



UNIVERSIDAD DEL PAPALOAPAN

CAMPUS LOMA BONITA

INGENIERÍA EN MECATRÓNICA

***SISTEMA TELEMÉTRICO PARA LA
PLANEACIÓN DE TRAYECTORIAS DE UN
ROBOT MÓVIL***

Tesis profesional para obtener el grado de

INGENIERO EN MECATRÓNICA

Presenta:

MARISSA YARETH LÓPEZ IVÁN

Asesor de tesis:

M. en C. Rafael Fernando González Zárate

Co-asesor de tesis:

Dr. Hiram Netzahualcoyotl García Lozano

LOMA BONITA, OAX.

AGOSTO 2014

DEDICATORIA

La presente tesis se la dedico especialmente a mi familia porque gracias a su apoyo y consejos he llegado a realizar la más grande de mis metas, la cual constituye la herencia más valiosa que pueda recibir. A mi madre, a quien le debo todo lo que soy, a quien me dio la vida y el amor necesario, por ser la amiga y compañera que me ha ayudado a crecer, brindándome fortaleza, apoyo ilimitado e incondicional para no darme por vencida. A mi padre por ser mi modelo a seguir en la vida, superándose en cada reto que se le presente a pesar de las adversidades y aprendiendo de los errores. A mi hermana por ser el ejemplo de una hermana mayor compartiendo su compañía, consejos y ánimos tanto en los buenos como en los malos momentos.

AGRADECIMIENTOS

Agradezco principalmente a Dios por permitirme llegar hasta este punto esencial de la vida, brindándome salud para lograr mis objetivos y compartirlo con mis seres queridos.

A mis amigos y compañeros por el interés y preocupación de los avances que tenía con respecto a este proyecto y que nunca dudaron de mi capacidad para concluirla. Especialmente a mi novio, amigo y compañero Dagoberto Ascanio Romero por compartir su amistad y tenerme paciencia durante la carrera profesional, sin tener que esperar nada a cambio.

Agradezco a la Universidad del Papaloapan por brindarme la oportunidad de obtener un título profesional, prestándome sus instalaciones. Al igual quiero agradecer a los profesores investigadores del campus Loma Bonita. Principalmente a los que me impartieron clases durante la carrera y también a los profesores que conocí durante la tesis. La relación que llegué a formar con ellos fue muy buena, gracias a sus ánimos, consejos, experiencias, paciencia, hizo que tuviera un ambiente de trabajo agradable.

En especial a mi asesor, co-asesor, sinodal y jefe de carrera, Rafael Fernando González Zárate, Hiram Netzahualcoyotl García Lozano, Eduardo Sánchez Soto y José Luis Najera Sánchez, quienes no solo estuvieron conmigo en esta etapa de la tesis, sino en el transcurso de mi carrera compartiendo su tiempo y conocimiento. Enseñándome que nunca es mal momento para tomarse una taza de café o té.

RESUMEN

La telemetría une sinérgicamente los sistemas de adquisición de datos con los sistemas de comunicación. Estos sistemas se utilizan para la medición de las variables del entorno de manera remota o a distancia. La telemetría es utilizada en este trabajo para realizar el envío de la información obtenida de un robot móvil a una estación de trabajo. En esta estación de trabajo se realizan los cálculos de la trayectoria para la navegación del robot.

El objetivo principal en esta tesis es diseñar y construir el sistema telemétrico y el sistema de generación de trayectorias para un robot móvil.

El sistema telemétrico realiza la adquisición de información por medio de cinco sensores ultrasónicos (para la detección de objetos) y dos encoders de cuadratura (para la posición y desplazamiento del robot). Un microcontrolador, instalado en el sistema telemétrico, es el encargado de leer los pulsos entregados por los encoders, así como de gestionar la lectura y comunicación de la información generada por los sensores ultrasónicos. Este mismo microcontrolador, mediante un protocolo de comunicación serial, realiza la comunicación a distancia a la estación de trabajo por medio de dos módulos de radiofrecuencia (transmisor y receptor).

El sistema de generación de trayectorias construye un camino, para el desplazamiento del robot, a fin de evadir obstáculos que se presenten en su trayectoria, por medio de la especificación de un punto inicial (donde se encuentra el robot), el punto donde se localiza el objeto a evadir y el punto final al cual el robot debe llegar.

El sistema se construyó e implementó en el laboratorio de control de la UNPA, campus Loma Bonita. Para realizar las pruebas de este sistema se construyó un robot tipo diferencial debido a que el robot síncrono para el cual están diseñados no se encuentra disponible por el momento.

Palabras clave: *telemetría, robot móvil, sistema de generación de trayectorias, sistema telemétrico.*

ABSTRACT

The telemetry binds synergistically the data acquisition systems with communication systems. These systems are used for measure environment variables in a remote manner or in distantly form. In this thesis the telemetry are used to send the information obtained by means of data acquisition system of a mobile robot to a work station. In this workstation the trajectory for the mobile robot navigation is computed.

The main purpose of this thesis is to design and build the telemetry system and the trajectories generation system for a mobile robot.

The telemetry system works acquiring information with five ultrasonic sensors (for the object detection) and two quadrature encoders (for the position and the displacement of the robot). A microcontroller, installed on the telemetric system, is in charge of reading the pulses delivered from the encoders, as well as to manage the reading and sending the information obtained from the ultrasonic sensors. This microcontroller, through a serial communication protocol, performs the not wireless communication to the work station with two modules of radiofrequency (transmitter and receptor).

The trajectory generation system builds a path, for the displacement of the robot, in order to avoid the obstacles in its trajectory, given a specified initial point (the place where the robot is positioned), the point where is located the object to be avoided and the endpoint which the robot must reach.

The system was built and implemented at the control laboratory of the UNPA, Loma Bonita campus. The evaluation of this system was carried out in a differential type robot given that the synchronous robot, for which the system was designed, was not available at that moment.

Keywords: *Telemetry, mobile robot, trajectories generation system, telemetry system.*

ÍNDICE

DEDICATORIA	II
AGRADECIMIENTOS	II
RESUMEN	IV
ABSTRACT	V
Capítulo 1 INTRODUCCIÓN	1
Capítulo 2 SENSORES	4
2.1 Desplazamiento lineal y giro.....	5
2.2 Potenciómetros para la medida de desplazamiento.....	5
2.3 Codificadores ópticos.....	6
2.4 Sensores de presencia y proximidad.....	10
2.5 Medidores de distancia.....	11
Capítulo 3 SISTEMA TELEMÉTRICO	18
3.1 Sistemas de telecomunicaciones electrónicas.....	19
3.2 Transmisores y receptores.....	21
3.2.1 Transceptor NRF24L01.....	22
3.2.2 Transmisor HM-T y receptor HM-R.....	23
3.2.3 Transmisor TWS-BS y receptor RWS-374.....	24
Capítulo 4 IMPLEMENTACIÓN DEL SISTEMA TELEMÉTRICO	27
4.1 Adquisición de datos para el robot móvil.....	28
4.1.1 Diseño del sistema de percepción ultrasónica.....	28
4.1.2 Diseño del sistema de medición de desplazamiento.....	34
4.2 Diseño del protocolo telemétrico de comunicación.....	36
4.3 Sistema de generación de trayectorias.....	41
4.3.1 Generación de trayectoria mediante el método de polinomios cúbicos con puntos de paso.....	47
Capítulo 5 CONCLUSIÓN	54

BIBLIOGRAFÍA	55
APENDICE A NORMAS Y PROTOCOLO DE COMUNICACIONES	57
APENDICE B PROGRAMAS PARA LA COMUNICACIÓN	77

ÍNDICE DE FIGURAS

Figura 2.1	<i>Clasificación de los tipos de sensores.....</i>	4
Figura 2.2	<i>Potenciómetro a) lineal y b) giratorio.....</i>	6
Figura 2.3	<i>Principio del codificador óptico.....</i>	7
Figura 2.4	<i>Codificadores ópticos rotacionales y lineales.....</i>	7
Figura 2.5	<i>Principio del codificador óptico.....</i>	8
Figura 2.6	<i>Motorreductor con soporte para montaje y encoder, a) Encoder, b) Encoder colocado en la montura de la rueda, c) Encoder acoplado a la rueda con un motor de CD Pololu.....</i>	9
Figura 2.7	<i>Captura en osciloscopio de la salida del encoder con la llanta.....</i>	9
Figura 2.8	<i>Motor-reductor usado en el robot móvil.....</i>	10
Figura 2.9	<i>Funcionamiento básico de los ultrasonidos.....</i>	12
Figura 2.10	<i>Campo de acción del pulso del sensor de ultrasonido.....</i>	12
Figura 2.11	<i>Sensor SRF10.....</i>	14
Figura 2.12	<i>Área de trabajo del transductor ultrasónico empleado en el SRF10.....</i>	16
Figura 2.13	<i>Soporte para el sensor SFR10.....</i>	17
Figura 3.1	<i>Proceso de la emisión y recepción de variables del robot móvil a la PC.....</i>	18
Figura 3.2	<i>Diagrama a bloques simplificado de un sistema de telecomunicaciones de una sola dirección.....</i>	20
Figura 3.3	<i>Diagrama a bloques simplificado de un sistema de telecomunicaciones en ambas direcciones.....</i>	20
Figura 3.4	<i>Esquema básico del transmisor.....</i>	21
Figura 3.5	<i>Esquema básico del receptor.....</i>	21
Figura 3.6	<i>Imagen del transceptor nRF24L01.....</i>	22
Figura 3.7	<i>Módulos de comunicación a) Transmisor HM-T b) receptor</i>	

	<i>HM-R</i>	23
Figura 3.8	<i>Transmisor TWS-BS (a) y receptor RWS-374 (b)</i>	25
Figura 4.1	<i>Robot móvil</i>	28
Figura 4.2	<i>Distribución de los sensores vista superior del robot móvil</i>	29
Figura 4.3	<i>Robot móvil con la altura de los sensores modificada</i>	32
Figura 4.4	<i>Dibujo de los ángulos de cada sensor</i>	33
Figura 4.5	<i>Ángulos totales de cada sensor</i>	34
Figura 4.6	<i>Diámetro y perímetro de las ruedas</i>	35
Figura 4.7	<i>Forma de comunicación en el envío de la información</i>	37
Figura 4.8	<i>Configuración de los sensores y sus triángulos</i>	43
Figura 4.9	<i>Triangulación entre dos sensores a un mismo objeto</i>	45
Figura 4.10	<i>Desplazamiento del robot móvil</i>	49
Figura 4.11	<i>Gráfica del desplazamiento en cm (r)</i>	51
Figura 4.12	<i>Gráfica de posición angular en grados (θ)</i>	51
Figura 4.13	<i>Gráfica de velocidad lineal</i>	52
Figura 4.14	<i>Gráfica de velocidad angular</i>	52
Figura 4.15	<i>Generación de trayectoria para evitar obstáculos</i>	53
Figura A.1	<i>Conector RS232</i>	58
Figura A.2	<i>Patillaje del conector DB-9</i>	58
Figura A.3	<i>Niveles de tensión lógicos para RS232</i>	60
Figura A.4	<i>Ejemplo de envío de un byte según norma RS232</i>	60
Figura A.5	<i>MAX232</i>	62
Figura A.6	<i>Conexión típica del MAX232</i>	62
Figura A.7	<i>Modo asíncrono de la comunicación USART en los PIC16F877</i>	63
Figura A.8	<i>En el protocolo de comunicación asíncrono, cada palabra de información o dato consta de 8 o 9 bits, estando precedidos por un bit de INICIO y les sigue un bit de PARADA. Los bits se transfieren al ritmo de la frecuencia de reloj establecida</i>	64

Figura A.9	<i>Esquema simplificado del conexionado entre dos módulos USART funcionando en modo asíncrono.....</i>	66
Figura A.10	<i>El maestro se conecta con todos los esclavos en el bus I2C con solo dos líneas. Por SDA circulan los datos en serie y por SCL el maestro envía los impulsos de reloj que genera.....</i>	68
Figura A.11	<i>Conexiones al bus.....</i>	69
Figura A.12	<i>Conexión de las resistencias a las líneas bidireccionales.....</i>	71
Figura A.13	<i>Formato del byte.....</i>	71
Figura A.14	<i>El reconocimiento.....</i>	72
Figura A.15	<i>Formato de los datos transferidos.....</i>	73
Figura A.16	<i>El maestro transmite al esclavo receptor. No cambia el byte de dirección.....</i>	73
Figura A.17	<i>El maestro recibe desde el esclavo después de enviarle a éste la dirección.....</i>	73
Figura A.18	<i>El maestro cambia el byte de dirección y selecciona un esclavo distinto.....</i>	74
Figura A.19	<i>El primer byte.....</i>	75

ÍNDICE DE TABLAS

Tabla 2.1	<i>Banco de registros del SRF10.....</i>	14
Tabla 2.2	<i>Comandos de direcciones del I²C del SRF10.....</i>	15
Tabla 3.1	<i>Características del transmisor HM-T.....</i>	24
Tabla 3.2	<i>Características del transmisor TWS-BS y receptor RWS-374.....</i>	25
Tabla 4.1	<i>Lectura de los cinco sensores con alcance de un metro y ganancia de 500.....</i>	30
Tabla 4.2	<i>Lectura de los cinco sensores con alcance de un metro y ganancia de 700.....</i>	31
Tabla 4.3	<i>Lectura de los cinco sensores con alcance de seis metros y ganancia de 700.....</i>	31
Tabla 4.4	<i>Lectura de los cinco sensores con alcance de un metro, ganancia de 700 y una altura de 25cm.....</i>	32
Tabla 4.5	<i>Selección del rango del divisor de frecuencia.....</i>	36
Tabla A.1	<i>Terminología del bus I2C.....</i>	70

Capítulo 1 INTRODUCCIÓN

Con el transcurso de los años el hombre ha tenido la necesidad de manipular sistemas a distancia, con la finalidad de hacer los trabajos en forma más práctica, reduciendo su tiempo de ejecución y con mayor seguridad. Con este fin surge un área de la tecnología que une sinérgicamente el área de la metrología con los sistemas de comunicación, conocida como telemetría.

La telemetría ayuda a entablar la comunicación alámbrica o inalámbrica, obteniendo mediciones de variables como son: distancia, velocidad, posición, entre otras variables.

Las aplicaciones se presentan en gran variedad de campos, entre los cuales se incluyen trabajos subterráneos (minería, construcción de túneles), misiones espaciales, aplicaciones militares, entre otras. Estas necesitan sensores que permitan recabar la información para poder conocer las condiciones a distancia de algún sistema.

Una de las aplicaciones en las cuales la presencia del sistema telemétrico es indispensable se encuentra en la manipulación y/o control a distancia de robots autónomos.

Considere un robot móvil con un sistema de locomoción simple y eficiente. Se busca que este pueda moverse en un ambiente no estructurado (con objetos móviles y estáticos) de un punto inicial a un punto final conocido. Para tal efecto son necesarios tres sistemas, a saber, un sistema de adquisición (sensores), un sistema de control (la guía del robot móvil) y un sistema de generación de trayectorias. Debido a que el robot debe ser capaz de conocer el ambiente en donde se desenvuelve y de actuar ante las condiciones que se le presenten, los sistemas de adquisición y control deben estar montados dentro del robot. Sin embargo, es deseable que el sistema generador de trayectorias no se encuentre en el robot, para dotarlo de mayor capacidad de procesamiento (de la que podría ser instalada en el robot). Es aquí en donde la telemetría juega un papel importante, dado que permite procesar la información de los

sensores a distancia, generar una trayectoria para evitar los obstáculos y enviar esta información al sistema de control para que lleve al robot a su destino.

El problema que se aborda en este trabajo es, el cómo realizar la comunicación bidireccional entre el sistema de adquisición de datos en el vehículo y la estación de trabajo, así como el proceso mismo de generación de trayectorias.

La comunicación entre diversos dispositivos puede ser realizada por dos medios: cableado e inalámbrico. Un sistema cableado limita el espacio de trabajo y provoca que el desplazamiento de cualquier tipo de robot móvil sea ineficiente. Por otra parte las comunicaciones inalámbricas permiten un desplazamiento más amplio en diversos entornos de trabajo (mayor distancia incluso con obstáculos). En este sentido, algunas de las alternativas son: por medio de transmisión óptica (infrarroja, láser) o por radiofrecuencia (AM, FM, entre otras).

Al existir obstáculos en el espacio de trabajo, los cuales en determinado momento pueden impedir la comunicación con el robot móvil, una comunicación óptica no es viable. Así, la comunicación por medio de radiofrecuencia se estima como la más recomendable para realizar la telemetría entre la estación de trabajo y el robot móvil.

El cálculo de la trayectoria no puede realizarse en el mismo robot móvil (como ya se mencionó anteriormente), debido a que el sistema de procesamiento instalado en el robot se encargará de las funciones necesarias para la detección de los obstáculos (por medio de sensores ultrasónicos), conocimiento de su posición (a través de encoders) y control del sistema de locomoción; es por tal motivo que se propone una estación de trabajo remota, dedicada a la generación de trayectorias.

Los resultados de este trabajo se presentan a continuación en los siguientes capítulos. El capítulo dos se encarga del estudio de los sensores y transductores que pudiesen ser utilizados en un robot móvil. Posteriormente en el capítulo tres se hace una descripción acerca de la forma en que se va a comunicar el sistema de adquisición de datos del robot con el sistema de generación de trayectorias. El siguiente capítulo se menciona la fase experimental del trabajo al igual que el método

utilizado para la generación de trayectorias. Por último se incluyen breves conclusiones del trabajo.

Capítulo 2 SENSORES

Para transportarse de un lugar a otro, un robot móvil requiere obtener información del entorno en donde se traslada. Para esto los robots móviles cuentan con diversos tipos de sensores que les permite conocer su posición y detectar objetos. En este capítulo se presentan diversos tipos de sensores así como su funcionamiento.

Un sensor es un dispositivo que, a partir de la energía del medio donde se mide, da una señal de salida transducible que es función de la variable medida [1].

Los sensores que son utilizados en robótica presentan en su salida una señal eléctrica dependiente del valor de la variable física medida. Esta señal puede ser acondicionada para su utilización como señal de medición o variable de instrumentación.

Debido a la gran cantidad de sensores que existen en el mercado, es necesario clasificarlos. Para ello existen diversos criterios como son el aporte de energía, la señal de salida y el modo de operación, figura 2.1:

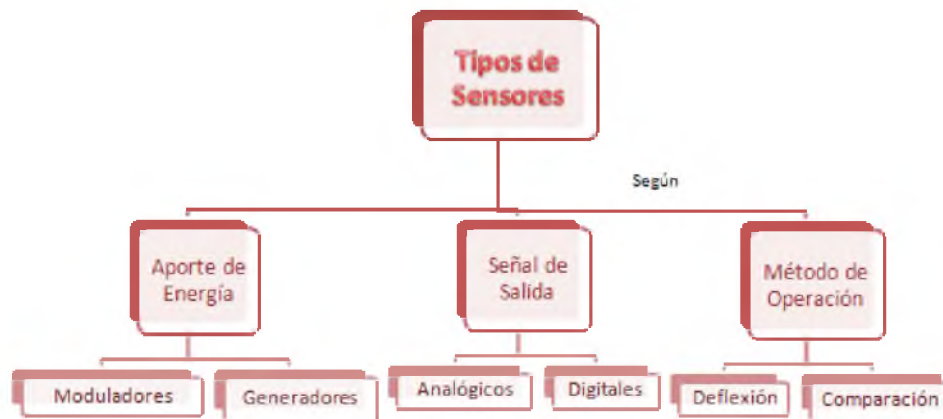


Figura 2.1: Clasificación de los tipos de sensores.

Cuando se habla del aporte de energía se refiere a que los sensores utilizan una fuente auxiliar (moduladores) o generan su propia energía (generadores). En el caso del modo de operación, este puede ser por medio de deflexión o comparación. En el modo de deflexión la magnitud medida produce un efecto físico, similar pero

opuesto, en alguna parte del instrumento. Para los sensores que funcionan por comparación, ellos intentan mantener nula la deflexión mediante la aplicación de un efecto bien conocido y opuesto al generado. Por ejemplo, en un galvanómetro la deflexión de la aguja es directamente proporcional a la corriente que pasa por su bobina. Por otro lado un sistema de comparación es el puente de Wheatstone, cuyo valor de una resistencia desconocida es determinado ajustando el valor de una resistencia conocida para llevar la corriente que pasa por el elemento central a cero.

Dependiendo de la forma en que se encuentra la señal en su salida se pueden clasificarse en analógicos y digitales. La señal de salida analógica es una función continua de la variable medida. La señal de salida digital representa la magnitud de la variable medida en forma de una serie de cantidades discretas (ceros y unos) codificadas en un sistema de notación [2].

