Capítulo 2
Unidad de Control

Actualmente se ha tenido un gran incremento en aplicaciones donde las computadoras digitales son utilizadas para controlar procesos de medición e instrumentación en experimentos complejos de laboratorio. Muchas de las cualidades de una computadora de escritorio, como son la facilidad de procesar la información y obtener resultados, pueden ser obtenidos de igual forma por un microcontrolador.

Casi todo producto electrónico a nuestro alcance tiene al menos un microcontrolador en un su interior. Las aplicaciones electrónicas frecuentemente los usan para motores y elementos de control como son las interfaces de operación, los teléfonos celulares no podrían existir sin un microcontrolador; dentro de las aplicaciones que se pueden nombrar en el uso de los microcontroladores, tenemos mandos a distancia, termómetros digitales, controles de acceso por puertas de seguridad, los sistemas ABS o EPS de los coches, control y sensores de maquinaria, automatización del hogar, microrobótica, monederos electrónicos…, etc. La actual sociedad moderna tiene más de 5 billones de microcontroladores que son parte esencial de los productos electrónicos que compramos año con año.

Pero… ¿Qué es un Microcontrolador?

2.1 Microcontrolador

Un microcontrolador es un circuito integrado programable que contiene todos los componentes necesarios para controlar el funcionamiento de una tarea determinada. Un sistema con microcontrolador debe disponer de una memoria donde se almacena el programa que gobierna el funcionamiento del mismo, que una vez programado y configurado, sólo sirve para realizar la tarea asignada.

La utilización de un microcontrolador en el circuito reduce notablemente el tamaño y número de componentes y, en consecuencia, disminuye el número de fallas, el volumen y el peso del equipo.

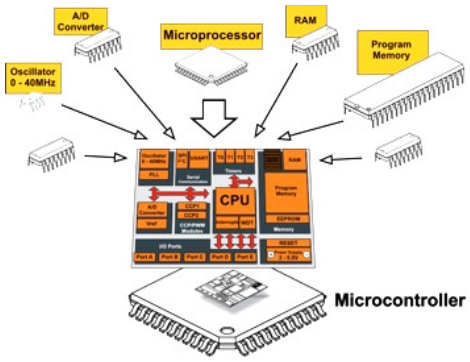


Figura 2.1 “Diagrama a bloques interno de un Microcontrolador.”

Uno de los microcontroladores más populares y fáciles de usar son los de la familia de Microchip Technology conocidos como PIC (por sus siglas en inglés "Peripheral Interface Controller)" con cientos de características diferentes, cada uno se diseña para ser óptimo en diferentes aplicaciones. Estas variaciones consisten en su configuración de memoria, diferentes puertos de I/O "del inglés Input/Output", una amplia gama de recursos dedicados, encapsulados y disposición de periféricos. Este amplio intervalo de opciones para el dispositivo no es único de los microcontroladores PIC, existen muchos otros microcontroladores que pueden tener cualidades similares pero con opciones diferentes para el diseñador. La diversidad tiene el objetivo fundamental de reducir costos.

Los objetivos que se persiguen al usar un microcontrolador para controlar el proceso de electroporación incluyen:

- ✓ Eficiencia en la operación.
- ✓ Facilidad de operación.
- ✓ Seguridad.
- ✓ Producto de mayor calidad.
- ✓ Reducción del tiempo de ejecución.
- ✓ Control secuencial.

2.2 Familia de Microcontroladores

Actualmente existen muchos tipos de microcontroladores, hechos por numerosos y diferentes fabricantes. Los fabricantes construyen las familias de microcontroladores alrededor de un microprocesador (µP), el que está formado por una Unidad de Procesamiento Central (CPU), una Unidad Lógica Aritmética (ALU) y dispositivos de Entrada/Salida (E/S). Los diferentes miembros de una familia son creados usando el mismo CPU, un CPU se combina con diferentes tipos de periféricos y tamaños de memoria creando así diferentes tipos de familia del microcontrolador. Esto es mostrado simbólicamente en la figura 2.2. Un CPU puede ser de 4 bits, son los más sencillos en todos los aspectos y de muy bajo costo, otro CPU es de 8 bits son los más usados actualmente por su gran diversidad y versatilidad, los CPU de 16 bits con aplicaciones típicas en el procesamiento digital de señales, y los CPU más sofisticados son de 32 bits que son más complejos y costosos con aplicaciones en inteligencia artificial, aplicaciones militares y almacenamiento masivo de datos. Existen cientos de microcontroladores dentro de cada familia cada uno con pequeñas diferencias y capacidades y algunos dirigidos hacia aplicaciones muy específicas.

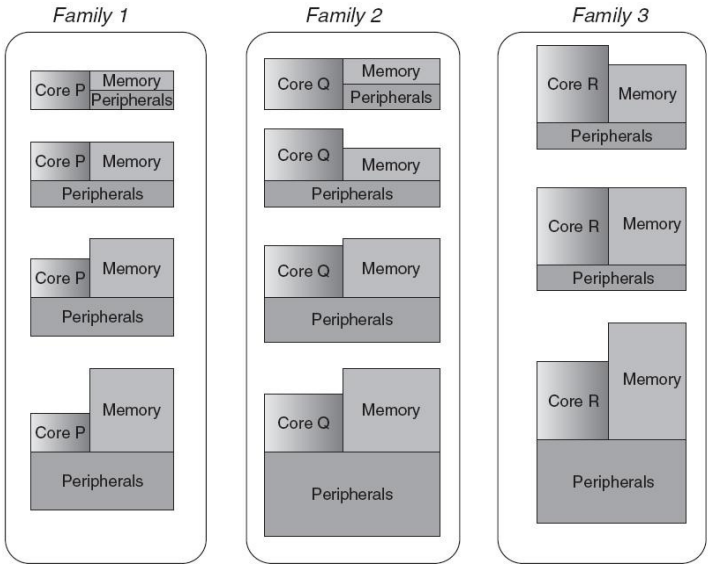


Figura 2.2 "Diferentes tipos de familia de microcontroladores"

2.3 Características del Microcontrolador.

La familia de la serie PIC16 (PIC16-series) son microcontroladores que han estado en el mercado por varios años. A pesar que son excelentes microcontroladores de propósito general, tienen ciertas limitaciones. Por ejemplo, la capacidad de la memoria de programa y de datos son limitadas, la pila es pequeña y la estructura de interrupciones es primitiva, y todas las fuentes de interrupción comparten el mismo vector de interrupción. La serie PIC16 de igual forma no posee el soporte para interfaces avanzadas como el USB, bus CAN, etc. Hacer la interfase con estos dispositivos no es del todo fácil. El conjunto de instrucciones para estos microprocesadores esta limitado, por ejemplo, no hay instrucciones de multiplicación o división, y los saltos son bastante simples, siendo una combinación de instrucciones "skip" y "goto".

De ahí que Microchip Inc. haya desarrollado la serie de microcontroladores PIC18 para uso en aplicaciones complejas. Los microcontroladores PIC18 ofrecen soluciones eficiencia-costo para aplicaciones de propósito general escritas en C que usa un sistema operativo en tiempo real (RTOS) y requiere un complejo protocolo de comunicación con la pila, así como: TCP/IP, CAN, USB, o ZigBee. La serie PIC18F proveen una memoria de programa flash de un tamaño que va desde 8 hasta 128 kbytes y una memoria de programa desde los 256 hasta 4 kbytes, con un intervalo de operación desde los 2.0 V hasta los 5.0V, y velocidades desde 0 Hz hasta 40MHz.

Por éstas razones, el desarrollo del sistema de control para el sistema de electroporación se centra en la utilización del microcontrolador PIC 18F452. El cual se describirá para entender mejor la arquitectura de software que se implantará en él.

Las características básicas de PIC18F serie de microcontroladores, son:

- ✓ 77 instrucciones.
- ✓ Compatibilidad de código con PIC16.
- ✓ Memoria del programa mayor a 2 Mbytes.
- ✓ Memoria de datos mayor a 4 kbytes.

