

Universidad de las Ciencias Informáticas

Facultad 2



“Desarrollo de una aplicación para la optimización de trayectorias en una mesa de coordenadas.”

**Trabajo de Diploma para optar por el título de
Ingeniero en Ciencias Informáticas**

Autores:

Jesús Borrego Rivas

Yusibel Cardo Sardiñas

Tutor:

Ing. Javier A. León Martínez

Ciudad de la Habana

Junio 2009

DECLARACIÓN DE AUTORÍA

DECLARACIÓN DE AUTORÍA

Declaramos ser los autores del presente trabajo de diploma y autorizamos a la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmo la presente a los ____ días del mes de _____ del año _____.

Yusibel Cardo Sardiñas

Jesús Borrego Rivas

Ing. Javier Alexander León Martínez

DATOS DE CONTACTO

Javier Alexander León Martínez

Graduado de Ingeniería en Automática en el Instituto Superior Politécnico José Antonio Echeverría en el año 2003.

Profesor de la asignatura Máquinas Computadoras de la Universidad de las Ciencias Informáticas desde el año 2003.

Investiga en los temas:

- Robótica
- Inteligencia Artificial aplicada al control
- Sistemas para la medición de campos magnéticos
- Desarrollo de hardware para la automatización.

Correo electrónico: jleon@uci.cu.



“El mundo camina hacia la era electrónica... Todo indica que esta ciencia se constituirá en algo así como una medida del desarrollo; quien la domine será un país de vanguardia. Vamos a volcar nuestros esfuerzos en este sentido con audacia revolucionaria”

Ernesto Che Guevara

DEDICATORIA

A mis padres Caridad Sardiñas Díaz y Daniel Cardo Valladares por los cuales me esfuerzo cada día para ser mejor, por el amor con que me han enseñado a caminar en la vida y por el apoyo que me han brindado en todo momento.

A mi hermano Danielito por quererme tanto y por querer estar siempre a mi lado, por escucharme y hacerme tanto caso.

A mi bisabuela Julia por demostrarme que todo es posible si de verdad se desea y a quien le debo parte de lo que soy.

A Yudit y Yeni por apoyarme tanto y darme la fuerza para seguir adelante en todas las situaciones difíciles, por haber depositado en mi toda su confianza.

A toda mi familia y amigos por su apoyo incondicional.

A todos aquellos que se preocupan por mi y que de una forma u otra han estado allí cuando los he necesitado.

A mi tutor Javier A. León ya que gracias a él no hubiera sido posible la realización de este trabajo y a mi compañero de tesis por todos los momentos difíciles que tuvimos que enfrentar juntos.

Yusibel

DEDICATORIA

A toda mi familia, madre, padres, hermanos y hermanas que me han orientado en todos los momentos de mi vida, enseñándome a ser mejor conmigo mismo y con todos los que me rodean, así como a todas aquellas personas que me apoyaron para poder terminar todos mis estudios.

A aquellas personas que en determinados momentos dudaron de mi, ya que gracias al hecho de haber tenido que demostrarles lo contrario me he convertido en la persona que soy hoy en día.

A todos los profesores que me han educado he indicado por los caminos correctos.

Así como también a todas mis relaciones pasadas, especialmente a Arianna que gracias a su comprensión y dedicación he logrado pasar toda mi carrera universitaria, ya que fue mi más grande apoyo en los momentos más difíciles de la carrera.

También existen personas que no pueden faltar ya que en conjunto con ellas logré desarrollar mi tesis, a mi tutor Javier A. León y especialmente a mi compañera de tesis por todas nuestras noches de insomnio y todas las dificultades que pasamos juntos.

En fin quiero a todas las personas que han influido en mi vida ya sea para bien o para mal.....

Jesús

AGRADECIMIENTOS

Agradecemos la colaboración para la realización de este trabajo al Dr. Rafael Valls Hidalgo, al MsC. Guillermo Álvarez Bestard, al Tec. Miguel Ángel Machirán Simón y demás trabajadores e investigadores del Instituto de Cibernética, Matemática y Física (ICIMAF).

A nuestro tutor Javier, a quien agradecemos por dedicarnos parte de su tiempo y por ser no solo un tutor, sino nuestro amigo y por habernos enseñado a ser cada día mejores.

