



ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

***FACULTAD DE INGENIERÍA EN ELECTRICIDAD Y
COMPUTACIÓN***

***“CONSTRUCCIÓN DE UN MARCADOR
ELECTRÓNICO TOTALMENTE PROGRAMABLE
UTILIZANDO PICS Y APLICADO EN EL DEPORTE
DE BALONCESTO”***

Informe Técnico

Previo a la obtención del título de

***INGENIERO EN ELECTRICIDAD
ESPECIALIZACIÓN ELECTRÓNICA***

Presentado por

PABLO GERARDO SOLÓRZANO BORJA

***GUAYAQUIL – ECUADOR
2004***

AGRADECIMIENTO

A Dios, a la vida como dice la canción, de Mercedes Sosa, que me ha dado tanto.

A mi esposa Fanny, a mis hijos, Ely y Pablito, por su amor y ternura.

A Rosita, mi madre, por su infinito amor, por ser acicate y refugio espiritual en mis horas de desaliento.

A mis hermanos, Javier y Linda, por su comprensión y respeto.

A mi padre, por ser ejemplo de vida, él definió a la electrónica como la ciencia del silicio, el cobre y la cerámica, según sus palabras... la mejor y mayor profesión del mundo.

Al Ing. Freddy Cevallos Barberán, dilecto amigo y mentor, el ateo más cristiano que conozco.

Al Ing. Pedro Vargas Gordillo, gentil hombre y amigo, Director de este informe, por su paciencia y comprensión.

A los Ingenieros Hugo Villavicencio y Miguel Yapur, Miembros del Tribunal de Honor.

Al Dr. Moisés Tacle Galárraga, Rector de la ESPOL, por creer y confiar.

Al Ing. Carlos Monsalve Arteaga, Decano de la Facultad de Ingeniería en Electricidad y Computación; y, al señor Fiscal, Ing. Hernán Gutiérrez Vera, quienes me honran con su amistad.

A quienes me han apoyado, por ser la fuente en donde he recargado mis baterías, a mis detractores y leales adversarios, pues gracias a ellos y su especial dedicación se convirtieron en el mayor estímulo y reto para tratar de superarme.

A la Ingeniería Electrónica por darme las mejores lecciones de reciedumbre y humildad.

Gracias a Dios y a la vida que me ha dado tanto. El aire que respiro, el amor, la ternura, la pasión y mi gran Némesis... la electrónica del cual soy su más ferviente admirador, aprendiz y eterno enamorado.

DEDICATORIA

A la memoria del gran ausente... mi padre,
Ing. Jorge Bolívar Solórzano Constantine.

TRIBUNAL

ING. HERNAN GUTIERREZ VERA
SUB DECANO DE LA FIEC

ING. PEDRO VARGAS GORDILLO
DIRECTOR DEL INFORME TÉCNICO

ING. MIGUEL YAPUR AUAD
MIEMBRO DEL TRIBUNAL

ING. HUGO VILLAVICENCIO VILLAVICENCIO
MIEMBRO DEL TRIBUNAL

DECLARACIÓN EXPRESA

“La responsabilidad por los hechos, ideas y doctrinas expuestos en esta tesis, me corresponden exclusivamente; y, el patrimonio intelectual de la misma, a la ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL”.

PABLO GERARDO SOLÓRZANO BORJA

PREFACIO

Desde la invención de los primeros rectificadores de Selenio en Inglaterra, en las postrimerías del siglo XVIV, para luego pasar al descubrimiento de las primeras válvulas termoiónicas y la creación en 1956 del transistor a cargo de los laureados con el premio Nóbel, John Bardeen, Walter H. Brattain y William B. Schokley, y los desarrollos, en la década del 60 del siglo pasado, de los primeros circuitos integrados, llevados a cabo en Norteamérica a partir de una oblea de silicio, han permitido un desarrollo tecnológico durante los últimos 50 años, sin parangón para toda la humanidad, efectivamente, nunca antes en periodo igual de la historia se ha alcanzado un puesto tan relevante.

Los avances posteriores en la creación de ordenadores, con poderosos microprocesadores nos han permitido dar pasos gigantescos, en el área de telecomunicaciones, informática y medicina.

El desarrollo de la tecnología satelital nos permite, hoy en día a través de la televisión y de la telefonía celular, comunicarnos casi de forma instantánea con los lugares más remotos del globo terráqueo.

El desarrollo de tomógrafos y la resonancia magnética nuclear, así como los últimos avances en ingeniería genética, la decodificación del genoma humano y la creación de nanomáquinas que fusionadas a proteínas de síntesis permitirán, crear en la próxima década los primeros ordenadores pensantes o inteligentes, que junto con la mecatrónica y la robótica, harán posible la erradicación de enfermedades incurables como el cáncer, la diabetes y el temible VIH, prolongando y mejorando la calidad de vida, y quiera Dios así, la de todos los seres humanos.

Esta es la razón por la cual me decidí a seguir en el apasionante mundo de la electrónica, y a realizar, este trabajo en un área tan fascinante como la de los microcontroladores programables, presentes en casi todos los aparatos que utilizamos en nuestros hogares o sitios de trabajo.

Para concluir, quisiera hacer más las frases de mi padre el Ing. Jorge Solórzano Constantine, cuando un día, hace ya 25 años le manifesté que quería estudiar ingeniería en electrónica, el exclamó: *“Tu profesión es silicio, cobre y cerámica, has escogido bien hijo, la mejor profesión. Y un día no muy lejano, la profesión del mundo”*.

Termino parafraseando a José de Ingenieros, diciendo: “Cuanta razón tenía mi padre”.

ÍNDICE GENERAL

	Pág.
AGRADECIMIENTO	II
DEDICATORIA	III
TRIBUNAL	IV
DECLARACIÓN EXPRESA	V
PREFACIO	VI
ÍNDICE GENERAL	VII
INTRODUCCIÓN	1
1. CAPÍTULO I: LA SOLUCIÓN ESTÁ EN UN CHIP.....	2
1.1 Qué es un Microcontrolador	2
1.1.1 Diferencia entre Microprocesador y Microcontrolador	4
1.2 Arquitectura Interna	5
1.3 El Procesador	6

1.4	Memoria del Programa.....	7
1.5	Memoria de Datos	10
1.6	Líneas E/S para los Controladores Periféricos.....	11
1.7	Recursos Auxiliares	11
1.8	Programación de Microcontroladores	12
1.9	Instrumental de Trabajo	13
2.	CAPÍTULO II	17
2.1	La Elección	17
2.2	La Familia de los PIC.....	19
2.3	Gama Enana: PIC 12C(F)XXX de 8 patitas	19
2.4	Gama Baja o Básica PIC 16CSX con instrucciones de 12 bits...	20
2.5	Gama media PIC 16CSX con instrucciones de 14 bits.....	20
2.6	Gama Alta PIC 17XXX con instrucciones de 16 bits	21
2.7	Descripción General y Aplicaciones.....	22
3.	CAPÍTULO III:MANEJANDO EL REPERTORIO DE INSTRUCCIONES	25
3.1	Repertorio RISC.....	25
3.2	Tipos de formato	26
3.3	Operaciones Orientadas a Manejar Registros de Tamaño	27
3.4	Operaciones Orientadas a Manejar bits.....	27
3.5	Operaciones que manejan un valor inmediato a Literal.....	28

3.6	Operaciones Incondicionales de Control de Flujo del Programa	28
3.7	Operaciones de Salto Incondicional.....	28
3.8	Precisiones sobre Nomenclatura y Símbolos	29
3.9	Instrucciones que Manejan Registros.....	30
3.10	Instrucciones que Manejan bits.....	30
3.11	Instrucciones de Brinco (Skip).....	30
3.12	Instrucciones que Maneja Operadores Inmediatos	30
3.13	Instrucciones de Control y Especiales.....	31
4.	CAPÍTULO IV: ARQUITECTURA, DIAGRAMA DE CONEXIONES Y REPERTORIO DE INSTRUCCIONES DE LA VERSÁTIL FAMILIA PIC16F87X	33
4.1	Introducción	33
4.2	Procesador RISC con Arquitectura Harvard.....	34
4.3	Organización de la Memoria de Programa.....	35
4.4	Organización de la Memoria de Datos RAM.....	35
4.5	Control de la Memoria de Datos	37
4.6	Diagrama de Conexionado.....	37
4.7	Repertorio de Instrucciones.....	37
5.	CAPÍTULO V: DESCRIPCIONES DEL FUNCIONAMIENTO Y LA FABRICACIÓN DE UN JUEGO DE MARCADORES ELECTRÓNICOS	

TOTALMENTE PROGRAMABLES	39
5.1 Diseño del Prototipo del Marcador utilizando el PIC 16f784.....	39
5.2 Breve Descripción del Algoritmo de Funcionamiento del Prototipo	40
5.3 Diseño y Construcción de un Marcador Electrónico	51
5.4 Diseño y Cálculos de Tarjeta de Displays Implementada con 156 Leds.....	51
5.5 Diseño y Cálculo de Displays (Lumenes, Candelas, Esterio Radian) Voltaje y Corriente de Operación de un Led	57
5.6 Cálculos y Diseño de la Fuente de Alimentación del Equipo.....	59
5.7 Cálculo de la Corriente Requerida en el Primario, el Número de Espiras del Secundario y el Número de Espiras del Primario.....	60
5.8 Cálculo de la Corriente del Primario.....	61
5.9 Cálculo de la Sección Transversal de los Conductores Requeridos tanto como en el Devanado Primario, como en el Devanado Secundario del Transformador.....	62
5.10 Cálculo de la Masa del Conductor Requerido para el Transformador	63
5.11 Construcción de la Estructura y Montaje del Marcador.....	64
5.12 Algoritmo del Programa de los Marcadores	66
CONCLUSIONES Y RECOMENDACIONES.....	76
ANEXOS: FOTOS, DIBUJOS, GRÁFICOS Y TABLAS	77

BIBLIOGRAFÍA

109

INTRODUCCIÓN

El propósito del siguiente Informe Técnico es dar a conocer de forma detallada el proceso de fabricación de un marcador electrónico utilizado en el baloncesto, sus ventajas, la versatilidad de utilizar pics, y los costos relativamente bajos al utilizar este tipo de arquitectura.

De los capítulos 1 al 4 me ocupo en dar la información y el acervo teórico requerido para el manejo de los pics, sus usos, su arquitectura, su fácil manejo, su alta performance, su bajo costo de implementación, y el uso práctico y facilidades del lenguaje assembler para el desarrollo de marcadores electrónicos.

En el capítulo 5 ilustro todo el proceso completo del marcador electrónico, desde el diseño y construcción de las tarjetas electrónicas del controlador hasta las técnicas propias, implementadas en el desarrollo y fabricación del circuito impreso, además se incluyen los cálculos para el diseño y fabricación del transformador utilizado en la fuente de poder y se explica el por qué de la utilización de leds para la implementación de los displays. Además, se indica la fabricación del molde y de la mascarilla, así como la estructura requerida por el marcador.

En el anexo se incluye una descripción fotográfica completa y detallada en la fabricación de cada uno de los componentes y piezas que integran el marcador.

En las conclusiones y recomendaciones explicamos las grandes ventajas desde el punto de vista electrónico, técnico y económico de la fabricación de este tipo de equipo.

CAPÍTULO I

LA SOLUCION ESTA EN UN CHIP

1.1 ¿QUE ES UN MICROCONTROLADOR?

Es un circuito integrado programable que contiene todos los componentes de un computador. 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).

En microcontrolador es un computador **dedicado**. En su memoria sólo reside un programa destinado a gobernar una aplicación determinada; sus línea de entrada/salida soportan el conexionado de los sensores y actuadores del dispositivo a controlar y todos los recursos complementarios disponibles tiene como única finalidad atender sus requerimientos. Una vez programado y configurado el microcontrolador solamente sirve para gobernar la tarea asignada.

Un microcontrolador es un computador completo, aunque de limitadas prestaciones, que está contenido en el chip de un circuito integrado y se destina a gobernar una sola tarea. El número de productos que funciona en base a uno a varios microcontroladores aumenta de forma exponencial. No es aventurado pronosticar que en el siglo XXI habrá pocos elementos que carezcan de microcontrolador. En esta línea de prospección del

futuro, la empresa **Dataquest** calcula que en cada hogar norteamericano existirá un promedio de 240 microcontroladores para el año 2004.

La industria Informática acapara gran parte de los microcontroladores que se fabrican. Casi todos los periféricos del computador, desde el ratón o el teclado hasta la impresora, son regulados por el programa de un microcontrolador (**Véase Figura-1**).

Los electrodomésticos de línea blanca (lavadoras, hornos, lavavajillas, etc.) y de línea marrón (televisores, videos, aparatos musicales, etc.) incorporan numerosos microcontroladores. Igualmente, los sistemas de supervisión, vigilancia y alarma en los edificios utilizan estos chips para optimizar el rendimiento de ascensores, calefacción, aire acondicionado, alarmas de incendio, robo, etc.

Las comunicaciones y sus sistemas de transferencia de información utilizan profusamente.

Estos pequeños computadores incorporándolos en los grandes automatismos y en los modernos teléfonos.

La instrumentación y la electromedicina son dos campos idóneos para la implantación de estos circuitos integrados. Una importante industria consumidora de microcontroladores es la de automoción, que los aplica en el control de aspectos tan populares

como la climatización, la seguridad y los frenos ABS. (Véase figura-2)

Las comunicaciones y los productos de consumo general absorben más de la mitad de la producción de microcontroladores. El resto se distribuye entre el sector de la automoción, los computadores y la industria.

1.1.1 Diferencia entre microprocesador y microcontrolador

El microprocesador es un circuito integrado que contiene la Unidad Central de Proceso (UCP), también llamada procesador, de un computador. La UCP esta formada por la Unidad de Control, que interpreta las instrucciones, y el Camino de Datos, que las ejecuta.

Las patitas de un microprocesador sacan al exterior las líneas de sus buses de direcciones, datos y control, para permitir conectarle con la Memoria y los Módulos de E/S y configurar un computador implementado por varios circuitos integrados. Se dice que un microprocesador es un **sistema abierto** porque su configuración es variable de acuerdo con la aplicación a la que se destine. (Véase figura -3).

Un microprocesador es un **sistema abierto** con el que puede construirse un computador con las características que se desee, acoplándole los módulos necesarios.

Un microcontrolador es un **sistema cerrado** que contiene un computador completo y de prestaciones limitadas que no se pueden modificar.

Si sólo se dispusiese de un modelo de microcontrolador, este debería tener muy potenciados todos sus recursos para poderse adaptar a las exigencias de las diferentes

aplicaciones.

Esta potenciación supondría en muchos casos un despilfarro. En la práctica cada fabricante de microcontroladores oferta un elevado número de modelos diferentes, desde los más sencillos hasta los más poderosos. Es posible seleccionar la capacidad de las memorias, el número de líneas de E/S, la cantidad y potencia de los elementos auxiliares, la velocidad, de funcionamiento, etc. Por todo ello, un aspecto muy destacado del diseño es la selección del microcontrolador a utilizar. (**Véase figura - 4**).

1.2 ARQUITECTURA INTERNA

Un microcontrolador posee todos los componentes de un computador, pero con unas características fijas que no pueden alterarse.

Las partes principales de un microcontrolador son:

1. Procesador
2. Memoria no volátil para contener el programa
3. Memoria de lectura y escritura para guardar los datos
4. Líneas de E/S para los controladores de periféricos:
 - a) Comunicación paralelo
 - b) Comunicación serie
 - c) Diversas puertas de comunicación (busI²C, USB, etc.)

5. Recursos auxiliares:
- a) Circuito de reloj
 - b) Temporizadores
 - c) Perro Guardián (“watchdog”)
 - d) Conversores AD y DA
 - e) Comparadores analógicos
 - f) Protección ante fallos de la alimentación
 - g) Estado de reposo o de bajo consumo

A continuación se pasa revista a las características más representativas de cada uno de los componentes del microcontrolador.

1.3 EL PROCESADOR

La necesidad de conseguir elevados rendimientos en el procesamiento de las instrucciones ha desembocado en el empleo generalizado de procesadores de arquitectura **Harvard** frente a los tradicionales que seguían la arquitectura de **Von Neumann**. Esta última se caracterizaba porque la UCP se conectaba con una memoria única, donde coexistían datos e instrucciones, a través de un sistema de buses. (**Véase figura - 5**).

En la arquitectura **Harvard** son independientes la memoria de instrucciones y la memoria de datos y cada una dispone de su propio sistema de buses para el acceso. Esta

dualidad, además de propiciar el paralelismo, permite la adecuación del tamaño de las palabras y los buses a los requerimientos específicos de las instrucciones y de los datos. También la capacidad de cada memoria es diferente. (Véase figura-6).

El procesador de los modernos microcontroladores responde a la arquitectura **RISC** (computadores de Juego de Instrucciones Reducidos), que se identifican por poseer un repertorio de instrucciones máquina pequeño y simple, de forma que la mayor parte de las instrucciones se ejecuta en un ciclo de instrucción.

Otra aportación frecuente que aumenta el rendimiento del computador es el fomento del paralelismo implícito, que 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.

El alto rendimiento y elevada velocidad que alcanzan los modernos procesadores, como el que poseen los microcontroladores PIC, se debe a la conjunción de tres técnicas:
--

- | |
|---|
| <ul style="list-style-type: none">- Arquitectura Harvard- Arquitectura RISC- Segmentación |
|---|

1.4 MEMORIA DE PROGRAMA

El microcontrolador está diseñado para que en su memoria de programa se almacenen todas las instrucciones del programa de control. No hay posibilidad de utilizar memorias externas de ampliación.

Como el programa a ejecutar siempre es el mismo, debe estar grabado de forma permanente. Los tipos de memoria adecuados para soportar esta función admiten cinco versiones diferentes:

1ª ROM con máscara

En este tipo de memoria el programa se graba en el chip durante el proceso de su fabricación mediante el uso de “máscaras”. Los altos costos de diseño e instrumental sólo aconsejan usar este tipo de memoria cuando se precisan series muy grandes.

2ª EPROM

La grabación de esta memoria se realiza mediante un dispositivo físico gobernado desde un computador personal, que recibe el nombre de **grabador**. En la superficie de la cápsula del microcontrolador existe una ventana de cristal por la que se puede someter al chip de la memoria a rayos ultravioletas para producir su borrado y emplearla nuevamente. Es interesante la memoria EPROM en la fase de diseño y depuración de los programas, pero su costo unitario es elevado.

3ª OTP (programable una vez)

Este modelo de memoria solo se puede grabar una vez por parte del usuario, utilizando el mismo procedimiento que con la memoria EPROM. Posteriormente no se puede borrar. Su bajo precio y la sencillez de la grabación aconsejan este tipo de memoria para prototipos finales y series de producción cortas.

4ª EEPROM

La grabación es similar a las memorias OTP y EPROM, pero el borrado es mucho más sencillo al poderse efectuar de la misma forma que el grabado, o sea, eléctricamente. Sobre el mismo zócalo del grabador puede ser programado y borrado tantas veces como se quiera, lo cual la hace ideal en la enseñanza y en la creación de nuevos proyectos. El fabuloso PIC16C84 dispone de 1 K palabras de memoria EEPROM para contener instrucciones y también tiene algunos bytes de memoria de datos de este tipo para evitar que cuando se retira la alimentación se pierda información.

Aunque se garantiza 1.000.000 de ciclos de escritura/borrado en una EEPROM, todavía su tecnología de fabricación tiene obstáculos para alcanzar capacidades importantes y el tiempo de escritura de las mismas es relativamente grande y con elevado consumo de energía.

5ª.FLASH

Se trata de una memoria no volátil, de bajo consumo, que se puede escribir y borrar en circuito al igual que las EEPROM, pero suelen disponer de mayor capacidad que estas últimas. El borrado solo es posible con bloques completos y no se puede realizar sobre posiciones concretas.

Son muy recomendables en aplicaciones en las que sea necesario modificar el programa a lo largo de la vida del producto, como consecuencia del desgaste o cambios

de piezas, como sucede con los vehículos.

Por sus mejores prestaciones esta sustituyendo a la memoria EEPROM, para contener instrucciones. De esta forma *Microchip* comercializa dos microcontroladores prácticamente iguales que solo se diferencian en que la memoria de programa de uno de ellos es tipo EEPROM y la del otro tipo Flash. Se trata del PIC16C84 y el PIC16F84, respectivamente.

1.5 MEMORIA DE DATOS

Los datos que manejan los programas varían continuamente, y esto exige que la memoria que les contiene deba ser de lectura y escritura, por lo que la memoria RAM estática (SRAM) es la más adecuada, aunque sea volátil.

Hay microcontroladores que disponen como memoria de datos una de lectura y escritura no volátil, del tipo EEPROM. De esta forma, un corte en el suministro de la alimentación no ocasiona la pérdida de la información, que esta disponible al reiniciarse el programa. El PIC16C84, el PIC16F83 y el PIC16F84 disponen de 64 bytes de memoria EEPROM para contener datos.

La memoria tipo EEPROM y la tipo Flash pueden escribirse y borrarse eléctricamente. Sin necesidad de sacar el circuito integrado del zócalo del grabador pueden ser escritas y borradas numerosas veces.

1.6 LÍNEAS E/S PARA LOS CONTROLADORES DE PERIFÉRICOS

A excepción de dos patitas destinadas a recibir la alimentación, otras dos para el cristal de cuarzo, que regula la frecuencia de trabajo, y una más para provocar el Reset, las restantes patitas de un microcontrolador sirven para soportar su comunicación con los periféricos externos que controla.

Las líneas de E/S que se adaptan con los periféricos manejan información en paralelo y se agrupan en conjuntos de ocho, que reciben el nombre de **puertas**. Hay modelos con líneas que implementan puertas de comunicación para diversos protocolos, como el I²C, el USB, etc.

1.7 RECURSOS AUXILIARES

Según las aplicaciones a las que orienta el fabricante cada modelo de microcontrolador, incorpora una diversidad de complementos que refuerzan la potencia y la flexibilidad del dispositivo.

Entre los recursos más comunes se citan a los siguientes:

- a) *Circuito de reloj*, encargado de generar los impulsos que sincronizan el funcionamiento de todo el sistema.
- b) *Temporizadores*, orientados a controlar tiempos.
- c) *Perro guardián* (“*watchdog*”), destinado a provocar una reinicialización cuando el programa queda bloqueado.

- d) *Conversores AD y DA*, para poder recibir y enviar señales analógicas.
- e) *Comparadores analógicos*, para verificar el valor de una señal analógica.
- f) Sistema de protección ante fallos de la alimentación.
- g) *Estado de reposo*, en el que el sistema queda (congelado) y el consumo de energía se reduce al mínimo.

1.8 PROGRAMACIÓN DE MICROCONTROLADORES

La utilización de los lenguajes mas cercanos a la maquina (de bajo nivel) representan un considerable ahorro de código en la confección de programas, lo que es muy importante dada la estricta limitación de la capacidad de la memoria de instrucciones. Los programas bien realizados en lenguaje Ensamblador optimizan el tamaño de la memoria que ocupan y su ejecución es muy rápida.