- ✓ Operación desde 0 Hz a 40 MHz.
- ✓ Multiplicadores de hardware de 8x8.
- ✓ Niveles de prioridad en interrupción.
- ✓ Instrucciones con ancho de palabra de 16-bit, datos con ancho de palabra de 8-bits.
- ✓ Hasta dos temporizadores/contadores de 8-bits.
- ✓ Hasta tres temporizadores/contadores de 16-bits.
- ✓ Hasta cuatro interrupciones externas.
- ✓ Terminales con suministro de corriente de 25 mA.
- ✓ Hasta cinco módulos de captura/compare/PWM.
- ✓ Master módulo de puerto serie síncrono (SPI y I^2C modos).
- ✓ Hasta dos módulos USART.
- ✓ Puerto esclavo paralelo (PSP).
- ✓ Convertidor analógico-digital de 10-bits.
- ✓ Módulo de Programable de detección de bajo voltaje (LVD).
- ✓ Power-on reset (POR), el encendido del temporizador (PWRT), y empezar a oscilador-up Temporizador (OST).
- ✓ Con temporizador, perro guardián (Watchdog , WDT) con el oscilador RC.
- ✓ Programación In-circuit

Además, algunos microcontroladores de la serie PIC18F ofrecen las siguientes características especiales:

- ✓ Interfase directa bus CAN 2.0
- ✓ Interfase directa bus USB 2.0
- ✓ Interfase TCP/IP
- ✓ Interfase ZigBee

*2.3.1 **Arquitectura Interna.***

Al igual que los demás miembros de su familia, el PIC18F452 se caracteriza por:

- ✓ Tener una arquitectura Harvard.
- ✓ Su procesador es segmentado ó Pipeline.
- ✓ Su procesador es tipo RISC.
- ✓ El formato de las instrucciones es ortogonal.
- ✓ La arquitectura está basada en banco de registros.

*2.3.2 **Arquitectura Harvard.***

Esta arquitectura dispone de dos memorias independientes a las que se conecta mediante dos grupos de buses separados.

- ✓ Memoria de datos
- ✓ Memoria de programa.

Ambos buses son totalmente independientes y pueden ser de distintos anchos, esto permite que la ALU pueda acceder de forma independiente y simultánea a la memoria de datos y la de instrucciones, consiguiendo que las instrucciones se ejecuten en menos ciclos de reloj.

Esta distribución de la memoria permite la adecuación del tamaño de las palabras y los buses a los requerimientos específicos de las instrucciones y los datos. Se puede concluir que las principales ventajas de esta arquitectura son.

El tamaño de las instrucciones no esta relacionado con el de los datos y por lo tanto, puede ser optimizado para que cualquier instrucción ocupe una sola posición de memoria de programa. Así se logra una mayor velocidad y una menor longitud de programa. El tiempo de acceso a las instrucciones puede superponerse con el de los datos, logrando una mayor velocidad de operación.

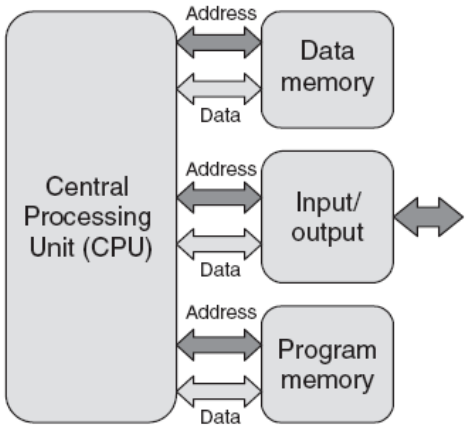


Figura 2.3 "Arquitectura Harvard"

2.4 Características del Microcontrolador PIC18F452

Dentro del circuito integrado que define al microcontrolador existen los recursos y prestaciones limitadas que lo definen.

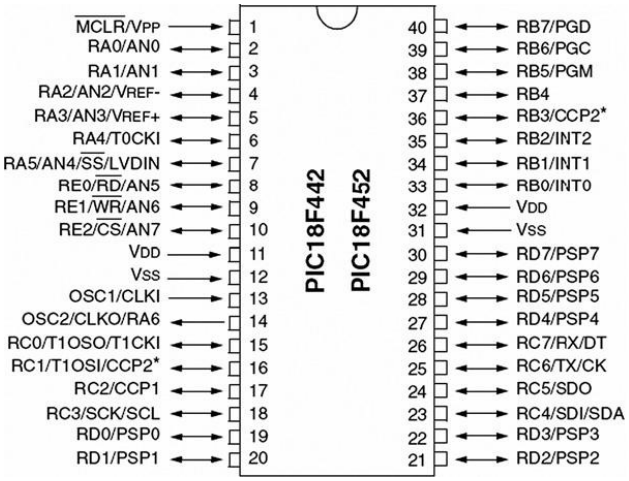


Figura 2.4 "Configuración de Terminales encapsulado DIP microcontrolador PIC 18F452"

Unidad de proceso:

Procesador.	
Memoria de programa.	32 kbytes
Memoria de datos	163864 bytes
Puertos de E/S	(A, B, C, D)

Periféricos Complementarios:

Temporizadores	(4)
Conversores A/D	10 bits, 8 canales de entrada
Puertos de comunicación	MSSP, USART, PSP, I^2C
Módulos de Captura/Comparación/PWM	2
Otros.	

Recursos Auxiliares:

Circuito de reloj.
Modos de bajo consumo.
Perro guardián.
Reset al conectar la alimentación.

Otros:

Tres Terminales para interrupción externa.
Temporizador 0: 8bit /16 bit/ contador con pre-escalador programable a 8bits.
Temporizador 1: 16-bit/contador.
Temporizador 2: 8-bit/ contador con 8-bit registros de periodo (base de tiempo para PWM).
Temporizador 3: 16-bit/contador.
Entrada de reloj por oscilación secundaria- Temporizador1 y Temporizador 3.

Dos módulos de Captura/Comparación/PWM (CCP). Las Terminales del CCP pueden ser configurados como:

1. Entrada de Captura: La captura es de 16-bits máximo, con resolución de 6.25 ns (TCY /16).
2. Comparación de 16-bits máximo resolución de 100ns (TCY).

2 Salida de PWM: el PWM alcanza una resolución de 1-10 bits, máximo
La frecuencia del PWM es

8-bits de resolución = 156 kHz
10-bits de resolución = 39 kHz

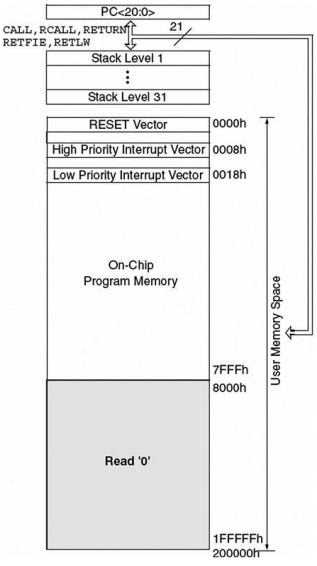
3 Modulo de Comunicación I^2C Maestro/Esclavo

2.4.1. Memoria EEPROM

La EEPROM, (por sus siglas en inglés (Electrically Erasable Programmable Read Only Memory), es una memoria no volátil que se puede ser borrada y reprogramar usando un dispositivo de programación adecuado, las memorias EEPROM son utilizadas para guardar información de configuración, valores de variables, datos de identificación, etc. Algunos microcontroladores tienen incorporados una memoria EEPROM. Por ejemplo, el PIC18F452 contiene una memoria EEPROM de 256 bytes en el que cada byte puede ser programado y borrado directamente por software.