A todas esas personas que contribuyeron y colaboraron de una forma u otra con la realización de este trabajo de diploma, a los cuales agradecemos por su apoyo incondicional.

Yusibel y Jesús

RESUMEN

En este trabajo se presenta el desarrollo de una aplicación que permite mejorar el rendimiento de una mesa de coordenadas destinada a la fabricación de circuitos impresos que forman parte de prototipos de dispositivos diseñados con fines de investigación y desarrollo, a partir de la optimización de ficheros Gerber. Para ello se decidió emplear el algoritmo del problema del viajante de comercio, mediante la solución del vecino más cercano (Nearest Neighbor). El sistema fue modelado empleando la metodología XP, y fue implementado utilizando el lenguaje de programación C#.

Palabras Claves: Fichero estándar Gerber; algoritmo; circuito impreso; cabezal; mesa de coordenadas; heurística.

ÍNDICE

DATOS DE CONTACTO.....	III
DEDICATORIA	VI
AGRADECIMIENTOS	VII
RESUMEN.....	VIII
ÍNDICE	IX
ÍNDICE DE TABLAS	XIV
ÍNDICE DE FIGURAS.....	XVII
INTRODUCCIÓN.....	1
Capítulo 1: Fundamentación Teórica	6
1.1 Introducción	6
1.2 Conceptos fundamentales para entender la creación de los circuitos impresos.....	6
1.2.1 ¿Qué son las mesas de coordenadas o plotters?	6
1.2.1.1 Elementos que componen la mesa de coordenadas	6
1.2.2 ¿Qué es Printed Circuit Board (PCB) o circuito impreso?	7
1.2.2.1 Tipos de circuitos impresos.....	7
1.2.3 ¿Qué son los ficheros de estándar Gerber?	8
1.3 Tendencias actuales	9
1.4 Estudio de los algoritmos para encontrar caminos mínimos entre dos puntos.	10
1.4.1 Algoritmo de Dijkstra.....	10
1.4.2 Algoritmo de Floyd – Warshall	10

1.4.3	Problema del Viajante de Comercio.....	11
1.4.3.1	Métodos Branch and Bound.....	11
1.4.3.2	Métodos de Branch and Cut.....	12
1.4.3.3	Teniendo en cuenta el número de secuencia en que la ciudad es visitada.....	12
1.4.3.4	Programación dinámica.....	12
1.5	Ejemplo de utilización del algoritmo del viajante para el perforado de placas.....	12
1.6	Metodología de desarrollo.....	13
1.6.1	¿Por qué XP?.....	14
1.7	Herramientas y tecnologías.....	15
1.7.1	Lenguaje de programación.....	15
1.7.2	Plataforma de desarrollo.....	15
1.8	Conclusiones.....	16
Capítulo 2: Características del Sistema.....		17
2.1	Introducción.....	17
2.2	Flujo de los procesos involucrados en el campo de acción.....	17
2.3	Análisis crítico del funcionamiento del sistema actual.....	17
2.4	Objeto de automatización.....	18
2.5	Características de la propuesta del sistema.....	18
2.5.1	Algoritmo para dar solución a la problemática planteada.....	19
2.6	Requerimientos.....	20
2.6.1	Requerimientos funcionales.....	21
2.6.2	Requerimientos no funcionales.....	21
2.7	Conclusiones.....	22

Capítulo 3: Exploración y Planificación.	23
3.1 Introducción	23
3.2 Fase de exploración.....	23
3.2.1 Historias de usuario	23
3.3 Planificación.....	26
3.3.1 Estimación de esfuerzo por Historias de usuario.....	26
3.3.2 Plan de iteraciones.....	26
3.3.3 Iteración 1	27
3.3.4 Iteración 2	27
3.3.5 Iteración 3	27
3.3.6 Plan de duración de iteraciones	27
3.3.7 Plan de entregas del sistema	28
3.4 Conclusiones	28
Capítulo 4: Implementación y Pruebas.....	29
4.1 Introducción.	29
4.2 Diseño del sistema.....	29
4.3 Iteración 1	31
4.3.1 Tareas de las Historias de usuario abordadas en la iteración.....	31
4.4 Iteración 2.....	34
4.5 Iteración 3.....	35
4.6 Pruebas	36
4.7 Pruebas de Aceptación.....	37
4.8 Pruebas de Unidad	41

