

**UNIVERSIDAD SAN FRANCISCO DE QUITO**

**Automatización y Monitoreo de un Brazo Robótico para la Manipulación, Transporte y Clasificación de piezas en un Área de Trabajo.**

**Sofía Elizabeth Yaguana Villamarín**

Tesis de grado presentada como requisito para la obtención del título de Ingeniería Eléctrica y Electrónica con Especialización en Control.

Quito, Octubre de 2010



**© Derechos de autor (Copyright)  
Sofía Elizabeth Yaguana Villamarín  
2010**

## **DEDICATORIA**

Antes de comenzar con el desarrollo de este trabajo, quiero agradecer primero a Dios, a ese ser supremo que me ha dado la inteligencia y sabiduría necesarias para poder realizar mi trabajo de investigación final y todas las experiencias vividas durante mi vida universitaria.

Además va dirigido con especial cariño a mis padres: Luis y María Luisa; quienes han cultivado en mí la semilla del saber, la honestidad, la responsabilidad y la disciplina.

Luego un agradecimiento sincero a todos mis profesores que me han enseñado los conocimientos necesarios para utilizarlos en mi vida profesional; en especial al Ing. Omar Aguirre quien ha sido el que me ha asesorado en el desarrollo de mi tesis.

Por último quiero dedicar a todos los profesionales que con sus conocimientos aportaron a la realización de esta tesis, ya que ellos tienen la experiencia que han adquirido en todos sus años de estudios, los cuales me fueron otorgados para aplicarlos en este trabajo.

## **AGRADECIMIENTOS**

La presente tesis va dedicada en primera instancia a la Universidad San Francisco de Quito por haberme brindado la oportunidad de poder realizar mis estudios y obtener mi título profesional de Ingeniera Electrónica.

Se merecen una mención especial todos aquellos profesionales que están constantemente en busca de la ciencia para mejorar el estilo de vida de los seres humanos, cuyo trabajo debe ser reconocido y valorado por todas las personas que conformamos la sociedad.

## **ABSTRACT**

Robotics were born in recent decades to complement automation, giving an innovative element degree of intelligence.

In the industrial context, automation is a technology that is related to the use of mechanics, electronics and computers. Based on the operation and control of production. This concept, to be updated, should include the use of robots.

Industrial Robot is a automatic device to be used in the manufacturing industry, it is easily programmable to meet different operations with varying degrees of freedom. It replaces man's physical activity in repetitive, monotonous, unpleasant or dangerous tasks.

This project benefits from the control of processes that involve the handling, transportation and storage of parts, which can lose their characteristics to human touch or are dangerous to the person who handles them. It aims to show how to scale a real automated process using robotic arms.

The automated classification of parts consists of a robot OWI 535. It has two sensors, one for metal and one for plastic, this project will use a switch to simulate the sensor. The control board for the robotic arm and communication interface between the arm and the computer. This report, will tell at first, how the design, implementation and operation were made, and the entire automation process, as well as the concepts and resources needed to do it.

## RESUMEN

La robótica nace en décadas recientes para complementarse con la automatización, aportándole como elemento innovador cierto grado de inteligencia. En el contexto industrial, la automatización es una tecnología que está relacionada con el empleo de sistemas mecánicos, electrónicos y basada en la informática en la operación y control de la producción. Este concepto, para ser actualizado, debe incluir el uso de robots.

El Robot Industrial es un dispositivo de maniobra destinado a ser utilizado en la industria y dotado de un brazo, fácilmente programable para cumplir operaciones diversas con varios grados de libertad y destinado a sustituir la actividad física del hombre en las tareas repetitivas, monótonas, desagradables o peligrosas.

El presente proyecto beneficia al control de procesos que involucran la manipulación, transporte y almacenamiento de piezas que puedan perder sus características al tacto humano o sean peligrosas para la persona que los manipula. Se pretende mostrar a manera de escala un proceso real automatizado con el uso de un brazo robótico.

El proceso automatizado de clasificación de piezas se compone de un robot OWI 535, dos sensores, uno para metal y otro para plástico, en este proyecto se utilizará un interruptor para simular el sensor, la tarjeta de control para el brazo robótico y la interfaz de comunicación entre el brazo y la computadora. En este informe se va a narrar en un principio como se llevó a cabo el diseño, la implementación y funcionamiento de todo el proceso de automatización, así como los conceptos y recursos que se necesitan para poder realizarlo.

# CONTENIDO

<b>DEDICATORIA</b> .....	iv
<b>AGRADECIMIENTOS</b> .....	v
<b>ABSTRACT</b> .....	vi
<b>RESUMEN</b> .....	vii
<b>CONTENIDO</b> .....	1
1. INTRODUCCION.....	5
2. ORIGEN Y TIPOS DE ROBOTS.....	14
2.1 Definición de Robot .....	14
2.2 Esquema General del Sistema Robot.....	15
2.3 Tipos de Robot.....	18
2.3.1 Robots Manipuladores.....	18
2.3.2 Robots Móviles.....	23
2.4 Estructura de los Robots Manipuladores.....	25
2.5 Tipos de articulaciones.....	26
2.6 Estructuras básicas .....	28
2.6.1 Configuración Cartesiana.....	30
2.6.2 Configuración Cilíndrica.....	30
2.6.3 Configuración Polar o Esférica .....	31
2.6.4 Configuración Angular .....	31
2.6.5 Configuración Scara .....	31
2.7 Orientación del Efecto Final .....	32
2.8 Efectores finales .....	34
2.9 Sensores.....	35
2.9.1 Características y Clasificación .....	37
2.9.2 Sensores de Presencia y Proximidad .....	38
3. DESCRIPCION DE LOS COMPONENTES DE LA PLANTA.....	42
3.1 Robot OWI 535 .....	42
3.2 Tarjeta controladora.....	44
3.2.1 PIC 18F4550.....	45

3.2.2	L293D.....	51
3.2.3	L293B.....	52
3.2.4	Conexión USB.....	54
3.3	Interfaz de presentación.....	57
3.3.1	MICROCODE STUDIO PLUS PICBASIC PRO.....	58
3.3.2	EASYHID USB WIZARD .....	59
3.3.3	VISUAL BASIC 6.0.....	61
3.4	Dispositivo de Interfaz Humana (HID) .....	62
4.	IMPLEMENTACION.....	64
5.	PROGRAMACION.....	73
5.1	Firmware del Microcontrolador 18f4550 .....	73
5.2	Software de la PC.....	77
6.	FUNCIONAMIENTO Y PRUEBAS.....	83
7.	CONCLUSIONES Y RECOMENDACIONES.....	87
7.1	Conclusiones.....	87
7.2	Recomendaciones .....	89
	BIBLIOGRAFIA .....	90
	ANEXOS.....	91

## Lista de Figuras

Fig. 1 Distribución de los robots en la industria.....	6
Fig. 2 Demanda de robots industriales .....	8
Fig. 3 Esquema Básico de un Robot.....	16
Fig. 4 Estructura de un robot Manipulador.....	26
Fig. 5 Tipos de Articulaciones.....	27
Fig. 6 Grados de Libertad de un Robot Manipulador.....	28
Fig. 7 Configuraciones Básicas .....	29
Fig. 8 Configuración Scara.....	32
Fig. 9 Configuración angular y muñeca con tres grados de libertad .....	33
Fig. 10 Efecto final en forma de pinza .....	35
Fig. 11 Sensores ópticos de presencia y proximidad .....	41
Fig. 12 Sistema Completo .....	42
Fig. 13 Brazo Robótico OWI 535.....	43
Fig. 14 Tarjeta Controladora .....	44
Fig. 15 Microchip 18f4550 .....	47
Fig. 16 Conexión de pines del L293D .....	52
Fig. 17 Conexión de pines del L293B .....	53
Fig. 18 Diferentes tipos de conectores USB.....	57
Fig. 19 MicroCode Studio Plus .....	58
Fig. 20 Interfaz del MicroStudio Plus.....	59
Fig. 21 Interfaz del EasyHID USB Wizard .....	60
Fig. 22 Microsoft Visual Basic 6.0.....	61
Fig. 23 Interfaz de Visual Basic 6.0.....	62
Fig. 24 Sistema de funcionamiento del brazo robótico.....	64
Fig. 25 Iniciando un programa de comunicación USB en MicroCode Studio.....	66
Fig. 26 Configuración de envío de datos hacia el host.....	67
Fig. 27 Asignando nombre al proyecto .....	68
Fig. 28 Generación del Proyecto .....	69
Fig. 29 Tarjeta de control para el brazo robótico.....	72
Fig. 30 Diagrama de Flujo para el PIC.....	73
Fig. 31 Dispositivo USB encontrado.....	76
Fig. 32 Dispositivo listo para usarse.....	77
Fig. 33 Diagrama de flujo para la PC .....	78
Fig. 34 Main Form del programa .....	80
Fig. 35 Interfaz gráfica del brazo robótico.....	81
Fig. 36 Generando tareas para el brazo robótico .....	82
Fig. 37 Objeto 1 .....	85
Fig. 38 Objeto 2.....	86

## ANEXOS

ANEXO 1: Datasheet PIC 18F4550

ANEXO 2: Datasheet L293B

ANEXO 3: Datasheet L293D

ANEXO 4: Diagrama Eléctrico

## 1. INTRODUCCION

Las líneas de montaje son las estrellas de la producción. Flexibles, individualizadas, siempre buscando una alta productividad. En ellas se utilizan la mayor parte de los robots industriales. Se gesta en prensas, para tomar forma, para ensamblar las chapas y soldar. Después se pinta, se ponen las puertas y se monta. De ahí directo a los controles de calidad. Si los supera, ya está. Ha nacido un nuevo auto, listo para ser vendido al cliente. Este es el ejemplo más conocido de líneas de montaje, pero no solamente existen en el sector automovilístico. Los robots industriales se encuentran en un número determinado de puestos y en cada uno de ellos se efectúa una tarea asignada en un tiempo específico.

La tendencia en la actualidad es trabajar con líneas de montaje mixtas, aquellas en las que se producen artículos diferentes. Una misma línea debe facilitar una inmensa multitud de variantes que se fabrican bajo pedido, no por lotes. De manera que un vehículo y el que viene detrás en la producción puedan tener una ejecución diferente. El ámbito del automóvil es el que arrastra históricamente más tradición en las líneas de montaje, y uno de los que más ha adoptado técnicas propias de la filosofía de producción "just in time".

Rapidez, fiabilidad, seguridad, calidad, facilitar la labor de los operarios, mejoras en su salud, etc. La industria ya no puede vivir sin los robots y aún menos la automovilística. Este sector es el que más demanda ha experimentado hasta el

año 2006. Según datos de la Asociación Española de Robótica y Automatización Tecnologías de la Producción (ART-ATP), por primera vez se vendieron 2.527 robots, de los que algo menos de la mitad (1.256) fueron a parar a los automóviles y sus componentes. En la figura 1 se observa una distribución del uso de robots en la industria.

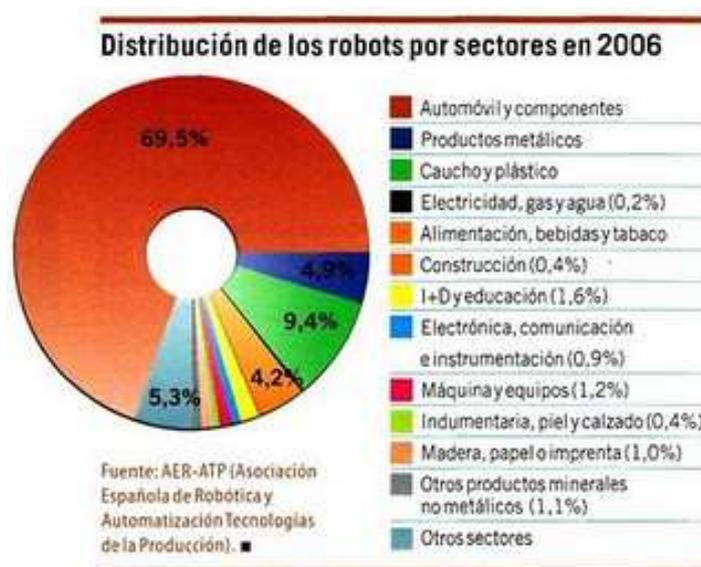


Fig. 1 Distribución de los robots en la industria

La historia de la automatización industrial está caracterizada por períodos de constantes innovaciones tecnológicas. Esto se debe a que las técnicas de automatización están muy ligadas a los sucesos económicos mundiales.

El uso de robots industriales junto con los sistemas de diseño asistidos por computadora (CAD), y los sistemas de fabricación asistidos por computadora (CAM), son la última tendencia en automatización de los procesos de fabricación. Estas tecnologías conducen a la automatización industrial a otra transición, de alcances aún desconocidos.

Aunque el crecimiento del mercado de la industria Robótica ha sido lento en comparación con los primeros años de la década de los 80's, de acuerdo a algunas predicciones, la industria de la robótica está en su infancia. Ya sea que éstas predicciones se realicen completamente, o no, es claro que la industria robótica, en una forma o en otra, permanecerá.

En la actualidad el uso de los robots industriales está concentrado en operaciones muy simples, como tareas repetitivas que no requieren tanta precisión. La Fig. 2 refleja el hecho de que en los 80's las tareas relativamente simples como las máquinas de inspección, transferencia de materiales, pintado automotriz, y soldadura son económicamente viables para ser robotizadas. Los análisis de mercado en cuanto a fabricación predicen que en ésta década y en las posteriores los robots industriales incrementarán su campo de aplicación, esto debido a los avances tecnológicos en sensores, los cuales permitirán tareas más sofisticadas como el ensamble de materiales.

Como se ha observado la automatización y la robótica son dos tecnologías estrechamente relacionadas. En un contexto industrial se puede definir la automatización como una tecnología que está relacionada con el empleo de sistemas mecánico-eléctricos basados en computadoras para la operación y control de la producción. En consecuencia la robótica es una forma de automatización industrial.

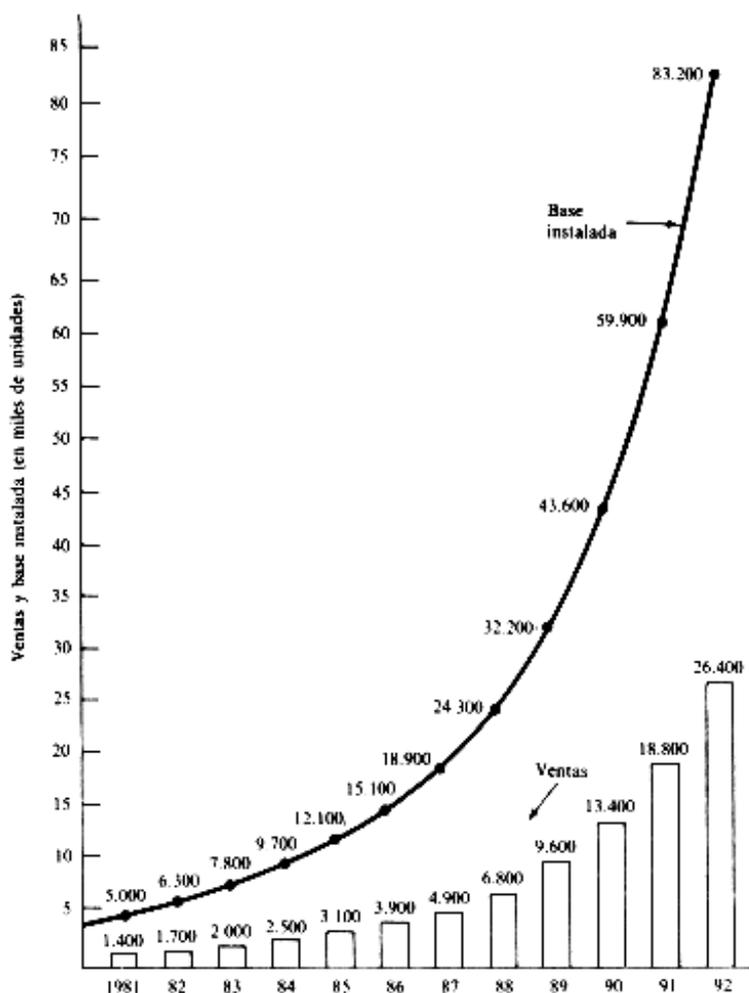


Fig. 2 Demanda de robots industriales

Hay tres clases muy amplias de automatización industrial: automatización fija, automatización programable, y automatización flexible.

La automatización fija se utiliza cuando el volumen de producción es muy alto, y por tanto se puede justificar económicamente el alto costo del diseño de equipo especializado para procesar el producto, con un rendimiento alto y tasas de producción elevadas. Además de esto, otro inconveniente de la automatización fija es su ciclo de vida que va de acuerdo a la vigencia del producto en el mercado.

La automatización programable se emplea cuando el volumen de producción es relativamente bajo y hay una diversidad de producción a obtener. En este caso el equipo de producción es diseñado para adaptarse a la variaciones de configuración del producto; ésta adaptación se realiza por medio de un programa de Software.

La automatización flexible, por su parte, es más adecuada para un rango de producción medio. Estos sistemas flexibles poseen características de la automatización fija y de la automatización programada.

Los sistemas flexibles suelen estar constituidos por una serie de estaciones de trabajo interconectadas entre sí por sistemas de almacenamiento y manipulación de materiales, controlados en su conjunto por una computadora.

De los tres tipos de automatización, la robótica coincide más estrechamente con la automatización programable.

En tiempos más recientes, el control numérico y el telecontrol son dos tecnologías importantes en el desarrollo de la robótica. El control numérico (NC) se desarrolló para máquinas herramienta a finales de los años 40 y principios de los 50's. Como su nombre lo indica, el control numérico implica el control de acciones de una máquina-herramienta por medio de números. Está basado en el trabajo original de John Parsons, que concibió el empleo de tarjetas perforadas, que contienen datos de posiciones, para controlar los ejes de una máquina-herramienta (Smid).

El campo del telecontrol abarca la utilización de un manipulador remoto controlado por un ser humano. A veces denominado teleoperador, el operador remoto es un dispositivo mecánico que traduce los movimientos del operador humano en movimientos correspondientes en una posición remota. A Goertz se le acredita el desarrollo del telecontrol. En 1948 construyó un mecanismo manipulador bilateral maestro-esclavo en el Argonne National Laboratory. El empleo más frecuente de los teleoperadores se encuentra en la manipulación de sustancias radiactivas, o peligrosas para el ser humano (ARGONNE National Laboratory).

Sin lugar a duda es evidente que el uso de la robótica en automatización en todas las formas antes citada cada día está incrementando y cambiando la forma de producir en el mundo, de igual manera se puede asumir que este tipo de tecnología tendrá grandes efectos en la industria, no sólo para los que consumidores, sino también para los productores y trabajadores. Se considera que los efectos de los sistemas automáticos y basados en robots en los sectores industrial y de servicios son de cuatro categorías: en primer lugar, probablemente afectarán a las tasas de empleo en aquellos campos de actividad en los que las tareas se conviertan en automatizadas; en segundo lugar, los modelos laborales y las características del empleo pueden cambiar, por lo cual será necesario la adquisición de nuevos conocimientos y formación; tercero, pueden producirse cambios en la organización empresarial, conforme las empresas se vayan adaptando para aprovechar todo el potencial de los sistemas robotizados; y en cuarto lugar, la robótica pudiera tener un impacto más general en la sociedad, en términos de nuevos patrones de ocio, cambios en el hogar (como resultado de la

coexistencia con robots de servicio) y una transformación del significado y valor del trabajo mismo.