Si bien esta clasificación nos permite dar un panorama general de los tipos de sensores, de acuerdo a sus características, en la robótica móvil debemos también conocer el funcionamiento particular de acuerdo a la variable física que es capaz de detectar.

Así se presenta a continuación algunos de los sensores más utilizados en la robótica móvil de acuerdo a la variable que detectan.

2.1 DESPLAZAMIENTO LINEAL Y GIRO

En la robótica móvil la posición y el desplazamiento son variables importantes debido a que sirven para conocer la ubicación y orientación del robot. A continuación se mencionan algunos ejemplos de este tipo de sensores.

2.2 POTENCIÓMETROS PARA LA MEDIDA DE DESPLAZAMIENTO

Los potenciómetros están formados por una pista resistiva con un contacto móvil sobre ella. Por medio de una variación en la posición del punto de contacto se produce un cambio en la resistencia. Este puede ser utilizado para generar una

diferencia de potencial proporcional al desplazamiento. Según se pretenda medir un giro o un desplazamiento, existen los potenciómetros lineales y rotatorios, tal como se muestra en la figura 2.2.

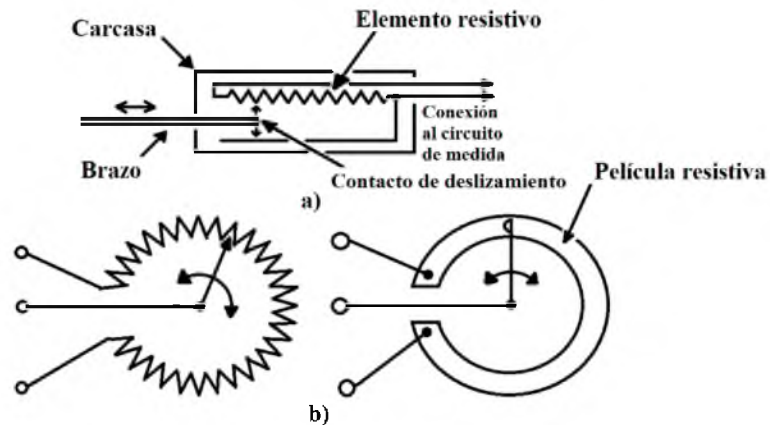


Figura 2.2: Potenciómetro a) lineal y b) giratorio.

Los potenciómetros son sensores relativamente de bajo costo, sin embargo, la precisión es limitada. En general, presentan problemas de fiabilidad debido a desgaste, fricciones, polvo, entre otros. Nótese también que la salida que suministran es analógica, por lo que es necesario un convertidor analógico digital para poder utilizarlo en un control digital.

2.3 CODIFICADORES ÓPTICOS

Los codificadores ópticos, por el contrario, convierten el desplazamiento lineal en una señal digital sin necesidad del convertidor. Este consta de un emisor, un receptor y una rejilla. Para realizar la medición del desplazamiento, una fuente incandescente o un diodo emisor de luz debe ser detectado por un elemento fotosensible (fotodiodo). Transiciones oscuro-luz y luz-oscurο son generadas cuando la luz incide en la rejilla, como se muestra en la figura 2.3.

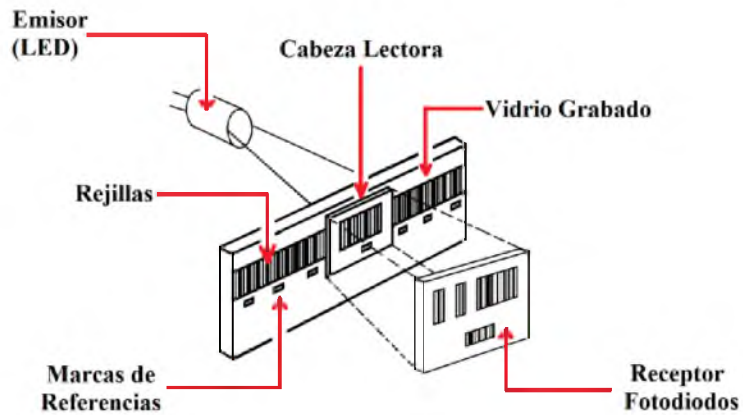


Figura 2.3: Principio del codificador óptico.

Para medir el desplazamiento se cuenta el número de dichas transiciones por medio de un circuito digital. El tipo de movimiento que registran estos sensores puede ser rotacional o lineal como se muestra en la figura 2.4.

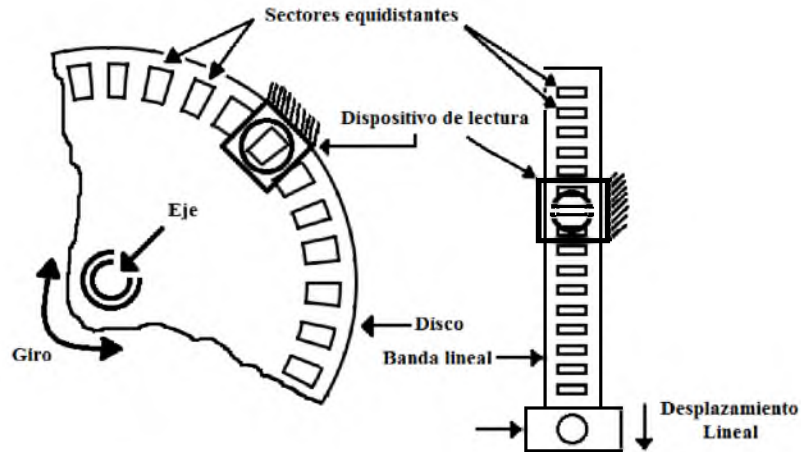


Figura 2.4: Codificadores ópticos rotacionales y lineales.

En los codificadores ópticos lineales no es posible detectar la dirección del desplazamiento por que en su rejilla solamente tiene una hilera de orificios. A este tipo de dispositivo se le conoce como de desplazamiento absoluto. En cambio, en los

rotacionales si es posible detectarlo a través del agregado de una o más rejillas. A este tipo de dispositivo se les conoce como incrementales.

Existen varios tipos de codificadores ópticos incrementales entre los cuales se encuentran los de cuadratura y los que nos permiten generar un código asociado a la posición del eje (codificador de posición digital).

Los codificadores ópticos de cuadratura constan de dos rejillas las cuales generan las respectivas señales digitales desfasadas en 90° . Dependiendo de cual se active primero se determina el sentido del giro.

Los codificadores ópticos de posición digital están formados por varios grupos de rendijas concéntricas, alternadas de forma tal que produzcan un código para cada posición del eje angular, como se puede ver en la figura 2.5. Normalmente este código es Gray (para evitar errores de lecturas). Una ventaja que existe en el uso de estos dispositivos es que tiene una buena resolución, sin embargo puede tener una mala robustez mecánica.

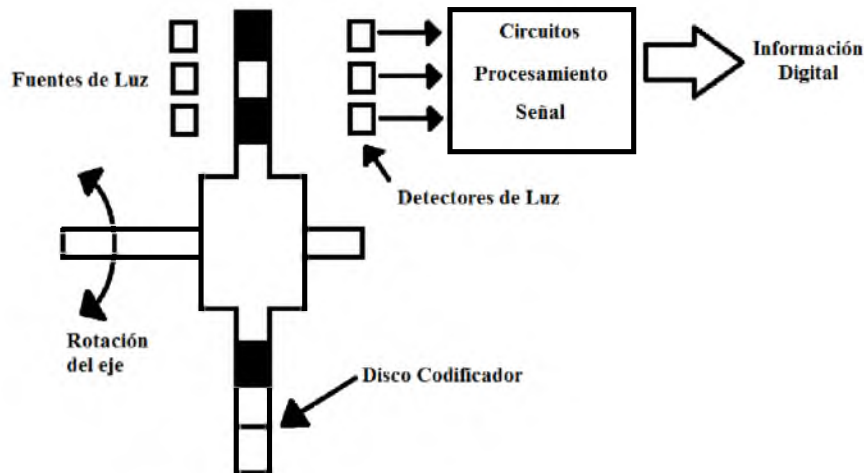


Figura 2.5: Principio del codificador óptico.

En el diseño del robot móvil se utiliza un encoder (codificador óptico) de cuadratura para ruedas Pololu de 42x19 mm. La montura de la rueda cuenta con 12 dientes lo que corresponde a una resolución lineal de 3 mm. Ocupa dos pares de

sensores infrarrojos para tomar medidas de desplazamiento. Estas características se ilustran en la figura 2.6.

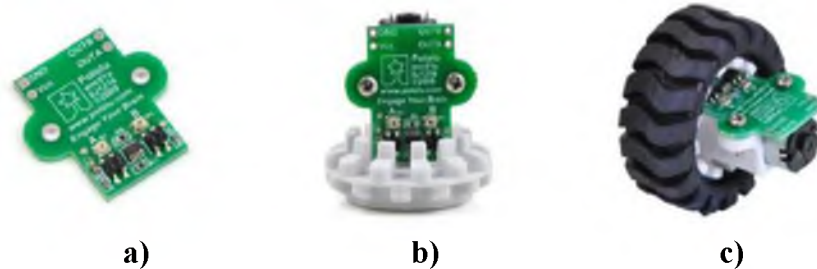


Figura 2.6: Motorreductor con soporte para montaje y encoder, a) Encoder, b) Encoder colocado en la montura de la rueda, c) Encoder acoplado a la rueda con un motor de CD Pololu.

El encoder utiliza una alimentación de 5V. Al conectarlo, conjuntamente con el motor, se activan y desactivan las salidas A y B. La figura 2.7 muestra estas salidas en el osciloscopio. En esta figura la señal superior es la salida del sensor A y la inferior es la del sensor B. Dado que el sensor B se enciende primero, entonces, el sentido de giro es de B hacia A.

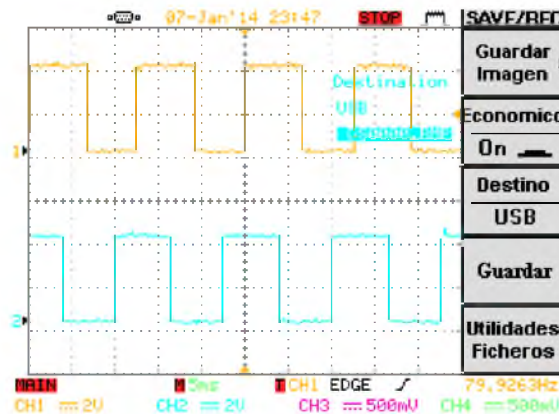


Figura 2.7: Captura en osciloscopio de la salida del encoder con la llanta.

Este dispositivo va montado sobre un motor metálico de CD, el cual sirve para la locomoción del motor, tipo miniatura, con dimensiones de 24 mm x 10 mm x 12

mm, con escobillas y utiliza una caja reductora de velocidad con relación de transmisión 250:1, tal como se muestra en la figura 2.8.



Figura 2.8: *Motor-reductor usado en el robot móvil.*

2.4 SENSORES DE PRESENCIA Y PROXIMIDAD

Los sensores de presencia y proximidad tienen un amplio espectro de aplicaciones. Estos cumplen con la función de detectar objetos o medir la distancia hasta ellos. Los sensores de presencia registran cuando el objeto entra en contacto físicamente con el sensor. Por su parte los sensores de proximidad registran el objeto sin necesidad de que exista un contacto.

Entre los sensores de presencia se encuentran los de fin de carrera. Estos son interruptores, colocados en los extremos de posible contacto, que al momento de ser activados o desactivados se detecta la presencia del objeto.

Los sensores de proximidad conforman un grupo más amplio. Su capacidad de detección es mucho mayor y además, al no tener contacto físico con el objeto, se reducen sustancialmente vibraciones y errores de medición.

No todos los sensores son capaces de detectar cualquier objeto. Dependiendo del material del objeto que va ser detectado, debe ser utilizado un sensor distinto. Ejemplo de ello son los sensores inductivos y de efecto Hall que se utilizan cuando los objetos son ferromagnéticos. Los sensores inductivos trabajan mediante la alteración del campo magnético que ocasiona la presencia del objeto ferromagnético. Los de efecto Hall, se basa en el principio de una fuerza de Lorentz.

Cuando el material no es ferromagnético se ocupan sensores capacitivos, los cuales permiten detectar cambios de capacidad inducidos por superficies cercanas. Los capacitivos a diferencia de los inductivos son menos precisos ya que están basados en un cambio de capacitancia que experimenta un condensador al introducir un objeto en su rango de medida.

Los sensores ópticos son otra forma de detectar proximidad. Para su funcionamiento es necesario que un dispositivo funcione como emisor de un haz de luz y que otro funcione como receptor, al ser interrumpido el haz de luz se genera una medición. Tienen diversas configuraciones entre el emisor y receptor del dispositivo. Al igual que los otros dispositivos, estos no generan ninguna medición de distancia.

Existen muchos otros sensores de proximidad cuyos principios de funcionamiento puede ser muy variado, siendo los mencionados anteriormente los más comunes. Por este motivo no se abordarán más en este trabajo.

2.5 MEDIDORES DE DISTANCIA

Muchas veces se requiere algo más que detectar un objeto, es decir, se necesita conocer la distancia a la que se encuentra el objeto. Para este fin se ocupan medidores de distancias. Existen gran variedad de medidores de distancia como son los ultrasónicos, sensores de láser de tiempo de vuelo y de medida de desfase.

Los sensores láser de medida de desfase miden el desplazamiento por medio del desfasamiento de la onda emitida y recibida reflejada por el objeto. Los sensores de láser de tiempo de vuelo se basan en la determinación del tiempo entre los pulsos emitidos y los que se reciben después de ser reflejados en el objeto. Si bien estos dispositivos son de alta precisión su principal desventaja es el elevado costo de los láseres. A diferencia de los láser, los sensores de ultrasonidos son de costo bajo, y trabajan de forma similar.

En la figura 2.9 se muestra el funcionamiento de los medidores de distancia ultrasónicos. Para iniciar la medición emiten pulsos de ultrasonido (mayor a 20kHz) y se determina el tiempo que tarda en recibir la señal reflejada por el objeto.

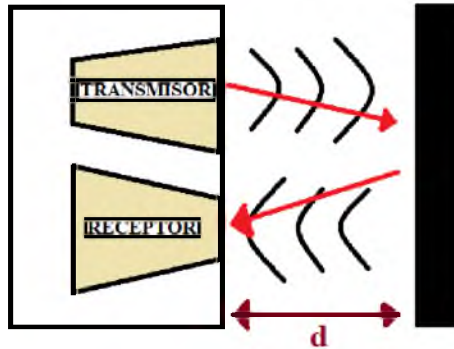


Figura 2.9: *Funcionamiento básico de los ultrasonidos.*

De esta forma teniendo en cuenta la velocidad de propagación del sonido (343 m/s), puede llegar a determinarse una distancia. Todo esto se representa mediante la siguiente ecuación:

$$d = \frac{1}{2} V * t$$

Existen inconvenientes que influyen de una forma determinante en las medidas realizadas. Entre dichos factores cabe destacar:

- a) El campo de acción del pulso transmitido tiene forma cónica. Este indica la presencia del objeto más cercano, por lo tanto, no especifica en ningún momento la localización angular (figura 2.10).

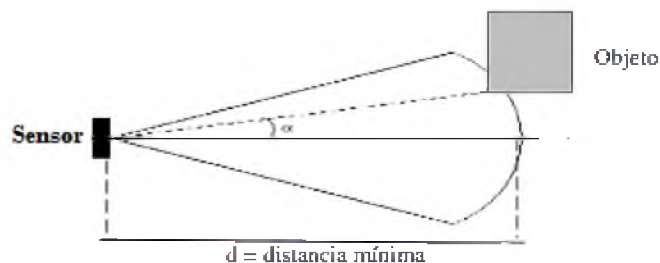


Figura 2.10: *Campo de acción del pulso del sensor de ultrasonido.*

- b) La cantidad de energía acústica reflejada por el obstáculo depende en una gran medida de la estructura de su superficie.
- c) Los factores ambientales tienen una gran repercusión sobre las medidas. Por ejemplo la temperatura influye sobre la velocidad de propagación de la onda de aire.
- d) Falsos ecos producidos por diversas razones. Ejemplo de ello es cuando la onda emitida se refleja varias veces en diversas superficies antes de que vuelva a incidir en el transductor o cuando se emplea una serie de sensores trabajando al mismo tiempo.
- e) Las ondas de ultrasonido obedecen a las leyes de reflexión (ángulo de incidencia igual al ángulo de reflexión respecto a la superficie normal).

Estos factores serán considerados en el presente trabajo, para disminuir los posibles errores en el funcionamiento.

DESCRIPCIÓN Y CONFIGURACIÓN DEL SENSOR ULTRASÓNICO SRF10

En el sistema de navegación el robot utiliza sensores ultrasónicos SRF10. Estos son capaces de detectar objetos a una distancia de seis metros y al comunicarse con un microcontrolador por medio de un protocolo I²C (Inter Integrated Circuit) pueden conectarse hasta 16 sensores en el mismo bus.

Entre las características principales de este sensor que lo hacen atractivo sobre otros modelos es que su tamaño es menor (32x15x10mm), se alimenta con una única fuente de 5V, requiere de 15mA para operar y 3mA mientras se encuentra en reposo (figura 2.11).

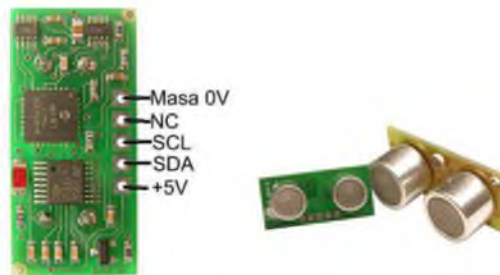


Figura 2.11: *Sensor SRF10.*

Para su conexión el SRF10 cuenta con cinco terminales: la alimentación, tierra y el pin señalado como NC (no conectar). Las otras terminales utilizadas para la comunicación en el bus I²C son el reloj (SCL) y el soporte de datos (SDA), cuyo funcionamiento se detalla en el apéndice A.

El funcionamiento de este sensor se basa en un banco de registros mediante los cuales el microprocesador interno se comunica con el usuario. En la tabla 2.1 se presenta la información de estos registros.

Dirección	Lectura	Escritura
0	Revisión de Software	Registro de comando
1	Sin uso (lee 0X80)	Registro de ganancia máx. (por defecto 16)
2	Byte alto de 1° eco	Registro de alcance de distancia (por defecto 255)
3	Byte alto de 2° eco	No disponible

Tabla 2.1: *Banco de registros del SRF10.*

En estos registros el usuario modifica la información que tiene el sensor ultrasónico (por defecto), es decir, cuando se escribe en el registro uno se puede cambiar la ganancia, o si modifica la dirección dos el alcance en distancia.

La dirección cero se conoce como **registro de comandos** y se utiliza para iniciar la sesión de cálculo de la distancia. Inicialmente se le envía un cero a este

registro para que el sensor espere a continuación un comando. Existen distintos comandos para iniciar una medición (Véase tabla 2.2).

COMANDOS		ACCIÓN
Decimal	Hexadecimal	
80	0X50	Modo cálculo distancia - Resultado en pulgadas
81	0X51	Modo cálculo distancia - Resultado en centímetros
82	0X52	Modo cálculo distancia - Resultado en microsegundos
160	0XA0	1° en la secuencia para cambiar la dirección I2C
165	0XA5	2° en la secuencia para cambiar la dirección I2C
170	0XAA	3° en la secuencia para cambiar la dirección I2C

Tabla 2.2: Comandos de direcciones del I²C del SRF10.

Estos comandos pueden modificar el formato en que va a entregar la medida de distancia del sensor o la dirección de identificación ante el I²C.

Para iniciar la medición se escribe el comando que indique el tipo de unidad métrica deseada y se espera el tiempo necesario para la ejecución de la operación. A continuación, se lee el resultado de la medición. Un buffer es definido al momento de concatenar los registros dos y tres (cada uno de ocho bits), para formar una palabra de dieciséis bits (el tamaño real de la medición).

El buffer se pone a cero al comienzo de cada medición. Transcurrido el tiempo de operación, el buffer se llena con el resultado de la medición. Este tiempo es por defecto de 65ms, sin embargo es posible reducir este periodo escribiendo en el registro de alcance (registro dos) antes de lanzar el comando de medición.

El alcance máximo del sensor SRF10 es de 11 metros, el cual está controlado por el tiempo de operación (65ms), este tiempo es el necesario para que el pulso genere el eco para la medición.

Debido a que los movimientos del robot son relativamente lentos no es necesario modificar este periodo, sin embargo las dos razones principales por las que es positivo reducir el tiempo de medición son las siguientes:

1. Para obtener la información sobre el alcance en menos tiempo.
2. Para poder realizar mediciones con el sensor SRF10 a una tasa más rápida.

