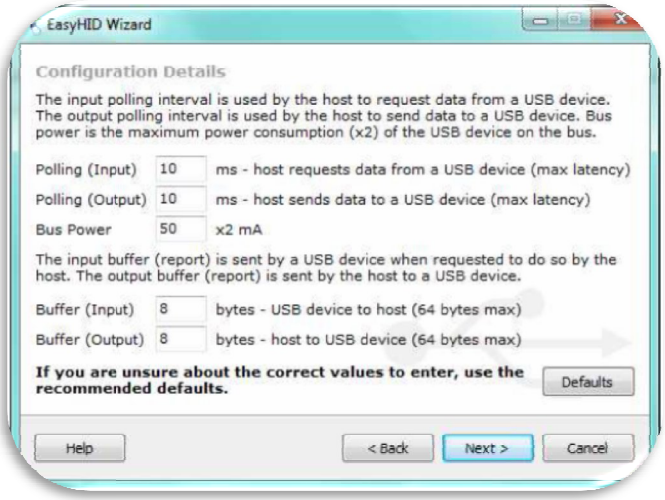


Figura 28. Configuración de la cantidad de bytes.

La figura 29, nos indica la pantalla más importante de todas, ya que como se observa, aquí es donde se selecciona el Pic con el que se va a trabajar, el programa con el cual se va a generar el código para el Pic, el programa que va a controlar el Pic, el nombre del archivo y su ubicación.

En nuestro caso, se va a usar un Pic 18F4550, se lo va a programar en PIC BASIC, y se va a usar VISUAL BASIC 5 para poder controlar la comunicación con el Pic.

Figura 29. Opciones de programas y pic's.

Una vez aceptado todo, se procede a compilar como se observa en la figura 30, se va indicar que no existen problemas y se generaron las aplicaciones correctamente.

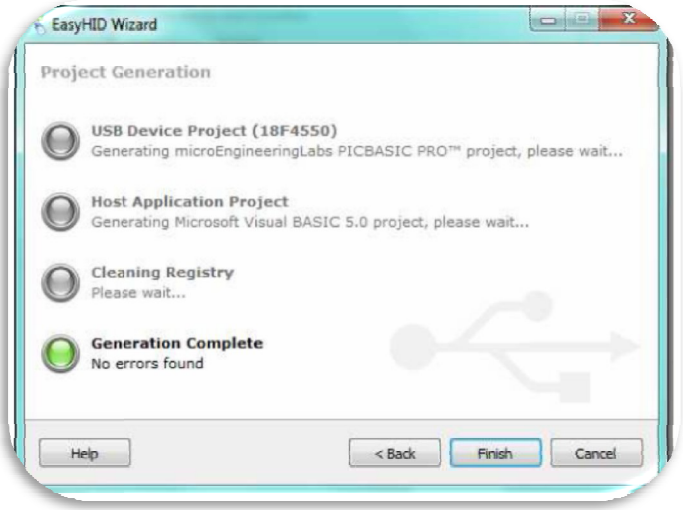


Figura 30. Confirmación de la generación de códigos.

En el lugar donde se guardaron los archivos podemos observar las plataformas creadas por el programa, como en la figura 31 nos muestra

con las flechas el programa generado con la extensión PBP que es para el Pic Basic Pro y VPB que es la extensión para el visual Basic.

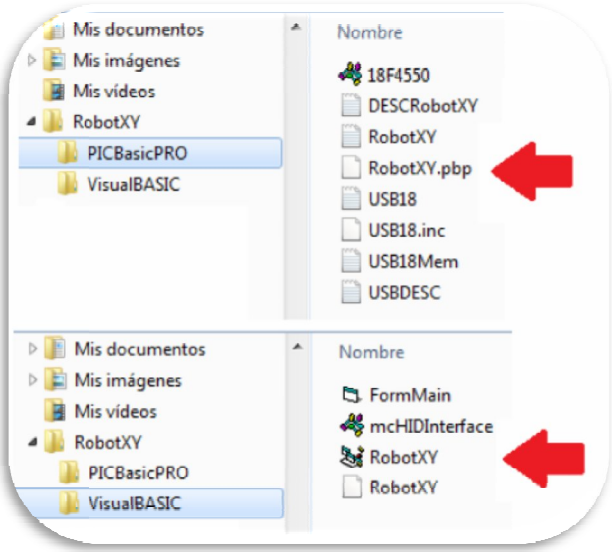


Figura 31. Programas creados con Easy Hid.

En el Pic Basic pro cuando se genera el código, se observan 3 partes importantes: la primera es donde se escribe el código, la segunda es la recepción de datos y la tercera el envió de datos.

```
 DEFINE OSC 48
#DEFINE LOADER_USED 1

USBBufferSizeMax    con 8   ' maximum buffer size
USBBufferSizeTX     con 8   ' input
USBBufferSizeRX     con 8   ' output

' the USB buffer...
USBBuffer         Var Byte[USBBufferSizeMax]
USBBufferCount    Var Byte

' ***************************************************************
' * main program loop - remember, you must keep the USB         *
' * connection alive with a call to USBService every couple     *
' * of milliseconds or so...                                    *
' ***************************************************************
usbinit ' initialise USB...
ProgramStart:
    gosub DoUSBIn
    gosub DoUSBOut
    goto ProgramStart

' ***************************************************************
' * receive data from the USB bus                               *
' ***************************************************************
DoUSBIn:
    USBBufferCount = USBBufferSizeRX              ' RX buffer size
    USBService                                    ' keep connection alive
    USBIn 1, USBBuffer, USBBufferCount, DoUSBIn   ' read data, if available
    return

' ***************************************************************
' * wait for USB interface to attach                            *
' ***************************************************************
DoUSBOut:
    USBBufferCount = USBBufferSizeTX              ' TX buffer size
    USBService                                    ' keep connection alive
    USBOut 1, USBBuffer, USBBufferCount, DoUSBOut ' if bus available, transmit da
    return
```

Figura 32. Codigo generado por el Easy Hid para el Pic Basic Pro.

Entre las especificaciones del USB, previamente se había puesto que se use buffer in y out 8 bytes, esto significa que en el PBP se va a enviar y recibir datos usando 8 espacios, es decir, USB Buffer[0], USB Buffer[1] USB Buffer[7] = 0 ó a 1.