En 1997, un Grupo de Estudio sobre Tendencias Sociales (SGST) llevó a cabo un estudio comparando las principales innovaciones en robótica obtenidas en estudios Delphi previos, que se realizaron en Alemania, España, Francia, Inglaterra y Japón (Peláez, Impactos de la Robotica y la Automazacion Avanzada en el Trabajo). De estos estudios se observó que el incremento gradual de las capacidades técnicas y la versatilidad de los robots, les permitirá interactuar con su medio y desarrollar los procesos de auto-aprendizaje, reparación y auto-diagnosis, de la misma manera cumplirían con las siguientes características:

- a) desarrollo gradual de las características antropomórficas para cumplir las necesidades del sector servicios;
- b) reducción del precio, los costes de programación y mantenimiento, convirtiéndolos en rentables para cualquier clase de empresa;
- c) desarrollo gradual de robots que requieran poca o ninguna programación.

Los robots industriales incrementarán su presencia en todos los sectores económicos, llevando niveles de automatización a muchas áreas de la empresa hasta llegar al 50% en el año 2010. Según las previsiones de los expertos, a largo plazo (año 2042) el número de actividades realizadas por los robots alcanzará el 80% de todos los sectores de la economía. A largo plazo, el progreso tecnológico podría permitir la expansión de la robótica al ocio y al hogar, siguiendo un

esquema similar al de la difusión de computadores personales (Peláez and Krux, The IPTS Report).

Otro de los efectos que deben considerarse, según los expertos, es una transformación del papel del trabajo y del ocio. La productividad total del sistema económico aumentaría, y la calidad de los bienes y servicios también mejoraría en cuanto al precio. Y como resultado del creciente nivel de automatización, una parte significativa de población trabajadora cambiará de actividad.

Uno de los principales impactos de la robótica en la sociedad en general, será la transformación de las estructuras productivas y de servicios. Esto finalmente haría necesarias las decisiones políticas para reducir el impacto de la robotización como provisiones de bienestar para los desempleados, nueva formación, creación de puestos de trabajo, etc. Las respuestas dadas por los expertos lo revelan clara y sistemáticamente. Será necesario alcanzar un consenso social para aprovechar la riqueza generada por la nueva economía tecnológica avanzada. Una economía con niveles más elevados de competitividad, productividad y calidad puede ser compatible con un desempleo generalizado de grandes sectores de la población.

La coincidencia en el tiempo de la automatización en los sectores industriales y de servicios y la dificultad de reciclar a muchos trabajadores reemplazados por los sistemas automáticos crearán presiones que la mayoría de los expertos considera que aumentarán durante el periodo en especial, en términos de aumento del desempleo estructural.

Es esta la razón por la que destacaron la necesidad de adaptar el sistema educativo a las nuevas demandas de la economía emergente, y simultáneamente la formación permanente y el reciclaje serán los factores principales para conseguir mejores niveles de adaptación e integración en un contexto tecnológico continuamente cambiante.

Como ya se ha explicado anteriormente; el uso de robots en la industria está creciendo velozmente, razón por la cual tuve un gran interés en hacer un proyecto donde se apliquen todas las condiciones y teorías de trabajo en cuanto a automatización y robótica concierne. En este informe se hablará del proceso que se va automatizar, así como de los componentes que se utilizarán, todo el proceso de implementación, las pruebas al sistema, y todos los resultados finales que se obtuvieron así como las mejoras que podrían realizarse a futuro.

## 2. ORIGEN Y TIPOS DE ROBOTS

### 2.1 Definición de Robot

En el término robot confluyen las imágenes de máquinas para la realización de trabajos productivos y de imitación de movimientos y comportamiento de seres vivos. Los robots actuales son obras de ingeniería y como tales concebidas para producir bienes y servicios a explotar recursos naturales. Desde esta perspectiva son máquinas con las que se continúa una actividad que parte de los propios orígenes de la humanidad y que desde el comienzo de la Edad Moderna se fundamenta esencialmente en conocimientos científicos.

En nuestro siglo el desarrollo de máquinas ha estado fuertemente influenciado por el progreso tecnológico. De esta forma se pasa de máquinas que tienen como objetivo exclusivo la amplificación de la potencia muscular del hombre, sustituyéndolo en su trabajo físico, a máquinas o instrumentos que son también capaces de procesar información, complementando o incluso sustituyendo, al hombre en algunas actividades intelectuales. Por otra parte, desde la antigüedad, el hombre ha sentido fascinación por las máquinas que imitan la figura o los movimientos de seres animados. Existe una larga tradición de autómatas desde el mundo griego hasta nuestro siglo, pasando por los autómatas de los artesanos franceses y suizos del siglo XVIII, que ya incorporaban interesantes dispositivos mecánicos para el control automático de movimientos.

El término robot aparece por primera vez en 1921, en la obra teatral R.U.R. (Rossum's Universal Robots) del novelista y autor dramático checo Karel Capek en

cuyo idioma la palabra “robota” significa fuerza del trabajo o servidumbre. Por aquellos años la producción en grandes series se había introducido en numerosas fábricas. Se discute ya del poder de las máquinas y la dominación de los hombres por las máquinas, argumento de ésta y otras obras teatrales y películas de los años veinte en los que aparecen trabajadores robóticos.

“Robots son máquinas en las que se integran componentes mecánicos, eléctricos electrónicos y de comunicaciones, y dotadas de un sistema informático para su control en tiempo real, percepción del entorno y programación” (Baturone).

Los robots industriales surgen de la convergencia de tecnologías del control automático y, en particular, del control de máquinas herramientas, de los manipuladores teleoperados, y de la aplicación de computadores en tiempo real. Mediante el control automático de procesos se pretende concebir y realizar ingenios que permitan gobernar un proceso sin la intervención de agentes exteriores, especialmente el hombre. En particular, se presentan problemas de seguimiento automático de señales de consigna mediante los denominados servosistemas. Estos sistemas generan automáticamente señales de control que tratan de anular la diferencia entre la señal de consigna y la señal medida en el proceso u objeto que se pretende controlar.

## 2.2 Esquema General del Sistema Robot

En la figura 3 se muestra el esquema básico de un robot. En ella se identifican un sistema mecánico, actuadores, sensores y el sistema de control como elemento básico necesario para cerrar la cadena actuación-medidas-actuación.

En el sistema mecánico puede distinguirse entre el órgano terminal, el brazo articulado y un vehículo. En la mayor parte de los robots industriales no existe tal vehículo, estando fija la base del brazo.

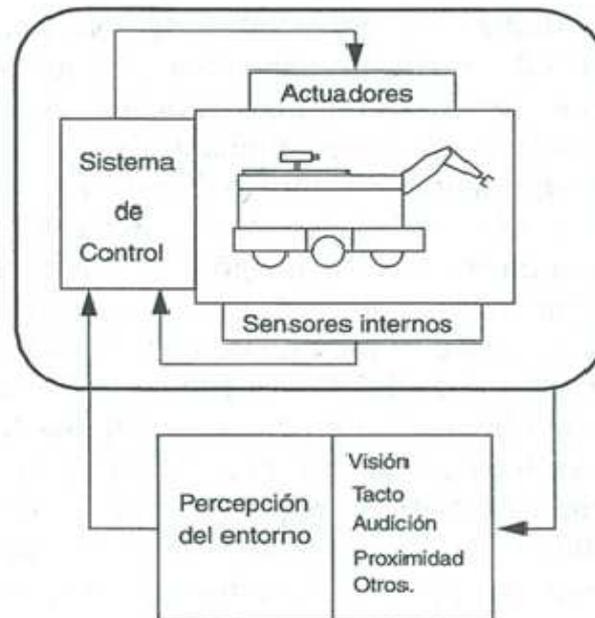


Fig. 3 Esquema Básico de un Robot

Desde el punto de vista del procesamiento de la información, en robótica se involucran funciones de control de movimiento, percepción y planificación. En un sentido amplio, el sistema de control involucra tanto bucles de realimentación de la información suministrada por los sensores internos, como del entorno. Los sensores internos miden el estado de la estructura mecánica y en particular, giros o desplazamientos relativos entre articulaciones, velocidades, fuerzas y pares. Estos sensores permiten cerrar bucles de control de las articulaciones de la estructura mecánica.

Los sensores externos permiten dotar de sentidos al robot. La información que suministran es utilizada por el sistema de percepción para aprender la realidad del entorno. Los sistemas de percepción sensorial hacen posible que un robot pueda adaptar automáticamente su comportamiento en función de las variaciones que se producen en su entorno, haciendo frente a situaciones imprevistas. Para ello el sistema de control del robot incorpora bucles de realimentación sensorial del entorno, generando automáticamente acciones en función de la comparación de dicha información sensorial con patrones de referencia.

El desarrollo de sistemas de percepción en Robótica surge a partir de los progresos tecnológicos en sensores tales como los de visión, tacto e incluso audición. Sin embargo, la percepción involucra no sólo la captación de la información sensorial, sino también su tratamiento e interpretación. Por tanto es necesario realizar una abstracción a partir de un cierto conocimiento previo del entorno. Es claro que la complejidad de la percepción artificial depende de lo estructurado que esté dicho entorno.

Por último la planificación tiene como objetivo encontrar una trayectoria desde una posición inicial a una posición objetivo, sin colisiones, y minimizando un determinado índice. En el caso más simple, el problema se plantea en un entorno que se supone conocido y estático. Se supone además que el robot es omnidireccional, que se mueve suficientemente lento y que es capaz de seguir el camino de forma perfecta.

## 2.3 Tipos de Robot

### 2.3.1 Robots Manipuladores

La mayor parte de los robots industriales actuales son esencialmente brazos articulados. De hecho, según la definición del "Robot Institute of America", un robot industrial es un manipulador programable multifuncional diseñado para mover materiales, pieza, herramientas o dispositivos especiales, mediante movimientos variados, programados para la ejecución de distintas tareas (Simpson, Friedrich and Redman) .

En la robótica subyace la idea de sustituir equipos capaces de automatizar operaciones concretas por máquinas de uso más general que puedan realizar distintas tareas. El concepto programable es también básico. La realización por programa de las funciones de control ofrece mucha mayor flexibilidad y la posibilidad de implantar funciones complejas necesarias para controlar el manipulador.

#### Sistema Mecánico

El sistema mecánico está compuesto por diversas articulaciones. Normalmente se distingue entre el brazo y el órgano terminal o efector final que puede ser intercambiable, empleando pinzas o dispositivos específicos para distintas áreas. El aumento de número de articulaciones aporta mayor maniobrabilidad pero dificulta el problema del control, obteniéndose normalmente menores precisiones por acumulación de errores. Muchos robots industriales actuales tienen menos de los seis grados de libertad de rotación o traslación que se requieren en general

para posicionar y orientar en el espacio el órgano terminal. Sin embargo, también se desarrollan manipuladores altamente redundantes con múltiples articulaciones para aplicaciones en áreas de trabajo de difícil acceso. Entre estos cabe destacar los robots tipo serpiente. Asimismo, se investiga en robots flexibles que permitan un largo alcance con un peso reducido.

En este punto conviene indicar que las ecuaciones que describen el movimiento del brazo articulado son ecuaciones diferenciales no lineales y acopladas para las que en un caso general, resulta difícil obtener soluciones analíticas. Físicamente, los términos de acoplamiento representan: pares gravitacionales de otras articulaciones y pares debidos a la aceleración de Coriolis y fuerzas centrífugas. La magnitud de estas interacciones depende de las características del brazo y de la carga.

### Actuadores

Los actuadores generan las fuerzas o pares necesarios para animar la estructura mecánica. Se utilizan tecnologías hidráulicas, para desarrollar potencias importantes, y neumáticas, pero en la actualidad se ha extendido el empleo de motores eléctricos, y en particular motores de corriente continua servocontrolados, empleándose en algunos casos motores paso a paso y otros actuadores electromecánicos sin escobillas. Existen también robots industriales de accionamiento directo que permiten eliminar los problemas mecánicos inherentes al empleo de engranajes y otras transmisiones.

Se investiga en nuevos actuadores que disminuyan la inercia, suministren un par elevado, aumenten la precisión, originen menos ruido magnético y sean bajos de peso y consumo. Por otra parte, se trata de buscar otras opciones al sistema convencional de accionamiento de articulaciones, empleándose para ello conceptos biomecánicos. De esta forma, se investiga en manipuladores con actuadores tipo músculo tanto para el brazo como para la mano del robot.

### Sensores y Sistemas de Control

Los sistemas de control de robots pueden considerarse funcionalmente descompuestos según una estructura jerárquica. En el nivel inferior se realizan las tareas de servo control y supervisión de las articulaciones. La mayor parte de los robots industriales actuales emplean servomecanismos convencionales con realimentación de posición y velocidad para generar señales de control sobre los actuadores de las articulaciones, típicamente los parámetros del controlador son fijos aunque varíen significativamente las condiciones de trabajo con la carga o con el propio movimiento. Nótese que las cargas inerciales, acoplamientos entre articulaciones, y efectos de gravedad son todos dependientes de la posición. El problema se amplía al aumentar la velocidad. Como resultado, en la mayor parte de los robots industriales actuales, la velocidad de operación debe ser pequeña,

El segundo nivel de control se ocupa de la generación de trayectorias, entendiendo por tal la evolución del órgano terminal cuando se desplaza de una posición a otra. El generador de trayectorias debe suministrar a los servomecanismos las referencias apropiadas para conseguir la evolución deseada

del órgano terminal a partir de la especificación del movimiento deseado en el espacio de la tarea. Para obtener las referencias que corresponden a las articulaciones de un determinado punto del espacio de trabajo, es necesario resolver el modelo geométrico inverso que no es lineal.

Los niveles superiores se ocupan de la comunicación con el usuario, interpretación de los programas, percepción sensorial y planificación.

Los primeros robots industriales eran programados exclusivamente por guiado manual, almacenando la secuencia de posiciones en memoria digital. La interacción con la tarea se limitaba a la apertura o cierre de una pinza u otro órgano terminal, indicándolo a un equipo externo, o esperando una señal de sincronización. Las aplicaciones típicas eran de “pick and place”, tales como la carga y descarga de máquinas, realizando tareas con movimientos absolutamente definidos y fijos. Es decir, se primaba la repetitividad sobre la adaptación. En cualquier caso, los robots podían ser reprogramados para la realización de otras tareas.

En la robótica industrial se han integrado los progresos en el control por computador de telemanipuladores, simultáneos al desarrollo de los primeros robots. Entre estos cabe destacar los trabajos de Shannon y Mynsky que, en 1958, propusieron un dispositivo, al que denominaron “sensor-controlled robot”, que consistía en un teleoperador equipado con distintos sensores conectados a un computador que le suministraba información suficiente para decidir las acciones necesarias para alcanzar un determinado objetivo (Kurfess).

Desde comienzos de los años setenta, y comienzo de los ochenta se adoptan lenguajes estructurados con herramientas de programación en tiempo real que progresivamente se introducen en los robots industriales comerciales. Así mismo se trabaja en lenguajes de programación orientados a la tarea basado en la incorporación de métodos de la inteligencia artificial para generación automática de planes, permitiendo también coordinar la actividad de un robot en sistemas de fabricación flexible.

En la década pasada surge también el concepto de realimentación visual en control de manipuladores, con aplicaciones al agarre de objetos en movimiento. Por otra parte, desde los años ochenta se progresa en la manipulación diestra de objetos mediante una mejor comprensión de la mecánica de la manipulación y su planificación.

Hoy en día existe una alta demanda en teoría de control, sistemas de percepción y aprendizaje, sistemas informáticos en tiempo real, y nuevos mecanismos, que se requieren para resolver los problemas planteados por el control de estructuras articuladas y la manipulación de objetos.

En general, junto a los progresos tecnológicos, se requieren desarrollos teóricos que permitan formular una metodología de diseño de estos nuevos sistemas de control, en los que se involucran bucles de realimentación sensorial y procesos de decisión y aprendizaje, que son difíciles de tratar con los métodos convencionales de la teoría de control.

### 2.3.2 Robots Móviles

El desarrollo de robots móviles corresponde a la necesidad de extender el campo de aplicación de la robótica, restringiendo inicialmente al alcance de una estructura mecánica anclada en uno de sus extremos. Se trata también de incrementar la autonomía limitando la intervención humana.

Desde el punto de vista de la autonomía, los robots móviles tienen como precedentes los dispositivos electromecánicos, tales como los denominados “micro-mouse”, creados desde los años treinta para desarrollar funciones inteligentes tales como descubrir caminos en laberintos.

La autonomía de un robot móvil se basa en el sistema de navegación automática. En estos sistemas se incluyen tareas de planificación, percepción y control. En los robots móviles, el problema de planificación, en el caso más general, puede descomponerse en planificación global de la misión, de la ruta, de la trayectoria y finalmente, evitar obstáculos no esperados. En un robot para interiores la misión podría consistir en determinar a qué habitación hay que desplazarse, mientras que la ruta establecería el camino desde la posición inicial a una posición en la habitación, definiendo puntos intermedios de paso. El vehículo puede desviarse de la ruta debido a la acumulación de imprecisiones mecánicas y de control.

La planificación de la trayectoria puede realizarse también en forma dinámica, considerando la posición actual del vehículo y los puntos intermedios de paso definido en la planificación de la ruta. La trayectoria se corrige debido a acontecimientos no considerados. La definición de la trayectoria debe tener en

cuenta las características cinemáticas del vehículo. Por ejemplo en vehículos con ruedas y tracción convencional, interesa definir trayectorias de curvatura continua que puedan ejecutarse con el menor error posible.

Además de las características geométricas y cinemáticas, puede ser necesario tener en cuenta modelos dinámicos de comportamiento del vehículo contemplando la interacción vehículo-terreno. Por otra parte, puede plantearse también el problema de la planificación de la velocidad teniendo en cuenta las características del terreno y del camino que se pretende seguir. Una vez realizada la planificación de la trayectoria, es necesario planificar movimientos concretos y controlar dichos movimientos para mantener al vehículo en la trayectoria planificada. De esta forma, se plantea el problema del seguimiento de caminos, que para vehículos con ruedas se concreta en determinar el ángulo de dirección teniendo en cuenta la posición y orientación actual del vehículo con respecto a la trayectoria. Así mismo es necesario resolver el problema de control y regulación de la velocidad del vehículo.

Los bucles de control se plantean tanto en el espacio de las variables articulares como en coordenadas del mundo, y las ecuaciones de movimiento son complejas, si se considera la interacción con el terreno. Mientras en manipuladores es relativamente fácil el cálculo y medida de los pares y fuerzas que se ejercen sobre la estructura mecánica, la determinación de estos pares en vehículos con ruedas es muy difícil. En la actualidad se emplean fundamentalmente métodos geométricos y modelos cinemáticos simplificados. No obstante la consideración de aspectos dinámicos es necesaria cuando la velocidad es alta.

El sistema de percepción de un robot móvil o vehículo autónomo tiene un triple objetivo: permitir una navegación segura, detectando y localizando obstáculos y situaciones peligrosas en general, modelar el entorno construyendo un mapa o representación de dicho entorno, y estimar la posición del vehículo de forma precisa. Asimismo, el sistema de percepción de estos robots puede aplicarse no solo para navegar sino también para aplicaciones tales como el control de un manipulador situado en el robot. Para el diseño de estos sistemas de percepción deben tenerse en cuenta diferentes criterios, algunos de los cuales son conflictivos entre sí. De esta forma, es necesario considerar la velocidad del robot, la precisión, el alcance, la posibilidad de interpretación errónea de datos y la propia estructura de la representación del entorno.

