



ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

FACULTAD DE INFORMÁTICA Y ELECTRÓNICA

**ESCUELA DE INGENIERÍA ELECTRÓNICA EN
TELECOMUNICACIONES Y REDES**

*“DISEÑO E IMPLEMENTACIÓN DE UN ROBOT ÁPODO
(GUSANO) CONTROLADO INALÁMBRICAMENTE”*

TESIS DE GRADO

Previa la obtención del título de

INGENIERO EN ELECTRÓNICA Y COMPUTACIÓN

Presentado por:

MAYRA ALEXANDRA AYALA VIMOS

*RIOBAMBA – ECUADOR
2011*

Agradecimiento

Agradezco a la ESCUELA SUPERIOR POLITECNICA DE CHIMBORAZO, y a todos los docentes que me brindaron sus conocimientos y como personas su amistad sincera.

A mi tutor el Ing. Paul Romero por brindarme su apoyo, comprensión y conocimientos.

A todos mis *profes* no sólo de la carrera sino de toda la vida, mil gracias porque de alguna manera forman parte de lo que ahora soy.

Son muchas las personas especiales a quienes me gustaría agradecer su amistad, apoyo, ánimo y compañía en las diferentes etapas de mi vida. Algunas están aquí conmigo y otras en mis recuerdos y en el corazón. Sin importar en dónde estén o si alguna vez llegan a leer estas palabras quiero darles las gracias por formar parte de mí, por todo lo que me han brindado y por todas sus bendiciones.

Al más especial de todos, a ti Señor porque hiciste realidad este sueño, por todo el amor con el que me rodeas y porque me tienes en tus manos. Esta tesis es para ti.

Dedicatoria

A las personas más importantes de mi vida:

A Ti Mami, no me equivoco si digo que eres la mejor mamá del mundo, porque a pesar de las adversidades que tuve que enfrentar jamás tuve temor de hacerlo porque sabes qué? Ahí estabas tú para brindarme confianza y apoyo, porque cuando pensaba desfallecer bastaban tan solo palabras de amor para iluminar mi camino por eso y más; gracias Madre Amada por siempre.

A Ti Papi, éste es un logro muy importante que quiero compartir contigo, fueron tus recuerdos el motor principal que me impulsaron a alcanzar uno de mis sueños. Aunque no estuviste físicamente conmigo yo sé que junto a ese ser tan hermoso que nos facultó la vida hiciste de mí lo que hoy soy.

Jordana Mayte, el amor de mi vida aunque todavía no puedes leer, un día vas a aprender y por eso también te dedico esta tesis, gracias por haber llegado cuando más te necesite, eres la bendición más grande de mi vida. Te Amo mi negrita.

Y a la persona más representativa de mi vida, mi tío Carlos Julio Bonifaz, quien fue padre, guía, amigo, el pilar importante en este logro tan ansiado. Este donde este mi corazón y gratitud por siempre y para siempre.

Por último dedico a todos los seres que luchan por alcanzar sus metas y creen que la educación abre las puertas de un mejor porvenir.

Firmas de Responsabilidad

NOMBRE	FIRMA	FECHA
Ing. Iván Menes DECANO DE LA FACULTAD DE INFORMÁTICA Y ELECTRÓNICA	_____	_____
Ing. Pedro Infante DIRECTOR DE LA ESCUELA DE INGENIERÍA ELECTRÓNICA EN TELECOMUNICACIONES Y REDES	_____	_____
Ing. Paúl Romero DIRECTOR DE TESIS	_____	_____
Ing. Edwin Altamirano MIEMBRO DEL TRIBUNAL	_____	_____
Tlgo. Carlos Rodríguez DIRECTOR DEL CENTRO DE DOCUMENTACIÓN	_____	_____
NOTA DE LA TESIS	_____	

Firma de Autoría

“Yo, MAYRA ALEXANDRA AYALA VIMOS, soy responsable de las ideas, doctrinas y resultados expuestos en esta tesis; y, el patrimonio intelectual de la Tesis de Grado pertenece a la ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO”.

Mayra Alexandra Ayala Vimos.

AUTOR

Índice de Abreviaturas

ALU	Unidad Aritmético Lógica.
A/D.	Analógico/Digital.
ASK	(Amplitude Shift Keying)Modulación por desplazamiento de Amplitud
CI.	Circuito Integrado.
D/A.	Digital/Analógico
E/S	Entrada/Salida
EEPROM	Memoria de solo lectura programable y borrrable eléctricamente.
FAMOS	Floating Gate Avalanche-Injection Metal Oxide Semiconductor
FLASH	Memoria no volátil de bajo consumo.
FSK	(Frecuency Shift Keying) Modulación por desplazamiento de Frecuencia
GPR	Registros de propósito General.
HS	Cristal de alta velocidad.
HSEROUT	Salida serial de hardware asíncrono.
INTOSC	Oscilador Interno.
I/O.	Input/Output.
Led	Diodo emisor de luz
LP	Cristal para baja potencia y bajo consumo de corriente.
MCLR	Master Clear (Reset).
MHz	Megahercios.
Mbps	Mega Bits por Segundo.
MHz.	Megahercios
mA	miliamperios
ms	milisegundos
mV	mili voltios
MPSK	(Multiple Phase Shift Keying)Modulación por desplazamiento de Fase Multiple

OSC1/CLKIN	Entrada del oscilador (cristal). Entrada de oscilador externo.
PIC	Peripheral Interface Controller.
SFR	Registros de Función Específica.
USART	Transmisor/Receptor Asíncrono Universal.
Vcc	Voltios de Corriente continúa
VCA	Voltios de corriente Alterna

ÍNDICE GENERAL

<u>ÍNDICE GENERAL</u> -----	8 -
<u>ÍNDICE DE FIGURAS</u> -----	14 -
<u>ÍNDICE DE TABLAS</u> -----	18 -
<u>CAPÍTULO I</u> -----	21 -
MARCO REFERENCIAL -----	21 -
1. ANTECEDENTES -----	21 -
2. JUSTIFICACIÓN-----	22 -
3. OBJETIVOS-----	22 -
1.3.1. OBJETIVO GENERAL-----	22 -
1.3.2. OBJETIVOS ESPECIFICOS -----	23 -
4. HIPÓTESIS -----	23 -
<u>CAPÍTULO II</u> -----	24 -
FUNDAMENTO TEÓRICO -----	24 -
2.1. LOCOMOCIÓN -----	24 -
2.1.1. Introducción -----	24 -
2.1.1.1. Tipos de locomoción-----	26 -
2.1.1.2. Locomoción de robots-----	26 -
2.1.1.3. Diseño de robots móviles-----	27 -
2.1.1.4. El problema de la locomoción -----	28 -
2.1.1.5. Robots ápodos-----	30 -
2.2. SENSORES-----	31 -
2.2.1. Introducción -----	31 -
2.2.2. Criterios para la Selección de Sensores -----	31 -
2.2.2.1. Sistemas Actuadores-----	32 -
2.2.2.2. Sistemas de Interfaz entre Sensores y Actuadores -----	33 -

2.2.3.	Sensores Sharp.	34
2.2.4.	Sensores Sharp en distintas familias.	34
2.2.4.1.	Reflexivo IS471F	34
2.2.4.2.	Sharp GP2Y0A02YK	36
2.2.4.3.	La Familia GP2DXX	38
2.2.4.4.	El sensor GP2D12.	40
2.2.4.4.1.	Curvas Características del GP2D12	47
2.3.	LOS SISTEMAS DE COMUNICACIÓN	49
2.3.1.	TRANSMISIÓN DE ONDAS DE RADIO	49
2.3.2.	COMPONENTES DE UN SISTEMA DE COMUNICACIÓN	52
2.3.3.	COMUNICACIÓN INALÁMBRICA	52
2.3.3.1.	Aspecto histórico y generalidades	53
2.3.3.2.	Campos de Utilización	54
2.3.3.3.	Algunos problemas asociados con la tecnología inalámbrica	54
2.3.3.4.	Equipo inalámbrico	54
2.4.	RED INALÁMBRICA	55
2.4.1.	Categorías	55
2.4.2.	Aplicaciones	55
2.5.	RADIOFRECUENCIA	56
2.5.1.	Características Generales	57
2.5.2.	Tipos de Comunicaciones Inalámbricas por RF	58
2.6.	MOTORES	58
2.6.1.	Servo Motores HITEC HS-311	59
2.6.1.1.	Características	59
2.6.1.2.	Control	60
2.6.1.3.	Conexión	61
2.6.1.4.	Driver de Potencia para Control de Motores	61

<u>CAPÍTULO III</u>	62 -
MICROCONTROLADORES APROPIADOS PARA EL CIRCUITO.....	62 -
3.1. MICROCONTROLADORES.....	62 -
3.1.1. Introducción	62 -
3.1.2. Definición de Microcontrolador	63 -
3.1.3. Características de los Microcontroladores	64 -
3.1.4. Tipos de Arquitecturas de Microcontroladores	64 -
3.1.4.1. Arquitectura Von Neumann	64 -
3.1.4.2. Arquitectura Harvard	65 -
3.1.5. Las Gamas de Pic.....	65 -
3.1.5.1. Gama Enana PIC12C (F) XXX.....	66 -
3.1.5.2. Gama Baja PIC16C5X	66 -
3.1.5.3. Gama Media PIC16CXXX	67 -
3.1.5.4. Gama Alta PIC17CXXX.....	68 -
3.1.6. Herramientas para el desarrollo de aplicaciones	68 -
3.1.6.1. Ensamblador.....	69 -
3.1.6.2. Compilador	69 -
3.1.6.3. Simulador	69 -
3.1.6.4. Programador.....	70 -
3.1.6.5. Paquetes IDE.....	70 -
3.2. MÉTRICAS PARA ELEGIR UN MICROCONTROLADOR	70 -
3.2.1. Estudio de la mejor alternativa de microcontrolador.	71 -
3.3. MICROCONTROLADOR 16F628A.....	73 -
3.3.1. Principales características.....	73 -
3.3.2. Características especiales:	74 -
3.3.3. Descripción de los terminales.....	74 -
3.3.3.1. Terminales de entrada-salida (16 en total)	74 -

3.3.3.2.	PORTA: RA0-RA7: -----	- 75 -
3.3.3.3.	PORTB: RB0-RB7:-----	- 75 -
3.3.3.4.	Otros terminales. -----	- 75 -
3.3.4.	Aspecto Interno del PIC 16F628. -----	- 76 -
3.3.5.	El Procesador.-----	- 77 -
3.3.6.	La Memoria de Datos.-----	- 78 -
3.3.7.	Direccionamiento de la Memoria de Datos.-----	- 79 -
3.3.7.1.	Direccionamiento Directo: -----	- 79 -
3.3.7.2.	Direccionamiento Indirecto: -----	- 79 -
3.3.8.	La memoria de Programa. -----	- 79 -
3.3.9.	El Reloj-----	- 80 -
3.4.	MICROCONTROLADOR 16F877A.-----	- 80 -
3.4.1.	Puertos paralelos de entrada / salida del Pic 16f877a.-----	- 82 -
3.4.1.1.	Puerto A (Porta) -----	- 82 -
3.4.1.2.	Puerto B (Portb) -----	- 83 -
3.4.1.3.	Puerto C (Portc) -----	- 84 -
3.4.1.4.	Puertos D y E.-----	- 85 -
3.4.1.5.	Puerto Serie Usart. -----	- 85 -
3.4.2.	Configuración de la Usart del Pic 16F877A.-----	- 86 -
3.4.3.	Proceso de Conversión Analógico/Digital.-----	- 86 -
3.4.4.	Registros de Control del Convertidor A/D del Pic 16F877A. -----	- 87 -
3.4.5.	Los Registros de Resultados del Convertidor A/D del Pic 16f877a-----	- 87 -
3.4.6.	El Módulo Temporizador del Pic 16F877A. -----	- 88 -
	<u>CAPÍTULO IV</u> -----	- 89 -
	DISEÑO Y CONSTRUCCIÓN -----	- 89 -
4.1.	ESQUEMA GENERAL DE FUNCIONAMIENTO -----	- 89 -

4.2.	DISEÑO Y DESCRIPCIÓN DE LOS BLOQUES QUE COMPONEN EL SISTEMA DEL ROBOT ÁPODO	90
4.2.1.	Etapa de procesamiento del Transmisor	90
4.2.1.1.	Algoritmo del Transmisor	90
4.2.1.2.	Circuito Transmisor Inalámbrico en Isis/Proteus	91
4.2.2.	Etapa de Transmisión y Recepción	92
4.2.3.	Software De Transmisión De Datos	93
4.2.4.	Etapa de Procesamiento del Receptor	96
4.2.4.1.	Algoritmo del Receptor	97
4.2.4.2.	Circuito Receptor Inalámbrico En Isis/Proteus	98
4.2.4.3.	Etapa de Ejecución	99
4.3.	ESTRUCTURA Y DISEÑO DEL ROBOT	100
4.3.1.	Diseño del programa.	101
	<u>CAPÍTULO V</u>	102
	ANÁLISIS Y RESULTADOS	102
5.1.	FUNCIONAMIENTO GENERAL	102
5.1.1.	Proceso de Comunicación	102
5.1.2.	Sistema de Locomoción	102
5.1.3.	Algoritmos de Control	103
5.2.	MOTORES UTILIZADOS	103
5.3.	SENSOR INFRARROJO EMPLEADO	104
5.4.	ALIMENTACIÓN PARA EL SISTEMA	104
5.5.	ENTORNO DE TRABAJO	105
5.6.	ANÁLISIS ECONÓMICO	106
5.6.1.	Costo de Materiales	106
5.6.2.	Presupuesto General	107
	CONCLUSIONES	108

RECOMENDACIONES-----	109 -
RESUMEN-----	110 -
ABSTRACT-----	111 -
GLOSARIO-----	112 -
ANEXOS-----	117 -

ÍNDICE DE FIGURAS

Figura II. 1 Ejemplos de robots con diferentes efectores para la locomoción terrestre.	- 25 -
Figura II. 2 Ejemplos de robots Ápodos.	- 31 -
Figura II. 3 Proceso de Adquisición de Datos.....	- 33 -
Figura II. 4 Ejemplo de Sensor Reflexivo.....	- 35 -
Figura II. 5 Funcionamiento del Sensor.....	- 35 -
Figura II. 6 Dimensiones del Sensor.....	- 36 -
Figura II. 7 Sensor Sharp.	- 37 -
Figura II. 8 Esquema interno y dimensiones del Sensor.....	- 37 -
Figura II. 9 Incidencia del haz de Luz.....	- 39 -
Figura II. 10 Sharp GP2D12 y su conector de tres pines.....	- 40 -
Figura II. 11 Detalle del conector del Sensor.....	- 41 -
Figura II. 12 Triangularización.	- 42 -
Figura II. 13 Reflexión del Haz de Luz.....	- 43 -
Figura II. 14 Dimensiones y Encapsulado.	- 43 -
Figura II. 15 Bloques Internos.	- 46 -
Figura II. 16 Cronograma.....	- 46 -

Figura II. 17 Características de Distancias.....	- 47 -
Figura II. 18 Tensión de Salida Analógica en función de Superficie Iluminada del Objeto Reflexivo.....	- 47 -
Figura II. 19 Tensión de Salida Analógica en función de la Distancia al objeto reflexivo.....	- 48 -
Figura II. 20 Tensión de Salida Analógica en función de la Temperatura Ambiente.	- 48 -
Figura II. 21 Tensión de Salida Analógica en función de la distancia de detección.	- 49 -
Figura II. 22. Reflexión de las ondas en la ionosfera.....	- 49 -
Figura II. 23. Sistema de comunicación.....	- 52 -
Figura II. 24. Servomotor.....	- 59 -
Figura II. 25. Ancho de Pulso	- 60 -
Figura II. 26. Conexión del Servo.....	- 61 -
Figura III. 27. Microcontrolador y Periféricos.....	- 63 -
Figura III. 28. Arquitectura Von Neumann.....	- 65 -
Figura III. 29. Arquitectura Harvard	- 65 -
Figura III. 30. Diagrama de conexiones de los PIC12Cxxx de la gama enana.	- 66 -
Figura III. 31. Diagrama de conexiones de los PIC16C54/56 de la gama baja.....	- 67 -
Figura III. 32. Diagrama de conexiones del PIC16C74, de la gama media	- 67 -
Figura III. 33. Encapsulado del PIC 16f628	- 73 -
Figura III. 34. Diagrama de pines 16F628	- 74 -

Figura III. 35. Pines del PIC 16F877A	- 81 -
Figura III. 36. Arquitectura Interna del Pic 16F877A.....	- 81 -
Figura III. 37. Registro PORTA (05H).....	- 82 -
Figura III. 38. Registro TRISA (85H).....	- 83 -
Figura III. 39. Conversión Analógico/Digital del PIC 16F877A.....	- 86 -
Figura III. 40. Registros de Control del convertidor A/D del PIC 16F877A.....	- 87 -
Figura III. 41. Registros de Resultados del Convertidor A/D.....	- 87 -
Figura IV. 42. Diagrama General del Sistema	- 89 -
Figura IV. 43. Diagrama de flujo para la transmisión de información	- 91 -
Figura IV. 44. Diagrama esquemático del transmisor.....	- 91 -
Figura IV. 45. Diseño del transmisor.....	- 92 -
Figura IV. 46. Transmisor.....	- 93 -
Figura IV. 47. Pantalla Principal.....	- 93 -
Figura IV. 48. Ingreso del Puerto.....	- 94 -
Figura IV. 49. Puerto Habilitado.....	- 94 -
Figura IV. 50. Secuencias de Desplazamiento.....	- 95 -
Figura IV. 51. Esquema de Conexión de la Placa de Comunicación Serial.....	- 96 -
Figura IV. 52. Diagrama de flujo para la recepción de información.....	- 97 -
Figura IV. 53. Diagrama esquemático del receptor	- 98 -

Figura IV. 54. Diseño del Receptor	- 99 -
Figura IV. 55. Diagrama de la etapa de ejecución	- 99 -
Figura IV. 56. Estructura Lateral	- 100 -
Figura IV. 57. Estructura del Eje Falso.....	- 100 -
Figura IV. 58. Estructura de cada Articulación.....	- 100 -
Figura IV. 59. Dimensiones de la Pieza Principal.....	- 101 -
Figura IV. 60. Estructura del Robot.....	- 101 -
Figura V. 61. Servomotor HITEC HS-311	- 103 -
Figura V. 62. Sensor SHARP GP2D120X.....	- 104 -
Figura V. 63. Fuentes de alimentación.....	- 105 -
Figura V. 64. Detección de obstáculos.....	- 105 -

ÍNDICE DE TABLAS

Tabla II. I. Calificaciones Máximas Absolutas.-----	36 -
Tabla II. II. Calificaciones Máximas Absolutas-----	37 -
Tabla II. III. Modelos de las Familias Sharp.-----	38 -
Tabla II. IV. Valores Máximos Absolutos.-----	44 -
Tabla II. V. Condiciones de Trabajo Recomendadas.-----	44 -
Tabla II. VI. Características Electro- Ópticas.-----	45 -
Tabla II. VII. Características de las capas de la tierra-----	51 -
Tabla II. VIII. Características de las capas de la tierra-----	59 -
Tabla II. IX. Asignación cuantitativa para las métricas-----	71 -
Tabla II. X. Calificaciones y costos-----	72 -
Tabla II. XI. Matriz de puntos-----	73 -
Tabla II. XII. Modulo del Temporizador del PIC 16F877A.-----	88 -
Tabla II. XIII.. Materiales empleados en el robot y sus costos-----	106 -
Tabla II. XIV. Tabla Presupuesto General-----	107 -

INTRODUCCIÓN

Desde la invención del circuito integrado, el desarrollo constante de la electrónica digital ha dado lugar a dispositivos cada vez más complejos. Entre ellos los microprocesadores y los microcontroladores. Estos se utilizan en circuitos electrónicos comerciales desde hace unos años de forma masiva, debido a que permiten reducir el tamaño y el precio de los equipos. En los últimos años se ha facilitado enormemente el trabajo con los microcontroladores al bajar los precios, aumentar las prestaciones y simplificar los montajes, de manera que en muchas ocasiones vale la pena utilizarlos en aplicaciones donde antes se utilizaba lógica discreta.

El ser humano lleva siglos soñando con las creaciones de robots autónomas y obedientes, capaces de llevar a cabo los trabajos más duros. A finales del siglo XX ese sueño comenzó a ser real. Los robots ya han demostrado ser excelentes sustitutos de los humanos para llevar a cabo tareas repetitivas que no requieran capacidad de improvisación. Ya hay numerosos robots trabajando en minas y excavaciones petrolíferas, fabricando bienes de consumo en cadenas de montaje, explorando el espacio y combatiendo en guerras. Por otro lado hay muchos trabajos que las personas no les gusta hacer, sea ya por ser aburrido o bien peligroso, siempre se va a tratar de evitar para no hacerlo.

Mediante un microcontrolador PIC, podemos realizar diferentes máquinas electrónicas que incluso puedan sustituir el trabajo humano. Además de ello también podemos hablar de robots que puedan ayudar en áreas poco accesibles por el hombre, entre ellos tenemos un robot ávido tipo gusano donde el principal objetivo es sobrepasar los obstáculos que se encuentren enfrente de su trayectoria de movimiento. Para ello, el microcontrolador que controla el robot, utilizará un sensor de donde obtendrá lecturas, de su trayectoria de movimiento, La solución a la problemática planteada se puede dividir en “hardware” y en “software”.

El presente trabajo de investigación se encuentra dividido en cinco capítulos para el análisis del DISEÑO Y CONSTRUCCIÓN DE UN ROBOT ÁPODO TIPO GUSANO, los que se detallan a continuación:

Capítulo I.- En este capítulo se especifica los antecedentes, la debida justificación y los objetivos de la realización de esta tesis.

Capítulo II y III.- En estos capítulos teóricamente se establecen los tipos de sensores, actuadores, microcontroladores, baterías y demás conceptos de los elementos que se utilizan posteriormente.

Capítulo IV.- Este capítulo contiene los pasos a considerar para el diseño y construcción del robot.

Capítulo V.- En este capítulo se detallará el Análisis respectivo así como los Resultados obtenidos en la implementación de la tesis.

Esta estructura permitirá arribar a conclusiones y proponer recomendaciones fundamentadas en el proceso de investigación realizado.

CAPÍTULO I

MARCO REFERENCIAL

1. ANTECEDENTES

En los últimos años, los robots ápodos han cobrado interés en el área de robótica móvil. Esto se debe a que este tipo de robots, que se valen únicamente de su cuerpo para moverse, tienen un sistema de locomoción robusto para contender con ambientes con superficies irregulares.

Los robots ápodos están frecuentemente inspirados en la naturaleza, sus patrones se basan en los movimientos de los gusanos y serpientes cuya estructura se compone de múltiples segmentos que permiten el movimiento ondulatorio.

El presente proyecto apunta a implementar un robot gusano (diseño, construcción y desarrollo de hardware y software), el cual, como su nombre lo dice, simulará el movimiento de este invertebrado basado en la utilización de servomotores y diversas secuencias de control. El robot se desplazará con movimiento ondulatorio por trayectos rectos sin la utilización de ruedas y/o

patas. El mismo podrá ser utilizado en espacios y ambientes difícilmente accesibles para el ser humano, ya sea por su peligrosidad o por su tamaño.

Las secuencias de desplazamiento serán controladas por medio de microcontroladores.

2. JUSTIFICACIÓN

Un robot ápodo se destaca por la ausencia de extremidades obligándole a realizar movimientos ondulatorios con su cuerpo. Este proyecto diseñará e implementará un robot ápodo tipo gusano cuyas articulaciones estarán formadas por una concatenación de servomotores controlados inalámbricamente, utilizando un módulo de radiofrecuencia. Para lo cual se utilizará como módulo principal un microcontrolador que manipulara las distintas articulaciones.

Además se utilizará un sensor infrarrojo con el fin de detectar obstáculos.

Esta es una alternativa para mejorar el desplazamiento de los robots en caminos con desniveles. Al comparar el desempeño del robot gusano con otros esquemas de desplazamiento, el robot ápodo brinda una mayor flexibilidad al momento de sobrepasar los obstáculos mas no los evade.

Podría ser utilizado en la participación de eventos relacionados a la robótica, etc.

3. OBJETIVOS

1.3.1. OBJETIVO GENERAL

- ✓ Diseñar e implementar un robot ápodo (tipo gusano) controlado inalámbricamente

1.3.2. OBJETIVOS ESPECIFICOS

- ✓ Desarrollar un sistema de comunicación inalámbrico confiable que permita un adecuado control del robot.
- ✓ Estudiar las posibles secuencias de control de movimiento del robot.
- ✓ Implementar algoritmos y programas de control para las secuencias de desplazamiento.
- ✓ Implementar la estructura del robot basándonos en el diseño elaborado.
- ✓ Realizar pruebas y puesta a punto del robot.

4. HIPÓTESIS

Con el diseño y construcción de un robot ápedo se pretende lanzar una alternativa diferente en cuanto a desplazamiento se refiere. Es decir que proporcione mayor flexibilidad al momento de detectar los obstáculos sobrepasándolos y no evadiéndolos.

CAPÍTULO II

FUNDAMENTO TEÓRICO

2.1. LOCOMOCIÓN

2.1.1. Introducción

La locomoción es la capacidad que tienen los seres vivos pertenecientes al reino animal que les permite trasladarse voluntariamente de un lugar a otro. Es una de las características diferenciadoras de los animales con respecto a las plantas. Hay dos aspectos importantes a tener en cuenta: el **control** y la **voluntariedad**. Para que el movimiento sea considerado locomoción el individuo tiene que querer realizarlo y además controlarlo. Así, los nenúfares que reposan sobre la superficie del agua se mueven por las corrientes o la acción de otros animales, pero no se considerará locomoción ya que no son voluntarios ni controlados.



Figura II. 1 Ejemplos de robots con diferentes efectores para la locomoción terrestre.

El estudio de la locomoción se divide en dos niveles:

El nivel inferior es el encargado del control y la coordinación de los músculos (o actuadores en el caso de los robots) para que el individuo pueda desplazarse. Engloba también los diferentes tipos de maneras de desplazarse que se pueden lograr (giros, desplazamiento en línea recta, desplazamiento lateral, etc.). Las preguntas que se resuelven en este nivel son: ¿Cómo me desplazo? ¿Cómo coordino todos los músculos (actuadores) para lograr la locomoción?

El nivel superior se encarga de la planificación de trayectorias, navegación y otras tareas de mayor nivel. Está relacionado con la voluntad. Las preguntas que definen este nivel son: ¿Dónde quiero ir? ¿Qué camino seguir?

En esta tesis nos centraremos en el nivel inferior de la locomoción, estudiando los mecanismos para que los robots ápodos puedan desplazarse.

2.1.1.1. Tipos de locomoción

En la naturaleza, la locomoción de los animales se ha adaptado al medio en el que viven. Se puede realizar una primera clasificación según el medio en el que se desplazan. Así, la locomoción puede ser: aérea, acuática o terrestre. Esta clasificación no es excluyente. Los mamíferos terrestres también son capaces de desplazarse cortas distancias por el agua, por ejemplo para cruzar un río. En este caso utilizan un modo de caminar diferente, que les permite nadar.

La locomoción terrestre se puede dividir a su vez en dos categorías, según los órganos empleados para realizar el desplazamiento: locomoción mediante patas (mamíferos, insectos) o mediante movimientos corporales (serpientes, orugas, lombrices).

2.1.1.2. Locomoción de robots

Una de las áreas de investigación en robótica es la locomoción: dotar a los robots de capacidades locomotivas para que puedan trasladarse de un lugar a otro. Estos robots reciben el nombre genérico de robots móviles. A su vez, el estudio de la locomoción se realiza en los dos niveles mencionados en los apartados previos. En las investigaciones del nivel superior se parte de que el robot puede desplazarse, haciéndose abstracción de los mecanismos para que sea posible (patas, ruedas...) y se centra en las tareas de alto nivel de navegación, planificación de trayectorias, visión, cooperación, etc.

Al igual que sucede con los animales, en el estudio del nivel inferior de la locomoción se pueden clasificar los robots según los efectores empleados para su desplazamiento: ruedas, orugas, patas o el cuerpo. Bajo el término robots ápodos quedan englobados aquellos robots,

que igual que sus homólogos en la naturaleza, realizan la locomoción mediante movimientos corporales. Estos son las cuatro categorías clásicas para el estudio de la locomoción, sin embargo la clasificación no es cerrada. En la figura 2.1 se muestran fotos de robots que usan diferentes efectores para realizar la locomoción: Ruedas, orugas, Whegs, el cuerpo y cuatro, seis y ocho patas.

Los temas de investigación en el nivel inferior de la locomoción son las propiedades de los diferentes efectores, cómo realizar la coordinación de los actuadores, los diferentes modos de caminar, algoritmos de control, etc.

2.1.1.3. Diseño de robots móviles

Lo mismo que ocurre en el reino animal, donde la locomoción de los individuos está especialmente adaptada al entorno en el que normalmente se desenvuelven, para el diseño de un robot móvil es esencial conocer el terreno por el que se va a desplazar. Es el entorno el elemento clave en la decisión de qué efectores elegir y qué tipos de modo de caminar implementar. Así por ejemplo, si el robot se va a desplazar por superficies lisas donde no existe la necesidad de superar obstáculos, bastará con utilizar ruedas o a lo sumo orugas.

El proceso de diseño se podría resumir en los siguientes pasos:

1. Estudiar el entorno en el que se va a desplazar el robot
2. Seleccionar los efectores
3. Implementar los modos de caminar

Estos pasos son muy críticos. Una mala elección a este nivel, implicaría tener que volver a reconstruir el robot. Esta es la razón por la que las investigaciones a este nivel son tan importantes: cuanto mejor se conozcan las propiedades de los efectores, los modos de caminar posibles, la eficiencia de cada modo de caminar, etc., mayor serán los datos disponibles para tomar una decisión de diseño correcta.

Leger abordó en su tesis doctoral el problema del diseño automático de robots, utilizando un enfoque evolutivo. Su idea central es que el espacio de búsquedas de soluciones al problema de la locomoción es tan amplio que son necesarias herramientas que permitan explorar la mayor cantidad posible de ellas antes de tomar una decisión de qué diseño implementar. Un error en la configuración del robot a este nivel es muy crítico. Por ello propuso utilizar algoritmos evolutivos para ayudar a los diseñadores en esta etapa.

Sin embargo, existen aplicaciones en las que es difícil conocer a priori y con detalle el terreno, por lo que hay mucha incertidumbre en la etapa inicial de diseño. Tal es el caso del diseño de robots para las operaciones de búsqueda y rescate o la exploración de planetas. Debido a ello, el robot tiene que tener la máxima versatilidad posible. Las investigaciones se centran en estudiar los efectores más versátiles y los diferentes modos de caminar posibles.

2.1.1.4. El problema de la locomoción

Uno de los grandes retos es el de desarrollar un robot que sea capaz de moverse por cualquier tipo de entorno por muy escarpado y complicado que sea. Es decir, un robot con una gran versatilidad.

Esto tiene especial interés en las aplicaciones en las que el entorno no es lo suficientemente conocido o es cambiante, como en la exploración de las superficies de otros planetas, exploración en entornos hostiles o operaciones de búsqueda y rescate. ¿Qué es mejor utilizar, patas, rudas, orugas...? ¿Qué número de patas? ¿Qué tipo de movimiento? ¿Qué configuración de las patas si es que las tiene?

La Nasa está especialmente interesada en este problema, financiando proyectos destinados a la construcción y evaluación de alternativas para que los robots se puedan desplazar en entornos escarpados.

Dos de estos proyectos en la etapa inicial (finales de los ochenta) fueron el CMU Ambler y el Dante II. Son dos ejemplos que ilustran el modelo de diseño descrito en el apartado anterior: diseños de estructuras específicas a partir de las especificaciones del entorno.

El Ambler es un robot autónomo para la exploración de planeta, pensado para moverse por la superficie de Marte. A partir de las especificaciones se diseñó el robot, dotado de 6 patas, 3.5m de altura y un peso de 2.500Kg. El tipo de locomoción seleccionado fue mediante patas, es el que presenta mayor eficiencia teórica. Sin embargo, este robot nunca fue enviado a Marte. Para cumplir con los requisitos, las dimensiones y peso del robot son excesivamente grandes. A parte de que el consumo de potencia es muy alto.

El Robot Dante II se diseñó también para explorar terrenos escarpados y se probó en 1994 para la exploración del volcán del monte Spurr en Alaska. En este caso, el robot poseía 8 patas con un sistema de locomoción denominado framewalker. A pesar de que se conocían las especificaciones del terreno, que el robot poseía un cable que lo mantenía enganchado a la cima y por el que iba descendiendo, al quinto día volcó y no pudo ser recuperado.

Para la exploración de Marte, la Nasa se decantó por la utilización de ruedas ya que hasta el momento han dado muy buenos resultados. Sin embargo, las rudas son muy limitadas. Sólo permiten que el robot se pueda desplazar por entornos muy controlados. Es una de las razones por las que es necesario planear con mucha antelación y cuidado el lugar al que se enviarán los robots, no sólo teniendo en cuenta el interés en la recolección de datos científicos, las condiciones atmosféricas... sino también que los robots puedan desplazarse correctamente por el terreno. Esto es una gran limitación.

2.1.1.5. Robots ápodos

Frente a la locomoción terrestre mediante patas, están los seres vivos que utilizan movimientos corporales. Los robots que utilizan este tipo de desplazamiento los denominaremos robots ápodos. El término ápodo significa “carente de patas”.

Estos robots tienen unas características que los hacen únicos, al igual que sus homólogos las serpientes y gusanos. Por un lado está su capacidad para cambiar su forma. Frente a las rígidas estructuras del resto de robots, los ápodos pueden doblarse y adoptar la forma del terreno por el que se desplazan.

Por otro lado, su sección es muy pequeña en comparación con su tamaño, lo que les permite introducirse por tubos u orificios pequeños y llegar a lugares donde otros robots no pueden.



Figura II. 2 Ejemplos de robots Ápodos.

2.2. SENSORES

2.2.1. Introducción

Los sensores de proximidad tienen mucha relevancia en el mundo de la industria debido a la preocupación en lo referente a la seguridad, tanto de personas como de la maquinaria.

A continuación se realizara la descripción del funcionamiento del sensor usado en la medición de proximidad en este caso el GP2D120.

2.2.2. Criterios para la Selección de Sensores

Sensibilidad

- Se define como el cociente entre la tasa de cambio de valores de salida para el cambio los valores de entrada.

Linealidad

- Es la medida de la constancia de la tasa de salida con respecto a la entrada.

Rango

- Es la medida de la diferencia entre el mínimo y el máximo valor medido.

Tiempo de Respuesta

- Puede ser un tiempo fijo o depender de cuánto varíe la magnitud a medir. Depende de la capacidad del sistema para seguir las variaciones de la magnitud de entrada.

Precisión

- Es el error de medida máximo esperado.

Repetitividad

- Error esperado al repetir varias veces la misma medida.

Resolución

- Mínima variación de la magnitud de entrada que puede apreciarse a la salida.

2.2.2.1. Sistemas Actuadores

Se les llama así a todo circuito, interfaz o motor, que ejecute una determinada acción o tarea en función de las entradas que nos otorguen los sensores.

Por ejemplo el robot tiene sistemas que le permitan moverse, según la información que se ha recibido del medio externo por medio de los sensores, permitiendo que el mismo llegue a cumplir la misión para el cuál fue diseñado desde un principio. Todo lo que le permita al robot reaccionar ante el medio externo cualquiera que sea este y cualquiera que sea el propósito para el que fue diseñado, se le conoce como actuador.

La salida de un actuador puede ser de varias formas:

- Movimiento mecánico
- Tensión
- Corriente
- Presión
- Nivel hidráulico
- Intensidad luminosa, etc.

2.2.2.2. Sistemas de Interfaz entre Sensores y Actuadores

Para que los sensores y actuadores cumplan su función e interactúen entre sí se necesita de un sistema, que informe a los actuadores que es lo que deben hacer en función de la información que ha sido recibida por los sensores, esta interfaz es diseñada para trabajar con datos de ingreso que nos otorgan los sensores y salidas que manejan los diferentes actuadores.

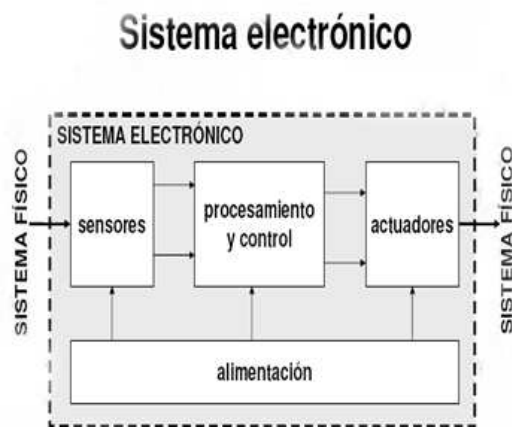


Figura II. 3 Proceso de Adquisición de Datos.

Esta interfaz que es un sistema de control de entradas y salidas, se han realizado para mayor facilidad y mejor control en los llamados microcontroladores, que permiten la ejecución de

múltiples tareas y procesos en un simple chip, lo cual reduce considerablemente los componentes electrónicos y el tamaño del diseño que antes se utilizaba.

2.2.3. Sensores Sharp.

Nos centraremos en los sensores infrarrojos Sharp y después pasaremos a explicar la familia GP2Dxx que es una de las familias más utilizadas tanto en lo que viene a denominarse robótica móvil casera como en el ámbito de la investigación debido principalmente a su facilidad de integración y su bajo coste.

Es importante destacar que hemos decidido explicar algunos sensores diferentes de la familia GP2Dxx para observar que no es la única familia utilizada por Sharp.

Finalmente hemos centrado nuestra atención en el sensor GP2D12.

2.2.4. Sensores Sharp en distintas familias.

2.2.4.1. Reflexivo IS471F

Sensor basado en el dispositivo SHARP IS471F inmune a interferencias de luz normal. Este sensor incorpora un modulador/demodulador integrado en su carcasa ya a través de su patilla 4 controla un diodo LED de infrarrojos externo, modulando la señal que este emitirá, para ser captada por el IS471F que contiene el receptor. Cuando un objeto se sitúa enfrente del conjunto emisor/receptor parte de la luz emitida es reflejada y re modulada para activar la salida en la patilla 2 que pasará a nivel bajo si la señal captada es suficientemente fuerte.

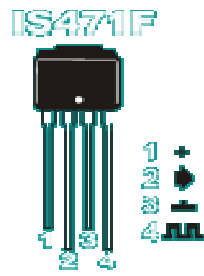


Figura II. 4 Ejemplo de Sensor Reflexivo.

El uso de la luz infrarroja modulada tiene por objeto hacer al sensor relativamente inmune a las interferencias causadas por la luz normal de una bombilla o la luz del sol.