Cuando comienza el proceso de medición la ganancia analógica por defecto tiene un valor de 700, este valor es adimensional, y equivale a escribir el valor 0x16 en el registro 01. Este valor puede ser modificado, existiendo para ello una serie de valores que van desde 0x00 (equivalente a una ganancia de 40), 0x01, y así sucesivamente hasta 0x16 (ganancia máxima de 700). La finalidad de limitar la ganancia máxima es reducir la sensibilidad de los módulos al eco distante más débil, mientras que al mismo tiempo sigue siendo capaz de detectar la proximidad de objetos.

El ángulo de detección no se puede cambiar. El foco de trabajo del SRF10 es un cono cuyo ancho depende del propio traductor y es fijo. La forma del área de trabajo del sensor ultrasónico empleado en el SRF10 se encuentra en la figura 2.12, tomada de la hoja de características del fabricante.

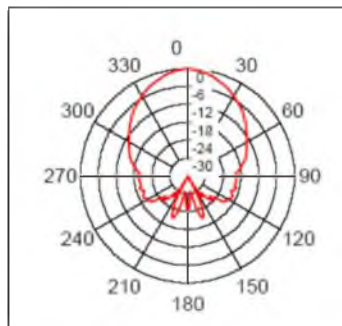


Figura 2.12: Área de trabajo del transductor ultrasónico empleado en el SRF10.

Debe tenerse en cuenta que el módulo SFR10 no presenta orificios de montaje. Esto se ha hecho deliberadamente para que el módulo sea lo más pequeño posible.

Para montar el modulo se utilizó un soporte S360142 que está especialmente diseñado para fijar el modulo sobre un servo motor. Este soporte se muestra en la figura 2.13.



Figura 2.13: *Soporte para el sensor SFR10.*

Capítulo 3 SISTEMA TELEMÉTRICO

El problema de la generación de un camino para un robot móvil ha sido tratado ampliamente. Ejemplos de algunas soluciones son las técnicas conocidas como métodos geométricos, la programación dinámica, los campos de potencial, el enfoque del histograma de campo vectorial, el método de velocidad-curvatura, el método de ventana dinámica, el método de esquemas de motores, entre otras. Estas técnicas varían dependiendo del tipo de navegación del robot. Esta puede ser local (el robot posee la capacidad de reaccionar ante situaciones inesperadas), o global (entornos conocidos, planificando una trayectoria óptima libre de obstáculos) [3].

El proceso de generar una trayectoria requiere de gran capacidad de procesamiento. Cuando este sistema se encuentra en el robot los dispositivos de procesamiento ocupan gran cantidad de espacio, dan mayor peso al robot por lo que requiere mayor energía, lo que no es conveniente para la maniobrabilidad del robot. En muchos casos este proceso se realiza fuera del robot (en una estación de trabajo) para que el robot sea más ligero y pueda tener solo dispositivos indispensables para la recopilación de la información del entorno.

Cuando el sistema de generación de trayectoria esta fuera del robot, se requieren de dos sistemas: uno de comunicación a distancia, para enviar la información de un punto a otro y el segundo que permita enviar esta información a la estación de trabajo. Para el primero, la telemetría nos brinda esta posibilidad, esto es, la medición de magnitudes físicas a distancia y su posterior envío para ser procesadas. El segundo está conformado por un sistema de comunicación serial entre el sistema telemétrico y la estación de trabajo. En la figura 3.1 se ilustra el sistema completo, el cual se implementará en el robot móvil.

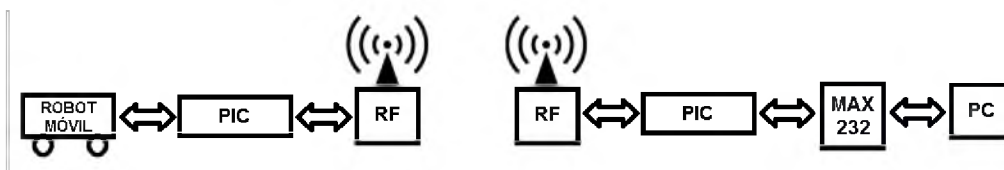


Figura 3.1: Proceso de la emisión y recepción de variables del robot móvil a la PC.

La comunicación se realiza por medio de módulos transmisor-receptor (sistema RF) el cual realiza una comunicación Half-Duplex con el sistema de comunicación serial a fin de enviar la información recabada por los sensores a la estación de trabajo y posteriormente generar la trayectoria del móvil.

Una vez enviada la información de un transmisor a un receptor, esta debe ser procesada para poder ser recibida por la estación de trabajo. La forma más común y sencilla de comunicar cualquier dispositivo con una computadora es a través de su puerto serie, que es compatible con el denominado estándar RS232 (o EIA232 Standard). En una computadora puede haber varios puertos series, normalmente denominados COM1 y COM2.

En las siguientes secciones se presentará la explicación detallada de estos dos sistemas que serán utilizados en este trabajo para la comunicación entre el robot móvil y el sistema de generación de trayectorias. Se hablará de los transmisores y receptores así como de distintos módulos RF disponibles en el mercado, finalizando con el módulo utilizado.

3.1.- SISTEMAS DE TELECOMUNICACIONES ELECTRÓNICAS.

Las comunicaciones electrónicas comienzan su desarrollo en el siglo XVIII. En el año 1900 Guglielmo Marconi, transmitió la primera señal inalámbrica trasatlántica [4]. A partir de esta transmisión los sistemas de comunicación han evolucionado hasta llegar a las comunicaciones satelitales. El primer satélite activo “Telstar 1” en 1962, permitió transmitir señales televisivas entre Estados Unidos y Europa.

Un sistema de comunicación a distancia (telecomunicación) consta de un conjunto de circuitos electrónicos que permiten transmitir, recibir y procesar cualquier tipo de información. En este sistema el transmisor envía la información de la fuente, por un medio de transmisión, a su destino (receptor), como se ilustra en la figura 3.2.

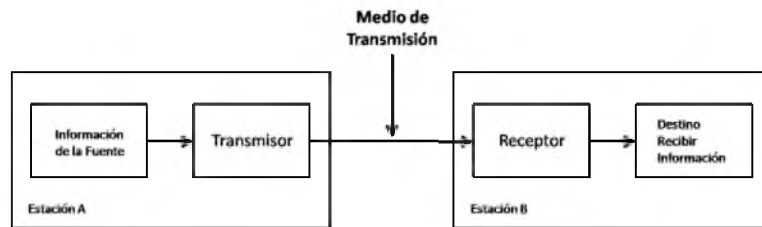


Figura 3.2: Diagrama a bloques simplificado de un sistema de telecomunicaciones de una sola dirección.

La *información de la fuente* se define como el conjunto de datos que se desean enviar durante la comunicación. Puede ser *analógica*, audio y/o video, o *digital*, (información codificada en binario) [5].

Existen aplicaciones donde se requiere la transmisión simultánea entre dos estaciones, esto se puede lograr de dos formas: por medio de dos bloques transmisor-receptor con su respectivo medio de transmisión en cada una de las estaciones de trabajo o transmitiendo toda la información por un medio común, figura 3.3.

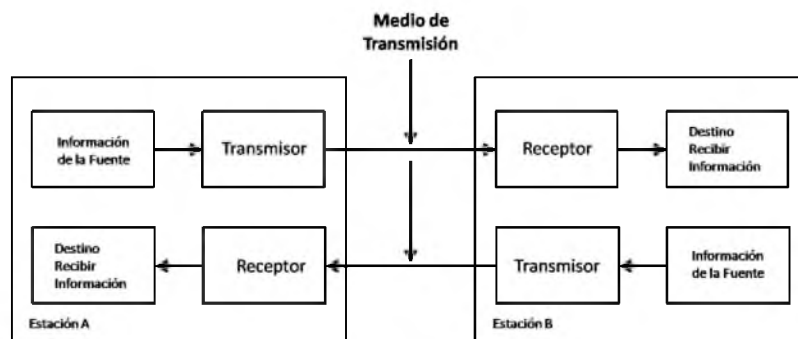


Figura 3.3: Diagrama a bloques simplificado de un sistema de telecomunicaciones en ambas direcciones.

En los siguientes puntos explicaremos más ampliamente cada uno de las partes de la comunicación así como los dispositivos que se utilizan.

3.2.- TRANSMISORES Y RECEPTORES.

Los transmisores son dispositivos que envían la información de la fuente a un receptor, indicador, registrador, controlador o combinación de estos. El esquema básico del transmisor consiste de una fuente de información, un modulador, un sintetizador de frecuencia, un amplificador, un filtro y una antena, figura 3.4.

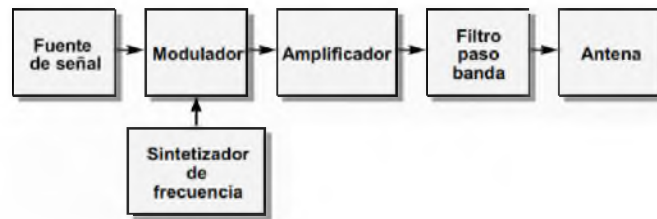


Figura 3.4: Esquema básico del transmisor.

La información que se desea enviar tiene que ser convertida en una señal más apta para ser transmitida. El modulador se encarga de realizar dicha conversión por medio de un sintetizador de frecuencia. El sintetizador de frecuencia cambia la baja frecuencia de la señal de la fuente a una frecuencia relativamente alta. Esta señal modulada es amplificada para poder ser transmitida a menor distancia. Para eliminar el ruido producido por esta modulación se utiliza un filtro pasa banda. Todo esto se enviará posteriormente a la antena.

El receptor, es el encargado de extraer del medio de transmisión la señal deseada y entregarla al transductor de salida; convierte la información recibida a su forma original. Los componentes de un receptor son una antena, un amplificador, un filtro RF, un demodulador y un amplificador banda base, como se muestra en la figura 3.5.



Figura 3.5: Esquema básico del receptor.

Las funciones del receptor son inversas a las que realiza el transmisor. La señal producida por el transmisor se propaga por el medio hasta llegar a la antena. La amplitud con que recibe la antena esta señal es pequeña, por este motivo es necesario amplificarla para poder manipularla. En el proceso de transmisión-recepción se pueden producir ruido e interferencia. El filtro nos ayuda a eliminarlos para que el demodulador regrese a su forma original la información de la fuente.

3.2.1.- TRANSCEPTOR NRF24L01.

Un transceptor es un dispositivo que puede transmitir y recibir al mismo tiempo, sin tener que esperar un tiempo para responder.

El transceptor basado en el integrado nRF24L01 de la empresa Nordic Semiconductor tiene un rango de radiofrecuencia de trabajo de 2.4-2.5GHz, con antena integrada en la placa. Como se puede ver en la figura 3.6, este dispositivo es de tamaño relativamente pequeño, lo que permite ser instalado en sistemas telemétricos móviles gracias a que pueden utilizar una fuente de alimentación de 1.9 a 3.6V (baterías de reloj).



Figura 3.6: *Imagen del transceptor nRF24L01.*

La configuración así como la transmisión y recepción de datos se realiza por medio de un protocolo de comunicación SPI (Interfaz Periférico Serial “Serial Peripheral Interface”). Este permite comunicar de manera síncrona mediante tres líneas SDO (salida de datos en serie), SDI (entrada de datos en serie), SCK (reloj de sincronización). La máxima velocidad de comunicación por este medio es de 8MHz

(máxima velocidad de transferencia a 2Mbps) y cuenta con 128 canales de transmisión seleccionables por este bus.

3.2.2.- TRANSMISOR HM-T Y RECEPTOR HM-R.

Los módulos de transmisión y recepción modelo HM-T y HM-R son fabricados por Co. Microelectronics Hope. Estos utilizan una modulación por corrimiento de frecuencia (FSK), la cual puede reemplazar aplicaciones que ocupen modulación por corrimiento de amplitud (ASK). Los beneficios de usar la tecnología FSK son la robustez a las interferencias y la máxima distancia de trabajo que se puede lograr con estos módulos.

Estos módulos tienen alta sensibilidad, bajo consumo de potencia, detección amplia de trabajo. Los módulos y su diagrama de conexiones se presentan a continuación (Figura 3.7).

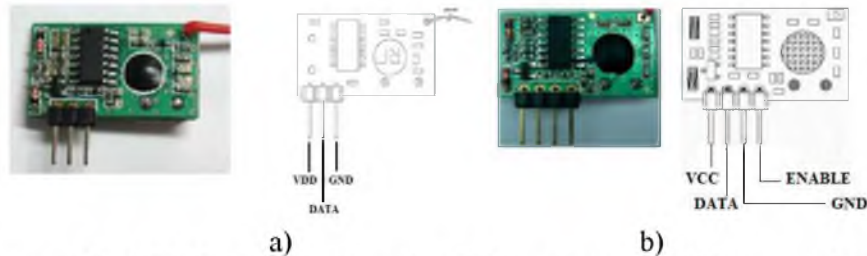


Figura 3.7: Módulos de comunicación a) Transmisor HM-T b) receptor HM-R

Una ventaja de este dispositivo es que al permanecer inactiva la terminal de datos durante más de 70ms el módulo entrará en modo de espera automáticamente. La tabla 3.1 nos presenta características generales de este módulo (disponibles en distintas frecuencias).

	MIN	CARACTERIZADO	MAX	UNIDAD
Voltaje	2.5	3	5	V
Temperatura	-35		80	°C
Rango de Datos	600	4800	9600	Bps

Tabla 3.1: Características del transmisor HM-T.

3.2.3: TRANSMISOR TWS-BS Y RECEPTOR RWS-374.

Los módulos de transmisión y recepción modelo TWS-BS y RWS-374 son fabricados por Co. Electronics Wenshing. Estos utilizan una modulación ASK con un rango de frecuencia de 433 MHz a 315 MHz. Una característica es su facilidad de configuración y su relativo bajo costo.

Las dimensiones del transmisor TWS-BS son 10.45mm x 22.25mm, por lo que se considera que el tamaño es suficientemente pequeño para el montaje en el robot móvil. La potencia de salida es de 14 dBm¹, la cual permite tener un alcance de transmisión de 140 metros al aire libre y hasta 60 metros en lugares cerrados. Para su conexión cuenta con cuatro pines. El pin uno y el pin tres se utilizan para la alimentación, tierra (GND) y alimentación (1.5-12 V) respectivamente, este voltaje de alimentación permite ocupar pilas de tamaño reducido tales como las del tipo 2032 (las usadas en las tarjetas madre en las computadoras). El pin dos es la entrada de datos en la cual se transmite a una razón de 4800 bps. El pin cuatro es la salida de datos del transmisor que va conectada a una antena con una longitud recomendada por el fabricante entre 30 a 35 cm.

El receptor RWS-374 posee la ventaja de que su frecuencia pueda ser ajustable gracias a un capacitor variable que viene incorporado [6]. Este módulo cuenta con ocho pines para su conexión, cinco pines de los cuales se utilizan para la alimentación. Su tamaño reducido (dimensión 10 mm x 43.2 mm) facilita su montaje en los sistemas móviles.

¹dBm es una unidad de medida utilizada en telecomunicaciones para expresar la potencia absoluta mediante una relación logarítmica. El dBm se define como el nivel de potencia en decibelios en relación a un nivel de referencia de 1mW.

Para su conexión el pin uno, seis y siete son tierra (GND) y el pin cuatro y cinco son alimentación (5 V). El pin ocho es la entrada de datos del receptor, la cual se conecta a la antena receptora. Esta debe contar con una longitud recomendada por el fabricante entre 30 a 35 cm.

Después de ser procesada la información recibida se encuentra disponible en el pin dos que es la salida de datos de forma digital.

A continuación se presentan los diagramas al igual que sus pines de conexión del módulo transmisor y receptor. (Figura 3.8)

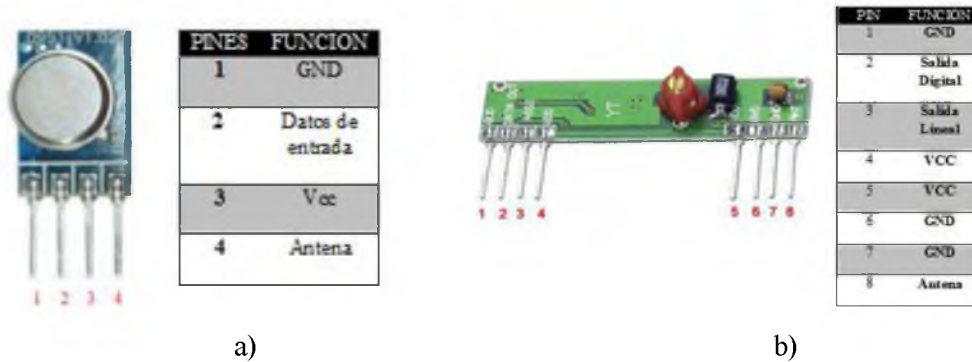


Figura 3.8: Transmisor TWS-BS (a) y receptor RWS-374 (b).

Las aplicaciones de este dispositivo son transmisión inalámbrica de datos, control remoto y domótica, entre otras. Las características eléctricas del transmisor y receptor se encuentran en la tabla 3.2.

CARACTERÍSTICAS ELÉCTRICAS DEL TRANSMISOR TWS-BS Y RECEPTOR RWS-374				
Características	Mínimo	Típico	Máximo	Unidades
Frecuencia de operación (± 250 KHz)	314.500	315.00	315.500	MHz
Rango de datos			8	Kbps
Consumo de corriente			8	mA
Potencia de salida			32	mW
Voltaje de operación	3		12	VDC
Sensibilidad	-106	-108	-110	dBm
Ancho de canal	-500		+500	KHz
Temperatura ambiente de funcionamiento	-20		+85	°C
Ruido equivalente BW		5	4	
Velocidad de datos de la placa base			3	KB/s
Tiempo en turno del receptor			3	Ms

Tabla 3.2: Características del transmisor TWS-BS y receptor RWS-374.

Debido a la configuración sencilla de los dos dispositivos, su fácil programación y bajo costo, se ocuparon los módulos TW-BS y RWS-374 para la implementación de la comunicación del sistema telemétrico entre el robot móvil y la estación de trabajo.

Capítulo 4 IMPLEMENTACIÓN DEL SISTEMA TELEMÉTRICO.

Cuando se habla de robots autónomos se refiere a sistemas que son capaces de realizar tareas sin que intervenga el hombre. Esto significa que el robot debe ser provisto de distintos componentes, tales como sensores, sistemas mecánicos, unidad de procesamiento, de tal forma que este pueda tomar decisiones, a fin de llevar a cabo en forma precisa su tarea asignada.

Para dar mayor capacidad de decisión a los robots se necesitan de procesadores más grandes, lo que requiere de un consumo mayor de energía, provocando que su tamaño y peso sean mayores. Una forma de resolver este problema es con el uso de la teleoperación [7] donde el control o toma de decisión se realiza a distancia y se envía para que el robot pueda actuar. A continuación se explicará de forma detallada el sistema telemétrico que se desarrolló para este trabajo.

La implementación del proyecto consiste en tres etapas: adquisición de datos para el robot móvil, diseño del protocolo de comunicación y el sistema de generación de trayectorias.

El robot móvil con el que se va a trabajar es un robot móvil terrestre a ruedas con configuración síncrona. El concepto de esta configuración es simple, se basa en la capacidad de impulsar y direccionar las tres llantas solo con dos motores. La velocidad lineal y angular de este tipo de robots están completamente desacopladas. Este tipo de diseño tiene la capacidad de girar sobre su propio centro de gravedad.

Debido a que la plataforma robótica no se encuentra aún disponible, se construyó el prototipo que se muestra en la figura 4.1, a fin de poder montar los sensores ultrasónicos y los encoders. Los sensores ultrasónicos se colocaron de la misma forma que estarán en el robot síncrono, un encoder en cada motor, sin embargo la configuración utilizada es de tipo diferencial.



Figura 4.1: *Robot móvil.*

La distribución de los sensores ultrasónicos se realizó con la finalidad de obtener la mayor “visibilidad” para el robot. Debido a que los objetos pueden presentarse en cualquier parte del camino se colocaron tres sensores en la parte frontal, para evitar colisiones con objetos estáticos y móviles, y dos en la parte trasera para evitar colisiones por alcance.

Los encoders se encuentran colocados en las llantas traseras del robot. Cada encoder sensa la posición de cada una de las llantas. Esta información es requerida para realizar el cálculo de la generación de trayectorias.

4.1.- ADQUISICIÓN DE DATOS PARA EL ROBOT MÓVIL

4.1.1.- DISEÑO DEL SISTEMA DE PERCEPCIÓN ULTRASÓNICA

El diseño del sistema de percepción ultrasónica se dividió en tres etapas: la ubicación de los sensores ultrasónicos, las pruebas de medición de distancia y las mediciones de los conos de detección generados por los sensores.