4.9	Conclusiones	43
Capítulo 5: Estudio de Factibilidad		44
5.1	Introducción	44
5.2	COCOMO II	44
5.3	Características del sistema.....	44
5.3.1	Entradas externas	45
5.3.2	Salidas externas	45
5.3.3	Consultas externas	45
5.3.4	Archivos lógicos internos.....	46
5.3.5	Archivos de interfaz externos	46
5.3.6	Puntos de función sin ajustar	47
5.4	Cálculo de instrucciones fuentes, esfuerzo, tiempo de desarrollo, cantidad de hombres y costo .	47
5.4.1	Cálculo del esfuerzo nominal	48
5.4.1.2	Ajuste del esfuerzo nominal	49
5.4.2	Cálculo del tiempo de desarrollo del software	50
5.4.3	Cálculo del costo total del proyecto	51
5.5	Beneficios tangibles e intangibles	52
5.6	Análisis del costo	52
5.7	Conclusiones	52
CONCLUSIONES GENERALES.....		53
RECOMENDACIONES		54
BIBLIOGRAFÍA.....		55
REFERENCIAS BIBLIOGRÁFICAS		58

ÍNDICE

GLOSARIO DE TÉRMINOS.....	59
ANEXOS.....	60
Anexo 1 Mesa de coordenadas.....	60
Anexo 2 Circuito impreso.....	62
Anexo 3. Manual Gerber.....	63

ÍNDICE DE TABLAS

Tabla 1- Historia de usuario Cargar fichero estándar Gerber.	23
Tabla 2- Historia de usuario Optimizar fichero barreno.	24
Tabla 3- Historia de usuario Optimizar fichero fresado.....	24
Tabla 4- Historia de usuario Optimizar fichero líneas.	25
Tabla 5- Historia de usuario Mostrar fichero optimizado.....	25
Tabla 6- Historia de usuario Salvar fichero.....	25
Tabla 7- Estimación de esfuerzo por Historia de usuario.	26
Tabla 8- Plan de duración de las iteraciones.....	27
Tabla 9- Plan de Entregas.	28
Tabla 10- Tarjeta CRC Clase Barrenados.....	30
Tabla 11- Tarjeta CRC Clase Líneas.	30
Tabla 12- Tarjeta CRC Clase Control_Gerber.....	30
Tabla 13- Tarjeta CRC Clase Optimizador_Líneas.	30
Tabla 14- Historias de usuario abordadas en la primera iteración.	31
Tabla 15- Tarea # 1 de la Historia de usuario Cargar fichero estándar Gerber.	31
Tabla 16- Tarea # 2 de la Historia de usuario Cargar fichero estándar Gerber.	32
Tabla 17- Tarea # 3 de la Historia de usuario Cargar fichero estándar Gerber.	32
Tabla 18- Tarea # 4 de la Historia de usuario Cargar fichero estándar Gerber.	32
Tabla 19- Tarea # 1 de la Historia de usuario Optimizar fichero barreno.....	33
Tabla 20- Tarea # 2 de la Historia de usuario Optimizar fichero barreno.....	33

