



Simuladores interconectables basados en lógica digital



Serie: Recursos didácticos

Tapa:
Imagen combinada de la Supernova Remnant captada
por el telescopio Hubble - NASA.

a u t o r i d a d e s

PRESIDENTE DE LA NACIÓN

Dr. Néstor Kirchner

MINISTRO DE EDUCACIÓN, CIENCIA Y TECNOLOGÍA

Lic. Daniel Filmus

DIRECTORA EJECUTIVA DEL INSTITUTO NACIONAL DE
EDUCACIÓN TECNOLÓGICA

Lic. María Rosa Almandoz

DIRECTOR NACIONAL DEL CENTRO NACIONAL DE
EDUCACIÓN TECNOLÓGICA

Lic. Juan Manuel Kirschenbaum

Simuladores interconectables basados en lógica digital

María Claudia Cesetti

Adrián Frapiccini

Daniel Cesca

Daniel Pace

Colección Serie "Recursos didácticos".

Director del Programa: Juan Manuel Kirschenbaum.

Coordinadora general: Haydeé Noceti.

Distribución de carácter gratuito.

Queda hecho el depósito que previene la ley n° 11.723. © Todos los derechos reservados por el Ministerio de Educación, Ciencia y Tecnología - Instituto Nacional de Educación Tecnológica.

La reproducción total o parcial, en forma idéntica o modificada por cualquier medio mecánico o electrónico incluyendo fotocopia, grabación o cualquier sistema de almacenamiento y recuperación de información no autorizada en forma expresa por el editor, viola derechos reservados.

Industria Argentina.

ISBN 950-00-0532-8

Cesetti, María Claudia

Simuladores interconectables basados en lógica digital / María Claudia Cesetti; Adrián Frapiccini; Daniel Cesca; coordinado por Juan Manuel Kirschenbaum. - 1a ed. - Buenos Aires: Ministerio de Educación, Ciencia y Tecnología de la Nación. Instituto Nacional de Educación Tecnológica, 2006. 152 p. + 1 CD ROM ; 22x17 cm. (Recursos didácticos; 23)

ISBN 950-00-0532-8

I. Electrónica Digital. I. Frapiccini, Adrián II. Cesca, Daniel
III. Kirschenbaum, Juan Manuel, coord. IV. Título

CDD 621.381 3

Fecha de catalogación: 3/01/2006

Impreso en MDC MACHINE S. A., Marcelo T. de Alvear 4346 (B1702CFZ), Ciudadela, en marzo 2006

Tirada de esta edición: 2.000 ejemplares

Serie: “**Recursos didácticos**”

- 1 Invernadero automatizado
- 2 Probador de inyectores y de motores paso a paso
- 3 Quemador de biomasa
- 4 Intercomunicador por fibra óptica
- 5 Transmisor de datos bidireccional por fibra óptica, entre computadoras
- 6 Planta potabilizadora
- 7 Medidor de distancia y de velocidad por ultrasonido
- 8 Estufa de laboratorio
- 9 Equipamiento EMA –características físicas de los materiales de construcción–
- 10 Dispositivo para evaluar parámetros de líneas
- 11 Biodigestor
- 12 Entrenador en lógica programada
- 13 Entorno de desarrollo para programación de microcontroladores PIC
- 14 Relevador de las características de componentes semiconductores
- 15 Instalación sanitaria de una vivienda
- 16 Equipamiento para el análisis de estructuras de edificios
- 17 Cargador semiautomático para máquinas a CNC de accionamiento electroneumático
- 18 Biorreactor para la producción de alimentos
- 19 Ascensor
- 20 Pila de combustible
- 21 Generador eólico
- 22 Auto solar
- 23 Simuladores interconectables basados en lógica digital
- 24 Banco de trabajo
- 25 Matricería. Matrices y moldes
- 26 Máquina de vapor
- 27 Sismógrafo
- 28 Tren de aterrizaje
- 29 Manipulador neumático
- 30 Planta de tratamiento de aguas residuales

LAS METAS, LOS PROGRAMAS Y LAS LÍNEAS DE ACCIÓN DEL INSTITUTO NACIONAL DE EDUCACIÓN TECNOLÓGICA

El Instituto Nacional de Educación Tecnológica -INET- enmarca sus líneas de acción, programas y proyectos, en las metas de:

- Coordinar y promover programas nacionales y federales orientados a fortalecer la educación técnico-profesional, articulados con los distintos niveles y ciclos del sistema educativo nacional.
 - Implementar estrategias y acciones de cooperación entre distintas entidades, instituciones y organismos –gubernamentales y no gubernamentales-, que permitan el consenso en torno a las políticas, los lineamientos y el desarrollo de las ofertas educativas, cuyos resultados sean considerados en el Consejo Nacional de Educación-Trabajo –CoNE-T- y en el Consejo Federal de Cultura y Educación.
 - Desarrollar estrategias y acciones destinadas a vincular y a articular las áreas de educación técnico-profesional con los sectores del trabajo y la producción, a escala local, regional e interregional.
 - Diseñar y ejecutar un plan de asistencia técnica a las jurisdicciones en los aspectos institucionales, pedagógicos, organizativos y de gestión, relativos a la educación técnico-profesional, en el marco de los acuerdos y resoluciones establecidos por el Consejo Federal de Cultura y Educación.
 - Diseñar y desarrollar un plan anual de capacitación, con modalidades presenciales, semipresenciales y a distancia, con sede en el Centro Nacional de Educación Tecnológica, y con nodos en los Centros Regionales de Educación Tecnológica y las Unidades de Cultura Tecnológica.
 - Coordinar y promover programas de asistencia económica e incentivos fiscales destinados a la actualización y el desarrollo de la educación técnico-profesional; en particular, ejecutar las acciones relativas a la adjudicación y el control de la asignación del Crédito Fiscal –Ley N° 22.317–.
 - Desarrollar mecanismos de cooperación internacional y acciones relativas a diferentes procesos de integración educativa; en particular, los relacionados con los países del MERCOSUR, en lo referente a la educación técnico-profesional.
- Estas metas se despliegan en distintos programas y líneas de acción de responsabilidad de nuestra institución, para el periodo 2003-2007:

Programa 1. Formación técnica, media y superior no universitaria:

- 1.1. Homologación y validez nacional de títulos.
- 1.2. Registro nacional de instituciones de formación técnica.
- 1.3. Espacios de concertación.
- 1.4. Perfiles profesionales y ofertas formativas.
- 1.5. Fortalecimiento de la gestión institucional; equipamiento de talleres y laboratorios.
- 1.6. Prácticas productivas profesionalizantes: Aprender emprendiendo.

Programa 2. Crédito fiscal:

- 2.1. Difusión y asistencia técnica.
- 2.2. Aplicación del régimen.
- 2.3. Evaluación y auditoría.

Programa 3. Formación profesional para el desarrollo local:

- 3.1. Articulación con las provincias.
- 3.2. Diseño curricular e institucional.
- 3.3. Información, evaluación y certificación.

Programa 4. Educación para el trabajo y la integración social.

Programa 5. Mejoramiento de la enseñanza y del aprendizaje de la Tecnología y de la Ciencia:

- 5.1. Formación continua.
- 5.2. Desarrollo de recursos didácticos.

Programa 6. Desarrollo de sistemas de información y comunicaciones:

- 6.1. Desarrollo de sistemas y redes.
- 6.2. Interactividad de centros.

Programa 7. Secretaría ejecutiva del Consejo Nacional de Educación Trabajo –CoNE-T–.

Programa 8. Cooperación internacional.

Los materiales de capacitación que, en esta ocasión, estamos acercando a la comunidad educativa a través de la serie “Recursos didácticos”, se enmarcan en el Programa 5 del INET, focalizado en el mejoramiento de la enseñanza y del aprendizaje de la Tecnología y de la Ciencia, uno de cuyos propósitos es el de:

- Desarrollar materiales de capacitación destinados, por una parte, a la actualización de los docentes de la educación técnico-profesional, en lo que hace a conocimientos tecnológicos y científicos; y, por otra, a la integración de los recursos didácticos generados a través de ellos, en las aulas y talleres, como equipamiento de apoyo para los procesos de enseñanza y de aprendizaje en el área técnica.

Estos materiales didácticos han sido elaborados por especialistas del Centro Nacional de Educación Tecnológica del INET y por especialistas convocados a través del Programa de las Naciones Unidas para el Desarrollo –PNUD– desde su línea “Conocimientos científico-tecnológicos para el desarrollo de equipos e instrumentos”, a quienes esta Dirección expresa su profundo reconocimiento por la tarea encarada.

María Rosa Almandoz

Directora Ejecutiva del Instituto Nacional de Educación Tecnológica.
Ministerio de Educación, Ciencia y Tecnología

LAS ACCIONES DEL CENTRO NACIONAL DE EDUCACIÓN TECNOLÓGICA

Desde el Centro Nacional de Educación Tecnológica –CeNET– encaramos el diseño, el desarrollo y la implementación de proyectos innovadores para la enseñanza y el aprendizaje en educación técnico-profesional.

El CeNET, así:

- Es un ámbito de desarrollo y evaluación de metodología didáctica, y de actualización de contenidos de la tecnología y de sus sustentos científicos.
- Capacita en el uso de tecnología a docentes, profesionales, técnicos, estudiantes y otras personas de la comunidad.
- Brinda asistencia técnica a autoridades educativas jurisdiccionales y a educadores.
- Articula recursos asociativos, integrando a los actores sociales involucrados con la Educación Tecnológica.

Desde el CeNET venimos trabajando en distintas líneas de acción que convergen en el objetivo de reunir a profesores, a especialistas en Educación Tecnológica y a representantes de la industria y de la empresa, en acciones compartidas que permitan que la educación técnico-profesional se desarrolle en la escuela de un modo sistemático, enriquecedor, profundo... auténticamente formativo, tanto para los alumnos como para los docentes.

Una de nuestras líneas de acción es la de diseñar y llevar adelante un sistema de capaci-

tación continua para profesores de educación técnico-profesional, implementando trayectos de actualización. En el CeNET contamos con quince unidades de gestión de aprendizaje en las que se desarrollan cursos, talleres, pasantías, conferencias, encuentros, destinados a cada educador que desee integrarse en ellos presencialmente o a distancia.

Otra de nuestras líneas de trabajo asume la responsabilidad de generar y participar en redes que vinculan al Centro con organismos e instituciones educativas ocupados en la educación técnico-profesional, y con organismos, instituciones y empresas dedicados a la tecnología en general. Entre estas redes, se encuentra la Red Huitral, que conecta a CeNET con los Centros Regionales de Educación Tecnológica –CeRET– y con las Unidades de Cultura Tecnológica –UCT– instalados en todo el país.

También nos ocupa la tarea de producir materiales de capacitación docente. Desde CeNET hemos desarrollado distintas series de publicaciones –todas ellas disponibles en el espacio web www.inet.edu.ar–:

- *Educación Tecnológica*, que abarca materiales que posibilitan una definición curricular del área de la Tecnología en el ámbito escolar y que incluye marcos teóricos generales, de referencia, acerca del área en su conjunto y de sus contenidos, enfoques, procedimientos y estrategias didácticas más generales.

- *Desarrollo de contenidos*, nuestra segunda serie de publicaciones, que nuclea fascículos de capacitación en los que se profundiza en los campos de problemas y de contenidos de las distintas áreas del conocimiento tecnológico, y que recopila, también, experiencias de capacitación docente desarrolladas en cada una de estas áreas.
- *Educación con tecnologías*, que propicia el uso de tecnologías de la información y de la comunicación como recursos didácticos, en las clases de todas las áreas y espacios curriculares.
- *Educadores en Tecnología*, serie de publicaciones que focaliza el análisis y las propuestas en uno de los constituyentes del proceso didáctico: el profesional que enseña Tecnología, ahondando en los rasgos de su formación, de sus prácticas, de sus procesos de capacitación, de su vinculación con los lineamientos curriculares y con las políticas educativas, de interactividad con sus alumnos, y con sus propios saberes y modos de hacer.
- *Documentos de la escuela técnica*, que difunde los marcos normativos y curriculares que desde el CONET –Consejo Nacional de Educación Técnica– delinearon la educación técnica de nuestro país, entre 1959 y 1995.
- *Ciencias para la Educación Tecnológica*, que presenta contenidos científicos asociados con los distintos campos de la tecnología, los que aportan marcos conceptuales que permiten explicar y fundamentar los problemas de nuestra área.
- *Recursos didácticos*, que presenta contenidos tecnológicos y científicos,

estrategias –curriculares, didácticas y referidas a procedimientos de construcción– que permiten al profesor de la educación técnico-profesional desarrollar, con sus alumnos, un equipamiento específico para integrar en sus clases.

Desde esta última serie de materiales de capacitación, nos proponemos brindar herramientas que permitan a los docentes no sólo integrar y transferir sus saberes y capacidades, sino también, y fundamentalmente, acompañarlos en su búsqueda de soluciones creativas e innovadoras a las problemáticas con las que puedan enfrentarse en el proceso de enseñanza en el área técnica.

En todos los casos, se trata de propuestas de enseñanza basadas en la resolución de problemas, que integran ciencias básicas y tecnología, y que incluyen recursos didácticos apropiados para la educación técnico-profesional.

Los espacios de problemas tecnológicos, las consignas de trabajo, las estrategias de enseñanza, los contenidos involucrados y, finalmente, los recursos didácticos están planteados en la serie de publicaciones que aquí presentamos, como un testimonio de realidad que da cuenta de la potencialidad educativa del modelo de problematización en el campo de la enseñanza y del aprendizaje de la tecnología, que esperamos que resulte de utilidad para los profesores de la educación técnico-profesional de nuestro país.

Juan Manuel Kirschenbaum

Director Nacional del Centro Nacional de
Educación Tecnológica.
Instituto Nacional de Educación Tecnológica

LA SERIE “RECURSOS DIDÁCTICOS”

Desde esta serie de publicaciones del Centro Nacional de Educación Tecnológica, nos proponemos:

- Poner a consideración de los educadores un equipamiento didáctico a integrar en los procesos de enseñanza y de aprendizaje del área técnica que coordinan.
- Contribuir a la actualización de los docentes de la educación técnico-profesional, en lo que hace a conocimientos tecnológicos y científicos.

Inicialmente, hemos previsto el desarrollo de veinte publicaciones con las que intentamos abarcar diferentes contenidos de este campo curricular vastísimo que es el de la educación técnico-profesional.

En cada una de estas publicaciones es posible reconocer una estructura didáctica común:

1 Problemas tecnológicos en el aula. En esta primera parte del material se describen situaciones de enseñanza y de aprendizaje del campo de la educación técnico-profesional centradas en la resolución de problemas tecnológicos, y se presenta una propuesta de equipamiento didáctico, pertinente como recurso para resolver esas situaciones tecnológicas y didácticas planteadas.

2 Encuadre teórico para los problemas. En vinculación con los problemas didácticos y tecnológicos que constituyen el punto de partida, se presentan conceptos

tecnológicos y conceptos científicos asociados.

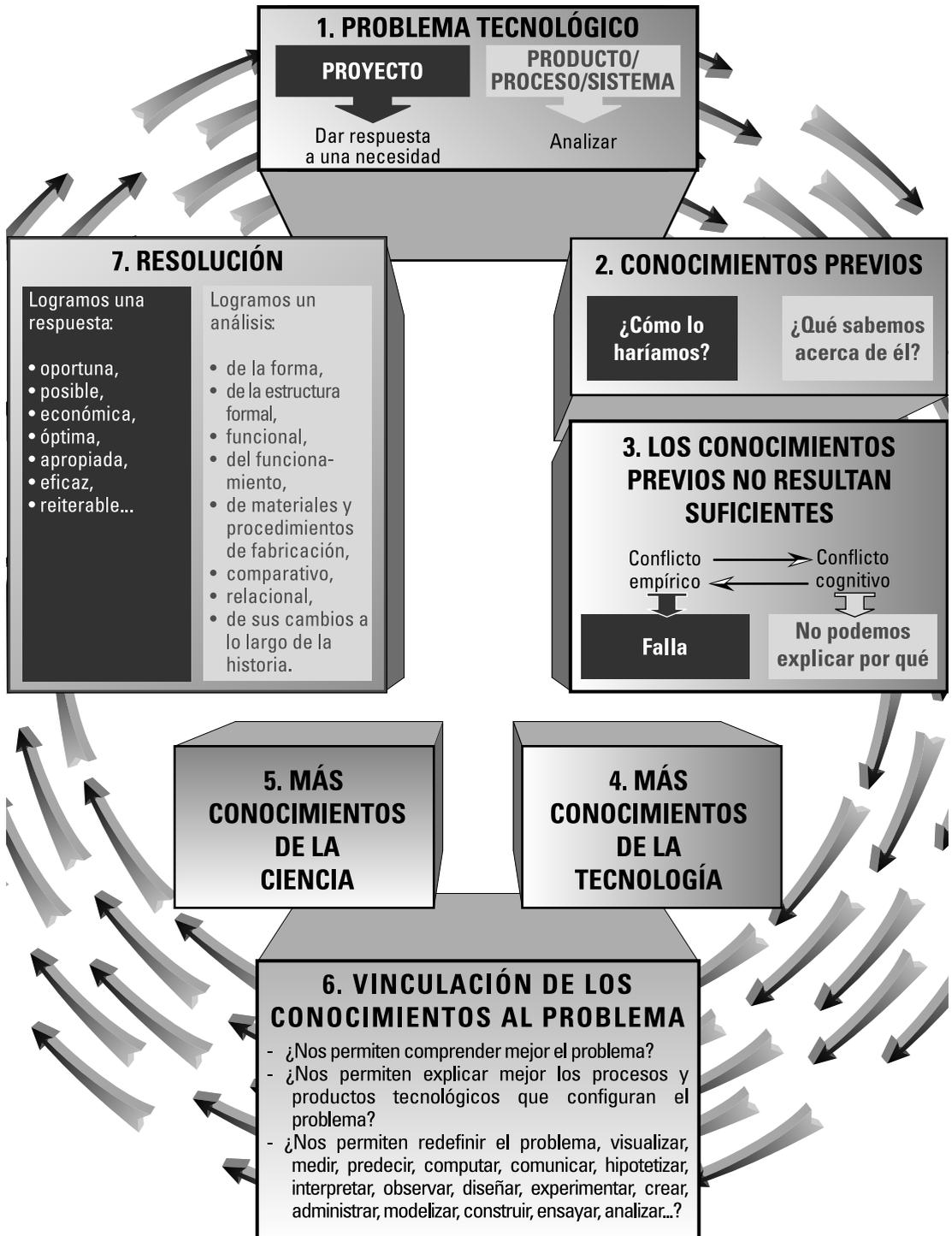
3 Hacia una resolución técnica. Manual de procedimientos para la construcción y el funcionamiento del equipo.

Aquí se describe el equipo terminado y se muestra su esquema de funcionamiento; se presentan todas sus partes, y los materiales, herramientas e instrumentos necesarios para su desarrollo; asimismo, se pauta el “paso a paso” de su construcción, armado, ensayo y control.

4 El equipo en el aula. En esta parte del material escrito, se retoman las situaciones problemáticas iniciales, aportando sugerencias para la inclusión del recurso didáctico construido en las tareas que docente y alumnos concretan en el aula.

5 La puesta en práctica. Este tramo de la publicación plantea la evaluación del material didáctico y de la experiencia de puesta en práctica de las estrategias didácticas sugeridas. Implica una retroalimentación –de resolución voluntaria– de los profesores destinatarios hacia el Centro Nacional de Educación Tecnológica, así como el punto de partida para el diseño de nuevos equipos.

Esta secuencia de cuestiones y de momentos didácticos no es azarosa. Intenta replicar –en una producción escrita– las mismas instancias de trabajo que los profesores de Tecnología ponemos en práctica en nuestras clases:



Es a través de este circuito de trabajo (problema-respuestas iniciales-inclusión teórica-respuestas más eficaces) como enseñamos y como aprenden nuestros alumnos en el área:

- La tarea comienza cuando el profesor presenta a sus alumnos una **situación codificada en la que es posible reconocer un problema tecnológico**; para configurar y resolver este problema, es necesario que el grupo ponga en marcha un proyecto tecnológico, y que encare análisis de productos o de procesos desarrollados por distintos grupos sociales para resolver algún problema análogo. Indudablemente, no se trata de cualquier problema sino de uno que ocasiona obstáculos cognitivos a los alumnos respecto de un aspecto del mundo artificial que el profesor –en su marco curricular de decisiones– ha definido como relevante.
- El proceso de enseñanza y de aprendizaje comienza con el planteamiento de esa situación tecnológica seleccionada por el profesor y con la construcción del espacio-problema por parte de los alumnos, y continúa con la búsqueda de **respuestas**.
- Esta detección y construcción de respuestas no se sustenta sólo en los conocimientos que el grupo dispone sino en la **integración de nuevos contenidos**.
- El enriquecimiento de los modos de “ver” y de encarar la resolución de un problema tecnológico –por la adquisición de nuevos conceptos y de nuevas formas técnicas de intervención en la situación

desencadenante– suele estar **distribuida materialmente** –en equipamiento, en materiales, en herramientas–.

No es lo mismo contar con este equipamiento que prescindir de él.

Por esto, lo que intentamos desde nuestra serie de publicaciones es acercar al profesor distintos recursos didácticos que ayuden a sus alumnos en esta tarea de problematización y de intervención –sustentada teórica y técnicamente– en el mundo tecnológico.

Caracterizamos como **recurso didáctico** a todo material o componente informático seleccionado por un educador, quien ha evaluado en aquél posibilidades ciertas para actuar como mediador entre un problema de la realidad, un contenido a enseñar y un grupo de alumnos, facilitando procesos de comprensión, análisis, profundización, integración, síntesis, transferencia, producción o evaluación.

Al seleccionar los recursos didácticos que forman parte de nuestra serie de publicaciones, hemos considerado, en primer término, su potencialidad para posibilitar, a los alumnos de la educación técnico-profesional, configurar y resolver distintos problemas tecnológicos.

Y, en segundo término, nos preocupó que cumplieran con determinados rasgos que les permitieran constituirse en medios eficaces del conocimiento y en buenos estructurantes cognitivos, al ser incluidos en un aula por un profesor que los ha evaluado como perti-

nentes. Las cualidades que consideramos fundamentales en cada equipo que promovemos desde nuestra serie de publicaciones "Recursos didácticos", son:

- Modularidad (puede adaptarse a diversos usos).
- Resistencia (puede ser utilizado por los alumnos, sin peligro de romperse con facilidad).
- Seguridad y durabilidad (integrado por materiales no tóxicos ni peligrosos, y durables).
- Adaptabilidad (puede ser utilizado en el taller, aula o laboratorio).
- Acoplabilidad (puede ser unido o combinado con otros recursos didácticos).
- Compatibilidad (todos los componentes, bloques y sistemas permiten ser integrados entre sí).
- Facilidad de armado y desarmado (posibilita pruebas, correcciones e incorporación de nuevas funciones).
- Pertinencia (los componentes, bloques funcionales y sistemas son adecuados para el trabajo con los contenidos curriculares de la educación técnico-profesional).
- Fiabilidad (se pueden realizar las tareas preestablecidas, de la manera esperada).
- Coherencia (en todos los componentes, bloques funcionales o sistemas se siguen las mismas normas y criterios para el armado y utilización).
- Escalabilidad (es posible utilizarlo en proyectos de diferente nivel de com-

plejidad).

- Reutilización (los diversos componentes, bloques o sistemas pueden ser desmontados para volver al estado original).
- Incrementabilidad (posibilidad de ir agregando piezas o completando el equipo en forma progresiva).

Haydeé Noceti

Coordinadora de la acción "Conocimientos científico-tecnológicos para el desarrollo de equipos e instrumentos".
Centro Nacional de Educación Tecnológica



23. Simuladores interconectables basados en lógica digital

Este material de capacitación fue desarrollado por:

María Claudia Cesetti

Es ingeniera electrónica (Universidad Nacional del Sur) e ingeniera laboral (Universidad Tecnológica Nacional. Facultad Regional Bahía Blanca), con capacitación docente para profesionales. Es docente en la Escuela de Educación Técnica N° 2 de Bahía Blanca, en las áreas de Electrónica e Informática. Se desempeña como secretaria del Departamento de Electrónica en la EET N°2.

Adrián Frapiccini

Es técnico electrónico y técnico superior en informática, con capacitación docente para profesionales. Es docente en la Escuela de Educación Técnica N° 2 y en el Instituto Técnico "La Piedad" de Bahía Blanca, en las áreas de Electrónica e Informática.

Daniel Cesca

Es ingeniero electricista con orientación electrónica, con capacitación docente para profesionales. Es docente en la Escuela de Educación Técnica N° 2 de Bahía Blanca, en las áreas de Electrónica, Electricidad e Informática. Se desempeña como Jefe del Departamento de Electrónica de la EET N°2. Fue Miembro de Comisión Mixta en el convenio de articulación y acreditación académica, entre la Universidad Tecnológica Nacional (Facultad Regional Bahía Blanca) y la EET N° 2. Participó en jornadas de definición de diseño curricular de los trayectos técnico-profesionales de la orientación Electrónica, en la provincia de Buenos Aires.

Daniel Pace

Es ingeniero electrónico, con capacitación docente para profesionales. Es docente en las Escuelas de Educación Técnica N° 2 y N° 3 de Bahía Blanca, en las áreas de Electrónica e Informática. Se desempeña como profesional personal civil en la Armada Argentina.

Dirección del Programa:
Juan Manuel Kirschenbaum

Coordinación general:
Haydeé Noceti

Diseño didáctico:
Ana Rúa

Administración:
Adriana Perrone

Monitoreo y evaluación:
Laura Irurzun

Diseño gráfico:
Tomás Ahumada
Karina Lacava
Alejandro Carlos Mertel

Diseño de tapa:
Laura Lopresti
Juan Manuel Kirschenbaum

Diseño de CD:
Sergio Iglesias

Retoques fotográficos:
Roberto Sobrado

Con la colaboración
del equipo de profesionales
del Centro Nacional
de Educación Tecnológica



Índice

Las metas, los programas y las líneas de acción del Instituto Nacional de Educación Tecnológica.....	VIII
Las acciones del Centro Nacional de Educación Tecnológica.....	X
La serie “Recursos didácticos”.....	XII

1 El recurso didáctico que proponemos.....	4
---	----------

2 El encuadre teórico.....	9
-----------------------------------	----------

- Sistemas y códigos de numeración
- Álgebra de Boole
- Circuitos integrados digitales
- Lógica combinatorial por bloques
- Lógica secuencial
- Microcontroladores
- Dispositivos relacionados con el equipo

3 Hacia una resolución técnica. Manual de procedimientos para la construcción y el funcionamiento del equipo.....	77
--	-----------

- La programación de los microcontroladores PIC. El entorno MPLAB®
- El equipo
- Los componentes
- El desarrollo del circuito impreso
- Los canales por los que llega el programa al PIC
- Otras consideraciones útiles para el manejo del equipo

4 El equipo en el aula.....	103
------------------------------------	------------

- Los pasos en la generación de un proyecto
- Las utilidades del sistema de simuladores
- Otras posibilidades

5 La puesta en práctica.....	112
-------------------------------------	------------

Anexo

CD con circuitos eléctricos, ruteos de pistas y vistas de cada una de las placas de los simuladores.



1. EL RECURSO DIDÁCTICO QUE PROPONEMOS

En nuestra escuela, la Escuela de Educación Técnica N° 2 de Bahía Blanca -provincia de Buenos Aires-, se desarrolla el espacio curricular "Instrumentos y herramientas de desarrollo aplicado".

En esta asignatura se abordan competencias de un tercer nivel de complejidad de las técnicas digitales electrónicas, las que permiten profundizar el desarrollo práctico de los contenidos y la integración de éstos a sistemas electrónicos reales, afianzando la capacidad

crítica y de diagnóstico de los alumnos, su trabajo en equipo, y la actitud positiva ante la innovación y el adelanto tecnológico.

Desde "Instrumentos y herramientas de desarrollo aplicado" organizamos los aprendizajes de nuestros alumnos en torno a un problema central que da unidad a los contenidos y actividades, y que permite un enfoque pluridisciplinario en el desarrollo de las subáreas de competencia.

SUBÁREAS DE COMPETENCIA

Diseñar y desarrollar productos de electrónica digital.

Programar microcontroladores.

Determinar las pruebas, ajustes y ensayos de calidad de los productos diseñados.

Montar y poner en marcha los sistemas electrónicos diseñados.

Seleccionar los componentes a utilizar en el montaje.

Manejar los instrumentos de medición, relacionándolos adecuadamente.

Identificar un emprendimiento y, luego, programarlo.

En nuestra tarea de diseño de las actividades de enseñanza y de aprendizaje, incorporamos criterios cognitivos orientados a garantizar que, a partir de ellas, los estudiantes pongan en juego capacidades complejas, transferibles a diferentes contextos.

"Instrumentos y herramientas de desarrollo aplicado" se organiza curricularmente a partir de las pautas de:

- Proyectar sistemas microcontrolados para las distintas aplicaciones electrónicas.
- Controlar, a través de equipos electrónicos inteligentes, el funcionamiento de sistemas.
- Diseñar programas informáticos, en lenguaje de bajo nivel, para la operación de equipos.
- Operar equipos programables.
- Mantener el funcionamiento de sistemas microcontrolados.

Las actividades formativas priorizan el trabajo concreto de los estudiantes con los dispositivos, componentes y equipos electrónicos, la resolución de problemas en los que ejerciten sus capacidades, el análisis de contenidos específicos y el uso de textos especializados: manuales, folletería industrial, procedimientos y normas.

Para cumplir con esta organización de la tarea educativa, nuestros alumnos -que ya cuentan con nociones acerca de microcontroladores y de sus aplicaciones- necesitan inte-

grar herramientas de programación y de simulación de los problemas reales cuya resolución encaran:

- programas de desarrollo para la simulación, edición y búsqueda de errores en el programa realizado como solución planteada,
- programas que se ocupen de grabar el microcontrolador a usar,
- sistema electrónico que permita cargar el programa realizado en el microcontrolador,
- sistema electrónico que permita probar el sistema, sin necesidad de reconstruir todo el problema.

Para contar con las mejores herramientas de trabajo en cada uno de estos rubros, nos ha resultado útil integrar el programa MPLAB®¹ que realiza las tareas de edición, simulación y búsqueda de errores de los programas que en él se desarrollan.

Es nuestra experiencia con este programa y con el equipo asociado que desarrollamos desde la EET N° 2 de Bahía Blanca, la que queremos compartir con usted.

Respecto de los programas que se ocupan de grabar el microcontrolador, en nuestras clases integramos distintos software que

¹ MPLAB® es accesible y se puede obtener gratuitamente en www.microchip.com.

también vamos a presentar en este material de capacitación, a los que se puede acceder fácilmente por Internet; Por ejemplo, el *EPICWin*[®] para el paralelo, *Noop*[®] o *ICprog*[®] para el serie o el *ProPI*[®] para ambos puertos.

Además de integrar estos programas, nuestros alumnos diseñan un programador de microcontrolador y lo construyen con componentes que se consiguen en el mercado a un costo razonable. También vamos a relatarle pormenorizadamente esta parte de nuestra experiencia.

Finalmente, compartiremos con usted cómo los estudiantes arman un sistema universal compuesto por tarjetas de interconexión que engloba la posibilidad de probar muchos proyectos sin realizar demasiadas modificaciones y que sólo requiere de la interconexión de componentes básicos de un sistema de prueba. Con ella, nuestros alumnos optimizan espacio, seguridad, dinero y riesgo de conexiones equivocadas, y cuentan con la posibilidad de probar cualquier circuito de distinto grado de complejidad, realizando un conexionado mínimo.

El programador y el simulador resultan, entonces, las partes constitutivas de la **estación de trabajo** que nuestros alumnos diseñan, desarrollan, prueban y ponen en marcha. Esta estación de trabajo -que se completa con los programas que mencionamos, con una PC y con documentación técnica- se convierte, así, en el *practicum* en el que desarrollamos nuestra asignatura.

En esta estación de trabajo, los alumnos pueden simular con el software de aplicación (*MPLAB*[®]), grabar en el microcontrolador elegido y, finalmente, probar el programa planteado como solución de cualquier situación real, sin necesidad de armar un circuito individual para cada aplicación.

Esta experiencia de enseñanza nos ha permitido integrar a las clases un modelo eficaz para que los alumnos experimenten y observen los cambios en los distintos programas y, con ello, las distintas alternativas en los sistemas que han de ser controladas con estos dispositivos, simulando y ensayando variadas situaciones, en entornos que se aproximan cada vez más a situaciones de trabajo del campo real.

En nuestra tarea en la EET N° 2 de Bahía Blanca, el equipo **simuladores interconectables basados en lógica digital** ha constituido un recurso didáctico versátil de entrenamiento para microcontroladores PIC *Microchip*[®] en toda su gama flash.

Su versatilidad radica en que permite programar, probar y simular los distintos proyectos realizados *in circuit*, es decir sin necesidad de quitar el microcontrolador de la placa y con el uso de dispositivos externos interconectados a ésta.

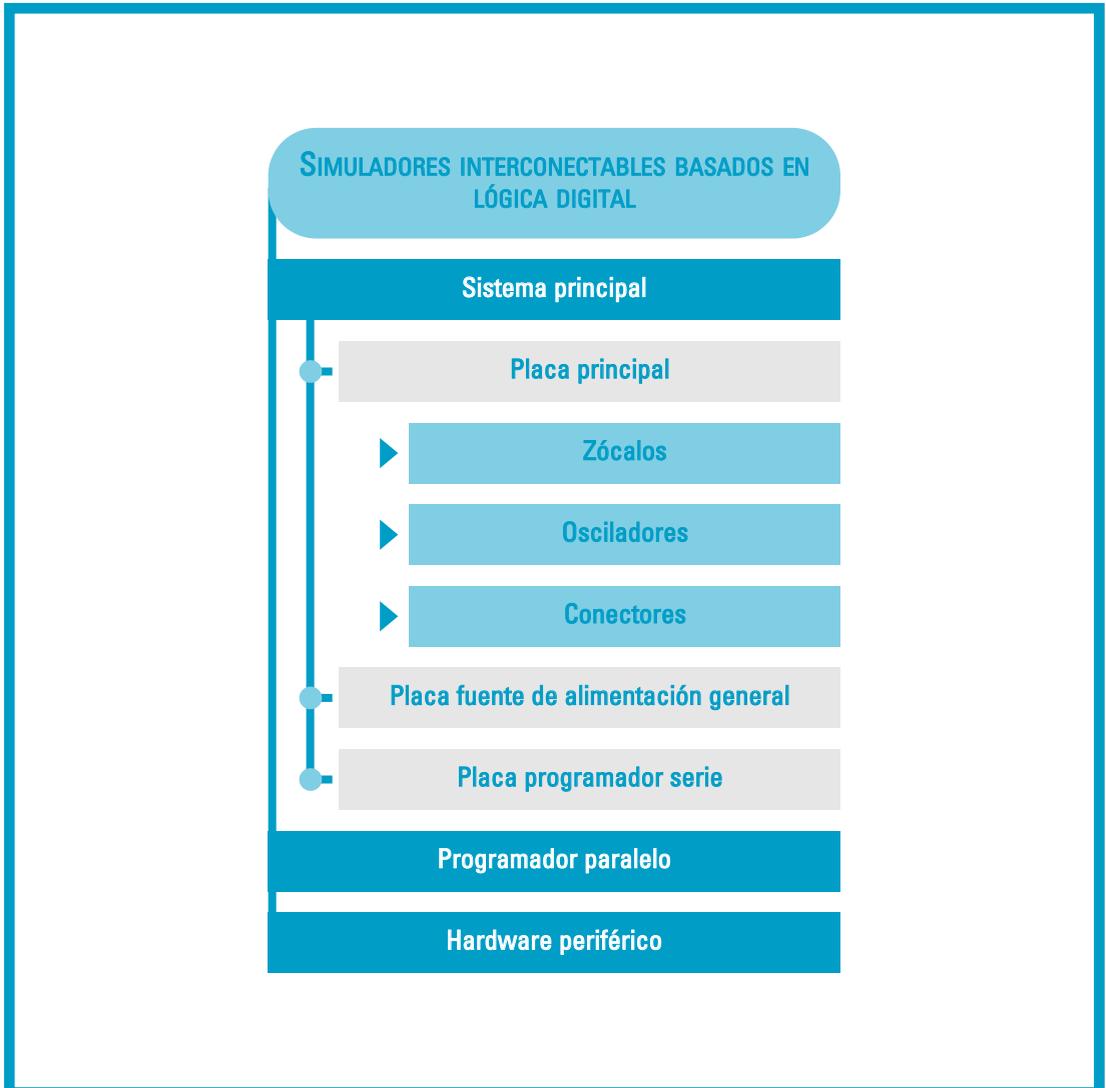
El equipo permite conectar distintas placas, de acuerdo con la necesidad que cada proyecto tecnológico plantea: desde el encendido de un simple led o una barra de

led, hasta el manejo de un display de 7 segmentos, por ejemplo.

Por su interconectabilidad -rasgo que deseamos enfatizar en el título de esta obra-, el equipo no queda limitado al grado de posibilidades para las que se lo ha diseñado sino que puede ser ampliado a otras nuevas, definidas por el usuario, y permite expe-

rimentar con la totalidad de las instrucciones del microcontrolador, sin necesidad de componentes adicionales.

El hecho de contar en la placa principal con tres zócalos permite aceptar todas las variedades de microcontroladores PIC, con la posibilidad de trabajar con osciladores RC o cristal de cuarzo.



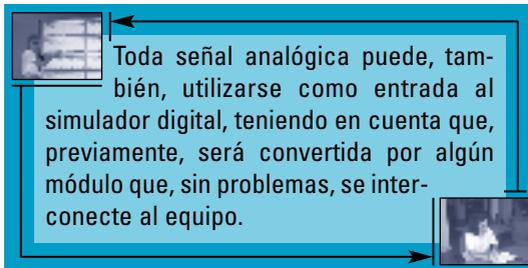
Modelizar este tipo de placas nos permite -y desde el control de las variables más sencillas esperamos que también a usted y a sus -por ejemplo, de apertura y de cierre de alumnos- contar con estaciones de trabajo en válvulas- hasta las más complejas -por ejemplo, en sistemas de control realimentado- las que los estudiantes puedan desarrollar

2. EL ENCUADRE TEÓRICO

En esta parte de nuestro material vamos a presentarle, a manera de referencia, la información teórica² necesaria para desarrollar algunos circuitos, empezando por los más básicos y sencillos, los que permiten utilizar los **simuladores interconectables basados en lógica digital** que proponemos.

La ventaja de trabajar con sistemas digitales consiste en que éstos son mucho más eficaces, rápidos, sistemáticos, seguros y confiables que los sistemas analógicos. Por otra parte, los sistemas digitales pueden trabajar con más información -esto es, con una mayor rapidez de su flujo- que los sistemas analógicos.

Esto no quiere decir que los sistemas digitales hayan desplazado a los analógicos; cada uno tiene sus aplicaciones bien definidas; pero, en este material de capacitación vamos a tomar en cuenta algunas características de los sistemas digitales y subsistemas asociados.



Toda señal analógica puede, también, utilizarse como entrada al simulador digital, teniendo en cuenta que, previamente, será convertida por algún módulo que, sin problemas, se interconecte al equipo.

² Le proponemos ampliarla con:

- Ginzburg, Mario Carlos. *Introducción a las técnicas digitales con circuitos integrados*. Edición del autor.
- Mandado, Enrique (1980; 3° ed.) *Sistemas electrónicos digitales*. Marcombo.
- Pérez, Julio M. (1980) *Técnicas digitales I y II*. Arsenal Naval Puerto Belgrano.

Sistemas y códigos de numeración

La noción de cantidad está asociada con la necesidad de representar un conjunto de objetos. Esta noción se manifiesta por medio de números y éstos, por razones de practicidad, a través de símbolos. Para operarlos en forma eficaz, también se debe definir un conjunto de reglas de operación.

Las distintas culturas han adoptado diversas formas de simbolizar los números. Por ejemplo, los romanos utilizaban signos de valor creciente que se agrupaban de derecha a izquierda, sumándose o restandose, según requiriera el orden decreciente o no: I, V, X, L, C, D, M.

Ejemplos:

$$\begin{aligned} \text{CXVII} &= 100 + 10 + 5 + 1 + 1 \\ \text{MCMV} &= 1.000 + (1.000 - 100) + 5 \end{aligned}$$

Los pueblos orientales y americanos desarrollaron sistemas posicionales, basados en un conjunto limitado de símbolos, entre los cuales se incluye el cero para indicar la ausencia de elementos. En estos sistemas cada símbolo, además del valor que posee considerado aisladamente, tiene un significado o peso distinto, según la ubicación que ocupa en el grupo de caracteres.

En un sistema de base b (cantidad de carac-

Otro ejemplo de A - B usando bit de signo:

A	0 25	0 25	A	0 11001	0 11001
-B	- 0 18	- 1 81	-B	- 0 10010	- 1 01101
	?	1 006			1 0 00110
		↘ +1			↘ +1
		0 07			0 00111

Consideremos el caso de $A < B$. El hecho de dar un número negativo en la resta binaria se detecta pues no aparece un 1 de transporte que excede la cantidad máxima de dígitos del problema. Eso indica que debemos complementar el resultado.

Por ejemplo: $18 - 25$

A	0 10010	0 10010
-B	- 0 11001	- 1 00110
	?	1 11000
		1 00111 ↓

Se entiende por **código** a una representación unívoca de las cantidades, de tal forma que a cada una de éstas se asigna una combinación de símbolos determinada.

Estos conocimientos sobre cómo se realizan las operaciones matemáticas en forma binaria son de utilidad, en el momento de comprender el funcionamiento interno del microcontrolador puesto que, de esta manera, es como se realizan las distintas operaciones matemáticas dentro de cualquier procesador digital como es el caso del utilizado en este módulo digital.

El código establece una correspondencia entre un conjunto de informaciones y otro de símbolos.

El sistema binario recibe el nombre de **código binario natural**: Con N cifras binarias o bit se pueden obtener 2^N combinaciones diferentes.

Si del código 8421, que permite 16 combinaciones, sólo usamos las 10 primeras correspondientes a los números del 0 al 9 del sistema decimal, se obtiene el código BCD natural -Decimal codificado en binario-.