Los lenguajes de alto nivel más empleados con microcontroladores son el C y el BASIC, de los que existen varias empresas que comercializan versiones de compiladores e intérpretes para diversas familias de microcontroladores. En el caso de los PIC es muy competitivo e interesante el compilador de C **PCM** de la empresa **CCS** y el **PBASIC** de **microLab Engineering**, ambos comercializados en USA por **Microsystems Engineering**.

Hay versiones de intérpretes de BASIC que permiten la ejecución del programa línea a línea y, en ocasiones, residen en la memoria del propio microcontrolador. Con ellos se

puede escribir una parte del código, ejecutarlo y comprobar el resultado antes de proseguir.

El lenguaje que utilizan los profesionales para la programación de los microcontroladores es el ensamblador, que es mas cercano a la maquina. También son frecuentes los programas en lenguaje C y en BASIC.

1.9 INSTRUMENTAL DE TRABAJO

Siempre que se diseña con circuitos integrados programables se precisan herramientas para la puesta a punto del hardware y del software.

Con referencia al software, además de los compiladores o interpretes de los lenguajes usados, es muy interesante disponer de simuladores software, que consisten en programas que simulan la ejecución de instrucciones representando el comportamiento interno del procesador y el estado de las líneas de E/S, Como se simula por software al procesador, el comportamiento no es idéntico aunque proporciona una aproximación aceptable, especialmente cuando no es esencial el trabajo en tiempo real.

Microchip pone libremente a disposición de sus usuarios, a través de Internet (<http://www.microchip.com>), ensambladores como el **MPASM** y simuladores como el **MPSIM**. **Microsystems Engineering** comercializa un simulador muy completo para el PIC16X84 y varios copiladores e interpretes de lenguaje C y BASIC

(<http://www.arrakis.es/~msyseng>).

Respecto a las herramientas hardware, una indispensable es el grabador, encargado de escribir el programa en la memoria del microcontrolador. Existen grabadores muy completos capaces de trabajar con muchos modelos de diferentes familias, pero su elevado precio los aleja de los usuarios personales. Para estos últimos existen bastantes versiones de sencillos grabadores, específicos para ciertos modelos de microcontroladores, que gobernados desde un computador personal se ofrecen por un precio ligeramente superior al de un libro. (**Véase figura -7**).

Los fabricantes de los microcontroladores disponen de numerosas herramientas apropiadas para sus productos. (**Véase figura -8**).

En la puesta a punto de proyectos industriales una potentísima herramienta es el **emulador en circuito**, que consiste en un complejo dispositivo físico que, controlado por un programa desde un computador personal, dispone de una “cabeza” con las mismas patitas que el microcontrolador que se trata de emular. La cabeza se introduce en el zócalo donde ira el microcontrolador con el programa definitivo en la placa de prototipo. El emulador en circuito hace funcionar al sistema como si hubiese un microcontrolador real, pero presentando en la pantalla del computador toda la información necesaria para conocer el desarrollo del programa y la actuación de lo periféricos. Los resultados que así se obtienen son idénticos a los del producto final,

puesto que la ejecución se realiza en tiempo real, a diferencia de los simuladores software. (**Véase figura -9**).

Finalmente, reciben el nombre de sistemas de desarrollo equipos que mediante una combinación de hardware y de software consiguen realizar la mayoría de las fases que exige un diseño basado en microcontrolador. Como la mayoría de las herramientas, también esta se gobierna por un programa desde un computador personal. Mantienen una buena comunicación y gestión de la edición de programas y su compilación o ensamblado. Tras la depuración del software y la obtención del código binario ejecutable disponen de un grabador que escribe el programa obtenido sobre la memoria de instrucciones, también puede verificar lo escrito en la memoria e incluso borrarlo, cuando es del tipo EEPROM o Flash.

Una vez grabado el microcontrolador sus patitas quedan conectadas a través de *jumper* con los periféricos que hay que controlar y se procede a la ejecución en tiempo real del programa. La inclusión de los periféricos clarifica el resultado final. Mediante el programa de comunicación con el PC se puede conocer en todo momento el estado de todos los elementos del computador. Por su sencillez, potencia y economía destacamos el sistema de desarrollo **MICRO'PIC TRAINER (Véase figura-10)** de **Microsystems Engineering**, que incorpora las siguientes prestaciones:

-Grabador de PIC.

- Hardware y software de adaptación a PC.
- Programa de comunicaciones.
- Tarjeta que incluye la mayoría de los periféricos típicos: Interruptores, leds, display de siete segmentos, potenciómetros para señales analógicas y pantalla LCD.
- Tutorial con fases de montaje y funcionamiento y ejercicios de experimentación y diseño de proyectos.

Para facilitar el montaje de los prototipos se comercializan diversas tarjetas de circuito impreso predefinido. En ellas existe una zona común donde se aloja el zócalo del microcontrolador y su circuitería auxiliar (reloj, pulsador de reset, regulación de voltaje, etc.), Hay otra zona libre para situar los periféricos propios de la aplicación, que pueden conectarse fácilmente con todas las líneas de E/S presentes en determinados puntos de la tarjeta. (**Véase figura-11**).

Mas adelante presentaremos la tarjeta para prototipos de **Micrsystems Engeneering** con la que se pueden desarrollar la mayoría de los proyectos y ejercicios que ofrecemos en este libro.

CAPÍTULO II

2.1 LA ELECCIÓN

Entre más de 50 fabricantes de microcontroladores que existen en el mundo es muy difícil seleccionar “el mejor”. En realidad no existe, porque en cada aplicación son sus características específicas las que determinan el más conveniente.

Por otra parte, los autores nos hemos equivocado a menudo y no somos una garantía de acierto. Personalmente, recuerdo la época, a principios de la década de los ochenta, cuando estaba convencido que las *Memorias de Burbujas Magnéticas* revolucionarían el mundo de la Microinformática con la misma importancia que lo hicieron antes los microprocesadores. En el día de hoy cada vez hay menos profesionales que saben de su existencia.

De todas formas, como el libro lo hacemos entre dos y mi coautor esta entusiasmado con los **PIC** y vive de ellos, igual esta vez tenemos más suerte.

Dejando a un lado las preferencias personales, lo cierto es que los **PIC** “ángel” y una gran aceptación en la comunidad de técnicos y aficionados que trabajan con microcontroladores.

¿Cuánto durará este éxito? Hasta que salga al mercado otro producto que le robe esa imagen.

¿Cuál es la razón de la excelente acogida? Porque hay detalles que nos vuelven locos a los profesionales de la Microelectrónica y Microinformática; citamos los más importantes a nuestro juicio.

- Sencillez de manejo
- Buena información
- Precio
- Buen promedio de parámetros: velocidad, consumo, tamaño, alimentación. Código compacto, etc.
- Herramientas de desarrollo fáciles y baratas
- Diseño rápido

Lo cierto es que a las puertas del siglo XXI y en su corta vida, los PIC ocupan las posiciones de cabeza en el ranking mundial, compitiendo codo a codo con gigantes como **Intel** y **Motorola**. En 1990 ocupaba el puesto vigésimo y actualmente son más de 100 millones de PIC los que vende **Microchip** cada año.

Dentro de los diferentes tipos de microcontroladores los hay que procesan datos de 4, 8,16 y 32 bits, sin embargo, el mas representativo y popular es el de 8 bits, al que pertenecen los PIC, porque resulta el más flexible y eficaz para la mayoría de los diseños típicos. (Véase **figura-12**).

2.2 LA FAMILIA DE LOS PIC

Una de las labores más importantes del ingeniero de diseño es la elección del modelo de microcontrolador que mejor satisfaga las necesidades del proyecto con el mínimo presupuesto.

En 1997 el fabricante de los PIC dispone de 52 versiones diferentes y cada año aumenta considerablemente su lista.

Microchip dispone de cuatro familias de microcontroladores de 8 bits para adaptarse a las necesidades de la mayoría de los clientes potenciales.

2.3 GAMA ENANA: PIC12C(F)XXX DE 8 PATITAS

Se trata de un grupo de PIC de reciente aparición que ha acaparado la atención del mercado.

Su principal característica es su reducido tamaño, al disponer todos sus componentes de 8 patitas. Se alimentan con un voltaje de corriente continua comprendido entre 2.5V y 5.5V, y consumen menos de 2mA cuando trabajan a 5 V y 4 MHz. El formato de sus instrucciones puede ser de 12 o de 14 bits y su repertorio es de 33 o 35 instrucciones, respectivamente.

En la (**Tabla- 1**) se presentan las principales características de los modelos de esta subfamilia, que el fabricante tiene la intención de potenciar en un futuro próximo. Los modelos 12C5xx pertenecen a la gama baja, siendo el tamaño de las instrucciones de 12 bits; mientras que los 12C6xx son de la gama media y sus instrucciones tienen 14bits. Los modelos 12F6xx poseen memoria Flash para el programa y EEPROM para los datos.

2.4 GAMA BAJA O BÁSICA: PIC16C5X Con Instrucciones De 12 BITS

Se trata de una serie de PIC de recursos limitados, pero con una de las mejores relaciones *coste/prestaciones*. Sus versiones están encapsuladas con 18 y 28 patitas y pueden alimentarse a partir de una tensión de 2,5V, lo que les hace ideales en las aplicaciones que funcionan con pilas. Tienen un repertorio de 33 instrucciones cuyo formato consta de 12 bits. No admiten ningún tipo de interrupción y la Pila solo dispone de dos niveles. (**Véase figura 13**).

2.5 GAMA MEDIA: PIC16XXX CON INSTRUCCIONES DE 14 BITS

Es la gama más variada y completa de los PIC. Abarca modelos con encapsulado desde 18 patitas hasta 68, cubriendo varias opciones que integran abundantes periféricos. Dentro de esta gama se halla el “fabuloso PIC16X64” y sus variantes.

El repertorio de instrucciones e de 35 de 14 bits cada una y compatible con el de la gama baja. Sus distintos modelos contienen todos los recursos que se precisan en las aplicaciones de los microcontroladores de 8 bits. También dispone de interrupciones y

una Pila de 8 niveles que permite el anidamiento de subrutinas.

La gama media puede clasificarse en las siguientes subfamilias:

- a) Gama media estándar (PIC16C55X);
- b) Gama media con comparador analógico (PIC16C62X/64X/66X);
- c) Gama media con módulos de captura (CCP), modulación de anchura de impulsos (PWM) y puerta serie (PIC16C6X);
- d) Gama media con CAD de 8 bits (PIC16C7X);
- e) Gama media con CAD de precisión (PIC14000);
- f) Gama media con memoria Flash y EEPROM (PIC16X8X) (a esta subfamilia es a la que se dedica este libro);
- g) Gama media con driver LCD (PIC16C92X)

Encuadrado en la gama media también se halla la versión PIC14C000, que soporta el diseño de controladores inteligentes para cargadores inteligentes para cargadores de baterías, pilas pequeñas, fuentes de alimentación ininterrumpibles y cualquier sistema de adquisición y procesamiento de señales que requiera gestión de la energía de alimentación. Los PIC14C000 admiten cualquier tecnología de las baterías como *Li Ion*, *NiMH*, *NiCd*, *Pb* y *Zinc*.

2.6 GAMA ALTA: PIC17XXX CON INSTRUCCIONES DE 16 BITS

Se alcanzan las 58 instrucciones de 16 bits en el repertorio y sus modelos disponen de un sistema de gestión de interrupciones vectorizadas muy potente. También incluyen variados controladores de periféricos, puertas de comunicación serie y paralelo con elementos externos un multiplicador hardware de gran velocidad.

Quizás la característica más destacable de los componentes de esta gama es su

arquitectura abierta, que consiste en la posibilidad de ampliación del microcontrolador con elementos externos. Para este fin, las patitas sacan al exterior las líneas de los buses de datos, direcciones y control, a las que se conectan memorias o controladores de periféricos. Esta filosofía de construcción del sistema es la que se empleaba en los microprocesadores y no suele ser una práctica habitual cuando se emplean microcontroladores. Las características más relevantes de los modelos de esta gama, que sólo se utilizan en aplicaciones muy especiales con grandes requerimientos. Con vista al siglo XXI, **Microchip** lanzará la gama “mejorada” PIC18CXXX.

2.7 DESCRIPCIÓN GENERAL Y APLICACIONES

La diversidad de los modelos de PIC tienen una finalidad: *poder seleccionar el más adecuado para cada aplicación. Por este motivo conviene tener un conocimiento fundamental y actualizado de cada gama.*

En cuanto a la gama enana, la familia PIC12C5X, se trata de un conjunto de microcontroladores CMOS encapsulados con 8 patitas. Tienen un precio muy bajo y resultan altamente competitivos.

En la gama media hay una subfamilia, PIC16C62X, con recursos muy especiales, pero manteniendo el bajo precio y el consumo reducido, que es muy recomendable en aplicaciones relativas a la gestión de cargadores de baterías, control de sensores remotos con bajo consumo, etc.

Con diez versiones diferentes, la subfamilia PIC16C62X, de la gama media se caracteriza por la incorporación de numerosos recursos (varios temporizadores, módulos de captura/comparación, puerta serie SPI e I²C, puerta paralela PSP, etc.). Mejorando estos modelos, la serie PIC16C7X es una muestra de las posibilidades del procesador RISC funcionando a 20 MHz, con un ciclo de instrucción de 200ns. Los componentes de esta serie son muy apreciados en los sistemas de alarma y seguridad, así como en el control remoto de sensores en automoción.

La memoria EEPROM, de igual forma que la Flash, puede grabarse y borrarse eléctricamente, sin someterla a rayos ultravioleta como sucede en el borrado de las EEPROM con ventana. Además, se puede realizar la grabación y el borrado en serie, lo cual posibilita la grabación de un programa, su depuración y su borrado tantas veces como se desee y manteniendo insertado el PIC en el zócalo de la aplicación. La memoria EEPROM admite hasta 1.000.000 de ciclos de escritura/borrado y almacena la información durante más de 40 años. La memoria Flash tiene un valor típico de 1.000 ciclos de escritura/borrado, pero aventaja técnicamente en varios aspectos a la EEPROM.

El hecho de poder emplear el mismo dispositivo cuando se realizan numerosos diseños o cuando se prueban muchos programas en la fase de aprendizaje resulta muy práctico y económico. Aplicaciones típicas de estos microcontroladores son el control de puertas de garaje, instrumentación, inmovilizadores de vehículos, tarjetas codificadas, pequeños sensores, etc.

La grabación de los PIC16X8X en el propio circuito les hace recomendables para el almacenamiento de datos de calibración y para la modificación del programa al variar las condiciones del entorno; por ejemplo, al cambiar una pieza de un automóvil, el programa de control del motor se modifica para su adecuación.

A las puertas del siglo XXI. **Microchip** presentó en el mercado mundial los nuevos microcontroladores RISC FLASH. Se trata de la serie PIC16F87X(3,4,5,6 y 7). Son unos modelos mucho más potentes que el PIC16F84X, pero que mantiene sus ventajas en su grabación y borrado eléctrico.

Es una serie que encaja dentro de la gama media y que alcanza una memoria de código de hasta 8k palabras de 14 bits en memoria FLASH. Soporta hasta 368 x 8 bytes de

RAM y 256 x 8 bytes de memoria de datos EEPROM. Pueden programarse sólo con + 5 VDC y existe acceso a lectura y escritura de la memoria de programa.

Estos potentes microcontroladores disponen de hasta tres Timer o contadores, dos módulos de Captura, Comparación y PWM, un Convertidor Analógico/Digital de 10 bits, Canal de comunicaciones síncrono SSP con modo 12C y SPI, canal USART/SCI, puerto paralelo esclavo asíncrono de 8 bits y detección de falta de alimentación (BOR).

Los lectores que completen con éxito el primer escalón en el manejo del PIC16F84 deberán tener muy en cuenta esta potentísima familia de PIC con memoria FLASH para superar sus diseños.

CAPÍTULO III

MANEJANDO EL REPERTORIO DE INSTRUCCIONES

3.1 REPERTORIO RISC

Todos los modelos de microcontroladores PIC responden a la arquitectura RISC, que significa “Computador de Juego de Instrucciones Reducido”. No solo implica que el número de instrucciones máquina que es capaz de interpretar y ejecutar el procesador es pequeño, como sucede en los PIC16X8X, que consta de 35, sino también que posee las siguientes características:

1a Las instrucciones son simples y rápidas

La falta de complejidad en la operación que realizan las instrucciones de los procesadores RISC permite que sean ejecutadas, mayoritariamente, en un solo ciclo de instrucción. Los PIC tardan en ejecutar todas las instrucciones un ciclo, excepto las de salto, que tardan el doble.

2a Las instrucciones son ortogonales

Apenas tienen restricciones en el uso de operandos. Cualquier instrucción puede usar cualquier operando

3a La longitud de las instrucciones y los datos es constante

Todas las instrucciones tienen la misma longitud, 14 bits en los PIC16X8X, y todos los datos también, un byte. La arquitectura Harvard del procesador aísla la memoria de instrucciones de la de datos, pudiendo tener diferente tamaño sus palabras.

3.2 TIPOS DE FORMATO

Las instrucciones de los PIC de la gama media, entre los que se encuentran los modelos PIC16X8X, tienen 14 bits de longitud. Dicho formato se divide en diferentes campos de bits, cada uno de los cuales referencia a operandos o elementos que maneja la instrucción en la operación que realiza en el procesador. Se describen dichos campos:

a) *Campo del código OP*

Los bits de este campo sirven para definir la operación que realiza la instrucción.

b) *Campo de los operandos fuente (f) y destino (d)*

Estos campos de bits definen los registros que actúan como operandos en la instrucción. Suelen referenciar la dirección que ocupan en la memoria de datos.

c) *Campo de operando inmediato o literal (k)*

Es un campo de bits que contiene el valor de un operando inmediato.

d) *Campo que referencia a un bit (b)*

Suele ser un campo de 3 bits que indica la posición de un bit concreto dentro de un registro de 8 bits.

e) *Campo de dirección del salto*

En las instrucciones de salto CALL y GOTO hay un campo de bits que contiene la dirección de la siguiente instrucción que hay que ejecutar. Dicho campo de bits se carga en el PC en las instrucciones de salto incondicional.

Para estudiar los diversos formatos que admiten las instrucciones del repertorio de los PIC16X8X se clasifican en cinco grandes grupos, atendiendo al tipo de operación que desarrollan.

1. Operaciones orientadas a manejar registros de tamaño byte.
2. Operaciones orientadas a manejar bits.
3. Operaciones que manejan valores inmediatos.
4. Operaciones incondicionales de control del flujo del programa.
5. Operaciones de salto condicional.

3.3 OPERACIONES ORIENTADAS A MANEJAR REGISTROS DE TAMAÑO BYTE

En formato de las instrucciones que corresponden a este grupo se divide en tres campos:

1. Campo del Código OP de 6 bits
2. Campo de la dirección del operando fuente (f) de 7 bits
3. Campo que define el operando destino (d) de 1 BIT

Las instrucciones de este grupo tiene como sintaxis: *nemónico f, d*. Cuando $d = 1$ el registro destino coincide con el fuente.

3.4 OPERACIONES ORIENTADAS A MANEJAR BITS

En formato al que responden las instrucciones de este grupo tiene tres campos y se muestra en la (Figura –15)

1. Campo del Código OP de 4 bits
2. Campo de la dirección del registro fuente (7bits)
3. Campo de la posición del bit en el registro (3bit)

3.5 OPERACIONES QUE MANEJAN UN VALOR INMEDIATO O LITERAL

El formato corresponde a este grupo de instrucciones solo tiene dos campos:

1. Campo del Código OP con 6 bits
2. Campo del valor inmediato (k) con 8 bits

3.6 OPERACIONES INCONDICIONALES DE CONTROL DEL FLUJO DEL PROGRAMA

Este tipo de instrucciones afectan al contenido del Contador de Programa (PC) y sirven para romper la secuencia ordenada de las instrucciones del programa. Las instrucciones de este grupo tienen un formato con solo dos campos:

1. Campo del Código OP de 3 bits.
2. Campo de la dirección del salto que se carga en el PIC de 11 bits.

3.7 OPERACIONES DE SALTO CONDICIONAL

Los PIC disponen de unas pocas instrucciones “ muy efectivas” que cuando se cumple una condición dan un brinco (*skip*). Llamamos brinco a un saltito muy pequeño, concretamente solo se salta una instrucción, la que hay detrás de la condicional. La condición es el estado de un bit de un registro o la puesta a cero de un registro tras un decremento o un incremento.

La instrucción *decfsz f,d* responde al grupo de instrucciones que manejan registros de un byte. En este caso, si $d = 1$, el destino es el propio registro fuente y lo que hace la instrucción es decrementar el contenido de f y dejar el resultado en f . Pero, además, se el nuevo valor de f es cero se produce un brinco. Si no es cero el flujo de control continuo con la siguiente instrucción.

3.8 PRECISIONES SOBRE NOMENCLATURA Y SÍMBOLOS

Hemos tomado la decisión de utilizar la nomenclatura y los símbolos que emplea MICROCHIP para su lenguaje Ensamblador MPASM. Recordamos los mas específicos.

SÍMBOLO	SIGNIFICADO
<i>Oxhh</i>	<i>Es la forma que se usa en el lenguaje MPASM para referenciar a los hexadecimales de dos dígitos (hh)</i>
<i>f</i>	<i>Representa la dirección en la memoria RAM de datos del registro fuente.</i>
	<i>Tiene un tamaño de 7 bits, con un direccionamiento de 128 posiciones</i>
	<i>Comprendidas entre la dirección 0x00 y la 0x7f</i>
<i>d</i>	<i>Es un bit que conforma el campo del formato de una instrucción que indica el registro destino. Si $d = 0$ es W, y si $d = 1$ es f.</i>
<i>k</i>	<i>Campo que contiene un valor inmediato, que puede ser un operando (8bits) o una dirección para el PIC (11bits).</i>
<i>x</i>	<i>Valor indeterminado de un bit. Puede ser 1 o 0.</i>
<i>()</i>	<i>Contenido.</i>
<i><></i>	<i>Campo de un bit de un registro. Por ejemplo: ESTADO<5>.</i>
<i>E</i>	<i>En el conjunto de Ejemplo: $d E (0,1)$</i>
<i>()</i>	<i>Opciones.</i>

3.9 INSTRUCCIONES QUE MANEJAN REGISTROS

Responden a la sintaxis *nemónico f, d*, siendo *f* y *d* los dos operandos fuente y destino que se hallan implementados por registros de 8 bits de la memoria de datos.

El registro *f* viene referenciado por la dirección de 7 bits que ocupa, mientras que el destino solo por 1, que si vale 0 es el W y si vale 1 es el fuente. En la (**Tabla - 2**) se muestra las instrucciones de este grupo con sus características mas interesantes.

3.10 INSTRUCCIONES QUE MANEJAN BITS

Solo hay dos instrucciones en este grupo, pero son muy flexibles. Una de ellas pone al (bsf) cualquier bit de un registro, mientras que la otra lo pone a 0 (bcf). (**Véase Tabla 3**)

En el formato de las instrucciones que manejan bits se representa con 3 bits la posición que ocupa en el registro el bit b. El registro se especifica mediante su dirección de 7 bits.