2.4.2. Organización de la memoria de programa (ROM Flash)

El microcontrolador está diseñado, para que en su memoria de programa se almacenen todas las instrucciones del programa de control. En sus 32,768 posiciones (32k) contiene el programa a ejecutar con las instrucciones que gobiernan la aplicación organizadas en palabras de 16 bits. Así pues, la memoria de programa comienza en la posición 0000h (posición inicial de reset) y llega hasta la 7FFFh. Está memoria es no volátil de tal forma que su grabado es de forma permanente. La información contenida en éstas memorias debe ser grabada previamente con un equipo físico denominado programador o grabador.



La figura 2.5 representa el espacio de la memoria de programa de la serie PIC18F, con las direcciones más importantes junto al Contador de Programa de 21 bits con la capacidad de direccionamiento de 2 Mbytes de espacio de memoria.

Las direcciones 0008H y 0018H son reservadas para vectores de interrupción de alta y baja prioridad respectivamente y es donde se debe de inicializar las rutinas de servicio de interrupción según sea el caso.

Figura 2.5 "Mapa de la memoria de programa del PIC 18F452"

2.4.3. Organización de la memoria RAM (Random Access Memory)

La memoria RAM o simplemente memoria (interna o principal) se utiliza para almacenar variables y datos; estos datos varían continuamente, por lo que esta memoria deber ser de lectura y escritura (volátil). La memoria RAM consta de un conjunto de celdas de memoria denominadas también palabras, el número de celdas de memoria suele ser de (1k, 2k, 4k, 8k, 16k, etc). Cada celda de memoria consta de un cierto número de bits. La unidad elemental de memoria se llama byte y está formado por un conjunto de unidades más pequeñas de almacenamiento denominadas bits, que son dígitos binarios (0 ó 1), por definición, se acepta que un byte contiene ocho bits.

Cada celda o byte tiene asociada una única dirección que indica su posición relativa en memoria y mediante la cual se puede acceder a la posición para almacenar o recuperar información. La información almacenada en una posición de memoria es su contenido. El contenido de estas direcciones o posiciones de memoria se llaman palabras.

En la memoria RAM se almacenan:

- Los datos enviados para procesarse desde los dispositivos de entrada.
- Los programas que realizaran los procesos.
- Los resultados obtenidos preparados para enviarse a un dispositivo de salida.

El mapa de la memoria RAM del PIC 18F452 se muestra en la figura 2.6 el bus de direcciones de la memoria de datos es de 12 bits con la capacidad de direccionamiento de 4 Mbytes. Un problema con cualquier espacio de memoria es el gran tamaño de direccionamiento del bus que debe tener. Una forma de evitar grandes buses de direcciones es dividir la memoria dentro de pequeños bloques llamados bancos de memoria cada uno de tamaño idéntico. Así un bus de direccionamiento de memoria más pequeño puede ser usado. La memoria en general del PIC18F452 consiste en 16 bancos, cada uno de 256 bits, donde solo 6 bancos son usados. El intercambio entre bancos ocurre automáticamente cuando se usan compiladores de alto nivel, de esta forma el programador no se debe preocupar por la selección del banco de memoria durante la programación.

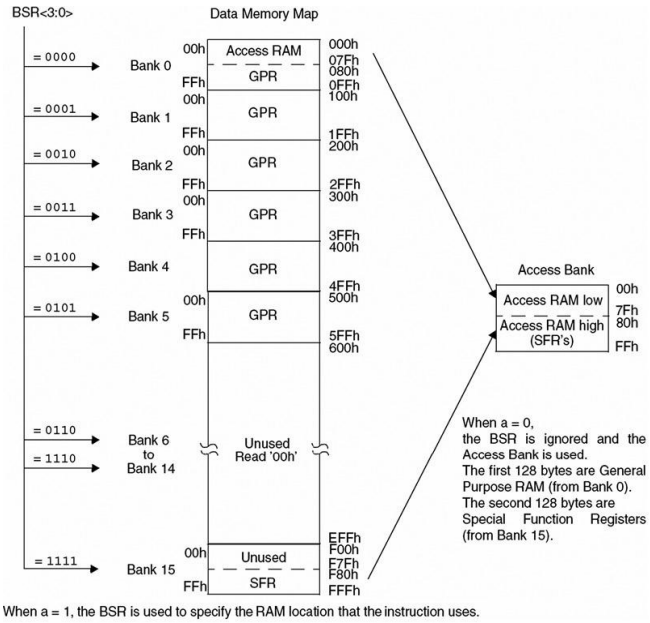


Figura 2.6 "Mapa de la memoria de datos del PIC 18F452"

La memoria de datos tiene posiciones implementadas en RAM y otras en EEPROM. En la sección RAM, se alojan los registros operativos fundamentales en el funcionamiento del procesador y en el manejo de todos sus periféricos, además de registros que el programador puede usar para información de trabajo propia de la aplicación. La RAM estática consta de 4 bancos con 128 bytes cada uno, en las posiciones iniciales de cada banco se ubican los registros de funciones especiales (SFR) y propósito general (GPR) que gobiernan al procesador y sus recursos.

Los registros del SFR están agrupados entre las direcciones ooh a FFh, mientras que el banco de registro de propósito general está formado por 68 posiciones de memoria, ya que sólo son operativos los del Banco 15 (direcciones desde la F8oh hasta la FFFh).

Address	Name	Address	Name	Address	Name	Address	Name
FFFh	TOSU	FDFh	INDF2(3)	FBFh	CCPR1H	F9Fh	IPR1
FFEh	TOSH	FDEh	POSTINC2(3)	FBEh	CCPR1L	F9Eh	PIR1
FFDh	TOSL	FDDh	POSTDEC2(3)	FBDh	CCP1CON	F9Dh	PIE1
FFCh	STKPTR	FDCh	PREINC2(3)	FBCh	CCPR2H	F9Ch	—
FFBh	PCLATU	FDBh	PLUSW2(3)	FBBh	CCPR2L	F9Bh	—
FFAh	PCLATH	FDAh	FSR2H	FBAh	CCP2CON	F9Ah	—
FF9h	PCL	FD9h	FSR2L	FB9h	—	F99h	—
FF8h	TBLPTRU	FD8h	STATUS	FB8h	—	F98h	—
FF7h	TBLPTRH	FD7h	TMR0H	FB7h	—	F97h	—
FF6h	TBLPTRL	FD6h	TMR0L	FB6h	—	F96h	TRISE(2)
FF5h	TABLAT	FD5h	T0CON	FB5h	—	F95h	TRISD(2)
FF4h	PRODH	FD4h	—	FB4h	—	F94h	TRISC
FF3h	PRODL	FD3h	OSCCON	FB3h	TMR3H	F93h	TRISB
FF2h	INTCON	FD2h	LVDCON	FB2h	TMR3L	F92h	TRISA
FF1h	INTCON2	FD1h	WDTCON	FB1h	T3CON	F91h	—
FF0h	INTCON3	FD0h	RCON	FB0h	—	F90h	—
FEFh	INDF0(3)	FCFh	TMR1H	FAFh	SPBRG	F8Fh	—
FEEh	POSTINC0(3)	FCEh	TMR1L	FAEh	RCREG	F8Eh	—
FEDh	POSTDEC0(3)	FCDh	T1CON	FADh	TXREG	F8Dh	LATE(2)
FECh	PREINC0(3)	FCCh	TMR2	FACh	TXSTA	F8Ch	LATD(2)
FEBh	PLUSW0(3)	FCBh	PR2	FABh	RCSTA	F8Bh	LATC
FEAh	FSR0H	FCAh	T2CON	FAAh	—	F8Ah	LATB
FE9h	FSR0L	FC9h	SSPBUF	FA9h	EEADR	F89h	LATA
FE8h	WREG	FC8h	SSPADD	FA8h	EEDATA	F88h	—
FE7h	INDF1(3)	FC7h	SSPSTAT	FA7h	EECON2	F87h	—
FE6h	POSTINC1(3)	FC6h	SSPCON1	FA6h	EECON1	F86h	—
FE5h	POSTDEC1(3)	FC5h	SSPCON2	FA5h	—	F85h	—
FE4h	PREINC1(3)	FC4h	ADRESH	FA4h	—	F84h	PORTE(2)
FE3h	PLUSW1(3)	FC3h	ADRESL	FA3h	—	F83h	PORTD(2)
FE2h	FSR1H	FC2h	ADCON0	FA2h	IPR2	F82h	PORTC
FE1h	FSR1L	FC1h	ADCON1	FA1h	PIR2	F81h	PORTB
FE0h	BSR	FC0h	—	FA0h	PIE2	F80h	PORTA

Figura 2.7 "Registros SFR del PIC18F452"

Explicar a detalle el funcionamiento del CPU esta lejos del alcance de este proyecto, lo que es importante decir, es que: el CPU esta hecho con tecnología RISC y que la reducción del set de instrucciones le da a los microcontroladores PIC dos grandes ventajas. Además el CPU puede reconocer y ejecutar 35 instrucciones simples.

