2 INTRODUCCIÓN A LOS MICROCONTROLADORES

La electrónica ha avanzado mucho. Casi todo lo que hace unos años se hacía mediante un grupo, a veces muy numeroso, de circuitos integrados conectados entre si, hoy se puede realizar conectando un microcontrolador y unos pocos componentes adicionales.

El concepto de controlador ha permanecido inalterable a lo largo de los años, sin embargo, la implementación física de los mismos ha variado frecuentemente. Hace tres décadas, los controladores se construían exclusivamente con componentes de lógica discreta; posteriormente se utilizaron los microprocesadores, que se rodeaban de chips de memoria y periféricos de E/S sobre una tarjeta de circuito impreso y, actualmente, todos los elementos del microcontrolador se han podido incluir en un único integrado que recibe el nombre de microcontrolador.

Un microcontrolador es un circuito integrado que nos ofrece todas las posibilidades de un pequeño computador. En definitiva, un microcontrolador es un circuito programable que contiene todos las tres unidades fundamentales de un computador: CPU, Memoria y unidades de E/S que nos permiten tener una interfaz, más o menos directa entre el dispositivo y el mundo real. Se utilizan para controlar el funcionamiento de una determinada tarea, y gracias a su reducido tamaño, suele ir incorporado en el propio dispositivo que gobierna. Esta última característica es la que le confiere la denominación de controlador incrustado (embedded controller).

Para definirlo de forma sencilla podemos decir que un microcontrolador es una máquina tonta, un objeto sin razonamiento ninguno, un cubo negro con patitas metálicas que se suelda a una placa con más o menos componentes electrónicos y cuya funcionalidad es tan sencilla como la de una calculadora; frente a datos de entrada, sigue un programa, un algoritmo dado por un programador y cambia su estado interior, ofreciendo una salida. Es por tanto como un computador delicado. En su memoria solamente reside un programa destinado a gobernar una aplicación determinada, sus líneas de E/S soportan la conexión de los sensores y actuadotes del dispositivo a controlar y todos los recursos complementarios disponibles tienen como única finalidad atender sus requerimientos. Una vez programado y configurado, el microcontrolador solamente sirve para gobernar la tarea asignada.

Como objetos o dispositivos de entrada o salida podemos encontrar diversos periféricos, desde simples líneas de entrada digital que pueden estar a cero o a uno, hasta complejos puertos usados en ordenadores que permiten comunicar con otros dispositivos externos como microcontroladores o PC.

Con esta idea en la cabeza nos encontramos que existen varios fabricantes que ponen a disposición de los diseñadores miles de modelos distintos con características comunes y otras más especificas según la aplicación a la que estén diseñados, varios tamaños, diferencias consumo, periféricos, memoria, etc. La diversidad, además, tiene un objetivo fundamental, reducir costes. Mientras más características o más memoria, más espacio necesita y por tanto más caro será de fabricar y con ello de adquirir. Por tanto la tendencia es establecer diseños sencillos y con características limitadas. Con diversos modelos cada diseño tendrá un dispositivo que se adapte a las necesidades, eligiendo aquel que cumpla con todas las características de las especificaciones del producto a desarrollar y a la vez sea el más económico.

El procesador CPU es el elemento más importante del microcontrolador y determina sus principales características, tanto a nivel hardware como a nivel de software. Se encarga de direccionar la memoria de instrucciones, recibir el código de la instrucción en curso, su decodificación y la ejecución de la operación que implica la instrucción, búsqueda de los operandos y el almacenamiento del resultado. Existen tres orientaciones en cuanto a la arquitectura y funcionalidad de los procesadores actuales:

CISC: Computadores de Juego de Instrucciones Complejo: Disponen de más de 80 instrucciones máquina en su repertorio, algunas de las cuales son muy sofisticadas y potentes, requiriendo muchos ciclos para su ejecución.

