

UNIVERSIDAD NACIONAL DE INGENIERÍA

FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA



**“DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE
TELEMANDO Y AUTOMATIZACIÓN DE BOMBEO DE
AGUA, BASADO EN MICROCONTROLADORES PIC”**

TESIS

PARA OPTAR EL TÍTULO PROFESIONAL DE:

INGENIERO ELECTRÓNICO

**PRESENTADO POR:
JAVIER CAMILO POMA PALACIOS**

PROMOCIÓN
1994-I

**LIMA-PERÚ
2003**

*A Dios, a mis padres, a Gloria y Anthony.
Fuentes inagotables de apoyo e inspiración.*

**DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE
TELEMANDO Y AUTOMATIZACIÓN DE BOMBEO DE
AGUA, BASADO EN MICROCONTROLADORES PIC**

SUMARIO

El abastecimiento de agua tanto para el campamento minero, como para el uso en la explotación minera se realiza con un sistema de bombeo (ver figura 7.1), que se encuentra a unos 12 Km. del campamento de administración. El funcionamiento del sistema de bombeo de agua, presentaba el inconveniente: (1) Llegado el horario o según requerimiento, un operario lo controla manualmente; (2) En horarios establecidos, se retrasa unos 25 minutos, debido a que encendida la electrobomba 1, el operario se dirige al segundo, que esta a una distancia de 1000 m., tardando aproximadamente 25 minutos en llegar y encender la electrobomba 2 y (3) Según requerimiento, primero se tenía que avisar al operario, retrasando unos 25 minutos más su funcionamiento. Todo esto causaba molestias, retrasaba el proceso de explotación minera y aumentaba su costo de explotación.

El inconveniente (1) fue superado al implementar el **Circuito de Control de Nivel**, que se encarga de automatizar el sistema de bombeo de agua y añadiendo un reloj calendario digital, fue superado el inconveniente (2). El inconveniente (3) fue superado implementando el **Circuito de Telemando**, que vía línea telefónica realiza el control ON/OFF del Circuito de Control de Nivel. También se tuvo que implementar Circuitos de Protección (Compatibilidad Electromagnética).

Instalado, calibrado y probado todo el sistema, este funciona normalmente teniendo presente las recomendaciones de funcionamiento.

ÍNDICE

PRÓLOGO	1
CAPÍTULO I	
SISTEMAS DE CONTROL Y AUTOMATIZACIÓN	4
1.1. Aspectos generales	4
1.2. Sistemas de control y automatización	5
1.2.1. Configuración de sistemas de control automático	6
1.2.2. Sistemas lineales y no lineales	8
1.2.3. Sistemas variantes e invariantes con el tiempo	9
1.2.4. Sistemas de control en tiempo continuo y en tiempo discreto	9
1.3. El control y la automatización industrial	10
1.3.1. Sistema automatizado	12
1.3.2. Técnicas de automatización	13
1.4. El microcontrolador en el control y la automatización	15
1.5. Las técnicas de telemando	16
1.5.1. Transmisión, recepción y acción	17
1.5.2. Tipos de telemando	18
1.5.3. Alcance y seguridad	19
1.6. Telemando por teléfono	21
1.6.1. Nociones de telefonía	22

CAPÍTULO II

SISTEMAS DE PROTECCIÓN, DISPOSITIVOS Y HERRAMIENTAS	26
2.1. Introducción	26
2.2. Sistemas de protección contra descargas atmosféricas, interferencia electromagnética y tensiones inducidas	27
2.2.1. Gap, bobina de choque, fusible, condensador, varistor y TVS	30
2.2.2. Filtros EMI	36
2.2.3. Sistema de puesta a tierra y pararrayo	38
2.3. Principales dispositivos utilizados en el sistema	40
2.3.1. Transductor de presión	40
2.3.2. Amplificador de instrumentación y convertidor A/D	43
2.3.3. EEPROM serial	46
2.3.4. Visualizador LCD	47
2.3.5. Bombas de agua, tipos y funcionamiento	49
2.3.6. Microcontrolador y sus principales características	51
2.4. Herramientas de diseño y simulación	56

CAPÍTULO III

EL MICROCONTROLADOR PIC16C55	62
3.1. Descripción general	62
3.2. Arquitectura	63
3.2.1. Asignación de pines	65
3.2.2. Reloj/Ciclo de instrucción	66
3.2.3. Flujo de instrucción/Segmentación	67

3.3.	Organización de la memoria	68
3.3.1.	Memoria de programa	68
3.3.2.	Memoria de datos	70
3.4.	Puertos de entrada/salida	74
3.4.1.	Puerto A	74
3.4.2.	Puerto B	74
3.4.3.	Puerto C	74
3.4.4.	Instrucción TRIS f	75
3.5.	Temporizador/Contador TMR0	75
3.6.	Características especiales	78
3.6.1.	Palabra de configuración	78
3.6.2.	Oscilador	79
3.6.3.	Reset	80
3.6.4.	Temporizador perro guardián	82
3.6.5.	Modo SLEEP o de bajo consumo	83
3.7.	Repertorio de instrucciones	84
3.8.	Características eléctricas	87

CAPÍTULO IV

EL MICROCONTROLADOR PIC16F876	88	
4.1.	Descripción general	88
4.2.	Arquitectura	90
4.2.1.	CPU	92
4.2.2.	Asignación de pines	93

4.2.3.	Reloj/Ciclo de instrucción	95
4.2.4.	Flujo de instrucción/Segmentación	96
4.3.	Organización de la memoria	97
4.3.1.	Memoria de programa	97
4.3.2.	Memoria de datos	100
4.4.	Memoria de programa FLASH y de datos EEPROM	107
4.5.	Puertos de entrada/salida	110
4.5.1.	Puerto A	110
4.5.2.	Puerto B	111
4.5.3.	Puerto C	112
4.6.	Temporizador/Contador (TMR0, TMR1, TMR2)	113
4.6.1.	Temporizador/Contador TMR0	113
4.6.2.	Temporizador/Contador TMR1	115
4.6.3.	Temporizador/Contador TMR2	117
4.7.	Módulo CCP	120
4.7.1.	Modo Captura	121
4.7.2.	Modo Comparación	122
4.7.3.	Modo de modulación de anchura de pulsos	123
4.8.	Módulo MSSP	125
4.8.1.	Modo SPI	126
4.8.2.	Modo I ² C	128
4.9.	Módulo USART	132
4.10.	Módulo convertidor A/D	133
4.10.1.	Registros de trabajo	134

4.10.2. Estructura interna y configuración del convertidor A/D	137
4.11. Recursos especiales del procesador	138
4.11.1. Palabra de configuración	138
4.11.2. Oscilador	139
4.11.3. Palabras de identificación	141
4.11.4. Reset	141
4.11.5. Interrupciones	144
4.11.6. Temporizador perro guardián	146
4.11.7. Modo reposo o SLEEP	146
4.11.8. Programación serial en circuito (ICSP)	148
4.11.9. Programación con voltaje bajo (LVP)	148
4.11.10. Depuración en circuito	148
4.12. Repertorio de instrucciones	149
4.13. Características eléctricas	151

CAPÍTULO V

HERRAMIENTAS DE DESARROLLO PARA LOS PIC	153
5.1. Editor de código fuente	153
5.2. Ensamblador	154
5.2.1. Vinculador MPLINK	155
5.2.2. Librería MPLIB	156
5.3. Simulador	157
5.4. Emulador	158
5.5. Programador	159

5.6.	Entorno de desarrollo integral	159
5.6.1.	MPLAB–IDE	159
5.6.2.	Herramientas de desarrollo de MPLAB–IDE	162

CAPÍTULO VI

	INGENIERÍA DEL SISTEMA IMPLEMENTADO	165
6.1.	Hardware del sistema	165
6.1.1.	Circuito fuente de alimentación	165
6.1.2.	Circuito de Telemando	169
6.1.3.	Circuito de Control de Nivel	178
6.2.	Software del sistema	189
6.2.1.	Para el PIC16C55	189
6.2.2.	Para el PIC16F876	204

CAPÍTULO VII

	DESCRIPCIÓN FUNCIONAL DEL SISTEMA IMPLEMENTADO	259
7.1.	Introducción	259
7.2.	Funcionamiento del sistema	261
7.2.1.	Funcionamiento manual	263
7.2.2.	Funcionamiento automático	263
7.3.	Especificaciones técnicas	267
7.4.	Manual del usuario	268

CAPÍTULO VIII**COSTOS Y PRESUPUESTO DEL SISTEMA IMPLEMENTADO 274**

8.1. Costos 274

8.2. Presupuesto 278

CONCLUSIONES 279**ANEXO A****M-8870 - DTMF RECEIVER 281****ANEXO B****CONSTATANDO SI FUNCIONAN LOS CIRCUITOS****IMPLEMENTADOS 288****BIBLIOGRAFÍA 293**

PRÓLOGO

Los microcontroladores están conquistando el mundo. Están presentes en nuestro trabajo, en nuestra casa y en nuestra vida, en general. El siglo XXI será testigo de la conquista masiva de estos diminutos computadores, que gobernarán la mayor parte de los aparatos que fabricamos y usamos los humanos. Las extensas áreas de aplicación de los microcontroladores, que pueden considerarse ilimitadas, exigen un gigantesco trabajo de diseño y fabricación.

El presente trabajo tiene el propósito de poner al alcance de los estudiantes, información amplia sobre los microcontroladores de Microchip de la gama baja (PIC16C55) y de la gama media (PIC16F876); así como sus herramientas de hardware y software que permitan la comprensión y el desarrollo de aplicaciones con estos microcontroladores, que sirven como punto de partida para continuar con microcontroladores superiores. Además, poner al alcance de la industria en general circuitos implementados en campo: (1) Circuito de protección contra interferencias electromagnéticas para alimentación de hasta 440 V A.C y de entradas/salidas de control; que pueden ser utilizados en nuestra serranía. (2) Circuito de Telemando, que vía línea telefónica puede ser utilizado para realizar control ON/OFF sin tener en cuenta la distancia y (3) Circuito de Control de Nivel, que se configura vía pulsadores, sensor no presenta contacto físico con el líquido y para un nivel de hasta 8,50 metros; puede ser utilizado en cualquier sistema que requiere control de nivel.

El sistema diseñado es esencialmente, un sistema digital basado en microprocesador orientado al control, que utiliza los microcontroladores PIC16C55 y PIC16F876, con receptor DTMF, sensor de presión, indicadores led's, salidas con relé, pulsadores y pantalla LCD. La fuente de alimentación; así como las entradas o salidas de control, presentan protección contra interferencias electromagnéticas. El sistema desarrollado presenta ciertas limitaciones, no críticas, tales como: (1) Enviada la señal de telemando para el funcionamiento de las electrobombas no hay forma de verificarlo, debido a que la señal de mando no podría llegar a su destino por el mal estado del cable o por que el sistema de protección se encuentre dañado. (2) No hay forma de saber si las electrobombas se encuentran trabajando realmente, aún estando en funcionamiento y (3) Producido cualquier desperfecto en la electrobomba el sistema no puede avisar de este hecho ni tampoco cancelar su funcionamiento.

El presente trabajo ha sido posible gracias al apoyo de personas e instituciones; las cuales paso a mencionar. A mis padres, que dieron todo de sí en provecho mío. A mi esposa e hijo, por soportarme durante todo este tiempo. A la Universidad Nacional de Ingeniería, por haberme instruido en esta fascinante especialidad. A la Unidad de Producción Julcani, por su decidido apoyo. Finalmente, mi agradecimiento de todo corazón a todas aquellas personas, que de una u otra forma, hicieron posible se realice este proyecto.

CAPÍTULO I

SISTEMAS DE CONTROL Y AUTOMATIZACIÓN

1.1. Aspectos generales

Los primeros sistemas de control se desarrollaron con la revolución industrial de fines del siglo XIX y principios del siglo XX. Al principio, se basaron casi exclusivamente en componentes mecánicos y electromecánicos, básicamente en engranajes, palancas, relés y pequeños motores. Toda la función reside en el cableado de los elementos que componen el sistema, por lo tanto una modificación exige nuevos componentes, cambios de cableado, trabajos de montaje y de soldadura y, por su puesto incremento económico, todo esto sin contar las pérdidas de tiempo y dinero en las pruebas y puesta a punto.

A partir de los años cincuenta empezaron a emplearse los semiconductores, que permitían el diseño de sistemas de menor tamaño y consumo, más rápidos y con menor desgaste. En la década de los sesenta, la complejidad y las prestaciones de los sistemas de control se incrementaron gracias al empleo de circuitos integrados y en particular los de tipo programable (sistemas basados en microprocesadores). Al tiempo que se desarrollaban los circuitos integrados lo hacían también los ordenadores digitales, si bien su empleo en la industria quedaba restringido al control de procesos muy complejos, debido a su elevado costo, necesidad de personal especializado para su instalación, manejo y la poca facilidad de interconexión (interfaz) con el proceso, donde se manejan habitualmente tensiones y corrientes

fuertes, para los cuales no suele estar preparado el ordenador. La demanda en la industria de un sistema, económico, robusto, flexible, fácilmente modificable y con mayor facilidad para tratar con tensiones y corrientes fuertes que la que tenía el ordenador, hizo que se desarrollasen los autómatas programables industriales o controladores lógicos programables (PLC).

En los últimos años los ordenadores compatibles PC están ampliando su presencia en el mercado de aplicaciones industriales. Con la mejora de sus características mecánicas, eléctricas y electromagnéticas, los compatibles compiten con otras alternativas de control digital, como las estaciones de trabajo de plataforma no PC y los sistemas modulares organizados en torno a un bus, en aplicaciones de control de procesos, supervisión y captura de datos.

1.2. Sistemas de control y automatización

El concepto de **control** es extraordinariamente amplio, abarca desde un simple interruptor que gobierna el encendido de una lámpara o la válvula que regula el paso de agua en una tubería, hasta el más complejo ordenador de proceso o el piloto automático de un avión.

Un sistema de control se encarga de gobernar la respuesta de un proceso (planta), sin que el operador intervenga directamente sobre sus elementos de salida. Dicho operador manipula únicamente las magnitudes denominadas consignas (órdenes) y el sistema de control se encarga de gobernar dicha salida a través de los accionamientos. El concepto lleva de alguna forma implícita que el sistema de control opera, en general, con magnitudes de baja potencia, llamadas genéricamente señales, y gobierna unos accionamientos que se encargan de controlar la planta.

La **automatización** es la sustitución de la acción humana por mecanismos, independientes o no entre sí, movidos por una fuente de energía exterior, capaces de realizar ciclos completos de operaciones que se pueden repetir indefinidamente. Un sistema automático supone siempre la existencia de una fuente de energía, de unos órganos de mando, que son los que ordenan el ciclo a realizar, y de unos órganos de trabajo, que son los que lo ejecutan.

Según el grado de automatización puede hablarse de dos niveles: (1) La automatización completa, se prefiere en la producción masiva de productos homogéneos en ciclo continuo (proceso de llenado de botellas, fármacos, ensamblaje, producción de piezas, etc.) y (2) La automatización parcial, que es propia de la producción variable y limitada. Desde el punto de vista de la programación, la automatización puede considerarse de ciclo fijo y de ciclo programado. El primer caso es adecuado para la fabricación de grandes series porque el automatismo es invariable. El segundo caso se orienta a la fabricación de piezas distintas, en series pequeñas y medias porque el dispositivo programador de que dispone el sistema puede ordenar el ciclo que convenga.

1.2.1. Configuración de sistemas de control automático

Existen dos configuraciones básicas para realizar el control automático de la planta:

A. Control en lazo abierto: Está se caracteriza porque la información o variables que controlan el proceso circulan en una sola dirección, desde el sistema de control al proceso (ver figura 1.1). El sistema de control no recibe la confirmación de que las acciones que a través de los actuadores ha de realizar sobre el proceso, se han ejecutado correctamente.

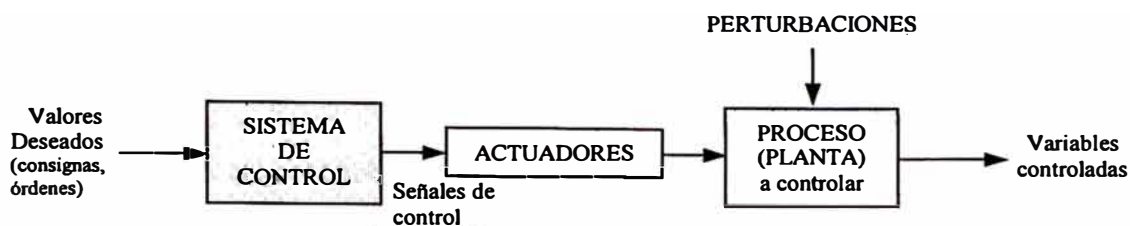


Figura 1.1 *Diagrama de bloques de un sistema en lazo abierto.*

B. Control en lazo cerrado: Se caracteriza por que existe una realimentación a través de los sensores desde el proceso hacia el sistema de control, que permite a este conocer si las acciones ordenadas a los actuadores se han realizado correctamente sobre el proceso. La figura 1.2 muestra su diagrama de bloques.

La mayoría de los procesos existentes en la industria utilizan el control en lazo cerrado; por que el producto que se pretende obtener o las variables que se controlan necesitan un control continuo en función de unos determinados parámetros de entrada, o bien porque el proceso a controlar se subdivide en una serie de acciones elementales de tal forma que, para realizar una determinada acción sobre el proceso, es necesario que previamente se hayan realizado otra serie de acciones elementales.

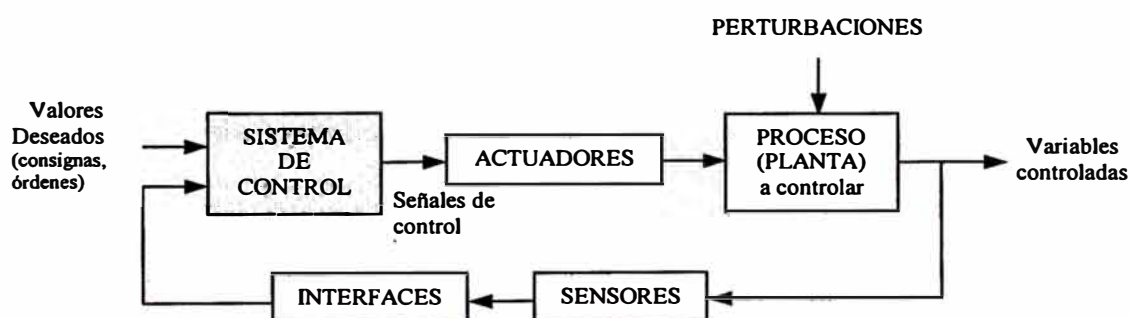


Figura 1.2 *Diagrama de bloques de un sistema en lazo cerrado.*

Los sistemas realimentados se pueden clasificar en diversas formas, dependiendo del propósito de la clasificación. Por ejemplo, de acuerdo con el método

de análisis y diseño, los sistemas de control se clasifican en **lineales y no lineales, variantes con el tiempo o invariantes con el tiempo**. De acuerdo con los tipos de señales usados en el sistema, se tiene sistemas en **tiempo continuo** y en **tiempo discreto**. A menudo, los sistemas de control se clasifican de acuerdo al propósito.

1.2.2. Sistemas lineales y no lineales

Estrictamente hablando, los sistemas lineales no existen en la práctica, ya que todos los sistemas físicos son no lineales en algún grado. Los sistemas de control realimentados son modelos ideales utilizados por el analista para simplificar el análisis y diseño.

Cuando las magnitudes de las señales en un sistema de control están limitadas en intervalos en los cuales los componentes del sistema exhiben una característica lineal (aplicable principio de superposición), el sistema es esencialmente lineal. Pero cuando las magnitudes de las señales se extienden más allá del intervalo de porción lineal, dependiendo de la severidad de la no linealidad, el sistema no se debe seguir considerando lineal.

Para sistemas lineales, existe una gran cantidad de técnicas analíticas y gráficas para fines de diseño y análisis. Por otro lado, los sistemas no lineales son difíciles de tratar en forma matemática, y no existen métodos generales disponibles para resolver una gran variedad de clases de sistemas no lineales. En el diseño de sistemas de control, es práctico, primero diseñar el controlador con base en un modelo de un sistema lineal despreciando las no linealidades del sistema. Entonces, el controlador diseñado se aplica al modelo del sistema no lineal para su evaluación o rediseño mediante simulación en computadora.

1.2.3. Sistemas variantes e invariantes con el tiempo

Cuando los parámetros del sistema de control son estacionarios con respecto al tiempo durante la operación del sistema, se denomina sistema **invariante con el tiempo**. En la práctica, la mayoría de los sistemas físicos contienen elementos que derivan o varían con el tiempo. Por ejemplo el sistema de control automático de un misil guiado en el cual la masa del misil decrece a medida que el combustible a bordo se consume durante el vuelo. Otro ejemplo, la resistencia de la bobina de un motor eléctrico variará cuando el motor es excitado por primera vez y su temperatura esta aumentando. A estos sistemas se denominan **variantes con el tiempo**; el análisis y diseño de esta clase de sistemas son mucho más complejos que los de un **sistema lineal invariante con el tiempo**.

1.2.4. Sistemas de control en tiempo continuo y en tiempo discreto

Un **sistema de control en tiempo continuo** es aquel en que las señales en varias partes del sistema son todas funciones de la variable de tiempo continuo t . Estos sistemas se pueden describir mediante ecuaciones diferenciales. Para el análisis y síntesis de estos sistemas se utiliza la herramienta matemática de la transformada de Laplace, esta transforma las ecuaciones diferenciales lineales invariantes en el tiempo en ecuaciones algebraicas en s .

Los **sistemas de control en tiempo discreto** son aquellos sistemas en los cuales una o más de las variables pueden cambiar sólo en valores discretos de tiempo. Los instantes, pueden especificar los tiempos en los que se lleva acabo alguna medición de tipo físico o los tiempos en los que se extraen los datos de la memoria de un computador digital.

Los sistemas de control en tiempo discreto difieren de los sistemas de control en tiempo continuo en que las señales en uno o más puntos del sistema, están en la forma de datos muestreados o en la forma digital. Si el sistema digital involucra una computadora digital como controlador, los datos muestreados se deben convertir a datos digitales.

Una herramienta matemática muy utilizada en el análisis y la síntesis de sistemas de control en tiempo discreto es la transformada z . El papel de la transformada z en sistemas en tiempo discreto es similar al de la transformada de Laplace en sistemas en tiempo continuo. En un sistema de control en tiempo discreto, una ecuación en diferencias lineal caracteriza la dinámica del sistema. Para determinar la respuesta del sistema a una entrada dada, se debe resolver dicha ecuación en diferencias. Con el método de la transformada z , las soluciones a las ecuaciones en diferencias se convierten en un problema de naturaleza algebraica.

1.3. El control y la automatización industrial

En el contexto actual, la **automática** se define como la Ciencia y Técnica de la automatización, que agrupa el conjunto de las disciplinas teóricas y tecnológicas que intervienen en la concepción, la construcción y el empleo de los sistemas automáticos. La automática constituye el aspecto teórico de la cibernética. Está estrechamente vinculada con las matemáticas, la estadística, la teoría de la información, la informática y técnicas de la ingeniería.

Podría realizarse una distinción entre: la **automática teórica**, conjunto de los métodos matemáticos de análisis y de síntesis de los sistemas automáticos y de sus elementos; y la **automática aplicada**, que trata más específicamente de los

problemas prácticos de automatización, que concierne a la teoría y a la tecnología de los captadores, los accionadores y los ordenadores.

El grado de automatización de un proceso viene determinado fundamentalmente por factores de tipo económico y tecnológico, por ello podemos encontrar una gama muy amplia y variada, dependiendo de los objetivos a alcanzar.

A. Nivel elemental: Se corresponde con el asignado a una máquina sencilla o parte de una máquina, asignándole tareas de vigilancia de tiempos muertos, posicionamiento de piezas y funciones de Seguridad.

B. Nivel intermedio: Se corresponde con la explotación de un conjunto de máquinas elementales o bien una máquina compleja. Este ha sido el dominio clásico de la automatización industrial.

C. Tercer nivel: Se caracterizan por ser de un proceso completo, e intervienen además del control elemental del proceso, otros aspectos tales como Supervisión, Optimización, Gestión de Mantenimiento, Control de Calidad, Seguimiento de la Producción.

D. Cuarto nivel: Se corresponde con el concepto de Fabricación Integrada por Computador, CIM (*Computer Integrated Manufacturing*), donde se contempla la inclusión de forma integrada a la Producción, conceptos tales como la Gestión Empresarial, Planificación, Programación etc.

En años recientes se ha incrementado el uso de controladores digitales en sistemas de control. Los controladores digitales se utilizan para alcanzar el desempeño óptimo, por ejemplo, en la forma de productividad máxima, beneficio máximo, costo mínimo o la utilización mínima de energía. Recientemente, la aplicación de control por computadora ha hecho posible el movimiento inteligente de

robots industriales, la optimización de economía de combustible en automóviles y el refinamiento en la operación de enseres y maquinas de uso domestico. La capacidad en la toma de decisiones y la flexibilidad en los programas de control son las mayores ventajas de los sistemas de control digital.

1.3.1. Sistema automatizado

La estructura de un Sistema Automatizado puede clasificarse en dos partes claramente diferenciadas: por un lado lo que denominaremos **Parte Operativa**, formada por un conjunto de dispositivos, máquinas o subprocesos, diseñados para la realización de determinadas funciones; de forma específica pueden tratarse de máquinas herramienta para la realización de operaciones de mecanizado más o menos sofisticadas o bien de subprocesos dedicados a otras tareas. Por otro lado tenemos la **Parte de Control o Mando**, que, independientemente de su implementación tecnológica: eléctrica, electrónica, neumática, hidráulica, etc., es el dispositivo encargado de realizar la coordinación de las distintas operaciones encaminadas a mantener a la Parte Operativa bajo control.

El sometimiento de la Parte Operativa se logra mediante el mantenimiento continuo de un intercambio de información entre la primera y la Parte de Control o mando. Dicho intercambio se establece a través de los captadores binarios, transductores analógicos y digitales y los dispositivos de accionamiento. A partir de los dos primeros se recoge información de los valores de las magnitudes físicas a controlar, así como de sus cambios de estado, enviando dicha información a la Parte de Control para su tratamiento. Tras el tratamiento de la información se envían acciones de mando a través de los accionadores. Los accionadores son dispositivos

que permiten el control de grandes potencias mediante las señales de pequeña potencia que son emitidas por la Parte de Control.

En suma, la automatización de un proceso industrial, (máquina, conjunto o equipo industrial) consiste en la incorporación al mismo de un conjunto de elementos y dispositivos tecnológicos que aseguren su control y buen comportamiento.

Uno de los objetivos deseables es que el automatismo sea capaz de reaccionar frente a las situaciones previstas de antemano y también que, frente a imponderables, tenga como objetivo situar al proceso y a los recursos humanos que lo asisten en la situación más favorable.

1.3.2. Técnicas de automatización

Según la naturaleza del automatismo empleado puede hablarse de automatización mecánica, neumática, hidráulica, eléctrica y electrónica.

Los sistemas de **automatización mecánica** suelen ser complicados, por la abundancia de mecanismos y de escasa flexibilidad. Por el contrario, la tecnología que regula su funcionamiento es relativamente accesible al personal poco calificado, lo que se traduce en un montaje y mantenimiento económico. Los mecanismos que los componen son ruedas dentadas y poleas para transmisiones del movimiento de biela-manivela, piñón-cremallera, etc., para la conversión del movimiento rectilíneo en circular y viceversa; levas y palancas para la obtención de recorridos controlados, etc. Existe gran variedad de automatismos mecánicos en la industria: desde las máquinas herramientas (tornos, fresadoras, limadoras, etc.), hasta los relojes mecánicos, pasando por los telares, motores de combustión interna y toda la maquinaria que formó parte de la revolución industrial.

La técnica de **automatización neumática** admite infinidad de aplicaciones en el campo de la máquina herramienta, especialmente en los trabajos de fijación de piezas, bloqueo de órganos, alimentación de máquinas y movimiento lineal de órganos que no requieran velocidades de actuación rigurosamente constantes. Prácticamente la totalidad de las automatizaciones industriales tienen, como elementos de mando, instalaciones neumáticas. Como principales características del mando neumático cabe destacar: La sencillez de los propios sistemas de mando (cilindros, válvulas, etc.); la rapidez de movimiento (respuesta) del sistema neumático; la economía una vez instalado; el mantenimiento del estado del aire, ya que debe mantenerse perfectamente limpio y seco.

Prácticamente lo dicho para la automatización neumática vale para la **automatización hidráulica**, aunque con algunas diferencias: por ejemplo, el mando hidráulico es más lento que el neumático, sin embargo, es capaz de desarrollar más trabajo. La hidráulica se prefiere en sistemas que deban desarrollar más trabajo y no sea primordial la velocidad de respuesta. Este tipo de mando se encuentra en prensas, diversas máquinas herramientas, y por supuesto, en el automóvil: frenos, dirección e incluso suspensión.

Se debe tener en cuenta, que cualquier máquina, por sencilla que sea, va a tener algún tipo de **automatización eléctrica**, encargado de gobernar los motores o como función de mando dentro del propio sistema (interruptores, pulsadores, relés, contactores, temporizadores, electroválvulas, electrobombas, seccionadores, etc.).

Por su puesto, la llegada de la electrónica a la industria ha supuesto una verdadera revolución. La **automatización electrónica** ha permitido que la automatización industrial de un paso de gigante. La base de este avance en la

automatización ha sido los sistemas digitales, que han desembocado en el computador, o en otros sistemas de automatización basados en microprocesadores y naturalmente en los PLC's.

1.4. El microcontrolador en el control y la automatización

El número de productos que funcionan procesando, gestionando y controlando en base a uno o varios microcontroladores aumenta de forma exponencial. No es aventurado pronosticar que el presente siglo habrá pocos circuitos electrónicos que carezcan de un microcontrolador.

La industria informática acapara gran parte de los microcontroladores, casi todos los periféricos del computador: mouse, teclado, fax/módem, tarjeta de vídeo, tarjeta de sonido, tarjeta de red, escáner, impresora, etc.

En los electrodomésticos de la línea blanca: lavadoras, horno de microondas, lavavajillas, etc.; en la línea marrón: televisores, videos, equipos musicales. Igualmente en los grandes edificios o residencias: ascensores, aire acondicionado y calefacción; así mismo el control de acceso, vigilancia y seguridad: identificación del código de barras, banda magnética, huella digital o retina ocular, activa las respectivas alarmas.

Las comunicaciones: telefonía, TV y transmisión de datos: controlador de comunicaciones, microcentral, central telefónica etc. y el moderno teléfono.

En la electromedicina: electrocardiógrafos, electroencefalógrafos, resonancia magnética, ecógrafos, endoscopio, cardiología, etc.

Para la producción de hortalizas y frutas, creando microclimas adecuados (invernaderos) controla temperatura, humedad, radiación solar, ventilación, etc.

Últimamente en automoción: la climatización, seguridad, encendido, combustión, y la seguridad de los frenos están gobernados por estos pequeños computadores.

Diferentes equipos de medición, de calibración y test; equipos emuladores, grabadores, etc. para el desarrollo de la electrónica.

Y en general en la industria, minería, agroindustria, alimenticia, pesquera, petroquímica, envasado/etiquetado, cervecería, textil, impresión gráfica, etc.

Irónicamente los microcontroladores han automatizado también la guerra, permitiendo el desarrollo de misiles guiados, satélites espías y armas cada vez mas sofisticados de una mortal precisión.

1.5. Las técnicas de telemando

El término telemando, se utiliza para designar cualquier evento de los sistemas de mando a distancia con los que se puede imaginar. Aquí trataremos únicamente los sistemas electrónicos de telemando, y más concretamente por la transmisión, con o sin hilos de las órdenes a ejecutar. La ejecución propiamente dicha, generalmente usa técnicas muy diferentes de la electrónica: eléctrica, mecánica, electromecánica, hidráulica, neumática, etc.

Prácticamente, cualquier dispositivo susceptible de ser conducido en modo local puede ser igualmente telemandado. Las técnicas de transmisión de las órdenes son siempre aproximadamente las mismas, mientras que la interfaz con el elemento mandado constituye, casi en cada ocasión, un caso particular.

Incluso si los mismos circuitos pueden servir en dos casos, será infinitamente más fácil el encendido y apagado de luces, que la apertura y el cierre de una puerta.

1.5.1. Transmisión, recepción y acción

Todo sistema de telemando cuenta con tres funciones básicas: una vez elaborada la orden apropiada, se transmite por un medio de transmisión capaz de hacerlo llegar hasta el lugar deseado. Allí, un sistema de recepción identifica la orden, y la dirige al dispositivo capaz de ejecutar la acción deseada.

En la práctica, veremos que si a la transmisión por medio de ondas electromagnéticas es irremplazable en ciertos casos particulares, a menudo es preferible sustituirla por otras técnicas, al menos de la misma calidad y que presenten ventajas especiales.

Los medios de transmisión se pueden clasificar en dos tipos: medio material y medio inmaterial.

Transmisiones por medio material: Rigurosamente hablando, se puede calificar de medio de transmisión material cualquier ligadura palpable entre dos lugares: cable eléctrico, y fibra óptica. Sería un grave error pensar que la realización de un telemando por hilos eléctricos obliga necesariamente a instalar cables especiales; a menudo es posible aprovechar instalaciones o redes ya existentes.

Los hilos de la red eléctrica, que pasan un poco por todas partes en los locales más diversos, recogen gustosamente señales de telemando como suplemento de la energía que transportan. A mayor escala, se utiliza este principio para controlar a distancia los contadores de doble tarifa instalados en los hogares de ciertos abonados.

Por su parte, la red telefónica internacional permite establecer, casi instantáneamente, un enlace de buena calidad entre dos puntos casi cualesquiera del planeta. Cuando no se trata más que de transmitir una orden de telemando muy breve; ello no cuesta más que un poco de dinero, sea cual sea la distancia.

Por otro lado, la fibra óptica no está todavía muy expandida entre los aficionados, pero esto cambiará. En efecto, la finalidad de los cabellos de vidrio es transportar ondas luminosas cargadas de información, por recorridos completamente diferentes a la famosa línea recta. Los progresos que se han producido debido a las necesidades de las telecomunicaciones modernas hacen que este medio de transmisión sea cada vez más accesible al aficionado. Una fibra óptica es, considerablemente, más fina que los más discretos cables eléctricos, y por ello no les afectan de ninguna forma los problemas de parásitos electromagnéticos. Además, es un aislante eléctrico prácticamente perfecto.

Transmisiones por medio inmaterial: Se entiende por medio inmaterial, cualquier vía de comunicación impalpable; dicho como los pioneros de la radio denominaban poéticamente es **el éter**.

Lógicamente, las ondas de radio pueden muy bien transportar toda clase de informaciones de telemando, a corta, media y larga distancia. Las frecuencias radioeléctricas son cada vez más raras de encontrar; sin embargo, se han desarrollado medios concurrentes para ciertas aplicaciones.

Actualmente, estas técnicas (sobre todo los infrarrojos) están destronando resueltamente a la radio en gran número de aplicaciones. Ahora, están disponibles a precios asequibles componentes de calidad, a pesar de sus prestaciones elevadas.

1.5.2. Tipos de telemando

Telemando todo o nada: Cada vez más, la electrónica evoluciona hacia las técnicas numéricas, digitales, o todo o nada: los circuitos operan con unos y ceros, es decir, pura y simplemente con presencias o ausencias de corriente. Diferentes artificios

permiten resolver casi todos los problemas prácticos, que utilizan al máximo estos procedimientos.

En telemando, la mayoría de las aplicaciones (aparte del modelismo) necesita una acción “todo o nada”: puesta en marcha o parada, apertura o cierre, encendido o apagado. Incluso, cuando es necesario un ajuste (volumen sonoro de la televisión, por ejemplo) es fácil actuar por puntos, dos órdenes distintas que permitan, para todo o nada, avanzar o retrasar un punto.

Telemando proporcional: Las aplicaciones de telemandos proporcionales que permiten *ajustar* a distancia una temperatura, una velocidad, o una posición con toda la agudeza deseable, conciernen más a los usuarios industriales que a los aficionados, salvo evidentemente a los diseñadores.

Los telemandos llamados **proporcionales** funcionan por transmisión de pulsos todo o nada, pero que se suceden a un ritmo rápido. No obstante, son posibles algunas aplicaciones, como la orientación a distancia de una antena de radio, TV o satélite. Pueden utilizarse técnicas especializadas, denominadas analógicas, pero los sistemas de transmisión todo o nada pueden también resolver el problema: son la naturaleza y el contenido de las órdenes transmitidas, lo que deberá adaptarse a estas aplicaciones particulares.

1.5.3. Alcance y seguridad

Un sistema de telemando puede tener un radio de acción que vaya desde algunos centímetros (mando de cerradura) a miles de kilómetros. Cada aplicación impone sus propios imperativos en materia de alcance, lo que evidentemente influye en la elección del modo de transmisión que se utilizará. Al mismo tiempo, intervienen

otros criterios: consumo de los equipos de transmisión y recepción, costo del material, costo de cada transmisión de orden, facilidad de utilización, instalación fija o móvil, enlace en vía directa o trabada por obstáculos, etc.

Técnicamente hablando, el problema del alcance apenas es crítico con los medios materiales: basta con colocar o utilizar la longitud de cable deseada. El límite es más bien de presupuesto: un cable largo es caro, lo mismo que una comunicación telefónica lejana de alguna duración.

A la inversa, con un medio inmaterial todo o casi todo se trata sobre el plano técnico: potencia del transmisor, sensibilidad del receptor, condiciones de propagación, despeje de las antenas o de los elementos ópticos, etc.

La organización de un telemando de largo alcance exigirá mucho cuidado y precisión en los ajustes; en resumen, un equipo de laboratorio de calidad y buenos conocimientos técnicos.

La seguridad de un sistema de telemando recae sobre dos puntos fundamentales:

- La insensibilidad a las falsas puestas en funcionamiento, a las consecuencias que puedan ir desde el disgusto al accidente grave.
- La fiabilidad de las puestas en funcionamiento controladas: aunque un rechazo de puesta en funcionamiento sea menos grave que una acción inoportuna, este tipo de defecto puede ser molesto.

Todas las diferentes técnicas susceptibles de resolver un problema dado no ofrecen la misma seguridad: por tanto, hay que hacer una selección crítica. En general, la "alta seguridad" se paga con una contrapartida: mayor complejidad, coste superior, alcance reducido, pérdida de reacción, dimensiones, etc. Por tanto, se deberá encontrar un compromiso en cada caso concreto.

La seguridad, y en cierta medida, la discreción de un telex pasan por una *codificación* de las órdenes transmitidas. Codificar una transmisión puede complicar bastante su contenido para que el riesgo de generación espontánea de un mensaje significativo sea casi nulo. Otra ventaja es que un pirata tendrá grandes dificultades para analizar y reproducir precisamente un mensaje lo suficientemente complejo.

Los procedimientos más eficaces descansan en el empleo de técnicas numéricas: un circuito apropiado (codificador) compone un largo “telegrama” formado por unos y ceros colocados en determinado orden. Este mensaje se transmite mediante un procedimiento cualquiera a un circuito decodificador que sólo autorizará la ejecución de la orden si recibe una o varias veces seguidas el telegrama correcto. La transmisión puede ser, quizás, muy rápida, con el fin de que este proceso no ralentice sensiblemente la ejecución de la orden.

Otra ventaja del procedimiento es que varios mensajes diferentes pueden corresponder a otras tantas acciones independientes (telex multicanal), o a diferentes receptores (domicilio y oficina, por ejemplo). En el mercado existe toda una variedad de circuitos integrados especialmente concebidos para este uso y ampliamente utilizados en las realizaciones profesionales.

1.6. Telex por teléfono

A pesar de la proliferación de nuevos servicios y de las aplicaciones telemáticas muchos abonados telefónicos apenas imaginan las posibilidades latentes de su línea. Es evidente que con la tecnología de Internet estas posibilidades son mucho más amplias. Cada abonado telefónico está directamente conectado mediante dos hilos, al que se ha calificado como la más enorme máquina jamás construida por el hombre:

la red telefónica conmutada mundial. Algunas cifras marcadas en el teclado basta para que en unos pocos segundos se establezca una conexión de audio en el rango de 300 a 3400 Hz con cualquier punto del globo, incluidos automóviles, aviones y barcos. Una vasta red de cabinas públicas permiten estar siempre cerca de un punto de acceso telefónico.

La modernización de la red, equipos equipados con la tecnología *Dual Tone Multifrequency* (DTMF), facilita aún más las operaciones de telegando. Tasadas según la duración, las comunicaciones automáticas no resultan caras cuando basta algunos segundos para comunicar una acción de mando a distancia y verificar su ejecución.

1.6.1. Nociones de telefonía

Las comunicaciones telefónicas tienen su origen en el siglo XIX. Alexander Graham Bell, en 1876, inventa el teléfono y, aunque tuvo una aceptación tardía, pronto se convierte en el medio de comunicación por excelencia.

A. Red Telefónica Conmutada

Se trata de una red de conmutación de circuitos, que permite la comunicación entre dos usuarios, si previamente se ha establecido una conexión entre ambos. Los servicios que ofrece, por lo general, es el de voz entre dispositivos o terminales telefónicos. Estos terminales telefónicos están conectados a centrales de terminación de la red, también denominadas centrales locales.

Para establecer la comunicación entre centrales de terminación o locales, existen centrales de conmutación, que configuran una Red Telefónica de Conmutación

(RTC). La Red conmutada se compone por distintas centrales, desde la central local. A modo de nodo de conmutación las centrales van realizando la comunicación por medio del circuito. Unas centrales hacen de conexión entre las mismas, su única función es la de conmutación y supervisión interna en la red conmutada. Actualmente disponemos tanto de redes analógicas, y redes digitales.

Por la aplicación de las técnicas digitales, es posible la transmisión de textos, imágenes, además de la voz.

Los terminales domésticos están conectados a la central más próxima, y desde allí tienen acceso, para su conexión, bien con los abonados de la misma central terminal, o a los del resto de la red conmutada.

Existen, por tanto, dos componentes globales de las redes telefónicas:

- Red de abonados o local, en la que se incluyen todos los abonados dependientes de la central terminal o local.
- Red de enlaces o centrales, circuitos de conmutación que unen las centrales entre sí. Para ello utilizan diferentes medios de transmisión, sobre todo el medio de transmisión que permite la mayor velocidad de transmisión posible actualmente; la fibra óptica.

B. Proceso de la comunicación telefónica

El proceso de la comunicación tiene tres partes bien diferenciadas:

- Señalización y establecimiento de la conexión.
- Comunicación e intercambio de la información entre origen y destino.
- Fin de la comunicación.

En primer lugar se marca un grupo de dígitos, el número del abonado destinatario de nuestra llamada. Las centrales locales intercambian unas señales. Estas permiten el establecimiento del circuito sobre el que viaja la comunicación. Además la comunicación, se encamina junto con otras, que tengan como destino la central local del destinatario, u otra muy próxima.

Finalizado el establecimiento, tiene lugar la comunicación propiamente dicha. La voz se recoge por un auricular, y si se trata de una red digital, es convertida en información digital desde el propio terminal. En el caso de redes analógicas, la voz se convierte en digital en la central local, ya que la red conmutada actualmente es totalmente de transmisión digital.

Finalizada la comunicación, cuando cuelga el terminal que realizó la llamada, se libera el camino que se estableció entre ambos. Así queda disponible para otra posible comunicación.

En reposo, una línea telefónica presenta, entre sus dos hilos, una tensión continua de 48 a 50 voltios de los que uno de los polos está a tierra. Para que timbre el teléfono, la central superpone una tensión alterna de alrededor de 80 V de 50 Hz a esta alimentación continua. El timbrado obedece a un periodo de 8 segundos que está formado de la siguiente manera: 2 segundos de timbrado por 4 segundos de pausa, siendo la empresa operadora la que fija el número máximo de repeticiones.

Cuando se descuelga el aparato (o cuando un sistema automático toma la línea), un circuito adecuado consume, aproximadamente, una corriente de 35 mA en la línea. La central clasifica a línea como ocupada. Si se descuelga mientras que el teléfono suena, entonces se establece la comunicación y comienza la tasación. En caso contrario, se obtiene el tono de invitación a marcar (440 Hz). Este se transforma

en el tono de ocupación si no se marca ningún número en veinte segundos (línea clasificada como falsa llamada).

Los teclados modernos son de tipo multifrecuencia DTMF. A cada tecla le corresponde un par de frecuencias, como se muestra en la figura 1.3. Contrariamente a los impulsos decimales, el destinatario puede recibir, en el curso de la comunicación, estas señales del teclado si usted lo acciona. Así, se puede ahorrar un aparato de acoplamiento acústico para muchas aplicaciones en el ámbito del telemando. Las teclas de la última columna usualmente no son usadas.

Columna 0 1209 Hz	Columna 1 1336 Hz	Columna 2 1477 Hz	Columna 3 1633 Hz	
1	2	3	A	Fila 0 697 Hz
4	5	6	B	Fila 1 770 Hz
7	8	9	C	Fila 2 852 Hz
*	0	#	D	Fila 3 941 Hz

Figura 1.3 Frecuencias de salida estándar para un teclado DTMF.

Por último, cuando se establece la comunicación, es preciso que la inyección y la extracción de las señales en línea se haga con una impedancia de 600Ω con una potencia bastante inferior al milivatio, y en la banda de 300 a 3400 Hz. La línea debe estar aislada galvánicamente de los circuitos que se la conectan.

CAPÍTULO II

SISTEMAS DE PROTECCIÓN, DISPOSITIVOS Y HERRAMIENTAS

2.1. Introducción

El incremento de la utilización de equipos electrónicos en la industria trae consigo un aumento de la exposición de diminutos dispositivos electrónicos a las perturbaciones eléctricas. Cada vez más, el control electrónico reemplaza a los controles mecánicos y eléctricos aunque en muchas ocasiones, nos olvidamos de proteger a esos equipos tan sofisticados de las descargas eléctricas producida por los fenómenos naturales (descargas atmosféricas) o por la propia calidad de la línea eléctrica de que disponemos.

Las telecomunicaciones, informática, equipos médicos y sistemas electrónicos en general nos ofrecen una mayor calidad de vida, pero también nos ocasionan grandes perjuicios económicos cuando sufren la veloz visita de una sobre tensión eléctrica, presentándose efectos primarios y secundarios.

Efectos primarios

- Daños materiales causados por la destrucción de las instalaciones, componentes eléctricos y electrónicos, pistas de los circuitos impresos y equipos de telecomunicaciones.
- Envejecimiento prematuro y fallos inexplicables de los equipos electrónicos.
- Daños de hardware y software informático (bloqueo de programas).

Efectos secundarios

Pérdidas económicas como consecuencias de la inoperatividad de los sistemas de comunicación, redes informáticas, sistemas de control y automatización, o cualquier otro sistema dentro de la cadena de explotación o producción.

La rapidez con la que viaja y se propaga el pico de sobretensión (de 10 a 1000 veces más voltaje y amperaje que lo habitual) es tal, que ningún elemento no destinado a tal efecto puede salvarse; uno, varios o todos los microprocesadores que llevan incorporados, una placa de ordenador, una central telefónica, o cualquier máquina industrial. La sobretensión transitoria es destructiva sin remisión.

2.2. Sistemas de protección contra descargas atmosféricas, interferencia electromagnética y tensiones inducidas

El buen funcionamiento de un sistema eléctrico/electrónico, donde se combinan dispositivos de potencia que producen perturbaciones y elementos o circuitos de pequeña señal, sensibles a dichas perturbaciones, está basado en el cumplimiento de ciertas reglas de diseño e instalación que permitan hacer compatibles los niveles de señal de unos con los de perturbación de otros.

El estudio de la problemática general de generación, propagación, influencia sobre otros circuitos y medidas de protección contra las perturbaciones electromagnéticas se agrupa bajo el título genérico de **Compatibilidad Electromagnética**, abreviado por las iniciales EMC. Cuando el efecto de tales perturbaciones llegue a alterar el funcionamiento de un determinado dispositivo o circuito, diremos que se produce Interferencia Electromagnética, abreviado con las iniciales EMI.

El origen de las interferencias en los equipos industriales suele ser la presencia o proximidad de cierto tipo de receptores eléctricos tales como relés, contactores, equipos de tiristores, conmutadores rápidos, etc. Dichos dispositivos producen perturbaciones en la red y generan en sus proximidades campos eléctricos y magnéticos fuertes que se acoplan a los sistemas electrónicos, induciendo en ellos señales parásitas que perturban su buen funcionamiento o pueden llegar incluso a destruir ciertos componentes de los mismos. En cuanto a las formas de propagación son esencialmente tres:

1. **Conducción:** Propagación a través de conductores, ya sea de la red de alimentación, de las entradas y salidas del sistema o de cables de comunicación. En especial deben controlarse los conductores de alimentación y el conductor de retorno o común de señales de entrada.
2. **Acoplamiento capacitivo o inductivo (campos próximos).** En este caso, el medio de propagación son los campos eléctrico y magnético generados por conductores de potencia próximos a conductores de señal. Para distancia menor que $\lambda/4$.
3. **Radiación (campos lejanos).** El medio de propagación son los campos electromagnéticos provocados por descargas atmosféricas, descargas electrostáticas, conductores más alejados o por transmisores de RF. Para distancia mayor que $\lambda/4$.

La tabla 2.1 muestra en forma resumida los posibles orígenes de las perturbaciones, los medios de propagación y las posibles víctimas de la interferencia. La problemática de las interferencias ha despertado un enorme interés en numerosos organismos internacionales de normalización por su trascendencia sobre el buen funcionamiento de numerosos equipos. En la tabla 2.2 se puede observar un resumen

de las normas internacionales más importantes en distintos campos. De todas ellas, y refiriéndonos al campo de instalación de autómatas Cabe destacar las IEC-65, IEC-801 e IEC-1000, con sus distintas partes.

FUENTES DE PERTURBACIÓN	MEDIOS DE PROPAGACIÓN	RECEPTORES O VÍCTIMAS
NATURALES Descargas atmosféricas. Descargas electrostáticas.	RADIACIÓN Antena–Antena. Radiación de chasis. Conductores.	Todo sistema eléctrico/electrónico. Principalmente en telecomunicaciones. Principalmente equipos industriales. (red de alimentación, fuente de alimentación, cableado de señal, conductor común)
PROVOCADAS Transmisores y receptores RF. Equipos industriales: motores, interruptores, elementos de arco. Equipos domésticos.	ACOPLAMIENTO Inductivo. Capacitivo.	

Tabla 2.1 *Los elementos básicos de la compatibilidad electromagnética.*

Ámbito	Definiciones Generales	Ordenadores, Periféricos Comunicación	Equipos Industriales	Instrumentos y Medida	Red Eléctrica
Normas Europeas	CISPR 7 CISPR 8 CISPR 9	CISPR 22 EN 55.022	CISPR 22 IEC-801 IEC-1000 IEC-65	CISPR 6 CISPR 16 CISPR 17	EN 60.555 IEC-555
Norma Alemana		VDE 0804 VDE 0871 VDE 0878 VDE 0879	VDE 0875	VDE 0565 VDE 0847 VDE 0848 VDE 0870 VDE 0876 VDE 0877	
Norma USA y América Latina		FFC 15-J		ANSI C63.2	

Tabla 2.2 *Principales normas internacionales sobre EMI.*

La instalación de elementos de protección escalonada contra las altas tensiones transitorias o sobre tensión es indispensable, ya que de una forma cómoda, simple y práctica dispondrá de seguridad y continuidad en el uso de los equipos eléctricos y electrónicos. La eficacia de la protección escalonada contra las tensiones transitorias depende principalmente de dos factores: (1) Una adecuada selección de los descargadores de tensión, y (2) Su correcta instalación.

La puesta de una barrera de sobretensión es muchísimo más económico que la sustitución de cualquier equipo electrónico destruido a falta de ella.

Para que el sistema de protección funcione correctamente debe contar con un sistema de puesta a tierra, a donde se desviarán los altos voltajes transitorios.

A continuación se describe brevemente los dispositivos utilizados para la compatibilidad electromagnética (EMC).

2.2.1. Gap, bobina de choque, fusible, condensador, varistor y TVS

A. Gap

El Gap tiene la propiedad que ante un voltaje alto presentan descargas en forma de arco entre los terminales y tierra. La tensión mínima para producir un arco eléctrico a una pequeña distancia entre dos conductores es aproximadamente de 1,5 KV. La protección de sobretensión actúa para cualquier tensión de gran voltaje transitorio.

Es uno de los dispositivos electrónicos principales y de gran importancia en la barrera de protección. Está constituido por dos pequeñas placas rectangulares de bronce que se encuentran conectadas a las líneas y una tercera entre ellas conectada a tierra, con una distancia de 1,5 mm; las tres placas son perpendiculares a las líneas. Se diseña para la condición de caída directa de descarga atmosférica.

B. Bobina de choque

Construido con alambre de Cobre esmaltado 18 AWG de aproximadamente 40 espiras circulares, de baja inductancia que se comporta como impedancia de gran valor para tensiones transitorias, oponiéndose con gran resistencia a la corriente transitoria de entrada. El valor de la inductancia se obtiene por el tiempo promedio de los transitorios y además por la corriente máxima a soportar, y para un sistema de filtro transitorio tendrá una impedancia alta. La inductancia de la bobina de choque es de 15 a 20 μH y se calcula utilizando:

$$L = (9,87 * d^2 * N^2 * k * 10^{-9}) / l \quad \dots (2.1)$$

Donde: L: inductancia en henrios, d: diámetro de la bobina en cm.

l : longitud de la bobina en cm., k: factor de forma (toma valores entre 0,3 y 1.0).

N: número de espiras de la bobina.

C. Fusible

El fusible se selecciona en función de la corriente de la carga y la tensión de línea. La carga lo constituye todos los circuitos a alimentar. El sistema tiene dos fusibles de protección para actuar cuando exista corriente mucho mayor que el valor nominal en la línea de entrada.

D. Condensador

Son condensadores para altos voltajes, pero de baja capacidad. Para el sistema de filtrado en un transitorio, estos ofrecen muy baja impedancia, derivando la sobre alta corriente hacia el punto de tierra. Los valores de las capacidades van desde 470 pF hasta 2,2 nF para voltajes del orden de 4 KV DC.

E. Varistor

Dispositivo electrónico fundamental e importante en el sistema de protección, cuya función es la suprimir los altos voltajes transitorios recortándolos al nivel nominal de trabajo, presentando una impedancia muy alta para una tensión nominal de trabajo y disminuyendo esta impedancia según aumente el voltaje, hasta hacerse cortocircuito cuando alcanza su voltaje clamping, protegiendo de esa manera los componentes sensibles del circuito. Los varistores se fabrican con material no homogéneo (Carburo de silicio).

Los varistores proveen protección confiable y económica contra altos transitorios que pueden ser producidos por: descarga atmosférica, conmutaciones o ruido eléctrico sobre líneas de poder en DC o AC. Los varistores pueden absorber energías transitorias altas, suprimiendo los transitorios positivos y negativos. Sus principales características son:

- Amplia gama de voltajes desde 14 V a 550 V (rms). Esto permite una selección fácil del componente correcto para una aplicación específica.
- Alta capacidad de absorción de energía respecto a la dimensión del dispositivo y comparado con el diodo TVS.
- Tiempo de respuesta menor a 20ns, absorbiendo el transitorio instantáneamente.
- Bajo consumo en estado de espera, virtualmente nada.
- Valores bajos de capacidad, lo que hace al varistor apropiado para la protección de circuitería en conmutación digital.
- Alto grado de aislamiento. Provee protección hasta 2500V, previniendo cortocircuito en las pistas o componentes adyacentes en la tarjeta de circuito impreso.

E.1. Máximo impulso de corriente no repetitiva: El pico máximo de corriente permitido a través del varistor depende de la forma del pulso, duración del ciclo y del número de pulsos. Con el fin de caracterizar la capacidad del varistor para resistir impulsos de corriente, se permite generalmente que garantice un máximo impulso de corriente no repetitiva. Este viene dado por un impulso caracterizado por la forma del impulso de corriente desde 8 microsegundos a 20 microsegundos siguiendo la norma IEC 60-2, con tal que la amplitud del voltaje del varistor medido a 1mA no lo hace cambiar más del 10% como máximo.

Un impulso mayor que el especificado puede ocasionar cortocircuitos o ruptura del propio componente; se recomienda por lo tanto instalar un fusible en el circuito que utiliza el varistor, o utilizar una caja protectora.

Si se aplica más de un de pulso o el pulso es de una duración mas larga, habría que estudiar las curvas que al efecto nos proporcionan los fabricantes. Estas curvas garantizan la máxima variación de voltaje (10%) en el varistor con 1 mA.

E.2. Energía máxima: Durante la aplicación de un pulso de corriente, una determinada energía será disipada por el varistor. La cantidad de la energía de disipación es una función de:

- La amplitud de la corriente.
- El voltaje correspondiente al pico de corriente.
- La duración del pulso.
- Energía que se disipa durante el tiempo entre 50% y 100% del pico de corriente en la bajada del pulso.
- La no linealidad del varistor.

A fin de calcular la energía disipada durante un impulso, se hace con la referencia generalmente a una onda normalizada de la corriente. Esta onda está prescrita por la norma IEC 60-2 tiene una forma que aumenta desde cero al valor pico en un tiempo corto, disminuyendo hasta cero de una manera exponencial, o bien sinusoidal. Esta curva es definida por el tiempo principal virtual (t_1) y el tiempo virtual al valor medio (t_2) como el mostrado en la Figura 2.1. El cálculo de energía durante la aplicación de tal pulso viene dado por la fórmula:

$$E = V_{\text{peak}} * I_{\text{peak}} * t_2 * K \quad \dots (2.2)$$

Donde: I_{peak} : corriente de pico

V_{peak} : voltaje a la corriente de pico.

K: constante que depende de t_2 , cuando t_1 va de 8 a 10 microsegundos.

Para $t_2 = 20, 50, 100, 1000 \mu\text{s}$; tiene su respectivo valor K de 1,0 a 1,4.

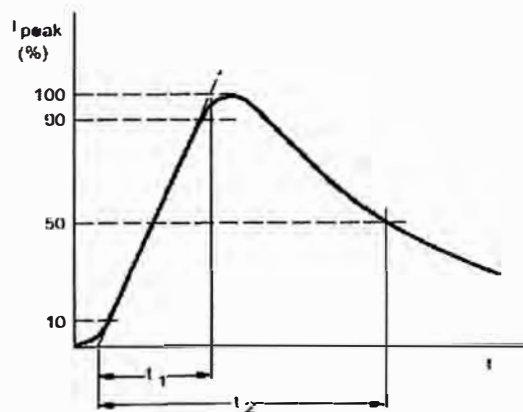


Figura 2.1 Curva de Máxima energía del varistor.

La energía máxima no representa entonces la calidad del varistor, pero puede ser un indicio valioso cuando comparamos diversas series de componentes que tienen el mismo voltaje. La energía máxima indicada por los fabricantes es válida para un pulso estándar de duración entre 10 y 1000 microsegundos, que dan como máxima

variación de voltaje un 10 % para 1 mA. Cuando se aplican más de un pulso, recurriremos a las tablas que a tal efecto nos proporcionan los fabricantes.

E.3. Características típica V/I de un varistor de ZnO: La relación entre la tensión y corriente en un varistor viene dada por:

$$V = C * I^\beta \quad \dots (2.3)$$

Donde: V: es el voltaje.

C: es el voltaje del varistor para una corriente de 1 A.

I: es la corriente actual que atraviesa el varistor.

β : es la tangente del ángulo que forma la curva con la horizontal. Este parámetro depende del material con que está fabricado el varistor; en el caso del ZnO su valor es $\beta = 0,035$.

En la figura 2.2 se muestra una curva característica de un varistor de ZnO.

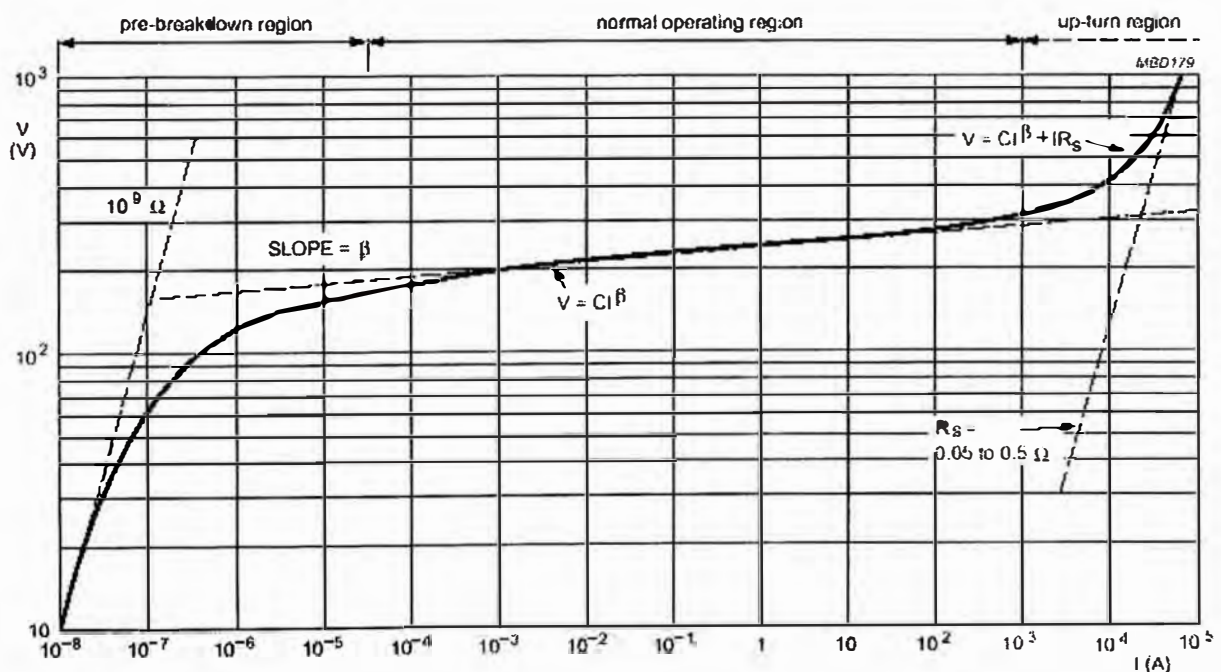


Figura 2.2 Típica curva característica V/I de un varistor de ZnO.

F. Diodo TVS

El diodo supresor de voltaje transitorio (TVS) es un dispositivo que absorbe (disipa) el voltaje transitorio, limitando el voltaje pico hasta un nivel inofensivo (*Clamplng voltage*) debido a su baja impedancia. Son usados para proteger dispositivos sensibles (tecnología CMOS y circuitos integrados de bajo voltaje de alimentación) a sobre tensiones eléctricas tal como aquellos causados por descargas atmosféricas, cargas inductivas conmutables y descarga electrostática. Se cuenta con diodos TVS para protección unidireccional y bidireccional. Los diodos TVS están disponibles para voltajes de operación desde 5V hasta 376V.

La mayoría de estos dispositivos están especificados con una forma de onda 10us/1000us (10us para elevarse el pico y 1000us de caída exponencial en un pulso), en un cambio reciente agregar 8/20 us en el mismo dispositivo. La potencia del pulso pico de las familias de diodos TVS son normalmente especificados en Kilowatts que van desde 0,3 KW hasta 15 KW.

La serie TPA son dispositivos de protección bidireccional, especialmente diseñados para la protección de sistemas de telefonía.

2.2.2. Filtros EMI

Las señales EMI se encuentran dentro rango de frecuencia hasta los 30 MHz y son portados principalmente en las líneas de alimentación A.C o D.C desde la fuente para afectar el sistema. Se presentan en modo común y modo diferencial (ver figura 2.3).

La EMI en modo diferencial, aparece como un voltaje entre las líneas de alimentación del equipo. Aquí la corriente circula completamente las líneas de alimentación y la fuente. No fluye del conductor a tierra.

La EMI en modo común, aparece como voltaje en ambas líneas con respecto a tierra. Aquí la corriente fluye directamente de ambas líneas hacia tierra. Si el sistema no tiene una apropiada tierra de conexión; la corriente fluye a través del capacitor, el chasis del sistema y tierra.

Un filtro EMI es esencialmente una red LC (inductor, capacitor) en configuración de filtro pasa bajo, diseñado para atenuar interferencias de frecuencias altas al mismo tiempo que la corriente de operación de baja frecuencia pasa directamente inafecto. Es importante notar que la aproximación básica para el filtrado EMI es para usar una combinación de inductores y capacitores para desviar el flujo de corrientes EMI lejos de la víctima, teniendo como característica que:

- Previene el ingreso de altas frecuencias de interferencia portados por la línea de alimentación A.C dentro de la protección del equipo.
- Previene la salida de energía de interferencia generada dentro de la protección del sistema, hacia la línea de alimentación A.C.

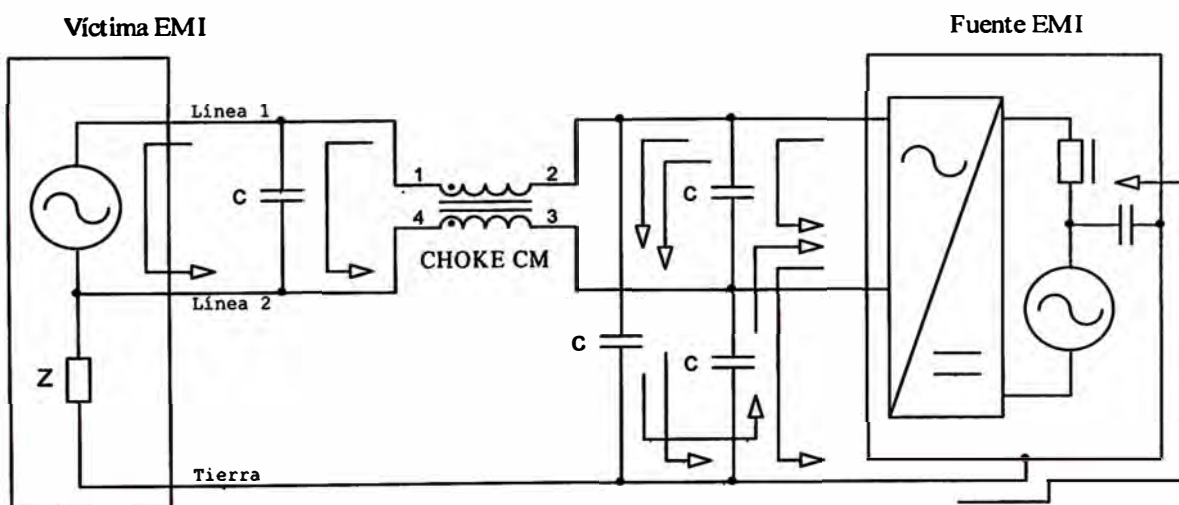


Figura 2.3 Muestra la disposición de los componentes (L , C) en el circuito para el filtrado de la interferencia electromagnética (EMI) en ambos modos. También el flujo de corriente en modo diferencial (línea continua) y en modo común (línea punteada).

El filtrado de la EMI en modo diferencial es usualmente reducido usando un par de inductancias acopladas llamado choque en modo común y dos capacitores desde cada línea a tierra, de capacidades relativamente bajos. En este arreglo la alta impedancia del inductor a frecuencias altas es usada para bloquear la salida de corriente hacia la víctima. En el modo diferencial el filtrado requiere capacitor entre las líneas.

2.2.3. Sistema de puesta a tierra y pararrayo

Es tan grande la masa del globo terráqueo que su potencial se mantiene prácticamente invariable cualquiera que sea la cantidad de carga eléctrica que se le aplique. En esta característica se basa el principio de puesta a tierra.

Desde el punto de toma de tierra del pararrayo hasta la fosa de baja resistencia se encuentra un cable de cobre con una sección mínima de 35 mm^2 sujetas al soporte por medio de guías de cable con aislador soldado o empotrado, la distancia de separación de estas debe tener un mínimo de 2 m., para la entrada del cable conductor a la fosa de baja resistencia debe existir un tubo aislante o chapa de hierro con una altura mínima aproximada de 2.5 m.

El electrodo (cuerpo metálico puesto en intimo contacto con el terreno y destinado a dispersar en el mismo las corrientes eléctricas) debe encontrarse a una distancia no menor de 50 cm por debajo del nivel del suelo, su longitud debe estar al rededor de los 2 m de hierro cobreado con un diámetro mínimo de 14 mm. Se debe procurar que la resistencia de tierra se encuentre alrededor de los 3Ω .

Existen métodos para reducir la resistividad. Por ejemplo, se puede recurrir a las sales minerales (cloruro de sodio, sulfato de magnesio, y sulfato de cobre) disueltas

en agua y vertidas sobre el terreno o en el electrodo si este es tubular. Sin embargo, los resultados no siempre se corresponden con lo previsto y en algunos casos se producen fenómenos de corrosión. Es mucho mejor, siempre que ello sea posible, hacer uso de un terreno que tenga contenido orgánico. Con la idea de modificar la naturaleza química del terreno, se puede recurrir a la colocación de capas de carbón (o grafito en polvo) situadas directamente en contacto con los electrodos. Este procedimiento, de difícil aplicación cuando los electrodos tienen forma de pica, puede ser utilizado siempre que se recurra a electrodos en forma de placa, malla.

El sistema que se encarga de servir como anzuelo para las descargas atmosféricas está constituido por el pararrayo, que en todos los tipos cuenta con una varilla punzante para que el campo eléctrico que se forme a su alrededor sea más intenso. Entre los tipos de pararrayos se tiene:

- Pararrayos de punta Franklin Estándar.
- Pararrayos de punta Franklin Múltiple.
- Pararrayos Ionizantes.

El **pararrayo Ionizante**, además de contener varias puntas similares a las del Franklin que hacen de absorbedores de rayo (descarga atmosférica) o chispa cuando se genera, llevan una placa ionizada que atrae, por su especial composición, con más eficacia la chispa eléctrica que se produce a su alrededor. Estas puntas se calibran para radios comprendidos entre 5 y 50 m. y su protección se extiende al volumen que forma una semiesfera de radio definido por su calibre, teniendo como centro la punta del pararrayos, colocada sobre una base cilíndrica del mismo diámetro y de la altura definida desde la punta del pararrayos hasta el suelo. La protección es total y, no forma un cono como en el **pararrayo Franklin**, sino que la bóveda y su base forman

el espacio de protección y todo lo comprendido dentro de el queda totalmente protegido. La punta del pararrayo Ionizante deberá quedar por lo menos 1 metro más elevada que cualquier otro punto que se desea proteger.

2.3. Principales dispositivos utilizados en el sistema

2.3.1. Transductor de presión

El MPX2100DP, es un transductor de presión de silicio; un chip calibrado y compensado en temperatura. Esta serie de dispositivos MPX2100 (100 KPa) son piezoresistencias de silicio sensibles a la presión que proporciona una variación de tensión exacta y lineal, directamente proporcional a la presión que se le aplica.

El sensor consta, de un diafragma monolítico de silicio para la medida del esfuerzo y una fina película en una red de resistencias integradas en un chip. El chip se ajusta, calibra y se compensa en temperatura por láser (ver figura 2.4).

Se utiliza en el control de electrobombas, robótica, indicador de niveles, diagnósticos médicos, cambio de presión, Barómetros, Altimetros, etc.

A. Características

- Compensación de temperatura por encima de 0°C a +85°C.
- Único silicio para la medida de la presión tensión y torsión.
- Distintos encapsulados del chip para facilitar su uso.
- Disponible para configuraciones de medida en modo absoluto y diferencial.
- Variado voltaje de alimentación.
- Voltaje de salida de 40 mVDC a escala completa.
- Linealidad: $\pm 0,25\%$ para el MPX2100D.

B. Tensión de salida proporcional a la presión diferencial aplicada: La tensión de salida diferencial del *X-ducer* (elemento sensor) es directamente proporcional al diferencial de la presión aplicada. El sensor absoluto tiene una entrada de referencia al vacío; la tensión de salida es menor al vacío, respecto a la presión ambiente; se introduce por el lado de la presión (P1). El voltaje de salida diferencial o del sensor de medida aumenta al aumentar la presión que se aplica por el lado de la presión (P1) respecto del lado del vacío (P2).

Estos dispositivos son los recomendados por Motorola para los futuros diseños y como mejor valor de un conjunto global.

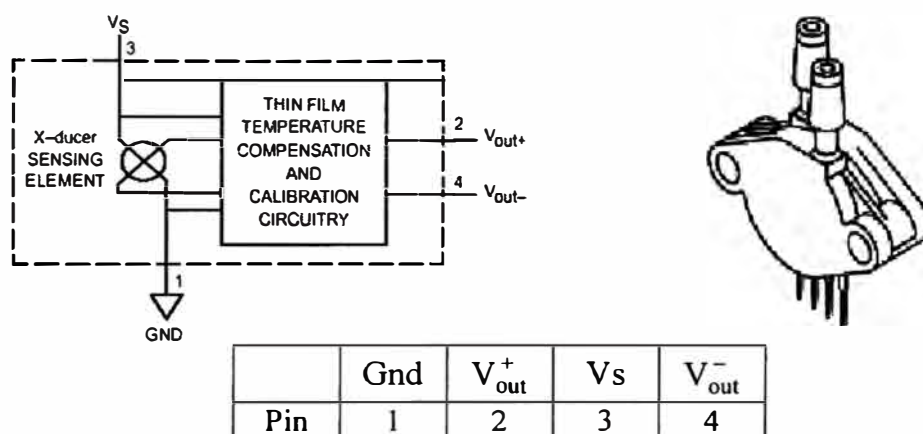


Figura 2.4 Muestra el diagrama de bloques interno del transductor de presión MPX2100 y su encapsulado para el tipo diferencial; así mismo la asignación de pines, encontrándose marcado el pin 1 con una muesca.

C. Linealidad: La linealidad indica que el transductor sigue la ecuación:

$$V_{out}^+ - V_{out}^- = V_{off} + S * P; \text{ donde: } P=\text{Presión, } S=\text{Sensibilidad} \quad \dots (2.4)$$

dentro del rango de presión de trabajo. Hay dos métodos básicos para calcular la no linealidad: (1) Trazando una línea que una los puntos externos con una línea recta, ver figura 2.5 o (2) una línea intermedia entre los puntos extremos. El segundo es el mejor de los casos de error de linealidad (el de menor valor numérico), los cálculos

requeridos son complejos y pesados. Recíprocamente, el trazado de un punto extremo dará “el peor caso” del error (a menudo más deseable en los cálculos de suposición de error) y los cálculos son más deseables para el usuario. La linealidad del sensor de presión especificado por Motorola está basado en el método de la línea entre puntos extremos de la presión de trabajo.

