



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

**DISEÑO Y ACTUALIZACIÓN TECNOLÓGICA DE UNA SONDA ROBOT
MOVIL**

T E S I S
COMO REQUISITO PARA OBTENER EL TÍTULO DE
INGENIERO ELÉCTRICO ELECTRÓNICO
(SISTEMAS DIGITALES)

P R E S E N T A
VICTOR MANUEL LOMAS BARRIÉ

DIRECTOR: ROMÁN OSORIO COMPARÁN

CIUDAD UNIVERSITARIA

MÉXICO, D.F.

2006





Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

Dedico este trabajo al pueblo de México.

Ya que gracias a su esfuerzo han hecho
de la UNAM, una universidad de clase mundial
Viva la educación científica, pública y gratuita

Todo mi agradecimiento a mimadre.
Que es el motor de mi inspiración.

Índice

INTRODUCCIÓN.....	1
PLANTEAMIENTO DEL PROBLEMA.....	1
1.1 Antecedentes.....	1
1.2 Objetivos.....	2
1.2.1 Alcances.....	2
1.3 Justificación.....	3
1.4 Estado del arte de la robótica.....	3
1.5 La robótica en México.....	11
1.6 Estado inicial de la Sonda Robot Móvil.....	13
COMPONENTES DE LA ACTUALIZACIÓN.....	19
2.1 Requerimientos computacionales de la Sonda Robot Móvil.....	19
2.2 Computadora principal de la SRM.....	21
2.2.1 Arquitectura x86.....	22
2.2.2 Tarjeta Madre EPIA TC 10000E.....	28
2.3 Expansión del puerto de datos.....	30
2.3.1 El USB.....	32
El hardware del USB.....	35
Conectores.....	35
Aspectos eléctricos.....	35
Identificación de la velocidad.....	36
Potencia eléctrica del USB.....	37
Protocolos del USB.....	38
Estructura de un paquete común.....	38
Tipos de paquetes.....	40
Endpoints.....	41
Pipes.....	42
Tipos de Endpoints.....	42
Descriptores del USB.....	45
2.3.2 Dispositivos USB enfocados al desarrollo.....	50
2.3.3 El PIC16C745/65.....	53
2.4 Comunicación entre la SRM y la estación de control.....	55
2.4.1 Estándar WiFi.....	57
2.5 Control Remoto Virtual.....	59
ACTUALIZACIÓN DE LA SONDA ROBOT MÓVIL.....	62
3.1 Esquema de la nueva configuración de la Sonda Robot Móvil.....	62
3.2 Computadora de Control (CC) de la Sonda Robot Móvil.....	64
3.2.1 Potencia eléctrica de la de la Sonda Robot Móvil.....	68
3.2.2 Instalación del Disco Duro en la Sonda Robot Móvil.....	71
3.2.3 Instalación del Módulo Inalámbrico en la Sonda Robot Móvil.....	74
3.2.4 Instalación del sistema operativo y controladores en la Computadora de Control.....	75
3.3 Tarjeta de Expansión de Puertos (TEP).....	78
3.3.1 Hardware de la Tarjeta de Expansión de Puertos.....	78
3.3.2 Software de la Tarjeta de Expansión de Puertos.....	83

3.4 Software de control de la Sonda Robot Móvil. ControlSRM y TightVNC.....	88
3.4.1 ControlSRM.....	88
3.4.1 TightVNC.....	90
3.5 Estación de Control.....	92
CONCLUSIONES.....	94
BIBLIOGRAFÍA.....	97
Apéndice A.....	99
Apéndice B.....	104
Apéndice C.....	108

INTRODUCCIÓN

El propósito de este trabajo es la mejora tecnológica de la Sonda Robot Móvil (SRM) desarrollada en el Departamento de Ingeniería de Sistemas Computacionales y Automatización de la Sección de Electrónica y Automatización, DISCA-SEA; del Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas (IIMAS), Universidad Nacional Autónoma de México (UNAM).

El proyecto que aquí se presenta consistió en la integración de un equipo de cómputo compatible con la arquitectura x86 a la SRM. Para llevarla a efecto, fue necesario seleccionar y adaptar la tecnología apropiada en hardware y software; además, se utilizó una interfaz USB para el control de los motores y sensores de la SRM. Como resultado final se esperaba una mejora en la operación, así como en la flexibilidad para incorporar nuevos componentes en el futuro, en la sonda robot.

Los motivos por los cuales se eligieron las soluciones tecnológicas entre un conjunto de diversas propuestas actuales, viables y disponibles consistieron principalmente en las siguientes condiciones:

- 1) La técnica adoptada debe ser expansible, es decir, que en cualquier actualización técnica de la sonda o al agregarle otro componente, este debería de mostrar cierta versatilidad para que la tarea resultara sencilla y evitar así, re-estructurar nuevamente el robot;
- 2) Debe existir compatibilidad entre los componentes de hardware y software con los que se cuenta así como con el resto de las actuales tecnologías;
- 3) Lograr el desarrollo de un robot móvil de características similares a los comerciales que contengan un procesador Pentium, a un precio muy por debajo del mercado.

En el primer capítulo se presentan con mayor detalle los motivos, así como los objetivos y justificación del desarrollo de este proyecto, además se describe el estado del arte de la robótica en el Mundo y en particular en nuestro país, México. Al final de ese capítulo hay una descripción del estado inicial de la SRM, es decir, antes de se realizara la actualización aquí reportada.

El planteamiento del problema acompañado de la formulación de distintas propuestas para resolver dichos problemas se presenta en el capítulo dos. Ahí se explican en detalle los pasos necesarios para

poder actualizar la sonda; como son la incorporación de una computadora de control, compatible con la arquitectura x86. Además se describe el funcionamiento del bus USB como solución para resolver el problema de la limitación del número de puertos y de compatibilidad que tiene el diseño original de la sonda, así como las ventajas que ofrece esta tecnología para la flexibilidad en su conexión e integración.

El último capítulo describe el proceso de actualización. Que involucra el diseño e implementación de componentes eléctricas, electrónicas, mecánicas y de computación. Este último punto es el de mayor relevancia, pues la actualización no solo consistió en la implementación de hardware nuevo, sino que hubo la necesidad de desarrollar software nuevo tanto para la computadora de control de la sonda como para el control de la etapa de expansión de puertos por medio del puerto USB.

Con el desarrollo de este proyecto se abre la posibilidad de incorporar el desarrollo de software de inteligencia artificial a la Sonda. Estas técnicas generalmente no se incorporan en máquinas digitales de baja escala por la dificultad de su implementación, sin embargo con esta actualización la sonda estará en posibilidades de ejecutar estos algoritmos inteligentes, al cubrir la demanda de mayor velocidad de procesamiento de cómputo.

PLANTEAMIENTO DEL PROBLEMA

1.1 Antecedentes

El DISCA-SEA cuenta con distintos proyectos de innovación tecnológica, que generalmente se orientan a la solución de un problema real ya sea de una empresa, del gobierno, de la sociedad, de instituciones educativas e incluso de la UNAM. En este caso, el problema tiene que ver con salvaguardar la integridad física del personal que labora en industrias con alto riesgo de trabajo.

EL proyecto de la SRM (Sonda Robot Móvil) plantea una alternativa de tecnología robótica para evitar accidentes laborales en zonas de trabajo de alto riesgo, bajo el precepto de que si alguien tiene que realizar cierta actividad que ponga en riesgo su salud, es mejor que la realice una máquina, como la sonda robot que puede ser teleoperada y así realizar dicha actividad. La SRM específicamente fue diseñada para utilizarse en plantas de generación de energía eléctrica a partir de energía nuclear, pero de igual forma el robot puede ser adaptado para trabajar en cualquier espacio de trabajo y ocuparse en otro tipo de aplicaciones industriales, científicas o domésticas.

En el DISCA-SEA se ha desarrollado una sonda robot, diseñada y construida para ser utilizada en plantas de energía nuclear y de ésta forma aislar el contacto del cuerpo humano con la radiación. En un primer diseño, esta sonda robot se manipulaba remotamente de forma alámbrica desde una zona segura y libre de radiación. El sistema de control, se basaba en un microcontrolador 8051 de la marca Intel. Este sistema tenía la función de: 1) recibir y enviar datos, desde y hacia la computadora de control a través de una interfaz RS-232, 2) controlar los motricidad de la sonda así como la captura de datos provenientes de sensores.

Con esto se cumplía el objetivo de que a través de una cámara de video montada en el robot y monitoreada desde la zona de control esta realice la supervisión de tuberías, válvulas, mangueras, paredes y tanques; así como la adquisición de datos provenientes de sensores de temperatura, radiación, humedad y presión instalados en la superficie del robot. Además, la sonda robot cuenta

con un manipulador, un brazo mecánico de seis grados de libertad, para abrir o cerrar válvulas, tomar o mover objetos o simplemente para recolectar muestras de algún material del cual se requiera realizar algún estudio. Como parte de ese diseño, el robot cuenta con sensores de ultrasonido que le permiten evitar colisiones con objetos o paredes. Y a través de un compás y un odómetro es posible calcular la posición, orientación y distancia recorrida de la sonda robot.

1.2 Objetivos

El objetivo principal de este trabajo es la actualización tecnológica de la Sonda Robot Móvil (SRM); que consiste de dos partes. La primera es sustituir el sistema de control basado en el 8051, por una plataforma de cómputo conocida como x86, la cual le brindará mayor velocidad y capacidad de procesamiento de cómputo. Aquí, además de elegir y posteriormente instalar los componentes de hardware, es necesario rediseñar el software (“Softelenav”¹) desarrollado para la operación (tracción, dirección, sensores) de la Sonda original en forma remota, pasando así de una comunicación por el puerto serie a una comunicación por red inalámbrica (WiFi). La segunda parte consiste en diseñar y desarrollar una interfaz que permita el control del sistema motriz de la sonda, así como la captura de datos de los diferentes sensores. Esta interfaz debe presentar características de compatibilidad y versatilidad por lo que se propone como medio de enlace, utilizar el puerto USB (Universal Serial Bus).

1.2.1 Alcances

Los alcances de este trabajo son los siguientes:

- a) Seleccionar, validar y poner en marcha el nuevo sistema de cómputo que cumpla con las características que requiere ésta actualización.

¹ Arteaga, Alma. *Tesis “Desarrollo del software de un sistema para la operación de un robot móvil en un ambiente de realidad virtual 3D”*, México, 2002, UNAM

- b) Diseñar e implementar la interfaz USB que recibe ordenes del nuevo sistema de cómputo para comandar sensores y motores en la sonda. Por lo que es necesario acoplar estas señales con dicho bus, el USB.
- c) Analizar los requerimientos de potencia eléctrica tanto de los viejos como de los nuevos sistemas de la sonda y en función de ello proponer el diseño de un sistema que suministre la energía que ésta requiere.
- d) Desarrollar el software para el control de la sonda robot.

1.3 Justificación

Sin duda el éxito de este proyecto, se verá reflejado al abrir la posibilidad de utilizarse en aplicaciones que requieran soluciones a problemas que requieran de una sonda robot, pues al contar ésta con un sistema de procesamiento computacional de amplia capacidad como lo es la arquitectura x86, la programación será mucho más fácil, adaptable, versátil y compatible para la incorporación de otros dispositivos. Este proyecto se puede adaptar para distintos propósitos como son: la exploración de minas, la exploración acuática, la exploración de cavernas y ríos subterráneos y para manipular materiales o sustancias peligrosas, entre otros.

1.4 Estado del arte de la robótica

Con el avance de tres ramas de la ingeniería: electrónica, informática y mecánica, los robots cada vez son y serán más eficientes y baratos. Circunstancia que en algún momento podrá hacer posible, que se conviertan en las fuerzas productivas de un país, y así mismo será posible abaratar los costos de los productos y mejorar su calidad.

Los primeros pasos se dieron apenas a mediados del siglo pasado, cuando George Devol patentó el primer robot en la historia, que curiosamente se desarrolló a la par de la primera computadora

electrónica ENIAC. En 1954, Jon Engeberger formó la empresa Unimation, la primera en desarrollar robots comerciales para la industria. Es por esto que Engeberger es considerado como el pionero de la robótica.

En 1962, General Motors incluyó un robot en su línea de ensamblaje de autos. Durante 1964, el Tecnológico de Massachusetts (MIT, *Massachusetts Institute of Technology*) implementó el laboratorio sobre inteligencia artificial, así como el Instituto de Investigación de Stanford (SRI, *Stanford Research Institute*) acompañado de la Universidad de Edimburgo del Reino Unido. En 1970, Victor Scheinman ingeniero mecánico de los laboratorios de Stanford y fundador de la empresa Vicarm, desarrolló un brazo mecánico llamado “Stanford Arm” (ver figura 1-1). El cual ayudó en el ensamble de la bomba de agua del vehículo Ford modelo T, complementado con sensores de tacto y ópticos².

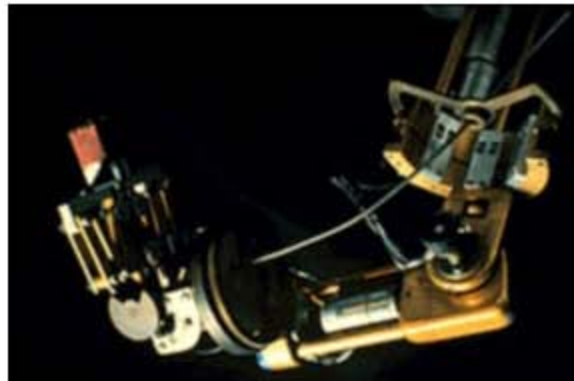


Figura 1-1. Standford Arm

En 1974, la NASA construye el Mars Rover (Explorador de Marte) con ayuda del Jet Propulsion Laboratory (JPL). En 1977, Unimation adquiere Vicarm y un año después desarrollan PUMA (*Programmable Universal Manipulator for Assembly*). A partir de éste año y hasta la fecha existe una carrera por desarrollar el mejor robot, entre las características deseadas se incluye el que sea económico, multifuncional, reprogramable, inteligente, rápido y preciso.

Una forma de clasificar a los robots es, de acuerdo a su capacidad de movimiento, en móviles o fijos. Los robots fijos incluyen todos los brazos manipuladores con fines industriales como los

² <http://www.thocp.net/reference/robotics/robotics.html>

cortadores, soldadores, fresadoras automáticas y en general máquinas de control numérico (CNC). Dentro de los móviles están los robots exploradores, las grúas, y otros. Por el tipo de movimiento, los robots se pueden clasificar en: cilíndricos, esféricos, volumétricos y *Selective Compliant Assembly Robot Arm* (SCARA).

A continuación se mencionan algunos de los más importantes y conocidos robots, cuyas características, han marcado las pautas para la construcción de robots en las últimas décadas.

Asimo³ (ver figura 1-2) es un robot desarrollado por la empresa japonesa Honda, se trata de un humanoide que fue creado desde 1986 y ha tenido distintas actualizaciones, de tal manera que ahora es un robot que entre sus múltiples características puede caminar, tarea muy complicada para sus programadores. Este robot mide 1.30 m, pesa 52 kg y alcanza una velocidad de 3 km/h cuando camina y cuenta con 26 grados de libertad



Figura 1-2. Asimo el robot de Honda.

Otro robot mascota que se ha desarrollado comercialmente es Aibo⁴, amigo en japonés (ver figura 1-3). Este robot tiene forma de un perro y fue desarrollado por la empresa líder en electrónica, Sony. Aibo tiene un procesador central de 64 bits Risc y una velocidad de 576 Mhz, su peso es de 1.65 kg

3 <http://asimo.honda.com/index.asp?bhcp=1>

4 <http://www.jp.aibo.com/>

y mide 32 cm de largo. La nueva versión de Aibo, distribuida en enero del 2005, puede grabar y reproducir música con formato mp3, wmv, wav así como grabar video. El verdadero reto de Aibo, es su capacidad para expresar emociones, ya que cuenta con un programa de inteligencia artificial, que simula los compartimientos de una mascota de verdad; cuenta con sensores para evitar colisiones con obstáculos, así como para poder sentir la textura de las superficies que toca. Aibo también es utilizado por los programadores de inteligencia artificial, en los partidos de fútbol que organiza la Federación de Robot Soccer⁵.



Figura 1-3. Aibo, amigo en japones.

Otra creación es el robot de la empresa iRobot⁶ el cual se conoce por el mismo nombre. Este robot está basado en un procesador K6-2 y cuenta con 64 MB de memoria RAM. Un navegador de páginas Web es utilizado para operarlo remotamente. Otro robot basado en una computadora personal en el B21 de la empresa RWI. Este robot utiliza un procesador Pentium y utiliza un enlace WaveLan para comunicarse remotamente con su centro de mando.

En la tabla 1-1 se muestra una comparación entre diferentes tipos de robots que son utilizados para realizar pruebas científicas y tecnologías y que están basados en una computadora personal, tal cual lo hace la SRM. El dato más interesante de esta tabla, es la relación que existe entre los precios de

5 <http://www.fira.net/>

6 <http://www.aboutai.net/DesktopDefault.aspx?article=aa090600a.htm&tabid=2>

cada robot con el costo de la SRM; es decir, el costo de la SRM no representa ni el 10% del precio en el mercado de robots.

<i>Nombre</i>	<i>Fabricante</i>	<i>Procesador /velocidad</i>	<i>RAM</i>	<i>Disco duro</i>	<i>Enlace</i>	<i>Motores</i>	<i>Sensores</i>	<i>Video</i>	<i>Precio [USD]</i>
iRobot	iRobot	K6-2/ 400MHz	64MB	6 GB	Wave LAN	-	-	Digital	\$10,000
B21	RWI	Pentium	-	-	Wave LAN	4	<ul style="list-style-type: none"> • 48 Sonar • 24 IR • 56 Táctiles 	No	\$20,000
Magellan	RWI	Pentium	-	-	Wave LAN	2	<ul style="list-style-type: none"> • 16 Sonar • 16 IR • 16 Táctiles 	No	\$10,000
SRM	DEA, IIMAS	VIA C3 10000 (compatible con Pentium)/1GHz	256MB	40 GB	WiFi 802.11a	8	<ul style="list-style-type: none"> • 2 Odómetros • 6 sonar • 1 compás 	USB Web Cam	\$800 (costo)

Tabla 1-1. Cuadro de comparación de diversos robots.

Sin duda donde la robótica tiene el más amplio grado de especialización y donde se invierten más recursos, es en la exploración espacial. En 1959, la sonda Lunik 2⁷ del programa espacial soviético, fue el primer objeto robotizado que se envió a la superficie lunar y pudo aterrizar. En 1975, las sondas Viking I y II son lanzadas hacia Marte y en agosto y julio de 1980 respectivamente, tocan suelo marciano. El 27 de Septiembre de 1997, el programa espacial estadounidense NASA, realiza la primera expedición a través de una sonda robot móvil llamada Sojourner⁸ (ver figura 1-4), la cual iba encapsulada en la sonda espacial *Mars Pathfinder*. El Sojourner es un vehículo con 6 ruedas de 28 cm de alto y 63 cm de largo. Al existir un retardo de alrededor de 10 minutos en la comunicación desde la Tierra hasta Marte y viceversa, dependiendo de la posición de la Tierra y Marte, era necesario que tuviera cierta inteligencia artificial, capaz de evitar que el robot chocara con una piedra o cayera de un risco. Para entonces este tenía un microprocesador Intel 80C85 de 100 kHz. La potencia eléctrica fue suministrada por una celda solar de 0.2 m². Además, contaba con un sistema de procesamiento de imágenes blanco, negro y color con el cual se pudo tomar una imagen

7 <http://www.solarviews.com/span/craft2.htm>

8 <http://www.solarviews.com/span/rover.htm>

panorámica de 120° de la superficie marciana. La parte sustancial de esta sonda es que portaba un espectrómetro de rayos alfa-protones-X (APXS) para determinar la composición de las rocas y del suelo marciano. La misión completa tuvo un costo aproximado de 250 millones de dólares.



Figura 1-4. Sojourner del programa espacial NASA

En el pasado y aún en el presente la soldadura, corte y manipulación de placas, tubos y barras de material ferroso y muy pesado, se realiza con la fuerza motriz del brazo de una persona, entonces el soldador o el herrero con soplete en mano, realiza los cortes o soldaduras con solo la precisión que le brinda años de experiencia laboral. De esta manera, el obrero arriesga su vida o sus miembros al exponerse a tales temperaturas y ante tales masas de material, donde el desprendimiento de una placa de menos de un cuarto de tonelada es suficiente para incapacitar al obrero de por vida, o el ruido generado por esmeriladoras podría dejarlo sordo, o la luz incandescente de la soldadura eléctrica es capaz de dejarlo ciego. Ante tal situación fue necesario idear máquinas herramientas, capaces de realizar tan peligrosas tareas. Ahora, existen grandes brazos robóticos auxiliados con grúas automatizadas, que realizan el corte de placas de más de 10 pulgadas de espesor y con un grado de precisión menor a la décima de pulgada, o robots submarinos que son capaces de soldar una tubería de gas natural de aproximadamente 4 pies de diámetro a más de 2000 km bajo el nivel

del mar. La industria automotriz realiza más del 50% de sus procesos de ensamblado con máquinas autómatas que realizan tareas como soldar, perforar, ensamblar, remachar y hasta probar las condiciones de un auto.

En la industria minera no se podría concebir la cantidad de material que se puede extraer actualmente, sin las máquinas autónomas o con cierto grado de inteligencia, que perforan, barrenan, extraen, trasladan y hasta clasifican los minerales. En la fundición, gracias a estos aparatos, se diseñan metales y aceros resistentes a las inclemencias del tiempo y a las deflexiones producidas por cargas, en menor tiempo y a mejor precio.

Los robots de servicio ya están al alcance del usuario común. Estos aparatos inteligentes le proporcionan al humano una excelente herramienta para tener una mejor calidad de vida. Las segadoras robóticas son una excelente solución que han adoptado los gobiernos para recortar todas las áreas verdes cubiertas de pasto en las zonas comunes de las ciudades, así como también los dueños de campos de Golf y de canchas de fútbol que se han visto beneficiados. Estas segadoras robots, gracias a su inteligencia artificial, generan rutas de podado estratégicas que evitan colisionar con objetos, con otras podadoras y pasar más de una vez por el mismo lugar.

Otros robots de servicio son las barredoras automáticas, existen terrestres y hay algunos prototipos para trabajar en mantos de agua, lagos, lagunas y mares. Estos robots planean sus rutas en virtud de que no volverán a pasar por la misma posición que ya limpiaron. Como aditamento, estas incluyen adicionalmente cepillos giratorios, mangueras con agua y jabón y aspiradora con los que absorben la basura removida para posteriormente arrojarla a contenedores.

Existe otra gama de robots que empiezan a posicionarse en el mercado de las empresas, y son los robots limpiadores de tuberías. Estos cuentan algoritmos avanzados, con equipamiento sofisticado para identificar basuras, orificios o grietas para prevenir fugas en oleoductos, gaseoductos o tuberías de sustancias peligrosas. Debido al ambiente tan adverso que representan las tuberías, estos robots presentan características morfológicas muy parecidas a las de un gusano u oruga. Cuentan con mecanismos muy caprichosos para poder desplazarse por medio de las curvas de toda una red de

tubos.

Dependiendo de la actividad que se realice en cada campo de aplicación, los robots adquieren ciertas formas y se les programan ciertas habilidades para adaptarse de mejor forma al proceso que van a desempeñar, de tal manera que existen robots tan grandes como las grúas que cargan a barcos cargueros, hasta robots del tamaño de una avispa capaces de tomar fotografías para el espionaje. Actualmente se está desarrollando una industria que va a revolucionar la forma de ver a la tecnología, que seguramente resolverá y tendrá múltiples aplicaciones, que es la nanotecnología; el motor más pequeño mide apenas 100 micras de metro y sus aplicaciones son bastas. Sin embargo, la nanociencia plantea que es posible alterar la estructura molecular para desarrollar mecanismos o circuitos electrónicos del tamaño de una mil millonésima parte de un metro. Con esto, cada átomo tendrá una tarea en particular y así se podrá conseguir que estos actúen como robots. De tal manera, que los nanorobots sean capaces de fabricar o reparar molecularmente a otros robots u otras máquinas. Otra aplicación importante sería en la medicina donde a partir de uno o una cuadrilla de nanorobots se introducirán en el torrente sanguíneo de un paciente que padezca colesterol y prácticamente se drenará la grasa acumulada en las arterias o venas. O bien introducir una dosis de nanorobots por medio de una jeringa, en tumores cancerígenos y no solo aislar las células dañadas, sino poder reparar molecularmente los tejidos de un órgano vital como es el cerebro.

Los alcances de la nanociencia y de la nanotecnología no tienen límites y quizá en el futuro se alcance a desarrollar un humanoide con inteligencia que sea capaz de realizar cualquier tarea que se le ordene, que tenga la fuerza de cualquier máquina, la precisión y desde luego que pudiera procesar sus propias fuentes de energía como pudiera ser la sintetización de alimentos orgánicos.

1.5 La robótica en México

Los avances de la robótica en México no están a la altura de otros países. De acuerdo a un estudio realizado por la Federación Internacional de Robótica (IFR⁹) y la *United Nations Economic Commission for Europe* (UNECE) en el 2005, se observa que 1,500,000 de robots industriales fueron vendidos desde finales de los años sesenta, al término del año 2004. Aunque claro, ya algunos no se encuentran en funcionamiento; por lo que aproximadamente existen entre 848,000 unidades funcionando. En este estudio se muestra una tabla comparativa del número de robots multipropósito industrial instalados al año en cada país y/o región (ver tabla 1-2).

En esta tabla se puede apreciar que Japón es el líder mundial, ya que representa el 42.05% con 356,483 unidades, mientras que México cuenta con solo el 0.25% del total de los robots fabricados hasta finales del 2004 en el Mundo. Con estos datos es fácil concluir que México no contribuye significativamente al desarrollo de robots a nivel mundial.

País	2004	2004 [%]	2008
América	125,235	14.77	159,900
México	2,137	0.25	-
EUA	115,283	13.60	-
Asia y Australia	443,193	52.28	532,900
Japón	356,483	42.05	-
Europa	278,906	32.90	348,100
Alemania	120,544	14.22	-
Africa	430	0.05	800
Total	847,764	100	1,041,700

Tabla 1-2. Total de robots industriales en operación por año y por país o región. Basada en datos de la Federación Internacional de Robótica

A pesar de las desalentadoras cifras mostradas anteriormente, la mayoría de las universidades y de los centros de investigación del país cuentan con laboratorios o departamentos especializados en el estudio de la robótica. Sin embargo, estos centros se encuentran desarticulados entre sí, pues no existe una estructura de trabajo compartida, donde las experiencias y aprendizaje adquiridos se compartan. Es más, dentro de cada universidad, los grupos de investigación ya sean de control y

9 <http://www.ifr.org/statistics/keyData2005.htm>

automatización con laboratorios de electrónica, o con laboratorios de inteligencia artificial ni siquiera existe una comunicación latente. Claro, se desarrollan conferencias y congresos relacionados con el tema, pero difícilmente se concreta un solo lazo de colaboración entre los distintos departamentos de investigación. La academia y la investigación no han encontrado una línea directa que pudiera agilizar la canalización de los alumnos hacia los centros de investigación. No existen grupos de trabajo, y si existen son de muy poco alcance para relacionar al alumno con esta área de investigación. En la UNAM, no existe una carrera o un posgrado que plantee de lleno un temario a la altura de las grandes universidades especializadas en el tema, que aborde el tema de los robots.

De los pocos lugares donde se desarrolla la robótica se puede concluir lo siguiente. Las empresas, los estudiantes y los investigadores enfocados en la robótica, emplean la mayoría de su tiempo y de sus recursos para competir en concursos de robótica. La mayoría de los robots que hasta ahora se han desarrollado en México tienen más bien propósito de diversión y entretenimiento, que para resolver un problema en la industria, en la medicina o en la investigación. En este sentido, existe un importante evento a nivel mundial para todos los interesados en la robótica, y es la “Robo-cup”; donde robots programados juegan al fútbol y desarrollan técnicas de compañerismo y de equipo para poder anotar. Otros grandes concursos, también a nivel mundial, son los de robots seguidores de línea, ya sea en un circuito cerrado, abierto, continuo, discontinuo con y sin bifurcaciones, el robot ganador debe llegar en el menor tiempo posible a la meta. Complicadas técnicas de programación hacen que los robots consigan el triunfo ya sea en cada nivel de competencia. También hay robots de exposición, es decir, gana el mejor diseño mecánico, antropomórfico o de programación.

Sin embargo, las técnicas desarrolladas, así como los mecanismos y los circuitos electrónicos que se emplean para estas competencias sirven en gran medida a la construcción de robots más capaces, más veloces y más baratos, que al final se puedan emplear en la industria médica o en la extracción de minerales por mencionar algunas.

De igual forma, el diseño de robots de servicio en México corren con la misma suerte. Sin importar que las empresas más importantes, a nivel mundial, de fabricación de autos tienen establecidas líneas

de ensamblaje en nuestro país, la industria de robots mexicanos no se ha desarrollado. En las líneas de ensamblaje de autos de estas empresas se utilizan manipuladores de gran tamaño que son diseñados y construidos en otros países. En la mayoría se utilizan robots llamados Kuka (ver figura 1-5) que son de origen alemán. Este manipulador tiene un costo de aproximadamente 250 mil dólares.



Figura 1-5 Robot KUKA en una línea de ensamblaje de autos.

1.6 Estado inicial de la Sonda Robot Móvil

En esta sección se describe el estado inicial del proyecto de la actualización de la SRM. Después de que fue creada esta sonda sufrió su primera actualización y se le agregaron los sensores ultrasónicos, un compás electrónico y 2 odómetros. Al final de esta primera actualización la SRM tenía las siguientes características.¹⁰

- La SRM es un vehículo de tres ruedas que soportan una placa hexagonal de aluminio (chasis) de 1cm de espesor con dimensiones de 66 cm x 43 cm x 50 cm (ver figura 1.6). Dos de las llantas son traseras (solo soporte) y tienen un diámetro de 13 cm, mientras que la tercer llanta es delantera (tracción y dirección) y tiene un diámetro de 14.5 cm.

¹⁰ Hernández, Luis., *Tesis "Percepción de un robot móvil durante su navegación para evadir obstáculos"*, México, 2002, UNAM

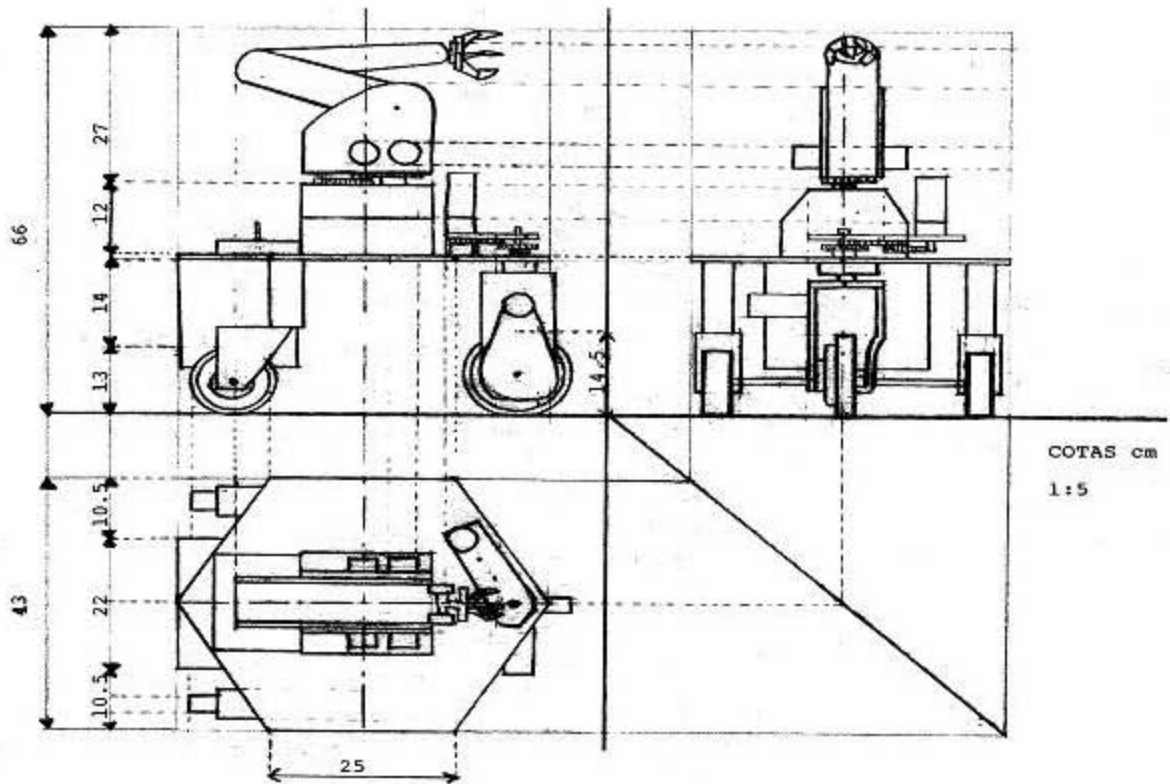


Figura 1-6. Esquema de la SRM

- En la parte superior está instalado un brazo manipulador de seis grados de libertad (base, hombro, codo, mano, muñeca y pinza), con 6 motores de pasos de 4 fases cada uno y se alimentan con $12 V_{CD}$ a 250 mA.

