Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ingeniería Mecánica Eléctrica

DISEÑO DE CIRCUITO ELECTRÓNICO CON MICROCONTROLADOR, POR MEDIO DE CONTROL NUMÉRICO COMPUTARIZADO, PARA COMUNICAR LUZ PILOTO DE LAS VIDEOCÁMARAS CON SWITCHER

William Leonardo Martínez Contreras
Asesorado por el Ing. Enrique Edmundo Ruiz Carballo

Guatemala, enero 2009

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

FACULTAD DE INGENIERÍA

DISEÑO DE CIRCUITO ELECTRÓNICO CON MICROCONTROLADOR, POR MEDIO DE CONTROL NUMÉRICO COMPUTARIZADO, PARA COMUNICAR LUZ PILOTO DE LAS VIDEOCÁMARAS CON SWITCHER

TRABAJO DE GRADUACIÓN

PRESENTADO A LA JUNTA DIRECTIVA DE LA
FACULTAD DE INGENIERÍA
POR:

WILLIAM LEONARDO MARTÍNEZ CONTRERAS

ASESORADO POR EL ING. ENRIQUE EDMUNDO RUIZ CARBALLO

CONFERÍRSELE EL TÍTULO DE

INGENIERO ELECTRÓNICO

GUATEMALA, ENERO DE 2009

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

FACULTAD DE INGENIERÍA

NÓMINA DE JUNTA DIRECTIVA

DECANO	Ing. Murphy Olympo Paiz Recinos
VOCAL I	Inga. Glenda Patricia García Soria
VOCAL II	Inga. Alba Maritza Guerrero de López
VOCAL III	Ing. Miguel Ángel Dávila Calderón
VOCAL IV	Br. José Milton De León Bran
VOCAL V	Br. Isaac Sultán Mejía
SECRETARIA	Inga. Marcia Ivonne Véliz Vargas

TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO

DECANO	Ing. Murphy Olympo Paiz Recinos
EXAMINADOR	Ing. Enrique Edmundo Ruiz Carballo
EXAMINADOR	Ing. Gustavo Adolfo Villeda
EXAMINADOR	Ing. Luis Duran Córdova
SECRETARIA	Inga. Marcia Ivonne Véliz Vargas

HONORABLE TRIBUNAL EXAMINADOR

Cumpliendo con los preceptos que establece la ley de la Universidad de San Carlos de Guatemala, presento a su consideración mi trabajo de graduación titulado:

DISEÑO DE CIRCUITO ELECTRÓNICO CON MICROCONTROLADOR, POR MEDIO DE CONTROL NUMÉRICO COMPUTARIZADO, PARA COMUNICAR LUZ PILOTO DE LAS VIDEOCÁMARAS CON SWITCHER,

tema que me fuera asignado por la Dirección de la Escuela de Ingeniería Mecánica Eléctrica, con fecha 26 de octubre de 2007.

William Leonardo Martínez Contreras

UNIVERSIDAD DE SAN CARLOS
DE GUATEMALA

FACULTAD DE INGENIERIA

Guatemala, 19 de mayo de 2008

Ingeniero
Julio César Solares Peñate
Coordinador Área de Electrónica
Escuela de Ingeniería Mecánica Eléctrica.

Estimado Ingeniero:

Por este medio le informo que he revisado el trabajo de graduación titulado: **Diseño de circuito electrónico con microcontrolador por medio de control numérico computarizado para comunicar luz piloto de videocámaras con switcher,** elaborado por el estudiante William Leonardo Martínez Contreras.

El mencionado trabajo llena los requisitos para dar mi aprobación e indicarle que el autor y mi persona somos responsables por el contenido y conclusiones de la misma.

Atentamente,

Ing. Enrique Edmundo Ruiz Carballo
ASESOR

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

FACULTAD DE INGENIERIA

Guatemala, 4 de julio 2008.

Señor Director
Ing. Mario Renato Escobedo Martínez
Escuela de Ingeniería Mecànica Elèctrica
Facultad de Ingeniería, USAC.

Señor Director:

Me permito dar aprobación al trabajo de Graduación titulado: DISEÑO DE CIRCUITO ELECTRÒNICO CON MICROCONTROLADOR POR MEDIO DE CONTROL NUMÈRICO COMPUTARIZADO PARA COMUNICAR LUZ PILOTO DE LAS VIDEOCÀMARAS CON SWITCHER, del estudiante: William Leonardo Martìnez Contreras, por considerar que cumple con los requisitos establecidos para tal fin.

Sin otro particular, aprovecho la oportunidad para saludarle.

Atentamente,

ID Y ENSEÑAD A TODOS

Ing. [illegible] César Solares Peñate
Coordinador Area de Electrònica

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
FACULTAD DE INGENIERIA
DIRECCION ESCUELA DE INGENIERIA MECANICA ELECTRICA

JCSP/sro

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

FACULTAD DE INGENIERIA

El Director de la Escuela de Ingeniería Mecànica Elèctrica, después de conocer el dictamen del Asesor, con el Visto Bueno del Coordinador de Área, al trabajo de Graduación del estudiante: WILLIAM LEONARDO MARTÌNEZ CONTRERAS, titulado: DISEÑO DE CIRCUITO ELECTRÒNICO CON MICROCONTROLADOR POR MEDIO DE CONTROL NÙMÈRICO COMPUTARIZADO PARA COMUNICAR LUZ PILOTO DE LAS VIDEOCÀMARAS CON SWITCHER, procede a la autorización del mismo.

Ing. Mario Renato Escobar Martínez
DIRECTOR

UNIVERSIDAD DE SAN CARLOS
FACULTAD DE INGENIERIA
DIRECCION ESCUELA DE INGENIERIA
MECANICA ELECTRICA
DE GUATEMALA

GUATEMALA, 07 DE JULIO 2,003.

Universidad de San Carlos
de Guatemala

Facultad de Ingeniería
Decanato

Ref. DTG. 005.2009

El Decano de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer la aprobación por parte del Director de la Escuela de Ingeniería Mecánica Eléctrica, al trabajo de graduación titulado: **DISEÑO DE CIRCUITO ELECTRÓNICO CON MICROCONTROLADOR, POR MEDIO DE CONTROL NUMÉRICO COMPUTARIZADO, PARA COMUNICAR LUZ PILOTO DE LAS VIDEOCÁMARAS CON SWITCHER,** presentado por el estudiante universitario **William Leonardo Martínez Contreras,** autoriza la impresión del mismo.

IMPRÍMASE.

Ing. Murphy Olympo Paiz Recinos
DECANO

Guatemala, enero de 2009

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
DECANO
FACULTAD DE INGENIERIA

/gdech

AGRADECIMIENTOS A:

MI DIOS: Quien me otorgó todas las herramientas necesarias para culminar este pasaje de mi vida, enseñándome que todo lo podemos con Él.

A MIS PADRES: Por darme el apoyo incondicional, su ejemplo, su amor, y fuerza para enfrentar todos los obstáculos que nos pone la vida día a día.

HERMANOS: Por tenerme paciencia en todos estos años.

MIS ABUELOS Y TIOS: También influyeron mucho en mi educación y en mi vida.

MI ASESOR: Gracias por el tiempo que me brindó para la realización de este trabajo, siendo un honor que haya sido mi asesor conociendo la calidad de catedrático y persona que es.

MIS AMIGOS: Quienes conocí a lo largo de la carrera y en el trabajo, gracias por ser parte de su vida.

DEDICATORIA A:

MI TÍA	**En memoria de una gran mujer, quien me apoyo y me dio ánimos en mis momentos de flaqueza. Encontré no sólo una tía, si no que una gran amiga. Que Dios te guarde, tía Olga, hasta pronto.**

ÍNDICE GENERAL

ÍNDICE DE ILUSTRACIONES

FIGURAS

TABLAS

LISTA DE SÍMBOLOS

$	Número hexadecimal
%	Número binario
0x	Número hexadecimal
A	Corriente
ALU	*Aritmetic logic unit*: unidad de aritmética lógica.
AND	Circuito lógico - Función lógica
ASCII	Código americano normalizado para el intercambio de información ASPL Lógica programable de aplicación específica
BCD	Decimal codificado en binario
Bit	*Binary Digit*: digito binario
BIT	Dígito binario - Puede tomar dos valores 0 ó 1
BYTE	Octeto, carácter compuesto de ocho bits
CAD	Diseño Asistido por Ordenador.
CAM	Fabricación asistida por ordenador.
CD	Disco compacto - Detección de portadora
CLK	Impulsos, señal de reloj.
CPU	*Central Processor Unit*: unidad central de procesamiento
DAC	Conversión analógica digital
E/S	Entrada o salida.
EEPROM	Memoria de solo lectura programable y borrable
EPROM	Memoria de solo lectura grabable y borrable
FIFO	Primero en entrar primero en salir
Hz	Hertz
I^2 L	Lógica de inyección integrada
Icc	Corriente de alimentación
LIFO	Ultimo en entrar primero en salir

LSB	Bit menos significativo
Mbps	Megabytes por segundo
MSB	Bit más significativo
NAND	AND negada - Circuito lógico - Función lógica
NOR	OR negada - Circuito lógico - Función lógica
NO-Y	Circuito lógico - Función lógica
PC	Contador de programa - PC Computador personal
PIC	Circuito integrado programable.
PIC	Circuito integrado programable
PROM	Memoria de solo lectura grabable una vez
Q	Salida lógica
RAM	*Random acces memory*: memoria de acceso aleatorio.
ROM	Memoria de solo lectura
RS-232	Norma de comunicación serie
RxD o RD	Recepción de datos
SW	*Switch* - Conmutador
Sync	Sincronismos
TxD o TD	Transmisión de datos
V	Voltio
Vref	Tensión de referencia
X-OR	exclusiva - Circuito lógico - Función lógica

GLOSARIO

Acoplador Dispositivo eléctrico o electrónico que permite acoplar dos circuitos, sea para reducir las pérdidas, para bloquear o discriminar alguna de sus señales, u obtener alguna muestra de señal.

Aislante Material o sustancia que presenta una conductividad eléctrica casi nula, debido a que los electrones de sus átomos están fuertemente ligados al núcleo, evitando su movimiento.

Amperímetro Aparato medidor de la intensidad de corriente eléctrica que recorre un circuito; se utiliza colocándolo en serie dentro del mismo, al contrario de lo que sucede con un voltímetro, que se coloca en paralelo. Los primeros amperímetros eran analógicos, y constaban de un galvanómetro cuya resistencia era muy pequeña, con el fin de interferir lo menos posible en el propio circuito que se deseaba medir. Los modernos amperímetros digitales tienen una resistencia interna prácticamente despreciable, lo cual permite medir corrientes muy pequeñas sin modificar las características del circuito en prueba.

Amperio Unidad de la intensidad de la corriente eléctrica cuyo símbolo es "A". Esta unidad está definida en el Sistema Internacional (SI) como la intensidad de corriente eléctrica constante que, mantenida entre dos conductores paralelos, rectilíneos, de longitud infinita, de sección circular despreciable y colocados en el vacío a una distancia de un metro el uno del otro, produce entre estos conductores una fuerza igual a 2×10^{-7} newton por cada metro de longitud. 1 amperio equivale a 1 Culombio por segundo.

Amplificación Acción de incrementarse el valor de un voltaje o señal eléctrica. Es la acción contraria de la atenuación.

Bidireccional Se dice de la doble dirección en que puede moverse una corriente eléctrica o un flujo de datos. Si la corriente sólo puede circular en un único sentido se dice que es unidireccional, por ejemplo como sucede en un diodo.

Binario En electrónica digital, código de numeración en base dos, utilizado en el tratamiento de la información por medio de señales eléctricas. Los dos únicos valores que puede tomar son el 0 y el 1.

Cable Alambre con propiedades conductoras de la energía eléctrica. Está formado por uno o varios conductores constituidos a su vez por varios hilos elementales de cobre; también pueden ser de aluminio pero se utilizan raramente. Todo ello va protegido por una cubierta aislante flexible y normalmente impermeable.

Canal Se dice normalmente de una vía o banda de frecuencias asignada.

Capacidad Razón entre el valor absoluto de la carga de una de las armaduras y la diferencia de potencial existente entre ellas. La unidad de capacidad en el Sistema Internacional es el faradio.

Capacitancia Capacidad que tiene un componente, cuerpo, circuito, etc., para acumular una carga eléctrica.

Carga Magnitud que indica la cantidad de electricidad de un cuerpo. Esta cantidad la manifiesta dicho cuerpo cuando, por efecto de una acción externa, varía el número de electrones respecto del de protones del núcleo de los elementos que lo constituyen. Las cargas de electricidad se caracterizan por las acciones de atracción o repulsión que se ejercen entre ellas.

Cátodo Polo o electrodo negativo de cualquier dispositivo eléctrico.

Demodulación Acción de obtener información de una onda modulada, eliminando la onda portadora, mediante un circuito resonante. Es la operación inversa de la modulación y se denomina también detección.

Demodulador Equipo que permite realizar la demodulación de una onda modulada.

Demultiplexor En electrónica digital, circuito con una entrada de información y varias posibles salidas, que identifica y separa dos o más señales combinadas y transmitidas a través de un canal único. Esto permite una utilización mayor de las líneas especializadas en la transmisión de datos a largas distancias.

Dieléctrico Se dice de la sustancia aislante o no conductor de la electricidad, es decir, capaz de mantener un campo eléctrico en estado de equilibrio sin que pase corriente eléctrica por él.

Digitalización Proceso de cuantificar una señal analógica, convertirla y representarla en forma digital. La digitalización de las señales analógicas consigue un significativo aumento de la velocidad de proceso, así como de la precisión durante la reproducción.

Electrodo Conductor eléctrico a través del cual puede entrar o salir una corriente eléctrica en un medio, ya sea una disolución electrolítica, un sólido, un gas o el vacío. En las disoluciones electrolíticas, en muchos sólidos y masas fundidas, un electrodo es un conductor en cuya superficie se realiza un cambio de conducción por electrones a conducción por iones.

Filtro Red de transmisión utilizada en sistemas eléctricos que se encarga de proporcionar la selección deseada de señales.

Fotodiodo Tipo de célula fotovoltaica que consiste en una unión pn de semiconductores que es sensible a la luz, de modo que la corriente que lo atraviesa depende de la intensidad de luz que incide sobre él. Se funda en el efecto fotoeléctrico interno. Se utiliza para determinar y controlar la cantidad de luz.

Fuente Se dice de cualquier elemento activo (pila, batería, alternador, etc.) capaz de generar una diferencia de potencial entre sus bornes, con destino a la alimentación de un circuito eléctrico o electrónico.

Fusible Dispositivo de seguridad, consistente en un hilo o chapa metálica, de fácil fusión, que se coloca en algunas partes de las instalaciones eléctricas, para que cuando la intensidad sea excesiva, la interrumpa fundiéndose.

Galvanoplastia Parte de la galvanotecnia que comprende las técnicas de producción de objetos metálicos por medio de moldes que actúan como ánodo o cátodo en el seno de disoluciones electrolíticas. Permite la reproducción exacta de un modelo al depositarse el metal en disolución sobre el molde. Los materiales más empleados en la construcción de los moldes son, entre los conductores: el acero, el níquel, el cobre, el latón y el aluminio. Entre los no conductores: las resinas epoxi, las resinas vinílicas, el yeso, la cera y la madera.

Ganancia Factor que indica el poder amplificador de un circuito. Es la relación entre una señal de entrada y la misma señal a la salida. Según el concepto a que se refiera puede ser: de potencia, de tensión y de intensidad. Suele expresarse en decibelios.

Half duplex Término inglés que, en comunicaciones, se aplica a un sistema que puede realizar una comunicación en los dos sentidos, pero no simultáneamente. Es decir, para que uno de los receptores pueda escuchar, tiene primero que dejar de transmitir, y viceversa en el receptor opuesto.

Hardware Voz inglesa que indica el conjunto de unidades físicas, circuitos y dispositivos electrónicos que componen un sistema informático. Su traducción literal al español, quincallería o quincalla, no se utiliza en el ámbito de la informática. El término hardware se usa en oposición al de software, que se refiere a los programas internos o rutinas y a todo el equipo que sirve de ayuda a la programación.

Hexadecimal En electrónica digital, sistema de numeración en base 16.

Impedancia Medida de la oposición que presenta un circuito, o una parte de él, al paso de la corriente eléctrica alterna sinusoidal.

Inducción electromagnética Producción de una fuerza electromotriz debido a la variación del flujo magnético. En un circuito se crea una corriente inducida (debido a la fuerza electromotriz producida) cuando éste está sometido a un campo magnético variable o bien, en el caso de un circuito móvil, cuando éste se mueve en el seno de un campo magnético de forma que varíe el flujo. La ley de Faraday establece el valor de la fuerza electromotriz inducida.

Inducción electrostática Fenómeno por el que un cuerpo neutro adquiere una carga eléctrica por medio de un campo eléctrico o un cuerpo cargado próximo a él. Así se consigue que determinadas zonas del cuerpo se electricen sin alterar la carga total.

Inductancia Reactancia inductiva en la corriente eléctrica. Es una relación entre la cantidad de flujo magnético y la corriente que circula por un inductor o bobina.

Inductor Bobina o parte de un circuito eléctrico que produce el flujo magnético de inducción, destinado a producir una fuerza electromotriz en el inducido. Puede ser fijo o móvil.

Infrarrojo Se dice de la zona invisible del espectro solar situado más allá del rojo, y de las radiaciones correspondientes a esta zona; las radiaciones tienen efectos térmicos, pero no luminosos ni químicos. La longitud de onda está comprendida entre 0,8 y 1000 micrómetros.

Interruptor Aparato destinado a interrumpir o permitir el paso de una corriente eléctrica por un circuito.

Joule Nombre del julio en la nomenclatura internacional.

Julio Unidad de energía o trabajo en el Sistema Internacional de unidades (SI). Se define como el trabajo realizado por la fuerza de 1 newton que desplaza su punto de aplicación 1 m en la dirección de la fuerza. Su símbolo es J.

Kilociclo Unidad de frecuencia que se emplea en radiodifusión. Equivale a 1.000 ciclos, vibraciones u oscilaciones por segundo. También quilociclo. Su símbolo es kc.

LED Es un dispositivo semiconductor emisor de luz policromática cuando es polarización directamente. Se utiliza habitualmente como piloto indicador en los equipos eléctricos o electrónicos.

magnetismo Parte de la física que trata de las interacciones entre imanes y entre cargas eléctricas en movimiento.

Megabyte En electrónica digital, medida de capacidad de almacenamiento equivalente a 1 millón de bytes aproximadamente (1,048,576 bytes).

Microfaradio Unidad de capacidad eléctrica, equivalente a la millonésima de faradio. Su símbolo es mF.

Microprocesador En electrónica digital, unidad central de proceso constituida por uno o más chips (pastillas) de circuitos integrados LSI.

Modulación Acción de variar alguna de las características de una onda (llamada onda portadora) en función de las características de otra onda (llamada onda moduladora) cuya información se desea transmitir. La onda resultante se denomina onda modulada.

Monitor Aparato para el control visual de señales.

Multiplexor En telecomunicaciones, dispositivo que permite dividir un ancho de banda en múltiples canales independientes, de forma que a través de un único medio de transmisión se puedan realizar tantas comunicaciones simultáneas como canales existan.

Ohm Nombre del ohmio en la nomenclatura internacional.

Oscilación Fenómeno por el cual una magnitud (corriente, desplazamiento, etc.) varía en función del tiempo de una forma periódica. El tiempo transcurrido entre dos estados iguales se llama período. La amplitud es la desviación máxima, y el número de oscilaciones por unidad de tiempo es la frecuencia. Las oscilaciones que se propagan en el tiempo y en el espacio reciben el nombre de ondas.

Oscilador Circuito electrónico que convierte la energía de corriente continua en una corriente alterna de frecuencia determinada.

Osciloscopio Tubo de rayos catódicos que permite obtener, sobre una pantalla fluorescente, una representación visual de los valores instantáneos y formas de ondas de magnitudes eléctricas rápidamente variables con el tiempo.

Polaridad Término que define la dirección del flujo de electrones y las condiciones que lo crearon.

Reactancia Parte imaginaria de la impedancia, derivada de la presencia de inducciones y capacidades en un circuito. Es una forma de oposición que los componentes electrónicos (condensadores y bobinas) presentan a la corriente alterna.

Rectificador Aparato o dispositivo para convertir una corriente alterna en unidireccional o continua. Un rectificador ideal posee una resistencia nula en un sentido e infinita en el otro. Cuando se intercala un rectificador en un circuito de corriente alterna, sólo circulará corriente mientras la polaridad de la misma esté en el sentido favorable. Por lo tanto, sólo circulará corriente en un sentido, rectificando así la corriente.

Rendimiento Medida de la eficacia de una máquina, un elemento electrónico, un motor, una reacción química, etc., definida como la relación entre la energía cedida (utilizable) por el sistema y la energía suministrada al sistema.

Semiconductor Sustancia que posee una conductividad eléctrica que varía con la temperatura. Un semiconductor intrínseco es una sustancia pura, la cual, no estando sometida a la acción del calor o de otra radiación, tiene elevada resistividad. La sustancia va disminuyendo su resistividad a medida que va recibiendo energía en forma de calor, luz, etc.

Tensión Aplicado a una fuente eléctrica, es sinónimo de Voltaje, Diferencia de potencial o Fuerza electromotriz.

Tierra Cable eléctricamente neutro. Se utiliza como punto de referencia de los circuitos eléctricos, tanto para medida como para seguridad.

Voltaje Diferencia de potencial entre las extremidades de un conductor eléctrico o entre los bornes de un generador de corriente.

Voltio Unidad de medida de la diferencia de potencial y fuerza electromotriz en el Sistema Internacional de unidades (SI), equivalente a la diferencia de potencial entre dos puntos de un hilo conductor que transporta una corriente de 1 amperio, cuando la potencia disipada entre dichos puntos es igual a 1 vatio. Su símbolo es V.

RESUMEN

En el diseño del circuito para comunicar la luz piloto de las videocámaras con el switcher, se utilizó una tecnología que muy pocas instituciones educativas o de capacitación poseen en Guatemala como lo es el CNC para circuitos electrónicos, el cual consiste en el desgastado, fresado y barrenado de la placa de cobre con la ayuda de un programa de diseño asistido por computadora en el cual se bosqueja el circuito para luego ser convertido a un archivo Gerber, preferido por su estándar internacional, el cual es interpretado y ejecutado por el CNC en movimientos mecánicos de dos dimensiones con una exactitud de 0.01 mm.

Un recurso que en los últimos años se ha hecho muy popular es la tecnología de los microcontroladores debido a la facilidad de su aprendizaje y su diversidad de aplicaciones. Para este circuito se utilizó el microcontrolador 16F84A de Microchip, elegido por su disponibilidad, número de puertos y tamaño físico; además se realizó un programa en lenguaje Basic que simula transmisión UART. Este programa redujo en gran medida la cantidad de dispositivos electrónicos necesarios para la aplicación y el tamaño del circuito electrónico en sí. El circuito de transmisión utiliza cuatro canales de entrada, transmitiendo la información sobre un canal de 433Mhz, hasta una distancia de 152 metros. Dicha información es recibida por cuatro cámaras, cada una con un circuito receptor conectada a la fuente de alimentación de la misma. En el receptor se conecta el tally, que es la luz roja que los conductores de televisión utilizan para saber que cámaras ver siempre. También este circuito se conecta al led que se encuentra en el visor del camarógrafo para saber en que estado se encuentra y no cometer ningún error cuando esté transmitiendo en vivo.

OBJETIVOS

General

Diseñar circuito electrónico con microcontrolador por medio de control numérico computarizado para comunicar luz piloto de videocámara con switcher.

Específicos

1. Dar a conocer las funciones, instrucciones y características más importantes de la familia de microcontroladores 16Fxx.

2. Elaborar una guía para la programación de microcontroladores en lenguaje Basic.

3. Establecer un procedimiento básico para elaborar circuitos electrónicos utilizando una máquina de control numérico computarizado.

4. Realizar la programación del microcontrolador 16F84 en lenguaje Basic, minimizando los componentes electrónicos del circuito en cuestión.

INTRODUCCIÓN

Existen en Guatemala, pequeñas empresas de televisión cuyo presupuesto limitado hace difícil la adquisición de un equipo necesario para realizar una edición de forma profesional, tratando de cubrir estas necesidades con equipo que en ocasiones son de uso doméstico. Uno de los problemas más comunes es la comunicación entre el camarógrafo y la persona que maneja el video en la cabina de control, debido a que no poseen un cable llamado multipin, que es el medio donde desde la cabina se manda la alimentación, la comunicación, video, audio y otras señales.

Por otro lado, existen también en nuestro país algunas instituciones que poseen la tecnología necesaria para diseñar y fabricar equipos para suplir las necesidades de las pequeñas empresas de televisión, a un bajo costo y con una calidad parecida o igual al equipo original. Entre esta tecnología, podemos mencionar el equipo CAD/CAM que se utiliza en manufactura, metal-mecánica, así como en el diseño y manufactura de circuitos electrónicos. Otra tecnología que es muy útil, sencilla de utilizar y al alcance de cualquier persona, son los microcontroladores, cuyas aplicaciones abarcan las industrias de automóviles y electrodomésticos, entre otras, los cuales se pueden programar no sólo en lenguaje ensamblador, si no que ya permite la programación en lenguajes de alto nivel como Pascal, Basic, C++, etc.

Con las herramientas y equipo mencionados anteriormente se propone un circuito que comunique el *switcher* con cuatro videocámaras máximo, de forma inalámbrica, utilizando circuitos que se empotren en los mismos, sustituyendo la parte de comunicación del cable *multipin*, a un precio al alcance del presupuesto de estaciones de televisión local, permitiendo realizar una producción de mejor calidad y agradable para el tele-espectador.

1. MICROCONTROLADORES PROGRAMABLES: CARACTERÍSTICAS GENERALES DE LAS FAMILIAS 16FXX

Un microcontrolador es un circuito integrado programable que ha tenido mucha demanda en la industria, debido a su versatilidad, economía, tamaño, etc. Este circuito integrado posee todos los componentes de un computador, con la diferencia que éste último posee un sistema abierto, esto quiere decir que dependiendo de la aplicación se puede adaptar periféricos o dispositivos que le ayuden a su ejecución; en cambio, un microcontrolador es un computador con una tarea determinada, es por ello que este sistema recibe el nombre de sistema cerrado, que posee prestaciones limitadas no modificables.

Una ventaja de utilizar microcontrolador, es que debido a su tamaño reducido, se puede colocar junto con el dispositivo que gobierna. Esta característica se denomina como Controlador Incrustado (*embeddeb controller)*.

1.1 Arquitectura interna

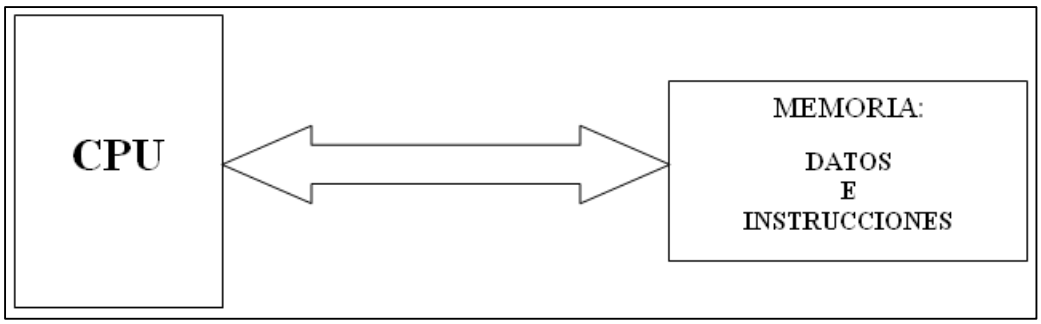
La arquitectura interna de un microcontrolador es muy similar a las de un microprocesador, siendo las partes principales:

a. Procesador
b. Memoria de programa
c. Memoria de datos
d. Líneas de E/S
e. Recursos auxiliares

1.1.1 El procesador

La arquitectura que utilizan tradicionalmente las computadoras es la llamada Arquitectura Von Neumann, la cual se caracteriza por la CPU que se conecta a una memoria única que contiene los datos e instrucciones del programa (figura 1).

Figura 1. Arquitectura Von Neumann



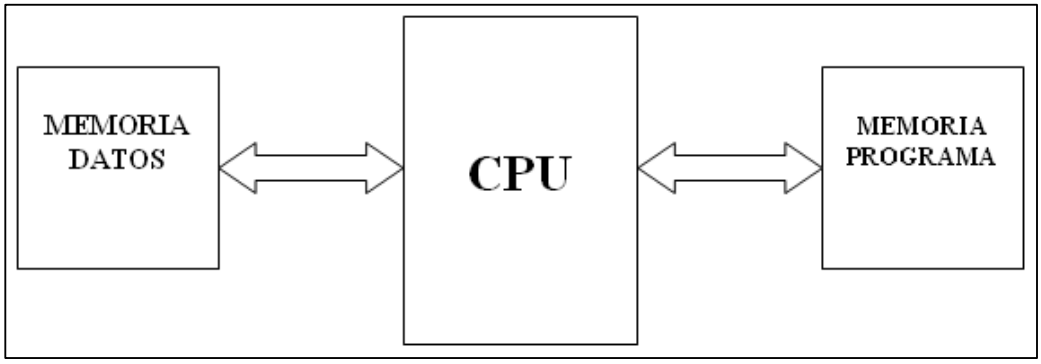
Las principales limitaciones de utilizar esta arquitectura son:

1. La longitud de los datos restringe a la de las instrucciones, provocando que el procesador realice varios accesos a memoria cuando trabaja con instrucciones complejas.

2. La velocidad de operación esta limitada por el único bus para datos e instrucciones.

La arquitectura que utilizan los microcontroladores es la **Arquitectura Harvard,** a diferencia de arquitectura Von Neumann, ésta posee bancos de memoria independientes para las instrucciones y datos, como se muestra en la figura 2, obteniendo así tamaños de palabras de diferente longitud, aprovechando a lo máximo el recurso de la memoria.

Figura 2. Arquitectura Harvard



Las ventajas que ofrece la Arquitectura Harvard son:

1. Los Buses de datos e instrucciones son independientes, permitiendo mayor velocidad del procesador.

2. Permite la ejecución de una instrucción y la búsqueda de la siguiente instrucción a ejecutar, en un ciclo de instrucción, que equivale a cuatro ciclos de reloj, el cual es llamada técnica de segmentación (*pipe-line).*

Una desventaja que posee la Arquitectura Harvard es, que requiere instrucciones especiales para acceder a la memoria del programa.

1.1.2 Memoria de programa

La memoria de programa es donde se almacena todas las instrucciones del programa de control, siendo esta una memoria no volátil, cuya capacidad depende del PIC a utilizar, estando en un intervalo de 512 *bytes* a 8K *bytes*. Según el tipo de memoria de programa que dispongan los microcontroladores, la aplicación y la utilización, existen 5 tipos de memorias no volátiles, siendo estos: *ROM* con máscara, *OTP*, *EPROM*, *EEPROM* y *flash*.

1.1.2.1 *ROM* con máscara

Es un tipo de memoria no volátil, de solo lectura cuyo contenido se graba durante la fabricación del chip. Debido a elevado costo de diseño de esta memoria, únicamente se utiliza cuando se precisan cantidades superiores a varios miles de unidades.

1.1.2.2 Memoria *OTP*

También llamada memoria **Programable Solo Una Vez** (*One Time Programming*), donde el usurario programa el chip por medio de un grabador controlado desde un computador; este tipo de memoria se utiliza para construcción de prototipos y series muy pequeñas.

1.1.2.3. *EPROM*

La *EPROM* (*Erasable Programmable Read Only Memory*), también llamada *UV-PROM*, posee la característica que puede borrarse y grabarse varias veces, siendo el mismo procedimiento que se realiza para el *OTP*. Para borrar el contenido de la memoria, este circuito integrado posee un ventana de cristal en su superficie, donde se incide rayos ultravioleta durante varios minutos, quedando esta para poderse utilizar nuevamente. La desventaja de esta memoria es su costo unitario elevado.

1.1.2.4 *EEPROM*

La *EEPROM* (*Electrical Erasable Programmable Read Only Memory*), se caracteriza de las demás memorias debido a la forma sencilla de programar y borrar, efectuando este procedimiento tantas veces como fuera necesario, ambas efectuadas eléctricamente sobre la misma base del programador. El tipo de *PIC* con memoria *EEPROM* se utiliza para propósitos de enseñanza y en la creación de nuevos proyectos. La memoria EEPROM posee una capacidad de 1.000.000 ciclos de escritura / borrado, y una tensión de programación de aproximadamente 13,5 voltios. Todavía posee ciertas desventajas, una de ellas es el tiempo de escritura es relativamente grande y su consumo de energía es elevado.

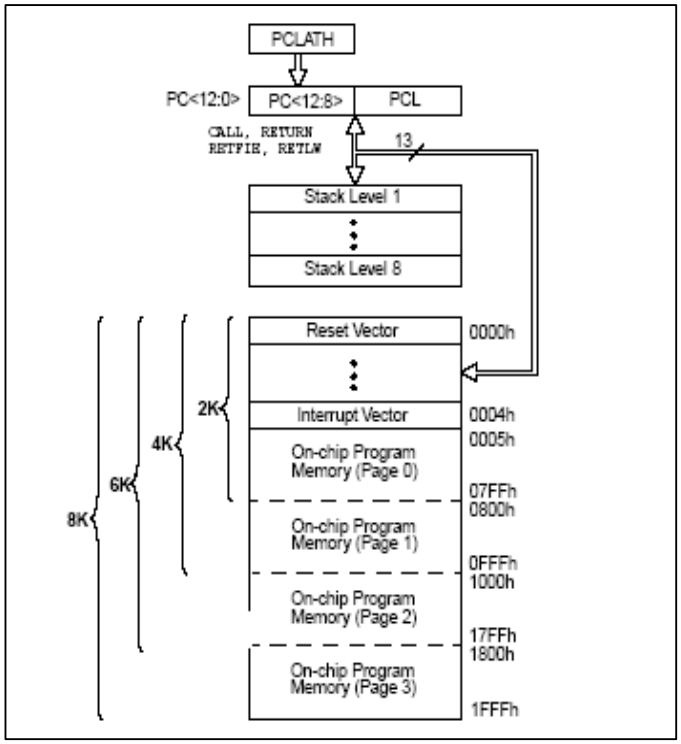
1.1.2.5 Memoria *flash*

La memoria flash es similar a la *EEPROM*, pero de mayor capacidad, más veloz y su consumo más bajo. La tensión de trabajo para este tipo de memoria es de 2 voltios a 5.5 voltios y 5 voltios para su programación.

1.1.2.6 Organización de memoria de programa

La gama media de los *PIC* poseen un contador de programa de 13 *bits*, capaz de direccionar 8Kx14 posiciones de memoria. Posee un mapa de memoria de programa capaz de contener 8,192 instrucciones de 14 *bits* cada una dividida en cuatro páginas de 2,048 posiciones. En la figura 3 se muestra la organización de la memoria del programa.

Figura 3. Arquitectura de memoria del programa y pila



Fuente: Microchip. **Mid-Range MCU Family Reference Manual**. Pág. 97

1.1.2.6.1. Contador del programa (PC)

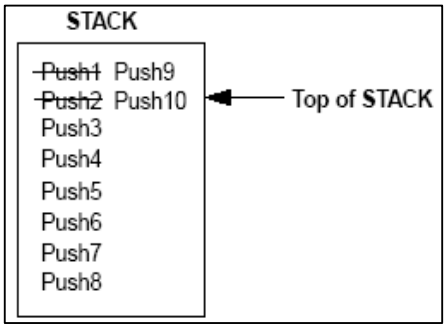
Este contador contiene la dirección de la próxima instrucción a ejecutar. Se incrementa automáticamente al ejecutar cada instrucción, de manera que la secuencia natural de ejecución del programa es lineal. Existen instrucciones que cambian el contenido de la *PC* alterando la ejecución del programa, dos de ellas son el *GOTO* y *CALL*. Al inicializarse el microprocesador, todos los *bits* del *PC* tomar valor 1, de manera que la dirección de arranque del programa es siempre la última posición de memoria de programa.

Dentro de un microcontrolador, el contador del programa es accesible al programador visto como un registro de memoria interna de datos en la posición 02, llamado *PCL*. Los microcontroladores de la gama media poseen un contador de 13 *bits*, cuyo 8 *bits* de menor peso corresponde a 8 *bits* de un registro llamado *PCL*, y los otros cinco de mayor peso corresponden al registro *PCLATH* (*Program Counter Latch High*). Para saltar entre las páginas, se puede modificar de forma indirecta los *bits* mas significativos del contador del programa, por medio del *PCLATH*, modificando los *bits* 3 y 4.