El funcionamiento de este sensor se observa en el siguiente esquema. El sensor se alimenta por las patillas 1 y 3 que corresponden a Vcc y Gnd respectivamente. La patilla 2 es la salida del detector y la patita 4 es la salida que modula al diodo LED externo. Mediante el potenciómetro P1 se varía la distancia a la que es detectado el objeto. Cuando más baja sea la resistencia de este potenciómetro, más intensa será la luz emitida por el diodo de infrarrojo y por lo tanto mayor la distancia a la que puede detectar el objeto.

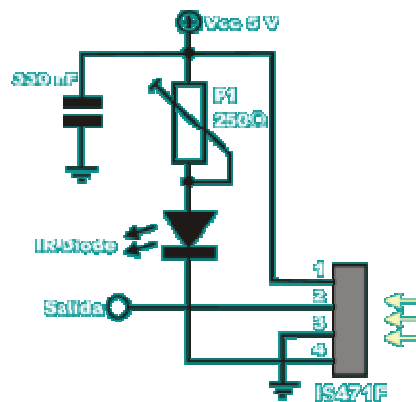


Figura II. 5 Funcionamiento del Sensor.

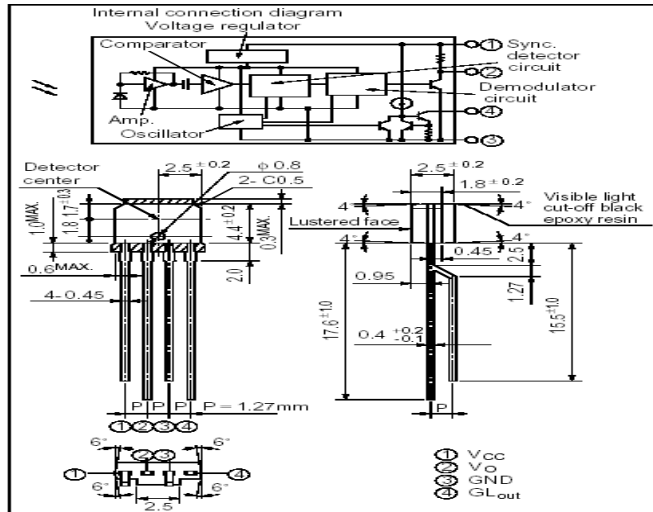


Figura II. 6 Dimensiones del Sensor.

Parameter		Symbol	Rating	Unit
Supply voltage		V_{CC}	-0.5 to 16	V
Output	Output voltage	V_O	16	V
	Output current	I_o	50	mA
*1 GL output	Output voltage	V_{GL}	16	V
Power dissipation		P	250	mW
Operating temperature		T_{opr}	-25 to +60	°C
Storage temperature		T_{stg}	-40 to +100	°C
*2 Soldering temperature		T_{sol}	260	°C

Tabla II. I. Calificaciones Máximas Absolutas.

2.2.4.2. Sharp GP2Y0A02YK

Es un sensor de infrarrojos que proporciona una lectura continua de la distancia medida como una tensión analógica dentro de un rango de 20 a 150 cm. La tensión de alimentación es de 5V y la tensión de salida varía unos 2 voltios de diferencia entre el margen mínimo y el máximo de la distancia medida. Tiene un encapsulado similar a otros sensores Sharp, pero presenta una mayor distancia entre la lente y el sensor con el fin de aumentar el rango de trabajo. La conexión se realiza mediante un conector JST de 3 vías, 2 para alimentación y una para la salida. La salida

está disponible de forma continua y su valor es actualizado cada 39 ms. Normalmente se conecta esta salida a la entrada de un convertidor analógico digital el cual convierte la distancia en un número que puede ser usado por el microprocesador.



Figura II. 7 Sensor Sharp.

Las aplicaciones son la detección de cuerpos humanos y aplicaciones en aparatos caseros.

Aquí presentamos el esquema interno de bloque las dimensiones del sensor y las calificaciones máximas absolutas.

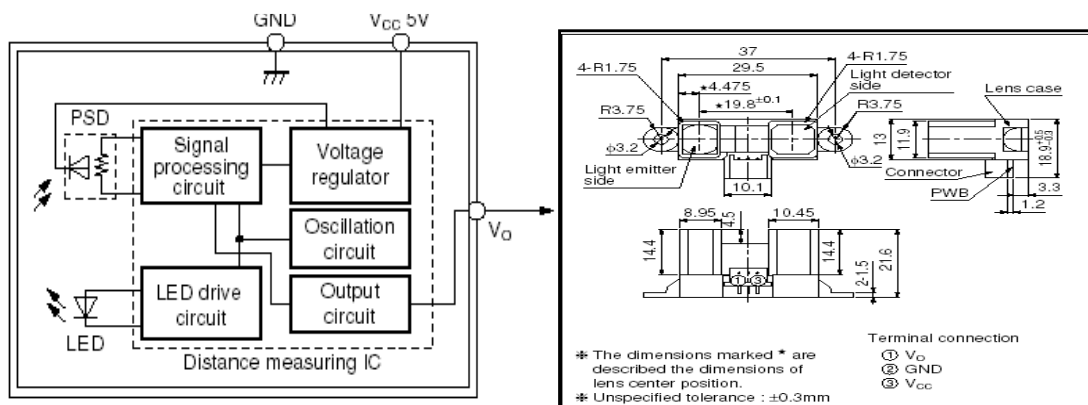


Figura II. 8 Esquema interno y dimensiones del Sensor.

Parameter	Symbol	Rating	Unit
Supply voltage	V_{CC}	-0.3 to +7	V
^{*1} Output terminal voltage	V_O	-0.3 to $V_{CC} + 0.3$	V
Operating temperature	T_{opr}	-10 to +60	°C
Storage temperature	T_{stg}	-40 to +70	°C

Tabla II. II. Calificaciones Máximas Absolutas

2.2.4.3. La Familia GP2DXX

Los sensores infrarrojos Sharp GP2DXX son una familia de sensores infrarrojos para la detección y medida de distancia a los objetos. En la página Web de Sharp encontramos los modelos que se muestran en esta tabla refiriéndonos a ésta familia.

Modelo	Características	Rangos absolutos Máximos		Característica Electro-Ópticas				
		VCC (V)	Topr (°C)	Rango de distancia de medida	V _{OH} (V) MIN VCC	V _{OL} (V) MAX	Operativo (mA) MAX	Standby (µA) MAX
GP2D02	Sensor que mide distancias con PSD (Detector Sensible a la Posición), LED infrarrojo y circuito de procesado de señal, 8 bits.	-0.3 a +10	-10 a +60	100 a 800	V _{CC} -0.3	0.3	17	8
GP2D021	Sensor que mide distancias con PSD (Detector Sensible a la Posición), LED infrarrojo y circuito de procesado de señal, 8 bits.	-0.3 a +10	-10 a +60	40 a 300	V _{CC} -0.3	0.3	35	8
GP2D05	Sensor que mide distancias con PSD (Detector Sensible a la Posición), LED infrarrojo y circuito de procesado de señal, 1 bits.	-0.3 a +10	-10 a +60	100 a 800	V _{CC} -0.3	0.3	22	8
GP2D12	Sensor que mide distancias con PSD (Detector Sensible a la Posición), LED infrarrojo y circuito de procesado de señal, valor analógico entre 0-3V dependiendo de la distancia	-0.3 a 7	-10 a +60	100 a 800	V _O (TYP) = 0.4V a 30 cm		MAX.50	
GP2D120	Sensor que mide distancias con PSD (Detector Sensible a la Posición), LED infrarrojo y circuito de procesado de señal, valor analógico entre 0-3V dependiendo de la distancia	-0.3 a 7	-10 a +60	40 a 300	V _O (TYP) = 0.4V a 30 cm		MAX.50	
GP2D150A	Sensor que mide distancias con PSD (Detector Sensible a la Posición), LED infrarrojo y circuito de procesado de señal, valor digital (0 o 1)	-0.3 a 7	-10 a +60	30 a 300 (Detección distancia typ. 15cm)	V _{CC} -0.3	0.6	MAX.50	
GP2D150T	Sensor que mide distancias con PSD (Detector Sensible a la Posición), LED infrarrojo y circuito de procesado de señal, valor digital (0 o 1)	-0.3 a 7	-10 a +60	30 a 300 (Detección distancia typ. 22cm)	V _{CC} -0.3	0.6	MAX.50	
GP2Y0D02YK	Sensor que mide largas distancias con PSD (Detector Sensible a la Posición), LED infrarrojo y circuito de procesado de señal, valor digital (a 80 cm)	-0.3 a 7	-10 a +60	200 a 1500	V _{CC} -0.3	0.6	MAX.50	

Tabla II. III. Modelos de las Familias Sharp.

Como ya hemos dicho los sensores GP2DXX de Sharp son dispositivos de reflexión por infrarrojos con medidor de distancia proporcional al ángulo de recepción del haz de luz que incide en un sensor lineal integrado, dependiendo del modelo utilizado, la salida puede ser analógica digital o booleana.

El funcionamiento general de esta familia es el siguiente:

El dispositivo emite luz infrarroja por medio de un LED emisor de infrarrojos, esta luz pasa a través de una lente que concentra los rayos de luz formando un único rayo lo más concentrado posible para mejorar la directividad del sensor, la luz va recta hacia delante y cuando encuentra un obstáculo reflectante rebota y retorna con cierto ángulo de inclinación dependiendo de la distancia. La luz que retorna es concentrada por otra lente es concentrada por otra lente y así todos los rayos de luz inciden en un único punto del sensor de luz infrarroja que contiene en la parte receptora del dispositivo. Estos sensores constan de un CCD lineal que, dependiendo del ángulo de recepción de la luz, incidirá ésta en un punto u otro del sensor pudiendo de esta manera obtener un valor lineal y proporcional al ángulo de recepción del haz de luz.

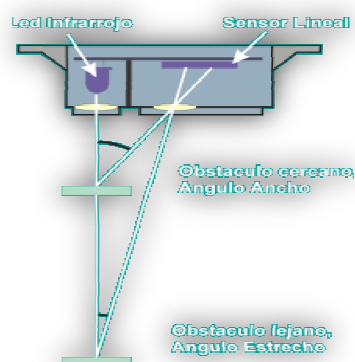


Figura II. 9 Incidencia del haz de Luz.

Dependiendo del modelo elegido leeremos de una manera u otra la salida de éste con lo cual tendríamos que analizar el datasheet de cada modelo para ver su funcionamiento interno. Esto puede ser consultado en la página de Sharp o en webs especializadas.

En los modelos analógicos la salida es un voltaje proporcional a la distancia medida.

En los modelos digitales la lectura será de 8 bits serie con reloj externo.

En los modelos booleanos la salida será de 1 bit, éste marca el paso por la zona de histéresis del sensor con lo cual solo tendremos medición de una distancia fija.

2.2.4.4. El sensor GP2D12.

El Sharp GP2D12 es un sensor medidor de distancias por infrarrojos que indica mediante una salida analógica la distancia medida. La tensión de salida varía de forma no lineal cuando se detecta un objeto en una distancia entre 10 y 80 cm. La salida está disponible de forma continua y su valor es actualizado cada 32 ms. Normalmente se conecta esta salida a la entrada de un convertidor analógico digital el cual convierte la distancia en un número que puede ser usado por el microprocesador. La salida también puede usada directamente en un circuito analógico. Hay que tener en cuenta que la salida no es lineal. El sensor utiliza solo una línea de salida para comunicarse con el procesador principal.



Figura II. 10 Sharp GP2D12 y su conector de tres pines.

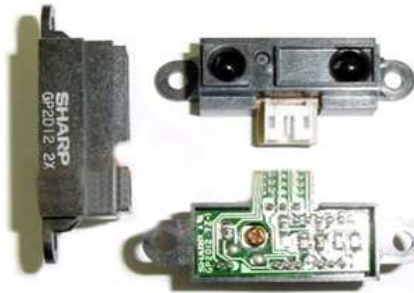


Figura II. 11 Detalle del conector del Sensor.

Este dispositivo emplea el método de triangulación utilizando un pequeño Sensor Detector de Posición (PSD) lineal para determinar la distancia o la presencia de los objetos dentro de su campo de visión.

Su modo de funcionamiento consiste en la emisión de un pulso de luz infrarroja, que se transmite a través de su campo de visión que se refleja contra un objeto. Si no encuentra ningún obstáculo, el haz de luz no refleja y en la lectura que se hace indica que no hay ningún obstáculo. En el caso de encontrar un obstáculo el haz de luz infrarroja se refleja y crea un triángulo formado por el emisor, el punto de reflexión (el obstáculo) y el detector.

La información de la distancia se extrae midiendo el ángulo recibido. Si el ángulo es grande, entonces el objeto está cerca (el triángulo es ancho). Si el ángulo es pequeño significa que está lejos (el triángulo es largo y, por tanto, delgado).

Observemos a continuación esto (este precisamente es el concepto de triangularización):



Figura II. 12 Triangularización.

Observemos ahora como se lleva a cabo la triangulación en el sensor. El LED infrarrojo emite el haz de luz a través de una pequeña lente convergente que hace que el haz emisor llegue de forma paralela al objeto. Cuando la luz choca con un obstáculo, una cierta cantidad de luz se refleja, si el obstáculo fuera un espejo perfecto, todos los rayos del haz de luz pasarían y sería imposible medir la distancia. Sin embargo, casi todas las sustancias tienen un grado bastante grande de rugosidad de la superficie que produce una dispersión hemisférica de la luz (reflexión no teórica). Alguno de estos haces de ésta luz rebota hacia el sensor que es recibida por la lente.

La lente receptora también es una lente convexa, pero ahora sirve para un propósito diferente, actúa para convertir el ángulo de posición. Si un objeto se pone en el plano focal de una lente convexa y los otros rayos de luz paralelos en otro lado, el rayo que pasa por el centro de la lente atraviesa inalterado o marca el lugar focal. Los rayos restantes también enfocan a este punto.

En el plano focal hay un Sensor Detecto de Posición (PSD). Éste dispositivo semiconductor entrega una salida cuya intensidad es proporcional a la posición respecto al centro (eficaz) de la

luz que incide en él. El rendimiento del PSD en la salida es proporcional a la posición del punto focal. Esta señal analógica tratada es la que se obtiene a la salida del sensor.

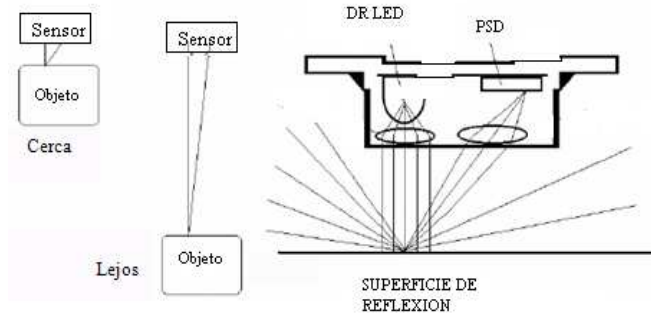


Figura II. 13 Reflexión del Haz de Luz.

Características:

- ✓ Menor influencia del color de los objetos reflexivos.
- ✓ Línea indicadora de distancia output/distance:
- ✓ Tipo de salida indicadora de la distancia analógica (tensión analógica). Distancia del detector de 10 a 80 cm.
- ✓ Bajo costo.

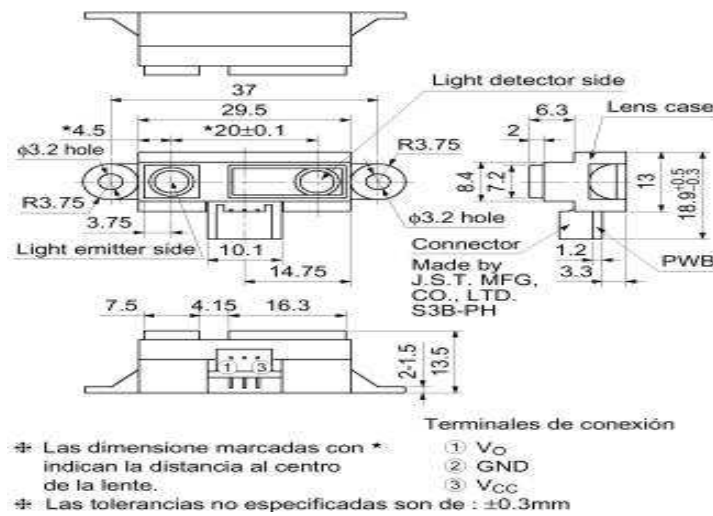


Figura II. 14 Dimensiones y Encapsulado.

Aplicaciones:

- ✓ En televisiones
- ✓ Computadoras Personales
- ✓ Automóviles
- ✓ Fotocopiadoras
- ✓ Sensores en sanitarios
- ✓ Sensores de cuerpo humano para los productos de consumo como ventiladores eléctricos y aires acondicionados.
- ✓ Sensores de garaje

Para $T_a = 25^\circ \text{C}$ y $V_{CC} = 5\text{V}$

Parámetro	Símbolo	Rangos	Unidades
Tensión de Alimentación	V_{CC}	-0.3 a 7	V
Tensión en el terminal de salida	V_O	-0.3 a $V_{CC}+0.3$	V
Temperatura de trabajo	T_{opr}	-10 a +60	$^\circ\text{C}$
Temperatura de almacenamiento	T_{stg}	-40 a +70	$^\circ\text{C}$

Tabla II. IV. Valores Máximos Absolutos.

Parámetro	Símbolo	Rangos	Unidades
Tensión de alimentación de trabajo	V_{cc}	4.5 a +5.5	V

Tabla II. V. Condiciones de Trabajo Recomendadas.

Parámetros		Símbolo	Condiciones	MIN.	TIP.	MAX.	Unidad
Rango de medida de distancia		ΔL	(*1) (*3)	10	-	80	cm
Tensión en el terminal de salida	GP2D12	V_O	L=80 cm (*1)	0.25	0.4	0.55	V
	GP2D15	V_{OH}	Tensión de salida a nivel alto (*1)	$V_{CC}-0.3$	--	--	V
		V_{OL}	Tensión de salida a nivel bajo (*1)	--	--	0.6	V
Incremento de la tensión de salida	GP2D12	ΔV_O	Cambio de salida de L=80 a 10 cm	1.75	2.0	2.25	V
Tensión de salida en función de la distancia	GP2D15	V_O	(*1)(*2)(*4)	21	24	27	cm
Corriente media de dispersión		I_{CC}	L = 80 cm (*1)	--	33	50	mA

Tabla II. VI. Características Electro- Ópticas.

✓ *1. Objeto reflexivo usando: Papel blanco (Para el color gris se usa la tarjeta R-27 de la Cía. de Kodak S.A. · la cara blanca, la proporción reflexiva; 90%).

- ✓ *2. Utilizamos el dispositivo después del ajuste siguiente: salida al cambiar la distancia L $24\text{cm} \pm 3\text{cm}$ debe medirse por el sensor.
- ✓ *3. Rango de distancia que mide el sistema del sensor óptico.
- ✓ *4. La salida de cambio tiene una anchura de la histéresis. La distancia especificada por V_o desde que la salida a nivel bajo (L) hasta que cambia a nivel alto (H).

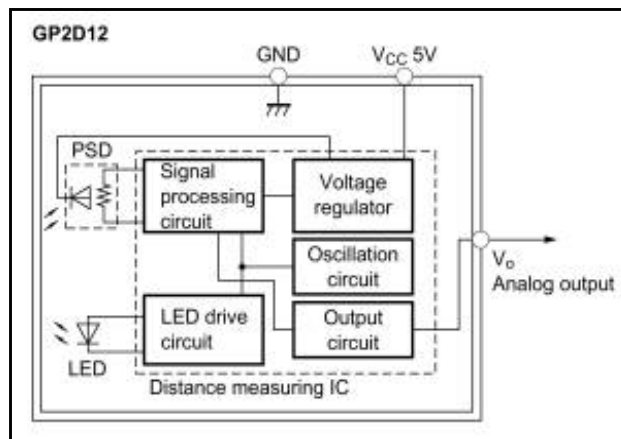


Figura II. 15 Bloques Internos.

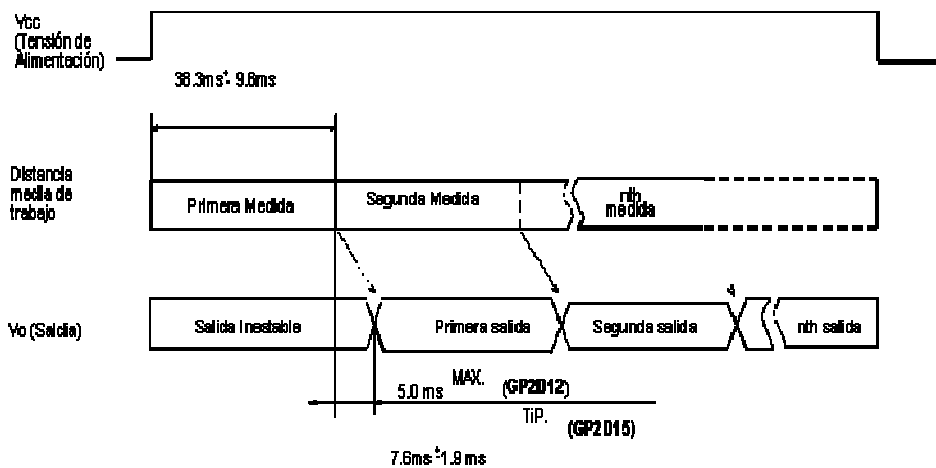


Figura II. 16 Cronograma.

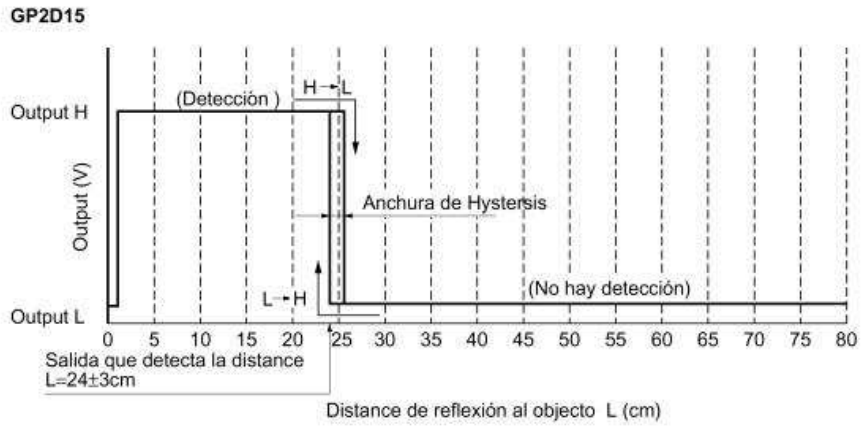


Figura II. 17 Características de Distancias.

2.2.4.4.1. Curvas Características del GP2D12

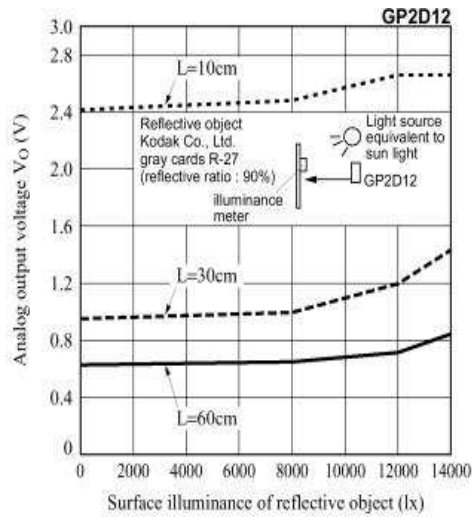


Figura II. 18 Tensión de Salida Analógica en función de Superficie Iluminada del Objeto Reflexivo

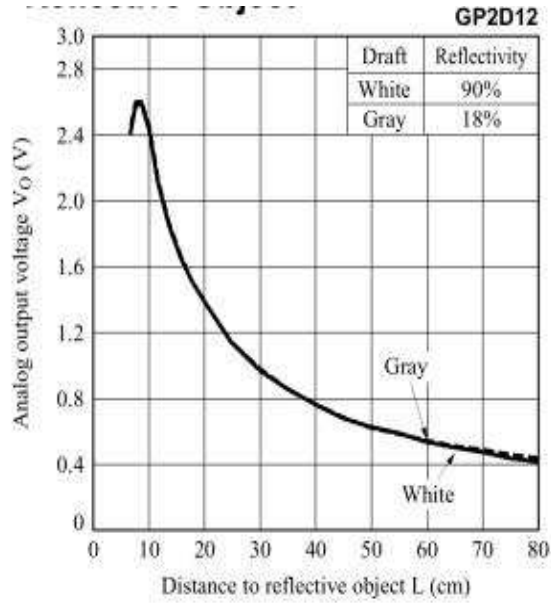


Figura II. 19 Tensión de Salida Analógica en función de la Distancia al objeto reflexivo

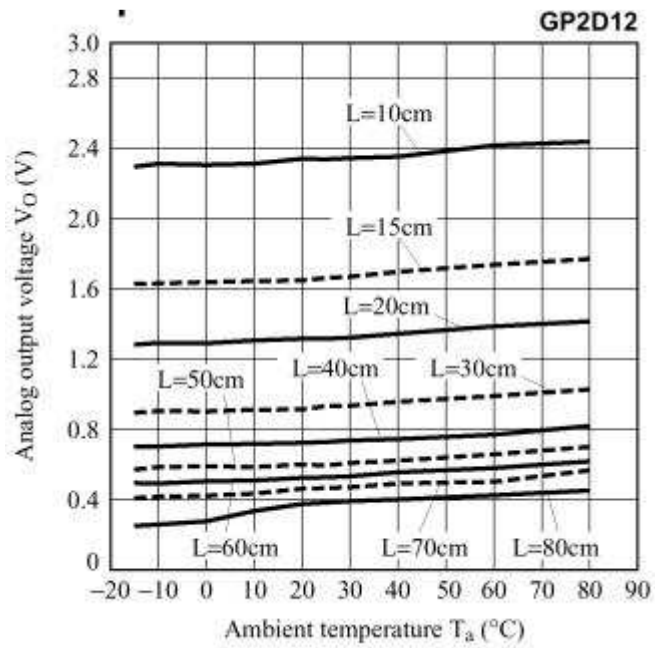


Figura II. 20 Tensión de Salida Analógica en función de la Temperatura Ambiente.

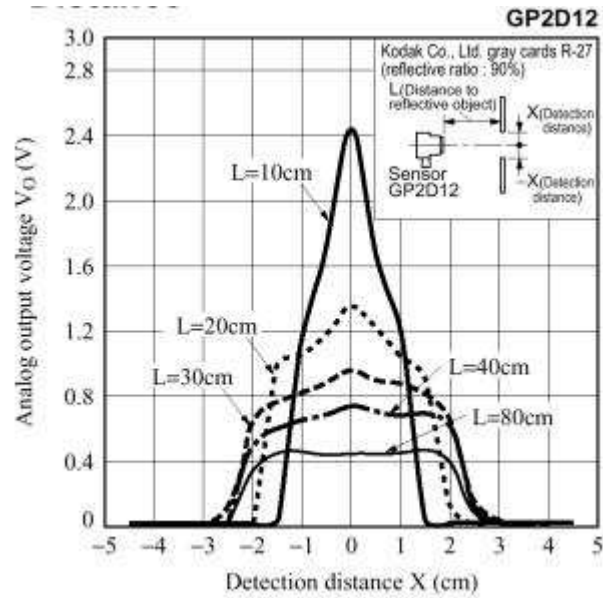


Figura II. 21 Tensión de Salida Analógica en función de la distancia de detección.

2.3. LOS SISTEMAS DE COMUNICACIÓN

El conjunto de elementos que intervienen en el proceso de intercambio de información forma un sistema de comunicación.

2.3.1. TRANSMISIÓN DE ONDAS DE RADIO

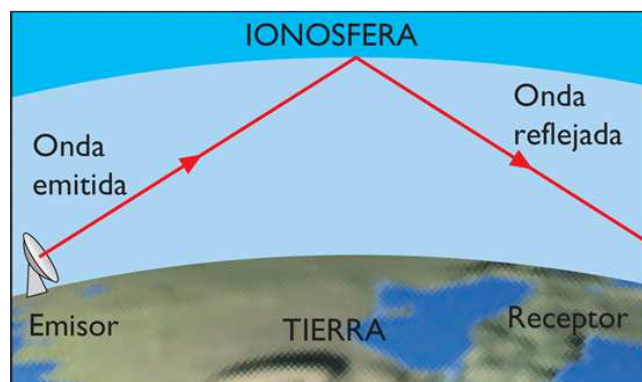


Figura II. 22. Reflexión de las ondas en la ionosfera

Las ondas electromagnéticas se propagan en línea recta. En consecuencia, si pretendemos enviar una señal de radio a larga distancia, dado que la Tierra tiene una superficie redondeada, la señal se alejará de la superficie de la Tierra y se perderá en el espacio. Sin embargo, las ondas de radio tienen la propiedad de reflejarse en las capas altas de la atmósfera, en concreto en la ionosfera.

La primera vez que se realizó una transmisión de radio a larga distancia fue en 1901. Entonces se desconocía la existencia de la ionosfera. Fue Marconi quien dispuso un transmisor y un receptor a ambos lados del Atlántico, entre Cornualles, en Inglaterra, y Terranova, en Canadá. Tras el éxito del experimento, Oliver Heavyside.

Kennelly descubrieron la existencia de la ionosfera y sus propiedades como reflectante de señales de cierta banda de frecuencias.

La ionosfera es la capa de la atmósfera situada entre los 90 y los 400 km de altura. Presenta la particularidad de que en ella los átomos se ionizan y liberan electrones por efecto de la luz solar. Según la concentración de iones, la ionosfera se puede dividir en varias capas, que se comportan de diferente forma ante la reflexión de las ondas. En cierto modo, al existir una nube electrónica en la ionosfera, esta se comporta como una pantalla para las señales eléctricas. No obstante, dependiendo de dicha concentración, existirá mayor o menor «blindaje» frente a las señales.

Capa de Ionosfera	Características de la capa	Rango frecuencias que refleja
Capa D	Capa apenas ionizada, únicamente refleja las frecuencias más bajas, debilitando las señales de frecuencias medias.	0-3 MHz

Capas E-F	Son dos capas de características similares, con igual comportamiento frente a las señales. Están situadas entre los 120 y 150 km, y reflejan las señales de frecuencia media. La reflexión de la señal depende de las variaciones en la concentración iónica entre el día, debido a la incidencia de la luz solar, y la noche, cuando decrece la concentración.	3-8 MHz
Capa F'	Es la capa más importante por la gran concentración de iones, cubriendo un espesor de hasta los 500 km.	8-15 MHz

Tabla II. VII. Características de las capas de la tierra

Las ondas reflejadas en la ionosfera que vuelven a la Tierra pueden ser de nuevo emitidas hacia el espacio y sufrir una segunda reflexión en la ionosfera. De hecho, este proceso se puede repetir sucesivas veces, de manera que las ondas podrán salvar grandes distancias gracias a las continuas reflexiones. Incluso, si se emite una señal con potencia y frecuencia adecuada, es posible que las ondas circunden la Tierra.

Por otra parte, no todas las frecuencias rebotan en la ionosfera. Las señales con frecuencias superiores a 15 MHz «escapan» a la reflexión de la atmósfera. Este es el rango de las señales de alta frecuencia (HF), de muy alta frecuencia (VHF), de ultra-alta frecuencia (UHF), así como de frecuencias superiores como las microondas. Este tipo de señales solo puede utilizarse para comunicaciones a corta distancia mediante estaciones terrestres repetidoras. Para salvar mayores distancias sería necesario recurrir al uso de satélites de comunicaciones.

2.3.2. COMPONENTES DE UN SISTEMA DE COMUNICACIÓN

En todo sistema de comunicación podemos distinguir los siguientes componentes:

- **Emisor:** es el elemento que transmite la información.
- **Receptor:** es el elemento que recibe la información.
- **Canal:** es el medio a través del cual tiene lugar el trasvase de información entre el emisor y el receptor.



Figura II. 23. Sistema de comunicación

El emisor y el receptor pueden encontrarse a unos pocos metros de distancia o bien estar alejados cientos e incluso miles de kilómetros, como ocurre en los sistemas de telecomunicaciones vía satélite o en los vuelos espaciales.

2.3.3. COMUNICACIÓN INALÁMBRICA

La comunicación inalámbrica (inglés *wireless*, sin cables) es aquella en la que extremos de la comunicación (emisor/receptor) no se encuentran unidos por un medio de propagación físico, sino que se utiliza la modulación de ondas electromagnéticas a través del espacio. En este sentido, los dispositivos físicos sólo están presentes en los emisores y receptores de la señal, entre los cuales encontramos: antenas, computadoras portátiles, PDA, teléfonos móviles, etc.¹

2.3.3.1. Aspecto histórico y generalidades

La comunicación inalámbrica, que se realiza a través de ondas de radiofrecuencia, facilita la operación en lugares donde la computadora no se encuentra en una ubicación fija (almacenes, oficinas de varios pisos, etc.); pero se trata de una tecnología sometida a investigación que en el futuro será utilizada de forma general.

Cabe también mencionar actualmente que las redes cableadas presentan ventaja en cuanto a transmisión de datos sobre las inalámbricas. Mientras que las cableadas proporcionan velocidades de hasta 1 Gbps (Red Gigabit), las inalámbricas alcanzan sólo hasta 108 Mbps.

Se puede realizar una “mezcla” entre inalámbricas y alámbricas, de manera que pueden funcionar de la siguiente manera: que el sistema cableado sea la parte principal y la inalámbrica sea la que le proporcione movilidad al equipo y al operador para desplazarse con facilidad en distintos campos (almacén u oficina).

Un ejemplo de redes a larga distancia son las Redes públicas de Conmutación por Radio. Estas redes no tienen problemas en pérdida de señal, debido a que su arquitectura está diseñada para soportar paquetes de datos en vez de comunicaciones por voz.

Actualmente, las transmisiones inalámbricas constituyen una eficaz herramienta que permite la transferencia de voz, datos y vídeo sin la necesidad de cableado. Esta transferencia de información es lograda a través de la emisión de ondas de radio teniendo dos ventajas: movilidad y flexibilidad del sistema en general.

En general, la tecnología inalámbrica utiliza ondas de radiofrecuencia de baja potencia y una banda específica, de uso libre para transmitir, entre dispositivos.

2.3.3.2. Campos de Utilización

La tendencia a la movilidad y la difusión hacen que cada vez sean más utilizados los sistemas inalámbricos, y el objetivo es ir evitando los cables en todo tipo de comunicación, no solo en el campo informático sino en televisión, telefonía, seguridad, etc.

2.3.3.3. Algunos problemas asociados con la tecnología inalámbrica

Los hornos de microondas utilizan radiaciones en el espectro de 2,45 Ghz. Es por ello que las redes y teléfonos inalámbricos que utilizan el espectro de 2,4 Ghz. pueden verse afectados por la proximidad de este tipo de hornos, que pueden producir interferencias en las comunicaciones. Otras veces, este tipo de interferencias provienen de una fuente que no es accidental. Mediante el uso de un perturbador o inhibidor de señal se puede dificultar e incluso imposibilitar las comunicaciones en un determinado rango de frecuencias.

2.3.3.4. Equipo inalámbrico

Son los equipos de punto de acceso que normalmente vienen con antena omni 2 Dbi, muchas veces desmontables, en las cuales se puede hacer enlaces por encima de los 500 metros y además se pueden interconectar entre sí. No debe haber obstáculos para que la señal sea excelente, ya que esto interfiere en la señal y puede haber problemas en la conexión.

2.4. RED INALÁMBRICA

El término red inalámbrica (Wireless network) en inglés es un término que se utiliza en informática para designar la conexión de nodos sin necesidad de una conexión física (cables), ésta se da por medio de ondas electromagnéticas. La transmisión y la recepción se realizan a través de puertos.

Una de sus principales ventajas es notable en los costos, ya que se elimina todo el cable ethernet y conexiones físicas entre nodos, pero también tiene una desventaja considerable ya que para este tipo de red se debe de tener una seguridad mucho más exigente y robusta para evitar a los intrusos. En la actualidad las redes inalámbricas son una de las tecnologías más prometedoras.

2.4.1. Categorías

Existen dos categorías de las redes inalámbricas.

1. **Larga distancia:** estas son utilizadas para distancias grandes como puede ser otra ciudad u otro país.
2. **Corta distancia:** son utilizadas para un mismo edificio o en varios edificios cercanos no muy retirados.

2.4.2. Aplicaciones

- ✓ Las bandas más importantes con aplicaciones inalámbricas, del rango de frecuencias que abarcan las ondas de radio, son la VLF (comunicaciones en navegación y submarinos), LF (radio AM de onda larga), MF (radio AM de onda media), HF (radio AM de onda corta), VHF (radio FM y TV), UHF (TV).

- ✓ Mediante las microondas terrestres, existen diferentes aplicaciones basadas en protocolos como Bluetooth o ZigBee para interconectar ordenadores portátiles, teléfonos u otros aparatos. También se utilizan las microondas para comunicaciones con radares (detección de velocidad u otras características de objetos remotos) y para la televisión digital terrestre.

- ✓ Las microondas por satélite se usan para la difusión de televisión por satélite, transmisión telefónica a larga distancia y en redes privadas, por ejemplo.

- ✓ Los infrarrojos tienen aplicaciones como la comunicación a corta distancia de los ordenadores con sus periféricos. También se utilizan para mandos a distancia, ya que así no interfieren con otras señales electromagnéticas, por ejemplo la señal de televisión. Uno de los estándares más usados en estas comunicaciones es el IrDA (*Infrared Data Association*). Otros usos que tienen los infrarrojos son técnicas como la tomografía, la cual permite determinar la temperatura de objetos a distancia.

2.5. RADIOFRECUENCIA

También llamada espectro de radiofrecuencia o RF, se aplica a la porción menos energética del espectro electromagnético situado entre 3 Hz y 300GHz.

Las ondas electromagnéticas pueden transmitirse aplicando la corriente alterna originada en un generador a una antena.

2.5.1. Características Generales

Permite comunicaciones de corto y medio alcance, puede atravesar obstáculos y paredes, el campo de aplicación es muy grande.

Dentro del rango de frecuencias que abarca dicha tecnología, únicamente nos interesa el rango de frecuencias que va de 30-3000MHz y cuya longitud de onda es de 1m-100mm, ya que es la banda donde encontramos las comunicaciones inalámbricas.

Esta tecnología no tiene un protocolo establecido, es por eso que no tiene unas características generales, una arquitectura de funcionamiento, etc, hecho que también la hace más versátil y dependiente de las características del dispositivo utilizado en la comunicación.

Para poder establecer una comunicación multipunto es necesario crear un protocolo que permita el intercambio de información entre los diferentes dispositivos.

Este protocolo debe contener tanto unos bits asignados a la información que quiere transmitirse, como unos bits que indiquen el dispositivo receptor de la información, bits de error para evitar pérdidas de comunicación, bits destinados a definir la modulación de dicha información, etc.