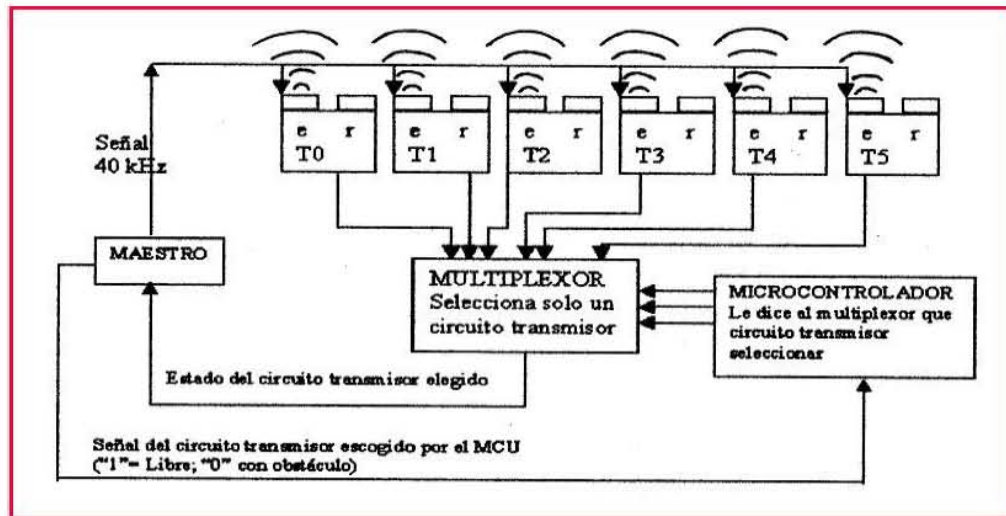


Figura 1-7 Diagrama de control del sensor ultrasónico.

- Para la tracción se utiliza un motor de corriente directa de 12 V a 2 A. Para la dirección se utiliza un motor de corriente directa de 12 V a 500 mA.
- En la parte posterior de la sonda, abajo del chasis, lleva una batería de 12 V a 7 Ah.
- Cuenta con 4 tarjetas con componentes electrónicos.
 - ✓ 2 tarjetas de potencia para el control de 3 motores de pasos cada una.
 - ✓ 1 tarjeta de potencia para el control de los motores de tracción y dirección.
 - ✓ 1 tarjeta de control que contiene un microcontrolador 8751H de INTEL, un convertidor A/D de 8 bits y 16 canales, una interfaz paralela PPI 82C55 y periféricos extras.

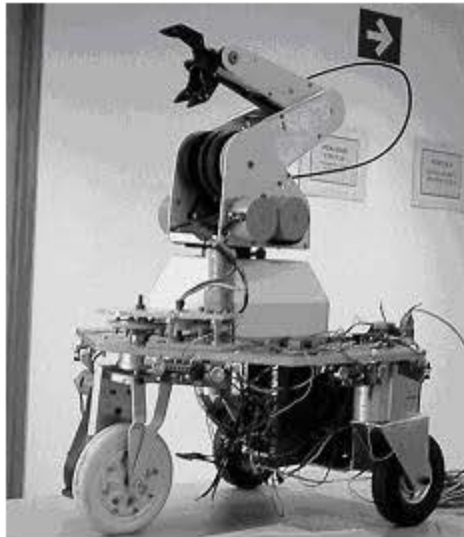


Figura 1-8 Fotografía de la SRM.

- 2 Odómetros
 - ✓ Disco ranurado de 7 cm de diámetro
 - ✓ 18° por pulso (4 cm por pulso)
 - ✓ Tiempo de muestreo cada 50 ms

- Sensor ultrasónico
 - ✓ 1 Circuito maestro.
 - ✓ 6 transmisores ultrasónicos de 40 kHz
 - ✓ Multiplexor analógico (ver figura 1-8)
 - ✓ Tiempo de muestreo 38.5 ms
 - ✓ 73° de campo de visión
 - ✓ Distancia mínima de detección 5 cm

- Compás electrónico
 - ✓ 360° de posiciones
 - ✓ Error de precisión 2°
 - ✓ Error de exactitud 5°

Para conocer la distribución de los puertos y los dispositivos asociados, se presenta la tabla 1-3.

<i>Puerto</i>	<i>Dispositivo</i>	<i>Función</i>	<i>Descripción de cada bit</i>	
Puerto A	82C55	Control del compás electrónico. Todas son salidas	PA.0	PC
			PA.1	SS
			PA.2	CLK
			PA.3	RESET
			PA.4	(+)
			PA.5	(-)
			PA.6	Libre
			PA.7	Libre
Puerto B	82C55	Control del multiplexor del sensor ultrasónico y motores de tracción y dirección. Todas son salidas	PB.0	Pto0 (Mux 0)
			PB.1	Pto1 (Mux 1)
			PB.2	Pto2 (Mux 2)
			PB.3	Libre
			PB.4	Tracción (atrás)
			PB.5	Tracción (adelante)
			PB.6	Dirección (horario)
			PB.7	Dirección (antihorario)
Puerto C	82C55	Control de dos motores de pasos del manipulador. Todas son salidas	PC.0	MP1.0
			PC.1	MP1.1
			PC.2	MP1.2
			PC.3	MP1.3
			PC.4	MP2.0
			PC.5	MP2.1
			PC.6	MP2.2
			PC.7	MP2.3
Puerto 1	8751H	Datos provenientes del sensor ultrasónico, odómetros y compás electrónico. Todas son entradas.	P1.0	Salida del 7805
			P1.1	Odómetro (derecho)
			P1.2	Odómetro (izquierdo)
			P1.3	Libre
			P1.4	Libre
			P1.5	Libre
			P1.6	EOC (compás)
			P1.7	SDO (compás)

Tabla 1-3 Disposición de puertos al inicio del proyecto

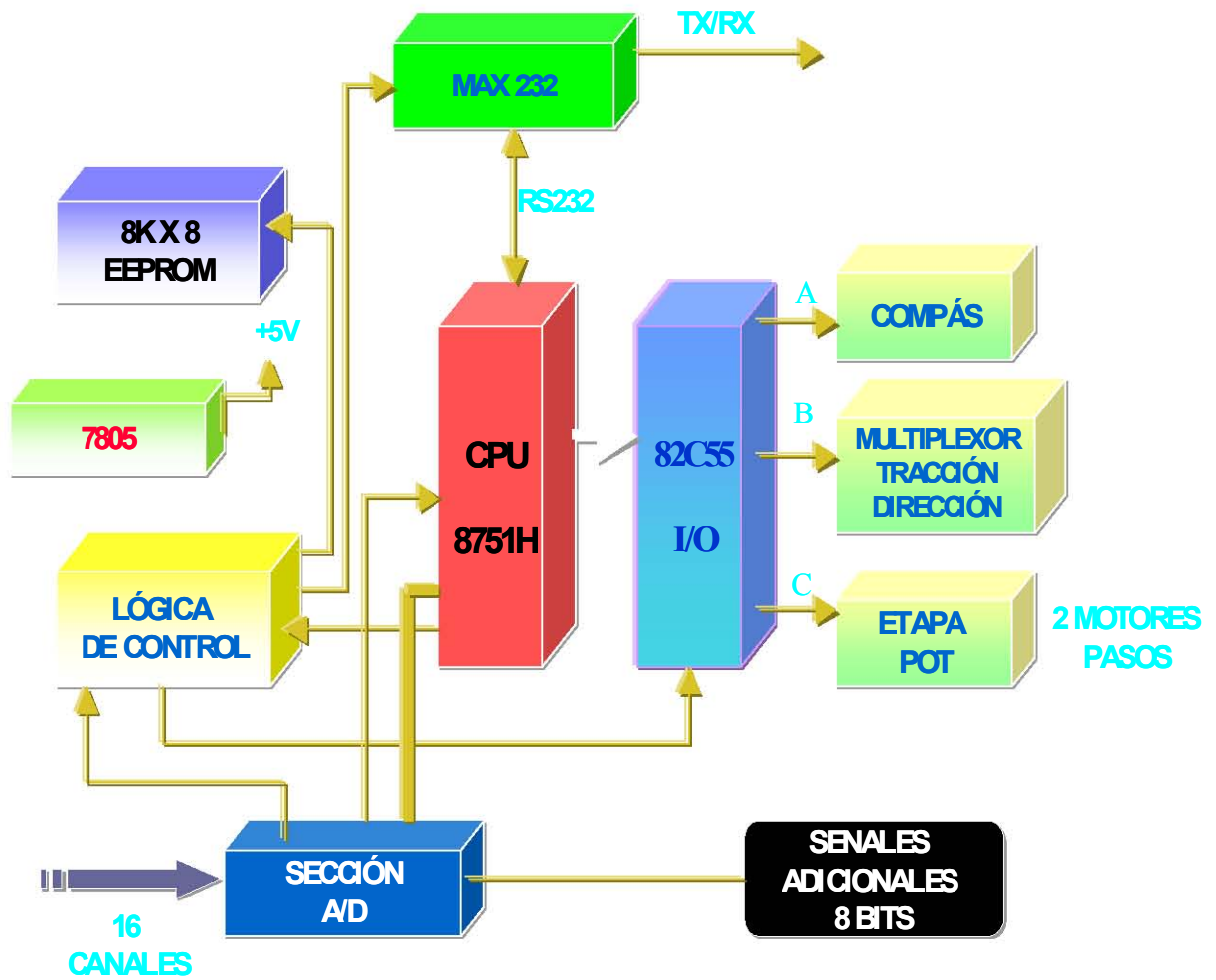


Figura 1-9. Esquema electrónico de la SRM

COMPONENTES DE LA ACTUALIZACIÓN

Con el propósito de exponer al lector el motivo de la elección de las tecnologías, así como de los dispositivos utilizados en la actualización de la SRM, en este capítulo se proponen alternativas que bien pudieron haber servido para la actualización, pero que por diversas razones (descritas posteriormente) fueron descartadas y se optó por el camino que llevó a la culminación de este trabajo. En la parte intermedia y final de este capítulo se describen las características de las tecnologías y dispositivos utilizadas en esta actualización.

2.1 Requerimientos computacionales de la Sonda Robot Móvil

En el capítulo anterior, se describieron las características de la Sonda Robot Móvil antes de la actualización. Pues si bien esta sonda ya contaba con una estructura, en cierto grado flexible, ésta no era lo suficientemente capaz de cumplir con los nuevos objetivos y metas que se plantearon en este proyecto. Es decir, no bastaba con que el módulo de control de la SRM (Sonda Robot Móvil) fuese capaz de controlar los motores, de revisar el estado de los sensores y sobre todo comunicarse con el centro de mando, sino que a largo plazo se requería de aumentar las posibilidades de operación y supervisión desde cualquier punto. Además era necesario poder, en cualquier momento, aumentar el número de dispositivos o herramientas (pinzas mecánicas, cortadores, destornilladores todo ellos automáticos) que facilitaran la operación en el área de trabajo. Pero la meta más ambiciosa de este proyecto, es la de otorgarle mayor autonomía a la sonda por medio de algoritmos de inteligencia artificial, que por lo complejo que son, no hubiese sido posible ejecutarlos en el sistema de cómputo con el que contaba. Parte de estas mejoras se prevé, en desarrollos posteriores a este trabajo, que por medio de una cámara de video instalada y conectada al nuevo sistema de cómputo sea posible implementar algoritmos de procesamiento de imágenes con el fin de auto dirigir la sonda a un determinado lugar. Esto complementado con el “Softelenav” (software desarrollado en el DEA del IIMAS¹¹) que es un escenario de realidad virtual. Este ambiente de realidad virtual tiene el propósito

¹¹ Arteaga, Alma. *Tesis “Desarrollo del software de un sistema para la operación de un robot móvil en un ambiente de realidad virtual 3D”*, México, 2002, UNAM

de que mediante coordenadas y especificaciones de tamaño, aparezcan objetos virtuales que representen escritorios, sillas o paredes del ambiente de trabajo donde opera la SRM.

La forma de auto dirigir la sonda será por métodos conocidos como “visión robótica”, que consisten en dotar de un medio de percepción del lugar donde el robot se desempeña. Estos algoritmos están siendo desarrollados en el laboratorio (DEA) y consisten en el seguimiento de rutas así como la identificación de objetos por su forma.

El seguimiento de rutas permitirá al robot, a través de líneas pintadas de diferentes colores sobre el suelo, paredes e incluso en el techo y sin tener conocimiento de ello, seguir la ruta que tendrá un destino predeterminado en función del croquis de la planta, taller, bodega o espacio de trabajo. Estas rutas llevarán a zonas de seguridad, centros de control, salidas, zonas riesgosas y otras. La sonda a través de su cámara y su sistema de procesamiento debe controlar los sistemas motrices para desplazarla.

Otra técnica utilizada se refiere al reconocimiento de formas y tamaños. Esto es, a partir de una imagen o una sucesión de imágenes, se analizan píxel a píxel y cuadro a cuadro las variaciones que existen entre cada píxel y cada cuadro. En función de esas diferencias o ausencia de diferencias se establece un patrón y en consecuencia una decisión. Así, los algoritmos que determinan dichas diferencias pueden asociarlas con formas circulares, rectangulares, o simplemente lineales. Una vez que se realiza esta asociación, las formas se pueden vincular a través de parámetros ya preestablecidos en bases de datos y así determinar la forma del objeto analizado. Con la identificación se pueden realizar diversas labores, como la autenticación de personas por medio de huellas dactilares, identificación de rostros, lectores de códigos de barras en una y hasta en tres dimensiones ó en lo que atañe a este trabajo, la identificación de válvulas, tuberías, extintores, herramientas y otros. Los algoritmos adecuados permitirán que el operador de la SRM agilice su actividad con cierto grado de seguridad de que la tarea que realiza es correcta.

Una vez que se realiza la identificación de formas, es posible mediante otro tipo de procesamiento, determinar la posición y distancia total relativa entre la sonda y el objeto en cuestión. Esto facilita en

mucho la manipulación de la sonda y del brazo robótico. Ya que con ayuda de esta inteligencia artificial el operador de la sonda se sentirá mas confiado al momento de tomar una pieza o de acelerar en una zona con obstáculos.

Los algoritmos de visión robótica, son en extremo complejos y requieren de grandes capacidades de cómputo. Actualmente el procesamiento de datos en celulares, cámaras de video, cámaras fotográficas digitales, grabadoras y reproductoras de audio y lectores de códigos de barras se realiza en procesadores digitales de alto desempeño, llamados DSP (Procesador digital de señales). Estos dispositivos integran funciones tanto en hardware como en software, módulos internos que realizan varias operaciones en uno o dos ciclos de reloj, lo que agiliza el procesamiento del dato. Por lo que bien se pudo haber planteado la solución a estos nuevos requerimientos, incorporando un DSP en la sonda. Sin embargo, las cantidad de código que arrojaría el desarrollar dicho software sería demasiado y seguramente se recurriría a incorporar más bancos de memoria.

2.2 Computadora principal de la SRM

El tema de la computadora principal en la SRM es el más complejo a desarrollar; debido a que es donde el robot toma decisiones, en función del estado de los sensores de posición (odometría y brújula digital), sensores de impacto (ultrasonido), instrucciones desde la estación de control y el software de control. Como se mostró en la sección anterior, se requiere que la capacidad de cómputo de la computadora principal de la sonda, sea de alta velocidad y con espacio suficiente para almacenar datos.

El microcontrolador con el que contaba la SRM, el Intel 8751¹², tiene una longitud de palabra de cada instrucción de 8 bits, cuenta con 128 bytes de memoria RAM y tiene una frecuencia de operación de 24 Mhz; con estas características el 8751 resulta poco práctico para implementar los algoritmos y exigencias que requieren la sonda.

Otra alternativa que se consideró fue la de reducir el grado de dependencia de la sonda. Es decir, ya

12 <http://www.intel.com/design/mcs51/datashts/27233504.pdf>

que se necesita cierta capacidad de procesamiento de cómputo, ¿por qué no, este es realizado desde la estación de control y solo envía los datos de control transmitidos inalámbricamente a la sonda? Es decir, que la estación de control cuente con una computadora que realice el procesamiento de las imágenes enviadas desde la cámara del robot, esta las procesara y solo enviará ciertas instrucciones que el procesador del robot tendrá que recibir, interpretar y actuar directamente con la fuerza motriz de la sonda. Sin embargo, se ha planteado que la sonda debe de poder tomar su propias decisiones, por lo cual se descartó esta posibilidad.

Dadas las propuestas anteriores y sus limitaciones se perfila como la mejor opción, proveer a la sonda robot con un sistema de cómputo basado en una arquitectura x86, además de un sistema operativo. Estas ventajas representan una buena opción para llevar a cabo la actualización de la SRM.

2.2.1 Arquitectura x86

Los lenguajes de programación de alto nivel¹³ han revolucionado el cómputo. Ya que a través de interpretaciones y de una estructura semi formal del lenguaje común de los humanos, el compilador realiza la traducción y síntesis de dicho lenguaje y lo convierte en una manera muy eficaz en instrucciones para la máquina, el lenguaje ensamblador. Con lo que prácticamente se puede construir cualquier software que el usuario requiera, por lo tanto desarrollar algoritmos para procesar imágenes digitales o implementar software de inteligencia artificial se reduce a unas cuantas instrucciones.

Sin embargo, estos lenguajes de programación están orientados al desarrollo de software con base en cierta estructura. El surgimiento de estos compiladores se debe a la respuesta natural del desarrollo de la computación, cuanto más sofisticado se requiere el software más herramientas sofisticadas se necesitan construir. Por lo que el desarrollo de nuevos o mejores compiladores y el aumento en las capacidades del sistema operativo van de la mano. Los sistemas operativos son la parte estratégica

¹³ Torres, J., *Conceptos de sistemas operativos*, Trillas, México, 2001, pp. 18-20.

del desarrollo computacional.

Un sistema operativo tiene la función de administrar los recursos de hardware de una computadora, como son el control del tiempo de un proceso, de un usuario, de un hardware, de un software y sobre todo de proporcionar un ambiente amigable al usuario¹⁴. Existe una gran cantidad de sistemas operativos desarrollados actualmente. Unos enfocados a controlar electrodomésticos, otros más complejos dirigidos a controlar máquinas automáticas, procesos de producción, servidores, estaciones de trabajo y computadoras personales. La principal diferencia entre cada sistema operativo radica en la simpleza o la complejidad de la función que deberá controlar dicho software. Por ejemplo, para un teléfono celular, su sistema operativo tendrá que atender las teclas que presiona el usuario y en función de ello tomar alguna decisión. A diferencia del sistema operativo de un servidor de páginas HTML, el cual está enfocado a proporcionar datos con cierta estructura cuando el cliente lo solicite.

Como se mencionó antes, el software desarrollado para la SRM requiere de ciertas condiciones que van en función del tipo de sistema operativo y del lenguaje de programación. El sistema operativo debe de poder ejecutar las librerías utilizadas por la aplicación correspondiente. Para el caso del software “Softelenav” es necesario utilizar las librerías de procesamiento gráfico de Microsoft Windows, controles ActiveX y como esta aplicación está desarrollada en Visual Basic 6.0 este sistema debe de ser compatible para ejecutar dicha aplicación.

Dadas las condiciones anteriormente expuestas es necesario que dicha plataforma de cómputo sea compatible con el sistema operativo Windows de Microsoft en cualquiera de sus versiones posteriores a la de 1998 (Windows 2000, Windows XP, etc.)¹⁵ debido a que dichas librerías solo son compatibles con estas versiones. Aquí surge un problema muy importante, y es que los requerimientos de hardware para ejecutar este sistema operativo deben de cumplir con ciertas características.

Para poder satisfacer estos requerimientos de cómputo es necesaria la incorporación de una

14 http://es.wikipedia.org/wiki/Sistema_operativo

15 http://es.wikipedia.org/wiki/Microsoft_Windows

computadora personal, compatible con la arquitectura x86, además de cumplir los requerimientos de la tabla comparativa 2-1. En esta tabla se presentan distintas configuraciones requeridas para ejecutar alguna de las versiones del sistema operativo de Microsoft y como medida de comparación, se agrega una fila del sistema operativo GNU/Linux.

<i>Sistema Operativo</i>	<i>Procesador</i>	<i>Velocidad del procesador</i>	<i>Memoria RAM</i>	<i>Espacio en disco duro</i>
Windows 98 SE	Pentium o equivalente	100MHz	24 Mbytes	195 Mbytes
Windows ME	Pentium o equivalente	150 MHz	32 Mbytes	420 Mbytes
Windows 2000	Pentium o equivalente	500 MHz	64 Mbytes	1 Gbytes
Windows XP	Pentium II o equivalente	450 MHz	256 Mbytes	1 Gbytes
Windows 2003	Pentium II o equivalente	450 MHz	256 Mbytes	1 Gbytes
Windows Vista*	Pentium III o equivalente	2.5 GHz	512 Mbytes	1.5 Gbyte
Kernel 2.8 de GNU/Linux	x386	16 MHz	2 Mbytes	64 Mbytes

Tabla 2-1 Características mínimas requeridas de hardware para ejecutar un sistema operativo. *En desarrollo

La estructura de una computadora personal consta de un microprocesador, memoria y dispositivos de entrada/salida. La base donde se coloca el microprocesador y la memoria se llama tarjeta principal, tarjeta madre o placa base. Esta última, contiene a la memoria y al procesador; la placa base concentra todos los circuitos electrónicos de soporte y las ranuras (slots) para insertar otras tarjetas de circuito que controlan a los periféricos como el monitor, teclado, ratón, audio, video, modems y otros.

La tarjeta madre sirve para brindar soporte físico a las tarjetas de expansión, recibe y distribuye la energía eléctrica a todos los componentes electrónicos, enlaza todos los dispositivos por medio de líneas físicas distribuidas por ambos lados de la tarjeta, incluso en capas intermedias, proporciona sincronía en tiempo y forma a todos los procesos y sobre todo realiza revisiones de temperatura para

supervisar y controlar su estructura.

De los principales componentes electrónicos que porta una placa base se encuentra el *chipset*. Este es el conjunto de dos integrados que prácticamente administran la configuración intrínseca de la tarjeta madre. Estos dos integrados se llaman *NorthBridge* y *SouthBridge*. El primero controla las funciones de acceso hacia y entre el microprocesador, la memoria RAM, el puerto gráfico AGP, y las comunicaciones con el *SouthBridge*. El segundo controla los dispositivos asociados, como la controladora de discos IDE (*Integrated Device Electronics*), puertos USB, Firewire, SATA, RAID (*Redundant Array of Inexpensive / Independent Disks*), ranuras PCI, ranura AMR (*Audio Modem Riser*), ranura CNR (*Communication and Network Riser*), puertos infrarrojos, disquetera, LAN y una larga lista de elementos integrados en la placa madre.

El BIOS (*Basic Inputs and Outputs*) es otro componente muy importante en la tarjeta madre, ya que en este circuito integrado se almacenan el programa (*firmware*) de configuración y del estado del sistema. En este dispositivo se almacena el listado de las características del hardware conectado a la placa madre que inicializa el sistema antes de que el sistema operativo entre en función.

Existen diversos modelos de tarjetas madre en el mercado y se pueden clasificar de diferentes formas, una de ellas es por su tamaño y su uso. Ya que dependiendo del tamaño físico y la cantidad de componentes que están soldados en las caras de la tarjeta, la distribución se hace óptima ya sea para servidores, consolas de video juegos, computadoras caseras y en este caso del proyecto, de computadoras embebidas. En la tabla 2-2 se muestran varios tipos de placas madre. Que si bien el tamaño importa, en todos los formatos se mantiene la misma estructura.

Las diferencias radican en el número de ranuras para memoria RAM y para puertos de expansión, en el grado de integración de los componentes (audio, video, LAN, modem integrados), ranuras para dispositivos de almacenamientos (discos duros, lectores de CD-ROM y DVD, tarjetas de memoria, etc.) y ranuras de puertos básicos (puerto paralelo, puerto serie, DIN, mini-DIN, USB). Además, el tipo de formato depende de la variedad de tecnologías que porte, por ejemplo puede traer más de un chip de BIOS para hacer respaldos o bien para realizar lo que se conoce como *overclocking*.

El microprocesador constituye la parte principal de la computadora personal, ahí se realizan todas las operaciones aritméticas, lógicas, administra las interrupciones, los accesos directos a memoria, envía y recibe datos y los transforma. Existen solo dos fabricantes que construyen microprocesadores para arquitectura x86, Intel Corp. y AMD Inc. Intel patentó su primer diseño de procesador 4004 que fue utilizado en una calculadora en el año 1971. Es así como se consolida como la empresa más grande en la fabricación de microprocesadores. En febrero de 1982, AMD firma un acuerdo con Intel mediante el cual obtiene la licencia de fabricación de los procesadores 8086 y 8088, que son propiedad intelectual de Intel¹⁶. IBM quería usar el procesador 8088 en su IBM PC, pero las políticas de IBM exigían al menos dos proveedores para sus chips. AMD produjo después, bajo el mismo acuerdo procesadores 80286, o 286, pero Intel canceló el contrato en 1986, rehusándose a revelar detalles técnicos de i386. La creciente popularidad del mercado de los clones de PC significaba que Intel podría producir procesadores según sus propios términos y no según los de IBM. Hoy en día Intel y AMD fabrican sus propios diseños de microprocesadores y firman acuerdos de compatibilidad.

<i>Formato</i>	<i>Ancho [cm]</i>	<i>Largo [cm]</i>	<i>Uso</i>
Extended ATX	30.48	33.02	Servidores, estaciones de trabajo. Más de un microprocesador.
Standard ATX	30.48	24.38	Computadoras personales.
Mini ATX	28.45	20.83	Computadoras personales.
MicroATX	24.4	24.4	Computadoras personales. Chasis pequeños.
Mini ITX	17	17	Computadoras personales, consolas, cajas registradoras, sistemas embebidos, etc.
Nano ITX	12	12	Computadoras personales, consolas, cajas registradoras, sistemas embebidos, etc.

Tabla 2-2 Formatos de computadoras personales¹⁷

La memoria de una computadora personal, se puede entender como el medio físico donde se almacenan los datos y el programa que va a ejecutar el microprocesador. Se puede hablar de dos

¹⁶ Brey, B., *Los Microprocesadores Intel, Arquitectura, Programación e Interfaces*, Prentice Hall, México, 1994, pp 256-257

¹⁷ Intel, *ATX Specifications*, Intel, 2002

tipos de memoria, la principal y la secundaria. Esta clasificación se refiere a la importancia de los datos que almacena en cierto momento. Es decir que en la memoria principal se tienen los datos y el programa que el microprocesador está ejecutando y que está devolviendo como producto de operaciones o manipulaciones. Existen dos tipos de memoria principal. La memoria tipo ROM (Read Only Memory) "Memoria de solo lectura" que almacenan códigos de programa grabados en fábrica. Y la memoria tipo RAM (Random Access Memory) "Memoria de acceso aleatorio", almacena datos que pueden ser escritos y borrados atendiendo a los procesos de computación.

<i>Nombre</i>	<i>Acrónimo</i>	<i>Memoria</i>	<i>Velocidad</i>	<i>Módulo</i>	<i>Usado en:</i>
DRAM	Dynamic RAM)	Asíncrona	80 a 70 ns	SIMM, DIMM	Computadoras 386
FPM-RAM	Fast Page RAM	Asíncrona	70 a 60 ns	-	Hasta los primeros Pentium
EDO-RAM	Extended Data Output RAM	Asíncrona	40 a 30 ns	-	-
SDRAM	Sinchronous RAM	Síncrona	25 a 10 ns	DIMM	Pentium 2
DDR RAM	Double Data Rate RAM	Síncrona	Envía datos dos veces por ciclo de reloj del sistema	-	Pentium 4

Tabla 2-3 Tabla comparativa de memorias RAM

Las memorias se agrupan en módulos, que se conectan a la tarjeta madre de la computadora. Según los tipos de conectores que lleven los módulos, se clasifican en Módulos SIMM (*Single In-line Memory Module*), con 30 o 72 contactos, módulos DIMM (*Dual In-line Memory Module*), con 168 contactos y módulos RIMM (*RAMBUS In-line Memory Module*) con 184 contactos. La tabla 2-3 muestra un comparativo entre distintos tipos de memoria RAM.

2.2.2 Tarjeta Madre EPIA TC 10000E

El modelo de Tarjeta Madre EPIA TC 10000E pertenece al formato de computadoras personales Mini ITX por lo que sus dimensiones físicas son de 17 x 17 cm. Esta tarjeta tiene componentes altamente integrados y son compatibles con la arquitectura x86. Tiene múltiples opciones de conectividad y multimedia que la hacen ser una excelente opción para ambientes de trabajo donde se requiera pequeño tamaño, poco peso y poca generación de ruido.

Esta tarjeta esta disponible para dos tipos de procesadores el VIA C3 o el VIA Eden ESP este último no requiere de medio de ventilación. Ambos fabricados por VIA Technologies. Al momento de solicitar una tarjeta madre es necesario indicar con que procesador se requiere; ya que el procesador se ensambla directamente en la tarjeta madre. Por lo que al hablar de esta tarjeta, también se habla del tipo de procesador. Para el caso del procesador VIA C3, el modelo de la tarjeta madre es EPIA TC 10000E y para el caso del procesador VIA Eden ESP, el modelo de la tarjeta madre es EPIA TC 6000E.

El procesador VIA C3 está diseñado para aplicaciones de mediana intensidad, a diferencia del procesador VIA Eden ESP que está diseñado para aplicaciones de muy bajo consumo de potencia y sistemas silenciosos (ver tabla 2-4).

<i>Parámetros</i>	<i>EPIA TC 6000E</i>	<i>EPIA TC 10000E</i>
Procesador	VIA Eden ESP 6000	VIA C3
Velocidad	600 MHz	1 GHz
Voltaje de operación	1.2 V	1.4 V
Cache L1	128 kB	128 kB
Cache L2	64 kB	64 kB
Soporte multimedia	MMX y 3D Now!	MMX y SSE
Costo (dolares)	150	170

Tabla 2-4 Características de los procesadores Eden ESP y VIA C3

En la tabla 2-5 se presentan algunas características de la EPIA TC. En el Apéndice A se muestran más detalles de esta tarjeta madre

Procesador	<ul style="list-style-type: none"> VIA C3/VIA Eden ESP
Memoria RAM	<ul style="list-style-type: none"> 1 x 200-pin DDR266 SODIMM socket Arriba de 1 GB
Video	<ul style="list-style-type: none"> Controlador de video integrado a la placa madre modelo VIA Uni Chrome 2D/3D, con acelerador para MPGE-2
Expansión	<ul style="list-style-type: none"> 1 PCI
Red	<ul style="list-style-type: none"> Controlador de red modelo VIA Networking Tahoe VT6103 Fast Ethernet 10/100
Sonido	<ul style="list-style-type: none"> Controlador de audio modelo VIA VT1612A 2-canales AC'97

Tabla 2-5 Características la Tarjeta EPIA TC

El *chipset* VIA CLE266 es el *North Bridge* de la EPIA TC, está diseñado para proporcionar alta calidad de video y DVD. Está encargado de controlar los módulos de memoria RAM y video. Por otra parte el *chipset* VIA VT8235 *South Bridge* esta encargado de controlar al ratón, el teclado, puertos serie y paralelo, audio, red, USB, PCI y discos duros.