RISC: Tanto la industria de los computadores comerciales como la de los microcontroladores están decantándose hacia la filosofía RISC (Computadores de Juego de Instrucciones Reducido). En estos procesadores el repertorio de instrucciones máquina es muy reducido y las instrucciones son simples y, generalmente, se ejecutan en un ciclo. La sencillez y rapidez de las instrucciones permiten optimizar el hardware y el software del procesador.

SISC: En los microcontroladores destinados a aplicaciones muy concretas, el juego de instrucciones, además de ser reducido, es “específico”, o sea, las instrucciones se adaptan a las necesidades de la aplicación prevista. Esta filosofía se ha bautizado con el nombre de SISC (Computadores de Juego de Instrucciones Específico).

Todos los microcontroladores disponen de bloques esenciales, como son: procesador, memoria de datos y memoria de instrucciones, módulos de E/S, oscilador de reloj y módulos controladores de periféricos. Además de estos elementos, existen una serie de recursos especiales que los fabricantes pueden ofertar tales como:

- **Temporizadores (Timers)**: Se emplean para controlar periodos de tiempo y para llevar la cuenta de acontecimientos que suceden en el interior.
- **Perro Guardián (Watchdog**): Es un temporizador que cuando se desborda y pasa por cero provoca un reset automáticamente en el sistema.
- **Protección frente a fallo de la alimentación (Brown-out)**

- **Estado de bajo consumo:** Es un estado del sistema donde se detiene el reloj principal y sus circuitos asociados con el objetivo de ahorrar energía en periodos de tiempo donde el microcontrolador se mantiene en espera de instrucciones.
- **Conversores A/D:** Procesa señales analógicas convirtiéndolas en señales digitales.
- **Modulador de Anchura de Pulso PWM:** Son circuitos que proporcionan en su salida impulsos de anchura variable, que se ofrecen al exterior a través de las patitas del encapsulado.
- **Comparadores analógicos:** Algunos modelos de microcontroladores disponen internamente de un amplificador operacional que actúa como comparador entre una señal fija de referencia y otra variable que se aplica por una de las patitas de la cápsula. La salida del comparador proporciona un nivel lógico 1 ó 0 según una señal sea mayor o menor que la otra.
- **Puertos de entrada y salida digital**
- **Puertos de comunicación** : Con objeto de dotar al microcontrolador de la posibilidad de comunicarse con otros dispositivos externos, otros buses de microprocesadores, buses de sistemas, buses de redes y poder adaptarlos con otros elementos bajo otras normas y protocolos. Algunos modelos disponen de recursos que permiten directamente esta tarea, entre los que destacan:

    - **UART**, adaptador de comunicación serie asíncrona.
    - **USART**, adaptador de comunicación serie síncrona y asíncrona.
    - **Puerta paralela esclava** para poder conectarse con los buses de otros microprocesadores.
    - **USB** (Universal Serial Bus), moderno bus serie para los PC.
    - **Bus I2C**, interfaz serie de dos hilos desarrollado por Philips.
    - **CAN** (Controller Area Network), permite la adaptación con redes de conexionado multiplexado desarrollado conjuntamente por Bosch e Intel para el cableado de dispositivos en automóviles.

Los microcontroladores pueden clasificarse según su arquitectura, que puede ser Von Neumann o Harvard.

La arquitectura **Von Neumann** se caracteriza por disponer de una sola memoria principal donde se almacenan datos e instrucciones de forma indistinta. A dicha memoria se accede a través de un sistema de buses único. El bus de direcciones es usado para identificar qué posición de memoria esta siendo accedida, mientras que el bus de datos es utilizado para trasladar información entre la CPU y alguna dirección de memoria o viceversa. Con un único sistema de buses, la arquitectura Von Neumann es usada secuencialmente para acceder a instrucciones de la memoria de programa y ejecutarlas regresando desde/hacia la memoria de datos. Esto significa que el ciclo de instrucción no puede solaparse con ningún acceso a la memoria de datos.