El protocolo RF más usado para la comunicación inalámbrica de dispositivos es el RS- 232. La velocidad de transmisión y de recepción de información depende en exclusiva de los dispositivos utilizados. Para evitar interferencias la velocidad de transmisión no debe superar los 4800Baudios y la de recepción los 2400Baudios. Como el mismo dispositivo integra tanto transmisor como receptor, el baud rate máximo para evitar las interferencias está, alrededor de los 2400Baudios.

2.5.2. Tipos de Comunicaciones Inalámbricas por RF

Las transmisiones de datos entre equipos electrónicos sin cables se están aplicando cada vez más debido a los medios tecnológicos actuales, que son los circuitos integrados que permiten hacer un diseño sin tener demasiados conocimientos de RF, ni disponer de cada instrumentación para RF, ya que estos dispositivos requieren pocos componentes externos y ningún tipo de ajuste en RF.

Primero se usaron módulos de RF con componentes discretos unidireccionales y precisamente para no tener que depender del diseño de una circuitería en RF. Posteriormente con la aparición de circuitos transmisores completamente integrados con las funciones de emisor y receptor, en diferentes bandas de frecuencia que se fueron estandarizando en las diferentes zonas (Europa y USA), han permitido poderlos utilizar en los diferentes campos de aplicación industrial, comercial, y medico, como: control remoto, transmisión de datos en sensores o sistemas de adquisición de datos, en monitorización médica o de la salud, etc.

Las comunicaciones inalámbricas por RF se pueden dividir en las que no cumplen ningún protocolo estándar y las que cumplen un protocolo estándar, y en las normativas sobre sus distintas frecuencias de trabajo, que a la vez definen velocidad de transmisión o ancho de banda y campo de aplicación.

2.6. MOTORES

Un motor eléctrico es una máquina eléctrica que transforma energía eléctrica en energía mecánica por medio de interacciones electromagnéticas.

2.6.1. Servo Motores HITEC HS-311



Figura II. 24. Servomotor

Es un aparato destinado a vencer grandes resistencias, mediante la amplificación de fuerzas aplicadas, relativamente más pequeñas; que consta de dos bombines hidráulicas.

El servomotor mostrado en la Figura II.24 fue diseñado como un actuador mecánico en cualquier sistema de servomecanismo que tiene por objeto llevar al sistema a una lectura cero, es decir la señal de error se redujo a cero, y existen dos tipos de servomotores, de corriente directa y de corriente alterna.

Sus características fundamentales para cd y ca se basaron en lo siguiente: la primera radicó en que el par de salida del motor sea aproximadamente proporcional a su voltaje de control aplicado y la segunda se basa en el hecho de que la dirección del par éste determinada por la polaridad instantánea del voltaje de control. El servomotor de dc campo controlado es aquel donde el par que produjo este motor es cero cuando el amplificador de error de dc no le suministra excitación de campo, y si se invierte la polaridad del campo, se invierte la dirección del motor.

2.6.1.1. Características

Velocidad:	0.23 seg/60 grados (260 grados/seg)
Par de salida:	3.7 Kg-cm
Dimensiones:	40 x 20 x 37 mm
Peso:	43 gr
Frec. PWM:	60Hz (20ms)
Rango giro:	180 grados

Tabla II. VIII. Características de las capas de la tierra

2.6.1.2. Control

Los servos se controlan aplicando una señal PWM por su cable de control. Las señales PWM (Pulse Width Modulation, Modulación por anchura de pulso) son digitales (pueden valer 0 ó 1) y permiten que usando un único pin de un microcontrolador podamos posicionar el servo. Esto es una gran ventaja, porque si por ejemplo, disponemos de un micro con 8 pines de salida, podremos posicionar 8 servos.

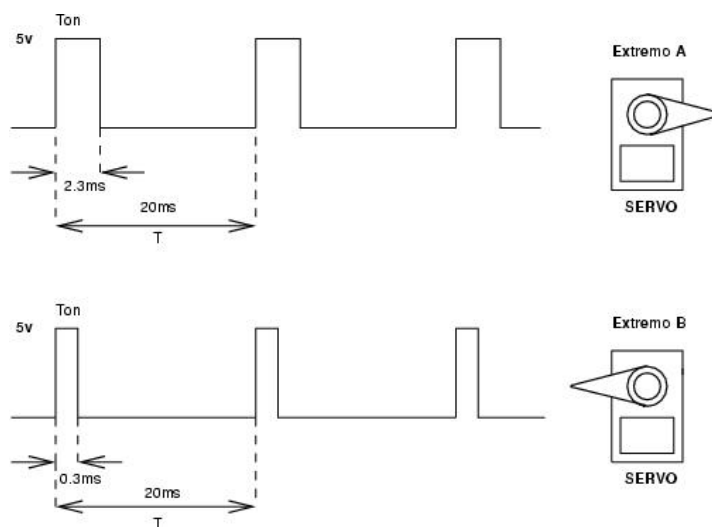


Figura II. 25. Ancho de Pulso

Para posicionar el servo hay que aplicar una señal periódica, de 60Hz. La anchura del pulso determina la posición del servo. Si la anchura es de 2.3ms, el servo se sitúa en un extremo y si la anchura es de 0.3ms se sitúa en el opuesto. Cualquier otra anchura entre 0.3 y 2.3 sitúa el servo en una posición comprendida entre un extremo y otro. Por ejemplo, si queremos que se sitúe exactamente en el centro, aplicamos una anchura de 1.3ms.

Cuando se deja de enviar la señal, el servo entra en un estado de reposo, y por tanto se podrá mover con la mano. Mientras se le aplique la señal, permanecerá fijado en su posición, haciendo fuerza para permanecer en ella.

2.6.1.3. Conexión

El servo dispone de un conector al que llegan tres cables. El rojo es el de alimentación (4.5-6 voltios). El negro es masa. Y el blanco es el de la señal de control

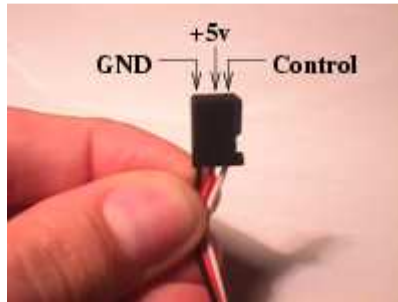


Figura II. 26. Conexión del Servo

2.6.1.4. Driver de Potencia para Control de Motores

La etapa de potencia es el elemento que hace de intermediario entre las partes mecánicas y electrónicas. La necesidad de usar esta etapa intermedia radica en la escasa capacidad para suministrar corriente, ya que por sí sola no sería capaz de mover los motores. El método más sencillo para lograr la intensidad requerida es mediante el uso de un transistor de cierta potencia funcionando como un interruptor controlado por la señal lógica emitida, es lo que comúnmente llamamos driver de corriente; de este modo, el circuito digital sólo enciende y apaga el transistor, siendo este el que enciende y apaga el motor.

La mayor parte de los dispositivos electromecánicos son inductivos, como es el caso de los motores, por lo que la corriente en ellos no puede variar de forma instantánea, tendiendo a mantener durante unos instantes la corriente en circulación tras la apertura del circuito, pudiendo generarse una chispa capaz de dañar el elemento que realizó la desconexión. Para evitar esto, se añade un diodo de protección a fin desahogar la corriente residual producida por el elemento inductivo.

CAPÍTULO III

MICROCONTROLADORES APROPIADOS PARA EL CIRCUITO

3.1. MICROCONTROLADORES

3.1.1. Introducción

En el año de 1980 aproximadamente, los diferentes fabricantes de circuitos integrados iniciaron la difusión de un nuevo circuito para control, medición e instrumentación al que llamaron microcomputador en un sólo chip o de manera más exacta microcontrolador.

El microcontrolador es un circuito integrado que contiene toda la estructura de un microcomputador o microprocesador, o sea CPU, RAM, ROM y circuitos de entrada y salida. En la actualidad existen microcontroladores más especializados que poseen además convertidores análogos digital, temporizadores, contadores y un sistema para permitir la

comunicación en serie y en paralelo; pudiendo crear aplicaciones ilimitadas con los microcontroladores, todo dependerá del diseño y las funciones que debe realizar.

3.1.2. Definición de Microcontrolador

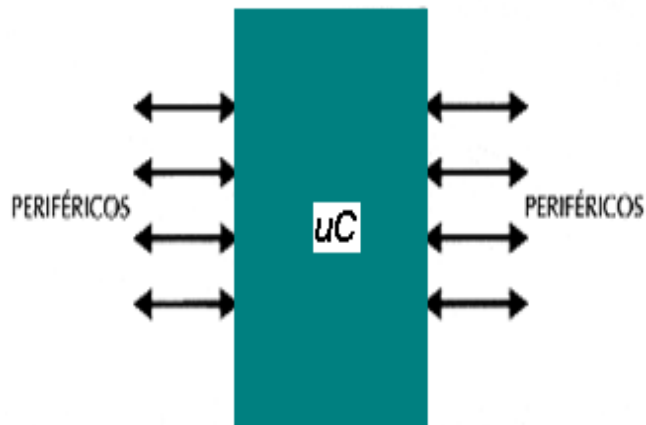


Figura III. 27. Microcontrolador y Periféricos

Como se observa la Figura III. 27 al microcontrolador se lo describe como un dispositivo electrónico capaz de llevar a cabo diferentes procesos lógicos que le permiten cumplir una determinada meta u objetivo. Dichos procesos o acciones son programados en lenguaje ensamblador o un lenguaje de alto nivel por el usuario, y son introducidos en este a través de un determinado programador hacia el microcontrolador para que realice las acciones o procesos para los cuales fue programado. Al ser el microcontrolador un sistema cerrado, todas las partes del procesador están contenidas en su interior y sólo salen al exterior las líneas que gobiernan los periféricos.

3.1.3. Características de los Microcontroladores

Entre las principales características que gobiernan a un microcontrolador se enumeran las siguientes:

- La arquitectura del procesador sigue el modelo Harvard permitiendo a la CPU acceder simultáneamente a las dos memorias.
- La segmentación permite al procesador realizar al mismo tiempo la ejecución de una instrucción y la búsqueda del código de la siguiente. De esta forma se puede ejecutar cada instrucción en un ciclo.
- Todas las instrucciones de los microcontroladores de la gama baja tienen una longitud de 12 bits. Las de la gama media tienen 14 bits y más las de la gama alta. Esta característica es muy ventajosa en la optimización de la memoria de instrucciones y facilita enormemente la construcción de ensambladores y compiladores.
- Todas las instrucciones son ortogonales es decir cualquier instrucción puede manejar cualquier elemento de la arquitectura como fuente o como destino.
- Arquitectura basada en un “banco de registros”, esto significa que todos los objetos del sistema (puertas de E/S, temporizadores, posiciones de memoria, están implementados físicamente como registros.

3.1.4. Tipos de Arquitecturas de Microcontroladores

Las diferentes arquitecturas se presentan a continuación:

3.1.4.1. Arquitectura Von Neumann

Esta arquitectura ha sido tradicional de computadoras y microprocesadores está basada en la arquitectura Von Neumann, en la cual la unidad central de proceso (CPU), está conectada a una

memoria única donde se guardan las instrucciones del programa y los datos. El tamaño de la unidad de datos o instrucciones está fijado por el ancho del bus que comunica la memoria con la CPU.

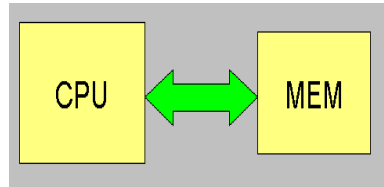


Figura III. 28. Arquitectura Von Neumann

3.1.4.2. Arquitectura Harvard

La arquitectura Harvard se la describe como la unidad central de proceso (CPU) conectada a dos memorias (una con las instrucciones y otra con los datos) por medio de dos buses diferentes. Una de las memorias contiene solamente las instrucciones del programa (Memoria de Programa), y la otra sólo almacena datos (Memoria de Datos). Ambos buses son totalmente independientes y pueden ser de distintos anchos.

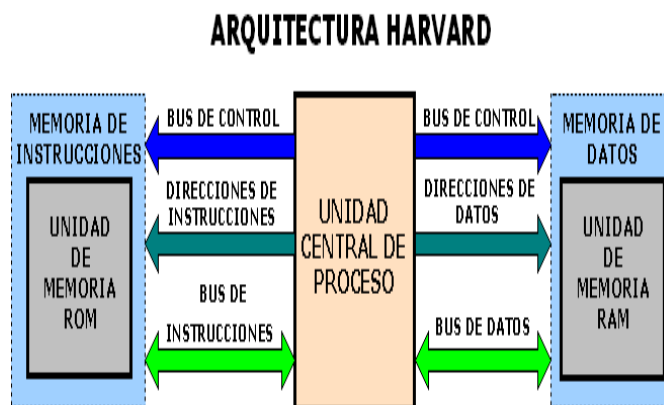


Figura III. 29. Arquitectura Harvard

3.1.5. Las Gammas de Pic

Los microcontroladores se han agrupado dependiendo de sus capacidades y características, a continuación se muestra la forma como están distribuidos los mismos.

3.1.5.1. Gama Enana PIC12C (F) XXX

Se trata de un grupo de PIC que en su aparición había acaparado la atención del mercado electrónico. Su principal característica fue su reducido tamaño, al disponer todos sus componentes de 8 patitas. Se alimentan con un voltaje de corriente continua comprendido entre 2,5 V y 5,5 V, y consumen menos de 2 mA cuando trabajan a 5 V y 4 MHz de ciclo de reloj. El formato de sus instrucciones puede ser de 12 o de 14 bits y su repertorio es de 33 o 35 instrucciones, respectivamente.

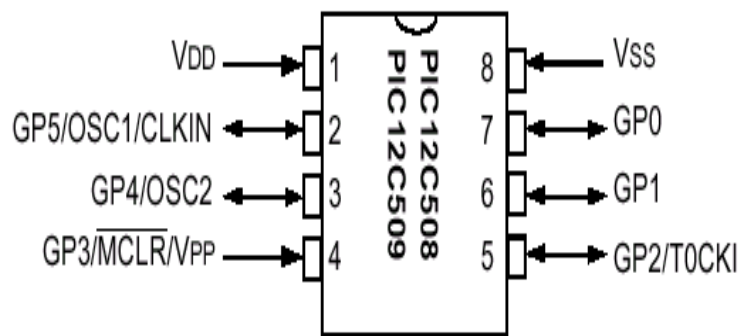


Figura III. 30. Diagrama de conexiones de los PIC12Cxxx de la gama enana.

Aunque los PIC de gama enana sólo tienen 8 patitas, pueden destinar hasta 6 como líneas de E/S para los periféricos porque disponen de un oscilador interno R-C, con el cual trabajan.

3.1.5.2. Gama Baja PIC16C5X

Se trata de una serie de PIC de recursos limitados, pero con una de las mejores relaciones costo/prestaciones. Sus versiones están encapsuladas con 18 y 28 patitas y pueden alimentarse a partir de una tensión de 2,5 V, lo que les hace ideales en las aplicaciones que funcionan con pilas teniendo en cuenta su bajo consumo (menos de 2 mA a 5 V y 4 MHz), con un repertorio de

33 instrucciones cuyo formato consta de 12 bits. No admiten ningún tipo de interrupción y la Pila sólo dispone de dos niveles.

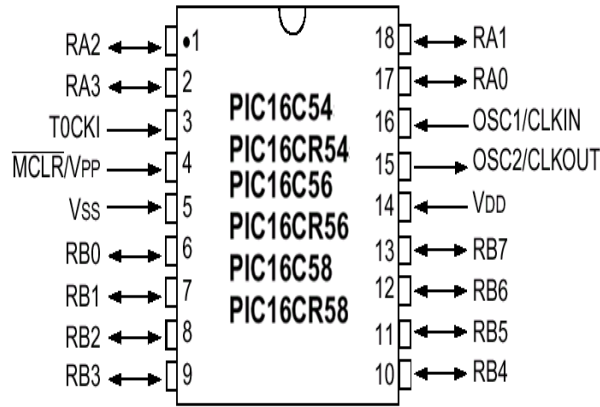


Figura III. 31. Diagrama de conexiones de los PIC16C54/56 de la gama baja

3.1.5.3. Gama Media PIC16CXXX

Es la gama más variada y completa de estos microcontroladores, la misma, abarca diferentes modelos con encapsulado desde 18 pines hasta los 68 pines, cubriendo variadas opciones que integran abundantes periféricos y dispositivos de entrada y salida.

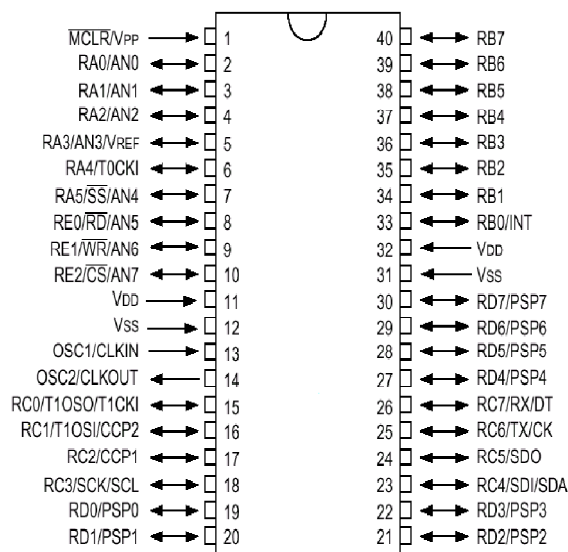


Figura III. 32. Diagrama de conexiones del PIC16C74, de la gama media

Al tener mayor cantidad de pines se puede controlar mayor cantidad de dispositivos con un solo chip, todo dependerá de la capacidad de memoria del mismo y de la forma de programación con la que se haya realizado dicho programa de control. En esta gama sus componentes añaden nuevas prestaciones a las que poseían los de la gama baja, haciéndoles más adecuados en las aplicaciones complejas. Admiten interrupciones, poseen comparadores de magnitudes analógicas, convertidores A/D, puertos serie y diversos temporizadores.

3.1.5.4. Gama Alta PIC17CXXX

Se alcanzan las 58 instrucciones de 16 bits en el repertorio y sus modelos disponen de un sistema de gestión de interrupciones vectorizadas muy potente. También incluyen variados controladores de periféricos, puertas de comunicación serie y paralelo con elementos externos, un multiplicador hardware de gran velocidad y mayores capacidades de memoria, que alcanza los 8 k palabras en la memoria de instrucciones y 454 bytes en la memoria de datos. Quizás la característica más destacable de los componentes de esta gama es su arquitectura abierta, que consiste en la posibilidad de ampliación del microcontrolador con elementos externos.

3.1.6. Herramientas para el desarrollo de aplicaciones

Para el desarrollo de las diferentes funciones que realiza el microcontrolador existen diferentes programas que facilitan el desarrollo de las mismas, las principales herramientas de ayuda al desarrollo de sistemas basados en microcontroladores son:

3.1.6.1. Ensamblador

La programación en lenguaje ensamblador permite desarrollar programas muy eficientes, ya que otorga al programador el dominio absoluto del sistema, aunque este lenguaje por su larga forma de programación ha sido reemplazado por lenguajes de alto nivel que permiten realizar la programación de forma más rápida y eficiente. Al ser el primer tipo de lenguaje que se utilizó sigue siendo usado para la enseñanza y aprendizaje del manejo de microcontroladores, ya que el programador decide qué tipo de lenguaje utilizar para el desarrollo de su aplicación.

3.1.6.2. Compilador

La programación en un lenguaje de alto nivel (como el C) permite disminuir el tiempo de desarrollo de una determinada aplicación. No obstante, si no se programa con cuidado, el código resultante puede ser mucho más ineficiente que el programado en ensamblador.

Es por ello que se utiliza un determinado compilador que depende del lenguaje de alto nivel con el que estemos programando, el cual nos ayuda a determinar si las sentencias de dicho código están correctamente escritas o caso contrario procede a señalar el error.

3.1.6.3. Simulador

Son capaces de ejecutar en un PC programas realizados para el microcontrolador. Los simuladores permiten tener un control absoluto sobre la ejecución de un programa, siendo ideales para la depuración de los mismos. Su gran inconveniente es que es difícil simular la entrada y salida de datos del microcontrolador. Tampoco cuentan con los posibles ruidos en las entradas, pero, al menos, permiten el paso físico de la implementación de un modo más seguro y menos costoso, puesto que ahorraremos en grabaciones de chips para la prueba.

3.1.6.4. Programador

Es un dispositivo físico o hardware que es conectado a un PC con el cual se permite grabar en el microcontrolador el programa desarrollado, el diseño del mismo depende del software que se utilice para su manejo.

Existen diferentes tipos de programadores los mismos que tienen conexión a la PC mediante el puerto paralelo, el puerto serial o el puerto USB necesarios para la comunicación con el microcontrolador.

3.1.6.5. Paquetes IDE

Actualmente existen paquetes de software denominados "Entornos de Desarrollo Integrado", IDE, que suelen funcionar bajo Windows y que incluyen editores de texto para el ensamblador o el compilador, permiten la simulación del programa y también pueden integrar el control de emuladores y programadores de dispositivos.

3.2. MÉTRICAS PARA ELEGIR UN MICROCONTROLADOR

Entre las características que ofrecen los microcontroladores se debe tomar en cuenta para su elección:

- ✓ **Sencillez de programación, manejo y grabación:** El encapsulado se recomienda que sea de tipo DIP para facilitar el modo de grabación y realizar las pruebas en un Bread Board.
- ✓ **Capacidad de Memoria:** El microprocesador debe tener la memoria de programa (Memoria Flash) en un rango entre 16 Kb y 32 Kb.

- ✓ **Alta velocidad**
- ✓ **Bajo precio:** Esta métrica tiene gran importancia para que permita introducir al mercado el producto con mayor facilidad.
- ✓ **Disponibilidad:** Se refiere a la facilidad de encontrarlo en nuestro país.
- ✓ **Tamaño:** El factor más importante.

3.2.1. Estudio de la mejor alternativa de microcontrolador.

Aplicamos el Método Ponderado para elegir los microcontroladores. Para lo cual asignamos pesos o valores para la ponderación de 1 a 10 (Tabla III.9).

Métricas o factores	Valores para la ponderación	Ponderación
Programación, manejo y grabación	10	0.23
Velocidad	8	0.19
Memoria	7	0.16
tamaño	10	0.23
Bajo precio y disponibilidad	8	0.19
Total	43	1.00

Tabla II. IX. Asignación cuantitativa para las métricas

Asignamos calificaciones o costos según la información presentada de los microcontroladores.

Factor Alternativa	Sencillez, manejo y programación	Velocidad	Memoria	Tamaño	Bajo precio y disponibilidad
16f628A	8	7	7	10	8
16F877A	8	8	8	8	7
PIC17CXXX	8	9	9	6	5

Tabla II. X. Calificaciones y costos

Con la información de la tabla III.9 obtenemos la matriz de puntos para determinar los microcontroladores adecuados. De la tabla (Tabla III.9) determinamos que los microcontroladores adecuados para el proyecto son el PIC16F628A y el PIC16F877A.

Factores	Ponderación	16F628A		16F877A		17CXXX	
		Calif	Calif. P	Calif.	Calif. P	Calif.	Calif. P
Sencillez, manejo y programación	0.23	8	1.84	8	1.84	8	1.84
Velocidad	0.19	7	1.33	8	1.52	8	1.52
Memoria	0.16	7	1.12	8	1.28	9	1.38

Tamaño	0.23	10	2.3	8	1.81	6	1.21
Bajo precio y disponibilidad	0.19	8	1.52	7	1.54	5	1.10
total			8.11		7.99		7.05

Tabla II. XI. Matriz de puntos

Por consiguiente se ha elegido Pic16F628A para el transmisor y el Pic16F877A para el receptor

3.3. MICROCONTROLADOR 16F628A



Figura III. 33. Encapsulado del PIC 16f628

3.3.1. Principales características

- ✓ Set de instrucciones reducido (RISC). Sólo 35 instrucciones.
- ✓ Las instrucciones se ejecutan en un sólo ciclo de máquina excepto los saltos que requieren 2 ciclos.
- ✓ Opera con una frecuencia de reloj de hasta 20 MHz (ciclo de máquina de 200 ns)
- ✓ Memoria de programa: 2048 posiciones de 14 bits.
- ✓ Memoria de datos: Memoria RAM de 224 bytes (8 bits por registro).

- ✓ Memoria EEPROM: 128 bytes (8 bits por registro).
- ✓ Stack de 8 niveles.
- ✓ 16 Terminales de I/O que soportan corrientes de hasta 25 mA.
- ✓ 3 Timers.
- ✓ Módulos de comunicación serie, comparadores, PWM.

3.3.2. Características especiales:

- ✓ La memoria de programa se puede reescribir hasta 1000 veces.
- ✓ La memoria EEPROM se puede reescribir hasta 1000000 de veces.
- ✓ Los datos almacenados en la memoria EEPROM se retienen por 40 años y no se borran al quitar la alimentación al circuito.
- ✓ 10 fuentes de interrupción, algunas de ellas son:
 - Señal externa (RB0).
 - Desborde de TMR0 y TMR1.
- ✓ Cambio en el estado de los terminales RB4, RB5, RB6 o RB7.
- ✓ Ciclo de escritura en la memoria EEPROM completado

3.3.3. Descripción de los terminales

3.3.3.1. Terminales de entrada-salida (16 en total)

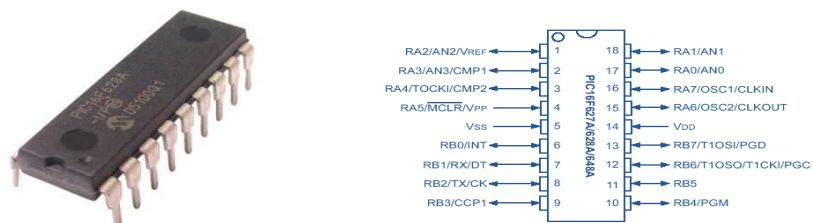


Figura III. 34. Diagrama de pines 16F628

3.3.3.2. PORTA: RA0-RA7:

- Los terminales RA0-RA3 y RA6 – RA7 son bidireccionales y manejan señales TTL.
- El terminal RA5 es una entrada Schmitt Trigger que sirve también para entrar en el modo de programación cuando se aplica una tensión igual a V_{pp} (13,4V mínimo).
- El terminal RA4 como entrada es Schmitt Trigger y cómo salida es colector abierto. Este terminal puede configurarse como reloj de entrada para el contador TMR0.

3.3.3.3. PORTB: RB0-RB7:

- Los terminales RB0-RB7 son bidireccionales y manejan señales TTL.
- Por software se pueden activar las resistencias de pull-up internas, que evitan el uso de resistencias externas en caso de que los terminales se utilicen como entrada (permite, en algunos casos, reducir el número de componentes externos).
- RB0 se puede utilizar como entrada de pulsos para provocar una interrupción proveniente del exterior.
- RB4-RB7 están diseñados para detectar la interrupción por cambio de estado, por ejemplo, cuando se pulsa una tecla de un teclado matricial.

3.3.3.4. Otros terminales.

- VDD: Positivo de alimentación. 2 a 5,5 Vcc.
- VSS: Negativo de alimentación.
- MCLR: Master Clear (Reset). Si el nivel lógico de este terminal es bajo (0 Vcc), el microcontrolador permanece inactivo.
- OSC1/CLKIN: Entrada del oscilador (cristal). Entrada de oscilador externo.
- OSC2/CLKOUT: Salida del oscilador (cristal).

3.3.4. Aspecto Interno del PIC 16F628.

Nuestro PIC pertenece a la familia de la gama media con una compactación de código superior a la de sus competidores, incorpora tres características de avanzada que son:

- Procesador tipo RISC (Computadores de Juego de Instrucciones Reducido).
- Procesador segmentado.
- Arquitectura HARVARD.

Con estos recursos el PIC es capaz de ejecutar instrucciones solamente en un ciclo de instrucción, salvo las de salto, que tardan el doble.

El juego de instrucciones se reduce a 35 y sus modos de direccionado se ha simplificado al máximo.

Con la estructura segmentada se pueden realizar simultáneamente las dos fases en que se descompone cada instrucción. Al mismo tiempo que se está desarrollando la fase de ejecución de una instrucción se realiza la fase de búsqueda de la siguiente.

La separación de los dos tipos de memoria son los pilares de la arquitectura Harvard, gracias a esto se puede acceder en forma simultánea e independiente a la memoria de datos y a la de instrucciones.

Al tener memorias separadas permite que cada una de ellas tenga el ancho y tamaño más adecuado. De esta manera en el PIC 16F628A el ancho de los datos es de un byte, mientras que la de las instrucciones es de 14 bits.

Otra característica de los PICs 16F628A es el manejo de los bancos de registros, los cuales participan de forma muy activa cuando se ejecutan las instrucciones.

En línea general, los registros se clasifican como de uso general (GPR – General Purpose Registers) y los especiales (SFR – Special Function Registers).

Los registros de uso general pueden ser usados a voluntad por el usuario, sin existir restricciones. Pueden servir para almacenar resultados que se reciben desde el registro W (works), datos que provienen de las puertas de entradas, etc.

Los registros de uso específicos no pueden ser usados como los anteriores. La verdad, que son ellos los que controlan prácticamente todo el funcionamiento de los microcontroladores, pues toda la configuración necesaria para el perfecto funcionamiento del microcontrolador es hecho a través de algún tipo de SFR.

3.3.5. El Procesador.

El procesador responde a la arquitectura RISC, que se identifica porque el juego de instrucciones se reduce a 35, donde la mayoría se ejecutan en un ciclo de reloj, excepto las instrucciones de salto que necesitan dos ciclos.

La ALU (Unidad Aritmético Lógica), ubicada dentro del procesador efectúa las operaciones lógicas - aritméticas con dos operandos, uno que recibe desde el registro W (registro de trabajo) y otro que puede provenir de cualquier registro o del propio código de la instrucción. El resultado de la operación puede almacenarse en cualquier registro o en el propio W.

3.3.6. La Memoria de Datos.

Formada por dos bloques bien diferenciados:

- El primero, es una memoria del tipo RAM (SRAM), dividida en cuatro bancos, los cuales contienen los registros de uso general (GPR), y los registros de uso específicos (SFR).
- Los registros especiales SFR ocupan los primeros 32 lugares de cada banco.
- Los registros generales GPR son 224 de 8 bits cada uno, y se pueden acceder de forma directa o indirecta a través del registro FSR.
- El segundo, es una memoria EEPROM, que puede almacenar datos que no se deseen perder una vez que se quita la alimentación, su acceso está controlado por unos registros especiales.
- Esta memoria es del tipo SRAM, formado por registros de 8 bits y organizada en dos áreas:
 - Registros de funciones especiales, controlan el funcionamiento del dispositivo.
 - Registros de propósito general, sirven para guardar datos temporalmente mientras se ejecuta el programa.

La memoria a su vez se divide en cuatro bancos (banco 0 al banco 3), para cambiar de banco se utilizan los bits 5 y 6 (RP0 y RP1 respectivamente) del registro STATUS.

3.3.7. Direccionamiento de la Memoria de Datos.

3.3.7.1. Direccionamiento Directo:

Es el modo más empleado. El operando que utiliza la instrucción en curso se referencia mediante su dirección, que viene incluida en el código OP de la misma, concretamente en los 7 bits de menos peso.

3.3.7.2. Direccionamiento Indirecto:

En este modo se emplea los registros INDF (posición 00h de la memoria de datos) y el registro FSR (File Select Register ubicado en la posición 04h de la memoria de datos).

La forma que funciona el direccionamiento indirecto es que la dirección de memoria del registro al que se accede se introduce en el registro FSR. Los 7 bits de menos peso apuntan la posición y de más peso, junto con el bit IRP del registro STATUS, seleccionan el banco. Este modo permite acceder a 256 direcciones.

Cuando se quiere acceder a un registro cuya dirección esta almacenada en el registro FSR, se usará el registro INDF. Este registro no está implementado físicamente, por lo que no se accede directamente a él.

Por ejemplo; la instrucción: `movwf INDF`, esta instrucción traslada el contenido del registro W al registro apuntado por FSR.

3.3.8. La memoria de Programa.

Es una memoria no volátil de tipo Flash, lo que permite que se borre y escriba eléctricamente. En el PIC16F628A se pueden almacenar hasta 2048 instrucciones de 14 bits cada uno.

3.3.9. El Reloj

Para que el PIC pueda procesar las instrucciones, es necesario un reloj cuya frecuencia es parámetro fundamental en el momento de establecer la velocidad de ejecución de las instrucciones y en el consumo de energía.

El tiempo en que tarda en ejecutarse una instrucción se llama ciclo de la instrucción.

En los PICs, un ciclo de instrucción emplea 4 períodos de reloj. Todas las instrucciones del PIC se realizan en un ciclo de instrucción, salvo las de salto, que tardan el doble. El PIC 16F628A posee 8 formas de configurar al oscilador. Un circuito RC como oscilador brinda una solución económica. El tipo LP empleado en aplicaciones de bajo consumo. El modo XT solo usa un cristal, el HS emplea cristales de alta velocidad. Por último el oscilador interno INTOSC configurable a alta o baja velocidad.

Los microcontroladores PIC(Peripheral Interface Controller), son fabricados por la empresa MICROCHIP Technology INC. Cuya central se encuentra en Chandler Arizona, esta empresa ocupa el primer lugar en venta de microcontroladores de 8 bits desde el año 2002. Su gran éxito se debe a la gran variedad (más de 180 modelos), gran versatilidad, gran velocidad, bajo costo, bajo consumo de potencia, y gran disponibilidad de herramientas para su programación.

3.4. MICROCONTROLADOR 16F877A.

En la elaboración de esta tesis se utilizo el microcontrolador 16F877A, a continuación describiremos las características microcontrolador, el mismo que es uno de los elementos primordiales es la base del modulo de adquisición de datos y el que interpreta las ordenes desde el pulsador, en la figura III.35 se muestra los pines del PIC.

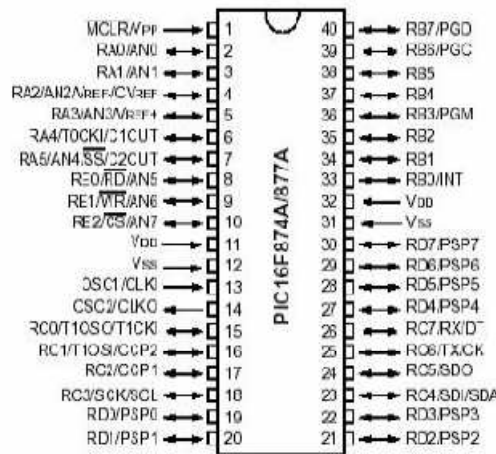
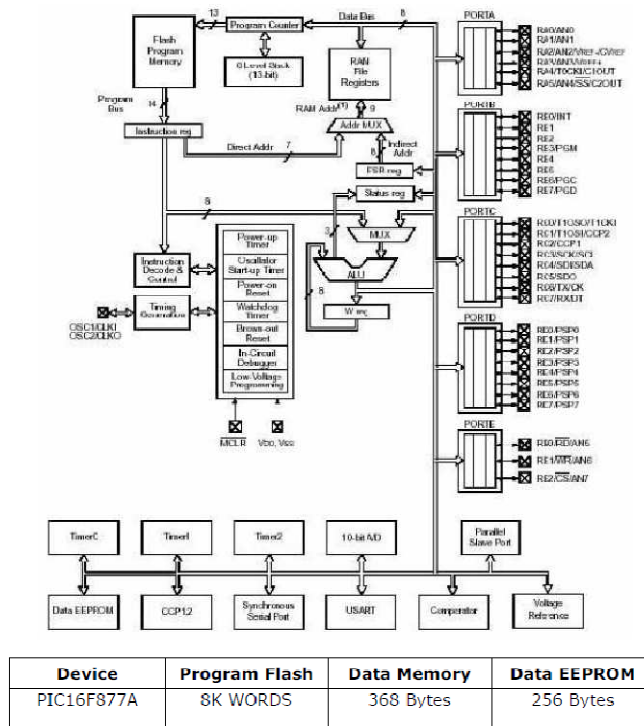


Figura III. 35. Pines del PIC 16F877A

Obviamente no todos los pines fueron utilizados en el PIC16F877A, solamente los necesarios.

Los pines correspondientes a la alimentación, oscilador de cristal de 20b MHz, tierra, y algunos pines correspondientes al Puerto B.

A continuación en la Figura III.36., se muestra la Arquitectura Interna del PIC 16F877A.



Device	Program Flash	Data Memory	Data EEPROM
PIC16F877A	8K WORDS	368 Bytes	256 Bytes

Figura III. 36. Arquitectura Interna del Pic 16F877A.

3.4.1. Puertos paralelos de entrada / salida del Pic 16f877a.

El circuito integrado PIC16F877 posee 5 puertos de entrada / salida, denominados PORTA, PORTB,..., PORTE, configuradas para trabajar como entradas o como salidas a selección del programador.

3.4.1.1. Puerto A (Porta)

El puerto A posee 6 líneas bidireccionales, los 3 registros asociados a este puerto son:

Registro PORTA (05H). Registro de estado del Puerto A. Cada uno de los 6 bits menos significativos (RA5,..., RA0) de este registro están asociados a la línea física correspondiente del puerto. Al hacer una lectura este registro se lee el estado de todas las patitas del puerto. Todas las escrituras al registro son operaciones del tipo “lee modifica escribe”, es decir, toda escritura al puerto implica que el estado de las patitas es leído, luego es modificado y posteriormente se escribe en la Figura III.29., se muestra el Registro PORTA.

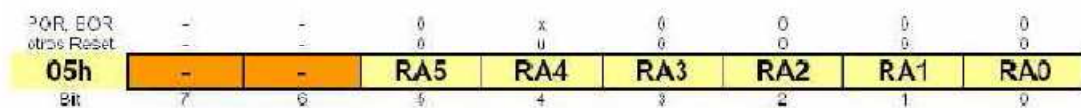


Figura III. 37. Registro PORTA (05H).

Registro TRISA (85H). Cada bit de este registro configura la dirección en que fluye la Información de la patita correspondiente del puerto A, así, para $k = 0, 1, \dots, 5$. Bit k de TRISA = 1 configura la patita RA_k del puerto A como Entrada Bit k de TRISA = 0 configura la patita RA_k del puerto A como Salida en la Figura III.38. Se muestra el Registro TRISA (85H).



Figura III. 38. Registro TRISA (85H).

Todas las patitas del puerto A poseen diodos de protección conectados a Vdd (contra altos voltajes de entrada) y a Vss (contra voltajes negativos) además, manejan niveles de entrada tipo TTL y como salidas se comportan como drivers tipo CMOS. Excepto la patita RA4, la cual como entrada posee un Disparador Schmitt y como salida es de Drenaje abierto, además RA4 sólo posee un diodo de protección conectado a Vss.