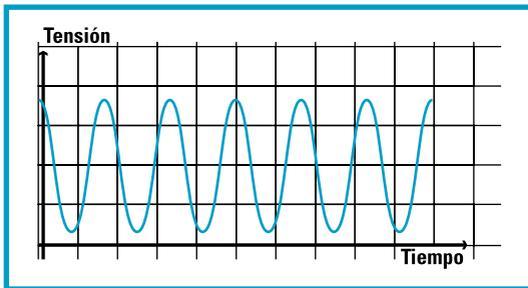
Tabla de códigos							
Decimal	BINNAT 8421	GRAY	BCD natural 8421	AIKEN (BCD) 2421	5421 (BCD)	EXC. TRES (BCD)	Johnson (BCD)
0	0000	0000	0000	0000	0000	0011	00000
1	0001	0001	0001	0001	0001	0100	00001
2	0010	0011	0010	0010	0010	0101	00011
3	0011	0010	0011	0011	0011	0110	00111
4	0100	0110	0100	0100	0100	0111	01111
5	0101	0111	0101	1011	1000	1000	11111
6	0110	0101	0110	1100	1001	1001	11110
7	0111	0100	0111	1101	1010	1010	11100
8	1000	1100	1000	1110	1011	1011	11000
9	1001	1101	1001	1111	1100	1100	10000
10	1010	1111					
11	1011	1110					
12	1100	1010					
13	1101	1011					
14	1110	1001					
15	1111	1000					

Álgebra de Boole

Un **álgebra de Boole** es toda clase o conjunto de elementos que pueden tomar dos valores perfectamente diferenciados que se designan como 0 y 1, y están relacionados por operaciones básicas.

Se define una señal analógica como una función que es continua en todo el dominio.

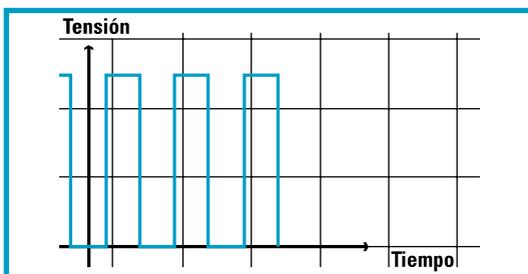
Si consideramos, por ejemplo, una tensión, ésta tendría una forma de onda:



La función, en este caso, toma todos los valores que ocurren en los instantes $t = t_1, t_2, \dots$

En los sistemas digitales, encontramos magnitudes, elementos, etc., que son especiales, en el sentido de que sólo pueden adoptar dos valores distintos. Es decir, son discretos.

En ellos, la tensión realiza transiciones abruptas entre dos niveles V_1 y V_2 sin asumir ningún otro valor.



Si decidiéramos, arbitrariamente, que el nivel más positivo sea el verdadero (uno, alto, existe, etc.) y el más negativo falso (cero, bajo, no existe, etc.), habremos adoptado la convención de lógica positiva; de lo contrario, será lógica negativa.

Considere usted esta particularidad:

- En los sistemas digitales se habla de **lógica combinacional** cuando cada combinación de las señales binarias existente en la entrada de un dispositivo da por resultado, siempre, la misma señal binaria de salida, sin importarle lo pasado anteriormente.
- En tanto, los circuitos son de **lógica secuencial** cuando las señales en las salidas dependen de las presentes en sus entradas, como también de las salidas pasadas (último registro de memoria).

Le presentamos, ahora, algunas definiciones clave:

- **Variable lógica.** Es cualquier símbolo literal A, B, \dots, Z empleado para representar dispositivos o magnitudes físicas que exhiben dos estados posibles, claramente definidos. En nuestro caso, 1 y 0. Por ejemplo: $Z = 1$, lámpara encendida. $Z = 0$, lámpara apagada.
- **Función lógica.** Z es una variable dependiente (función) de otras variables; por ejemplo, A y $B \Rightarrow Z = f(A, B)$. Una función se define dando una regla u otra información, por lo que se puede determinar la variable dependiente (Z), cuando se especifican las variables independientes (A y B). Una función del álgebra

de Boole es una variable binaria cuyo valor depende de una expresión algebraica en la que se relacionan entre sí las variables binarias por medio de operaciones básicas.

- **Compuerta lógica.** Es un circuito lógico cuya operación puede ser definida por una función del álgebra de Boole, que seguidamente desarrollaremos.
- **Tabla de verdad.** Es la tabulación ordenada de todas las combinaciones posibles de las variables asociadas a la función. Con N variables podemos realizar 2^N combinaciones. La tabla de verdad de una función lógica es, entonces, la representación donde se indica el estado lógico "1" o "0" que toma la función lógica para cada una de las combinaciones de las variables de las cuales depende.

Funciones básicas



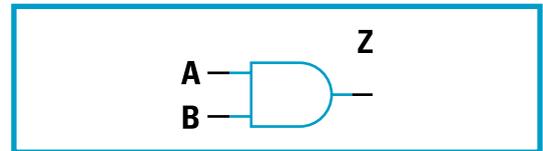
La **función and** se puede caracterizar diciendo que Z es: uno, verdadero, existe, etc. sólo si A y B también son: uno, verdadero, existen, etc.

$$Z = A \cdot B$$

La tabla de verdad es:

A	B	Z
0	0	0
0	1	0
1	0	0
1	1	1

El dispositivo que realiza esta operación se denomina compuerta y se simboliza:



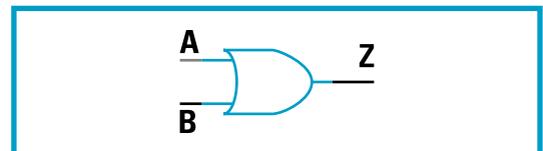
La **función or** es uno si A o B lo son:

$$Z = A + B$$

Su tabla de verdad es:

A	B	Z
0	0	0
0	1	1
1	0	1
1	1	1

El símbolo de la compuerta es:



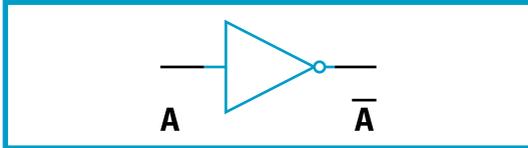
La **función complementación o not** es una negación o inversión.

- Si decimos que Z existe cuando A existe $\Rightarrow Z = A$
- Si asumimos la otra alternativa, que Z existe cuando A no existe $\Rightarrow Z = \bar{A}$

Su tabla de verdad:

A	Z
0	1
1	0

Su símbolo es:



Propiedades		
Commutativa	Asociativa	Distributiva
$A + B = B + A$	$A + (B + C) = (A + B) + C$	$A + B \cdot C = (A + B) \cdot (A + C)$
$A \cdot B = B \cdot A$	$A \cdot (B \cdot C) = (A \cdot B) \cdot C$	$A \cdot (B + C) = A \cdot B + A \cdot C$

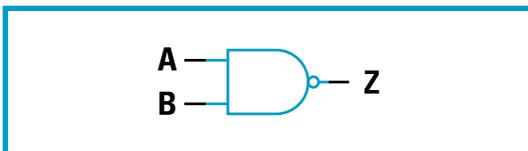
Teoremas del álgebra de Boole			
$A + 0 = A$	$A + 1 = 1$	$A + A = A$	$A + \bar{A} = 1$
$A \cdot 0 = 0$	$A \cdot 1 = A$	$A \cdot A = A$	$A \cdot \bar{A} = 0$
$\bar{\bar{A}} = A$			

Las **funciones nand** y **nor** surgen por la combinación de dos funciones básicas

nand = and - not

$$Z = \overline{A \cdot B}$$

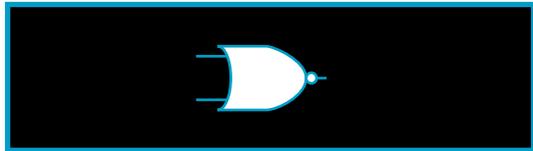
A	B	Z
0	0	1
0	1	1
1	0	1
1	1	0



nor = or - not

$$Z = \overline{A + B}$$

A	B	Z
0	0	1
0	1	0
1	0	0
1	1	0



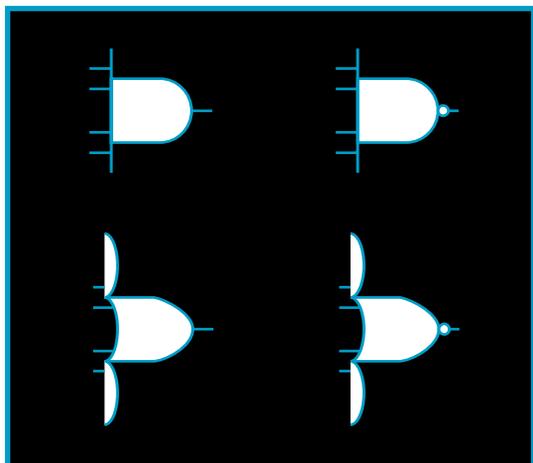
Las funciones básicas *and*, *or* y las combinadas *nand*, *nor* no necesariamente deben ser de dos variables como fueron vistas, sino que pueden estar conformadas por N de ellas:

$$Z = A \cdot B \cdot C \cdot D$$

$$Z = \overline{A \cdot B \cdot C \cdot D}$$

$$Z = A + B + C + D$$

$$Z = \overline{A + B + C + D}$$



El **teorema de Morgan** se cumple para N variables.

$$\begin{aligned} 1 \quad & \overline{A + B} = \overline{A} \cdot \overline{B} \\ 2 \quad & \overline{A \cdot B} = \overline{A} + \overline{B} \end{aligned}$$

Todas las funciones que le hemos presentado, así como los teoremas y propiedades, pueden ser perfectamente simulados como práctica en el equipo que le proponemos desarrollar.

Mediante la utilización de las propiedades, leyes y teoremas, siempre es posible llevar una función lógica a una forma típica preestablecida.

Existen dos formas canónicas:

- la **forma minterma** (suma lógica de productos lógicos) y
- la **forma maxterma** (producto lógico de sumas lógicas).

Cada una de ellas formadas por mintermos y maxtermos, respectivamente.

- **mintermo** es un producto lógico en el cual aparecen todas las variables del problema, negadas o no.
- **maxtermo**, lo mismo pero en suma lógica.

Consideremos un ejemplo.

Para cuatro variables

Mintermo: $A \cdot B \cdot C \cdot D$
 Maxtermo: $A + B + C + D$

La **función suma de productos** es una forma canónica de una función particular asociada por la suma lógica de todos los minitérminos que satisfacen dicha función.

Para tres variables:

Mintermo	ABC
m0	000
m1	001
m2	010
m3	011
m4	100
m5	101
m6	110
m7	111

$$F = \alpha_0 m_0 + \alpha_1 m_1 + \alpha_2 m_2 + \alpha_3 m_3 + \alpha_4 m_4 + \alpha_5 m_5 + \alpha_6 m_6 + \alpha_7 m_7$$

$$F = \sum_{i=0}^{2^n-1} \alpha_i m_i$$

Donde:

- α_i adopta el valor cero o uno, según la función analizada.
- $\alpha = 0$ cuando el término no aparece; $\alpha = 1$ cuando el término aparece.

La **función producto de sumas** es una forma canónica de una función particular formada por el producto lógico de todos los maxitérminos que satisfacen dicha función.

$$F_M = (\alpha_0 + M_0) (\alpha_1 + M_1) (\alpha_2 + M_2) (\alpha_3 + M_3) \dots (\alpha_Z + M_Z)$$

$$F_M = \prod_{i=0}^{2^n-1} (\alpha_i M_i)$$

Donde:

- $\alpha = 0$ si el término aparece y $\alpha = 1$ cuando no interviene la combinación.

La **función or exclusiva** expresa: Z es uno si exclusivamente A es uno o si exclusivamente B es uno, es decir que A y B no sean uno simultáneamente.

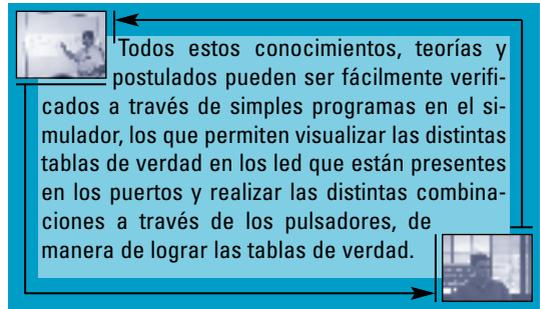
En esta reducción por mapas:

- **Celda:** Es cada una de las pequeñas divisiones de un mapa y representa una combinación particular. En el caso que esa combinación haga valer uno a la función, dicho uno se trasladará al mapa en la celda correspondiente.
- **Celdas compatibles:** Celdas que poseen un lado en común y se combinan en una reducción inicial.
- **Bloque:** Conjunto compatible de celdas que potencian una reducción (contiene 2^N celdas).
- **Bloque principal:** Es el bloque más grande que incluye una celda dada.
- **Bloque esencial:** Bloque principal que posee por lo menos una celda no contenida en ningún otro bloque.
- **Orden de bloque:** Lo da N y nos dice cuántas variables elimina.

Procedimiento:

- Se toman todos los "unos" que no se pueden combinar con ningún otro.
- Se forman los grupos de "2" unos que no pueden formar un grupo de "4".
- Se forman los grupos de "4" unos que no pueden formar uno de "8".
- Así, hasta cubrir todos los unos, siempre respetando 2^N .

En ocasiones, existen combinaciones de las variables que no se presentan nunca. Este tipo de combinaciones se denomina **redundancia** y, puesto que no aparecen en la función, pueden tomarse como uno o cero, según convenga a la reducción. Se las nota con X.



← Todos estos conocimientos, teorías y postulados pueden ser fácilmente verificados a través de simples programas en el simulador, los que permiten visualizar las distintas tablas de verdad en los led que están presentes en los puertos y realizar las distintas combinaciones a través de los pulsadores, de manera de lograr las tablas de verdad. →

Circuitos integrados digitales

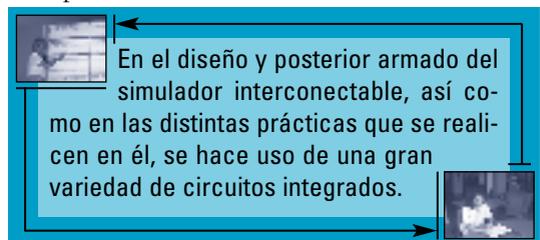
En la actualidad, los circuitos integrados (CI) son la base fundamental del desarrollo de la electrónica, debido a que facilitan y economizan tareas.

Un **circuito integrado** es una pieza o encapsulado -generalmente, de silicio o de algún otro material semiconductor- que, utilizando las propiedades de los semiconductores, es capaz de realizar las funciones definidas por la unión, en un circuito, de varios elementos electrónicos (resistencias, condensadores, transistores, etc.).

Dentro de las distintas clasificaciones de los circuitos integrados, podemos mencionar a la que los divide entre:

- los de operación fija y
- los programables.

Los circuitos integrados de operación fija funcionan bajo el álgebra de Boole, utilizando una compuerta digital para cada una de las operaciones.



← En el diseño y posterior armado del simulador interconectable, así como en las distintas prácticas que se realizan en él, se hace uso de una gran variedad de circuitos integrados. →

La complejidad de un CI puede medirse por el número de compuertas lógicas que contiene. Los métodos de fabricación actuales permiten construir circuitos integrados cuya complejidad está en el rango de una a 10^5 o más compuertas por circuito integrado.

Según esto, los CI se clasifican en los siguientes niveles o escalas de integración:

- **SSI** (pequeña escala), menor de 10 compuertas.
- **MSI** (media escala), entre 10 y 100 compuertas.
- **LSI** (alta escala), entre 100 y 10.000 compuertas.
- **VLSI** (muy alta escala), a partir de 10.000 compuertas.

La capacidad de integración depende, a su vez, de dos factores:

- El **área** ocupada por cada compuerta; depende, a su vez, del tipo y del número de transistores utilizados para realizarla. Cuanto menor sea esta área, mayor será la capacidad de integración a gran escala.

- El **consumo** de potencia. En un circuito integrado, el consumo total del chip depende del número de compuertas utilizadas. Si éstas son demasiadas, el calor generado por efecto Joule puede provocar un aumento de temperatura que haga que el circuito se deteriore.

Por esto, existen distintos tipos de encapsulados. Estos componentes están estandarizados, lo que permite compatibilidad entre fabricantes, de forma que las características más importantes sean comunes.

Los componentes lógicos se engloban -en términos generales- dentro de una de dos **familias lógicas**:

- **TTL**, diseñada para una alta velocidad.
- **CMOS**, diseñada para un bajo consumo.

Actualmente, dentro de estas dos familias se han creado otras, que intentan conseguir lo mejor de ambas: un bajo consumo y una alta velocidad. La familia lógica ECL se encuentra entre la TTL y la CMOS, y nació como un intento de conseguir la rapidez de TTL y el bajo consumo de CMOS; pero, en raras ocasiones se emplea.

Cuadro comparativo de las familias

Parámetro	TTL estándar	TTL 74L	TTL Schottky de baja potencia (LS)	Fairchild 4000B CMOS (con $V_{cc}=5V$)	Fairchild 4000B CMOS (con $cc=10V$)
Tiempo de propagación de compuerta	10 ns	33 ns	5 ns	40 ns	20 ns
Frecuencia máxima de funcionamiento	35 MHz	3 MHz	45 MHz	8 MHz	16 MHz
Potencia disipada por compuerta	10 mW	1 mW	2 mW	10 nW	10 nW
Margen de ruido admisible	1 V	1 V	0.8 V	2 V	4 V
Fan out	10	10	20	50 (*)	50 (*)

(*) O lo que permita el tiempo de propagación admisible

Lógica combinacional por bloques

Se denomina **circuito combinacional** a un conjunto de circuitos en los que se cumple la condición de que sus salidas son exclusivamente función de sus entradas, sin que intervenga el último valor en que se encontraban dichas salidas.

Un circuito combinacional se analiza determinando la salida de los elementos lógicos que lo constituyen (normalmente, compuertas lógicas), partiendo de las variables de entrada y avanzando en el sentido de la señal hacia la salida.

CIRCUITOS COMBINACIONALES

Multiplexores

Demultiplexores

Decodificadores

Codificadores

Comparadores

Aritméticos

Generadores, verificadores de paridad

De opción

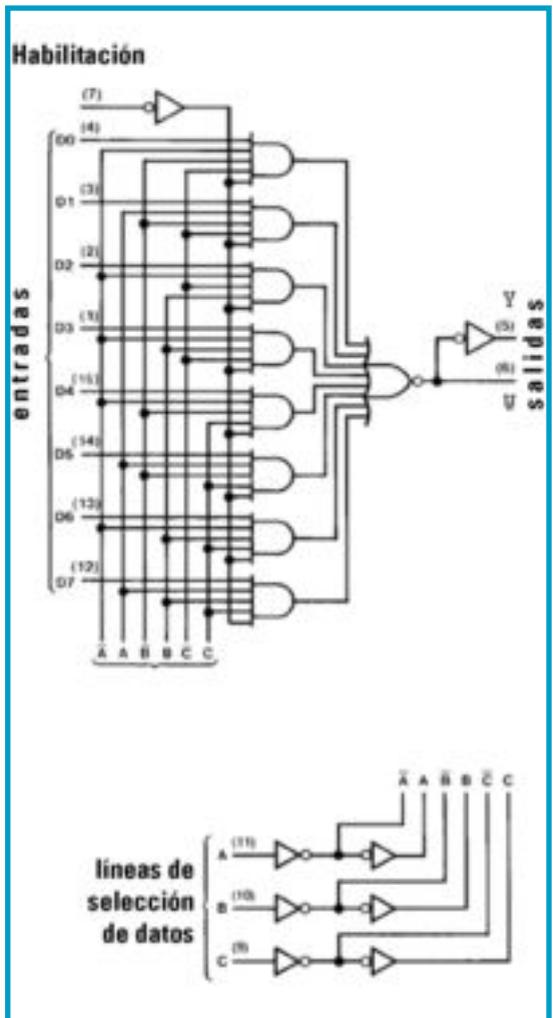
Habilitadores

Buffer

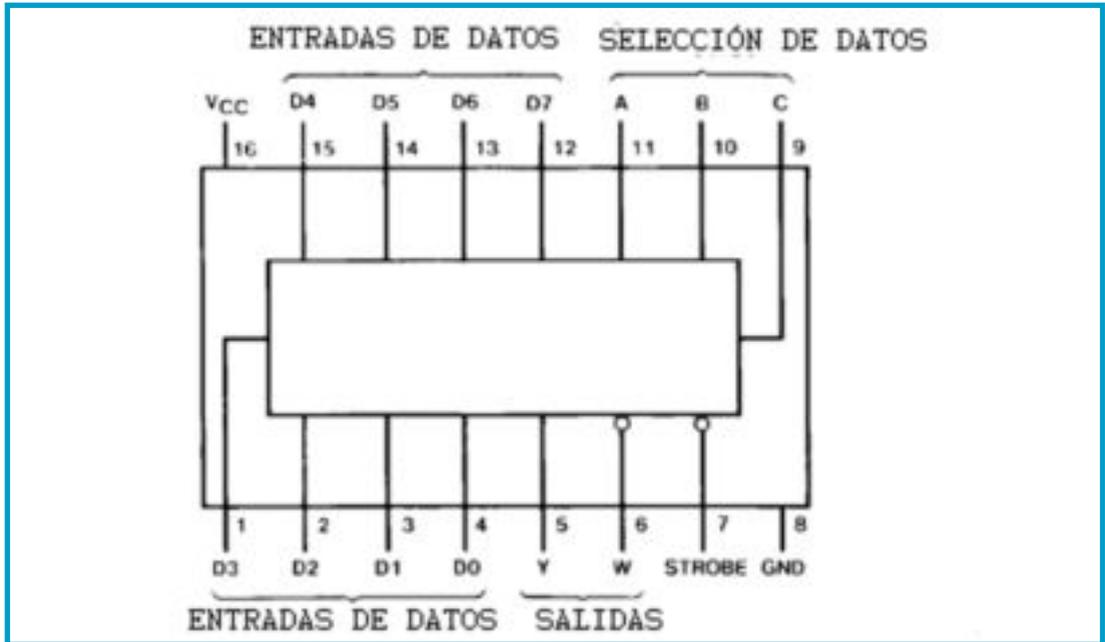
Circuitos multiplexores

Son circuitos que envían, por un solo canal de salida, alguna de las informaciones presentes en varias líneas de entrada, previo direccionamiento binario.

Tomaremos como ejemplo el *multiplexor 74151* con 8 entradas, cuyo esquema lógico es:



El diagrama de conexiones y la tabla de verdad correspondientes son:



Entradas				Salidas	
Selección			Strobe S	Y	W
C	B	A			
X	X	X	H	L	H
L	L	L	L	D0	$\overline{D0}$
L	L	H	L	D1	$\overline{D1}$
L	H	L	L	D2	$\overline{D2}$
L	H	H	L	D3	$\overline{D3}$
H	L	L	L	D4	$\overline{D4}$
H	L	H	L	D5	$\overline{D5}$
H	H	L	L	D6	$\overline{D6}$
H	H	H	L	D7	$\overline{D7}$

H = 1
L = 0
W = Salida negada

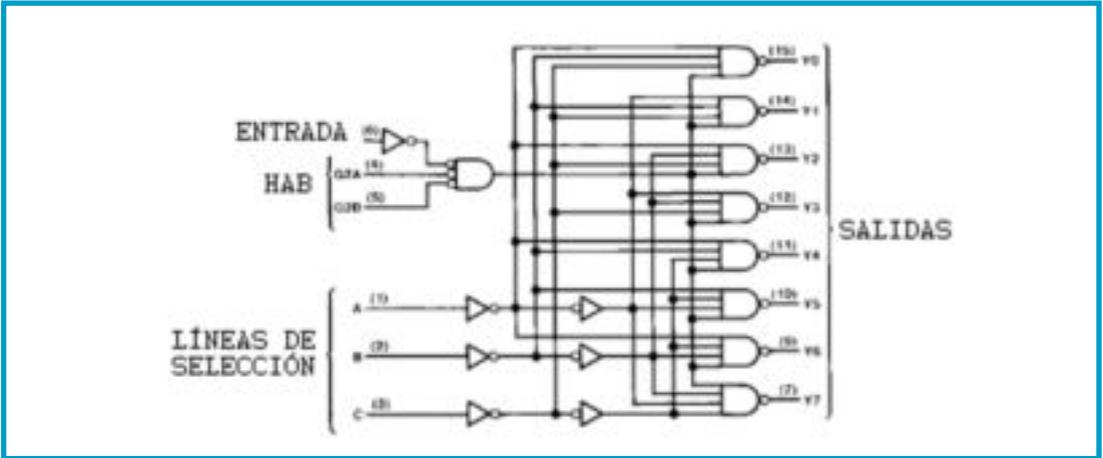
El *strobe* es una habilitación (hab) para permitir el funcionamiento o no del multiplexor.

Circuitos demultiplexores

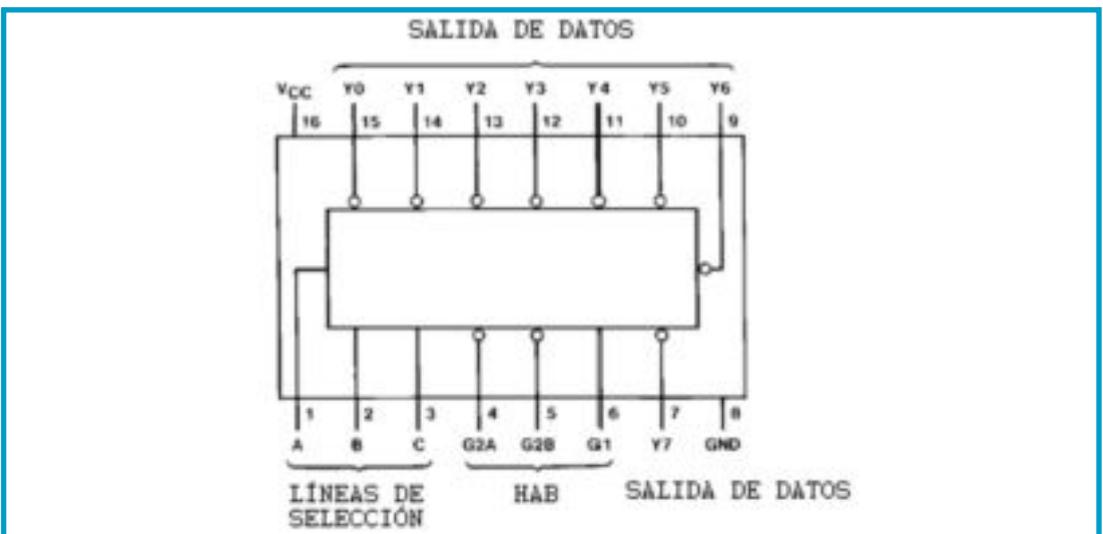
Utilizan la función inversa de los multiplexores. La información de la entrada se transmite a la línea de salida seleccionada, mediante las entradas de control o de direccionamiento; es decir, se encauzan los datos desde una fuente común de entrada hacia uno de 2^n destinos de salida.

Es importante comentar que los demultiplexores pueden trabajar también como decodificadores.

Tomemos como ejemplo el *demultiplexor 74138* con 8 salidas, cuyo esquema lógico vemos a continuación:



El diagrama de conexiones y la tabla de verdad correspondientes son:



Entradas					Salidas							
Habilitadas		Seleccionadas										
G1	G2 nota	C	B	A	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
X	H	X	X	X	H	H	H	H	H	H	H	H
L	X	X	X	X	H	H	H	H	H	H	H	H
H	L	L	L	L	L	H	H	H	H	H	H	H
H	L	L	L	H	H	L	H	H	H	H	H	H
H	L	L	H	L	H	H	L	H	H	H	H	H
H	L	L	H	H	H	H	H	L	H	H	H	H
H	L	H	L	L	H	H	H	H	L	H	H	H
H	L	H	L	H	H	H	H	H	H	L	H	H
H	L	H	H	L	H	H	H	H	H	H	L	H
H	L	H	H	H	H	H	H	H	H	H	H	L

H = 1.
L = 0, siendo G1 la entrada de dato del demultiplexor.

En la tabla de verdad, la habilitación (hab) $G2 = G2A + G2B$.

Circuitos decodificadores

Un decodificador es un circuito lógico combinacional, que convierte un código de entrada binario de n bits en m líneas de salida (n puede ser cualquier entero y m es un entero menor o igual a 2^n), tales que cada línea de salida será activada para una sola de las combinaciones posibles de entrada.

Puesto que cada una de las entradas puede ser 1 o 0, hay 2^n combinaciones o códigos de entrada. Para cada una de estas combinaciones de entrada, sólo una de las m salidas estará activada 1, para lógica positiva; todas las otras salidas estarán en 0.

Muchos decodificadores se diseñan para pro-

ducir salidas 0 activas, lógica negativa, donde la salida seleccionada es 0, mientras que las otras son 1. Esto último se indica siempre por la presencia de pequeños círculos en las líneas de salida del diagrama del decodificador.

Algunos decodificadores no usan todos los 2^n códigos posibles de entrada, sino sólo algunos de ellos. Por ejemplo, un decodificador BCD a decimal, tiene un código de entrada de 4 bits que sólo usa diez grupos codificados BCD, 0000 hasta 1001. Algunos de estos decodificadores se diseñan de tal manera que, si cualquiera de los códigos no usados se aplica a la entrada, ninguna de las salidas se activará.

Consideremos el ejemplo del decodificador BCD a 7 segmentos 7447.

Muchas veces, es necesario leer mediante barras luminosas los dígitos decimales

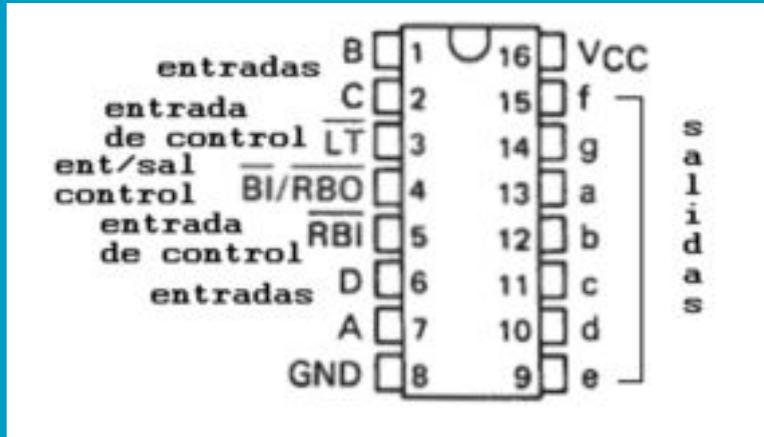
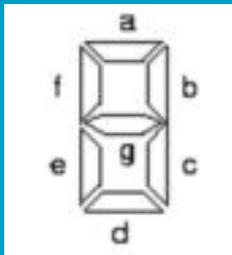
correspondientes a los resultados parciales o totales de algún sistema que opera -por ejemplo, en código BCD (decimal codificado en binario; o sea, números binarios codificados del 0 al 9)-. Esto requiere pasar por un código de 7 barras, por ejemplo, cuyas combinaciones harán encender 7 segmentos luminosos (led), tal que forman los números decimales del 0 al 9.

Como cada led está asociado a una letra, se puede construir una tabla con las combinaciones binarias como entrada y las letras a, b, c, d, e, f, g como salida. De esta manera, cada función tomará el valor 1, de acuerdo con los segmentos que deban iluminarse, según el número decimal que deba aparecer en coincidencia con cada combinación binaria de las entradas. Por ejemplo, para el 0: a = b = c = d = e = f = 1 y g = 0.

Tanto los displays como los respectivos conversores son utilizados en nuestro equipo simulador para visualizar conteos, por ejemplo, o datos de alguna variable tomada en las entradas del microcontrolador.

En este caso, este circuito pasaría a ser un conversor del código BCD a los 7 segmentos.

		Entradas						Salidas								
N°	LT	RBI	D	C	B	A	BI/RBO	a	b	c	d	e	f	g		
0	1	1	0	0	0	0	1	1	1	1	1	1	1	0		
1	1	x	0	0	0	1	1	0	1	1	0	0	0	0		
2	1	x	0	0	1	0	1	1	1	0	1	1	0	1		
3	1	x	0	0	1	1	1	1	1	1	1	0	0	1		
4	1	x	0	1	0	0	1	0	1	1	0	0	1	1		
5	1	x	0	1	0	1	1	1	0	1	1	0	1	1		
6	1	x	0	1	1	0	1	0	0	1	1	1	1	0		
7	1	x	0	1	1	1	1	1	1	1	0	0	0	0		
8	1	x	1	0	0	0	1	1	1	1	1	1	1	1		
9	1	x	1	0	0	1	1	1	1	1	0	0	1	1		I
10	1	x	1	0	1	0	1	0	0	0	1	1	0	1	<	N
11	1	x	1	0	1	1	1	0	0	1	1	0	0	1	<	V
12	1	x	1	1	0	0	1	0	1	0	0	0	1	1	<	Á
13	1	x	1	1	0	1	1	1	0	0	1	0	1	1	<	L
14	1	x	1	1	1	0	1	0	0	0	1	1	1	1	<	I
15	1	x	1	1	1	1	1	0	0	0	0	0	0	0	<	D
BI	x	x	x	x	x	x	0	0	0	0	0	0	0	0		O
RBI	1	0	0	0	0	0	0	0	0	0	0	0	0	0		
LT	0	x	x	x	x	x	1	1	1	1	1	1	1	1		

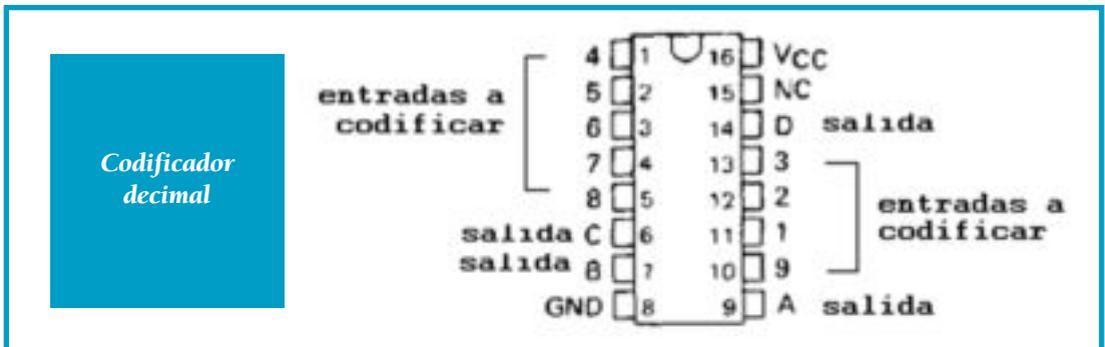


Circuitos codificadores

Se trata de circuitos combinatoriales que poseen 2^n entradas y n salidas, y cuya estructura es tal que, al activarse una de las entradas adoptando un estado lógico determinado 0 o 1, en la salida aparece la combi-

nación binaria correspondiente al número decimal asignado a dicha entrada.

Tomaremos como ejemplo el *codificador decimal 74147* de 10 entradas a cuatro salidas binarias, cuyo esquema lógico vemos a continuación:

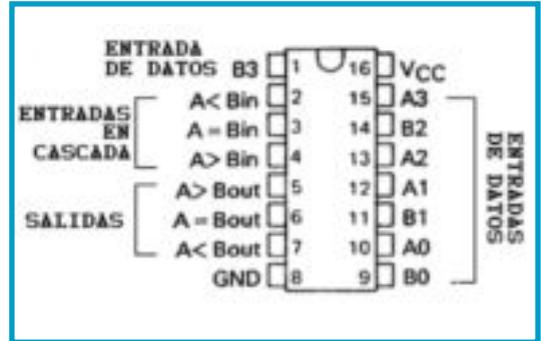


Circuitos comparadores

Comparan dos combinaciones binarias y nos dicen si son iguales o no, o si una es mayor o menor que la otra mediante un 1 colocado en la salida correspondiente.

a	b	A = B	A > B	A < B
0	0	1	0	0
0	1	0	0	1
1	0	0	1	0
1	1	1	0	0

Tomemos como ejemplo el circuito comparador 7485 con magnitud de 4 bits:



Entradas de datos a comparar				Entradas en cascada			Salidas		
A3 B3	A2 B2	A1 B1	A0 B0	A > B	A < B	A = B	A > B	A < B	A = B
A3 > B3	X	X	X	X	X	X	H	L	L
A3 < B3	X	X	X	X	X	X	L	H	L
A3 = B3	A2 > B2	X	X	X	X	X	H	L	L
A3 = B3	A2 < B2	X	X	X	X	X	L	H	L
A3 = B3	A2 = B2	A1 > B1	X	X	X	X	H	L	L
A3 = B3	A2 = B2	A1 < B1	X	X	X	X	L	H	L
A3 = B3	A2 = B2	A1 = B1	A0 > B0	X	X	X	H	L	L
A3 = B3	A2 = B2	A1 = B1	A0 < B0	X	X	X	L	H	L
A3 = B3	A2 = B2	A1 = B1	A0 = B0	H	L	L	H	L	L
A3 = B3	A2 = B2	A1 = B1	A0 = B0	L	H	L	L	H	L
A3 = B3	A2 = B2	A1 = B1	A0 = B0	X	X	H	L	L	H
A3 = B3	A2 = B2	A1 = B1	A0 = B0	H	H	L	L	L	L
A3 = B3	A2 = B2	A1 = B1	A0 = B0	L	L	L	H	H	L

Circuitos aritméticos

Los sistemas digitales realizan una variedad de tareas de procesamiento de información. Entre las funciones básicas encontradas están las diversas funciones aritméticas.

Sin duda, la operación aritmética básica es la

suma de dos dígitos binarios. Esta suma simple consta de cuatro operaciones elementales posibles; a saber:

$$\begin{aligned}
 0 + 0 &= 0 \\
 0 + 1 &= 1 \\
 1 + 0 &= 1 \\
 1 + 1 &= 10
 \end{aligned}$$

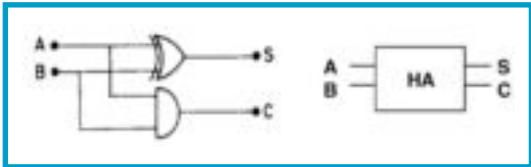
Las primeras tres operaciones producen una suma de un dígito o bit de longitud; pero, cuando los bits sumandos son iguales a 1, la suma binaria consta de dos dígitos, correspondiente al número decimal 2.

El bit más significativo (el de mayor peso) de este resultado se denomina **acarreo**.

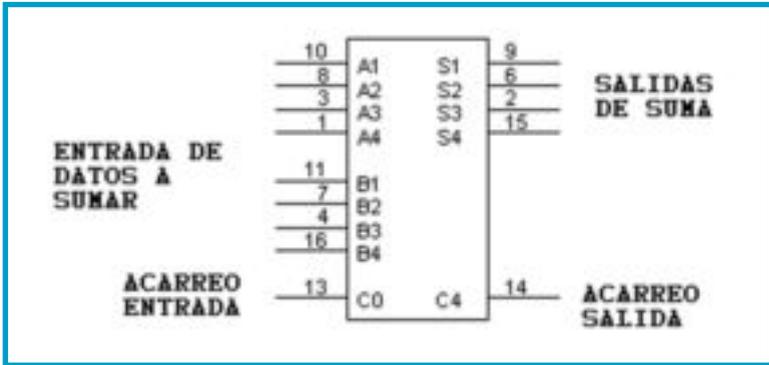
Un circuito combinacional que realiza la suma de dos bits se denomina **semisumador** o *half adder*. De allí que son circuitos combinatoriales que realizan operaciones matemáticas.

A	B	Suma	Acarreo
0	0	0	0
0	1	1	0
1	0	0	0
1	1	0	1

Realizando su implementación mediante compuertas, queda:



Tomemos como ejemplo el circuito 7483, sumador completo binario de 4 bits con acarreo de salida:



Este sumador es un circuito digital capaz de realizar la suma aritmética de dos o más dígitos binarios, además de un posible acarreo de entrada. Este último es particularmente útil cuando se diseña en forma modular un sumador de dos números de n bits.

Además, este integrado se puede utilizar como parte de un circuito sumador-restador binario, junto a circuitos complementadores y a una lógica combinacional adecuada.



Es de interés recordar que resulta más sencillo realizar estas operaciones con un programa y desarrollarlo mediante un microcontrolador como los que se utilizan en nuestro recurso didáctico.

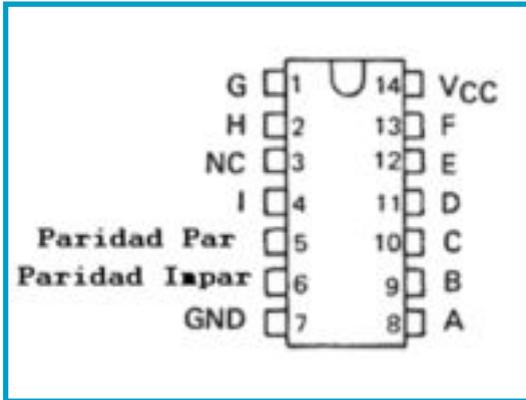


Circuitos generadores, verificadores de paridad

Al hablar de códigos, hemos visto que muchas veces se utiliza un bit llamado de paridad P, tal que para cada número codificado existe siempre un número par o impar de unos.

El generador de paridad es un sistema combinatorial que realiza, en esencia, la suma de la cantidad de unos que posee el dato de entrada, y en consecuencia, genera su salida. Con un 1 lógico, indica si dicha cantidad es par o impar.

Tomemos como ejemplo el circuito 74280, generador de paridad de 9 entradas y 2 salidas:



Nº de entradas desde A hasta I que están en "1"	Salidas	
	Paridad par	Paridad impar
0, 2, 4, 6, 8	1	0
1, 3, 5, 7, 9	0	1

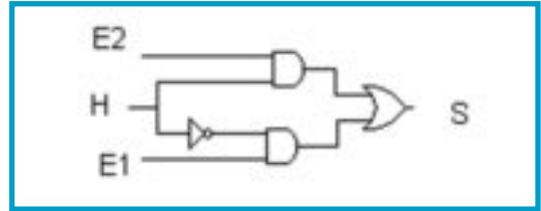
Luego de recibido el dato en el receptor, se debe verificar la paridad enviada con la generada en el receptor, para detectar errores en la transmisión.

Circuitos de opción

Son circuitos combinacionales que permiten obtener, a la salida, una de dos posibles variables de entrada según la opción adoptada.

