

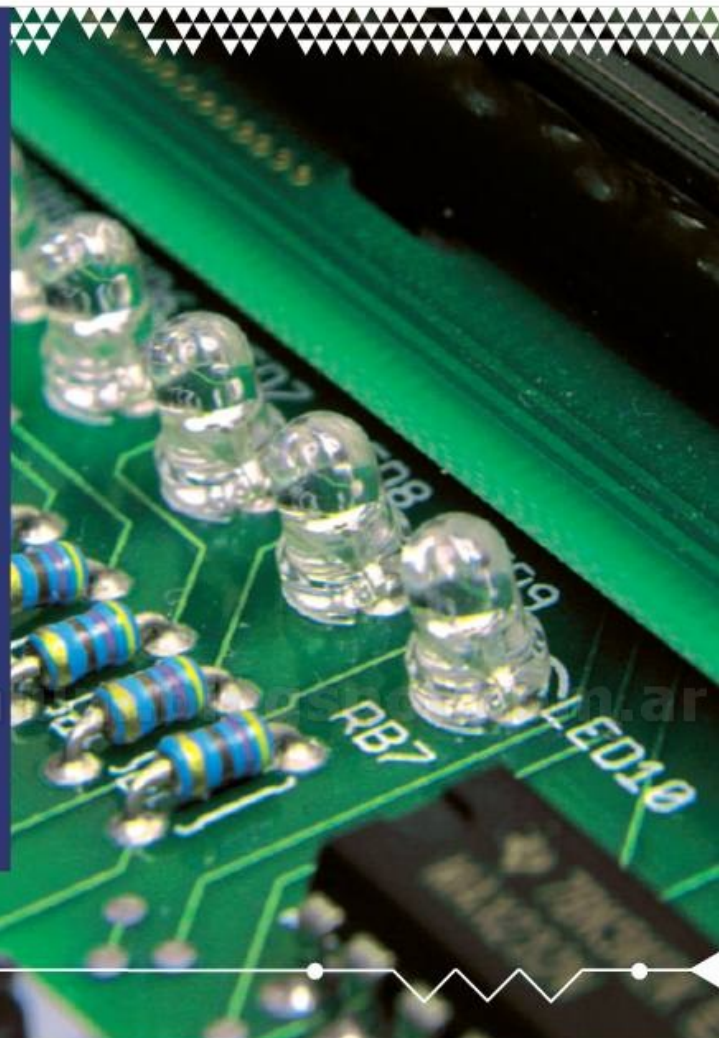
TÉCNICO en ELECTRÓNICA

CONCEPTOS FUNDAMENTALES Y PRÁCTICA PROFESIONAL

Microprocesadores y microcontroladores

- ▶ Arquitecturas de computadoras
- ▶ Elementos de un microcontrolador
- ▶ Lenguaje Assembler
- ▶ Registros y memoria

www.reduserspremium.com.ar



USERS

TÉCNICO en ELECTRÓNICA

CONCEPTOS FUNDAMENTALES Y PRÁCTICA PROFESIONAL

Coordinación editorial

Paula Budris

Asesor técnico

Federico Pacheco

Nuestros expertos

Diego Aranda

Esteban Aredez

Luis Ávila

Alejandro Fernández

Agustín Liébana

Lucas Lucyk

Luis Francisco Macías

Mauricio Mendoza

Norberto Morel

Juan Novo

David Pacheco

Federico Pacheco

Gerardo Pedraza

Mariano Rabioglio

Luciano Redolfi

Juan Ignacio Retta

Alfredo Rivamar



Técnico en electrónica es una publicación de Fox Andina en coedición con Dálaga S.A. Esta publicación no puede ser reproducida ni en todo ni en parte, por ningún medio actual o futuro sin el permiso previo y por escrito de Fox Andina S.A. Distribuidores en Argentina: Capital: Vaccaro Sánchez y Cia. S.C., Moreno 794 piso 9 (1091), Ciudad de Buenos Aires, Tel. 5411-4342-4031/4032; Interior: Distribuidora Interplazas S.A. (DISA) Pte. Luis Sáenz Peña 1832 (C1135ABN), Buenos Aires, Tel. 5411-4305-0114. Bolivia: Agencia Moderna, General Acha E-0132, Casilla de correo 462, Cochabamba, Tel. 5914-422-1414. Chile: META S.A., Williams Rebolledo 1717 - Ñuñoa - Santiago, Tel. 562-620-1700. Colombia: Distribuidoras Unidas S.A., Carrera 71 Nro. 21 - 73, Bogotá D.C., Tel. 571-486-8000. Ecuador: Disandes (Distribuidora de los Andes) Calle 7° y Av. Agustín Freire, Guayaquil, Tel. 59342-271651. México: Distribuidora Intermex, S.A. de C.V., Lucio Blanco #435, Col. San Juan Tlihuaca, México D.F. (02400), Tel. 5255 52 30 95 43. Perú: Distribuidora Bolivariana S.A., Av. República de Panamá 3635 piso 2 San Isidro, Lima, Tel. 511 4412948 anexo 21. Uruguay: Espert S.R.L., Paraguay 1924, Montevideo, Tel. 5982-924-0766. Venezuela: Distribuidora Continental Bloque de Armas, Edificio Bloque de Armas Piso 9no., Av. San Martín, cruce con final Av. La Paz, Caracas, Tel. 58212-406-4250.

Impreso en Sevagraf S.A. Impreso en Argentina.

Copyright © Fox Andina S.A. VI, MMXIII.

Anónimo

Técnico en electrónica / Anónimo ; coordinado por Paula Budris. - 1a ed. - Buenos Aires : Fox Andina; Dalaga, 2013.

576 p. ; 27x19 cm. - (Users; 23)

ISBN 978-987-1949-14-4

1. Informática. I. Budris, Paula, coord. II. Título.

CDD 005.3

En esta clase veremos

UNO DE LOS TEMAS MÁS APASIONANTES DE LA ELECTRÓNICA DIGITAL EN LA ACTUALIDAD:
LOS MICROPROCESADORES Y LOS MICROCONTROLADORES.



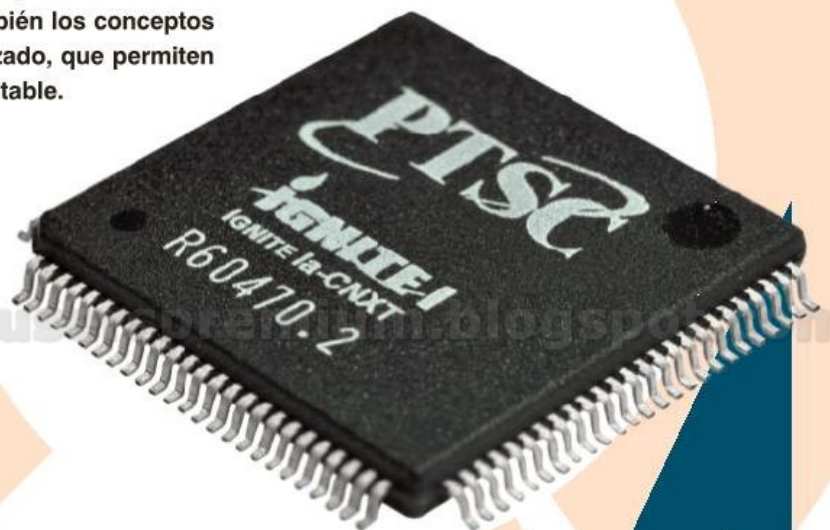
En esta oportunidad, introduciremos un tema muy conocido, incluso, por gente no tan cercana a la electrónica, sino más bien a la informática: los microprocesadores y los microcontroladores. Para ello, comenzaremos presentando la arquitectura de computadoras, su diseño conceptual, y los modelos de Von Neumann y de Harvard, y los conceptos asociados, la unidad de control, unidad aritmética y lógica, unidad de punto flotante, bus de control y bus de datos, así como también los conjuntos de instrucciones CISC y RISC.

Veremos, además, lo referente a los mecanismos de entrada y salida de los microcontroladores y las interrupciones. En cuanto a la CPU, el cerebro del micro, veremos qué son los registros y la palabra de estado. En cuanto a las memorias, veremos los distintos tipos básicos: ROM, RAM y caché.

Finalmente, para terminar de entender el funcionamiento a bajo nivel (cerca de las plaquetas) de los microcontroladores y los microprocesadores, necesitaremos además conocer sobre el famoso lenguaje ensamblador y lenguaje de máquina, con sus instrucciones y nemónicos, así como también los conceptos de compilador, ensamblado y enlazado, que permiten transformar código escrito en ejecutable.

SUMARIO

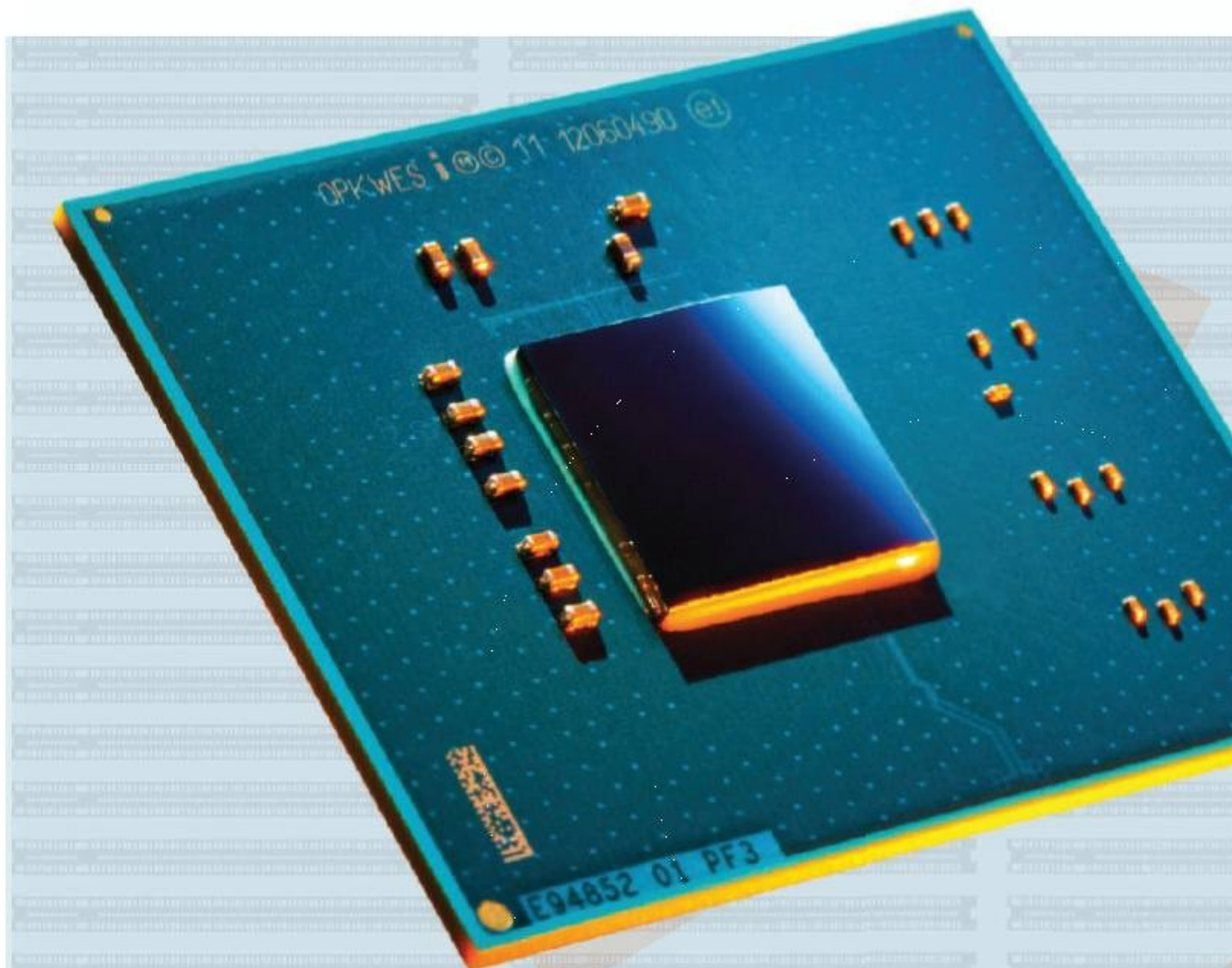
- 02** ARQUITECTURA DE COMPUTADORAS
Microcontroladores, microprocesadores y principales arquitecturas.
- 10** ELEMENTOS DE UN MICROCONTROLADOR
Conceptos fundamentales de una CPU.
- 18** LENGUAJE ENSAMBLADOR
Características básicas y uso del lenguaje.





