



UNIVERSIDAD
DE LOS ANDES
MERIDA VENEZUELA

UNIVERSIDAD DE LOS ANDES
FACULTAD DE INGENIERIA
ESCUELA DE INGENIERIA ELECTRICA

DISEÑO E IMPLEMENTACION DE TARJETA DE CONTROL
PARA UN BRAZO ROBOTICO CS-113, QUE SE CONTROLE
VIA USB

Br. Adolfo José Martínez Sifuentes

Mérida, Junio 2019

Reconocimiento-No comercial- Compartir igual



UNIVERSIDAD
DE LOS ANDES
MERIDA VENEZUELA

UNIVERSIDAD DE LOS ANDES
FACULTAD DE INGENIERIA
ESCUELA DE INGENIERIA ELECTRICA

DISEÑO E IMPLEMENTACION DE TARJETA DE CONTROL
PARA UN BRAZO ROBOTICO CS-113, QUE SE CONTROLE
VIA USB

Trabajo de Grado presentado como requisito parcial para optar al título de Ingeniero
Electricista

Br. Adolfo José Martínez Sifuentes.
Tutor (es): Prof. Luis R. Araujo R.

Mérida, Junio, 2019



UNIVERSIDAD DE LOS ANDES
FACULTAD DE INGENIERIA
ESCUELA DE INGENIERIA ELECTRICA

DISEÑO E IMPLEMENTACION DE TARJETA DE CONTROL
PARA UN BRAZO ROBOTICO CS-113, QUE SE CONTROLE
VIA USB

Br. Adolfo José Martínez Sifuentes

Trabajo de Grado, presentado en cumplimiento parcial de los requisitos exigidos para optar al título de Ingeniero Electricista, aprobado en nombre de la Universidad de Los Andes por el siguiente Jurado.

Ing. Antonio Jose Pérez Zambrano

Ing. Arlex Joel Caliz Marquez

MSc. Luis Ramón Araujo Rangel

Br. Adolfo José Martínez Sifuentes. Diseño e implementación de tarjeta de control para un brazo robótico CS-113, que se controle vía USB. Universidad de Los Andes.
Tutor (es): Prof. Luis R. Araujo R.

Resumen

Un brazo robótico es básicamente un dispositivo diseñado con el fin de realizar trabajos de alta precisión y reduciendo tiempos de ejecución, a esto se le suma la ventaja de realizar trabajos peligrosos donde se podría poner en riesgo la vida de los trabajadores de una empresa, hoy en día también se utilizan en el mundo de la medicina, logrando realizar operaciones donde el cirujano se encuentra a kilómetros de distancia, esto se puede realizar gracias a la precisión y el control que se logran con los motores de pasos. Por lo general los brazos robóticos son utilizados ampliamente en líneas de ensamblaje, teniendo como tarea principal realizar soldaduras, pintar, verificar errores en el producto final, etc. En este trabajo se presenta un prototipo de tarjeta de control para un brazo robótico CS-113 el cual es un robot de uso didáctico y se encuentra disponible en el laboratorio de electrónica de la Universidad de Los Andes, este brazo sirve de mucha ayuda para explicar a los estudiantes de ingeniería eléctrica como es el funcionamiento de estos equipos. En este trabajo se detalla cuáles fueron los componentes utilizados y pasos a seguir para crear un prototipo de tarjeta de control para dicho brazo.

Descriptores: Brazo robótico, USB, prototipo.

INDICE GENERAL

APROBACIÓN	i
RESUMEN.....	ii
INTRODUCCION.....	1

Capitulo	pp.
1. EL PROBLEMA.....	3
1.1 Planteamiento del Problema.....	3
1.2 Justificación.....	4
1.3 Objetivos.....	4
1.3.1 Objetivos generales.....	5
1.3.2 Objetivos específicos.....	5
1.4 Metodología.....	6
1.5 Alcance.....	6
1.6 Limitaciones.....	7
2. MARCO TEORICO.....	8
2.1 HARDWARE.....	9
2.1.1 Microcontrolador PIC.....	9
2.1.2 Motor Paso a Paso.....	10
• El Motor de Pasos de Reluctancia Variable (VR).....	11
• El Motor de Pasos de Rotor de Imán Permanente.....	12
• El Motor de Pasos Híbrido.....	12
• Motor de Pasos Unipolares.....	14
• Motor de Pasos Bipolares.....	14

2.1.3 Puente H.....	16
2.2 SOFTWARE	18
2.2.1 Compilador MPLAB XC8.....	19
2.2.2 IDE MPLAB X	19
2.2.3 IDE QT Creator	21
2.2.4. Proteus Design Suite.....	22
• ISIS.....	23
• ARES.....	23
• Modulo VSM.....	25
3. DISEÑO DEL PROTOTIPO.....	26
3.1 Diseño del Hardware.....	26
3.2 Diseño de tarjeta de control.....	29
3.3 Conexión USB.....	32
3.4 Diseño de Conexión PIC a Puentes H.....	33
3.5 Diseño del Software.....	37
3.5.1 Software del Microcontrolador PIC.....	37
3.5.2 Software del Computador.....	40
4. PRUEBAS DE FUNCIONAMIENTO.....	42
4.1 Análisis del Software del Computador.....	42
4.2 Pasos Para Puesta en Funcionamiento.....	45
4.3 Prueba De Funcionamiento.....	47
4.4 Recomendaciones de uso.....	52
CONCLUSIONES.....	53
RECOMENDACIONES.....	55
REFERENCIAS.....	56
ANEXOS	58

ÍNDICE DE FIGURAS

Figura	pp.
1.1 Tarjeta de Control Original del Brazo Robótico (Z80).....	4
2.1 Estructura Interna de Microcontrolador.....	10
2.2 Motor de Pasos de Reluctancia Variable.....	11
2.3 Motor de Pasos de Rotor con Imán Permanente.....	12
2.4 Motor de Pasos Híbrido.....	13
2.5 Rotor de un Motor de Pasos Híbrido.....	13
2.6 Esquema de Motor de Paso Unipolar.....	14
2.7 Esquema de Motor de Pasos Bipolar.....	15
2.8 Estructura Básica de Puente H.....	16
2.9 Diagrama Interno del Puente H (L298KV).....	17
2.10 Encapsulado del Puente H (L298KV).....	18
2.11. IDE MPLAB X.....	21
2.12. IDE QT Creator 4.6.2.....	22
2.13. Proteus Design Suite 8.0.....	23
2.14 ISIS.....	24
2.15 ARES.....	24
3.1 Diagrama de Bloques del Prototipo.....	28
3.2 Disposición de los Componentes del Prototipo.....	29
3.3 Puerto USB Tipo B.....	30
3.4 Puerto de Alimentación (12V).....	30
3.5 Conexiones Puerto USB.....	32
3.6 Diagrama de Conexiones Para los Motores de Paso de la Muñeca.....	33
3.7 Esquema de Conexiones de los Puentes H con el Microcontrolador.....	34
3.8 Diagrama Del Programa (Microcontrolador).....	37
3.9 Ejemplo de Cadena Enviado Desde Computador.....	39

3.10 Interface de Usuario.....	40
3.11 Partes y Dimensiones del Brazo Robótico CS-113.....	41
4.1 Interface de Usuario Detallada.....	42
4.2 Interface de usuario (Modo Consola).....	44
4.3 Administrador de Dispositivos de Windows.....	45
4.4 Dispositivos e Impresoras.....	46
4.5 Lista de Dispositivos USB (COM).....	46
4.6 Circuito en Protoboard.....	47
4.7 Fuente de Poder.....	48
5.1 Diagrama de conexiones de la muñeca.....	58
5.2 Diagrama de conexiones de la base.....	59
5.3 Diagrama de conexiones del hombro.....	59
5.4 Diagrama de conexiones de la codo.....	60
5.5 Diagrama de conexiones de la pinza.....	60

www.bdigital.ula.ve

INDICE DE TABLAS

Tabla	pp.
2.1 Secuencia de Activación de Bobinas	15
2.2 Condiciones de operación Optima del puente H (L298KV)	17
2.3 Tabla de la Verdad (L298KV).....	18
3.1 Componentes Utilizados Para el Desarrollo Del Prototipo.....	27
3.2 Secuencia de Activación Motor de pasos de la Base (Sentido horario).	35
3.3 Secuencia de Activación Motor de pasos del Hombro (Arriba).....	35
3.4 Secuencia de Activación Motor de pasos del codo (Arriba).....	35
3.5 Secuencia de Activación Motor de pasos de la muñeca (Sentido Horario)	36
3.6 Secuencia de Activación Motor de pasos de la Muñeca (Arriba)	36
3.7 Secuencia de Activación Motor de pasos de la pinza (Abrir)	36
3.8 Lista de Numero de Pasos Máximos	39

INTRODUCCION

Debido a la actual tendencia de automatizar los procesos industriales, es necesario emplear brazos robóticos para realizar labores donde no se tiene un fácil acceso, además de tener alta precisión y reducir drásticamente los tiempos de trabajo en las líneas de ensamblajes.

El propósito de este trabajo es el desarrollo de un prototipo de hardware y software basado en un microcontrolador PIC, esto con la finalidad de reemplazar la tarjeta de control original del brazo robótico CS-113, la tarjeta original está basada en un microcontrolador Z80, y utiliza una comunicación por puerto paralelo (LPT1), la cual ya se encuentra obsoleta, por lo cual es necesario que este prototipo deba comunicarse con un computador vía USB, ya que esta es la comunicación que hoy en día poseen la mayoría de los computadores.

En el computador se debe desarrollar una interfaz de usuario para controlar el brazo robótico, dicha interface está desarrollada en un software libre llamado QT Creator versión 4.6.2. con esta propuesta se podrá contar con una interface la cual tendrá control sobre todos los movimientos de este brazo robótico, dicha interface de usuario es muy cómoda sencilla e intuitiva, para de esta manera lograr un control sencillo, funcional y total de este robot.

La tarjeta de control anterior no poseía una interface de usuario en el computador para su control, por lo tanto, se debía hacer mediante un terminal que le enviaba la información por el puerto paralelo (LPT1), al realizar una interface de usuario se facilita el control del brazo robótico lo cual es una mejora importante.

En este trabajo se explica el procedimiento para realizar la tarjeta de control del brazo robótico, cuáles son los componentes esenciales para la construcción y como ponerlo en funcionamiento, todo esto con la finalidad de que cualquier persona pueda utilizar este brazo robótico.

En el capítulo I de este trabajo se explica el planteamiento del problema, la justificación, limitaciones, alcance y los objetivos a desarrollar, el capítulo II consta del marco teórico,

en este se explican los conceptos básicos para entender este trabajo de mejor manera, el capítulo III describe el diseño del prototipo así como la implementación del hardware y desarrollo del software, se describen las conexiones y componentes que se utilizaron para el desarrollo del prototipo, también se muestra como es el protocolo de comunicación y se explica, cómo son las cadenas de caracteres enviadas desde el computador hasta el microcontrolador PIC, en el capítulo IV se muestran los resultados de las pruebas realizadas al prototipo, también se explica cómo poner el prototipo en funcionamiento, las recomendaciones de uso y se detalla cómo es la interface de usuario, que función tiene las diferentes partes de la interfaz, luego de este capítulo se exponen las conclusiones y recomendaciones para mejorar el prototipo.

www.bdigital.ula.ve

CAPÍTULO I

EL PROBLEMA

En el presente capítulo se da a conocer el planteamiento del problema, la justificación, los objetivos generales y específicos, limitaciones, metodología, alcance, aspectos que indican la necesidad, la importancia y lo previsto para resolver lo planteado.

1.1 PLANTEAMIENTO DEL PROBLEMA

La automatización de los procesos industriales es una tendencia que se está empleando cada vez más, debido a que logra mejorar el funcionamiento de estos procesos le añade confiabilidad y precisión, logrando disminuir el tiempo de ejecución y empleando menos recursos humanos, lo cual significa una ventaja enorme a la hora de realizar trabajos donde existen grandes riesgos y un corto tiempo de operación.

Hoy en día es necesario que los estudiantes de la escuela de ingeniería eléctrica de la universidad de los andes, cuente con un prototipo de un brazo robótico para comprender su funcionamiento, y de esta manera poder enfrentar los problemas futuros que se puedan presentar en una empresa, en la cual exista una línea de ensamblaje automatizada, lo cual es muy común conseguir hoy en día en las industrias.

El brazo robótico disponible en el laboratorio de electrónica de la Universidad de Los Andes es el CS-113 Service-Arm, este brazo robótico posee una tarjeta de control la cual utiliza una conexión de datos obsoleta, dicha conexión es por puerto paralelo, esta conexión ya no la poseen los computadores actuales, por lo tanto, los brazos robóticos disponibles en el laboratorio de electrónica fueron descontinuados.

1.2 JUSTIFICACION

El robot CS-113 Service-Arm es un brazo robótico diseñado para simular un robot manipulador industrial. Se trata de un brazo articulado con cinco grados de libertad y una pinza como herramienta terminal. Las articulaciones y la pinza se mueven con seis motores paso a paso de 12V. el robot cuenta con una tarjeta de control, basada en un microprocesador Z80. El control se lleva a cabo conectándolo a un computador a través del puerto paralelo.

Los computadores actuales ya no cuentan con un puerto paralelo y por esta razón existe la necesidad de diseñar una nueva tarjeta de control, que permita llevar a cabo el control a través de un puerto serial universal USB, y utilizando microcontroladores PIC.



Figura 1.1. Tarjeta de Control Original del Brazo Robótico (Z80). [1]

1.3 OBJETIVOS

Seguidamente se presentan los objetivos para darle una solución al problema planteado.

1.3.1 Objetivo General

- Diseñar e implementar una tarjeta de control, que reemplazará la tarjeta actual existente en el interior del brazo robótico CS-113, y que permita controlarse desde un computador, vía puerto serial universal USB, utilizando un microcontrolador PIC.
- Crear las librerías necesarias para controlar desde un computador todas las actividades de control del brazo robótico.

1.3.2 Objetivos Específicos

- Realizar una inspección de la tarjeta de control actual, del brazo robótico, y determinar que componentes son reusables para el nuevo diseño.
- Investigar los microcontroladores PIC que poseen puerto serial universal USB, y seleccionar uno que esté disponible en el mercado local.
- Diseñar la conexión entre el microcontrolador PIC y los motores de paso del brazo robótico, a través de puentes H necesarios para tal fin.
- Diseñar una aplicación para el microcontrolador PIC, que permita llevar a cabo el control del brazo robótico, a través de un computador conectado a través de puerto serial universal USB.
- Programar unas librerías, que los usuarios del brazo robótico, puedan utilizar, para controlarlo desde un computador.
- Realizar una documentación detallada del trabajo.

1.4 METODOLOGIA

La metodología para la realización de este trabajo, consta de la investigación bibliográfica acerca de la comunicación vía USB entre un microcontrolador PIC y un computador, además de la investigación acerca del manejo y control de los motores de pasos disponibles en el brazo robótico, y los algoritmos necesarios para llevar a cabo el control, también se realizará una investigación bibliográfica de trabajos de grados previos en el que se empleó el brazo robótico CS-113.

Seguido de esto, la implementación en el lenguaje de programación C para el microcontrolador PIC, utilizando las herramientas de Microchip, que permiten realizar el control del brazo robótico, desde un computador mediante puerto serial universal USB, también se debe hacer una implementación en lenguaje C++ en el computador, para realizar la interface de usuario y con esta controlar el brazo robótico, dicha interface debe ser capaz de comunicarse con el microcontrolador PIC por vía USB.

1.5 ALCANCE

El siguiente trabajo abarcará el estudio del comportamiento de los motores de paso disponibles en el brazo robótico CS-113, así como el diseño de una tarjeta capaz de ejercer control sobre dichos motores y establecer una comunicación con un computador mediante puerto serial universal (USB).

También se desarrollará el software del microcontrolador PIC, encargado de interpretar y ejecutar movimientos dictados por el computador, a esto sumamos el desarrollo de una interface de usuario que se podrá ejecutar en Windows, para de esta manera lograr un control completo de todas las articulaciones de dicho robot.

Se dejará el código fuente de la interface de usuario y del software del microcontrolador PIC por si se desea posteriormente optimizar estos códigos.

Estarán disponibles las librerías del microcontrolador PIC, los esquemas de conexiones, la lista de componentes y las librerías de la interface de usuario, esto con la finalidad de que, si

se presentase alguna falla, exista una documentación detallada para conseguir una solución viable, también esta información se puede utilizar en caso de que, se desee implementar en circuito impreso.

1.6LIMITACIONES

La principal limitación para el desarrollo de este trabajo radica en el hecho de no poder conseguir de manera sencilla, todos los componentes necesarios para la implementación de una tarjeta de control, aparte del alto costo de dichos componentes electrónicos.

Muchos de los componentes utilizados para las pruebas fueron suministrados por la Universidad de Los Andes, lo cual fue una ventaja, pero a su vez existe la limitación de soldarlos a una placa debido a que se deben devolver al taller, también se utilizaron componentes reciclados de otros dispositivos los cuales se encontraban dañados.

Otra de las grandes limitaciones que se presentaron en el desarrollo de este trabajo fueron las constantes fallas en el suministro de energía eléctrica, debido a estas fallas hubo un atraso significativo en el desarrollo del prototipo, ya que no se podían hacer las pruebas necesarias.

Otra de las limitaciones importantes fue el hecho de no poseer una licencia para el Proteus Design Suite, esto nos limitó debido a que el programa en versión gratuita nos permite realizar pruebas mas no se pueden guardar las simulaciones, por lo tanto, cada vez que se deban realizar pruebas se tiene que montar el circuito completo desde cero.