La variable H, en este caso, permite optar entre E1 o E2, de la siguiente manera:

H	S
0	E1
1	E2

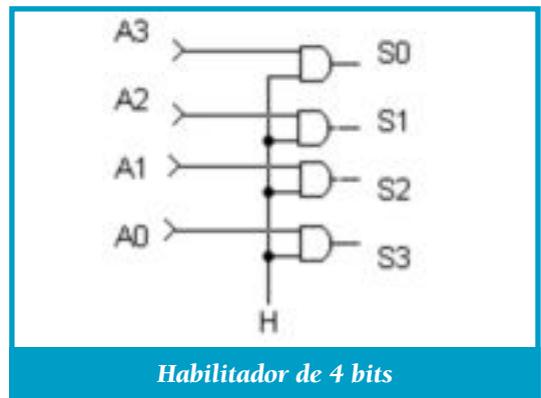


Circuitos habilitadores

Son circuitos combinacionales que permiten o prohíben el paso de determinada cantidad de bits tomados como entrada hacia una salida de la misma cantidad de bits.

En este caso, la línea h es la que habilita el paso o lo deshabilita, de la siguiente manera:

H	S0	S1	S2	S3
0	0	0	0	0
1	A0	A1	A2	A3



Circuitos buffer

Estos circuitos, además de actuar como separadores en circuitos más complejos y como niveladores de corriente de trabajo, cumplen

una función similar a la del habilitador cuando son buffers con activación.

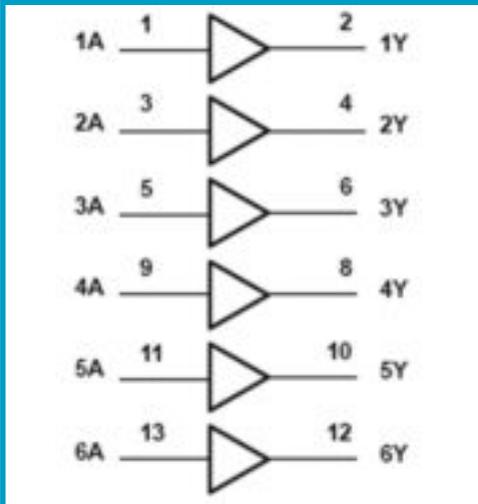
Esta función -la llamamos, por ejemplo, H- tiene la característica que:

- si se activa con un 1 lógico al colocarlo allí, la salida reproduce inmediatamente la entrada presente sea ésta 0 o 1;
- si se coloca un 0 lógico en H, la salida permanece en estado de alta impedancia,

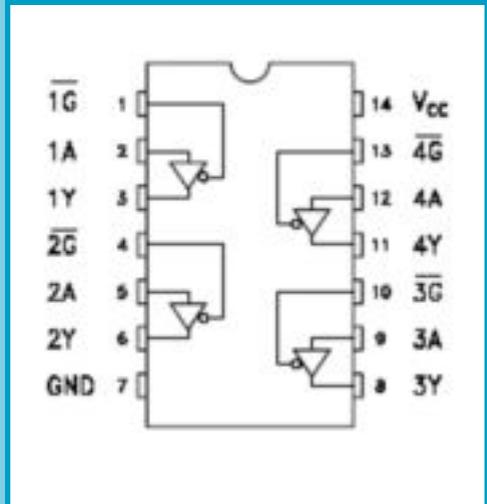
de *stand by* o de desconexión momentánea del dispositivo, hasta que éste vuelva a activarse colocando un 1 lógico en la línea H.

Este tipo de compuerta se denomina *Tri-State* -tres estados; 0, 1 o alta impedancia-.

Tomemos como ejemplos los circuitos 7407 *buffer drivers* y 74HC125, *buffer drivers* con habilitador, utilizados en el equipo simuladores interconectables.



74LS07



74HC125; pines 1,4, 10 y 13 habilitaciones por "0"



Todos los circuitos de lógica combinal por bloques que hemos detallado pueden ser usados en las distintas prácticas que se planteen, para ser programadas y simuladas con nuestro equipo.

Muchas funciones pueden ser programadas

en el microcontrolador y varios de estos circuitos integrados se asocian a la resolución de las prácticas, formando parte del hardware que soporta el equipo, conectándolo en los pines adicionales de que dispone, para distintos módulos armados.



Lógica secuencial

A partir de aquí, iremos analizando:

Un **sistema secuencial** es aquel cuya salida no sólo es función de las señales de entrada, sino de la salida en un instante anterior.

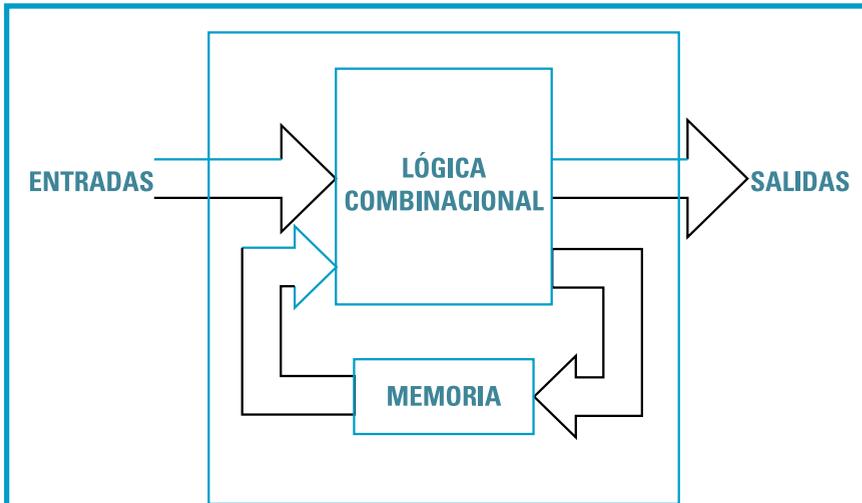
- Flip flop o biestables
- Contadores
- Registros de desplazamiento
- Contadores en anillo

Flip flop

Podemos notar, entonces, la presencia de la variable tiempo (t) y, así, hablar de un sistema lógico con memoria.

El *flip flop* o biestable se caracteriza por poseer dos estados estables. Es una memoria de un bit, ya que la salida permanece en un estado hasta que una combinación de señales de entrada provoca un cambio en la salida.

Es memoria volátil debido a que, si el circuito se queda sin electricidad, al reponerse ésta, la información almacenada anteriormente es borrada.



Esquema de un sistema secuencial

Si las entradas en el esquema del sistema secuencial son A, B y C, y la salida se denomina Q, la función secuencial es:

$$Q^+ = A, B, C, Q$$

De acuerdo con las premisas o lógica combinacional, los biestables se clasifican en:

- FF-RS,
- FF-JK,
- FF-D
- FF-T.

Bi estable o FF	Premisas	Ecuación
RS R = Reset S = Set	R = 0 S = 1 Q ⁺ = 1 R = 1 S = 0 Q ⁺ = 0 R = S = 0 Q ⁺ = Q R = S = 1 restricción	$Q^+ = Q\bar{R} + \bar{Q}S$
JK J = S, K = R. cuando J = K = 1 la salida cambia de estado.	J = 1 K = 0 Q ⁺ = 1 J = 0 K = 1 Q ⁺ = 0 J = 0 K = 0 Q ⁺ = Q J = 1 K = 1 Q ⁺ = \bar{Q}	$Q^+ = Q\bar{K} + \bar{Q}J$
D Data o delay	Q ⁺ = D	Q ⁺ = D
T Toggle o trigger	T = 0 Q ⁺ = Q T = 1 Q ⁺ = \bar{Q}	$Q^+ = Q\bar{T} + \bar{Q}T$

Otra forma de clasificar a los biestables es de acuerdo con un funcionamiento sincrónico o asincrónico.

El **sincronismo** establece que el funcionamiento del FF es comandado por una señal adicional a la entrada, llamada *clock* -Ck, pulso de reloj- que habilita la operación de las compuertas.

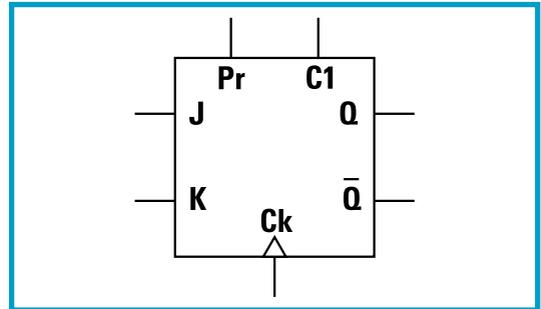
En el funcionamiento **asincrónico**, los FF (excepto el RS) poseen dos entradas adicionales que operan independiente del pulso de reloj, que se denominan *preset* y *clear*. En este caso, la salida corresponde en cada instante al valor asignado al Pr y Cl -asumiendo que Pr y Cl activan por cero-.

$$Pr = 0, Cl = 1 \rightarrow Q = 1$$

$$Pr = 1, Cl = 0 \rightarrow Q = 0$$

$$Pr = 0, Cl = 0 \rightarrow Q = X. \text{ Condición prohibida}$$

$$Pr = 1, Cl = 1 \rightarrow Q = \text{Funcionamiento sincrónico}$$



Consideremos un último criterio para clasificar a los biestables, de acuerdo con el disparo del clock.

- **Por nivel.** Cuando el nivel de entrada de sincronismo esté alto (o bajo) $Ck = 1$ (0), entonces las entradas estarán habilitadas para actuar sobre la salida Q^+ del FF. Por lo tanto, Q varía cuando las entradas lo hacen. El estado final de Q (0 o 1) debe leerse en la salida, en el momento que el clock es igual a 0 (1); es decir, cuando el pulso de sincronismo desaparece.

- **Por flanco.** El valor de las entradas está habilitado para actuar sobre Q^+ sólo en el momento en que el flanco ascendente o descendente del pulso de reloj alcanzó su nivel; luego de dicho instante, Q^+ adopta su valor correspondiente.

Contadores

El contador es un circuito secuencial construido con biestables (FF) en cuyas salidas debe ir mostrando una cuenta preestablecida por cada pulso de reloj que ingresa en ellos.

Es posible realizar cuentas ascendentes, descendentes o arbitrarias.

Los contadores se pueden clasificar como:

- Asincrónicos.
- Sincrónicos.

Un contador es **asincrónico** cuando los pulsos a contar se reciben en el flip flop que representa el LSB o bit menos significativo y los FF siguientes son activados de acuerdo a la salida (o la salida negada) del biestable precedente; es decir que el pulso de clock del siguiente se genera con Q ó \bar{Q} del biestable inmediato anterior.

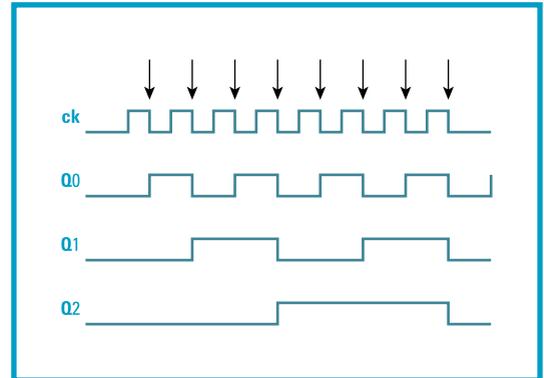
Por ejemplo, si se desea realizar la cuenta $0 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6 \rightarrow 7$ se necesitarán tres FF para poder representar todas las combinaciones binarias.

Este tipo de contador se llama ascendente, cíclico y módulo 8. El módulo M indica que se puede contar hasta $m-1$. Si la cuenta fuese

superior se utiliza mayor cantidad de biestables, uno por cada salida $\Rightarrow Q_0, Q_1, Q_2, Q_3 \dots$

La evolución en el tiempo de las salidas de los FF debe ir dando la cuenta previamente elegida. Suponiendo que el biestable actúa por flanco descendente:

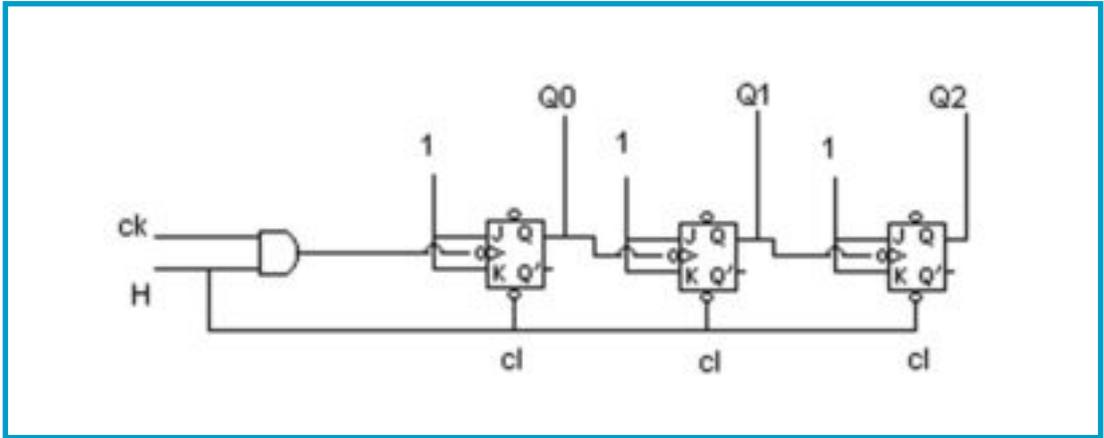
Q_2	Q_1	Q_0	Q_2^+	Q_1^+	Q_0^+	Ck
0	0	0	0	0	1	1
0	0	1	0	1	0	2
0	1	0	0	1	1	3
0	1	1	1	0	0	4
1	0	0	1	0	1	5
1	0	1	1	1	0	6
1	1	0	1	1	1	7
1	1	1	0	0	0	8



Como se puede observar, cada una de las salidas genera un clock para ser usado -si es necesario- de frecuencia múltiplo del reloj original; es decir, se genera un divisor de frecuencia.

Q_0 genera un divisor en 2, Q_1 en 4, Q_2 en 8, ...

Un circuito que implementa en forma genérica el contador es:



H es una habilitación externa que activa los *clear* con H=0 y permite el conteo cuando H=1.

Hemos elegido FF-JK, pero pueden utilizarse FF-T. Las entradas J = K = 1 permiten el cambio de estado al recibir el flanco descendente del pulso de reloj.

Existen variaciones. Por ejemplo, para una cuenta descendente, con H = 0 se activan los *preset* de cada biestable y se debe cambiar el flanco de activación del clock. Esto se logra modificando la conexión de ck a \bar{Q} lo que da como cuenta: 7 → 6 → 5 → 4 → 3 → 2 → 1 → 0

Los contadores **asincrónicos** son aquellos que comienzan y finalizan en cualquier combinación, avanzando siempre de a una cuenta.

El comienzo se activa con los *Pr* y *Cl* de cada FF y a través de H. Por ejemplo, si se desea comenzar de $Q_2 = 0, Q_1 = 1, Q_0 = 1$ entonces se asignan Pr_0, Pr_1, Cl_2 . El retorno se realiza con una compuerta *nand* y la combinación elegida, activando nuevamente los *Pr* y *Cl* de los FF para que el contador sea cíclico.

Para que no sea cíclico y memorice el último estado de cuenta, en lugar de activar los *Pr* y *Cl* se inhabilita la entrada de reloj.

En el caso de **contadores sincrónicos** los pulsos de reloj de entrada a contar ingresan en todas las entradas de clock simultáneamente. Éstos permiten realizar cualquier tipo de cuenta ascendente o descendente comenzando y finalizando en cualquier valor, y hasta obviar combinaciones.

Para proyectar un contador de este tipo se procede como sigue:

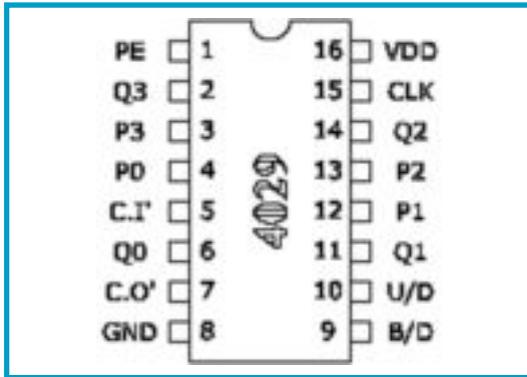
- Se escribe la tabla de verdad
- Se reduce por mapas de Karnaugh a cada Q^+
- Se escribe la función y se implementa

En la actualidad, está disponible gran variedad de circuitos integrados contadores que no sólo realizan la generación de secuencias binarias de conteo, sino que poseen otras funciones adicionales -como opciones de selección de secuencia ascendente o descendente, borrado e inicialización, reset o vuelta a cero, elección de conteo binario o de déca-

da (las 10 primeras combinaciones en BCD) entre otras-, todas relacionadas con la aplicación o modo de funcionamiento y la configuración.

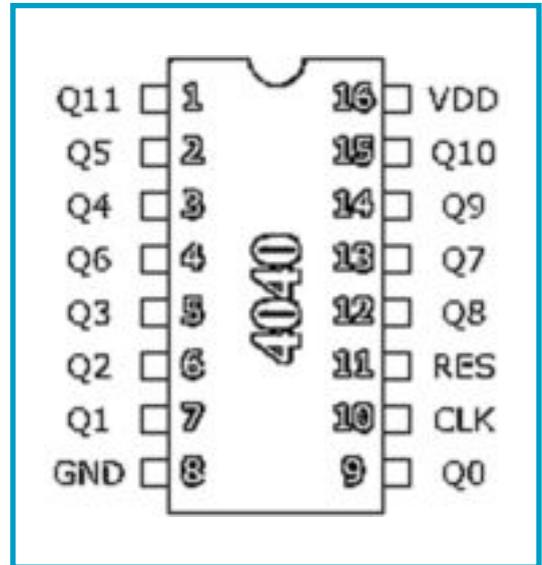
Tomemos como ejemplos los siguientes circuitos integrados contadores, describiendo algunos de sus pines:

Contador binario de décadas Up/Down CD 4029:



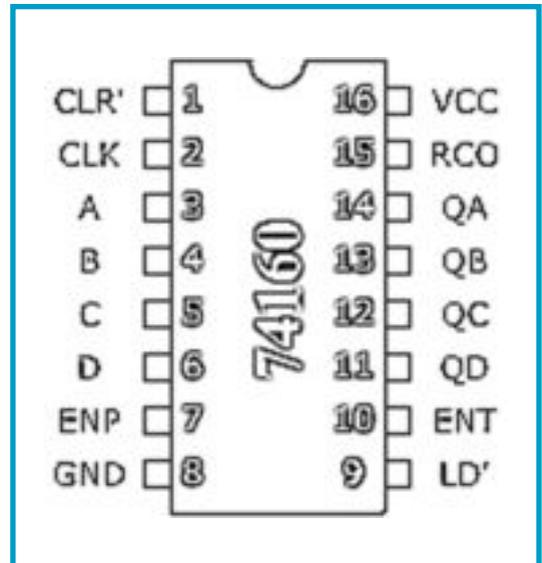
PE	Preset Enable: Esta entrada se utiliza para cargar los datos Pi en las salidas Qi.
P0...P3	Presets: Entradas de fijación. Se utilizan en forma conjunta con PE.
Q0...Q3	Salidas binarias.
C.I'	Carry In. Entrada para detener la secuencia. En 0, cuenta; en 1, se detiene.
C.O'	Carry Out. Salida para indicar rebasamiento. 0 sin acarreo, 1 con acarreo.
B/D	Bin/Dec: Selección del tipo de funcionamiento. 1 Cuenta en Binario, 0 Cuenta en década.
U/D	Up/Down: Entrada de selección de secuencia. 1 Ascendente, 0 descendente.
Vdd, Gnd	Alimentación.

Contador binario (12 bits) CD 4040:



Q0 ... Q11	Salidas binarias.
CLK	Clock: Entrada de reloj.
RES	Reset: Entrada de borrado.
Vdd, Gnd	Voltaje, Ground: Alimentación.

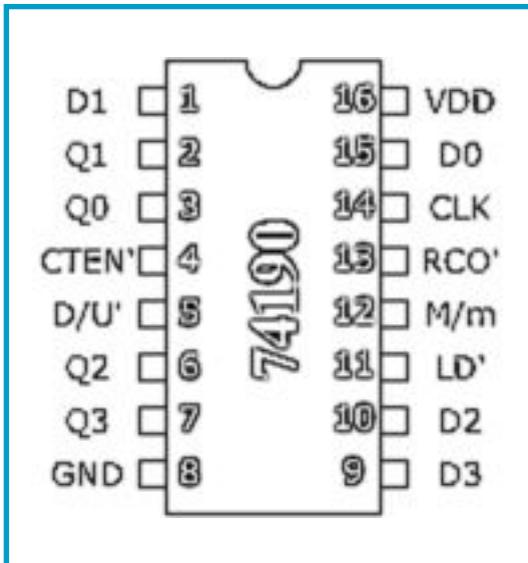
Contador de décadas (4 bits) SN74160:



CLR'	Clear: Entrada de borrado. En cero, inicializa todas las salidas a cero.
CLK	Clock: Entrada de reloj.
A, B, C, D	Entradas de datos paralelo.
ENP, ENT	Entradas de mantenimiento de cuenta.
LD'	Load: Entrada de habilitación de carga de datos. Se usa de forma conjunta con las entradas A, B, C, D.
QA, QB, QC, QD	Salidas.
RCO	Salida de acarreo. Cuando el contador llega al último dígito, se genera un acarreo (salida en 1).

D0, D1, D2, D3	Entradas paralelo.
CTEN'	Count Enable: Entrada de habilitación para el contador.
D/U'	Down/Up: Entrada de selección de la secuencia, ascendente o descendente.
Q0, Q1, Q2, Q3	Salidas binarias.
LD'	Load: Entrada de carga de los datos presentes en D0...D3.
M/m	Max: Salida de señalización de acarreo del contador. En 1 indica que hay carry al pasar el número 1001 en modo ascendente o cuando alcanza el 0000 en modo descendente.
RCO	Salida de propagación para conexión en cascada con otros contadores.
CLK	Clock: entrada de reloj.
VDD, GND	Alimentación.

Contador de décadas Up/Down (4 bits) SN74190:



Registros de desplazamiento

En el manejo de información de tipo digital podemos encontrarnos con la necesidad de almacenar datos temporalmente, manteniéndolos un cierto tiempo, quizás realizar un cambio en algún bit intermedio almacenado y, así, luego de modificados, procesarlos o enviarlos a otro sitio.

Los dispositivos digitales donde se realiza este almacenamiento temporal son los **registros de desplazamiento**. Son cir-

Por lo general, se suele llamar **registro** a un conjunto de 8 o más *biestables*.

cuitos secuenciales que se diseñan con flip-flops o biestables donde se almacena un bit de la palabra binaria total o dato a procesar; en ellos, las informaciones pueden ser desplazadas globalmente, en uno u otro sentido, según la señal sincrónica de clock o reloj lo determine. Así, podemos notar que estos registros tienen dos características fundamentales: memoria y desplazamiento de información.

Los registros desplazamiento se confeccionan con flip-flops tipo D y tipo JK. Los más comunes son los de 8 bits, porque se los relaciona con el **byte** de información.

Un **registro de desplazamiento básico** es entonces un conjunto de biestables conectados para que los números binarios almacenados en él sean desplazados de un flip-flop al siguiente, cada vez que el pulso de reloj aplicado lo indique. En la figura podemos apreciar un ejemplo, conectando 9 biestables tipo D perteneciente al circuito integrado SN74164.

Para el desplazamiento de los datos se puede elegir el flanco ascendente o descendente del clock.

Existen distintos tipos de entradas y salidas en los registros de desplazamiento, que están relacionadas con las distintas formas de cargar o extraer información en el registro.

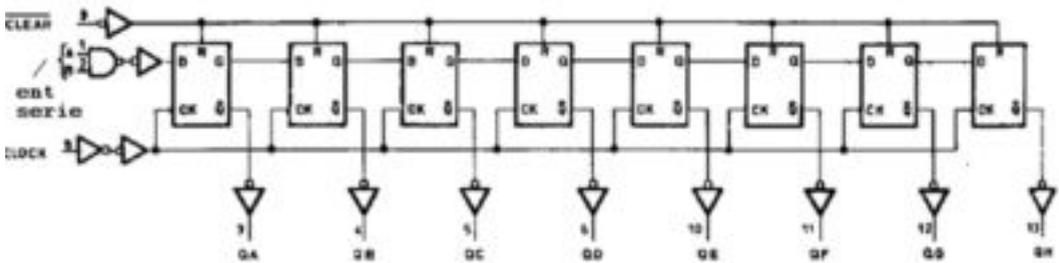
Las combinaciones de Entrada/Salida de datos más comunes en los registros de desplazamiento son:

- **Entrada serie / Salida paralelo.**

En este tipo de registro, la entrada es única y los datos llegan de a uno por allí, para almacenarse desplazándose; en la salida están todos los bits del registro disponibles. Poseen, además, una entrada adicional asincrónica (activada cuando el usuario lo desee) CLR que es usada para poner un 0 lógico en todos los bits del registro.

Un ejemplo de circuito integrado que tiene este funcionamiento es el **74HC164**.

Verifique cómo la salida de un biestable se conecta a la entrada del siguiente.



Registro de desplazamiento básico

- **Entrada paralelo / Salida serie:** En este tipo, cuando se activa el pin identificado como load, las entradas en paralelo se almacenan en los flip-flops internos (entrada asincrónica). Cuando se activa en modo *shift*, los bits almacenados se desplazan hacia la derecha según los pulsos de clock (entrada sincrónica), y comienzan a aparecer por la salida serie.

En el mercado encontramos circuitos integrados como el 74HC165 que funcionan según lo descrito.

de elegir la dirección en que se transmiten los datos a través de una señal de control que permite seleccionar el sentido de desplazamiento de los datos.

Tomaremos el **circuito integrado 74HC374** de entrada paralelo y salida paralelo que posee 8 flip-flops tipo D con salidas tipo buffer con clock de activación por flanco ascendente.

- D0...D7: Entrada paralelo
- Q0...Q7: Salida paralelo
- OE: Output enable

Existen **registros de desplazamiento bidireccionales**. Este tipo de registro tiene la opción

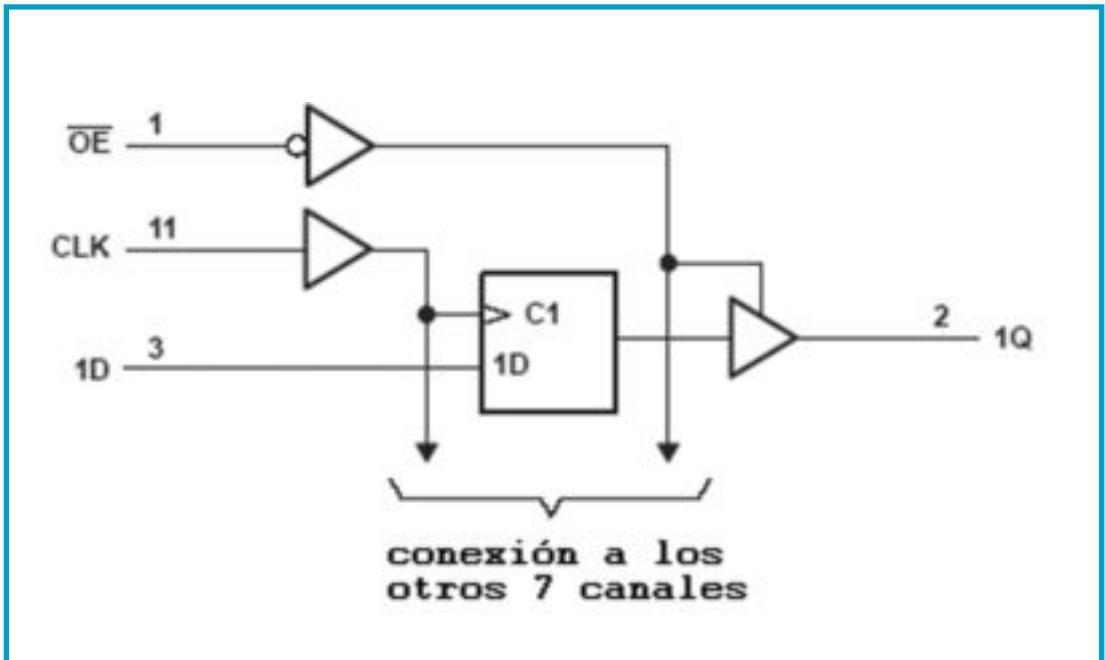


Diagrama lógico del CI74HC374

En esta tabla podemos ver el funcionamiento de cada uno de los biestables que forman el registro:

Entradas			Salidas
\overline{OE}	CLK	D	Q
L	↑	H	H
L	↑	L	L
L	H or L	X	Q0
H	X	X	Z

Los registros de desplazamiento tienen varias aplicaciones; entre ellas podemos mencionar:

- Transmisión de datos.
- Conversión de protocolo serie en paralelo y viceversa.
- Puertos de salida de los microcomputadores o microcontroladores.
- En circuitos secuenciadores.

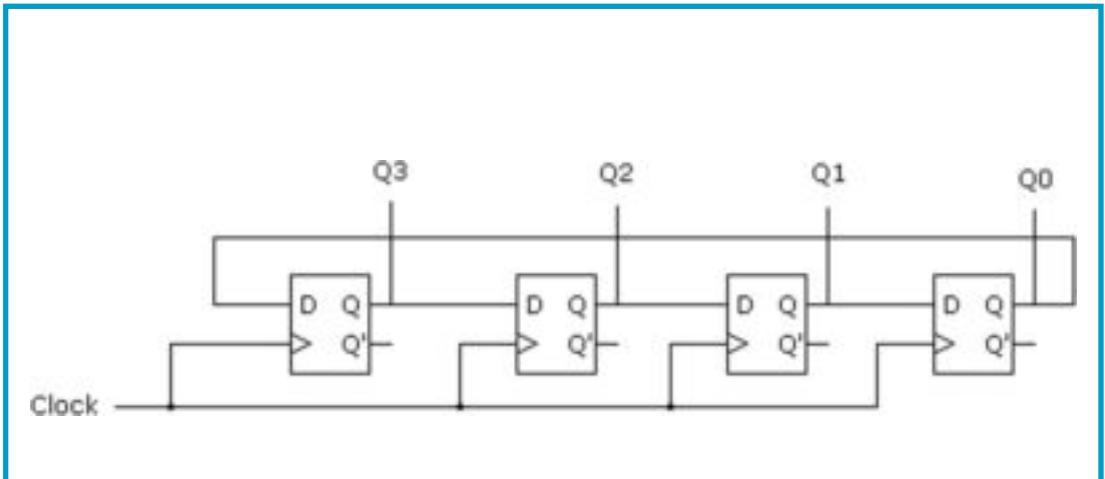
- Para realizar multiplicaciones y divisiones por 2, 4, 8, 16 bits.

Su funcionamiento puede ser implementado mediante un programa y simulado en el sistema de simuladores que diseñamos y armamos.

Varios de los circuitos integrados mencionados forman parte del hardware de resolución que se adiciona al equipo.

Contador en anillo

El contador en anillo es un registro de desplazamiento básico en el cual los datos se reinyectan a la entrada; es decir, se conecta la salida Q del último a la entrada del primero y, así, la información se desplaza y rota en forma de anillo mediante la interconexión de los biestables extremos.



Contador en anillo de 4 bits

Como podemos ver, la construcción es muy sencilla, a partir de un registro de desplazamiento básico. Pero, recordemos que también existen contadores de este tipo en forma integrada en el mercado.

Otro tipo de contador en anillo es el contador Johnson, que tiene un funcionamiento muy similar al contador en anillo visto; excepto que, en él, que la conexión del último al primer biestable se hace a través de una compuerta inversora -podría, también, hacerse con la entrada \bar{Q} -.



Todos los circuitos contadores presentados pueden ser fácilmente realizables con una programación relativamente sencilla y verificar los estados de cuenta con led indicadores conectados a los puertos del equipo simulador propuesto.

De la misma manera, pueden ser realizados y probados circuitos de registro de desplazamiento, temporizaciones, retardos y generación de ondas cuadradas de oscilación.

Además, la realización de estos circuitos en lógica digital discreta ayuda, en gran manera, a la comprensión del funcionamiento de los contadores internos de un microcontrolador y, así, facilita su programación.



Le recomendamos consultar los espacios:

- TTL Databooks. En Texas Instruments: www.ti.com y education.ti.com/educationportal
- CMOS Databooks. En National Semiconductors: www.national.com

Microcontroladores³

Los microcontroladores son parte fundamental de diversos sistemas de nuestra vida cotidiana y laboral: una multiprocesadora de alimentos, hornos microondas, un lavarropas automático, un teléfono, etc. y de otras aplicaciones tales como instrumentación electrónica o el control de una planta industrial.

Recibe el nombre de **controlador** el dispositivo que se emplea para el gobierno de uno o varios procesos. Por ejemplo, el controlador que regula el funcionamiento de un horno dispone de un sensor que mide constantemente su temperatura interna y, cuando ésta traspasa los límites prefijados, genera las señales adecuadas que accionan los actuadores que intentan llevar el valor de la temperatura dentro del rango estipulado.

En la actualidad, el desconocimiento de esta herramienta de trabajo nos apartaría del avance electrónico continuo que vivimos.

La implementación física de los controladores ha variado considerablemente, desde los controladores que se construían con componentes de lógica digital básica (por ejem-

³ Le recomendamos profundizar en:

- Angulo Usategui, José María; Angulo Martínez, Ignacio. *Microcontroladores PIC. Diseño práctico de aplicaciones*. Mc Graw-Hill.
- Edison Duque (1997) *Curso básico de microcontroladores PIC*. Cekit. Compañía Editorial Tecnológica.
- Edison Duque (1997) *Curso avanzado de microcontroladores PIC*. Cekit. Compañía Editorial Tecnológica.
- Curso de microcontroladores PIC 1º nivel (2004) *Revista Web Electrónica*. Club SE (Saber electrónica). Quark. <http://webelectronica.com.ar>
- Hoja de datos. Microchip PIC 16X84. www.microchip.com

plo: compuertas); hasta los que empleaban microprocesadores con integrados adicionales de memoria y E/S (entrada/salida) sobre un mismo circuito impreso. Actualmente, todos los elementos del controlador se incluyen en un único circuito integrado llamado **microcontrolador**.

Un **microcontrolador** es un solo circuito integrado que incorpora la mayor parte de los elementos que conforman un controlador y que se utilizan con el fin de solucionar un problema determinado.

Una presentación básica de las partes que conforman un microcontrolador es:

- Procesador o CPU -unidad central de proceso-.
- Memoria RAM para contener los datos.
- Memoria para el programa tipo ROM/PROM/EEPROM/FLASH.
- Líneas de E/S para la comunicación con el exterior.
- Diversos módulos para el control de periféricos: temporizadores, puertos serie y paralelo, conversores analógico/digital y digital/analógico, etc.
- Generador de pulsos de reloj que sincronizan el funcionamiento de todo el sistema.

Algunas de las ventajas de utilizar un microcontrolador en los diversos sistemas son:

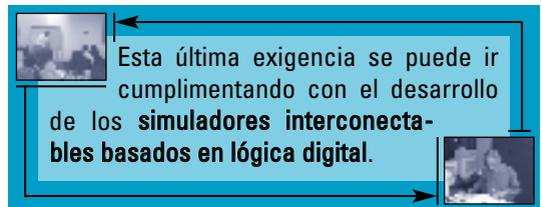
- Aumenta las prestaciones, dado que el control lo realiza un solo elemento.
- Mayor fiabilidad con reducción de

tamaño, dado que al estar integrado en un solo chip reduce drásticamente la cantidad de componentes, disminuyendo los problemas de ruido y aumentando el tiempo medio entre fallas (ya que el mismo es función inversa a la cantidad de componentes de la placa).

- Mayor flexibilidad, debido a que para el control de las variables no es necesario modificar el circuito eléctrico sino, simplemente, variar la programación del microcontrolador.

Podemos mencionar también algunos requerimientos:

- Exige del usuario un nivel de preparación acorde.
- Requiere disponer y conocer las distintas herramientas de desarrollo relacionadas con la tarea.

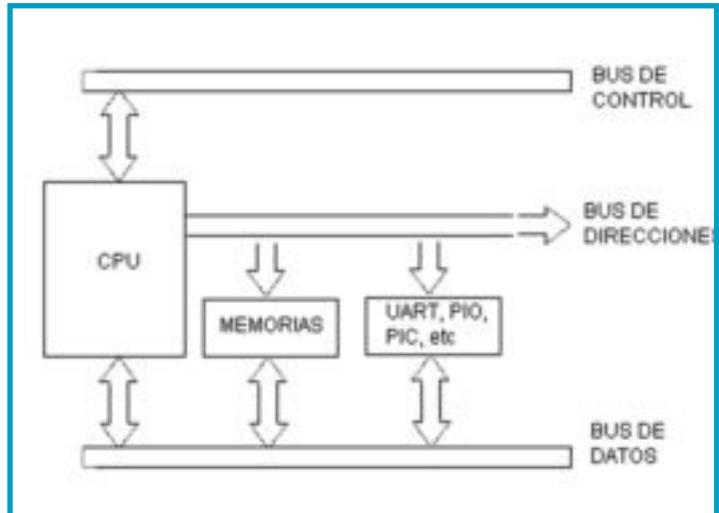


¿Cuál es la diferencia entre microprocesador y microcontrolador?

El **microprocesador** es un circuito integrado que tiene por misión ejecutar un programa para realizar ciertos cálculos y tomar algunas decisiones. Esta tarea es realizada por un conjunto de componentes denominado unidad central de proceso -CPU-.

La CPU está formada por:

- El decodificador de instrucciones, que interpreta las instrucciones del programa a ejecutar.
- El generador de impulsos de control que ordena las tareas.
- La unidad aritmética lógica -ALU- que realiza las comparaciones y los cálculos.
- Los registros de propósito general, donde se almacenan los resultados temporales y se mantiene el estado del microprocesador.
- Los buses o líneas de dato, dirección y control que permiten la conexión al exterior del microprocesador formando un sistema.

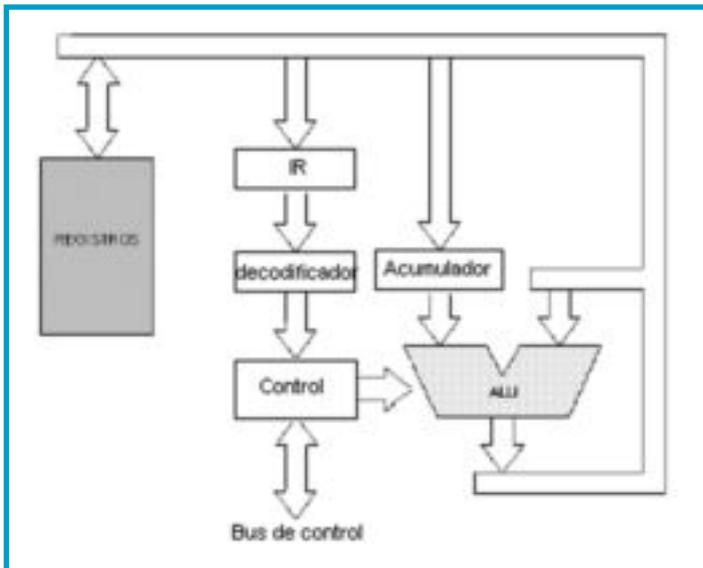


Estructura de sistema basado en un microprocesador

La misma evolución de la electrónica planteó la posibilidad de integración de todas las partes de este sistema en un solo chip; nace, así, el microcontrolador. Éste permite aliviar la tarea de quien se ocupa del desarrollo de sistemas de control, dado que:

- Sólo requiere conocer en forma completa al microcontrolador.
- No resulta necesario conocer las características de cada interface o periférico usado.
- Se puede elegir el microcontrolador más adecuado, de acuerdo con la aplicación a realizar.

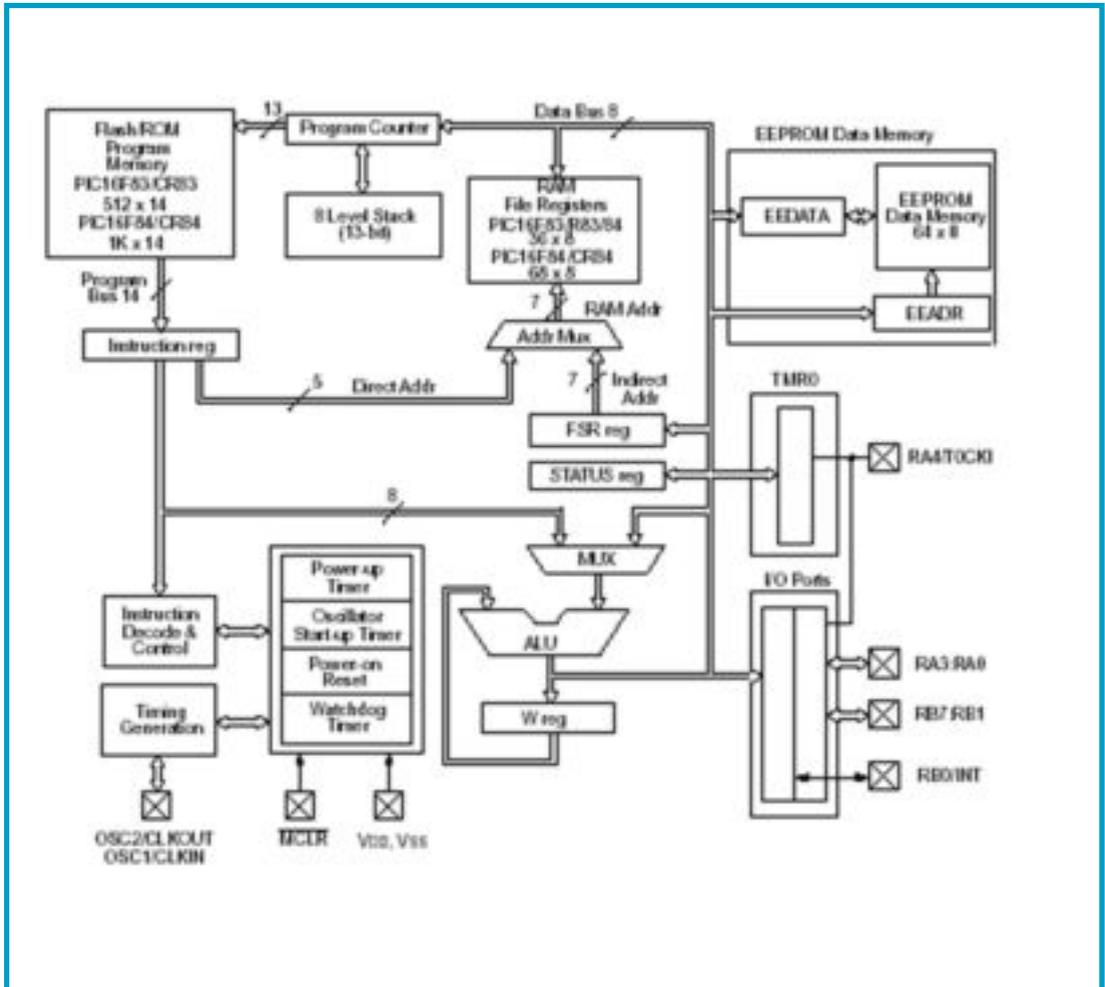
En referencia a esto último, podemos decir que existen diversos modelos de microcontroladores, de diferentes fabri-



Estructura de un microprocesador

cantes, de los más sencillos hasta los más complejos, ofreciendo cada fabricante y cada modelo distintas prestaciones. Dentro de cada familia de microcontrolador podemos encontrar distintos tipos de características de acuerdo a la utilidad. La tarea del diseñador es conocerlas y elegir el dispositivo que más se adapte a la aplicación a realizar.