Una desventaja de esta arquitectura podría ser que el contador de programa o algún otro registro se corrompieran y apuntaran a la memoria de datos y se

tomara ésta momentáneamente como memoria de programa. Consecuentemente se ejecutaría una instrucción no deseada o un error en la decodificación de la instrucción.

La **Arquitectura Harvard** se caracteriza por disponer de dos memorias independientes, una que contiene sólo instrucciones y otra con sólo datos. Ambas disponen de sus respectivos sistemas de buses de acceso y es posible realizar operaciones de acceso (lectura o escritura) simultáneamente en ambas memorias. Una de las ventajas de esta arquitectura es que la operación del microcontrolador puede ser controlada más fácilmente si se presentara una anomalía en el contador de programa.

Existe otra arquitectura que permite accesos a tablas de datos desde la memoria de programa. Esta arquitectura es la llamada arquitectura **Harvard modificada**. Esta última arquitectura es la dominante en los microcontroladores actuales ya que la memoria de programa es usualmente ROM, OTP, EPROM o FLASH mientras que la memoria de datos es usualmente RAM, permitiendo que las tablas de datos estén en la memoria de programa para que estas no se pierdan cada vez que el sistema se apaga. Otra ventaja importante en la arquitectura Harvard modificada es que las transferencias de datos pueden ser solapadas con los ciclos de decodificación de instrucciones. Esto quiere decir que la siguiente instrucción puede ser cargada de la memoria de programa mientras se está ejecutando una instrucción que accede a la memoria de datos. La desventaja de la arquitectura Harvard modificada podría ser que se requieren instrucciones especiales para acceder a valores en memoria RAM y ROM haciendo la programación un poco complicada.

En cuanto a las técnicas de fabricación, casi la totalidad de los microcontroladores actuales se fabrican en tecnologías CMOS (Complementary Metal Oxide Semiconductor) que supera a todas las tecnologías existentes por su bajo consumo y su inmunidad frente al ruido.

2.1 ¿Para qué se usan los microcontroladores?

La aplicación de los microcontroladores a un proyecto de ingeniería tiene su límite en la imaginación de cada diseñador. Con los diversos modelos disponibles podemos afrontar multitud de proyectos distintos desde los más simples hasta los más complejos[1].

Las principales ventajas que se pueden encontrar con el uso de microcontroladores son:

- Gestión eficiente de procesos
- Aumento de la fiabilidad
- Reducción del tamaño, consumo y coste
- Mayor flexibilidad

Por nombrar algunos ejemplos de aplicaciones: mandos a distancia, termómetros digitales, controles de acceso por puertas de seguridad, los sistemas ABS o EPS de los coches, control y sensórica de maquinaria, domótica del hogar, microrobótica, monederos electrónicos …

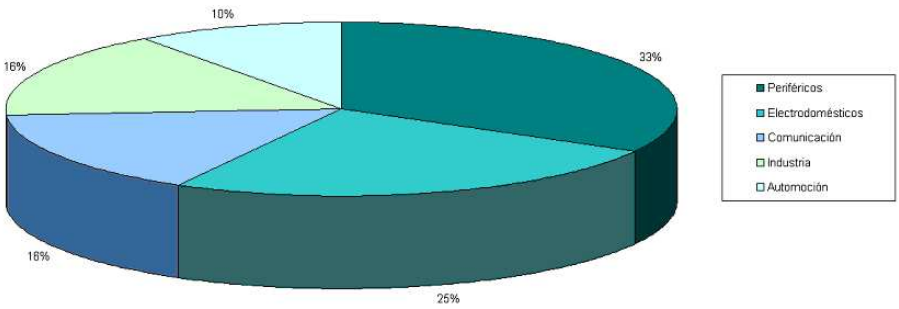


Fig. 4 Utilización de los microcontroladores por sectores.