CAPITULO 2

MARCO TEORICO

El brazo robótico CS-113 producido por la empresa Service Arm, es de uso didáctico, por esta razón la Universidad de Los Andes adquirió estas unidades, pero con el tiempo, el puerto de comunicación del mismo quedó obsoleto porque los computadores actuales dejaron de incluir un puerto paralelo, pero el brazo sigue siendo de gran utilidad por sus múltiples aplicaciones, como muestra se tiene lo expuesto por Pallejà T, Teixidó M y otros (2013), titulado “Desarrollo de un Caso Práctico de Aprendizaje Combinando Visión Artificial y un Brazo Robot” [2] donde muestra el uso del brazo CS-113 en combinación con técnicas modernas de visión por computadora . Otro ejemplo de aplicación de este brazo es el trabajo realizado por Hernández, J (2015), titulado “Adaptación del brazo robótico CS-113 para jugar ajedrez” [3], muestra cómo se controla el brazo por comunicación USB, y la interacción con una GUI realizada en QT-Creator, teniendo como resultado un robot con una interfaz de comunicación moderna y una adaptación capaz de actuar como contrincante de una partida de ajedrez. Con respecto al diseño se hace necesario actualizar el robot con una nueva interfaz basada en un microcontrolador PIC, existen trabajos previos como el trabajo presentado por Córdova, J, y Stalin, P. (2009), titulado “Diseño y construcción de un brazo robótico a base de un PIC, para transportar objetos entre dos puntos fijos, e implementación de un sistema HMI para su monitoreo” [4] donde se expone una arquitectura basada en PIC para controlar un brazo robótico, con la finalidad de aprendizaje y motivar a la investigación a los estudiantes del Instituto Tecnológico Ramón Barba Naranjo de Ecuador. Basado en esto, se propone un nuevo diseño que permita adaptar el hardware del robot CS-113 a un microcontrolador 18F4550, el cual posee interfaz USB.

Con el fin de comprender de manera más clara lo expuesto en los próximos capítulos, a continuación, se explicarán los conceptos básicos referentes al trabajo presentado.

2.1 HARDWARE

En esta sección se da una breve explicación de los dispositivos utilizados en este trabajo, entre estos dispositivos se encuentran los microcontroladores PIC, motores paso a paso y los puentes H.

2.1.1 MICROCONTROLADOR

Un microcontrolador PIC (Peripheral Interface Controller), es un circuito integrado el cual consta de miles de transistores, este arreglo de transistores es llamado microprocesador; las primeras microcomputadoras se fabricaron al añadir periféricos externos a los microprocesadores, tales como memorias, líneas de entradas y salidas, temporizadores entre otros.

La necesidad posterior de integrar todos estos componentes en un solo circuito, permitió crear el primer circuito integrado el cual contiene el procesador y los periféricos en un solo chip, de esta manera nació la primera microcomputadora y fue denominada microcontrolador. [5]

Un microcontrolador PIC es un circuito el cual se puede programar, es capaz de ejecutar ordenes que se encuentren grabadas en una memoria, está constituido de varios bloques funcionales, los cuales cumplen una tarea específica. Todo microcontrolador PIC posee una memoria EEPROM, (Electrically Erasable programmable Read Only Memory), memoria de solo lectura programable y borrable eléctricamente, dicha memoria puede ser programada, borrada y reprogramada eléctricamente; cuando un microcontrolador PIC es fabricado, no contiene datos en la memoria EEPROM, para que pueda controlar algún proceso es necesario, crear un programa y grabarlo en esta memoria, este programa puede estar escrito en lenguaje ensamblador u otro lenguaje para microcontroladores PIC. [7]

En un microcontrolador se encuentran los 3 elementos principales de un computador los cuales son: unidad centra de procesamiento (CPU), memoria (RAM, ROM) y periféricos de entrada y salida. [6]

La estructura interna de un microcontrolador PIC se muestra en la figura 2.1, todos estos elementos mostrados se encuentran en un solo circuito integrado, a este circuito se le puede llamar con toda propiedad microcomputadora.

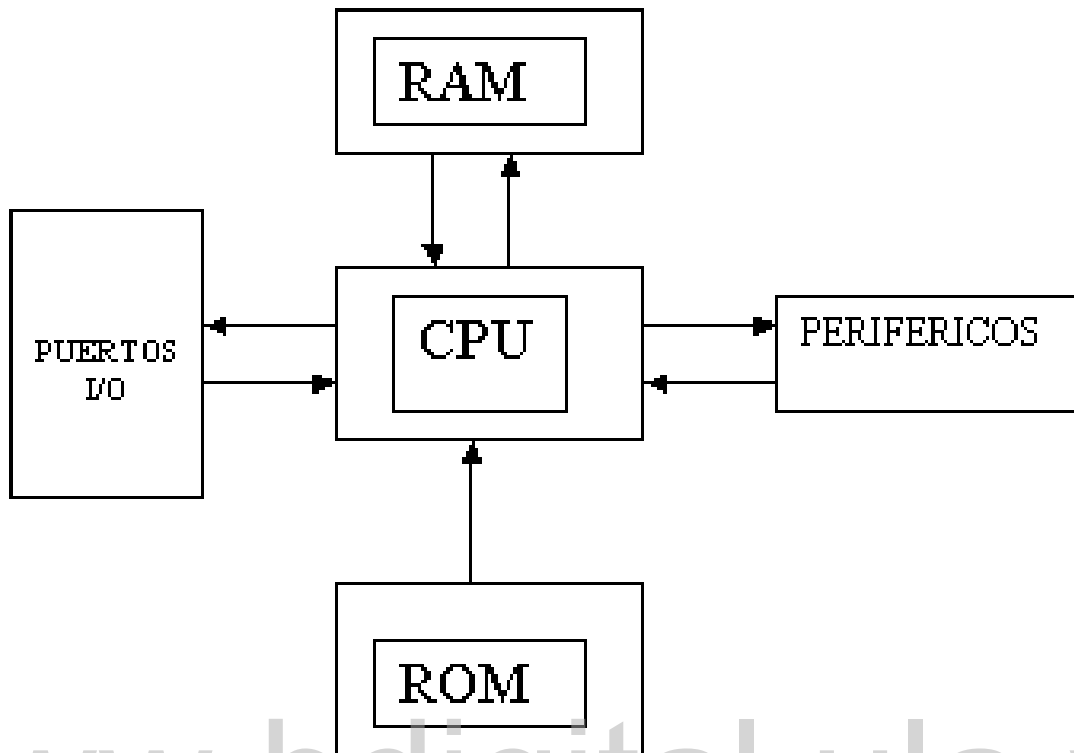


Figura 2.1. Estructura interna de un microcontrolador. [7]

2.1.2 MOTOR PASO A PASO

El motor paso a paso, es un dispositivo electromecánico que convierte una serie de impulsos eléctricos en desplazamientos angulares, dichos ángulos pueden variar dependiendo del orden de activación de las bobinas; esto significa que es capaz de girar una cantidad de grados dependiendo de sus entradas de control.

Estos motores pueden ser gobernados por impulsos procedentes de un microprocesador o de cualquier sistema digital, este motor tiene las ventajas de tener precisión y repetitividad en cuanto a posicionamiento, por lo que es ideal para implementar en brazos robóticos, donde se debe tener un control preciso para que la tarjeta de control sepa en qué posición se encuentra el brazo robótico. [8]

Existen varios tipos de motores de paso según su configuración interna, entre estos tenemos:

- **El Motor de Pasos de Reluctancia Variable (VR)**

Este motor consta con un rotor de hierro de múltiples polos y un estátor devanado, opcionalmente laminado. Rota cuando los dientes más cercanos del rotor son atraídos por las bobinas del estátor, que se encuentran energizadas (obteniéndose, por lo tanto, la ruta de menor reluctancia). La respuesta de este motor es muy rápida, pero la inercia bajo carga es pequeña. Cuando los devanados no están energizados, el par estático de este tipo de motor es cero.

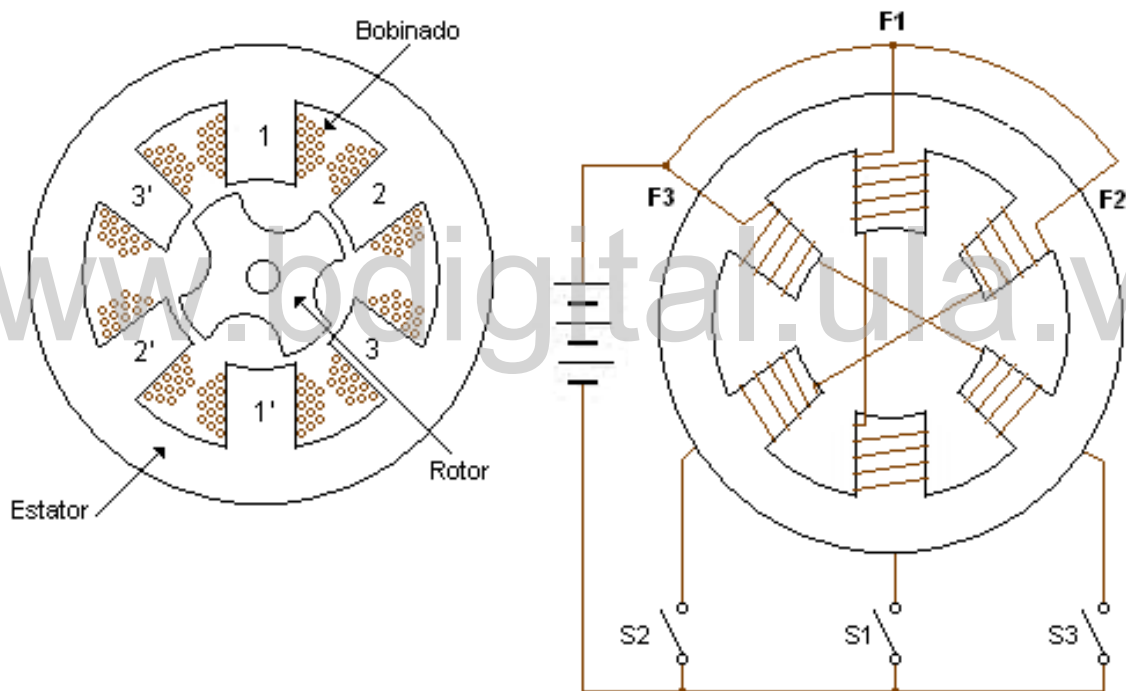


Figura 2.2 Motor de pasos de reluctancia variable. [9]

- **El Motor de Pasos de Rotor de Imán Permanente**

Este motor de pasos logra mantener un par diferente de cero cuando las bobinas del motor no están energizadas. Dependiendo de la construcción del motor, es típicamente posible

obtener pasos angulares de 7,5; 11,25; 15; 18; 45 o 90°. El ángulo de rotación se determina por el número de polos en el estator.

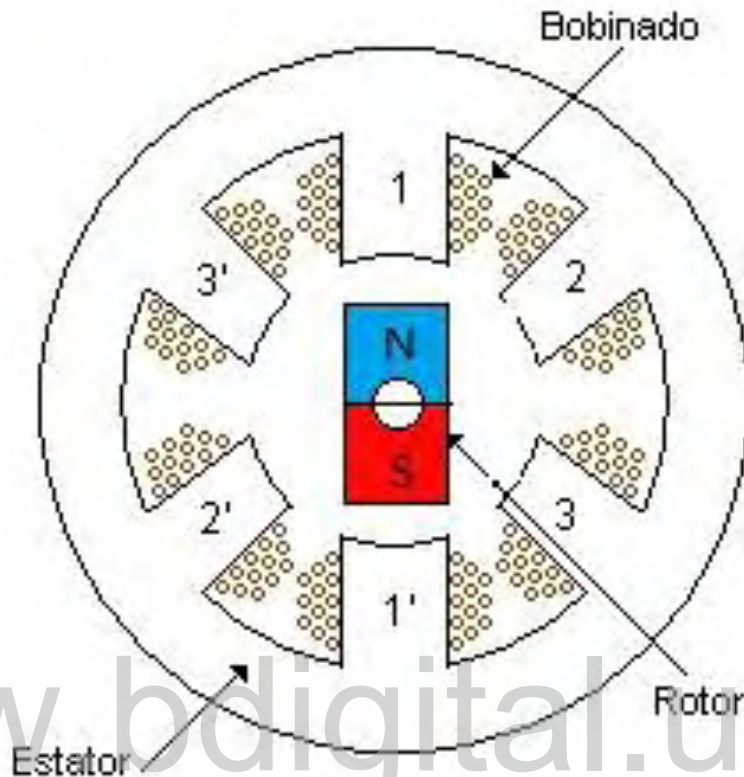


Figura 2.3. Motor de Pasos de Rotor con Imán Permanente. [9]

- **El Motor de Pasos Híbrido**

Este motor se caracteriza por tener varios dientes en el estator y en el rotor, el rotor posee un imán concéntrico magnetizado axialmente alrededor de su eje. Se puede ver que esta configuración es una mezcla de los 2 tipos de motores de paso, los de reluctancia variable y los de imán permanente. Este tipo de motor tiene una alta precisión y alto par, se puede configurar para suministrar un paso angular tan pequeño como 1,8°.

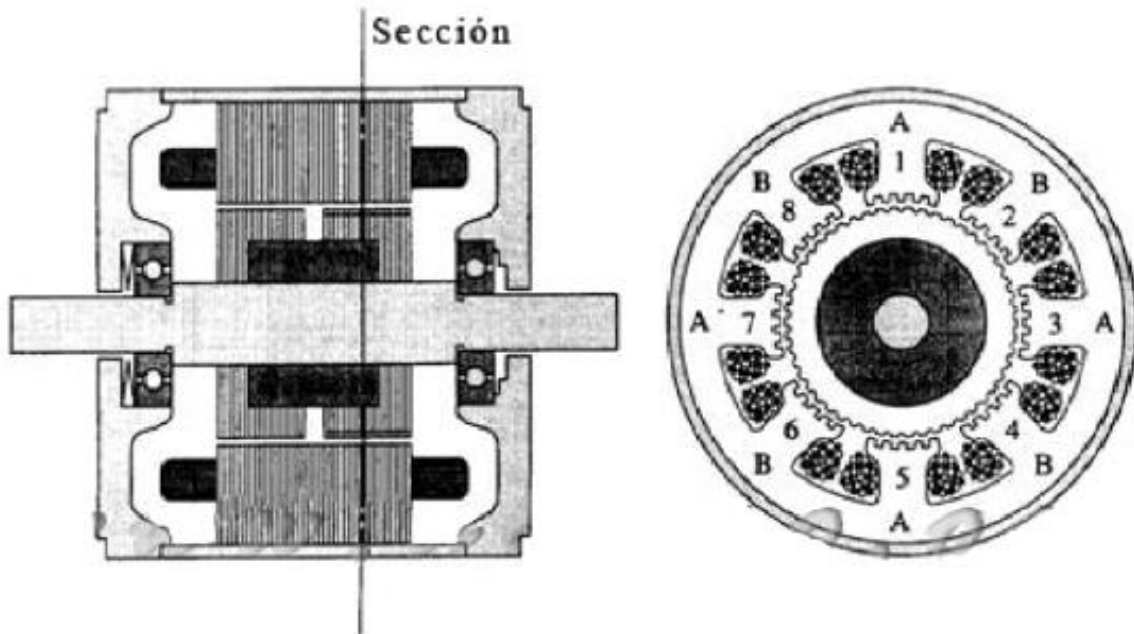


Figura 2.4. motor de pasos híbrido. [9]

www.bdigital.ula.ve

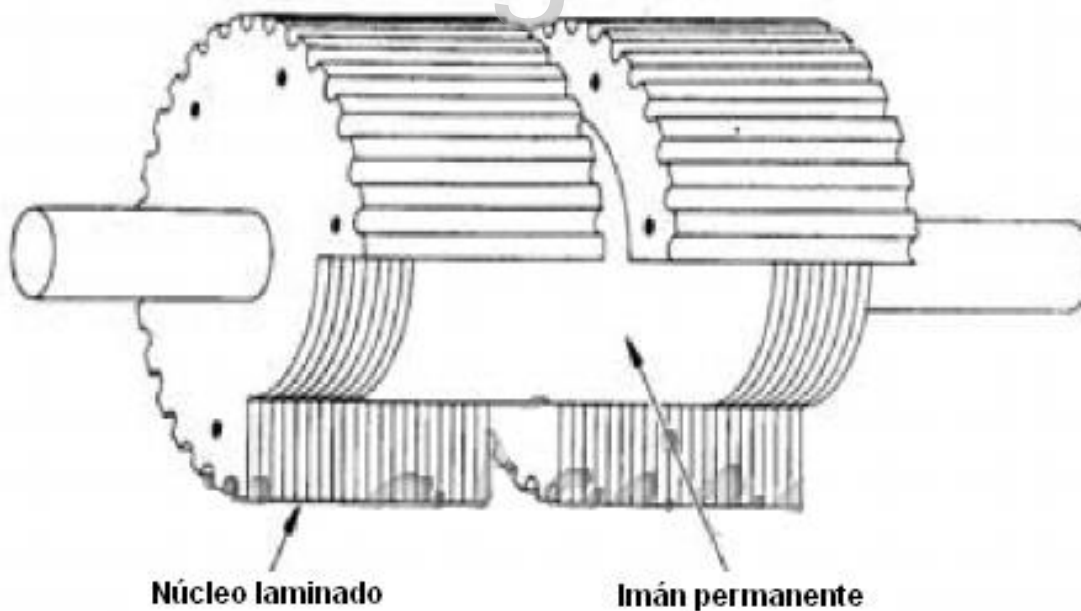


Figura 2.5. Rotor de un Motor de Pasos Híbrido. [9]

- **Motores de Paso Unipolares**

Estos motores tienen de 5 a 6 conexiones, esto depende de sus conexiones internas, dichos motores se caracterizan por ser más sencillos de controlar, estos utilizan una conexión común a la fuente de alimentación y posteriormente se van colocando las otras líneas a tierra, en un orden específico para generar cada paso, si tienen 6 conexiones es debido a que cada par de bobinas tiene un común separado, en caso de tener 5 conexiones es debido a que las 4 bobinas tienen un común separado, un ejemplo de esto se encuentra en la figura 2.6.

