



Universidad del Azuay

Facultad de Ciencia y Tecnología

Escuela de Ingeniería Electrónica

**DISEÑO Y CONSTRUCCIÓN DE UN EQUIPO SUPERVISOR DE
RENDIMIENTO PARA LA MAQUINARIA DE PROCESO
INDIVIDUAL DE CURTIEMBRE RENACIENTE S.A.**

**Trabajo de graduación previo a la obtención del título de
Tecnólogo Electrónico**

Autores: Iván Mauricio Montero Armijos

Fernando Giovanni Gordillo Valladarez

Director: Ing. Oswaldo López

Cuenca, Ecuador

2006

Índice de Contenidos

Índice de Contenidos.....	ii
Índice de Figuras:.....	vii
Resumen.....	x
Abstract.....	xi
Introducción	1
CAPITULO I:.....	3
MICROCONTROLADORES PIC.....	3
1.1 Introducción a los microcontroladores y a la familia PIC	3
1.2 Arquitectura Interna	5
1.2.1 Características generales.....	5
1.2.2 Arquitectura y direccionamiento de la Memoria de Programas	7
1.2.3 Arquitectura y direccionamiento de la Memoria de Datos	10
1.3 Líneas de entrada - salida y definición de los puertos	14
1.3.1 Características generales.....	14
1.3.2 Líneas de propósito específico.....	15
1.3.3 Puerto B.....	16
1.3.4 Puerto C.....	16
1.3.5 Puerto D	17
1.3.6 Puerto E.....	17
1.3.7 Circuito oscilador.....	17
1.3.8 Circuito de reset externo	18
1.4 Conexión y manejo de periféricos más comunes.....	19
1.4.1 Leds.....	19
1.4.2 Display de cristal líquido o LCD	20
1.4.2.1 Introducción	20
1.4.2.2 Identificación de los pines de conexión de un modulo LCD no matricial.....	22
1.4.2.3 Tiempos mínimos requeridos para que una instrucción o un dato puedan ser ejecutados.....	25
1.4.2.4 Bus de Datos de 4 y 8 Bits de Longitud.....	28
1.4.2.5 Inicialización del modulo LCD.....	29
1.4.2.6 Conjunto de Instrucciones básicas de un modulo LCD:	31

1.4.2.7 Conexión de un modulo LCD a un Microcontrolador PIC16F84.....	32
1.4.3 Teclas y teclados (teclado matricial).....	33
1.4.3.1 Teclas “Push Button”	33
1.4.3.2 Teclado Matricial 4x4	34
1.4.4 Cargas optoacopladas.....	36
1.4.4.1 Acoplamiento de cargas en régimen de corriente continua	36
1.4.5 Relés.....	37
1.5 Programación del Microcontrolador PIC	38
1.5.1 Introducción	38
1.5.2 Tipos de instrucciones.....	38
1.5.2.1 Orientadas a Byte	39
1.5.2.2 Orientadas a Bits	40
1.5.2.3 Para operar literales y de control	41
1.5.3 Formato de codificación.....	42
1.5.4 Tiempo de duración de las instrucciones	42
1.6 Interrupciones, timers y contadores	43
1.6.1 Trabajo con interrupciones.....	43
1.6.2 Fuentes de interrupciones.....	44
1.6.2.1 Básicas	44
1.6.2.2 Adicionales.....	44
1.6.3 Secuencia de una interrupción	45
1.6.4 Registros para el manejo de interrupciones	46
1.6.4.1 Registro de Control de interrupciones básicas INTCON.....	46
1.6.4.2 Registros de Control y de Banderas de interrupciones adicionales	47
1.6.5 Módulo TMR0	49
1.6.5.1 Especificaciones técnicas generales.....	49
1.6.5.2 Arquitectura interna	50
1.6.5.3 Registros para el Control del Timer 0.....	50
1.6.5.3.1 Registro OPTION (dir. 81h del banco 1).....	50
1.6.5.3.2 Registro INTCON (dir. 0h, 8Bh, 10Bh, 18Bh).....	51
1.6.5.4 Modos de operación del TMR0	51
1.6.5.4.1 Modo Timer	51
1.6.5.4.2 Modo Contador	52

1.7 Comunicación I2C y Modulo SSP	52
1.7.1 Introducción	52
1.7.2 El Bus I2C (Inter IC).....	53
1.7.2.1 Introducción de las especificaciones I2C.....	53
1.7.2.2 El concepto del Bus I2C.....	54
1.7.2.3 Generalidades.....	55
1.7.2.4 Características Generales	56
1.7.2.5 Transferencia del Bit.....	57
1.7.2.6 Instrucciones en el Bus I2C	57
1.7.2.7 Transfiriendo datos	59
1.7.2.8 Reconocimiento	59
1.7.2.9 Arbitraje y generación de señales de Reloj.....	60
1.7.2.10 Formato	63
1.7.2.11 Direccionamiento	65
1.7.2.12 Explicación del 1er byte de direccionamiento	66
1.7.2.13 Especificaciones eléctricas y de tiempos	68
1.7.3 Módulo de Puerto Serial Sincrónico (SSP).....	69
1.7.3.1 Resumen del módulo SSP	69
1.7.3.2 I2C Resumen.....	70
1.7.3.3 Operación SSP en modo I2C.....	74
1.7.3.4 Modo Esclavo	76
1.7.3.5 Direccionamiento	77
1.7.3.6 Recepción (7-bits).....	78
1.7.3.7 Transmisión (7-bits).....	78
1.7.3.8 Modo maestro (Firmware)	80
1.7.3.9 Modo Multi-master (Firmware).....	80
CAPITULO II	82
CARACTERÍSTICAS DEL EQUIPO.....	82
2.1 Características funcionales.....	82
2.2 Características técnicas	82
2.3 Fuentes de Poder Reguladas (Estabilizadas).....	84
2.3.1 Introducción	84
2.3.2 Componentes Básicos	85

2.3.2.1 Transformador de entrada	85
2.3.2.2 Rectificador	86
2.3.2.3 Filtro	88
2.3.2.4 Regulador	89
2.3.3 Fuente Regulada en Tensión y Corriente	95
2.4 Generador de Tiempo Real DS1307	96
2.4.1 Características Generales	96
2.4.2 Tipos disponibles en el mercado para ordenar	97
2.4.3 Asignación y descripción de pines	97
2.4.4 Funcionamiento	98
2.4.5 Descripción de las señales	99
2.4.6 Precisión del reloj	101
2.4.7 Mapa de direcciones de los registros del reloj (RTC) y de la memoria RAM	101
2.4.8 Reloj y Calendario	102
2.4.9 Registro de Control	103
2.4.10 Bus de datos serial a dos hilos	104
2.4.11 Valores absolutos máximos	108
2.4.12 Valores de DC recomendados para cumplir las condiciones de operación ..	109
2.4.13 Características eléctricas de corriente continua	109
2.4.14 Características eléctricas de corriente alterna	110
CAPITULO III	112
DISEÑO Y CONSTRUCCIÓN	112
3.1 Diagrama de bloques y circuitos:	112
3.1.1 Bloque 1: Fuente de alimentación:	112
3.1.2 Bloque 2: Generador de tiempo real:	113
3.1.3 Bloque 3: Teclado matricial:	115
3.1.4 Bloque 4: Display de cristal líquido LCD:	115
3.1.5 Bloque 5: PIC 16F872 (slave):	116
3.1.6 Bloque 6: PIC 16F877 (master):	117
3.2 Programación y diagramas de flujo	118
3.2.1 Diagrama de flujo principal y subrutina de interrupción	119
3.2.2 Diagrama de flujo del LCD y subrutinas	121
3.2.3 Diagrama de flujo del Reloj DS1307 y subrutinas	123

3.2.3 Diagramas de flujo de la comunicación I2C y el modulo MSSP con el esclavo PIC 16F872	127
3.2.4 Programa en lenguaje ensamblador	129
3.3 Pruebas y chequeos, ajustes y comprobación de operación.....	129
3.4 Ensamblaje del equipo completo	130
CAPITULO IV.....	132
MANUAL DEL USUARIO.....	132
4.1 Instrucciones de conexión, puesta en marcha y operación	132
4.2 Mantenimiento	133
CAPITULO V	135
CONCLUSIONES Y BIBLIOGRAFIA	135
5.1 Conclusiones y Recomendaciones	135
5.2 Bibliografía	137
ANEXOS.....	139

Índice de Figuras

Figura 1	Arquitectura tipo Harvad vs. Arquitectura “Von Neuman”.....	6
Figura 2	Esquema de la Memoria de Programas.....	9
Figura 3	Modos de Direccionado de la MD en los MC PICS.....	12
Figura 4	Mapa de la memoria de datos.....	13
Figura 5	Pin out característico de la serie 877.....	15
Figura 6	Pines con funciones especiales.....	16
Figura 7	Configuración para el oscilador principal en los PICs	17
Figura 8	Circuito de RESET externo.....	18
Figura 9	Activación de dos leds con un mismo pin de puerto.	19
Figura 10	Módulo LCD.....	20
Figura 11	Dimensiones de configuración	21
Figura 12	Matriz utilizada para representar un carácter en un modulo LCD	22
Figura 13	Configuración de pines del módulo LCD.....	23
Figura 14	Diagrama de tiempos para una instrucción.....	26
Figura 15	Diagrama de tiempos para escribir un dato.....	27
Figura 16	Diagrama de tiempo para leer un dato.....	27
Figura 17	Inicializaciones para bus de datos de 4 u 8 bits.....	29
Figura 18	Esquema de los circuitos.....	33
Figura 19	Esquema general de conexión de un pulsante tipo “push button”...	34
Figura 20	Teclado matricial 4x4.....	34
Figura 21	Conexionado interno de un teclado matricial 4x4 hacia el puerto B	35
Figura 22	Conexión aislada de un pin de salida.....	36
Figura 23	Conexión aislada de una señal a un pin de entrada digital.....	36
Figura 24	Esquema de conexión para relé.....	37
Figura 25	Instrucciones orientadas a byte.....	39
Figura 26	Instrucciones orientadas a bits.....	40
Figura 27	Operaciones con literales y control	41
Figura 28	Grupos de bits que codifican cada parte de la instrucción.....	42
Figura 29	Registro INTCON.....	46
Figura 30	Registro de Habilitación de Interrupciones PIE1.....	48
Figura 31	Registro de Banderas de Interrupciones PIR1.....	48

Figura 32	Diagrama de Bloques del Timer 0.....	50
Figura 33	Registro OPTION.....	50
Figura 34	Registro INTCON.....	51
Figura 35	Factores de división del PSCALER para el tmr0 y el Watch Dog...	52
Figura 36	Configuración de comunicación I2C.....	54
Figura 37	Conexión del Bus I2C entre dos dispositivos.....	56
Figura 38	Transferencia del Bit en el Bus I2C.....	57
Figura 39	Condiciones de Inicio y Parada.....	58
Figura 40	Transferencia de datos en el Bus I2C.....	59
Figura 41	Reconocimiento en el Bus I2C.....	59
Figura 42	Sincronización del Reloj.....	61
Figura 43	Procedimiento de arbitraje entre dos Masters.....	62
Figura 44	Transferencia completa de datos.....	63
Figura 45	Secuencia de master como transmisor.....	63
Figura 46	Secuencia de master como receptor.....	64
Figura 47	Esquema de formato combinado.....	64
Figura 48	Formato de direccionamiento a 7 bit.....	66
Figura 49	Formato de direccionamiento a 10 bit.....	66
Figura 50	Primer byte luego del procedimiento de Inicio (START).....	67
Figura 51	Definición de bits en el primer byte.....	67
Figura 52	Características de la líneas SDA y SCL para dispositivos I2C.....	69
Figura 53	Diagrama de Bloques SSP (Modo I2C).....	74
Figura 54	Diagrama de recepción de 7 bits.....	78
Figura 55	Diagrama de transmisión de 7 bits.....	79
Figura 56	Bits y direcciones respectivas asociados con el modo I2C.....	81
Figura 57	Componentes de una fuente de alimentación.....	84
Figura 58	Esquema básico de un transformador.....	86
Figura 59	Transformador de armadura y Toroidal “O”.....	86
Figura 60	Conexión de un rectificador.....	87
Figura 61	Rectificador en puente de Greatz.....	87
Figura 62	Aspecto físico de puentes rectificadores encapsulados.....	87
Figura 63	Filtro en conexión básica.....	88
Figura 64	Condensador.....	89

Figura 65	Fuente de alimentación regulada.....	89
Figura 66	Cuadro de tensiones estándar de salida.....	90
Figura 67	Fuente de alimentación para 5V / 500mA.....	91
Figura 68	Forma de los encapsulados más comunes.....	92
Figura 69	Fuente de alimentación con salida variable.....	92
Figura 70	Sentido de circulación de la corriente por un regulador.....	93
Figura 71	Esquema básico mejorado de una regulación.....	94
Figura 72	Esquema mejorado de una regulación ajustable.....	94
Figura 73	Fuente de poder Regulada en Tensión y Corriente.....	95
Figura 74	Asignación de pines en el DS1307.....	97
Figura 75	Circuito operativo típico de un RTC DS1307.....	98
Figura 76	Diagrama de bloques del RTC DS1307.....	99
Figura 77	Montaje recomendado para el cristal.....	101
Figura 78	Mapa de direcciones del RTC y de la RAM.....	102
Figura 79	Registros que guardan la configuración de tiempos.....	103
Figura 80	Valores de frecuencia ajustable con la combinación RS1 y RS0.....	104
Figura 81	Configuración típica den bus de datos serial a dos hilos.....	104
Figura 82	Transferencia de datos sobre un bus serial a dos hilos (I2C).....	106
Figura 83	Escritura de datos del modo esclavo receptor.....	107
Figura 84	Lectura de datos del modo esclavo transmisor.....	108
Figura 85	Diagrama de tiempos DS1307.....	111
Figura 86	Fuente de Alimentación.....	112
Figura 87	Fotografía de la Fuente.....	113
Figura 88	Generador de Tiempo Real.....	113
Figura 89	Fotografía del Generador de Tiempo Real.....	114
Figura 90	Teclado Matricial, Diagrama y Foto.....	115
Figura 91	Display LCD, diagrama y foto.....	115
Figura 92	Diagrama y foto PIC 16F872 (Slave).....	116
Figura 93	Diagrama y foto PIC 16F877 (Master).....	117
Figura 94	Equipo Ensamblado.....	131
Figura 95	Diagrama de Operación del Teclado.....	132

Resumen

DISEÑO Y CONSTRUCCIÓN DE UN EQUIPO SUPERVISOR DE RENDIMIENTO PARA LA MAQUINARIA DE PROCESO INDIVIDUAL DE CURTIEMBRE RENACIENTE S.A.

Expresa el dato en unidades producidas por tiempo. Además podemos ver: Tiempo total de operación de máquina y número total de unidades producidas.

Usa dos circuitos basados en micro controladores PIC de la Microchip:

1 Un circuito MAESTRO (PIC16F877) para manejo de periféricos:

- Teclado matricial.
- Pantalla LCD.
- Reloj de tiempo real.

2 Un circuito ESCLAVO (PIC 16F872), interfaz que conecta al circuito principal con cada máquina.

Estos datos son transferidos entre los circuitos MAESTRO y ESCLAVO utilizando el protocolo de comunicación serial I2C.

Abstract

**DESIGN AND CONSTRUCTION OF A PERFORMANCE SUPERVISING
EQUIPMENT FOR THE INDIVIDUAL PROCESS MACHINERY IN
CURTIEMBRE RENACIENTE S.A.**

Data is expressed in units produced per time. Also, we can view: Total operation time of machine and total number of produced units.

It uses two circuits based upon the microcontrollers PIC from Microchip:

1. A MASTER circuit (PIC16F877), to manage the peripherals:

- Matricial Keyword.
- LCD screen.
- Real time clock chip.

2. A SLAVE circuit (PIC 16F872), interface that connects to the main circuit of each machine.

All data is transferred between the MASTER and SLAVE circuits using the serial communication protocol I2C.

Montero Armijos, Iván

Gordillo Valladarez, Fernando

Trabajo de Graduación

Director: Ing. Oswaldo López

Mayo de 2006

**Diseño y construcción de un equipo supervisor de rendimiento para la
maquinaria de proceso individual de Curtiembre Renaciente S.A.**

Introducción

Actualmente, en la empresa Curtiembre Renaciente S.A., no se controlan ciertos parámetros que son de vital importancia para mejorar su productividad, estos son:

Tiempo real de operación de cada máquina, que serviría para determinar el momento exacto para las intervenciones de mantenimiento preventivo y que actualmente es asumido como 8 horas de operación por turno lo cual es inexacto, ya que no se descuentan los tiempos de preparación de materias primas (*set up 's*) y tiempos perdidos por innumerables causas, esta falta de medición ocasiona que muchas de las veces las intervenciones de mantenimiento preventivo sean tempraneras con la consiguiente pérdida de vida útil real de los repuestos y lubricantes, a más de la pérdida de producción por tiempo de parada, es decir “perdida de dinero”.

Otro punto importante en la medición del tiempo real de operación, conjuntamente con el conteo de las unidades producidas, es el de poder determinar el rendimiento del turno de operadores (unidades producidas / horas de operación). Actualmente se considera que las máquinas producen un valor “X” de unidades/hora de acuerdo a encuestas realizadas a los mismos operadores o a mediciones y muestreo de tiempos anunciados (“dime como me mides y te diré como me comporto” es un axioma fabril muy utilizado), esto ocasiona baja producción ya que los operadores han dimensionado a su gusto el trabajo.

Una vez determinado el valor del rendimiento, se podrá comparar entre turnos de operación, presionar mayor trabajo de acuerdo a los picos medidos, dimensionar con exactitud y de forma real la capacidad de la planta, para así colocar metas de producción y no fallar en los compromisos de cumplimiento con los clientes, esto nos dará un mayor volumen de producción en menor tiempo con disminución de recursos energéticos (perdidas de máquina en vacío) y humanos (operadores inactivos), es decir aumentará la productividad de la planta con una ínfima inversión que costaría el desarrollar este equipo, frente al gran beneficio productivo y por ende económico que se conseguiría.

En resumen: Mayor vida útil de repuestos, lubricantes, piezas de recambio, y de la maquinaria en conjunto, debido a un mejor y mas planificado mantenimiento con tiempos exactos; mayor volumen de producción por cada máquina y en general; menor tiempo perdido de operadores; menos energía consumida por disminución de tiempos de vacío o tiempos muertos de máquina; y mejora en los cumplimientos a los clientes por dimensionamiento correcto de la producción de planta. Todo esto se traduce directamente en mayor rentabilidad económica para la empresa.

Con estos antecedentes se ha justificado la necesidad de desarrollar el equipo que sera objeto de la siguiente memoria técnico-descriptiva, misma que llevara desarrollada solamente la parte teórica y el resultado practico de nuestra investigación, por lo que anticipamos que lo que se estudie en este documento no sera la totalidad de lo que a microcontroladores y a los distintos periféricos se refiere, sino solamente se estudiaran las distintas partes que hemos utilizado para obtener el resultado final.

CAPITULO I:

MICROCONTROLADORES PIC

1.1 Introducción a los microcontroladores y a la familia PIC

En 1971 aparece en el mercado el primer microprocesador (μP) que supuso un cambio decisivo en las técnicas de instrumentación y control. Un microprocesador es un chip programable, que integra pocos recursos de hardware; básicamente los relacionados con el procesamiento de información (CPU¹) y con el trabajo aritmético (ALU²). Para completar el desempeño de los microprocesadores aparecieron un conjunto de chips periféricos, tales como puertos de entrada salida, memoria, temporizadores; entre otros. Tales periféricos formaron parte de una familia de chips discretos con los que el μP debía comunicarse empleando básicamente tres tipos de buses³: bus de datos, bus de direcciones y bus de control.

En el año 1976, gracias al aumento de la capacidad de integración aparece el primer microcontrolador (μC). La diferencia fundamental de un μC con un μP es que el Microcontrolador integra la mayor cantidad de recursos en un solo chip y se comunica con el exterior solamente a través de líneas de entrada / salida o líneas de puerto. En la actualidad la solución de la mayoría de los proyectos electrónicos es pensada en primera instancia utilizando microcontroladores. Tal es el desarrollo alcanzado por las tecnologías de microcontroladores actuales que el papel de los μP ha sido relegado a la fabricación de PCs o a proyectos de gran escala. Algunas razones que justifican la elección de un microcontrolador son las siguientes:

Bajo costo, puesto que integra muchos de los recursos que en μP aparecen de forma discreta y se miniaturizan los diseños, lo que supone abaratar costos de fabricación.

1 CPU: *Central Processing Unit*.

2 ALU: *Arithmetic Logic Unit*.

3 BUS: Conjunto de líneas que permiten la comunicación en sistemas digitales.

Fiabilidad. Un μC integra la mayor parte de los recursos, por lo que se minimizan las interconexiones en la tarjeta de circuito impreso lográndose así un diseño más fiable.

Ahorro de tiempo en el desarrollo de los diseños.

En el mundo existen más de 50 fabricantes de Microcontroladores. Evidentemente, en estas condiciones es muy difícil elegir “el mejor”. Realizando un análisis justo, “el mejor”, en este tipo de tecnología no existe, porque cada aplicación generará necesidades específicas. Ahora bien *Microchip*, es la empresa que ha logrado reunir en sus MICROS, la mayor cantidad de características ventajosas para satisfacer numerosas necesidades en diferentes tipos de aplicaciones. Algunas de las razones por las que los PICS han sido aceptados ampliamente, son las siguientes:

Poseen una relación costo-desempeño relativamente baja, comparada con la de sus competidores.

Trabajan a velocidades grandes (40 Mhz, la última serie 18XXX.). Hasta la gama media, poseen Arquitectura RISC (*Reduced Instruction Set Computer*), o sea manejan un *Set* Reducido de Instrucciones.

Amplia variedad de CHIPS, Microchip ha optimizado la variedad de periféricos que incluye en sus MICROS, con el objetivo de que Ud. encuentre el óptimo para su aplicación y no carezca o derroche recursos electrónicos en el CHIP. Así existen pequeños μC de 8 pines muy poderosos, existen series muy baratas y con gran desempeño, como la de los PICS 16F870/871 y series ultra poderosas, que pueden inclusive emular el modo de trabajo de MICROPROCESADORES como la serie 18XXX. Otra gran ventaja de MICROCHIP, es la migrabilidad del código, entre diferentes tipos de CHIPS y familias, esto permite que un programa escrito para correr en un PIC12C508 de 8 pines sea fácilmente reelaborado para correr en un PIC16C57 que pertenece a la familia básica.

Tecnología fácil de aprender. Ud. puede aprender Microchip de manera modular. ¿Qué significa esto?: Para que se comprenda, ilustremos con un ejemplo. Un PIC16F84, posee

los módulos básicos: puerto A, puerto B, TMR0 y EEPROM de datos y es capaz de manejar 4 fuentes de interrupciones. Un PIC16F870 posee los mismos módulos y maneja también estas 4 fuentes de interrupción, pero adicionalmente posee dos timers, un módulo CCP, un puerto serie, y un conversor A/D de 10 BITS y maneja 7 fuentes de interrupciones adicionales. Si no se comprende aún la idea, esto quiere decir que si Ud. asimiló el trabajo con los módulos incluidos en un F84, solo deberá estudiar en el 870, los módulos adicionales que aparecen. Dos detalles curiosos: ambos micros manejan el mismo SET de INSTRUCCIONES y un PIC16F870 es más barato que un F84.

Existe gran variedad de Herramientas de Desarrollo, proporcionadas por Microchip o empresas alternativas. Gran cantidad de herramientas para el desarrollo las proporciona Microchip de manera gratuita, además de una gran cantidad de Notas de Aplicación, que merecen una profunda consulta.

1.2 Arquitectura Interna

1.2.1 Características generales

La Arquitectura Interna⁴ de los procesadores tipo RISC, como los PICs se caracteriza (entre otros) por los siguientes aspectos:

- Utilizan Arquitectura tipo *HARVARD*.
- Instrucciones largas que se codifican en una sola palabra.
- Arquitectura “registro-fichero” para la Memoria de Datos.
- Instrucciones simétricas.

Arquitectura HARVARD.

Este tipo de arquitectura supone la separación de los Buses que comunican la *Memoria de Programas* y la *Memoria de Datos*, a diferencia de la tradicional arquitectura tipo *Von Neuman* (figura. 1).

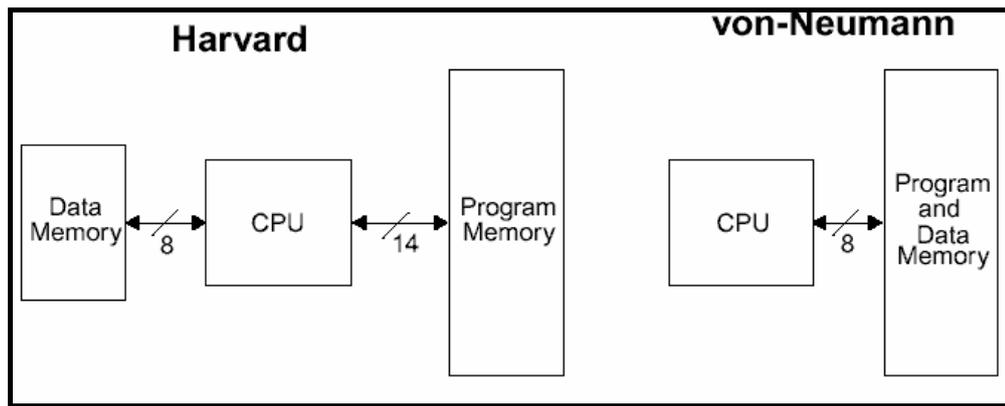


Figura 1. Arquitectura tipo *Harvard* vs. Arquitectura tradicional “*Von Neuman*”.

La separación de buses posee las siguientes ventajas:

Acceso a la Memoria de Datos (MD) y la Memoria de Programas (MP) al mismo tiempo, esto permite realizar multitarea, esto es: a la misma vez que se busca el código de una instrucción es posible ejecutar la que ya se decodificó.

Como los buses son separados, es viable construir un bus para la Memoria de Programas más extenso, esto hace posible que el código de las instrucciones pueda ser grabado en la Memoria de Programas en una sola palabra de longitud mayor a 8 bits⁵ (longitud tradicional). La codificación de las instrucciones en una sola palabra es ventajosa porque de esta manera es posible decodificar toda la instrucción en un solo acceso a la MP, pues el código de la instrucción se lee de una vez.

Simetría (Ortogonalidad de la Instrucciones).

Un *Set* de instrucciones es simétrico, cuando con una misma instrucción es posible direccionar y actuar, ya sea sobre una localización de RAM o un registro de propósito específico. Esta característica es indispensable para lograr un *Set* de Instrucciones tipo RISC. La simetría es posible gracias a que la Memoria de Datos soporta un tipo de arquitectura denominada “registro-fichero”, que permite direccionar de igual manera un Registro de Propósito Específico y uno de Propósito General. Los procesadores PICS, incluyen dentro del mismo espacio de Memoria de Datos a las localizaciones de

⁴ “**Arquitectura Interna**” se refiere a la estructura interna del microcontrolador.

⁵ 8 bits era la longitud tradicional en los procesadores tipo CISC (*Complex Instruction Set Computer*)

propósito general (o del usuario) y a las dedicadas a propósitos específicos (registros especiales). Los Registros de Propósito General (G.P.R) sirven para almacenar datos que garantizan el funcionamiento de determinada aplicación. Pueden ser utilizados “libremente”. Por otra parte los Registros de Propósito Específico (S.F.R) almacenan información relacionada con el funcionamiento del Microcontrolador⁶.

Ciclo de Instrucción en Microcontroladores PICS.

La ejecución de lo que se denomina “Ciclo de Instrucción (CI)” comienza en el estado 1 del Ciclo de Máquina⁷ y es el tiempo que demora la ejecución de una instrucción. Cada familia de Microcontroladores el **CM** queda definido según la cantidad de períodos de reloj que incluya.

Un CI puede requerir de uno o más Ciclos de Máquina (CM) en función de las características de cada instrucción. En los PICS la ejecución de la mayoría de las instrucciones dura un ciclo de máquina, por lo tanto en la mayoría de los casos⁸ el $CM = CI$.

Un CM en PICS es igual a 4 veces el período del oscilador externo. Por ejemplo: si la frecuencia del oscilador externo es igual a 16 Mhz, se tiene un $CM = 0,25 \mu S$.

1.2.2 Arquitectura y direccionamiento de la Memoria de Programas

Arquitectura de la Memoria de Programas en los PICS.

Antes de tocar este tema, cabe recalcar que algunos de los aspectos tratados aquí sólo están limitados a los PICS de la familia media a la que pertenece nuestro PIC, por lo que pudieran sufrir ligeras variaciones al migrar a otras familias.

⁶ Puede referirse al epígrafe Memoria de Datos para ampliar información.

⁷ **Ciclo de Máquina (CM)**. Secuencia de períodos de la señal de reloj con que esta trabajando el Microcontrolador. Para cada familia de Microcontroladores el **CM** queda definido según la cantidad de períodos de reloj que incluya.

⁸ Refiérase al epígrafe “Set de Instrucciones” para aclaración de los casos particulares.

En la Memoria de Programas (MP) quedan almacenados los códigos de las instrucciones de determinada aplicación (programa). Es una memoria no volátil, por lo que la información grabada en ella permanece invariante al suprimir la alimentación al chip. En los MC de la Familia Media de PICs la MP es una memoria tipo “Flash” o “EEPROM” por lo que puede borrarse y escribirse de manera rápida utilizando electricidad. La MP en los PICs de la familia media está organizada en páginas de “2K palabras cada una”. Decimos “2K palabras cada una”, porque los códigos de las localizaciones de la MP en esta familia tienen una longitud de 14 bits. Anteriormente se hacían notar las ventajas asociadas con esta característica. Los PICs de esta familia poseen como máximo 4 páginas de “2K palabras”, lo que hace un total de 8192 posibles instrucciones para cualquier programa.

Para direccionar esta cantidad de memoria se necesitan 13 bits ($2^{13} = 8192$). Para direccionar la MP existe un contador – puntero denominado Contador de Programas (PC). El PC almacena en cada momento la dirección de la instrucción que se está ejecutando. Debe aclararse que el PC sólo almacena la dirección de la instrucción, no la instrucción. Como la cantidad máxima posibles de instrucciones de un programa es 8192, el PC es un contador de 13 bits de longitud.

En la figura 2 se presenta un esquema general de la MP. Debe considerarse que no todos los procesadores de la familia poseen las 8K palabras de memoria. Por ejemplo el PIC16F871 posee sólo 2K palabras y el PIC16F877 posee 8K palabras. El salto entre páginas de la MP se logra escribiendo los dos bits más significativos (bits 11 y 12) de PC. El PC no puede escribirse de una vez porque es de 13 bits, por lo que el fabricante implementó dos registros en la Memoria RAM: el PCLATH y el PCL. El PCLATH es un registro de 5 bits que almacena la parte alta del PC y el PCL es un registro que almacena los 8 bits menos significativos.

Vector de *Reset* y Vector de Interrupción.

El Vector de *Reset* es la primera instrucción que deberá ejecutar el PIC después de un RESET. Se graba en la localización 0000h de la MP. El Vector de Interrupción es siempre una instrucción de salto incondicional (goto XXXX) a la dirección (XXXX) de

la subrutina⁹ donde se atiende(n) la(s) interrupción(es). El Vector de Interrupción se coloca siempre en la localización cuya dirección es 0004h. El Vector de Interrupción estará presente si se trabaja con interrupciones en la aplicación que se está desarrollando.

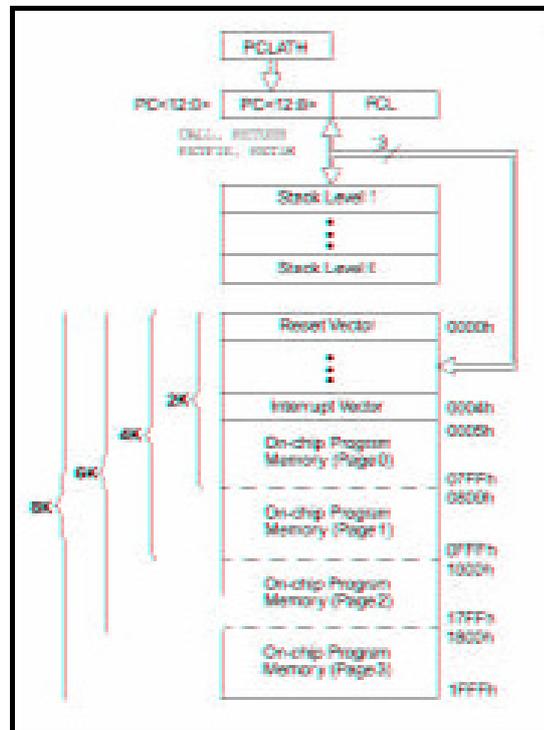


Figura 2. Esquema de la Memoria de Programas.

Direccionado de la Memoria de Programas.

Direccionar la MP significa apuntar (en el sentido directo “señalar”) a la instrucción que en cada momento debe ejecutarse. Durante la ejecución de un programa el direccionado puede ocurrir secuencialmente de manera automática a medida que el PC se va incrementando o se pueden producir “saltos”. Un salto o ruptura del proceso secuencial de la ejecución de un programa se produce cuando se ejecutan instrucciones de tipo *call*, *goto*, *return*, *retfie*, *retlw* ó en determinados casos en instrucciones del tipo *btfs f,b* y *btsc f,b*.

⁹ Una subrutina es un pequeño programa que se ejecuta cada vez que es “llamado” utilizando una instrucción de tipo CALL. Para ampliar información refiérase al tema “Set de Instrucciones”.

La instrucción “*call*” permite realizar “llamadas” a subrutinas. Una subrutina es un programa que se ejecuta cada vez que sea llamado. Cualquier subrutina que sea llamada con un *call* debe terminar en una instrucción *return*. La instrucción *call* modifica el contenido del PC y por lo tanto provoca un salto dentro de la MP.

La instrucción “*goto*” se utiliza para “desviar” la ejecución de un programa hacia otro conjunto de instrucciones. Se diferencia del *call* en que no retorna a la instrucción que le sucede a menos que se ejecute un nuevo *goto* hacia la instrucción deseada al final del proceso que se invocó. La instrucción *goto* modifica el contenido del PC y por lo tanto provoca un salto dentro de la MP. Un *goto* no permite recuperar el valor del PC después de ejecutado un proceso.

Por último las instrucciones *retfie* y *retlw* son casos de “retornos especiales”. El primero permite retornar al programa principal desde una subrutina de atención a interrupción y el segundo retorna de una subrutina con un número grabado en el registro W (“*working register*”). Los dos casos se tratan de manera especial en los temas de “atención a interrupciones” e “implementación y manejo de tablas en la Memoria de Programas”.

1.2.3 Arquitectura y direccionamiento de la Memoria de Datos

Arquitectura de la Memoria de Datos en PICs.

La Memoria de Datos (MD) es una memoria de tipo volátil¹⁰ formada por registros de 8 bits, que comparte espacios para el almacenamiento de los datos necesarios para el funcionamiento de cualquier aplicación (Registros de Propósito General) y para el funcionamiento del Microcontrolador (Registros de Propósito Específico). La MD es una memoria también paginada. Cada página en este caso se refiere como banco. Los procesadores de la serie 877, poseen hasta 4 bancos de 128 *bytes* cada uno; lo hace un total de 512 localizaciones. A diferencia de la Memoria de Programas la MD es volátil de tipo RAM (*Random Access Memory*) y sus localizaciones son de 8 bits. Como ya se había señalado la MD soporta Arquitectura Registro – Fichero, lo que significa que los

¹⁰ Su información se pierde al suprimir la alimentación al chip.

SFRs y GPRs comparten el mismo espacio (en este caso banco). En cada banco los SFRs ocupan siempre las primeras localizaciones y los GPRs las localizaciones superiores. Algunos SFRs que se utilizan con mucha frecuencia aparecen replicados en varios o todos los bancos.

Direccionado de la Memoria de Datos.

Para direccionar la MD es necesario establecer la dirección dentro del banco (cada banco posee 128 localizaciones) y en cuál de los 4 bancos posibles se está trabajando. Para seleccionar el banco a direccionar se necesitan dos.

Modos de direccionado de la Memoria de Datos.

Las direcciones en la MD podrán tener como máximo 9 bits pues existe un total de 512 localizaciones ($2^9 = 512$). Basado en esto, existen dos modos para formar la dirección completa de una localización de la MD: directo e indirecto.

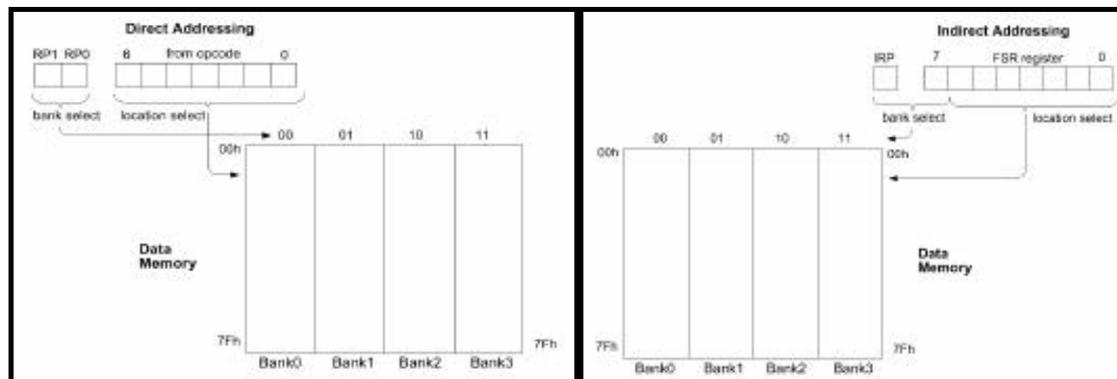


Figura 3. Modos de Direccionado de la MD en los MC PICS. Indirecto (izquierda). Directo (derecha).

Modo directo.

En el Modo Directo (figura 3 izquierda) 7 bits de la dirección vienen en el código de la instrucción. Estos 7 bits son la dirección de un registro (SFR ó GPR) dentro de uno de los 4 bancos posibles (cada banco posee como máximo 128 (7F) bytes).

Para formar la dirección completa faltarían dos *bits*; estos dos bits son los bits 6 y 5 del registro *STATUS* (denominados RP1 y RP0 respectivamente). El registro *STATUS* se encuentra en la dirección 03h de los 4 bancos, o sea es la cuarta localización de c/u de los bancos.

Modo indirecto.

En el Modo Indirecto (figura 3 derecha), la localización de memoria se direcciona mediante un puntero: el registro FSR y se modifica mediante un registro auxiliar, el INDF. El proceso de direccionado ocurre de la manera siguiente: En el registro FSR (puntero de la MD) se cargan los 8 BITS de la dirección base. Como la dirección cargada es de 8 BITS, esta puede referirse a una localización que se encuentre ubicada entre las direcciones 00h y la FFh ó entre la dirección 100 y 1FF, lo que significa que es necesario también definir a que bloque nos referiremos, si al banco 0 y 1 ó al banco 2 y 3. Para formar entonces la dirección completa, es necesario entonces definir el estado del bit “IRP”, que es el BIT # 7 del registro *STATUS*. Una vez que la dirección ha sido formada es necesario “actuar” sobre el registro de la RAM que está siendo apuntado. Para operar sobre el registro se utiliza un registro auxiliar denominado INDF. El INDF ocupa la dirección 00h de todos los bancos.