2.2 Fabricantes y tipos de Microcontroladores

Realmente la dificultad no está en usar un microcontrolador para afrontar un proyecto hardware, sino en elegir el fabricante y el modelo adecuado para la aplicación. Una muestra de los posibles fabricantes a los que podemos acceder sería [2]:

- **MICROCHIP**
- **NATIONAL SEMICONDUCTOR**
- **MOTOROLA (FREESCALE)**
- **ZILOG**
- **INTEL**
- **THOMSON**
- **ANALOG DEVICES**
- **TEXAS INSTRUMENTS**

De todos los fabricantes que existen actualmente, los más utilizados suelen ser los dispositivos ofertados por Microchip, como ocurre con el proyecto que vamos a desarrollar. Esto se deba a las facilidades que proporciona la empresa para acceder a sus dispositivos, con una extensa documentación para cada modelo disponible y accesible desde su página web y ofreciendo muestras gratuitas a la universidad.

Además, se puede demostrar, con un estudio comparativo de todas las marcas y modelos como la mejor oferta de los dispositivos en cuanto a tamaño de ocupación en memoria y a velocidad de ejecución es la de MICROCHIP, lo que hace decantarnos por estos microcontroladores como primera opción a la hora de desarrollar muestro sistema.

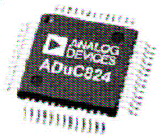
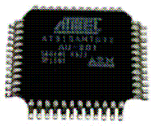
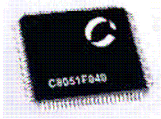
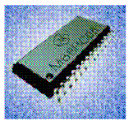
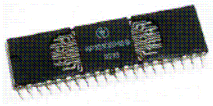
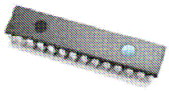
Empresa	Ejemplo Microcontrolador	Observaciones	Más Información
ANALOG DEVICES	 ADuC824	Fabricante de DSP's de 16 y 32 bits incluyendo sus herramientas de desarrollo. Amplia variedad de componentes analógicos de precision.	http://www.analog.com
ATMEL	 AT91SAM	Incluye microcontroladores derivados del 8051, el AT91SAM basados en ARM, y sus arquitecturas propias AVR y AVR32, dispositivos de radiofrecuencia, memorias EEPROM y Flash, ASICs, WiMAX,	http://www.atmel.com
CYGNAL	 C8051F040	Familia de Microcontroladores rápidos basados en 8051 incluyendo conversores A/D y sistemas de depurado.	http://www.cygnal.com
DALLAS SEMICONDUCTOR	MAXQ2000	Microcontroladores Flash Rápidos basados en 8051 añadiendo una bateria para mantener los datos en la SRAM. Extensas herramientas de desarrollo.	http://www.maxim-ic.com
freescale semiconductor	 M68HC08	Fabricante Estadounidense creado en 2004 a partir de la división de Motorola. Se centra en el mercado de los sistemas integrados y las comunicaciones.	http://www.freescale.com
intel Leap ahead	 KR1810VM86	Intel fue el fabricante de la arquitectura 8051, con más capacidad de memoria, con la familia MCS-48, despues la familia MCS-5; optimizados para usos orientados al control que requieren una extensa memoria RAM, alta integración y alto rendimiento.	http://www.intel.com/design/embcontrol
National Semiconductor	COP8SAx	Microcontroladores CompactRISC Y COP8, Fabrica un excelente microcontrolador de muy poco consume, el CompactRISC, con disponibilidad de modelos que incluyen Bluetooth, USB y CAN.	http://www.national.com/appinfo/mcu/
MICROCHIP	 PIC18F2550	Una arquitectura sencilla y eficaz, tecnologías de memoria flexibles, instrumentos de desarrollo fáciles de usar, apuntes comprensivos de aplicación, completan la documentación técnica y fijan el diseño. MArca más utilizada de Microcontroladores.	http://www.microchip.com
TEXAS INSTRUMENTS	MSP430	Líder mundial en DSP's. Produce también microcontroladores de 16 bits de bajo consumo, componentes analógicos, gíreles, productos de telecomunicaciones.	http://www.ti.com/

Fig. 5 Tabla de distintos fabricantes de Microcontroladores.