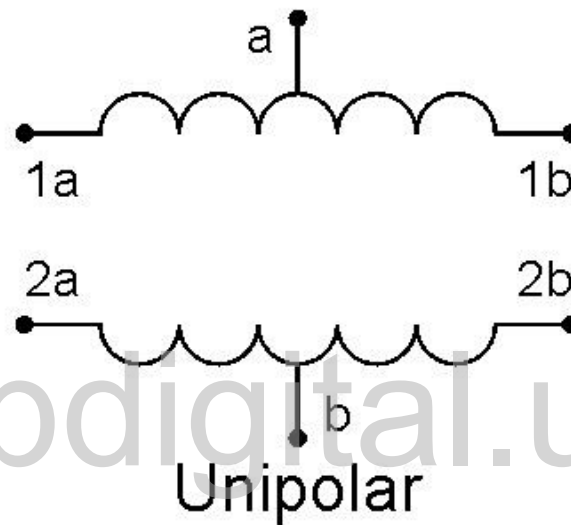


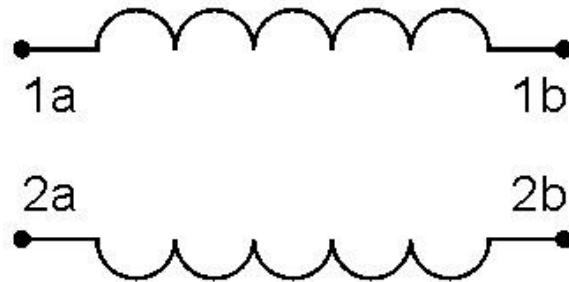
Figura 2.6. Esquema de Motor de Paso Unipolar. [10]

- **Motores de paso bipolares**

Estos motores de pasos constan de 4 conexiones, estos no son tan sencillos de controlar como los motores unipolares, para lograr un buen funcionamiento necesitan una serie de activación de bobinas específica para ser controlados, esto debido a que requieren del cambio de dirección de flujo de corrientes a través las bobinas. [8]

En este trabajo se emplearon motores de paso bipolares los cuales ya estaban disponibles en el brazo robótico; en la figura 2.7 se muestra un esquema básico para entender las conexiones internas de estos motores. para hacerlos girar correctamente es necesario activar las bobinas en una secuencia específica como se muestra en la tabla 2.1, donde Vcc2 es el voltaje de la

fuente de alimentación (12V) y EN se encuentra conectado a tierra (GND), de esta manera garantizamos el cambio de flujo eléctrico en los motores.



Bipolar

Figura 2.7. Esquema de Motor de paso Bipolar. [10]

www.bdigital.ula.ve

Tabla 2.1. Secuencia de Activación de Bobinas. [8]

PASO	TERMINALES			
	1a	1b	2a	2b
1	Vcc2	EN	EN	EN
2	EN	Vcc2	EN	EN
3	EN	EN	Vcc2	EN
4	EN	EN	EN	Vcc2

2.1.3 PUENTE H

Un puente H es un circuito electrónico que generalmente se utiliza para el control de motores eléctricos DC, o en este caso para controlar motores de paso, estos son utilizados ampliamente en robótica y como conversores de potencia. [11]

El termino puente H proviene de la típica representación gráfica del circuito, un puente H se construye a partir de 4 transistores como se muestra en la figura 2.8.

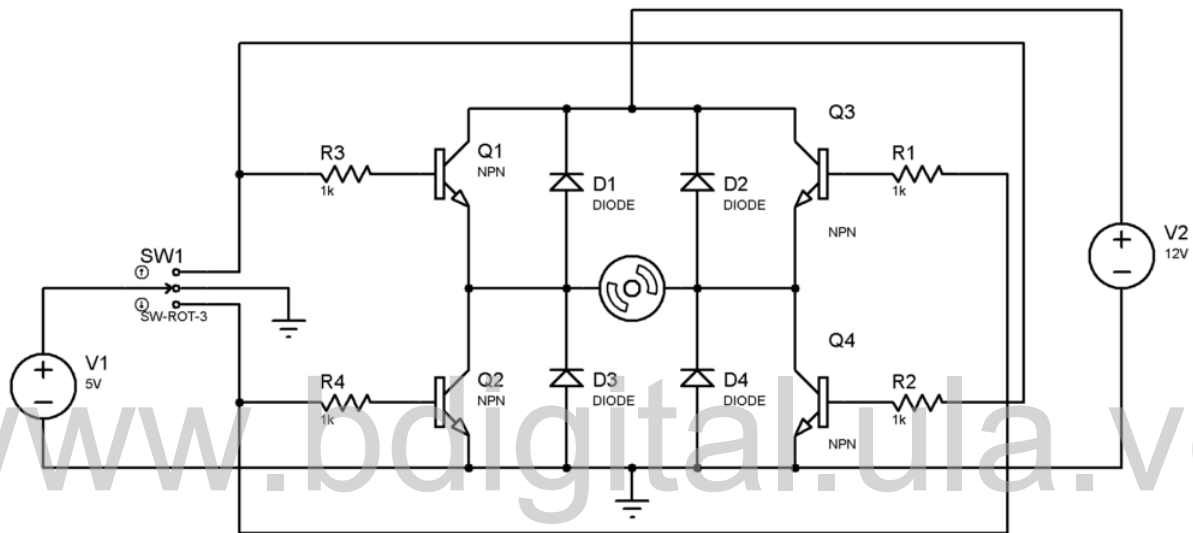


Figura 2.8. Estructura Básica de un Puente H. [12]

En este caso se utilizó un puente H (L298KV) el cual consta de 4 transistores capaces de conmutar corrientes de hasta 2A y voltajes de 5V hasta 50V.

Debido a que el microcontrolador PIC no puede aportar la corriente ni el nivel de voltaje necesario para controlar los motores de pasos, es necesario emplear puentes H, dichos puentes H son los encargados de realizar la etapa de potencia, ya que estos pueden manejar las corrientes y voltajes exigidos por los motores de paso.

Para este trabajo se seleccionó el integrado L298KV debido a que posee las características eléctricas necesarias para el control de los motores de pasos del brazo robótico, no se emplearon disipadores de calor en los puentes H debido a que el brazo robótico va a ser

utilizado con fines didácticos, por lo tanto, no se necesita movilizar objetos pesados lo que significa una reducción en el amperaje necesario para el funcionamiento de este.

Las características más importantes de los puentes H se muestran en la tabla 2.2, en los que destacan los voltajes y corrientes que puede manejar estos puentes H.

Tabla 2.2. Condiciones de Operación Óptima del Puente H. [13]

Características eléctricas	MIN	MAX	Unidades
Voltaje Lógico Suministrado, Vcc1	5	7	V
Voltaje de Salida, Vcc2	Vcc1	50	V
Voltajes por puertos de entrada, A o EN	-0.3	7	V
Corriente en operación continua	2	2	A
Pico de corriente de salida combinada	3	3	A
Temperatura de operación (aire libre)	-40	150	C°

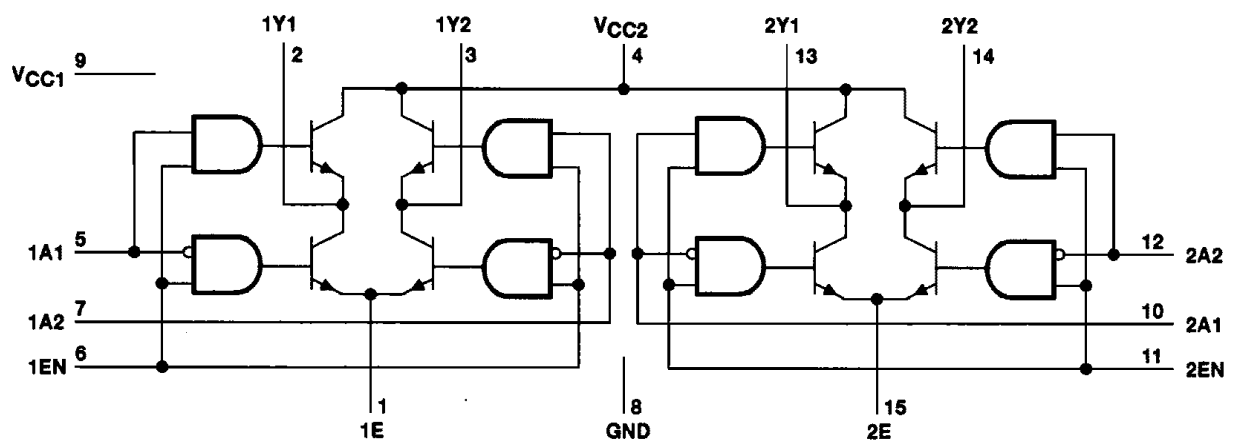


Figura 2.9. Diagrama interno del Puente H (L298KV). [13]

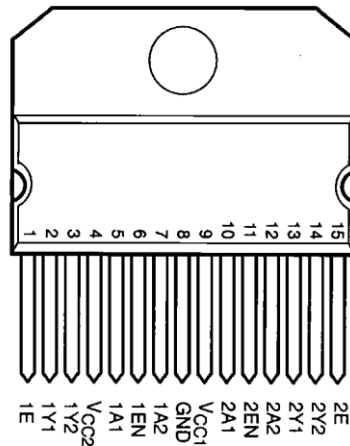


Figura 2.10. Encapsulado del puente H (L298KV). [13]

Tabla 2.3. Tabla de la verdad (L298KV). [13]

Entradas A	Entradas EN	Salidas Y
H	H	H
L	H	L
X	L	Z

Para controlar los motores de paso del brazo robótico es necesario emplear un puente H completo por cada motor, debido a que estos motores son bipolares de 4 cables, para la cual en la tabla 2.1 se describió la secuencia correcta para su funcionamiento.

2.2 SOFTWARE

En esta sección se muestran los entornos de programación y software de simulación necesarios para realizar este trabajo, entre los programas utilizados tenemos: compilador Mplab XC8, IDE Mplab X, IDE QT Creator, Proteus Desing Suite, ARES, ISIS, Modulo VSM.

2.2.1 Compilador MPLAB XC8

Un compilador es un software encargado de transformar un programa que se encuentra escrito en lenguaje de alto nivel a otro lenguaje, usualmente se transforma a lenguaje de máquina, el compilador también es el encargado de optimizar el código además de comprobar si existen errores de síntesis.

El compilador XC8 fue diseñado y desarrollado por una empresa llamada HITECH, la cual posteriormente paso a ser parte de Microchip, este compilador es el utilizado por el MPLAB X, Los errores que muestra el compilador son del tipo sintácticos, esto quiere decir que, si el programa tuviera algún error, por ejemplo, que esperase a que se ponga un bit en “0” y nunca pasase, se estará en un bucle infinito a pesar de que el compilador compilará perfectamente porque no hay error de sintaxis.

También existen mensajes y advertencias, las advertencias tienen un poco más de peso, por ejemplo: el microcontrolador PIC seleccionado no es el mismo que está definido en el programa, en ambos casos de existir mensajes y advertencias, la compilación termina satisfactoriamente, pero hay que tener en cuenta siempre lo que nos dicen estos para prevenir errores.

Una vez terminada la compilación el MPLAB nos genera un archivo cuya extensión es (.HEX) el cual es completamente entendible para el microcontrolador PIC. Es decir, solo resta grabarlo al microcontrolador PIC por medio de una interfaz como por ejemplo el programador Easy PIC 6 de microchip. Una vez completado esto, se alimenta al mismo y el programa ya se estará ejecutando.

Este compilador tiene una versión gratuita la cual nos genera un archivo de programación para el microcontrolador PIC (.HEX) el cual no está de la mejor manera compilado (ocupa más espacio de memoria), para tener una mejor versión de este archivo se tiene que activar la versión paga, el compilador XC8 da la opción de probar la versión paga por 30 días.

2.2.2 IDE MPLAB X

Un IDE (Integrated Development Environment), entorno de desarrollo integrado, es una aplicación que proporciona servicios integrales para facilitarle al programador el desarrollo

del software, normalmente un IDE consiste en un editor de código fuente, herramientas de construcción automáticas y un depurador.

El IDE MPLAB X es un editor gratuito, destinado a productos de la marca Microchip, este editor es modular y permite seleccionar distintos microcontroladores PIC, además de permitir la grabación de estos mismos.

En este IDE se utiliza el lenguaje C como lenguaje nativo, el cual es de alto nivel, una vez que el programa es compilado nos genera un archivo (.HEX), este archivo es completamente entendible para el microcontrolador PIC, solamente restaría grabar el microcontrolador PIC con dicho archivo.

El MPLAB X es un programa que corre en los siguientes sistemas operativos, Windows, MacOS y Linux, Presenta las clásicas barras de programa, de menú, de herramientas de estado, etc. El ambiente MPLAB posee editor de texto, compilador y simulación (no en tiempo real).

Para comenzar un programa desde cero para luego grabarlo al microcontrolador en MPLAB los pasos a seguir son:

- Crear un nuevo archivo con extensión .ASM y nombre cualquiera
- Crear un Proyecto nuevo eligiendo un nombre y ubicación
- Agregar el archivo .ASM como un SOURCE FILE
- Elegir el microcontrolador a utilizar desde SELECT DEVICE del menú CONFIGURE

Una vez realizado esto, se está en condiciones de empezar a escribir el programa respetando las directivas necesarias y la sintaxis para luego compilarlo y grabarlo en el microcontrolador PIC. [14]

Las ventajas que presenta el IDE MPLAB X son las siguientes:

- Utilidad de importación para una migración rápida y sencilla de proyectos de la plataforma anterior IDE MPLAB
- Terminación de código y menús de contexto mediante editor avanzado
- Ofrece soporte para múltiples versiones de compilador simultáneamente

- Herramientas de colaboración en equipo para detección de errores y control de código fuente [14]

En la figura 2.11 se muestra como es el entorno de desarrollo del MPLAB X en el que se utilizan menús clásicos como en cualquier programa de Windows.

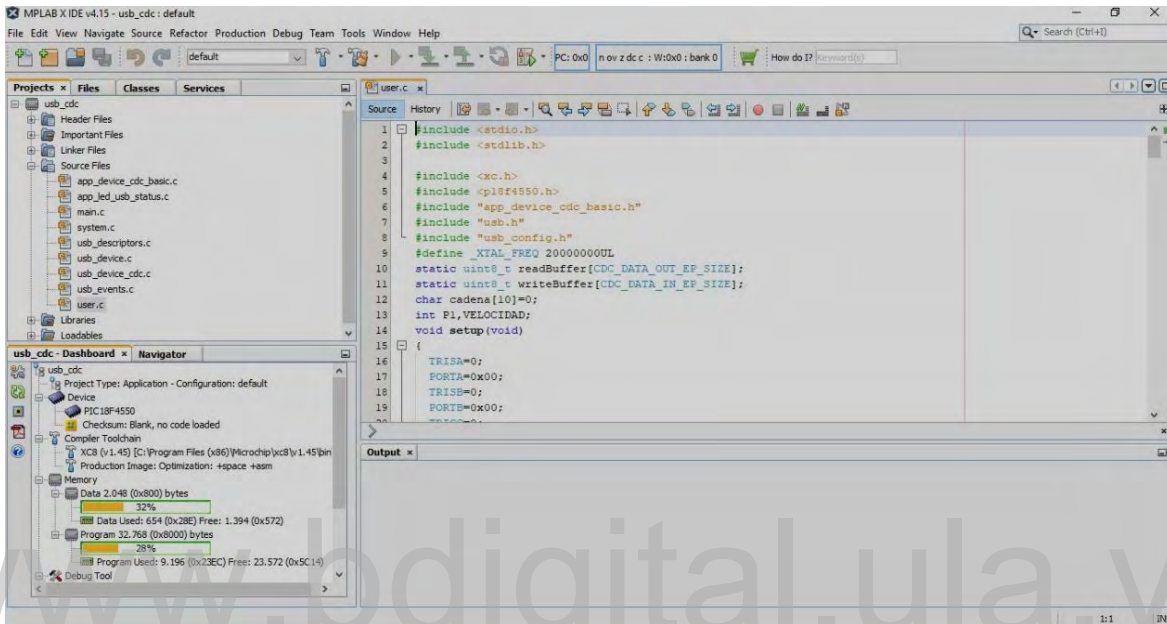


Figura 2.11. IDE MPLAB X.

En el panel derecho se encuentra un cuadro donde se escriben las líneas de comando que va a ejecutar el microcontrolador PIC, en la ventana superior izquierda se muestran las librerías que posee el proyecto, y debajo de esta ventana se encuentra otro cuadro en el que se detalla la memoria que ocupa el programa, que compilador se va a utilizar, que microcontrolador PIC se está programando, etc.

2.2.2. IDE QT CREATOR

Qt Creator es un IDE multiplataforma el cual es programado en lenguaje C++, JavaScript y QML, este programa fue creado por la empresa Trolltech, este programa es utilizado para el desarrollo de aplicaciones con interfaces graficas de usuario.