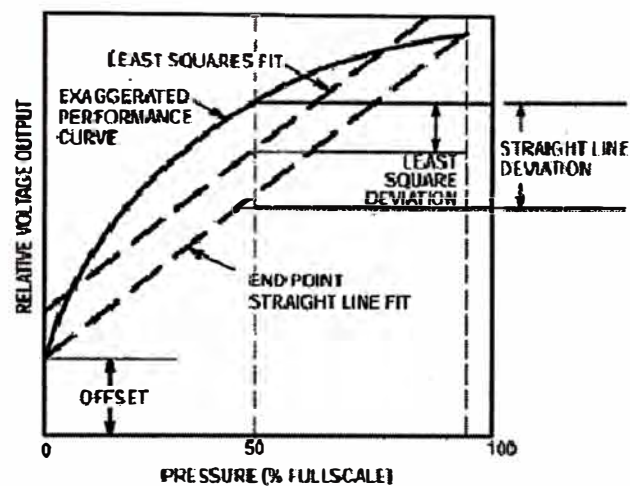


Figura 2.5 Comparación de las especificaciones de linealidad.

D. Calibración y compensación de temperatura en el chip: La Figura 2.6 muestra la característica de salida del MPX2100 a 25 °C. La Salida es directamente proporcional a la diferencia de presión y es prácticamente una línea recta.

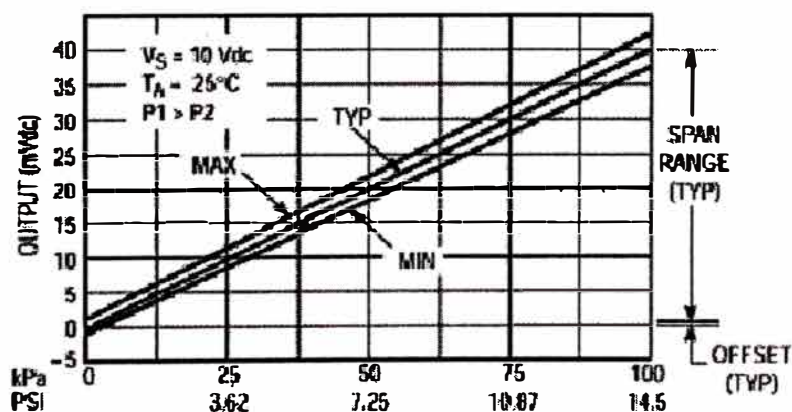


Figura 2.6 Tensión de salida en función de la diferencia de presión.

E. Identificando lado de presión P1 o vacío P2: Motorola designa los lados del sensor como lado de presión (P1) y lado de la presión de vacío (P2). El lado de la presión P1 contiene el gel de silicona que aísla el lado. El diferencial sensible a la medida se diseña para operar con una diferencia de presión positiva $P1 > P2$. El sensor absoluto se diseña aplicando el vacío en el lado P1.

2.3.1. Amplificador de instrumentación y convertidor A/D

A. Amplificador de instrumentación

Básicamente es un módulo de ganancia controlada, que procesa voltajes diferenciales con elevada precisión y con rango de ganancia programable entre 1 y 1000. Idealmente estos amplificadores responden únicamente a señales diferenciales y rechazan totalmente las de modo común.

La categoría de instrumentación implica que el amplificador debe poseer alta impedancia de entrada, bajo offset, elevada linealidad, ganancia estable y reducida impedancia de salida. Los amplificadores de instrumentación tienen gran aplicación en el proceso de señales de termocuplas, puentes, shunts de corriente, señales biológicas, preamplificación de señales diferenciales con elevado modo común y cambio de señales diferenciales flotantes a señal con referencia a tierra.

El modelo básico está constituido por tres amplificadores operacionales y siete resistencias, como se muestra en la figura 2.7. Las entradas (V1) y (V2) se encuentran conectadas a seguidores de voltaje, seguidos de un amplificador diferencial que en conjunto otorgan elevada impedancia, ganancia controlable a partir de R_g y ajuste de tensión DC en la salida mediante V_{ref} . La señal de salida V_o como función de la señal diferencial es:

$$V_o = \frac{R_3}{R_2} \left(1 + \frac{2R_1}{R_g} \right) (V_2 - V_1) + V_{ref} \quad \dots (2.5)$$

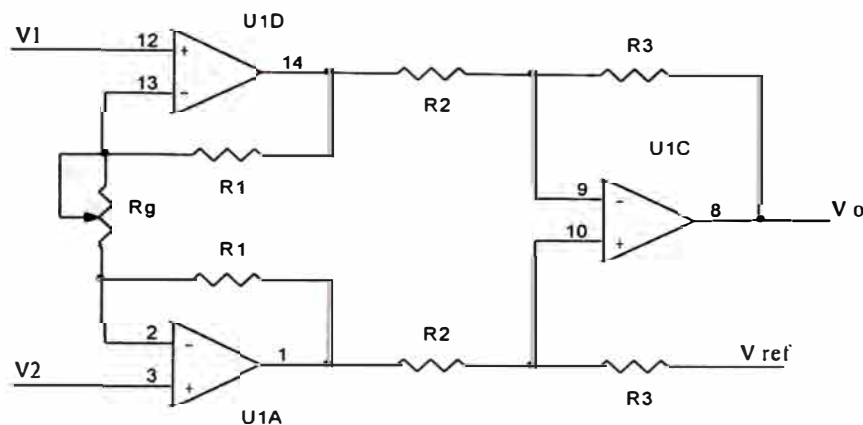


Figura 2.7 Modelo básico del amplificador de instrumentación.

El amplificador de instrumentación utilizado en este trabajo es el AD623 fabricado por Analog Device, para simple o doble alimentación y encapsulado DIP de 8 pines. Este amplificador de instrumentación modificado, se basa en el modelo básico; cuyas entradas diferenciales se aplican a transistores PNP que actúan como excitadores de voltaje y proporcionan un modo común para las señales de entrada. Las siete resistencias son iguales a $50K\Omega$, siendo la salida V_o :

$$V_o = \left(1 + \frac{100 K\Omega}{R_g} \right) (V_2 - V_1) + V_{ref} \quad \dots (2.6)$$

B. Convertidor A/D

El convertidor analógico a digital (ADC) permite que el mundo analógico se comunique con la computadora, y esta a su vez se comunique con el hombre y con los procesos físicos a través del convertidor digital analógico (DAC). Las características de los diferentes tipos de convertidores A/D (de simple rampa, de

doble rampa, paralelo, de aproximaciones sucesivas, de descarga, etc.) dependen en gran parte de la forma en que realiza la conversión.

El convertidor de aproximaciones sucesivas tiene una circuitería más compleja que el ADC rampa, pero un tiempo de conversión más pequeño. Además estos convertidores tienen un tiempo fijo de conversión que no depende del valor de la señal analógica V_{in} .

La figura 2.8 muestra la configuración básica de un convertidor analógico digital de aproximaciones sucesivas. No utiliza un contador para generar la entrada al DAC sino que emplea un registro de aproximaciones sucesivas (SAR). La lógica de control modifica bit por bit el contenido del SAR hasta que el contenido de éste se convierta en el equivalente digital de la entrada analógica V_{in} .

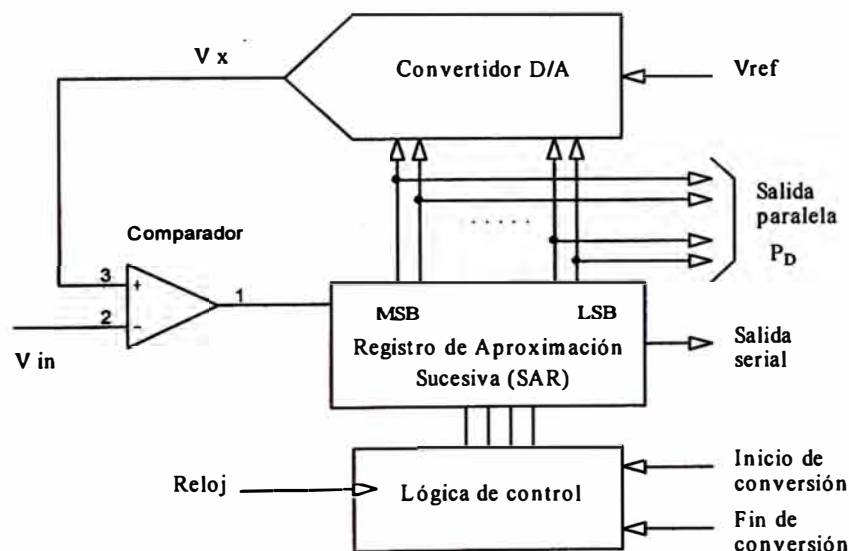


Figura 2.8 Diagrama de bloques de ADC por aproximaciones sucesivas.

La operación del convertidor comienza cuando la lógica de control borra todos los bits del SAR; luego el DAC transforma cada entrada digital en una salida analógica V_x , esta se compara con V_{in} . El comparador le dice al SAR si V_{in} es mayor

o menor que V_x . Si es mayor el bit del registro se pone a 0 y si es menor el bit se pone a 1. Las comparaciones se hacen comenzando desde el MSB y termina en el LSB. Al terminar la lógica de control envía una señal indicando que finalizó la conversión, entonces el equivalente digital de V_{in} dentro de la resolución del convertidor, está en el registro (P_D).

En general la salida de un DAC está dado por:

$$V_{sal} = R * VP_D \quad \dots (2.7)$$

$$R = \left(\frac{V_{ref}}{2^n} \right), \quad VP_D = \left(\frac{2^{n-1} * A_{n-1} + 2^{n-2} * A_{n-2} + \dots + 2^0 * A_0}{2^n} \right) \quad \dots (2.8)$$

Donde: R: resolución; VP_D : valor decimal del número binario P_D .

A_{n-1} (MSB), A_{n-2} , . . . , A_0 (LSB) son los bits del número binario P_D .

Para un convertidor ADC, en general la salida P_D esta dado por:

$$P_D \cong \text{Binario equivalente de} \left(\frac{V_{in}}{R} \right) \quad \dots (2.9)$$

2.3.2. EEPROM serial

La EEPROM serial, es una de las tecnologías de memoria no volátil que emergen como principal solución en control. Memoria que se utiliza en sistemas que requieren pequeño espacio, bajo consumo y alta flexibilidad; pero la mayor parte de diseñadores no están enterados de los beneficios de la EEPROM serial. Igualmente la documentación de soporte son muchas veces no adecuadas debido a información incompleta o ambigua.

Las aplicaciones más comunes de la EEPROM serial se encuentran: En consumo: TV, CD, cámaras, radios y control remoto. En la oficina: Impresoras,

copiadoras, PC's y PC's portátiles. Telecomunicaciones: celulares, teléfonos, fax, módem y otros. Industria: código de barras, tarjetas, equipos de medida, etc.

Luego de decidido el uso de una EEPROM serial, el próximo paso es seleccionar uno de los protocolos de comunicación serial, ya sea el protocolo de 2 o 3 líneas.

Un producto de 2 líneas es utilizado en aplicaciones que requieren un bus I²C, inmunidad al ruido, limitadas líneas de E/S en el microcontrolador o un buffer de escritura para múltiples bytes para ser almacenados con una instrucción (dispositivos de la serie 24XXXX y 85XXXX). Solo los pines SDA y SCL son esenciales para la operación del bus, los otros pines son suplementarios: SCL (reloj serial), SDA (dato serial), A<2-0> (selecciona chip o bloc). El protocolo I²C utiliza comunicación Maestro/Esclavo bidireccional y datos de 8 bits.

Un producto de 3 líneas es utilizado en aplicaciones que tienen limitados requerimientos de protocolo; un protocolo SPI (dispositivos de la serie 93XXXX), requiere de alta frecuencia de reloj o aplicaciones de 16 bits de dato. Para el control del dispositivo se requieren 3 pines: DI (ingreso de dato), DO (salida de dato), CLK (reloj). El protocolo SPI permite la transferencia de datos de 8 o 16 bits según se configure, que pueden ser transmitidos y recibidos de forma síncrona y simultánea.

En el trabajo realizado se utiliza EEPROM serial de protocolo SPI, que se encuentra al interior del microprocesador PIC16F876.

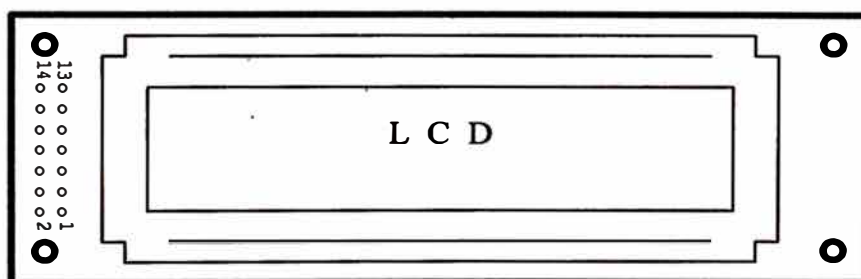
2.3.3. Visualizador LCD

El módulo L2012 es una Pantalla de Cristal Liquido (LCD) de bajo consumo, periférico de salida de visualización, flexible, eficaz y económico, siendo los microcontroladores los dispositivos más adecuados para optimizar sus prestaciones y

control. Módulo de contraste variable y un excitador de LCD CMOS basado en el microcontrolador HD44780 de Hitachi. El microcontrolador genera caracteres desde una ROM o RAM. Todas sus funciones son controladas por comandos y el módulo es fácilmente conectado a un microcontrolador (ver figura 2.9).

Características

- LCD de dos líneas de 20 caracteres de 5 x 7 o 5x10 pixeles y cursor.
- 192 tipos de caracteres ASCII generados desde la ROM (tamaño 5x7 pixeles).
- Ocho caracteres generados por el usuario en la RAM (CG RAM).
- Memoria RAM de datos para visualizar de hasta 80 bytes (DD RAM).
- Capacidad de interface de 4 o 8 bits con el microprocesador.
- Entre los comandos tenemos: Borrar pantalla, Cursor en posición 0, Pantalla o Cursor encendido/apagado, Parpadeo de caracter, Desplazamiento de cursor, y Desplazamiento de pantalla.
- Incluye circuito oscilador y Reset automático cuando se conecta la alimentación.
- Alimentación de +5 VDC.



	Vss	Vdd	Vo	RS	R/#W	E	D0	D1	D2	D3	D4	D5	D6	D7
Pin	1	2	3	4	5	6	7	8	9	10	11	12	13	14

Figura 2.9 Aspecto físico del módulo LCD y su asignación de pines.

El módulo dispone de 14 pines de conexión, cuyas funciones son:

- Ocho pines (D0-D7) reciben los caracteres ASCII a presentar, así como códigos de control que regulan los efectos de visualización. También por ellos el módulo LCD envía información sobre su estado interno.
- En los pines Vdd y Vss se aplica la alimentación de +5 V y tierra.
- El pin Vo regula el contraste del LCD al aplicarle una tensión variable comprendida entre 0 y 5 V.
- Los tres pines restantes controlan las funciones del módulo: E (Habilitador) =1 habilita módulo, E =0 módulo desactivado y no funcionan las restantes señales. Si R/#W (Lectura/Escritura) =0 se escribe en el módulo y si R/W# =1 es leído. Si RS (Selección de Registro) =0 de control y RS =1 de datos.

2.3.4. Bombas de agua, tipos y funcionamiento

Debido a que raramente se decide a priori si el grupo a instalar ha de tener tal o cual tipo de flujo o un número de impulsores establecido, el criterio de clasificación, quizás el más operativo es por *la disposición del eje de rotación*: Bombas verticales y Bombas horizontales.

Dicho criterio no establece una clasificación fija ya que, por ejemplo, en determinados casos, una bomba de diseño horizontal puede disponerse verticalmente, o una bomba sumergida bien puede instalarse en posición horizontal.

A. Bomba horizontal

Utilizados para elevar aguas superficiales o de pozos poco profundos, teniendo por lo general impulsión centrífuga. La altura de aspiración no debe ser superior a 7 metros.

Su característica más representativa es la de estar ubicados la bomba y el motor en superficie. La conexión más común entre ambos es la de manguito semielástico o unión directa para motores eléctricos; y la de poleas y correas o eje Cardam, en motores Diesel. Su capacidad de elevación en altura manométrica, difícilmente supera los 60-70 metros. Pueden ir solidariamente acopladas a un motor generalmente eléctrico (bombas monoblock) o ser suministradas con eje libre, para su acople al motor deseado, para lo cual habrá de disponerse una bancada que actúa como soporte común a motor y bomba y un acoplamiento entre ambos ejes. En estos casos la alineación correcta de los dos elementos es fundamental para prevenir posibles averías, entre otras, de los cojinetes de ambas máquinas.

B. Bomba vertical

Se utiliza para elevar aguas de pozos estrechos y profundos, encontrándose la bomba sumergida y el motor en la superficie. Por lo general tienen impulsión axial. Fabricados para ser utilizados en pozos de profundidad media, en donde el nivel dinámico del agua se encuentra a más de 5 m. Su característica principal es la de estar sumergida la bomba (a la profundidad que requiera el nivel dinámico) y el motor instalado en superficie. La transmisión se realiza mediante un eje, a veces de decenas de metros, que se instala en el interior de la tubería de impulsión, sujetado mediante cojinetes. Cuando la profundidad empieza a ser importante (no es aconsejable su instalación a más de 80 m de profundidad) este modelo de grupos deja de ser interesante por diversos motivos, entre otros: Que al no ser perfecta al 100% la verticalidad del eje se producen desgastes y averías, al elevado costo de instalación, las grandes pérdidas de rendimiento y lo costoso del mantenimiento.

2.3.5. Microcontrolador y sus principales características

Entre más de 50 fabricantes de microcontroladores como Intel, Microchip, Motorola, Zilog, Dallas y Scenix, es muy difícil seleccionar el mejor; son las características específicas de la aplicación que determinan el más conveniente.

Microchip Technology Inc. fabrica los microcontroladores PIC. Hasta 1999 disponía de 60 versiones diferentes que cada año aumenta. Los microcontroladores PIC de 8 bits se agrupan en cinco familias: Gama enana PIC12C(F)XX, Gama baja PIC16C5X, Gama media PIC16C(F)XXX, Gama alta PIC17CXXX y Gama mejorada PIC18C(F)XXX.

Como el sistema diseñado utiliza el PIC16C55 y el PIC16F876 (descritos en capítulos siguientes); se realiza una descripción general de un microcontrolador PIC.

Microcontrolador

El microcontrolador es un computador completo que incluye un CPU, memoria de programa, memoria de datos, puertos de E/S, temporizadores, canales de comunicación, sistemas de interrupciones, sistemas de seguridad, modo de bajo consumo y un largo etcétera; todo en un solo chip de circuito integrado. Fabricadas en tecnología CMOS, para bajo consumo de energía y alta velocidad.

En su memoria no volátil solo reside un programa destinado a gobernar una determinada aplicación, una vez programado y configurado el microcontrolador solamente sirve para gobernar dicha aplicación. Sus líneas de entradas/salidas soportan el conexionado de los sensores y actuadores del dispositivo a controlar y todos los recursos complementarios disponibles tienen como única finalidad atender sus requerimientos.

Arquitectura

Un microcontrolador posee todos los componentes de un computador, pero con unas características fijas que no pueden alterarse. La figura 2.10 muestra el diagrama de bloques principales de su arquitectura, los cuales a continuación se describen.

A. Unidad central de proceso (CPU)

La necesidad de conseguir elevados rendimientos en el procesamiento de las instrucciones ha desembocado en el empleo generalizado de procesadores de arquitectura *Harvard* frente a los tradicionales que seguían la arquitectura *Von Neumann*. Esta última se caracterizaba porque la CPU se conectaba con una memoria única, donde coexistían datos e instrucciones, a través de un sistema de buses.

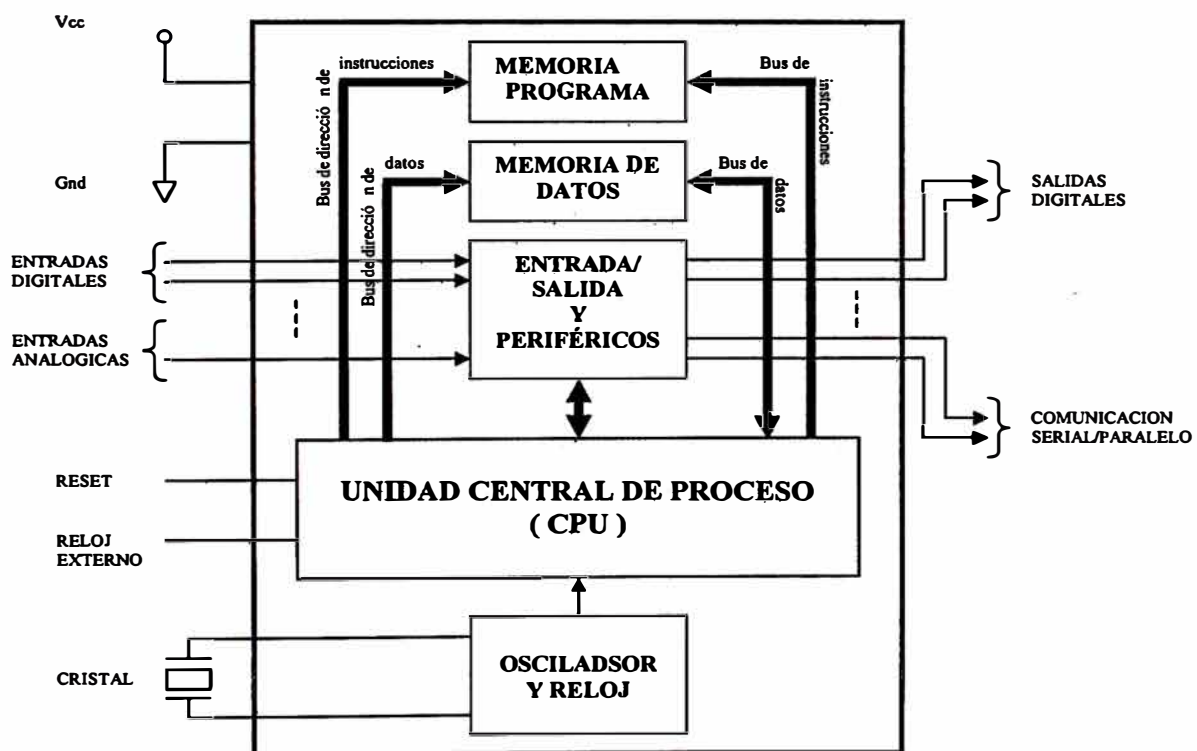


Figura 2.10 Diagrama de bloques de un microcontrolador PIC, mostrando los bloques principales de su arquitectura.

En la arquitectura *Harvard* es independiente la memoria de instrucciones y la memoria de datos, y cada una dispone de su propio sistema de buses para el acceso. Esta dualidad, además de propiciar el paralelismo, permite la adecuación del tamaño de las palabras y los buses a los requerimientos específicos de las instrucciones y de los datos. También la capacidad de cada memoria es diferente (figura 2.11).

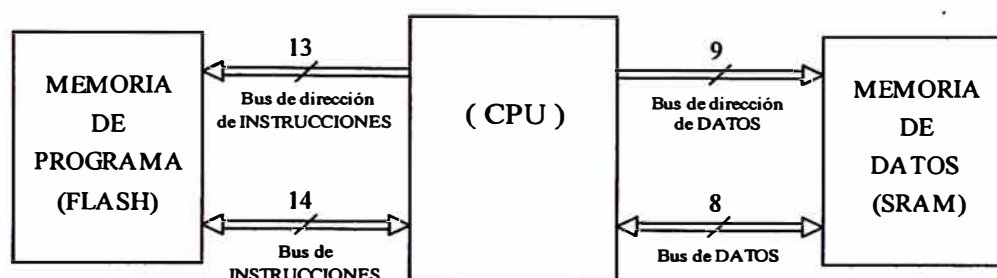


Figura 2.11 *En la arquitectura Harvard la memoria de instrucciones y la de datos son independientes, lo que permite optimizar sus características.*

El procesador de los modernos microcontroladores responde a la arquitectura RISC (Computador de Juego de Instrucciones Reducido), que se identifica por poseer un repertorio de instrucciones de máquina pequeño y simple, de forma que la mayor parte de las instrucciones se ejecuta en un ciclo de instrucción. Otra aportación frecuente que aumenta el rendimiento del computador es el fomento del paralelismo implícito, que consiste en la segmentación del procesador (*pipe-line*), descomponiéndolo en etapas para poder procesar una instrucción diferente en cada una de ellas y trabajar con varias a la vez.

B. Memoria de programa

El microcontrolador está diseñado para que en su memoria de programa se almacenen todas las instrucciones del programa. No hay posibilidad de utilizar memorias externas de ampliación, excepto en la familia PIC17CXXX.

Como el programa a ejecutar siempre es el mismo, debe estar grabado de forma permanente. Los tipos de memoria no volátil utilizados son:

ROM: En este tipo de memoria el programa se graba durante el proceso de su fabricación mediante el uso de máscaras. Los altos costos de diseño e instrumental sólo aconsejan usar este tipo de memoria cuando se precisan series muy grandes.

EPROM: La grabación de esta memoria se realiza mediante un dispositivo físico gobernado desde un computador personal, que recibe el nombre de grabador. En la superficie de la cápsula del microcontrolador existe una ventana de cristal por la que se puede someter al chip de la memoria a rayos ultravioletas para producir su borrado y emplearla nuevamente. Es interesante la memoria EPROM en la fase de diseño y depuración de los programas, pero su coste unitario es elevado.

OTP: Este modelo de memoria sólo puede ser grabado una vez por el usuario, utilizando el mismo procedimiento de la memoria EPROM. Posteriormente no se puede borrar. Su bajo precio y la sencillez de la grabación aconsejan este tipo de memoria para prototipos finales y series de producción cortas.

EEPROM: La grabación es similar a las memorias OTP y EPROM, pero el borrado es mucho más sencillo al poderse efectuar de la misma forma que el grabado, o sea, eléctricamente. Sobre el mismo zócalo del grabador puede ser programada y borrada tantas veces como se quiera, lo cual la hace ideal en la enseñanza y en la creación de nuevos proyectos.

Aunque se garantizan 1.000.000 de ciclos de escritura/borrado en una EEPROM, todavía su tecnología de fabricación tiene obstáculos para alcanzar capacidades importantes y el tiempo de escritura de las mismas es relativamente grande y con elevado consumo de energía.

FLASH: Se trata de una memoria no volátil, de bajo consumo, que se puede escribir y borrar en circuito al igual que las EEPROM, pero suelen disponer de mayor capacidad que estas últimas. El borrado sólo es posible con bloques completos y no se puede realizar sobre posiciones concretas. Son muy recomendables en aplicaciones en las que sea necesario modificar el programa a lo largo de la vida del producto, como consecuencia del desgaste o cambios de piezas, como sucede con los vehículos.

C. Memoria de datos

Los datos que manejan los programas varían continuamente, esto exige que la memoria que los contiene debe ser de lectura y escritura, por lo que la memoria RAM estática (SRAM) es la más adecuada, aunque sea volátil.

Hay microcontroladores que disponen de memoria tipo EEPROM de datos para lectura y escritura. De esta forma, un corte en el suministro de la alimentación no ocasiona la pérdida de la información, que está disponible al reiniciarse el programa.

D. Líneas de entrada/salida

A excepción de dos pines destinadas a recibir la alimentación, otras dos para el cristal de cuarzo (se puede configurar con reloj interno RC), que regula la frecuencia de trabajo, y una más para provocar el Reset; los pines restantes del microcontrolador sirven para soportar su comunicación con los periféricos externos.

Las líneas de E/S que se adaptan con los periféricos manejan información en paralelo y se agrupan en conjuntos de ocho (comunicación paralelo) que reciben el nombre de Puertos. Hay modelos con líneas que soportan la comunicación en serie

(comunicación serie); otros disponen de conjuntos de líneas que implementan puertos de comunicación para diversos protocolos, como el I²C, el USB, etc.

E. Recursos auxiliares

Según las aplicaciones a las que orienta el fabricante cada modelo de microcontrolador, incorpora una diversidad de complementos que refuerzan la potencia y la flexibilidad del dispositivo. Entre los recursos más comunes se citan a los siguientes:

1. Circuito de reloj, encargado de generar los impulsos que sincronizan el funcionamiento de todo el sistema.
2. Temporizadores, orientados a controlar tiempos.
3. Perro Guardián (Watchdog), destinado a provocar una reinicialización cuando el programa queda bloqueado.
4. Convertidores AD y DA, para poder recibir y enviar señales analógicas.
5. Captura, comparación y modulación por anchura de pulso, diversas aplicaciones.
6. Comparadores analógicos, para verificar el valor de una señal analógica.
7. Sistema de protección ante fallos de la alimentación.
8. Estado de Reposo, en el que el sistema queda congelado y el consumo de energía se reduce al mínimo.

2.4. Herramientas de diseño y simulación

Si tenemos en cuenta que el sector electrónico es uno de los que se ven especialmente presionados, se hace imprescindible el uso y conocimiento de herramientas de Diseño Asistido por Computadora (CAD).

Para la automatización del diseño electrónico se cuenta con herramientas de Diseño Electrónico Automático (EDA), que hacen posible realizar un trabajo carente de defectos y en el menor tiempo posible, teniendo siempre en mira elevar la productividad al máximo, minimizando toda tarea rutinaria. Sin duda *OrCAD Release 9.1* será una elección acertada como herramienta EDA. *OrCAD Release 9.1* es un poderoso entorno de diseño electrónico que es igualmente fácil de aprender y usar, gracias a la plataforma Windows. Los módulos fundamentales del entorno gráfico son: *OrCAD Capture*, *OrCAD PSpice A/D*, *OrCAD Layout*, *OrCAD Express*.

A. OrCAD Capture

OrCAD Capture es la aplicación del programa que nos permite diseñar esquemas de circuitos eléctricos–electrónicos bajo un entorno gráfico, tanto analógicos como digitales, o bien mixtos.

El diseño del esquema es la fase más importante y la primera que tenemos que realizar en la mayor parte de los sistemas de diseño electrónico. A continuación se describe brevemente los pasos a seguir:

1. El inicio de una sesión de trabajo lo podemos llevar a cabo mediante el menú *File*, seleccionando la opción *New* para crear un nuevo proyecto, o bien *Open* para abrir un proyecto ya creado. *Capture* muestra el diseño en una ventana administrador de proyecto.
2. Dentro del menú *Options*, se configura las preferencias por defecto para el diseño actual, tales como tamaño de hoja, tamaños de letra, aspecto, colores, etc.
3. Seleccione un dispositivo con *Part* desde el menú *Place* o desde el botón de la paleta de herramientas (para diseñar un esquema seleccionar herramientas desde

el menú, o desde una paleta de herramientas; cada botón de la paleta de herramientas proporciona una función eléctrica o gráfica; como añadir cables y componentes o colocar objetos gráficos), mostrándose una ventana de dialogo para ubicar el dispositivo en las distintas librerías; de esta manera se colocan todos los dispositivos utilizados en el circuito. *Edit Part* del menú *Place* nos permite crear nuevos componentes o modificar los existentes, que igualmente dispone de una paleta de herramienta.

4. Enseguida se puede girar con *Rotate*, borrar con *Delete* del menú *Edit* y mover arrastrándolo con el mouse. Colocados todos los dispositivos se procede a unirlos eléctricamente con *Wire* del menú *Place*. Luego se realiza el procesamiento con *Annotate...*, *Update Properties...*, *Desing Rules Check*, *Create Netlist*, *Cross Reference*, *Bill of Materials*, etc. del menú *Tools*.
5. Finalmente se tendrán los reportes del circuito diseñado imprimiendo el listado de componentes con o sin ubicación dentro del esquema y el esquema del circuito.

B. OrCAD PSpice A/D

Orcad PSpice A/D es la aplicación del programa que nos permite visualizar todas las señales y formas de ondas que tienen lugar en cualquier punto del circuito simulado.

Los tipos de simulación que pueden realizarse son: Transitorio, A.C, D.C y Fourier.

Para realizar la simulación de circuitos tanto analógicos y/o digitales se siguen los pasos siguientes:

1. Utilizando *OrCAD CAPTURE* se diseñará el esquema del circuito deseado, con todos sus elementos. Se debe utilizar los componentes de las librerías de *PSpice*.

Una vez realizado el circuito, ejecutaremos el comando *New Simulation Profile* que se encuentra dentro del menú *PSpice*. Pondremos un nombre al proyecto y ejecutaremos *Create*.

2. El siguiente paso será definir los atributos de cada elemento. Haciendo doble clic se mostrara una donde se definirán los nuevos valores. *OrCAD PSpice* tomará los valores por defecto en el sistema internacional, es decir Ohmios, Voltios, Amperios, Henrios, Faradios, etc. En *Edit Simulation Settings* se selecciona el tipo de simulación, la duración y el inicio del mismo. Además se configuran otras opciones en *Options*. Se guarda el proyecto utilizando *Save As* de *File*.
3. Marcamos sobre el circuito las intensidades (tensión, tensión diferencial, corriente, potencia, energía etc.) que se quieren obtener, *ejecutando Current Into Pin* que se encuentra en el menú *Markers*.
4. Se simula el circuito utilizando *Run* del menú *PSpice*, obteniendo directamente las formas de ondas de las intensidades marcadas. Los valores numéricos se obtienen por medio de *Display de Cursor de Trace* del menú *PSpice*. Para obtener los valores, se hace clic sobre el símbolo que aparece a su izquierda mostrándose dicho valor en la ventana *Probe Cursor*.

C. OrCAD Layout

OrCAD Layout es una poderosa herramienta para el diseño de tarjetas de circuito impreso, tiene todas las funciones automatizadas que usted necesita para fácilmente completar el diseño de su tarjeta.

A continuación se describe brevemente los pasos a seguir para el diseño de un circuito impreso:

1. Cuando se haya confeccionado el esquema del circuito libre de errores, se coloca el encapsulado (un encapsulado es el correspondiente físico del símbolo utilizado en el esquema) correspondiente a todos los elementos del esquema, bien mediante un fichero o manualmente con el comando *Edit*. De hecho podemos encontrarnos con varios encapsulados para un mismo símbolo: por ejemplo una resistencia que siempre se representa con un determinado símbolo puede presentar diferentes encapsulados según sea el tamaño, potencia, que vaya a adoptar en el montaje real.
2. Luego se pasara a crear el *Netlist* del esquema. En el se detalla como deben realizarse las conexiones, evitando errores en el trabajo final debido a conexiones incorrectas, quedando garantizado que las pistas de la tarjeta seguirán con fidelidad el cableado realizado en el archivo del esquema. Enseguida ingresamos a *OrCAD Layout* directamente desde el menú *Tools* o ejecutando el programa *OrCAD Layout*.
3. Una vez dentro de *OrCAD Layout* procedemos a realizar todos los pasos a seguir para el diseño de la tarjeta, entre los cuales tenemos: contorno de tarjeta, editar *Obstacle*, posición de los encapsulados, trazado de pistas, rellenar áreas, colocar textos. Y generar un archivo con todas las coordenadas (x, y), así como el diámetro de las brocas correspondientes, para enviar a la taladradora de control numérico, de modo que esta quede programada.
4. Una vez concluido el diseño, para realizar la tarjeta de circuito impreso de debe tener en cuenta la impresión de las hojas siguientes:
Componentes: Cara netamente de los dispositivos electrónicos (Chips, resistencias, condensadores, bobinas, etc.), para lo cual se tiene en cuenta su

encapsulado (*Footprint*) permitiendo que a la hora de soldar los componentes no se sobrepongan y queden posesionados estéticamente sobre la tarjeta. En esta cara también se encuentran las pistas de cobre que comienzan y terminan en esta cara; otras pistas de cobre las cuales se comunican con la cara de soldadura, a través de *Pads* pasantes metalizados; los *Pads* de los dispositivos electrónicos (encapsulados de montaje superficial), además de los dispositivos cuyos *Pads* utilizan agujeros y otras pistas de soldadura las cuales se comunican con la cara de soldadura, a través de un *Pad* pasante metalizado.

Soldadura: Cara netamente de soldadura; la cual esta constituida por las pistas de cobre que comienzan y terminan en esta cara; otras pistas de cobre las cuales se comunican con la cara de componentes, a través de *Pads* pasantes metalizados; los *Pads* de los (*Footprint*) que tienen agujero y otros de montaje superficial. En esta cara también se encuentran componentes mayormente de montaje superficial.

Mascara de soldadura: Esta se emplea para señalar las áreas que no deben ser cubiertos con barniz, ya que en ellos se realizara la soldadura; tanto en la cara de componentes como en la cara de soldadura.

Brocas: Muestra las distintas medidas de las brocas que serán utilizadas para realizar los agujeros de la tarjeta de circuito impreso.

CAPÍTULO III EL MICROCONTROLADOR PIC16C55

3.1. Descripción general

El microcontrolador PIC16C55 pertenece a la familia básica (PIC16C5X) de los microcontroladores de Microchip Technology Inc. de bajo costo, totalmente estático, alta performance, 8 bits, tecnología CMOS EPROM/OTP. Emplea la Arquitectura RISC con solo 33 instrucciones. Todas las instrucciones se ejecutan en un ciclo de instrucción excepto las de salto, altamente simétrico resultando en 2:1 la compresión de código sobre otros microcontroladores de 8 bits de la misma clase. Además el microcontrolador está equipado con características especiales que reducen el costo del sistema y requerimientos de energía.

El PIC16C55 está disponible en dos tipos de encapsulado: (1) Encapsulado sin ventana con memoria tipo OTP, y por tanto más barato, es idóneo para grandes volúmenes. (2) Encapsulado con ventana con EPROM son ideales para desarrollar aplicaciones, dado que su memoria de programa se puede borrar utilizando luz ultravioleta. También dispone de un bit de protección anticopia. Cuando se programa este bit, se hace imposible la lectura del código del microcontrolador programado.

Características del CPU

- Procesador de Arquitectura RISC avanzada.
- Juego de 33 instrucciones con 12 bits de longitud.

- Frecuencia de funcionamiento hasta 20 MHz.
- Memoria de programa (EPROM u OTP) de 512 palabras de 12 bits cada uno.
- Memoria de datos: 24 bytes tipo SRAM.
- Pila de 2 niveles.
- Modos de direccionamiento directo, indirecto y relativo.
- Palabra de configuración /Código de protección programable
- Seleccionable tipo de oscilador (LP, XT, HS y RC).
- Modo de bajo consumo o SLEEP.
- Baja potencia, alta velocidad, tecnología CMOS EPROM/OTP.
- Voltaje de alimentación comprendido entre 2 y 6,25 V.
- Bajo consumo (menos de 2mA a 5V/4MHz y menos de 1uA en modo SLEEP).

Características de los periféricos

- Hasta 20 líneas de entrada/salida (E/S): Puerto A, Puerto B y Puerto C.
- Temporizador perro guardián (WDT).
- TMR0: temporizador contador de 8 bits con predivisor de 8 bits.
- POR y BOR. PWRT y OST.

3.2. Arquitectura

La alta performance del microcontrolador PIC16C55 se puede atribuir a un número de características que comúnmente se encuentran en los microprocesadores con arquitectura RISC (Computador con Juego Reducido de Instrucciones). Estos incluyen: Arquitectura Harvard, Instrucciones simples, Estructura segmentada.

generador de reloj con el circuito de supervisión tipo perro guardián (WDT) y los registros de función especial PORTA, PORTB y PORTC que contienen los datos de los puertos de entrada o salida; así como los respectivos registros TRISX que programan sus líneas como entradas o salidas.

3.2.1. Asignación de pines

El diagrama de asignación de pines del microcontrolador PIC16C55-XT/P encapsulado en 28 pines se muestra en la figura 3.2. Algunos pines son multifuncionales, debido a que realizan diversas funciones según estén programados.

- **#MCLR/Vpp (28):** activa en nivel bajo Reset/entrada voltaje de programación.
- **OSC1/CLKIN (27):** entrada del cristal de cuarzo/entrada reloj externo.
- **OSC2/CLKOUT (26):** salida oscilador de cristal/salida con la cuarta parte de OSC1 ($F_{osc}/4$) en modo RC.
- **TOCKI (1):** entrada de reloj externo para el temporizador/contador TMR0.
- **Vdd (2):** entrada de alimentación, típico 5 VDC.
- **N/C (3) y (5):** no se conectan.
- **Vss (4):** conexión a tierra.

Puerto A

- **RA0 (6), RA1 (7), RA2 (8) y RA3 (9):** E/S digital, con entrada de nivel TTL.

Puerto B

- **RB0 (10), RB1 (11), RB2(12), RB3 (13), RB4 (14), RB5 (15), RB6 (16) y RB7(17):** E/S digital, con entrada de nivel TTL.

Puerto C

- **RC0 (18), RC1 (19), RC2 (20), RC3 (21), RC4 (22), RC5 (23), RC6 (24) y RC7 (25):** E/S digital, con entrada de nivel TTL.

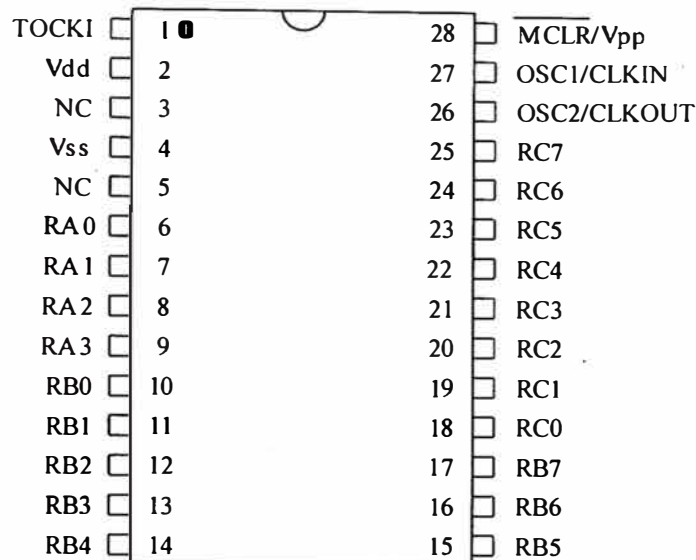


Figura 3.2 Diagrama de asignación de pines del microcontrolador PIC16C55.

3.2.2. Reloj/Ciclo de instrucción

La frecuencia de trabajo del microcontrolador es un parámetro fundamental a la hora de establecer la velocidad en la ejecución de instrucciones y el consumo de energía.

Cada ciclo de instrucción tarda en ejecutarse 4 ciclos de reloj y trabajando el microcontrolador a la frecuencia F_{osc} , el ciclo de instrucción está dado por:

$$T_{\text{ciclo de instrucción}} = 4 * T_{osc}; \quad T_{osc} = \left(\frac{1}{F_{osc}} \right) \quad \dots (3.1)$$

Los pulsos de reloj T_{osc} ingresan por el pin OSC1/CLKIN y se divide por 4 internamente, generando las señales Q1, Q2, Q3 y Q4, mostradas en la figura 3.3. Durante un ciclo de instrucción, que comprende las 4 señales mencionadas, se desarrollan las siguientes operaciones:

Q1: durante este pulso de reloj se incrementa el contador de programa PC.

Q4: durante este pulso de reloj se busca el código de la instrucción en la memoria del programa y se carga en el registro de instrucciones.

Q2-Q3: durante la activación de estas señales se produce la decodificación y la ejecución de la instrucción.

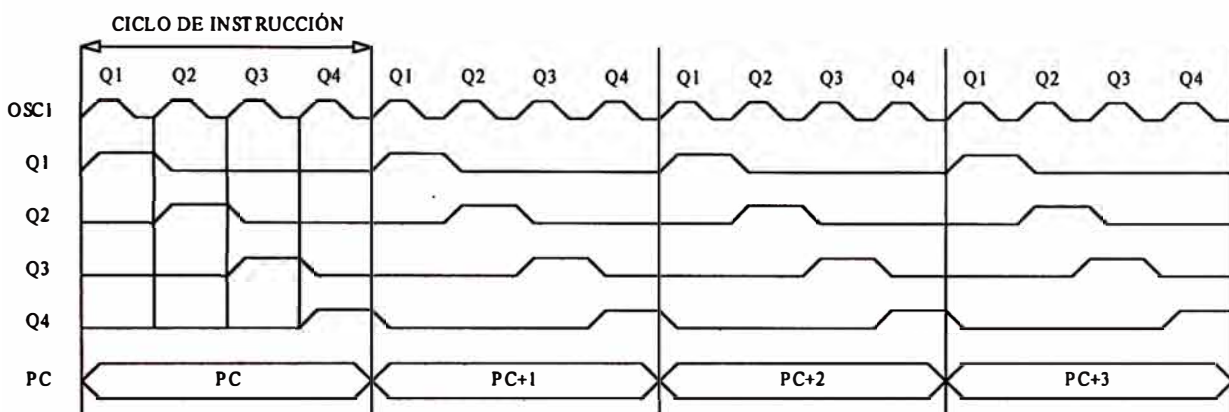


Figura 3.3 *Los pulsos del reloj externo (OSC1/CLKIN) se dividen en 4 formando las señales Q1, Q2, Q3 y Q4, que configuran un ciclo de instrucción.*

3.2.3. Flujo de instrucción /segmentación

Para conseguir ejecutar cada instrucción en un ciclo de instrucción, excepto las de salto que tardan el doble; se aplica la técnica de la segmentación, que consiste en realizar en paralelo las dos fases que comprende cada instrucción.

En realidad cada instrucción tarda en ejecutarse dos ciclos: en el primero se lleva a cabo la fase de búsqueda del código de la instrucción en la memoria de programa, y en el segundo se decodifica y se ejecuta. La estructura segmentada permite realizar al mismo tiempo la fase de ejecución de una instrucción y la de búsqueda de la siguiente instrucción. Cuando la instrucción ejecutada corresponde a un salto no se conoce cuál será la siguiente hasta que se complete, en esta situación se sustituye la

fase de búsqueda de la siguiente instrucción por un ciclo vacío, originando que las instrucciones de salto tarden en ejecutarse dos ciclos de instrucción.

3.3. Organización de memoria

Hay dos bloques de memoria en el PIC16C55. Éstas son: la memoria de programa (EPROM u OTP) y la memoria de datos. La memoria de datos dispone de memoria SRAM. Cada bloque tiene su propio sistema de buses, para que el acceso a cada bloque pueda ocurrir durante el mismo ciclo.

3.3.1. Memoria de programa

La memoria de programa del microcontrolador PIC16C55 consta de 512 palabras de 12 bits cada uno. La estructura de la memoria de programa se ilustra en la figura 3.4.

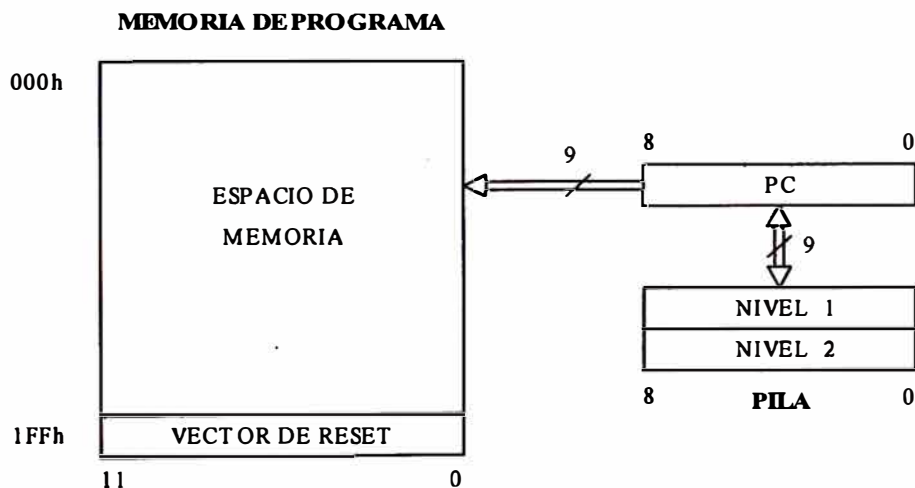


Figura 3.4 Organización de la memoria de programa.

A. Contador de programa (PC)

La secuencia de ejecución de las instrucciones se controla a través del denominado Contador de Programa (PC) utilizando direcciones de 9 bits. Cuando se ejecutan

instrucciones de salto, o cuando se invoca a una subrutina se deben tomar algunas precauciones. Los problemas pueden surgir cuando se asuma (error típico en la programación de estos dispositivos) que los datos se codifican en 8 bits, lo cual es imposible con operandos que tienen una anchura de 9 bits. El programador siempre debe tener en cuenta la anterior restricción relativa a la longitud de la palabra. Con la excepción de la instrucción GOTO, las instrucciones que alteran el flujo del programa solo se pueden utilizar en la mitad inferior del bloque de memoria (bit 8 = 0). Además, los bits 5 y 6 del registro STATUS siempre se encuentran con valor 0, debido a que el PIC16C55 solo cuenta con el banco 0.

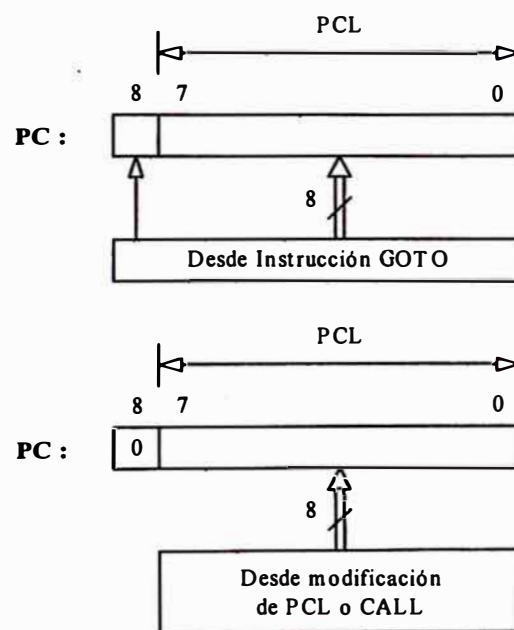


Figura 3.5 Muestra el PC como resultado de una operación, en las instrucciones GOTO y CALL.

B. Pila

La Pila es una zona aislada de las memorias de programa y la memoria de datos. Tiene una estructura LIFO, en la que el último valor guardado es el primero que sale. Tiene 2 niveles de profundidad cada uno de 9 bits.

El PIC16C55 dispone de dos registros de Pila que se caracterizan por su utilización desde el punto de vista hardware. Una instrucción CALL sitúa el valor del PC en curso, incrementado en una unidad, en el nivel 1 de la pila. A continuación, el nivel 1 de la pila se envía al nivel 2. Esta característica permite que el programa "salte" dos veces de su flujo normal. Si se anidan más llamadas a subrutinas, las direcciones de regreso se pierden. Una instrucción RETLW carga de nuevo el contenido del nivel 1 de la pila en el contador de programa, mientras que el contenido del nivel 2 de la pila se copia en el nivel 1. El nivel 2 retiene su valor.

C. Vector de Reset

Un Reset fuerza al PC a la dirección 1FFh; esta dirección se llama vector Reset. Cualquier Reset realizado limpia el contenido del registro PCL.

3.3.2. Memoria de datos

La Memoria de Datos tiene posiciones implementadas en SRAM, donde se alojan los registros operativos fundamentales en el funcionamiento del procesador y en el manejo de todos sus periféricos, además registros que el programador puede usar.

La RAM estática contienen Registros de Propósito General (GPR) y Registros de Función Especial (SFR), que se extiende hasta 1Fh (32 bytes). Las más bajas posiciones de cada banco son reservadas para los Registros de la Función Especial.

Los **registros de propósito general** son usados por el programador dependiendo de la aplicación; para registros temporales, constantes, cuenta, de información, test, etc. Este modelo cuenta con 24 registros de 8 bits cada uno. Los **registros de función especial** son usados por la CPU y los módulos periféricos, para controlar el

dirección es igual al contenido del registro FSR. La combinación del registro INDF y del registro FSR crea múltiples formas de manejar áreas de memoria.

B. Registro PCL

El Contador de Programa (PC) genera las direcciones de las posiciones de la memoria de programa. Para el microcontrolador PIC16C55, el contador de programa y su pila hardware de dos niveles asociada tienen una anchura de 9 bits, utilizando el bit 8 como extensión de los bits <7-0>.

Normalmente, el contador de programa se incrementa en una unidad después de la ejecución de cada instrucción. Sin embargo, las siguientes instrucciones hacen que el contador de programa tenga un comportamiento distinto; (1) En la instrucción **GOTO** los bits <8-0> se cargan directamente y se utilizan para indicar la dirección de destino; (2) La instrucción **CALL** difiere de la **GOTO** en que el bit 8 siempre se pone a 0. En consecuencia, el rango de direccionamiento de esta instrucción es limitado (ver figura 3.5). El resto de instrucciones de escritura en el PC funcionan de la misma forma que la instrucción **CALL**, esto es, borran el bit 8.

C. Registro STATUS

En la figura 3.7 se muestra y se describe la asignación de bits del registro STATUS. Los señalizadores C, D y Z de este registro proporcionan el estado de una operación aritmética realizada por la ALU. El señalizador #PD (bit 3) y el #TO (bit 4) indican el estado de inicialización (Reset), mientras que el PA0 (bit 5) y el PA1 (bit 6) para el PIC1655 siempre se encuentran en 0. Los bits #PD y #TO del registro de la palabra de estado no son afectados por una operación de escritura.

Si este registro se utiliza para indicar el resultado de una operación aritmética debe tenerse en cuenta que los bits de estado se ponen a uno después de la siguiente escritura.

Registro STATUS

---	PA1	PA0	#TO	#PD	Z	DC	C
bit 7							bit 0

Bit 7 No se encuentra implementado, se lee como 0.

RP1:RP0 Como en el PIC16C55 no existen bancos, siempre es 00. Ver figura 3.4.

#TO Time Out.
1: vale 1, después de la conexión de alimentación o al ejecutarse las instrucciones CLRWDT o SLEEP.

0: vale 0, cuando se produce el desbordamiento del WDT.

#PD Power down.
1: vale 1, después de la conexión de alimentación o al ejecutar la instrucción CLRWDT.

0: vale 0, al ejecutarse la instrucción SLEEP.

Z Señalizador de cero.
1: si vale 1, el resultado de una operación lógico- aritmético es 0.

0: si vale 0, el resultado de una operación lógico- aritmético no es 0.

DC Señalizador de llevada en el bit3. Igual que C, referido al bit3. De interés en operaciones en BCD.

C Señalizador de llevada en el bit de mas peso. Para las instrucciones de resta la correspondencia es inversa, es decir si vale 0 hay llevada.

1: si vale 1, indica llevada en el bit de mas peso en la operación suma.

0: si vale 0, no se ha producido llevada en la operación suma.

Figura 3.7 *Registro STATUS. Los bits 3 y 4 sólo se pueden leer, mientras los demás también se pueden escribir.*

D. Registro FSR

En el microcontrolador PIC16C55 los bits <4-0> seleccionan uno de los 32 registros disponibles en el modo de direccionamiento indirecto, esto es, utilizando el registro INDF. Los bits <7-5> son sólo de lectura y siempre están a 1.

Si no se utiliza el direccionamiento indirecto, el FSR se puede utilizar como un registro de propósito general de 5 bits de longitud.

3.4. Puertos de entrada/salida

Los Puertos de entrada/salida (E/S) constituyen la interfaz entre el software y los dispositivos hardware externos. En el PIC16C55 los Puertos son controlados por los registros PORTA, PORTB y PORTC, los cuales pueden leer o escribir. Una instrucción de lectura siempre lee el estado de un pin, independientemente de si éste está programado como entrada o salida. Después de una inicialización, todos los pines se definen como entradas (modo de alta impedancia).

3.4.1. Puerto A

Dispone de 4 líneas bidireccionales denominadas RA3-RA0 que admiten entrada TTL y salida CMOS; las demás líneas RA7-RA4 no están implementados y se leen siempre como 0. Los bits del registro TRISA configuran las líneas como entrada o salida y los datos que ingresan o salen se encuentran en el registro PORTA que ocupa la dirección 05h.

3.4.2. Puerto B

Cuenta con 8 líneas bidireccionales denominadas RB7-RB0 que admiten entrada TTL y salida CMOS, cuyo sentido de E/S se configura con el registro TRISB y los datos que ingresan o salen se encuentran en el registro PORTB que ocupa la dirección 06h. La figura 3.8 muestra el circuito equivalente de una línea de E/S.

3.4.3. Puerto C

Cuenta con 8 líneas bidireccionales denominadas RC7-RC0 que admiten entrada TTL y salida CMOS, cuyo sentido de E/S se configura mediante el registro TRISC y

los datos que ingresan o salen se encuentran en el registro PORTC que ocupa la dirección 07h. Ver figura 3.8.

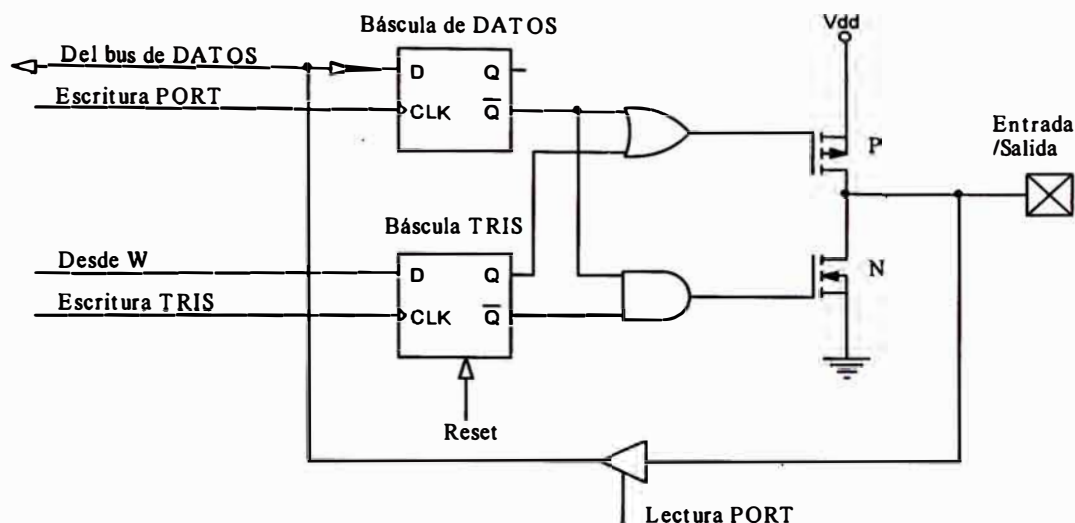


Figura 3.8 Circuito equivalente de una línea de E/S para los Puertos A, B y C.

3.4.4. Instrucción TRIS f

La instrucción **TRIS f** ($f = 5, 6, 7$) se utiliza para configurar las líneas de E/S de los Puertos A, B y C. Cada línea de un Puerto está representado por un bit en el respectivo registro TRIS. Un 1 en la correspondiente posición del bit configura la línea como entrada y un 0 configura la línea como salida. El registro TRIS f es del tipo de sólo escritura utilizando la instrucción TRIS f que sitúa su contenido a través del registro W. Después de un Reset, todos sus bits están a 1.

3.5. Temporizador/Contador TMR0

El registro TMR0 está ubicado en la dirección 01h de la memoria de datos, se incrementa con cada flanco. Tiene las siguientes características:

- Temporizador/contador ascendente de 8 bits, leíble y escribible.

- Reloj interno ($F_{osc}/4$) o externo (pin TOCKI).
- Elección del tipo de flanco en el reloj externo.
- Predivisor de la frecuencia de reloj programable.

En realidad, estos microcontroladores disponen de dos temporizadores, el TMR0 y el Temporizador perro guardián (WDT). El primero se encarga del control de tiempos y contaje de pulsos. El otro vigila que el programa no se cuelgue, para esto cada cierto tiempo comprueba si el programa se está ejecutando normalmente. En caso contrario, si el control esta detenido en un bucle infinito a la espera de algún acontecimiento que no se produce, el WDT avisa, lo que se traduce en un Reset que reinicializa todo el sistema.

A menudo el TMR0 y WDT precisan controlar largos intervalos de tiempo y necesitan aumentar la duración de los pulsos de reloj que les incrementa. Para cubrir este requisito se dispone de un circuito programable denominado divisor de frecuencia, que divide la frecuencia utilizada por diversos rangos. Para programar el comportamiento del TMR0, WDT y el divisor de frecuencia se utilizan algunos bits del registro OPTION_REG, y otros bits de la palabra de configuración.

Cuando funciona como temporizador conviene cargarle con el valor de los pulsos que se quiere temporizar, pero expresados en complemento a 2, incrementándose en cada pulso $F_{osc}/4$. De esta manera, al llegar el número de pulsos deseado, se desborda al pasar de FFh a 00h. Para calcular los tiempos a controlar con TMR0 se utilizan las siguientes fórmulas.

$$\text{Temporización} = 4 * T_{osc} * (\text{Valor en TMR0}) * (\text{Divisor}) \quad \dots (3.2)$$

$$\text{Valor en TMR0} = \text{Temporización} / (4 * T_{osc} * (\text{Divisor})) \quad \dots (3.3)$$

Cuando funciona como contador de sucesos, se incrementa la cuenta en cada pulso que se aplica al pin T0CKI. Al llegar al valor FFh se desborda el contador y con el siguiente pulso pasa a 00h.

En la figura 3.9 se ofrece el esquema de funcionamiento del TMR0. Se observa que hay un bloque que retrasa 2 ciclos el conteaje para sincronizar el momento del incremento producido por la señal aplicado al pin T0CKI con el que producen los pulsos internos del reloj. Cuando se escribe el TMR0 se retrasan 2 ciclos su reincremento y se pone a 0 el divisor de frecuencia.

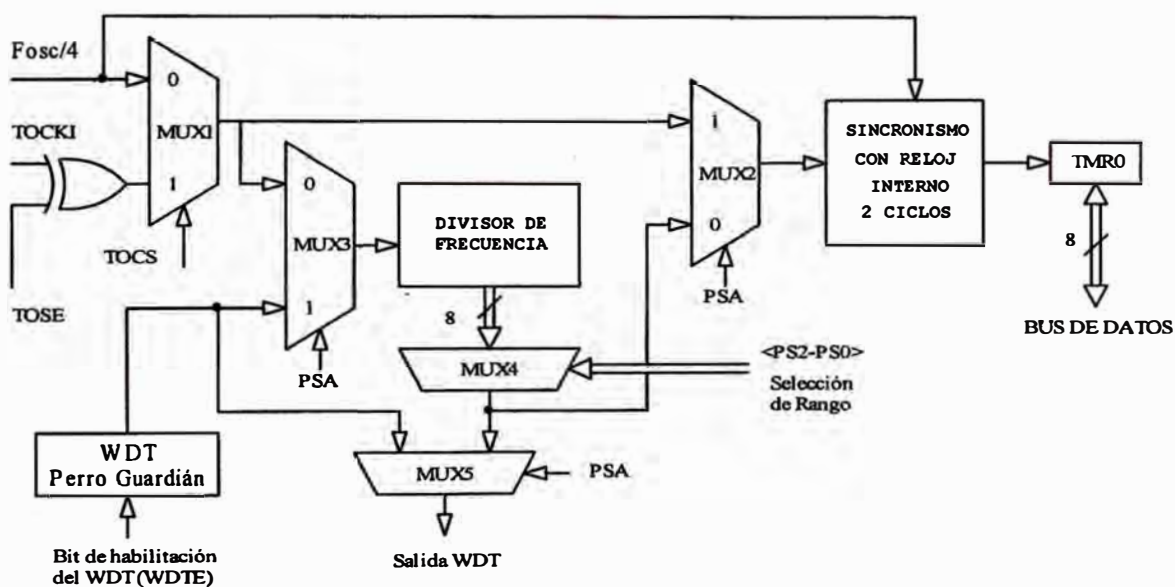


Figura 3.9 Esquema general del funcionamiento del TMR0.

Registro OPTION

Este registro se utiliza para programar el predivisor del temporizador/contador TMR0 o del temporizador perro guardián (WDT). El registro OPTION tiene una anchura de 6 bits y es del tipo de sólo escritura utilizando la instrucción OPTION que sitúa su contenido a través del registro W. Las funciones de cada uno de los bits en

particular se puede ver en la figura 3.10. En este caso, también todos los bits se sitúan a 1 después de un Reset.

Registro OPTION

---	---	TOCS	TOSE	PSA	PS2	PS1	PS0
bit 7							bit 0

Bit <7:6> No implementados. Se leen como 0.

TOCS Tipo de reloj para el TMR0.
1: pulsos introducidos a través de TOCKI (contador).
0: pulsos de reloj interno $F_{osc}/4$ (temporizador).

TOSE Tipo de flanco en TOCKI.
1: incremento del TMR0 cada flanco descendente.
0: incremento del TMR0 cada flanco ascendente.

PSA Asignación del divisor de frecuencia.
1: el divisor de frecuencia se le asigna al WDT.
0: el divisor de frecuencia se le asigna al TMR0.

PS2:PS0 Valor con el que actúa el divisor de frecuencia.

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

Figura 3.10 Asignación de funciones de los bits del registro OPTION.

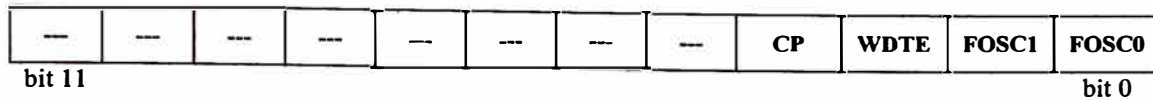
3.6. Características especiales

3.6.1. Palabra de configuración

Es una posición reservada de la memoria de programa que ocupa la dirección FFFh y que solo es accesible durante la programación (configuración de grabación) del PIC.

El valor de sus bits determina algunas características del sistema (ver figura 3.11).

Palabra de Configuración



Bit <11-4> No implementados. Se leen como 0.

CP Bit de protección del código de programa.
1: no hay protección.
0: hay protección.

WDTE Bit de habilitación del WDT.
1: WDT habilitado.
0: WDT no habilitado.

FOSC1:FOSC0 Selección del tipo de oscilador.

FOSC1:FOSC0		Tipo de oscilador
0	0	LP
0	1	XT
1	0	HS
1	1	RC

Figura 3.11 Asignación de los 12 bits de la Palabra de Configuración.

3.6.2. Oscilador

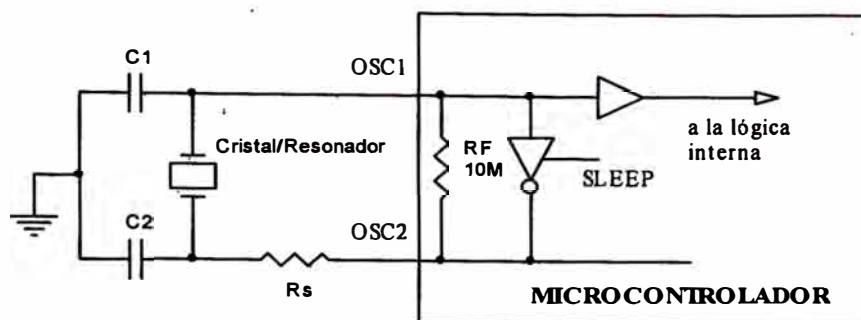
El microcontrolador admite cuatro tipos de osciladores (LP, XT, HS y RC) externos para aplicarles la frecuencia de funcionamiento Fosc. El tipo empleado debe especificarse en dos bits (FOSC1 y FOSC0) de la Palabra de Configuración.

En los tipos LP, XT o HS, un cristal de cuarzo o resonador cerámico es conectado entre los pines OSC1/CLKIN y OSC2/CLKOUT, ver figura 3.12. Los condensadores C1 y C2 se eligen de la tabla 3.1 según sea el tipo de oscilador.

A. Oscilador tipo LP: Oscilador de bajo consumo con cristal de cuarzo o resonador cerámico, diseñado para en el rango de frecuencias de 32 a 200 KHz.

B. Oscilador tipo XT: Oscilador de cristal o resonador para frecuencias comprendidas entre 100 KHz y 4MHz.

C. Oscilador tipo HS: Oscilador de alta velocidad comprendido entre 4 y 20 MHz, basado en un cristal cuarzo o un resonador cerámico.



R_s : necesario en algunas versiones tipo HS, en otros tipos corto circuito.

Figura 3.12 Esquema del general para los osciladores tipo LP, XT y HS. El resistor R_s sólo es necesario en algunas versiones tipo HS.

Tipo Oscilador	Frecuencia de Resonancia	C1	C2
LP	32 KHz	15 pF	15 pF
	200 KHz	15-30 pF	15-82 pF
XT	200 KHz	15-30 pF	100-200 pF
	1 MHz	15-30 pF	15-30 pF
	4 MHz	15-47 pF	15-47 pF
HS	4 MHz	15-30 pF	15-30 pF
	8 MHz		
	20 MHz		

Tabla 3.1 Elección de C1 y C2 para cristal de cuarzo.

D. Oscilador tipo RC: Se trata de un oscilador de bajo costo formado por una resistencia y un condensador, conectados al exterior del microcontrolador. Proporciona una estabilidad mediocre de la frecuencia, cuyo valor depende de los valores de los elementos de la red R-C.

3.6.3. Reset

Existen diversas causas que reinician (Reset) el microcontrolador PIC16C55:

1. Reset por conexión de la alimentación (*POR: Power-On Reset*).
2. Activación del pin #MCLR, en operación normal.

3. Activación del pin #MCLR, estando el PIC en modo SLEEP.
4. Reset por desbordamiento del WDT en operación normal.
5. Reset por desbordamiento del WDT, estando el PIC en modo SLEEP.

El PIC16C55 incorpora en el chip un circuito POR que proporciona un reset interno; para usar esta característica simplemente conectar el pin #MCLR/Vpp a Vdd. Para muchas situaciones es recomendable un reset externo formado con un circuito RC, tal como se usa en el circuito de telemando.

El microcontrolador permanece en el estado de inicialización durante el tiempo de Reset del dispositivo (DRT) y/o mientras que la entrada #MCLR se mantenga a nivel lógico bajo. El DRT tiene aproximadamente una duración de 18 ms y opera con un oscilador RC interno propio; manteniéndose durante este tiempo el procesador en Reset. El DRT inicia su cuenta al detectar nivel alto en el pin #MCLR, al terminar su tiempo de cuenta termina el Reset, pasando el procesador a funcionamiento normal.

También se produce Reset por una caída de voltaje en Vdd (BOR: *Brown Out Reset*), el cual se protege con circuito externo.

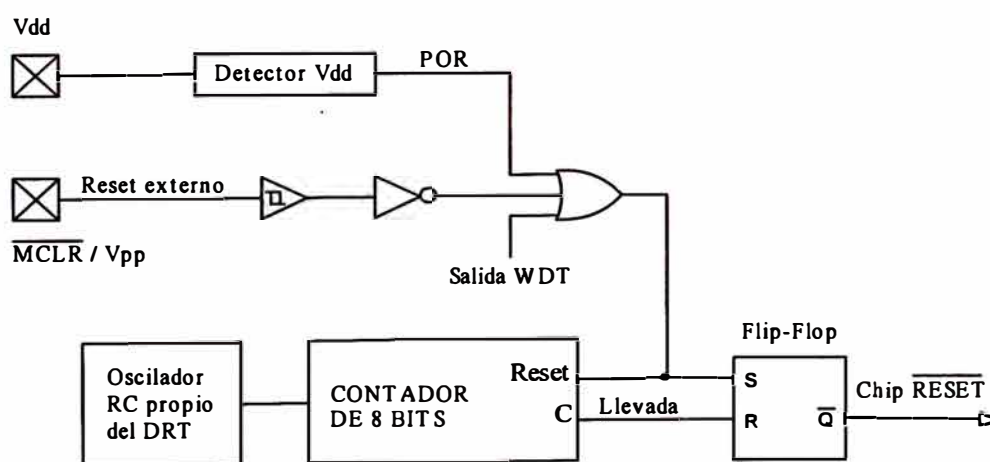


Figura 3.13 Esquema para la generación de Reset, según la causa que lo produce (tensión de alimentación, #MCLR y desbordamiento del WDT).

Durante una condición de reinicialización, el estado del microcontrolador viene definido de la siguiente forma:

- El oscilador está funcionando o comenzando a funcionar (encendido o arranque desde el modo SLEEP).
- Todos los pines de E/S de los puertos se sitúan en el modo de alta impedancia al programar los registros TRIS todos a 1 (esto es, todas las líneas como entrada).
- El contador de programa se sitúa todo a 1, esto es 1FFh en el PIC16C55.
- El registro OPTION se sitúa todo a 1.
- El temporizador perro guardián y su predivisor se ponen a 0.
- Los tres bits <7-6> del registro STATUS se ponen a 0 y permanecen así para el PIC16C55.
- La señal CLKOUT del pin OSC2 se sitúa a nivel bajo (solo en los microcontroladores con oscilador tipo RC).

3.6.4. Temporizador perro guardián

Se trata de un contador interno de 8 bits, su control de tiempos es independiente del TMR0 y esta basado en una simple red RC. Su funcionamiento es opcional y puede bloquearse para que no funcione programando el bit WDTE (WDTE en 1, se activa) de la palabra de configuración.

Para evitar que se desborde el WDT hay que refrescarle previamente, lo cual consiste en ponerlo a cero mediante las instrucciones CLRWDT y SLEEP. Este temporizador es muy útil en sistemas de control automáticos donde los fallos pueden desencadenar situaciones críticas. El tiempo antes de que WDT se desborde esta comprendido entre 9 y 30 ms; si se requiere mayor tiempo se utiliza el predivisor.

La instrucción CLRWDT borra el WDT y reinicia su cuenta; sin embargo la instrucción SLEEP además de borrar WDT detiene el sistema y lo pone en estado de reposo o de bajo consumo. Si no se desactiva el WDT al entrar en modo de reposo, al completar su cuenta provocara un Reset y sacará al PIC del modo de bajo consumo.

3.6.5. Modo SLEEP o de bajo consumo

El modo de funcionamiento de bajo consumo se activa ejecutando la instrucción SLEEP. Esta instrucción inicializa el temporizador perro guardián (si fue habilitado), borra el bit #PD del registro STATUS, activa el bit #TO y para el oscilador del reloj. Los puertos de E/S mantienen el estado que tenían antes de ejecutar la instrucción SLEEP. Para obtener el mínimo consumo posible en este modo, todos los pines de E/S deben estar a un nivel fijo, esto es, conectados a Vdd o Vss. Cuando se para el oscilador del reloj, el TMR0 también se para.