Sin embargo, a modo de ejemplo vemos la Fig. 5 Tabla de distintos fabricantes de Microcontroladores. con la característica principal de diseño de microcontroladores de algunos fabricantes y un modelo de ejemplo para cada uno de ellos. Se detallan además a continuación algunas características de los microcontroladores más populares:

- 8048 (Intel): es el padre de los microcontroladores actuales, el primero de todos. Su precio, disponibilidad y herramientas de desarrollo hacen que todavía hoy sea muy demandado.

- 8051 (Intel): es sin duda el microcontrolador más popular. Fácil de programar pero potente. Esta bien documentado y posee cientos de variantes e incontables herramientas de desarrollo.

- 68HC11 (Motorota y Toshiba): Es un microcontrolador que llego a tener cierto existo. Bastante potente y con una gran cantidad de variantes.

- 683xxx (Motorota): Surgido a partir de la familia 68k, a la que se incorporan algunos periféricos. Son microcontroladores de altísimas prestaciones.

PIC (Microchip): Familia de microcontroladores que gana en popularidad día tras día. Fueron los primeros microcontroladores RISC.

2.3 Microcontrolador PIC

Entre los distintos fabricantes de microcontroladores nos decantamos por Microchip Technology. Los microcontroladores PIC[1] de Microchip están muy extendidos actualmente en el mercado gracias a su gran variedad y bajo coste. Otra razón de su éxito es su utilización, ya que una vez se ha aprendido a utilizar uno, conociendo su arquitectura y juego de instrucciones, es muy fácil emplear otro modelo diferente.

Para clasificar los distintos tipos de PIC se utiliza una nomenclatura que atiende a sus características, tanto físicas, indicando el número de pins que posee el encapsulado del PIC, como eléctricas o de procesamiento, ya que podemos indicar el tamaño de la palabra de memoria que es capaz de manejar el procesador UAL (Unidad Aritmético Lógica), o lo que es lo mismo, el ancho o número de bits de la palabra en la memoria de datos (registros en la terminología usada por Microchip).

La variedad de PICs disponibles en el mercado es bastante amplia, desde PIC pequeños de 6-pin y 8-bits con sólo 16 bytes de memoria de datos con los que

[1] PIC se refieren a una extensa familia de microcontroladores fabricados por Microchip Tecnology Inc.

únicamente podemos fabricar una I/O digital básica, hasta dispositivos de 100-pin y 16-bit, con 30 kilobytes de memoria de datos, diseñados para ofrecer posibilidades prácticamente infinitas para el procesamiento digital de la señal.

Los PIC menos prestigiosos o con menos prestaciones, aquellos que operan con palabras de datos de 8 bits, están divididos en 3 familias que se diferencian básicamente por su estructura:

- **Baseline (12-bit instuctions)**

El diseño de estos PIC se basa en la arquitectura original de los procesadores, volviendo a las estructuras de 1970, año en que Gilbert Hyatt patenta el microprocesador como un único chip. Estos dispositivos tienen muchas limitaciones comparándolos con el resto de las opciones disponibles en el mercado, ya que, por ejemplo, no presentan mecanismos de detección de eventos por interrupciones por lo que su manejo es más complejo; sin embargo poseen un diseño muy simple y son bastante sencillos a la hora de trabajar con ellos en ensamblador. Algunos de estos dispositivos que podemos encontrar hoy en día serían los incluidos en la serie 10F de 6-pines, o el 12F509 de 8-pines e incluso el dispositivo 16F506 de 14-pines.

- **Midrange (14-bit instructions)**

Aparecen cuando el diseño básico de la arquitectura convencional comenzó a desarrollarse, añadiendo el soporte necesario pata poder trabajar con interrupciones, más memoria, temporizadores, algunos periféricos, incluso PWM, controles de motores, choppers para sensores en ambientes ruidosos y algunas otras aplicaciones. También se añadió el soporte para la transferencia de datos en serie, interfaces I^2C y SPI y controladores con pantalla LCD. Algunos ejemplos de esta clase de dispositivos los encontramos en las series 12F ó 16F, como los dispositivos: 12F629 de 8-pines, 16F629 de 20-pines o el 16F917 de 40-pines.