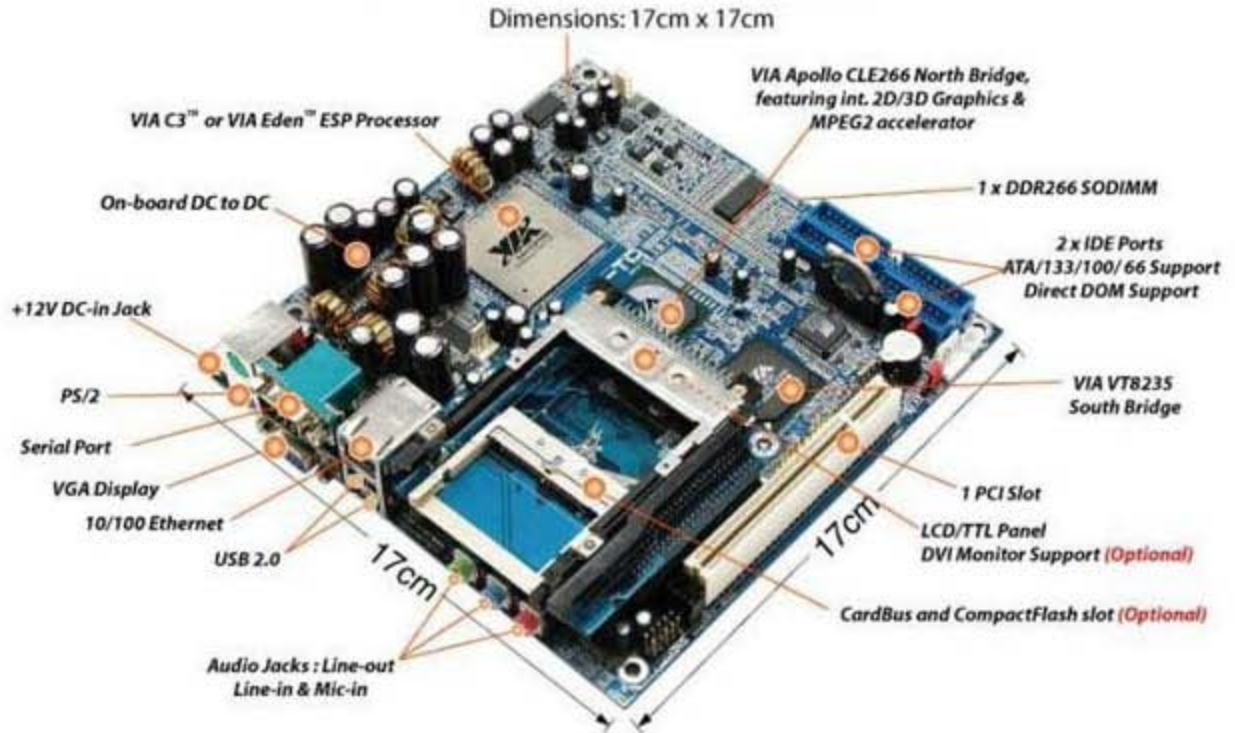


Figura 2-1 Tarjeta Madre EPIA TC

2.3 Expansión del puerto de datos

En el esquema inicial de la sonda robot móvil, se tenían contemplados cierto número de puertos tanto de entrada como de salida de datos digitales o analógicos. Para el envío de información, controlar las etapas de potencia de los motores de tracción y dirección, realizar el barrido de todos los sensores de ultrasonido, recabar información de los sensores de variables ambientales, manipular las etapas de potencia del brazo mecánico, es necesario cierto número de puertos con cierto tamaño en bits. El número de puertos se determina en función de los requerimientos de los actuadores y sensores de la SRM, sin embargo cuando se quiso aumentar el número de dispositivos a controlar, el diseño original de la tarjeta maestra no permitía aumentar el número de puertos. Esto se debe a que la arquitectura de la tarjeta madre incluye al microcontrolador 8751 junto con la PPI 8255 los cuales solo cuentan con 6 puertos de 8 bits cada uno. Estos fueron suficientes mientras el proyecto no creció, ya que cada línea de salida o entrada estaba ocupada intercomunicando con su respectivo dispositivo. Debido a estas razones la actualización contempló incluir más puertos de entrada y salida, dado a que son escasos en el diseño original. Incluso en el capítulo anterior se muestran 2 motores de pasos del brazo manipulador, que no habían sido conectados. Ya ni hablar de incorporar más sensores para la toma de variables ambientales, pues no quedaban líneas disponibles para su control.

Un diseño bien estructurado de una tarjeta madre, permite expandir sus capacidades e incorporar más líneas de datos digitales o analógicos, cosa que no ocurrió en el diseño original. Desafortunadamente los microcontroladores tienen limitado el número de puertos de datos, debido a que un microcontrolador con más de 5 puertos de 8 bits, implica contar con al menos 40 pines sin contar los pines de control, osciladores, tierras y alimentación. Sin embargo existe la posibilidad de configurar un microcontrolador para poder expandir el número de puertos. La arquitectura tipo Von Neumann¹⁸, permite direccionar 2^n puertos, donde n es el número de bits de dirección. Esto es, que si por ejemplo el microcontrolador cuenta con 16 líneas de dirección, se pueden direccionar hasta 65,536 puertos.

18 Tanenbaum, A., *Organización de computadoras un enfoque estructurado*, Pearson Educación, México, 2000, pp. 18-20

Parte de los objetivos de esta actualización a la SRM es el de diseñar una arquitectura expansible. Es decir que cuando el diseño final de la SRM tenga cierto número de puertos y que en un futuro se desee agregar un actuador o un sensor más, este pueda adecuarse agregando únicamente otra tarjeta de puertos sin tener que modificar la tarjeta principal.

El apartado anterior describe el por que utilizar una tarjeta madre de una computadora personal es la mejor opción. De esta manera, la expansión del número de puertos del diseño original al diseño de este proyecto, resulta muy beneficiada. Si ha de mencionarse las ventajas de la arquitectura x86 sobre otros diseños, es precisamente el tema de la gran cantidad de puertos que tiene y la versatilidad del bus de datos para convertirse en puertos de datos.

El primer diseño de computadora personal que lanzó IBM adoptó la gama de procesadores x86 de Intel. Esta computadora personal llamada IBM PC solo contaba con un puerto paralelo y 2 puertos de comunicación serie, incluido el enlace con el teclado¹⁹. Sin embargo, en el diseño de la IBM PC, 1981, contaba con 5 ranuras de expansión (slot) llamado bus ISA, el cual estaba conectado con la mayoría de los pines del microprocesador. Parte de estos pines están en el bus de datos (16 bits), el bus de direcciones (20 bits), señales de alimentación (5 y 12V) y otros de control de escritura, control de lectura así como señales de interrupción. Con este bus ISA se fueron integrando más componentes como modems, tarjetas de video, tarjetas de sonido, tarjetas de red, etc. Poco tiempo después, el bus ISA y su expansión (XT) dejó de ser práctico y surgieron otros modelos de bus más rápidos y versátiles. Ejemplo de ello es el bus PCI (*Peripheral Component Interconnect*), el AGP (*Accelerated/Advance Graphics Port*), SCSI (*Small Computer System Interface*) y últimamente el bus *Firewire* y el USB (*Universal Serial Bus*). En el caso del primero, cada dispositivo conectado al bus es configurado dinámicamente a diferencia del bus ISA, donde por medio de jumpers el técnico asignaba la IRQ (*Interrupt ReQuest*) manualmente para cada dispositivo y así evitar colisiones de datos. Los fabricantes de modems, tarjetas de sonido, tarjetas de video, tarjetas de red alámbricas e inalámbricas han usado el bus PCI por su gran velocidad y diseño estructurado. El segundo, AGP, se ha enfocado a dispositivos de video. El bus AGP presenta ventajas muy superiores al PCI en cuanto a velocidad se refiere, por tal motivo las mejores marcas de tarjetas de video lo han adoptado para su

¹⁹ http://es.wikipedia.org/wiki/IBM_PC

modelo de hardware y software. El sistema SCSI se planteó como una alternativa para conectar dispositivos externos a través de un bus especializado. Actualmente se ha dejado de utilizar en computadoras personales, pero Sun System y Apple Macintosh lo siguen utilizando. Por último, queda el *Firewire* y el USB, que cuentan con características muy propicias para adecuarlas al nuevo diseño de la SRM y que son muy semejantes, a diferencia de las velocidades a las que transmiten.

El USB es la opción más viable para fabricantes de dispositivos, ya que el Firewire apenas comienza su carrera de conseguir un buen prestigio en el mercado y todavía es escasa la información disponible para el desarrollo de aplicaciones. Mientras que el USB lleva ya casi 10 años en el mercado y ya cuenta con una segunda versión mucho más potente que la anterior. Además en la comunidad de desarrolladores tanto de software como de hardware, han adoptado al USB. Además, el USB proporciona una estructura de programación muy adecuada para el desarrollo de pequeñas aplicaciones. En Internet se pueden encontrar gran diversidad de librerías disponibles para la programación de dispositivos con este estándar. “Si USB 2.0 fuera un político, seguramente sería presidente”²⁰, es lo que se lee en una publicación de *National Instruments* de sus productos de LabView 8. El artículo se refiere a la numerosa cantidad de dispositivo que son comunicados con USB. Debido a su “reducción en el tiempo de configuración , interoperabilidad entre Pcs portátiles y de escritorio, y la posibilidad de compartir hardware a través de departamentos o laboratorios. Por último, *National Instrument* indica que los diseñadores recurren con más frecuencia a instrumentos de medición o adquisición de datos que se comunican con USB.

2.3.1 El USB

El USB sirve para conectar periféricos como ratones, teclados, escáneres, cámaras digitales, impresoras, discos duros y componentes de red a una computadora personal. Para dispositivos multimedia como escáneres y cámaras digitales, el USB se ha convertido en el método estándar de conexión. Para impresoras, el USB ha crecido tanto en popularidad que ha empezado a desplazar a

²⁰ “USB 2.0 para presidente”. *Instrumentation Newsletter*. Volúmen 17, Número 4, Primer Semestre 2006. Contraportada

los puertos paralelos porque a través del USB es sencillo agregar más de una impresora a una computadora personal.

El USB es un bus externo que provee un estándar de transmisión de datos en serie para conectar dispositivos a computadoras personales, servidores, estaciones de trabajo e incluso a dispositivos portátiles como los teléfonos celulares. Este estándar fue desarrollado a finales de 1996 por ocho empresas: Intel, IBM, Compaq, Northern, Telecom, Microsoft, DEC y NEC.²¹ El sistema USB consiste en un anfitrión y múltiples dispositivos conectados en una estructura de árbol utilizando concentradores especiales. Se pueden conectar hasta 127 dispositivos a un solo servidor, pero la suma debe incluir también a los concentradores, así que el total de dispositivos realmente usables es algo menor.

Este bus está basado en el paso de un testigo (*token*), semejante a otros buses como los de las redes locales en anillo con paso de testigo y las FDDI (*Fiber Distributed Data Interface*). El controlador USB distribuye testigos por el bus. El dispositivo cuya dirección coincide con la que porta el testigo, responde aceptando o enviando datos al controlador. La topología de conexión del USB permite el funcionamiento simultáneo de 127 dispositivos a la vez. En la raíz o vértice de las capas, está el controlador *host* o anfitrión que controla el tráfico que circula por el bus. Esta topología permite a muchos dispositivos conectarse a un único bus lógico sin que los dispositivos que se encuentran abajo, en la pirámide sufran retardo. A diferencia de otras arquitecturas, el USB no es un bus de almacenamiento y envío, de forma que no se produce retardo en el envío de un paquete de datos hacia capas inferiores. Un detalle sorprendente es que cada puerto utiliza una única solicitud de interrupción (IRQ) independientemente de los periféricos que tenga conectados (sea 1 ó 127), por lo tanto no hay riesgo de conflictos entre una cantidad de dispositivos que de otra forma no podrían ser conectados por falta de recursos; de la misma manera tampoco utilizan DMA (asignación de memoria).

El estándar incluye la polarización del dispositivo, conectado a través del mismo cable. Algunos dispositivos requieren una potencia mínima, así que se pueden conectar varios simultáneamente, sin

21 Hyde, J. *USB Design By Example*. Intel Press. E.U.A. 2001, pp. 10-50

necesitar fuentes de alimentación extra. La mayoría de los concentradores incluyen fuentes de alimentación que brindan energía a los dispositivos conectados a ellos, pero algunos dispositivos gastan tanta energía que necesitan su propia fuente de alimentación. Los concentradores con fuente de alimentación pueden proporcionarle corriente eléctrica a otros dispositivos sin quitarle corriente al resto de la conexión (dentro de ciertos límites). Más adelante se explica a detalle la conexión eléctrica.

El estándar USB 1.1 define dos velocidades de transferencia: La “velocidad baja” (*Low Speed*) que corre a 1.5 Mbit/s y que es utilizada para comunicar teclados, ratones y joysticks. La otra es la llamada “velocidad completa” (*Full Speed*) y que corre a 12 Mbit/s. Existe otra velocidad que es por mucho más rápida que las anteriores, sin embargo esta solamente está definida para USB 2.0. Esta velocidad es conocida como “alta velocidad” (*High Speed*) y corre a 480 Mbit/s. Este modo compite directamente con el *Firewire*.

La gran ventaja de este bus, es que la conexión con la computadora se hace de forma dinámica, lo cual quiere decir que el usuario puede conectar el dispositivo a la PC y retirarlo cuando así lo desee sin necesidad de recargar todo el sistema operativo, a diferencia de otros buses como el PCI, AGP y otros, donde es necesario quitar la alimentación de la PC, abrir el gabinete que la contiene, insertar la tarjeta en la ranura correspondiente y posteriormente reiniciar la carga del sistema para que se reconozca y configure los nuevos dispositivos agregados. El proceso dinámico se realiza de la siguiente manera; cuando el usuario conecta el dispositivo a la computadora, esta detecta e interroga al dispositivo y carga el correspondiente controlador. Así, el usuario no tendrá que preocuparse de si consiguió una IRQ, una dirección o de tener que reiniciar a la PC. Lo mismo ocurre cuando el usuario desea retirar el dispositivo; al hacerlo la computadora descarga de inmediato el software de la memoria.

La identificación del hardware del dispositivo USB se lleva a cabo usando dos números; el PID (*Product IDentification*) y el VID (*Vendor IDentification*). Ambos se refieren a las características, tanto del producto como del fabricante de dicho producto.

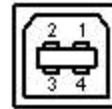
El hardware del USB

Conectores

El cable que conecta el dispositivo USB con la computadora (anfitrión) tiene en ambos extremos dos conectores (ver figura 2-2). El del lado del anfitrión es el tipo A; el que se conecta hacia el dispositivo es el tipo B.



Conector USB tipo A



Conector USB tipo B

Figura 2-2 Conectores tipo A y B

Para evitar el contacto físico entre diferentes pines, los conectores no son mecánicamente intercambiables, por lo que el conector tipo A nunca podrá ser conectado al dispositivo y viceversa. El conector tipo A se instala comúnmente en la tarjeta madre de las computadoras y en concentradores. El conector tipo B siempre está en el dispositivo USB.

En la especificación del USB 2.0 introduce un tercer conector, el mini-USB conector tipo B. Este conector es usado para conectar a dispositivos electrónicos muy pequeños, como son teléfonos celulares, PDA (*Personal Digital Assistant*), dispositivos de audio (*Ipod de Mac*) y cámaras digitales.

Aspectos eléctricos

La transmisión serie del USB utiliza el denominado código de no retorno a cero invertido (NRZI), que representa la información por cambio de niveles. De esta manera, un cambio de nivel se utiliza para representar un “1” lógico y la ausencia de transición representa la asignación del “0”. Estos métodos no ofrecen al receptor un medio para determinar el ritmo con el que el emisor envía los bits, es decir, el ritmo del reloj del emisor.

Por esta razón se prefiere emplear una técnica denominada codificación Manchester; o bien una

variante conocida como codificación diferencial Manchester. Que consiste en una transición de bajo a alto cuando se trata de un “0” lógico y de alto a bajo cuando es un “1” lógico.

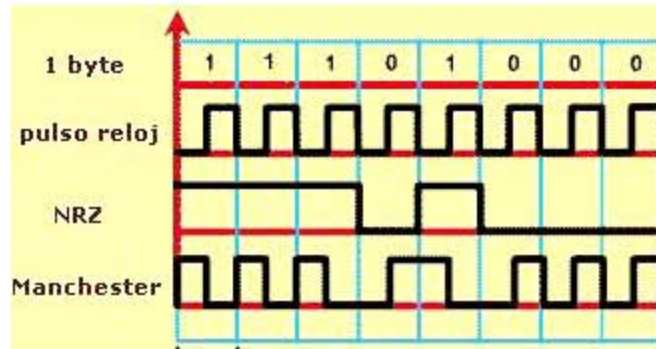


Figura 2-3 Codificación NRZ y Manchester

Para los casos donde el USB está configurado en modo de velocidad alta y completa se colocan las señales D+ y D- (ver tabla 2-6) según lo siguiente: para transmitir un “1” la señal D+ es puesta arriba de 2.8 V con una resistencia de 15 kΩ puesta a tierra y D- se polariza por debajo de 0.3 V con una resistencia de 1.5 kΩ puesto a 3.6 V. Para transmitir un “0”, D- se polariza a más de 2.8 V y D+ a menos de 0.3 V con sus respectivas resistencias de *pull up* o *pull downs*. El receptor toma un “1” cuando D+ es 200 mV más grande que D- y un “0” cuando D+ es 200 mV menos que D-. A estos modos de velocidad, se considera que el bus tiene una impedancia de 90 ohms +/- 15%. A alta velocidad (480 Mbits/s) usa 17.78 mA en la señal para reducir el ruido.

<i>Número de pin</i>	<i>Color del filamento</i>	<i>Función</i>
1	Rojo	V _{BUS} (5 voltios)
2	Blanco	D-
3	Verde	D+
4	Negro	Tierra

Tabla 2-6 Distribución de los pines en un conector USB

Identificación de la velocidad

Para que la computadora identifique a que velocidad transmitirá y recibirá la información, el dispositivo USB tiene que polarizar cualquiera de las líneas D+ o D- a 3.3 V (ver figura 2-3). Para una velocidad completa se coloca una resistencia *pull up* en la línea D+. Estas resistencias ya vienen

integradas al dispositivo y también sirven para detectar la presencia del dispositivo conectado a ese puerto. Sin esa resistencia, el anfitrión asume que no hay ningún dispositivo conectado al bus.

En algunos dispositivos la resistencia *pull up*, es configurable, es decir que desde el *firmware* se selecciona cual de las dos líneas, D+ o D- lleva la resistencia *pull up*.

Potencia eléctrica del USB

El manejo de la corriente es un tema muy delicado que se debe de tomar en cuenta para el diseño de dispositivos conectados por USB. Como el anfitrión proporciona cierta potencia eléctrica al componente USB, es importante configurar la cantidad de corriente que le será suministrada el anfitrión. Si este consume más corriente de la que fue configurada desde un principio, el hardware del anfitrión interrumpe la alimentación. En la configuración también se puede especificar, que la toma de energía eléctrica del dispositivo es de forma externa al anfitrión.

Existen tres formas de configurar un dispositivo USB en cuanto a potencia eléctrica se refiere. Baja potencia (*Low-power*), alta potencia (*High-power*) y auto potencia (*Self-power*). El primero indica que es de bajo consumo de potencia y solo alcanza a suministrar 100 mA; además debe estar diseñado para trabajar con bajo voltaje a 4.40 V y el máximo voltaje es de 5.25 V. La alta potencia puede suministrar el anfitrión al bus, no más de 100 mA hasta que termina de configurarse y después puede drenar hasta 500 mA, como máximo. El auto suministro de potencia es otra posibilidad, siempre y cuando el dispositivo no se desprenda de su fuente externa, el anfitrión le proporcionará 100 mA.

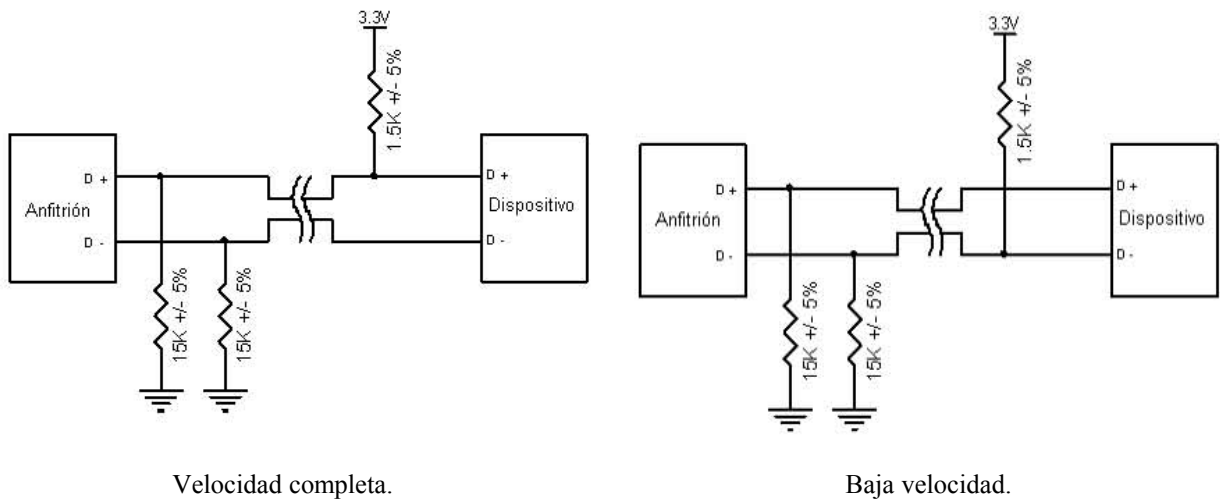


Figura 2-4 Diagrama de resistencias *pull up* para distintas velocidades

Protocolos del USB

A diferencia de la interfaz RS-232 donde el formato de los datos y los comandos están determinados por el diseñador, el USB tiene un formato formal distribuido en distintas capas de protocolos. Cada transacción en el USB consiste en paquetes de paso de testigo (*token*), paquetes de datos y paquetes de estado. Al momento de conectar el dispositivo USB al anfitrión, este envía el primer paquete *token* que describe las pautas de inicialización de los dispositivos e indica si los datos van a ser enviados o recibidos así como la dirección del *endpoint* (nombre dado al canal donde serán depositados o bien desde donde se envían los datos dentro de un dispositivo USB) que está llamando su atención. El siguiente paquete generalmente de datos, lleva consigo la información y es seguido por un paquete de confirmación (*Handshake*), reporta además si el dato o el *token* ha sido recibido correctamente o no esta disponible.

Estructura de un paquete común

Los datos transmitidos empiezan por el bit menos significativo y tienen la siguiente estructura.

- Campo “Sync”: Esta es una señal de sincronización de reloj. Todos los paquetes deben empezar con este campo. La longitud de “Sync” es de 8 bits en baja velocidad y de 32

bits en completa y alta velocidad.

- Campo “PID”: Es el número que identifica al paquete. En la tabla 2-7 se muestra los posibles valores.

<i>Tipo</i>	<i>Valor PID</i>	<i>Identificador de paquete</i>
<i>Token</i>	0001	<i>Token de salida</i>
	1001	<i>Token de entrada</i>
	0101	<i>SOF Token</i>
	1101	<i>Token de configuración</i>
Dato	0011	Dato 0
	1011	Dato 1
	0111	Dato 2
	1111	M Dato
Confirmación (Handshake)	0010	ACK
	1010	NAK
	1110	STALL
	0110	NYET
Especial	1100	PREamble
	1000	ERR
	1000	Split
	0100	Ping

Tabla 2-7 Valores de PID

- Campo “ADDR”: Este campo especifica la dirección del dispositivo a quien va designado. La longitud es de 7 bits, es decir 127 posibles combinaciones que equivalen al número de dispositivos por anfitrión. El valor “0” no es válido, ya que cualquier dispositivo que no tiene asignada una dirección, puede responder a esta dirección.
- Campo “ENDP”: El campo de *endpoint* esta compuesto por 4 bits, permitiendo 16 posibles *endpoints*. Para el modo de baja velocidad pueden ser 4 *endpoints* como máximo.

- Campo “CRC”: *Cyclic Redundancy Checks*; todos los paquetes están compuestos de 5 bits CRC, mientras que un paquete de datos tiene 16 bits de CRC.
- Campo “EOP”: *End of Packet*, final del paquete).

Tipos de paquetes

Existen cuatro tipos de paquetes. Los paquetes *token* indican el tipo de transacción a realizar, los paquetes de datos contienen la información, los paquetes de confirmación son usados para reportar errores o confirmaciones de transmisión y por último están los paquetes de inicio de marco (*frame packets*) que indican el inicio de un nuevo marco.

- Paquetes *token*: se divide en tres tipos.
 - ✓ *In*. El anfitrión informa al dispositivo que requiere leer información.
 - ✓ *Out*. El anfitrión informa al dispositivo que requiere enviar información.
 - ✓ *Setup*. Usado para empezar el control de las transferencias.

Formato de un paquete *token*.

Sync	PID	ADDR	ENDP	CRC5	EOP
------	-----	------	------	------	-----

- Paquetes de datos: Hay dos tipos. Cada uno con capacidad de 1024 bytes.
 - ✓ Baja velocidad. El tamaño de los datos es de 8 bytes.
 - ✓ Velocidad completa El tamaño de los datos es de 1023 bytes.
 - ✓ Alta velocidad. El tamaño de los datos es de 1024 bytes.

Formato de un paquete de datos.

Sync	PID	Data	CRC1	EOP
			6	

- Paquetes de confirmación: Hay tres tipos de paquetes de confirmación.
 - ✓ *ACK*. Confirmación de que un paquete ha sido recibido correctamente.
 - ✓ *NAK*. Indica que un dispositivo no puede enviar ni recibir datos temporalmente. También

se usa durante las transacciones de interrupción, para informar al anfitrión que no hay datos para enviar.

- ✓ *STALL*. Indica que el dispositivo encontrado requiere de intervención por parte del anfitrión.

Formato de un paquete de confirmación.

Sync	PID	EOP
------	-----	-----

- Paquetes de inicio de marco: Este paquete es de 11 bits.

Formato de inicio de marco.

Sync	PID	Frame Number	CRC5	EOP
------	-----	--------------	------	-----

Endpoints

La forma de colocar un paquete de datos, *token* o de confirmación en el bus no es trivial como en el caso del bus de datos en la tarjeta madre. Es decir, no basta con copiar el dato desde el dispositivo al bus, pues hay que recordar que hay que pedir permiso al anfitrión y una vez que este lo conceda, se procede a realizar dicho trabajo. Pero el dispositivo no puede estar a la espera de que el anfitrión le atienda. Ya que esto resultaría poco práctico y causaría retardos que sin importar la extraordinaria velocidad de transmisión del USB, el dispositivo permanecería detenido la mayor parte del tiempo. De esta manera es necesario que exista un lugar físico donde los datos arrojados de un proceso que realiza la aplicación USB o bien los datos provenientes desde el anfitrión, y que en ese momento no pueden ser atendidos por el dispositivo, permanezcan intactos hasta se atendidos por el anfitrión o el dispositivo respectivamente. El término *endpoint* se refiere a un punto final a donde llegan los datos o de donde salen. Se puede pensar en un *buffer*, este es un dispositivo físico que guarda un dato un determinado tiempo. El lugar donde termina un dato enviado por la computadora personal al dispositivo USB se denomina EP1 OUT, y cuando un dato quiere ser enviado desde el dispositivo al anfitrión se denomina EP1 IN.

Un *endpoint* también puede ser visto como una interfaz entre el hardware del dispositivo USB y el

firmware que corre en dicho dispositivo. Este *endpoint* está integrado físicamente al dispositivo USB

Pipes

Mientras el dispositivo envía y recibe datos desde una serie de *endpoints*, el software transfiere datos a través de tuberías (*pipes*). Una *pipe* es una conexión lógica entre el anfitrión y los *endpoints*. Estas también tienen parámetros asociados con el ancho de banda, el tipo de transferencia, la dirección de los datos y el máximo tamaño de paquetes.

Tipos de Endpoints

Existen cuatro formas de transferencia especificadas en el estándar USB, transferencias por control (*Control Transfer*), transferencias por interrupción (*Interrupt Transfer*), transferencias isócronas (*Isochronous Transfer*) y transferencias masivas (*Bulk Transfer*).

- Transferencias por control

Este modo de transferencia es usado para enviar comandos y recibir datos de estado desde el dispositivo. La longitud de un paquete en el modo de transferencia por control en baja velocidad es de 8 bytes, para alta velocidad es de 8, 16, 32 o 64 bytes y para velocidad completa es de 64 bytes.

- Transferencias por interrupción

Generalmente cuando se desarrollan aplicaciones que intercambian información con dispositivos externos por ejemplo el teclado, tarjeta de sonido, tarjetas de adquisición de datos y otros necesariamente tienen que hacerle saber al procesador central que un evento ha ocurrido (interrupción) y que es necesario que lo atienda, ya sea para leer datos o escribirlos. En el caso del USB esto no ocurre así. Si un dispositivo USB requiere de la atención del anfitrión, tendrá que esperar a que el anfitrión revise su estado. Es decir que el anfitrión, en el modo de transferencias por

interrupción, continuamente revisa cada *endpoint* y si este solicita una interrupción, entonces y hasta ese momento se concede la interrupción. Una interrupción requerida por el dispositivo USB es puesta en espera en el dispositivo hasta que el anfitrión lo revise.

El tamaño máximo de datos transferidos en este modo para baja velocidad es de 8 bytes, para velocidad completa es de 64 bytes y 1024 bytes para alta velocidad.

El proceso de transferencia se realiza de dos formas, una de entrada y otra de salida (ver figura 2-5).

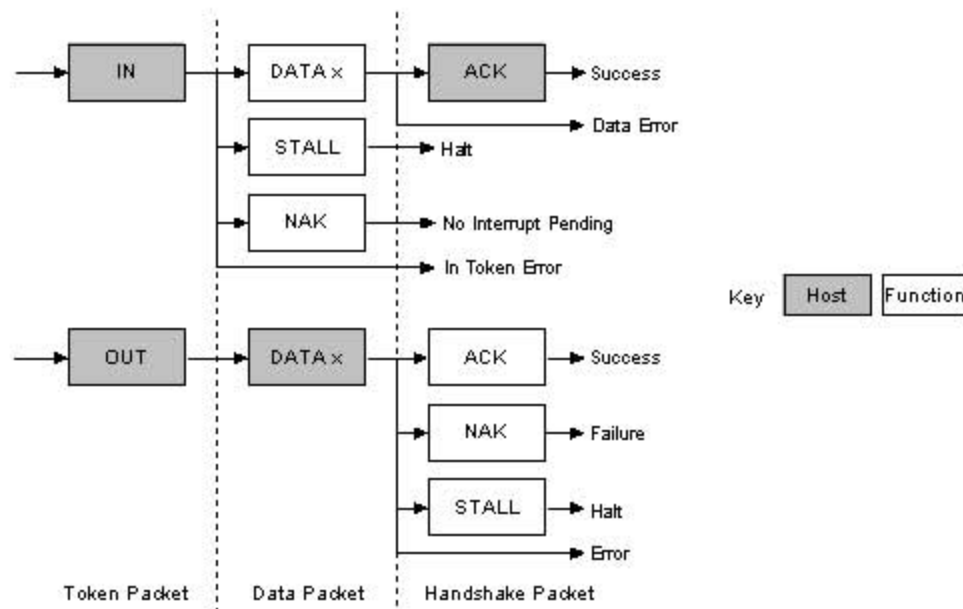


Figura 2-5 Esquema del formato de una interrupción de entrada (IN) y una interrupción de salida (OUT)

- ✓ Proceso de transferencia de entrada (IN): El anfitrión periódicamente revisa si hay interrupciones en el *endpoint* y envía un *IN Token*. El tiempo de poleo se establece en el descriptor del *endpoint*.

Si este paquete es corrupto, el dispositivo ignora el paquete y continua supervisando el bus para nuevos *tokens*. Si el dispositivo colocó una petición de interrupción y el anfitrión envía un *IN token*, el *firmware* del dispositivo envía el paquete con el dato que

quiere transmitir. Entonces el anfitrión envía un paquete de confirmación (ACK). Pero si el dato enviado por el dispositivo es corrupto el anfitrión no envía ninguna respuesta.

Si no ha sido solicitada ninguna interrupción, el anfitrión envía un paquete NAK al bus advirtiéndole que no hay interrupciones pendientes. En caso de que ocurra un error en el *endpoint*, este envía un paquete STALL solicitando que vuelva a enviarse otro *IN Token*.