El tiempo de ejecución es casi el mismo para todas las instrucciones, la única excepción son los saltos y las ramas de la instrucción cuyo tiempo de ejecución es el doble de largo. Esto hace que la velocidad de operación del microcontrolador sea de 20 MHZ, el tiempo de ejecución será de 20 ns, el programa se ejecutará a una velocidad de 5 millones de instrucciones por segundo.

La ALU (por sus siglas en inglés "Arithmetic Logic Unit") realiza operaciones aritméticas y lógicas, tales como suma, resta, multiplicación, división y comparaciones. La ALU del PIC18F452 es de 8 bits y también el registro de trabajo W, del que normalmente recibe un operando que puede ser cualquier registro, memoria, puerto de entrada/salida o el propio código de instrucción.

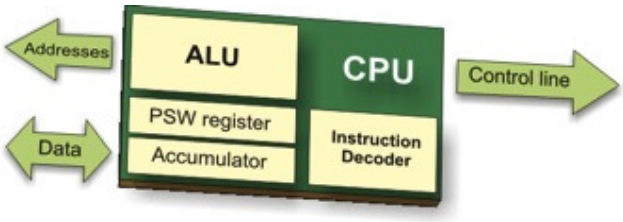


Figura 2.8"Diagrama a bloques del μC"

Existen cinco puertos para la comunicación con el mundo exterior:

PORTA de 7 bits <RA6:RA0>
PORTB de 8 bits <RB7:RB0>
PORTC de 8 bits <RC7:RC0>
PORTD de 8 bits <RD7:RD0>
PORTE de 3 bits <RE2:RE0>

Estos puertos de entrada, sirven para introducir datos (información) al microcontrolador para su proceso. Los datos se leen de los puertos de comunicación y se almacenan en la memoria central o interna (Flash). Los puertos de entrada convierten la información de entrada en señales eléctricas que se almacenan en la memoria central. Los puertos de salida permiten representar los resultados (salida) del proceso de los datos

El contador de programa coordina las actividades del microcontrolador y determina cuales operaciones se deben realizar y en qué orden, además, controla y sincroniza todo el proceso de la μC. Cuenta con 21 bits, lo que en teoría permitiría direccionar 2 Mbytes de palabras de memoria.

2.4.4. Interrupciones

Las interrupciones son un importante concepto de lo que es un microcontrolador. Una interrupción provoca que el microcontrolador responda a eventos tanto externos como internos. Cuando una interrupción ocurre, el μC abandona el programa o rutina que se ejecuta y salta a la subrutina de programa conocida como ISR (por sus siglas en inglés Interrupt Service Routine). El código dentro de la rutina ISR se ejecuta, al finalizar su ejecución la rutina de programa ISR, regresa el control del programa a la rutina interrumpida reanudando así el flujo normal del programa.

El ISR comienza desde una localidad de memoria de programa conocida normalmente como dirección del vector de interrupción. Algunos μC con múltiples interrupciones tienen tan solo una dirección de vector de interrupción, mientras que otros tienen diferentes vectores de interrupción, una dirección para cada fuente de interrupción. Una característica importante de un sistema con múltiples interrupciones es que para diferentes fuentes de interrupción se pueden asignar diferentes niveles de prioridad.

2.4.5. Convertidor Analógico Digital

El convertidor Analógico-Digital (A/D) es usado para convertir señales analógicas como puede ser un voltaje, la voz, señales de salida de sensores electrónicos, así como muchas de las señales que viajan a través de un medio guiado como un cable, o no guiado como es el aire que son de tipo continuo y pueden tomar valores infinitos a lo largo del tiempo. Con el fin de adaptar las señales a circuitos digitales para su lectura, almacenamiento o procesamiento de un µC es necesario digitalizar la señal analógica acotándola dentro un intervalo de tiempo (muestreo) y con valores de voltaje (niveles de cuantización). El muestreo implica que tenemos que tomar una muestra de la señal cada T segundos ya que no hay memoria suficiente capaz de almacenar los puntos infinitos de una señal en cualquier intervalo de tiempo.

La cuantificación implica adquirir una muestra de la amplitud de la señal redondeándola a valores fijos de amplitud. Estos valores van a depender del número de bits que se vayan a almacenar para cada muestra. Los convertidos A/D generalmente son de 8 a 10 bits teniendo niveles de cuantización desde los 256 hasta los 1024 niveles. Muchos de los µC PIC tienen múltiples entradas para una conversión A/D. Los convertidores A/D son inicializados desde el programa del usuario y le toma algunos microsegundos en completar la conversión. Los convertidores A/D generalmente originan interrupciones una vez que la conversión de la señal es completada.

2.4.6. Temporizadores

Los temporizadores (Timers) son parte importante de un microcontrolador. Un temporizador es básicamente un contador, el cual recibe impulsos de reloj ya sea de una fuente externa o del oscilador interno del propio µC. Un temporizador puede tener una longitud de 8 o 16 bits, la información puede ser cargada por el programa de control hacia el temporizador, el cual pude iniciar o detener su funcionamiento. La mayoría de los temporizadores son configurados para generar una interrupción cuando alcanzan cierto valor de cuenta (generalmente cuando el temporizador se desborda). El programa principal puede usar esta interrupción para cumplir con tiempos precisos de alguna operación dentro del microcontrolador. Los microcontroladores PIC18F tienen por lo menos tres temporizadores como en el caso del PIC 18F452. Algunos µC tienen incorporados módulos de captura y comparación, donde el valor del temporizador es leído cuando ocurre un evento externo ó el valor del temporizador es comparado con algún valor preestablecido generando una interrupción cuando la comparación es exitosa. Los PIC18F cuentan por lo menos con 2 módulos de este tipo.