ÍNDICE DE TABLAS

Tabla 21- Tarea # 3 de la Historia de usuario Optimizar fichero barreno.....	33
Tabla 22- Historias de usuario abordadas en la segunda iteración.	34
Tabla 23- Tarea # 1 de la Historia de usuario Optimizar fichero fresado.	34
Tabla 24- Tarea # 1 de la Historia de usuario Optimizar fichero líneas.	35
Tabla 25- Historias de usuario abordadas en la tercera iteración.	35
Tabla 26- Tarea # 1 de la Historia de usuario Mostrar fichero optimizado.	35
Tabla 27- Tarea # 2 de la Historia de usuario Mostrar fichero optimizado.	36
Tabla 28- Tarea # 1 de la Historia de usuario Salvar fichero estándar Gerber.	36
Tabla 29- Pruebas de aceptación para la Historia de usuario Cargar fichero estándar Gerber.	37
Tabla 30- Pruebas de aceptación para la Historia de usuario Cargar fichero estándar Gerber.	37
Tabla 31- Pruebas de aceptación para la Historia de usuario Mostrar fichero optimizado.	38
Tabla 32- Pruebas de aceptación para la Historia de usuario Mostrar fichero optimizado.	39
Tabla 33- Pruebas de aceptación para la Historia de usuario Salvar fichero estándar Gerber.	39
Tabla 34- Pruebas de unidad a la Historia de usuario Cargar fichero estándar Gerber.	41
Tabla 35- Pruebas de unidad a la Historia de usuario Optimizar fichero barreno.	41
Tabla 36- Pruebas de unidad a la Historia de usuario Optimizar fichero líneas.	42
Tabla 37- Pruebas de unidad a la Historia de usuario Mostrar fichero optimizado.	42
Tabla 38- Pruebas de unidad a la Historia de usuario Salvar fichero estándar Gerber.	43
Tabla 39- Entradas externas.	45
Tabla 40- Salidas externas.	45
Tabla 41- Consultas externas.	46
Tabla 42- Archivos lógicos internos.	46
Tabla 43- Archivos de interfaz externos.	47

ÍNDICE DE TABLAS

Tabla 44- Puntos de función sin ajustar.	47
Tabla 45- Características generales del proyecto.	48
Tabla 46- Factores de escala.....	49
Tabla 47- Multiplicadores de esfuerzo.....	49
Tabla 48- Resultados.....	51

ÍNDICE DE FIGURAS

Figura 1 Mesa de coordenadas.....	60
Figura 2 Mesa de coordenadas.....	60
Figura 3 Mesa de coordenadas.....	60
Figura 4 Herramienta para el barrenado o fresado.....	61
Figura 5 Cabezal portaplumas	61
Figura 6 Proceso de pintar el circuito impreso antes y después de optimizado	61
Figura 7 Proceso de barrenar el circuito impreso antes y después de optimizado	62
Figura 8 Circuito impreso	62

INTRODUCCIÓN

Los grandes avances en la rama de la electrónica, la informática y la automática han llevado consigo la creación de equipos electrónicos cada vez más complejos. Estos a su vez requieren de un manejo que va más allá del simple hecho de apretar un botón o ejecutar una sentencia determinada, convirtiéndose en una necesidad predominante el uso de los algoritmos, herramientas de inestimable ayuda, para llevar a cabo el difícil manejo de estos artefactos.

El origen del término "algoritmo" se remonta al siglo IX y se le atribuye su invención al matemático árabe Abu Ja'far Muhammad ibn Musa al-Khwarizmi, considerado el primer pensador algorítmico, quien explicó que, mediante una especificación clara y concisa de cómo calcular sistemáticamente, se podrían definir algoritmos que fueran usados en dispositivos mecánicos en vez de las manos.

Un algoritmo es un conjunto finito de pasos definidos, estructurados en el tiempo y formulados con base a un conjunto finito de reglas no ambiguas, que proveen un procedimiento para dar la solución o indicar la falta de esta a un problema en un tiempo determinado [1].

Los algoritmos se pueden agrupar según las técnicas para su diseño, los considerados voraces (Greedy) que seleccionan los elementos más prometedores de un conjunto de candidatos hasta encontrar una solución, los paralelos que permiten la división de un problema en subproblemas de forma que se puedan ejecutar de forma simultánea en varios procesadores, los algoritmos de programación dinámica que son esenciales para resolver los problemas de costo computacional, los denominados divide y vencerás donde dividen un problema complejo en subconjuntos disjuntos¹ obteniendo una solución de cada uno de ellos para después unirlos, logrando así la solución al problema completo, entre otros.

Se pueden resaltar una gama de innumerables aplicaciones entre las que se encuentran hallar el trayecto más corto entre dos lugares determinados, alcanzar aproximaciones bien específicas del mundo real, así como dirigir el posicionamiento de un telescopio o en la rama de la medicina para brindar un diagnóstico médico. Los problemas de optimización juegan un papel fundamental entre sus aplicaciones, un ejemplo fehaciente es para llevar a cabo la creación de los circuitos impresos, en cuanto al proceso de

INTRODUCCIÓN

barrenar, pintar y fresar², donde se tiene que tener en cuenta además, gran precisión y exactitud en cada una de las mediciones.