Esto nos indica que cuando el USB Buffer[x] está activado (=1) realice la acción deseada. Por ejemplo:

If USB Buffer [0] = 1 then (botón activado desde el Visual Basic activa)

High PortB.0 (Puerto b.0 se active (se prende un led))

Pause 1000 (se mantiene prendido por 1 segundo)

Low PortB.0 (Puerto b.0 se desactiva (se apaga un led))

Pause 1000 (se mantiene apagado por 1 segundo)

EndIf (finaliza la acción)

Para enviar un dato hacia el visual Basic se configura de la siguiente manera:

portB.1 = USB Buffer[x] (x es el número de buffer usado)

En la figura 33 como se puede ver, está la programación generada por Easy HID en este caso para Visual Basic. Aquí podemos encontrar 8 partes las cuales nos ayudan a mantener primero la conexión viva y segundo el envio y la recepción de datos.

Entre los detalles más importantes que se pueden resaltar está la definición de Vendor ID y del Producto ID, la cantidad de bytes de entrada y de salida que se van a usar, el llamado al driver cuando se carga el programa, e igualmente la finalización de la llamada al driver cuando se desconecta y la lectura - escritura de datos hacia el Pic.

Cuando se escriben datos al Pic, como se había definido 8 bytes de buffer in, en el Visual Basic el conteo no comienza en 0 sino en 1, lo que provoca que lo que en el PBP es 4 en el VB es 5. De tal forma que se tiene BufferOut (1), (2),… BufferOut (8).

Para escribir un dato hacia el Pic desde el visual Basic se usa:

Private Sub dere_MouseDown() (clic en un botón generado en VB)

BufferOut (2) = 1 (USB Buffer [1] en PBP es activado)

HidWriteEx VendorID, ProductID, BufferOut (0) (rutina de detección)

End Sub (finalización del envio)

De la misma forma para leer un dato desde el Pic se usa parte del código generado por el HID, el cual metiendo el buffer deseado a una variable conseguimos que el dato se traído hacia el Visual Basic desde el microcontrolador

'***

' on read event...

'***

Public Sub OnRead(ByVal pHandle As Long)

' read the data (don't forget, pass the whole array)...

If hidRead(pHandle, BufferIn(0)) Then

' ** YOUR CODE HERE **

Temp = BufferIn(8) (Variable igual al dato del microcontrolador)

Tactual.Caption = Temp (variable visible en visual Basic)

' first byte is the report ID, e.g. BufferIn(0)

' the other bytes are the data from the microcontrollller...

End If (fin del IF)

End Sub (Fin de la lectura)

```
' vendor and product IDs
Private Const VendorID = 6017
Private Const ProductID = 2005

' read and write buffers
Private Const BufferInSize = 8
Private Const BufferOutSize = 8
Dim BufferIn(0 To BufferInSize) As Byte
Dim BufferOut(0 To BufferOutSize) As Byte

'*********************************************************************
' when the form loads, connect to the HID controller - pass
' the form window handle so that you can receive notification
' events...
'*********************************************************************
Private Sub Form_Load()
   ' do not remove!
   ConnectToHID (Me.hwnd)
End Sub

'*********************************************************************
' disconnect from the HID controller...
'*********************************************************************
Private Sub Form_Unload(Cancel As Integer)
   DisconnectFromHID
End Sub

'*********************************************************************
' a HID device has been plugged in...
'*********************************************************************
Public Sub OnPlugged(ByVal pHandle As Long)
   If hidGetVendorID(pHandle) = VendorID And hidGetProductID(pHandle) = ProductID Then
      ' ** YOUR CODE HERE **
   End If
End Sub

'*********************************************************************
' a HID device has been unplugged...
'*********************************************************************
Public Sub OnUnplugged(ByVal pHandle As Long)
   If hidGetVendorID(pHandle) = VendorID And hidGetProductID(pHandle) = ProductID Then
      ' ** YOUR CODE HERE **
   End If
End Sub
```

```
'*********************************************************************
' controller changed notification - called
' after ALL HID devices are plugged or unplugged
'*********************************************************************
Public Sub OnChanged()
   Dim DeviceHandle As Long

   ' get the handle of the device we are interested in, then set
   ' its read notify flag to true - this ensures you get a read
   ' notification message when there is some data to read...
   DeviceHandle = hidGetHandle(VendorID, ProductID)
   hidSetReadNotify DeviceHandle, True
End Sub

'*********************************************************************
' on read event...
'*********************************************************************
Public Sub OnRead(ByVal pHandle As Long)

   ' read the data (don't forget, pass the whole array)...
   If hidRead(pHandle, BufferIn(0)) Then
      ' ** YOUR CODE HERE **
      ' first byte is the report ID, e.g. BufferIn(0)
      ' the other bytes are the data from the microcontrolller...
   End If
End Sub

'*********************************************************************
' this is how you write some data...
'*********************************************************************
Public Sub WriteSomeData()
   BufferOut(0) = 0   ' first by is always the report ID
   BufferOut(1) = 10  ' first data item, etc etc

   ' write the data (don't forget, pass the whole array)...
   hidWriteEx VendorID, ProductID, BufferOut(0)
End Sub
```

Figura 33. Codigo generado por Easy Hid para visual basic.

Una vez que se ha entendido el funcionamiento del USB, antes de cualquier modificación al PBP se debe correr y generar el archivo .HEX es el archivo hexadecimal con el cual se programan los microcontroladores.

Una vez programado el Pic, se prosigue a conectar el conector USB a la computadora. Como se observa en la figura 34, el sistema operativo reconoce el dispositivo mediante el driver mcHID.dll. Este driver es instalado en la carpeta del Easy HID el momento en que se instala el programa.

Figura 34. Dispositivo conectado con la computadora.

Finalizada con éxito la conexión, el microcontrolador está listo para ser programado, y poder trabajar con éste ya que la computadora lo reconoció y el plug and play de los dispositivos USB está listo para usarse.

Diseño y construcción del prototipo

La construcción del prototipo, debe ser de tamaño real, es decir, que debe satisfacer las necesidades del mercado. Las placas que se pueden manejar en este prototipo varían entre placas de 5 [cm] x 5 [cm] hasta el tamaño de 15 [cm] x 15 [cm], de tal forma que pueda taladrar en cada punto en esta área.