1.3 Líneas de entrada - salida y definición de los puertos

1.3.1 Características generales

Como ya se había señalado, un microcontrolador se comunica con el entorno solamente a través de líneas de entrada / salida o puertos que vienen integrados al chip. En este sentido, para realizar la conexión de cualquier dispositivo periférico es imprescindible conocer las características relevantes de los puertos de la tecnología que se trabaje. Las características ventajosas de las líneas de puerto en los PICS, también han determinado en gran medida la enorme tasa de acogida entre los diseñadores electrónicos. Dentro de las características generales más sobresalientes pudieran citarse las siguientes:

Pueden manejar hasta 25 mA de corriente, tanto como fuente o sumidero, esto hace que sean capaces de manejar *leds* sin necesidad de *Buffers*.

Son configurables individualmente como salidas ó entradas, mediante registros denominados “TRIS”. Existe un registro TRIS para cada puerto. Cada *bit* del cada registro está asociado al pin físico del puerto en cuestión. Al escribir un UNO en un *bit* de un TRIS queda programado el pin correspondiente como una entrada y al escribir un CERO queda programado como salida. Los pines vienen programados por defecto como entradas.

Muchas de ellas están multiplexadas para realizar una de varias funciones: por ejemplo: una misma línea pudiera ser configurada como entrada ó salida digital ó como una entrada analógica. Al igual que otros periféricos, los puertos en los PICS tratan de mantener las mismas características constructivas al migrar de un dispositivo a otro, esto facilita el aprendizaje de la tecnología.

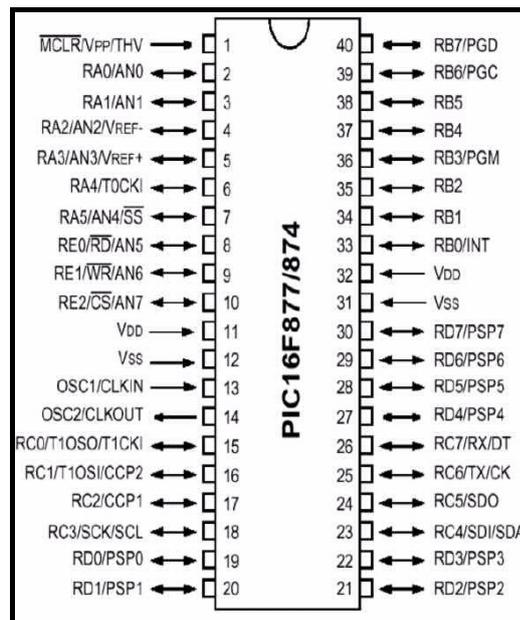
Todas las líneas tienen protección contra ESD (*Electrostatic Discharge*) y en el caso particular de las líneas de puerto tipo “B”, existen resistencias de “*PULL UP*”¹¹ internas, habilitables o no por SW (Software), limpiando el bit 7 de un registro

¹¹ Las resistencias de “*pull up*” son resistores conectados a la fuente. Su función es mantener un nivel estable de UNO cuando la línea se programa como entrada y queda desconectada

(registro “*OPTION*”) que se trata en la sección: “*Timers* integrados al los PICs de la Familia Media”.

Descripción de los Puertos integrados a la serie 87X.

En la figura 5 se puede apreciar un “*pin out*” del microcontrolador que servirá para el desarrollo de este proyecto.



(Disponible en Web: <http://micropic.free.fr/16F877.jpg>)

Figura 5. Pin out característico de la serie 877.

1.3.2 Líneas de propósito específico

Todas las líneas del chip son líneas de puerto de propósito general excepto las que se resumen en la figura 6:

Pines con funciones especiales.

Nombre del pin	Número del pin	Función
(*)MCLR	1	Pin para generar un RESET externo .
OSC1	13	Pin para conexión del oscilador externo
OSC2	14	Pin para conexión del oscilador externo
VSS		Pin de referencia. Generalmente tierra.
VDD		Pin de Alimentación. Generalmente + 5V

Figura 6. Pines con funciones especiales.

1.3.3 Puerto B

Es un puerto de entrada / salida digital de 8 bits, con *pull up*. El *pull up* se puede activar limpiando el bit 7 del registro OPTION_REG. El registro OPTION_REG es la localización 0x81 del banco1 de la Memoria RAM.

1.3.4 Puerto C

Es un puerto de entrada / salida digital de 8 bits que multiplexa algunas funciones para sus líneas. Las líneas multiplexadas más aplicadas son:

Rc2: puede ser un pin de entrada / salida digital o la salida de una onda de PWM generada a partir de un recurso de HW denominado módulo CCP.

Rc6 y Rc7: pueden ser pines de entrada / salida digitales o los pines de comunicación para el USART Tx y Rx respectivamente. Como los niveles de salida en estos pines son CMOS (0 => 5V), es necesario conectarlos a un chip que convierta niveles TTL/CMOS a RS232.

Rc3 y Rc4: son pines q se pueden configurar como ingresos y salidas para la comunicación I2C, para la interconexión de circuitos integrados.

1.3.5 Puerto D

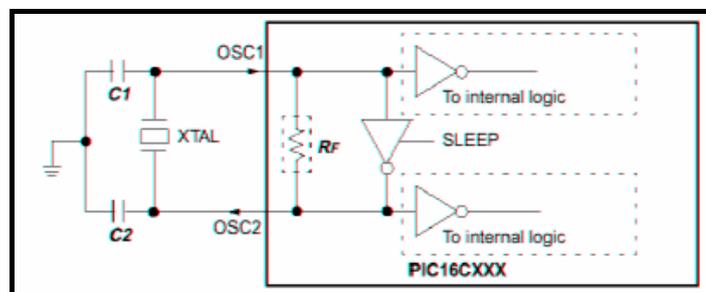
Es un puerto de entrada / salida digital de 8 bits, que multiplexa funciones con el periférico denominado Puerto Paralelo Esclavo (PSP).

1.3.6 Puerto E

Es un puerto de entrada / salida de 3 pines (re0, re1 y re2) cuyas líneas pueden ser configuradas como entrada(s) / salida(s) digital(es) ó entrada(s) analógica(s).

1.3.7 Circuito oscilador

Debe conectarse entre los pines OSC1 y OSC2 un cristal de la manera que se muestra en la figura 7. La conexión externa del cristal con dos condensadores conectados a tierra forma un oscilador con el inversor integrado al pic y conectado entre los pines OSC1 y OSC2¹².



(Disponible en Web: http://www.gte.us.es/usr/jon/sed_iaei/tema8.pdf - pag61)

Figura 7. Configuración típica para el oscilador principal en los PICs de la familia media.

Los osciladores que puede utilizar esta familia se subdividen en las siguientes categorías:

HS. Oscilador a cristal cuya frecuencia es típicamente mayor a 8 Mhz.

¹² CAMPOS CERDA, Manuel. Los microprocesadores Microchip PIC. Escuela Superior de Ingenieros [en línea], [España] 2004-05-16, [citado: 2004-12-20]. Disponible en World Wide Web: http://www.gte.us.es/usr/jon/sed_iaei/tema8.pdf

XT. Oscilador a cristal cuya frecuencia es típicamente menor a 8 Mhz.

LP. Oscilador basado en resonador cerámico o cuarzo de baja potencia, cuya frecuencia está en el orden de los KHz.

RC. Este tipo de oscilador permite conectar, en lugar del tradicional cristal una resistencia y un condensador externo al pin OSC1, para formar junto a la circuitería integrada un oscilador de baja frecuencia muy económico pero también muy impreciso.

1.3.8 Circuito de reset externo

Generar un RESET al PIC significa cargar el Contador de Programas (PC) con el valor 0000h. Esto provoca que se ejecute la primera instrucción de cualquier aplicación, que es denominada “Vector de *RESET*”. Para generar un *RESET* externo es necesario llevar a cero durante un determinado tiempo el pin 1 del chip (etiquetado como (*) MCLR). La figura 8 muestra la circuitería necesaria para generar un pequeño pulso (activo en bajo) en el pin (*) MCLR.

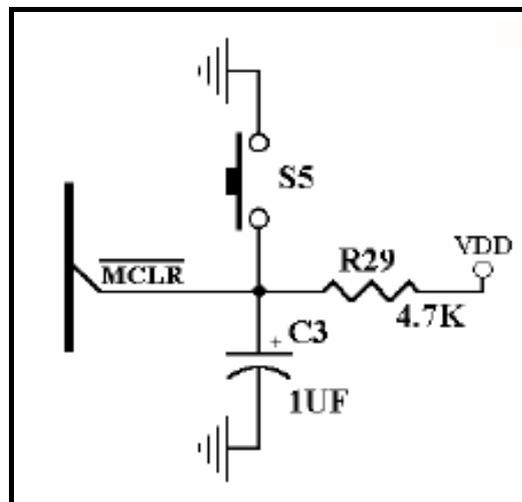


Figura 8. Circuito de RESET externo. El pin MCLR se mantendrá en un nivel bajo durante el tiempo de carga del capacitor C3.

Existen otras fuentes de RESET internas en los PICs de esta familia. Para ampliar información puede consultarse el *datasheet* del PIC16F871/877.

1.4 Conexión y manejo de periféricos más comunes

1.4.1 Leds

Los *leds* pueden conectarse directamente utilizando una resistencia limitadora para la corriente. Para activar un *led* es necesario que el pin al cual se conecta sea programado como salida, escribiendo un cero en el *bit* del registro TRIS correspondiente.

Conexión de dos *leds* a un mismo pin, uno activo con 0 y otro con 1 (figura 9):

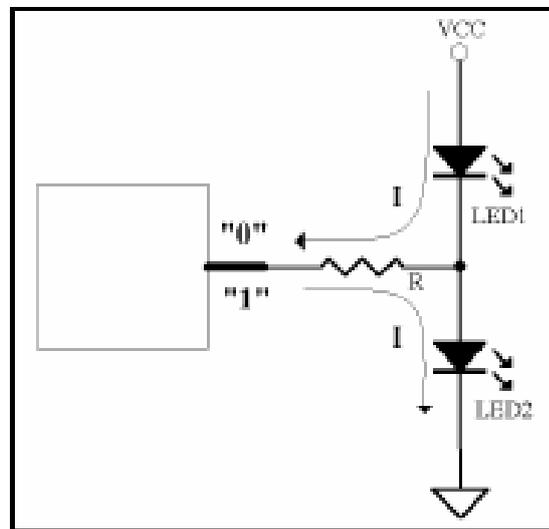


Figura 9. Activación de dos leds con un mismo pin de puerto.

El led1 se activa con “0” y el led2 se activa con “1”.

En todos los casos anteriores la resistencia R puede calcularse suponiendo una corriente de activación para el *led* I de alrededor de 10 mA y un voltaje de activación para el *led* de unos 2V. Bajo estas suposiciones:

$$R = (V_{cc} - 2V) / 10 \text{ mA.}$$

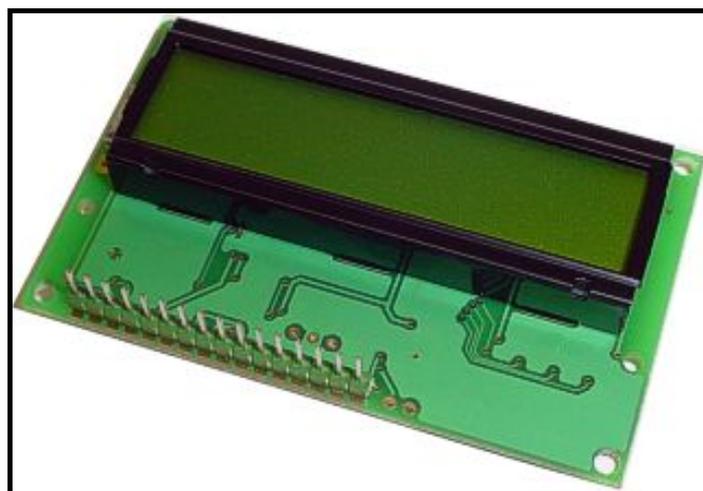
Para $V_{cc} = 5V$ se tiene que:

$$R = 300\Omega.$$

1.4.2 *Display* de cristal líquido o LCD

1.4.2.1 Introducción

Antes de aparecer los módulos LCD, los diseños electrónicos utilizaban los *Displays* de siete segmentos para poder mostrar la información, además de su gran limitación de poder mostrar los caracteres alfa numéricos y símbolos especiales, también consumían demasiada corriente y ocupaban demasiado espacio físico. Posteriormente aparecieron otros tipos de *displays* más complejos que podían mostrar algunos caracteres y símbolos; pero tenían de igual manera mucho consumo de corriente y espacio físico desperdiciado. Finalmente aparecieron los módulos LCD o pantalla de cristal líquido (figura 10), la cual tiene la capacidad de mostrar cualquier caracter alfa numérico. Estos dispositivos ya vienen con su pantalla y toda la lógica de control pre-programada en la fábrica y lo mejor de todo es que el consumo de corriente es mínimo y no se tendrán que organizar tablas especiales como se hacía anteriormente con los *displays* de siete segmentos. Las aplicaciones de los módulos LCD son infinitas ya que podrán ser aplicados en la informática, comunicaciones, telefonía, instrumentación, robótica, automóviles, equipos industriales, etc. Toda la gran cantidad de aplicaciones que tiene un modulo LCD queda a su imaginación. En este proyecto nos servirá como un módulo de salida o visualización de datos.

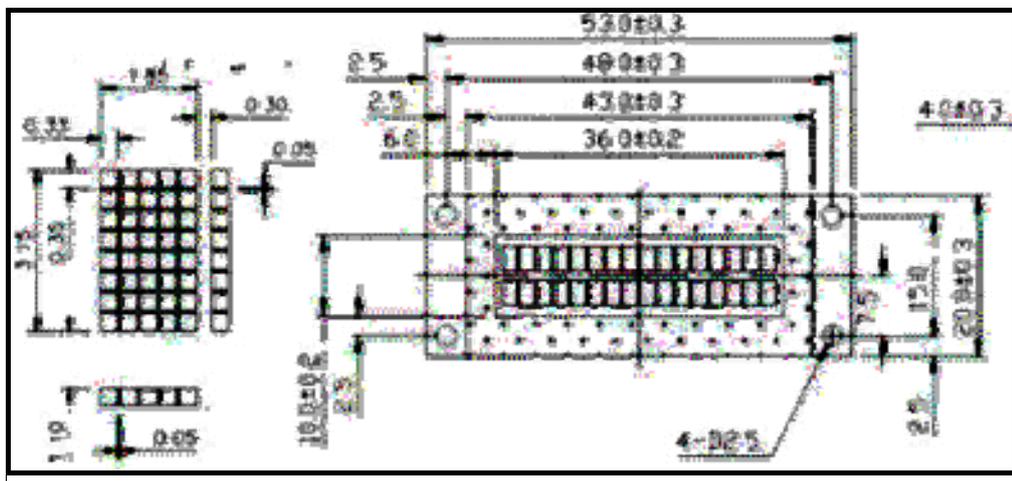


(Disponible en Web: <http://www.ilustrados.com/publicaciones/multimedia/modulo1.jpg>)

Figura 10. Módulo LCD

En la actualidad los módulos LCD tienen una gran variedad de versiones clasificadas en dos grupos. El primer grupo esta referido a los módulos LCD de caracteres (solamente se podrán presentar caracteres y símbolos especiales en las líneas predefinidas en el modulo LCD) y el segundo grupo esta referido a los módulos LCD matriciales (Se podrán presentar caracteres, símbolos especiales y gráficos). Los módulos LCD varían su tamaño físico dependiendo de la marca; por lo tanto en la actualidad no existe un tamaño estándar para los módulos LCD.

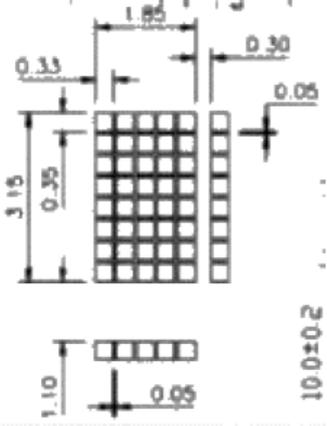
La figura 11 muestra las dimensiones de una configuración típica de un modulo LCD de dos líneas por 16 caracteres por cada línea, incluyendo los detalles de la matriz de como esta conformado un carácter.



(Disponible en Web: <http://www.ilustrados.com/publicaciones/multimedia/modulo2.gif>)

Figura 11. Dimensiones de configuración

Otro patrón importante es el tamaño de los caracteres donde las dimensiones de la matriz que forma los caracteres tienen longitudes diferentes. La siguiente imagen muestra la matriz utilizada para poder representar un símbolo o un carácter alfa numérico en un modulo LCD. Esta matriz define algunos aspectos importantes del carácter o el símbolo que están mostrando. Los aspectos que definen esta matriz son:

Matriz de punto para un solo carácter en un modulo LCD	Aspectos importantes que define la matriz de puntos para un solo carácter en un modulo LCD
	<p>1-. Altura del carácter definida por dos variables: Alto de cada punto que conforma la matriz y longitud de separación entre cada punto que conforma la matriz.</p> <p>2-. Ancho del Carácter definido por dos variables: Ancho de cada punto que conforma la matriz y longitud de separación entre cada punto que conforma la matriz.</p> <p>3-. Calidad gráfica del carácter (A mayor cantidad de puntos dentro de la matriz, mayor será la calidad visual del carácter presentado por el modulo LCD.</p>

(Disponible en Web: <http://www.ilustrados.com/publicaciones/EpyVFEkEEVEBKYSIJ.php>)

Figura 12: Matriz utilizada para poder representar un carácter en un modulo LCD

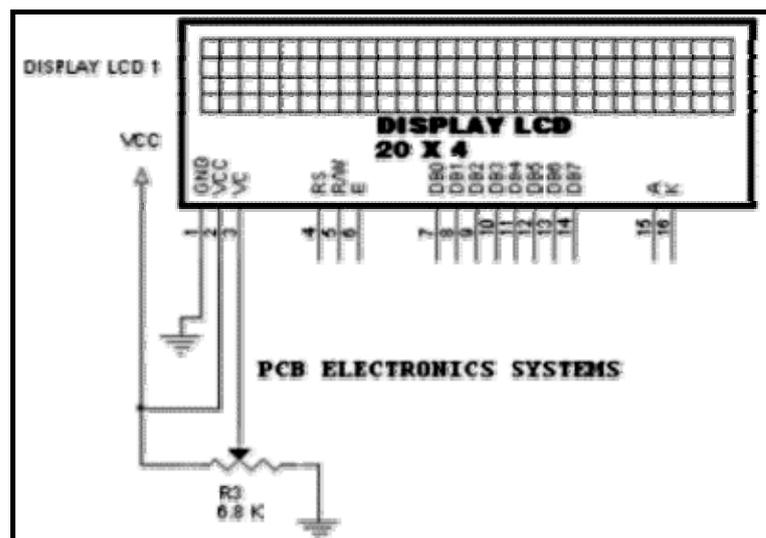
Los primeros módulos LCD tenían los caracteres de color negro y el fondo de la pantalla era de color verdoso claro. Posteriormente se crearon otros colores en donde los caracteres eran de color plata y así sucesivamente fueron variando los colores en el fondo y en los caracteres incluyendo una luz posterior para los módulos LCD denominada *Back Light* diseñada especialmente para mejorar la visualización de la pantalla sobre todo en lugares muy oscuros.

1.4.2.2 Identificación de los pines de conexión de un modulo LCD no matricial

Los pines de conexión de un modulo LCD han sido estandarizados por lo cual en la mayoría de ellos son exactamente iguales siempre y cuando la línea de caracteres no sobrepase los ochenta caracteres por línea. Por otro lado es de suma importancia localizar exactamente cual es el pin Numero 1 ya que en algunos módulos se encuentra hacia la izquierda y en otros módulos se encuentra a la derecha.

PIN No.	SIMBOLOGIA	NIVEL	I/O	FUNCIÓN
1	VSS	-	-	0 Vlts. Tierra (GND).
2	VCC	-	-	+ 5 Vlts. DC.
3	Vee = Vc	-	-	Ajuste del Contraste. 0= Escribir en el modulo LCD.
4	RS	0/1	I	1= Leer del modulo LCD 0= Entrada de una Instrucción.
5	R/W	0/1	I	1= Entrada de un dato.
6	E	1	I	Habilitación del modulo LCD
7	DB0	0/1	I/O	BUS DE DATO LINEA 1 (LSB).
8	DB1	0/1	I/O	BUS DE DATO LINEA 2
9	DB2	0/1	I/O	BUS DE DATO LINEA 3
10	DB3	0/1	I/O	BUS DE DATO LINEA 4
11	DB4	0/1	I/O	BUS DE DATO LINEA 5
12	DB5	0/1	I/O	BUS DE DATO LINEA 6
13	DB6	0/1	I/O	BUS DE DATO LINEA 7
14	DB7	0/1	I/O	BUS DE DATO LINEA 8 (MSB).
15	A	-	-	LED (+) Back Light
16	K	-	-	LED (-) Back Light.

El Pin numero 1 y 2 están destinados para conectarle los 5 Voltios que requiere el modulo para su funcionamiento y el Pin numero 3 es utilizado para ajustar el contraste de la pantalla; es decir colocar los caracteres mas oscuros o mas claros para poderse observar mejor. Observe a continuación la figura 13., en esta se indica cómo deben estar conectados los tres primeros pines. La resistencia representada como R3 es un potenciómetro variable que puede oscilar entre 10 K y 20 K indiferentemente.



(Disponible en Web: <http://www.ilustrados.com/publicaciones/multimedia/modulo4.gif>)

Figura 13. Configuración de pines del módulo LCD

El Pin número 4: denominado "RS" trabaja paralelamente al Bus de datos del modulo LCD (Bus de datos son los Pines del 7 al 14). Este bus es utilizado de dos maneras, ya que usted podrá colocar un dato que representa una instrucción o podrá colocar un dato que tan solo representa un símbolo o un carácter alfa numérico; pero para que el modulo LCD pueda entender la diferencia entre un dato o una instrucción se utiliza el Pin Numero 4 para tal fin. Si el Pin numero 4 = 0 le dirá al modulo LCD que está presente en el bus de datos una instrucción, por el contrario, si el Pin numero 4 = 1 le dirá al modulo LCD que esta presente un símbolo o un carácter alfa numérico.

El Pin número 5: denominado "R/W" trabaja paralelamente al Bus de datos del modulo LCD (Bus de datos son los Pines del 7 al 14). También es utilizado de dos maneras, ya que se le podrá decir al modulo LCD que escriba en pantalla el dato que esta presente en el Bus; por otro lado también podrá leer qué dato esta presente en el Bus. Si el Pin numero 5 = 0 el modulo LCD escribe en pantalla el dato que esta presente el Bus; pero si el Pin numero 5 = 1 significa que usted necesita leer el dato que esta presente el bus del modulo LCD.

El Pin número 6: denominado "E" que significa habilitación del modulo LCD tiene una finalidad básica: conectar y desconectar el modulo. Esta desconexión no estará referida al voltaje que le suministra la corriente al modulo; la desconexión significa tan solo que se hará caso omiso a todo lo que este presente en el bus de datos de dicho modulo LCD. En la mayoría de los circuitos electrónicos modernos que incluyan elementos electrónicos como Microcontroladores, Memorias y Módulos LCD, utilizan el mismo bus de datos. Esto es para no tener un bus de datos independientemente por cada elemento electrónico, esto implicaría que los circuitos electrónicos sean mucho más grandes por la cantidad de conexiones necesarias a cada uno de los elementos. Ahora como los Microcontroladores, memorias y módulos LCD utilizan el mismo bus de datos, deberá existir en cada uno de ellos un Pin de habilitación "E" que permita desconectar y conectar cuando sea necesario. Por ejemplo si se necesita trabajar con la memoria RAM para obtener o escribir cierta información, será necesario que se deshabilite el modulo LCD para que no presente basura en la pantalla, o se ejecuten instrucciones no deseadas.

Los Pines desde el número 7 hasta el número 14 representan 8 líneas que se utilizan para colocar el dato que representa una instrucción para el modulo LCD o un carácter alfa numérico. El Bus de datos es de 8 *Bits* de longitud y el *Bit* menos significativo esta representado en el Pin numero 7, el Pin mas significativo esta representado en el Pin numero 14.

Los Pines 15 y 16: estarán destinados para suministrar la corriente al *Back Light*. Es importante conocer que no todos los módulos LCD disponen del Back Light aunque tenga los pines de conexión en el circuito impreso.¹³

1.4.2.3 Tiempos mínimos requeridos para que una instrucción o un dato puedan ser ejecutados.

Los Pines de control (E, RS y R/W) están estrechamente relacionados ya que por medio de ellos podemos especificar si queremos ejecutar una instrucción o leer / escribir un dato en la pantalla o la memoria RAM; sin embargo existe una condición importante que deberá tomarse en cuenta, referida directamente al tiempo necesario que se necesita para cambiar de un estado a otro en los pines de control. (E, RS y R/W). En el caso de que este tiempo sea mas pequeño que el tiempo mínimo requerido, entonces el modulo LCD no tendrá el tiempo suficiente para responder a las instrucciones solicitadas por el usuario y por consecuencia se perderán los datos o instrucciones según sea el caso.

En otras palabras, las personas suelen cometer un error común cuando se está intentando hacer funcionar un modulo LCD en el cual no consideran la velocidad de proceso del microprocesador o el microcontrolador específicamente en los pines de control (E, RS y R/W), esto quiere decir que si se tuviera conectado un modulo LCD a un microcontrolador que tiene una velocidad de proceso demasiado alta en los pines de control, cuando se ejecuta una solicitud de cualquier tipo (escritura / lectura e Instrucción.), el modulo LCD no tendrá la capacidad de entender la solicitud hecha por el microcontrolador ya que esta se ejecuto demasiado rápida. Para ello los programas o

¹³ OROZCO SALGUERO, Mauricio A. Control de un módulo LCD con Pic16f84a, [en línea], 2003-10-04, [citado: 2004-12-17] Disponible en World Wide Web: <http://www.ilustrados.com/publicaciones/EpyVFEKEEVEBKYSIJ.php>

los circuitos electrónicos que manejan un modulo LCD deberán respetar los siguientes diagramas de tiempo.

Diagrama de tiempo para una Instrucción.

Para enviarle una instrucción al modulo, primero hay que colocar esa instrucción en el bus de datos (Pines del 7 al 14). Una vez que está presente la instrucción en el bus de datos se procede a ejecutar el diagrama de tiempo requerido para una instrucción en los pines de control. Este diagrama de tiempo es muy sencillo de entender, tan solo se deberá colocar el Pin RS = 0, el Pin R/W = 0 y el Pin E = 0; Una vez colocados los pines con las tensiones mencionadas, se procede a cambiar el estado del Pin E = 1. El nuevo estado de este Pin "E" deberá permanecer por lo menos 450 ns antes de volver a cambiar de estado para que la pantalla pueda entender la instrucción.

Este sencillo diagrama lo podemos visualizar en la figura 14 a continuación.

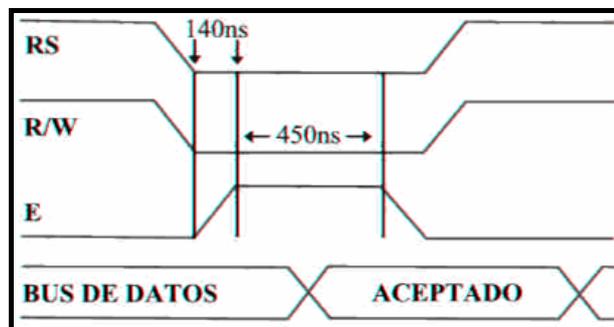
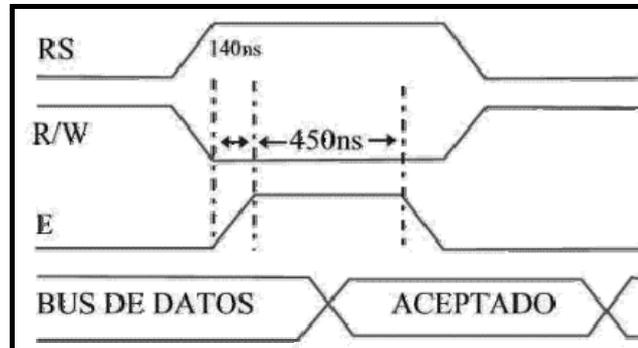


Figura 14. Diagrama de tiempos para una instrucción.

Diagrama de tiempo para escribir un dato.

Para escribir un dato en el modulo LCD, primero hay que colocar el dato en el bus (Pines del 7 al 14). Una vez que está presente el dato en el bus se procede a ejecutar el diagrama de tiempo requerido para escribir un dato en los pines de control. Este diagrama de tiempo es muy sencillo de entender, tan solo se deberá colocar el Pin RS=1, el Pin R/W=0 y el Pin E=0; Una vez colocados los pines con las tensiones mencionadas, se procede a cambiar el estado del Pin E=1. El nuevo estado de este Pin "E" deberá

permanecer por lo menos 450 ns antes de volver a cambiar de estado para que la pantalla pueda entender la instrucción. Gráficamente se lo entiende de mejor manera en la figura 15.

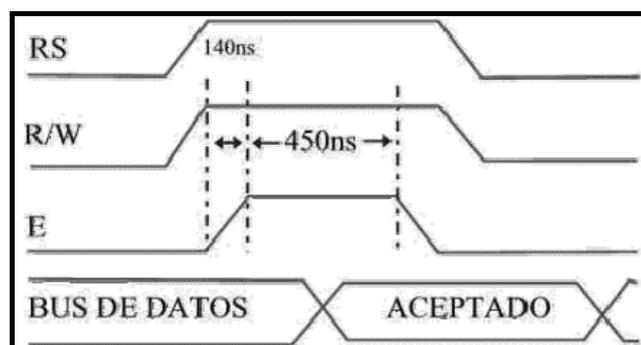


(Disponible en Web: <http://www.ilustrados.com/publicaciones/multimedia/modulo6.gif>)

Figura 15. Diagrama de tiempos para escribir un dato

Diagrama de tiempo para leer un dato

Para leer un dato de la pantalla o la memoria RAM en el modulo LCD, los pines de control deberán estar colocados como sigue: Pin RS = 1, Pin R/W = 1 y el Pin E = 0. Una vez colocados los pines con las tensiones mencionadas, proceda a cambiar el estado del Pin E=1. El nuevo estado de este Pin "E" deberá permanecer por lo menos 450 ns antes de volver a cambiar de estado para que la pantalla pueda entender la instrucción.



(Disponible en Web: <http://www.ilustrados.com/publicaciones/multimedia/modulo6.gif>)

Figura 16. Diagrama de tiempo para leer un dato

Nota importante:

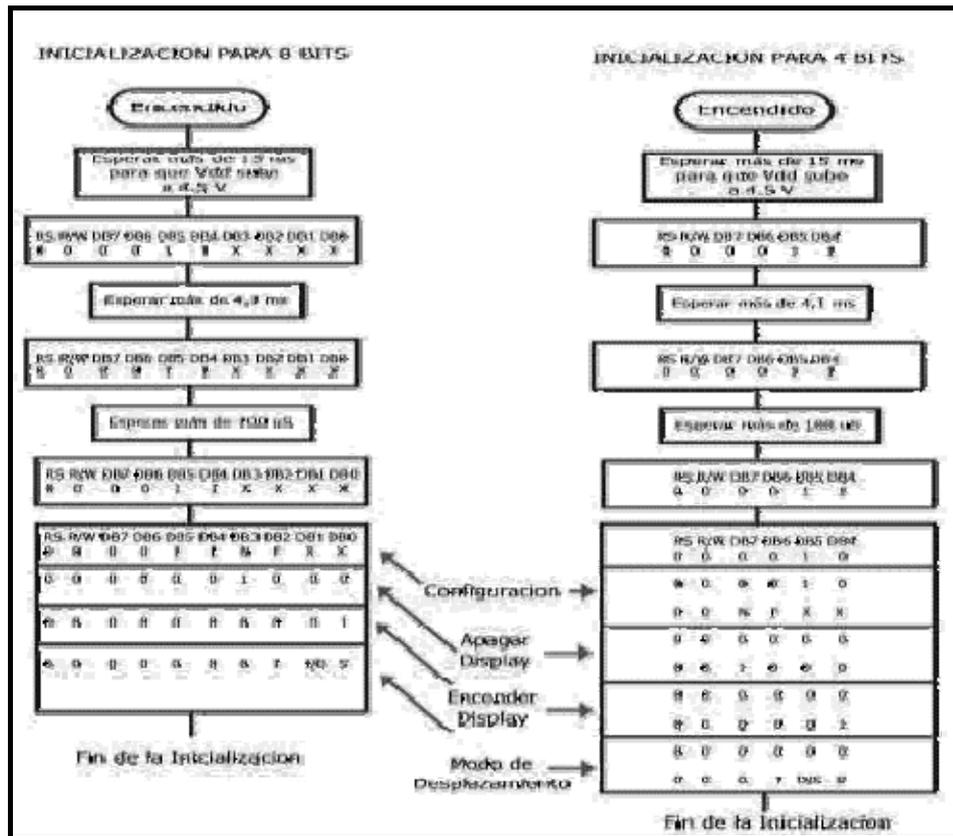
Nótese que en los tres diagramas de tiempos el Pin denominado como "E" tiene que estar previamente en el estado lógico "CERO", posteriormente se cambia el estado lógico a "UNO" la cual permanecerá por lo menos unos 450 nano segundos y finalmente vuelve al estado lógico "CERO", en el preciso momento que el Pin "E" del modulo LCD tiene el flanco de bajada, es cuando se ejecuta la instrucción o el dato presente en el bus de control.

1.4.2.4 Bus de Datos de 4 y 8 Bits de Longitud

El Bus de datos de un modulo LCD puede ser configurado para trabajar con 4 Bits y con 8 Bits. Para los diseños electrónicos que están limitados por la cantidad de líneas utilizadas en el Bus de datos, se podrá utilizar un bus de datos con una longitud de 4 Bits como en este proyecto; sin embargo si este no fuera el caso, se podrá utilizar el bus de datos completo de 8 Bits.

Las señales de control (RS - R/W - E) y los diagramas de tiempo explicados anteriormente, trabajan igual sea para un bus de datos de 4 Bits o de 8 Bits. Sin embargo, si se esta interesado en trabajar el bus de datos con una longitud de 8 Bits, se deberá saber que cuando se enciende el modulo LCD la configuración para 8 Bits entra por defecto; es decir que no necesitara programarse, pero la configuración del bus de datos con una longitud de 4 Bits requiere una secuencia de instrucciones previas inmediatamente después de encender el modulo LCD.

La longitud escogida para trabajar el bus de datos deberá hacerse en el principio de la programación del modulo LCD. En la figura 17, se puede observar la inicialización de un modulo LCD para trabajar con un bus de datos de 8 Bits (izquierda) y de 4 Bits (derecha).



(Disponible en Web: <http://www.ilustrados.com/publicaciones/multimedia/modulo8.gif>)

Figura 17. Inicializaciones para bus de datos de 4 u 8 bits

1.4.2.5 Inicialización del modulo LCD

Todo modulo LCD deberá inicializarse, esta inicialización indicará cómo deberá operar la pantalla. La inicialización representan las instrucciones que deberán ser ejecutadas por el modulo LCD antes de su funcionamiento normal. Las instrucciones que están dentro de la inicialización solamente se ejecuta después que se enciende el modulo LCD y no podrán ser cambiadas posteriormente. Por ejemplo tenemos algunos parámetros que pueden ser ejecutados en la inicialización antes de comenzar a funcionar nuestro modulo LCD:

- Selección de la longitud del bus de datos (4 Bits / 8 Bits).
- Activar el numero de líneas que se visualizaran en el modulo LCD.
- Encender el Modulo LCD.

Las siguientes instrucciones también podrán ser colocadas en la inicialización, con la diferencia que podrán ser cambiadas en cualquier parte del programa:

- Mantener el mensaje fijo y desplazar el cursor.
- Desplazar el mensaje y mantener el cursor fijo.
- Hacer que el carácter señalado parpadee o no.

1.4.2.6 Conjunto de Instrucciones básicas de un modulo LCD:

Instrucción.	CODIGO										Descripción	Tiempo de ejecución
	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0		
Borrar Pantalla	0	0	0	0	0	0	0	0	0	1	Borra la pantalla y retorna el cursor a la dirección 0 (Home)	1.64 mS.
Cursor Home	0	0	0	0	0	0	0	0	1	*	Retorna el cursor al inicio (Dirección 0)	1.64 mS.
Modo de entrada de caracteres	0	0	0	0	0	0	0	1	I/D	S	Donde I/D=0 Decrementa la posición del cursor, I/D=1 incrementa la posición del cursor,. S=0 El texto de la pantalla no se desplaza, S=1 El texto de la pantalla se desplaza en el momento que se escribe un carácter	40 uS.
Apagado y encendido de la pantalla.	0	0	0	0	0	0	1	D	C	B	Donde D=0 Pantalla apagada, D=1 Pantalla encendida, C=0 Cursor apagado, C=1 Cursor encendido, B=0 Intermitencia del cursor apagado, B=1 Intermitencia del cursor encendido.	40 uS..
Cursor and Display Shift	0	0	0	0	0	1	S/C	R/L	*	*		40 uS.
Function Set	0	0	0	0	1	DL	N	F	*	*		40 uS.
Set CG RAM address	0	0	0	1	ACG							40 uS.
Set DD RAM address	0	0	1	ADD								40 uS.
Ready busy flag & address	0	1	BF	AC								1 uS.
Write data to CG or DD RAM	1	0	Escribir el Dato									120 uS.
Read data to CG or DD RAM	1	1	Leer el Dato									40 uS.