En muchas aplicaciones se requiere tener en cuenta diversas condiciones de navegación con requerimientos de percepción diferentes. De esta forma, puede ser necesario estimar de forma muy precisa, aunque relativamente lenta, la posición del robot y a la vez, detectar obstáculos lo suficientemente rápido, aunque no se necesite una gran precisión en su localización.

## 2.4 Estructura de los Robots Manipuladores

Los robots manipuladores son esencialmente, brazos articulados. De forma más precisa, un manipulador industrial convencional es una cadena cinemática abierta formada por un conjunto de eslabones o elementos de la cadena interrelacionados mediante articulaciones o pares cinemáticos, tal como se ilustra en la figura. 4. Las articulaciones permiten el movimiento relativo entre los sucesivos eslabones.



Fig. 4 Estructura de un robot Manipulador

## 2.5 Tipos de articulaciones

Existen diferentes tipos de articulaciones. Las más utilizadas en robótica son las que se indican en la figura 5. La articulación de rotación suministra un grado de libertad consistente en una rotación alrededor del eje de la articulación. Esta articulación es, con diferencia la más empleada.

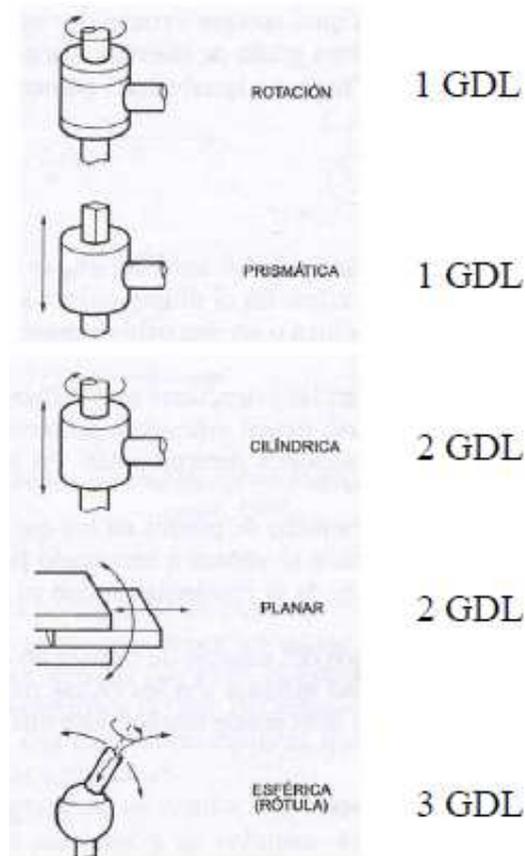


Fig. 5 Tipos de Articulaciones

En la articulación prismática el grado de libertad consiste en una traslación a lo largo del eje de la articulación. En la articulación cilíndrica existen dos grados de libertad: una rotación y una traslación. La articulación planar está caracterizada por el movimiento de desplazamiento en un plano que se muestra en la figura 6, existiendo por tanto, dos grados de libertad. Por último la articulación esférica combina tres giros en tres direcciones perpendiculares en el espacio.

Los grados de libertad son el número de parámetros independientes que fijan la situación del órgano terminal. El número de grados de libertad suele coincidir con el número de eslabones en la cadena cinemática. Así, en la figura 6a, se ilustra

una estructura con dos eslabones, dos articulaciones prismáticas y dos grados de libertad. Sin embargo, pueden existir casos degenerados, tal como se ilustra en la figura 6b en la cual se aprecia que, aunque existan dos eslabones y dos articulaciones prismáticas, tan solo se tiene un grado de libertad. Por consiguiente, en general, el número de grados de libertad es menor o igual que el número de eslabones de la cadena cinemática.

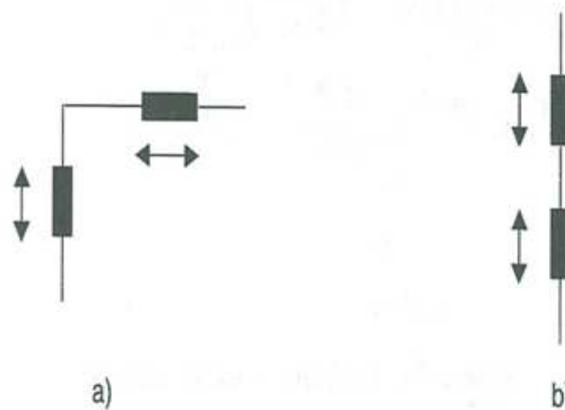


Fig. 6 Grados de Libertad de un Robot Manipulador

## 2.6 Estructuras básicas

La estructura típica de un manipulador consiste en un brazo compuesto por elementos con articulaciones entre ellos. En el último enlace se coloca un órgano terminal o efector final tal como una pinza o un dispositivo especial para realizar operaciones.

Se consideran en primer lugar, las estructuras más utilizadas como brazo de un robot manipulador.

Estas estructuras tienen diferentes propiedades en cuanto a espacio de trabajo y accesibilidad a posiciones determinadas. En la figura 7 se muestran cuatro configuraciones básicas.

El espacio de trabajo es el conjunto de puntos en los que puede situarse el efector final del manipulador. Corresponde al volumen encerrado por las superficies que determinan los puntos a los que accede el manipulador con su estructura totalmente extendida y totalmente plegada. Por otra parte todos los puntos del espacio de trabajo no tienen la misma accesibilidad. Los puntos de accesibilidad mínima son los de las superficies que delimitan el espacio de trabajo ya que a ellos solo puede llegarse con una única orientación.

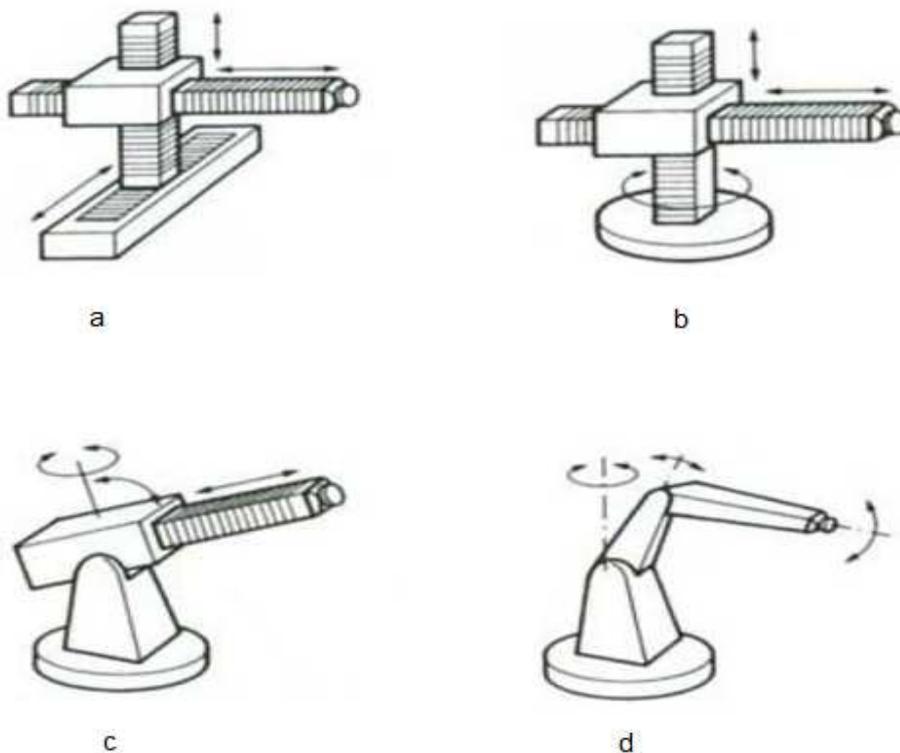


Fig. 7 Configuraciones Básicas

### 2.6.1 Configuración Cartesiana

Se ilustra en la figura 7a. La configuración tiene tres articulaciones prismáticas. Esta configuración es bastante usual en estructuras industriales, tales como pórticos, empleadas para el transporte de cargas voluminosas.

La especificación de la posición de un punto se efectúa mediante las coordenadas cartesianas  $(x,y,z)$ . Los valores que deben tomar las variables articulares corresponden directamente a las coordenadas que toma el extremo del brazo. Por consiguiente, en esta configuración, se simplifica la tarea del controlador del robot que debe generar las órdenes para ejecutar una trayectoria definida mediante una secuencia de puntos expresados en coordenadas cartesianas.

Sin embargo la configuración no resulta adecuada para acceder a puntos situados relativamente cerrados y su volumen de trabajo es pequeño cuando se compara con el que puede obtenerse con otras configuraciones.

### 2.6.2 Configuración Cilíndrica

Esta configuración tiene dos articulaciones prismáticas y una de rotación. La primera articulación es normalmente de rotación, como se muestra en la figura 7b. La posición se especifica de forma natural en coordenadas cilíndricas.

Esta configuración puede ser de interés en una célula flexible, con el robot situado en el centro de la célula sirviendo a diversas máquinas dispuestas radialmente a su alrededor.

### 2.6.3 Configuración Polar o Esférica

Esta configuración se caracteriza por dos articulaciones de rotación y una prismática. En este caso, las variables articulares expresan la posición del extremo del tercer enlace en coordenadas polares, tal como se muestra en la figura 7c. Esta configuración permite un buen volumen de trabajo.

### 2.6.4 Configuración Angular

Esta configuración es una estructura con tres articulaciones de rotación, tal como se muestra en la figura 7d. La posición del extremos final se especifica de forma natural en coordenadas angulares.

Esta estructura tiene mejor acceso a espacios cerrados y es fácil desde el punto de vista constructivo. Es muy empleada en robots manipuladores industriales, especialmente en tareas de manipulación que tengan una cierta complejidad.

### 2.6.5 Configuración Scara

Esta configuración está especialmente diseñada para realizar tareas de montaje en un plano. Está constituida por dos articulaciones de rotación con respecto a dos ejes paralelos, y una de desplazamiento en sentido perpendicular al plano. En la figura 8 se puede observar este tipo de estructura.

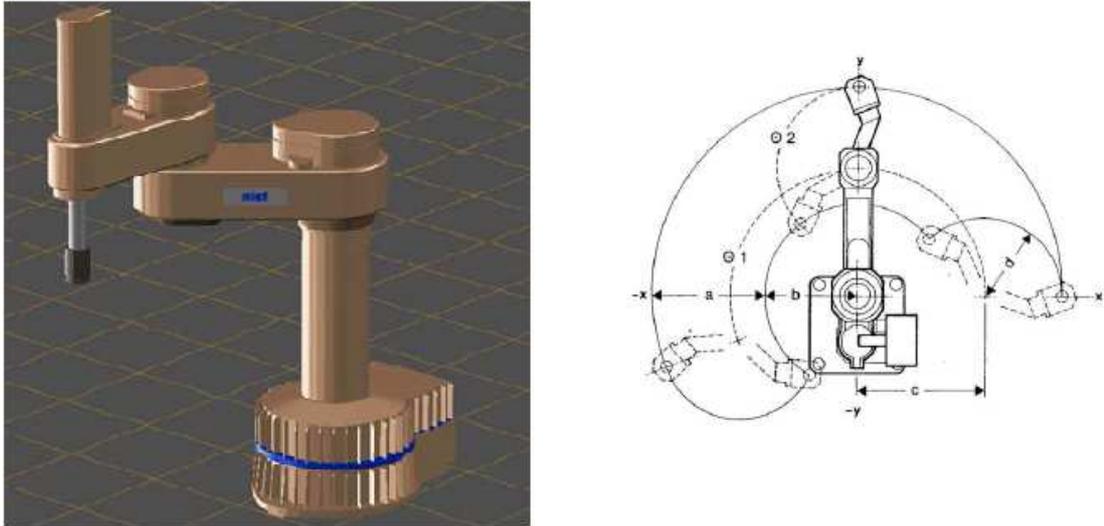


Fig. 8 Configuración Scara

## 2.7 Orientación del Efecto Final

Como se ha mencionado anteriormente, el movimiento de un brazo robótico provisto de una muñeca con un efecto final es frecuente tratarlo en dos pasos. En primer lugar, se mueve el brazo para posicionar el extremo del último enlace y, posteriormente, se orienta la muñeca para que el efecto final tenga la orientación adecuada. No obstante, existen tareas que pueden requerir el movimiento simultáneo del brazo y la muñeca.

En las tareas de montaje en un plano que sea mencionado, puede que no sea necesario ningún grado de libertad adicional ya que se trabaja siempre en dirección perpendicular al plano de montaje, tal como sucede en el montaje de componentes electrónicos.

Sin embargo en otras tareas de manipulación, suele ser necesario que el efector final tenga una determinada orientación en el espacio. Esto se consigue con la muñeca del manipulador. En la figura 9 se muestra un manipulador angular provisto de una muñeca que añade tres grados de libertad de rotación a la estructura. Estos tres ángulos permiten especificar la orientación del efector final en el espacio. Obsérvese que de esta forma, se llega a los seis grados de libertad que se necesitan en un caso general para especificar una posición y una orientación en el espacio.

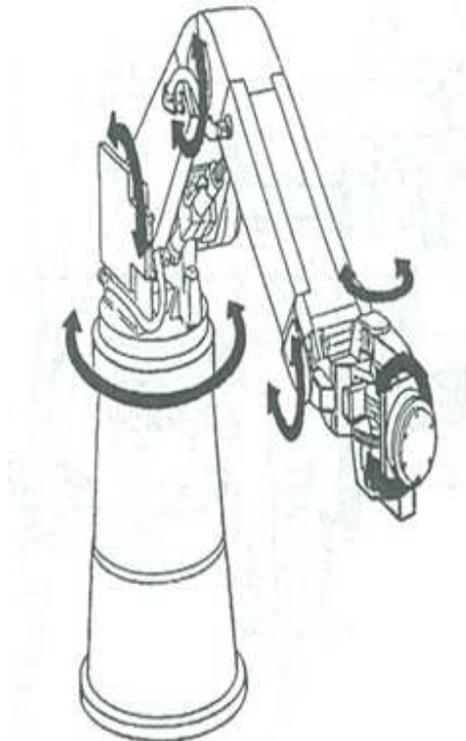


Fig. 9 Configuración angular y muñeca con tres grados de libertad

## 2.8 Efectores finales

Es el elemento que se coloca en el extremo del último enlace del manipulador y que suministra la capacidad de agarre del objeto que se pretende manipular, o la colocación de una herramienta apropiada para la tarea.

Las características que hay que tener en cuenta para su diseño son: capacidad de carga, fuerza de agarre, geometría y dimensiones de los objetos que debe manejar, tolerancias, tipos de movimientos que puede realizar, alimentación, tiempo de actuación del mecanismo de agarre y características de la superficie de contacto.

Los efectores finales más simples son pinzas mecánicas, típicamente con dos dedos y accionamiento neumático todo/nada. Se emplean también accionamientos eléctricos con control proporcional. Existen también dedos con material deformable para evitar que se produzca el corrimiento de la pieza. Asimismo, es posible utilizar sensores táctiles en los dedos, empleando un bucle de control del esfuerzo de agarre. Se dispone también de pares de dedos con diferentes cavidades para piezas de distintos tamaños. En la figura 10 se observa un efector final en forma de pinza.

Se dispone de numerosas pinzas mecánicas para su empleo en el agarre de piezas pesadas o voluminosas, dedos de apertura amplia, manos con sujeciones interiores y exteriores, o manos dobles que pueden utilizarse para soltar una pieza y agarrar otra al mismo tiempo.

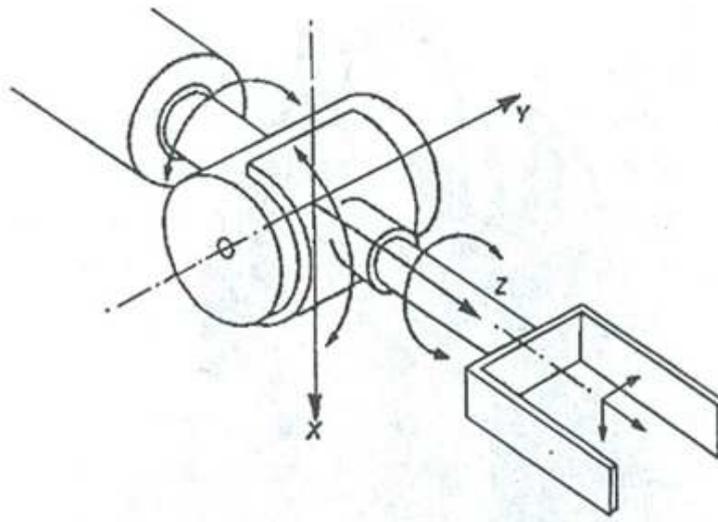


Fig. 10 Efector final en forma de pinza

## 2.9 Sensores

Los sensores permiten la adquisición de la información necesaria para el control del robot. En el estudio de los sensores debe involucrarse la medida de las magnitudes y su representación en forma compatible para su procesamiento.

En la toma de medidas siempre existe un cierto grado de incertidumbre. En principio, el incremento de la información hace posible la reducción de la incertidumbre. Para ello se trata de tomar más medida o de emplear sensores redundantes. Existen diferentes portadores de información basados en distintos principios físicos y químicos. Entre los principios y parámetros involucrados cabe mencionar:

- Mecánica: posición, velocidad, tamaño, fuerza
- Termotecnia: temperatura, calor, entropía

- Electricidad: voltaje, intensidad, resistencia, capacidad
- Magnetismo: intensidad de campo, densidad de flujo, permeabilidad
- Química: concentración de un material, estructura cristalina
- Radiación de todas las frecuencias, desde ondas de radio a rayos gama: intensidad, frecuencia, polarización, fase

Con respecto al procesamiento y transmisión de la información, pueden emplearse también distintas tecnologías con limitaciones físicas diferentes:

- Hidráulica mediante el empleo de componentes fluídicos. En este caso, existe el límite de la velocidad del sonido en un fluido, que es de aproximadamente  $10^3$  m/s.
- Eléctrica y electrónica. En la actualidad se emplean circuitos electrónicos. El límite de velocidad bien dado por la movilidad de las cargas en un material semiconductor, que es de aproximadamente  $10^5$  m/s.
- Radiante empleando componentes ópticos. El límite es la velocidad de la luz en la guía: aproximadamente:  $10^8$  m/s.

En la actualidad, se emplea casi con exclusividad el procesamiento electrónico. Para su empleo es necesario traducir las magnitudes a señales eléctricas. Cabe notar que los sensores realizan frecuentemente transformaciones de energía. Así por ejemplo en un codificador óptico, la entrada es la rotación mecánica del eje y la salida una señal eléctrica. En esta transformación se emplea un haz de luz como

fuente auxiliar de energía. Esta energía auxiliar, modulada por la rotación mecánica, produce la señal de la salida.

En numerosas aplicaciones, además de las propias magnitudes, interesa conocer sus derivadas en el espacio o en el tiempo. Para ello puede procederse a la medida de la magnitud y al cálculo de la derivada mediante procesamiento. También es posible la medida directa de la derivada, tal como la velocidad de giro de un eje mediante un tacómetro.