Qt Creator es desarrollada como un software libre, utiliza el lenguaje de programación C++ de forma nativa, funciona en las principales plataformas y tiene un amplio apoyo. [15]

Qt creator presenta grandes ventajas para los nuevos usuarios, entre las principales ventajas tenemos:

- Editor de código con soporte para C++, QML y ECMAScript
- Herramientas para la rápida navegación del código
- Resaltado de sintaxis y auto-completado de código
- Control estático de código y estilo a medida que se escribe
- Soporte para refactorización de código
- Ayuda sensitiva al contexto
- Plegado de código (*code folding*)

En la figura 2.12 se puede observar como es el entorno del este programa, es de fácil comprensión y sencillo manejo.

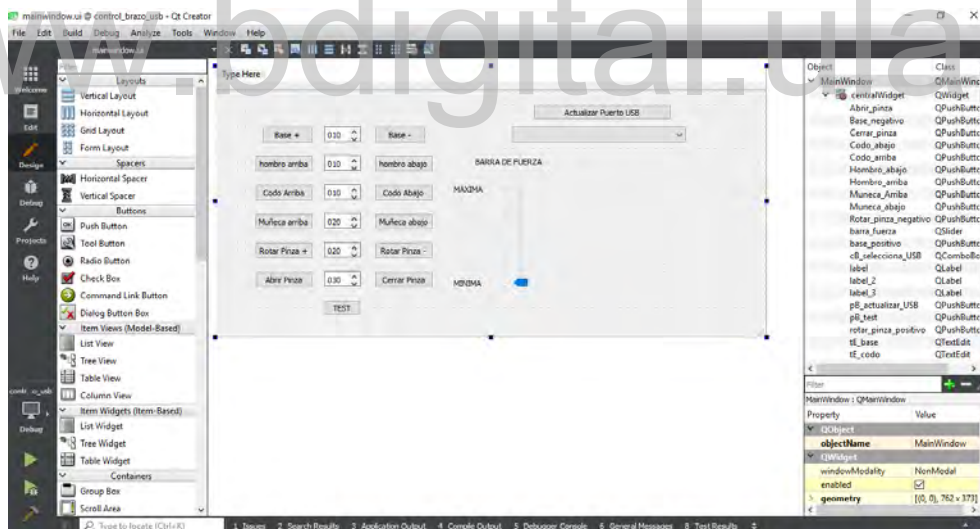


Figura 2.12. IDE QT Creator 4.6.2

2.2.3. Proteus Design Suite 8.0

Es un software de diseño electrónico, dicho software sirve para realizar diseño de automatización de procesos, esto es debido a que posee la capacidad de simular el funcionamiento de microcontroladores PIC cargando el archivo de programación (.HEX)

directamente al programa, este software es desarrollado por Labcenter Electronics Ltd., y consta de los dos programas principales: Ares e Isis, y los módulos VSM y Electra. [16]

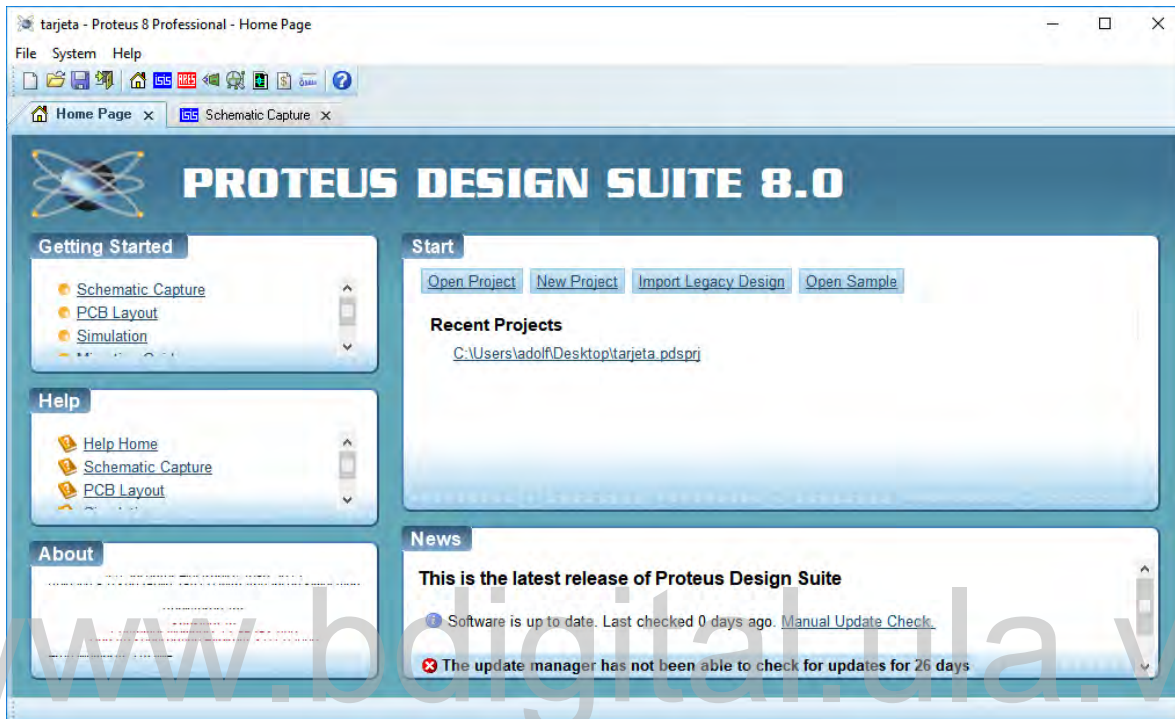


Figura 2.13. Proteus Desingn Suite 8.0

- **ISIS**

El programa ISIS, Intelligent Schematic Input System (Sistema de enrutado de esquemas inteligente), permite diseñar el diagrama del circuito que se desea realizar, además consta de una librería de componentes muy variada, desde simples resistencias, hasta algunos que otros microcontroladores PIC, incluye fuentes de alimentación, generadores de señales y muchos otros componentes con prestaciones diferentes. Los diseños realizados en ISIS pueden ser simulados en tiempo real mediante el módulo VSM, asociado directamente con ISIS. [17]

- **ARES**

El programa ARES, Advanced Routing and Editing Software (Software de Edición y Ruteo Avanzado), es una herramienta integrada al PROTEUS DESING SUITE, desinada a la fabricación de placas de circuito impreso, este programa tiene la habilidad de ubicar los

componentes donde se desee, siempre y cuando respete los criterios de diseños integrados en el programa, también permite editar generalmente, las capas superficial, y de soldadura. [18]

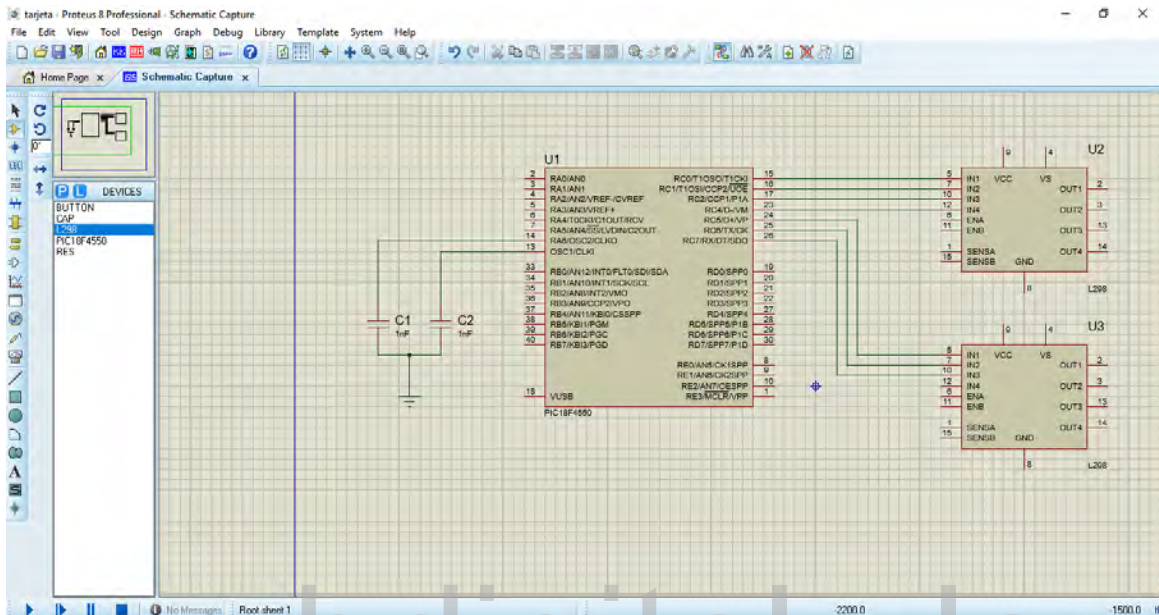


Figura 2.14. ISIS

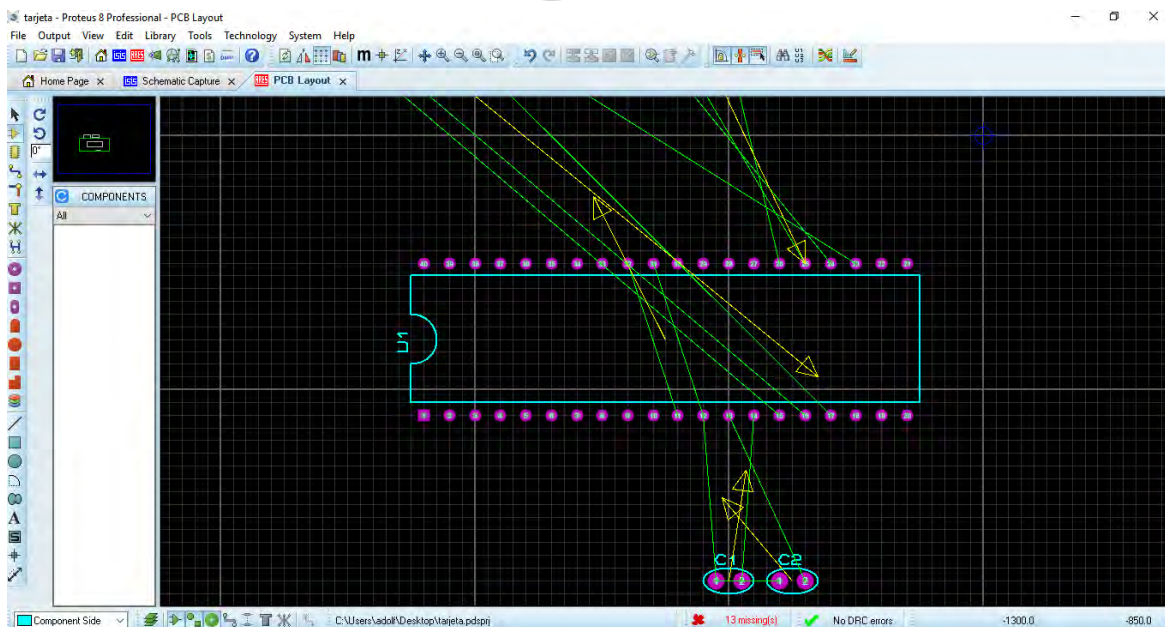


Figura 2.15. ARES

- **MODULO VSM**

Una de las prestaciones de Proteus integrada con ISIS, es VSM, (Virtual System Modeling), Sistema Virtual de Modelado, una extensión integrada con ISIS, con la cual se puede simular en tiempo real o con posibilidad de más rapidez; todas las características de varias familias de microcontroladores PIC, con la ventaja de introducir el programa que ejecutara el microcontrolador PIC, este módulo simula cada una de sus salidas, y a la vez, simulando las tareas que debe llevar a cabo.

Se pueden simular circuitos con microcontroladores PIC conectados a distintos dispositivos, como motores eléctricos, pantallas de cristal líquido (LCD), teclados en matriz, etc. Incluye las familias de microcontrolador PIC, tal como PIC10, PIC12, PIC16, PIC18, PIC24 y dsPIC33. ISIS es el corazón del entorno integrado Proteus. Combina un entorno de diseño de una potencia excepcional con una enorme capacidad de controlar la apariencia final de los esquemas. [19]

En este trabajo se utilizó la versión gratuita de este programa en el cual se tuvo la limitación de no poder guardar la simulación, cuando se tenían que hacer las simulaciones se tenía que montar todo el circuito desde cero y probarlo.

CAPITULO 3

DISEÑO DEL PROTOTIPO

En este capítulo se explica el proceso de diseño del hardware de la tarjeta de control, también se explica cómo se realizó software necesario para el control del brazo robótico CS-113 SERVICE-ARMS

3.1. DISEÑO DEL HARDWARE

En primer lugar, hay que destacar que la tarjeta de control del brazo robótico es una interface entre un computador y el propio brazo robótico, esta es la encargada de tomar las señales que envía el computador, procesarlas, tomar una decisión y ejecutar. Para poder hacer esto posible es necesario tener un microcontrolador PIC, el cual sea capaz de establecer una comunicación mediante vía USB con el computador, luego de un estudio del mercado para tener en cuenta cuales son los microcontroladores PIC con comunicación USB, se seleccionó el PIC 18F4550, debido a que es común en el mercado, es de bajo costo y este posee conexión USB.

Este microcontrolador PIC consta con el número de pines necesarios para el control de los 6 motores de paso que posee el brazo robótico, cada motor de paso consta de 4 conexiones por lo tanto se necesitan 24 pines para controlar solamente los motores de paso, esto sin contar las funciones adicionales que ofrecía la tarjeta de control original, en las cuales se encuentran un pulsador de RESET y uno de TEST, a esto se le adicionan 2 pines más para el control de los leds de encendido y de error; el brazo posee un pulsador adicional que se activa cuando la guaya encargada de abrir y cerrar la pinza esta tensa, para que este tensa la guaya, debe estar totalmente cerrada la pinza o estar sujetando algún objeto que se desee posteriormente mover, esta función adicional requiere de un pin más.

El microcontrolador PIC no proporciona la corriente necesaria para alimentar los motores de paso que posee el brazo robótico, por lo tanto, se debe emplear 6 puentes H (L298KV) uno por cada motor de pasos, los cuales tomaran las señales digitales provenientes del microcontrolador PIC y de esta forma poder ejecutar los movimientos dictados por él. Estos puentes H tomaran las tensiones y corrientes necesarias provenientes de una fuente de poder externa la cual proporciona un voltaje de 13.8V y una corriente de 5A, con estas especificaciones se logra un correcto funcionamiento de los motores de pasos.

Para llevar a cabo la construcción de la tarjeta de control se utilizó la siguiente lista de materiales:

Tabla 3.1. Componentes utilizados para el desarrollo del prototipo.

Componente	Cantidad
Microcontrolador PIC 18F4550	1
Puente H para motores de pasos (L298KV)	6
Oscilador de 20MHz	1
Capacitores 22 pf	2
Motores de paso Bipolares	6
Puerto de Alimentación Estándar (Hembra)	1
Puerto de Alimentación Estándar (Macho)	1
Puerto USB Estándar Tipo B (Hembra)	1
Cable USB	1
Espadines (conexión hacia los motores de paso)	32

A partir de la descripción del prototipo se planteó el diagrama de bloques de la figura 3.1, en esta figura se muestra el orden de jerarquía y esta remarcado con un cuadro el prototipo de tarjeta de control.



Figura 3.1. Diagrama de bloques del prototipo.

El microcontrolador PIC 18F4550 tiene 40 pines de conexión entrada/salida, para destinar a cualquier propósito, ciertos pines tienen una función específica predeterminada tales como: alimentación a nivel de 5V, conexión a neutro GND, conexión del oscilador, MCLR, conexión USB, comunicación I2C, comunicación UART entre otros.

Para este proyecto se utilizó los pines del puerto USB, MCLR, los del oscilador y alimentación, de esta manera se pueden destinar los pines restantes para el control de los motores de paso y demás funciones.

Gracias a que el microcontrolador PIC trabaja con voltajes lógicos de 0 a 5V, es totalmente compatible con los niveles de tensión de los puentes H, por lo tanto, no es necesario emplear adaptadores ni opto acopladores, logrando así una simplificación en el circuito y una inversión menor a la hora de realizar el prototipo de tarjeta de control.

El microcontrolador PIC necesita una alimentación de 5V, la cual obtendremos del voltaje suministrado por el puerto USB, esto significa una ventaja debido a que no se necesita emplear un regulador de voltaje, este voltaje suministrado por el puerto USB se utilizara también para alimentar ciertos pines de los puentes H, lo cual es necesario para el funcionamiento de dichos puentes.

3.2. DISEÑO DE TARJETA DE CONTROL

Debido a que no se cuenta con una máquina para realizar el circuito impreso y realizar el circuito en placa perforada es muy complicado por el número de conexiones que se deben realizar, se optó por dejar planteado un diseño de prototipo de tarjeta de control esto con la finalidad de poder ser empleado cuando se disponga de la máquina o de los medios económicos para realizar el circuito impreso, para el diseño se recomienda utilizar una placa de 14 cm x 10cm si se desea realizar a una capa, si se desea realizar a 2 capas se puede reducir considerablemente el tamaño de la tarjeta de control.

Al prototipo de tarjeta de control se le deben realizar 3 perforaciones para ser atornillada al brazo robótico para que quede inmóvil; teniendo en cuenta las dimensiones de la tarjeta de control se puede agregar una fuente de poder, por ejemplo; logrando así un equipo más compacto y con menos accesorios que trasladar.