1.1.2.6.2 La pila

La pila es una zona aislada de las memorias de instrucciones y datos. Tienen una estructura *LIFO* (*Last in first out*), en la que el último valor guardado es el primero que sale. Tiene 8 niveles de profundidad cada uno con 13 *bits*. Funciona como un *buffer* circular, de manera que el valor que se obtiene al realizar la novena operación es igual al que se obtuvo en el primero como se observa en la figura 4. Los *PIC* no disponen de instrucciones específicas para manejar directamente la pila. Tampoco se dispone de algún señalizador que indique cuando se produce el desbordamiento de los 8 niveles de la pila.

Figura 4. Desbordamiento de la pila



Fuente: Microchip. "**Mid-Range MCU Family Reference Manual"**. Pág. 100

La única manera de cargar la pila, es por medio de una subrutina o interrupción. Para recuperar la instrucción o instrucciones guardadas en la pila, tiene que finalizar le interrupción o la subrutina.

1.1.3. Memoria de datos

Los programas utilizan datos de forma continua, lo que exige que esta unidad de almacenamiento sea de lectura y escritura. Los tipos de memoria que se utiliza son la *RAM* estática (*SRAM*) y la *EEPROM*.

1.1.3.1. Organización de la memoria de datos

La memoria de datos se dividen en varios segmentos iguales de tamaño llamados bancos, dependiendo del *PIC*, este puede dividirse en dos o en cuatro. El usuario tiene la capacidad de elegir el banco de memoria donde puede trabajar únicamente habilitando los *bits* llamados *RP0* y *RP1* del registro *STATUS* tal como lo indica la tabla I.

Tabla I. Elección del banco por *RP1* Y *RP0*

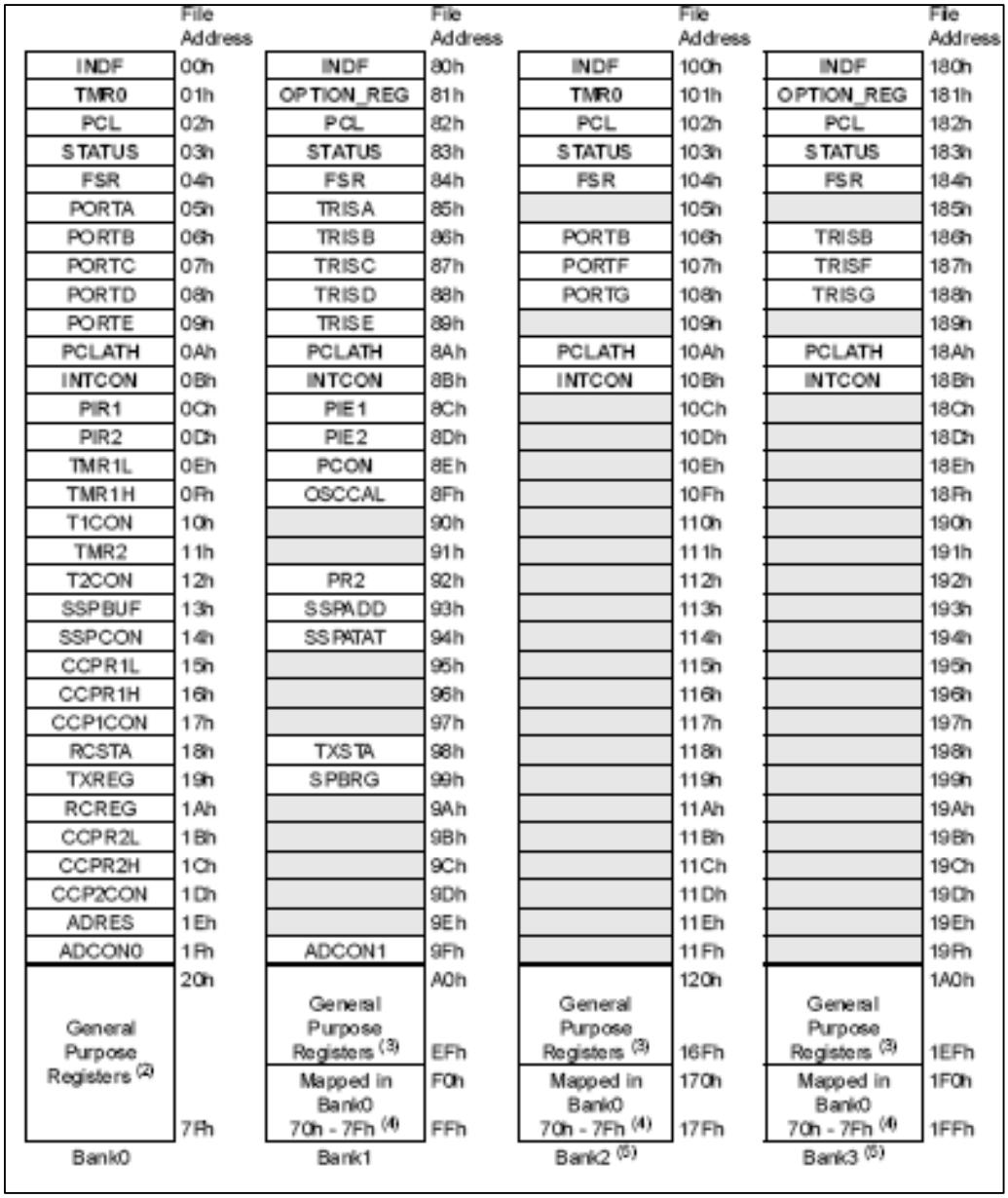
BANCO	*RP1*	*RP0*
0	0	0
1	0	1
2	1	0
3	1	1

1.1.3.1.1. Registros de funciones especiales y de propósitos generales

Los registros de funciones especiales (*FRS*) son registros usados por la *CPU* y los módulos periféricos encargados de controlar el funcionamiento adecuado del dispositivo. Los *FRS* se clasifican en dos tipos, el primero son los registros que corresponden a la *CPU* y los que controlan los periféricos.

Los bancos contienen los registros de propósito generales (*GPR*) y los registros de funciones especiales (*SFR*). La cantidad de cada banco depende del tipo de microcontrolador, por ejemplo, el *PIC*16F84 posee únicamente dos bancos mientras que *PIC*16F877 posee cuatro bancos (Fig. 5 y 6). Pero para cualquier microcontrolador, las posiciones mas bajas están reservadas para los registros de funciones especiales. Arriba de los *SFR* se encuentran los registros de propósito general, que se utilizan como posiciones de memoria *RAM* estática. Algunos *SFR* están reflejados, esto quiere decir que el mismo registro se encuentra en varios bancos para reducir el código y tener un acceso más rápido.

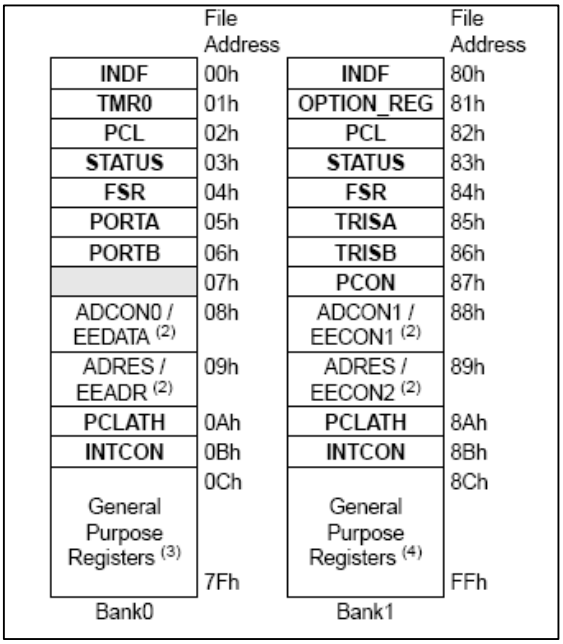
Figura 5. Mapa de archivo de registros *PIC*16F877



	File Address		File Address		File Address		File Address
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	PORTF	107h	TRISF	187h
PORTD	08h	TRISD	88h	PORTG	108h	TRISG	188h
PORTE	09h	TRISE	89h		109h		189h
PCLATH	0Ah	PCLATH	8Ah	PCLATH	10Ah	PCLATH	18Ah
INTCON	0Bh	INTCON	8Bh	INTCON	10Bh	INTCON	18Bh
PIR1	0Ch	PIE1	8Ch		10Ch		18Ch
PIR2	0Dh	PIE2	8Dh		10Dh		18Dh
TMR1L	0Eh	PCON	8Eh		10Eh		18Eh
TMR1H	0Fh	OSCCAL	8Fh		10Fh		18Fh
T1CON	10h		90h		110h		190h
TMR2	11h		91h		111h		191h
T2CON	12h	PR2	92h		112h		192h
SSPBUF	13h	SSPADD	93h		113h		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
ADRES	1Eh		9Eh		11Eh		19Eh
ADCON0	1Fh	ADCON1	9Fh		11Fh		19Fh
General Purpose Registers (2)	20h – 7Fh	General Purpose Registers (3)	A0h – EFh	General Purpose Registers (3)	120h – 16Fh	General Purpose Registers (3)	1A0h – 1EFh
		Mapped in Bank0 70h - 7Fh (4)	F0h – FFh	Mapped in Bank0 70h - 7Fh (4)	170h – 17Fh	Mapped in Bank0 70h - 7Fh (4)	1F0h – 1FFh
Bank0		Bank1		Bank2 (5)		Bank3 (5)	

Fuente: Microchip. ***Mid-Range MCU Family Reference Manual***. **Pág. 104**

Figura 6. Mapa de archivo de registros *PIC*16F84

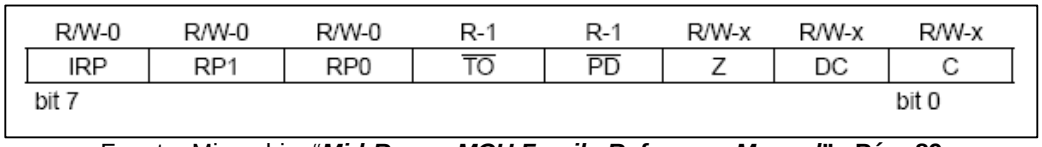


Fuente: Microchip. "***Mid-Range MCU Family Reference Manual***". **Pág. 104**

1.1.3.1.2. Registro de estado

Ocupa la tercera dirección tanto del banco 0 como del banco 1, de la memoria de datos. Este registro se encarga de los estados de los resultados de la *ALU*, como por ejemplo la de acarreo, resultado cero, acarreo y llevar. Otra función es que indica el estado de inicialización, también seleccionan el banco a acceder en la memoria de datos. Los *bits* del registro estado se puede observar en la figura 7.

Figura 7. Registro Estado



Fuente: Microchip. "***Mid-Range MCU Family Reference Manual***". **Pág. 89**

Los *bits* 3 y 4 del registro estado son únicamente de lectura, los otros son de lectura / escritura.

***Bit* 0: *C*. Acarreo / llevada en el *bit* de más peso**

1: Cuando se ha producido un acarreo en el *bit* más significativo.

0: No se ha producido acarreo.

Bit* 1: *DC*. Acarreo / llevada en el cuarto *bit

1: Cuando se ha producido un acarreo en el cuarto *bit* más significativo. Utilizado en operaciones en *BCD*.

0: No se ha producido acarreo.

***Bit* 2: *Z*. Cero**

1: El resultado de una instrucción lógica aritmética ha sido 0.

0: El resultado de una instrucción lógica aritmética no ha sido 0.

Bit* 3: *PD#. Power down

1: Luego que el *PIC* ha sido encendido o al ejecutar la instrucción clrwt.

0: Cuando se pone a dormir el *PIC*.

Bit* 4: *TO#. Time out

1: después de la conexión de la alimentación o al ejecutarse las instrucciones clrwt y *sleep*

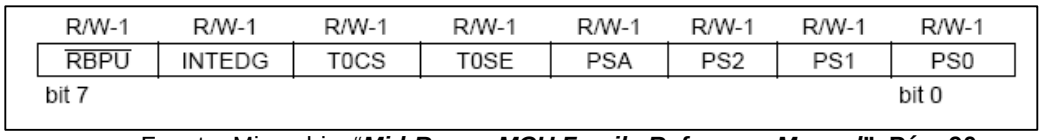
***Bit* 5 y 6: *RP1-RP0*. Selección de banco de direccionamiento directo.** Dependiendo del tipo de microcontrolador, se utiliza *RP0* para microcontroladores de dos bancos y se utilizan *RP1* y *RP0* para microcontroladores de cuatro bancos.

***Bit* 7: *IRP*. Selección del bando en direccionamiento indirecto.** Este *bit* junto al *bit* de más peso del registro *FSR*, sirven para determinar el bando de la memoria de datos seleccionado. En los microcontroladores de dos bancos no se utiliza este *bit*, y debe de ponerse en 0.

1.1.3.1.3. Registro *OPTION*

Este registro puede ser leído o escribir sobre cada *bit* que lo compone. Su función principal es la de controlar el *TMR0* y el divisor de frecuencia. Ocupa la posición 81 H de la memoria de datos, ubicada en el banco 1. En la figura 8 muestra la distribución de los *bits* de *OPTION*.

Figura 8. Registro *OPTION*



Fuente: Microchip. "***Mid-Range MCU Family Reference Manual***". **Pág. 90**

***Bit* 0-2: *PS0-PS2*,** rango de actuación del divisor de frecuencia.

Tabla II. División del *TMR0* y *WDT*

PS2	PS1	PS0	División del *TMR0*	División del *WDT*
0	0	0	1:2	1:1
0	0	1	1:4	1:2
0	1	0	1:8	1:4
0	1	1	1:16	1:8
1	0	0	1:32	1:16
1	0	1	1:64	1:32
1	1	0	1:128	1:64
1	1	1	1:256	1:128

Bit* 3: *PSA, (*Preescaler Asignement*) Asignación del divisor de frecuencia.

1: Divisor de frecuencia se le asigna al *WDT*.

0: Divisor de frecuencia se le asigna al *TMR0*.

Bit* 4: *TOSE, (*Timer 0 Clock Source Select)* Se asigna el tipo de flanco con que el TMR0 cambia de estado.

1: Incremento de *TMR0* en cada flanco descendente.

0: Incremento de *TMR0* en cada flanco ascendente.

***Bit* 5: *TOSC*,** (*Timer 0 Clock Edge Select*) Tipo de reloj para el *TMR0*.

1: Los pulsos se introducen a través de *TOCKI*

0: Los pulsos de reloj interno *Fosc*/4.

***Bit* 6: *INTEDG*,** (*Interrupt Edge*) Flanco de control de interrupciones.

1: Interrupción por flanco ascendente en el pin *RB0/INT*.

0: Interrupción por flanco descendente en el pin *RB0/INT*.

Bit* 7: *RBPU#, (*RB Pull-Up*) Resistencia de *Pull-up* en el puerto B.

1: Resistencia de *pull-up* desactivada.

0: Resistencia de *pull-up* activada.

1.1.3.1.4. Registro *INTCON*

La mayor parte de los señalizadores y *bits* de permiso de las fuentes de interrupción en los *PIC* 16x8xx, se encuentran ubicados en este registro, que ocupa la dirección 0B H del banco 0, duplicado en el banco 1. Los *bits* del registro *INTCON* se encuentra ordenados en la figura 9. Son 13 posibles causas de interrupciones para los microcontroladores de 28 patitas, 14 para los de 40 patitas.

Figura 9. Registro *INTCON*

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
GIE	PEIE (3)	T0IE	INTE (2)	RBIE (1, 2)	T0IF	INTF (2)	RBIF (1, 2)
bit 7							bit 0

Fuente: Microchip. ***Mid-Range MCU Family Reference Manual*”**. **Pág. 127**

***Bit* 0: *RBIF*,** señalizador de cambio de estado en las patas RB4:RB7. También conocido como GPIF.

1: Cuando cámbiale estado de alguna de estas 4 líneas. Se borra por el programa.
0: No se ha producido un cambio de nivel en los pines RB4:RB7.

Bit* 1: *INTF, Señalizador de estado de la interrupción externa *INT*.
1: La interrupción externa se ha producido. Se borra por el programa.
0: Indica que *INT* aún no se ha activado.

Bit* 2: *T0IF, señalizador de desbordamiento de *TMR0*.
1: *TMR0* se ha desbordado. Se borra por programa.
0: *TMR0* no se ha desbordado.

Bit* 3: *RBIE, *bit* de habilitación por cambio de nivel en el puerto B. También llamado *GPIE*.
1: Habilita la interrupción.
0: No habilita la interrupción.

Bit* 4: *INTE, *bit* de habilitación de la interrupción externa por la línea RB0/INT. No todos los *PIC* contienen este *bit*.
1: Permite la interrupción.
0: No permite la interrupción.

Bit* 5: *T0IE, *bit* de habilitación de la interrupción por desbordamiento del *TMR0*.
1: Permite la interrupción al activarse RB0/*INT*.
0: Prohíbe esta interrupción.

Bit* 6: *PEIE, *Bit* de habilitación de interrupciones de los periféricos que no se controla con el registro *INTCON*. En algunos microcontroladores también es llamado *EEIE* o *ADIE*.
1: Habilita el permiso de interrupciones de los periféricos.
0: Inhabilita las interrupciones de los periféricos.

***Bit* 7: *GIE*.** Permiso global de interrupciones.
1: Habilita el permiso de interrupciones.
0: Inhabilita el permiso de interrupciones.

1.1.3.1.5. Registro *PIE1*

Contiene los *bits* que permiten o prohíben las interrupciones provocadas por los periféricos internos del microcontrolador y que no estaban contempladas en *INTCON*. Ocupa la dirección 8C H. La posición de los *bits* en el registro, depende del tipo de microcontrolador que se este utilizando, en forma general, se describe cada uno de los *bits* de dicho registro. Para que el registro *PIE1* cumpla su función, primero se habilita en el registro *INTCON*, poniendo a uno el *bit* 6 llamado *PEIE*.

***TMR1IE*:** *Bit* de habilitación de interrupción por desbordamiento del *TMR1*.
1: Habilita la interrupción por desbordamiento.
0: Inhabilita la interrupción por desbordamiento.

TMR2IE: *Bit* de habilitación de interrupción por desbordamiento del *TMR2*.
1: Habilita la interrupción por desbordamiento.
0: Inhabilita la interrupción por desbordamiento.

***CCP1IE*:** *Bit* de habilitación de interrupción del módulo *CCPI* cuando se produce una captura o una comparación.
1: Habilita la interrupción del *CCPI*.
0: Inhabilita la interrupción del *CCPI*.

SSPIE**:** *Bit* de habilitación de interrupción por el puerto serie síncrono.
1: Habilita la interrupción del *SSP*.
0: Inhabilita la interrupción del *SSP*.

TXTIE**:** *Bit* de interrupción al transmitir por el *USART* cuando se vacía el *buffer*.
1: Habilita la interrupción de transmisión por el *USART*.
0: Inhabilita la interrupción de transmisión por el *USART*.

RCIE: *Bit* de habilitación de interrupción en recepción por el *USART*, cuando se llena el *buffer.*
1: Habilita la interrupción por recepción en el *USART*.
0: Inhabilita la interrupción por recepción en el *USART*.

ADIF**:** *Bit* de habilitación de interrupción por finalización de la conversión A/D.
1: Habilita la interrupción del convertidor A/D.
0: Inhabilita la interrupción del convertidor A/D.

PSPIE**:** *Bit* de habilitación de interrupción por lectura / escritura en el puerto paralelo esclavo. Para los modelos de 40 pines, manteniéndose en 0 los que poseen 28 patas.
1: Habilita la interrupción por lectura / escritura en el *PSP*.
0: Inhabilita la interrupción por lectura / escritura en el *PSP*.

1.1.4. Líneas de E/S

Los microcontroladores poseen líneas que le permite comunicarse con periféricos. Dependiendo del modelo de microcontrolador, puede tener uno o más conjuntos de líneas, el cual se le conoce con el nombre de puerto o puerta. Estos puertos poseen la capacidad de comunicarse en distintos protocolos, por ejemplo *I²C*, *USB*, entre otros. También posee pines con doble función como por ejemplo convertidor analógico digital, para conectar pantalla de cristal líquido, resistencia *pull-up*, entre otras, las cuales se puede habilitar con el registro que las controla.

Hay que tomar en cuenta que en los microcontroladores con 40 patitas disponen de 5 puertas denominadas: PA, PB, PC, PD Y PE de entrada y salida con un total de 33 líneas. A diferencia de los microcontroladores de 28 patitas, solo poseen tres puertas, denominados: PA, PB Y PC con 22 líneas de entrada y salida. Los nombres de las patitas para los microcontroladores de la gama media, se resumen en las siguientes tablas.

Tabla III. Patitas de propósito general

NOMBRE	DESCRIPCIÓN
OSC1/*CLKIN*	Entrada del cristal de cuarzo o del oscilador externo.
OSC2/*CLKOUT*	Salida del cristal de cuarzo. En modo RC la patita OSC2 saca la cuarta parte de la frecuencia que se introduce por OSC1, determina el ciclo de instrucción.
Vss	Conexión a tierra.
V_{DD}	Alimentación positiva.
MCLR#/VPP/THV	Entrada de reinicio o entrada del voltaje de programación o voltaje alto en el modo prueba.

Tabla IV. PUERTO A

NOMBRE	DESCRIPCIÓN
RA0/AN0 RAN1/AN1	Posee dos funciones, como línea digital de entrada / salida o como entrada analógica del conversor analógico digital
RA2/AN2/V_{REF}-	Línea digital de entrada / salida, entrada analógica o entrada del voltaje negativo de referencia
RA2/AN2/V_{REF}+	Línea digital de entrada / salida, entrada analógica o entrada del voltaje positivo de referencia
RA4/*T0CK*I	Línea digital de entrada / salida del contador 0. Salida con colector abierto.
RA5/*SS#AN4*	Línea digital de entrada / salida, entrada analógica o selección como esclavo de la puerta serie síncrona.

Tabla V. PUERTO B

NOMBRE	DESCRIPCIÓN
RB0/INT	Línea digital de entrada / salida o entrada de petición de interrupción externa.
RB1	Línea digital de entrada / salida.
RB2	
RB3/*PGM*	Línea digital de entrada / salida o entrada del voltaje bajo para programación.
RB4	Línea digital de entrada / salida.
RB5	
RB6/*PGC*	Línea digital de entrada / salida. En la programación serie recibe las señales de reloj.
RB7/*PGD*	Línea digital de entrada / salida. En la programación serie actúa como entrada de datos.

Tabla VI. PUERTO C

NOMBRE	DESCRIPCIÓN
RC0/T1OSO/T1CKI	Línea digital de entrada / salida. Salida del oscilador del temporizador 1 o como entrada del temporizador 1
RC1/T1OSI/CCP2	Línea digital de entrada / salida. Entrada al oscilador del contador 1. Entrada al módulo Captura2/salida comparación2/salida de PWM2
RC2/CCP1	Línea digital de entrada / salida. Entrada al módulo Captura1/salida comparación1/salida de PWM1
RC3/SCK/SCL	Línea digital de entrada / salida. Entrada de reloj serie sÍncrona / salida de los modos SPI e I2C.
RC4/SDI/SDA	Línea digital de entrada / salida. Entrada de datos en modo SPI o I/O datos en modo I2C
RC5/SDO	Línea digital de entrada / salida. Salida de datos en modo SPI.
RC6/TX/CK	Línea digital de entrada / salida. Patita del transmisor del USART asíncrono o como reloj del síncrono.
RC7/RX/DT	Línea digital de entrada / salida. Receptor del USART asíncrono. Receptor de datos en el síncrono.

Tabla VII. PUERTO D

NOMBRE	DESCRIPCIÓN
RD0/PSP0-RD7/PSP7	Línea digital de entrada / salida. Líneas para la transferencia de información en la comunicación de la puerta paralela esclava.

Tabla VIII. PUERTO E

NOMBRE	DESCRIPCIÓN
RE0/RD#AN5	Entrada / salida digital. Señal de lectura para la puerta paralela esclava. Entrada analógica.
RE1/WR#/AN6	Entrada / salida digital. Señal de escritura para la puerta paralela esclava. Entrada analógica al conversor analógico-digital.
RE2/CS#/AN7	Entrada / salida digital. Activación o desactivación de la puerta paralela esclava. Entrada analógica

2. PROGRAMACIÓN DEL *PIC* EN *BASIC*

La programación de alto nivel posee la ventaja que se puede realizar programas para aplicaciones industriales complicados con un reducido número de líneas de instrucción, cosa que no se puede realizar en ensamblador. La ventaja que posee ensamblador y es por ello que no ha sido descontinuado, que la velocidad del microcontrolador es mucho mas alta debido a la sencillez de instrucciones del ensamblador.

2.1 Términos básicos del lenguaje *Basic*

Existen programas de alto nivel que se basan en Basic para trabajar con los microcontroladores, pero básicamente la diferencia entre estos es el conjunto de instrucciones o librerías que utiliza, siguiendo siempre la estructura de programación de *Basic*. Esto hace que la programación de los microcontroladores sea más sencilla y práctica, comparándolo con lo tedioso que es trabajar en lenguaje ensamblador.

Los programas que más se utiliza para trabajar *PIC* en *Basic* son *PicBasic, Picsimulatoride, MikroBasic*, entre otros, los cuales son gratuitos. En el presente trabajo se trabajará con *MikroBasic*, debido a que posee diversas librerías muy sencillas de utilizar, siempre y cuando exista un conocimiento previo de lo fundamental de *Basic.*

2.1.1 Fundamentos de *MiKroBasic*

En todos los programas en Basic, siempre se utiliza el apostrofo (') para identificar donde inicia los comentarios, los cuales tienen como objetivo hacer más sencillo la lectura y ubicación de rutinas o instrucciones dentro del programa, en especial cuando se trata de estructuras muy grandes.

Para representar enteros, en base 10, 2 ó 16, existen prefijos que nos ayuda a diferenciar estos sistemas numéricos del uno del otro. Si un número no posee prefijo, indica que está en base 10. El signo de dólar ($) o el prefijo 0x indica que es un numero en hexadecimal. Si el prefijo es el signo de porcentaje (%), indica que el número esta escrito en binario.

Los tipos de datos antes mencionados no pueden ser mayores a 2127483647, puesto que al compilarlo el programa detectará un error. También existen lo que se llaman las palabras reservadas, los cuales no pueden ser utilizadas por el usuario como identificadores, puesto que son propios del programa. Las palabras reservadas se muestran en la tabla IX.

Tabla IX. Palabras reservadas

absolute	*clear*	*for*	*is*	*org*
abs	*const*	*function*	*loop*	*print*
and	*dim*	*goto*	*label*	*procedure*
array	*div*	*gosub*	*mod*	*program*
asm	*do*	*if*	*module*	*read*
begin	*double*	*incluye*	*message*	*select*
boolean	*else*	*in*	*new*	*sub*
case	*end*	*int*	*next*	*step*
char	*exit*	*integer*	*not*	*string*
chr	*float*	*interrupt*	*or*	*switch*

El identificador no puede iniciar con un número o con un carácter especial, como por ejemplo %, $, etc. *MikroBasic* no identifica las mayúsculas de las minúsculas. Si se escribe Raza, raza o rAzA, el programa lo identifica como el mismo identificador, a esto en lo que se le llama sensibilidad. Algunos programas poseen la opción que permite que las palabras antes escritas sean tres identificadores distintos, pero al desactivarlo las lee como una sola.

2.1.1.1 Organización del programa

La organización del programa básicamente está comprendido en dos partes: declaración de variables y constantes y el programa como tal. *Mikrobasic* detecta la organización del programa, si escribe una declaración fuera de lugar, al momento de compilarlo existe la posibilidad de que no sea leído correctamente, entonces el programa no funcionaría como se esperaría. A continuación se presenta un esquema donde se observa el orden correcto de programación.

```
programa     <Nombre del programa>
include      <Incluye otros módulos>
'* Declaraciones (globales):
'*********************************************************
Symbol                ' declaraciones de símbolos
Const                 ' declaraciones de constantes
Dim                   ' declaraciones de variables
sub procedure nombre del procedimiento (...)   'declaraciones de
                                               'procedimientos
  <Declaraciones locales>
end sub
sub function nombre de la función(...)     ' Declaraciones de funciones
  <Declaraciones locales>
end sub
'*********************************************************
'* Cuerpo del programa:
'*********************************************************
Principal:
 'En esta sección se escribe el cuerpo del programa
end.
```

2.1.1.2 Variables y constantes

Una variable es un objeto cuyo valor puede cambiar durante todo el programa, siempre y cuando sea declarada antes de utilizarla. Cuando asigna un valor a una variable, lo que está haciendo es ocupar un espacio en la memoria, cuyo tamaño dependerá del valor asignado. Esto hay que tomarlo en cuando al momento de programas los microcontroladores, puesto que posee un espacio limitado en su memoria de programación. A continuación se presenta ejemplos de variables.

Dim identificador ***as*** tipo_dato
Dim t, v, p ***as*** *byte*
Dim potencia, temp ***as*** *Word*

Una constante es un dato cuyo valor no cambia durante la ejecución del programa, el cual no puede ser modificado durante el mismo. Este tipo de dato no ocupa lugar en la memoria. A continuación se dan ejemplos de la estructura que debe de llevar al momento de declarar una constante. Si el programador no define el tipo de dato a declarar, entonces el programa automáticamente asume el tipo más adecuado.

Const nombre_constante [***as type***] = cantidad
Const lux ***as*** l*ongint* = 9999
Const min = 1584 ' compilador asume tipo *Word*
Const mensaje= "Hello" ' compilador asume tipo *string*

2.1.1.3 Funciones y procedimientos

Funciones y procedimientos son subprogramas que realizan una cierta tarea, dependiendo de las condiciones que se cumplan en el programa principal. La diferencia que existe entre las funciones y procedimiento, es que el primero regresa un valor, cuando es ejecutado, en cambio el segundo no lo hace. Para declarar una función, la estructura básica se muestra a continuación.

```
sub function nombre_funcion(lista_parametros) as carácter_retorno
  [Declaraciones locales]
                                        'Cuerpo de la función
end sub
```

El nombre de la función debe ser cualquier identificador permitido por el programa. La lista de parámetros es similar cuando se declara variables. Define el tipo de dato del valor de la función, este puede ser de cualquier tipo. La parte de declaraciones locales, se pueden definir variables y constantes, este es opcional dependiendo de las necesidades del programa.

A continuación se ejemplifica la estructura de una función.

```
sub function power(Dim x, n as byte) as longint
        Dim i as byte
        i = 0
        result = 1
                if n > 0 then
                        for i = 1 to n
                                result = result*x
                        next i
                 end if
end sub
```

El nombre del procedimiento puede ser cualquiera que no sea palabra reservada o ya esté utilizada por otro procedimiento, función, variable, etc. La lista de parámetros es similar cuando se declara variables en el programa principal. Las declaraciones locales son opcionales, aquí es donde se declaran las variables y constantes para ser utilizadas en el cuerpo del programa. La estructura básica de un procedimiento y un ejemplo se muestra a continuación

```
sub procedure nombre_procedimiento(Lista_parametros)
  [ declaraciones locales ]
                                    'Cuerpo del programa
end sub
```

```
sub procedure time_prep(Dim byref seg, min, hr as byte)
          seg  = ((seg and $F0) >> 4)*10 + (seg and $0F)
          min  = ((min and $F0) >> 4)*10 + (min and $0F)
           hr   = ((hr  and $F0) >> 4)*10 + (hr  and $0F)
end sub
```

2.1.1.4 Tipos de datos

Los tipos de datos nos ayudan a definir no solo el tipo de variable o constante que se va utilizar, si no que también el tamaño correcto de la localidad de memoria. Dependiendo del origen del dato existe dos, el primero definido por el usuario donde se mencionan las estructuras, y el segundo es el que está predefinido por el programa, entre los cuales se encuentran: tipos de datos simples (*simple types),* arreglos (*arrays*), cadena de caracteres (*strings*).

2.1.1.4.1 Datos simples

El tipo de datos simples son aquellos que no pueden ser divididos en partes más pequeñas, entre los cuales se encuentran el *byte*, *char*, *word*, *short*, *integer*, *longint*, *float*, cuyos tamaños y rangos se muestran en la tabla X.

Tabla X. Datos simples

TIPO	TAMAÑO (*bits*)	RANGO
Byte	8	0 a 255
Char	8	0 a 255
Word	16	0 a 65535
Short	8	-128 a 127
Integer	16	-32768 a 32767
Longint	32	-2147483648 a 2147483647
Float	32	$\pm 1.17549435082 * 10^{-38}$ a $\pm 6.80564774407 * 10^{38}$

2.1.1.4.2 Arreglos

Los arreglos son matrices de datos del mismo tipo, esto quiere decir que en la matriz, cada dato ya sea constante o variable posee una ubicación única, pero los datos pueden ser iguales en el mismo arreglo. En la estructura básica del arreglo, cabe una longitud de 0 hasta n-1, donde n es la longitud del tipo de dato, dentro del par de corchetes. Estos datos deben de ser identificados correctamente, escribiendo el tipo que sea del lado izquierdo de los corchetes.

type[longitud_arreglo]

Ejemplo:

Dim dias_semana **as *byte***[7]

Dim muestra **as *Word***[50]

Const meses **as *byte***[12] = (31,28,31,30,31,30,31,31,30,31,30,31)

Const numero **as *byte***[4][4] = ((0, 1, 2, 3), (5, 6, 7, 8), (9, 10, 11,12), (13,14, 15, 16))

2.1.1.4.3 Cadena de caracteres

La cadena de caracteres o *string*, es un equivalente de un arreglo de char. Para declarar un *string,* se encierra entre corchetes el número de caracteres a trabajar y a la par de los corchetes, colocar el código *string.* Cada carácter entre el arreglo, posee un número, el cual el último carácter coincide con el dato encerrado entre los corchetes. La estructura básica para *string* se presenta a continuación.

string[*longitud_string*]

Ejemplo:

Dim msg1 **as string**[10]

Dim msg2 **as string**[20]

principal:

msg1 = "Hola mundo"

msg2 = "Hola mundo, de nuevo"

2.1.1.5 Operadores

Los operadores, son símbolos que realizar una operación determinada dentro del computador. Básicamente existen tres tipos de operadores, los cuales son operaciones aritméticas, operaciones lógicas y las operaciones de relación. Las operaciones aritméticas realizan operaciones matemáticas, como suma resta, multiplicación, división, etc., tal como lo muestra en la tabla XI.

Tabla XI. Operadores

Operador	Operación	Operandos	Resultado
+	Suma	*byte, short, integer, word, longint, float*	*byte, short, integer, word, longint, float*
-	Resta	*byte, short, integer, word, longint, float*	*byte, short, integer, word, longint, float*
*	Multiplicación	*byte, short, integer, word, float*	*integer, word, longint, float*
/	División con punto flotante	*byte, short, integer, word, float*	*byte, short, integer, word, float*
Div	División, aproxima al entero menor cercano	*byte, short, integer, word, longint, float*	*byte, short, integer, word, longint*
mod	Modulo	*byte, short, integer, word, longint*	*byte, short, integer, word, longint*

Las operaciones lógicas son realizadas *bit* a *bit* de los datos a operar. Para realizar este tipo de operación se necesitan dos datos para trabajar, excepto la operación *Not*, que utiliza el complemento de un solo dato. Las funciones lógicas trabajan de izquierda a derecha de los datos, excepto la operación *Not*, que opera de derecha a izquierda. Estas operaciones únicamente son para datos binarios, si existe otro sistema numérico, entonces es necesario realizar la conversión adecuada. Las operaciones lógicas se muestran en la tabla XII.

Tabla XII. Operaciones lógicas

Operador	Descripción
And	El resultado es verdadero si los dos *bits* son verdaderos, de lo contrario el resultado es falso.
Or	El resultado es verdadero si uno o los dos *bits* son verdadero, de lo contrario, el resultado es falso
Not	Invierte el estado de cada *bit*.
Xor	Compara dos datos, si son iguales el resultado es falso, si son distintos el resultado es verdadero.
<<	Desplaza un *bit* a la izquierda, introduciendo un cero a la derecha para mantener la misma cantidad de datos.
>>	Desplaza un *bit* a la derecha, introduciendo un cero a la izquierda para mantener la misma cantidad de datos.

2.1.1.6 Declaraciones

Una declaración define una acción dentro del programa, definido por el usuario por medio de una instrucción o alguna librería ya establecida dentro del programa utilizado. Una declaración incluye llamadas de rutina, saltos condicionales, etc., las cuales pueden ser combinadas con otras estructuras para ejecutar determinada tarea de forma ordenada dentro del código fuente.