### 2.9.1 Características y Clasificación

El empleo de fuentes de energía adicional a la de la señal de entrada permite realizar una primera clasificación de sensores. Así, existen sensores que realizan la conversión directa de la energía de la señal. Como ejemplo, cabe mencionar las células fotoeléctricas de selenio. Para el funcionamiento básico de estos sensores no se requiere fuente de potencia adicional. Producen salida cero para entrada cero.

Sin embargo, en otros casos, se requiere una modulación con un aporte de energía con fuente diferente de la señal de entrada. Este principio es de interés para realizar medidas con señal débil. Como ejemplo, cabe mencionar un puente de galgas extensiométricas. Se necesita hacer pasar corriente por resistencias para producir la salida.

Otra clasificación posible de sensores es según el carácter absoluto o relativo de las medidas que produzcan. Así, cabe distinguir entre los sensores, tales como los potenciómetros, cuya salida es un valor absoluto que representa sin ambigüedad a

la señal de entrada, y los sensores incrementales cuya salida sólo indica la magnitud del cambio en la señal de entrada.

Entre las características más significativas para evaluar los sensores se encuentran:

- Linealidad. La característica entrada/salida es lineal. Normalmente se evalúa la separación máxima de la línea recta.
- Histéresis. La salida del sensor para una determinada entrada varía dependiendo de que la entrada esté aumentando o disminuyendo.
- Repetitividad. Es la variabilidad de la salida ante la misma entrada.
- Resolución. Es el cambio más pequeño en la entrada que puede ser detectable a la salida.
- Sensibilidad. Un pequeño cambio en la entrada causa un pequeño cambio en la salida. Normalmente se cuantifica por la relación entre el cambio en la salida dividido por el cambio de la entrada.
- Ruido. Es el nivel de señal espuria en la salida que no corresponde a un cambio en la entrada.

### 2.9.2 Sensores de Presencia y Proximidad

En primer lugar, cabe mencionar los sensores que detectan la presencia de un objeto. Los más simples son posiblemente los interruptores mecánicos tales como los habituales finales de carrera. En este caso, existe un contacto mecánico con el objeto que hace que se cierre un contacto eléctrico que genera una señal binaria.

Se emplean también otros sensores que detectan la presencia de objetos sin necesidad de que exista un contacto mecánico. Si el objeto es metálico se utilizan sensores inductivos basados en el cambio de inductancia que se produce por la presencia de un objeto de material ferromagnético en un campo creado por una bobina arrollada a un imán permanente. La presencia del objeto modifica el campo induciendo en la bobina una corriente que se detecta midiendo la tensión en la bobina. Estos sensores, que son muy utilizados en la industria, permiten detectar la presencia de objetos a distancias muy cortas.

Otros sensores para detectar la presencia de objetos materiales ferromagnéticos son los basados en el efecto Hall, que relaciona la tensión entre dos puntos de un material conductor o semiconductor con un campo magnético a través del material. Si se utiliza un imán permanente para crear un campo, el efecto Hall permite detectar dicho campo cuando no existe en él un material ferromagnético. Sin embargo, la presencia de un objeto de material ferromagnético hace que el campo se debilite, disminuyendo también la tensión a través del semiconductor, lo cual se utiliza para detectar la presencia del objeto. De esta forma, es posible construir sensores de silicio en los que se integran circuitos de detección.

Si el objeto no es de material ferromagnético, pueden emplearse sensores capacitivos, que permiten detectar cambios de capacidad inducidos por superficies cercanas. Los cambios de capacidad se detectan mediante circuitos apropiados, tales como osciladores que hacen que se inicie la oscilación cuando la capacidad es superior a un determinado umbral, o en circuitos en los cuales los desfases son proporcionales a las variaciones de capacidad.

La presencia de objetos puede detectarse también mediante sensores ópticos tipo barrera en los que el objeto interseca la línea entre el emisor y el receptor, tal como se ilustra en la figura 11a. El principio de funcionamiento de interrupción del haz es el mismo que se empleaba en los codificadores ópticos. Se emplean diodos emisores de luz y fotodetectores tales como fotorresistores, fotodiodos o fototransistores. Los fotorresistores, o fotocélulas, son resistencias cuyo valor cambia con la intensidad de la luz recibida. Su empleo y conexión a microcontroladores es muy sencilla. Los fotodiodos tienen una mayor sensibilidad, y producen una señal lineal en un rango muy amplio de niveles de intensidad. Sin embargo su salida necesita ser amplificada.

Se emplean también sensores ópticos en los que el emisor y el receptor se montan sobre el robot detectándose la presencia del objeto por la reflexión de la luz en el objeto tal como se ilustra en la figura 11b. El enfoque se consigue mediante lentes apropiadas. En particular, se utilizan sensores de infrarrojo cercano que son sensibles en longitudes de onda, tales como 880nm, inmediatamente inferiores a la visible. Sin embargo, conviene poner de manifiesto que, en principio, no suministran ninguna medida de distancia, sino tan solo una señal binaria indicando si existe o no un objeto próximo en un rango de distancia característico del sensor.

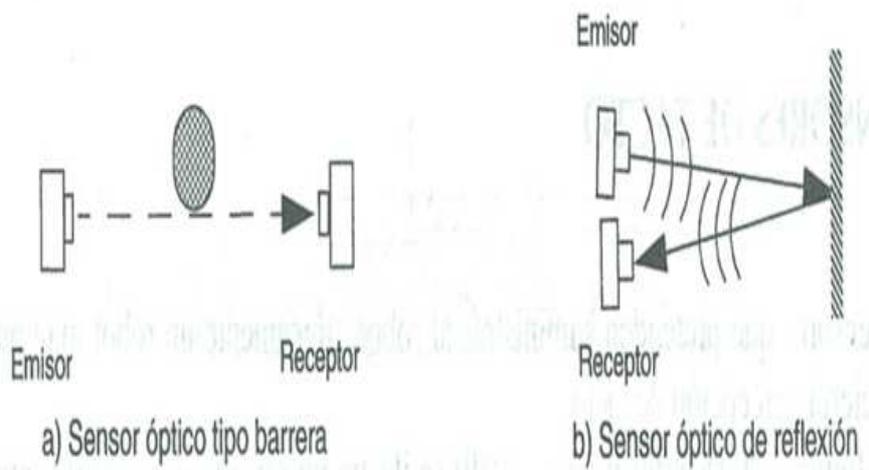


Fig. 11 Sensores ópticos de presencia y proximidad

### 3. DESCRIPCION DE LOS COMPONENTES DE LA PLANTA

La planta consta de un sistema de recolección, transporte y clasificación de piezas, en la figura 12 se puede observar el sistema completo.

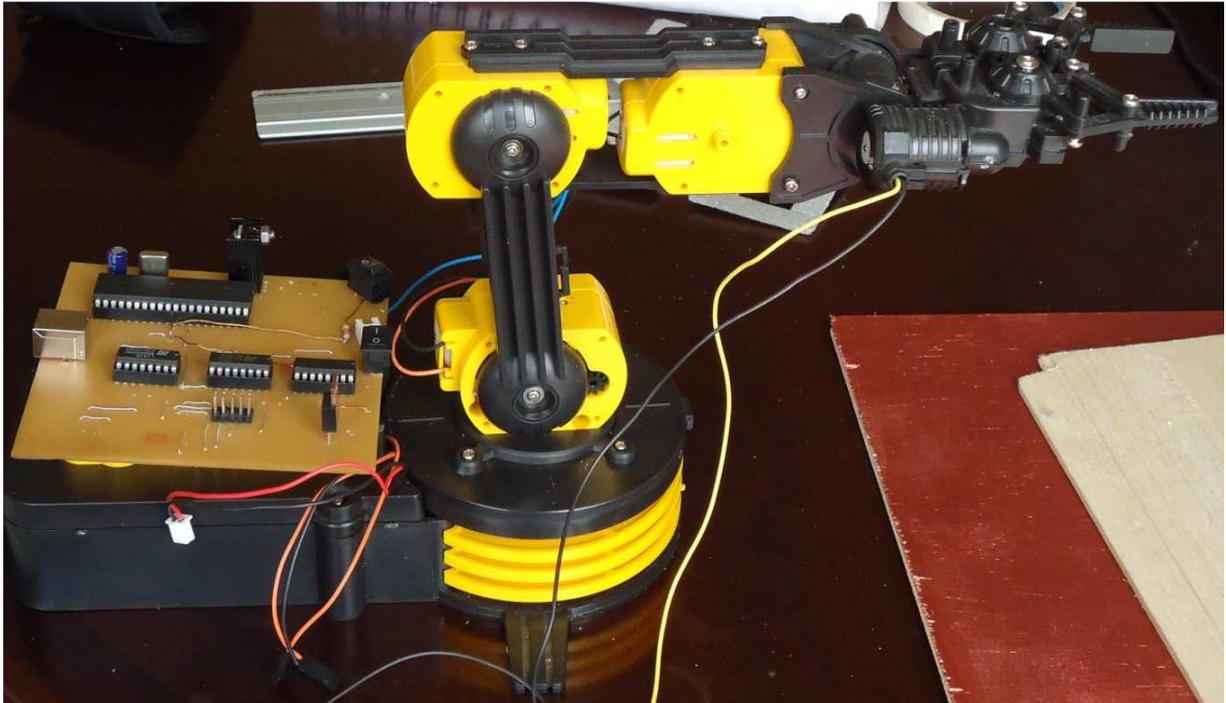


Fig. 12 Sistema Completo

El sistema está compuesto de los siguientes elementos:

#### 3.1 Robot OWI 535

La OWI-535 Brazo Robótico Edge Kit es un kit de brazo robótico a control remoto en una configuración angular. Cada motor posee una caja de cambios, el OWI 535 cuenta con cinco grados de libertad: la muñeca con un movimiento de 120°, el codo con un movimiento de 300°, el hombro con un movimiento de 180°, la base con un movimiento de 270°, y el gripper con una apertura de agarre de 0-1.77" (0-

4.5cm). Cuando una de las cajas de cambio encuentra una resistencia excesiva al movimiento, se produce un ruido que avisa que debe detener el movimiento del brazo en esa dirección. Además posee un led que iluminará lo que la pinza está sosteniendo, el led blanco está montado en la "mano" del brazo.

En la figura 13 se puede observar el brazo robótico OWI 535.



Fig. 13 Brazo Robótico OWI 535

Este robot funciona con switches, es manejado por un control, la idea principal es crear una tarjeta de control que permita, hacer los movimientos manuales de forma automática, ya sea generado por una señal externa o por un programa de computadora.

### 3.2 Tarjeta controladora

Esta tarjeta consta principalmente de un microcontrolador, específicamente el PIC 18f4550, más los drivers de los motores que son el L298n y 2 L293b y una conexión USB para tener control desde una computadora.

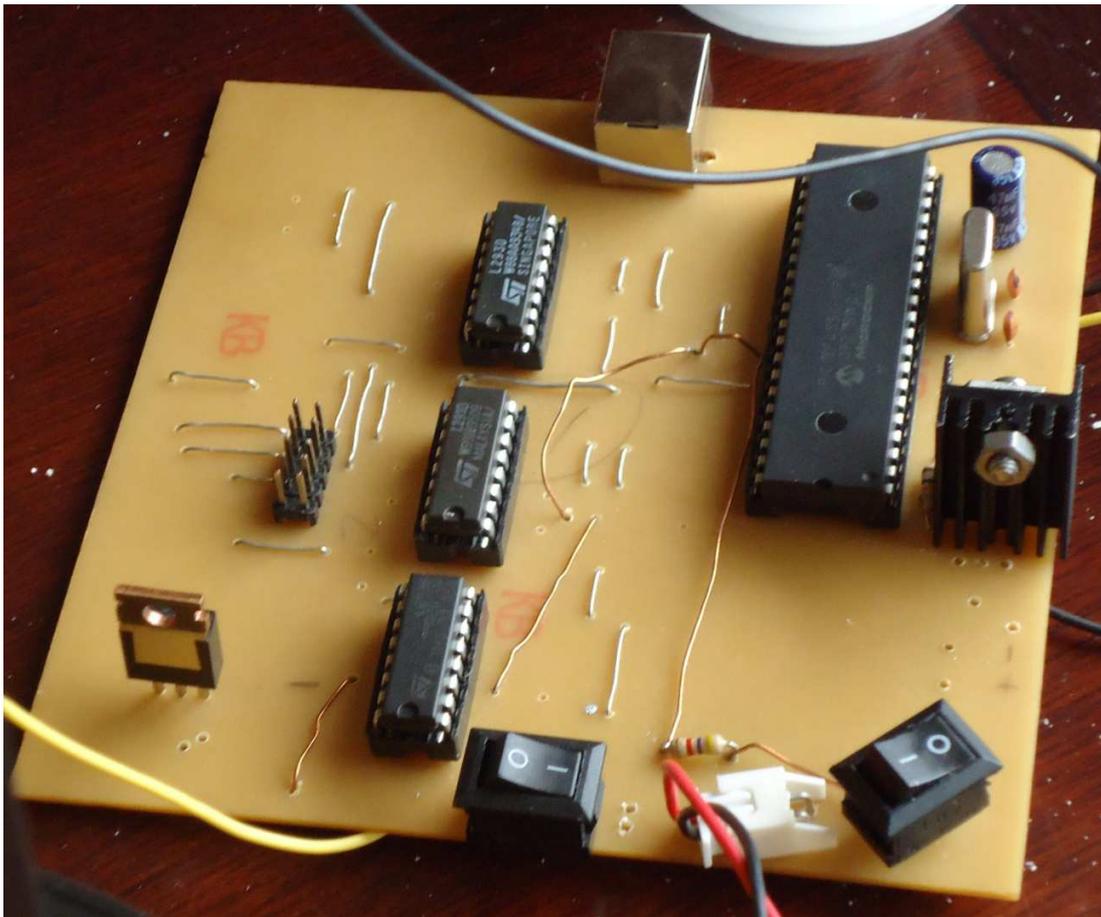


Fig. 14 Tarjeta Controladora

Se escogió el microchip 18f4550 porque tiene varios puertos de entrada y salida, tiene una gran capacidad de memoria externa y porque soporta el protocolo USB. Y se escogieron los puentes L293B y L293D por la corriente que aportan a las cargas.

### 3.2.1 PIC 18F4550

El PIC 18f4550 es PIC de la familia 18fxxx de la Empresa Microchip. Este integrado tiene una arquitectura RISC avanzada de Hardware; permite trabajar con 8 bits de datos, permite realizar hasta 32Kbytes de programa, se puede trabajar hasta una frecuencia máxima de 40 Mhz, así como también tiene un periférico de comunicación avanzado como es el USB.

En la tabla 3.1 se pueden observar las principales características del microchip 18f4550.

CARACTERISTICAS	PIC18F4450
Frecuencia de Operación	Hasta 48MHz
Memoria de Programa (bytes)	32.768
Memoria RAM de Datos (bytes)	2.048
Memoria EEPROM Datos (bytes)	256
Interrupciones	20
Líneas de E/S	35
Temporizadores	4
Módulos de Comparación/Captura/PWM (CCP)	1
Módulos de Comparación/Captura/PWM mejorado (ECCP)	1
Canales de Comunicación Serie	MSSP,EUSART
Canal USB	1
Puerto Paralelo de Transmisión de Datos (SPP)	1
Canales de Conversión A/D de 10 bits	13 Canales
Comparadores analógicos	2
Juego de instrucciones	75 (83 ext.)
Encapsulados	PDIP 40 pines QFN 40 pines TQFP 40 pines

Tabla 3.1 Características del microchip 18f4550

El integrado 18f4550 dispone de una serie de Unidades Funcionales que permiten realizar las siguientes tareas:

- Realizar tareas específicas especializadas (conversión A/D, transmisión/recepción de datos, generación de señales digitales con temporizadores programables).
- Optimizar el rendimiento del microchip, ya que estas unidades trabajan en paralelo a la CPU permitiendo que ésta se centre en otras tareas como procesado de datos, calculo, movimiento de datos, etc.

En la tabla 3.2 se puede observar las principales unidades funcionales del microchip 18f4550.

Puertos de E/S	Unidad de Comparación/Captura/PWM mejorada (ECCP)
Temporizador 0	Canal de comunicación serie EUSART
Temporizador 1	Canal de comunicación serie MSSP
Temporizador 2	Canal de comunicación serie USB
Temporizador 3	Módulo analógico de comparación
Convertidor A/D	Canal de transmisión de datos en paralelo (SPP)
Unidad de Comparación/Captura/PWM (CCP)	

Tabla 3.2. Principales Unidades Funcionales del PIC 18f4550

El PIC 18f4550 dispone de 5 puertos de entrada y salida; que incluyen un total de 35 líneas, en la tabla 3.3 se puede observar los tipos de puertos del chip 18f4550.

PUERTO	LINEAS DE ENTRADA/SALIDA
PORTA	7 LINEAS DE ENTRADA/SALIDA
PORTB	8 LINEAS DE ENTRADA/SALIDA
PORTC	6 LINEAS DE ENTRADA/SALIDA+ 2 LINEAS DE ENTRADA
PORTD	8 LINEAS DE ENTRADA/SALIDA
PORTE	3 LINEAS DE ENTRADA/SALIDA + 1 LINEAS DE ENTRADA

Tabla 3.3. Puertos E/S del microchip 18f4550

Todas las líneas digitales de E/S disponen de al menos una función alternativa asociada a alguna circuitería específica del microchip. Cuando una línea trabaja en el modo alternativo no puede ser utilizada como línea digital de E/S estándar.

La figura 15 muestra una imagen del microchip 18f4550.