UBICACIÓN DE LOS SENSORES

Como se mencionó en el capítulo dos los sensores ultrasónicos emiten una señal de sonido no audible, esperando el eco y midiendo el tiempo que tarda en retornar. La ubicación alternada de los sensores (figura 4.2) fue seleccionada para minimizar las interferencias provocadas por la emisión de los sensores contiguos.

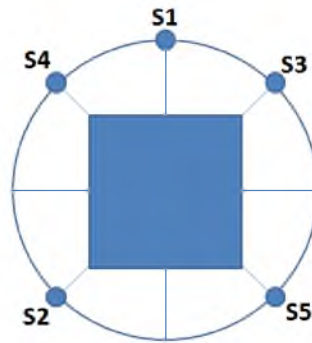


Figura 4.2: *Distribución de los sensores vista superior del robot móvil.*

PRUEBAS DE MEDICIÓN DE DISTANCIA

Se realizaron cinco pruebas de medición de distancia para la caracterización de los sensores ultrasónicos. Estas pruebas se hicieron modificando los parámetros de configuración del sensor (ganancia y alcance máximo de detección). Los objetivos de las pruebas fueron tres; la detección de objetos, conforme se adquirían las distancias en el dispositivo, se obtuvo un margen de error en su medición y la detección del alcance máximo con la finalidad de tomar ese alcance como una distancia suficiente para la evasión del objeto.

PRUEBA 1 (Funcionamiento de los sensores).- Estableciendo un alcance de un metro con una ganancia de 500, se ubicó el robot en una mesa de trabajo en donde se encontraban objetos a diferentes distancia y tamaños. Los resultados registrados son los siguientes (tabla 4.1).

SENSORES	S1	S2	S3	S4	S5
DISTANCIA MEDIDA POR EL SENSOR (CENTÍMETROS)	15	49	39	37	32
DISTANCIA REAL (CENTÍMETROS)	17	50	39	36	35

Tabla 4.1: Lectura de los cinco sensores con alcance de un metro y ganancia de 500.

PRUEBA 2 (Margen de error).- Utilizando un objeto de 12.3x8.2x2cm se realizaron distintas pruebas para detectar la diferencia de la medición del sensor con el valor real a distancia mínima. Se comenzó con el sensor uno colocando un objeto a una distancia de 9cm. El sensor registró 10cm, esto se debe a que en las especificaciones del sensor se precisa que estos pueden tener un margen de error de uno o dos centímetros.

Debido a que la distancia mínima de detección del sensor es de 6cm, para los objetos posicionados en una distancia menor se registran los mismos 6 cm.

PRUEBA 3 (Medidas máximas).- Cuando se utiliza el sensor ultrasónico a nivel de piso las imperfecciones de este provocan un eco antes de detectar un objeto. La mayoría de los pisos no son lisos, por lo cual para realizar las mediciones máximas se propuso levantar los sensores para tener una mejor distancia de detección.

Para realizar estas mediciones el robot móvil se ubicó 67 cm por arriba del nivel del suelo. En esta altura los objetos más cercanos se encontraban a más de un metro de distancia. Se tomaron las distancias máximas con cada uno de los sensores, con una ganancia de 700 y una distancia de un metro de alcance máximo. A continuación se muestra en la tabla 4.2 las nuevas distancias tomada por los sensores.

SENSORES	S1	S2	S3	S4	S5
DISTANCIA MEDIDA POR EL SENSOR (CENTÍMETROS)	15	55	50	39	49
DISTANCIA REAL (CENTÍMETROS)	450	190	290	720	210

Tabla 4.2: Lectura de los cinco sensores con alcance de un metro y ganancia de 700.

Ninguno de los sensores marcaba la distancia establecida. Por lo tanto se modificó el alcance máximo a una distancia de 6 metros, con la finalidad de que al detectar objetos cuente con suficiente distancia para evadirlos.

PRUEBA 4 (Distancia máxima de 6 metros).- Con esta nueva distancia seleccionada, los sensores a 67 cm de altura y una ganancia de 700, comenzaron a realizar medidas aceptables a un metro, sin embargo seguían sin registrar objetos a seis metros, esto se puede ver en la tabla 4.3.

SENSORES	S1	S2	S3	S4	S5
DISTANCIA MEDIDA POR EL SENSOR (CENTÍMETROS)	154	114	142	112	141
DISTANCIA REAL (CENTÍMETROS)	450	190	290	720	210

Tabla 4.3: Lectura de los cinco sensores con alcance de seis metros y ganancia de 700.

Cabe mencionar que a pesar de no registrar objetos a la distancia de 6 metros, es posible usar esas distancias para evadir los objetos.

PRUEBA 5.- El robot móvil debe trasladarse a nivel del piso no importando la superficie. Por lo tanto se regreso a nivel de piso el robot. Los rebotes seguían persistentes, sin importar la ganancia ni la distancia fijada en el sensor. Es por este motivo que se modificó la altura de los sensores, siendo a 25 cm del suelo la altura

en la cual se mantuvo la distancia de un metro sin rebotes (figura 4.3). Estos datos se muestran en la tabla 4.4.



Figura 4.3: Robot móvil con la altura de los sensores modificada.

SENSORES	S1	S2	S3	S4	S5
DISTANCIA MEDIDA POR EL SENSOR (CENTÍMETROS)	94	86	51	79	86
DISTANCIA REAL (CENTÍMETROS)	450	190	290	720	210

Tabla 4.4: Lectura de los cinco sensores con alcance de un metro, ganancia de 700 y una altura de 25cm.

CARACTERIZACIÓN DE LOS CONOS DE DETECCIÓN GENERADOS POR LOS SENSORES

Con las distancias de medición fijas, fue necesario saber los límites de detección angular de cada sensor. Esto es debido a que a pesar de que son fabricados por la misma compañía, suelen haber diferencias en torno al patrón de detección.

Para calcular el ángulo de detección de cono de cada sensor se realizó un

dibujo que contiene un semicírculo con el radio de la distancia máxima de medición (un metro). En la figura 4.4 se ilustran, con diferentes colores para cada sensor, los semicírculos con el radio de la distancia máxima de medición y con otro color diferente el cono de detección. Por ejemplo, para el sensor uno (S1) su semicírculo es de color rosa y su cono de detección es de color verde y su ángulo se llama alpha (α). Los ángulos α_1 y α_2 son diferentes debido a la dirección en la que se encontraba el robot móvil con respecto al dibujo.

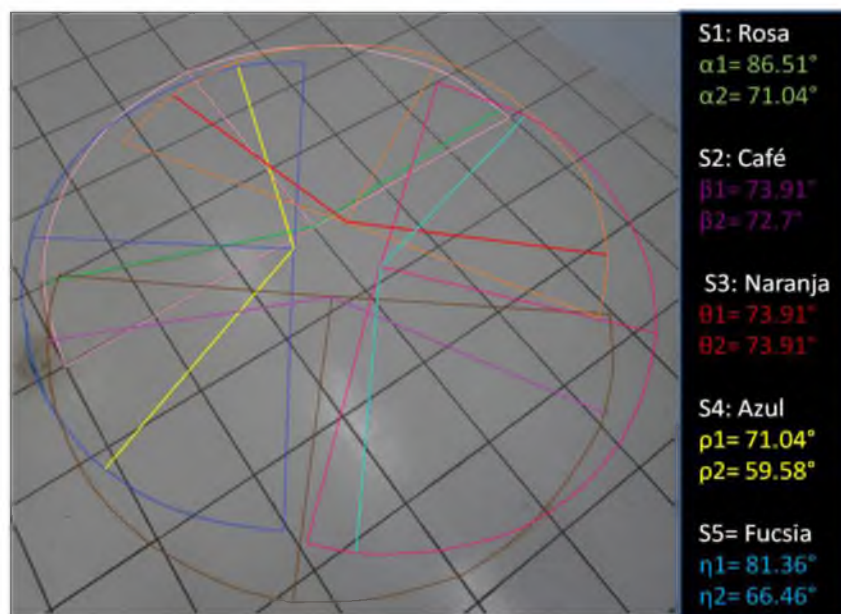


Figura 4.4: Dibujo de los ángulos de cada sensor.

Para saber el límite de ambos lados se utilizó un objeto de 59x76 cm. Con este objeto se realizó un barrido con el cual se midieron los puntos en donde cada sensor empezaba a detectar el objeto, hasta el punto en que el sensor dejara de detectarlo. Para terminar, se realizó el cálculo angular de cada cono por medio de la fórmula de longitud de arco.

$$\sigma = \theta * r$$

Dado que la variable que se desea calcular es el ángulo y se conoce el valor del radio. Entonces se despeja la longitud de arco:

$$\theta = \frac{\sigma}{r}$$

El resultado está en radianes por lo que se necesitará realizar la conversión de radianes a grados.

$$\text{Grados} = \frac{(180 * \text{radianes})}{\pi}$$

A continuación el resultado de los ángulos totales de cada sensor se muestra en la figura 4.5.

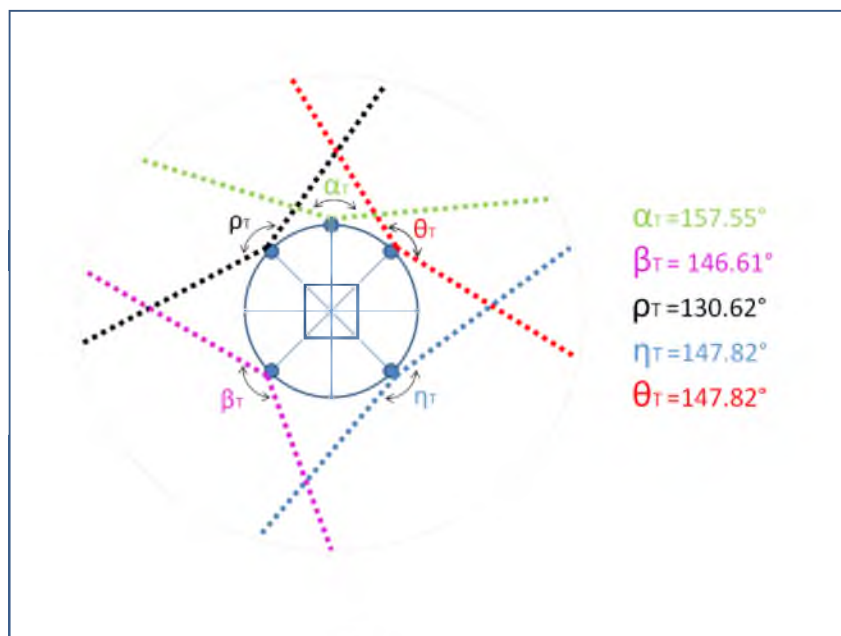


Figura 4.5: *Ángulos totales de cada sensor.*

4.1.2.- DISEÑO DEL SISTEMA DE MEDICIÓN DE DESPLAZAMIENTO.

Para el diseño del sistema de medición de desplazamiento se utilizaron ruedas pololu. Esta rueda, ilustrada en la figura 4.6, consta de una sección dentada cuyo diámetro es de 4cm, recubierta de una rueda de goma con diámetro exterior es de 7cm. La sección dentada consta de 12 dientes con separación de 1cm.

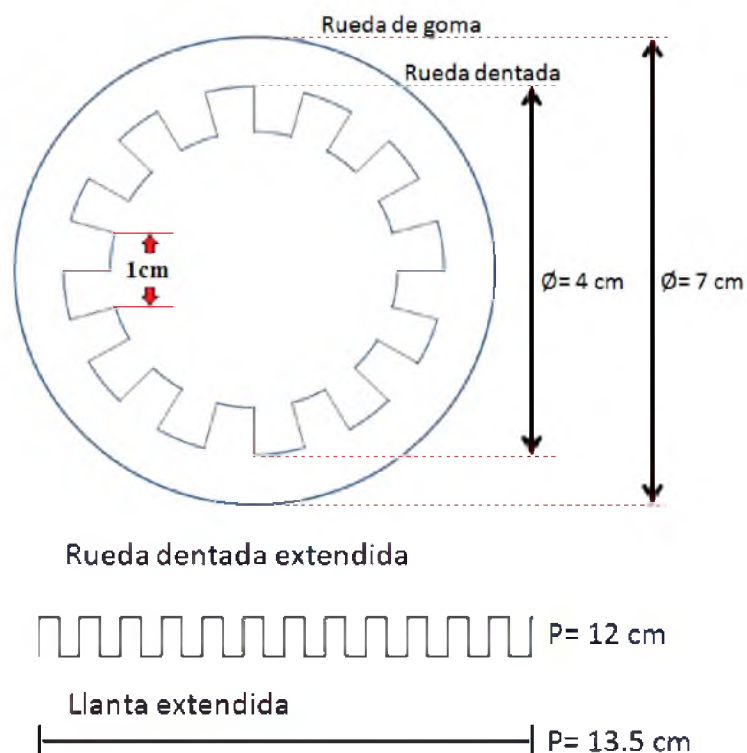


Figura 4.6: *Diámetro y perímetro de las ruedas.*

El encoder óptico, montado en la ruedas, detecta cada ausencia de diente en la sección dentada de la rueda, generando un pulso (figura 4.7). Para determinar el desplazamiento del robot, se debe contar el número de pulsaciones y multiplicarlo por la distancia en el arco exterior generada por este desplazamiento.

El conteo de los pulsos de los encoders se realiza en un sistema periférico especializado incluido en la arquitectura del microcontrolador, llamado timer. El PIC16F877A cuenta con tres timers: TMR0, TMR1 y TMR2, los dos primeros pueden funcionar de dos maneras como temporizador interno o contador externo, cuenta internamente con un divisor de frecuencia (preescaler), el cual se puede configurar mediante la tabla 4.5 y se puede configurar el tipo de flanco activo del conteo.

PS2	PS1	PS0	Divisor del TMR0	Divisor del WDT
0	0	0	1:2	1:1
0	0	1	1:4	1:2
0	1	0	1:8	1:4
0	1	1	1:16	1:8
1	0	0	1:32	1:16
1	0	1	1:64	1:32
1	1	0	1:128	1:64
1	1	1	1:256	1:128

Tabla 4.5: Selección del rango del divisor de frecuencia.

En este caso los timers TMR0 y TMR1 están configurados para trabajar con pulsos externos, así cada pulso detectado incrementa el contador, lo cual corresponde a un avance de 1.125cm.

4.2.- DISEÑO DEL PROTOCOLO TELEMÉTRICO DE COMUNICACIÓN

Existe una gran variedad de protocolos de comunicación, entre ellos están: RS232C, I2C, SPI, Ethernet, solo por mencionar algunos. Estos protocolos se describen de forma detallada en el apéndice A. Todos estos métodos no son más que reglas las cuales se deben cumplir para lograr la comunicación entre dos dispositivos. Cada uno de estos sigue diferentes estándares para establecer su forma de sincronización, codificación, así como la posible detección y corrección de errores en la transmisión del mensaje enviado de un transmisor a un receptor.

En este trabajo se emplean el transmisor TWS-BS y receptor RWS-374, los cuales pueden trabajar por medio de una comunicación RS232C. Este protocolo está diseñado para manejar palabras de 8 o a lo más 9 bits. Sin embargo, para representar la información de la lectura del encoder se utiliza una representación en punto

flotante, la cual es de 32 bits. Es por este motivo que se optó por realizar una comunicación propia la cual se describe a continuación.

La forma de comunicación desarrollada se basa en el envío de información bit a bit, para lo cual se utilizan dos bits de sincronía. Estos dos bits tienen un tiempo de duración de 10ms, uno en alto (1) y el otro en bajo (0). La transmisión de un dato comienza con el envío del bit de sincronía en cero, seguido por el bit de sincronía en uno. Esto se realiza con la finalidad de que el transmisor y el receptor omitan el ruido presente en la transmisión. A continuación se envía la cadena de bits a transmitir, en este caso los 32 bits de la lectura del encoder. Cada bit de información tendrá una duración de 4ms. Para un dato cero se tendrá una duración de 1ms en alto seguido por un nivel bajo con duración de 3ms. En el caso de un dato uno tendrá una duración de 3ms en alto seguido de 1ms en nivel bajo. Todo esto se puede ver en la figura 4.7.

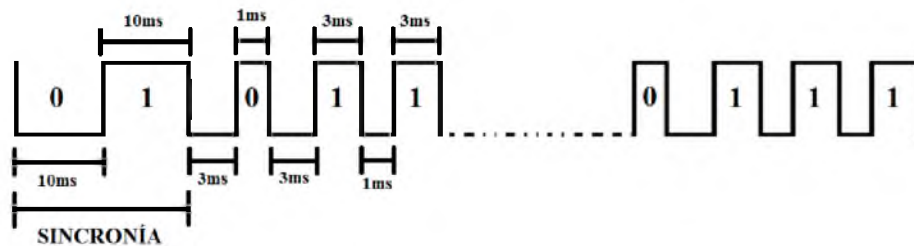


Figura 4.7: Forma de comunicación en el envío de la información.

La etapa de recepción se realiza mediante el proceso descrito anteriormente. El receptor entra en un estado de espera, en el que se verifica el bit de sincronía, y comienza a recibir el dato. Una vez que en esta etapa se recibe el dato, este es separado en paquetes de ocho bits para su posterior envío a través del PIC por medio del RS232 a la estación de trabajo.

A continuación se muestra la función que realiza el protocolo telemétrico de la comunicación para la transmisión y recepción de los datos, tanto de los encoders

como de los sensores ultrasónicos. En el apéndice B se encuentra completo el programa de comunicación telemétrica.

Función tx32 para la transmisión de datos de los encoders de 32 bits con su bit de sincronía.

```
void tx32 (float dato32){ // envío de datos del encoder a la función de transmisión
int i;
    Salida = 0;           // Asegura inicio en cero.
    Salida = 1;  delay_ms(1); // Pulso alto de 10 mseg.
    Salida = 0;  delay_ms(1); // Pulso bajo de 10 mseg.
    for(i=32;i>0;i--){    //leyendo la posición
        if(bit_test(dato32,i-1) == 1 ){
            Salida = 1;   delay_us(300);//delay_ms(3); // Envía un pulso alto de 3 mseg.
            Salida = 0;   delay_us(100);//delay_ms(1); // Envía un pulso bajo de 1mseg.
        }
        Else
        {
            Salida = 1;   delay_us(100);//delay_ms(1); // Envía un pulso alto de 1 mseg.
            Salida = 0;   delay_us(300);//delay_ms(3); // Envía un pulso bajo de 3mseg.
        }
    }
    delay_us(100);       // Pulso bajo de 10ms entre trenes de pulsos
}
```

Función tx16 para la transmisión de datos de los sensores ultrasónicos de 16 bits con su bit de sincronía.

```
void tx16 (int16 dato16){ //envío de datos de los sensores
int i;                    // ultrasónicos a la función de transmisión
```

```

Salida = 0;           // Asegura inicio en cero.
Salida = 1;  delay_ms(1); // Pulso alto de 10 mseg.
Salida = 0;  delay_ms(1); // Pulso bajo de 10 mseg.
for(i=16;i>0;i--){   // leyendo el paquete de la medida
  if(bit_test(dato16,i-1) == 1 ){
    Salida = 1;  delay_us(300);//delay_ms(3); // Envía un pulso alto de 3 mseg.
    Salida = 0;  delay_us(100);//delay_ms(1); // Envía un pulso bajo de 1 mseg.
  }
  Else
  {
    Salida = 1;  delay_us(100);//delay_ms(1); // Envía un pulso alto de 1 mseg.
    Salida = 0;  delay_us(300);//delay_ms(3); // Envía un pulso bajo de 3 mseg.
  }
}
delay_us(100);      // Pulso bajo de 10ms entre trenes de pulsos
}

```

Función rx32 para la recepción de datos de los encoders de 32 bits con su bit de sincronía.

```

float rx32(){
  Tiempo = 0;
  while(Tiempo < 900 ){ // Detectar el primer pulso alto de 10 mseg.
    while(Entrada == 0); // Está bajo y espera que suba
    set_timer1(0);      // Pone a cero el Timer 1.
    while(Entrada == 1); // Se mantiene en el pulso alto.
    Tiempo = get_timer1(); // Toma el tiempo en alto, al salir de alto, en
  }//fin while Tiempo de sincronía // microsegundos.
  set_timer1(0);
  Tiempo=0;
}

```

```

for (i=32;i>0;i--){
    while(Entrada == 0);
    set_timer1(0);
    while(Entrada == 1);
    Tiempo = get_timer1();
    if ((Tiempo) > 280)
        { bit_set(dato32,i-1);}
    if ((Tiempo) < 110)
        { bit_clear(dato32,i-1);}
} // Fin del for
return dato32;
}

```

Función rx16 para la recepción de datos de los sensores ultrasónicos de 16 bits con su bit de sincronía.

```

int rx16(){
    int i=0;
    Tiempo = 0;
    while(Tiempo < 900 ){ // Detectar el primer pulso alto de 10 mseg.
        while(Entrada == 0); // Está bajo y espera que suba
        set_timer1(0); // Pone a cero el Timer 1.
        while(Entrada == 1); // Se mantiene en el pulso alto.
        Tiempo = get_timer1(); // Toma el tiempo en alto, al salir de alto, en
microsegundos.
    } //fin while Tiempo de sincronía
    set_timer1(0);
    Tiempo=0;
    for (i=16;i>0;i--){
        while(Entrada == 0);

```

```

set_timer1(0);
while(Entrada == 1);
Tiempo = get_timer1(); // Num-=1; es Num = Num -1;
if ((Tiempo) > 280)
    { bit_set(dato16,i-1);}
if ((Tiempo) < 110)
    { bit_clear(dato16,i-1);}
} // Fin del for
return dato16;
}

```

4.3.- SISTEMA DE GENERACIÓN DE TRAYECTORIAS

A diferencia de los robots manipuladores, los cuales se mueven dentro de espacios completamente conocidos (estructurados), es común que los robots móviles se desplacen en medios desconocidos o que en el mejor de los casos estén sujetos a cambios (entornos no estructurados).