- ✓ Proceso de transferencia de salida (OUT): Cuando el anfitrión quiere enviar un dato, lo envía precedido de un *OUT Token*. Si ambos paquetes están corruptos el dispositivo los ignora. Si el *endpoint* está vacío, el propio *endpoint* generará un paquete con la señal ACK informando al anfitrión que el envío fue exitoso. En caso contrario, si el *endpoint* está lleno este envía una señal al anfitrión de NAK. Si ocurre un error en el *endpoint*, este envía una señal STALL.

- Transferencias isócronas

Este modo de transferencia es ideal para transmitir audio o video, ya que el envío lo hace de forma continua o periódica. Esto es así por que cualquier retardo en la señal provocado por el anfitrión o por el mismo dispositivo, generaría una discontinuidad y esto afectaría el resultado esperado en una señal de audio o video. En este modo se pueden enviar paquetes de datos de hasta 1023 bytes para velocidad completa y 1024 para alta velocidad.

- Transferencias masivas

Con este modo se pueden transmitir gran cantidad de datos, que pueden ser necesarios para enviar información a la impresora o recibir información de una imagen explorada por un escáner. Con este modo es posible detectar errores y corregirlos, así se asegura la transmisión. En este modo solo puede transmitirse a alta velocidad y a velocidad completa.

Descriptores del USB

Todos los dispositivos USB tienen una jerarquía de descriptores que le proporcionan información al anfitrión; por ejemplo de que dispositivo se trata, quien es el fabricante, que versión de firmware está ejecutando, las formas en que puede ser configurado y el número de *endpoints*. Hay descriptores de dispositivo, configuración, interfaz, *endpoint* y por último el de texto (ver figura 2-6).

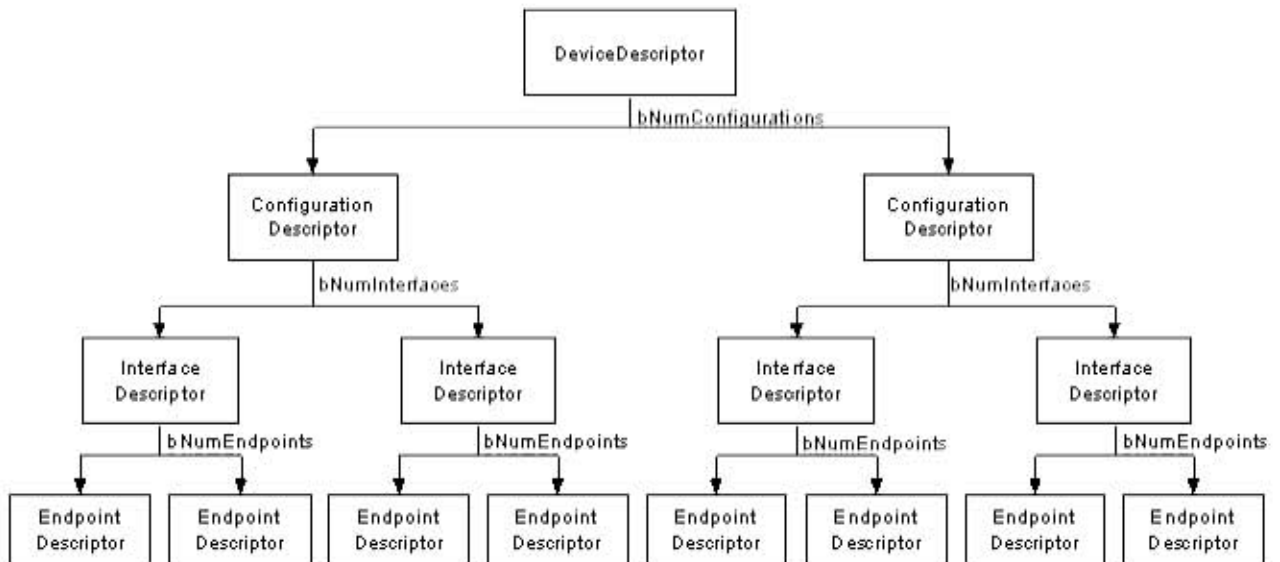


Figura 2-6. Esquema de de los descriptores del USB

- Composición de cada descriptor de un dispositivo USB

Todos los descriptores tienen el mismo formato. El primer byte especifica la longitud del descriptor y el segundo byte el tipo de descriptor. Para comprobar la veracidad de esta información y que esta no este viciada con ruido electromagnético, el anfitrión hace algunas pruebas de autenticidad. Si la longitud del descriptor es más pequeña que la definida, entonces el anfitrión ignora esta información y continua su proceso. En cambio si la longitud del descriptor es más grande de lo que espera el anfitrión, este elimina el número de bytes extras y comienza a leer el siguiente descriptor.

- Descriptor de dispositivo

Para que la computadora anfitrión del USB tenga conocimiento de que clase de dispositivo USB se está conectando, es necesario que el *firmware* del dispositivo describa sus características para poder cargar en memoria el controlador designado a ese dispositivo USB. Estas características se refieren

al nombre del fabricante, nombre del producto así como la versión de desarrollo. Cada dispositivo USB tiene solo un descriptor. El formato de este descriptor se muestra en la tabla 2-8.

<i>Offset</i>	<i>Campo</i>	<i>Tamaño</i>	<i>Valor</i>	<i>Descripción</i>
0	bLenght	1	Número	Tamaño del descriptor en bytes (18 bytes)
1	bDescriptorType	1	Constante	Descriptor de dispositivo (0x01)
2	bcdUSB	2	BCD	Número de versión de USB
4	bDeviceClass	1	Clase	Código de clase, asignado por la Organización USB
5	bDeviceSubClass	1	Subclase	Código de subclase, asignado por la Organización USB
6	bDeviceProtocol	1	Protocolo	Código de protocolo, asignado por la Organización USB
7	bMaxPacketSize	1	Número	Máximo tamaño de paquete
8	idVendor	2	ID	Identificador del fabricante, asignado por la Organización USB
10	idProduct	2	ID	Identificador del producto, asignado por la Organización USB
12	bcdDevice	2	BCD	Número de versión
14	iManufacturer	1	Índice	Nombre del fabricante
15	iProduct	1	Índice	Nombre del producto
16	iSerialNumber	1	Índice	Número de serie
17	bNumConfigurations	1	Entero	Número de posibles configuraciones

Tabla 2-8. Formato de descriptor de dispositivo

- Descriptor de configuración

Para especificar las características de configuración de un dispositivo USB está el descriptor de configuración. Este descriptor indica ciertos aspectos referentes a la ponencia eléctrica que suministra el hardware del anfitrión al hardware de dispositivo USB. El descriptor de configuración proporciona datos como la cantidad de potencia que requiere el dispositivo y si este la obtendrá de forma externa.

Cuando un dispositivo se en lista, al inicio de las transacciones, el anfitrión lee el descriptor del dispositivo y toma una decisión de cual configuración habilitar, solo una puede ser habilitada. Cada

dispositivo puede tener más de un descriptor de configuración. Una vez que todas las configuraciones han sido leídas por el anfitrión, este envía un comando de configuración establecida (*SetConfiguration*) con un valor diferente de cero que corresponde al número de configuración (*bConfigurationValue*), así se selecciona la configuración indicada. El formato de este descriptor se muestra en la tabla 2-9.

<i>Offset</i>	<i>Campo</i>	<i>Tamaño</i>	<i>Valor</i>	<i>Descripción</i>
0	bLenght	1	Número	Tamaño del descriptor en bytes
1	bDescriptorType	1	Constante	Descriptor de configuración (0x02)
2	wTotalLenght	2	Número	Tamaño total de la longitud de los datos regresados
4	bNumInterfaces	1	Número	Número de interfaces
5	bConfigurationValue	1	Número	Valor usado como argumento para seleccionar esta configuración
6	iConfiguration	1	Índice	Índice del descriptor de texto que describe a esta configuración
7	bmAttributes	1	Bitmap	D7 – USB 1.0 D6 – Auto alimentación D5 – Inicio remoto D4..0 – Ceros
8	bMaxPower	1	mA	Máxima corriente en unidades de 2 mA

Tabla 2-9. Formato de descriptor de configuración

- Descriptor de interfaz

Existe otro descriptor que agrupa las posibles interfaces en un dispositivo USB. Este se llama descriptor de interfaz. Su función es la de agrupar *endpoints*. Por ejemplo, en el caso de un dispositivo multifuncional que consiste en un fax, escáner y una impresora que estén conectados a través de un USB a una computadora personal, cada componente puede ser descrito con un descriptor de interfaz. A diferencia del descriptor de configuración, el número de descriptores de interfaz activos pueden ser más de uno y trabajar al mismo tiempo. El formato de este descriptor se muestra en la tabla 2-10.

<i>Offset</i>	<i>Campo</i>	<i>Tamaño</i>	<i>Valor</i>	<i>Descripción</i>
0	bLenght	1	Número	Tamaño del descriptor en bytes (9 bytes)
1	bDescriptorType	1	Constante	Descriptor de interfaz (0x04)
2	bInterfaceNumber	1	Número	Número de interfaz
3	bAlternateSetting	1	Número	Valor usado para seleccionar una configuración alterna
4	bNumEndpoints	1	Número	Número de <i>endpoints</i> usados en la interfaz
5	bInterfaceClass	1	Clase	Código de clase, asignado por la Organización USB
6	bInterfaceSubClass	1	Subclase	Código de subclase, asignado por la Organización USB
7	bInterfaceProtocol	1	Protocolo	Código de protocolo, asignado por la Organización USB
8	iInterface	1	Índice	Índice del descriptor de texto que describe a esta interfaz

Tabla 2-10. Formato de descriptor de interfaz

<i>Offset</i>	<i>Campo</i>	<i>Tamaño</i>	<i>Valor</i>	<i>Descripción</i>
0	bLenght	1	Número	Tamaño del descriptor en bytes (9 bytes)
1	bDescriptorType	1	Constante	Descriptor de <i>endpoint</i> (0x05)
2	bEndpointAddress	1	Endpoint	Dirección de <i>endpoint</i>
3	bmAttributes	1	Bitmap	Descripción en la tabla 2-12
4	wMaxPacketSize	2	Número	Tamaño máximo de paquete en el que es capaz de transmitir y recibir el <i>endpoint</i>
6	bInterval	1	Número	Tiempo que tarda el anfitrión en atender cada <i>endpoint</i>

Tabla 2-11. Formato de descriptor de *endpoint*

- Descriptor de *endpoint*

Este descriptor describe el funcionamiento de cada *endpoint*, a excepción del *endpoint* cero. El *endpoint* cero siempre asume que es un *endpoint* de control y es configurado antes de cualquier descriptor. El anfitrión solicita esta información para determinar el ancho de banda que requiere el bus. El formato de este descriptor se muestra en la tabla 2-11.

Bits	1	0	Tipo de transferencia
	0	0	Control
	0	1	Isócrono
	1	0	<i>Bulk</i>
	1	1	Interrupción
Bits	3	2	Tipo de sincronización (solo isócrono)
	0	0	Sin sincronización
	0	1	Asíncrono
	1	0	Adaptable
	1	1	Síncrono
Bits	5	4	Tipo de uso (solo isócrono)
	0	0	Dato <i>endpoint</i>
	0	1	Retroalimentación <i>endpoint</i>
	1	0	Retroalimentación explícita <i>endpoint</i>
	1	1	Reservado

Tabla 2-12 Formato del campo “bmAttributes” descriptor del *endpoint*

Offset	Campo	Tamaño	Valor	Descripción
0	bLength	1	Número	Tamaño del descriptor en bytes
1	bDescriptorType	1	Constante	Descriptor de texto (0x03)
2	WLANGID[0]	2	Número	Lenguaje soportado. Código cero. (Por ejemplo) 0x0409 Inglés – EUA
4	WLANGID[1]	2	Número	Lenguaje soportado. Código uno. (Por ejemplo) 0x0c09 Inglés – Australia
n	WLANGID[x]	2	Número	Lenguaje soportado. Código x. (Por ejemplo) 0x0407 Alemán - Estándar
	bString	n	Unicode	Texto codificado en Unicode

Tabla 2-13 Formato de descriptor de texto

- Descriptor de texto

El descriptor de texto proporciona información comprensible para el humano, es decir que a través de este descriptor, el firmware le da a conocer información al anfitrión y este al usuario, referente al nombre del producto, quien lo desarrollo, la versión y otros. Este descriptor está codificado en el formato “Unicode”. El formato de este descriptor se muestra en la tabla 2-13.

2.3.2 Dispositivos USB enfocados al desarrollo

El propósito de incorporar un dispositivo USB a la sonda es el de contar con un medio de comunicación con los sensores y los motores. Para esto es necesario que el dispositivo USB haga llegar esta información de alguna forma, proveniente de la computadora de control, hasta los motores o simplemente capturar los datos proporcionados por los sensores y enviarlos a la computadora de control. Sin embargo esta tarea es bastante compleja.

Como se vio en la sección anterior la información en el USB se transmite en serie. Pero los sensores y motores de la sonda realizan sus funciones con datos digitales en paralelo e incluso usan señales analógicas. De tal manera que el dispositivo USB tiene que convertir los datos provenientes de la computadora de control de formato USB a paralelo, los datos de los sensores digitales tanto paralelos como serie los tiene que convertir al formato USB, además de que en algunos casos tendrá que convertir una señal analógica en una señal digital y de ahí a formato USB.

Es por eso que es necesario diseñar una tarjeta que contenga al microcontrolador o un dispositivo que soporte el USB y que además pueda multiplexar los datos entre los distintos motores o sensores. De esta manera se piensa en la idea de un bus de datos que conecte cada motor o sensor con el dispositivo USB. Sin embargo, por cuestiones electrónicas los dispositivos no pueden conectarse directamente con el microcontrolador. Es por eso que se hace uso de *buffers* o transceptores. Además se necesita alguna forma de controlar los tiempos de lectura y escritura, así como la de indicarle a cada *buffer* y transceptor que le toca transmitir. Pero esto se discutirá en el siguiente capítulo.

<i>Dispositivo</i>	<i>Fabricante</i>	<i>USB</i>	<i>Características</i>
CY7C63001	Cypress Semiconductor	<ul style="list-style-type: none"> • Baja velocidad 1.5 Mbps • 1 Control • 1 Endpoint 	<ul style="list-style-type: none"> • Microcontrolador de 8 bits RISC • 6 MHz • 128 kbytes RAM • 2 kbytes EPROM
EZ-USB (AN21XX)	Cypress Semiconductor	<ul style="list-style-type: none"> • Velocidad completa 12 Mbps • Re-enumeración dinámica 	<ul style="list-style-type: none"> • Enhanced 8051 core • 612MHz
EZ-USB (CY7C646xx)	Cypress Semiconductor	<ul style="list-style-type: none"> • Velocidad completa 12 Mbps • Re-enumeración dinámica • 32 endpoints 	<ul style="list-style-type: none"> • Enhanced 8051 core • 2 USARTS • 3 Timers/counters
PIC16C745	Microchip	<ul style="list-style-type: none"> • Baja velocidad 1 Mbps • 6 Endpoints 	<ul style="list-style-type: none"> • Microcontrolador de 8 bits RISC • 8 kbytes EPROM • 20 MHz • Convertidores A/D de 8 bits • USART/SCI • OTP • 28 pines
PIC16C765	Microchip	<ul style="list-style-type: none"> • Baja velocidad 1 Mbps • 6 Endpoints 	<ul style="list-style-type: none"> • Microcontrolador de 8 bits RISC • 8 kbytes EPROM • 20 MHz • Convertidores A/D de 8 bits • USART/SCI • OTP • 40 pines
PIC18F2X50	Microchip	<ul style="list-style-type: none"> • Velocidad completa 12 Mbps • 16 Endpoints 	<ul style="list-style-type: none"> • Microcontrolador de 8 bits RISC • 32 kbytes Flash • 20 MHz • Convertidores A/D de 10 bits • USART/SCI • OTP • 28 pines
PIC18F4X50	Microchip	<ul style="list-style-type: none"> • Velocidad completa 12 Mbps • 16 Endpoints 	<ul style="list-style-type: none"> • Microcontrolador de 8 bits RISC • 32 kbytes Flash • 20 MHz • Convertidores A/D de 10 bits • USART/SCI • OTP • 40 pines
NET2888	NetChip Technology	<ul style="list-style-type: none"> • Compatible con versión USB 1 y 1.1 • Transferencia por control y 4 endpoints modificados ✓ EP1 – 8 bytes BULK OUT ✓ EP2 – 8 bytes INT IN ✓ EP1 – 64 bytes BULK/ISO OUT ✓ EP1 – 64 bytes BULK/ISO IN 	<ul style="list-style-type: none"> • 48 Mhz • 48 pines
NET2890	NetChip Technology	<ul style="list-style-type: none"> • Compatible con versión USB 1 y 1.1 • Transferencia por control y 4 endpoints configurables 	<ul style="list-style-type: none"> • 48 Mhz • 48 pines
68HC705JB3	Motorola	<ul style="list-style-type: none"> • Baja velocidad 1.5 Mbps • 1 endpoint de transferencia por control • 1 endpoint de transferencia por interrupción 	<ul style="list-style-type: none"> • Basado en el HC05 • 2560 bytes de ROM • 144 bytes de RAM • 20 pines

Dispositivo	Fabricante	USB	Características
68HC705JB4	Motorola	<ul style="list-style-type: none"> • Baja velocidad 1.5 Mbps • 1 endpoint de transferencia por control • 1 endpoint de transferencia por interrupción 	<ul style="list-style-type: none"> • Basado en el HC05 • 3584 bytes de ROM • 176 bytes de RAM • 28 pines • 6 canales de conversión A/D de 8 bits de resolución
PDIUSB11	Phillips Semiconductor	<ul style="list-style-type: none"> • Velocidad completa 12 Mbps • 1 endpoint de transferencia por control • 6 endpoint de configurables 	<ul style="list-style-type: none"> • USB a I²C • 12 Mhz • 16 pines
PDIUSB12	Phillips Semiconductor	<ul style="list-style-type: none"> • Alta velocidad 2 MBps • 1 endpoint de transferencia por control • 4 endpoint de configurables 	<ul style="list-style-type: none"> • USB a puerto paralelo • DMA • 28 pines
FT8U232AM	FTDI	<ul style="list-style-type: none"> • Velocidad configurables • Modo de transferencia configurable 	<ul style="list-style-type: none"> • USB a RS232/RS422 • Salidas y niveles compatibles con RS232 • Reloj 6 Mhz • Velocidad RS232 2000 kbps • 32 pines
FT8U245AM	FTDI	<ul style="list-style-type: none"> • Velocidad configurables • Modo de transferencia configurable 	<ul style="list-style-type: none"> • USB a 8 bits en paralelo • Salidas y niveles compatibles con IEEE1024 • Reloj 6 Mhz • 32 pines
IOWarriorIOWarrior 24	IOWarrior	<ul style="list-style-type: none"> • Baja velocidad • Compatible con USB 1.2 y 2.0 	<ul style="list-style-type: none"> • USB a I²C/SPI • Compatible con módulos LCD • Soporte de Mac, Linux y Windows • 24 pines
USBN9602	National Semiconductor	<ul style="list-style-type: none"> • 1 Endpoint de transferencia por control • Alta velocidad 	<ul style="list-style-type: none"> • 8 bits paralelos • 48 Mhz • 28 pines

Tabla 2-14. Dispositivos USB

Por ahora basta mencionar los distintos dispositivos que portan un módulo USB. Algunos de ellos son microcontroladores que tienen integrado uno o dos módulos USB en su estructura. Estos dispositivos, aparte del hardware para controlar a nivel físico las señales del bus, el fabricante proporciona *firmware*, librerías y archivos complementarios para desarrollar aplicaciones que controlen al USB. Otros dispositivos convierten la señal USB al formato RS-232 o a un formato de bits en paralelo. En este caso se necesita de un sistema que configure a los descriptores, controladores, etc. para poder hacer funcionar el protocolo. Una comparativa entre los distintos dispositivos USB se muestra en la tabla 2-14.

2.3.3 El PIC16C745/65

En la tabla 2-14 se presentan varios dispositivos con soporte USB. Sin embargo esta tesis basa el diseño de la Tarjeta de Expansión de Puertos (TEP) de datos en el PIC16C745 por razones de disponibilidad, versatilidad y económicas. El uso de este dispositivo para la TEP garantiza que debido a las facilidades técnicas para su programación física (programador de PIC's), a la sencillez de sus instrucciones de programación, a la disponibilidad de ambientes de desarrollo de programación en el departamento, al conocimiento y experiencia previa del funcionamiento de esta familia de microcontroladores y por la manera de adquirirlos a buen precio y buen tiempo en nuestro país.

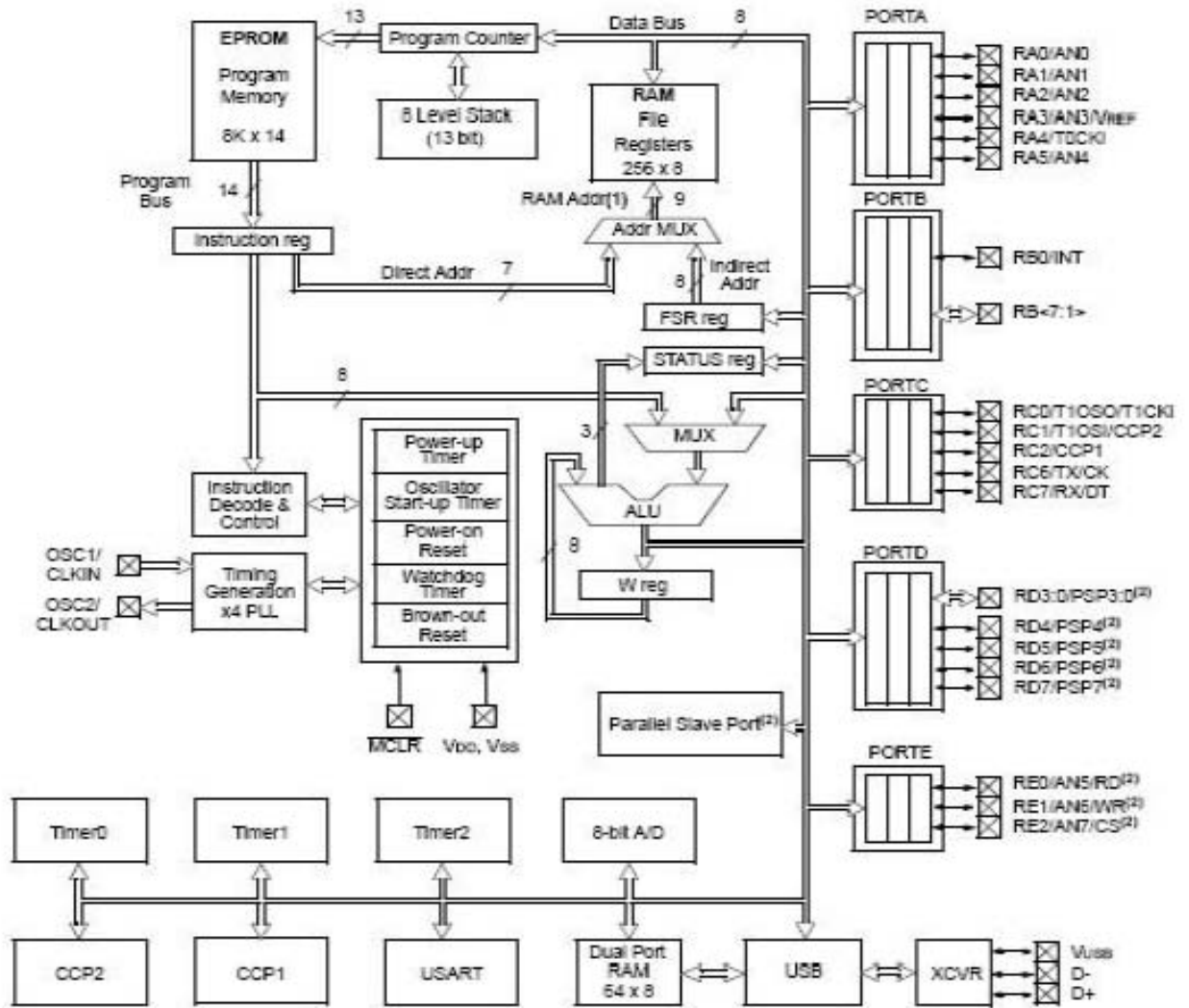
El PIC16C745 pertenece a la familia PIC16C7X5. Son dispositivos de bajo costo, alto desempeño, y tecnología CMOS. Su arquitectura es del tipo RISC (*Reduced Instruction Set Computer*) de 8 bits y son considerados de mediano rango en la familia de microcontroladores de Microchip PIC16CXX.

Cuenta con 8 niveles de pila y varias fuentes de interrupción tanto externas como internas. Las instrucciones son de 14 bits y el manejo de datos es de 8 bits. La tecnología *pipeline* le permite al microcontrolador ejecutar cualquier instrucción en un solo ciclo de reloj a excepción de las instrucciones de condición, que se ejecutan en dos ciclos. Esta familia de microcontroladores tiene 35 instrucciones.

Además tiene 22 pines de entrada/salida. Con 256 bytes de RAM. En la parte de periféricos, cuenta con tres *timers/counters*, 2 módulos de captura/comparador/PWM y 2 puertos serie. La característica principal de este dispositivo es que está provisto de un sistema de comunicación USB a baja velocidad. Otra característica muy importante es que cuenta con un convertidor analógico/digital de 5 canales de alta velocidad y resolución de 8 bits.

Tiene 4 formas para configurar el oscilador. La forma EC utiliza una señal de reloj externa, E4 además de utilizar una señal de reloj externa se vale de la ayuda de un PLL (*Phase Locked-Loop*) para definir su velocidad, HS es una forma para operar el microcontrolador a alta velocidad y que es

excitado con un cristal/resonador y por último la forma H4 también es usada para la alta velocidad y que es excitado con un cristal/resonador con ayuda de un PLL.



Notas: 1) Bits de orden alto son del registro STATUS
 2) No disponibles para el PIC16C745

Figura 2-7 Diagrama a bloques del PIC16C745/765

Además tiene la capacidad de entrar en modo “suspendido” (*sleep*) el cual permite utilizar el mínimo de recursos al grado de aparentar estar apagado. Pero con interrupciones externa, internas o reiniciar el dispositivo, éste “despierta” para seguir funcionando normalmente. Para evitar bloqueos de software, el PIC16C745 tiene un módulo llamado “*watchdog*” (perro guardián).

Este microcontrolador tiene dos tipos de empaquetado. El llamado Cerdip que cuenta con una ventana en la parte superior para aplicar rayos ultravioleta (UV) y reiniciar las casillas de memoria para poder grabar otro programa. El otro empaquetado es el llamado OTP (*One Time Programmable*), que una vez que se programa el circuito es imposible borrarlo.

2.4 Comunicación entre la SRM y la estación de control

Uno de los objetivos planteados, desde que se concibió la idea de la sonda robot, es que el operador pueda manipular la sonda a distancia de forma inalámbrica. Es por eso que debe de existir una forma de comunicación entre la SRM y la estación de control.

En el diseño original de la sonda robot se planteó un canal de comunicación alámbrico, es decir, corría

un cable de varios hilos desde la sonda hasta la estación de control. Esto representaba un verdadero problema y es el hecho de que la distancia máxima entre la sonda y la estación de control depende tanto de lo largo de dicho cable, como de la atenuación que sufre la señal eléctrica (RS-232) a largas distancias. Sin lugar a dudas esto limita el alcance de la sonda y la hace en extremo ineficiente. Por tal motivo fue necesario marcar otro rumbo distinto en el tema de comunicación; por lo que se decidió utilizar la comunicación inalámbrica.

La tecnología inalámbrica ha propiciado el crecimiento de las comunicaciones. Ejemplo de ellos es el crecimiento estrepitoso de la telefonía celular. Por lo que la tendencia es que los dispositivos electrónicos se comuniquen entre si por medios inalámbricos. Esto se debe a que la comunicación inalámbrica representa menores costos de instalación, mayor movilidad y sobre todo mayor comodidad. Es por eso que la sonda robot móvil cuenta con un enlace con la estación terrena de forma inalámbrica.

<i>Equipo</i>	<i>Máx. Dist. Int/Ext [m]</i>	<i>Vel. Trans. [Kbps]</i>	<i>Proto -colo</i>	<i>Frec [Ghz]</i>	<i>Interco-nexión</i>	<i>Ventajas</i>	<i>Desventajas</i>
XBee-Pro	100 / 1,200	250	Propio	2.4	Serie	<ul style="list-style-type: none"> • Bajo costo • Bajo consumo de potencia 	<ul style="list-style-type: none"> • Programar protocolos de comunicación • Baja velocidad de transferencia • Uninodal • Poco alcance
Xbee-Pro PKG-U	100 / 1,200	250	Propio	2.4	USB a Serie	<ul style="list-style-type: none"> • Bajo consumo de potencia • Dispositivo USB 	<ul style="list-style-type: none"> • Programar protocolos de comunicación • Baja velocidad de transferencia • Uninodal • Poco alcance • Alto costo
9XTend-PKG-R	900 / 64,000	115.2	Propio	0.9	Serie	<ul style="list-style-type: none"> • Bajo consumo de potencia • Gran alcance 	<ul style="list-style-type: none"> • Programar protocolos de comunicación • Baja velocidad de transferencia • Uninodal • Alto costo
9XTend-PKG-U	900 / 64,000	115.2	Propio	0.9	USB	<ul style="list-style-type: none"> • Bajo consumo de potencia • Gran alcance • Dispositivo USB 	<ul style="list-style-type: none"> • Programar protocolos de comunicación • Baja velocidad de transferencia • Uninodal • Alto costo
WL-167G	40 / 360	54,000	IEEE802.11g	2.4	USB	<ul style="list-style-type: none"> • Bajo consumo de potencia • Dispositivo USB • Estándar IEEE802.11g • Tecnología WiFi • Multinodal • Bajo costo • Alta velocidad de transferencia 	<ul style="list-style-type: none"> • Poco alcance

Tabla 2-15 Tabla de módulos de comunicación inalámbrica.

Debido a la gran variedad de tecnologías disponibles en el mercado de dispositivos que realizan la comunicación inalámbrica, fue necesario hacer una selección de las distintas opciones. En la tabla 2-15 se presentan algunos dispositivos inalámbricos con sus características, ventajas y desventajas.

Dado que se busca una tecnología que permita conectar la sonda con la estación remota de una manera estable, segura, continua, rápida y económica se puede determinar que la opción más viable para la comunicación inalámbrica es un adaptador de red inalámbrica de tecnología WiFi (*Wireless Fidelity*). De modo que la solución adoptada para el mejoramiento de la SRM fue optar por la opción del equipo WL-167G con adaptador USB.

2.4.1 Estándar WiFi

Las redes inalámbricas presentan una opción muy favorable para las empresas, gobiernos y ahora en ámbitos domésticos. Ya que con el uso de estas tecnologías los cables son cosa del pasado y se evitan todos los inconvenientes que implica instalar una red con cables. Sin embargo las redes sin cables tienen poco tiempo explorando mercados, e instalar una red casera requiere de una inversión moderada que podría obligar a optar por instalar una alámbrica. Existen varios tipos de redes inalámbricas.

- **Redes inalámbricas personales**

Son redes de poca potencia que conectan dispositivos electrónicos entre sí, para establecer comunicación y transferir información. Ejemplo de ello son el puerto Infrarrojo IR, *Bluetooth* y en general las redes creadas por los usuarios que contienen sus propios estándares y protocolos. Estas redes sirven para enviar archivos entre computadoras, PDAs, Pcs o teléfonos celulares además, es posible enviar a imprimir a una impresora que soporte dicho protocolo. Entre sus desventajas para ser utilizada en esta actualización, es la escasa velocidad de transmisión y baja potencia de disipación que provoca poco alcance de comunicación entre los dispositivos.

- **Redes inalámbricas de consumo**

Son redes ofrecidas por las compañías que ofrecen servicios de telefonía celular. Entre sus

estándares están el CDMA de origen estadounidense y el GSM (*Global System for Mobile communication*) de origen europeo. Presentan características muy especiales de comunicación, pero la velocidad de transferencia de 9600 bps es motivo suficiente para no optar por esta tecnología. Sin embargo el rango de cobertura de una célula es bastante grande y puede ocupar incluso toda una ciudad.