Si sólo existiera un único modelo de microcontrolador, éste debería ser tan complejo que resultaría poco económico y de difícil manejo para el usuario.



Todas las partes que componen el microcontrolador están contenidas en su interior; sólo salen al exterior las líneas que gobiernan los periféricos

Cada vez existen más productos que incorporan un microcontrolador con el fin de aumentar sus prestaciones, reducir su tamaño y costo, mejorar su fiabilidad y disminuir el consumo.

Tanto la producción en gran escala de las distintas familias de microcontroladores, como la incorporación de nuevos componentes o integrantes, da idea del uso masivo que tiene este componente dentro la electrónica y su constante expansión.

La gran diversidad de microcontroladores se fabrica prácticamente en su totalidad con tecnología CMOS -*Complementary Metal Oxide Semiconductor*-. Esta tecnología supera a las otras tecnologías por su bajo consumo y su alta inmunidad al ruido.

Cuando debemos elegir el microcontrolador a utilizar en un diseño concreto, es necesario considerar una serie de factores tales como:

- **Costos de las herramientas de desarrollo.** Consideramos las herramientas de apoyo tales como: emuladores, simuladores, ensambladores, compiladores, etc.
- **Velocidad.** Tenemos en cuenta la velocidad de procesamiento de datos.
- **Capacidad de memoria.** Es imprescindible realizar una estimación de cuánta memoria es necesaria.
- **Número de entradas salida.** Es conveniente dibujar el hardware en bloques del sistema a desarrollar, para identificar la cantidad y tipo de señales a controlar.
- **Número de bits de datos.** Los requeri-

mientos de la aplicación imponen seleccionar un microcontrolador de 8 bits o un byte, 16 o 32 bits. Este último, por su costo, se utiliza en aplicaciones que requieran altas prestaciones.

- **Características especiales:**
 - Conversor analógico-digital y digital-analógico.
 - I²C.
 - Cantidad de temporizadores y número de bits con que operan.
 - Cantidad de interrupciones.
 - Memorias EEPROM de datos.
 - Interfaces de comunicación, por ejemplo: RS232, USB, etc.
- **Consumo.** Es conveniente que el microcontrolador consuma la menor cantidad de energía posible; para ello se recomienda que esté en estado de bajo consumo y que se active ante la presencia de una señal -por ejemplo, una interrupción-, ejecutado el programa como respuesta a ella.

Algunas de las familias de microcontroladores para tener presentes a la hora de elegir son:

Intel®, dentro de la cual podemos mencionar como uno de los más usados el 8051, de fácil programación y con variadas herramientas de desarrollo asociada. Otros son el 80186, 80188 y 80386, en los cuales, la principal ventaja es el aprovechamiento de las herramientas de desarrollo para PC.

Freescale® (ex Motorola). Algunos de los integrantes de esta familia son, por ejemplo: el 68HC11 que es un microcontrolador de 8

bits, el 683xx que incorpora algunos periféricos con mayores prestaciones, etc.

Microchip®. Es una familia de microcontroladores económicos y de gran popularidad. Sus integrantes se conocen mundialmente como PIC *-Programmable Interfaces Controller-*, los primeros microcontroladores RISC.

En nuestro equipo de simuladores optamos por un microcontrolador porque, para el trabajo en el aula, no siempre disponemos de diversos programadores y simuladores ajustados a los distintos tipos y familias. En nuestro caso, hemos optado por los microcontroladores de Microchip® porque son fácilmente accesibles en nuestro medio y, además, porque tienen un costo poco significativo para las diferentes experiencias que se realizarán con los simuladores.

Desde el punto de vista de los bloques, la estructura interna que conforma los microcontroladores es similar en las distintas familias. Así, podemos distinguir en todas ellas los siguientes bloques constitutivos:

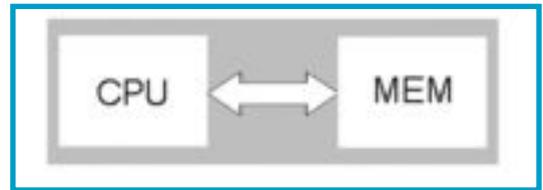
- Procesador o CPU.
- Memoria -de datos, de instrucciones-.
- I/O ports (puertos de entrada y de salida).
- Reloj -interno o externo-.
- Módulos especializados.

El último ítem abarca la gran variedad de microcontroladores que existe y da cuenta de la competencia entre los distintos fabricantes por desarrollarlos cada vez más potentes y con módulos más especializados.

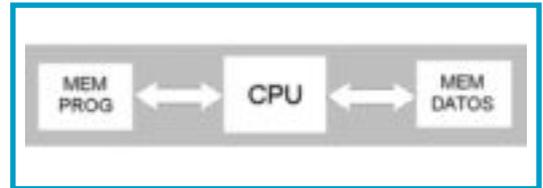
Desde el punto de vista del funcionamiento

interno de un microcontrolador, existen dos tipos de arquitectura: la clásica de von Neumann utilizada por Freescale, y la arquitectura Harvard usada por Microchip, por ejemplo.

La **arquitectura de Von Neumann** es utilizada por *Motorola®* (ahora, Freescale). En esta forma de trabajo existe un solo ámbito de memoria donde los datos y programas se guardan, y se utiliza un solo camino (bus de direcciones, datos y control) para acceder a dicho ámbito.



La **arquitectura Harvard**, a diferencia de la anterior, dispone de dos ámbitos distintos de memoria (datos e instrucciones) y accede a cada banco por caminos (buses) distintos.



Los microcontroladores PIC -a los que nos referimos más adelante, al centrar nuestro análisis en los **simuladores interconectables basados en lógica digital-** responden a la arquitectura Harvard.

Ahora analizaremos, en forma separada, cada uno de los bloques mencionados.

Procesador o CPU. Es el responsable de la

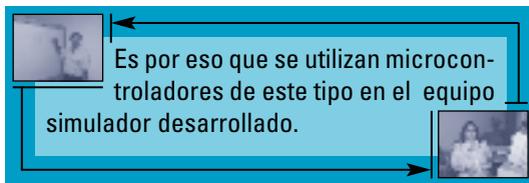
interpretación y la ejecución de las instrucciones que le llegan a través del bus, direccionando la memoria de instrucciones, recibiendo el código de operación de la instrucción en curso, decodificándola y ejecutando la operación que implica la instrucción. Estas instrucciones pueden ser simples o complejas en su forma. Aquellos microcontroladores que optan por las instrucciones simples reciben el nombre de RISC -*Reduce Instruction Set Computer*-; los que trabajan con instrucciones complejas se denominan CISC -*Complex Instruction Set Computer*-.

Memoria. Hoy en día, la memoria más utilizada es la FLASH, que permite la regrabación por medios eléctricos, hasta un millón de veces; esta memoria es la utilizada en la mayoría de los microcontroladores. Esta posibilidad otorga al sistema una gran flexibilidad, siendo muy útil a la hora de ensayar un desarrollo.

Otro tipo de memoria es la OTP -*One time programmable*-. Se recomienda utilizarla sólo cuando se tiene la certeza que el programa está optimizado, en la fase de producción. La memoria OTP puede ser grabada una sola vez.

En los microcontroladores existe, además, una memoria RAM que guarda las variables y los datos. Algunos también poseen memorias tipo EEPROM.

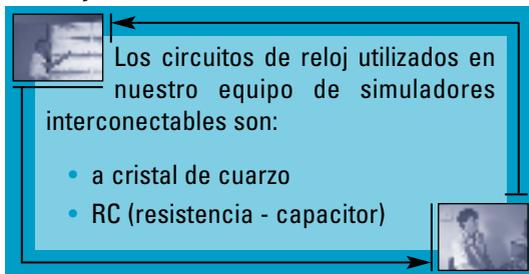
Las memorias EEPROM y FLASH son muy útiles al permitir que los microcontroladores que las incorporan puedan ser reprogramados "en circuito"; es decir, sin tener que sacar el circuito integrado de la tarjeta.



I/O ports (puertos de entrada y de salida). La cantidad de líneas de entradas/salidas que comunican al microcontrolador con los periféricos está en relación directa con la potencia de este chip. Es decir, aquella de menor cantidad de pines I/O tendrá menor cantidad de prestaciones y viceversa.

A la hora de realizar un proyecto, es muy importante evaluar la cantidad de I/O necesarias; éste es uno de los puntos a considerar en la elección del microcontrolador. Según la configuración interna del microcontrolador, cada una de estas líneas se puede utilizar como entrada, salida o control.

Reloj -interno o externo-. La onda cuadrada originada por el reloj se utiliza en la sincronización de todas las operaciones del sistema. En los microcontroladores modernos, el reloj puede estar dentro del chip; su frecuencia es fijada por el fabricante y, opcionalmente, permiten adicionar un reloj externo que establece la frecuencia en el valor que el diseñador desee.



Módulos especializados. Estos componentes son los que marcan la diferencia entre un microcontrolador y otro de una misma fami-

lia. Existen módulos básicos que se encuentran en la mayoría de los microcontroladores:

- **Timers -temporizadores-**. Estos registros son usados para contabilizar eventos que suceden en el exterior (contadores) y controlar los tiempos. Para medir el tiempo, se carga un registro con un valor previamente calculado; éste se va incrementando o decrementando, según los pulsos de reloj, hasta que se desborda y llega a 0, avisando de ello al microcontrolador. La cantidad de estos módulos y el número de bits de cada uno de ellos es importante a la hora de elegir un microcontrolador.
- **Watchdog -perro guardián-**. Casi todos los microcontroladores lo poseen. Se ocupa de vigilar que el programa no entre en un *loop -bucle* o *ciclo-* del cual no pueda salir. Resulta muy útil; sobre todo, al momento de realizar un sistema de control o uno que no pueda darse el lujo de dejar de vigilar los parámetros de cierto proceso.
- **Protección ante fallo de alimentación.** Por medio de este circuito se protege al microcontrolador, además de asegurar que empiece de nuevo el programa, reseteándolo si es que ocurre esta falla. Esta protección se activa cuando el voltaje de alimentación (VDD) es inferior a un voltaje mínimo; así, comienza a funcionar normalmente cuando sobrepasa este valor.
- **Estado de reposo o de bajo consumo.** En

ciertas aplicaciones es útil colocar al microcontrolador en un estado de bajo consumo de energía; sobre todo, si el sistema será usado con baterías, esperando, sin hacer nada, a que se produzca algún evento externo que lo ponga de nuevo en funcionamiento. En este estado, se detiene el reloj principal y se "congelan" sus circuitos asociados.

- **Convertor A/D.** Ciertos miembros de la familia tienen módulos especializados como éste, mediante el cual se pueden realizar conversiones de una cantidad analógica a una digital de 10 bits. Suelen, además, disponer de un multiplexor que permite aplicar diversas señales analógicas a la entrada del convertor.
- **Convertor D/A.** Este módulo es el opuesto al anterior; de la cantidad digital nos permite conseguir el valor analógico correspondiente.
- **Modulador por ancho de pulsos - PWM-**. En una de las salidas del microcontrolador, este módulo especializado ofrece la posibilidad de conseguir pulsos de ancho variable, de acuerdo a condiciones de entrada.
- **Puertos de E/S.** Todos los microcontroladores poseen líneas de E/S asociadas a un grupo de pines, formando puertos que, mediante un registro interno, se pueden configurar como entrada o salida. La cantidad de líneas que existe define al microcontrolador y resulta un parámetro importante a la hora de elegirlo.

- **Puertos de comunicación.** Mediante este módulo logramos la comunicación del microcontrolador con otros dispositivos externos, inclusive con una PC. Existen ciertos estándares como son:

- UART, adaptador de comunicación serie asincrónica.
- USART, adaptador de comunicación serie sincrónica y asincrónica.
- USB *-Universal Serial Bus-*, que es un moderno bus serie para las PC.
- Bus I²C, que es una interfaz serie de dos hilos.

Herramientas para el desarrollo de aplicaciones

Uno de los factores más importantes para seleccionar un microcontrolador para un determinado proyecto es el soporte, tanto de software como de hardware, que dispone; es decir optar por un buen conjunto de herramientas de desarrollo.

Las principales herramientas de desarrollo de sistemas basados en microcontroladores son:



Ensamblador	La programación en lenguaje ensamblador, a veces compleja, permite desarrollar programas eficientes, pues el programador posee dominio de todo el sistema. Los fabricantes suelen proporcionar el programa ensamblador de forma gratuita (como es el caso de la versión básica del <i>MPLab</i> de Microchip).
Compilador	La programación en un lenguaje de alto nivel (como el C) permite disminuir el tiempo de desarrollo de un producto. Debe programarse cuidadosamente para que no resultar ineficiente. Las versiones suelen ser caras.
Simulador	Son capaces de ejecutar programas realizados para el microcontrolador, en una PC. Los simuladores permiten tener un control absoluto sobre la ejecución de un programa, siendo ideales para su depuración.
Placas de evaluación	<p>Se trata de pequeños sistemas con un microcontrolador ya montado, que suelen conectarse a una PC desde la que se cargan los programas que se ejecutan en el microcontrolador. Las placas suelen incluir visualizadores LCD, teclados, led, fácil acceso a los pines de E/S, etc.</p> <p>El sistema operativo de la placa recibe el nombre de programa monitor. El programa monitor de algunas placas de evaluación, aparte de permitir cargar programas y datos en la memoria del microcontrolador, posibilita realizar ejecución paso a paso en cualquier momento, monitorear el estado del microcontrolador o modificar los valores almacenados los registros o en la memoria.</p> <div data-bbox="370 1175 1164 1380" style="border: 1px solid black; padding: 5px;">  <p>Nuestra placa de simuladores interconectables no contiene un programa monitor. Sí incorpora la capacidad de poder programar el PIC directamente, sin necesidad de utilizar otra distinta, y de simular su funcionamiento directamente para las distintas aplicaciones que se propongan, repitiendo este proceso hasta la depuración total del programa.</p>  </div>
Emuladores en circuito	Se trata de un instrumento que se coloca entre la PC y el zócalo del circuito impreso donde se alojará el microcontrolador definitivo. El programa se ejecuta desde la PC; pero, para la tarjeta de aplicación es como si lo hiciese el mismo microcontrolador que luego irá en el zócalo.

La familia de los PIC como elección

Un **microcontrolador PIC** -*Programmable Interfaces Controller*- es un circuito integrado programable que contiene todos los componentes de una computadora. Se emplea para controlar el funcionamiento de una tarea determinada y, debido a su reducido tamaño, suele ir incorporado en el propio dispositivo al que gobierna. Esta última característica es la que le confiere la denominación de controlador incrustado -*embedded controller*-. El PIC es una computadora con dedicación exclusiva; es decir que, una vez que es programado y configurado, sólo tiene por función realizar la tarea que se le asignó. Para ello, en su memoria reside el programa destinado a gobernar la aplicación y, en sus líneas de entrada-salida, se encuentran conectados los sensores y actuadores del dispositivo a controlar y todos los recursos complementarios disponibles que tienen como única finalidad atender sus requerimientos.

En determinadas aplicaciones, elegir un PIC⁴ es la mejor solución. Los detalles más importantes por los cuales los profesionales se vuelcan a trabajar con PIC son:

- Un juego de instrucciones reducido, lo que implica sencillez de manejo.
- Mucha información, fácil de conseguir y económica.
- Buen precio.
- En general, buenos parámetros: velocidad, consumo, tamaño, alimentación, código compacto, etc.

⁴ Otras familias de microcontroladores pueden ser más eficientes en una aplicación específica; especialmente, si en ellas predomina una característica concreta.

- Herramientas de desarrollo fáciles y económicas. Muchas de éstas, accesibles en Internet.
- Gran variedad de herramientas hardware que permiten grabar, depurar, borrar y comprobar el comportamiento de los PIC.
- Diseño rápido.
- Gran variedad de modelos de PIC, lo que permite elegir el que mejor responde a los requerimientos de la aplicación.

Le recomendamos explorar con sus alumnos la página web de Microchip®:

www.microchip.com

Otra empresa de microcontroladores (que no son PIC) es Freescale®:

www.freescale.com

Otra ventaja de usar un PIC es que, una vez que se conoce su arquitectura y su repertorio de instrucciones, es muy fácil emplear otro modelo.

Las características relevantes de los PIC:

Arquitectura. La arquitectura del procesador sigue el modelo Harvard. La CPU se conecta de forma independiente y con buses distintos con la memoria de instrucciones y con la de datos. La arquitectura Harvard permite a la CPU acceder, simultáneamente, a las dos memorias de instrucciones y datos, lo que posibilita la adecuación del tamaño de las palabras y los buses a los requerimientos específicos de las instrucciones y los datos.

El procesador de los microcontroladores más moderno responde a la arquitectura RISC que posee un repertorio de instrucciones-máquina pequeño (aproximadamente,

35), de forma que la mayor parte de las instrucciones se ejecuta en un ciclo de instrucción.

Otro aporte que aumenta el rendimiento de la CPU consiste en la segmentación del procesador *-pipe-line-*, descomponiéndolo en etapas para poder procesar una instrucción diferente en cada una de ellas y trabajar con varias a la vez.

Formato de las instrucciones. El formato de todas las instrucciones es de la misma longitud. Algunas de las variedades de PIC suelen tener instrucciones de una longitud de 12 bits y, otras, de 14 bits o más. Esta característica es muy ventajosa en la optimización de la memoria de instrucciones.

Instrucciones ortogonales. Cualquier instrucción puede manejar cualquier elemento de la arquitectura como fuente o como destino.

Arquitectura basada en un banco de registros. Todos los objetos del sistema (puertos de E/S, temporizadores, posiciones de memoria, etc.) están implementados físicamente como registros. Cada modelo de microcontroladores integra prestaciones y recursos diferentes; la gran variedad de modelos de microcontroladores PIC permite que el usuario pueda seleccionar el más conveniente para su proyecto.

Instrumental de trabajo. Siempre que se diseñe con circuitos integrados programables se requiere herramientas para la puesta a punto del hardware y del software. Con referencia al software, además de los compiladores es interesante disponer de simuladores que per-

mitan la ejecución de instrucciones, representando el comportamiento interno del procesador y el estado de las líneas de E/S. Respecto del hardware, una herramienta indispensable es el grabador, parte integrante de nuestro equipo programador y simulador interconectable, basado en lógica digital.

A través de Internet, Microchip pone a disposición de sus usuarios, libremente, ensambladores como el MPASM y simuladores como el MPSIM.
www.microchip.com

Las gamas de PIC⁵

Una tarea importante a la hora de realizar un proyecto determinado es la elección del microcontrolador adecuado para ese fin, tratando que sea, además, el de más bajo presupuesto.

Existen distintas gamas; cada una de las cuales se diferencia por las características de: tamaño, número de pines, alimentación, consumo, encapsulado, líneas de entrada-salida, frecuencia máxima, memoria de programa, memoria de datos, timers, etc.

A continuación presentamos un resumen de los diversos microcontroladores PIC más importantes:

⁵ Hemos optado por describir productos de Microchip®, que son los que integramos en el equipo programador y simulador interconectable, basado en lógica digital. Usted y sus alumnos pueden, perfectamente, decidirse por otra alternativa de equipamiento.

MICROCONTROLADORES PIC⁶

	Memoria	Bytes	Words	EEPROM Data memory	RAM	IO Pins	ADC	Comparador	AmpOp	Timers/ WDT	USB	Máx vel MHz	Vdd mín	Vdd máx	Encapsulado
PIC 12F509A	Flash	1536	1024	0	41	6	0	0	No	1-8bit 0-16bit 1-WDT	No	4			8/MSOP 8/PDIP300 8/SOIC 150mil 8/SOIC150 8/SOIC208
PIC 12F629	Flash	1792	1024	128	64	6	0	1	No	1-8bit 1-16bit 1-WDT	No	20			8/DFN 8/PDIP 8/PDIP 300mil 8/PDIP300 8/SOIC 150mil 8/SOIC150
PIC 12F635	Flash	1792	1024	128	64	6	0	1	No	1-8bit 1-16bit 1-WDT	No	20			8/DFN 8/PDIP300 8/SOIC 150mil 8/SOIC150
PIC 12F675	Flash	1792	1024	128	64	6	4/10 Bit	1	No	1-8bit 1-16bit 1-WDT	No	20			8/DFN 8/PDIP 8/PDIP 300mil 8/PDIP300 8/SOIC 150mil 8/SOIC150
PIC 12F683	Flash	3584	2048	256	128	6	4/10- Bit	1	No	2-8bit 1-16bit 1-WDT	No	20			8/DFN 8/PDIP 8/SOIC 150mil 8/SOIC150 14/PDIP 300mil 80/TQFP

En esta selección observamos que se listan los 12FXXX y 16FXXX dado que utilizan memoria FLASH, que como ya aclaramos, se pueden grabar y borrar muchas veces, lo que facilita las diversas pruebas a las que se verán

sometidos los programas de los distintos proyectos. Los otros tipos de PIC que usan memorias OTP son más utilizados en producción.

⁶ www.microchip.com

	Memoria	Bytes	Words	EEPROM Data memory	RAM	I/O Pins	ADC	Compa-rador	AmpOp	Timers/WDT	USB	Interface	Máx vel MHz	Vdd mín	Vdd máx	Encapsulado
PIC 16F505	Flash	1536	1024	0	72	12	0	0	No	1-8bit 0-16bit 1-WDT	No		20			14/PDIP300 14/SOIC150mil 14/TSSOP
PIC 16F54	Flash	768	512	0	25	12	0	0	No	1-8bit 0-16bit 1-WDT	No	AUSART AUSART AUSART	20	2	5.5	18/PDIP 18/PDIP300 18/SOIC300mil 18/SOIC300 20/SSOP208mil
PIC 16F57	Flash	3072	2048	0	72	20	0	0	No	1-8bit 0-16bit 1-WDT	No		20			28/PDIP300mil 28/PDIP600mil 28/PDIP600 28/SOIC300mil 28/SOIC300 28/SPDIP 28/SSOP208mil 28/SSOP208
PIC 16F59	Flash	3072	2048	0	134	32	0	0	No	1-8bit 0-16bit 1-WDT	No		20			40/PDIP600 44/TQFP
PIC 16F627A	Flash	1792	1024	128	224	16	0	2	No	2-8bit 1-16bit 1-WDT	No		20			18/PDIP 18/SOIC300mil 18/SOIC300 20/SSOP208mil 28/QFN
PIC 16F628A	Flash	3584	2048	128	224	16	0	2	No	2-8bit 1-16bit 1-WDT	No		20			18/PDIP 18/SOIC300mil 18/SOIC300 20/SSOP208mil 28/QFN
PIC 16F630	Flash	1792	1024	128	64	12	0	1	No	1-8bit 1-16bit 1-WDT	No		20			14/PDIP 14/PDIP300mil 14/PDIP300 14/SOIC150mil 14/SOIC150 14/TSSOP
PIC 16F636	Flash	3584	2048	256	128	12	0	2	No	1-8bit 1-16bit 1-WDT	No		20			14/PDIP300 14/SOIC150 14/TSSOP
PIC 16F639	Flash	3584	2048	256	128	12	0	2	No	1-8bit 1-16bit 1-WDT	No	No	20 20			Please call for package information
PIC 16F648A	Flash	7168	4096	256	256	16		2	No	2-8bit 1-16bit 1-WDT						18/PDIP 18/SOIC300mil 18/SOIC300 20/SSOP208mil 28/PDIP300mil 28/QFN

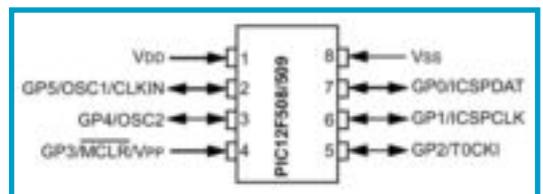
	Memo- ria	Bytes	Words	EEPROM Data memory	RAM	I/O Pins	ADC	Compa- rador	AmpOp	Timers/ WDT	USB	Interface	Máx vel MHz	Vdd mín	Vdd máx	Encapsulado
PIC 16F676	Flash	1792	1024	128	64	12	8/10- Bit	1	No	1-8bit 1-16bit 1-WDT	No		20			14/PDIP 14/PDIP 300mil 14/SOIC 150mil 14/SOIC150 14/TSSOP
PIC 16F684	Flash	3584	2048	256	128	12	8/10- Bit	2	No	2-8bit 1-16bit 1-WDT	No		20			14/PDIP 14/PDIP 300mil 14/SOIC 150mil 14/SOIC150 14/TSSOP 16/QFN 20/PDIP 80/TQFP
PIC 16F685	Flash	7168	4096	256	256	18	12/10- Bit	2	No	2-8bit 1-16bit 1-WDT	No		20	5.5	2	20/PDIP 20/SOIC 300mil 20/SSOP 208mil
PIC 16F687		3584	2048	256	128	18	12/10- Bit	2	No	1-8bit 1-16bit 1-WDT	No	EUSART I ² C™ Compatible /SPI™	20	5.5	2	20/PDIP 20/SOIC 300mil 20/SSOP 208mil
PIC 16F688	Flash	7168	4096	256	256	12	8/10- Bit	2	No	1-8bit 1-16bit 1-WDT	No	EUSART	20			14/PDIP 14/PDIP 300mil 14/SOIC 150mil 14/SOIC150 14/TSSOP
PIC 16F689	Flash	7168	4096	256	256	18	12/10- Bit	2	No	1-8bit 1-16bit 1-WDT	No	EUSART I ² C™ Compatible /SPI™	20	5.5	2	20/PDIP 20/SOIC 300mil 20/SSOP 208mil
PIC 16F690	Flash	7168	4096	256	256	18	12/10- Bit	2	No	2-8bit 1-16bit 1-WDT	No	EUSART I ² C™ Compatible /SPI™	20	5.5	2	20/PDIP 20/SOIC 300mil 20/SSOP 208mil
PIC 16F716	Flash	3584	2048	0	128	13	4/8- Bit	0	No	2-8bit 1-16bit 1-WDT 2-8bit 1-16bit 1-WDT	No No No No		20			18/PDIP 18/SOIC 300mil 18/SOIC300 20/SSOP 208mil 28/PDIP 300mil 28/QFN
PIC 16F72	Flash	3584	2048	0	128	22	5/8- Bit	0	No	2-8bit 1-16bit 1-WDT	No	I ² C™ Compatible /SPI™	20			28/PDIP 300mil 28/QFN 28/SOIC 300mil 28/SOIC300 28/SPDIP 28/SSOP 208mil 28/SSOP208

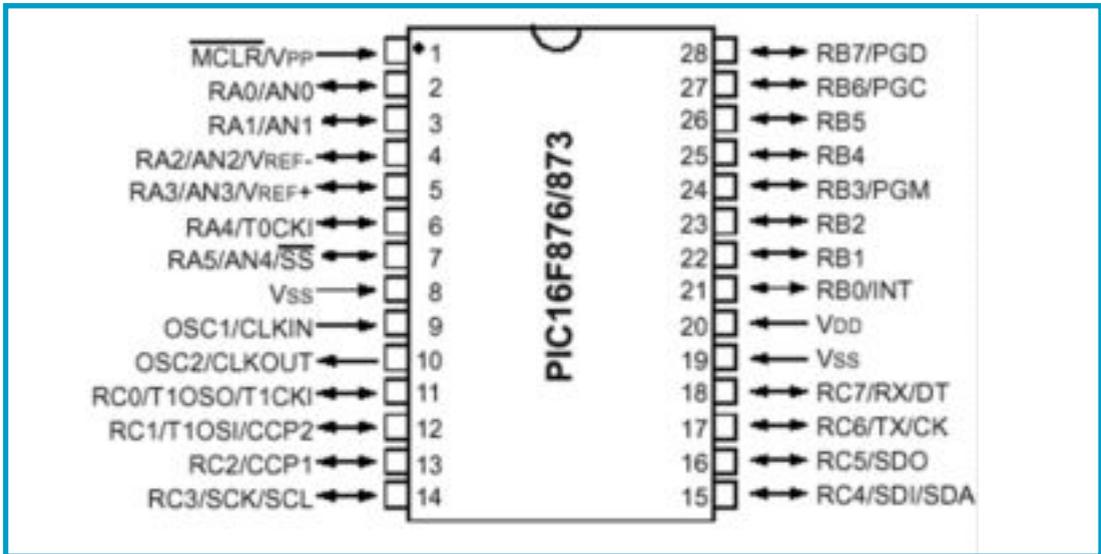
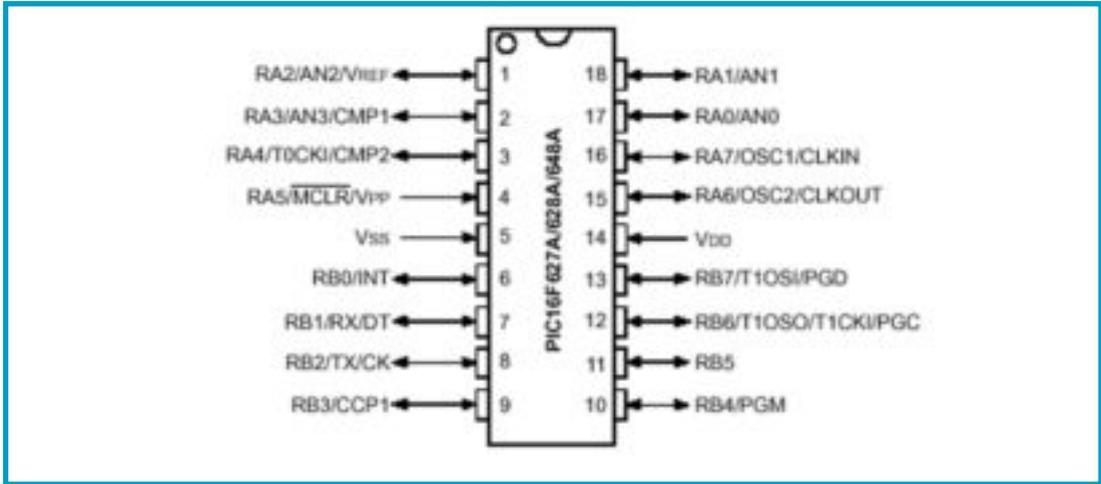
	Memoria	Bytes	Words	EEPROM Data memory	RAM	I/O Pins	ADC	Compa-rador	AmpOp	Timers/WDT	USB	Interface	Máx vel MHz	Vdd mín	Vdd máx	Encapsulado
PIC 16F73	Flash	7168	4096	0	192	22	5/8-Bit	0	No	2-8bit 1-16bit 1-VWDT	No	USART I ² C™ Compatible /SPI™	20			28/PDIP 300mil 28/PDIP300 28/QFN 28/SOIC 300mil 28/SOIC300 28/SPDIP 28/SSOP 208mil 28/SSOP208
PIC 16F737	Flash	7168	4096	0	368	25	11/10-Bit	2	No	2-8bit 1-16bit 1-VWDT	No	AUSART MI ² C Compatible /SPI™	20			28/PDIP 300mil 28/QFN 28/SOIC 300mil 28/SPDIP 28/SSOP 208mil
PIC 16F74	Flash	7168	4096	0	192	33	8/8-Bit	0	No	2-8bit 1-16bit 1-VWDT	No	USART I ² C™ Compatible /SPI™	20			40/PDIP 40/PDIP 600mil 44/PLCC 44/QFN 44/TQFP
PIC 16F747	Flash	7168	4096	0	368	36	14/10-Bit	2	No	2-8bit 1-16bit 1-VWDT	No	AUSART MI ² C Compatible /SPI™	20			40/PDIP 40/PDIP 600mil 44/QFN 44/TQFP
PIC 16F76	Flash	14336	8192	0	368	22	5/8-Bit	0	No	2-8bit 1-16bit 1-VWDT	No	USART I ² C™ Compatible /SPI™	20			28/PDIP 300mil 28/PDIP300 28/QFN 28/SOIC 300mil 28/SOIC300 28/SPDIP 28/SSOP 208mil 28/SSOP208
PIC 16F767	Flash	14336	8192	0	368	25	11/10-Bit	2	No	2-8bit 1-16bit 1-VWDT	No	AUSART MI ² C Compatible /SPI™	20			28/PDIP 300mil 28/QFN 28/SOIC 300mil 28/SPDIP 28/SSOP 208mil
PIC 16F77	Flash	14336	8192	0	368	33	8/8-Bit	0	No	2-8bit 1-16bit 1-VWDT	No	USART I ² C™ Compatible /SPI™	20			40/PDIP 40/PDIP 600mil 40/PDIP600 44/PLCC 44/QFN 44/TQFP
PIC 16F777	Flash	14336	8192	0	368	36	14/10-Bit	2	No	2-8bit 1-16bit 1-VWDT	No	AUSART MI ² C Compatible /SPI™	20			40/PDIP 40/PDIP 600mil 44/QFN 44/TQFP
PIC 16F785	Flash	3584	2048	256	128	18	12/10-Bit	2	Yes	2-8bit 1-16bit 1-VWDT	No		20			20/PDIP 20/SOIC 300mil 20/SSOP 208mil

	Memoria	Bytes	Words	EEPROM Data memory	RAM	I/O Pins	ADC	Compa- rador	AmpOp	Timers/ WDT	USB	Interface	Máx vel MHz	Vdd mín	Vdd máx	Encapsulado
PIC 16F818	Flash modifi- cada	1792	1024	128	128	16	5/10- Bit	0	No	2-8bit 1-16bit 1-WDT	No	I ² C™ Compatible /SPI™	20	2	5.5	18/PDIP 18/SOIC 300mil 20/SSOP 208mil 28/QFN
PIC 16F819	Flash modifi- cada	3584	2048	256	256	16	5/10- Bit	0	No	2-8bit 1-16bit 1-WDT	No	I ² C™ Compatible /SPI™	20	2	5.5	18/PDIP 18/PDIP300 18/SOIC 300mil 18/SOIC300 20/SSOP 208mil 28/QFN
PIC 16F84A	Flash	1792	1024	64	68	13	0	0	No	1-8bit 0-16bit 1-WDT	No		20			18/PDIP 18/PDIP300 18/SOIC 300mil 20/SSOP 208mil
PIC 16F87	Flash modifi- cada	7168	4096	256	368	16	0	2	No	2-8bit 1-16bit 1-WDT	No	AUSART I ² C™ Compatible /SPI™	20	2	5.5	18/PDIP 18/SOIC 300mil 20/SSOP 208mil 28/QFN
PIC 16F870	Flash modifi- cada	3584	2048	64	128	22	5/10- Bit	0	No	2-8bit 1-16bit 1-WDT	No	AUSART	20	2	5.5	28/PDIP 300mil 28/SOIC 300mil 28/SPDIP 28/SSOP 208mil
PIC 16F871	Flash modifi- cada	3584	2048	64	128	33	8/10- Bit	0	No	2-8bit 1-16bit 1-WDT	No	AUSART	20	2	5.5	40/PDIP 44/PLCC 44/TQFP
PIC 16F872	Flash modifi- cada	3584	2048	64	128	22	5/10- Bit	0	No	2-8bit 1-16bit 1-WDT	No	MI ² C Compatible /SPI™	20	2.2	5.5	28/PDIP300 28/SOIC 300mil 28/SOIC300 28/SPDIP 28/SSOP 208mil 28/SSOP208
PIC 16F873A	Flash modifi- cada	7168	4096	128	192	22	5/10- Bit	2	No	2-8bit 1-16bit 1-WDT	No	AUSART MI ² C Compatible /SPI™	20	2	5.5	28/PDIP 300mil 28/QFN 28/SOIC 300mil 28/SOIC300 28/SPDIP 28/SSOP 208mil 28/SSOP208
PIC 16F874 A	Flash modifi- cada	7168	4096	128	192	33	8/10- Bit	2	No	2-8bit 1-16bit 1-WDT	No	AUSART MI ² C Compatible /SPI™	20	2	5.5	40/PDIP 40/PDIP 600mil 44/PLCC 44/QFN 44/TQFP

	Memoria	Bytes	Words	EEPROM Data memory	RAM	I/O Pins	ADC	Compa-rador	AmpOp	Timers/WDT	USB	Interface	Máx vel MHz	Vdd mín	Vdd máx	Encapsulado
PIC 16F876A	Flash modificada	14336	8192	256	368	22	5/10-Bit	2	No	2-8bit 1-16bit 1-WDT	No	AUSART M ² C Compatible /SPI TM	20	2	5.5	28/PDIP 300mil 28/PDIP300 28/QFN 28/SOIC 300mil 28/SOIC300 28/SPDIP 28/SSOP 208mil 28/SSOP208
PIC 16F877A	Flash modificada	14336	8192	256	368	33	8/10-Bit	2	No	2-8bit 1-16bit 1-WDT	No	AUSART M ² C Compatible /SPI TM	20	2	5.5	40/PDIP 40/PDIP 600mil 40/PDIP600 44/PLCC 44/QFN 44/TQFP
PIC 16F88	Flash modificada	7168	4096	256	368	16	7/10-Bit	2	No	2-8bit 1-16bit 1-WDT	No	AUSART I ² C TM No Compatible No No No /SPI TM	20	2	5.5	18/PDIP 18/SOIC 300mil 18/SOIC300 20/SSOP 208mil 28/QFN
PIC 16F913	Flash	7168	4096	256	256	25	5/10-Bit	2	No	2-8bit 1-16bit 0-WDT		AUSART I ² C TM Compatible /SPI TM	20	0	0	28/QFN 28/SOIC 300mil 28/SPDIP 28/SSOP 208mil
PIC 16F914	Flash	7168	4096	256	256	36	8/10-Bit	2	No	2-8bit 1-16bit 1-WDT		AUSART I ² C TM Compatible /SPI TM	20 20 20			40/PDIP 44/TQFP
PIC 16F916	Flash	14336	8192	256	352	25	5/10-Bit	2	No	2-8bit 1-16bit 0-WDT		AUSART I ² C TM Compatible /SPI TM				28/QFN 28/SOIC 300mil 28/SPDIP 28/SSOP 208mil
PIC 16F917	Flash	14336	8192	256	352	36	8/10-Bit	2	No	2-8bit 1-16bit 0-WDT		AUSART I ² C TM Compatible /SPI TM				40/PDIP 44/TQFP

A manera de ilustración, presentamos algunos encapsulados de los modelos más utilizados en los distintos desarrollos:





A. REPERTORIO DE INSTRUCCIONES

MICROCONTROLADORES PIC

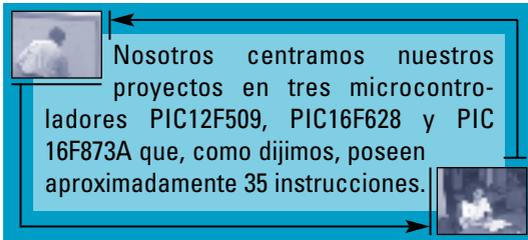
- a. Repertorio de instrucciones
- b. Manejo de interrupciones
- c. Registros
- d. Tablas y subrutinas

Decíamos que el PIC posee arquitectura RISC; o sea que su juego de instrucciones es reducido; aproximadamente, 35 instrucciones. Estas instrucciones son, además, sencillas y rápidas, puesto que casi todas se ejecutan en un único ciclo de máquina (equivalente a 4 del reloj principal). Sus operandos

son de gran flexibilidad; puede actuar cualquier objeto como fuente y como destino.

El PIC posee tres tipos bien diferenciados de direccionamiento; éstos son:

- **Inmediato.** El valor del dato está incluido en el propio código OP, junto a la instrucción.
- **Directo.** La dirección del dato está incluida en el propio código OP, junto a la instrucción.
- **Indirecto.** La dirección de la memoria de datos que guarda el operando está contenida en un registro.



B. MANEJO DE INTERRUPCIONES

La **interrupción** es una técnica que coloca al programa temporalmente, en suspenso, mientras el microcontrolador ejecuta otro conjunto de instrucciones en respuesta a un suceso. Las causas de una interrupción pueden ser externas o internas:

- Las **internas** podrían ser el desbordamiento, por ejemplo, del temporizador TMR0, el fin de la escritura de la EEPROM, la finalización de la conversión A/D.

- Las **externas** principales son la activación del pin 0 del puerto B (RBO/INT), el cambio de estado en la parte alta del puerto B, pines 4-7.

Al activarse una interrupción, el microprocesador ejecuta una **rutina de servicio de interrupción (RSI)** y, al terminar, el programa principal continúa donde fue interrumpido, dirección que guarda el mismo microcontrolador en el lugar de la memoria llamado *pila*. La dirección en la que se debe situar la rutina de interrupción es la 04. Para terminarla correctamente, se debe usar la instrucción *retfie*, dado que al activarse una interrupción, el mecanismo de éstas se deshabilita, colocando un cero en el bit *gie* del registro *intcon* como medida de seguridad y, entonces, el *retfie* sirve para rehabilitarlas, volviendo dicho bit automáticamente a uno.

Como las rutinas pueden modificar el contenido de los registros del microprocesador, al iniciarlas conviene guardar en la pila su valor y restaurarlo antes de finalizarlas. Antes de finalizar la rutina de interrupción se debe, por programa, volver a cero el bit del *flag* correspondiente a la interrupción producida en el registro *intcon*.

El contenido del registro W debe guardarse primero, junto con todos los registros que deban ser usados en la rutina de interrupción. El hecho de mover W a otro registro modifica la bandera Z, cambiando el registro de estado.