2

► Clase 13 //

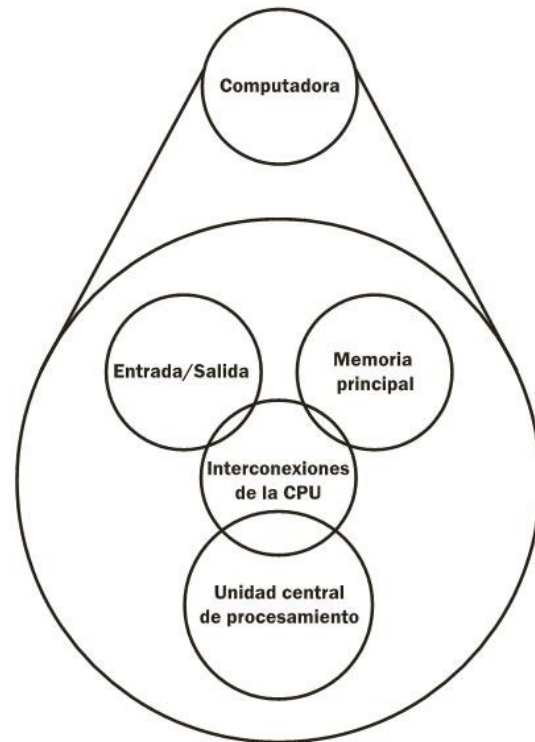


ARQUITECTURAS DE COMPUTADORAS

www.raduserspremium.blogspot.com.ar

DEFINIREMOS CONCEPTOS BÁSICOS SOBRE ARQUITECTURA DE COMPUTADORAS. VEREMOS LAS VENTAJAS Y DESVENTAJAS DE CISC Y RISC, LAS DIFERENCIAS ENTRE MICROPROCESADORES Y MICROCONTROLADORES.





Estructura básica de una computadora.

▼ **Procesamiento de datos**, que puede adoptar una gran variedad de formas; el rango de requisitos de procesado es amplio.

▼ **Almacenamiento de datos**, incluso si se procesan datos al vuelo (es decir, se introducen, se procesan y se entregan inmediatamente), estos se deben guardar con los que se están usando, en forma temporal.

▼ **Transferencia de datos** entre él mismo y el mundo exterior. El proceso se conoce como entrada-salida (E/S), y el dispositivo recibe el nombre de **periférico**.

▼ **Control** de las tres funciones anteriores.

Estructura

Encontramos, también, cuatro componentes dentro de la estructura básica de una computadora:

▼ **Unidad central de Procesamiento** (CPU, *Central Processing Unit*): controla el funcionamiento de la computadora y lleva a cabo sus funciones de procesamiento de datos.

▼ **Memoria principal**: almacena datos e información.

P

ara empezar, debemos entender de qué estamos hablando cuando nos referimos a “arquitectura de computadoras”. Cuando se habla de arquitectura de computadoras se hace referencia a los **atributos de un sistema que son visibles al programador**, es decir, que tienen importancia en el momento de ejecución de un programa. No se debe confundir con **organización de computadoras**, que es la forma en que se van a interconectar y armar los elementos del sistema, y el tipo de tecnología que se usará. Esto es transparente para el programador.

Una computadora tiene actualmente muchísimos componentes electrónicos, esto hace que sea fundamental, tanto para el análisis como para el diseño, la división jerárquica en subsistemas interrelacionados. Cada uno de estos se desarrolla en estructuras hasta alcanzar el nivel más bajo o elemental. De cada nivel importan el funcionamiento y la estructura:

▼ **Funcionamiento**: operación de cada componente individual como parte de la estructura.

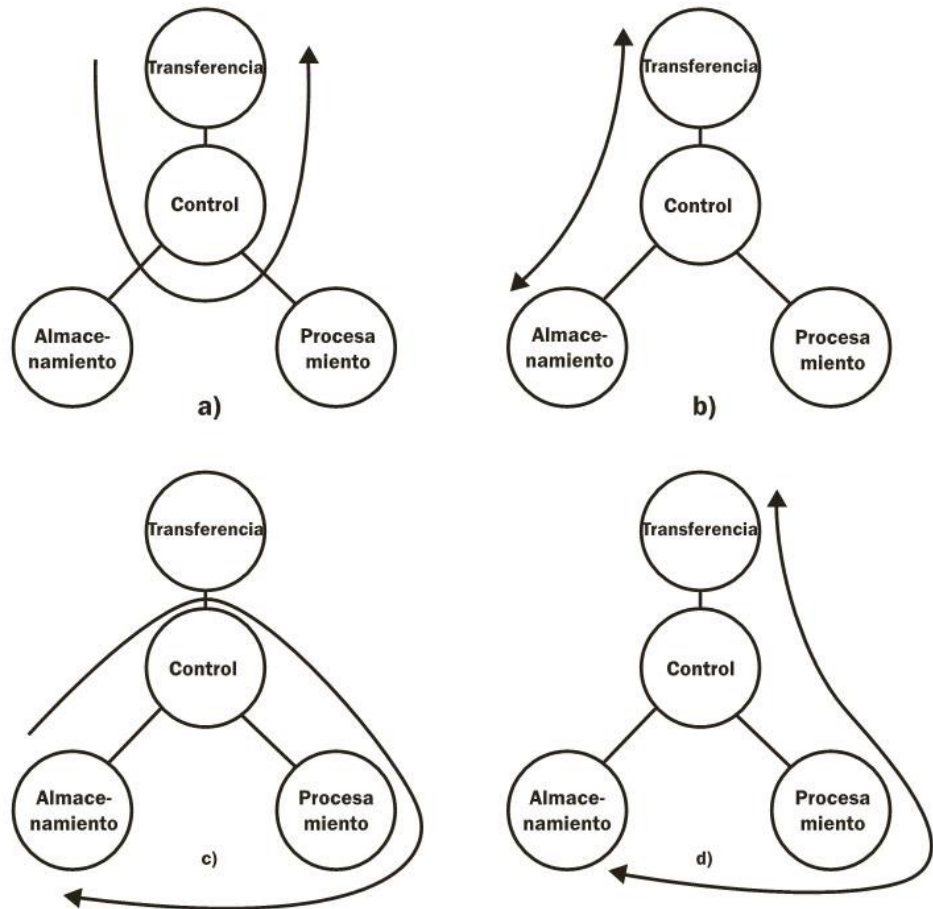
▼ **Estructura**: modo en el que los componentes están interrelacionados.

Funciones básicas

En términos generales, solo hay cuatro funciones básicas que una computadora lleva a cabo:

COMPATIBILIDAD EN EL TIEMPO

Desde hace mucho tiempo y aún hoy en día, la diferencia entre organización y arquitectura siempre ha sido importante. Muchos fabricantes mantienen su arquitectura, pero su organización va cambiando año a año. Creando **familias de modelos**, se pueden establecer escalas con distintos niveles de prestaciones y precios; se puede cambiar entre uno y otro, porque tienen la misma arquitectura y logran ser **compatibles** entre sí. Así es como las arquitecturas permanecen en el tiempo, pero, en cambio, la organización cambia de acuerdo a los avances tecnológicos.



▲ Posibles operaciones de una computadora. a) Transferencia de datos de un periférico a otro. b) Almacenamiento de datos. c) Procesamiento de datos almacenados. d) Procesamiento de datos en tránsito.

▼ **E/S:** transfiere datos entre la computadora y el entorno externo.

▼ **Sistemas de interconexión:** mecanismo que proporciona la comunicación entre la CPU, la memoria principal y la E/S.

La forma en la que se dispone la estructura de una computadora ha dado origen a los modelos de Von Neumann y de Harvard.

La **arquitectura de Von Neumann** tiene la particularidad de que los procesadores poseen el mismo dispositivo de almacenamiento tanto para guardar información como para las instrucciones, haciendo que un **único bus de datos** sea requerido para comunicarse con la CPU y mejorar la eficiencia en la utilización de memoria. La contrapartida es una ambigüedad para reconocer los datos, y genera lo que se conoce como **cuello de botella de Von Neumann (Von Neumann Bottleneck)**. Se ha resuelto este problema introduciendo una memoria caché entre la CPU y la memoria principal para mejorar los algoritmos del predictor de saltos.

En cambio, la **arquitectura de Harvard** se diferencia por la separación entre la memoria de datos y la de instrucciones, usando **distintos buses de información**. Con esta arquitectura, se reduce la ambigüedad, aunque no resulta tan eficiente en la utilización de la memoria.

RISC vs CISC

Con el correr de los años, se ha puesto mucho entusiasmo en mejorar el rendimiento de las computadoras, con el objetivo de

EN LA ACTUALIDAD, LA MAYORÍA DE LOS ORDENADORES SE BASAN EN LA ARQUITECTURA VON NEUMANN, QUE, AUNQUE PUEDEN RESULTAR UN POCO MÁS LENTAS, OFRECEN UNA MAYOR FLEXIBILIDAD.





incrementar la velocidad de funcionamiento de sus componentes. En general, las mejoras más notables se lograban debido al aumento en la frecuencia de la CPU. El diseño y la evolución de los microprocesadores llevó a la controversia entre arquitecturas RISC y CISC.

Veamos qué significan estos últimos términos, y sus respectivas ventajas y desventajas.

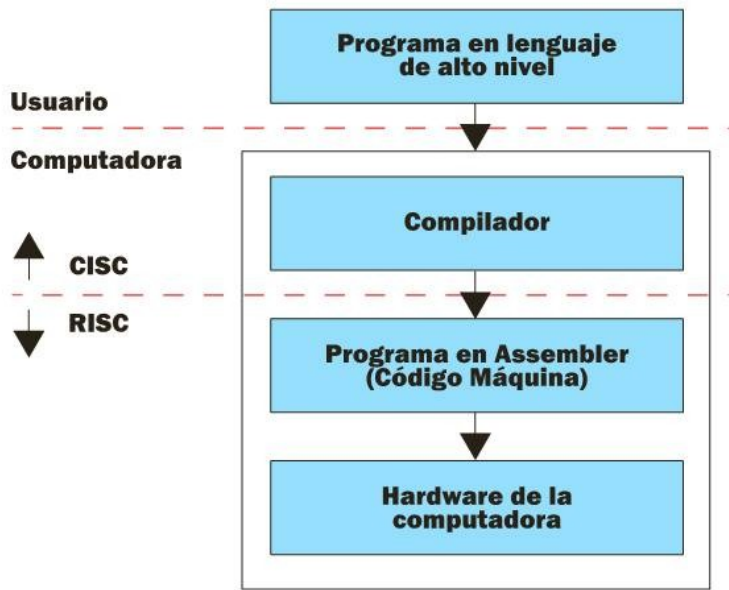
CISC viene del inglés *Complex Instruction Set Computer*, en español computadora con conjunto de instrucciones **complejas**. En contraparte, **RISC** es *Reduced Instruction Set Computer*, computadora con conjunto de instrucciones **reducidas**.

Pero... ¿a qué nos referimos con la expresión "conjunto de instrucciones complejas o reducidas"? Para entender estos conceptos, observemos la siguiente figura, donde se ven las distintas **capas** que unen el lenguaje de máquina que "entiende" el microprocesador, con el lenguaje de alto nivel realizado por el programador.

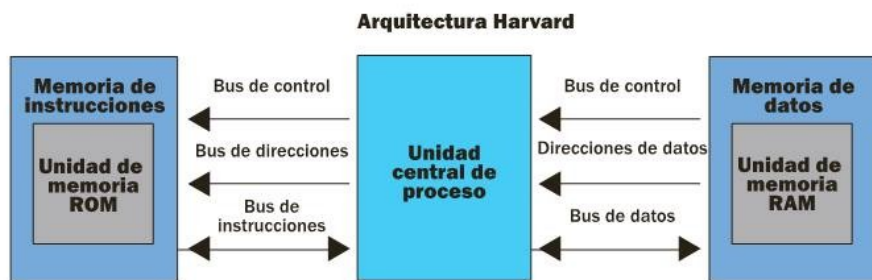
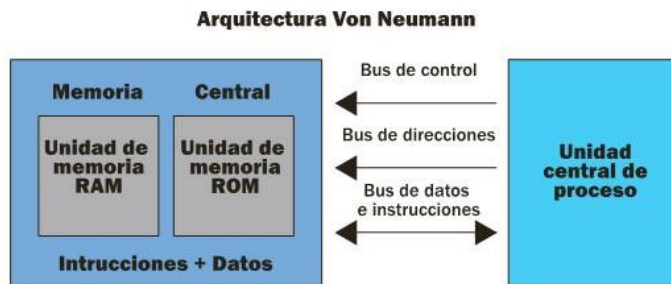
Las arquitecturas CISC nacieron frente a la necesidad de tener un amplio conjunto de instrucciones, dado que muchos programas fueron desarrollados en lenguaje de máquina directamente. Mediante una Unidad de Control Microprogramada, se obtenía fácil implementación y uso eficiente del espacio en el chip, además de la posibilidad de modificar el conjunto de instrucciones y simular nuevas arquitecturas.