En los últimos años, la mayoría de los fabricantes de dispositivos electrónicos han asumido para almacenar la información sobre circuitos impresos, el formato Gerber. El fichero estándar Gerber presenta todas las instrucciones para dirigir el funcionamiento de la maquinaria que lleva a cabo el proceso de creación de los mismos.

En nuestro país se encuentra el Instituto de Cibernética, Matemática y Física (ICIMAF) encargado de desarrollar investigaciones de carácter teórico y aplicado en las áreas de cibernética, matemática y física como institución perteneciente al Ministerio de Ciencias, Tecnología y Medio Ambiente. En el mismo existe desde hace más de veinte años una mesa de coordenadas de fabricación húngara que fue donada a nuestro país como copiadora de planos. Con tal de adaptar esta útil y costosa herramienta a los nuevos estándares y requerimientos tecnológicos su interfaz de hardware ha sido objeto de sucesivos procesos de modernización durante estos años, desarrollando además una nueva interfaz para manejar la misma por el puerto USB. Luego de diversas transformaciones, esta máquina devino en un mecanismo único y su uso pasó a estar destinado principalmente a la creación de circuitos impresos, los cuales forman parte de prototipos de dispositivos diseñados fundamentalmente con fines de investigación y desarrollo.

Como consecuencia de que la mesa de coordenadas es muy antigua y teniendo en cuenta las características que técnicas que la distinguen, el proceso de creación de los circuitos impresos, no es eficiente por lo que no alcanza suficiente rapidez y ante un evento inesperado, como puede ser una falla en el suministro eléctrico, provoca que se pierdan los recursos empleados en el mismo, como son tiempo y materias primas, pues una vez que se comienza con la fabricación de la placa, el proceso no se puede interrumpir.

En base a lo antes expuesto se plantea el siguiente **problema científico**: ¿Cómo garantizar el funcionamiento del cabezal de la mesa minimizando el costo recursos?

Teniendo como **objeto de estudio**: Procesos para la optimización de trayectorias en el fichero estándar Gerber siendo el **campo de acción**: Los procesos automatizados para la optimización de

trayectorias en el fichero estándar Gerber empleado en una mesa de coordenadas en el Instituto de Cibernética, Matemática y Física (ICIMAF).

El **objetivo general** de este trabajo consiste en desarrollar una aplicación capaz de garantizar el funcionamiento óptimo del cabezal de una mesa de coordenadas, minimizando el costo de recursos.

Para lograr el cumplimiento de este objetivo se plantean los siguientes **objetivos específicos**:

- ❖ Seleccionar el algoritmo para la optimización de trayectorias, que permita minimizar el costo de recursos para cada una de las funcionalidades de la mesa de coordenadas.
- ❖ Validar el algoritmo para la optimización de trayectorias seleccionado.
- ❖ Desarrollo de la aplicación.

Preguntas Científicas

¿Cómo está conformado el proceso de creación de los circuitos impresos?

¿Cuáles son los algoritmos que se pueden utilizar para la optimización del recorrido del cabezal de la mesa, teniendo en cuenta minimizar el costo de recursos?

¿Cómo quedará conformada la aplicación para la garantizar el funcionamiento del cabezal de la mesa teniendo en cuenta minimizar el costo de recursos?

Para cumplir con los objetivos anteriormente planteados, se trazan las siguientes **tareas de la investigación**:

1. Realizar entrevistas a los miembros del equipo de desarrollo que estuvo a cargo de la realización de la versión anterior del sistema.
2. Buscar y analizar la documentación referente a mesas de coordenadas.
3. Efectuar un estudio comparativo de métodos matemáticos empleados para la optimización de trayectorias.

4. Analizar las especificaciones del estándar Gerber para ficheros de diseño de circuitos impresos.

Para el desarrollo de las tareas de la investigación del trabajo de diploma se utilizaron métodos teóricos y empíricos.