El dispositivo puede despertar (de ahí el nombre de la instrucción) y pasar a ejecutar la instrucción direccionada por PC+1 por las siguientes causas: (1) Al finalizar el tiempo del temporizador perro guardián, y (2) Por la detección de un flanco ascendente al final de un nivel bajo aplicado a la entrada #MCLR. En ambos casos, el microcontrolador PIC invertirá un periodo de arranque del oscilador antes de que el programa se inicie.

El bit #PD del registro de estado se puede utilizar para determinar si el procesador está en secuencia de encendido, o en secuencia de salida del modo de bajo consumo. De forma análoga, el bit #TO nos indica si la salida del modo de bajo consumo se debe a una señal #MCLR externa o a una finalización del período del temporizador perro guardián.

3.7. Repertorio de instrucciones

Las instrucciones constan de dos partes: un código de operación (la instrucción propiamente dicha) y un operando (una posición de memoria o un registro). El operando es el objeto del código de operación.

El microcontrolador PIC16C55 trabaja con palabras de 12 bits de longitud que contienen el código de la instrucción y los operandos. Esta configuración tiene sus ventajas y sus desventajas. Por ejemplo, la estructura de 12 bits limita a 33 el número de posibilidades para el código de instrucción (ver figura 3.14). Por otra parte, tiene la ventaja de ofrecer un procesado muy rápido de la instrucción dado que toda la información se capta en un solo ciclo.

Tipos de Formato

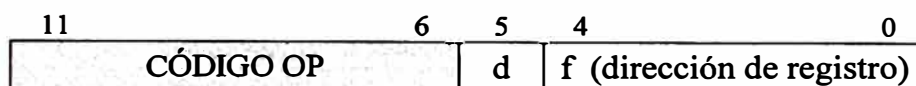
Las instrucciones de los PIC16C55 tienen 12 bits de longitud. El formato de la instrucción se divide en diferentes campos, referenciando cada uno a operandos o elementos que maneja la instrucción en la operación que realiza en el procesador. Se describen dichos campos:

1. Campo del código OP: Los bits de este campo sirven para definir la operación que realiza la instrucción.
2. Campo de los operandos fuente (f) y destino (d): Estos campos de bits definen los registros que actúan como operandos en la instrucción.
3. Campo de operando inmediato o literal (k): Es un campo de bits que indica el valor de un operando.
4. Campo que referencia a un bit (b): Campo de 3 bits que indica la posición de un bit concreto dentro de un registro de 8 bits.

5. Campo de la dirección del salto: En las instrucciones CALL y GOTO hay un campo de bits que contiene la dirección de la siguiente instrucción que hay que ejecutar. Dicho campo se carga en el PC en las instrucciones de salto incondicional.

Los diversos formatos que admiten las instrucciones de los microcontroladores se clasifican en 3 grandes grupos, según el tipo de operación que desarrollan.

- A. Operaciones orientadas a manejar registros de tamaño byte:** Las instrucciones de este grupo presentan 3 campos y tienen como sintaxis: **nemónico f,d**.



d = 1, el registro destino es f; d = 0, el registro destino es W

- B. Operaciones orientadas a manejar bits:** Las instrucciones de este grupo presentan 3 campos y tienen como sintaxis: **nemónico f,b**.



- C. Operaciones orientadas al literal y a control:** Las instrucciones de este grupo presentan 2 campos, se presentan dos casos.

Formato general



Formato para las instrucciones CALL y GOTO



NEMÓNICOS	PARÁMETROS	OPERACIÓN	CICLOS	FORMATO 12 BITS	SEÑALIZADOR
INSTRUCCIONES QUE MANEJAN REGISTROS					
addwf	f, d	Suma W con f	1	0001 11df ffff	C, DC, Z
andwf	f, d	AND de W con f	1	0001 01df ffff	Z
clrf	F	Borrado de f	1	0000 011f ffff	Z
clrw		Borrado de W	1	0000 0100 0000	Z
comf	f, d	Complemento de f	1	0010 01df ffff	Z
decf	f, d	Decremento de f	1	0000 11df ffff	Z
decfsz	f, d	Decremento de f; brinca si es 0	1	0010 11df ffff	Z
incf	f, d	Incremento de f	1	0010 10df ffff	Z
incfsz	f, d	Incremento de f; brinca si es 0	1	0011 11df ffff	Z
iorwf	f, d	OR de W con f	1	0001 00df ffff	Z
movf	f, d	Movimiento de f	1	0010 00df ffff	Z
movwf	f, d	Movimiento de W a f	1	0000 001f ffff	
nop	f, d	No operación	1	0000 0000 0000	
rlf	f, d	Rotación de f a la izquierda	1	0011 01df ffff	C
rrf	f, d	Rotación de f a la derecha	1	0011 00df ffff	C
subwf	f, d	Resta de W a f	1	0000 10df ffff	C, DC, Z
swapf	f, d	Intercambio de 4 bits de +peso por los 4 bits de -peso	1	0011 10df ffff	
xorwf	f, d	OR exclusivo de W con f	1	0001 10df ffff	Z
INSTRUCCIONES QUE MANEJAN BITS					
bcf	f, b	Puesta a 0 del bit b de f	1	0100 bbbf ffff	
bsf	f, b	Puesta a 1 del bit b de f	1	0101 bbbf ffff	
btfs	f, b	Test del bit b de f; brinca si 0	1 (2)	0110 bbbf ffff	
btfs	f, b	Test del bit b de f; brinca si 1	1 (2)	0111 bbbf ffff	
INSTRUCCIONES QUE MANEJAN LITERAL Y DE CONTROL					
andlw	k	AND de k con W	1	1110 kkkk kkkk	Z
call	k	Llamada a subrutina	2	1001 kkkk kkkk	
clrwdt		Borrado del Watchdog	1	0000 0000 0100	#TO,#PD
goto	k	Salto incondicional	2	101k kkkk kkkk	
iorlw	k	OR de k con W	1	1101 kkkk kkkk	Z
movlw	k	Movimiento de k a W	1	1100 kkkk kkkk	
option		Cargar registro option	1	0000 0000 0010	
retlw	k	Retorno devolviendo k en W	2	1000 kkkk kkkk	
sleep		Microcontrolador en reposo	1	0000 0000 0011	#TO,#PD
tris	k	Cargar registro tris	1	0000 0000 0fff	
xorlw	k	OR exclusivo de k con W	1	1111 kkkk kkkk	Z

Figura 3.14 Principales características de las 33 instrucciones que componen el repertorio del microcontrolador PIC16C55.

3.8. Características eléctricas

Se debe señalar que valores por encima del listado en “Rangos Máximos Absolutos” (ver tabla 3.2) puede causar daño permanente al dispositivo. Solo se señala un rango y no es implícito el funcionamiento del dispositivo a esos u otras condiciones por encima de los indicados en el listado de especificaciones de operación; pudiéndose dañar el dispositivo al estar expuesto a condiciones de rango máximo por periodos largos.

Mayor información se encuentra disponible en www.microchip.com.

Características	Rango
Temperatura ambiente	- 55 °C a 125 °C
Temperatura de almacenamiento	- 65 °C a 150 °C
Frecuencia de trabajo	20 MHz
Voltaje en Vdd con respecto a Vss	0 V a 7 V
Voltaje en #MCLR con respecto a Vss	0 a 14 V
Voltaje de grabación	12 V a 14 V
Voltaje en otros pines respecto a Vss	- 0, 8V a (Vdd + 0,6 V)
Potencia total de disipación	800 mW
Máxima corriente de salida del pin Vss	150 mA
Máxima corriente en el pin Vdd	100 mA
Máxima corriente en la entrada del pin TOCKI	± 500 uA
Entrada de corriente clamp, I_{IK} ($V_I < 0$ ó $V_I > V_{dd}$)	± 20 mA
Salida de corriente clamp, I_{OK} ($V_O < 0$ ó $V_O > V_{dd}$)	± 20 mA
Máxima corriente absorbida por cualquier pin E/S	25 mA
Máxima corriente suministrada por cualquier pin E/S	20 mA
Máxima corriente absorbida por una línea de E/S del Puerto A, B ó C	50 mA
Máxima corriente suministrada por una línea de E/S del Puerto A, B ó C	40 mA

Tabla 3.2 Muestra los valores de “Rangos Máximos Absolutos”, los cuales se deben tener en cuenta para cada familia de microcontrolador.

CAPÍTULO IV EL MICROCONTROLADOR PIC16F876

4.1. Descripción general

Bajo la denominación de PIC16F87X se hace referencia a una subfamilia de microcontroladores PIC de la gama media, al cual pertenece el microcontrolador PIC16F876. Se identifica por tener memoria de programa tipo FLASH, la cual se puede leer y escribir directamente con las instrucciones sin la necesidad de un programador; la memoria de datos tiene implementados posiciones en memoria tipo SRAM y otras en tipo EEPROM; y una serie de recursos semejantes a los modelos mas potentes PIC16C73/74/76/77, que tienen el inconveniente de que su memoria de programa es de tipo EPROM.

Características del CPU

- Procesador de Arquitectura RISC avanzada.
- Juego de 35 instrucciones con 14 bits de longitud. Todas ellas se ejecutan en un ciclo de instrucción, excepto las de salto que se ejecutan en dos ciclos.
- Frecuencia de funcionamiento hasta 20 MHz.
- Memoria de programa tipo FLASH de 8K palabras de 14 bits.
- Memoria de datos: 368 bytes tipo SRAM y 256 bytes tipo EEPROM.
- Encapsulado compatible con el PIC16C76.
- Hasta 14 fuentes de interrupción internas y externas.

- Pila de 8 niveles.
- Modos de direccionamiento directo, indirecto y relativo.
- Lectura y escritura de la memoria de programa.
- Palabra de configuración/código de protección programable
- Seleccionable tipo de oscilador.
- Modo de bajo consumo o SLEEP.
- Baja potencia, alta velocidad, tecnología CMOS FLASH/EEPROM.
- Voltaje de alimentación comprendido entre 2 y 5,5 V.
- Bajo consumo (menos de 2mA a 5V/4MHz y menos de 1uA en modo SLEEP).
- Programación serie en circuito (ICSP).
- Programación con voltaje bajo (LVP).
- Depuración en circuito.

Características de los Periféricos

- Hasta 22 líneas de entrada/salida (E/S): Puerto A, Puerto B y Puerto C.
- Temporizador perro guardián (WDT).
- TMR0: temporizador contador de 8 bits con predivisor de 8 bits.
- TMR1: temporizador contador de 16 bits con predivisor.
- TMR2: temporizador contador de 8 bits con predivisor y postdivisor.
- Dos módulos de Captura, Comparación y PWM (CCP).
- Convertidor A/D de 10 bits para 5 canales de entrada.
- POR y BOR. PWRT y OST.
- Puerto serie síncrono (MSSP) con SPI e I²C
- USART.

4.2. Arquitectura

La alta performance de los microcontroladores de Microchip Technology se puede atribuir a un número de características que comúnmente se encuentran en los microprocesadores con arquitectura RISC (Computador con Juego Reducido de Instrucciones). Estos incluyen:

- Arquitectura Harvard.
- Longitud de instrucción variable.
- Instrucciones simples.
- Estructura segmentada.
- Solo un ciclo de instrucción.
- Arquitectura de registros.
- Reducido juego de instrucciones.
- Ortogonalidad de instrucciones.

La arquitectura interna del PIC16F876 se presenta en la figura 4.1 y consta de 8 bloques fundamentales.

1. Memoria de programa FLASH de 8K x 14 bits.
2. Memoria de datos implementada en dos áreas: RAM estática, donde se alojan 75 bytes para registros de función especial (SFR) y 368 bytes para registros de propósito general (GPR), y otro tipo EEPROM de 256 bytes.
3. Decodificador de instrucciones y unidad de control.
4. Camino de datos con una ALU de 8 bits y un registro de trabajo W.
5. Direccionamiento de la memoria de programa en base al contador de programa (PC) ligado a una Pila de 8 niveles de profundidad.
6. Direccionado directo e indirecto de la memoria de datos SRAM.

7. Diversos recursos conectados al bus de datos, tales como los puertos de entrada/salida (Puerto A, B y C), temporizadores (TMR0, TMR1 y TMR2), módulo de captura-comparación y anchura de pulso (CCP1 y CCP2), módulo convertidor A/D, puerto serie síncrono (MSSP) con SPI e I²C y USART.
8. Base de tiempos y circuitos auxiliares.

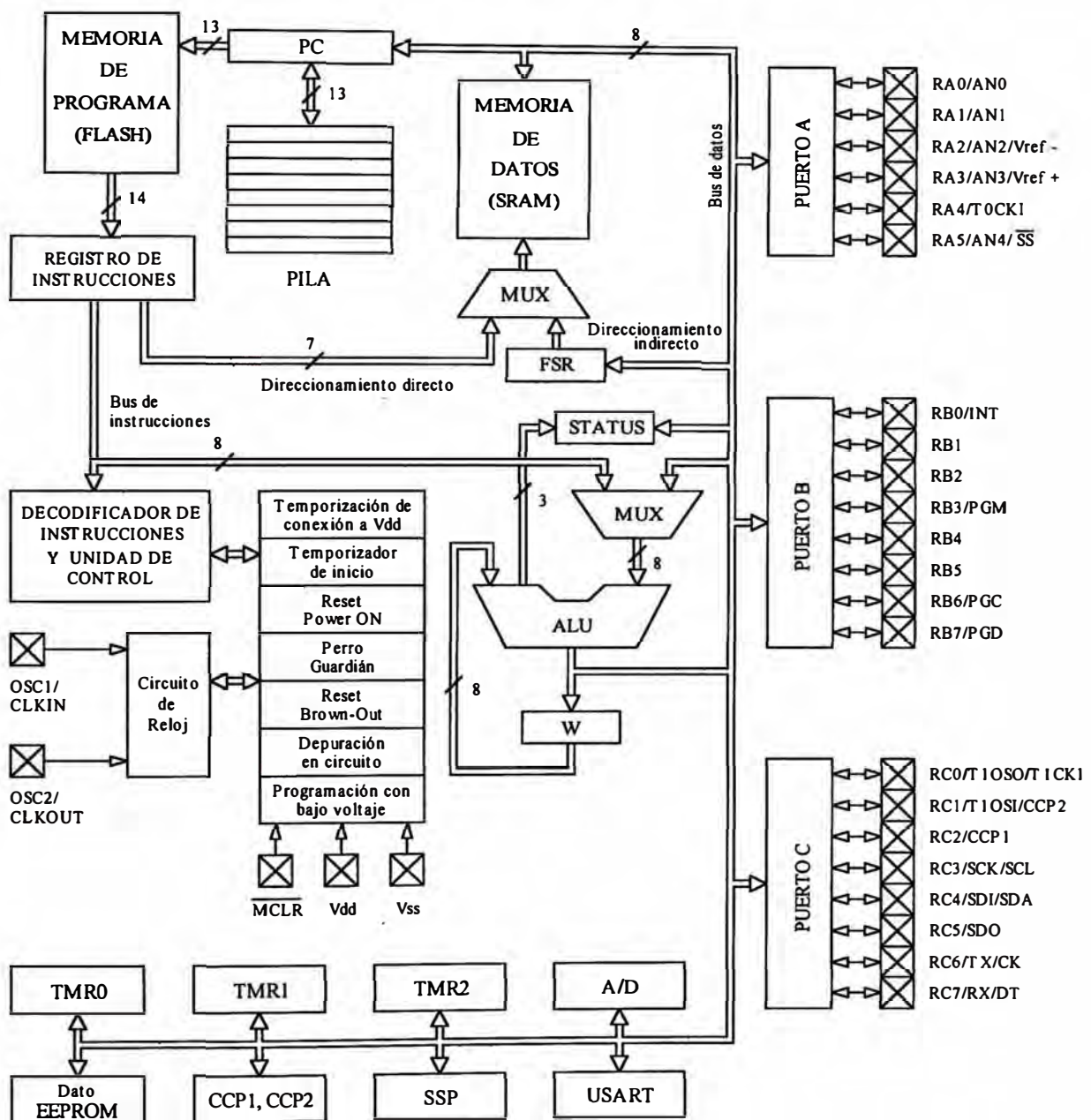


Figura 4.1 Arquitectura interna del microcontrolador PIC16F876.

4.2.1. CPU

La unidad central de proceso (CPU: *Unit Central Proceso*) que contiene todo el circuito de la unidad de control y la unidad lógica aritmética (ALU), se considera como el cerebro del microcontrolador. Es el responsable de traer la instrucción correcta, decodificar esa instrucción, y luego ejecutarla. La ALU completa la ejecución de la instrucción en operaciones lógicas–aritméticas.

La CPU controla el bus de direcciones y el bus de instrucciones de la memoria de programa, el bus de direcciones y el bus de datos de la memoria de datos, el bus de control que manejan los periféricos y recursos auxiliares, y el acceso a la Pila.

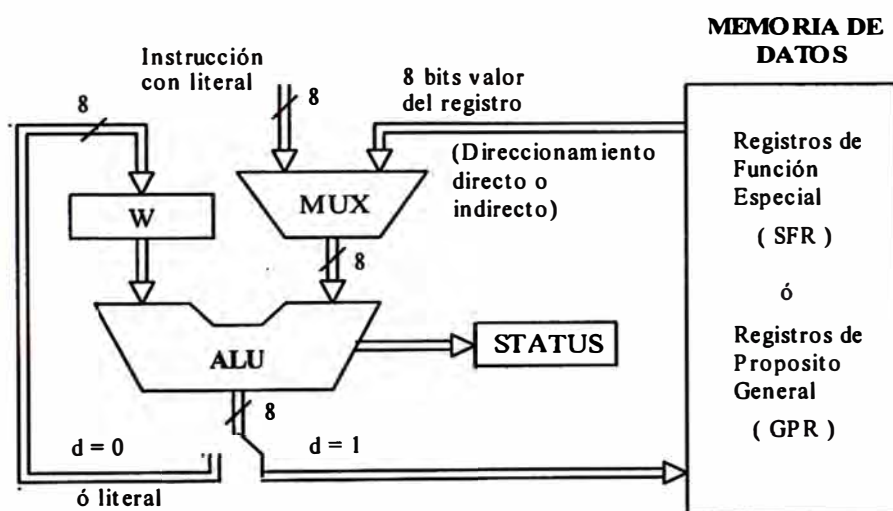


Figura 4.2 *Un operando de la ALU lo envía el registro W y el otro lo selecciona un multiplexor desde la propia instrucción con literal (k) o desde una de las posiciones de la memoria de datos.*

La CPU tiene una ALU de 8 bits y un registro de trabajo W de 8 bits, que ocupa la dirección 200h. La ALU es de propósito general, ejecuta funciones lógicas y aritméticas entre el dato W y cualquier registro fuente (f).

En los dos operandos de una instrucción, típicamente un operando es el registro de trabajo (W); el otro operando es un registro f o un valor literal k (figura 4.2).

Dependiendo de la instrucción ejecutada, el ALU puede afectar los valores de los bits de los señalizadores: de llevada C (STATUS<0>), llevada en el bit3 DC (STATUS<1>) y cero Z (STATUS<2>).

4.2.2. Asignación de pines

El diagrama de asignación de pines del microcontrolador PIC16F876-20/SP encapsulado en 28 pines se muestra en la figura 4.3. Los pines son multifuncionales, debido a que realizan diversas funciones según estén programados.

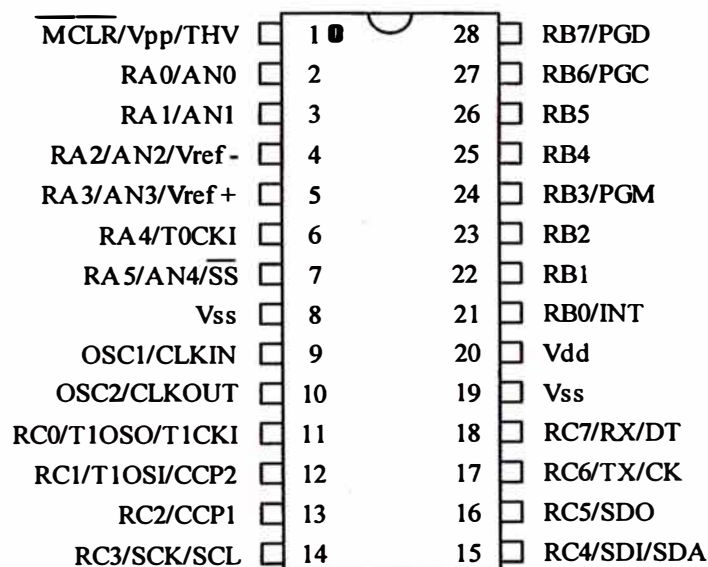


Figura 4.3 Diagrama de asignación de pines del microcontrolador PIC16F876.

- **OSC1/CLKIN (9):** entrada del cristal de cuarzo/entrada reloj externo.
- **OSC2/CLKOUT (10):** salida oscilador de cristal conectado en modo oscilador/salida con la cuarta parte de OSC1 ($F_{osc}/4$) en modo RC.
- **#MCLR/Vpp/THV (1):** activa en nivel bajo Reset/entrada voltaje de programación/voltaje alto en el modo test.
- **Vss (8, 19):** conexión a tierra.

- **Vdd (20):** entrada de alimentación, típico 5 V D.C.

Puerto A

- **RA0/AN0 (2):** E/S digital/entrada analógica al convertidor A/D canal 0.
- **RA1/AN1 (3):** E/S digital/entrada analógica al convertidor A/D canal 1.
- **RA2/AN2/Vref - (4):** E/S digital/entrada analógica al convertidor A/D canal 2/voltaje de referencia negativo.
- **RA3/AN3/Vref + (5):** E/S digital/entrada analógica al convertidor A/D canal 3/voltaje de referencia positivo.
- **RA4/TOCKI (6):** E/S digital/entrada de reloj externo para alimentar el temporizador–contador TMR0.
- **RA5/AN4/#SS (7):** E/S digital/entrada analógica al convertidor A/D canal 4/selección como esclavo del puerto serie síncrono.

Puerto B

- **RB0/INT (21):** interrupción cambio de nivel. E/S digital/entrada de petición de interrupción externo.
- **RB1 (22):** interrupción cambio de nivel. E/S digital.
- **RB2 (23):** interrupción cambio de nivel. E/S digital.
- **RB3/PGM (24):** E/S digital/entrada de voltaje bajo para programación.
- **RB4 (25):** interrupción cambio de nivel. E/S digital.
- **RB5 (26):** interrupción cambio de nivel. E/S digital.
- **RB6/PGC (27):** E/S digital/depuración en circuito. Reloj en la programación.
- **RB7/PGD (28):** E/S digital/depuración en circuito. Dato en la programación.

Puerto C

- **RC0/T1OSO/T1CKI (11):** E/S digital/salida oscilador TMR1/entrada de reloj para TMR1.
- **RC1/T1OSI/CCP2 (12):** E/S digital/entrada oscilador TMR1/entrada módulo captura 2. Para módulo comparación 2 y módulo PWM2 salida.
- **RC2/CCP1 (13):** E/S digital/entrada módulo captura 1. Para módulo comparación 1 y módulo PWM1 salida.
- **RC3/SCK/SCL (14):** E/S digital, entrada reloj serie síncrono/salida en los modos SPI e I²C.
- **RC4/SDI/SDA (15):** E/S digital/entrada datos modo SPI/datos de E/S modo I²C.
- **RC5/SDO (16):** E/S digital/salida datos modo SPI.
- **RC6/TX/CK (17):** E/S digital/transmisor USART asíncrono/reloj del síncrono.
- **RC7/RX/DT (18):** E/S digital/receptor USART asíncrono/datos del síncrono.

4.2.3. Reloj/ciclo de instrucción

La frecuencia de trabajo del microcontrolador es un parámetro fundamental a la hora de establecer la velocidad en la ejecución de instrucciones y el consumo de energía.

Cada ciclo de instrucción tarda en ejecutarse 4 ciclos de reloj y trabajando el microcontrolador a la frecuencia F_{osc} , el ciclo de instrucción esta dado por:

$$T_{ciclo\ de\ instrucción} = 4 * T_{osc} \quad \dots (4.1)$$

$$T_{osc} = \left(\frac{1}{F_{osc}} \right) \quad \dots (4.2)$$

Los pulsos de reloj T_{osc} ingresan por el pin OSC1/CLKIN y se divide por 4 internamente, generando las señales Q1, Q2, Q3 y Q4, mostradas en la figura 4.4.

Durante un ciclo de instrucción, que comprende las 4 señales mencionadas, se desarrollan las siguientes operaciones:

Q1: durante este pulso de reloj se incrementa el contador de programa PC.

Q4: durante este pulso de reloj se busca el código de la instrucción en la memoria del programa y se carga en el registro de instrucciones.

Q2-Q3: durante la activación de estas señales se produce la decodificación y la ejecución de la instrucción.

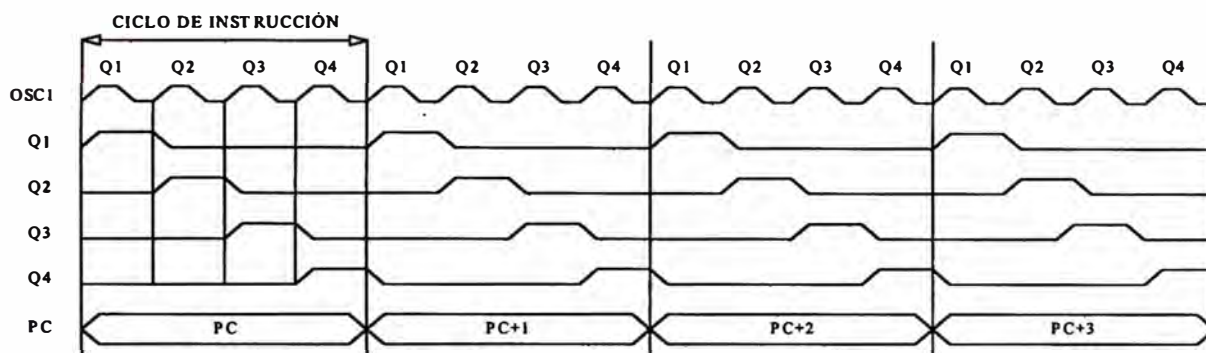


Figura 4.4 *Los pulsos del reloj externo (OSC1/CLKIN) se dividen en 4 formando las señales Q1, Q2, Q3 y Q4, que configuran un ciclo de instrucción.*

4.2.4. Flujo de instrucción/segmentación

Para conseguir ejecutar cada instrucción en un ciclo de instrucción, excepto las de salto que tardan el doble; se aplica la técnica de la segmentación, que consiste en realizar en paralelo las dos fases que comprende cada instrucción.

En realidad cada instrucción tarda en ejecutarse dos ciclos: en el primero se lleva a cabo la fase de búsqueda del código de la instrucción en la memoria de programa, y en el segundo se decodifica y se ejecuta. La estructura segmentada permite realizar al mismo tiempo la fase de ejecución de una instrucción y la de búsqueda de la siguiente instrucción. Cuando la instrucción ejecutada corresponde a un salto no se conoce cuál será la siguiente hasta que se complete, en esta situación se sustituye la

fase de búsqueda de la siguiente instrucción por un ciclo vacío, originando que las instrucciones de salto tarden en ejecutarse dos ciclos de instrucción.

4.3. Organización de la memoria

Hay dos bloques de memoria en el PIC16F876. Éstas son: la memoria de programa de tipo FLASH y la memoria de datos. La memoria de datos dispone de memoria SRAM y memoria EEPROM. Cada bloque tiene su propio sistema de buses, para que el acceso a cada bloque pueda ocurrir durante el mismo ciclo.

4.3.1. Memoria de programa

La memoria FLASH en la que se graba el programa de aplicación, tiene una capacidad de 8K palabras de 14 bits cada una. Dicha memoria esta dividida en paginas de 2K palabras de 14 bits y esta direccionado con el PC, que tiene un tamaño de 13 bits. El vector de Reset ocupa la dirección 0000h y el vector de interrupción la dirección 0004h, tal como se muestra en la figura 4.5.

A. Vector de Reset

Un Reset fuerza al PC a la dirección 0000h; esta dirección se llama vector Reset. Cualquier Reset realizado limpia el contenido del registro PCLATH. Esto significa que cualquier bifurcación a la dirección del vector Reset ordena saltar a esa dirección en la Página 0 de la memoria de programa.

B. Vector de interrupción

Cuando se reconoce una interrupción, el PC se fuerza a la dirección 0004h; esta dirección se llama vector de interrupción. Cuando el PC se fuerza al vector de

interrupción, el registro de PCLATH no es modificado. Antes de que el registro de PCLATH sea modificado por la rutina de servicio a la interrupción (ISR) los contenidos del PCLATH pueden necesitar ser guardados, para que pueda restaurarse antes de volver a la ISR.

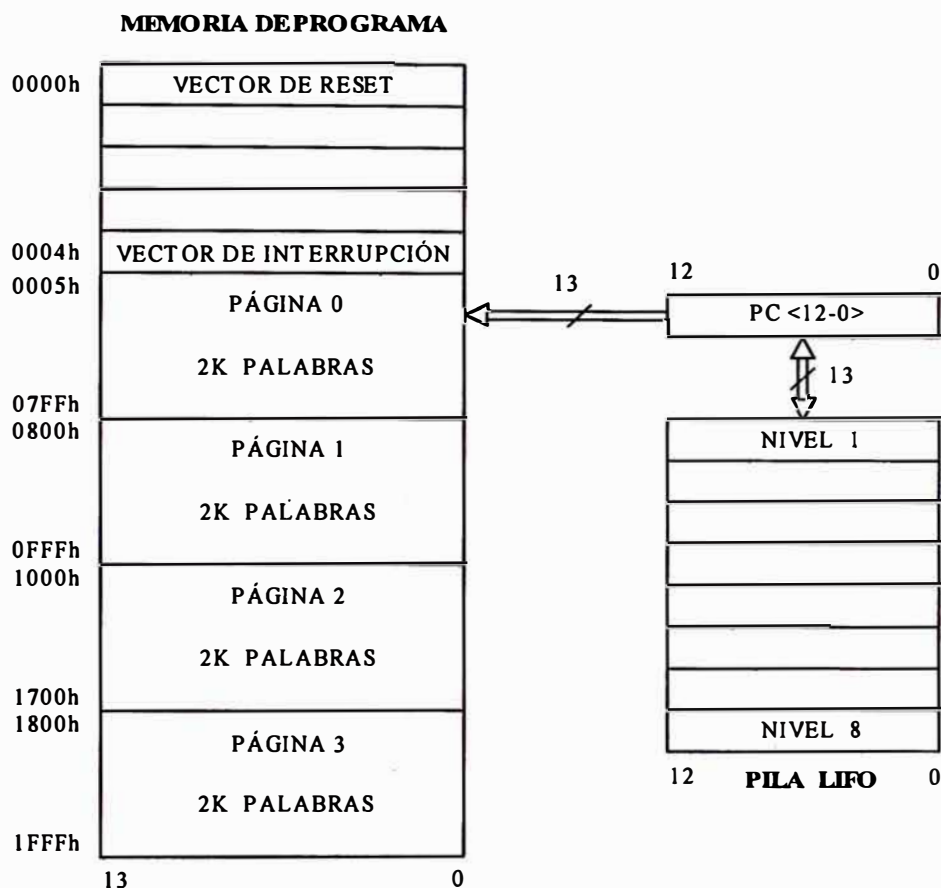


Figura 4.5 Organización de la memoria de programa tipo FLASH en el microcontrolador PIC16F876.

C. Contador de programa (PC)

Los 13 bits contenidos en el PC, que direccionan la memoria de programa, están guardados en dos registros de función especial. Cuando el PC se carga como resultado de la ALU, el registro PCL que ocupa repetido la posición 02h en los cuatro bancos, guarda los 8 bits de menos peso, puede ser leído y escrito. Los bits de mas peso PC<12-8>, residen en los 5 bits de menos peso del registro PCLATH, que

ocupa la posición 0Ah. En las instrucciones GOTO y CALL los 11 bits de menos peso del PC provienen del código de la instrucción y los otros dos de los bits PCLATH<4:3>; tal como se muestra en la figura 4.6.

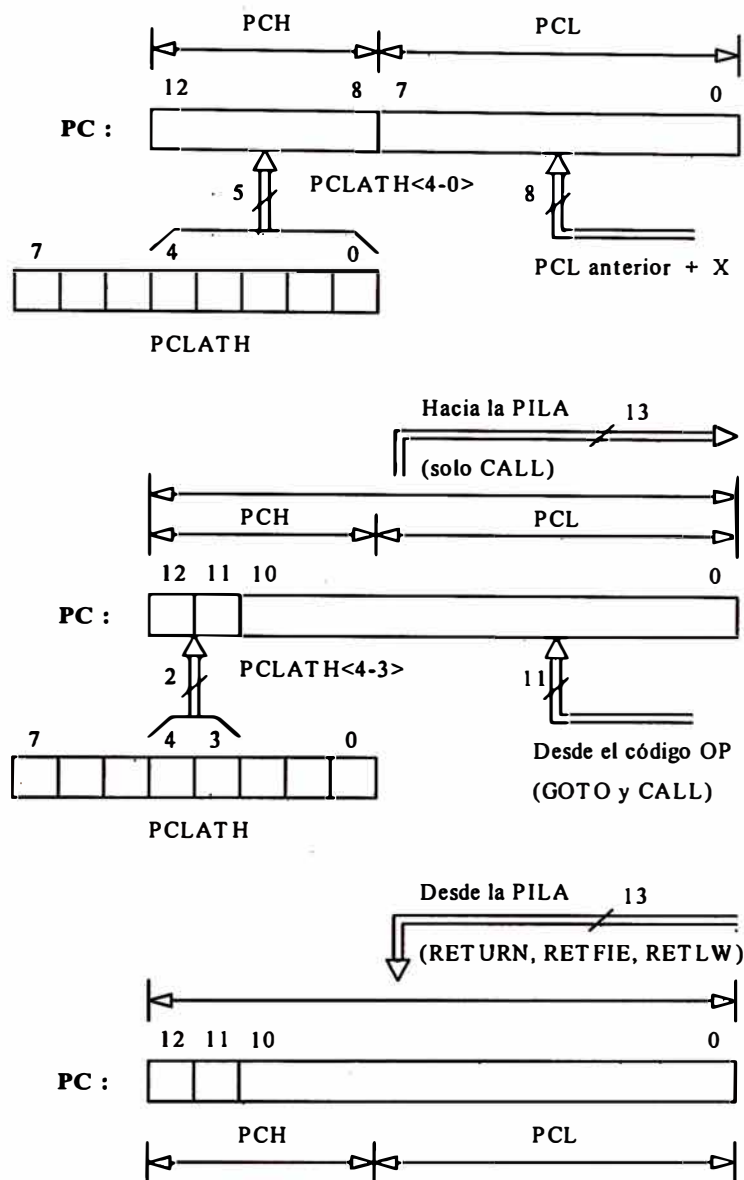


Figura 4.6 El PC como resultado de una operación, en las instrucciones GOTO y CALL, y en las instrucciones de retorno de subrutina.

La instrucción RETURN se limita a extraer de la cima de la Pila el valor que carga en el PC. La instrucción RETLW k además de hacer lo mismo que RETURN, carga en W el valor literal k, es decir devuelve un parámetro desde la subrutina. La

instrucción **RETFIE** carga en el PC el contenido de la cima de la Pila y pone el bit GIE a 1, pues al atender una interrupción este bit se pone automáticamente a 0 para evitar que cuando se atiende una interrupción se produzca otra. Figura 4.7.

Con los 11 bits que se cargan en el PC desde el código de las instrucciones de salto **GOTO** y **CALL**, se puede direccionar una página de 2K de memoria. Los bits restantes PC<12:11> tienen la misión de seleccionar uno de las 4 páginas de la memoria de programa, dichos bits proceden de PCLATH<4:3>.

D. Pila

La Pila es una zona aislada de las memorias de programa y la memoria de datos. Tiene una estructura LIFO, en la que el último valor guardado es el primero que sale. Tiene 8 niveles de profundidad cada uno de 13 bits. Funciona como un *buffer* circular, de manera que el valor que se obtiene al realizar el noveno desempilado (*pop*) es igual al que se obtuvo en el primero.

Las instrucción **CALL** y las interrupciones originan la carga del contenido del PC en el nivel superior o cima de la Pila. Dicho contenido se saca al ejecutar las instrucciones **RETURN**, **RETLW** y **RETFIE**. El contenido del registro PCLATH no es afectado por la entrada o salida de información de la Pila.

4.3.2. Memoria de datos

La Memoria de Datos tiene posiciones implementadas en RAM estática y otras en memoria EEPROM. En la zona SRAM, se alojan los registros operativos fundamentales en el funcionamiento del procesador y en el manejo de todos sus periféricos, además registros que el programador puede usar.

La RAM estática consta de cuatro bancos, que contienen Registros de Propósito General (GPR) y Registros de Función Especial (SFR), cada banco se extiende hasta 7Fh (128 bytes). Las más bajas posiciones de cada banco son reservadas para los Registros de la Función Especial. Para seleccionar el banco al que se desea acceder se emplea el bit 6 (RP1) y 5 (RP0) del registro STATUS.

Los **registros de propósito general** son usados por el programador dependiendo de la aplicación; para registros temporales, constantes, cuenta, de información, test, etc. Este modelo cuenta con 368 registros de 8 bits cada uno. Los **registros de función especial** son usados por la CPU y los módulos periféricos, para controlar el funcionamiento deseado del microcontrolador. Este modelo cuenta con 96 registros de 8 bits cada uno, de los cuales solo 75 registros se encuentran implementados.

La figura 4.8 presenta los cuatro bancos de la RAM estática, indicando en las primeras posiciones de cada uno los nombres de los registros que contienen. Se observa que algunos SFR se encuentran repetidos en la misma dirección de los bancos, lo cual reduce su código y acelera su acceso. Las posiciones con trama oscura no están implementadas físicamente y siempre se leen como cero.

Los SFR pueden ser clasificados en dos grupos; del núcleo (CPU) y periféricos. Los registros asociados con las funciones del núcleo se describen en detalle en esta sección. Aquellos relacionados a caracterizar el funcionamiento de los periféricos se describen en detalle dentro de cada periférico.

A. Registro de estado (STATUS)

Es el registro mas usado de todos, sus bits están destinados a controlar las funciones vitales del procesador. Por este motivo se encuentra en los 4 bancos.

Registro STATUS

IPR	RP1	RP0	#TO	#PD	Z	DC	C
bit 7							bit 0

IRP Selección de banco en modo indirecto. Ver figura 4.14.
 1: banco 2 y banco 3.
 0: banco 0 y banco 1.

RP1:RP0 Selección de banco en modo indirecto. Ver figura 4.14.

RP1	RP0	Banco seleccionado
0	0	Banco 0 (00h – 7Fh)
0	1	Banco 1 (80h – FFh)
1	0	Banco 2 (100h – 17Fh)
1	1	Banco 3 (180h – 1FFh)

#TO Time Out.
 1: vale 1, después de la conexión de alimentación o al ejecutarse las instrucciones CLRWDT o SLEEP.
 0: vale 0, cuando se produce el desbordamiento del WDT.

#PD Power down.
 1: vale 1, después de la conexión de alimentación o al ejecutar la instrucción CLRWDT.
 0: vale 0, al ejecutarse la instrucción SLEEP.

Z Señalizador de cero.
 1: si vale 1, el resultado de una operación lógico-aritmético es 0.
 0: si vale 0, el resultado de una operación lógico-aritmético no es 0.

DC Señalizador de llevada en el bit3. Igual que C, referido al bit3. De interés en operaciones en BCD.

C Señalizador de llevada en el bit de mas peso. Para las instrucciones de resta la correspondencia es inversa, es decir si vale 0 hay llevada.
 1: si vale 1, indica llevada en el bit de mas peso en la operación suma.
 0: si vale 0, no se ha producido llevada en la operación suma.

Figura 4.7 *Registro STATUS. Los bits 3 y 4 sólo se pueden leer, mientras los demás también se pueden escribir.*

B. Registro de opciones (OPTION_REG)

La misión principal de este registro es controlar TMR0 y WDT. Ocupa la dirección 81h y 181h de la memoria de datos en los bancos 2 y 4 respectivamente (ver figura 4.9). El bit T0CS selecciona la procedencia de los pulsos de reloj, que puede ser del oscilador interno ($F_{osc}/4$) o que se aplican desde el exterior por el pin RA4/T0CKI.

El divisor de frecuencia puede usarse con el TMR0 o con el WDT. Con el TMR0 actúa como predivisor, es decir los impulsos pasan primero por el divisor y

BANCO 0	Direc.	BANCO 1	Direc.	BANCO 2	Direc.	BANCO 3	Direc.
INDF	00h	INDF	80h	INDF	100h	INDF	180h
TMR0	01h	OPTION_REG	81h	TMR0	101h	OPTION_REG	181h
PCL	02h	PCL	82h	PCL	102h	PCL	182h
STATUS	03h	STARUS	83h	STARUS	103h	STARUS	183h
FSR	04h	FSR	84h	FSR	104h	FSR	184h
PORTA	05h	TRISA	85h		105h		185h
PORTB	06h	TRISB	86h	PORTB	106h	TRISB	186h
PORTC	07h	TRISC	87h		107h		187h
	08h		88h		108h		188h
	09h		89h		109h		189h
PCLATH	0Ah	PCLATH	8Ah	PCLATH	10Ah	PCLATH	18Ah
INTCON	0Bh	INTCON	8Bh	INTCON	10Bh	INTCON	18Bh
PIR1	0Ch	PIE1	8Ch	EEDATA	10Ch	EECON1	18Ch
PIR2	0Dh	PIE2	8Dh	EEADR	10Dh	EECON2	18Dh
TMR1L	0Eh	PCON	8Eh	EEDATH	10Eh	Reservado	18Eh
TMR1H	0Fh		8Fh	EEADRH	10Fh	Reservado	18Fh
T1CON	10h		90h		110h		190h
TMR2	11h	SSPCON2	91h		111h		191h
T2CON	12h	PR2	92h		112h		192h
SSPBUF	13h	SSPADD	93h		113h		193h
SSPCON	14h	SSPSTAT	94h		114h		194h
CCPR1L	15h		95h	Registros de	15h	Registros de	195h
CCPR1H	16h		96h	Propósito	116h	Propósito	196h
CCP1CON	17h		97h	General	117h	General	197h
RCSTA	18h	TXSTA	98h		118h		198h
TXREG	19h	SPBRG	99h	(16 Bytes)	119h	(16 Bytes)	199h
RCREG	1Ah		9Ah		11Ah		19Ah
CCPR2L	1Bh		9Bh		11Bh		19Bh
CCPR2H	1Ch		9Ch		11Ch		19Ch
CCP2CON	1Dh		9Dh		11Dh		19Dh
ADRESH	1Eh	ADRESL	9Eh		11Eh		19Eh
ADCON0	1Fh	ADCON1	9Fh		11Fh		19Fh
Registros de Propósito General (96 Bytes)	20h	Registros de Propósito General (80 Bytes)	A0h	Registros de Propósito General (80 Bytes)	120h	Registros de Propósito General (80 Bytes)	1A0h
			EFh				16Fh
	70h	Mapeados con 70h-7Fh	F0h	Mapeados con 70h-7Fh	170h	Mapeados con 70h-7Fh	1F0h
	7Fh		FFh		17Fh		1FFh

Figura 4.8 Memoria de datos con los 4 bancos de la memoria SRAM.

luego se aplican al TMR0, una vez aumentada su duración. Con el WDT actúa después, realizando la función de postdivisor. Los pulsos, que divide por un rango el divisor, pueden provenir de la señal de reloj interno ($F_{osc}/4$) o desde el exterior.

Registro OPTION_REG

#RBPU	INTEDG	TOCS	TOSE	PSA	PS2	PS1	PS0
bit 7							bit 0

- #RBPU** Resistencia *Pull up* Puerto B.
1: desactivadas.
0: activadas.
- INTEDG** Flanco activo de la interrupción externa.
1: flanco ascendente.
0: flanco descendente.
- TOCS** Tipo de reloj para el TMR0.
1: pulsos introducidos a través de TOCKI (contador).
0: pulsos de reloj interno $F_{osc}/4$ (temporizador).
- TOSE** Tipo de flanco en TOCKI.
1: incremento del TMR0 cada flanco descendente.
0: incremento del TMR0 cada flanco ascendente.
- PSA** Asignación del divisor de frecuencia.
1: el divisor de frecuencia se le asigna al WDT.
0: el divisor de frecuencia se le asigna al TMR0.
- PS2:PS0** Valor con el que actúa el divisor de frecuencia.

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

Figura 4.9 Asignación de funciones de los bits del registro OPTION_REG.

C. Registro de control de interrupciones (INTCON)

Se trata de un registro leíble y escribible que, para facilitar su acceso se halla en los 4 bancos, ocupando las direcciones 0Bh, 8Bh, 10Bh y 18Bh (ver figura 4.10). Tiene la misión de controlar las interrupciones provocadas por el TMR0, cambio de estado en

las 4 líneas de mas peso del Puerto B y activación del pin RB0/INT. El bit PEIE (permiso de interrupción de periféricos) actúa como una segunda llave parcial de permiso o prohibición de las causas de interrupción que no están contempladas en INTCON y que provocan los restantes periféricos. Con 1 se permiten y con 0 se prohíben las interrupciones.

Registro INTCON

GIE	PEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF
bit 7							bit 0

GIE	Permiso global de todas las interrupciones.
PEIE	Permiso de los periféricos que no se controlan con el registro INTCON.
TOIE	Permiso de interrupción del TMR0.
INTE	Permiso de interrupción externo por el pin RB0/INT.
RBIE	Permiso de interrupción por cambio en de estado en RB7-RB4.
TOIF	Señalizador de desbordamiento en TMR0. 1: vale 1, cuando ocurre el desbordamiento del TMR0. 0: cuando no se produce el desbordamiento.
INTF	Señalizador de activación del pin RB0/INT. 1: vale 1, al activarse RB0/INT. 0: cuando aun no se ha activado.
RBIF	Señalizador de cambio de estado en RB7-RB4. 1: vale 1, cuando cambia el estado de alguna de estas 4 líneas. 0: no ha cambiado el estado de ninguna de las 4 líneas.

Figura 4.10 Misión de cada uno de los bits del registro INTCON.

D. Registro de permiso de interrupciones 1 (PIE1)

La figura 4.11 muestra su asignación de bits, que permiten (con 1) o prohíben (con 0) las interrupciones provocadas por los periféricos internos del microcontrolador y que no estaban contempladas en INTCON. Ocupa la dirección 8Ch y para que cumplan su función los bits de PIE1 es necesario que PEIE = 1.

E. Registro de permiso de interrupciones 2 (PIE2)

La figura 4.12 muestra su asignación de bits. Los tres bits de permiso (con 1) o prohíben (con 0) las tres causas de interrupción que no configuran PIE1. Fin de

escritura en la EEPROM, colisión de bus en el modo SSP y en la captura o comparación del módulo CCP2. El bit 6 es un bit reservado y su valor siempre es 0.

Cuando se leen los bits que no tienen asignada función se obtiene 0.

Registro PIE1

PSPIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE
bit 7							bit 0

- PSPIE** Permiso de interrupción para el Puerto Paralelo Esclavo al realizar una operación de lectura/escritura. Solo para los de 40 pines, manteniéndose en 0 en los de 28 pines.
- ADIE** Permiso de interrupción par el convertidor A/D al finalizar la conversión.
- RCIE** Permiso de interrupción para el receptor del USART cuando el buffer se llena.
- TXIE** Permiso de interrupción para el transmisor del USART cuando el buffer se vacía.
- SSPIE** Permiso de interrupción para el puerto serie síncrono.
- CCP1IE** Permiso de interrupción para el módulo CCP1 cuando se produce una captura o comparación.
- TMR2IE** Permiso de interrupción para el TMR2 con su desbordamiento.
- TMR1IE** Permiso de interrupción para el TMR1 con su desbordamiento.

Figura 4.11 Designación y funciones de los bits del registro PIE1.

Registro PIE2

---	0	---	EEIE	BCLIE	---	---	CCP2IE
bit 7							bit 0

- EEIE** Permiso de interrupción por fin de escritura en la EEPROM.
- BCLIE** Permiso de interrupción par colisión de bus en el SSP cuando dos o mas maestros tratan de transferir al mismo tiempo.
- CCP2IE** Permiso de interrupción por el módulo CCP2.

Figura 4.12 Designación y funciones de los bits del registro PIE2.

F. Registros de los señalizadores de interrupciones 1 y 2 (PIR1 y PIR2)

En correspondencia con los bits de permiso/prohibición de las causas de interrupción recogidas en el registro PIE1 y PIE2, existen otros dos registros PIR1 y PIR2 (ver figura 4.13), cuyos bits actúan de señalizadores de la causa que provoca la interrupción, independientemente si está permitida o prohibida. Ocupan las direcciones 0Ch y 0Dh respectivamente.

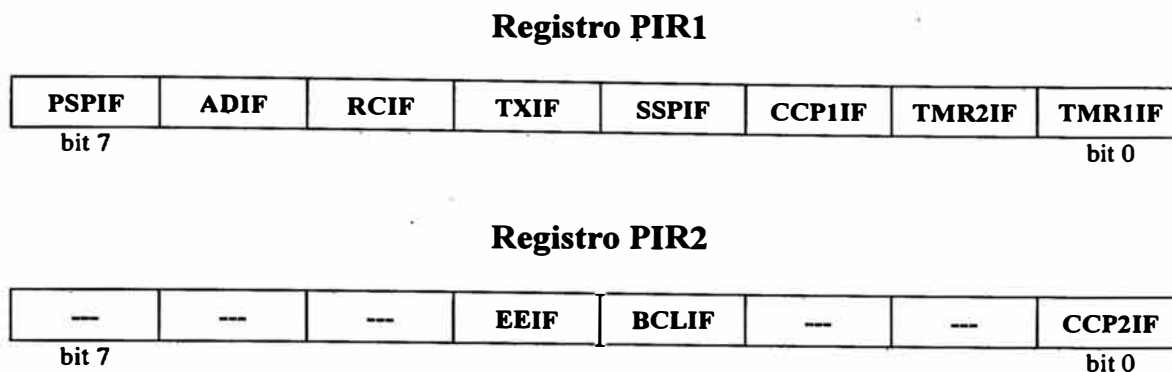


Figura 4.13 *Los bits de los registros PIR1 y PIR2 contienen los señalizadores de las causas que producen las interrupciones.*

G. Direccionamiento de la memoria de datos

Para direccionar la memoria de datos, existen dos modos diferentes; tal como se muestra en la figura 4.14: En el modo de direccionamiento directo, los bits RP0 y RP1 del registro STATUS se encargan de seleccionar el banco, mientras que la dirección de la memoria dentro del banco lo determinan los 7 bits que viene incluido en el código OP de la instrucción. El modo de direccionamiento indirecto se usa cuando en una instrucción se utiliza como operando el registro INDF, que no esta implementado físicamente y ocupa la dirección 00h en los 4 bancos. Los 7 bits de menos peso del registro FSR señalan la dirección y el banco lo determinan el bit IRP del registro STATUS concatenado con el bit de mas peso del registro FSR.

4.4. Memoria de programa FLASH y de datos EEPROM

En el PIC16F876 se puede leer o escribir la memoria de programa tipo FLASH. Esto significa que un programa dinámicamente puede generar información que se puede grabar en la memoria FLASH directamente, sin necesidad del grabador externo.

La memoria EEPROM no está mapeado en la zona de la memoria de datos para poder leerlo y escribirlo durante el funcionamiento normal del microcontrolador hay

que utilizar los SFR. En el registro EEADR, ubicado en la dirección 10Dh del banco 3, se carga la dirección a acceder de la EEPROM. En el registro EEDATA, ubicado en la dirección 10Ch del banco 3, se depositan los datos que se leen o se escriben.

Para señalar los 13 bits que direccionan la memoria FLASH, se utiliza EEADR concatenado con EEADRH, que contiene los 5 bits de mas peso de la dirección. El registro EEDATAH se concatena con el registro EEDATA, que contiene los 6 bits de mas peso de los 14 bits de una instrucción leída o a escribir en la memoria FLASH.

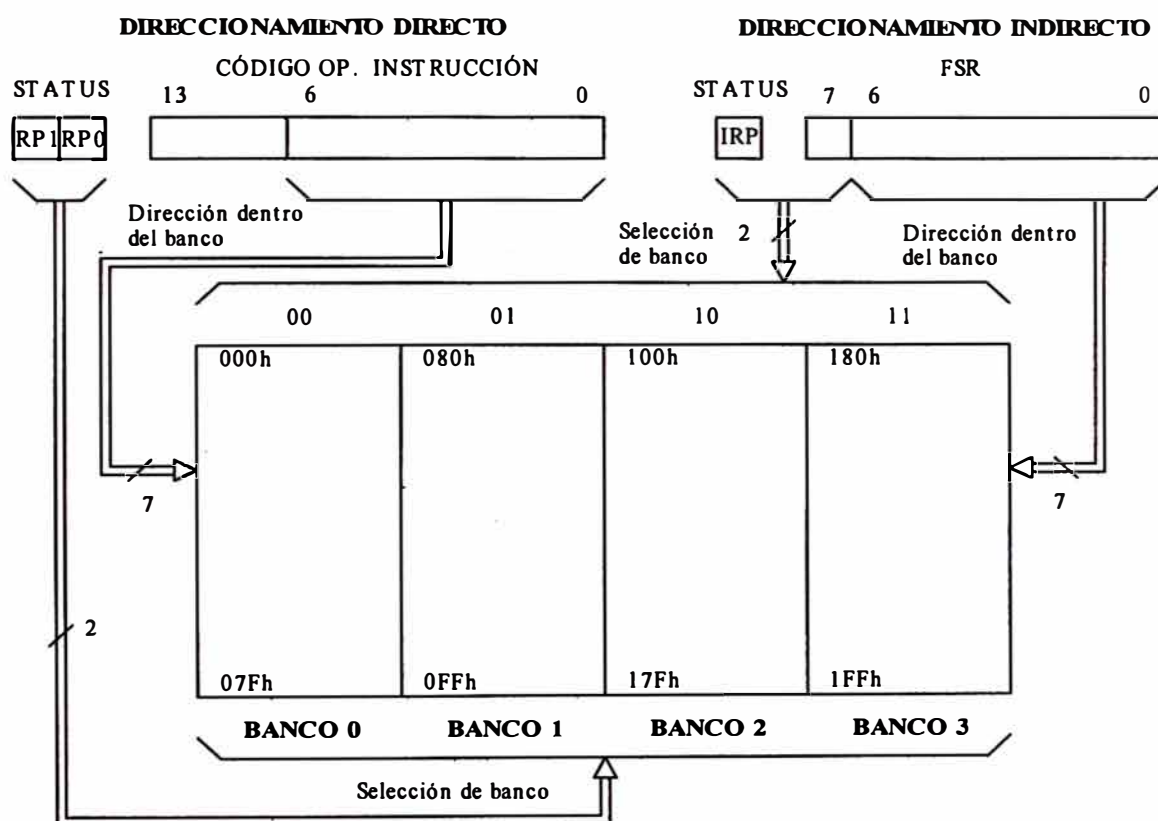


Figura 4.14 Formas de seleccionar el banco y la dirección dentro del banco, en los direccionamientos directo e indirecto.

Para controlar la operación de lectura/escritura de las memorias EEPROM y FLASH se utilizan los registros EECON1 y EECON2. El registro EECON1, ubicado en la dirección 18Ch del banco 4, tiene la misión de controlar las operaciones y el

registro EECON2, ubicado en la dirección 18Dh del banco 4, solo asume funciones de seguridad en el proceso de escritura, que tiene una duración de 2ms aproximada. Antes de iniciar la escritura de una palabra se escribe en EECON2 primero el dato 55h y luego el AAh. Ver figura 4.15 y tabla 4.1.

Registro EECON1

EEPGD	---	---	---	WRERR	WREN	WR	RD
bit 7							bit 0

- EEPGD** Selecciona acceso a la memoria.
1: memoria FLASH.
0: memoria EEPROM.
- WRERR** Señalizador de error en escritura.
1: vale 1, cuando la operación de escritura a terminado prematuramente.
0: vale 0, cuando la operación de escritura se ha completado correctamente.
- WREN** Permiso de escritura.
1: permite escritura.
0: prohíbe escritura.
- WR** Control de escritura.
1: inicializa ciclo de escritura, luego pasa a 0 automáticamente.
- RD** Control de lectura.
1: inicializa ciclo de lectura, luego pasa a 0 automáticamente.

Figura 4.15 Denominación y misión de los bits del registro EECON1.

Configuración de bits			Posiciones de memoria FLASH	Lectura interna	Escritura interna	Lectura ICSP	Escritura ICSP
CP1	CP0	WRT					
0	0	X	Toda la memoria	Si	No	No	No
0	1	0	Áreas no protegidas	Si	No	Si	No
0	1	0	Áreas protegidas	Si	No	No	No
0	1	1	Áreas no protegidas	Si	Si	Si	No
0	1	1	Áreas protegidas	Si	No	No	No
1	0	0	Áreas no protegidas	Si	No	Si	No
1	0	0	Áreas protegidas	Si	No	No	No
1	0	1	Áreas no protegidas	Si	Si	Si	No
1	0	1	Áreas protegidas	Si	No	No	No
1	1	0	Toda la memoria	Si	No	Si	Si
1	1	1	Toda la memoria	Si	Si	Si	Si

Tabla 4.1 Diversas posibilidades de protección de la memoria FLASH ante operaciones de lectura y escritura, de acuerdo con los valores de los bits WRT, CP1 y CP0 de la palabra de configuración.

4.5. Puertos de entrada/salida

El microcontrolador PIC16F876 dispone de tres Puertos de E/S: Puerto A, Puerto B y Puerto C, los cuales se pueden leer o escribir. Todas las líneas de estos Puertos son multifuncionales, es decir realizan diversas funciones según estén programados. Sin embargo, todos tienen la capacidad de trabajar como líneas de E/S digitales.

4.5.1. Puerto A

Dispone de 6 líneas denominadas RA5-RA0 que admiten entrada TTL y salida CMOS. Los bits del registro TRISA que ocupa la dirección 05h del banco 0 configuran las líneas del Puerto A (PORTA); como entrada si están a 1 y como salida si están a 0. Si funciona como salida el contenido de la báscula de salida se aplica al pin correspondiente, según puede apreciarse en la figura 4.16.

Al leer el registro PORTA que ocupa la dirección 05h del banco 1, se lee la información que ingresa o sale por los pines del Puerto A, que es el que se halla escrito en la báscula de datos de la figura 4.16. La escritura entraña una operación de lectura-modificación-escritura, es decir, lee los pines, luego modifica su valor y finalmente se escribe en la báscula de datos.

Los pines RA0/AN0, RA1/AN1 y RA2/AN2, además de líneas de E/S digitales también pueden actuar como canales 0, 1 y 2 para el convertidor A/D. El pin RA3/AN3/Vref + también puede actuar como entrada de la tensión de referencia para los periféricos que la precisan. El pin RA4/TOCKI actúa como E/S digital y como entrada de la señal de reloj externo para el TMR0. Por último, el pin RA5/AN4/#SS tiene multiplexados tres funciones: E/S digital, canal 4 para el convertidor A/D y selección del modo esclavo cuando trabaja con la comunicación serie síncrona.

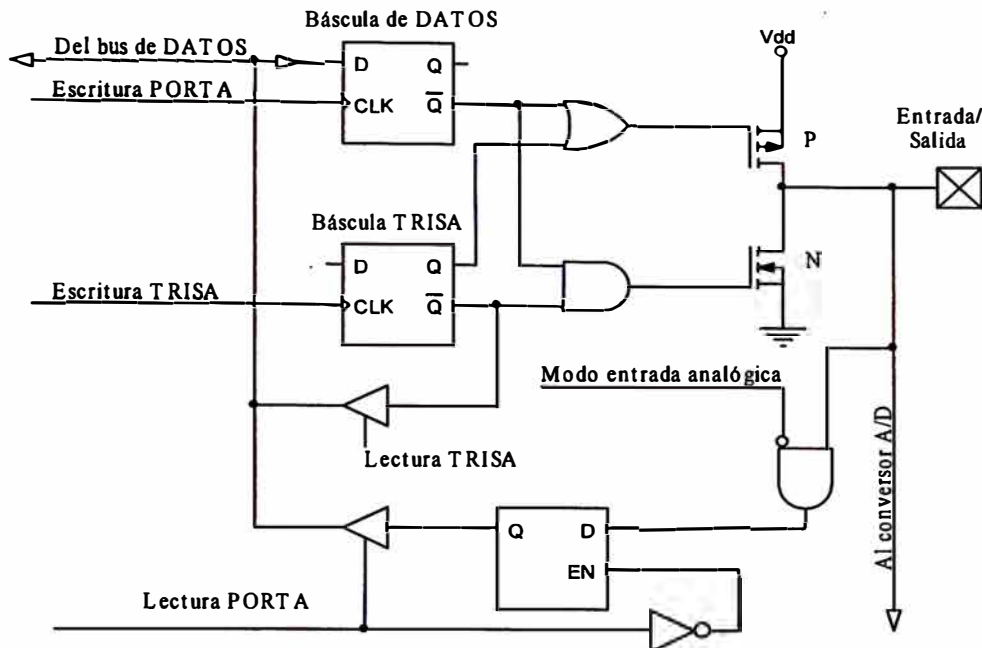


Figura 4.16 Esquema de conexión de los pines RA5 y RA3-RA0, que multiplexan la función de E/S digital y entrada de señal analógica al ADC.

Para seleccionar si las líneas del Puerto A va a trabajar como E/S digitales o como canales de entrada para el convertidor A/D, hay que configurar adecuadamente el registro ADCON1, que se detalla al describir el módulo convertidor A/D.

4.5.2. Puerto B

Cuenta con 8 líneas bidireccionales denominadas RB7-RB0, cuyo sentido de E/S se configura mediante el registro TRISB que ocupa la dirección 06h del banco 0 y la información que ingresan o salen se encuentran en el registro PORTB que ocupa la dirección 06h del banco 1.

A todas las líneas de este Puerto se les permite conectar una resistencia interna *pull-up* de elevado valor con el Vdd. Para este fin hay que programar el bit #RBPU a 0 que se encuentra en el registro OPTON_REG. La resistencia de *pull-up*, que es un transistor CMOS tipo P como se aprecia en la figura 4.17, se conecta

automáticamente siempre que la línea este configurada como salida. Cuando se produce un Reset por la conexión de alimentación se desconectan todas las resistencias *pull-up*.

Las líneas RB7-RB4 pueden programarse, para generar una interrupción cuando una de ellas cambia de estado. Se deben configurar como entradas y el valor que se introduce por ellas se compara con el anterior y si no coinciden se genera una interrupción, siempre que lo autorice el bit de permiso. En cualquier caso el señalizador RBIF del registro INTCON, se pondrá a 1.

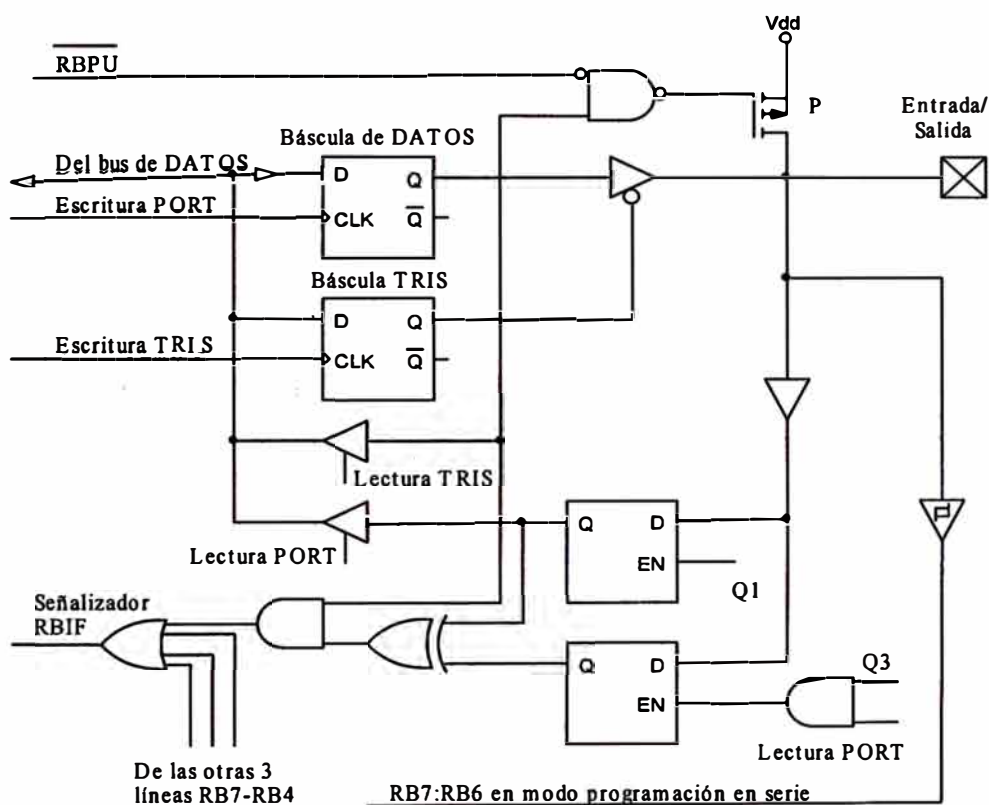


Figura 4.17 Conexión de los pines RB7-RB0 del Puerto B.

4.5.3. Puerto C

Cuenta de 8 líneas bidireccionales denominadas RC7-RC0, cuyo sentido de E/S se configura mediante el registro TRISC que ocupa la dirección 07h del banco 0 y la

información que ingresan o salen por sus pines se encuentran en el registro PORTC que ocupa la dirección 07h del banco 1. Todos los pines de este Puerto tienen multiplexados diferentes funciones.

4.6. Temporizador/contador (TMR0, TMR1, TMR2)

Una de las labores más habituales en los programas de control de dispositivos suele ser determinar intervalos concretos de tiempo, el encargado de realizar esta función se llama temporizador. También son frecuentes contar los pulsos que se producen en el exterior y el elemento destinado a este fin se denomina contador.

Si las labores del temporizador o contador las asignamos al programa principal quitarían mucho tiempo al procesador en detrimento de actividades más importantes. Por este motivo se diseñan recursos específicamente orientados a estas misiones. El PIC16F876 dispone de un potente conjunto de temporizadores para manejar eficientemente todas las operaciones que involucran el tiempo y contaje. Dichos temporizadores son tres y se denominan: TMR0, TMR1 y TMR2.

4.6.1. Temporizador/Contador TMR0

El registro TMR0 está ubicado en la dirección 01h y 101h de los bancos 0 y 3 de la memoria de datos respectivamente, se incrementa con cada flanco. Tiene las siguientes características:

- Temporizador/contador ascendente de 8 bits.
- Leíble y escribible.
- Reloj interno ($F_{osc}/4$) o externo pin TOCKI.
- Selección del flanco en el reloj externo.

- Predivisor de la frecuencia de reloj programable.
- Generación de interrupción opcional en el desbordamiento.

En realidad, estos microcontroladores disponen de dos temporizadores, el TMR0 y el Temporizador perro guardián (WDT). El primero se encarga del control de tiempos y contaje de pulsos. El otro vigila que el programa no se cuelgue, para esto cada cierto tiempo comprueba si el programa se está ejecutando normalmente. En caso contrario, si el control esta detenido en un bucle infinito a la espera de algún acontecimiento que no se produce, el WDT avisa, lo que se traduce en un Reset que reinicializa todo el sistema.

A menudo el TMR0 y WDT precisan controlar largos intervalos de tiempo y necesitan aumentar la duración de los pulsos de reloj que les incrementa. Para cubrir este requisito se dispone de un circuito programable denominado divisor de frecuencia, que divide la frecuencia utilizada por diversos rangos. Para programar el comportamiento del TMR0, WDT y el divisor de frecuencia se utilizan algunos bits del registro OPTION_REG, y otros bits de la palabra de configuración.

Cuando funciona como temporizador conviene cargarle con el valor de los pulsos que se quiere temporizar, pero expresados en complemento a 2, incrementándose en cada pulso $F_{osc}/4$. De esta manera, al llegar el número de pulsos deseado, se desborda al pasar de FFh a 00h y se activa el señalizador TOIF y/o se produce una interrupción. Para calcular los tiempos a controlar con TMR0 se utilizan las siguientes fórmulas:

$$\text{Temporización} = 4 * T_{osc} * (\text{Valor en TMR0}) * (\text{Divisor}) \quad \dots (4.3)$$

$$\text{Valor en TMR0} = \text{Temporización} / (4 * T_{osc} * (\text{Divisor})) \quad \dots (4.4)$$

Cuando funciona como contador de sucesos, se incrementa la cuenta en cada pulso que se aplica al pin RA4/T0CKI. Al llegar al valor FFh se desborda el contador y con el siguiente pulso pasa a 00h, activando un señalizador y/o provocando una interrupción.

En la figura 4.18 se ofrece el esquema de funcionamiento del TMR0. Se observa que hay un bloque que retrasa 2 ciclos el conteo para sincronizar el momento del incremento producido por la señal aplicado al pin RA4/T0CKI con el que producen los pulsos internos del reloj. Cuando se escribe el TMR0 se retrasan 2 ciclos su reincremento y se pone a 0 el divisor de frecuencia.

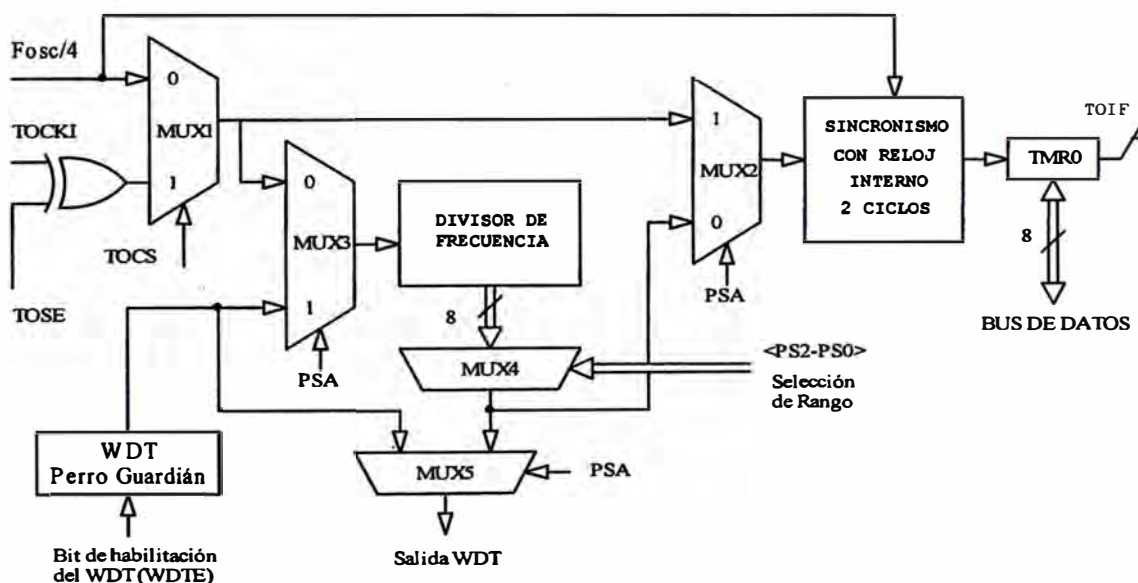


Figura 4.18 Esquema general del funcionamiento del TMR0. Cuando se desborda se pone a 1 el señalizador TOIF.

4.6.2. Temporizador/Contador TMR1

El registro TMR1 es el único Temporizador/contador ascendente con un tamaño de 16 bits, lo que requiere el uso de dos registros concatenados de 8 bits. TMR1H y TMR1L, ubicados en la dirección 0Eh y 0Fh del banco 0, son los encargados de

guardar el valor del contaje en cada momento. Dicho valor se incrementa desde 0000h hasta FFFFh, instante en el que se activa el señalizador TMR1IF y regresa al valor inicial 0000h. También, si desea se puede provocar una petición de interrupción. El valor contenido en TMR1H:TMR1L puede ser leído o escrito y los pulsos de reloj que originan el contaje ascendente pueden provenir del exterior o de la frecuencia de funcionamiento del microcontrolador ($F_{osc}/4$).

El TMR1 es capaz de funcionar de tres formas: temporizador, contador síncrono y contador asíncrono. En el modo temporizador el valor concatenado TMR1H:TMR1L se incrementa con cada ciclo de instrucción ($F_{osc}/4$). En el modo contador, el incremento se puede producir con los flancos ascendentes de un reloj externo, cuya entrada se aplica a las líneas RC1 y RC0 del Puerto C, o por pulsos aplicados en la línea RC0. En la figura 4.19 se muestra de bloques del TMR1.

El registro T1CON

El funcionamiento del TMR1 esta gobernado con los bits programados del registro T1CON (ver figura 4.20), que ocupa la dirección 10h de la memoria de datos. El bit TMR1ON gobierna el permiso o la prohibición de funcionamiento del TMR1. El bit TMR1CS selecciona la fuente de los pulsos de contaje interno o externo. Cuando los pulsos proceden de un reloj externo, es preciso que el bit T1OSCEN tenga valor 1, en cuyo caso los pines RC0/T1OSO/T1CKI y RC1/T1OSI/CCP2 actúan como entradas del oscilador externo. Si T1OSCEN vale 0 los pulsos vendrán a través de RC0/T1OSO/T1CKI. En ambos casos, el TMR1 funciona como contador de eventos externos, los bits 1 y 0 de TRISC carecen de significado, ya que uno o ambos pines no pueden actuar al mismo tiempo como entrada de pulsos y como líneas de E/S.