Registro ADCON1 (9FH). Las patitas RA0, RA1, RA2, RA3 y RA5 están multiplexadas con las entradas analógicas AN0,..., AN4, de manera que antes de utilizarlas debemos configurar si serán usadas como entradas analógicas o como entradas/salidas digitales.

3.4.1.2. Puerto B (Portb)

El puerto B es un puerto digital de 8 bits, todas sus patitas son bidireccionales y trabaja en forma similar al puerto A. Tiene tres registros asociados: El registro de datos PORTB, el registro de dirección de los datos TRISB y el registro OPTION_REG.

Registro PORTB (06H, 106H). Los ocho bits que contiene reflejan directamente el estado de las ocho patitas del puerto B RB0,..., RB7.

Registro TRISB (86H, 186H). En forma similar a TRISA, al poner un 0 en un bit de TRISB se configura la patita RB correspondiente como salida y al poner un 1 en un bit de TRISB se configura la patita RB correspondiente como entrada.

Las Patitas RB4,..., RB7. Estas cuatro patitas del puerto B tienen la capacidad de generar una solicitud de interrupción a la CPU cuando están configuradas como entradas. El estado de estas patitas es comparado con el último estado que tenían durante la última lectura a PORTB, los bits que indican que hay una diferencia entre estos valores por cada patita están conectados a una puerta OR cuya salida activa el bit RBIF del registro INTCON solicitando con esto una interrupción.

Esta interrupción es especialmente útil para despertar al dispositivo de su estado de SLEEP cuando alguna de las cuatro líneas es activada, por ejemplo, en respuesta a la presión de una tecla. Esta característica de solicitud de interrupción cuando se detecta un cambio junto con las resistencias “pullup” configurables para estas cuatro patitas, las hacen ideales para Descripción Detallada del PIC16F877 el manejo de teclados en dispositivos portátiles que requieren “dormirse” durante largos ratos para economizar baterías y “despertarse” cuando una tecla es presionada. En las siguientes figuras se muestra el alambrado interno de las patitas del puerto B.

3.4.1.3. Puerto C (Portc)

El puerto C consta de 8 líneas bidireccionales. Trabaja en forma similar a los dos puertos anteriores y tiene asociados los registros:

Registro PORTC (07H). Es el registro de datos cuyos 8 bits RC7, RC6,..., RC0 reflejan directamente el valor lógico de las líneas físicas del puerto C.

Registro TRISC (87H). Registro de control de dirección de las líneas del puerto C. Poniendo un 1 en un bit del registro TRISC se configura la línea correspondiente como entrada y poniendo un 0 se configura la línea correspondiente como salida.

3.4.1.4. Puertos D y E.

El Puerto D es un puerto de 8 líneas configurables como entradas o salidas mediante el registro **TRISD** (88H) y cuyas líneas pueden ser accedidas mediante el registro **PORTD** (08H). Cuando se configuran como entradas éstas poseen un disparador Schmitt.

El Puerto E sólo posee 3 líneas configurables como entradas o salidas mediante el los 3 bits menos significativos del registro **TRISE** (89H). Sus líneas pueden ser accedidas mediante los 3 bits menos significativos del registro **PORTE** (09H).

El Puerto D puede configurarse para trabajar simultáneamente con sus 8 bits como un puerto esclavo (Parallel Slave Port) de comunicación paralela bidireccional con líneas de protocolo proporcionadas por las tres líneas del Puerto E, para ello se deberá activar el bit PSPMODE (TRISE<4>).

3.4.1.5. Puerto Serie Usart.

La USART (Universal Synchronous Asynchronous Receiver Transmitter) es uno de los dos periféricos contenidos en el PIC que le permiten realizar comunicación en serie. El otro es el MSSP (Master Synchronous Serial Port).

La USART, también conocida como SCI (Serial Communications Interface) puede configurarse como una unidad de comunicación en serie para la transmisión de datos asíncrona con dispositivos tales como terminales de computadora.

3.4.2. Configuración de la Usart del Pic 16F877A.

La USART del PIC puede ser configurada para operar en tres modos:

- ✓ Modo Asíncrono (full duplex (transmisión y recepción simultáneas)).
- ✓ Modo Síncrono – Maestro (half dúplex).
- ✓ Modo Síncrono – Esclavo (half dúplex).

En este modo Asíncrono la USART usa un formato estándar NRZ asíncrono, el cual para la sincronización usa: 1 bit de inicio (I), 8 o 9 bits de datos y 1 bit de paro (P). Mientras no se están transmitiendo datos la USART envía continuamente un bit de marca. El modo asíncrono se selecciona limpiando el bit SYNC del registro TXSTA (98H). Cada dato es transmitido y recibido comenzando por el LSB (bits menos significativos)

3.4.3. Proceso de Conversión Analógico/Digital.

En la Figura III.39., se muestra el diagrama de tiempo y los eventos que tienen lugar durante el proceso de una conversión analógico / digital.

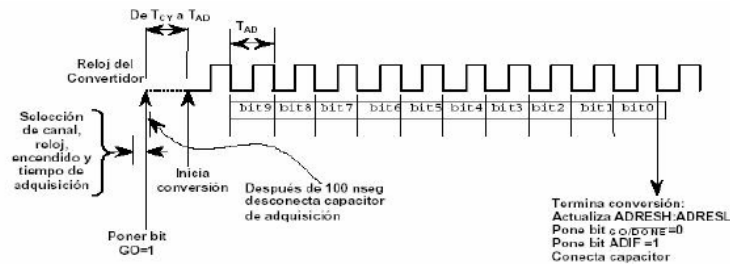


Figura III. 39. Conversión Analógico/Digital del PIC 16F877A.

3.4.4. Registros de Control del Convertidor A/D del Pic 16F877A.

Los registros de control relacionados con la operación del convertidor, son los siguientes Registro ADCON0 (1Fh) y Registro ADCON1 (9Fh). En la figura III.40 se muestra la estructura de los Registros de Control del convertidor A/D.

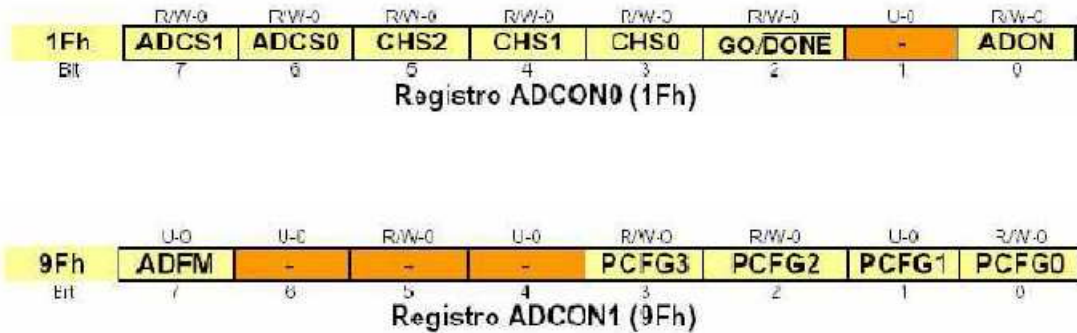


Figura III. 40. Registros de Control del convertidor A/D del PIC 16F877A.

3.4.5. Los Registros de Resultados del Convertidor A/D del Pic 16f877a

El par de registros ADRESH: ADRESL (1Eh:9Eh) son cargados con el dato (D) resultante de una conversión analógico / digital al terminar ésta. Cada uno de estos registros es de 8 bits, por lo tanto, juntos pueden guardar hasta 16 bits. Sin embargo, como el resultado D es de 10 bits, el módulo de conversión A/D permite justificarlo (alinearlo) en la parte izquierda o derecha de los 16 bits disponibles, para elegir alguna de las dos opciones se usa el bit ADFM (ADCON1<7>) como se muestra en la siguiente figura III.41.

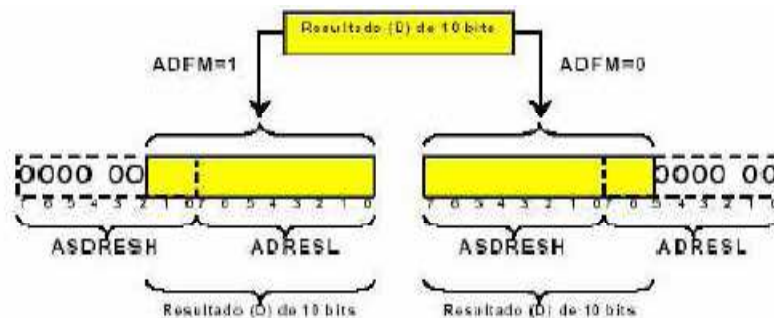


Figura III. 41. Registros de Resultados del Convertidor A/D

3.4.6. El Módulo Temporizador del Pic 16F877A.

Este módulo consta de tres contadores/temporizadores denominados TMR0, TMR1 y TMR2, dos módulos CCP (Captura, Comparación y PWM (Modulación de ancho de pulso) denominados CCP1 y CCP2.

Módulo	Características
TMR0	<ul style="list-style-type: none"> - TMR0 es un Contador/Temporizador de 8 bits. - Leíble y escribible. - Reloj interno o externo. - Selección de flanco activo en el reloj externo. - Solicitud de interrupción opcional en el desbordamiento (de FFh a 00h)
TMR1	<ul style="list-style-type: none"> - TMR1 es un Contador/Temporizador de 16 bits - Leíble y escribible - Reloj interno o externo - Solicitud de interrupción opcional en el desbordamiento - Re inicialización opcional desde los módulos CCP
TMR2	<ul style="list-style-type: none"> - TMR2 es un Contador/Temporizador de 8 bits - Dispone de un registro de periodo de 8 bits (PR2) - Leíble y escribible - Solicitud de interrupción opcional al coincidir TMR2 y PR2 - Posibilidad de generar impulsos al módulo SSP (puerto serie síncrono)
CCP1 y CCP2	<ul style="list-style-type: none"> - Modo de captura - Modo de comparación - Modo PWM (modulación de ancho de pulso)

Tabla II. XII. Modulo del Temporizador del PIC 16F877A.

CAPÍTULO IV

DISEÑO Y CONSTRUCCIÓN

En este capítulo se da a conocer el conjunto de especificaciones que integran el sistema, permitiendo su implementación.

4.1. ESQUEMA GENERAL DE FUNCIONAMIENTO

El diseño del robot empezó con la formulación de una idea central y de ahí la conformación de un diagrama de bloques que sintetiza el funcionamiento y desarrollo del Hardware.

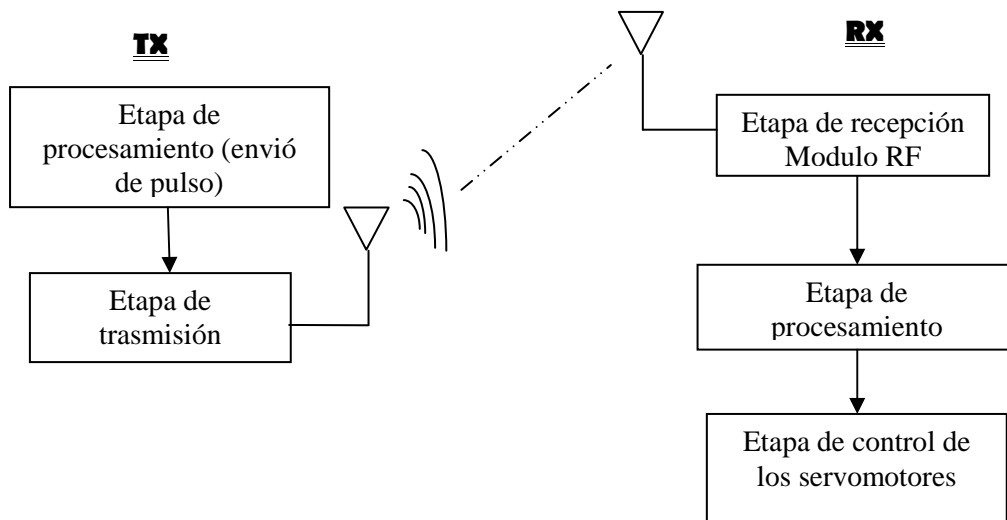


Figura IV. 42. Diagrama General del Sistema

4.2. DISEÑO Y DESCRIPCIÓN DE LOS BLOQUES QUE COMPONENTEN EL SISTEMA DEL ROBOT ÁPODO

El sistema de control y manejo del robot ha sido dividido en dos partes principales: la parte de transmisión y recepción. A continuación se muestra información acerca de cada uno de los bloques que componen el sistema:

4.2.1. Etapa de procesamiento del Transmisor

Esta etapa es una de las más importantes en el sistema, en ella recae la lógica necesaria para la ejecución de las instrucciones.

4.2.1.1. Algoritmo del Transmisor

El transmisor, se realizó con el seguimiento de procesos que llevaron a la investigación de estas tecnologías necesarias para la implementación. El PIC16f628A tiene un cristal RC interno que trabaja a 4 Mhz, a 5V, inicio con la configuración del PIC16f628A, se encera el sistema, luego se lee la señal que viene desde el computador a través de comunicación serial poniéndola en una variable que verifica la validez de la opción. Si la señal no está dentro de las opciones pedirá nuevo ingreso de la misma y si es correcta procede a codificarla y enviarla al modulo de transmisión. Por último este envía la señal al Receptor.

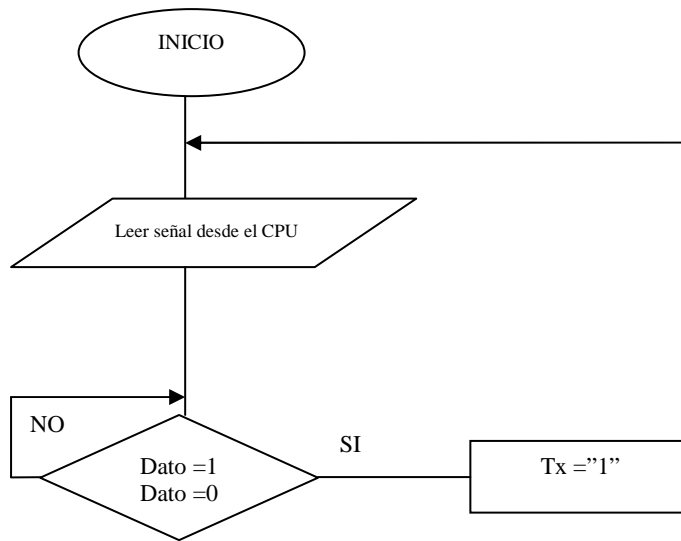


Figura IV. 43. Diagrama de flujo para la transmisión de información

4.2.1.2. Circuito Transmisor Inalámbrico en Isis/Proteus

A continuación se muestra en ISIS/PROTEUS el diseño y construcción del Transmisor para un robot ápedo.

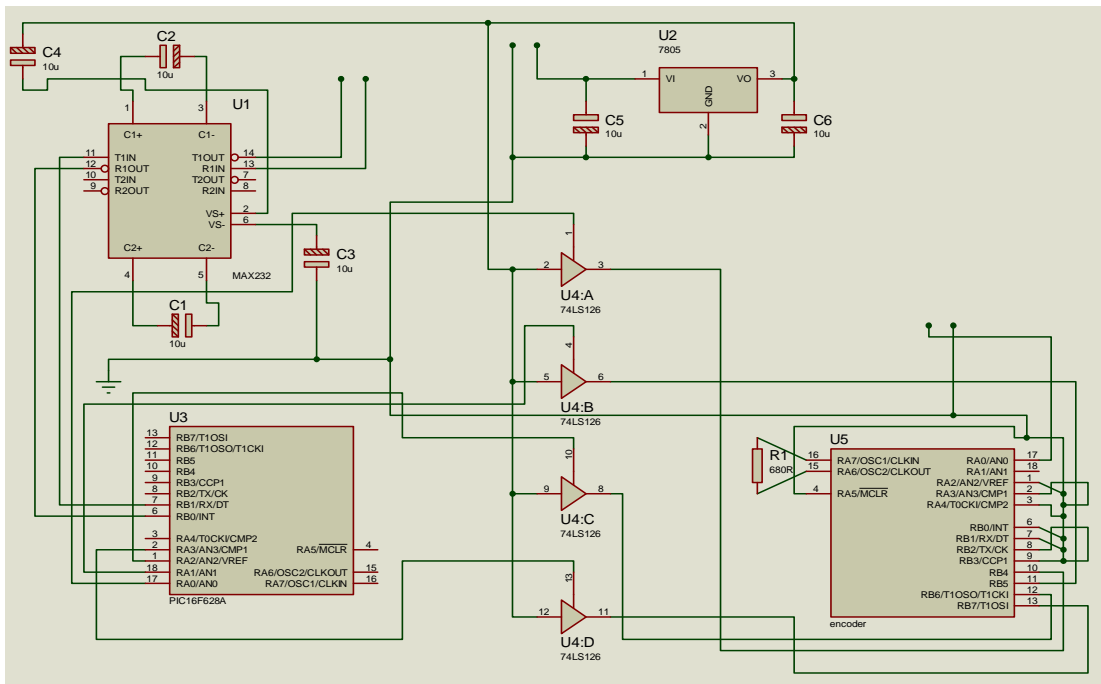


Figura IV. 44. Diagrama esquemático del transmisor

Para la transmisión de la información se ha diseñado un circuito con el PIC 16F628A, como se puede observar en las figuras IV.36 y IV. 37, no se colocó un oscilador puesto que este micro tiene un oscilador interno RC, que oscila a 4Mhz. Además se establece una comunicación serial entre el CPU y el pic a través del pin 7, el circuito integrado 74ls126 permanecerá siempre activo (conectado a Vcc) a través de los pines 2, 5, 9, 12 este a su vez recibirá el dato por los pines 1, 4, 10 y 13. Estos datos son recibidos por el ENCODER HT-12E para luego ser codificados y enviados al módulo de transmisión. La tecnología que se maneja es Radio Frecuencia. Véase Figura IV.36.

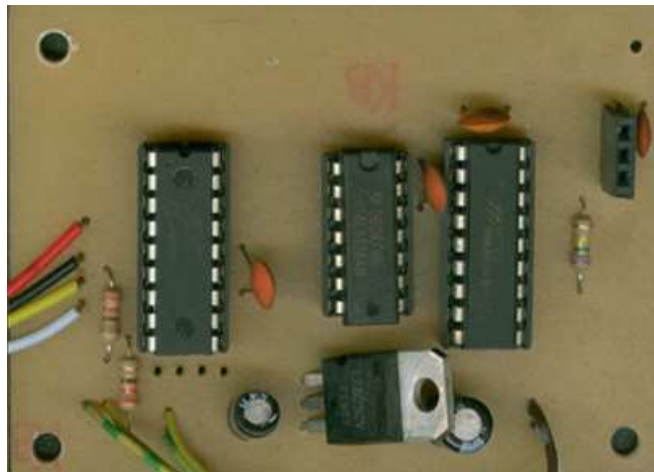


Figura IV. 45. Diseño del transmisor.

4.2.2. Etapa de Transmisión y Recepción

Consta de un transmisor/receptor de radiofrecuencia TWS-434 y RWS-434 se utiliza como el transceptor de datos inalámbricos de corto intervalos de tiempo, son sencillos de utilizar, extremadamente pequeños, ofrecen mejor estabilidad y fiabilidad, utilizan modulación del tipo ASK. Tiene una potencia de salida de hasta 8mW a 433.92MHz, alcanzando distancias de aproximadamente 140 metros en espacios abiertos y de 60 metros en espacios internos donde se tengan obstáculos. Acepta tanto señales lineales como digitales de entrada, puede operar con una tensión que va desde 1.5 Volts a 12 Volts-DC. Se encarga de enviar y recibir la información a una velocidad de 4800 bps aproximadamente, utilizando el LM7805 para su respectiva

alimentación. El transmisor de nuestro diseño conectando con el codificador HT-12E de Holtek, es un integrado utilizado en controles remotos de 4 bits y tiene 8 bits de direcciones.

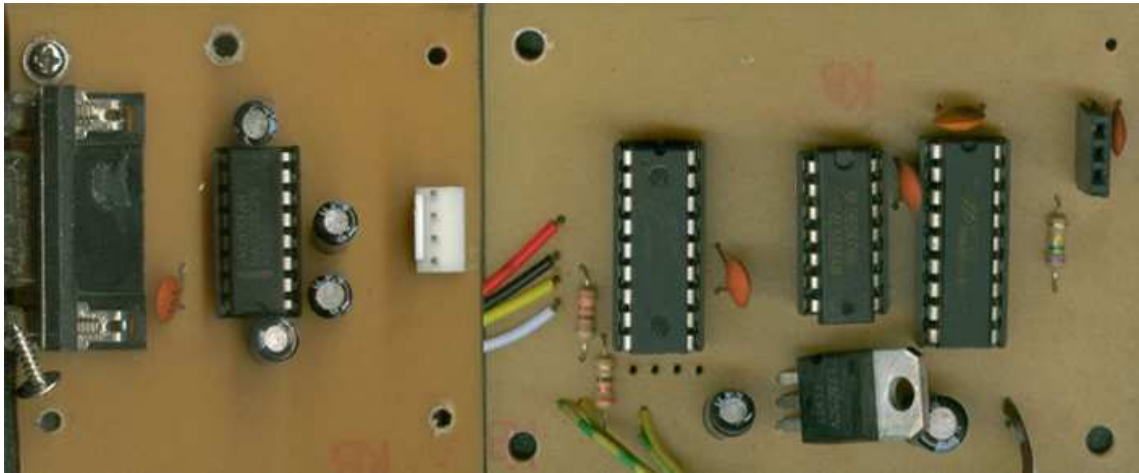


Figura IV. 46. Transmisor

4.2.3. Software De Transmisión De Datos

Para la transmisión de datos se ha creado una interfaz utilizando la herramienta Visual Basic..

La Figura IV.47 muestra la presentación inicial del programa.



Figura IV. 47. Pantalla Principal

Si presionamos directamente en el botón **ON** nos emitirá un mensaje de advertencia, es decir que se debe ingresar primero el puerto a través del menú desplegable que ofrece. El puerto que siempre se escoge es el que aparece como última opción. Por último aceptamos, la fig. IV.48 muestra el resultado.



Figura IV. 48. Ingreso del Puerto

Una vez elegido el puerto aparecerá la confirmación de que ese puerto ha sido habilitado, luego aceptamos, los resultados en la figura IV.49.



Figura IV. 49. Puerto Habilitado

Por último en la figura IV.50 se mostrarán, las diferentes secuencias de desplazamiento programadas que podrán ser ejecutadas por nuestro robot.



Figura IV. 50. Secuencias de Desplazamiento

Los datos que son ingresados a través de esta interfaz pasan a la memoria del robot, utilizando un cable de comunicación serial. El mismo que se conecta por un lado al puerto serial de la computadora y el otro a la placa de comunicación que se ha diseñado.

La placa consta de un conector serial y del MAX232 que permite realizar la comunicación serial con la PC, estos datos se graban en la memoria EEPROM del microcontrolador de la tarjeta de control general.

La forma de conexión se muestra en la Figura IV.51.

PLACA DE CX SERIAL



Figura IV. 51. Esquema de Conexión de la Placa de Comunicación Serial

4.2.4. Etapa de Procesamiento del Receptor

El Receptor RWS-434 es un módulo que opera a 433.92MHz, y tiene una sensibilidad de 3uV. Este opera con una alimentación entre 4.5 y 5.5 Volts-DC y tiene tanto salida lineal como digital, además contiene un capacitor variable para el ajuste de la frecuencia de recepción utilizando un destornillador plástico.

Construiremos el receptor conectando la unidad receptora con el decodificador HT-12D de Holtek.

Esta etapa es una de las más importantes en el sistema, en ella recae la lógica necesaria para la decodificación y ejecución de las instrucciones.

4.2.4.1. Algoritmo del Receptor

La recepción de información es un circuito diseñado con el PIC 16F877A para lo cual se procede a la configuración de los parámetros como: TRISA, TRISB, ADCON1, luego se encera el sistema, se lee el carácter enviado por el transmisor, luego se realiza el respectivo proceso para ejecutar las diferentes secuencias. Véase figura IV.52.

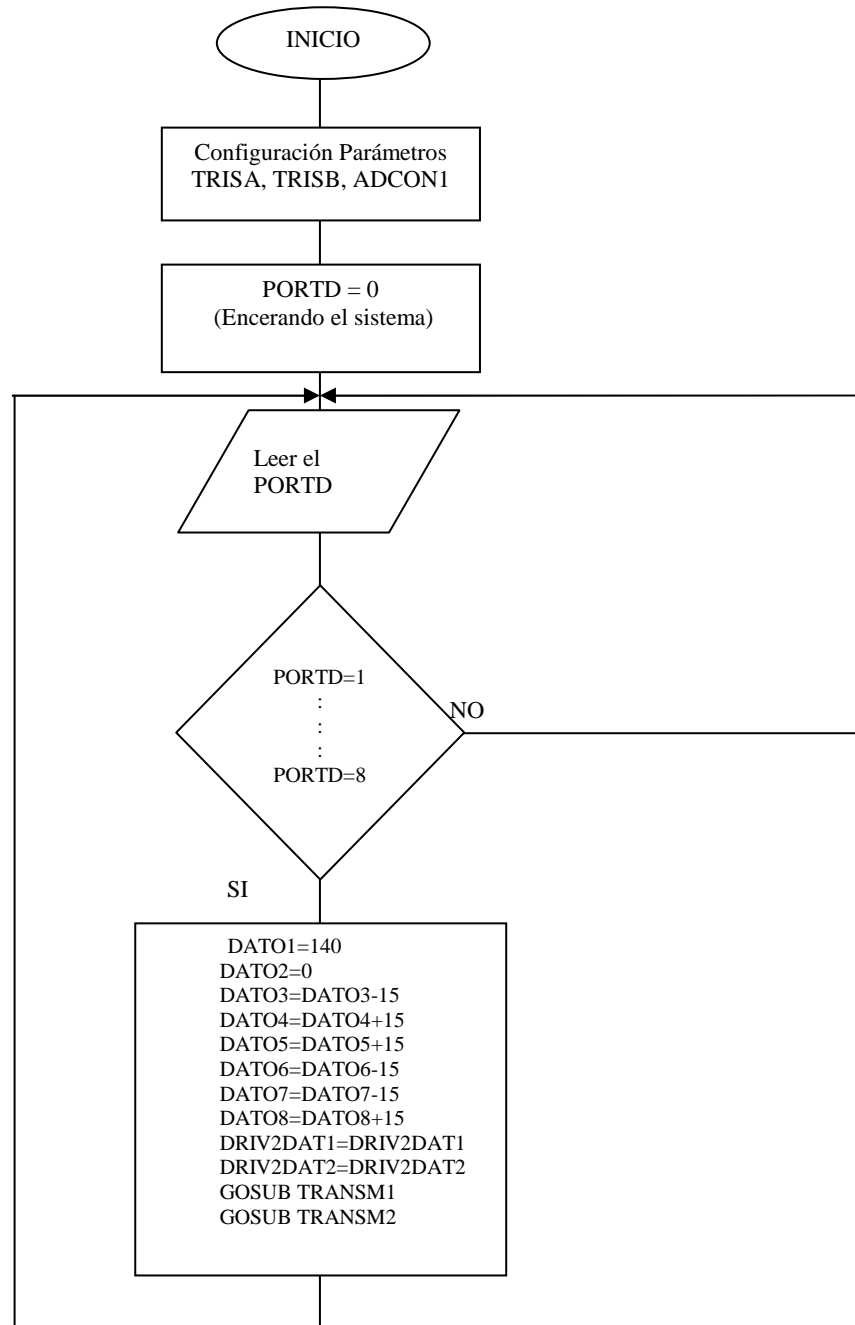


Figura IV. 52. Diagrama de flujo para la recepción de información

4.2.4.2. Circuito Receptor Inalámbrico En Isis/Proteus

En base al estudio e investigación de las tecnologías necesarias para el diseño y construcción del circuito receptor del robot apodo, se visualiza a continuación su esquema.

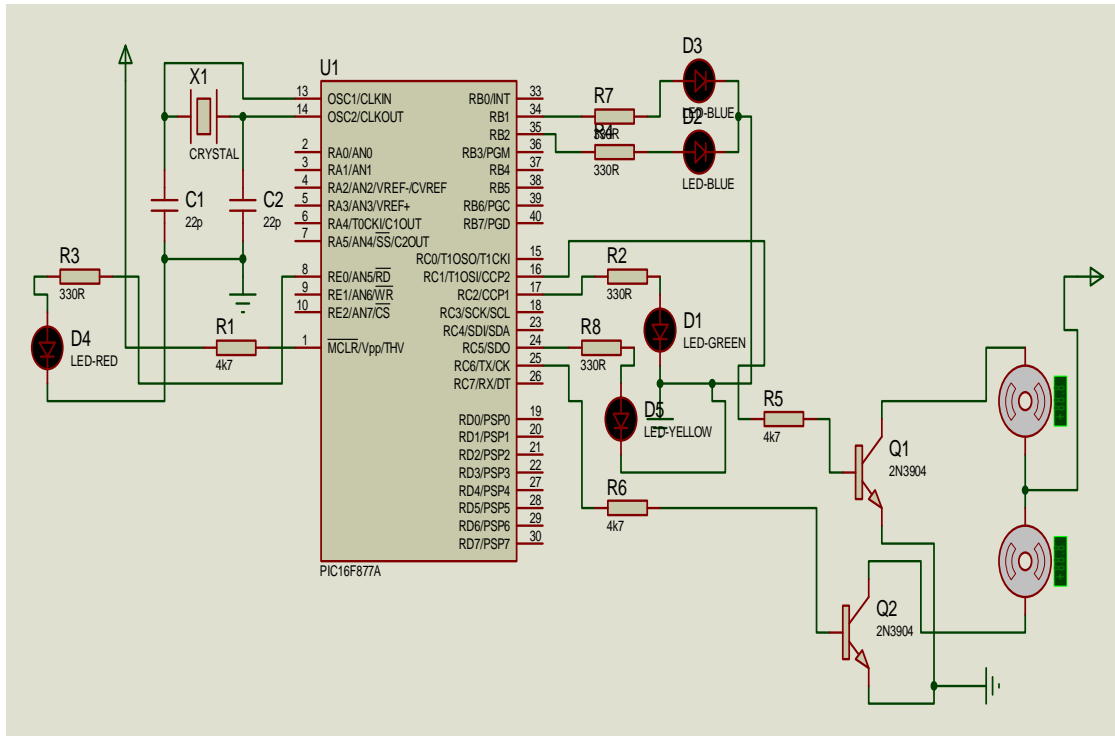


Figura IV. 53. Diagrama esquemático del receptor

La recepción de información se la hace a través del módulo Receptor RWS-434, el cual está conectada al ENCODER HT-12D para decodificar la señal y enviarla al PIC 16F877A a través del PORTD respectivamente. El micro procesa dicha información y la envía por el PORTC.1 serialmente al driver de potencia el cual nos permite ejecutar los diferentes movimientos los cuales son reflejados por los servomotores. Véase Figura IV.54.

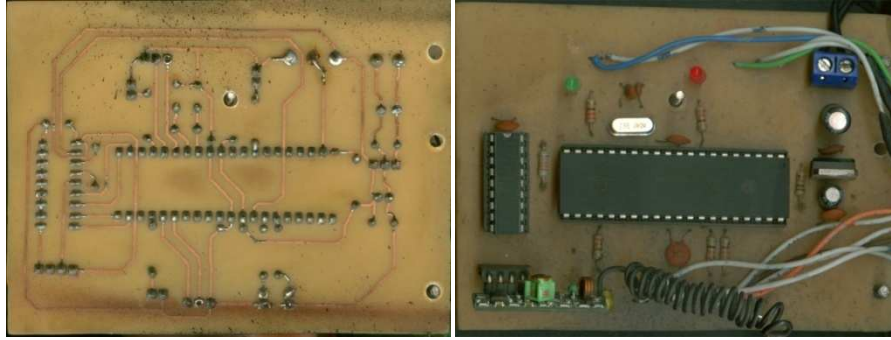


Figura IV. 54. Diseño del Receptor

4.2.4.3. Etapa de Ejecución

La etapa de ejecución básicamente se refiere al conjunto de servomotores los mismos que se eligieron por sus relevantes características. Siendo su principal ventaja su potencia.

Esta etapa se realizó utilizando servomotores alineados uno a continuación de otro, los mismos que van a formar cada una de las vertebras del robot, estos serán controlados por el receptor en la etapa de procesamiento realizando las sentencias preestablecidas para este objetivo.



Figura IV. 55. Diagrama de la etapa de ejecución

4.3. ESTRUCTURA Y DISEÑO DEL ROBOT

El diseño del robot se lo ha efectuado en el programa Solidworks. La implementación se lo ha realizado en madera MDF un material resistente y liviano. Este material ha sido la base para la estructura principal del diseño. Además se adicionó un eje falso a cada servomotor facilitando la movilidad del robot. Las partes del mismo se encuentran detalladas en las figuras IV.56, 57,58.

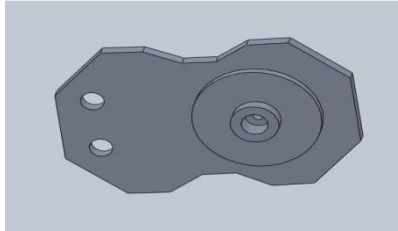


Figura IV. 56. Estructura Lateral

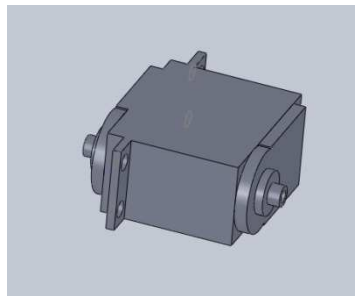


Figura IV. 57. Estructura del Eje Falso

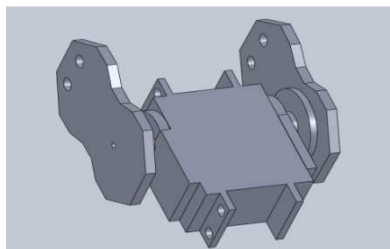


Figura IV. 58. Estructura de cada Articulación

La figura IV.59, muestra cada una de las dimensiones de la pieza principal de la estructura del robot, misma que se detalla a continuación, de esta forma podemos observar cómo fue construido el robot, las medidas son:

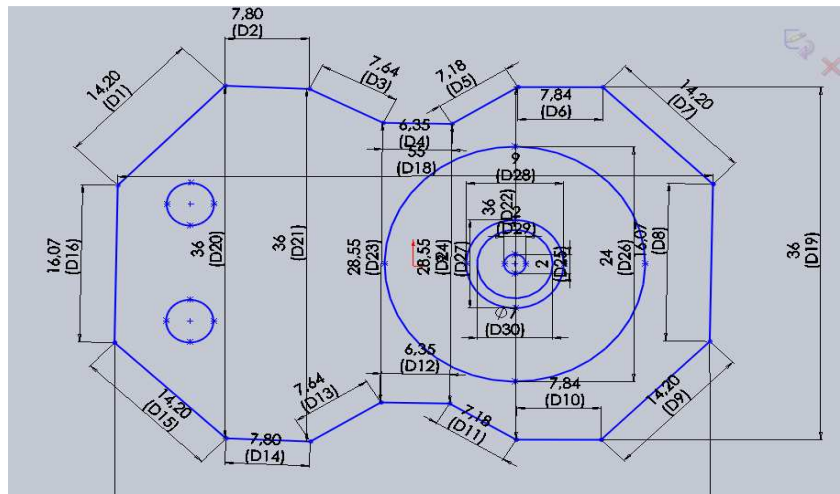


Figura IV. 59. Dimensiones de la Pieza Principal

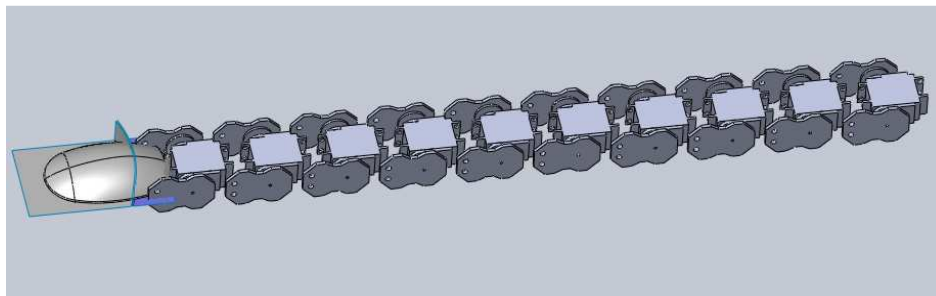


Figura IV. 60. Estructura del Robot

4.3.1. Diseño del programa.

El software fue realizado con ayuda del programa PicBasic de MicroCode Studio; PicBasic Pro es un programa editor de texto diseñado exclusivamente para facilitar la programación de los microcontroladores. Ver Anexo B.

CAPÍTULO V

ANÁLISIS Y RESULTADOS

5.1. FUNCIONAMIENTO GENERAL

5.1.1. Proceso de Comunicación

El sistema de comunicación inalámbrico utilizado genera una potencia de salida de hasta 8mW a 433.92MHz, alcanzando distancias de aproximadamente 140 metros en espacios abiertos y de 60 metros en espacios internos donde se tengan obstáculos

La distancia de transmisión del nuestro sistema con paredes de 30 centímetros de espesor, es de 10m en línea recta entre el transmisor y el receptor, teniendo como máximo 3 paredes de por medio.

Por lo tanto al probar el sistema se pudo observar que la transmisión a corta o larga distancia, con o sin obstáculos se la hace sin dificultad, llegando a obtener una eficiencia del 100%.

5.1.2. Sistema de Locomoción

Esta etapa se considera como la más importante ya que depende de su estructura ósea para permitir la locomoción del robot. Es decir se efectuaran los movimientos de una forma controlada y coordinada a la vez.

Por ello debido al diseño con el que ha sido construido es capaz de desenvolverse casi sin ningún problema en las diferentes superficies planas en las cuales ha sido probado, obteniendo una eficiencia del 90%.

5.1.3. Algoritmos de Control

El robot consta de 5 algoritmos de control los cuales permiten realizar diferentes secuencias de desplazamiento. Para cada una de las secuencias de desplazamiento se han ejecutado sin mayor complicación cada una de las pruebas, logrando así una eficiencia del 85%.

5.2. MOTORES UTILIZADOS

Para la elaboración de la estructura ósea se realizó el estudio de los diferentes motores de corriente continua que existen en el mercado, llegando a la conclusión: que los servomotores de doble eje son los más adecuados para la construcción de este robot. Debido a su forma son más estables y eficientes para la realización de este tipo de proyectos.

La estructura ósea del robot está compuesta por 10 servomotores de la empresa HITEC modelo HSR-311 dándole 10 grados de libertad en su movilidad.

A nuestro servomotor se le adicionó un eje falso proporcionándole al robot mayor estabilidad así como la fácil construcción de la estructura misma.