- **High-end (16-bits instructions)**

Se conoce con este nombre a los dispositivos de la serie 18F, que superan muchas de las limitaciones que presentan los dispositivos de las dos familias anteriores. Provistos de más memoria (más de 128k de memoria de programa y aproximadamente 4k para memoria de datos) y de periféricos mucho más modernos incluyendo comunicación USB o Ethernet. La arquitectura que presentan los dispositivos de la familia 18F está desarrollada para soportar la programación con lenguaje C. Ejemplos de estos dispositivos serían los de 18-pines, como el 18F1220, de 28-pines: 18F2455 y de incluso con 80-pines, dispositivo 18F8520.

Es importante recordar que los dispositivos de la familia 18F tienen 16-bit para definir las instrucciones de programa que operan con datos de anchura 8 bits, por lo que serán considerados dispositivos de 8-bits. Existen otros microcontroladores o dispositivos de 16-bit, como los pertenecientes a la familia PIC24F, pero no los desarrollaremos [3].

Con las tres gamas de PIC se dispone de gran diversidad de modelos y encapsulados, pudiendo seleccionar el que mejor se acople a las necesidades de acuerdo con el tipo y capacidad de las memorias, el número de líneas de E/S y las funciones auxiliares precisas. Sin embargo, todas las versiones están construidas alrededor de una arquitectura común, un repertorio mínimo de instrucciones y un conjunto de opciones muy apreciadas, como el bajo consumo y el amplio margen del voltaje de alimentación.

La nueva familia PIC18F Microchip pretendía resolver dos grandes metas [1]:

1. **COMPATIBILIDAD:**
   - Manteniendo el núcleo de la nueva familia compatible con el de los PIC 16F y sus periféricos.
   - Soportando la compatibilidad de software a nivel de código Ensamblador y de lenguaje C
2. **ELEVADO RENDIMIENTO**:
   - Máxima eficiencia del compilador C, que supera en un 50% al de los PIC16F.
   - Elevada productividad.

Las letras que hay en los diferentes modelos de cada PIC, en este caso la F (18F2550), indican el tipo de memoria de programa que utiliza. En este caso se tiene una memoria de tipo Flash, que es una memoria no volátil, de bajo consumo y que se puede escribir y borrar eléctricamente. Su funcionamiento es como el de las memorias ROM y RAM, pero consume menos y es más pequeña. A diferencia de la ROM, la memoria Flash es programable en el propio circuito. Es más rápida y de mayor densidad que la EEPROM. La alternativa Flash es más recomendable que la EEPROM cuando se necesita gran cantidad de memoria de programa no volátil. Es más rápida y tolera más ciclos de escritura/borrado.

Estos microcontroladores tienen además una serie de mejoras técnicas de las que destacamos:

⇒ Tecnología “nanoWat”. Reduce el consumo de energía considerablemente ofreciendo la posibilidad al usuario de elegir el modo de bajo consumo en cualquier parte del programa.

⇒ Espacio de direccionamiento lineal. Se puede controlar un espacio de direccionamiento lineal de la memoria de programa de hasta 2MB.

⇒ Compatibilidad Software. Todos los modelos de la gama mejorada poseen un repertorio de 77 instrucciones máquina, de una longitud de 16bits, muchas de las cuales son similares a las usadas en los PIC16F.

⇒ Compatibilidad Hardware. Independientemente de la capacidad de memoria todos los modelos de la gama mejorada comparten los mismos periféricos, lo que facilita una rápida y fácil transición entre ellos.

⇒ Orientación al lenguaje C. El aumento de la capacidad de memoria y en los modos de direccionamiento facilita el trabajo de los compiladores C. Estos recursos han propiciado el uso del lenguaje C, con direccionamiento pre y post incremento y flexibilidad en el manejo de la pila.