2.1.1.6.1 Declaración condicional

Las declaraciones condicionales permiten tomar decisiones y realizar un proceso repetidas veces. La declaración condiciones es la sentencia *If*, cuya sintaxis se escribe a continuación. Si la condición es verdadera, entonces una o más sentencias después del *Then* se ejecutan. Si la condición es falsa, entonces la condición *Else* se cumple. Cuando trabaje la sentencia *If*, tiene que trabajar con *Then*, pero *Else* es opcional, esto depende del programador.

if *expresión* ***then***

Sentencias

else

Sentencias

end if

2.1.1.6.2 Sentencia *select case*

Permite realizar una acción de entre varias opciones posibles de una misma expresión. A continuación se muestra la estructura básica de dicha sentencia. La expresión es numérica o alfanumérica. Los valores pueden ser expresiones, expresiones a expresiones, operador de relación expresión o combinación de las anteriores, pero separadas por comas.

select case expresion

case valores

Sentencia

case valores_1

sentencia_1

case else

sentencia_2

end select

2.1.1.7 Sentencias de ciclos

2.1.1.7.1 Sentencia *While*

La sentencia *While* se utiliza para generar bucles mientras se cumpla una condición inicial. La sintaxis de la sentencia se escribe a continuación. Mientras la expresión se cumpla, el ciclo se va a ejecutar. La expresión puede ser una comparación de la cantidad de una variable con un número, un número que se compara con un contador dentro de las sentencias, etc., si la condición se cumpla, entonces el programa sale de ciclo.

while *expresión*
    Sentencias
wend

2.1.1.7.2Sentencia *Do*

El programa estará dentro de un bucle, mientras la condición dada sea verdadera. Las condiciones de las sentencias son ejecutadas repetidamente, evaluando la expresión en cada ciclo verificando si la condición se cumple, si no se cumple, el programa estará encerrado en el bucle, de lo contrario el programa ejecutará otras sentencias. La estructura de dicha sentencia se presenta a continuación.

do
    Sentencias
loop until expresión

2.1.1.7.3 Sentencia *For*

Esta sentencia provoca que el programa entre a un lazo, ejecutando una serie de instrucciones cierto número de veces. La forma general de del ciclo *For* se muestra más adelante. En la sintaxis de *For*, se asigna un valor un valor a la variable y comprueba si el dato es mayor que expresión_2, si es verdadero, sale del ciclo, de lo contrario el programa estará encerrado en el bucle, incrementando la variable hasta que se convierta en verdadero. La expresión_3 nos indica si el dato incrementa o decrementa en un cierto valor establecido.

for variable = *expression_1 to expression_2 [step expression_3]*
  sentencias
next variable

2.1.1.8 Librerías

Mikrobasic provee un conjunto de librerías, cuyo fin es reducir el tiempo de programación para ciertas funciones específicas de los *PIC*. Como por ejemplo, si el usuario necesita utilizar el convertidor analógico digital del *PIC*16F877A, únicamente debe de escribir la librería respectiva y colocar los parámetros que le convenga al programador. A continuación se escribirá las librerías más útiles para los microcontroladores de la serie 16F8XX.

2.1.1.8.1 Convertidor A/D

En la estructura de la librería del convertidor A/D del *PIC*, se define la variable donde se va a almacenar el dato digital de 10 *bits* de la señal analógica ya convertida. El registro *ADCON1*, debe de ser definido por el usuario antes de utilizar dicha librería, puesto que aquí es donde se definen los pines analógicos o digitales, así como también el voltaje de referencia. A continuación se presenta la estructura y un ejemplo del uso de la librería.

Variable = ***Adc_Read***(Número de canal)

Ejemplo:

```
'Es un programa que lee el dato analógico y lo almacena el la variable
'Sal_digi, exponiendo los 8 bits menos significativos en el puerto D.

program Lectura_A/D
Dim sal_digi as Word        'Es necesario que sea declarado como
main:                       'Word, debido a que son 10 bits.
        ADCON1 = $80            ' Configuración entradas analógicas
                                'y Vref
        TRISA  = $FF            ' Puerto A como entrada
        TRISD  = $0             ' Puerto D como salida

        while TRUE
                sal_digi = Adc_Read(2)  'Lee dato analógico del canal 2.
                PORTD = temp_res        ' Envía los 8 BMS al
                                        ' Puerto D
                PORTB = Word(temp_res >> 2)  'Manda 2 BMS al
                                             'Puerto B
         wend
end.
```

2.1.1.8.2 *EEPROM*

La librería que se usa para trabajar con *EEPROM*, permite que esta sea escrita y leída. En caso de que la memoria va a ser leída, entonces únicamente se define la dirección donde se quiere acceder, y el dato será cargado a una variable. Si la memoria va a ser escrita, se necesita la dirección y el dato que será guardado en la memoria. Es necesario definir la dirección como palabra y el dato como *byte*. A continuación se presenta la librería para leer y escribir en la *EEPROM*, seguido de un ejemplo.

Variable = ***Eeprom_Read***(dirección)
Variable = ***Eeprom_Write***(dirección, dato)

Ejemplo:

'*Este fragmento de programa escribe 5 datos en la memoria, luego,*
'*estos datos son expuestos al puerto B.*

```
Dim i, j as char
main:
    TRISB = 0
    for i = 0 to 5                          ' Almacena 5 datos en las
                                            'direcciones
            Eeprom_Write(i, i + 6)                  '0 a la 5.
    next i

    for i = 0 to 5
            PORTB = Eeprom_Read(i)      ' Datos almacenados son
                                        'escritos en el
                for j = 0 to 200        'puerto B
                    Delay_us(500)
                next j
    next i
end.
```

2.1.1.8.3 Retardos

En esta librería, existen diferentes tipos de retrasos. En un tipo de retardo el usuario define el tiempo a esperar, indicando si lo quiere en milisegundos o microsegundos, con la cantidad deseada. Otro tipo de retardo es utilizando una variable, donde se carga un valor y la instrucción lee dicho número en milisegundos. El último tipo de atraso es utilizando el tiempo interno del *PIC*, hace esperar n veces el reloj interno del microcontrolador. A continuación se escribe la forma básica de dichas instrucciones.

Delay_us(n)

Ejemplo:

Delay_us(10) ‘hace un retardo de 10 microsegundos

Delay_ms(n)

Ejemplo:

Delay_ms(10) ‘Realiza un retardo de 10 milisegundos

Vdelay_ms(variable)

Ejemplo

atraso = 1000

Vdelay_ms(atraso) ’ Realiza un atraso de 1 segundo

Delay_Cyc(n)

Ejemplo:

Delay_Cyc(10) ‘Realiza un retardo de 10 veces el reloj del *MCU*.

2.1.1.8.4 *PWM*

La librería para trabajar modulación por ancho de pulso, consiste en cuatro instrucciones, donde hay que especificar el módulo a utilizar, la frecuencia de trabajo y la duración del ciclo. El modulo hay que elegirlo si el microcontrolador posee mas de uno. La frecuencia de trabajo está dado en Hertz y la duración de ciclo se determina con un valor cuyo rango es de 0 (0 %) a 255 (100 %), el programador coloca el más adecuado según la aplicación. Las instrucciones se muestran a continuación.

Pwm_Init(valor) ‘ valor es una frecuencia en Hertz.
Pwm_Change_Duty(n) ‘n es un numero de la duración del ciclo
Pwm_Start ‘el módulo *PWM* comienza a trabajar.
Pwm_Stop ‘detiene el módulo *PWM*.

Ejemplo:

```
program Prueba_PWM
Dim j as byte

main:
    j = 0
    PORTC = $FF          ' Configura Puerto C como entradas
    Pwm_Init(5000)       ' Módulo trabajará a una frecuencia de 5Khz.
    Pwm_Start            ' Inicia a trabajar
        while true
            for i = 0 to 20
                Delay_us(500)
                Inc(j)
                Pwm_Change_Duty(j)  ' Cambia la duración de
                                    'ciclo acorde j.
        wend
end.
```

2.1.1.8.5 LCD de 8 *bits*

Para trabajar la librería de la pantalla de cristal líquido en modo de 8 *bits*, se requieren los siguientes datos: definir los puertos a trabajar y definir las filas y columnas donde se van a desplegar las variables o constantes en la LCD. También existen otras instrucciones que nos permite colocar el cursor donde se requiera, apagar o encender la pantalla, entre otros. A continuación se presenta la librería para controlar la pantalla, así como otros comandos permitidos en la tabla XIII.

Lcd8_Init(PuertoX, PuertoY)
Lcd8_Out(fila, columna, "variable o texto a desplegar")
Lcd8_Out_Cp("variable o texto a desplegar ")
Lcd8_Chr_Cp("despliega solo un carácter")
Lcd8_Cmd(ejecuta un comando)

Tabla XIII. Comandos especiales

Comando	Descripción
LCD_FIRST_ROW	Mueve el cursor a la primera fila
LCD_SECOND_ROW	Mueve el cursor a la segunda fila
LCD_THIRD_ROW	Mueve el cursor a la tercera fila
LCD_FOURTH_ROW	Mueve el cursor a la cuarta fila
LCD_CLEAR	Limpia la pantalla
LCD_RETURN_HOME	Regresa al cursor a su posición inicial
LCD_CURSOR_OFF	Apaga el cursor
LCD_UNDERLINE_ON	El cursor se coloca debajo del texto
LCD_BLINK_CURSOR_ON	El cursor parpadea
LCD_MOVE_CURSOR_LEFT	Cursor se mueve a la izquierda
LCD_MOVE_CURSOR_RIGHT	Cursor se mueve a la derecha
LCD_TURN_ON	Enciende la pantalla
LCD_TURN_OFF	Apaga la pantalla

2.1.1.8.6 *USART*

La librería *USART* da una serie de instrucciones que nos permite comunicar de forma asíncrona *full duplex* con una computadora con protocolo RS-232 u otro microcontrolador. El programador necesita definir los módulos *USART*, así como los puertos adecuados, esto es dependiendo del microcontrolador. La tasa de baudios también se debe definir puesto que si no lo define, el compilador lo detectará como error. A continuación se presenta la estructura básica, así como un ejemplo.

Usart_Init(Baudios)
Usart_Data_Ready = 1
Usart_Read
Usart_Read_Text(txt)
Usart_Write("Dato a mandar")
Usart_Write_Text("Texto a mandar")

Ejemplo:

El siguiente fragmento de programa comunica al microcontrolador con una computadora vía RS232, cuando el microcontrolador recibe el dato, este de un aviso que recibido.

```
program com_usart
Dim dato_recibido as byte
main:
  Usart_Init(2400)              ' Initialize USART module
  while true                    ' con la cantidad de baudios
    if Usart_Data_Ready = 1 then    ' Si el dato es recibido
      dato_recibido = Usart_Read   'almacena en dato_recibido
      Usart_Write(dato_recibido)    ' Manda un aviso de dato recibido
    end if
  wend
end.
```

2.1.8.7 Sonido

Existe una librería que permite que el usuario disponga de distintos sonidos para ciertos eventos. Para ello necesita colocar una bocina piezoeléctrica o un buzer para producir el sonido. El programador necesita determinar la patita de un puerto como salida, debe de calcular el periodo y cuantos periodos va sonar. Dicho periodo que desea ingresar el usuario, debe de dividirlo dentro de 10, como lo pide la instrucción. También hay que calcular cuantos periodos desee que vaya a sonar la alarma.

```
Sound_Init(PuertoX, #_pin)
  Sound_Play(Periodo, #_veces)

  program Prueba_Sonido
Dim valor_ADC as integer
main:
  PORTB  = 0              ' Limpia puerto B
  TRISB  = 0             ' Puerto B como salidas
  INTCON = 0              ' Deshabilita todas las interrupciones.
  ADCON1 = $82              ' Configura VDD como Vref
  TRISA  = $FF             ' Puerto a como entrada
  Sound_Init(PORTB, 2)        ' Inicializa sonido
  while verdadero
    valor_ADC = Adc_Read(2)     'Se almacena el valor convertido
    Sound_Play(valor_ADC, 200)  'Inicia el sonido
  wend
end.
```

2.1.1.8.8 Matemática

En la librería de matemática, se reúnen las funciones más comunes en la programación. El usuario debe de tener en cuenta que varias de las funciones trabajan únicamente en un rango, por ejemplo el la raíz cuadrada. En trigonometría existen instrucciones que trabajan en grados y otras en radianes. A continuación se presenta la tabla la tabla XIV, donde se hace una breve descripción de algunas funciones matemáticas más comunes.

Tabla XIV. Operaciones matemáticas.

Nombre	Descripción
Acos(x)	Función trigonométrica de Arc Coseno de x, el resultado está dado en radianes.
Asin(x)	Función trigonométrica de Arc Seno de x, el resultado está dado en radianes.
Atan(x)	Función trigonométrica de Arc Tangente de x, el resultado está dado en radianes.
Ceil(x)	Aproxima la variable a un valor próximo entero mayor.
Cos(x)	Realiza la función coseno de X, el resultado es en radianes.
Exp(x)	Eleva la base e a una potencia.
Fabs(x)	Realiza el valor absoluto de un número
Log(x)	Logaritmo natural de x.
Log10(x)	Logaritmo de base 10
Pow(x,y)	Eleva a la base x a una potencia y
Sin(x)	Seno de x, el cual esta en radianes.
Sqtr(x)	Realiza la raíz cuadrada de un valor x positiva.
Tan(x)	Ejecuta la tangente de un valor de x, el cual está en radianes.
Tanh(x)	Ejecuta la tangente hiperbólica de un valor de x, el cual está en radianes.

2.1.1.8.9 Comunicación a un solo cable

La librería de comunicación a un solo cable, nos permite comunicar al microcontrolador con cualquier periférico por medio de una sola patita de un puerto especificado. Las ventajas de esta librería es que es de bajo costo, su taza de transferencia alcanza los 16Kbps, se puede utilizar a distancias de hasta 300 metros. El programador debe de definir la patita del puerto donde se va a leer o escribir. En seguida se escribirá las tres instrucciones para dicha librería.

Ow_Reset(PuertoX, PinY) ‘Reestablece el Pin Y del Puerto X.

Temporal = Ow_Read(PuertoX, PinY) ‘Lee el dato que está en el Pin Y del
‘Puerto X y lo almacena en la
‘variable temporal.

Ow_Write(PuertoX, PinY Dato) **‘**Escribe el dato en el pin Y del
‘puerto X.

Ejemplo:

```
program Cable_unico

Dim i, j1,  as byte
  Temporal  as byte

Principal:
   adcon1 = 255          ' Configura RA5 como entrada digital
   PORTA  = 255
   PORTD  =   0
   TRISA  = 255          ' Puerto A como entrada
   TRISD  =   0          ' Puerto D como salida
```

```
  while TRUE
    ow_reset(PORTA, 5)            ' Reinicia la señal
    ow_write(PORTA, 5, $CC)       ' Escribe $CC en RA5
    delay_us(120)
    ow_reset(PORTA, 5)           'Reinicia la señal
    Temporal = ow_read(PORTA, 5)  'Lee del dato de RA5
    PORTA = Temporal                    'Imprime el dato en Puerto A
    Delay_ms(500)
end.
```

2.1.1.8.10 PS/2

Esta lista de instrucciones de la librería PS/2, permite comunicar un teclado de computadora con el microcontrolador, pero no de forma inversa. Para dicha acción el microcontrolador necesita un reloj arriba de 6MHz y los pines que se conectan al *PIC*, deben tener resistencias *pull-up*. Cada pulsador del teclado posee un valor que es leído por el microcontrolador, el cual puede almacenar como código *ASCCI*, una breve descripción de dicha librería se presenta a continuación, así como la codificación de cada pulsador del teclado en la Tabla XV.

Tabla XV. PS/2

Tecla	Código	Tecla	Código	Tecla	Código
F1	1	*Enter*	13	*Caps Lock*	25
F2	2	*Page Up*	14	*End*	26
F3	3	*Page Down*	15	*Home*	27
F4	4	*Backspace*	16	*Scroll Lock*	28
F5	5	*Insert*	17	*Num Lock*	29
F6	6	*Delete*	18	*Left Arrow*	30
F7	7	*Windows*	19	*Right Arrow*	31
F8	8	*Ctrl*	20	*Up Arrow*	32
F9	9	*Shift*	21	*Down Arrow*	33
F10	10	*Alt*	22	*Escape*	34
F11	11	*Print Screen*	23	*Tab*	35
F12	12	*Pause*	24		

Ps2_Init(PuertoX): Asigna como el puerto X para conectar el puerto PS/2.

Ps2_Config(PuertoX, Pin_Reloj, Pin_Dato): Establece que en el puerto X se va asignar un RX como linea de reloj y otro como línea de dato.

Ps2_Key_Read(valor, especial, presionado)= 1: Si la instrucción anterior es verdadero, indica que alguna tecla ha sido presionada, el valor es la tecla que el usuario desee que sea oprimida, especial verifica si alguna tecla de función especial ha sido presionada, como por ejemplo F1, Esc, etc. Presionado nos indica el evento que nos indica si alguna tecla ha presionado o dejado de presionar.

Ejemplo:

‘Este fragmento de programa nos indica cuando la tecla 13 o Enter ‘sea pulsado.

```
do
  if Ps2_Key_Read(val, spec, press) = 1 then
    if (val = 13) and (spec = 1) then
      break
    end if
  end if
loop until FALSE
```

2.2. Programador y compilador

El programador es un circuito especializado para grabar el programa diseñado en un lenguaje ya sea de bajo o alto nivel, dentro de la memoria no volátil del microcontrolador. Existen diversidad de estos circuitos, pero básicamente estos se conectan a un puerto de la computadoras, ya sea serial, paralelo o *USB*, a los pines correspondientes del *PIC*.

El compilador se utiliza dentro del programa que utiliza el usuario para desarrollar la aplicación, cuyo objetivo es convertir el trabajo del usuario a un idioma que la computadora pueda interpretar, en este caso, es necesario que el programa genere un archivo en hexadecimal, que la mayoría de programadores necesitan para poder grabar en el microcontrolador.

2.3 Programa de ejemplo

La programación en lenguaje de Basic o en cualquier lenguaje de alto nivel es amigable para el usuario, los programas que en ensamblador eran muy complicados y extensos, en Basic se reducen más del 50% del tiempo de programación y las líneas de programación. Pero hay que tomar en cuenta que el lenguaje ensamblador nunca va a ser sustituido completamente por los de alto nivel, debido a que dicho lenguaje de bajo nivel se utiliza para realizar correcciones minuciosas en programas extensos. A continuación se presenta un ejemplo de programación, donde se utiliza el módulo *USART* del 16F877A.

```
program Usart_lib
Dim i as byte
Principal:
      Usart_init(2400)          'Inicialización del USART, con una
                                'tasa de 2400  baudios.
  while true
   if Usart_Data_Ready = 1 then     ' Si la información es recibida
    i = Usart_Read                  ' Lee la información y lo guarda en un
                                    ' registro
    Usart_Write(i)                  ' Transmite la información que posee
                                    el registro i.
   end if
  wend
end.
```

3. FUNCIONAMIENTO BÁSICO DEL CONTROL NUMÉRICO COMPUTARIZADO

Se le da el nombre de control numérico computarizado o CNC (*Computer Numerical Control*) a los dispositivos que son capaces de controlar la posición de herramientas en un sistema de coordenadas en dos o tres dimensiones, de forma automática, utilizando el diseño asistido por computadora (CAD: *Computer Aided Design*) y/o la fabricación asistida por computadora (CAM: *Computer Aided Manufacturing*), ejecutado desde un ordenador, con una intervención mínima del operador. Las funciones básicas que realiza el CNC en aplicaciones de máquinas y herramientas son: torneado y fresado, teniendo también tiene aplicaciones en electrónica, ebanistería, carpintería, textiles, entre otros.

Al utilizar CNC o CN (Control Numérico) los prototipos son precisos, cumplen con las especificaciones deseadas y reduce la dificultad para manufacturar partes; así también como para la planeación de operaciones, disminución de costos por herramienta, flexibilidad de maquinado, entre otros. En la actualidad se utiliza la misma tecnología en la realización de circuitos impresos, realizándolo por medio de dos funciones básicas: taladrado y fresado, esto quiere decir que para realizar un circuito impreso desgasta el cobre que no se utiliza y abre los agujeros para los dispositivos. Si se realiza circuitos de superficie, únicamente se tiene la función de fresado.

La placa virgen está conformado por dos materiales: cobre y sustrato no conductor. El objetivo del cobre es proporcionar conexión eléctrica a los dispositivos electrónicos, mientras que el sustrato es una capa de aislante que le brinda a los dispositivos electrónicos soporte mecánico. A través de la historia, el desarrollo del PCB ha ido mejorando junto con la tecnología. Actualmente en las industrias que desarrollan circuitos impresos prefieren lo que es montaje superficial debido al ahorro de espacio y economía.

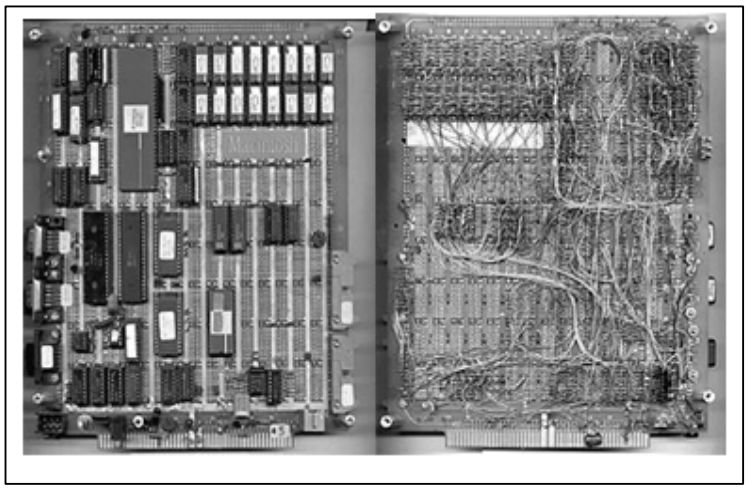
3.1 Historia del PCB

Al principio la conexión entre los distintos componentes se realizaba por medio de cables y a mano, procedimiento que tenía un alto costo económico, funcionalmente no era fiable, físicamente era de gran tamaño, entre otros inconvenientes. A través de los años, muchas personar dedicaron su tiempo pensando y experimentado la manera de mejorar las conexiones entre dispositivos electrónicos. En 1906, Thomas Alba Edison, dejo en sus escritos la posibilidad de sustituir el cableado por un polvo sobre una placa no conductora. En 1927 una compañía alemana lanzó al mercado un amplificador de audio cuyo cableado consistía en unas laminas de latón que intercomunicaba los dispositivos, sobre una placa de material no conductor.

La técnica más utilizada y conocida es la de punto a punto o *wire-wrap* (figura 10 y 11), el cual básicamente comprendían en la colocación directa de los cables conductores a los dispositivos electrónicos. La base que sostenía la circuiteria por lo regular era de madera o de material aislante. El ingeniero austriaco Paul Eisler (1907-1995), en Inglaterra alrededor de 1936, fabricó el primer circuito impreso para una radio, posteriormente fue utilizado para la producción a gran escala de radios con fines bélicos en la Segunda Guerra Mundial, posteriormente en 1943 presentó la primera placa de dos caras.

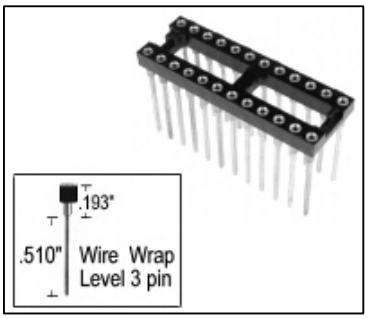
Luego de la guerra, esta tecnología fue liberada para uso comercial, pero no fue hasta en la década de los 50 se convirtió de uso popular en la fabricación de aparatos electrónicos, debido a al desarrollo del auto ensamblaje por parte de los Estados Unidos.

Figura 10. Placa armada con el método *Wire-Wrap*.



Fuente: www.macintosh.com

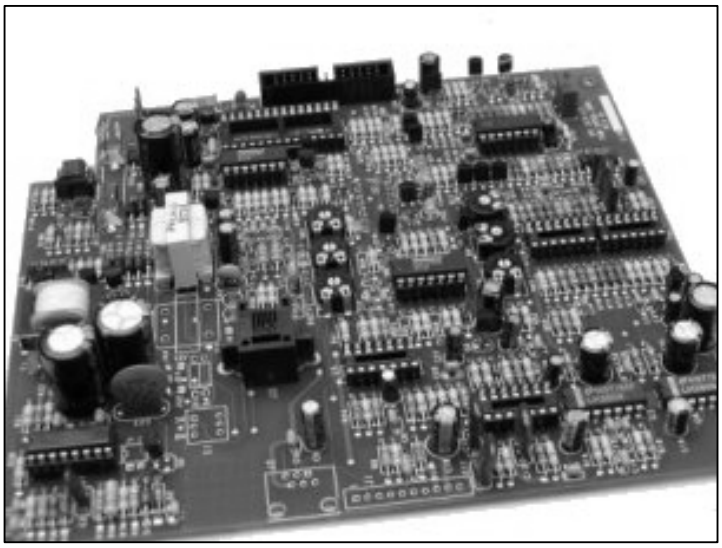
Figura 11. Dispositivo para conexión *Wire-Wrap*.



Fuente: www.phoenixenterprises.com

El método de armado, que permite el auto ensamblaje, es el llamado *Through-Hole* (figura 12 y 13), básicamente consiste en que cada componente posee patas cortas y en la placa agujeros por cada pata, estas eran soldadas a las pistas del circuito impreso. Esta placa posee una cara de cobre el cual le ayuda a conectar eléctricamente los componentes para formar el circuito, este método es el que se utiliza actualmente, para ensamblaje de baja y mediana escala, para gran escala, los productores prefieren el ensamblaje superficial, puesto que se ahorran tiempo, dinero y espacio en la fabricación. La técnica utilizada para soldar era por ola de estaño.

Figura 12. PCB utilizando técnica *Through Hole*.



Fuente: www.answers.com

Figura 13. Dispositivos de montaje *Through-Hole* y superficial.

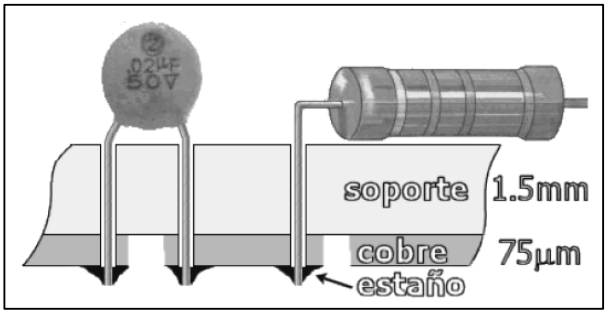
Fuente: www.answers.com

En 1960, en Estados Unidos se patenta la primera placa multicapa con taladros metalizados. En 1965, se realiza el método de baños químicos de metalización para rellenar de material conductor los agujeros de placas multicapas, permitiendo diseños de circuitos con altas densidades de interconexión. En 1971, una compañía holandesa desarrolla el primer circuito integrado para montaje superficial (SMD), abriendo un campo actualmente cotizado, puesto que abrió a nuevas tecnologías en componentes, diseño, montaje, soldadura de montaje en superficie y paso fino entre pistas (FTP) y entre dos patillas consecutivas de un componente.

3.2 Tipos de placa cobrizada

La placa consta de una o varias láminas de material conductor, por lo regular cobre, que descansan en un material aislante, llamado sustrato, en el cual requiere un buen aislamiento eléctrico, buenas propiedades mecánicas, retardo de llama, estabilidad dimensional y fácil mecanizado. Dicho aislante puede ser de fibra de vidrio, Pértinax, baquelita, fibra de cerámica, politetrafluoretileno, resina silicona, resina melamínica, etc. Básicamente la placa posee dos caras, una donde se coloca los dispositivos electrónicos y la otra donde se puede encontrar las pistas, islas y donde se realiza el proceso de soldado, (figura 14).

Figura 14. Composición básica de una placa cobrizada.



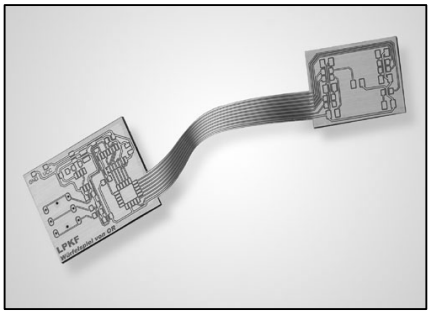
Fuente: **Curso de electrónica práctica**. Pág. 42

El pértinax, que consiste en papel prensado a base de resinas fenólicas es uno de los más comunes, y comercialmente posee códigos dependiendo de ciertas capacidades eléctricas, mecánicas, y tolerancia al calor, los más comunes son: XXXP, XXPC y FR-2. Existen otros códigos para el pértinax, pero básicamente la cantidad de X indica la tolerancia eléctrica así como su comportamiento a alta frecuencia, la P indica que el material es de pértinax y la C indica si es punzonado en frío o en calor. El código FR nos indica la tolerancia que tiene el sustrato ante las llamas.

La placa de vibra de vidrio, también comercial en nuestro medio, requiere herramienta de carburo de tungsteno, debido al esfuerzo que debe realizar por la dureza del material de la placa. Esto incrementa los costos, pero a cambio de una mejor flexión y resistencia a las llamas que el pértinax. La placa cobrizada en base a baquelita, es la más conocida entre las anteriores, es la más económica pero sus propiedades mecánicas y eléctricas dejan mucho que desear. Los circuitos impresos que trabajan a radiofrecuencias elevadas, utilizan plásticos con permitividad baja, los cuales poseen el inconveniente de ser pobres en propiedades mecánicas, pero sus propiedades eléctricas son mejores que los de pértinax y fibra de vidrio.

Los circuitos impresos utilizados en ambientes donde no puede existir enfriamiento por convección, utilizan placas con núcleos conductores de calor, como el cobre y el aluminio, para la disipación de calor de los dispositivos electrónicos, evitando un daño en el PCB. También existen los circuitos impresos que son flexibles, utilizados en equipos de alta integración, como cámaras de video y fotográficas, reproductores de video, entre otros. El objetivo de este tipo de circuitos es el ahorro de espacio entre el equipo, el cual el lugar donde se colocan en muy limitado, por ello que son diseñados para la flexibilidad, estos circuitos son llamados circuitos flexibles o circuitos rígido-flexibles (figura 15).

Figura 15. Circuito rígido flexible



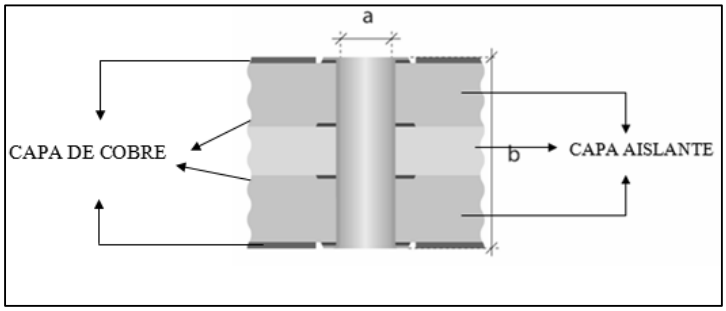
Fuente: www.lpkf.es

El ancho del aislante para la placa cobrizada normal puede variar dependiendo de la aplicación y de la necesidad, estas medidas pueden ser: 0.2, 0.5, 0.7, 0.8, 1.0, 1.2, 1.5, 1.6, 2.0, 2.4, 3.2 hasta 6.4 mm, siendo la más común de 2 mm. El espesor de la placa conductora puede variar de 18, 35, 70 a 105 µm. La placa cobrizada, dependiendo del recubrimiento de sus dos lados, existe placa de una cara y de dos caras. Las placas de dos caras se utilizan en circuitos de mayor complejidad que los de una cara. También se pueden formar placas de más de dos caras con técnicas especiales, estos se utilizan para circuitos de mucha más complejidad, como por ejemplo tarjeta madre de computadora.

Otra nomenclatura que se utiliza en la compra de placas de cobre es la onza. Una onza equivale a un grosor de 35 micrómetros de un cuadrado de 1 pie cuadrado. Media onza equivale un grosor de 18 micrómetros y 2 onzas equivale a un grosor de 70 micrómetros. Las placas con soporte pueden venir con la siguiente nomenclatura: FR4 x/y, donde FR4 nos indica alta estabilidad térmica, no inflamable, soporta choques, **x** indica la cantidad de onzas, **y** nos indica si es de doble cada o simple cara, si este valor es cero, entonces la placa es de una cara, si es uno, la placa es de dos caras. Ejemplo: fibra de vidrio FR4-1/1, mecánicamente es adecuada para realizar circuitos impresos, es de una onza con doble cara.

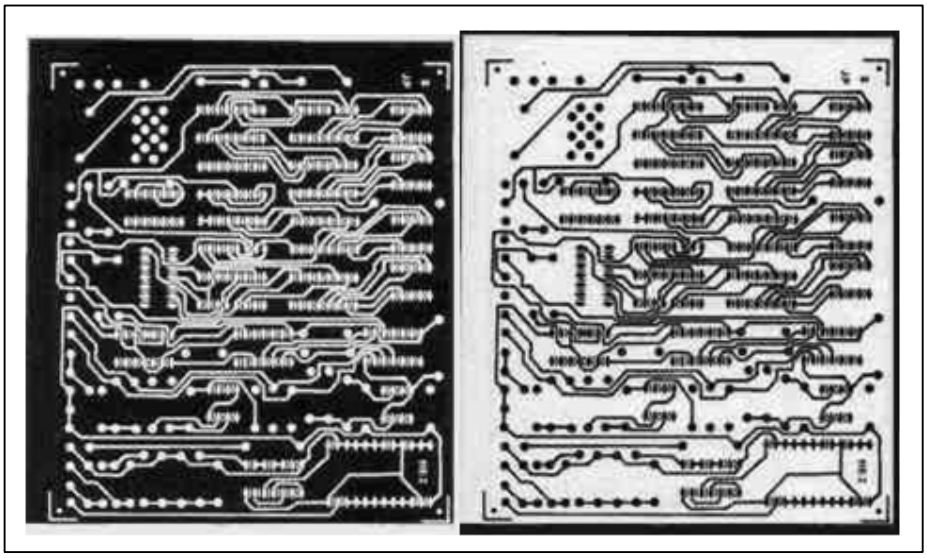
La necesidad de utilizar un circuito multicapas (figura 16) inicia cuando la aplicación requiere circuitos de poco peso, volumen de las interconexiones sea mínimo, blindaje de alguna interconexión, el circuito es muy complicado, entre otras. Hay que tomar en cuenta que cuando se trabaja en más de dos caras, se requiere herramienta, equipo y técnicas especializadas, como por ejemplo, los agujeros para interconectar las caras de la placa, se necesita de buena precisión. Las desventajas de utilizar este tipo de placa es que la simulación de la prueba se convierte en una tarea tediosa, la reparación se vuelve si no difícil, imposible; el tiempo del diseño se incrementa así como el tiempo de fabricación, entre otros.

Fig. 16. Circuito impreso con 4 capas.



Dependiendo del proceso de obtención del circuito impreso, se puede dividir en dos tipos: placa normal y placa fotosensible. La placa normal es más utilizada. La placa fotosensible posee un barniz sensible a la luz, que al entrar en contacto con la luz ultravioleta queda en la placa la imagen positiva o negativa del circuito impreso. La placa positiva es aquella en la cual el las pistas e islas quedan impregnadas en la placa. Figura 17. Mientras que la placa negativa es aquella donde las partes donde no se desea el cobre, son impregnadas por la película fotosensible. Figura 17.

Figura 17. Placa negativa y placa positiva



Fuente: www.elprisma.com

3.3 **Herramienta básica**

La herramienta necesaria para desarrollar un circuito impreso por medio de la tecnología CAD/CAM, depende del material de la placa a trabajar. Por ejemplo se requiere de una herramienta menos especializada cuando se trabaja con placa de pértinax que con fibra de vidrio. Por conveniencia se trabaja con esta última, debido a que soporta más la exposición de las llamas y su resistencia mecánica. Para ello se requiere herramienta construida con carburo de tungsteno cuyo valor triplica a una herramienta de uso común.

Dependiendo de los tipos de trazos que requiere el circuito impreso, el CNC, realiza tres tareas básicas: taladrar, fresar y desgastar, para ello existen herramientas para cada uno de las tareas anteriores, esto depende del tipo de punta que tenga. El largo de estas herramientas es de 38 mm (1.5”) y un diámetro de su base de 3.2 mm (1/8”). Las medidas que se utilizan para este tipo de herramienta son el Sistema Ingles (pulgadas) y el Internacional (milímetros). En el caso del sistema inglés, se utiliza la medida en mils (milésima de pulgada). Para el proceso de taladrar, que consiste en abrir agujeros en la placa para colocar los elementos del circuito se utilizan brocas (figura 18), que dependiendo de la marca van de 8.3 mils a 141.7 mils.

Figura 18. Brocas.

Fuente: http://www.t-tech.com

El proceso de desgastado consiste en quitar en el borde de los circuitos según el diagrama como primer paso y luego quitar el excedente de cobre, figura 19. Para ello existen dos herramientas denominadas *Endmill y T-milling*. La diferencia entre las dos herramientas es que la primera posee el extremo plano para abarcar más excedente de cobre, mientras que la segunda se utiliza también para desgastar cobre, pero en zonas donde el área es muy pequeña, lo que indica que el extremo termina en punta, Figura 20. Las medidas para que se manejan comúnmente en *Endmill* ven desde 4 mil hasta 125 mil, mientras que para el *T-milling*, va desde 8 mil hasta 14 mil.