Figura V. 61. Servomotor HITEC HS-311

5.3. SENSOR INFRARROJO EMPLEADO

Para la detección de obstáculos se utilizó un sensor infrarrojo medidor de distancia siendo el más apropiado el SHARP GP2D120X, debido a su existencia en el mercado ecuatoriano y a las características que posee para la detección de objetos.

Este sensor es capaz de detectar obstáculos desde una distancia de 10 a 80 cm, dependiendo del rango de sensado que se programe para la detección. En este robot se colocó el sensor en la cabeza. Está programado para que detecte obstáculos a una distancia de 15 cm y una vez detectado el obstáculo se ejecutará la rutina preestablecida.



Figura V. 62. Sensor SHARP GP2D120X

5.4. ALIMENTACIÓN PARA EL SISTEMA

Para el funcionamiento del robot se utilizaron: una batería CC de 9 V que alimenta el circuito principal y una batería prefabricada CC de 10 V que alimenta los 10 servomotores.

El circuito principal trabaja a un voltaje de 5 Vcc, por lo que se tuvo que adaptar un regulador de voltaje para reducirlo de 9 a 5V.

Los servomotores trabajan a 6 Vcc por lo que se utilizó una batería a 10 Vcc con un consumo de corriente de 2200mA, dicha batería se adaptó para estos servomotores, con lo que se evita problemas en la ejecución de las acciones, siendo suficiente para mantener alimentados los 10 servomotores, el tiempo de uso varía con la complejidad de los movimientos.

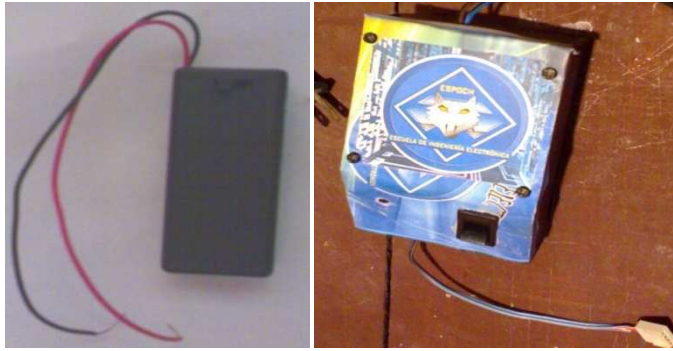


Figura V. 63. Fuentes de alimentación

5.5. ENTORNO DE TRABAJO

Se obtuvo un robot ápodo de tipo gusano, que mide 93 cm de largo, 9 cm de alto y 8 cm de ancho, el cual puede interactuar con el usuario a través del modulo de transmisión de radio frecuencia.

En el momento de arrancar, el robot espera las ordenes de su usuario, al emitirle una orden, la recepta a través del modulo de recepción, este es procesado y lo transmite mediante comunicación serial al PIC16F877A, el cual está dispuesto exclusivamente para la comparación de comandos recibidos por el modulo de transmisión, según el manejo de la aplicación en el computador el robot realizara su rutina, estos movimientos programados, hacen que el robot sea capaz de levantarse, desplazarse por una superficie plana evitando obstáculos.

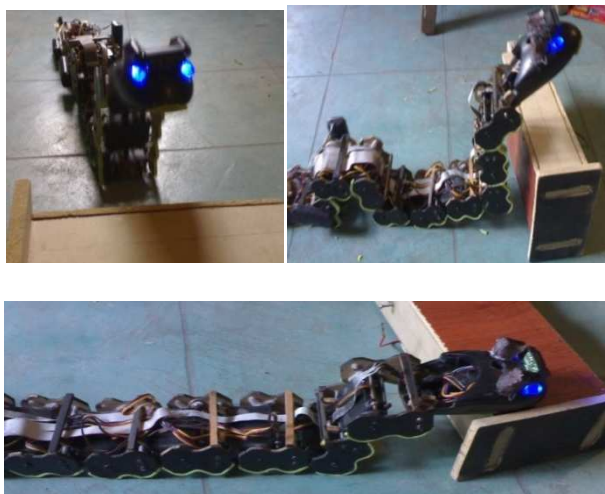


Figura V. 64. Detección de obstáculos

5.6. ANÁLISIS ECONÓMICO

Todo proyecto o actividad que se realice conlleva un gasto económico. Es por ello que a continuación se detalla cada uno de los elementos utilizados con sus respectivos precios.

5.6.1. Costo de Materiales

En la tabla V.13 se detalla todo lo que ha sido utilizado para que nuestro robot ápododo pueda funcionar.

PARTE MECÁNICA			
Cant.	DESCRIPCIÓN	V. Unitario	V. Total
1	carcasa	5,00	5,00
10	Servos	17,00	170,00
HARDWARE			
1	Sharp	23,00	23,00
2	Baterías		58,00
1	Modulo de Radiofrecuencia		15,00
	Resistencias		2,00
	Capacitores		2,00
2	Encoder HT12E y HT12D	7,00	14,00
1	PIC16F628A	4,00	4,00
1	PIC16F877A	8,00	8,00
2	Driver de Potencia	45,00	90,00
10	Ejes Falsos	5,00	50,00
1	Cable Usb-Serial	15,00	15,00
VARIOS			
1	Vaquelita de Fibra de Vidrio		9,00
2	Brocas - topes - tornillos- cables - ácido		22,00
	Molex - sócalos		18,00
	Madera MDF		5,00
TOTAL :			510,00

Tabla II. XIII.. Materiales empleados en el robot y sus costos

5.6.2. Presupuesto General

Para el desarrollo de este proyecto se necesitó de una inversión en tiempo y dinero. Para lo cual he realizado un resumen que se muestra en la tabla V.14.

TIPO DE RECURSO	TIPO	DESCRIPCIÓN	\$ MONTO
Recursos humanos	Personal directo	Autor: Mayra Alexandra Ayala Vimos	400,00
	Personal indirecto	Director y Miembros de la Tesis.	
Recursos materiales	Computadora como equipo remoto.	Computadora básica, Corel Duo 1,8 Ghz, DD 160 Gb, Biostar, 1Gb Ram.	600,00
Recursos tecnológicos	Software	Programa de adquisición de datos en Visual Basic	50,00
	Hardware	Elaboración del circuito impreso Transmisor-Receptor	310,00
TOTAL			1360,00

Tabla II. XIV. Tabla Presupuesto General

CONCLUSIONES

1. El robot ápedo tipo gusano puede desenvolverse casi sin mayor problema en cualquier tipo de ambiente externo, ya que supera de forma correcta cada uno de los obstáculos que se pueden presentar.
2. El sistema de censado detecta un obstáculo a una distancia de 10cm medido desde el robot al obstáculo.
3. La distancia de transmisión del sistema con paredes de 30 centímetros de espesor, es de diez metros en línea recta entre el transmisor y el receptor, teniendo como máximo 3 paredes de por medio.
4. El sistema de alimentación del robot, principalmente el de los servomotores tiene una duración de 4 horas como máximo en trabajo continuo, o dicha duración dependerá de los tiempos de descanso que se le asignen al robot entre una y otra secuencia a ejecutar.
5. En general, la tecnología inalámbrica utiliza ondas de radiofrecuencia de baja potencia y una banda específica, de uso libre para transmitir, entre dispositivos.
6. El programa MicroCode Studio es muy práctico en lo que tiene que ver con la programación de Microcontroladores, puesto que es de fácil manejo y tiene una interfaz muy potente. De ahí nuestro interés por utilizarlo.

RECOMENDACIONES

1. En el área de manejo se debe evitar que existan objetos demasiado grandes que no pueden ser sobrepasados por el robot.
2. La superficie del área de actuación debe ser en lo posible plana para tener mejor desempeño en el desplazamiento.
3. El receptor debe encontrarse a un máximo de diez metros para que pueda captar la señal que ha sido enviada por el computador o el mando a distancia sin problema alguno, en caso de existir hasta 3 paredes de por medio.
4. En el ambiente en el que se vaya a realizar las actividades no debe existir otro modulo de transmisión, ya que el robot puede detectar este tipo de frecuencias y las puede tomar como ordenes enviadas desde el transmisor principal.
5. Se debe tener cuidado en las conexiones del sistema de alimentación de las tarjetas electrónicas y los servomotores ya que una mala polarización de los mismos pueden dañar los elementos del robot.
6. El proceso de construcción de este tipo de proyectos debe realizarse a nivel modular de esta manera optimizaremos tiempo en cuanto a ubicación de daños se refiere.

RESUMEN

Se diseñó e implementó un robot ápodo tipo gusano, con la finalidad de brindar al robot mayor flexibilidad en su desplazamiento al momento de encontrar un obstáculo es decir que no lo evada sino lo sobrepase.

La estructura ósea del robot se construyó en madera MDF, con 10 servomotores HSR-311, un sensor infrarrojo SHARP GP2D120X para la detección de objetos. Módulos de radiofrecuencia TWS-434 y RWS-434 para la comunicación inalámbrica, un computador personal, dispositivos electrónicos, etc. Se empleo el método experimental para modelar el robot además de técnicas de lectura investigativa, técnicas de diseño utilizando el programa Solidworks y técnicas de programación empleándose Visual Basic 6.0 y Microcode Basic.

Finalmente se obtuvo un robot en forma de gusano alcanzando una longitud de 93cm de largo, 8cm de ancho, altura de 9cm a través de la concatenación de 10 servomotores que permiten la locomoción del robot en una forma controlada y coordinada a la vez. Cuenta con algoritmos de control que me permiten ejecutar cada una de las secuencias de desplazamiento reflejadas en los servomotores. La distancia de transmisión con paredes de 30 cm de espesor, es de 10m en línea recta entre el transmisor y el receptor, teniendo como máximo 3 paredes de por medio.

Por lo tanto al probar el robot construido podemos decir que puede ser utilizado, con un rendimiento aceptable y óptimo del 90% en el área de trabajo. Llegando a cumplir con el objetivo propuesto.

Se recomienda tener en cuenta la sincronización de cada uno de los servomotores al momento de ejecutar las secuencias de desplazamiento.

ABSTRACT

Designed and implemented a worm legless robot, in order to give the robot more flexibility in moving the time to find an obstacle that is not evade but exceed it.

The bone structure of the robot was built in MDF, with 10 servo HSR-311, an infrared sensor to detect SHARP GP2D120X objects. RF Modules TWS-434 and RWS-434 for wireless communication, a personal computer, electronic devices, etc.. Experimental method was used to model the robot reading skills in addition to research, design techniques using Solidworks software and programming techniques using Visual Basic 6.0 and Basic Microcode.

Finally got a worm-shaped robot reaching a length of 93cm long, 8cm wide, 9cm high by the concatenation of 10 actuators that allow the robot locomotion in a controlled and coordinated at a time. Has control algorithms that allow me to perform each movement sequences reflected in the servomotors. The transmission distance with walls 30 cm thick, is 10m in a straight line between the transmitter and receiver, with a maximum of 3 walls in between.

So to test the robot that we built can be used with acceptable and optimal performance of 90% in the work area. Getting to meet the objective.

It is recommended to consider the timing of each of the servomotors when executing movement sequences.

GLOSARIO

CA.- Es la simbología de corriente alterna, es aquella en que la que la intensidad cambia de dirección periódicamente en un conductor. Como consecuencia del cambio periódico de polaridad de la tensión aplicada en los extremos de dicho conductor.

CD.- Es la simbología de corriente directa, La corriente directa (CD) o corriente continua (CC) es aquella cuyas cargas eléctricas o electrones fluyen siempre en el mismo sentido en un circuito eléctrico cerrado, moviéndose del polo negativo hacia el polo positivo de una fuente de fuerza electromotriz, tal como ocurre en las baterías, las dinamos o en cualquier otra fuente generadora de ese tipo de corriente eléctrica.

Aplicación.- Programa que realiza una serie de funciones y con el cual trabajamos en el ordenador.

Arquitectura.- Término que se refiere al tipo de estructura hardware de la máquina y que también se aplica a la clasificación de los microprocesadores o el tipo de ranuras de expansión.

Asíncrono.- Término utilizado para especificar la ejecución de distintos procesos de forma independiente unos de los otros respecto al +'

Bit (binary digit).- Unidad básica de información representada por ceros y unos que se van sucediendo para conformar los distintos significados.

Chip.- Circuito integrado y encapsulado.

Frecuencia.- Número de veces que se repite un proceso periódico por unidad de tiempo.

Hardware.- Componentes materiales propios del computador, sean mecánicos eléctricos o electrónicos.

Interfaz.- Es el punto, el área, o la superficie a lo largo de la cual dos cosas de naturaleza distinta convergen. Por extensión, se denomina interfaz a cualquier medio que permita la interconexión de dos procesos diferenciados con un único propósito común.

Microcontrolador.- Es un circuito integrado o chip que incluye en su interior las tres unidades funcionales de una computadora: CPU, Memoria y Unidades de E/S, es decir, se trata de un computador completo en un solo circuito integrado.

Módulo.- Son procedimientos que nos permiten la realización de una tarea deseada.

Ohm.- Unidad de medición de la resistencia eléctrica, representada por la letra griega.

Programa.- Conjunto de órdenes, palabras y símbolos producto de un análisis efectuado, creado expresamente para ordenar de modo \downarrow los pasos que una computadora ha de dar para realizar un trabajo.

Resistencia.- Es la medida de cuanto se opone un circuito al paso de la corriente eléctrica a través de él.

Señal.- Variación de una corriente eléctrica u otra magnitud que se utiliza para transmitir información.

Software.- Elemento intangible (programas), sin el cual la computadora no podría funcionar.

Usart.- Es uno de los dos periféricos contenidos en el PIC que le permiten realizar comunicación en serie.

Voltios.- Unidad de medición de la diferencia de potencial o tensión eléctrica.

BIBLIOGRAFÍA

- 1.- CORRALES, S. Electrónica Práctica con Microcontroladores PIC. Quito, Gráfica, 2006. pp. 62-97.
- 2.- DOGAN, I. Programación de Microcontroladores Pic. Estados Unidos, Interamericana, 2007. pp. 150-200.
- 3.- PALLÁS, A. MICROCONTROLADORES: Fundamentos y Aplicaciones con Pic. Estados Unidos, MacGranw-Hill/Interamericana, 2007. 480 p.
- 4.- REYES, C. Aprenda Rápidamente a Programar Microcontroladores. Quito, Ayerve, 2004. 193 p.

BIBLIOGRAFIA INTERNET

- 5.- Internetworking con microprocesadores PIC
<http://www.ljcv.net/projects/pic10t/index.html>
[09/02/2008]
- 6.- Código de Ejemplo para el manejo del ENC28J60
www.microchip.com/Ethernet
[05/07/2009]
- 7.- Tutorial de programación de PICs en Ensamblador en la Asociación de Robótica y Domótica
<http://www.Asrobdompics/>
[10/01/2010]
- 8.- TodoPIC, todo en microcontroladores PIC
[http://softwarepic.50webs.com/ Compiladores C, C++, Basic, pascal, etc](http://softwarepic.50webs.com/Compiladores_C,_C++,_Basic,_pascal,_etc)
[10/01/2010]
- 9.- WikiPIC
<http://www.microchip.com/>
[10/01/2010]
- 10.- HobbyPIC, afición por los microcontroladores PIC!
<http://microspics.blogspot.com/>
[10/01/2010]

ANEXOS



PIC16F627A/628A/648A

Data Sheet

FLASH-Based

8-Bit CMOS Microcontrollers



PIC16F627A/628A/648A

18-pin FLASH-Based 8-Bit CMOS Microcontrollers

High Performance RISC CPU:

- Operating speeds from DC - 20 MHz
- Interrupt capability
- 8-level deep hardware stack
- Direct, Indirect and Relative Addressing modes
- 35 single word instructions
 - All instructions single cycle except branches

Special Microcontroller Features:

- Internal and external oscillator options
 - Precision Internal 4 MHz oscillator factory calibrated to $\pm 1\%$
 - Low Power Internal 37 kHz oscillator
 - External Oscillator support for crystals and resonators.
- Power saving SLEEP mode
- Programmable weak pull-ups on PORTB
- Multiplexed Master Clear/Input-pin
- Watchdog Timer with independent oscillator for reliable operation
- Low voltage programming
- In-Circuit Serial Programming™ (via two pins)
- Programmable code protection
- Brown-out Reset
- Power-on Reset
- Power-up Timer and Oscillator Start-up Timer
- Wide operating voltage range. (2.0 - 5.5V)
- Industrial and extended temperature range
- High Endurance FLASH/EEPROM Cell
 - 100,000 write FLASH endurance
 - 1,000,000 write EEPROM endurance
 - 100 year data retention

Low Power Features:

- Standby Current:
 - 100 nA @ 2.0V, typical
- Operating Current:
 - 12 μ A @ 32 kHz, 2.0V, typical
 - 120 μ A @ 1 MHz, 2.0V, typical
- Watchdog Timer Current
 - 1 μ A @ 2.0V, typical
- Timer1 oscillator current:
 - 1.2 μ A @ 32 kHz, 2.0V, typical
- Dual Speed Internal Oscillator:
 - Run-time selectable between 4 MHz and 37 kHz
 - 4 μ s wake-up from SLEEP, 3.0V, typical

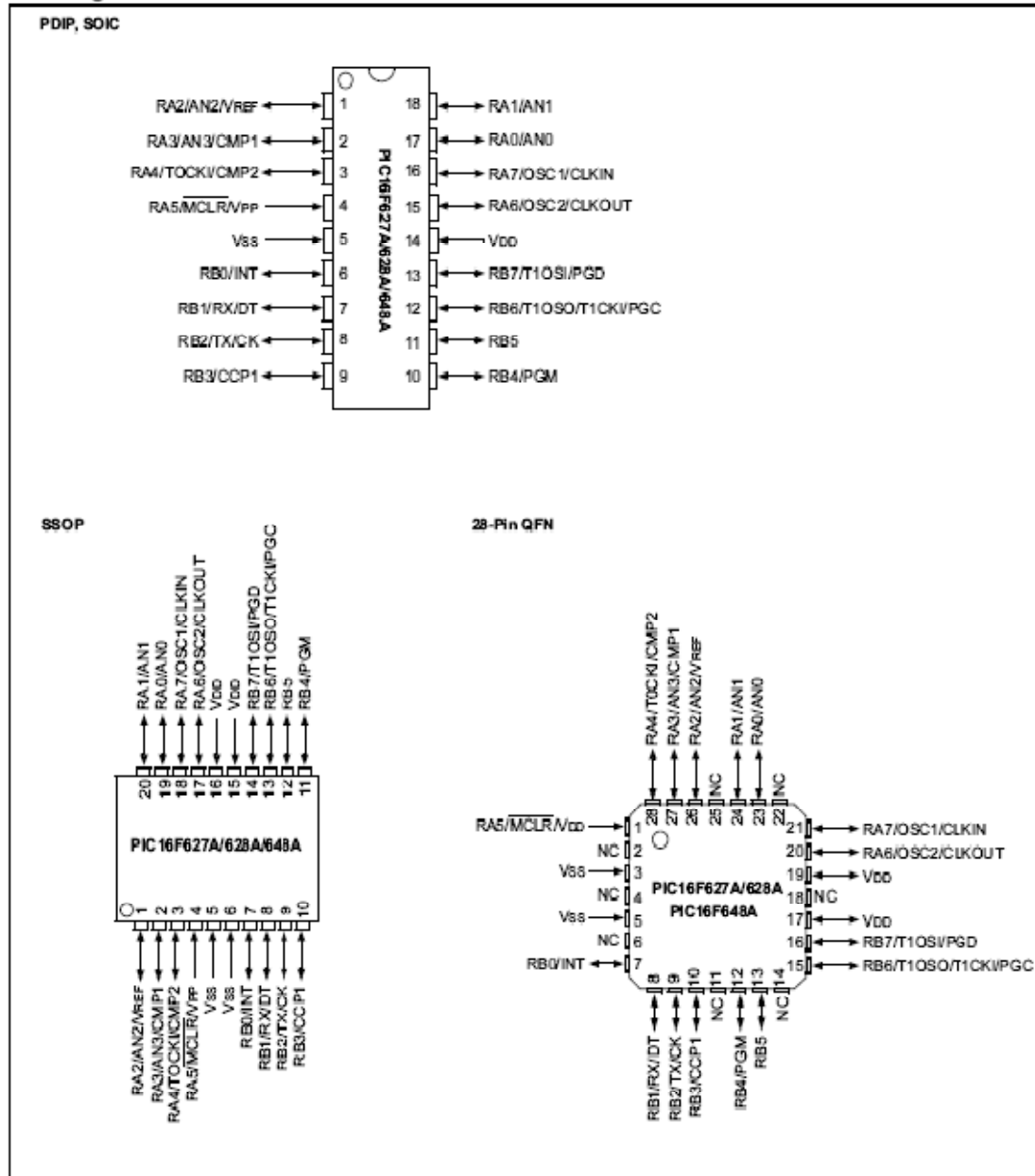
Peripheral Features:

- 16 I/O pins with individual direction control
- High current sink/source for direct LED drive
- Analog comparator module with:
 - Two analog comparators
 - Programmable on-chip voltage reference (VREF) module
 - Selectable internal or external reference
 - Comparator outputs are externally accessible
- Timer0: 8-bit timer/counter with 8-bit programmable prescaler
- Timer1: 16-bit timer/counter with external crystal/clock capability
- Timer2: 8-bit timer/counter with 8-bit period register, prescaler and postscaler
- Capture, Compare, PWM module
 - 16-bit Capture/Compare
 - 10-bit PWM
- Addressable Universal Synchronous/Asynchronous Receiver/Transmitter USART/SCI

Device	Program Memory	Data Memory		IO	CCP (PWM)	USART	Comparators	Timers 8/16-bit
	FLASH (words)	SRAM (bytes)	EEPROM (bytes)					
PIC16F627A	1024	224	128	16	1	Y	2	2/1
PIC16F628A	2048	224	128	16	1	Y	2	2/1
PIC16F648A	4096	256	256	16	1	Y	2	2/1

PIC16F627A/628A/648A

Pin Diagrams



PIC16F627A/628A/648A

1.0 GENERAL DESCRIPTION

The PIC16F627A/628A/648A are 18-Pin FLASH-based members of the versatile PIC16CXX family of low cost, high performance, CMOS, fully-static, 8-bit microcontrollers.

All PICmicro[®] microcontrollers employ an advanced RISC architecture. The PIC16F627A/628A/648A have enhanced core features, eight-level deep stack, and multiple internal and external interrupt sources. The separate instruction and data buses of the Harvard architecture allow a 14-bit wide instruction word with the separate 8-bit wide data. The two-stage instruction pipeline allows all instructions to execute in a single-cycle, except for program branches (which require two cycles). A total of 35 instructions (reduced instruction set) are available, complemented by a large register set.

PIC16F627A/628A/648A microcontrollers typically achieve a 2:1 code compression and a 4:1 speed improvement over other 8-bit microcontrollers in their class.

PIC16F627A/628A/648A devices have integrated features to reduce external components, thus reducing system cost, enhancing system reliability and reducing power consumption.

The PIC16F627A/628A/648A has 8 oscillator configurations. The single-pin RC oscillator provides a low cost solution. The LP oscillator minimizes power consumption, XT is a standard crystal, and INTOSC is a self-contained precision two-speed internal oscillator. The

HS is for High-Speed crystals. The EC mode is for an external clock source.

The SLEEP (Power-down) mode offers power savings. Users can wake-up the chip from SLEEP through several external interrupts, internal interrupts and RESETS.

A highly reliable Watchdog Timer with its own on-chip RC oscillator provides protection against software lock-up.

Table 1-1 shows the features of the PIC16F627A/628A/648A mid-range microcontroller families.

A simplified block diagram of the PIC16F627A/628A/648A is shown in Figure 3-1.

The PIC16F627A/628A/648A series fits in applications ranging from battery chargers to low power remote sensors. The FLASH technology makes customizing application programs (detection levels, pulse generation, timers, etc.) extremely fast and convenient. The small footprint packages makes this microcontroller series ideal for all applications with space limitations. Low cost, low power, high performance, ease of use and I/O flexibility make the PIC16F627A/628A/648A very versatile.

1.1 Development Support

The PIC16F627A/628A/648A family is supported by a full-featured macro assembler, a software simulator, an in-circuit emulator, a low cost in-circuit debugger, a low cost development programmer and a full-featured programmer. A Third Party "C" compiler support tool is also available.

TABLE 1-1: PIC16F627A/628A/648A FAMILY OF DEVICES

		PIC16F627A	PIC16F628A	PIC16F648A	PIC16LF627A	PIC16LF628A	PIC16LF648A
Clock	Maximum Frequency of Operation (MHz)	20	20	20	4	4	4
	FLASH Program Memory (words)	1024	2048	4096	1024	2048	4096
Memory	RAM Data Memory (bytes)	224	224	256	224	224	256
	EEPROM Data Memory (bytes)	128	128	256	128	128	256
	Timer module(s)	TMR0, TMR1, TMR2	TMR0, TMR1, TMR2	TMR0, TMR1, TMR2	TMR0, TMR1, TMR2	TMR0, TMR1, TMR2	TMR0, TMR1, TMR2
Peripherals	Comparator(s)	2	2	2	2	2	2
	Capture/Compare/PWM modules	1	1	1	1	1	1
	Serial Communications	USART	USART	USART	USART	USART	USART
	Internal Voltage Reference	Yes	Yes	Yes	Yes	Yes	Yes
Features	Interrupt Sources	10	10	10	10	10	10
	I/O Pins	16	16	16	16	16	16
	Voltage Range (Volts)	3.0-5.5	3.0-5.5	3.0-5.5	2.0-5.5	2.0-5.5	2.0-5.5
	Brown-out Reset	Yes	Yes	Yes	Yes	Yes	Yes
	Packages	18-pin DIP, SOIC, 20-pin SSOP, 28-pin QFN	18-pin DIP, SOIC, 20-pin SSOP, 28-pin QFN	18-pin DIP, SOIC, 20-pin SSOP, 28-pin QFN	18-pin DIP, SOIC, 20-pin SSOP, 28-pin QFN	18-pin DIP, SOIC, 20-pin SSOP, 28-pin QFN	18-pin DIP, SOIC, 20-pin SSOP, 28-pin QFN

All PICmicro[®] Family devices have Power-on Reset, selectable Watchdog Timer, selectable Code Protect and high I/O current capability. All PIC16F627A/628A/648A Family devices use serial programming with clock pin RB6 and data pin RB7.

PIC16F627A/628A/648A

2.0 PIC16F627A/628A/648A DEVICE VARIETIES

A variety of frequency ranges and packaging options are available. Depending on application and production requirements, the proper device option can be selected using the information in the PIC16F627A/628A/648A Product Identification System, at the end of this data sheet. When placing orders, please use this page of the data sheet to specify the correct part number.

2.1 FLASH Devices

FLASH devices can be erased and re-programmed electrically. This allows the same device to be used for prototype development, pilot programs and production.

A further advantage of the electrically erasable FLASH is that it can be erased and reprogrammed in-circuit, or by device programmers, such as Microchip's PICSTART[®] Plus, or PRO MATE[®] II programmers.

2.2 Quick-Turnaround-Production (QTP) Devices

Microchip offers a QTP Programming Service for factory production orders. This service is made available for users who chose not to program a medium to high quantity of units and whose code patterns have stabilized. The devices are standard FLASH devices but with all program locations and configuration options already programmed by the factory. Certain code and prototype verification procedures apply before production shipments are available. Please contact your Microchip Technology sales office for more details.

2.3 Serialized Quick-Turnaround- Production (SQTPSM) Devices

Microchip offers a unique programming service where a few user-defined locations in each device are programmed with different serial numbers. The serial numbers may be random, pseudo-random or sequential.

Serial programming allows each device to have a unique number, which can serve as an entry-code, password or ID number.

PIC16F627A/628A/648A

3.0 ARCHITECTURAL OVERVIEW

The high performance of the PIC16F627A/628A/648A family can be attributed to a number of architectural features commonly found in RISC microprocessors. To begin with, the PIC16F627A/628A/648A uses a Harvard architecture, in which program and data are accessed from separate memories using separate buses. This improves bandwidth over traditional von Neumann architecture where program and data are fetched from the same memory. Separating program and data memory further allows instructions to be sized differently than 8-bit wide data word. Instruction opcodes are 14-bits wide making it possible to have all single word instructions. A 14-bit wide program memory access bus fetches a 14-bit instruction in a single cycle. A two-stage pipeline overlaps fetch and execution of instructions. Consequently, all instructions (35) execute in a single-cycle (200 ns @ 20 MHz) except for program branches.

Table 3-1 lists device memory sizes (FLASH, Data and EEPROM).

TABLE 3-1: DEVICE MEMORY LIST

Device	Memory		
	FLASH Program	RAM Data	EEPROM Data
PIC16F627A	1024 x 14	224 x 8	128 x 8
PIC16F628A	2048 x 14	224 x 8	128 x 8
PIC16F648A	4096 x 14	256 x 8	256 x 8
PIC16LF627A	1024 x 14	224 x 8	128 x 8
PIC16LF628A	2048 x 14	224 x 8	128 x 8
PIC16LF648A	4096 x 14	256 x 8	256 x 8

The PIC16F627A/628A/648A can directly or indirectly address its register files or data memory. All Special Function Registers, including the program counter, are mapped in the data memory. The PIC16F627A/628A/648A have an orthogonal (symmetrical) instruction set that makes it possible to carry out any operation, on any register, using any Addressing mode. This symmetrical nature and lack of 'special optimal situations' make programming with the PIC16F627A/628A/648A simple yet efficient. In addition, the learning curve is reduced significantly.

The PIC16F627A/628A/648A devices contain an 8-bit ALU and working register. The ALU is a general purpose arithmetic unit. It performs arithmetic and Boolean functions between data in the working register and any register file.

The ALU is 8-bit wide and capable of addition, subtraction, shift and logical operations. Unless otherwise mentioned, arithmetic operations are two's complement in nature. In two-operand instructions, typically one operand is the working register (W register). The other operand is a file register or an immediate constant. In single operand instructions, the operand is either the W register or a file register.

The W register is an 8-bit working register used for ALU operations. It is not an addressable register.

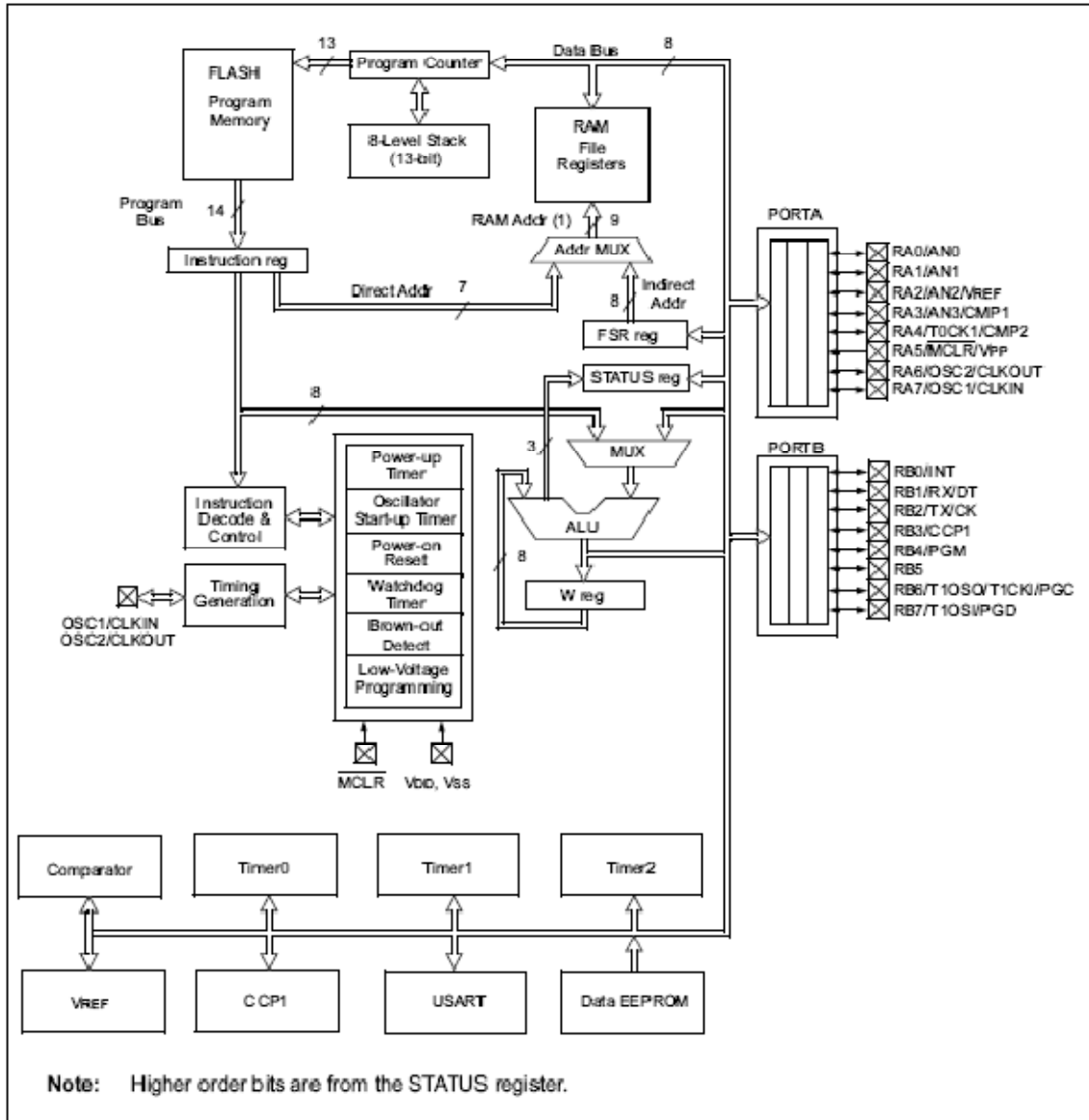
Depending on the instruction executed, the ALU may affect the values of the Carry (C), Digit Carry (DC), and Zero (Z) bits in the STATUS register. The C and DC bits operate as a Borrow and Digit Borrow out bit, respectively, bit in subtraction. See the `SUBLW` and `SUBWF` instructions for examples.

A simplified block diagram is shown in Figure 3-1, and a description of the device pins in Table 3-2.

Two types of data memory are provided on the PIC16F627A/628A/648A devices. Non-volatile EEPROM data memory is provided for long term storage of data such as calibration values, look up table data, and any other data which may require periodic updating in the field. These data are not lost when power is removed. The other data memory provided is regular RAM data memory. Regular RAM data memory is provided for temporary storage of data during normal operation. Data are lost when power is removed.

PIC16F627A/628A/648A

FIGURE 3-1: BLOCK DIAGRAM



PIC16F627A/628A/648A

TABLE 3-2: PIC16F627A/628A/648A PINOUT DESCRIPTION

Name	Function	Input Type	Output Type	Description
RA0/AN0	RA0	ST	CMOS	Bi-directional I/O port
	AN0	AN	—	Analog comparator input
RA1/AN1	RA1	ST	CMOS	Bi-directional I/O port
	AN1	AN	—	Analog comparator input
RA2/AN2/VREF	RA2	ST	CMOS	Bi-directional I/O port
	AN2	AN	—	Analog comparator input
	VREF	—	AN	VREF output
RA3/AN3/CMP1	RA3	ST	CMOS	Bi-directional I/O port
	AN3	AN	—	Analog comparator input
	CMP1	—	CMOS	Comparator 1 output
RA4/T0CKI/CMP2	RA4	ST	OD	Bi-directional I/O port
	T0CKI	ST	—	Timer0 clock input
	CMP2	—	OD	Comparator 2 output
RA5/MCLR/VPP	RA5	ST	—	Input port
	MCLR	ST	—	Master clear. When configured as MCLR, this pin is an active low RESET to the device. Voltage on MCLR/VPP must not exceed V _{DD} during normal device operation.
	VPP	—	—	Programming voltage input.
RA6/OSC2/CLKOUT	RA6	ST	CMOS	Bi-directional I/O port
	OSC2	—	XTAL	Oscillator crystal output. Connects to crystal or resonator in Crystal Oscillator mode.
	CLKCUT	—	CMOS	In RC/INTOSC mode, OSC2 pin can output CLKOUT, which has 1/4 the frequency of OSC1
RA7/OSC1/CLKIN	RA7	ST	CMOS	Bi-directional I/O port
	OSC1	XTAL	—	Oscillator crystal input
	CLKIN	ST	—	External clock source input. RC biasing pin.
RB0/INT	RB0	TTL	CMOS	Bi-directional I/O port. Can be software programmed for internal weak pull-up.
	INT	ST	—	External interrupt.
RB1/RX/DT	RB1	TTL	CMOS	Bi-directional I/O port. Can be software programmed for internal weak pull-up.
	RX	ST	—	USART receive pin
	DT	ST	CMOS	Synchronous data I/O.
RB2/TX/CK	RB2	TTL	CMOS	Bi-directional I/O port. Can be software programmed for internal weak pull-up.
	TX	—	CMOS	USART transmit pin
	CK	ST	CMOS	Synchronous clock I/O.
RB3/CCP1	RB3	TTL	CMOS	Bi-directional I/O port. Can be software programmed for internal weak pull-up.
	CCP1	ST	CMOS	Capture/Compare/PWM I/O

Legend: O = Output
 — = Not used
 TTL = TTL Input

CMOS = CMOS Output
 I = Input
 OD = Open Drain Output

P = Power
 ST = Schmitt Trigger Input
 AN = Analog

PIC16F627A/628A/648A

TABLE 3-2: PIC16F627A/628A/648A PINOUT DESCRIPTION

Name	Function	Input Type	Output Type	Description
RB4/PGM	RB4	TTL	CMOS	Bi-directional I/O port. Interrupt-on-pin change. Can be software programmed for internal weak pull-up.
	PGM	ST	—	Low voltage programming input pin. When low voltage programming is enabled, the interrupt-on-pin change and weak pull-up resistor are disabled.
RB5	RB5	TTL	CMOS	Bi-directional I/O port. Interrupt-on-pin change. Can be software programmed for internal weak pull-up.
RB6/T1OSO/T1CKI/PGC	RB6	TTL	CMOS	Bi-directional I/O port. Interrupt-on-pin change. Can be software programmed for internal weak pull-up.
	T1OSO	—	XTAL	Timer1 oscillator output.
	T1CKI	ST	—	Timer1 clock input.
	PGC	ST	—	ICSP Programming Clock.
RB7/T1OSI/PGD	RB7	TTL	CMOS	Bi-directional I/O port. Interrupt-on-pin change. Can be software programmed for internal weak pull-up.
	T1OSI	XTAL	—	Timer1 oscillator input.
	PGD	ST	CMOS	ICSP Data I/O
Vss	Vss	Power	—	Ground reference for logic and I/O pins
VDD	VDD	Power	—	Positive supply for logic and I/O pins

Legend: O = Output
 — = Not used
 TTL = TTL Input

CMOS = CMOS Output
 I = Input
 OD = Open Drain Output

P = Power
 ST = Schmitt Trigger Input
 AN = Analog

PIC16F627A/628A/648A

3.1 Clocking Scheme/Instruction Cycle

The clock input (OSC1/CLKIN/RA7 pin) is internally divided by four to generate four non-overlapping quadrature clocks namely Q1, Q2, Q3 and Q4. Internally, the program counter (PC) is incremented every Q1, the instruction is fetched from the program memory and latched into the instruction register in Q4. The instruction is decoded and executed during the following Q1 through Q4. The clocks and instruction execution flow is shown in Figure 3-2.