⇒ Auto-programabilidad. Los PIC18F son capaces de escribir en sus propios espacios de memoria de programa bajo el control de un software interno.

⇒ Diversidad de periféricos integrados. Entre los múltiples recursos complemnentarios que integran los PIC 18F destacan los módulos de comunicación disponibles como el USART, SSP, SPI, I2C, USB, etc. También los módulos CCP de Captura/Comparación/PWM y los conversores AD de hasta 16 canales.

Para trabajar con un PIC es necesario:

⇒ Un ordenador personal, con sistema operativo Windows XP y conexión con puerto USB.
⇒ Elección del PIC.
⇒ Un programador para el PIC.
⇒ Software de desarrollo, como un ensamblador, un editor, depurador y un compilador de C.

2.3.1 PIC Programmers

Hace algunos años, para borrar la información contenida en un PIC, estos debían ser expuestos a una luz Ultra Violeta a través de una ventana sobre el Chip; y para cargar sobre ellos un programa era necesario tener un programador especial.

Actualmente los PIC utilizan memorias conocidas como Flash, que pueden ser borradas eléctricamente miles de veces, sin necesidad de ningún equipamiento especial y puede ser programados a través de un protocolo denominado **ICSP (In-Circuit Serial Programming)**. Este protocolo se desarrolló con la finalidad de resultar lo más cómodo posible para el programador, por lo que en lugar de preocuparse por adaptar el diseño del circuito para el que programaremos el PIC a las necesidades del protocolo ICSP, se optó por mover el PIC desde su circuito original a una tarjeta prototipo propia del programador adaptada a las necesidades de conexión del protocolo ICSP y así poder usar el mismo programador para cualquier circuito que queramos diseñar.

Microchip oferta como programador para sus dispositivos el **PICkit 2**. Utilizar en nuestro diseño un dispositivo y un programador desarrollados por el mismo fabricante asegurara el acceso a todas las posibilidades del PIC sin encontrarnos con problemas derivados de incompatibilidades de características.

En la Fig. 6 vemos el programador que Microchip pone a nuestra disposición para trabajar con los microcontroladores.

Fig. 6 Programador PICkit 2. Microchip

2.3.2 Placa o Tarjeta Prototipo

Usando una programación bajo el estándar de programación ICSP necesitamos, como ya hemos explicado con anterioridad, conectarnos con el PIC para descargar la aplicación que hemos diseñado. Para esto Microchips desarrolló el “Low Pin Count - LPC- Demo Borrad" diseñado para ser utilizado junto con el programador PICkit 2.

La LPC Demo Borrad incluye la base de la tarjeta con el soporte sobre el que conectar un PIC para que el programa pueda ser transferido, cuatro LEDs, un interruptor de contacto y un potenciómetro (resistencia variable). Además sobre la placa, tenemos un área en la que podemos añadir componentes que necesitemos para nuestra configuración.

La LPC Demos Board soporta todos los PIC modernos (que utilizan memorias Flash) 8-,14, y 20-pin PIC Baseline y Midrange. Muchas de las líneas de I/O del PIC están conectadas como salidas en el soporte de 14 pin que se encuentra en el lateral de la placa prototipo, permitiendo que la tarjeta sea conectada a otro circuito, permitiendo configurar y depurar tu diseño sin necesidad de fabricar la placa con el PIC integrado en cada ocasión.

Existen actualmente un número determinado de tarjetas prototipo, algunas de ellas vienen con más periféricos integrados, lo que permite configurar muchas más aplicaciones, sin embargo tener una tarjeta u otra no es limitación alguna para trabajar de manera completa con nuestro PIC para depurar nuestro diseño, ya que es muy sencillo construir nuestra propia tarjeta prototipo adaptada a nuestras necesidades de diseño, incluso que puedan soportar un PIC más largo, como es el caso del utilizado en nuestro proyecto, 18F2550.