➤ Redes inalámbricas 802.11

Estas presentan características muy favorables para la conexión entre computadoras grandes o medianas y su uso se ha popularizado. EL 802.11 es un estándar de la IEEE (*Institute Electrical Electronic Engineer*) que establece un modelo OSI en capas física y de enlace. Existen 11 versiones de este estándar pero solo 3 de ellos son los de mayor relevancia. En la tabla 2-16 se muestra una comparación entre los distintos estándares.

<i>Estándar</i>	<i>Velocidad [Mbps]</i>	<i>Frecuencia de operación</i>	<i># usuarios por PA</i>	<i>Ventajas</i>	<i>Desventajas</i>
802.11a	54/74	5 GHz	64	<ul style="list-style-type: none"> •Velocidad •Poca interferencia 	<ul style="list-style-type: none"> •Incompatibilidad con 802.11b/g. •No tiene QoS
802.11b	11/22	2.4 GHz	32	<ul style="list-style-type: none"> •Bajos precios debido a su alta demanda •Gratuidad de la banda 	<ul style="list-style-type: none"> •Alta interferencia •Las mismas que el 802.11a
802.11g	54	2.4 GHz	32	<ul style="list-style-type: none"> •Mayor velocidad que el 802.11b 	<ul style="list-style-type: none"> •Las mismas que el 802.11b

Tabla 2-16 estándares 802.11x

El conjunto de esta tecnología involucra, adaptadores de red inalámbrico y puntos de acceso (PA). Los primeros existen del tipo PCI, USB o PCMCIA y los segundos son módulos independientes que permiten los enlaces entre los primeros y/o con concentradores, ruteadores.

Dos de los mayores inconvenientes de las redes inalámbricas son la distancia y la seguridad. Debido a que entre el transmisor y el receptor existe cierta atenuación de la señal electromagnética causada por el ambiente (paredes, ventanas, muebles, gases, lluvia, etc.) y esto provoca que a mayor distancia

entre el punto de acceso y el adaptador de red inalámbrico disminuya la velocidad de transferencia hasta llegar a desaparecer el enlace. El máximo alcance de una red inalámbrica normal es de 300 m.

Por otro lado está el aspecto de la seguridad de la red, la cual consiste en prevenir y proteger los ataques o amenazas hacia el sistema²². Estos consisten en la interrupción de un recurso del sistema, la interceptación de la señal, la modificación de datos y la suplantación de identidades para infiltrarse en sistemas protegidos.

2.5 Control Remoto Virtual

Para enlazar dos computadoras por medio de una red, ya sea alámbrica o inalámbrica, es necesario utilizar un software que las comunique. Ese software debe de cumplir con los protocolos compatibles con la red de comunicación. Generalmente ese protocolo es el TCP-IP de Internet, que tiene normas muy estrictas y ya establecidas por la IEEE.

Actualmente muchos lenguajes de programación virtuales y orientados a objetos ofrecen librerías o controladores para habilitar una comunicación vía TCP-IP. De cualquier manera sin importar el lenguaje de programación, el sistema operativo e incluso la aplicación, la disposición de la red tendrá que cumplir con las características que se establecen en el diagrama de la figura 2.8. Donde se requiere una computadora que sea el servidor y otra el cliente. La estación de control es el cliente y la SRM es el servidor.

La aplicación para comunicar estas dos computadoras puede estar basado en el conjunto de protocolos TCP-IP como son HTTP (*Hyper Text Type Protocol*), FTP (*File Transfer Protocol*), Telnet, SSH y otros. Sin embargo, estos protocolos cuentan con cierta estructura para cumplir los lineamientos para lo que fueron diseñados, por lo que resultaría muy complicado y a veces imposible tratar de adaptarlos al tipo de aplicación que se quiere utilizar. Pero bien se podría utilizar un servicio TCP-IP que bien cumpla con la estructura que se busca para este desarrollo. Una alternativa

²² Gonzalez, Luis. *Tesis "Seguridad de redes inalámbricas en un ambiente académico"*. México, 2005, UNAM

a explorar, para trabajos posteriores, es que se podría utilizar lo que se conoce como *socket*, que a partir de un lenguaje de programación de alto nivel, se establecen los comandos para poder transferir información a través de un protocolo TCP-IP personalizado.

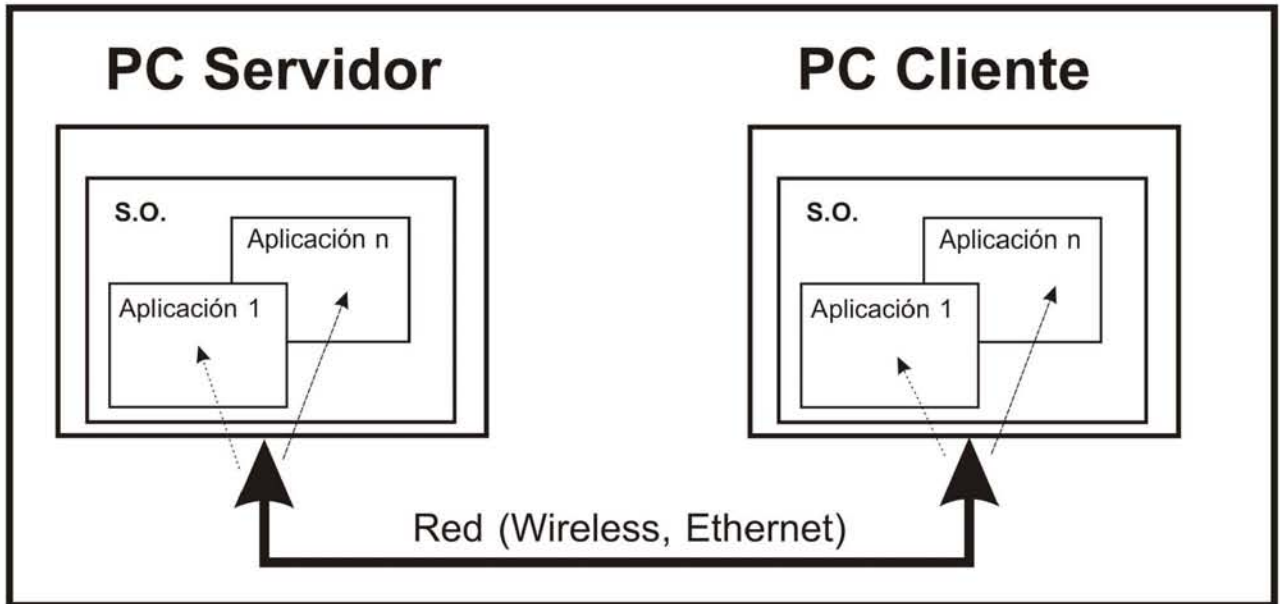


Figura 2.8 Esquema TCP-IP

Un servicio/protocolo muy usado en la administración de servidores Unix, GNU/Linux y Windows Server y hasta en las computadoras de alto desempeño, es el VNC (*Virtual Network Computing*²³). Se trata de un servicio, al igual que el SSH, que permiten el control de un servidor vía remota. El servicio VNC es comúnmente utilizado para conectarse a servidores desde lugares remotos o simplemente desde cuartos cercanos a él. Ya que se requiere que estos equipos estén instalados en cuartos limpios con humedad y temperatura controlada, imposibilita que los usuarios puedan estar cerca de él. Con los servicios VNC, cualquier PC o cualquier máquina digital que entienda del protocolo, es posible conectarlo remotamente y tener una sesión activa en las tareas del procesador central del servidor o estación de trabajo.

Existen VNC gráficos y de texto. Los de texto son más rápidos, pero a veces es necesario visualizar gráficos. En cambio los gráficos abren una sesión remota en una terminal idéntica a la que se estaría viendo si se tuviera contacto físico con el monitor del servidor.

²³ http://en.wikipedia.org/wiki/Virtual_Network_Computing

Existen varios tipos de VNC, algunos privados que cobran licencias con altos costos por su uso y otros de distribución libre, mejor conocidos como “software libre”. También dependiendo de la arquitectura del cliente y del servidor existen distintas versiones, pero si pertenecen al mismo desarrollo entonces se puede estar ejecutando una sesión gráfica de un servidor Unix en un cliente con Windows. Y al revés.

Cabe mencionar que para el caso de este proyecto, la conexión cliente-servidor deberá de ser unívoca, es decir punto a punto, ya que por cuestiones de seguridad no es para nada conveniente que varios usuarios tengan acceso sobre el servidor VNC y solo uno pueda acceder. Esto se consigue cuando el servidor solo recibe peticiones desde la dirección IP de la estación de control.

Con estos servicios, se tiene absoluto control sobre el hardware y software de una máquina desde lejos. Incluso desde el otro lado del mundo.

ACTUALIZACIÓN DE LA SONDA ROBOT MÓVIL

Una vez localizados los inconvenientes que presentaba el diseño original de la SRM, se determinaron los aditamento necesarios que debían de ser incorporados, de acuerdo a los objetivos planteados, para actualizar dicha sonda; así como la eliminación de ciertos dispositivos que aparecían en el diseño original que sin embargo, no eran necesarios debido a que parte de la solución incluía las funciones del mismo y más. En el capítulo anterior se discutió cuales deberían de ser los posibles caminos en cuanto a soluciones se trata, para poder ir resolviendo paso a paso las metas planteadas desde el principio del proyecto. Estas decisiones se basaron en tres aspectos, la disponibilidad tecnológica, el costo económico de la integración de los nuevos componentes y sobre todo los aspectos técnicos para la actualización de la sonda.

La actualización no solo contemplaba agregar componentes electrónicos sino también implicaba hacer nuevos programas de cómputo que tenían que ser integrados a los ya existentes y la adaptación del nuevo software. En este capítulo se presentan todos los cambios y los nuevos diseños que fueron realizados a la sonda. En primer lugar se describe el esquema propuesto para actualizar la sonda, además de las implicaciones para la integración de la computadora personal a la sonda robot tanto en software como en hardware. Posteriormente se habla del diseño de la tarjeta encargada de interactuar con las etapas de potencia y la computadora personal. Y por último, se presenta la estructura y la interacción entre los distintos programas de cómputo que controlan a la sonda.

3.1 Esquema de la nueva configuración de la Sonda Robot Móvil

El esquema de actualización de la sonda se muestra en la figura 3-1. Este tiene dos componentes básicos, la “Estación de Control” (EC) y la SRM. La estación de control puede ser cualquier computadora personal, pero que contiene el Software de Control Remoto y a través del cual el usuario le envía ordenes, vía inalámbrica, a la sonda.

En la SRM se eliminó la tarjeta de control que contenía al microcontrolador 8751 y se sustituyó con dos nuevos bloques, la “Computadora de Control” (CC) y la “Tarjeta de Expansión de Puertos” (TEP). En la CC se realiza el control de la sonda. También aquí se realizan los enlaces con la EC, la comunicación con la TEP. Se envían y reciben datos desde la EC indicados por el usuario. Aquí se ejecutan el ambiente virtual “Softelenav”, el software de control de la TEP y se posteriormente se ejecutarán los programas de inteligencia artificial y de seguimiento de rutas una vez que estén concluido.

Otro componente importante en la SRM y en particular del proyecto de actualización es la TEP. En esta tarjeta se envían señales digitales hacia las etapas de potencia que a su vez controlan los motores del brazo manipulador, de los motores de dirección y tracción, así como de los distintos sensores. Esta tarjeta tiene el propósito de comunicar a la CC con los sensores y actuadores de la sonda.

En las secciones posteriores se explicaran con mayor detalle los componentes de la sonda.

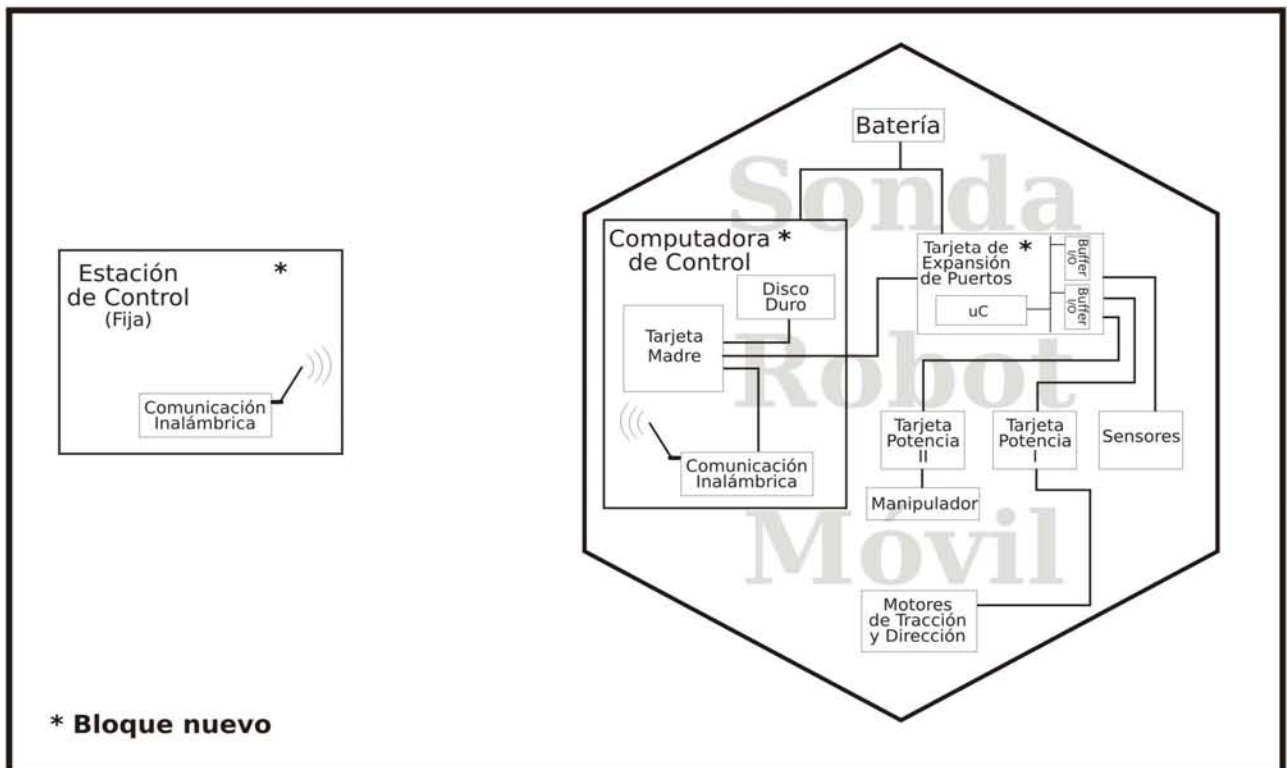


Figura 3-1 Esquema de la actualización de la SRM

3.2 Computadora de Control (CC) de la Sonda Robot Móvil

Ya en el capítulo anterior se mencionó el porque es necesario utilizar un sistema de cómputo compatible con una computadora de arquitectura x86 para controlar a la sonda. Cabe mencionar que solo es necesario utilizar lo que se conoce como el CPU. Ya que lo que interesa es aprovechar los recursos de cómputo de un procesador, memoria RAM, un medio de almacenamiento de datos masivo (disco duro) y la placa madre donde se colocan todos estos dispositivos. De esta manera el uso de periféricos como tarjeta de video, tarjeta de sonido, ratón, teclado, monitor, impresora y otros no será necesarios. Sin embargo, existen varios modelos, tanto de tarjetas madre como de procesadores, en el mercado compatibles con este proyecto. Por lo que fue necesario hacer una selección entre diversos tipos.

En la tabla 2-1 del capítulo 2, se muestra una tabla de los requerimientos mínimos sugeridos por el fabricante, del tipo de hardware para que el sistema operativo Windows versión 98 en adelante pueda ejecutarse. Por lo que para cualquier sistema operativo de Windows que se elija, el procesador que deberá utilizarse tiene que ser igual o superior a la gama de procesadores Pentium de Intel o su equivalente en su versión de AMD. Así que queda descartado utilizar procesadores iguales o inferiores al 486 de Intel o su equivalente en su versión de AMD.

Otra característica a considerar tiene que ver con el grado de portabilidad que puede ofrecer al proyecto de actualización. Es decir, se necesita que la computadora en general sea físicamente pequeña, que demande un bajo consumo de potencia, que los componentes sean de un tamaño adecuado que no extralimite los tamaños de la sonda, ni que impida la actualización de la misma por el hecho de que algún componente sobresalga del área hexagonal o supere la altura disponible del vehículo.

El tema del consumo de potencia de la sonda es también puesto a consideración; pues no hay que olvidar que la sonda ya cuenta con los espacios asignados para el suministro de potencia eléctrica (batería), lo que impide que la computadora personal requiera de niveles de voltaje superiores a los que una batería puede ofrecer. Además se trata de una sonda inalámbrica, es decir que debe ser

completamente independiente y no estar atada o restringir su movimiento a un determinado espacio, por lo que resulta imposible suministrarle potencia eléctrica a través de un cable.

Tampoco hay que olvidar los niveles de voltaje con los que opera una placa madre y los niveles de voltaje de una batería. Es decir que en la mayoría de los casos, la placa madre compatible con el estándar ATX requiere de distintos niveles de voltaje. Los cuales satisface la fuente de poder ATX, que convierte la corriente alterna de la línea de potencia eléctrica con valor de $127 V_{rms}$ a los valores de -12 V, -5V, tierra, 3.3 V, 5 V y 12 V. Como la batería instalada solo proporciona 12 V de corriente directa, es necesario, en algunos casos reducir el nivel de voltaje, y en otros proporcionar un suministro de potencia eléctrico negativo. En caso de reducir el nivel de voltaje, este puede implementarse con un regulador lineal de voltaje. Estos dispositivos reducen el nivel de voltaje, pero debido a su linealidad disipan demasiada potencia. Para el caso de voltajes negativos y también de los positivos se pueden generar con una fuente de voltaje conmutada también conocida como “*chopper*”.

Las fuentes conmutadas son dispositivos que convierten una señal de entrada de corriente directa, a otra señal también de corriente directa pero con diferentes niveles de voltaje que pueden ser mayor o menor al de la entrada. Construir un *chopper* que proporcione los 5 niveles de voltaje antes mencionados requiere de dispositivos que son costosos y que su construcción requiere de tareas que están fuera del alcance de los objetivos de este trabajo. Motivos suficientes para considerar otras opciones como puede ser otro tipo de tarjeta madre. En algunos modelos de placas madres de pequeño tamaño como son la Mini ITX y las Nano ITX basta con alimentarla con 12 V de corriente directa sin necesidad de aplicarle otros niveles de voltaje.

De entrada, a partir de la tabla 2-2 se puede tomar una decisión del tipo de placa madre que cuenta con las características más adecuadas al proyecto. Se había mencionado la necesidad de utilizar una tarjeta madre que ocupara el menor tamaño dentro de la sonda robot. Por lo que un tarjeta madre con formato Mini ITX o Nano ITX resultan opciones muy favorables. En la tabla 3-1 se muestran distintos tipos de tarjetas madres con formatos Mini y Nano ATX.

En la mayoría de los modelos mostrados en la tabla 3-1 se observa que se mantiene la misma cantidad de periféricos entre cada tarjeta. En algunos casos varía la cantidad de memoria RAM que puede ser colocada en la placa madre, así como del tipo de conector que utiliza la memoria. Y en el caso del modelo de los procesadores lo que cambia es la velocidad de procesamiento, que va desde 800MHz hasta 1.3 Ghz en el caso de la tarjeta EPIA SP. Esta diferencia poco importa en el diseño de este proyecto, ya que la versión de sistema operativo que se adoptó puede ejecutarse en el modelo de tarjeta madre de menor capacidad de las mostradas.

Si la decisión consiste en buscar la tarjeta madre que contenga el mayor número de periféricos, se estaría desviando la atención de lo que realmente es importante para el proyecto. Es decir, en la mayoría de los casos todas las placas madres tienen al menos 2 puertos USB, 1 *slot* PCI, conectores para ratón y teclado, tarjeta de video integrada a la placa base, conectores para disco duro, tarjeta de red integrada, puerto serie y paralelo. Atributos que le son muy favorables al diseño del proyecto ya que las necesidades se satisfacen en buena medida.

Dadas las condiciones mencionadas en los párrafos anteriores, para tomar alguna decisión sobre cuál es el mejor modelo de tarjeta que debiese adoptarse para este proyecto, se tomaron 3 consideraciones; la placa madre debe de ser del formato Mini ITX debido a que:

<i>Modelo Marca</i>	<i>Formato</i>	<i>Procesador</i>	<i>Memoria</i>	<i>Periféricos</i>	<i>Alimentación Consumo</i>
EPIA ML VIA	Mini ITX	VIA C3 a 800Mhz (ML8000)	DDR266 hasta 1Gb	<ul style="list-style-type: none"> • 1 slot PCI • 2 ATA 133 • 1 LAN 10/100 • Audio AC' 97 • VGA • 2 USB 2.0 • 1 Paralelo • 1 Serie • PS2 Teclado • PS2 Ratón 	ATX 27.416 W
EPIA M VIA	Mini ITX	VIA C3 a 1Ghz (M10000)	DDR266 hasta 1Gb	<ul style="list-style-type: none"> • 1 slot PCI • 2 ATA 133 • 1 LAN 10/100 • Audio AC' 97 • VGA • TV out • 2 USB 2.0 • 1 Paralelo • 1 Serie • PS2 Teclado • PS2 Ratón • 1 Firewire 	ATX 30.404 W
EPIA PD VIA	Mini ITX	VIA C3 a 1Ghz	DDR266 hasta 1Gb	<ul style="list-style-type: none"> • 1 slot PCI • 2 ATA 133 • 1 LAN 10/100 • 4 USB 2.0 • 1 Paralelo • 1 Serie 	ATX 27.481 W

<i>Modelo Marca</i>	<i>Formato</i>	<i>Procesador</i>	<i>Memoria</i>	<i>Periféricos</i>	<i>Alimentación Consumo</i>
		(PD10000)		<ul style="list-style-type: none"> • Audio AC' 97 • VGA • TV out • PS2 Teclado • PS2 Ratón • 1 Firewire 	
EPIA MII VIA	Mini ITX	VIA C3 con ventilador a 1Ghz (MII10000E) y a 1,2GHZ (MII12000E)	DDR266 hasta 1Gb	<ul style="list-style-type: none"> • 1 slot PCI • 2 ATA 133 • 1 LAN 10/100 • Audio AC' 97 • VGA • TV out • 1 Firewire • 4 USB 2.0 • 1 Paralelo • 1 Serie • PS2 Teclado • PS2 Ratón • 1 Card Bus • 1 Compact Flash 	ATX 27.708 W
EPIA TC VIA	Mini ITX	VIA C3 a 1Ghz (TC10000E)	SODIMM 200-pin (portátiles) DDR266 hasta 1Gb	<ul style="list-style-type: none"> • 1 slot PCI • 2 ATA 133 • 1 LAN 10/100 • Audio AC' 97 • VGA • 4 USB 2.0 • 1 Paralelo • 1 Serie • PS2 Teclado • PS2 Ratón 	12 V _{DC} 24.214 W
EPIA SP VIA	Mini ITX	VIA C3 a 1,3Ghz (SP13000)	DDR400 hasta 1Gb	<ul style="list-style-type: none"> • 1 slot PCI • 2 ATA 133 • 2 SATA • 1 LAN 10/100 • Audio AC' 97 • VGA • 6 USB 2.0 • 1 Paralelo • 1 Serie • PS2 Teclado • PS2 Ratón • FIR (InfraRojos) • VIP (VideoInterface) 	ATX 27.708 W
EPIA N VIA	Nano ITX	LUKE CoreFusion (N10000)	SODIMM DDR400 (portátiles) hasta 1Gb	<ul style="list-style-type: none"> • 1 slot miniPCI • 2 ATA 133 • 1 SATA • 1 LAN 10/100 • 6 USB 2.0 • Audio AC' 97 • VGA 	Mini ATX 15.677 W

Tabla 3-1 Tarjetas madres de formatos Mini y Nano ITX

1. Son las tarjetas de menor tamaño físico y es posible integrarla de mejor manera en la estructura física de la sonda robot móvil,
2. Presenta el menor consumo de potencia eléctrica a diferencia de los formatos grandes y
3. en específico, la única placa madre del formato Mini ITX que se alimenta con solo una señal de 12V de corriente directa, es el modelo EPIA TC 10000E. Esta razón es suficiente para descartar los otros modelos que son alimentados con fuentes ATX.

3.2.1 Potencia eléctrica de la de la Sonda Robot Móvil

Es necesario realizar un análisis de potencia de la SRM, ya que a partir de ello se sabrán los límites y las características de los dispositivos que proporcionan dicha potencia.

En la tabla 3-2 se muestra un estudio de las características eléctricas de cada dispositivo de la SRM. En la segunda columna aparecen los voltajes nominales de cada dispositivo. Aquí se encuentra generalizado el uso de 12 y 5V, lo cual no representa ningún problema, pues existen reguladores lineales de voltaje para dichos voltajes.

<i>Dispositivo</i>	<i>Voltaje_{DC}</i> <i>[V]</i>	<i>Corriente_{max}</i> <i>[A]</i>	<i>Potencia_{max}</i> <i>[W]</i>
6 motores de pasos (Brazo manipulador)	12	0.25 c/u. Un total de 1.5	18
Motor de tracción	12	2	24
Motor de dirección	12	0.5	6
Compás	5	0.004	0.02
Sensor ultrasonido (maestro)	12	.011	0.132
Sensor ultrasonido (transmisores)	12	.010	0.12
Sensor ultrasonido (multiplexor)	12	.015	0.18
Odómetro	5	.010	0.05
PIC16C745	5	0.2	1
74LS138	5	0.0064	.032
74LS245	5	0.095 c/u. Un total de 0.76	3.8
VIA EPIATC10000E ²⁴	12	5	60
Samsung MP0402H	5	0.48	2.4
TOTAL		10.4964	115.734

Tabla 3-2. Voltajes de operación de la EPIA TC 10000E.

²⁴ Users's Manual EPIA TC Mini ITX Mainboard, VIA Technologies Inc., 2004

En la tercera columna se muestran los valores de corriente de cada dispositivo. Cabe destacar que este valor se considera cuando el dispositivo consume su máximo de potencia. En la parte posterior de la tabla se realiza una suma de todas las corrientes (10.4964 A). En la última columna aparece la potencia de cada dispositivo y la suma total (Potencia máxima consumida) es de $P_{\text{consumida}}=115.734$ W.

Si el factor de corriente \times tiempo de cada batería es de 7Ah y estas son colocadas en serie, se produce una potencia $P_{\text{baterías}} = (7A) (24V) = 168$ W en una hora. Como este valor es mayor que la potencia consumida, $P_{\text{baterías}} > P_{\text{consumida}}$, se garantiza que al menos en una hora la sonda no tendrá problemas de falta de energía. Además la $P_{\text{consumida}}$ es un valor muy conservador, ya que se basa en el hecho de que los dispositivos consumen su máxima potencia y lo más importante es que no todos los dispositivos están encendidos todo el tiempo. Como es el caso de los motores y en algunos sensores. Ya que no todo el tiempo se esta moviendo la sonda y el brazo. Así que es probable que la SRM dure más de una hora cuando las baterías se encuentran a carga completa.

Cada motor y sensor cuenta con su fase de acoplamiento o de suministro de potencia eléctrica desde las baterías, por lo que en este trabajo no se desarrollará hardware extra. Sin embargo para el caso de la tarjeta madre y de la interfaz USB.

En el capítulo anterior y en el Apéndice A se muestran las características principales de la tarjeta madre EPIA TC 10000E. Para alimentar esta tarjeta es necesario una fuente de voltaje de 12 V de corriente directa. Esto no es problema ya que las baterías que suministran la potencia eléctrica a la sonda son de 12V cada una, lo que da un voltaje de 24V. Sin embargo estas baterías, en la medida en que se usan, disminuyen su voltaje y llega el momento en que los niveles mínimos de voltaje de operación de la tarjeta madre, la obligan a apagarse. Por lo que es necesario implementar un dispositivo para supervisar, en todo momento, el nivel de voltaje y evitar así el corte intempestivo de corriente que puede dañar a la tarjeta madre y al disco duro que va conectado a ella.

Para esto se diseño una Tarjeta de Alimentación (TA) (ver figura 3-2) que tiene la función de regular el voltaje de salida para 12 V_{DC}, 5 V_{DC}, el control del voltaje de encendido que será utilizado por el

PIC para encender a la CC, trae una extensión del conector F_PANEL de la Tarjeta Madre donde se controla a la CC para su encendido, su reinicio, y leds para determinar el estado de la CC.

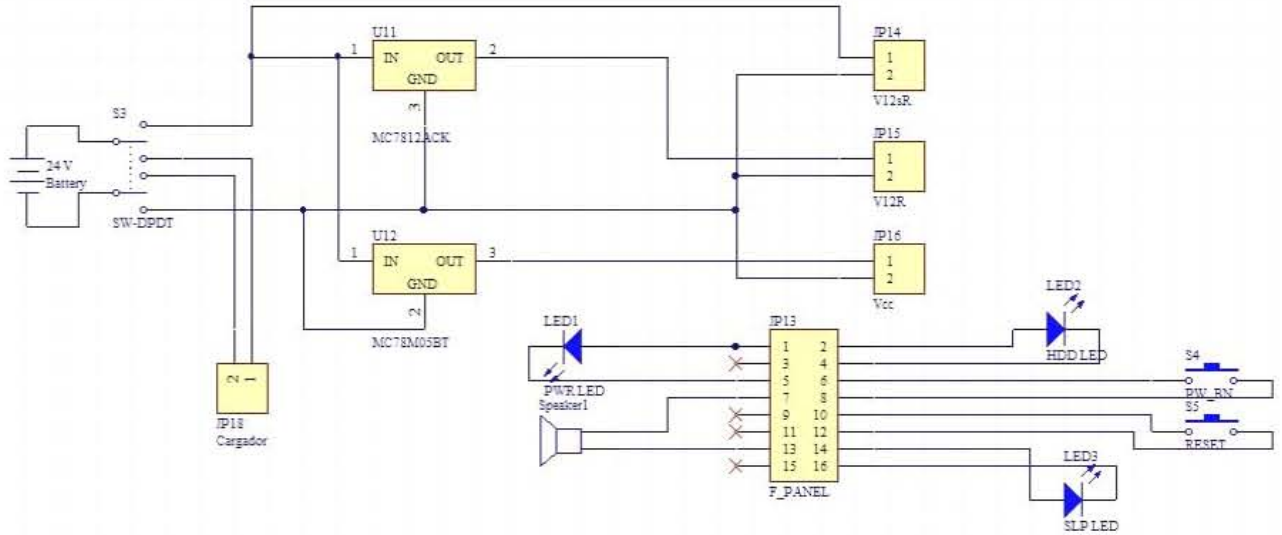


Figura 3-2 Tarjeta de Alimentación

El switch S3 conecta ya sea con el conector JP14 (cargador) para realizar el proceso de carga de la batería o con el resto del circuito de la TA. Los integrados U11 y U12 (LM7805 y LM78H12) son reguladores lineales de voltaje. En el primer caso regula el voltaje entregado por la batería ($0 < V_{\text{batería}} < 24 \text{ V}_{\text{DC}}$) a 5 V_{DC} (V_{CC}) para el siguiente circuito el LM78H12 también se trata de un regulador lineal de voltaje, que entrega a su salida 12 V_{DC} (V_{V12R}) con una corriente de 5 A. El LED4 se encenderá si existe un voltaje adecuado para alimentar a la tarjeta madre.

El botón PW_BN enciende la CC. El botón RESET reinicia la CC. Los leds PWR_LED, HDD_LED indican si la CC está encendida y el disco duro también, respectivamente. Para escuchar los sonidos que genera el sistema se conecta una bocina entre las terminales 7 y 13 del conector F_PANEL.

Ya en trabajos posteriores a este, se realizarán las acciones que conlleven a tomar la mejor decisión cuando se presente el bajo nivel de voltaje así como el desarrollo del sistema para indicar y guiar a la SRM que se dirija a su estación de recarga cuando ocurra tal suceso. Aquí solamente se propone el diseño de la tarjeta y de las señales necesarias para su control.