Las características más sobresalientes de las arquitecturas CISC son:

- ▼ Instrucciones de longitud variable que depende del modo de direccionamiento usado en los operandos.
- ▼ Las instrucciones necesitan múltiples ciclos de reloj para ejecutarse.
- ▼ Las CISC soportan cero, uno o más operandos.



Niveles de abstracción de una arquitectura.



Modelos de arquitectura Harvard y de Von Neumann.

IMPLEMENTAR UNA INSTRUCCIÓN DE ALTO NIVEL DIRECTAMENTE EN UNA INSTRUCCIÓN DE CÓDIGO DE MÁQUINA PERMITE OBTENER CÓDIGO MÁS SIMPLE Y MÁS EFICIENTE.





El microprocesador de 4 núcleos ARM Cortex A15 se puede encontrar en celulares de alta gama, como el Samsung Galaxy S4.

▼ Variedad del direccionamiento de operandos, de registro a registro, registro a memoria y memoria a registro.

▼ Diversos modos de direccionamiento, como el directo de memoria, indirecto de memoria y el indexado a través de registros.

Hoy en día, CISC tiene a **x86** como su mayor exponente, con **AMD** y, sobre todo, **Intel** a la cabeza de su desarrollo. Prácticamente cualquier computadora de escritorio o portátil desde los años 80 ha utilizado un procesador x86.

En cambio, la filosofía RISC nace a fines de los años setenta cuando, en investigaciones de la frecuencia de empleo de una instrucción en un CISC y el tiempo que tardaba en llevarse a cabo, se observó lo siguiente:

▼ Cerca del 80% del tiempo total en la ejecución de un programa está dominado por alrededor de solo el 20 % de las instrucciones.

Existen encadenamientos de instrucciones simples que logran un resultado idéntico al de secuencias complejas predeterminadas, pero demandan lapsos de ejecución más breves.

Por otra parte, se analizaban mejoras como la de **pipeline** (segmentación de cauce), que permitía superponer etapas en cada ciclo de instrucción para optimizar los tiempos de espera y lograr mayor eficiencia. Estas mejoras necesitaban espacio en el chip, y reducir la complejidad de las instrucciones se veía como una opción prometedora.

De esta manera nacen las arquitecturas RISC, que tenían las siguientes características principales:

▼ Poseen un número significativamente menor de instrucciones y, a la vez, más simples.

▼ La mayoría de las instrucciones son de la misma longitud, lo que permite que una instrucción se busque con una operación individual.

▼ La mayoría de las instrucciones se ejecutan en un solo ciclo, esto permite la capacidad de manejar varias instrucciones al mismo tiempo implementando la segmentación (pipelining).

Aunque se logró mayor velocidad, se encontraban desventajas, como la ne-

DESVENTAJAS DE CISC Y RISC

Entre las desventajas de CISC, encontramos las siguientes:

- ▼ La complejidad del conjunto de instrucciones sube, incluso, a veces se incorporan instrucciones que no se utilizan tanto.
- ▼ Las instrucciones de longitud variable reducen el rendimiento del sistema.

RISC presenta las siguientes desventajas:

- ▼ Extremada dependencia en la efectividad del compilador
- ▼ Se hace más laboriosa la depuración de los programas por la programación de instrucciones (**Instruction Scheduling**).
- ▼ El tamaño del código del lenguaje de máquina aumenta.
- ▼ Se necesitan memorias más veloces.

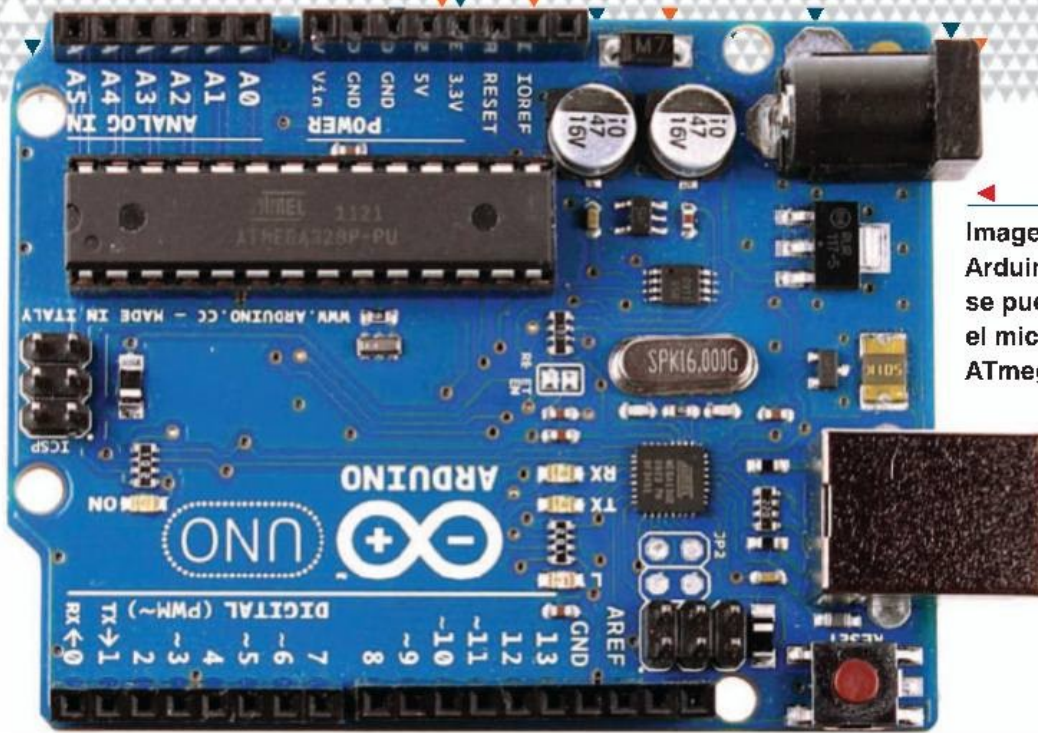


Imagen real de un kit Arduino UNO, donde se puede observar el microcontrolador ATmega328 de Atmel.



cesidad de tener un compilador más robusto para poder unir la brecha entre el lenguaje de alto nivel y el código de máquina.

En la actualidad, los **productos ARM** son el principal ejemplo de **procesador RISC**; estos se usan en casi todos los dispositivos móviles, pero también en otros terrenos, como las supercomputadoras.

Microprocesadores y microcontroladores

Es probable que el lector sepa que, dentro de su computadora personal, tiene un microprocesador Intel o AMD, memorias RAM, disco duros y demás componentes. Cada uno con una determinada función. El microprocesador es el encargado de la parte lógica, el que tiene una unidad de control y ejecución para realizar las tareas. En las memorias RAM, se guardan los programas que se están ejecutando en el sistema operativo; el disco duro solo es utilizado para almacenar información en archivos.

Para citar un ejemplo, el microcontrolador **ATmega328**, creado por la empresa **Atmel**, es utilizado hoy en día en los kits de desarrollo **Arduino** y tiene las siguientes especificaciones:

- ▼ CPU: 8-bit AVR (frecuencia máxima de operación de 20 MHz)
- ▼ 32 KBytes de memoria flash, 1 KB EEPROM, 2 KB SRAM
- ▼ 23 puertos de entrada salida para propósitos generales
- ▼ Contadores/temporizadores flexibles
- ▼ Conversores analógico-digitales

Por este motivo, no deben confundirse un **microprocesador** y un **microcontrolador**. El μP (microprocesador) o CPU es,

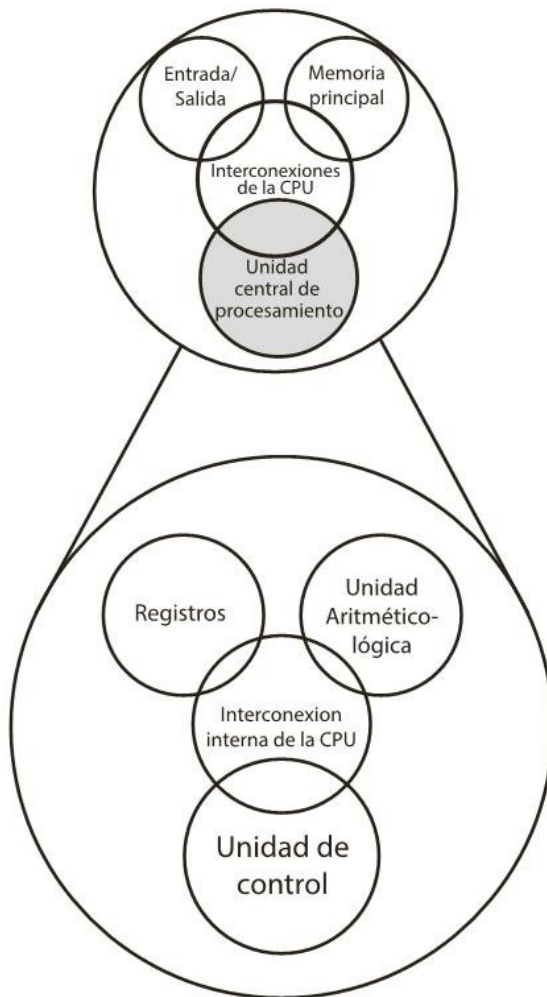
como su nombre lo indica, la unidad donde se lleva a cabo el procesamiento de la información, y su función es comandar la ejecución de los programas.

Por lo general, la CPU está constituida por uno o más de cada uno de los siguientes componentes:

- ▼ **ALU** (unidad aritmético-lógica): realiza las funciones de procesamiento de datos de la computadora, como por ejemplo suma, resta, multiplicación, operaciones lógicas entre distintos operandos, etc.
- ▼ **Unidad de Control**: ee encarga de buscar nuevas instrucciones, interpretarlas, y llevarlas a la ALU para ser ejecutadas.
- ▼ **Registros**: proporcionan almacenamiento interno a la CPU.
- ▼ **Interconexiones**: mecanismos que proporcionan comunicación entre los tres componentes nombrados antes.

HOY EN DÍA, AMBOS TIPOS DE ARQUITECTURAS HAN MEZCLADO SUS CAMINOS INICIALES Y PODEMOS ENCONTRAR QUE LAS CISC INCORPORAN CONCEPTOS DE RISC, Y VICEVERSA.





▲ Esquema interior de la Unidad Central de Procesamiento (CPU).

CUANDO HABLAMOS DE UN MICROCONTROLADOR O MCU, HACEMOS REFERENCIA A UN CHIP QUE CONTIENE LAS PARTES BÁSICAS DE UNA COMPUTADORA PERSONAL, TODO EN ESCALA REDUCIDA. ✉

www.reduccionpremium.com.ar

Arquitecturas x86

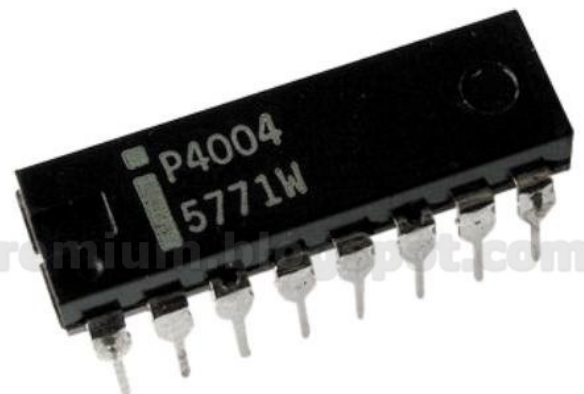
La **arquitectura x86** es la base de los procesadores de la aún corta historia de la informática. Son millones los microprocesadores vendidos que implementan esta arquitectura, y ha sido el comienzo del éxito para grandes compañías como **Microsoft** e **Intel**. x86 es un antes y un después en materia de tecnología y merece ser analizada con profundidad.

Comenzaremos con un breve repaso del contexto histórico en el que nació la familia de procesadores x86, una de las creaciones CISC más importantes en el mercado del hardware.