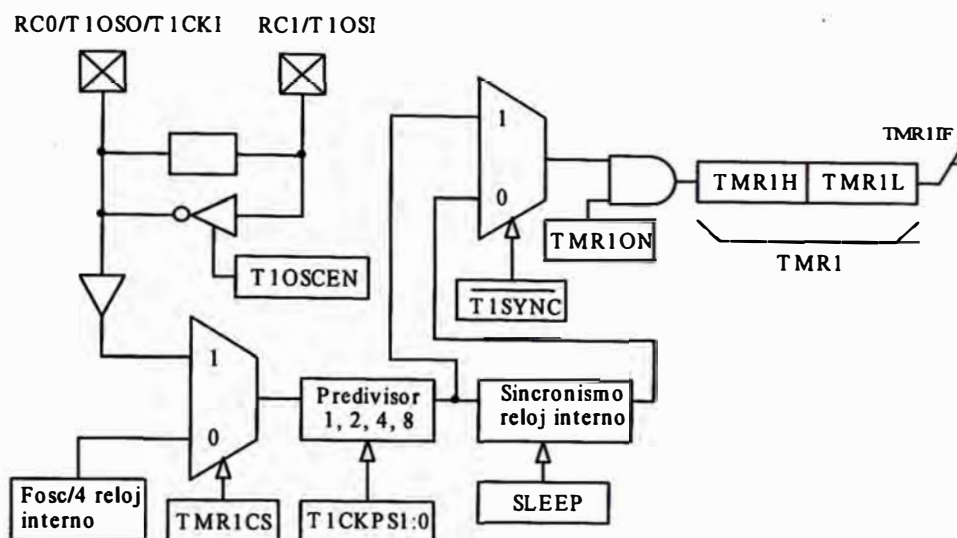


Figura 4.19 Diagrama de bloques del TMR1 con sus señales de control.

Entre los pines RC0 y RC1 se puede colocar un cristal de hasta una frecuencia de 200KHz. En estas condiciones, el oscilador puede seguir funcionando aunque se fuerce el estado de bajo consumo SLEEP.

El predivisor de frecuencia es un simple divisor de la frecuencia de los pulsos que se aplican al TMR1 por 1, 2, 4 u 8. El rango de división lo eligen los bits <T1CKPS1:T1CKPS0>.

El bit #T1SYNC determina la posible sincronización o no de los pulsos del reloj externo con los del reloj interno, según valga 0 o 1 respectivamente.

Registro T1CON

----	----	T1CKPS1	T1CKPS0	T1OSCEN	#T1SYNC	TMR1CS	TMR1ON
bit 7							bit 0

Figura 4.20 Distribución de los bits del registro de control del TMR1.

4.6.3. Temporizador/Contador TMR2

Se trata de un temporizador ascendente de 8 bits, que se puede leer y escribir; también puede realizar funciones especiales para el puerto serie síncrono, y con los

módulos de captura y comparación. En el modo reposo, se detiene el oscilador interno dejando de funcionar el TMR2. Ocupa la dirección 11h de los SFR.

La señal de reloj del TMR2 es interna $F_{osc}/4$, y antes de ser aplicada pasa por un predivisor con rangos de 1:1, 1:4, 1:16. La salida del TMR2 atraviesa un postdivisor de frecuencia con rangos de división desde 1:1 a 1:16, pasando por los 16 valores posibles.

El registro T2CON

Para controlar el funcionamiento del TMR2 se usa el registro T2CON, el cual ocupa la dirección 12h de la memoria de datos, cuya distribución y asignación de bits se muestra en la figura 4.21.

Registro T2CON

---	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2SCKPS1	T2SCKPS0
bit 7							bit 0

Figura 4.21 Distribución de los bits del registro de control del TMR2.

Los bits $\langle T2CKPS1:T2CKPS0 \rangle$ sirven para seleccionar el rango de división del predivisor de pulsos de la siguiente forma:

$\langle T2CKPS1:T2CKPS0 \rangle$	RANGO DEL PREDIVISOR
00	1:1
01	1:4
1x	1:16

El bit TMR2ON sirve para permitir (1) o prohibir (0) el funcionamiento del TMR2. El bit de mas peso no es significativo y los cuatro bits restantes $\langle TOUTPS3-TOUTPS0 \rangle$ determinan el rango por el que divide la frecuencia el postdivisor de acuerdo a lo siguiente:

<TOUTPS3-TOUTPS0>	RANGO DEL POSTDIVISOR
0000	1:1
0001	1:2
0010	1:3
...	...
1111	1:16

Al entrar el microcontrolador en modo SLEEP, se detiene el oscilador interno y al no existir la señal $F_{osc}/4$ deja de funcionar el TMR2.

El señalizador de desbordamiento del TMR2 es el TMR2IF del registro PIR1. El predivisor y el postdivisor se ponen a 0 al escribir el TMR2, al escribir el T2CON o con un Reset. Sin embargo, cuando se escribe el T2CON no se borra el TMR2.

El TMR2 tiene asociado un registro de período PR2, que ocupa la dirección 92h. Cuando el valor de conteo del TMR2 coincide con el valor cargado en PR2 se genera un pulso en la salida EQ (figura 4.22) y se borra el TMR2. El pulso EQ puede ser dividido por el postdivisor antes de activar el señalizador TMR2IF.

Este temporizador también tiene la capacidad de producir una petición de interrupción habilitándolo con el bit TMR2IE del registro PIE1. Además, con la programación adecuada, los pulsos de salida del TMR2 antes del postdivisor pueden actuar como base de tiempos para el módulo SSP.

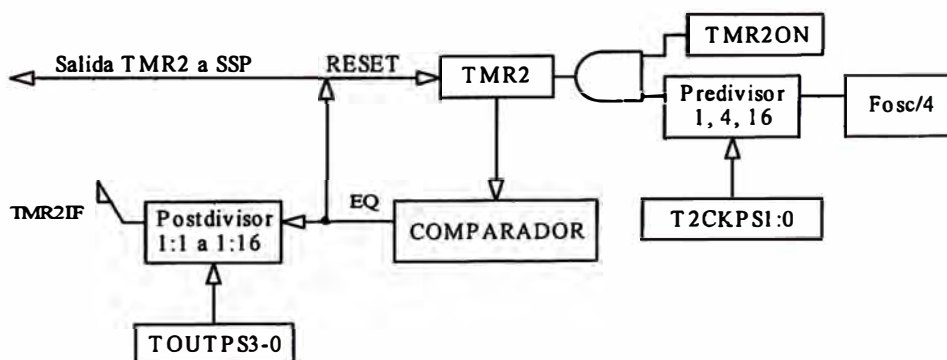


Figura 4.22 Diagrama de bloques del TMR2 con sus señales de control.

4.7. Módulo CCP

Los microcontroladores PIC16F87X disponen de dos módulos CCP, llamados CCP1 y CCP2 (ver figura 4.23 y tabla 4.2), que son idénticos excepto en lo referente a la modalidad de disparo especial. Dada esta similitud, la descripción se orienta hacia el módulo CCP1. Estos módulos pueden realizar tres funciones principales:

1. Modo captura: Una pareja de registros de un módulo CCPX captura el valor que tiene el TMR1 cuando ocurre un evento especial en el pin RC2/CCP1 (para el módulo CCP1) o en el pin RC1/T1OSI/CCP2 (para el módulo CCP2).
2. Modo comparación: Se compara el valor de 16 bits del TMR1 con otro valor cargado en una pareja de registros de un módulo CCPX y cuando coinciden se produce un evento en los pines RC2/CCP1 y/o RC1/T1OSI/CCP2.
3. Modo modulación de anchura de pulsos (PWM): Dentro del intervalo del periodo de un pulso controla la anchura en que la señal vale nivel alto.

El módulo CCP1 utiliza un registro de trabajo de 16 bits que esta formado por la concatenación de los registros CCPR1H:CCPR1L (direcciones 16h y 15h). El registro de control CCP1CON del módulo CCP1 ocupa la dirección 17h. El módulo CCP2 tiene como registro de trabajo a CCPR2H:CCPR2L (direcciones 1Ch y 1Bh) y como registro de control a CCP2CON que ocupa la dirección 1Dh. Las parejas de registros son las encargadas de capturar el valor del TMR1, de comparar el valor que tienen con el valor de TMR1 o en el modo PWM, de modular la anchura del pulso.

Registro CCPxCON

—	—	CCPxX	CCPxY	CCPxM3	CCPxM2	CCPxM1	CCPxM0
bit 7							Bit 0

Figura 4.23 *Asignación de los bits de los registros CCPxCON para los módulos CCP1 y CCP2. Todos sus bits son leíbles y escribibles.*

<CCPXM3-CCPXM0>	MODO DE TRABAJO DEL MÓDULO
0000	Modulo CCPX desconectado
0100	Modo captura con cada flanco descendente en RCY/CCPX
0101	Modo captura con cada flanco ascendente en RCY/CCPX
0110	Modo captura cada 4 flancos ascendentes en RCY/CCPX
0111	Modo captura cada 16 flancos ascendentes en RCY/CCPX
1000	Modo comparación que activa el pin RCY/CCPX al coincidir valores
1001	Modo comparación que desactiva el pin RCY/CCPX al coincidir valores
1010	Modo comparación que genera una interrupción por software no afecta RCY/CCPX
1011	Modo comparación en el que se produce un disparo especial diferente para cada módulo
11xx	Modo PWM

Tabla 4.2 *Se configuran los 4 bits <CCPXM3-CCPXM0> del registro CCPXCON para seleccionar el modo de trabajo de los módulos CCP1 o CCP2.*

4.7.1. Modo Captura

En este modo, la pareja de registros CCPXH:CCPXL del módulo CCPX captura el valor de 16 bits que contiene el TMR1 cuando sucede un evento en los pines RCY/CCPX del Puerto C, que previamente ha sido configurado como entrada poniendo a 1 el bit correspondiente del registro TRISC.

Los eventos posibles que pueden ocurrir sobre los pines RCY/CCPX para producir la captura del TMR1 sobre la pareja de registros CCPXH:CCPXL son:

1. Un flanco ascendente.
2. Un flanco descendente.
3. Cada 4 flancos ascendentes.
4. Cada 16 flancos ascendentes.

Los 4 bits <CCP1M3-CCP1M0> del registro CCP1CON seleccionan el evento adecuado en el módulo CCP1; de modo similar los 4 bits <CCP2M3-CCP2M0> del registro CCP2CON seleccionan el evento en el módulo CCP2.

Al efectuar la captura, se activa el señalizador CCP1IF del registro PIR1. Además, si se pone a 1 el bit de permiso de interrupción PIE1<CCP1IE>, se genera una petición de interrupción cuando se carga en CCP1H:CCP1L el valor de TMR1. En la figura 4.24 se muestra los elementos mas significativos que intervienen en el trabajo del modo de captura.

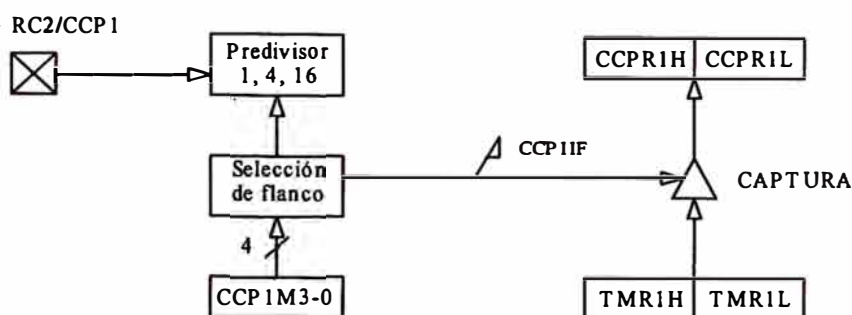


Figura 4.24 Diagrama de bloques del módulo CCP1 en modo captura.

Cuando se emplea el módulo CCP1, el TMR1 debe estar configurado para trabajar como temporizador o como contador síncrono. Nunca en modo asíncrono. Si se van a cambiar las condiciones de funcionamiento en el modo captura, conviene detener o desactivar al modulo CCP para evitar que se produzcan falsas interrupciones durante la operación.

4.7.2. Modo Comparación

El TMR1 debe trabajar en modo temporizador o contador síncrono, nunca en modo asíncrono. En esta forma de trabajo, la pareja de registros CCP1H:CCP1L compara su contenido, de forma continua, con el valor del TMR1. Cuando coinciden ambos valores, en el pin RC2/CCP1, que se halla configurada como salida le acontece uno de los siguientes eventos, de acuerdo con la programación de los bits <CCP1M3-CCP1M0>:

1. Pasa a nivel alto.
2. Pasa a nivel bajo.
3. No cambia su estado pero se produce una interrupción.

Al coincidir los valores del TMR1 con los de la pareja CCPR1H:CCPR1L se pone a 1 el señalizador CCP1IF. Si el bit de permiso de interrupción esta en 1, cuando coinciden los valores mencionados se origina una petición de interrupción. En la figura 4.25 se observa un esquema de bloques de la estructura general del módulo CCP1 en modo comparación.

Si con los bits <CCP1M3-CCP1M0> se selecciona el modo de trabajo de disparo especial, el módulo CCP1 pone a 0 el TMR1 y el CCPR1 funciona como un registro de periodo, capaz de provocar periódicamente interrupciones.

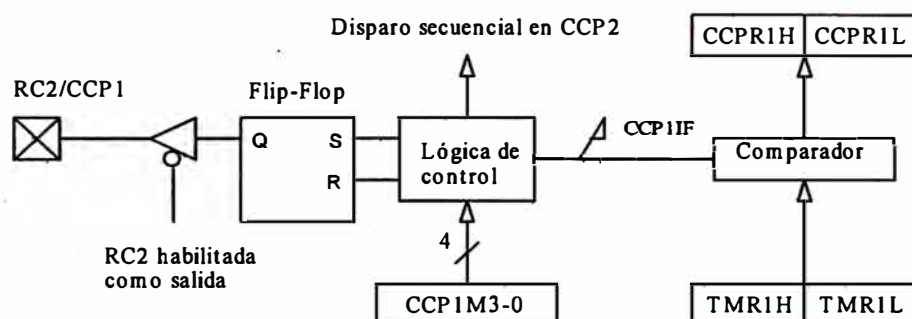


Figura 4.25 Diagrama de bloques del módulo CCP1, en modo Comparación.

4.7.3. Modo de modulación de anchura de pulsos

En este modo de trabajo se consiguen pulsos lógicos cuya anchura del nivel alto es de duración variable, que son de enorme aplicación en el control de dispositivos tan populares como los motores y los triacs.

El pin RC2/CCP1 está configurado como salida y bascula entre los niveles lógicos 0 y 1 a intervalos variables de tiempo. Lo que se intenta es obtener un pulso

cuyo nivel alto tenga una anchura variable (*Duty Cycle*) dentro del intervalo del periodo de trabajo. Para lograr el basculado en el pin de salida RC1/CCP1 se usa un comparador que pone a 1 un flip flop, cuando el valor del registro PR2 coincide con la parte alta del TMR2. De esta manera, variando los valores en PR2 y en CCPR1L (que luego se pasa a CCPR1H) se varía el intervalo de tiempo para el cual el pin esta a 1 y a 0.

Cuando se trabaja con una precisión de 10 bits, los 2 bits CCP1CON<5:4> se conectan con los 8 de CCPR1L y de la misma forma, los 8 bits de mas peso del TMR2 se concatena con los dos bits de menos peso del reloj interno, haciendo que el TMR2 cuente cada T_{OSC} en vez de cada $4T_{OSC}$.

El tiempo que dura el periodo de la onda depende del valor cargado en PR2, según la formula siguiente:

$$\text{Periodo} = (\text{PR2} + 1) * T_{OSC} * (\text{Pre-divisor TMR2}) \quad \dots (4.5)$$

Cuando el valor de TMR2 coincide con el del PR2 suceden tres acontecimientos:

1. Se borra el TMR2.
2. El pin RC/CCP1 se pone a 1.
3. El valor de CCPR1L, que determina el ancho del pulso, se carga en CCPR1H.

El tiempo que la salida se encuentra a nivel alto, que es la anchura del pulso, depende del contenido cargado en CCPR1L y de los dos bits 5 y 4 del CCP1CON, cuando se trabaja con una presión de 10 bits. El ancho del pulso (A_p) esta dada por:

$$A_p = (\text{CCPR1L} : \text{CCP1CON} \langle 5:4 \rangle) * T_{OSC} * (\text{Pre-divisor TMR2}) \quad \dots (4.6)$$

El valor CCPR1L:CCP1CON<5:4> puede cargarse en cualquier momento, puesto que el mismo no se traspasa a CCPR1H y se compara hasta que coincidan PR2 con TMR2. En el modo PWM el registro CCPR1L solo puede ser leído.

4.8. Módulo MSSP

En los microcontroladores PIC16F87X, Microchip ha implementado en el silicio de sus chips el módulo MSSP (*Master Synchronous Serial Port*), que proporciona una excelente interfaz de comunicación de los microcontroladores y diversos periféricos, entre los que destacan las memorias EEPROM serie, los convertidores A/D, los controladores de displays, etc. De esta forma, el usuario dispone de algunos pines del Puerto C, que correctamente programadas sirven para manejar directamente la comunicación serie. Además, el módulo MSSP admite dos de las alternativas más usadas en la comunicación serie síncrona:

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

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

El módulo MSSP consta básicamente de dos registros: SSPSR, que es un registro de desplazamiento que transforma la información serie en paralelo y viceversa, y el registro SSPBUF, que actúa como buffer de la información que se recibe o transmite. El funcionamiento del módulo MSSP es muy sencillo. En transmisión, el byte que se quiere transmitir se carga en el registro SSPBUF a través del bus de datos interno y automáticamente se traspa al registro SSPSR, que va desplazando bit a bit el dato, sacándolo ordenadamente al exterior al ritmo de los

pulsos de reloj. En recepción, los bits van ingresando al ritmo de reloj por un pin y se va desplazando en el SSPSR hasta que lo llenan, en cuyo momento la información se traspassa al SSPBUF, donde queda lista para su lectura. Este doble almacenamiento del dato recibido permite iniciar la recepción de un nuevo dato antes de que se haya leído el último. Cuando se han recibido 8 bits durante la recepción en SSPSR, se traspassa dicha información a SSPBUF y entonces el bit señalizador BF (*Buffer Full*) se pone a 1, al igual que el flag de interrupción SSPIF. Cualquier escritura en el SSPBUF se ignora durante una transferencia de información y se señala poniendo a 1 el bit WCOL, que advierte de este tipo de colisiones. Recae en la responsabilidad del programador pasar el bit WCOL a 0 una vez completada la escritura en SSPBUF.

4.8.1. Modo SPI

Permite la transferencia de datos de 8 bits en serie, que pueden ser transmitidos y recibidos de forma síncrona y simultánea. Para el establecimiento de la comunicación se utilizan tres líneas:

1. SDO (*Serial Data Out*), salida de datos en serie.
2. SDI (*Serial Data In*), entrada de datos en serie.
3. SCK (*Serial Clock*), reloj de sincronización.

Puede ser necesario utilizar una cuarta línea de control mas cuando el PIC que se utiliza trabaja en modo esclavo. En este caso, el pin #SS (selección de esclavo) se debe conectar a tierra. En la figura 4.26 se ofrece un esquema de un sistema interconectado en modo SPI en el que un PIC actúa como maestro y otros dos como esclavos. Las 4 líneas que utilizan se corresponden con los pines multifunción RC3/SDO, RC4/SDI, RC5/SCK y RA5/#SS.

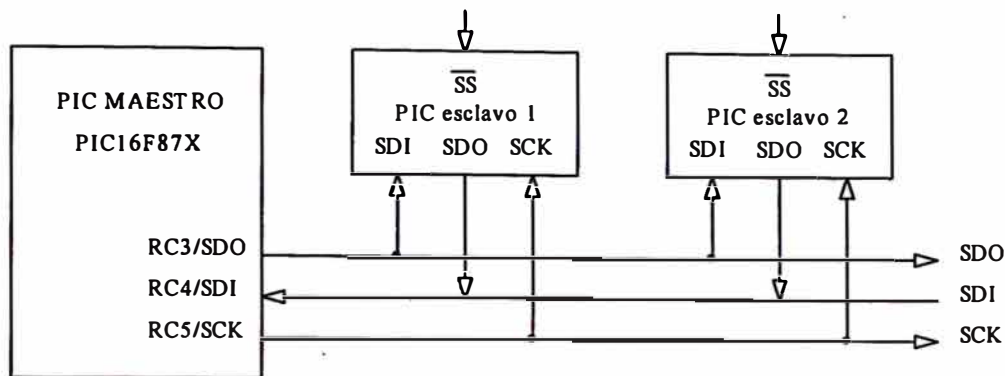


Figura 4.26 Comunicación SPI entre un PIC maestro y otros dos esclavos.

La conexión habitual del PIC maestro se suele realizar con circuitos de memoria con el objeto de ampliar su capacidad. Como se aprecia en la figura 4.25, la línea SDO del maestro se corresponde con las SDI de los esclavos y la línea SCK por la que circulan los pulsos de reloj siempre parte del maestro, que es el encargado de generar y controlar la sincronización.

Al comenzar el funcionamiento en el modo SPI, es necesario establecer las condiciones de trabajo programando adecuadamente los bits <5-0> del registro SSPCON y los bits <7:6> del registro SSPSTAT, para determinar las siguientes especificaciones:

1. Si trabaja en modo maestro, SCK debe ser salida.
2. Si trabaja en modo esclavo, la línea SCK del reloj debe ser entrada.
3. Hay que determinar la polaridad de la señal de reloj, es decir su estado de inactividad.
4. Se debe precisar el momento en el que se toma la muestra del dato de entrada, que puede ser en la mitad o al final del pulso.
5. Flanco de reloj activo, que puede ser al ascendente o descendente de SCK.
6. Rango de trabajo del reloj en modo maestro.

7. Selección del modo esclavo, en caso de trabajar de esta forma.

En la figura 4.27 se presenta la distribución y nomenclatura de los bits del registro SSPSTAT, que ocupa la dirección 94h y es uno de los que seleccionan las características mencionadas.

Registro SSPSTAT

SMP	CKE	D/ #A	P	S	R/ #W	UA	BF
bit 7							bit 0

Figura 4.27 Asignación de los bits del registro SSPSTAT para configurar el módulo MSSP en modo SPI o I²C.

En la figura 4.28 se presenta la distribución de los bits del registro SSPCON, que está ubicado en la dirección 14h y sirve para programar el módulo MSSP en los dos modos que puede trabajar.

Registro SSPCON

WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0
bit 7							bit 0

Figura 4.28 Asignación de los bits del registro SSPCON para configurar el módulo MSSP en modo SPI o I²C.

4.8.2. Modo I²C

El protocolo de comunicación serie I²C (*Inter-Integrated Circuit*) fue desarrollado por Philips para cubrir sus propias necesidades en la implementación de diversos productos electrónicos que requerían una elevada interconexión de circuitos integrados. Soporta circuitos integrados (C.I) fabricados mediante cualquier proceso (NMOS; CMOS, TTL, etc.). Cada C.I es reconocido por una única dirección, ya sea un microordenador, un excitador LCD, una memoria, un adaptador de teclado, etc. y

puede operar como receptor o transmisor dependiendo de la función del C.I que se considere. Obviamente un excitador de LCD es solo un receptor, mientras que una memoria puede actuar como transmisor o como receptor (ver figura 4.29).

Además de receptores o transmisores, los C.I pueden ser maestros o esclavos. Un maestro es un C.I que inicia una transferencia de datos sobre el bus y genera las señales de reloj que permiten esta transferencia. Se llama esclavo a cualquier dispositivo que sea direccionado por un maestro.

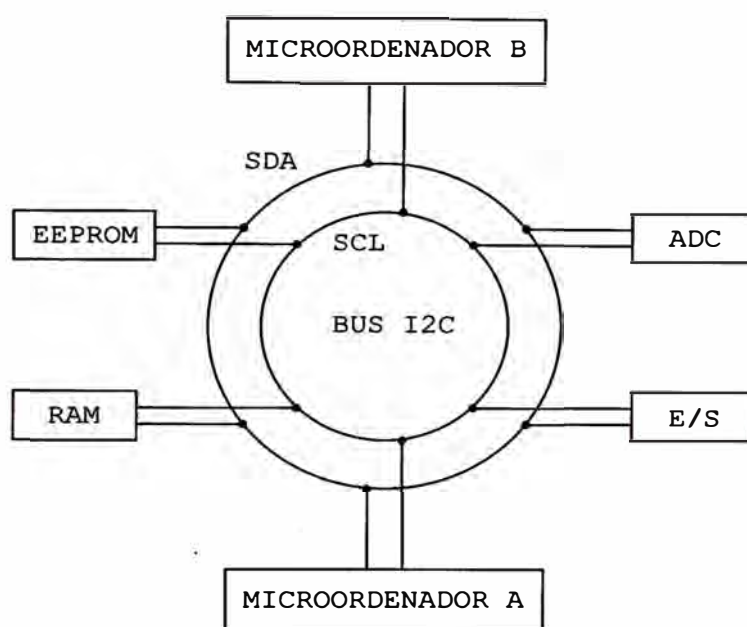


Figura 4.29 Configuración típica del bus I²C.

El protocolo I²C utiliza únicamente dos líneas para la transferencia de información entre los elementos que se acoplan al bus. Una de dichas líneas se dedica a soportar los datos y se llama SDA; la otra lleva los pulsos de reloj para la sincronización y recibe el nombre de SCL. Los pulsos de reloj siempre los genera el maestro y tiene la función de sincronizar la transferencias con todos los esclavos colgados a las dos líneas como se observa en la figura 4.30.

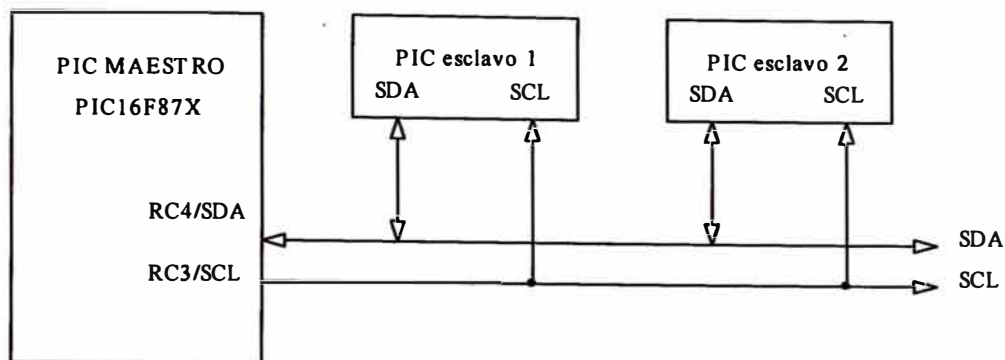


Figura 4.30 *El maestro se conecta con todos los esclavos en el bus I²C con solo dos líneas. Por SDA circulan los datos en serie y por SCL el maestro envía los pulsos de reloj que genera.*

Las líneas SDA y SCL son bidireccionales y están conectadas a tensión de alimentación Vdd a través de unas resistencias *pull up*. Cuando el bus esta libre, ambas se encuentran en estado alto. Las etapas de salida de los dispositivos conectados al bus deben ser colector abierto para poder realizar la función AND cableada. Se pueden transferir datos por el bus a una velocidad máxima de 100 Kbits/seg. La capacidad máxima en el bus es de 400 pF y el número de dispositivos conectados no debe superarla.

En los PIC16F87X, el bus I²C esta implementado en silicio tanto en el modo maestro (*Master*) como en el esclavo (*slave*). Como en general es el modo maestro el mas usado, las explicaciones se refieren a dicho modo.

Recuerde que en el modo maestro es el dispositivo, en este caso el PIC16F87X, el que inicia y finaliza la transferencia y genera los pulsos de reloj. También selecciona el esclavo al que se destina la información.

Como ya se ha comentado al hablar del bus I²C, en general, cada transferencia comienza con la condición de inicio (*Start*) y termina con la condición de parada (*Stop*). Ambas condiciones las genera el maestro, y la primera consiste en un flanco

descendente en SDA mientras SCL tiene nivel alto. La condición de parada es un flanco ascendente mientras SCL tiene nivel alto. Los datos que se colocan sobre la línea SDA son bytes, que comienzan con el MSB y terminan en el LSB, al cual sigue el noveno bit la condición de reconocimiento (ACK). Para este caso, el transmisor pone SDA= 1, mientras que el receptor pone SDA= 0 en dicho pulso de reloj, prevaleciendo sobre la línea el nivel bajo.

El primer byte que envía el maestro tras la condición de inicio contiene la dirección del esclavo con el que se desea realizar la comunicación. El código 0 se usa para realizar una llamada general sobre todos los esclavos. La dirección en realidad consta de 7 bits, estando destinado el octavo a indicar la operación a realizar (lectura/escritura). Tras este byte inicial de direccionamiento se manda otro byte que especifica las características de la operación a realizar.

Para activar el bus I²C hay que poner el bit SSPEN=1 que es el bit 5 del registro SSPCON (SSPCON <5>). A partir de ese momento los pines SDA y SCL quedan configuradas para soportar el protocolo I²C. Previamente dichos pines de la Puerta C (RC3/SCL y RC4/SDA) han de estar configuradas como entradas mediante la escritura de los bits correspondientes del registro TRISC. Se cuenta con seis registros para controlar el bus I²C.

1. SSPCON: registro de control 1.
2. SSPCON2: registro de control 2.
3. SSPSTAT: registro de estado.
4. SSPBUF: buffer para dato.
5. SSPSR: registro de desplazamiento no accesible directamente.
6. SSPADD: registro de dirección.

Mediante el bit CKE se ponen a nivel alto los pines SDA y ACL en el modo maestro y esclavo. Cuando CKE=0 los niveles de dichos pines se configuran según las especificaciones del bus I²C. Los 4 bits de menos peso del registro SSPCON sirven para seleccionar el modo de trabajo entre tres posibilidades:

1. Modo I²C maestro. $\text{Relej} = (\text{SPADD} + 1) (\text{OSC}/4)$.
2. Modo I²C esclavo con 7 bits para la dirección.
3. Modo I²C esclavo con 10 bits para la dirección.

4.9. Módulo USART

El módulo USART (*Universal Síncrono Asíncrono Receptor Transmisor*) es uno de los dos módulos serial E/S. El USART puede configurarse como un sistema de comunicación *full duplex* asíncrono que pueden comunicarse con dispositivos periféricos, tales como el monitor CRT y PC; o puede configurarse como un sistema *half duplex* síncrono para soportar periféricos como convertidores A/D, D/A, memorias, etc. El USART puede ser configurado de los siguientes modos:

1. Asíncrono (*full duplex*)
2. Síncrono-Maestro (*half duplex*)
3. Síncrono-Esclavo (*half duplex*)

El bit SPEN del registro RCSTA y los bits TRISC<7:6> tienen que ser establecidos para configurar los bits RC6/TX/CK y RC7/RX/DT como USART.

En el modo asíncrono, se usa la norma RS-232C, donde cada palabra de información o dato (8 o 9 bits) se envía independientemente de los demás. En el modo síncrono el USART se configura en modo maestro o esclavo; en estos modos los datos son transmitidos de manera *half duplex*.

4.10. Módulo convertidor A/D

De los recursos contenidos en el PIC16F876, quizá el convertidor A/D sea del que mas aplicaciones estamos acostumbrados a ver en nuestra vida diaria. Sensores de temperatura, presión, nivel, caudal, luz, campo magnético, PH, conductividad, humedad, humo, etc., son valores que nos interesa monitorizar; pero que necesitan ser traducidos para ser entendidos por el sistema de control, labor que realizan los convertidores A/D. La tarea de conversión es siempre igual, sea cual sea el sensor utilizado, teniendo en cuenta que lo que variara en el programa al cambiar de sensor será el tratamiento que se haga del valor una vez convertido.

El microcontrolador PIC16F876 tiene un convertidor A/D de 10 bits de resolución y 5 canales de entrada. A través del canal de entrada seleccionado, se aplica la señal analógica a un condensador de captura y mantenimiento (*sample and hold*) y luego se introduce al convertidor, el cual proporciona un resultado digital de 10 bits de longitud usando la técnica de aproximaciones sucesivas. La tensión de referencia es seleccionable entre la tensión interna V_{dd}, o externa que se introduce por el pin RA3/AN3/V_{ref+} o pin RA2/AN2/ V_{ref-}.

El convertidor A/D es el único dispositivo que puede funcionar en modo SLEEP, para ello el reloj del convertidor deberá conectarse al oscilador RC interno del convertidor A/D.

A continuación se indican los pasos que hay que efectuar para realizar una conversión A/D:

1. Configurar al modo de convertidor A/D.
 - Configurar los pines que actuaran como entradas analógicas, las que trabajan como E/S digitales y las usadas para la tensión de referencia (ADCON1).

- Seleccionar el reloj de la conversión (ADCON0).
 - Seleccionar el canal de entrada A/D (ADCON0).
 - Activar el modulo A/D (ADCON0).
2. Activar, si se desea la interrupción escribiendo sobre PIE1 y PIR1.
 - Borrar el señalizador ADIF.
 - Poner a 1 el bit ADIE y poner a 1 los bits habilitadores GIE y PEIE.
 3. Tiempo de espera para que transcurra el tiempo de adquisición.
 4. Inicio de la conversión.
 - poner a 1 el bit GO/#DONE (ADCON0).
 5. Tiempo de espera para completar la conversión A/D que puede detectarse:
 - Por exploración del bit GO/#DONE, que al completarse la conversión pasa a valer 0.
 - Esperando a que se produzca la interrupción si se ha programado, al finalizar la conversión.
 - Aunque no se permita interrupción, el señalizador ADIF se pondrá a 1 al finalizar la conversión.
 6. Leer el resultado en los 10 bits válidos de ADRESH:ADRESL y borrar el señalizador ADIF.
 7. Para una nueva conversión regresar al 1 o al 2.

4.10.1. Registros de Trabajo

El funcionamiento del convertidor A/D requiere la manipulación de cuatro registros:

1. ADRESH: parte alta del resultado de la conversión.
2. ADRESL: parte baja del resultado de la conversión.

3. ADCON0: registro de control 0.
4. ADECON1: registro de control 1.

En la pareja de registros ADRESH:ADRESL se deposita el resultado de la conversión, que al estar compuesta por 10 bits, solo son significativos 10 de los bits de dicha pareja.

El registro ADCON0 controla la operación del convertidor A/D, mientras que el ADCON1 sirve para configurara los pines del Puerto A como entradas analógicas o E/S digitales. Ver figura 4.31.

Registro ADCON0

ADCS1	ADCS0	CHS2	CHS1	CHS0	GO/ #DONE	----	ADON
bit 7							bit 0

Registro ADCON1

ADFM	----	----	----	PCFG3	PCFG2	PCFG1	PCFG0
bit 7							bit 0

Figura 4.31 *Asignación de bits de los registros de control del convertidor A/D ADCON0 y ADCON1.*

Los bits ADCON0<7:6> sirven para seleccionar la frecuencia de reloj que se emplea en la conversión, tal como se muestra en la tabla 4.3.

Si se elige como reloj para la conversión al oscilador RC interno del C A/D, éste sigue funcionando cuando ingresa el microcontrolador al modo SLEEP. En los restantes modos, se aborta la conversión y se desactiva el convertidor A/D.

Se designa como T_{DA} el tiempo que dura la conversión de cada bit y en el caso de trabajar con valores digitales de 10 bits, se requiere un tiempo mínimo de $12T_{DA}$. El valor de T_{DA} se selecciona por software mediante los bits <ADCS1:ADCS0> y en el PIC16F876 nunca debe ser menor de 1,6 microsegundos.

<ADCS1:ADCS0>	FRECUENCIA
00	$F_{osc}/2$
10	$F_{osc}/8$
10	$F_{osc}/32$
11	F_{RC} (oscilador RC interno C A/D)

SELECCIÓN DE T_{DA}		FRECUENCIA DE TRABAJO			
T_{DA}	<ADCS1:ADCS0>	20 MHz	5 MHz	1,25 MHz	333,3 KHz
$2 * T_{osc}$	00	100 ns	400 ns	1,6 μ s	6 μ s
$8 * T_{osc}$	01	400 ns	1,6 ns	6,4 μ s	24 μ s
$32 * T_{osc}$	10	1,6 μ s	6,4 μ s	26 μ s	96 μ s
RC	11	2-6 μ s	2-6 μ s	2-6 μ s	2-6 μ s

Tabla 4.3 Muestra los valores que toma T_{DA} , según la programación de <ADCS1:ADCS0> y la frecuencia de funcionamiento.

Los bits <CHS2-CHS0> seleccionan el canal por el que ingresa la señal analógica a convertir, según la Tabla 4.4.

<CHS2-CHS0>	CANAL
000	Canal 0 (RA0/AN0)
001	Canal 1 (RA1/AN1)
010	Canal 2 (RA2/AN2)
011	Canal 3 (RA3/AN3)
100	Canal 4 (RA5/AN4)
101	Canal 5 (RE0/AN5). Los PIC16F87X de 28 pines no tienen
110	Canal 6 (RE1/AN6). Los PIC16F87X de 28 pines no tienen
111	Canal 6 (RE1/AN6). Los PIC16F87X de 28 pines no tienen

Tabla 4.4 Elección del canal por el que ingresa la señal analógica.

El bit GO/#DONE es el bit de estado de la conversión. Poniéndolo a 1 se inicia la conversión y mientras este en 1 esta realizándose dicha operación. Cuando GO/#DONE pasa a 0 confirma el final de la conversión y la puesta del resultado en la pareja de registros ADRESH:ADRESL.

El bit ADON activa el funcionamiento del C A/D con 1 y lo prohíbe con 0.

4.10.2. Estructura Interna y Configuración del Convertidor A/D

En la figura 4.32 se ofrece un esquema del conexionado del convertidor A/D con los pines que soportan los canales de entrada y las de tensión de referencia.

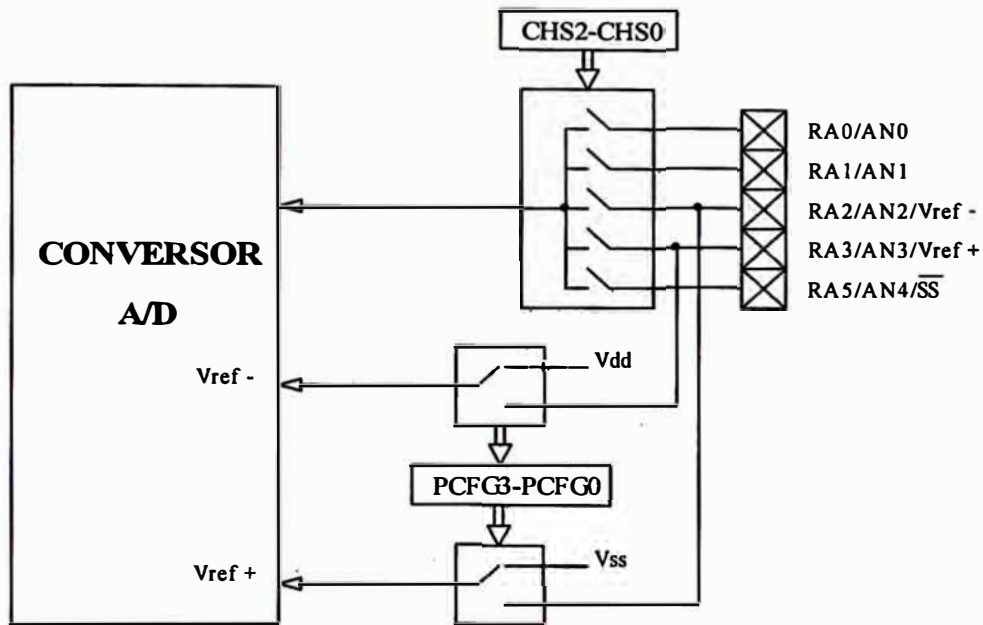


Figura 4.32 Estructura del conexionado del convertidor A/D.

El bit ADFM del registro ADCON1 selecciona el formato del resultado de la conversión. Si vale 1, el resultado está justificado en el registro ADRESH, que tiene sus 6 bits de más peso a 0; mientras que si vale 0 la justificación se realiza sobre el registro ADRESL, que tiene sus 6 bits de menos peso a 0 (ver figura 4.33).

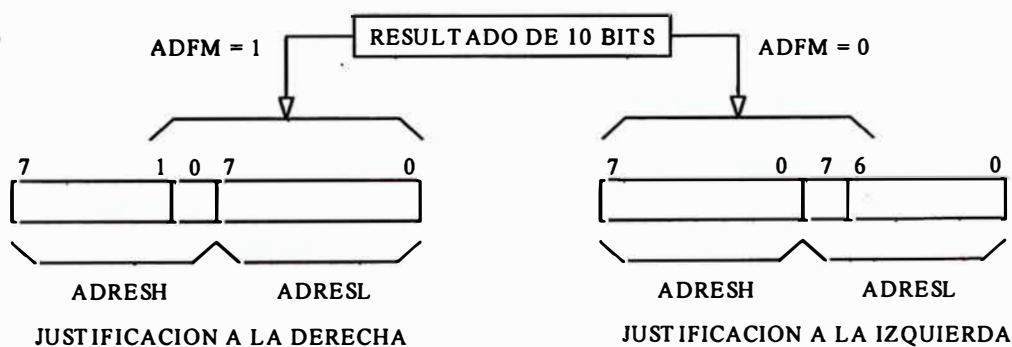


Figura 4.33 Alineamiento a la izquierda o derecha del resultado de 10 bits.

Los 4 bits <PCFG3-PCFG0> de ADCON1 se usan para configurar los pines de los canales de entrada al convertidor como analógicas o como digitales según tabla 4.5.

PCFG3: PCFG0	AN7/ RE2	AN6/ RE1	AN5/ RE0	AN4/ RA5	AN3/ RA3	AN2/ RA2	AN1/ RA1	AN0/ RA0	Vref+	Vref-	CHS/ REF
0000	A	A	A	A	A	A	A	A	Vdd	Vss	8/0
0001	A	A	A	A	Vref-	A	A	A	RA3	Vss	7/1
0010	D	D	D	A	A	A	A	A	Vdd	Vss	5/0
0011	D	D	D	A	Vref+	A	A	A	RA3	Vss	4/1
0100	D	D	D	D	A	D	A	A	Vdd	Vss	3/0
0101	D	D	D	D	Vref+	D	A	A	RA3	Vss	2/1
011x	D	D	D	D	D	D	D	D	Vdd	Vss	0/0
1000	A	A	A	A	Vref+	Vref-	A	A	RA3	RA2	6/2
1001	D	D	A	A	A	A	A	A	Vdd	Vss	6/0
1010	D	D	A	A	Vref+	A	A	A	RA3	Vss	5/1
1011	D	D	A	A	Vref+	Vref-	A	A	RA3	RA2	4/2
1100	D	D	D	A	Vref+	Vref-	A	A	RA3	RA2	3/2
1101	D	D	D	D	Vref+	Vref-	A	A	RA3	RA2	2/2
1110	D	D	D	D	D	D	D	A	Vdd	Vss	1/0
1111	D	D	D	D	Vref+	Vref-	D	A	RA3	RA2	1/2

Tabla 4.5 Configurar los pines del PIC para que actúan como entradas analógicas o como E/S digitales, según los bits <PCFG3-PCFG0>.

4.11. Recursos Especiales del Procesador

El microcontrolador PIC16F876 tiene un procesador con recursos especiales proyectado para maximizar la fiabilidad del sistema, minimizar el costo a través de la eliminación de componentes externos, proporcionar el ahorro de energía en los modos de operación, código de protección, diversas fuentes de interrupción y Reset, emulación y grabación en circuito, entre otros.

4.11.1. Palabra de configuración

Es una posición reservada de la memoria de programa FLASH, que ocupa la dirección 2007h y que solo es accesible durante la programación (grabación) del PIC.

El valor de sus bits (ver figura 4.34) determinan algunas características del sistema.

Palabra de Configuración

CP1	CP0	DEBUG	—	WRT	CPD	LVP	BODEN	CP1	CP0	#PWRTE	WDTE	FOSC1	FOSC0
bit 13										bit 0			

CP1:CP0 Bits de protección de la memoria de programa FLASH. En protección las instrucciones no se pueden leer, evitando copias.

CP1:CP0		Protección desde	Hasta
0	0	0000h	1FFFh
0	1	1000h	1FFFh
1	0	1F00h	1FFFh
1	1	No hay protección de la memoria	

- DEBUG** Modo depurador en circuito.
1: desactivado. RB7, RB6 actúan como líneas E/S.
0: activado. RB7, RB6 actúan en modo depurador utilizando MPLAB.
- WRT** Permiso de escritura en la memoria FLASH.
1: se puede escribir en la parte no protegida de la memoria FLASH.
0: prohibición de escritura.
- CPD** Código de protección de la memoria EEPROM de datos.
1: no hay protección.
0: protección de datos.
- LVP** Bit de permiso para programación en bajo voltaje.
1: RB3/PGM tiene permitida la grabación en bajo voltaje.
0: RB3/PGM funciona como E/S digital. Programación en alto voltaje.
- BODEN** Bit de permiso para el Reset por caída de tensión.
1: BOR activado.
0: BOR desactivado.
- #PWRTE** Bit de permiso para el temporizador de conexión de alimentación.
1: PWRT desactivado.
0: PWRT activado.
- WDTE** Bit de permiso del temporizador del perro guardián.
1: WDT activado.
0: WDT desactivado.

FOSC1:FOSC0 Selección del tipo de oscilador.

FOSC1:FOSC0		Tipo de oscilador
0	0	LP
0	1	XT
1	0	HS
1	1	RC

Figura 4.34 Asignación de los 14 bits de la Palabra de configuración.

4.11.2. Oscilador

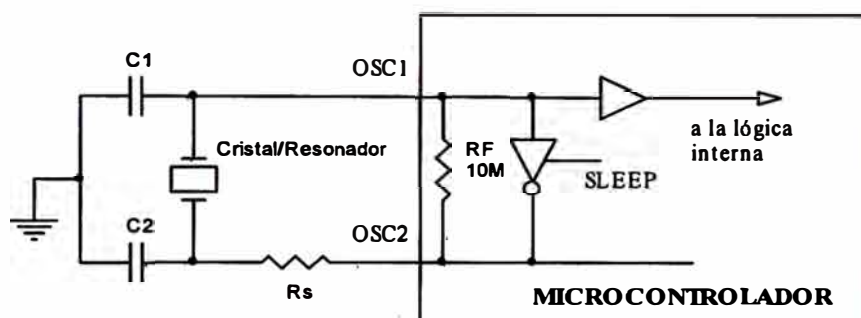
El microcontrolador admite cuatro tipos de osciladores (LP, XT, HS y RC) externos para aplicarles la frecuencia de funcionamiento Fosc. El tipo empleado debe

especificarse en dos bits (FOSC1 y FOSC0) de la Palabra de Configuración. En los tipos LP, XT o HS, un cristal de cuarzo o resonador cerámico es conectado entre los pines OSC1/CLKIN y OSC2/CLKOUT (ver figura 4.35). Los condensadores C1 y C2 se eligen de la tabla 4.6 según sea el tipo de oscilador.

A. Oscilador tipo LP: Oscilador de bajo consumo con cristal de cuarzo o resonador cerámico, diseñado para en el rango de frecuencias de 32 a 200 KHz.

B. Oscilador tipo XT: Oscilador de cristal o resonador para frecuencias comprendidas entre 100 KHz y 4MHz.

C. Oscilador tipo HS: Oscilador de alta velocidad comprendido entre 4 y 20 MHz, basado en un cristal cuarzo o un resonador cerámico.



Rs: necesario en algunas versiones tipo HS, en otros tipos corto circuito.

Figura 4.35 Esquema del general para los osciladores tipo LP, XT y HS. El resistor R_s sólo es necesario en algunas versiones tipo HS.

Tipo Oscilador	Frecuencia de Resonancia	C1	C2
LP	32 KHz	33 pF	33 pF
	200 KHz	15 pF	15 pF
XT	200 KHz	47-68 pF	47-68 pF
	1 MHz	15 pF	15 pF
	4 MHz	15 pF	15 pF
HS	4 MHz	15 pF	15 pF
	8 MHz	15-33 pF	15-33 pF
	20 MHz	15-33 pF	15-33 pF

Tabla 4.6 Elección de C1 y C2 para cristal de cuarzo.

D. Oscilador tipo RC: Se trata de un oscilador de bajo costo formado por una simple resistencia y un condensador, conectados al exterior del microprocesador. Proporciona una estabilidad mediocre de la frecuencia, cuyo valor depende de los valores de los elementos de la red R-C. La figura 4.36 muestra su esquema.

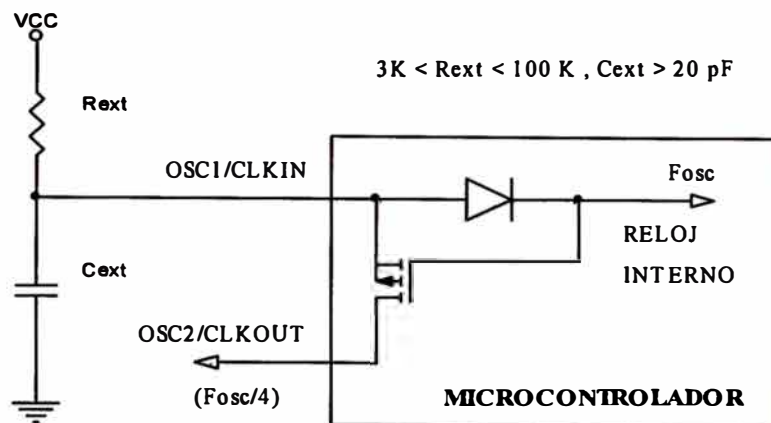


Figura 4.36 Esquema del oscilador RC. R_{ext} y C_{ext} son componentes externos.

4.11.3. Palabras de identificación

Ocupan cuatro palabras de la memoria de programa que se hallan en la dirección 2000h al 2003h y están reservadas para que el usuario las pueda emplear en funciones de comprobaciones o *checksums*, códigos de identificación, números de serie, fecha, modelo, lote, etc. estas cuatro posiciones solo son accesibles en lectura y escritura durante la operación de programación/verificación. Solo se deben emplear los 4 bits de menos peso de cada una de las palabras de identificación.

4.11.4. Reset

Tras un Reset, los SFR quedan con un valor conocido y el PC queda cargado con el valor 0000h (vector Reset) en todos los casos, excepto cuando se produce por el desbordamiento del WDT o cuando se despierta del modo SLEEP por una

interrupción, en cuyos casos el PC se carga con el valor PC+1. Existen diversas causas que reinician (Reset) el microcontrolador:

1. Reset por conexión de la alimentación (POR: *Power On Reset*). El valor de la tensión de alimentación Vdd sube entre 1,2 a 1,7 Voltios. Ver tabla 4.7.
2. Reset provocado por una caída de voltaje (BOR: *Brown Out Reset*). Vdd se encuentra por debajo de 4,2 Voltios, por un tiempo aproximadamente de 100 μ s.

#POR	#BOR	#TO	#PD	TIPO DE RESET
0	x	1	1	Conexión de alimentación (POR)
0	x	0	x	Ilegal
1	x	x	0	Ilegal
1	0	1	1	Por caída de tensión (BOR)
1	1	0	1	Por WDT (operación normal)
1	1	0	0	Por WDT (modo reposo)
1	1	u	u	Activación normal (#MCLR)
1	1	1	0	#MCLR en reposo o interrupción para despertar

Tabla 4.7 Muestra la codificación según el tipo de Reset que ocurre. La "x" significa que pueden tener valor 0 o 1 indistintamente y "u" significa su valor no varía con respecto al que tenía antes del Reset.

3. Activación del pin #MCLR, con nivel bajo durante una operación normal.
4. Activación del pin #MCLR, estando el PIC en modo SLEEP.
5. Reset por el desbordamiento del WDT en operación normal.
6. Reset por el desbordamiento del WDT durante el modo SLEEP.

En la figura 4.37 se muestra el esquema en el que se representan las causas y señales que originan un Reset. Se deduce que el Reset ocurre como consecuencia de la salida de un nivel lógico alto de la puerta OR, la cual es controlada por todas las causas que reinician al PIC.

Tras un Reset, el PC queda cargado con el valor 0000h en todos los casos, menos cuando se produce por desbordamiento del WDT o cuando se despierta.

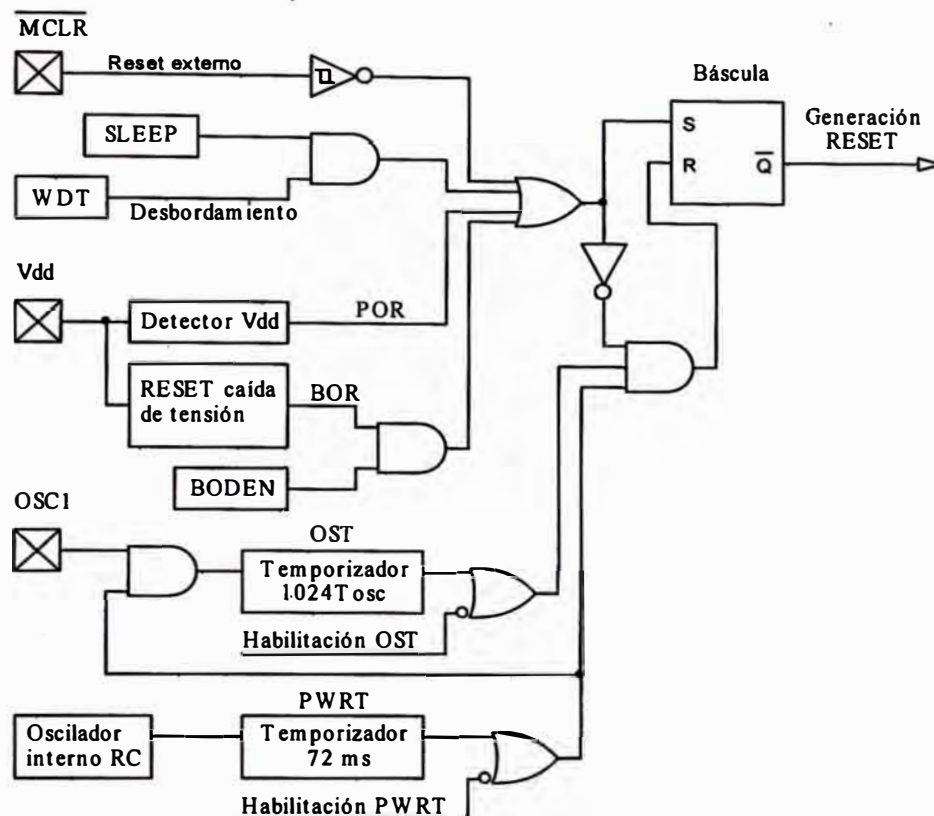


Figura 4.37 La generación del Reset se produce al activar la entrada S de la búsqueda, cuando sucede alguna de las condiciones aplicada a la puerta OR. Los temporizadores PWRT y OST retardan el Reset.

El temporizador PWRT (*Power up timer*) retarda el Reset un tiempo fijo de 72ms, con al finalidad de garantizar la estabilidad de Vdd. El temporizador OST (*Oscillator Start up Timer*) produce un retardo de 1024 ciclos de la frecuencia aplicada en el pin OSC1/CLKIN, asegurando que el cristal o resonador cerámico empleados en los osciladores tipo LP, XT o HS este estabilizado y en marcha. OST comienza a funcionar cuando termina el retardo de PWRT.

Tipo de Oscilador	#POR		#BOR	Despertar desde SLEEP
	PWRTE=0	PWRTE=1		
LP, XT, HS	72ms + 1024 Tosc	1024 Tosc	72ms + 1024 Tosc	1024 Tosc
RC	72 ms	---	72 ms	---

Tabla 4.8 Tiempo de salida en varias situaciones.

Registro de control de energía PCON: contiene 8 bits de los cuales solo se encuentran implementados 2 bits; tal como se muestra en la figura 4.38, que se encargan de señalar la ocurrencia o no de Reset por BOR o POR. El bit0 es #BOR bit de estado de *Brown On Reset*. El bit1 #POR bit de estado de *Power On Reset*.

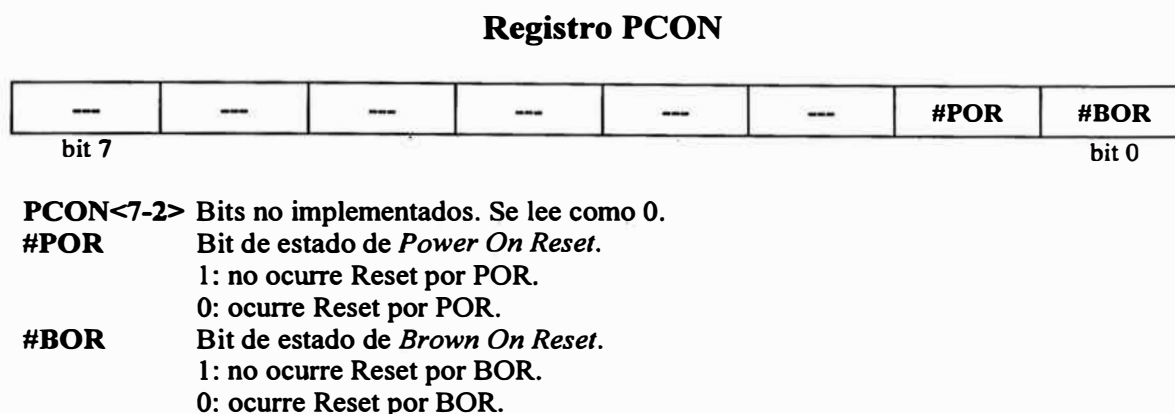


Figura 4.38 *Asignación de los bit del registro PCON.*

4.11.5. Interrupciones

Las interrupciones son desviaciones del flujo de control del programa, originadas asincrónicamente por diversos sucesos que no se hallan bajo la supervisión de las instrucciones. Dichos sucesos pueden ser externos, como la generación de un flanco o nivel activo en un pin del microcontrolador; o bien internos, como el desbordamiento de un contador.

El comportamiento del microcontrolador en una interrupción es similar al de la instrucción CALL (llamada a subrutina). En ambos casos se detiene la ejecución del programa en curso, se salva la dirección actual del PC en la Pila y se carga el PC con una dirección, que en el caso de CALL viene acompañando a la propia instrucción, y en el caso de una interrupción es una dirección reservada de la memoria de programa, llamada vector de interrupción.

La **rutina de servicio a la interrupción (ISR)** suele comenzar guardando en la memoria de datos algunos registros específicos del procesador, concretamente aquellos que la RSI va emplear. Antes de retornar al programa principal se recuperan los valores guardados y se restaura completamente el estado del procesador. Algunos procesadores salvan estos registros en la Pila, pero el PIC no dispone de instrucciones para meter (*push*) y sacar (*pop*) información de la Pila, utilizando para este fin GPR de la memoria de datos.

El microcontrolador PIC16F876 tiene hasta 14 fuentes de interrupción. El registro de control de interrupción (INTCON) guarda individualmente la respuesta de interrupción en sus bits. Igualmente tienen bits de permiso de interrupción individual y global. El bit de interrupción global GIE (*Global Interrupt Enable*) cuando vale 0 prohíbe todas las interrupciones. Al comenzar la RSI el bit GIE se pone automáticamente a 0, para no atender nuevas interrupciones hasta que se termine la que ha comenzado. En el retorno final de la interrupción, GIE pasa automáticamente a 1 para volver a explorar los señalizadores de interrupción, dicho retorno de interrupción se realiza mediante la instrucción RETFIE. Antes del retorno conviene borrar el señalizador de la causa de interrupción que se atiende, si bien los señalizadores se ponen automáticamente a 1 en cuanto se produce la causa que indican, la puesta a 0 se hace vía programa.

Los señalizadores de interrupción del pin RB0/INT, interrupción por cambio de estado Puerto B e interrupción por desbordamiento del TMR0 están contenidos en el registro INTCON.

Los señalizadores de interrupción de periféricos están contenidos en los registros PIR1 y PIR2. Los correspondientes bits de permiso de interrupción están contenidos

en los registros PIE1 y PIE2, y el bit de permiso de interrupción de periféricos se encuentra en el registro INTCON.

4.11.6. Temporizador perro guardián

Se trata de un contador interno de 8 bits, su control de tiempos es independiente del TMR0 y esta basado en una simple red RC. Su funcionamiento es opcional y puede bloquearse para que no funcione programando el bit WDTE (WDTE en 1, se activa) de la palabra de configuración. Este temporizador es muy útil en sistemas de control automáticos donde los fallos pueden desencadenar situaciones críticas.

Para evitar que se desborde el WDT hay que refrescarlo previamente, lo cual consiste en ponerlo a cero mediante las instrucciones CLRWDT y SLEEP. La instrucción CLRWDT borra el WDT y reinicia su cuenta; sin embargo la instrucción SLEEP además de borrar WDT detiene el sistema y lo pone en estado de reposo o de bajo consumo. Si no se desactiva el WDT al entrar en modo de reposo, al completar su cuenta provocara un Reset y sacará al PIC del modo de bajo consumo.

4.11.7. Modo reposo o SLEEP

Este modo de funcionamiento del microcontrolador se caracteriza por el reducido consumo de energía que requiere; el cual teniendo un consumo típico de 2 mA aproximadamente se reduce a menos de 10 μ A en modo reposo. Esta recomendado en aquellas aplicaciones en las que hay largos periodos de espera hasta que se produzca algún suceso asíncrono, como la pulsación de una tecla.

Para ingresar al modo reposo hay que ejecutar la instrucción SLEEP y se produce una situación muy especial de funcionamiento pareciendo que el

microcontrolador se ha congelado, requiriendo el mínimo suministro de energía para mantener el estado del procesador.

En modo reposo el pin TOCKI se conecta a Vdd o a tierra, para evitar la entrada de pulsos externos al TMR0. Como se detiene el oscilador principal que genera los pulsos Tosc, también se congela TMR0. Los pines de E/S mantienen el estado anterior al modo reposo y las que no se hallan conectadas a periféricos actúan como entradas de alta impedancia, se aconseja conectarlas a Vdd o tierra. El pin #MCLR debe conectarse a nivel alto. Sin pulsos de reloj el procesador se congela y deja de ejecutar las instrucciones hasta salir del modo reposo. Para salir de este modo (despertar) y pasar a ejecutar la instrucción direccionada por PC+1 existen varias causas, entre ellas tenemos:

1. Activación externa del pin #MCLR.
2. Desbordamiento del WDT trabajando en modo reposo.
3. Interrupción generado por el pin RB0/INT o por cambio de estado en uno de los 4 pines de mas peso del Puerto B.
4. Interrupción por alguno de los periféricos:
 - Interrupción del TMR1
 - Interrupción del módulo CCP en modo captura
 - Disparo especial del TMR1 funcionamiento asíncrono con reloj externo
 - Interrupción en el módulo de comunicación SSP
 - Transmisión o recepción del SSP en modo esclavo (SPI/I²C)
 - Transmisión o recepción del USART
 - Fin de conversión módulo A/D
 - Fin de la operación de escritura en la memoria EEPROM

4.11.8. Programación serial en circuito (ICSP)

El PIC permite que se grabe en su memoria de programa, el programa sobre el mismo circuito de aplicación. Esto permite a los fabricantes de productos finales construir la tarjeta electrónica con el microcontrolador sin grabar y realizar la grabación justo cuando se realiza la venta.

La programación en serie del PIC16F876 requiere 5 líneas: RB6 para los pulsos de reloj, RB7 para los datos en serie, Vdd a 5 V, Vss a tierra y la el tensión de programación especial Vpp que varia de 12 a 14 V aplicado al pin #MCLR/Vpp. El microcontrolador pasa al modo de programación/verificación colocando a nivel bajo los pines RB6, y RB7 aplicando la tensión Vpp; a partir de ese momento RB6 recibe los pulsos de reloj y por RB7 se transfieren los datos en serie.

4.11.9. Programación con voltaje bajo (LVP)

La programación con voltaje bajo (LVP: *Low Voltage Programming*), no requiere la tensión de 12 a 14 V. Para grabar en este modo, hay que poner el bit LVP en 1, que reside en la palabra de configuración y el pin RB3/PGM se debe conectar a nivel alto; entonces por el pin #MCLR/Vpp se aplica la tensión de 5 V mientras dura el modo de programación/verificación.

4.11.10. Depuración en circuito

Cuando el bit DEBUG de la Palabra de Configuración es programado con 0, la función depuración en circuito es habilitado. Esta función permite cuando se usa MPLAB® ICD funciones de depuración simple. Cuando el microcontrolador tiene esta función habilitada, algunos de los recursos no están disponibles para uso

general. Para el uso de esta función, el diseño debe implementar un conector de programación serial con las líneas #MCLR/Vpp, Vdd, Vss, RB6 y RB7.

4.12. Repertorio de instrucciones

Todos los modelos de microcontroladores PIC responden a la arquitectura RISC (Computador con Juego de Instrucciones Reducido). No solo implica que el número de instrucciones máquina que es capaz de interpretar y ejecutar el procesador es pequeño, solo 35 instrucciones para esta subfamilia (ver figura 4.39), sino que también posee las siguientes características:

1. Las instrucciones son simples y rápidas

Las operaciones que realizan las instrucciones de los procesadores RISC permite que sean ejecutadas mayoritariamente en un ciclo de instrucción, excepto las de salto, que tardan el doble.

2. Las instrucciones son ortogonales

Apenas tienen restricciones en el uso de operandos.

3. La longitud de las instrucciones y los datos es constante

Todas las instrucciones tienen la misma longitud, 14 bits en los PIC16F87X y todos los datos también, un byte.

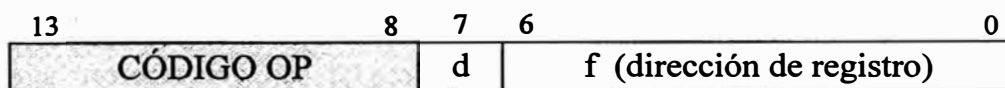
Tipos de formato

Las instrucciones de los PIC16F87X tienen 14 bits de longitud. El formato de la instrucción se divide en diferentes campos, referenciando cada uno a operandos o elementos que maneja la instrucción en la operación que realiza en el procesador se describen dichos campos:

1. Campo del código OP: los bits de este campo sirven para definir la operación que realiza la instrucción.
2. Campo de los operandos fuente (f) y destino (d): estos campos de bits definen los registros que actúan como operandos en la instrucción.
3. Campo de operando inmediato o literal (k): es un campo de bits que indica el valor de un operando.
4. Campo que referencia a un bit (b): campo de 3 bits que indica la posición de un bit concreto dentro de un registro de 8 bits.
5. Campo de la dirección del salto: en las instrucciones CALL y GOTO hay un campo de bits que contiene la dirección de la siguiente instrucción que hay que ejecutar. Dicho campo se carga en el PC en las instrucciones de salto incondicional.

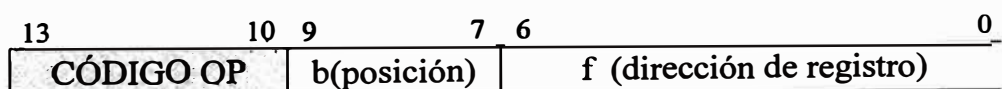
Los diversos formatos que admiten las instrucciones de los microcontroladores se clasifican en 3 grandes grupos, según el tipo de operación que desarrollan.

A. Operaciones orientadas a manejar registros de tamaño byte: Las instrucciones de este grupo presentan 3 campos y tienen como sintaxis: **nemónico f,d**.



d = 1, el registro destino es f; d = 0, el registro destino es W

B. Operaciones orientadas a manejar bits: las instrucciones de este grupo presentan 3 campos y tienen como sintaxis: **nemónico f,b**.



C. Operaciones orientadas al literal y a control: Las instrucciones de este grupo presentan 2 campos. Se presentan dos casos.

Formato general



Formato para las instrucciones CALL y GOTO



4.13. Características eléctricas

Se debe señalar que valores por encima del listado en “Rangos Máximos Absolutos” (tabla 4.9) puede causar daño permanente al dispositivo. Solo se señala un rango y no es implícito el funcionamiento del dispositivo.

Características	Rango
Temperatura ambiente	- 55 a 125 °C
Temperatura de almacenamiento	- 65 a 150 °C
Frecuencia de trabajo	20 MHz
Voltaje de grabación	5 V ó (12 a 14 V)
Voltaje en cualquier pin respecto a Vss (excepto Vdd, #MCLR y RA4)	- 0,3 V a (Vdd + 0,3 V)
Voltaje en Vdd con respecto a Vss	- 0,3 a 7,5 V
Voltaje en #MCLR con respecto a Vss	0 a 14 V
Voltaje en RA4 con respecto a Vss	0 a 8,5 V
Potencia total de disipación	1,0 W
Máxima corriente de salida del pin Vss	300 mA
Máxima corriente de entrada en el pin Vdd	250 mA
Entrada de corriente clamp, I_{IK} ($V_I < 0$ ó $V_I > V_{dd}$)	± 20 mA
Salida de corriente clamp, I_{OK} ($V_O < 0$ ó $V_O > V_{dd}$)	± 20 mA
Máxima corriente absorbida por cualquier pin E/S	25 mA
Máxima corriente suministrada por cualquier pin E/S	25 mA
Máxima corriente absorbida por el Puerto A y Puerto B	200 mA
Máxima corriente suministrada por el Puerto A y Puerto B	200 mA
Máxima corriente absorbida por el Puerto C	200 mA
Máxima corriente entregada por el Puerto C	200 mA

Tabla 4.9 Muestra los valores de “Rangos Máximos Absolutos.”

NEMÓNICOS	PARÁMETROS	OPERACIÓN	CICLOS	FORMATO 14 BITS	SEÑALIZADOR
INSTRUCCIONES QUE MANEJAN REGISTROS					
addwf	f, d	Suma W con f	1	00 0111 dfff ffff	C, DC, Z
andwf	f, d	AND de W con f	1	00 0101 dfff ffff	Z
clrf	f	Borrado de f	1	00 0001 1fff ffff	Z
clrw		Borrado de W	1	00 0001 0xxx xxxx	Z
comf	f, d	Complemento de f	1	00 1001 dfff ffff	Z
decf	f, d	Decremento de f	1	00 0011 dfff ffff	Z
decfsz	f, d	Decremento de f; brinca si es 0	1	00 1011 dfff ffff	Z
incf	f, d	Incremento de f	1	00 1010 dfff ffff	Z
incfsz	f, d	Incremento de f; brinca si es 0	1	00 1111 dfff ffff	Z
iorwf	f, d	OR de W con f	1	00 0100 dfff ffff	Z
movf	f, d	Movimiento de f	1	00 1000 dfff ffff	Z
movwf	f, d	Movimiento de W a f	1	00 0000 1fff ffff	
nop	f, d	No operación	1	00 0000 0xx0 0000	
rlf	f, d	Rotación de f a la izquierda	1	00 1101 dfff ffff	C
rrf	f, d	Rotación de f a la derecha	1	00 1100 dfff ffff	C
subwf	f, d	Resta de W a f	1	00 0010 dfff ffff	C, DC, Z
swapf	f, d	Intercambio de 4 bits de +peso por los 4 bits de -peso	1	00 1110 dfff ffff	
xorwf	f, d	OR exclusivo de W con f	1	00 0110 dfff ffff	Z
INSTRUCCIONES QUE MANEJAN BITS					
bcf	f, b	Puesta a 0 del bit b de f	1	01 00bb bfff ffff	
bsf	f, b	Puesta a 1 del bit b de f	1	01 01bb bfff ffff	
btfs	f, b	Test del bit b de f; brinca si 0	1 (2)	01 10bb bfff ffff	
btfs	f, b	Test del bit b de f; brinca si 1	1 (2)	01 11bb bfff ffff	
INSTRUCCIONES QUE MANEJAN LITERAL Y DE CONTROL					
addlw	k	Suma de k con W	1	11 111x kkkk kkkk	C, DC, Z
andlw	k	AND de k con W	1	11 1001 kkkk kkkk	Z
call	k	Llamada a subrutina	2	10 0kkk kkkk kkkk	
clrwdt		Borrado del Watchdog	1	00 0000 0110 0100	#TO,#PD
goto	k	Salto incondicional	2	10 1kkk kkkk kkkk	
iorlw	k	OR de k con W	1	11 1000 kkkk kkkk	Z
movlw	k	Movimiento de k a W	1	11 00xx kkkk kkkk	
retfie		Retorno de interrupción	2	00 0000 0000 1001	
retlw	k	Retorno devolviendo k en W	2	11 01xx kkkk kkkk	
return		Retorno de subrutina	2	00 0000 0000 1000	
sleep		Microcontrolador en reposo	1	00 0000 0110 0011	#TO,#PD
sublw	k	Resta W de k	1	11 110x kkkk kkkk	C, DC, Z
xorlw	k	OR exclusivo de k con W	1	11 1010 kkkk kkkk	Z

Figura 4.39 Principales características de las 35 instrucciones que componen el repertorio del microcontrolador PIC16F876.

CAPÍTULO V HERRAMIENTAS DE DESARROLLO PARA LOS PIC

5.1. Editor de código fuente

En general se puede usar cualquier editor de texto para editar el código fuente. La extensión **.asm* es la que deben llevar todos los programas escritos en ensamblador. Para la edición debemos tener en cuenta, que en las primeras filas se definen modo de ensamblado, tipo de microcontrolador, constantes, variables, etc.; en las siguientes filas (edición de programa) la primera columna del editor está reservada para las etiquetas (RAM o utilizados en saltos: condicionales, incondicionales, subrutinas, tablas). Estas deben empezar siempre por una letra. Además se debe tener en cuenta que no pueden usarse expresiones que ya utiliza el ensamblador tales como:

- Nemónicos o instrucciones
- Directivas del propio ensamblador.
- Nombres de los registros de función especial (SFR).
- Nombre de cada uno de los bit de los registros especiales y además,
- Los nombres no pueden duplicarse.

En las siguientes columnas, se puede comenzar a escribir el nemónico de la instrucción o las directivas del ensamblador.

Por último hay que decir que se pueden y se deben añadir comentarios después de un “;” (punto y coma) que son indispensables en muchos casos para seguir el razonamiento de los programas sin perderse. Cuando MPLAB encuentra “;” en todo

lo que sigue no genera código de maquina. En todos estos campos los espacios en blanco no son significativos y las líneas en blanco tampoco.

El uso de mayúsculas y minúsculas en los programas obedece a una serie de reglas o normas de estilo, comunes entre los programadores en ensamblador, que si bien no son obligatorias, facilitan la lectura del código fuente. Estas reglas son:

- Las directivas del ensamblador se escriben en mayúsculas.
- Los nombres de las variables se escriben en mayúsculas.
- Los mnemónicos de las instrucciones se escriben en minúsculas.
- El programa se escribe utilizando los tabuladores para definir las distintas columnas, tales como etiquetas, comienzo de líneas de programa y columna donde empiezan los comentarios.

5.2. Ensamblador

MPASM es un Macro Ensamblador Universal, es un ensamblador simbólico presentado en la PC. Soporta a todas las familias de microcontroladores PIC de Microchip.