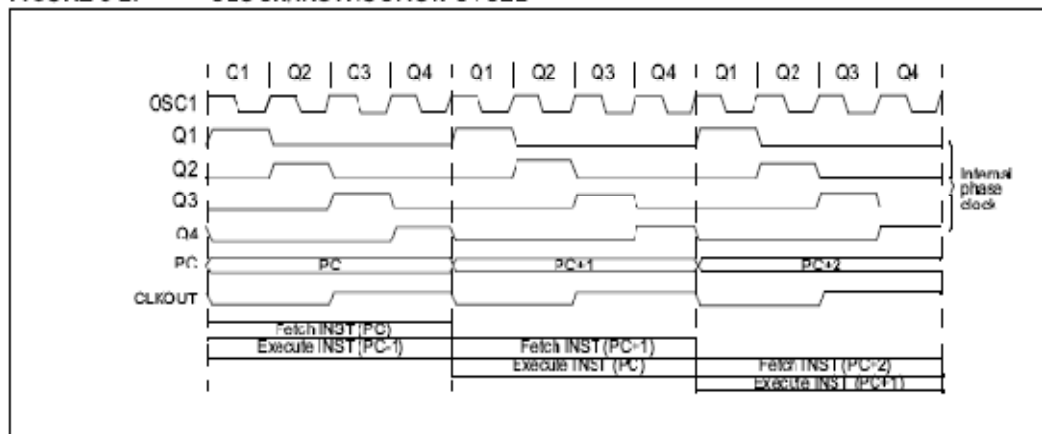
3.2 Instruction Flow/Pipelining

An instruction cycle consists of four Q cycles (Q1, Q2, Q3 and Q4). The instruction fetch and execute are pipelined such that fetch takes one instruction cycle while decode and execute takes another instruction cycle. However, due to the pipelining, each instruction effectively executes in one cycle. If an instruction causes the program counter to change (e.g., GOTO) then two cycles are required to complete the instruction (Example 3-1).

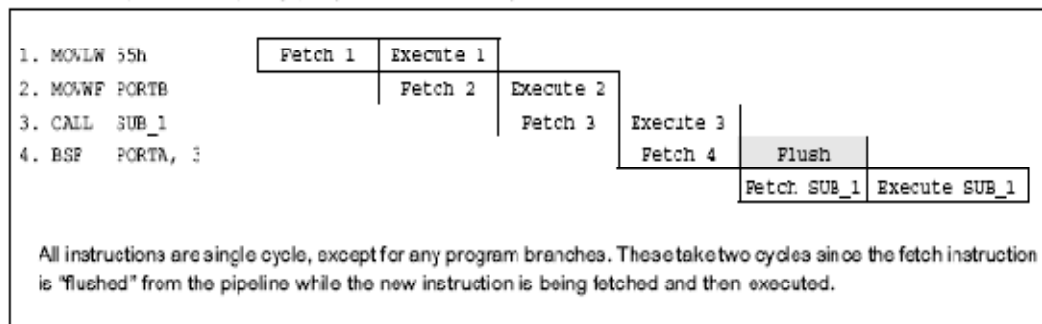
A fetch cycle begins with the program counter (PC) incrementing in Q1.

In the execution cycle, the fetched instruction is latched into the Instruction Register (IR) in cycle Q1. This instruction is then decoded and executed during the Q2, Q3, and Q4 cycles. Data memory is read during Q2 (operand read) and written during Q4 (destination write).

FIGURE 3-2: CLOCK/INSTRUCTION CYCLE



EXAMPLE 3-1: INSTRUCTION PIPELINE FLOW



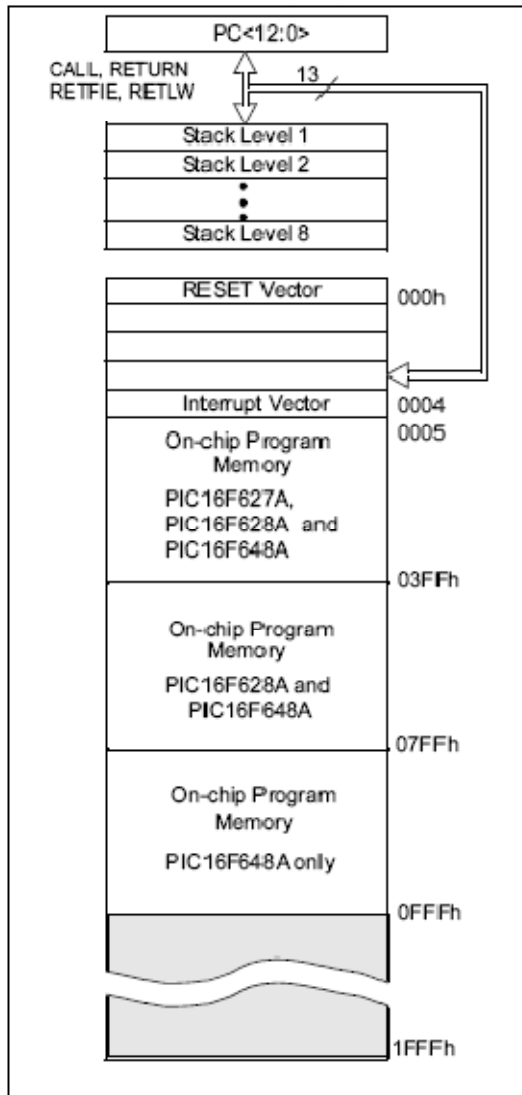
PIC16F627A/628A/648A

4.0 MEMORY ORGANIZATION

4.1 Program Memory Organization

The PIC16F627A/628A/648A has a 13-bit program counter capable of addressing an 8K x 14 program memory space. Only the first 1K x 14 (0000h - 03FFh) for the PIC16F627A, 2K x 14 (0000h - 07FFh) for the PIC16F628A and 4K x 14 (0000h - 0FFFh) for the PIC16F648A are physically implemented. Accessing a location above these boundaries will cause a wrap-around within the first 1K x 14 space (PIC16F627A), 2K x 14 space (PIC16F628A) or 4K x 14 space (PIC16F648A). The RESET vector is at 0000h and the interrupt vector is at 0004h (Figure 4-1).

FIGURE 4-1: PROGRAM MEMORY MAP AND STACK



4.2 Data Memory Organization

The data memory (Figure 4-2 and Figure 4-3) is partitioned into four banks, which contain the general purpose registers and the Special Function Registers (SFR). The SFR's are located in the first 32 locations of each Bank. There are general purpose registers implemented as static RAM in each Bank. Table 4-1 lists the general purpose register available in each of the four banks.

TABLE 4-1: GENERAL PURPOSE STATIC RAM REGISTERS

	PIC16F627A/628A	PIC16F648A
Bank0	20-7Fh	20-7Fh
Bank1	A0h-FF	A0h-FF
Bank2	120h-14Fh, 170h-17Fh	120h-17Fh
Bank3	1F0h-1FFh	1F0h-1FFh

Addresses F0h-FFh, 170h-17Fh and 1F0h-1FFh are implemented as common RAM and mapped back to addresses 70h-7Fh.

Table 4-2 lists how to access the four banks of registers via the STATUS Register bits RP1 and RP0.

TABLE 4-2: ACCESS TO BANKS OF REGISTERS

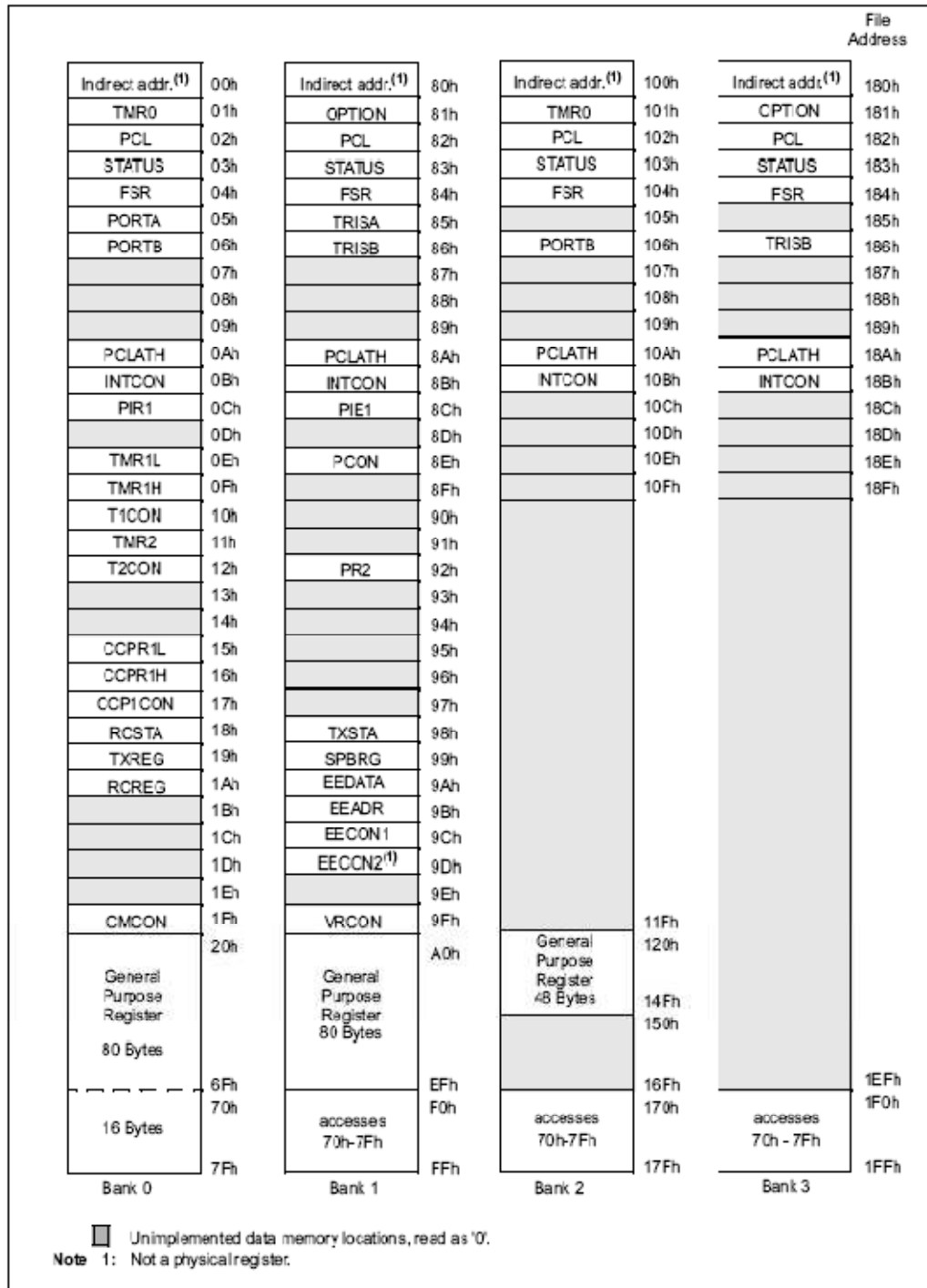
	RP1	RP0
Bank0	0	0
Bank1	0	1
Bank2	1	0
Bank3	1	1

4.2.1 GENERAL PURPOSE REGISTER FILE

The register file is organized as 224 x 8 in the PIC16F627A/628A and 256 x 8 in the PIC16F648A. Each is accessed either directly or indirectly through the File Select Register (FSR). See Section 4.4.

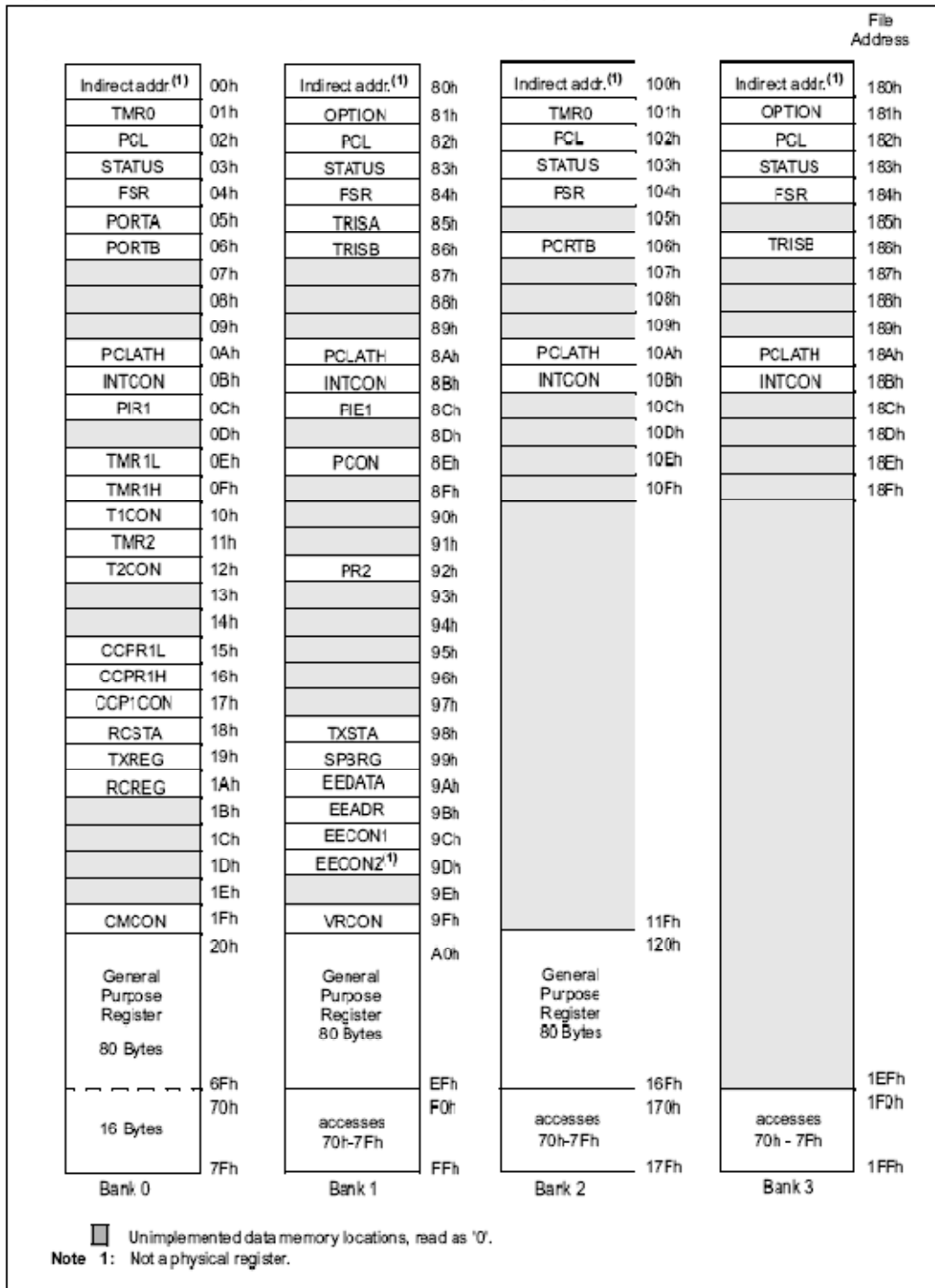
PIC16F627A/628A/648A

FIGURE 4-2: DATA MEMORY MAP OF THE PIC16F627A AND PIC16F628A



PIC16F627A/628A/648A

FIGURE 4-3: DATA MEMORY MAP OF THE PIC16F648A



PIC16F627A/628A/648A

4.2.2 SPECIAL FUNCTION REGISTERS

The SFRs are registers used by the CPU and Peripheral functions for controlling the desired operation of the device (Table 4-3). These registers are static RAM.

The special registers can be classified into two sets (core and peripheral). The SFRs associated with the "core" functions are described in this section. Those related to the operation of the peripheral features are described in the section of that peripheral feature.

TABLE 4-3: SPECIAL REGISTERS SUMMARY BANK 0

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR Reset ⁽¹⁾	Details on Page
Bank 0											
00h	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)								XXXX XXXX	28
01h	TMR0	Timer0 module's Register								XXXX XXXX	45
02h	PCL	Program Counter's (PC) Least Significant Byte								0000 0000	28
03h	STATUS	IRP	RP1	RP0	TO	PD	Z	DC	C	0001 1xxx	22
04h	FSR	Indirect data memory address pointer								XXXX XXXX	28
05h	PORTA	RA7	RA6	RA5	RA4	RA3	RA2	RA1	RA0	XXXX 0000	31
06h	PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	XXXX XXXX	36
07h	—	Unimplemented								—	—
08h	—	Unimplemented								—	—
09h	—	Unimplemented								—	—
0Ah	PCLATH	—	—	—	Write buffer for upper 5 bits of program counter				---	0 0000	28
0Bh	INTCON	GIE	PEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF	0000 000x	24
0Ch	PIR1	EEIF	CMIF	RCIF	TXIF	—	CCP1IF	TMR2IF	TMR1IF	0000 -000	26
0Dh	—	Unimplemented								—	—
0Eh	TMR1L	Holding register for the Least Significant Byte of the 16-bit TMR1								XXXX XXXX	48
0Fh	TMR1H	Holding register for the Most Significant Byte of the 16-bit TMR1								XXXX XXXX	48
10h	T1CON	—	—	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	TMR1CS	TMR1ON	--00 0000	48
11h	TMR2	TMR2 module's register								0000 0000	52
12h	T2CON	—	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0	-000 0000	52
13h	—	Unimplemented								—	—
14h	—	Unimplemented								—	—
15h	CCPR1L	Capture/Compare/FWM register (LSB)								XXXX XXXX	55
16h	CCPR1H	Capture/Compare/FWM register (MSB)								XXXX XXXX	55
17h	CCP1CON	—	—	CCP1X	CCP1Y	CCP1M3	CCP1M2	CCP1M1	CCP1M0	--00 0000	55
18h	RCSTA	SPEN	RX9	SREN	CREN	ADEN	FERR	OERR	RX9D	0000 000x	69
19h	TXREG	USART Transmit data register								0000 0000	76
1Ah	RCREG	USART Receive data register								0000 0000	79
1Bh	—	Unimplemented								—	—
1Ch	—	Unimplemented								—	—
1Dh	—	Unimplemented								—	—
1Eh	—	Unimplemented								—	—
1Fh	CMCON	C2OUT	C1OUT	C2INV	C1INV	CS	CM2	CM1	CM0	0000 0000	61

Legend: — = Unimplemented locations read as '0', u = unchanged, x = unknown, q = value depends on condition, shaded = unimplemented

Note 1: For the Initialization Condition for Registers Tables, refer to Table 14-6 and Table 14-7.

PIC16F627A/628A/648A

TABLE 4-4: SPECIAL FUNCTION REGISTERS SUMMARY BANK1

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR Reset ⁽¹⁾	Details on Page
Bank 1											
80h	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)								xxxx xxxx	28
81h	OPTION	RBPV	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	23
82h	PCL	Program Counter's (PC) Least Significant Byte								0000 0000	28
83h	STATUS	IRP	RP1	RP0	TQ	PD	Z	DC	C	0001 1xxx	22
84h	FSR	Indirect data memory address pointer								xxxx xxxx	28
85h	TRISA	TRISA7	TRISA6	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	1111 1111	31
86h	TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	1111 1111	36
87h	—	Unimplemented								—	—
88h	—	Unimplemented								—	—
89h	—	Unimplemented								—	—
8Ah	PCLATH	—	—	—	Write buffer for upper 5 bits of program counter				---	0000	28
8Bh	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	24
8Ch	PIE1	EEIE	CMIE	RCIE	TXIE	—	CCP1IE	TMR2IE	TMR1IE	0000 -000	25
8Dh	—	Unimplemented								—	—
8Eh	PCON	—	—	—	—	OSCF	—	POR	BOR	---- 1-0x	27
8Fh	—	Unimplemented								—	—
90h	—	Unimplemented								—	—
91h	—	Unimplemented								—	—
92h	PR2	Timer2 Period Register								1111 1111	52
93h	—	Unimplemented								—	—
94h	—	Unimplemented								—	—
95h	—	Unimplemented								—	—
96h	—	Unimplemented								—	—
97h	—	Unimplemented								—	—
98h	TXSTA	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	0000 -010	71
99h	SPBRG	Baud Rate Generator Register								0000 0000	71
9Ah	EEDATA	EEPROM data register								xxxx xxxx	89
9Bh	EEADR	EEPROM address register								xxxx xxxx	90
9Ch	EECON1	—	—	—	—	WRERR	WREN	WR	RD	---- x000	90
9Dh	EECON2	EEPROM control register 2 (not a physical register)								---- ----	90
9Eh	—	Unimplemented								—	—
9Fh	VRCON	VREN	VROE	VRR	—	VR3	VR2	VR1	VR0	000- 0000	67

Legend: — = Unimplemented locations read as '0', u = unchanged, x = unknown, q = value depends on condition, shaded = unimplemented

Note 1: For the Initialization Condition for Registers Tables, refer to Table 14-6 and Table 14-7.

PIC16F627A/628A/648A

TABLE 4-5: SPECIAL FUNCTION REGISTERS SUMMARY BANK2

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR Reset ⁽¹⁾	Details on Page
Bank 2											
00h	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)								xxxx xxxx	28
01h	TMR0	Timer0 module's Register								xxxx xxxx	45
02h	PCL	Program Counter's (PC) Least Significant Byte								0000 0000	28
03h	STATUS	IRP	RP1	RP0	\overline{TO}	\overline{PD}	Z	DC	C	0001 1xxx	22
04h	FSR	Indirect data memory address pointer								xxxx xxxx	28
05h	—	Unimplemented								—	—
06h	PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	xxxx xxxx	36
07h	—	Unimplemented								—	—
08h	—	Unimplemented								—	—
09h	—	Unimplemented								—	—
0Ah	PCLATH	—	—	—	Write buffer for upper 5 bits of program counter					---0 0000	28
0Bh	INTCON	GIE	PEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF	0000 000x	24
0Ch	—	Unimplemented								—	—
0Dh	—	Unimplemented								—	—
0Eh	—	Unimplemented								—	—
0Fh	—	Unimplemented								—	—
10h	—	Unimplemented								—	—
11h	—	Unimplemented								—	—
12h	—	Unimplemented								—	—
13h	—	Unimplemented								—	—
14h	—	Unimplemented								—	—
15h	—	Unimplemented								—	—
16h	—	Unimplemented								—	—
17h	—	Unimplemented								—	—
18h	—	Unimplemented								—	—
19h	—	Unimplemented								—	—
1Ah	—	Unimplemented								—	—
1Bh	—	Unimplemented								—	—
1Ch	—	Unimplemented								—	—
1Dh	—	Unimplemented								—	—
1Eh	—	Unimplemented								—	—
1Fh	—	Unimplemented								—	—

Legend: — = Unimplemented locations read as '0', u = unchanged, x = unknown, q = value depends on condition, shaded = unimplemented.

Note 1: For the Initialization Condition for Registers Tables, refer to Table 14-6 and Table 14-7.

PIC16F627A/628A/648A

TABLE 4-6: SPECIAL FUNCTION REGISTERS SUMMARY BANK3

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR Reset ⁽¹⁾	Details on Page	
Bank 3												
180h	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)									xxxx xxxx	28
181h	OPTION	RBP1	INTEG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 :111	23	
182h	PCL	Program Counter's (PC) Least Significant Byte									0000 :000	28
183h	STATUS	IRP	RF1	RP0	\overline{TD}	\overline{PD}	Z	DC	C	0001 :xxx	22	
184h	FSR	Indirect data memory address pointer									xxxx :xxx	28
185h	—	Unimplemented									—	—
186h	TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	1111 :111	36	
187h	—	Unimplemented									—	—
188h	—	Unimplemented									—	—
189h	—	Unimplemented									—	—
18Ah	PCLATH	—	—	—	Write buffer for upper 5 bits of program counter					---0 :000	28	
18Bh	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 :00x	24	
18Ch	—	Unimplemented									—	—
18Dh	—	Unimplemented									—	—
18Eh	—	Unimplemented									—	—
18Fh	—	Unimplemented									—	—
190h	—	Unimplemented									—	—
191h	—	Unimplemented									—	—
192h	—	Unimplemented									—	—
193h	—	Unimplemented									—	—
194h	—	Unimplemented									—	—
195h	—	Unimplemented									—	—
196h	—	Unimplemented									—	—
197h	—	Unimplemented									—	—
198h	—	Unimplemented									—	—
199h	—	Unimplemented									—	—
19Ah	—	Unimplemented									—	—
19Bh	—	Unimplemented									—	—
19Ch	—	Unimplemented									—	—
19Dh	—	Unimplemented									—	—
19Eh	—	Unimplemented									—	—
19Fh	—	Unimplemented									—	—

Legend: — = Unimplemented locations read as '0', u = unchanged, x = unknown, q = value depends on condition, shaded = unimplemented

Note 1: For the Initialization Condition for Registers Tables, refer to Table 14-6 and Table 14-7.

PIC16F627A/628A/648A

4.2.2.1 STATUS Register

The STATUS register, shown in Register 4-1, contains the arithmetic status of the ALU; the RESET status and the bank select bits for data memory (SRAM).

The STATUS register can be the destination for any instruction, like any other register. If the STATUS register is the destination for an instruction that affects the Z, DC or C bits, then the write to these three bits is disabled. These bits are set or cleared according to the device logic. Furthermore, the TO and PD bits are non-writable. Therefore, the result of an instruction with the STATUS register as destination may be different than intended.

For example, CLRF STATUS will clear the upper-three bits and set the Z bit. This leaves the status register as "000uu1uu" (where u = unchanged).

It is recommended, therefore, that only BCF, BSF, SWAPF and MOVWF instructions are used to alter the STATUS register because these instructions do not affect any STATUS bit. For other instructions, not affecting any STATUS bits, see the "Instruction Set Summary".

Note 1: The C and DC bits operate as a Borrow and Digit Borrow out bit, respectively, in subtraction. See the SUBLW and SUBWF instructions for examples.

REGISTER 4-1: STATUS REGISTER (ADDRESS: 03h, 83h, 103h, 183h)

	R/W-0	R/W-0	R/W-0	R-1	R-1	R/W-x	R/W-x	R/W-x
	IRP	RP1	RP0	TO	PD	Z	DC	C
	bit 7					bit 0		
bit 7	IRP: Register Bank Select bit (used for indirect addressing) 1 = Bank 2, 3 (100h - 1FFh) 0 = Bank 0, 1 (00h - FFh)							
bit 6-5	RP1:RP0: Register Bank Select bits (used for direct addressing) 00 = Bank 0 (00h - 7Fh) 01 = Bank 1 (80h - FFh) 10 = Bank 2 (100h - 17Fh) 11 = Bank 3 (180h - 1FFh)							
bit 4	TO: Time out bit 1 = After power-up, CLRWDT instruction, or SLEEP instruction 0 = A WDT time out occurred							
bit 3	PD: Power-down bit 1 = After power-up or by the CLRWDT instruction 0 = By execution of the SLEEP instruction							
bit 2	Z: Zero bit 1 = The result of an arithmetic or logic operation is zero 0 = The result of an arithmetic or logic operation is not zero							
bit 1	DC: Digit carry/borrow bit (ADDWF, ADDLW, SUBLW, SUBWF instructions) (for borrow the polarity is reversed) 1 = A carry-out from the 4th low order bit of the result occurred 0 = No carry-out from the 4th low order bit of the result							
bit 0	C: Carry/borrow bit (ADDWF, ADDLW, SUBLW, SUBWF instructions) 1 = A carry-out from the Most Significant bit of the result occurred 0 = No carry-out from the Most Significant bit of the result occurred							
	Note: For borrow, the polarity is reversed. A subtraction is executed by adding the two's complement of the second operand. For rotate (RRF, RLF) instructions, this bit is loaded with either the high or low order bit of the source register.							

Legend:

R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared x = Bit is unknown

PIC16F627A/628A/648A

4.2.2.2 OPTION Register

The OPTION register is a readable and writable register which contains various control bits to configure the TMR0/WDT prescaler, the external RB0/INT interrupt, TMR0 and the weak pull-ups on PORTB.

Note: To achieve a 1:1 prescaler assignment for TMR0, assign the prescaler to the WDT (PSA = 1). See Section 6.3.1.

REGISTER 4-2: OPTION REGISTER (ADDRESS: 81h, 181h)

	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
	RBPU	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0
bit 7								bit 0

- bit 7 **RBPU:** PORTB Pull-up Enable bit
1 = PORTB pull-ups are disabled
0 = PORTB pull-ups are enabled by individual port latch values
- bit 6 **INTEDG:** Interrupt Edge Select bit
1 = Interrupt on rising edge of RB0/INT pin
0 = Interrupt on falling edge of RB0/INT pin
- bit 5 **T0CS:** TMR0 Clock Source Select bit
1 = Transition on RA4/T0CKI pin
0 = Internal instruction cycle clock (CLKOUT)
- bit 4 **T0SE:** TMR0 Source Edge Select bit
1 = Increment on high-to-low transition on RA4/T0CKI pin
0 = Increment on low-to-high transition on RA4/T0CKI pin
- bit 3 **PSA:** Prescaler Assignment bit
1 = Prescaler is assigned to the WDT
0 = Prescaler is assigned to the Timer0 module
- bit 2-0 **PS2:PS0:** Prescaler Rate Select bits

Bit Value	TMR0 Rate	WDT Rate
000	1:2	1:1
001	1:4	1:2
010	1:8	1:4
011	1:16	1:8
100	1:32	1:16
101	1:64	1:32
110	1:128	1:64
111	1:256	1:128

Legend:
 R = Readable bit W = Writable bit U = Unimplemented bit, read as '0'
 -n = Value at POR '1' = Bit is set '0' = Bit is cleared x = Bit is unknown

PIC16F627A/628A/648A

4.2.2.3 INTCON Register

The INTCON register is a readable and writable register, which contains the various enable and flag bits for all interrupt sources except the comparator module. See Section 4.2.2.4 and Section 4.2.2.5 for a description of the comparator enable and flag bits.

Note: Interrupt flag bits get set when an interrupt condition occurs regardless of the state of its corresponding enable bit or the global enable bit, GIE (INTCON<7>).

REGISTER 4-3: INTCON REGISTER (ADDRESS: 0Bh, 8Bh, 10Bh, 18Bh)

	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-x
	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF
								bit 0
bit 7								bit 0

- bit 7 GIE:** Global Interrupt Enable bit
 1 = Enables all un-masked interrupts
 0 = Disables all interrupts
- bit 6 PEIE:** Peripheral Interrupt Enable bit
 1 = Enables all un-masked peripheral interrupts
 0 = Disables all peripheral interrupts
- bit 5 T0IE:** TMR0 Overflow Interrupt Enable bit
 1 = Enables the TMR0 interrupt
 0 = Disables the TMR0 interrupt
- bit 4 INTE:** RB0/INT External Interrupt Enable bit
 1 = Enables the RB0/INT external interrupt
 0 = Disables the RB0/INT external interrupt
- bit 3 RBIE:** RB Port Change Interrupt Enable bit
 1 = Enables the RB port change interrupt
 0 = Disables the RB port change interrupt
- bit 2 T0IF:** TMR0 Overflow Interrupt Flag bit
 1 = TMR0 register has overflowed (must be cleared in software)
 0 = TMR0 register did not overflow
- bit 1 INTF:** RB0/INT External Interrupt Flag bit
 1 = The RB0/INT external interrupt occurred (must be cleared in software)
 0 = The RB0/INT external interrupt did not occur
- bit 0 RBIF:** RB Port Change Interrupt Flag bit
 1 = When at least one of the RB7:RB4 pins changed state (must be cleared in software)
 0 = None of the RB7:RB4 pins have changed state

Legend:

R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared x = Bit is unknown

PIC16F627A/628A/648A

4.2.2.4 PIE1 Register

This register contains interrupt enable bits.

REGISTER 4-4: PIE1 REGISTER (ADDRESS: 8Ch)

R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0
EEIE	CMIE	RCIE	TXIE	—	CCP1IE	TMR2IE	TMR1IE
bit 7							bit 0

- bit 7 **EEIE:** EE Write Complete Interrupt Enable Bit
1 = Enables the EE write complete interrupt
0 = Disables the EE write complete interrupt
- bit 6 **CMIE:** Comparator Interrupt Enable bit
1 = Enables the comparator interrupt
0 = Disables the comparator interrupt
- bit 5 **RCIE:** USART Receive Interrupt Enable bit
1 = Enables the USART receive interrupt
0 = Disables the USART receive interrupt
- bit 4 **TXIE:** USART Transmit Interrupt Enable bit
1 = Enables the USART transmit interrupt
0 = Disables the USART transmit interrupt
- bit 3 **Unimplemented:** Read as '0'
- bit 2 **CCP1IE:** CCP1 Interrupt Enable bit
1 = Enables the CCP1 interrupt
0 = Disables the CCP1 interrupt
- bit 1 **TMR2IE:** TMR2 to PR2 Match Interrupt Enable bit
1 = Enables the TMR2 to PR2 match interrupt
0 = Disables the TMR2 to PR2 match interrupt
- bit 0 **TMR1IE:** TMR1 Overflow Interrupt Enable bit
1 = Enables the TMR1 overflow interrupt
0 = Disables the TMR1 overflow interrupt

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

PIC16F627A/628A/648A

4.2.2.5 PIR1 Register

This register contains interrupt flag bits.

Note: Interrupt flag bits get set when an interrupt condition occurs regardless of the state of its corresponding enable bit or the global enable bit, GIE (INTCON<7>). User software should ensure the appropriate interrupt flag bits are clear prior to enabling an interrupt.

REGISTER 4-5: PIR1 REGISTER (ADDRESS: 0Ch)

R/W-0	R/W-0	R-0	R-0	U-0	R/W-0	R/W-0	R/W-0
EEIF	CMIF	RCIF	TXIF	—	CCP1IF	TMR2IF	TMR1IF
bit 7							bit 0

- bit 7 **EEIF:** EEPROM Write Operation Interrupt Flag bit
 1 = The write operation completed (must be cleared in software)
 0 = The write operation has not completed or has not been started
- bit 6 **CMIF:** Comparator Interrupt Flag bit
 1 = Comparator output has changed
 0 = Comparator output has not changed
- bit 5 **RCIF:** USART Receive Interrupt Flag bit
 1 = The USART receive buffer is full
 0 = The USART receive buffer is empty
- bit 4 **TXIF:** USART Transmit Interrupt Flag bit
 1 = The USART transmit buffer is empty
 0 = The USART transmit buffer is full
- bit 3 **Unimplemented:** Read as '0'
- bit 2 **CCP1IF:** CCP1 Interrupt Flag bit
Capture Mode
 1 = A TMR1 register capture occurred (must be cleared in software)
 0 = No TMR1 register capture occurred
Compare Mode
 1 = A TMR1 register compare match occurred (must be cleared in software)
 0 = No TMR1 register compare match occurred
PWM Mode
 Unused in this mode
- bit 1 **TMR2IF:** TMR2 to PR2 Match Interrupt Flag bit
 1 = TMR2 to PR2 match occurred (must be cleared in software)
 0 = No TMR2 to PR2 match occurred
- bit 0 **TMR1IF:** TMR1 Overflow Interrupt Flag bit
 1 = TMR1 register overflowed (must be cleared in software)
 0 = TMR1 register did not overflow

Legend:

R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared x = Bit is unknown

PIC16F627A/628A/648A

4.2.2.6 PCON Register

The PCON register contains flag bits to differentiate between a Power-on Reset, an external MCLR Reset, WDT Reset or a Brown-out Reset.

Note: BOR is unknown on Power-on Reset. It must then be set by the user and checked on subsequent RESETS to see if BOR is cleared, indicating a brown-out has occurred. The BOR STATUS bit is a "don't care" and is not necessarily predictable if the brown-out circuit is disabled (by clearing the BOREN bit in the Configuration word).

REGISTER 4-6: PCON REGISTER (ADDRESS: 8Eh)

	U-0	U-0	U-0	U-0	R/W-1	U-0	R/W-0	R/W-x
	—	—	—	—	OSCF	—	POR	BOR
bit 7								bit 0

- bit 7-4 **Unimplemented:** Read as '0'
- bit 3 **OSCF:** INTOSC oscillator frequency
 1 = 4 MHz typical
 0 = 37 kHz typical
- bit 2 **Unimplemented:** Read as '0'
- bit 1 **POR:** Power-on Reset STATUS bit
 1 = No Power-on Reset occurred
 0 = A Power-on Reset occurred (must be set in software after a Power-on Reset occurs)
- bit 0 **BOR:** Brown-out Reset STATUS bit
 1 = No Brown-out Reset occurred
 0 = A Brown-out Reset occurred (must be set in software after a Brown-out Reset occurs)

Legend:

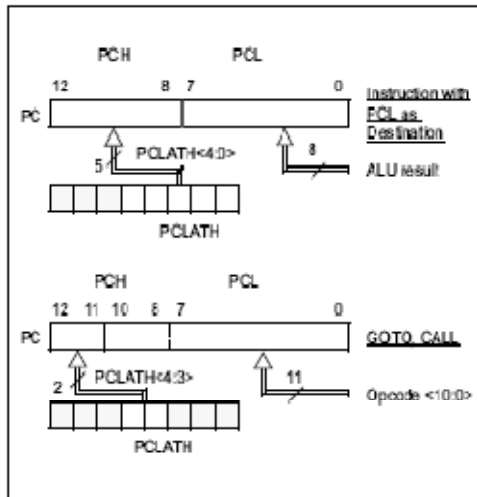
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared x = Bit is unknown

PIC16F627A/628A/648A

4.3 PCL and PCLATH

The program counter (PC) is 13-bits wide. The low byte (PC<12:0>) comes from the PCL register, which is a readable and writable register. The high byte (PC<12:8>) is not directly readable or writable and comes from PCLATH. On any RESET, the PC is cleared. Figure 4-4 shows the two situations for loading the PC. The upper example in Figure 4-4 shows how the PC is loaded on a write to PCL (PCLATH<4:0> → PCH). The lower example in Figure 4-4 shows how the PC is loaded during a CALL or GOTO instruction (PCLATH<4:3> → PCH).

FIGURE 4-4: LOADING OF PC IN DIFFERENT SITUATIONS



4.3.1 COMPUTED GOTO

A computed GOTO is accomplished by adding an offset to the program counter (ADDWF PCL). When doing a table read using a computed GOTO method, care should be exercised if the table location crosses a PCL memory boundary (each 256-byte block). Refer to the application note "Implementing a Table Read" (AN556).