Recomendamos una secuencia de código que permite salvar y restaurar los registros sin modificarlos. La mostramos en la siguiente secuencia de código:

```

, ***** SALVAR *****
MOVWF cont1_W      ; Guardamos contenido de W en su sitio
SWAPF estado,w    ; Swap del contenido de estado en W
MOVWF cont2_S     ; Guarda el contenido de estado en la memoria
...
...
...

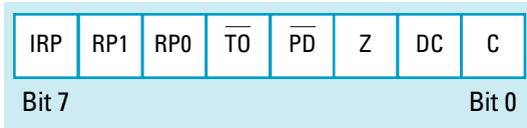
, ***** FIN RUTINA RSI *****
SWAPF cont2_S,w   ; Deja estado como estaba
MOVWF estado     ; Y lo restaura
SWAPF cont1_W,f
SWAPF cont1_W,w
RETFIE

```

La instrucción *swapf* mueve los datos sin afectar a la bandera Z del registro de estado. Aunque los conjuntos de 4 bits se invierten en el proceso, posteriormente son restaurados en su situación inicial. Si se emplea la instrucción *movf*, se modifica el bit Z.

C. REGISTROS

Registro status. Tiene por misión mantener el estado del microcontrolador; entendemos por estado el signo de las cuentas que el micro va haciendo.

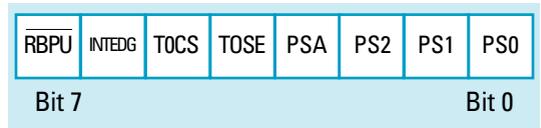


Algunos de los bits de este registro son:

- Bit **Z**. Se ocupa de indicar que la cuenta dio cero.
- Bit **C**. Se ocupa de indicar si hubo acarreo en una suma o resta.

- Bit **RP0**. Se ocupa de realizar el cambio de banco de memoria.
- El bit **DC**. Se ocupa de indicar si hubo acarreo del bit 3 al 4.

Registro option. Tiene por misión fundamental, manejar el registro *TMRO*, que es el que se ocupa de medir tiempos.



Algunos de los bits de este registro son:

- Bit **PS0,1,2**. Se denomina preescaler y tiene por misión dividir la frecuencia que ingresa al *TMRO* en una relación preestablecida (1:2 hasta 1: 256).
- Bit **PSA**. Se ocupa de asignar el preescaler al *TMRO* o al *WDT*.
- Bit **INTEDG**. Decide el flanco activo en la interrupción por *RB0/INT*.

Registros de interrupción y banderas, *intcon*.

Cada causa de interrupción actúa con dos señales. Una de ellas como señalizador, flag o bandera que indica, con un "1", si se ha producido o no la interrupción, mientras que la otra funciona como habilitación o prohibición de la interrupción en sí.

El PIC 16F84, por ejemplo, dispone de 4 fuentes de interrupción que se habilitan poniendo a "1" los correspondientes bits del registro *intcon* que ocupa la dirección 0B H del banco 0, encontrándose duplicado en el banco 1.

Pasaremos a describir el funcionamiento de cada uno de los bits del **Registro *intcon***

GIE	EEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF
Bit 7				Bit 0			

- El bit **GIE** -*Global Interrupt Enable*-1: Permite la ejecución de todas las interrupciones que también posea su bit de permiso individual habilitado. 0: Prohíbe todas las interrupciones. Las interrupciones son capaces de despertar al chip de su estado de reposo. El bit GIE se borra en cuanto se está atendiendo una interrupción, para evitar que se atienda otra. Vuelve a ponerse a 1 si se vuelve de la interrupción mediante un *retfie*.
- El bit **RBIE** habilita la interrupción producida en la parte alta del Puerto B; es decir, la interrupción ante cambios en los pines RB4-RB7. **RBIF** es la bandera o *flag* que indica que se ha producido esta interrupción.

- El bit **INTE** activa la interrupción por el pin RBO/INT. El bit **INTF** es la bandera o *flag* que indica si se ha producido esta interrupción.
- El bit **TOIE** habilita la interrupción por desbordamiento del TMR0. El bit **TOIF** es la bandera o *flag* que indica si se ha producido la interrupción.
- El bit **EEIE** habilita la interrupción por finalización de la escritura en la EEPROM de datos. La bandera o *flag* que indica si se ha producido la interrupción se supone por descarte, en relación con las otras.

Todas las banderas o *flags* mencionadas deben ponerse a "0" por programa y sólo el microcontrolador las cambia a "1" cuando se atiende la interrupción como aviso de identificación para el programador; luego, al finalizar la interrupción, también por programa, deben ser puestas a "0" nuevamente .

En muchos casos es importante y muy útil conocer la existencia y el manejo del *watchdog* o perro guardián; esta herramienta es un contador de 8 bits que, al desbordarse, produce el reseteo del micro. La única forma de evitar este reseteo es, por tanto, borrarlo por software cada cierto tiempo con la instrucción **CLRWDT**, que devuelve su valor a 0. Su variación es la de una cuenta por cada ciclo de instrucción, aunque puede asignársele el preescaler (al igual que al TMR0) para variar su frecuencia desde el registro *option* del microcontrolador.

Su utilización es opcional, y se activa o desactiva durante el proceso de grabación del microcontrolador. Todos los grabadores que

conocemos y hemos usado tienen en sus menús o en sus funciones esta opción específica, que sirve para evitar posibles problemas de grabación no controlados o controlables -como, por ejemplo, bucles infinitos, esperas exageradamente largas de alguna determinada entrada, etc.- y que es especialmente útil en ambientes con mucho ruido, haciendo que el PIC ejecute líneas al azar.

D. TABLAS Y SUBRUTINAS

El uso de rutinas es esencial, puesto que simplifica los programas, haciéndolos, además, más modulares.

Llamamos a una subrutina con la orden **CALL**, seguida de la etiqueta que la encabeza o su dirección en memoria. Para regresar a la siguiente instrucción tras el **CALL** basta con situar, en la última línea de la subrutina, el comando **RETURN**; también es posible emplear **RETLW k**, que se diferencia del anterior por situar en el registro W el valor k.

En general, en los microcontroladores existe una manera específica de regresar en caso de interrupción: **RETFIE**. Las interrupciones generan un salto a la dirección 0x04 que es tratado como una subrutina para permitir la continuación normal del programa a partir del punto en que se llevó a cabo la interrupción. Este regreso se realiza mediante **RETFIE**, que, además de restaurar el contador de programa, habilita de nuevo las interrupciones (ya que éstas son deshabilitadas mientras se está atendiendo una).

La pila *-stack-*, en la mayoría de los microcontroladores permite guardar hasta 8 saltos; es decir, admite un máximo de 8 saltos a sub-

rutina anidados. Recuerde este dato para no excederse. Recuerde, también, siempre, un posible salto por interrupción, para no pasar de 7 si ésta está habilitada.

Una tabla, en cambio, es una lista ROM de constantes en la memoria de programa o memoria EEPROM de datos. Éstos son útiles para la conversión de códigos, como pasar de hexadecimal a información a ser mostrada en un display. Para generarlas se aprovecha la cualidad de **RETLW** para situar un dato en el registro W o se utilizan rutinas de grabación en EEPROM de datos dadas por el fabricante.

Dispositivos relacionados con el equipo⁷



Memorias

Las memorias son registros de desplazamiento de entrada en paralelo (la información está presente en todos los bits, simultáneamente) con salidas que, también, se hallan en paralelo (salidas accesibles en todos los biestables al mismo tiempo).

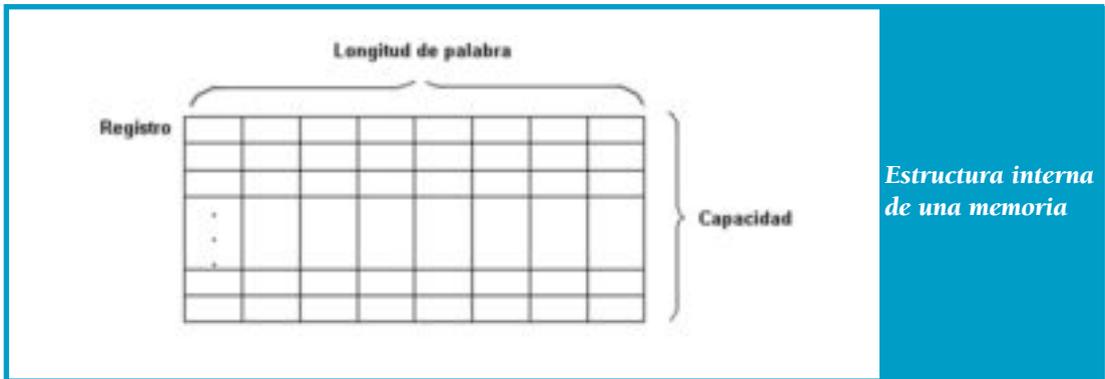
⁷ Le proponemos ampliar con:

- Gil Padilla, Antonio (1990) *Electrónica general. Dispositivos básicos y analógicos*. Mc Graw-Hill.
- Revista *Electrónica & computadores*. Cekit.
www.cekit.com.co/index.php

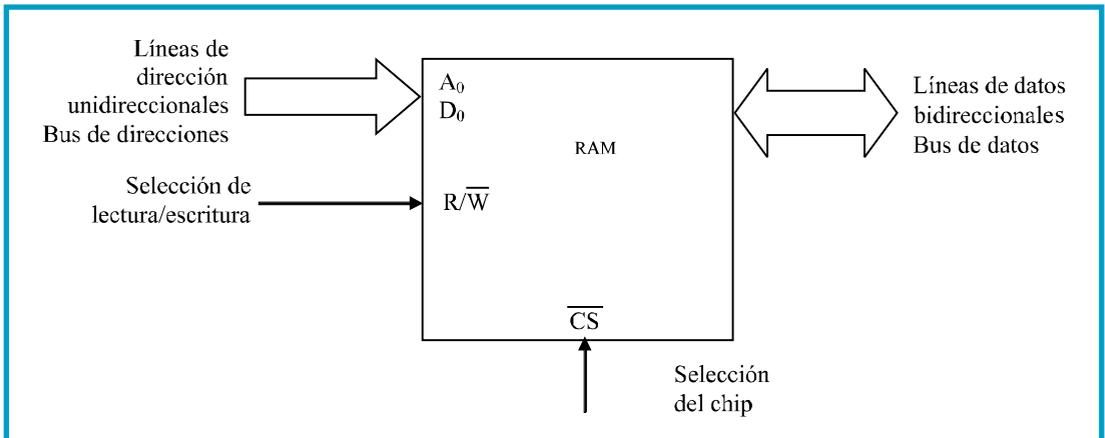
Una **memoria** es una unidad de almacenamiento de información binaria en la que el contenido permanece allí, guardado, hasta que una nueva entrada lo modifica; es decir, hasta cuando se almacena un nuevo dato.

Las memorias semiconductoras de tipo comercial vienen en forma de chip o encapsulado integrado, definidas por su longitud de palabra y su capacidad; contienen una matriz de memoria, un decodificador de direcciones, los transductores correspondientes y el tratamiento lógico de algunas señales de control.

- La **longitud de palabra** es la cantidad de bits que se pueden manejar en paralelo (en forma simultánea). Las memorias suelen tener 2, 4, 8 etc. bits de longitud de palabra.
- La **capacidad** de una memoria está representada por la cantidad de lugares o registros de almacenamiento -la longitud de palabra determinada-, que posee dicha memoria, de acuerdo con su posibilidad de direccionamiento (por ejemplo, 1 k, 4 k, etc.). La posibilidad de direccionamiento está dada por la cantidad de líneas de dirección que entra al chip.



Estructura interna de una memoria



Estructura externa de una memoria ; por ejemplo, una memoria RAM

- El **bus de datos** es el conjunto de líneas de tres estados que transportan la información almacenada en memoria. El bus de datos se puede conectar a las líneas correspondientes de varios integrados.
- El **bus de direcciones** es un conjunto de "m" líneas que transportan la dirección, que permite codificar la posición de memoria a la que se desea acceder.

A modo de ejemplo, supongamos una memoria RAM (a definir más adelante) de 1 k x 8. Esto implica que tiene una capacidad de 1 k y una longitud de palabra de 8 bits (que es equivalente a 1 byte), de lo que se desprende que el bus de datos de la memoria es $D_0 - D_7$.

1 k = 1024 registros o lugares.

$1024 = 2^{10} \rightarrow$ la memoria necesita 10 líneas de dirección para seleccionar alguno de los 1024 lugares a leer o escribir.

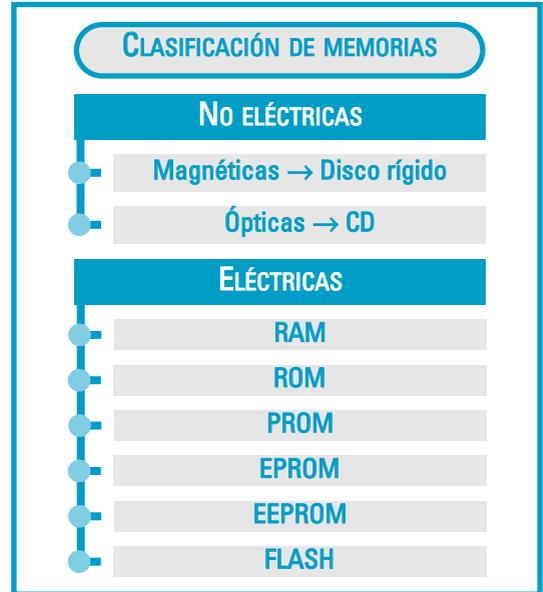
En consecuencia, el bus de direcciones es $A_0 - A_9$.

La memoria es un bloque fundamental de cualquier sistema de computadoras o micro-controladores; su función consiste en almacenar los datos y las instrucciones. La **memoria principal** es el órgano que almacena los datos e instrucciones de los programas en ejecución; pero, a veces, la memoria principal no tiene la suficiente capacidad como para contener todos los datos e instrucciones; en este caso se precisan otras memorias auxiliares o secundarias, que funcionan como periféricos del sistema y cuya información se transfiere a la memoria principal cuando se necesita.

La memoria sólo puede realizar dos opera-

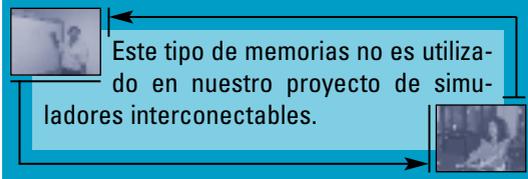
ciones básicas:

- **Lectura.** El dispositivo de memoria recibe una dirección de la posición de la que se quiere extraer la información depositada previamente.
- **Escritura.** Además de la dirección, se suministra la información que se desea grabar.



Las memorias no eléctricas:

- Permiten un almacenamiento masivo.
- Son memorias periféricas (normalmente, son externas a lo que es la CPU).
- Requieren de un controlador.
- Son de velocidades de acceso variables.
- Son no volátiles (la información permanece almacenada aún cuando se corta la alimentación).
- Pueden ser de acceso aleatorio o secuencial.



Memorias RAM. Son las memorias de acceso aleatorio *-Random Access Memory-*. En ellas se puede leer o escribir un dato aleatoriamente, según se desee. Son memorias volátiles; su contenido desaparece al cortarse la alimentación del circuito integrado.

Encontramos dos tipos de memoria RAM:

- **DRAM *-Dynamic Random Access Memory-***. Es la memoria de acceso aleatorio dinámica. Está organizada en direcciones de memoria *-addresses-* que son reactivadas varias veces por segundo (refresco). A mayor cantidad de este tipo de memoria, más datos se pueden tener en ella y más aplicaciones pueden estar funcionando simultáneamente a mayor velocidad de proceso.
- **SRAM *-Static Random Access Memory-***. Es la memoria estática de acceso aleatorio, alternativa a la DRAM. No necesita tanto consumo de energía para su refresco y reemplazo de las direcciones, funcionando más rápido. Son de mayor costo que la DRAM.

Ambos tipos son volátiles, lo que significa que pueden perder su contenido cuando se desconecta la alimentación.

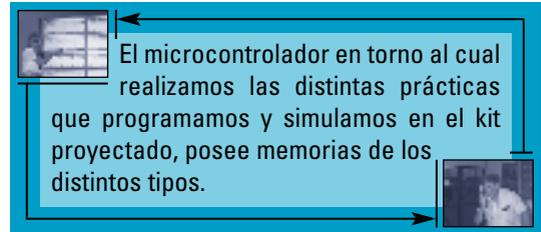
Memorias ROM. Son memorias de sólo lectura *-Read Only Memory-*. En el proceso de fabricación son cargadas con cierta información que es permanente; permiten la lectura

de cada posición las veces que sea necesario. Son no volátiles y de acceso aleatorio.

Memorias PROM. Constituyen un tipo de memoria ROM. Los PROM *-Programmable Read Only Memory;* memoria programable de sólo lectura- son dispositivos de almacenamiento de lectura solamente, que se pueden reprogramar después de su manufactura, por medio de equipo externo. Los PROM son, generalmente, circuitos integrados.

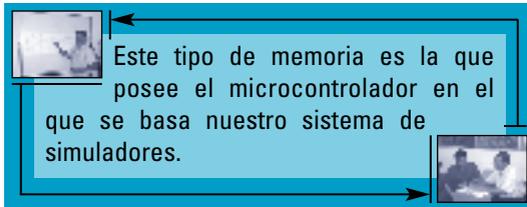
Memorias EPROM. Las memorias EPROM *-la E corresponde a erasable, borrrable-* son ROM que se pueden borrar totalmente con luz ultravioleta y, luego, en condiciones limitadas, reprogramarse. Las EPROM resultan mucho más económicas que las PROM porque pueden reutilizarse.

Memorias EEPROM. Aún mejores que las EPROM son las EEPROM *-eléctricamente borrrables-*, también llamadas EAROM *-ROM eléctrica alterables-*, que pueden borrarse mediante impulsos eléctricos, sin necesidad de ser introducidas en un lugar especial para ser expuestas a luz ultravioleta.



Memorias FLASH. Son del tipo no volátil, similares en algunos aspectos a las de tecnología EEPROM, pero con ventajas respecto a estas últimas, que las han hecho memorias muy utilizadas en los modernos microcontroladores. Su tecnología es del tipo borrrable y programable eléctricamente; pero, a dife-

rencia de la EEPROM presenta una reducción a mínima expresión de los circuitos auxiliares de acceso orientados al byte. Es decir, no es posible acceder al borrado de un solo byte en forma arbitraria, ya que no se cuenta con los mecanismos de acceso (circuitos de direccionamiento) que permitan el borrado de un byte independientemente del resto. Este rasgo redundante en una reducción de los costos de esta memoria y de su implementación.



Conversores

Las variables involucradas en los distintos sistemas de tipo digital no son necesariamente discretas, variando continuamente en el dominio *tiempo*. Es por ello que, para ser procesadas, deben ser convertidas a digital. Luego, en muchos casos, deben volver a su forma analógica.

Convertor A/D (ADC). El convertor analógico-digital es un dispositivo que convierte una señal que varía continuamente en el tiempo en otra que lo hace a saltos -es decir, que tiene valores discretos-.

El proceso de conversión analógico a digital involucra cuatro pasos fundamentales: muestreo, mantenimiento, cuantificación y codificación.

Uno de los parámetros más importantes de

esta conversión es el error de cuantificación. Éste constituye la diferencia que existe entre la salida digital y la entrada analógica, dependiendo de la cantidad de bits utilizados en la conversión.



Diagrama básico de convertor A/D

Convertor D/A (DAC). Un convertor digital-analógico es un dispositivo que genera una señal proporcional a la palabra digital presente en sus entradas. El convertor D/A más simple que se puede describir consta de una tensión de referencia y de un conjunto de resistencias que entran en juego en el circuito, de acuerdo a si se conecta o no el correspondiente interruptor:

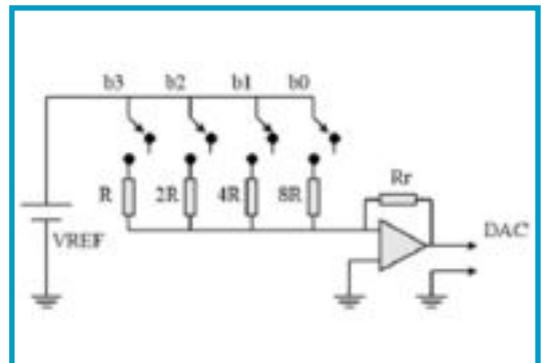


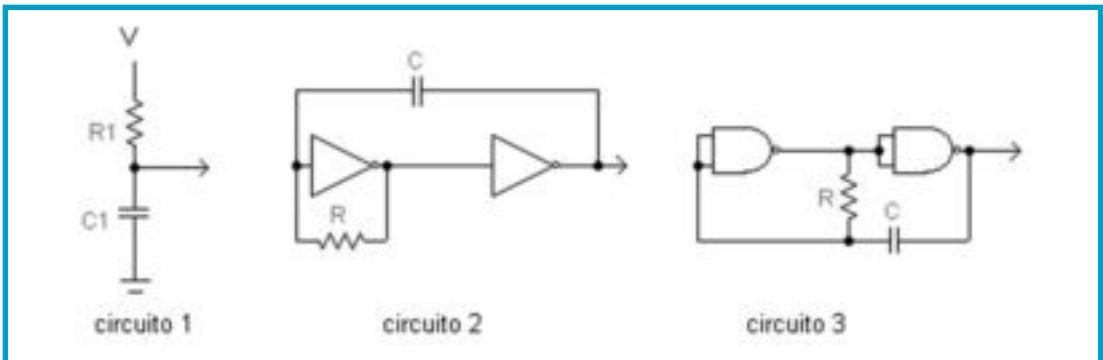
Diagrama de convertor D/A

Osciladores. Temporizadores. Generadores de pulsos de clock o reloj

La mayoría de los circuitos que hemos analizado -biestables, contadores o registros de desplazamiento- necesita de una señal de clock o reloj para su funcionamiento como circuitos secuenciales; lo mismo ocurre con los circuitos con microcontroladores utilizados en el recurso didáctico que le proponemos desarrollar. Los osciladores o generadores de pulso o temporizadores son los circuitos que cumplen esta función.

Un **oscilador** es un circuito que tiene como característica la de cambiar el estado lógico de su salida continuamente, entre dos valores 1 y 0. La tensión de cada uno de estos niveles depende de la tecnología que se utilice.

Trabajando con componentes conocidos como resistencias, capacitores, inversores y compuertas, se pueden construir diversos generadores de clock o reloj.



Algunos ejemplos de circuitos generadores de pulsos de clock o reloj

El **circuito 1** es un económico oscilador RC, en el que el tiempo de carga es, aproximadamente, $T = R \cdot C$ [s], siendo $[R] = \Omega$ y $[C] = F$

En el **circuito 2**, si la salida está en nivel alto y C está descargado, en el instante inicial, la entrada del primer inversor es igual a la salida del segundo. C se carga a través de R, cerrándose el circuito por la salida del primer inversor que se encuentra en nivel bajo. Cuando la entrada del primer inversor llega a nivel bajo conmuta y su salida pasa a nivel alto, siendo la salida del oscilador un nivel bajo. C se descarga a través de R, cerrándose el circuito por la salida del segundo inversor hasta que la entrada del primer inversor alcanza un nivel alto y el oscilador pasa también a nivel alto; y, así, sucesivamente, generándose los pulsos deseados.

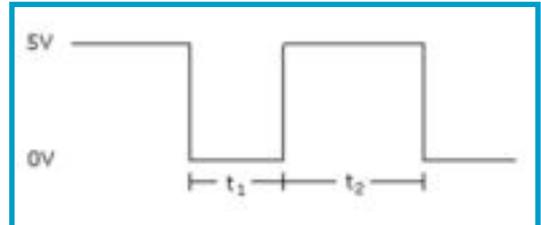
El **circuito 3** está realizado con compuertas *nand* con el mismo criterio de generación de pulsos que el anterior. La resistencia y el capacitor colocados allí determinan la frecuencia de oscilación de estos circuitos.

En estos circuitos es importante considerar tiempos de conmutación cortos; por esto, los circuitos de tecnología de fabricación CMOS resultan útiles.

Existen dos tipos de osciladores muy utilizados actualmente:

- El multivibrador astable, usando el circuito integrado CI-555.
- El sencillo oscilador con disparador de Schmitt.

Como queda establecido por el funcionamiento en modo astable, la señal de salida es una onda cuadrada con valores extremos 1 (5 V) y 0 (0 V):



Señal de salida del astable



Estos tipos de osciladores pueden ser utilizados para las distintas temporizaciones necesarias en el sistema de simuladores que proponemos como recurso didáctico. Estableceremos aquí sólo los aspectos básicos de su montaje y construcción, sin profundizar en su funcionamiento ni en el análisis de su operación.



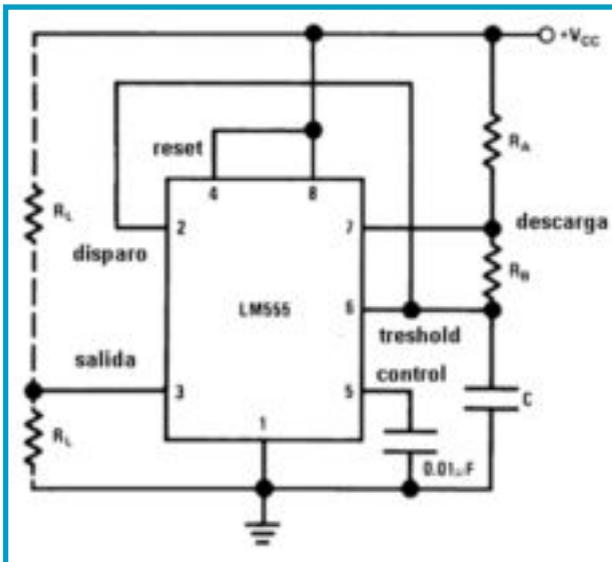
Donde:

- $t_1 = 0,693 \cdot R_B \cdot C$
- $t_2 = 0,693 \cdot (R_A + R_B) \cdot C$

Deducimos que el período será $T = t_1 + t_2$ y, lógicamente, la frecuencia es la inversa de éste.

Existen consideraciones de diseño, fundamentalmente en los valores de los componentes usados, para que el circuito oscile adecuadamente y no se deteriore el integrado. Éstas son:

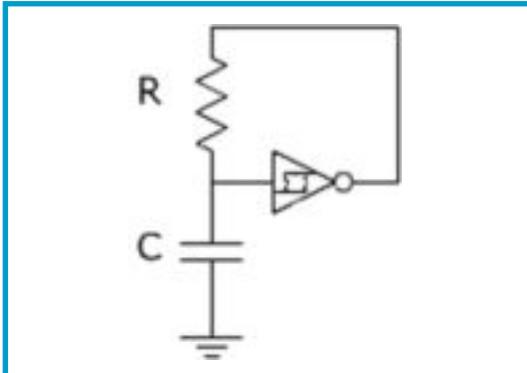
- $R_A \leq 1 \text{ K}\Omega$
- $R_A + R_B \leq 6,6 \text{ M}\Omega$
- $C \geq 500 \text{ pF}$



Oscilador astable con CI-555

Podemos observar cómo conectar los distintos componentes a los pines del circuito integrado para que trabaje como astable. Los valores de las resistencias R_A , R_B y C determinan la frecuencia de oscilación de la señal de salida.

En la próxima figura podemos observar el **oscilador con disparador de Schmitt**, un sencillo diseño con sólo dos componentes adicionales a la compuerta lógica, la resistencia R y el capacitor C . De ellos depende la forma de onda aproximadamente cuadrada resultante a la salida.



Oscilador con disparador de Schmitt

Con el **circuito integrado CI-555** también podemos diseñar un **multivibrador monoestable** que, por su montaje, produce el retardo o temporización necesarios en muchas implementaciones de electrónica digital actuales.

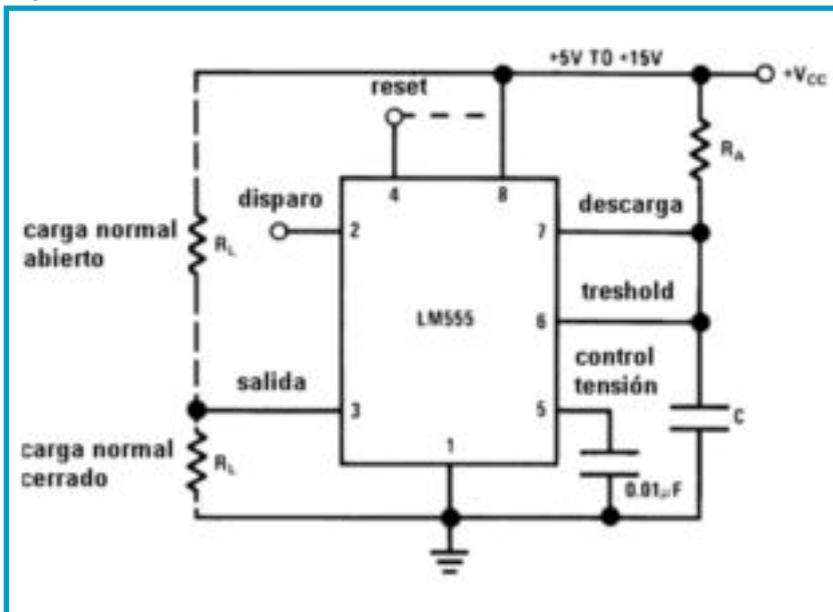
Disparo; éste debe permanecer en nivel alto (1 lógico) hasta el momento de empezar la temporización. Cuando este pin 2 va a "0", comienza el retardo calculado.

Debemos aclarar que, para evitar disparos aleatorios que variarían el tiempo de retardo, este pin 2 debe volver a su nivel alto antes de terminar la temporización.

Mientras el pin 2 de disparo está a nivel "1" lógico, la salida por el pin 3 permanece a nivel bajo (0 lógico), lo que debe tenerse en cuenta para un mejor aprovechamiento del dispositivo.

El **retardo** de tiempo se calcula:

$$T = 1,1 \cdot R_a \cdot C$$



Como ejemplo podríamos considerar la posibilidad de usar este tipo de temporizador para permitir encender y apagar una luz (lámpara o diodo led), o para arrancar y parar un motor colocado en su salida con un cierto retardo previsto y calculado. Recuerde que la salida puede entregar una corriente de, aproximadamente,

En la figura podemos apreciar el montaje de un monoestable. La principal característica es la conexión del pin 2 del integrado, llamado

200 mA como máximo; por esto, en muchas ocasiones es necesario colocar un relé para dar más potencia a la salida.



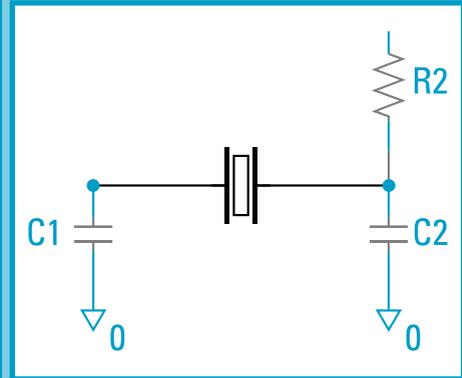
Todos los circuitos contadores presentados pueden ser fácilmente realizables con una programación relativamente sencilla y verificar los estados de cuenta con led indicadores conectados a los puertos del equipo simulador propuesto.

De la misma manera, pueden ser realizados y probados circuitos de registro de desplazamiento, temporizaciones, retardos y generación de ondas cuadradas de oscilación.

Además, la realización de estos circuitos en lógica digital discreta ayuda, en gran manera, a la comprensión del funcionamiento de los contadores internos de un microcontrolador y, así, facilita su programación.



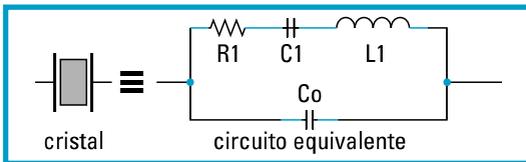
Un circuito básico, que utilizamos en nuestro equipo de simuladores interconectables es:



Este circuito utiliza dos capacitores que componen la capacidad de carga y, junto al cristal, establecen la frecuencia de oscilación del circuito.



En el desarrollo de algunos proyectos va a ser necesario contar con un oscilador de mucha exactitud y estabilidad en la frecuencia de oscilación. Puede integrarse, entonces, un **oscilador a cristal de cuarzo**. El cristal (XTAL) actúa como una bobina en serie con un pequeño condensador.



En este circuito, el capacitor C_o representa - en total- la capacidad entre los electrodos del cristal más la capacidad de la carcasa y sus terminales. R_1 , C_1 y L_1 ; éstos conforman la rama principal del cristal, donde L_1 representa la masa vibrante del cristal, C_1 la elasticidad del cuarzo y R_1 las pérdidas que ocurren dentro del cristal.

El cristal viene especificado por el fabricante, para trabajar con una determinada frecuencia.

Fuentes de alimentación

En muchas ocasiones es necesario construir una fuente de alimentación estabilizada para alimentar un circuito electrónico con especificaciones digitales. Por ello, es importante considerar algunas características.



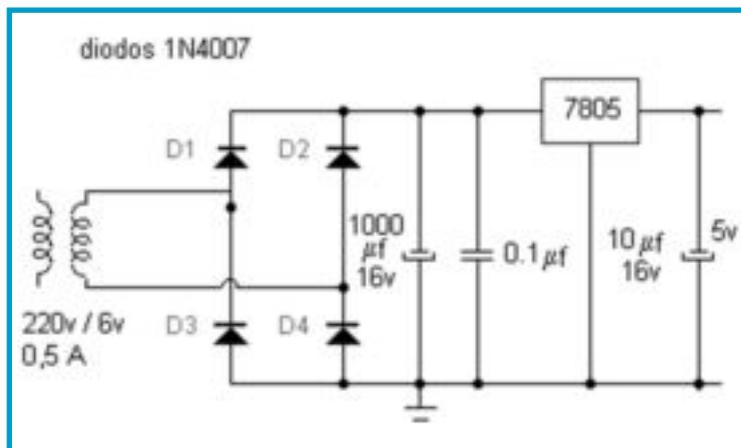
Estas fuentes son utilizadas para la alimentación de los programadores y del probador que componen el sistema de simuladores interconectables basados en lógica digital que proyectamos.



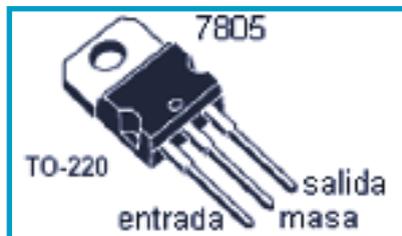
Para un diseño simple de fuentes de alimentación estabilizadas, podemos utilizar reguladores integrados monolíticos (reguladores fijos), con encapsulado TO-220 de 3 pines, otorgando por ejemplo 1A (ampere) -como máximo, con la debida disipación- de corriente de salida y una variada gama de tensiones de salida:

Tipo	Tensión/Salida
LM7805	5
LM7806	6
LM7808	8
LM7809	9

Si tomamos como ejemplo al integrado LM7805 de la tabla, podemos diseñar una fuente de alimentación de 5 V y 500 mA, cuyo esquema es el siguiente:

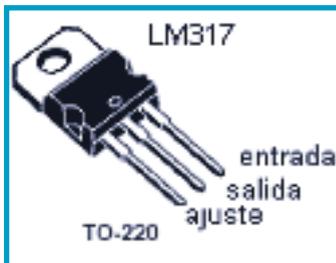


Fuente de alimentación fija, utilizando un regulador LM7805



Encapsulado de un regulador LM7805

En el mercado existen otros reguladores ajustables de tres o más pines, con diferentes encapsulados según la potencia a manejar. Podemos mencionar, entre ellos, los LM317 y LM337, con corrientes de trabajo menores a 1 A. Estos circuitos permiten que el pin común esté flotante y no conectado a masa, lo que posibilita una tensión regulable.



Encapsulado de un regulador ajustable LM317

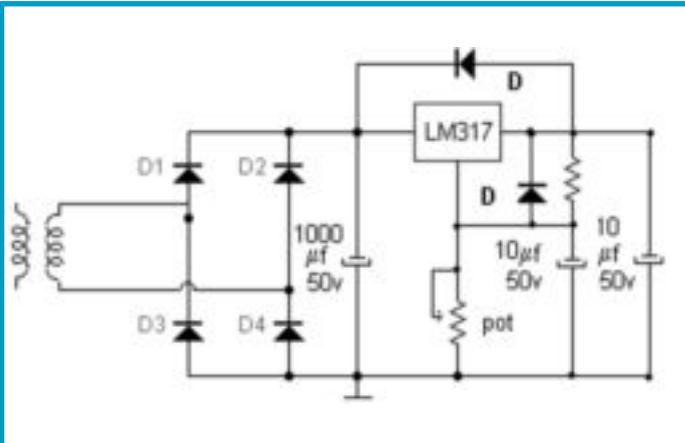
Los fabricantes de los reguladores recomiendan que la diferencia de potencial entre los pines de entrada y salida sea, como mínimo, 3 V, para que el regulador mantenga constante la tensión de salida.

Para eliminar tensiones alternas no deseadas y mejorar el *ripple* de la rectificación se colocan los capacitores de 10 µf. Las tensiones inversas y parásitas pueden evitarse con los diodos de seguridad "D". Se recomienda

Es muy importante incluir estos diodos "D".

colocar un disipador de aluminio adecuado para la disipación de temperatura, dada la corriente que maneja.

El potenciómetro "pot", permite ajustar la tensión de salida que se desee en cada momento.

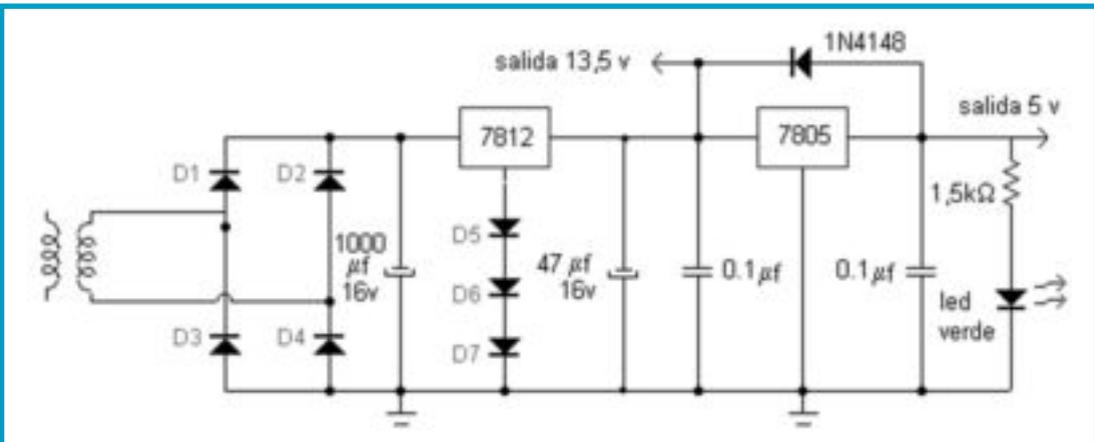


Fuente regulada utilizando LM317



En estos ejemplos, dispone usted de esquemas sobre fuentes de alimentación reguladas que pueden servirle como punto de partida para otros proyectos.

Parte del equipo que proponemos, utiliza una fuente de alimentación general incluida en la misma placa donde se conecta el microcontrolador a grabar y otra fuente, en placa separada.



Fuente de alimentación regulada, utilizada en el equipo de simuladores interconectables

Otros dispositivos

Otros dispositivos asociados con el **Equipo programador y simulador interconectable, basado en lógica digital** se relacionan con la representación y visualización de señales.

Display de led. Un led -Light Emitting Diode- es un diodo en el que la energía de separación entre bandas debe estar entre 1,8 eV y

3,1 eV, para poder generar luz visible y permitir la obtención de emisiones de luz en distintos colores, según las características constitutivas. Desde el punto de vista eléctrico, se comporta como un diodo común; pero, con diferentes niveles de tensión y de corriente de acuerdo con el color de la emisión.

Los displays basados en led pueden ser de 7 segmentos o más, como ya se trató oportunamente.

Display de cristal líquido -Liquid Crystal Display; LCD-. En esta tecnología, los display están físicamente compuestos por dos capas de vidrio que encierran entre sí un líquido compuesto por un enorme número de cristales en forma de hebras en suspensión.

Los LCD son ópticamente pasivos (no emiten luz); sólo bloquean o no su paso, por lo que se caracterizan por consumir cantidades mínimas de energía. Se basan en la propiedad del líquido de rotar la luz polarizada en forma relativa a un par de polarizadores cruzados laminados en ambas caras del *display*.

Hoy día hay dos tipos principales de displays simples de tipo LCD: los más económicos, de tipo TN -*Twisted Nematic*- y los de tipo STN -*Super Twisted Nematic*-, que tienen mejor contraste y ángulo de visión.

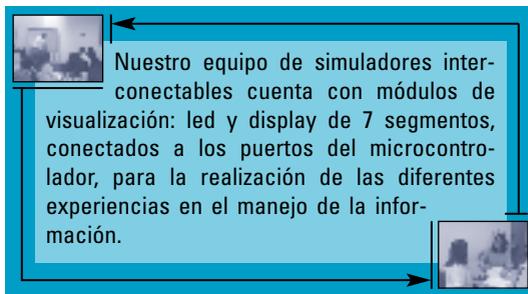
Para el caso de displays complejos (por ejemplo, del tipo usado en *notebooks*) se han desarrollado otras tecnologías.

En el caso de los displays de formato estándar pueden diferenciarse tres grandes tipos de display:

- **De segmentos**, para la generación de dígitos o símbolos especiales. Se presentan displays de 2, 3 ½, 4, 5, 6 y 8 dígitos, en los que las mayores diferencias están dadas por el tamaño de los dígitos y posibles indicaciones auxiliares. Estos displays no suelen tener controladores dedicados y su excitación debe ser resuelta por el diseñador.
- **De matriz de puntos organizados como displays de caracteres**. Los displays

alfanuméricos están organizados como una matriz de columnas y filas de caracteres; en ella, cada carácter está resuelto, a su vez, mediante una matriz de 8 puntos en alto por 5 puntos en ancho. Al igual que en el tipo anterior, la gran diferencia adicional es el alto de cada carácter, de importancia según la distancia a la que deberá poder ser leído el display. Estos displays incorporan un controlador que facilita enormemente su uso, programado mediante un bus de datos de 4 u 8 bits, y 3 líneas de control.

- **De matriz de puntos organizados como displays gráficos**. Los displays gráficos están organizados como una matriz de columnas y filas de puntos -*dot pixels*-, donde cada punto puede ser cuadrado o ligeramente rectangular. La cantidad y tamaño de los pixel define el grado de detalle de la imagen y la distancia a la que puede ser leído el display. Para la especificación de los displays gráficos se emplea un código compuesto de 9 a 11 letras y números, con el mismo formato al detallado para los displays de caracteres.