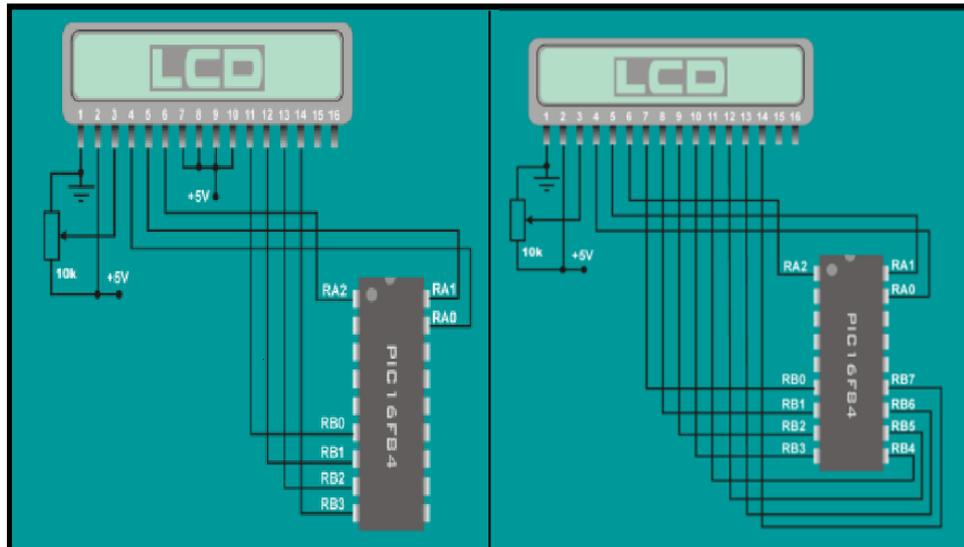
Abreviatura	
DD RAM	Display Data RAM
CG RAM	Generador de Caracteres RAM

Nomenclatura	Variable = 1	Variable = 0
I/D	I/D=1 Incrementa el Cursor en una posición	I/D=0 Decrementa el Cursor en una posición.
D	D=1 Pantalla Encendida	D=0 Pantalla Apagada.
C	C=1 Cursor Encendido.	C=0 Cursor Apagado.
B	B=1 Intermitencia del cursor encendida.	B=0 Intermitencia del cursor apagado
S/C	S/C=1 Mover todo el texto.	S/C=0 Mover el cursor.
R/L	R/L=1 Mover todo el texto a la izquierda.	R/L=1 Mover todo el texto a la derecha.
DL	DL=1 Bus de datos de 8 Bits.	DL=0 Bus de datos de 4 Bits.
S	S=1 Desplazamiento del texto.	S=0 No desplazamiento del texto
BF	BF=1 Operación Interna en progreso.	BF=0 No puede aceptar instrucción
F	F=1 Matriz para el carácter de 5 X 10 dots	F=0 Matriz del carácter de 5 x 7 Dost
N	N=1 Activación de dos líneas.	N=0 Activación de 1 línea

1.4.2.7 Conexión de un modulo LCD a un Microcontrolador PIC16F84

En la actualidad los microcontroladores son los elementos electrónicos de mayor utilidad y ahora por razones didácticas y de aprendizaje básico, describimos en detalle como utilizar un modulo LCD con un microcontrolador Microchip modelo PIC16F84. La conexión entre un módulo LCD y un microcontrolador PIC16F84 debe realizarse como se ilustra en la figura 18, en donde se observan las dos posibilidades, sea para un bus de 4 bits (izquierda) o para uno de 8 bits (derecha), en el caso de nuestro proyecto se utilizará un bus de 4 bits¹⁴:

¹⁴ MARTÍN "IONITRON", Daniel C. Funcionamiento de Teclados Matriciales y *Displays* LCD, [en línea], [Barcelona, España], 2001-06-20, [citado 2005-10-12]. Disponible en World Wide Web: <http://www.x-robotics.com/rutinas.htm>



(Disponible en Web: <http://www.x-robotics.com/images/lcd4bits.gif>, <http://www.x-robotics.com/images/lcd8bits.gif>)

Figura 18. Esquema de los circuitos

1.4.3 Teclas y teclados (teclado matricial)

1.4.3.1 Teclas “*Push Button*”

Las líneas que se utilicen para conectar teclas deben programarse como entradas, puesto que las teclas son periféricos de entrada al MC. Para conectar teclas resulta excelente el puerto B puesto que posee resistencias de pull ups internas como ya se había señalado. En la figura 19 se muestra la conexión directa de pulsantes a través de una resistencia de 220Ω . Las teclas entregan en este caso un nivel bajo cuando son pulsadas y un nivel alto cuando están en estado abierto. La resistencia de 220Ω se coloca para proteger la línea de puerto contra cortocircuitos a tierra ante una situación indeseada como la siguiente:

Suponga que por error se programan los pines del puerto B como salidas (registro $TRISB = 00000000$) y además escribe un “1” en la línea a la cual conectada una tecla. Si la resistencia no estuviera presente, al pulsar la tecla se conectaría a tierra la línea de puerto que en ese momento está a un voltaje alto, por lo que pudiera circular una

corriente por ella lo suficientemente alta como para producir daños. A pesar de la existencia de la resistencia limitadora de 220Ω , el nivel de “0” no se afecta.

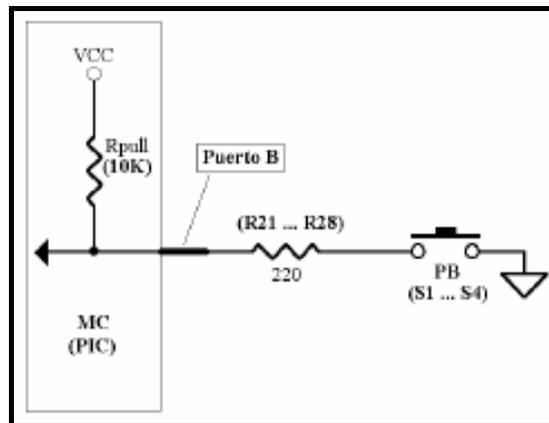
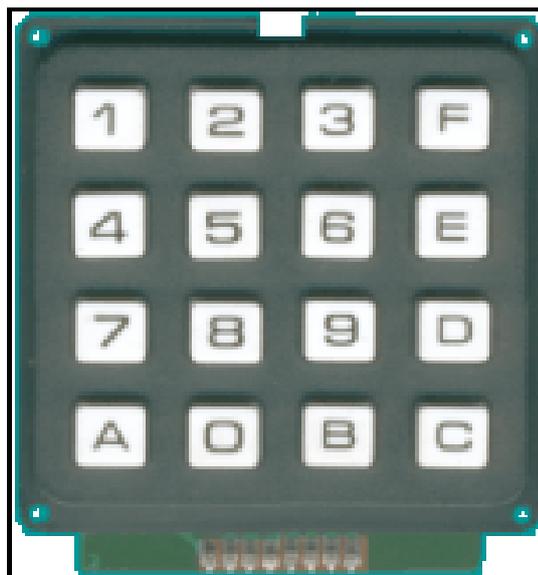


Figura 19. Esquema general de conexión de un pulsante tipo “push button”.

1.4.3.2 Teclado Matricial 4x4

Un teclado matricial 4x4 (figura 20), es un dispositivo de entrada de datos que consta de 16 teclas o pulsadores, dispuestos e interconectados en filas y columnas. Dispone de un conector SIL (*Single In Line*) macho de 8 pines que se corresponden con las 4 filas y las cuatro columnas de las que dispone.

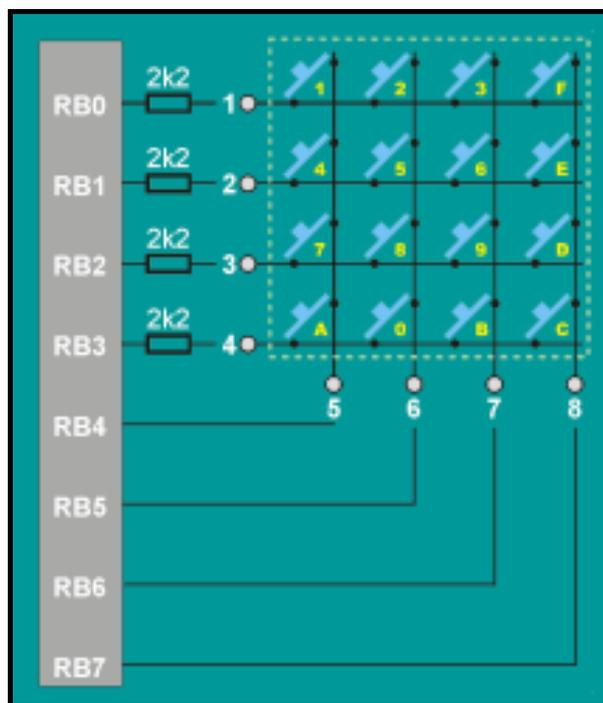


(Disponible en Web: <http://www.x-robotics.com/images/teclado4x4.gif>)

Figura 20. Teclado matricial 4x4

En la figura 21 a continuación, vemos el esquema de conexión interno del teclado matricial y sus correspondientes pines de salida numerados de izquierda a derecha, mirando el teclado tal y como se ve en la foto anterior (figura 20). Cuando se presiona un pulsador se conecta una fila con una columna, teniendo en cuenta este hecho es muy fácil averiguar que tecla fue pulsada. También podemos ver el conexionado típico con el puerto B del μC PIC. Las resistencias de $2\text{k}\Omega$ son necesarias para poder compartir el puerto del PIC independientemente del teclado y por ejemplo poder conectar un LCD o una barra de leds al mismo tiempo. Durante la fase de lectura del teclado la mitad del puerto B es configurada como entrada y la otra mitad como salida y durante la escritura en el LCD u otro sistema, el puerto B es configurado como salida.

Entonces se podrían cortocircuitar accidentalmente las salidas de los puertos provocando su destrucción, si pulsásemos alguna tecla en ese momento, con lo que al poner estas resistencias evitamos este hecho y así si se produjera el cortocircuito tan solo circularía una pequeña corriente y el puerto del μC no correría ningún riesgo.



(Disponible en Web: <http://www.x-robotics.com/images/esquimateclado4x4.gif>)

Figura 21. Conexionado interno de un teclado matricial 4x4 hacia el puerto B

1.4.4 Cargas optoacopladas

Se utiliza opto-acoplamiento para aislar cargas que pudieran introducir interferencias en el funcionamiento del MC. Tal es el caso de los motores, debido al ruido inductivo, o de conmutación en el caso del motor de escobillas que generan. En otros casos se implementa optoaislamiento cuando no se desea compartir la misma referencia para la tarjeta de control y algún elemento externo por razones de seguridad eléctrica. Tal es el caso de muchos equipos destinados a uso médico o a zonas con alto riesgo de explosión. Seguidamente se ofrecen dos alternativas de conexión:

1.4.4.1 Acoplamiento de cargas en régimen de corriente continua

En las figuras 22 y 23 respectivamente se muestra un esquema alternativo para la conexión de una salida y una entrada digital.

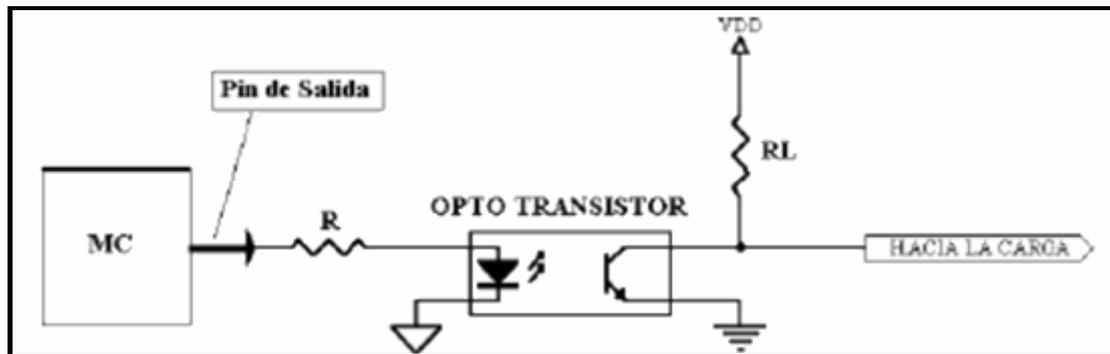


Figura 22. Conexión aislada de un pin de salida que maneja una carga que trabaja en régimen de DC.

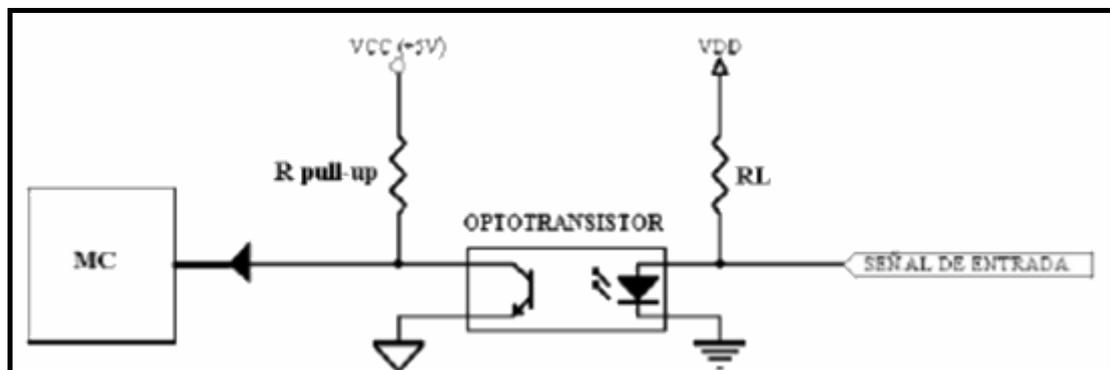


Figura 23. Conexión aislada de una señal a un pin de entrada digital.

En el caso de la salida digital el fototransistor es excitado por la señal de luz que se genera en el led conectado al MC. Observe que el “lado del led” comparte la referencia del MC y el “lado del fototransistor” comparte la referencia de la carga, por lo tanto este es un esquema de conexión aislada.

Por otra parte la entrada digital es conmutada por el fototransistor, cada vez que es activado por los pulsos de luz que se generan en el led a partir de la señal de entrada. Observe que en este caso el “lado del led” comparte la referencia de la señal de entrada y el “lado del fototransistor” comparte la referencia del MC, por lo tanto también este es un esquema de conexión aislada.

1.4.5 Relés

Usualmente la bobina de la mayoría de los relevadores posee baja impedancia y trabaja a un nivel de voltaje superior al de la alimentación del MC. Estas dos razones hacen necesaria la conexión de relés utilizando transistores, que sirven de *buffers* y acopladores de nivel. En la figura 24, se muestra el acople de un relé a una salida digital empleando un transistor NPN. La activación del relé se produce en este caso con un “1” lógico.

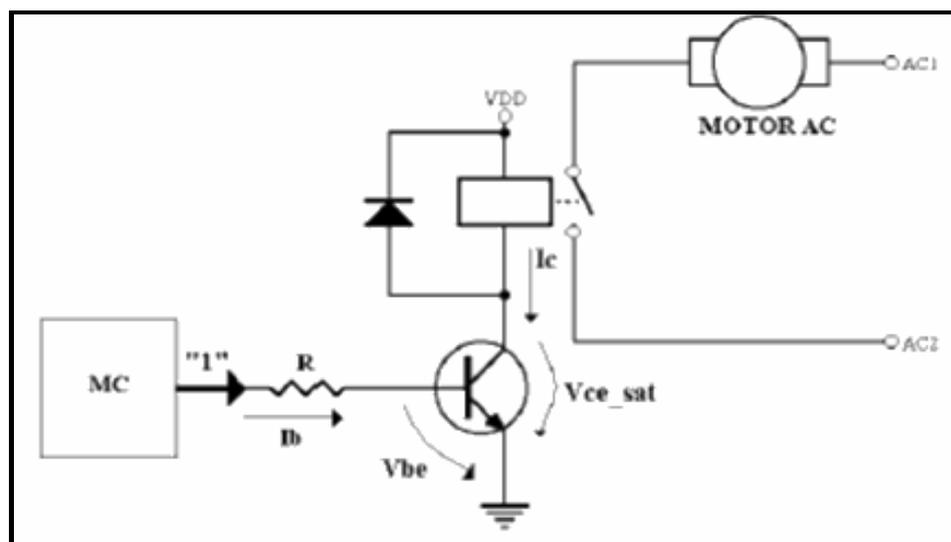


Figura 24. Esquema de conexión para relé.

1.5 Programación del Microcontrolador PIC

1.5.1 Introducción

Los PIC's son microcontroladores contruidos de tal manera que poseen un set de instrucciones reducido (arquitectura RISC). Recordemos que un procesador RISC (*Reduced Instruction Set Computer*) posee las siguientes ventajas:

- Debido a que realizan la canalización (*pipelining*), son mucho más veloces, porque realizan procesamiento paralelo.
- Para los programadores esto es una ventaja substancial pues solo deben aprender un número pequeño de instrucciones (en esta familia, solo un número de 35). A pesar de ser pequeño, este juego de instrucciones tiene casi las mismas prestaciones, en términos de compacidad de código que el juego de los microcontroladores convencionales, que a priori parecen mucho más completos.

Otras características ventajosas de las instrucciones que manejan los MC Microchip, están relacionadas con la filosofía *Harvard* de su arquitectura:

- **Instrucciones simétricas** que hacen posible llevar a cabo cualquier operación sobre cualquier registro usando cualquier modo de direccionado. Ud. se refiere a una localización de RAM con la misma instrucción que a un SFR o a un BIT de una localización o de un puerto.
- **Palabras de instrucciones largas.** Como el bus de instrucciones es separado, este se puede hacer de modo que se cargue de una vez el código de la instrucción.
- **Instrucciones de una sola palabra.** Relacionado con lo anterior, la instrucción se codifica en una sola palabra, en este caso de 14 bits.

1.5.2 Tipos de instrucciones

Las instrucciones se agrupan en 3 categorías:

- Instrucciones orientadas a *Byte*.
- Instrucciones orientadas a *Bits*.
- Instrucciones para operaciones con literales y de control.

1.5.2.1 Orientadas a *Byte*

En estas instrucciones un operando “f” representa un registro determinado y “d” representa el destino que tendrá el resultado de la operación que se realiza, “f” es un número de 7 bits que puede representar un Registro de Propósito Específico (SFR) o un Registro de Propósito General (GPR) dentro de la RAM, “d”, solo puede tener dos valores [0,1], si “d=0” significa que el resultado de la operación realizada será depositado en el registro acumulador (W) y si “d=1” significa que el resultado quedará grabado en el propio registro “f” especificado en la instrucción. A continuación observaremos dos ejemplos de instrucciones orientadas a *Byte*, estas son ADDWF y DECF f,d

ADDWF	suma W y "f"
Sintaxis	ADDWF f,d
operandos	0 ≤ f ≤ 127 d ∈ [0,1]
operación	(W) + (f) → destino
banderas que afecta	C, DC, Z
descripción	Suma el contenido del registro "f" con W. Si d = 0, el resultado se coloca en W, si es = 1 el resultado se coloca en el propio registro "f"
DECF f,d	decrementa el reg "f"
Sintaxis	DECF f,d
operandos	0 ≤ f ≤ 127 d ∈ [0,1]
operación	(f) - 1 → destino
banderas que afecta	Z
descripción	Dec. el contenido del reg "f". Si d = 0 el resultado se coloca en el acumulador y si d = 1 el resultado se coloca en el registro "f".

Figura 25. Instrucciones orientadas a *byte*

1.5.2.2 Orientadas a Bits

Esta categoría está especialmente dedicada al manejo independiente de *Bits*, siguiendo el mismo principio de simetría, o sea con estas instrucciones se pueden manejar por separado bits de cualquier registro, ya sea un Registro de Propósito Específico o un Registro de Propósito General. En estas instrucciones un operando “b” representa el número del *BIT* dentro del registro “f”, que será afectado por la operación, por lo tanto, como todos los registros son solo de 8 bits, “b” es un número decimal que va desde 0 hasta 7; “f” es un operando similar al empleado en la categoría anterior, o sea un número entre 0 y 127 (decimal).

A continuación unos ejemplos:

BCF	lleva a cero el bit "b" del registro "f"
Sintáxis	[label] BCF f,b
operandos	0 ≤ f ≤ 127 0 ≤ b ≤ 7
operación	0 → (f)
banderas que afecta	no afecta ninguna bandera
descripción	Bit 'b' in register 'f' is cleared.
BTFSC	Realiza el "test" del bit "b", de cualquier registro "f" y si es igual a 0, no ejecuta la instrucción que sigue.
Sintáxis	BTFSC f,b
operandos	0 ≤ f ≤ 127 0 ≤ b ≤ 7
operación	skip if (f) = 0
banderas que afecta	ninguna
descripción	Si el bit "b" en el registro "f" es 1 se ejecuta la instrucción que sigue. En este caso el "btfsc" demora un ciclo. Si el bit "b" es igual a cero, entonces no se ejecuta la proxima instrucción. En este caso la demora es de dos ciclos de instrucción.

Figura 26. Instrucciones orientadas a bits

1.5.2.3 Para operar literales y de control

Estas instrucciones, están diseñadas para realizar operaciones aritméticas y lógicas sobre literales (constantes) y para cambiar el contenido del contador de programas de manera incondicional. Emplean el operador “k” que es un valor de 8 ó 9 bits según sea el caso. A continuación observaremos dos ejemplos de instrucciones, el primero para operar literales y el segundo para control:

ADDLW	Add Literal and W	
Sintáxis	ADDLW	k
operandos	$0 \leq k \leq 255$	Note que en este caso "k" es un número de 8 bits, porque es una operación aritmética.
operación	$(W) + k \rightarrow (W)$	
banderas que afecta	C, DC, Z	
descripción	Suma el contenido del acumulador con el número de 8 bits "K" y coloca el resultado siempre en el acumulador.	
CALL	Llamada a Subrutina.	
Sintáxis	CALL	k
operandos	$0 \leq k \leq 2047$	En este caso "k" es un número entre cero y 3ff
operación	$(PC) + 1 \rightarrow TOS,$ $k \rightarrow PC<10:0>,$ $(PCLATH<4:3>) \rightarrow PC<12:11>$	
banderas que afecta	Ninguna.	
descripción	Primero, guarda la dirección PC+1 en la pila. Los 11 primeros bits de la dirección del salto se cargan en el PC (bits 10 : 0). Los bits superiores (11 y 12) se cargan desde el registro PCLATH, con esto se forma al fin la dirección de 13 bits. El CALL es una instrucción que siempre dura dos ciclos.	

Figura 27: Operaciones con literales (arriba) y de control (abajo).

1.5.3 Formato de codificación

Las instrucciones Microchip se codifican en una sola palabra, en el caso de la familia media la longitud de esta palabra es de 14 bits. Dentro de la palabra de 14 bits, según la categoría o tipo de instrucción, existen grupos de bits que codifican cada parte de la instrucción.

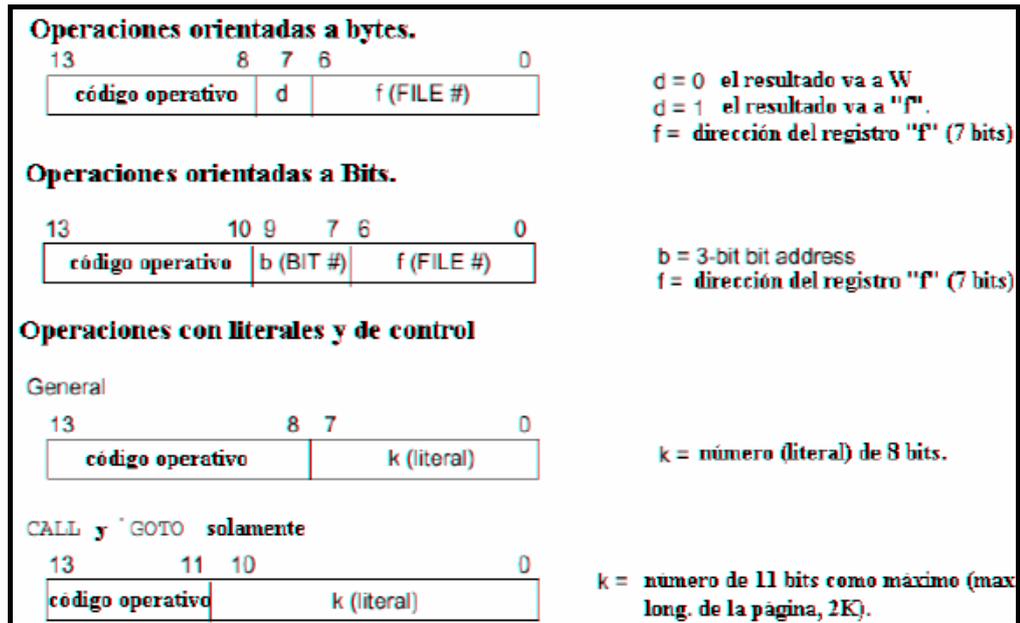


Figura 28. Grupos de bits que codifican cada parte de la instrucción

1.5.4 Tiempo de duración de las instrucciones

La mayoría de las instrucciones Microchip se ejecutan en un solo ciclo de instrucción, que es igual a 4 veces el período del reloj externo, aunque existen las siguientes excepciones:

Se ejecutarán siempre en dos ciclos de instrucción:

CALL k Llamada a una subrutina.

GOTO k Salto incondicional a la dirección "k".

RETFIE Retorno de una subrutina de atención a interrupción.

RETURN Retorno de un CALL.

RETLW k Retorno de un CALL pero grabando el número “k” en el Acumulador.

Se ejecutarán en uno o dos ciclos de instrucción:

DECFSZ f,d Decrementa el registro “f” y salta la instrucción siguiente si “f” llegó a cero (en este caso dura dos ciclos).

INCFSZ f,d Similar, pero incrementando “f”, o sea se salta la instrucción siguiente cuando “f” sufre un “*overflow*”. En este caso la instrucción demora dos ciclos.

BTFSC f,b Hace el test del BIT “b” dentro del registro “f” y salta la próxima instrucción cuando este está en cero. En este caso la instrucción dura dos ciclos.

BTFSS f,b Hace el test del BIT “b” dentro del registro “f” y salta la próxima instrucción cuando este está en uno. En este caso la instrucción dura dos ciclos.

1.6 Interrupciones, timers y contadores

1.6.1 Trabajo con interrupciones

El trabajo con interrupciones (INT) se caracteriza por ser un servicio directo entre periféricos y la CPU, siempre que esta “desea establecer el diálogo con el periférico”. En términos concretos, esto significa que la interrupción del periférico había sido habilitada. La característica fundamental y ventaja principal del servicio de interrupciones es la inmediatez; lo que significa que pueden eliminarse total o parcialmente los ciclos de consulta. Otra ventaja es que puede inhibirse la interrupción del periférico dado cuando se considera que esta es “inoportuna” y por tanto perjudica en alguna medida la marcha del proceso principal, o sea la corrida del resto del programa. Esta forma de trabajo es indispensable en el trabajo de control de procesos en tiempo real. Una de las formas de iniciar una interrupción es por medio de un periférico o entidad externa ajena al MC a través de los pines específicos del mismo; en estas condiciones la interrupción es externa. Pero cuando la interrupción inicia dentro del propio chip MC entonces la interrupción es interna.

1.6.2 Fuentes de interrupciones

Los PICS pueden manejar diversas fuentes de interrupción. Por lo general con cada periférico integrado al MC hay asociada una o más fuentes de interrupción. A continuación se resumen todas las fuentes de INT posibles en los MC de esta familia. Los tipos de fuente de INT se subdividen en dos categorías: Básicas y Adicionales.

1.6.2.1 Básicas

- Interrupción por pin externo (pin RB0/INT). El pin RB0 es un pin de puerto que está multiplexado con la función de INT externa.
- Por desbordamiento del TMR0.
- Por cambio en los pines de puerto RB7: RB4.
- Cuando se completa un ciclo de escritura en la EEPROM de Datos. La EEPROM de datos es otro periférico muy común en los MC de esta familia. Es una memoria tipo EEPROM y por lo tanto NO volátil, que puede escribirse y leerse desde el programa de aplicación.

1.6.2.2 Adicionales

- INT por desbordamiento del TMR1.
- INT por desbordamiento del TMR2.
- Por cambio en el Módulo Comparador (este se refiere a un comparador analógico que puede estar incluido en algunos MC de esta familia).
- INT del Puerto Paralelo Esclavo.
- Interrupciones varias del USART (*Universal Synchronous Asynchronous Receiver Tx*).
- INT por completamiento del Ciclo de Conversión en los conversores A/D.

- INT del Módulo LCD.
- INT por el Módulo CCP.
- INT por el Módulo SSP (*Serial Synchronous Port*).

1.6.3 Secuencia de una interrupción

En los MC PICS cuando algún periférico (interno o externo) solicita INT, ocurre la siguiente secuencia de eventos:

- Se completa la ejecución de la instrucción que se está ejecutando en ese momento.
- Se incrementa el PC y se guarda en la PILA (“*STACK*”).
- El PC (*Program Counter* o Contador del Programa) se carga con la dirección 0004 de la Memoria de Programas (PC = 0004), que es donde se encuentra ensamblado el “Vector de Interrupción”. El Vector de Interrupción es una instrucción de salto¹⁵ (*goto* INT, por ejemplo) a partir de la cual se ensambló la subrutina de atención a interrupción.
- Una vez dentro de la subrutina de atención a interrupción se procede a identificar la fuente de interrupción (encuestando las banderas correspondientes) y a “atender” el periférico que en ese momento solicita la interrupción. Para ello todos los PICS poseen registros donde se encuentran las banderas que sirven de aviso sobre el periférico que generó el pedido de interrupción.
- Finalmente antes de abandonar la subrutina de atención, deben limpiarse las banderas del o los periférico(s) que fueron “atendidos” en ella para evitar una interrupción reentrante. De la subrutina se sale con la instrucción “*retfie*”, que es la encargada de cargar el PC con el valor de retorno que había sido guardado en la PILA.

¹⁵ **Nota:** el salto a una INT demora 3 ciclos de máquina.

1.6.4 Registros para el manejo de interrupciones

Para el manejo de INT en los PICS existen dos tipos de registros especiales: los registros de control y los registros banderas de interrupción. Los primeros permiten habilitar las distintas fuentes de INT y los segundos son necesarios para averiguar cuál periférico generó la INT. La manera más fácil de estudiar estos registros es relacionarlos con el periférico del CHIP que se esté empleando.

1.6.4.1 Registro de Control de interrupciones básicas INTCON

El registro INTCON (figura 29), aparece desde los procesadores más primitivos de la familia media (PIC16F84, por ejemplo). En este registro se incluyen los bits para habilitar las fuentes básicas de interrupción, así como los bits que sirven de banderas para c/u de las fuentes individuales de interrupción.

REGISTER 2-3: INTCON REGISTER (ADDRESS 0Bh, 8Bh, 10Bh, 18Bh)							
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-x
GIE	PEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF
bit7							bit0
bit 7:	GIE: Global Interrupt Enable bit 1 = Enables all un-masked interrupts 0 = Disables all interrupts						
bit 6:	PEIE: Peripheral Interrupt Enable bit 1 = Enables all un-masked peripheral interrupts 0 = Disables all peripheral interrupts						
bit 5:	TOIE: TMR0 Overflow Interrupt Enable bit 1 = Enables the TMR0 interrupt 0 = Disables the TMR0 interrupt						
bit 4:	INTE: RB0/INT External Interrupt Enable bit 1 = Enables the RB0/INT external interrupt 0 = Disables the RB0/INT external interrupt						
bit 3:	RBIE: RB Port Change Interrupt Enable bit 1 = Enables the RB port change interrupt 0 = Disables the RB port change interrupt						
bit 2:	TOIF: TMR0 Overflow Interrupt Flag bit 1 = TMR0 register has overflowed (must be cleared in software) 0 = TMR0 register did not overflow						
bit 1:	INTF: RB0/INT External Interrupt Flag bit 1 = The RB0/INT external interrupt occurred (must be cleared in software) 0 = The RB0/INT external interrupt did not occur						
bit 0:	RBIF: RB Port Change Interrupt Flag bit 1 = At least one of the RB7:RB4 pins changed state (must be cleared in software) 0 = None of the RB7:RB4 pins have changed state						

R = Readable bit
 W = Writable bit
 U = Unimplemented bit, read as '0'
 - n= Value at POR reset

Figura 29. Registro INTCON. Fuente: *DataSheet Serie 87X*

El registro INTCON se encuentra en las localizaciones: 0Bh, 8Bh, 10Bh y 18Bh de los bancos 0, 1, 2 y 3 de la Memoria de Datos; por lo tanto para escribir o leer su contenido NO es necesario cambiar de banco modificando los bits correspondientes del registro *STATUS*.

1.6.4.2 Registros de Control y de Banderas de interrupciones adicionales

La denominación de los registros para el control de interrupciones adicionales se codifica de manera general con las letras PIE (*Peripheral Interrupt Enable register*), más un número (por ejemplo PIE1, PIE2, etc.). En un mismo PIC pueden existir varios de estos registros en función de la cantidad de periféricos integrados al MC que sean capaces de generar interrupción. Por otra parte, las banderas que indican los eventos de interrupción adicionales, se agrupan en registros tipo PIR (*Peripheral Interrupt Registers*). De manera similar a lo que ocurre con los registros tipo PIE, pueden existir varios tipos PIR (por ejemplo, PIR1, PIR2, etc.). Las figuras 30 y 31 ilustran dos tipos de registros: uno de control (PIE1) y otro de banderas (PIR1). Note que los bits de habilitación y bandera se encuentran en posiciones similares dentro de ambos registros; por ejemplo: el bit 0 del registro PIE1, sirve para habilitar la interrupción del timer1 y el bit 0 del registro PIR1 sirve de bandera de interrupción por desbordamiento del timer1.

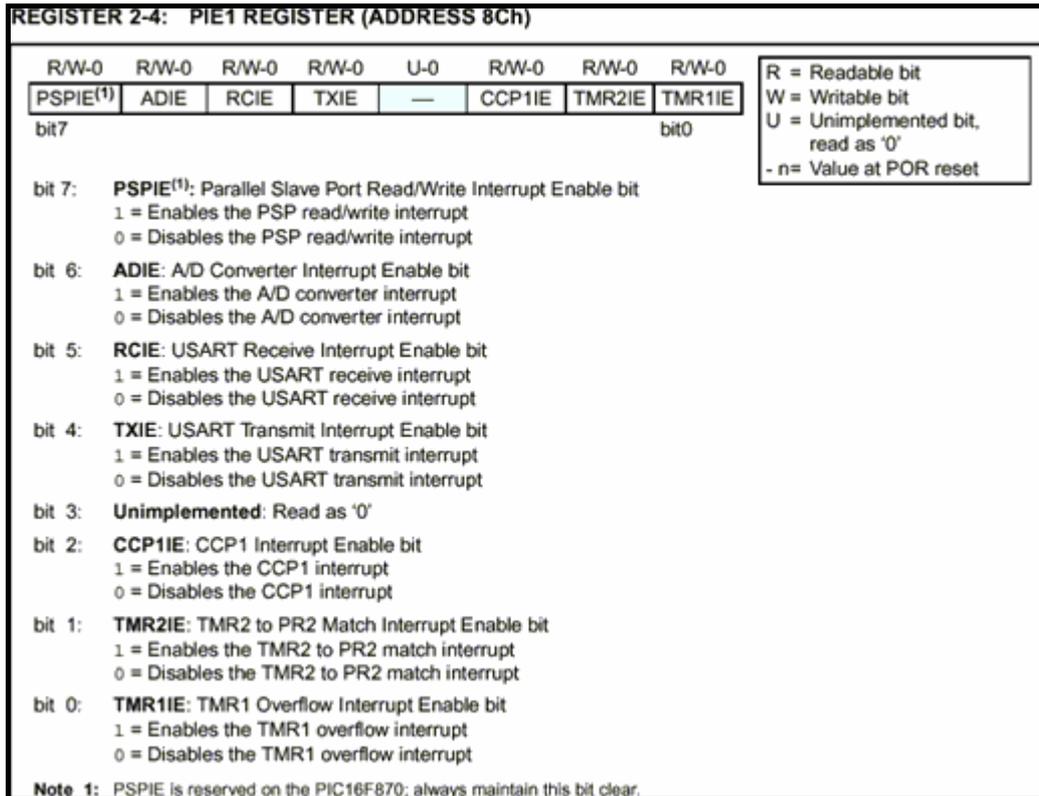


Figura 30. Registro de Habilitación de Interrupciones PIE1.

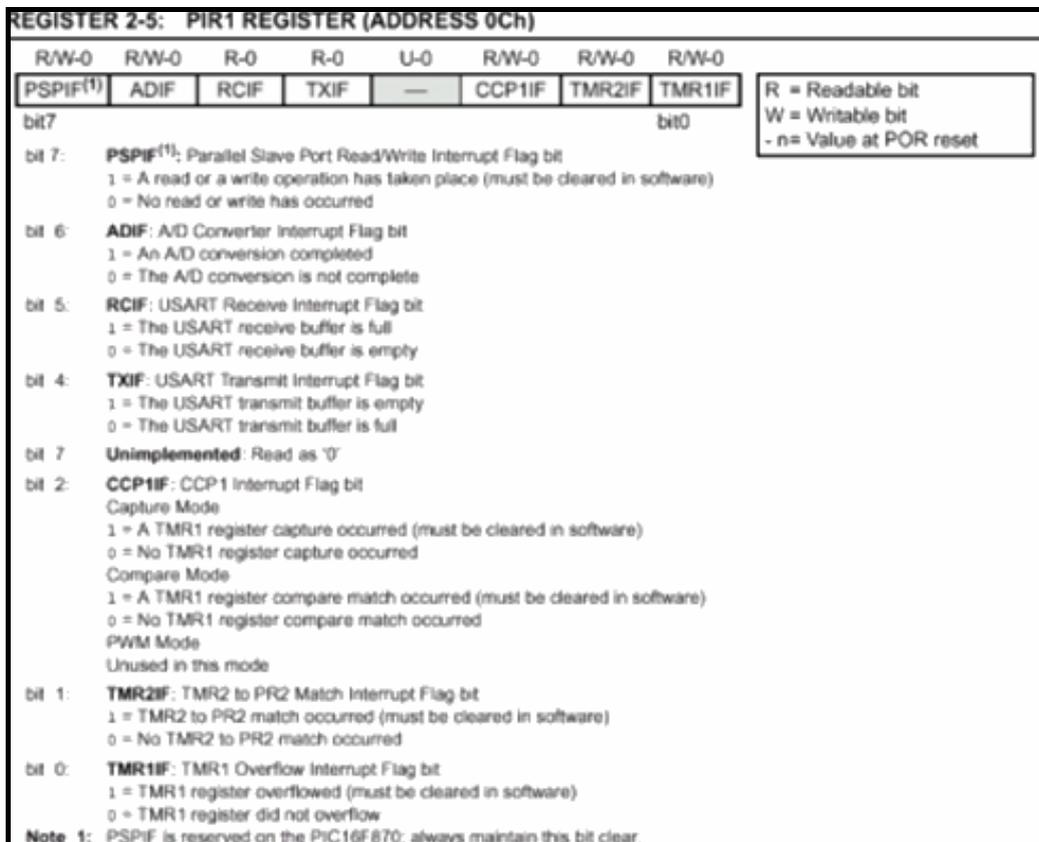


Figura 31. Registro de Banderas de Interrupciones PIR1.

1.6.5 Módulo TMR0

El módulo TMR0 es el timer/contador primario que incluyen los Microcontroladores PICS que poseen timer.

1.6.5.1 Especificaciones técnicas generales

Puede ser programado como temporizador (*timer*) o contador (*counter*) de 8 BITS ascendente.

El conteo o la temporización se lleva en un registro (TMR0), que se puede leer y escribir.

Posee un PSCALER programable por SW de 8 BITS.

Fuente de reloj seleccionable (interna o externa), según se trabaje como temporizador o contador.

Genera una INT por desbordamiento (FF = 00) si el BIT T0IE del registro INTCON es igual a 1.

Cuando se utiliza para contar eventos externos se puede seleccionar el frente de conteo, mediante el BIT T0SE del registro "OPTION".

1.6.5.2 Arquitectura interna

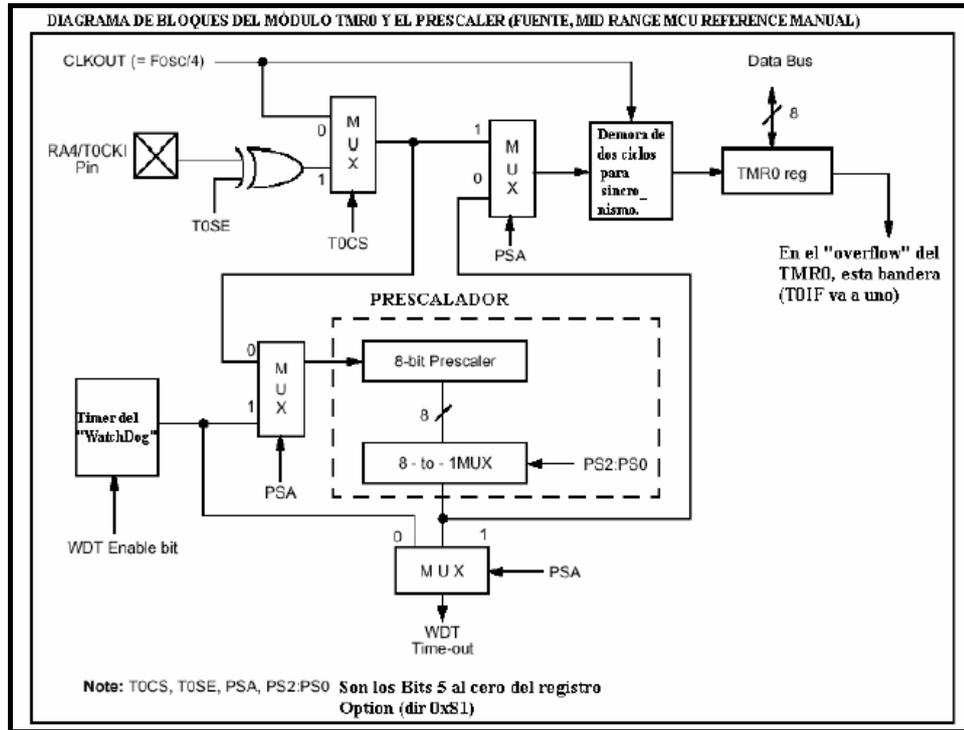


Figura 32. Diagrama de Bloques del Timer 0.