MPASM ofrece maximizar capacidades de Macro, ensamblado condicional, y diversos formatos de fuente y listado. Genera varios formatos de código objeto que soportan las herramientas de desarrollo de Microchip, así como programadores de tercer orden. Puede generar el listado de programa directamente en un archivo de formato hex para programar el microcontrolador, o puede generar archivos objeto para MPLINK.

MPASM tiene las siguientes características que ayudan en el desarrollo de un programa en las aplicaciones de uso específico.

- Proporciona un traductor de código fuente ensamblador a código objeto, para todos los microcontroladores de Microchip.
- Capacidad de ensamblar Macros.
- Al ensamblar produce todos los archivos (Objeto O, listado LIST, símbolo COD, HEX, errores ERR y especiales) requeridos por el depurador simbólico en sistemas de emulación.
- Soporta formatos de listado hexadecimal (por defecto), decimal, octal y binario.
- El archivo HEX, puede ser utilizado por programadores de tercer orden.

MPASM proporciona un lenguaje valioso de directivas que apoyan en la programación del microcontrolador PIC. Las directivas son útiles dentro del desarrollo permitiendo un control completo sobre el proceso de ensamblado del código fuente, haciéndolo más rápido y facilitando su mantenimiento.

5.2.1. Vinculador MPLINK

MPLINK es un vinculador para el compilador C (MPLAB-C), y el ensamblador MPASM. MPLINK se introduce con el MPLAB-C v2.00 y sólo puede usarse con éstos o versiones posteriores. MPLINK permite producir el código modular, reusable con MPLAB-C y MPASM.

El control sobre el proceso de vinculación es realizado a través de un archivo vinculador de escritura y con las opciones de comandos en línea. MPLINK asegura que todos los símbolos de referencias están determinados y ese código, y datos se adaptan dentro del dispositivo disponible.

MPLINK combina múltiples entradas de módulos de objeto generados por MPLAB-C o MPASM, solo ejecutando el archivo ejecutable. Se asignarán las

direcciones reales de datos y la situación de funciones cuando MPLINK se ejecuta. Esto significa que usted dará las instrucciones a MPLINK para poner los códigos y datos en alguna parte dentro de las secciones asignadas de memoria, no en las ubicaciones físicas específicas.

Una vez vinculado memoria de programa y secciones de memoria RAM disponible en el dispositivo designado; analiza todos los archivos de entrada; intentará ubicar las rutinas de la aplicación en la memoria de programa y asignar estas variables de datos en la RAM disponible. Si hay demasiado código o demasiadas variables para ubicar, se mostrará en una ventana un mensaje señalando el posible error.

5.2.2. Librería MPLIB

La librería MPLIB es usada para crear módulos objeto usando MPASM v2.0, MPASMWIN v2.0, MPLAB-C v2.0 o posteriores. MPLIB maneja la creación y modificación de archivos de librería. Un archivo de librería es una colección de módulos de objeto que se guardan en un solo archivo. Hay diversas razones para crear los archivos de librería entre ellos tenemos:

- Las librerías hacen la vinculación más fácil. Desde que los archivos de librería pueden contener muchos archivos de objeto, el nombre de un archivo de librería puede usarse en lugar de los nombres de muchos archivos objeto que se encuentran separados antes de vincularse.
- Las librerías hacen proyectos que facilitan su mantenimiento. Si una librería es incluida en un proyecto, la adición o supresión de llamadas a esa librería no requiere un cambio en el proceso de vinculado.

- Las librerías ayudan a mantener códigos pequeños. Luego el vinculador sólo usa los archivos de objeto requeridos que se encuentran contenidos en la librería.
- Las librerías ayudan a llevar el propósito de un grupo de módulos de objeto. Las librerías hacen proyectos fáciles de mantenimiento. Si una librería es incluida en un proyecto, la adición o supresión de llamadas a esa librería no requiere un cambio en el proceso de vínculo.

5.3. Simulador

El programa simulador MPLAB–SIM es una herramienta libre que sirve para evaluar los diseños y productos de Microchip. El uso del simulador ayuda en gran escala a depurar el programa, particularmente los algoritmos. Dependiendo de la complejidad del diseño proyectado se compara el simulador con un emulador en el aspecto de beneficio tiempo/costo. Para proyectos que tienen múltiples ingenieros en el desarrollo, el simulador junto con un emulador puede controlar los costos y velocidad de depuración de problemas difíciles.

El Simulador MPLAB–SIM simula toda la serie de microcontroladores PIC en un nivel de instrucción. En cualquier instrucción dada, el usuario puede examinar o modificar cualquiera de las áreas de los datos o puede proporcionar estímulo externo a cualquiera de los pines. Las señales de entrada/salida pueden ser colocados por el usuario y la ejecución puede realizarse en; paso a paso, ejecute hasta la parada, o en modo de rastreo. MPLAB–SIM usa un soporte de depuración simbólica que es el MPLAB–C, y el MPASM. El programa simulador ofrece flexibilidad de bajo costo de desarrollo y depuración del código fuera del ambiente del laboratorio que le hace una excelente herramienta de desarrollo multiproyecto.

5.4. Emulador

El emulador universal en circuito MPLAB-ICE esta proyectado para proporcionar al ingeniero de desarrollo de producto una herramienta completa de diseño para microcontroladores PIC de 8 bit, con los requisitos del usuario. El programa de control de MPLAB-ICE es proporcionado por MPLAB-IDE que permite la corrección, ensamble y descarga, y depurado de la fuente desde un solo entorno.

Los módulos intercambiables del procesador permiten al sistema ser fácilmente reconfigurados para la emulación de diferentes procesadores. La arquitectura universal del MPLAB-ICE permite la expansión para soportar nuevos microcontroladores PIC. Este sistema modular consiste de un chasis de emulador, un módulo de procesador, un adaptador de dispositivo, y zócalos de transformación.

El Emulador MPLAB-ICE se ha diseñado como un sistema de la emulación en tiempo real con características avanzadas que generalmente se encuentran en las herramientas de desarrollo más caras. La plataforma de PC y el entorno Windows 95/98 se seleccionó para hacer mejor estas características, disponible en el usuario.

El sistema emulador MPLAB-ICE 2000 es un premio a la calidad proporcionando máxima velocidad de emulación, operación a bajo voltaje, 32K x 128-bits de rastreo, y hasta 65,535 puntos de interrupción. El funcionamiento complejo del MPLAB-ICE 2000 proporciona sofisticado análisis de rastreo y precisos puntos de interrupción. El analizador de rastreo captura las direcciones de ejecución en tiempo real, *Opcodes*, y lectura/escritura de datos externos. También rastrea todos los registros RAM usados mostrando las direcciones internas y valores de dato, así como acceso a todos los registros de función especial, incluyendo E/S, temporizadores, y periféricos.

5.5. Programador

El sistema de desarrollo PICSTART Plus se ejecuta en cualquier PC compatible en el entorno Windows. Además, los controladores están disponibles para clientes que utilizan Windows NT. Se conecta a la PC vía uno de los puertos COM (RS-232).

El programador PICSTART Plus entrega al desarrollador de producto la capacidad para programar el software del usuario en cualquiera de los microcontroladores soportados. El programa proporciona control total interactivo sobre el programador ejecutándose bajo MPLAB. PICSTART Plus no es recomendado para programar grandes cantidades, debido a que no hace la verificación de la memoria de programa a Vddmin y Vddmax.

El sistema de desarrollo PICSTART Plus de Microchip proporciona al ingeniero de desarrollo de producto una herramienta de diseño muy flexible, bajo costo que soporta todos sus microcontroladores de 8 bits. Los encapsulados tipo DIP de hasta 40 pines son directamente soportados. Otros tipos de encapsulado pueden ser soportados con su respectivo adaptador (por ejemplo se encuentra disponible el adaptador PLCC para el PIC16C92X, PIC17C75X, y PIC17C76X).

5.6. Entorno de desarrollo integral

5.6.1. MPLAB-IDE

MPLAB es un programa para Windows, basado en un Entorno de Desarrollo Integrado (IDE) de Microchip Technology Inc. que incorpora a las familias de microcontroladores PIC.

MPLAB-IDE le permite escribir, depurar, y optimiza las aplicaciones de microcontroladores PIC para el diseño estable del producto. MPLAB-IDE incluye un

editor de texto, administrador de proyectos, ensamblador/compilador y simulador. MPLAB-IDE también soporta emuladores (PICMASTER® y MPLAB-ICE), programadores (PICSTART® Plus y PRO MATE® II), y otras de Microchip o herramientas de desarrollo de tercer orden.

Microchip provee programas de soporte en www.microchip.com, para ayudar rápidamente al ingeniero diseñador a familiarizarse con sus herramientas de desarrollo y con sus familias de microcontroladores.

El software MPLAB-IDE brinda una facilidad de desarrollo de programas para microcontroladores de 8-bits. MPLAB-IDE basó en ventanas las aplicaciones que contiene:

- Característica completa de editor.
- Tres modos de operación:
 - Editor.
 - Simulador.
 - Emulador.
- Un administrador de proyecto.
- Barra de herramientas configurable y personalizado.
- Una barra de estado con información del proyecto.
- Extensa ayuda en línea.

MPLAB-IDE permite:

- Crear y editar los archivos fuente. Esto incluye:
 - Lenguaje ensamblador MPASM.
 - Lenguaje C (MPLAB-C).
 - Otros lenguajes de tercer orden.

- En una sola acción ensamblar (o compilar) y descargar las herramientas para el microcontrolador deseado (automáticamente se actualiza toda la información).
- Use el depurador de código fuente (*Debug*). Depura el código fuente con la ayuda de la ventana *Build Results* que muestra los errores existentes luego de ensamblar, compilar, y *Linker* cuando generan archivos ejecutables.

Memoria de programa.

Archivos fuente.

Listado de programa.

- Depurar utilizando el simulador o emulador.
- Programar el microcontrolador.

La figura 5.1 muestra el escritorio típico de MPLAB-IDE en medio de un proyecto. Algunas de las principales características son:

- Barra de herramientas, múltiples opciones y configurable por usuario.
- Estado, modo de configuración, e información del botón en la barra de estado.
- Las ventanas múltiples, tal como:

Código fuente.

Listado de programa, ventana de memoria de datos EEPROM.

Ventana de registros de propósito general.

Ventana de registros de función especial.

Ventanas de observación (para mirar el registro específico).

Ventana de parada de reloj para los cálculos de tiempos o ciclos.

El macro ensamblador MPASM conjuntamente con MPLINK/MPLIB proporcionan los archivos de datos de la memoria de programa, listado de programa, y los archivos especiales requeridos para la depuración simbólica.

Escritorio de MPLAB-IDE mostrando las principales ventanas de observación en la simulación del programa ensamblado.

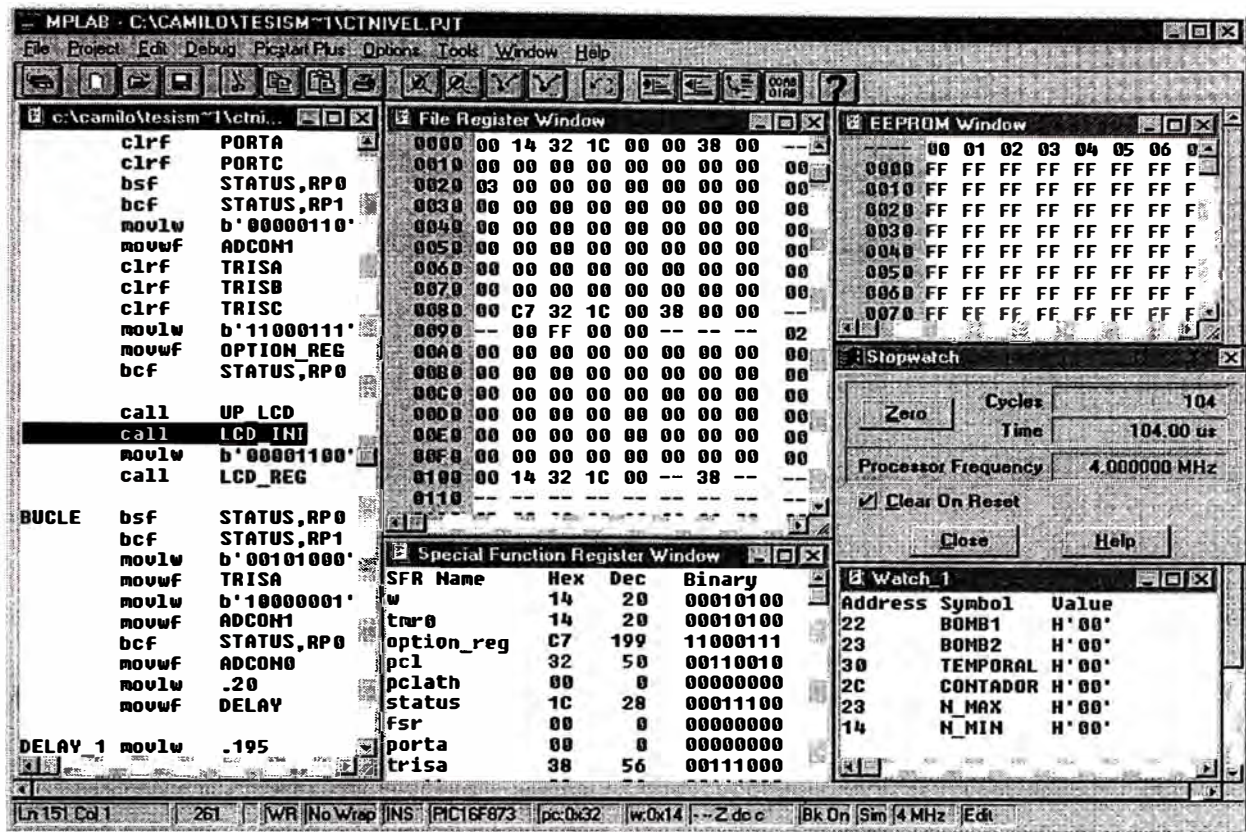


Figura 5.1 Escritorio de MPLAB-IDE mostrando las principales ventanas de observación en la simulación del programa.

El programa simulador MPLAB-SIM permite al usuario diseños estables en microcontroladores aislando los problemas de código y depurándolos. Simula las funciones del así como la mayoría de los periféricos del microcontrolador. Es particularmente conveniente para optimizar algoritmos donde la emulación en tiempo real no se requiere. Luego de realizar la depuración y ensamblado final se puede programar el microcontrolador PIC.

5.6.2. Herramientas de desarrollo de MPLAB-IDE

MPLAB-IDE integra diversas herramientas para proporcionar un completo entorno

de desarrollo. Su principal característica es que cuenta con un Administrador de Proyectos, el cual se encarga de manejar el proyecto; para su creación, edición, compilado, depurado, simulación, emulación y programación del microcontrolador. Cuando usa un proyecto, las diversas tareas; tales como ensamblado, simulación o emulación se realizan fácilmente haciendo solo un clic con el mouse.

A continuación se describe brevemente las herramientas de MPLAB-IDE.

A. Editor

Use el Editor de MPLAB-IDE para crear y editar los archivos de texto tales como archivos fuente, código, y archivo vinculador (*Linker*).

B. Ensamblador universal MPASM/vinculador MPLINK/librería MPLIB

El ensamblador MPASM permite ensamblar el código fuente fuera del entorno de MPLAB. MPLINK crea la aplicación final (listado de programa) vinculando archivos objeto desde MPASM y/o MPLAB-C. MPLIB administra librerías personalizadas utilizadas para generar archivos objeto, buscando el máximo rehusos del código. Estas herramientas ya se describieron anteriormente.

C. Compilador C MPLAB-C

El MPLAB-C es un completo compilador C para la familia de los microcontroladores PIC. El compilador proporciona las capacidades de integración poderosa y facilidad de uso no encontradas en otros compiladores. Para facilitar el nivel de depuración de la fuente, el compilador proporciona información simbólica que es compatible con la memoria que muestra el MPLAB-IDE, ventanas de Reloj,

ventanas de registro, ventanas de observación, listado de programa, etc.

D. Simulador MPLAB–SIM

El programa de simulación MPLAB–SIM modela las instrucciones (mnemónicos), las entradas/salidas y algunas características especiales del microcontrolador. El simulador imita la ejecución de un programa del microcontrolador en la PC.

E. Emuladores MPLAB–ICE y PICMASTER

Estos emuladores usan hardware para emular los microcontroladores PIC en tiempo real, con o sin un sistema designado. El MPLAB–ICE es el más nuevo emulador de Microchip.

F. Programadores PROMATE ® II y PICSTART ® Plus

Desarrolle el código con el simulador o emulador, ensámblelo o compílelo, y entonces use una de estas herramientas para programar los dispositivos. Todo esto puede ser logrado con MPLAB–IDE. Aunque PROMATE II no requiere MPLAB para operar, programar con MPLAB–IDE es más fácil.

G. Herramientas de desarrollo de tercer orden

Muchas otras compañías tienen las herramientas de desarrollo para productos de Microchip que trabajan con MPLAB–IDE, estos incluyen compiladores, emuladores, programadores, y otras herramientas.

CAPÍTULO VI INGENIERÍA DEL SISTEMA IMPLEMENTADO

6.1. Hardware del sistema

El hardware diseñado e implementado para el **Sistema de Telemando y Automatización de Bombeo de Agua Basado en Microcontroladores PIC**, esta constituido por: (1) El circuito de la Fuente de Alimentación, el cual presenta protección escalonada contra interferencias electromagnéticas; (2) El circuito de Telemando, que se encarga de realizar el control a distancia vía la línea telefónica y (3) El circuito de Control de Nivel, que se encarga de automatizar el sistema de bombeo de agua. Estos circuitos fueron diseñados y algunas partes de estos circuitos modelados y simulados, utilizando las herramientas de *OrCAD Release 9.1*.

En la figura 6.1 se muestra el diagrama de bloques del sistema global.

6.1.1. Circuito de la fuente de alimentación

En la figura 6.2 se muestra el diagrama de bloques del circuito de la fuente de alimentación. Debemos destacar el circuito de protección contra descargas atmosféricas (protección escalonada), el cual se describe a continuación. Consta de una etapa de descarga de muy alta tensión (transitorio de mas de 10 KV con duración de 8us a 20us), formado por los Gap's que están puestos entre las líneas con respecto a tierra, realizan descargas formando arcos a partir de 1,5KV y derivándolas hacia tierra; Luego siguen en cada línea la bobina de choque (construido con alambre de

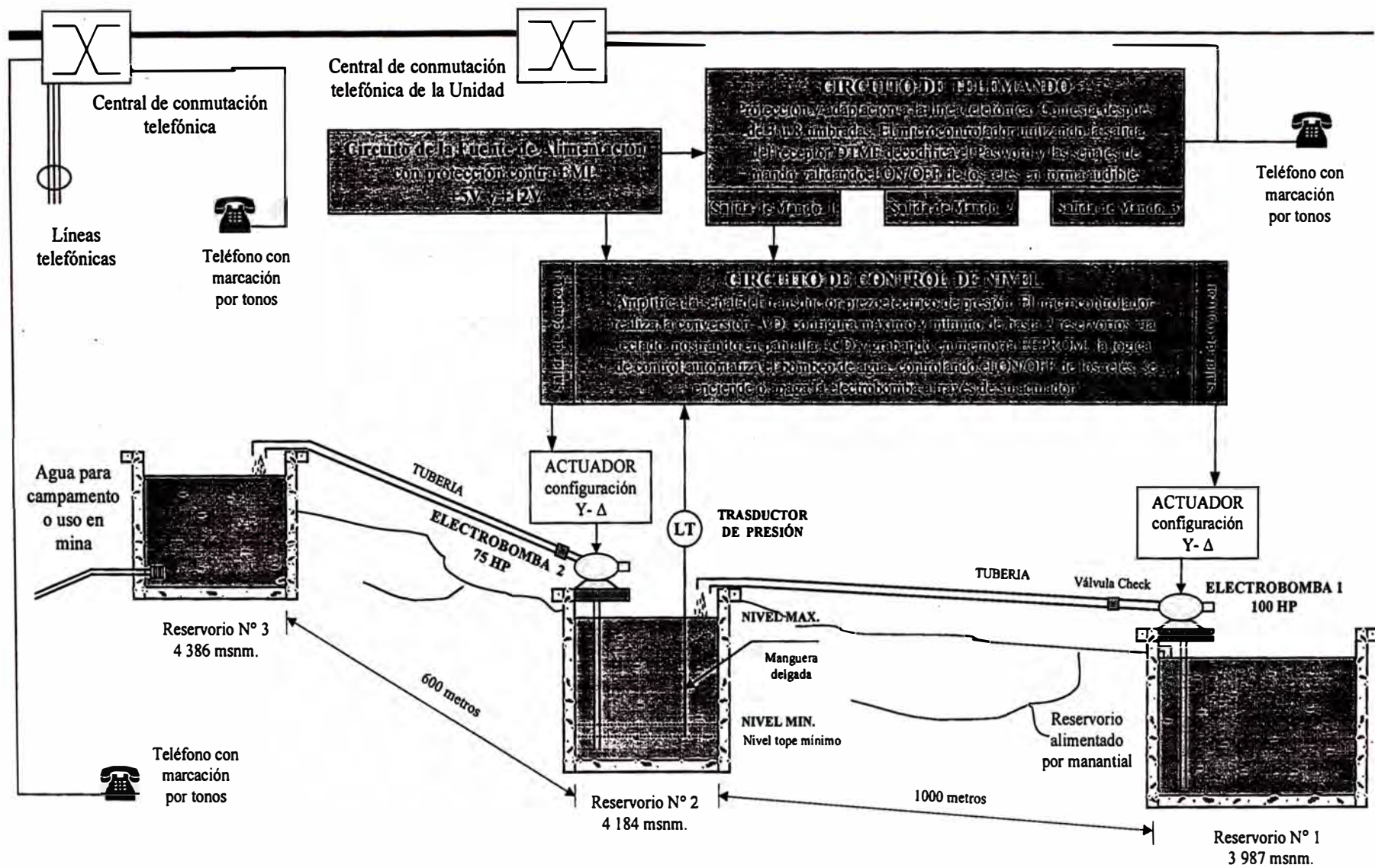


Figura 6.1 Diagrama de bloques: Sistema de Telemando y Automatización de Bombeo de Agua Basado en Microcontroladores PIC.

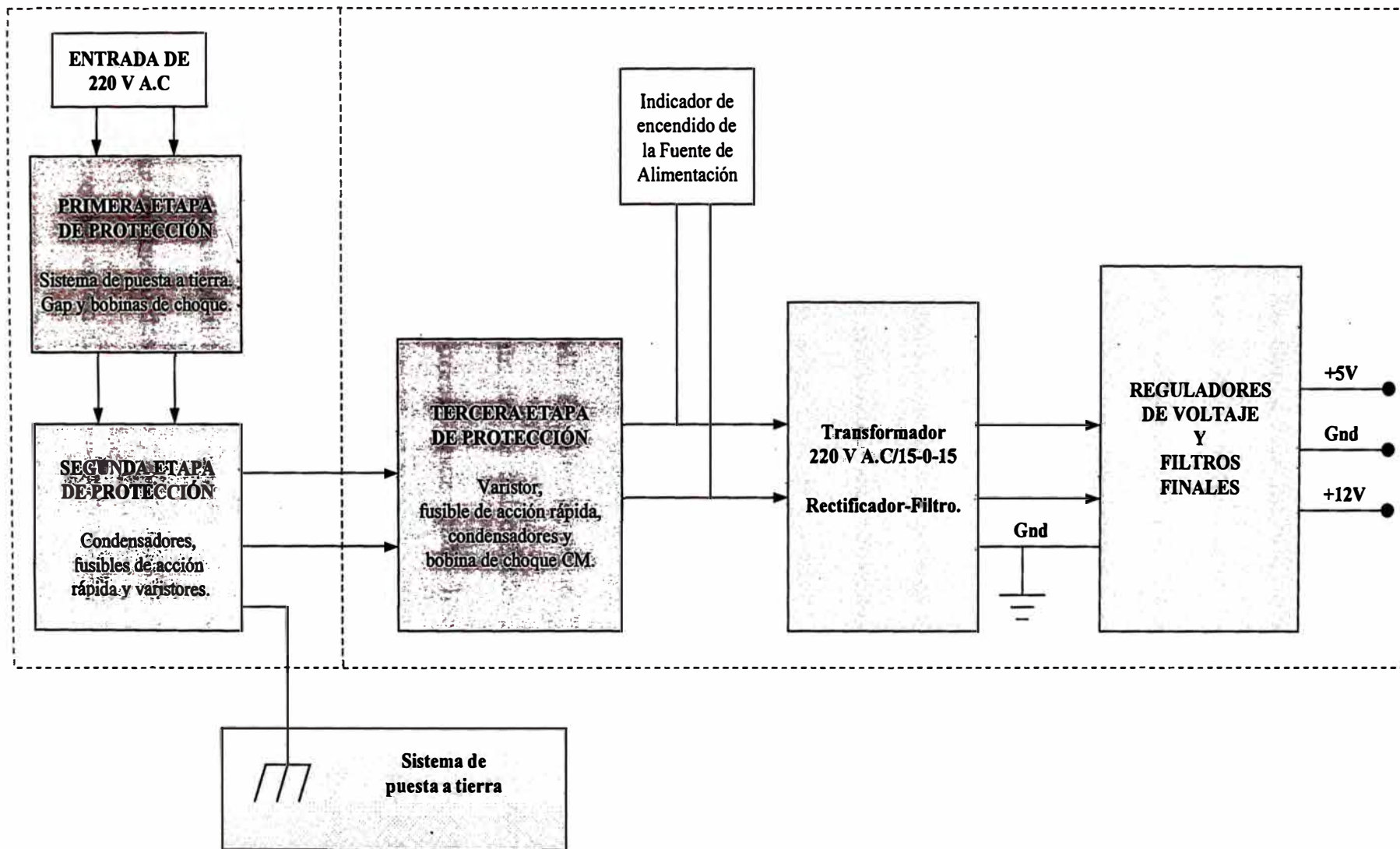


Figura 6.2 Diagrama de bloques del circuito de la Fuente de Alimentación, que muestra las etapas de protección (barreras) contra interferencias electromagnéticas.

Cobre esmaltado 18AWG de aproximadamente 40 espiras circulares, con inductancia entre 10uH a 15uH) que impiden el paso de las altas corrientes D.C transitorios facilitando el trabajo de los Gap's; Los fusibles colocados en cada línea actúan cuando los varistores colocados en delta (una entre líneas y una entre la línea con tierra), no soportan su máximo voltaje de trabajo, comportándose casi como un corto circuito, lo cual activa los fusibles. Finalmente los condensadores de bloqueo (valores entre 10nF a 470pF y con voltajes de hasta 4 KV D.C) ofrecen muy baja impedancia, derivando la sobre corriente (tensión) hacia el punto de tierra. Los fusibles y bobinas son diseñados de acuerdo a la carga a manejar.

El esquema electrónico del circuito de protección de entrada se muestra en la figura 6.3; debe ser montado sobre fibra de canibacita de un centímetro de espesor, procurando buen aislamiento debido a que esta expuesta a muy altas tensiones.

Partiendo de un voltaje de suministro de 220 V A.C, se minimiza los altos voltajes transitorios de las descargas atmosféricas con el circuito de protección y a esta se filtra las señales EMI. El filtrado de la EMI en modo diferencial es usualmente reducido usando un par de inductancias acopladas llamado choque en modo común y dos capacitores desde cada línea a tierra, de capacidades relativamente bajos. En este arreglo la alta impedancia del inductor a frecuencias altas es usada para bloquear la salida de corriente hacia la víctima. En el modo diferencial el filtrado requiere capacitor entre las líneas.

Una vez realizado el filtrado EMI, esta señal pasa por el transformador para luego rectificar utilizando dos diodos y filtrando con el condensador electrolítico. El regulador de voltaje es un C.I en encapsulado TO220 que recibe una entrada de voltaje D.C relativamente constante y suministra una salida de voltaje D.C, que el

regulador mantiene fijo o regulado sobre un amplio rango de corriente de carga, y voltaje de entrada. Las tensiones de la fuente de alimentación son: $V_{cc}=+5$ V D.C y $V_{cc1}=+12$ V D.C. En la figura 6.4 se muestra el esquema electrónico del circuito.

6.1.2. Circuito de Telemando

Cuando se enciende el circuito de Telemando, el microcontrolador PIC16C55 configura puertos, inicializa los GPR, timbradas, las salidas de mando =0 y lee el estado del pulsador operación manual.

Si en cualquier momento se habilita el pulsador (SW1) se ingresa a operación manual, caso contrario el microcontrolador PIC16C55 espera detectar la llamada telefónica, realizando el control a distancia vía teléfono.

En la figura 6.5 se muestra el diagrama de bloques del circuito de Telemando.

A. Operación manual

En este modo de operación, se conecta o desconecta cualquiera de las salidas de mando (en una de las cuales se encuentra el circuito de Control de Nivel) manualmente mediante el pulsador llamado operación manual. Cuando es conectada, el led correspondiente se encenderá.

Para conectarla accione el pulsador el número de veces que señala la salida de mando (1, 2 o 3) y nuevamente el mismo número de veces, si desea desconectarla.

B. Control por teléfono

La línea telefónica que ingresa al circuito de Telemando presenta un circuito de protección de entrada contra interferencias electromagnéticas que está formado por

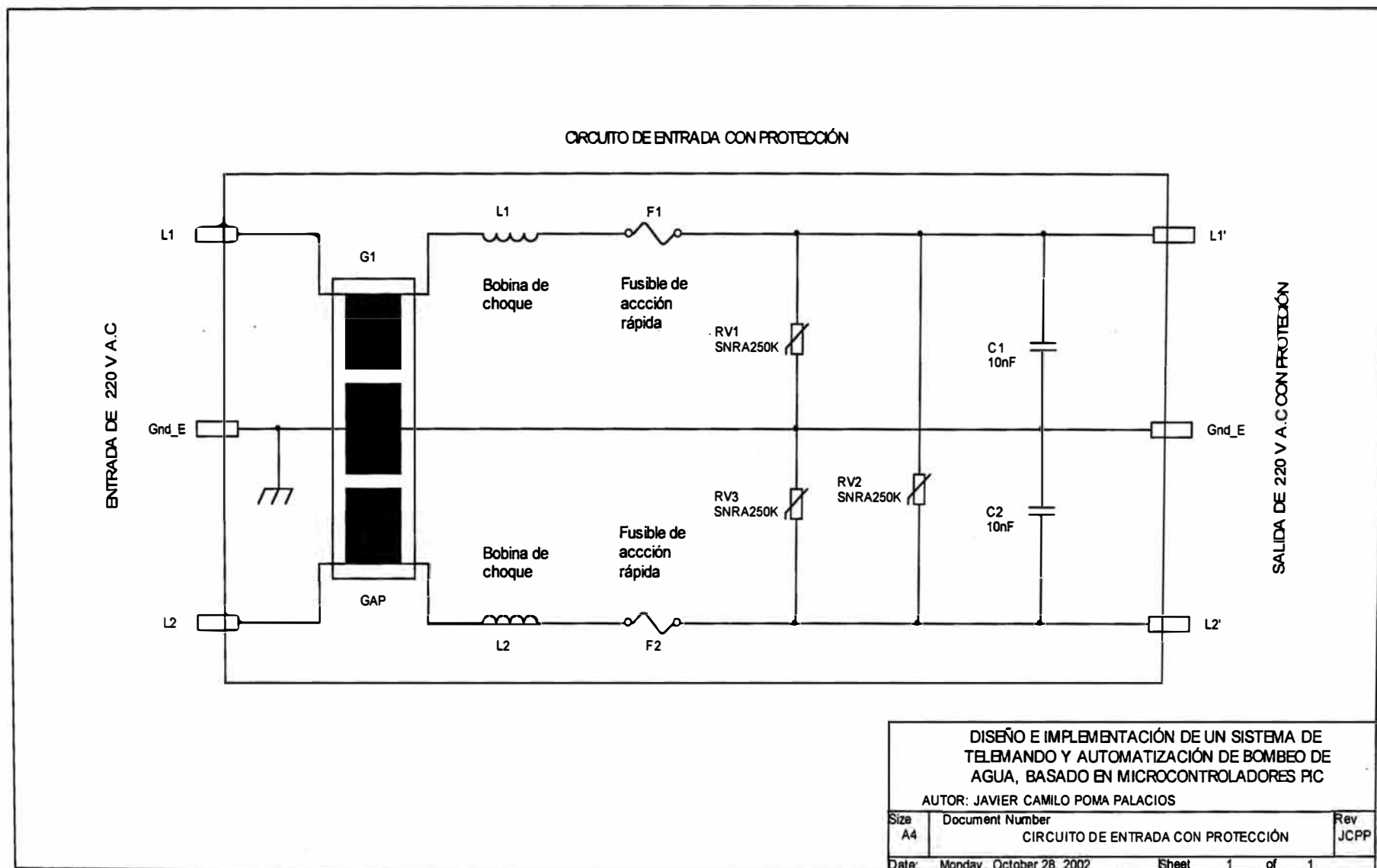


Figura 6.3 Esquema electrónico del circuito de entrada con protección que utiliza la Fuente de Alimentación o la salida de control.

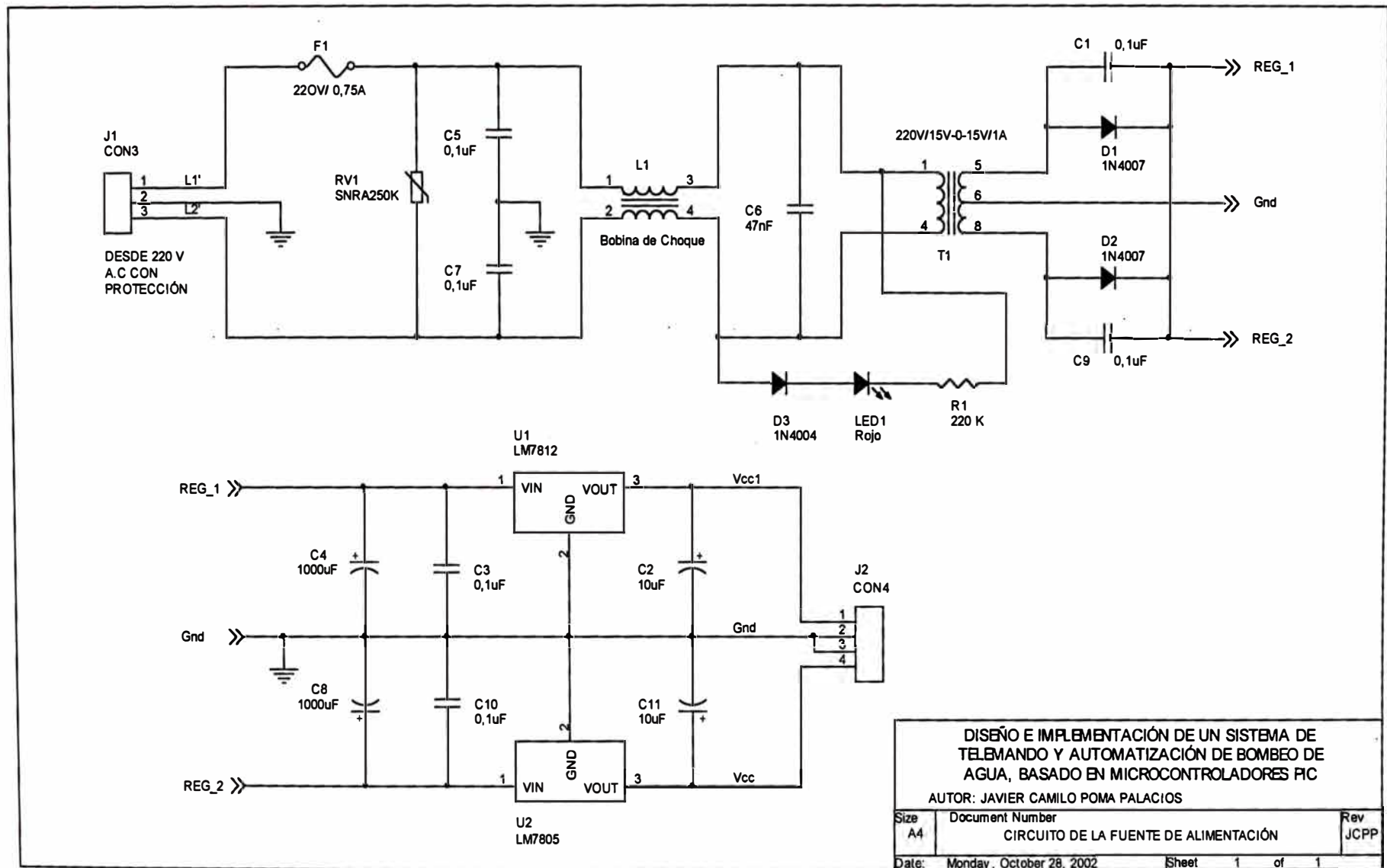


Figura 6.4 Esquema electrónico del circuito de la Fuente de Alimentación.

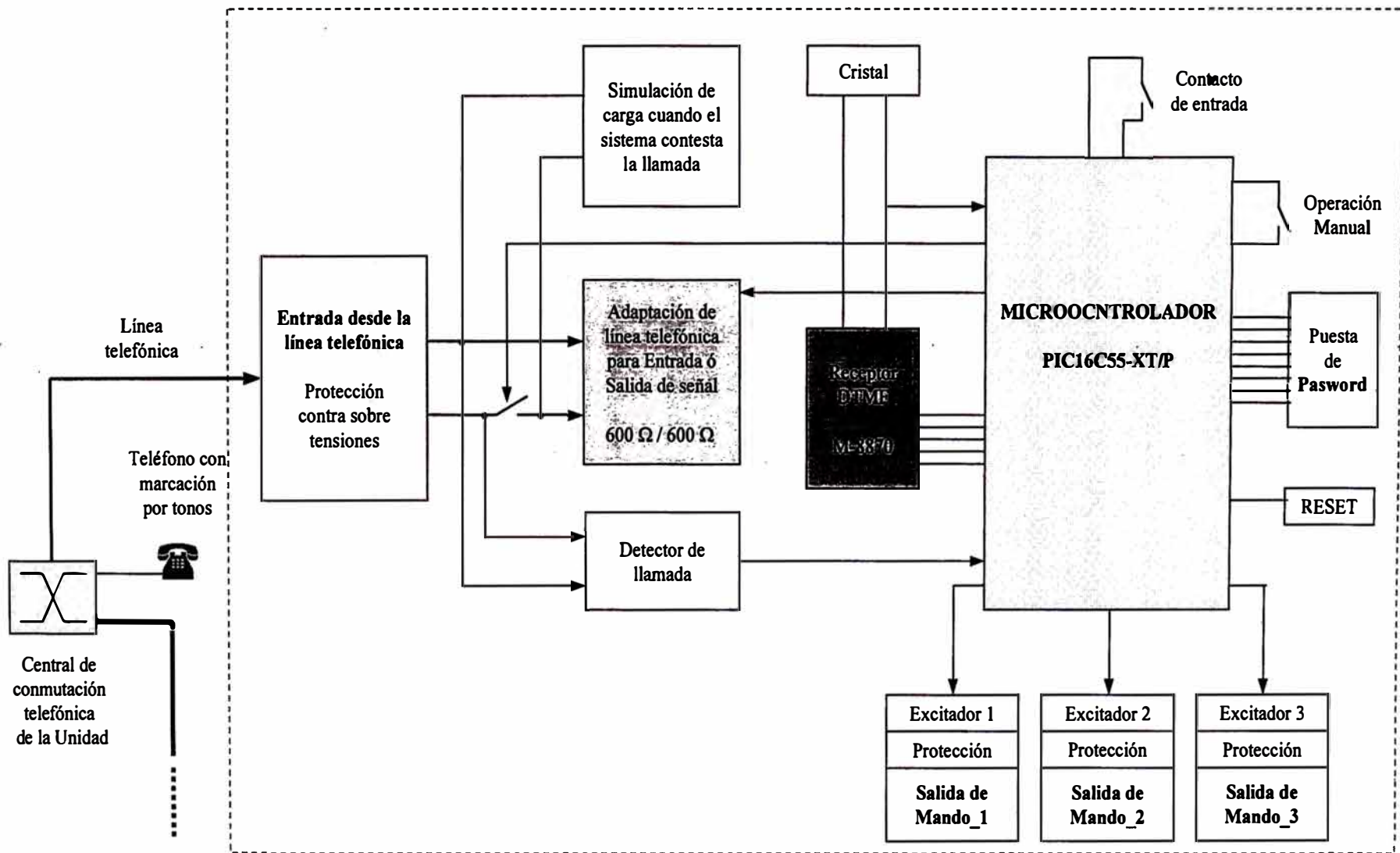


Figura 6.5 Diagrama de bloques del circuito de Telemando, que muestra la protección a la conexión de la línea telefónica.

C14, C15, L1, L2, TVS1, TVS2 y RV1. El circuito detecta el timbrado (ver figura 6.6) o llamada vía C13, R18, opto acoplador 4N25 y los componentes asociados, ingresando al microcontrolador PIC16C55 a través del pin 1 para ser contados y al cumplirse 3 u 8 timbradas, el PIC16C55 ordena contestar activando el relé K1 a través del pin 19. Se simula una carga de 35 mA en paralelo (circuito asociado a los componentes Q1, ZD1, C1, D1, D2, D3, D4, etc. mostrado en la figura 6.7), que es la corriente que se consume al descolgar el teléfono y se acopla la impedancia con el transformador de acoplo $600\Omega/600\Omega$ para ingresar o sacar señal de línea telefónica. Luego el microcontrolador le informa el estado de la entrada y salidas (el informe es de tipo sonoro: un sonido de corta duración, señala entrada o salida desconectada y dos sonidos de corta duración, señala entrada o salida conectada), además revisa el pin 25 en donde se debe registrar un nivel alto, proveniente del M-8870, que indica que se recibió un valor DTMF válido. Luego de lo anterior el microcontrolador solo debe leer el dato que está presente en el puerto RC <6-3> el cual corresponde al dígito pulsado en la tecla del teléfono. Comprobará si el código de seguridad es el correcto, teniendo un máximo de 3 oportunidades y un tiempo máximo de espera entre cada una de 20 segundos. Si pasado este tiempo o las oportunidades no lo hace, el circuito terminará la transmisión y colgara. Vuelva a llamarlo y pruebe de nuevo.

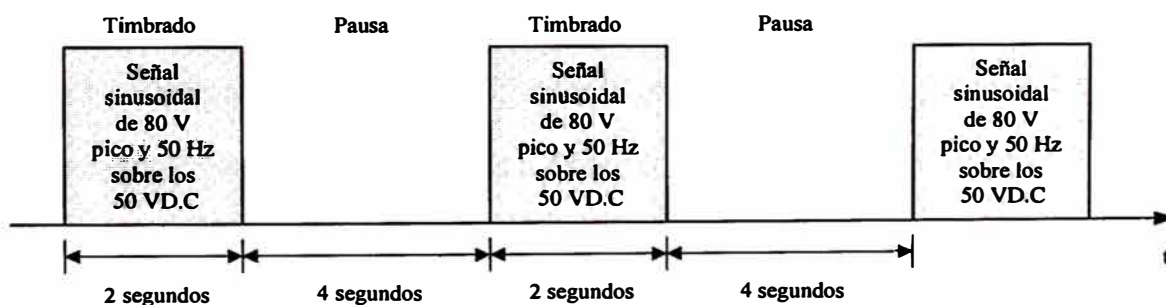


Figura 6.6 Para que timbre el teléfono, se superpone una tensión alterna de 80 V pico de 50 Hz a los 50 V D.C, tiene un periodo de 6 segundos.

Si introdujo correctamente el código podrá realizar el mando a distancia e inmediatamente le informará el estado de la entrada y de las salidas de mando.

Entrada de contacto: El circuito incorpora una entrada para detectar el funcionamiento o estado de otros dispositivos: acceso, motores, iluminación, etc. Esta entrada detecta el cierre de contactos, en ningún caso tensión. Esquemáticamente podría describirse como un interruptor que es reconocido cuando esta abierto y cuando esta cerrado. Por ello, solo podrá conectar dispositivos como, interruptor, relé o dispositivo análogo al borne de entrada. El sistema le informa si está conectada o desconectada.

Salidas de mando: Las 3 salidas de mando son relés, lo cual permite la conexión de cualquier aparato siempre y cuando su consumo no supere los 10A. El circuito de protección de salida de mando (ver figura 6.10), es el que se describe a continuación. El circuito de protección incluido en la tarjeta recibe la señal con un varistor, cuya función es de proteger contra las sobretensiones restantes que pudieran afectar a los dispositivos de la tarjeta, luego pasa por bobinas de choques de 15uH los cuales detendrán los transitorios y por último la señal pasa por un filtro RC el cual sirve para que la señal de control no afecte al dispositivo, en este caso los contactos del relé. El microcontrolador PIC16C55 dará la orden al relé en un proceso determinado y este a su vez conecta o desconecta a cualquiera de las tres salidas de mando (en una de las cuales se encuentra el circuito de Control de Nivel).

Para proteger al relé se usa una resistencia en serie con un condensador ($R=39\ \Omega$ y $C= 0.01\mu\text{F}/ 1\text{KV}$) actúan reduciendo la chispa producido cuando se cierran los

contactos del relé. Cuando se controla regular corriente el cierre de los contactos del relé origina chispas eléctricas y con el tiempo los contactos se deterioran.

Código de acceso: Necesita su código de acceso propio para poder acceder a sus funciones cuando se establece la comunicación telefónica. El código de acceso es configurado por usuario en el 8-Dip-switch (SW2) fijando unidades y decenas (ver tabla 6.1).

8-Dip-switch		Valor decimal, tanto para la unidad o decena
Decenas	Unidades	
ON ○ ○ ○ ○ ○ ○ ○ ○	○ ○ ○ ○ 1 ○ ○ ○ ○	
○ ○ ○ ○ ○ ○ ○ ○	○ ○ ○ ○ ○ ○ ○ ○	0
○ ○ ○ ● ○ ○ ○ ●	○ ○ ○ ● ○ ○ ○ ●	1
○ ○ ● ○ ○ ○ ● ○	○ ○ ● ○ ○ ○ ● ○	2
● ○ ○ ○ ● ○ ○ ○	● ○ ○ ○ ● ○ ○ ○	8
● ○ ○ ● ● ○ ○ ●	● ○ ○ ● ● ○ ○ ●	9

Tabla 6.1 *Configurando el código de acceso en el 8-Dip-switch, donde el puente respectivo significa posición del dip-switch en ON.*

En la figura 6.7 se observa el esquema electrónico del circuito de Telemando, destacándose el circuito de protección de línea telefónica.

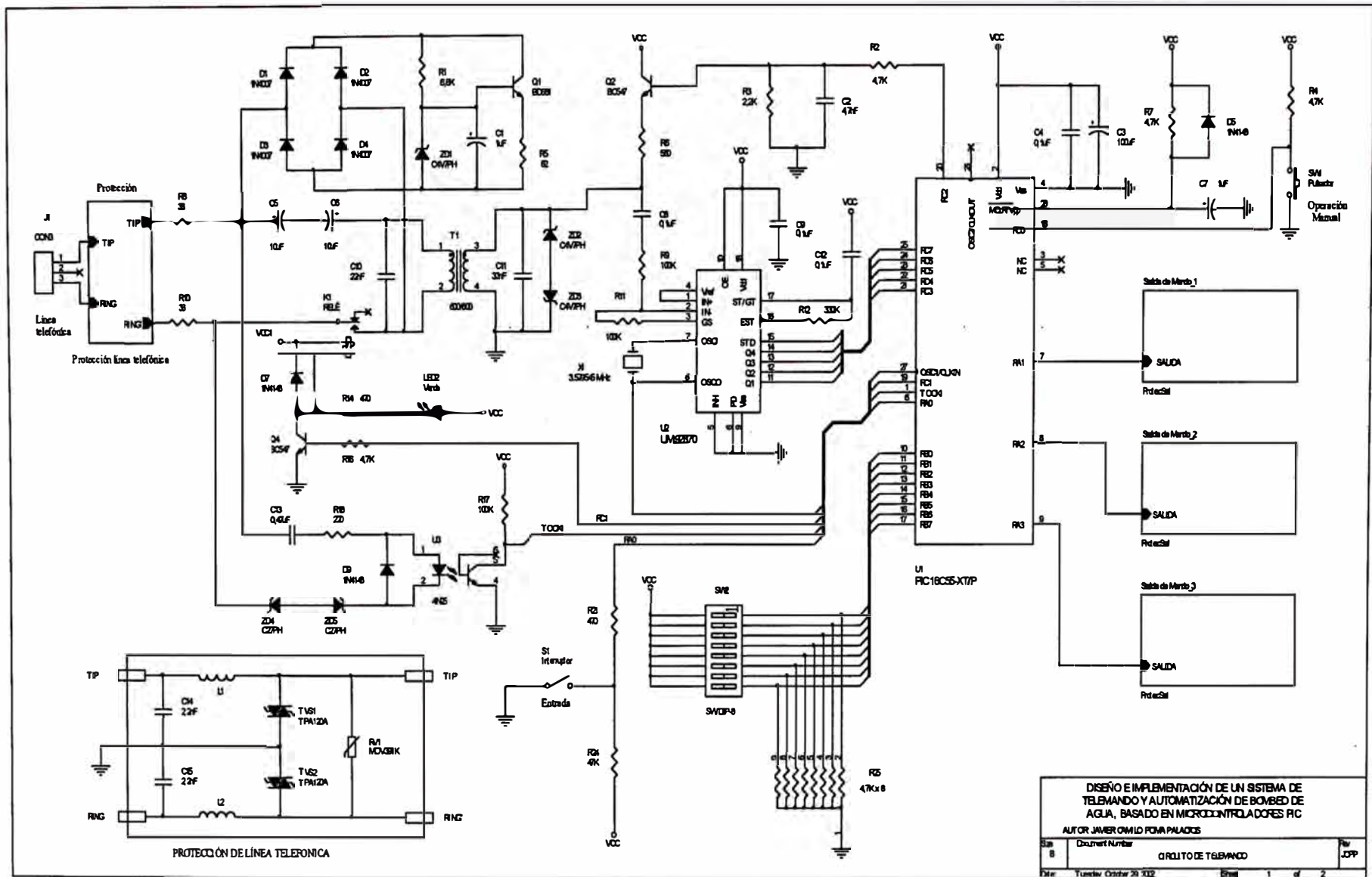


Figura 6.7 Esquema electrónico del circuito de Telemando.

6.1.3. Circuito de Control de Nivel

Cuando se conecte el circuito de Control de Nivel, en operación manual o vía mando telefónico; el microcontrolador PIC16F876 configura puertos, inicializa los GPR, el módulo LCD y los periféricos, lee los datos de configuración de las variables desde la memoria de datos EEPROM, configura interrupciones y lee el estado de los pulsadores.

Si los tres primeros pulsadores se mantienen pulsados (o posteriormente después de realizado un Reset), se ingresa al MENU de configuración donde se ajustan los valores de las variables.

En funcionamiento normal el PIC16F876 monitorea el nivel agua (amplificado desde el transductor de presión MPX2100 con el amplificador de instrumentación AD623 y convertido a digital con el módulo convertidor A/D del PIC16F876), muestra su valor en el formato de la unidad de medida elegida en la pantalla LCD, señalando el estado de las electrobombas, y realiza el testeo de los pulsadores. El módulo convertidor A/D del PIC16F876 muestrea los valores de presión para ser procesados digitalmente y convertidos a su equivalente de nivel de agua.

La figura 6.8 muestra el diagrama de bloques del circuito de Control de Nivel.

A. Voltaje analógico de entrada al módulo convertidor A/D

Al encontrarse el transductor de presión MPX2100DP (alimentado con 12 V D.C) conectado por una delgada manguera en el lado de presión (P1) hasta el nivel tope mínimo en que se debe encontrarse el nivel de agua; la salida de voltaje diferencial D.C es: $V_{out}^+ - V_{out}^- = V_{off} + S * P$, siendo $S = 0,4 \text{ mV/KPa}$; este voltaje diferencial analógico ingresa a los pines +IN y -IN del amplificador de instrumentación AD623

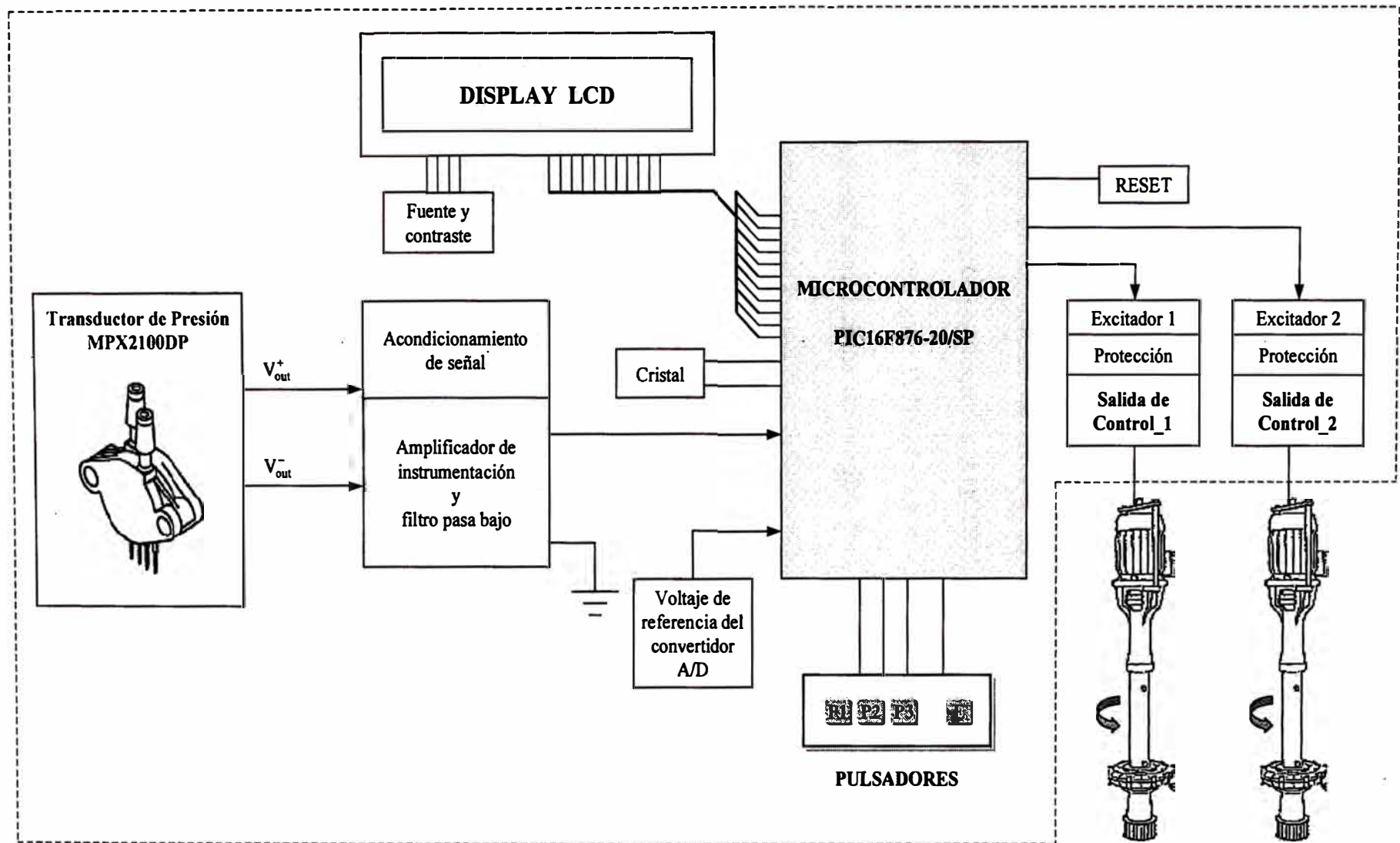


Figura 6.8 Diagrama de bloques del circuito de Control de Nivel, que muestra las electrobombas.

(alimentado con 12 V D.C), siendo amplificado con ganancia de aproximadamente 123, utilizando el potenciómetro de precisión R5. El voltaje de referencia (pin 5) es fijado con el potenciómetro R6 en 500 mV. El voltaje de salida V_o en el pin 6 del AD623, es filtrado utilizando un filtro pasa bajo RC. Siendo $R= 150\text{ K}\Omega$ y $C= 0,1\mu\text{F}$ se tiene una frecuencia de corte para el filtro $f_c= 10,6\text{ Hz}$. El voltaje que sale del filtro ingresa al convertidor A/D del PIC16F876 a través del pin 7 para ser procesado y convertido a su equivalente de nivel de agua. El convertidor A/D de 10 bits es configurado por software (registros ADCON1, ADCON2), tiene como $V_{ref} = 5\text{ V}$, entonces la Resolución será: $5\text{V}/1024= 4,8\text{ mV}$.

Los valores muestreados de presión para ser convertidos a su equivalente de nivel de agua utilizan la siguiente equivalencia: 1 cm de $\text{H}_2\text{O}=0,09806\text{ KPa}$.

Unidad	cm de H_2O	mm Hg	PSI
1 KPa	10,1973	7,5006	0,14504

Tabla 6.2 *Un Kilo Pascal (1KPa) y sus constantes de conversión.*

Así por ejemplo calculemos el equivalente a la salida del convertidor (P_D) de 10 cm de H_2O ; considerando que el $V_{off\ final} = 0$ voltios, debido a que por software se logra esta condición. Entonces $10\text{ cm}=(0,9806\text{ KPa}) (0,4\text{ mV/KPa}) (122,5)= 48,0494\text{ mV}$, entonces $V_o=48,0494\text{ mV}$ y al dividir este valor por la resolución tenemos 10,010291 cuyo valor equivalente en hexadecimal es 000Ah. De la misma forma fueron calculados los valores de salida hexadecimal que se muestra en la tabla 6.3.

B. Interface del usuario

Las interfaces del usuario consisten del módulo LCD, cuatro pulsadores de control, un pulsador de Reset, y dos led's de ON de las electrobombas.

El módulo LCD es controlado por el PIC16F876. Módulo de dos líneas de 20 caracteres alfanuméricos de 5 x 7 pixeles.

Cuando cualquiera de los cuatro pulsadores es pulsado el respectivo pin del Puerto pasa a nivel bajo. Los cuatro pulsadores son conectados al PORTB <7-4> y son llamados de izquierda a derecha P1, P2, P3 y E.

Nivel de agua en cm	Valor en KPa	Valor en mV	Salida del convertidor A/D	
			Decimal	Hexadecimal
1	0,09806	4,8049	1	0001
10	0,98060	48,0494	10	000A
25	2,45150	120,1235	25	0019
50	4,90300	240,2470	50	0032
75	7,35450	360,3705	75	004B
100	9,80600	480,4940	100	0064
125	12,25750	600,6175	125	007D
150	14,70900	720,7410	150	0096
175	17,16050	840,8645	175	00AF
200	19,61200	960,9880	200	00C8
250	24,51500	1201,2350	250	00FA
300	29,41800	1441,4820	300	012C
350	34,32100	1681,7290	350	015E
400	39,22400	1921,9760	400	0190
450	44,12700	2162,2230	450	01C2
500	49,03000	2402,4700	500	01F4
550	53,93300	2642,7170	550	0226
600	58,83600	2882,9640	600	0258
650	63,73900	3123,2110	650	028A
700	68,64200	3363,4580	700	02BC
750	73,54500	3603,7050	750	2EE
800	78,44800	3843,9520	800	0320
850	83,35100	4084,1990	850	0352
900	88,25400	4324,4460	900	0384

Tabla 6.3 *Valores de nivel de agua en diferentes equivalencias.*

Las figuras 6.9 y 6.10 muestra el esquema electrónico completo del circuito de Control de Nivel.

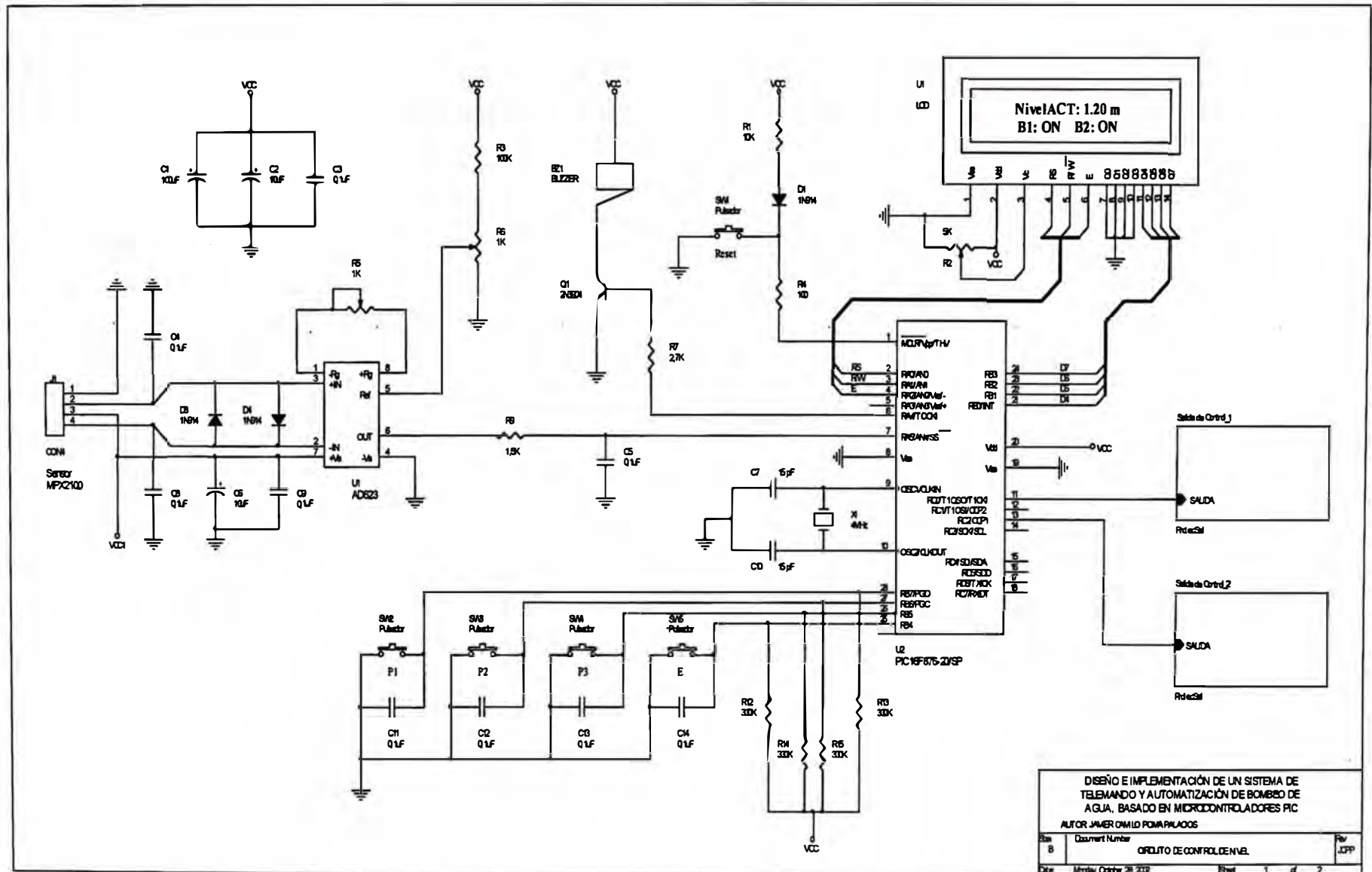


Figura 6.9 Esquema electrónico del circuito de Control de Nivel.

DISÑO E IMPLEMENTACIÓN DE UN SISTEMA DE
TELEMANDO Y AUTOMATIZACIÓN DE BOMBEO DE
AGUA, BASADO EN MICROCONTROLADORES PIC
AUTOR JAMER OMILIO ROMARILLOS

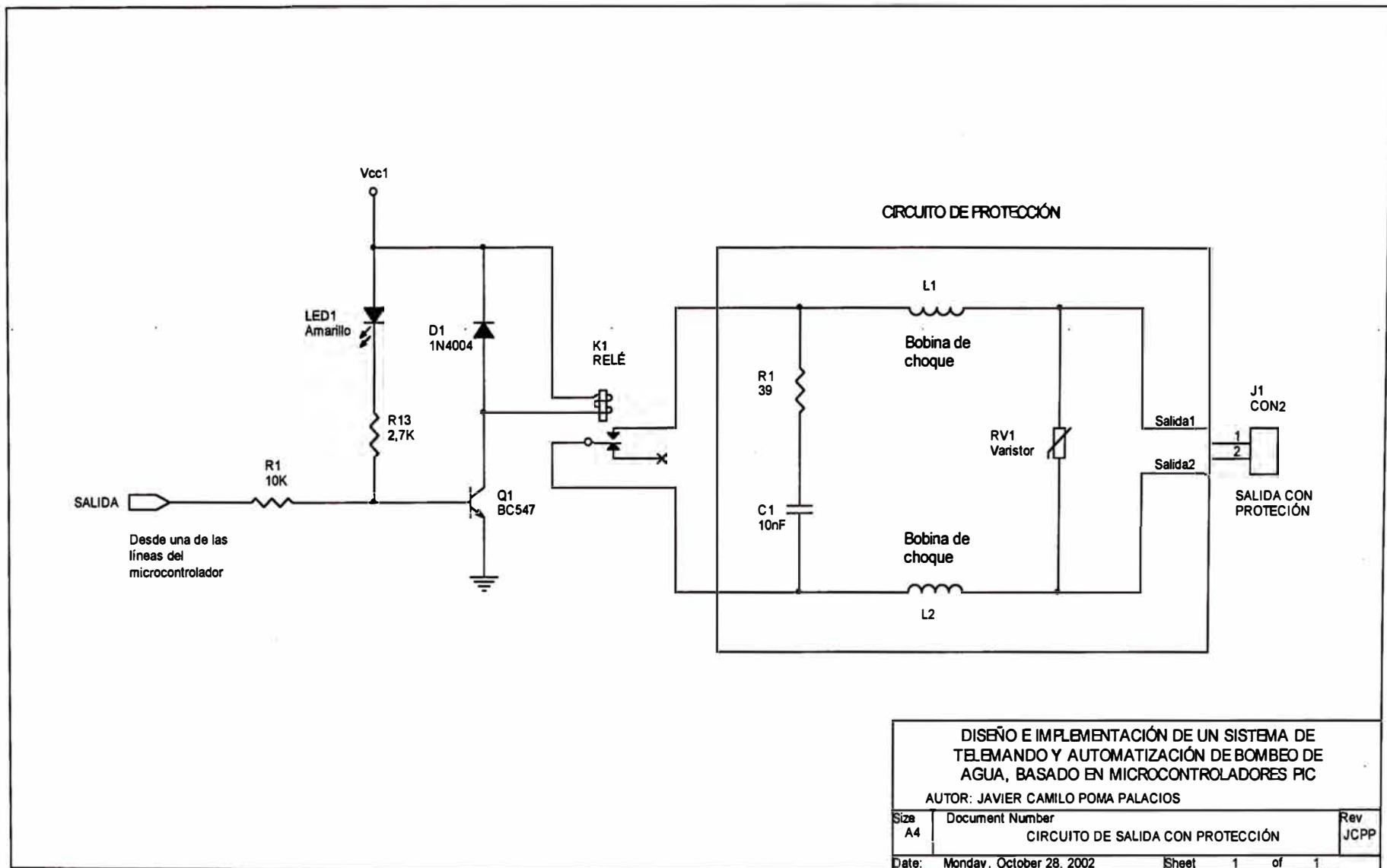


Figura 6.10 Esquema electrónico del circuito de salida de mando o salida de control.

C. Salidas de Control

El esquema de cualquiera de las salidas de control se muestra en la figura 6.10. La Salida de Control_2 es conectado directamente; mientras que en la Salida de Control_1 se añade el circuito de protección de entrada de 220 V A.C, debida a la distancia de 1 Km (en este caso en ambos extremos).

El circuito de protección de salida, es el que se describe a continuación. Luego de ser atenuado las sobre tensiones transitorias, el circuito de protección incluido en la tarjeta recibe la señal con un varistor, cuya función es de proteger contra las sobretensiones restantes que pudieran afectar a los dispositivos de la tarjeta, luego pasa por bobinas de choques de 15uH los cuales detendrán los transitorios y por ultimo la señal pasa por un filtro RC el cual sirve para que la señal que controla no afecte al dispositivo en este caso los contactos del relé (reduciendo las chispas ocasionados al cerrar el contacto). El microcontrolador PIC16F876 dará la orden al relé en un determinado proceso y este a su vez al actuador respectivo, para controlar el funcionamiento de la electrobomba utilizando la señal de 220 V A.C.

Automatizando el sistema de bombeo de agua

El circuito de Control de Nivel se encarga de automatizar el sistema de bombeo de agua, controlando el nivel mínimo y nivel máximo del reservorio N° 2. El diseño del sistema que automatiza el bombeo de agua se realizó seleccionando un sistema de **control en lazo cerrado** (ver figura 6.11), el cual se caracteriza por que la realimentación se realiza a través del transductor de presión MPX2100 desde el proceso hacia el **controlador digital ON-OFF**, permitiendo conocer si las acciones ordenadas a los actuadores se han realizado correctamente sobre el proceso

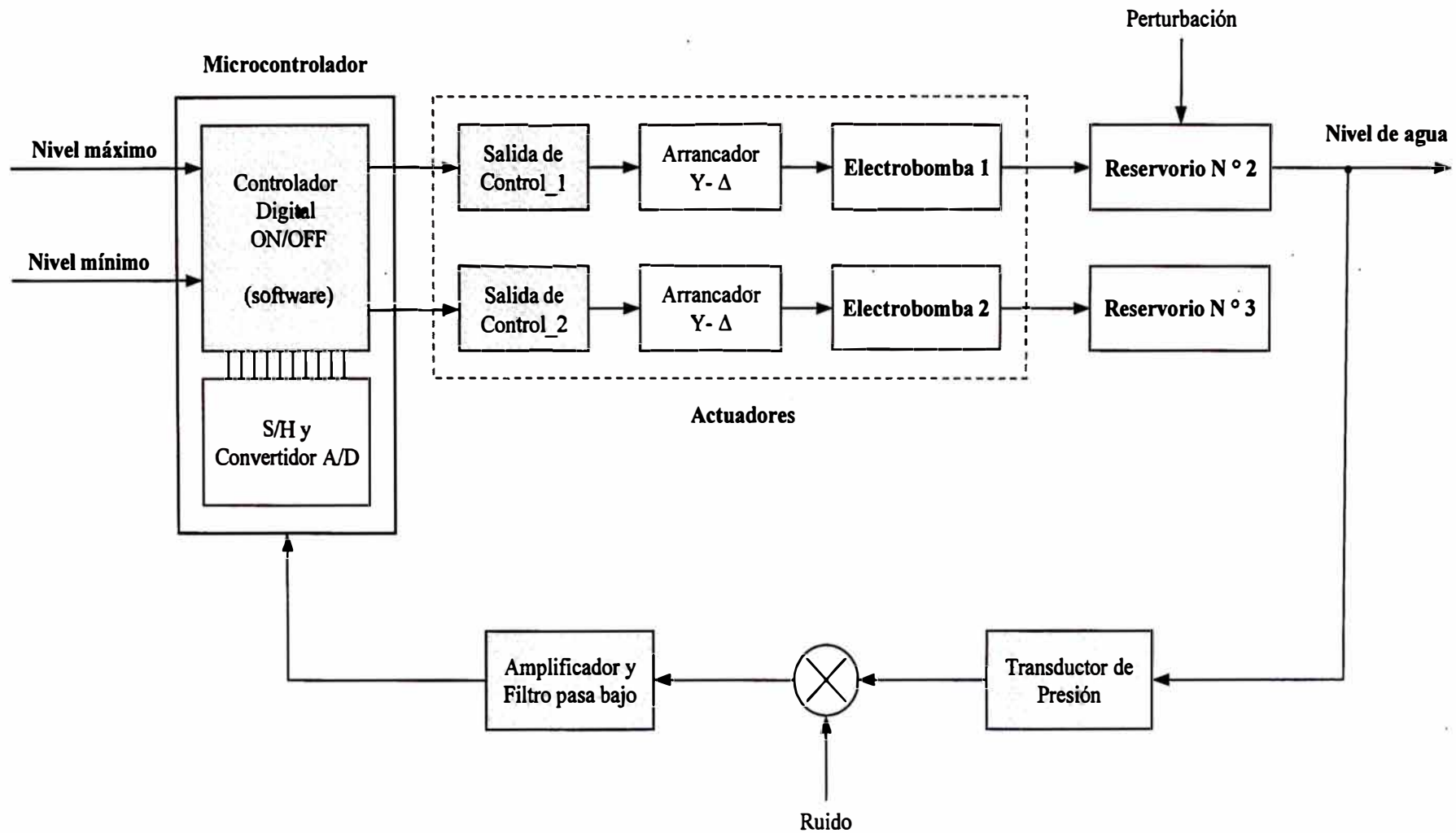


Figura 6.11 Diagrama de bloques de control en lazo cerrado para el Control de Nivel.

(electrobombas). La variable nivel de agua que se controla necesita un control continuo en función de los valores de referencia nivel máximo y nivel mínimo de entrada. En el control ON-OFF la variable dependiente es el estado del corrector final (electrobomba encendida o apagada), y la variable independiente es la señal de error en la comparación realizada con los valores NivelMAX o NivelMIN.

Control ON-OFF digital: En el control (ON-OFF), el corrector final sólo tiene dos estados de operación. Según la lógica de control implementado en software, el controlador envía al corrector final a un estado u otro y al ser actuada una electrobomba, se enciende o se apaga.

Brecha diferencial: Ningún controlador de ON-OFF puede presentar el comportamiento ideal de pasar instantáneamente a encendido o apagado. Todos los controladores de ON-OFF tienen una brecha diferencial. La brecha diferencial en un controlador ON-OFF se define como el rango menor de valores que debe atravesar el valor medido para hacer que el dispositivo corrector pase de un estado al otro. La brecha diferencial está definida específicamente para el control ON-OFF.

Tarjetas de circuito impreso

Las tarjetas de circuito impreso fueron diseñadas utilizando el módulo *OrCAD Layout* de *OrCAD Release 9.1*. Las tarjetas de circuito impreso del circuito de la Fuente de Alimentación, circuito de Telemando y circuito de Control de Nivel, que muestra la cara de componentes y la cara de soldadura se encuentran en las figuras 6.12, 6.13 y 6.14 respectivamente.