En 1971, la empresa Intel presentó el primer microprocesador comercial del mercado, el 4004. Estaba pensado para ser usado en calculadoras, con un modelo de bus de 4 bits; puede parecer poco, pero fue toda una novedad para esa época. A él, lo siguieron los Intel 8008 y 8080, también pensados para un uso de cálculo, pero con un set de instrucciones un poco más avanzado.

Luego de ver que estos modelos prometían, Intel comenzó un enorme proyecto con el que buscaba reinventar el mundo de los procesadores. A mediados de 1978 comenzó a venderse el Intel 8086, desarrollado un par de años antes, y que tenía retrocompatibilidad con el software de los anteriores 8008 y 8080. Formado por transistores de 3 micras (alrededor de 135 veces más grandes que los existentes hoy de 22 nanómetros), tenía una frecuencia máxima de operación de 4.77 MHz.

Tomando las quejas de los consumidores como un nuevo reto, Intel presenta, en 1979, el **8088**, una versión que tenía casi todas las características del 8086, pero con un precio reduci-



▲ Imagen del primer microprocesador comercial de Intel, el 4004.



Imagen del Intel Pentium IV que arrasó en los años 2000.

do. Para este lanzamiento, contó con IBM como su aliada más importante. La IBM PC 5150, considerada la primera computadora personal o PC (*Personal Computer*) que se vendió en forma masiva en la historia, utilizó un 8088 en vez de otros procesadores de la competencia, en especial por tres razones:

- ▼ Se podía conseguir fácilmente debido a su **alta disponibilidad**.
- ▼ Ofrecía diversas facilidades a la hora de programar.
- ▼ Como ya se ha mencionado, tenía un **costo reducido**.

Este equipo de IBM, si bien tenía un precio aproximado de 3000 dólares, obtuvo un éxito de ventas enorme desde su lanzamiento en agosto de 1981.

En este contexto, Intel centra como su principal producto una nueva familia de chips: los **procesadores x86**, que fueron definidos así por utilizar un juego de instrucciones que, aunque ha evolucionado con el correr del tiempo, sigue utilizando muchas de las características originales.

Luego del 8088, le siguieron los 80186, 80286, 80386 y 80486, más conocidos como 186, 286, 386 y 486, respectivamente, que continuaron con el éxito. Después de los 80X86, Intel comenzó con los populares **Pentium**, el primero en 1993, seguido de **Pentium Pro**, **Pentium II**, **Pentium III** y **Pentium 4**. Hasta los **Ivy Bridge** de hoy

conservan la estampilla Pentium en alguno de sus modelos de menores prestaciones.

La considerada principal arquitectura CISC de la historia, ha tenido competencia con Motorola, DEC y, sobre todo IBM, que han presentado procesadores CISC a lo largo de la historia.

Parte del éxito que hoy en día tiene Intel fue gracias a que tomó un do-

ble papel ya que, además de crear y diseñar sus procesadores, **también vende las licencias** a otros fabricantes. El ejemplo más claro de esta venta de licencia es con **AMD**, que, desde hace ya un tiempo, tiene un acuerdo con Intel y le permite fabricar chips con el mismo juego de instrucciones, que los hace compatibles en software, incluso con un hardware distinto.



Logo tradicional de la empresa Intel.

DENTRO DE TODO MICROCONTROLADOR,
 EXISTE UN MICROPROCESADOR. SE PODRÍA
 MENCIONAR COMO EL CEREBRO DEL
 MICROCONTROLADOR.





ELEMENTOS DE UN MICROCONTROLADOR

UN MICROCONTROLADOR INTEGRA EN UN CHIP UN MICROPROCESADOR,
MEMORIAS DE DATOS Y DE PROGRAMA, UNIDADES DE ENTRADA /
SALIDA Y OTROS DISPOSITIVOS PERIFÉRICOS, COMO TEMPORIZADORES.



E

El **microcontrolador** interactúa con el mundo exterior: recibe datos desde él y también se los envía luego de procesarlos. Este intercambio se realiza a través de los puertos de **entrada y salida (E/S)** asociados a los pines del encapsulado del microcontrolador. Para interactuar con el mundo exterior, en los pines de entrada conectamos sensores y de esta forma es posible recibir los datos.

En los **pinos de salida**, conectamos actuadores para que el microcontrolador envíe órdenes y así podamos realizar ciertas acciones de control. Algunos **puertos de E/S** tienen características que les permiten manejar salidas con determinados requerimientos de corriente. Por lo general, a cualquier pin de E/S lo consideramos de propósito general, y es compartido con los pines de otros periféricos.

Podemos usar un pin con cualquiera de las características asignadas si lo configuramos por medio de los registros específicos. Una interrupción por hardware provoca un salto en una subrutina de software mediante un evento de hardware interno o externo. A diferencia de las interrupciones generadas por software, en estas, el microcontrolador nunca sabe en qué momento ocurrirá. Cuando sucede la interrupción por hardware, el microprocesador deja el programa en ejecución para atender la interrupción. Finalizada la interrupción, el microprocesador retoma el programa que estaba ejecutando y continúa con él sin alteraciones.

Según sea el origen de las interrupciones, las clasificamos en externas (por los periféricos) o internas (microprocesador). Las interrupciones externas se producen al aplicar un pulso o un estado lógico en un pin del microcontrolador y se denominan **INT**. El hardware interno del microcontrolador, convertidor AD, temporizadores, etc., genera las interrupciones

EN UNA INTERRUPCIÓN ENMASCARABLE, EL BIT DE HABILITACIÓN DEBE ESTAR ACTIVADO. LAS NO ENMASCARABLES OCURREN INMEDIATAMENTE. ☒

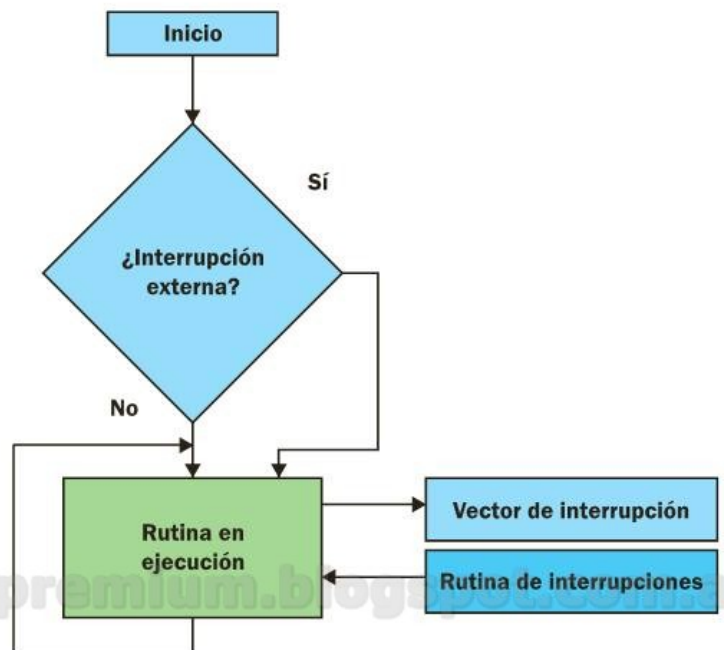


internas. De acuerdo con el control del microprocesador sobre la interrupción, las interrupciones son enmascarables o no enmascarables.

Un **PIC16F84A** de Microchip tiene cuatro mecanismos diferentes de **interrupción por hardware**. Dos de ellas son: **INT**, o interrupción externa en el **pin RB0/INT**, y **RBI**, o interrupción por cambio de estado en las líneas altas del Puerto B. Las otras dos: **TOI**, o interrupción por desbordamiento del

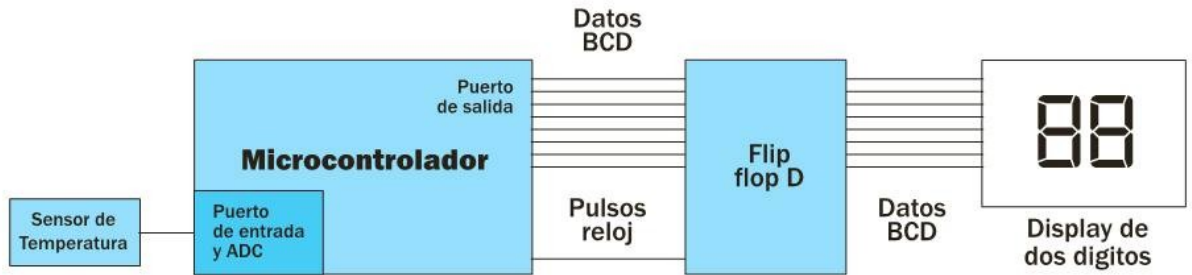
VECTOR DE INTERRUPCIÓN

Una interrupción se encuentra a partir de una posición de memoria fija que se conoce como **vector de interrupción**. De acuerdo con el tipo de microcontrolador utilizado, podríamos tener varios vectores de interrupción e, incluso, uno para cada dispositivo de hardware que solicite una interrupción. También podría existir un vector de interrupciones para una interrupción enmascarable, y otro, para una interrupción no enmascarable.



Esquema que nos permite observar gráficamente cómo se ejecuta el proceso de una interrupción externa ocasionada por un periférico.





Si conectamos un sensor de temperatura a un microcontrolador, tendremos un sistema para medir temperatura y leerla en un display.

TMRO, y EEI, o interrupción por finalización de escritura en EEPROM. Los mecanismos de interrupción utilizan dos bits de control: uno habilita la interrupción, y otro indica que se ha generado. También tenemos un bit adicional que nos permite deshabilitar todas las interrupciones.

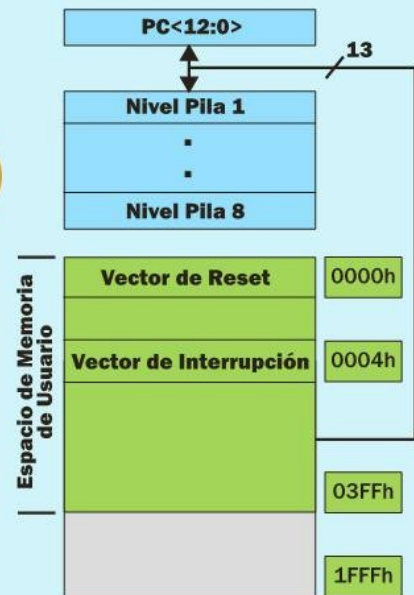
CPU

La **memoria de datos**, de tipo RAM, almacena datos temporales de propósito general y los registros específicos cuyos bits controlan el funcionamiento del microcontrolador. Contiene bancos de memoria con una cantidad de posiciones de datos de una determinada longitud en bits, por ejemplo 4 bancos de memoria de 128 posiciones de datos con una longitud de 8 bits cada una de ellas. Cada banco posee **registros de propósito general (GPR)** y **registros de funciones especiales (SFR)**. En los GPR, almacenamos datos que utiliza el programa en ejecución, mientras que los SFR contienen bits de control para el microprocesador y los periféricos.

Si los datos provienen del usuario, se almacenan en los GPR dentro de la memoria de datos para requerirlos cuando se



MAPA DE MEMORIA EN UN PIC



Mapa de memoria de programa de un microcontrolador PIC16F84A de Microchip y la posición de memoria donde se encuentra el vector de interrupción, en este caso, la posición 0004h, en hexadecimal.

En nuestro mapa de memoria, cuando ocurre una interrupción, la rutina que se está desarrollando en ese momento saltará a la posición 04h de la memoria de programa, lugar donde se encuentra el vector de interrupción. A partir de allí, se ejecutarán las instrucciones que encuentre, relacionadas con el tipo de interrupción solicitada, hasta alcanzar la instrucción retfie por medio de la cual la ejecución regresará hacia el punto de la rutina donde se generó esa interrupción.

Un microcontrolador, a diferencia de un microprocesador, es un sistema cerrado, ya que no es posible modificar sus componentes.

ejecuta una determinada instrucción. Los SFR ocupan las primeras posiciones de memoria de datos, relacionadas con el funcionamiento y la configuración del microprocesador. Cada registro SFR está asociado a una función general, y allí cada uno de los bits tiene una función específica.

El **registro de estado** o de **status** es un registro de control que almacena en su interior el estado de las operaciones aritméticas de la **ALU**, el estado del reset y los bits necesarios para seleccionar el banco de la memoria de datos adecuado. A cada bit del Registro de Estado, por ejemplo de 8 bits, se le asigna una función, y, al conjunto de bits, lo llamamos **palabra de estado**.