Para desplazarse de un punto inicial a un punto final, sin colisionar con objetos en su espacio de trabajo, el robot móvil debe generar una trayectoria espacial y temporal desde su origen a su destino. Además, debido a que por lo general el ambiente en el que se desplaza el robot móvil es no estructurado, debe ser capaz de modificar esta trayectoria cuando por medio de sus dispositivos de sensado detecta un objeto en la ruta trazada, esto es planificación de caminos.

Existen gran cantidad de enfoques para realizar esta tarea. Entre ellos destacan los métodos geométricos, polinomiales, la programación dinámica o los campos potenciales, los histogramas de campo vectorial, el método de velocidad-curvatura, el método de ventana dinámica.

En este trabajo, de acuerdo al alcance planteado, la planeación de trayectorias se realizo por medio de una técnica de interpolación, esto debido a que el movimiento del robot será en espacios cerrados (In door), por lo que en este sentido el ambiente es

estructurado. Para la planificación de caminos (esto es detección y evasión de obstáculos) se utilizó un método basado en modelos geométricos. Esto es debido a que es un método relativamente sencillo de utilizar y que el robot móvil satisface las premisas necesarias para desarrollarlo.

A continuación se presentan estas técnicas para la generación de trayectorias para el robot móvil.

Como se explico anteriormente, cuando un sensor detecta un objeto es porque se encuentra dentro de su cono de detección regresando una lectura lineal de la distancia entre el sensor y el objeto.

Esto genera dos problemas importantes:

- No se conoce en realidad la posición del objeto con respecto al Robot (solo la distancia).
- En el caso en que dos sensores adyacentes detectan un objeto, como saber si es el mismo objeto o no.

Para resolver este problema se utilizará un enfoque geométrico considerando la siguiente configuración de los sensores en donde es posible caracterizar cada uno de los triángulos que se generan entre dos sensores adyacentes y el centro del robot como se muestra a continuación en la figura 4.8.

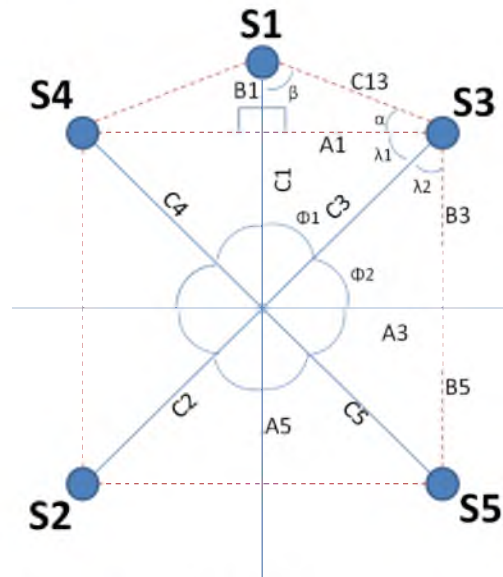


Figura 4.8: Configuración de los sensores y sus triángulos.

Primero se calculan los ángulos internos de los triángulos del S3. Conociendo las variables A3 y C3 se obtiene B3. Esto nos ayuda para calcular los ángulos internos. A continuación se desarrollan los cálculos.

$$B3 = \sqrt{(C3^2 - A3^2)}$$

$$B3 = \sqrt{((15)^2 - (10.6)^2)}$$

$$B3 = 10.61cm$$

$$\phi2 = \sin^{-1} \frac{B3}{C3} = \sin^{-1} \left(\frac{10.61}{15} \right)$$

$$\phi2 = 45.01^\circ$$

$$\lambda2 = \sin^{-1} \frac{A3}{C3} = \sin^{-1} \left(\frac{10.6}{15} \right)$$

$$\lambda 2 = 44.96^\circ$$

$$\phi 1 = 90 - \phi 2 = 44.99^\circ$$

$$\lambda 1 = 90 - \lambda 2 = 45.04^\circ$$

A continuación se realizan los cálculos del triángulo que se encuentran entre los sensores S1 y S3.

$$B1 = C1 - B3 = 15 - 10.61 = 4.39\text{cm}$$

$$\text{Como } A3 = A1 \therefore A1 = 10.6\text{ cm}$$

$$C13 = \sqrt{(A1^2 + B1^2)}$$

$$C13 = \sqrt{((10.6)^2 + (4.39)^2)}$$

$$C13 = 11.47\text{cm}$$

$$\alpha = \sin^{-1} \frac{B1}{C13} = \sin^{-1} \left(\frac{4.39}{11.47} \right)$$

$$\alpha = 22.50^\circ$$

$$\beta = \sin^{-1} \frac{A1}{C13} = \sin^{-1} \left(\frac{10.6}{11.47} \right)$$

$$\beta = 67.54^\circ$$

Cuando dos sensores en posiciones adyacentes detectan un mismo objeto, se puede utilizar una triangulación para determinar la posición xy de este, debido a que se conocen los tres lados del triángulo como se muestra en la figura 4.9.

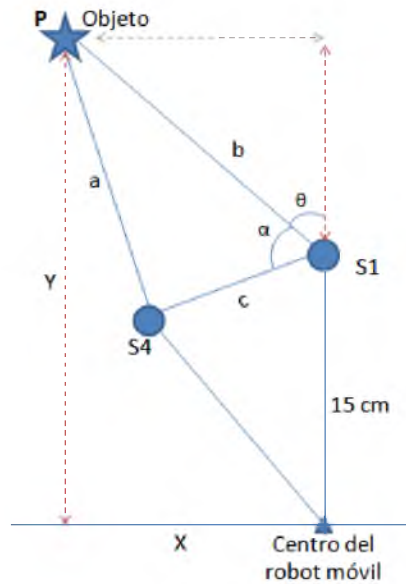


Figura 4.9: Triangulación entre dos sensores a un mismo objeto.

Se aplica la ley de los cosenos para el triángulo con los puntos entre los sensores y el objeto, S4, S1 y P respectivamente, para calcular α y así obtener θ .

$$\alpha = \sin^{-1} \left[\frac{a^2 - b^2 - c^2}{2bc} \right]$$

$$\theta = 90 - \alpha$$

Dado que ya se conoce el ángulo θ se pueden conocer las componentes (x,y) del punto P, las cuales son:

$$P = (x, y) = (a \sin \theta, (b \cos \theta + 15))$$

La posición (x,y) del objeto con respecto al centro del robot es el resultado de estas operaciones.

Esta misma configuración se sigue para cada uno de los lados del pentágono irregular que se genera al tomar cada sensor como una arista.

A fin de conocer si el objeto detectado por los dos sensores es el mismo se evaluarán las coordenadas (x, y) obtenidas en la ecuación del círculo de radio r para cada sensor, estas son:

$$15 \cos 45 = 10.6$$

$$15 \sin 45 = 10.6$$

S1:

$$x^2 + (y - 15)^2 = r^2$$

$$x^2 + y^2 - 30y + 225 = r^2$$

S2:

$$(x + 10.6)^2 + (y + 10.6)^2 = r^2$$

$$x^2 + 21.12x + 112.36 + y^2 + 21.12y + 112.36 = r^2$$

S3:

$$(x - 10.6)^2 + (y - 10.6)^2 = r^2$$

$$x^2 - 21.12x + 112.36 + y^2 - 21.12y + 112.36 = r^2$$

S4:

$$(x + 10.6)^2 + (y - 10.6)^2 = r^2$$

$$x^2 + 21.12x + 112.36 + y^2 - 21.12y + 112.36 = r^2$$

S5:

$$(x - 10.6)^2 + (y + 10.6)^2 = r^2$$

$$x^2 + 21.12x + 112.36 + y^2 - 21.12y + 112.36 = r^2$$

Se estima que si el resultado de esta evaluación es el mismo en la ecuación de dos sensores adyacentes, el objeto es el mismo, en caso contrario se estarían tomando dos objetos diferentes.

4.3.1.- GENERACIÓN DE TRAYECTORIA MEDIANTE EL MÉTODO DE POLINOMIOS CÚBICOS CON PUNTOS DE PASO.

Existen distintas formas de representar una curva en el plano. A fin de generar la trayectoria deseada para el robot, la representación de la curva en el plano (x, y) se realizara por medio de sus ecuaciones paramétricas.

$$p(\lambda) = [x(\lambda), y(\lambda)] = [r(\lambda), \theta(\lambda)] \quad (1)$$

Donde λ es el parámetro, $x(\lambda)$ e $y(\lambda)$ son funciones asociadas en coordenadas rectangulares y $r(\lambda)$, $\theta(\lambda)$ son funciones asociadas en coordenadas polares.

Para la generación de las trayectorias es necesario considerar que la curva generada tiene primera derivada, es decir la velocidad de la curva es al menos continua a pedazos. Por consiguiente, las restricciones son:

$$p(\lambda_0) = p_0 \quad (2)$$

$$p(\lambda_f) = p_f \quad (3)$$

$$p'(\lambda_0) = p'_0 \quad (4)$$

$$p'(\lambda_f) = p'_f \quad (5)$$

Para que se cumplan estas restricciones se requiere un polinomio de grado al menos tres (cúbico):

$$p(\lambda) = a_0 + a_1\lambda + a_2\lambda^2 + a_3\lambda^3 \quad (6)$$

La primera derivada de este polinomio es

$$p'(\lambda) = a_1 + 2a_2\lambda + 3a_3\lambda^2 \quad (7)$$

Y la segunda derivada es

$$p''(\lambda) = 2a_2 + 6a_3\lambda \quad (8)$$

Considerando $\lambda_0=0$ tenemos

$$p(\lambda_0) = a_0 + a_1(0) + a_2(0) + a_3(0) = p_0 \quad (9)$$

Por lo tanto

$$a_0 = p_0 \quad (10)$$

Ahora para λ_f

$$p_f = a_0 + a_1\lambda_f + a_2\lambda_f^2 + a_3\lambda_f^3 \quad (11)$$

De la misma forma para las derivadas

$$p'(\lambda_0) = a_1 + a_2(0) + a_3(0) \quad (12)$$

$$a_1 = p'_0 \quad (13)$$

y

$$p_f = a_1 + 2a_2\lambda_f + 3a_3\lambda_f^2 \quad (14)$$

Resolviendo se obtiene

$$a_0 = p_0 \quad (15)$$

$$a_1 = p'_0 \quad (16)$$

$$a_2 = \left(\frac{3}{\lambda_f^2}\right)(p_f - p_0) - \left(\frac{2}{\lambda_f}\right)p'_0 - \left(\frac{1}{\lambda_f}\right)p'_f \quad (17)$$

$$a_3 = \left(\frac{-2}{\lambda_f^3}\right)(p_f - p_0) + \left(\frac{1}{\lambda_f^2}\right)(p'_0 + p'_f) \quad (18)$$

A continuación se realizará un ejemplo con el método de polinomios cúbicos para entender el comportamiento lineal y angular en un robot móvil.

-Ejemplo

Se tiene el robot móvil síncrono el cual se desea pasar de un ángulo de 0° y una posición inicial P (que sin pérdida de generalidad puede ser cero), a un desplazamiento $P=1,1$ con un ángulo de $\theta=45^\circ$ en 2 segundos, tal como se muestra en la figura 4.10. Por medio del método de polinomios cúbicos se obtienen dos polinomios uno para el desplazamiento lineal y otro para el desplazamiento angular.

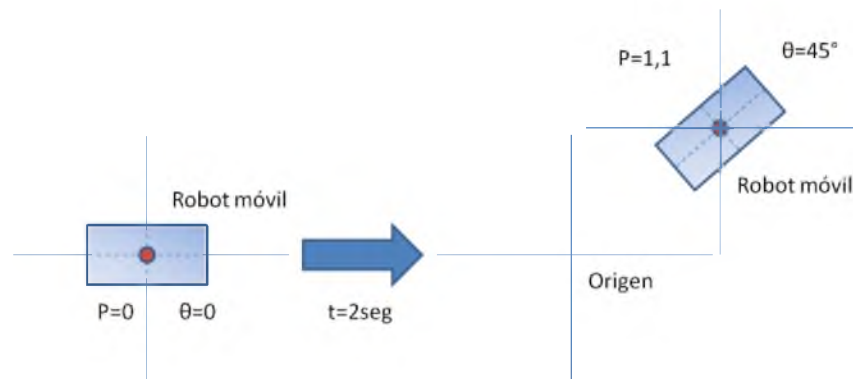


Figura 4.10: Desplazamiento del robot móvil.

Las ecuaciones paramétricas de la ecuación (1) son los valores a calcular. Teniendo un polinomio de 3^{er} orden para el desplazamiento angular:

$$\theta(t) = a_0 + a_1t + a_2t^2 + a_3t^3 \quad (19)$$

Con condiciones iniciales y finales provenientes de las ecuaciones (2), (3), (4) y (5):

$$\theta(t_0) = 0 \quad \text{Pos. Inicial}$$

$$\theta(t_f) = 45 \quad \text{Pos. Final}$$

$$\theta'(t_0) = 0 \quad \text{Vel. Inicial}$$

$$\theta'(t_f) = 0 \quad \text{Vel. Final}$$

Estas condiciones se aplican a las ecuaciones (15), (16), (17) y (18). Por lo tanto los parámetros son los siguientes:

$$\begin{aligned}
 a_0 &= 0 \\
 a_1 &= 0 \\
 a_2 &= \frac{3}{t_f^2}(P_f) = \frac{3}{2^2}(45) = 33.75 \\
 a_3 &= \frac{-2}{t_f^3}(P_f) = \frac{-2}{2^3}(45) = -11.25
 \end{aligned}$$

Estos parámetros se sustituyen en la ecuación (19)

$$\therefore \theta(t) = 33.75t^2 - 11.25t^3$$

El cálculo del polinomio de desplazamiento lineal se utiliza las mismas ecuaciones que ayudaron a obtener el desplazamiento angular. Por lo tanto el polinomio quedaría de la siguiente forma:

$$r(t) = \bar{a}_0 + \bar{a}_1 t + \bar{a}_2 t^2 + \bar{a}_3 t^3$$

Con los parámetros:

$$\begin{aligned}
 \bar{a}_0 &= 0 \\
 \bar{a}_1 &= 0 \\
 \bar{a}_2 &= \frac{3}{2^2}(\sqrt{2}) \\
 \bar{a}_3 &= \frac{-2}{2^3}(\sqrt{2})
 \end{aligned}$$

En grados sería:

$$\bar{a}_2 = 1.06066017 \quad \bar{a}_3 = -0.3535534$$

$$\therefore r(t) = \frac{3}{2^2}(\sqrt{2})t^2 - \frac{2}{2^3}(\sqrt{2})t^3$$

Las gráficas del comportamiento del desplazamiento lineal y angular son las siguientes.

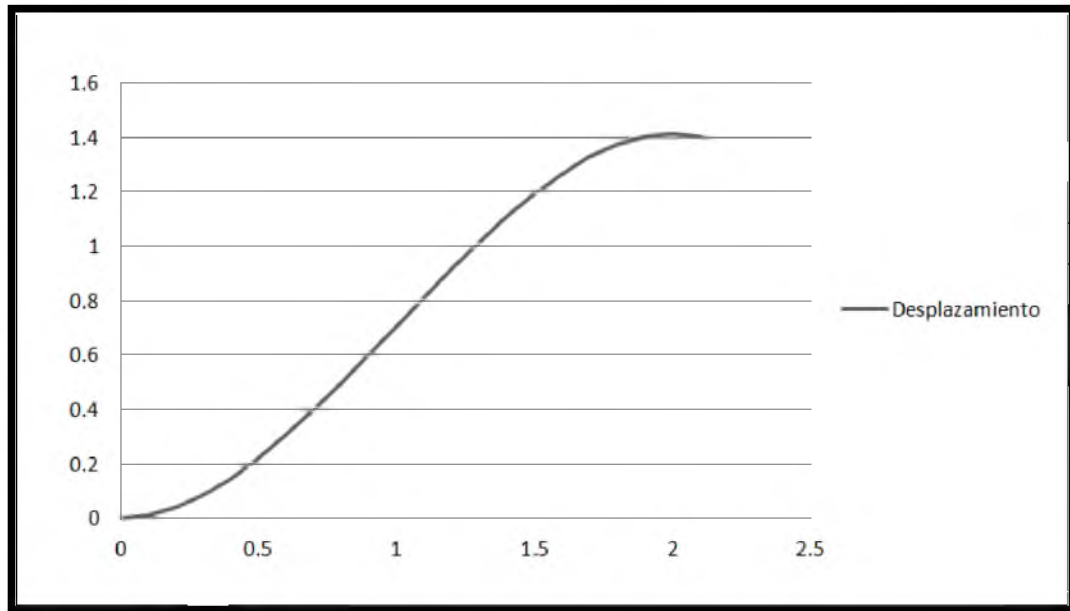


Figura 4.11: Gráfica del desplazamiento en cm (r).

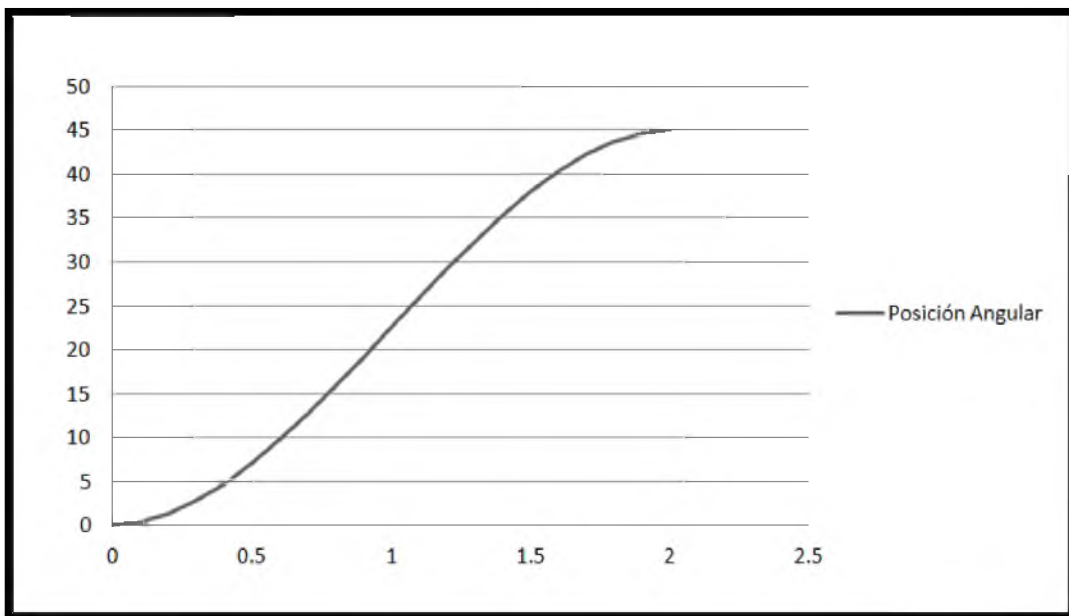


Figura 4.12: Gráfica de posición angular en grados (θ).

Para la etapa de control se requiere utilizar como entradas de control las velocidades tanto lineal como angular. Estas gráficas se presentan a continuación.