3.11 INSTRUCCIONES DE BRINCO (SKIP)

Sólo hay cuatro instrucciones de salto condicional en los PIC de la gama media. Dos de ellas testan un bit de un registro y según valga 1 o 0, brincan o no. Recuérdese que un brinco es un “saltito” pequeño, solo se salta la instrucción siguiente a la condicional. Las otras dos instrucciones incrementan o decrementan un registro y la posibilidad del brinco se efectúa se con es operación el valor del registro ha llegado a cero. Cuando estas instrucciones no brincan porque no se cumple la condición, tardan 1 ciclo de instrucción en ejecutarse. Caso que brinquen tarda el doble. (**Véase en la Tabla 4**)

3.12 INSTRUCCIONES QUE MANEJAN OPERANDOS INMEDIATOS

Se trata de media docena de instrucciones que realizan una operación con un valor inmediato de 8 bits que se proporciona dentro del formato de la instrucción, el cual solo tiene dos campos: el del Código OP (6 bits) y el del operando inmediato (8 bits). (**Véase**

Tabla 5)

3.13 INSTRUCCIONES DECONTROL Y ESPECIALES

En este grupo se incluyen las instrucciones que rompen la secuencia normal del programa porque alteran el contenido del PIC y también las instrucciones especiales.

La instrucción de salto incondicional *goto* carga en el PC la dirección de la nueva instrucción. La instrucción *call* de Llamada a Subrutina, antes de cargar el PC con la dirección de la instrucción a saltar, salva la dirección de partida guardando en la cima de la Pila la dirección de regreso del programa principal.

Para realizar un retorno de una subrutina se pueden emplear dos instrucciones. La mas habitual es *return*, que se limita a extraer de la cima de la Pila el valor que carga en el PC. Otra más compleja es *retlw k*, que, además de hacer lo mismo que *return*, carga en W el valor inmediato k que contiene. Es decir, devuelve un parámetro desde la subrutina.

Para el final de las interrupciones hay otra instrucción cuyo nemónico es *retfie*. La operatividad de esta instrucción consiste en cargar en el PC el contenido de la cima de la Pila y poner el bit GIE = 1, pues al comenzar la instrucción este bit se pone automáticamente a 0 para evitar que cuando se atiende una interrupción se produzca otra. GIE es el bit de permiso de todas las interrupciones.

En cuanto a las instrucciones especiales, se han incluido dos en este grupo: *clrwdt* y *sleep*.

La primera pone a 0 el contenido del Perro Guardián, es decir, lo refresca o lo reinicializa. El Perro Guardián si se desborda (pasa de 0xff a 0x00) provoca un Reset. La instrucción *clrwdt* hay que colocarla estratégicamente en ciertos puntos del programa para evitar la reinicialización.

La instrucción *sleep* introduce al procesador en un modo de funcionamiento que se llama de Reposo o de Bajo Consumo. Detiene al oscilador y el procesador queda congelado, no ejecutando instrucciones y manteniendo el mismo valor las Puertas de E/S. También pone los PD# =0 y TO# = 1 y borra el Perro Guardián y al Divisor de frecuencia. (Véase **Tabla- 6**)

CAPÍTULO IV

ARQUITECTURA, DIAGRAMA DE CONEXIONES Y

REPERTORIO DE INSTRUCCIONES DE LA

VERSÁTIL FAMILIA PIC 16F87X

4.1 INTRODUCCION

Bajo la denominación de PIC16F87X se hace referencia a una subfamilia de microcontroladores PIC de la gama media, que se identifica por tener como memorias de programa una de tipo FLASH y una serie de recursos semejante a los modelos más potentes, como por ejemplo los PIC16C73/4, teniendo estos últimos el inconveniente de que su memoria de programa es de tipo EPROM.

Dos de los cuatro modelos que componen esta subfamilia están encapsulados con 28 patitas (PIC16F873/6), mientras que los otros dos tiene 40 patitas (PIC16F874/7). Con la intención de seguir potenciando la línea con memoria FLASH, Microchip también comercializa los microcontroladores PICF16F62X, que con un precio “rompedor” mantiene el encapsulado de 18 patitas, aumentando considerablemente los recursos internos en comparación con el PIC16F84.

Como se mostró en el capítulo anterior en la tabla de la figura 5 las principales diferencias entre los PIC16F87X con 28 patitas y los de 40 se concentran especialmente en el número de líneas de E/S disponibles, pero a continuación se citan las tres diferencias más relevantes:

1. Los modelos encapsulados con 40 patitas disponen de 5 puertas (PA, PB, PC, PD y PE) de E/S con un total de 33 líneas para conectar a los periféricos exteriores. Los de 28 patitas sólo tienen 3 puertas (PA, PB y PC) con 22 líneas de E/S.
2. El convertor A/D en los PIC con 28 patitas tiene 5 canales de entrada, pero en los de 40 patitas tiene 8.
3. Sólo los encapsulados con 40 patitas integran una puerta paralela esclava.

4.2 PROCESADOR RISC CON ARQUITCTURA HARVARD

Esta arquitectura aplicada por *Microchip* en sus microcontroladores se caracteriza por la independencia entre la memoria de código y la de datos. Así, tanto la capacidad como el tamaño de los buses de cada memoria se adaptan estrictamente a las necesidades del diseño, facilitando el trabajo en paralelo de las dos memorias, lo que permite obtener altas cotas de rendimiento. La filosofía RISC se hace patente en el reducido número de instrucciones que forman su repertorio. Sólo consta de 35 instrucciones, que se ejecutan en un ciclo de instrucción, equivalente a cuatro períodos de reloj, excepto las de salto que necesitan dos ciclos. (Véase **Figura- 14**)

Una primera aproximación a la arquitectura interna de los PIC16F87X como se muestra en la **figura 15**. Debe tenerse en cuenta que la mayor diferencia entre los modelos de 40 patitas y los de 28 reside en el número de Puertas de E/S que en el primer caso asciende a 5 y en el segundo a 3.

Obsérvese en la **Figura 15** que la memoria de código está direccionada por el PIC (Contador de Programa) en conexión con la Pila de 8 niveles. La memoria de datos RAM contiene el Banco de Registros Específicos y el Banco de los Registros de Propósito General y transfiere información bidireccional por el bus de datos de 8 líneas que interconecta todos los elementos. Finalmente, el Camino de Datos está formado por una ALU de 8 bits que trabaja conjuntamente con el Registro de Trabajo W.

En el esquema de la **(Figura-16)** se presenta con más detalle la arquitectura correspondiente a los modelos PIC16F874/7 de 40 patitas. Los otros dos modelos de la subfamilia tienen idéntica arquitectura, pero carecen de las puertas D y E, así como de las funciones que éstas soportan.

4.3 ORGANIZACIÓN DE LA MEMORIA DE PROGRAMA

La memoria FLASH, en la que se graba el programa de aplicación en los PIC16F87X, puede tener una capacidad de 4 K u 8 K palabras de 14 bits cada una. Dicha memoria está dividida en páginas de 2 K palabras y está direccionada con el PC, que tiene un tamaño de 13 bits. La Pila, que tiene 8 niveles de profundidad, es transparente para el usuario, es decir, funciona automáticamente y no dispone de instrucciones para guardar o sacar de ella información. Con la instrucción CALL y con las interrupciones el valor del PC se salva en el nivel superior. Con las instrucciones RETURN, RERFIE y RETLW el valor contenido en el nivel superior de la Pila se carga en el PC. Al poseer la Pila solo 8 niveles le corresponde al programador preocuparse por los anidamientos en las subrutinas para no sobrepasar dicho valor. El vector de Reset ocupa la dirección 000h y el vector de Interrupción la 0004h, igual que en el PIC16F84, **(Figura 17)**

4.4 ORGANIZACIÓN DE LA MEMORIA DE DATOS RAM

La memoria de datos tiene posiciones y implementadas en RAM y otras en EEPROM. En la sección RAM, se alojan los registros operativos fundamentales en el funcionamiento del procesador y en el manejo de todos sus periféricos, además de registros que el programador puede usar para información de trabajo propia de la aplicación. Más adelante, se estudiará la memoria EEPROM para guardar datos de forma no volátil, a la que se considera como un dispositivo especial.

La RAM estática consta de 4 bancos con 128 bytes cada uno. En las posiciones iniciales de cada banco se ubican los Registros Específicos que gobiernan al procesador y sus recursos. Dos modelos de PIC16F87X tienen 192 bytes de RAM y otros dos 368 bytes. Los modelos de menor capacidad no tienen implementadas físicamente algunas

posiciones. **La tabla 7** presenta los cuatro bancos de la RAM, indicando en las primeras posiciones de cada uno los nombres de los registros que contienen.

En las Figuras de la **Tabla 7 y 8** las posiciones con trama oscura no están implementadas físicamente y siempre se leen como 0. los registros específicos marcados con la nota (1) no se hallan implementados en los PIC de 28 patitas.

Para seleccionar el banco al que se desea acceder en la RAM se emplean los bits 6 y 5 del *Registro de Estado*, denominados RP1 y RP0 respectivamente, según el código siguiente:

BANCO	RP1	RP0
0	0	0
1	0	1
2	1	0
3	1	1

Para manipular adecuadamente los elementos internos de los PIC hay que conocer la misión de los bits de los Registros Específicos.

Registro específicos para el control de la memoria de programa

Los 13 bits contenidos en el PC, que direccionan la memoria de código, están guardados en dos registros específicos. El registro PCL guarda los 8 bits de menos peso y se puede leer y escribir. Los bits <12:8> del PC se alojan en el registro PCH, que al no poder ser leídos ni escritos, se accede a él a través del PCLATH.

Las instrucciones de salto CALL y GOTO solo proporcionan 11 bits de la dirección a saltar. Esto limita el salto dentro de cada banco de 2K. Cuando se desea salir del banco actual hay que programar correctamente los bits PCLATH <4:3> que seleccionan al banco. Es labor del programador modificar el valor de dichos bits en las instrucciones CALL y GOTO. (Véase **Figura 18**)

4.5 CONTROL DE LA MEMORIA DE DATOS

Para direccionar la memoria RAM de datos estructurada en 4 bancos de 128 bytes cada uno existen dos modos diferentes:

<p>Direccionamiento indirecto</p> <p>Direccionamiento directo</p>

En el modo de direccionamiento directo, los bits RP1 y RP0 del *Registro de Estado* <6:5> se encargan de seleccionar el banco, mientras que la dirección dentro del banco la determinan 7 bits procedentes del código OP de la instrucción. Para el direccionamiento indirecto se usa el registro FSR en el que sus 7 bits de menos peso señalan la dirección, y el banco lo determina el bit de más peso de FSR concatenado con el bit IRP del *Registro de Estado* <7>. (Véase **Figura 19**)

4.6 DIAGRAMA DE CONEXIONADO

En la figura 20 se muestra el diagrama de distribución y asignación de las 28 ó 40 patitas de los encapsulados PDIP (doble helera de patitas de plástico) de los PIC16F87X. Los encapsulados con menos patitas carecen de las puertas D y E.

La **Figura 21** muestra algunas de las posibles opciones de encapsulado que utiliza *Microchip* para proteger a sus microcontroladores.

La asignación de funciones para las diferentes patitas es la siguiente:

4.7 REPERTORIO DE INSTRUCCIONES

Los mismos formatos, iguales modos de direccionamiento y las mismas 35 instrucciones que tenía PIC16F84X, por lo que no vamos a volver a repetirlos.

No obstante, en los nuevos PIC al contener más recursos, existen nuevos Registros Específicos de control cuyos bits se deberán escribir o leer para su gobierno. Dicho recursos y sus registros asociados serán el tema de los próximos capítulos.

En la **TABLA 9** se ofrece un cuadro resumido del repertorio de instrucciones.

CAPÍTULO V

DESCRIPCIÓN DEL FUNCIONAMIENTO Y LA FABRICACION DE UN JUEGO DE MARCADORES ELECTRÓNICOS TOTALMENTE PROGRAMABLES

Los marcadores existentes en el mercado permiten a través de 2 displays realizar un conteo decreciente, con paradas y congelamiento de tiempo según las necesidades del usuario, que están contempladas en los reglamentos del BALONCESTO.

El conteo se realiza a partir de los 24 segundos según las ultimas regulaciones de la “NBA”, normalmente poseen una botonera en donde se encuentra una tecla que se denomina TIEMPO que realiza la parada y congelamiento del dato y su reinicialización, además tienen una tecla para resetear y reinicializar el marcador en cualquier instante, y que este regrese al inicio a realizar el conteo, poseen además una tecla para hacer sonar la bocina en cualquier instante.

La gran desventaja es que constantemente están cambiando las reglas de juego y para hacerlo mas vistoso y rápido acortan los tiempos, y seria entonces necesario reprogramarlo.

Sin embargo, esto suele resultar muy oneroso económicamente hablando, esta es la razón por la cual me decidí a diseñar y fabricar un marcador programable, a fin de que el usuario fácilmente pueda cambiar el tiempo de conteo pero manteniendo el mismo algoritmo de funcionamiento.

5.1 DISEÑO DEL PROTOTIPO DEL MARCADOR UTILIZANDO EL PIC16F84

Para el diseño del prototipo utilizaremos el PIC16F84 de la siguiente manera:

Los pines que corresponden a RA0, y RA1 serán configurados como puertos de entrada y se les asignara respectivamente como nemónicos a las variables TIEMPO Y 24 SEG, (que es el valor del tiempo que actualmente la asociación mundial de basketball “NBA” impone en los partidos y torneos). El Pin RA2 se lo hará oscilar en una onda cuadrada cuyo periodo total es de 10 milisegundos es decir 5 milisegundos a nivel alto y 5 milisegundos a nivel bajo, este Pin hará encender y apagar al displays de decenas y unidades alternativamente, a través de un transistor NPN, 2N3904 al que se lo hace trabajar como switch, es decir en la región de corte y saturación del transistor.

De forma tal que al estar encendido el display de decenas durante este periodo de tiempo, se le suplementera el dato codificado a través de los pines RB0 hasta RB6 que previamente han sido configurados como salidas.

De tal igual forma se hará con el display de unidades, durante el periodo de corte del transistor.

El Pin RA3 servirá para el encendido automático de la luz y la bocina por un tiempo de 5 segundos cuando el marcador llegue a “00 segundos”, se adicionara también un botón para a través de Hardware encender la bocina en cualquier momento.

La descripción del circuito se muestra en la **(Fotografía 1)**

5.2 BREVE DESCRIPCIÓN DEL ALGORITMO DE FUNCIONAMIENTO DEL PROTOTIPO

El conteo se hará en secuencia decreciente desde “24” hasta “0” segundos. A través de la multiplexación y envió de los datos previamente codificados en una tabla que nos permita encender a los displays de 7 segmentos, no se habilitara ningún servicio de interrupción en el integrado, lo único que se hará es censar los niveles de las entradas TIEMPO Y 24 SEGUNDOS, que previamente han sido asignadas y configuradas como entradas en los pines RA0 y RA1 respectivamente.

El periodo de encendido y apagado se hará en 5 milisegundos respectivamente, y se repetirá en un lazo 100 veces de forma tal que el dato se muestre en un tiempo total de 1 segundo, a partir de ahí podrá ser decrementado o podrá ser nuevamente enviado a que realice el conteo desde el inicio, podrá pararse en cualquier tiempo, y quedará congelado este valor, y reiniciarse el conteo a partir de la parada, una vez que llegue a cero si se desea podrá reiniciarse el conteo.

```
1  PROGRAMA DE DECREMENTO DE TIEMPO INCLUIDO ADORNO, ESPERA
   Y VUELTA A 24SEG:
2  LIST P=1 6 F 8 4;
3  INICIO
4  W EQU 0X00
5  F EQU 0X01
6  Z EQU 0X02
7  TMR0_OPT EQU 0X01
8  PC EQU 0X02
9  STATUS EQU 0X03
10 INTCON EQU 0X0B
11 PORTA EQU 0X05
12 PORTB EQU 0X06; UNIDADES
13 UNI EQU 0X0C; DECENAS
14 DEC EQU 0X0D
15 TRISA EQU 0X05
16 TRISB EQU 0X06
17 CONT EQU 0X0E ; SE CARGARA 100 VECES PARA EL RETARDO DE 5MS
18 CONT1 EQU 0X0F
19 CONT2 EQU 0X10
20 DEC1 EQU 0X11
```

```
21  UNI1 EQU 0X12
22  DEC2 EQU 0X13
23  UNI2 EQU 0X14
24  ORG  0
25  GOTO INICIO
26  ORG  5
27  INICIO BSF STATUS.5
28  MOVLW 0X03
29  MOVWF TRISA
30  CLRF  PORTB
31  MOVLW 0X07
32  MOVWF TMR0_OPT
33  BCF   STATUS.5
34  CLRF  PORTA
35  MOVLW 0X06
36  MOVWF CONT1
37  MOVLW 0X02
38  MOVWF DEC
39  MOVLW 0X04
40  MOVWF UNI
41  LAZ MOVLW 0X00
42  MOVWF DEC2
43  MOVLW 0X01
44  MOVWF UNI2
45  MOVLW 0X50
46  MOVWF CONT
47  LAZ2 MOVF  DEC2,W
48  CALL  DSEG
```

```
49 BCF    PORTA,2
50 MOVWF PORTB
51 CALL   DELAY
52 CLRW
53 MOVF   UNI2,W
54 CALL   DSEG
55 BSF    PORTA,2
56 MOVWF PORTB
57 CALL   DELAY
58 DECFSZ CONT,F
59 GOTO   LAZ2
60 MOVLW 0X02
68 MOVWF DEC2
69 MOVLW 0X00
70 MOVWF UNI2
71 MOVLW 0X50
72 MOVWF CONT
73 LAZ4      MOVF   DEC2,W
74 CALL   DSEG
75 BCF    PORTA,2
76 MOVWF PORTB
77 CALL   DELAY
78 CLRW
79 MOVF   UNI2,W
80 CALL   DSEG
81 BSF    PORTA,2
82 MOVWF PORTB
83 CALL   DELAY
```

```
84  DECFSZ CONT,F
85  GOTO  LAZ2
86  DECFSZ CONT1,F  GOTO
87  LAZ
88  MOVLW 0xFF
89  MOVWF PORTB
90  BUCLE      BTFSS PORTA,0
91  GOTO BUCLE
92  REG        MOVLW 0X03
93  MOVWF CONT1
94  MOVLW 0X02
95  MOVWF DEC
96  MOVLW 0X04
97  MOVWF UNI
98  LAZO1      MOVLW 0X64
99  MOVWF CONT
100 LAZO2      MOVF  DEC,W
101 CALL  D7SEG
102 BCF   PORTA,2
103 MOVWF PORTB
104 CALL  DELAY
105 CLRW
106 MOVF  UNI,W
107 CALL  D7SEG
108 BSF   PORTA,2
109 MOVWF PORTB
110 CALL  DELAY
111 DECFSZ CONT,F
```

```
112 GOTO LAZO2
113 CALL ESPERA
114 BTFSS PORTA,1
115 GOTO SIGA
116 GOTO REG
117 SIGA CALL ESPERA
118 DECFSZ UNI,F
119 GOTO LAZO1
120 CALL ESPERA
121 BTFSS PORTA,1
122 GOTO SIGA1
123 GOTO REG
124 SIGA1 CALL ESPERA
125 LAZO4 MOVLW 0X64
126 MOVWF CONT
127 LAZO3 MOVF DEC,W
128 CALL D7SEG
129 BCF PORTA,2
130 MOVWF PORTB
131 CALL DELAY
132 CLRW
133 MOVF UNI,W
134 CALL D7SEG
135 BSF PORTA,2
136 MOVWF PORTB
137 CALL DELAY
138 BTFSS PORTA,1
139 GOTO SIGA2
```



```
140 GOTO REG
141 SIGA2      DECFSZ CONT,F
142 GOTO LAZO3
143 CALL  ESPERA
144 BTFSS PORTA,1
145 GOTO SIGA3
146 GOTO REG
147 SIGA3      MOVLW 0X02
148 MOVWF CONT2
149 LAZO5      MOVLW 0X09
150 MOVWF UMI
151 DECF  DEC,F
152 LAZO6      MOVLW 0X64
153 MOVWF CONT
154 LAZO7      MOVF  DEC,W
155 CALL  D7SEG
156 BCF  PORTA,2
157 MOVWF PORTB
158 CALL  DELAY
159 CLRW
160 MOVF  UNI,W
161 CALL  D7SEG
162 BSF  PORTA,2
163 MOVWF PORTB
164 CALL  DELAY
165 DECFSZ CONT,F
166 GOTO LAZO7
167 CALL  ESPERA
```

```
168 BTFSS PORTA,1
169 GOTO SIGA4
170 GOTO REG
171 SIGA4      DECFSZ UNI,F
172 GOTO LAZO6
173 CALL  ESPERA
174 BTFSS PORTA,1
175 GOTO SIGA5
176 GOTO REG
177 SIGA5      MOVLW 0X64
178 MOVWF CONT
179 LAZO8      MOVF  DEC,W
180 CALL  D7SEG
181 BCF  PORTA,2
182 MOVWF PORTB
183 CALL  DELAY
184 CLRW
185 MOVF  UNI,W
186 CALL  D7SEG
187 BSF  PORTA,2
188 MOVWF PORTB
189 CALL  DELAY
190 DECFSZ CONT,F
191 GOTO LAZO8
192 BTFSS PORTA,1
193 GOTO SIGA6
194 GOTO REG
195 SIGA6      DECFSZ CONT2,F
```

```
196 GOTO LAZO5
197 BSF  PORTA,3
198 MOVLW 0X03
199 MOVWF CONT1
200 LAZOA      MOVLW 0X64
201 MOVWF CONT
202 LAZO9      MOVF  DEC,W
203 CALL  D7SEG
204 BCF  PORTA,2
205 MOVWF PORTB
206 CALL  DELAY
207 CLRW
208 MOVF  UNI,W
209 CALL  D7SEG
210 BSF  PORTA,2
211 MOVWF PORTB
212 CALL  DELAY
213 DECFSZ CONT,F
214 GOTO  LAZO9
215 DECFSZ CONT1,F
216 GOTO  LAZOA
217 BCF  PORTA,3
218 LAZOB MOVLW 0X64
219 MOVWF CONT
220 LAZOC      MOVF  DEC,W
221 CALL  D7SEG
222 MOVWF PORTB
223 CALL  DELAY
```

```
224 CLRW
225 MOVF  UNI,W
226 CALL  D7SEG
227 BSF   PORTA,2
228 MOVWF PORTB
229 CALL  DELAY
230 DECFSZ CONT,F
231 GOTO  LAZOC
232 BTFSS PORTA,0
233 GOTO  LAZOB
234 GOTO  REG
235 ESPERA    BTFSS PORTA,0
236 GOTO  FIN
237 MOVLW 0X64
238 MOVWF CONT2
239 MV1      MOVF  DEC,W
240 CALL  D7SEG
241 BCF   PORTA,2
242 MOVWF PORTB
243 CALL  DELAY
244 CLRW
245 MOVF  UNI,W
246 CALL  D7SEG
247 BSF   PORTA,2
248 MOVWF PORTB
249 CALL  DELAY
250 DECFSZ CONT2,F
251 GOTO  MV1
```

```
252 BTFSS PORTA,0

253 GOTO MV1
254 GOTO FIN
255 FIN      RETURN
256 DSEG     ADDWF PC,F
257 RETLW 12
258 RETLW 71
259 RETLW 47
260 D7SEG    ADDWF PC,F
261 RETLW 40
262 RETLW 79
263 RETLW 24
264 RETLW 30
265 RETLW 19
266 RETLW 12
267 RETLW 02
268 RETLW 78
269 RETLW 00
270 RETLW 18
271 DELAY    BCF     INTCON,2
272 MOVLW 0XEC
273 MOVWF TMR0_OPT
274 DEL1     BTFSS  INTCON,2
275 GOTO    DEL1
276 RETURN
277 END
```

5.3 DISEÑO Y CONSTRUCCIÓN DE UN MARCADOR ELECTRÓNICO PROGRAMABLE DE 2 DIGITOS O DISPLAYS UTILIZANDO EL PIC 16F876

JUSTIFICACIÓN:

Para nuestro diseño utilizaremos un micro controlador de la familia 16F876. Debido a su versatilidad, su memoria y su mayor disponibilidad de pines que nos dan mayor cantidad de acceso para conectar dispositivos de entrada y salida (E/S).