El registro de **contador de programa** (PC) pertenece a la memoria de datos y tiene N de bits para acceder a 2N posiciones de memoria, en las que se almacenan los datos en forma de grupos o palabras que se transfieren simultáneamente desde la memoria y hacia ella.

Debido a la gran cantidad de posiciones dentro de la memoria, es necesario un sistema que decodifique una posición específica de modo de direccionar hacia allí los datos de entrada o leerlos, seleccionando el estado adecuado para un pin específico. La capacidad de la memoria surge de multiplicar su número de posiciones por la longitud de cada palabra de datos.

Según el método de acceso a los datos almacenados, tenemos memorias de acceso aleatorio y de acceso secuencial. En acceso aleatorio, memorias RAM y ROM, accedemos a cualquier dato con igual tiempo de acceso independientemente de la dirección. Las memorias **RAM** (*Random Access Memory* o memoria de acceso aleatorio) pertenecen al grupo de las memorias volátiles.

Una memoria **ROM** (*Read Only Memory* o memoria de solo lectura) se escribe en fábrica y no podemos modificar ni borrar sus datos. Una memoria flash es una **EEPROM** (*Electrically Erasable PROM* o memoria borrable y regrabable eléctricamente) más veloz y con menor consumo, que se utilizan por su bajo costo en dispositivos de almacenamiento USB y tarjetas de memoria.

Un microprocesador consta de tres partes: **CPU** (unidad central de procesamiento) reconoce y ejecuta las instrucciones, interfaces de entrada/salida que comunican la CPU con el exterior y en la memoria almacenamos instrucciones y datos. La **PCU** (*Process Control Unit* o unidad de control de proceso) o microprocesador es el cerebro de sistemas computacionales o basados en microcontroladores. En una PCU, el elemento principal es la **ALU** (*Arithmetic Logic Unit* o unidad aritmético-lógica), donde realizamos las operaciones lógicas y

UN REGISTRO ES UN
CONJUNTO DE ELEMENTOS
BÁSICOS DE MEMORIA QUE
ALMACENAN INFORMACIÓN
POR UN TIEMPO HASTA QUE
SEA SOLICITADA.

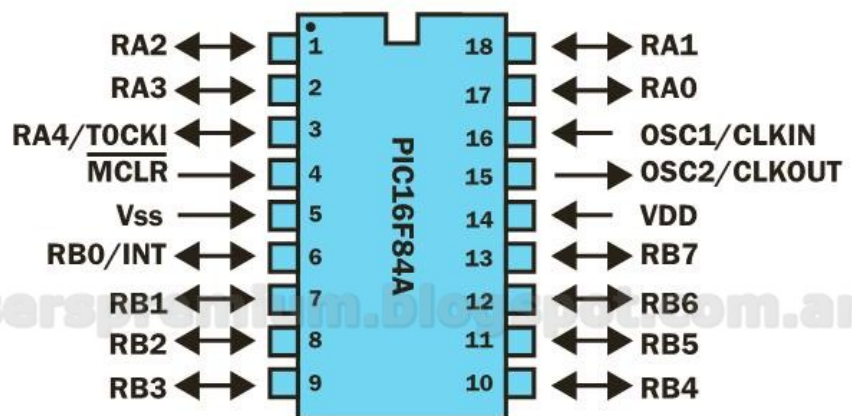


aritméticas requeridas por los procesos que se ejecutan en el microprocesador.

Los **registros de almacenamiento temporal** de datos almacenan datos internos de la CPU durante un proceso, y la unidad de control es responsable de efectuar la temporización y la secuencia de las operaciones realizadas.

La ALU almacena temporalmente sus resultados en un registro especial o acumulador mientras se ejecuta un proceso. El registro de estado contiene la información del resultado de la última operación de la ALU, y unos bits individuales o banderas nos indican el estado de la última operación, por ejemplo si se ha generado acarreo en la suma de dos números.

En el **contador de programa** se almacena la dirección de la



El PIC16F84 es un microcontrolador CMOS de 8 bits de Gama Media con 13 pines de entrada/salida organizados en dos puertos, A y B.

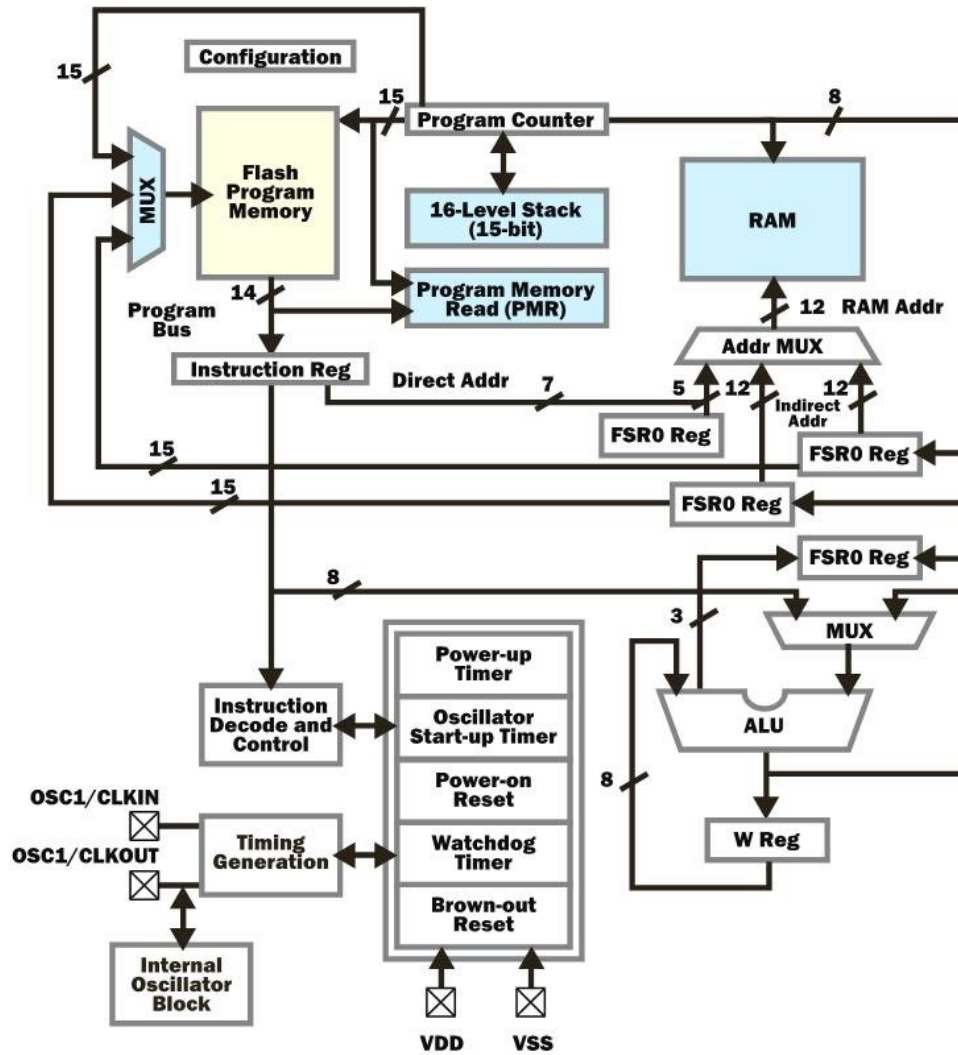


Diagrama en bloques del microcontrolador PIC16F1512 de Microchip. Es un microcontrolador flash de 28 pines, tecnología XLP y CPU RISC de alta performance.



▲ Sensor de distancia por ultrasonido que mide distancias de 2 cm a 3 m y requiere un pin I/O para conectarlo a un microcontrolador.

memoria correspondiente a la próxima instrucción por ejecutar en la ALU. El registro de instrucciones almacena el código del programa, y el registro de direccionamiento, la dirección de los datos.

Los **registros de propósito general** almacenan temporalmente datos o direcciones cuando se ejecuta un programa. El registro de apuntador de la pila es un registro especial que almacena la dirección del tope de la pila de la memoria RAM, un área especial que guarda los valores del contador de programa durante una subrutina.

Por medio de los **bloques del microprocesador**, se busca una instrucción y se la ejecuta para obtener un deter-

minado resultado. El dato es indicativo de la dirección donde se encuentra la próxima instrucción por ejecutar, que se coloca en el bus de direcciones. Posteriormente la **unidad de control**, mediante el bus de control, envía la señal para leer la dirección de la memoria de programa. Los datos de la dirección de memoria de programa se aplican al bus de datos y procesan para recuperar el código de la instrucción. Almacenamos el código de la instrucción en el registro de instrucciones, y los datos, en el registro de almacenamiento temporal. El contador de programa se incrementa a la siguiente instrucción, y la ALU ejecutará la operación indicada por la unidad de

control. Finalmente, el resultado se almacenará en el registro interno (acumulador) o en alguna posición dentro de la memoria de datos.

Una **ALU** estándar realiza operaciones de suma y resta de números enteros, lógicas **AND**, **OR** y **NOT** y operaciones relacionadas con el desplazamiento de bits. La **unidad de punto flotante**, o **FPU**, es responsable por las operaciones sobre los números decimales. Un bus facilita que las señales binarias se desplacen de un sector a otro mediante varios conductores por donde circulan señales eléctricas.

Cuando un chip desea enviar sus datos por el bus, al tratarse de un medio compartido, espera su turno hasta que el bus sea liberado. Un microprocesador tiene tres tipos de buses: de datos, de direcciones y de control, cada uno de ellos con una función específica.

Los datos asociados a las funciones de la CPU viajan por el bus de datos hacia/desde la CPU y la memoria o interfaces entrada/salida. Cada conductor transporta un dígito binario.

A mayor cantidad de conductores, más larga será la palabra en el microprocesador. El bus de direcciones trae señales que indican dónde están los datos, en una dirección de memoria o los puertos de entrada/salida.

Las señales referidas al control viajan por el bus de control y transporta señales de reloj que sincronizan acciones del microprocesador.

DSP es el acrónimo de **procesadores de señal digital** (*Digital Signal Processor*) que se encargan de utilizar un microprocesador optimizado para procesar digitalmente una señal y algoritmos que realizan gran número de operaciones matemáticas aplicadas, rápida y en forma repetida, a una serie de muestras de datos.

UN MICROCONTROLADOR
ES UN CHIP DE USO ESPECÍFICO,
EN EL QUE SE HAN INTEGRADO
EL MICROCONTROLADOR, LA
MEMORIA Y LAS ENTRADAS/SALIDAS.



PUERTOS DE I/O

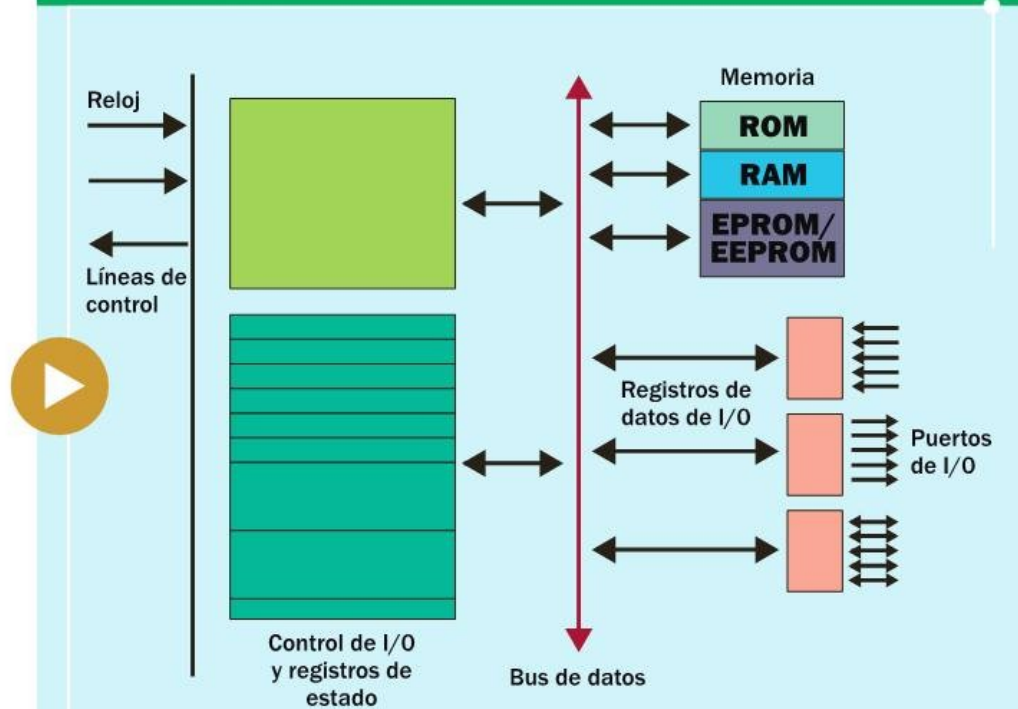
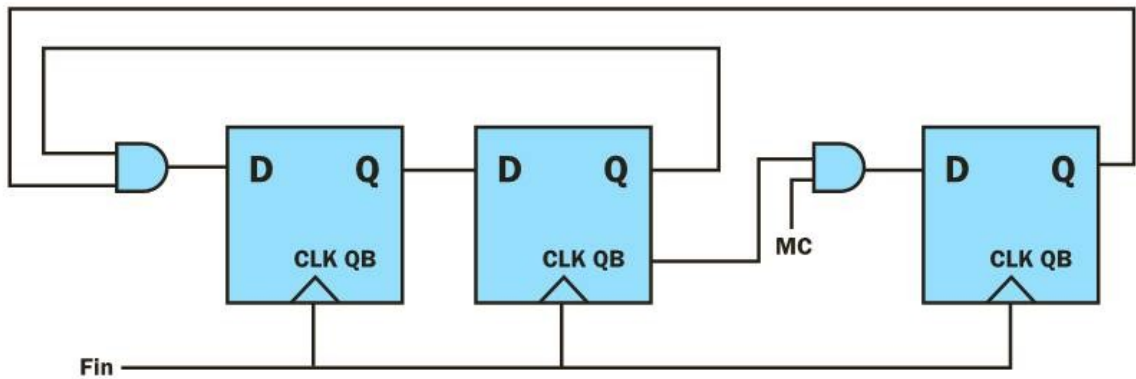