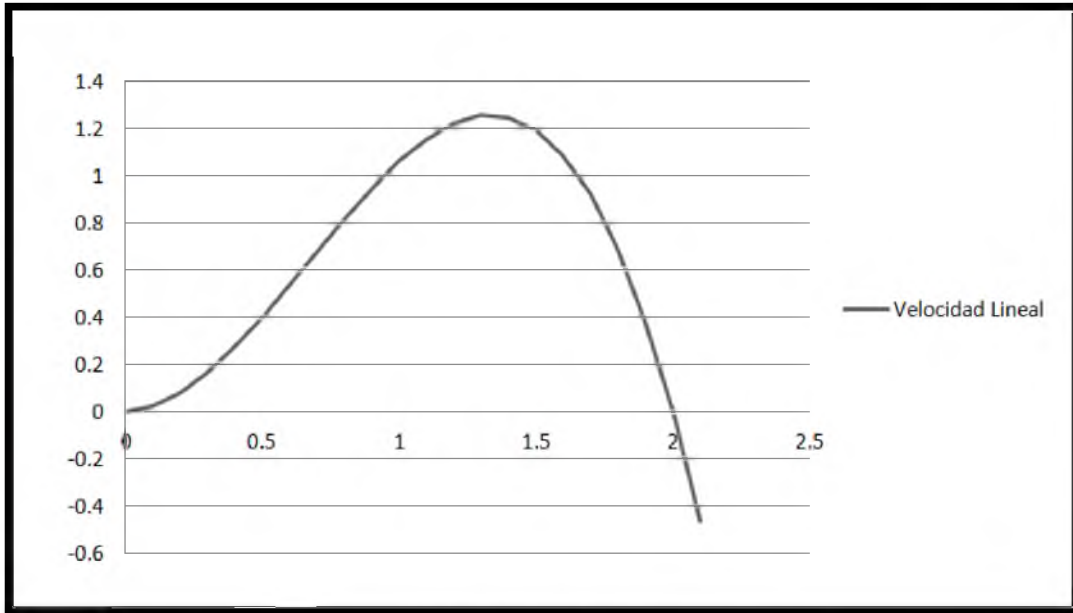


Figura 4.13: *Gráfica de velocidad lineal.*

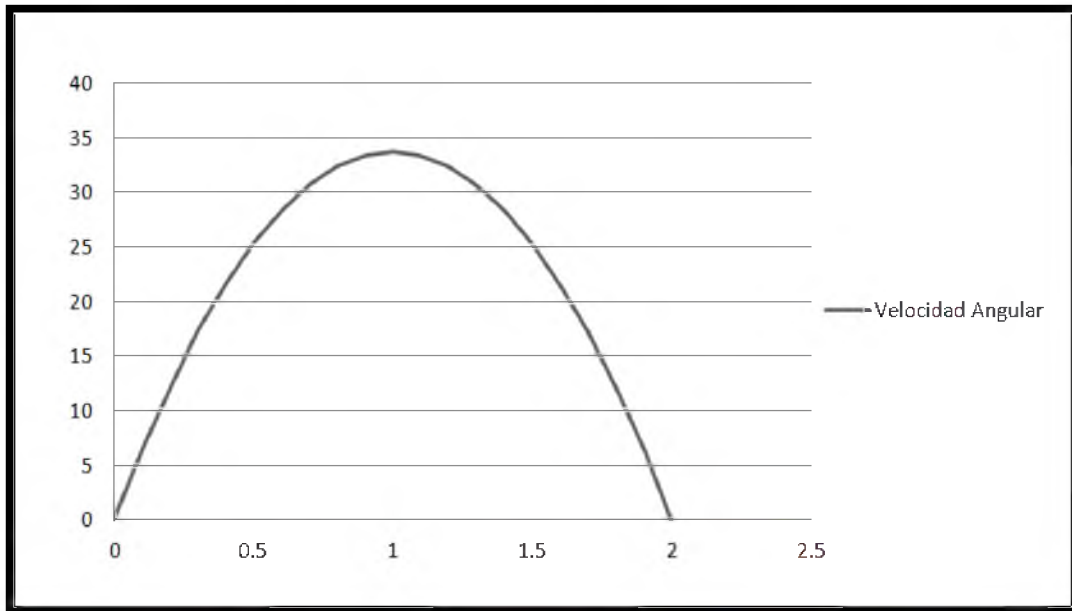


Figura 4.14: *Gráfica de velocidad angular.*

Estas graficas representan la trayectoria que debe seguir el robot entre dos puntos con velocidades iniciales y finales cero.

Ahora consideremos que se tiene un robot móvil, situado en el punto inicial y con destino a un punto final que debe evitar un obstáculo detectado mediante el sistema sensorial. La generación de la trayectoria puede ser planteada como en el caso anterior dado un punto inicial Q_{in} y un punto final Q_{fin} , pasando por uno o más puntos intermedios Q_{pas-i} , como se ve en la figura 4.15.

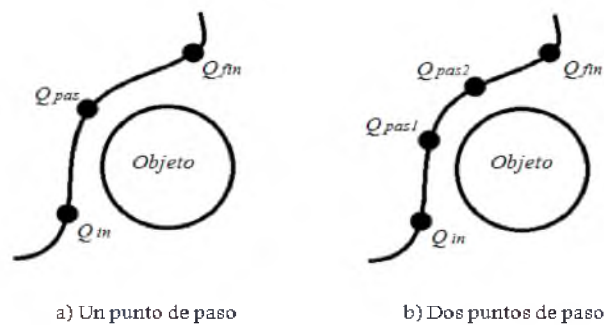


Figura 4.15: *Generación de trayectoria para evitar obstáculos.*

Estos puntos se les llaman puntos de paso. Los puntos de paso normalmente se definen con una cierta tolerancia con respecto a los obstáculos, con lo cual es frecuente que no se exija que el robot pase exactamente por el punto de paso, sino suficientemente próximo.

Para ocupar el algoritmo anterior se calculan más de una trayectoria. Por ejemplo si solo se tiene un punto de paso, el robot móvil calcula dos trayectorias. El primer punto se toma como punto inicial y la primera trayectoria se dirige al punto de paso. En la siguiente trayectoria el punto de paso se toma como un nuevo punto inicial y la trayectoria se dirige al punto final, como se puede ver en la figura 4.15a. Cuando se tiene más de un punto de paso, como se ve en la figura 4.15b, se incrementa el número de trayectorias de manera secuencial siguiendo este mismo procedimiento.

Capítulo 5 CONCLUSIONES

En este trabajo se construyeron tres sistemas independientes con el propósito de realizar el sistema de telemétrico para la generación de trayectorias en un robot móvil. El primero un sistemas de adquisición sensorial capaz de percibir tanto el desplazamiento del robot como objetos que se encuentren en su entorno, cuantificando estos parámetros. El segundo es un sistema de comunicación full duplex, inalámbrico vía radio frecuencia. Para este sistema se implemento un protocolo de comunicación para satisfacer las necesidades de transmisión de los datos de acuerdo a las características particulares de cada uno de ellos. El tercer sistema es una comunicación diseñada sobre un protocolo rs232 que permite la comunicación del sistema telemétrico con una estación de trabajo dedicada al procesamiento de estos datos.

Como parte del sistema de generación de trayectorias, utilizando los datos recibidos, se calculó una nueva trayectoria del robot móvil a fin de que este pueda evadir un obstáculo detectado por el sistema de adquisición de información. Este camino fue obtenido por medio del método de polinomios cúbicos.

Cada uno de estos sistemas fue diseñado, simulado e implementado físicamente para lo cual fue necesaria la construcción de un prototipo robótico, en el cual fueran montados estos sistemas. Una vez montados se realizaron pruebas experimentales de los sistemas en conjunto.

Este trabajo es parte de un proyecto el cual consiste en el diseño y construcción de un robot móvil autónomo. Esta tesis parte medular sobre la cual está basado este sistema. Por lo tanto, este trabajo junto con trabajos futuros integrarán por completo el robot móvil.

BIBLIOGRAFÍA

Capítulo 2

[1] Ramón Pallás Areny. 2001, Sensores y acondicionadores de señal, 3º Edición. Alfaomega, marcombo.

[2] Antonio Creus. 1997, Instrumentación industrial, 6 Edición. Alfaomega, marcombo.

Capítulo 3

[3] Aracely Y., Nelson S. Artículo. “Planeación y seguimiento de trayectorias para un robot móvil”. Escuela politécnica nacional, Quito-Ecuador. 10p.

[4] Leon W. Couch, 2008, Sistema de comunicación digitales y análogos. 7ma Edición. Pearson Educación. 784p.

[5] Wayne Tomasi. 2003, Sistemas de Telecomunicaciones Electrónicas, 4ta Edición. Prentice Hall.

[6] Jaime A. R. Izquierdo, 2011. Diseño e implementación de un sistema electrónico de rehabilitación para niños especiales del instituto carlos garbay. Tesis de ingeniero en electrónica y computación. Escuela superior politécnica de Chimborazo. Riobamba Ecuador. 130p.

Capítulo 4

[7] Jesús M. G. de G., Aníbal O. B. y Alfonso J. G. C. 2006. Teleoperación y Telerrobótica. Pearson Prentice Hall. Madrid. 236p.

APENDICES

- Enrique P., Fernando R. y Lucas J. L. 2006. Microcontrolador PIC16F84, Desarrollo de proyectos. 2^{da} Edición. Alfaomega. 623p.
- José M. A. U., Susana R. Y. e Ignacio A. M. Micronroladores PIC Diseño práctico de aplicaciones. Mc Graw Hill. 231p.
- María Monserrat Godínez García. 2003. Desarrollo de un instrumento virtual para el control de un equipo de potenciales evocados. Tesis de licenciatura en ingeniería biomédica. Universidad autónoma metropolitana. Iztapalapa.

- Eduardo G. B. 2008. Compilador C CCS y simulador PROTEUS para microcontroladores PIC. Primera Edición. Alfaomega. 276p.

APÉNDICE A

NORMAS Y PROTOCOLO DE COMUNICACIONES

A.1 NORMA RS232

En los años 60, cada fabricante usaba una interfaz diferente para comunicar diferentes dispositivos por su incompatibilidad, por lo tanto, la interconexión entre equipos de diferentes fabricantes requerían el uso de convertidores de los niveles de voltajes y fabricación de cables y conectores especiales.

En 1969, el EIA junto con el Bell Laboratories y otros fabricantes establecieron un estándar para la interfaz entre equipos de terminal de datos (DTE por sus siglas en inglés) y equipo de terminal de circuito de datos (DCE por sus siglas en inglés). El objetivo de este estándar era simplificar la interconexión de equipos fabricados por diferentes firmas.

Este equipo llegó a ser el RS-232-C (Recommended Standard number 232, revisión C from the Electronic Industry Association). Un estándar similar fue desarrollado en Europa por el CCITT (Comité Consultatif Internatinal de Telegraphie et Telephonie) conocido como V.24 (descripción funcional) y V.28 (especificaciones eléctricas). El RS-232-C fue adoptado por la mayor parte de fabricantes de terminales y equipamiento.

El 1980 la creciente de la industria de las PC encontró el estándar RS-232-C barato y apropiado para conectar periféricos a la PC. El RS-232-C llegó a ser rápidamente un estándar para conectar a la PC; impresoras, cintas magnéticas, terminales y otras PC's.

Como el estándar solamente soporta velocidades de transmisión hasta 20 kbps y distancias hasta 16 metros, se adoptaron nuevos estándares por la EIA. El RS449 (descripción mecánica) y RS423 (descripción eléctrica) son compatibles con el RS-232-C y se puede operar a velocidades de hasta 10 Mbps y alcanzar distancias de hasta 1200 metros. Sin embargo la adopción de un nuevo estándar es un proceso largo y costoso.

La forma más común y sencilla de comunicar cualquier dispositivo con un ordenador es a través de su puerto serie, que es compatible con el denominado estándar RS232 (o EIA232 Standard). En un ordenador puede haber varios puertos series, normalmente nominados COM1, COM2.

Los puertos series son accesibles mediante conectores. La norma RS232 establece dos tipos de conectores llamados DB-25 (de 25 pines) y DB-9 (de 9 pines), machos y hembras, como se muestra en la figura A.1.



Figura A.1: Conectores R232.

La norma RS232 se estableció para conectar un ordenador con un modem, por lo que aparecen muchas patillas en los conectores DB-25 que en otro de aplicaciones no se utilizan y en las que es más común utilizar el DB-9.

Cada una de las patillas del conector RS232 tiene una función específica por la norma. Hay uno de los terminales por los que se transmiten y reciben datos y otros que controlan el establecimiento, flujo y cierre de la comunicación, que se podrán mostrar en la figura A.2.

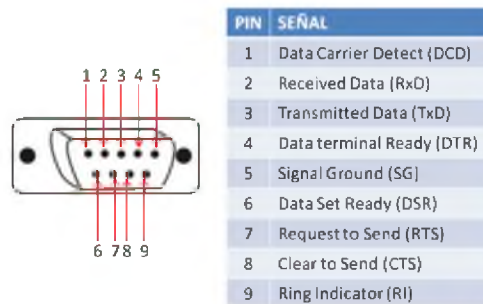


Figura A.2: Patillaje del conector DB-9.

Para llevar a cabo una comunicación con el microcontrolador se debe tener en cuenta la línea de transmisión (TxD, pin 3 Transmitted Data), la línea de recepción (RxD, pin 2 Received Data) y el pin de masa (SG, pin 5 Signal Ground).

Las especificaciones del puerto serie están contenidas en la norma RS232 (o EIA232 Standard). Las que son necesarias para comunicarse con un microcontrolador son la velocidad de transmisión, que es la cantidad de información enviada por la línea de transmisión en la unidad de tiempo. Hay distintas unidades para expresar esta medida, la más utilizada es el Baudio, que es proporcional a los Bits/segundo (bps), definidos como el número de bits de información enviados por segundo.

La velocidad a la que pueden trabajar los puertos COM de un ordenador está normalizada a 75, 150, 300, 600, 1200, 2400, 9600 Baudios, etc. Estos valores son demasiados pequeños para los estándares de hoy en día, pero suficientemente rápidos para multitud de aplicaciones.

Los niveles lógicos también son necesarios para la comunicación y precisamente requieren de ciertos requisitos para cumplir la transmisión serie según la norma RS232.

De acuerdo con la figura A.3, los datos se transmiten con lógica negativa, es decir, un voltaje positivo en la conexión representa un "0", mientras que un voltaje negativo representa un "1". Para garantizar un "0" lógico una línea debe mantener un voltaje entre +3 y +15V. Del mismo modo para "1" lógico garantizado debe estar entre -3 y -15V. Los voltajes más usados son +12V para el "0" y -12V para el "1". Es importante resaltar que cuando un puerto serie no está transmitiendo mantiene el terminal de transmisión a "1" lógico a -12V, normalmente. La banda muerta entre +3V y -3V se conoce como la región de transición donde los niveles lógicos no están definidos. Esto significa que cualquier valor entre +3 y -3 voltios puede interpretarse ambiguamente como "0" ó "1".

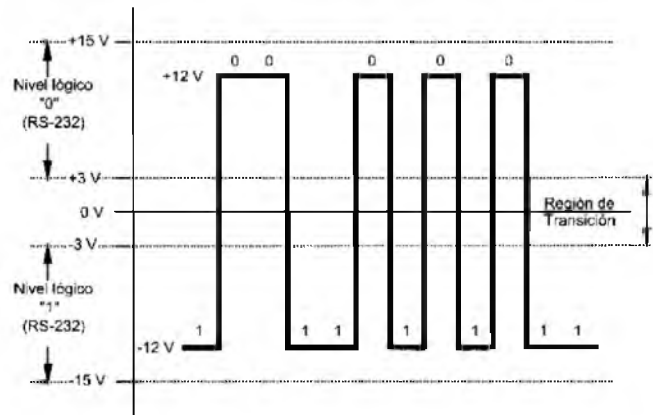


Figura A.3: Niveles de tensión lógicos para RS232.

La comunicación de datos en un puerto serie RS232 se usa normalmente para efectuar comunicaciones asíncronas sin tiempo preestablecido para iniciarse. Los datos llegan en paquetes de información normalmente de 8 bits. Algunos equipos envían por carácter, otros guardan muchos caracteres en la memoria y cuando les toca enviarlos los envían uno tras otro.

El protocolo establecido por la norma RS232 envía la información estructurada en cuatro partes, ilustradas en la figura A.4:

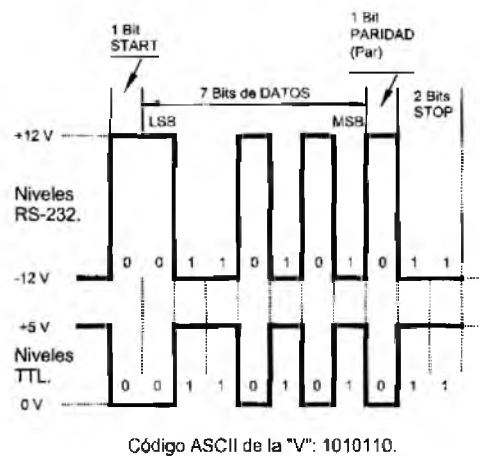


Figura A.4: Ejemplo de envío de un byte según norma RS232.

La figura A.4 consiste de el bit de inicio o arranque (Start) consiste en un paso de -12V a +12V, es decir, de un "1" a un "0" lógico en lógica negativa de la norma

RS232, seguido del bit de datos (Datos) estos son enviados al receptor después del bit de Start. Dentro del bit de datos se encuentran los bits de menos pesos LBS (Least Significant Bit) es transmitido primero y el de mayor peso MSB (Most Significant Bit) el último. También se encuentra el bit de paridad (Parity) dependiendo de la configuración de la transmisión un bit de paridad puede ser enviado después de los bits de datos. Con este bit puede descubrir errores en la transmisión. Y por último, el bit de parada (STOP) la línea queda a -12V después del último bit enviado, es decir, queda a "1" en la lógica negativa de la norma RS232. Indica la finalización de la transmisión de una palabra de datos.

La función que tiene la figura anterior es enviar un bit de inicio, una palabra de 7 bits (1010110), que corresponde a la letra "V" en código ASCII, un bit de paridad par y luego dos bits de paro.

Es importante tomar en cuenta, que el microcontrolador trabaja con lógica positiva a diferencia del puerto RS232 del ordenador que trabaja con lógica negativa. Por lo tanto la forma de la señal en el microcontrolador (niveles TTL) será inversa a los niveles RS232.

En el mercado hay muchos circuitos integrados que permiten la conversión entre niveles TTL y niveles RS232. Entre ellos destaca el transceptor MAX232 fabricado por Dallas Semiconductor-MAXIM.

El MAX232 convierte los niveles RS232 (cerca de +12V y -12V) a voltajes TTL (0 a +5V) y viceversa sin requerir nada más que una fuente de +5V. El chip contiene dos drivers TTL→RS232 y dos RS232→TTL. Necesita cuatro condensadores externos de unos pocos microfaradios para generar el voltaje RS232 internamente tal y como se muestra en la figura A.5 y figura A.6.

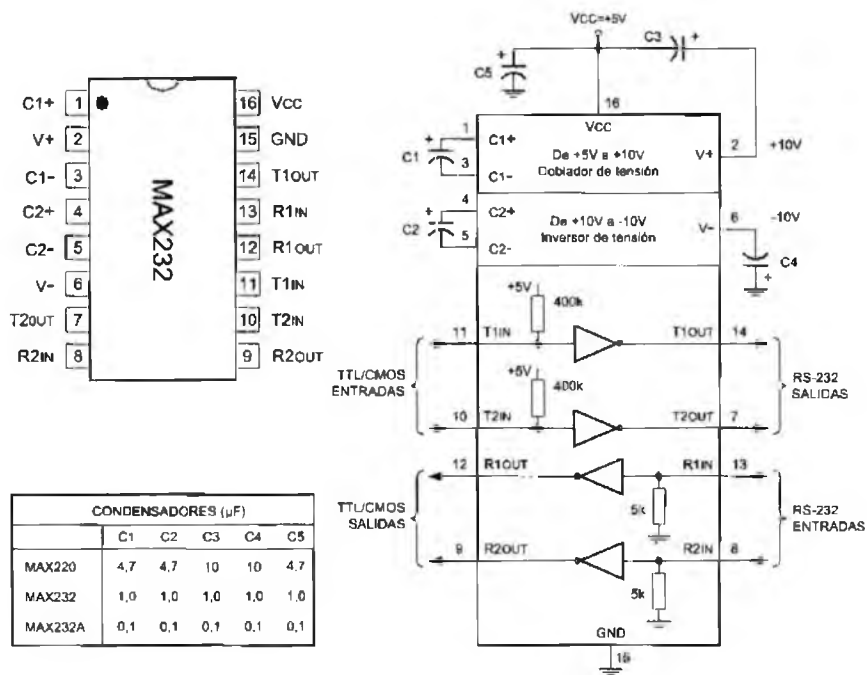


Figura A.5: MAX232.

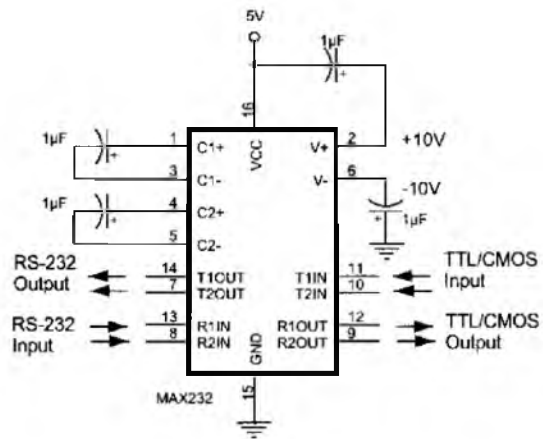


Figura A.6: Conexión típica del MAX232.