Dentro del diseño se ha pensado en una caja base, de esta forma se puede poner en el interior la placa y la fuente

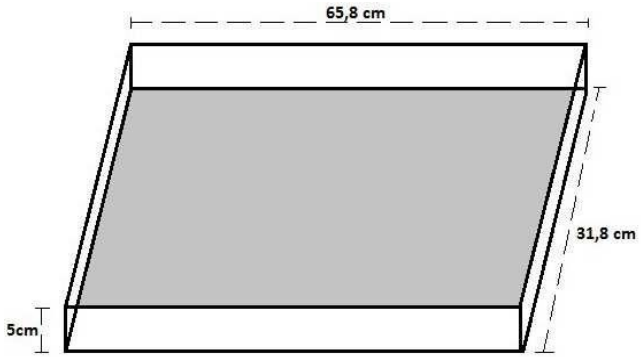


Figura 35: Caja interior para la fuente y placa del controlador

Sobre la caja, va la base la cual tiene 64 [cm] de largo, 30 [cm] de ancho y 5,5 [cm] de alto. En los bordes de la base, se va poner los soportes donde van a estar sostenidas las rieles y el motor paso a paso que va a controlar el eje Y. Los orificios diseñados están a una altura de 2,5 [cm] de esta forma la plataforma movible va a estar en las rieles a 1,5 [cm] de altura para evitar el roce.

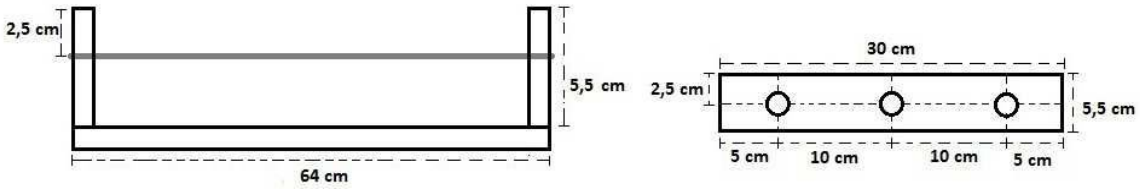


Figura 36: vista lateral y frontal rieles y soportes

La plataforma movible como se observa en la figura 37, es una plataforma que tiene el largo y el ancho para una placa de tamaño 15 [cm] x 15 [cm]. En el caso que la placa sea de menor tamaño, se ha considerado que debe tener un ajuste para que la placa no se mueva.

En la parte inferior de la misma podemos observar que tiene 3 orificios, en los cuales los dos laterales son por donde los rieles van a pasar para mantener equilibrio, y el orificio del medio es el que contiene la tuerca la que va a trasladar el movimiento rotacional de los motores hacia el movimiento transnacional.

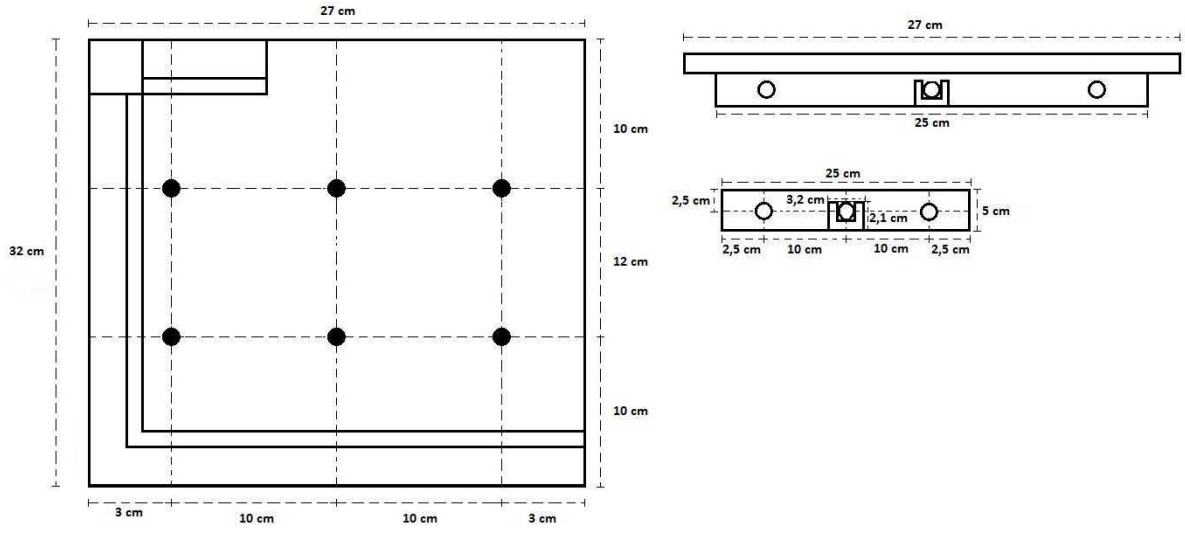


Figura 37: Plataforma movible

En la figura 38 como se observa es la unión de la plataforma con la base, lo que nos permite observar que si la plataforma se mueve hasta el tope contrario, se mueve la mitad de la distancia de la base.

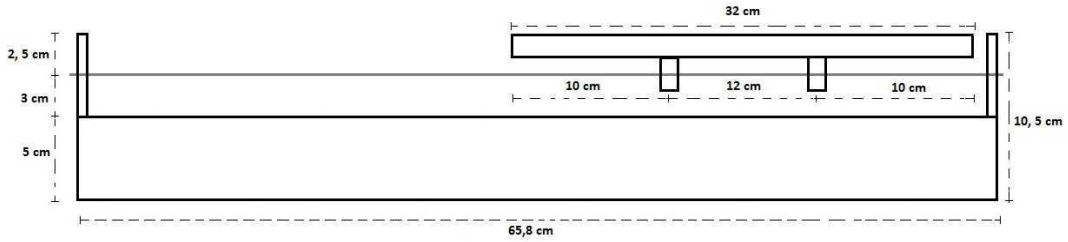


Figura 38: Unión base y plataforma movible

Al igual que la base, se tiene un soporte superior que nos va permitir que el efector final se mueva en la dirección de X. De igual forma como se observa en la figura 39, podemos encontrar que se tienen dos rieles para mantener el equilibrio y al medio el tornillo milimétrico para poder mover traslacionalmente el efector final.