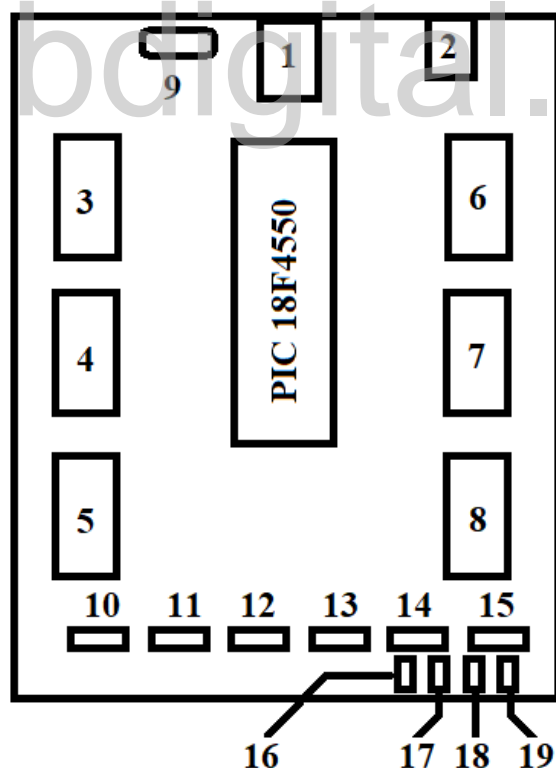
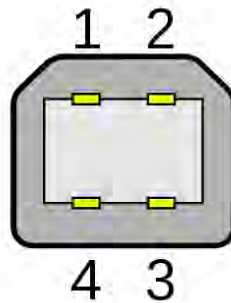


Figura 3.2. disposición de los componentes del prototipo.

En la figura 3.2 se muestra la disposición de los componentes del prototipo en una placa de 14 cm x 10 cm, a continuación, se especifican las partes de la tarjeta de control prototipo según el número que tiene marcado:

- **1) Entrada USB estándar tipo B (Hembra):** este es el puerto de comunicación entre el prototipo y el computador, se utilizó un puerto USB común tipo B (figura3.3) con la finalidad de conseguir fácilmente un cable para la conexión de datos.



Type-B

Figura 3.3. Puerto USB Tipo B. [22]

- **2) Puerto de alimentación (12V):** este puerto de conexión es el que les provee energía eléctrica a los puentes H, por lo tanto, debe ser un puerto robusto capaz de soportar corrientes elevadas, en la figura 3.4. se muestra el puerto de alimentación seleccionado.



Figura 3.4. Puerto de Alimentación. [23]

- **3) Puente H:** son los encargados de realizar la etapa de potencia, recibiendo las señales digitales en niveles lógicos, se disponen a los lados del microcontrolador PIC logrando conexiones cortas, los puentes H están especificados con los números **3,4,5,6,7,8**.
- **9) Oscilador:** este oscilador estrictamente debe ser de 20Mhz, en la hoja de datos del microcontrolador PIC 18F4550 especifican la velocidad del reloj externo, para lograr una conexión vía USB, de lo contrario no se tendrá la velocidad necesaria para establecer la comunicación y se perderán los datos.
- **10) Puerto de Conexión (Codo)**
- **11) Puerto de Conexión (Pinza)**
- **12) Puerto de Conexión (Muñeca I)**
- **13) Puerto de Conexión (Muñeca II)**
- **14) Puerto de Conexión (Base)**
- **15) Puerto de Conexión (Hombro)**
- **16) Puerto de Conexión (Pulsador de la Pinza):** esta conexión tiene la finalidad de indicarle a la tarjeta de control prototipo que ya la pieza que se desea movilizar se encuentra sujeta (esta función no se empleó en el programa, debido a que los motores de paso no tienen la suficiente fuerza para lograr la tensión en la guaya, por lo tanto, no activan el pulsador, sin embargo, se deja disponible el puerto de conexión para cuando se desee habilitar dicha función).
- **17) Puerto de Conexión (Led de Error):** este led se enciende cuando se envía un numero de pasos mayor al que permite la articulación, esto con el fin de mostrarle al usuario que el número de pasos no es el indicado debido a que está excediendo las capacidades físicas de la articulación.
- **18) Puerto de Conexión (Pulsador de Test):** al presionar este pulsador la tarjeta prototipo procede a ejecutar un breve test, con la finalidad de probar todos los componentes de brazo.
- **19) Puerto de Conexión (MASTER CLEAR)** en este puerto se conecta el pulsador de RESET que posee el brazo en el panel delantero.

3.3 CONEXIÓN USB

Para lograr la conexión USB entre el computador y el PIC 18F4550, fue necesario analizar la conexión física y comprender el funcionamiento del puerto USB:

El puerto USB consta de 4 conductores de los cuales 2 son conexiones de datos (D+, D-), el signo + y el - es porque es una señal diferencial cuyo valor depende de la velocidad del bus (3,3v baja velocidad y 400mV alta velocidad), los otros 2 conductores son de alimentación, esta distribución la podemos ver en la figura 3.5.

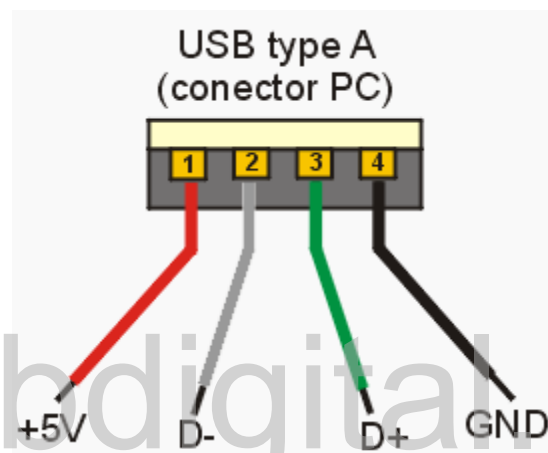


Figura 3.5. conexiones puerto USB. [24]

Existen 2 maneras principales para conectar el microcontrolador PIC con el computador mediante vía USB, la primera se llama HID (Human Interface Device), algunos ejemplos de equipos que trabajan por este tipo de comunicación son los teclados, ratones, pantallas táctiles entre otros, otra forma de comunicación se llama CDC (Communications Device Class), el ejemplo principal para este tipo de comunicación son los módems y los dispositivos Arduino que también utilizan este tipo de comunicación.

Para este trabajo se utilizó la comunicación CDC, debido a que es una comunicación más sencilla de trabajar, es ampliamente utilizada y es compatible con QT creator 4.6.2, también fue necesario descargar todas las librerías para el microcontrolador PIC para lograr dicha comunicación, estas librerías fueron descargadas del sitio oficial de Microchip.

3.4 DISEÑO DE CONEXIÓN PIC A PUENTE H (L298KV)

Debido a que el microcontrolador PIC no posee las características eléctricas necesarias para encender las bobinas de los motores de paso, es necesario implementar los puentes H (L298KV), estos puentes H ejecutan la etapa de potencia recibiendo las señales dictadas por el microcontrolador PIC.

Para lograr un correcto funcionamiento se deben colocar las conexiones según lo muestra la figura 3.6 mostrado a continuación:

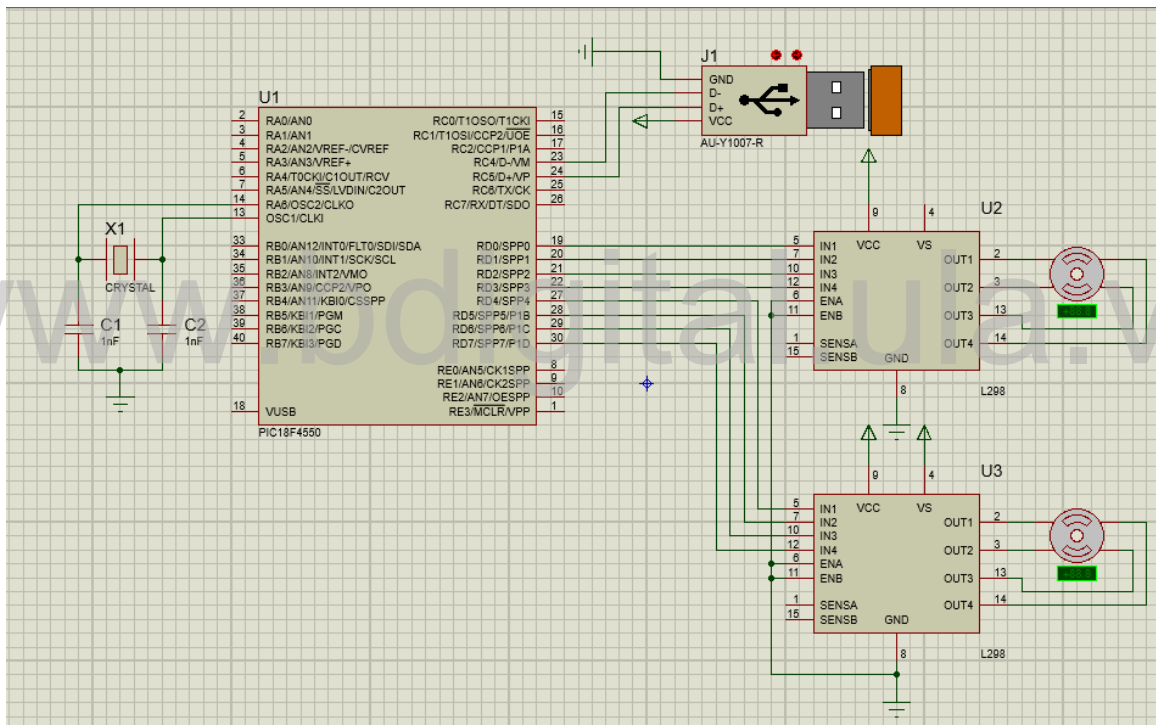


Figura3.6. Diagrama de conexiones para los motores de pasos de la muñeca.

En la figura 3.6 se muestra un ejemplo de conexión entre el microcontrolador PIC y los motores de paso de la muñeca, de igual manera se deben conectar los otros motores, las conexiones de los otros motores de paso se muestran con más detalle en los anexos. La figura 3.7 muestra los puertos asignados para la conexión de cada motor.

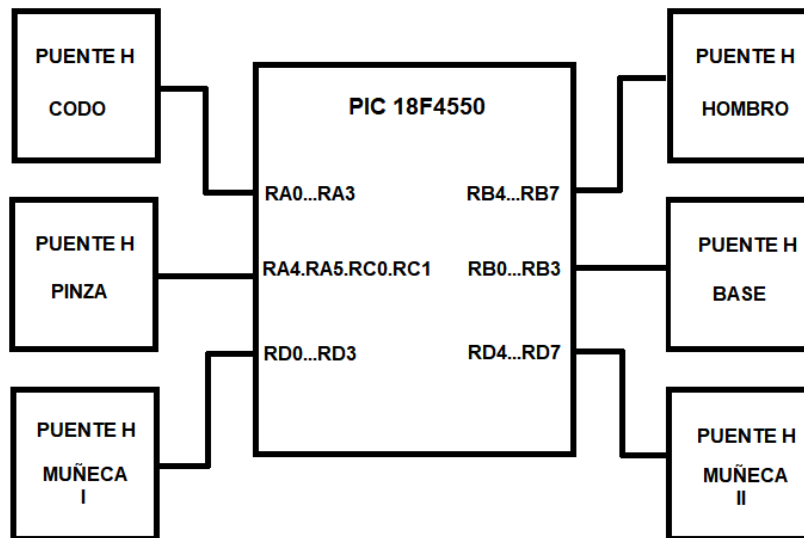


Figura 3.7. Esquema de Conexiones de los Puentes H con el Microcontrolador.

A continuación, se muestra una lista donde se detalla las conexiones entre el microcontrolador PIC y los Puentes H:

- Base-----Puerto B desde el RB0 hasta RB3
- Hombro-----Puerto B desde el RB4 hasta RB7
- Codo-----Puerto A desde el RA0 hasta RA3
- Muñeca-----Puerto D desde RD0 hasta RD7
- Pinza----- Puerto A y C, pines RC0, RC1, RA4 y RA5

Para mover la muñeca es necesario controlar 2 motores de paso, por esto es necesario asignarle todo el puerto D a esta articulación, la muñeca tiene la particularidad de realizar 4 movimientos, arriba, abajo, rotar sentido horario, rotar sentido anti horario.

Para lograr mover la muñeca para arriba o para abajo se deben activar los motores de paso en el mismo sentido y para rotarla se deben accionar en sentidos contrarios.

En las tablas 3.2, 3.3, 3.4, 3.5, 3.6, 3.7 se muestran las secuencias de activación de las salidas del microcontrolador PIC para el correcto funcionamiento de los motores de paso:

Tabla 3.2. secuencia de activación motor de pasos de la base (Horario).

PUERTO B	RB3	RB2	RB1	RB0
PRIMER PASO	0	0	0	1
SEGUNDO PASO	0	0	1	0
TERCER PASO	0	1	0	0
CUARTO PASO	1	0	0	0

Tabla 3.3. secuencia de activación motor de pasos del hombro (Arriba)

PUERTO B	RB7	RB6	RB5	RB4
PRIMER PASO	0	0	0	1
SEGUNDO PASO	0	0	1	0
TERCER PASO	0	1	0	0
CUARTO PASO	1	0	0	0

Tabla 3.4. secuencia de activación motor de pasos del codo (Arriba).

PUERTO A	RA3	RA2	RA1	RA0
PRIMER PASO	0	0	0	1
SEGUNDO PASO	0	0	1	0
TERCER PASO	0	1	0	0
CUARTO PASO	1	0	0	0

Tabla 3.5. secuencia de activación motor de pasos de la muñeca (sentido horario).

PUERTO D	RD7	RD6	RD5	RD4	RD3	RD2	RD1	RD0
PRIMER PASO	1	0	0	0	0	0	0	1
SEGUNDO PASO	0	1	0	0	0	0	1	0
TERCER PASO	0	0	1	0	0	1	0	0
CUARTO PASO	0	0	0	1	1	0	0	0

Tabla 3.6. secuencia de activación motor de pasos de la muñeca (arriba).

PUERTO B	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0
PRIMER PASO	0	0	0	1	0	0	0	1
SEGUNDO PASO	0	0	1	0	0	0	1	0
TERCER PASO	0	1	0	0	0	1	0	0
CUARTO PASO	1	0	0	0	1	0	0	0

Tabla 3.7 secuencia de activación motor de pasos de la pinza (abrir pinza)

PUERTO A y C	RA5	RA4	RC3	RC2
PRIMER PASO	1	0	0	0
SEGUNDO PASO	0	1	0	0
TERCER PASO	0	0	1	0
CUARTO PASO	0	0	0	1

3.5 DISEÑO DEL SOFTWARE

En esta sección se explica el desarrollo del software del microcontrolador PIC y el del computador (interface de usuario).

3.5.1 SOFTWARE MICROCONTROLADOR PIC

Para el diseño del software del microcontrolador PIC utilizamos el IDE MPLAB X el cual utiliza el lenguaje de programación C, también se utiliza el compilador XC8 el cual nos genera un archivo de programación (.HEX), al terminar la compilación este archivo puede ser utilizado para grabar el microcontrolador PIC.

Se descargaron las librerías que proporciona la página oficial de Microchip Technology, luego se hizo un estudio del funcionamiento y de cuales elementos eran necesarios y cuales se podían eliminar para simplificar el código, una ventaja de esto es que el código ocupa menos espacio y el programa se puede ejecutar con más fluidez y rapidez.

El programa del microcontrolador PIC tiene la siguiente secuencia de la figura 3.8.



Figura 3.8. diagrama del programa (Microcontrolador PIC).

Al realizar la conexión física por primera vez entre la tarjeta prototipo y el computador ocurre un proceso llamado enumeración, este proceso ocurre una sola vez y es donde el microcontrolador PIC le comunica a la computadora como debe ser controlado, se transfieren los datos sobre el VID (El ID del vendedor) y el PID (El ID del producto), también se especifican el consumo de corriente de dispositivo, el tipo de transferencia de datos que se va a utilizar, versión USB soportada, entre otros, una vez pasa el proceso de enumeración ya el computador tiene en su base de datos todas las especificaciones del dispositivo conectado, por lo tanto cuando se vuelva a conectar el computador establece una comunicación directa con el dispositivo y no es necesaria la enumeración, debido a que es un proceso que se realiza una sola vez, no se incluyó en el diagrama del programa.

Una vez realizado todo el proceso de enumeración, el programa corre como lo indica la figura 3.8 esto significa que no se necesita realizar otra vez la enumeración, sino que por el puerto se envía la cadena de información, el microcontrolador PIC la analiza la ejecuta y se queda esperando la siguiente orden.

En las librerías proporcionadas por Microchip se realizaron modificaciones, una de las modificaciones fue poner el nombre BRAZO ROBOTICO ULA, el cual aparecerá al momento de realizar la conexión física al computador por primera vez, estos datos son transmitidos al computador cuando ocurre la enumeración, en esta etapa se transmiten todos los datos que se encuentran en la librería USB_DESCRIPTOR.

Cuando la computadora envía una orden, el microcontrolador PIC la atiende de inmediato debido a que está activa una interrupción, si se envían varias órdenes al mismo tiempo el microcontrolador PIC no será capaz de atenderlas todas provocando que alguna de las ordenes que se enviaron no se ejecute.

En la figura 3.9 se muestra un ejemplo típico de una cadena de caracteres enviada desde el computador al microcontrolador PIC. La primera letra de la cadena (A) se utiliza para seleccionar el motor que se quiere mover y el sentido, se utilizan (/) para separar los elementos de la cadena, después del primer (/) viene un número de 3 cifras, este número se utiliza para dictarle al microcontrolador PIC el número de paso que se quiere mover el motor de paso, este número no puede exceder el número de pasos total que permite la articulación,

si sobrepasa este número el microcontrolador PIC no ejecutara el movimiento y encenderá un led el cual indica que existe un error en la información enviada desde la computadora, luego del segundo (/) viene otro numero el cual especifica la cantidad de fuerza que se va a aplicar en cada movimiento, el 2 significa que sea menos fuerte el movimiento pero más rápido, en la tabla 3.8 se muestran los valores máximos que se pueden enviar al microcontrolador PIC sin presente algún error:

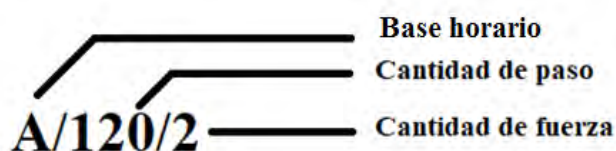


Figura 3.9. Ejemplo de cadena enviada desde computador.