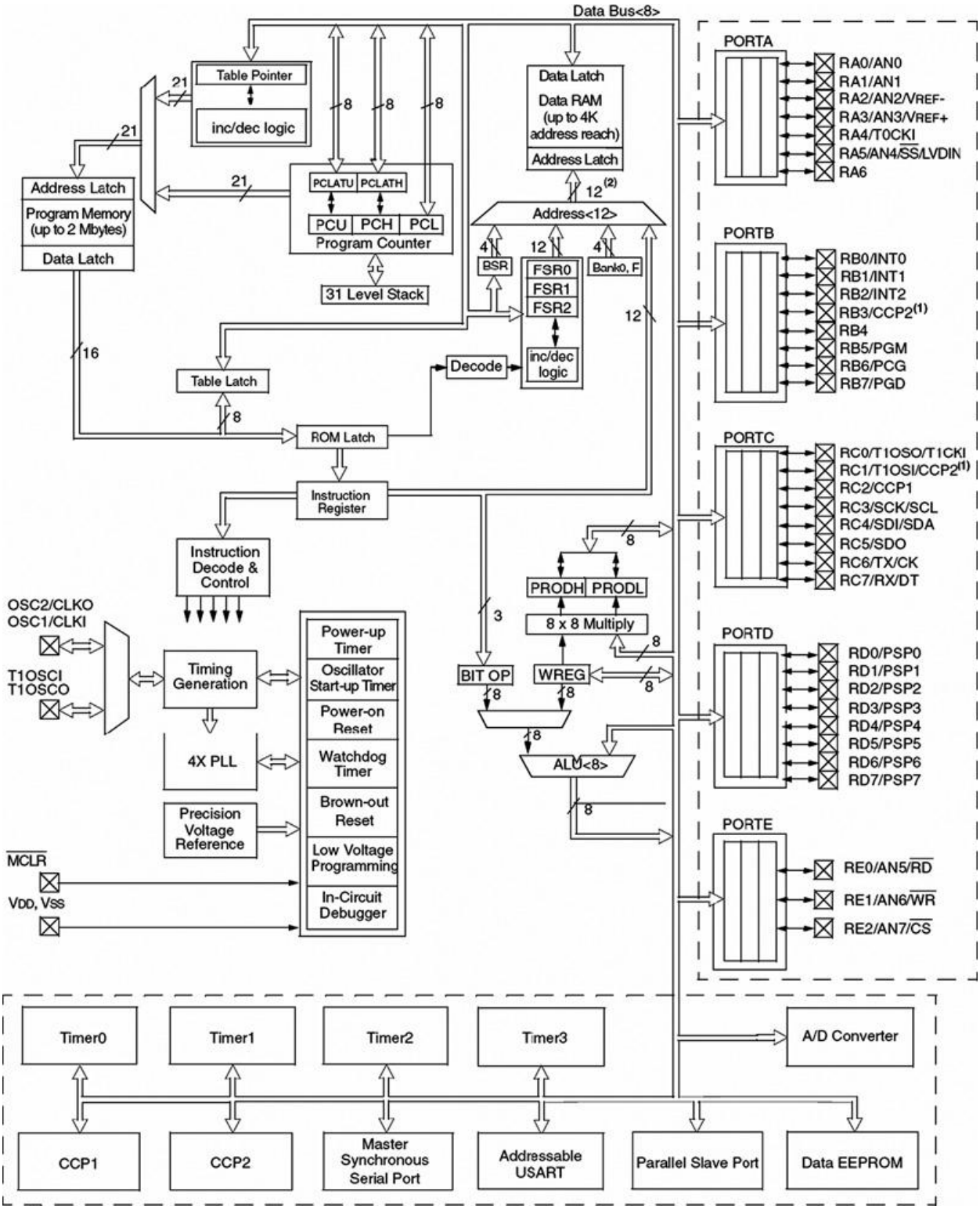


Figura 2.9 “Diagrama de bloques de un microcontrolador PIC 18F452”

2.5 Herramientas de desarrollo de software

Las herramientas de desarrollo de software son programas de computadora, que permiten al programador crear, modificar y probar las aplicaciones de nuevos programas, así mismo, ayudan a los programadores a desarrollar y probar sistemas complejos en un tiempo relativamente corto.

Algunas herramientas comunes de desarrollo de software son:

4 Editores de texto
5 Ensambladores / Compiladores
6 Simuladores
7 Simuladores en lenguaje de alto nivel
8 Entornos integrados de Desarrollo (IDE)

2.5.1. Editores de Texto

Un editor de texto es usado para crear o editar programas y archivos de texto. Windows posee un editor de texto llamado bloc de notas (Notepad). Usando bloc de notas, podemos crear un nuevo archivo de programa, modificar un archivo ya existente, o visualizar o imprimir el contenido de un archivo. La mayoría de los ensambladores y compiladores vienen con editores de texto ya instalados, haciendo posible la creación de programas y después ensamblarlos o compilarlos sin la necesidad de salir del editor.

2.5.2. Ensambladores y compiladores

Los lenguajes ensamblador, generan código maquina ejecutable, que puede ser cargado dentro de la memoria flash del microcontrolador.

Los compiladores, generan código maquina ejecutable desde un lenguaje de alto nivel. Los compiladores más usados para la serie PIC18 son BASIC, C y PASCAL.

Los lenguajes ensamblador son usados en aplicaciones donde la velocidad de procesamiento es crucial y el microcontrolador debe responder a interrupciones internas y externas en un tiempo sumamente pequeño, sin embargo, es difícil desarrollar programas complejos usando código ensamblador.

Los lenguajes de alto nivel, por otra parte, son fáciles de aprender y programas complejos pueden ser desarrollados y probados en un periodo de tiempo sumamente corto.

Existen variedad de compiladores de C disponibles para el desarrollo de microcontroladores PIC, algunos de los más populares son:

CCS C [1]
Hi-Tech C[2]
C18 C [3]
mikroC C [4]
Wiz-C C [5]

Sin embargo, la mayoría de los compiladores C en esencia son lo mismo, cada uno posee sus propias características y modificaciones o se ha agregado algún tipo de ventajas al lenguaje estándar.

2.5.3. Simuladores

Un simulador es un programa de computadora que corre sobre una plataforma PC y es usado para probar el programa de aplicación sin la necesidad del hardware de aplicación. Simula el comportamiento de la aplicación en hardware interpretando las instrucciones del programa creado; provee ayuda en la revisión de correcciones de un algoritmo o un programa y pueden ser corregidos bastantes errores durante la simulación.

Los simuladores pueden visualizar el contenido de registros, memoria de datos y el estatus de los puertos de entrada y salida dentro del microcontrolador. De igual forma el programa pude ser ejecutado en modo paso a paso (single-step), de esta forma la memoria y los registros pueden ser examinados hasta que el programa trabaje perfectamente, el código máquina ejecutable es cargado al hardware de aplicación (microcontrolador) por medio de un programador.

Algunos programas en ensamblador, contienen simuladores como por ejemplo:

MPLAB IDE [3]
Oshon Software PIC18 simulator [6]
Forest Electronics PIC18 assembler [5]

2.5.4. Entornos integrados de Desarrollo (IDE)

Los entornos integrados de Desarrollo (IDE) son herramientas poderosas de programación sobre una plataforma PC, los cuales incluyen todo: editores, ensambladores, compiladores, administradores de proyecto (Linkers), simuladores, etc. Pueden depurar los programas y descargar el código máquina ejecutable hacia el microcontrolador usando un dispositivo de programación. Estos programas poseen entornos gráficos de desarrollo (GUI), donde el programador puede seleccionar varias opciones desde el programa sin la necesidad de salirse de él. Los IDE son útiles cuando se programan microcontroladores.

El entorno de programación usado en ésta tesis es el MPLAB IDE, que es un "paquete" que coordina todas las herramientas en una simple interfase gráfica automática. Por ejemplo, una vez que el código es escrito, puede ser convertido a instrucciones ejecutables y descargar el código en el microcontrolador para verificar su funcionamiento. MPLAB IDE tiene los siguientes componentes internos:

Project Manager: El administrador de proyectos provee integración y comunicación entre el IDE y las herramientas del lenguaje.

Editor: El editor es una herramienta completa de programación que también sirve como una ventana dentro del depurador.

Ensamblador y herramientas de lenguaje: El ensamblador puede ser usado por si solo para ensamblar un archivo simple, o puede ser usado con el linker para construir un proyecto

desde archivos fuente por separado, bibliotecas y objetos recompilados. El linker es el responsable de posicionar el compilador de código dentro de las áreas de memoria del microcontrolador a usar.

Debugger: El depurador de Microchip permite puntos de ruptura del programa (breakpoints), compilación paso a paso, y ventana de visualización de variables (watch window), y todas las características de compilación de los modernos IDE.

Simulación: Existe software de simulación en MPLAB IDE para todos los dispositivos y los µC PIC y dsPIC. Estos simuladores usan el PC para simular las instrucciones del µC y algunas funciones propias de los µC PIC y dsPIC. De manera opcional tenemos emuladores In-circuit y depuradores in-circuit que están disponibles para censar el código mientras se ejecuta en la aplicación de hardware correspondiente.