Figura 19. Placa sin exceso y con exceso de cobre

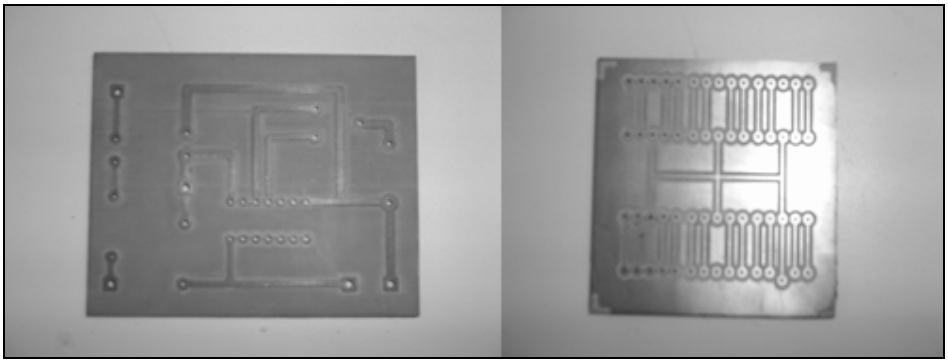
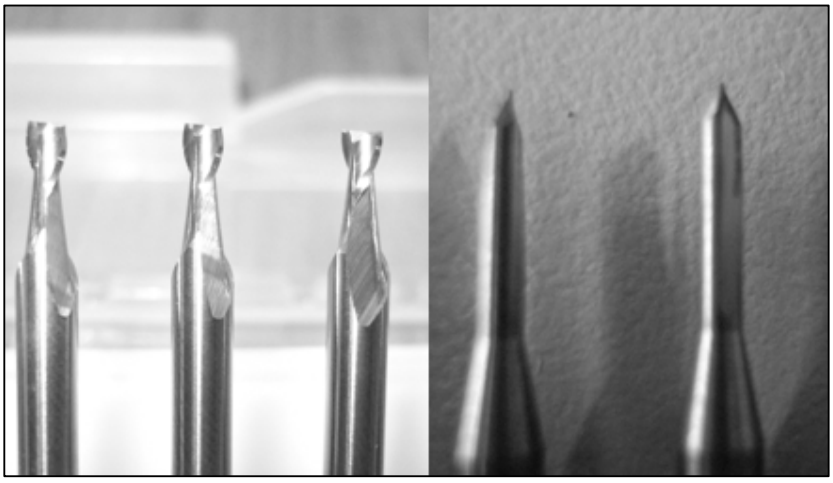


Figura 20. Herramientas *endmill y t-mill*



El proceso de fresado consiste en abrir agujeros de gran tamaño que las brocas no pueden realizar o también figuras necesarias dentro de la placa. Cuando se lleva a cabo esta función hay que tomar en cuenta que el cobre junto con el soporte es extraído, figura 21. La herramienta que realiza este trabajo es comúnmente llamado *Router*, en la industria es llamado comúnmente Ruteador de contorno. Como se observa en la figura 22, el extremo del mismo posee la geometría necesaria para realizar cortes, lo único que debe de tener en cuenta el operario es que esta al ser usado en un corte largo por un tiempo prolongado, se calienta y corre el riesgo de quebrarse. Esta posee dimensiones de 31 mil a 62 mil.

Figura 21. Agujero dejado por un ruteador de contorno.

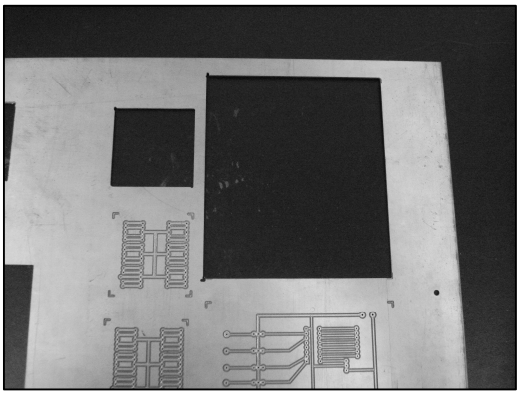
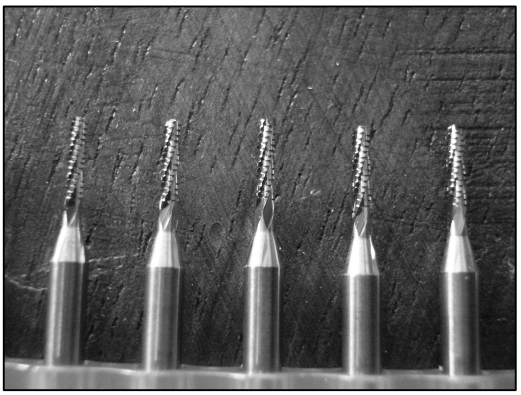


Figura 22. Ruteador de contorno.



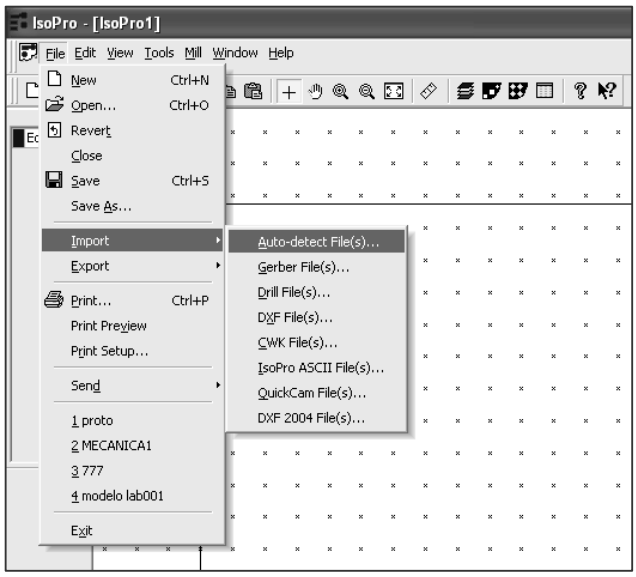
3.4 Fundamentos de IsoPro

IsoPro no se puede considerar un programa CAD en su totalidad, puesto que este tipos de programa permite la realización del diseño desde cero, permitiendo elegir los tipos de encapsulado para los dispositivos electrónicos, terminales de alimentación, salida y entrada, según necesidad y exigencias de la aplicación del PCB. Los programas CAD nos dan una visión dimensional del producto terminado, así como una simulación del funcionamiento del circuito terminado. IsoPro solo convierte estos datos al lenguaje de máquina necesarios para que la máquina realice el PCB. Estos datos que son interpretados por la máquina son llamados archivos Gerber.

Los archivos Gerber o también llamados archivos G, son datos en códigos *ASCII* con coordenadas e instrucciones simples, las cuales son interpretadas por la máquina de manufactura asistida por computadora. Estos datos nos proporcionan el largo y ancho de las pistas, el diámetro de las islas y agujeros, define cual cara es para soldar y para sostener los componentes, si se utilizan dispositivos de superficie, área de trabajo, entre otros datos, en pocas palabras, estos archivos le indica que herramienta utilizar el momento de elaborar el circuito. Estos archivos, con los años se han convertido en un estándar a nivel mundial.

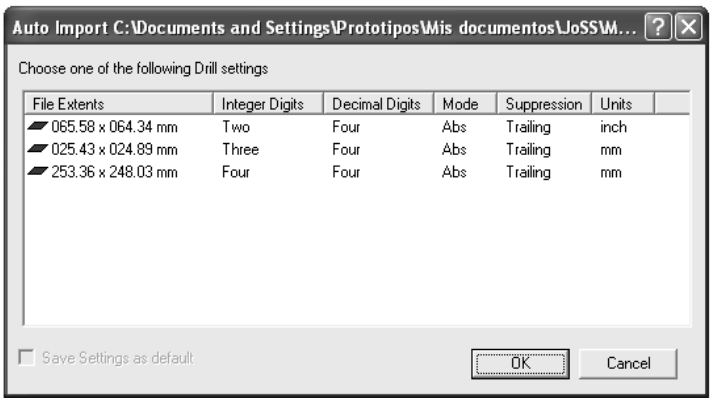
Para cargar los archivos del proyecto del programa tipo CAD, seleccionar de la barra de herramienta *File* > *Import* > *autodetect File (s)* (figura 23). Luego aparece una ventana donde se selecciona el proyecto que consta de tres archivos con extensiones *.GBR, *.SOL, *.NCD. El archivo con extensión GBR, contiene todos los datos de los componentes de la placa. El archivo con extensión SOL, contiene todos los datos donde se ubican las soldadoras y el archivo con formato NCD contiene toda la información de los puntos a taladrar.

Figura 23. Menú para importar archivos



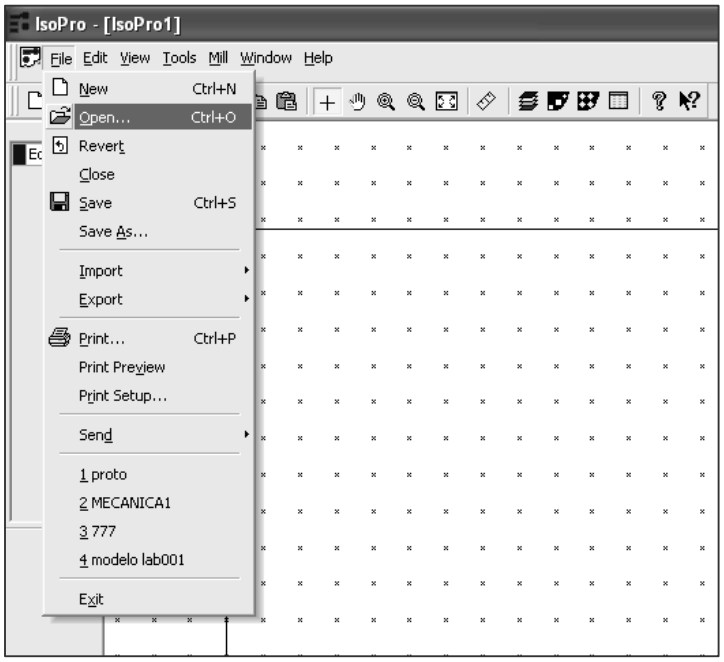
Se debe marcar los tres archivos al mismo tiempo y abrirlos. Luego aparece otra ventaja donde muestra el tamaño del área de trabajo, figura 24, la cantidad de dígitos con que se van a trabajar (enteros y decimales). Luego se selecciona el tamaño de la placa a trabajar, estos datos dependen del diseñador. Existen otros programas que utilizan extensiones distintas, pero son compatibles con dicho programa, tal es el caso de las extensiones *.DXF, que contienen datos importantes, como tamaño de las pistas, agujeros, tamaño de la placa, etc.

Figura 24. Menú para importar automáticamente



Para abrir un archivo propio de Isopro, de la barra de herramientas *File > open,* figura 25, aparece todos los archivos con extensión ISO y se abre el archivo que se desea. Cuando el diseñador marca un archivo aparece una vista en miniatura del archivo, esto sirve para tener la certeza de que archivo se quiere abrir. Estos archivos, contiene todos los datos necesarios para realizar el PCB, como tamaño de la placa, dimensiones de las pistas, islas y agujeros. Para esto, esta dividido en capas que ayuda a trabajar con facilidad cada característica de la placa.

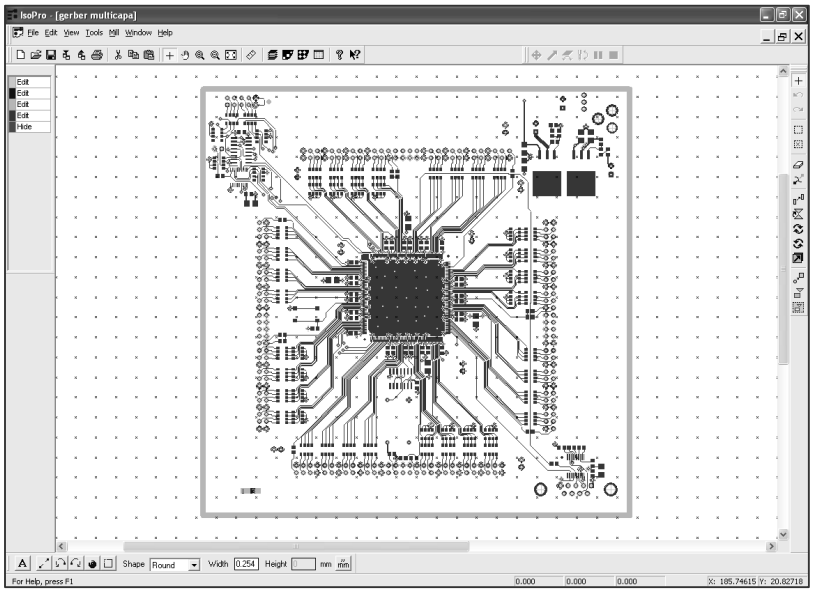
Figura 25. Menú para abrir archivo



El IsoPro trabaja con tres capas básicas, representadas por distinto color, la primera es la capa que contienen todos los datos de los componentes llamado *component,* la segunda es la que contiene todos los datos necesarios para abrir los agujeros, llamado *Drill,* y por ultimo la capa donde básicamente contiene los datos de los puntos donde se va a soldar, llamado *Solder.*

Las capas pueden trabajar de tres maneras distintas una es en *Edit*, donde se puede realizar modificar, seleccionar, borrar y editar la capa respectiva. El segundo es *View* donde únicamente se puede observar la capa, no se puede realizar cambios a la capa. Por ultimo *Hide*, oculta la capa seleccionada, esto es muy útil cuando se trabaja con varias capas. (Figura 26).

Figura 26. Área de trabajo con opciones de visualización de capas



En ocasiones, es necesario cambiar características físicas de las pistas, el diámetro de los agujeros, el tamaño de las islas, entre otros cambios; por lo tanto se debe de seguir el siguiente procedimiento: En primer lugar debe de utilizar la lupa para magnificar el objeto a modificar, esto para mayor comodidad y no modificar objetos aledaños. En segundo lugar marcar con el ratón el objeto a tratar, luego posicionarse arriba del objeto y con el botón secundario del ratón aparecerá un menú, ubicar las propiedades del objeto y aparecerá un submenú con distintas características, figura 27 y 28.

Figura 27. Propiedades de los elementos del diagrama

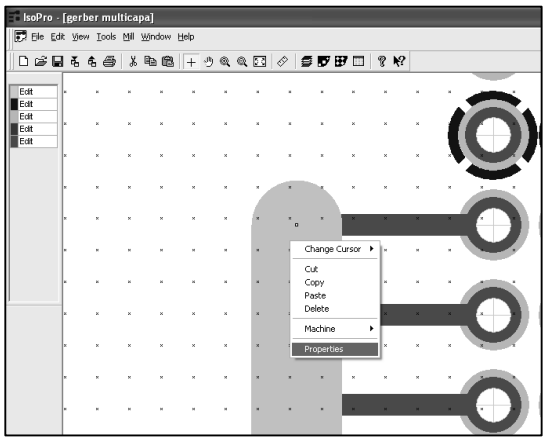
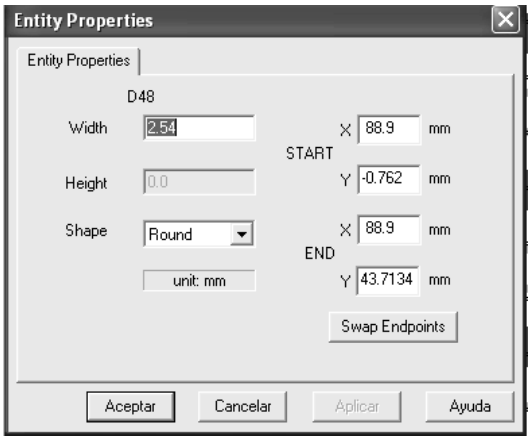


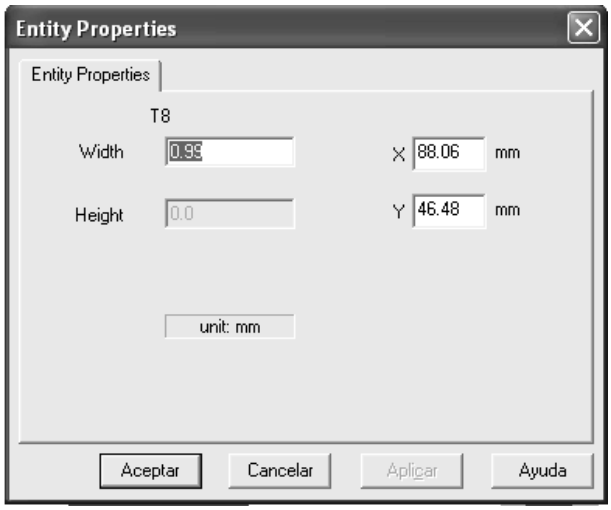
Figura 28. Menú de las propiedades de los elementos del diagrama tipo D



En las propiedades del objeto seleccionado nos indica las características del mismo, por ejemplo, la letra D indica que es una pista y los números siguientes indican el orden correlativo de las pistas del diagrama. También muestran el ancho de la pista que es modificable según las necesidades del usuario, figura 29. *Shape* da la forma que puede tener la pista, ya sea de bordes redondo, cuadrado, entre otros. También los indica donde inicia y donde termina la pista en los ejes X y Y del área de trabajo.

En el caso de las islas, en el menú de las propiedades podemos elegir si sean cuadrados o circulares, según la aplicación, y son identificadas con el código DXX. En cambio el agujero para el barrenado, se identifica con la letra TXX, que únicamente se puede modificar su diámetro y su posición en el espacio con las coordenadas X y Y. El sistema que están siendo medidas se puede observar en *Unit,* la cual puede ser en Sistema Internacional o Sistema Ingles.

Figura 29. Menú de las propiedades de los elementos del diagrama tipo T



Al hacer las modificaciones requeridas, el diagrama al ser calado en el cobre es la imagen de lo que se diseña en el programa, para ello, existe la función *Mirror*, figura 30, la cual da la imagen de la figura que marquemos con el ratón. Esta función se encuentra en la barra de herramientas *Edit* > *Mirror.* Para utilizar esta función se tiene que verificar que todas las capas estén en modo de que puedan se editadas, luego se debe de marcar los objetos y de último aplicar la función *Mirror,* figura 31. Hay que tomar en cuenta que al encontrar la imagen del circuito, se mueve dentro del área de trabajo, por lo que hay que reubicar en el punto deseado de trabajo.

Figura 30. Demostración del uso de la función *Mirror*

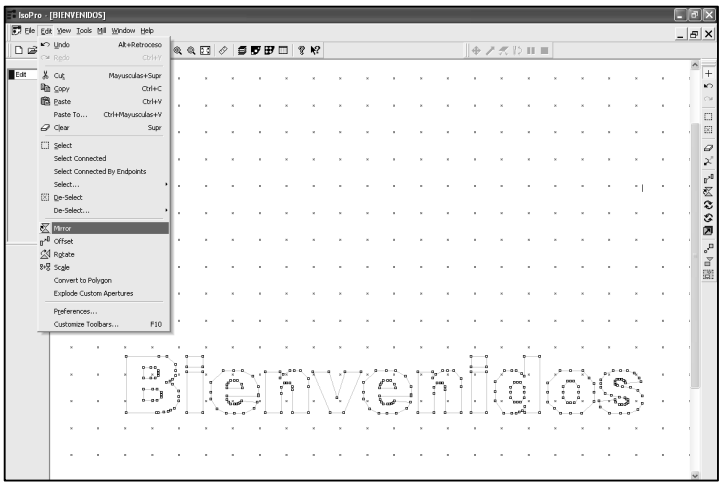
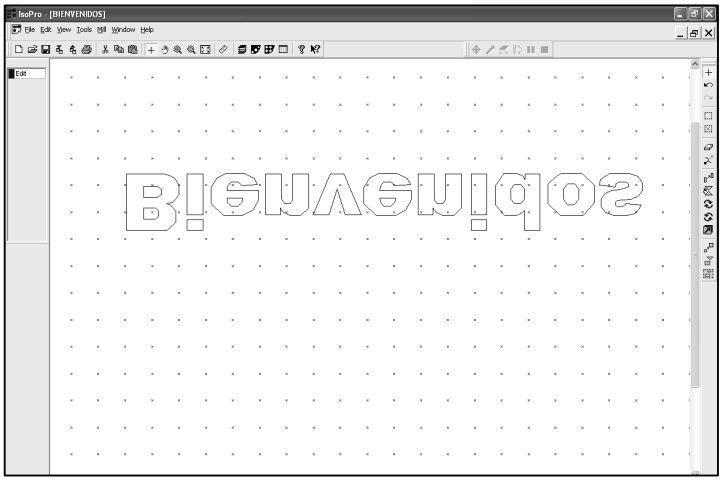


Figura 31. Figura con la función *Mirror* aplicada



El siguiente paso a seguir, ya que el diagrama no requiera modificaciones, es crear la capa de aislamiento. Esta capa de aislamiento se utiliza para separar las pistas, del exceso de cobre que no sirve en el circuito. Para ello debe de marcar todas las capas que se tiene en el proyecto, y ubicar la función *Isolate*, figura 32, cuyo acceso rápido es ctrl+I, ubicada en la barra de herramientas *Tools* > *Isolate,* el cual va a generar un menú con las capas que se desea aislar, figura 33, así como una lista de las herramientas que se tiene disponible.

Figura 32. Ubicación de la función *Isolated*

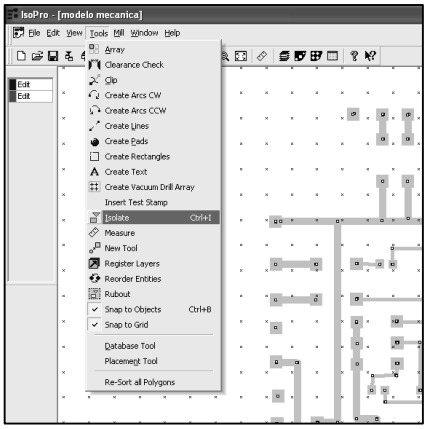


Figura 33. Menú de la función *Isolated*

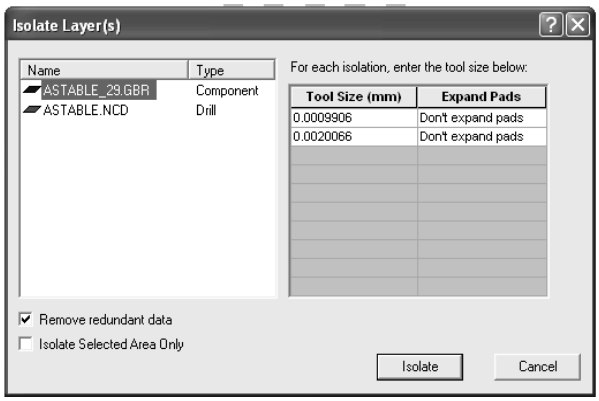
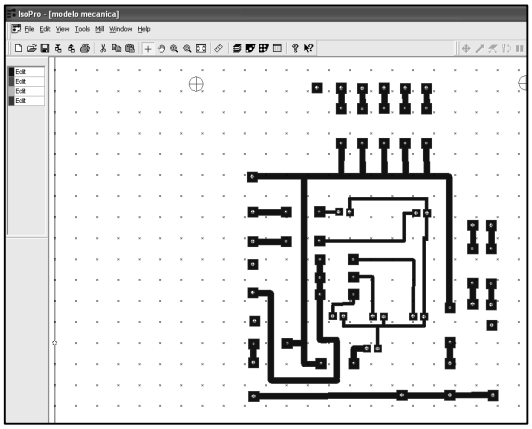


Figura 34. Creación de capas



En la figura 34 se generaron dos capas más, las cuales nos ayudaran al momento del desgastado en el cobre. Cuando se generan las capas de aislamiento, hay que tomar en cuenta que mientras menor es el espacio entre las pistas, menor será la herramienta a utilizar, por lo que se recomienda que el espacio entre las pistas sea lo más uniforme posible y a una distancia prudente. Ahora que se definió el circuito a desgastar en el cobre, es necesario quitar el excedente del cobre, esto se realiza con la función *Rubout*, figura 35, ubicado en la barra de herramientas Tools > *Rubout*. Esta generará otra capa, la cual será desgastada y finalmente quedará únicamente el circuito en el PCB, figura 36.

Figura 35. Ubicación de la función *Rubout*

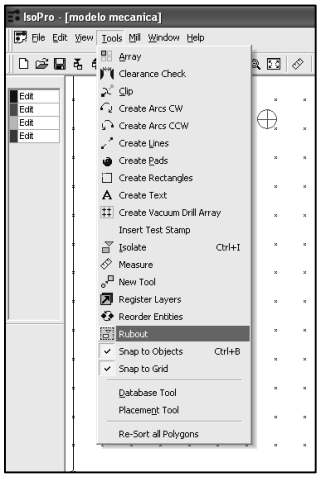
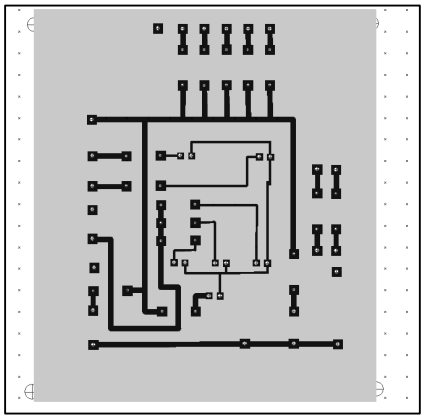


Figura 36. Función *Rubout* aplicada



3.5 Fundamentos de Edwin XP

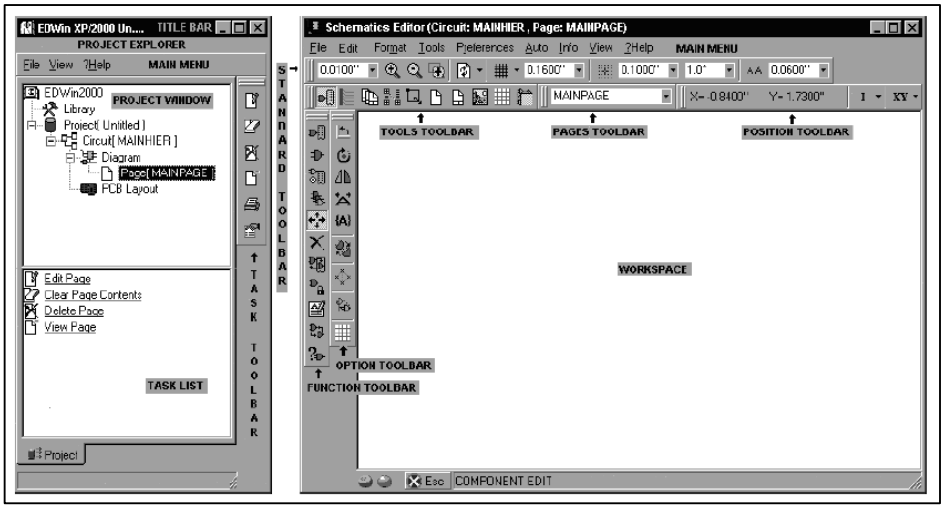
Edwin (*Electronic Design for Windows*) es un paquete CAD que corre bajo la plataforma de Windows. Dicho programa posee cinco módulos, que pueden interactuar entre sí. Dichos módulos son: el editor esquemático, que es donde el diseñador realiza el diseño con la simbología correspondiente; el editor de diseño, donde prepara el PCB, así como los agujeros y pistas; módulos de librería, donde el usuario puede crear sus propios diagramas esquemáticos y simbólicos; y por último el simulador, donde se puede realizar pruebas de funcionamiento del circuito, así como comportamiento térmico y electromagnético entre los dispositivos de la placa y su entorno.

3.5.1 Editor esquemático

El editor esquemático es donde se realiza el diagrama esquemático de los circuitos electrónicos, empezando primero con la configuración del tamaño de la hoja de trabajo, la elección de los circuitos correspondientes, elegir el tipo de encapsulado de los componentes, conexión entre los dispositivos y finalmente colocar etiquetas y mensajes necesarios para entender el circuito.

Para todo lo anterior, este programa contiene el llamado Interfaz Gráfico Del Usuario (IGU), que se muestra en la figura 37, permite al programador interactuar con el recurso de manera amigable.

Figura 37. Interfaz gráfico del usuario (IGU)

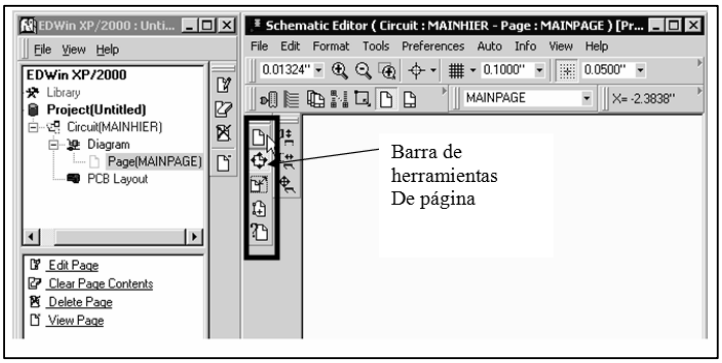


El IGU está comprendido en dos secciones, el Explorador de Proyectos y el Área de trabajo. El primero contiene un listado de tareas, llamadas Funciones, que esta cambiará dependiendo del área de trabajo que se está trabajándole cual se utiliza para el diseño del Editor Esquemático, Editor de Diseño, Módulo de Librería y el Simulador. Para el primero, las barras de herramientas se dividen en nueve grupos, los cuales son: barra de herramientas estándar, de posición, de tamaño, de visualización esquemática, de páginas, de texto, de objetos, de alineación, de funciones y de opciones.

Antes de iniciar con el diseño del circuito el usuario debe de configurar la página, dependiendo de sus necesidades, esto se logra con la barra de herramientas de páginas.

El tamaño límite para una hoja es de una de 4 metros x 4 metros con capacidad de 99 hojas, puede trabajar en formato Americano y Europeo, en tamaños A, B, C, D, E y A4, A3, A2, A1, A0 respectivamente, la barra de herramientas de pagina se muestra en la figura 38. Posteriormente que ya se ha elegido el tamaño del área de trabajo, debe de iniciar con el diseño del circuito.

Figura 38. Barra de herramientas de página



Antes de colocar el primer componente del circuito, debe de seleccionar el la rejilla de puntos, el cual ayuda a observar la distancia y la alineación de los componentes dentro de la hoja, esta función se encuentra ubicada en la barra de herramientas estándar. Para colocar los dispositivos, se selecciona el icono *Open part library* mostrado en la figura 39, luego aparece las funciones de esta tarea, se selecciona la opción *Add Component* (figura 40) donde saldrá una ventana donde el usuario escribe el nombre del componente (figura 41) y automáticamente en el puntero saldrá el símbolo del dispositivo deseado donde será colocado en el lugar requerido, figura 42.

Figura 39. Selección del icono *Open part library*

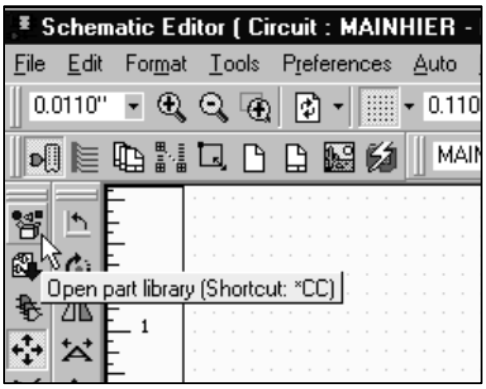


Figura 40. Menú de la opción *Open part library*

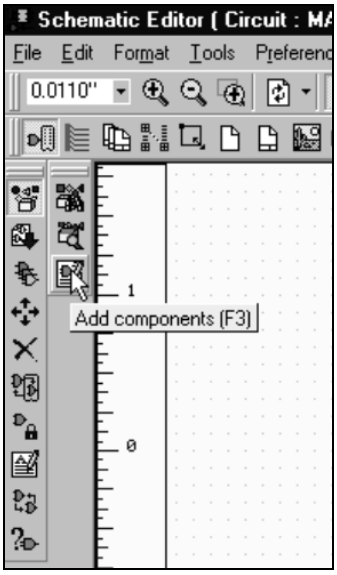


Figura 41. Submenú del icono *Add componentes*

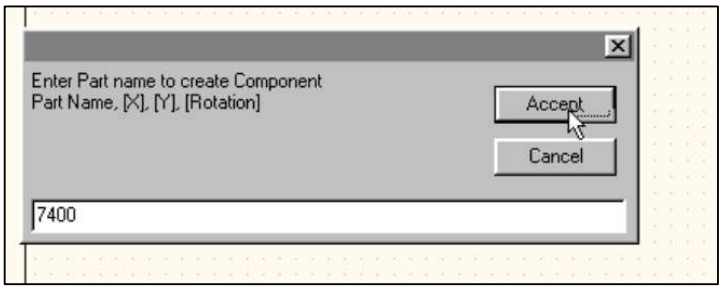
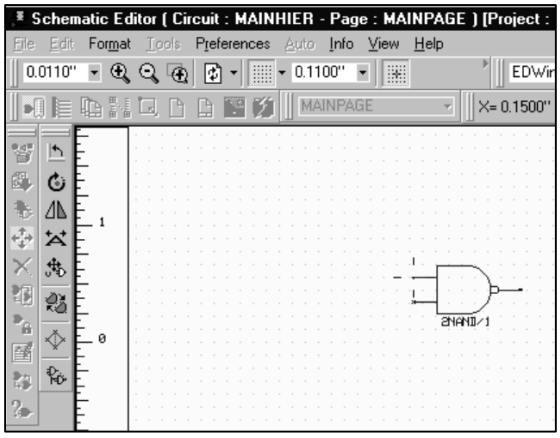


Figura 42. Componente agregado en el área de trabajo



Luego que se han colocado los dispositivos, se debe de tomar en cuenta que, como por ejemplo los circuitos integrados, contienen varias compuertas lógicas dentro del mismo, por lo que hay que definir los grupos para que al momento de realizar el PCB, no exista mal funcionamiento. Para empaquetar, en la barra de herramientas seleccionar *Pack/Unpack* componente (figura 43), el cual ayudará a seleccionar a que integrado pertenece a cada compuerta por medio de un menú de ayuda (figura 44).

Figura 43. Función de empaquetamiento de elementos

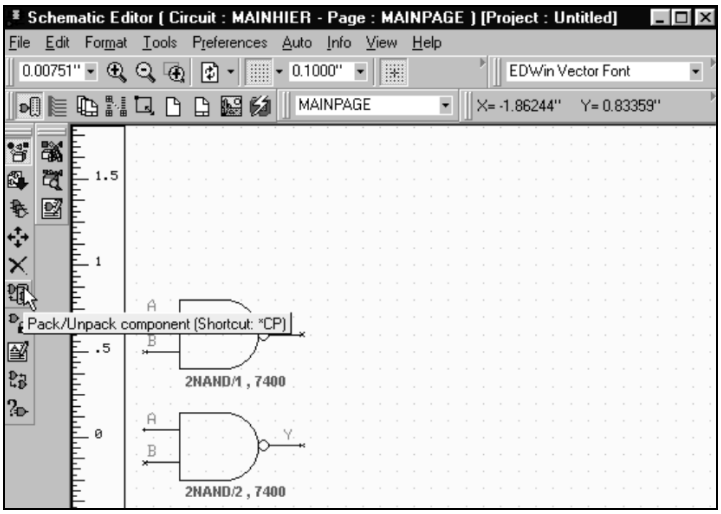
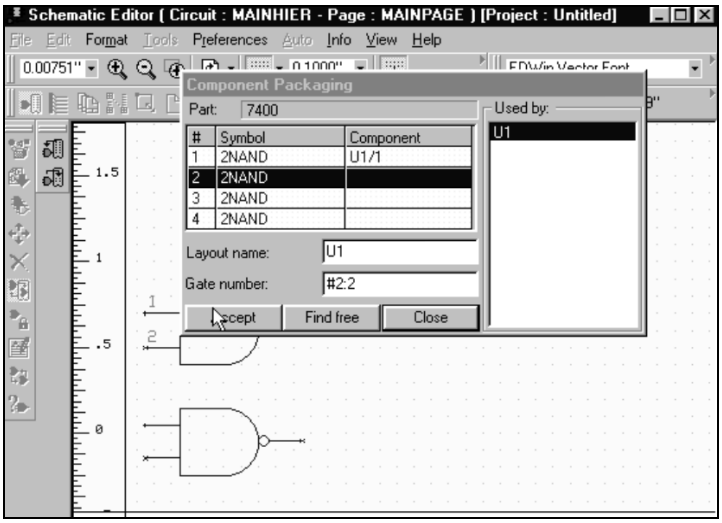


Figura 44. Menú de la función de empaquetamiento de elementos



Ya que los componentes estén en la hoja de trabajo, se debe de conectar para formar el circuito electrónico, esto se puede realizar utilizando ya sea conexiones simples o buses. Para este procedimiento en la barra de herramientas presionar *Wires* (figura 45), aparecerán funciones, la cual para conectar de forma convencional hacer uso de la función *Route Wires* (figura 46), donde aparecerán opciones. La manera de conectar los dispositivos es haciendo un *clic* en la terminal del dispositivo y haciendo otro en hasta donde se requiere la trayectoria (Figura 47).