Tabla 3.8. lista de numero de pasos máximo por motor.

Letra	MOVIMIENTO	#PASOS MAXIMO
A	BASE HORARIO	330
B	BASE ANTIHORARIO	330
C	HOMBRO ABAJO	140
D	HOMBRO ARRIBA	140
F	CODO ABAJO	100
G	CODO ARRIBA	100
H	CERRAR PINZA	500
I	ABRIR PINZA	500
J	MUÑECA HORARIO	900
K	MUÑECA ANTIHORARIO	900
L	MUÑECA ARRIBA	230
M	MUÑECA ABAJO	230

3.5.2 SOFTWARE DEL COMPUTADOR

Para el diseño del software del computador se utilizó el IDE QT Creator 4.6.2, se empleó este IDE debido a que es un software para interfaces de usuario, gratuito y de muy sencillo manejo, para comenzar con la programación se hizo una investigación de las librerías necesarias para que el QT Creator reconociera los dispositivos conectados vía USB, una vez logrado esto se pasó al desarrollo de la interface de usuario, se colocaron 12 botones (2 por cada motor de paso) para enviar la información necesaria al microcontrolador PIC; existen unos cuadros de texto, donde se pueden escribir el número de pasos que se desea mover cada motor cuando se pulse uno de los botones, también se adiciono una barra donde se controla la fuerza ejercida por los motores, esto se logra cambiando el tiempo que permanecen encendidas las bobinas.

Para saber cuánto es el tiempo máximo y el tiempo mínimo en el que se deben encender las bobinas de los motores de paso se hizo un estudio experimental, de este estudio se obtuvo un rango de tiempos en los cuales el movimiento del brazo robótico es óptimo, dicho rango está comprendido entre los 20 ms y los 90ms, si se exceden estos valores por arriba o por debajo de ese rango, el brazo pierde el número de pasos y comienza a moverse de manera incorrecta creando vibraciones, por lo tanto no se puede hacer un control preciso sobre él, a continuación se muestra en la figura 3.10 la interface de usuario:

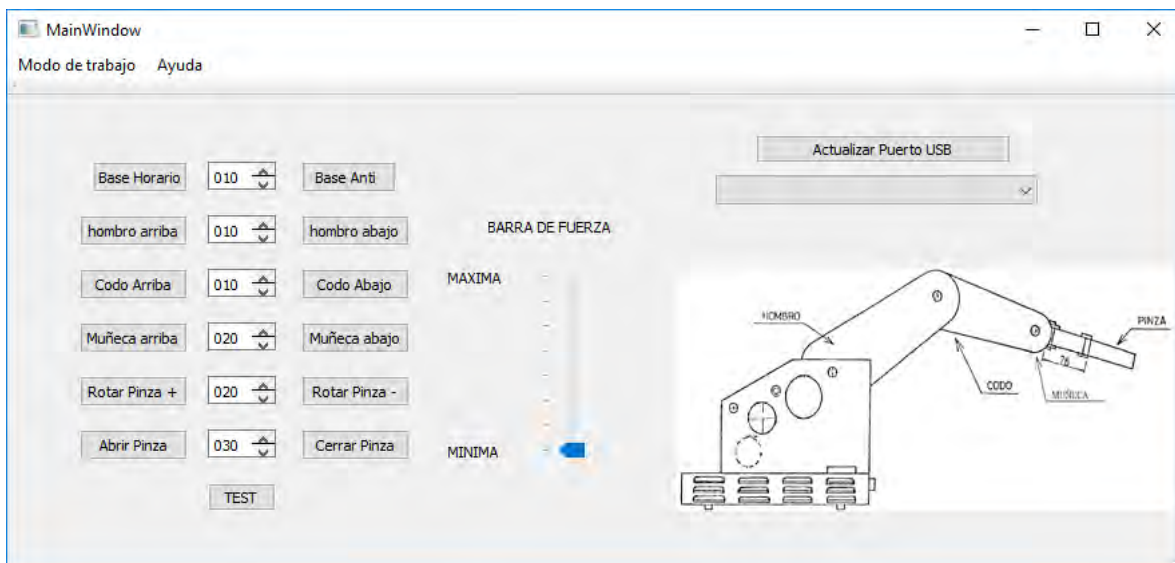


Figura 3.10 interface de usuario.

Para entender qué movimiento se va a realizar cada vez que se presione uno de los botones se tiene la figura 3.11, donde se muestran las diferentes partes de las que está constituida el robot, también se muestran las dimensiones, algo muy importante porque de esta manera sabemos cuál es el espacio de trabajo aproximado, esto hay que tenerlo en cuenta a la hora de realizar cualquier maniobra.

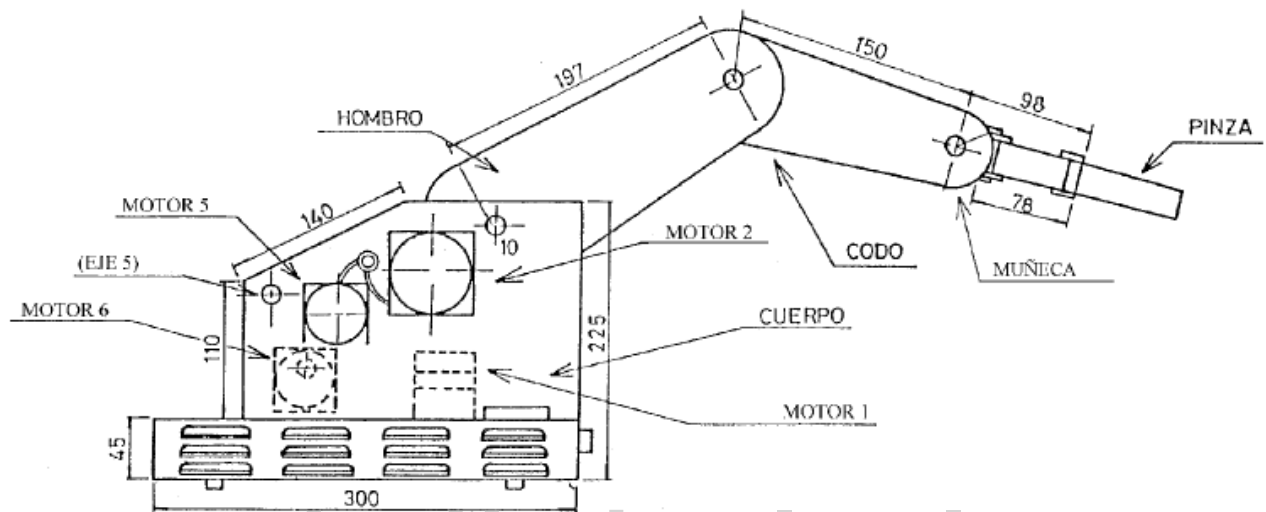


Figura 3.11. Partes y Dimensiones del Brazo Robótico CS-113.

CAPITULO 4

PRUEBAS DE FUNCIONAMIENTO

En este capítulo se exponen los resultados obtenidos de la integración entre el software y el hardware, así como las interfaces finales para el usuario y las instrucciones de uso para un correcto funcionamiento.

4.1 ANALISIS DEL SOFTWARE DEL COMPUTADOR

En primer lugar, el contacto con el software se obtiene abriendo el archivo ejecutable BRAZO ROBOTICO.EXE, una vez abierto este programa se va a desplegar una ventana, esta se puede observar en la figura 4.1, en esta ventana se encuentra la interfaz de usuario creada para el control del brazo robótico, como se puede notar tiene un estilo muy sencillo e intuitivo, esto con la finalidad de ser de fácil comprensión para el usuario de tal manera que no necesite un experto para operar el brazo.

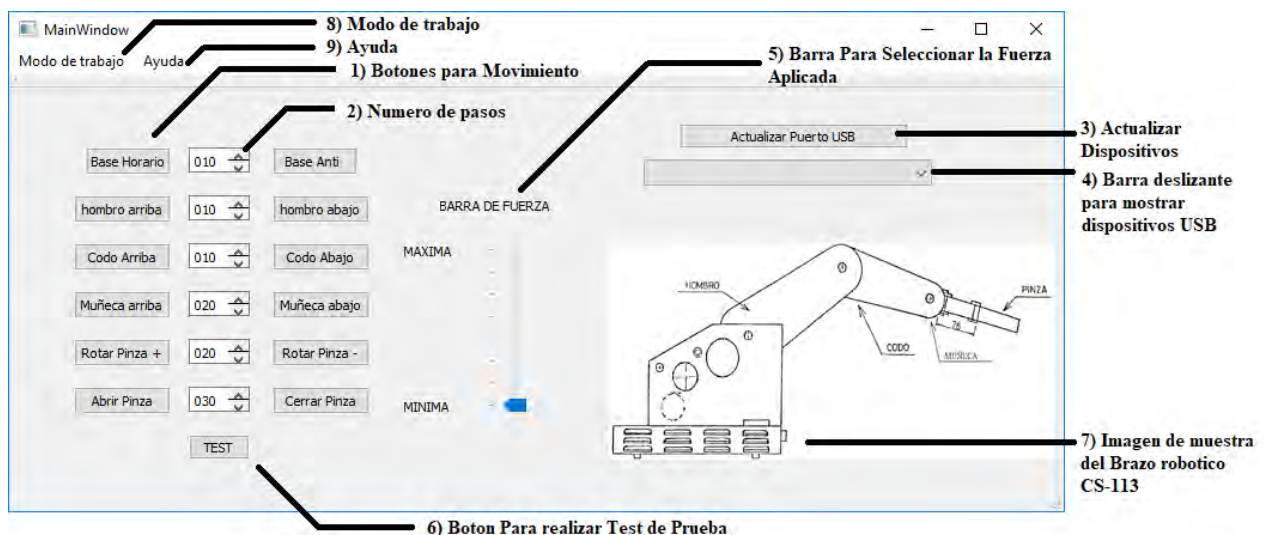


Figura 4.1 Interfaz de Usuario Detallada

En la figura 4.1 se muestra de forma detallada la interfaz de usuario creada para el control del brazo robótico, explicaremos a continuación el funcionamiento de cada uno de los ítems mostrados:

- 1) **Botones para movimiento:** se encuentran a la izquierda de la ventana en 2 columnas separadas en la mitad por unos cuadros de textos, al hacer clic con el ratón de la computadora, el programa arma automáticamente una cadena en la cual asigna una letra, seguidamente el número de pasos que está especificado en el cuadro de texto, que se encuentra al lado del botón seleccionado y por último toma el valor en el que se encuentre la barra de fuerza, una vez armada toda la cadena es enviada al microcontrolador. (ejemplo de cómo el computador arma una cadena se encuentra en la figura 3.9 del capítulo 3)
- 2) **Numero de pasos:** estos son cuadros de textos en los cuales se especifican el número de pasos que se desean mover los motores, estos cuadros permiten introducir un numero de 3 dígitos, esta información es utilizada para armar la cadena enviada al microcontrolador PIC, en la imagen se muestra un ejemplo de cómo se debe ser el número de pasos introducido. (números de pasos máximo permitidos por los motores se encuentran en la tabla 3.8 del capítulo 3)
- 3) **Actualizar Dispositivos USB:** este es un botón utilizado para actualizar los dispositivos conectados vía USB, solamente muestra los dispositivos USB con comunicación CDC en la barra deslizante que se encuentra debajo del botón Actualizar Dispositivos USB, esto se hizo con la finalidad de mostrar solamente los dispositivos compatibles con el programa.
- 4) **Barra deslizante:** en esta barra se muestran todos los dispositivos USB con comunicación CDC conectados a la computadora.
- 5) **Barra de fuerza:** esta es una barra la cual posee 7 posiciones, cabe destacar que mientras menor sea el numero seleccionado en la barra, el brazo se moverá con menos fuerza, pero con más velocidad.

- 6) **Botón para realizar Test:** se agregó un botón para realizar el test, esto con la finalidad de poder realizar la prueba sin necesidad de presionar el botón físico.
- 7) **Imagen del Brazo Robótico CS-113:** se colocó una imagen en la interface de usuario para saber que parte especifica se está moviendo del brazo robótico cuando se pulsan los botones de movimiento.
- 8) **Modo de Trabajo:** en este menú se puede elegir el Modo Consola, el cual es una interface de usuario primitiva, esta interface trabaja como la antigua consola de control del brazo robótico, al elegir este modo de trabajo se despliega la consola de control y una tabla donde se especifican el número de pasos máximo que permiten las articulaciones, esta interface se puede ver en la figura 4.2.



Figura 4.2. Interface de usuario (Modo consola).

- 9) **Ayuda:** en este menú se presentan 2 opciones, la primera es la tabla de pasos, en esta se muestra una tabla con las especificaciones de pasos máximo que soportan las articulaciones, en la segunda opción se presenta un ejemplo de una cadena de información que se puede enviar mediante la interface (Modo Consola).

Como se observa en la figura 4.1, y la figura 4.2 las interfaces de usuarios son bastantes intuitivas y sencillas de utilizar, se diseñaron de esta manera para que los usuarios puedan controlar el brazo rápidamente.

4.2 PASOS PARA PUESTA EN FUNCIONAMIENTO

Para poner en funcionamiento el brazo robótico solamente hay que seguir una serie de pasos sencillos los cuales mostraremos a continuación:

- 1) Encender la fuente de poder: se debe poner en funcionamiento porque esta fuente es la encargada de energizar los puentes H, la fuente que vaya a ser conectada al brazo robótico debe cumplir con los requerimientos mínimos de corriente y voltajes que son 2A y 12V, si se emplea una fuente de menor amperaje los motores de pasos no tendrán la suficiente fuerza para levantar objetos o simplemente no se mueve.
- 2) Conectar el brazo robótico mediante USB al computador que va a realizar el control, y que posee el programa para dicho propósito, el computador reconocerá el prototipo como un dispositivo USB (COM#), se puede visualizar el dispositivo como se muestra en la figura 4.3 (administrador de dispositivos de Windows), o en la figura 4.4 (dispositivos e impresoras de Windows).

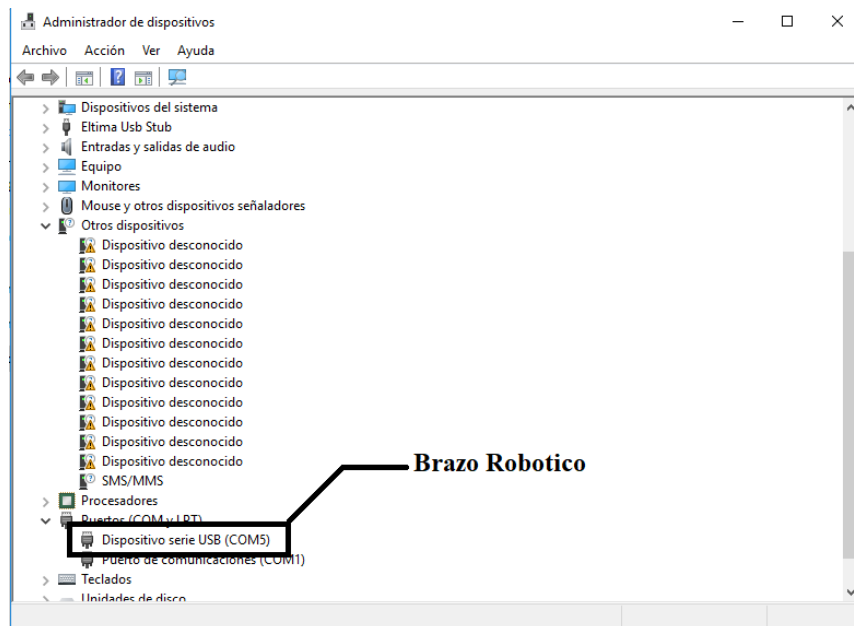


Figura 4.3. Administrador de dispositivos de Windows.

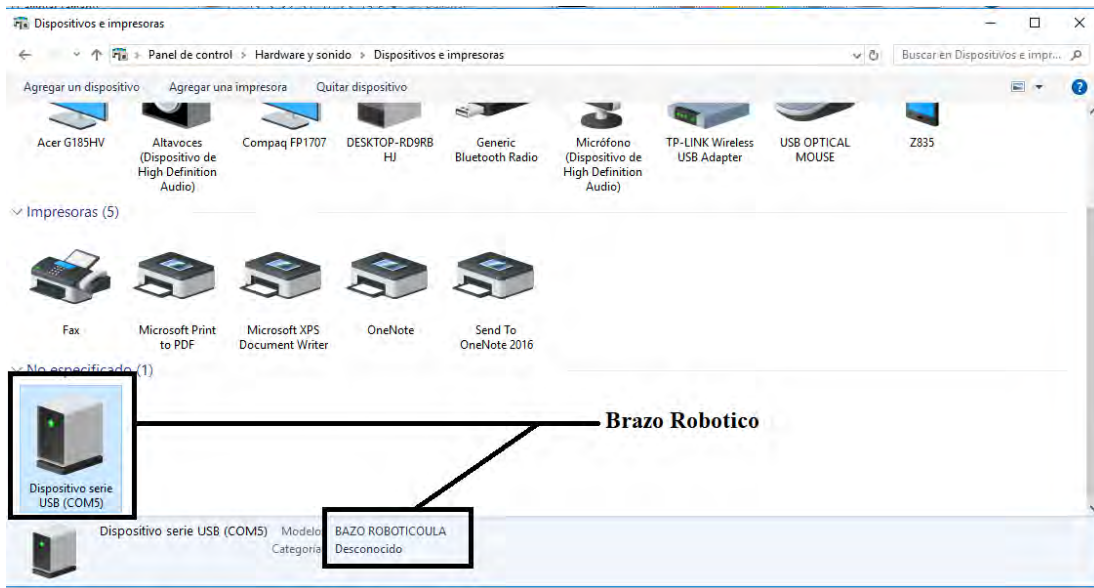


Figura 4.4 dispositivos e impresoras de Windows.