2.5.5. Componentes adicionales para MPLAB IDE

Algunos componentes opcionales que pueden trabajar en conjuntos con MPLAB IDE son:

- Lenguajes de Compilación: MPLAB C18[7] y MPLAB C30[8] compiladores en C de Microchip proveen una amplia integración con el entorno de MPLAB IDE, optimizando código. En conjunto con los compiladores HI-TECH[2,9], IAR[10], microEngineering Labs[11], CCs[1] y Byte Craft[12], son adjuntados por el administrador de proyecto de MPLAB IDE para compilar el código, esto hace que el código sea automáticamente cargado al µC, compilado, depurado y verificado en un instante.

- Programadores: MPLAB PM3[13], PICSTART Plus[14], PICkit[15] 1, 2 y 3, así como MPLAB ICD 2[16] depurador y MPLAB REAL ICE[17] in-circuit emuladores, pueden programar código dentro de los µC ha utilizar.

- In-Circuit Emuladores: MPLAB REAL ICE y MPLAB ICE 2000 son sistemas emuladores in-circuit para µC PIC y dsPIC. Son conectados a la PC y permiten un completo control sobre las aplicaciones del microcontrolador.

2.6 Ciclo de Desarrollo

El proceso de desarrollo de una aplicación es frecuentemente descrito, como ciclo de desarrollo, rara vez todos los pasos de diseño e implementación pueden ser hechos perfectamente a la primera vez, cada vez más código es escrito, probado y modificado de manera de producir una aplicación que trabaje correctamente. El ciclo de desarrollo empieza escribiendo el programa de aplicación usando un editor de texto, luego el programa es traducido a código máquina con la ayuda de ensambladores o compiladores. Si el programa posee muchos módulos, un organizador de proyectos (linker manager) es usado para combinarlos en una sola aplicación, cualquier error de sintaxis es detectado por el ensamblador o compilador y debe ser corregido antes que se genere el código ejecutable del microcontrolador.

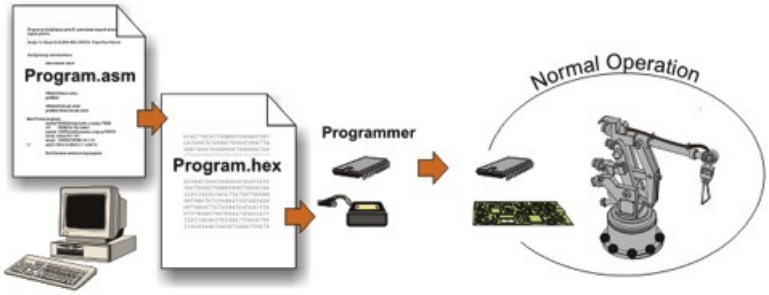


Figura 2.10"Diagrama esquemático del ciclo de desarrollo."

2.7 Bus I²C

El bus I²C (por sus siglas en inglés Inter-Integrated Circuit) es un bus de comunicación serial síncrona desarrollado por Phillips Semicondutors a principios de los años 80´s, con la principal intención de interconectar una cierta cantidad de circuitos integrados dentro de sus diversos productos electrónicos.

En el bus son utilizadas dos líneas de comunicaciones, una para los datos llamada SDA y otra para el reloj SCL. Cada dispositivo que se conecta al bus es direccionable por software, a través de una única e irrepetible dirección dentro del bus. La misma es determinada a través de una combinación de Hardware/Software, donde el fabricante define los bits más significativos.

El número de dispositivos que pueden ser conectados al bus I²C, está limitado por la capacitancia que estos representen al mismo, a un máximo de 400 pF. Las transferencias de información se realizan a través de paquetes de 8 bits bidireccionales, y pueden ser efectuadas a tres velocidades o modos: el modo normal (Standard) a unos 100 kbit/s, modo rápido a 400 kbit/s, o modo de alta velocidad 3.4 Mbit/s

Los dispositivos pueden clasificarse en maestro (master o principal) o esclavo (slave o secundario). El maestro es el que inicia la transferencia de datos y genera la señal de reloj. Cualquiera de los dispositivos seleccionado por un maestro se considera un esclavo. El I²C es un bus multi-maestro; puede haber más de un maestro conectados y controlando el bus.

Tanto las líneas SDA como SCL son líneas bidireccionales que se conectan a $+V_{dd}$ mediante resistores de jalón (pull-up), tal y como se muestra en la figura 2.11.

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

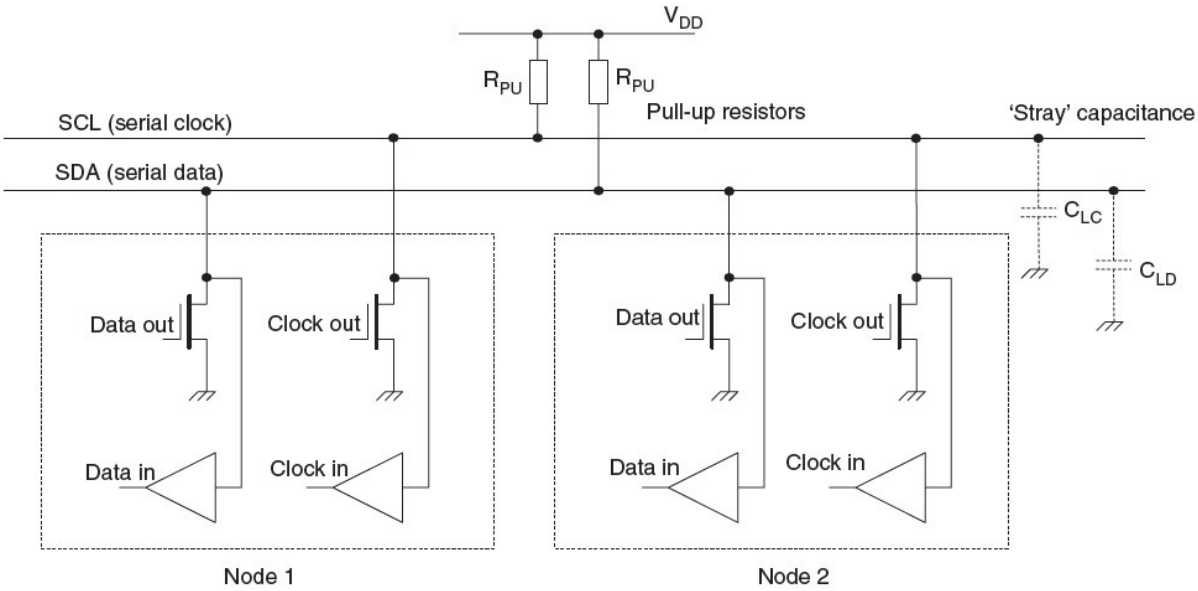


Figura 2.11 "Diagrama de interconexión I²C"

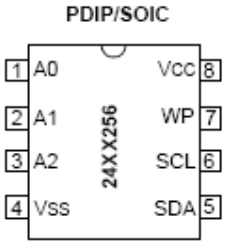
2.8 Memoria EEPROM de datos

Es una pequeña área de memoria de datos de lectura y escritura no volátil, gracias a la cual, un corte del suministro de la alimentación no ocasiona la pérdida de la información, que estará disponible al reinicializarse el programa.

Las memorias EEPROM seriales, poseen las siguientes características:

- ✓ Se pueden conectar fácilmente con microprocesadores o microcontroladores, inclusive algunos de ellos tienen Terminales dedicados para esta labor.
- ✓ Transferencia de datos de manera serial, lo que permite ahorrar Terminales del microcontrolador para dedicarlos a otras funciones.
- ✓ Ocupan la décima parte del espacio de las memorias que trabajan en paralelo, esto permite ahorrar dinero debido al menor tamaño del circuito impreso.
- ✓ El consumo de corriente es mucho menor que en las memorias en paralelo, esto las hace ideales para sistemas portátiles que funcionan con baterías.