Figura 45. Función *Route Wires*

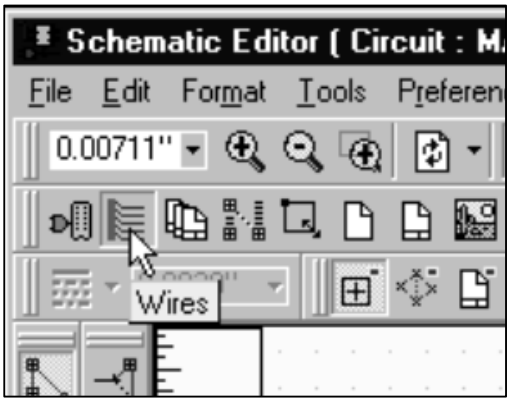


Figura 46. Menú de la función *Route Wires*

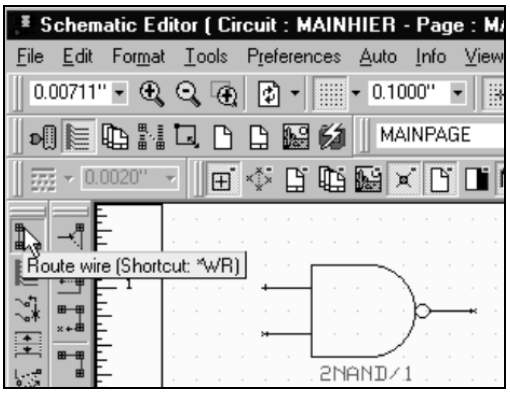
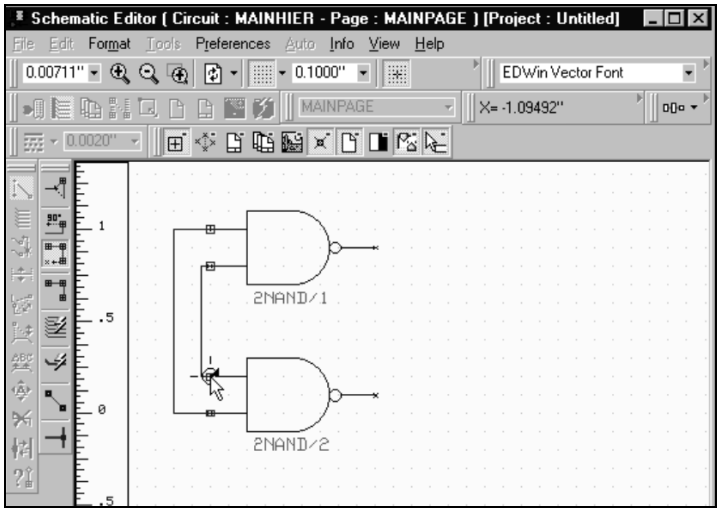


Figura 47. Procedimiento de cableado entre dispositivos



Ya que el circuito esté finalizado, es necesario identificarlo, con el objetivo de que cualquier persona al abrir el archivo reconozca el funcionamiento básico del mismo. Para colocar nombre o etiqueta al circuito seleccionar en la barra de herramientas, la función *Design Notes* (figura 48), la cual desplegará su propia barra de herramientas, de ésta elegir el icono *Create DN Grafic Item* (figura 49), donde el usuario encontrara diversas opciones para identificar sus proyectos.

Figura 48. Ubicación de la función *Design Notes*

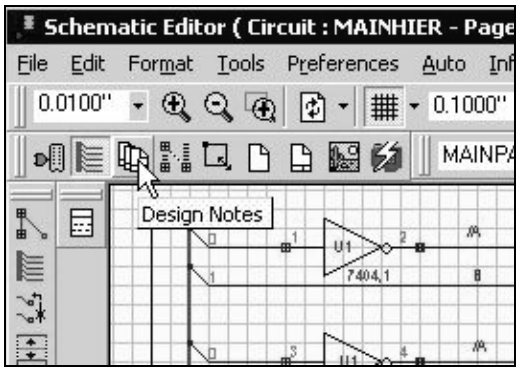
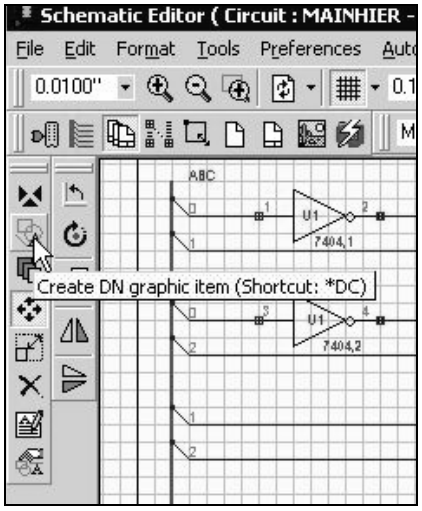


Figura 49. Menú de la función *Design Notes*



3.5.2 Editor de diseño

En el editor de diseño, es donde se genera los archivos necesarios para realizar el diagrama para el PCB, tales como ancho y largo de las pistas, diámetros, lados o formas de las islas, diámetros de los agujeros, tamaño físico de los componentes como de sus accesorios, buses, terminales, tamaño de la placa así también de las caras a utilizar en el PCB. Por medio del editor esquemático se puede importar al editor de diseño *Proyect Explorer > PCB Layout > Edit PCB Layout*.

El ambiente de trabajo de este módulo se muestra en la figura 50. Las dos barras de herramientas principales de este módulo son *View layout display y Tools (object Oriented)* mostradas en la figura 51 y 52 respectivamente. .

Figura 50. Ambiente de trabajo del módulo *PCB Layout*

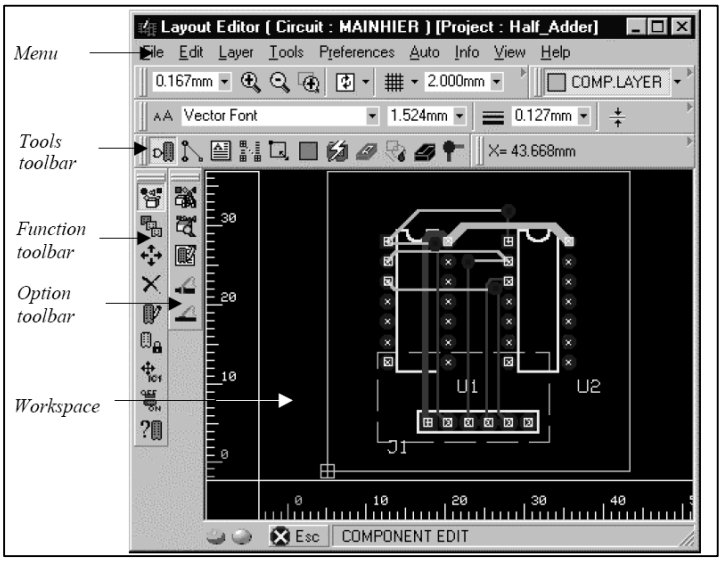


Figura 51. Barra de herramientas de *View layout display*

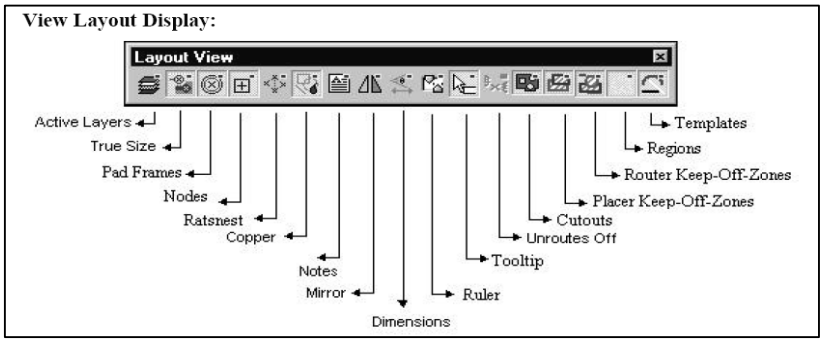
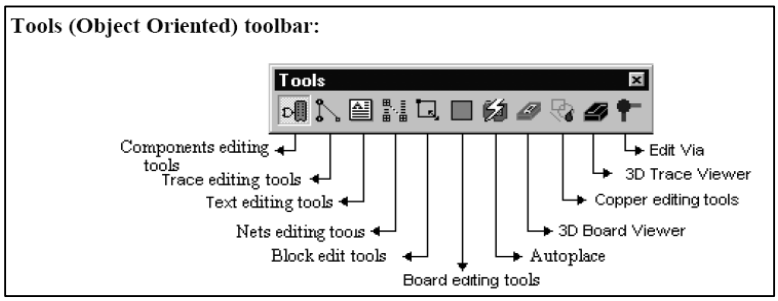


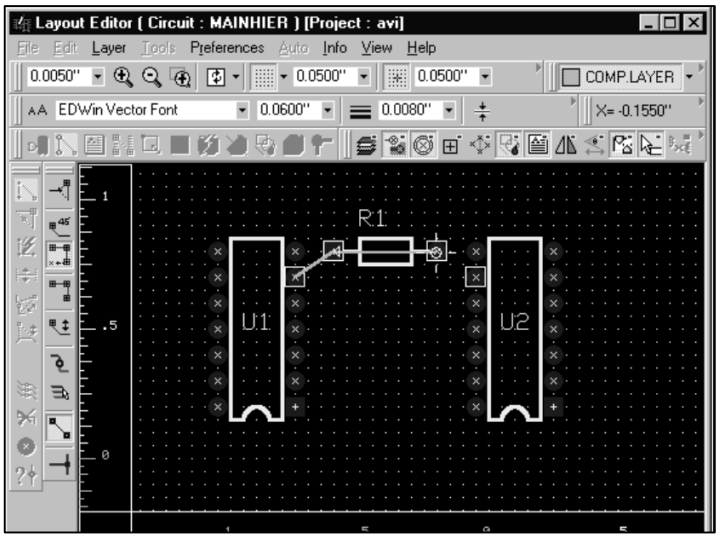
Figura 52. Barra de herramientas de *Tools (object Oriented)*



El tamaño máximo para un PCB es de 4metros x 4 metros, igual que el módulo anterior, soporta los formatos americano y europeo. Cuando se importa el proyecto del editor esquemático, aparecen ya los dispositivos empaquetados, solo se debe de colocar de tal manera que las pistas no se crucen entre sí, respetando el diagrama original. Al procedimiento de conectar los dispositivos por medio de pistas se le llama comúnmente Ruteo, el cual se debe de efectuar inmediatamente al poseer los dispositivos en la hoja de trabajo.

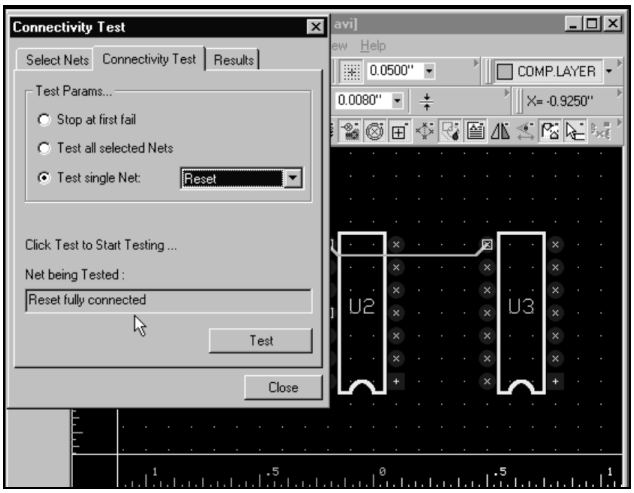
Existen dos tipos de ruteo, manual y el automático. El primero es el recomendado debido a que el usuario elige la posición de las pistas, cuidando la estética del PCB. En cambio el ruteo automático o autoruteo realiza la conexión sin cuidar aspectos importantes al momento de la presentación final del trabajo. Para realizar el ruteo manual, hay que dirigirse a Route Trace > Pin to pin. Para iniciar con la selección anterior, únicamente posicionar el ratón del computador sobre la terminal del dispositivo y finalizar en el otro extremo donde se requiere (figura 53).

Figura 53. Demostración del proceso de ruteo



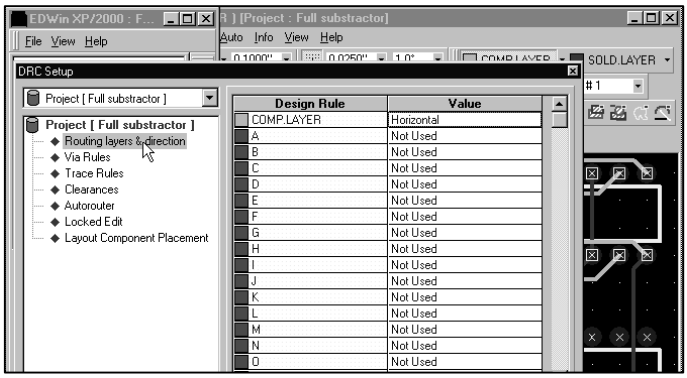
Cuando el circuito deseado ya está completamente conectado, existen errores que son comunes en este tipo de programa, que son la conexión falsa entre dispositivos, dos islas en el mismo lugar, corto circuito entre pistas o islas. Por ello existe una función llamada *Connectivity Test*, figura 54, que se encarga de revisar continuidad eléctrica entre las pistas e islas. Esta función se encuentra ubicada en *Tools* > *Nets.* Esta función posee la capacidad de revisar todas las conexiones del circuito o las que se le indique, esto depende de las exigencias de usuario.

Figura 54. Revisión de continuidad eléctrica de las pistas



Cuando el usuario haya finalizado de realizar el circuito, o sea las pistas, islas y los agujeros, es necesario realizar la última prueba, la cual consiste en una revisión general del estado del PCB, antes de fabricarlo, esto quiere decir que realiza una profunda revisión en lo que son las pistas, islas y dispositivos, que no exista ningún cortocircuito, mala conexión de cualquier tipo, tamaño del PCB, entre otros. Esta función se denomina *Design Rule Check*, ubicada en la dirección Proyect > Proyect Design Rules y se abrirá una ventana llamada DRC, donde puede efectuar dicho función de forma manual, semiautomática y automática, figura 55.

Figura 55. Revisión de conexiones eléctricas y físicas del circuito

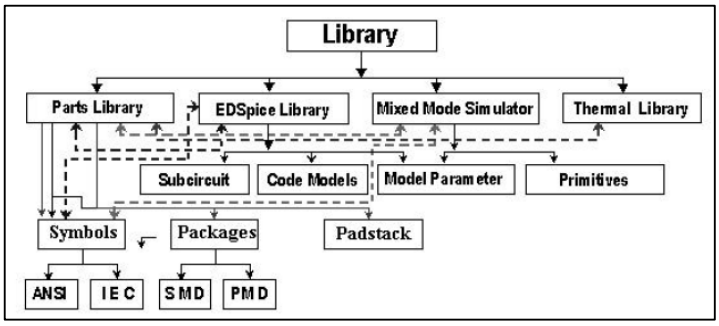


3.5.3 Módulo de librería

Este módulo consiste en dos partes importantes, uno es la Librería de Simulación, la cual permite enlazar con cada simulador y analizador a su respectiva librería, cuyas extensiones son: *.DLL, *.SBC, *.Sparam y *.Mparam. El módulo de librería permite al usuario con todas sus funciones, crear nuevos elementos, así como modificar los ya existentes características básicas: parte, símbolo y empaquetado, los cuales poseen extensiones **.PART*, **.SYMBOL* y **.PACKAGE* respectivamente, en la figura 56 muestra la relación que existen entre todos los elementos de la librería.

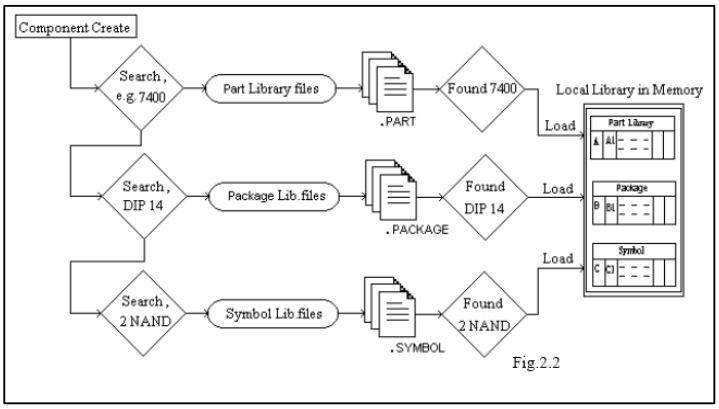
También posee una librería exclusiva para islas, que dependiendo de las exigencias del usuario, puede utilizar las ya existentes, crear una nueva o modificar una ya existente, la extensión de las islas es **.PADSTACK*. En la figura 57 muestra la estructura básica de la construcción de un componente.

Figura 56. Relación entre librerías de EdWinXP



Fuente: **Tutorial de IsoPro.** Capitulo 3. Pag. 3-1

Figura 57. Estructura básica de la construcción de un componente

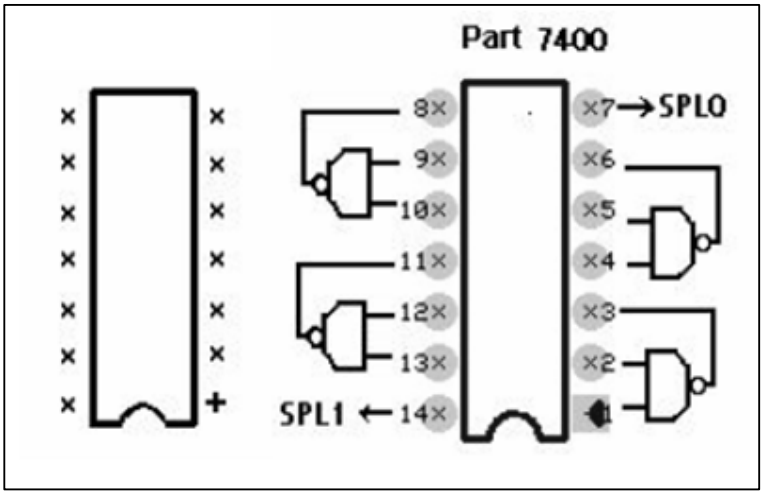


Fuente. EdWinXP. **Manual de EdWinXP**. Pág. 3-2

Durante el diseño del símbolo, hay que tomar en cuenta que se puede crear utilizando las normas ANSI o IEC para simbología eléctrica y electrónica. Para identificar estas dos normas, al inicio de la librería aparece la letra A, que indica que pertenece a la simbología de la norma estandarizada ANSI, o si empieza con la letra I, pertenece a la simbología norma estandarizada IEC. Por ejemplo: A_7400.SYMBOL e I_7400.SYMBOL, las dos números pertenecen a un circuito integrado inversor, pero el primero pertenece a simbología ANSI y la segunda a IEC.

Para realizar el empaquetado, hay que tomar en cuenta el tipo de dispositivo a diseñar, si es de superficie o no, puesto que los elementos son distintos. En la figura 58 se muestra el empaquetado para un circuito integrado no superficial. Para crear los archivos de las partes del dispositivo, es necesario que el símbolo y el empaquetado estén ya creados, debido a que este archivo es la unión de los dos anteriores, lo cual ayuda a relacionar el símbolo con el tipo de empaquetado se va a utilizar en el diseño del circuito.

Figura 58. Símbolo y empaquetado del dispositivo 7400



3.5.4 Módulo de Simulación

Simular el comportamiento de un circuito bajo ciertas circunstancias, necesarias para comprobar su funcionamiento, es una parte importante dentro del diseño del mismo, debido a que le da al diseñador una seguridad de su trabajo, ahorrando tiempo y dinero en la construcción de su PCB. En el módulo de simulación existen dos niveles, el primero llamado Simulación a Nivel de Circuito, se utiliza para desarrollar pruebas de voltaje y corriente, ya sea analógico o digital en las entradas y salidas del circuito, y por medio de instrumentos virtuales propio del programa ver si es el comportamiento idóneo del circuito. El segundo es llamado Simulación a Nivel del PCB, donde se realiza la prueba térmica y electromagnética de todos los dispositivos dentro del PCB, para que no exista interferencia de ningún tipo.

3.2.4.1 Simulación a nivel de circuito

La simulación a nivel de circuito existen dos modos de simulación, Modo Mixto y el Modo *EDSpice*. El primer modo consiste en realizar análisis de transitorios, de corriente directa, de corriente alterna, de Montecarlo y de sensibilidad, estas pueden ser aplicadas en la salida, entrada y dentro del circuito. El Modo *EDSpice* se basa en los estándares de Spice, el cual nos permite realizar análisis más amplios que el anterior debido a la plataforma. Los distintos tipos de análisis que se puede realizar en Modo *EDSpice* son: análisis de transitorios, de señal pequeña de *AC*, función de transferencia, de distorsión, de ruido, sensibilidad de señales *AC/DC* y polos y ceros.

3.2.4.2 Simulación a nivel de PCB

La simulación a nivel de PCB nos permite realizar dos pruebas importantes para el desarrollo del mismo, uno es el Análisis Térmico y el otro es el Análisis Electromagnético. El análisis térmico nos permite revisar la distribución del calor en el PCB, prebendo problemas del mismo tipo en el circuito impreso, esto por medio de su analizador mostrado en la figura 59 y 60. El análisis electromagnético nos permite observar la distribución de las líneas de campo provocadas por el voltaje y la corriente dentro del circuito. Este análisis es importante debido a los distintos efectos que pueda surgir dentro de un circuito impreso, tales como el efecto capacitivo y la inducción electromagnética, estas influencias de campos se pueden observar tal como se muestra en las figura 61 y 62.

Figura 59. Pantalla de análisis térmico estándar

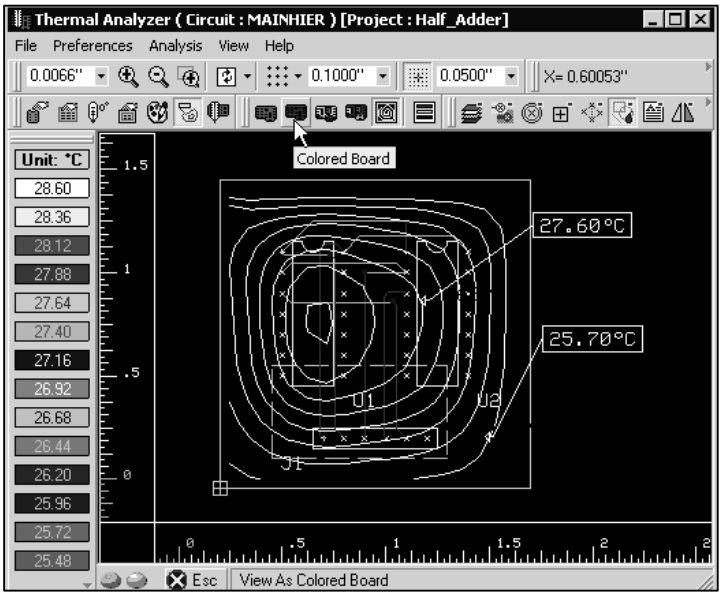


Figura 60. Pantalla de análisis térmico a color

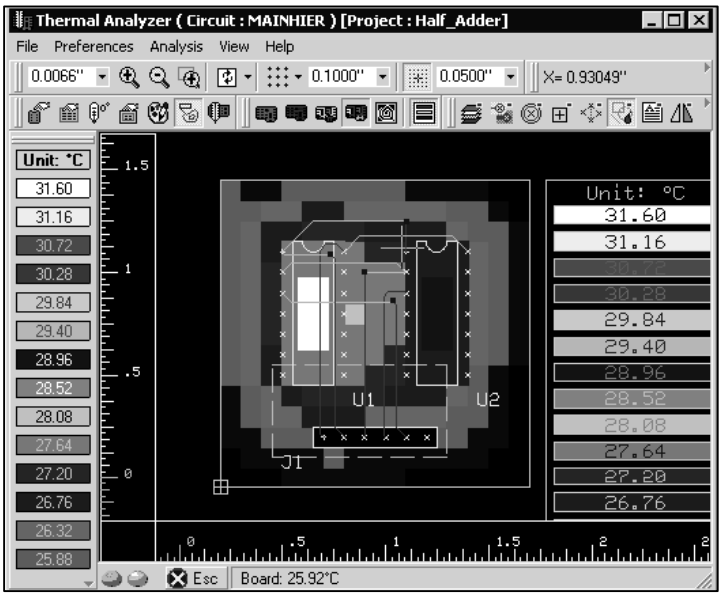


Figura 61. Pantalla de análisis electromagnético estándar

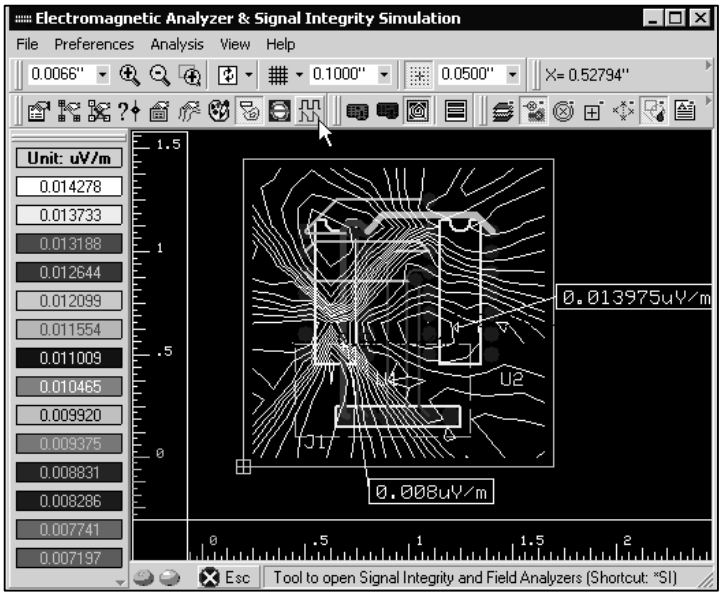
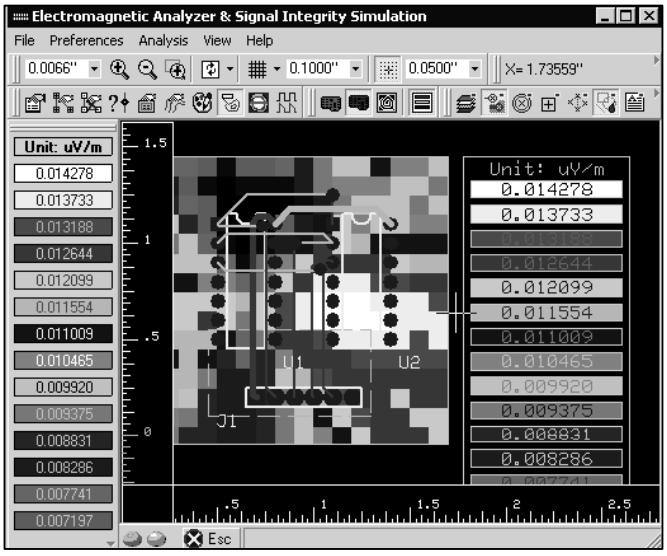
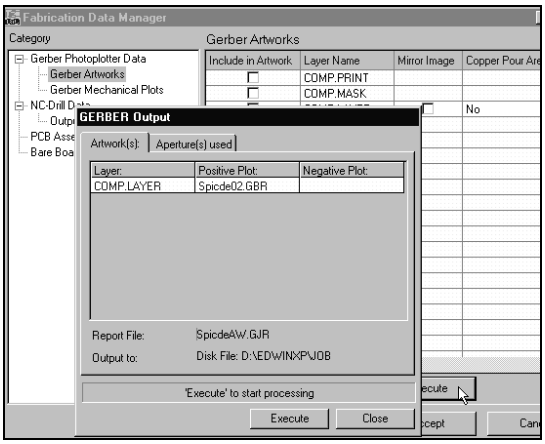


Figura 62. Pantalla de análisis electromagnético a color



Cuando en la realización del PCB en el programa CAD, en este caso IsoPro, se hace necesario crear los archivos G para elaborar el circuito en el sistema CAM. Para ello en dicho programa en el menú principal buscar *Fabrication Manager > setup.* Allí abrirá una ventana llamada *Fabrication Data Manager*, figura 63. De allí se selecciona los archivos Gerber, ejecutando el programa, todos los datos necesarios de la placa son convertidos a los archivos necesarios para utilizarlo con el programa CAM o los que el fabricante necesita en ese momento.

Figura 63. Generación de archivos G



3.6 Proceso de fabricación del circuito en el PCB

Antes de realizar el circuito en el PCB, es necesario diseñarlo en un programa tipo CAD, en este caso con Edwin XP, que es donde el diseñador elige la forma de la placa, para que posteriormente lo convierta en archivos G, para que el equipo CAM, interprete estos datos. En el caso del equipo CAM, entre el programa IsoPro, que es donde se visualiza la forma de las pistas que la máquina va a desarrollar.

Lo único que debe de tomar en cuenta el fabricante es que para el desarrollo de PCB se necesita de ciertas reglas básicas para su construcción. El objetivo de este tema no es citar las normas americanas y europeas para la fabricación de circuitos impresos, pero se mencionarán las principales que serán suficientes para que el circuito funcione de forma correcta.

3.6.1 Normas básicas para la realización de un PCB

El ancho de las pistas conductoras debe de ser suficiente para no crear resistencia al paso de la corriente, pero sin tener una anchura excesiva que puede producir pérdidas debidas a la resistencia óhmica de las conexiones. Las pistas de masa o tierra deben ser lo mas anchas posible. Las pistas de los circuitos digitales deben ser lo mas uniforma posible en lo que se refiere a ancho y la separación entre ellas. En circuitos analógicos la separación entre pistas depende de la tensión de ruptura. Las pistas de alimentación deben de ir en la misma cara de la placa. Cuando el PCB esté terminado, bañar la placa con un barniz aislante, para evitar los efectos entre pistas.

Los efectos comunes son las interferencias electromagnéticas (EMI) provenientes del exterior o propias de la geometría del circuito, tales como impedancias entre alimentación y tierra, efecto capacitivo e inductivo, etc., las cuales no se pueden eliminar en su totalidad, pero se puede reducir hasta un punto que no interfiera con el funcionamiento del circuito. Para ello se recomienda que la alimentación, exista un capacitor de desacople de tantalio o de aluminio por placa, o en caso de existir circuitos integrados, estos deben de llevar uno.

Cuando en una misma placa existen circuitos analógicos y digitales, es necesario realizar una tierra analógica y una digital, no es conveniente trazar las pistas de circuitos lógicos cerca de señales analógicas de baja señal. El ancho mínimo para la pista de alimentación es de 2 mm. Los transformadores de aislamiento, optoacopladores, capacitores, etc., deben de conectarse cerca de las terminales de entrada y salida.

Las interferencias electromagnéticas también son producidas por los circuitos digitales que trabajan a altas frecuencias, así como los circuitos analógicos que manejan elevadas corrientes. En caso de los circuitos de alta frecuencia, la pista del reloj, oscilador o cristal debe de ser lo más corto posible, por ello en aplicaciones especiales, se utiliza placas multicapas para reducir su efecto y para la colocación de los circuitos integrados se necesita bases de bajo perfil. No trazar las pistas de reloj junto a las de control como las de los microcontroladores y microprocesadores. Un efecto muy común en los circuitos es la llamada diafonía o acoplamiento electromagnético.

La diafonía o acoplamiento electromagnético puede ser de dos tipos: diafonía capacitiva y diafonía inductiva. La primera consiste en la interacción del campo eléctrico que produce un pico de corriente transitoria dentro del circuito. La diafonía inductiva consiste en la interacción del campo magnético que produce un pico de tensión transitoria en las pistas del PCB. Para evitar la diafonía, se recomienda que las pistas de señal deben de poseer un ancho mínimo de 0.5 mm, con una placa con sustrato de baja constante dieléctrica.

También hay que tomar en cuenta la posición en el espacio de los componentes dentro de la placa. Si se está diseñando un circuito que trabaja señales muy precisas, leer las hojas técnicas para conectarlo de tal manera que no interfiera su funcionamiento por cuestiones internas o externas de la placa. Se recomienda colocar los dispositivos de forma lógica y ordenada, de tal manera que todos lleven un mismo sentido, figura 64.

No se recomiendan los puentes, peso si es necesario, que estos se coloquen en la cara de los dispositivos, con un alambre conductor aislado. Los elementos de potencia deben de tener suficiente espacio para disipar el calor y no afectar a los que estén alrededor. Las pistas con un ángulo de 90^{O} causan EMI, por lo que se recomienda de un quiebre de 45 grados entre las pistas involucradas (figura 65 y 66).

Figura 64. Colocar los dispositivos de forma lógica y ordenada

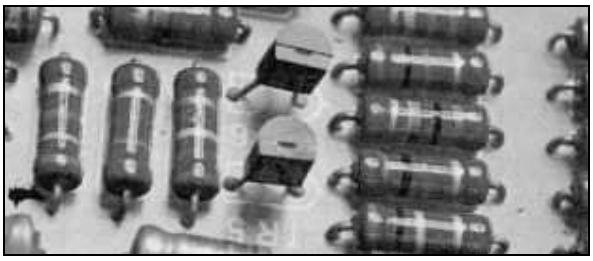


Figura 65. A: Pista mal diseñada. B: pista bien diseñada

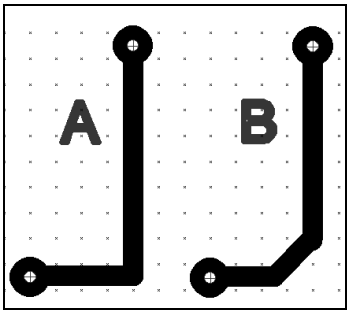
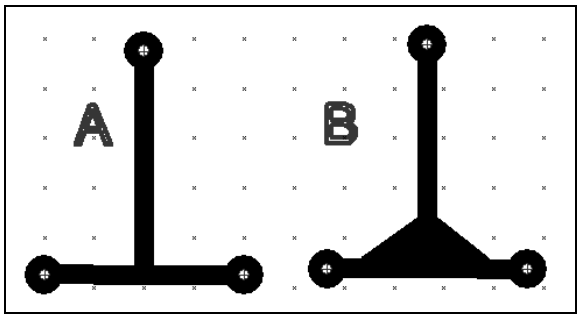


Figura 66. A: pista mal diseñada. B: pista bien diseñada



Se recomienda que las islas circulares posean un diámetro de por lo menos el doble del ancho de la pista que en él termina. En general, una pista de 0.8 mm puede soportar, dependiendo el espesor de la misma alrededor de 2 amperios, 2 mm unos 5 amperios y 4.5 mm 10 amperios, no olvidando que la distancia mínima aceptable entre pistas o pistas e islas son de 0.8 mm. La distancia entre pistas y los bordes debe de ser de por lo menos 5 mm. Todos los componentes se colocarán paralelos a los bordes de la placa. Pasar pistas entre las terminales de componentes activos, tales como transistores, circuitos integrados, tiristores no está permitido. (figura 67 y 68).

Figura 67. A: pista mal diseñada. B: pista bien diseñada

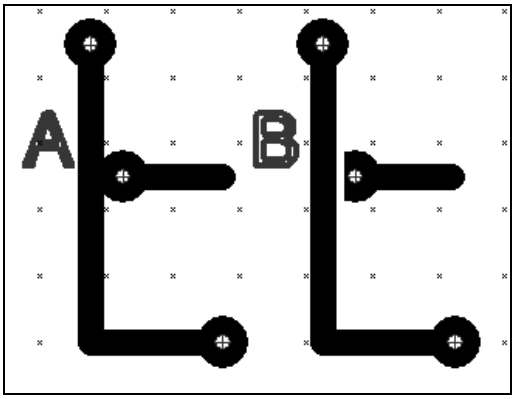
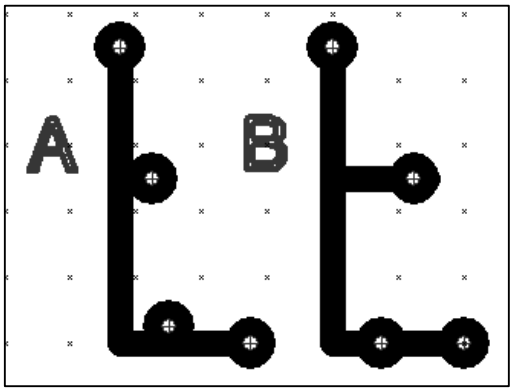


Figura 68. A: pista mal diseñada. B: pista bien diseñada



Al momento de soldar los componentes, hay que dejar una distancia de 2.5 mm a 5 mm del cuerpo del dispositivo del punto de soldadura. Las tarjetas ya construidas deben ir sobre un soporte, lo cual muchas veces requiere un agujero, éste debe de ser de 3.5 mm de diámetro en cada esquina de la placa. El nodo puede ir sobre la pista o de forma tangencial. Estas islas no deben de poseer áreas grandes de cobre, puesto que al momento de soldar el dispositivo, se corre el riesgo de que el estaño se escurra formando cortocircuitos, figuras 69 y 70.