- Analítico-Sintético. Se utiliza este método centrándose en el procesamiento de la información, permitiendo la selección de los elementos más importantes, de manera que se elabore correctamente la información.
- Inductivo-Deductivo. Se utiliza este método con el objetivo de llegar a conocimientos generalizadores que permitan determinar qué algoritmo de optimización es el indicado llevando a cabo un análisis en particular de los más representativos.
- Análisis Histórico-Lógico. Este método se utiliza con el objetivo de constatar teóricamente la evolución de los algoritmos y problemas matemáticos de optimización para dar cumplimiento al objetivo propuesto.
- Métodos Empíricos (Matemáticos y Medición): Este método se basa fundamentalmente en la observación, cálculos, mediciones y en la investigación relacionada con este trabajo de diploma, además en la selección de las herramientas necesarias para el desarrollo de la aplicación.

El desarrollo de este trabajo estará organizado de la siguiente manera:

Capítulo 1: Fundamentación Teórica. Se introducen los principales conceptos relacionados a mesas de coordenadas, ficheros de tipo Gerber y circuitos impresos. Además, se analizan las tendencias actuales y las tecnologías necesarias para el desarrollo del trabajo.

Capítulo 2: Soluciones Técnicas. Se exponen las características que presentará el sistema como solución a los problemas planteados, realizando un análisis crítico del sistema actual.

INTRODUCCIÓN

Capítulo 3: Exploración y planificación. Se presenta un análisis de las fases de exploración y planificación. Dando a conocer además cada uno de los artefactos generados.

Capítulo 4: Implementación y pruebas. Se lleva a cabo un análisis de las fases de implementación y prueba, dando a conocer cada uno de los artefactos generados así como una descripción de los mismos.

Capítulo 5: Estudio de la factibilidad. Se lleva a cabo un estudio de la factibilidad del sistema dando a conocer los beneficios que brinda.

1.1 Introducción

El objetivo de este capítulo es dar a conocer los principales conceptos relacionados con la creación de los circuitos impresos, las mesas de coordenadas o plotter, circuitos impresos y los ficheros de tipo Gerber. Se realizará un análisis de las tendencias actuales, así como de las herramientas y tecnologías que serán necesarias utilizar para el desarrollo del sistema.

Además se lleva a cabo un estudio de los principales algoritmos para encontrar el camino mínimo entre dos puntos estableciendo una comparación entre estos para dar solución a la problemática planteada teniendo en cuenta las características del sistema.

1.2 Conceptos fundamentales para entender la creación de los circuitos impresos.

1.2.1 ¿Qué son las mesas de coordenadas o plotters?

Un plotter o trazador gráfico es un dispositivo de impresión conectado a un ordenador, y diseñado específicamente para trazar gráficos vectoriales o dibujos lineales tales como planos, dibujos de piezas, entre otros. Esta clase de dispositivos efectúa con gran precisión impresiones gráficas que con una impresora común no se podrían lograr. Han sido empleados en diversos campos, tales como ambientes científicos, la ingeniería, el diseño, la arquitectura, entre otros. Los primeros usaban plumillas de diferentes trazos o colores pero, con la ampliación de la gama de usos y aplicaciones de los mismos, fueron transformándose hasta convertirse además en dispositivos de corte y calado [2]. (Ver anexo1)

1.2.1.1 Elementos que componen la mesa de coordenadas

La mesa de coordenadas hacia la cual tributa sus resultados este trabajo, ha sido objeto de varias modificaciones y cambios en el hardware de control e interfaz con el objetivo de adaptarla a los sucesivos avances tecnológicos. La misma consta de un cabezal móvil capaz de desplazarse en dos ejes de

Capítulo 1: Fundamentación Teórica

coordinadas, a partir de los mecanismos de precisión correspondientes, accionados por motores paso a paso. Así mismo, al cabezal pueden ser acopladas tres tipos diferentes de herramientas, según la operación que se vaya a realizar. Estas son:

- ❖ Herramienta para el barrenado o fresado. (Ver anexo1)
- ❖ Cabezal óptico.
- ❖ Cabezal portaplumas. (Ver anexo1)

1.2.2 ¿Qué es Printed Circuit Board (PCB) o circuito impreso?