Estas memoria usan la comunicación a 2 hilos empleando la interfase I^2C cuyas referencias más conocidas son 24C64/256/512. La velocidad de transferencia de información para estos dispositivos es de 100-400 kHz (aunque el límite lo impone el protocolo I^2C y no la tecnología del dispositivo). Como característica importante de este elemento se tiene la inmunidad al ruido, dado que este integrado tiene filtros en las Terminales de comunicación.



1,2,3	A0, A1, A2	Dirección del dispositivo en el bus.
4	Vss	Tierra
5	SDA	Datos y direcciones seriales I/O
6	SCL	Reloj
7	WP	Protección de escritura. Si está en "0" habilita escritura, en "1" la deshabilita.
8	Vcc	+5V

Figura 2.12 "Encapsulado dip 24Cxxx"

Estas memorias utilizan el bus de 2 hilos para comunicarse con otros dispositivos. Dado que cumplen con el protocolo I^2C, tiene una terminal llamada SCL que recibe los pulsos generados por el dispositivo maestro (microcontrolador) y otro llamado SDA que maneja el flujo de datos de forma bidireccional (entrada/salida). En la figura 2.12 se muestra el diagrama de terminales correspondiente a estas memorias.

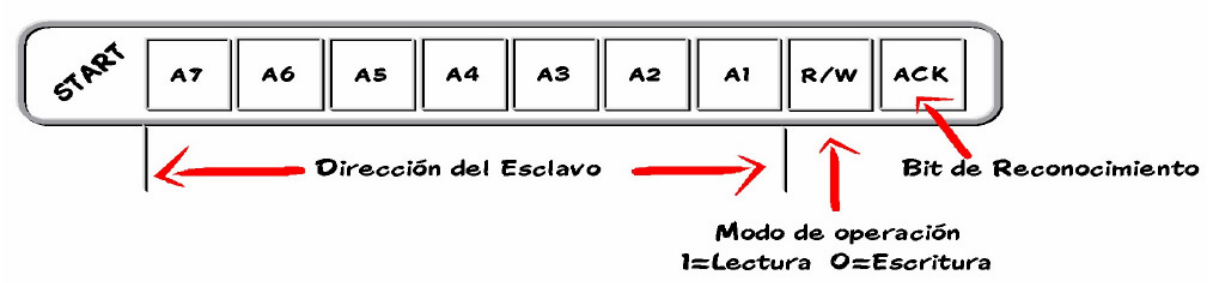
Este dispositivo no requiere de una terminal de habilitación o chip select, ya que en este esquema la transferencia de información solo se puede iniciar cuando el bus este libre. En este caso, como cada dispositivo tiene su dirección determinada mediante las terminales A0, A1 y A2, solamente responderá la memoria cuya dirección coincida con la dirección que va encabezando la trama de información.

2.8.1. Transferencia de la información

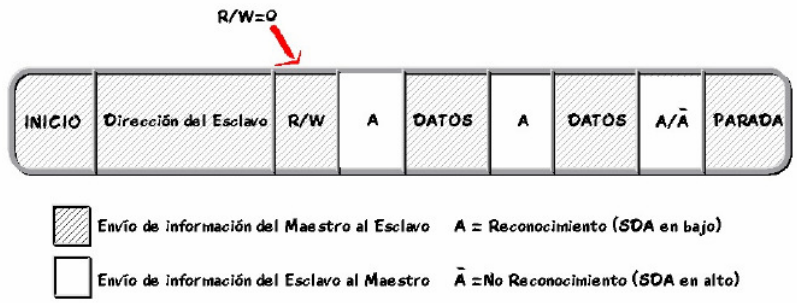
La figura 2.13 muestra la forma del como el microcontrolador entabla comunicación con la memoria, mostrando la serie de bits que debe enviarle a está. La información que el microcontrolador manda a la memoria es la siguiente:

- ✓ Se envía el bit de arranque o start bit
- ✓ El código 1010 (propio de estas memorias).
- ✓ La dirección del dispositivo (A2,A1,A0).
- ✓ Un bit que indica que se desea escribir ("0") en la memoria.

Luego la memoria debe enviar un reconocimiento para informarle al microcontrolador que recibió la información, dicho reconocimiento, llamado ACK, consiste en poner el bus en un nivel bajo (lo hace la memoria). Después, el microcontrolador debe enviar los bits que corresponden a la posición de memoria que se quiere leer o escribir; nuevamente la memoria envía un reconocimiento. El paso siguiente depende de la operación que se vaya a ejecutar.



(a)



(b)

Figura 2.13 (a) "Direccionamiento de dispositivo I^2C con dirección de 7 bits" (b) "Direccionamiento de dispositivo I^2C con dirección de 10 bits"

Si se trata de un proceso de escritura, el microcontrolador solo debe enviar el dato a ser almacenado y esperar el reconocimiento por parte de la memoria para confirmar que llegó correctamente. Si se trata de una lectura, nuevamente se debe repetir los primeros cuatro pasos, solo que en lugar de un "0" que indica escritura, se debe enviar un "1" que indica lectura. Después se espera el reconocimiento y acto seguido se puede leer el byte con el dato que estaba en la posición de memoria que se indicó anteriormente. Cuando se termina la operación, el microcontrolador debe enviar una señal de parada o stop bit. En la figura 2.15 se muestra el diagrama de tiempos correspondiente a todo el proceso descrito anteriormente.

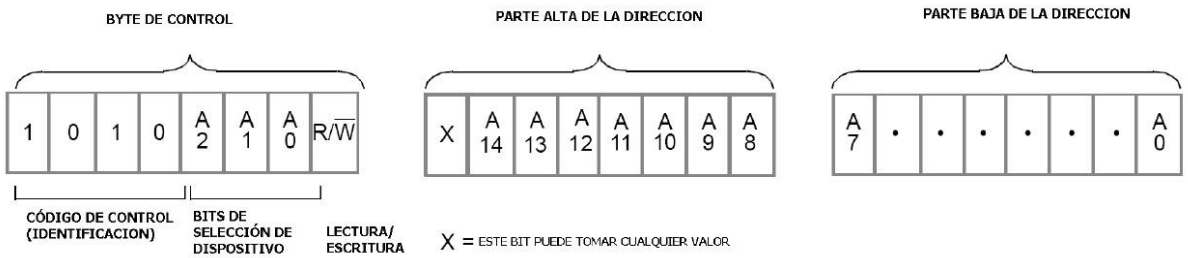


Figura 2.14 "Byte de control y direccionamiento de la memoria 24LC256"

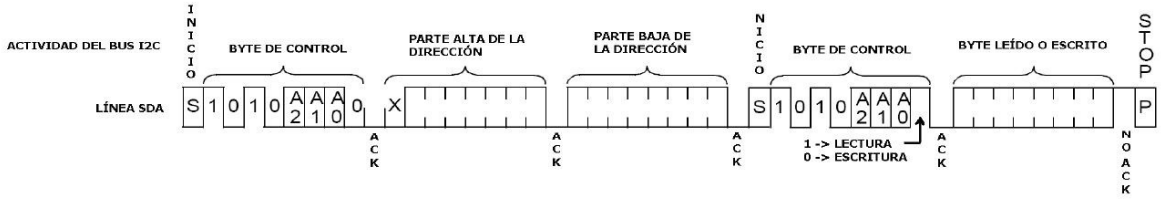
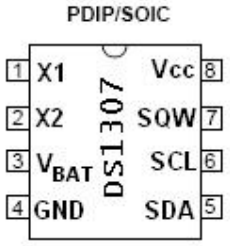


Figura 2.15 "Actividad del bus I^2C durante la lectura/escritura de la memoria 24LC256"

2.9 Reloj-Calendario DS1307

El DS1307 es un reloj en tiempo real RTC (por sus siglas en inglés Real Time Clock) con líneas de conexión a un bus I^2C. Este circuito integrado es un poderoso reloj y calendario de tiempo real, que cumple perfectamente con muchas de las necesidades normales en la adquisición y registro del tiempo. Sus características más destacadas son:

- ✓ Fabricado por Dallas Semiconductors[18] en encapsulado de 8 terminales.
- ✓ El DS1307 es un reloj y calendario de tiempo real que cuenta los segundos, los minutos, las horas, los días de la semana, los días del mes, los meses y los años, válido hasta el año 2100.
- ✓ Almacena los datos en formato BCD para que se pueda trabajar directamente con ellos.
- ✓ Tiene 56 bytes RAM no volátil para almacenamiento de datos.
- ✓ En su terminal SQW/OUT proporciona una onda cuadrada programable.
- ✓ Tiene una circuitería interna de "respaldo" para alimentación en caso de fallo de la alimentación principal, por tanto, es capaz de mantener el tiempo y la fecha actualizados aún cuando el sistema esta apagado.
- ✓ Se puede alimentar entre 4.5 a 5.5V, siendo su valor típico 5V.
- ✓ Posee un bajo consumo de corriente menor a 500 nA en el modo de respaldo. Utiliza un cristal de cuarzo propio de 32.768 Hz para lograr tiempos exactos y no depender del microcontrolador.
- ✓ El último día del mes es automáticamente ajustado a 28, 29, 30 ó 31 días según corresponda, tiene en cuenta los años bisiestos.
- ✓ Puede trabajar en formato europeo de 24 horas o el americano de 12 horas con indicador de AM/PM.
- ✓ Se activa cuando recibe la dirección válida indicada en la figura.



1,2	X_1 X_2	Cristal de cuarzo de 32768 Hz
3	V_{BAT}	Batería de respaldo
4	GND	Tierra
5	SDA	Línea de datos
6	SCL	Línea de reloj
7	SQW/OUT	Señal cuadrada
8	Vcc	+5 V

Figura 2.17 "Encapsulado DIP DS1307"

2.9.1. Conexión del DS1307 al microcontrolador

La figura 2.17 muestra las terminales del reloj DS1307. Las terminales SDA y SDL del DS1307 se conectan a las líneas del µC que conformen el bus I^2C.Los resistores de Jalón tienen el valor característico de 4.7 kΩ.

Si falla la alimentación principal, el DS1307 se alimenta de la batería de respaldo, manteniendo la información del tiempo y la fecha mientras la alimentación principal se mantenga apagada. Si esta batería no se utiliza, los datos de calendario del DS1307 no se

actualizarán en caso de fallo de la alimentación. El valor debe estar comprendido entre los 2 y 3.5V. puede usarse cualquier tipo de batería, aunque el fabricante recomienda una batería de litio de al menos 48 mAh que garantiza una conservación de la información en el DS1307 para más de 10 años.

2.9.2. Registros del DS1307

00h	SEGUNDOS
01h	MINUTOS
02h	HORAS
03h	DIA
04h	FECHA
05h	MES
06h	AÑO
07h	CONTROL
08h	RAM 56X8
.	
.	
.	
3Fh	

En la tabla se muestra el mapa de direcciones de la memoria RAM del DS1307, los registros del calendario se localizan en las direcciones ooh hasta la 07h. Desde la 08h hasta la 3Fh hay 56 posiciones de memoria RAM que pueden ser utilizadas para almacenar datos.

El valor del tiempo y calendario se obtiene mediante la lectura de los registros apropiados. La figura 2.18 muestra los registros del reloj y calendario:

ADDRESS	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0	FUNCTION	RANGE
00h	CH	10 Seconds			Seconds				Seconds	00–59
01h	0	10 Minutes			Minutes				Minutes	00–59
02h	0	12	10 Hour	10 Hour	Hours				Hours	1–12 +AM/PM 00–23
		24	PM/ AM							
03h	0	0	0	0	0	DAY			Day	01–07
04h	0	0	10 Date		Date				Date	01–31
05h	0	0	0	10 Month	Month				Month	01–12
06h	10 Year				Year				Year	00–99
07h	OUT	0	0	SQWE	0	0	RS1	RS0	Control	—
08h–3Fh									RAM 56 x 8	00h–FFh

Figura 2.18 "Registros del DS1307"

El formato de todos los datos está en BCD.
El bit 7 del registro ooh es el bit de puesta en marcha Clock Halt (CH):

- ✓ Si CH=0, pone en marcha el reloj.
- ✓ Si CH=1, impide el funcionamiento del reloj, el cual permanecerá parado.

El formato de las horas puede seguir el modelo americano o el europeo controlado por el bit 6 del registro 02h.

- ✓ Si (Bit 12/24)=0, elige el modo europeo de 24 horas.
- ✓ Si (Bit 12/24)=1, elige el modo americano de 12 horas.

En el modo americano de 12 horas, el bit 5 (A/P) del registro 02h es el bit que determina si la hora es AM (A/P=0) ó PM (A/P=1).

En el modo europeo de 24 horas, el bit 5 es el de mayor peso de las decenas de hora.
El contenido del registro 07h controla la señal cuadrada de la terminal SQW/OUT.

2.9.3. Escritura en el DS1307

La transferencia de datos desde el µC al DS1307 sigue el procedimiento de escritura del maestro al esclavo de un bus I^2C esquematizado en la figura 2.19 .La transferencia de datos se efectúa de la siguiente manera:

- ✓ Primero el µC maestro envía la condición de "Start".
- ✓ Luego envía la dirección del DS1307 (Slave Address) en modo escritura que es la b´11010000´ ó Doh.
- ✓ A continuación el maestro envía un puntero con la primera dirección del registro a escribir (Word Address).
- ✓ Después se transmiten los datos a escribir. La dirección del registro a escribir se incrementa automáticamente.
- ✓ Cuando termina de escribir el µC maestro envía la condición de "Stop".

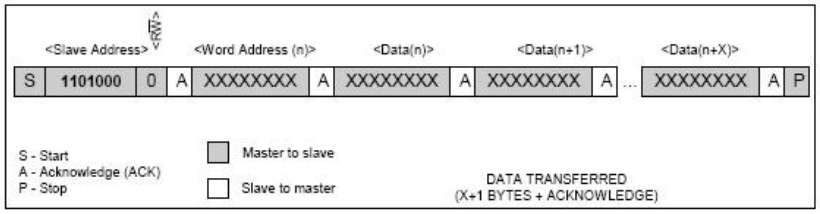


Figura 2.19 "Protocolo de escritura del DS1307"

2.9.4. Lectura del DS1307

La lectura de los datos del DS1307 por parte del microcontrolador sigue el procedimiento de lectura del esclavo por parte del maestro y como se muestra en la Figura 2.20. La transferencia de datos se efectúa de la siguiente manera:

- ✓ Primero el µC maestro envía la condición de "Start".
- ✓ Luego envía la dirección del DS1307 (Slave Address) en modo lectura que es la b´11010001´D1h.
- ✓ Después el maestro lee los datos de los registros. La dirección del registro a leer se incrementa automáticamente.
- ✓ Por último el µC maestro envía la condición de "Stop".

El primer registro leído será el señalado en la última operación anterior realizada. Si este dato se desconoce, primero habrá que realizar una operación de escritura para conocer exactamente el valor de este puntero.

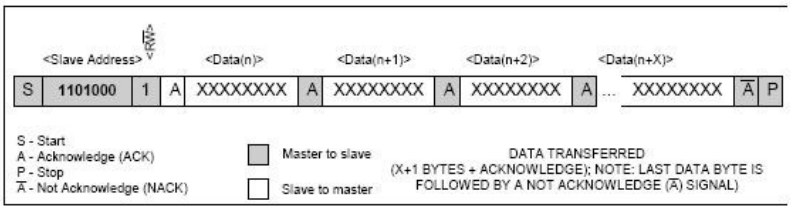


Figura 2.20 "Protocolo de lectura del DS1307"