Figura 69. A: pista mal diseñada. B: pista bien diseñada

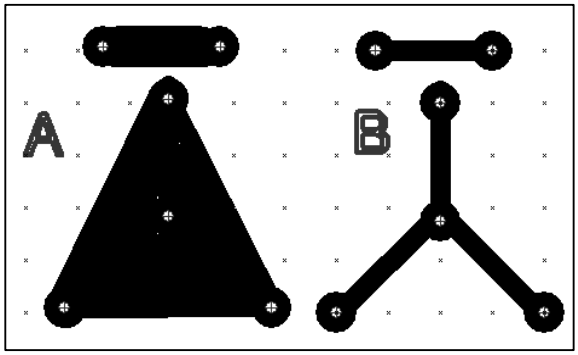
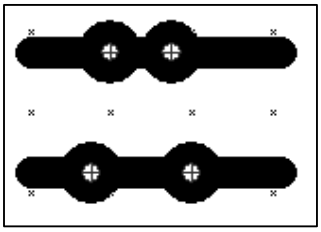


Figura 70. A: pista mal diseñada. B: pista bien diseñada



Para pistas de tierra o de blindaje sobre grandes extensiones de cobre, se recomienda diseñar un cuadriculado fino o en diagonal. Si es necesario unir dos nodos, siempre debe de existir una pista entre ellos para evitar que al soldar una isla, se desuelde la otra, figura 71.

Al trazar pistas con un punto en común no debe de formarse ángulos agudos entre ellos, puesto que provoca problemas al soldar, figura 72. El diámetro de los agujeros define el tamaño de la broca, *endmill, t-mill* o ruteador a utilizar, por ello se establecen dimensiones similares para no estar cambiando de herramienta en cada momento.

Figura 71. A: pista mal diseñada. B: pista bien diseñada

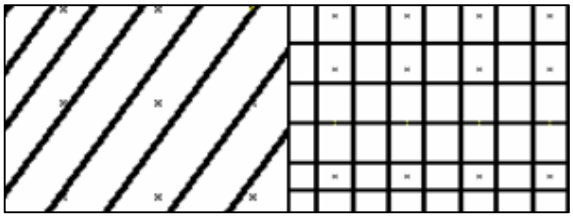
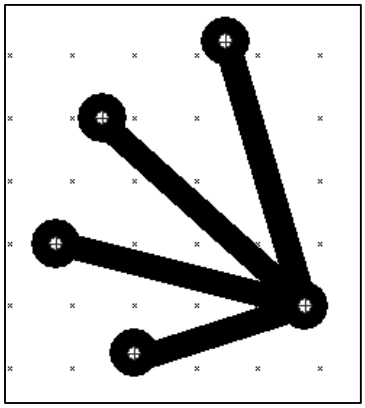
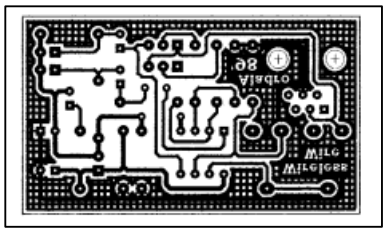


Figura 72. Pista mal diseñada



Para los agujeros de los circuitos integrados y componentes de baja potencia se recomienda que la broca sea de 39 mils equivalente a 1 mm aproximadamente con una tolerancia de 0.1 %. Para la inserción de terminales, potenciómetros, en pocas palabras, pines más gruesos que los anteriores, se necesita una broca de 1.25 mm con una tolerancia del 0.1 %, equivalente a 49 mils. Para los tornillos de fijación de la placa y para otros componentes, es necesario una broca de 3.5 mm equivalente a 138 mils. El tamaño de la herramienta para desgastar el cobre depende de la forma del circuito. Figura 73.

Figura 73. Pista de circuito finalizado respetando normas básicas



3.6.2 Descripción breve de la máquina elaboradora de circuitos electrónicos

La máquina para elaborar circuitos impresos es llamada *Quick Circuit* 700, figura 74, la cual será abreviada con las iniciales QC, está diseñada para trabajar prototipos de circuitos, no para la producción en masa. El QC, básicamente está compuesta por la base de labrado y el equipo que comunica ésta con la computadora. El equipo que comunica con la computadora posee tres partes, una es el que controla las revoluciones del motor, otra que recibe la información de la computadora e interpreta los datos en movimiento de la base de labrado y la tercera que permite la regulación de una aspiradora, figura 75.

Figura 74. Base de labrado

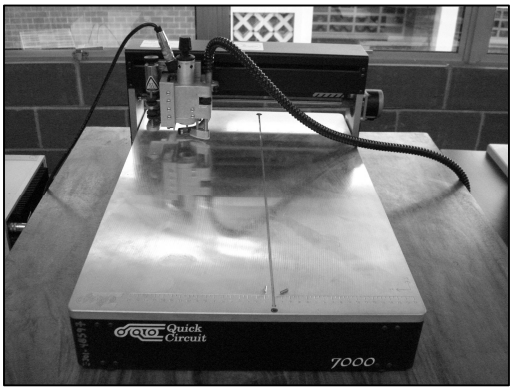
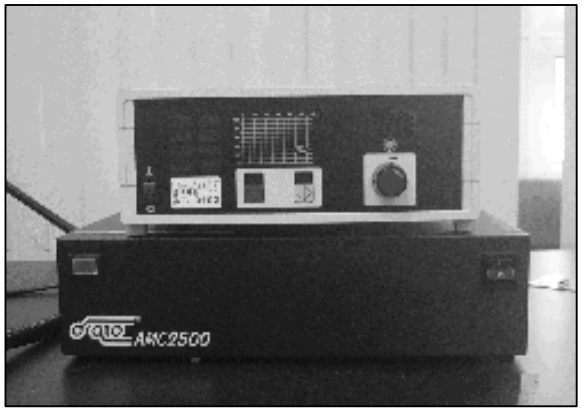
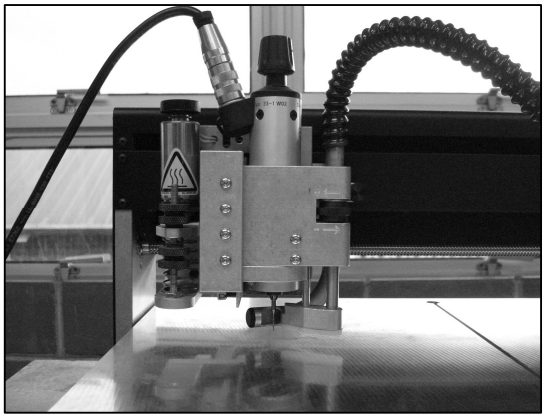


Figura 75. Regulador de velocidad del motor e interfaz de la computadora



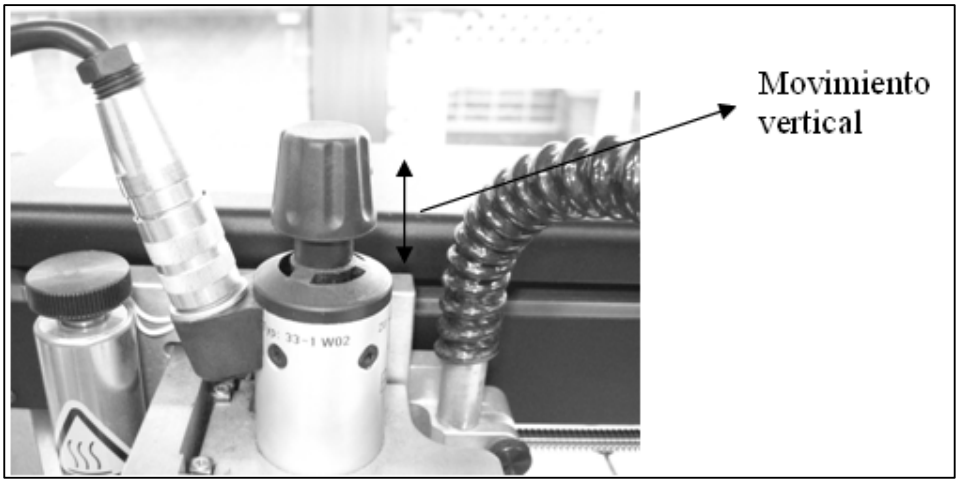
En la figura 76 se observa de cerca el motor donde se aseguran las herramientas. En la parte izquierda se observa un solenoide que permite al motor bajar o subir dependiendo de las necesidades del usuario o del programa. Por otra parte del lado derecho se encuentra un tornillo que permite ajustar la altura de la herramienta con más precisión que el solenoide. Este solenoide no se debe de mover demasiado puesto que se calienta cuando el avance de la cabeza es demasiado, reduciendo así su vida útil.

Figura 76. Vista del motor con sus accesorios



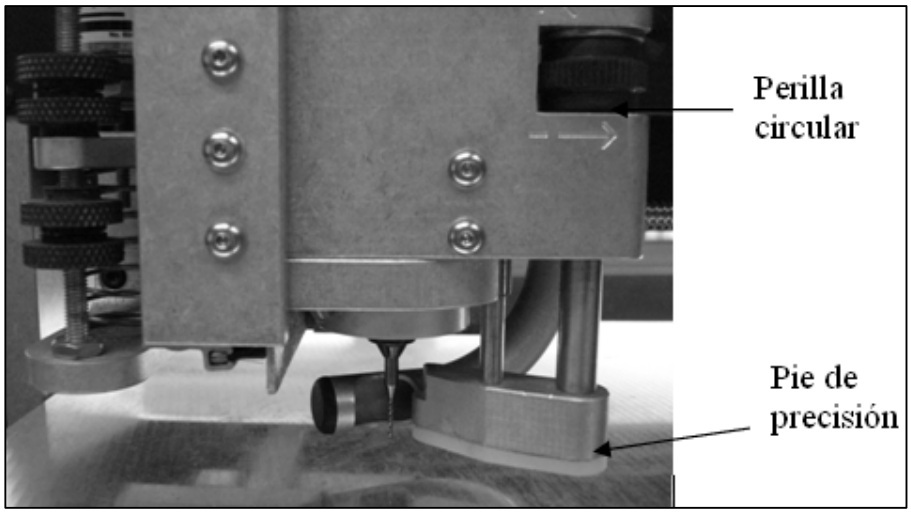
En la parte superior se encuentra un tornillo que permite asegurar la herramienta a trabajar, figura 77, hay que hacer la observación que para asegurar la pieza puesta en la cabeza del motor, es necesario jalar hacia arriba, si no se realiza esta acción, al momento de trabajar el motor, debido a la velocidad del mismo va soltando la pieza de tal forma que ésta corre el riesgo de ser dañada permanentemente. También existe una manguera, la cual es una aspiradora de vacío, la que atrapa todo el excedente que el trabajo de taladrado y desgastado realiza en el proceso de labrado.

Figura 77. Perilla para asegurar herramienta



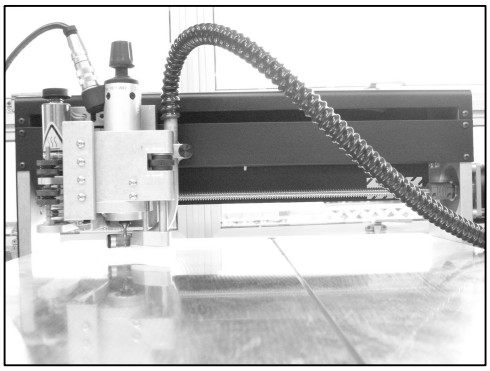
Del lado izquierdo también se observa una perilla circular, figura 78, su función es mover el pie de precisión de forma vertical, la cual limita el movimiento también vertical del motor, la cual permite al operario calibrar la profundidad con que la herramienta puede trabajar. Al trabajar con la maquina el solenoide por su funcionamiento se calienta, lo cual el operario debe de estar pendiente., Si dicho elemento a muy alta temperatura, se recomienda detener el proceso y reiniciarlo cuando este se encuentre a una temperatura razonable. Esto ocurre cuando el circuito es demasiado grande o no se está dando buen uso de la misma.

Figura 78. Vista de la parte inferior del motor



Todo lo anterior está sujeto en un tornillo sin fin que permite mover en coordenadas X-Y al motor junto con la herramienta, figura 79, mientras que ésta se mueve en el sentido del eje Z. Este tornillo se mueve por medio de un motor de pasos con una precisión de 0.25 mm, permitiendo realizar circuitos con una amplia aplicación industrial. Esta base que sujeta al motor, está sobre lo que es la base de labradora la cual posee dos dimensiones, en eje X mide 39 cm. y en el eje Y dependiendo de la aplicación, puede medir hasta el tamaño total de la cama efectiva que es de 50 cm.

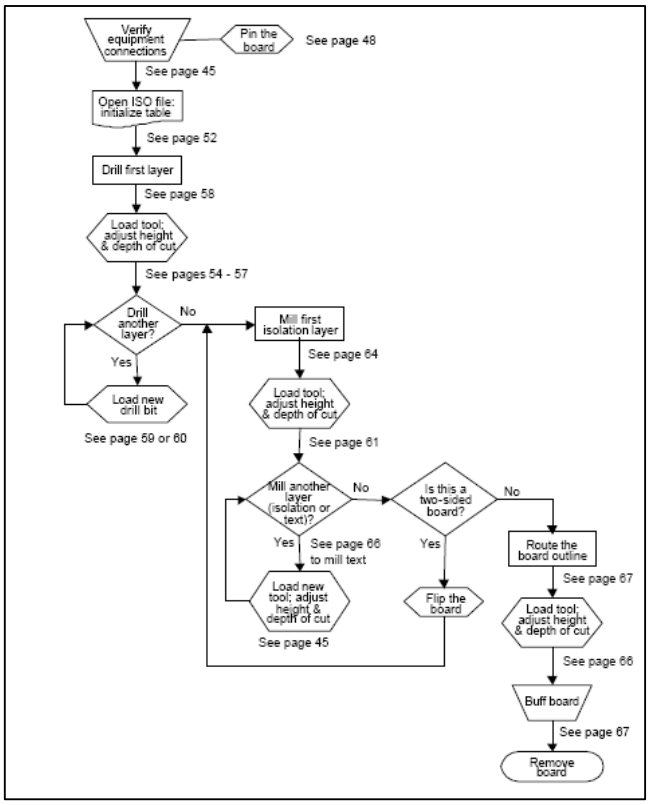
Figura 79. Vista del tornillo sin fin donde se moviliza el motor



3.6.3 Proceso de fabricación del circuito en la placa virgen

El proceso que recomienda el fabricante de está maquina para realizar un circuito impreso se observa en la figura 80. Según el algoritmo, al principio debe de ser revisado si todos los cables están en su lugar, y probar si la aspiradora de vacío esté funcionando correctamente. Luego en la computadora, por medio de IsoPro, se carga el programa a trabajar, cuando se realiza este paso es necesario que la computadora reconozca la maquina así como la posición del motor, por ello en IsoPro se debe de seleccionar *Mill > initialize* en el menú principal.

Figura 80. Proceso global para realizar un circuito impreso

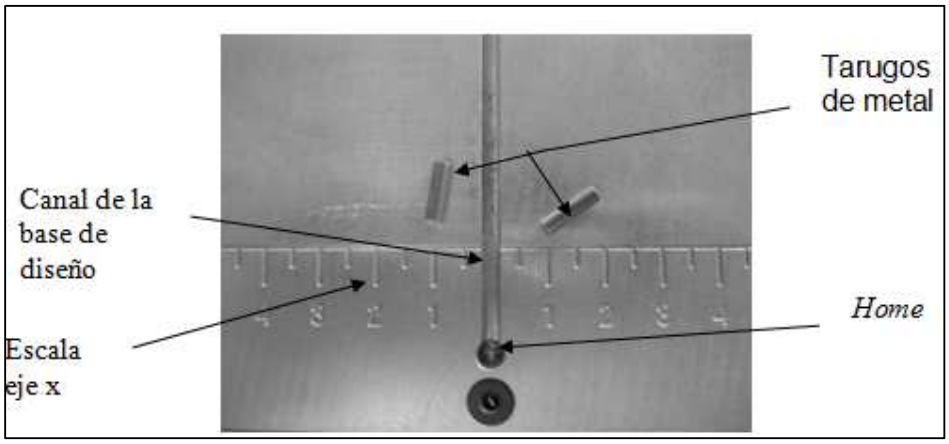


Fuente. *Quick Circuit Systems*. ***User´s Manual***. Pág. 49.

El siguiente paso es colocar la herramienta, esto se logra localizando en el menú principal *Mill* > *Tool change*, y la maquina se posicionará en un lugar donde el usuario puede cambiarla sin ningún tropiezo. Los primeros agujeros que debe de realizar es para colocar la placa de cobre virgen a un soporte compuesto básicamente de papel prensado. Esto debido a que la base de labrado o cama es de metal, y si alguna herramienta toca este punto automáticamente está será dañado permanentemente. Para esto se necesita de unas terminales que permite prensar las dos placas sobre la cama.

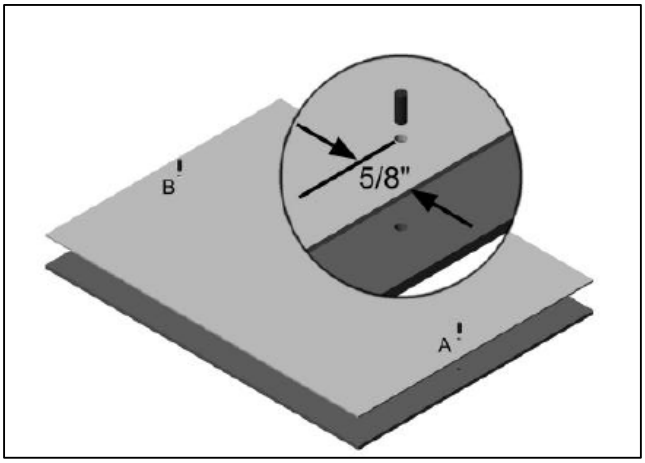
Para cambiar la pieza, en la figura 81 se observa la pieza que debe de girarse para soltar y apretar la broca. Ya puesta la broca manualmente se baja el barreno, de tal manera que se pueda ajustar. El proceso para abrir el agujero es el siguiente, ya que colocó la herramienta correcta, presionar CTRL + H, en el teclado y automáticamente la máquina se posiciona en un punto llamado *Home*. Este punto es un agujero profundo para abrir un hoyo a las placas, manualmente se presiona hacia abajo el barreno para asegurarse que no tope con la cama. Cuando el operario se cercioró que la herramienta no topa con la base de labrado, jala hacia arriba la perilla de ajuste de la broca para asegurarla totalmente.

Figura 81. Vista de puntos importantes del área de diseño



Dejando una distancia del borde de la placa de cobre de 15.8 mm (5/8”), y alineándolo con el eje de la cama, en el menú principal buscar *Mill* > *Sprindle*, el barreno iniciará a girar, cuando el motor ya halla vencido la inercia del arranque, está listo para buscar en el menú Mill > Head up/down, al presionar una vez, el motor automáticamente bajará abriendo el agujero deseado en la placa con el soporte unido. Se abren dos agujeros con así como se muestra la figura 82. Luego del menú *Mill* > *Move*, el motor se mueve hacia un punto contrario, que permite colocar libremente sobre la cama, de tal manera que los agujeros se alineen con el canal que la misma posee, cuando esto se logra, se colocan las bases para asegurar la placa con soporte sobre la base de labrado. El fabricante recomienda que a los bordes de la placa de cobre se termine de asegurar con la cama colocando cinta adhesiva de papel.

Figura 82. Lugar geométrico de los agujeros dentro de la placa



Fuente. *Quick Circuit Systems.* ***User´s Manual***. Pág. 54.

Ya cargado los archivos G dentro en IsoPro, hay que introducir las medidas de las herramientas que se posee, esto para que automáticamente el programa seleccione la pieza más adecuada para su ejecución, esto en *View > Tool Table*.

Para colocar la broca o el ruteador de contorne, se realiza la misma operación para cambiar la herramienta, la diferencia es que se debe de calibrar estas dos herramientas de tal manera que al momento de abrir un agujero tener el cuidado que perfore la placa y parte del soporte. En el menú *Mill* > *Jog* permite colocar la herramienta a un lado de la placa, y manualmente se baja el barreno para observar que cuando este baje de forma automática realice bien los agujeros. Este procedimiento debe de realizarse cuantas veces el programa pida cambio de herramienta.

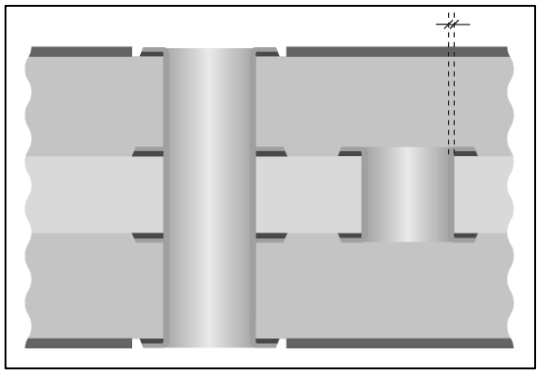
Para el caso de las herramientas que desgastan únicamente el cobre, el proceso de ajuste de pieza es más delicado. Los pasos para cambiar de herramienta es el mismo, lo que cambia es que ahora la *endmill* o el *T-mill* debe de colocarse con la placa, de tal manera que calculando a simple vista al bajar el barreno únicamente desgaste el cobre. Cuando esto se ha realizado, en la función *Jog*, existe dos iconos *Sprindle* y *Head Up/Down*, que nos permitirá una calibración más exacta. La primera función hace que el motor inicie a girar, mientras que la segunda hace que la cabeza suba o baje según su posición. Nunca hay que bajar el motor si no está funcionando, esto provocaría daño permanente en la pieza.

El motor ya está girando y en posición de abajo, entonces con las flechas se puede mover la cabeza, de tal manera que el diseñador observe que tan profundo está desgastando la pieza. Si la herramienta no desgasta mucho o lo contrario, está muy metida en la placa de cobre, es necesario ajustarlo por medio del tornillo que mueve el pie cuya precisión es de 0.010 mm (0.0004”). Esto se mueve hasta que el desgaste sea idóneo para el diseñador de la placa. Si el *endmill* o el *T-mill* puesto en el barreno, desgasta mucho material, estará provocando un desgaste más rápido, requiriendo su reemplazo de manera temprana. Este proceso se repite cuantas veces el programa lo solicite.

3.7 Metalizado

En el proceso de fabricación de circuito impreso de dos o más caras, es necesario abrir agujeros que conecte eléctricamente ambas caras de la placa. Existen varias técnicas para resolver este problema, uno es utilizando pequeños tarugos de metal y soldando en sus islas, este es un método tedioso en especial cuando existe una cantidad considerable de estos agujeros. Otra menos sofisticada es conectar un cable conductor a través del agujero y soldando de la misma manera. En el proceso de metalizado básicamente consiste, por medio de electrolisis, recubrir estos agujeros con cloruro de plata, de tal manera que estos conecten a ambas caras de la placa. Figura 83.

Figura 83. Agujeros en placas multicapas



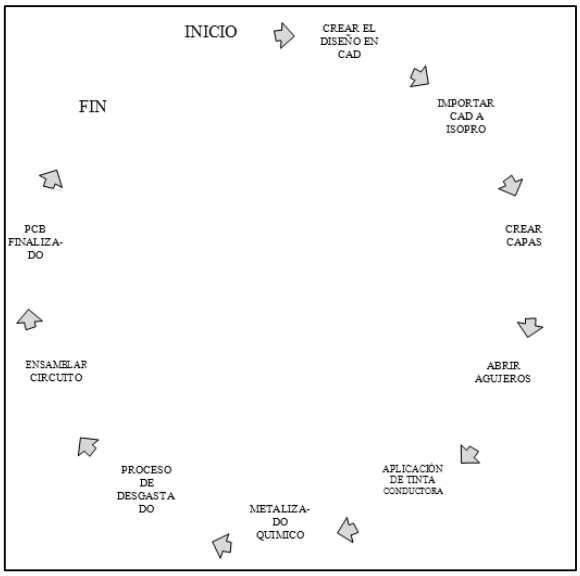
Básicamente el metalizado por procesos químicos consiste en colocar en un recipiente plástico, los componentes establecidos por la tabla XVI. Todos estos componentes deben de mezclarse según porciones establecidas, de tal manera que debe de respetarse para que el proceso sea lo más eficiente posible. Ya que los químicos estén disueltos en el tanque, debe de colocarse unas bolas de plástico vacías por dentro, esto debido que el procero de metalizado, estos químicos se cristalizan y las bolas no permiten su total cristalización.

Tabla XVI. Proporciones para metalizado

COMPONENTE	COMPONENETE ESPECIFICO	1 LITRO	23 LITROS	64 LITROS	UNIDADES
Agua destilada	Agua destilada	0.4210	9.683	26.944	Litros
Sulfato de cobre	CuSo4 líquido 32oz/gal	0.375	8.625	24.000	Litros
Acido Sulfúrico	Acido de batería al 35%	0.186	4.278	11.904	Litros
Acido Clorhídrico	Acido Clorhídrico al 35%	13.586	312.478	869.504	mL
Quick Brigthener	Quick Brigthener	5.000	115.000	320.000	mL

En el proceso de metalizado por electricidad, se necesitan de dos electrodos llamados Ánodo, que van a ser las placas de metal. El otro electrodo, llamado Cátodo es la placa del circuito impreso, la cual lleva un proceso de curación con una tinta conductiva Los ánodos se colocan en extremos opuestos del recipiente, mientras que la placa se coloca en medio de estas dos. En la figura 84, explica el procedimiento completo del desarrollo de un PCB de dos o más caras, observe que el proceso de metalizado se encuentra antes de la formación de las pistas en la placa.

Figura 84. Proceso de fabricación de PCB con metalizado



Antes que el proceso de metalizado de inicio, es necesario curar la placa con una tinta conductiva, de tal manera que los agujero que conectaran con la otra cara queden cubiertos en su totalidad. Esta tinta permite que el cobre que está en la cubeta se impregne alrededor de los agujeros, de tal manera que éstos queden rodeados de una capa conductora. El diseñador debe de tomar en cuenta que no debe de dejar rastros de tinta conductiva, en áreas donde no se desee, puesto que estas serán cubiertas con cobre.

Luego con una aspiradora, quitar el exceso de la tinta conductiva, de tal manera que todos los agujeros queden bañados, puesto que si alguno no quedó bañado, éste no será cubierto por cobre. Si algún agujero quedó con residuos de tinta, el momento de realizar el electrometalizado, éste será tapado por el cobre. Posteriormente, la placa debe ser puesto dentro de un horno tipo industrial precalentado a 100^{o}, durante 30 minutos, después de esto, la placa queda lista para el proceso de metalizado eléctrico, introduciéndolo en medio del recipiente con el electrolito preparado.

Para calcular el tiempo que debe estar la placa dentro del proceso se necesitan los siguientes datos: espesor del cobre en onzas, área de metalizado neto de ambas caras, amperaje a trabajar. Por ejemplo, se tiene una placa con media onza de grosor y el área de la placa sumergida en el electrolito es de 9 * 10.5 pulgadas, y se va a colocar una corriente de 26 A. Lo primero es encontrar el área de la placa que es: 9*10.5 * 2 = 189 pulgadas cuadradas, el dos aparece debido a que son dos caras de cobre. Se convierte a pies cuadrados el resultado anterior, dando un el resultado es 1.31 pies cuadrados. (Tabla XII).

Tabla XVII. Tiempo de metalizado

Onza	Tiempo (m) por 10 A/p^2	Tiempo (m) por 15 A/p^2	**Tiempo (m) por 20 A/p^2**	Tiempo (m) por 25 A/p^2
0.075	11	7	5	4
0.150	21	14	11	8
0.188	27	18	14	11
0.225	32	21	16	13
0.300	42	28	21	17
0.375	53	35	27	21
0.450	64	42	32	25
0.520	74	50	**37**	30
0.560	80	53	40	32
0.600	85	57	42	34
0.670	95	64	48	38
0.750	106	71	53	42
0.930	133	89	67	54
1.130	160	106	80	64
1.310	186	124	93	74
1.500	212	142	106	84

El amperaje a que se desea trabajar es a 26 amperios, esto seleccionado en el panel de control de corriente del equipo de metalizado, figura 85 y 86. En este momento hay que dividir el amperaje con el área efectiva de ambas caras, dando un resultado de 19.8 A/p^2, se busca este dato en la tabla, y se observa que el dato mayor cercano es de 20 A/p^2, por tanto se busca en esa columna el tiempo en que la placa va a estar en el proceso, guiado con la cantidad de onzas, dando así un resultado de aproximadamente 37 minutos. Hay que tomar en cuenta que el diámetro del agujero puede disminuir a la mitad luego del tratamiento, pero la cantidad de onza de cada cara siempre tiene que ser la misma, esta no varía. Luego se baña con agua para limpiar los excesos del químico, se deja secar y la placa está lista para iniciar con el proceso de ruteado.

Figura 85. Recipiente plástico de metalizado con accesorios

Figura 86. Selector de amperaje para el proceso de metalizado

4. DISEÑO DE CIRCUITO ELECTRÓNICO CON MICROCONTROLADOR, POR MEDIO DE CONTROL NUMÉRICO COMPUTARIZADO, PARA COMUNICAR LUZ PILOTO DE LAS VIDEOCAMARAS CON EL SWITCHER

En el proceso del diseño del circuito se utilizará el microcontrolador de la familia Microchip, debido a que esta casa es la más distribuida en nuestro medio. Se utilizará el equipo *CAD/CAM* para realizar el circuito, tomando en cuenta todas las recomendaciones mencionadas en otros capítulos. El proceso de diseño es el más complicado, debido a que se debe de tomar en cuenta que todos los componentes necesarios para la aplicación se encuentren a la mano y que económicamente sea factible realizar el proyecto que comprar el equipo ya construido.

4.1 Planteamiento del problema

En Guatemala, la televisión es el segundo medio de comunicación con mayor cobertura en nuestro medio, por lo que existen pequeñas empresas de televisión, tales como las que existen en el interior de la República, organizaciones no gubernamentales, universidades, aficionados, entre otros, que no poseen el capital suficiente para adquirir el equipo necesario para realizar una buena producción en vivo o pregrabado. La ventaja con una producción pregrabada es que con el software se puede corregir errores. En la mayoría de errores, el televidente no se percata de los mismos, pero el error más común e incomodo para el publico es el llamado Camarazo, que consiste en una toma movida donde no se distingue ninguna imagen.

4.1.1 Conceptos generales de la producción de televisión

La persona que está a cargo en la producción se le llama Productor, quien es el que decide el concepto general del programa, define presupuesto, sus colaboradores, los talentos, decide quien es el director, guía la dirección general de la producción, entre muchos otros cargos. Mientras tanto que el director se encarga de lo que es preproducción, o sea que coordina las actividades del *staff*, posición de las cámaras así como de los talentos, selecciona las tomas durante la producción, esto quiere decir que decide que imágenes puede ver el público, y el trabajo de postproducción, que es la edición total del programa, corregido por medio de programas especiales para la televisión.

En ocasiones se escucha de un productor asociado, que no es más que el colaborador del productor. También se encuentran los escritores, los cuales preparan un guión, que es una guía a seguir, indica que hacer en el momento y lugar adecuado. Se le llama talento a los actores, conductores y locutores que de alguna manera están en la producción. Otra persona que ayuda al director es el llamado director técnico, el cual se encuentra en la cabina de control operando el video *switcher* o consola de video. El director de iluminación es el encargado que el ambiente lumínico sea lo suficiente para que las tomas con las videocámaras sean lo mas claras posibles para el publico.

El director de set es el que diseña el ambiente visual dependiendo de las exigencias del guión. El maquillador o estilista y el vesturarista son las personas que se encargan de arreglar a los actores dependiendo de la situación de la producción. El director o técnico de audio, prepara, instala y verifica el equipo de grabación como micrófonos, verifica la calidad del audio, entre otras ocupaciones. El operador *boom* se encarga de realizar pruebas de audio pero define el tipo de micrófonos a utilizar, dependiendo del ambiente o situación del guión.

Los camarógrafos, son los encargados de manejar las videocámaras así de la calidad de imagen que reportan a cabina, estos están en contacto con el director técnico para estar pendientes de la toma que salen al aire.

En televisión existen tres fases para la producción: preproducción, producción y postproducción. La preproducción es donde se coordinan todos los detalles para colocar un escenario con el ambiente adecuado, sonido, maquillaje, video. Se realiza ensayos para prevenir los errores durante la producción. La fase de producción es el total desarrollo del evento, se puede transmitir al público ya sea en vivo o grabado, la ventaja de este último es que se puede corregir los errores que se cometieron durante la producción. Y finalmente la fase de postproducción está vinculada la parte de edición, también es donde todas las personar deben de guardar su equipo, puesto que el trabajo ya está terminado, excepto para las personas que ejecutan la edición.

El equipo básico que se requiere para la producción de televisión es: videocámaras, monitores, *switcher*, videograbadoras o algún medio para grabar la información que se está tomando, micrófonos, trípodes, iluminación, etc. Existen dos tipos de cámaras, las portátiles llamadas *ENG* y las de estudio, figuras 86 y 87. Los monitores son aparatos donde se visualiza la imagen provenientes de las videocámaras, estas por lo regular son de blanco y negro debido a que de esta manera se puede realizar un chequeo constante de la configuración de blancos obtener imágenes claras, figura 89. Las videograbadores son utilizadas para almacenar toda la grabación que se realiza, por el tipo del medio de almacenamiento puede ser clasificado por digital y analógica. Esta segunda ya no se utiliza mucho puesto que el almacenamiento digital posee la ventaja de guardar más información en un espacio reducido con una calidad similar a la analógica.

Figura 87. Cámara *ENG*

Fuente. www.aite.es

Figura 88. Cámara para estudio de televisión

Fuente. **http//recursos.cnice.mec.es**

Figura 89. Monitor

Fuente. www.escuelatrazos.es

Los micrófonos se dividen en micrófonos de manos, lavaliere, cañon, piezoeléctrico, de contacto y de estudio. Los micrófonos de mano, figura 90, como su nombre lo indica se usan en la mano del talento. El micrófono lavailiere es más conocido como micrófono de solapa, figura 91. El micrófono piezoeléctrico se utiliza para grabar sonidos transmitidos a través de superficies duras. Los micrófonos de contacto son aquellos que se encuentra directamente colocados con la fuente de sonido, como por ejemplo los utilizados en instrumentos musicales. Los micrófonos de estudio se utilizan en grabaciones dentro de un estudio de grabación.

Figura 90. Micrófono de mano

Fuente. **www.bcnaudio.com**

Figura 91. Micrófono de solapa

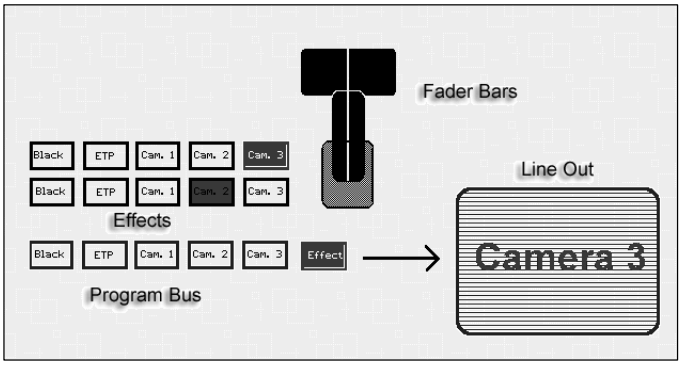
Fuente. **http://precio2.buscape.com.ar**

El *switcher* es un equipo necesario no solo para dar efecto a las imágenes, si no que nos ayudan a canalizar una imagen a la vez, puesto que en un estudio de televisión se pueden tener hasta más de 4 monitores trabajando a la vez. En la figura 92 se encuentra un mezclador de video básico, los más modernos son un poco más complejos debidos a los efector que puede proporcionar. En la figura 93 hay una ampliación de las funciones más importantes de un mezclador de video, como se muestra en la figura se encuentra un botón de negro, el cual es muy importante en la calibración de colores, y conexión para tres cámaras.