3.2.2 Instalación del Disco Duro en la Sonda Robot Móvil

El sistema operativo y el software de control de la sonda robot deben de estar almacenados en algún dispositivo físico para que cuando el procesador requiera datos, estos estén disponibles. El medio físico ideal para esta acción son los discos duros o tarjetas de memoria *flash*. Estos últimos empiezan a posicionarse en el mercado de los medios de almacenamiento y en un futuro próximo seguramente sustituirán a los discos duros. Debido a que las características con las que está fabricados (semiconductores) tienen tiempos más largos de retención de la información. Además son más veloces que los magnéticos. Sin embargo, este tipo de tecnología es todavía bastante cara y aún no existen dispositivos de memoria *flash* con capacidades de almacenamiento similares a las de los discos magnéticos.

En este proyecto se uso un disco duro magnético. Existen dos tipos de discos duros para PC los del tipo IDE (Integrated device Electronics) y los SATA (Serial Advanced Technology Attachment), estos últimos son los de nueva tecnología y entre sus mejoras ofrecen mayor velocidad que los IDE. Sin embargo la tarjeta madre EPIA TC 10000E no ofrece conectores de este tipo y solo tiene dos conectores tipo IDE; por lo que este fue el tipo de disco duro que se eligió para el proyecto. Existen dos tamaños para estos discos duros, para las PC de escritorio y para las portátiles.

En la tabla 3-3 se muestra una comparación entre estos dos tipos de discos duros IDE, uno para computadoras portátiles y uno para computadoras de escritorio. La tabla es contundente al mostrar las ventajas del primero con respecto al segundo. El tamaño, el peso, el consumo de potencia y el precio son factores que se tomaron en cuenta para elegir al disco duro. Por lo que se opto por utilizar el disco duro Samsung MP0402H, sus características se muestran en el Apéndice A.

		<i>Seagate Barracuda 5400.1</i>	<i>Samsung MP0402H</i>
Uso		PC de escritorio	PC portátil (Laptop)
Interfaz		ULTRA ATA/100	ULTRA ATA/100
Costo [pesos]		\$ 620	\$ 1,150
Capacidad [GB]		40	40
Velocidad [rpm]		5400	5400
Tamaño [mm]		20 x 102 x 146.9	9 x 100 x 69
# de pines		40	44
Peso [g]		450	100
Voltaje de alimentación [V]		+12 V _{DC} ±10%+5 V _{DC} ±10%	+5 V _{DC} ±5%
Consumo de Potencia [W]	Busca	6	2.4
	Lee/escrbe	6.4	2.4
	Marcha lenta	4.5	0.85
	Espera	0.9	0.25
	Suspendido	0.9	0.1

Tabla 3-3 Comparación entre dos discos duros IDE.

Como el conector IDE de este disco duro es de 44 pines y el conector IDE de la EPIA TC 10000E es de 40 pines se hizo uso de un adaptador que hace la conversión. La diferencia consiste en que los discos duros portátiles toman la energía eléctrica del mismo conector IDE de 44 pines, a diferencia de los discos duros de las computadoras portátiles que tienen otro tipo de conectores destinado a ello. En la figura 3-3 se muestra el adaptador que se utilizó.



Figura 3-3 Adaptador de 40 a 44 pines IDE.

Debido a que el estándar IDE permite la conexión en paralelo de varios discos duros y/o unidades ópticas (lectores/grabadores CD-ROM, DVD) es necesario establecer una jerarquía entre todos los dispositivos de almacenamiento masivo conectados a una computadora. Es decir que el BIOS de la tarjeta madre requiere saber, antes de la inicialización del sistema operativo, que disco es el que lo contiene, asignarles un orden de lectura y empezar el proceso de carga de sistema. Esta configuración se hace conectando de distintas maneras 4 pines con un *jumper* que tienen todos los discos duros internos. En los últimos 4 pines del extremo derecho del conector de 44 pines se realiza la configuración de la jerarquía del disco duro(ver figura 3-4).

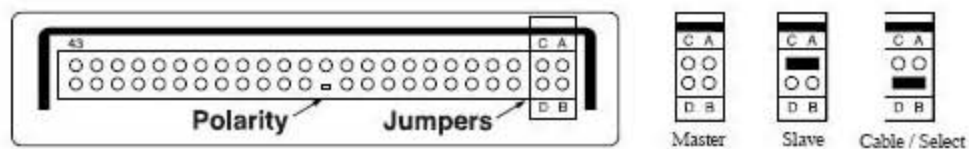


Figura 3-4 Configuración de un disco duro portátil IDE.

Si no se conecta ningún pin, se estará seleccionando el modo Maestro (*Master*) y el BIOS sabrá que esa es la unidad donde debe empezar a leer. Si se conectan los pines C y A se estará seleccionando el modo “Esclavo” (*Slave*), este se usa si existe otro disco duro o alguna otra unidad óptica conectado a la PC. Al conectar los pines D y B se estará seleccionando el modo “Selección por Cable” (*Cable / Select*) donde automáticamente el BIOS, dependiendo de la disposición de conexión por medio del cable IDE entre la tarjeta madre y las unidades, determinará cual es el Maestro y cual es el Esclavo.

Para instalar el sistema operativo se requirió conectar una unidad lectora de CD-ROM a la tarjeta

madre junto con el disco duro Samsung MP0402H. De cualquier forma el disco duro siempre estará configurado como Maestro.

3.2.3 Instalación del Módulo Inalámbrico en la Sonda Robot Móvil

La selección del módulo inalámbrico se hizo a partir de las características físicas que ofrecen los módulos conectados a través de un bus USB en comparación con las tarjetas conectadas con el puerto PCI de la placa madre de una PC. Las tarjetas PCI son más grandes que los módulos USB. Además, debido a la disposición del estándar PCI que obliga a que las ranuras de este puerto sean perpendiculares a la superficie de las placas madre y como las tarjetas PCI son tabletas de circuito impreso ocupan, por encima de la placa madre, un espacio que pudiese ocuparse para otro dispositivo. En cambio los módulos USB son más pequeños y en particular el dispositivo inalámbrico que se adquirió para este proyecto cuenta con una “unión universal” (ver figura 3-5) telescópica que facilita posicionar al dispositivo inalámbrico en la mejor postura para recibir la señal de radiofrecuencia.

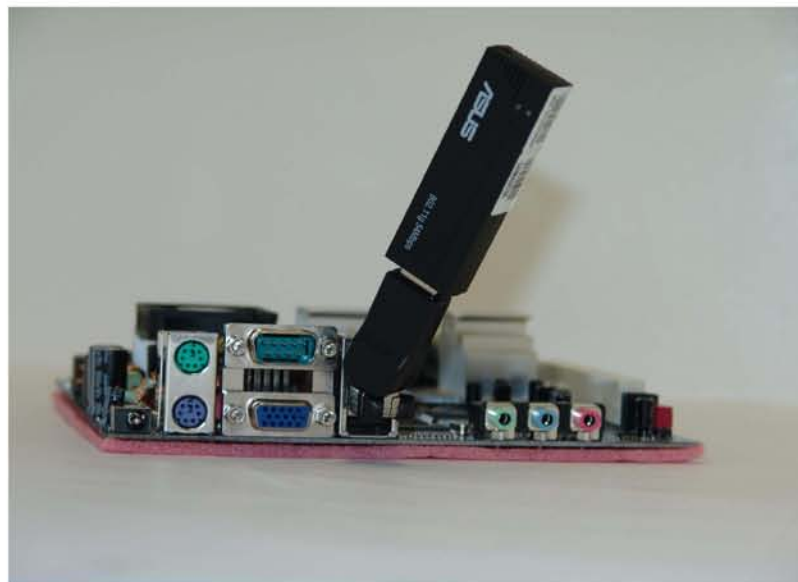


Figura 3-5 Módulo inalámbrico por USB

La instalación de este dispositivo consiste en insertar el CD-ROM y copiar los controladores del

módulo USB al disco duro. Una vez hecho esto basta con conectar el módulo al puerto USB de la tarjeta madre. Si la instalación del software es correcta el sistema reconoce al hardware e inicia el programa de control de la red inalámbrica.

Este programa solicita algunos datos al usuario para poder realizar la configuración TCP-IP. En la tabla 3-4 se anotan los datos que se le proporcionaron.

Cabe resaltar que el tipo de red se seleccionó “Ad-Hoc” para poder prescindir de un punto de acceso y que la comunicación se realice punto a punto, es decir se establece comunicación directa sin intermediarios entre la SRM y la estación de control.

<i>Parámetro</i>	<i>Datos</i>
Tipo de red (Ad-Hoc o AP)	Ad-Hoc
Dirección IP	192.168.1.30
Máscara de red	255.255.255.0
Puerta de enlace	192.168.1.1
Nombre de red (SSID)	Robot
Canal	1
Taza de transferencia	Automática
Palabra WEP	Ninguna

Tabla 3-4 Configuración de la red inalámbrica en la SRM

3.2.4 Instalación del sistema operativo y controladores en la Computadora de Control

Como se ha venido mencionando es necesario que la CC de la SRM ejecute alguna versión del sistema operativo Windows de Microsoft, para realizar los propósitos planteados al principio de este trabajo. Es por eso que se requiere que un sistema operativo se esté ejecutando en la memoria de la Tarjeta Madre de la CC y que este interactúe con el hardware y software desarrollados en este proyecto.

En la tabla 2-1 se especifican las condiciones mínimas de hardware para ejecutar Windows. Con

base en ello y otras consideraciones se optó por la Tarjeta Madre EPIA TC 10000E. Sin embargo, existen al menos 4 versiones del sistema operativo Windows que se acoplan al proyecto. Estas son la versión 98, 2000, XP y 2003 de Windows.

<i>Nombre del Software</i>	<i>Tipo</i>	<i>Función</i>	<i>Fabricante</i>	<i>Versión</i>	<i>Espacio en disco duro</i>	<i>Espacio en RAM</i>
Microsoft Windows2000	Sistema Operativo	Administrar el hardware y software	Microsoft	5.00.2195 Service Pack 4	606 MB	70 MB
Asus WLAN Card Utilities/Drivers	Controlador	Red inalámbrica	ASUS	2.2.2.8	1.07 MB	12 MB
VIA CLE266 Display Drivers and Utilities	Controlador	Video	VIA	6.14.10.55	ND	ND
TightVNC	Programa / Servicio	Monitor Virtual (Servicio)	TightVNC	1.2.9	1.98 MB	ND
VIA Audio Driver	Controlador	Audio	VIA	6.14.1.3950	12 kB	ND
VIA NIC SET	Controlador	Ethernet	VIA	3.22.0.407	176 kB	ND
Driver USB	Controlador	USB	USB.org	2.0	ND	ND
HIDCOMM	Controlador	Controladores HID USB	Microchip	1.0	810 kB	ND
Softelenav	Programa	Ambiente virtual de la SRM	DEA, DISCA, IIMAS	1.0	10 MB	10 MB
ControlSRM	Programa	Control de la SRM	DEA, DISCA, IIMAS	1.0	5 MB	5 MB

ND = No disponible

Tabla 3-5 Software instalado en la SRM

A principios del 2006, Microsoft anunció que ha dejado de proporcionar soporte a Windows 98; esto significa que en dado caso de que hubiese algún error sin su correspondiente solución por el equipo de soporte de esta versión, ya no existirán más correcciones. Esto, aunado al hecho de que la mayoría de los fabricantes de hardware y software no desarrollarán más, versiones de sus

controladores para Windows 98. La versión Windows 2003 está enfocada a aplicaciones de servidores que no son útiles a los requerimientos de este proyecto. Entre sus cualidades de esta versión están el control de cuentas de usuario, administración de páginas Web, servicios de FTP, SMTP y POP. Dado esto se descarta la posibilidad de integrarlo a la CC de la SRM. Por último, están las versiones 2000 y XP, similares en general, sin embargo la primera ofrece mayor estabilidad y proporciona alternativas para mejorar su administración de mejor manera que lo hace XP, ya que este último está enfocado para facilitar su administración a usuarios finales no avanzados. Con la versión 2000 se pueden controlar de mejor manera los servicios, cuentas de usuario, seguridad, la administración de procesos y programas que bien puede beneficiar al diseño de este robot.

Así la versión 2000 de Windows fue la elegida para instalarse en la CC de la SRM. En la tabla 3-5 se muestra el software instalado en el disco duro de la CC. También fue necesario instalar el software de control de la SRM (Sección 3.4) y los controladores para el manejo del hardware (ver figura 3-6).



Figura 3-6 Hardware instalado en la SRM.

3.3 Tarjeta de Expansión de Puertos (TEP)

Parte de la actualización de la SRM incluye agregar más puertos de datos digitales para el control de todos los motores de pasos del manipulador, motores de corriente directa para la dirección y la tracción, direccionar cada transmisor ultrasónico y recibir la respuesta del sensor maestro ultrasónico y en general administrar todos los sensores analógicos y digitales.

El problema que se tenía en el diseño anterior de la SRM es que el número de puertos de datos digitales era limitado, y se tuvo que sacrificar el uso de algunos dispositivos (locomoción del brazo manipulador) para poder incluir otros que son indispensables para el control de la sonda. Esto representa un serio problema, debido a que al no contar con las conexiones pertinentes para comunicar la tarjeta de control del robot con la etapa de potencia, y ésta a su vez con algunos de los motores de pasos del brazo manipulador, no hay una interacción directa y completa con el manipulador y no se logra el propósito para el que fue construido.

Para solucionarlo fue necesario construir una tarjeta que comunicará a la CC con las etapas de potencia. La Tarjeta de Expansión de Puertos (TEP) cumple con este propósito. En el capítulo anterior se explica por que fue necesario utilizar una interfaz basada en el bus USB y que dispositivo (PIC16C745) usar para la programación de la tarjeta.

A continuación se describe el proceso de diseño, primero se presenta la arquitectura física de la disposición de los componentes y posteriormente se describe el software que comanda al PIC.

3.3.1 Hardware de la Tarjeta de Expansión de Puertos

En la figura 2-6 aparece un diagrama de bloques de la arquitectura del PIC16C745/65 que indica el número de puertos de datos externos. El microcontrolador que se uso para este proyecto es el PIC16C745 debido a que cumple con los requerimientos necesarios y es más económico que el PIC16C765. El PIC16C745 cuenta con 3 puertos cada uno de 8 bits (Puerto A, Puerto B y Puerto C).

En la tabla 1-3 del capítulo 1, se muestra que se requieren 26 líneas de datos, ya sea de entradas o salidas sin contar las que faltan para incluir los sensores extras y el resto de los motores a pasos del manipulador, que no están contemplados en esa tabla. Eso aunado a que se requiere usar el convertidor analógico digital del PIC16C745 y que cada canal es una línea de datos que se utiliza del puerto A. De esta manera, la única alternativa que se tiene es expandir algún puerto de datos del PIC y conectar cada puerto añadido con un bus de datos. Para esto se requiere un selector de puertos, que evitará colisiones de datos dentro del bus y sobre todo para que cada puerto sea independiente.

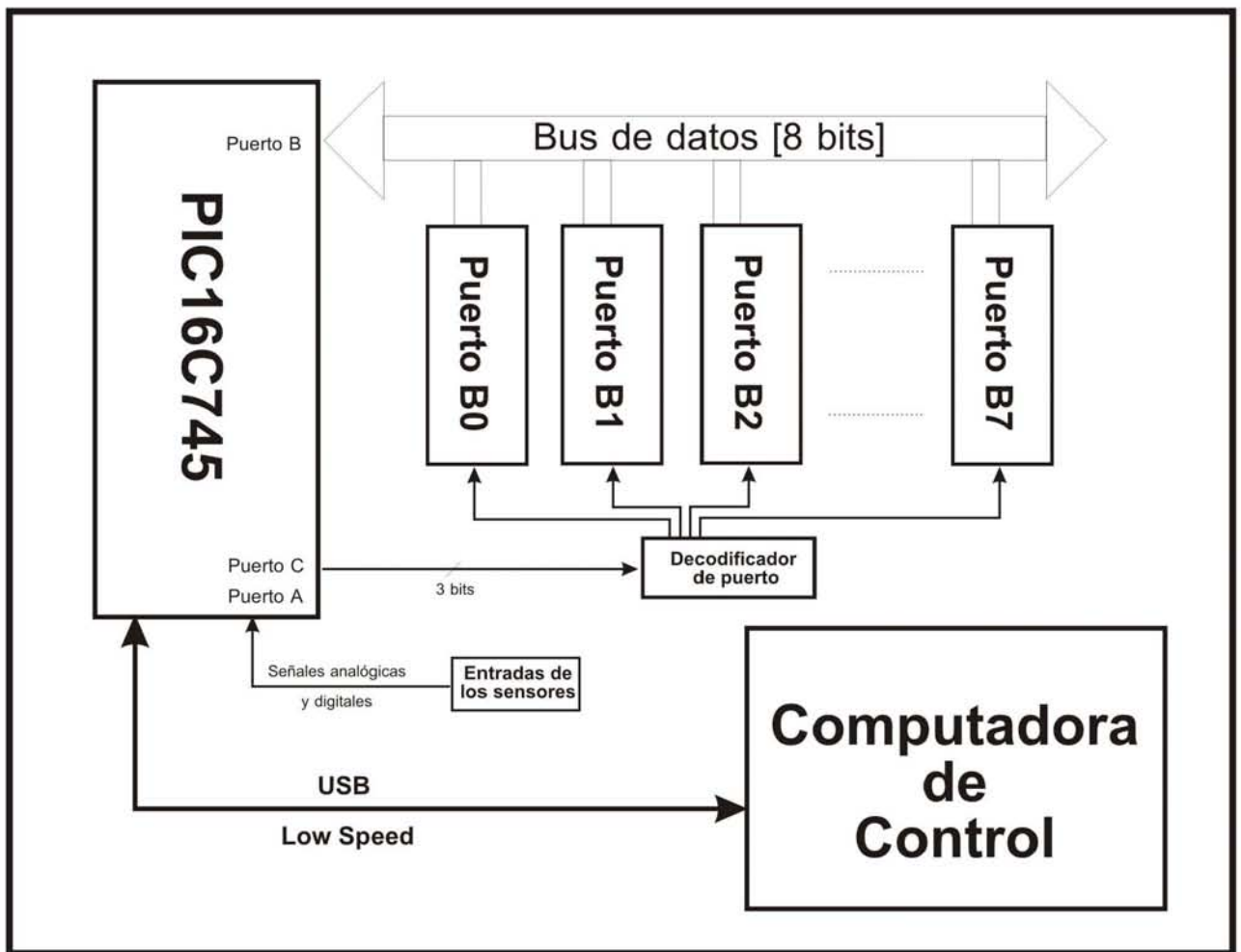


Figura 3-7 Esquema de la TEP

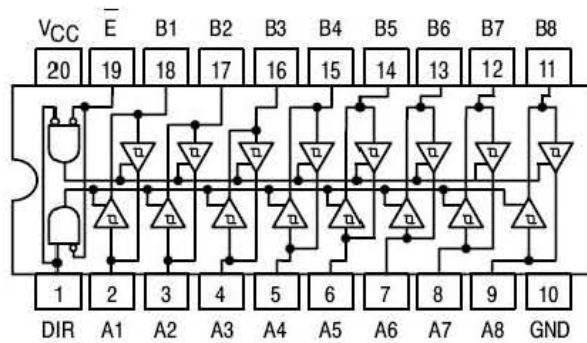


Figura 3-8 *Buffer* de 8 bits 74LS245

Para conseguir esto, se optó por utilizar un bus bidireccional que pudiera multiplexar los datos digitales desde la fuente (el PIC16C745) hacia todos los puertos, pero solo uno leerá esos datos y los hará pasar a través de él, para entregarlos a su destino (etapas de potencia, compás electrónico, etc.). Para que solo un puerto lea los datos del bus es necesario que un circuito lo seleccione (decodificador de puerto). Este circuito se comanda desde el programa de control de la TEP; lo mismo ocurre en dirección contraria. Cuando hay datos en un puerto provenientes de un sensor, el decodificador de puerto lo selecciona y el PIC16C745 lee el dato del bus, (ver figura 3-7).

Para conectar el bus de datos con el PIC16C745 se utilizó el puerto B, esto se debe a que este puerto es de 8 bits y no tiene conectado internamente algún otro periférico. Cada puerto de la TEP está conformado por un *buffer*. El circuito integrado que se usó para cada *buffer* es el 74LS245 (ver figura 3-8). Este circuito es un *buffer* de 8 bits con tres estados en sus salidas y niveles TTL. Tiene un selector de dirección para definir en qué sentido fluyen los datos, si de A a B o de B a A. Además, tiene una línea de habilitación a nivel bajo, que al aplicarle 5 V, sus salidas entran en alta impedancia. Este *buffer* aísla eléctricamente el bus de datos con los dispositivos conectados a la TEP (etapas de potencia, sensores).

La lógica de control para seleccionar el *buffer* que va a transmitir se hace a través de un decodificador de 3 a 8 bits 74LS138 (ver figura 3-9). Debido a que solo hay disponibles 3 bits del puerto C (RC.0, RC.1 y RC.2) del PIC16C745 se usan estas líneas para seleccionar el puerto PBX a través del decodificador. Así, cuando en las entradas del decodificador A0, A1 y A2 tienen ceros, se

selecciona la línea O0 que a su vez habilita la línea E del *buffer* 74LS245. De esta manera, se selecciona cada puerto con solo variar A0, A1 y A2 desde #000b hasta #111b (0 a 7 en binario). Para habilitar este circuito los pines E1 y E2 se conectan a tierra y E3 a V_{CC}.

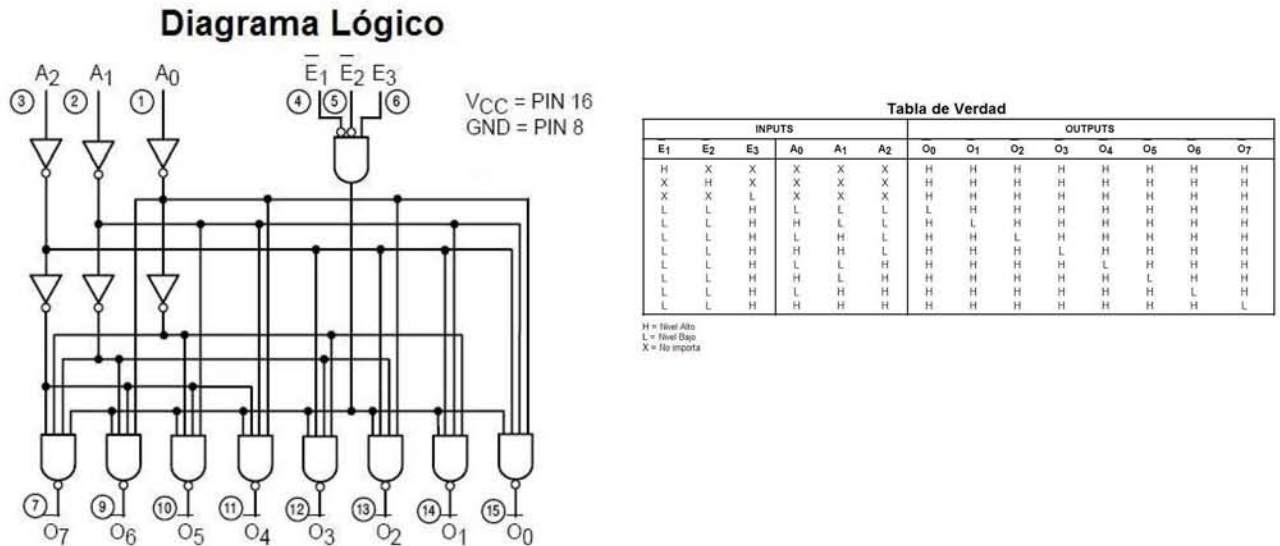


Figura 3-9 Decodificador 74LS138

Los pines del puerto A del PIC16C745 conforman los canales para el convertidor analógico-digital, excepto el pin RA.4, que se utiliza para recibir la señal de salida del sensor maestro de ultrasonido. El resto de los pines del puerto A (RA.0, RA.1, RA.2, RA.3 y RA.5) se reservaron para conectar los sensores que producen señales analógicas. Por parte del puerto C del PIC16C745 quedan libres los pines RC.6 y RC.7 que comparten la conexión con la USART (*Universal Serial Synchronous Asynchronous Receiver Transmitter*). Probablemente este circuito tenga que conectarse con otro tipo de brazo manipulador (en desarrollo, IIMAS-SEA), que utiliza una USART para comunicarse. En la tabla 3-6 se muestra la relación de la disposición y asignación de los puertos de la tarjeta y del propio PIC.

Para generar el circuito de la TEP (figura 3-10), se utilizó el programa PROTEL DXP 2004. El diagrama PCB de esta tarjeta se muestra en el Apéndice B. En el extremo derecho del circuito aparecen los 8 *buffer* conectados a través del bus que conforman los puertos. Los primeros 4 puertos

(PB0 – PB3) tienen ya establecida la dirección del flujo de datos (de A a B, es decir del bus hacia los conectores). A diferencia de los 4 puertos restantes (PB4 – PB7) que están conectados con un *pull-up* y un *Dip-switch* para seleccionar el sentido del flujo de datos.

Para seleccionar el flujo de datos de A hacia B se debe colocar cada switch en I y para que el flujo de datos vaya de B a A se debe colocar cada switch en 0. Esto se hizo con la finalidad de poder reconfigurar la disposición de puertos dependiendo de las necesidades que se requieran para operar al sistema. Todos los conectores JX se conectan con las otras tarjetas, las de potencia (3), con la tarjeta del compás electrónico, con la tarjeta de alimentación y con el conector USB que va hacia la CC.

Dispositivo		# bits	Entrada o Salida	Puerto
Sensores analógicos(4 Canales)		4	Entrada	PA (RA0-RA5)
Sensor de sonar		1	Entrada	PA (RA4)
Brazo manipulador	Brazo	4	Salida	PB0
	Muñeca ₁	4	Salida	
	Muñeca ₂	4	Salida	PB1
	Codo	4	Salida	
	Base	4	Salida	PB2
	Pinza	4	Salida	
Tracción		2	Salida	PB3
Dirección		2	Salida	
Sonar, multiplexor		3	Salida	
Encendido de Tarjeta Madre		1	Salida	
Odómetros		2	Entrada	PB4
Sensor de voltaje, 12 V		1	Entrada	
Puerto I/O		8	Entrada o Salida	PB5
Puerto I/O		8	Entrada o Salida	PB6
Puerto I/O		8	Entrada o Salida	PB7
Decodificador para el bus		3 bits	Salida	PC(RC0-RC2)

Tabla 3-6 Disposición de los puertos de la TEP

El circuito (U1) es el PIC16C745 que tiene un cristal de 6 Mhz y cuenta con configuración para

- Servir de medio de conversión de un formato serie de envío de datos y convertirlos a un formato paralelo necesario para controlar los motores y los sensores.
- Adquirir datos provenientes de los sensores, convertirlos a formato digital y enviarlos a la CC.
- Control absoluto sobre todo el hardware de la SRM.

Existen dos formas de programar un PIC, en lenguaje ensamblador y en lenguaje C, que éste último a final de cuentas traduce el código de alto nivel a lenguaje ensamblador. Desarrollar el software en C resulta muy conveniente para programar dispositivos que requieren de gran cantidad y complejidad de funciones, como es el caso de programar un dispositivo HID USB. Para esto es necesario contar con las librerías adecuadas para su programación, un compilador, un ensamblador y un enlazador. Sin embargo el fabricante de estos componentes, Microchip, no ofrece de forma gratuita las librerías de código C para programar dispositivos HID USB con el PIC16C745/65, por lo que complica el desarrollo de esta aplicación. Sin embargo sí ofrece librerías para ensamblador para dicho PIC.

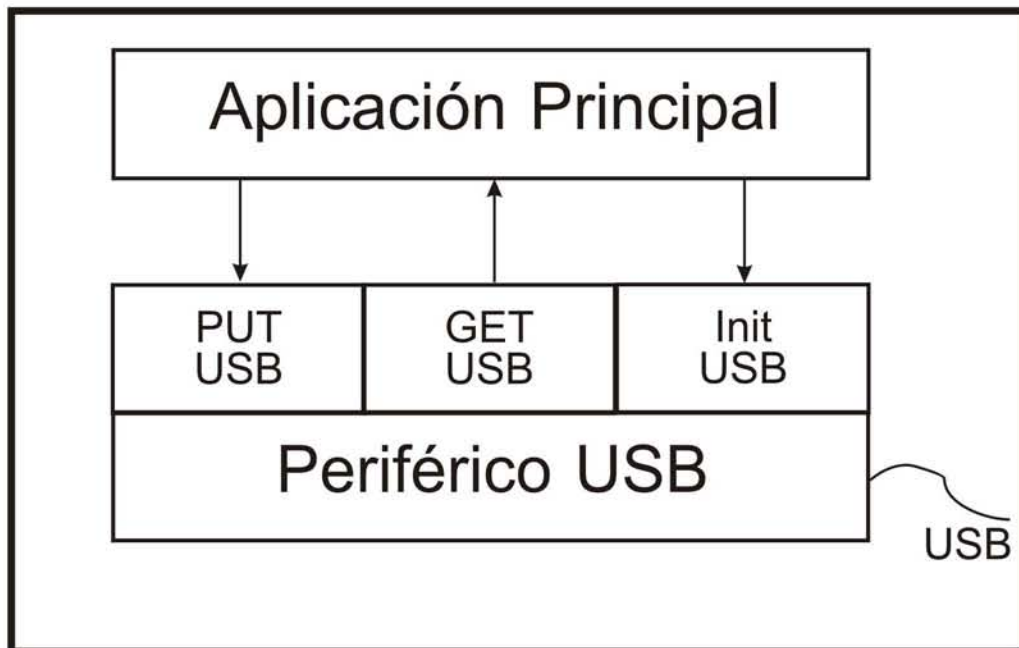


Figura 3-11 Esquema de la interfaz USB en el PIC16C745

Así, todo el software que controla la TEP fue desarrollado usando lenguaje ensamblador auxiliado

de librerías programadas en el mismo formato. El ensamblador que se utilizó fue el programa MPASM, el enlazador MPLINK y para simular el código se utilizó MPLAB Sim. En general se utilizó una IDE (*Integrated development environment*) para el desarrollo de la aplicación MPLAB IDE v7.10 de Microchip.

El esquema de programación de la aplicación HID USB en el PIC16C745 establece 3 funciones InitUSB, GETUSB y PUTUSB (ver figura 3-11). InitUSB sirve para inicializar al dispositivo y permite al anfitrión enumerarlo. GETUSB recibe datos desde el anfitrión y PUTUSB envía datos hacia el anfitrión. Las herramientas que proporciona Microchip de forma libre para la implementación de dispositivos HID USB incluye 4 archivos. USB_CH9.ASM, DESCRIPT.ASM, HIDCLASS.ASM y USB_MAIN.ASM. A continuación se describe la función de cada uno de ellos. El listado del código de cada archivo se muestra en el Apéndice C.

- **USB_CH9.ASM.** En este archivo se incluye las funciones de la interfaz y en general, del núcleo de programación necesario para la enumeración del bus.
- **DESCRIPT.ASM.** Contiene los descriptores del dispositivo, de configuración, de interfaz, de *endpoint* y el descriptor de secuencias.
- **HIDCLASS.ASM.** Provee algunas especificaciones de HID CLASS.
- **USB_MAIN.ASM.** Es un ejemplo para el programa principal de la aplicación. Incluye los procedimientos para iniciar el USB, obtener y enviar datos.

Para ensamblar estos cuatro archivos se requiere incluirlos en un proyecto de MPLAB IDE (ver figura 3-12). El archivo USB_MAIN.ASM fue renombrado por PRUEBA12.ASM que es una versión final del software de la TEP, sin embargo conserva los elementos esenciales del archivo USB_MAIN.ASM con algunas modificaciones para ejecutar las rutinas de control de los motores y de los sensores.

El programa prueba12.asm tiene el control de la TEP. En la figura 3-13 se muestra el diagrama de

flujo del software de control. Este software comienza configurando el USB, entabla comunicación con la CC, carga variables de inicio y configura el compás. Posteriormente espera a que el usuario, a través del software ControlSRM, le envíe ordenes. Estas ordenes pueden ser leer compás, leer odómetro, leer puertos analógicos o bien puede recibir la orden de escribir datos en los puertos, así como mover los motores del manipulador, el de tracción ó el de dirección. Si no hay datos en el *endpoint* del USB HID, el PIC continua en espera hasta un nuevo evento.