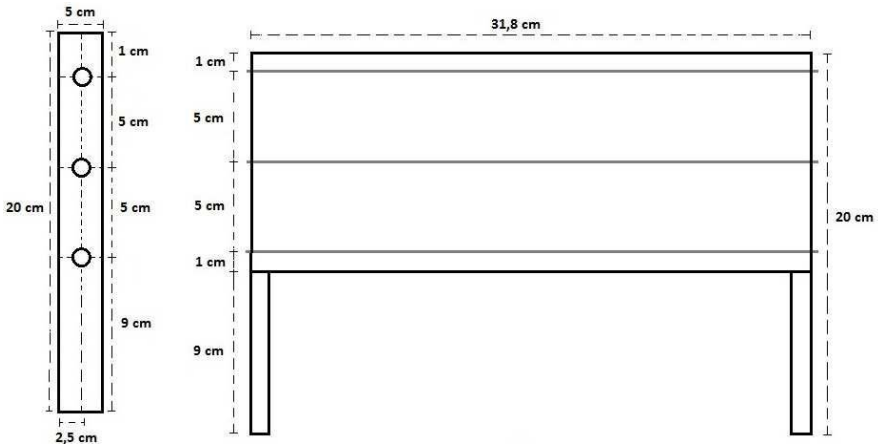


Figura 39: Soporte superior y soporte rieles

Esta sección de la maqueta, tiene por objetivo sostener el motor del eje Z. Este eje nos ayuda a mover hacia arriba y hacia abajo el efector final de la misma manera, podemos observar que para que no pierda estabilidad está compuesto por dos rieles laterales y un tornillo sin fin para que el efector final suba y baje.

Dentro de los detalles de diseño, se pensó que para que el efector final no se demore mucho subiendo y bajando, se diseñó que el motor paso a paso para controlar éste, esté a un cuarto del tamaño de la placa. Por eso podemos observar que existe un hueco por el cual entra el motor paso a paso para que el rotor pueda ir casi al borde del soporte.

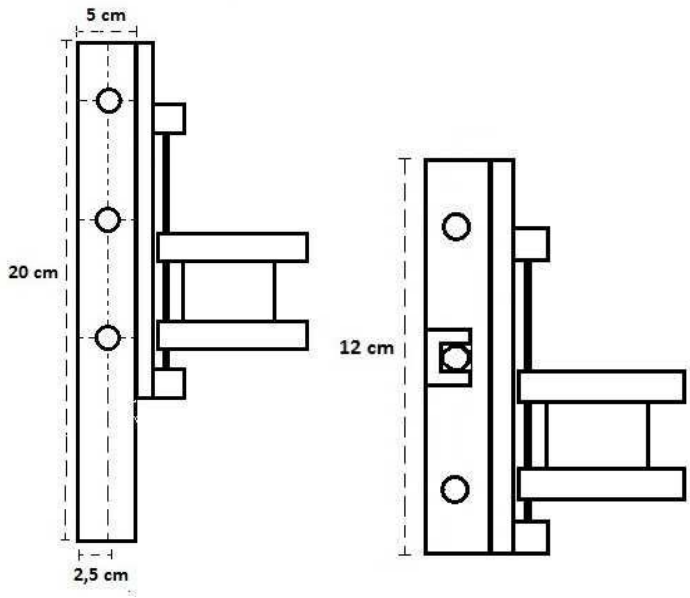


Figura 40: Soporte para el taladro y soporte para rieles superior

Como se observa en las vistas laterales superiores, figura 41, debido al diámetro de nuestro motor DC, el soporte de éste lo ajusta tanto en la parte superior y en la parte inferior, de esta forma, conseguimos que existan dos puntos de sujeción para que cuando haga presión contra la placa no se salga o no se descentre.

De igual forma podemos observar que los orificios por donde van las rieles son la misma distancia obteniendo un movimiento más suave, haciendo que las rieles sirvan para estabilizar más que para guiar.

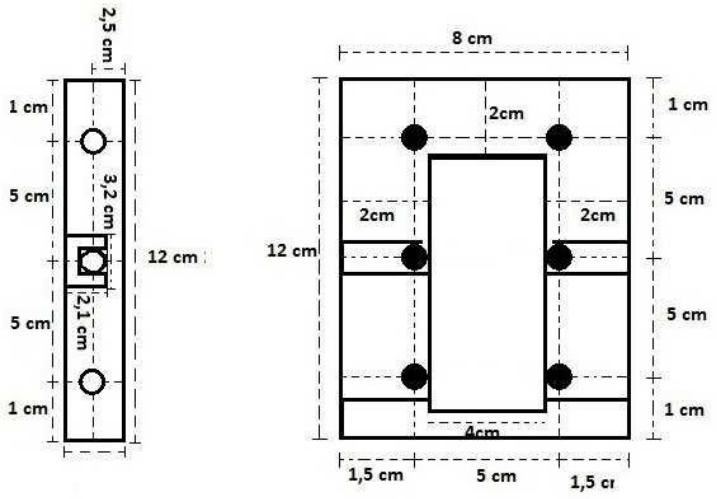


Figura 41: Vista lateral soporte de taladro

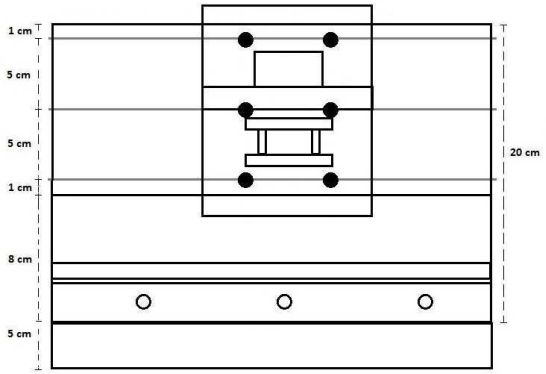


Figura 42: Vista frontal completo

Figura 43: Diseño Real

CAPITULO II

Funcionamiento *(Manual del usuario)*

En este capitulo, se va indicar como es el funcionamiento tanto con el Livewire – PCB Wizzard y el Proteus – Ares y se va a explicar el funcionamiento del control manual.