Figura 92. *Switcher* o mezclador de video

Fuente. **www.rincondelvago.com**

Figura 93. Configuración básica de un mezclador de video



Fuente. **http://www.internetcampus.com**

Cuando se desea que una cámara entre al aire, es necesario presionar un botón en la línea contraria donde otra cámara está al aire, éste comenzará a parpadear para indicar que ya está listo, mientras que en el visor del camarógrafo observa una luz también parpadeando. Cuando el técnico cambia de posición la palanca, entonces entra al aire la cámara seleccionada y su luz piloto se queda fija, mientras que la luz piloto del camarógrafo también se queda fija. Hay que observar que son tres estados de estas luces, el primer estado es la de apagado, indica que la cámara está fuera del aire, el segundo estado es cuando está parpadeando, esto indica que va a entrar al aire y el ultimo estado es la de apagado, quiere decir que está fuera del aire. Así es como los conductores saben siempre a que cámara ver cuando se realizan tomas distintas, pero la luz que les indica eso es llamada *Tally.*

4.1.2 Descripción del problema

En la producción de algún evento televisivo la comunicación es muy importante entre los miembros de la producción, esto para evitar problemas en la grabación. Un ejemplo importante es la relación entre el director técnico y sus camarógrafos, estos utilizan un radio o intercomunicador el cual está conectado por medio de un cable llamado *multipin*, este cable lleva la alimentación para las cámaras, alimentación, comunicación, luz para el tally y demás señales, esto dependiendo del tipo de cámara. En algunos canales locales no poseen el capital para comprar todo el equipo, incluso compran cámaras para aficionados, las cuales no poseen la misma calidad y funciones que una semi profesional o profesional.

Esta dificultad hace que se utilicen radios de comunicación independientes indicándoles cuando su toma será grabada. El inconveniente existente es que el director técnico está ocupado con el control de los videos que salen al aire, en la calidad de las imágenes en los monitores olvidando en algunas ocasiones comunicarse con sus camarógrafos, y en ocasiones no están en su posición por diversas causas causando grabaciones con mala calidad, si es una producción en vivo muestra la seriedad del canal y si es un programa grabado requiere más trabajo y tiempo para la edición del mismo. En la figura 94 representa un bosquejo de los posibles lugares de las cámaras en un escenario, y en la figura 95 la forma de conectas a un *switcher*.

Figura 94. Bosquejo de la producción en un escenario

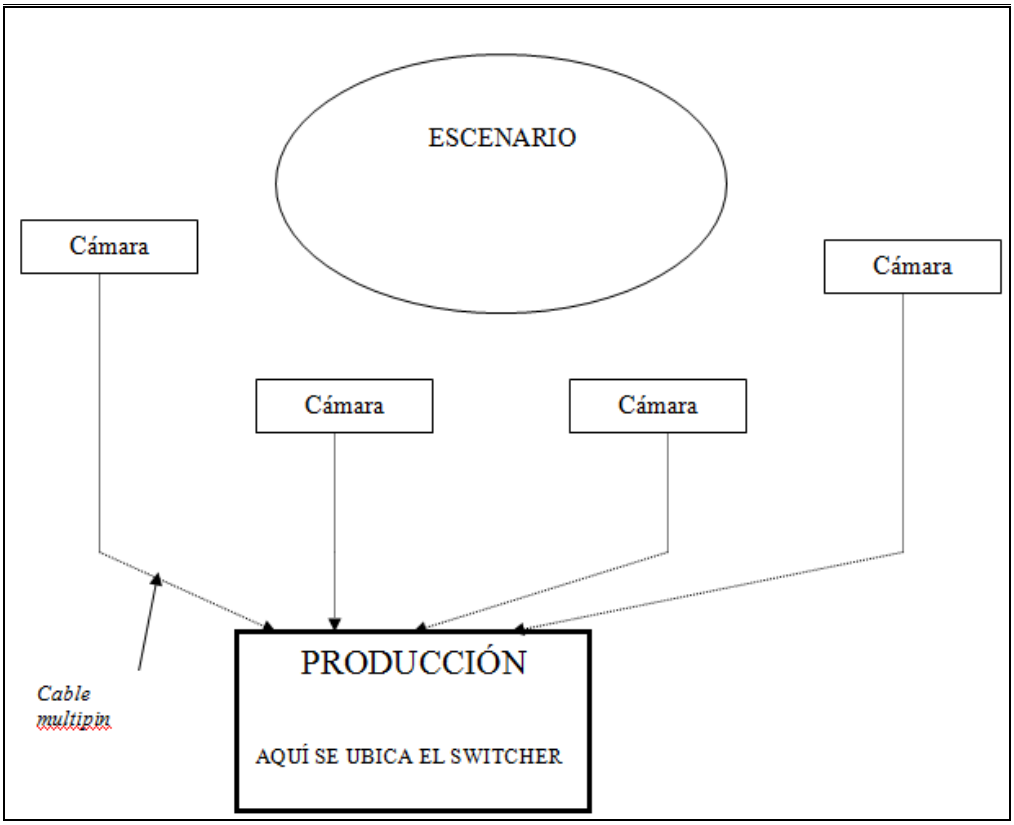
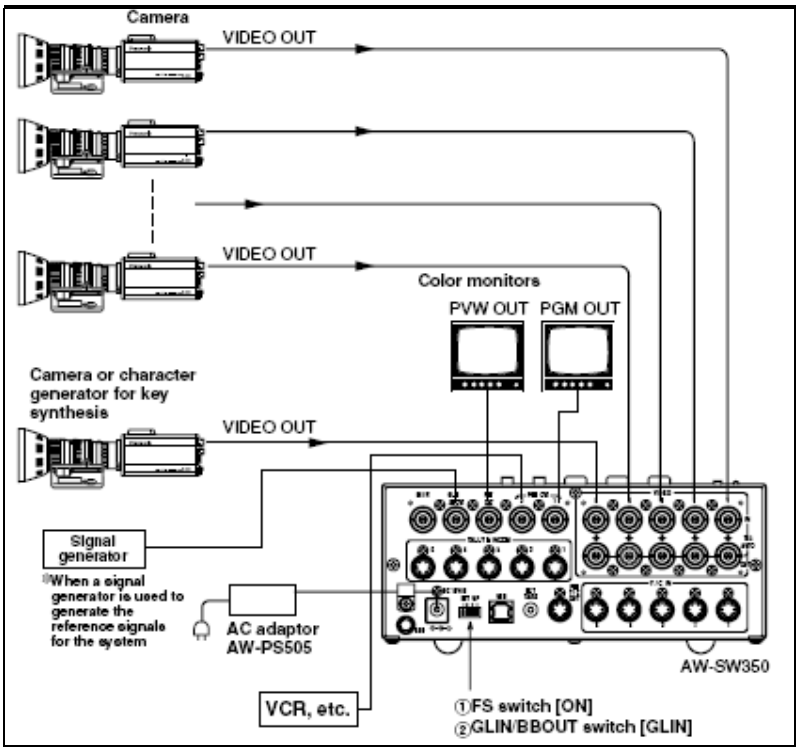


Figura 95. Conexión *switcher*-cámaras



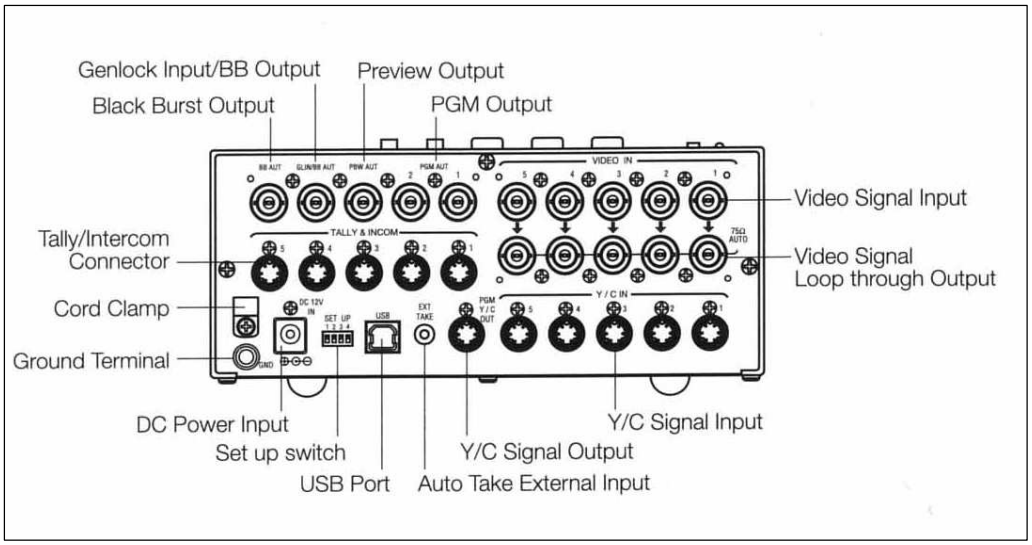
Fuente. **http://catalog2.panasonic.com**

4.1.3 Propuesta de diseño

Ya que se identifico el problema, el cual es el alto costo de cable *multipin* a comparación del precio que tiene un cable coaxial blindado que se utilizaría para transmitir el video. Entonces es necesario proponer el diseño de un circuito que se conecte al mezclador de video y a las cámaras de video, de tal manera que las señales que van al *tally* no sean guiadas por cable si no en forma inalámbrica, así tanto el camarógrafo como el talento esté siempre pendiente del estado de las cámaras que están a su alrededor.

Las conexiones del *switcher* depende de la marca y el modelo, pero las funciones básicas son siempre las mismas. Lo que cambia también dependiendo es el tipo de conector, por lo que el usuario de este circuito debe de adaptar el conector con el mismo, puesto que la tarjeta trabajará de manera estándar para cualquier tipo de *switcher*. En la figura 96 muestra la configuración de la parte anterior de un *switcher* como se puede observar la comunicación y la conexión del *tally* van juntas en el mismo conector.

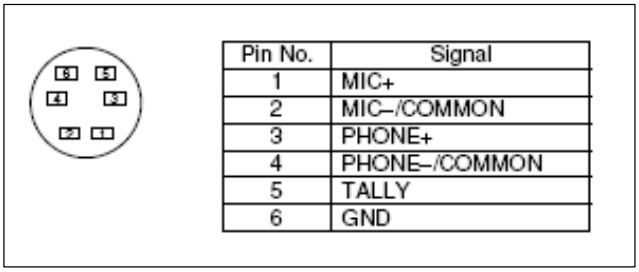
Figura 96. Parte anterior de un *switcher*



Fuente. **http://catalog2.panasonic.com**

En la figura 97 se puede observar la configuración básica de dicho conector, que es de 6 patitas. En este conector lleva la señal para el micrófono, audífonos y el *tally*, en este caso solo utilizaremos la señal del último.

Figura 97. Configuración de *pines*

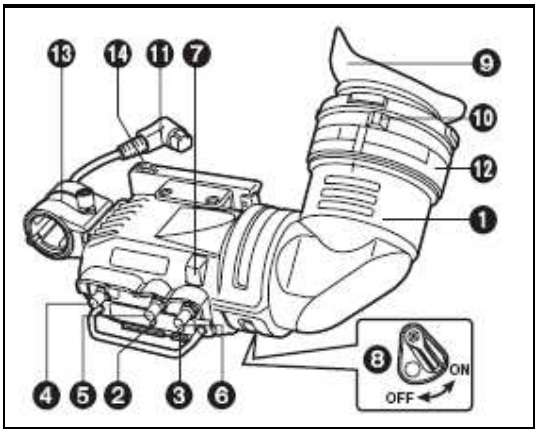


Pin No.	Signal
1	MIC+
2	MIC–/COMMON
3	PHONE+
4	PHONE–/COMMON
5	TALLY
6	GND

Fuente. http://catalog2.panasonic.com

En el caso del circuito que hay que conectar en la videocámara, es necesario localizar el *tally* así como la luz indicadora en el *viewfinder,* éste es donde el camarógrafo puede ver la imagen que está tomando, y datos como calidad de video, balance de blancos, sonido, entre otras. Dependiendo del tipo no se encuentra el *tally*, únicamente se encuentra la luz piloto del camarógrafo del estado del video o si se encuentra ambas hay que analizar que si las dos están juntas o separadas. En la figura 98 se muestra un *viewfinder* en el cual se encuentra el *tally* en el numeral 7 y en el numeral 3 se encuentra el interruptor de encendido. En el apartado 8 enciende la luz indicadora en su monitor cuando está grabando.

Figura 98. *Viewfinder*



Fuente. **http://catalog2.panasonic.com**

4.2 Elección del microcontrolador

El circuito que debe diseñarse debe de multiplexar cuatro señales codificadas y de forma inalámbrica, transmitirlo hacia las cámaras de video. Estas deben de poseer otro circuito de tal manera que reciba la señal que le corresponda, dependiendo del código que se le asigne a cada señal. El dispositivo que puede resolver el problema con estos requerimientos es un microcontrolador, debido a que existe diversidad para distintas aplicaciones. Otro punto importante es el tamaño, no se puede utilizar un microprocesador con memoria, circuitos secuenciales o combinacionales debido al espacio físico que ocupa. Otras cuestiones son también el precio, velocidad de trabajo, consumo de potencia, disponibilidad, confiabilidad, entre otros aspectos, que por la aplicación, el microcontrolador es el más apto.

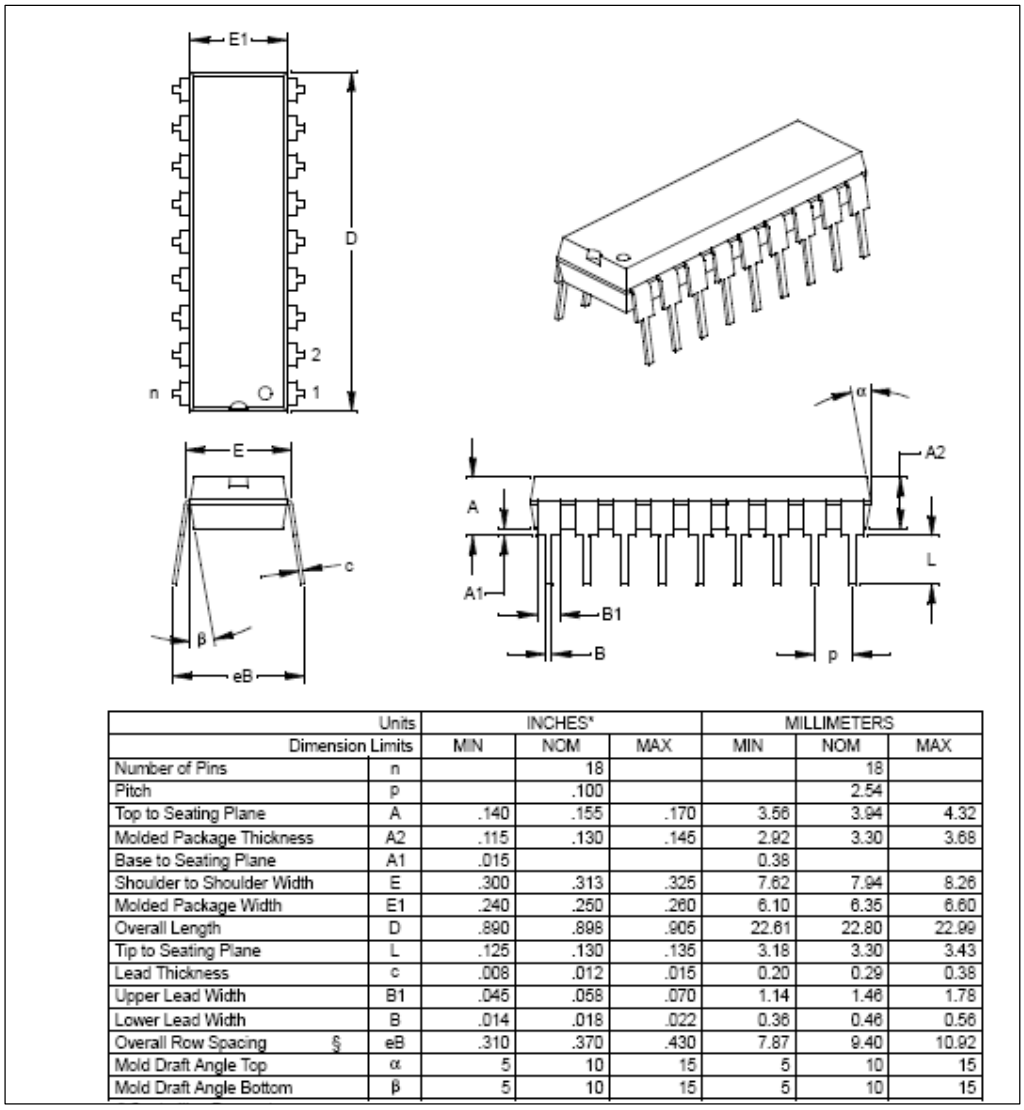
En nuestro medio el microcontrolador más común es de la familia Microchip, por lo tanto hay que analizar cuantos pines se van a utilizar ya sea en la transmisión o en la recepción de datos. En el caso de la transmisión de datos se requiere cuatro entradas y una salida, dando un total de 5 patitas del microcontrolador, esto si no se desea colocar luces indicadoras para ver el estado de las señales de entrada, si es así en total son 9 patitas. En el caso de la recepción, se necesita 1 entrada y 1 salida, pero estos circuitos deben de poseer la característica de poder asignar a cualquier cámara configurándolas desde una serie de interruptores pequeños para que reciban la señal del *switcher* que le corresponda.

La configuración se realizaría por medio de interruptores, siendo estos 4 debido a la cantidad de cámaras, y se agrega una señal de error de configuración, por si el usuario realiza algún procedimiento no adecuado. En total son 7 patitas a utilizar el microcontrolador. Como se puede analizar, son pocas las patitas que se van a utilizar del microcontrolador, los cuales existen de 8, 10, 18, 28, 10 y 40 patitas. Hay que tomar en cuenta que en el mercado nacional los *PIC* que se manejan comercialmente son el 16F84 y 16F877A, por lo que hay que elegir por alguno de los dos microcontroladores.

Antes de elegir alguno de los microcontroladores anteriores, hay que definir cual circuito integrado posee las mejores características de diseño. La transmisión de datos es un proceso importante, pues va a definir la complejidad del programa y el tamaño del circuito. El 16F877A posee comunicaciones en serie como el módulo SSP (*synchronous serial port*) y el *USART* (*universal synchronous asynchronous receiver transmitter*) y comunicaciones en paralelo *PSP* (*parallel slave port*), mientras que el 16f84 no posee ningún módulo de comunicación, por lo que es una dificultad pero no mayor.

Una desventaja de utilizar el 16F877A es el tamaño físico, pues es aproximadamente cuatro veces mayor que el 16F84, cosa que no se puede cambiar y lo que buscamos es realizar un circuito sumamente pequeño para que no estorbe cuando se utilice en la cámara y en el *switcher*. Económicamente hablando este último es más barato. En la figura 99 y 100 se muestra los dos microcontroladores mencionados como una referencia en el tamaño físico. El tamaño de memoria no interesa puesto que el programa a efectuar no será muy grande. Según los datos anteriores, el mejor microcontrolador por su precio y tamaño físico es el 16F84, puesto que se consigue fácilmente en el mercado nacional, no posee módulos de comunicación serial pero se va a solucionar realizando un programa que sustituya esta necesidad.

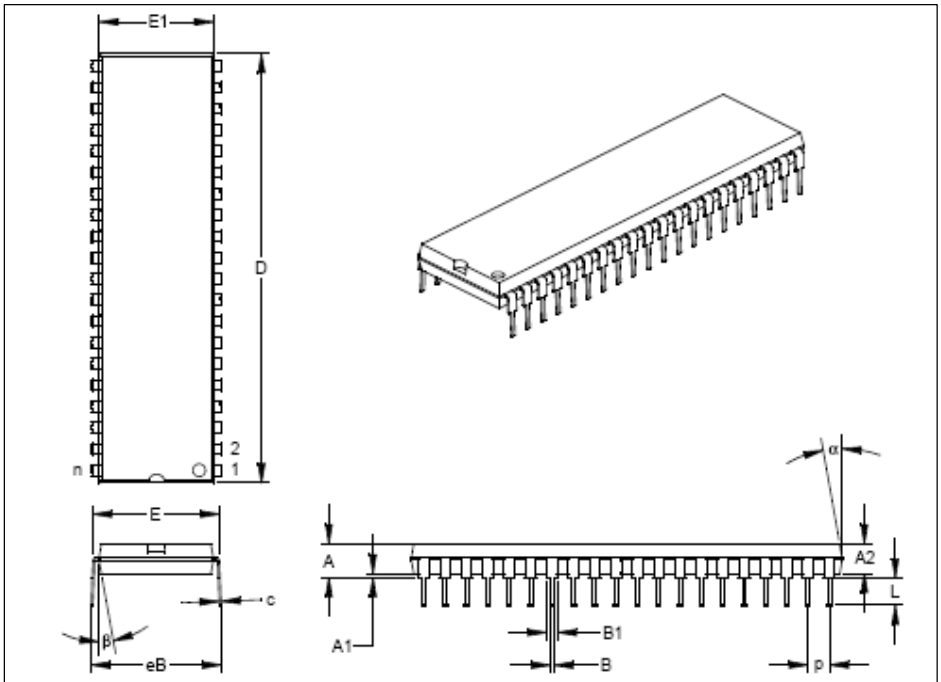
Figura 99. Dimensiones del *PIC* 16F84



Units		INCHES*			MILLIMETERS		
Dimension Limits		MIN	NOM	MAX	MIN	NOM	MAX
Number of Pins	n		18			18	
Pitch	p		.100			2.54	
Top to Seating Plane	A	.140	.155	.170	3.56	3.94	4.32
Molded Package Thickness	A2	.115	.130	.145	2.92	3.30	3.68
Base to Seating Plane	A1	.015			0.38		
Shoulder to Shoulder Width	E	.300	.313	.325	7.62	7.94	8.26
Molded Package Width	E1	.240	.250	.260	6.10	6.35	6.60
Overall Length	D	.890	.898	.905	22.61	22.80	22.99
Tip to Seating Plane	L	.125	.130	.135	3.18	3.30	3.43
Lead Thickness	c	.008	.012	.015	0.20	0.29	0.38
Upper Lead Width	B1	.045	.058	.070	1.14	1.46	1.78
Lower Lead Width	B	.014	.018	.022	0.36	0.46	0.56
Overall Row Spacing §	eB	.310	.370	.430	7.87	9.40	10.92
Mold Draft Angle Top	α	5	10	15	5	10	15
Mold Draft Angle Bottom	β	5	10	15	5	10	15

Fuente. ***Microchip. Data sheet*. Pág. 72**

Figura 100. Dimensiones del *PIC* 16F877A



Units		INCHES*			MILLIMETERS		
Dimension Limits		MIN	NOM	MAX	MIN	NOM	MAX
Number of Pins	n		40			40	
Pitch	p		.100			2.54	
Top to Seating Plane	A	.160	.175	.190	4.06	4.45	4.83
Molded Package Thickness	A2	.140	.150	.160	3.56	3.81	4.06
Base to Seating Plane	A1	.015			0.38		
Shoulder to Shoulder Width	E	.595	.600	.625	15.11	15.24	15.88
Molded Package Width	E1	.530	.545	.560	13.46	13.84	14.22
Overall Length	D	2.045	2.058	2.065	51.94	52.26	52.45
Tip to Seating Plane	L	.120	.130	.135	3.05	3.30	3.43
Lead Thickness	c	.008	.012	.015	0.20	0.29	0.38
Upper Lead Width	B1	.030	.050	.070	0.76	1.27	1.78
Lower Lead Width	B	.014	.018	.022	0.36	0.46	0.56
Overall Row Spacing §	eB	.620	.650	.680	15.75	16.51	17.27
Mold Draft Angle Top	α	5	10	15	5	10	15
Mold Draft Angle Bottom	β	5	10	15	5	10	15

Fuente. ***Microchip. Data sheet*. Pág. 211**

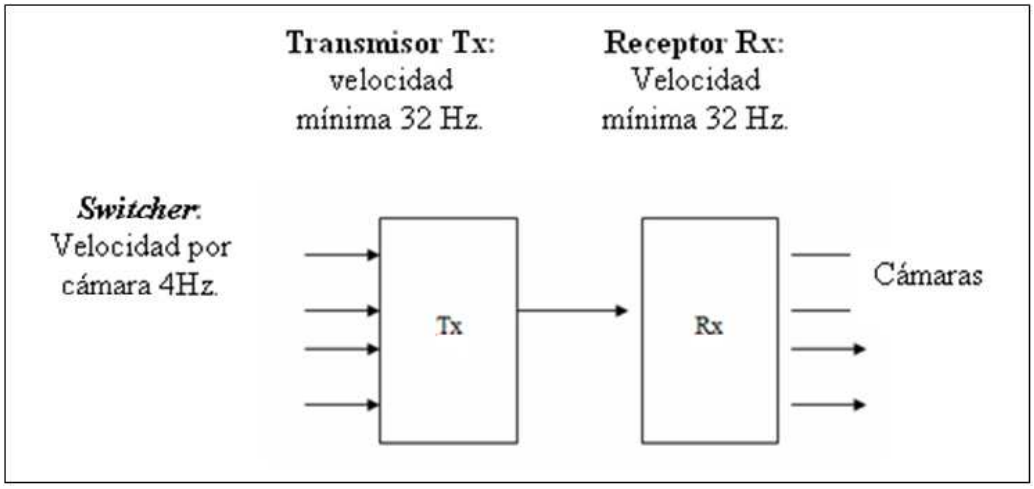
4.3 Programación y grabado del microcontrolador

En el proceso de diseño hay que tomar en cuenta los tipos de entrada y salida para el microcontrolador, puesto que estos trabajan con señales de 5 voltios, excepto los que poseen salidas a *pull-up o pull-down*. Para el diseño de programación se asumirá que las entradas y salidas van a ser lógicas *TTL* puesto que únicamente servirá de control, si existe alguna etapa de potencia se resolverá con los dispositivos adecuados en su momento.

El microcontrolador a utilizar será el 16F84, el inconveniente es que no posee módulos de protocolos de comunicación, esto quiere decir que el programa a realizar debe de dar solución a este problema. Debido a la complejidad en software en utilizar protocolos de comunicación serial tales como el *USART* y el *I2C*, es el programa, por lo que se utilizará otro tipo de comunicación. El programa primero lo que debe de realizar es multiplexar cuatro señales en paralelo en la entrada y convertirlas a una sola, enviándolo así hacia los receptores respectivos los cuales según la forma de programación recibirán la señal dependiendo de la cámara a que estén asignados.

El *tally* y la luz indicadora del camarógrafo oscilarán cuatro veces por segundo como mínimo. El objetivo es reproducir la señal del *switcher* para en las cámaras. Si utilizamos el teorema de muestreo: $fs \geq 2Nft$, donde fs es la frecuencia de muestreo, N es el número de canales y ft es la frecuencia de la señal a transmitir. Esto quiere decir que si la señal a transmitir posee una frecuencia de 4Hz y son 4 canales, la frecuencia que el microcontrolador debe de muestrear es por lo menos de 32 Hz o un periodo de 31.25 ms, estos datos son los mínimos de diseño, necesarios para reconstruir una señal similar a la original, un diagrama de bloques se muestra en la figura 101.

Figura 101. Proceso Tx-Rx



Para la programación del microcontrolador, se necesita transmitir en algún protocolo de comunicación serial, debido a que el 16F84 no posee el módulo interno, existen librerías de programa de *I2C, USART, UART*, entre otros, que realizan la comunicación serial simulado por programa. La librería a utilizar es la de *UART*, que es una comunicación serial asíncrona que comúnmente se utiliza para comunicar una computadora con un microcontrolador, pero también se puede utilizar para comunicar *PIC* a *PIC*. A continuación se describe el programa para transmitir información.

```
program transmisor UART
dim registro as byte
trisa = %00000010
porta = 0
trisb = $11111111
portb = 0
principal:
   while true
    registro = portb
     Soft_Uart_Init(PORTA, 1, 0, 1200, 0)
     Soft_Uart_Write(registro)
```

```
    goto principal
  wend
end.
```

El programa se llama transmisor *UART*, se declaran la variable a utilizar, y se asigna el puerto a como salida y el puerto B como entrada. Posteriormente los datos que recibe el puerto C, el cual esta conectado a la salida del swticher y es almacenado en la variable registro. La librería *Soft_Uart_Init(PORTX, RX, TX, B, S)*, se utiliza para inicializar el proceso de transmisión, es necesario configurar el puerto a utilizar, la patita de recepción de datos (RX), de transmisión de datos (TX), la velocidad a que se va a transmitir los datos y S es un *bit*, si es uno, manda el complemento del bus de transmisión, si es cero, estos datos son mandados tal como se almacenan en el registro respectivo.

La velocidad de transmisión es limitada por el oscilador o el cristal del microcontrolador. La librería *Soft_Uart_Write*(registro), es la que se utiliza para iniciar la transmisión de datos, como se observa el contenido de la variable registro se va a transmitir de forma serial desde un pin y a una velocidad definido por la librería anterior. Lo equivalente de este programa en lenguaje ensamblador es bastante tedioso y se necesita de mucha experiencia en programación de bajo nivel. El lenguaje ensamblador actualmente con los modernos programas de alto nivel únicamente se utiliza para realizar tareas muy específicas.

El programa de recepción se utilizó la librería *UART* receptor, la cual se muestra adelante. Primero se configura el puerto A como entradas a excepción del *bit* 1 y 2 que va a ser salida. Mientras que en el puerto B, se van a utilizar los cuatros *bits* menos significativos como entradas y el resto como salida. La librería *Soft_Uart_Init*(PORTA, 0, 1, 1200, 0), es similar igual que la utilizada para el transmisor. La librería registro = *Soft_Uart_Read*(registro), es diferente a la de transmisión, puesto que se va a recibir un byte de información y se va a almacenar en una variable designada por el diseñador.

Luego que se detectó el *bit* a trabajar, los receptores se configuraran de tal manera que si se va a instalar en la cámara uno, hay que colocar con la ayuda de un interruptor el uno en binario. Si se quiere colocar en la cámara dos, entonces se coloca dos en binario, si se coloca en la cámara tres, se coloca cuatro en binario y si se coloca en la cámara cuatro, se configura el 8 en binario. De tal manera que si no se configura ninguna de las opciones anteriores, entonces no funcionará el circuito, de manera contrario, si se activan varios interruptores a la vez, entonces se activará la de menor rango.

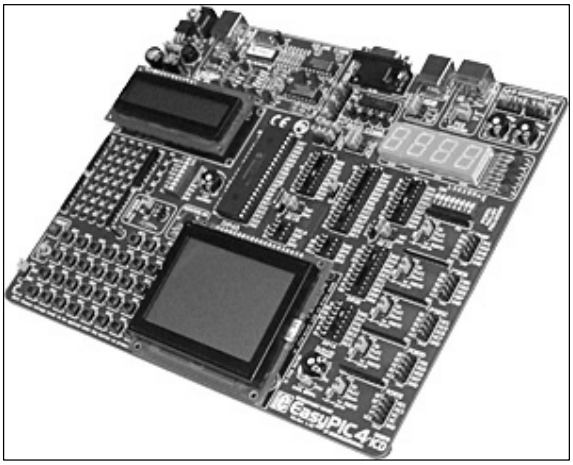
```
program receptoruart
dim registro,x as byte

inicio:
  trisa = %11111001
  porta = 0
  trisb = %00001111
  portb = 0

principal:
    while true
```

```
        Soft_Uart_Init(PORTA, 0, 1, 1200, 0)
        registro = Soft_Uart_Read(registro)
        gosub deteccion
        portb= (registro << x)
        goto principal
    wend
deteccion:
        if (portb.0 = 1) then
            x = 4
            return
        end if
        if (portb.1 = 1) then
            x = 3
            return
        end if
        if (portc.b = 1) then
            x = 2
            return
        end if
        if (portc.b = 1) then
            x = 1
            return
        end if
        goto principal
        x = 0
end.
```

4.4 Fabricación de circuito en PCB

Con los programas finalizados, es momento de programar los microcontroladores. Este proceso depende del tipo de programador que se utilice, hay unos que se la programación se realiza por el puerto serial o paralelo, los programadores mas modernos se alimentan y se programan por medio del puerto *USB*, un ejemplo se muestra en la figura 102. Hay programadores que solo son para eso mismo y otros que sirven para entrenamiento para uno o varios microcontroladores de diferentes pines y hasta distinta casa fabricante. Se recomienda los programadores entrenadores debido a que también se puede realizar pruebas antes de diseñar la placa.

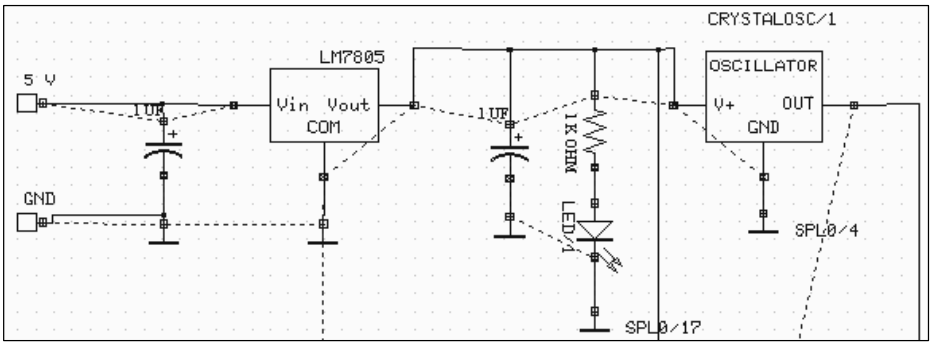
Figura 102. Programador de *PIC* moderno



Fuente. www.mikroe.com

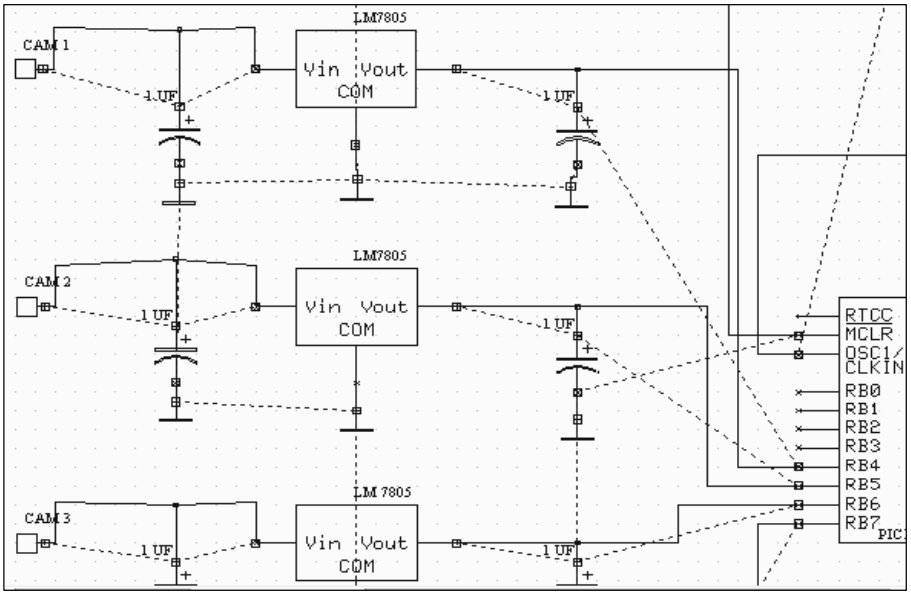
Con el programa trabajando de forma adecuada, es momento de realizar el diagrama del circuito. Hay que tomar en cuenta que el voltaje de salida del *switcher* para manejar el *tally* es de 12 Vdc, y que está cuando esté en estado astable tendrá una frecuencia mínima de 4Hz. La alimentación del circuito será de 12 voltios, el cual se reducirá a 5 V, de igual manera los voltajes de entrada provenientes del *switcher*. Existirá un led indicador de encendido, así también como uno que indica que el transmisor esté trabajando. El diagrama total se muestra en la figura 103. En la figura 102 se encuentra la parte del diagrama de la fuente de alimentación para el oscilador y el microcontrolador, la salida del oscilador va directamente al *PIC*.

Figura 103. Alimentación para oscilador y microcontrolador



En la figura se observan líneas punteadas, la cual indica las conexiones directas que existen entre los dispositivos. La entrada de la cámara 1 se conecta a RB4 del microcontrolador, la entrada de la cámara 2 se conecta a RB5, cámara 3 se conecta a RB6 y cámara 4 se conecta a RB7. La patita de reinicio se conecta a 1 para tenerla deshabilitada. Los capacitares a utilizar son de 1 µF, los reguladores son LM7805. En la figura 104 se muestra la representación esquemática de la parte de las entradas de las cámaras.

Figura 104. Entradas de las cámaras hacia el *PIC*



La salida RA0, se va a utilizar como salida serial del microcontrolador, como se observa, en la figura 105, esta patita se conecta hacia una antena y se lanza al espacio, la distancia máxima a de alcance va a depender de la antena a utilizar. La patita RA5 va a estar encendido siempre y cuando se transmitan datos, de lo contrario éste se apaga. Dichos led deben de ser de baja potencia para que el circuito no disipe mucha potencia pudiéndose utilizar también en equipos portátiles. El diagrama total se muestra en la figura 106.