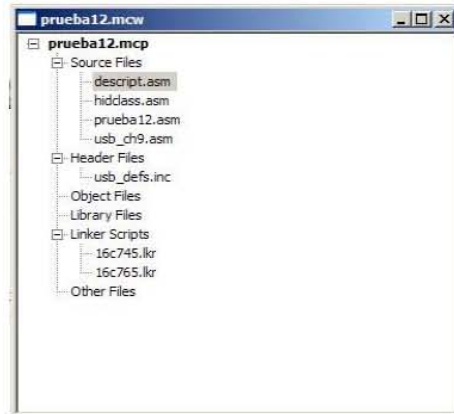


Figura 3-12 Ventana de administración del proyecto en MPLAB IDE

Para enviar los mensajes de control o de datos se utilizan dos *buffers* de 8 bits cada uno tanto para la entrada como para la salida de datos. El núcleo USB del PIC soporta el envío de 8 *buffers* de 8 bits, sin embargo, para esta aplicación no es necesario usar todo. El *buffer* de entrada se llama “outbuffer” y consta de dos bytes. El primero para designar el comando y el segundo contiene los datos. En el comando se envían dos informaciones, los primeros 4 bits de mayor valor significativo, le indican al PIC que debe realizar, es decir que rutina ejecutar. Las rutinas son tres, leer datos del compás, leer datos de los odómetros o enviar datos a los puertos PBx. Los 4 bits restantes, los de menor valor significativo, son utilizados para direccionar un puerto PBx, en el caso de que se ejecute la rutina de enviar datos a PBx.

Cuando se ordena ejecutar las rutinas de lecturas de datos el segundo byte de “outbuffer” va vacío. Y para enviar los datos leídos por el PIC, provenientes de los sensores, se utiliza un *buffer* de 2 bytes llamado “sensor”.

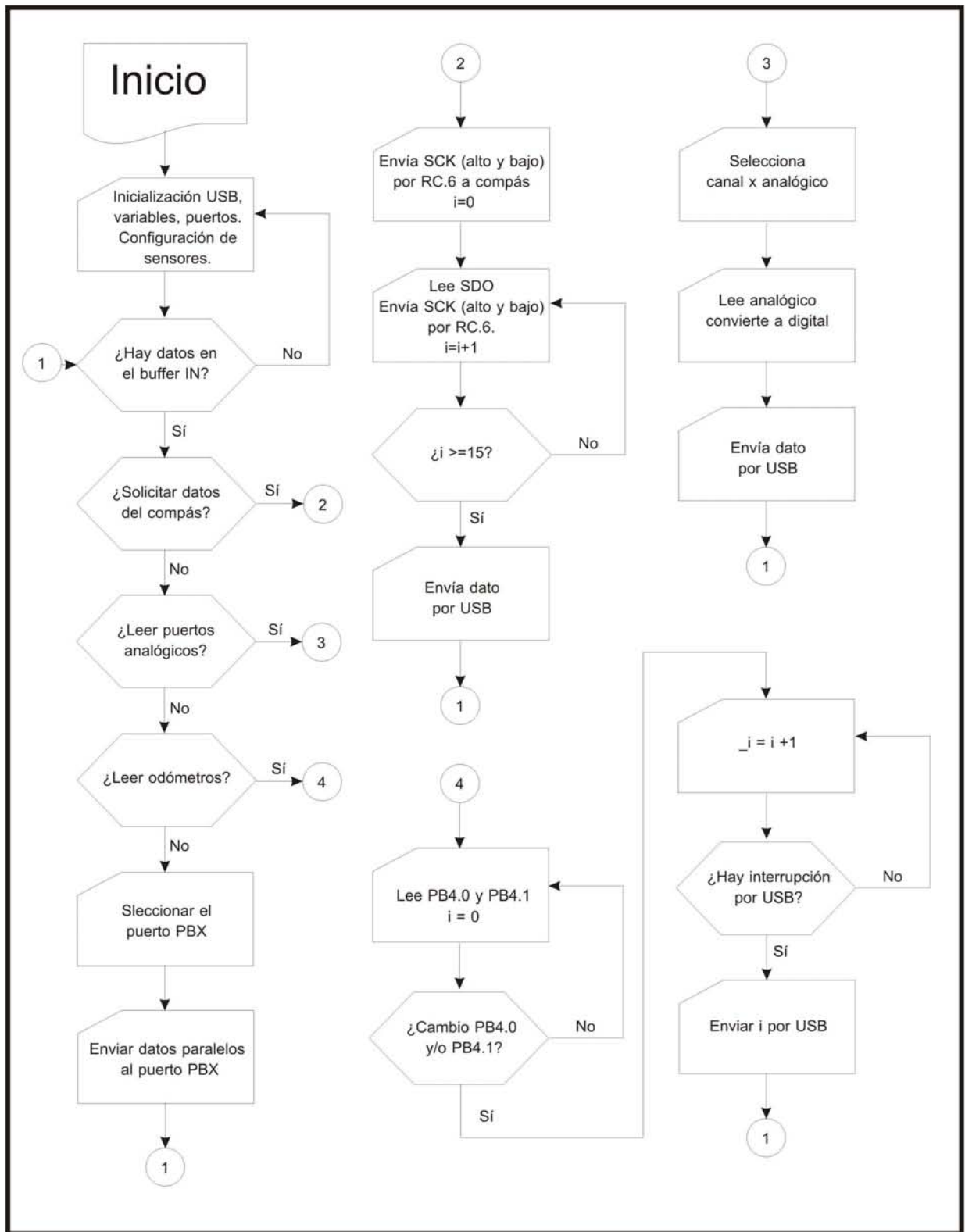


Figura 3-13 Diagrama de flujo del programa de control de la TEP.

3.4 Software de control de la Sonda Robot Móvil. ControlSRM y TightVNC

El software de control de la sonda tiene como objetivos:

- Interactuar con el usuario.
- Conectarse con el PIC16C745 a través del bus USB y establecer las negociaciones pertinentes para el envío y recepción de datos.
- Enviar los comandos hacia la TEP para el control de todos los motores del manipulador y los de tracción y dirección.
- Enviar los comandos hacia la TEP para el control de los transmisores ultrasónicos.
- Desplegar el estado de los sensores y de sus variables digitales o análogas que entregan a la TEP.
- Administrar y controlar los recursos de la SRM vía remota desde una Estación de Control.

Los primeros 5 puntos los realiza el programa ControlSRM y para el último se cuenta con el software de servicio TightVNC.

3.4.1 ControlSRM

Con este fin fue necesario realizar la programación en Visual Basic 6.0, por ser el lenguaje que se utilizó y que se está utilizando para desarrollar los proyectos paralelos a este. Fue necesaria la integración de una librería dinámica que tiene la función de comunicarse con los dispositivos HID USB, en este caso la TEP. Esta librería lleva por nombre HIDCOMM.dll e incluye un componente para agregarlo como objeto en el proyecto de Visual Basic 6.0.

La operación de esta librería es muy sencilla y hay que incluir el componente en una forma de Visual Basic. Algunos métodos de este componente se muestran en la tabla 3-7.

<i>Método</i>	<i>Función</i>	<i>Argumentos</i>
HIDComm.connect	Conectarse a un dispositivo HID	Ninguno
HIDComm.WriteTo(Buffer, Tamaño)	Escribe datos en un dispositivo HID	Buffer y Tamaño
HIDComm.ReadFrom(Tamaño)	Lee datos de un dispositivo HID	Tamaño
HIDComm.GetFeature() As Byte()	Obtiene las características de un dispositivo HID	Ninguno
HIDComm.Uninit()	Descargar de memoria un dispositivo HID	Ninguno

Tabla 3-7 Métodos del control ActiveX HIDComm

El código fuente del software ControlSRM se transcribe en el Apéndice C. En la figura 3-14 se muestra la consola de control del ControlSRM. En esta consola aparecen varios módulos que corresponden a cada motor y a cada sensor.

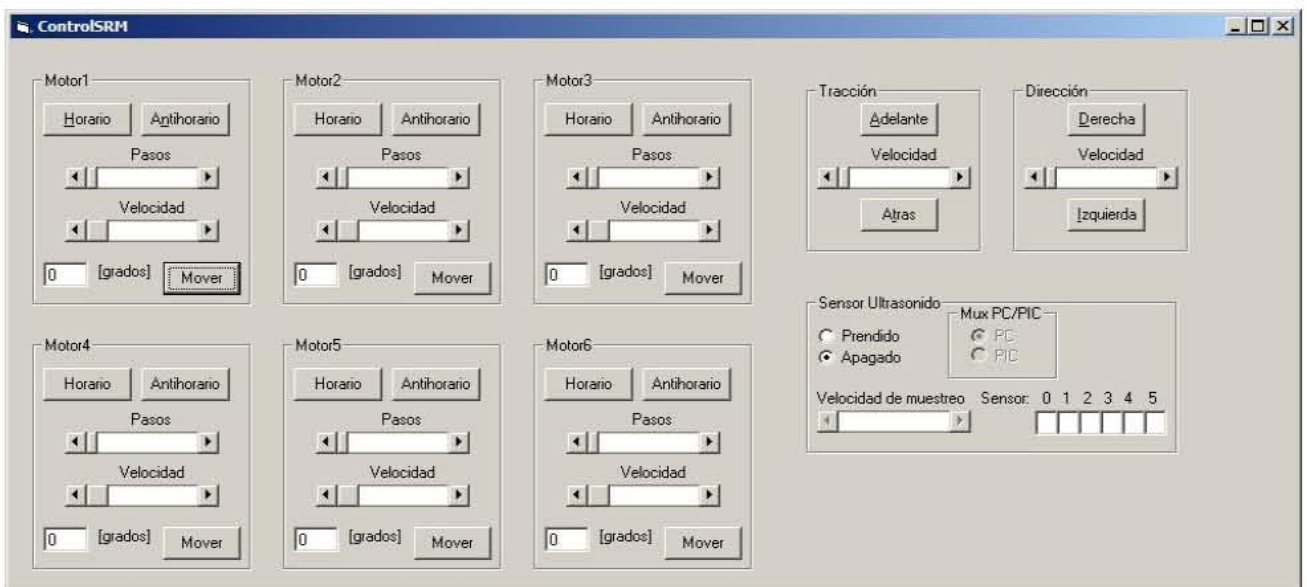


Figura 3-14 Consola del programa ControlSRM.

En cada módulo “Motor x” aparecen 3 botones, una caja de texto y dos barras desplazadoras. Estos módulos corresponden a los motores de pasos del manipulador. Con los botones “Horario” y “Antihorario” se realiza una secuencia de pasos en su respectiva dirección. Cada secuencia de pasos está determinada por la barra desplazadora titulada “Pasos”. Así, si la barra está al principio del total de su recorrido, se estará ejecutando un paso por cada vez que se presione el botón Horario o

“Antihorario”. Si la barra está a la mitad del total de su recorrido, se estarán ejecutando 50 pasos por cada vez que se presione el botón “Horario” o “Antihorario”. La barra “Velocidad” determina el tiempo que tarda en ejecutarse cada paso. Por último, al presionar el botón “Mover”, el motor se moverá a la posición indicada en la caja de texto. Este valor está en grados.

Para el caso de los módulos Tracción y Dirección cada uno tiene 2 botones y una barra desplazadora. De igual forma que en el caso de los módulos de los motores del manipulador, esta barra determina la velocidad para cada motor de corriente directa. Los botones de “Adelante”, “atrás”, “Derecha” e “Izquierda” operan los motores de tracción y dirección respectivamente.

La operación de los sensores ultrasónicos se realiza en el módulo “Sensor Ultrasonido”. Para iniciar la selección de transmisores ultrasónicos hay que seleccionar la opción de “Prendido”. Una vez realizado esto, empezará a desplegarse un “1” o un “0” en cada una de las seis cajas de texto numeradas desde el cero hasta el cinco. Si aparece un “1” en alguna caja, significa que ese sensor detecta algún objeto, en cambio si aparece un “0” representa que no hay objetos cercanos a ese sensor. Se puede seleccionar el tiempo de muestreo con la barra “Velocidad de muestreo”. También puede direccionarse cada transmisor desde el programa que se ejecuta en el PIC, esto se hace seleccionando la opción “PIC” del módulo “MUX PC/PIC”.

3.4.1 TightVNC

En la sección 2.5 del capítulo anterior se expone la necesidad de controlar esta sonda a través de un software de conexión, que convierte a la SRM en un servidor de sesiones para que en otra computadora, cliente, se inicie una sesión remota. Este software se considera un servicio ya que no depende de la actividad del usuario, sino de lo que realiza en sí la computadora, al igual que el servicio de impresión. Existen al menos 4 software de este tipo. En la tabla 3-8 se realiza una comparación entre los distintos software.

Nombre del software	S.O. Soportados		Última versión	Licencia
	Servidor	Cliente (viewer)		
RealVNC	Unix, Win32, Linux, Mac, GGI y RiscOS.	Unix, Win32, Mac, GGI, RiscOS, PalmPilot, BeOS, Java, WinCE, DOS, OS/2, SVGALib y Geos	4.0	• Edición libre. GPL (reducida) • Edición <i>Enterprise</i> privada
TightVNC	Unix, Win32, Linux.	Unix, Win32, Linux y Java.	• Estable 1.2.9 • Desarrollo 1.3	GPL
OSXVNC	Mac	No tiene	1.7	GPL
UltraVNC	Win32	Win32, Linux, Mac	1.1	GPL
VNCScan	Win32	Win32	2006.1.5	Privado

Tabla 3-8 Versiones VNC

En la tabla 3-8 se ilustra que la mejor opción para utilizar un VNC es el software TightVNC; ya que dispone de versiones para la mayoría de los sistemas operativos y tiene licencia GPL (*General Public License*).

Cada software VNC consiste en dos paquetes principales, el cliente (*viewer*) y el servidor. En la SRM se instaló el servidor. En general, basta con instalar el software y configurar la contraseña de acceso. Debido a que se trata de un servicio, este se carga al iniciar Windows. En la figura 3-15 se muestra el icono en la barra de herramientas de Windows 2000, instalado en la CC de la SRM.



Figura 3-15 Icono de VNC (círculo rojo)

En la figura 3-16 se muestran dos pruebas del servicio VNC. Estas pruebas fueron realizadas en la computadora de la sonda, e indican la carga de trabajo que ejecuta el procesador central (CPU) de la tarjeta Epia TC 10000E. En la figura se gráfica el tiempo de proceso contra el uso del procesador, cuando se está ejecutando el VNC servidor (CC de la SRM) y se ha realizado una conexión desde el VNC cliente (Estación de control). El círculo “1” se trata de una conexión con rápida compresión de datos, donde se observa que el tamaño en las abscisas es considerable y con picos de gran magnitud,

que indican el inicio de la conexión. El círculo “2” tiene una mejor compresión de datos. El círculo “3” simplemente hace una petición de conexión por parte del cliente, pero esta ha sido rechazada.

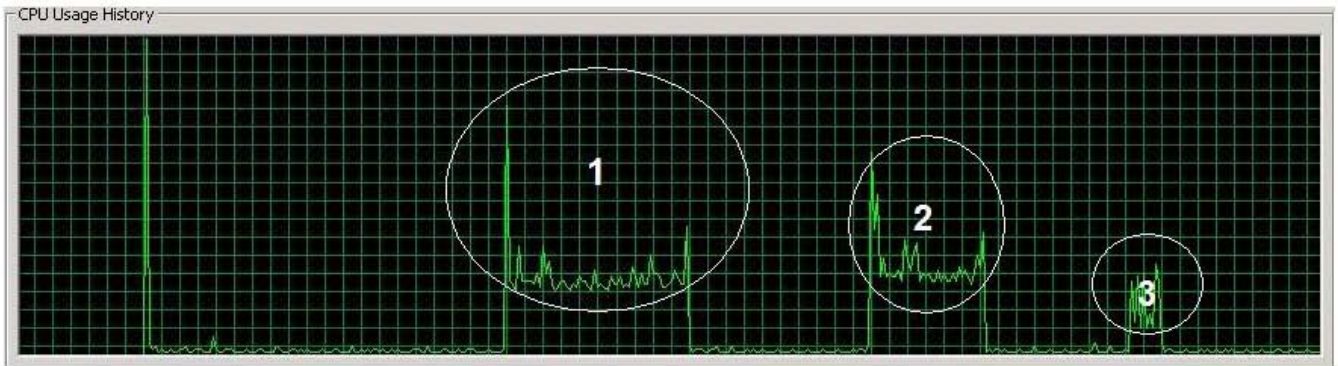


Figura 3-16 Uso del CPU

3.5 Estación de Control.

La estación de control se utiliza para comandar a la SRM remotamente. Se utiliza cualquier computadora de escritorio, estación de trabajo, servidor y otros. Que debe cumplir con dos requisitos. El primero, es que su sistema operativo sea compatible con cualquier cliente (*viewer*) VNC (tabla 3-8) y el segundo, es que cuente con un enlace inalámbrico compatible con el estándar 802.11g.

Para cuestiones de pruebas se utilizó una laptop Compaq V139 con una tarjeta *wireless* Broadcom 802.11g/b con sistema operativo Windows XP.

Se realiza la conexión inalámbrica con la SRM; una vez conseguido el enlace se ejecuta el software TightVNC cliente (figura 3.18-a) y se teclea la dirección IP de la SRM. El servidor le solicita al usuario de la computadora cliente, una contraseña para autenticarse (figura 3.18-b) y si es válida dicha contraseña es válida se carga una sesión en la computadora de la Estación de Control (figura 3.18-c y d)

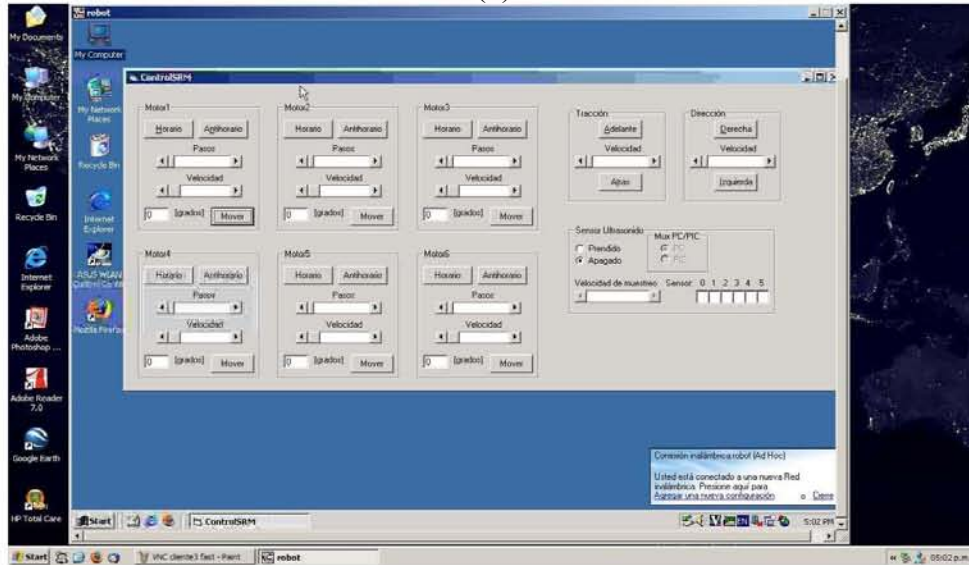
Estos elementos integran el diseño final de la actualización a la SRM.



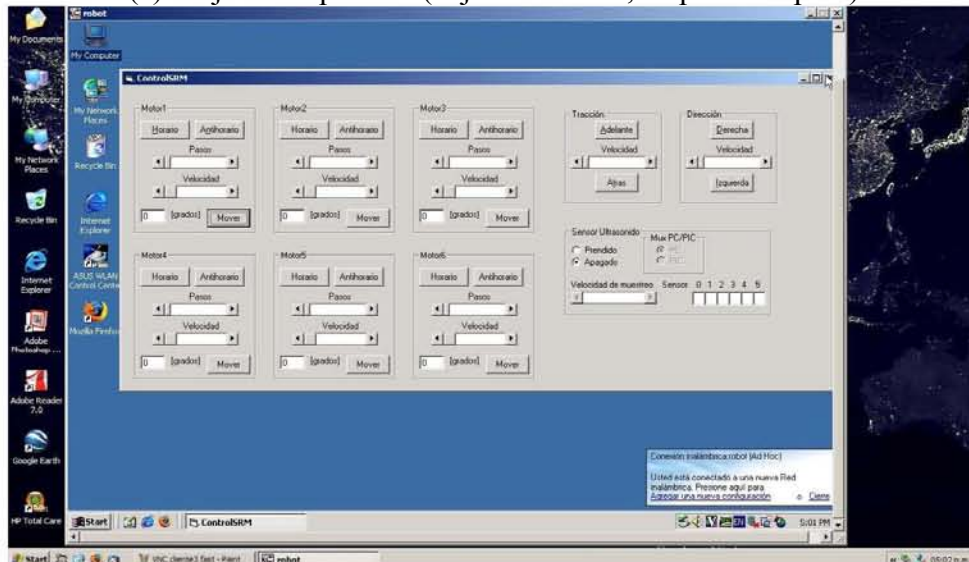
(a)



(b)



(c) Mejor Compresión (baja resolución, respuesta rápida)



(d) Compresión rápida (Alta resolución, respuesta lenta)
 Figura 3-18 (a,b,c y d) Software en la Estación de Control.

CONCLUSIONES

La Sonda Robot Móvil constituye un proyecto que ha tenido diversas fases para el mejoramiento del mismo. Desde el principio, este proyecto tenía objetivos muy precisos que conllevarían a tener un robot competitivo, funcional y que en algún momento se pudiese comercializar para que fuera operado en labores de rescate, exploración e investigación. Sin embargo, la tecnología que imperaba en el tiempo de su creación, hacia que las mejoras tecnológicas limitaran su funcionamiento. La actualización que aquí ha sido reportada proporciona las pautas para que a partir de ésta y durante los próximos años se disponga de una estructura sólida, que permita al grupo de investigación del laboratorio, el poder diseñar, probar, implementar y validar nuevas tecnologías de control, inteligencia artificial, procesamiento de voz y audio, eficiencia energética y otras

Esta actualización consistió en la integración de una plataforma de cómputo, compatible con la arquitectura x86, a un robot móvil, con lo cual se lograron ventajas como son la capacidad de almacenar grandes cantidades de datos en disco duro, desarrollar y ejecutar programas en lenguaje de alto nivel, obtener las ventajas que proporciona un sistema operativo como Windows y las de contar con un procesador de propósito general de alta escala de integración compatible con la familia Pentium, el utilizar memoria RAM del tipo DDR de alta velocidad de lectura y escritura, así como de gran capacidad de almacenamiento y por último, la implementación de una tarjeta de entrada y salida de datos a través del USB.

Con el uso del puerto USB se eliminó la necesidad de adquirir tarjetas de adquisición de datos a través del bus ISA o PCI, así mismo se mejoraron las capacidades *plug-and-play*, permitiendo con esto poder conectar o desconectar el dispositivo del sistema sin necesidad de reiniciar. Otra ventaja, es que otorga la facilidad de acomodar (por medio de su cableado y de sus dimensiones física) en un espacio reducido dentro del mismo robot. El USB se ha convertido en el método universal de conexión de periféricos, que incluye a los dispositivos de almacenamiento, adaptadores de red inalámbricos o no, y los dispositivos denominados HID (*Human Interface Device*) que principalmente son los ratones, teclados, palancas de juego y otros.

En la tabla (1-3) se muestran diferentes robots disponibles en el mercado que bien pueden compararse con este proyecto, ya que han integrado a sus diseños un sistema de cómputo similar. Basta con ver la columna de precios para darse cuenta que esta sonda, además de tener ventajas tecnológicas, resultará mucho más barata que el resto. Este hecho, representa un valor agregado a la SRM. Es decir, con menor presupuesto se consiguió desarrollar un robot móvil con algunas de las capacidades iguales o mejores que su competencia.

Si bien se alcanzaron los objetivos, surgieron inquietudes para favorecer de mejor manera esta actualización. Ya que parte de la experiencia que se adquirió al desarrollar este proyecto, ayudó a analizar la eficiencia del sistema operativo y en general del software que se utilizó. Un inconveniente que tiene el sistema operativo Windows, es que al adaptar el dispositivo USB (PIC16C745) se presentaron dificultades para programarlo, esto se debe a que el desarrollo de software depende de los controladores programados por Microchip; y en general construir adaptadores de software que controlen cualquier dispositivo físico para transferir datos de la computadora hacia el dispositivo, implica conocer la estructura misma del núcleo de Windows. Esto representa un problema dado que Windows es un software privado que no distribuye el código y que ha impuesto políticas a los fabricantes de hardware para que todo dispositivo cumpla con las normas de Windows y limitando la fabricación de programas y circuitos por parte de fabricantes independientes, con la intención de privilegiar a las empresas en las que Microsoft tiene intereses.

Aún cuando se tuviera el *know how* de programar controladores en Windows, las herramientas para el desarrollo de software, en esta plataforma, requieren del pago de una licencia que imposibilita cualquier desarrollo independiente; por lo que la opción más viable a utilizar es GNU/ Linux, ya que parte del gran éxito que tiene este sistema en el mundo de los desarrolladores, es el hecho de que su filosofía se basa en compartir libremente el código fuente del *kernel*, de todos los programas y de todos los controladores, además de las herramientas de programación. Esto facilita la programación de dispositivos conectados a la computadora. Además, GNU/Linux es un sistema multiusuario, multiproceso, y cuenta con distintos programas y/o servicios que facilitan la administración de una computadora vía remota, situación relevante para el desarrollo del proyecto que aquí se presenta. En cuanto a las ventajas que presenta al programar dispositivos externos, este sistema operativo es

superior a Windows ya que se tiene una estructura modular y es posible prescindir de aquellos módulos no necesarios para el proyecto en particular, obteniendo con esto mejores resultados al aprovechar óptimamente los recursos de la SRM. A diferencia de Windows, en GNU/Linux el sistema de ventanas y el núcleo del sistema son módulos independientes con lo que el hardware no tendrá que hacer uso de la totalidad de los recursos disponibles para el manejo de ventanas y gráficos, ya que no son necesarios para este proyecto y solo es necesario el *kernel* del sistema, que se encarga de administrar el hardware. Sin embargo GNU/Linux no fue utilizado debido a que gran parte del software que la SRM utiliza habían sido desarrollados con anterioridad, para versiones de Windows.

Por otro lado es necesario mencionar que el microcontrolador que se utilizó (PIC16C745) es de memoria EPROM, lo que dificultó demasiado la depuración del *firmware* al desarrollar el proyecto. Ya que como es sabido, es necesario utilizar durante aproximadamente 15 minutos una fuente de luz ultravioleta (UV) para borrar las localidades de dicha memoria para poder programar de nuevo al microcontrolador. Con lo que se retrasó el desarrollo del software que finalmente porta el dispositivo USB, cosa que no hubiera ocurrido si se utilizase un microcontrolador con otro tipo de tecnología como es la *flash* ROM. Existe un nuevo microcontrolador de la misma empresa Microchip PIC18F4550 que por sus características de velocidad, espacio en RAM y disposición de memoria de programa del tipo flash, se concluye que sería una buena opción trabajar con este dispositivo.

BIBLIOGRAFÍA

- AXELSON, Jan. *USB Complete, Everything You Need to Develop Custom USB Peripherals*. Segunda Edición. Madison, EUA: Lakeview Research. 2001. ISBN 0-9650819-5-8.
- ARTEAGA, Alma. *Desarrollo del software de un sistema para la operación de un robot móvil en un ambiente de realidad virtual 3D*. México: Tesis de Licenciatura, UNAM. 2002.
- BREY, B. *Los Microprocesadores Intel, Arquitectura, Programación e Interfaces*. México: Prentice Hall. 1994.
- GARNEY, John et al. *USB Hardware & Software*. California, EUA: Annabooks. 1998. ISBN 0-929392-37-X.
- GONZALEZ, Luis. *Seguridad de redes inalámbricas en un ambiente académico*. México: Tesis de Licenciatura, UNAM. 2005.
- HERNANDEZ, Luis. *Percepción de un Robot Móvil durante su navegación para evadir obstáculos*. México: Tesis de Licenciatura, UNAM. 2002.
- HYDE, John. *USB Design by Example, A Practical Guide to Building I/O Devices*. Segunda Edición. EUA: INTEL PRESS. 2001 ISBN 0-9702846-5-9.
- INTEL, *ATX Specifications*, Intel, 2002
- MICROCHIP Data Sheet DS41124C. PIC16C745/765. Microchip Technology Incorporated. 2000
- TANENBAUM, A. *Organización de computadoras un enfoque estructurado*. México: Pearson Educación. 2000.
- TORRES, J. *Conceptos de sistemas operativos*. México: Trillas. 2001.
- VIA. *Users's Manual EPIA TC Mini ITX Mainboard*, VIA Technologies Inc., 2004

Enlaces electrónicos

- <http://www.thocp.net/reference/robotics/robotics.html>
- <http://asimo.honda.com/index.asp?bhcp=1>
- <http://www.jp.aibo.com/>
- <http://www.fira.net/>
- <http://www.solarviews.com/span/craft2.htm>
- <http://www.solarviews.com/span/rover.htm>
- <http://www.ifr.org/statistics/keyData2005.htm>
- <http://www.intel.com/design/mcs51/datashts/27233504.pdf>
- http://es.wikipedia.org/wiki/Sistema_operativo
- http://es.wikipedia.org/wiki/Microsoft_Windows
- http://es.wikipedia.org/wiki/IBM_PC
- <http://en.wikipedia.org/wiki/Teleoperator>
- <http://www.tightvnc.com/>
- <http://www.aboutai.net/DesktopDefault.aspx?article=aa090600a.htm&tabid=2>
- http://en.wikipedia.org/wiki/Virtual_Network_Computing

Apéndice A

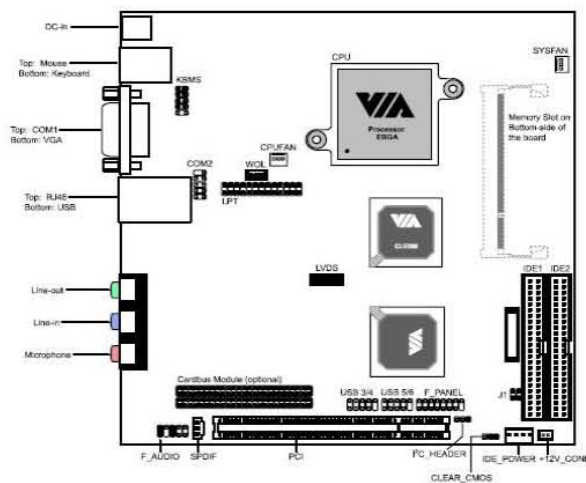
➤ Especificaciones de la Tarjeta Madre EPIA TC10000E

<i>Componente</i>	<i>Descripción</i>
Procesador	<ul style="list-style-type: none"> VIA C3/VIA Eden ESP
Chipset	<ul style="list-style-type: none"> VIA CLE266 North Bridge VIA VT8235 South Bridge
Memoria RAM	<ul style="list-style-type: none"> 1 x 200-pin DDR266 SODIMM socket Arriba de 1 GB
Conectores IDE integrado	<ul style="list-style-type: none"> 2 x UltraDMA 133/100/66 Direct DOM
CardBus/CompactFlash integrados (opcionales)	<ul style="list-style-type: none"> CardBus Tipo I y II Controlador RicohR5C476 CardBus
Conectores del panel trasero	<ul style="list-style-type: none"> 1 entrada DC 1 puerto PS2 para ratón 1 puerto PS2 para teclado 1 puerto para LAN RJ-45 1 puerto serie 2 puertos USB versión 2.0 1 puerto para VGA 3 conectores de audio (entrada de línea, salida de línea y entrada de micrófono)
Conectores integrados a la tarjeta madre	<ul style="list-style-type: none"> 4 puertos USB versión 2.0 1 conector para audio frontal 1 conector para puerto serie 1 conector para puerto paralelo 1 conector para CIR 1 conector para Wake-on-LAN 2 conectores para ventilador (CPU y sistema) 1 modulo LVDS 1 conector para 12 V_{DC}
VGA	<ul style="list-style-type: none"> Controlador de video integrado a la placa madre modelo VIA UniChrome 2D/3D, con acelerador para MPGE-2
Ranuras de expansión	<ul style="list-style-type: none"> 1 PCI
Tarjeta de red integrada a la placa madre	<ul style="list-style-type: none"> Controlador de red modelo VIA Networking Tahoe VT6103 Fast Ethernet 10/100
Tarjeta de sonido integrada a la placa	<ul style="list-style-type: none"> Controlador de audio modelo VIA VT1612A 2-canales