La tarjeta controladora maneja dos displays implementados con 156 leds utilizando para ello una multiplexación en base de tiempo. Los dibujos requeridos para la fabricación de las tarjetas electrónicas se obtuvieron a través del manejo de los programas para dibujos de circuitos impresos, EAGLE Y PROTEL. (Véase fotografía 2 y 3)

5.4 Diseño y cálculos de tarjetas de displays implementadas con 156 leds

El dibujo en las tarjetas se hará usando los programas de EAGLE y PROTEL, a partir del layout o diagrama esquemático del circuito, este programa nos genera un dibujo a partir de un autoruteo que tiene el programa que se encarga de posicionar los componentes y hacer el trazado de las pistas lo que nos da como resultado el área utilizada de cobre, en definitiva las dimensiones o tamaño de la placa.

Los cálculos del ancho de la pista vienen dados por la siguiente formula analítica:

$$I = 3 \times 10^{-3} \sqrt{L^2}$$

Donde L representa el ancho de la pista en mm, e I la corriente en amperios.

Esta fórmula tiene un rango de exactitud que varía entre un 8% hasta un 15% aproximadamente, su exactitud depende del espesor de la lámina de cobre adherida a la

baquelita, cuyos valores fluctúan entre 0025 y0070mm.

La otra fórmula que puede aplicarse es la siguiente:

$$I = 2P^{0.625} \times A^{0.8}$$

Donde P representa la cantidad de onzas por pie cuadrado, valor que fluctúa desde 1/2oz hasta 3oz por cada pie cuadrado de material.

Donde A representa el ancho de cada pista.

Casi siempre el valor medio se toma para P=2 y en la práctica se diseñan pistas que lleven como máximo corrientes de hasta 10 amp.

La segunda fórmula tiene un error que fluctúa en un rango comprendido entre el 1 y 3% dependiendo de la pureza de la lámina de cobre.

Dibujo en las tarjetas: utilización de proceso de estampado en caliente para lo cual se usa un transfer y los pasos que detallamos a continuación: **(Véase fotografía 4)**

La primera acción a tomarse es la selección del tipo de baquelita a utilizarse para la tarjeta controladora en vista a que es un circuito a doble lado baquelita del tipo FR4(fibra de vidrio a dos caras y baquelita del tipo FR2 de una sola cara, para implementar los visualizadores o displays de 2 dígitos. una vez elegida la baquelita se procederá a un procedimiento de limpieza y desengrase para lo cual deberán ser lavadas con una solución concentrado de agua con detergente o en su defecto una disolución de agua con jabón de litio al 30%.El proceso deberá de ser realizado con guantes para evitar que la placa vuelva a engrasarse.

Al terminar de realizarse este proceso deberá de terminar de limpiarse la placa usando un isopo de algodón empapado en alcohol isopropílico, alcohol etílico o de preferencia

diluyente. Finalmente deberá de ser guardada en una funda a fin de evitar cualquier contaminación.

Para proceder al proceso de estampado sobre las placas deberá de obtenerse el dibujo en un acetato cuidando que la impresión de este sea la mas fidedigna posible y que los trazos de las pistas queden en color negro brillante. Posteriormente deberá de obtenerse una fotocopia sobre un transfer en imagen virtual.

Cuidadosamente y utilizando guantes se procederá a extraer de las fundas a las baquelitas previamente limpiadas y desengrasadas, a estas cuidadosamente se les frotara una disolución de acetona con lanolina hidrosoluble al 10% , esto se hace para que en el proceso de estampado que se detalla a continuación el papel del transfer no se vaya a quedar adherido.

Una vez realizado esto se calentaran las placas de baquelita a una temperatura 120° C, sobre las cuales se han colocado cuidadosamente los transfers y se cerrara la plancha de estampado por un tiempo no mayor a 40seg, posteriormente se retirará la placa de la plancha una vez que esta se haya enfriado y se procederá cuidadosamente a retirar el papel a fin de que en la superficie del cobre de las baquelitas queden estampados los dibujos de nuestros indicadores visuales. Si en el proceso de retirar el papel se a lastimado alguna pista del circuito impreso se retocara usando marcadores de tinta permanente de preferencia STAEDTLER del tipo LUMO COLOR, lo que no permitirá que esa pista se deteriore durante el proceso de denudación. **(Véase fotografía 5)**

El proceso de denudación se hará en un recipiente plástico tipo bandeja en donde se colocara una solución al 1.2 molar del percloruro de hierro mas agua, a fin de acelerar el proceso es conveniente que esta solución haya sido precalentada a una temperatura de 60°C, en esta se colocaran las placas y es conveniente que esta sea agitada permanentemente para que el proceso de denudación se realice de manera uniforme, al estar la superficie de cobre expuesta a esta solución se tornara de un color casi naranja

intenso que produce una reacción química en el cobre que da como resultado que este se desgaste produciendo una deposición de percloruro de cobre en la solución . El proceso completo tarda aproximadamente 15 minutos si se hace con la solución a 60°C si se hace a temperatura ambiente el proceso tarda de 30 a 40 minutos se puede reducir el tiempo en alrededor de 3 minutos utilizando una solución de ácido nítrico al 20% el gran problema de usar este ultimo es que debe de ser realizado en un sitio muy ventilado y que tenga un extractor de aire por los vapores rutilantes que emanan durante el proceso de denudación. **(Véase fotografía 6)**

El proceso de perforación de las placas deberá ser realizado utilizando de preferencia un taladro de alta velocidad (sobre las 10000 rpm) y brocas cuyos diámetros normalmente varían desde 1/32plg hasta 5/32plg una ves perforadas las placas deberá de adicionarse los componentes estos se fijaran a la placa por medio de soldadura de estaño plomo (60%Sn, 40%Pb) que tenga pez griega, el proceso de soldado deberá de realizarse con un caufín de 40W, para no dañar las pistas. La pez griega y la grasa ácida de soldar nos producen soldaduras eutécticas con acabado brillante al terminar de soldar los componentes los sobrantes de conductor en los mismos deben de ser debidamente cortados con una pinza de tipo pico de loro o cortados a ras con un corta uñas, la limpieza de las placas se hará con una brocha de 1plg cuyas cerdas han sido cortadas a 1cm, esta se empapara en diluyente y se pasará por la placa de baquelita, ya que la pez griega cristalizada junto con la grasa ácida en el decurso del tiempo si se las deja se transforman en un conductor eléctrico, lo que puede producir un corto circuito en las placas. Finalmente se procederá a aislar y a proteger de la corrosión a las placas para ello podremos utilizar, pintura transparente en aerosol, barniz DOLPHIN tipo ER-41, resina poliéster del tipo palatal 5 o cualquier barniz que tenga un buen índice de rigidez dieléctrica (recomendable sobre un 1Kv/mm).

El transfer es un papel que esta constituido por celulosa a la que se le a adicionado una película de látex polivinílico.

Lo primero a realizarse es la impresión en el transfer del dibujo que se va a utilizar el circuito impreso, este dibujo deberá imprimirse en lo que se llama imagen virtual o espejo, a fin de que cuando se haga la transferencia la imagen real sea la que se obtenga después del proceso de denudación del circuito impreso.

Obtención del circuito impreso por medio de la denudación del cobre utilizando una solución de percloruro férrico a una concentración de 1.2 g/cm^3

Proceso de PERFORACIÓN de las tarjetas, se utilizaran brocas de 1/32 plg, 1mm, 5/32 plg.

El proceso descrito anteriormente se usa en la gran mayoría de los casos en las tarjetas de una faz (cara o lado).

Para las tarjetas que son de doble lado lo mas recomendable es utilizar una tarjeta foto sensibilizada previamente en lo que se lo llama fotolitos ya sea este positivo o negativo dependiendo si se utiliza luz blanca, (provenientes de lámparas floreciente) luz ultravioleta o simplemente luz solar. En el caso de no conseguirse material foto sensibilizado, se recomienda seguir el siguiente proceso para foto sensibilizar la superficie de cobre de la baquelita.

1. Se tomará la placa y se la limpiara y se la desengrasara con la metodología descrita anteriormente
2. Se procederá a preparar una solución de alcohol polivinilico 0.25molar, para ello se tomaran alcohol polivinilico en granallas o en polvo y se disolverá una cantidad de 250gr.y se lo disolverá en un litro de agua, de preferencia agitándolo fuerte mente o introduciéndolo en una licuadora. Una vez realizado este proceso se dejara reposar la solución por un tiempo estimado de 1 hora posteriormente se tomara esta solución y se la calentara en baño de maría revolviéndola constantemente, a fin de que esta no quede grumosa, cuando la solución se torna amarillenta, deberá entonces cesar el

baño de maría y cuando esta este bien tibia es decir a una temperatura aproximada de entre 50 y 60 C procederemos a cernirla en una tela de hilo blanca, el contenido deberá de ser depositado en alguna botella obscura, una vez que la solución este total mente fría, se adicionara 100gr. de Dicromato de Potasio ($CR_2O_7K_2$), esta solución finalmente deberá ser guardada en un sitio oscuro.

3. Para sensibilizar nuestra placa necesitaremos un cuarto oscuro con una iluminación tenue que debe de ser provista por una lámpara incandescente de 15W y que nos provea de una luz de color rojo.
4. Deberemos de construir una pequeña centrífuga cuyo armazón permitirá adosar la placa de baquelita y a su vez sujetarla a un taladro cuyas revoluciones deberán de modificarse con un dimer de luz a fin de que estas no excedan las 300rpm.
5. Deberá verterse sobre el centro de la placa la solución previamente preparada que tendrá una consistencia aproximada a la de la goma escolar, se adosara la placa a la centrífuga y se la pondrá a funcionar, hasta que la solución se allá dispersado en la placa de manera homogénea y uniforme (la exigencia de hasta 300rpm. se debe a que la tensión superficial soporta hasta esa velocidad, pues en caso contrario se desparramaría.)
6. Luego de esto deberá secarse la solución utilizando un secador de mano por un tiempo aproximado de 30 minutos.
7. Posteriormente sin quemar la placa se le aplicara calor colocándola en una lamina metálica plana, puesto este conjunto sobre un reverbero en un periodo no mayor a 2 minutos, a fin que quede la solución muy bien fijada (vulcanizada)
8. La placa sensibilizada deberá guardarse de una funda negra para que no se vele.
9. Una vez obtenido los fotolitos del circuito impreso, deberá utilizarse uno de los 2, si escogemos el positivo, lo pondremos sobre la baquelita sensibilizada, y lo revelaremos utilizando la luz de 2 lámparas florecientes de 40W cada una, ubicando la placa a 30cm. de distancia por un periodo de un minuto.
10. Posteriormente por un lapso de 30 segundos deberá revelarse la placa en una solución similar a las que se usan para el proceso de revelado fotográfico (solución de revelador, fijador y agua).

11. Posteriormente enjuagar con abundante agua la placa y luego someterla al proceso de denudación descrito anteriormente.

5.5 ISEÑO Y CALCULOS DE LOS DISPLAYS: (LUMENES, CANDELAS, ESTEREO RADIAN) VOLTAJE Y CORRIENTE DE OPERACIÓN DE UN LED).

Según catálogo ALLIED de componentes electrónicos Pág. 351 ver tabla (10) para los leds que típicamente se consiguen en el mercado local y cuyo numero de stock es el 505-21117, la intensidad "I" es igual a 7 micro candelas, con el elemento de estas características se construyeron los displays, luego Se hicieron pruebas empíricas para determinar el grado de visibilidad de un LED encendido.

A fin de determinar la distancia media en la cancha a la que debe de estar el operador de los marcadores y que fácilmente pueda observar con comodidad su funcionamiento, se desarrollo la siguiente formula empírica:

$$D = (\eta I_s) / 2$$

D = distancia media a la cual se encuentra el operador y publico, medida en metros

I = intensidad luminosa medida en micro candelas

η = numero total de leds en los displays

S = sensibilidad relativa del ojo humano en función de la longitud de onda (**Véase figura 22**)

Lo que nos daría una distancia de 54.6 mtrs.

Con esta distancia y conociendo el ángulo medio de visibilidad de los leds (ver tabla 10) podemos calcular la longitud del arreglo de cada segmento del display, utilizando la siguiente fórmula empírica:

$$X = 2.3D \text{ Sen } (\theta/2)$$

D = distancia media a la cual se encuentra el operador y público, medida en metros

X = es la distancia en milímetros de cada arreglo de leds.

Θ = el ángulo de visibilidad del leds. **(Véase figura 22)**

Lo que nos da un valor de 80.72 milímetros, asumiendo una distancia de separación de 15 milímetros entre led y led nos daría un promedio de 5,3 leds, por lo tanto los arreglos serán de filas de 5 ó 6 como máximo y reforzando en otra fila para mejorar la visibilidad.

Datos que concuerdan en la práctica, ya que estos marcadores normalmente se utilizan en pares y a una distancia máxima de 50 metros; en vista de que la presentación del displays se asemeja a un displays de 7 segmentos, cada segmento, entonces estará constituido por 11 LEDS (del segmento “a” al segmento “f”) y el segmento “g” por 12, lo que totaliza un total de 78 LEDS por displays. **(Véase fotografía 7)**

Si cada LED se conecta en serie y tiene un voltaje promedio de operación de 3.15Volts. Y si cada segmento está constituido por un arreglo de LEDS conectados convenientemente a un ánodo común esto nos da un voltaje de operación que varía de entre 35V hasta 37V aproximadamente ya que el segmento “g”(central) requiere de este voltaje para su apropiada operación, es decir que necesitamos una fuente que nos de un valor de hasta 37V DC. Lo que nos indica que requerimos un transformador con un voltaje en el secundario de 27 V. AC. Si cada LED para iluminación plena requiere aproximadamente 10ma DC. requeriremos entonces de un transformador que nos da como resultado una corriente de secundario de 1.6 amp. si el pito o claxon y la luz que se utiliza como señalizadores que indican la finalización del conteo tienen un consumo de 0.8amp y de 0.1amp respectivamente(valores medidos), totalizaría entonces 2,5amp aproximadamente a esto habrá que agregar el consumo del PIC de los demás elementos electrónicos y del relay lo que totaliza aproximadamente 200ma. Más, lo que entonces daría un gran total aproximado de 2.7amp.

Asumiendo una tolerancia del 10% nos daría entonces un transformador de 3 amp

5.6 CALCULOS Y DISEÑO DE LA FUENTE DE ALIMENTACIÓN DEL EQUIPO

DISEÑO Y CONSTRUCCIÓN DEL TRANSFORMADOR:

Dado que el transformador que requerimos para alimentar el circuito de fuerza es de 27 VAC y 3 AMP, podríamos calcular el área del hierro del transformador asumiendo un flujo de 1.2 Wb/m^2 , el área del hierro del transformador será

$$A(\text{Fe}) = 1.2 \sqrt{P_{\text{prim}} \text{KW}}$$

Idealmente la potencia del primario será igual a la potencia del secundario en KVA, asumiendo que se desprecian las pérdidas por ser muy pequeñas, por lo tanto:

$$P_{\text{prim}} = P_{\text{sec}}$$

Por lo tanto son 81 VA lo que nos da un área de hierro de:

$$A(\text{Fe}) \text{m}^2 = 1.2 \sqrt{81}$$

El área de fe neta será igual a: $A = A(\text{Fe}) \cdot K$ donde k es el factor de apilamiento que por tratarse de láminas nuevas debe de ser considerado como $K=0.97$ por lo tanto el área $12.6 \times 0.97 = 12.222$, considerando que las "E" de las cuales están hechas los núcleos de los transformadores son de hierro silicio al 4% y que vienen fabricadas en varios tamaños seleccionamos el que se describe a continuación:

- 1.- Longitud = 9.55
- 2.- Ancho = 6.35 cm
- 3.- Ancho de la parte central del núcleo = 3.15 cm = An

- 4.- Ancho de la E = 1.6 cm
- 5.- Ancho de la ventana = 1.6 cm
- 6.- Altura de la ventana = 6.35 cm (Hv)
- 7.- Espesor de la E = 0.55 cm
- 8.- Longitud de la I = 9.55 cm
- 9.- Ancho de la I = 1.6 cm
- 10.- ancho total del transformador: 7.95 cm

Lo que nos da lo siguiente: El $A = A_n \cdot L$ donde L es el espesor del transformador por lo tanto $L = A/A_n = 3.88$ cm ; sea N_t el número de láminas del transformador, sería $N_t = L/E = 70.54$; por lo tanto aproximamos a un número par de láminas, lo que nos daría un total de 70 láminas; es decir 35 “E” y 35 “I”.

5.7 CALCULO DE LA CORRIENTE REQUERIDA EN EL PRIMARIO, EL NUMERO DE ESPIRAS DEL SECUNDARIO Y EL NUMERO DE ESPIRAS DEL PRIMARIO.

Para la mayoría de los transformadores que se producen el voltaje viene dado por la siguiente expresión matemática donde $\omega = 2\pi f$

$$E = \frac{\sqrt{2}}{2} N \omega \Phi$$

$$E_{\max} = -N d \Phi / dt ; \quad \Phi = \Phi_{\max} \cos t ; \quad \Phi_{\max} = B_{\text{afe}}$$

$$E_{\max} = -N d / dt (B_{\text{fe}} \cos t)$$

$$E_{\max} = N B A F_e \omega \sin \omega t$$

$$E_{\max} = N B A F_e 2 \pi f$$

$$E_{\text{RMS}} = E_{\max} * \frac{\sqrt{2}}{2} = \frac{\sqrt{2}}{2} (2 \pi f) B A F_e N$$

$$ERMS = 4.4442 BNAFef \quad \text{pero } B = 1,4\text{Wb/m}^2 \quad \text{pero Fe - Si al 4\% si } F = 60 \text{ Hz}$$

Donde $E = 373.128 N Afef 10^{-4}$ donde $Afef$ esta dada en cm^2

Donde $Ns = 59,21$ espiras }

∴ Por similar análisis $Np = 252,17$ espiras; con un voltaje de 115vlt., para un voltaje de 230volts. Requeriríamos $Np = 504,34$ espiras:

Es decir nuestro transformador redondeando el numero de espiras que requiere será $Ns = 59$ espiras o vueltas ya que se desprecia la parte decimal ya que no supero a 0,5, lo que nos permitiría redondear las espiras al número inmediato superior. Para el primario serán $Np = 504/252$ espiras ya que el voltaje de operación es de 230/115volts., es decir entre los extremos del bobinado del primario ubicamos un tab central a 252 espiras,

5.8 SELECCIÓN CALCULO DE LA CORRIENTE DEL PRIMARIO.

Sabiendo que el numero de volt./amp. del secundario ya que despreciamos la pérdida por calentamiento, por histéresis y por corrientes de EDDY, tendríamos la siguiente relación.

$$\frac{Ep}{Es} = \frac{Is}{Ip}$$

$$\text{Si: } Ep = 115\text{V} \quad \therefore Ip = \frac{(27)(3)}{115} = 0.704\text{amp.}$$

$$\text{Con: } Ep = 230\text{V} \quad \therefore Ip = 0.35\text{amp}$$

5.9 CÁLCULO DE LA SECCION TRANSVERSAL DE LOS CONDUCTORES REQUERIDOS TANTO COMO EN EL DEVANADO PRIMARIO, COMO EN EL DEVANADO SECUNDARIO DEL TRANSFORMADOR:

Para la mayoría de los transformadores se asume una densidad de corriente $J = 3 \text{ amp/mm}^2$.

Sabiendo que $I = J \cdot A_{\text{cu}}$ donde “ A_{cu} ” es la sección transversal del conductor tendríamos para nuestro caso:

$$A_{\text{cu}}(\text{p}) = \frac{I_p}{J} = \frac{0.704}{3 \text{ amp} / \text{mm}} = 0.235 \text{ mm}^2$$

$$A_{\text{cu}}(\text{s}) = \frac{I_s}{J} = \frac{3 \text{ amp}}{3 \text{ amp} / \text{mm}} = 1 \text{ mm}^2$$

Con estas secciones transversales podríamos definir el número AWG de los conductores requeridos para ello utilizamos la siguiente fórmula:

$$A_{\text{cu}} = 2^{(17 - \#) / 3}$$

donde # representa el numero AWG del conductor requerido.