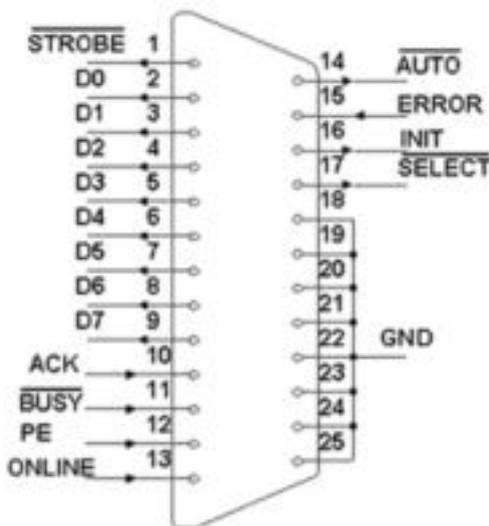
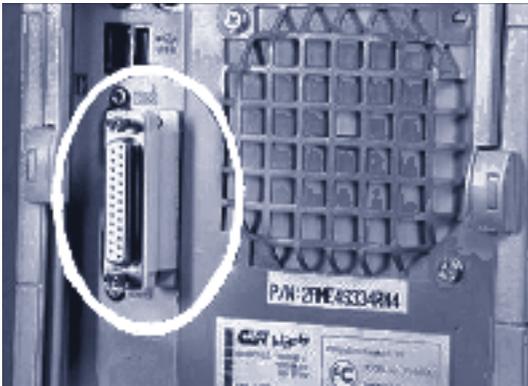


Puerto paralelo. Uno de los puertos de entrada/salida estándar de la PC es el puerto paralelo. Existen, además, otros dos puertos serie (RS 232).

El puerto paralelo permite, por ejemplo, conectar la impresora a la PC. Es utilizado por la CPU -unidad central de proceso- para enviar datos a la impresora usando líneas de entrada / salida como control para la comunicación.

La utilidad de estas señales es dejar que la CPU conozca el estado de la impresora y controle el dispositivo.

El hardware de este puerto consiste en 8 bits de salida de datos, 5 bits de entradas de control y 5 bits de salidas de control.



Salidas:

- STROBE. Indica a la impresora que los datos están disponibles para ser leídos.
- INIT. Resetea la impresora.
- SLCT IN. Selecciona la impresora al pasar a nivel bajo.
- AUTO FD. Indica a la impresora que imprima una línea en blanco seguida por una señal de "enter".
- D0-D7. Bits de datos.

Entradas:

- ACK. Indica a la CPU que los datos fueron recibidos correctamente.
- BUSY. Indica a la CPU que no debe enviar más datos.
- SLCT. Indica a la CPU que la impresora está conectada.
- ERROR. Indica a LA CPU que no debe enviar más datos pues se ha producido un error.
- PE: Indica que el papel se ha acabado.

Todas estas señales de nivel lógico TTL están conectadas a un conector hembra de 25 pines (DB25).

En el sistema operativo MS-DOS, puede manejar tres puertos paralelos LPT1, LPT2 y LPT3, cada uno asociado a una dirección 3BCh, 378h, 278h -lo usual es disponer sólo uno en la PC-; como ejemplo, el LPT1 utiliza las direcciones 378h (líneas de datos), 379h (líneas de status) y 37Ah (líneas de control) del mapa de entradas/salidas de la CPU.

Puerto serie. Las comunicaciones serie se utilizan para enviar datos a través de largas dis-

tancias. Los datos serie recibidos desde un módem u otros dispositivos se convierten a paralelo, para ser operados por la PC.

Los equipos de comunicaciones serie se pueden dividir entre *simplex*, *half-duplex* y *full-duplex*:

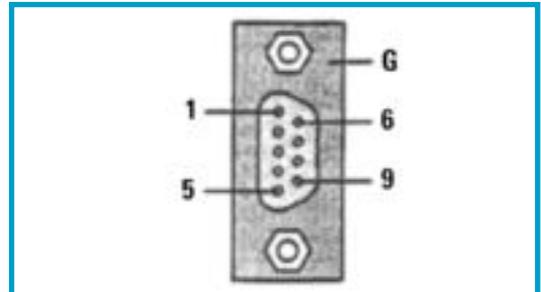
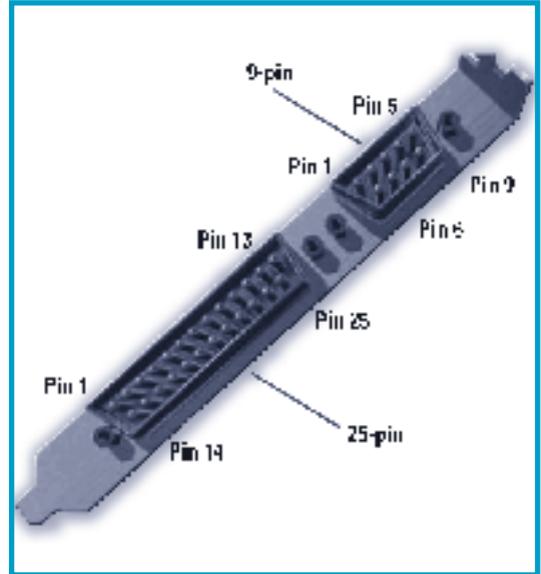
- Una comunicación serie *simplex* envía información en una sola dirección (por ejemplo, una emisora de radio comercial).
- *Half-duplex* significa que los datos pueden ser enviados en ambas direcciones entre dos sistemas; pero, en una sola dirección al mismo tiempo.
- En una transmisión *full-duplex* cada sistema puede enviar y recibir datos al mismo tiempo.

El puerto serie de la PC realiza una comunicación asincrónica, donde un bit identifica el comienzo de la palabra de datos *-start-* y otro bit identifica el final *-stop-*. La palabra digital está compuesta por 7 u 8 bits, con bit de paridad incluido para la verificación de errores, depende del alfabeto internacional que se utiliza y envía primero el bit menos significativo.

El puerto serie de la PC es compatible con el estándar de comunicación RS-232C que posee 25 pines de señal. El conector de la PC es macho y el conector hembra es el de cualquier dispositivo externo a comunicar (en nuestro caso, el equipo **programador y simulador interconectable**).

Muchos pines del conector DB 25 macho no son utilizados; por ello se usan los DB-9 macho. Lo usual es encontrar dos puertos

serie (COM1 y COM2): uno DB 25 macho y otro DB 9 macho en la parte trasera de la PC.



Pin	Función
1	PGND
2	TxD
3	RxD
4	RTS
5	CTS
6	PROG
7	SGND
8	0 V out
9	+5 V out

Los voltajes para un nivel lógico alto están entre -3 V y -15 V. Un nivel lógico bajo tiene un voltaje entre +3 V y +15 V. Los voltajes más usados son +12 V y -12 V.

Las señales más utilizadas se listan a continuación:

- /DTR -Data-Terminal-Ready-. La PC le indica al módem que esta lista para trabajar.
- /DSR -Data-Set-Ready-. El módem le indica a la PC que está listo para transmitir o recibir datos.
- /RTS -Request-To-Send-. La PC indica que existe un dato a transmitir.
- /CD -Carrier-Detect-. El módem reconoce que hay señal desde la CPU.
- /CTS -Clear-To-Send-. El módem está preparado para transmitir datos.
- TxD: El módem recibe datos desde la PC.
- RxD: El módem transmite datos a la PC.



En los **simuladores interconectables basados en lógica digital** hacemos uso de los puertos serie y paralelo para la conexión de los programadores construidos con la PC, en conjunto con los softwares de programación utilizados, para programar los microcontroladores.



Motores paso a paso (pap). Los motores paso a paso son motores eléctricos de corriente continua, que convierten pulsos eléctricos en movimientos discretos.

La clave del motor paso a paso radica en alineaciones entre el estator y el rotor, en forma espaciada, creadas por polos magnéticos

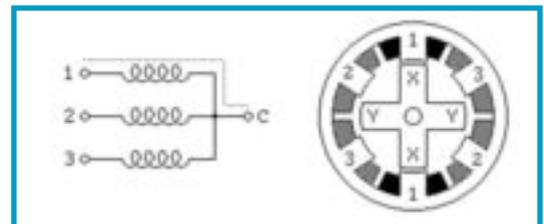
alternados dispuestos radialmente.

La característica principal de estos motores es el hecho de poder moverse un paso por cada pulso que se le aplica. Este paso puede variar desde 90° hasta pequeños movimientos de tan solo 1.8° e, incluso, hasta de 0.72°; es decir que, para el caso de un movimiento de 90°, se necesitan 4 pasos para completar un giro completo del motor (360°) y para el caso de un movimiento de 1,8° se necesitan 200 pasos para completar ese giro de 360°.

Los bobinados se encuentran ubicados en el estator y, según su distribución, podemos tener mayor o menor recorrido del rotor, dando como resultado mayor o menor ángulo de movimiento. Esta relación pasos-grados define la resolución de un motor pap.

Entonces, además de caracterizarse por el hecho de poder moverse un paso por cada pulso que se le aplica, este tipo de motor permite mantener la posición mecánica -también llamada enclavamiento-, sólo manteniendo la combinación de bobinados correspondientes alimentados con tensión; y, por el contrario, queda completamente libre si no circula corriente por ninguna de sus bobinas.

Los motores pap pueden ser de dos tipos: de imán permanente y de reluctancia variable.



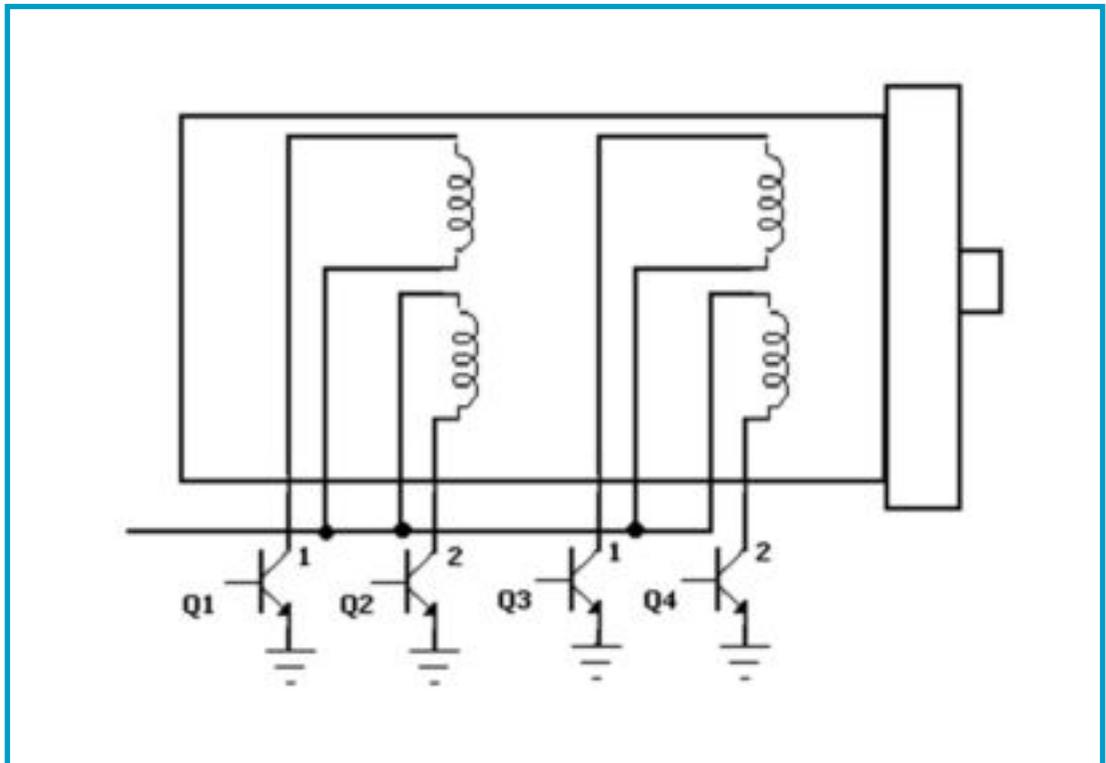
Motor pap de reluctancia variable; el estator del motor tiene tres bobinas conectadas con un terminal común, C

Los motores pap de imán permanente están constituidos por un rotor sobre el que van aplicados distintos imanes permanentes y por un cierto número de bobinas excitadoras bobinadas en su estator, de 2 o 4 devanados (bobinados). La cantidad corresponde al número de terminales que llegan al exterior. Por ejemplo lo bipolares cuatro cables (o menos) y los multipolares más de cuatro cables. ▶

Existe una multiplicidad de conexionados. Optamos por mostrar aquí uno de los más utilizados; se observan dos bobinas en cada uno de los estatores, donde la polaridad se invierte cuando se aplica tensión a uno de los devanados.



Estator de cuatro bobinas



Pasos	Q1	Q2	Q3	Q4
1	On	Off	On	Off
2	On	Off	Off	On
3	Off	On	Off	On
4	Off	On	On	Off
1	On	Off	On	Off

Para seleccionar un motor pap es conveniente tener en cuenta algunas de las siguientes características:

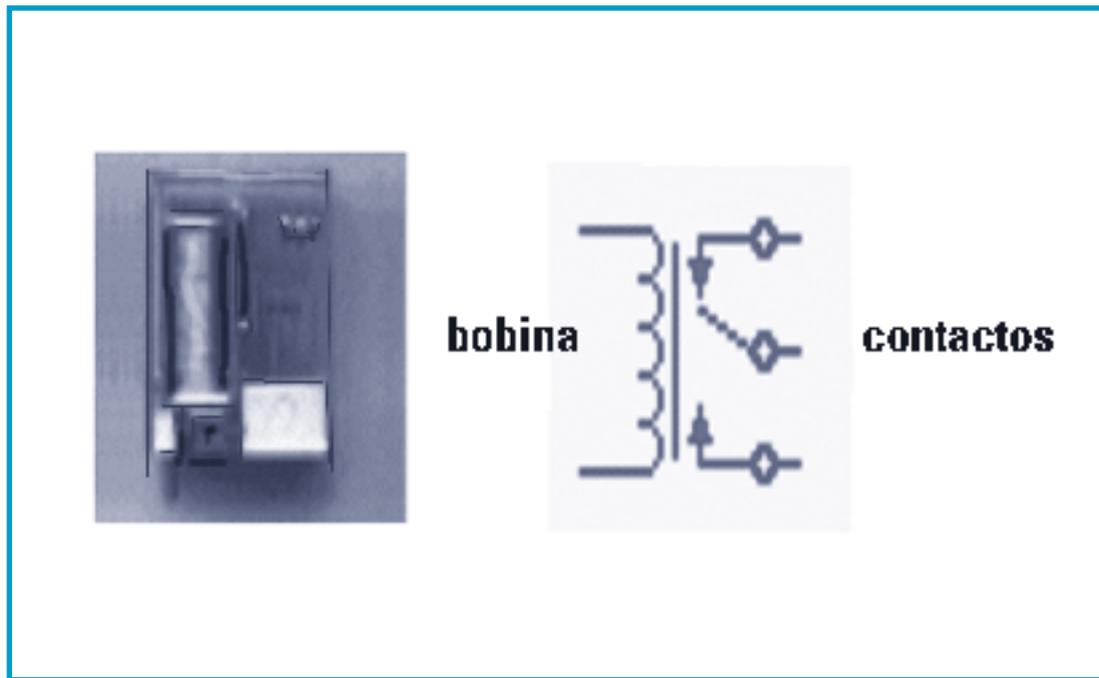
- El **voltaje**. Éste se encuentra impreso sobre la carcasa del motor o se especifica en la hoja de datos del fabricante. A veces es necesario exceder el voltaje nominal para obtener el par deseado; pero, esto ocasiona un mayor calentamiento e, incluso, al acortamiento de la vida del motor.
- La **resistencia** por bobina. Ésta determina la corriente del estator y, por tanto, afecta a la curva característica del par y a la velocidad máxima.
- La **resolución**. Es uno de los más importantes factores a tener en cuenta para la aplicación deseada. Plantea el ángulo de giro por cada paso. Los números grados-paso habituales son: 0.72, 1.8, 3.6, 7.5, 15 e, incluso, 90.
- El **tiempo de duración y la frecuencia de los pulsos aplicados**. Debido a que los motores paso a paso son dispositivos mecánicos y, como tal, deben vencer cier-

tas inercias, este aspecto reviste importancia. En tal sentido, el motor debe alcanzar el paso antes que la próxima secuencia de pulsos comience. En aplicaciones con secuencia lenta, el motor alcanza el reposo al final de cada paso. Si la velocidad aumenta, el movimiento cambia de discreto a continuo. En este caso, si no se tienen en cuenta las especificaciones, su comportamiento puede ser inestable; por ejemplo, el motor puede quedar fijo (no realiza ningún movimiento), puede comenzar a vibrar pero sin llegar a girar, puede girar erráticamente o puede llegar a girar en sentido opuesto. Para obtener un arranque suave y preciso, es recomendable comenzar con una frecuencia de pulso baja y, gradualmente, ir aumentándola hasta la velocidad deseada sin superar la máxima tolerada. El giro en inversa debería también ser realizado previamente, bajando la velocidad de giro y, luego, cambiar el sentido de rotación.

Relés. Son componentes electromagnéticos de interconexión entre circuitos de control y circuitos a controlar. Son dispositivos de potencia, compuestos por un electroimán que actúa como intermediario para activar un interruptor -que es totalmente independiente-.

Un relé consiste, entonces, en una bobina arrollada sobre un soporte metálico, de modo que, al circular por la bobina una cierta corriente, provoca la atracción de una lámina sobre el soporte metálico, activando los contactos eléctricos, y provocando el cierre o la apertura. Los parámetros básicos

para su manejo son la tensión de alimentación y el consumo de la bobina, y la corriente máxima que admiten los contactos.

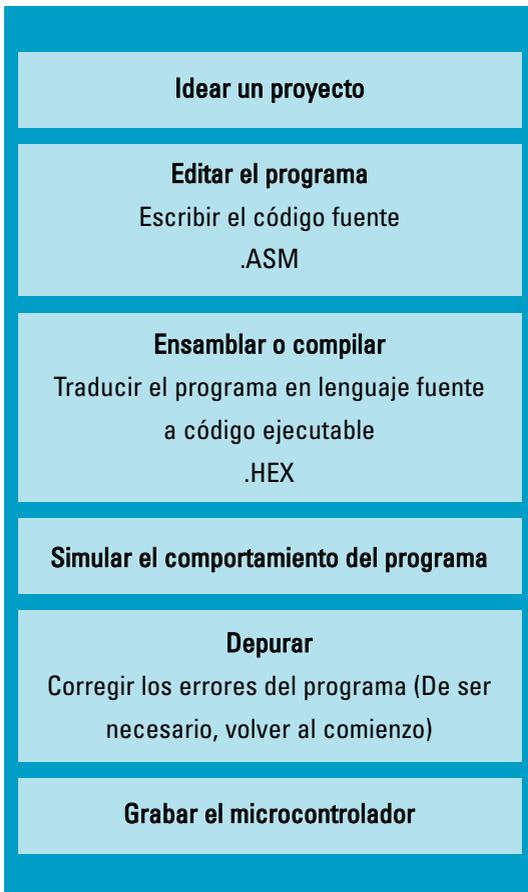


3. HACIA UNA RESOLUCIÓN TÉCNICA

Manual de procedimientos para la construcción y el funcionamiento del equipo

La programación de los microcontroladores PIC. El entorno MPLAB®

Éste es el diagrama que nos permite entender la secuencia de esa tarea.



Idear un proyecto y editar un programa. En primer lugar, para realizar y simular nuestros programas, necesitamos de un software en entorno Windows. Para ello disponemos del IDE® *-Integrate Development Enviroment-*, un ambiente de desarrollo integrado⁸.

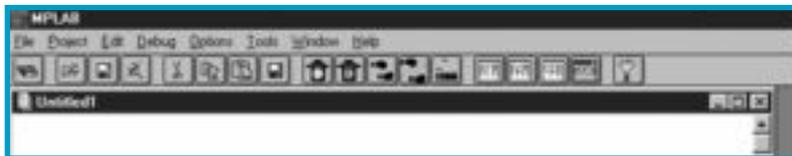
Este software es el encargado de compilar nuestro código fuente, el conjunto de instrucciones que editamos en el MPLAB® y que tienen extensión .asm.

Una vez depurado nuestro programa, éste genera un archivo de salida .hex. Este archivo se carga en una interface de usuario; las interfaces más conocidas son

- EPIC, para programación por puerto paralelo de la PC, y
- ICPROG, para programación por puerto serie.

⁸ Antes de comenzar a escribir programas, es necesario conocer las herramientas disponibles para desarrollarlos. Entre los muchos posibles, el entorno que nos resulta más interesante para formar parte integrante de nuestro equipo, es el MPLAB-IDE®; básicamente, porque es accesible y porque se puede obtener gratuitamente en www.microchip.com.

El MPLAB® utiliza, por defecto, el ensamblador MPASM, bajo MS-DOS. Una vez ingresado al programa, aparece la siguiente ventana:



Para comenzar a trabajar, vamos a File en la barra de herramientas y, dentro de esa opción, elegimos *New* en la ventana desplegable.

Allí, escribimos el programa, como si fuera un editor de textos, siguiendo algunas reglas de sintaxis que vamos a presentarle en un momento más. Una vez terminada esta tarea, seleccionamos la opción *File* de la barra de herramientas y, dentro de ésta, *Save*, asignándole un nombre con la extensión *.asm*.

Si este archivo ya existe dentro de la PC, para abrirlo utilizamos la opción *Open* del menú *File* o el icono típico:



En la primera línea, colocamos **List P = 16F84**, que permite definir el microcontrolador a utilizar durante todo el proceso.

La directiva **include** tiene por finalidad, a la hora de la compilación o ensamblado, tomar en cuenta los mnemónicos que en dicho archivo existen. Es útil usar esta directiva con el fin de ahorrar tiempo en la definición de registros en nuestro programa, cuidando utilizar el mismo nombre de registro usado por el archivo **.inc* correspondiente a nuestro microcontrolador.

La directiva **org**, seguida de una posición de memoria, indica al ensamblador dónde debe situar el siguiente fragmento de código. Se recomienda incluirla en todo programa, antes de la primera instrucción, como mínimo.

La directiva **end** es imprescindible, ya que le indica al ensamblador el final del programa.

El **;** es empleado a modo de comando **rem**; es decir, indica que, a continuación, se escribe un comentario.

El ensamblador exige una cierta tabulación mínima de sus distintos elementos. De este modo, la definición de variables puede escribirse en la primera columna de cualquier línea, mientras que las directivas e instrucciones deben ir en la segunda columna, como mínimo.

Las cifras se expresan de acuerdo con la tabla:

BASE	
DECIMAL	d'12' / .12
HEXADECIMAL	0x0c / h'0c' / 0c / 0ch
BINARIO	b'1010'

El uso de las mayúsculas y minúsculas obedece a una serie de normas de estilo que facilitan la lectura del código fuente. Un resumen de estas reglas es el siguiente:

- Directivas del compilador, en mayúsculas.
- Nombres de variables, en minúsculas.
- Mnemónicos (instrucciones), en mayúsculas.
- Programa bien tabulado.

A manera de ejemplo, listamos un programa:

```

;programa ilustrativo
;vversión 1.0
;fecha
;programa que enciende y apaga un
led con un pulsador
;colocado en Rb0

        #include"P16f83"
        List P=16f84

Flag    equ    0c
        org    00h
        goto   Principio
        org    04h
        goto   interr

Principio
        bsf    satus, 5
        movlw b'00000001'
        movwf TRISB
        movlw b'00000000'
        novwf TRISA
        movlw b'10010000'
        movwf INTCON
        bcf   flag, 0
        bcf   status, 5

de_nuevo
        btfss flag, 0
        goto  apaga
        goto  prende

apaga   bcf   PORTB,1
        goto  de_nuevo

prende  bsf   PORTB,1
        goto  de_nuevo

interr  btfsc PORTB,0
        goto  salir
        call delay_100
        btfss PORTB,0
        goto  $-1
        btfss INTCON, INTF
        goto  salir
        btfsc flag,0
        goto  poner_a_cero
        goto  poner_a_uno

poner_a_uno
        bsf   flag,0
        bcf   INTCON, INTF
        goto  salir

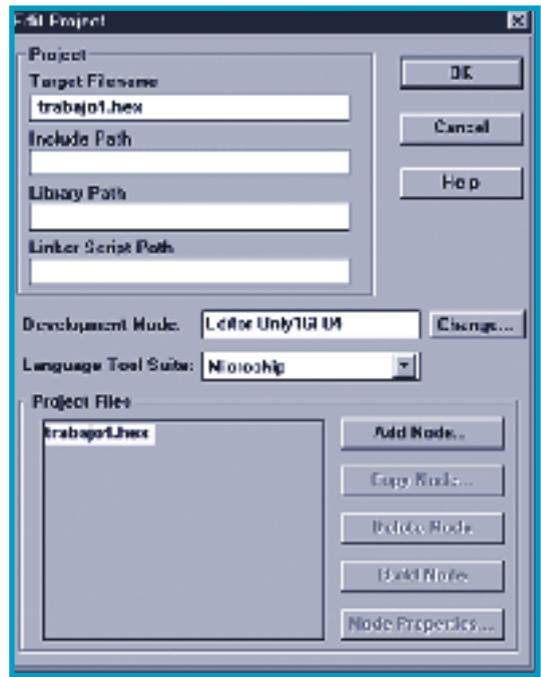
poner_a_cero
        bcf   flag,0
        bcf   INCTON, INTF
        goto  salir
salir   retfie
        end

```

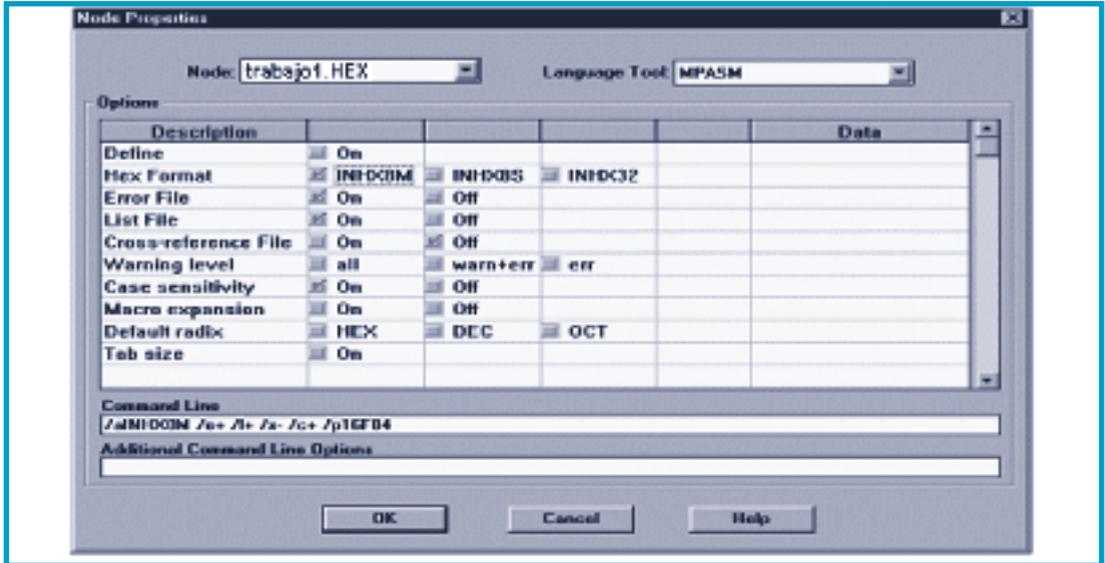
Una vez que el archivo .asm existe, creamos un proyecto que relacione el archivo .hex con el anterior.

Para hacer esto, de la barra de herramientas *Project*, elegimos *New project*. Aparece una pantalla con ese título. En esta pantalla, asignamos el nombre al proyecto con extensión .pjt (por ejemplo: trabajo1.pjt.) y damos OK.

Nos encontramos, entonces, con la ventana *Edit project*:



Manteniendo nuestro ejemplo, del sector Project files seleccionamos trabajo1 [.hex]. Aparece la opción Node properties. Si la elegimos, vamos a poder seleccionar el Hex format deseado; es decir, el tipo de fichero en el que el ensamblador debe convertir los *.asm.



Luego de pulsar OK, encontramos que el proyecto sigue estando vacío. Es necesario que añadamos el programa (*.asm) para que pueda ensamblarse y probarse.

Esto es posible, si el proyecto y programa están en la misma carpeta. Podemos observar que, ahora, tenemos activa la opción *Add node* de la ventana *Project files*. Entonces, seleccionamos y añadimos el archivo *.asm.

Ensamblar o compilar. En este momento, estamos en condiciones de crear un nuevo proyecto. Para hacerlo, oprimimos la tecla *F10* que tiene por misión ensamblar el archivo *.asm y convertirlo en un archivo *.hex.

Podemos realizar el proceso de ensamblado de la siguiente manera:

- Sobre el menú *Project*, basta con pulsar la opción *Build node*, que ensambla sólo

la ventana activa o la opción *Build all*, que ensambla todos los nodos del proyecto.

- Por último, la opción *Make Project* ensambla todos los nodos de un proyecto y los unifica en un único *.hex.

Es importante destacar que el programa va a crear el archivo *.hex siempre y cuando no haya errores en la compilación. Si los hubiere, el programa se encarga de marcarlos, dando los números de línea donde figura el error. Con un doble clic sobre este número, el programa bifurca directamente al código, indicando dónde está el error.

Ahora, la tarea consiste en ir corrigiendo estos errores, uniendo esta corrección con el pulsado de la tecla *F10*, para que realice una nueva compilación hasta que ésta aparezca libre de errores. Cuando sucede esto, el programa emite el mensaje *Build succesfully*, luego de lo cual es conveniente hacer un *Save project* del menú *Project*.

Simular el comportamiento del programa. Una vez corregidos todos los errores, el programa está listo para ser grabado en el PIC y probado. Pero, resulta más fiable si, antes, llevamos a cabo una simulación por software.

MPLAB® tiene una herramienta de simulación de software, el MPLAB-SIM, que puede ser activada, desde la barra de herramientas, con la opción *Debug*.



La simulación comienza mediante la opción *Run* (semáforo verde) y se detiene con *Halt* (semáforo rojo); pero, no comienza a correr si no hace un *Reset* mediante la pulsación de:



Este icono es *Step*, ejecución paso a paso; avanza una línea de programa cada vez que se lo pulsa.



Este icono es *Change Program Counter*; es utilizado para cambiar el contador de programa (salto a otra línea de memoria de programa en ejecución para, por ejemplo, probar sólo una rutina específica, si el resto ya se sabe correcto).



Este icono es *Create a New Watch Window*; permite editar variables para ver su valor durante la ejecución.



Este comando permite cambiar el valor de direcciones de memoria en tiempo de simulación y en cualquiera de sus áreas (datos, programa, EPROM, pila); lo hace directamente (por su número) o a

través de su nombre (mnemotécnico definido en el ensamblador; en el ejemplo es *Operando1*). Combinado con los dos anteriores, es muy útil para probar fácilmente todas las variantes de una rutina o zona de código determinada, sin tener que ejecutar todo el código, de nuevo, para cada una.



Con este comando se pueden definir puntos de parada *-Breaks-* en la ejecución para, mediante *Run*, no recorrer línea a línea todo el programa. Esto es imprescindible si deseamos ejecutar todo un proceso de golpe hasta esa línea.



Define condiciones de parada *-Conditional Breaks-*; es decir, valores de variables o pines (E/S) ante las que parar, si se producen.

Otras posibilidades que contiene el menú *Options* son:

- Por ejemplo, con su comando *Development mode...* es posible elegir el tipo de microprocesador sobre el que simular y activar el modo de simulación (*MPLAB-SIM simulator*), no siempre activo por defecto.
- Con el submenú *Processor Setup* podemos cambiar condiciones como la velocidad del reloj (para controlar el tiempo de ejecución) o la activación del *Watchdog*.
- A través del menú *Window*, puede ver la memoria de programa (submenú *Program Memory*) y la de la *EEPROM* (si lo desea y la va a utilizar; submenú *EEPROM Memory*).

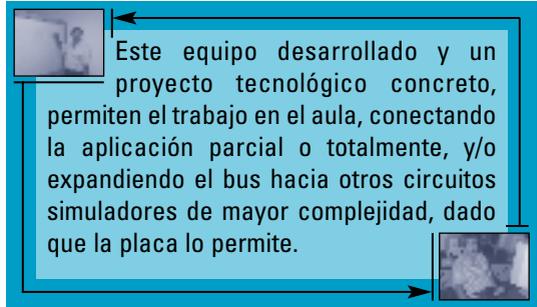
Vamos a plantearle, ahora, un ejemplo de aplicación, a modo de visualización de los distintos pasos que sus alumnos van a tener en cuenta, desde que se plantea el problema a solucionar hasta el uso del equipo desarrollado.

Una vez que los alumnos han preparado el código fuente (*.asm) de nuestro programa y lo han convertido en un fichero *.hex, utilizando el ensamblador MPLAB®, el programa puede ser volcado sobre la memoria de programa flash del microcontrolador. Estamos, entonces, en la fase de programación.

```
:020000000528D1  
:06000A008316F0308500B2  
:0E001000FF30860083120608FF3A85000B2899  
:00000001FF
```

*Aspecto que tiene el archivo *.hex*

Este código es el que va a ser volcado al microcontrolador.



Depurar y grabar el microcontrolador. Para cumplimentar este trabajo, necesitamos dos elementos; el programa de grabado de microcontroladores y el programador.

La primera tarea es fácilmente realizable, pues todos los programas necesarios son gratuitos y fáciles de obtener.

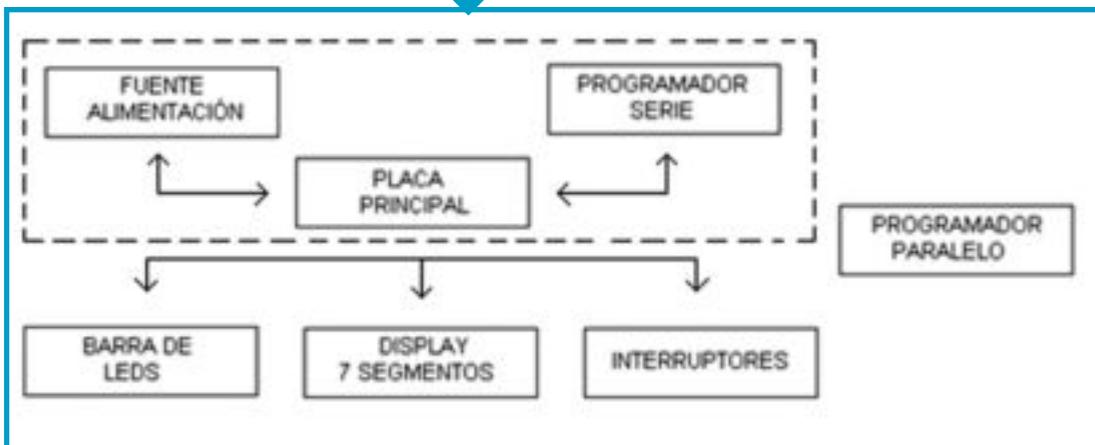
Para la segunda tarea, existen programadores que utilizan el puerto serie de la PC y otros que utilizan el puerto paralelo; ambos son sencillos de llevar a cabo, no tienen grandes complicaciones de fabricación y forman parte del proyecto que estamos manejando. Cualquiera sea el programador elegido, el programa que está en la PC se descarga utilizando el programa adecuado y concretándose, entonces, la grabación del PIC.

El equipo

Decíamos que, luego de realizado el programa de un proyecto determinado, es necesario grabar en el microcontrolador seleccionado y probar si el desarrollo funciona.

Es aquí donde el equipo de **simuladores interconectables basados en lógica digital**, encuentra su aplicación.

Consideremos el diagrama en bloques que esquematiza la conformación del equipo:



¿Advierte, usted, el por qué del término "interconectables"? Las distintas placas se interconectan de acuerdo con la necesidad de los productos.

El equipo puede soportar otros dispositivos como, por ejemplo, motores paso a paso en el caso que se diseñe y anexe una placa de potencia para esos motores -la que no está incluida en esta versión- o el manejo de un display LCD, cuyo diseño de placa también queda como tarea para los estudiantes de su curso. Es decir que el *kit* no queda limitado a las posibilidades que le presentamos sino que puede ser ampliado a otras nuevas, ya que permite realizar experimentos con la totalidad de las instrucciones del microcontrolador, sin la necesidad de componentes adicionales.

Esta modularidad del sistema da idea de su practicidad, a la hora de interconectar varias placas de aplicación para un mismo diseño.

Los componentes

Vamos a describir, ahora, las diversas placas que conforman el equipo.



Placa fuente de alimentación

La fuente de alimentación del circuito se compone de un transformador de 220 V / 12 V - 500 mA, que se conecta a la placa a través de los pines JP1, un rectificador de onda completa (D1-D4) con filtro capacitivo (C1) y estabilización de tensión usando reguladores de la línea LM78XX.

La fuente tiene dos salidas, una de 5 V a través de un LM7805 (IC2) y otra de 13,8 V que utiliza un regulador LM7812 (IC1) con 3 diodos en serie (D5-D7) para elevar el nivel de tensión, logrando el adecuado.

La intensidad de corriente no es demasiado elevada; resulta, aproximadamente, de valores menores a 200 mA.

También dispone de un diodo 1N4148 (D8) para protección del circuito frente a posibles inversiones de polaridad y un led (L1) rojo que muestra la condición de funcionamiento.

Integra una segunda tira de pines JP2 que permite la conexión de esta placa fuente al resto de las placas que lo necesiten.

Los componentes de la placa fuente de alimentación son:

Resistencias:

R1: 1,5 k Ω

R2: 1 k Ω

Transformador:

T: 220 V/12 V - 500 mA

Capacitores:

C1: 1000 microfaradios 50 V electrolítico

C2, C3: 0.1 microfaradios 50 V cerámico

C4, C5: 47 microfaradios 50 V electrolítico

Semiconductores:

L1: Led rojo

D1, D2, D3, D4, D5, D6, D7: 1N4001

D8: 1N4148

IC1: LM 7812

IC2: LM 7805

Conectores:

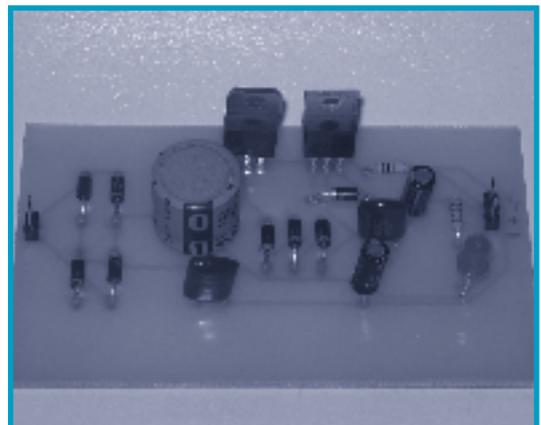
JP1: Tira de 2 pines SIP

JP2: Tira de 3 pines SIP

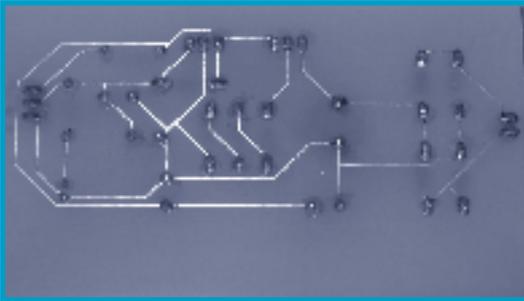
En el CD que acompaña esta publicación, usted puede acceder a:

Placa fuente:

- Esquema eléctrico.
- Ruteo de pistas.
- Vista superior.



Placa fuente terminada; vista superior



Placa fuente terminada; vista inferior

Placa principal

En esta placa se ubican tres zócalos donde se insertarán los integrados IC3, IC4 e IC5. Cada uno de ellos tiene un número determinado de pines, Z1 = 8, Z2 = 18 y Z3 = 28 respectivamente, que permiten cubrir la gama de modelos de microcontroladores detallada en la tabla general de Microchip.

La placa trabaja con una frecuencia de reloj de 4 MHz, en general. Ésta se consigue mediante un oscilador a cristal de cuarzo o un oscilador RC. Se puede seleccionar uno u otro mediante el conmutador S1. La frecuencia del oscilador RC puede variarse en forma libre, con el preset P1.

Esta placa posee, además, tres conectores DB25 hembra que permiten la interconexión entre el microcontrolador y las distintas placas periféricas (led, interruptores, etc.), para la simulación del proyecto a desarrollar.

Debemos tener muy presente que se puede programar o probar sólo un microcontrolador a la vez.

Mediante la doble llave inversora S2 se selecciona una de las dos posibilidades: programación serie o simulación.

El pulsador S3 permite un reset en el microcontrolador en el MCLR, para reiniciarlo.

La tira de 9 pines JP3 permite la conexión de la placa principal con la fuente de alimentación y el programador serie.

En el CD va a encontrar:

Placa principal:

- Esquema eléctrico.
- Ruteo de pistas.
- Ruteo de los componentes.
- Ruteo completo.
- Vista superior.

Los componentes de la placa principal son:

Resistencias:

R3: 270 Ω

R4: 4,7 k Ω

Preset P1: 100 k Ω

Capacitores:

C6, C6a, C6b, C6c: 100 nanofaradios cerámico

C7, C8: 27 picofaradios cerámico

Cristal:

4 MHz

Conectores:

JP3: Tira de 9 pines SIP

X1, X2, X3: DB25 hembra

Zócalos:

Z1: DIP 8 pines

Z2: DIP 18 pines

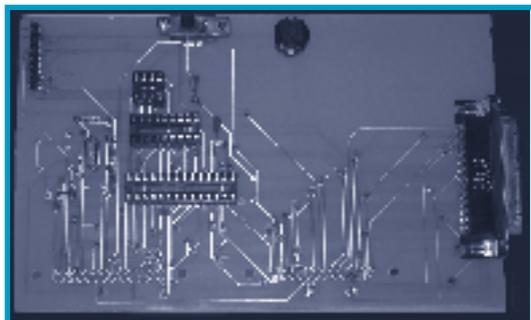
Z3: DIP 28 pines

Interruptores:

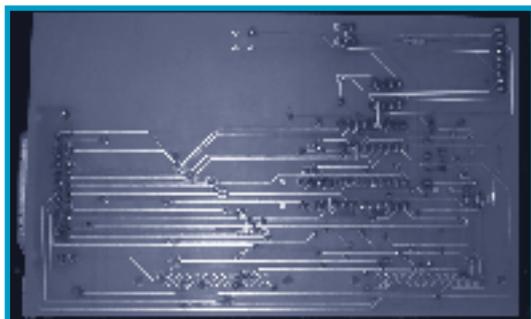
S1: Conmutador simple

S2: Llave doble inversora

S3: Pulsador



Placa principal terminada; vista superior



Placa principal terminada; vista inferior

Placa del programador por puerto serie

El programa correspondiente al proyecto a desarrollar debe ser escrito en la memoria de instrucciones del microcontrolador y, para ello, es necesaria la placa de grabación.