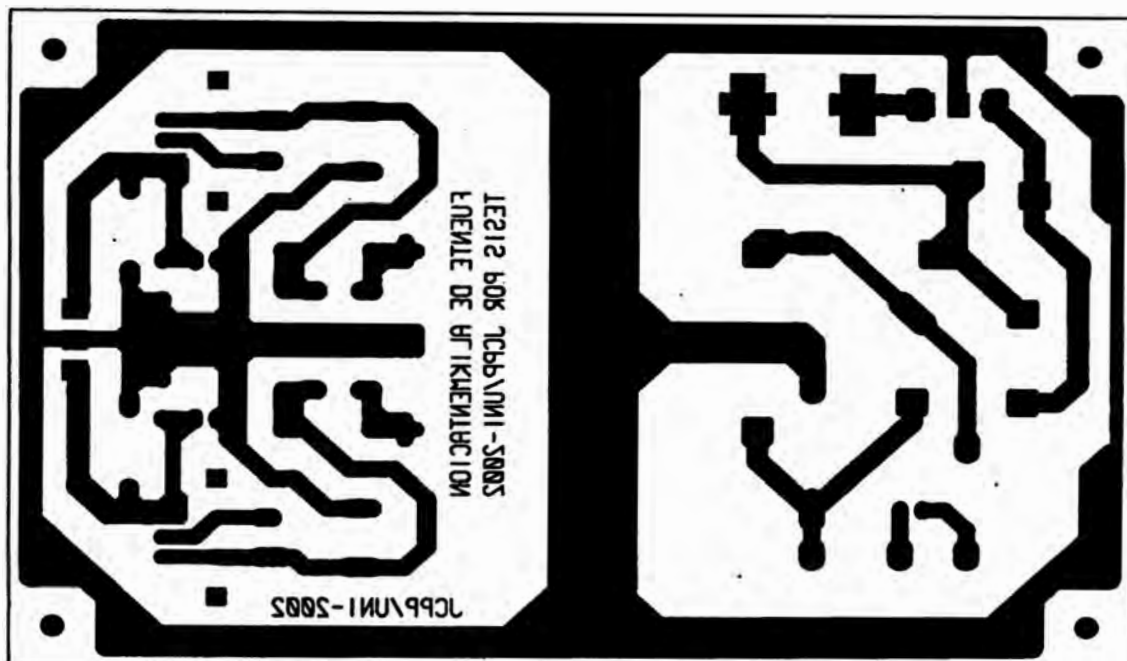
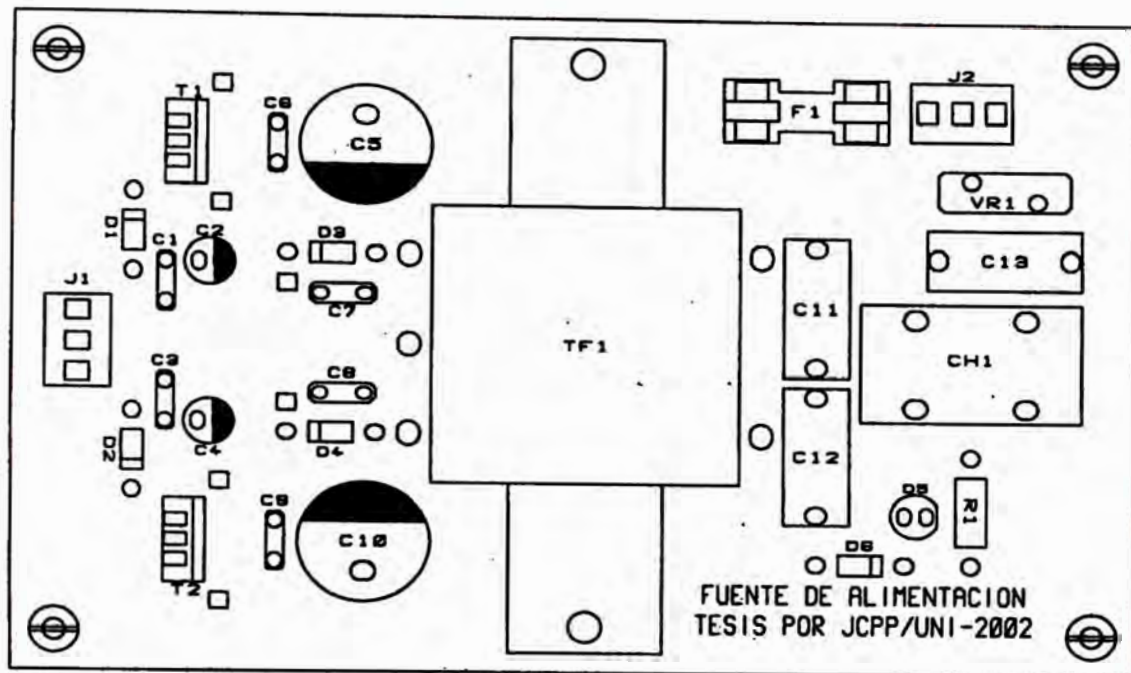


Figura 6.12 Cara de componentes y cara de soldadura que corresponde al circuito de la Fuente de Alimentación.

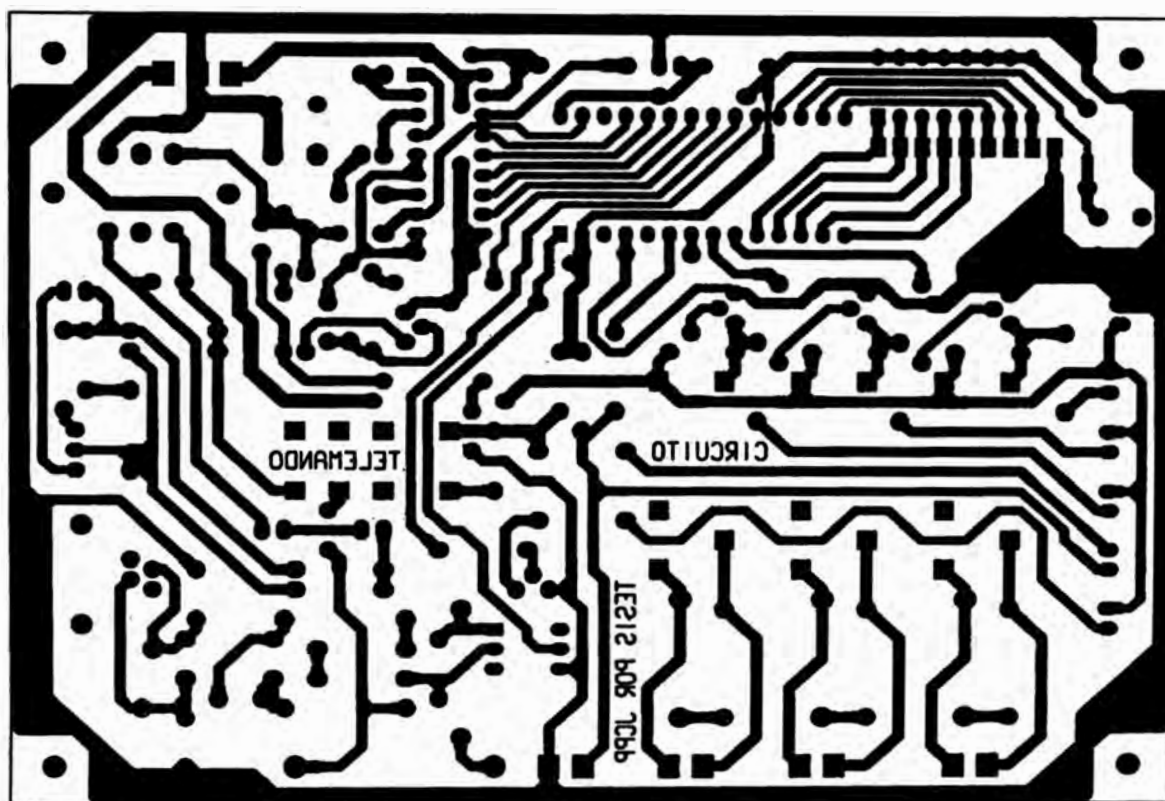
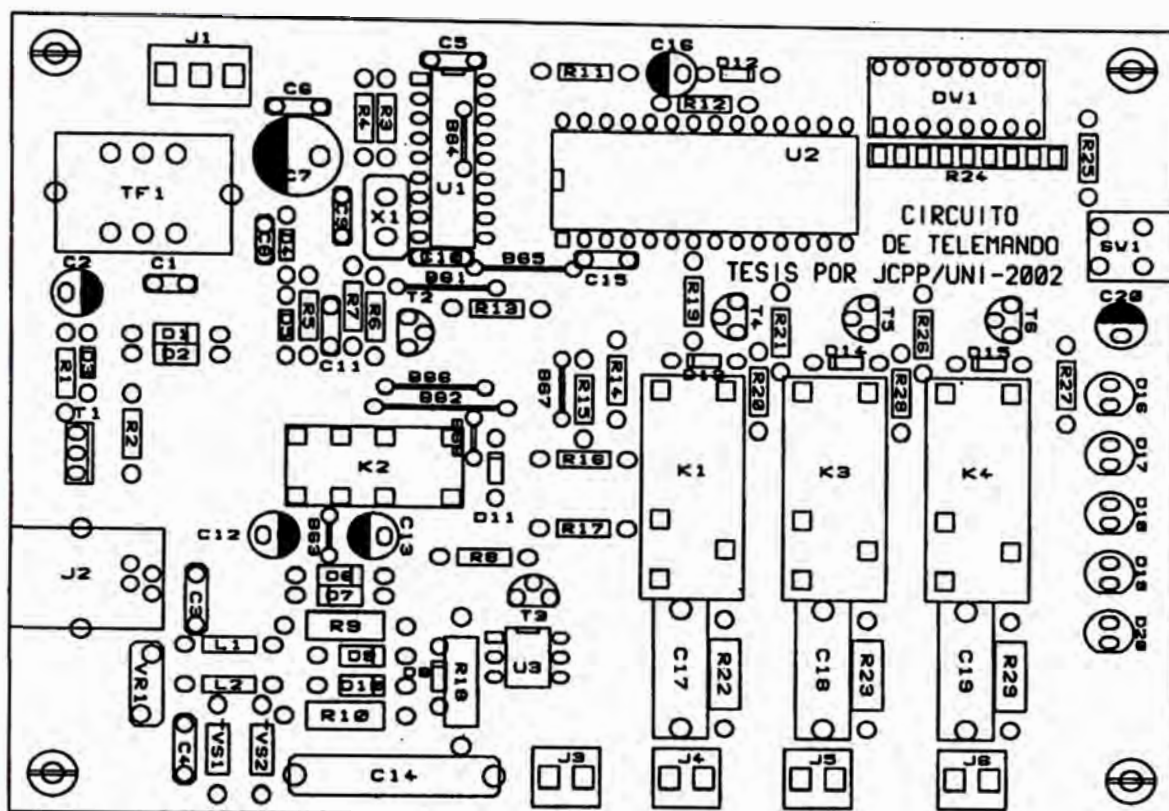


Figura 6.13 Cara de componentes y cara de soldadura que corresponde al circuito de Telemando.

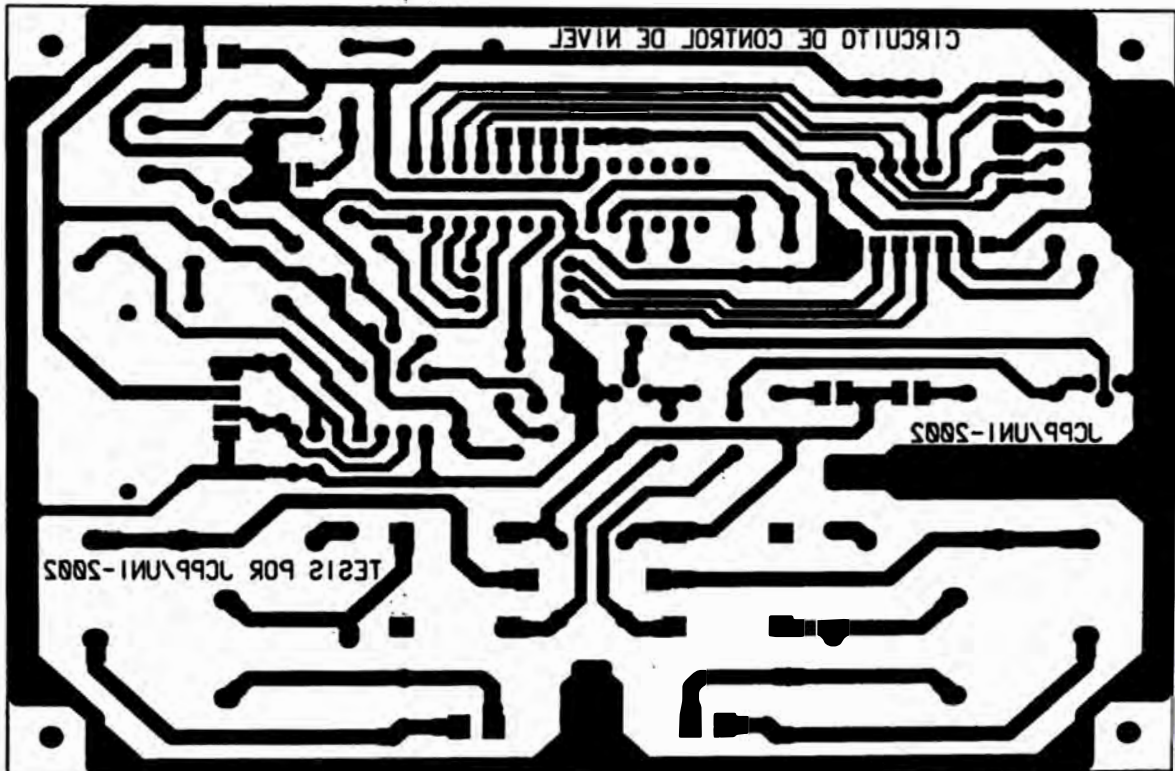
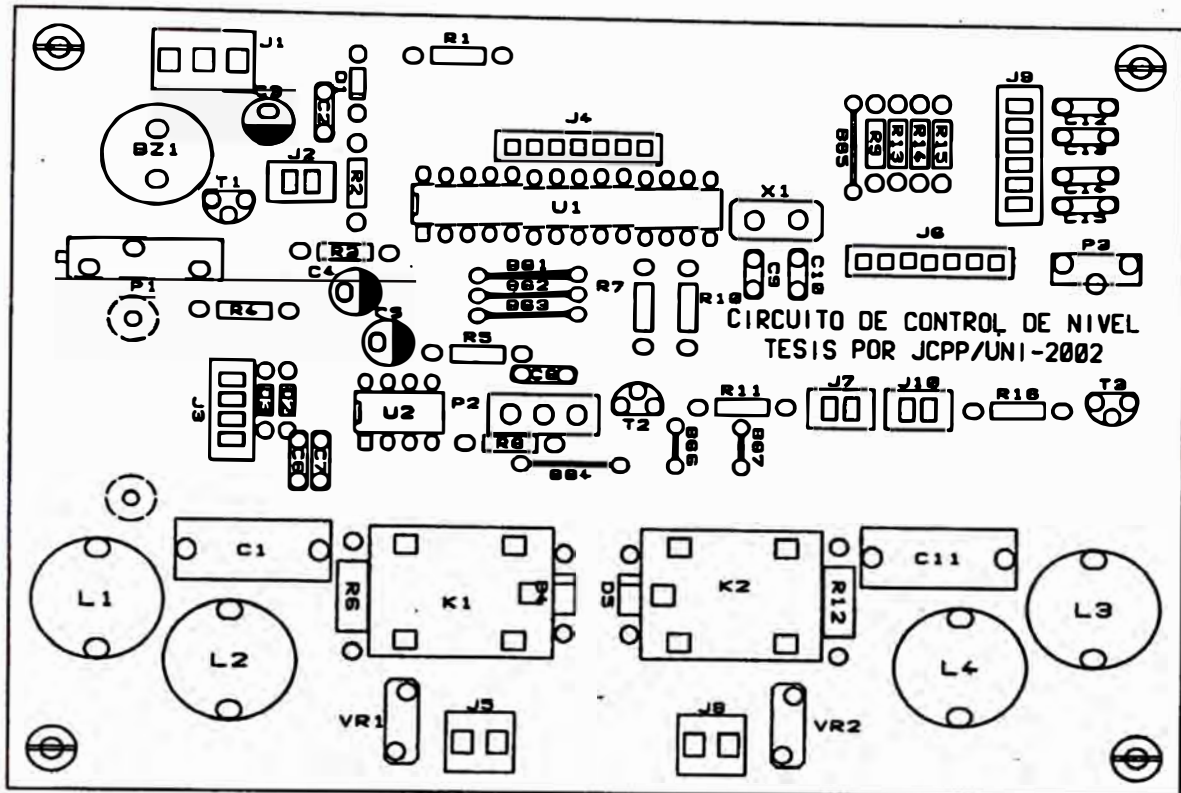


Figura 6.14 Cara de componentes y cara de soldadura que corresponde al circuito de Control de Nivel.

6.3. Software del sistema

Los programas que utiliza el **Sistema de Telemando y Automatización de Bombeo de Agua, Basado en Microcontroladores PIC** son: **teleman.asm** y **ctnivel.asm**. Los cuales una vez ensamblados, simulados, depurados y finalmente ensamblados genera los archivos de extensión **.hex**: **teleman.hex** y **ctnivel.hex** que se utilizan para programar el microcontrolador PIC16C55 y el microcontrolador PIC16F876 respectivamente. La edición de los programas, el ensamblado, depuración y simulación se realizaron usando las herramientas de MPLAB-IDE versión 5.00.00.

6.3.1. Para el PIC16C55

El programa en ensamblador **teleman.asm** realiza las siguientes tareas: Una vez hecha la llamada al número telefónico, en el que se encuentra conectado el sistema; el programa permite establecer la conexión con la línea telefónica luego de tres u ocho timbradas. Detectado el número de timbradas, el sistema contesta automáticamente y luego de unos segundos informa el estado en que se encuentran: la Entrada, Salida de Mando_1, Salida de Mando_2 y Salida de Mando_3; en ese orden. Un sonido de corta duración, señala desconectado y dos sonidos de corta duración señala conectado.

Establecida la comunicación el programa controla el ingreso del código de seguridad y el respectivo control de las tres salidas de mando (relés). Así mismo se señala la cancelación de la comunicación con un sonido de larga duración, luego de tres intentos de ingreso de código o al no pulsar ninguna tecla en 20 segundos.

Las figuras 6.15 y 6.16 muestran el diagrama de flujo del programa **teleman.asm** y la figura 6.17 muestra el diagrama de flujo para operación manual.

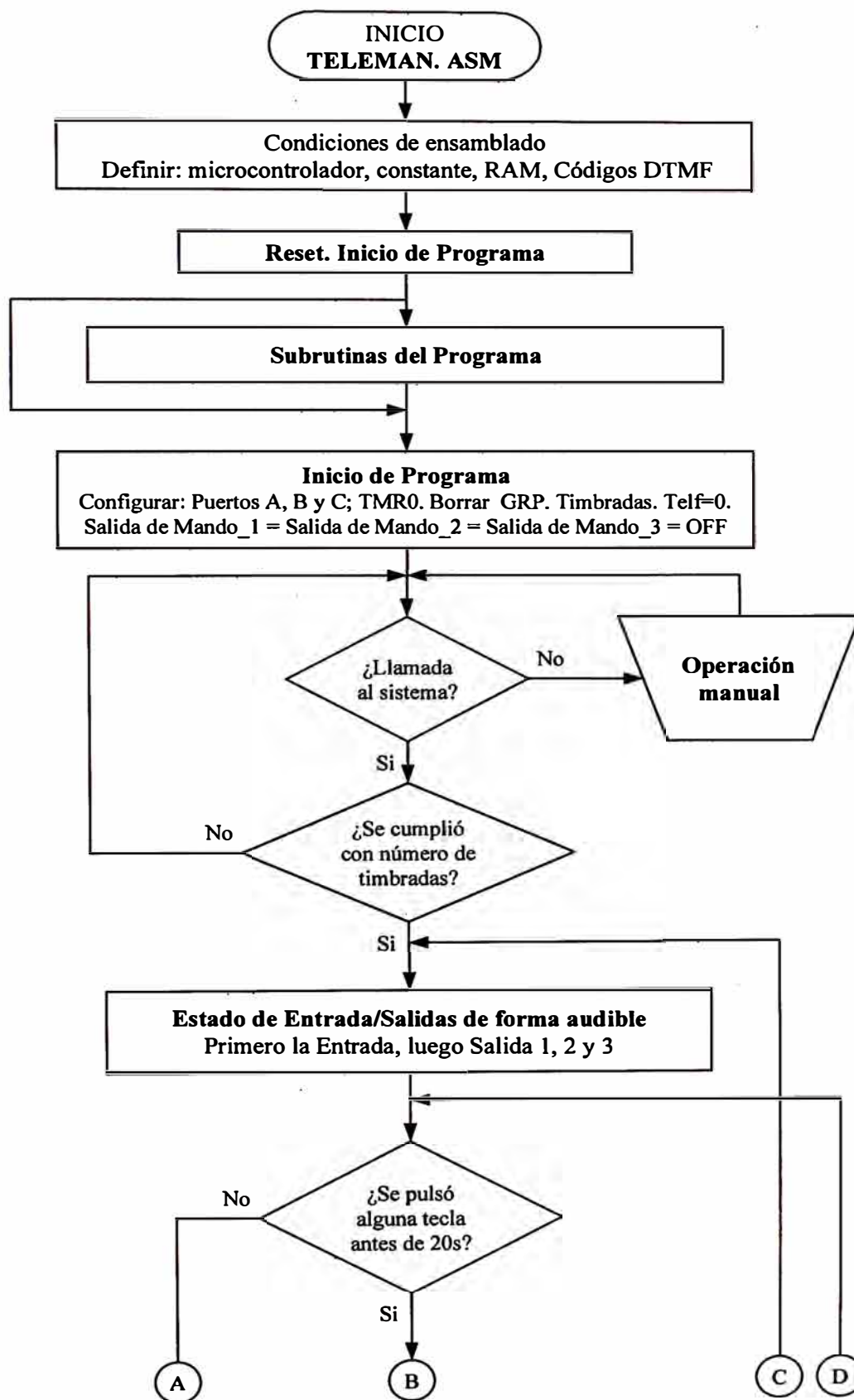


Figura 6.15 Diagrama de flujo del programa *teleman.asm*, que es utilizado por el circuito de Telemando.

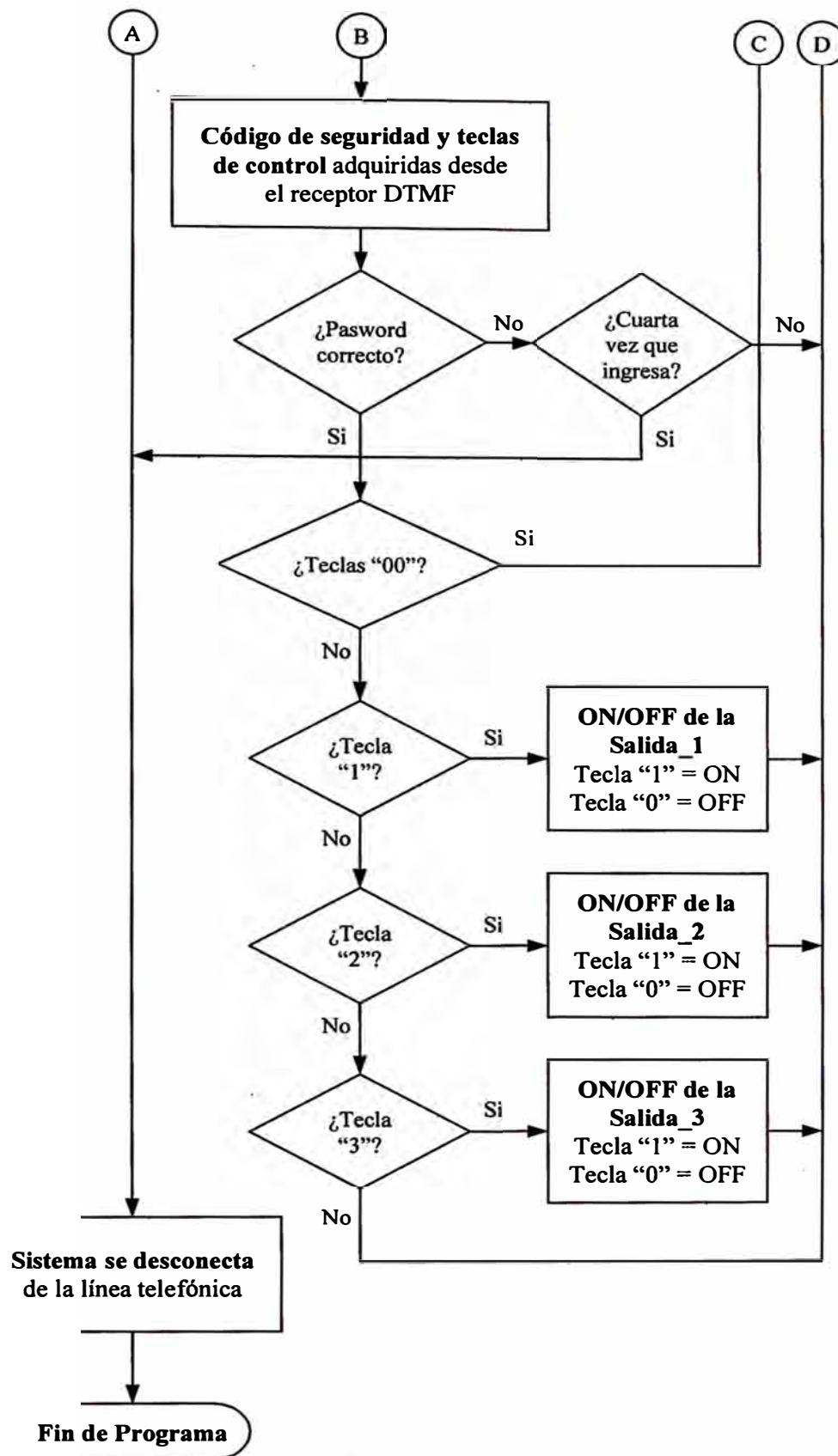


Figura 6.16 Diagrama de flujo del programa *teleman.asm*, que es utilizado por el circuito de *Telemando* (continuación).

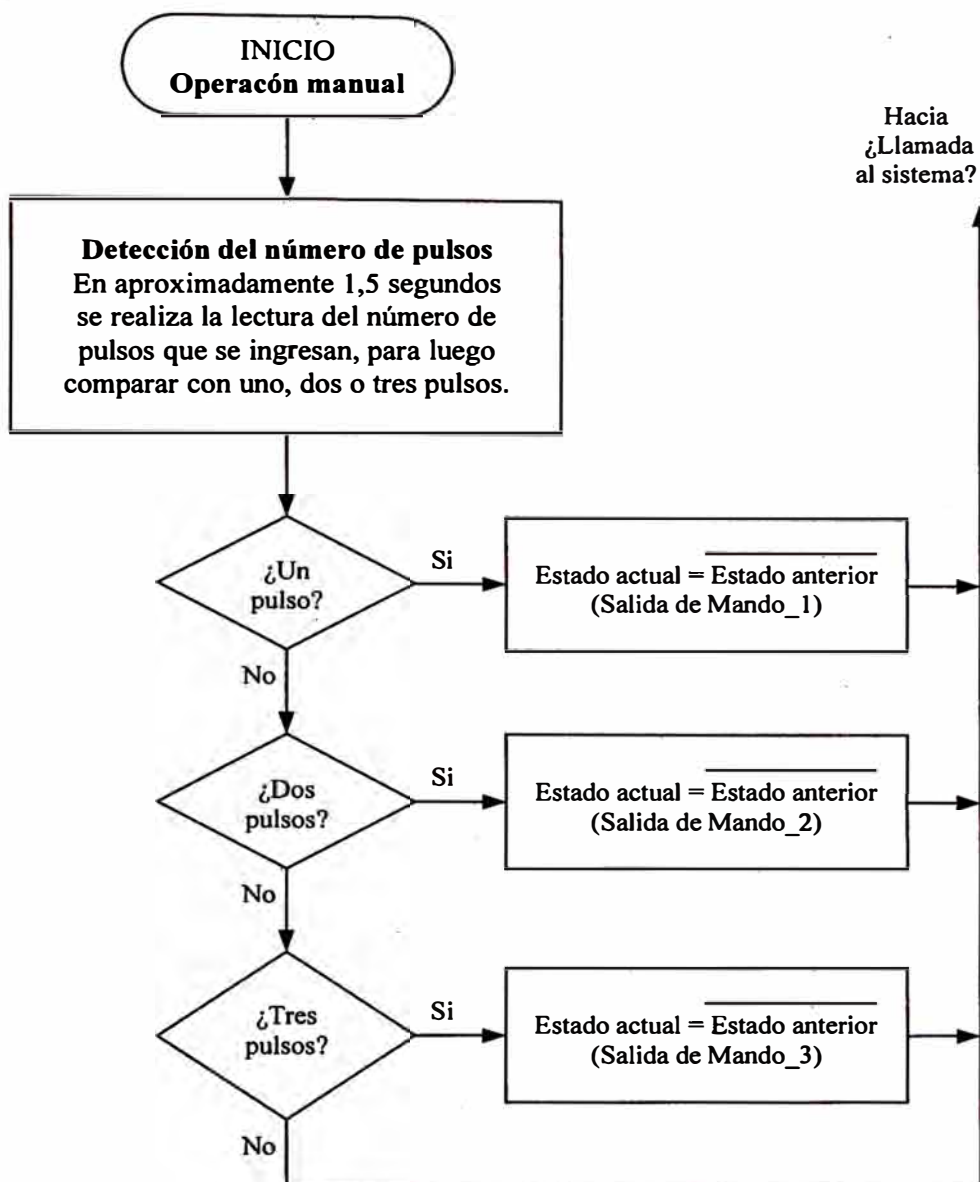


Figura 6.17 Diagrama de flujo para la operación manual en el programa *teleman.asm*.

Los tres comandos utilizados para realizar el control a distancia vía teléfono son:

1. Información del estado de entrada, Relé1, Relé2, Relé3 CODIGO "00"
2. Cambiar relé a ON CODIGO "<1-3>1"
3. Cambiar relé a OFF CODIGO "<1-3>0"

Para concluir se muestra el listado del programa, editado en lenguaje ensamblador para el microcontrolador PIC16C55 de Microchip.

```

;*****
; Listado del código de Programa: teleman.asm
; Ensamblado usando MPLAB-IDE 5.00.00

    TITLE    "PROGRAMA PARA EL CIRCUITO DE TELEMANDO"

LIST    ST=ON    ;Suprime listado de símbolos
LIST    N=0      ;Suprime listado de saltos de página

;=====
;
;                UNIVERSIDAD NACIONAL DE INGENIERÍA
;
; Título de la tesis:
;
;    "DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE TELEMANDO
;      Y AUTOMATIZACIÓN DE BOMBEO DE AGUA, BASADO EN
;      MICROCONTROLADORES PIC"
;
; Autor: JAVIER CAMILO POMA PALACIOS
;
; Lima, Julio del 2002
;=====

LIST    P=16C55
RADIX   HEX

    #INCLUDE    <P16C5X.INC>

; -----
; CONSTANTE
; =====

INTENTOS    EQU    .3        ; Máximo número de intentos de
                                ; ingreso de código de seguridad

; -----
; DEFINICIÓN DE ETIQUETAS
; =====
; Definición de Registros de Propósito General desde 0x08.

    CBLOCK 0x08
DIGITO
DIGITEMP
DIGITOS_COD
MODO
CODIGO        ; Código de seguridad
PULSOS
TIMBRADAS

```

```

NUM_TIMBRADAS      ; Número de timbradas antes de contestar
NUM_INTENTOS       ; Número de intentos de ingreso de código

```

```

TEMP1
TEMP2
CONTA
CONTA1
CONTA2
    ENDC

```

```

; -----
; CÓDIGOS DTMF
; =====

```

```

#DEFINE UNO        .1      ; Dígito 1
#DEFINE DOS        .2      ; Dígito 2
#DEFINE TRES       .3      ; Dígito 3
#DEFINE CUATRO     .4      ; Dígito 4
#DEFINE CINCO      .5      ; Dígito 5
#DEFINE SEIS       .6      ; Dígito 6
#DEFINE SIETE      .7      ; Dígito 7
#DEFINE OCHO       .8      ; Dígito 8
#DEFINE NUEVE      .9      ; dígito 9
#DEFINE CERO       .10     ; Dígito 0
#DEFINE ASTE       .11     ; Dígito *
#DEFINE MICHI      .12     ; Dígito #

```

```

#DEFINE ON.OFF_TELF PORTC,1; Activa o desactiva comunicación
                          ; con el circuito de Telemando
#DEFINE NIVEL_STD   PORTC,7; Indica código DTMF válido

```

```

; -----
; VECTOR DE RESET
; =====

```

```

    ORG    0x01FF      ; Vector de Reset

```

```

    goto  INICIO
    ORG    0x0000      ; Dirección de inicio en la
                          ; memoria de programa

```

```

; -----
; Subrutina TONO_DECODIF
; =====

```

```

; Se encarga de obtener el valor decodificado del DTMF en el
; registro DIGITEMP.

```

```

TONO_DECODIF
    btfss NIVEL_STD
    goto  $-1
    movf  PORTC,W
    andlw B'01111000'
    movwf DIGITEMP
    bcf   STATUS,C
    rlf   DIGITEMP,F

```

```

    swapf  DIGITEMP, F
    btfsc  NIVEL_STD
    goto   $-1
    retlw  0

; -----
; Subrutinas de RETARDO
; =====
; Retardo variable múltiplo de microsegundo o milisegundo.
; TEMP1 contiene el número de microsegundos o milisegundos.

RETARDO_XXXUS                                ; RETARDO_XXXUS= TEMP1*3 + 1)*Tins
    decfsz TEMP1, F                            ; TEMP1=165, RETARDOL_V=500us
    goto   $-1
    retlw  0

RETARDO_XXXMS                                ; TEMP1=162
    movwf  TEMP1                                ; RETARDO_XXXMS=124,75ms
    clrf  TEMP2
    decfsz TEMP2, F
    goto   $-1
    decfsz TEMP1, F
    goto   $-3
    retlw  0

RETARDO_75MS
    movlw  .150
    movwf  TEMP2
    movlw  .165
    decfsz TEMP1, F
    goto   $-1
    decfsz TEMP2, F
    goto   $-4
    retlw  0

; -----
; Subrutina SONIDO
; =====
; Genera un tono de 800Hz de una duración aproximada de 75ms.

SONIDO
    movlw  .60
    movwf  TEMP2
    bsf    PORTC, 2
    movlw  .208
    movwf  TEMP1

    call   RETARDO_XXXUS
    bcf    PORTC, 2
    movlw  .208
    movwf  TEMP1

    call   RETARDO_XXXUS
    decfsz TEMP2, F
    goto   $-9
    retlw  0

```

```

; -----
; Subrutina BEEP_LARG
; =====
; Genera un tono de 800Hz por aproximadamente 375ms.

BEEP_LARG
    call    SONIDO
    call    SONIDO
    call    SONIDO
    call    SONIDO
    call    SONIDO
    retlw   0

; -----
; INICIO DEL PROGRAMA
; =====

INICIO
    movlw   B'00000001'
    TRIS    PORTA
    bcf     PORTA,1
    bcf     PORTA,2
    bcf     PORTA,3
    movlw   B'11111111'
    TRIS    PORTB
    movlw   B'11111001'
    TRIS    PORTC
    bcf     ON.OFF_TELF      ; Relé de comunicación en OFF
    bcf     PORTC,2
    movlw   B'00101000'
    OPTION      ; Configura registro OPTION
    clrf    TMRO
    movlw   0x08          ; Inicio de los GPR
    movwf   FSR

BORRA_REG
    clrf    INDF          ; Borra GPR
    incf    FSR,F        ; Incrementa puntero
    movf    FSR,W
    xorlw   0x1F
    btfss   STATUS,Z     ; Es el último GPR a borrar?

    goto    BORRA_REG    ; No, borrar otro GPR
    movlw   .8           ; Si
    movwf   NUM_TIMBRADAS ; NUM_TIMBRADAS=8
    movf    PORTC,W      ; Lee PORTC
    andlw   B'00000001'
    xorlw   0x00
    btfss   STATUS,Z

    goto    TELEMAN
    movlw   .3
    movwf   NUM_TIMBRADAS ; NUM_TIMBRADAS=3

; Se realiza el chequeo respectivo para decidir, si se

```

```
; encuentra en Telemando via teléfono o operación
; manual.
```

```
TELEMAN
```

```
    clrf    TIMBRADAS
    clrf    PULSOS
```

```
DETEC_RING
```

```
    clrf    TMR0                ; Borra TMR0
    btfss   TMR0,0
    goto    MANUAL
    movlw   .8
    movwf   CONTA2
    movlw   .162
    call    RETARDO_XXXMS
    decfsz  CONTA2,F
    goto    $-3
    incf    TIMBRADAS,F
    movf    NUM_TIMBRADAS,W
    xorwf   TIMBRADAS,W
    btfss   STATUS,Z
    goto    DETEC_RING
    bsf     ON.OFF_TELF
    goto    CODIGO_00
```

```
; Operación manual
```

```
MANUAL
```

```
    movf    PORTC,W            ; Lee PORTC
    andlw   B'00000001'
    xorlw   0x00
    btfss   STATUS,Z
    goto    TELEMAN
    movlw   .75
    movwf   CONTA2
```

```
RETOR
```

```
    movlw   .25                ; Retardo de unos 20ms para evitar
    movwf   TEMP1              ; rebotes en el pulsador
    call    RETARDO_XXXMS
    movf    PORTC,W
    andlw   B'00000001'
    xorlw   0x01
    btfss   STATUS,Z
    goto    RETOR
    goto    INCPUL
```

```
ESTADO_0
```

```
    movlw   .25
    movwf   TEMP1
    call    RETARDO_XXXMS
    movf    PORTC,W
    andlw   B'00000001'
    xorlw   0x00
    btfss   STATUS,Z
    goto    DECCONTA2
```

```

ESTADO_1
    movlw    .25
    movwf   TEMP1
    call    RETARDO_XXXMS
    movf    PORTC,W
    andlw   B'00000001'
    xorlw   0x01
    btfsc   STATUS,Z
    goto    INCPUL

DECCONTA2
    decfsz  CONTA2,F
    goto    ESTADO_1
    goto    SALIDA_X

INCPUL
    incf    PULSOS,F
    goto    ESTADO_0

SALIDA_X
    movlw   .1
    xorwf   PULSOS,W
    btfsc   STATUS,Z
    goto    SALIDA1
    movlw   .2
    xorwf   PULSOS,W
    btfsc   STATUS,Z
    goto    SALIDA2
    movlw   .3
    xorwf   PULSOS,W
    btfsc   STATUS,Z
    goto    SALIDA3
    goto    TELEMAN

SALIDA1
    btfss   PORTA,1
    bsf     PORTA,1
    bcf     PORTA,1
    goto    TELEMAN

SALIDA2
    btfss   PORTA,2
    bsf     PORTA,2
    bcf     PORTA,2
    goto    TELEMAN

SALIDA3
    btfss   PORTA,3
    bsf     PORTA,3
    bcf     PORTA,3
    goto    TELEMAN

; -----
; Bucle de MANDO
; =====
; Permite el ingreso del código de seguridad, si no es válido

```



```

; escuchará un SONIDO de duración larga.
; Si después de tres intentos no ingresa el código correcto
; la comunicación queda cancelada. También permite el ingreso
; de los dígitos de control de los relés de mando.

```

BUCLE_MANDO

```

movlw  INTENTOS
movwf  NUM_INTENTOS    ; NUM_INTENTOS=INTENTOS
clrf   MODO
bsf    MODO,0

```

MANDO

```

movlw  .160            ; Durante 20s se encuentra
movwf  CONTA1          ; activo el Sistema de Telemando

```

R_20MS1

```

movlw  .250            ; Si no pulsa ninguna tecla
movwf  CONTA2          ; durante este tiempo se
                        ; cancela la comunicación

```

R_20MS2

```

movlw  .165
call   RETARDO_XXXUS
btfsc  ON.OFF_TELF
goto   BLOQUE_DTMF
decfsz CONTA2,F
goto   R_20MS2
decfsz CONTA1,F
goto   R_20MS1
bcf    ON.OFF_TELF
movlw  .162
call   RETARDO_XXXMS
call   BEEP_LARG
goto   TELEMAN

```

BLOQUE_DTMF

```

movlw  .162
call   RETARDO_XXXMS
btfsc  MODO,0
goto   DECO_CODIGO
btfsc  MODO,1
goto   DECO_0123
goto   DIGITO_01

```

DECO_CODIGO

```

movlw  .2
movwf  DIGITOS_COD
call   TONO_DECODIF
movf   DIGITEMP,W
xorlw  CERO
btfss  STATUS,Z
goto   COD1
clrf   DIGITEMP

```

COD1

```

movf   DIGITEMP,W

```

```

movwf CODIGO
decfsz DIGITOS_COD, F
goto MANDO
swapf CODIGO, W
andlw 0xF0
movwf CODIGO
movf DIGITEMP, W
andlw 0x0F
iorwf CODIGO, F
movf PORTB, W
xorwf CODIGO, W
btfsc STATUS, Z
goto INC_MODO

; Aquí código de seguridad no válido

COD_ERROR
movlw .162
call RETARDO_XXXMS
call BEEP_LARG

decfsz NUM_INTENTOS, F
goto DECO_CODIGO ; Otro intento
bcf ON.OFF_TELF ; Cancela comunicación telefónica
goto TELEMAN

; -----
; Comandos y sonidos utilizados
; =====
; Tres comandos utilizados:
; 1. Información del estado de:
; Entrada, Rele1, Rele2, Rele3 ..... CÓDIGO "00"
; 2. Cambiar relé a ON ..... CÓDIGO "<1-3>1"
; 3. Cambiar relé a OFF ..... CÓDIGO "<1-3>0"

; Un SONIDO de larga duración, señala código no válido.
; Un SONIDO de larga duración, señala desconexión de la línea
; telefónica.
; Un SONIDO de corta duración, señala entrada o salida
; desconectada (OFF).
; Dos SONIDOS de corta duración, señala entrada o salida
; conectada (ON).

DECO_0123
movlw TRES
movwf DIGITO
call TONO_DECODIF
movf DIGITEMP, W
subwf DIGITO, W
btfss STATUS, C
goto C_CERO ; DIGITEMP > DIGITO
goto DIGI123 ; DIGITEMP <= DIGITO

C_CERO
movf DIGITEMP, W
xorlw CERO

```

```

    btfss STATUS, Z
    goto MANDO
    clrf DIGITEMP

DIGI123
    movf DIGITEMP, W
    movwf DIGITO
    goto INC_MODO

DIGITO_01
    call TONO_DECODIF
    movf DIGITEMP, W
    xorlw CERO

    btfsc STATUS, Z
    clrf DIGITEMP
    swapf DIGITO, W
    iorwf DIGITEMP, F
    movlw 0x00
    xorwf DIGITEMP, W
    btfsc STATUS, Z
    goto CODIGO_00
    movlw 0x10
    xorwf DIGITEMP, W

    btfsc STATUS, Z
    goto RELE_OFF
    movlw 0x20
    xorwf DIGITEMP, W
    btfsc STATUS, Z
    goto RELE_OFF
    movlw 0x30
    xorwf DIGITEMP, W
    btfsc STATUS, Z
    goto RELE_OFF
    movlw 0x11
    xorwf DIGITEMP, W

    btfsc STATUS, Z
    goto RELE_ON
    movlw 0x21
    xorwf DIGITEMP, W
    btfsc STATUS, Z
    goto RELE_ON
    movlw 0x31
    xorwf DIGITEMP, W
    btfsc STATUS, Z
    goto RELE_ON
    goto MANDO

CODIGO_00
    movlw .24 ; Retardo de 3 segundos
    movwf CONTA2
    movlw .162
    call RETARDO_XXXMS
    decfsz CONTA2, F

```

```

    goto    $-3
    movlw   .4
    movwf   CONTA
    movlw   B'00000001'
    movwf   DIGITEMP
    andwf   PORTA,W

    btfsc   STATUS,Z
    goto    BEEP1
    call    RETARDO_75MS
    goto    DOS_BEEP

INFORMA
    movwf   DIGITEMP
    andwf   PORTA,W
    btfss   STATUS,Z
    goto    BEEP1
    call    RETARDO_75MS

DOS_BEEP
    call    SONIDO           ; Envía beeps a la línea
                            ; telefónica de un tono de 800Hz
                            ; de una duración de 75ms

BEEP1
                            ; intervalo de 75ms
    movlw   .150
    movwf   TEMP2
    movlw   .165

    call    RETARDO_XXXUS
    decfsz  TEMP2,F
    goto    $-3

UN_BEEP
    call    SONIDO
    bcf     STATUS,C
    rlf     DIGITEMP,F

    decfsz  CONTA,F
    goto    INFORMA
    goto    BUCLE_MANDO

; Se activa o desactiva el relé respectivo.

RELE_OFF
    movlw   B'11111101'
    movwf   DIGITEMP

OF_OTRO
    decfsz  DIGITO,F
    goto    OF_ROT
    goto    OF_SIN

OF_ROT
    bsf     STATUS,C
    rlf     DIGITEMP,F
    goto    OF_OTRO

```

```
OF_SIN
    andwf    PORTA, F
    goto     INFORMA
    goto     BUCLE_MANDO
```

```
RELE_ON
    movlw    B'00000010'
    movwf    DIGITEMP
```

```
ON_OTRO
    decfsz   DIGITO, F
    goto     ON_ROT
    goto     ON_SIN
```

```
ON_ROT
    bcf      STATUS, C
    rlf      DIGITEMP, F
    goto     ON_OTRO
```

```
ON_SIN
    iorwf    PORTA, F
    goto     BUCLE_MANDO
```

```
; Se incrementa el MODO para seguir la secuencia programada
; para realizar el mando a distancia.
```

```
INC_MODO
    bcf      STATUS, C
    rlf      MODO, F
    btfss    MODO, 3
    goto     MANDO
    clrf     MODO
    goto     BUCLE_MANDO
```

```
END                                ; Directiva que indica fin de
                                   ; código de programa
```

```
;*****
```

6.3.2. Para el PIC16F876

El programa en ensamblador `ctnivel.asm` realiza las tareas siguientes: Configura los puertos; inicializa los GPR, el módulo LCD y los periféricos; lee datos de configuración de las variables de la memoria de datos EEPROM, configura interrupciones: TMR1-CCP1, TMR2 y PORTB <7-4>, muestra un mensaje de inicio en la pantalla y lee el estado de los pulsadores. Si los tres primeros pulsadores se mantienen pulsados (accionado se indica con x) o posteriormente después de realizado un Reset, se ingresa al MENU de configuración y si estos valores no son válidos se obliga a ser ingresadas nuevamente; así mismo se obtiene el nivel offset, se muestra los valores finales de las variables, y finalmente se graba estos valores en la EEPROM. Cuando se accionan los pulsadores emiten un beep (subrutina BEEP).

En funcionamiento normal el programa monitorea el nivel de agua, lo procesa para controlar el nivel máximo o mínimo y lo muestra en la pantalla LCD, señalando el estado de funcionamiento de las electrobombas.

Configurando el circuito de Control de Nivel

Luego de ingresado al MENU de configuración, se realiza la configuración (subrutina `MENU_CONFIG`) del circuito de Control de Nivel, vía los pulsadores P1, P2, P3 y E. La tabla 6.4 describe la función de los pulsadores para este caso.

Pulsador				Función
P1	P2	P3	E	
x	x	x		Se ingresa a MENU de configuración
x	x			Modo configuración
	x	x		Obtención del Nivel Offset
x		x		Muestra las variables con sus valores actuales
			x	Sale de cada función

Tabla 6.4 *Función de los pulsadores en la subrutina `MENU_CONFIG`.*

Configurando valores de las variables

En este caso vía los pulsadores se configuran los valores de las variables: unidad de medida (Unidad_M), nivel máximo (NivelMAX), nivel mínimo (NivelMIN) y nivel tope mínimo (NivelTOP); además los válida. Ver la tabla 6.5.

Pulsador				Función
P1	P2	P3	E	
		x		Cambia de forma cíclica la Unidad de Medida
x				Incrementa dígito seleccionado
	X			Selecciona próximo dígito
			x	Sale de cada variable configurada

Tabla 6.5 *Función de los pulsadores en la subrutina CONFIGURANDO.*

Formato de presentación de la variable en la pantalla LCD

La presentación de la variable y su respectivo valor en la configuración del sistema, obedece a un formato; tanto para las filas (ver tabla 6.6) o para las posiciones del caracter según la unidad de medida que fue elegida, como se muestra en la tabla 6.7.

En la 1ra. fila de la pantalla LCD se muestra la variable	En la 2da. fila de la pantalla LCD se muestra su valor
Unidad M	Unidad de medida
NivelMAX	Nivel Máximo de agua
NivelMIN	Nivel Mínimo de agua
NivelTOP	Nivel Tope mínimo de agua
NivelACT	Nivel Actual de Agua

Tabla 6.6 *Forma de presentación de la variable y su respectivo valor.*

Variables	Formato
En centímetros	□ □ □ cm
NivelMAX, NivelMIN, NivelTOP y NivelACT	
En metros	□. □ □ m
NivelMAX, NivelMIN, NivelTOP y NivelACT	

Tabla 6.7 *Formato de presentación de las variables en el módulo LCD, según la unidad de medida seleccionada.*

Memoria de datos EEPROM

Los valores por defecto de las variables, son grabados desde la dirección 00h (ver tabla 6.8) usando las subrutinas EEPROM_WR y VALORESACT_EE.

Variable	Dirección	Valor por defecto	Descripción
DIGI1NSER	00h	03h	Número de serie del equipo
DIGI2NSER	01h	07h	
U_MEDIDA1	02h	'c'	Unidad de medida
U_MEDIDA2	03h	'm'	
DIGI1NMAX	04h	01h	Nivel máximo de agua en el reservorio
DIGI2NMAX	05h	02h	
DIGI3NMAX	06h	00h	
DIGI1NMIN	07h	00h	Nivel mínimo de agua en el reservorio
DIGI2NMIN	08h	02h	
DIGI3NMIN	09h	00h	
DIGI1NTOP	0Ah	00h	Nivel mínimo tope en el agua en el reservorio
DIGI2NTOP	0Bh	00h	
DIGI3NTOP	0Ch	00h	
NIVELOFF0	0Dh	FFh	Nivel offset de agua en el reservorio.
NIVELOFF1	0Eh	FFh	

Tabla 6.8 *Valores por defecto de las variables en la memoria EEPROM.*

Adquisición y procesamiento

La adquisición de los niveles de agua se realiza con la subrutina CONVERTOR_AD y el procesamiento se realiza en el bucle principal del programa. Se controla el nivel de agua del reservorio N° 2 de la siguiente manera: Cuando baja del valor mínimo, se activa el relé de la Salida de Control_1 que enciende la electrobomba 1 y desactiva el relé de la Salida de Control_2 apagando la electrobomba 2. Cuando sube por encima del valor máximo desactiva el relé apagando la electrobomba 1 y activa el relé que enciende la electrobomba 2, presentándose el estado de memoria. Cuando se enciende estando el nivel de agua dentro de los niveles máximo y mínimo, las dos electrobombas funcionan, retrasado en 3 segundos la electrobomba 2.

Las figuras siguientes muestran el diagrama de flujo del programa ctnivel.asm.

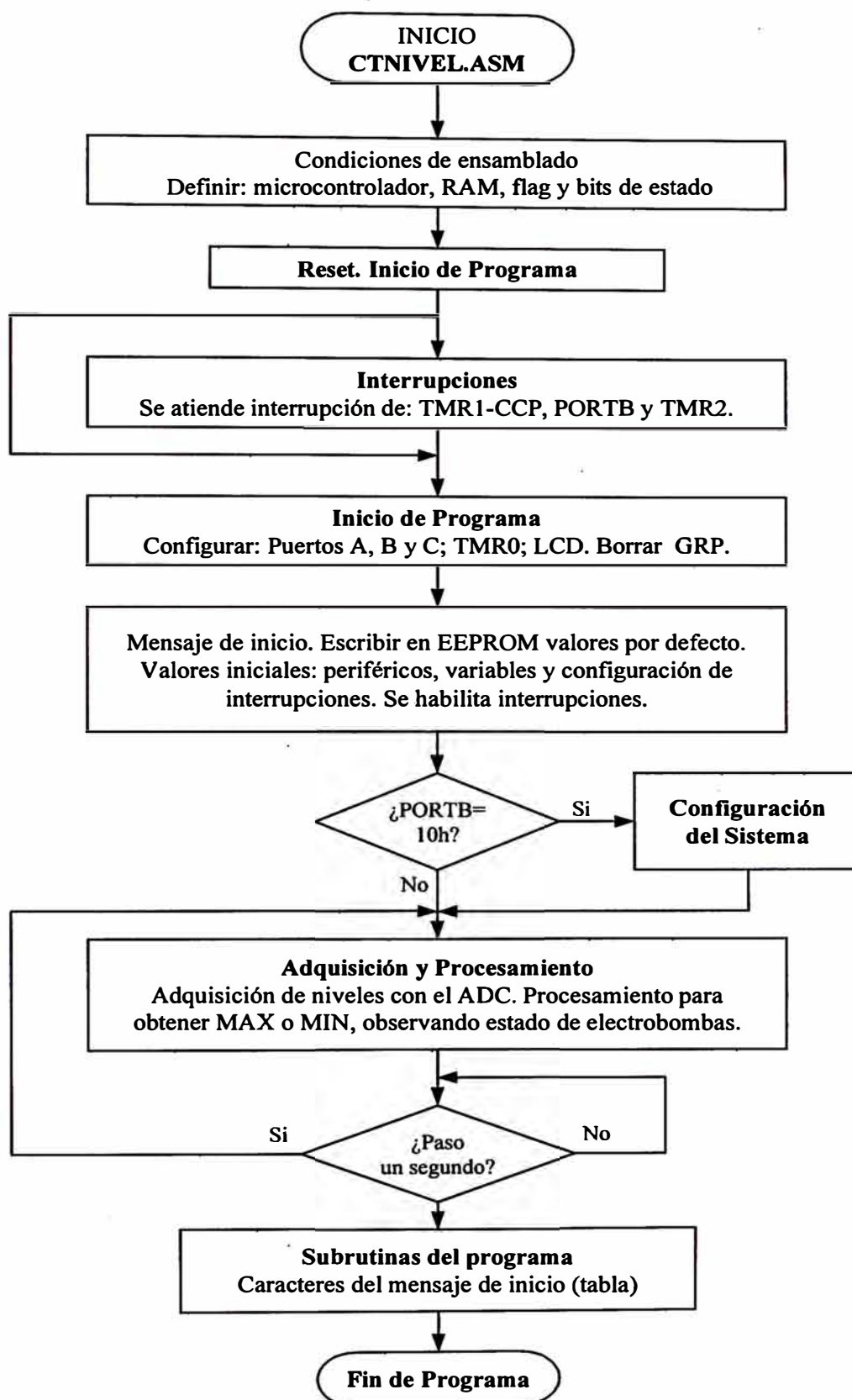


Figura 6.18 Diagrama de flujo del programa *ctnivel.asm*, que es utilizado por el circuito de Control de Nivel.

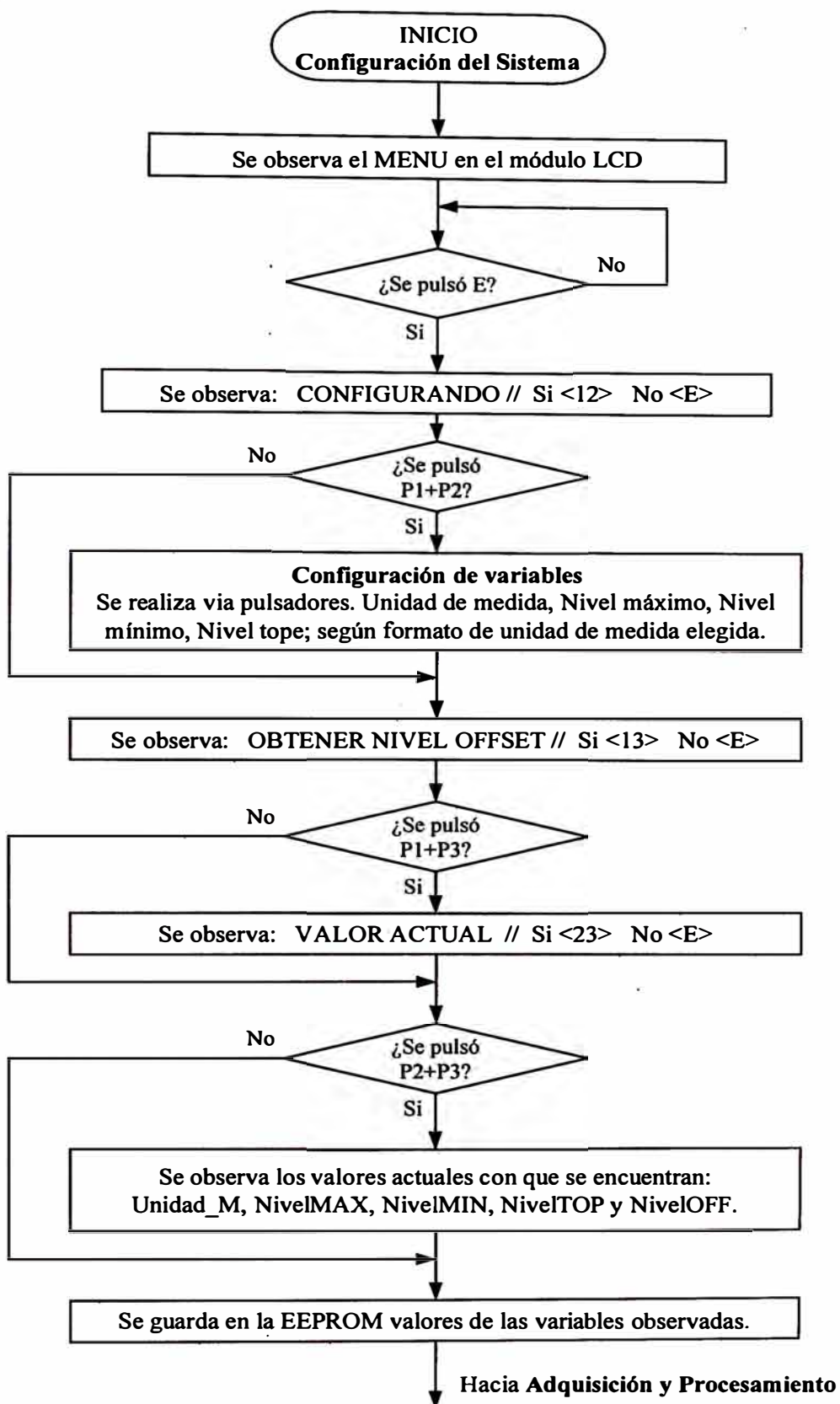


Figura 6.19 Diagrama de flujo de la subrutina MENU_CONFIG del programa *ctnivel.asm*. Presenta un MENU para configurar el sistema.

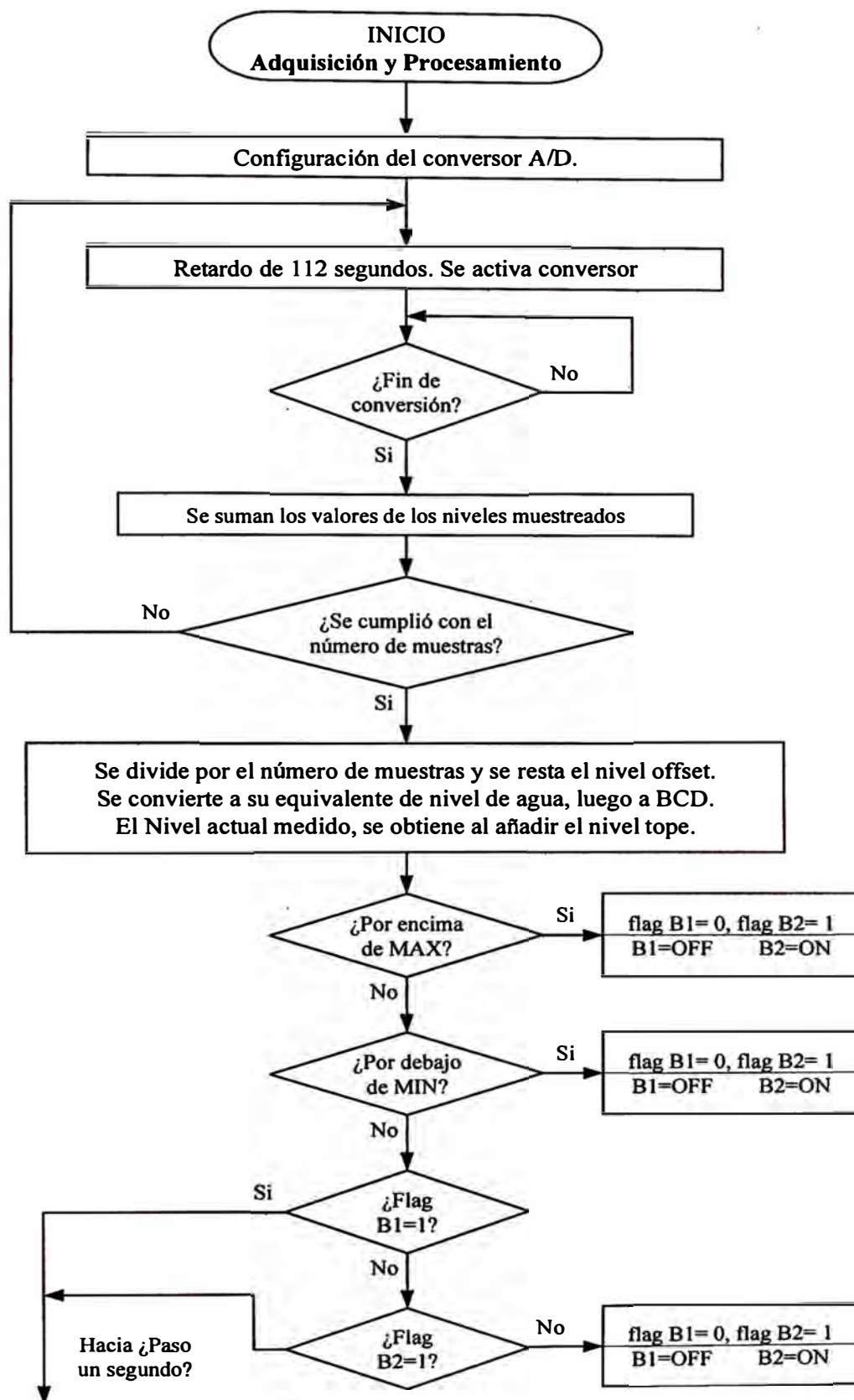


Figura 6.20 Diagrama de flujo para la adquisición y procesamiento de niveles de agua en el programa *ctnivel.asm*.

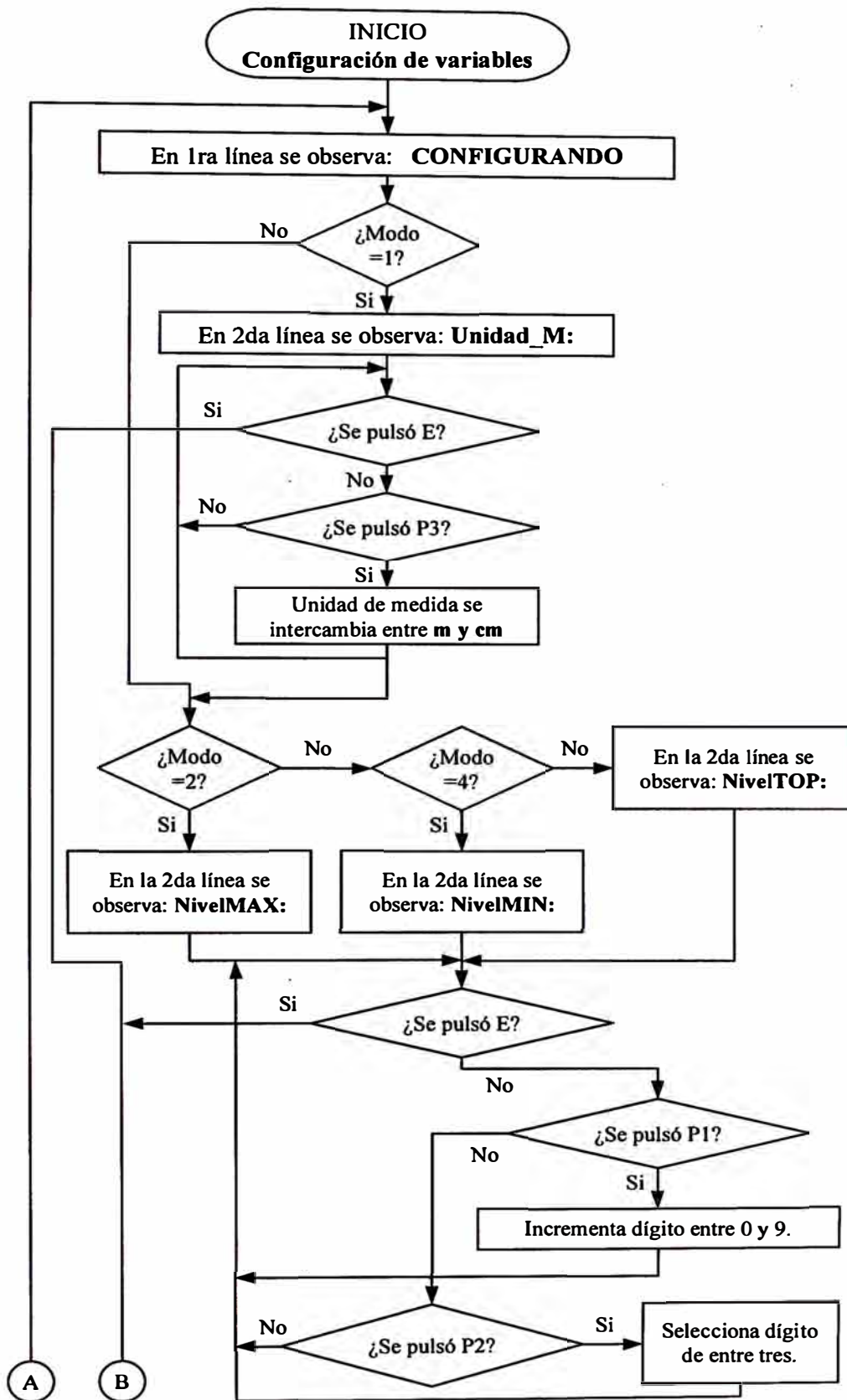


Figura 6.21 Diagrama de flujo de la subrutina CONFIGURA del programa *ctnivel.asm*. Configura los valores de las variables.

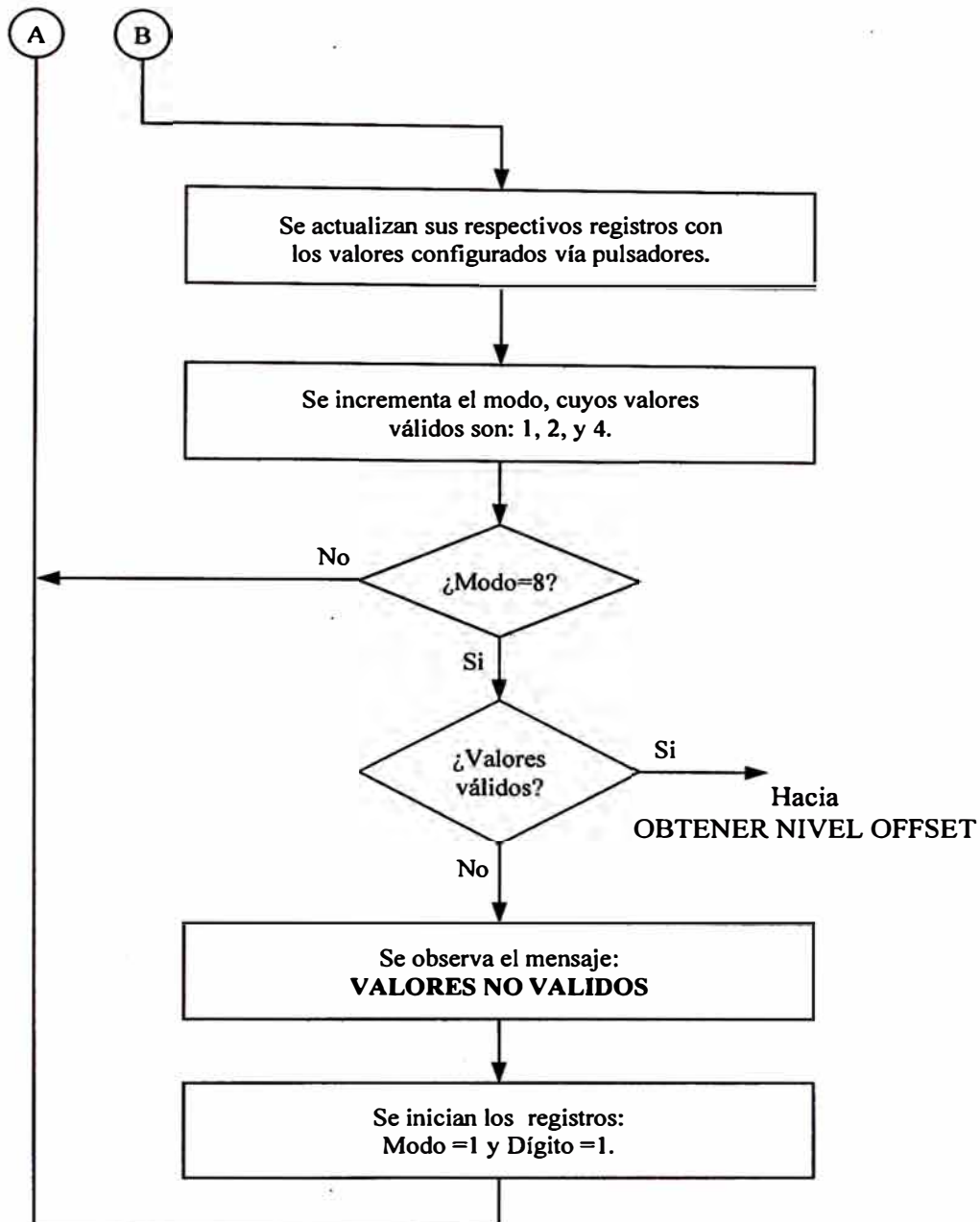


Figura 6.22 Diagrama de flujo de la subrutina CONFIGURA del programa *ctnivel.asm*. Configura los valores de las variables (continuación).

Además el programa se encarga de mostrar el valor actual del nivel de agua (NivelACT:) en el formato de la unidad elegida en la primera fila de la pantalla, señalando el estado de funcionamiento de las electrobombas (ON) o (OFF) en la segunda fila de la pantalla LCD.

Finalmente se muestra el listado del programa en nemónico para el PIC16F876.

```

;*****
;
; Listado del código de Programa: ctnivel.asm
;
; Ensamblado usando MPLAB-IDE 5.00.00
;
    TITLE      "PROGRAMA PARA EL CIRCUITO DE CONTROL DE NIVEL"

LIST      ST=ON   ;Suprime listado de símbolos
LIST      N=0     ;Suprime listado de saltos de página

;-----
;
;          UNIVERSIDAD NACIONAL DE INGENIERÍA
;
; Título de la tesis:
;
;          "DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE TELEMANDO
;          Y AUTOMATIZACIÓN DE BOMBEO DE AGUA, BASADO EN
;          MICROCONTROLADORES PIC"
;
; Autor: JAVIER CAMILO POMA PALACIOS
;
; Lima, Julio del 2002

;-----

LIST      P=16F876
RADIX     HEX

ERRORLEVEL   -302      ; Suprime mensaje "operando no en
                       ; esta en banco 0"
ERRORLEVEL   -306      ; Suprime mensaje "cruza límite de
                       ; página"

#INCLUDE     <P16F876.INC>

; -----
; DEFINICIÓN DE ETIQUETAS
; -----
; Registros de Propósito General desde la dirección 0x20.

    CBLOCK 0x20
DIGI1NSER          ; Número de serie del Equipo
DIGI2NSER
U_MEDIDA1
U_MEDIDA2
DIGI1NMAX          ; Nivel máximo, MSB
DIGI2NMAX
DIGI3NMAX
DIGI1NMIN          ; Nivel mínimo, MSB
DIGI2NMIN
DIGI3NMIN
DIGI1NTOP          ; Nivel tope mínimo, MSB
DIGI2NTOP
DIGI3NTOP

```

```

NIVELOFF0          ; Nivel offset, MSB
NIVELOFF1
DIGI1TEMP          ; Dígitos temporales
DIGI2TEMP
DIGI3TEMP
NIVELMAX1
NIVELMIN1
NIVELTOP1
PUNTERO            ; Inicio de mensaje
CALMODO            ; Modo de configuración
CALDIG             ; Dígito a seleccionar

; En operaciones matemáticas
AARGB0             ; Argumento de 32 bits, MSB
AARGB1
AARGB2
AARGB3             ; LSB
BARGB0             ; MSB
BARGB1
REMB0              ; Byte de residuo de la división

BCD_0              ; Variables BCD, MSB
BCD_1
BCD_2
TEMP0              ; Temporales, MSB
TEMP1
TEMP2
TEMP3
    ENDC

; Las variables localizadas en las direcciones 70H-7FH de la
; memoria RAM están disponibles en todos los bancos.

    CBLOCK 0x70
FLASH
UPDATO
CONTA
ADRTEMP
DATATEMP
LCDDATA
BOTONTEMP          ; Usado solo en ISR
FLAG_BOTON         ; Indica pulsador presionado
FLAG_BOMBA         ; Indica estado de bomba

; Guardar los registros antes de la interrupción.

TEMPFSR            ; Guardar el registro FSR durante ISR
TEMPPCLATH         ; Guardar el registro PCLATH durante ISR
TEMPW              ; Guardar el registro W durante ISR
TEMPSTAT           ; Guardar el registro STATUS durante ISR
CCPCONTA           ; CCP1 contador
    ENDC

; -----
; FLAG Y BIT DE ESTADO
; =====

```

```

#DEFINE BOTON1      FLAG_BOTON,7 ; Flag de estado de P1
#DEFINE BOTON2      FLAG_BOTON,6 ; Flag de estado de P2
#DEFINE BOTON3      FLAG_BOTON,5 ; Flag de estado de P3
#DEFINE BOTON4      FLAG_BOTON,4 ; Flag de estado de E
#DEFINE BOTON_IN    FLAG_BOTON,0
#DEFINE ENABLE      bsf   PORTA,2 ; E=1, activa LCD
#DEFINE DISABLE     bcf   PORTA,2 ; E=0, desactiva LCD
#DEFINE LEE         bsf   PORTA,1 ; R/W=1 Pone LCD en Modo RD
#DEFINE ESCRIBE     bcf   PORTA,1 ; R/W=0 Pone LCD en Modo WR
#DEFINE MOD_COMANDO bcf   PORTA,0 ; RS=0 (modo comando)
#DEFINE MODODATO    bsf   PORTA,0 ; RS=1 (modo datos)

      #INCLUDE      <N_DEFJC.INC> ; Valores por defecto

; -----
; VECTORES DE RESET E INTERRUPCIÓN
; =====

      ORG      0x0000      ; Vector de Reset
      goto    INICIO

      ORG      0x0004      ; Vector de Interrupción
      movwf   TEMPW        ; Guardar registros
      swapf   STATUS,W
      movwf   TEMPSTAT
      movf    FSR,W
      movwf   TEMPFSR
      movf    PCLATH,W
      movwf   TEMPPCLATH
      clrf   PCLATH        ; Página 0
      bcf    STATUS,RP0    ; SFR banco 0
      bcf    STATUS,RP1    ; SFR banco 0

; Averigua el flag que causa la interrupción y utilizar su
; respectiva subrutina.

      btfsc   PIR1,CCP1IF ; Test flag de interrupción del
                        ; CCP1
      goto    TMR1ISR
      btfsc   INTCON,RBIF  ; Test flag de interrupción del
                        ; PORTB
      goto    BOTONISR    ; Test flag cambio de estado
                        ; PORTB RB<7-4>
      btfsc   PIR1,TMR2IF ; Test flag de interrupción del
                        ; TMR2
      goto    TMR2ISR

; Si no hay interrupción, se retorna al programa principal.

      bcf    INTCON,TOIE   ; Desactiva interrupciones del
                        ; TMR0
      goto    POP

; CCP1 solo genera una interrupción. La tasa de interrupción
; es dividido por 2, para obtener una tasa de interrupción de
; un segundo.