- 3) Ejecutar el programa de control del brazo robótico.
- 4) Presionar el botón (Actualizar puerto USB).
- 5) Seleccionar en la barra deslizante el brazo robótico, en la figura 4.5 se muestra los dispositivos conectados y cual se debe seleccionar.

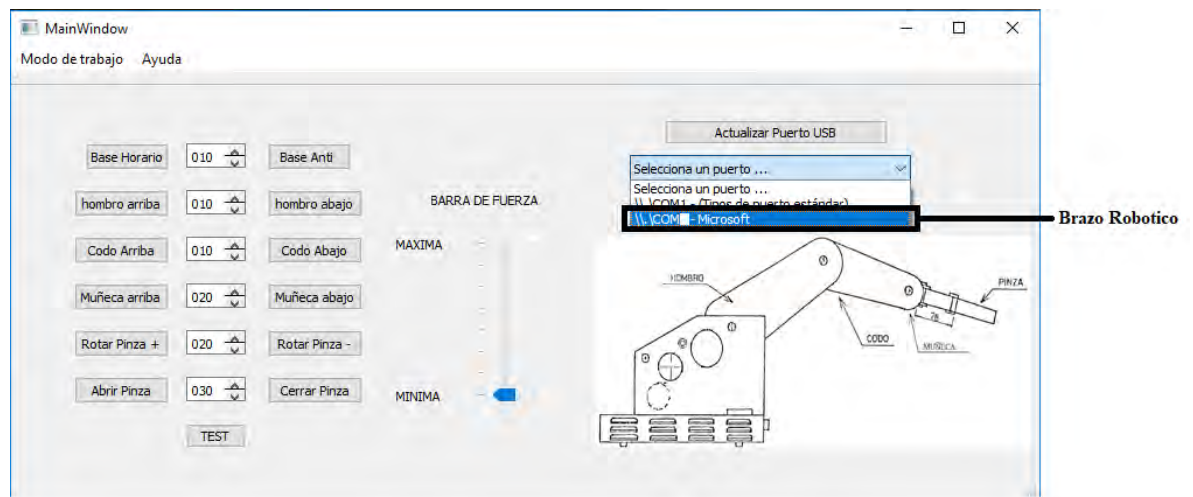


Figura 4.5. lista de dispositivos USB (COM)

- 6) Empezar a controlar el brazo robótico.

4.3 PRUEBA DE FUNCIONAMIENTO

En primer lugar, se tuvo que implementar el circuito en un protoboard, esto con la finalidad de comprobar el correcto funcionamiento del diseño, el protoboard tiene una desventaja, cualquier movimiento en los cables puede ocasionar una falla o la desconexión de alguno de los componentes, por lo tanto, es necesario una vez verificado el correcto funcionamiento del diseño, mudar los componentes a un circuito impreso con la finalidad de evitar posibles fallas, en la figura 4.6 se muestra el circuito en el protoboard.

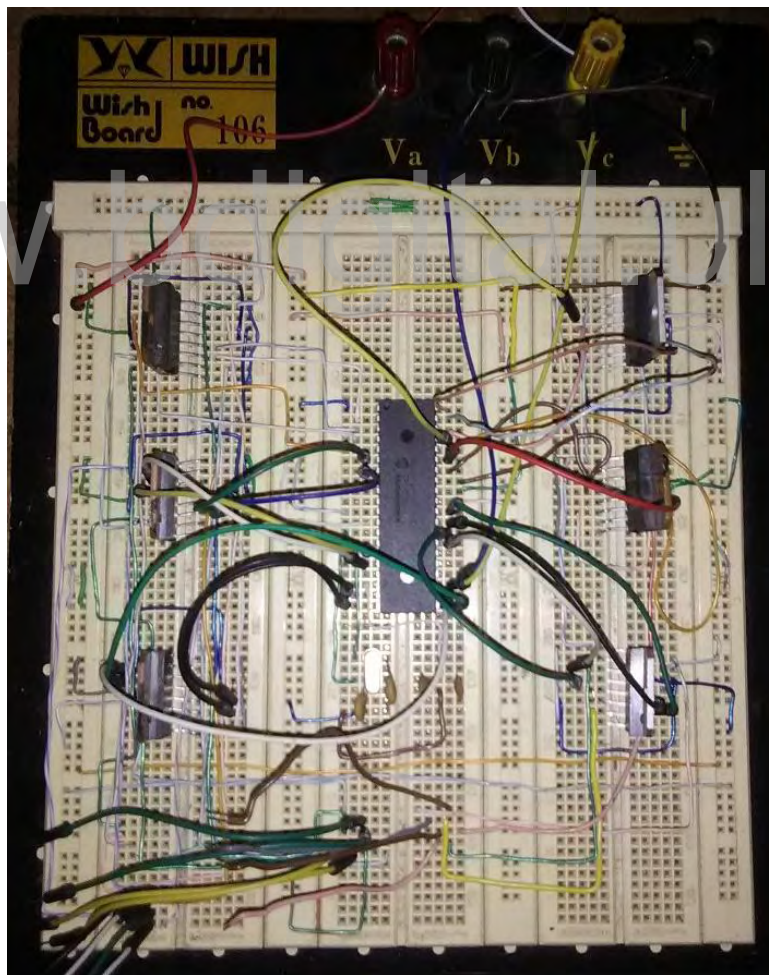


Figura 4.6. Circuito en Protoboard.

Para realizar las pruebas también fue necesario emplear una fuente de poder que cumpliera con las especificaciones eléctricas que exigen los motores de pasos, para esto empleamos una fuente de poder Nippon America DVP-712, esta fuente de poder es de 13,8V y puede suministrar una corriente de 5A, en la figura 4.7 se muestra la fuente de poder empleada para las pruebas.

Una de las primeras pruebas que se realizaron fue medir la corriente que el brazo exige a la hora de ser utilizado, esta corriente fue aproximadamente de 0,77 A, en vacío, por lo tanto, se recomienda utilizar una fuente de poder que pueda proporcionar un voltaje de 12 V y una corriente de 2 A.



Figura 4.7. Fuente de Poder.

Una vez puesto en marcha el brazo robótico se comenzó a realizar las primeras pruebas de funcionamiento, lo primero que se hizo fue variar la barra de fuerza, como se comentó en el capítulo 3, al desplazar la barra para que ejerza más fuerza el brazo se mueve más lento debido a que las bobinas permanecen más tiempo activas, cuando se selecciona un valor

menor de fuerza el brazo robótico se mueve mucho más rápido, pero se ve afectada considerablemente la fuerza ejercida, por lo tanto el brazo tiene un control inversamente proporcional.

Luego se cambió el número de pasos, se empleó medios pasos lo cual sería el control más preciso que se podría hacer con estos motores, este control consta de activar una bobina y luego la bobina ya activada y la siguiente al mismo tiempo, al ponerlo en marcha no se realizaban movimientos por lo tanto observamos que no es conveniente utilizar este control, luego se realizó el encendido de una bobina a la vez por cada paso, lo cual tampoco funciona adecuadamente, el problema de esto fue que se perdían el número de pasos y no se lograba un movimiento fluido, esto es debido a que el brazo ya tiene mucho tiempo de uso lo cual afecta al movimiento de las articulaciones; con el tiempo dichas articulaciones se vuelven menos suave esto es debido a que nunca se ha realizado mantenimiento a este brazo robótico lo cual afecta en la precisión, sumando a esto el desgaste natural que tienen los componentes.

Para lograr un movimiento fluido y que el movimiento sea constante cada vez que se le pida un paso, se tuvo que activar las 4 bobinas por cada paso, una bobina a la vez, con esta configuración se obtiene la precisión necesaria para realizar cualquier rutina, si se encienden todas las bobinas al mismo tiempo no se logra movimiento alguno.

Encender todas las bobinas al mismo tiempo es necesario si el robot estuviera ejerciendo fuerza constantemente, pero para este trabajo no es conveniente debido a que se necesitaría una fuente de poder con especificaciones de corrientes más elevadas, a parte se deberían colocar disipadores de calor a los puentes H, debido a la alta corriente circulando por ellos, por lo tanto, solo se activan las bobinas necesarias, después de realizado el movimiento se apagan todas las bobinas.

También se realizó una prueba que consiste en introducirle un número de pasos mayor al que permite la articulación, la tarjeta prototipo responde encendiendo el led de error ubicado en la parte delantera inferior del brazo robótico, y no realiza movimiento, todo esto con la finalidad de proteger de un posible daño físico al brazo robótico, ya que este consta de cadenas para trasladar la fuerza ejercida por los motores de pasos a las articulaciones, si se

llegara a pasar el número de paso estas cadenas podrían sufrir daños como romperse o estirarse, teniendo como consecuencia que los movimientos sean menos precisos.

El brazo también dispone de 2 pulsadores en el panel delantero, uno es de RESET, este pulsador está conectado directamente al MCLR del microcontrolador PIC, esto con la finalidad de que si llega a existir algún error y la tarjeta prototipo no lo detectara, fácilmente puede ser reiniciada, también existe otro pulsador, el de TEST, una vez presionado este botón la tarjeta prototipo empieza a dictar órdenes a los puentes H, esto sirve para comprobar el funcionamiento de todos los motores de pasos, al presionar este botón hay que tener cuidado de que el espacio de trabajo del brazo robótico este despejado, si hay algún objeto en el espacio de trabajo de este, puede que lo golpee o se trabe el brazo robótico y de esta manera pierda el número de pasos, teniendo como consecuencia que al terminar el test no quede en la posición inicial de trabajo, por lo tanto las restricciones de movimiento puestas por programación no serían efectivas, pudiendo causar así el daño del brazo robótico o el mal funcionamiento por no tener el número correcto de pasos.

También se probó el pulsador que se encuentra dentro del robot, este pulsador tiene una polea por donde pasa la guaya que abre y cierra la pinza, al sujetar un objeto esta guaya se tensiona moviendo la polea y activando el pulsador, al activarse el pulsador le indica a la tarjeta prototipo que el objeto ya está sujeto; se quería activar esta función lo cual no se logró, debido a que el sistema no funciona adecuadamente, se debe ejercer mucha tensión sobre la guaya para que esta logre mover la polea y active el pulsador, dicha fuerza no puede ser ejercida por el motor de pasos encargado de esta función, esto puede ser debido a que ya son motores con muchos años de servicio y han perdido facultades, también se analizó el estado de la guaya y esta, no está en buenas condiciones, sin embargo se habilitó un pin del microcontrolador PIC para realizar la conexión de este pulsador, mas no se incluyó esta función en el programa, porque si lo incluimos al dar la orden de cerrar la pinza el programa quedaría en un bucle infinito esperando a que el pulsador se active.

En el mismo panel inferior delantero se encuentra otra led el cual es el que indica si la fuente de poder está alimentando la tarjeta de control, lo cual es necesario tener en cuenta, porque si ocurre una falla en la alimentación los puentes H dejarían de realizar movimiento, el microcontrolador PIC va a seguir ejecutando las rutinas debido a que el recibe alimentación

directa del puerto USB, al apagarse los puentes H se perdería automáticamente el número de pasos por lo tanto se debería regresar el robot a la posición inicial de manera manual y reiniciar todo el equipo para que funcione adecuadamente.

En las especificaciones del brazo se muestra la cantidad de peso que se puede levantar y trasladar con dicho brazo, este peso es aproximadamente de medio kilo, una de las pruebas que se realizó fue medir la corriente que exige este brazo a la fuente; al levantar un peso de medio kilo la corriente fue de 0.98 A, esta corriente es menor a la especificada por el fabricante la cual es de 5 A, esto es debido a que no estamos activando todas las bobinas al mismo tiempo al finalizar el movimiento, también una parte importante en la reducción de la corriente fue el hecho de simplificar la circuitería, el prototipo que se realizó tiene un consumo mucho menor al de la tarjeta original debido a que posee un número significativamente menor de circuitos integrados.

La última prueba que se realizó fue poner en funcionamiento el modo consola, el cual es un terminal que simula el antiguo funcionamiento del brazo robótico, para realizar esto se preparó una rutina con las cadenas necesarias para trasladar un objeto desde una posición a otra que se encuentra un poco más elevada, a continuación, se muestran las cadenas en el orden de ejecución:

A/120/2.....Moviendo base a la mayor velocidad posible, 120 pasos sentido horario.

C/070/9.....Moviendo hombro con velocidad baja, 70 pasos hacia abajo.

F/010/9.....Moviendo codo con velocidad baja, 10 pasos hacia abajo.

M/130/2.....Moviendo Muñeca a la mayor velocidad posible, 130 pasos hacia abajo.

H/250/9.....Cerrando la pinza, con la menor velocidad (ejerciendo máxima fuerza).

D/070/2.....Moviendo hombro con la mayor velocidad, 70 pasos hacia arriba.

B/250/2.....Moviendo base a la mayor velocidad posible, 250 pasos sentido anti horario.

C/030/9.....Moviendo hombro con la menor velocidad, 30 pasos hacia abajo.

I/250/2.....Abriendo pinza con la mayor velocidad posible.

Esta rutina se puede hacer de una manera más sencilla gracias a la nueva interface de usuario en la cual se arma la cadena automáticamente al presionar los botones de movimientos y seleccionando en la barra de fuerza un numero para controlar la velocidad y la fuerza.

RECOMENDACIONES DE USO

En el momento de hacer el encendido del brazo hay que asegurarse que el área de trabajo este despejada, de lo contrario se puede tropezar o con algo y perder el sincronismo con la tarjeta prototipo lo cual ocasionaría un completo descontrol.

El brazo se debe colocar en la posición inicial, esta posición es indicada con unas flechas que están en las articulaciones del brazo, al alinearlas automáticamente está en la posición inicial, esta posición es necesaria debido a que es el punto medio de la articulación.

Se debe instalar una fuente de poder para la alimentación de los puentes H que cumpla con las especificaciones eléctricas exigidas, de lo contrario no lograríamos un correcto funcionamiento.

Para activar el modo TEST se debe tener el brazo conectado mediante vía USB al computador de lo contrario no se va a ejecutar, esto se hizo con la finalidad de que se realice el TEST cuando el brazo esté conectado al computador y esperando una orden.

Cuando se conecte el brazo robótico por primera vez en un computador se debe asegurar cual puerto COM se le asignó al brazo robótico, esto se puede verificar como se muestra en el capítulo 4, también en la parte inferior derecha saldrá una notificación donde indica que puerto se le asigno.

Al momento de levantar algún objeto con el brazo hay que asegurarse de que el peso de dicho objeto no exceda el medio kilo, ya que si lo excede puede crear calentamiento excesivo en los motores y los puentes H, también se pueden perder los pasos ya que los motores no tienen la capacidad de mover este peso, lo cual no es conveniente ya que las restricciones de movimiento se aplicarían antes de llegar al tope de la articulación o después, por lo tanto la tarjeta prototipo no sabría cuál es la posición en la que se encuentra el brazo.

CONCLUSIONES

Mediante los resultados obtenidos se puede dar como cumplidos los objetivos especificados en el capítulo 1, donde se detalla que se debe realizar un prototipo que reemplace la tarjeta de control original del brazo robótico, este prototipo se realizó con un microcontrolador PIC 18F4550 el cual es uno de los controladores más potentes y económicos del mercado, además de tener un diseño compacto y contar con comunicación vía USB, el cual es uno de los requisitos que se exige.

También se analizó la tarjeta de control anterior para estudiar cuales componentes se pueden reutilizar en el nuevo prototipo, por lo cual decidimos que los únicos elementos que se puede utilizar son los puentes H, todos los otros componentes son memorias y otros dispositivos que utiliza el microcontrolador Z80 para su funcionamiento, todos estos componentes están disponibles en el PIC 18f4550, por lo tanto, no es necesario emplear esos circuitos.

Se creó una biblioteca con las librerías necesarias para el control de la tarjeta prototipo, para ello usamos las librerías que proporciona Microchip en su página oficial, luego las descargamos y modificamos para lograr una comunicación entre el computador y el prototipo, se hizo una modificación en las librerías para que el microcontrolador PIC analizara las cadenas enviadas por el computador, aparte de esto le colocamos el nombre de la BRAZO ROBOTICO ULA.

Se diseñó todas las conexiones necesarias entre el microcontrolador PIC y los motores de pasos, para esto utilizamos puentes H (L298KV), los cuales son la mejor opción disponible debido a que tienen una gran capacidad de corriente y son muy comunes en el mercado, gracias a que estos puentes H reciben señales digitales en niveles lógicos iguales a los del microcontrolador PIC, no fue necesario agregar opto acopladores o algún otro dispositivo que permita la comunicación entre ellos. de igual manera estos puentes H pueden ser

fácilmente reemplazados por otro modelo respetando las conexiones del diseño y las características eléctricas originales mostrados en este trabajo.

También se creó un software de control, para esto se utilizó QT creator 4.6.2, este software es libre y permite crear interfaces de usuario, también tiene compatibilidad con los puertos USB lo cual es un requisito necesario para lograr el control del prototipo, el QT creator utiliza C++ como lenguaje nativo por lo tanto es una herramienta de fácil manejo y comprensión.

www.bdigital.ula.ve

RECOMENDACIONES

A pesar de que todos los objetivos fueros cumplidos, existen algunas mejoras que se le pueden realizar a este prototipo, la primera mejora es realizar el circuito impreso en una máquina y mudar todos los componentes, esto le daría más robustez y durabilidad al prototipo.