Para comunicarse con un microcontrolador se pueden utilizar solo tres patillas del puerto serie RS232. Éstas son el pin de transmisión (TxD), el pin de recepción (RxD) y el pin de masa (SG).

A.4 COMUNICACIÓN DE SERIE ASÍNCRONA:

Para la realización de la comunicación entre el PIC16F877 y la computadora se utilizará el USART, porque es un interfaz de comunicación serial (SCI) que puede funcionar como:

- Asíncrona (Full dúplex, bidireccional)
- Síncrona-Maestro (Half dúplex, unidireccional)
- Síncrona-Esclavo (Half dúplex, unidireccional)

En este caso se ocupará el modo asíncrono porque se adapta a la comunicación entre el microcontrolador y la PC y viceversa.

El funcionamiento del USART de modo asíncrono se explicará a continuación mediante la figura A.7:

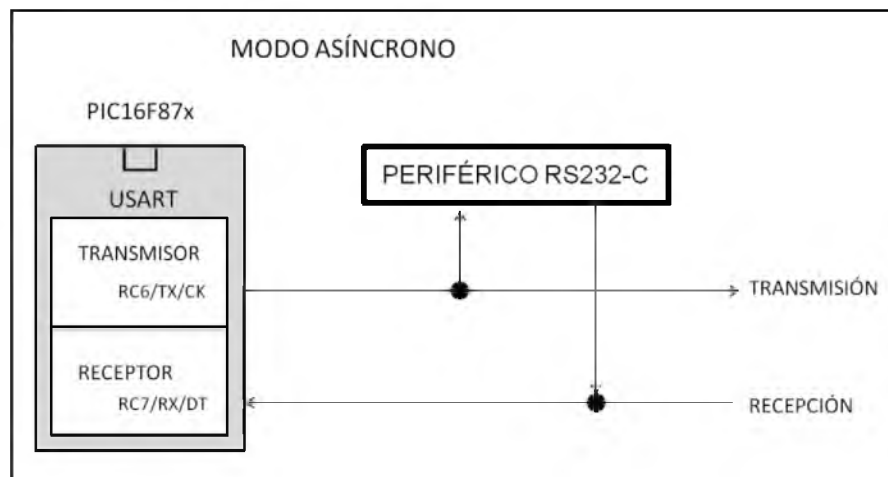


Figura A.7: Modo asíncrono de la comunicación USART en los PIC16F877.

La transferencia de información se realiza sobre dos líneas: TX (transmisión) y RX (recepción); saliendo y entrando los bits por dichas líneas al ritmo de una frecuencia controlada internamente por el USART. Dentro de esta comunicación se dividen dos líneas de más peso de la Puerta C: RC6/TX/CK y RC7/RX/DT.

En esta forma de comunicación se usa la norma RS-232-C, donde cada palabra de información o dato se envía independientemente de los demás. Suele constar de 8 o 9 bits y van precedidos por un bit de START (inicio) y detrás de ellos se coloca un bit de STOP (parada), de acuerdo con las normas del formato estándar NRZ (NonReturn-to-Zero). Los bits se transfieren a una frecuencia fija y normalizada. (Véase figura A.8)

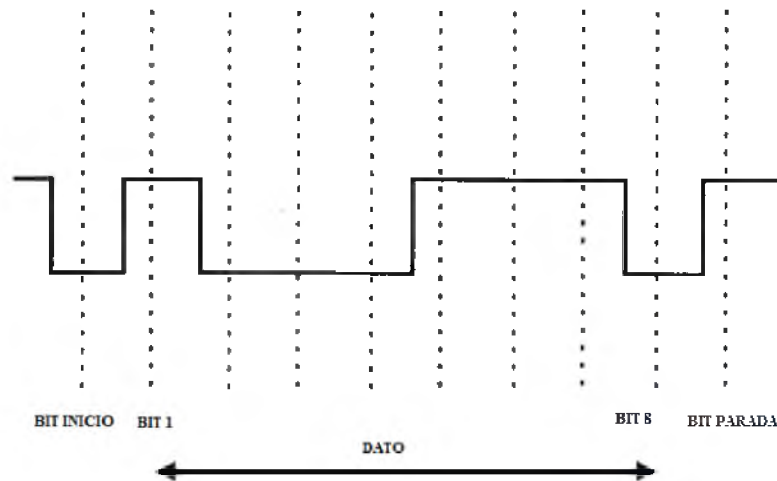


Figura A.8: En el protocolo de comunicación asíncrono, cada palabra de información o dato consta de 8 o 9 bits, estando precedidos por un bit de INICIO y les sigue un bit de PARADA. Los bits se transfieren al ritmo de la frecuencia de reloj establecida.

Los cuatro bloques que configuran la arquitectura del USART, en modo asíncrono, son:

1. Circuito de muestreo.
2. Generador de baudios.
3. Transmisor asíncrono.
4. Receptor asíncrono.

1.-El circuito de muestreo actúa sobre la patita RC7/RX/DT, que es por donde se recibe el bit de información o control y se encarga de muestrear tres veces su valor, para decidir éste por mayoría.

2.-Para que el generador de baudios realice la transferencia de dato, se necesita una frecuencia de baudios (bit por segundo) a un valor normalizado por el protocolo asíncrono RS-232-C, los cuales son los siguientes: 330, 600, 1.200, 2.400, 4.800, 9.600, 19.200, 38.400, etc. Y se puede generar en el USART con el registro del microcontrolador BRG, cuyo valor es controlado por el contenido grabado en el registro SPBRG. Además del valor X cargado en el registro SPBRG, la frecuencia en baudios del generador depende del bit BRGH del registro TXSTA<2>. En el caso de que BRGH sea 0 se trabaja en baja velocidad y si BRGH es igual a 1 se trabaja a alta velocidad. Según este bit se obtendrá el valor de una constante K necesaria en la determinación de la frecuencia de funcionamiento.

$$\text{Frecuencia en baudio} = F_{osc} / (K * (X + 1))$$

X es el valor encargado en el registro SPBRG

Si BRGH=0, baja velocidad y K=64

Si BRGH=1, alta velocidad y K=16

Donde se desprende que:

$$X = F_{osc} / \text{Frecuencia} / k - 1$$

3.- En la figura A.9 se explicará el funcionamiento del transmisor asíncrono.

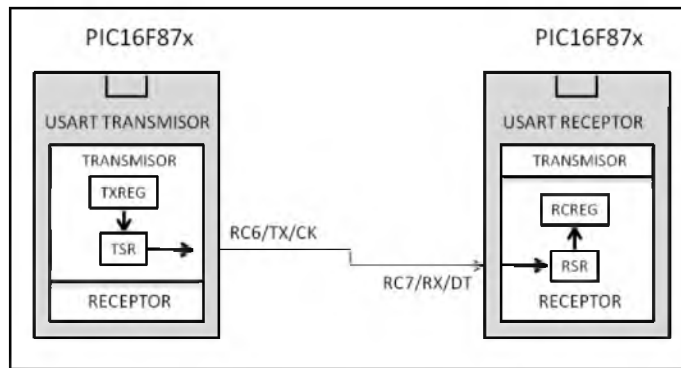


Figura A.9: Esquema simplificado del conexionado entre dos módulos USART funcionando en modo asíncrono.

El dato que se desea transmitir en el USART transmisor, se deposita en el registro TXREG y a continuación se traspa al registro de desplazamiento TSR, que va sacando los bits secuencialmente y a la frecuencia establecida. Además antes de los bits del dato de información incluye un bit de inicio y después de sacar todos los bits añade un bit de parada. El USART receptor recibe uno a uno, los bits, elimina los dos de control y los de información una vez que han llenado el registro de desplazamiento RSR los traslada automáticamente al registro RCREG, donde quedan disponibles para su posterior procesamiento.

4.-En el receptor asíncrono los datos se reciben en serie, bit a bit, por el pin RC7/RX/DT y se van introduciendo secuencialmente en el registro de desplazamiento RSR (figura A.9), que funciona a una frecuencia 16 veces más rápida que la de trabajo. Cuando el dato consta de 9 bits hay que programar el bit RX9=1 y el noveno bit de información se colocará en el bit RX9D del registro RCSTA.

A.4.- COMUNICACIÓN SERIE.

La comunicación serie es una forma muy apreciada de transferir datos digitales entre sistemas y circuitos integrados, dada la reducida cantidad de líneas que precisa. En el PIC16F84, cuando se quería comunicarle en serie con otro dispositivo,

había que destinar varios de sus pines generales de E/S a soportar las líneas de datos y de reloj. También había que desarrollar un programa específico para soportar el protocolo de comunicación.

En los PIC16F87x, Microchip ha implantado en el silicio de sus chips el módulo MSSP (Master Synchronous Serial Port), que proporciona un excelente interfaz de comunicación de los microcontroladores con otros microcontroladores y diversos periféricos, entre los que destacan las memorias EEPROM serie, los conversores A/D, los controladores de displays, etc. De esta forma, el usuario dispone de algunos pines de la Puerta C, que correctamente programadas sirven para manejar directamente la comunicación serie. Además, el módulo MSSP admite dos de las alternativas más usadas en la comunicación serie síncrona:

- 1.- SPI (Serial Peripheral Interface)
- 2.- I2C (Inter-Integrated Circuit)

La comunicación serie en modo SPI la utilizan principalmente las memorias (RAM y EEPROM) y utiliza tres líneas para llevarla a cabo. En el modo I2C sólo se emplean dos líneas y, recientemente, ha conseguido una importante implantación en la comunicación de circuitos integrados, existiendo en el mercado todo tipo de periféricos capaces de trabajar con este protocolo (memorias, controladores, relojes, conversores, etc.).

A.4.1.- Descripción del protocolo I2C.

El protocolo de comunicación serie I2C fue desarrollado por Philips para cubrir sus propias necesidades en la implementación de diversos productos electrónicos que requerían una elevada interconexión de circuitos integrados. El protocolo I2C (Inter-Integrated Circuit) utiliza Únicamente dos líneas para la transferencia de información entre los elementos que se acoplan al bus. Una de dichas líneas se dedica a soportar los datos, es bidireccional y se llama SDA (Serial Data); la

otra lleva los impulsos de reloj para la sincronización, es unidireccional y recibe el nombre de SCL (Serial Clock). Los impulsos de reloj siempre los genera el maestro y tienen la función de sincronizar las transferencias con todos los esclavos colgados a las dos líneas. (Véase figura A.10.)

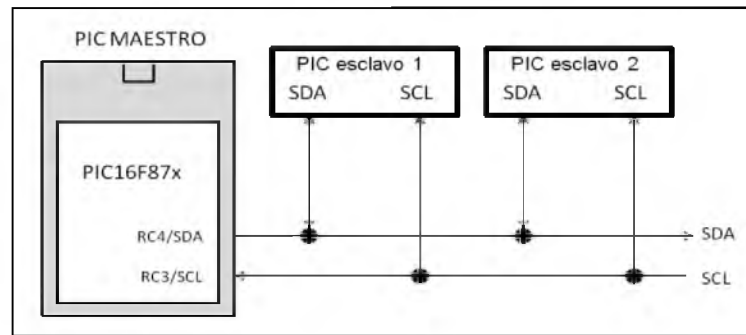


Figura A.10: El maestro se conecta con todos los esclavos en el bus I2C con solo dos líneas. Por SDA circulan los datos en serie y por SCL el maestro envía los impulsos de reloj que genera.

Dos líneas, SDA (datos) y CSL (reloj), transportan la información entre los diferentes dispositivos conectados al bus. Cada dispositivo se identifica por una única dirección y puede transmitir o recibir dependiendo de la operación que se vaya a realizar.

Los dispositivos pueden clasificarse en maestro (master o principal) o esclavo (slave o secundario). El maestro es el que inicia la transferencia de datos y genera la señal del reloj. Cualquiera de los dispositivos direccionados por un maestro se considera un esclavo.

El I2C es un bus multi-maestro; para haber más de un maestro conectado y controlando el bus. Normalmente se trata de microcontroladores o microcomputadores.

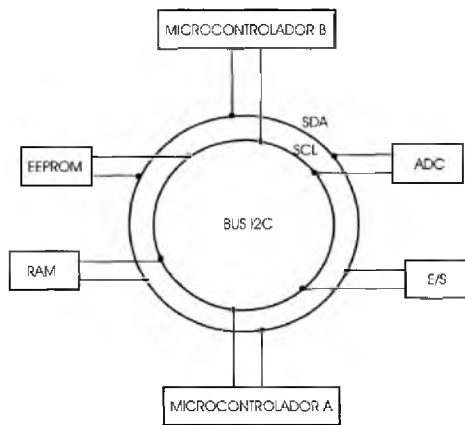


Figura A.11: *Conexiones al bus.*

Supongamos que en un momento dado el microcontrolador A actúa como *maestro*:

1^{er} caso: envía información al B

- a) A, emisor, dirección a B.
- b) A, emisor, envía el dato a B, receptor.
- c) A, termina la transferencia.

2^{do} caso: recibe información desde el B

- a) A, emisor, direcciona a B.
- b) A, receptor, recibe el dato desde el B, transmisor.
- c) A, termina la transferencia.

Esta configuración puede no ser permanente. En otro momento, puede ser el B quien actúe de maestro. Todo depende de la dirección del dato a transferir en un momento dado.

Dada la posibilidad de que existieran varios maestros, podría ocurrir que más de uno inicie la transferencia en un mismo instante de tiempo. Para evitar el caos que pueda producir esta situación, se ha desarrollado un sistema de arbitraje del bus. El procedimiento consiste en una conexión tipo AND entre todos los dispositivos conectados al bus.

Cuando uno o más maestros colocan información en la línea SDA, verifican si el bit que ellos sacan coincide con el nivel lógico de dicha línea. Si un maestro saca un «1» pero la línea SDA está a «0» coincidiendo con un pulso de reloj, pierde la posesión del bus cancelando desde ese momento la transmisión. El nivel lógico «0» (bit dominante) presente en la línea SDA procederá de un maestro distinto.

Las señales de reloj durante el arbitraje del bus son una combinación entre las señales de clock de los distintos maestros conectadas entre sí a la línea SCL mediante una conexión tipo AND.

En forma resumida se podría decir que la terminología del bus I2C consta con los elementos mostrados en la siguiente tabla:

Receptor	Dispositivo que recibe información por la línea SDA.
Maestro	Dispositivo que inicia la transferencia, genera la señal del reloj y finaliza la transferencia.
Esclavo	Dispositivo seleccionado por el maestro.
Multi-maestro	Cuando más de un maestro puede hacerse con el control del bus sin corromper la información.
Arbitraje	Procedimientos que garantizan que, si más de un maestro accede simultáneamente al bus, únicamente uno de ellos se hace cargo del mismo con objeto de no alterar la información.
Sincronización	Procedimiento para sincronizar la señal de reloj en un sistema multi-maestro.

Tabla A.1: Terminología del bus I2C.

A.4.2.- Características Generales.

Tanto las líneas SDA como SCL son líneas bidireccionales que se conectan a +Vdd mediante resistencias de carga pull-up, tal y como se muestra en la figura A.12.

Cuando el bus está libre, ambas líneas están a nivel lógico «1». Los transistores de salida conectados a las líneas del bus I2C deben ser de colector abierto para que todos ellos se puedan conectar entre sí formando una conexión tipo AND.

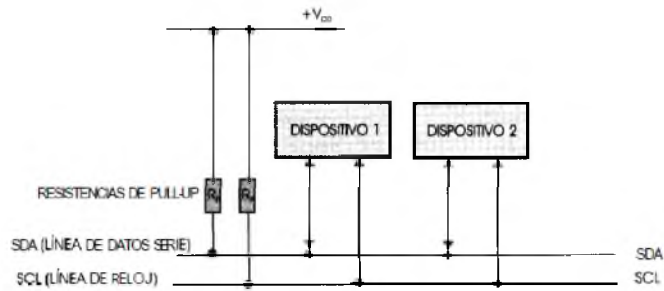


Figura A.12: *Conexión de las resistencias a las líneas bidireccionales.*

Los bits de datos sobre el bus pueden transferirse a una velocidad de 100 Kbits/s. La capacidad máxima en el bus es de 400 pF y el número de dispositivos conectados no debe superarla.

A.4.2.1.- Transferencia de datos.

Formato del byte.

Todos los bytes colocados sobre la línea SDA deben constar de 8 bits. Cada byte va seguido de un bit de reconocimiento, ACK, y comienza siempre por el bit de más peso (MSB), tal y como se muestra en la figura A.13.

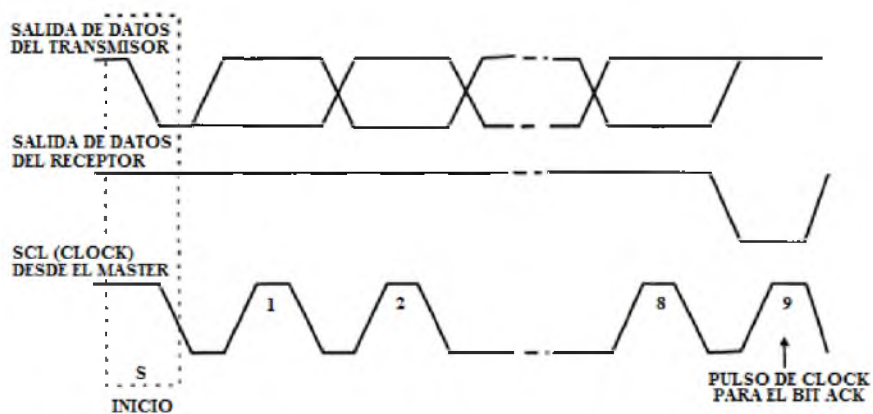


Figura A.13: *Formato del byte.*

Entre un byte y el siguiente puede haber un tiempo «t» de espera que se puede aprovechar para, por ejemplo, dar servicio a una interrupción. En este caso, la línea SCL se mantiene a «0» para forzar al transmisor a un estado de espera.

Un mensaje cualquiera puede finalizar mediante la generación de la condición de parada durante la transferencia de un byte. En este caso no se generará el bit de reconocimiento ACK.

Reconocimiento

El bit de reconocimiento ACK es obligatorio en la transferencia de cada byte. El pulso de reloj asociado a este bit n° 9 lo genera el maestro. El transmisor pone la línea SDA a «1» durante dicho pulso de reloj.

El receptor, por su parte, pone a «0» la línea SDA durante el pulso de reloj correspondiente al ACK y lo mantiene estable durante todo el período de dicho pulso. En la línea SDA prevalece por tanto el nivel «0», como se aprecia en la figura A.14.

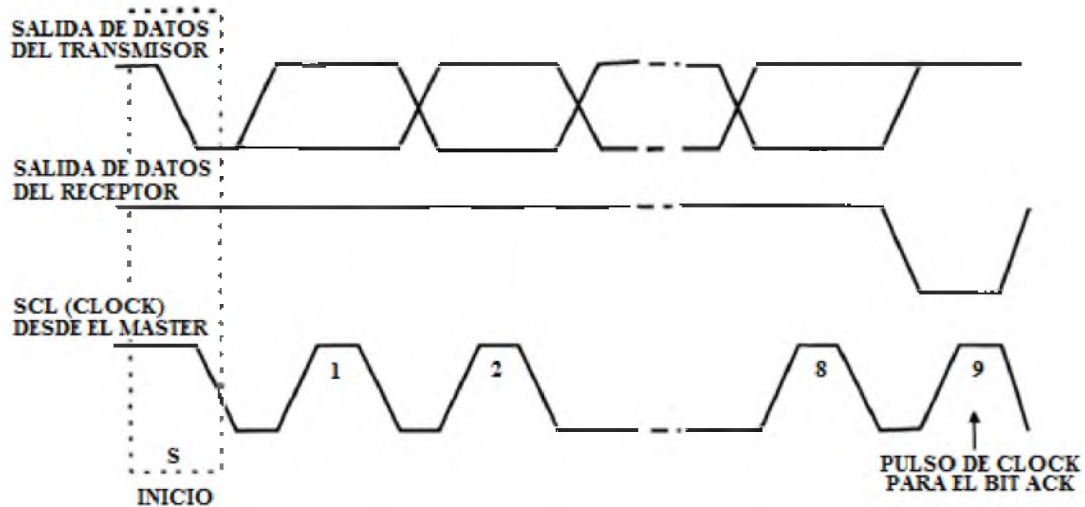


Figura A.14: *El reconocimiento.*

A.4.2.2.- Formato

Los datos que se transfieren tienen el formato mostrado en la siguiente figura:

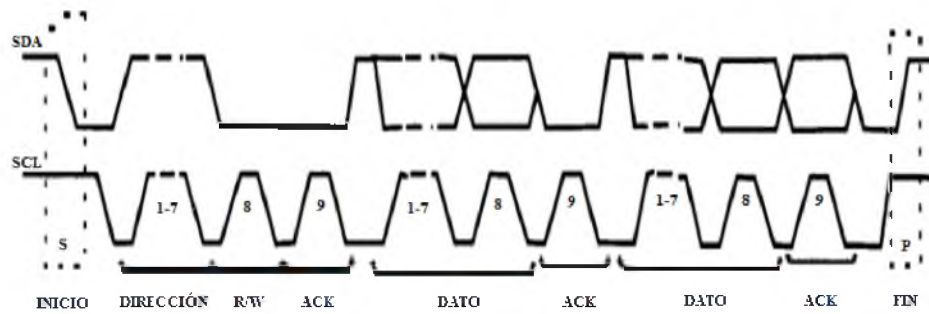


Figura A.15: *Formato de los datos transferidos.*

Tras la condición de inicio, el maestro envía la dirección del esclavo al que se quiere dirigir. Esta dirección es de 7 bits más el bit R/W# que indica lectura (R/W# = 1) o escritura (R/W# = 0) del esclavo. Toda transferencia de datos finaliza con la condición de parada, también generada por el maestro. A pesar de todo, si un maestro aún desea comunicar por el bus, puede generar otra condición de inicio y direccionar a otro esclavo sin generar previamente la condición de parada.