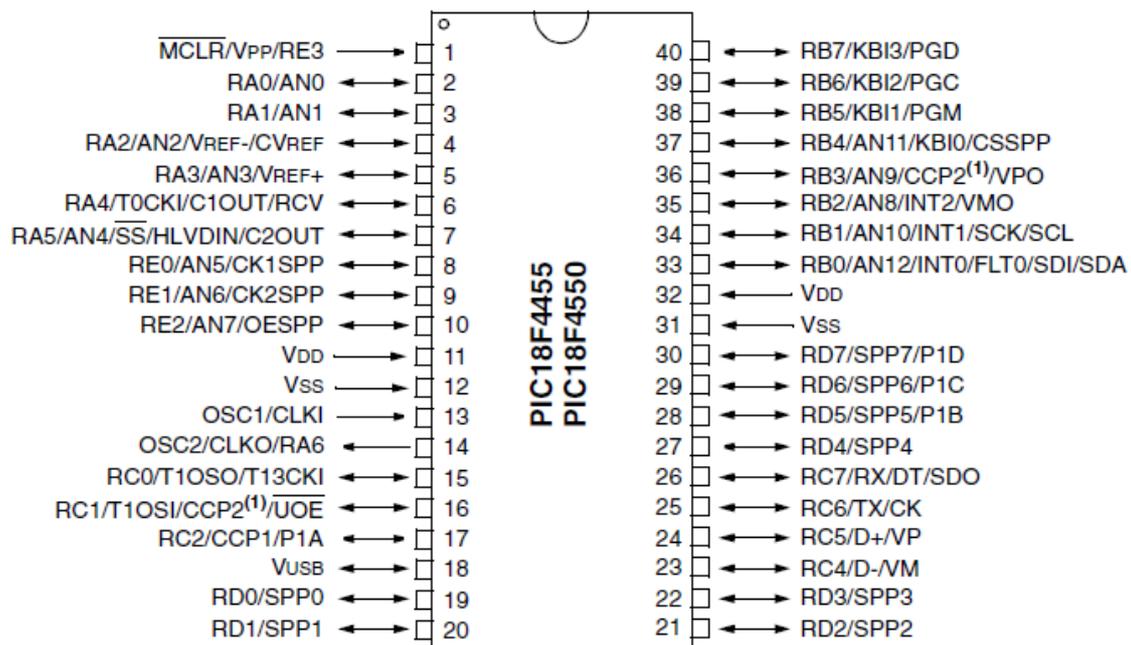


Fig. 15 Microchip 18f4550

### Descripción de los puertos

#### PUERTO A:

Dispone de 7 líneas de E/S. Las funciones alternativas son:

- RA0: entrada analógica (AN0)/ entrada de comparación (C1IN-)
- RA1: entrada analógica (AN1)/ entrada de comparación (C2IN-)

- RA2: entrada analógica (AN2)/ entrada de comparación (C2IN+)
- RA3: entrada analógica (AN3)/ entrada de comparación (C1IN+)
- RA4: entrada de reloj del Temporizador 0 (T0CKI)/salida de comparación (C1OUT)
- RA5: entrada analógica (AN4)/ salida de comparación (C2OUT)/HLVDIN entrada de detección de tensión alta/baja
- RA6: entrada del oscilador principal (OSC2)/salida de señal de reloj (CLK0)

#### PUERTO B:

Dispone de 8 líneas de E/S. Las funciones alternativas son:

- RB0: entrada analógica (AN12)/ interrupción externa 0 (INT0)/entrada de fallo del ECCP (FLT0)/entrada de datos del SPI (SDI)/línea de datos del I2C (SDA)
- RB1: entrada analógica (AN10)/ interrupción externa 1 (INT1)/línea de reloj del SPI (SDI)/línea de reloj del I2C (SDA)
- RB2: entrada analógica (AN8)/ interrupción externa 2 (INT2)/salida de datos del USB (VCMO)
- RB3: entrada analógica (AN9)/ línea de E/S del CCP2 (CCP2)/salida de datos del USB (VPO)

- RB4: entrada analógica (AN11)/ interrupción por cambio en pin (KBI0)/ salida de CS del SSP (CSSP)
- RB5: interrupción por cambio en pin (KBI1)/ línea de programación (PGM)
- RB6: interrupción por cambio en pin (KBI2)/ línea de programación (PGC)
- RB7: interrupción por cambio en pin (KBI3)/ línea de programación (PGD)

#### PUERTO C:

Dispone de 5 líneas de E/S (RC0, RC1, RC2, RC6 y RC7) y 2 líneas de solo entrada (RC4 y RC5). Las funciones alternativas son:

- RC0: salida del oscilador del Temporizador 1 (T1OSO)/ entrada de contador de los Temporizadores 1 y 3 (T13CKI)
- RC1: entrada del oscilador del Temp. 1 (T1OSI)/ línea de E/S del CCP2 (CCP2)/salida OE del transceiver del USB (UOE)
- RC2: línea de E/S del CCP1 (CCP1)/ salida PWM del ECCP1 (P1A)
- RC4: línea menos del bus USB (D-) / línea de entrada del USB (VM)
- RC5: línea más del bus USB (D-) / línea de entrada del USB (VP)
- RC6: salida de transmisión del EUSART (TX)/ línea de reloj del EUSART (CK)
- RC7: entrada de recepción del EUSART (RX)/ línea de datos síncrona del EUSART (DT)/ salida de datos del SPI (SDO)

#### PUERTO D:

Dispone de 8 líneas de E/S. Las funciones alternativas son:

- RD0: línea de datos del SPP (SPP0)
- RD1: línea de datos del SPP (SPP1)
- RD2: línea de datos del SPP (SPP2)
- RD3: línea de datos del SPP (SPP3)
- RD4: línea de datos del SPP (SPP4)
- RD5: línea de datos del SPP (SPP5) / salida PWM del ECCP1 (P1B)
- RD6: línea de datos del SPP (SPP6) / salida PWM del ECCP1 (P1C)
- RD7: línea de datos del SPP (SPP7) / salida PWM del ECCP1 (P1D)

PUERTO E:

Dispone de 3 líneas de E/S (RE0, RE1 y RE2) y 1 línea de solo entrada (RE3). Las funciones alternativas son:

- RE0: entrada analógica (AN5)/ salida de reloj 1 del SPP (CK1SPP)
- RE1: entrada analógica (AN6)/ salida de reloj 2 del SPP (CK2SPP)
- RE2: entrada analógica (AN7)/ salida de habilitación del SPP (OESPP)
- RE3: Línea de reset externo (MCLR) / línea de programación (VPP)

CONVERTIDOR ANALÓGICO-DIGITAL:

Características fundamentales:

- 10 bits de resolución
- 13 canales multiplexados
- Señal de reloj de conversión configurable

- Tiempo de adquisición programable (0 a 20TAD)
- Posibilidad de establecer el rango de tensiones de conversión mediante tensiones de referencia externas

Con el microcontrolador 18f4550 se va a controlar los movimientos del robot, el brazo robótico funciona con 5 motores DC; por tanto en total son 10 sentidos de rotación, para poder controlar cada movimiento necesitamos 10 salidas desde el PIC; para lo cual se va a utilizar las ocho salidas del puerto B y dos salidas del puerto D, mas cinco señales para habilitar a los puentes H.

Los motores funcionan con una fuente de 6 voltios, pero el microchip solo entrega 5 voltios en cada una de sus salidas, además como son 5 motores, las corrientes también se limitan, por lo que se necesita una etapa de potencia o una etapa intermedia entre el chip y los motores. Para ello se van a utilizar puentes H.

### 3.2.2 L293D

El L293D es un integrado diseñado para proveer corrientes bidireccionales de hasta 600 mA y trabaja a voltajes desde 4.5 V hasta 36 V. este dispositivo se utiliza para controlar cargas inductivas como reles, solenoides, motores dc, motores de paso.

En la figura 16 se puede observar los pines 1 y 9 que son los dos pines que habilitan las cargas independientemente, de las señales que estén en los pines de entradas. Este integrado trabaja con dos voltajes, uno para control que es de 5 voltios y otro para potencia que puede ir desde 4.5 hasta 36 voltios.

Con este integrado se pueden manejar 2 motores.

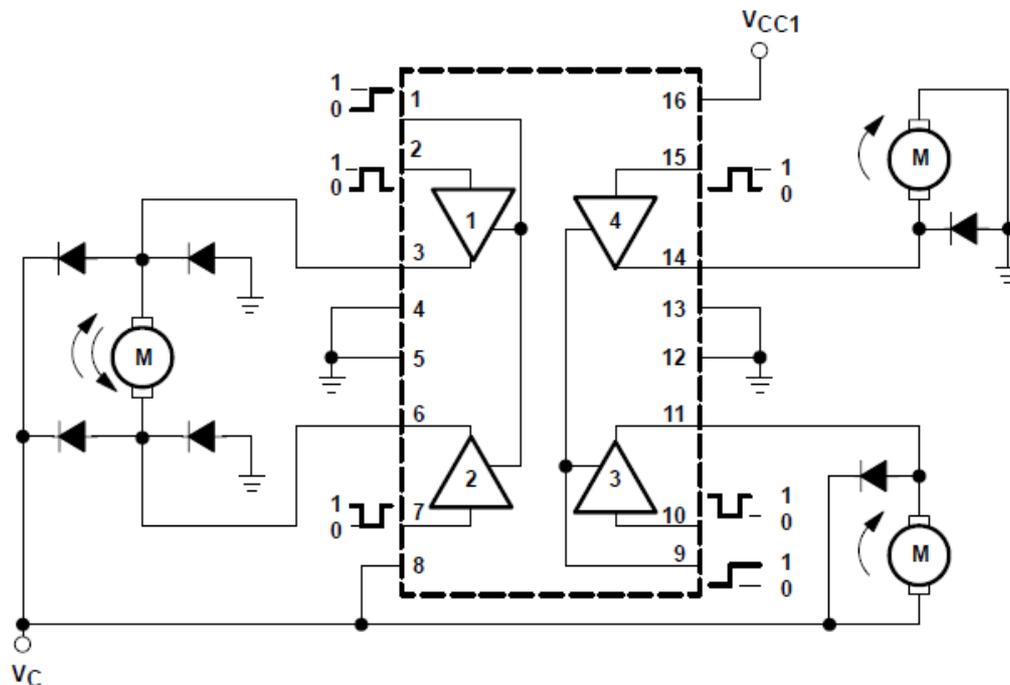


Fig. 16 Conexión de pines del L293D

### 3.2.3 L293B

El L293B es un dispositivo conductor capaz de entregar corrientes de salida de 1A por canal. Cada canal es controlado por una entrada lógica compatible con una señal TTL; y cada par de conductores (un puente completo) es equipado con una entrada que impide que se apaguen los cuatro transistores. Una entrada de alimentación separada proporciona la lógica, la cual permite que el chip ejecute las funciones a un voltaje más bajo para reducir la disipación.

El L293B es implementado en un chip de 16 pines, además usa 4 pines del centro para conducir el calor a la placa impresa.

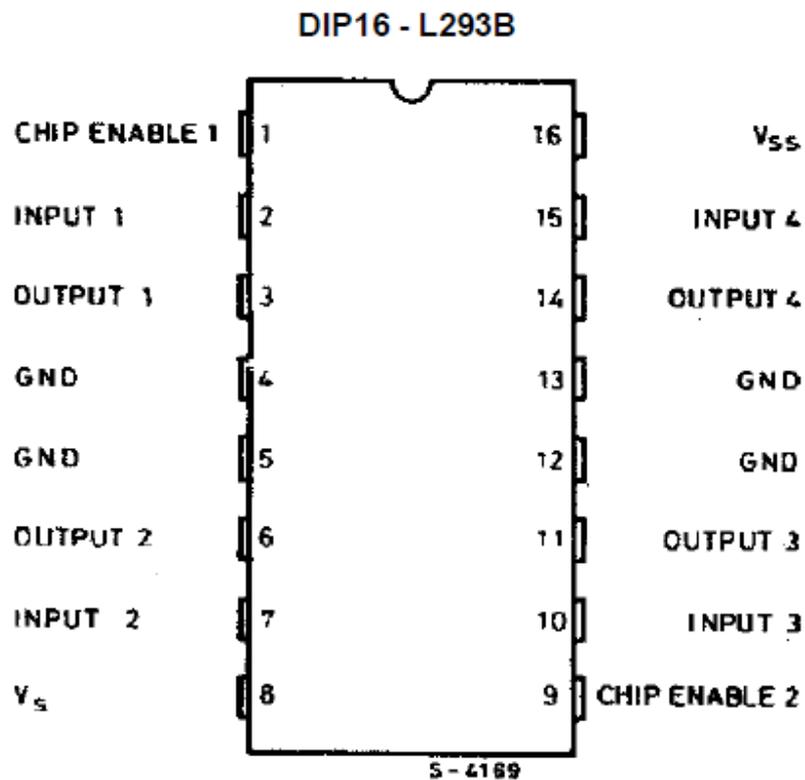


Fig. 17 Conexión de pines del L293B

En la figura 17 se puede observar los pines 1 y 9 que son los dos pines que habilitan las cargas independientemente de las señales que estén en los pines de entradas. Este integrado trabaja con dos voltajes, uno para control que es de 5 voltios y otro para potencia que puede ir desde 5 hasta 36 voltios.

Con este integrado se pueden manejar 2 motores.

Para este proyecto se necesitan 3 drivers para manejar 5 motores, por lo cual se usó 1 L298N y 2 L293B.

### 3.2.4 Conexión USB

El Bus Serial Universal (USB) es una interfaz rápida y flexible para conectar equipos a computadoras. Cada nueva PC tiene al menos un par de puertos USB. La interfaz es versátil para usar periféricas estándar como teclados, discos duros, así como componentes electrónicos más sofisticados (Axelson).

Un dispositivo USB puede usar uno de los cuatro tipos de transferencia y tres velocidades. Una vez conectado a la PC, el dispositivo debe responder a una serie de requerimientos que la PC necesita para habilitar al dispositivo y establecer la comunicación. En la PC, cada dispositivo debe tener un driver de bajo nivel que maneje las comunicaciones entre las aplicaciones y los driver de USB del sistema.

Algunas de las características que han convertido en uno de los protocolos de comunicación más usado en los últimos tiempos son los siguientes:

- Fácil de usar, no se necesita una configuración compleja y detalles de instalación.
- Rápido, la interfaz no llega a punto de cuello de botella, que disminuyen la comunicación.
- Confiable, el error es mínimo, y la corrección de errores es automática.
- Flexible, muchas clases de periféricas pueden usar esta interfaz.
- Barato
- Conservación de la potencia, para ahorrar energía en computadoras portátiles.

- Habilitado por el sistema operativo, los desarrolladores no tienen que luchar con programar drivers de bajo nivel para las periféricas que usa la interfaz.

USB es probablemente la solución en cualquier momento que se quiera usar una computadora para comunicar con dispositivos afuera de la computadora. Esta interfaz es exitosa porque permite complacer a dos tipos de audiencias: los usuarios quienes quieren usar las periféricas, y los desarrolladores quienes diseñan el hardware y escriben el código que comunica con el dispositivo.

Por el lado de la periférica, el hardware de cada dispositivo USB debe incluir un chip controlador que maneja la comunicación USB. Algunos controladores son microcomputadoras completas que incluyen un CPU y memoria que almacena un código específico para cada dispositivo que corre dentro del periférico. Otros manejan solo las tareas específicas del USB, con un bus de datos que conecta otro microcontrolador que representa las funciones no relacionadas al USB y comunica con el controlados USB que se necesita.

El periférico es responsable de responder las peticiones para enviar y recibir datos de configuración, y para leer y escribir datos cuando se lo pide. En algunos chips, algunas funciones son microdificadas en hardware y no necesita ser programado.

Los componentes físicos del USB consisten de circuitos, conectores, y cables entre el host y uno o más dispositivos. El host es una PC u otra computadora que contiene dos componentes: el controlador del host y una red por donde se transmite toda la información. Estos componentes trabajan juntos para habilitar el

sistema operativo para comunicar con los dispositivos del bus. El controlador del host formatea los datos para transmitir en el bus y trasladar los datos recibidos para ser formateados y así los componentes del sistema puedan entender.

En este proyecto se utilizó el protocolo USB, por lo que es necesario identificar qué componentes se necesitan para crear el driver de comunicación entre el brazo robótico y la computadora.

Un periférico USB necesita lo siguiente:

- Un chip controlador con una interfaz USB.
- Código en el periférico para establecer la comunicación.
- Cualquier hardware y código en el periférico que establezca sus propias funciones como procesamiento de datos, lectura de entradas y escritura en las salidas.
- Un host que soporte USB.
- Un driver del dispositivo en el host que habilite las aplicaciones para comunicar con el periférico.
- Si el periférico no es un dispositivo estándar, el host debe tener el software de aplicación que permita a los usuarios acceder al periférico.

Para desarrollar un periférico USB se necesita las siguientes herramientas.

- Un ensamblador o compilador para crear el firmware (código que se corre en chip controlador del dispositivo).
- Un dispositivo programador que permita grabar el código ensamblado o compilado en la memoria del chip controlador.

- Un lenguaje de programación y ambiente de desarrollo en el host para escribir y correr el software del host.
- Un programa que monitoree, un analizador de protocolo, que ayude a desarrollar el hardware.

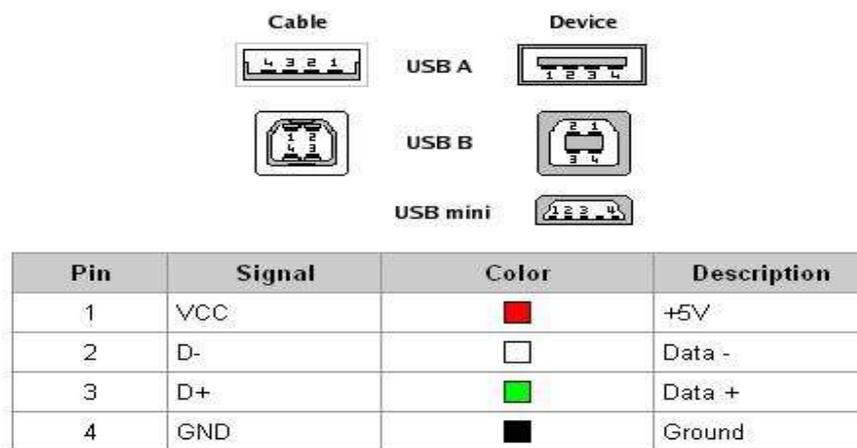


Fig. 18 Diferentes tipos de conectores USB

En la figura 18 se puede apreciar los diferentes tipos de conectores USB.

### 3.3 Interfaz de presentación

La interfaz de presentación es un programa de fácil entendimiento que le permite al usuario observar y manejar al brazo robótico, sin la necesidad de entender toda la circuitería y programación que se encuentra atrás de la tarjeta de control.

Las herramientas para el desarrollo de este proyecto son:

- MicroCode Studio Plus PicBasic PRO
- EasyHID
- Visual Basic 6.0

A continuación hablaremos de cada uno de estos programas, como el tema no es los entornos de programación no se profundiza en la explicación detallada de cada uno ellos, sino que se explica las herramientas que se utilizó para la realización del proyecto.

### 3.3.1 MICROCODE STUDIO PLUS PICBASIC PRO

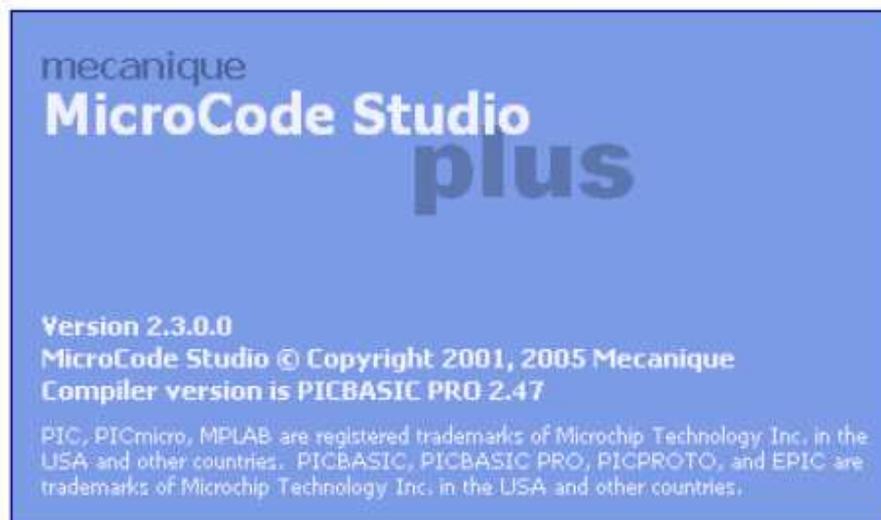


Fig. 19 MicroCode Studio Plus

El MicroCode Studio Plus es una interfaz en la cual se escribe el código del programa para el microcontrolador, en este caso el PIC 18f4550, la programación se la realiza en lenguaje Basic. Este programa corrige errores de sintaxis. El MicroCode Studio se enlaza con el PICBASIC PRO. De esta forma cuando el

programa ha sido terminado, se compila y se genera el archivo \*.HEX, los programas son guardados en formato en PicBasic \*.BAS.