Diagrama en bloques de un microcontrolador que integra, en un chip, un microprocesador y otros dispositivos, como memorias, interfaces I/O y otros periféricos, por ejemplo temporizadores.

En general, los puertos de I/O tienen 8 líneas para transportar una palabra de datos de 8 bits. En una palabra de 16 bits, utilizamos dos puertos, los puertos pueden ser solo de entrada o de salida, o programables para funcionar como entrada y salida. Los microcontroladores PIC16C6/7, Motorola 68HC11 e Intel 8051 tienen un bus de datos de 8 bits; 16 bits, el 68HC16 de Motorola, y 32 bits, el PIC32MX220F032B-I/SP.





Podemos construir una memoria elemental de un bit utilizando los flip-flops tipo D, donde cada uno almacena un dígito binario, 0 o 1.

Buses

Debemos tener en cuenta que los buses se caracterizan por el número de bits que pueden transmitir en un determinado momento. Por ejemplo, un equipo con un bus de 8 bits de datos, transmite 8 bits de datos cada vez, mientras que uno con un bus de 16 bits de datos es capaz de transmitir 16 bits de datos en forma simultánea.

Como el bus es parte integral de la transmisión interna de datos y como los usuarios suelen tener que añadir componentes adicionales al sistema, la mayoría de los buses de los equipos informáticos pueden ampliarse mediante uno o más zócalos de expansión. Al agregarse estas placas permiten la conexión eléctrica con el bus y se convierten en parte efectiva del sistema.

De esta forma, el bus se refiere al camino que recorren los datos desde una o varias fuentes hacia uno o varios destinos y es una serie de hilos contiguos. En el sentido estricto que corresponde a la definición del concepto, esto sólo se aplica a la interconexión entre el procesador y los periféricos.

Así, un bus es simplemente un conjunto compartido de pistas trazadas en la placa de circuito principal, al que se conectan todas las partes que controlan y forman la computadora.

Cuando un dato pasa de un componente a otro, viaja a lo largo de este camino común para alcanzar su destino. Cada chip de control y cada byte de memoria de la computadora están conectados directa o indirectamente al bus. Cuando un nuevo componente se inserta en uno de los conectores de expansión, queda unido directamente al bus, convirtiéndose en un objeto más de la unidad completa.

Cualquier información que entra o sale de un sistema ordenador se almacena temporalmente en al menos una de las distintas localizaciones que existen a lo largo del bus. La mayor parte de las veces el dato se sitúa en la memoria principal, que en la familia PC está formada por miles de posiciones de memoria de



Los microcontroladores utilizan la arquitectura Harvard, caracterizada por separar físicamente las instrucciones de los datos.

LA OPERACIÓN DE BÚSQUEDA DE UNA INSTRUCCIÓN ES FUNDAMENTAL Y CONSISTE EN ANALIZAR EL DATO CONTENIDO EN EL CONTADOR DE PROGRAMA.



8 bits. Pero algún dato puede acabar en un puerto, o registro, durante unos instantes, mientras espera que la CPU lo envíe a una posición adecuada.

Los buses más comúnmente conocidos son:

▼ **Bus de datos** (*Data Bus*): a través del bus de datos circulan los datos entre los elementos componentes del ordenador. Estos datos pueden ser de entrada o salida respecto a la CPU. Comunican a ésta con la memoria y con los controladores de entrada/salida. Este bus también es conocido como bus de entrada/salida.

▼ **Bus de direcciones** (*Address Bus*): la información que circula en este bus son direcciones de posiciones de memoria. El sentido en que circulan estas direcciones es siempre desde la CPU hacia la memoria principal.

▼ **Bus de control** (*Control Bus*): a través de este bus circulan las señales de control de todo el sistema. Este bus, al contrario que el de direcciones, es de entrada y salida, debido a que la CPU envía señales de control a los dispositivos periféricos y estos envían a la CPU información sobre su estado de funcionamiento.

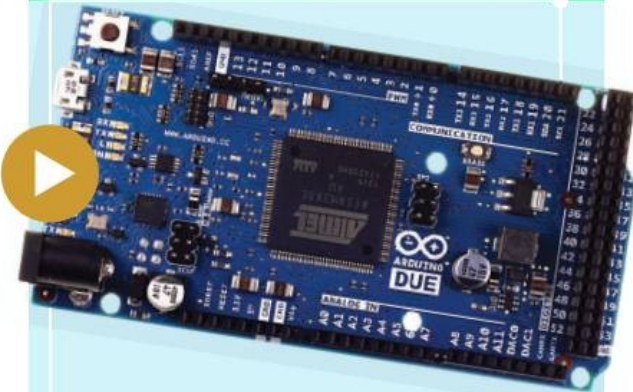
Consideremos que el bus de direcciones, se encuentra vinculado al bloque de control de la CPU para tomar y colocar datos en el sub sistema de memoria durante la ejecución de los procesos ejecutados por el procesador,

Por otra parte, el bus de control se encarga de realizar el transporte de las señales de estado de las operaciones efectuadas por el CPU con las demás unidades. Por ejemplo, una tarjeta madre tipo ATX tiene tantas pistas eléctricas destinadas a buses, como anchos sean los canales de buses del microprocesador de la CPU: de esta forma encontramos 64 para el bus de datos y 32 para el bus de direcciones.

El ancho de canal explica la cantidad de bits que pueden ser transferidos simultáneamente. De esta forma, el bus de datos transfiere 8 bytes a la vez.

Para el bus de direcciones, el ancho de canal explica así mismo la cantidad de ubicaciones o direcciones diferentes que el microprocesador puede alcanzar. Esa cantidad de ubicaciones resulta de elevar el 2 a la 32 potencia. 2 porque son dos las señales binarias, los bits 1 y 0; y 32 potencia porque las 32 pistas del bus de direcciones son, en un instante dado, un conjunto de 32 bits. Así, el canal de direcciones del microprocesador para una PC-ATX puede direccionar más de 4 mil millones de combinaciones diferentes para el conjunto de 32 bits que corresponden a su bus.

ARDUINO DUE



DUE es una placa de desarrollo basada en el microcontrolador CortexM3 ARM de 32 bits que incrementa el poder de procesamiento para todos los usuarios de Arduino.

Contiene 54 pines de entrada/salida, de los cuales podemos usar 12 como salidas PWM, 12 entradas analógicas, 4 UART, un reloj a 84 MHz y dos convertidores DA. Para compilar el código en el procesador ARM, necesitamos la última versión del IDE Arduino. Opera con 3,3 Volt y no es compatible directamente con la mayoría de los shields para Arduino que funcionan con 5 volts. Respecto de la capacidad de memoria, la memoria flash es de 512 KB para el usuario y la SRAM (RAM estática) de 96 KB.

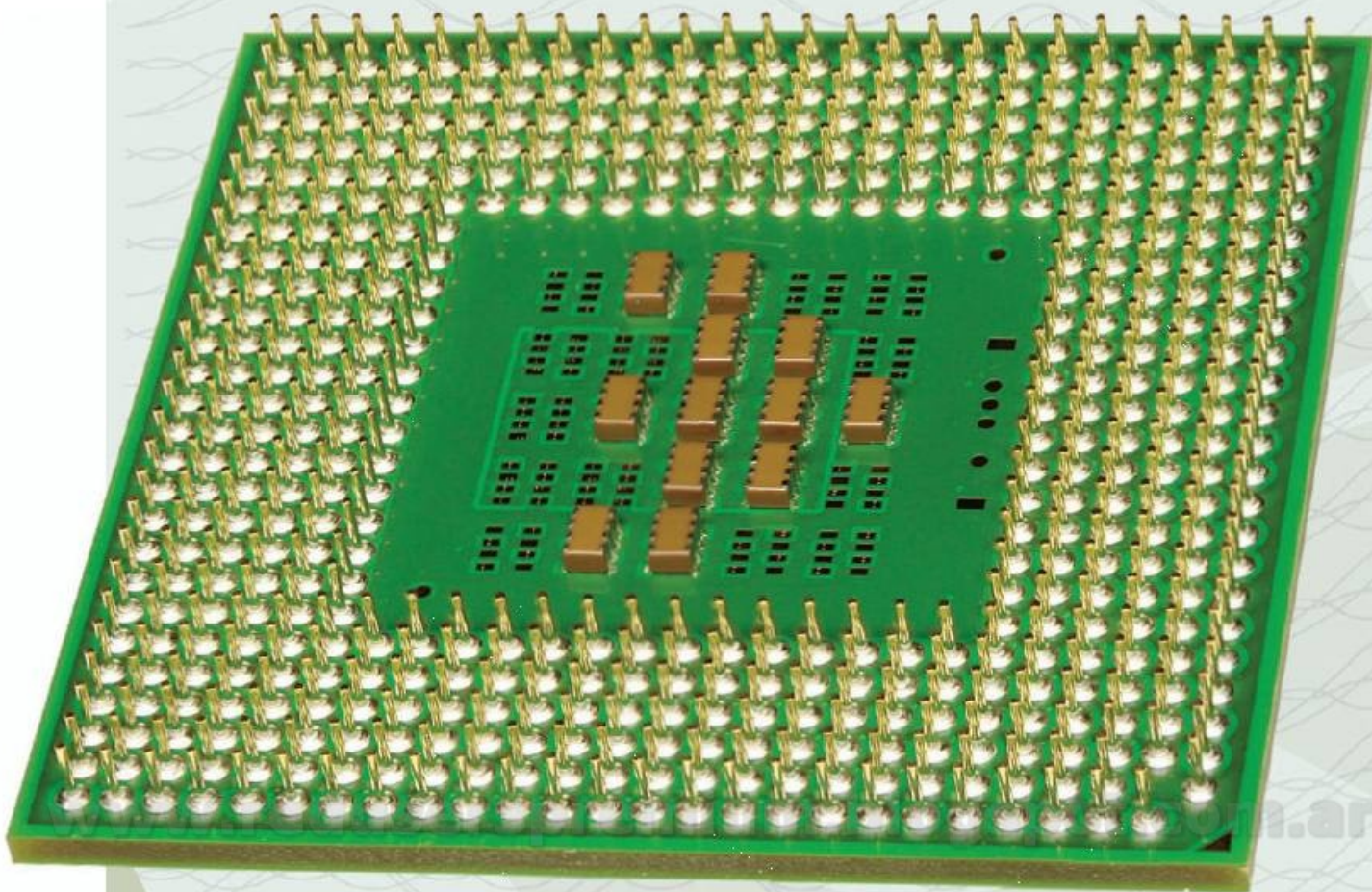
PODEMOS COMPARTIR UN
BUS ENTRE DISTINTOS CHIPS
SIMPLIFICANDO EL DISEÑO Y
MEJORANDO LA CONFIABILIDAD
FRENTE A CONEXIONES
SEPARADAS.