4.3.2 STACK

The PIC16F627A/628A/648A family has an 8-level deep x 13-bit wide hardware stack (Figure 4-1). The stack space is not part of either program or data space and the stack pointer is not readable or writable. The PC is PUSHed onto the stack when a CALL instruction is executed or an interrupt causes a branch. The stack is POPed in the event of a RETURN, RETLW or RETFIE instruction execution. PCLATH is not affected by a PUSH or POP operation.

The stack operates as a circular buffer. This means that after the stack has been PUSHed eight times, the ninth push overwrites the value that was stored from the first push. The tenth push overwrites the second push (and so on).

Note 1: There are no STATUS bits to indicate stack overflow or stack underflow conditions.

Note 2: There are no instructions/mnemonics called PUSH or POP. These are actions that occur from the execution of the CALL, RETURN, RETLW and RETFIE instructions, or the vectoring to an interrupt address.

4.4 Indirect Addressing, INDF and FSR Registers

The INDF register is not a physical register. Addressing the INDF register will cause indirect addressing.

Indirect addressing is possible by using the INDF register. Any instruction using the INDF register actually accesses data pointed to by the file select register (FSR). Reading INDF itself indirectly will produce 00h. Writing to the INDF register indirectly results in a no-operation (although STATUS bits may be affected). An effective 9-bit address is obtained by concatenating the 8-bit FSR register and the IRP bit (STATUS<7>), as shown in Figure 4-5.

A simple program to clear RAM location 20h-2Fh using indirect addressing is shown in Example 4-1.

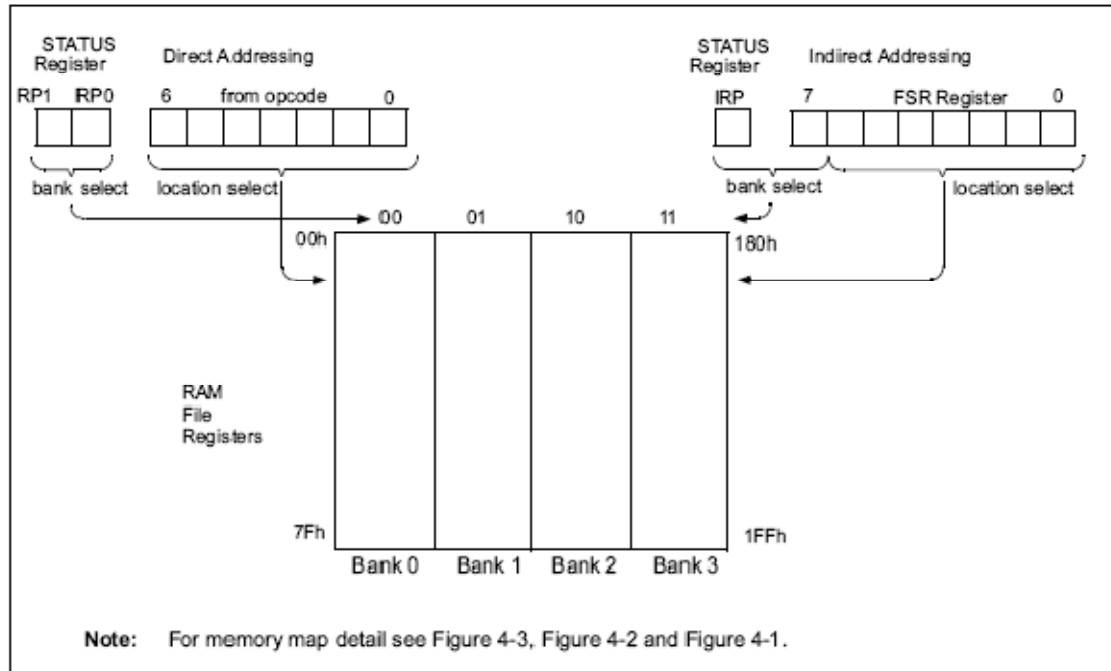
EXAMPLE 4-1: Indirect Addressing

```

MOVW 0x20 ;initialize pointer
MOVWF FSR ;to RAM
NEXT  CLRF INDF ;clear INDF register
      INCF FSR ;inc pointer
      BTFSS FSR,4 ;all done?
      GOTO NEXT ;no clear next
                      ;yes continue
    
```

PIC16F627A/628A/648A

FIGURE 4-5: DIRECT/INDIRECT ADDRESSING PIC16F627A/628A/648A



PIC16F627A/628A/648A

5.0 I/O PORTS

The PIC16F627A/628A/648A have two ports, PORTA and PORTB. Some pins for these I/O ports are multiplexed with alternate functions for the peripheral features on the device. In general, when a peripheral is enabled, that pin may not be used as a general purpose I/O pin.

5.1 IPORTA and TRISA Registers

PORTA is an 8-bit wide latch. RA4 is a Schmitt Trigger input and an open drain output. Port RA4 is multiplexed with the TOCK1 clock input. RA5⁽¹⁾ is a Schmitt Trigger input only and has no output drivers. All other RA port pins have Schmitt Trigger input levels and full CMOS output drivers. All pins have data direction bits (TRIS registers) which can configure these pins as input or output.

A '1' in the TRISA register puts the corresponding output driver in a High-impedance mode. A '0' in the TRISA register puts the contents of the output latch on the selected pin(s).

Reading the PORTA register reads the status of the pins whereas writing to it will write to the port latch. All write operations are read-modify-write operations. So a write to a port implies that the port pins are first read, then this value is modified and written to the port data latch.

The PORTA pins are multiplexed with comparator and voltage reference functions. The operation of these pins are selected by control bits in the CMCON (comparator control register) register and the VRCON (voltage reference control register) register. When selected as a comparator input, these pins will read as '0's.

Note 1: RA5 shares function with V_{REF}. When V_{REF} voltage levels are applied to RA5, the device will enter Programming mode.

2: On RESET, the TRISA register is set to all inputs. The digital inputs (RA<3:0>) are disabled and the comparator inputs are forced to ground to reduce current consumption.

3: TRISA<6:7> is overridden by oscillator configuration. When PORTA<6:7> is overridden, the data reads '0' and the TRISA<6:7> bits are ignored.

TRISA controls the direction of the RA pins, even when they are being used as comparator inputs. The user must make sure to keep the pins configured as inputs when using them as comparator inputs.

The RA2 pin will also function as the output for the voltage reference. When in this mode, the V_{REF} pin is a very high-impedance output. The user must configure TRISA<2> bit as an input and use high-impedance loads.

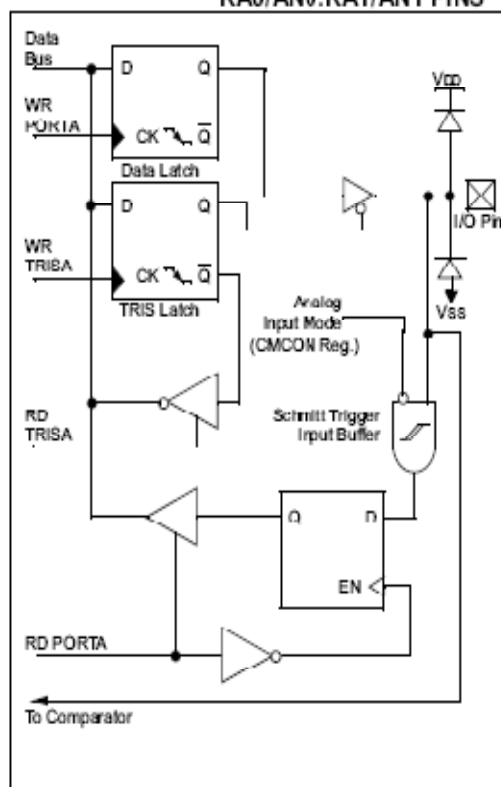
In one of the Comparator modes defined by the CMCON register, pins RA3 and RA4 become outputs of the comparators. The TRISA<4:3> bits must be cleared to enable outputs to use this function.

EXAMPLE 5-1: Initializing PORTA

```

CLRWF  PORTA      ;Initialize PORTA by
                  ;setting
                  ;output data latches
MOVLW  0x07       ;Turn comparators off and
MOVWF  CMCON      ;enable pins for I/O
                  ;functions
BCF    STATUS, RP1 ;Select Bank1
BSF    STATUS, RP0 ;Select Bank1
MOVLW  0x1F       ;Value used to initialize
                  ;data direction
MOVWF  TRISA      ;Set RA<4:0> as inputs
                  ;TRISA<5> always
                  ;read as '1'.
                  ;TRISA<7:6>
                  ;depend on oscillator
                  ;mode
    
```

FIGURE 5-1: BLOCK DIAGRAM OF RA0/AN0:RA1/AN1 PINS



PIC16F627A/628A/648A

TABLE 5-1: PORTA FUNCTIONS

Name	Function	Input Type	Output Type	Description
RA0/AN0	RA0	ST	CMOS	Bi-directional I/O port
	AN0	AN	—	Analog comparator input
RA1/AN1	RA1	ST	CMOS	Bi-directional I/O port
	AN1	AN	—	Analog comparator input
RA2/AN2/VREF	RA2	ST	CMOS	Bi-directional I/O port
	AN2	AN	—	Analog comparator input
	VREF	—	AN	VREF output
RA3/AN3/CMP1	RA3	ST	CMOS	Bi-directional I/O port
	AN3	AN	—	Analog comparator input
	CMP1	—	CMOS	Comparator 1 output
RA4/T0CKI/CMP2	RA4	ST	OD	Bi-directional I/O port. Output is open drain type.
	T0CKI	ST	—	External clock input for TMR0 or comparator output
	CMP2	—	OD	Comparator 2 output
RA5/MCLR/VPP	RA5	ST	—	Input port
	MCLR	ST	—	Master clear. When configured as MCLR, this pin is an active low RESET to the device. Voltage on MCLR/VPP must not exceed VDD during normal device operation.
	VPP	HV	—	Programming voltage input.
RA6/OSC2/CLKOUT	RA6	ST	CMOS	Bi-directional I/O port
	OSC2	—	XTAL	Oscillator crystal output. Connects to crystal resonator in Crystal Oscillator mode.
	CLKOUT	—	CMOS	In RC or INTOSC mode. OSC2 pin can output CLKOUT, which has 1/4 the frequency of OSC1
RA7/OSC1/CLKIN	RA7	ST	CMOS	Bi-directional I/O port
	OSC1	XTAL	—	Oscillator crystal input. Connects to crystal resonator in Crystal Oscillator mode.
	CLKIN	ST	—	External clock source input. RC biasing pin.

Legend: O = Output
 — = Not used
 TTL = TTL Input

CMOS = CMOS Output
 I = Input
 OD = Open Drain Output

P = Power
 ST = Schmitt Trigger Input
 AN = Analog

PIC16F627A/628A/648A

TABLE 5-2: SUMMARY OF REGISTERS ASSOCIATED WITH PORTA⁽¹⁾

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR	Value on All Other RESETS
05h	PORTA	RA7	RA6	RA5 ⁽²⁾	RA4	RA3	RA2	RA1	RA0	xxxx 0000	qqqq 0000
85h	TRISA	TRISA7	TRISA6	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	1111 1111	1111 1111
1Fh	CMCON	C2OUT	C1OUT	C2INV	C1INV	CIS	CM2	CM1	CM0	0000 0000	0000 0000
9Fh	VRCON	VREN	VROE	VRR	—	VR3	VR2	VR1	VR0	000- 0000	000- 0000

Legend: — = Unimplemented locations read as '0', u = unchanged, x = unknown, q = value depends on condition, shaded = unimplemented

Note 1: Shaded bits are not used by PORTA.

2: MCLR Configuration Bit sets RA5 functionality.

5.2 PORTB and TRISB Registers

PORTB is an 8-bit wide bi-directional port. The corresponding data direction register is TRISB. A '1' in the TRISB register puts the corresponding output driver in a High-impedance mode. A '0' in the TRISB register puts the contents of the output latch on the selected pin(s).

PORTB is multiplexed with the external interrupt, USART, CCP module and the TMR1 clock input/output. The standard port functions and the alternate port functions are shown in Table 5-3. Alternate port functions may override TRIS setting when enabled.

Reading PORTB register reads the status of the pins, whereas writing to it will write to the port latch. All write operations are read-modify-write operations. So a write to a port implies that the port pins are first read, then this value is modified and written to the port data latch.

Each of the PORTB pins has a weak internal pull-up ($\approx 200 \mu\text{A}$ typical). A single control bit can turn on all the pull-ups. This is done by clearing the RBPU (OPTION<7>) bit. The weak pull-up is automatically turned off when the port pin is configured as an output. The pull-ups are disabled on Power-on Reset.

Four of PORTB's pins, RB<7:4>, have an interrupt-on-change feature. Only pins configured as inputs can cause this interrupt to occur (i.e., any RB<7:4> pin configured as an output is excluded from the interrupt-on-change comparison). The input pins (of RB7:RB4) are compared with the old value latched on the last read of PORTB. The "mismatch" outputs of RB7:RB4 are OR'ed together to generate the RBIF interrupt (flag latched in INTCON<0>).

This interrupt can wake the device from SLEEP. The user, in the interrupt service routine, can clear the interrupt in the following manner:

- Any read or write of PORTB. This will end the mismatch condition.
- Clear flag bit RBIF.

A mismatch condition will continue to set flag bit RBIF. Reading PORTB will end the mismatch condition and allow flag bit RBIF to be cleared.

This interrupt on mismatch feature, together with software configurable pull-ups on these four pins allow easy interface to a key pad and make it possible for wake-up on key-depression. (See AN552)

Note: If a change on the I/O pin should occur when a read operation is being executed (start of the Q2 cycle), then the RBIF interrupt flag may not get set.

The interrupt-on-change feature is recommended for wake-up on key depression operation and operations where PORTB is only used for the interrupt-on-change feature. Polling of PORTB is not recommended while using the interrupt-on-change feature.

PIC16F627A/628A/648A

TABLE 5-3: PORTB FUNCTIONS

Name	Function	Input Type	Output Type	Description
RB0/INT	RB0	TTL	CMOS	Bi-directional I/O port. Can be software programmed for internal weak pull-up.
	INT	ST	—	External interrupt.
RB1/RX/DT	RB1	TTL	CMOS	Bi-directional I/O port. Can be software programmed for internal weak pull-up.
	RX	ST	—	USART Receive Pin
	DT	ST	CMOS	Synchronous data I/O
RB2/TX/CK	RB2	TTL	CMOS	Bi-directional I/O port
	TX	—	CMOS	USART Transmit Pin
	CK	ST	CMOS	Synchronous Clock I/O. Can be software programmed for internal weak pull-up.
RB3/CCP1	RB3	TTL	CMOS	Bi-directional I/O port. Can be software programmed for internal weak pull-up.
	CCP1	ST	CMOS	Capture/Compare/PWM/I/O
RB4/PGM	RB4	TTL	CMOS	Bi-directional I/O port. Interrupt-on-pin change. Can be software programmed for internal weak pull-up.
	PGM	ST	—	Low voltage programming input pin. When low voltage programming is enabled, the interrupt-on-pin change and weak pull-up resistor are disabled.
RB5	RB5	TTL	CMOS	Bi-directional I/O port. Interrupt-on-pin change. Can be software programmed for internal weak pull-up.
RB6/T1OSO/T1CKI/PGC	RB6	TTL	CMOS	Bi-directional I/O port. Interrupt-on-pin change. Can be software programmed for internal weak pull-up.
	T1OSO	—	XTAL	Timer1 Oscillator Output
	T1CKI	ST	—	Timer1 Clock Input
	PGC	ST	—	ICSP Programming Clock
RB7/T1OSI/PGD	RB7	TTL	CMOS	Bi-directional I/O port. Interrupt-on-pin change. Can be software programmed for internal weak pull-up.
	T1OSI	XTAL	—	Timer1 Oscillator Input
	PGD	ST	CMOS	ICSP Data I/O

Legend: O = Output CMOS = CMOS Output P = Power
 — = Not used I = Input ST = Schmitt Trigger Input
 TTL = TTL Input OD = Open Drain Output AN = Analog

TABLE 5-4: SUMMARY OF REGISTERS ASSOCIATED WITH PORTB⁽¹⁾

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR	Value on All Other RESETS
06h, 106h	PORTB	RB7	RB6	RB5	RB4 ⁽²⁾	RB3	RB2	RB1	RB0	xxxx xxxx	uuuu uuuu
86h, 186h	TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	1111 1111	1111 1111
81h, 181h	OPTION	RBPU	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	1111 1111

Legend: u = unchanged, x = unknown

Note 1: Shaded bits are not used by PORTB.

Note 2: LVP Configuration Bit sets RB4 functionality.

PIC16F627A/628A/648A

5.3 I/O Programming Considerations

5.3.1 BI-DIRECTIONAL I/O PORTS

Any instruction that writes, operates internally as a read followed by a write operation. The `BCF` and `BSF` instructions, for example, read the register into the CPU, execute the bit operation and write the result back to the register. Caution must be used when these instructions are applied to a port with both inputs and outputs defined. For example, a `BSF` operation on bit5 of `PORTB` will cause all eight bits of `PORTB` to be read into the CPU. Then the `BSF` operation takes place on bit5 and `PORTB` is written to the output latches. If another bit of `PORTB` is used as a bi-directional I/O pin (e.g., bit0) and is defined as an input at this time, the input signal present on the pin itself would be read into the CPU and rewritten to the data latch of this particular pin, overwriting the previous content. As long as the pin stays in the Input mode, no problem occurs. However, if bit0 is switched into Output mode later on, the content of the data latch may now be unknown.

Reading a port register reads the values of the port pins. Writing to the port register writes the value to the port latch. When using read-modify-write instructions (ex. `BCF`, `BSF`, etc.) on a port, the value of the port pins is read, the desired operation is done to this value, and this value is then written to the port latch.

Example 5-2 shows the effect of two sequential read-modify-write instructions (ex., `BCF`, `BSF`, etc.) on an I/O port.

A pin actively outputting a Low or High should not be driven from external devices at the same time in order to change the level on this pin ("wired-or", "wired-and"). The resulting high output currents may damage the chip.

EXAMPLE 5-2: READ-MODIFY-WRITE INSTRUCTIONS ON AN I/O PORT

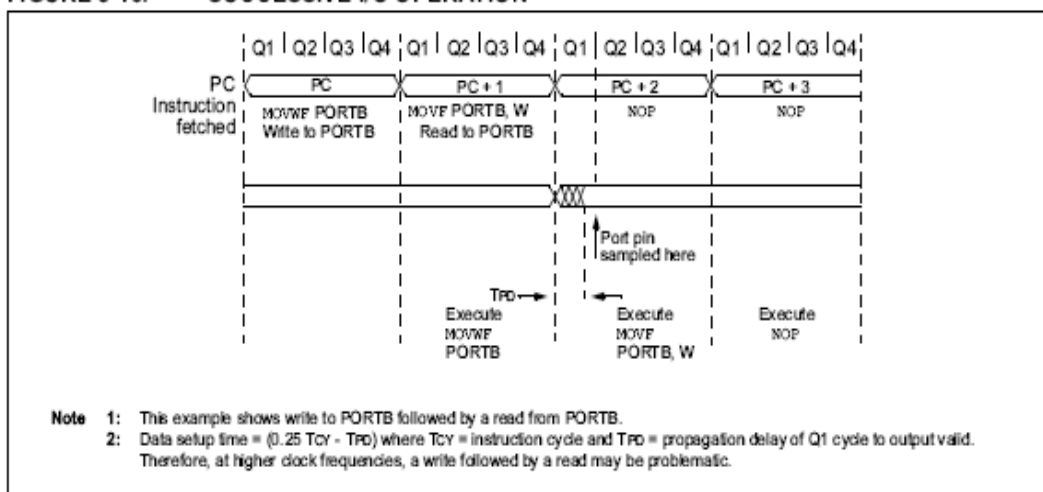
```

;Initial PORT settings:PORTB<7:4> Inputs
;                       PORTB<3:0> Outputs
;PORTB<7:6> have external pull-up and are
;not connected to other circuitry
;
;                       PORT latchPORT Pins
;                       -----
BCF STATUS, RP0      ;
BCF PORTB, 7         ;01pp pppp 11pp pppp
BSF STATUS, RP0      ;
BCF TRISB, 7         ;10pp pppp 11pp pppp
BCF TRISB, 6         ;10pp pppp 10pp pppp
;
;Note that the user may have expected the
;pin values to be 00pp pppp. The 2nd BCF
;caused RB7 to be latched as the pin value
;(High).
    
```

5.3.2 SUCCESSIVE OPERATIONS ON I/O PORTS

The actual write to an I/O port happens at the end of an instruction cycle, whereas for reading, the data must be valid at the beginning of the instruction cycle (Figure 5-16). Therefore, care must be exercised if a write followed by a read operation is carried out on the same I/O port. The sequence of instructions should be such to allow the pin voltage to stabilize (load dependent) before the next instruction, which causes that file to be read into the CPU, is executed. Otherwise, the previous state of that pin may be read into the CPU rather than the new state. When in doubt, it is better to separate these instructions with a `NOP` or another instruction not accessing this I/O port.

FIGURE 5-16: SUCCESSIVE I/O OPERATION



PIC16F627A/628A/648A

6.0 TIMER0 MODULE

The Timer0 module timer/counter has the following features:

- 8-bit timer/counter
- Read/Write capabilities
- 8-bit software programmable prescaler
- Internal or external clock select
- Interrupt on overflow from FFh to 00h
- Edge select for external clock

Figure 6-1 is a simplified block diagram of the Timer0 module. Additional information is available in the PICmicro™ Mid-Range MCU Family Reference Manual, DS33023.

Timer mode is selected by clearing the T0CS bit (OPTION<5>). In Timer mode, the TMR0 register value will increment every instruction cycle (without prescaler). If the TMR0 register is written to, the increment is inhibited for the following two cycles. The user can work around this by writing an adjusted value to the TMR0 register.

Counter mode is selected by setting the T0CS bit. In this mode the TMR0 register value will increment either on every rising or falling edge of pin RA4/T0CKI. The incrementing edge is determined by the source edge (T0SE) control bit (OPTION<4>). Clearing the T0SE bit selects the rising edge. Restrictions on the external clock input are discussed in detail in Section 6.2.

The prescaler is shared between the Timer0 module and the Watchdog Timer. The prescaler assignment is controlled in software by the control bit PSA (OPTION<3>). Clearing the PSA bit will assign the prescaler to Timer0. The prescaler is not readable or writable. When the prescaler is assigned to the Timer0 module, prescale value of 1:2, 1:4, ..., 1:256 are selectable. Section 6.3 details the operation of the prescaler.

6.1 Timer0 Interrupt

Timer0 interrupt is generated when the TMR0 register timer/counter overflows from FFh to 00h. This overflow sets the T0IF bit. The interrupt can be masked by clearing the T0IE bit (INTCON<5>). The T0IF bit (INTCON<2>) must be cleared in software by the Timer0 module interrupt service routine before re-enabling this interrupt. The Timer0 interrupt cannot wake the processor from SLEEP since the timer is shut off during SLEEP.

6.2 Using Timer0 with External Clock

When an external clock input is used for Timer0, it must meet certain requirements. The external clock requirement is due to internal phase clock (Tosc) synchronization. Also, there is a delay in the actual incrementing of Timer0 after synchronization.

6.2.1 EXTERNAL CLOCK SYNCHRONIZATION

When no prescaler is used, the external clock input is the same as the prescaler output. The synchronization of T0CKI with the internal phase clocks is accomplished by sampling the prescaler output on the Q2 and Q4 cycles of the internal phase clocks (Figure 6-1). Therefore, it is necessary for T0CKI to be high for at least $2T_{osc}$ (and a small RC delay of 20 ns) and low for at least $2T_{osc}$ (and a small RC delay of 20 ns). Refer to the electrical specification of the desired device.

When a prescaler is used, the external clock input is divided by the asynchronous ripple-counter type prescaler so that the prescaler output is symmetrical. For the external clock to meet the sampling requirement, the ripple-counter must be taken into account. Therefore, it is necessary for T0CKI to have a period of at least $4T_{osc}$ (and a small RC delay of 40 ns) divided by the prescaler value. The only requirement on T0CKI high and low time is that they do not violate the minimum pulse width requirement of 10 ns. Refer to parameters 40, 41 and 42 in the electrical specification of the desired device. See Table 17-9.

PIC16F627A/628A/648A

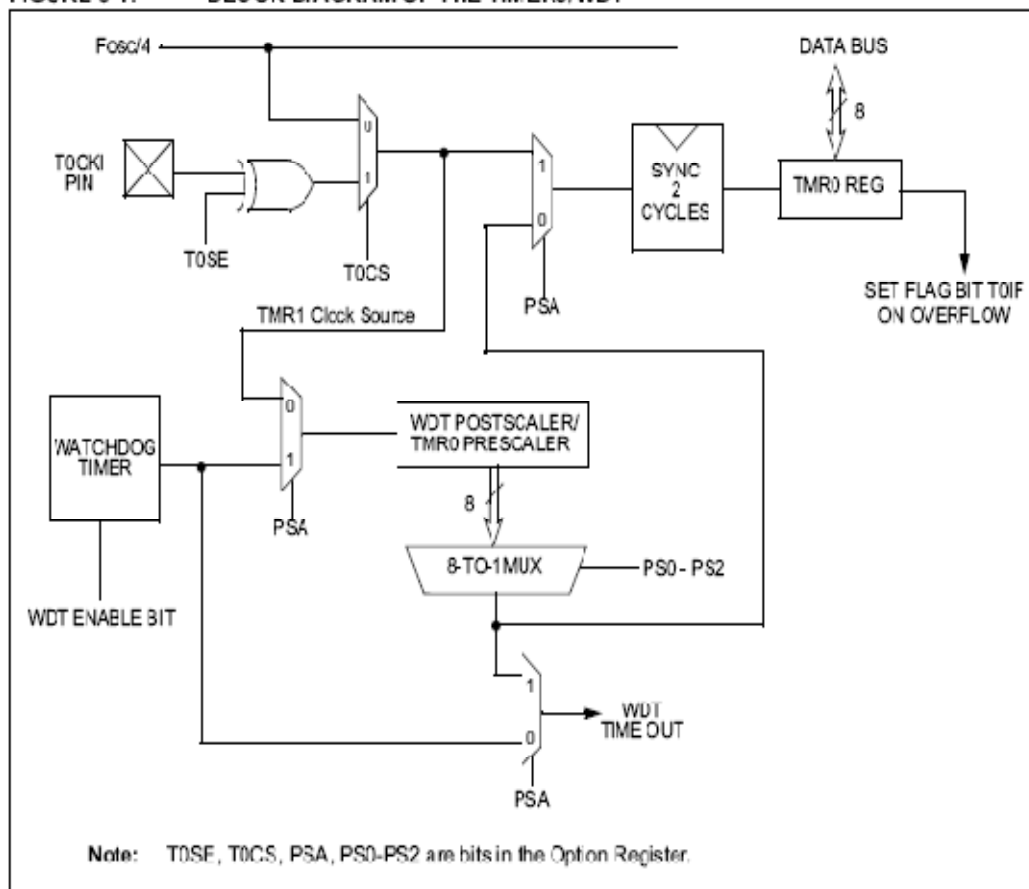
6.3 Timer0 Prescaler

An 8-bit counter is available as a prescaler for the Timer0 module, or as a postscaler for the Watchdog Timer. A prescaler assignment for the Timer0 module means that there is no postscaler for the Watchdog Timer, and vice-versa.

The PSA and PS2:PS0 bits (OPTION<3:0>) determine the prescaler assignment and prescale ratio.

When assigned to the Timer0 module, all instructions writing to the TMR0 register (e.g., CLRF 1, MOVWF 1, BSF 1, x...etc.) will clear the prescaler. When assigned to WDT, a CLRWDT instruction will clear the prescaler along with the Watchdog Timer. The prescaler is not readable or writable.

FIGURE 6-1: BLOCK DIAGRAM OF THE TIMER0/WDT



PIC16F627A/628A/648A

7.0 TIMER1 MODULE

The Timer1 module is a 16-bit timer/counter consisting of two 8-bit registers (TMR1H and TMR1L) which are readable and writable. The TMR1 register pair (TMR1H:TMR1L) increments from 0000h to FFFFh and rolls over to 0000h. The Timer1 Interrupt, if enabled, is generated on overflow of the TMR1 register pair which latches the interrupt flag bit TMR1IF (PIR1<0>). This interrupt can be enabled/disabled by setting/clearing the Timer1 interrupt enable bit TMR1IE (PIE1<0>).

Timer1 can operate in one of two modes:

- As a timer
- As a counter

The Operating mode is determined by the clock select bit TMR1CS (T1CON<1>).

In Timer mode, the TMR1 register pair value increments every instruction cycle. In Counter mode, it increments on every rising edge of the external clock input.

Timer1 can be enabled/disabled by setting/clearing control bit TMR1ON (T1CON<0>).

Timer1 also has an internal "RESET" input. This RESET can be generated by the CCP module (Section 9.0). Register 7-1 shows the Timer1 control register.

For the PIC16F627A/628A/643A, when the Timer1 oscillator is enabled (T1OSCEN is set), the RB7/T1OSI and RB6/T1OSO/T1CKI pins become inputs. That is, the TRISB<7:6> value is ignored.

REGISTER 7-1: T1CON: TIMER1 CONTROL REGISTER (ADDRESS: 10h)

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	
—	—	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	TMR1CS	TMR1ON	
bit 7								bit 0

- bit 7-6 **Unimplemented:** Read as '0'
- bit 5-4 **T1CKPS1:T1CKPS0:** Timer1 Input Clock Prescale Select bits
 11 = 1:8 Prescale value
 10 = 1:4 Prescale value
 01 = 1:2 Prescale value
 00 = 1:1 Prescale value
- bit 3 **T1OSCEN:** Timer1 Oscillator Enable Control bit
 1 = Oscillator is enabled
 0 = Oscillator is shut off⁽¹⁾
- bit 2 **T1SYNC:** Timer1 External Clock Input Synchronization Control bit
TMR1CS = 1
 1 = Do not synchronize external clock input.
 0 = Synchronize external clock input
TMR1CS = 0
 This bit is ignored. Timer1 uses the internal clock when TMR1CS = 0.
- bit 1 **TMR1CS:** Timer1 Clock Source Select bit
 1 = External clock from pin RB6/T1OSO/T1CKI (on the rising edge)
 0 = Internal clock (F_{osc}/4)
- bit 0 **TMR1ON:** Timer1 On bit
 1 = Disables Timer1
 0 = Stops Timer1

Note 1: The oscillator inverter and feedback resistor are turned off to eliminate power drain.

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

PIC16F627A/628A/648A

7.1 Timer1 Operation in Timer Mode

Timer mode is selected by clearing the TMR1CS (T1CON<1>) bit. In this mode, the input clock to the timer is $F_{osc}/4$. The synchronize control bit T1SYNC (T1CON<2>) has no effect since the internal clock is always in sync.

7.2 Timer1 Operation in Synchronized Counter Mode

Counter mode is selected by setting bit TMR1CS. In this mode the TMR1 register pair value increments on every rising edge of clock input on pin RB7/T1OSI when bit T1OSCEN is set or pin RB6/T1OSO/T1CKI when bit T1OSCEN is cleared.

If T1SYNC is cleared, then the external clock input is synchronized with internal phase clocks. The synchronization is done after the prescaler stage. The prescaler stage is an asynchronous ripple-counter.

In this configuration, during SLEEP mode, the TMR1 register pair value will not increment even if the external clock is present, since the synchronization circuit is shut off. The prescaler however will continue to increment.

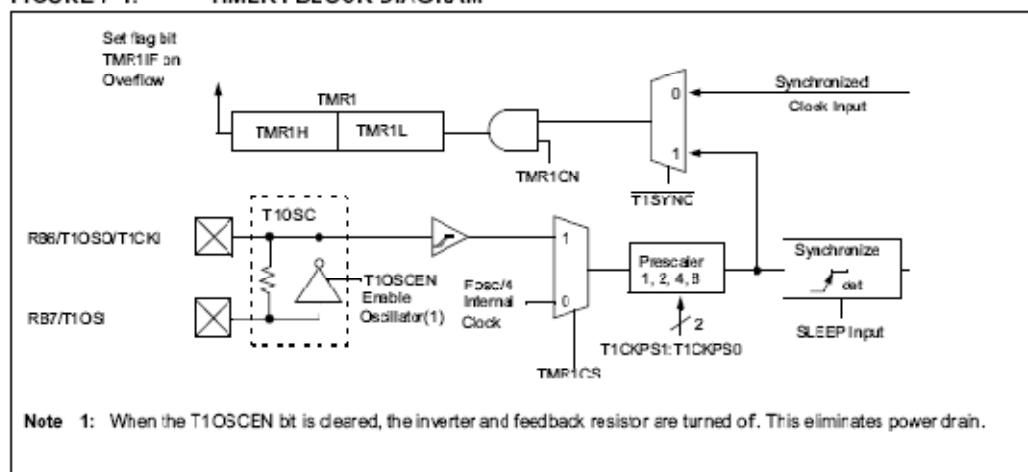
7.2.1 EXTERNAL CLOCK INPUT TIMING FOR SYNCHRONIZED COUNTER MODE

When an external clock input is used for Timer1 in synchronized Counter mode, it must meet certain requirements. The external clock requirement is due to internal phase clock (T_{osc}) synchronization. Also, there is a delay in the actual incrementing of the TMR1 register pair value after synchronization.

When the prescaler is 1:1, the external clock input is the same as the prescaler output. The synchronization of T1CKI with the internal phase clocks is accomplished by sampling the prescaler output on the Q2 and Q4 cycles of the internal phase clocks. Therefore, it is necessary for T1CKI to be high for at least $2T_{osc}$ (and a small RC delay of 20 ns) and low for at least $2T_{osc}$ (and a small RC delay of 20 ns). Refer to the appropriate electrical specifications, parameters 45, 46, and 47.

When a prescaler other than 1:1 is used, the external clock input is divided by the asynchronous ripple-counter type prescaler so that the prescaler output is symmetrical. In order for the external clock to meet the sampling requirement, the ripple-counter must be taken into account. Therefore, it is necessary for T1CKI to have a period of at least $4T_{osc}$ (and a small RC delay of 40 ns) divided by the prescaler value. The only requirement on T1CK high and low time is that they do not violate the minimum pulse width requirements of 10 ns). Refer to the appropriate electrical specifications, parameters 45, 46, and 47.

FIGURE 7-1: TIMER1 BLOCK DIAGRAM



Note 1: When the T1OSCEN bit is cleared, the inverter and feedback resistor are turned off. This eliminates power drain.

PIC16F627A/628A/648A

7.3 Timer1 Operation in Asynchronous Counter Mode

If control bit $\overline{T1SYNC}$ (T1CON<2>) is set, the external clock input is not synchronized. The timer continues to increment asynchronous to the internal phase docks. The timer will continue to run during SLEEP and can generate an interrupt on overflow, which will wake-up the processor. However, special precautions in software are needed to read/write the timer (Section 7.3.2).

Note: In Asynchronous Counter mode, Timer1 cannot be used as a time-base for capture or compare operations.

7.3.1 EXTERNAL CLOCK INPUT TIMING WITH UNSYNCHRONIZED CLOCK

If control bit $\overline{T1SYNC}$ is set, the timer will increment completely asynchronously. The input clock must meet certain minimum high and low time requirements. Refer to Table 17-9 in the Electrical Specifications Section, timing parameters 45, 46, and 47.

7.3.2 READING AND WRITING TIMER1 IN ASYNCHRONOUS COUNTER MODE

Reading the TMR1H or TMR1L register while the timer is running, from an external asynchronous clock, will produce a valid read (taken care of in hardware). However, the user should keep in mind that reading the 16-bit timer in two 8-bit values itself poses certain problems since the timer may overflow between the reads.

For writes, it is recommended that the user simply stop the timer and write the desired values. A write contention may occur by writing to the timer registers while the register is incrementing. This may produce an unpredictable value in the timer register.

Reading the 16-bit value requires some care. Example 7-1 is an example routine to read the 16-bit timer value. This is useful if the timer cannot be stopped.

EXAMPLE 7-1: READING A 16-BIT FREE-RUNNING TIMER

```
; All interrupts are disabled
MOVWF TMR1H, W ;Read high byte
MOVWF TMPH ;
MOVWF TMR1L, W ;Read low byte
MOVWF TMPL ;
MOVWF TMR1H, W ;Read high byte
SUBWF TMPH, W ;Sub 1st read with
;2nd read
BTFSC STATUS, 2 ;Is result = 0
GOTO CONTINUE ;Good 16-bit read
;
; TMR1L may have rolled over between the
; read of the high and low bytes. Reading
; the high and low bytes now will read a good
; value.
;
MOVWF TMR1H, W ;Read high byte
MOVWF TMPH ;
MOVWF TMR1L, W ;Read low byte
MOVWF TMPL ;
; Re-enable the Interrupts (if required)
CONTINUE ;Continue with your
;code
```


PIC16F627A/628A/648A

7.4 Timer1 Oscillator

A crystal oscillator circuit is built in between pins T1OSI (input) and T1OSO (amplifier output). It is enabled by setting control bit T1OSCEN (T1CON<3>). It will continue to run during SLEEP. It is primarily intended for a 32.768 kHz watch crystal. Table 7-1 shows the capacitor selection for the Timer1 oscillator.

The user must provide a software time delay to ensure proper oscillator start-up.

TABLE 7-1: CAPACITOR SELECTION FOR THE TIMER1 OSCILLATOR

Freq	C1	C2
32.768 kHz	15 pF	15 pF
These values are for design guidance only. Consult AN826 (DS00826) for further information on Crystal/Capacitor Selection.		

7.5 Resetting Timer1 Using a CCP Trigger Output

If the CCP1 module is configured in Compare mode to generate a "special event trigger" (CCP1M3:CCP1M0 = 1011), this signal will RESET Timer1.

Note: The special event triggers from the CCP1 module will not set interrupt flag bit TMR1IF (PIR1<0>).

Timer1 must be configured for either timer or synchronized Counter mode to take advantage of this feature. If Timer1 is running in Asynchronous Counter mode, this RESET operation may not work.

In the event that a write to Timer1 coincides with a special event trigger from CCP1, the write will take precedence.

In this mode of operation, the CCPxH:CCPxL registers pair effectively becomes the period register for Timer1.

7.6 Resetting Timer1 Register Pair (TMR1H, TMR1L)

TMR1H and TMR1L registers are not reset to 00h on a POR or any other RESET except by the CCP1 special event triggers.

T1CON register is reset to 00h on a Power-on Reset or a Brown-out Reset, which shuts off the timer and leaves a 1:1 prescale. In all other RESETS, the register is unaffected.

7.7 Timer1 Prescaler

The prescaler counter is cleared on writes to the TMR1H or TMR1L registers.