Por lo tanto el número requerido tanto para el conductor del primario como en el conductor secundario es:

$$\therefore \#S = 17\text{AWG}$$

$$\therefore \#P = 23\text{AWG}$$

5.10 CÁLCULO DE LA MASA DEL CONDUCTOR REQUERIDO PARA EL TRANSFORMADOR

Para el bobinado del primario la masa de cobre requerida, $M_{cu(p)}$, será igual a la densidad del cobre multiplicada por el volumen del alambre, es decir:

$$M_{cu(p)} = \frac{\pi}{4} \cdot D_p^2 \cdot \delta_{cu} \cdot P_p$$

Para lo cual es necesario conocer el perímetro P_p del conductor del bobinado primario, este lo calculamos de la siguiente manera:

Si $A_n = 3.15\text{cm}$, $H_v = 6.35$ y $D_p = 0.57\text{ mm}$ (Diámetro del conductor # 23), y $N_p = 504$
El perímetro P_p vendrá dado en función de el # de espiras exactas que entran por hilera de devanado en donde el # de espiras por hilera viene dado por la siguiente expresión

$$K_p = N_p / D_p = 111 \text{ espiras}$$

$K_p' = N_p - 4 K_p$ (para este caso, ya que K_p' no completa una hilera y por lo tanto es un sobrante)

Por inducción matemática, ya que por cada vuelta el perímetro aumenta en función del diámetro del conductor para nuestro caso P_p vendría dado por la siguiente expresión matemática:

$$P_p = 2K_p(A_n + L) + 2K_p(A_n + L / D_p) + 2K_p(A_n + L + 8D_p) + 2K_p(A_n + L + 16D_p) + K_p'(A_n + L + 20D_p)$$

(Expresión que nos da el perímetro o longitud total del alambre en cm)

$$\text{Donde } P_p = 7087.15 \text{ cm}$$

Finalmente, la masa requerida para el primario:

$$M_{cu(p)} = 161.32\text{g} + 10 \%, \text{ es decir:}$$

$M_{cu(p)} = 177 \text{ g}$ (véase fotografía 8)

Para calcular la masa del secundario debemos tener en cuenta que el arrollamiento del devanado secundario va sobre el primario, y este está constituido por 60 espiras de alambre # 17 cuyo diámetro es 1.15 mm, por lo tanto este va en 2 hileras de espiras y el perímetro vendría dado por la siguiente fórmula

$P_s = N_s (A_n + L + 20Ds) + N_s' (A_n + L + 24Ds)$;

donde $N_s' = 5$ espiras, por lo tanto el perímetro sería:

$P_s = 562.1 \text{ cm}$

Por lo tanto, la masa del conductor requerida para el devanado secundario será:

$M_{cu(s)} = 52.08 + 10\%$, lo que nos da aproximadamente:

$M_{cu(s)} = 60 \text{ gramos}$ (Véase fotografía 8)

5.11 CONSTRUCCION DE LA ESTRUCTURA, MONTAJE Y FUNCIONAMIENTO DEL MARCADOR

Para el montaje del marcador se utilizó por su versatilidad, ligereza y facilidad de corte y maquinado, canales y tubos de aluminio cortados en una dimensión de 7.6 cm por 47.2 cm y 3.8cm por 39.5 cm respectivamente. **(Véase fotografía 9).**

Este conjunto fue remachado adosando 4 ángulos de aluminio en una dimensión de 5 x 5 cm.

Posteriormente se utilizaron 16 remaches de 1/8 de pulg. de forma tal que quedó armada la estructura.

Para la mascarilla se utilizó acrílico verde de 2 mm de espesor en una dimensión de 47,7 por 32,2 cm, esta mascarilla fue perforada utilizando para ello una matriz construida en lámina de hierro gris de 1/8 de pulg. y con las mismas dimensiones de la lámina de acrílico. **(Véase fotografías 10 y 11).**

Por esta mascarilla es que salen los 156 leds las luces y el sonido del marcador.

Para la implementación del sonido del marcador se utilizó un par de pitos de carro a 12 V. y un filtro pasa banda construido con un condensador de 0.1 μ f /630 V. conectado en serie con una resistencia de 1 k a fin de evitar que este se quemara por cualquier distorsión armónica mayor a 400 Hz.

La tapa posterior del marcador se fabricó utilizando cartón prensado de 4mm de espesor **(Véase fotografía 12).**

La transmisión de la señal se lleva a cabo a través de un integrado que es el MAX 232 este permite la comunicación serial y sincronización entre los dos marcadores además de poseer un buffer interno que nos garantiza la transmisión de la señal en cable telefónico cuya longitud sea de hasta 500 m.

Es decir, la transmisión y comunicación de datos entre los marcadores se hace utilizando un sistema maestro / esclavo (master/slave).

El envío de los datos del PIC16F876 a los Displays para el encendido de los siete segmentos se hacen a través del circuito integrado ULN2004 que es un buffer con inversor con una capacidad de manejo de corriente de hasta 500 ma. Lo que garantiza una óptima iluminación de los leds en cualquier representación numérica.

La polarización de los displays para su encendido y apagado viene controlada a través de los pines RC1 y RC0 del PIC., estos a su vez manejan un par de transistores 2N3904, que con una conveniente red de polarización, ponen en corte y saturación a dos transistores tipo darlington, número TIP32 los cuales energizan y desenergizan a los displays en sincronización con el PIC a fin de mostrar el dato, en un intervalo de 5 ms, por cada uno, lo que totaliza 10 ms que repetido en un lazo de 100 veces nos da el segundo de conteo requerido.

En las **fotografías 13 y 14** se muestra, la botonera o control maestro y los marcadores terminados.

5.12 ALGORITMO DEL PROGRAMA DE LOS MARCADORES

```
*****
00001 LIST P = 16F876,
00002 ; Reloj de Cristal.
00003 ; Reloj a 4 Mhz
00004
*****
00005 ; Inicio de Programa.
00006 ; DESCRIPCIÓN DEL PROGRAMA
00007 ;
00008 ; Este programa corre con el PIC16F876.
00009 ; DISPLAYS
00010 ; Constituidos por 78 leds en arreglos de 7 segmentos, multiplexados en base de
           tiempo y en configuración ánodo común sin punto de separación decimal;
00011 ; Las salidas son controladas a través del Puerto B y las señales de control a
           través de dos pines del Puerto C así como se usaran 3 pines del puerto A
           como puertos de entrada- salida (E/S);
00012 ;
```

00013 ; VALORES DE VARIABLE
00014 ; 00041 PIC16F876 equ H'01FF'
00015 ; 00042 PIC16F876 equ H'03FF'
00016 ; 00044 POINTER equ H'00'FF'
00017 ; 00045 TMR0 equ H'01'
00018 ; 00046 PC equ H'02'
00019 ; 00047 STATUS equ H'03' ; F3 Registro estado.
00020 ; 00048 FSR equ H'04'
00021 ; 00050 PORT_A equ H'05' ; Display de 7 segmentos configuración ánodo
común.
00022 ; 00051 PORT_B equ H'06' ; ()
00023 ; Status Reg.
00024 ; 00052RRY equ 0 ; Carry Bit es cero
00025 ; 00053equ 0
00026 ; 00054ARRY equ 1
00027 ; 00055 equ 1
00028 ; 00056_bit equ 2 ; Bit 2 of F3 es cero bit
00029 ; 00057qu 2
00030 ; 00058OWN equ 3
00031 ; 00059qu 3
00032 ; 00060UT equ 4
00033 ; 00063 equ 4
00034 ; 00064equ 5 ;16F87 status bits
00035 ; 00065A1 equ 6 ;16F876 status bits
00036 ; 00066A2 equ 7 ;16F876 status bits
00037 ; 00068 CERO equ H'7E'
00038 ; 00069 UNO equ H'0C'
00039 ; 00070 DOS equ H'B6'
00040 ; 00071 TRES equ H'9E'
00041 ; 00072 CUATRO equ H'CC'

00042 ; CINCO equ H'DA'

00043 ; 00074 SEIS equ H'FA ; Asignación de display en puerto B

00044 : 00075 SIETE equ H'OE'

00045 ; 00076 OCHO equ H'FE'

00046 ; 00077 NUEVE equ H'CE'

00047 ; 0F0 00079 T equ H'F0'

00048 : 00000 00080 BLANK equ H'00'

00049 ; Selección de salidas para display.

00050 ; 000FE 00091 DISP1 equ B'11111110'

00051; 000FD 00092 DISP2 equ B'111111101' Selección de display Puerto B
(PORT_B)

00052 ; 000FB 00093 DISP3 equ B'11111011'

00053 ; 000F7 00094 DISP4 equ B'11110111'

00054 ; 000FF 00095 DISPOFF equ H'FF'

00055; 0000E 00096 SWITCH equ B'00001110' ; Activado RA0 hasta RA3 como
entradas asignación de banderas

00056 ; 00004 00103 SW1 equ H'4' ; Asignación de Banderas con switches iguales a 1
lógico.

00057 ; 00007 00106 SW_ON equ H'7' ; cuando el swith es
presionado

00058 ; DEFINICIÓN DE PARÁMETROS

00059 ; VARIABLES

00060 ; 00008 00109 keys equ H'08' ; localización de variables,
que llaves están presionadas?

00061 ; 0000B 00112 display equ H'0B' ; localización de
variables con actualización de display.

00062 ; 0000C 00113 digit1 equ H'0C'

00063 ; 0000D 00114 digit2 equ H'0D'

00064 ; 0000E 00115 digit3 equ H'0E'

00065 ; 0000F 00116 digit4 equ H'0F'

00066 ; Inicio de conteo
 00067 ; Variables de tiempo
 00068 ; Conteo desde 00segs hasta 99segs.
 00069 ; 00010 00120 sec_nth equ H'10'
 00070 ; 00011 00121 seconds equ H'11' ; seconds
 00071 ; 00014 00124 var equ H'14' ; variable para almacenamiento temporal de datos
 00072 ; 00015 00125 count equ H'15' ; Variable para conteo del lazo.
 00073 ; 00016 00126 count2 equ H'16' ; Segundo lazo de conteo
 00074 ; Inicializando las salidas para displays en blanco
 00075 ; 0C03 00133 START movlw H'03' ;
 00076 ; 0002 00134 option ; Prescaler TMR0, 1:16
 00077 ; 0C00 00136 movlw 0
 00078 ; 0005 00137 tris PORT_A;Selección de los pines de salida
 00079 ; 0006 00138 tris PORT_B
 00080 ; 0C00 00139 movlw BLANK
 00081 ; 0026 00140 movwf PORT_B ; Blanqueo del display
 00082 ; 04C3 00141 bcf STATUS,PA1
 00083 ; 04A3 00142 bcf STATUS,PA0
 00084 ; inicialización de variables
 00085 ; 0C01 00145 movlw H'01'
 00086 ; 0021 00146 movwf TMR0
 00087 ; 0CFE 00147 movlw H'FE'
 00088 ; 002B 00148 movwf display ; selecciona primer display
 00089 ; 0C00 00149 movlw BLANK; pone los dos displays en blanco
 00090 ; 002C 00150 movwf digit1
 00091 ; 002D 00151 movwf digit2
 00092 ; 0C00 00162 movlw H'00'
 00093 ; 0029 00163 movwf flags
 00094 ; Espera el desbordamiento del TMR0
 00095 ; 00170 TMR0_FILL


```
00096 ; 0201 00171 movf TMR0,0
00097 ; 0743 00172 btfss STATUS,Z
00098 ; 0A1C 00173 goto TMR0_FILL
00099 ; 00180 CHECK_SW
00100 ; 07E9 00181 btfss flags,SW_ON ; Si es que no se aplastan los interruptores.
00101 ; 0A3A 00182 goto SET_TIME
00102 ; 0689 00183 btfsc flags,SW1
00103 ; 0A3A 00184 goto SET_TIME
00104 ; 0031 00186 movwf seconds ; poner el conteo a 00
00105 ; 0C7F 00187 movlw H'7F'
00106 ; 07A9 00189 btfss flags,SW2
00107 ; 0CAF 00191 movlw H'AF'
00108 ; 0A5E 00201 goto CHECK_TIME
00109 ; 0A5E 00204 goto CHECK_TIME
00110 ; 00206 SET_TIME
00111 ; 0509 00207 bsf flags,SEC ; segundos mostrados
00112 ; 0569 00208 bsf flags,CHG ; un cambio fue hecho
00113 ; 03F1 00209 incfsz seconds,1 ; incrementa 1 seg
00114 ; 0A52 00210 goto TIME_DONE
00115 ; 0031 00212 movwf seconds 00213
00116 ; 0569 00215 bsf flags,CHG
00117 ; 0A52 00219 goto TIME_DONE
00118 ; 0569 00224 bsf flags,CHG
00119 ; 01F0 00226 addwf sec_nth,1
00120 ; 0A52 00228 goto TIME_DONE
00121 ; 0C03 00231 movlw ADJDAY
00122 ; 00234 TIME_DONE
00123;0769 00235 btfss flags,CHG ; que interruptor aplasto
00124 ; 0A8F 00236 goto CYCLE ; cambiar valor de display
00125 ; 00239 CHECK_SECONDS
```

```
00126 ; 0789 00241 btfss flags,SW1
00127 ; 0A5E 00242 goto CHECK_TIME
00128 ; 0C00 00243 movlw H'00'
00129 ; 002D 00244 movwf digit2
00130 ; 0091 00248 subwf seconds,0
00131 ; 002C 00249 movwf digit1 ;variable de conteo temporal
00132 ; 0A67 00250 goto SPLIT_HEX
00133 ; 00252 CHECK_TIME
00134 ; 0C00 00253 movlw H'00'
00135 ; 002F 00254 movf digit2
00136 ; 002D 00255 movwf digit2
00137 ; 002C 00261 movwf digit1
00138 ; 0C02 00267 movlw H'02'
00139 ; 0035 00268 movwf count ; Conteo de números de lazos
00140 ; FSR = digit1, 2nd time FSR = digit2
00141 ; 0C0C 00271 movlw digit1
00142 ; 0024 00272 movwf FSR
00143 ; 0A6E 00273 goto LOOP
00144 ; 0C0E 00275 LOOP2 movlw digit2
00145 ; 0024 00276 movwf FSR 00277
00146 ; 00278 LOOP
00147 ; 0C0A 00279 movlw D'10'
00148 ; 00A0 00280 subwf POINTER,1
00149 ; 0603 00281 btfsc STATUS,C ; verificación del reg estado
00150 ; 0A74 00282 goto INCREMENT_1S ; adiciona "1" al conteo
00151 ; 01E0 00283 addwf POINTER,1
00152 ; 0A78 00284 goto NEXT_DIGIT;salta al siguiente dígito
00153 ; 00286 INCREMENT_1S
00154 ; 02A4 00287 incf FSR,1
00155 ; 02A0 00288 incf POINTER,1
```

```

00156 ; 00E4 00289 decf FSR,1
00157 ; 00290 goto LOOP
00158 ; 00291 Escribe decf ADDR_CON2,1
00159 ; 00292 movf ADDR_CON2,0
00160 ; 00293 addlw INICIO
00161 ; 00294 movwf FSR
00162 ; 00295 movf INDF,0
00163 ; 00296 movwf DATA_CON1
00164 ; 0C0C 00297 ; secuencia de escritura
00165 ; 0024 00298 bcf INTCON,7
00166 ; 0C04 00299 bsf STATUS,5 movlw H'04'
00167 ; 0035 00300 movlw h '55' movwf count
00168 ; 00301 movwf ADDR_CON2 NEXT_HEX
00169 ; 0200 00302 movlw H'AA' POINTER,0
00170 ; 09C0 00303 movlw ADDR_CON2
00171 ; call RETURN_CODE ;
00172 ; 0020 00304 movlw b'11000000' POINTER
00173 ; 02A4 00305 movwf INTCON incf FSR,1
00174 ; 02F5 00306 bsf DATA_CON1,2 decfsz count,1
00175 ; 0A7E 00307 bsf DATA_CON1,2 goto NEXT_HEX
00176 ; 00308 WAIT slepp ; Se espera el fin de la escritura
        ; El WDT siempre tarda más en rebosar; que la escritura en completarse
00177 ; 00309 bcf STATUS,5 FIX_DISPLAY
00178 ; 0C7E 00310 clrw movlw ZERO
00179 ; 008F 00311 addwf ADDR_CON2,0 subwf digit2,0
00180 ; 0743 00312 btfss STATUS,2
00181 ; 0A8A 00313 goto ESCRIBE
00182 ; 0C00 00314 COMPROB movlw BLANK
00183 ; 002F 00315 movwf N,0 digit4
00184 ; 00316 movwf ADDR_CON2

```

```
00185 ; 0789 00317 FIX_SEC btfss flags,SW1
00186 ; 0A8D 00318 LEE Decf ADDR_CON2,1
00187 ; 002E 00319 movf ADDR_2,0 movwf digit2
00188 ; 00320 addlw INICIO
00189 ; 00321 movwf FSR CLEAR_FLAGS
00190 ; 0CF0 00322 bsf STATUS,5 movlw H'F0'
00191 ; 0169 00323 bsf DATA_CON1,0 andwf flags,1
00192 ; 00324 bcf STATUS 5
00193 ; 00325 bsf DATA_CON1,0 CYCLE
00194 ; 00326 bcf STATUS 5
00195 ; 0CFF 00327 ; comprobación movlw DISPOFF
00196 ; 0025 00328 movwf INDF,0 PORT_A
00197 ; 0C0E 00329 subwf DATA_CON1,0 movlw SWITCH
00198 ; 0006 00330 btfsc STATUS,2 tris PORT_B
00199 ; 0C0F 00331 goto FIN movlw H'0F'
00200 ; 0169 00332 ; siguiente andwf flags,1
00201 ; 0000 00333 clrw nop
00202 ; 0000 00334 addwf ADDR_CON2,0 nop
00203 ; 0000 00335 btfss STATUS,2 nop
00204 ; 0206 00336 goto LEE movf PORT_B,0
00205 ; 0034 00337 FIN slepp movwf var
00206 ; 0734 00338 goto FIN btfss var,1
00207 ; 0A9F 00339 goto SWITCH2
00208 ; 0569 00340 bsf flags,CHG
00209 ; 0589 00341 bsf flags,SW1
00210 ; 05E9 00342 bsf flags,SW_ON
00211 ; 0754 00343 SWITCH2 btfss var,2
00212 ; 0AA4 00344 goto SWITCH3
00213 ; 0569 00345 bsf flags,CHG
00214 ; 05A9 00346 bsf flags,SW2
```

```
00215 ; 05E9 00347 bsf flags,SW_ON
00216 ; 0774 00348 SWITCH3 btfss var,2
00217 ; 0AA9 00349 goto SETPORT
00218 ; 0569 00350 bsf flags,CHG
00219 ; 05C9 00351 bsf flags,SW3
00220 ; 05E9 00352 bsf flags,SW_ON
00221 ; 00354 SETPORT movlw H'00'
00222 ; 00355 tris PORT_B
00223 ; 00356 movlw BLANK
00224 ; 00357 movwf PORT_B
00225 ;
00226 ;070B 00360 btfss display,0
00227 ; 020F 00361 movf digit4,0
00228 ; 072B 00362 btfss display,1
00229 ; 020E 00363 movf digit2,0
00230 ; 074B 00364 btfss display,2
00231 ; 020D 00365 movf digit2,0
00B3 076B 00366 btfss display,2
00232 ; 020C 00367 movf digit1,0
00233 ; 0026 00368 movwf PORT_B
00235 ; 0506 00370 bsf PORT_B,0
00236 ; 020B 00371 movf display,0
00237 ; 0025 00372 movwf PORT_A
00238 ; 002B 00373 movwf display
00239 ; 036B 00374 rlf display,1
00240 ; 050B 00375 bsf display,0
00241 ; 078B 00376 btfss display,2
00242 040B 00377 bcf display,0
00243 ; 0A1C 00381 goto MAIN
00244 ; 00383 RETURN_CODE
```

00245 ; 01E2 00385 addwf PC,1
00246 ; 087E 00386 retlw ZERO
00247 ; 080C 00387 retlw ONE
00248 ; 08B6 00388 retlw TWO
00249 ; 089E 00389 retlw THREE
00250 ; 08CC 00390 retlw FOUR
00251 ; 08DA 00391 retlw FIVE
00252 ; 08FA 00392 retlw SIX
00253 ; 080E 00393 retlw SEVEN
00254 ; 08FE 00394 retlw EIGHT
00255 ; 08CE 00395 retlw NINE
00256 ; 00398 org PIC54
00257 ; 0A00 00399 goto START
00258 ; END

CONCLUSIONES Y RECOMENDACIONES

En el proceso de fabricación de este marcador se ha utilizado un microcontrolador PIC de la familia 16f876, por su versatilidad y economía lo que permite una reducción importante en cuanto a cantidad y costo de componentes, las tarjetas fueron hechas por el método de transfer en caliente lo que permite bajar su costo hasta 1.6 ctvs por cm², teniendo un costo total por marcadores de USD 200, lo que hace que este producto sea altamente rentable y competitivo en el mercado, ya que su similar no programable de procedencia argentina o norteamericana, tiene un precio CIF de USD 800 y 900, respectivamente.

Las versiones programables tienen un precio CIF de USD 1300, lo que nos daría un amplio rango en el precio para poder competir eficientemente en el mercado.

Actualmente estamos trabajando en una versión inalámbrica (con control remoto), que producida a partir de 6 juegos de marcadores tendría el mismo costo que el anterior, lo que nos permitiría ubicarnos en el mercado en una posición de liderazgo.

ANEXO

FOTOS, DIBUJOS, GRAFICOS Y TABLAS

TABLA 1

MODELO	MEMORIA PROGRAMA	MEMORIA DATOS	FREC. MAXIMA	LINEAS E/S	ADC 8 BITS	TEMP. TMRO+WDT	PATITAS
PIC12C508	512*12	25*8	4 MHZ	6		TMRO+WDT	8
PIC12C509	1024*12	41*8	4 MHZ	6		TMRO+WDT	8
PIC12C670	512*14	80*8	4 MHZ	6		TMRO+WDT	8
PIC12C671	1024*14	128*8	4 MHZ	6	2	TMRO+WDT	8
PIC12C272	2048*14	128*8	4 MHZ	6	2	TMRO+WDT	8
PIC12C680	512*12 FLASH	80*8 16*8 EEPROM	4 MHZ	6	4	TMRO+WDT	8
PIC12C681	1024*14 FLASH	80*8 16*8 EEPROM	4 MHZ	6		TMRO+WDT	8

TABLA 2

INSTRUCCIONES QUE MANEJAN REGISTROS				
SINTAXIS	OPERACIÓN	CICLOS	FORMATO 14 BITS	SEÑALIZADORES
ADDWF f,d	SUMA W y f	1	00 0111 dfff ffff	C,DC,Z
ADDWF f,d	AND W con f	1	00 0101 dfff ffff	Z
CLRF f	BORRA f	1	00 0001 1fff ffff	Z
CLRW -	Borra W	1	00 0001 0xxx xxxx	Z
COMF f,d	Complementa I (invierte)	1	00 1001 dfff ffff	Z
DECF f,d	Decrementa f	1	00 0011 dfff ffff	Z
INCF f,d	Incrementa I	1	00 1010 dfff ffff	Z
IORWF f,d	OR entre W y F	1	00 0100 dfff ffff	Z
MOVF f,d	Mueve I	1	00 1000 dfff ffff	Z
MOVW f	Mueve W y I	1	00 0000 1fff ffff	---
NOP ---	No opera	1	00 0000 0xx0 0000	---
RLF f,d	Rota I a la izqda. A través del acarreo	1	00 1101 dfff ffff	C
RRF f,d	Rota I a la dcha. a través del acarreo	1	00 1100 dfff ffff	C
SUBWF f,d	Resta W a f	1	00 0010 dfff ffff	C,DC,Z
SWAPF f,d	Intercambia nibbles	1	00 1110 dfff ffff	---
XORWF f,d	XOR de W con f	1	00 0110 dfff ffff	Z.