```



```

TMR1ISR
  bcf      PIR1,CCP1IF      ; Borra flag de interrupción de
                              ; CCP1
  incf    CCPCONTA,F        ; Incrementa CCPCONTA
  btfsc   CCPCONTA,0        ; Divide tasa de interrupción x 2
  goto    POP
  bsf     UPDATO,0          ; Actualiza pantalla LCD
  incf    FLASH,F
  btfss   FLASH,0
  bsf     FLASH,7          ; Se pone a uno cada 2 segundos
  goto    POP

```

; Cuando un pulsador es pulsado, se guarda su estado en
; BOTONTEMP y se inicia TMR2 para evitar rebotes.

```

BOTONISR
  movf    PORTB,W          ; Lee PORTB
  andlw   0xF0             ; pulsadores están en RB<7-4>
  movwf   BOTONTEMP
  bcf     INTCON,RBIF      ; Borra flag de interrupción del
                              ; PORTB
  bcf     INTCON,RBIE      ; No interrupción por PORTB
  clrf    TMR2             ; TMR2=0
  bcf     PIR1,TMR2IF      ; Borra flag de interrupción del
                              ; TMR2
  bsf     STATUS,RP0       ; Banco 1
  bsf     PIE1,TMR2IE      ; Habilita interrupción por TMR2
  bcf     STATUS,RP0       ; Banco 0
  bsf     T2CON,TMR2ON     ; Se inicia TMR2
  goto    POP

```

; Después que TMR2 produce la interrupción, se lee nuevamente
; el PORTB se compara con el leído anteriormente si son
; iguales se pone a 1 su bit de estado o se descarta.

```

TMR2ISR
  bcf     T2CON,TMR2ON     ; Apaga TMR2
  bsf     STATUS,RP0       ; Banco 1
  bcf     PIE1,TMR2IE      ; No interrupción por TMR2
  bcf     STATUS,RP0       ; Banco 0
  bcf     PIR1,TMR2IF      ; Borra flag de interrupción del
                              ; TMR2
  clrf    FLAG_BOTON       ; Borra estado de pulsadores
  movf    PORTB,W          ; Lee PORTB
  bcf     INTCON,RBIF      ; Borra flag de interrupción del
                              ; PORTB
  bsf     INTCON,RBIE      ; Interrupción, cambio en PORTB
  andlw   0xF0
  subwf   BOTONTEMP,W      ; Compara con BOTONTEMP
  btfsc   STATUS,Z         ; coinciden las lecturas?
  goto    CHKBOTONES      ; Si
  clrf    BOTONTEMP       ; No, se ignora
  goto    POP

```

; La variable BOTONx retorna el bit de estado del pulsador
; presionado. Bit 0=1 del FLAG_BOTON señala que algún

```
; pulsador se pulso, Bit 1=1 señala que se pulso pulsador 1.
```

```
CHKBOTONES
```

```
  movf  BOTONTEMP,W
  btfsc STATUS,Z           ; Ignorar pulsador
  goto  POP
  call  BEEP
  bsf   BOTON_IN          ; Es presionado un pulsador
  btfss BOTONTEMP,7       ; Test pulsador 1
  bsf   BOTON1            ; Se pulso pulsador 1
  btfss BOTONTEMP,6       ; Test pulsador 2
  bsf   BOTON2            ; Se pulso pulsador 2
  btfss BOTONTEMP,5       ; Test pulsador 3
  bsf   BOTON3            ; Se pulso pulsador 3
  btfss BOTONTEMP,4       ; Test pulsador 4
  bsf   BOTON4            ; Se pulso pulsador 4
  goto  POP
```

```
; Restaura los valores que tenían los registros W, STATUS,
; PCLATH y FSR justo antes que se atienda la interrupción.
```

```
POP  movf  TEMPPCLATH,W
     movwf PCLATH
     movf  TEMPFSR,W
     movwf FSR
     swapf TEMPSTAT,W
     movwf STATUS
     swapf TEMPW,F
     swapf TEMPW,W
     retfie           ; Retorna desde interrupción
```

```
; -----
; INICIO DEL PROGRAMA
; =====
```

```
INICIO
```

```
  clrf  PORTA           ; Borra puertos A, B y C
  clrf  PORTB
  clrf  PORTC
  clrf  STATUS
  bsf   STATUS,RP0      ; Banco 1
  bcf   STATUS,RP1
  movlw B'00000110'
  movwf ADCON1
  movlw B'10000111'
  movwf OPTION_REG
  clrf  PORTC
  bcf   STATUS,RP0
  clrf  INTCON
  call  UP_LCD          ; Configura puertos para el LCD
  call  LCD_INI
  movlw B'00001100'    ; LCD ON, cursor OFF
  call  LCD_CMD
  movlw B'00000001'    ; Borra LCD y cursor en posición 0
  call  LCD_CMD
  movlw B'00000110'
```

```

    call    LCD_CMD
    call    MENSAJE_INI

; La subrutina WRDATOS_CONFIG escribe los valores por defecto
; del archivo N_DEFJC.INC en la memoria de datos EEPROM. Solo
; se utiliza en una unidad nueva.

    call    WRDATOS_CONFIG
    call    VALORES_INIT
    bsf     INTCON,GIE      ; Habilita las interrupciones
    movf    PORTB,W        ; Lee pulsadores
    andlw   0xF0
    sublw   0x10           ; Tres primeros pulsadores
    btfss   STATUS,Z       ; Tres primeros P1,P2 y P3?
    goto    CONTINUA      ; No
    call    MENU_CONFIG    ; Si
    call    VALORESACT_EE

CONTINUA
    movlw   B'00000001'    ; Borra LCD y cursor en posición 0
    call    LCD_CMD

; -----
; Bucle del PROGRAMA
; -----
; Visualiza en la pantalla LCD el nivel actual de líquido, el
; cual se actualiza cada segundo.

MONIT_NIVEL
    bsf     STATUS,RP0     ; Banco 1
    bcf     STATUS,RP1
    movlw   B'00100000'    ; RA<5> entrada
    movwf   TRISA
    movlw   B'10000010'    ; RA<5:3-0> entradas analógicas
    movwf   ADCON1
    bcf     STATUS,RP0
    movlw   B'10100001'    ; Configura y activa ADC
    movwf   ADCON0        ; para el canal 4
    call    CONVERSOR_AD

; Se pasa el valor binario obtenido a su valor equivalente de
; nivel de líquido, luego se pasa a BCD.

    clrf    BARGB0         ; MSB
    movlw   FACTCONV
    movwf   BARGB1
    call    MUL_AB
    call    BITS16_BCD

; Se resta el Nivel Offset (NivelOFF) y se suma el Nivel Tope
; mínimo (NivelTOP) al valor adquirido.

    clrf    BCD_2
    call    BCD5_16BITS
    movf    BARGB0,W
    movwf   AARGB0

```

```

movf    BARGB1,W
movwf   AARGB1
movf    NIVELOFF0,W
movwf   BCD_0
movf    NIVELOFF1,W
movwf   BCD_1
clrf    BCD_2
call    BCD5_16BITS
call    RESTA_AB           ; Resultado en AARGB
movf    DIGI1NTOP,W
movwf   BCD_0
swapf   DIGI2NTOP,W
andlw   0xF0
movwf   BCD_1
movf    DIGI3NTOP,W
andlw   0x0F
iorwf   BCD_1,F
clrf    BCD_2
call    BCD5_16BITS
call    SUMA_AB
movf    AARGB0,W
movwf   AARGB2
movf    AARGB1,W
movwf   AARGB3
call    BITS16_BCD

; Visualiza el resultado en el pantalla LCD.

bsf     STATUS,RP0         ; Banco 1
movlw   B'00000110'       ; Puerto A digital
movwf   ADCON1
bcf     STATUS,RP0
call    UP_LCD            ; Configura puertos para el LCD
movlw   0x82
call    LCD_CMD
movlw   'N'
call    LCD_DATO
movlw   'i'
call    LCD_DATO
movlw   'v'
call    LCD_DATO
movlw   'e'
call    LCD_DATO
movlw   'l'
call    LCD_DATO
movlw   'A'
call    LCD_DATO
movlw   'C'
call    LCD_DATO
movlw   'T'
call    LCD_DATO
movlw   ':'
call    LCD_DATO
movlw   ' '
call    LCD_DATO
movlw   'm'

```

```

xorwf  U_MEDIDA1,W
btfss  STATUS,Z
goto   ENCM
movf   BCD_0,W
andlw  0x0F
iorlw  0x30
call   LCD_DATO
movlw  '.'
call   LCD_DATO
swapf  BCD_1,W
andlw  0x0F
iorlw  0x30
call   LCD_DATO
movf   BCD_1,W
andlw  0x0F
iorlw  0x30
call   LCD_DATO
movlw  ' '
call   LCD_DATO
movf   U_MEDIDA1,W
call   LCD_DATO
goto   COMPARA

```

ENCM

```

movlw  0x00
xorwf  BCD_0,W
btfss  STATUS,Z
goto   NCERO
movlw  ' '
call   LCD_DATO
goto   CER01

```

NCERO

```

movf   BCD_0,W
andlw  0x0F
iorlw  0x30
call   LCD_DATO

```

CER01

```

swapf  BCD_1,W
andlw  0x0F
iorlw  0x30
call   LCD_DATO
movf   BCD_1,W
andlw  0x0F
iorlw  0x30
call   LCD_DATO
movlw  ' '
call   LCD_DATO
movf   U_MEDIDA1,W
call   LCD_DATO
movf   U_MEDIDA2,W
call   LCD_DATO

```

; Se compara el nivel medido del reservorio con el máximo o
; mínimo para activar o desactivar la BOMBA1 o BOMBA2.

COMPARA

```

movf   BCD_0,W
subwf  DIGI1NMAX,W
btfss  STATUS,C
goto   SUP_MAX           ; BCD_0 > DIGI1NMAX
btfss  STATUS,Z         ; BCD_0 <= DIGI1NMAX
goto   MIN_NIVEL        ; BCD_0 < DIGI1NMAX
movf   NIVELMAX1,W      ; BCD_0 = DIGI1NMAX
subwf  BCD_1,W
btfsc  STATUS,C
goto   SUP_MAX           ; BCD_1 >= NIVELMAX1

```

MIN_NIVEL

```

movf   BCD_0,W
subwf  DIGI1NMIN,W
btfss  STATUS,C
goto   NORMAL           ; BCD_0 > DIGI1NMIN
btfss  STATUS,Z         ; BCD_0 <= DIGI1NMIN
goto   INF_MIN          ; BCD_0 < DIGI1NMIN
movf   BCD_1,W          ; BCD_0 = DIGI1NMIN
subwf  NIVELMIN1,W
btfss  STATUS,C
goto   NORMAL           ; BCD_1 > NIVELMIN1

```

INF_MIN

```

btfsc  FLAG_BOMBA,0     ; Bomba1
goto   R_ADQUI
bsf    PORTC,0          ; BCD_1 <= NIVELMIN1, Bomba1 ON
bcf    PORTC,2          ; Bomba2 OFF
bsf    FLAG_BOMBA,0
bcf    FLAG_BOMBA,1
bcf    FLAG_BOMBA,2
movlw  0xC3
call   LCD_CMD
movlw  'B'
call   LCD_DATO
movlw  '1'
call   LCD_DATO
movlw  ':'
call   LCD_DATO
movlw  ' '
call   LCD_DATO
movlw  'O'
call   LCD_DATO
movlw  'N'
call   LCD_DATO
movlw  ','
call   LCD_DATO
movlw  'B'
call   LCD_DATO
movlw  '2'
call   LCD_DATO
movlw  ':'
call   LCD_DATO
movlw  ' '
call   LCD_DATO

```

```

movlw 'O'
call LCD_DATO
movlw 'F'
call LCD_DATO
movlw 'F'
call LCD_DATO
goto R_ADQUI

```

NORMAL

```

btfsc FLAG_BOMBA,0 ; Bomba1
goto R_ADQUI
btfsc FLAG_BOMBA,1
goto R_ADQUI
btfsc FLAG_BOMBA,2 ; Bomba2
goto R_ADQUI
bsf PORTC,0 ; Bomba1 ON
call RETARDO_1S
call RETARDO_1S
bsf PORTC,2 ; Bomba2 ON
bcf FLAG_BOMBA,0
bcf FLAG_BOMBA,1
bcf FLAG_BOMBA,2
movlw 0xC3
call LCD_CMD
movlw 'B'
call LCD_DATO
movlw '1'
call LCD_DATO
movlw ':'
call LCD_DATO
movlw ' '
call LCD_DATO
movlw 'O'
call LCD_DATO
movlw 'N'
call LCD_DATO
movlw ' '
call LCD_DATO
movlw ' '
call LCD_DATO
movlw 'B'
call LCD_DATO
movlw '2'
call LCD_DATO
movlw ':'
call LCD_DATO
movlw ' '
call LCD_DATO
movlw 'O'
call LCD_DATO
movlw 'N'
call LCD_DATO
goto R_ADQUI

```

SUP_MAX

```

btfsc FLAG_BOMBA,2 ; Bomba2

```

```

goto    R_ADQUI
bcf     PORTC, 0           ; Bomba1 OFF
bsf     PORTC, 2           ; Bomba2 ON
bcf     FLAG_BOMBA, 0
bcf     FLAG_BOMBA, 1
bsf     FLAG_BOMBA, 2
movlw   0xC3
call    LCD_CMD
movlw   'B'
call    LCD_DATO
movlw   '1'
call    LCD_DATO
movlw   ':'
call    LCD_DATO
movlw   ' '
call    LCD_DATO
movlw   'O'
call    LCD_DATO
movlw   'F'
call    LCD_DATO
movlw   'F'
call    LCD_DATO
movlw   ','
call    LCD_DATO
movlw   'B'
call    LCD_DATO
movlw   '2'
call    LCD_DATO
movlw   ':'
call    LCD_DATO
movlw   ' '
call    LCD_DATO
movlw   'O'
call    LCD_DATO
movlw   'N'
call    LCD_DATO

R_ADQUI
    bcf     UPDATO, 0

RET_ADQ
    btfsc   UPDATO, 0           ; Ha pasado un segundo?
    goto    MONIT_NIVEL       ; Si
    goto    RET_ADQ           ; No. Esperar

; -----
; INICIO DE SUBROUTINAS DEL PROGRAMA
; =====
; -----
; Subrutina MENS_MENU
; -----
; Muestra el mensaje * MENU <12>,<13>,<23>,<E> *, que indica
; la configuración del sistema.

MENS_MENU
    movlw   B'00000001'       ; Borra LCD y cursor en posición 0

```



```

call    LCD_CMD
movlw   0x86
call    LCD_CMD
movlw   '*'
call    LCD_DATO
movlw   ' '
call    LCD_DATO
movlw   'M'
call    LCD_DATO
movlw   'E'
call    LCD_DATO
movlw   'N'
call    LCD_DATO
movlw   'U'
call    LCD_DATO
movlw   ' '
call    LCD_DATO
movlw   '*'
call    LCD_DATO
movlw   0xC1
call    LCD_CMD
movlw   '<'
call    LCD_DATO
movlw   '1'           ; Configuración
call    LCD_DATO
movlw   '2'
call    LCD_DATO
movlw   '>'
call    LCD_DATO
movlw   ','
call    LCD_DATO
movlw   '<'           ; Obtener NivelOFF
call    LCD_DATO
movlw   '1'
call    LCD_DATO
movlw   '3'
call    LCD_DATO
movlw   '>'
call    LCD_DATO
movlw   ','
call    LCD_DATO
movlw   '<'           ; Valores actuales
call    LCD_DATO
movlw   '2'
call    LCD_DATO
movlw   '3'
call    LCD_DATO
movlw   '>'
call    LCD_DATO
movlw   ' '
call    LCD_DATO
movlw   '<'
call    LCD_DATO
movlw   'E'
call    LCD_DATO
movlw   '>'

```

```

    call    LCD_DATO
    return                                ; Retorna desde subrutina

; -----
; Subrutina SI_NO
; -----
; Esta subrutina nos da opción de elegir si se configura (Si)
; la variable activa o se abandona sin configurar (No).

SI_NO
    movlw  0xC3
    call   LCD_CMD
    movlw  'S'
    call   LCD_DATO
    movlw  'i'
    call   LCD_DATO
    movlw  '<'
    call   LCD_DATO
    movf   TEMP0,W
    call   LCD_DATO
    movf   TEMP3,W
    call   LCD_DATO
    movlw  '>'
    call   LCD_DATO
    movlw  ' '
    call   LCD_DATO
    movlw  ' '
    call   LCD_DATO
    movlw  ' '
    call   LCD_DATO
    movlw  'N'
    call   LCD_DATO
    movlw  'o'
    call   LCD_DATO
    movlw  '<'
    call   LCD_DATO
    movlw  'E'
    call   LCD_DATO
    movlw  '>'
    call   LCD_DATO
    return                                ; Retorna desde subrutina

; -----
; Subrutina MENS_CONFIG
; -----
; Muestra el mensaje CONFIGURANDO.

MENS_CONFIG
    movlw  B'00000001' ; Borra LCD y cursor en posición 0
    call   LCD_CMD
    movlw  0x84
    call   LCD_CMD
    movlw  'C'
    call   LCD_DATO
    movlw  'O'
    call   LCD_DATO

```

```

movlw  'N'
call   LCD_DATO
movlw  'F'
call   LCD_DATO
movlw  'I'
call   LCD_DATO
movlw  'G'
call   LCD_DATO
movlw  'U'
call   LCD_DATO
movlw  'R'
call   LCD_DATO
movlw  'A'
call   LCD_DATO
movlw  'N'
call   LCD_DATO
movlw  'D'
call   LCD_DATO
movlw  'O'
call   LCD_DATO
return                                ; Retorna desde subrutina

```

```

; -----
; Subrutina MENS_NIVELOFF
; -----

```

```

MENS_NIVELOFF
movlw  B'00000001' ; Borra LCD y cursor en posición 0
call   LCD_CMD
movlw  0x80
call   LCD_CMD
movlw  'O'
call   LCD_DATO
movlw  'B'
call   LCD_DATO
movlw  'T'
call   LCD_DATO
movlw  'E'
call   LCD_DATO
movlw  'N'
call   LCD_DATO
movlw  'E'
call   LCD_DATO
movlw  'R'
call   LCD_DATO
movlw  ' '
call   LCD_DATO
movlw  'N'
call   LCD_DATO
movlw  'I'
call   LCD_DATO
movlw  'V'
call   LCD_DATO
movlw  'E'
call   LCD_DATO
movlw  'L'

```

```

    call    LCD_DATO
    movlw  ' '
    call    LCD_DATO
    movlw  'O'
    call    LCD_DATO
    movlw  'F'
    call    LCD_DATO
    movlw  'F'
    call    LCD_DATO
    movlw  'S'
    call    LCD_DATO
    movlw  'E'
    call    LCD_DATO
    movlw  'T'
    call    LCD_DATO
    return                                ; Retorna desde subrutina

; -----
; Subrutina VALORN_OFF
; -----
; Obtiene el NivelOFF estando el sensor fuera del reservorio.

VALORN_OFF
    bsf    STATUS,RP0                    ; Banco 1
    bcf    STATUS,RP1
    movlw  B'00100000'
    movwf  TRISA
    movlw  B'10000010'                  ; RA<5:3-0> entradas analógicas
    movwf  ADCON1
    bcf    STATUS,RP0
    movlw  B'10100001'                  ; Configura y activa ADC,
    movwf  ADCON0                       ; entrada analógica por RA5
    call   CONVERSION_AD
    clrf   BARGB0                        ; MSB
    movlw  FACTCONV
    movwf  BARGB1
    call   MUL_AB
    call   BITS16_BCD
    bsf    STATUS,RP0                    ; Banco 1
    movlw  B'00000110'
    movwf  ADCON1
    bcf    STATUS,RP0                    ; Banco 0
    call   MENS_NIVELOFF
    movlw  0xC2
    call   LCD_CMD
    movlw  'N'
    call   LCD_DATO
    movlw  'i'
    call   LCD_DATO
    movlw  'v'
    call   LCD_DATO
    movlw  'e'
    call   LCD_DATO
    movlw  'l'
    call   LCD_DATO
    movlw  'O'

```

```

call    LCD_DATO
movlw   'F'
call    LCD_DATO
movlw   'F'
call    LCD_DATO
movlw   ':'
call    LCD_DATO
movlw   ' '
call    LCD_DATO
movf    BCD_0,W
movwf   NIVELOFF0
andlw   0x0F
movwf   DIGI1TEMP
swapf   BCD_1,W
andlw   0x0F
movwf   DIGI2TEMP
movf    BCD_1,W
movwf   NIVELOFF1
andlw   0x0F
movwf   DIGI3TEMP
call    VER_NORMAL      ; Muestra valor de NivelOFF
return      ; Retorna desde subrutina

```

```

; -----
; Subrutina MENS_VALORESACT
; -----
; Muestra el mensaje VALOR ACTUAL.

```

```

MENS_VALORESACT
movlw   B'00000001'      ; Borra LCD y cursor en posición 0
call    LCD_CMD
movlw   0x84
call    LCD_CMD
movlw   'V'
call    LCD_DATO
movlw   'A'
call    LCD_DATO
movlw   'L'
call    LCD_DATO
movlw   'O'
call    LCD_DATO
movlw   'R'
call    LCD_DATO
movlw   ' '
call    LCD_DATO
movlw   'A'
call    LCD_DATO
movlw   'C'
call    LCD_DATO
movlw   'T'
call    LCD_DATO
movlw   'U'
call    LCD_DATO
movlw   'A'
call    LCD_DATO
movlw   'L'

```

```

    call    LCD_DATO
    return                                     ; Retorna desde subrutina

; -----
; Subrutina VER_VALORESACT
; -----
; Muestra los valores con que operará el sistema luego de
; terminado la configuración.

VER_VALORESACT
    clrf    FLAG_BOTON      ; Borra estado de pulsadores
    call    MENS_VALORESACT
    btfss   CALMODO,0
    goto    VAL_1
    movlw   0xC4
    call    LCD_CMD
    movlw   'U'
    call    LCD_DATO
    movlw   'n'
    call    LCD_DATO
    movlw   'i'
    call    LCD_DATO
    movlw   'd'
    call    LCD_DATO
    movlw   'a'
    call    LCD_DATO
    movlw   'd'
    call    LCD_DATO
    movlw   ' '
    call    LCD_DATO
    movlw   'M'
    call    LCD_DATO
    movlw   ':'
    call    LCD_DATO
    movlw   ' '
    call    LCD_DATO
    goto    V_DISP

VAL_1
    movlw   0xC2
    call    LCD_CMD
    movlw   'N'
    call    LCD_DATO
    movlw   'i'
    call    LCD_DATO
    movlw   'v'
    call    LCD_DATO
    movlw   'e'
    call    LCD_DATO
    movlw   'l'
    call    LCD_DATO
    btfss   CALMODO,1
    goto    VAL_2
    movlw   0xC7
    call    LCD_CMD
    movlw   'M'
    call    LCD_DATO

```

```
    movlw  'A'
    call   LCD_DATO
    movlw  'X'
    call   LCD_DATO
    movlw  ':'
    call   LCD_DATO
    movlw  ' '
    call   LCD_DATO
    goto   V_DISP

VAL_2
    btfss  CALMODO, 2
    goto   VAL_3
    movlw  0xC7
    call   LCD_CMD
    movlw  'M'
    call   LCD_DATO
    movlw  'I'
    call   LCD_DATO
    movlw  'N'
    call   LCD_DATO
    movlw  ':'
    call   LCD_DATO
    movlw  ' '
    call   LCD_DATO
    goto   V_DISP

VAL_3
    btfss  CALMODO, 3
    goto   VAL_4
    movlw  0xC7
    call   LCD_CMD
    movlw  'T'
    call   LCD_DATO
    movlw  'O'
    call   LCD_DATO
    movlw  'P'
    call   LCD_DATO
    movlw  ':'
    call   LCD_DATO
    movlw  ' '
    call   LCD_DATO
    goto   V_DISP

VAL_4
    movlw  0xC7
    call   LCD_CMD
    movlw  'O'
    call   LCD_DATO
    movlw  'F'
    call   LCD_DATO
    movlw  'F'
    call   LCD_DATO
    movlw  ':'
    call   LCD_DATO
    movlw  ' '
    call   LCD_DATO
```

```

V_DISP
    clrf    FLAG_BOTON      ; Borra estado de pulsadores
    btfsc  CALMODO,0
    goto   VE_MED
    btfsc  CALMODO,1
    goto   V_MAX
    btfsc  CALMODO,2
    goto   V_MIN
    btfsc  CALMODO,3
    goto   V_TOP
    movf   NIVELOFF0,W
    movwf  BCD_0
    movf   NIVELOFF1,W
    movwf  BCD_1
    movf   BCD_0,W
    andlw  0x0F
    movwf  DIGI1TEMP
    swapf  BCD_1,W
    andlw  0x0F
    movwf  DIGI2TEMP
    movf   BCD_1,W
    andlw  0x0F
    movwf  DIGI3TEMP
    goto   VE_VAL

VE_MED
    movlw  0xCE
    call   LCD_CMD
    movf   U_MEDIDA1,W
    call   LCD_DATO
    movf   U_MEDIDA2,W
    call   LCD_DATO
    clrf   FLAG_BOTON      ; Borra estado de pulsadores

VAL  btfss  BOTON_IN
    goto  VAL
    btfsc  BOTON4
    goto  INCVAL
    clrf   FLAG_BOTON      ; Borra estado de pulsadores
    goto  VAL

V_MAX
    movf   DIGI1NMAX,W
    movwf  DIGI1TEMP
    movf   DIGI2NMAX,W
    movwf  DIGI2TEMP
    movf   DIGI3NMAX,W
    movwf  DIGI3TEMP
    goto   VE_VAL

V_MIN
    movf   DIGI1NMIN,W
    movwf  DIGI1TEMP
    movf   DIGI2NMIN,W
    movwf  DIGI2TEMP
    movf   DIGI3NMIN,W

```



```

        movwf DIGI3TEMP
        goto  VE_VAL

V_TOP
        movf  DIGI1NTEMP,W
        movwf DIGI1TEMP
        movf  DIGI2NTEMP,W
        movwf DIGI2TEMP
        movf  DIGI3NTEMP,W
        movwf DIGI3TEMP

VE_VAL
        call  VER_NORMAL
        clrf  FLAG_BOTON      ; Borra estado de pulsadores
        goto  VAL

INCVAL
        clrf  FLAG_BOTON
        bcf   STATUS,C
        rlf   CALMODO,F
        btfss CALMODO,5
        goto  VER_VALORESACT
        clrf  CALMODO
        return                ; Retorna desde subrutina

; -----
; Subrutina VALORESACT_EE
; -----
; Graba en la memoria de datos EEPROM los valores mostrados.

VALORESACT_EE
        movlw 0x02
        movwf ADRTEMP
        movf  U_MEDIDA1,W
        movwf DATATEMP
        call  EEPROM_WR
        incf  ADRTEMP,F
        movf  U_MEDIDA2,W
        movwf DATATEMP
        call  EEPROM_WR
        incf  ADRTEMP,F
        movf  DIGI1NMAX,W
        movwf DATATEMP
        call  EEPROM_WR
        incf  ADRTEMP,F
        movf  DIGI2NMAX,W
        movwf DATATEMP
        call  EEPROM_WR
        incf  ADRTEMP,F
        movf  DIGI3NMAX,W
        movwf DATATEMP
        call  EEPROM_WR
        incf  ADRTEMP,F
        movf  DIGI1NMIN,W
        movwf DATATEMP
        call  EEPROM_WR

```

```

incf   ADRTEMP, F
movf   DIGI2NMIN, W
movwf  DATATEMP
call   EEPROM_WR
incf   ADRTEMP, F
movf   DIGI3NMIN, W
movwf  DATATEMP
call   EEPROM_WR
incf   ADRTEMP, F
movf   DIGI1NTOP, W
movwf  DATATEMP
call   EEPROM_WR
incf   ADRTEMP, F
movf   DIGI2NTOP, W
movwf  DATATEMP
call   EEPROM_WR
incf   ADRTEMP, F
movf   DIGI3NTOP, W
movwf  DATATEMP
call   EEPROM_WR
incf   ADRTEMP, F
movf   NIVELOFF0, W
movwf  DATATEMP
call   EEPROM_WR
incf   ADRTEMP, F
movf   NIVELOFF1, W
movwf  DATATEMP
call   EEPROM_WR
return                                ; Retorna desde subrutina

```

```

; -----
; Subrutina CONVERSADOR_AD
; -----
; Se adquieren 10 muestras del sensor MPX2100 El valor que
; se obtiene en AARGB0:AARGB1 es equivalente a la presión del
; nivel de líquido.

```

```
CONVERSADOR_AD
```

```

clrf   AARGB1
clrf   AARGB0
movlw  .10
movwf  CONTA

```

```
AD_REPITE
```

```

movlw  0x25
movwf  TEMP1
decfsz TEMP1, F          ; TEMP1=37, RETARDOL_V=112us
goto   $-1
bcf    PIR1, ADIF
bsf    ADCON0, GO

```

```
AD_WAIT
```

```

btfss  PIR1, ADIF
goto   AD_WAIT
bsf    STATUS, RPO      ; Banco 1
movf   ADRESL, W

```

```

    bcf     STATUS,RP0
    movwf  BARGB1
    movf   ADRESH,W
    movwf  BARGB0
    call   SUMA_AB
    decfsz CONTA,F
    goto   AD_REPITE
    movlw  .10
    movwf  BARGB0
    call   DIVIDE_AB
    return                                ; Retorna desde subrutina

; -----
; Subrutina VALORES_INIT
; -----
; Configura los periféricos con sus valores iniciales.
; También trae los valores de las variables a configurar
; desde la EEPROM, estos son: la Unidad de medida, NivelMAX,
; NivelMIN, NivelTOP mínimo y NivelOFF.

VALORES_INIT
    bcf     STATUS,RP0                ; Banco 0
    bcf     STATUS,RP1
    movlw  B'00110001'                ; Configurar TMR1
    movwf  T1CON                      ; 8:1 predivisor
    clrf   CCPCONTA                   ; Borra CONTA
    movlw  0xF9
    movwf  CCPR1H
    movlw  0xFF
    movwf  CCPR1L
    movlw  B'00001011'
    movwf  CCP1CON
    movlw  B'00111001'                ; Configurando TMR2
    movwf  T2CON                      ; 4:1 predivisor, 8:1 postdivisor
    movlw  0x20                       ; Inicio de GPRs
    movwf  FSR

BORRAR
    clrf   INDF                       ; Borra GPR
    incf   FSR,F
    movf   FSR,W
    xorlw  0x80                       ; Test el registro 80h
    btfss  STATUS,Z                   ; Es el registro 80h?
    goto   BORRAR                     ; No, borrar GPR
    movlw  .15
    movwf  CONTA
    clrf   ADRTEMP
    bcf    STATUS,IRP                 ; Banco 0 o 1
    movlw  DIGI1NSER
    movwf  FSR

DAT_DEF
    call   EEPROM_RD
    movf   DATATEMP,W
    movwf  INDF
    incf   ADRTEMP,F

```

```

    incf   FSR, F
    decfsz CONTA, F
    goto   DAT_DEF
    call   VAL_MMT

CONFIG_INT
    clrf   PIR1
    clrf   INTCON
    bsf    STATUS, RPO           ; Banco 1
    clrf   PIE1
    clrf   PIE2
    bsf    PIE1, CCP1IE         ; Habilita interrupción por CCP1
    bcf    STATUS, RPO         ; Banco 0
    bsf    INTCON, RBIE        ; Interrupción por cambio en PORTB
    bsf    INTCON, PEIE        ; Interrupción por Periféricos
    return                               ; Retorna desde subrutina

VAL_MMT
    swapf  DIGI2NMAX, W         ; Valores MAX1, MIN1 y TOP1
    andlw  0xF0
    movwf  NIVELMAX1
    movf   DIGI3NMAX, W
    andlw  0x0F
    iorwf  NIVELMAX1, F
    swapf  DIGI2NMIN, W
    andlw  0xF0
    movwf  NIVELMIN1
    movf   DIGI3NMIN, W
    andlw  0x0F
    iorwf  NIVELMIN1, F
    swapf  DIGI2NTOP, W
    andlw  0xF0
    movwf  NIVELTOP1
    movf   DIGI3NTOP, W
    andlw  0x0F
    iorwf  NIVELTOP1, F
    return

; -----
; Subrutina MENSAJE_INI
; -----
; Muestra mensajes iniciales en la pantalla LCD.

MENSAJE_INI
    movlw  B'00000001'         ; Borra LCD y cursor en posición 0
    call   LCD_CMD
    movlw  0x86
    call   LCD_CMD
    clrf   PUNTERO
    call   MENS
    movlw  .75
    movwf  TEMP1
    call   RETARDO_X10MS
    movlw  0xC2
    call   LCD_CMD
    movlw  .9

```

```

movwf  PUNTERO
call   MENS
call   PARPADEO
call   RETARDO_1S
call   RETARDO_1S
movlw  B'00000001'      ; Borra LCD y cursor en posición 0
call   LCD_CMD
movlw  .26
movwf  PUNTERO
call   MENS
movlw  .75
movwf  TEMP1
call   RETARDO_X10MS
movlw  0xC2
call   LCD_CMD
movlw  .47
movwf  PUNTERO
call   MENS
call   PARPADEO
call   RETARDO_1S
call   RETARDO_1S
movlw  B'00000001'
call   LCD_CMD
movlw  0x87
call   LCD_CMD
movlw  .64
movwf  PUNTERO
call   MENS
movlw  .75
movwf  TEMP1
call   RETARDO_X10MS
movlw  0xC1
call   LCD_CMD
movlw  .70
movwf  PUNTERO
call   MENS
call   PARPADEO
call   RETARDO_1S
call   RETARDO_1S
return      ; Retorna desde subrutina

; -----
; Subrutina MENS_NOVALIDOS
; -----
; Este mensaje se muestra si los valores de configuración no
; son válidos.

MENS_NOVALIDOS
movlw  B'00000001'      ; Borra LCD y cursor en posición 0
call   LCD_CMD
movlw  0x81
call   LCD_CMD
movlw  'V'
call   LCD_DATO
movlw  'A'
call   LCD_DATO

```

```

movlw  'L'
call   LCD_DATO
movlw  'O'
call   LCD_DATO
movlw  'R'
call   LCD_DATO
movlw  'E'
call   LCD_DATO
movlw  'S'
call   LCD_DATO
movlw  ' '
call   LCD_DATO
movlw  'N'
call   LCD_DATO
movlw  'O'
call   LCD_DATO
movlw  ' '
call   LCD_DATO
movlw  'V'
call   LCD_DATO
movlw  'A'
call   LCD_DATO
movlw  'L'
call   LCD_DATO
movlw  'I'
call   LCD_DATO
movlw  'D'
call   LCD_DATO
movlw  'O'
call   LCD_DATO
movlw  'S'
call   LCD_DATO
return                                ; Retorna desde subrutina

; -----
; Subrutina MENU_CONFIG
; -----
; Muestra un MENU para realizar la configuración del sistema.

MENU_CONFIG
    clrfs    FLAG_BOTON    ; Borra estado de pulsadores
    call     MENS_MENU

MEN    btfss  BOTON_IN     ; Se pulso algún pulsador?
      goto   MEN           ; No. Repetir
      btfsc  BOTON4
      goto   MENU_1
      clrfs  FLAG_BOTON
      goto   MEN
      btfss  CALMODO,0
      goto   VAL_1

MENU_1
    clrfs    FLAG_BOTON    ; Borra estado de pulsadores
    call     MENS_CONFIG
    movlw   '1'

```

```

    movwf  TEMP0
    movlw  '2'
    movwf  TEMP3
    call   SI_NO           ; Seleccionar Si o No

MEN1  btfss  BOTON_IN      ; Esperar hasta pulsar un pulsador
      goto  MEN1
      btfsc  BOTON4
      goto  MENU_2
      btfsc  BOTON1      ; Se pulso pulsador 1?
      goto  M_BOT2      ; Si, test pulsador 2
      clrf  FLAG_BOTON   ; Borra estado de pulsadores
      goto  MEN1

M_BOT2
      btfsc  BOTON2      ; Se pulso pulsador 2?
      goto  MCONFIG     ; Si, realizar configuración
      clrf  FLAG_BOTON   ; Borra estado de pulsadores
      goto  MEN1

MCONFIG
      movlw  0x01
      movwf  CALMODO
      movwf  CALDIG
      call  CONFIGURA   ; Llama subrutina CONFIGURA

MENU_2
      clrf  FLAG_BOTON   ; Borra estado de pulsadores
      call  MENS_NIVELOFF
      movlw  '1'
      movwf  TEMP0
      movlw  '3'
      movwf  TEMP3
      call  SI_NO           ; Seleccionar Si o No

MEN2
      btfss  BOTON_IN      ; Esperar hasta pulsar un pulsador
      goto  MEN2
      btfsc  BOTON4
      goto  MENU_3
      btfsc  BOTON1      ; Se pulso pulsador 1?
      goto  M_BOT3      ; Si, test pulsador 3
      clrf  FLAG_BOTON   ; Borra estado de pulsadores
      goto  MEN2

M_BOT3
      btfsc  BOTON3      ; Se pulso pulsador 3?
      call  VALORN_OFF    ; Si, mostrar valor de NivelOFF
      clrf  FLAG_BOTON   ; Borra estado de pulsadores
      goto  MEN2

MENU_3
      clrf  FLAG_BOTON   ; Borra estado de pulsadores
      call  MENS_VALORESACT
      movlw  '2'
      movwf  TEMP0

```

```

    movlw  '3'
    movwf  TEMP3
    call   SI_NO          ; Seleccionar Si o No

MEN3
    btfss  BOTON_IN      ; Esperar hasta pulsar un pulsador
    goto   MEN3
    btfsc  BOTON4
    return
    btfsc  BOTON2        ; Se pulso pulsador 2?
    goto   MBOT3         ; Si, test pulsador 3
    clrf   FLAG_BOTON    ; Borra estado de pulsadores
    goto   MEN3

MBOT3
    btfsc  BOTON3        ; Se pulso pulsador 3?
    goto   ACTUAL        ; Si, mostrar valores actuales
    clrf   FLAG_BOTON    ; Borra estado de pulsadores
    goto   MEN3

ACTUAL
    movlw  0x01
    movwf  CALMODO
    call   VER_VALORESACT
    clrf   FLAG_BOTON    ; Borra estado de pulsadores
    return ; Retorna desde subrutina

; -----
; Subrutina CONFIGURA
; -----
; Utilizado para configurar vía pulsadores el sistema.

CONFIGURA
    clrf   FLAG_BOTON    ; Borra estado de pulsadores
    call   MENS_CONFIG
    btfss  CALMODO,0
    goto   MODO_1
    movlw  0xC4
    call   LCD_CMD
    movlw  'U'
    call   LCD_DATO
    movlw  'n'
    call   LCD_DATO
    movlw  'i'
    call   LCD_DATO
    movlw  'd'
    call   LCD_DATO
    movlw  'a'
    call   LCD_DATO
    movlw  'd'
    call   LCD_DATO
    movlw  ' '
    call   LCD_DATO
    movlw  'M'
    call   LCD_DATO
    movlw  ':'

```



```
    call    LCD_DATO
    movlw  ' '
    call    LCD_DATO
    goto   M_DISP

MODO_1
    movlw  0xC2
    call    LCD_CMD
    movlw  'N'
    call    LCD_DATO
    movlw  'i'
    call    LCD_DATO
    movlw  'v'
    call    LCD_DATO
    movlw  'e'
    call    LCD_DATO
    movlw  'l'
    call    LCD_DATO
    btfss  CALMODO,1
    goto   MODO_2
    movlw  0xC7
    call    LCD_CMD
    movlw  'M'
    call    LCD_DATO
    movlw  'A'
    call    LCD_DATO
    movlw  'X'
    call    LCD_DATO
    movlw  ':'
    call    LCD_DATO
    movlw  ' '
    call    LCD_DATO
    goto   M_DISP

MODO_2
    btfss  CALMODO,2
    goto   MODO_3
    movlw  0xC7
    call    LCD_CMD
    movlw  'M'
    call    LCD_DATO
    movlw  'I'
    call    LCD_DATO
    movlw  'N'
    call    LCD_DATO
    movlw  ':'
    call    LCD_DATO
    movlw  ' '
    call    LCD_DATO
    goto   M_DISP

MODO_3
    movlw  0xC7
    call    LCD_CMD
    movlw  'T'
    call    LCD_DATO
```

```

    movlw  'O'
    call   LCD_DATO
    movlw  'P'
    call   LCD_DATO
    movlw  ':'
    call   LCD_DATO
    movlw  ' '
    call   LCD_DATO

M_DISP
    clrf   FLAG_BOTON      ; Borra estado de pulsadores
    clrf   CALDIG          ;
    bsf    CALDIG,0        ; Selecciona dígito 1
    btfsc  CALMODO,0
    goto   VER_M
    btfsc  CALMODO,1
    goto   MAX
    btfsc  CALMODO,2
    goto   MIN
    movf   DIGI1NTOPI,W    ; Valor del NivelTOP
    movwf  DIGI1TEMP
    movf   DIGI2NTOPI,W
    movwf  DIGI2TEMP
    movf   DIGI3NTOPI,W
    movwf  DIGI3TEMP
    goto   VER_MOD

CAL1 btfss  BOTON_IN      ; Se pulso un pulsador?
    goto   CAL1           ; No, otra vez
    btfsc  BOTON4         ; Si. Es el pulsador 4?
    goto   EN_RAM
    btfsc  BOTON3
    goto   MEDIDA
    clrf   FLAG_BOTON      ; Borra estado de pulsadores
    goto   CAL1

MEDIDA
    movlw  'c'            ; Unidad de medida "cm"
    xorwf  U_MEDIDA1,W
    btfsc  STATUS,Z
    goto   SOLO_M
    movlw  'c'
    movwf  U_MEDIDA1
    movlw  'm'
    movwf  U_MEDIDA2
    goto   VER_M

SOLO_M
    movlw  'm'            ; Unidad de medida "m"
    movwf  U_MEDIDA1
    movlw  '-'
    movwf  U_MEDIDA2

VER_M
    movlw  0xCE
    call   LCD_CMD

```

```

    movf    U_MEDIDA1,W
    call    LCD_DATO
    movf    U_MEDIDA2,W
    call    LCD_DATO
    clrf    FLAG_BOTON      ; Borra estado de pulsadores
    goto    CAL1

MAX  movf    DIGI1NMAX,W      ; Valor del Nivel máximo
     movwf   DIGI1TEMP
     movf    DIGI2NMAX,W

     movwf   DIGI2TEMP
     movf    DIGI3NMAX,W
     movwf   DIGI3TEMP
     goto    VER_MOD

MIN  movf    DIGI1NMIN,W      ; Valor del Nivel mínimo
     movwf   DIGI1TEMP
     movf    DIGI2NMIN,W
     movwf   DIGI2TEMP
     movf    DIGI3NMIN,W
     movwf   DIGI3TEMP

VER_MOD
     clrf    FLASH
     call    VER_VALOR
     clrf    FLAG_BOTON      ; Borra estado de pulsadores

CAL2
     btfsc   BOTON_IN        ; Se pulso un pulsador?
     goto    OTROBOT         ; Si
     btfsc   UPDATO,0        ; No
     call    VER_VALOR
     goto    CAL2

OTROBOT
     btfsc   BOTON4
     goto    EN_RAM
     goto    CKBOT2         ; Test pulsador 2

INCMOD
     clrf    FLAG_BOTON      ; Borra estado de pulsadores
     bcf     STATUS,C        ; Borra bit carry antes de rotar
     rlf     CALMODO,F       ; Próximo MODO de configuración
     btfss   CALMODO,4       ; Modo válido?
     goto    CONFIGURA     ; Si
     clrf    FLAG_BOTON      ; No
     goto    VALIDAR

CKBOT2
     btfsc   BOTON2         ; Se pulso pulsador 2?
     goto    PROXDIG        ; Si, test pulsador 1
     btfsc   BOTON1         ; Se pulso pulsador 1?
     goto    CON_DIG        ; Si
     clrf    FLAG_BOTON      ; Borra estado de pulsadores
     goto    CAL2

```

```

PROXDIG
    clrf    FLAG_BOTON    ; Borra estado de pulsadores
    bcf     STATUS,C      ; Borra bit carry antes de rotar
    rlf     CALDIG,F      ; Selecciona próximo dígito
    btfss   CALDIG,3      ; Test dígito 3
    goto    CON_DEC       ; Válido dígito seleccionado
    movlw   0x01
    movwf   CALDIG

CON_DEC
    clrf    FLAG_BOTON
    goto    CAL2          ; Espera hasta pulsar un pulsador

CON_DIG
    clrf    FLAG_BOTON
    btfsc   CALDIG,0      ; Es seleccionado dígito 1?
    goto    INC_UNO       ; Si, dígito 1
    btfsc   CALDIG,1      ; Es seleccionado dígito 2?
    goto    INC_DOS       ; Si, dígito 2
    goto    INC_TRES      ; Si, dígito 3

INC_UNO
    incf    DIGI1TEMP,F   ; Incrementa dígito 1
    movlw   .10
    subwf   DIGI1TEMP,W
    btfsc   STATUS,Z
    clrf    DIGI1TEMP
    movf    DIGI1TEMP,W
    call    VER_VALOR
    goto    CON_DEC

INC_DOS
    incf    DIGI2TEMP,F   ; Incrementa dígito 2
    movlw   .10
    subwf   DIGI2TEMP,W
    btfsc   STATUS,Z
    clrf    DIGI2TEMP
    movf    DIGI2TEMP,W
    call    VER_VALOR
    goto    CON_DEC

INC_TRES
    incf    DIGI3TEMP,F   ; Incrementa dígito 3
    movlw   .10
    subwf   DIGI3TEMP,W
    btfsc   STATUS,Z
    clrf    DIGI3TEMP
    movf    DIGI3TEMP,W
    call    VER_VALOR
    goto    CON_DEC

EN_RAM
    clrf    FLAG_BOTON    ; Borra estado de pulsadores
    btfsc   CALMODO,0
    goto    MOD01
    btfsc   CALMODO,1

```

```

    goto    MODO2
    btfsc  CALMODO, 2
    goto    MODO3
    movf   DIGI1TEMP, W           ; Se guardan los valores actuales
    movwf  DIGI1NTOPI           ;
    movf   DIGI2TEMP, W
    movwf  DIGI2NTOPI           ;
    movf   DIGI3TEMP, W
    movwf  DIGI3NTOPI           ;
    goto    INCMOD              ; Salta a seleccionar próximo MODO

MODO1
    goto    INCMOD

MODO2
    movf   DIGI1TEMP, W
    movwf  DIGI1NMAX
    movf   DIGI2TEMP, W
    movwf  DIGI2NMAX
    movf   DIGI3TEMP, W
    movwf  DIGI3NMAX
    goto    INCMOD

MODO3
    movf   DIGI1TEMP, W
    movwf  DIGI1NMIN
    movf   DIGI2TEMP, W
    movwf  DIGI2NMIN
    movf   DIGI3TEMP, W
    movwf  DIGI3NMIN
    goto    INCMOD              ; Salta a seleccionar próximo MODO

VALIDAR
    call   VAL_MMT
    movf   DIGI1NMAX, W
    subwf  DIGI1NMIN, W
    btfss  STATUS, C
    goto   SI_VAL                ; DIGI1NMAX > DIGI1NMIN
    btfss  STATUS, Z             ; DIGI1NMAX <= DIGI1NMIN
    goto   NO_VAL                ; DIGI1NMAX < DIGI1NMIN
    movf   NIVELMAX1, W         ; DIGI1NMAX = DIGI1NMIN
    subwf  NIVELMIN1, W
    btfss  STATUS, C
    goto   SI_VAL                ; NIVELMAX1 > NIVELMIN1
    goto   NO_VAL                ; NIVELMAX1 <= NIVELMIN1

SI_VAL
    movf   DIGI1NMIN, W
    subwf  DIGI1NTOPI, W
    btfss  STATUS, C
    return                                ; DIGI1NMIN > DIGI1NTOPI
    btfss  STATUS, Z             ; DIGI1NMIN <= DIGI1NTOPI
    goto   NO_VAL                ; DIGI1NMIN < DIGI1NTOPI
    movf   NIVELMIN1, W         ; DIGI1NMIN = DIGI1NTOPI
    subwf  NIVELTOPI, W
    btfss  STATUS, C

```

```

    return                                ; NIVELMIN1 > NIVELTOP1

NO_VAL
    call    MENS_NOVALIDOS ; NIVELMIN1 <= NIVELTOP1
    call    RETARDO_1S
    movlw   0x01
    movwf   CALMOD0
    movwf   CALDIG
    goto    CONFIGURA

; -----
; Subrutina VER_VALOR
; -----
; Muestra valor en pantalla LCD, según unidad de medida.

VER_VALOR
    bcf     UPDATO, 0
    btfsc   FLASH, 0
    goto    VER_FLASH

VER_NORMAL
    movlw   0xCC
    call    LCD_CMD
    movlw   'm'
    xorwf   U_MEDIDA1, W
    btfss   STATUS, Z
    goto    EN_CM
    movf    DIGI1TEMP, W
    iorlw   0x30
    call    LCD_DATO
    movlw   '.'
    call    LCD_DATO
    movf    DIGI2TEMP, W
    iorlw   0x30
    call    LCD_DATO
    movf    DIGI3TEMP, W
    iorlw   0x30
    call    LCD_DATO
    movlw   ' '
    call    LCD_DATO
    movf    U_MEDIDA1, W
    call    LCD_DATO
    return                                ; Retorna desde subrutina

EN_CM
    movlw   0x00
    xorwf   DIGI1TEMP, W
    btfss   STATUS, Z
    goto    NO_CERO
    movlw   ' '
    call    LCD_DATO
    goto    CERO

NO_CERO
    movf    DIGI1TEMP, W
    iorlw   0x30

```

```

    call    LCD_DATO

CERO
    movf   DIGI2TEMP,W
    iorlw  0x30
    call   LCD_DATO
    movf   DIGI3TEMP,W
    iorlw  0x30
    call   LCD_DATO
    movlw  ' '
    call   LCD_DATO
    movf   U_MEDIDA1,W
    call   LCD_DATO
    movf   U_MEDIDA2,W
    call   LCD_DATO
    return                                ; Retorna desde subrutina

VER_FLASH
    movlw  0xCC
    call   LCD_CMD
    movlw  'm'
    xorwf  U_MEDIDA1,W
    btfss  STATUS,Z
    goto   FEN_CM
    btfss  CALDIG,0
    goto   VER1
    movlw  ' '
    call   LCD_DATO
    goto   ESPA1
    call   LCD_DATO

VER1
    movf   DIGI1TEMP,W
    iorlw  0x30
    call   LCD_DATO

ESPA1
    movlw  '.'
    call   LCD_DATO
    btfss  CALDIG,1
    goto   VER2
    movlw  ' '
    call   LCD_DATO
    goto   ESPA2

VER2
    movf   DIGI2TEMP,W
    iorlw  0x30
    call   LCD_DATO

ESPA2
    btfss  CALDIG,2
    goto   VER3
    movlw  ' '
    call   LCD_DATO
    goto   ESPA3

```

```

VER3
    movf    DIGI3TEMP,W
    iorlw  0x30
    call   LCD_DATO

ESPA3
    movlw  ' '
    call   LCD_DATO
    movf   U_MEDIDA1,W
    call   LCD_DATO
    return                                ; Retorna desde subrutina

FEN_CM
    btfss  CALDIG,0
    goto   FVER1
    movlw  ' _ '
    call   LCD_DATO
    goto   FCERO

FVER1
    movlw  0x00
    xorwf  DIGI1TEMP,W
    btfss  STATUS,Z
    goto   FNO_CERO
    movlw  ' '
    call   LCD_DATO
    goto   FCERO

FNO_CERO
    movf   DIGI1TEMP,W
    iorlw  0x30
    call   LCD_DATO

FCERO
    btfss  CALDIG,1
    goto   FVER2
    movlw  ' _ '
    call   LCD_DATO
    goto   FESPA2

FVER2
    movf   DIGI2TEMP,W
    iorlw  0x30
    call   LCD_DATO

FESPA2
    btfss  CALDIG,2
    goto   FVER3
    movlw  ' _ '
    call   LCD_DATO
    goto   FESPA3

FVER3
    movf   DIGI3TEMP,W
    iorlw  0x30
    call   LCD_DATO

```



```

; -----
; Subrutina WRDATOS_CONFIG
; -----
; Escribe los valores por defecto para la configuración, desde
; el archivo N_CALJC.INC en la EEPROM.

```

```
WRDATOS_CONFIG
```

```

    clrf    ADRTEMP
    movlw   DEFNSER1
    movwf   DATATEMP
    call    EEPROM_WR
    incf    ADRTEMP, F
    movlw   DEFNSER2
    movwf   DATATEMP
    call    EEPROM_WR
    incf    ADRTEMP, F
    movlw   DEFU_ME1
    movwf   DATATEMP
    call    EEPROM_WR
    incf    ADRTEMP, F
    movlw   DEFU_ME2
    movwf   DATATEMP
    call    EEPROM_WR
    incf    ADRTEMP, F
    movlw   DEFNMAX1
    movwf   DATATEMP
    call    EEPROM_WR
    incf    ADRTEMP, F
    movlw   DEFNMAX2
    movwf   DATATEMP
    call    EEPROM_WR
    incf    ADRTEMP, F
    movlw   DEFNMAX3
    movwf   DATATEMP
    call    EEPROM_WR
    incf    ADRTEMP, F
    movlw   DEFNMIN1
    movwf   DATATEMP
    call    EEPROM_WR
    incf    ADRTEMP, F
    movlw   DEFNMIN2
    movwf   DATATEMP
    call    EEPROM_WR
    incf    ADRTEMP, F
    movlw   DEFNMIN3
    movwf   DATATEMP
    call    EEPROM_WR
    incf    ADRTEMP, F
    movlw   DEFNTOP1
    movwf   DATATEMP
    call    EEPROM_WR
    incf    ADRTEMP, F
    movlw   DEFNTOP2
    movwf   DATATEMP
    call    EEPROM_WR
    incf    ADRTEMP, F

```

```

    movlw   DEFNTOP3
    movwf   DATATEMP
    call    EEPROM_WR
    return                                     ; Retorna desde subrutina

; -----
; Subrutina de TIEMPOS
; -----
; Retardos variables RETARDO_V, TEMP1*10ms y de un segundo.
; Utiliza el registro temporal TEMP1.

RETARDOL_V                                     ; RETARDOL_V=(TEMP1*3 + 1)*Tins
    decfsz TEMP1,F                             ; TEMP1=33, RETARDOL_V=100us
    goto   $-1
    return                                     ; Retorna desde subrutina

RETARDO_X10MS
    movlw   0xD9                               ; [Tins*(256-TMR0)*(DivFre)]*TEMP1
    movwf   TMR0
    btfss   INTCON,2
    goto   $-1
    bcf     INTCON,2
    decfsz  TEMP1,1
    goto   $-6
    return

RETARDO_1S                                     ; Retardo de un segundo
    movlw   0x64
    movwf   TEMP1
    call    RETARDO_X10MS
    return                                     ; Retorna desde subrutina

; -----
; Subrutina BEEP
; -----
; Genera un beep al pulsar un pulsador, utiliza TEMP1 y TEMP2.

BEEP
    movlw   .50
    movwf   TEMP2
    bsf     PORTA,3
    movlw   .82
    movwf   TEMP1
    call    RETARDOL_V
    bcf     PORTA,3
    movlw   .82
    movwf   TEMP1
    call    RETARDOL_V
    decfsz  TEMP2,F
    goto   $-9
    return                                     ; Retorna desde subrutina

; -----
; Subrutina MENS
; -----
; Esta subrutina saca mensajes por la pantalla LCD, siendo el

```

```

; primer caracter aquel que esta en la posición apuntada por
; PUNTERO, y debiendo ser el último 0x00.

MENS
    call    MENSAJE            ; Coge el caracter
    iorlw  0x00                ; Actualiza flags
    btfsc  STATUS,Z           ; Comprueba si es el último
    return
    call    LCD_DATO           ; Visualiza el caracter
    incf   PUNTERO,F          ; Incrementa contador de caracteres
    goto   MENS

; -----
; Subrutina PARPADEO
; -----
; Mantiene la pantalla LCD activado durante un segundo y a
; continuación genera un parpadeo de 5 veces, el encendido y
; apagado del LCD de un 1/4 segundo cada uno.

PARPADEO
    movlw  .100
    movwf  TEMP1
    call   RETARDO_X10MS      ; Retardo de un segundo
    movlw  .5
    movwf  TEMP2              ; 5 veces de intermitencia

PARPA1
    movlw  B'00001000'
    call   LCD_CMD            ; Apaga LCD
    movlw  .25
    movwf  TEMP1
    call   RETARDO_X10MS      ; Retardo de un 1/4 segundo
    movlw  B'00001100'
    call   LCD_CMD            ; Enciende LCD
    movlw  .25
    movwf  TEMP1
    call   RETARDO_X10MS      ; Retardo de un 1/4 segundo
    decfsz TEMP2,1
    goto   PARPA1
    return                    ; Retorna desde subrutina

; -----
; Subrutina RESTA_AB
; -----
; Resta AARG y BARG (16 - 16 bits) -> AARG(16).

RESTA_AB
    Comf   BARGB1,F           ; Negando BARGB (-BARGB -> BARGB)
    Incf   BARGB1,F
    Btfsc  STATUS,Z
    Decf   BARGB0,F
    Comf   BARGB0,F

; -----
; Subrutina SUMA_AB
; -----

```

```

; Suma AARG y BARG (16 + 16 bits) -> AARG(16).
SUMA_AB
    movf    BARGB1,W           ; AARGB + BARGB -> AARGB
    addwf   AARGB1,F           ; Sumando los LSB
    btfsc  STATUS,C           ; Sumando llevada
    incf   AARGB0,F
    movf   BARGB0,W
    addwf  AARGB0,F           ; Sumando los MSB
    return                                ; Retorna desde subrutina

; -----
; Subrutina MUL_AB
; -----
; Multiplica AARG*BARG (16*16 bits) -> AARG(16).
MUL_AB
    movlw  .16                 ; Inicialización de contador
    movwf  CONTA
    clrf   TEMPO
    clrf   TEMP1

MUL_LP
    rrf    BARGB0,F           ; Multiplica por 2
    rrf    BARGB1,F
    btfss  STATUS,C
    goto   NO_SUMA

    movf   AARGB1,W
    addwf  TEMP1,F
    btfsc  STATUS,C
    incf   TEMPO,F
    movf   AARGB0,W
    addwf  TEMPO,F

NO_SUMA
    rrf    TEMPO,F
    rrf    TEMP1,F
    rrf    TEMP2,F
    rrf    TEMP3,F
    decfsz CONTA,F
    goto   MUL_LP
    movf   TEMPO,W
    movwf  AARGB0
    movf   TEMP1,W
    movwf  AARGB1
    movf   TEMP2,W
    movwf  AARGB2
    movf   TEMP3,W
    movwf  AARGB3
    return                                ; Retorna desde subrutina

; -----
; Subrutina DIVIDE_AB
; -----
; Divide AARG y BARG (16/8 bits)->AARG (16) y residuo en REM.

```

```

DIVIDE_AB
    clrf    REMB0
    movlw  0x08
    movwf  CONTA

```

```

DIV16A
    rlf    AARGB0,W
    rlf    REMB0,F
    movf   BARGB0,W
    subwf  REMB0,F
    btfsc  STATUS,C
    goto   D168A
    addwf  REMB0,F
    bcf    STATUS,C

```

```

D168A
    rlf    AARGB0,F
    decfsz CONTA,F
    goto   DIV16A
    clrf   TEMP1
    movlw  0x08
    movwf  CONTA

```

```

DIV16B
    rlf    AARGB1,W
    rlf    REMB0,F
    rlf    TEMP1,F
    movf   BARGB0,W
    subwf  REMB0,F
    clrf   TEMP2
    clrw
    btfss  STATUS,C
    incfsz TEMP2,W
    subwf  TEMP1,F
    btfsc  STATUS,C
    goto   D168B
    movf   BARGB0,W
    addwf  REMB0,F
    clrf   TEMP2
    clrw
    btfsc  STATUS,C
    incfsz TEMP2,W
    addwf  TEMP1,F
    bcf    STATUS,C

```

```

D168B
    rlf    AARGB1,F
    decfsz CONTA,F
    goto   DIV16B
    return                                ; Retorna desde subrutina

```

```

; -----
; Subrutina BITS16_BCD
; -----
; Convierte un número binario de 16 bits (AARGB2:AARGB3) a 5
; dígitos BCD, para mostrar en el módulo LCD.

```

```

BITS16_BCD
    bcf     STATUS,C
    clrf   CONTA
    bsf    CONTA,4
    clrf   BCD_0
    clrf   BCD_1
    clrf   BCD_2

LOOP_16
    rlf    AARGB3,F
    rlf    AARGB2,F
    rlf    BCD_2,F
    rlf    BCD_1,F
    rlf    BCD_0,F
    decfsz CONTA,F
    goto   AJUSTE
    return                                ; Retorna desde subrutina

AJUSTE
    movlw  BCD_2
    movwf  FSR

    call   AJUSTE_BCD
    movlw  BCD_1
    movwf  FSR
    call   AJUSTE_BCD
    movlw  BCD_0
    movwf  FSR
    call   AJUSTE_BCD
    goto   LOOP_16

AJUSTE_BCD
    movlw  0x03
    addwf  INDF,W
    movwf  TEMP0

    btfsc  TEMP0,3
    movwf  INDF
    movf   INDF,W
    addlw  0x30
    movwf  TEMP0
    btfsc  TEMP0,7
    movwf  INDF
    return                                ; Retorna desde subrutina

; -----
; Subrutina BCD5_16BITS
; -----
; Convierte un número BCD de 5 dígitos (BCD_0:BCD_1:BCD_2) a
; un número binario de 16 bits BARGB0:BARGB1.

MUL10B
    andlw  0x0F
    addwf  BARGB1,F
    btfsc  STATUS,C
    incf   BARGB0,F

```

```

MUL10A
    bcf     STATUS,C
    rlf     BARGB1,W
    movwf   TEMP1
    rlf     BARGB0,W
    movwf   TEMPO
    bcf     STATUS,C
    rlf     BARGB1,F
    rlf     BARGB0,F
    bcf     STATUS,C
    rlf     BARGB1,F
    rlf     BARGB0,F
    bcf     STATUS,C
    rlf     BARGB1,F
    rlf     BARGB0,F
    movf    TEMP1,W
    addwf   BARGB1,F
    btfsc   STATUS,C
    incf    BARGB0,F
    movf    TEMPO,W
    addwf   BARGB0,F
    return                                     ; Retorna desde subrutina

BCD5_16BITS
    clrf    BARGB0
    movf    BCD_0,W
    andlw   0x0F
    movwf   BARGB1
    call    MUL10A
    swapf   BCD_1,W
    call    MUL10B
    movf    BCD_1,W
    call    MUL10B
    swapf   BCD_2,W
    call    MUL10B
    movf    BCD_2,W
    andlw   0x0F
    addwf   BARGB1,F
    btfsc   STATUS,C
    incf    BARGB0,F
    return                                     ; Retorna desde subrutina

; -----
; Subrutinas para el manejo del módulo LCD
; -----
; -----
; Subrutina UP_LCD
; -----
; Configura el microcontrolador para que trabaje con el LCD.

UP_LCD
    bsf     STATUS,RP0                       ; Banco 1
    movlw   B'00100000'                       ; RA2=E, RA1=R/W, RA0=RS,
    movwf   TRISA                               ; RA<5> entrada y RA<4-0> Salidas
    movlw   B'11110000'                       ; RB<7-4> entradas, RB<3-0> salidas
    movwf   TRISB

```



```

    bcf     STATUS,RP0      ; Banco 0
    MOD_COMANDO             ; RS=0
    DISABLE                 ; E=0
    return                  ; Retorna desde subrutina

; -----
; Subrutina LCD_BUSY
; -----
; Chequea el estado del flag BUSY del módulo LCD indicando,
; cuando esta activado o aún no ha terminado de procesar el
; comando anterior.

LCD_BUSY
    LEE                    ; Pone el LCD en Modo RD
    bsf     STATUS,RP0     ; Selecciona banco 1
    movlw  0xFF
    movwf  TRISB           ; Puerta B como entrada
    bcf     STATUS,RP0     ; Selecciona el banco 0
    ENABLE                 ; E=1
    nop                    ; Pulso de unmicrosegundo

BUSY_L
    swapf  PORTB,W         ; Lee <7-4> bits del flag BUSY,
    andlw  0xF0            ; Mascara para la salida
    movwf  TEMP1
    btfsc  TEMP1,7         ; Test flag BUSY
    goto   BUSY_L
    DISABLE                 ; E=0
    bsf     STATUS,RP0     ; Banco 1
    movlw  0xF0
    movwf  TRISB           ; RB<7-4> entradas, RB<3-0> salidas
    bcf     STATUS,RP0     ; Banco 0
    ESCRIBE
    return                  ; Retorna desde subrutina

; -----
; Subrutina LCD_DATO
; -----
; Esta subrutina pasa el contenido cargado en el registro W,
; el cual contiene un caracter ASCII, al PUERTO B; para ser
; visualizado en la pantalla LCD o escribirlo en la CGRAM.

LCD_DATO
    MOD_COMANDO             ; RS=0 (modo comando)
    movwf  LCDDATA         ; Valor ASCII a sacar por PORTB
    swapf  LCDDATA,W
    andlw  0x0F            ; 4 bits superiores
    movwf  PORTB
    call   LCD_BUSY        ; Espera a que se libere el LCD
    MODO_DATO              ; RS=1 (modo dato)
    ENABLE
    DISABLE
    MOD_COMANDO             ; RS=0
    movf   LCDDATA,W
    andlw  0x0F            ; 4 bits inferiores
    movwf  PORTB

```

```

    call    LCD_BUSY          ; Espera a que se libere el LCD
    MODO_DATO
    ENABLE          ; Se activa LCD, E=1
    DISABLE        ; E=0
    return         ; Retorna desde subrutina

; -----
; Subrutina LCD_CMD
; -----
; Subrutina parecida a la anterior, pero el contenido de W
; es el código de un comando para el LCD, que es necesario
; enviarlo también al PUERTO B para su ejecución.

LCD_CMD
    MOD_COMANDO          ; RS=0 (modo comando)
    movwf  LCDDATA       ; Valor ASCII a sacar por PORTB
    swapf  LCDDATA,W
    andlw  0x0F          ; 4 bits superiores
    movwf  PORTB
    call   LCD_BUSY     ; Espera a que se libere el LCD
    ENABLE
    DISABLE
    MODO_DATO
    nop                ; Pulso de un microsegundo
    MOD_COMANDO
    movf   LCDDATA,W
    andlw  0x0F          ; 4 bits inferiores
    movwf  PORTB
    call   LCD_BUSY     ; Espera a que se libere el LCD
    ENABLE          ; Se activa LCD, E=1
    DISABLE
    MODO_DATO
    return         ; Retorna desde subrutina

; -----
; Subrutina LCD_INI
; -----
; Se encarga de realizar la secuencia de inicialización del
; módulo LCD de acuerdo al fabricante.

LCD_INI    ESCRIBE
    movlw  B'00000011'
    movwf  PORTB
    ENABLE          ; Se activa LCD, E=1
    DISABLE
    movlw  .6        ; RETARDO=TEMP1*((3*256)+2)*Tins
    movwf  TEMP1     ; RETARDO=4,62ms
    clrf   TEMP2
    decfsz TEMP2,F
    goto   $-1
    decfsz TEMP1,F
    goto   $-3
    movlw  B'00000011'
    movwf  PORTB
    ENABLE          ; Se activa LCD, E=1
    DISABLE

```

```

movlw  .43
movwf  TEMP1
decfsz TEMP1,F          ; TEMP1=43, RETARDOL_V=130us
goto   $-1
movlw  B'00000011'
movwf  PORTB
ENABLE
DISABLE
movlw  .43
movwf  TEMP1
decfsz TEMP1,F
goto   $-1
movlw  0x02
movwf  PORTB
ENABLE          ; Se activa LCD, E=1
DISABLE
movlw  0x28      ; Interface de 4 bits, 2 líneas
call   LCD_CMD
return        ; Retorna desde subrutina

ORG     TB_MENSAJE

; -----
; CARACTERES DEL MENSAJE
; =====
; Mensaje a mostrarse en el LCD al iniciarse el programa.

MENSAJE
movlw  HIGH(TB_MENSAJE)
movwf  PCLATH      ; PCLATH para saltar a esta tabla
movf   PUNTERO,W   ; Inicio del mensaje
addwf  PCL,F

DT     'U','N','I',' ','F','I','E','E',0x00
DT     'I','N','G',' ',' ','E','L','E','C','T','R','O'
DT     'N',' ','I','C','A', 0x00

DT     'S','I','S','T','E','M','A',' ','D','E',' ','T'
DT     'E','L','E','M','A','N','D','O',0x00
DT     'Y',' ','A','U','T','O','M','A','T','I','Z','A'
DT     'C',' ','I','O','N', 0x00
DT     'A','U','T','O','R',0x00
DT     'J','A','V','I','E','R',' ',' ','C',' ',' ','P'
DT     'O','M','A',' ',' ','P',' ',' ',0x00

; -----
; Fin de PROGRAMA
; =====
; Fin del código de programa.

ORG     0x0800      ; Página 1
clr     PCLATH      ; Selecciona página 0
FILL   (goto 0x00),0x1000-$

END          ; Directiva que indica fin del
            ; código de programa.

```

```

;-----
; Listado del archivo: n_defjc.inc

      NOLIST

; -----
; Valores por DEFECTO
; -----
; Antes de iniciar la configuración del sistema las variables
; deben tener un valor inicial, estos valores se denominan
; valores por defecto y se encuentran guardados en la memoria
; de datos EEPROM.

      CONSTANT TB_MENSAJE=0X0780

#DEFINE DEFNSER1    0x03    ; Número de serie, MSB
#DEFINE DEFNSER2    0x07

#DEFINE DEFU_ME1    'c'    ; Unidad de medida
#DEFINE DEFU_ME2    'm'

#DEFINE DEFNMAX1    0x01    ; Nivel máximo, MSB
#DEFINE DEFNMAX2    0x02
#DEFINE DEFNMAX3    0x00

#DEFINE DEFNMIN1    0x00    ; Nivel mínimo, MSB
#DEFINE DEFNMIN2    0x02
#DEFINE DEFNMIN3    0x00

#DEFINE DEFNTOP1    0x00    ; Nivel tope mínimo, MSB
#DEFINE DEFNTOP2    0x00
#DEFINE DEFNTOP3    0x00

#DEFINE FACTCONV    0x32    ;Factor del convertidor

      LIST

;-----
;*****

```

CAPÍTULO VII

DESCRIPCIÓN FUNCIONAL DEL SISTEMA IMPLEMENTADO

7.1. Introducción

El abastecimiento de agua tanto para el campamento minero, como para el uso en la explotación minera se realiza con un sistema de bombeo (ver figura 7.1), se encuentra a cargo de un operario, el cual pone en funcionamiento a las dos electrobombas en horarios de funcionamiento establecidos o según requerimiento.

El sistema de bombeo de agua, es propiedad de la Unidad de Producción Julcani de la Compañía de Minas Buenaventura S.A.A.; consta principalmente de tres reservorios, dos electrobombas, tuberías de 1000 m. y 600 m. aproximadamente con sus respectivas válvulas Check, tableros eléctricos que se encargan de controlar el funcionamiento manual de las electrobombas, y tubería de alimentación desde el manantial al reservorio N° 1; tal como se muestra en la figura 7.1. Se encuentra en la localidad de Pampas, en el departamento de Huancavelica.