LENGUAJE ENSAMBLADOR

EL LENGUAJE ENSAMBLADOR CONSTA DE UNA SERIE DE NEMÓNICOS O ABREVIACIONES QUE REPRESENTAN UNA INSTRUCCIÓN EN BINARIO. ESCRIBIMOS UN PROGRAMA MÁS COMPRESIBLE QUE CON LENGUAJE MÁQUINA.





P

Podemos programar un microprocesador utilizando un lenguaje máquina, un lenguaje ensamblador o un lenguaje de alto nivel. En el lenguaje en código de máquina, cada instrucción está escrita en **código hexadecimal** específico para ese microprocesador. Todos los programas escritos en lenguaje ensamblador o en alto nivel se convierten a código máquina con un software específico.

Mediante un lenguaje de alto nivel, accedemos a la escritura de programas en un lenguaje más cercano a la interpretación humana.

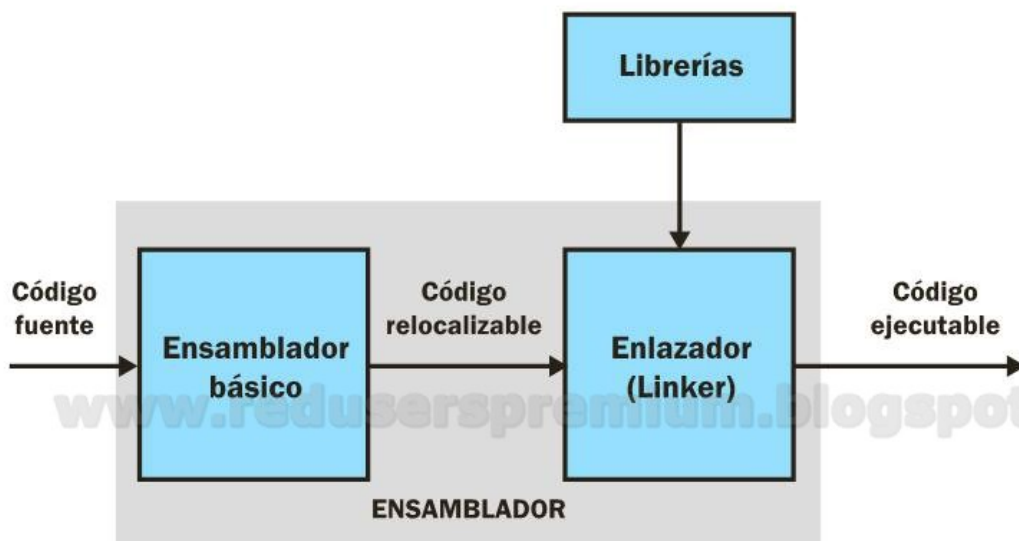
Para ejecutar estos programas en un microprocesador, los traducimos a lenguaje máquina con un programa intérprete o compilador. El **ensamblador (assembler)** es un tipo de lenguaje intermedio entre el lenguaje máquina y el lenguaje en alto nivel. En este caso, no recurrimos al set de instrucciones de un microprocesador en particular y su equivalencia hexadecimal (lenguaje máquina), sino a la equivalencia del set de instrucciones a nemónicos (abreviaturas de una instrucción binaria) correspondientes. Entonces, a partir del archivo objeto, tomamos cada una de las expresiones en nemónico y la sustituimos por un código binario.

Un programa tiene encabezado, definición de constantes, definición de origen, bloque de instrucciones y final del programa. En el encabezado, definimos algunas cuestiones de tipo general, directivas, que modifican el

funcionamiento del ensamblador. En los microcontroladores PIC, al emplear MPASM, recurrimos a la directiva **list** para indicar el modelo de microcontrolador.

En la parte de **constantes**, definimos las constantes que necesariamente serán reconocidas en los demás puntos de nuestro programa. Si usamos el nombre de la constante asociada en lugar del valor en hexadecimal requerido, simplificamos la lectura del programa. La definición del origen del programa explicita el sitio de la memoria donde se almacenará el código binario luego del ensamblado. Definiendo **org 0** (del inglés *origin*) le indicamos al ensamblador que debe almacenarse desde la línea 0 de la memoria de programa.

LA PROGRAMACIÓN DE MICROPROCESADORES DE DISTINTOS FABRICANTES QUE UTILIZAN EL MISMO CÓDIGO MÁQUINA ES INCOMPATIBLE ENTRE ELLOS.



En la estructura de un ensamblador, encontramos bloques muy importantes para cumplir con la función de convertir el código fuente en código ejecutable.



Las **instrucciones de programa** constan de una etiqueta seguida del código de operación, los operandos y los comentarios (;). El final de programa se indica mediante una instrucción sencilla como **end** en el ensamblador MPASM de la familia PIC Microchip.

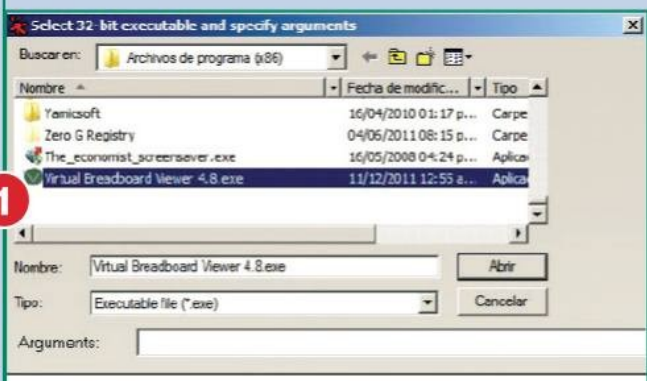
Etiqueta identifica una posición de memoria del microcontrolador e indica puntos específicos del programa, por ejemplo Inicio. Comenzamos a escribir en la columna **Etiqueta** respetando la máxima longitud en caracteres.

La columna **instrucción** hace referencia a una operación básica que puede ejecutar el microcontrolador. Un ejemplo sería **MOVLW**. La operación MOVLW significa **almacenar un valor literal en el registro W**, y W es un registro en la memoria del microcontrolador.

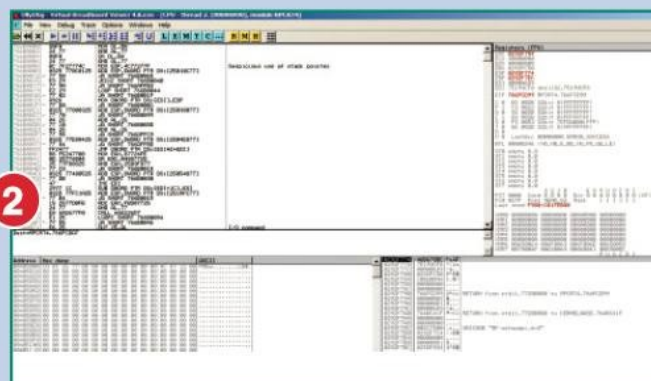
La columna **operando** incluye un elemento que utiliza la instrucción. Las instrucciones más simples no utilizan ningún operando, pero no todas las instrucciones son simples, algunas pueden utilizar un operando, y las más complejas, dos operandos. El primer operando se denomina **operando fuente** y el segundo, que complementa al anterior, se designa como **operando destino**. Tenemos que respetar la jerarquía definida para encaminar la información desde el operando fuente hacia el operando destino.

Comentario es una columna con formato de texto donde podemos escribir nuestros comentarios de modo de documentar el programa. Puesto que no interesan en la ejecución en lenguaje ensamblador, los ignoramos con el carácter (;) al comienzo del comentario. Recordemos que el código nemónico es específico para el microcontrolador por utilizar y no podemos emplearlo en otras familias.

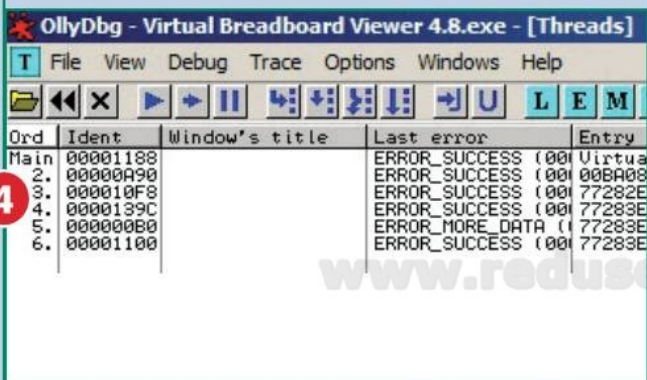
REGISTROS Y MEMORIA PASO A PASO



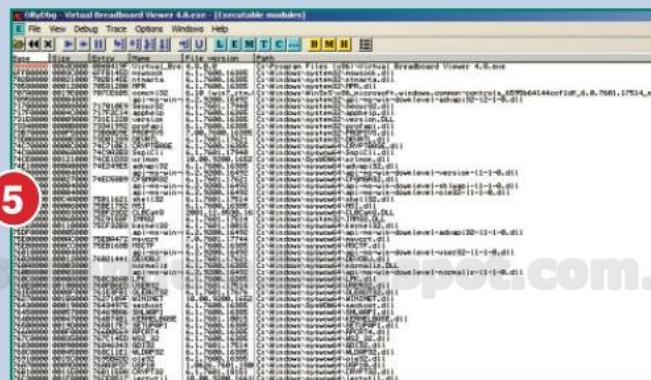
Una vez que descargamos e instalamos **OllyDBG** (www.immunityinc.com/products-immdbg.shtml), abrimos la opción File para encontrar y seleccionar el archivo .EXE para analizar.



Una vez seleccionado el archivo con extensión .EXE, tenemos la pantalla de la CPU y sus registros. Si presionamos **F9**, iniciamos la depuración (**Run**).



Con **ALT+T**, seleccionamos la vista de los Threads producto de la depuración. También accedemos a ella seleccionando el icono correspondiente.



La opción **ALT+E** nos muestra la vista de los módulos ejecutables, en inglés, *Executable Modules*. Podemos pausar la depuración presionando **F12**.

Una vez obtenido el **código máquina**, tenemos que transferirlo de alguna forma a la memoria de programa del microcontrolador. Bloques muy importantes en un ensamblador genérico son el ensamblador básico, el enlazador (linker) y el control de librerías.

El **enlazador básico** genera un **archivo binario relocable** para almacenar en alguna zona de la memoria del microcontrolador. A continuación el linker, o enlazador, crea un archivo binario ejecutable para implementar directamente en el microcontrolador. El control de librerías genera archivos binarios que pueden ser unidos con otros bloques de código binario ya existentes y nos facilita la reutilización de código generado en otros proyectos, simplificando así el desarrollo de nuestros programas.

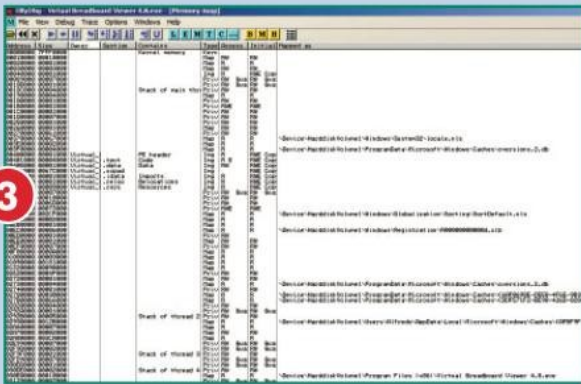


PICKIT 2 y PICKIT 3 son herramientas proporcionadas por Microchip para grabar el programa que se ejecutará en un microcontrolador PIC; estos incluyen el IDE MPLAB.

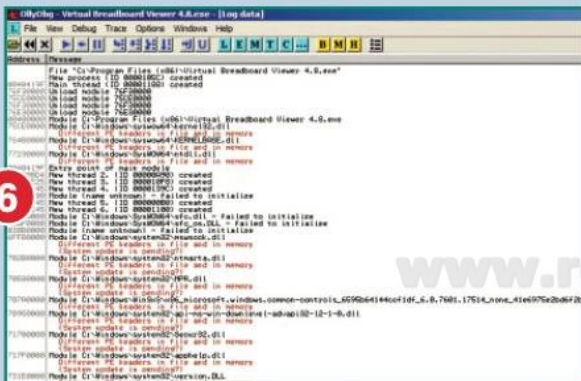
DEENSAMBLADORES

Un desensamblador es un software que traduce el lenguaje de máquina a lenguaje ensamblador, la operación inversa de la que hace el ensamblador cuando transforma el código binario en instrucciones básicas de la PC en la que se ejecuta. El programa desensamblador que se usará, por trabajar con lenguaje máquina y ensamblador, es aplicable solo al microprocesador utilizado, lo que nos permite reconocer las instrucciones del código binario a una arquitectura de la PC y al sistema operativo en uso. En cualquier depurador, observaremos el desensamblado del programa bajo depuración.