Fig. 7 Placa Prototipo PicKit 2

2.3.3 Software Necesario

El lenguaje nativo de los microcontroladores es el ASM, sin embargo, el ASM es un lenguaje que esta mucho mas cerca del hardware que del programador. Esto, unido al aumento de la capacidad de memoria de los microprocesadores, ha fomentado la aparición de compiladores de lenguajes de alto nivel.

Los lenguajes de programación más extendidos son el Basic y el C, aunque también se pueden encontrar compiladores de lenguaje orientado a objetos como son el C++ o java, sin embargo estos compiladores no se encuentran tan desarrollados y optimizados como para tenerlos en consideración para una aplicación compleja de programar.

Todos los PIC necesitan, por tanto, un traductor del microchips. **MPLAB** es un entorno de desarrollo integrado (**IDE**) que permite hacer esta transformación. Es un programador serial o USB de una gran cantidad de micros PIC y otros integrados de la gran familia de MICROCHIP. Con la tecnología de Microchip para realizar Debugger "In-Circuit", es posible bajar el programa a su sistema, ejecutarlo en tiempo real y al mismo tiempo examinar en detalle el programa con la ayuda de MPLAB. Es posible ver el valor de las variables, elegir puntos de ruptura (breakpoints) en C o en ensamblador. Se puede analizar el código en C o ensamblador paso por paso.

Esta herramienta incluye un ensamblador propio de Microchip, conocido como **MPASM**, un editor y un software de simulación que te permite depurar la aplicación antes de trasladarla al Chip. Hace algún tiempo una herramienta como esta, tan sofisticada, podía costar bastante, sin embargo, actualmente MPLAB es un programa de distribución gratuita, completamente libre, distribuido por el propio Microchip.

MPLAB soporta directamente la conexión con el programador PICkit 2 para todos los dispositivos de Midrange y Baseline PIC, por lo que también necesitaremos para trabajar el software que permite el control del programador y que viene incluido en el CD junto con la tarjeta o placa prototipo pero que también podemos encontrar en la página de Microchip para descargarlo de forma gratuita. Conviene además comprobar que tenemos la última versión, ya que cada nueva versión añade características que permiten soportar nuevos PICs.

MPLAB, desde la versión 7.41, incluye una copia de libre distribución del compilador de lenguaje de programación C llamado **CCS**. Sin embargo esto en muchas ocasiones no supone un gran avance, ya que la disponibilidad de la versión gratuita es tan limitada que hace que, en muchos casos, necesites usar un ensamblador.

Estudiando todas las posibilidades de diseño comprobamos que es posible usar MPLAB como una herramienta de desarrollo integrado completa para ensamblar los programas de forma tan completa como si usáramos un programador de lenguaje C.

Los dispositivos de la serie 18F tienen además que utilizar el llamado compilador C18, que se vende de forma separada y que no es barato; sin embargo, existe una versión para estudiantes con algunas características desactivadas pero funcionando completamente en cuanto a prestaciones se refiere y que también se encuentra disponible en una versión libre que podemos descargar sin problemas, además, al igual que los otros compiladores de C, C18 permite estar completamente integrado dentro de MPLAB.

Sin embargo, esta no es la única forma de trabajar con el dispositivo. El PICkit 2 viene con una versión limitada del compilador de C distribuido por Hi-Tech's, llamado "PICC-Lite". Este soporta un pequeño número de dispositivos de Midrange y para los dispositivos High-end, como el 16F690 no permite usar toda la memoria del PIC; por es muy útil para iniciarse en los programadores de C para dispositivos Midrange, pero queda descartado de la selección por sus limitaciones.

En este proyecto trabajaremos con lenguaje de **programación C** y el **compilador C18** que presenta Microchip. Ha sido elegido por su capacidad de generar un código de gran compactación y por su alta difusión en el mundo de la enseñanza, gracias a la versión gratuita.

Todo este software se detallará en el siguiente apartado (Punto 2 Software necesario para realizar la aplicación) de esta memoria, detallando algunas características de los mismos.