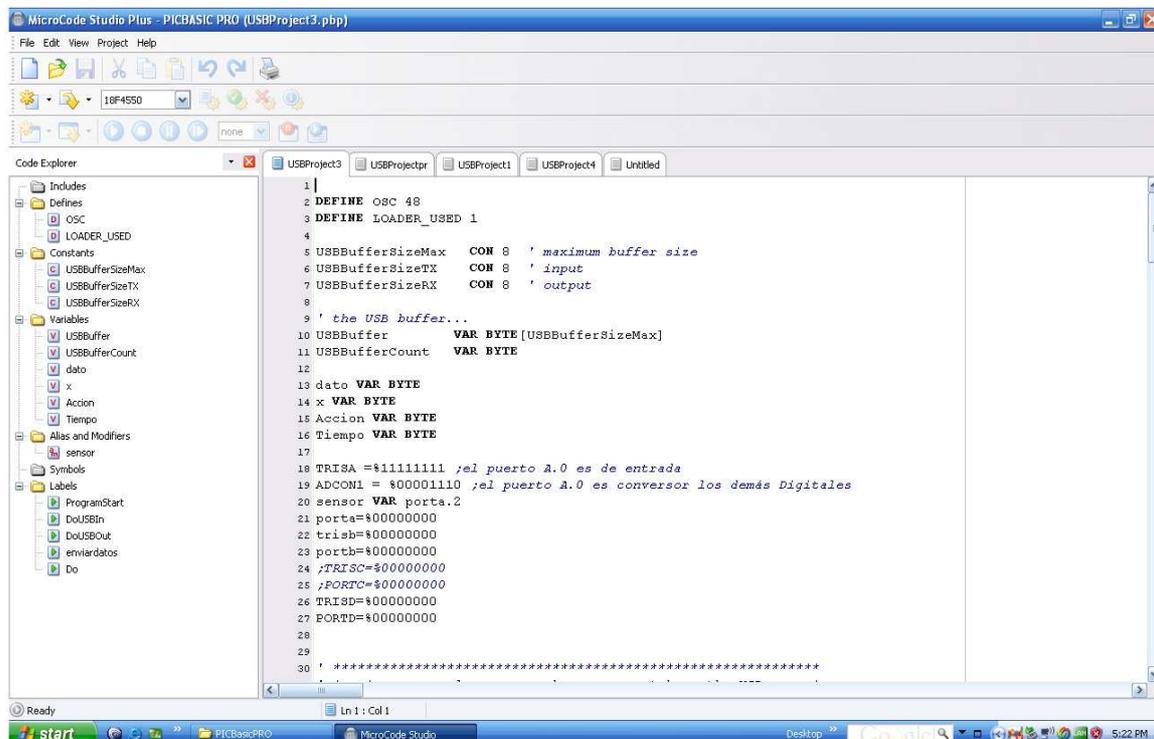


Fig. 20 Interfaz del MicroStudio Plus

El PicBasic es el compilador que transforma un archivo \*.BAS a \*.HEX. el programa cuenta con su propio set de instrucciones, muy fáciles de usar, utiliza comandos típicos de Basic como IF...FOR..., etc. Por otro lado tiene comandos para poder realizar comunicación USB.

### 3.3.2 EASYHID USB WIZARD

EL EASYHID es un wizard que nos ayuda a comunicar un microcontrolador con el PC vía USB; generando dos programas. El primer programa es un software

implementado en Visual Basic para el computador, el segundo programa es el Firmware para el microcontrolador 18f4550.

Es necesario ingresar algunos parámetros para que el Wizard genere una identificación ID, para que el computador pueda reconocer al microcontrolador como un HID (Dispositivo de Interfaz Humana), muy similar a una impresora, o un teclado USB.

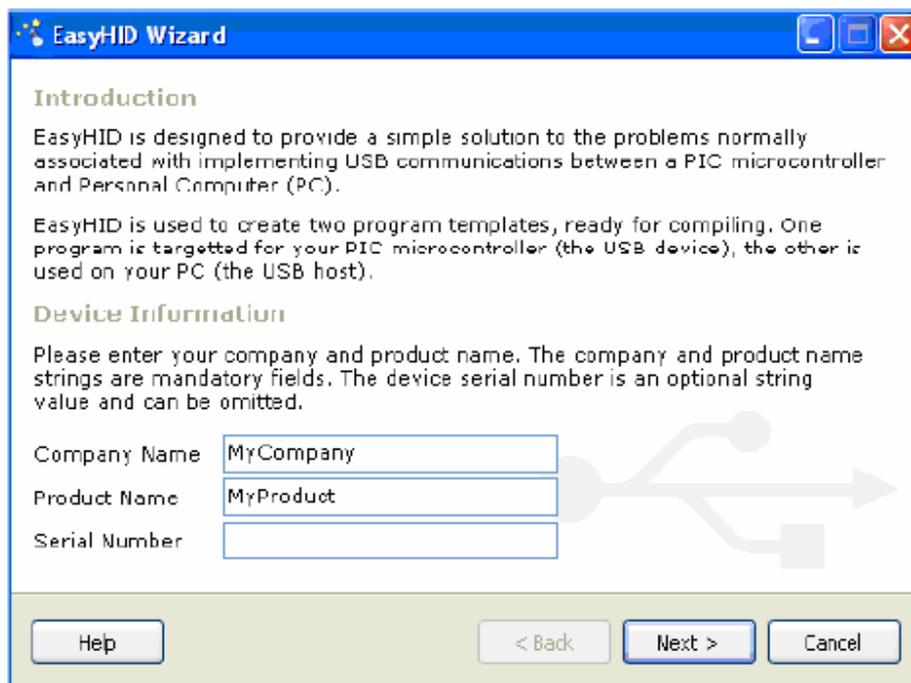


Fig. 21 Interfaz del EasyHID USB Wizard

### 3.3.3 VISUAL BASIC 6.0

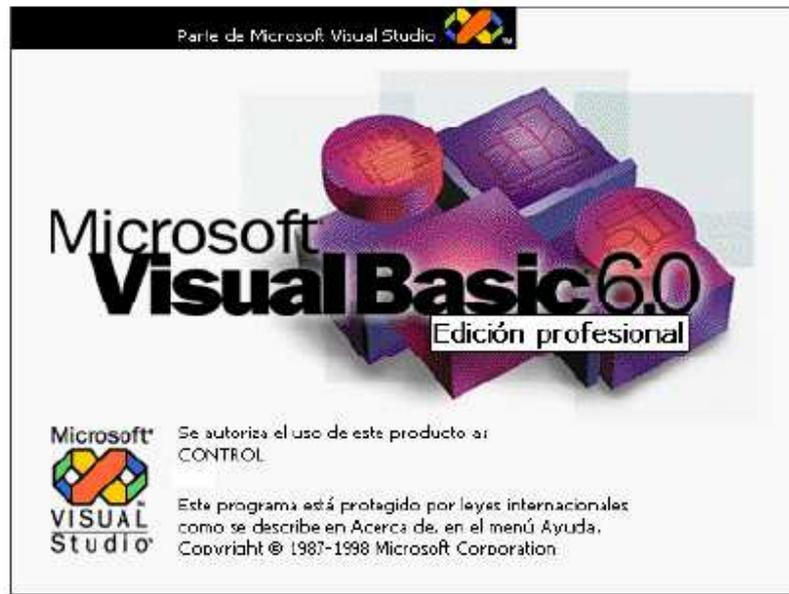


Fig. 22 Microsoft Visual Basic 6.0

Visual Basic es un programa con una interfaz grafica de usuario para crear aplicaciones para Windows basado en el lenguaje Basic y en la programación orientada a objetos.

La palabra “Visual” se refiere al método que se utiliza para crear la interfaz grafica de usuario; pues ya no es necesario escribir demasiadas líneas de código para realizar una interfaz grafica; pues ahora solo se utiliza el Mouse para arrastrar y colocar objetos prediseñados dentro de un formulario; por ejemplo un botón.

Por otro lado la palabra “Basic” significa (Beginners Allpurpose Symbolic Instruction Code). Una de las ventajas de este programa es que sus instrucciones, funciones, están relacionadas directamente con la interfaz grafica de Windows (Microsoft Corporation).

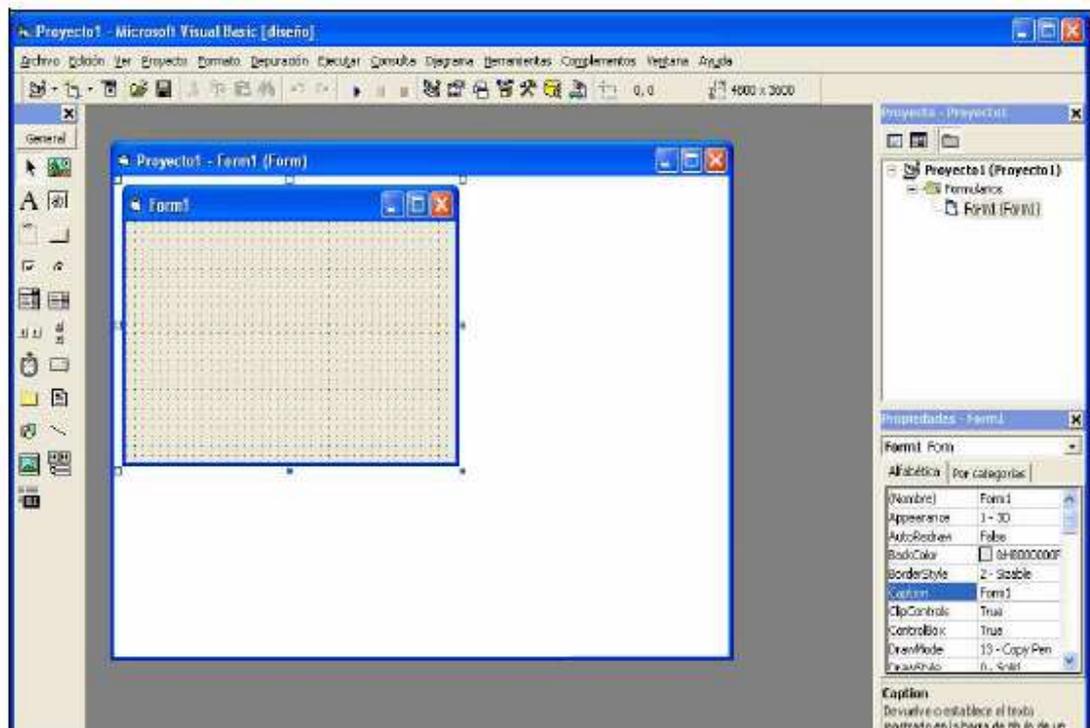


Fig. 23 Interfaz de Visual Basic 6.0

Una de las razones para utilizar este programa es que el código generado por EasyHID es para ejecutarlo en Visual Basic y poder realizar la comunicación USB con el microcontrolador PIC 18f4550.

### 3.4 Dispositivo de Interfaz Humana (HID)

Los HIDs son una clase de dispositivos USB que proporcionan una estructura a los datos que serán transferidos entre el Host y el Dispositivo. Durante el proceso de enumeración (Proceso en el que se conecta el dispositivo al Host) el dispositivo describe la información que puede recibir y enviar. Esto permite que el Host maneje los datos que son recibidos desde el dispositivo USB sin necesitar un

driver del mismo. Ejemplos típicos de dispositivos HID son: teclados, ratones, etc. Entonces no es necesario un driver para este tipo de dispositivos, pues el sistema operativo sabe exactamente qué función va a cumplir. Lo que si se requiere es que el host cuente con los drivers USB pertenecientes a este tipo de clase USB. Y estos son: hid.dll, hidclass.sys, usbstor.sys.

## 4. IMPLEMENTACION

Para la realización del proyecto se procedió como primer paso a conseguir los elementos.

El primero en armarse fue el brazo robótico, que es enviado por partes desde la fábrica, el brazo robótico fue construido con ayuda de la guía y es controlado manualmente mediante un control remoto, el control remoto posee 5 switches que permiten realizar los siguientes movimientos:

- SW1: Girar la Base
- SW2: Movimiento del Hombro
- SW3: Movimiento del Codo
- SW4: Movimiento de la Muñeca
- SW5: Movimiento del Gripper

La figura 24 muestra el sistema de funcionamiento de brazo robótico.



Fig. 24 Sistema de funcionamiento del brazo robótico

Como se puede observar en el esquema; la PC debe enviar datos al microchip para poder mover el brazo. Primero se mostrara como está la conexión entre el pic y los motores, y que datos se debe enviar desde la PC al PIC.

Para la tarjeta de control se hizo la siguiente asignación:

ASIGNACION DE VARIABLE PARA EL PIC	ARTICULACION	COLOR DE CABLES PARA RECONOCIMIENTO
M1	BASE	Negro/Naranja
M2	HOMBRO	Negro/Naranja
M3	CODO	Negro/Azul
M4	MUÑECA	Negro/Azul
M5	GRIPPER	Negro/Amarillo

Ahora se observará como está hecha la conexión de los motores al PIC.

PortB.7 = M4

PortD.7 = X

PortB.6 = M4

PortD.6 = X

PortB.5 = M3

PortD.5 = X

PortB.4 = M3

PortD.4 = X

PortB.3 = M2

PortD.3 = X

PortB.2 = M2

PortD.2 = X

PortB.1 = M1

PortD.1 = M5

PortB.0 = M1

PortD.0 = M5

Ahora es necesario describir cuales son los pasos que se deben seguir en el programa para poder realizar la comunicación USB. Primero es necesario ejecutar el MicroCode Studio.

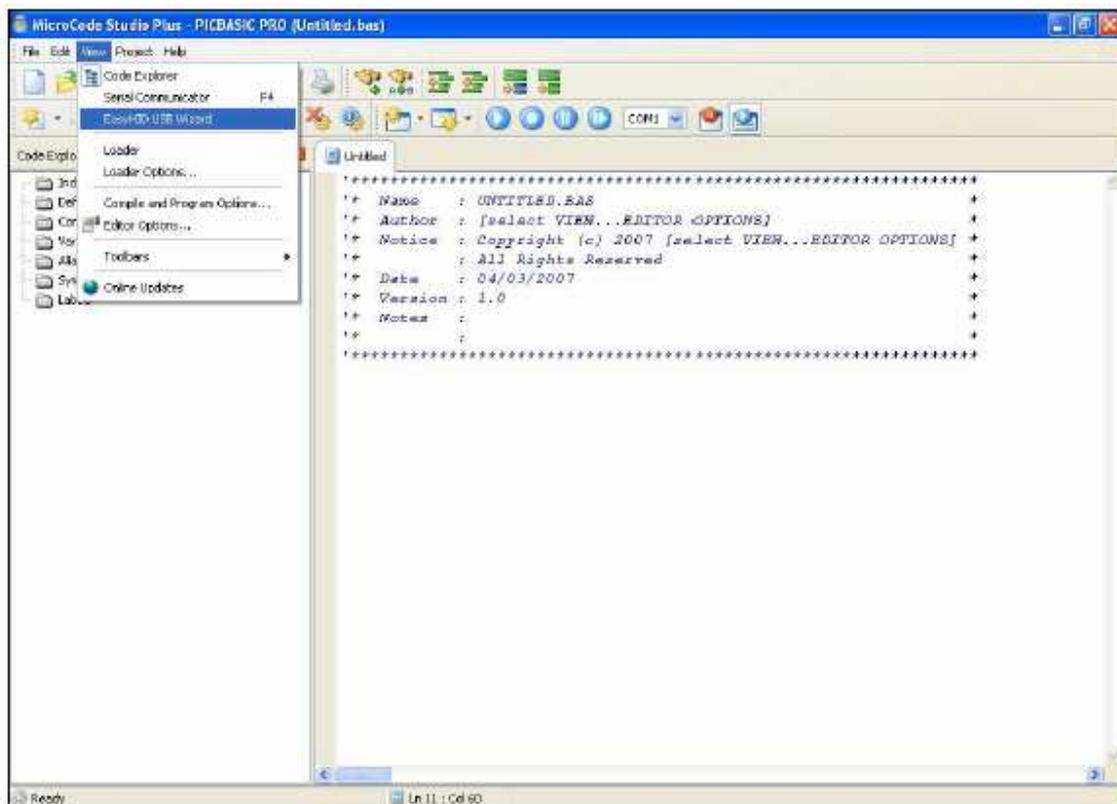


Fig. 25 Iniciando un programa de comunicación USB en MicroCode Studio

En la figura 25 se muestra la interfaz en MicroCode Studio, y ahora es necesario ejecutar el EasyHID Wizard.

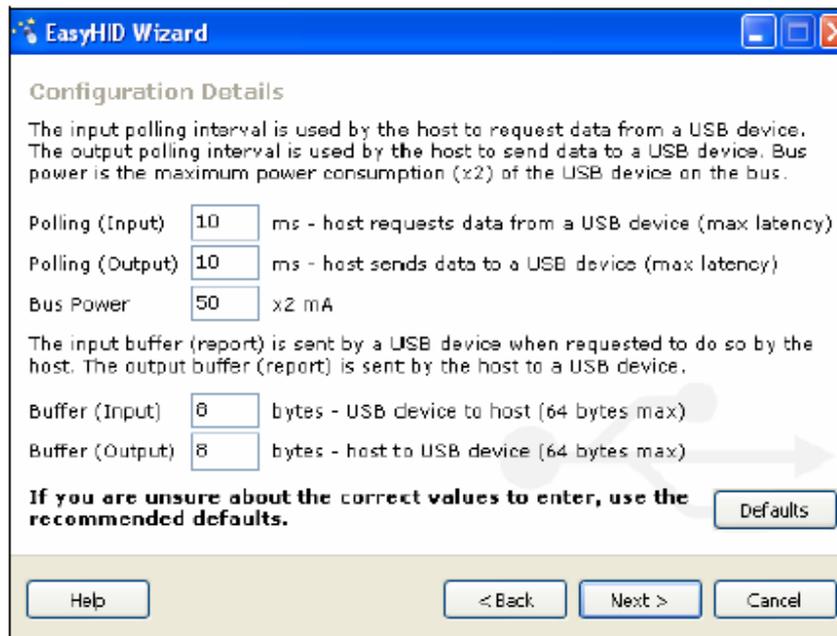


Fig. 26 Configuración de envío de datos hacia el host

La pantalla de la figura 26 muestra en donde se realiza una configuración de parámetros para realizar la comunicación con el Host (PC).

El host realiza una comunicación bidireccional con el dispositivo USB, intercambiando información continuamente durante un determinado intervalo de tiempo. Es posible manipular este intervalo dependiendo de las necesidades del sistema, por default se manejan tiempos de 10 ms. También se puede configurar la corriente que el bus USB entrega al dispositivo, es importante tener en cuenta que el bus del computador entrega una corriente máxima de 500 mA; en el caso de que el dispositivo requiera una corriente mayor se debe implementar una fuente externa que nos proporcione una mayor corriente, en nuestro caso el microcontrolador PIC 18f4550 funciona con una fuente externa y no toma los 5 voltios que el computador genera.

El sistema nos permite transmitir la información a través de buffers tanto de entrada como de salida, el número máximo de buffers que se puede implementar es de 64. Cada buffer es de 256 bits.

Uno de los inconvenientes es que el EasyHID Wizard no es totalmente gratis; por lo que si se cambia algunos de los datos que por default se presentan, el programa no funcionara, aunque este compile sin tener errores. Así que es recomendable no hacer ningún cambio en la figura 25

Se da un click y aparece la pantalla de la figura 26, en donde se ingresa el nombre del proyecto y donde va a ser guardado. Es importante guardar el proyecto con un nombre no muy largo, y la localización de este no hacerla en el escritorio del computador ni en direcciones muy largas, pues existe problemas en la compilación, ya que la extensión del archivo es muy grande.