Cuando -con la llave S2, de la placa principal- se selecciona el modo correspondiente a programación, se aplica una tensión de continua de 13,8 V en la entrada MCLR. Entonces, el microcontrolador cambia su modo de trabajo a modo de programación. Además S2 conecta a masa los terminales de habilitación del buffer integrado IC6 (74HCT125) de la placa del programador serie.

La tensión de programación se genera en la placa de fuente de alimentación que ya hemos descrito.

El diodo led L2 verde es el que indica la activación en el modo de programación.

La placa se conecta a la PC a través de la interfaz serie, por medio de un conector DB9, X4. La conexión se puede realizar mediante un cable normal RS 232 (no un cable de módem nulo), con un conector DB9 macho en el lado de la placa y otro conector tradicional DB9 hembra en el lado de la PC.

Verifique que el módulo esté conectado, antes de iniciar la secuencia.

La conexión a la PC debe realizarse con la unidad desconectada.

Las dos señales de programación, *Data* y *Clock*, llegan a través de los controladores IC6 pin 8 e IC6 pin 11 (escritura respecto del microcontrolador) hacia las entradas del microcontrolador RB6 y RB7.

El controlador IC6 pin 5 suministra la lectura de datos desde el microcontrolador hacia la inter-

faz. La etapa de controladores IC6 pin 3 se utiliza para controlar la tensión de programación y la señal de Reset del microcontrolador.

La tira de 6 pines JP4 permite la conexión del programador serie a la placa principal.

Los componentes de la placa del programador por puerto serie son:

Resistencias:

R5, R6, R7, R8, R9, R10: 10 k Ω

R 11: 470 Ω

R12: 3,3 k Ω

RA: 470 Ω

Semiconductores:

IC6: 74HCT125

Q1, Q2: BC548C

L2: Led verde

D9: 1N4148

Zócalo:

Z4: DIP 14 pines

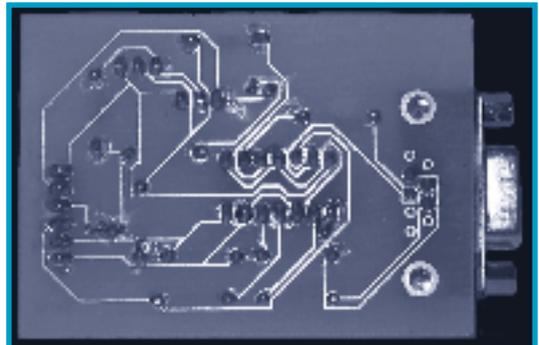
Conector:

X4: DB9 hembra

JP4: Tira de 6 pines SIP



Placa del programador por puerto serie terminada; vista superior



Placa del programador por puerto serie terminada; vista inferior

En el CD va a encontrar:

Placa del programador por puerto serie:

- Esquema eléctrico.
- Ruteo de pistas.
- Vista superior.

Placa del programador paralelo

Aunque las memorias de PIC se programan, generalmente, por el puerto serie de la PC, el equipo de simuladores interconectables agrega un programador también en placa separada que se conecta al puerto paralelo de la PC. Esto se debe a que este puerto se puede controlar fácilmente por software y, además, a que suministra niveles TTL para ser utilizados. También se desarrolla en función de que, mientras se realiza la programación se deben disponer de algunas líneas de control para conmutar las distintas alimentaciones

del microcontrolador y esto es más fácil en programación por puerto paralelo que por serie.

Este programador consta de tres zócalos (Z5, Z6 y Z7) de 8, 18 y 28 pines, que permiten programar los microcontroladores PIC sobre la placa. Cuenta, además, con una tira de pines JP5 que permite programar, en forma externa, cualquier otro microcontrolador de la empresa *Microchip*® que no se adapte a los zócalos Z5, Z6 y Z7.

La alimentación del programador se consigue a través de un transformador (T1) 220 V / 12 V - 500 mA con un puente de diodos (D10 a D13) para la rectificación y un par de capacitores (C9 - C10) para su filtrado. Esta fuente entrega dos tensiones estabilizadas; una, a través del IC9 7805 de 5 V (Vcc) para la alimentación y, otra, con el IC8 7812 de 12 V y tres diodos en serie (D14 a D16) para lograr los 13,8 V (V1) necesarios para la programación.

La intensidad de corriente no es demasiado elevada; aproximadamente, sus valores son menores a 200 mA.

También dispone de un diodo led (L4) verde que se enciende cuando el programador está bajo tensión, con el fin de señalar el buen funcionamiento de la alimentación.

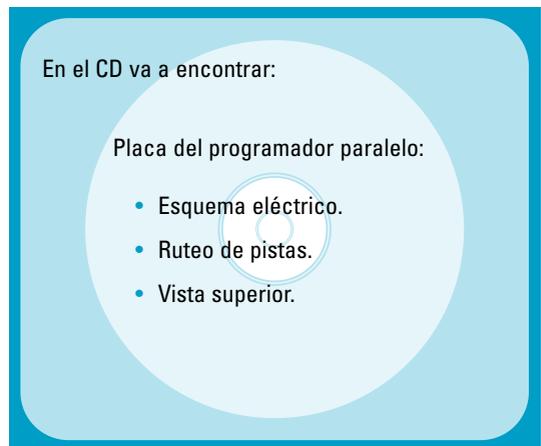
Por medio del buffer inversor IC7 (SN74LS07) los datos son transferidos del puerto paralelo al microcontrolador -y viceversa- a través de una ficha de conexión DB9 hembra (X6) dispuesta en la placa y una ficha de conexión DB25 macho (X5) al puerto paralelo de la PC.

Los datos a programar en el circuito transitan por la puerta IC7E, (pin 10) pasando por IC7F (pin 13) en caso de lectura del circuito.

En cuanto al reloj de programación, éste pasa por IC7D (pin 8).

El transistor BC558C Q3, que toma los datos del pin 4 del buffer IC7, permite aplicar la tensión alta de programación VPP a los correspondientes pines de cada uno de los tres zócalos de programación. El led rojo L3, conectado al colector de este transistor, da indicación de la fase de programación.

El transistor Q4 es un BC558C y entrega la tensión normal de alimentación VDD, aplicada a los tres zócalos, lo que permite alimentar el circuito a programar sólo cuando es necesario acceder a él. De esta forma, evita cualquier problema durante su inserción o extracción del circuito integrado del zócalo de programación. El led verde L4 se conecta al colector de este transistor.



Los componentes de la placa del progra-

mador paralelo son:

Resistencias:

R13, R14, R15: 8,2 k Ω

R16, R17: 1,8 k Ω

R18, R19: 12 k Ω

R20: 10 k Ω

R21: 560 Ω

R22: 1,2 k Ω

Capacitores:

C9: 1000 microfaradios 16 V electrolítico

C10: 1 microfaradios 16 V electrolítico

C11: 100 nanofaradios

Semiconductores:

L3: Led rojo

L4: Led verde

D10, D11, D12, D13, D14, D15, D16:
1N4007

Q3, Q4: BC558B

IC7: SN74LS07

IC8: LM 7812

IC9: LM 7805

Zócalos:

Z5: DIP 8 pines

Z6: DIP 18 pines

Z7: DIP 28 pines

Z8: DIP 14 pines

Conectores:

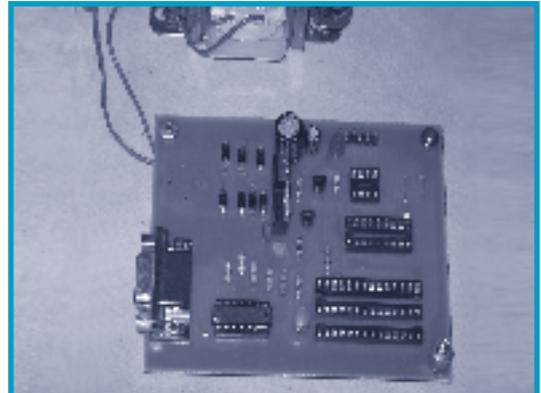
X5: DB25 (macho)

X6: DB9 (hembra)

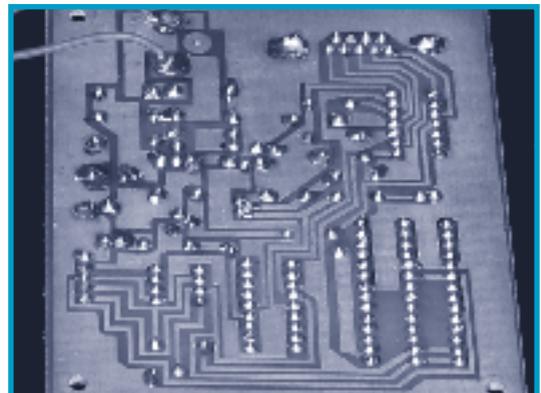
JP5: Tira de 5 pines

Transformador:

T: 220 V - 12 V - 500 mA



*Placa del programador paralelo terminada;
vista superior*



*Placa del programador paralelo terminada;
vista inferior*

Placas para el hardware periférico

Además del sistema principal compuesto por la placa de la fuente de alimentación, el programador serie y la placa principal, se cuen-

Además, se dispone una placa adicional para programación del microcontrolador a través del puerto paralelo de la PC, que posteriormente detallaremos.

ta con un hardware periférico compuesto por tres placas interconectables separadas para realizar los distintos ensayos:

- Placa de led.
- Placa de interruptores.
- Placa del display de siete segmentos.

Cada una de estas placas puede montarse sobre cualquier otro proyecto más complejo a desarrollar, dado que cada entrada-salida puede seleccionarse en forma individual.

Placa de led. Los diodos led que van numerados de L5 a L18 indican los valores lógicos presentes en los puertos RA y RB del microcontrolador; más precisamente:

- de L5 a L10 para RA y
- de L11 a L18 para RB.

Estas indicaciones lumínicas son útiles cuando se están depurando aplicaciones a baja velocidad mediante el oscilador RC. También se pueden utilizar en ejercicios básicos de programación -por ejemplo, un diodo led que parpadee, una luz que se desplace, un visualizador de barra gráfica o un regulador de intensidad con diodos led-.

La alimentación de cada led es provista por los puertos del microcontrolador. Cada led toma masa independientemente, a través de los microconmutadores o jumpers JP6 y JP7; de esta manera, se habilitan o inhabilitan en forma individual, de acuerdo a la necesidad.

El conector (X7) DB25 permite la interconexión de la placa de led a la placa principal.

El transistor Q5 de la placa de led se utiliza

para activar al led L9, ya que la salida RB4 es a colector abierto y, si se hubiera conectado igual que los otros led, éste no hubiera funcionado.



Los componentes de la placa de led son:

Resistencias:

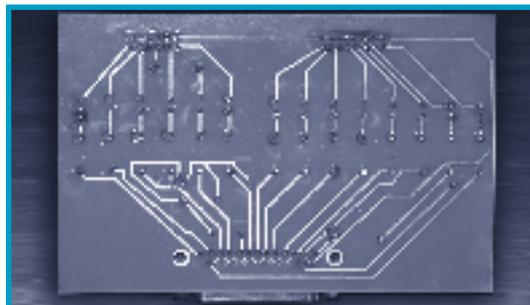
R23, R24, R25, R26, R27, R28, R29, R30, R31, R32, R33, R34, R35, R36: 560 Ω
R37: 470 k Ω

Semiconductores:

L5, L6, L7, L8, L9, L10, L11, L12, L13, L14, L15, L16, L17, L18: Led rojos
Q5: BC 547

Conectores:

JP6: 6 jumpers
JP7: 8 jumpers
X7: DB25 (hembra)



Placa de led terminada; vista inferior

Placa de interruptores

Esta placa consta de 8 pulsadores S1 a S8, que permiten simular interrupciones externas, activaciones, etc. sobre el microcontrolador.

Con los jumpers JP8 se obtiene la posibilidad de habilitar o inhabilitar en forma individual a cada uno de los pulsadores, de acuerdo con la necesidad del proyecto.

Al presionar sobre cada pulsador, éstos se cierran y se conectan a masa, generando un nivel lógico "0" en las entradas del puerto RB0 a RB7.

A su vez, los pulsadores S5 a S8 pueden quedar fijos a nivel "0" por medio de los jumpers JP9 a JP12.

En el CD va a encontrar:

Placa de interruptores:

- Esquema eléctrico.
- Ruteo de pistas.
- Vista superior.

Los componentes de la placa de interruptores son:

Resistencias:

R38, R40, R42, R44, R46, R48, R50, R52: 100 Ω

R39, R41, R43, R45, R47, R49, R51, R53: 10 k Ω

Capacitores:

C12, C13, C14, C15, C16, C17, C18, C19: 100 nanofaradios

Conectores:

X8: DB25 (hembra)

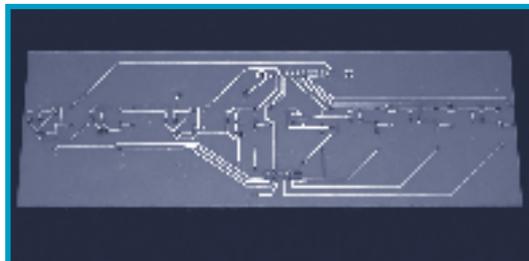
JP8: 8 jumpers

JP9, JP10, JP11, JP12: jumpers

S1, S2, S3, S4, S5, S6, S7, S8: pulsadores



Placa de interruptores terminada; vista superior



Placa de interruptores terminada; vista inferior

Placa del display de siete segmentos

Otra placa que compone el equipo de simuladores es un módulo visualizador con cuatro display de siete segmentos, de ánodo común. Los display pueden habilitarse o inhabilitarse en forma individual, de acuerdo con la necesidad del proyecto.

Para poder aprovechar la mayor cantidad de entradas-salidas del microcontrolador, se ha utilizado un decodificador de BCD a display

de segmentos, como manera de reducir la cantidad de conexiones necesarias para poder manejar el display.

Además, se usan cuatro transistores 2N3906 para realizar el multiplexado de los datos, evitando, de esta manera, la utilización de cuatro decodificadores de BCD a 7 segmentos.

También permite la posibilidad de conexión o desconexión, según la dificultad de la simulación, aprovechando al máximo los cuatro display o utilizando menor cantidad -al dejar los pines correspondientes del microcontrolador libres para poder ser utilizados con otro propósito-.

Estos display pueden ser utilizados para visualizar cuentas o cantidades numéricas, tarea que no se podría realizar por ejemplo con la placa de led; permite, además, reducir los circuitos asociados al desarrollo planteado.

Semiconductores:
Q6, Q7, Q8, Q9: 2N3906
IC10: 74LS47

Resistencias:
R54, R55, R56, R57: 3,9 k Ω
R58: 1 k Ω
R59, R60, R61, R62, R63, R64, R65: 47 Ω

Conectores:
X9: DB25 (hembra)
JP13: Jumpers doble
JP14, JP15, JP16: Jumpers simple

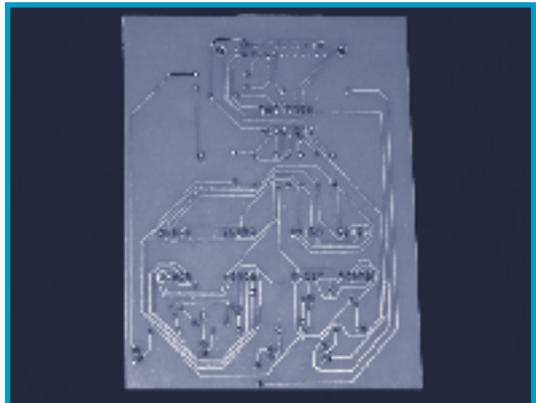
En el CD va a encontrar:

Placa del display de siete segmentos:

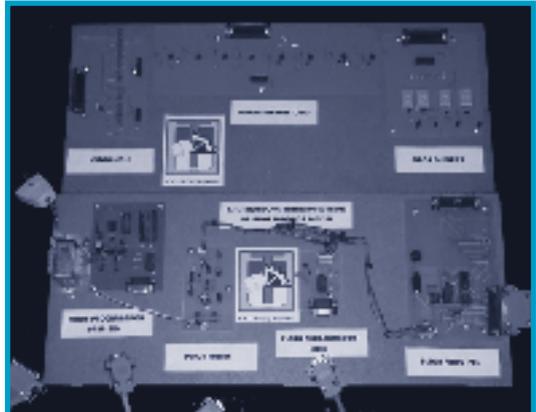
- Esquema eléctrico.
- Ruteo de pistas.
- Vista superior.

Los componentes de la placa del display de siete segmentos son:

Display:
DIS1, DIS2, DIS3, DIS4: HD - H101 (ánodo común)



Placa del display de siete segmentos terminada; vista inferior



Interconexión final de las placas

El desarrollo del circuito impreso

El desarrollo de un circuito impreso es un paso imprescindible para la concreción de un circuito electrónico.

Una somera descripción de los materiales y herramientas utilizados es:

- **Taladro miniatura.** Para realizar los agujeros donde irán montados los componentes. Es importante que la pieza que sujeta las brocas garantice un buen agarre a éstas para que no se descentren mientras se realiza el taladrado, y para que las perforaciones sean circulares y no queden deformadas.
- **Brocas.** Apropriadas para metal y de grosores comprendidos entre los 0.7 mm para los pads⁹ y 3 mm para el mecanizado. Se utilizan también las de 0.8, 1 y 1.2 mm. Es necesario tener varias de repuesto, porque se parten fácilmente. Las mechas de diámetros menores al milímetro son, a veces, difíciles de conseguir; por lo tanto, los diámetros aquí propuestos no deben tomarse como una regla fija sino, más bien, como una sugerencia de trabajo.
- **Soldador.** De tipo lápiz y de unos 30 watt de potencia.
- **Estaño.** Adquirido en casas de electrónica, para respetar la proporción de estaño-plomo 60/40.
- **Cubetas de plástico.** Son utilizadas para hacer la reacción química. Su tamaño

depende del de la placa a realizar. Sus bordes son lo suficientemente altos como para no derramar nada al mover los líquidos.

- **Guantes de plástico y pinzas para tomar las placas.**
- **Téster.** Para comprobar la continuidad de las pistas y la presencia de señal en algún pin.
- **Placas cobreadas.** Se adquieren en los comercios de electrónica; le sugerimos elegir una de buena calidad.
- **Caminos transferibles, fibrón indeleble.** Para imprimir el diseño de las pistas.
- **Percloruro férrico.** Como químico para eliminar el cobre sobrante del diseño de la placa.
- **Barniz eléctrico.** Para proteger la oxidación las pistas, una vez soldados todos los componentes.
- **Alcohol etílico común, algodón y virulana.** Para la limpieza de las placas.

a. La fabricación del circuito impreso

Con el diseño del ruteo de pistas que le acercamos en el CD anexo, procedemos a cortar la placa de cobre con una sierra, según la medida correspondiente a cada una de ellas.

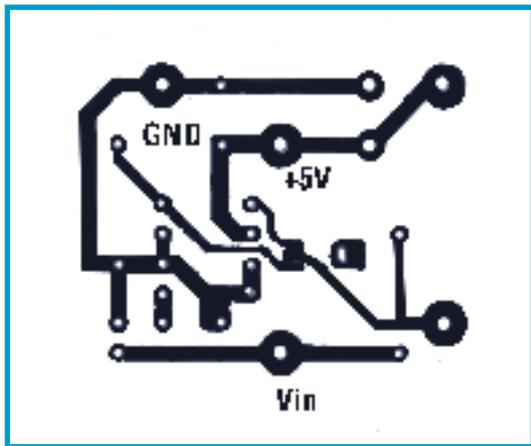
Luego, limpiamos las placas cuidadosamente con una virulana o con algún otro dispositivo mecánico similar, en forma muy suave para no marcar el cobre.

⁹ Islas formadas por el engrosamiento de la pista en las cuales se soldarán los componentes.

Las placas más sencillas, de simple faz, se pueden fabricar en forma manual, usando ruteadores de pads y caminos transferibles, o trazando las pistas con fibrón indeleble.

Para placas de diseño complicado -por la cantidad de componentes y por su disposición, o por ser placas doble faz como la principal-, le recomendamos un método más eficiente. Un tipo de traslado del diseño del impreso a la placa cobreada suele ser el óptico; pero, un equipo de estas características resulta muy oneroso, por lo que le recomendamos tercerizar la tarea a empresas dedicadas a tal efecto.

Para el caso de las placas más sencillas (placa de led, placa de interruptores, placa de display, placa de fuente, placa de programador serie), procedemos a dibujar o transferir los pads sobre el cobre. Luego, los unimos a través de los caminos transferibles o pintamos las pistas correspondientes.



Una vez terminado el diseño, colocamos las placas en la cubeta con percloruro férrico, para quitar el cobre que no fue cubierto por la tinta indeleble o los caminos transferibles.

Tenemos en cuenta que este producto químico es muy corrosivo, capaz de atacar tela, metales, etc., por lo que actuamos con mucho cuidado en su manipulación y tomamos las medidas de seguridad adecuadas.

Para esta tarea recomendamos:

- Usar guantes.
- Colocar suficiente percloruro férrico en la cubeta, como para cubrir la placa.
- Introducir la placa con las pistas hacia arriba, con mucho cuidado de no salpicar. Luego de unos minutos, el cobre que no estaba protegido por la tinta indeleble y/o los caminos transferibles se ha ido.
- Con la pinza pertinente, procedemos a retirar las placas de la cubeta, lavándolas con abundante agua.

b. El taladrado

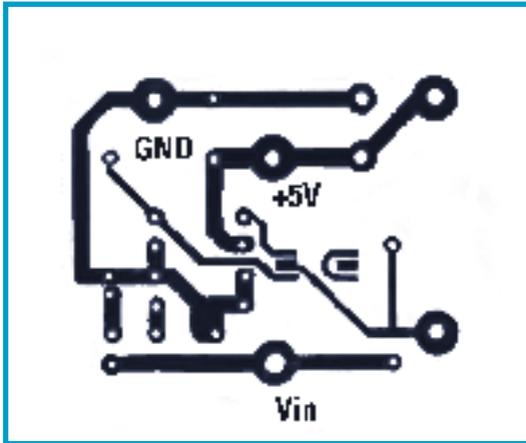
Para hacer los agujeros de los pads donde se insertarán los componentes, utilizamos el taladro miniatura con varios grosores de broca, todas ellas para metal.

Primero, utilizamos una broca de 0.7 mm, para hacer todos los agujeros. Si el diseño tiene el centro del pad en blanco, evitará tener que marcar éste con un punzón para que la broca no se deslice, quedando perfectamente centrado.

Luego, repasamos con una broca de mayor

diámetro los pads que requieran un diámetro mayor, usando siempre la de menor grosor posible.

c. La soldadura



Luego de limpiar la placa, podemos dar una fina capa de barniz eléctrico a la placa, para detener su proceso de oxidación, especialmente si pasará mucho tiempo hasta comenzar las tareas de soldado.

Antes de realizar la primera soldadura, comprobamos que todos los componentes entren perfectamente, para evitar realizar un agujero entre componentes ya soldados, pues ello resulta muy complicado.

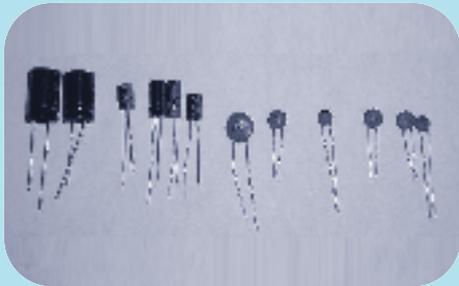
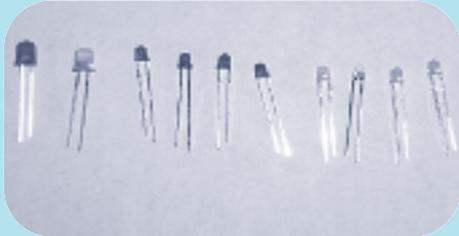
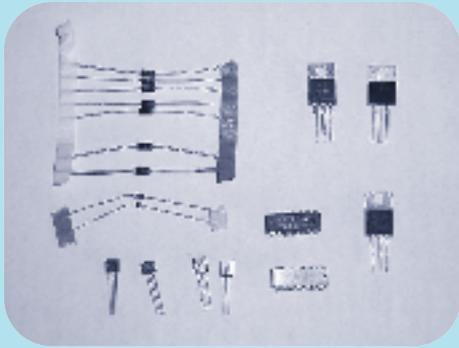
Comenzamos soldando los puentes, zócalos de integrados y las resistencias, por ser las de menor altura.

Seguimos con los capacitores y, luego, con los componentes semiconductores: diodos, transistores, etc., evitando así el calentamiento de estos últimos.

Es necesario aclarar que, en la placa princi-

pal, por ser ésta doble faz, debemos comenzar soldando las uniones entre ambas superficies o caras de la placa y, luego, continuamos soldando de acuerdo con el orden establecido previamente.





Componentes

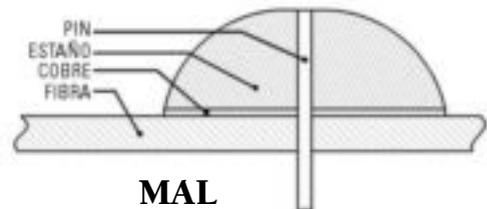
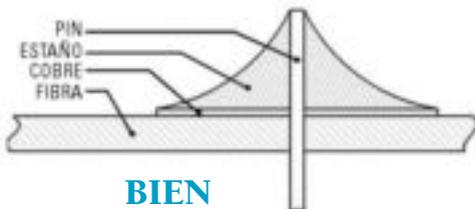
El soldador debe contar con conexión a toma de tierra, para que las cargas electrostáticas no estropeen los componentes.

Realizamos la soldadura introduciendo todos los pines o extremos de los componentes en el pad correspondiente, de acuerdo con la ubicación en el lado "componentes" del circuito impreso, según la disposición mostrada en las distintas figuras. Verificamos, previamente, la limpieza de los terminales a soldar.



Con el soldador calentamos el pad y la pata de conexión para, luego, aplicar el estaño, que se funde inmediatamente, y recubre todo el pad y el pin, homogéneamente.

El estaño debe quedar brillante y tener forma de carpa y no de globo; porque, esto último es un indicio de una mala soldadura (llamada soldadura fría) que no produce un buen contacto eléctrico.



Es importante no sobrecalentar los componentes con el soldador, ya que se podrían quemar; sobre todo, si se trata de semiconductores.

Una vez soldados todos los componentes, cortamos los restos de patas sobrantes y protegemos la placa con el barniz, para que las pistas de cobre no se oxiden con el tiempo.

Puede hacer su propio protector, diluyendo un poco de resina vegetal (se consigue en cualquier ferretería) en acetona y aplicándola con un pincel. Esta solución es económica pero deja todo pegajoso. Puede optar por un spray de aplicación, que es más limpio y cómodo.

Sólo queda probar el circuito.

Los canales por los que llega el programa al PIC

En las figura siguientes, a modo de ejemplo, podemos ver una conexión típica entre el sistema de simuladores interconectables y la PC.



Volvamos, ahora, al IDE es decir, al ambiente de desarrollo integrado MPLAB. Una vez bien depurado el programa, éste genera un archivo de salida con extensión .hex.

Este archivo .hex se carga, primero, en una interface de usuario; la más conocida y usada es *IC-Prog®* o *EPIC-Win®*.

En el *IC-Prog®* se configuran el tipo de puerto de comunicación (en el caso de nuestro equipo de simuladores interconectables usamos el puerto serie en el programador serie diseñado y lo mantenemos *in circuit* para toda la tarea; pero no sucede así en el programador paralelo), el microcontrolador a utilizar y el oscilador (RC o cristal) que dará la frecuencia de reloj, entre otras posibilidades.

El *EPIC-Win®* se usa para el programador por puerto paralelo de la PC desarrollado en la placa del mismo nombre y puede ser confi-

Recuerde que el microcontrolador se quita de esta placa para colocarlo en la placa de prueba; es decir, no queda *in circuit*.

gurado de acuerdo con las necesidades del usuario -igual que el anterior-.

El archivo **.hex** contiene el código de operación (Código OP) que es enviado a la memoria de programa (Flash) dentro del PIC, por medio del cable paralelo o serie, y del circuito del programador que forma parte del equipo.

Como el PIC sólo entiende código binario, se necesita siempre de un software de aplicación que compile o traduzca a binario las instrucciones que se editan y que se conocen como código fuente (archivo con extensión **.asm**)

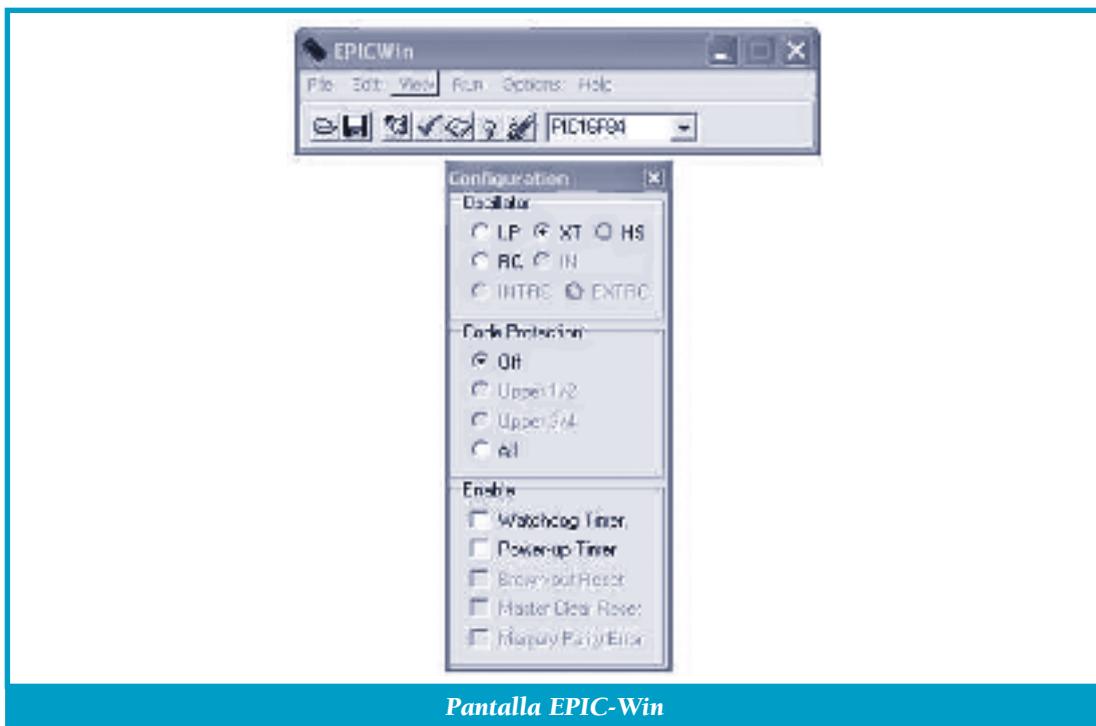
Una vez grabado el código de operación dentro de la memoria de programa, el PIC está listo para realizar la función encomendada.

Numerosos programas como los mencionados están disponibles gratuitamente en Internet.

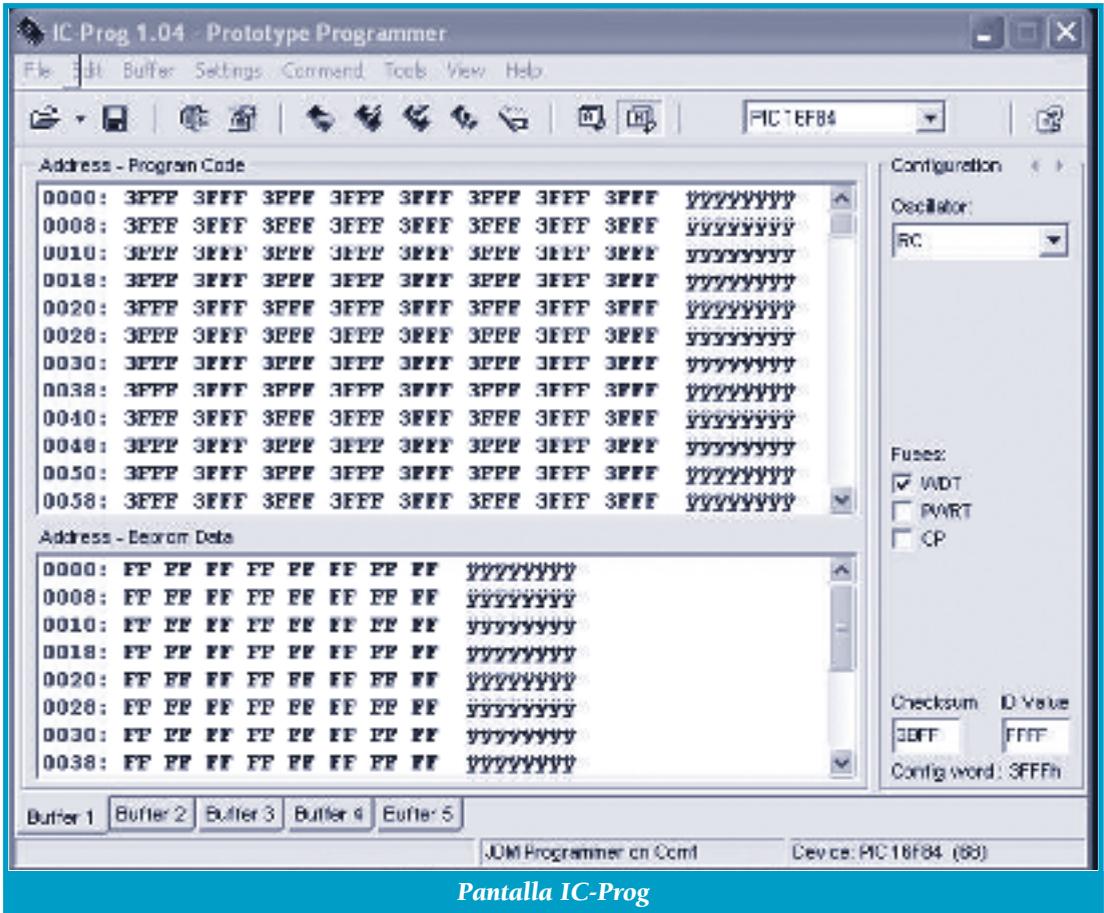
Antes de ejecutar el *EPIC-Win* o el *IC-Prog*, conectamos el programador al puerto paralelo o serie de la PC, respectivamente.

Luego, alimentamos desde la fuente y verificamos su presencia mediante el encendido de los led destinados a esa tarea. El led verde debe estar encendido y el rojo puede estar encendido o apagado, según lo hecho previamente con el puerto; el led rojo parpadea en la programación del microcontrolador.

Cualquiera que sea su modo de ejecución, se accede a la pantalla principal:



Pantalla EPIC-Win



Pantalla IC-Prog

a. El manejo de EPIC-Win®

En la pantalla que aparece, comenzamos desplegando el menú *File.. Open*, donde buscamos el archivo **.hex** a ser grabado.

A continuación, desplegando ahora el menú *Run*, realizamos el borrado del PIC, utilizando la opción *Erase*.

Antes de ser grabado el programa, configuramos el PIC usando la opción *View* en el desplegable *Configuration*, por medio de la cual elegimos:

- Tipo de oscilador (RC o XT).
- WDT -*Watch Dog Timer*- o no.
- PWT -*Power up Timer*- o no.
- Protección contra lectura o no (evita lecturas no deseadas del **.hex**).
- Tipo de microcontrolador a programar.

A continuación, también desde *Run*, utilizando la opción *Program*, realizamos el grabado del PIC con la configuración de trabajo indicada anteriormente, recordando que durante esta operación el led rojo parpadeará.

Si la opción contra lectura no está activada, utilizando la opción *Verify*, podemos corroborar si la grabación fue satisfactoria.

Una vez finalizada esta tarea, quitamos la alimentación al programador y, luego, retiramos el PIC para ser instalado en la placa de simulación y prueba, que forma parte del equipo.

Podemos repetir la secuencia descrita tantas veces como sea necesario, hasta conseguir que la aplicación funcione de la manera deseada.

El programa es modificado utilizando el MPLAB® y luego, sí, con el *EPIC-Win*, realizamos una nueva carga del programa usando el archivo con extensión **.hex** corregido.

b. El manejo de *IC-Prog*®

El uso de este programa es casi intuitivo.

Comenzamos con la ventana *File*. Mediante el desplegable *Open File* elegimos el archivo a grabar con extensión **.hex** -que, como sabemos, es creado por el MPLAB-

La ventana *Command* o las teclas de pantalla correspondientes permiten realizar una lectura del PIC, su borrado, la verificación de lo grabado o un chequeo para saber si está en blanco: *Read PIC*, *Erase PIC*, *Verify* o *Blank Check*.

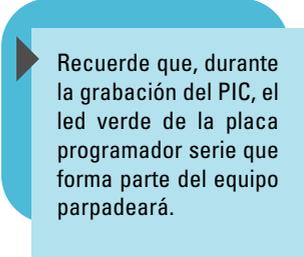
De la ventana *Setting* con el desplegable *Hardware* o con la tecla relacionada, podemos elegir el programador a usar y el tipo de puerto de comunicación (*Com Port*):

COM1, COM2, COM3 o COM4 para el caso serie; para el programador serie que forma parte del equipo, usamos el *JDM Programmer* en el COM1.

Desde la ventana principal o a través de la ventana *Setting* con el desplegable *Device*, podemos elegir el dispositivo sobre el cual realizar las distintas operaciones (por ejemplo, *Microchip*® PIC 16F84).

Antes de la grabación, se programan los *Fuses* para indicarle al PIC con qué oscilador va a trabajar: LP, XT, HS o RC; también, acti-

var o no el *Power on Timer* (PWRT); activar o no el *Watch Dog Timer* (WDT) y activar o no la opción *Code Protect* (CD) que permite o no, leer un PIC ya grabado.



Recuerde que, durante la grabación del PIC, el led verde de la placa programador serie que forma parte del equipo parpadeará.

Otras consideraciones útiles para el manejo del equipo

Para la interconexión del equipo de simuladores interconectables:

- Conectamos el transformador a la placa fuente, mediante los jumpers JP1.
- Conectamos, a través de JP2 y JP4, la placa fuente y la placa programador serie, respectivamente, al jumper JP3 de la placa principal.
- Colocamos el PIC a programar con el programador serie, en el zócalo corres-

pondiente en la placa principal¹⁰.

- Con la llave S2 (doble inversora) en la posición de *Programación*, conectamos la ficha DB9 en el puerto serie de la PC.
- Alimentamos al equipo de simuladores interconectables, energizando la placa fuente, enchufando a la red de 220 volt el transformador. De esta manera, el equipo estará listo para poder programar el PIC.
- Una vez programado el PIC con el *IC-Prog*, movemos la llave S2 pasándola al modo *Simulación*, sin quitar el PIC del zócalo. En este punto, debemos tener listas las placas adicionales (leds, llaves, display, etc.) conectadas y seleccionadas, con los respectivos jumpers, y las entradas y salidas individuales a utilizar en cada una de las placas, para percibir la simulación¹¹.
- En referencia a la placa de llaves, recordamos que, las S5, S6, S7 y S8 poseen, en paralelo, un conector JP9, JP10, JP11 y JP12 para poder dejar el pulsador cerrado en forma permanente, de manera simular el accionar de una llave, de un fin de carrera, etc.
- En la placa de displays hay cuatro jumpers que permiten elegir la cantidad

de displays a utilizar, teniendo en la parte alta del Puerto B (RB4 a RB7) el conversor BCD a 7 segmentos 74LS47 y en la parte baja del puerto B los transistores para hacer el multiplexado de los *displays*. Para el caso de necesitar solamente un display de 7 segmentos, está previsto en el conector (JP13) poder dejar el transistor asociado conectado en forma permanente y, así, liberar el pin de la parte alta del puerto B para otra aplicación -con otra de las placas conectadas, por ejemplo-.

- Consideramos el siguiente diagrama de conexiones del DB25 conectado en la placa principal:

Pin DB25	Señal
1	RA0
2	RA1
3	RA2
4	RA3
5	RA4
6	RA5
7	RB0
8	RB1
9	RB2
10	RB3
11	RB4
12	RB5
13	RB6
14	RB7
15	RC7
16	RC6
17	RC5
18	RC4
19	RC3
20	RC2
21	RC1
22	RC0
23	NC
24	VCC
25	GND

¹⁰Recordamos que sólo podemos ubicar un microcontrolador PIC por vez, para ser programado en el zócalo respectivo.

¹¹Para esto, recordamos que debemos seleccionar, mediante los jumpers (JPx en los esquemas eléctricos), cuáles son los leds, llaves y/o displays a utilizar, puesto que no siempre usaremos el puerto en forma completa. Dado que se pueden conectar hasta tres placas adicionales en forma simultánea (existen tres conectores DB25) y todas ellas están asociadas a los mismos pines de los microcontroladores, la selección individual con los jumpers permitirá el uso de varias placas simultáneamente, evitando la superposición y la posible destrucción del microcontrolador.

- En el caso de utilizar la placa paralelo para programar el PIC usando el *EPIC-Win*, comenzamos colocando el microcontrolador en el zócalo correspondiente en dicha placa. Luego, conectamos la ficha DB25 al puerto paralelo de la PC para finalizar energizando la placa conectando a 220 volt el transformador. Una vez cargado el programa en el PIC usando el *EPIC-Win*, procedemos en sentido inverso a la conexión, tal que desconectaremos primero el transformador de los 220

volt, luego la ficha DB25 de la PC y, por último, quitaremos el microcontrolador programado del zócalo para probarlo. Posteriormente, insertaremos ese PIC en la placa principal, en el zócalo correspondiente, colocando la llave S2 en modo *Simulación*, conectando a la misma placa las placas adicionales de hardware periféricos necesarios para llevar a cabo la simulación correspondiente. Esta placa principal está alimentada con la placa fuente.

4. EL EQUIPO EN EL AULA

El equipo de **simuladores interconectables basados en lógica digital** permite un amplio rango de aplicaciones, desde el manejo de una variable hasta un grupo de variables más complejas que posibilitan, por ejemplo, el encendido de un simple led o el desarrollo de un reloj controlado por cristal.