· Funcionamiento con Ares

· Funcionamiento con Livewire - PCBWizzard

· Control Manual

(*Se recomienda leer el Manual del Usuario que se encuentra en la carpeta de documentos).*

I. FUNCIONAMIENTO CON ARES:

La configuración del Proteus se la realiza en la pantalla principal del generador de circuitos impresos. Como se mencionó antes, el editor de circuitos impresos del Proteus se llama Ares. Antes de comenzar a utilizar el Ares, como se observa en la figura 44, primero se debe determinar el espacio entre cada punto, es decir, que debemos configurar que entre punto y punto exista la distancia necesaria para que no se choquen los caminos.

La distancia mínima que se debe utilizar es de 2,54 [mm] o 0,1 [inches] como norma que se encuentra en la mayoría de editores de circuitos. De esta forma podemos configurar que la misma distancia entre puntos tanto en el Livewire - PCBWizzard como en el Proteus – Ares la cual va a ser la misma, por lo tanto la distancia de movimientos de los dos motores va a ser exactamente como se ha calibrado inicialmente para los dos programas.

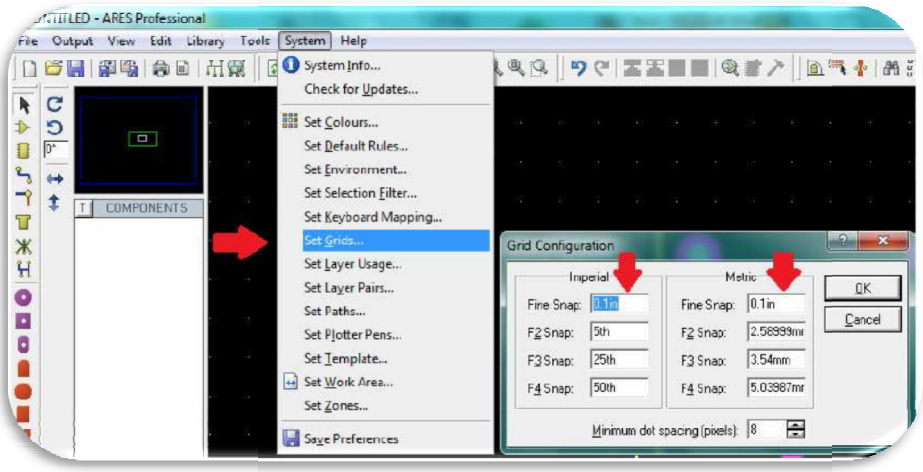


Figura 44. Configuración proteus.

Una vez configurada la distancia entre los puntos, se debe configurar el área de trabajo, esto nos sirve, ya que debemos utilizar un cuadrante para determinar: primero los puntos (0,0) y determinar los puntos (X, Y) para los componentes que se van a insertar. Como se observa en la figura 45, nos indica cómo poner el área de trabajo. Observamos que la flecha superior nos indica que se va a poner un rectángulo de área de trabajo, y de placa de caminos inferior.

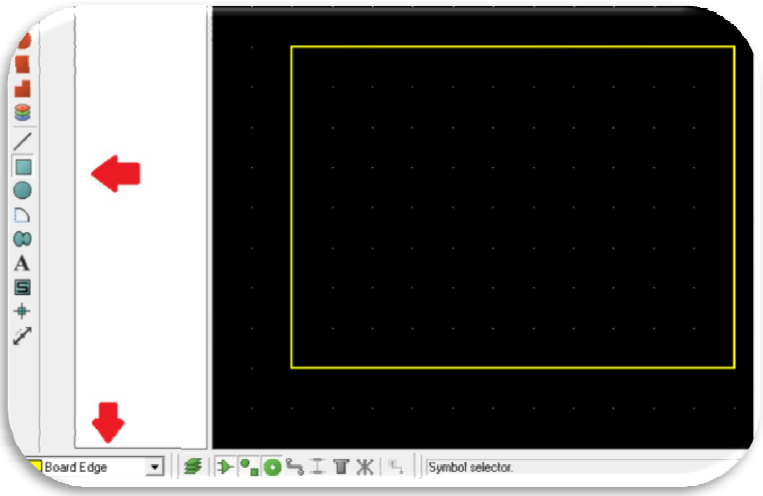


Figura 45. Área de trabajo.

Una vez definida nuestra área de trabajo, debemos determinar el punto de inicio. Ya que se ha configurado nuestro robot para que trabaje desde arriba hacia abajo y se ha determinado que el trabajo a realizar sea en el cuarto cuadrante. Esto significa, que las coordenadas en el eje X son positivas y la coordenadas del eje Y son negativas. Esto significa que va a realizar un barrido de arriba hacia abajo y de izquierda a derecha.

Como se observa en la figura 46, en la opción output/ Set Output Origin y nos da la opción de colocar el punto de origen donde uno desee. De esta forma ponemos nuestro punto de origen en la esquina superior izquierda.

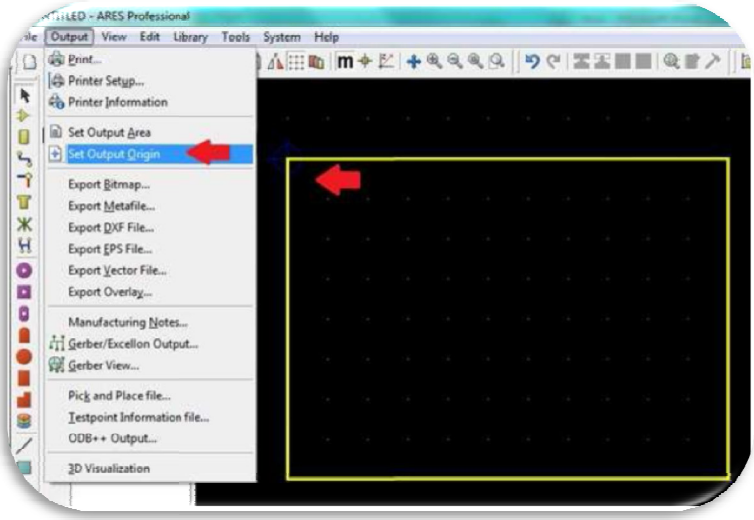


Figura 46. Determinación punto de origen.