<i>Componente</i>	<i>Descripción</i>
madre	AC'97
Sistema de monitoreo y administración	<ul style="list-style-type: none"> • Analizador de voltaje del CPU • Wake-on-LAN • Keyboard-Power-on • Timer-Power-on • Administrador del sistema de potencia
Temperatura de operación	• 0 a 50°C
Humedad de operación	• 0% a 93%
Formato	• Mini ITX (4 capas de integración) 17 x 17 cm

Especificaciones del Manual de Usuario de la EPIA TC10000E. http://www.via.com.tw/download/mainboards/3/4/epia-tc_v120.pdf

➤ Disposición de los componentes en la EPIA TC10000E



Especificaciones del Manual de Usuario de la EPIA TC10000E. http://www.via.com.tw/download/mainboards/3/4/epia-tc_v120.pdf

➤ Niveles de voltaje de la EPIA TC10000E

<i>Parámetro</i>	<i>Mínimo</i>	<i>Típico</i>	<i>Máximo</i>
Voltaje de entrada	11.4 V	12 V	12.6 V
Voltaje de salida	3.135 V	3.3 V	3.465 V
	4.75 V	5 V	5.25 V
	10.8 V	12 V	13.2 V

Voltajes de operación de la EPIA TC 10000E

➤ Especificaciones del Disco Duro Samsung MP0402H

<i>Parámetro</i>	<i>Característica</i>
Interfaz	ATA-6
Bytes por sector	512
Velocidad	5200 RPM
Temperatura de operación	5 ~ 55°C
Humedad	5 ~ 90%
Altitud de operación	-1,000 a 10,000 pies
Voltaje	+5V ±5%

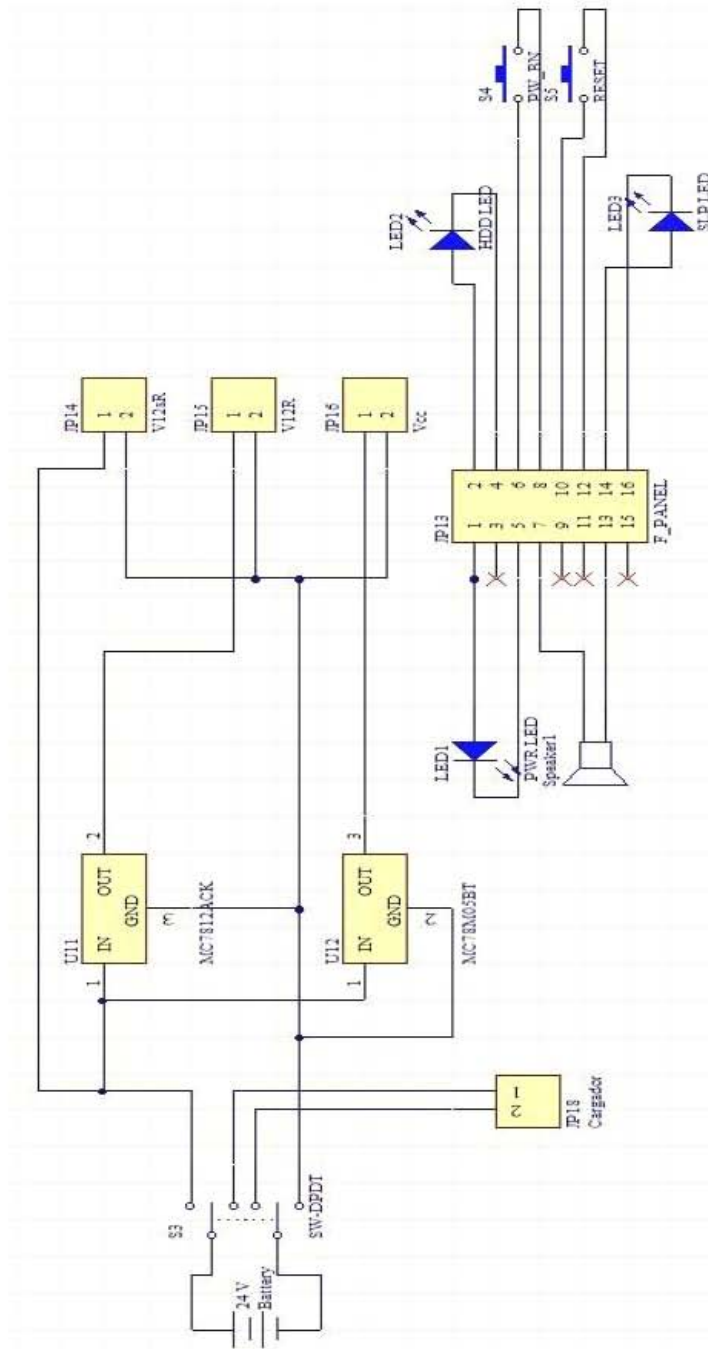
Especificaciones del Manual de Usuario del Disco Duro Samsung MP0402H.
http://www.samsung.com/Products/HardDiskDrive/SpinPointMSeries/HardDiskDrive_SpinpointMseries_MP0402H_sp.htm

➤ Especificaciones del módulo de red inalámbrico ASUS USB WL-167g

<i>Parámetro</i>	<i>Especificación</i>
Estándar	802.11g
Interfaz	USB
Tecnología	<i>Orthogonal Frequency Division Multiplexing (OFDM)</i> <i>Direct Sequence Spread Spectrum (DSSS)</i>
Taza de transferencia	802.11g: 54Mbps 802.11b: 11Mbps
Tipo de red	Redes establecidas con Puntos de Acceso redes "Ad-Hoc"
Banda de frecuencia	2.4 GHz ~ 2.5 GHz
Seguridad	64-bit/128-bit encriptado configurable WEP
Distancia de operación	Interiores @ 11Mbps : 40 m Interiores @ 54Mbps : 25 m Exteriores @ 11Mbps (sin obstáculos) : 310 m Exteriores @ 54Mbps (sin obstáculos) : 60 m
Compatibilidad	Compatible con dispositivos IEEE 802.11b y 802.11g
Requerimientos de potencia	Alimentado por el bus USB.
Sistemas operativos soportados	Windows® 98SE/ME/2000/XP
Software incluido en el CD-ROM de instalación	<i>Control Center</i> <i>One Touch Wizard</i> <i>Wireless Settings</i> <i>Mobile Manager</i> <i>Site Monitor</i> <i>TroubleShooting</i>

Especificaciones del Manual de Usuario del ASUS USB WL-167g. http://dlsvr01.asus.com/pub/ASUS/wireless/WL-167g/Eng_pdf_1971.zip

Circuito de la Tarjeta de Alimentación



- Lista de componentes de la Tarjeta de alimentación

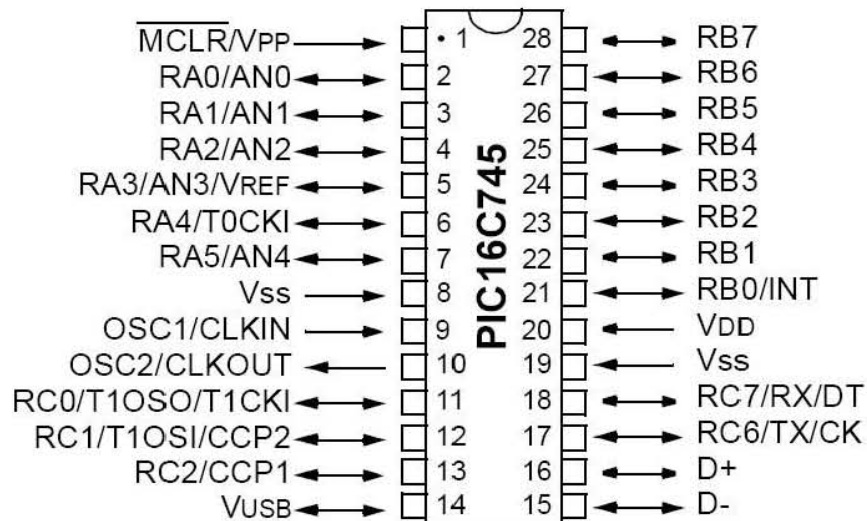
<i>Componente</i>	<i>Nombre</i>	<i>Descripción</i>
BAT	24 V	Batería
F_PANEL	JP13	Header, 8-Pin, Dual row
V12sR	JP14	Header, 2-Pin
V12R	JP15	Header, 2-Pin
Vcc	JP16	Header, 2-Pin
Cargador	JP18	Header, 2-Pin
PWR LED	LED1	Led
HDD LED	LED2	Led
SLP LED	LED3	Led
SW-DPDT	S3	Switch 2 polos, 2 tiros
PW_BN	S4	Switch
RESET	S5	Switch
Bocina	Bocina	Bocina
MC7812ACK	U11	3-Terminal Positive Fixed Voltage Regulator
MC78M05BT	U12	3-Terminal Medium-Current Positive Fixed Voltage Regulator

Apéndice B

➤ Especificaciones del PIC16C745

<i>Características</i>	<i>PIC16C745</i>	<i>PIC16C765</i>
Frecuencia de operación	6 Mhz o 24 MHz	6 Mhz o 24 MHz
Resets	POR, BOR (PWRT, OST)	POR, BOR (PWRT, OST)
Memoria de programa (palabra de 14 bits)	8K	8K
Memoria de datos	256 bytes	256 bytes
RAM	64	64
Fuentes de interrupción	11	12
Puertos I/O	22 (Puertos A, B y C)	33 (Puertos A, B, C, D y E)
Timers	3	3
Módulos de captura/compara/PWM	2	2
Convertidor A/D	5 canales de 8 bits	8 canales de 8 bits
Puerto paralelo esclavo	No	Sí
Comunicación serie	USB, USART/SCI	USB, USART/SCI

➤ Patigrama del PIC16C745



➤ Lista de componentes de la Tarjeta de Expansión de puertos

<i>Componente</i>	<i>Nombre</i>	<i>Descripción</i>
Cap	C1	Capacitor
Cap	C2	Capacitor
Cap	C3	Capacitor
USB	JP1	Header, 4-Pin
Vcc	JP2	Header, 2-Pin
AUSB	JP3	Header, 3-Pin
PB0	JP4	Header, 8-Pin
PB1	JP5	Header, 8-Pin
PB2	JP6	Header, 8-Pin
PB3	JP7	Header, 8-Pin
PB4	JP8	Header, 8-Pin
PB5	JP9	Header, 8-Pin
PB6	JP10	Header, 8-Pin
PB7	JP11	Header, 8-Pin
Serie	JP12	Header, 2-Pin
A/D sensor	JP13	Header, 6-Pin
Res 1	R1	Resistor
Res 1	R3	Resistor
Res 1	R4	Resistor
Res 1	R5	Resistor
Res 1	R6	Resistor
Res 1	R7	Resistor
SW-DIP4	S1	DIP Switch
Reset	S2	Switch
PIC16C745/JW	U1	8-Bit CMOS Microcontroller with USB, 8K (x14-Bit words) EPROM, 256 Bytes RAM, 22 I/O Pins, 28-Pin CERDIP, UV Erasable, 4.35 to 5.25V Supply Range, Industrial Temperature
MC74AC138N	U2	1-of-8 Decoder/Demultiplexer
SN74LS245N	U3	Octal Bus Transceiver
SN74LS245N	U4	Octal Bus Transceiver
SN74LS245N	U5	Octal Bus Transceiver
SN74LS245N	U6	Octal Bus Transceiver
SN74LS245N	U7	Octal Bus Transceiver
SN74LS245N	U8	Octal Bus Transceiver
SN74LS245N	U9	Octal Bus Transceiver
SN74LS245N	U10	Octal Bus Transceiver
XTAL	Y1	Crystal Oscillator

Apéndice C

➤ Listado del código del software del PIC

El código del PIC16C745 se compone de 4 archivos USB_CH9.ASM, DESCRIPT.ASM, HIDCLASS.ASM y USB_MAIN.ASM. Los tres primeros archivos y parte de la estructura del cuarto han sido desarrollados por Microchip por lo que pueden ser descargados de su página. Sin embargo gran parte del archivo USB_MAIN.ASM fue desarrollado como parte de la actualización de la SRM. Las partes desarrolladas por Microchip se han dejado intactas (comentarios y nombres de variables originales). En este Apéndice solo se listará el código de este archivo.

```

;Autor: Victor Lomas
;Revision: 12.0
;Fecha: Diciembre 2005
;Configuration Bits: H4 Oscillator, WDT Off, Power up timer off
;
;Proyecto de actualización de la SRM DISCA, IIMAS. UNAM
;
movf    STATUS,W
clrf    STATUS           ; force to page 0
movwf   Status_save     ; save STATUS
movf    PCLATH,w
movwf   PCLATH_save     ; save PCLATH
movf    FSR,w
movwf   FSR_save       ; save FSR

ifdef __16C765           ; Trabaja con 16c745 or 16c765
#include <p16c765.inc>
endif
ifdef __16C745
#include <p16c745.inc>
endif

#include "usb_defs.inc"

; comment this line out in all asm files after testing is complete
#define SHOW_ENUM_STATUS ;use portb to display USB
connection status

errorlevel -302 ; supress "register not in bank0, check page bits" message
__CONFIG      _H4_OSC & _WDT_OFF & _PWRTE_OFF &
_CP_OFF

unbanked udata_shr
W_save     res      1

bank0      udata
Status_save res      1      ; registers for saving context
PCLATH_save res      1
FSR_save   res      1
outbuffer  res      2      ; buffer proveniente del anfitrión
temp1      res      1
temp2      res      1
temp3      res      1
sensor     res      2
cont       res      1
pbyte      res      1

extern     InitUSB
extern     PutEP1
extern     GetEP1
extern     ServiceUSBInt
extern     CheckSleep
extern     RemoteWakeup

STARTUPcode
pagesel   main
goto     main
nop

InterruptServiceVector
movwf    W_save           ; save W

;*****
; Interrupt Service Routine
; First we step through several stages, attempting to identify the source
; of the interrupt.
;
;*****
*****

Process_ISR
; Step 1, what triggered the interrupt? only USB and RX interrupts will be
used

btfscl   INTCON,T0IF     ; Timer 0
nop

btfscl   INTCON,RBIF     ; Port B
nop

btfscl   INTCON,INTF     ; External Interrupt
nop

BANKSEL   PIR1

pagesel   ServiceUSBInt ; interrupción USB
btfscl   PIR1,USBIF
call     ServiceUSBInt

btfscl   PIR1,ADIF      ; AD Done?
nop

btfscl   PIR1,RCIF
nop

btfscl   PIR1,TXIF
nop

btfscl   PIR1,CCP1IF
nop

btfscl   PIR1,TMR2IF
nop

btfscl   PIR1,TMR1IF
call     IntTimer1

```

```

        btfscl PIR2,CCP2IF
        nop
;
;*****
;*****
; End ISR, restore context and return to the main program
;
;*****
;*****
EndISR
        clrf STATUS ; select bank 0
        movf FSR_save,w ; restore the FSR
        movwf FSR
        movf PCLATH_save,w ; restore PCLATH
        movwf PCLATH
        movf Status_save,w ; restore Status
        movwf STATUS
        swapf W_save,f ; restore W without corrupting STATUS
        swapf W_save,w
        retfie

        code

        bankisel outbuffer
        movlw outbuffer
        movwf FSR
        movlw 0x2
        call GetEP1

        pagesel ProcessOUTBuffer
        btfs STATUS,C ;if we didn't get any data
        goto LoopForData ;loop back and try again

        call ProcessOUTBuffer ; process the data

        pagesel PutEP1 ;now send results back
        bankisel sensor
        banksel sensor
        movlw sensor
        movwf FSR
        movlw 0x2 ; 2 bytes
        call PutEP1 ; the data can not be read using

;HIDCOMM.readfrom

        EndLoopForData
        pagesel LoopForData
        goto LoopForData ; repeat

;*****
; Main program that sets up the buffers and calls the ISR for processing. ProcessOUTBuffer
;
;
;
;*****
main
        movlw .30 ; delay 16 uS to wait for USB
to reset
        movwf W_save ; SIE before initializing
registers
        decfsz W_save,f ; W_save is merely a convenient register
        goto $-1 ; to use for the delay counter.

        BANKSEL OPTION_REG
        bsf OPTION_REG,NOT_RBPU ; turn off pullups

        BANKSEL TRISB
        clrf TRISB
        movlw 0xF0
        movwf TRISA
        movwf TRISC
        movlw 0x07 ;deshabilitar A/D
        movwf ADCON1 ;todos digitales
        BANKSEL PORTA
        movlw 0x00
        movwf PORTA
        call ConfCompas ;Configura compás

        bankisel outbuffer ;select the bank where outbuffer is stored
        btfscl outbuffer,7 ;¿Lee compás?
        goto Compas
        btfscl outbuffer,6 ;¿Lee analógicos?
        goto Analogicos
        btfscl outbuffer,5 ;¿Lee odómetros?
        goto Odometros
        btfscl outbuffer,4 ;¿Lee Ultrasonido?
        goto Ultrasonido
        movf outbuffer,w ;Envía a PBx
        movwf PORTA ;Dirección en la parte baja de outbuffer
        movf outbuffer+1,w
        movwf PORTB ;Datos a PBx
        call Retardo ;Retardo
        return

;*****
;*****
; Rutinas para interactuar con sensores.
;
;*****
;*****

;*****
; This is the main loop of the program
;*****
LoopForData
        pagesel GetEP1 ;test if data send from PC from Analogicos
        return

;HIDCOMM.writeto
        bankisel outbuffer
        Analogicos
    
```

	return		movlw .8	;inicializa contador con 7
Ultrasonido	movf outbuffer,w		movwf cont	
	movwf PORTA	;Selecciona PB3	otrobit2	call reloj
	movf outbuffer+1,w		movf PORTC,w	
	movwf PORTB	;direcciona y activa transmisor	andlw 0x01	
	call RETARDO		bcf STATUS,C	;limpia carry
	btfs PORTA,4	;Prueba si esta alto o bajo	rlf pbyte,f	;rota a la izquierda
	goto sisen	;salta si hay objeto	iorwf pbyte,f	
	clrf sensor	;no hay objeto	decfsz cont,f	
	return	;avisa con ceros al anfitrión	goto otrobit2	
sisen	movlw 0xFF		bsf PORTA,3	;selecciona latch
	movwf sensor	;pon 1's en sensor avisando	bsf PORTB,5	;SS alto
	return	;que si hay objeto	bcf PORTA,3	;deselecciona latch
			movlw 0x01	;direcciona buffer 1
ConfTimer1			movwf PORTA	
	banksel PIR1	;banco 0	movf pbyte,w	;manda pbyte2 a puerto B
	bcf PIR1,TMR1IF	;limpia int tmr1	movwf sensor+4	
	bsf T1CON,TMR1ON	;prende tmr1	movwf bruja+1	
	bcf T1CON,TMR1CS		movwf PORTB	
	bcf T1CON,T1CKPS1	;Preescaler 8	return	
	bcf T1CON,T1CKPS0		ConfCompas	
	movlw 0x00		bsf STATUS,RP0	;banco 1
	movwf TMR1L		bsf TRISC,1	;habilita RC1 como entrada
	movwf TMR1H		bcf STATUS,RP0	;banco 0
	movwf temp1		bsf PORTA,3	;selecciona latch
	bsf STATUS,RP0	; banco 1	movlw 0xF8	;Enciende
	bsf PIE1,TMR1IE	; activa int por timer	movwf PORTB	
	movlw 0X00		call RETARDO	
	movwf TRISC		bcf PORTB,7	;Reset
	return		call RETARDO	
reloj			bsf PORTB,7	;Fin Reset
	bcf STATUS,RP0	;banco 0	call RETARDO	
	movlw 0x04	;selecciona buffer 4	bcf PORTB,4	;P/C abajo
	movwf PORTA		call RETARDO	;12ms con P/C abajo
repite	nop		call RETARDO	
	bcf PORTB,0	;pone en 0 clk	bcf PORTB,4	;P/C alto
	call RETARDO	;250hz	call RETARDO	;esperando a que EOC sea alto
	bsf PORTB,0	;4ms abajo, 4ms arriba	call RETARDO	
	call RETARDO	;pone en 1 clk	bcf PORTB,4	
	return		NoEOC	
Serie			btfs PORTC,1	
	bsf STATUS,RP0	;banco 1	goto NoEOC	
	bsf TRISC,0	;habilita RC0 como entrada	call RETARDO	;12ms para empezar la
	bcf STATUS,RP0	;banco 0	recepción	
	clrf pbyte		call RETARDO	
	call reloj		call RETARDO	
	movf PORTC,w	;primer bit	bcf PORTB,5	;SS abajo
	andlw 0x01	;solo el bit 0	bcf PORTA,3	;deselecciona latch
	movwf pbyte		call Serie	
	movlw .7	;inicializa contador con 7	return	
	movwf cont		IntTimer1	
otrobit1			banksel PIR1	;banco 0
	call reloj		incfsz temp1,f	
	movf PORTC,w		goto nec	
	andlw 0x01		incf PORTC,f	
	bcf STATUS,C	;limpia carry	movlw 0x00	
	rlf pbyte,f	;rota a la izquierda	movwf TMR1L	
	iorwf pbyte,f		movwf TMR1H	
	decfsz cont,f		bcf PIR1,TMR1IF	
	goto otrobit1		return	
	movf pbyte,w		RETARDO	;4ms
	movwf sensor+3	;restantes 8 bits	movlw 0xFF	
			movwf temp2	
			call RETARDO1	
			decfsz temp2,f	

```

        goto    RET
        return

RETARDO1
        movlw  0x1F
        movwf  temp3
        decfsz temp3,f
        goto  $-1
        return

        latch
        bsf    PORTA,3
        movf  outbuffer+1,w
        movwf PORTB
        bcf    PORTA,3
        return

        end                ;Fin del programa
    
```

➤ Listado del código del software ControlSRM

El código del programa ControlSRM consta de 2 archivos. ControlSRM.frm y Mover.bas

✓ ControlSRM.frm

```

Dim i As Integer                "Presionaste")                255) + 100
Dim muestreo_sensor As Integer End Sub                End Sub
Dim tiempo As Integer

Private Sub Form_Load()                Private Sub OSPC_Click()
'Conecta un dispositivo USB al iniciar el programa                TSENSOR.Enabled = True
                End Sub
                HIDComm1.Connect
                TSENSOR.Enabled = False                Private Sub OSPIC_Click()
                OSPC.Enabled = True                TSENSOR.Enabled = False
                OSPIC.Enabled = False                End Sub
                HSSUV.Enabled = False
                TSU0.Locked = True
                TSU1.Locked = True                Private Sub OSUA_Click()
                TSU2.Locked = True                OSPC.Enabled = False
                TSU3.Locked = True                OSPIC.Enabled = False
                TSU4.Locked = True                TSENSOR.Enabled = False
                TSU5.Locked = True                End Sub
                OSUA.Value = True                Private Sub OSUP_Click()
                tiempo = 0                OSPC.Enabled = True
                End Sub                OSPIC.Enabled = True
                HSSUV.Enabled = True
                Private Sub Form_Terminate()                TSU0.Enabled = True
'Si se desconecta un dispositivo USB, termina el programa                TSU1.Enabled = True
                HIDComm1.Uninit                TSU2.Enabled = True
                End Sub                TSU3.Enabled = True
                TSU4.Enabled = True
                TSU5.Enabled = True
                Private Sub HIDComm1_ConnectSuccess(ByVal Status As Long)                If OSPC.Value = True Then
'Activa las ventanas si un USB está conectado                End If
                CM1HO.Enabled = True                OSPIC.Value = True
                CM1AN.Enabled = True                End Sub
                Caption = "USB PIC - Conectado"                Private Sub Timer1_Timer()
                End Sub                'Intenta reconectar el PIC
                Private Sub HIDComm1_Disconnected(ByVal Status As Long)                If HIDComm1.Connected = False Then
'Desactiva el boton si un USB se ha desconectado                HIDComm1.Connect
                End If                End Sub
                CM1AN.Enabled = False
                Caption = "USB PIC - Desconectado"                Private Sub CM1HO_Click()
                End Sub                CM1HO.Enabled = False
                pasos = HS1P.Value
                mover "h", pasos, 0
                Private Sub Form_KeyPress(KeyAscii As Integer)                Private Sub HSSUV_Change()
                respuesta = MsgBox(KeyAscii, vbCritical, TSENSOR.Interval = HSSUV.Value * (366 / CM1HO.Enabled = True
    
```

```

End Sub
Private Sub CM1AN_Click()
    CM1AN.Enabled = False
    pasos = HS1P.Value
    mover "a", pasos, 0
    CM1AN.Enabled = True
End Sub

Private Sub CM1MO_Click()
    CM1MO.Enabled = False
    pasos = Abs(Int(T1GR / 30))
    For i = 1 To pasos Step 1
        If T1GR > 0 Then
            mover "h", 1, 0
        Else
            mover "a", 1, 0
        End If
    Next i
    CM1MO.Enabled = True
End Sub

Private Sub CM2HO_Click()
    CM2HO.Enabled = False
    pasos = HS2P.Value
    mover "h", pasos, 1
    CM2HO.Enabled = True
End Sub

Private Sub CM2AN_Click()
    CM2AN.Enabled = False
    pasos = HS2P.Value
    mover "a", pasos, 1
    CM2AN.Enabled = True
End Sub

Private Sub CM2MO_Click()
    CM2MO.Enabled = False
    pasos = Abs(Int(T2GR / 30))
    For i = 1 To pasos Step 1
        If T2GR > 0 Then
            mover "h", 1, 1
        Else
            mover "a", 1, 1
        End If
    Next i
    CM2MO.Enabled = True
End Sub

Private Sub CM3HO_Click()
    CM3HO.Enabled = False
    pasos = HS3P.Value
    mover "h", pasos, 2
    CM3HO.Enabled = True
End Sub

Private Sub CM3AN_Click()
    CM3AN.Enabled = False
    pasos = HS3P.Value
    mover "a", pasos, 2
    CM3AN.Enabled = True
End Sub

Private Sub CM3MO_Click()
    CM3MO.Enabled = False
    pasos = Abs(Int(T3GR / 30))
    For i = 1 To pasos Step 1
        If T3GR > 0 Then
            mover "h", 1, 2
        Else
            mover "a", 1, 2
        End If
    Next i
    CM3MO.Enabled = True
End Sub

Private Sub CM4HO_Click()
    CM4HO.Enabled = False
    pasos = HS4P.Value
    mover "h", pasos, 3
    CM4HO.Enabled = True
End Sub

Private Sub CM4AN_Click()
    CM4AN.Enabled = False
    pasos = HS4P.Value
    mover "a", pasos, 3
    CM4AN.Enabled = True
End Sub

Private Sub CM4MO_Click()
    CM4MO.Enabled = False
    pasos = Abs(Int(T4GR / 30))
    For i = 1 To pasos Step 1
        If T4GR > 0 Then
            mover "h", 1, 3
        Else
            mover "a", 1, 3
        End If
    Next i
    CM4MO.Enabled = True
End Sub

Private Sub CM5HO_Click()
    CM5HO.Enabled = False
    pasos = HS5P.Value
    mover "h", pasos, 4
    CM5HO.Enabled = True
End Sub

Private Sub CM5AN_Click()
    CM5AN.Enabled = False
    pasos = HS5P.Value
    mover "a", pasos, 4
    CM5AN.Enabled = True
End Sub

Private Sub CM5MO_Click()
    CM5MO.Enabled = False
    pasos = Abs(Int(T5GR / 30))
    For i = 1 To pasos Step 1
        If T5GR > 0 Then
            mover "h", 1, 4
        Else
            mover "a", 1, 4
        End If
    Next i
    CM5MO.Enabled = True
End Sub

Private Sub CM6HO_Click()
    CM6HO.Enabled = False
    pasos = HS6P.Value
    mover "h", pasos, 5
    CM6HO.Enabled = True
End Sub

Private Sub CM6AN_Click()
    CM6AN.Enabled = False
    pasos = HS6P.Value
    mover "a", pasos, 5
    CM6AN.Enabled = True
End Sub

Private Sub CM6MO_Click()
    CM6MO.Enabled = False
    pasos = Abs(Int(T6GR / 30))
    For i = 1 To pasos Step 1
        If T6GR > 0 Then
            mover "h", 1, 5
        Else
            mover "a", 1, 5
        End If
    Next i
    CM6MO.Enabled = True
End Sub

Private Sub Timer2_Timer()
    tiempo = 1
End Sub

Private Sub TSENSOR_Timer()
    Dim buffer() As Byte
    ReDim buffer(3)
    On Error Resume Next
    buffer(0) = &H82
    i = i + 1
    If i < 6 Then
        buffer(1) = i
    Else
        i = 0
        buffer(1) = i
    End If
    Form1.HIDComm1.WriteTo buffer(), 8
    buffer() = HIDComm1.ReadFrom(8)
    sensor = buffer(0)
    Select Case i
        Case 1
            If sensor = 0 Then
                TSU0.Text = 1
            Else
                TSU0.Text = 0
            End If
        Case 2
            Text3.Text = ""
            If sensor = 0 Then
                TSU1.Text = 1
            Else
                TSU1.Text = 0
            End If
        Case 3
            Text3.Text = ""
            If sensor = 0 Then
                TSU2.Text = 1
            Else
                TSU2.Text = 0
            End If
        Case 4
            Text3.Text = ""
            If sensor = 0 Then
                TSU3.Text = 1
            Else
                TSU3.Text = 0
            End If
        Case 5
    
```



```

Text3.Text = ""
If sensor = 0 Then
    TSU4.Text = 1
Else
    TSU4.Text = 0
End If
Case 0
    Text3.Text = ""
    If sensor = 0 Then
        TSU5.Text = 1
    Else
        TSU5.Text = 0
    End If
Case Else
    Text3.Text = "ND"
End Select
End Sub

Private Sub TSU0_Click()
    Dim buffer() As Byte
    ReDim buffer(2)
    On Error Resume Next
    buffer(0) = &HFF

    buffer(1) = 0
    Form1.HIDComm1.WriteTo buffer(), 8
End Sub
Private Sub TSU1_Click()
    Dim buffer() As Byte
    ReDim buffer(2)
    On Error Resume Next
    buffer(0) = &HFF
    buffer(1) = 1
    Form1.HIDComm1.WriteTo buffer(), 8
End Sub
Private Sub TSU2_Click()
    Dim buffer() As Byte
    ReDim buffer(2)
    On Error Resume Next
    buffer(0) = &HFF
    buffer(1) = 2
    Form1.HIDComm1.WriteTo buffer(), 8
End Sub

Private Sub TSU3_Click()
    Dim buffer() As Byte
    ReDim buffer(2)

    On Error Resume Next
    buffer(0) = &HFF
    buffer(1) = 3
    Form1.HIDComm1.WriteTo buffer(), 8
End Sub
Private Sub TSU4_Click()
    Dim buffer() As Byte
    ReDim buffer(2)
    On Error Resume Next
    buffer(0) = &HFF
    buffer(1) = 4
    Form1.HIDComm1.WriteTo buffer(), 8
End Sub
Private Sub TSU5_Click()
    Dim buffer() As Byte
    ReDim buffer(2)
    On Error Resume Next
    buffer(0) = &HFF
    buffer(1) = 5
    Form1.HIDComm1.WriteTo buffer(), 8
End Sub

```

✓ Mover.bas

```

Attribute VB_Name = "Module2"
Public Function mover(sentido, pasos, puerto)
    Dim buffer() As Byte
    ReDim buffer(2)
    On Error Resume Next
    buffer(0) = puerto
    If sentido = "h" Then
        MsgBox ("horario")
        For i = 1 To pasos Step 1
            buffer(1) = &H99
            Form1.HIDComm1.WriteTo buffer(), 8
            buffer(1) = &H33
            Form1.HIDComm1.WriteTo buffer(), 8
            buffer(1) = &H66
            Form1.HIDComm1.WriteTo buffer(), 8
            buffer(1) = &HCC
            Form1.HIDComm1.WriteTo buffer(), 8
        Next i
    Else
        For i = 1 To pasos Step 1
            buffer(1) = &HCC
            Form1.HIDComm1.WriteTo buffer(), 8
            buffer(1) = &H66
            Form1.HIDComm1.WriteTo buffer(), 8
            buffer(1) = &H33
            Form1.HIDComm1.WriteTo buffer(), 8
            buffer(1) = &H99
            Form1.HIDComm1.WriteTo buffer(), 8
        Next i
    End If
    buffer(1) = &H0
    Form1.HIDComm1.WriteTo buffer(), 8
End Function

```