TABLE 7-2: REGISTERS ASSOCIATED WITH TIMER1 AS A TIMER/COUNTER

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR	Value on all other RESETS
08h, 88h, 108h, 188h	INTCON	GIE	PBE	TOIE	INTE	RBIE	TOIF	INTF	RBIF	0000 (00x)	0000 000u
0Ch	PIR1	EEIF	CMIF	RCIF	TXIF	—	CCP1IF	TMR2IF	TMR1IF	0000 -000	0000 -000
8Ch	PIE1	EEIE	CME	RCIE	TXIE	—	CCP1IE	TMR2IE	TMR1IE	0000 -000	0000 -000
0Eh	TMR1L	Holding register for the Least Significant Byte of the 16-bit TMR1 register								xxxx xxxx	uuuu uuuu
0Fh	TMR1H	Holding register for the Most Significant Byte of the 16-bit TMR1 register								xxxx xxxx	uuuu uuuu
10h	T1CON	—	—	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	TMR1CS	TMR1ON	--00 (000)	--uu uuuu

Legend: x = unknown, u = unchanged, - = unimplemented read as '0'. Shaded cells are not used by the Timer1 module.

Anexo B.-

Programa del Transmisor:

```
@ device INTRC_OSC_NOCLKOUT;          DA2 VAR PORTA.1
@ device MCLR_OFF;                    DA3 VAR PORTA.2
@ device BOD_OFF;                     DA4 VAR PORTA.3
@ device WDT_OFF;                     POWER=1
@ device PWRT_ON;                     inicio:

CMCON=7                               IF DATO = "1" Then
TRISA=%00000000                       LEDTX =1
TRISB=%00000010                       DA1=1:DA2=1:DA3=1:DA4=0
PORTB=0                                ENDIF
PORTA=%00001111                       IF DATO = "2" Then
DEFINE HSER_RCSTA 90h                  LEDTX =1
DEFINE HSER_TXSTA 20h                  DA1=1:DA2=1:DA3=0:DA4=1
DEFINE HSER_BAUD 2400                  ENDIF
DEFINE HSER_SPBRG 25                   IF DATO="3" Then
;CONFIG INTERRUPTACIONES              LEDTX =1
ON INTERRUPT GoTo DATOS                DA1=1:DA2=1:DA3=0:DA4=0
PIE1.5=1      'RCIE = 1                ENDIF
INTCON.6= 1    'PEIE=1                  IF DATO="4" Then
INTCON.7= 1    'GIE=1                  LEDTX =1
DATO VAR BYTE                             DA1=1:DA2=0:DA3=1:DA4=1
POWER VAR PORTB.0                          ENDIF
LEDTX VAR PORTB.3                          IF DATO = "5" Then
DA1 VAR PORTA.0                           LEDTX =1
```

```

    DA1=1:DA2=0:DA3=1:DA4=0          INTCON.6= 1
ENDIF                                INTCON.7= 1
IF DATO = "6" Then                   Resume
    LEDTX =1                          Enable
    DA1=1:DA2=0:DA3=0:DA4=1          End
    Pause 300
    LEDTX =0
ENDIF
IF DATO="7" Then
    LEDTX =1
    DA1=1:DA2=0:DA3=0:DA4=0
    Pause 300
    LEDTX =0
ENDIF
IF DATO="8" Then
    LEDTX =1
    DA1=0:DA2=1:DA3=1:DA4=1
    Pause 300
    LEDTX =0
ENDIF
GoTo inicio
Disable
DATOS:
IF PIR1.5=1 Then
    HSerin[DATO]
ENDIF
PIE1.5=1

```

Programa del Receptor:

@ device BOD_OFF;

@ device WDT_ON;

@ device PWRT_ON;

ADCON1=%00001110

DEFINE OSC 4

INCLUDE "modedefs.bas"

DEFINE CHAR_PACING 1000

DEFINE ADC_BITS 8

DEFINE ADC_CLOCK 3

DEFINE ADC_SAMPLEUS 50

DEFINE HSER_RCSTA 90h

DEFINE HSER_TXSTA 20h

DEFINE HSER_BAUD 2400

DEFINE HSER_SPBRG 25

CLEARWDT

TRISD=%00001111

TRISB=%00000000

TRISC=%10000000

TRISE=0

PORTB=0

PORTC=0

PORTD=0

PORTE=0

DATO1 VAR BYTE

DATO2 VAR BYTE

DATO3 VAR BYTE

DATO4 VAR BYTE

DATO5 VAR BYTE

DATO6 VAR BYTE

DATO7 VAR BYTE

DATO8 VAR BYTE

DRIV2DAT1 VAR BYTE

DRIV2DAT2 VAR BYTE

DRIV2DAT3 VAR BYTE

DRIV2DAT4 VAR BYTE

DRIV2DAT5 VAR BYTE

DRIV2DAT6 VAR BYTE

DRIV2DAT7 VAR BYTE

DRIV2DAT8 VAR BYTE

SAN VAR BYTE

S1 VAR BYTE

S2 VAR BYTE

S11 VAR BYTE

S22 VAR BYTE

I var byte

J var byte

k var byte

DATOS var byte

PASO VAR WORD

CONPLEMENTO VAR WORD

OJO1 var PORTB.1

OJO2 var PORTB.2

LEDTX1 var portC.2

LEDRX VAR PORTC.5	DRIV2DAT6=90
TX2 VAR PORTC.1	DRIV2DAT7=90
POWER VAR PORTE.0	DRIV2DAT8=90
AY var byte	gosub transm1
ERL1 VAR PORTD.0	gosub transm2
ERL2 VAR PORTD.1	LEDTX1=1
ERL3 VAR PORTD.2	PAUSE 500
ERL4 VAR PORTD.3	LEDTX1=0
POWER=1	pause 500
AY=0	GOSUB OJOS
pause 500	OJO1=1:OJO2=1
HSerOUT[255]	CLEARWDT
SEROUT TX2,T2400,[255]	INICIO:
SEROUT TX2,T2400,[255]	IF PORTD=6 THEN
DATO1=90	CLEARWDT
DATO2=90	DATO1=90:
DATO3=90	DATO2=90:DATO3=90:
DATO4=90	DATO4=90
DATO5=90	DRIV2DAT1=90:DRIV2DAT2=90:LEDTX1=1
DATO6=90	GOSUB TRANSM1:
DATO7=90	gosub transm2:LEDTX1=0
DATO8=90	FOR I=1 TO 9
DRIV2DAT1=90	DATO2=DATO2-10:LEDTX1=1:
DRIV2DAT2=90	GOSUB TRANSM1:PAUSE 50
DRIV2DAT3=90	LEDTX1=0
DRIV2DAT4=90	NEXT
DRIV2DAT5=90	FOR I=1 TO 9

```
DATO3=DATO3-10
LEDTX1=1
GOSUB TRANSM1
PAUSE 50
LEDTX1=0
NEXT
FOR I=1 TO 9
    DATO2=DATO2+10
    LEDTX1=1
    GOSUB TRANSM1
    PAUSE 50
    LEDTX1=0
NEXT
DATO4=0
LEDTX1=1
GOSUB TRANSM1
LEDTX1=0
pause 100
FOR I=1 TO 9
    DATO2=DATO2+10
    DATO1=DATO1+10
    LEDTX1=1
    GOSUB TRANSM1
    PAUSE 50
    LEDTX1=0
NEXT
FOR I=1 TO 4
```

```
DATO1=DATO1+10
LEDTX1=1
GOSUB TRANSM1
PAUSE 50
LEDTX1=0
NEXT
FOR J=1 TO 3
    FOR I=1 TO 4
        DATO6=DATO6-15
        DATO7=DATO7-15
        DATO8=DATO8+15
        DRIV2DAT1=DRIV2DAT1+15
        DRIV2DAT2=DRIV2DAT2-15
        LEDTX1=1
        GOSUB TRANSM1
        gosub transm2
        LEDTX1=0
    NEXT
    PAUSE 200
    FOR I=1 TO 4
        DATO6=DATO6+15
        DATO7=DATO7+15
        DATO8=DATO8-15
        DRIV2DAT1=DRIV2DAT1-15
        DRIV2DAT2=DRIV2DAT2+15
        LEDTX1=1
        GOSUB TRANSM1
```

```

        gosub transm2
        LEDTX1=0
    NEXT
    pause 200
NEXT
FOR I=1 TO 15
    DATO4=DATO4+6
    LEDTX1=1
    GOSUB TRANSM1
    PAUSE 50
    LEDTX1=0
NEXT
FOR I=1 TO 15
    DATO1=DATO1-6
    DATO2=DATO2-6
    DATO3=DATO3+6
    LEDTX1=1
    GOSUB TRANSM1
    PAUSE 50
    LEDTX1=0
NEXT
endif
IF PORTD=7 THEN
    AY=0
    CLEARWDT
    DATO1=140
    DATO2=0
    DATO3=90
    DATO4=90
    DRIV2DAT1=90
    DRIV2DAT2=90
    LEDTX1=1
    GOSUB TRANSM1
    gosub transm2
    LEDTX1=0
    FOR I=1 TO 4
        DATO1=140
        DATO2=0
        DATO3=DATO3-15
        DATO4=DATO4+15
        DATO7=DATO7-15
        DATO8=DATO8+15
        DRIV2DAT1=DRIV2DAT1+15
        DRIV2DAT2=DRIV2DAT2-15
        LEDTX1=1
        GOSUB TRANSM1
        gosub transm2
        LEDTX1=0
    NEXT
    PAUSE 1000
    FOR I=1 TO 4
        DATO1=140
        DATO2=0
        DATO3=DATO3+15

```

```

    DATO4=DATO4-15
    DRIV2DAT1=DRIV2DAT1-15
    DRIV2DAT2=DRIV2DAT2+15
    LEDTX1=1
    GOSUB TRANSM1
    gosub transm2
    LEDTX1=0
NEXT
ENDIF
IF PORTD=8 THEN
    AY=0
    CLEARWDT
    DATO1=90
    DATO2=90
    DATO3=90
    DRIV2DAT1=90
    DRIV2DAT2=90
    LEDTX1=1
    GOSUB TRANSM1
    gosub transm2
FOR I=1 TO 9
    DATO7=DATO7-10
    DATO8=DATO8+10
    LEDTX1=1
    GOSUB TRANSM1
    LEDTX1=0
next
    DATO4=DATO4-15
    DATO7=0
    dato8=180
    LEDTX1=1
    GOSUB TRANSM1
    LEDTX1=0
    FOR I=1 TO 9
        DATO7=DATO7+10
        DATO8=DATO8-10
        LEDTX1=1
        GOSUB TRANSM1
        LEDTX1=0
    next
    DATO7=90
    dato8=90
    LEDTX1=1
    GOSUB TRANSM1
    LEDTX1=0
    FOR I=1 TO 9
        DATO5=DATO5-10
        DATO6=DATO6+10
        LEDTX1=1
        GOSUB TRANSM1
        LEDTX1=0
    next
    DATO5=0
    DATO6=180
    LEDTX1=1

```


GOSUB TRANSM1	DATO3=90
LEDTX1=0	DATO4=90
FOR I=1 TO 9	FOR I=1 TO 9
DATO5=DATO5+10	DATO1=DATO1-10
DATO6=DATO6-10	DATO2=DATO2+10
LEDTX1=1	LEDTX1=1
GOSUB TRANSM1	GOSUB TRANSM1
LEDTX1=0	LEDTX1=0
next	next
DATO5=90	DATO1=0
DATO6=90	DATO2=180
FOR I=1 TO 9	FOR I=1 TO 9
DATO3=DATO3-10	DATO1=DATO1+10
DATO4=DATO4+10	DATO2=DATO2-10
LEDTX1=1	LEDTX1=1
GOSUB TRANSM1	GOSUB TRANSM1
LEDTX1=0	LEDTX1=0
next	next
DATO3=0	DATO1=90
DATO4=180	DATO2=90
FOR I=1 TO 9	FOR I=1 TO 9
DATO3=DATO3+10	DATO3=DATO3-10
DATO4=DATO4-10	DATO4=DATO4+10
LEDTX1=1	LEDTX1=1
GOSUB TRANSM1	GOSUB TRANSM1
LEDTX1=0	LEDTX1=0
next	next

```

DATO3=0
DATO4=180
FOR I=1 TO 9
    DATO3=DATO3+10
    DATO4=DATO4-10
    LEDTX1=1
    GOSUB TRANSM1
    LEDTX1=0
next
    DATO3=90
    DATO4=90
FOR I=1 TO 9
    DATO5=DATO5-10
    DATO6=DATO6+10
    LEDTX1=1
    GOSUB TRANSM1
    LEDTX1=0
next
    DATO5=0
    DATO6=180
    LEDTX1=1
    GOSUB TRANSM1
    LEDTX1=0
FOR I=1 TO 9
    DATO5=DATO5+10
    DATO6=DATO6-10
    LEDTX1=1
GOSUB TRANSM1
    LEDTX1=0
    GOSUB TRANSM1
    LEDTX1=0
    ENDIF
    IF PORTD=5 THEN
        CLEARWDT
        DRIV2DAT1=90
        DRIV2DAT2=90
GOSUB TRANSM1
    LEDTX1=0
    DATO5=90
    DATO6=90
    LEDTX1=1
    GOSUB TRANSM1
    LEDTX1=0
next
    DATO7=0
    dato8=180
    LEDTX1=1
    GOSUB TRANSM1
    LEDTX1=0
    DATO7=90
    dato8=90
    LEDTX1=1
    GOSUB TRANSM1
    LEDTX1=0
    ENDIF
    IF PORTD=5 THEN
        CLEARWDT
        DRIV2DAT1=90
        DRIV2DAT2=90

```

LEDTX1=1	LEDTX1=0
gosub transm2	DRIV2DAT1=50
LEDTX1=0	DRIV2DAT2=50
DATO1=90	LEDTX1=1
DATO2=90	gosub transm2
DATO3=90	LEDTX1=0
DATO8=90	FOR I=1 TO 5
LEDTX1=1	DATO1=DATO1+4
GOSUB TRANSM1	DATO2=DATO2+4
LEDTX1=0	DATO3=DATO3-4
FOR I=1 TO 5	DATO4=DATO4-4
DATO1=DATO1-4	DATO5=DATO5+4
DATO2=DATO2-4	DATO6=DATO6+4
DATO3=DATO3+4	DATO7=DATO7-4
DATO4=DATO4+4	DATO8=DATO8-4
DATO5=DATO5-4	LEDTX1=1
DATO6=DATO6-4	GOSUB TRANSM1
DATO7=DATO7+4	LEDTX1=0
DATO8=DATO8+4	next
LEDTX1=1	DRIV2DAT1=70
GOSUB TRANSM1	DRIV2DAT2=70
LEDTX1=0	LEDTX1=1
next	gosub transm2
DRIV2DAT1=70	LEDTX1=0
DRIV2DAT2=70	FOR I=1 TO 5
LEDTX1=1	DATO1=DATO1+4
gosub transm2	DATO2=DATO2+4

DATO3=DATO3-4	next
DATO4=DATO4-4	DRIV2DAT1=110
LEDTX1=1	DRIV2DAT2=110
GOSUB TRANSM1	LEDTX1=1
LEDTX1=0	gosub transm2
next	LEDTX1=0
DRIV2DAT1=90	DRIV2DAT1=130
DRIV2DAT2=90	DRIV2DAT2=130
LEDTX1=1	LEDTX1=1
gosub transm2	gosub transm2
LEDTX1=0	LEDTX1=0
DATO1=90	FOR I=1 TO 5
DATO2=90	DATO1=DATO1-4
DATO3=90	DATO2=DATO2-4
DATO4=90	DATO3=DATO3+4
LEDTX1=1	DATO4=DATO4+4
GOSUB TRANSM1	LEDTX1=1
LEDTX1=0	GOSUB TRANSM1
FOR I=1 TO 5	LEDTX1=0
DATO1=DATO1+4	next
DATO2=DATO2+4	DRIV2DAT1=110
DATO3=DATO3-4	DRIV2DAT2=110
DATO4=DATO4-4	LEDTX1=1
DATO5=DATO5+4	gosub transm2
LEDTX1=1	LEDTX1=0
GOSUB TRANSM1	ENDIF
LEDTX1=0	IF PORTD=4 THEN

```

CLEARWDT
DRIV2DAT1=90
DRIV2DAT2=90
LEDTX1=1
gosub transm2
LEDTX1=0
DATO1=130
DATO2=0
DATO3=90
DATO4=90
DATO5=90
LEDTX1=1
GOSUB TRANSM1
LEDTX1=0
gosub ANALOG
FOR I=1 TO 10
    DATO1=DATO1-4
    DATO2=DATO2-4
    DATO3=DATO3+4
    DATO4=DATO4+4
    DATO5=DATO5-4
    LEDTX1=1
    GOSUB TRANSM1
    LEDTX1=0
next
DRIV2DAT1=70
DRIV2DAT2=90
LEDTX1=1
gosub transm2
LEDTX1=0
gosub ANALOG
FOR I=1 TO 10
    DATO1=DATO1+4
    ;DATO2=DATO2+4
    DATO3=DATO3-4
    DATO4=DATO4-4
    DATO5=DATO5+4
    DATO6=DATO6+4

```

```

    DATO7=DATO7-4
    DATO8=DATO8-4
    LEDTX1=1
    GOSUB TRANSM1
    LEDTX1=0
next
    DRIV2DAT1=70
    DRIV2DAT2=90
    LEDTX1=1
    gosub transm2
    LEDTX1=0
    gosub ANALOG
    FOR I=1 TO 10
        DATO1=DATO1+4
        DATO2=DATO2+4
        DATO3=DATO3-4
        DATO4=DATO4-4
        DATO5=DATO5+4
        LEDTX1=1
        GOSUB TRANSM1
        LEDTX1=0
    next
    gosub ANALOG
    DRIV2DAT1=90
    DRIV2DAT2=90
    LEDTX1=1
    gosub transm2
    LEDTX1=0
    DATO1=130
    DATO2=0
    DATO3=90
    DATO4=90
    DATO5=90
    LEDTX1=1
    GOSUB TRANSM1
    LEDTX1=0
    ENDIF
    IF PORTD=3 THEN
        FOR K=1 TO 5
            DATO1=140
            DATO2=0
            DATO3=90
            DATO4=90
            DRIV2DAT1=90
            DRIV2DAT2=90
            LEDTX1=1
            GOSUB TRANSM1
            gosub transm2
            LEDTX1=0
        FOR I=1 TO 2
            DATO2=DATO2+10
            LEDTX1=1
            GOSUB TRANSM1
            gosub transm2
        
```

```

    LEDTX1=0
NEXT
NEXT
FOR K=1 TO 1
    CLEARWDT
    DATO1=90
    DATO2=90
    DATO3=90
    DATO4=90
    DRIV2DAT1=90
    DRIV2DAT2=90
    LEDTX1=1
    GOSUB TRANSM1
    gosub transm2
    LEDTX1=0
DRIV2DAT1=30
    DRIV2DAT2=150
    LEDTX1=1
    gosub transm2
    LEDTX1=0
    DRIV2DAT1=90
    DRIV2DAT2=90
    LEDTX1=1
    gosub transm2
    LEDTX1=0
FOR I=1 TO 9
    DATO7=DATO7-10
    DATO8=DATO8+10
    LEDTX1=1
    GOSUB TRANSM1
    LEDTX1=0
next
    DATO7=0
    dato8=180
    LEDTX1=1
    GOSUB TRANSM1
    LEDTX1=0
    FOR I=1 TO 9
        DATO7=DATO7+10
        DATO8=DATO8-10
        LEDTX1=1
        GOSUB TRANSM1
        LEDTX1=0
    next
    DATO7=90
    dato8=90
    LEDTX1=1
    GOSUB TRANSM1
    LEDTX1=0
    FOR I=1 TO 9
        DATO5=DATO5-10
        DATO6=DATO6+10
        LEDTX1=1
        GOSUB TRANSM1

```

LEDTX1=0	DATO4=DATO4-10
next	LEDTX1=1
DATO5=0	GOSUB TRANSM1
DATO6=180	LEDTX1=0
LEDTX1=1	next
GOSUB TRANSM1	DATO3=90
LEDTX1=0	DATO4=90
FOR I=1 TO 9	FOR I=1 TO 9
DATO5=DATO5+10	DATO1=DATO1-10
DATO6=DATO6-10	DATO2=DATO2+10
LEDTX1=1	LEDTX1=1
GOSUB TRANSM1	GOSUB TRANSM1
LEDTX1=0	LEDTX1=0
next	next
DATO5=90	DATO1=0
DATO6=90	DATO2=180
FOR I=1 TO 9	FOR I=1 TO 9
DATO3=DATO3-10	DATO1=DATO1+10
DATO4=DATO4+10	DATO2=DATO2-10
LEDTX1=1	LEDTX1=1
GOSUB TRANSM1	GOSUB TRANSM1
LEDTX1=0	LEDTX1=0
next	next
DATO3=0	DATO1=90
DATO4=180	DATO2=90
FOR I=1 TO 9	FOR I=1 TO 9
DATO3=DATO3+10	DATO3=DATO3-10


```

    DATO4=DATO4+10
    LEDTX1=1
    GOSUB TRANSM1
    LEDTX1=0
next
    DATO3=0
    DATO4=180
    FOR I=1 TO 9
        DATO3=DATO3+10
        DATO4=DATO4-10
        LEDTX1=1
        GOSUB TRANSM1
        LEDTX1=0
    next
    DATO3=90
    DATO4=90
    FOR I=1 TO 9
        DATO5=DATO5-10
        DATO6=DATO6+10
        LEDTX1=1
        GOSUB TRANSM1
        LEDTX1=0
    next
    DATO5=0
    DATO6=180
    LEDTX1=1
    GOSUB TRANSM1
    DATO5=DATO5+10
    DATO6=DATO6-10
    LEDTX1=1
    GOSUB TRANSM1
    LEDTX1=0
next
    DATO5=90
    DATO6=90
    FOR I=1 TO 9
        DATO7=DATO7-10
        DATO8=DATO8+10
        LEDTX1=1
        GOSUB TRANSM1
        LEDTX1=0
    next
    DATO7=0
    dato8=180
    LEDTX1=1
    GOSUB TRANSM1
    LEDTX1=0
    FOR I=1 TO 9
        DATO7=DATO7+10
        DATO8=DATO8-10
        LEDTX1=1
        GOSUB TRANSM1

```

```

    LEDTX1=0
next
    DATO7=90
    dato8=90
    LEDTX1=1
    GOSUB TRANSM1
    LEDTX1=0
    DRIV2DAT1=30
    DRIV2DAT2=150
    LEDTX1=1
    gosub transm2
    LEDTX1=0
    DRIV2DAT1=90
    DRIV2DAT2=90
    LEDTX1=1
    gosub transm2
    LEDTX1=0
FOR K=1 TO 1
    DATO1=90
    DATO2=90
    DATO3=90
    DATO4=90
    DRIV2DAT1=90
    DRIV2DAT2=90
    LEDTX1=1
    GOSUB TRANSM1
    gosub transm2
    LEDTX1=0
    FOR I=1 TO 9
        DATO2=DATO2-10
        LEDTX1=1
        GOSUB TRANSM1
        PAUSE 50
        LEDTX1=0
    NEXT
    FOR I=1 TO 9
        DATO3=DATO3-10
        LEDTX1=1
        GOSUB TRANSM1
        PAUSE 50
        LEDTX1=0
    NEXT
    FOR I=1 TO 9
        DATO2=DATO2+10
        LEDTX1=1
        GOSUB TRANSM1
        PAUSE 50
        LEDTX1=0
    NEXT
    FOR I=1 TO 4
        DATO1=DATO1-10
        LEDTX1=1
        GOSUB TRANSM1
        PAUSE 50
    NEXT

```

```

    LEDTX1=0
NEXT
FOR I=1 TO 9
    DATO4=DATO4-10
    LEDTX1=1
    GOSUB TRANSM1
    PAUSE 50
    LEDTX1=0
NEXT
DATO4=0
LEDTX1=1
GOSUB TRANSM1
LEDTX1=0
pause 100
FOR I=1 TO 9
    DATO2=DATO2+10
    DATO1=DATO1+10
    LEDTX1=1
    GOSUB TRANSM1
    PAUSE 50
    LEDTX1=0
NEXT
FOR I=1 TO 4
    DATO1=DATO1+10
    LEDTX1=1
    GOSUB TRANSM1
    PAUSE 50

```

```

    LEDTX1=0
NEXT
FOR J=1 TO 3
    FOR I=1 TO 4
        DATO6=DATO6-15
        DATO7=DATO7-15
        DATO8=DATO8+15
        DRIV2DAT1=DRIV2DAT1+15
        DRIV2DAT2=DRIV2DAT2-15
        LEDTX1=1
        GOSUB TRANSM1
        gosub transm2
        LEDTX1=0
    NEXT
    PAUSE 200
    FOR I=1 TO 4
        DATO6=DATO6+15
        DATO7=DATO7+15
        DATO8=DATO8-15
        DRIV2DAT1=DRIV2DAT1-15
        DRIV2DAT2=DRIV2DAT2+15
        LEDTX1=1
        GOSUB TRANSM1
        gosub transm2
        LEDTX1=0
    NEXT
    pause 200

```

NEXT	DATO1=140
FOR I=1 TO 15	DATO2=0
DATO4=DATO4+6	DATO3=90
LEDTX1=1	DATO4=90
GOSUB TRANSM1	DRIV2DAT1=90
PAUSE 50	DRIV2DAT2=90
LEDTX1=0	LEDTX1=1
NEXT	GOSUB TRANSM1
FOR I=1 TO 15	gosub transm2
DATO1=DATO1-6	LEDTX1=0
LEDTX1=1	FOR I=1 TO 4
GOSUB TRANSM1	DATO1=140
PAUSE 50	DATO2=0
LEDTX1=0	DATO3=DATO3-15
NEXT	DATO4=DATO4+15
FOR I=1 TO 15	DATO5=DATO5+15
DATO1=DATO1-6	DATO6=DATO6-15
DATO2=DATO2-6	DATO7=DATO7-15
DATO3=DATO3+6	DATO8=DATO8+15
LEDTX1=1	DRIV2DAT1=DRIV2DAT1+15
GOSUB TRANSM1	DRIV2DAT2=DRIV2DAT2-15
PAUSE 50	LEDTX1=1
LEDTX1=0	GOSUB TRANSM1
NEXT	gosub transm2
NEXT	LEDTX1=0
FOR K=1 TO 2	NEXT
CLEARWDT	PAUSE 1000

```

FOR I=1 TO 4
    DATO1=140
    DATO2=0
    DATO3=DATO3+15
    DATO4=DATO4-15
    DATO5=DATO5-15
    DATO6=DATO6+15
    DATO7=DATO7+15
    DATO8=DATO8-15
    DRIV2DAT1=DRIV2DAT1-15
    DRIV2DAT2=DRIV2DAT2+15
    LEDTX1=1
    GOSUB TRANSM1
    gosub transm2
    LEDTX1=0
NEXT
next
FOR K=1 TO 2
    DATO1=140
    DATO2=0
    DATO3=90
    DATO4=90
    DATO5=90
    DRIV2DAT1=90
    DRIV2DAT2=90
    LEDTX1=1
    GOSUB TRANSM1
    gosub transm2
    LEDTX1=0
    FOR I=1 TO 4
        DATO1=140
        DATO2=0
        DATO3=DATO3-15
        DATO4=DATO4+15
    
```

```

DATO5=DATO5+15
DATO6=DATO6-15
DATO7=DATO7-15
DATO8=DATO8+15
DRIV2DAT1=DRIV2DAT1+15
DRIV2DAT2=DRIV2DAT2-15
LEDTX1=1
GOSUB TRANSM1
  gosub transm2
LEDTX1=0
NEXT
PAUSE 1000
FOR I=1 TO 4
  DATO1=140
  DATO2=0
  DATO3=DATO3+15
  DATO4=DATO4-15
  DATO5=DATO5-15
  DATO6=DATO6+15
  DATO7=DATO7+15
  DATO8=DATO8-15
  DRIV2DAT1=DRIV2DAT1-15
  DRIV2DAT2=DRIV2DAT2+15
  LEDTX1=1
  GOSUB TRANSM1
  gosub transm2
LEDTX1=0
NEXT
NEXT
NEXT
FOR K=1 TO 1
  DATO1=90
  DATO2=90
  DATO3=90
  DATO4=90
  DRIV2DAT1=90
  DRIV2DAT2=90
  LEDTX1=1
  GOSUB TRANSM1
  gosub transm2
  LEDTX1=0
  FOR I=1 TO 9
    DATO2=DATO2-10
    LEDTX1=1
    GOSUB TRANSM1
    PAUSE 50
    LEDTX1=0
  NEXT
  FOR I=1 TO 9
    DATO3=DATO3-10
    LEDTX1=1
    GOSUB TRANSM1
    PAUSE 50
    LEDTX1=0
  NEXT
NEXT

```

FOR I=1 TO 9	DATO2=DATO2+10
DATO2=DATO2+10	DATO1=DATO1+10
LEDTX1=1	LEDTX1=1
GOSUB TRANSM1	GOSUB TRANSM1
PAUSE 50	PAUSE 50
LEDTX1=0	LEDTX1=0
NEXT	NEXT
FOR I=1 TO 4	FOR I=1 TO 4
DATO1=DATO1-10	DATO1=DATO1+10
LEDTX1=1	LEDTX1=1
GOSUB TRANSM1	GOSUB TRANSM1
PAUSE 50	PAUSE 50
LEDTX1=0	LEDTX1=0
NEXT	NEXT
FOR I=1 TO 9	FOR J=1 TO 3
DATO4=DATO4-10	FOR I=1 TO 4
LEDTX1=1	DATO6=DATO6-15
GOSUB TRANSM1	DATO7=DATO7-15
PAUSE 50	DATO8=DATO8+15
LEDTX1=0	DRIV2DAT1=DRIV2DAT1+15
NEXT	DRIV2DAT2=DRIV2DAT2-15
DATO4=0	LEDTX1=1
LEDTX1=1	GOSUB TRANSM1
GOSUB TRANSM1	gosub transm2
LEDTX1=0	LEDTX1=0
pause 100	NEXT
FOR I=1 TO 9	PAUSE 200

FOR I=1 TO 4	FOR I=1 TO 15
DATO6=DATO6+15	DATO1=DATO1-6
DATO7=DATO7+15	DATO2=DATO2-6
DATO8=DATO8-15	DATO3=DATO3+6
DRIV2DAT1=DRIV2DAT1-15	LEDTX1=1
DRIV2DAT2=DRIV2DAT2+15	GOSUB TRANSM1
LEDTX1=1	PAUSE 50
GOSUB TRANSM1	LEDTX1=0
gosub transm2	NEXT
LEDTX1=0	next
NEXT	FOR K=1 TO 1
pause 200	CLEARWDT
NEXT	DATO1=90
FOR I=1 TO 15	DATO2=90
DATO4=DATO4+6	DATO3=90
LEDTX1=1	DATO4=90
GOSUB TRANSM1	DRIV2DAT1=90
PAUSE 50	DRIV2DAT2=90
LEDTX1=0	LEDTX1=1
NEXT	GOSUB TRANSM1
FOR I=1 TO 15	gosub transm2
DATO1=DATO1-6	LEDTX1=0
LEDTX1=1	DRIV2DAT1=30
GOSUB TRANSM1	DRIV2DAT2=150
PAUSE 50	LEDTX1=1
LEDTX1=0	gosub transm2
NEXT	LEDTX1=0


```

FOR I=1 TO 9
    DATO7=DATO7-10
    DATO8=DATO8+10
    LEDTX1=1
    GOSUB TRANSM1
    LEDTX1=0
next
DATO7=0
dato8=180
LEDTX1=1
GOSUB TRANSM1
LEDTX1=0
FOR I=1 TO 9
    DATO7=DATO7+10
    DATO8=DATO8-10
    LEDTX1=1
    GOSUB TRANSM1
    LEDTX1=0
next
DATO7=90
dato8=90
LEDTX1=1
GOSUB TRANSM1
LEDTX1=0
FOR I=1 TO 9
    DATO5=DATO5-10
    DATO6=DATO6+10
LEDTX1=1
GOSUB TRANSM1
LEDTX1=0
next
DATO3=DATO3-10
DATO4=DATO4+10
LEDTX1=1
GOSUB TRANSM1
LEDTX1=0
next
DATO3=0
DATO4=180

```

```
FOR I=1 TO 9
    DATO3=DATO3+10
    DATO4=DATO4-10
    LEDTX1=1
    GOSUB TRANSM1
    LEDTX1=0
```

next

DATO3=90

DATO4=90

```
FOR I=1 TO 9
```

DATO1=DATO1-10

DATO2=DATO2+10

LEDTX1=1

GOSUB TRANSM1

LEDTX1=0

next

DATO1=0

DATO2=180

```
FOR I=1 TO 9
```

DATO1=DATO1+10

DATO2=DATO2-10

LEDTX1=1

GOSUB TRANSM1

LEDTX1=0

next

DATO1=90

DATO2=90

```
FOR I=1 TO 9
```

DATO3=DATO3-10

DATO4=DATO4+10

LEDTX1=1

GOSUB TRANSM1

LEDTX1=0

next

DATO3=0

DATO4=180

```
FOR I=1 TO 9
```

DATO3=DATO3+10

DATO4=DATO4-10

LEDTX1=1

GOSUB TRANSM1

LEDTX1=0

next

DATO3=90

DATO4=90

```
FOR I=1 TO 9
```

DATO5=DATO5+10

DATO6=DATO6-10

LEDTX1=1

GOSUB TRANSM1

LEDTX1=0

next

DATO5=90

DATO6=90

```

FOR I=1 TO 9
    DATO7=DATO7-10
    DATO8=DATO8+10
    LEDTX1=1
    GOSUB TRANSM1
    LEDTX1=0
next
DATO7=0
dato8=180
LEDTX1=1
GOSUB TRANSM1
LEDTX1=0
FOR I=1 TO 9
    DATO7=DATO7+10
    DATO8=DATO8-10
    LEDTX1=1
    GOSUB TRANSM1
    LEDTX1=0
next
DATO7=90
dato8=90
LEDTX1=1
GOSUB TRANSM1
LEDTX1=0
FOR I=1 TO 15
    DRIV2DAT1=30
    DRIV2DAT2=150
    LEDTX1=1
    gosub transm2
    LEDTX1=0
    DRIV2DAT1=90
    DRIV2DAT2=90
    LEDTX1=1
    gosub transm2
    LEDTX1=0
NEXT
FOR K=1 TO 2
    DATO1=140
    DATO2=0
    DATO3=90
    DATO4=90
    DRIV2DAT1=90
    DRIV2DAT2=90
    LEDTX1=1
    GOSUB TRANSM1
    gosub transm2
    LEDTX1=0
FOR I=1 TO 2
    DATO2=DATO2+10
    LEDTX1=1
    GOSUB TRANSM1
    gosub transm2
    LEDTX1=0
NEXT

```

```

NEXT
DATO1=90
DATO2=90
DATO3=90
DATO4=90
DRIV2DAT1=90
DRIV2DAT2=90
LEDTX1=1
GOSUB TRANSM1
  gosub transm2
LEDTX1=0
  PAUSE 5000
ENDIF
if PORTD=2 then
  CLEARWDT
  DATO1=140:DATO2=0:DATO3=90:DATO4=90
  :DATO5=90:DATO6=90:DATO7=90:DATO8
  =90:DRIV2DAT1=90:DRIV2DAT2=90:LEDTX
  1=1
  GOSUB TRANSM1:
  gosub transm2:LEDTX1=0
  FOR I=1 TO 2
    DATO2=DATO2+10:LEDTX1=1:
  GOSUB TRANSM1
    gosub transm2:LEDTX1=0
  NEXT
endif

```

```

GOTO INICIO
OJOS :
  OJO1=1:OJO2=1
  PAUSE 200
  OJO1=0:OJO2=0
  RETURN
ANALOG:
  ADCIN 0, DATOS
  PAUSE 100
  IF (DATOS>=45)&(DATOS<=74) Then
    LEDRX=1
  DATO1=130:DATO2=90:DATO3=0:DATO4=9
  0:DATO5=90:DATO6=90:DATO7=90:DATO8
  =90:DATO1=160:LEDTX1=1:GOSUB
  TRANSM1: LEDTX1=0
  DRIV2DAT1=90:DRIV2DAT2=90:LEDTX1=1
  FOR J=1 TO 10
    FOR I=1 TO 10
      DATO2=DATO2-4: DATO4=DATO4+4
      DATO5=DATO5-4:DATO6=DATO6-4:
      DATO7=DATO7+4DATO8=DATO8+4:
      LEDTX1=1:GOSUB TRANSM1:LEDTX1=0
    NEXT
  DRIV2DAT1=70:DRIV2DAT2=90:LEDTX1=1
  ;DRIV2DAT2=70
  gosub transm2:LEDTX1=0

```

```

FOR I=1 TO 10
DATO2=DATO2-4:DATO4=DATO4+4
DATO5=DATO5-4:DATO6=DATO6-4:
DATO7=DATO7+4
DATO8=DATO8+4:LEDTX1=1
GOSUB TRANSM1:LEDTX1=0
    next
DRIV2DAT1=10:DRIV2DAT2=90:LEDTX1=1
    gosub transm2:LEDTX1=0
DRIV2DAT1=70:DRIV2DAT2=90:LEDTX1=1
    gosub transm2:LEDTX1=0
DRIV2DAT1=90:DRIV2DAT2=90:LEDTX1=1
    gosub          transm2:LEDTX1=0
DATO1=130:DATO2=90:DATO3=0:DATO4=9
0:DATO5=90
    DATO6=90:DATO7=90:DATO8=90
    LEDTX1=1:GOSUB TRANSM1
    LEDTX1=0
next
DATO1=90:DATO2=0:DATO3=90:LEDTX1=1
    GOSUB TRANSM1:LEDTX1=0:
pause 300:
DATOS=40:LEDRX=0
    ENDIF
RETURN TRANSM1:
HSerOUT[DATO1,DATO2,DATO3,DATO4,D
ATO5,DATO6,DATO7,DATO8]
    RETURN
TRANSM2:
    SEROUT
TX2,T2400,[DRIV2DAT1]',DRIV2DAT2,DRIV
2DAT3,DRIV2DAT4,DRIV2DAT5,DRIV2DAT
6,DRIV2DAT7,DRIV2DAT8]
    PAUSE 50
    SEROUT TX2,T2400,[DRIV2DAT2]
    PAUSE 50
    SEROUT TX2,T2400,[DRIV2DAT3]
    PAUSE 50
    SEROUT TX2,T2400,[DRIV2DAT4]
    PAUSE 50
    SEROUT TX2,T2400,[DRIV2DAT5]
    PAUSE 50
    SEROUT TX2,T2400,[DRIV2DAT6]
    PAUSE 50
    SEROUT TX2,T2400,[DRIV2DAT7]
    PAUSE 50
    SEROUT TX2,T2400,[DRIV2DAT8]
    PAUSE 50
    RETURN
END

```