Una vez determinado nuestro punto de origen y nuestra área de trabajo podemos insertar los dispositivos eléctricos que previamente usamos en la simulación. Como se observa en la figura 47, podemos observar los dispositivos en posición para poder generar los archivos de puntos XY.

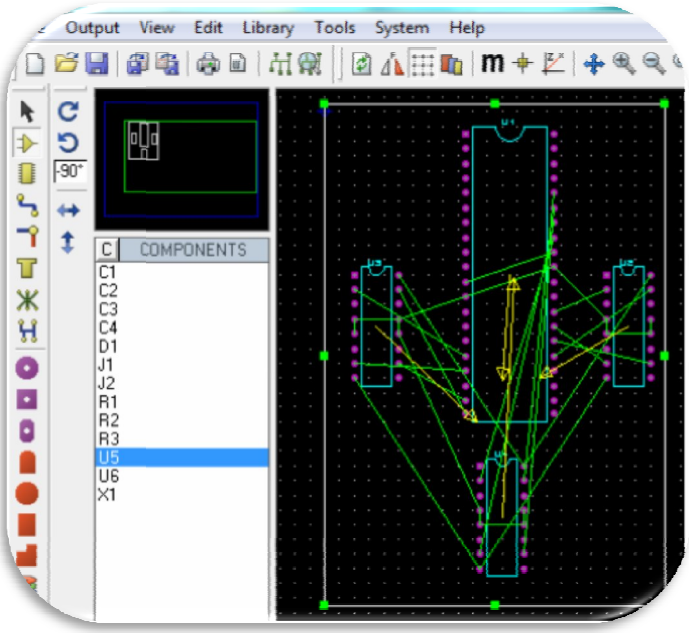


Figura 47. Dispositivos eléctricos situados en la placa.

Cuando se ha finalizado con todos los componentes, se puede generar el autoruouting, el cual nos ayuda a generar automáticamente los caminos. Como se observa en la figura 48, observamos el botón del autoruouting que nos genera los caminos automáticamente.

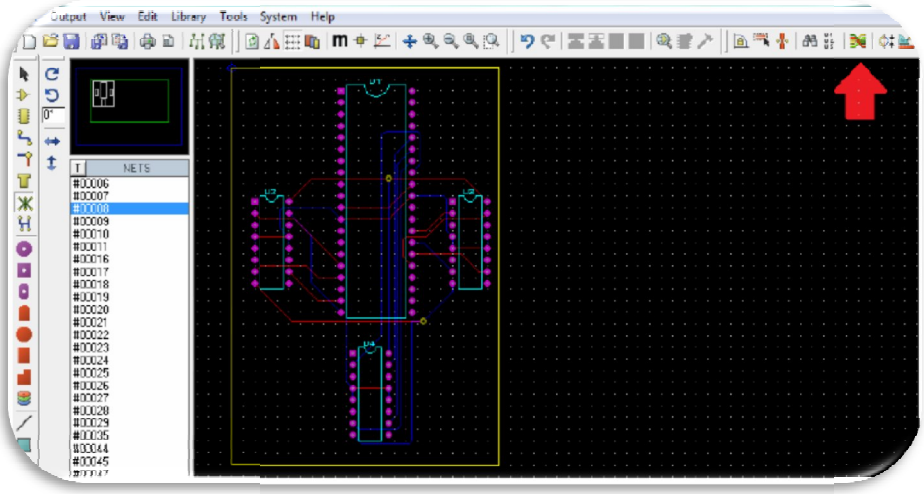


Figura 48. Generacion de caminos para la baquela.

Finalizado este proceso, ya podemos comenzar tanto a imprimir como a modificar la placa.

Una vez aceptado el formato y la calidad, las uniones y todos los detalles de la placa, procedemos a crear el archivo, el cual va a contener toda la información sobre los puntos y los diámetros de los orificios a realizar. Para esto nos vamos a Output/GerberExellon Output, haciendo clic, nos entrega la pantalla que se observa en la figura 49 donde vamos a proceder a poner:

· El nombre del archivo

· El lugar donde se va a guardar el archivo

· Qué tipo de archivo se va a generar, en nuestro caso es un tipo drill

· Las unidades, imperial que nos entrega puntos XY con números en miles, lo que nos permite tener mayor precisión.

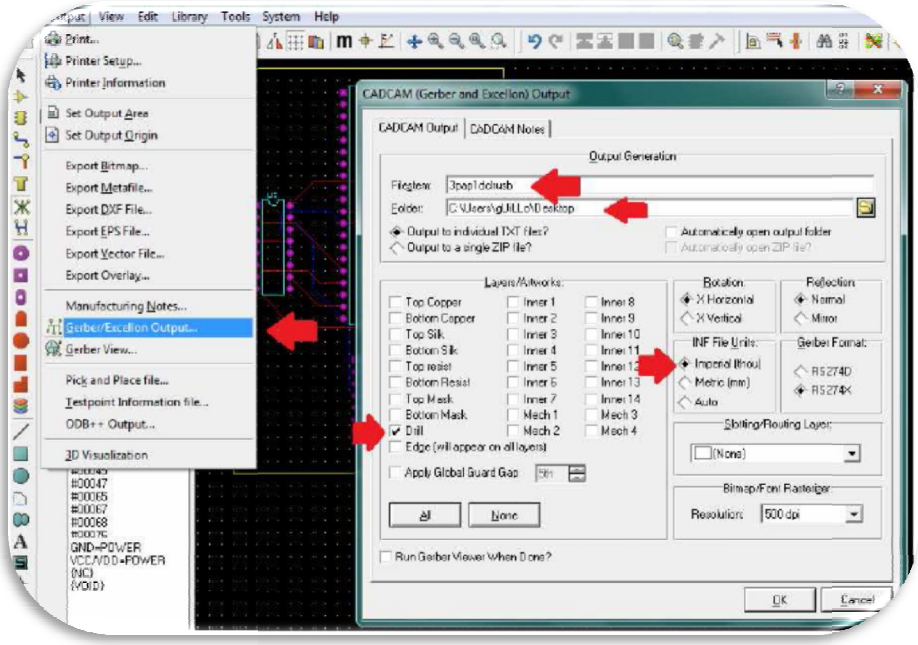


Figura 49. Generación de archivo drill.