Figura 105. Conexión del microcontrolador

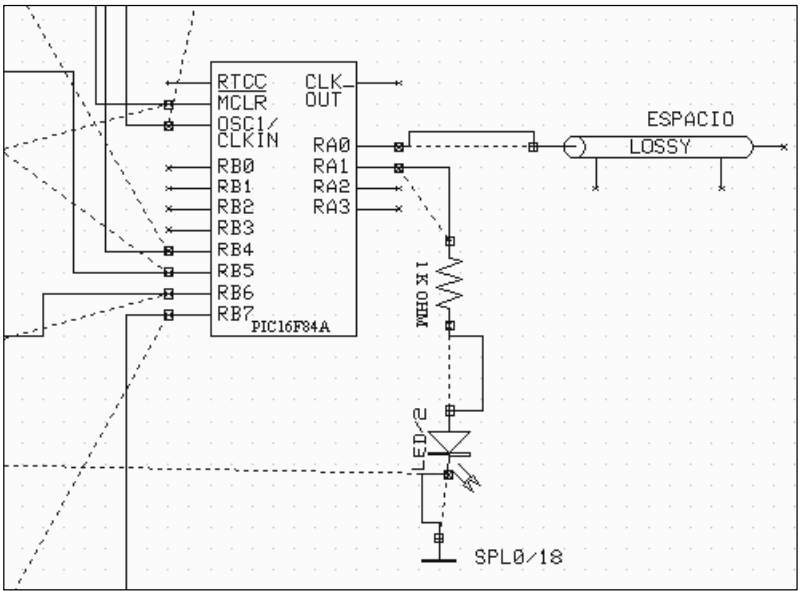
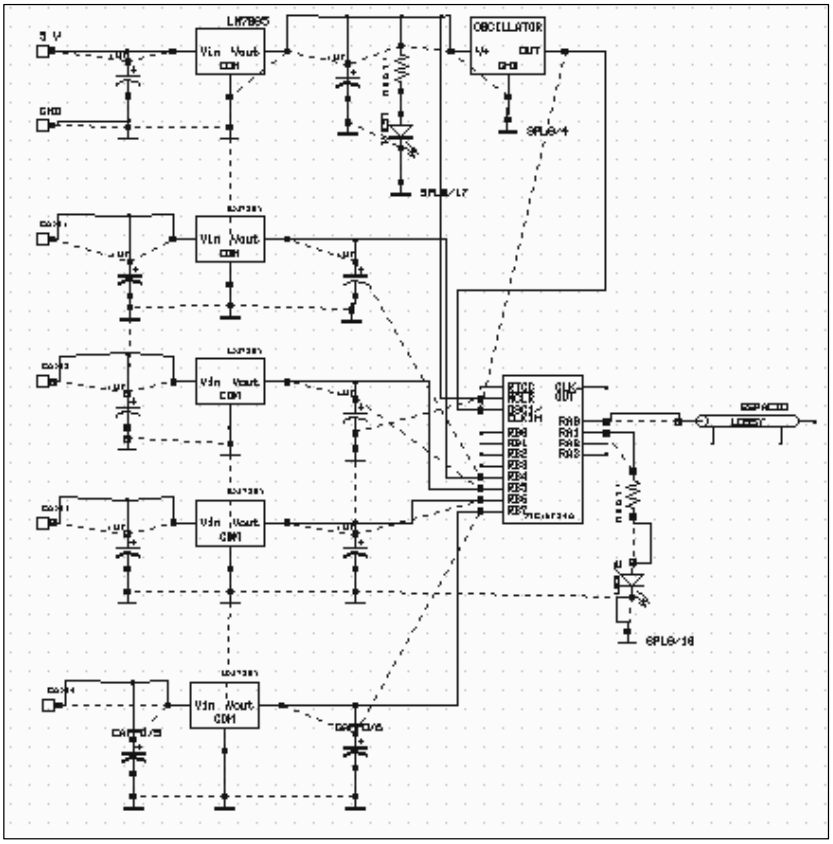


Figura 106. Diagrama total del circuito



El receptor tiene una antena que recibe la señal del *switcher*, y por medio de interruptores, se elegirá la cámara a trabajar. Si no se configura bien cada receptor, la recepción no será correcta. En la figura 107 muestra la fuente de alimentación así como el oscilador. En la figura 108 se encuentra los interruptores que se utiliza para configurar la cámara que se desea visualizar. Hay que tomar en cuenta que si alguna persona coloca en 1 los interruptores, la señal se recibirá será del menor número. Por ejemplo si se activa el interruptor 2 y 3 al mismo tiempo y el resto están desactivados, la señal recibida será del interruptor dos que es de la cámara dos. Si el usuario desea recibir la señal de la cámara 4, entonces únicamente el interruptor 4 debe estar activado, ver tabla XVIII.

Figura 107. Alimentación al oscilador y al *PIC*

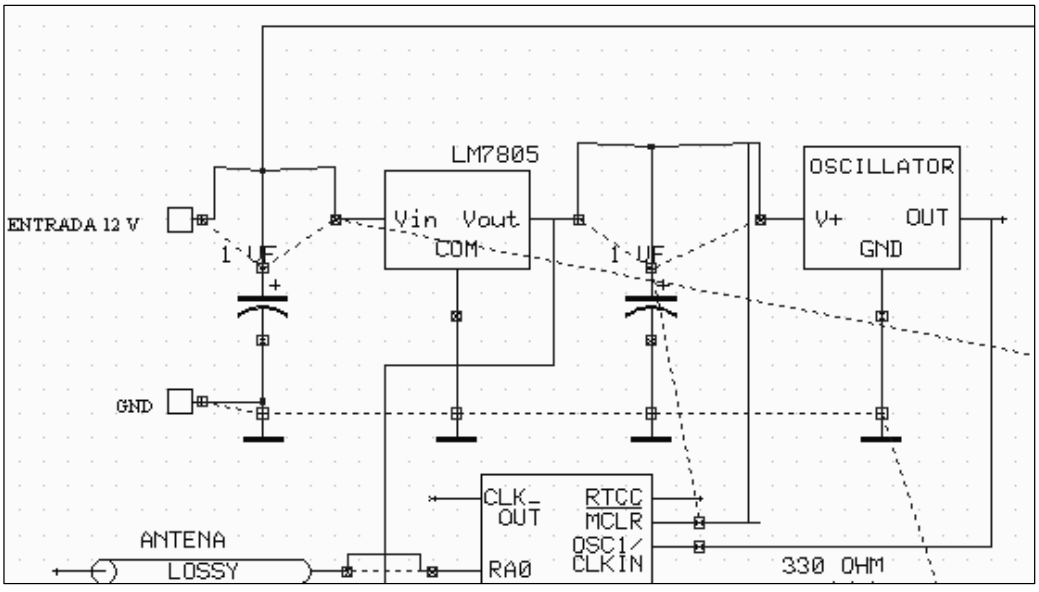


Figura 108. Configuración de interruptores

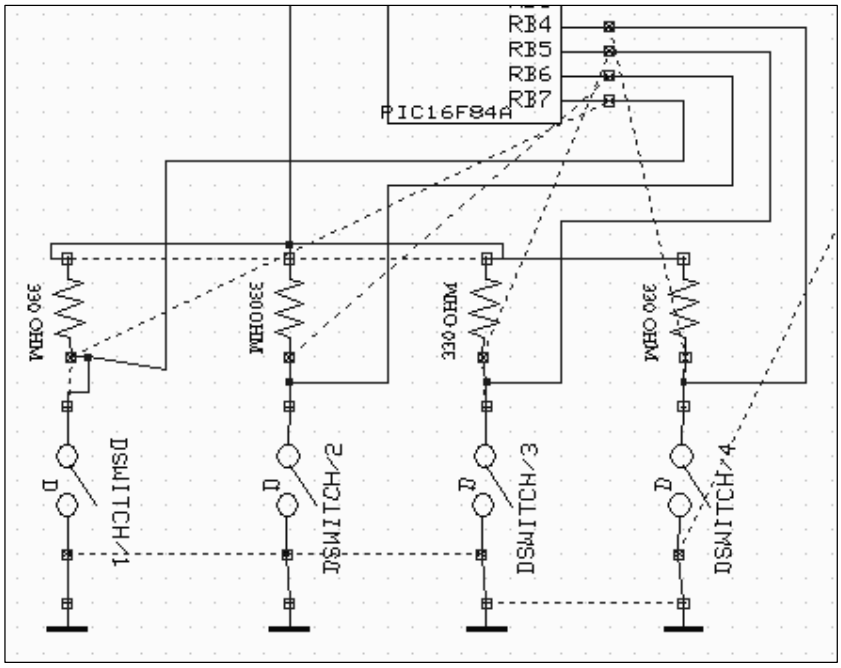


Tabla XVIII. Configuración para cámaras

SW 1	SW 2	SW 3	SW 4	CAMARA ACTIVA
0	0	0	0	NO ACTIVA
0	**0**	**0**	**1**	**4**
0	**0**	**1**	**0**	**3**
0	0	1	1	3
0	**1**	**0**	**0**	**2**
0	1	0	1	2
0	1	1	0	2
0	1	1	1	2
1	**0**	**0**	**0**	**1**
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

La señal de la antena está conectada a la patita RA0 del microcontrolador, la salida está en RB0 cuyos datos es recibido por un optoacoplador para aislar la parte de control con el de potencia. Este dispositivo óptico (figura 109) se encuentra conectado a un transistor configurado como interruptor, la señal del colector se conecta a la base de dos transistores más, uno para conectar una pequeña carga, por ejemplo un led, mientras que el otro, que es un transistor de potencia media, se puede conectar una carga de 12 voltios hasta 10 amperios, esto si lo permite la fuente de alimentación de la cámara. Este último es para conectar el *tally* o alguna otra luz indicadora (figura 110). El circuito completo se encuentra en la figura 111.

Figura 109. Salida del receptor

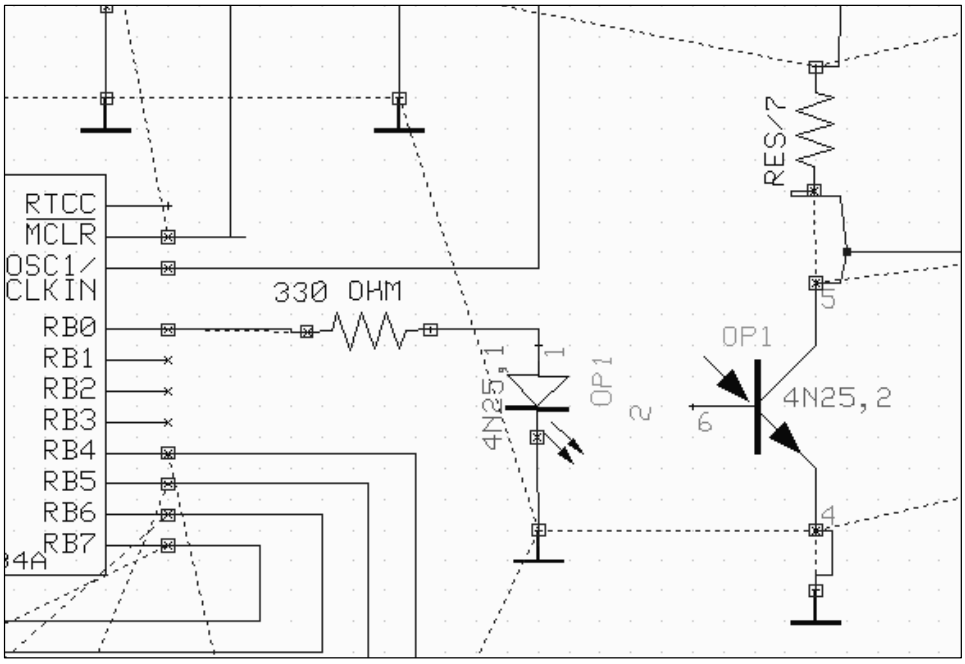


Figura 110. Salida para led y *tally*

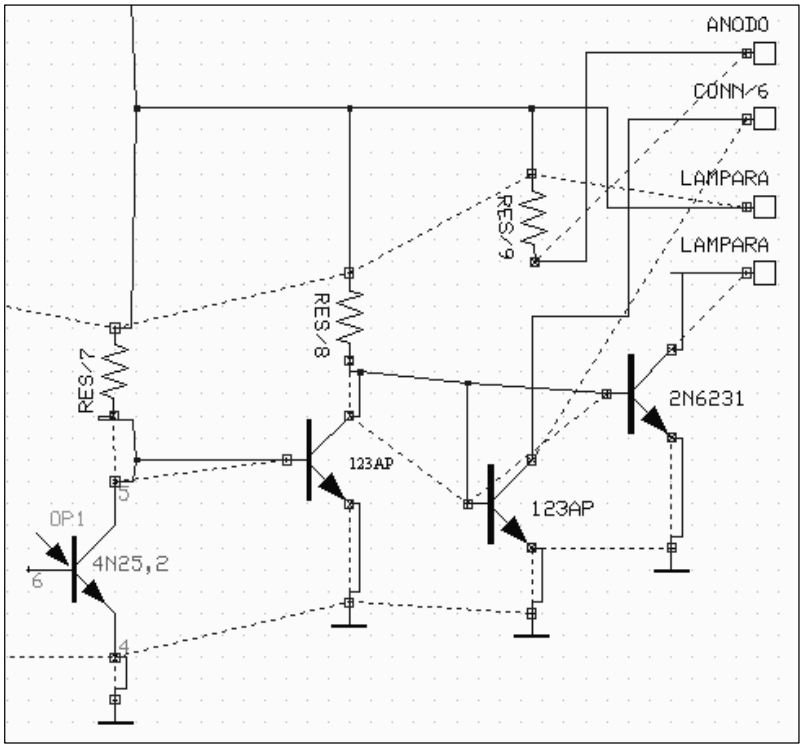
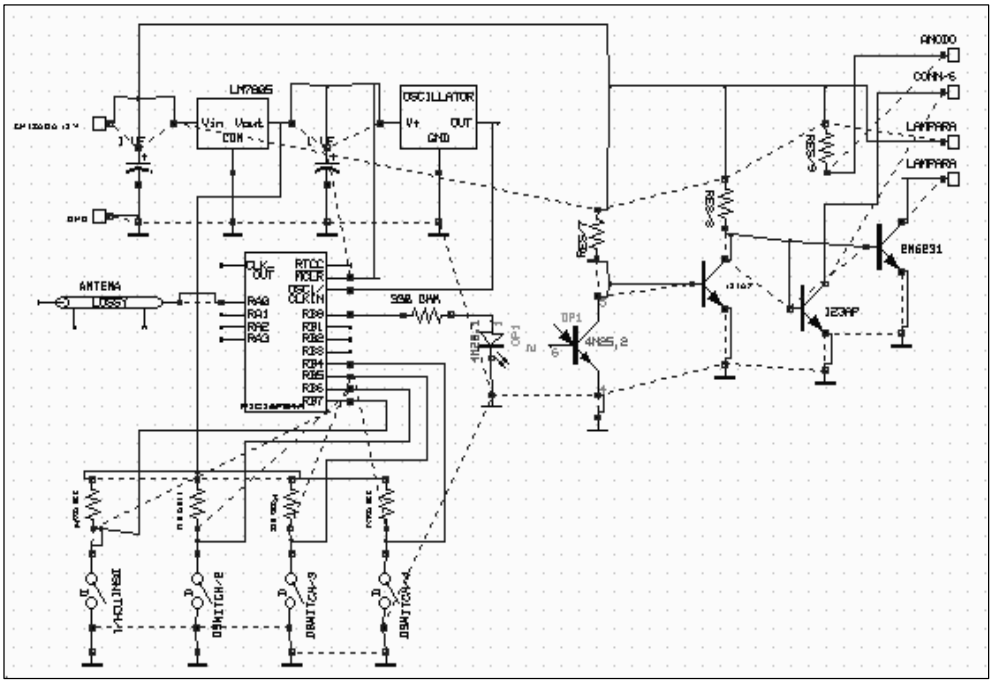


Figura 111. Circuito receptor completo



Las máscaras de las pistas del transmisor y del receptor se muestran en las figuras 112 y 113, y en las figuras 114 y 115 se muestra una visualización en tercera dimensión de la placa. Esto es posible si todos los componentes de la placa son soportados por el programa. El diseño de las pistas se realizo tomando en cuenta los criterios de diseño antes visto, para no generar algún problema ya sea en la transmisión o recepción de datos. Luego que el diseño de las pistas este finalizado, el siguiente paso es convertir estos archivos en formato G, para que la computadora del equipo *CAD/CAM*, los interprete y realice el circuito impreso.

Figura 112. Pistas del circuito transmisor

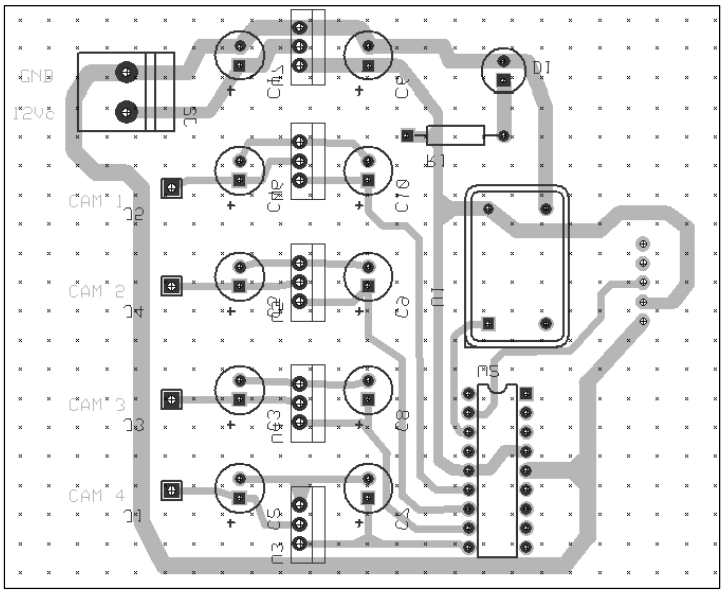


Figura 113. Pistas del circuito receptor

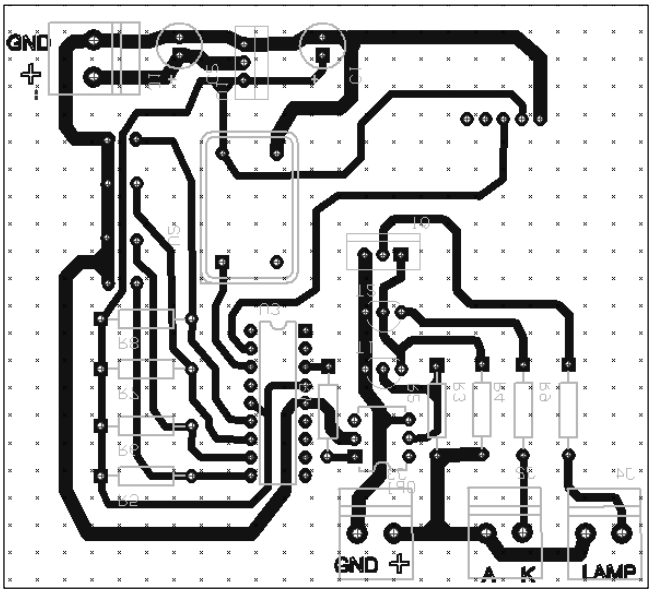


Figura 114. Vista 3D del transmisor

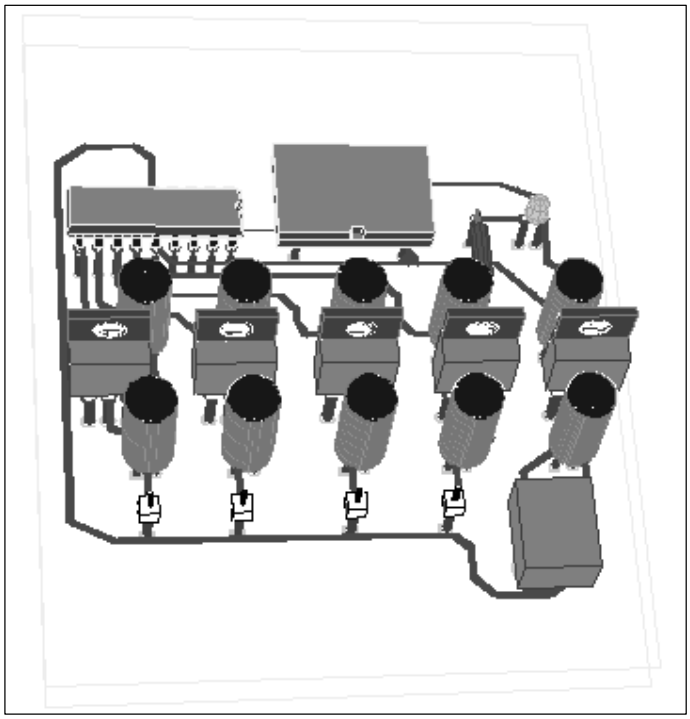
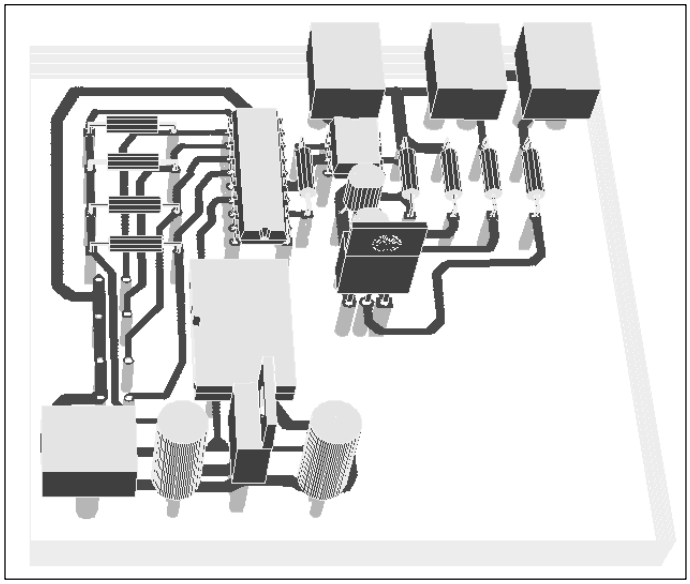


Figura 115. Vista 3D del receptor



El paso a seguir, ya teniendo los archivos G, es darle un retoque a las pistas donde quiera que sea necesario. Ya que todo el diagrama esté bien, es momento de crear las capas necesarias para desgastar y barrenar, quedando los circuitos de transmisión y recepción tal como se ven en las figuras 116 y 117. No olvide que estas figuras son la imagen de lo que se diseño en un principio. Ya que las capas son creadas el siguiente paso es desgastar el cobre para calar el circuito, posteriormente se abren agujeros, el cual es el ultimo paso, y únicamente queda cortar los circuitos justo a su tamaño, estando listos los circuitos impresos, tal como se ven en las figuras 118 y 119.

Figura 116. Circuito de transmisión con capas

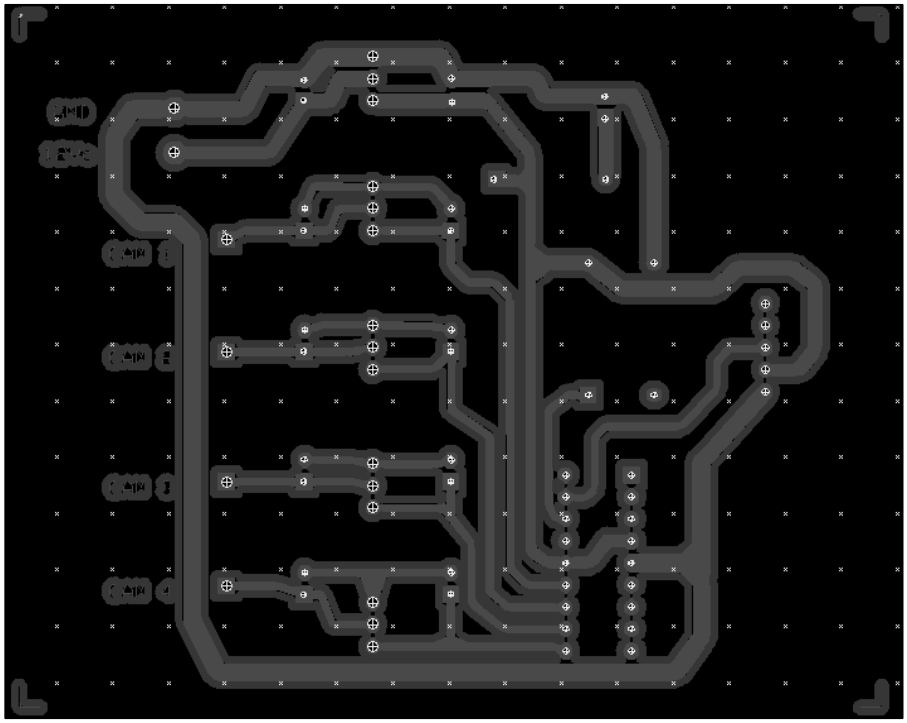


Figura 117. Circuito de recepción con capas

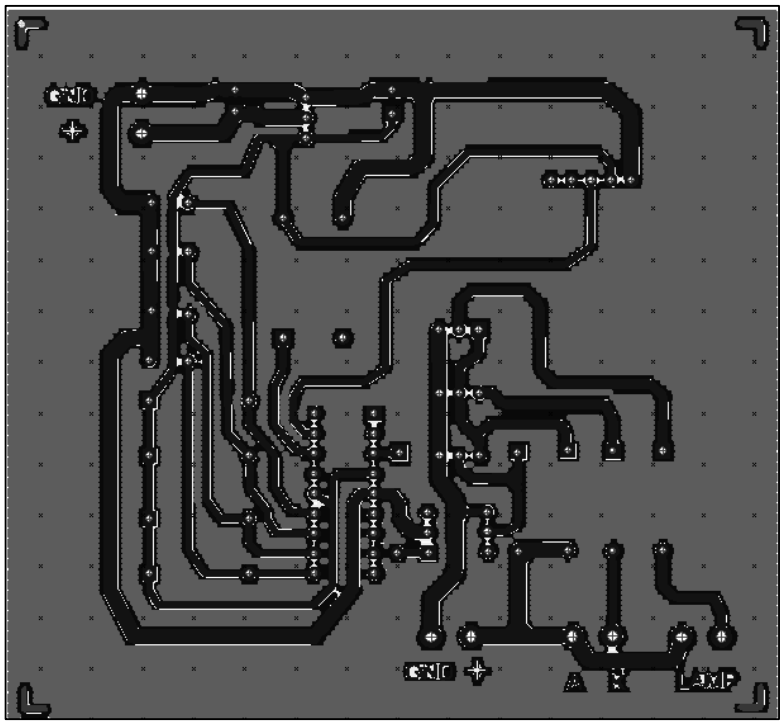


Figura 118. Circuito impreso del transmisor

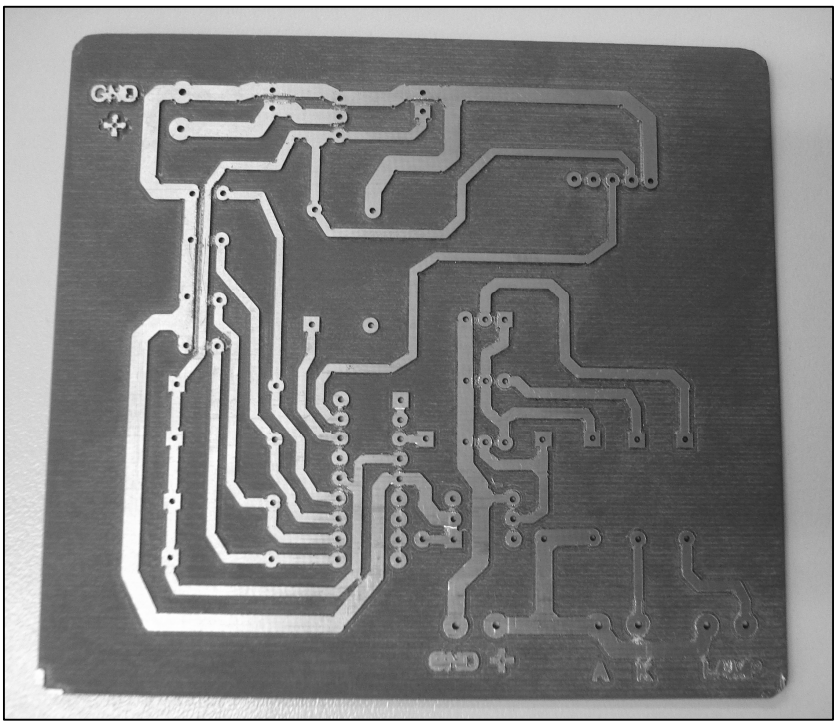
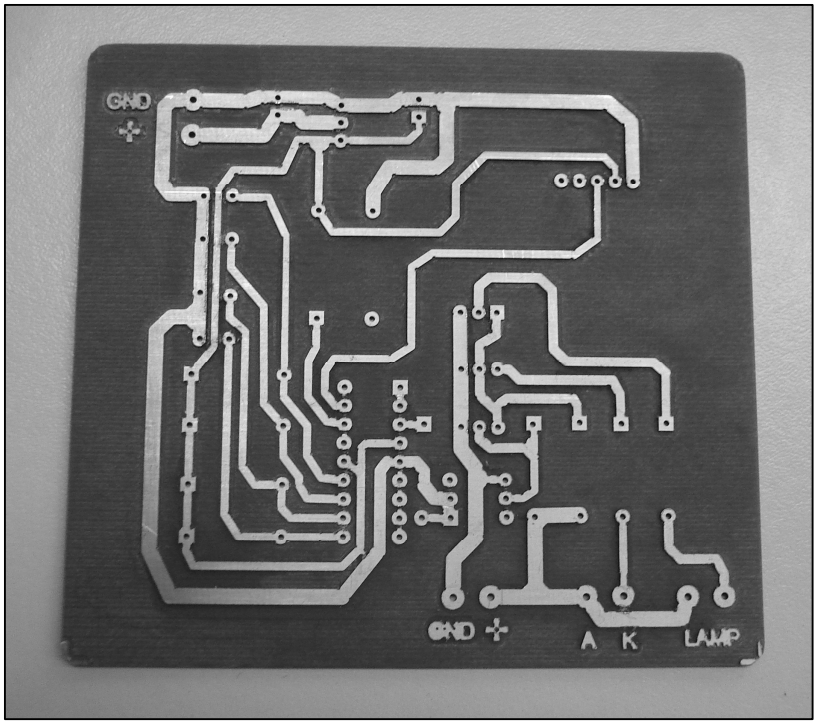


Figura 119. Circuito impreso del receptor



Las herramientas a utilizar en la fabricación del *PCB*, depende de la distancia entre pistas e islas. El tamaño más pequeño que se puede realizar entre pistas depende del tamaño de la herramienta, por ejemplo el *endmill* más pequeña para trabajar cualquier circuito, excepto para telecomunicaciones es de 0.1016 mm, pero en esta aplicación, la más pequeña a utilizar será de 0.254 mm, y la más grande será de 1 mm, esto únicamente para marcar la del circuito en la placa. La broca más pequeña a utilizar será de 0.508 mm, y la más grande de 1.55 mm.

Las recomendaciones más comunes al utilizar el equipo *CAD/CAM* son que la computadora debe poseer únicamente los programas necesarios para la aplicación, debe de estar ubicado en un cuarto con una temperatura ambiente o menor, para evitar que las piezas que sufren aumento en la temperatura durante su funcionamiento lo hagan más rápido de lo normal, nunca engrasar alguna pieza de la maquina, a menor que el fabricante lo indique. Usar equipo de seguridad tal como protectores industriales para el rostro. Tener en cuenta si la máquina es únicamente para prototipos o para producción en serie a escala, esto para regular el tiempo de trabajo y no sobrecalentar piezas delicadas.

CONCLUSIONES

1. Un microcontrolador es un circuito integrado programable que ha tenido mucha demanda en la industria, debido a su versatilidad, economía, tamaño y precio.

2. El microcontrolador es un sistema cerrado, esto quiere decir que no se puede modificar sus características físicas, como por ejemplo memoria.

3. El microprocesador posee sistema abierto, esto quiere decir que dependiendo de las aplicaciones puede añadir periféricos extras para mejorar su capacidad.

4. La ventaja principal del microcontrolador ante el microprocesador es el tamaño reducido del primero, esto hace que se coloque junto al dispositivo que va a gobernar.

5. La memoria del programa son memorias no volátiles, tales como ROM con máscara, OTP, EPROM, EEPROM y *flash*.

6. La ventaja del lenguaje ensamblador ante lenguajes de alto nivel como Visual Basic, es que el primero utiliza instrucciones muy sencillas lo que el microcontrolador las lee y ejecuta con mayor rapidez que un lenguaje de alto nivel.

7. La desventaja de utilizar lenguaje ensamblador es que los programas son grandes, tediosos y poco amigable con el usuario.

8. Existen librerías por medio de programa, en lenguajes de alto nivel, los cuales pueden sustituir módulos internos de los microcontroladores, pudiendo utilizar por ejemplo el protocolo UART, en un 16F84A si tener físicamente dicho módulo.

9. El diseño y manufactura asistida por computadora es un sistema que se utiliza actualmente en la industria de textiles, mecánica industrial y actualmente en Guatemala se utiliza para realizar circuitos impresos.

10. Las tres funciones básicas de trabajo del CNC para circuitos impresos son: desgastado y barrenado.

11. Los circuitos impresos por medio del equipo CAD/CAM, realiza las pistas por medio del desgastado con fresas de carburo de tungsteno y brocas del mismo material.

12. La vida útil de las herramientas para el CNC depende de las propiedades mecánicas del material con que se va a trabajar, por ello se recomienda trabajar con fibra de vidrio.

13. La placa cobrizada de baquelita y pértinax, que son las dos más comunes en el mercado, se recomienda para aplicaciones estudiantiles, debido a que en la industria están sometidas a ambientes hostiles muchas veces y estos materiales no están preparados para tales eventualidades.

14. El metalizado químico utiliza el principio de la electrólisis para formar una capa de cobre, cuyo fin es rellenar los agujeros en placas de dos caras y así conectar los dos lados de la placa cobrizada.

15. Al momento de realizar circuitos impresos hay que tomar en cuenta que existen máquinas que se especializan en realizar prototipos, las cuales si se exige un trabajo de larga duración, puede sufrir daños irreparables por las elevadas temperaturas que maneja.

16. Guatemala posee las personas y la herramienta necesaria diseñar circuitos electrónicos para sustituir al original, de una manera más económica y de funcionamiento similar a la original.

17. El transmisor y el receptor que se diseño se puede utilizar para transmitir cualquier información digital, cuya tasa de transmisión va a depender del reloj del microcontrolador y del ancho de banda del transmisor / receptor.

RECOMENDACIONES

1. Antes de realizar una aplicación con un microcontrolador, es indispensable que el diseñador conozca muy bien el proceso que va a controlar, con base a esto, ya se tienen los elementos necesarios para la elección del microcontrolador adecuado, tales como capacidad de memoria, puertos de comunicación, velocidad, entre otros parámetros.

2. Se recomienda que en las aplicaciones con microcontrolador, buscar algún lenguaje de alto nivel para dicha actividad, puesto que esta herramienta ayuda a realizar mejores aplicaciones, ahorrando tiempo y dinero en el desarrollo de proyectos de control.

3. Para asegurar un buen diseño de un PCB, es necesario utilizar un programa CAD adecuado para dicho propósito, siempre y cuando se respete las normas básicas de diseño de pistas y de materiales para garantizar el funcionamiento del circuito con un mínimo de interferencias electromagnéticas autoinducidas.

4. En Guatemala existen las herramientas necesarias para satisfacer ciertas deficiencias tecnológicas en la industria, por lo que se recomienda que las instituciones con equipos especiales, abran programas de divulgación, capacitación y asistencia técnica para ponerlo al servicio del mercado nacional, para que no sea necesario buscar soluciones fuera de nuestra nación.

BIBLIOGRAFIA

1. Angulo Usategui, José e Ignacio Martínez. **Microcontroladores PIC Diseño práctico de aplicaciones**. México: McGraw-Hill, 1999. 295pp.

2. Angulo Usategui, José e Ignacio Martínez. Microcontroladores PIC Diseño práctico de aplicaciones. Segunda parte: Pic 16F87X. México: McGraw-Hill, 2000. 229pp.

3. D. W. Smith. **Pic in practice**. Estados Unidos: Newnes, 2002. 250pp.

4. **Mikrobasic**. Belgrado: Mikroe, 2007. 342pp,

5. PCB Design Guidelines. Estados Unidos: s.e.2001. 14pp.

6. http://www.microchip.com, marzo 2008.

7. http://www. DonTronics.com, abril 2008.

8. http://www.lawebdeltutotial.com, abril 2008.

9. www.mikroe.com, mayo 2008.

10. www.wikipedia.com, mayo 2008.

11. www.monografias.com, junio 2008.

12. **Manual del equipo Quick Circuit 7000.**

13. **Manual del programa Edwin XP.**

14. **Manual del programa IsoPro.**