Una placa de circuito impreso es una plancha de material rígido aislante, cubierta por unas pistas de cobre por una de sus caras o por ambas para servir como conexiones entre los distintos componentes que se montarán sobre ella. La materia prima consiste en esa plancha aislante, típicamente fibra de vidrio, cubierta completamente por una lámina de cobre en una o ambas caras. Dependiendo del tipo de placa, el cobre puede ir a su vez protegido por una capa de resina fotosensible. [3]

Existen varios tipos de circuitos impresos: de simple cara, de doble cara, multicapa, flexible o flexo rígido, así como otros destinados a aplicaciones especiales. En el caso específico que se trata en este trabajo, solo se pueden lograr circuitos impresos de simple o doble cara, dadas las características técnicas de la mesa de coordenadas, lo cual es suficiente para desarrollar un gran número de aplicaciones.

El término normalizado que se designa a este componente es placa impresa, pero en uso común se emplea circuito impreso. Igualmente, en inglés el término Printed Circuit Board (PCB) es de uso corriente, mientras que printed circuit se emplea prácticamente solo para referirse a la técnica de la fabricación de una placa impresa [4].

1.2.2.1 Tipos de circuitos impresos

Simple cara: Son la variedad más simple, su mayor aplicación es en consumo electrónico doméstico y entretenimiento y son los más baratos de producir [5].

Capítulo 1: Fundamentación Teórica

Doble cara: Permite la interconexión eléctrica entre las dos caras a través de agujeros metalizados. Son utilizados en productos de consumo más sofisticados y extensivamente en computadoras y equipos periféricos las placas doble caras son más caras que las simple cara [5].

Multicapa: Es un proceso de encapsulamiento de diversos circuitos en una sola capa, su uso está dirigido a las microcomputadoras, equipos aeroespaciales, instrumentación y telecomunicaciones. Las placas multicapas son el tipo de circuito impreso más caro de producir [5].

Flexible o flexo rígido: Son difíciles de crear, utilizados para ahorrar espacio principalmente en cámaras y audífonos [5].

1.2.3 ¿Qué son los ficheros de estándar Gerber?

El desarrollo de dispositivos electrónicos consta de varias etapas, que de manera general son:

1. Diseño del esquema circuital del dispositivo.
2. Simulación del mismo.
3. Diseño del circuito impreso (Printed Circuit Board o PCB).
4. Montaje y puesta a punto de un prototipo, con vista a validar un funcionamiento adecuado.
5. Fabricación en serie del dispositivo.

Las etapas 1 y 2 se llevan a cabo en distintos paquetes de programas, tales como [Proteus, Orcad, Electronic Workbench, Circuit Maker]³, entre otros. Estos constituyen verdaderos entornos de desarrollo para el diseño y la simulación de circuitos electrónicos. A su vez permiten generar, a partir del diseño esquemático y considerando determinadas restricciones establecidas por el usuario, el circuito impreso (PCB) correspondiente. El formato Gerber es un archivo que presenta todas las instrucciones para dirigir el funcionamiento del cabezal de la mesa para el proceso de creación de los mismos.

A continuación se muestra un ejemplo de archivo Gerber para ilustrar la estructura y el contenido del formato:

G90 * 1

G70 * 2

Capítulo 1: Fundamentación Teórica

G54D10 * 3
G01X0Y0D02 * 4
X450Y330D01 * 5
X455Y300D03 * 6
G54D11 * 7
Y250D03 * 8
Y200D03 * 9
Y150D03 * 10
X0Y0D02 * 11
M02 * 12

Los números de línea en el extremo derecho no son parte del fichero. Cada línea representa una orden de mando y el asterisco (*) es el final del comando. Existen varios tipos de órdenes:

- Instrucciones comenzando con G, D, M (Ver anexo2)
- X, Y datos de coordenadas.

1.3 Tendencias actuales

No existe una aplicación que se encargue específicamente de ordenar las instrucciones del fichero estándar Gerber, muchas empresas productoras como son Kompass International Neuenschwander SA, DirectIndustry venden el equipamiento completo para la creación de los circuitos impresos pero no brindan ningún dato específico relacionado con ello.