La electrobomba 1 con una potencia de 100 HP, se encuentra instalado en el reservorio N° 1 que está ubicado a 3987 msnm., se encarga de alimentar al reservorio N° 2 que es de capacidad menor que las restantes y está a una distancia de 1000 m. La electrobomba 2 tiene una potencia de 75 HP, está instalado en el reservorio N° 2 que está ubicado a 4184 msnm., se encarga de alimentar al reservorio N° 3 que es de mayor capacidad que las restantes y está a una distancia de 600 m. El reservorio N° 1 es alimentado permanentemente desde un manantial.

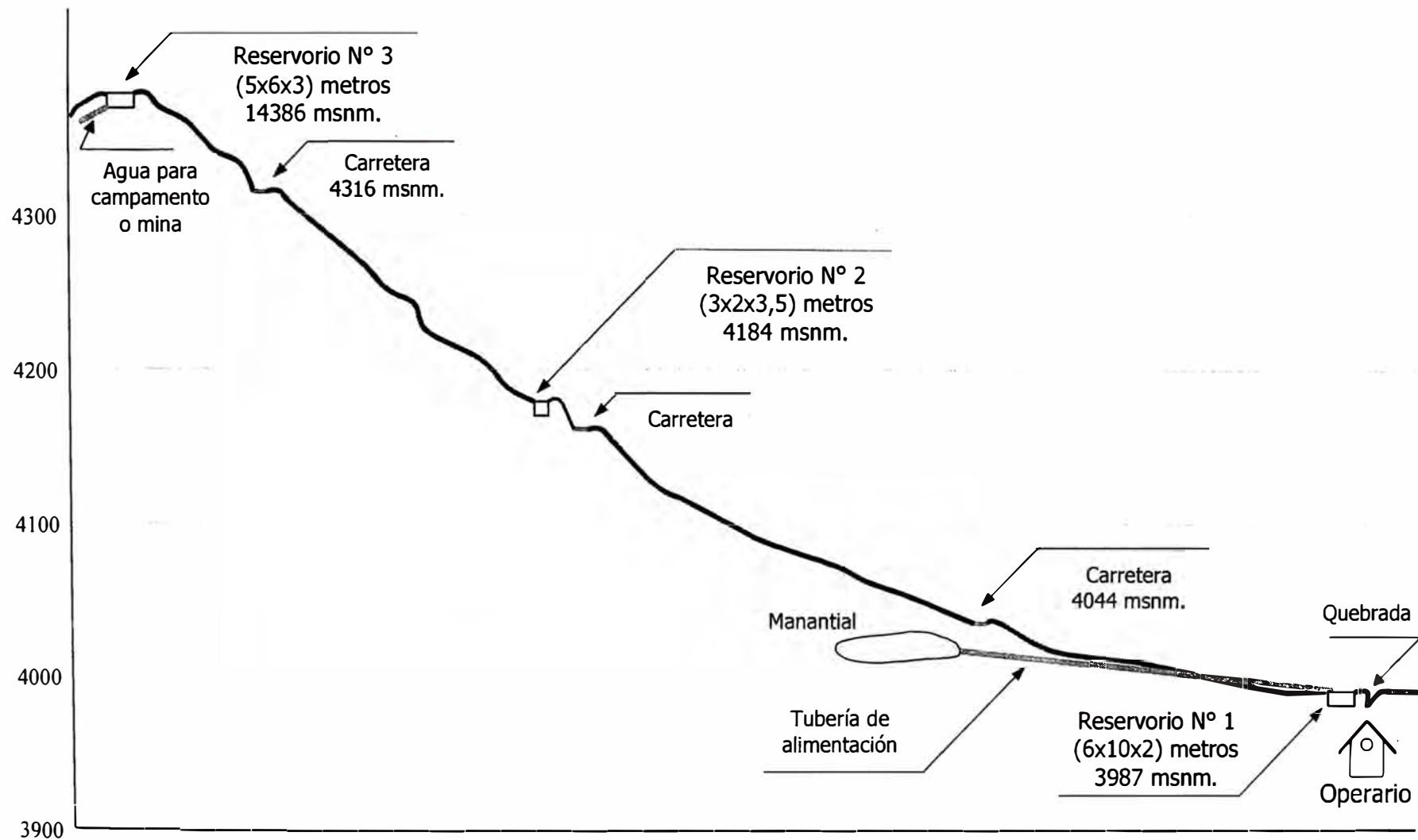


Figura 7.1 Ubicación física del sistema de bombeo de agua, el cual se encuentra en la localidad de Pampas de la provincia de Angaraes en el departamento de Huancavelica. Pertenece a la Unidad de Producción Julcani, de la Compañía de Minas Buenaventura S. A. A.

7.2. Funcionamiento del sistema

Al sistema instalado que se describió anteriormente, le realizaron la instalación de un pararrayo y el sistema de puesta a tierra cercana a los tableros eléctricos que controlan las electrobombas y además se amplió la línea telefónica hasta donde se encuentra el tablero de control de la electrobomba 2.

El sistema diseñado e implementado, es instalado en un armario y se ubica al lado del tablero eléctrico que controla el funcionamiento de la electrobomba 2. En el sistema así formado se han realizado las conexiones siguientes:

1. El terminal respectivo a la línea telefónica. El circuito puede estar conectado en cualquier punto de la línea. El circuito está preparado para compartir la línea con distintos teléfonos, pero no con otros como contestadores, fax, etc.
2. El circuito de Control de Nivel, a una de las salidas de mando (1, 2 o 3) del circuito de Telemando.
3. La Salida de Control_1 (contacto normalmente abierto del relé respectivo) para controlar el funcionamiento de la electrobomba 1; esta conexión se realiza desde esta salida hasta su tablero de control eléctrico, con par trenzado de una longitud de más de 1000 m.
4. La Salida de Control_2 (contacto normalmente abierto del relé respectivo) para controlar el funcionamiento de la electrobomba 2.
5. La entrada para detectar el funcionamiento o estado de otro dispositivo.
6. El terminal respectivo a la alimentación de 220 V A.C.

Una vez realizado la instalación del sistema para controlar el funcionamiento de las electrobombas (ver figura 7.2), el sistema de bombeo de agua puede ser controlado por funcionamiento manual o funcionamiento automático.

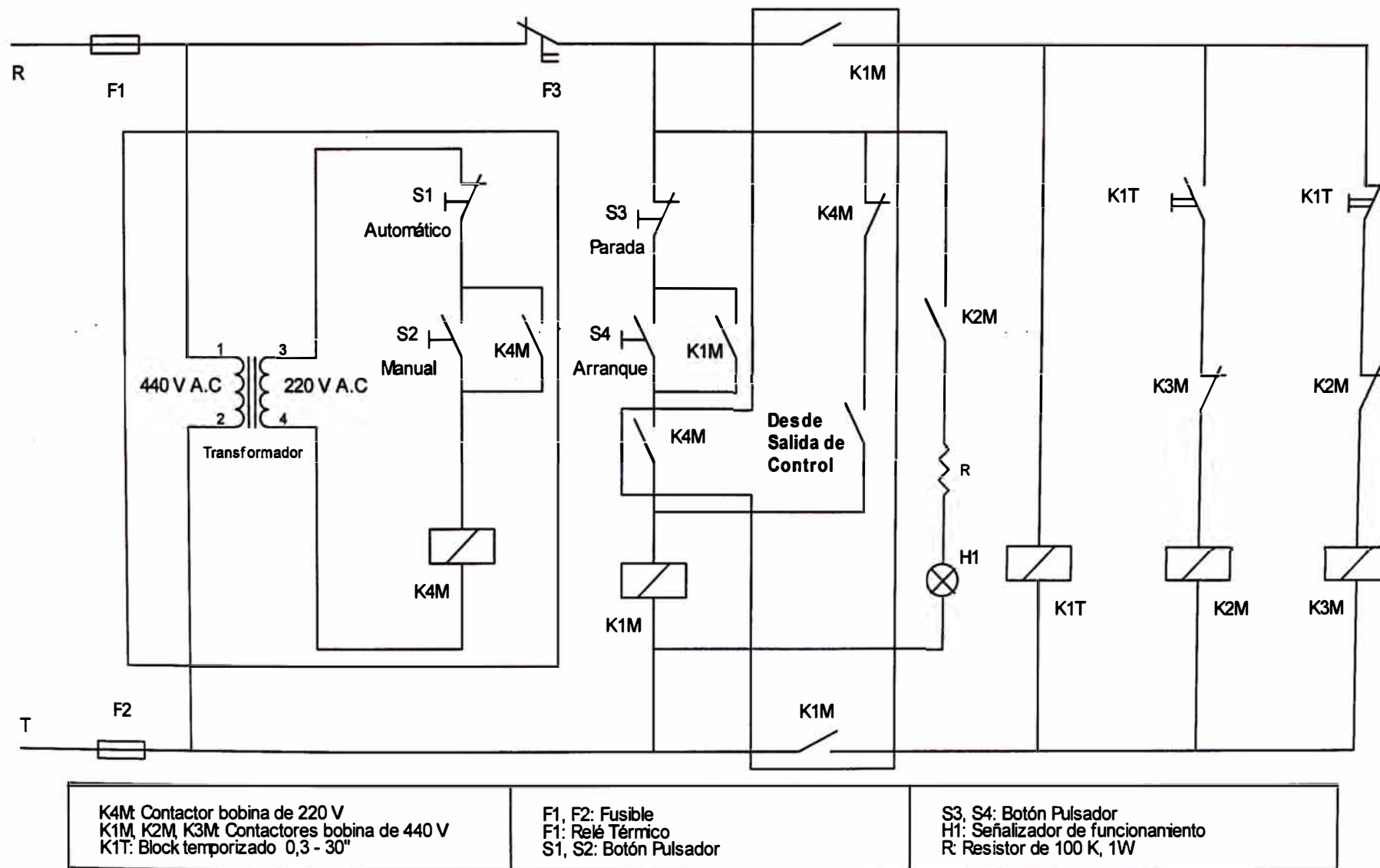


Figura 7.2 Diagrama de Control: Arrancador Y-Δ, 100 HP o 75 HP, 440 V, 3φ, 60Hz. También conexión de la salida de control.

7.2.1. Funcionamiento manual

El funcionamiento de las electrobombas en horarios establecidos o según requerimiento, lo controla manualmente un operario (solo en este caso activar el pulsador de funcionamiento manual, ver figura 7.2), tal como sigue: llegado el horario o según requerimiento de agua, el operario enciende la electrobomba 1, luego el operario se dirige al reservorio N° 2, que está a una distancia de 1000 m., tardando aproximadamente 25 minutos en llegar y encender la electrobomba 2. Según requerimiento, primero se tiene que avisar al operario, retrasando unos 25 minutos más su funcionamiento. Una vez encendida la electrobomba 2, el operario controla manualmente su funcionamiento; encendiéndola al alcanzar su nivel máximo el reservorio N° 2 o apagándola al alcanzar su nivel mínimo, repitiéndose este proceso durante el horario establecido o durante el tiempo requerido.

Este tipo de funcionamiento solo debe ser aplicado en casos de fuerza mayor.

7.2.2. Funcionamiento automático

En este caso se presenta la operación manual o control por teléfono, ambos casos utilizan los circuitos implementados en el desarrollo del proyecto de tesis propuesto.

El circuito de Telemando; realiza el control a distancia vía teléfono de hasta 3 cargas o aparatos diferentes de una intensidad máxima de 10 A (en una de las cuales se encuentra el circuito de Control de Nivel). Dispone además de una entrada de contacto. También permite comprobar el estado de algún dispositivo, incluso desde la otra parte del mundo o localmente a través de los led's respectivos.

El control a distancia esta diseñado para operar con las teclas de cualquier teléfono, siempre que este sea del tipo digital (aquellos que producen un beep

diferente con cada uno de los números o signos). El circuito genera unas señales de tipo sonoro para informar que el dispositivo determinado (que se encuentra instalado en una de las tres salidas de mando o en la entrada) ha sido conectado (ON) o desconectada (OFF). La operación está protegida por un código definido por el usuario, que es activado mediante los números de las teclas del aparato telefónico. Cuando ninguna tecla es pulsada dentro de 20 segundos, el sistema se desconecta automáticamente de la línea telefónica. Así mismo; incluye operación manual para conectar o desconectar las salidas sin recurrir a la llamada telefónica.

El circuito de Control de Nivel se encarga de automatizar el sistema de bombeo de agua, controlando el nivel mínimo y nivel máximo de un reservorio (enciende o apaga las electrobombas). El sensor no presenta ningún contacto físico con el líquido y es conectado por una delgada manguera en el lado de presión (P1) hasta el nivel tope mínimo de nivel de agua, para no dañar la electrobomba.

Los valores de las variables (Unidad de medida, Nivel máximo, Nivel mínimo y Nivel tope) son configurados ingresando a un MENU (se muestran en la pantalla LCD), accionando (producen un beep) los pulsadores P1, P2, P3 y E, y si los valores de las variables configuradas no son válidos se obliga a ser ingresadas nuevamente; así mismo se obtiene el nivel offset (la manguera que conecta al sensor debe encontrarse fuera del reservorio), se muestra los valores finales de las variables con que el sistema operará y, finalmente se graba estos valores en la memoria EEPROM.

A. Operación manual

En este modo de operación, para realizar el control del sistema de bombeo agua, se debe seguir los pasos siguientes:

1. Se conecta el circuito de Control de Nivel manualmente mediante el pulsador llamado operación manual. Cuando es conectada, el led correspondiente se encenderá (correspondiente a la salida de mando). Para conectarla accione el pulsador el número de veces que señala la salida de mando (1, 2 o 3) y nuevamente el mismo número de veces, si desea desconectarla.
2. Si se enciende el sistema por primera vez, los valores de las variables deben ser configurados necesariamente (se pueden configurar en otra ocasión); esto se realiza solo localmente. Accionando los pulsadores P1, P2, P3 y E; se muestra dinámicamente en la pantalla LCD mensajes y valores, tal como se detalla:

Ingresar a MENU: Realizar Reset y mantener presionado P1+P2+P3.

Ingresar a elegir unidad: Presentado el MENU, se pulsa E. Pulsando P3 se puede cambiar la unidad de forma cíclica entre m. y cm. Para continuar pulsar E.

Ingresar a configurando: Pulsar P1+P2. Aquí se ingresan los valores deseados de las variables Nivel máximo (NivelMAX), Nivel mínimo (NivelMIN), Nivel tope mínimo (NivelTOP). Pulsando P1 se selecciona el dígito y pulsando P2 se incrementa el dígito seleccionado. Si no se desea variar el valor de una variable o se desea continuar pulsar E. Si algún valor no es válido se muestra un mensaje y se retorna al inicio de configurando.

Ingresar a obtener Nivel offset: Se presenta luego de pulsar E al concluir satisfactoriamente configurando, teniendo como condición de que la manguera que se encuentra conectado al sensor se encuentre fuera del reservorio o tanque. Es aceptado pulsando P1+P3 y no es aceptado o se continua pulsando E.

Ingresar a observar los valores actuales: se presenta al pulsar E luego de concluido lo anterior. Muestra las variables (Unidad_M, NivelMAX, NivelMIN,

NivelTOP y NivelOFF) y sus respectivos valores secuencialmente, al pulsar E. Finalizado y al pulsar E se sale del MENU; pasando al monitoreo y procesamiento del nivel de agua, mostrándose en la primera línea de la pantalla LCD el nivel actual de agua, según formato establecido y en la segunda línea, el estado de las electrobombas.

3. Una vez conectado el circuito de Control de Nivel (caso en que no se ingresa al MENU para configurar el sistema), en la pantalla LCD se muestra un mensaje de inicio; luego se pasa al monitoreo y procesamiento del nivel de agua, mostrándose en la primera línea del LCD el nivel actual de agua, según formato establecido y en la segunda línea, el estado de las electrobombas.

B. Control por teléfono

Este modo de funcionamiento solo se utiliza para conectar o desconectar el circuito de Control de Nivel; se debe seguir los pasos siguientes:

1. Verifique la conexión a la línea telefónica.
2. Llame al número de teléfono (o anexo) en el que se encuentra conectado el sistema implementado.

Después de tres u ocho timbres el sistema contestará. Unos segundos después podrá escuchar estado de la entrada y salidas de mando. Primero la entrada, después la Salida_1, Salida_2 y Salida_3. Un sonido de corta duración, señala desconectado (OFF) y dos sonidos de corta duración, señala conectado (ON). Si desea volver a escuchar la información del estado de entrada y salidas; introduzca el código de seguridad y después "00". Puede repetir este proceso cuantas veces desee.

3. Establecida la comunicación telefónica; introduzca el código de seguridad seguido del número de la salida donde se encuentra conectado el circuito de Control de Nivel e inmediatamente “1”, para conectar el circuito de Control de Nivel. Después de unos segundos volverá a escuchar la situación de entrada y salidas actualizadas.

Para desconectar el circuito de Control de Nivel; introduzca el código de seguridad seguido del número de la salida donde se encuentra conectado el circuito de Control de Nivel e inmediatamente “0”. Después de unos segundos volverá a escuchar la situación de entrada y salidas actualizadas.

Si se conecta el circuito de Control de Nivel; en el remoto: la pantalla LCD muestra un mensaje de inicio; luego el sistema pasa a monitorear y procesar del nivel de agua, mostrándose en la primera línea del LCD el nivel actual de agua, según formato establecido y en la segunda línea, el estado de las electrobombas.

7.3. Especificaciones técnicas

Del circuito de Telemando

- Tres salidas de mando a relé de 12 V D.C/10A, con indicadores led; que se encuentran protegidas contra interferencias electromagnéticas.
- Una entrada de contacto con indicador led.
- Número de timbradas a elegir entre tres (3) u ocho (8).
- Informe del estado de entrada y salidas inmediato a la recepción, en el siguiente orden: Entrada, Salida_1, Salida_2 y Salida_3. El informe del estado se realiza de forma audible, tal como se indica:

Un sonido de corta duración, señala entrada o salida desconectada (OFF).

Dos sonidos de corta duración, señala entrada o salida conectada (ON).

- Un sonido de larga duración, señala código no válido o señala desconexión de la línea telefónica.
- Desconexión automática de la línea telefónica si no se pulsa ninguna tecla dentro de 20 segundos.
- Código de seguridad programable del 00 al 99.
- Tres opciones de ingreso de código de seguridad.
- Opción de operación manual.
- Alimentación 12 V D.C y 5 V D.C / 250 mA.

Del circuito de Control de Nivel

- Dos salidas de control a relé de 12 V D.C/10A, con indicadores led; que se encuentran protegidas contra interferencias electromagnéticas.
- Cuatro pulsadores, que emiten un beep al ser accionados. Los que se utilizan conjuntamente con la pantalla LCD para configurar el sistema.
- En funcionamiento normal monitorea el nivel de agua del reservorio y lo muestra en la pantalla LCD, señalando el estado de las electrobombas.
- Contraste variable de la pantalla LCD.
- Alimentación 12 V D.C y 5 V D.C / 200 mA.

7.4. Manual del usuario

El circuito de Telemando, realiza el control a distancia vía teléfono, protegido con código de seguridad; permitiendo controlar el circuito de Control de Nivel. También

permite comprobar el estado de algún dispositivo, incluso desde la otra parte del mundo.

El control a distancia está diseñado para operar con las teclas de cualquier teléfono, siempre que este sea del tipo DTMF. El circuito genera unas señales de tipo sonoro para informar que el dispositivo ha sido conectado (ON) o desconectada (OFF). Cuando ninguna tecla es pulsada dentro de 20 segundos, el circuito se desconecta automáticamente de la línea telefónica. Así mismo; tiene la opción de operación manual, en caso se requiera.

El circuito de Control de Nivel se encarga de automatizar el sistema de bombeo de agua, controlando el nivel mínimo y nivel máximo de un reservorio. El sensor no presenta ningún contacto físico con el líquido y es conectado por una delgada manguera en el lado de presión (P1) hasta el nivel tope mínimo en que se debe encontrar el nivel de agua. Los valores de las variables (unidad de medida, nivel máximo, nivel mínimo y nivel tope) son configurados ingresando a un MENU, accionando los pulsadores P1, P2, P3 y E (se muestra en el LCD), si los valores no son válidos se muestra un mensaje y se obliga a ser ingresados nuevamente; así mismo se obtiene el nivel offset (la manguera que conecta al sensor debe estar fuera del reservorio), se muestra los valores finales de las variables con que el sistema operará, y finalmente se graba estos valores en la memoria EEPROM.

En funcionamiento normal el sistema monitorea el nivel de agua existente en el reservorio y lo muestra en la pantalla LCD señalando el estado de las electrobombas.

Conexión del sistema

- Conecte el terminal respectivo del circuito de Telemando a la línea telefónica.

- Conecte el circuito de Control de Nivel, a una de las tres salidas de mando del circuito de Telemando. Las salidas de mando son contactos N.A de un relé.
- Conecte la Salida de Control_1 (contacto normalmente abierto) para controlar el funcionamiento de la electrobomba 1.
- Conecte la Salida de Control_2 (contacto normalmente abierto) para controlar el funcionamiento de la electrobomba 2.
- La entrada para detectar el funcionamiento o estado de otro dispositivo.
- Luego conecte el terminal respectivo a la alimentación de 220 V A.C.

Operación manual

En este modo de funcionamiento, para realizar el control del sistema de bombeo agua, se debe seguir los pasos siguientes:

- Se conecta el circuito de Control de Nivel manualmente mediante el pulsador llamado operación manual. Para conectarla accione el pulsador el número de veces que señala la salida de mando (1, 2 o 3) y nuevamente el mismo número de veces, si desea desconectarla. Cuando es conectada, el led correspondiente se encenderá.
- Si se enciende el sistema por primera vez, los valores de las variables deben ser configurados necesariamente (se pueden configurar en otra ocasión); esto se realiza solo localmente. Accionando los pulsadores P1, P2, P3 y E; se muestra dinámicamente en la pantalla LCD mensajes y valores, tal como se detalla:

Ingresar a MENU: Realizar Reset y mantener presionado P1+P2+P3.

Ingresar a elegir unidad: Presentado el MENU, se pulsa E. Pulsando P3 se puede cambiar la unidad de forma cíclica entre m. y cm. Para continuar pulsar E.

- Ingresar a configurando: Pulsar P1+P2. Aquí se ingresan los valores deseados de las variables Nivel máximo (NivelMAX), Nivel mínimo (NivelMIN), Nivel tope mínimo (NivelTOP). Pulsando P1 se selecciona el dígito y pulsando P2 se incrementa el dígito seleccionado. Si no se desea variar el valor de una variable o se desea continuar pulsar E. Si algún valor no es válido se muestra un mensaje y se retorna al inicio de configurando.
- Ingresar a obtener Nivel offset: Se presenta luego de pulsar E al concluir satisfactoriamente configurando, teniendo como condición de que la manguera que se encuentra conectado al sensor se encuentre fuera del reservorio o tanque. Es aceptado pulsando P1+P3 y no es aceptado o se continua pulsando E.
- Ingresar a observar los valores actuales: se presenta al pulsar E luego de concluido lo anterior. Muestra las variables (Unidad_M, NivelMAX, NivelMIN, NivelTOP y NivelOFF) y sus respectivos valores secuencialmente al pulsar E. Finalizado y al pulsar E se sale del MENU de configuración; pasando al monitoreo y procesamiento del nivel de agua, mostrándose en la primera línea de la pantalla LCD el nivel actual de agua, según formato establecido y en la segunda línea, el estado de las electrobombas.
- Una vez conectado el circuito de Control de Nivel (caso en que no se ingresa al MENU para configurar el sistema), en la pantalla LCD se muestra un mensaje de inicio; luego se pasa al monitoreo y procesamiento del nivel de agua, mostrándose en la pantalla LCD, señalando el estado de las electrobombas.

Ajustando el número de timbradas: El circuito esta predeterminada para contestar la llamada después de 8 timbradas. Si por cualquier motivo se desea reducir esto a 3

timbradas, deberá mantener presionado el pulsador de operación manual al encender el equipo, quedando fijado el número de timbradas en 3.

Definiendo el código de seguridad: El código de seguridad configurable se realiza en el 8-Dip-switch (unidad y decena). El valor esta comprendido entre 00 y 99.

Control por teléfono

Estableciendo la conexión

- Llame al número de teléfono (o anexo) al que esta conectado el sistema (circuito de Telemando, circuito de Control de nivel, etc.).
- Después de tres u ocho timbres el circuito de Telemando contestará.
Unos segundos después podrá escuchar estado de la entrada y salidas de mando. Primero la entrada de contacto, después la Salida_1, Salida_2 y Salida_3. Un sonido de corta duración, desconectada (OFF) y dos sonidos de corta duración, señala conectada (ON).

Recuperando la información: Introduzca el código de seguridad y después “00”. Lo que permite volver a escuchar la información del estado de entrada y salidas. Puede repetir este proceso cuantas veces desee.

Conectando el circuito de Control de Nivel: Introduzca el código de seguridad seguido del número de la salida donde se encuentra conectado el circuito de Control de Nivel e inmediatamente “1”. Después de unos segundos volverá a escuchar la situación de entrada y salidas actualizadas.

Por ejemplo si el código de seguridad es 57 y como la salida es “1”. Deberá pulsar solo una vez, las siguientes teclas: 5-7-1-1.

Desconectando el circuito de Control de Nivel: Introduzca el código de seguridad seguido del número de la salida donde se encuentra conectado el circuito de Control de Nivel e inmediatamente “0”. Después de unos segundos volverá a escuchar la situación de entrada y salidas actualizadas.

Por ejemplo si el código de seguridad es 57 y la salida es “1”. Deberá pulsar solo una vez, las siguientes teclas: 5-7-1-0.

En el caso que se conecta el circuito de Control de Nivel; en el remoto: la pantalla LCD muestra un mensaje de inicio; luego el circuito de Control de Nivel pasa a monitorear y procesar el nivel de agua, mostrando en la primera línea del LCD el nivel actual de agua, según formato establecido y en la segunda línea, se muestra el estado de las electrobombas.

CAPÍTULO VIII COSTOS Y PRESUPUESTO DEL SISTEMA IMPLEMENTADO

8.1. Costos

El proceso de diseño e implementación del sistema propuesto ha tenido un costo relativamente alto. Se ha requerido de una serie de equipos, instrumentos, herramientas y accesorios, muchos de ellos difíciles de encontrar en el mercado local. El tiempo invertido es también considerable, teniendo cuenta la escasa, difícil y lenta obtención de información referente al tema. Todo el sistema se ha desarrollado en un lapso de dos años cronológicos.

Se debe señalar también que el sistema de puesta a tierra fue costeado e instalado en su totalidad por la Unidad de Producción Julcani.

En las tablas siguientes se detalla los recursos empleados para llevar a cabo el proyecto, según la naturaleza del recurso como: equipos/instrumentos, herramientas, accesorios/componentes electrónicos, software y bibliografía impresa.

Equipo/Instrumento	Cantidad	Costo unitario (US \$)	Costo total (US \$)
Computadora Personal	1	800	800
Impresora Láser	1	600	600
Programador de PIC's	1	350	350
Osciloscopio*	1	1000	1000
Multímetro digital	1	100	100
Pinza amperimétrica*	1	100	100
Aparatos telefónicos*	2	10	20
SUBTOTAL:			2 970

Tabla 8.1 *Costo de equipos e instrumentos empleados en desarrollo del sistema.*

Herramienta	Cantidad	Costo unitario (US \$)	Costo total (US \$)
Pelador de cable	1	10	10
Alicates	1	10	10
Destornilladores	1	12	10
Pistola de soldar	1	50	50
SUBTOTAL:			82

Tabla 8.2 Costo de herramientas empleados en desarrollo del sistema.

Accesorio/Componente	Cantidad	Costo unitario (US \$)	Costo total (US \$)
Varistor	10	0,5	5
Condensador de alto voltaje	10	0,5	5
Fusible y porta fusible	6	1	6
Barra de bronce pequeño	1	5	5
Tarro de Bamiz	1	20	20
Tarro de Cemento plástico	1	10	10
Otros accesorios:* Alambre esmaltado, Chasis, etc.	---	---	25
SUBTOTAL:			76

Tabla 8.3 Accesorios/Componentes empleados en el circuito de Protección de Entrada.

Accesorio/Componente	Cantidad	Costo unitario (US \$)	Costo total (US \$)
Reguladores de voltaje	4	0,5	2
Varistor	1	0,5	0,5
Diodos rectificadores	5	0,1	0,5
Condensadores de A.C.	3	1,5	4,5
Condensadores de 1mF/25V	2	0,5	1
Condensadores de 10uF/25V	4	0,25	1
Condensadores de 0,1uF	5	0,1	0,5
Choque en modo común	1	4	4
Fusible, diodo led, resistencia	1	1	1
Interruptor de power On/OFF	1	1	1
Transformador	1	3	3
Conectores	2	1	2
Tarjeta de circuito impreso	1	25	25
SUBTOTAL:			46

Tabla 8.4 Accesorios/Componentes empleados en el circuito de la Fuente de Alimentación.

Accesorio/Componente	Cantidad	Costo unitario (US \$)	Costo total (US \$)
Protoboard	3	5	15
PIC16C55-XT/P	2	7,5	15
M-8870	2	3,5	7
4N25	1	1	1
Cristal	1	1,5	1,5
Transistores	5	0,4	2
Darlington	1	1	1
Diodos led's	5	0,1	0,5
Diodos rectificadores	10	0,1	1
Diodos zener	5	0,5	2,5
Condensadores electrolíticos	4	0,25	1
Condensador de ,47uF/250V	1	1	1
Condensadores cerámicos	10	0.1	1
Condensadores de 10nF/1KV	3	0,5	1,5
Resistencia integrada	1	0,5	0,5
Resistencias	30	0.05	1,5
Relés	4	1	4
Dip-switches	1	1	1
Pulsador e interruptor	1	2	2
Transformador de 600Ω/600Ω	1	1,5	1,5
Conectores	4	1	4
Cable telefónico por metro	2	2,5	5
Varistores	3	0,5	1,5
Tarjeta de circuito impreso	1	25	25
Otros*	---	---	30
SUBTOTAL:			127

Tabla 8.5 *Accesorios/Componentes empleados en el circuito de Telemando.*

Accesorio/Componente	Cantidad	Costo unitario (US \$)	Costo total (US \$)
Protoboard	3	5	15
PIC16F876-20/SP	2	25	25
MPX2100	2	30	60
AD623	2	3,5	7
Pantalla LCD	1	10	10
Cristal	1	1,5	1,5
Transistores	5	0,4	2
Buzzer	1	1,5	1,5

Tabla 8.6 *Accesorios/Componentes empleados en el esquema electrónico del circuito de Control de Nivel.*

Accesorio/Componente	Cantidad	Costo unitario (US \$)	Costo total (US \$)
Relés	2	1	2
Diodos	5	0,1	0,5
Diodos led	5	0,1	0,5
Condensadores de tantalio	4	0,5	2
Condensadores de 10uF/25V	4	0,25	1
Condensadores cerámicos	15	0,1	1,5
Potenciómetros de precisión	2	1,5	3
Potenciómetro trimer	1	0,5	0,5
Resistencias	20	0,05	1
Pulsadores	5	1	5
Conectores	4	1	4
Varistores	2	0,5	1
Tarjeta de circuito impreso	1	25	25
Otros*	---	---	25
SUBTOTAL:			194

Tabla 8.7 *Accesorios/Componentes empleados en el circuito de Control de Nivel (continuación).*

Bibliografía/Software	Cantidad	Costo unitario (US \$)	Costo total (US \$)
Libros	3	25	75
Revistas y fotocopias	---	50	50
Días de Internet (2 h/día)	200	1	200
Software EDA	---	150	150
Software varios	---	50	50
SUBTOTAL:			525

Tabla 8.8 *Costos de bibliografía y software utilizados en el desarrollo del sistema.*

Subtotales	Costo (US \$)
Equipos e instrumentos	2970
Herramientas	82
Accesorios/componentes	443
Bibliografía/software	525
TOTAL GENERAL:	4020

Tabla 8.9 *Subtotales y costo total realizado en el desarrollo del Sistema de Telemando y Automatización de Bombeo de Agua Basado en Microcontroladores PIC.*

Cabe anotar que los costos mencionados son en algunos casos aproximados (e incluso subvaluados) debido a la dificultad de cuantificar u obtener precios exactos. Esto ocurre, por ejemplo, con los programas EDA cuyos costos verdaderos son bastante elevados. Para la realización del proyecto se ha utilizado versiones un tanto antiguas o de libre distribución obtenidas por Internet o directamente a través de las mismas compañías que proporcionan facilidades para uso académico.

(*) Indica que estos son de propiedad o que fueron asumidos por la Unidad de Producción Julcani y fueron utilizados por el autor.

8.2. Presupuesto

Este trabajo ha sido financiado con el generoso apoyo de la Unidad de Producción Julcani y prácticamente en su integridad con recursos económicos del autor.

	Presupuesto	Porcentaje
Unidad de Producción Julcani	\$ 1 200	29,85 %
Recursos propios del autor	\$ 2 820	70,15 %

Tabla 8.10 *Presupuesto para el desarrollo del sistema.*

CONCLUSIONES

1. Es posible realizar telemando basado en el PIC16C55, vía línea telefónica, controlando el ON/OFF de cualquiera de las salidas de mando (relés), desde cualquier teléfono con marcación por tonos, sin importar la distancia.
2. El circuito de Control de Nivel, basado en el PIC16F876, automatiza el sistema de bombeo de agua controlando los niveles máximo y mínimo. El nivel es monitoreado y mostrado en pantalla LCD.
3. Los circuitos de protección contra EMI (principalmente descargas atmosféricas), de entrada, de salida de mando o control, cumplen su función óptimamente, siempre que el sistema de puesta a tierra sea de buena calidad.
4. Las perturbaciones causadas por el funcionamiento de las electrobombas de gran potencia, no altera el funcionamiento normal de los circuitos del sistema.
5. El consumo de energía es considerablemente bajo (del orden de miliamperios) y se debe principalmente al consumo de las bobinas de los relés.
6. El transductor de presión no presenta problemas de oxidación debido a que no tiene ningún contacto físico con el agua.
7. Debido a que los dispositivos trabajan a temperaturas inferiores a 0 °C, se utilizaron componentes de uso industrial.
8. Se recomienda el mantenimiento de la empaquetadura que esta ubicado en el cuello de la electrobomba, en un periodo de seis meses.

ANEXOS

ANEXO A M-8870 - DTMF RECEIVER



M-8870 DTMF Receiver

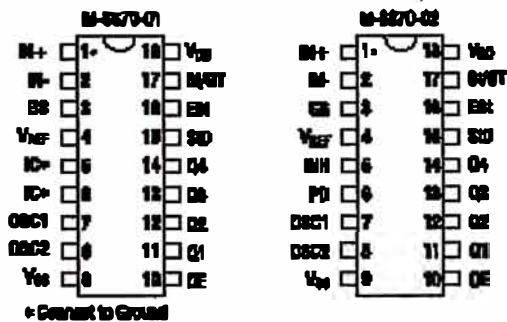
Features

- Low Power Consumption
- Adjustable Acquisition and Release Times
- Central Office Quality and Performance
- Power-down and Inhibit Modes (-02 only)
- Inexpensive 3.58 MHz Time Base
- Single 5 Volt Power Supply
- Dial Tone Suppression

Applications

- Telephone switch equipment
- Remote data entry
- Paging systems
- Personal computers
- Credit card systems

Pin Configuration



Description

The M-8870 is a full DTMF Receiver that integrates both bandsplit filter and decoder functions into a single 18-pin DIP

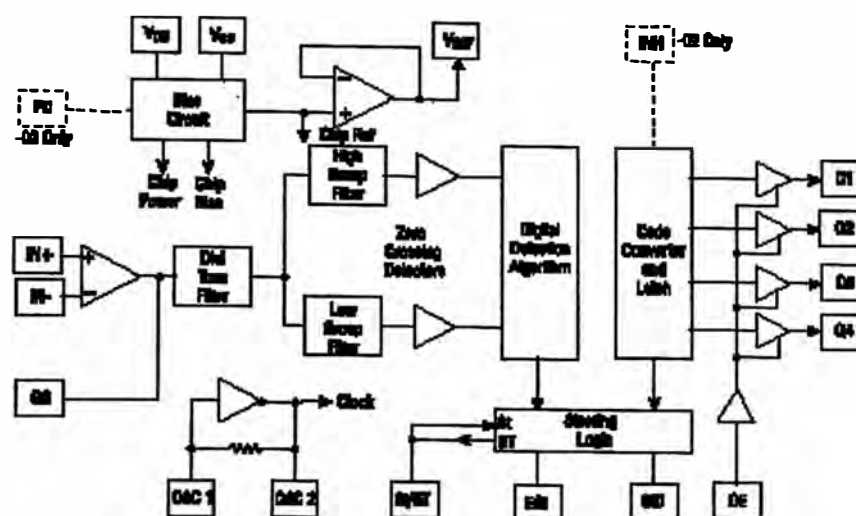
or SOIC package. Manufactured using CMOS process technology, the M-8870 offers low power consumption (35 mW max) and precise data handling. Its filter section uses switched capacitor technology for both the high and low group filters and for dial tone rejection. Its decoder uses digital counting techniques to detect and decode all 16 DTMF tone pairs into a 4-bit code. External component count is minimized by provision of an on-chip differential input amplifier, clock generator, and latched tri-state interface bus. Minimal external components required include a low-cost 3.579545 MHz color burst crystal, a timing resistor, and a timing capacitor.

The M-8870-02 provides a "power-down" option which, when enabled, drops consumption to less than 0.5 mW. The M8870-02 can also inhibit the decoding of fourth column digits (see Tone Decoding table on page 5).

Ordering Information

Part #	Description
M-8870-01	18-pin plastic DIP
M-8870-01SM	18-pin plastic SOIC
M-8870-01SMTR	18-pin plastic SOIC, tape and reel
M-8870-02	18-pin plastic DIP, power-down, option
M-8870-02SM	18-pin plastic SOIC, power-down, option
M-8870-02T	18-pin plastic SOIC, power-down option, tape and reel

Block Diagram



Absolute Maximum Ratings

Parameter	Symbol	Value
Power supply voltage ($V_{DD} - V_{SS}$)	V_{DD}	6.0 V max
Voltage on any pin	V_{DC}	$V_{SS} - 0.3, V_{DD} + 0.3$
Current on any pin	I_{DD}	10 mA max
Operating temperature	T_A	-40°C to +85°C
Storage temperature	T_S	-65°C to +150°C

Absolute Maximum Ratings are stress ratings. Stresses in excess of these ratings can cause permanent damage to the device. Functional operation of the device at these or any other conditions beyond those indicated in the operational sections of this data sheet is not implied. Exposure of the device to the absolute maximum ratings for an extended period may degrade the device and affect its reliability.

Note:
Exceeding these ratings may cause permanent damage. Functional operation under these conditions is not implied.

DC Characteristics

Parameter	Symbol	Min	Typ	Max	Units	Test Conditions
Operating supply voltage	V_{DD}	4.75	-	5.25	V	-
Operating supply current	I_{DD}	-	3.0	7.0	mA	-
Standby supply current (see Note 3)	I_{DDQ}	-	-	100	μ A	$PD = V_{DD}$
Power consumption	P_n	-	15	35	mW	$f = 3.579$ MHz, $V_{DD} = 5.0$ V
Low level input voltage	V_{IL}	-	-	1.5	V	-
High level input voltage	V_{IH}	3.5	-	-	V	-
Input leakage current	I_{IN}/I_L	-	0.1	-	μ A	$V_{IN} = V_{SS}$ or V_{DD} (see Note 2)
Putup (source) current on OE	I_{SO}	-	6.5	15.0	μ A	OE = 0 V
Input impedance, signal inputs 1, 2	R_{IN}	8	10	-	k Ω	@ 1 kHz
Steering threshold voltage	V_{TSH}	2.2	-	2.5	V	-
Low level output voltage	V_{OL}	-	-	0.03	V	No load
High level output voltage	V_{OH}	$V_{DD} - 0.03$	-	-	V	No load
Output low (sink) current	I_{OL}	1.0	2.5	-	mA	$V_{OUT} = 0.4$ V
Output high (source) current	I_{OH}	0.4	0.8	-	mA	$V_{OUT} = V_{DD} - 0.4$ V
Output voltage V_{REF}	V_{REF}	2.4	-	2.7	V	No load
Output resistance V_{REF}	R_{OR}	-	10	-	k Ω	-

*Typical figures are at 25°C and are for design aid only; not guaranteed and not subject to production testing.

Operating Characteristics - Gain Setting Amplifier

Parameter	Symbol	Min	Typ	Max	Units	Test Conditions
Input leakage current	I_{IN}	-	± 100	-	nA	$V_{SS} < V_{IN} < V_{DD}$
Input resistance	R_{IN}	4	-	-	M Ω	-
Input offset voltage	V_{OS}	-	± 25	-	mV	-
Power supply rejection	PSRR	50	-	-	dB	1 KHz
Common mode rejection	CMRR	55	-	-	dB	$-3.0V < V_{CM} < 3.0V$
DC open loop voltage gain	A_{VOL}	60	-	-	dB	-
Open loop unity gain bandwidth	f_c	1.2	1.5	-	MHz	-
Output voltage swing	$V_{O,P}$	3.5	-	-	V _{pp}	$R_L \approx 100\text{ K}\Omega$ to V_{SS}
Tolerable capacitive load (GS)	C_L	-	-	100	pF	-
Tolerable resistive load (GS)	R_L	-	-	50	k Ω	-
Common mode range	V_{CM}	2.5	-	-	V _{r,p}	No load

*Typical figures are at 25°C and are for design aid only; not guaranteed and not subject to production testing.

Notes:

1. All voltages referenced to V_{SS} unless otherwise noted. For typical values, $V_{DD} = 5.0V$, $V_{SS} = 0V$, $T_A = 25^\circ C$.

Description

M-8870 operating functions (see block diagram on page 1) include a bandsplit filter that separates the high and low tones of the received pair, and a digital decoder that verifies both the frequency and duration of the received tones before passing the resulting 4-bit code to the output bus.

Filter

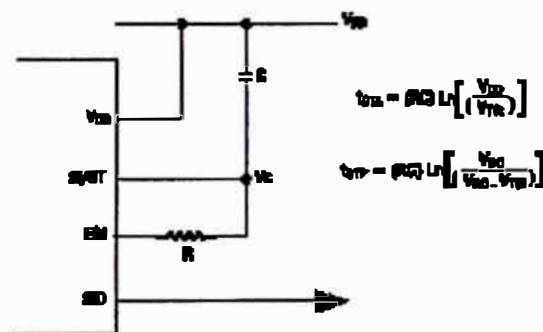
The low and high group tones are separated by applying the dual-tone signal to the inputs of two 6th order switched capacitor bandpass filters with bandwidths that correspond to the bands enclosing the low and high group tones. The filter also incorporates notches at 350 and 440 Hz, providing excellent dial tone rejection. Each filter output is followed by a single-order switched capacitor section that smooths the signals prior to limiting. Signal limiting is performed by high-gain comparators provided with hysteresis to prevent detection of unwanted low-level signals and noise. The comparator outputs provide full-rail logic swings at the frequencies of the incoming tones.

Decoder

The M-8870 decoder uses a digital counting technique to determine the frequencies of the limited tones and to verify that they correspond to standard DTMF frequencies. A complex averaging

algorithm is used to protect against tone simulation by extraneous signals (such as voice) while tolerating small frequency variations. The algorithm ensures an optimum combination of immunity to talkoff and tolerance to interfering signals (third tones) and noise. When the detector recognizes the simultaneous presence of two valid tones (known as signal condition), it raises the Early Steering flag (EST). Any subsequent loss of signal condition will cause EST to fall.

Basic Steering Circuit Functional

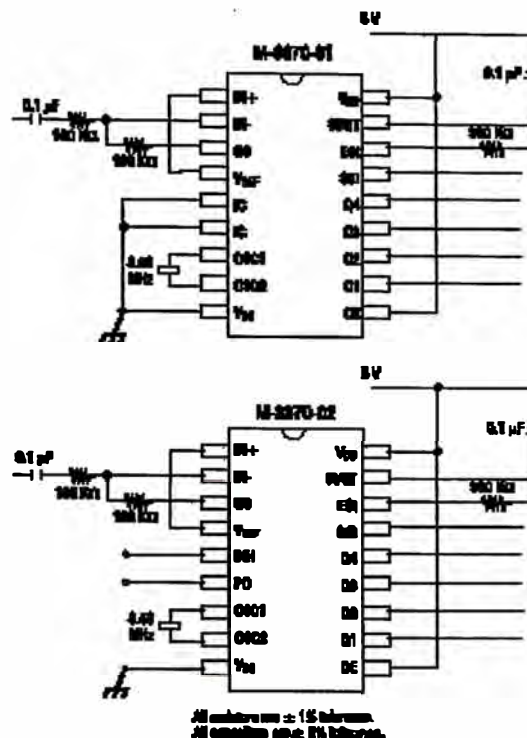


Steering Circuit

Before a decoded tone pair is registered, the receiver checks for a valid signal duration (referred to as character-recognition-condition). This check is performed by an external RC time constant driven by EST. A logic high on EST causes V_C (see block diagram on page 2) to rise as the capacitor discharges. Provided that signal condition is maintained (EST remains high) for the

validation period (t_{GT}), V_C reaches the threshold (V_{Tst}) of the steering logic to register the tone pair, thus latching its corresponding 4-bit code (see DC Characteristics on page 2) into the output latch. At this point, the GT output is activated and drives V_C to V_{DD} . GT continues to drive high as long as EST remains high. Finally, after a short delay to allow the output latch to settle, the delayed steering output flag (StD) goes high, signaling that a received tone pair has been registered. The contents of the output latch are made available on the 4-bit output bus by raising the three-state control input (OE) to a logic high. The steering circuit works in reverse to validate the interdigit pause between signals. Thus, as well as rejecting signals too short to be considered valid, the receiver will tolerate signal interruptions (dropouts) too short to be considered a valid pause. This capability, together with the ability to select the steering time constants externally, allows the designer to tailor performance to meet a wide variety of system requirements.

Single-Ended Input Configuration



Pin Functions

Pin	Name	Description
1	IN+	Non-inverting input
2	IN-	Inverting input
3	GS	Gain select. Gives access to output of front-end amplifier for connection of feedback resistor.
4	V_{REF}	Reference voltage output (nominally $V_{DD}/2$). May be used to bias the inputs at mid-rail.
5	INH*	Inhibits detection of tones representing keys A, B, C, and D.
6	PD*	Power down. Logic high powers down the device and inhibits the oscillator. Internal pulldown.
7	OSC1	Clock input
8	OSC2	Clock output
9	VSS	Negative power supply (normally connected to 0 V).
10	OE	Tri-statable output enable (input). Logic high enables the outputs Q1 - Q4. Internal pullup.
11-14	Q1, Q2, Q3, Q4	Tri-statable data outputs. When enabled by OE, provides the code corresponding to the last valid tone pair received (see Tone Decoding table on page 5).
15	StD	Delayed steering output. Presents a logic high when a received tone pair has been registered and the output latch is updated. Returns to logic low when the voltage on S/GT falls below V_{Tst} .
16	EST	Early steering output. Presents a logic high immediately when the digital algorithm detects a recognizable tone pair (signal condition). Any momentary loss of signal condition will cause EST to return to a logic low.
17	S/GT	Steering input/guard time output (bidirectional). A voltage greater than V_{Tst} detected at St causes the device to register the detected tone pair and update the output latch. A voltage less than V_{Tst} frees the device to accept a new tone pair. The GT output acts to reset the external steering time constant, and its state is a function of EST and the voltage on St. (See Common Crystal Connection on page 5).
18	V_{DD}	Positive power supply. (Normally connected to +5V.)

* -02 only. Connect to VSS for -01 version

Guard Time Adjustment

Where independent selection of signal duration and interdigit pause are not

required, the simple steering circuit of Basic Steering Circuit is applicable.

Component values are chosen according to the formula:

$$t_{REC} = t_{DP} + t_{GTP}$$

$$t_{GTP} @ 0.67 RC$$

The value of t_{DP} is a parameter of the device and t_{REC} is the minimum signal duration to be recognized by the receiver. A value for C of 0.1 μF is recommended for most applications, leaving R to be selected by the designer. For example, a suitable value of R for a t_{REC} of 40 ms would be 300 k. A typical circuit using this steering configuration is shown in the Single-Ended Input Configuration on page 4. The timing requirements for most telecommunication applications are satisfied with this circuit. Different steering arrangements may be used to select independently the guard times for tone-present (t_{GTP}) and tone-absent (t_{GTA}). This may be necessary to meet system specifications that place both accept and reject limits on both tone duration and interdigit pause.

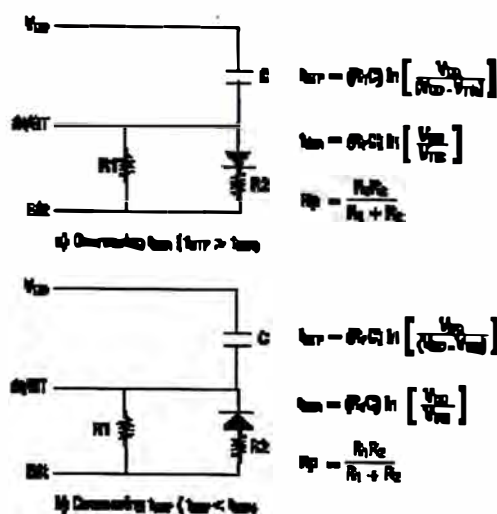
Guard time adjustment also allows the designer to tailor system parameters such as talkoff and noise immunity. Increasing t_{REC} improves talkoff performance, since it reduces the probability that tones simulated by speech will maintain signal condition long enough to be registered. On the other hand, a relatively short t_{REC} with a long t_{DO} would be appropriate for extremely noisy environments where fast acquisition time and immunity to dropouts would be required. Design information for guard time adjustment is shown in the Guard Time Adjustment below.

Power-down and Inhibit Mode (-02 only)

A logic high applied to pin 6 (PD) will place the device into standby mode to minimize power consumption. It stops the oscillator and the functioning of the filters. On the M-8870-01 models, this pin is tied to ground (logic low).

Inhibit mode is enabled by a logic high input to pin 5 (INH). It inhibits the detection of 1633 Hz. The output code will remain the same as the previous detected code (see Pin functions table on page 4). On the M-8870-01 models, this pin is tied to ground (logic low).

Figure 5 Guard Time Adjustment



Input Configuration

The input arrangement of the M-8870 provides a differential input operational amplifier as well as a bias source (V_{REF}) to bias the inputs at mid-rail. Provision is made for connection of a feedback resistor to the opamp output (GS) for gain adjustment.

In a single-ended configuration, the input pins are connected as shown in the Single-Ended Input Configuration on page 3 with the opamp connected for unity gain and V_{REF} biasing the input at $\frac{1}{2} V_{DD}$. The Differential Input Configuration below permits gain adjustment with the feedback resistor R5.

DTMF Clock Circuit

The internal clock circuit is completed with the addition of a standard 3.579545 MHz television color burst crystal. The crystal can be connected to a single M-8870 as shown in the Single-Ended Input Configuration on page 3, or to a series of M-8870s. As illustrated in the Common Crystal Connection below, a single crystal can be used to connect a series of M-8870s by coupling the oscillator output of each M-8870 through a 30 pF capacitor to the oscillator input of the next M-8870.

Tone Decoding

FLOW	FHIGH	Key (ref.)	OE	Q4	Q3	Q2	Q1
697	1209	1	H	0	0	0	1
697	1336	2	H	0	0	1	0
697	1477	3	H	0	0	1	1
770	1209	4	H	0	1	0	0
770	1336	5	H	0	1	0	1
770	1477	6	H	0	1	1	0
852	1209	7	H	0	1	1	1
852	1336	8	H	1	0	0	0
852	1477	9	H	1	0	0	1
941	1336	0	H	1	0	1	0
941	1209	S	H	1	0	1	1
941	1477	#	H	1	1	0	0
697	1633	A	H	1	1	0	1
770	1633	B	H	1	1	1	0
852	1633	C	H	1	1	1	1
941	1633	D	H	0	0	0	0
ANY	ANY	ANY	L	Z	Z	Z	Z

L = logic low, H = logic high, Z = high impedance

AC Characteristics

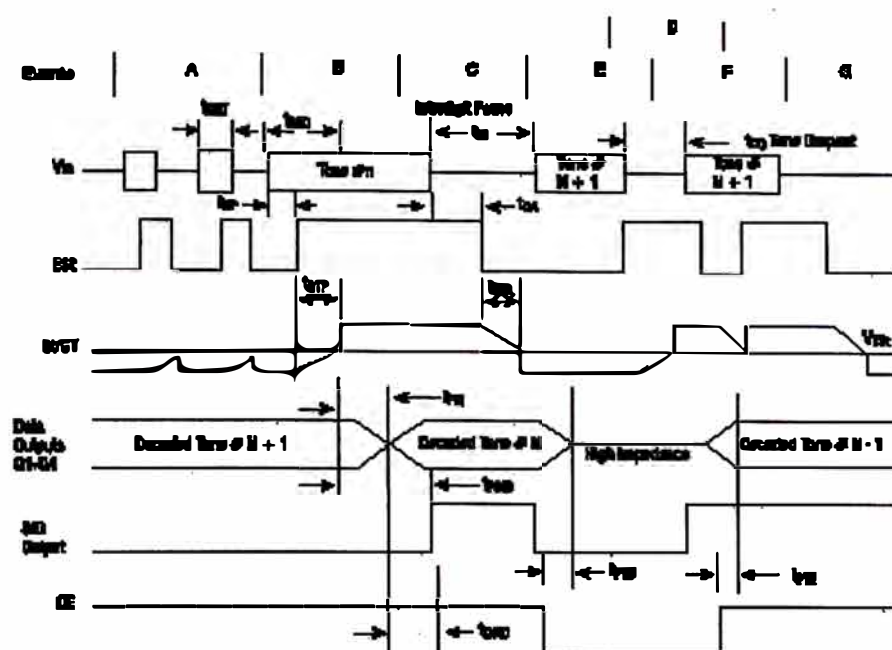
Parameter	Symbol	Min	Typ	Max	Units	Notes
Valid input signal levels (each tone of composite signal)	-	-29	-	+1	dBm	1,2,3,4,5,8
	-	27.5	-	869	mVRMS	
Positive twist accept	-	-	-	10	dB	2,3,4,8
Negative twist accept	-	-	-	10	dB	
Frequency deviation accept limit	-	-	-	$\pm 1.5\% + 2$ Hz	Nom.	2,3,5,8,10
Frequency deviation reject limit	-	$\pm 3.5\%$	-	-	Nom.	2,3,5
Third tone tolerance	-	-25	-16	-	dB	2,3,4,5,8,9,13,14
Noise tolerance	-	-	-12	-	dB	2,3,4,5,8,8,9
Dial tone tolerance	-	+18	+22	-	dB	2,3,4,5,7,8,9
Tone present detection time	t_{TP}	5	8	14	ms	See Timing Diagram on page 7
Tone absent detection time	t_{TA}	0.5	3	8.5	ms	
Minimum tone duration accept	t_{REF}	-	-	40	ms	User adjustable (see Basic Steering Circuit and Guard Time Adjustment on pages 3 and 4.)
Maximum tone duration reject	t_{RFF}	20	-	-	ms	
Minimum interdigit pause accept	t_{IP}	-	-	40	ms	
Maximum interdigit pause reject	t_{IPD}	20	-	-	ms	
Propagation delay (St to Q)	t_{STQ}	-	6	11	μ s	OE = VDD
Propagation delay (St to StD)	t_{SSD}	-	9	16	μ s	
Output data setup (O to StD)	t_{OSD}	-	4.0	-	μ s	$R_L \approx 10$ k Ω , $C_L = 50$ pF
Propagation delay (OE to Q), enable	t_{OTE}	-	50	60	ns	
Propagation delay (OE to Q), disable	t_{OTD}	-	300	-	ns	
Crystal clock frequency	f_{CLK}	3.5759	3.5795	3.5831	MHz	-
Clock output (OSC2), capacitive load	C_{LD}	-	-	30	pF	-

-All voltages referenced to Vss unless otherwise noted. For typical values VDD=5.0 V, Vss=0 V, TA = 25°C, f CLK=3.579545 MHz.
*Typical figures are at 25°C and are for design aid only; not guaranteed and not subject to production testing.

Notes:

1. dBm = decibels above or below a reference power of 1 mW into a 600load.
2. Digit sequence consists of all 16 DTMF tones.
3. Tone duration = 40 ms. Tone pause = 40 ms.
4. Nominal DTMF frequencies are used, measured at GS.
5. Both tones in the composite signal have an equal amplitude.
6. Bandwidth limited (0 to 3 kHz) Gaussian noise.
7. The precise dial tone frequencies are (350 and 440 Hz) $\pm 2\%$.
8. For an error rate of better than 1 in 10,000.
9. Referenced to lowest level frequency component in DTMF signal.
10. Minimum signal acceptance level is measured with specified maximum frequency deviation.
11. Input pins defined as IN+, IN-, and OE.
12. External voltage source used to bias VREF.
13. This parameter also applies to a third tone injected onto the power supply.
14. Referenced to Single - Ended Input Configuration on page 3. Input DTMF tone level at -28 dBm.

Timing Diagram



Explanation of Events

- (A) Tone bursts detected, tone duration invalid, outputs not updated.
- (B) Tone # n detected, tone duration valid, tone decoded and latched in outputs.
- (C) End of tone # n detected, tone absent duration valid, outputs remain latched until next valid tone.
- (D) Outputs switched to high impedance state.
- (D) Tone # $n + 1$ detected, tone duration valid, tone decoded and latched in outputs (currently high impedance).
- (F) Acceptable dropout of tone # $n + 1$, tone absent duration invalid, outputs remain latched.
- (G) End of tone # $n + 1$ detected, tone absent duration valid, outputs remain latched until next valid tone.

Explanation of Symbols

VIN	DTMF composite input signal.
Est	Early steering output. Indicates detection of valid tone frequencies.
St/GT	Steering input/guard time output. Drives external RC timing circuit.
Q1-Q4	4-bit decoded tone output.
StD	Delayed steering output. Indicates that valid frequencies have been present/absent for the required guardtime, thus constituting a valid signal.
OE	Output enable (input). A low level shifts Q1 - Q4 to its high impedance state.
t_{REC}^-	Maximum DTMF signal duration not detected as valid.
t_{REC}	Minimum DTMF signal duration required for valid recognition.
t_{ID}	Minimum time between valid DTMF signals.
t_{DO}	Maximum allowable dropout during valid DTMF signal.
t_{DP}	Time to detect the presence of valid DTMF signals.
t_{DA}	Time to detect the absence of valid DTMF signals.
T _{GTP}	Guard time, tone present.
T _{GTA}	Guard time, tone absent.

ANEXO B

CONSTATANDO SI FUNCIONAN LOS CIRCUITOS IMPLEMENTADOS

Debido a que los circuitos de la Fuente de Alimentación, de Telemando y de Control de Nivel; diseñados e implementados, en el desarrollo del proyecto de tesis propuesto; solo pueden ser probados en su funcionamiento una vez instalados, allá donde se encuentra el sistema de bombeo de agua (Unidad de Producción Julcani, ver foto B.1); y siendo necesario constatar su correcto funcionamiento de forma práctica y consistente, se tuvo que montar un módulo para este propósito. El módulo esta formado por: (1) Un cuerpo metálico (ver foto B.4) de 2 m. de alto por 75 cm. de ancho y una pequeña plataforma de 40 x 25 cm. (2) Dos tubos de PVC de 4 pulgadas de diámetro y 1.40 m. de altura, que tienen a 20 cm. de su base una salida de 0,5 pulgadas para desagüe. (3) Dos electroválvulas que se conectan a través de mangueras a la salida del desagüe y (4) Los circuitos implementados en el desarrollo del proyecto, que se instalan en la pequeña plataforma. En la foto B.7 se observan: Las dos electroválvulas, el circuito de protección contra descargas atmosféricas (chasis de metal), el transductor de presión con su respectiva manguera, la tarjeta de la Fuente de Alimentación, los circuitos de Telemando y Control de Nivel montados en protoboard, y cables; que son utilizados en el montaje del módulo.

El montaje del módulo (ver fotos B.2 y B.3), se realizó en el ambiente de la sala N° 8 llamado Proyectos de Investigación, pabellón A de la FIEE, en la UNI. En el módulo completamente instalado (ver foto B.4), el tubo de la parte inferior hace de

reservorio N° 2 y el tubo de la parte superior hace de reservorio N° 1; y sus respectivas electroválvulas conectadas hacen de electrobomba 2 y electrobomba 1.

El autor opera el módulo (ver foto B.5) en operación manual, realizando la configuración del circuito de Control de Nivel, para luego el sistema funcionar; tal como se describió en el Capítulo 7. Posteriormente se explica el funcionamiento del módulo al especialista de tesis, ingeniero Raúl Benites (foto B.6); respondiendo satisfactoriamente, todas las preguntas planteadas.

Finalmente puedo decir, que se constató el funcionamiento correcto de los circuitos diseñados e implementados, en el desarrollo del proyecto de tesis propuesto.

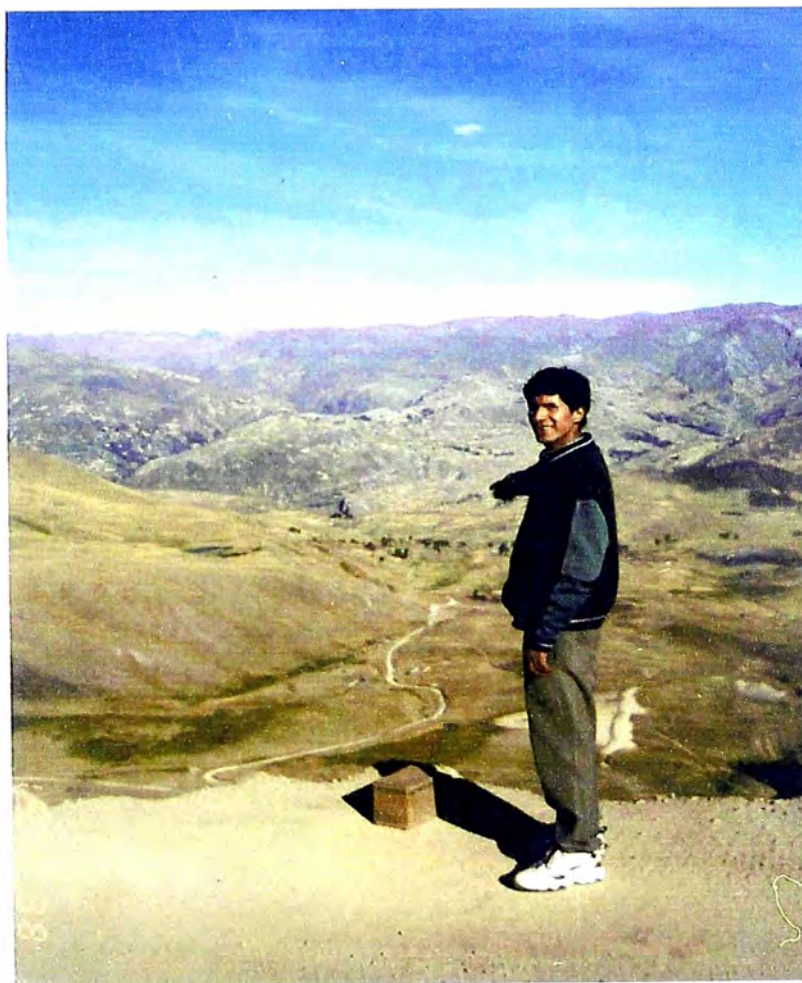


Foto B.1 *Se observa que el autor señala la trayectoria que sigue la instalación del sistema de bombeo de agua, en lo accidentado de la geografía.*

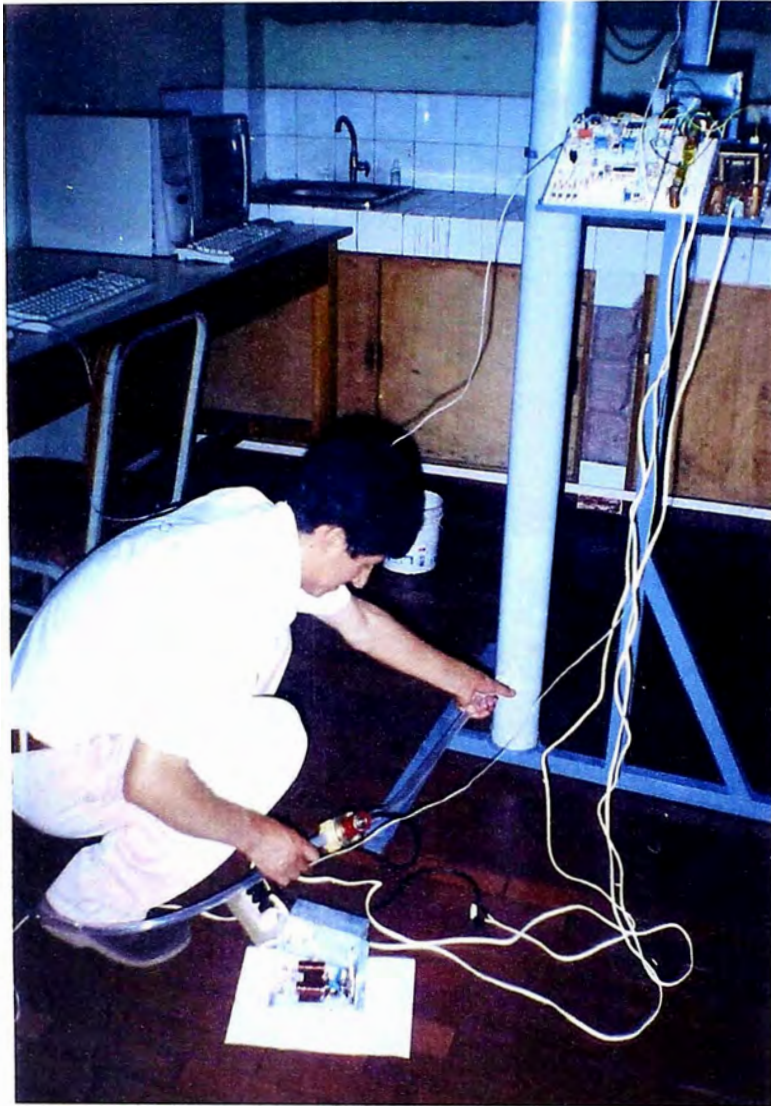


Foto B.2 *Se encuentra instalado los circuitos del sistema. El autor realiza el montaje de la electroválvula.*

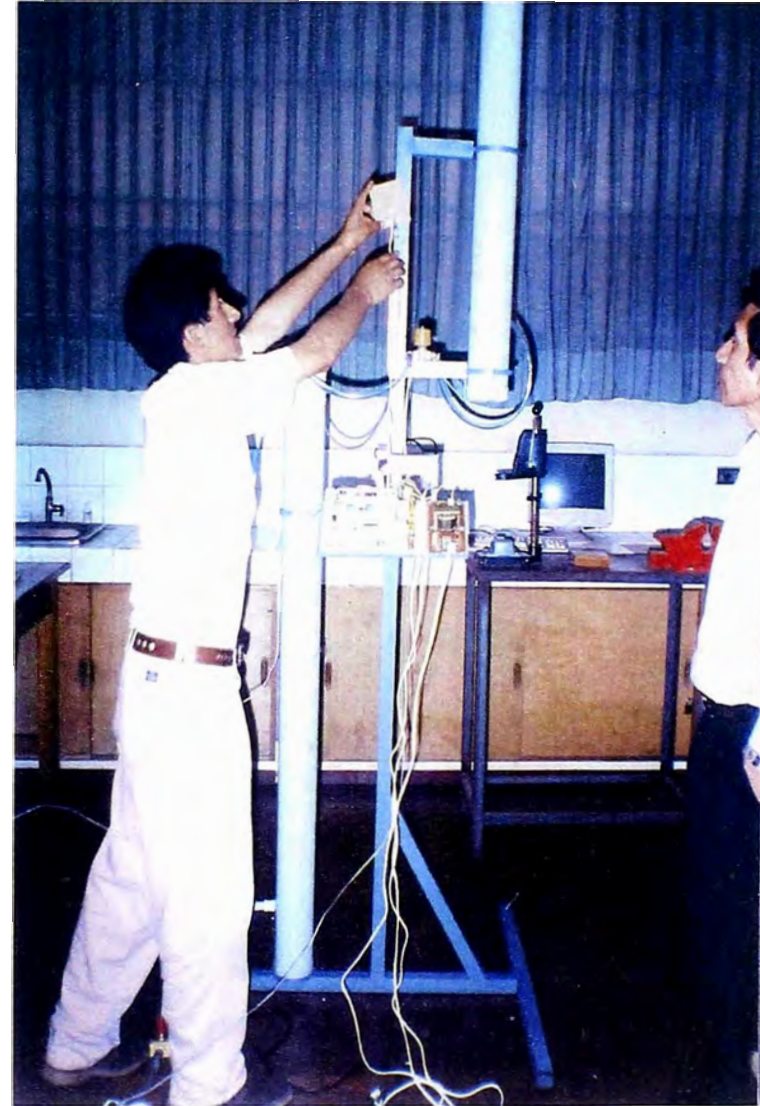


Foto B.3 *Se finaliza el montaje del módulo instalando el sensor de presión en la parte superior.*

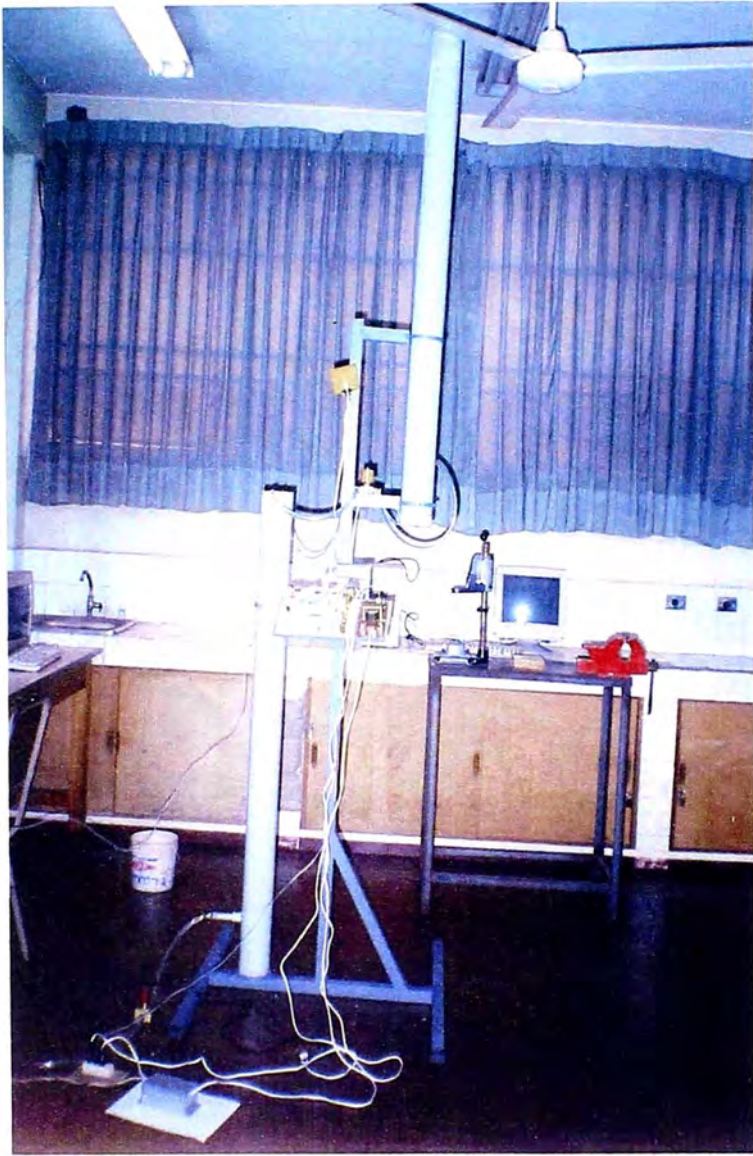


Foto B.4 *Se observa el módulo completamente montado.*



Foto B.5 *Se observa al autor operando el módulo.*



Foto B.6 *El autor explica el funcionamiento del módulo al especialista de tesis.*



Foto B.7 *Se observa diferentes elementos utilizados en el montaje del módulo.*

BIBLIOGRAFÍA

1. **INSTALACIONES DE PUESTA A TIERRA**, Jaime Riba Sardá. Marcombo-Boixareu Editores, España 1989.
2. **AMPLIFICADORES OPERACIONALES Y CIRCUITOS INTEGRADOS LINEALES**, Robert F. Coughlin–Frederick F. Driscoll. Prentice-Hall Hispanoamericana S. A. Cuarta Edición, México 1993.
3. **Revista DELEK ELEKTOR ELECTRONICS ESPAÑA** números 47/48, Editorial A.G.B.S.L., España 1994.
4. **TELEMANDOS. Técnica y Realización**, M. José Gómez Caño. Editorial Paraninfo S. A. Segunda Edición, España 1995.
5. **SISTEMAS DE CONTROL AUTOMÁTICO**, Benjamín C. Kuo. Prentice-Hall Hispanoamericana S. A. Séptima Edición, México 1996.
6. **ELECTRÓNICA INDUSTRIAL MODERNA**, Timothy J. Maloney. Prentice-Hall Hispanoamericana S. A. Tercera Edición, México 1997.
7. **AUTÓMATAS PROGRAMABLES**, Josep Balcells–José Luis Romeral. (Serie Mundo Electrónico). Alfaomega Grupo Editor S. A. de C.V., México 1998.
8. **MICROCONTROLADORES “PIC”. Diseño Práctico de Aplicaciones**, José M. Angulo Usategui–Ignacio Angulo Martínez. McGraw-Hill/Interamericana de España S. A. U. Segunda Edición, España 1999.
9. **SISTEMAS DE REGULACIÓN Y CONTROL AUTOMÁTICOS. Comunicaciones Industriales**, Pedro Morcillo Ruiz–Julián Cócera Rueda. Paraninfo Thomson Learning, España 2000.
10. **AUTOMATIZACIÓN DE PROCESOS INDUSTRIALES**, Emilio García Moreno. Alfaomega Grupo Editor S. A. de C.V, México 2001.
11. **MICROCHIP TECHNICAL LIBRARY (2 CD-ROM)**. First Edition 2001. Microchip Technology Inc., U.S.A. 2001.
12. **DATASHEET de los dispositivos de ANALOG DEVICES**, MOTOROLA, HITACHI, CLARE, PHILIPS, Y SGS-THOMSON, U.S.A. 2002.