TABLA 3

INSTRUCCIONES QUE MANEJAN BITS				
SINTAXIS	OPERACIÓN	CICLOS	FORMATO 14 BITS	SEÑALIZADORES
BCF f,b	Borra bit de f	1	01 00bb bfff ffff	---
BSF f,b	Pone a 1 el bit f	1	01 01bb bfff ffff	---

TABLA 4

INSTRUCCIONES DE BRINCO				
SINTAXIS	OPERACIÓN	CICLOS	FORMATO 14 BITS	SEÑALIZADORES
BTFSC f,d	Explora un bit de f y brinca si vale 0	1(2)	01 10bb bfff ffff	---
BTFSS f,d	Explora un bit de f y brinca si vale 1	1(2)	01 11b bfff ffff	---
DECFSZ f,d	Decrementa f y si es 0, brinca	1(2)	00 1011 dfff ffff	---
INCFSZ f,d	Decrementa f y si es 1, brinca	1(2)	00 1111 dfff ffff	---

TABLA 5

INSTRUCCIONES QUE MANEJAN OPERANDOS INMEDIATOS				
SINTAXIS	OPERACION	CICLOS	FORMATO 14 bits	SENAIZADORES
ADDLW k	Suma inmediata con W	1	11 111x kkkk kkkk	C,DC,Z
ANDLW k	AND inmediato con W	1	11 1101 kkkk kkkk	Z
IORLW ik	OR inmediato con W	1	11 1000 kkkk kkkk	Z
MOVLW k	Mueve a W un valor inmediato	1	11 00xx kkkk kkkk	---
SUBLW k	Resta W de un inmediato	1	11 110x kkk kkkk	C,DC,Z
XORLW k	Or exclusiva con W	1	11 1010 kkkk kkkk	Z

TABLA 6

INSTRUCCIONES DE CONTROL Y ESPECIALES				
SINTAXIS	OPERACION	CICLOS	FORMATO 14 bits	SENAIZADORES
CALL k	Llamada a subrutina	2	10 0kkk kkkk kkkk	---
CLRWDT	Borra, refresca el Perro Guardián	1	00 0000 0110 0100	---
GOTO k	Salto incondicional	2	10 1kkk kkkk kkkk	---
RETFIE	Retorno de Interrupción (GIE=1)	2	00 0000 0000 (00)	---
RETLW k	Retorno subrutina y carga W=k	2	11 01xx kkkk kkkk	---
RETURN	Retorno de subrutina	2	00 0000 0000 1000	---
SLEEP	Pasa al modo de reposo	1	00 0000 0110 0011	TO#,PD#

TABLA 7

INDF	00h	INDF	80h	INDF	100h	INDF	180h
TMR0	01h	OPTION_REG	81h	TMR0	101h	OPTION_REG	181h
PCL	02h	PCL	82h	PCL	102h	PCL	182h
STATUS	03h	STATUS	83h	STATUS	103h	STATUS	183h
FSR	04h	FSR	84h	FSR	104h	FSR	184h
PORTA	05h	TRISA	85h		105h		185h
PORTB	06h	TRISB	86h	PORTB	106h	TRISB	186h
PORTC	07h	TRISC	87h		107h		187h
PORTD (1)	08h	TRISD (1)	88h		108h		188h
PORTE (1)	09h	TRISE (1)	89h		109h		189h
PCLATH	0Ah	PCLATH	8Ah	PCLATH	10Ah	PCLATH	18Ah
INTCON	0Bh	INTCON	8Bh	INTCON	10Bh	INTCON	18Bh
PIR1	0Ch	PIE1	8Ch	EEDATA	10Ch	EECON1	18Ch
PIR2	0Dh	PIE 2	8Dh	EEADR	10Dh	EECON2	18Dh
TMR1L	0Eh	PCON	8Eh	EEDATH	10Eh	Reservado	18Eh
TMR1H	0Fh		8Fh	EEADRH	10Fh	Reservado	18Fh
T1CON	10h		90h	Registros de	110h	Registes de	190h
TM2	11h	SSPCON2	91h	Prósito	111h	Proposito	191h
T2CON	12h	PR2	92h	General	112h	General	192h
SSPBUF	13h	SSPADD	93h	80 Bytes	113h	16 Bytes	193h
SSPCON	14h	SSPSTAT	94h		114h		194h
CCPR1L	15h		95h		115h		195h
CCPR1H	16h		96h		116h		196h
CCP1CON	17h		97h		117h		197h
RCSTA	18h	TXSTA	98h		118h		198h
TXREG	19h	SPBRG	99h		119h		199h
RCREG	1Ah		9Ah		11Ah		19Ah
CCPR2L	1Bh		9Bh		11Bh		19Bh
CCPR2H	1Ch		9Ch		11Ch		19Ch
CCP2CON	1Dh		9Dh		11Dh		19Dh
ADRESH	1Eh	ADRESL	9Eh		11Eh		19Eh
ADCON0	1Fh	ADCON1	9Fh		11Fh		19Fh
Registros de Propósitos General 96 Bytes	20 h	Registros de Propósitos General 80 Bytes	A0h	Registros de Propósitos General 80 Bytes	120h	Registros de Propósitos General 80 Bytes	1Eh
			Efh		16Fh		1Eh
		Mapeados con 70h-7Fh	F0h	Mapeados con 70h-7Fh	170h	Mapeados con 70h-7Fh	1F0h
	7 Fh		FFh		17Fh		1FFh
Banco 0		Banco 1		Banco 2		Banco 3	

TABLA 8

INDF	00h	INDF	80h	INDF	100h	INDF	180h
TMR0	01h	OPTION_REG	81h	TMR0	101h	OPTION_REG	181h
PCL	02h	PCL	82h	PCL	102h	PCL	182h
STATUS	03h	STATUS	83h	STATUS	103h	STATUS	183h
FSR	04h	FSR	84h	FSR	104h	FSR	184h
PORTA	05h	TRISA	85h		105h		185h
PORTB	06h	TRISB	86h	PORTB	106h	TRISB	186h
PORTC	07h	TRISC	87h		107h		187h
PORTD (1)	08h	TRISD (1)	88h		108h		188h
PORTE (1)	09h	TRISE (1)	89h		109h		189h
PCLATH	0Ah	PCLATH	8Ah	PCLATH	10Ah	PCLATH	18Ah
INTCON	0Bh	INTCON	8Bh	INTCON	10Bh	INTCON	18Bh
PIR1	0Ch	PIE1	8Ch	EEDATA	10Ch	EECON1	18Ch
PIR2	0Dh	PIE2	8Dh	EEADR	10Dh	EECON 2	18Dh
TMR1L	0Eh	PCON	8Eh	EEDATH	10Eh	Reservado	18Eh
TMR1H	0Fh		8Fh	EEADRH	10Fh	Reservado	18Fh
T1CON	10h		90h				190h
TMR2	11h	SSPCON2	91h				
T2CON	12h	PR2	92h				
SSPBUF	13h	SSPADD	93h				
SSPCON	14h	SSPSTAT	94h				
CCPR1L	15h		95h				
CCPR1H	16h		96h				
CCP1CON	17h		97h				
RCSTA	18h	TXSTA	98h				
TXREG	19h	SPBRG	99h				
RREG	1Ah		9Ah				
CCPR2L	1Bh		9Bh				
CCPR2H	1Ch		9Ch				
CCP2CON	1Dh		9Dh				
ADRESH	1Eh	ADRESL	9Eh				
ADCON0	1Fh	ADCON1	9Fh				
Registros de Propósitos general 96 Bytes	20 H	Registros de Propósito General 96 Bytes	A0h	Mapeados con 20h-7Fh	12h	Mapeados con A0h-FFh	1°0H
Banco 0		Banco 1		Banco 2		Banco 3	

TABLA 9

NEMÓNICOS	PARÁMETROS	OPERACIÓN	CICLOS	FORMATO 14 bits	SEÑALIZADORES
INSTRUCCIONES QUE MANEJAN REGISTROS					
addwf	f,d	SUMA de W con f	1	00 0111 dfff ffff	C,DC, Z
andwf	f,d	AND de W con f	1	00 0101 dfff ffff	Z
clrf	f,d	BORRADO de f	1	00 0001 1fff ffff	Z
clrw	ae	BORRADO de W	1	00 0001 0xxx xxxx	Z
comf	f,d	COMPLEMENTO de f	1	00 1001 dfff ffff	Z
decf	f,d	DECREMENTO de f	1	00 0011 dfff ffff	Z
incf	f,d	INCREMENTO de f	1	00 1010 dfff ffff	Z
iorwf	f,d	OR de W con f	1	00 0100 dfff ffff	Z
movf	f,d	MOVIMIENTO de f	1	00 1000 dfff ffff	Z
movwf	f,d	MOVIMIENTO de W a f	1	00 0000 1fff ffff	
nop	ae	NO OPERACIÓN	1	00 0000 0xx0 0000	
rlf	f,d	ROTACIÓN de f a izqd. con carry	1	00 1101 dfff ffff	C
rrf	f,d	ROTACIÓN de f a dcha con carry	1	00 1100 dfff ffff	C
subwf	f,d	RESTA de W a f (f-W)	1	00 0010 dfff ffff	C,DC,Z
swapf	f,d	INTERCAMBIO de 4 bits de peso por los 4 - peso	1	00 1110 dfff ffff	
xorwf	f,d	OR exclusiva de W con f	1	00 0110 dfff ffff	Z
INTRUCCIONES QUE MANEJAN BITS					
bcf	f,b	PUESTA a 0 del bit b de f	1	01 00bb bfff ffff	
bsf	f,b	PUESTA a 1 del bit b de f	1	01 01bb bfff ffff	
INTRUCCIONES DE BRINCO					
btfsc	f,b	TESTEO del bit b de f; BRINCO si 0			
btfss	f,b	TESTEO del bit b de f; BRINCO si 1			
decisz	f,d	DECREMENTO de f; BRINCO si 0			
incfsz	f,d	INCREMENTO de f; BRINCO si 0			
INTRUCCIONES QUE MANEJAN OPERANDOS INMEDIATOS					
addlw	k	SUMA de literal con W	1	11 111x kkkk kkkk	C,DC,Z
andlw	k	AND de literal con W	1	11 1101 kkkk kkkk	Z
iorlw	k	OR de literal con W	1	11 1000 kkkk kkkk	Z
movil	k	MOVIMIENTO de literal a W	1	11 00xx kkkk kkkk	
sublw	k	Resta de literal (K-W)	1	11 110x kkk kkkk	C,DC,Z
xorlw	k	OR exclusiva de literal con W	1	11 1010 kkkk kkkk	Z
INTRUCCIONES DE CONTROL Y ESPECIALES					
call	k	Llamada a subrutina	2	10 0kkk kkkk kkkk	---
clrwtd		Borrado de WAATCHDOG	1	00 0000 0110 0100	---
goto	k	SALTO a una direccion	2	10 1kkk kkkk kkkk	---
terfie		RETORNO de interrupcion	2	00 0000 0000 (00)	---
retlw	k	RETORNO devolviendo literal en W	2	11 01xx kkkk kkkk	---
return		RETORNO de subrutina	2	00 0000 0000 1000	---
sleep		PUESTA del microprocesador en reposo	1	00 0000 0110 0011	TO#,PD#

Tabla 10

Stock No.	Mfr.'s Type	Color/lens/Comments	Wave Length	(MCD) Typ.	Viewing Angle	EACH	
						1-49	50-up
505-2102	AND113R	Red/Red Clear	700	3.5	50	0.12	0.1
505-2104	AND114R	Red/Red Diffused	700	1.5	80	0.12	0.1
505-2117	AND114S	Red/Red Diffused	635	7	80	0.17	0.15
505-2101	AND113G	Green/Green Clear	565	17	50	0.14	0.12
505-2103	AND114G	Green/Green Diffused	565	5	80	0.14	0.12
505-5100	AND113Y	Yellow/Yellow Clear	585	9	50	0.18	0.16
505-5110	AND114Y	Yellow/Yellow Diffused	585	5	80	0.18	0.16

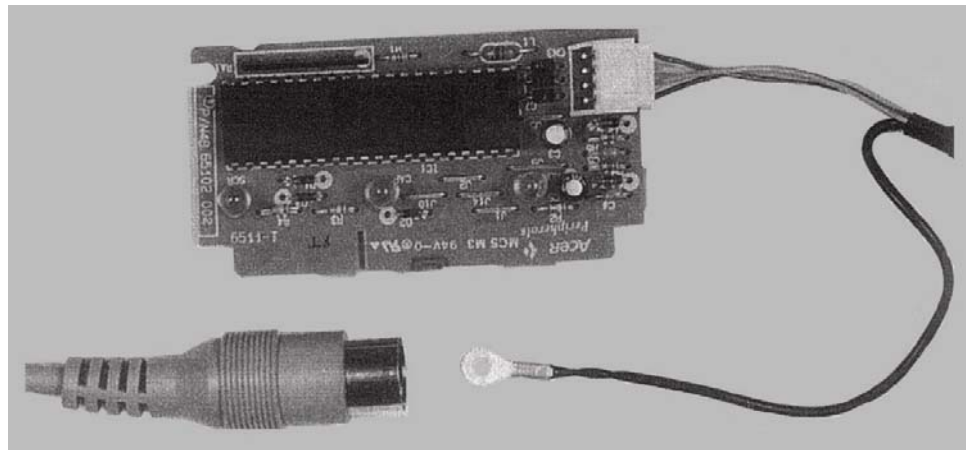


FIGURA 1

Fotografía de la tarjeta electrónica usada para el control del teclado de un computador. El único circuito integrado es un microcontrolador de INTEL que está encargado de detectar la tecla pulsada y transferir su código ASCII.

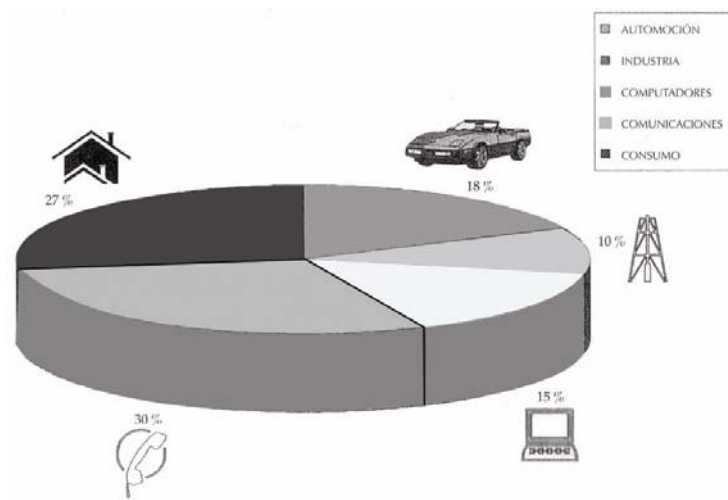


FIGURA 2
Gráfico que muestra la distribución de la producción mundial de microcontroladores en las diversas áreas de aplicación.

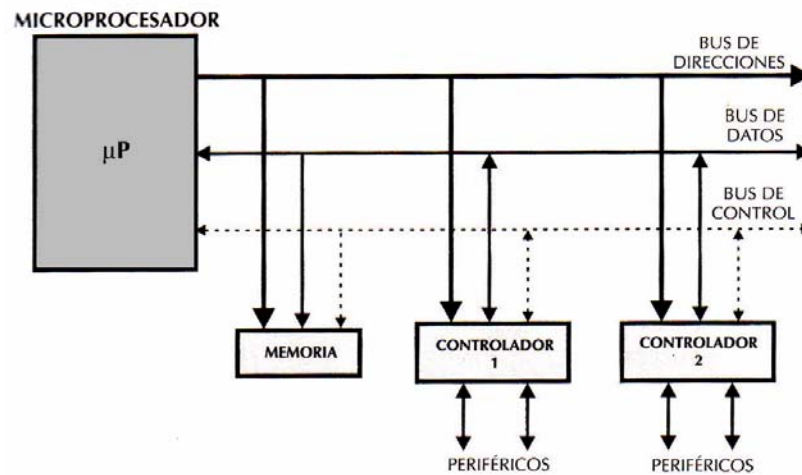


FIGURA 3

Estructura de un sistema abierto basado en un microprocesador. La disponibilidad de los buses en el exterior permite que se configure a la medida de la aplicación

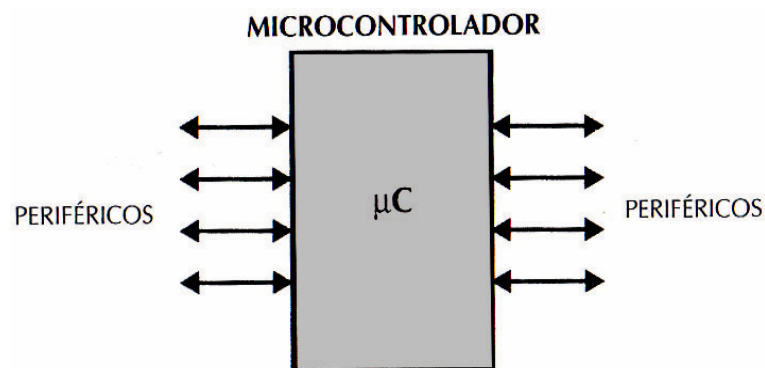


FIGURA 4

El microcontrolador en un sistema cerrado. Todas las partes del computador están contenidas en su interior y sólo salen al exterior las líneas que gobiernan los periféricos.

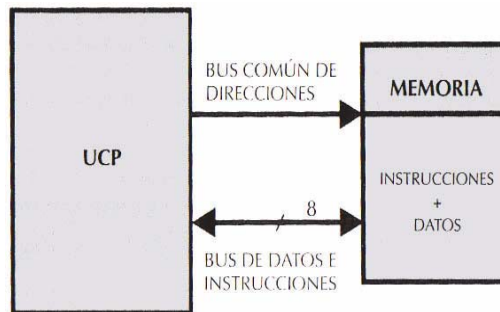


FIGURA 5

En la arquitectura de Von Neumann la UCP se comunicaba a través de un sistema de buses con la Memoria , donde se guardaban las instrucciones y los datos.

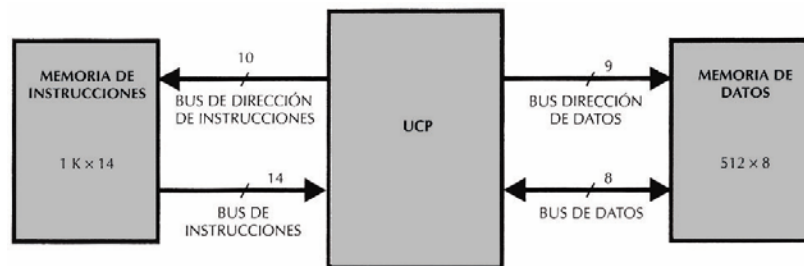


FIGURA 6

En la arquitectura Harvard la memoria de instrucciones y la de datos son independientes, lo que permite optimizar sus características y propiciar el paralelismo. En la figura, la memoria de instrucciones tiene 1 K posiciones de 14 bits cada una, mientras que la de datos sólo dispone de 512 posiciones de un byte.



FIGURA 9

*Fotografía de un emulador en circuito para microcontroladores es PIC.
Se trata del modelo ICEPIC de la empresa R.F. Solutions*

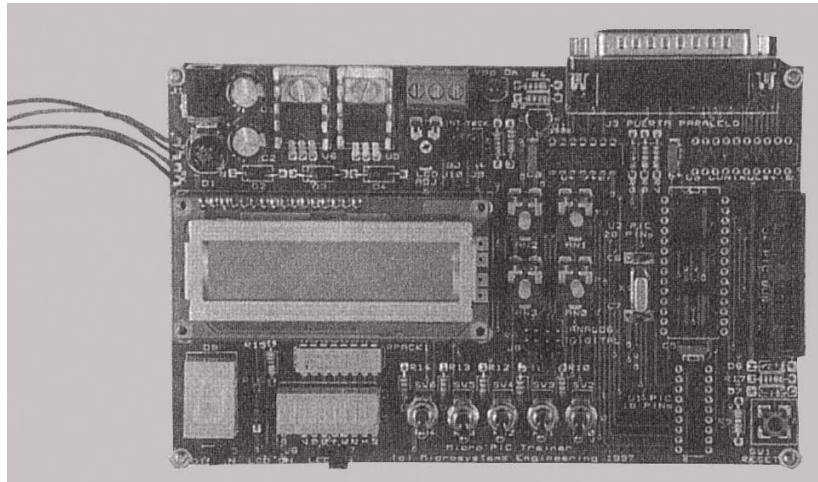


FIGURA 10
Fotografía del sistema de desarrollo MICRO'PIC TRAINER..

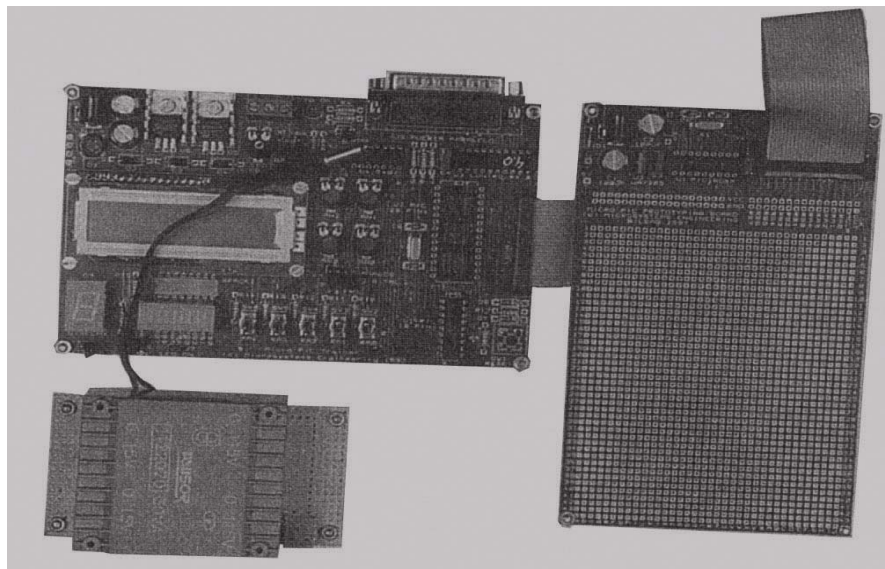


FIGURA 11
Tarjeta para soportar prototipos basados en el PIC16X84 y conectarse al MICRO'PIC TRAINER.

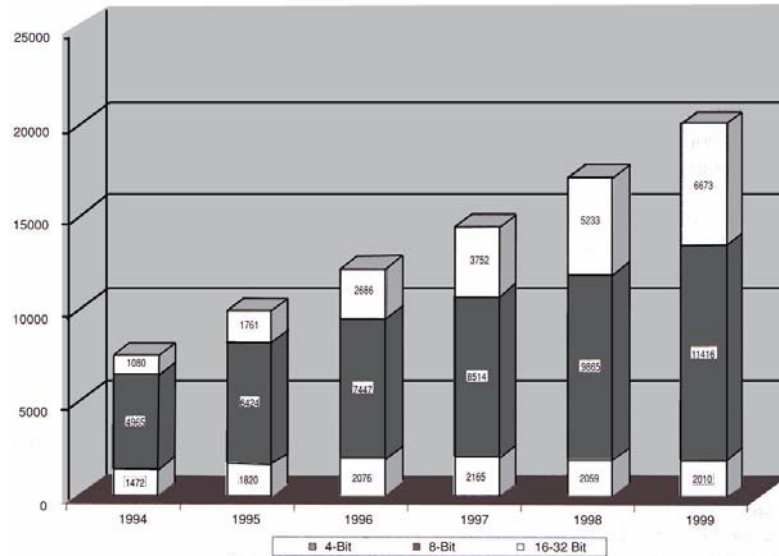


FIGURA 12

Distribución, por segmentos, del mercado mundial de microcontroladores. Las ventas de los de 8 bits superan al resto ampliamente.