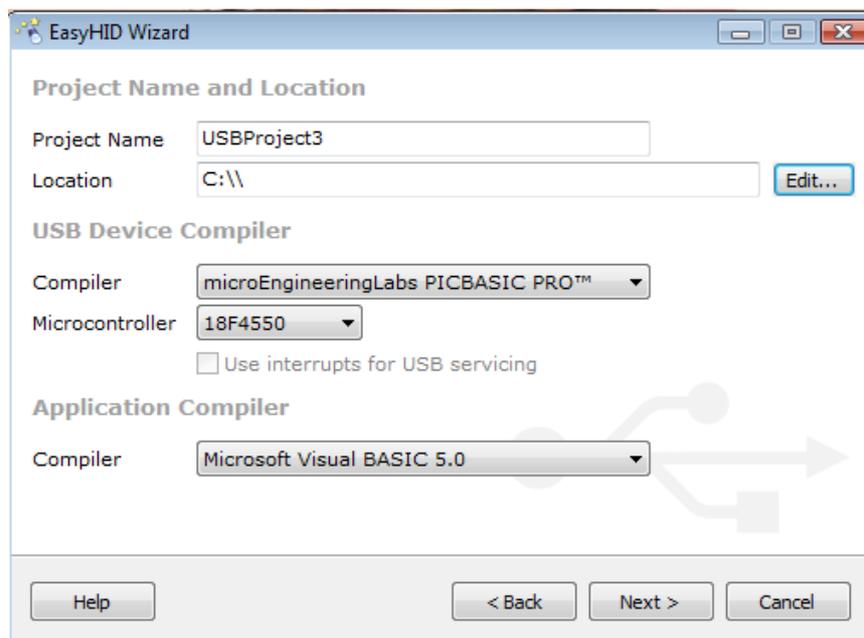


Fig. 27 Asignando nombre al proyecto

Es necesario seleccionar el microcontrolador que se va a utilizar, en este caso el PIC 18f4550. Por último es necesario seleccionar el compilador para el cual se desea que EasyHID genere las líneas de código, en este caso es PicBasic Pro y Visual Basic 5.0.

Como último paso se tiene la pantalla de la figura 28 donde el EasyHID compila todos los datos ingresados para generar los programas para el microcontrolador y para la PC. Si se realiza todo correctamente el wizard compila el proyecto y finalizará sin ningún problema.



Fig. 28 Generación del Proyecto

La PC enviará un primer byte de datos, el cual es para decidir si es manual o automático, después se enviarán dos bytes, en el caso de manual serán los bytes para definir los puertos en portb y portd, en cambio en el modo automático los dos siguientes bytes, el primero es la acción y el segundo es el tiempo de duración.

El primer byte es de la siguiente manera:

Binario	Decimal	Acción
00000000	0	Manual
11111111	255	Automático

En el modo manual quedaría de la siguiente manera:

Byte 2 para puerto D	Byte 3 para puerto B	Acción
00000100	00000001	Base Derecha
00000100	00000010	Base Izquierda
00001000	00000100	Hombro Atrás
00001000	00001000	Hombro adelante
00010000	00010000	Codo adelante
00010000	00100000	Codo Atrás
00100000	01000000	Muñeca Adelante
00100000	10000000	Muñeca Atrás
01000001	00000000	Abrir Pinza
01000010	00000000	Cerrar Pinza

En el modo automático quedaría de la siguiente forma:

Binario	Decimal	Acción
00000001	1	Base Derecha
00000010	2	Base Izquierda
00000011	3	Hombro Atrás
00000100	4	Hombro adelante
00000101	5	Codo adelante

00000110	6	Codo Atrás
00000111	7	Muñeca Adelante
00001000	8	Muñeca Atrás
00001001	9	Abrir Pinza
00001010	10	Cerrar Pinza

La secuencia quedaría de la siguiente manera:

Byte 2: Acción #...

Byte 3: Duración

.....

.....

Byte 2: Acción #...

.....

.....

Ejemplo:

11111111

00000001

00001010

.....

.....

00000100

00000001

Esta secuencia significa: Base a la derecha por 10 segundos, etc. etc. Hombro adelante 1 segundo.

Una vez se conecte la placa se espera 256 instrucciones; 128 de instrucciones y 128 de duraciones, las cuales se almacenarán en la EEPROM del chip e inmediatamente empezará la secuencia.

En la figura 29 se observa el diagrama de conexiones de la tarjeta de control al brazo robótico.

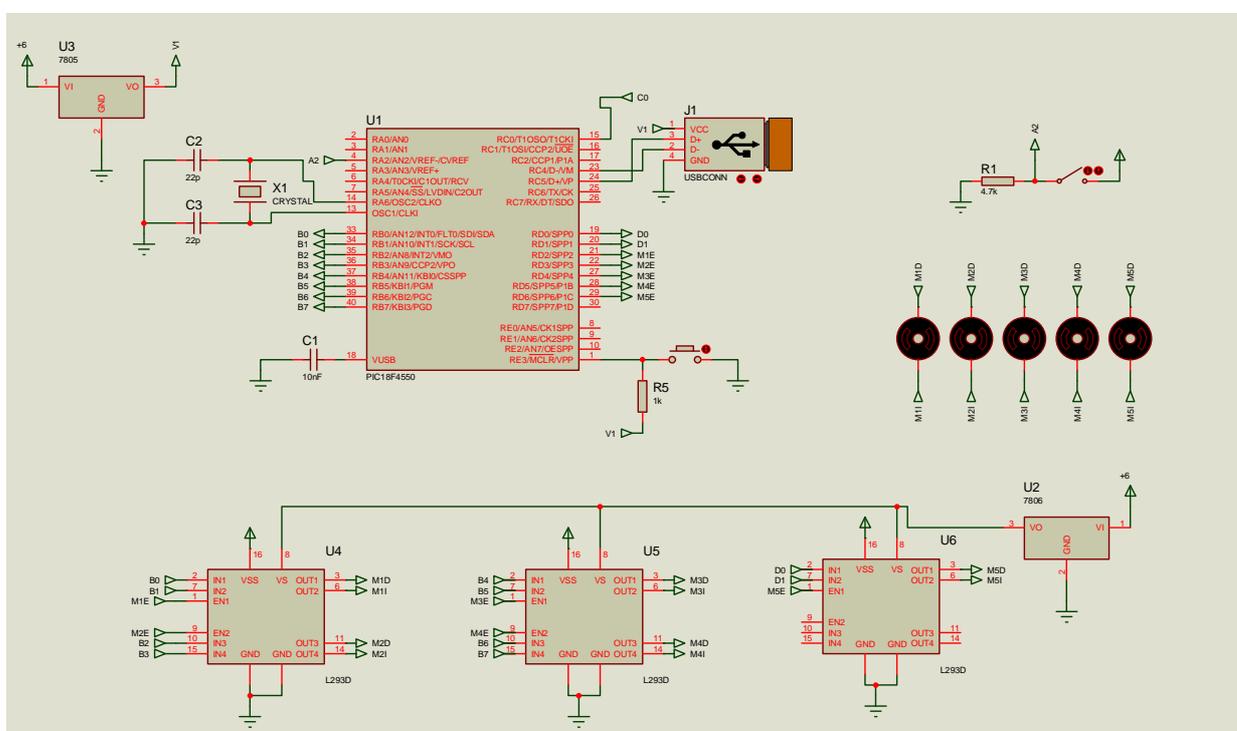


Fig. 29 Tarjeta de control para el brazo robótico

## 5. PROGRAMACION

El primer paso para la programación fue definir qué tipo de software se iba a utilizar, para este proyecto se decidió utilizar para el software de la computadora Visual Basic Studio 6.0, y para el firmware del microchip se utilizó Pic Basic Pro y para generar el código de comunicación se utilizó EasyHID, que es una herramienta que viene implementada en Microcode de Mecanique.

### 5.1 Firmware del Microcontrolador 18f4550

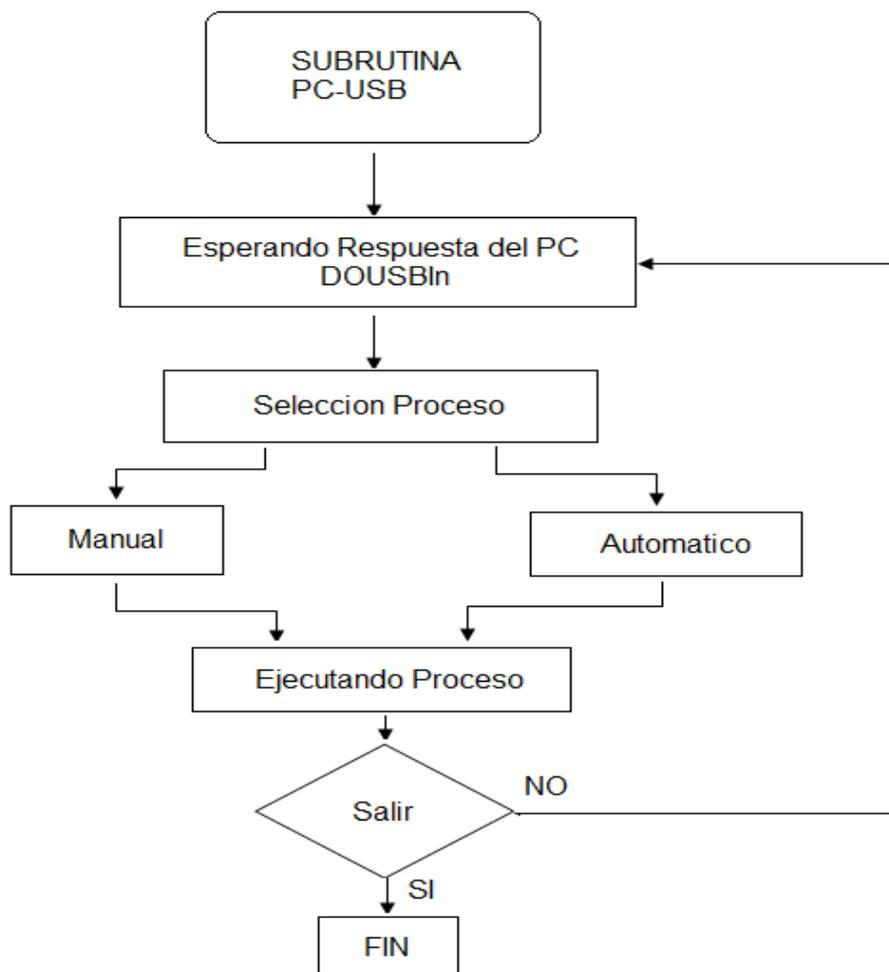


Fig. 30 Diagrama de Flujo para el PIC

A continuación se muestra las líneas de código que el EasyHID genera, y se detalla cómo fue realizado todo el programa. A la par con el diagrama de flujo de la figura 30 se explicara cómo se realiza la comunicación USB.

```
DEFINE OSC 48
```

```
DEFINE LOADER_USED 1
```

```
'Definición de la variables del programa
```

```
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 ProgramStar
```

```
! *****
```

```
' * 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 data

**RETURN**

### **FIN DEL PROGRAMA**

El programa empieza a ejecutarse inicializando el USB a través del comando USBINIT, y espera a que el computador le envíe algún dato. El Buffer USB tanto de entrada como de salida tiene 8 Buffers de 256 bits cada uno, que van desde el USBBuffer [0...7] respectivamente.

El programa constantemente está verificando si ha llegado algún dato proveniente desde el PC, esto lo hace saltando a la subrutina DoUSBIn.

Cuando el PC envía un dato el programa hace una selección comparando la información que se envía en los Buffers; por ejemplo si el USBBuffer [8] = 1, entonces es manual, de esta manera se puede ejecutar todas las acciones que son necesarias para este proceso como: automático, manual.

El computador le envía al microcontrolador la información de los movimientos que debe realizar el brazo robótico, así como los comandos para ejecutar las mismas acciones en forma automática.

Para enviar datos desde el PIC al computador se coloca la información en el Buffer USB por ejemplo USBBuffer[3]=3, y se ejecuta el comando DoUSBOut, en este momento el programa se dirige a esta subrutina y envía el dato al computador.

El computador da por finalizado el proceso, y el programa regresa a la posición inicial.

Una vez hecha la programación es importante garantizar que el programa está bien, por lo que es necesario conectar el microcontrolador al computador. Conectamos el cable USB al computador y se observa lo siguiente:



Fig. 31 Dispositivo USB encontrado

Al momento de conectar el controlador al PC aparece la etiqueta de Nuevo Hardware Encontrado como se muestra en la figura 31. El computador lo configura y le permite realizar una configuración bidireccional con el mismo. Es importante destacar que el computador debe constar con todos los driver necesarios para conexión USB, los cuales ya se mencionaron anteriormente, para que el dispositivo pueda ser reconocido.



Fig. 32 Dispositivo listo para usarse

Después de que el computador ha configurado el dispositivo, aparece la etiqueta que el nuevo hardware está listo para ser utilizado, tal como se muestra en la figura 32.

## 5.2 Software de la PC

La segunda parte de este proyecto es desarrollar un software que controle el brazo robótico, este software es una interfaz grafica desarrollada en Visual Basic 6.0, la cual controla los movimientos y tiempos.

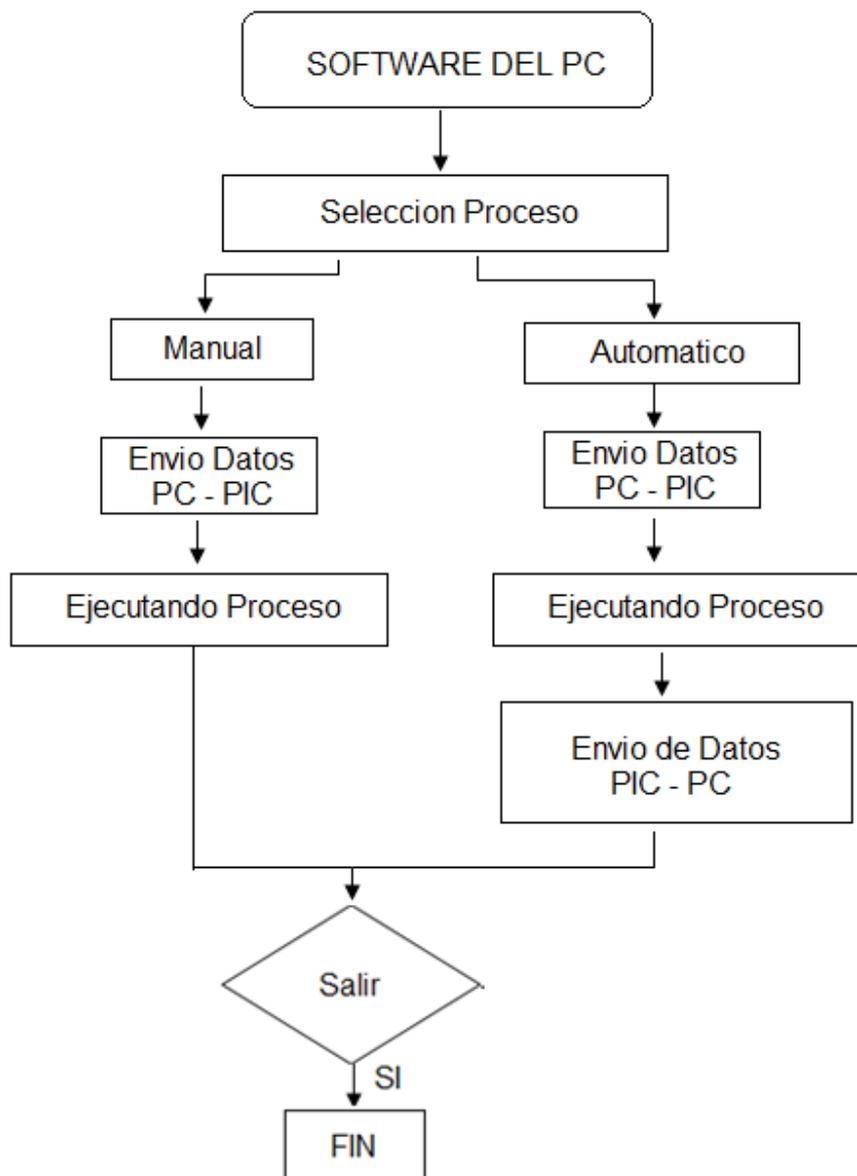


Fig. 33 Diagrama de flujo para la PC

En la figura 33 se puede observar el diagrama de flujo que se utilizó para desarrollar el programa para la PC.

Para desarrollar la aplicación en Visual Basic se debe abrir el código generado por EasyHID. En este archivo existen dos carpetas, la primera llamada formularios, que contiene el Main Form o formulario principal en donde se desarrollará el

programa, la segunda carpeta llamada Módulos contiene unas líneas de código que generan un dll. Un DLL es un acrónimo de “Dynamic Linking Library”, término con el que se refiere a los archivos con código ejecutable que se cargan bajo demanda del programa por parte del sistema operativo. Este modulo contiene el dll para el controlador HID, el cual se enlazara con el Firmware del microcontrolador para poder realizar la comunicación USB. No se necesita hacer ningún cambio en este modulo.

En el formulario principal se encuentran otras líneas de código, que interactúan con el código del modulo anterior para poder realizar la comunicación USB. Estas líneas de código son las encargadas de realizar el proceso de Enumeración USB, es decir, pedirle al microcontrolador alguna información como su ID, a que vendedor pertenece, además detecta que se ha conectado un dispositivo USB con características de HID, como es el caso del PIC 18f4550, luego lo configura y lo deja listo para que pueda intercambiar información con el computador.

Al igual que en el microcontrolador, en este programa existe un número determinado de Buffers, tanto de entrada como de salida, en este caso se tiene 9 Buffers que van desde cero hasta el ocho respectivamente. El BufferIn (0) y el BufferOut (0), no se los debe utilizar, ya que estos Buffers contienen el reporte ID, por lo tanto los buffers que se pueden utilizar para transmitir datos son desde el uno al ocho [1...8]. En Visual Basic los buffers son mayores en uno a los buffers del microcontrolador, por ejemplo el USBBuffer [1] del PIC corresponde al BufferIn (2) de Visual Basic.

La programación en Visual Basic es orientada a objetos, esto quiere decir que para desarrollar una interfaz no es necesario implementar muchas líneas de código; por ejemplo se tiene la opción de insertar un botón, en el cual se puede escribir líneas de código para que se ejecute una determinada acción en el momento en el que el botón sea pulsado.

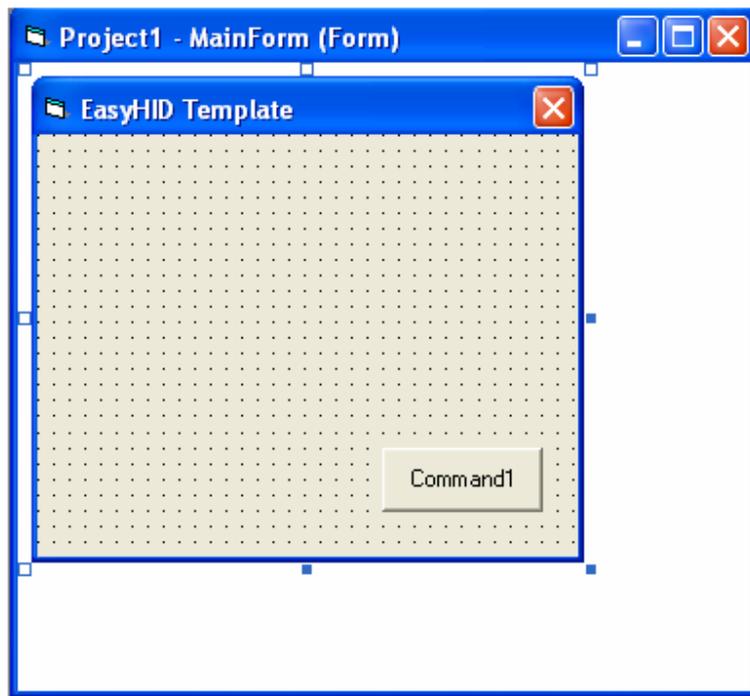


Fig. 34 Main Form del programa

En la figura 34 se observa el Main Form, en donde se ha insertado un botón, en el momento en que se hace click sobre él se ejecuta una rutina ya establecida.

Para poder transmitir un dato al microcontrolador es necesario asignarle un valor a uno de los buffers, por ejemplo: `BufferOut (8) = 1`, ahora para transmitir el dato se debe colocar el siguiente código:

```
hidWriteEx VendorID, ProductID, BufferOut(0)
```

En cambio cuando el microcontrolador envíe un dato al computador, es necesario verificar los buffers de entrada. Por ejemplo si BufferOut (8) =1, se debería ejecutar una acción que ha sido programada por el usuario.

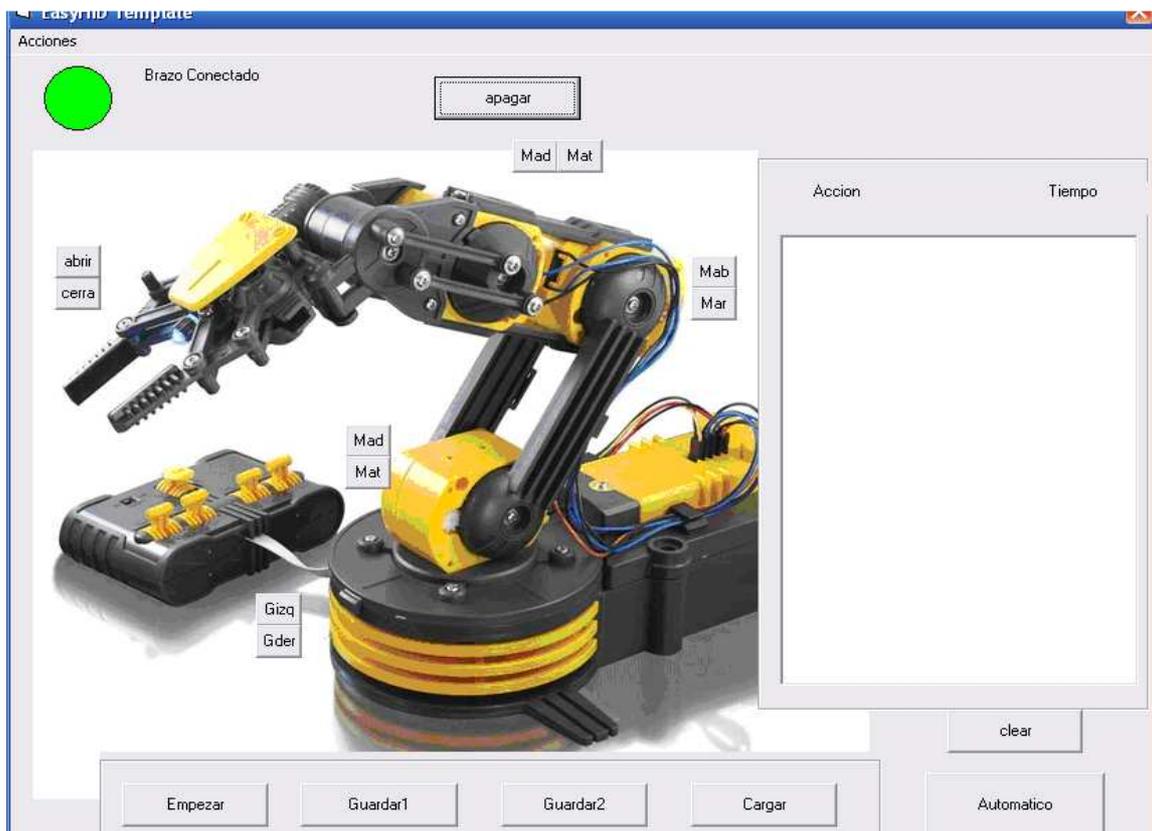


Fig. 35 Interfaz gráfica del brazo robótico

En la figura 35 se observa la interfaz desarrollada en Visual Basic para controlar el brazo robótico. Como se puede observar en la figura 35 la interfaz muestra cada uno de los botones de selección que ejecutarán una acción diferente como mover el hombro, cerrar la pinza, etc.

En la figura 36 se puede observar un ejemplo de las tareas que se pueden ejecutar con el brazo robótico.

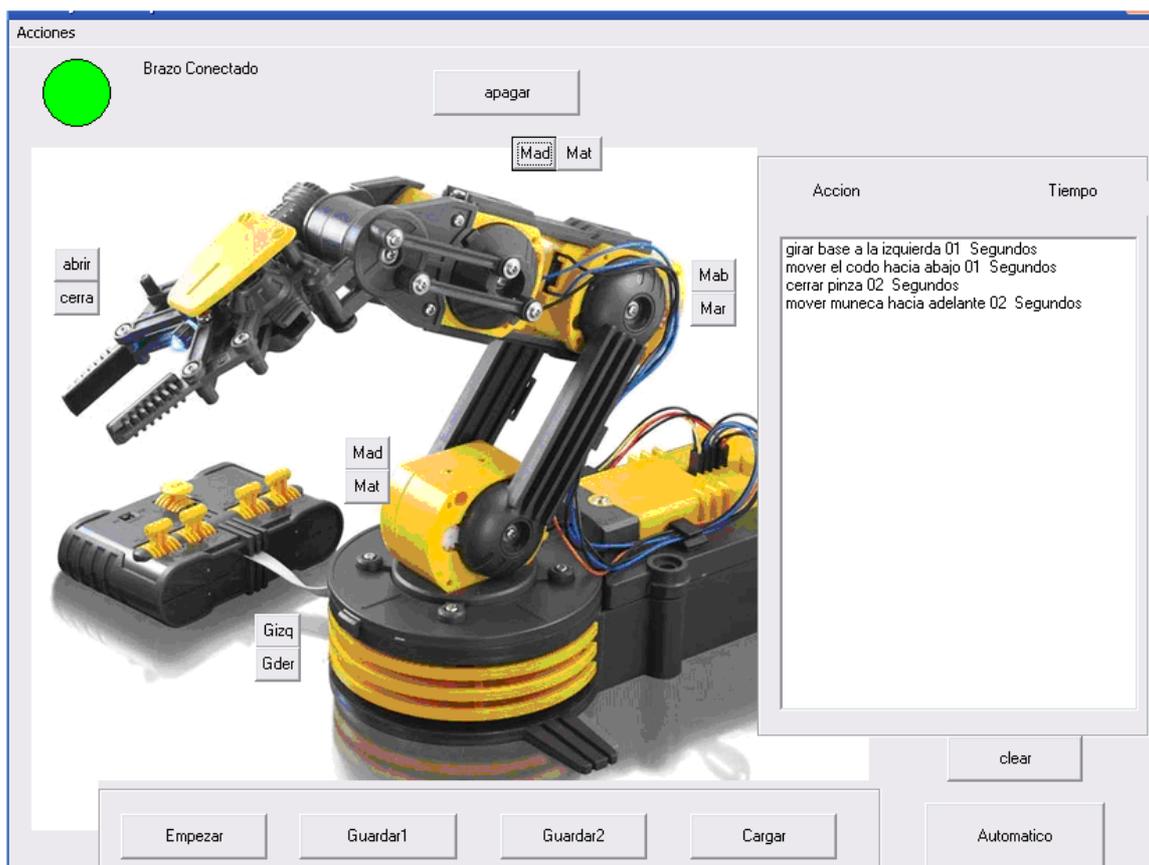


Fig. 36 Generando tareas para el brazo robótico

## 6. FUNCIONAMIENTO Y PRUEBAS

La primera prueba que se realizó, fue la verificación del funcionamiento del robot después de haber sido ensamblado, es decir que el control remoto hiciera los movimientos correctos, y no exista ningún tipo de daño o error en las conexiones eléctricas y mecánicas.

La siguiente prueba fue la ejecución del movimiento de los motores, primero se procedió a probar cómo funcionaban los puentes, primero se hizo una simulación en el programa PROTEUS, que es un software de simulación electrónica, aquí se procedió a enviar señales a través de entradas digitales, se observó que funcionó, por lo que se procedió a hacer la prueba conectando un puente L293D a un motor con las protecciones debidas, y luego se mandó señales de 0 y 5 V a través de una fuente, en este caso se usaron 2 cargadores, uno para las señales de control, y el otro para la alimentación de los motores. La prueba salió muy bien de acuerdo a como se enviaba las señales se designaba el sentido de giro del motor.

La siguiente prueba fue enviar las mismas señales al puente, pero ya no desde una fuente, sino desde el pic, por lo que primero se hizo una simulación en PROTEUS, para esto primero se escribió el programa en microcode y luego fue cargado en el pic, en esta simulación se enviaba señales para activar 5 motores en tiempos indistintos, la simulación funcionó muy bien por lo que se procedió a montar el circuito en un proto, el programa del pic fue descargado a través de un programador de pics y del programa winpic, la alimentación se hizo de la misma forma, con dos fuentes, una para señales de control y alimentación del pic, y la

otra para la alimentación de los motores. La prueba dio resultados exitosos, por lo que la siguiente prueba fue la comunicación entre el pic y la pc.

La comunicación USB entre el PIC y la PC funcionó correctamente, se insertó un grupo de órdenes, se enviaron y el pic las ejecutó tal y como fueron enviados, por lo que la primera parte del proceso estaba funcionando perfectamente.

La segunda parte era tener un control interactivo del brazo robótico, es decir mientras se mantenía pulsado el click del mouse el brazo se movía, una vez que se quita el click, el brazo se detenía, y en el programa hecho en Visual Basic, se grababa esa posición y el tiempo, y después se ejecutarían todas las funciones que se guardaron. Se escribió el código necesario para este proceso y funcionó perfectamente, por lo que hasta este momento se tenía ya un control interactivo y un control automático y autónomo.

La tercera prueba fue realizar las rutinas para cada sensor, y que sean activadas solo con las señales que estos daban al microcontrolador.

Se utilizó un switch con señal enclavada, para simular las señales del sensor, es decir un cero es la señal del sensor para el objeto 1, y 1 es la señal del sensor para el objeto 2.

En esta parte, la prueba tuvo pequeños problemas, ya que el tiempo de visual basic tiene cierto desfase con el tiempo real, por lo cual tiene acumulación de errores y la secuencia que se envió en un comienzo no se repite con exactitud, sin embargo se asemeja mucho a las órdenes iniciales.

Para solucionar el problema del retraso de tiempos, en los programas tanto de visual basic como de microcode, se programo los temporizadores a centésimas de segundo y a milésimas de segundo para la pinza, de esta forma las repetición de rutinas es más exacta.

En la figura 37 y 38 se puede observar al brazo robótico tomando a los diferentes objetos.



Fig. 37 Objeto 1

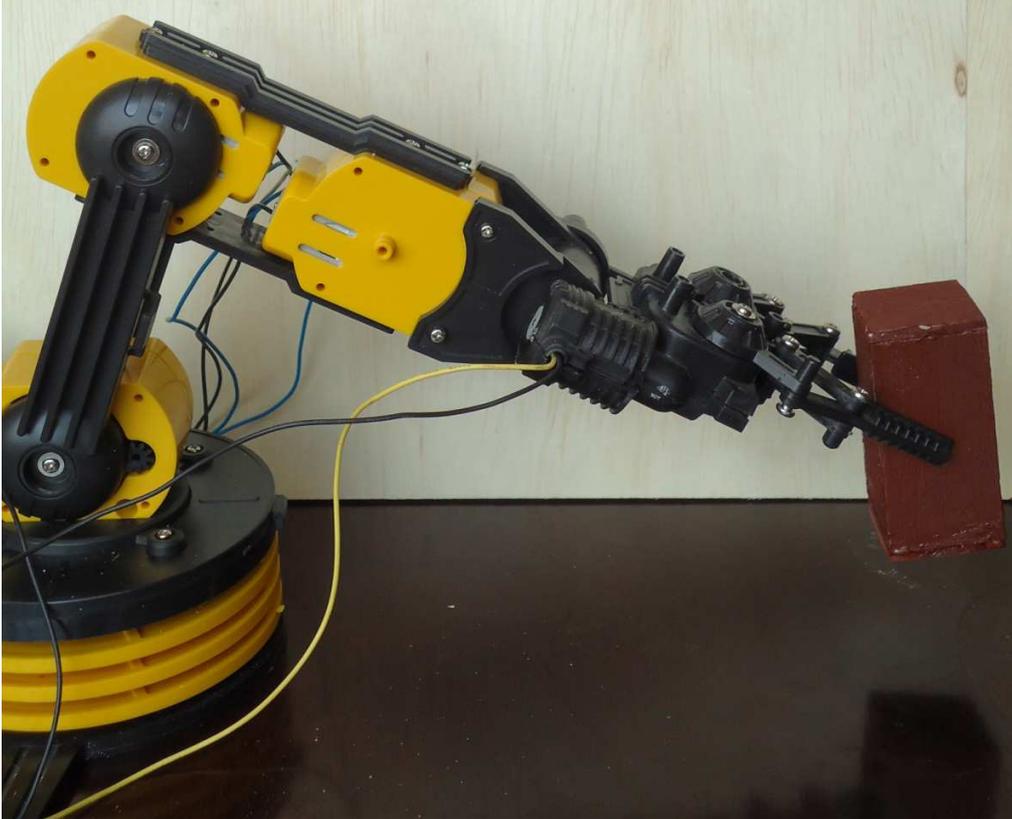


Fig. 38 Objeto 2

## 7. CONCLUSIONES Y RECOMENDACIONES

### 7.1 Conclusiones

A continuación se explican los objetivos principales y como fueron obtenidos satisfactoriamente.

- *Comunicación*

En primer lugar la prueba de comunicación no dio los resultados deseados en los primeros intentos, ya que se conectaba el cable usb, este era detectado por la pc, se ejecutaba las órdenes en Visual Basic, sin embargo no pasaba nada, ni un solo movimiento. Después de revisar los programas, se encontró que la falla era la conversión de decimal a binario, lo cual no transmitía correctamente el dato hacia el pic, por lo cual éste no detectaba nada y no hacia ningún movimiento.

Cuando se detecto el error , se hizo los cambios necesarios, y la comunicación USB entre el PIC y la PC funciono exitosamente.

- *Movimiento Interactivo*

El objetivo del movimiento interactivo era tener un control sobre el brazo robótico, es decir, cuando se hacia click sobre el botón, una de las partes del brazo se movía, y cuando se daba otro click sobre el mismo botón, el movimiento se detenía.

Además este sistema permite guardar todas las órdenes para luego ser ejecutadas cuando el usuario lo desee, al mismo tiempo que puede modificarlas o retirarlas.

- *Automatización*

El objetivo de la automatización era generar un proceso que esté trabajando continuamente, el cual dependía de dos sensores, en este caso se utilizó un swich, esto funcionó correctamente ya que al presionar en automatico el proceso empieza a trabajar, y cuando se cambia el swich de 0 a 1 o viceversa ella rutina de trabajo cambia, y empieza el otro proceso.

Este proceso se detiene manualmente, ya que el botón de parado se encuentra en la tarjeta de control.

- *Pruebas*

El objetivo de las pruebas era comprobar que el proceso automático funciona correctamente con cada uno de los objetos. La primera vez que se probó se observó que los tiempos generaban retrasos, por lo que a ejecutarse la siguiente vez el objeto era tomado y colocado en diferente lugar, para corregir este error se tuvo que configurar el tiempo en centésimas y milésimas de segundo, con lo cual el error de retardo se redujo al máximo.

Ahora el brazo funciona correctamente dependiendo del objeto que encuentre en el tablero.

Otro de los objetivos era crear una placa electrónica versátil, que permitiera trabajar con otro tipo de máquinas, lo cual es posible ya que esta placa permite trabajar hasta con cinco motores, de una forma local o de forma remota, además de la utilización de más sensores ya que uno de los puertos del PIC queda libre para esta tarea, esta placa podría ser utilizada como una tarjeta de adquisición de datos.

El software fue creado en Visual Basic, por lo que puede crearse un ejecutable que puede ser instalado en cualquier computadora, aunque esta no tenga el programa.

El sistema completo entre placa electrónica y software del PC puede ser utilizado en cualquier sistema que requiera control de máquinas y monitoreo del mismo.

## 7.2 Recomendaciones

Se recomienda tener toda la información sobre el funcionamiento del protocolo USB, y de cómo implementarlo en el PIC 18f4550.

También se recomienda observar las corrientes de los motores, ya que los puentes utilizados en la placa electrónica soportan hasta 600 mA.

Se recomienda no utilizar cargas que sobrepasen los 200 gr, ya que este es el máximo que soporta el brazo robótico.

## BIBLIOGRAFIA

ARGONNE National Laboratory. 31 de Mayo de 2010

<[http://www.anl.gov/Science\\_and\\_Technology/History/Anniversary\\_Frontiers/alchemy.html#master](http://www.anl.gov/Science_and_Technology/History/Anniversary_Frontiers/alchemy.html#master)>.

Axelsson, Jan. USB Complete. 2 Edición. Lakeview Research, 2001.

Baturone, Anibal. Robotica: Manipuladores y Robots Moviles. Marcombo Ediciones Tecnicas, 2007.

Kurfess, Thomas. Robotics and Automation Handbook. CRC Press, 2005.

Microsoft Corporation. «Microsoft Visual Basic 6.0 Referencia del Lenguaje.» Vol. 1. MCGrawHill, 1999.

Peláez, Antonio Lopez. Impactos de la Robotica y la Automazacion Avanzada en el Trabajo. Estudio Delphi. Madrid: Editorial Sistema, 2000.

Peláez, Antonio Lopez y Miguel Krux. «The IPTS Report.» 2010. Junio de 2010 <[http://www.sappiens.com/CASTELLANO/articulos.nsf/Gestión\\_Tecnológica/Impactos\\_sociales\\_de\\_la\\_robótica\\_y\\_de\\_la\\_automatización\\_avanzada\\_en\\_el\\_año\\_2010/7A2C33366766724441256B2500500B8F!opendocument](http://www.sappiens.com/CASTELLANO/articulos.nsf/Gestión_Tecnológica/Impactos_sociales_de_la_robótica_y_de_la_automatización_avanzada_en_el_año_2010/7A2C33366766724441256B2500500B8F!opendocument)>.

Simpson, Janice, Otto Friedrich y Christopher Redman. «Time.» 8 de Diciembre de 1980. Mayo de 2010 <<http://www.time.com/time/magazine/article/0,9171,922173-4,00.html>>.

Smid, Peter. CNC Programming Handbook. A Comprehensive Guide to Practical CNC Programming. 3 Edición. 2008.

Datasheet PIC 18f4550

Datasheet L293B

Datasheet L293D

## **ANEXOS**