1.6.5.3 Registros para el Control del Timer 0

1.6.5.3.1 Registro OPTION (dir. 81h del banco 1)

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1																											
RBPU	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0																											
bit7							bit0																											
<p>bit 7: RBPU: Habilitación de los PULL UPS del puerto B. 1 = Inhabilitados. 0 = Habilitados.</p> <p>bit 6: INTEDG: Frente de disparo de la INT por pin RB0 1 = Con la subida 0 = Con la caída.</p> <p>bit 5: T0CS: Selección de la fuente de reloj para el Tmr0. 1 = pin RA4/T0CKL (modo contador) 0 = cuenta ciclos de instrucción (modo "timer").</p> <p>bit 4: T0SE: Frente de conteo por el PIN RA4/T0CKL (modo contador). 1 = cuenta con la subida 0 = cuenta con la caída.</p> <p>bit 3: PSA: Bit de asignación del Prescaler. 1 = Pscaler al watch Dog. 0 = Pscaler al Tmr0.</p> <p>bit 2-0: PS2:PS0: Factor de división del Pscaler.</p>																																		
						<table border="1"> <thead> <tr> <th>Bit Value</th> <th>TMR0 Rate</th> <th>WDT Rate</th> </tr> </thead> <tbody> <tr> <td>000</td> <td>1:2</td> <td>1:1</td> </tr> <tr> <td>001</td> <td>1:4</td> <td>1:2</td> </tr> <tr> <td>010</td> <td>1:8</td> <td>1:4</td> </tr> <tr> <td>011</td> <td>1:16</td> <td>1:8</td> </tr> <tr> <td>100</td> <td>1:32</td> <td>1:16</td> </tr> <tr> <td>101</td> <td>1:64</td> <td>1:32</td> </tr> <tr> <td>110</td> <td>1:128</td> <td>1:64</td> </tr> <tr> <td>111</td> <td>1:256</td> <td>1:128</td> </tr> </tbody> </table>		Bit Value	TMR0 Rate	WDT Rate	000	1:2	1:1	001	1:4	1:2	010	1:8	1:4	011	1:16	1:8	100	1:32	1:16	101	1:64	1:32	110	1:128	1:64	111	1:256	1:128
Bit Value	TMR0 Rate	WDT Rate																																
000	1:2	1:1																																
001	1:4	1:2																																
010	1:8	1:4																																
011	1:16	1:8																																
100	1:32	1:16																																
101	1:64	1:32																																
110	1:128	1:64																																
111	1:256	1:128																																

Figura 33. Registro OPTION

1.6.5.3.2 Registro INTCON (dir. 0h, 8Bh, 10Bh, 18Bh).

Este registro permite controlar las interrupciones de los periféricos básicos, incluyendo la del *Timer 0*.

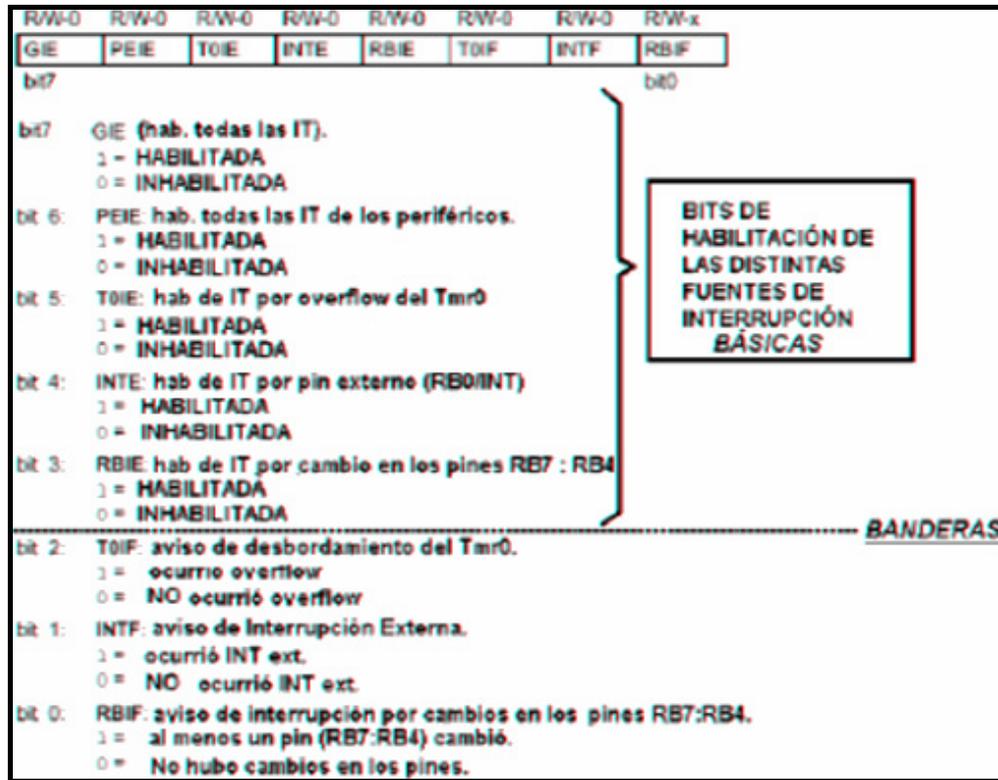


Figura 34. Registro *INTCON*

1.6.5.4 Modos de operación del TMR0

1.6.5.4.1 Modo *Timer*

Se selecciona haciendo 0 el bit T0CS del reg. *OPTION* (81h). Si no se selecciona el *PSCALER*, en este modo el timer cuenta Ciclos de Máquina ($fosc/4$). El resultado de la temporización se puede leer en el registro *TMR0*, que se encuentra en la dirección 01h del banco 0 exclusivamente. El registro *TMR0* también puede ser alterado por el programador, o sea durante la temporización o el conteo. Si se altera el *TMR0* su incremento se retarda en dos Ciclos de Máquina. Si se utiliza el *Pscaler* ($PSA = 0$), el reloj de entrada al *timer* ($Fosc/4$) se divide por los siguientes factores (Figura 35), según

la combinación que se coloque en los bits PS2:PS0 del registro “*OPTION*”.

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

Figura 35. Factores de división del PSCALER para el tmr0 y el *Watch Dog*.

1.6.5.4.2 Modo Contador

Se selecciona poniendo el BIT TOCS = 1, con esto se habilita el conteo de eventos externos por el pin Ra4, con el flanco de subida o de caída, según el valor del BIT 4 (T0SE), del registro “*OPTION*”. Si T0SE = 1, cuenta con la caída y en caso contrario con la subida.

Nota: recuerde que el PSCALER es un divisor de frecuencia compartido con el “*Watch Dog*” Si Ud. utiliza “*Watch Dog*” y timer0 y no desea PSCALER para ninguno de los dos periféricos, simplemente ponga el bit PSA en 1 y los bits PS2:PS0 en cero.

1.7 Comunicación I2C y Modulo SSP

1.7.1 Introducción

Phillips Semiconductors inventó el bus de dos alambres I2C para comunicación entre ICs en 1980 y desde entonces, se ha convertido en el bus serial estándar, implementado en un gran número de ICs y con licencias otorgadas a más de 50 compañías con un total de 1000 dispositivos compatibles I2C. Originalmente especificado para 100 kbits/s e intencionalmente para control simple y señales de status, el bajo costo, la versatilidad técnica y la simplicidad del bus I2C aseguraron su popularidad. El bus serial I2C ha

sido extendido para soportar velocidades de hasta 3.4 Mbits/s. Combinado con una función de desplazamiento del nivel de voltaje, en modo *High-speed* (Hs-mode) ofrece una solución ideal para los sistemas de tecnología mezclada, donde las altas velocidades y la variedad de voltajes (5 V, 3 V o menor) son comúnmente usados.

El modo Hs es compatible con todos los sistemas existentes del bus I2C, incluyendo el estándar original (S-mode) y el modo *Fast* (F-mode), actualización introducida en 1992, proveyendo 400 kbits/s en transferencia. Diferentes sistemas de velocidad pueden ser mezclados fácilmente, con un dispositivo maestro en modo Hs especialmente desarrollado, la conexión en paralelo es usada para conectar las partes más lentas del sistema, todos los dispositivos en modo Hs se comunicarán bi-direccionalmente a toda velocidad mientras se utilicen los modos F/S (*Fast/Standard*) de velocidades con las partes más lentas del sistema. Phillips fue su inventor hace 20 años atrás, y hoy está firmemente establecido como solución mundial para aplicaciones integradas. Usado en gran variedad de microcontroladores y aplicaciones de telecomunicaciones como en control, diagnóstico y administración de potencia. Su simplicidad ha sido retenida independiente de las mejoras a la especificación original.

1.7.2 El Bus I2C (Inter IC)

El bus I2C es una interface serial de dos alambres desarrollada por la Corporación Philips ®. La especificación original, o modo de estándar, fue para transferencia de datos hasta 100 Kbps. La especificación mejorada (modo rápido) está también implementada en los PIC's. La comunicación entre dispositivos se puede realizar en el modo estándar y en el modo rápido si los dispositivos están unidos al mismo bus. El *clock* determinará la velocidad de los datos. La interface I2C emplea un protocolo amplio para asegurar una transmisión y recepción de datos fiable.

1.7.2.1 Introducción de las especificaciones I2C

Está orientado a las aplicaciones de 8-bit controladas por un microprocesador y estas son básicamente los criterios que se deben establecer:

- Un sistema consiste en al menos un microcontrolador y varios sistemas periféricos como memorias o circuitos diversos.
- El costo de conexión entre los varios dispositivos dentro del sistema debe de ser el mínimo.
- El sistema que utiliza este Bus no requiere una alta tasa de transferencia de datos
- La total eficacia del sistema depende de la correcta selección de la naturaleza de los dispositivos y de la interconexión de la estructura del bus.

1.7.2.2 El concepto del Bus I2C

El bus I2C soporta cualquier tipo de componente (NMOS, CMOS, bipolar, etc.). Dos hilos físicos uno de datos (SDA) y otro de reloj (SCL) transportan la información entre los diversos dispositivos conectados al bus. Cada dispositivo es reconocido por una única dirección (si es un microcontrolador, LCD, memoria o teclado) y puede operar cualquiera como transmisor o emisor de datos, dependiendo de la función del dispositivo. Un *display* es solo un receptor de datos mientras que una memoria recibe y transmite datos. En función de que envíe o reciba datos se debe considerar los dispositivos como Maestros (*Master*) o esclavos (*Slaves*).

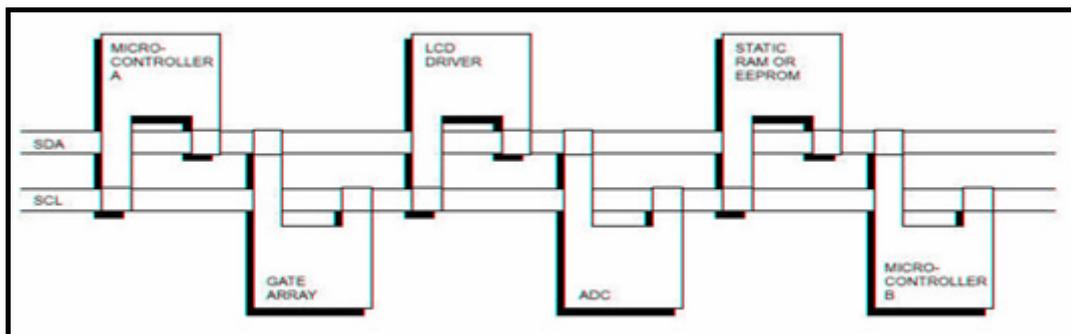


Figura 36. Configuración de comunicación I2C usando dos microcontroladores

Terminología básica del bus I2C

TRANSMISOR:	El dispositivo que envía datos al Bus.
RECEPTOR:	El dispositivo que recibe datos desde el Bus.
MAESTRO (MASTER):	El dispositivo que inicia una transferencia, genera las señales del reloj y termina un envío de datos.
ESCLAVO (SLAVE):	El dispositivo direccionado por un maestro.
MULTI-MAESTRO:	Más de un maestro puede controlar el Bus al mismo tiempo sin corrupción de los mensajes.
ARBITRAJE:	Procedimiento que asegura que si uno o más maestros simultáneamente deciden controlar el Bus, solo uno es permitido a controlarlo y el mensaje saliente no es deteriorado.
SINCRONIZACIÓN:	Procedimiento para sincronizar las señales del reloj de dos o más dispositivos.

1.7.2.3 Generalidades

Los maestros son generalmente microcontroladores, por lo que un microcontrolador puede ser unas veces maestro y otras esclavo.

Para trasladar a un circuito eléctrico la imagen del Bus, son dos cables a los que se conectan diversos circuitos o chips en cantidad variable según las necesidades, controlado el conjunto por uno o mas microcontroladores que dan instrucciones para el buen funcionamiento del conjunto. La posibilidad de conectar mas de un microcontrolador al Bus significa que uno o más microcontroladores pueden iniciar el envío de datos al mismo tiempo. Para prevenir el caos que esto ocasionaría se ha desarrollado un sistema de arbitraje.

Si uno o mas maestros intentan poner información en el Bus, es la señal del reloj si esta a "1" o a "0" lo que determina los derechos de arbitraje. La generación de señales de reloj (SCL) es siempre responsabilidad de los dispositivos Maestro, cada Maestro genera su propia señal de reloj cuando envía datos al Bus, las señales de reloj de un Maestro solo pueden ser alteradas cuando la línea de reloj sufre una caída por un dispositivo Esclavo o por el dominio del control del Bus por el arbitraje de otro

microcontrolador.

Los dispositivos conectados al Bus deben ser de colector abierto o drenaje abierto ("en paralelo"), así los estados de salida de las líneas de reloj (SCL) y dato (SDA) desempeñan la función de "cable en *AND*" del Bus. Durante el tiempo en que no hay transferencia de datos (tiempo inactivo), tanto la línea del reloj (SCL) como la línea de datos (SDA) son "tiradas" arriba a través de resistencias externas *pull-up*. La única limitación en la conexión de dispositivos al Bus depende de la capacidad máxima que no puede superar los 400 pF. Los tipos de transferencia de datos en el bus son:

- Modo Estándar aproximadamente a 100 kBits/Sg.
- Modo Rápido aproximadamente a 400kbits/Sg.
- Modo Alta velocidad más de 3,4 Mbits/Sg.

1.7.2.4 Características Generales

Tanto la línea de datos (a partir de ahora SDA) como la Señal de Reloj (a partir de ahora SCL) son bidireccionales conectadas a una fuente de tensión positiva vía suministro común o resistencias de carga. Ver Figura. 37

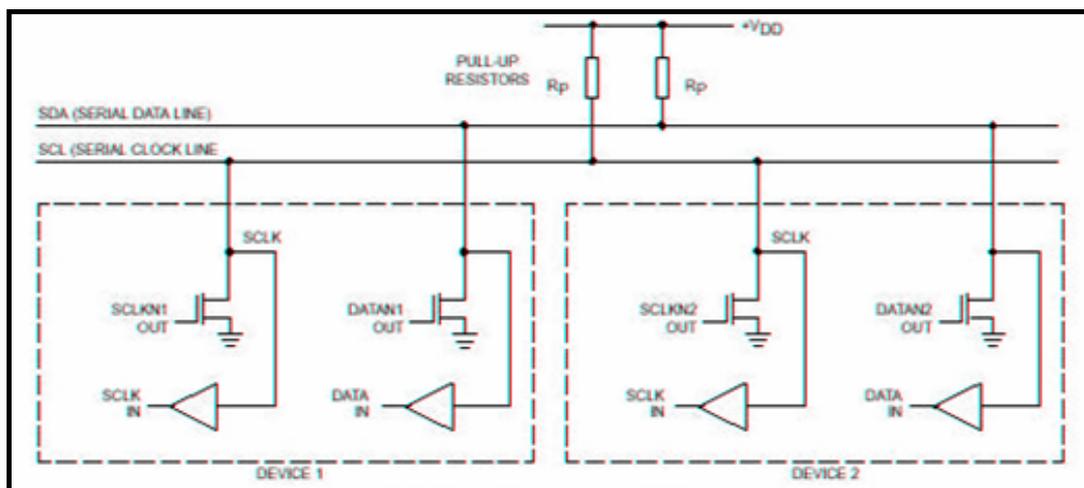


Figura 37. Conexión del Bus I2C entre dos dispositivos

1.7.2.5 Transferencia del Bit

Debido a la variedad de diferentes tecnologías usadas en los dispositivos conectados al Bus I2C los niveles lógicos de "0" (Bajo) y "1" (Alto) no están fijados y dependen de la tensión de alimentación del circuito. Un pulso de reloj se genera por cada bit de datos transferidos.

Los bits de datos transferidos en la línea SDA deben ser estables cuando la línea SCL esta a nivel "1". El estado de la línea SDA en "1" o "0" solo puede cambiar cuando en la línea SCL la señal es "0" Ver Fig. 38.

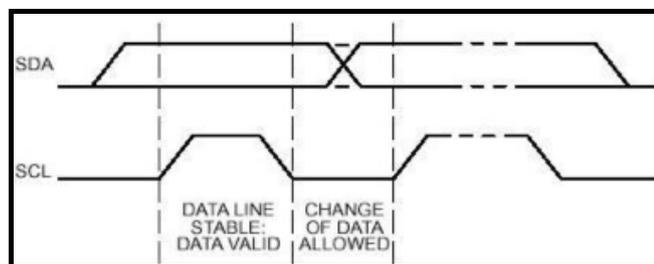


Figura 38. Transferencia del Bit en el Bus I2C

1.7.2.6 Instrucciones en el Bus I2C

Para operar un esclavo sobre el Bus I2C solo son necesarios seis simples códigos, suficientes para enviar o recibir información.

- Un bit de Inicio
- 7-bit o 10-bit de direccionamiento
- Un R/W bit que define si el esclavo es transmisor o receptor
- Un bit de reconocimiento
- Mensaje dividido en bytes
- Un bit de Stop

Estos campos se explican luego.

Condiciones de Inicio (*Start*) y Parada (*Stop*)

Dentro del proceso de transferencia de datos en el Bus I2C hay dos situaciones básicas que son el Inicio y el *Stop* de toda transferencia de datos. Estas son:

INICIO (*START*) - Una transición de "1" a "0" (caída) en la línea de datos (SDA) mientras la línea del reloj (SCL) esta a "1".

PARADA (*STOP*) - Una transición de "0" a "1" (ascenso) en la línea de datos (SDA) mientras la línea de reloj (SCL) esta a "1" . Ver Fig. 39.

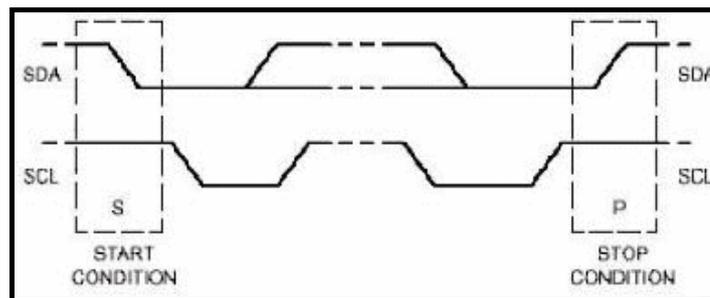


Figura 39. Condiciones de Inicio y Parada

Las condiciones de Inicio y Parada son siempre generadas por el Maestro. El Bus I2C se considera ocupado después de la condición de Inicio. El Bus se considera libre de nuevo después de un cierto tiempo tras la condición de Parada.

Es decir al pulso "1" de la línea SCL le puede corresponder un pulso "0" o "1" de la línea SDA en función de la información del byte que se envíe , recordemos que a cada bit de SDA le corresponde un bit de SCL , pero nunca, salvo en la condición de Inicio, a un bit de SCL le corresponde una situación de "1" a "0" o sea pasa por dos estados la línea SDA, al revés ocurre en la condición de Parada que el Maestro envía un bit a la línea SCL mientras cambia en la SDA de "0" a "1" durante el tiempo que esta enviando la señal de "1" a SCL.

1.7.2.7 Transfiriendo datos

El numero de *bytes* que se envían a la línea SDA no tiene restricción. Cada *byte* debe ir seguido por un bit de reconocimiento, el *byte* de datos se transfiere empezando por el *bit* de mas peso (7) precedido por el *bit* de reconocimiento (ACK). Ver Figura. 40.

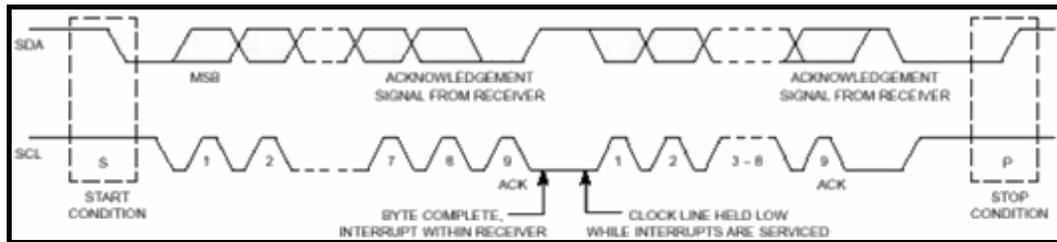


Figura 40. Transferencia de datos en el Bus I2C

Si un dispositivo esclavo no puede recibir o transmitir un byte de datos completo hasta que haya acabado alguno de los trabajos que realiza, puede mantener la línea SCL a "0" lo que fuerza al Maestro a permanecer en un estado de espera. Los datos continúan transfiriéndose cuando el dispositivo esclavo esta listo para otro *byte* de datos y desbloquea la línea de reloj SCL.

1.7.2.8 Reconocimiento

El bit de reconocimiento es obligatorio en la transferencia de datos. El pulso de reloj correspondiente al bit de reconocimiento (ACK) es generado por el Maestro. El transmisor desbloquea la línea SDA ("1") durante el pulso de reconocimiento. El receptor debe poner a "0" la línea SDA durante el pulso ACK de modo que siga siendo "0" durante el tiempo que el maestro genera el pulso "1" de ACK. Ver Figura 41.

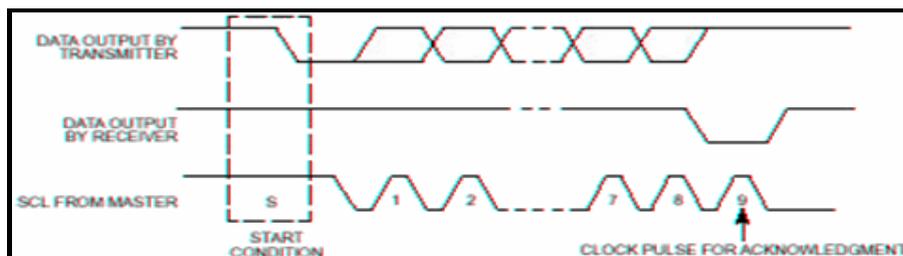


Figura 41. Reconocimiento en el Bus I2C

Normalmente un receptor cuando ha sido direccionado esta obligado a generar un ACK después de que cada *byte* a sido recibido. Cuando un dispositivo esclavo no genera el bit ACK (porque esta haciendo otra cosa y no puede atender el Bus) debe mantener en el Esclavo la línea SDA a nivel "1" durante el bit ACK. El Maestro entonces puede generar una condición de *STOP* abortando la transferencia de datos o repetir la condición de Inicio enviando una nueva transferencia de datos.

Si un Esclavo-receptor que esta direccionado no desea recibir mas *bytes*, el Maestro debe detectar la situación y no enviar mas *bytes*. Esto se indica porque el Esclavo no genera el bit ACK en el primer byte que sigue. El esclavo pone la línea SDA a "1" lo que es detectado por el Maestro el cual genera la condición de Parada o repite la condición de Inicio. Si un Maestro-receptor esta recibiendo datos de un Esclavo-transmisor debe generar un bit ACK tras cada *byte* recibido de transmisor, para finalizar la transferencia de datos no debe generar el ACK tras el ultimo *byte* enviado por el esclavo. El esclavo-transmisor debe permitir desbloquear la línea SDA generando el master la condición de parada o de Inicio.

1.7.2.9 Arbitraje y generación de señales de Reloj

Sincronización

Todos los *Master* generan su propia señal de reloj sobre la línea SCL al transferir datos sobre el Bus I2C. Los bit de datos son solo validos durante los periodos "1" del reloj. Un control es necesario para mantener un orden en los diversos bit que se generan. La sincronización del reloj se realiza mediante una conexión *AND* de todos los dispositivos del Bus a la línea SCL. Esto significa que una transición de un *Master* de "1" a "0" en la línea SCL hace que la línea pase a "0", esto mantiene la línea SCL en ese estado. Sin embargo la transición de "0" a "1" no cambia el estado de la línea SCL si otro reloj esta todavía en su periodo de "0". Por lo tanto la línea SCL permanecerá a "0" tanto como el periodo mas largo de cualquier dispositivo cuyo nivel sea "0". Los dispositivos que tienen un periodo mas corto de reloj "0" entran en un periodo de espera.

Cuando todos los dispositivos conectados al Bus han terminado con su periodo "0", la línea del reloj se desbloquea y pasa a nivel "1". Por lo que hay que diferenciar entre los estados de reloj de los dispositivos y los estados de la línea SCL, y todos los dispositivos empiezan a nivel "1". El primer dispositivo que completa su nivel "1" pone nuevamente la línea SCL a "0". Resumiendo, la sincronización de la línea SCL se genera a través de la señal a "0" por el dispositivo con el mas largo periodo de nivel a "0", y la señal a "1" por el dispositivo con el mas corto periodo de nivel a "1". Ver figura 42.

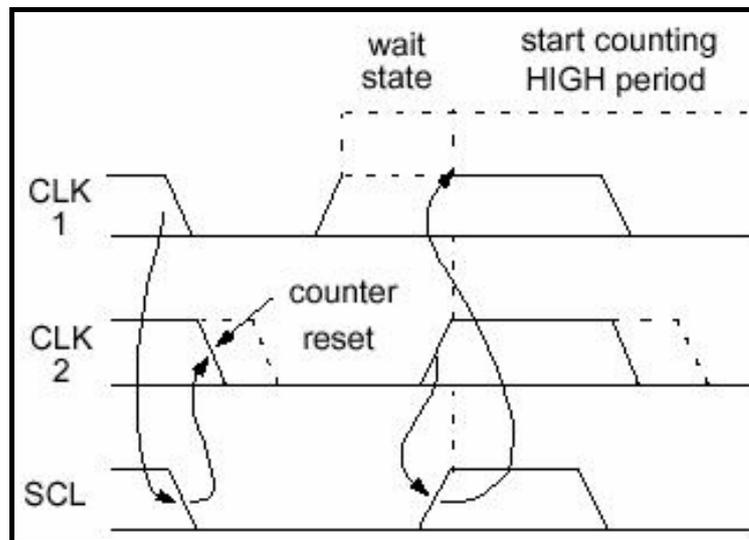


Figura 42. Sincronización del Reloj

Arbitraje

Un *master* puede iniciar una transmisión solo si el bus esta libre. Dos o más *master* pueden generar una condición de Inicio en el bus lo que da como resultado una condición de Inicio general. Cada *Master* debe comprobar si el bit de datos que transmite junto a su pulso de reloj, coincide con el nivel lógico en la línea de datos SDA. El sistema de arbitraje actúa sobre la línea de datos SDA, mientras la línea SCL esta a nivel "1", de una manera tal que el *master* que transmite un nivel "1", pierde el arbitraje sobre otro *master* que envía un nivel "0" a la línea de datos SDA. Esta situación continua hasta que se detecte la condición de *Stop* generada por el master que se hizo cargo del Bus. Ver figura 43.

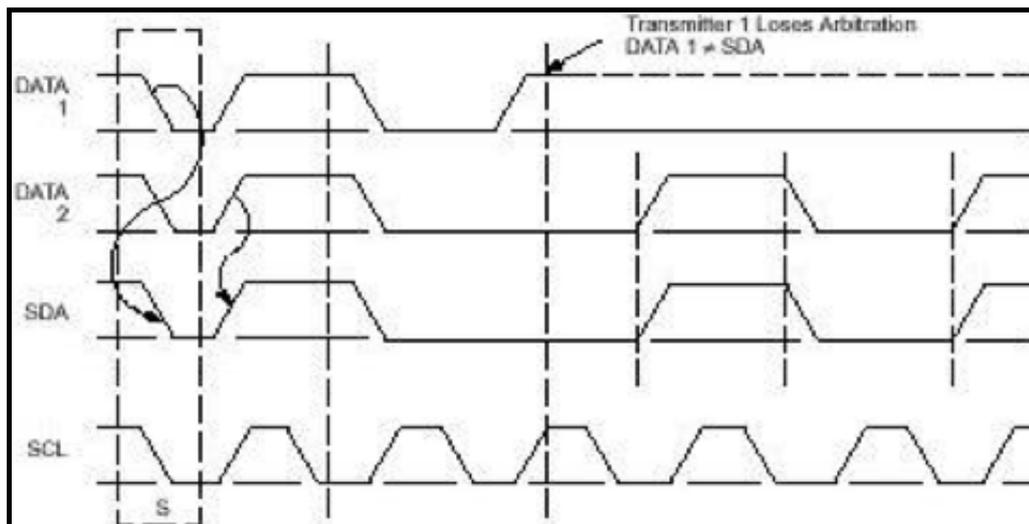


Figura 43. Procedimiento de arbitraje entre dos *Masters*

En esta figura se ve el arbitraje entre dos masters, aunque más pueden estar involucrados dependiendo de cuantos microcontroladores hay conectados al bus. En el momento en que hay una diferencia entre el nivel interno de la línea de datos del *master* DATA1 y el actual nivel de la línea de datos SDA, su salida de datos es interrumpida, lo cual significa que un nivel "1" está dominando en el Bus. Esto no afecta los datos transferidos inicialmente por el master ganador.

El arbitraje puede continuar varios bits hasta que se de la circunstancia de control del Bus por uno de los *Master*.

Tras el arbitraje los *Master* perdedores se deben poner inmediatamente en modo *Master-receptor* y esclavo pues los datos que envíe el *Master* dominante pueden ser para uno de ellos. Un *master* que pierde el arbitraje puede generar pulsos de reloj hasta el fin de *byte* en el cual el pierde el arbitraje. En el momento que un *master* toma el control solo este *master* toma las decisiones y genera los códigos de dirección, no existen *master* centrales, ni existen ordenes prioritarias en el Bus.

Especial atención debe ponerse si durante una transferencia de datos el procedimiento de arbitraje está todavía en proceso justo en el momento en el que se envía al Bus una condición de *Stop*. Es posible que esta situación pueda ocurrir, en este caso el *master* afectado debe mandar códigos de Inicio o *Stop*.

1.7.2.10 Formato

Los datos transferidos tienen la forma de la figura siguiente.

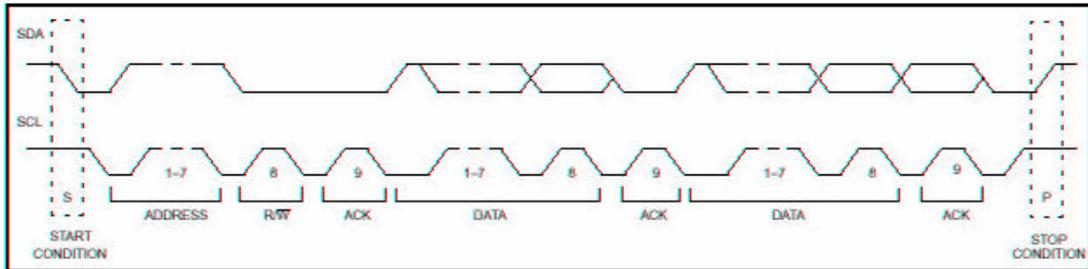


Figura 44. Transferencia completa de datos

Después de la condición de *Start* un código de dirección de un esclavo es enviada, esta dirección tiene 7 bits seguidos por un octavo código que corresponde a una dirección R/W (0-indica transmisión/1-indica solicitud de datos). Una transferencia de datos siempre acaba con una condición de *Stop* generado por el *master*, sin embargo si un *master* todavía desea comunicarse con el bus, puede generar repetidamente condiciones de *Start* y direccionar a otro esclavo sin generar primero la condición de *stop*.

Varias combinaciones de lectura y escritura son posibles dentro de una misma transferencia de datos.

Los posibles formatos de transferencia son:

Master transmite al esclavo-receptor. No cambia el bit de dirección.

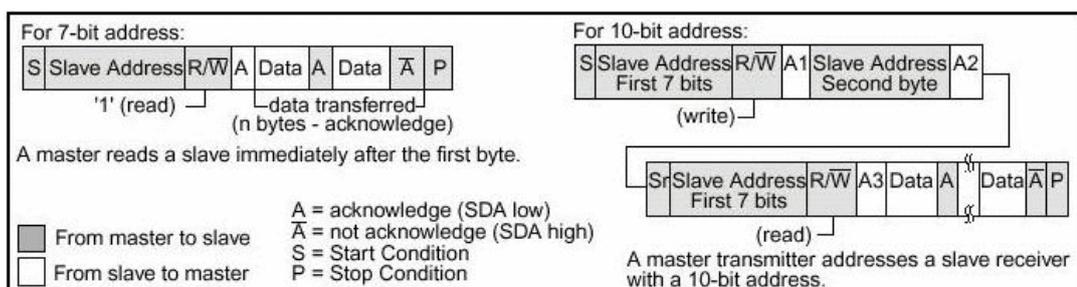


Figura 45: Secuencia de *master* como transmisor

Master lee a un esclavo inmediatamente después del primer *byte*.

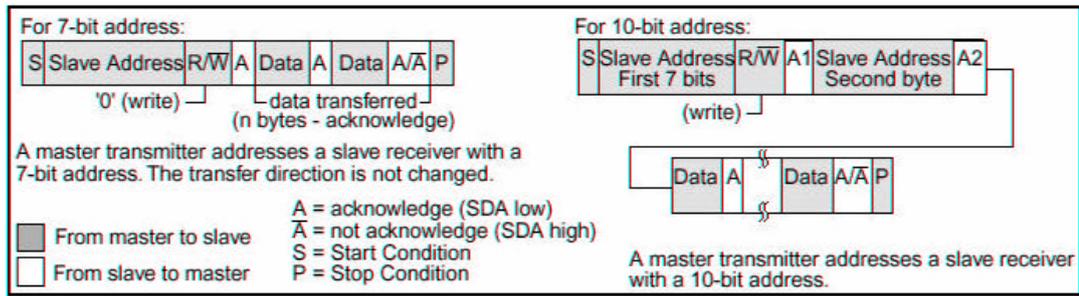


Figura 46: Secuencia de *master* como receptor

En el momento del primer reconocimiento el *master*-transmisor se convierte en un *master*-receptor y el esclavo-receptor en un esclavo-transmisor. El primer reconocimiento es aun generado por el esclavo. La condición de *stop* es generada por el *Master*, el cual a enviado previamente un no-reconocimiento.

Formato combinado

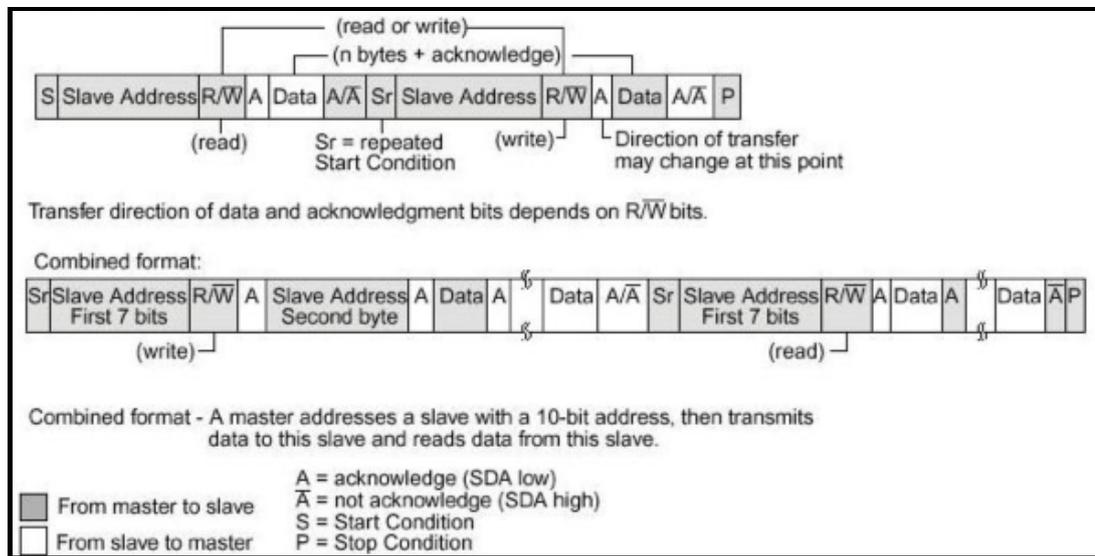


Figura 47. Esquema de formato combinado

Durante un cambio de dirección dentro de una transferencia, la condición de *Start* y la dirección del esclavo son ambos repetidos, pero con el bit R/W invertido. Si un *Master*-receptor envía una condición repetida de *Start*, el esclavo previamente ha enviado un no-reconocimiento.

Notas:

Se pueden combinar diversos formatos de direccionamiento

Las decisiones para el mayor o menor acceso a las posiciones de las memorias debe ser tomada por el diseñador del dispositivo.

Durante el primer *byte* de datos la posición de la memoria interna debe ser escrita.

Después de la condición *Start* la dirección del esclavo es repetida los datos pueden ser transferidos.

Cada *byte* es seguido por un bit de reconocimiento como indican los bloques en la secuencia.

Una condición de *Start* inmediatamente seguida por una condición de *Stop* es un formato ilegal.

Los dispositivos compatibles con el bus I2C deben poder reajustar su bus lógico a la recepción de una o mas condiciones de *Start*.

1.7.2.11 Direccionamiento

El procedimiento de dirección para el Bus I2C es tal que el primer *byte* después de la condición de *Start* usualmente determina que esclavo ha sido seleccionado por el *Master*.

La excepción se da en la "llamada general" (*byte* 0000 0000) con la que se direcciona a todos los dispositivos, cuando esta dirección es usada, todos los dispositivos en teoría deben responder con un reconocimiento (A), sin embargo algunos dispositivos pueden estar condicionados a ignorar esta dirección. El segundo *byte* de la "llamada general" define entonces la acción a tomar.

Hay dos formatos de dirección. El más simple es el formato de 7-bit con un bit R/W que permite direccionar hasta 128 dispositivos, que en la práctica se reduce a 112 debido a

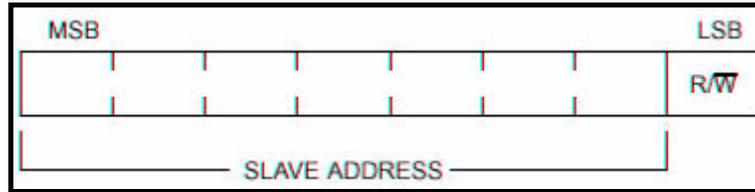


Figura 50. Primer *byte* luego del procedimiento de Inicio (*START*)

El octavo bit determina la dirección del mensaje, un 0 en la posición 8 del *byte* significa que el *Master* escribirá información en el esclavo seleccionado y un 1 en la posición 8 del *byte* significa que el *Master* leerá información del esclavo.