Una vez creado el archivo, podemos proceder a abrir el archivo con el block de notas, para poder observar los puntos generados. Como se observa en la figura 50, podemos ver las partes que se indicó previamente sobre este archivo, obteniendo los puntos donde se debe taladrar.

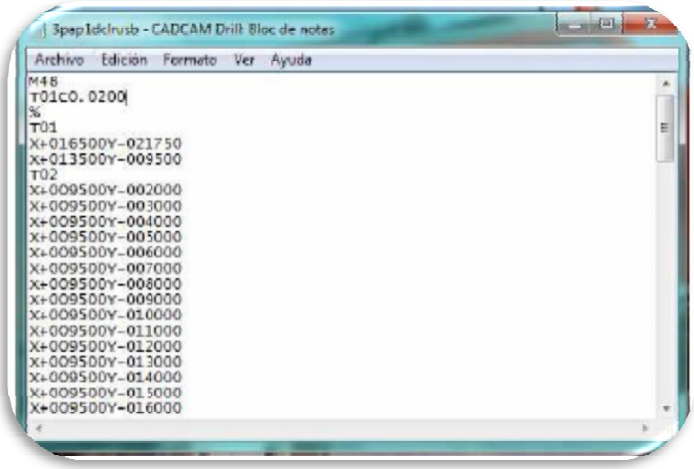


Figura 50. Archivo generado por Proteus- Ares.

Una vez generado el archivo procedemos a abrir la interfaz visual, es decir el programa generado en Visual Basic. En este programa como se mencionó previamente los componentes, podemos proceder a escoger:

- Clic en Cargar desde Proteus

- Buscamos el archivo generado y lo cargamos a la interfaz.

- Verificamos con el Proteus - Ares que los puntos generados con los mismos
- Confirmamos que se encuentra en el Punto (0,0)

II. FUNCIONAMIENTO CON LIVEWIRE – PCB WIZZARD:

De igual manera, como se trató al archivo del Proteus - Ares, podemos tratar al archivo que nos genera el Livewire - PCBWizzard, de tal forma que los dos archivos sean lo más iguales posible. Esto significa, que en preferencia tengan la misma extensión, que cada coordenada tenga la misma cantidad de dígitos, y que sean los mismos puntos.

En el Livewire – PCBWizzard, de igual forma debemos trabajar con las mismas distancias y mismos detalles que el Proteus – Ares. Como se observa en la figura 51, podemos escoger la distancia entre punto y punto.

Si bien recordamos en el Proteus - Ares, se puso que la distancia sea de 0,1 pulgadas, de igual manera, se puede realizar en el Livewire - PCBWizzard. Si hacemos clic en VIEW/GRID_SNAP/0.1IN obtenemos que la distancia va a ser la misma que en el Proteus - Ares, por lo tanto los movimientos de los motores van a ser idénticos para cada archivo.

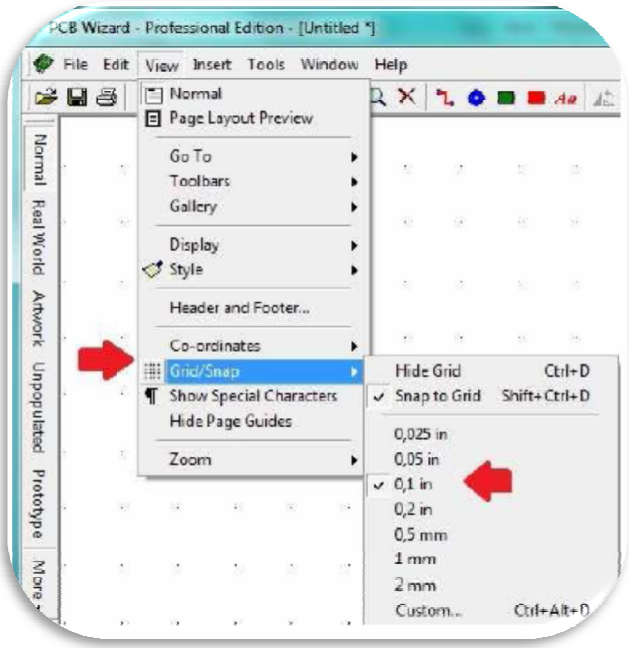


Figura 51. Configuración distancias Livewire – PCB Wizzard.

De igual manera, debemos configurar el área de trabajo, el punto de origen y el sistema métrico que se va a usar. Como se observa en la figura 52, entrando en la configuración de las coordenadas, confirmamos que la distancia entre punto y punto se va a utilizar pulgadas.

El tercer paso nos indica qué icono nos permite generar nuestra área de trabajo. De igual manera que realizamos en Proteus - Ares, debemos generar un rectángulo para que las coordenadas obtenidas sean del tipo XY, es decir, coordenadas rectangulares.

Así mismo entrando en VIEW/CO-ORDINATES/ORIGIN/CHANGE ORIGIN, nos da la opción de cambiar de lugar el punto origen. Como previamente se mencionó, el punto de origen debe utilizar el cuarto cuadrante ya que el diseño mecánico se mueve hacia adelante y hacia la derecha. Esto se consigue utilizando X positivas y Y negativas. Es por esta razón que el punto de origen se lo debe poner en la esquina superior izquierda de nuestra área de trabajo. Estas especificaciones mencionadas se las puede observar más claramente en la figura 52.

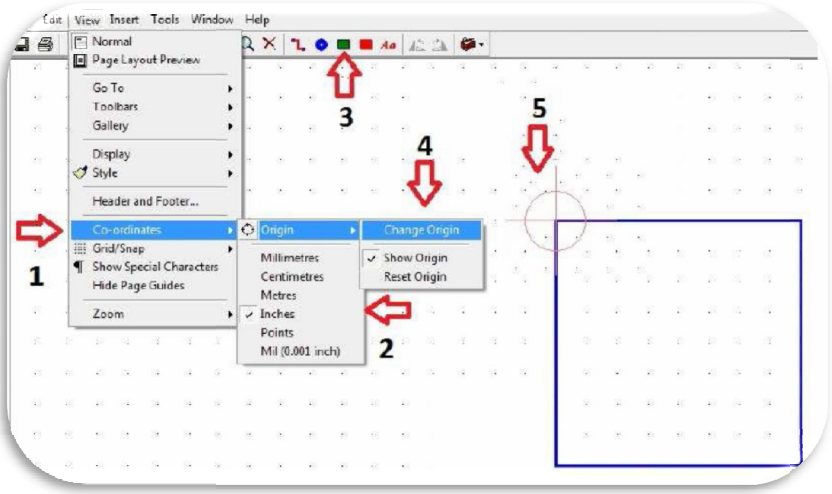


Figura 52. Configuración área de trabajo y origen de coordenadas.