UTILIZAMOS UN PROGRAMA PARA
 TRADUCIR UN ARCHIVO FUENTE
 EN ENSAMBLADOR A UN ARCHIVO
 OBJETO DE EJECUCIÓN EN EL
 MICROCONTROLADOR.



Al hacer clic en la opción **View**, y dentro de ella, en **Memory Map (ALT+M)**, obtenemos el mapa con el uso de la memoria consecuencia de la depuración.



La vista log data nos muestra los datos involucrados. Podemos finalizar la depuración presionando **ALT+F2** o en el icono correspondiente,



Una vez finalizado el ensamblado, tendremos en la misma carpeta del archivo .ASM un archivo .HEX con el mismo nombre que el archivo fuente. El archivo .HEX es código máquina con instrucciones en hexadecimal para grabar la memoria de programa del microcontrolador.

El archivo .HEX, sumamente importante, contiene todos los errores encontrados en el proceso de ensamblado del código fuente; .LST es el archivo de listado que contiene una lista del proceso de ensamblado que se realizó e incluye una copia del código fuente. También contiene los errores generados, el código de máquina y las particularidades de uso de la memoria, entre otros aspectos.

Escribimos un programa mediante un simple editor de texto, con el entorno **IDE MPLAB** de Microchip, o con otras herramientas libres. Para grabar el microcontrolador, utilizamos la herramienta libre IC-PROG, PICKit 2 o PICKit 3 de Microchip u otras herramientas específicas de acuerdo con el microcontrolador utilizado.

Los usos actuales del assembler incluyen sistemas embebidos (impresoras, cámaras, automóviles, armamento, juguetes, etc.). Además, aplicaciones en tiempo real (industria y manufactura, adquisición de datos, robótica) y transporte marítimo y aéreo. En tecnología espacial, graficación, multimedia, cine y videojuegos, procesamiento digital de señales (DSP), voz e imágenes.



Si bien se trata de un software tipo shareware, antes de descargarlo y utilizarlo debemos registrarnos en www.ollydbg.de.

SOFTWARE IDE ENSAMBLADOR

Turbo Assembler (TASM) es un paquete ensamblador de alto nivel para uso en computadoras personales. Se complementa con el enlazador **Turbo Linker** y el depurador **Turbo Debugger**. Es un desarrollo de Borland. Otro ensamblador de alto nivel es **Microsoft Macro Assembler (MASM)**, originalmente utilizado en el desarrollo del sistema operativo MS-DOS. Aunque ya no se comercializa, continúa el soporte por la gran base de código en Assembler existente dentro de Microsoft. Por ello se ha transformado en el ensamblador con mejor soporte aun con microprocesadores de 64 bits.

EL PROCESO DE CONVERTIR EL
CÓDIGO FUENTE EN LENGUAJE
MÁQUINA SE DENOMINA ENSAMBLADO
Y AL RESPONSABLE LO LLAMAMOS
ENSAMBLADOR.





Para funcionar, Immunity Debugger requiere que instalemos el lenguaje de programación de alto nivel Python en su versión 2.7.1.

Visualización de registros y memoria

Un **depurador** (*Debugger*) es un tipo de software que utilizamos para probar y eliminar los errores en un determinado programa. Entre las funciones avanzadas, podemos ejecutar un programa paso a paso, detener el programa y seguir determinadas variables de interés.

Los depuradores pueden modificar un programa mientras se está ejecutando. La ejecución se realiza normalmente hasta que el depurador detiene su ejecución, y nos permite analizar lo ocurrido.

OllYDbg es una herramienta depuradora de código ensamblador de 32 bits, shareware, para sistemas operativos Microsoft Windows. **Immunity Debugger** es otra herramienta para explorar software, analizar malware y realizar ingeniería inversa de archivos binarios.

Esta herramienta, de la compañía Immunity, incluye tanto una interfaz gráfica de usuario (GUI) como una línea de comandos. Podemos descargar esta aplicación visitando el sitio

SOFTWARE ENSAMBLADOR LIBRE

Si bien tenemos una gran cantidad de software libre, tanto un entorno de desarrollo IDE como los programas para transferir el código en lenguaje máquina a la memoria de programa del microcontrolador son de aplicación exclusiva para una familia determinada. Por ejemplo, para la familia Microchip tenemos el ensamblador MPASM incluido en el entorno de desarrollo IDE MPLAB, que podemos descargar desde el sitio web **www.microchip.com**, y el programa IC-Prog para enviar el código máquina a la memoria del PIC, descargable desde la página web oficial **www.ic-prog.com**.



Una gran parte de las aplicaciones electrónicas que poseen sistemas embebidos utilizan lenguaje ensamblador que incluye el procesamiento digital de imágenes.



web que encontramos en la dirección www.immunityinc.com/products-immdbg.shtml después de completar el formulario de registraci3n.

Con **IDA Debugger** depuramos archivos que pueden estar ejecutándose en cualquiera de las tres plataformas: Windows, Linux y Mac OS X.

De forma directa y gracias al poder de depuraci3n de servidores remotos, es posible depurar y ejecutar desde cualquier plataforma. Es f3cil de conectar



El programador JDM es sencillo y f3cil de construir, funciona con la mayoría de los PIC y tambi3n con alg3n AVR. Lo conectamos a la PC mediante puerto serie.

CON ESTAS HERRAMIENTAS,
PODEMOS VISUALIZAR LOS
REGISTROS Y LA MEMORIA
DE UNA PC CUANDO
REALIZAMOS
LA DEPURACI3N
DE UN PROGRAMA.



tanto a procesos locales como a procesos remotos, y nos proporciona soporte para sistemas de 64 bits.

IDA tiene la característica de la depuraci3n instantánea, y no es necesario completar el análisis para iniciar la depuraci3n. Tambi3n podemos depurar distintos archivos x86 Windows PE, AMD Windows PE y x86 Linux ELF tanto de manera local como en forma remota. Para realizar la depuraci3n local, IDA Debugger facilita una simple consola programada en lenguaje C compilada bajo Windows.

IDA Debugger puede ser descargada desde el sitio web del fabricante en www.hex-rays.com/products/ida/debugger. en este sitio encontraremos un listado detallado de las características de la aplicaci3n así como tambi3n una colecci3n de manuales que nos ayudarán a dar los primeros pasos en su uso.

Tambi3n encontraremos una tabla que nos informa sobre el uso del debugger en las distintas plataformas soportadas, en ella se muestran tanto los sistemas de escritorio como los sistemas operativos móviles

¿TE RESULTA ÚTIL?

Lo que estás leyendo es el fruto del trabajo de cientos de personas que ponen todo de sí para lograr un mejor producto. Utilizar versiones "pirata" desalienta la inversi3n y da lugar a publicaciones de menor calidad.

NO ATENTES CONTRA LA LECTURA. NO ATENTES CONTRA TI. COMPRA S3LO PRODUCTOS ORIGINALES.

Nuestras publicaciones se comercializan en kioscos o puestos de voceadores; librerías; locales cerrados; supermercados e internet (usershop.redusers.com). Si tienes alguna duda, comentario o quieres saber m3s, puedes contactarnos por medio de usershop@redusers.com



PRÓXIMA ENTREGA



Microcontroladores PIC

EN LA SIGUIENTE ENTREGA VEREMOS EN DETALLE LOS MICROCONTROLADORES PIC, CONOCEREMOS SU EVOLUCIÓN Y DESCRIBIREMOS EL FUNCIONAMIENTO DE ESTA TECNOLOGÍA. ANALIZAREMOS EL PIC 16, 18 Y 32.



TÉCNICO en ELECTRÓNICA

CONCEPTOS FUNDAMENTALES Y PRÁCTICA PROFESIONAL



PROFESORES EN LÍNEA

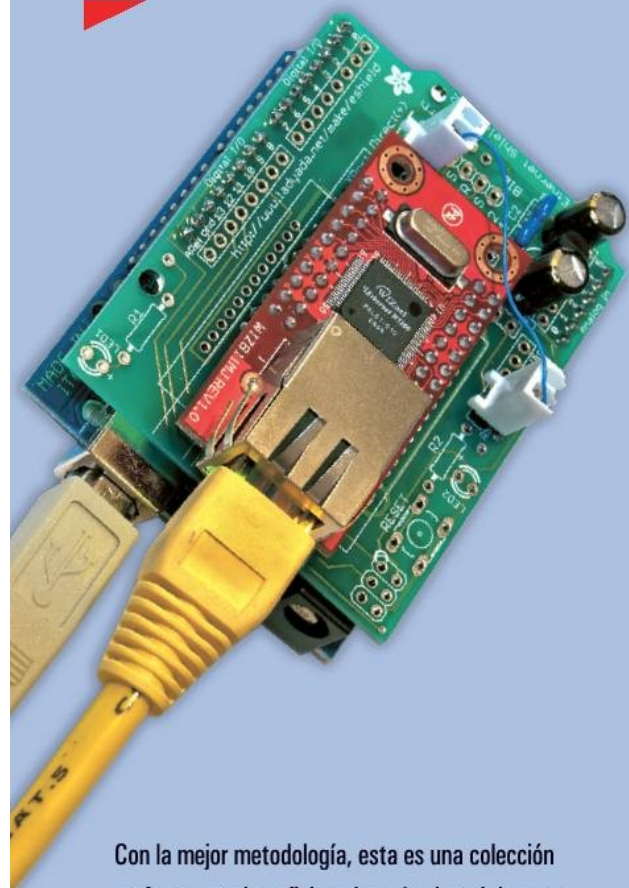
profesor@redusers.com

SERVICIOS PARA LECTORES

usershop@redusers.com

SOBRE LA COLECCIÓN

CURSO VISUAL Y PRÁCTICO QUE BRINDA CONCEPTOS Y CONSEJOS NECESARIOS PARA CONVERTIRSE EN UN TÉCNICO EXPERTO EN ELECTRÓNICA. LA OBRA INCLUYE RECURSOS DIDÁCTICOS COMO INFOGRAFÍAS, GUÍAS VISUALES Y PROCEDIMIENTOS REALIZADOS PASO A PASO PARA MEJORAR EL APRENDIZAJE.



Con la mejor metodología, esta es una colección perfecta para los aficionados a la electrónica que deseen profesionalizarse y darle un marco teórico a su actividad, y para todos aquellos técnicos que quieran actualizar y profundizar sus conocimientos.

CONTENIDO DE LA OBRA

13/24

- 1 ▲ INTRODUCCIÓN A LAS REDES INFORMÁTICAS
- 2 ▲ PRINCIPIOS DE ELECTRÓNICA
- 3 ▲ EL LABORATORIO DE ELECTRÓNICA
- 4 ▲ CORRIENTE CONTINUA
- 5 ▲ CORRIENTE ALTERNA
- 6 ▲ DISPOSITIVOS ELECTRÓNICOS
- 7 ▲ CONSTRUCCIÓN DE CIRCUITOS
- 8 ▲ PROYECTOS: LUCES AUDIORÍTMICAS Y MICRÓFONO FM
- 9 ▲ DISEÑO DE CIRCUITOS IMPRESOS
- 10 ▲ SIMULACIÓN DE CIRCUITOS EN LA PC
- 11 ▲ ELECTRÓNICA DIGITAL Y COMPUERTAS LÓGICAS
- 12 ▲ TÉCNICAS DIGITALES APLICADAS
- 13 ▲ MICROPROCESADORES Y MICROCONTROLADORES
- 14 ▼ MICROCONTROLADORES PIC
- 15 ▼ PROYECTO: ANALIZADOR DE ESPECTRO CON PIC
- 16 ▼ CONECTIVIDAD POR CABLE
- 17 ▼ CONECTIVIDAD INALÁMBRICA
- 18 ▼ DISPLAYS
- 19 ▼ SENSORES Y TRANSDUCTORES
- 20 ▼ PROYECTO: MODIFICADOR DE VOZ
- 21 ▼ FUENTES DE ALIMENTACIÓN
- 22 ▼ PLATAFORMAS ABIERTAS
- 23 ▼ PLATAFORMA ARDUINO
- 24 ▼ PROYECTO: SISTEMA DE TELEMETRÍA CON ARDUINO



9 789871 949144

00013



www.redusers.com.blogspot.com.ar