Cuando un *Master* envía una dirección después de la condición de Inicio cada dispositivo comprueba los siete primeros bits de la dirección con la suya propia. El que coincida se considera el dispositivo direccionado por el *Master* siendo un esclavo-receptor o esclavo emisor dependiendo del bit R/W.

Una dirección puede tener una parte fija y otra programable. Con lo que se pueden conectar dispositivos idénticos al sistema siendo activados por la parte fija y controlados por la parte programable. Existen una serie de direcciones reservadas en los Bus I2C que no se deben utilizar dado que son direcciones determinadas por el fabricante para usos generales. La combinación 11110xx de las direcciones esclavo se reservan para las direcciones de 10 bits. Dos grupos de 8 direcciones (0000xxx y 1111xxx) se reservan según la figura que se ve a continuación:

SLAVE ADDRESS	R/ bit	DESCRIPTION
0000 000	0	General call address
0000 000	1	START byte
0000 001	X	CBUS address
0000 010	X	Address reserved for different bus format
0000 011	X	Reserved for future purposes
0000 1XX	X	
1111 1XX	X	
1111 0XX	X	10-bit slave addressing

Figura 51. Definición de bits en el primer *byte*

Notas:

No se permite reconocer a ningún dispositivo en el *byte* de *Start*.

La dirección CBUS están reservadas para permitir la compatibilidad entre dispositivos I2C y CBUS en el mismo sistema. Los dispositivos compatibles del Bus I2C no están autorizados para responder a esta dirección.

Esta dirección está reservada para combinar el formato I2C con diferentes formatos de Bus y otros protocolos. Solo los dispositivos compatibles con estos protocolos y formatos que funcionan según I2C responden a esta dirección.

1.7.2.13 Especificaciones eléctricas y de tiempos

Dado la gran cantidad de diferentes dispositivos que se pueden conectar en el Bus I2C las tensiones dependen por un lado de las necesarias para cada uno de los componentes y de una cierta normativa bastante elástica para las líneas SDA y SCL. Se debe pretender que la alimentación de las líneas SCL y SDA debe ser a 5Vlt. manteniéndose las siguientes tolerancias:

Máxima tensión permitida a nivel bajo ("0") ---> 1,5V.

Mínima tensión permitida a nivel alto ("1") ---> 3 V.

A continuación se puede observar la tabla de tiempos.

PARAMETER	SYMBOL	STANDARD-MODE I ² C-BUS		FAST-MODE I ² C-BUS		UNIT
		Min.	Max.	Min.	Max.	
SCL clock frequency	f_{SCL}	0	100	0	400	kHz
Bus free time between a STOP and START condition	t_{BUF}	4.7	–	1.3	–	μ s
Hold time (repeated) START condition. After this period, the first clock pulse is generated	$t_{HD,STA}$	4.0	–	0.6	–	μ s
LOW period of the SCL clock	t_{LOW}	4.7	–	1.3	–	μ s
HIGH period of the SCL clock	t_{HIGH}	4.0	–	0.6	–	μ s
Set-up time for a repeated START condition	$t_{SU,STA}$	4.7	–	0.6	–	μ s
Data hold time: for CBUS compatible masters (see NOTE, Section 9.1.3) for I ² C-bus devices	$t_{HD,DAT}$	5.0 0 ¹⁾	– –	– 0 ¹⁾	– 0.9 ²⁾	μ s μ s
Data set-up time	$t_{SU,DAT}$	250	–	100 ³⁾	–	ns
Rise time of both SDA and SCL signals	t_r	–	1000	$20 + 0.1C_b$ ⁴⁾	300	ns
Fall time of both SDA and SCL signals	t_f	–	300	$20 + 0.1C_b$ ⁴⁾	300	ns
Set-up time for STOP condition	$t_{SU,STO}$	4.0	–	0.6	–	μ s
Capacitive load for each bus line	C_b	–	400	–	400	pF

Figura 52. Características de las líneas SDA y SCL para dispositivos con bus I2C

1.7.3 Módulo de Puerto Serial Sincrónico (SSP)

1.7.3.1 Resumen del módulo SSP

El módulo de Puerto Serial Sincrónico (SSP) es una interfase serial muy usada para la comunicación con otros dispositivos periféricos microcontroladores. Estos dispositivos periféricos pueden ser memorias Seriales EEPROM, registros de desplazamiento, *display drivers*, convertidores A/D, etc.

El módulo SSP puede operar en uno de dos modos:

- Interfase Periférica Serial (SPI)
- Inter-Circuito Integrado (I2C)

El módulo SSP en modo I2C está habilitado en los dispositivos desde el PIC16C6X hacia adelante.

1.7.3.2 I2C Resumen

A continuación se presentará un resumen con lo más relevante del bus *Inter-Integrated Circuit* (I2C), se discutirá principalmente la operación del módulo SSP en modo I2C.

Todas las partes del protocolo esclavo se implementan en el *hardware* del módulo SSP, excepto el soporte de la llamada general, mientras las partes del protocolo maestro se direccionarán en el *software* del PIC16FXX.

Existen dos registros de control en el módulo SSP, estos son:

Register SSPSTAT: Synchronous Serial Port Status Register							
R/W-0	R/W-0	R-0	R-0	R-0	R-0	R-0	R-0
SMP	CKE	D/ \bar{A}	P	S	R/ \bar{W}	UA	BF
bit 7						bit 0	

En donde:

Bit 7 SMP: Fase de muestra de entrada de dato SPI

Bit 6 CKE: SPI Selección del flanco del reloj

Bit 5 D/A: Bit de Dato/*Address* (solamente modo I2C)

1 = Indica que el último *byte* recibido o transmitido fue un dato

0 = Indica que el último *byte* recibido o transmitido fue una dirección

Bit 4 P: Bit de *Stop* (Solamente en modo I2C. Este bit es borrado cuando el módulo SSP es deshabilitado)

1 = Indica que un bit de *stop* fue lo último detectado (este bit es '0' sobre el *RESET*)

0 = Indica que un bit de *stop* no fue lo último detectado

Bit 3 S: Bit de *Start* (Solamente en modo I2C. Este bit es borrado cuando el módulo SSP es deshabilitado)

1 = Indica que un bit de *start* fue lo último detectado (este bit es '0' sobre el *RESET*)

0 = Indica que un bit de *start* no fue lo último detectado

Bit 2 R/W: Bit de información *Read/Write* (Solamente en modo I2C)

Estos bits mantienen la información R/W siguiendo la última dirección completada. Este bit es solamente válido desde la dirección completa hasta el próximo bit de *start*, bit *stop*, or bit *not ACK*.

1 = Leer

0 = Escribir

Bit 1 UA: *Update Address* (10-bit Solamente en modo I2C)

1 = Indica que el usuario necesita actualizar la dirección en el registro SSPADD

0 = La dirección no necesita ser actualizada

Bit 0 BF: Bit *Buffer Full Status*

Recepción (modos SPI y I2C)

1 = Recepción completa, SSPBUF está *full*

0 = Recepción no completa, SSPBUF está vacío

Transmisión (Solamente en modo I2C)

1 = Transmisión en progreso, SSPBUF está *full*

0 = Transmisión completa, SSPBUF está vacío

SSPCON: Synchronous Serial Port Control Register							
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0
bit 7							bit 0

En donde:

Bit 7 WCOL: Bit de detección de Colisión de Escritura

1 = El registro SSPBUF es escrito mientras aún está transmitiendo la palabra previa (debe ser borrado en *software*)

0 = Sin colisión

Bit 6 SSPOV: Bit indicador de *Overflow* en la recepción

En modo I2C:

1 = Un *byte* es recibido mientras el registro SSPBUF espera el *byte* previo. SSPOV es un "sin importancia" en modo transmisión. SSPOV debe ser borrado en *software* en otro modo.

0 = No *overflow*

Bit 5 SSPEN: Bit de habilitación de SSP

En ambos modos, cuando se habilita, este pin debe ser apropiadamente configurado como entrada o salida.

En modo I2C:

1 = Se habilita el puerto serial y se configura SDA y SCL como la fuente de los pines del puerto serial

0 = Se deshabilita los puertos y configuraciones de los pines como pines de puerto I/O

Bit 4 CKP: Bit de Selección de polaridad del *Clock*

En modo I2C: SCK *release* control

1 = Habilitación del *clock*

0 = Mantiene el *clock* bajo (estiramiento del *clock*) (usado para asegurar un tiempo de seteo de los datos)

Bits 3, 2, 1 y 0 SSPM3-SSPM0: Bit de selección de modo SSP

0110 = I2C *slave mode*, 7-bit *address*

0111 = I2C *slave mode*, 10-bit *address*

1000 = *Reserved*

1001 = *Reserved*

1010 = *Reserved*

1011 = I2C modo maestro controlado por *firmware* (*slave* inactivo)

1100 = *Reserved*

1101 = *Reserved*

1110 = I2C *slave mode*, 7-bit *address with start and stop bit interrupts enabled*

1111 = I2C *slave mode*, 10-bit *address with start and stop bit interrupts enabled*

1.7.3.3 Operación SSP en modo I2C

El módulo SSP en modo I2C implementa todas las funciones de esclavo, excepto el soporte para llamada general, y provee interrupciones en hardware sobre los bits de *start* y *stop* para facilitar la implementación *firmware* de las funciones maestras. El módulo SSP implementa el modo estándar de especificación de dirección esto es 7-bit y 10-bit. Dos pines son usados para la transferencia de datos. Estos son el pin SCL RC3/SCK/ y el pin SDA RC4/SDI/. El usuario debe configurar estos pines como entradas o salidas a través de los bits TRISC<4:3>. Las funciones del módulo SSP son habilitadas seteando el bit SSPEN (SSPCON<5>).

El módulo SSP tiene 5 registros para operación I2C. Ver Figura 53. Estos son:

- Registro de Control SSP (SSPCON)
- Registro de Status SSP (SSPSTAT)
- Buffer Serial Receptor/Transmisor (SSPBUF)
- Registro de Desplazamiento SSP (SSPSR) - No directamente accesible
- Registro de Dirección SSP (SSPADD)

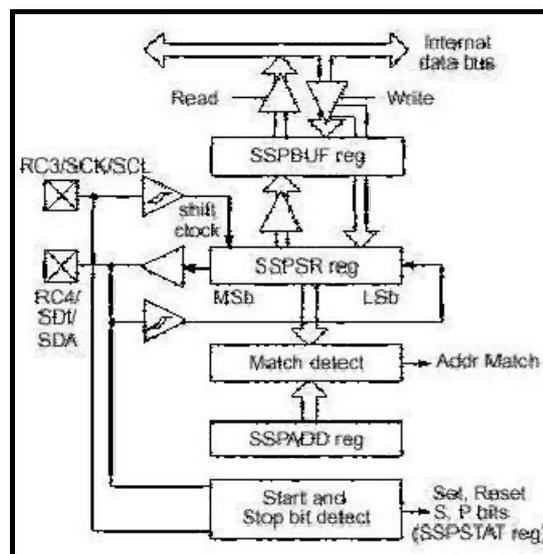


Figura 53. Diagrama de Bloques SSP (Modo I2C)

El registro SSPCON permite el control de la operación I2C. Cuatro bits (SSPCON<3:0>) permiten uno de los siguientes modos I2C a ser seleccionados:

- Modo Esclavo I2C (7-bit *address*)
- Modo Esclavo I2C (10-bit *address*)
- Modo esclavo I2C (7-bit *address*), con bit de interrupción de de *start* y *stop* habilitado.
- Modo Esclavo I2C (10-bit *address*), con bit de interrupción de de *start* y *stop* habilitado.
- Modo Maestro Controlado por *Firmware I2C*, esclavo está inactivo.

La selección de cualquier modo I2C, con el bit SSPEN elegido, fuerza a los pines SCL y SDA a ser de drenaje abierto, de esta forma los pines son programados como entradas seteando los bits TRISC apropiados. El registro SSPSTAT entrega el estatus de la transferencia de datos. Esta información incluye la detección de un bit de *stop* o *start*, especificando si el *byte* recibido fue dato o dirección si el próximo *byte* es la terminación de la dirección de 10-bit, y si este sería una transferencia de datos de lectura o escritura. El registro SSPSTAT es solamente de lectura.

El SSPBUF es el registro para el cual la transferencia de datos es escrita hacia o leída desde. El registro SSPSR desplaza el dato dentro o fuera del dispositivo. En operación de recepción, el SSPBUF y SSPSR crean un receptor "*doubled buffered*". Cuando llegan dos bytes uno se recibe mientras el anterior es leído. Cuando el *byte* completo es recibido, se transfiere hacia el registro SSPBUF y el *flag* bit SSPIF es elegido. Si otro *byte* completo es recibido antes de que el registro SSPBUF sea leído, se recibe un *overflow* y el bit SSPOV (SSPCON<6>) es elegido y el *byte* en SSPSR se pierde.

El registro SSPADD mantiene la dirección del esclavo. En modo 10-bit, el usuario necesita primero escribir el *byte* alto de la dirección (1111 0 A9 A8 0). En seguida se carga el *byte* bajo (A7:A0) que completa la dirección.

1.7.3.4 Modo Esclavo

En modo esclavo, los pines SCL y SDA deben ser configurados como entradas (TRISC<4:3> elegidos). El módulo SSP sustituye el estado de entrada con el dato de salida cuando sea requerido (esclavo-transmisor). Cuando la transferencia de una dirección se completa, el hardware automáticamente genera el pulso *acknowledge* (ACK), y entonces el registro SSPBUF es cargado con el valor que recibe el registro SSPSR.

Hay condiciones en que el módulo SSP no entrega este pulso ACK. Estas son , (una o ambas):

- El bit *buffer full* BF (SSPSTAT<0>) fue elegido antes de que la transferencia fuera recibida.
- El bit *overflow* SSPOV (SSPCON<6>) fue elegido antes de que la transferencia fuera recibida.

En este caso, el valor del registro SSPSR no está cargado en el SSPBUF, pero el bit SSPIF (PIR1<3>) es elegido. La tabla siguiente muestra qué sucede cuando un *byte* en la transferencia de datos es recibido, dado el estatus de los bits BF y SSPOV. Las celdas sombreadas muestran donde el uso no apropiado del *software* borra la condición de *overflow*. El bit *Flag* BF es borrado por lectura del registro SSPBUF mientras el bit SSPOV es borrado a través de *software*.

DATA TRANSFER RECEIVED BYTE ACTIONS				
Status Bits as Data Transfer is Received		SSPSR → SSPBUF	Generate $\overline{\text{ACK}}$ Pulse	Set bit SSPIF (SSP Interrupt occurs if enabled)
BF	SSPOV			
0	0	Yes	Yes	Yes
1	0	No	No	Yes
1	1	No	No	Yes
0	1	No	No	Yes

Al momento de transferir datos el esclavo tendrá que poner a cero el bit SSPCON<4> que habilita la dilatación del reloj cuando este es un receptor.

1.7.3.5 Direccionamiento

Una vez que el módulo SSP ha sido habilitado, se espera que ocurra una condición de *start*. Siguiendo la condición de *start*, los 8 bits son desplazados hacia el registro SSPSR. Todos los bits entrantes son muestreados con el flanco de subida de la línea del reloj (SCL). El valor del registro SSPSR<7:1> es comparado con el valor del registro SSPADD. La dirección es comparada sobre el flanco de bajada del octavo pulso (SCL).

Si la dirección se completa y los bits BF y SSPOV van a "0", ocurren los siguientes eventos:

- El valor del registro SSPSR es cargado al registro SSPBUF.
- El bit *buffer full*, BF es elegido.
- Se genera un pulso ACK.
- El bit de *flag* de interrupción SSP, SSPIF (PIR1<3>) es elegido (la interrupción es generada si está habilitada) - sobre el flanco de bajada del noveno pulso SCL.

En el modo de dirección 10-bit, dos *bytes* de direcciones se necesitan para ser recibido por el esclavo. El quinto bit más significativo (MSbs) del primer *byte* de dirección, especifica si esta es una dirección de 10-bit. El bit R/W (SSPSTAT<2>) debe especificar una escritura tal, que el dispositivo esclavo reciba el segundo *byte*. Para una dirección de 10-bit el primer *byte* será igual a '1111 0 A9 A8 0', donde A9 y A8 son los MSbs de la dirección. La secuencia de eventos para la dirección de 10-bit es como sigue, con los pasos 7- 9 para un esclavo-transmisor:

1. Recibe el *byte* alto de la dirección (bits SSPIF, BF, y bit UA (SSPSTAT<1>) son elegidos).
2. Se actualiza el registro SSPADD con el segundo *byte* de dirección (limpia el bit UA y libera la línea SCL).
3. Lee el registro SSPBUF con lo cual los bits BF y SSPIF se limpian.
4. Recibe el segundo *byte* de dirección (los bits SSPIF, BF, y UA son elegidos).

5. Se actualiza el registro SSPADD con el *byte* alto de la dirección, esto libera la línea SCL y limpia el bit UA.
6. Lee el registro SSPBUF con lo cual los bits BF y SSPIF se limpian.
7. Se recibe la condición de *Start* repetida.
8. Se recibe el primer *byte* de dirección (los bits SSPIF y BF son elegidos).
9. Se lee el registro SSPBUF (se limpia el bit BF) y se limpia el bit *flag* SSPIF.

1.7.3.6 Recepción (7-bits)

Cuando el bit R/W del *byte* de dirección está en cero y se completa la dirección, el bit R/W del registro SSPSTAT será borrado. La dirección recibida es cargada en el registro SSPBUF. Cuando exista la condición de *overflow* en el *byte* de dirección, entonces no se entregará el pulso *acknowledge* (ACK). Una condición de *overflow* será definida según si el bit BF (SSPSTAT<0>) es elegido o el bit SSPOV (SSPCON<6>) es elegido. Una interrupción SSP es generada para cada *byte* transferido. El bit *Flag* SSPIF (PIR1<3>) debe ser borrado en *software*.

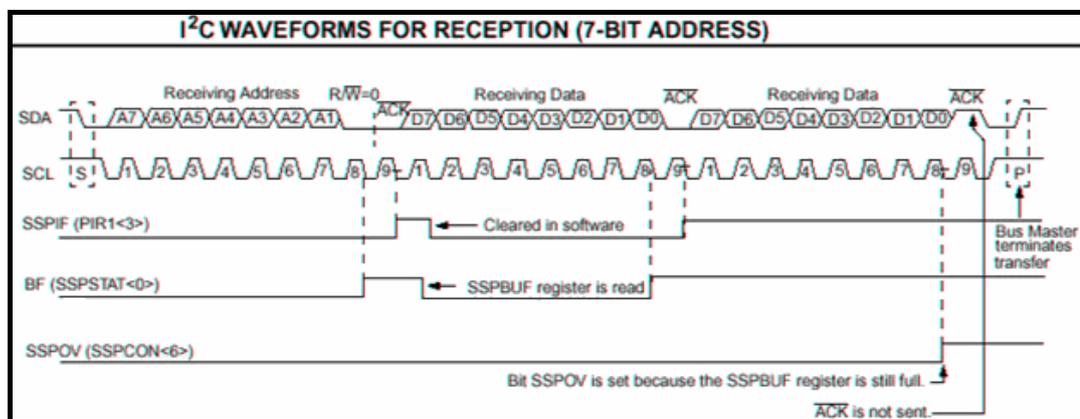


Figura 54. Diagrama de recepción de 7 bits

1.7.3.7 Transmisión (7-bits)

Cuando el bit R/W del *byte* de dirección entrante es elegido y la dirección se completa, el bit R/W del registro SSPSTAT es elegido. La dirección recibida es cargada en el registro SSPBUF.

El pulso ACK será enviado sobre el noveno bit, y el pin RC3/SCK/SCL se mantiene bajo. El dato transmitido debe ser cargado en el registro SSPBUF, el cual también carga al registro SSPSR. El pin RC3/SCK/SCL podría ser habilitado seteando el bit CKP (SSPCON<4>). El maestro debe monitorear el pin SCL antes de direccionar otro pulso de reloj. El dispositivo esclavo debe mantenerse desconectado del maestro para dilatar al reloj. El octavo bit de dato cae fuera de la entrada SCL. Esto asegura que la señal SDA es válida durante el tiempo alto SCL.

Una interrupción SSP es generada para cada *byte* de dato transferido. El bit *Flag* SSPIF debe ser borrado en *software*, y el registro SSPSTAT es usado para determinar el estatus del *byte*. El bit Flag SSPIF es elegido sobre el flanco de caída del noveno pulso del *clock*.

Como un esclavo-transmisor, se envía el pulso ACK desde el maestro receptor sobre el flanco de subida del noveno pulso de entrada SCL. Si la línea SDA fue alta (*not* ACK), entonces la transferencia de datos es completa. Cuando el ACK es enviado por el esclavo, el esclavo lógico es reseteado (*reset* del registro SSPSTAT) y monitorea la línea esperando otra ocurrencia del bit *start*. Si la línea SDA está baja (ACK), el dato transmitido debe ser cargado en el registro SSPBUF, el cual también carga al registro SSPSR. Entonces el pin RC3/SCK/SCL será habilitado por seteo del bit CKP.

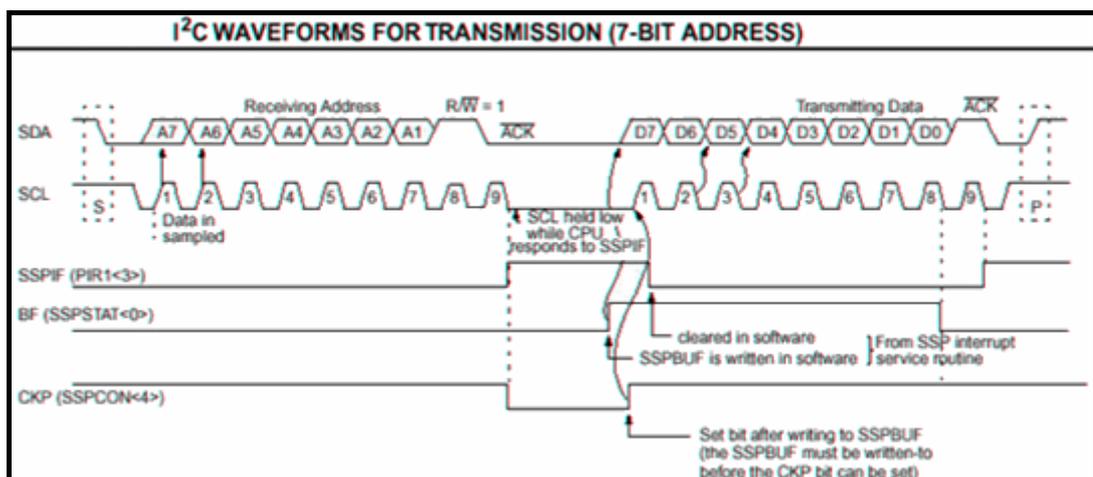


Figura 55. Diagrama de transmisión de 7 bits

1.7.3.8 Modo maestro (*Firmware*)

El modo maestro de operación es soportado en *firmware* usando generación de interrupción sobre la detección de las condiciones de *Start* y *Stop*. Los bits *Stop* (P) y *Start* (S) son puestos a cero con un *reset* o cuando el módulo SSP está deshabilitado. El control del bus I2C debe ser tomado cuando el bit P es elegido, o el bus está inactivo y los bits S y P son borrados.

En modo maestro el SCL y la línea SDA son manipuladas limpiando los correspondientes bits TRISC<4:3>. El nivel de salida es siempre bajo, independiente de los valores en PORTC<4:3>. De modo que cuando se transmitan datos, un '1' debe ser elegido en TRISC<4> (entrada) y un '0' en TRISC<4> borrado (salida). El mismo escenario es cierto para la línea SCL con el bit TRISC<3>.

Los siguientes eventos causarán una Interrupción en el bit *Flag* SSPIF, al ser elegido (si está habilitada la interrupción SSP):

- Condición de *START*
- Condición de *STOP*
- *Byte* de transferencia de datos transmitido/recibido

El modo maestro de operación puede ser hecho con el modo esclavo inactivo (SSPM3:SSPM0 = 1011) o con el esclavo activo (SSPM3:SSPM0 = 1110 o 1111). Cuando el modo maestro y esclavo son habilitados, se necesita el *software* para diferenciar las fuentes de interrupción.

1.7.3.9 Modo Multi-master (*Firmware*)

En modo multi-*master*, la generación de interrupción sobre la detección de las condiciones *START* y *STOP* permiten la determinación de cuando el bus está libre. Los bits *STOP* (P) y *START* (S) son borrados desde un *reset* o cuando el módulo SSP está

deshabilitado. El control del bus I2C puede ser tomado cuando el bit P (SSPSTAT<4>) es elegido, o el bus está inactivo y los bits S y P están borrados. Cuando el bus está ocupado, se habilita la interrupción SSP que generará la interrupción cuando ocurra la condición *STOP*.

En operación multi-*master*, la línea SDA debe ser monitoreada para ver si el nivel de señal es el nivel esperado en la salida. Este chequeo necesita ser hecho solamente cuando un nivel alto es salida. Si un nivel alto se espera y un nivel bajo se presenta, los dispositivos necesitan liberar las líneas SDA y SCL (se elige TRISC<4:3>). Hay dos estados cuando este arbitraje puede ser perdido, estas son:

- Transferencia de Dirección
- Transferencia de Dato

Cuando el esclavo lógico está habilitado, el esclavo continúa recibiendo. Si el arbitraje fue perdido durante la transferencia de la dirección, la comunicación hacia el dispositivo puede estar en progreso. Si un esclavo fue direccionado se genera el pulso ACK. Si el arbitraje fue perdido durante la transferencia de datos, el dispositivo necesita retransferir el dato un tiempo después.

REGISTERS ASSOCIATED WITH I ² C OPERATION											
Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	Value on all other resets
0Bh, 8Bh, 10Bh, 18Bh	INTCON	GIE	PEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF	0000 000x	0000 000u
0Ch	PIR1	PSPIF ⁽¹⁾	⁽²⁾	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
8Ch	PIE1	PSPIE ⁽¹⁾	⁽²⁾	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
13h	SSPBUF	Synchronous Serial Port Receive Buffer/Transmit Register								xxxx xxxx	uuuu uuuu
93h	SSPADD	Synchronous Serial Port (I ² C mode) Address Register								0000 0000	0000 0000
14h	SSPCON	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	0000 0000	0000 0000
94h	SSPSTAT	SMP ⁽³⁾	CKE ⁽³⁾	D/Ā	P	S	R/Ī	UA	BF	0000 0000	0000 0000
87h	TRISC	PORTC Data Direction register								1111 1111	1111 1111

Legend: x = unknown, u = unchanged, - = unimplemented locations read as '0'.
 Shaded cells are not used by SSP module in SPI mode.
 Note 1: PSPIF and PSPIE are reserved on the PIC16C66, always maintain these bits clear.
 2: PIR1<6> and PIE1<6> are reserved, always maintain these bits clear.
 3: The SMP and CKE bits are implemented on the PIC16C66/67 only. All other PIC16C6X devices have these two bits unimplemented, read as '0'.

Figura 56: Bits y direcciones respectivas asociados con el modo I2C

CAPITULO II

CARACTERÍSTICAS DEL EQUIPO

2.1 Características funcionales

Dentro de las principales características funcionales del equipo podemos anotar las siguientes:

Fácil interpretación de los datos, se debe a que directamente se los podrá visualizar en forma digital a través del display LCD , a más de que los cálculos matemáticos del rendimiento (unidades producidas/hora) están siendo realizados internamente por el PIC mediante el software programado.

Interfaz amigable, ya que el teclado nos permite navegar por menús desplegables haciendo que la elección de la opción deseada sea más fácil y rápida, sin necesidad de digitar funciones especiales o que causen confusión al usuario.

Multifuncional, debido a que este equipo se lo puede instalar en un único lugar de la planta en donde converjan los cables que traen señales de todas las máquinas a través de sus PIC esclavos, por lo tanto no se necesita un equipo por cada máquina, ya que este mismo equipo a través de su menú, nos posibilita la elección de la máquina (esclavo) de la cual obtenemos los datos deseados.

2.2 Características técnicas

Para lograr este diseño y construcción, nos hemos basado en una aplicación con microcontroladores, y hemos elegido al PIC 16F877 como Maestro y al PIC 16F872 como esclavo, además de que para obtener un producto final de buena calidad y garantizado, se han tomado en cuenta varias características técnicas que son clave para su correcta operación, como por ejemplo:

El funcionamiento continuo (24 horas) del supervisor, y la aplicación industrial que se le va a dar exige que la fuente de alimentación cumpla con varios requisitos para poder soportar las horas de funcionamiento, y la variabilidad en las tensiones de alimentación debido a las entradas y salidas de cargas abruptas en la planta industrial, por lo tanto se ha diseñado, construido y probado dicha fuente para que siempre entregue una tensión y corriente reguladas y estabilizadas.

El almacenamiento y manejo de datos es igualmente de tratamiento especial por las mismas razones antes indicadas, por lo que, los datos obtenidos (conteo y totalización de unidades producidas) de las distintas máquinas, se guardan en la memoria EEPROM del PIC de cada máquina y cuyo protocolo de comunicación I2C es perfectamente manejable por el PIC MASTER. Anteriormente ya estudiamos al PIC con sus funciones, entre ellas el protocolo I2C y el módulo SSP.

El control de los tiempos reales de operación de las máquinas, se lo efectúa con la ayuda de un circuito integrado DS1307, ya que este posee su propio generador de tiempo real y que lo estudiaremos con mayor detalle más adelante, este al comunicarse con el PIC mediante protocolo I2C, nos entrega los siguientes datos: año, mes, día, hora, minuto y segundo, con los que se procede a calcular los tiempos operativos de cada máquina, los tiempos totales acumulados y el rendimiento.

El acceso externo hacia los datos almacenados en el supervisor y los parámetros a ser controlados como nombre de la máquina, tiempo de operación, unidades producidas y rendimiento, es posible gracias a que se ha utilizado como interfaz HMI (*Human Machine Interface*), un teclado matricial de 16 teclas y un display LCD de 16 caracteres por 4 líneas, mismos que nos permitirán visualizar con claridad estos valores para su lectura y control. Estos dos dispositivos los pudimos estudiar anteriormente en el capítulo de los PIC's bajo el título de "Conexión de Periféricos".

El cerebro principal, como ya lo habíamos acotado y estudiado anteriormente, nuestro equipo supervisor tendrá como cerebro master al micro controlador PIC 16F877 ya que este tipo de PIC es rico en recursos internos como por ejemplo puertos serie, protocolo de comunicación I2C, módulo SSP, convertidor análogo / digital, a más de que disponen de posibilidades más amplias de interrupciones y manejan el mismo juego de instrucciones. Este PIC es el encargado de realizar el barrido del teclado, del manejo del

chip DS1307 y de la lectura de los demás PIC's esclavos (1 por cada máquina a supervisar) activando el modo SSP de I2C y de presentar los resultados medidos y calculados en el LCD.

2.3 Fuentes de Poder Reguladas (Estabilizadas)

2.3.1 Introducción

La función de una fuente de alimentación es convertir la tensión alterna en una tensión continua y lo mas estable posible, en esta ocasión se revisarán los puntos más importantes a tener en cuenta para construir una fuente de alimentación estabilizada, con unas características adecuadas para alimentar un circuito electrónico con especificaciones digitales, para ello se usan los siguientes componentes: 1.- Transformador de entrada; 2.- Rectificador a diodos; 3.- Filtro para el rizado; 4.- Regulador (o estabilizador) lineal, este último es imprescindible para garantizar una correcta operación cuando se trata de aplicaciones digitales.

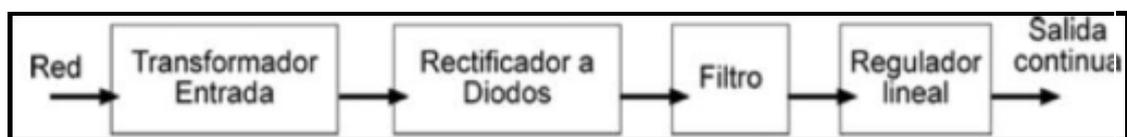


Figura 57. Componentes de una fuente de alimentación

La tensión continua disponible a la salida del filtro de un rectificador puede que no sea lo suficientemente buena, debida al rizado, o que varíe su valor ante determinado tipo de perturbaciones, como variaciones de la tensión de entrada, de la carga o de la temperatura.

En estos casos se necesitan circuitos de regulación o estabilización para conseguir que la tensión continua a utilizar sea lo más constante posible. Lo ideal sería que la tensión de salida fuera constante para cualquier condición del circuito pero esto es imposible debido a:

- La tensión de red puede tener variaciones de hasta el 20% de su valor nominal que sería uno de los parámetros más importantes a tomar en cuenta en nuestro caso, debido a la entrada y salida de cargas a nivel industrial.
- El circuito de carga conectado al rectificador puede absorber más o menos corriente. Al aumentar la corriente por la carga, la tensión de salida disminuirá debido a la caída en la resistencia del transformador, de los diodos, los semiconductores, etc.
- En la salida aparece un rizado.
- Cuando se utilizan dispositivos semiconductores, la tensión de salida varía con la temperatura.

Más adelante estudiaremos entonces la forma de conseguir que la tensión sea constante (regulada y estable) por medio de reguladores o estabilizadores lineales, ahora pondremos nuestra atención en los primeros componentes que conforman una fuente:

2.3.2 Componentes Básicos

2.3.2.1 Transformador de entrada

El transformador de entrada reduce la tensión de red (generalmente 220 o 120 V) a otra tensión más adecuada para ser tratada. Solo es capaz de trabajar con corrientes alternas esto quiere decir que la tensión de entrada será alterna y la de salida también. Consta de dos arroyamientos sobre un mismo núcleo de hierro, ambos arroyamientos, primario y secundario, son completamente independientes y la energía eléctrica se transmite del primario al secundario en forma de energía magnética a través del núcleo. El esquema de un transformador simplificado se lo puede observar en la figura 58.

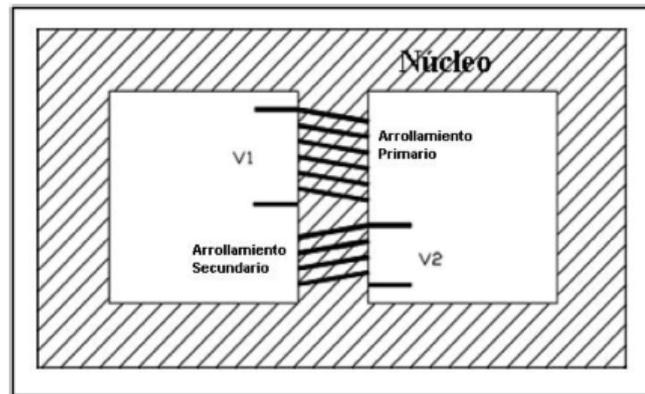


Figura 58. Esquema básico de un transformador

Hay dos tipos de transformador los de armadura F o E-I y los toroidales O (figura 59), estos últimos tienen un mejor rendimiento, no obstante no es determinante, es importante que los devanados estén separados físicamente y deben ser de hilo de cobre, no de aluminio.



(Disponible es Web: <http://www.hispavila.com/3ds/elimages/transfo1.gif>,
<http://www.hispavila.com/3ds/elimages/toroide1.gif>)

Figura 59. Transformador de armadura y Toroidal "O"

2.3.2.2 Rectificador

El rectificador es el que se encarga de convertir la tensión alterna que sale del transformador en tensión continua. Para ello se utilizan diodos. El rectificador se conecta después del transformador, por lo tanto le entra tensión alterna y tendrá que sacar tensión continua, es decir, un polo positivo y otro negativo (Figura 60).



Figura 60. Conexión de un rectificador

Rectificador en puente

El rectificador más usado es el llamado rectificador en puente, su esquema es el siguiente:

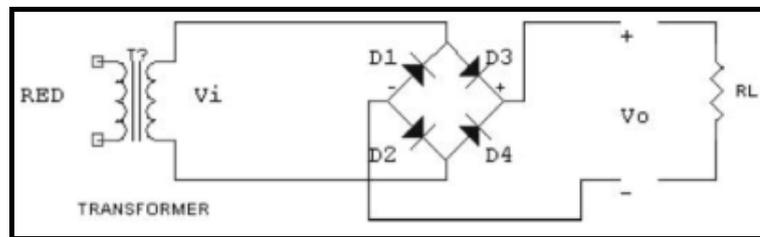


Figura 61. Rectificador en puente de Graetz

Luego de conectar este dispositivo a la salida del transformador, todavía no se ha conseguido una tensión de salida demasiado estable, por ello, será necesario filtrarla después. Es tan común usar este tipo de rectificadores que se venden ya preparados los cuatro diodos en un solo componente. Suele ser recomendable usar estos puentes rectificadores, ocupan menos que poner los cuatro diodos y para corrientes grandes vienen ya preparados para ser montados en un radiador. Este es el aspecto de la mayoría de ellos:

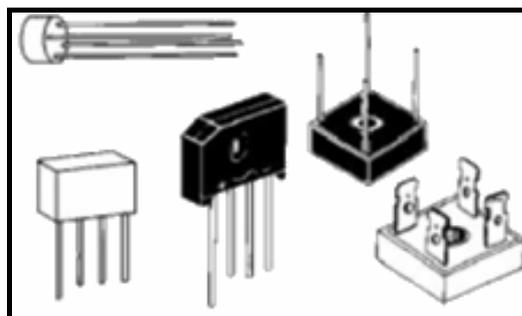


Figura 62. Aspecto físico de varios modelos de Puentes Rectificadores encapsulados

Tienen cuatro terminales, dos para la entrada en alterna del transformador, uno la salida positiva y otro la negativa o masa. Las marcas en el encapsulado suelen ser:

- ~ Para las entradas en alterna
- + Para la salida positiva
- Para la salida negativa o masa.

2.3.2.3 Filtro

La tensión en la carga que se obtiene de un rectificador es en forma de pulsos. En un ciclo de salida completo, la tensión en la carga aumenta de cero a un valor de pico, para caer después de nuevo a cero. Esta no es la clase de tensión continua que precisan la mayor parte de circuitos electrónicos. Lo que se necesita es una tensión constante, similar a la que produce una batería. Para obtener este tipo de tensión rectificadas en la carga es necesario emplear un filtro.

El tipo mas común de filtro es el del condensador a la entrada, en la mayoría de los

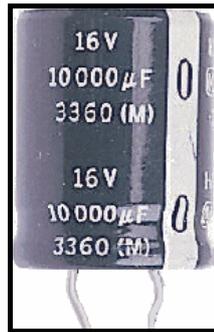
casos perfectamente válido. Sin embargo en algunos casos puede no ser suficiente y tendremos que echar mano de algunos componentes adicionales.

Filtro con condensador a la entrada

Este es el filtro mas común y seguro que se lo conoce, basta con añadir un condensador (Fig. 63) en paralelo con la carga (R_L), de esta forma:



Figura 63. Filtro en conexión básica



(Disponible en web: <http://www.hispavila.com/3ds/elimages/electrolitic.gif>)

Figura 64. Condensador

2.3.2.4 Regulador

Un regulador o estabilizador es un circuito que se encarga de reducir el rizado y de proporcionar una tensión de salida de la tensión exacta que queramos. En esta sección nos centraremos en los reguladores integrados de tres terminales que son los más sencillos y baratos que hay, en la mayoría de los casos son la mejor opción.