Otra mejora importante que se podría hacer es construir una fuente de poder compacta e introducirla al lado de la tarjeta prototipo, esto se puede hacer ya que la tarjeta de control actual ocupa la mitad del tamaño que la original.

También se debe hacer mantenimiento a las articulaciones para que el movimiento sea más suave, con esto lograríamos tener aún más precisión y se podrían habilitar los medios pasos de los motores.

Se debe reemplazar la guaya que transmite la fuerza del motor de pasos a la pinza debido a que este se está deshilando, también se debe reemplazar los motores de pasos que están desgastados por el uso como por ejemplo el de la pinza, ya que no tiene la suficiente fuerza para tensionar la guaya y activar el pulsador que indica que la pinza está cerrada, una vez reemplazado lo indicado se puede activar la función en el microcontrolador PIC ya que la conexión física existe solo hay que habilitarla por programación.

REFERENCIAS

[1] Forum Arduino. Reconstrucción de robot CS-113. Consultado febrero 2019 desde:

<https://forum.arduino.cc/index.php?topic=503485.0>

[2] Pallejà Cabrè, T., Teixidó Cairol, M., Font Calafell, D., Tresánchez Ribes, M., & Palacín Roca, J. (2013). Desarrollo de un caso práctico de aprendizaje combinando vision artificial y un brazo robot. *VAEP-RITA. Versión Abierta Español-Portugués, 2013, vol, 1, núm. 1, p. 61-68.*

[3] Hernández, J. (2015). *Adaptación del brazo robótico CS-113 para jugar ajedrez*. Mérida, Venezuela: UNIVERSIDAD DE LOS ANDES.

[4] Córdova, J., & Stalin, P. (2009). *Diseño y construcción de un brazo robótico a base de un PIC, para transportar objetos entre dos puntos fijos, e implementación de un sistema HMI para su monitoreo* (Bachelor's thesis, LATACUNGA/ESPE/2009).

[5] Wikipedia. Wikipedia enciclopedia libre. Consultado febrero 2019 desde:

https://es.wikipedia.org/wiki/Microcontrolador_PIC

[6] Wikipedia. Wikipedia enciclopedia libre. Consultado febrero 2019 desde:

<https://es.wikipedia.org/wiki/Microcontrolador>

[7] Electroestudio. ¿Qué es un Microcontrolador? Consultado febrero 2019 desde:

<https://www.electronicaestudio.com/que-es-un-microcontrolador/>

[8] Wikipedia. Wikipedia enciclopedia libre. Consultado febrero 2019 desde:

https://es.wikipedia.org/wiki/Motor_paso_a_paso

[9] Motores de Paso. Consultado marzo 2019 desde:

<http://manueldelgadocrespo.blogspot.com/p/motores-paso-paso.html>

[10] Sal del planeta. Motores de Paso. Consultado marzo 2019 desde:

http://www.sadelplata.org/articulos/groel_060909.html

[11] Wikipedia. Wikipedia enciclopedia libre. Consultado marzo 2019 desde:

[https://es.wikipedia.org/wiki/Puente_H_\(electrónica\)](https://es.wikipedia.org/wiki/Puente_H_(electrónica))

[12] Panamahitek. El puente h invirtiendo el sentido de giro de un motor. Consultado

marzo 2019 desde: <http://panamahitek.com/el-puente-h-invirtiendolo-el-sentido-de-giro-de-un-motor-con-arduino/>

[13] ST Microelectronics. Datasheet L298KV Quadruple Half-H Drivers. Consultado

marzo 2019 desde: <https://datasheetspdf.com/pdf-file/69646/STMicroelectronics/L298/1>

[14] Wikipedia. Wikipedia enciclopedia libre. Consultado marzo 2019 desde:

<https://es.wikipedia.org/wiki/MPLAB>

[15] Inter empresas. MPLAB X. Consultado marzo 2019 desde:

<http://www.interempresas.net/Electronica/Articulos/170506-Microchip-presenta-entorno-desarrollo-integrado-fuente-abierta-soporte-plataforma.html>

[16] Wikipedia. Wikipedia enciclopedia libre. Consultado marzo 2019 desde:

<https://es.wikipedia.org/wiki/MPLAB>

[17] Wikipedia. Wikipedia enciclopedia libre. Consultado marzo 2019 desde:

https://es.wikipedia.org/wiki/Qt_Creator

[18] Wikipedia. Wikipedia enciclopedia libre. Consultado marzo 2019 desde:

https://es.wikipedia.org/wiki/Proteus_Design_Suite

[19] Labcenter. Visual designer. Consultado marzo 2019 desde:

<https://www.labcenter.com/visualdesigner/>

[20] Labcenter. PCB. Consultado marzo 2019 desde:

<https://www.labcenter.com/pcb/>

[21] Labcenter. Simulation. Consultado marzo 2019 desde:

<https://www.labcenter.com/simulation/>

[22] Culturacion. Diferencias entre puertos USB. Consultado marzo 2019 desde:
<http://culturacion.com/cual-es-la-diferencia-entre-un-micro-usb-y-un-mini-usb/>

[23] Pololu. Puerto de alimentación 12V. Consultado marzo de 2019 desde:
<https://www.pololu.com/product/1139#lightbox-picture0J2209;main-pictures>

[24] Inventable. Conexiones de puerto USB. Consultado marzo de 2019 desde:
<https://www.inventable.eu/2013/07/05/conectar-leds-al-usb/>

[25] platea.pntic.mec.es. dimensiones del brazo CS-113. Consultado marzo de 2019 desde:
http://platea.pntic.mec.es/vgonzale/cyr_0204/cyr_01/robotica/cs113.ht

www.bdigital.ula.ve

ANEXOS

En esta sección se muestra información complementaria necesaria para la elaboración de la tarjeta prototipo.

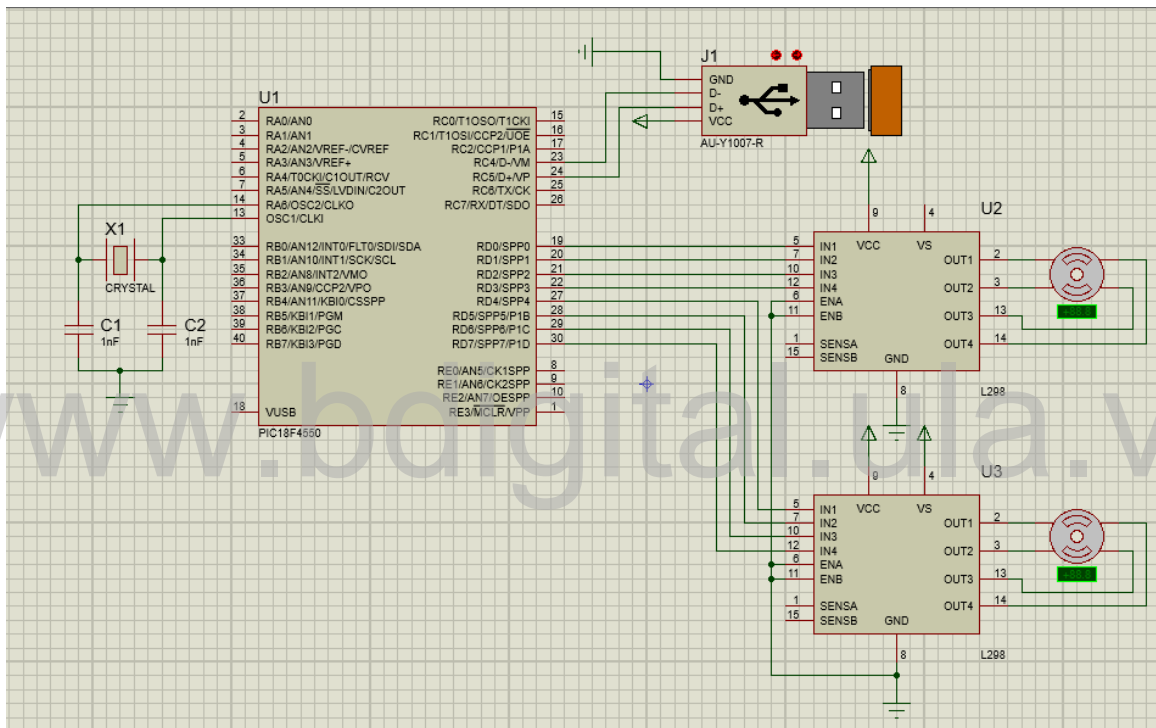


Figura 5.1. diagrama de conexiones de la muñeca.

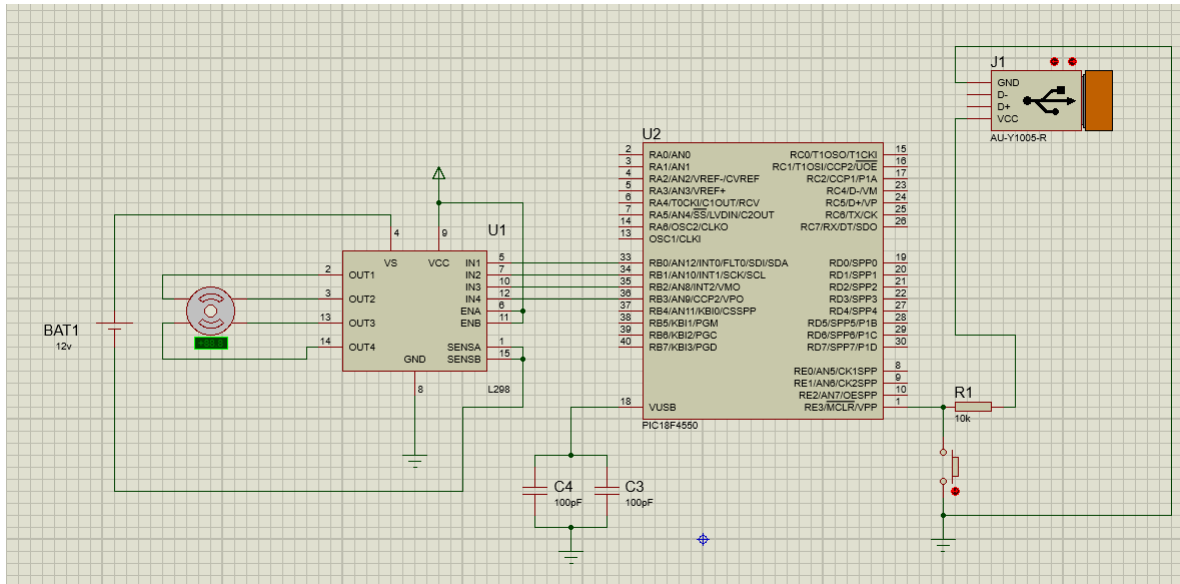


Figura 5.2 Diagrama de conexiones del base.

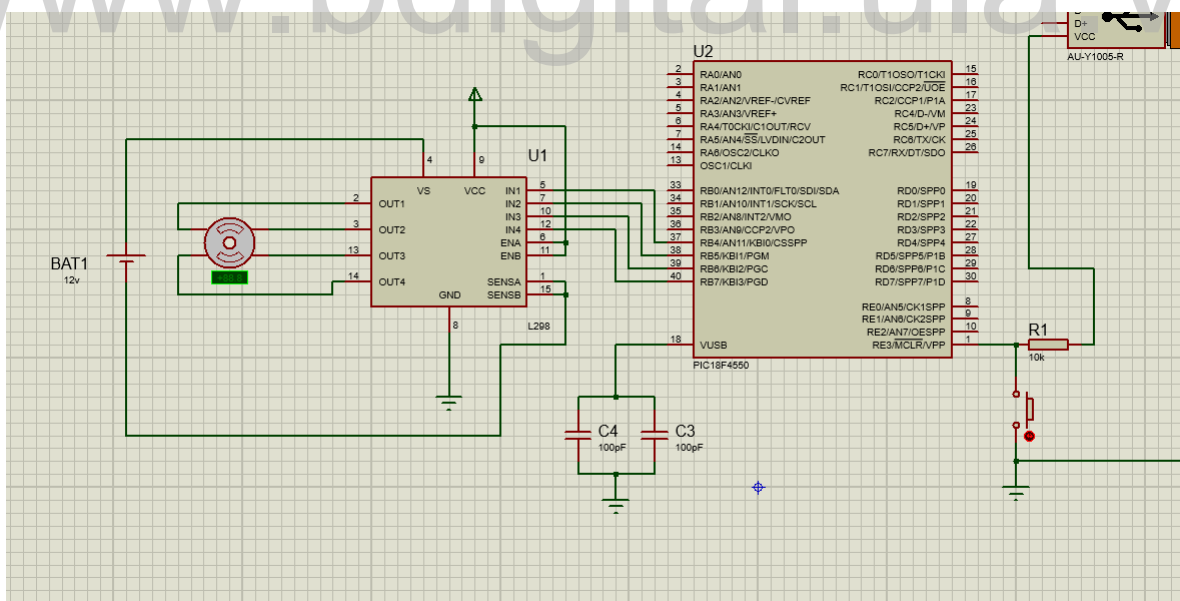


Figura 5.3. Diagrama de conexiones del hombro.

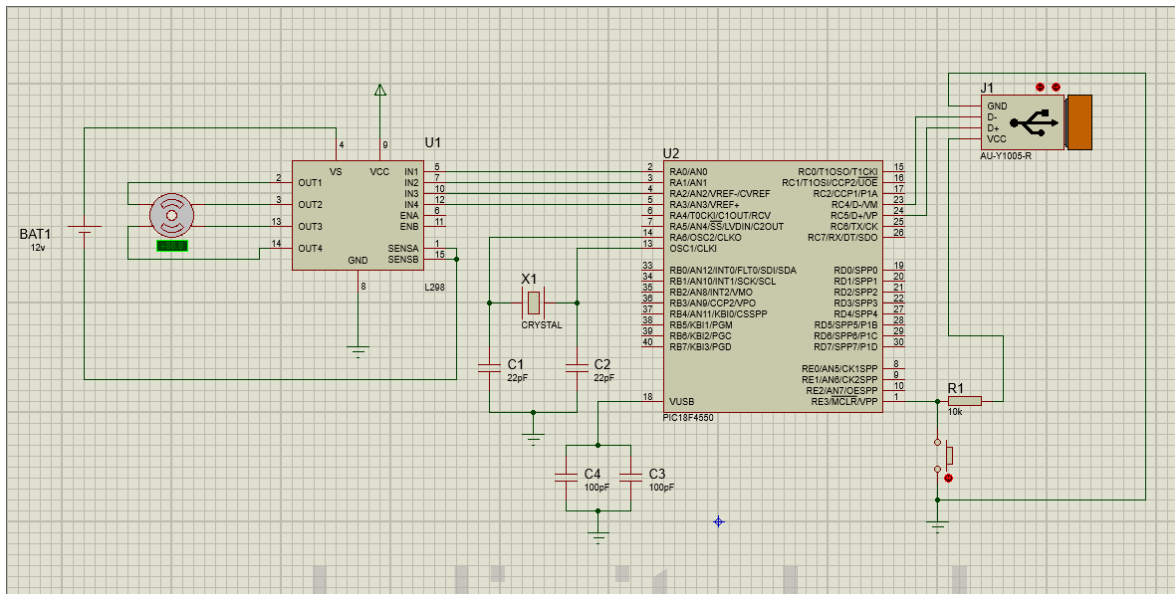


Figura 5.4. diagrama de conexiones del codo.

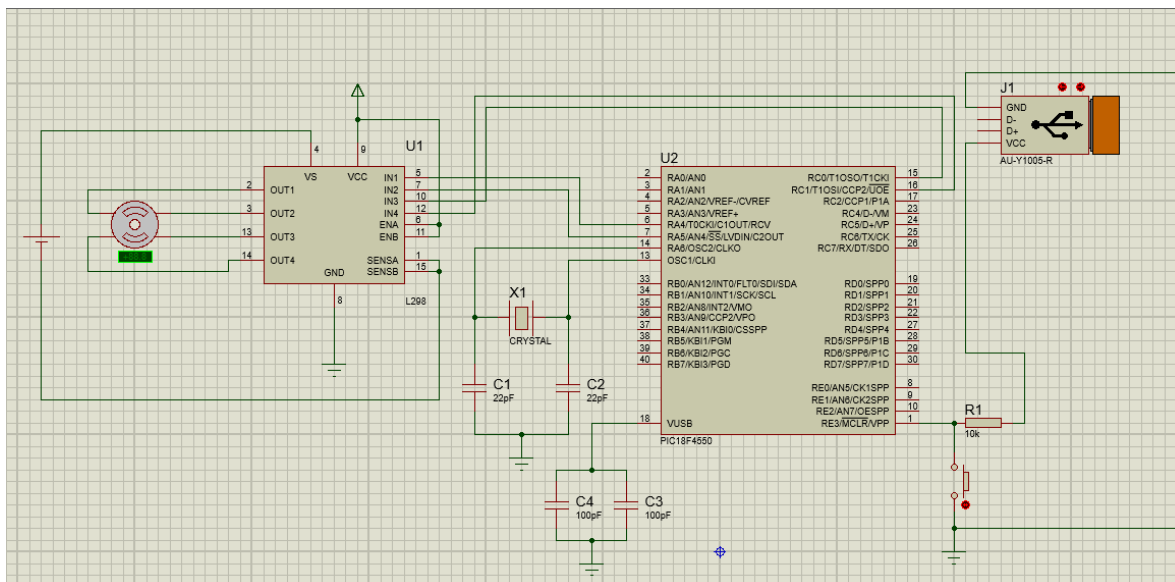


Figura 5.5. Diagrama de conexiones de la pinza.