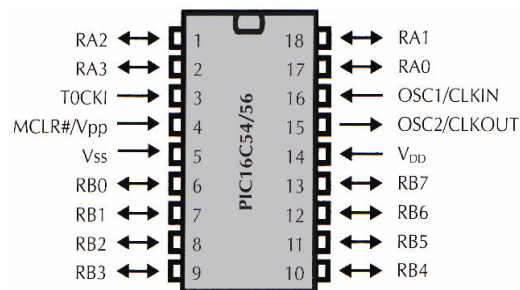


FIGURA 13

Diagrama de patitas de los PIC de la gama baja que responden a la nomenclatura PIC16C54/56

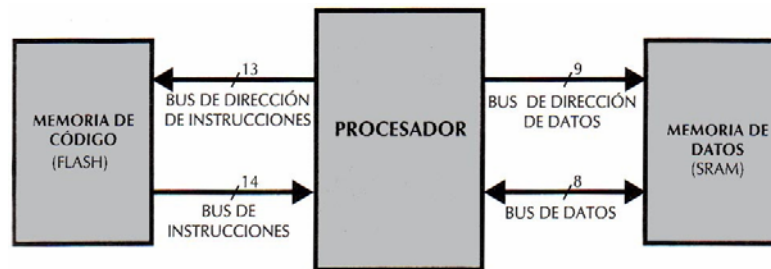


FIGURA 14

Los buses para instrucciones y datos son totalmente independientes y se ajustan a las necesidades de cada memoria, permitiendo el acceso simultáneo.

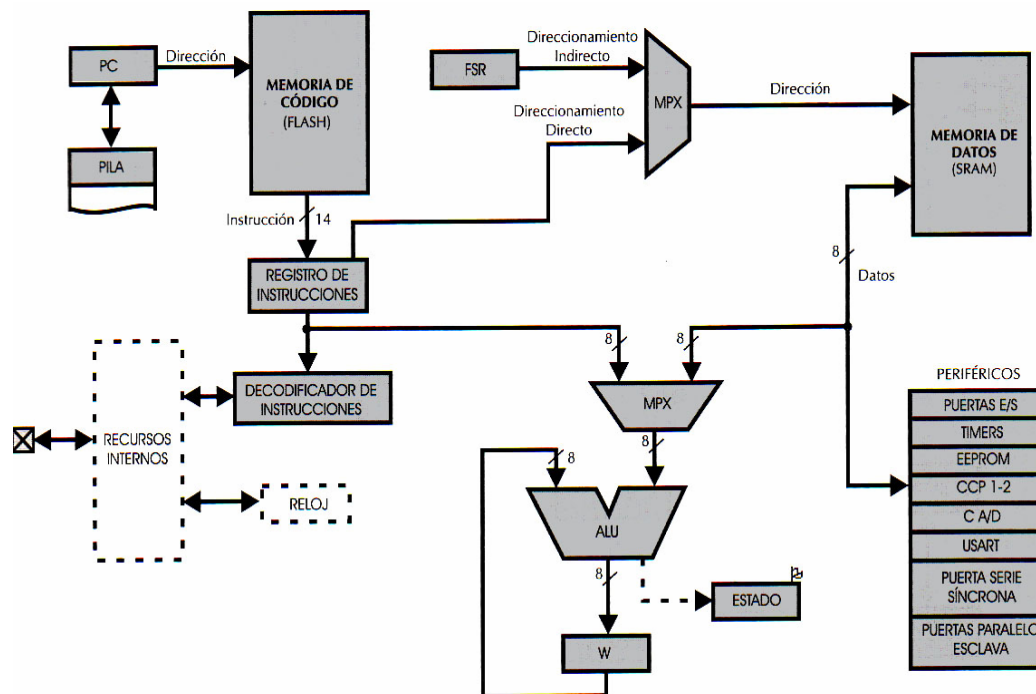


FIGURA 15

Arquitectura tipo Harvard simplificada para los microcontroladores PIC16F87X.

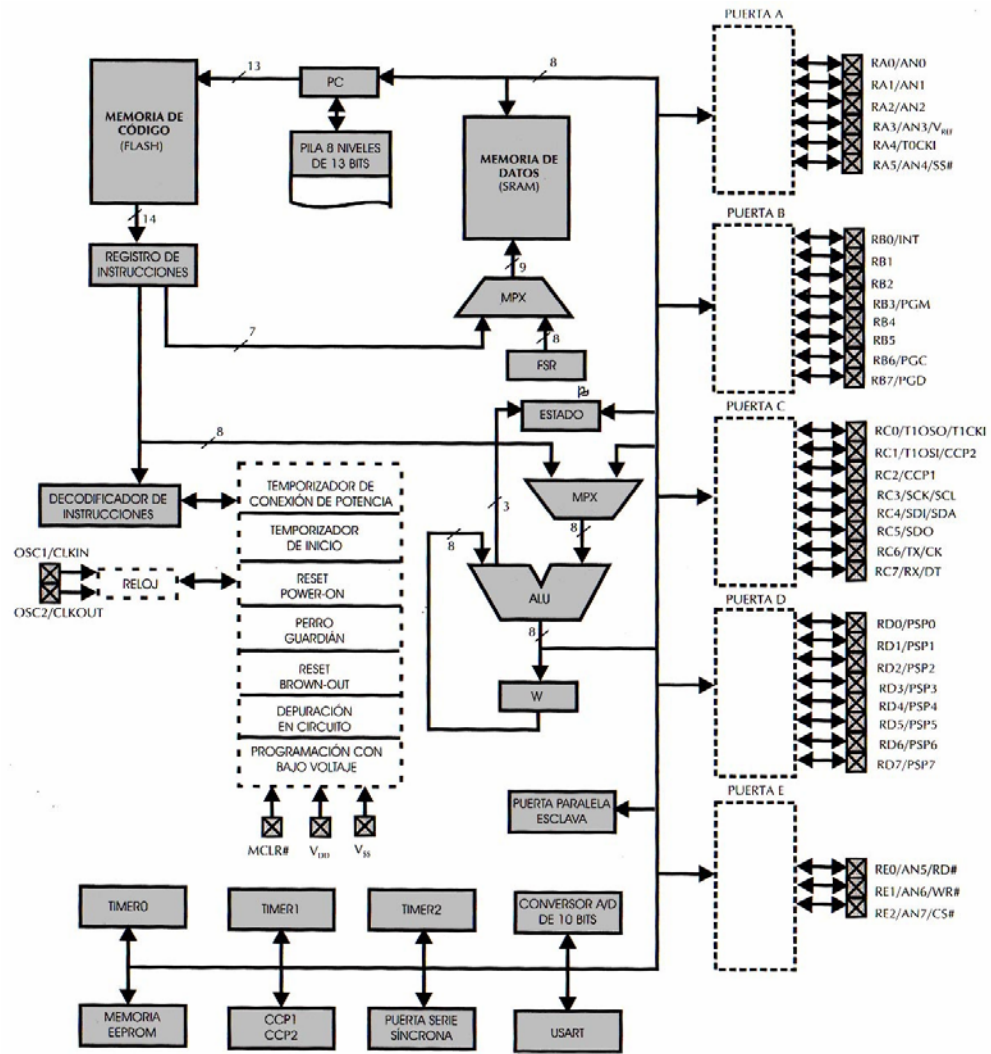


FIGURA 16

Arquitectura de los PIC16F87/7 de 40 patitas. Para los modelos de 28 patitas la arquitectura sólo difiere en que carecen de las Puertas de E/S D y E.

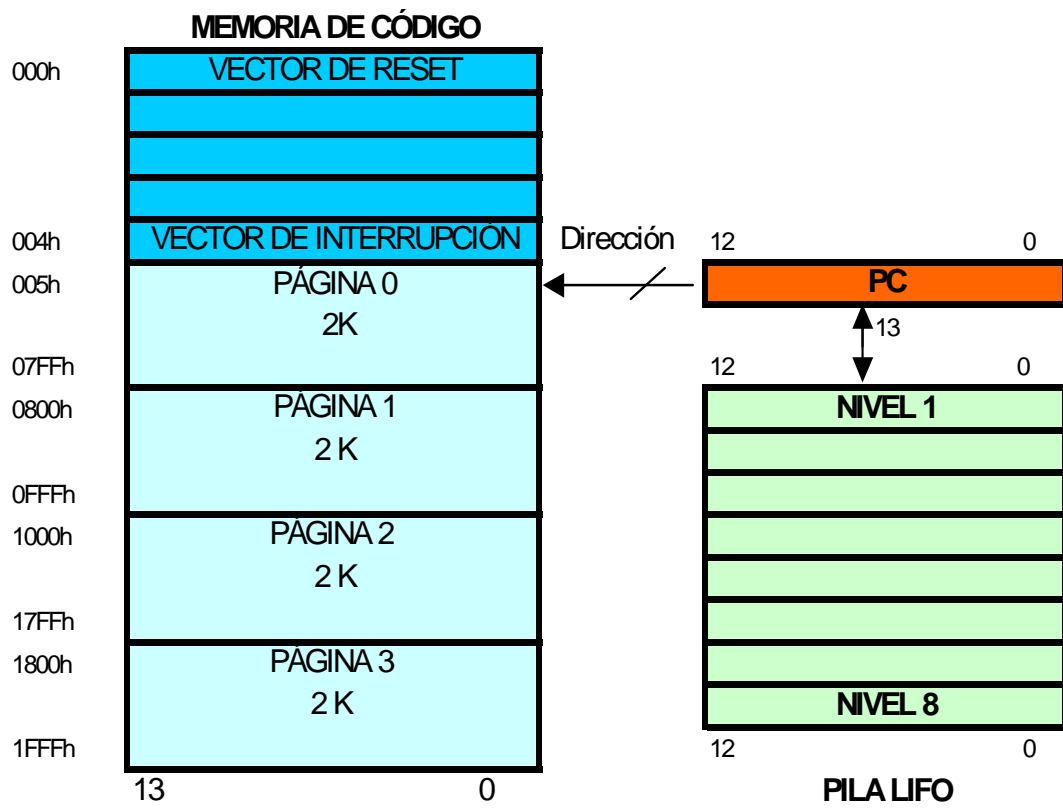


FIGURA 17
Organización de la memoria de programa tipo FLASH en los PIC16F87x

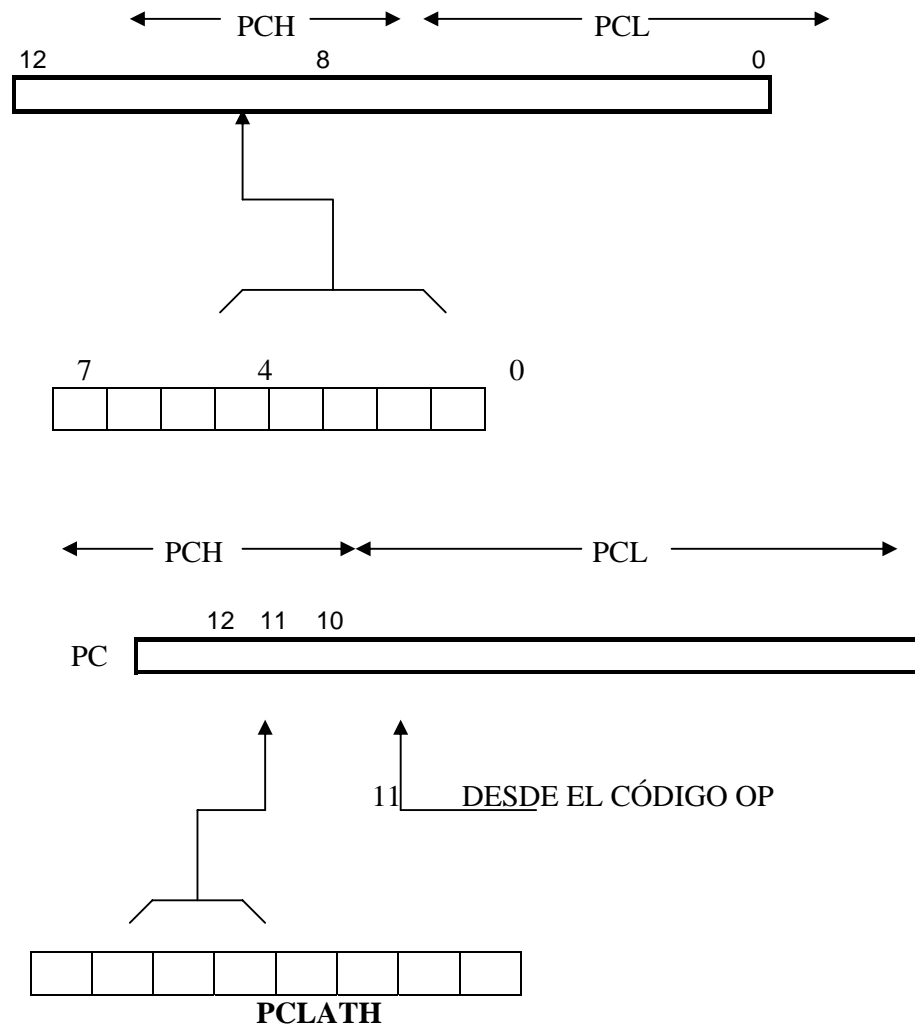


FIGURA 18

En la figura superior se muestra cómo se carga el PC. Abajo se muestra la carga del PC con las instrucciones CALL y GOTO.

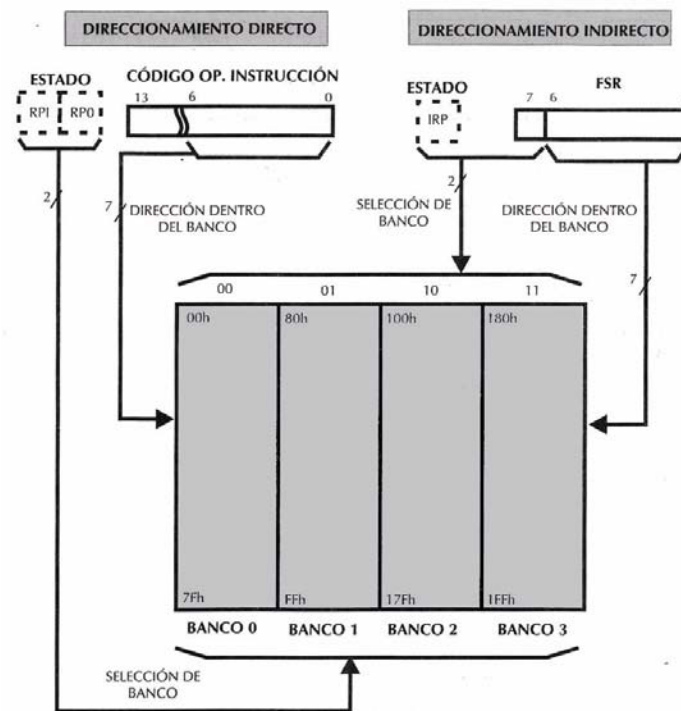


FIGURA 19

Formas de seleccionar el banco y la dirección e la memoria RAM en los direccionamientos directo e indirecto.

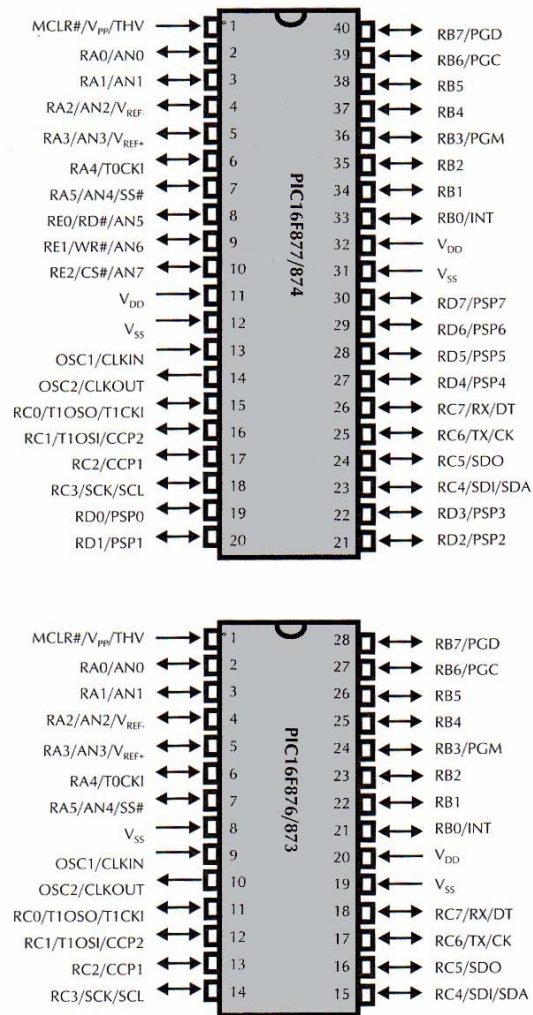


FIGURA 20

Diagrama de asignación y conexión de las patitas de los dos posibles encapsulados en los modelos de la subfamilia

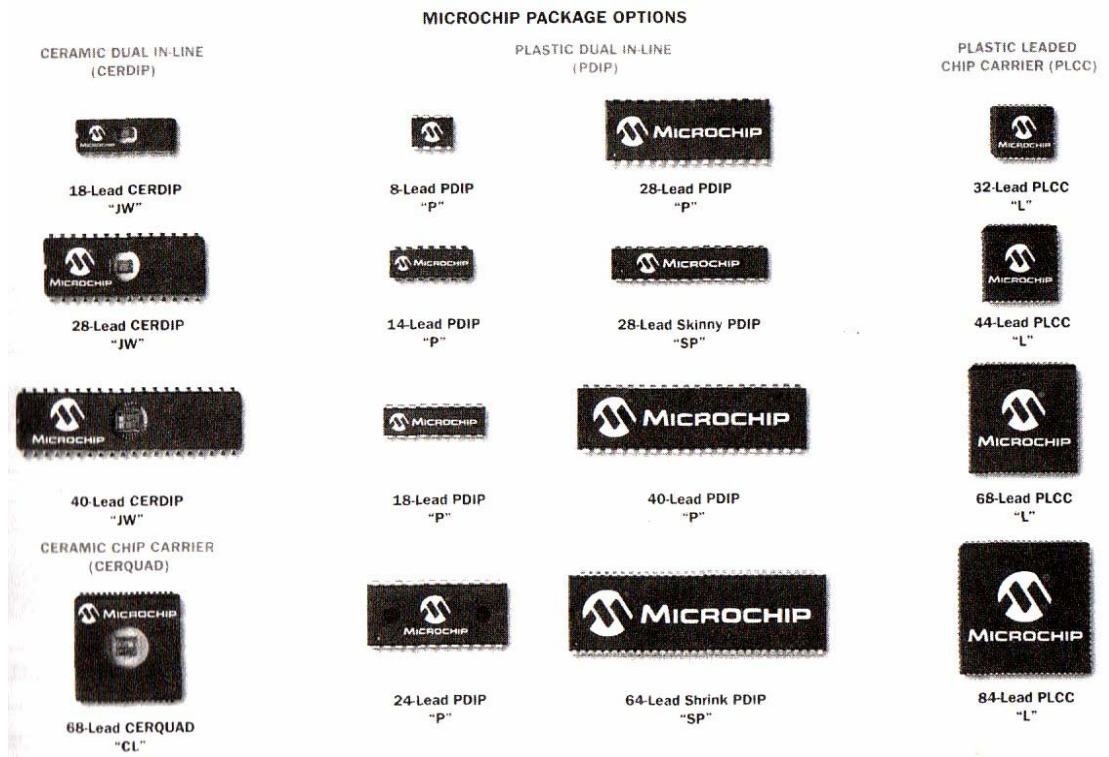


FIGURA 21
Opciones de encapsulados para los microcontroladores PIC. (Cortesía de Microchip).