Este es el esquema de una fuente de alimentación regulada con uno de estos reguladores:

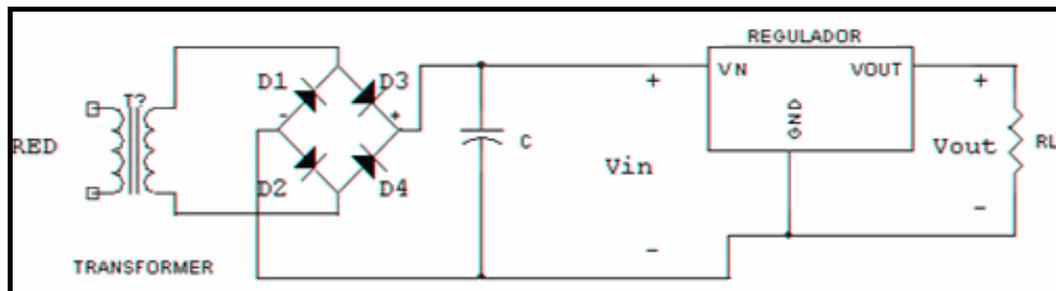


Figura 65. Fuente de alimentación regulada

Es muy corriente encontrarse con reguladores que reducen el rizado en 10000 veces (80 dB), esto significa que si usas la regla del 10% el rizado de salida será del 0.001%, es decir, inapreciable.

Las ideas básicas de funcionamiento de un regulador de este tipo son:

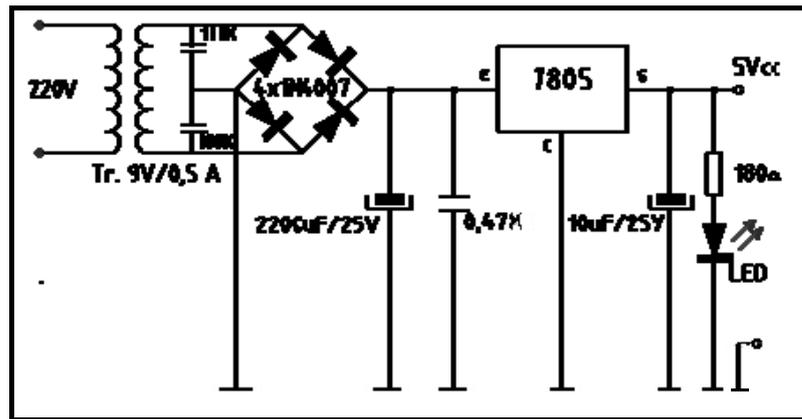
- La tensión entre los terminales V_{out} y GND es de un valor fijo, no variable, que dependerá del modelo de regulador que se utilice.
- La corriente que entra o sale por el terminal GND es prácticamente nula y no se tiene en cuenta para analizar el circuito de forma aproximada. Funciona simplemente como referencia para el regulador.
- La tensión de entrada V_{in} deberá ser siempre unos 2 o 3 V superior a la de V_{out} para asegurarnos el correcto funcionamiento.

El diseño de fuentes de alimentación estabilizadas mediante reguladores integrados monolíticos (reguladores fijos), resulta sumamente fácil. Concretamente para 1A (amperio) de salida, en el comercio con encapsulado TO-220, se dispone de los más populares en las siguientes tensiones estándar de salida:

Tipo 1A positivo	Tensión/Salida
UA7805	5
UA7806	6
UA7808	8
UA7809	9
UA7812	12
UA7915	15
UA7818	18
UA7824	24
UA7830	30
UA78XX	Versión negativo =

Figura 66. Cuadro de tensiones estándar de salida según el tipo de dispositivo

Todos estos reguladores tienen en común que son fijos y que proporcionan adecuadamente refrigerados una corriente máxima, de 1 A. Vemos a continuación en la Fig. 67, un ejemplo del esquema básico de una fuente de alimentación de 5 V y 500 mA.



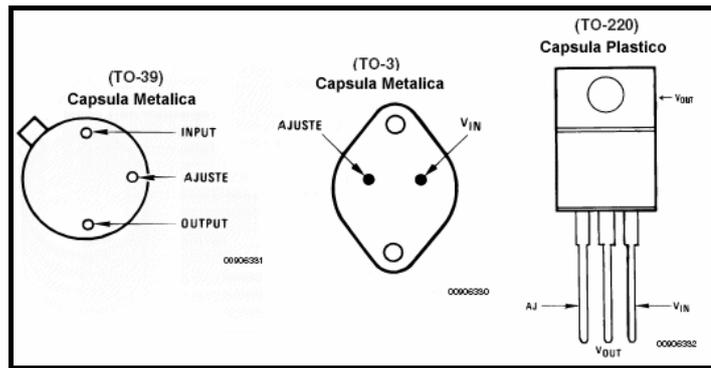
(Disponible en Web: <http://www.hispavila.com/3ds/elimages/fig301.gif>)

Figura 67. Fuente de alimentación para 5V / 500mA

Además de estos, en el mercado se pueden encontrar los reguladores ajustables de tres patillas o más, con diferentes encapsulados en TO-220AB, TO-3 y SIL, según la potencia. Los más populares son los 78MG, LM200, LM317, LM337 y LM338, etc. Los fabricantes de los reguladores recomiendan que la tensión entregada por el secundario del transformador debe ser como mínimo 3V superior a la tensión nominal del regulador (para un 7812, la tensión del secundario mínima será de 15V o mayor), esto también tiene que ver con la intensidad que se le exija a la salida de la fuente.

En el caso de necesitar corrientes superiores a 1A, pueden utilizarse los reguladores de la serie 78HXX, LM3XX, en cápsula TO-3, capaces de suministrar 5A. El problema reside en que sólo se disponen de 5V, 12V y 15V, que en la mayoría de los casos es suficiente.

En el supuesto de necesitar una tensión regulable (ajustable) desde 1.7V a 24V el regulador a utilizar podría ser uno de la serie LM317, LM350 o LM338, la diferencia con los anteriores es que el terminal común, en lugar de estar conectado a masa, es del tipo flotante y por lo tanto esto permite ajustarle en tensión. La figura 68 a continuación nos muestra los encapsulados típicos.



(Disponible en Web: <http://www.hispavila.com/3ds/elimages/capsreg-1.gif>)

Figura 68. Forma de los encapsulados más comunes

Regulador ajustable LM317

Este regulador de tensión proporciona una tensión de salida variable sin más que añadir una resistencia y un potenciómetro. Se puede usar el mismo esquema para un regulador de la serie 78XX pero el LM317 tiene mejores características eléctricas. El aspecto es el mismo que los anteriores, pero este soporta 1,5A. el esquema a seguir es el siguiente:

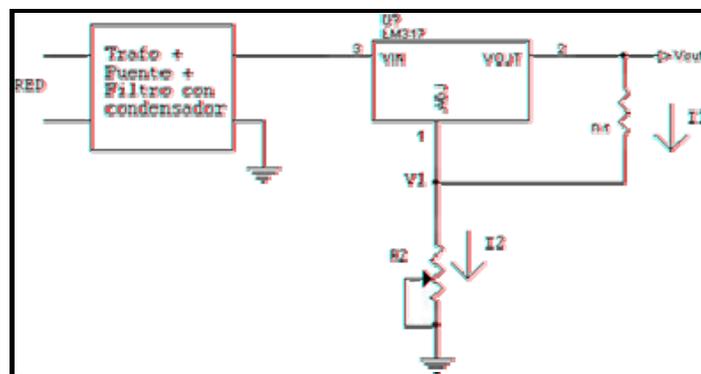


Figura 69. Fuente de alimentación con salida variable

Disipación de potencia en los reguladores

Cuando un regulador está funcionando y se calienta se debe a que parte de la potencia tomada del rectificador es disipada en el regulador. La potencia disipada depende de la corriente que se esté entregando a la carga y de la caída de tensión que haya en el regulador.

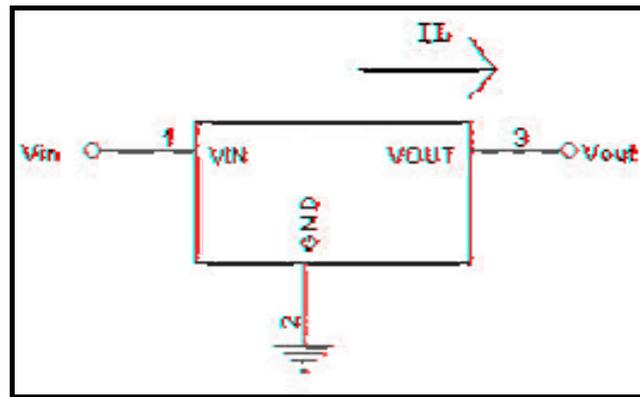


Figura 70. Sentido de circulación de la corriente por un regulador

La figura 70 nos muestra un regulador funcionando. La corriente que lo atraviesa es la corriente de la carga I_L . Recordemos también que para que un regulador funcione correctamente la tensión de entrada V_{in} tenía que ser mayor que la tensión de salida V_{out} .

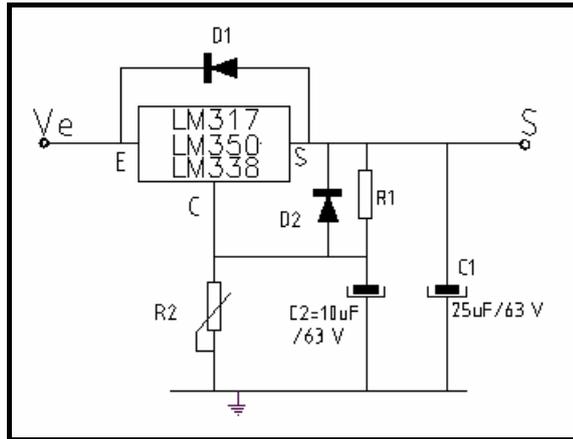
Por lo tanto la caída de tensión en el regulador V_r será:

$$V_r = V_{in} - V_{out}$$

Y la potencia disipada vendrá dada por la siguiente ecuación:

$$PD = V_r * I_L$$

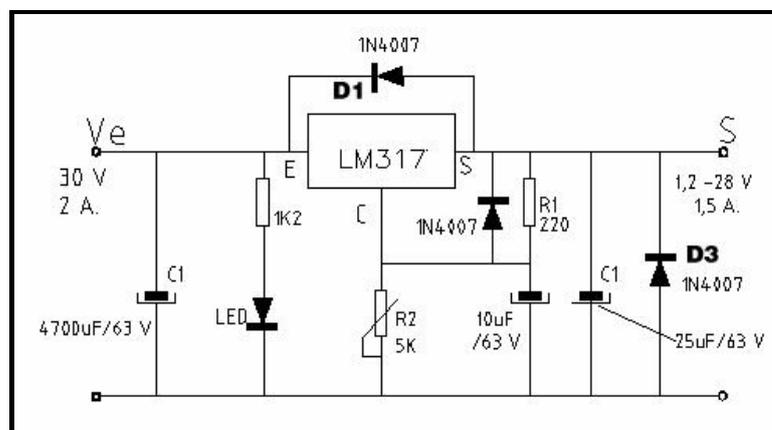
Los reguladores que hemos visto son capaces de disipar una potencia de 2 o 3 W como mucho por si solos. Si se llega a esta potencia es necesario montarlos sobre unos radiadores adecuados, que serán mas grandes cuanto mas potencia queramos disipar. Para evitar que la potencia disipada sea lo menor posible se tendrá que procurar que V_{in} no sea mucho mayor que V_{out} . En la figura 71, se presenta un esquema básico mejorado. Como ya lo habíamos mencionado, los condensadores C_1 y C_2 se emplean con el fin de eliminar tensiones alternas residuales y mejorar el rizado de la rectificación; en cuanto a los diodos D_1 y D_2 , sirven para la seguridad del regulador contra tensiones inversas y evitar las tensiones parásitas. Es muy recomendable poner los mencionados diodos.



(Disponible en Web: <http://www.hispavila.com/3ds/elimages/fig302.gif>)

Figura 71. Esquema básico mejorado de una regulación

Finalmente en la figura 72 se presenta una fuente de alimentación regulable de 1,7V a 28 V, respetando los valores de la misma. Para evitar dañar el regulador por exceso de calor, se recomienda refrigerarlo mediante un disipador de aluminio adecuado como ya lo habíamos acotado anteriormente y que se lo encuentra en los comercios especializados del ramo. El potenciómetro ajustable R2, permite ajustar la tensión de salida que se desee en cada momento. El diodo D1, protege al regulador de corrientes inversas, mientras que el diodo D3, evita que una conexión fortuita cause problemas a la fuente por polaridad invertida. Esta fuente de tensión regulada ajustable no dispone de sistema cortocircuitable externo, por lo que habría que tener cuidado de no producir ningún cortocircuito.

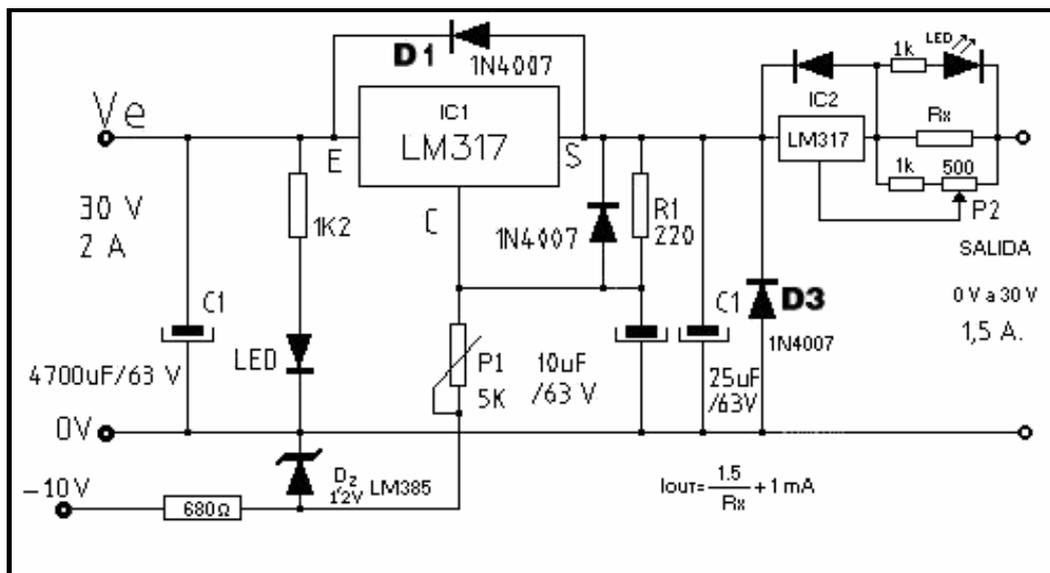


(Disponible en Web: <http://www.hispavila.com/3ds/elimages/fig303.gif>)

Figura 72. Esquema mejorado de una regulación ajustable

2.3.3 Fuente Regulada en Tensión y Corriente

Al circuito anterior, se lo puede mejorar considerablemente con sólo añadir un nuevo regulador que nos permita ajustar la corriente de salida de forma lineal mediante un potenciómetro P2 de 500W. Este regulador IC2, se conecta como regulador de corriente, lo que se consigue conectando la patilla 'flotante' o de masa a la patilla de salida mediante una resistencia Rx, que en nuestro caso se encuentra en paralelo con el conjunto de resistencias de 1k y un potenciómetro de 500W para su ajuste lineal. Además, le hemos añadido una tensión negativa de -10V, limitada por una resistencia y un diodo zener de 1,2 V (LM385), que se encargará de proporcionar un punto de tensión negativa en la patilla 'flotante' o de masa del regulador IC1, encargado de proporcionar la tensión regulable mediante el potenciómetro P1, como ocurría en el anterior esquema, esto nos permitirá obtener una tensión de salida comprendida entre 0V y los 30V.¹⁷ Nuestro esquema descrito se lo puede apreciar finalmente en la figura 73.



(Disponible en Web: <http://www.hispavila.com/3ds/elimages/fig303b.gif>)

Figura 73. Fuente de poder Regulada en Tensión y Corriente

¹⁷ HISPAVILA. Lecciones de Electrónica - Fuentes de Poder, [en línea], [España] 2000-08-21, [citado 2005-01-05]. Disponible en World Wide Web : <http://www.hispavila.com/3ds/lecciones/lecc3.htm>

2.4 Generador de Tiempo Real DS1307

(64 x 8 Serial *Real-Time Clock* RTC)

El Generador de Tiempo Real DS1307 *Serial Real-Time Clock*, es un pequeño chip reloj/calendario de baja potencia que cuenta con una memoria RAM no volátil de 56 bytes. El direccionamiento y los datos son transmitidos a través de un bus bidireccional de dos hilos en forma serial (I2C). El reloj/calendario nos provee información de segundos, minutos, horas, día, fecha, mes y año. La fecha final del mes es ajustada automáticamente para meses con menos de 31 días incluida la corrección por año bisiesto. El reloj opera en cualquiera de los dos formatos 24 horas o 12 horas con indicación AM/PM. Tiene además incorporado internamente un circuito sensor de energía que detecta cualquier anomalía por variación o falla en la tensión alimentación y que conmuta automáticamente hacia la alimentación por batería de respaldo.

2.4.1 Características Generales

- El Generador de Tiempo Real (RTC) cuenta segundos, minutos, horas, día del mes, mes, día de la semana y año incluido año bisiesto hasta el 2.100.
- Memoria RAM no volátil (NV) respaldada por batería, para almacenamiento de datos de hasta 56 bytes.
- Interface serial a dos hilos (I2C).
- Salida de onda cuadrada programable.
- Detector automático de “falla de alimentación” que conmuta al circuito de respaldo.
- Consumo menor a 500nA en modo batería de respaldo con el oscilador operando.
- Rango de temperatura industrial (opcional) de -40°C a +85°C.
- Disponible en 8 pines tipo DIP o SOIC.

- Certificación de calidad (UL *Underwriters Laboratory*).

2.4.2 Tipos disponibles en el mercado para ordenar

DS1307	8-Pin DIP (300-mil)
DS1307Z	8-Pin SOIC (150-mil)
DS1307N	8-Pin DIP (Industrial)
DS1307ZN	8-Pin SOIC (Industrial)

2.4.3 Asignación y descripción de pines

A continuación observamos en la figura 74 la asignación respectiva de los pines del RTC:

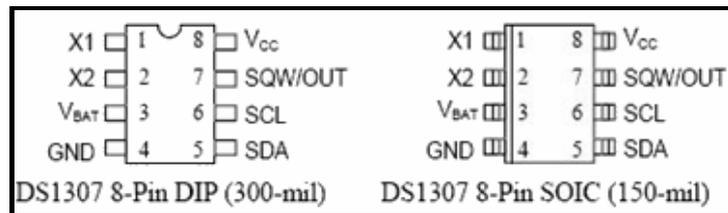


Figura 74. Asignación de pines en el DS1307

en donde:

VCC	- Tensión de alimentación principal
X1, X2	- Conexión de cristal de 32.768kHz
VBAT	- Entrada de batería de respaldo+3V
GND	- Referencia tierra o negativo.
SDA	- Transmisión serial de datos
SCL	- Transmisión serial de señal de reloj

SQW/OUT

- Salida del Oscilador Square Wave/Output Driver

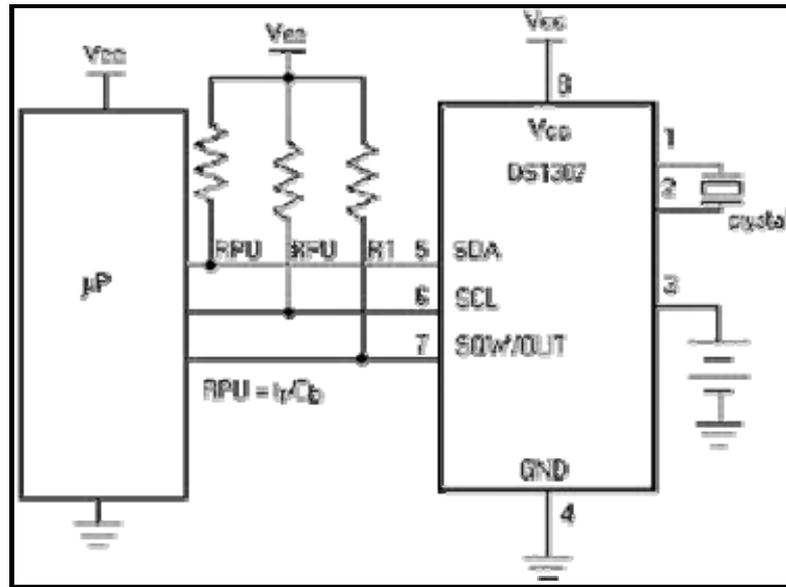


Figura 75. Circuito operativo típico de un RTC DS1307

2.4.4 Funcionamiento

El DS1307 opera en modo esclavo en el bus de comunicación serial. El acceso se obtiene implementando una condición de *START* y proporcionando un código de identificación del dispositivo seguido por una dirección de registro. Los registros seguidos son leídos en forma secuencial hasta que una condición de *STOP* sea ejecutada. Cuando la tensión de alimentación principal falla por debajo del valor de $1.25 \times V_{BAT}$, el equipo interrumpe y finaliza un acceso en marcha y resetea el contador de dirección. Las entradas al dispositivo no serán reconocidas en este momento, para evitar que datos erróneos sean escritos en la memoria del dispositivo por una falla que esta fuera de la tolerancia del sistema.

Cuando la tensión de alimentación principal falla por debajo del valor de V_{BAT} , el equipo conmuta hacia el modo de batería de respaldo en baja corriente. Al encender el dispositivo, este conmutará del modo batería hacia *VCC* siempre que el valor de *VCC* sea mayor que $V_{BAT} + 0.2V$, y reconocerá las señales de ingreso cuando *VCC* sea mayor que $1.25 \times V_{BAT}$. El diagrama de bloques en la figura 76 nos muestra los elementos principales que conforman el generador de tiempo real RTC.

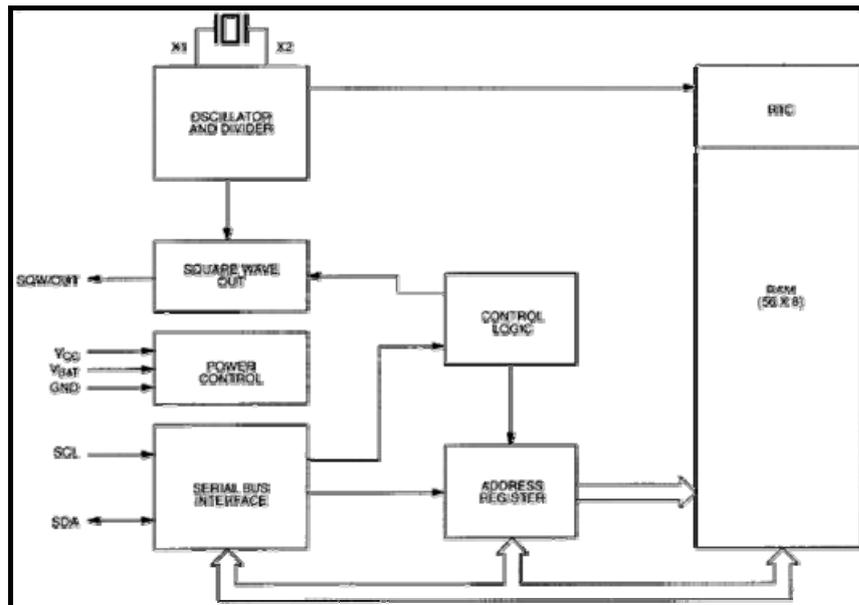


Figura 76. Diagrama de bloques del RTC DS1307

2.4.5 Descripción de las señales

VCC, GND

La tensión continua se alimenta al dispositivo a través de estos pines. El valor de VCC al ingreso es de +5VDC. Cuando se aplican los 5V estables, el dispositivo permite un acceso total a todas sus funciones de escritura y lectura. Cuando la batería de 3V esta conectada al dispositivo y el valor de VCC está por debajo de $1.25 \times V_{BAT}$, la lectura y escritura están inhabilitadas. No obstante las funciones que guardan el conteo del tiempo continúan si ser afectadas debido al bajo voltaje de entrada. Mientras la caída de tensión en VCC permanezca por debajo del valor de VBAT, la memoria RAM y la función que guarda el conteo de tiempo están conmutadas fuera de la fuente externa de alimentación hacia la alimentación por batería VBAT.

VBAT

Entrada de batería para cualquier batería estándar del tipo litio de 3V u otra fuente de energía. El voltaje de la batería debería mantenerse entre los 2.0V y 3.5V para garantizar una operación correcta. El valor puntual de voltaje en el cual el acceso de escritura al

RTC y a la memoria RAM son negados, esta fijado en el circuito interno del dispositivo y es de $1.25 \times V_{BAT}$ nominales. Una batería del tipo litio con 48 mAh/hr de carga o más, respaldará al DS1307 por más de 10 años en ausencia de energía a 25°C. Certificación de calidad UL que asegura la protección del dispositivo en contra de las corrientes de carga inversas siempre que se use junto a una batería de litio. Ver “*Conditions of Acceptability*” en la URL:

<http://www.maxim-ic.com/TechSupport/QA/ntrl.htm>

SCL (*Serial Clock Input*)

La señal del pin SCL es usada para sincronizar la transferencia de datos del interfase serial.

SDA (*Serial Data Input/Output*)

El SDA es un pin para la señal de entrada/salida de datos para el interfase serial de dos hilos. El SDA es un pin de drenaje abierto y requiere una resistencia externa de pull-up.

SQW/OUT (*Square Wave/Output Driver*)

Está habilitado cuando el bit SQWE esta en 1, entonces la salida de SQW/OUT puede tener uno de los cuatro valores de frecuencia de onda cuadrada (1Hz, 4kHz, 8kHz, 32kHz). El SQW/OUT es un pin de drenaje abierto y requiere una resistencia externa de pull-up. SQW/OUT podrá funcionar con cualquiera de las dos alimentaciones que sean aplicadas VCC o VBAT.

X1, X2

Pines para la conexión de un cristal de cuarzo estándar de 32.768kHz. El oscilador interno del circuito esta diseñado para trabajar con un cristal que tenga una carga capacitiva específica (CL) de 12.5pF. Para mayor información sobre la selección

correcta del cristal y consideraciones para el montaje, por favor consulte la “*Application Note 58, Crystal Considerations with Dallas Real-Time Clocks.*” del *Data Sheet* del fabricante. El DS1307 se puede también manejar por un oscilador externo de 32.768kHz. En esta configuración el pin X1 está conectado con la señal externa del oscilador y el pin X2 queda flotante.

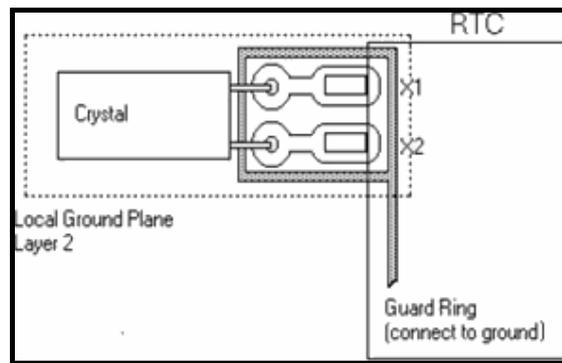


Figura 77. Montaje recomendado para el cristal

2.4.6 Precisión del reloj

La precisión del reloj es dependiente de la precisión del cristal y de la precisión que exista en el contraste entre la carga capacitiva del circuito oscilador y la carga capacitiva para las cuales el cristal fue ajustado. Se puede considerar un error adicional por las fluctuaciones en la frecuencia ocasionadas por el cambio de la temperatura. El ruido externo del circuito introducido en el circuito oscilador, puede dar lugar a que el reloj funcione más rápido. Para mayor y más detallada información, ver las notas “*Application Note 58, Crystal Considerations with Dallas Real-Time Clocks*” y la “*Application Note 95, Interfacing the DS1307 with a 8051-Compatible Microcontroller*” del *Data Sheet* del fabricante.

2.4.7 Mapa de direcciones de los registros del reloj (RTC) y de la memoria RAM

El mapa de direcciones de los registros RTC y de la RAM del DS1307 se lo puede observar en la figura 78. Los registros del reloj están ubicados en las direcciones desde la 00h hasta la 07h. Los registros de la RAM están ubicados en las direcciones desde la 08h hasta la 3Fh. Durante un acceso multi-byte, cuando el puntero alcanza la dirección

3Fh, al final del espacio de la RAM, este se guarda alrededor de la dirección 00h, al comienzo del espacio del reloj.

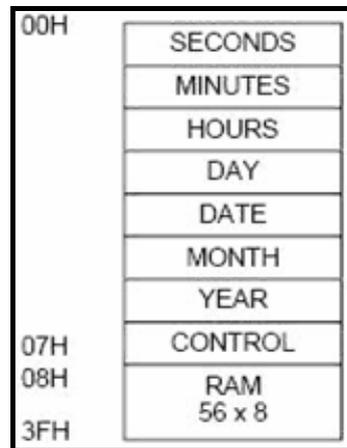


Figura 78. Mapa de direcciones del RTC y de la RAM

2.4.8 Reloj y Calendario

La información de reloj y calendario se obtiene por medio de la lectura de la combinación de *bytes* que se encuentre en los registros específicos. El tiempo del reloj y el calendario son ajustados o inicializados escribiendo la combinación de *bytes* deseada en el registro apropiado. El contenido de los registros de reloj y calendario están en formato BCD. El bit 7 del registro 0 es el bit de parada del reloj “*clock halt*” (CH). Cuando este bit es ajustado a 1, el oscilador es inhabilitado. Cuando lo limpiamos a 0, el oscilador está habilitado. Obsérvese que al encender el dispositivo, el estado inicial de todos los registros es indefinido. Por lo tanto es muy importante habilitar el oscilador (bit CH=0) durante la configuración inicial.

El DS1307 puede operar sea en el modo de 12 horas o de 24 horas. El bit 6 del registro de horas está definido como el bit de selección de modo 12 ó 24 horas. Cuando está en alto (1) el modo de 12 horas ha sido seleccionado. En el modo 12 horas, el bit 5 es el de AM/PM y en estado lógico alto inicia PM. En modo 24 horas, el bit 5 es el bit de las segundas 10 horas (20-23 horas). Activando el *START* del protocolo de comunicación a dos hilos, el valor de tiempo actual se transfiere a un segundo set de registros. La información de este tiempo se lee en estos registros secundarios, mientras el reloj puede

continuar funcionando. Esto elimina la necesidad de re-leer estos registros en caso de una actualización de los registros principales durante la lectura.

BIT7									BIT0
00H	CH	10 SECONDS			SECONDS				00-59
	0	10 MINUTES			MINUTES				00-59
	0	12 24	10 HR A/P	10 HR	HOURS				01-12 00-23
	0	0	0	0	0	DAY			1-7
	0	0	10 DATE		DATE				01-28/29 01-30 01-31
	0	0	0	10 MONTH	MONTH				01-12
	10 YEAR				YEAR				00-99
	07H	OUT	0	0	SQWE	0	0	RS1	RS0

Figura 79. Registros que guardan la configuración de tiempos

2.4.9 Registro de Control

El registro de control del DS1307 es usado para controlar la operación del pin SQW/OUT.

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
OUT	0	0	SQWE	0	0	RS1	RS0

OUT (Output control): Este bit controla el nivel de salida del pin SQW/OUT cuando la salida de onda cuadrada está inhabilitada. Si SQWE=0, el nivel lógico en el pin SQW/OUT es 1 si OUT=1 y es 0 si OUT=0.

SQWE (Square Wave Enable): Cuando este bit es ajustado en nivel lógico 1, habilitará la salida del oscilador. La frecuencia de la onda cuadrada de salida depende del valor de los bits RS0 y RS1. Con la salida de onda cuadrada ajustada a 1Hz, los registros del reloj se actualizan con el flanco de caída de la onda cuadrada.

RS (Rate Select): Estos bits controlan la frecuencia de la onda cuadrada a la salida

cuando esta salida de onda cuadrada ha sido habilitada. La figura 80 a continuación muestra una tabla con los valores de frecuencia de onda cuadrada que se pueden seleccionar con las combinaciones de los bits RS.

RS1	RS0	SQW OUTPUT FREQUENCY
0	0	1Hz
0	1	4.096kHz
1	0	8.192kHz
1	1	32.768kHz

Figura 80. Tabla con los valores de frecuencia ajustable con la combinación RS1 y RS0

2.4.10 Bus de datos serial a dos hilos

El DS1307 soporta un protocolo de comunicación con un bus de dos hilos y transmisión de datos de forma bi-direccional (I2C). Un dispositivo que envía datos sobre el bus de comunicación es designado como el transmisor y un dispositivo que recibe datos es el receptor. El dispositivo que controla el mensaje es llamado Maestro (*Master*). A los dispositivos que están controlados por el maestro se los llama esclavos (*slaves*). El bus de comunicación se debe controlar con un dispositivo maestro que genere la señal de reloj para la transmisión serial “*serial clock*” (SCL), controle el acceso al bus, y genere las condiciones de inicio (*START*) y parada (*STOP*). El DS1307 opera como esclavo en el bus serial. Una configuración típica de un bus usando el protocolo de comunicación a 2 hilos (I2C) se muestra en la figura 81.

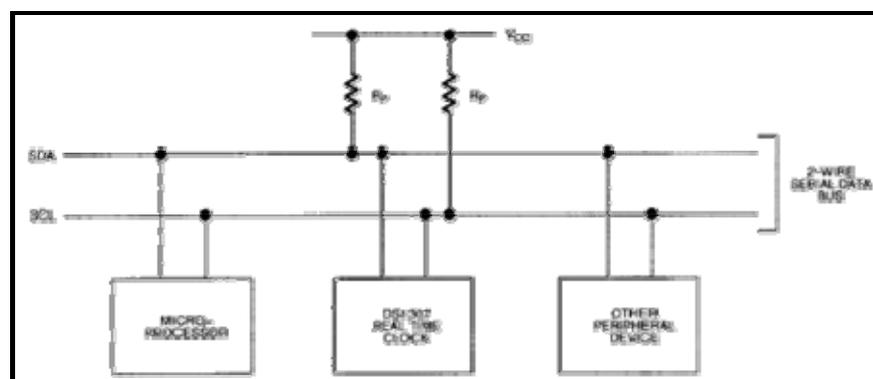


Figura 81. Configuración típica de un bus de datos serial a dos hilos

Las figuras 82, 83 y 84 detallan cómo los datos son transferidos sobre el bus de dos

hilos.

- La transferencia de datos puede iniciarse sólo cuando el bus no está ocupado.
- Durante la transferencia de datos, la línea de datos debe estar estable y siempre la línea de reloj deberá estar en ALTO (*HIGH*).

Los cambios en la línea de datos, mientras que la línea del reloj esté en ALTO, serán interpretados como señales de control.

Por consiguiente, se han definido las condiciones siguientes del bus:

Bus no ocupado (*Bus not busy*): La línea de datos y la de reloj permanecen en ALTO

Inicio de transferencia de datos (*Start data transfer*): Un cambio en el estado de la línea de datos de ALTO (*HIGH*) a BAJO (*LOW*) mientras la de reloj está en ALTO (*HIGH*), se define como una condición de inicio (*START*).

Parada de transferencia de datos (*Stop data transfer*): Un cambio en el estado de la línea de datos de BAJO (*LOW*) a ALTO (*HIGH*) mientras la línea de reloj está en ALTO (*HIGH*) se define como una condición de parada (*STOP*).

Dato válido (*Data valid*): El estado de la línea de datos representa un dato válido cuando, luego de una condición de inicio (*START*), la línea de datos permanece estable mientras dure el período ALTO (*HIGH*) de la señal de reloj. Los datos en la línea pueden ser cambiados mientras dure el período BAJO (*LOW*) de la señal de reloj. Hay un pulso de reloj por bit de datos.

Cada dato transferido es iniciado por una condición de *START* y terminado por una condición de *STOP*. El número de bytes de datos transferidos entre una condición de *START* y *STOP* es ilimitado, y está determinado por el dispositivo Maestro (*Master*). La información es transferida como “*byte inteligente*” y cada receptor la reconoce con un noveno bit. Dentro de las especificaciones del bus a dos hilos se definen un modo regular (velocidad del reloj 100kHz) y un modo rápido (velocidad del reloj 400kHz). El DS1307 opera sólo en modo regular (100kHz).

Reconocimiento (*Acknowledge*): Cada dispositivo receptor, cuando es direccionado, esta obligado a generar un reconocimiento (*acknowledge*) luego de la recepción de cada *byte*. El dispositivo Maestro debe generar un pulso de reloj extra que está asociado con el bit de reconocimiento. Un dispositivo que reconoce debe bajar la línea SDA durante el pulso de reloj de reconocimiento de modo que la línea SDA permanezca estable en BAJO (*LOW*) durante el período ALTO (*HIGH*) del pulso de reloj relacionado con el reconocimiento. Por supuesto, el tiempo de ajuste (*setup*) y espera (*hold*) deben ser tomados en cuenta. El maestro (*master*) debe señalar un fin de datos al esclavo para no generar un bit de reconocimiento sobre último *byte* que ha sido registrado por el esclavo. En este caso, el esclavo debe dejar la línea de datos ALTA (*HIGH*) para permitir al maestro (*master*) generar la condición de PARADA (*STOP*).

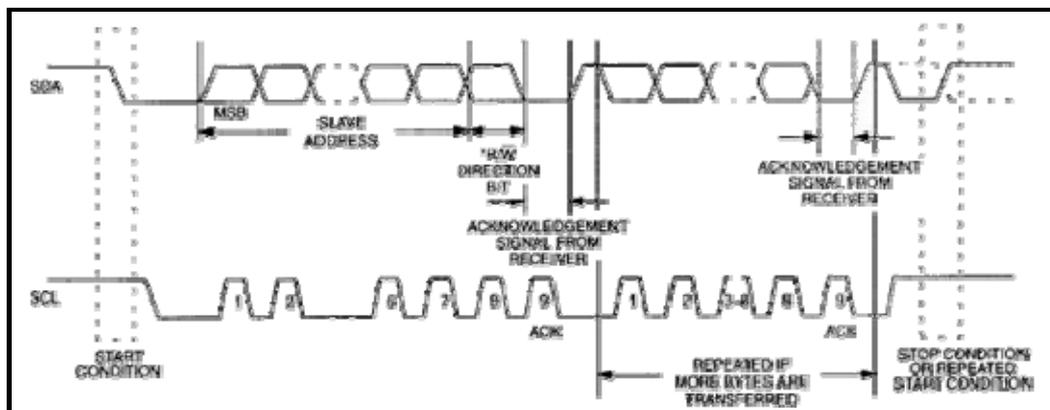


Figura 82. Transferencia de datos sobre un bus serial a dos hilos (I2C)

Dependiendo del estado del bit R/W, existen dos tipos de transferencia posibles:

Transferencia de datos de un maestro transmisor a un esclavo receptor El primer *byte* transmitido por el maestro es la dirección del esclavo. Después sigue un número de *bytes* de datos. El esclavo devuelve un bit de reconocimiento después de cada *byte* recibido. Los datos son transferidos con el bit más significativo (MSB) primero.

Transferencia de datos de un esclavo transmisor a un maestro receptor El primer *byte* (dirección del esclavo) es transmitido por el maestro. Entonces el esclavo devuelve un bit de reconocimiento. Esto es seguido por el esclavo que transmite un número de *bytes* de datos. El maestro devuelve un bit de reconocimiento después de todos los otros *bytes* recibidos que el último *byte*. Al final del último *byte* recibido, un " no

reconocimiento" es devuelto. El dispositivo maestro genera todos los pulsos de reloj seriales y las condiciones de inicio (*START*) y parada (*STOP*). Una transferencia es terminada con una condición de parada (*STOP*) o con una condición de inicio (*START*) repetida. Ya que una condición de arranque repetida es también el principio de la siguiente transferencia serial, el Bus no será liberado. Los datos son transferidos con el bit más significativo (MSB) primero.