Luego de haber concluido con el área de trabajo y definiendo nuestro punto de inicio, se puede proseguir con el diseño. En esta etapa, insertamos los componentes que se van a usar y colocamos los componentes como se prefiera. Normalmente se trata que la placa sea del tamaño más reducido que se pueda, ya que eso nos permite ahorrar espacio y tiempo ya que si recordamos la tercera etapa de la fabricación de placas PCB, es colocar la placa en ácido férrico para dejar solo los caminos deseados. En el caso que la placa sea más grande se va a demorar más ya que el ácido debe trabajar más sobre el cobre restante.

Una vez que tenemos todo nuestro diseño realizado y estamos conforme con la placa, como se observa en la figura 53.

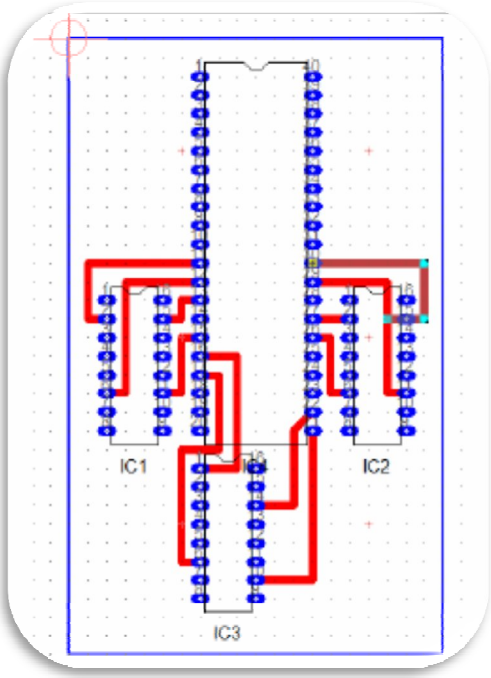


Figura 53. Placa Livewire/PCB Wizzard.

Una vez que tenemos todo en su sitio, podemos proceder a crear el archivo que va a utilizar nuestra interface virtual. Dado que el Livewire/PCBWizzard nos permite manipular los datos que se van a generar, es decir, poder cambiar cuantas unidades y decimales queremos.

III. UTILIZANDO EL CONTROL MANUAL:

El control manual como se mencionó previamente, es una utilidad de la interface gráfica que nos permite mover los motores como el usuario lo desee. El control manual consta de 4 a 5 botones y su objetivo principal es permitir al usuario ubicarse en algún punto requerido y perforar. Esto nos puede servir tanto para cambiar el tamaño de la broca, mejorar o crear algún orificio extra.

CONCLUSIONES

Para concluir nuestro trabajo, primero cabe recordar que, para desarrollar un circuito impreso se puede necesitar mucho tiempo si se hace por el metodo convencional, ademas, la duracion del diseño de un circuito impreso, tambien depende del tamaño del circuito que se este desarrollando.

Gracias a la robotica, este proceso de diseño y desarrollo de un circuito impreso puede ser mas corto, de esta forma generamos un producto que nos ayuda a reducir el tiempo de fabricacion con menor costo y mayor eficiencia.

Como se pudo observar a lo largo de este trabajo, nuestro proyecto logra optimizar la actividad de perforar o abrir los orificios de la baquela sin perder precision ni exactitud, recordando que al ser un producto mecanico, se debe someter a calibracion, limpieza y mantenimiento para un correcto uso y funcionamiento del mismo.

Finalmente, podemos concluir que todos los objetivos planteados en un inicio, fueron completados con gran éxito, ya que gracias a la investigación previamente realizada, se pudo usar el microcontrolador, con la comunicación y la interfaz para automatizar el proceso de perforación de las baquelas.

INFOGRAFÍA

- http://r-luis.xbot.es/index.html
- http://www.unrobotica.com/
- http://cstep.luberth.com/
- http://www.puntoflotante.net/USBPICPROG.htm
- http://www.mikroingenieria.com
- http://www.taringa.net/posts/ciencia-educacion/10632975/Microcode-Studio.html
- http://concurso.cnice.mec.es/cnice2006/material107/operadores/ope_tuerca.htm
- http://www.edaboard.com/thread126795.html
- http://robots-argentina.com.ar/Prueba_PIC628-RS232.htm
- http://www.youtube.com/watch?v=9vEDy8ODKAY
- http://www.telecable.es/personales/jrubi/index.htm?trucos/tip00187.htm
- http://www.winpicprog.co.uk/pic_tutorial1.htm
- http://www.es.wikipedia.org/wiki/Riel
- http://www.freewebs.com/glafebre/tp2550.htm
- http://www.unpocodelectronica.netau.net/mis-primeros-pasos-con-el-18f4550-parte5#picusb

BIBLIOGRAFÍA

- Guillermo Burbano B - Mesa robótica XY para la perforación de placas PCB con conexión USB.
- Datasheet – PIC18F87XA
- Manual ISIS Proteus
- Datasheet Mitsumi M35SP-11HPK
- Microcontrolador PIC 16F84 desarrollo de proyectos
- Datasheet – TIP2293
- Dr. Ing. Laurent Sass – Clases de Robotica 2009. Datasheet – L293 (Puente H)

AGRADECIMIENTOS

Principalmente a Dios y a todas las personas de nuestras familias, las cuales nos apoyaron tanto moral como economicamente durante este arduo y duro proceso.

Damos nuestros sinceros agradecimientos al licenciado R-Luis y al estudiante de electrónica Guillermo Burbano B., cuyas personas nos inspiraron al desarrollo y diseño de nuestro proyecto, siendo ellos de gran ayuda para nosotros en nuestro proceso de investigación.