El sistema permite realizar experimentos con la totalidad de las 35 instrucciones del microcontrolador, sin necesidad de ningún componente adicional.

Los pasos en la generación de un proyecto

Un proyecto tecnológico nace de una necesidad detectada no satisfecha.

A partir de este problema, desplegamos distintas etapas de trabajo con nuestros alumnos.

1. Para resolver este problema, ¿debe utilizarse un microcontrolador o no es recomendable? La respuesta suele ser afirmativa, dado que los costos de estos CI son tan accesibles que no justifican dar una respuesta desarrollada con lógica discreta.
2. Seleccionamos la familia de microcontro-

ladores a utilizar. La elección de Microchip® es buena, debido a que todo su soporte de software es gratuito y el hardware es posible de ser concretado; por otra parte, la cantidad de integrantes que tiene su familia -es decir, la variedad de componentes distintos, cada uno con prestaciones que lo caracterizan- es amplia, por lo que es posible seleccionar el componente más ajustado al proyecto a realizar. Entonces, con la idea y con la familia, decidimos el hardware a implementar y el integrante de la familia más adecuado al proyecto a realizar.

3. Realizamos el programa, utilizando el MPLAB®. Como veíamos, este software permite la edición del programa, la búsqueda de errores del programa ya realizado y la simulación en la PC del programa. Cumplidas estas tres tareas, estamos en condiciones de probar nuestro programa en la aplicación a realizar.
4. Grabamos el PIC con el programa recién realizado.
5. Probamos el PIC ya grabado. Lo hacemos sobre algún circuito mediante el cual podamos evaluar su buen funcionamiento. Este circuito debe ser confeccionado de tal manera que sea útil no sólo para probar este proyecto sino para probar otros que surjan. Entonces, preferentemente, usamos un circuito probador universal que cumple con todas las condiciones requeridas. Instalamos el PIC ya

grabado en este probador que armamos. Lo evaluamos, modificando sus entradas y observando cómo varían sus salidas, en función de lo programado. El equipo de simuladores interconectables, además, dispone de conectores que nos permiten conectar entradas y salidas de acuerdo con el proyecto a realizar. La evaluación de funcionamiento puede no terminar en esta quinta etapa; si el monitoreo es negativo, volvemos atrás y regrabamos el PIC -esta vez, con el programa modificado-, para corregir el error observado en la prueba. Desarrollamos este proceso de control tantas veces como sea necesario, hasta que la quinta etapa sea cumplida satisfactoriamente.

6. Diseñamos el circuito impreso utilizando cualquier programa de CAD que permita el ruteo de pistas. Una vez obtenido este diseño, desarrollamos el circuito impreso definitivo. Éste sólo necesita de los componentes y del PIC ya grabado para ponerse en funcionamiento.
7. Evaluamos el producto ya terminado.

Las utilidades del sistema de simuladores

En un proyecto tecnológico, muchas veces nos enfrentamos con la necesidad de dar señales de aviso visuales que, por ejemplo, nos indiquen la presencia de un intruso o si se sobrepasó algún límite importante en un proceso químico.

El desarrollo de una respuesta a este problema implica, por ejemplo, encender y apagar un led por un tiempo determinado; algo simple y útil al mismo tiempo. La solución puede concretarse con nuestro recurso didáctico en forma sencilla, sin necesidad de sacar el microcontrolador, pues el equipo permite programar y probar lo realizado.

Para concretar una solución, utilizando el editor de textos que se encuentra dentro del programa MPLAB®, escribimos el código de la aplicación que deseamos realizar.

Por ejemplo, éste que presentamos abajo intenta encender durante un segundo un led, apagarlo y volverlo a encender; así, en forma indefinida:

```

pc      equ    02h
status equ    03h
ptoa   equ    05h    ;el puerto A está en la dirección 05 de la RAM
ptob   equ    06h    ;el puerto B está en la dirección 06 de la RAM
trisa  equ    85h    ;registro de configuración del puerto A
trisb  equ    86h    ;registro de configuración del puerto B
w      equ    00h    ;indica que el resultado se guarda en W

reset  org    0      ;el vector de reset es la dirección 00
       goto   inicio ;se salta al inicio del programa

```

```

                org    5           ;el programa empieza en la dirección de memoria 5

inicio         bsf     status,5   ;se ubica en el segundo banco de RAM
                movlw  0f0h       ;se carga el registro W con 0f
                movwf  trisa      ;se programan los pines del puerto A como salidas
                movlw  0ffh       ;se carga el registro W con ff
                movwf  trisb      ;se programan los pines del puerto B como entradas
                bcf     status,5   ;se ubica en el primer banco de memoria RAM

empiezo       bsf     porta,0
                call   retardo1_seg
                bcf     porta,0
                call   retardo1_seg
                goto   empieza

                retardo1_seg

subrutina de retardo de 1000 milisegundos
                movlw  D'100'     ;el registro loops contiene el número
                movwf  loops      ;de milisegundos del retardo
top2          movlw  D'110'     ;
                movwf  loops2     ;
top           nop
                nop
                nop
                nop
                nop
                nop
                decfsz loops2     ;pregunta si termino 1 ms
                goto   top
                decfsz loops     ;pregunta si termina el retardo
                goto   top2
                retlw  0

```

Utilizando la parte del programa que realiza la compilación -es decir, la corrección- de lo escrito por los alumnos, verificamos si lo realizado es correcto.

Realizamos el grabado del microcontrolador con la tarjeta que utiliza el puerto paralelo o

con la que utiliza el puerto serie. Si optamos por la segunda, tenemos la ventaja de no necesitar sacar el microcontrolador entre el grabado y la prueba. Si debemos corregir el programa, es necesario volver al MPLAB® de donde partimos con el trabajo.

También podemos plantear situaciones donde sea necesario utilizar el bus de expansión que tiene el equipo, como alternativa para tener mayores posibilidades de prueba; e, incluso, adicionar a este probador circuitos previamente diseñados en plaquetas independientes por los propios alumnos.

Continuando con el ejemplo anterior y tornándolo algo más complejo en cuanto al manejo de entradas y salidas del microcontrolador, podríamos proponer a nuestros alumnos el proyecto de chequear el estado de los sensores de presencia de las distintas zonas de un sistema de alarma, para que se encienda el led correspondiente a la zona donde se encuentra el intruso.

Como una primera solución alternativa, utilizando el bus de expansión, pretendemos

observar el estado de ciertos interruptores/sensores y encender el diodo emisor de luz que corresponda.

La utilización del bus de expansión es sencilla; en él se encuentran todas las señales disponibles, las que sólo hay que tomar.

El siguiente código mira el estado de los pulsadores y enciende el diodo led que le corresponde; es decir, si oprime el pulsador 1 se enciende el led 1, si oprime el pulsador 2 se enciende el led 2; y, así, siguiendo.

Para realizar este trabajo debemos usar el bus de expansión para sacar de la tarjeta a los pulsadores y usar los led que están en la tarjeta como indicadores de qué pulsador es oprimido:

```
;Este programa lee el estado de 4 interruptores y de acuerdo
;a ello enciende o no 4 led.
```

```
;En caso de que un número se escriba D'15': significa número decimal
;En caso de que el número se escriba B'00010101': significa número binario
;En caso de que un número se escriba 15H: significa número hexadecimal
;Si no se especifica nada, se supone numeración hexadecimal
```

```
; definición de registros
```

```
pc      equ    02h
status  equ    03h
ptoa    equ    05h    ;el puerto A está en la dirección 05 de la RAM
ptob    equ    06h    ;el puerto B está en la dirección 06 de la RAM
trisa   equ    85h    ;registro de configuración del puerto A
trisb   equ    86h    ;registro de configuración del puerto B
w       equ    00h    ;indica que el resultado se guarda en W

reset   org    0      ;el vector de reset es la dirección 00
        goto   inicio ;se salta al inicio del programa

        org    5      ;el programa empieza en la dirección de memoria 5
```

```

inicio    bsf      status,5 ;se ubica en el segundo banco de RAM
          movlw   0f0h  ;se carga el registro W con 0f
          movwf   trisa  ;se programan los pines del puerto A como salidas
          movlw   0ffh  ;se carga el registro W con ff
          movwf   trisb  ;se programan los pines del puerto B como entradas
          bcf     status,5 ;se ubica en el primer banco de memoria RAM

ciclo     movf    ptob,w  ;el valor de puerto B lo pasa al registro W
          xorlw   0ffh  ;con una operación xor se invierte el valor
                    ;del dato leído del puerto B
          movwf   ptoa  ;pasa el valor de W al puerto A
          goto   ciclo

          end

;=====
;      Fusibles de programación

;      Osc          XT
;      Watchdog     OFF
;      Code protect  OFF
;      Power-Up-Timer ON
;      Micro.       PIC16F84

;=====

```

Otro ejemplo práctico y útil para implementar podría ser que un motor paso a paso avance en un sentido o el otro, de acuerdo con la entrada de ciertos pulsadores.

Este problema requiere la utilización del bus de expansión del equipo que permite obtener la cantidad de líneas para que -en otra tarjeta no incluida en nuestro equipo- se monten los circuitos de potencia necesarios para mover los motores.

Consideremos un cuarto problema:

El personal de un invernadero debe recopilar los datos de temperatura y humedad a lo largo del día, durante un lapso de varias semanas para, así, tomar decisiones acerca de cómo acondicionar el ambiente.

Tomando como base de discusión esta problemática, comenzamos a pensar cómo sería el desarrollo del proyecto.

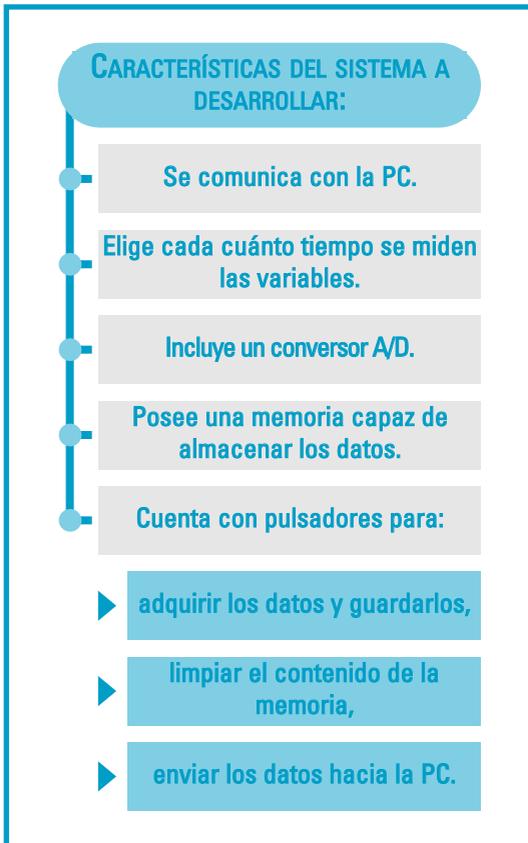
Supongamos que un grupo de alumnos sugiere utilizar una PC para realizar el trabajo. Esto es válido, por supuesto; pero, el costo del emprendimiento es elevado, además de constituir una limitación para el uso de otras aplicaciones en distintos sitios del invernadero. El uso de la PC como solución sólo es aceptable si, durante todo el día, pudiésemos distraer el equipo de otros usos, para realizar la tarea de medición. Y, aún siendo esto posible, se necesitaría de electrónica de interface entre la PC y la medición en sí.

Otro grupo de alumnos sugiere otra solu-

ción: Disponer de un sistema mínimo que se ocupe de tomar y guardar los datos requeridos para, luego de un determinado tiempo, trasladar los datos a un lugar donde se encuentra la PC y, allí, descargar los datos guardados y graficar los resultados.

Teniendo en cuenta las dos alternativas, evaluamos los rasgos a favor y en contra que cada una plantea. Teniendo en cuenta el costo del emprendimiento y la utilización de la computadora sólo en el lapso breve de la descarga de los datos-, optamos por la segunda.

A partir de esta decisión, comenzamos a pensar qué características generales debe tener el sistema a desarrollar.



Tiene la posibilidad de realizar una comunicación con la PC. Este requerimiento es indispensable dado que, luego de un determinado tiempo, se hace necesario descargar los datos y hacer una evaluación de éstos sobre la computadora.

Para cumplir con esta exigencia, contamos con diferentes opciones:

- usar el puerto paralelo de la PC,
- usar el puerto serie de la PC.

Decidimos optar por la segunda alternativa, debido a que:

- Contamos con programas de recepción hechos en *Windows*®, que se ocupan de recibir los datos. Por medio del programa *Hyperterminal* se puede intercambiar información con la PC utilizando el puerto serie.
- Desafectamos el puerto paralelo que, normalmente, se utiliza para imprimir.
- Como, habitualmente, la PC tiene más de un puerto serie, no entorpecemos su funcionamiento para otras actividades.

De todas maneras, el equipo de simuladores programables posee en placa separada un programador paralelo como otra opción posible.

Cuenta con la posibilidad de elegir cada cuánto tiempo realizar la medición de variables que van a ser evaluadas. Este rasgo le permite recolectar la mayor cantidad de datos en poco tiempo e, incluso, hacer una medición prolongada, recolectando datos

más espaciados en el tiempo.

Incluye un conversor A/D (analógico-digital).

Éste tiene por misión convertir la señal analógica en digital, para que pueda ser recibida por el microcontrolador. Las opciones también son dos:

- usar un conversor A/D con salida paralelo,
- usar un conversor A/D con salida serie.

Podemos elegir la segunda alternativa que no consume muchos pines del microcontrolador -éstos son necesarios para otras tareas del proyecto-.

Posee una memoria capaz de almacenar los datos convertidos de las variables. La capacidad de la me-

Capacidad es la cantidad de datos distintos que se pueden guardar.

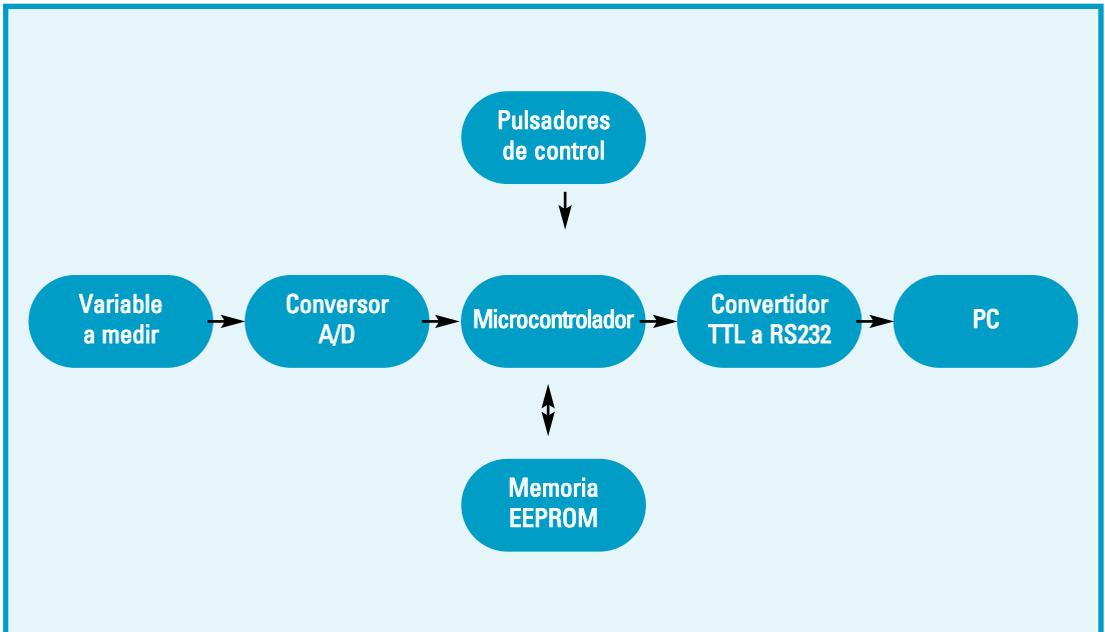
moria está relacionada con la cantidad de información a almacenar, en función del intervalo de tiempo requerido entre muestras.

Si se muestrean y almacenan los datos con mayor velocidad, la memoria se completa antes, lo que implica refrescar los datos a la PC en forma más rápida. Caso contrario, el refresco es más paulatino.

Cuenta con pulsadores para:

- adquirir los datos y guardarlos en la memoria,
- limpiar el contenido de la memoria con el objetivo de comenzar una nueva recolección de datos,
- enviar los datos guardados hacia la PC.

Realizamos, entonces, un diagrama de bloques de la situación planteada.



Para terminar con la descripción del sistema, incluimos las siguientes indicaciones:

- que se está realizando una transmisión hacia la PC,
- que se toma un nuevo dato,
- que se está efectuando el borrado de la memoria,
- que la memoria se encuentra llena.

La utilización del sistema es sencilla. Supongamos que optamos por evaluar la temperatura del invernadero una vez por minuto -60 veces por hora, 60 . 24 veces por día-. Esto nos permitiría realizar una gráfica de la variación de la temperatura a lo largo del día con una precisión aceptable, representando los datos obtenidos.

Una vez elegida la opción de cada cuánto deseamos medir, activamos el modo de medición por medio del cual todos los componentes del sistema trabajan para:

- medir,
- convertir a un valor digital,
- guardar en memoria y
- comenzar un nuevo proceso de medición.

El sistema avisa cuándo la memoria está llena. En ese momento, se conecta la PC a través del puerto serie y, utilizando otro pulsador, se realiza la transferencia de los datos guardados en la memoria del sistema, almacenándolos en la memoria de la PC para su análisis; para la transferencia se utiliza el programa *Hyperterminal*®, disponible en el paquete Windows.

El sistema queda, entonces, liberado y listo para otra tarea. Y, en la PC encontramos los datos, que pueden evaluarse por simple inspección o a través de una grafica -por ejemplo, para el caso de la temperatura como función de la hora del día-.

Otras posibilidades

Podemos integrar los **simuladores interconectables basados en lógica digital** en proyectos tecnológicos que exigen:

- manejar puertos o conexión de interruptores como entradas del microcontrolador y de led como salidas;
- visualizar el dato proveniente de un conteo o la información tomada de un muestreo -por ejemplo, de temperatura, presión, humedad, nivel, etc.- o del estado de máquinas, utilizando displays de 7 segmentos; aquí suele ser necesaria la conversión analógica-digital y digital-analógica de las distintas señales que se manejan como entrada y/o salida en el microcontrolador, para poder procesarlas dentro del microcontrolador o en el exterior;
- contar con líneas de entrada/salida suficientes en un dispositivo; por ejemplo, el multiplexado de teclados y la visualización en displays;
- visualizar un mensaje relacionado con el estado de una máquina, a controlar con instrucciones indicativas, con valores de instrumentos de medida, en los que un display de 7 segmentos no sería suficiente ya que debe mostrar caracteres

alfanuméricos; esta situación se relaciona con la necesidad de utilizar módulos de cristal líquido o LCD;

- utilizar un microcontrolador como principal elemento para guiar la posición de un motor paso a paso o una máquina de características especiales con bases digitales;
-

- variar la velocidad de motores de corriente continua con la técnica de modulación por ancho de pulso.

5. LA PUESTA EN PRÁCTICA

Esta parte final de nuestro módulo de capacitación contiene un cuadernillo para la evaluación del recurso didáctico que le presentamos y, de las experiencias didácticas y contenidos propuestos a partir de él:

Esta evaluación tiene dos finalidades:

- Brindarle a usted, como docente que utiliza este material, la oportunidad de documentar el seguimiento de las actividades que realice con sus alumnos, a partir de nuestras propuestas y, en función de esta memoria de acciones, propiciar una reflexión acerca de los cambios, mejoras o enriquecimiento de su propia tarea de enseñanza.
- Obtener de su parte, como usuario de este material, información sobre todos los aspectos en torno a los cuales gira la propuesta.

Para este relevamiento de información, usted encontrará, a continuación, una serie de cuestionarios organizados básicamente en tablas o matrices para completar. Con los datos que usted exprese en ellos esperamos tener una realimentación que nos permita mejorar todos los componentes de la serie de publicaciones “Recursos didácticos” y enriquecerla con propuestas o documentación complementaria para aquellos docentes que planteen iniciativas, interro-

gantes o dificultades específicas con relación a la construcción del recurso didáctico, a las actividades de aula, a los contenidos científicos y tecnológicos, a la metodología de enseñanza, a los procedimientos incluidos, a la información sobre materiales y a otros aspectos.

Dada la importancia que esta información de retorno tiene para nuestro trabajo de seguimiento, mejora y actualización, le agradecemos que nos remita el cuadernillo con todas las observaciones, comentarios o sugerencias adicionales que nos quiera hacer llegar. Para ello puede remitirnos una copia, a través de correo postal, a

Área de Monitoreo y Evaluación –CeNET–
Oficina 112
Saavedra 789. C1229ACE.
Ciudad Autónoma de Buenos Aires.
República Argentina.

O, si lo prefiere, solicitarnos el archivo electrónico de las páginas que siguen a evcenet@inet.edu.ar, enviándonos la versión digitalizada de sus respuestas a través del mismo correo electrónico.

Desde ya, muchas gracias.

Identificación del material:

Las dimensiones que se consideran para la evaluación del módulo de capacitación y del recurso didáctico son:

- | | |
|--|--|
| 1. Nivel educativo | 5. Documentación |
| 2. Contenidos científicos y tecnológicos | 6. Otras características del recurso didáctico |
| 3. Componentes didácticos | 7. Otras características del material teórico |
| 4. Recurso didáctico | 8. Propuestas o nuevas ideas |

1. Nivel educativo en el que trabajó el material:

Nivel educativo	EGB 2	EGB 3	Polimodal (*)			Escuela técnica (*)						Trayecto técnico- profesional (*)	Formación profesional (*)	Otra (*)	
			1	2	3	1	2	3	4	5	6				
Nivel en el que usted lo utilizó															

Asignatura/espacio curricular en el que usted lo utilizó:.....

(*) Por favor, indique la modalidad, la orientación, la especialidad, etc.

2. Contenidos científicos y tecnológicos trabajados:

.....

.....

.....

.....

.....

.....

.....





3. Componentes didácticos:

3.1. Testimonios (situaciones problemáticas) presentados en el material

	Sí	No	Otro ¹
a. ¿Le resultaron motivadores para iniciar las actividades propuestas?			
b. ¿Le facilitaron el desarrollo de contenidos curriculares que usted tenía previstos?			
c. A su criterio, ¿están vinculados con el recurso didáctico que se le propone desarrollar?			
d. ¿Le facilitan la organización de situaciones didácticas para el trabajo de los contenidos científicos y tecnológicos propuestos?			
e. El nivel de las situaciones problemáticas que se plantean, ¿es el adecuado al nivel educativo para el que está previsto?			
f. En caso negativo, ¿permiten adecuaciones para ser trabajados en el nivel educativo de sus alumnos o en otro nivel educativo?			
g. Los testimonios iniciales, ¿permiten generar diferentes soluciones (soluciones tecnológicas o didácticas)?			

En caso que su respuesta sea negativa (en cualquier ítem), le pedimos que nos indique por qué (señale el número del ítem a que corresponde su comentario).....

.....

.....

.....

.....

.....

.....

Otro (indique el ítem al que corresponde el comentario):

.....

.....

.....

.....

.....

¹ Utilice esta opción para indicar que agregará comentarios al final de este sector de la matriz.

3.2. Estrategias

A partir de la utilización de las propuestas de trabajo en el aula contenidas en el material y del recurso didáctico con el que se asocian, le solicitamos que nos indique (tomando como referencia su forma de trabajo anterior a disponer del material), cómo resolvió las actividades consignadas en la tabla siguiente:

3.2.1. Contextualización de la estrategia didáctica Con respecto a su forma habitual de trabajo, usted logró:	Mejor	Igual	No aplicado ²	Incorporado ³
a. Determinar las capacidades, habilidades, conocimientos previos necesarios para iniciar las actividades propuestas.				
b. Organizar, asociar, relacionar los conocimientos científicos y tecnológicos para resolver un problema tecnológico.				
c. Recortar (identificar) los contenidos científicos y tecnológicos a trabajar con sus alumnos para el desarrollo de un sistema/producto tecnológico como el propuesto por el material.				
d. Vincular estos conocimientos con los saberes previos de los alumnos.				
e. Establecer la secuencia adecuada de los contenidos científicos y tecnológicos, y de los procedimientos para generar una solución tecnológica (la propuesta por el material u otra diferente).				
f. Organizar una experiencia didáctica integrando conocimientos científicos y tecnológicos, metodología de resolución de problemas y procedimientos propios del trabajo tecnológico.				
g. Otras (que haya incorporado o hecho mejor con el recurso).				

² No aplicado: No lo hizo antes ni ahora con este recurso didáctico.

³ Incorporado: Integró la estrategia a sus clases a partir de la utilización del recurso didáctico propuesto.



3.2.2. Desarrollo de la estrategia didáctica	Mejor	Igual	No aplicado	Incorporado
Con respecto a su forma habitual de trabajo, usted logró:				
h. Encuadrar la tarea a partir de la formulación de uno (o varios) problemas.				
i. Explicitar consignas de trabajo que plantean una situación problemática.				
j. Organizar las actividades de aprendizaje atendiendo a las etapas propias de la resolución de problemas.				
k. Utilizar técnicas de trabajo grupal.				
l. Promover el trabajo colaborativo y cooperativo.				
m. Otras (que haya incorporado o hecho mejor con el recurso).				

3.2.3. Aspectos cognitivos (proceso de aprendizaje de sus alumnos)	Mejor	Igual	No aplicado	Incorporado
Con respecto a su forma habitual de trabajo, usted logró:				
n. Estimular a sus alumnos en la búsqueda de información e investigación en torno al problema eje del material.				
o. Promover la consulta a variadas fuentes de información.				
p. Rescatar, incorporar los aportes del grupo para identificar aspectos o variables críticas del problema.				
q. Evaluar los conflictos cognitivos propios del proceso de aprendizaje.				
r. Detectar, evaluar, la comprensión asociativa.				
s. Promover la reflexión sobre las actividades realizadas y las estrategias utilizadas en cada parte del proceso.				
t. Otras (que haya incorporado o hecho mejor con el recurso).				

4.1.2. ¿Realizó todo el proceso de construcción del recurso didáctico con sus alumnos? (Conteste este apartado en caso de que haya construido un equipo igual al propuesto. En caso contrario, pase al apartado 5 “Documentación”)

Sí	No
<input type="checkbox"/>	<input type="checkbox"/>

4.1.3. En caso de que su respuesta sea afirmativa, le pedimos que nos indique:

	Sí	No
a. ¿Pudo seguir sin dificultades los procedimientos indicados en el “Manual de construcción”?	<input type="checkbox"/>	<input type="checkbox"/>
b. La secuencia indicada, ¿fue la adecuada para la construcción?	<input type="checkbox"/>	<input type="checkbox"/>
c. El grado de complejidad, ¿fue el apropiado para el nivel educativo a que se dirige el recurso?	<input type="checkbox"/>	<input type="checkbox"/>
d. Los contenidos científicos asociados, ¿son pertinentes para el desarrollo del recurso propuesto?	<input type="checkbox"/>	<input type="checkbox"/>
e. Los contenidos tecnológicos asociados, ¿son pertinentes para el desarrollo del recurso propuesto?	<input type="checkbox"/>	<input type="checkbox"/>
f. Con sus alumnos, ¿construyó el recurso didáctico siguiendo el proceso y la metodología de resolución de problemas?	<input type="checkbox"/>	<input type="checkbox"/>
g. ¿Siguió todos los procedimientos propuestos para la construcción pero incorporó sus propios contenidos científicos y tecnológicos?	<input type="checkbox"/>	<input type="checkbox"/>
h. Por el contrario, ¿hizo adaptaciones en los procedimientos de construcción pero mantuvo los mismos contenidos?	<input type="checkbox"/>	<input type="checkbox"/>
i. ¿Realizó la construcción siguiendo las actividades de aula propuestas en el material?	<input type="checkbox"/>	<input type="checkbox"/>
j. ¿Diseñó sus propias experiencias en función de su grupo de alumnos?	<input type="checkbox"/>	<input type="checkbox"/>

Sí	No
<input type="checkbox"/>	<input type="checkbox"/>

¿Completó todas las etapas del proceso de construcción propuesta?

En caso negativo, indíquenos a qué fase llegó:

a. <input type="checkbox"/> Planificación.	b. <input type="checkbox"/> Diseño en dos dimensiones.
c. <input type="checkbox"/> Construcción, armado.	d. <input type="checkbox"/> Ensayo y control.
e. <input type="checkbox"/> Superación de dificultades (evaluación del funcionamiento, siguiendo las indicaciones y la lista de control que brinda el material).	
f. <input type="checkbox"/> Construcción de otro equipo que se adapta más a sus necesidades curriculares (Si marcó esta alternativa, lo invitamos a responder, directamente, el apartado 4.1.5.).	

4.1.4. Complete este ítem sólo si realizó el proceso de construcción del equipo siguiendo los procedimientos indicados en el Manual. Si no fue así, lo invitamos a responder el apartado 4.1.5.

Acerca de los materiales, herramientas e instrumentos:

	Si	No
a. La especificación de los materiales para la construcción, ¿fue suficiente para conseguirlos?		
b. ¿Utilizó los mismos materiales (en calidad y tipificación) indicados en la documentación?		
c. ¿Reemplazó materiales, instrumentos, componentes, piezas, etc., sin alterar el resultado final previsto en el material?		
d. La especificación de las herramientas a utilizar, ¿le resultó adecuada?		
e. La cantidad de herramientas indicadas, ¿fue la necesaria?		
f. Los instrumentos, ¿estuvieron bien especificados?		
g. El tipo y cantidad de instrumentos, ¿fueron los adecuados para armar el recurso didáctico?		

4.1.5. En caso de que usted haya construido un recurso didáctico diferente al propuesto por el material de capacitación, le pedimos que nos indique si la razón fue:

<p>a. <input type="checkbox"/> El propuesto no se ajustaba a sus necesidades curriculares.</p>	<p>b. <input type="checkbox"/> No pudo conseguir los materiales o instrumentos indicados.</p>
<p>c. <input type="checkbox"/> No pudo interpretar el manual de construcción.</p>	<p>d. <input type="checkbox"/> Otra (Por favor, especifíquela).</p> <p>.....</p> <p>.....</p> <p>.....</p> <p>.....</p> <p>.....</p> <p>.....</p> <p>.....</p>



4.1.6. ¿Qué características específicas destacaría en este recurso didáctico diferente al propuesto por el material, que sus alumnos han construido. (Marque todas las opciones que considere necesarias):



a. Se ajusta mejor a los contenidos curriculares que necesita trabajar.

b. Es más económico.

c. Permite su reutilización (mediante el desarme y armado, en función de necesidades didácticas).

d. Es más adaptable (a diversos usos).

e. Otra (Por favor, especifique):

.....

.....

.....

.....

f. Descripción del recurso didáctico construido:

.....

.....

.....

.....

.....

g. Indique las principales diferencias con el equipo propuesto (estructurales, funcionales, didácticas):

.....

.....

.....

.....

.....

4.2.2. Ya sea que haya desarrollado el recurso didáctico con sus alumnos según las especificaciones del material, ya sea que haya construido otro diferente o que haya utilizado un equipo ya construido, en relación con las actividades que usted venía realizando, la utilización del recurso didáctico propuesto por el material le permitió (seleccione la opción que coincida con sus experiencias):

Con respecto a su forma habitual de trabajo, este recurso didáctico le permitió a usted, como docente:	Mejor	Igual	No aplicable ⁴	Otro ⁵
a. Integrar contenidos científicos y tecnológicos en la solución de situaciones problemáticas de carácter tecnológico.				
b. Diseñar situaciones de enseñanza y de aprendizaje centradas en la resolución de problemas tecnológicos.				
c. Planificar y promover en sus alumnos la organización del trabajo (planificación y secuenciación de tareas), según el proceso tecnológico.				
d. Favorecer la identificación de aspectos o variables críticas de una situación problemática.				
e. Organizar las actividades de manera que facilite la toma de decisiones por parte de los alumnos (determinación y selección de alternativas, opciones de diseño, materiales, etc.).				
f. Organizar la actividad de sus alumnos en función de soluciones diversas a los problemas planteados.				
g. Agregue otras que usted considere haber logrado de una mejor manera con este recurso didáctico				

⁴NA: No aplicable; es una actividad que no realizó antes ni ahora.

⁵Otro: Recuerde utilizar esta opción para indicar que agregará comentarios al final de este sector de la tabla.

Con respecto a su forma habitual de trabajo, este recurso le permitió a los alumnos (habilidades intelectuales):	Mejor	Igual	No aplicable	Otro
Capacidad de planificar				
h. Identificar variables o aspectos fundamentales de un problema tecnológico.				
i. Organizar su trabajo en etapas (identificar y seguir la secuencia de operaciones de un proceso).				
j. Ejecutar las actividades en los plazos o etapas previstas.				
k. Seleccionar materiales, herramientas y piezas, de acuerdo con las necesidades del diseño.				
l. Anticipar y resolver dificultades que podrían surgir en el proceso.				
m. Prever puntos críticos de todo el proceso.				
n. Agregue otras que considere que sus alumnos alcanzaron mejor con este recurso didáctico				
.....				
.....				
.....				
.....				
.....				
.....				
.....				
.....				
.....				
.....				
.....				
.....				
.....				
.....				
.....				
.....				



Capacidad de aplicar y transferir	Mejor	Igual	No aplicable	Otro
s. Interrelacionar los datos, técnicas y procedimientos en el diseño de la solución.				
t. Utilizar técnicas de representación adecuadas al equipo que se construye o en el ya construido que se utiliza.				
u. Integrar los conocimientos científicos y tecnológicos en los momentos pertinentes para el diseño de la solución.				
v. Relacionar, ensamblar componentes en la secuencia adecuada.				
w. Utilizar de manera correcta la simbología y los lenguajes propios de la tecnología (representación gráfica, simbólica, etc.).				
x. Transferir conocimientos científicos y tecnológicos en otras actividades similares.				
y. Agregue otras que considere que sus alumnos alcanzaron mejor con este recurso didáctico				
.....				
.....				
.....				
.....				
.....				
.....				
.....				
.....				
.....				
.....				

Otro (Por favor, exprese aquí los comentarios que tenga, identificando el ítem con la letra que corresponda):

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....





5. Documentación (Material teórico, manual de procedimientos y propuestas didácticas):

5.1. ¿Cómo calificaría los aportes del material recibido (encuadre y desarrollo teórico, y experiencias propuestas para el aula)?

	MV ⁶	V	PV
a. Por su potencialidad didáctica (sugerencias, propuestas de trabajo en el aula, papel motivador, etc.).			
b. Para sus necesidades curriculares (desarrollo de los contenidos y experiencias previstas en su planificación).			
c. Para organizar, planificar, concretar experiencias didácticas relacionadas con problemas de Educación Tecnológica.			
d. Para renovar, actualizar, ampliar (subraye el que se ajusta más a su experiencia) los contenidos que desarrolla en su área/ disciplina.			
e. Para trabajar conocimientos científicos y tecnológicos de manera asociada a un problema tecnológico.			
f. Para organizar experiencias de aprendizaje en torno a la utilización de recursos didácticos.			
g. Para utilizar un recurso didáctico en el marco de experiencias didácticas organizadas en función de la resolución de problemas.			
h. Para integrar mejor contenidos científicos y tecnológicos en la solución de problemas de carácter tecnológico.			
i. Para estimular la generación creativa de otros recursos didácticos.			

Otras (Especifíquelas, por favor)

.....

.....

.....

.....

.....

.....

.....

.....

.....

⁶ Escala= MV: Muy valioso / V: Valioso / PV: Poco valioso

5.2. Manual de procedimientos para la construcción y el funcionamiento del recurso didáctico

En caso de que haya seguido los procedimientos contenidos en el Manual (ya sea para hacer un equipo igual o uno diferente al propuesto), le pedimos nos indique si:

	Sí	No	Otro
a. ¿Pudo seguir todos los procedimientos descritos, sin dificultad?			
b. ¿La secuencia descrita le resultó la adecuada?			
c. ¿La secuencia establecida le planteó alternativas según algún criterio (disponibilidad de los materiales, trabajo de contenidos específicos, etc.)?			
d. ¿La finalidad (para qué sirve) del equipo está indicada con claridad?			
e. ¿Se establecen cuáles son los contenidos (científicos o tecnológicos) que se asocian al equipo a construir?			
f. ¿Se determina la relación entre conocimientos implicados, procedimientos a seguir, materiales a utilizar y experiencias posibles de realizar?			
g. ¿Considera que la relación anterior es pertinente (es la que corresponde) para la construcción que se propone?			
h. ¿La descripción de los procedimientos le facilitaron la organización de las experiencias de trabajo con sus alumnos?			
i. ¿Pudo seguir las indicaciones para la puesta en funcionamiento?			
j. ¿Todas las indicaciones para el uso son claras?			

Por favor, fundamente sus respuestas negativas o agregue los comentarios que crea pertinentes (identifique el ítem a que se refiere):

.....

.....

Otro (identifique con la letra que corresponda el ítem sobre el que hace observaciones)

.....

.....

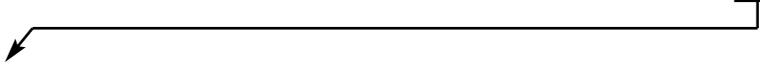
.....



6.3. Didácticas (Por favor, complete tanto si construyó el equipo como si utilizó uno ya construido)



	Sí	No
a. Congruencia. Tiene relación con los testimonios de realidad incluidos en el módulo de capacitación.		
b. Pertinencia. Los componentes, bloques funcionales y sistemas son adecuados para el trabajo con los contenidos curriculares de la educación técnico-profesional.		
c. Integración. Posibilita el tratamiento asociado de los conocimientos científicos y tecnológicos propuestos en el material.		
d. Escalabilidad. Es posible utilizarlo con proyectos o problemas con diferentes niveles de complejidad.		
e. Complejidad creciente. Las soluciones alcanzadas para una parte del problema, sirven de base para las siguientes o permite que, agregando componentes, sea utilizado como solución a problemas más complejos.		
f. Adaptabilidad. Permite su adaptación a soluciones diversas en torno a las problemáticas planteadas.		



Si su respuesta es negativa en alguna de ellas, indique por qué, identificándola con la letra correspondiente:

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

7. Otras características del material teórico:

¿Cómo calificaría el diseño del módulo escrito (desarrollo de contenidos científicos y tecnológicos, y propuestas de experiencias didácticas)?

	MB ⁷	B	R	M
a. Formato gráfico del material (distribución del contenido, márgenes, distribución de texto e imágenes, inserción de gráficos, diseño gráfico global, etc.).				
b. Lenguaje utilizado (claridad, adecuación al destinatario).				
c. Organización (secuencia entre cada parte).				
d. Adecuación al destinatario (evidencia que se toma en cuenta que es un material para ser trabajado en un ámbito escolar).				
e. Pertinencia de los conocimientos científicos con las problemáticas planteadas.				
f. Pertinencia de los conocimientos tecnológicos con las problemáticas planteadas.				
g. Vinculación (pertinencia) del recurso didáctico que propone con las situaciones didácticas planteadas.				
h. Congruencia (vinculación) de los contenidos propuestos con el recurso didáctico.				
i. Aporte metodológico para enriquecer sus estrategias didácticas.				
j. Aporte teórico (en general) para su trabajo docente.				
k. Valor motivador para el trabajo con sus alumnos.				
l. Valor orientador para generar sus propios recursos didácticos.				
m. Concepción innovadora para el trabajo didáctico en la educación técnico-profesional.				

Si marcó la opción “Malo”, le pedimos que nos explique por qué:

.....

.....

.....

⁷ Escala= MB: Muy bueno / B: Bueno / R: Regular / M: Malo



8. Propuestas o nuevas ideas:

Tanto para los autores de este material, como para el CeNET como institución responsable de su elaboración y distribución, una de las finalidades más importantes es suscitar en los educadores nuevas ideas, aplicaciones o propuestas creativas a partir de la lectura o el trabajo con el módulo.

En función de ello, le solicitamos que nos indique:

Si a partir del módulo (contenido teórico y recurso didáctico) usted, en su calidad de (marque todas las opciones que correspondan):

a. <input type="checkbox"/> docente a cargo de un grupo de alumnos	b. <input type="checkbox"/> directivo
c. <input type="checkbox"/> responsable de la asignatura:	d. <input type="checkbox"/> lector del material
e. <input type="checkbox"/> otro (especifique):	

ha generado nuevas ideas o propuestas:

Respecto de los contenidos (independientemente del recurso didáctico):

	Sí	No
a. Organización de su asignatura.	<input type="checkbox"/>	<input type="checkbox"/>
b. Contenidos científicos y tecnológicos (formas de asociarlos, ampliarlos, desarrollarlos, etc.)	<input type="checkbox"/>	<input type="checkbox"/>
c. Planificación de las experiencias didácticas.	<input type="checkbox"/>	<input type="checkbox"/>
d. Trabajo con resolución de problemas.	<input type="checkbox"/>	<input type="checkbox"/>

En relación con el recurso didáctico. Le pedimos que nos relate (libremente) las nuevas ideas o propuestas que el trabajo con este material le ha suscitado:



A series of horizontal dotted lines providing space for the user to write their response to the question above.

Títulos en preparación de la serie “**Desarrollo de contenidos**”.

- Colección: **Tecnología química en industrias de procesos**
 - El aire como materia prima
 - El azufre como materia prima
 - Los minerales como materia prima –bauxita y minerales de hierro

- Colección: **Construcciones**
 - Construcción de edificios. Cómo enseñarla a través de la resolución de problemas
 - Construcciones en hormigón armado: tecnología, diseño estructural y dimensionamiento

- Colección: **Telecomunicaciones**
 - Técnicas de transmisión banda base aplicadas a redes LAN y WAN
 - Cálculo de enlaces alámbricos

- Colección: **Materiales**
 - Fundamentos y ensayos en materiales metálicos

- Colección: **Tecnología en herramientas**
 - Historial de las herramientas de corte
 - Diseño y fabricación de herramientas de corte

- Colección: **Electricidad, electrónica y sistemas de control**
 - Instalaciones eléctricas
 - Familia TTL (Lógica transistor-transistor)
 - Familia lógica CMOS



MINISTERIO *de*
EDUCACIÓN
CIENCIA y TECNOLOGÍA
PRESIDENCIA *de la* NACIÓN



Argentina

ineti
*Instituto Nacional de
Educación Tecnológica*