El DS1307 puede operar en los dos modos siguientes:

Modo esclavo receptor (modo escritura DS1307): Los datos seriales y el reloj son recibidos por SDA Y SCL respectivamente. Después de que cada byte es recibido un bit de reconocimiento es transmitido. Las condiciones de inicio (*START*) y parada (*STOP*) son reconocidas como el principio y el final de la transferencia serial. El reconocimiento de dirección es realizado por el hardware después de la recepción de la dirección del esclavo y del bit de direccionamiento * “direction bit”.

El *byte* de dirección es el primer *byte* recibido después de que la condición de inicio es generada por el maestro. El *byte* de dirección contiene 7 bits de la dirección del DS1307, la que es 1101000, seguido del “direction bit” (R/W) que, para la escritura, es 0. Después de recibir y decodificar el *byte* de dirección el dispositivo saca un reconocimiento sobre la línea SDA. Después de que el DS1307 reconoce la dirección del esclavo mas el bit escrito, el maestro transmite una dirección de registro al DS1307. Esto pondrá un indicador de registro sobre el DS1307. El maestro entonces comenzará a transmitir cada *byte* de datos con el DS1307 que reconoce cada *byte* recibido. El maestro generará una condición de parada al terminar de escribir los datos.

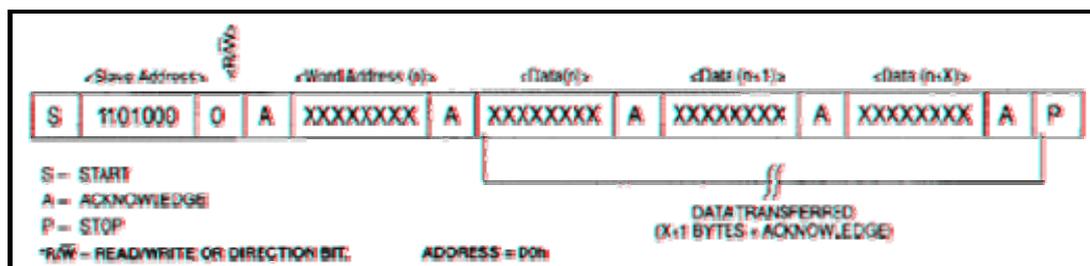


Figura 83. Escritura de datos del modo esclavo receptor

Modo esclavo transmisor (modo lectura DS1307): El primer *byte* es recibido y tratado como en el modo esclavo receptor. Sin embargo en este modo, el bit de direccionamiento “direction bit” indicará que la dirección de transferencia es inversa. Los datos seriales son transmitidos por el DS1307 sobre la línea SDA mientras que la señal de reloj serial esta ingresando sobre la línea SCL.

Las condiciones de inicio (*START*) y parada (*STOP*) son reconocidas como el principio y final de la transferencia serial. El *byte* de dirección es el primer *byte* recibido luego que la condición de inicio es generada por el maestro. El *byte* de dirección contiene los 7 bits de dirección del DS1307 que es 1101000, seguido por el bit de direccionamiento “direction bit” (R/W) que, para la lectura, es 1.

Después de recibir y decodificar el *byte* de dirección el dispositivo introduce un reconocimiento sobre la línea SDA. El DS1307 entonces comienza a transmitir datos iniciando con la dirección de registro señalada por el indicador de registro. Si el indicador de registro no es escrito antes de la iniciación de un modo de lectura la primera dirección que es leída es la última almacenada en el indicador de registro. El DS1307 debe recibir un "no reconocimiento" al terminar la lectura.

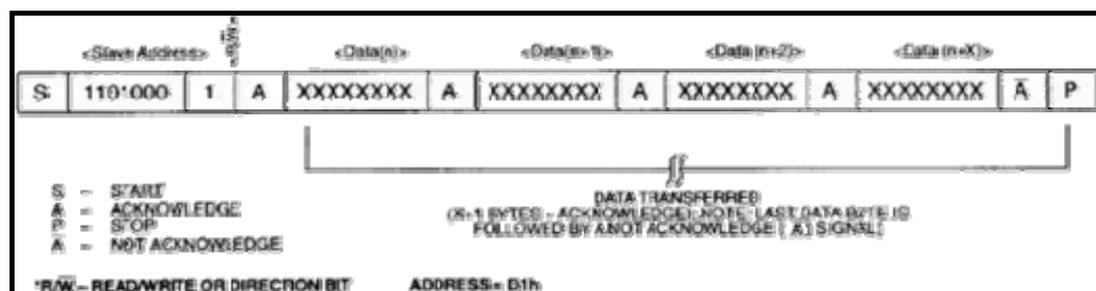


Figura 84. Lectura de datos del modo esclavo transmisor

2.4.11 Valores absolutos máximos

- Voltaje en cualquier pin con respecto a tierra: -0.5v a +7.0V
- Temperatura de almacenaje: -55°C a +125°C
- Temperatura de soldadura: 260°C por 10 segundos DIP

Ver Estándar JPC/JEDEC J-STD-020A para Dispositivos de Montaje Superficiales.

Estos son sólo valores extremos y la operación funcional del dispositivo en estas o en cualquier otra condición sobre de aquellas indicadas en las secciones de operación de esta especificación no está implícita. La exposición a condiciones dentro del rango de los valores absolutos máximos durante períodos extensos de tiempo puede afectar la confiabilidad.

Range	Temperature	V _{CC}
Commercial	0°C to +70°C	4.5V to 5.5V V _{CC1}
Industrial	-40°C to +85°C	4.5V to 5.5V V _{CC1}

2.4.12 Valores de DC recomendados para cumplir las condiciones de operación

PARAMETER	SYMBOL	MIN	TYP	MAX	UNITS
Supply Voltage	V _{CC}	4.5	5.0	5.5	V
Logic 1	V _{IH}	2.2		V _{CC} + 0.3	V
Logic 0	V _{IL}	-0.5		+0.8	V
V _{BAT} Battery Voltage	V _{BAT}	2.0		3.5	V

2.4.13 Características eléctricas de corriente continua

PARAMETER	SYMBOL	MIN	TYP	MAX	UNITS	NOTES
Input Leakage (SCL)	I _{LI}			1	μA	
I/O Leakage (SDA & SQW/OUT)	I _{LO}			1	μA	
Logic 0 Output (I _{OL} = 5mA)	V _{OL}			0.4	V	
Active Supply Current	I _{CCA}			1.5	mA	7
Standby Current	I _{CCS}			200	μA	1
Battery Current (OSC ON); SQW/OUT OFF	I _{BAT1}		300	500	nA	2
Battery Current (OSC ON); SQW/OUT ON (32kHz)	I _{BAT2}		480	800	nA	
Power-Fail Voltage	V _{PF}	1.216 x V _{BAT}	1.25 x V _{BAT}	1.284 x V _{BAT}	V	8

2.4.14 Características eléctricas de corriente alterna

PARAMETER	SYMBOL	MIN	TYP	MAX	UNITS	NOTES
SCL Clock Frequency	f_{SCL}	0		100	kHz	
Bus Free Time Between a STOP and START Condition	t_{BUF}	4.7			μs	
Hold Time (Repeated) START Condition	t_{HD_STA}	4.0			μs	3
LOW Period of SCL Clock	t_{LOW}	4.7			μs	
HIGH Period of SCL Clock	t_{HIGH}	4.0			μs	
Set-up Time for a Repeated START Condition	t_{SU_STA}	4.7			μs	
Data Hold Time	t_{HD_DAT}	0			μs	4.5
Data Set-up Time	t_{SU_DAT}	250			ns	
Rise Time of Both SDA and SCL Signals	t_R			1000	ns	
Fall Time of Both SDA and SCL Signals	t_F			300	ns	
Set-up Time for STOP Condition	t_{SU_STO}	4.7			μs	
Capacitive Load for each Bus Line	C_B			400	pF	6
I/O Capacitance ($T_A = 25^\circ C$)	C_{IO}		10		pF	
Crystal Specified Load Capacitance ($T_A = 25^\circ C$)			12.5		pF	

Notas referentes a los cuadros anteriores:

1. **Standby Current (Corriente en estado de espera):** Especificada con $VCC=5.0V$ y $SDA, SCL = 5.0V$.
2. **Battery Current (Corriente de la batería):** Especificada con $VCC = 0V$, $V_{BAT}=3V$.
3. **Hold Time (Repeated) START Condition (Tiempo de espera en condición de inicio):** Luego de este período, el primer pulso de reloj es generado.
4. **Data Hold Time (Tiempo de espera de datos):** El dispositivo internamente debe proporcionar un tiempo de espera de al menos 300ns para la señal de SDA (con referencia al $V_{IH\ MIN}$ de la señal de SCL) para cruzar la región indefinida del flanco de bajada de SCL.
5. **Data Hold Time (Tiempo de espera de datos):** El máximo $t_{HD:DAT}$ debe ser cumplido solo si el dispositivo no extiende el período BAJO (*LOW*) de la señal de SCL.
6. **Capacitive Load for each Bus Line (Carga capacitiva por cada línea del bus):** Valor total de capacitancia de una línea del bus en pF.

7. **Active Supply Current (Suministro activo de corriente):** Con el reloj de la línea SCL a la frecuencia máxima = 100kHz.
8. **Power-Fail Voltage (Voltaje de falla de alimentación):** Medido con un VBAT=3.0V.

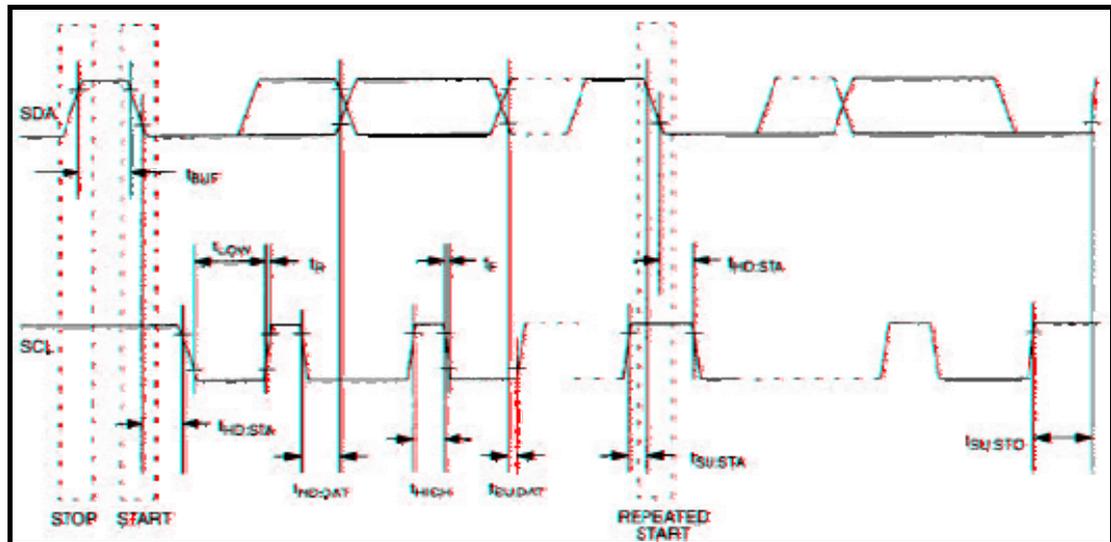


Figura 85. Diagrama de tiempos DS1307

CAPITULO III

DISEÑO Y CONSTRUCCIÓN

3.1 Diagrama de bloques y circuitos:

A continuación presentaremos el diseño del circuito completo y se irán explicando uno a uno, cómo funcionan, cómo están conformados y qué factores se tomaron en cuenta para el diseño de cada uno de los distintos bloques que estructuran este circuito.

3.1.1 Bloque 1: Fuente de alimentación:

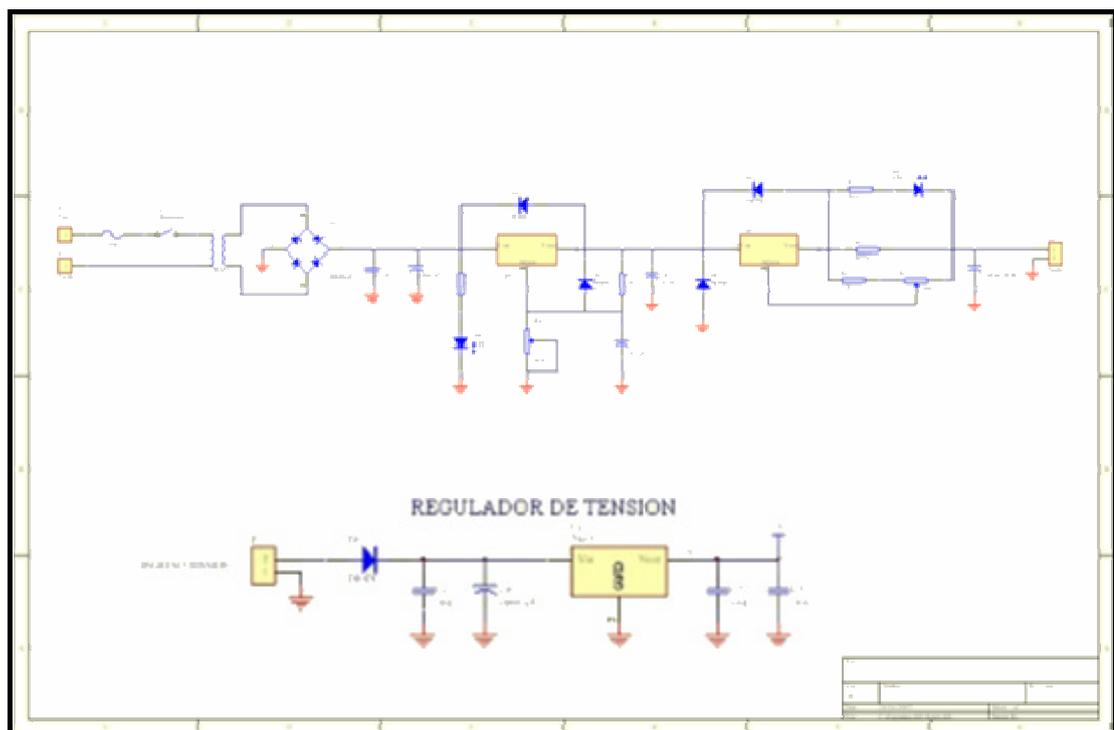


Figura 86. Fuente de Alimentación.

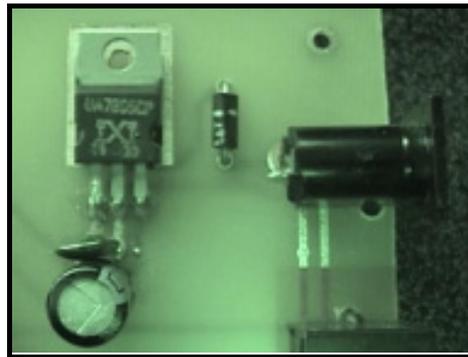


Figura 87. Fotografía de la Fuente

En este bloque podemos observar el circuito electrónico de como esta conformada nuestra fuente de alimentación, misma que se conectara a un siguiente regulador mas de tensión para garantizar la correcta operación de nuestro equipo, dicha fuente suministrara la tensión y corriente estabilizados de acuerdo a los criterios estudiados anteriormente en el capítulo 2, Características del Equipo, en el parágrafo 2.3.3 referente a fuentes de tensión reguladas y estabilizadas, por lo que no volveremos a profundizar en este tema.

3.1.2 Bloque 2: Generador de tiempo real:

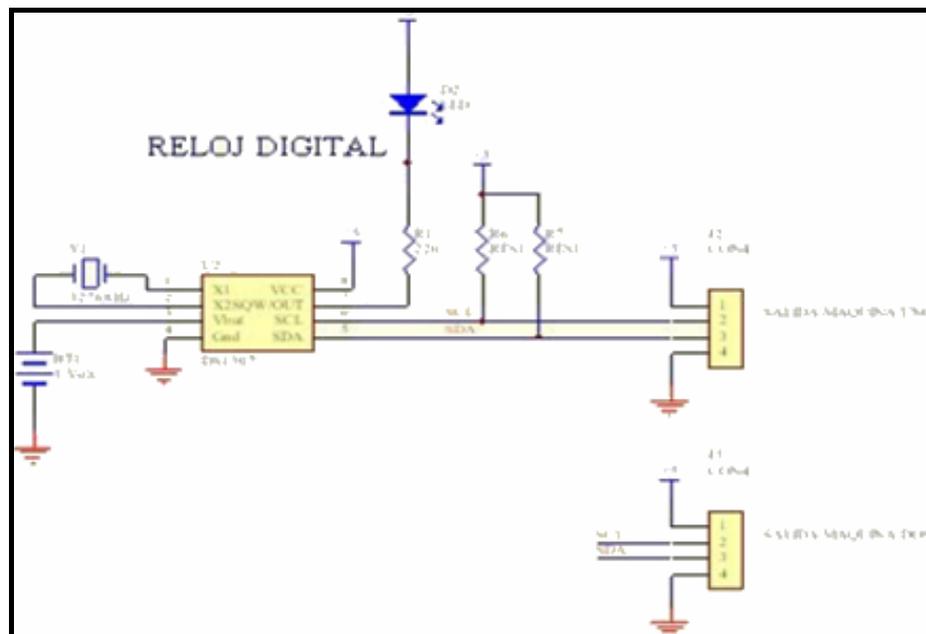


Figura 88. Generador de Tiempo Real

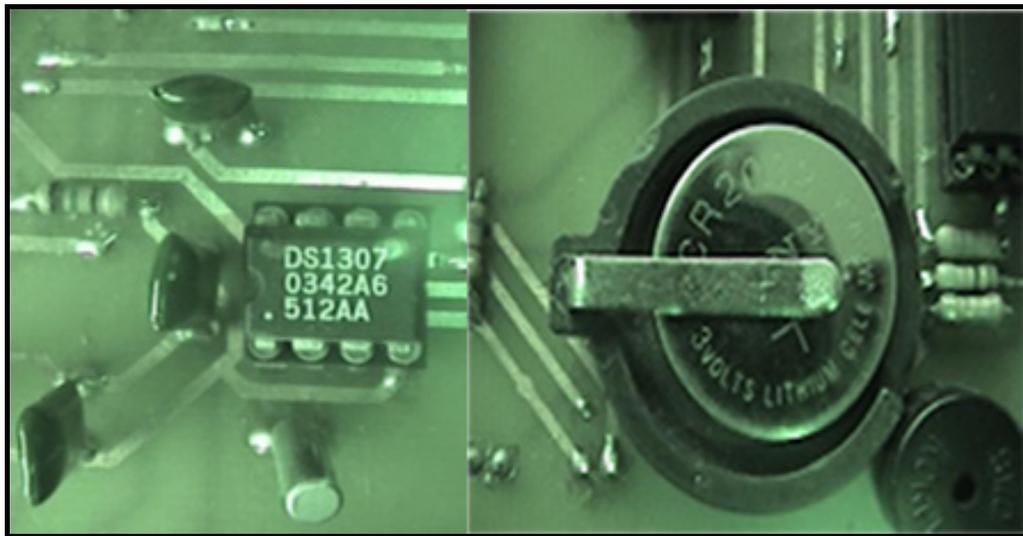


Figura 89. Fotografía del Generador de Tiempo Real

En las figuras anteriores (en diagrama y en foto), podemos apreciar el bloque correspondiente al Generador de tiempo real o Reloj Digital, mismo que esta conformado por el circuito integrado DS1307 que también lo estudiamos anteriormente y que por lo tanto tampoco volveremos a profundizar sobre su funcionamiento, aquí podemos además observar tanto físicamente como en diagrama como esta dispuesto para su conexión. En los elementos necesarios que tiene este circuito para operar correctamente observamos en primer lugar el cristal de cuarzo de 32768 Hz (33KHz) que genera las pulsaciones para la sincronización de los tiempos programados y para la comunicación serial I2C a través de las líneas SCL y SDA, la batería de respaldo que guarda la configuración del reloj, las resistencias de pull up que van siempre sobre las líneas de comunicación, y que en este caso se conectan a los distintos PICS esclavos uno por cada maquina a monitorear, y finalmente un led conectado a la salida que sirve como piloto para monitorear la operación del reloj y la sincronización del tiempo.

3.1.3 Bloque 3: Teclado matricial:

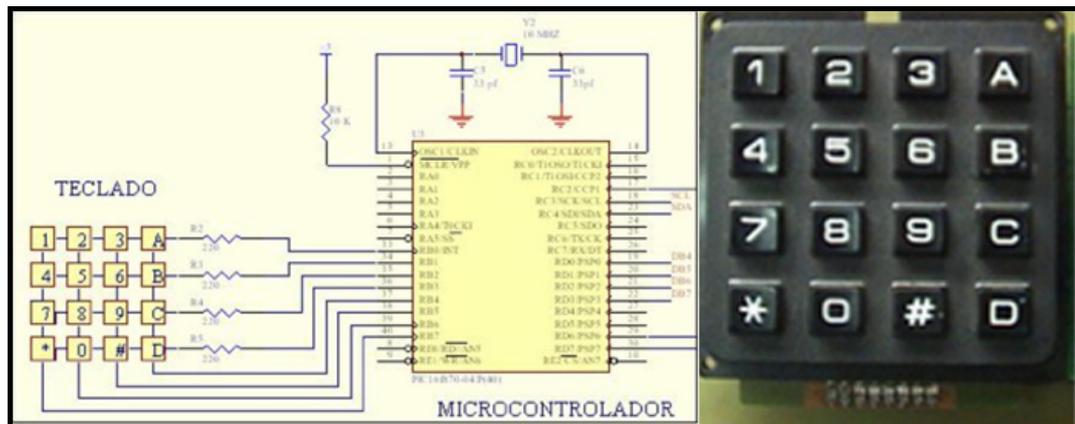


Figura 90. Teclado Matricial, Diagrama y Foto

En este tercer bloque vemos la disposición de la conexión del teclado matricial 4x4 que nos servirá para digitar los comandos de ingreso al equipo supervisor, su conexión es sencilla y lo único que requiere es de las resistencias de pull up en serie a los puertos de entrada RB0 al RB3 pines del 33 al 36 del PIC. Sus distintas configuraciones también las estudiamos anteriormente en el párrafo 1.4 correspondiente a la conexión de los periféricos más comunes a los PICS.

3.1.4 Bloque 4: *Display* de cristal líquido LCD:

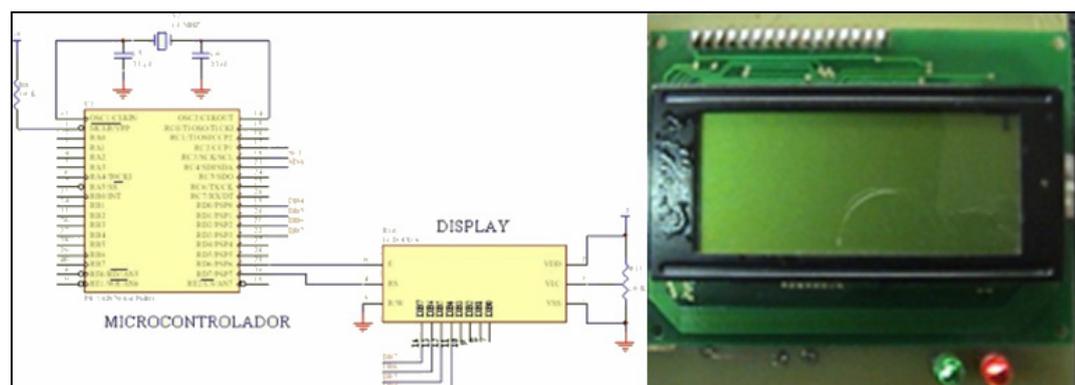


Figura 91. *Display* LCD, diagrama y foto

Al igual que en el caso del teclado matricial, el *display* de cristal liquido o LCD se lo estudio a profundidad anteriormente en el capítulo 1 en lo referente a conexión de periféricos mas comunes, solo nos resta por observar en el ejemplo practico, como

hemos conectado nuestro LCD al microcontrolador en el bus de datos de salida, su habilitación, su alimentación y el potenciómetro de control de contraste.

3.1.5 Bloque 5: PIC 16F872 (*slave*):

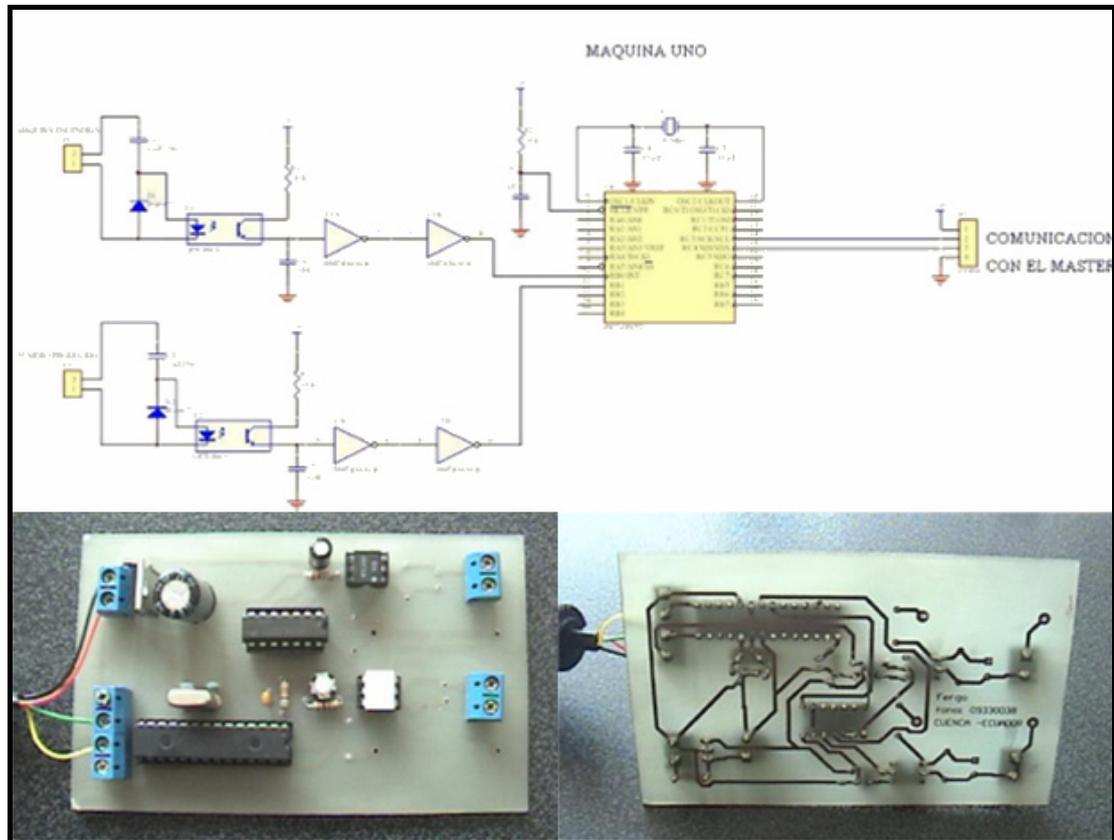


Figura 92. Diagrama y foto PIC 16F872 (*Slave*)

En este penúltimo bloque podemos observar la conexión del PIC esclavo (*slave*) hacia los dos sensores que captan las señales de máquina encendida y unidades producidas, con estos datos y con el dato de tiempo real que le solicita al PIC maestro (*master* 16F877) a través del puerto I2C, este realiza el cálculo matemático del rendimiento y le entrega este dato al master para que lo presente en el LCD; de ser el caso, este PIC esclavo también entrega los datos de la máquina encendida y número de unidades producidas para los acumuladores de horas totales trabajadas y unidades totales producidas que también se los podrán presentar con el PIC master al LCD dependiendo del menú que se elija a través del teclado.

3.1.6 Bloque 6: PIC 16F877 (*master*):

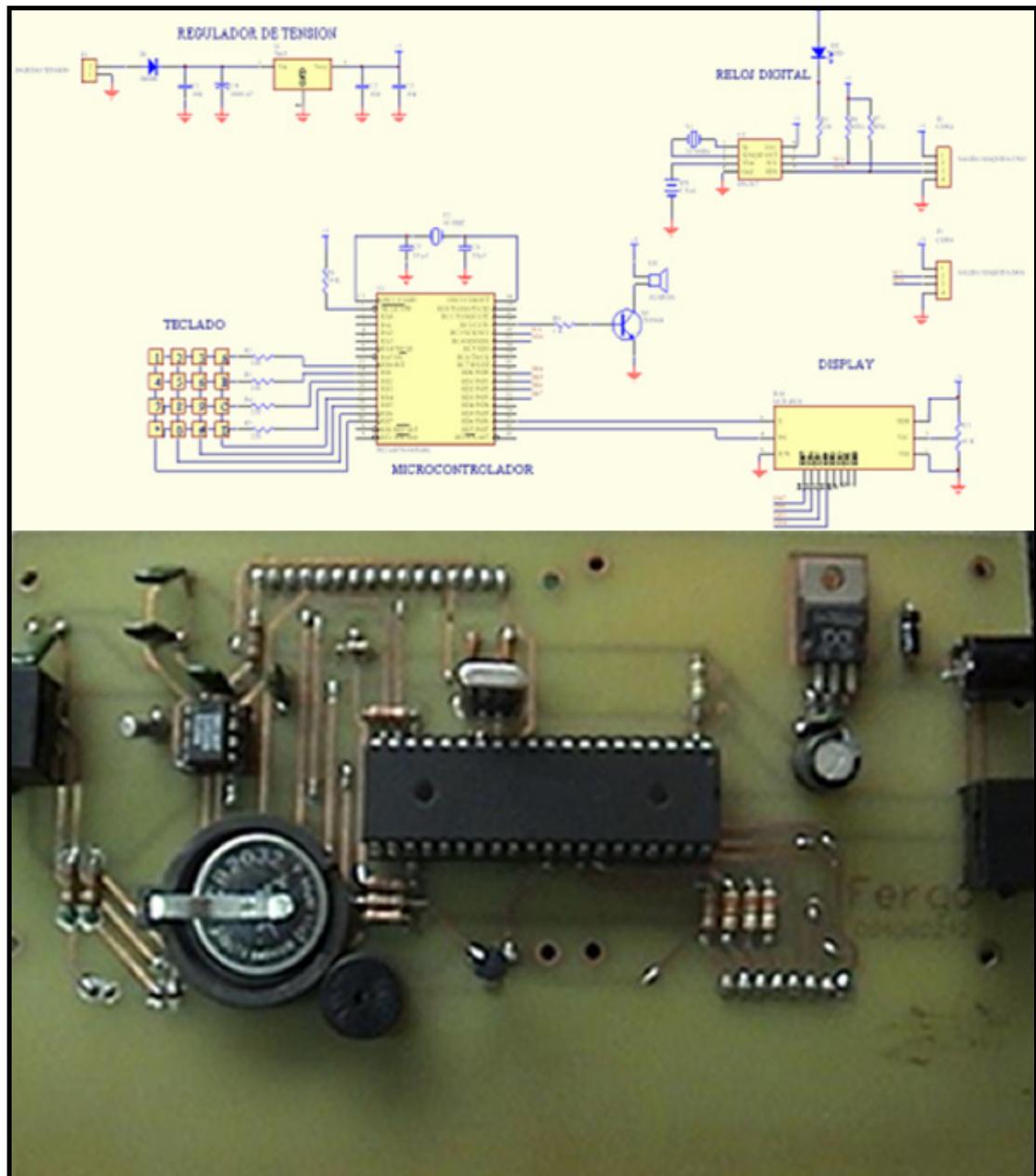


Figura 93. Diagrama y foto PIC 16F877 (*Master*)

Finalmente, en este último bloque podemos observar como el PIC 16F877 trabaja como *MASTER* para los demás esclavos que son los PIC 16F872 de cada máquina y el reloj digital DS1307, además de que al ser el PIC principal, llevará consigo el programa principal que maneja los módulos de operación de los periféricos que son el teclado y el LCD en donde se presentarán los datos de rendimiento, horas totales trabajadas y unidades totales producidas por cada máquina a ser supervisada por nuestro equipo. Se

observa además la conexión del cristal de cuarzo de 10MHz que entrega la señal de sincronización y tiempos de instrucciones y una pequeña alarma acústica que dará un bip con cada pulsación del teclado.

3.2 Programación y diagramas de flujo

En el siguiente párrafo nos encargaremos de presentar uno a uno los diagramas de flujo que se siguieron como parte de la filosofía de programación o algoritmo de programación, que optamos para el desarrollo de este proyecto.

3.2.1 Diagrama de flujo principal y subrutina de interrupción

DIAGRAMA DE FLUJO PRINCIPAL

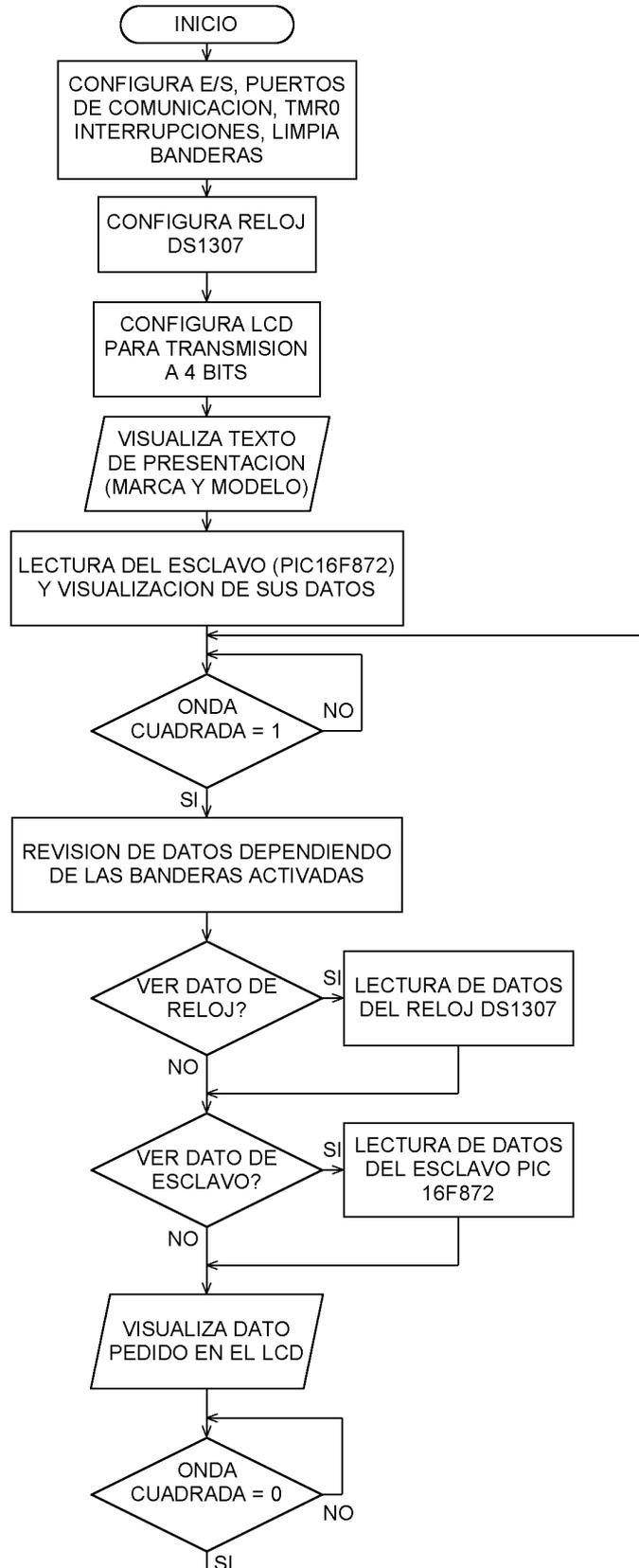
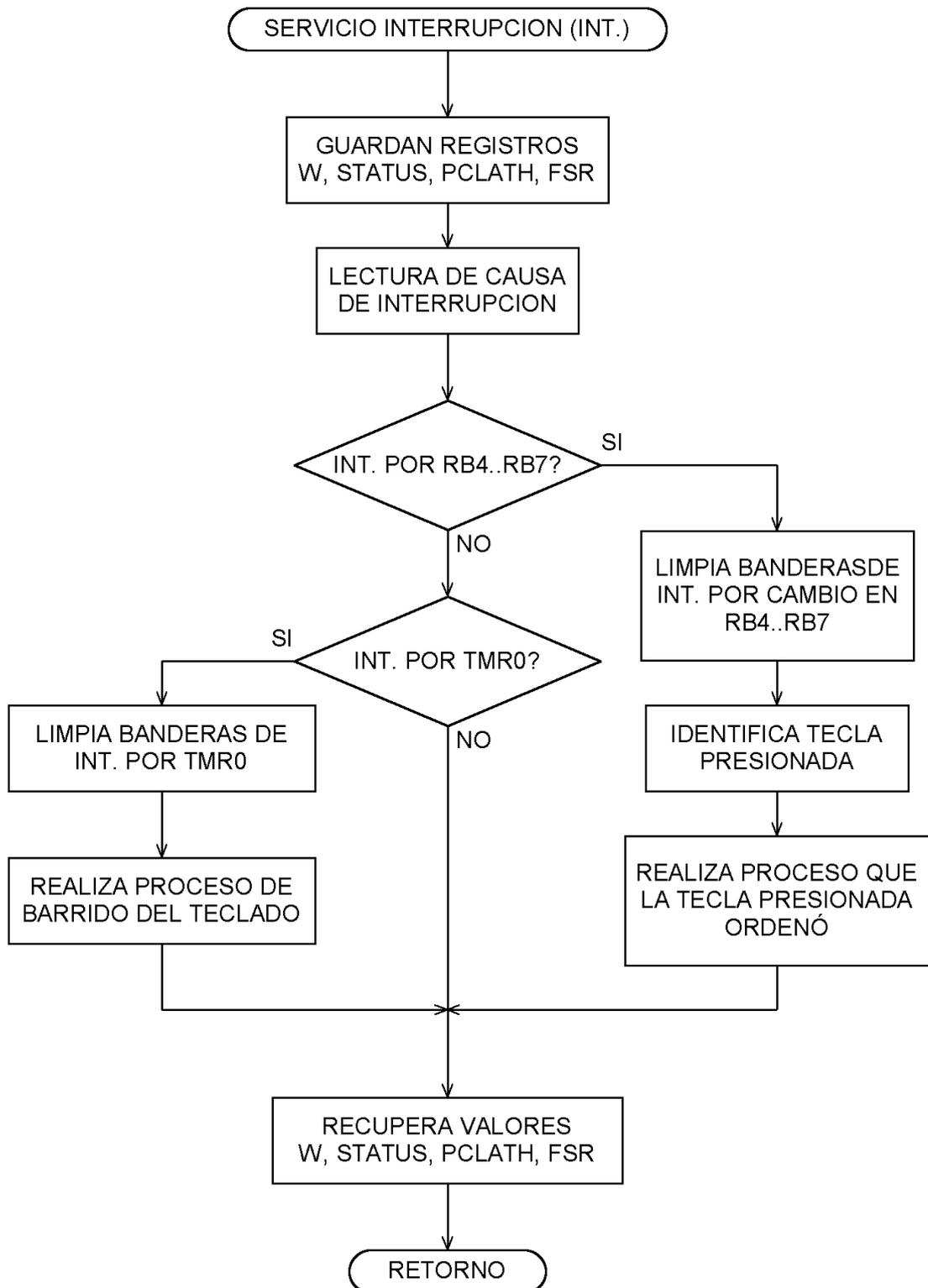
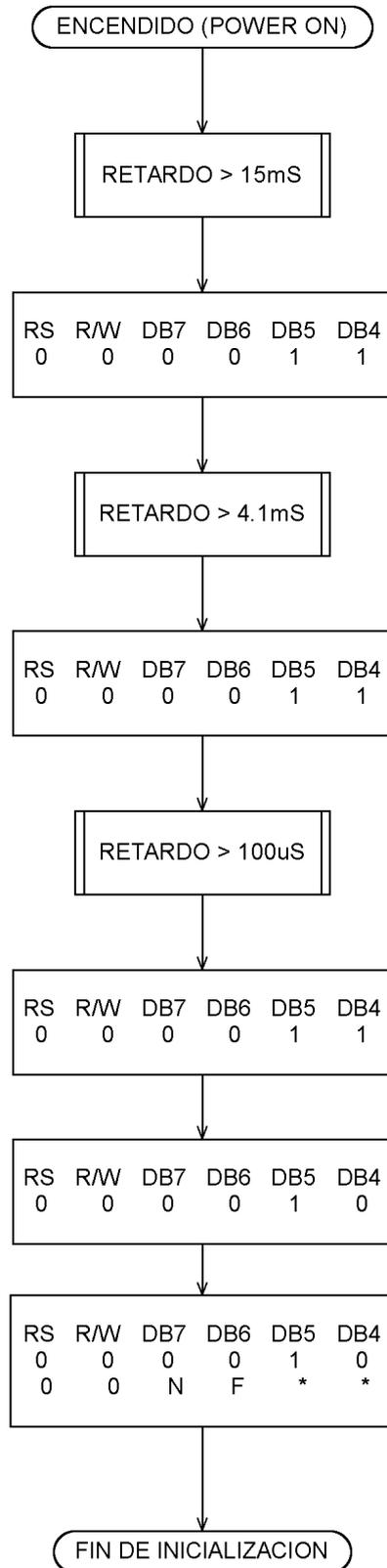