**FIGURA UTILIZADA PARA DEFINIR EL TAMAÑO DE
LOS DISPLAYS DEL JUEGO DE MARCADORES
ELECTRONICOS.**

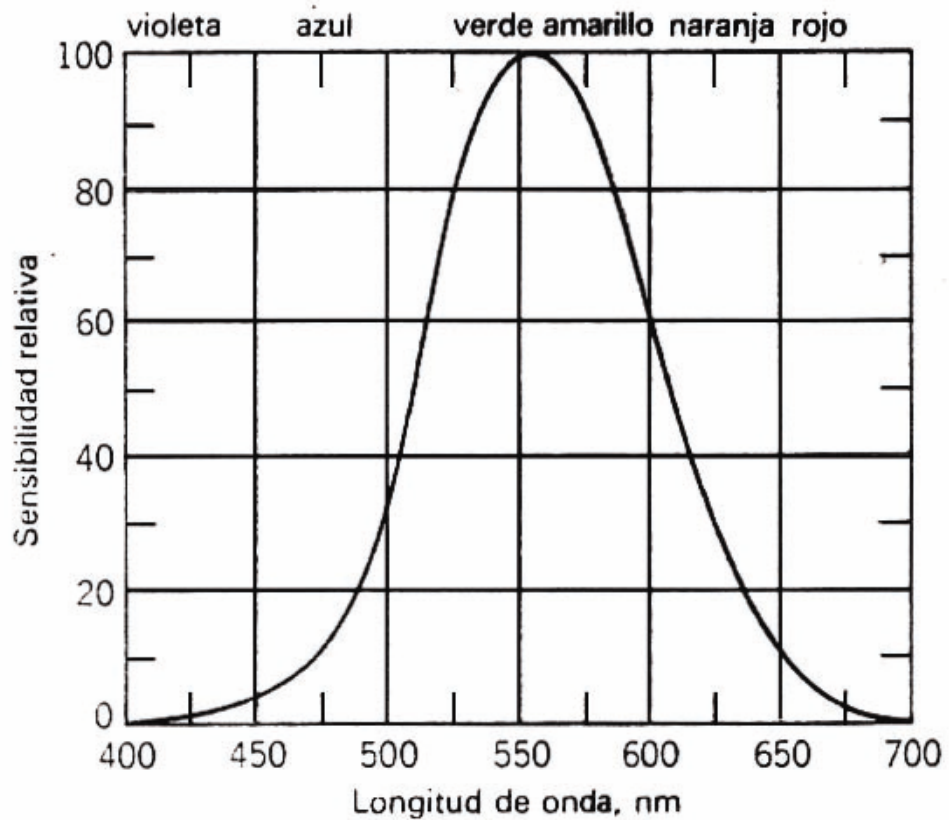
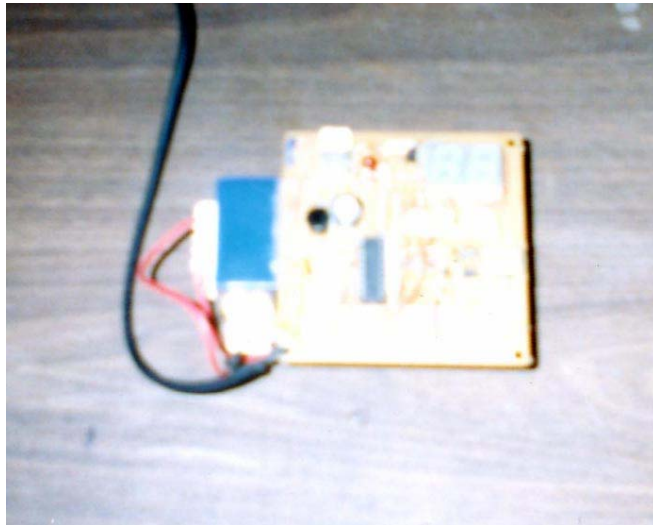


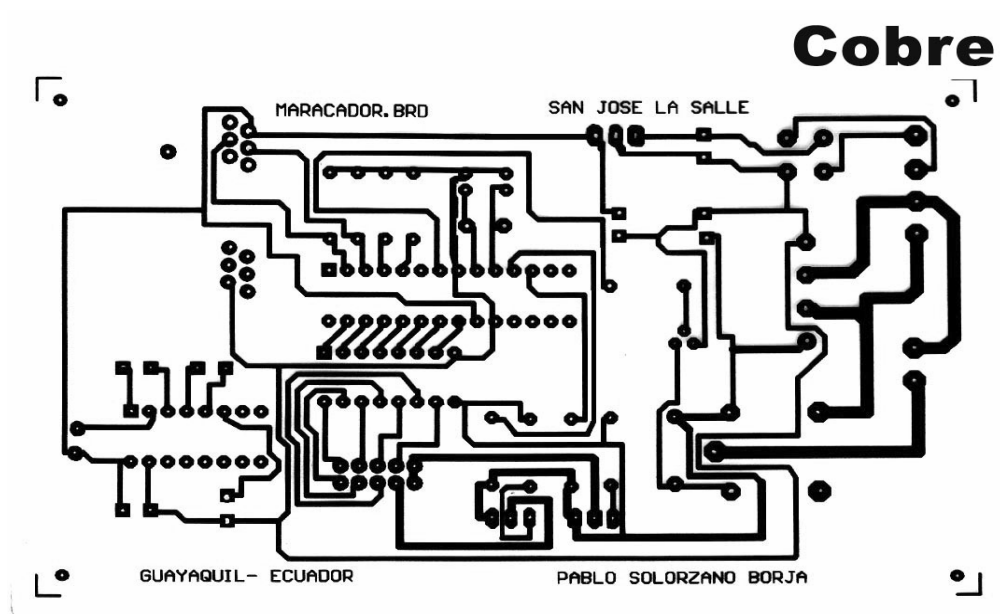
FIGURA 22

Sensibilidad relativa del ojo de un supuesto observador patrón para longitudes de onda diferentes y un nivel de iluminación específico.

***FOTOGRAFÍAS QUE DESCRIBEN LA SECUENCIA
COMPLETA DE FABRICACIÓN DE UN JUEGO DE
MARCADORES COMPLETAMENTE PROGRAMABLES.***

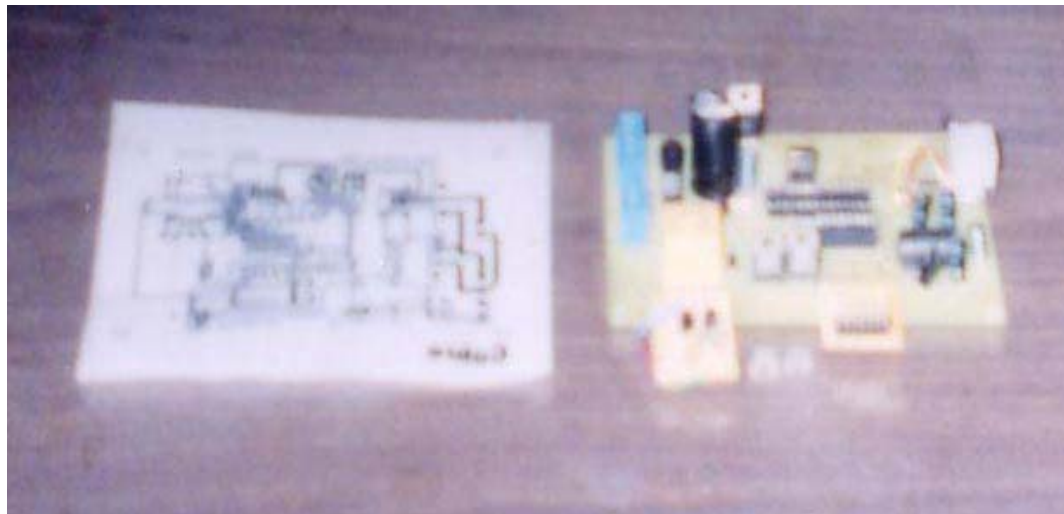


FOTOGRAFÍA 1
Construcción de prototipo utilizando el PIC16F84 y Display de Anodo común.



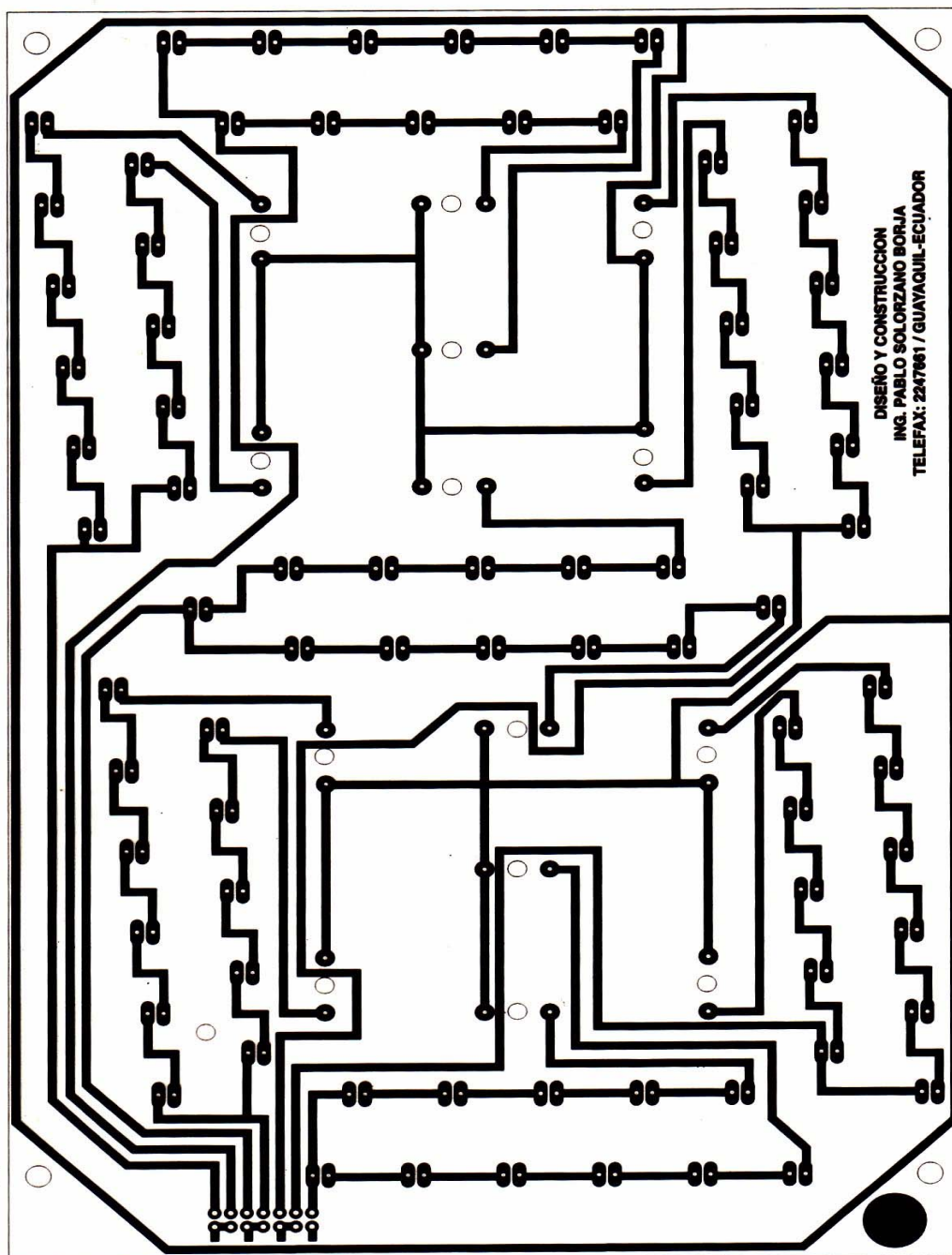
FOTOGRAFIA 2

*Diseño de tarjeta controladora utilizando el programa EAGLE
de la
CAD SOFT*



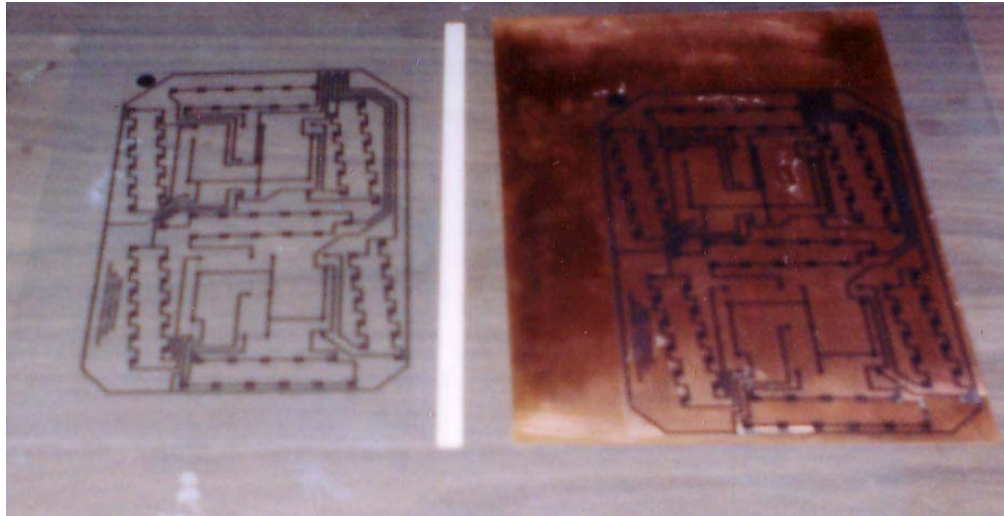
FOTOGRAFIA 3

Dibujo de circuito impreso y tarjeta controladora



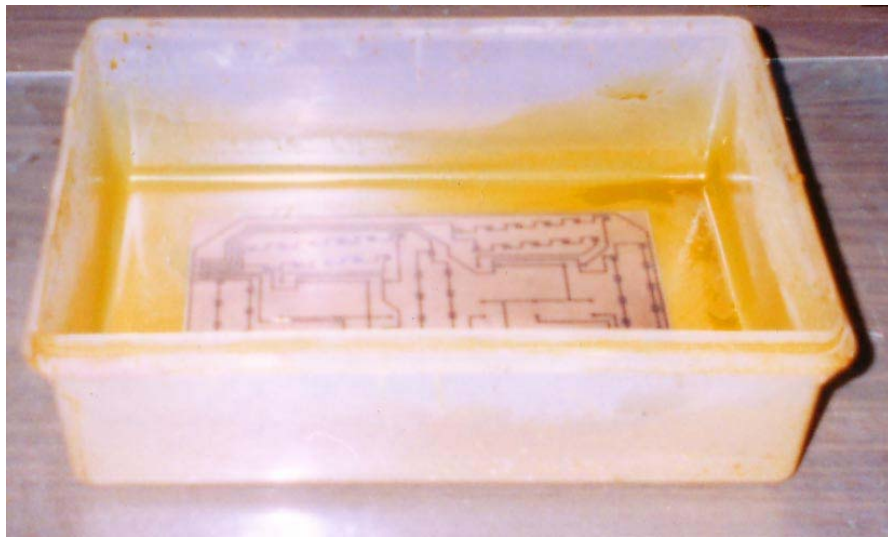
FOTOGRAFIA 4

Acetato del circuito impreso para la fabricación de un display con 78 leds, este se estampara en la baquelita utilizando un transfer en caliente.



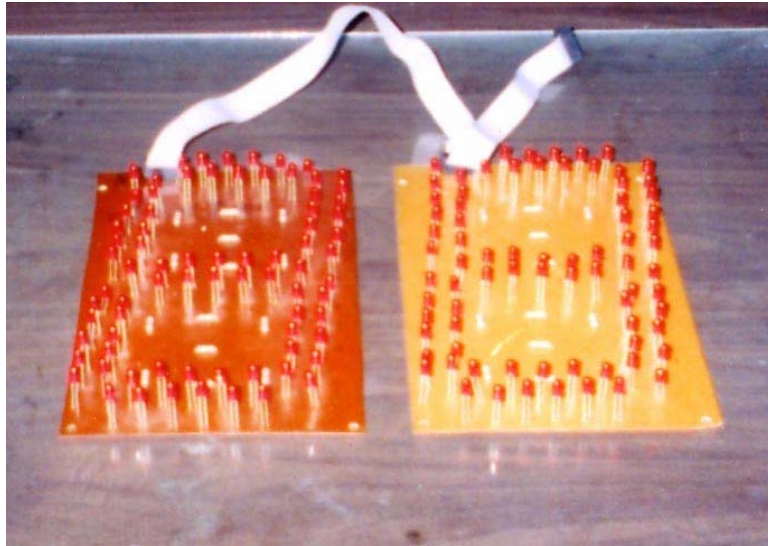
FOTOGRAFIA 5

Sistema de estampado en caliente utilizando un transfer sacado de un acetato en baquelita tipo FR-2



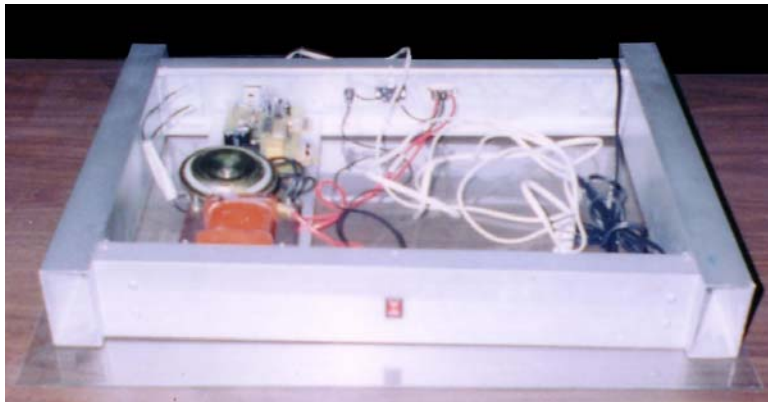
FOTOGRAFIA 6

Proceso de desnudación utilizando una solución de percloruro férrico para obtener el circuito impreso para las tarjetas de los displays.



FOTOGRAFÍA 7

Displays del marcador fabricados con baquelita tipo FR-2 con 156 LEDS en arreglo de 7 segmentos siguiendo formulación descrita en el apartado 5.5



FOTOGRAFÍA 8

Montaje del transformador en la estructura de aluminio, con los demás componentes que integran el circuito y conexiones del marcador.



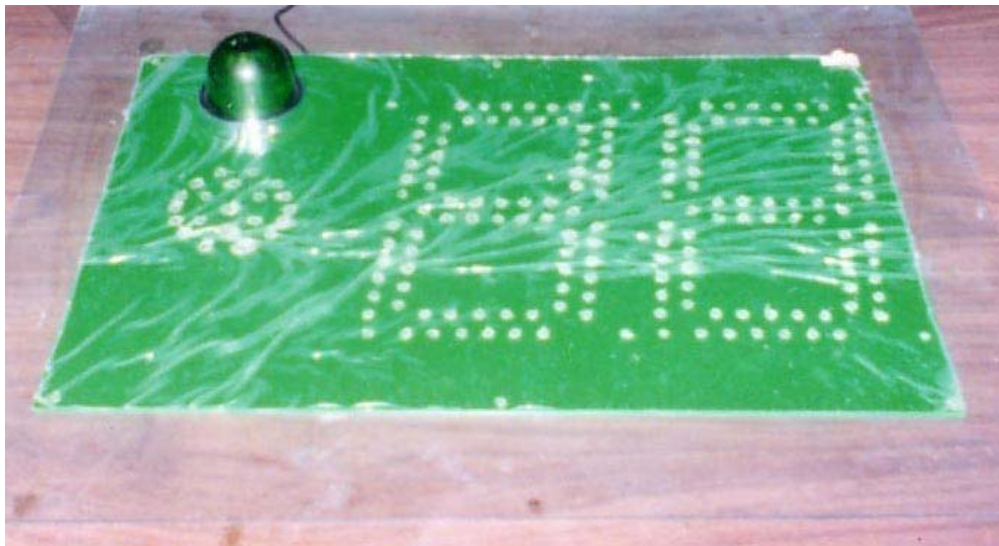
FOTOGRAFÍA 9

Perfilaría de aluminio que constituye la estructura con la cual está fabricada el marcador.



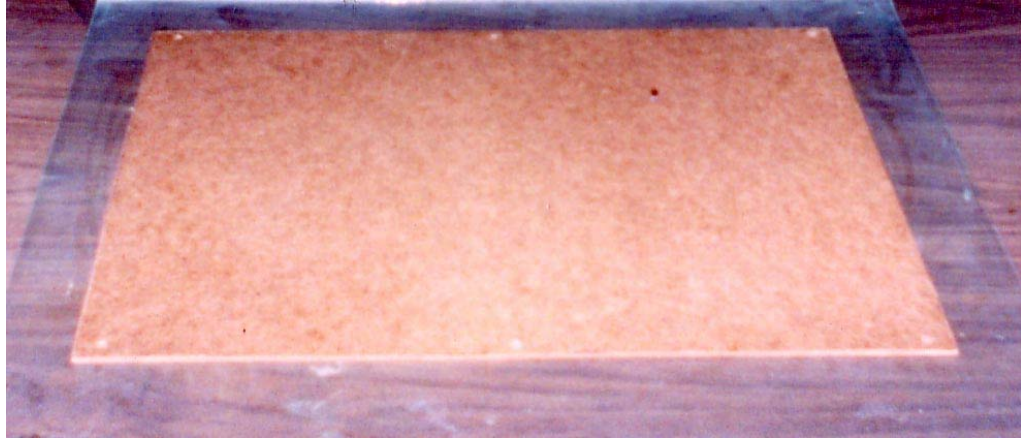
FOTOGRAFÍA 10

Molde en hierro gris de 4mm de espesor utilizado en la fabricación de la mascarilla del marcador.



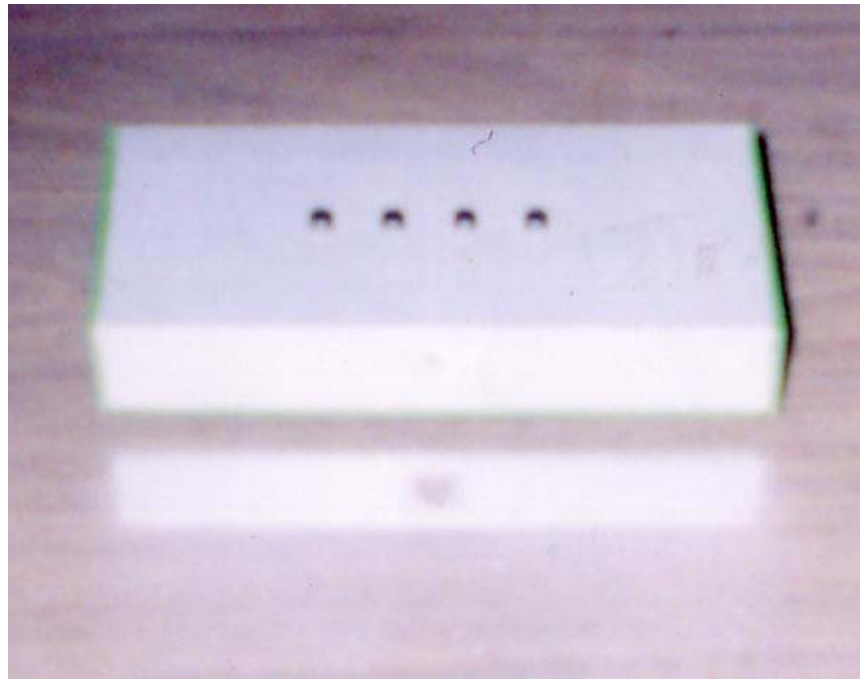
FOTOGRAFÍA 11

Mascarilla de los marcadores utilizando para su confección lámina acrílica de 2 mm de espesor.



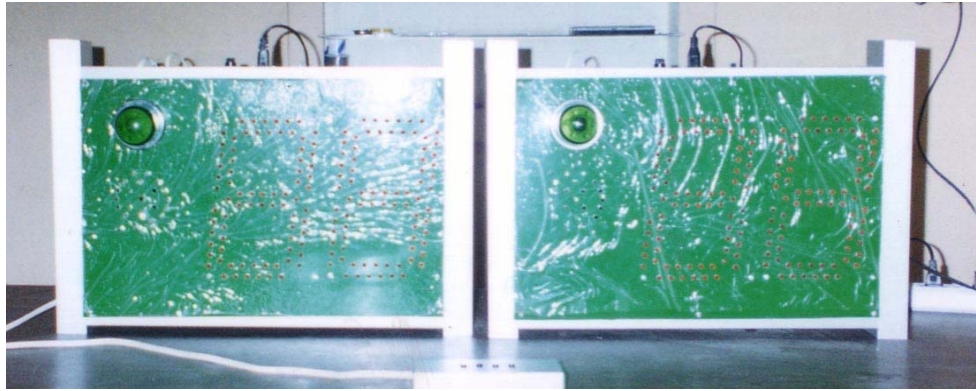
FOTOGRAFÍA 12

Tapa posterior del marcador fabricada en cartón prensado de 4 mm de espesor.



FOTOGRAFÍA 13

Control maestro con circuito y botoneras terminado fabricado en perfil de aluminio y acrílico..



FOTOGRAFÍA 14
Juego de marcadores y control maestro terminados.

BIBLIOGRAFÍA

MICROCONTROLADORES:

**DISEÑO PRÁCTICO DE APLICACIONES 1 Y 2 PARTE POR JOSE M.
ANGULO**

SUSANA ROMERO YESA

IGNACIO ANGULO MARTINEZ

DISEÑOS Y CIRCUITOS DIGITALES DE FLETCHER

ELECTRÓNICA DE BOYLESTAD

ELECTRÓNICA DE MILMAN

FÍSICA DE RESNICK- HALLIDAY TOMO II