A continuación se muestran en las figuras A.16, A.17 y A.18 los posibles formatos de transferencia.

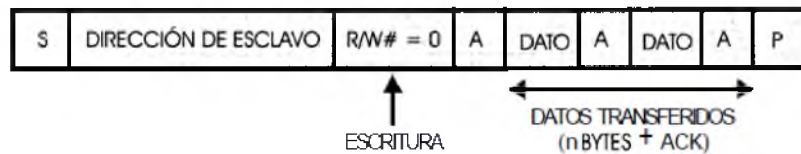


Figura A.16: *El maestro transmite al esclavo receptor. No cambia el byte de dirección.*

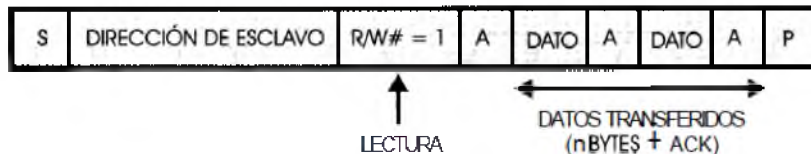


Figura A.17: *El maestro recibe desde el esclavo después de enviarle a éste la dirección.*

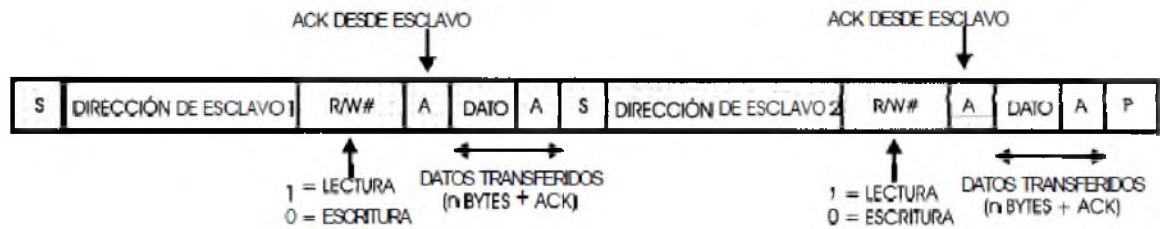


Figura A.18: El maestro cambia el byte de dirección y selecciona un esclavo distinto.

Tras el primer ACK enviado por el esclavo, el maestro transmisor se convierte en receptor y el esclavo receptor en transmisor.

El formato combinado de la figura A.18 permite seleccionar un esclavo para leerlo, por ejemplo, y luego para escribirlo. En el caso de un dispositivo I2C de tipo memoria RAM puede ser útil.

El tipo de datos o comandos que se envían pueden servir para realizar distintos tipos de operaciones según el dispositivo I2C empleado. Habrá que consultar las características y posibilidades de cada cual.

A.4.3.- Direccionamiento.

El proceso de direccionamiento del bus I2C consiste en que el primer byte que envía el maestro tras la condición de inicio es un código que determina y selecciona a un determinado esclavo. Existe una excepción en el código denominada «llamada general». Cuando un maestro realiza una «llamada general» (código 0000 000) todos los esclavos existentes deben responder con el bit ACK. Puede darse el caso de dispositivos que ignoren esta llamada. El siguiente byte tras la misma indicará la operación a realizar.

A.4.3.1.- Definición del primer byte.

Los 7 bits de más peso del primer byte se emplean para direccionar a un determinado esclavo. El de menos peso, el octavo enviado, determina si se realizará

una operación de lectura o de escritura (R/W#) sobre el esclavo direccionado, tal y como se muestra en la figura A.19.

MSB		LSB
S	DIRECCIÓN DEL ESCLAVO 1	R/W#

DIRECCIÓN	R/W#	FUNCIÓN
0000 000	0	Dirección de <<llamada general>>
0000 000	1	Byte de inicio (2)
0000 001	x	Dirección CBUS (3)
0000 010	x	Reservada para un formato diferente (4)
0000 011	x	No definida
0000 100	x	No definida
0000 101	x	No definida
0000 110	x	No definida
0000 111	x	No definida

- (1) Para aquellos que la admitan y/o necesiten.
- (2) No esta permitido el bit ACK tras la recepción del byte. Empleado por ciertos dispositivos.
- (3) La dirección CBUS está reservada para permitir la combinación de dispositivos I2C y CBUS en un mismo sistema. Los dispositivos I2C no responden a esta dirección.
- (4) Esta dirección está reservada para combinar el formato I2C con otros protocolos. Únicamente los dispositivos I2C que estén diseñados para trabajar con esos formatos y protocolos responden a esta dirección.

Figura A.19: *El primer byte.*

Cuando el maestro envía una dirección por el bus, todos los esclavos conectados la comparan con la suya propia interna. Aquel en que coincida se considerará seleccionado por el maestro.

De los 7 bits que forman una dirección hay una parte que son fijos y ya están definidos internamente por el propio dispositivo esclavo, y otra parte que son programables. De esta forma, es posible conectar al mismo bus dispositivos idénticos cuya parte fija de dirección es la misma pero la programable es diferente.

El número de dispositivos iguales que se pueden instalar a un mismo bus depende del número de bits programables disponibles. Por ejemplo, si un determinado modelo de dispositivo recibe una dirección de 7 bits, 4 de los cuales son fijos y los 3 restantes programables, se podrán conectar al bus un máximo de 8 dispositivos de ese mismo modelo.

El comité 12C creado al efecto coordina las direcciones asignadas a los diferentes dispositivos. La dirección 1 11 1 11 1 está reservada como extensión de dirección en el que el proceso de direccionamiento de un dispositivo continua en los siguientes bytes transferidos por el maestro. La dirección 11 11 xxx está reservada también con propósitos de direccionamientos extendidos y la emplean ciertos dispositivos que tienen 10 bits de direccionado. La dirección 0000 xxx está reservada para un grupo especial de dispositivos.

APENDICE B

PROGRAMAS PARA LA COMUNICACIÓN

B.1 TRANSMISIÓN DE DATOS

```
#include <16F877A.h>
#fuses XT,NOWDT,NOPROTECT,NOLVP
#use delay(clock= 4000000)
#use i2c(master,sda=PIN_C4,scl=PIN_C3,force_hw,fast)

#include <lcd.c>    // PARA UN LCD DE 16*2

#byte TRISA = 0x85
#byte TRISB = 0x86
#byte PORTB = 0x06
#byte TRISC = 0x87
#byte PORTC = 0x07
#byte TRISD = 0x88
#byte PORTD = 0x08
#byte TRISE = 0x89

#define sensor_00 0XE0 //ahora se llamará sensor_00 por ser la DIR base
#define sensor_01 0XE2 //a partir de aquí empezarán sensor_01, sensor_02...
#define sensor_02 0XE4
#define sensor_03 0XE6
#define sensor_04 0XE8
#define sensor_05 0XEA
#define sensor_06 0XEC
#define sensor_07 0XEE
#define sensor_08 0XF0
```

```

#define sensor_09 0XF2
#define sensor_10 0XF4
#define sensor_11 0XF6
#define sensor_12 0XF8
#define sensor_13 0XFA
#define sensor_14 0XFC
#define sensor_15 0XFE

#define inch 0X50
#define cm 0X51
#define useg 0X52

#define Salida = 0x05.1 //el dato sale por RA1

int temp,tempa;
float motorder=0.0, motorizq=0.0,dato32;

void inicia_sensor(int dir){
    i2c_start(); //condicion de inicio del I2C
    i2c_write(dir); //direccion del sensor_0n
    i2c_write(0x00); //Escritura de comando (registro 0)
    i2c_write(cm); //medicion en "cm" o "inch"
    i2c_stop();
    delay_ms(100);
}

int16 lee_sensor(int dir){
    int8 lee_01H = 0, lee_01L = 0;
    int16 lectura;
    inicia_sensor(dir); //llama la funcion de inicio del sensor

```

```

i2c_start();
i2c_write(dir);
i2c_write(0x02);    // registro del alcance
i2c_write(0x18);    //0x18 = 1mt, 0x8c =6mts (por defecto 255 a 11mts)
i2c_write(0x01);    //registro de la ganancia maxima (por defecto 16)
i2c_write(0x14);    //0x12 = 350, 0x14 = 500, 0x16 =700
i2c_start();
i2c_write(dir|0x01); //instruccion de lectura del sensor n
lee_01H = i2c_read();
//i2c_start();
lee_01L = i2c_read(0); //el cero le dice que no envíe el ACK
i2c_stop();
lectura = make16(lee_01H,lee_01L); //concatenacion del paquete de bytes
return lectura;
}

```

```

void cambia_dir(int nueva_dir, int ant_dir){
    i2c_start();        //Secuencia de inicio
    i2c_write(ant_dir); //Dirección I2C del SRF10 modo escritura
    i2c_write(0x00);    //Dirección interna 0x00 (registro de comandos)
    i2c_write(0xA0);    //1ª secuencia para el cambio de dirección
    i2c_stop();         //Secuencia de stop

    i2c_start();        //Secuencia de inicio
    i2c_write(ant_dir); //Dirección I2C del SRF10 modo escritura
    i2c_write(0x00);    //Dirección interna 0x00 (registro de comandos)
    i2c_write(0xAA);    //2ª secuencia para el cambio de dirección
    i2c_stop();         //Secuencia de stop

    i2c_start();        //Secuencia de inicio

```

```

i2c_write(ant_dir);    //Dirección I2C del SRF10 modo escritura
i2c_write(0x00);      //Dirección interna 0x00 (registro de comandos)
i2c_write(0xA5);      //3ª secuencia para el cambio de dirección
i2c_stop();           //Secuencia de stop

i2c_start();          //Secuencia de inicio
i2c_write(ant_dir);   //Dirección I2C del SRF10 modo escritura
i2c_write(0x00);      //Dirección interna 0x00 (registro de comandos)
i2c_write(nueva_dir); //Nueva dirección I2C
i2c_stop();           //Secuencia de stop
}

void config(){
    lcd_init();
    setup_timer_0(RTCC_EXT_L_TO_H|RTCC_DIV_1);
    setup_timer_1(T1_EXTERNAL|T1_DIV_BY_1);
    set_tris_a(0b00010001);
    set_tris_b(0xFF);
    set_tris_c(0x01);
    set_tris_d(0x00);
    PORTB = 0;
    PORTC = 0;
    PORTD = 0;
}

/////Función de transmisión de datos para los encoders de 32 bits con su bit de
///// sincronía

void tx32 (float dato32){    // int16 <=> 32(float) dato32
int i;

```

```

Salida = 0;           // Asegura inicio en cero.
Salida = 1;  delay_ms(1); // Pulso alto de 10 mseg.
Salida = 0;  delay_ms(1); // Pulso bajo de 10 mseg.
for(i=32;i>0;i--){    // 16 <=> 32
  if(bit_test(dato32,i-1) == 1 ){
    Salida = 1;    delay_us(300);//delay_ms(3); // Envía un pulso alto de 3 mseg.
    Salida = 0;    delay_us(100);//delay_ms(1); // Envía un pulso bajo de 1mseg.
  }
  Else
  {
    Salida = 1;  delay_us(100);//delay_ms(1); // Envía un pulso alto de 1 mseg.
    Salida = 0;  delay_us(300);//delay_ms(3); // Envía un pulso bajo de 3mseg.
  }
}
delay_us(100);       // Pulso bajo de 10ms entre trenes de pulsos
}

```

///// Función de transmisión de datos de los sensores de 16 bits con su bit de
 ///// sincronía

```

void tx16 (int16 dato16){ // int16 <=> 32(float) dato32
int i;
  Salida = 0;           // Asegura inicio en cero.
  Salida = 1;  delay_ms(1); // Pulso alto de 10 mseg.
  Salida = 0;  delay_ms(1); // Pulso bajo de 10 mseg.
  for(i=32;i>0;i--){    // 16 <=> 32
    if(bit_test(dato16,i-1) == 1 ){
      Salida = 1;    delay_us(300);//delay_ms(3); // Envía un pulso alto de 3 mseg.

```

```

        Salida = 0;      delay_us(100);//delay_ms(1); // Envía un pulso bajo de 1
mseg.
    }
    Else
    {
        Salida = 1;  delay_us(100);//delay_ms(1); // Envía un pulso alto de 1 mseg.
        Salida = 0;  delay_us(300);//delay_ms(3); // Envía un pulso bajo de 3
mseg.
    }
}
delay_us(100);      // Pulso bajo de 10ms entre trenes de pulsos

}
///// Programa principal...

```

```

void main() {
    int16 valor[5]; int16 aux; int i=0; float aux32;
    config();
    printf(lcd_putc,"Iniciando ...");
    delay_ms(1000);
    temp = PORTB;
    set_timer0(0);
    set_timer1(0);
    while (1) {

// aqui empieza la lectura de los encoders...
        motorder = (1.125*get_timer0());
        lcd_gotoxy(3,1);
        printf(lcd_putc,"mDer = %f cm",motorder);
        tx32(motorder);
    }
}

```

```
motorizq = (1.125*get_timer1());  
lcd_gotoxy(3,2);  
printf(lcd_putc,"mIzq = %f cm",motorizq);  
tx32(motorizq);
```

```
// aqui empieza la lectura de los sensores en el lcd...
```

```
valor[i] = lee_sensor(sensor_01);  
lcd_gotoxy(5,1);  
printf(lcd_putc,"Snsr1 = %lu cm",valor[i]);  
tx16(valor[i]);  
i++;
```

```
valor[i] = lee_sensor(sensor_02);  
lcd_gotoxy(5,1);  
printf(lcd_putc,"Snsr2 = %lucm",valor[i]);  
tx16(valor[i]);  
i++;
```

```
valor[i] = lee_sensor(sensor_03);  
lcd_gotoxy(5,1);  
printf(lcd_putc,"Snsr2 = %lucm",valor[i]);  
tx16(valor[i]);  
i++;
```

```
valor[i] = lee_sensor(sensor_04);  
lcd_gotoxy(5,1);  
printf(lcd_putc,"Snsr4 = %lucm",valor[i]);  
tx16(valor[i]);
```

```

    i++;

    valor[i] = lee_sensor(sensor_05);
    lcd_gotoxy(5,1);
    printf(lcd_putc,"Snsr2 = %lucm",valor[i]);
    tx16(valor[i]);
    i++;

}
}

// CAMBIO DE DIRECCION (no mover a menos que sea necesario)
//cambia_dir(sensor_04,sensor_03); //dir_nueva,dir_vieja
//delay_ms(100);

```

B.2 RECEPCIÓN DE DATOS.

```

#include <16f877a.h>

#fuses XT,NOWDT,NOPROTECT,NOLVP
#use delay(clock=4000000)
#use rs232(baud = 9600, xmit = pin_c6, rcv = pin_c7, bits = 8)

#define use_portb_lcd TRUE
#include<lcd.c>

#byte TRISA = 0x85
#byte TRISB = 0x86
#byte PORTB = 0x06
#byte TRISC = 0x87
#byte PORTC = 0x07

```

```

#byte TRISD = 0x88
#byte PORTD = 0x08
#byte TRISE = 0x89

#bit Entrada = 0x05.0 // Valores del registro del PIC.
#use fast_io(A)

int16 Tiempo; // Es el valor leído del Timer1, acumulado
int16 dato16; // dato recibido de 16 Bits.
int16 sensor01, sensor02, sensor03, sensor04, sensor05;
int8 i, hdata=0, ldata=0;
float dato32=0,dato32a=0.0,dato32b=0.0; //dato recibido de 32 Bits.

float rx32(){
    int i=0;
    Tiempo = 0;
    while(Tiempo < 900 ){ // Detectar el primer pulso alto de 10 mseg.
        while(Entrada == 0); // Está bajo y espera que suba
        set_timer1(0); // Pone a cero el Timer 1.
        while(Entrada == 1); // Se mantiene en el pulso alto.
        Tiempo = get_timer1(); // Toma el tiempo en alto, al salir de alto, en
microsegundos.
    }//fin while Tiempo de sincronia
    set_timer1(0);
    Tiempo=0;
    for (i=32;i>0;i--){ // 16 <=> 32
        while(Entrada == 0);
        set_timer1(0);
        while(Entrada == 1);
    }
}

```

```

    Tiempo = get_timer1(); // Num-=1; es Num = Num -1;
    if((Tiempo) > 280)
        { bit_set(dato32,i-1);}
    if((Tiempo) < 110)
        { bit_clear(dato32,i-1);}
} // Fin del for
return dato32;
}

int rx16(){
    int i=0;
    Tiempo = 0;
    while(Tiempo < 900 ){ // Detectar el primer pulso alto de 10 mseg.
        while(Entrada == 0); // Está bajo y espera que suba
        set_timer1(0); // Pone a cero el Timer 1.
        while(Entrada == 1); // Se mantiene en el pulso alto.
        Tiempo = get_timer1(); // Toma el tiempo en alto, al salir de alto, en
            //microsegundos.
    } //fin while Tiempo de sincronia
    set_timer1(0);
    Tiempo=0;
    for (i=16;i>0;i--){ // 16 <=> 32
        while(Entrada == 0);
        set_timer1(0);
        while(Entrada == 1);
        Tiempo = get_timer1();
        if((Tiempo) > 280)
            { bit_set(dato16,i-1);}
        if((Tiempo) < 110)
            { bit_clear(dato16,i-1);}
    }
}

```

```

    }// Fin del for
    return dato16;
}

void configuracion(void){
    setup_adc_ports(NO_ANALOGS);
    setup_adc(ADC_OFF);
    setup_timer_0(RTCC_INTERNAL|RTCC_DIV_1);
    setup_timer_1(T1_INTERNAL|T1_DIV_BY_1);
    lcd_init();
    set_tris_a(0b00000001); // 1 entrada 0 salida.
}
//*****
*****

void main()
{
    configuracion();
    printf(lcd_putc,"iniciando Rx ...");
    delay_ms(1000);
    while(true){

        dato32a=rx32();
        lcd_gotoxy(1,1);
        printf(lcd_putc," MDer: %f",dato32a);

        printf(lcd_putc,"/f");

        dato32b=rx32();
        lcd_gotoxy(1,2);
        printf(lcd_putc," MIzq: %f",dato32b);
    }
}

```

```
printf(lcd_putc, "/f");
```

```
sensor01=rx16();  
lcd_gotoxy(1,1);  
printf(lcd_putc, "snsr01: %lu", sensor01);  
delay_ms(1000);
```

```
printf(lcd_putc, "/f");
```

```
sensor02=rx16();  
lcd_gotoxy(1,1);  
printf(lcd_putc, "snsr01: %lu", sensor02);  
delay_ms(1000);
```

```
printf(lcd_putc, "/f");
```

```
sensor03=rx16();  
lcd_gotoxy(1,1);  
printf(lcd_putc, "snsr01: %lu", sensor03);  
delay_ms(1000);
```

```
printf(lcd_putc, "/f");
```

```
sensor04=rx16();  
lcd_gotoxy(1,1);  
printf(lcd_putc, "snsr01: %lu", sensor04);  
delay_ms(1000);
```

```
printf(lcd_putc, "/f");
```

```
sensor05=rx16();
lcd_gotoxy(1,1);
printf(lcd_putc,"snsr01: %lu",sensor05);
delay_ms(1000);

printf(lcd_putc,"/f");

} // Fin del while(True)
} // Fin del Main
//*****
*****
```