DIAGRAMA DEL SERVICIO DE INTERRUPCION



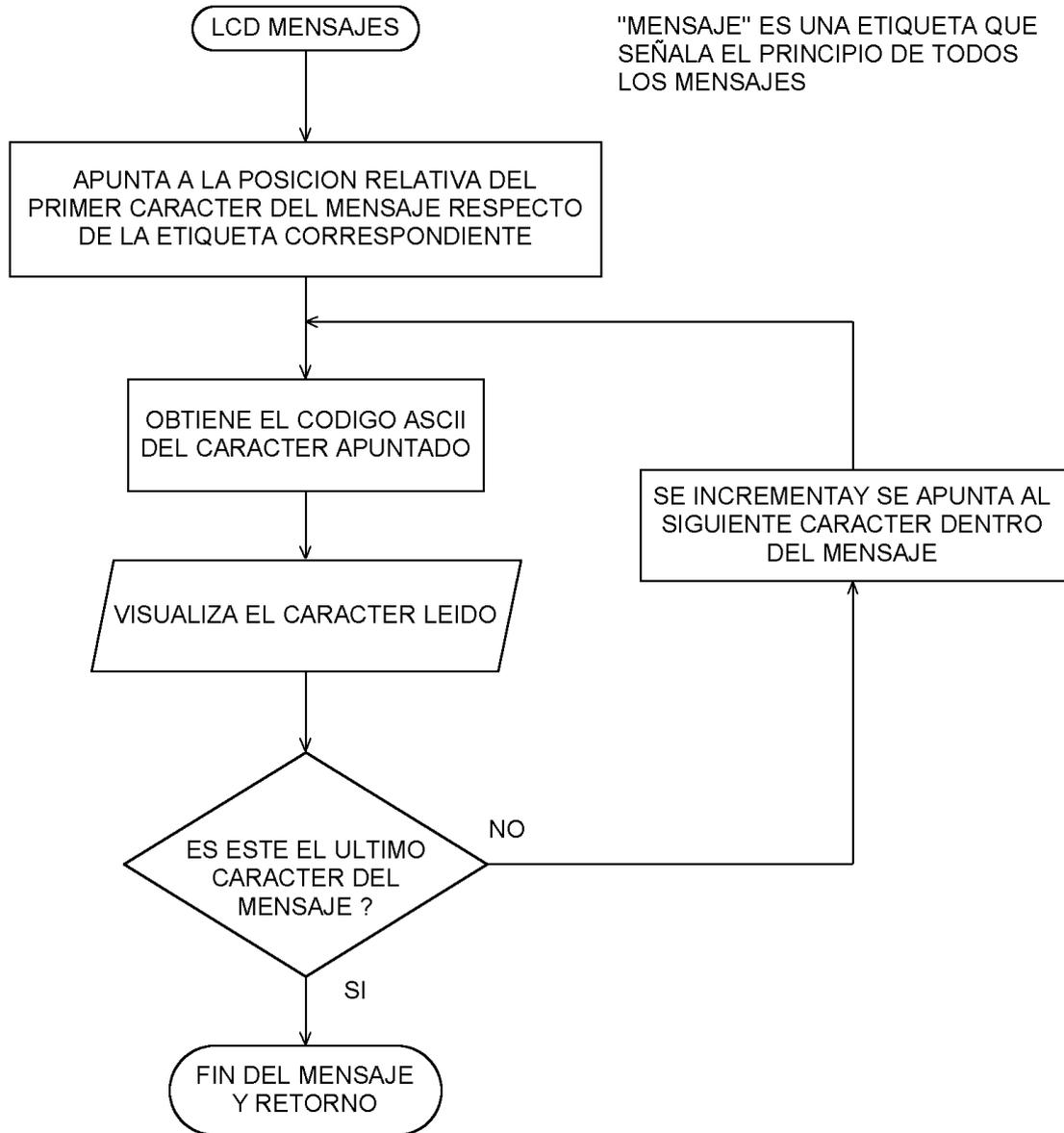
3.2.2 Diagrama de flujo del LCD y subrutinas

DIAGRAMA DE FLUJO PARA INICIALIZACION DEL LCD A 4 BITS



N=0 PANTALLA LCD 1 LINEA
 N=1 PANTALLA LCD 2 O MAS LINEAS
 F=0 CARACTERES 5x 7 PUNTOS
 F=1 CARACTERES 5x10 PUNTOS

DIAGRAMA DE FLUJO DE LA SUBROUTINA PARA LA VISUALIZACION DE MENSAJES FIJOS



3.2.3 Diagrama de flujo del Reloj DS1307 y subrutinas

DIAGRAMA DE FLUJO DE LA SUBROUTINA RELOJ

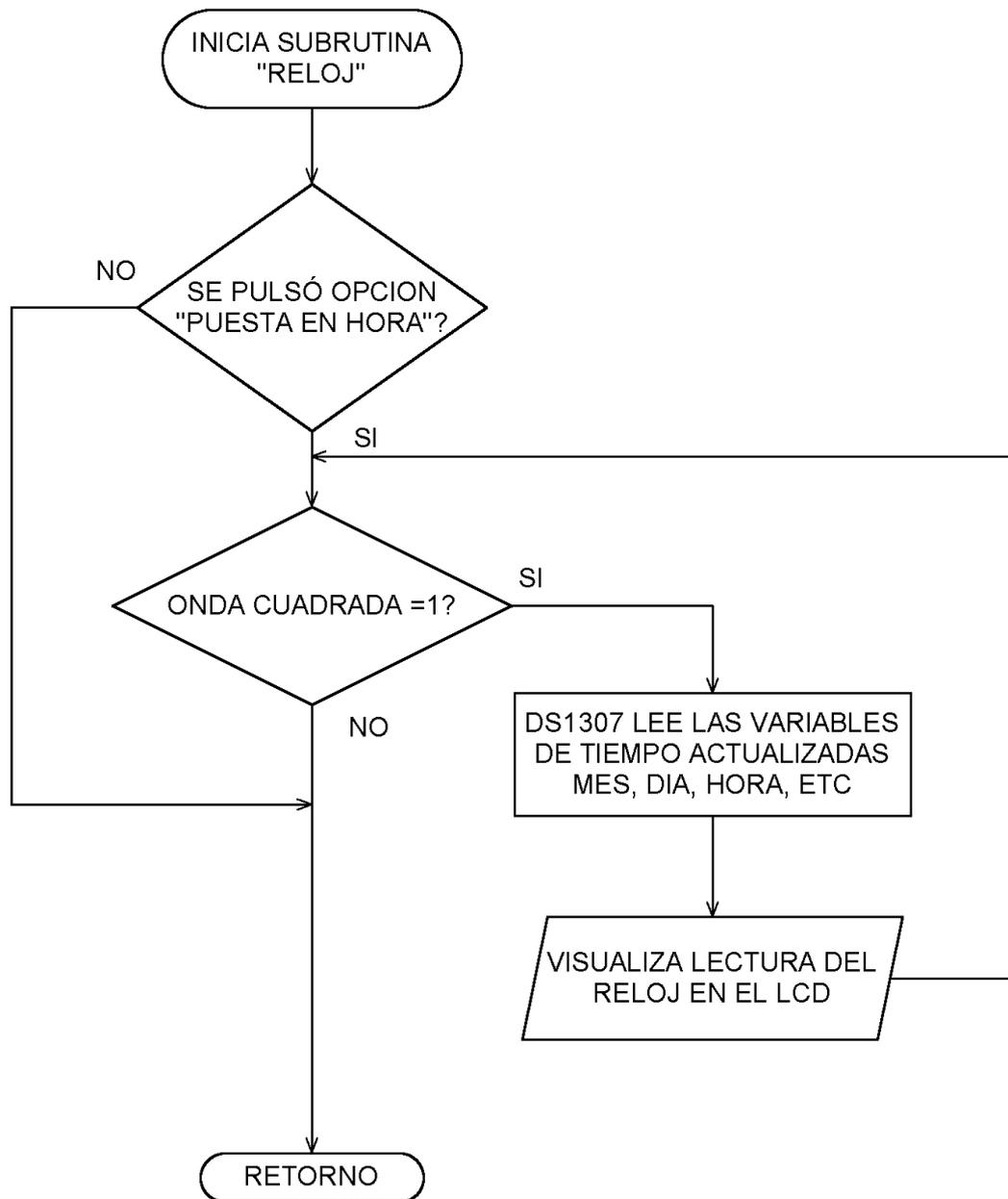


DIAGRAMA DE FLUJO PRINCIPAL DEL RELOJ DS1307

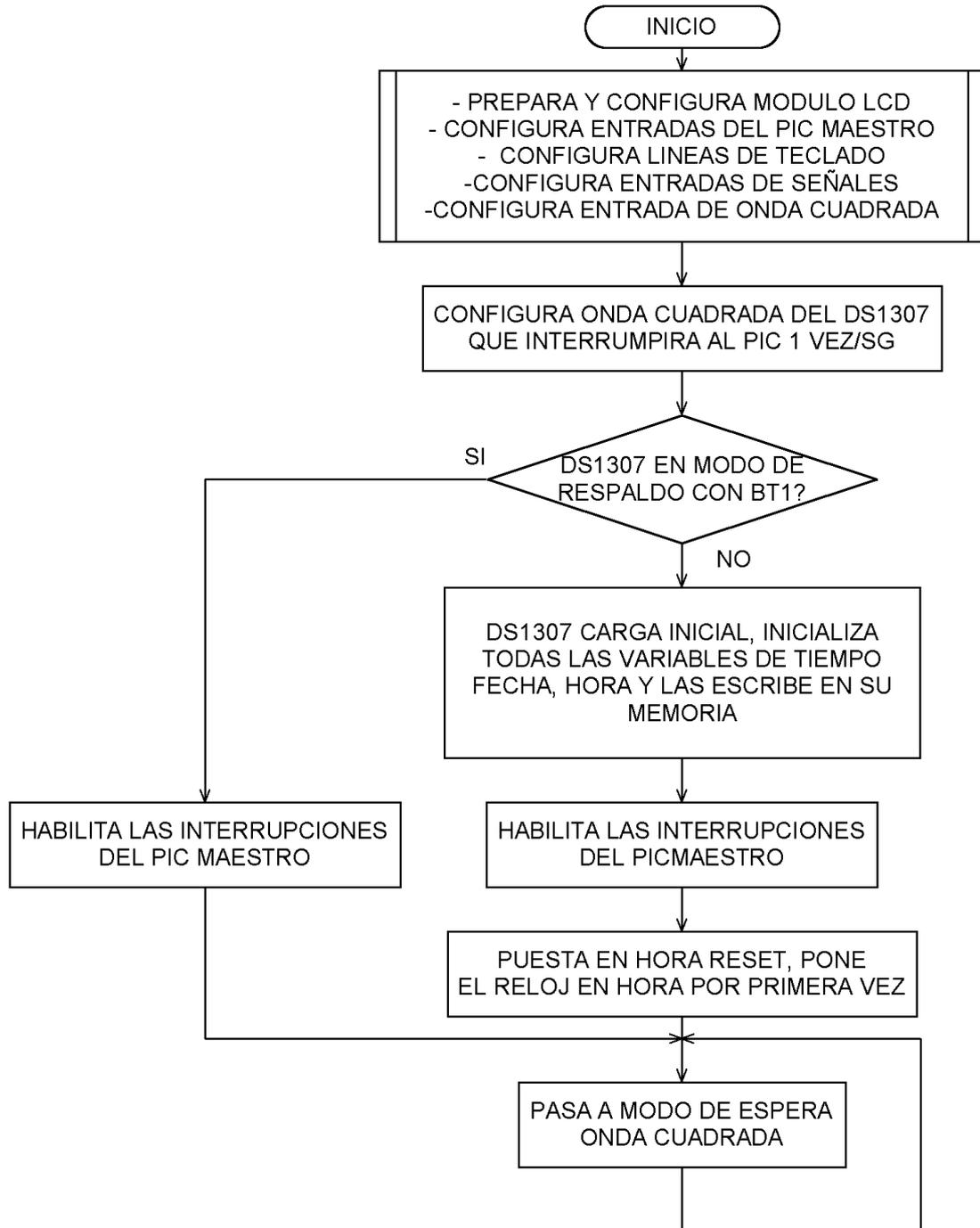


DIAGRAMA DE LA SUBROUTINA DEL SERVICIO DE INTERRUPCION DEL DS1307

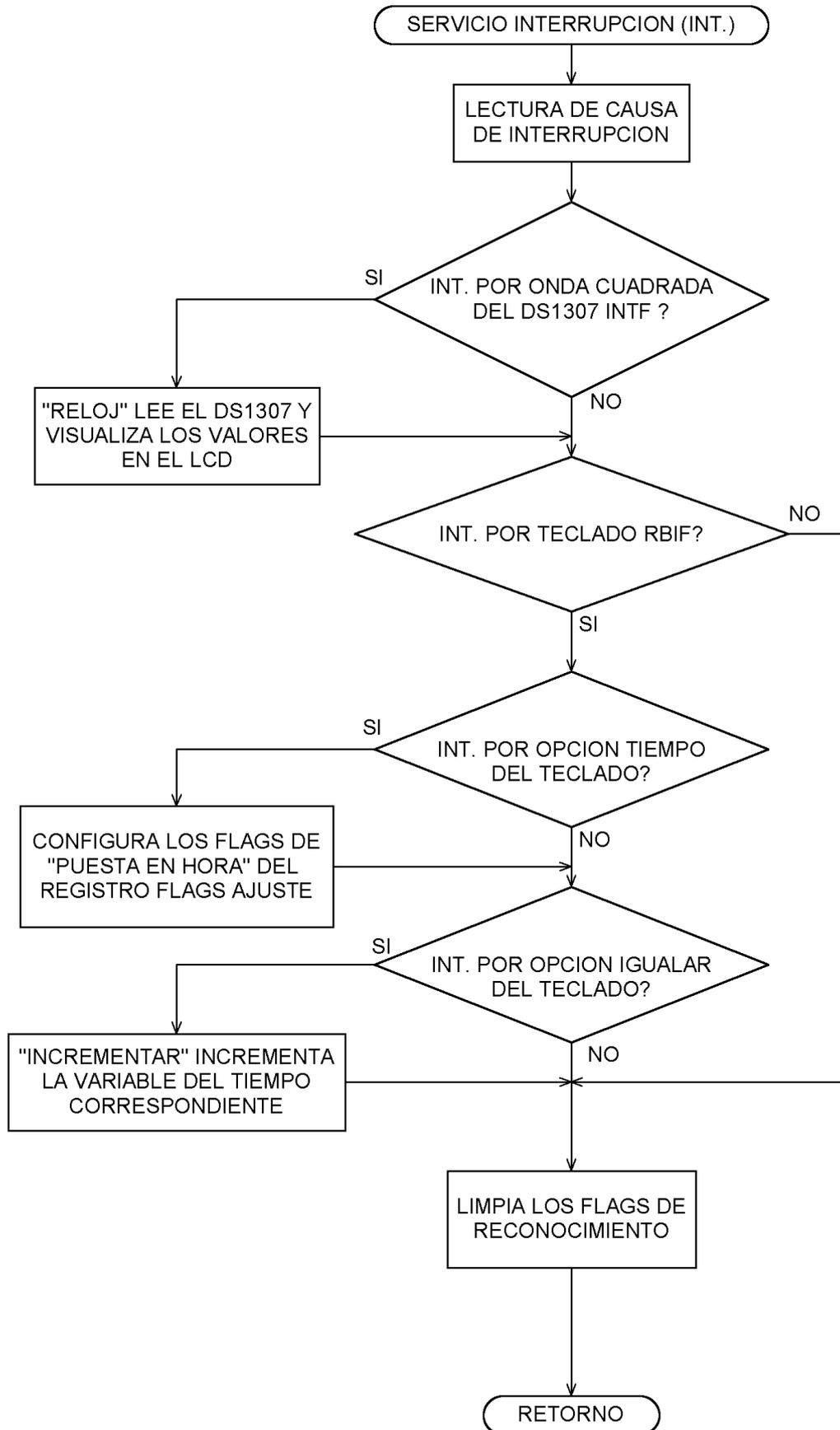
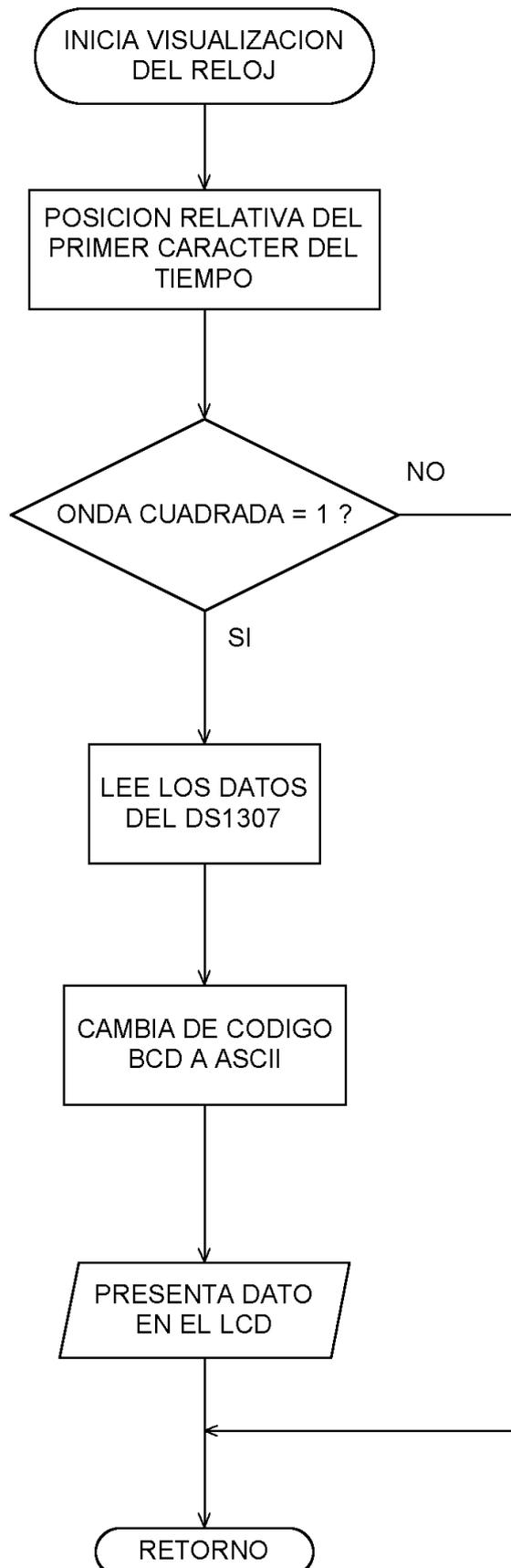


DIAGRAMA DE FLUJO DE LA SUBROUTINA PARA VISUALIZACION DEL RELOJ EN EL LCD



3.2.3 Diagramas de flujo de la comunicación I2C y el modulo MSSP con el esclavo PIC 16F872

DIAGRAMA DE FLUJO DE TRANSFERENCIA DE DATOS AL ESCLAVO (PIC 16F872) EN MODO LECTURA

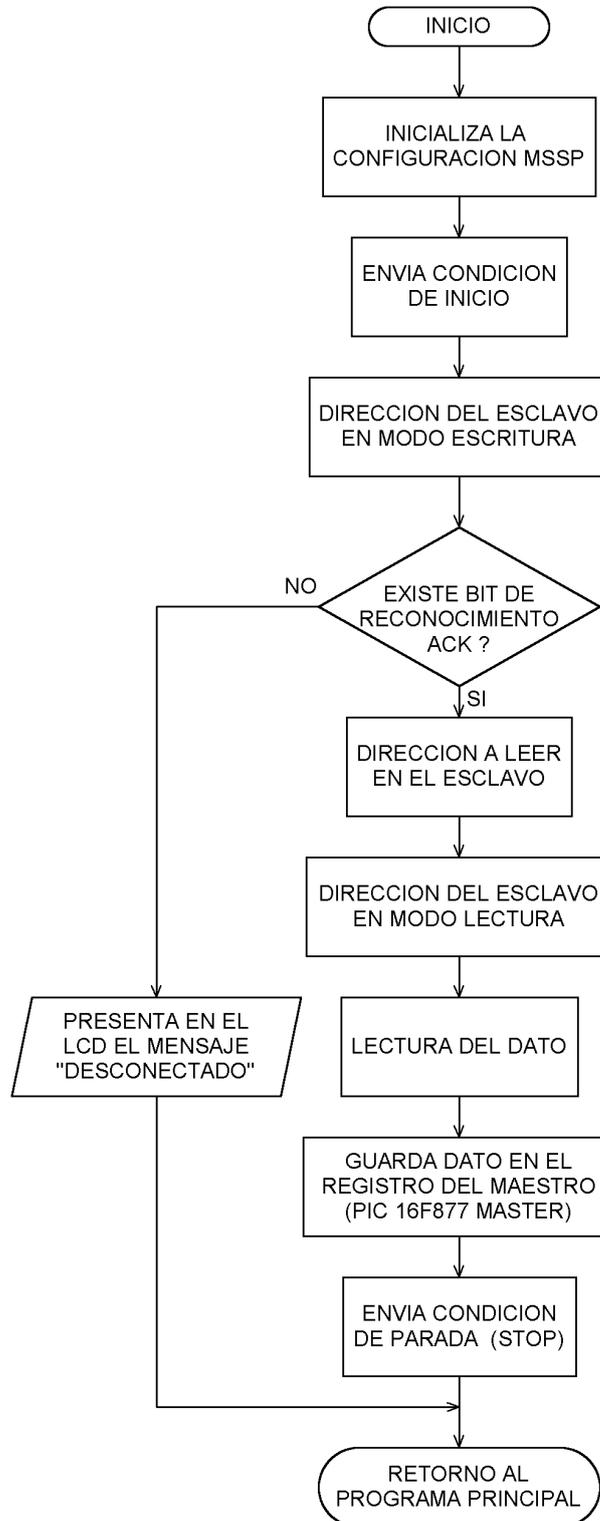
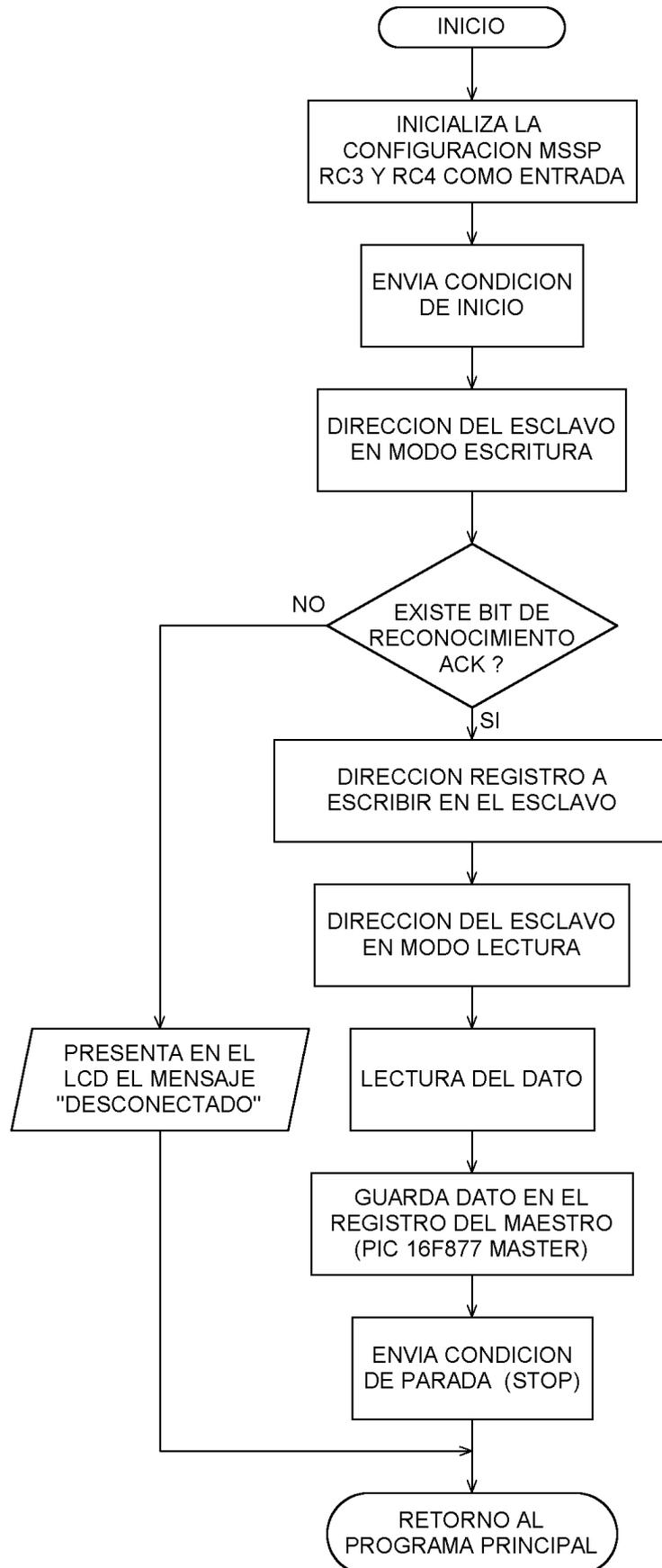


DIAGRAMA DE FLUJO DE TRANSFERENCIA DE DATOS AL ESCLAVO (PIC 16F872) EN MODO ESCRITURA



3.2.4 Programa en lenguaje ensamblador

En este punto del trabajo, insertamos el programa en lenguaje ensamblador, que con la ayuda del MP-LAB (*software* que proporciona Microchip, fabricante de los microcontroladores PIC) se ha desarrollado, simulado en el software y probado en el equipo con el apoyo de un grabador de PIC's conocido como el *PIC STAR PLUS*. Queremos anticipar las disculpas del caso por no incluir impreso en papel dicho programa debido a la gran extensión del mismo (más de 3000 líneas) por lo que nos hemos visto obligados a incluirlo dentro de un disco compacto adjunto a este trabajo en la sección de anexos.

3.3 Pruebas y chequeos, ajustes y comprobación de operación

Una vez acabado el montaje de los componentes en las tarjetas respectivas, se procedió a conectar a la red de suministro y verificar los niveles de tensión que entrega la fuente de alimentación, se midieron + 9 voltios a su salida que es lo correcto. A continuación, se conectó a la tarjeta principal, se midió la tensión presente a la salida del regulador IC7805 que es igual a + 5 voltios.

Después, verificamos las tensiones de alimentación en los zócalos del DS1307, PIC 16F877, LCD, en los pines respectivos; también se midieron los + 5 voltios. Al terminar esta verificación, el led de encendido debe estar iluminado indicando que el circuito se encuentra energizado, si no se ilumina, hay razones para pensar que hay algo mal conectado, se deberá entonces comprobar si el led esta bien polarizado y si las conexiones entre la fuente y la tarjeta están correctas. Cuando se encienda el led, se procede a desconectar la alimentación y colocar el PIC16F877 en su zócalo. Una vez conectada la alimentación de nuevo, habrá una señal en alto (+5VCC) en el pin 1 MCLR. Los terminales RB4, RB5, RB6, RB7 pasan a nivel bajo durante un tiempo determinado y luego a nivel alto, si los niveles continúan sucediéndose quiere decir que todo esta bien de momento. Si esto no ocurre hay que revisar el zócalo del PIC16F877 y comprobar las conexiones de todas las patillas y del cristal de cuarzo.

A continuación se monta el teclado y el LCD, al conectar la alimentación se ilumina y se inicializa el LCD luego se presentan los mensajes de inicio, si en el *display* no aparece nada, se deberán comprobar las conexiones entre el circuito principal y el LCD, asegurándose de la presencia de los +5 voltios. Si todo está bien hasta aquí, hay que cortar nuevamente la alimentación e insertar el circuito integrado DS1307, luego volver a conectar la alimentación.

El PIC16F877 se encargará de inicializar el DS1307 y configurarlo para que genere una señal cuadrada con una frecuencia de 1 Hz de modo que el microcontrolador sea interrumpido una vez por segundo. La salida de esta línea esta en drenaje abierto siendo necesaria una resistencia de *pull-up* para su correcto funcionamiento. Se observa ahora que el diodo verde parpadea al ritmo de los segundos. Ahora pulsando las teclas deben presentarse los datos y mensajes en el LCD que corresponden a cada tecla. Seguidamente se conecta el esclavo con la tarjeta del PIC16F872, previa desenergización de la tarjeta maestro, se energiza y se revisa la tensión en la tarjeta del esclavo, se introducen las señales de unidades producidas y maquina encendida en el esclavo y se verifica en la tarjeta principal cuando se presenta en el LCD, también se puede revisar los demás datos de la maquina, si en el LCD se presenta el mensaje desconectado hay que revisar el cable de comunicación.

3.4 Ensamblaje del equipo completo

Antes de introducir la tarjeta en la caja, hubo que taladrar los orificios del LCD, teclado, conectores de comunicación y alimentación.

La sujeción de la tarjeta principal se la hizo con la ayuda de tornillos milimétricos, los orificios de los conectores se taladraron con un diámetro ligeramente inferior, agrandándoles luego con la lima. A continuación se colocó el panel frontal y se montaron los diferentes elementos externos, por ser un equipo sencillo no necesitó de muchas calibraciones.



Figura 94. Equipo Ensamblado

CAPITULO IV

MANUAL DEL USUARIO

4.1 Instrucciones de conexión, puesta en marcha y operación

Quizás hablar de instrucciones de operación para este equipo sea un poco exagerado ya que es muy sencillo de manejarlo. Veamos tan solo una pequeña síntesis sobre algunos de los controles que tiene. El menú de nuestro equipo se ha reproducido en la figura siguiente y es la base con la que siempre trabajaremos. En este caso la maquina a revisar es la que se encuentra en el numero uno.

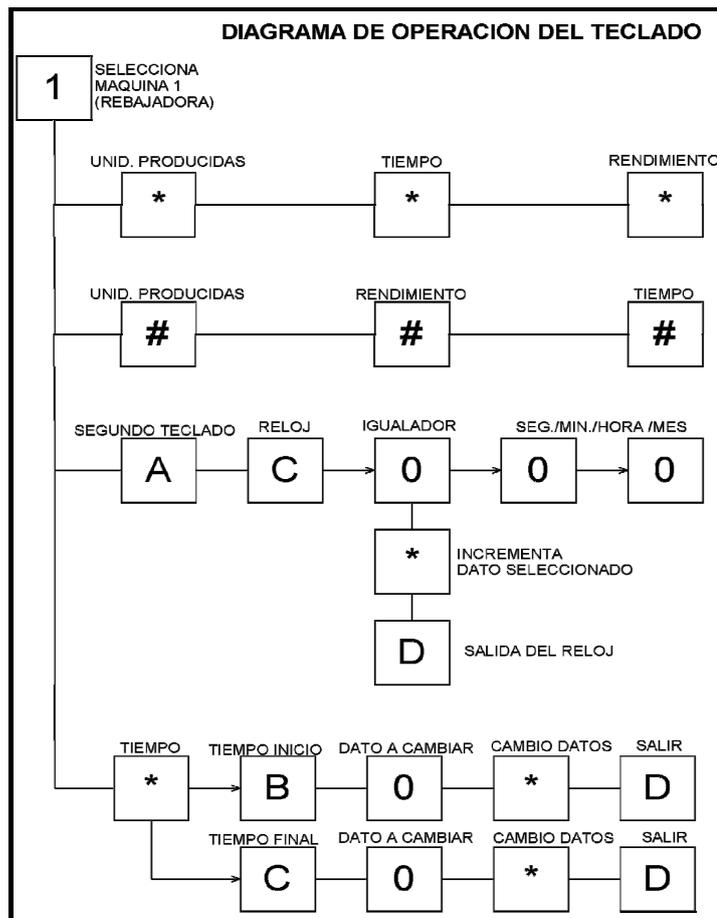


Figura 95. Diagrama de Operación del Teclado

- 1.- Cuando se conecta el equipo, se selecciona la Máquina Rebajadora pulsando el No. 1 del teclado.
- 2.- Para revisar los demás datos de esta máquina como unidades producidas, rendimiento y tiempo total de operación, se presiona las teclas * y/o #.
- 3.- Para revisar los datos del reloj se debe presionar la tecla **A** y luego **C**, en el LCD se presenta los datos del reloj.
- 4.- Si se requiere igualar el reloj se presiona la secuencia **A, C, 0**, cuando se presiona **0** se va seleccionando qué dato se desea incrementar mes, día, hora, minuto, segundo, y con las teclas * y # se incrementa los registros y se graban automáticamente en el DS1307.
- 5.- Si se requiere revisar otra máquina se presiona otra tecla **2...9** (por expandir).
- 6.- Si se desea introducir la fecha en que se analizará el rendimiento, se ubica el menú primero en la máquina deseada, para el ejemplo será el No. 1 que corresponde a la máquina rebajadora (numeral 1), luego siguiendo las instrucciones del numeral 2, nos ubicamos en el modo tiempo y luego se presiona la tecla **B** o **C**, para incrementar hay que presionar *.
- 7.- Para salir de todos estos datos se presiona la tecla **D**.

4.2 Mantenimiento

Como la mayoría de equipo electrónico de hoy en día, el buen mantenimiento se rige básicamente al medio en donde dicho equipo vaya a prestar sus servicios, siendo el nuestro para un medio industrial, se debiera tomar muy en cuenta que en el montaje y ubicación del mismo hay que considerar una zona libre, en la medida de lo posible, de polvos y humedad, libre de vibraciones y aislada tanto física como eléctricamente, apantallando la carcasa y conectándola a una tierra independiente y de buena calidad.

Se deberá periódicamente chequear la correcta y ajustada conexión de todos los cables de entrada y salida de señales, ajuste de los chips a los zócalos, se deberán revisar

también calentamientos anormales principalmente en la fuente, chips, resistencias y pistas del circuito impreso y en el caso de encontrar alguna anomalía de este tipo, lo correcto será cambiar el dispositivo y chequear la causa del calentamiento como sobre corrientes por sobrecargas o por conexiones flojas o defectuosas.

Ahora bien, es por todos también conocido que los equipos electrónicos tienen una alta confiabilidad por lo que los periodos de disponibilidad en estos equipos son muy grandes y los de mantenibilidad muy cortos ya que se dañan muy poco y cuando esto ocurre, lo más conveniente en la mayoría de los casos, es el reemplazo total.

CAPITULO V

CONCLUSIONES Y BIBLIOGRAFIA

5.1 Conclusiones y Recomendaciones

Dentro de este proceso de diseño, desarrollo y puesta en marcha del Equipo Supervisor, podemos concluir que:

- La investigación y desarrollo de trabajos y prototipos con PIC's, resulta muy conveniente y económico.
- Al poseer dentro de sus características muy pocos limitantes tanto técnicos como de capacidad, el PIC tiene un gran alcance dentro de la inventiva que cualquier desarrollador de aplicaciones pueda tener.
- En este proyecto específicamente se puede concluir que, aunque el desarrollo y puesta en marcha del equipo se demoró, no por razones de falta de funcionalidad ni limitantes técnicos, sino mas bien por el poco tiempo que se pudo destinar a este proyecto, las aplicaciones PIC son rápidas y confiables.
- El protocolo de comunicación I2C tiene una excelente y muy confiable respuesta, inclusive dentro de ambientes industriales, es decir con altos niveles de ruido eléctrico y armónicos.
- Recomendamos que para tener un mejor y mas confiable equipo electrónico digital de diseño y fabricación local, se debe considerar como prioritaria la acción de aislarlo eléctricamente de los probables ruidos de inducción mediante sistemas opto acoplados, transformadores de aislamiento y carcasas aterradas.
- Finalmente se puede concluir y recomendar la aplicación PIC y su protocolo I2C a una gran mayoría de sistemas de automatización y control industrial, por su alto rendimiento, bajo costo, excelente *performance* con un potente *software*, flexibilidad,

fácil programación y operación, alta confiabilidad, buena disponibilidad de herramientas para el desarrollo y bajo mantenimiento.

5.2 Bibliografía

Referencias Bibliográficas:

- NUÑO G, Fernando. Comunicación serie para sistemas basados en microcontroladores PIC módulo SSP (*synchronous serial port*) interfase I2C, ©ATE-Universidad de Oviedo. Primera Edición, Agosto 1999,
- PÉREZ R, Leonel. Módulos sobre PIC'S, recopilación. Curso de graduación de Tecnología Electrónica. Universidad del Azuay, Septiembre 2005.

Referencias Electrónicas:

- OROZCO SALGUERO, Mauricio A. Control de un módulo LCD con Pic16f84a, [en línea], 2003-10-04, [citado: 2004-12-17] Disponible en World Wide Web: <http://www.ilustrados.com/publicaciones/EpyVFEkEEVEBKYSIYj.php>
- CAMPOS CERDA, Manuel. Los microprocesadores Microchip PIC. Curso de Ingeniería de en Automática y Electrónica Industrial. Escuela Superior de Ingenieros [en línea], [España] 2004-05-16, [citado: 2004-12-20]. Disponible en World Wide Web: http://www.gte.us.es/usr/jon/sed_iaei/tema8.pdf
- MICROCHIP, Datasheet PIC16F87X [en línea] 1990-02-12, [citado: 2004-12-25]. Disponible en World Wide Web:
<http://ww1.microchip.com/downloads/en/DeviceDoc/30292c.pdf>
- HISPAVILA. Lecciones de Electrónica - Fuentes de Poder, [en línea], [España] 2000-08-21, [citado 2005-01-05]. Disponible en World Wide Web :
<http://www.hispavila.com/3ds/lecciones/lecc3.htm>
<http://www.hispavila.com/3ds/lecciones/lecc3part2.htm>
<http://www.hispavila.com/3ds/lecciones/lecc3part3.htm>

- MAXIM-IC. Hoja de especificaciones DS1307, [en línea], 2005-07-18, [citado 2005-10-16]. Disponible en World Wide Web:

http://www.maxim-ic.com/quick_view2.cfm/qv_pk/2688

<http://pdfserv.maxim-ic.com/en/ds/DS1307.pdf>

- MARTÍN “IONITRON”, Daniel C. Funcionamiento de Teclados Matriciales y Displays LCD, [en línea], [Barcelona, España], 2001-06-20, [citado 2005-10-12]. Disponible en World Wide Web: <http://www.x-robotics.com/rutinas.htm>

ANEXOS

A continuación anexamos un disco compacto con el programa ensamblador que se cargo dentro del PIC y este documento de texto en formato PDF.