La mesa de coordenadas constituyó una base para la creación de circuitos impresos en nuestro país, dando paso a que en el Instituto Central de Investigación Digital (ICID) comenzaran a llevar a cabo dicho proceso pero con tecnologías más avanzadas. En el ICID se pueden crear desde los circuitos más simples hasta los más complejos como los multicapas que pueden ser utilizados tanto para la exportación como para el mercado nacional, y aunque han alcanzado grandes avances con este tipo de tecnología, todavía existen algunos que no son capaces de producir por las exigencias tecnológicas.

La fabricación de circuitos impresos constituye un proceso altamente costoso, es por esto que aunque la mesa de coordenadas no alcance a fabricar circuitos tan complejos como los multicapa su uso es indispensable, ya que constituye un elemento fundamental para la fabricación de prototipos de dispositivos, que son diseñados fundamentalmente con fines de investigación y desarrollo y no se tienen que poner a disposición tecnologías avanzadas y costosas para lograr este objetivo.

1.4 Estudio de los algoritmos para encontrar caminos mínimos entre dos puntos.

Las perforaciones constituyen uno de los pasos a seguir para llevar a cabo todo el proceso de creación de los circuitos impresos en la mesa de coordenadas, para lograr minimizar el costo de recursos empleados para realizar esta tarea es necesario optimizar dicho proceso, el algoritmo empleado para dar solución a esta problemática puede ser utilizado además para los procesos de pintar y fresar, a continuación se presenta un estudio de varios algoritmos para encontrar el camino mínimo entre dos puntos.

1.4.1 Algoritmo de Dijkstra

Empleado para soluciones de caminos mínimos, en la determinación del camino más corto entre un vértice origen y un vértice destino en un grafo dirigido⁴ y ponderado⁵. Su nombre se refiere a Edsger Dijkstra, quien lo describió por primera vez en 1959.

Visita todos los nodos del grafo pero el camino que devuelve no los implica a todos, constituyendo su principal desventaja pues se precisa recorrer todos los puntos especificados por las coordenadas presentes en el fichero (que no son más que los nodos del grafo) de manera óptima, o sea por la menor distancia.

1.4.2 Algoritmo de Floyd – Warshall

Se emplea para encontrar el camino más corto en grafos dirigidos y ponderados, devolviendo cada uno de los costos existentes de un vértice dado, a los restantes vértices del grafo (ver anexo 2), el mismo fue descrito en 1959 por Bernard Roy.

Al igual que el algoritmo de Dijkstra expuesto anteriormente, no brinda el recorrido completo del grafo de manera óptima.

1.4.3 Problema del Viajante de Comercio

El Problema del Viajante (también conocido como problema del viajante de comercio o por sus siglas en inglés: TSP) brinda una solución al problema planteado, pues se basa en recorrer todos nodos del grafo y que la distancia recorrida sea mínima.

A pesar de su relativa sencillez, la implementación directa del TSP no es computacionalmente aplicable debido al número de trayectorias que se precisa para obtener su resultado, siendo de orden $(n - 1)!$ cuando se conoce el punto de partida, donde n representa las coordenadas que se obtienen del fichero, que ubican cada uno de los puntos para barrenar o para el comienzo y fin de las líneas o fresas.

El TSP es considerado un problema NP-Hard⁶ debido a la cantidad de coordenadas a recorrer y la distribución de las mismas en la placa.

Una opción para resolver este problema es aplicar métodos de soluciones aproximadas (heurísticas) que consiste en encontrar soluciones que no estén garantizadas a ser óptimas pero que estén cerca del óptimo. Las heurísticas se utilizan para encontrar una solución inicial que proporcione un punto de partida para sucesivos mejoramientos [6].

Un ejemplo de heurística es el método Greedy (voraz) que consiste en hacer cada caso lo que es localmente mejor, como por ejemplo la del vecino más cercano teniendo en cuenta la distancia entre los mismos [6].

Métodos para encontrar la solución exacta del problema del viajante.

1.4.3.1 Métodos Branch and Bound

Estos métodos ven al TSP como un problema combinatorio de programación lineal entera, dentro de las ventajas que presentan es que son eficientes en problemas fáciles (aquellos en los que la ubicación de las ciudades son al azar) que tienen hasta miles de ciudades.

Pero su desventaja es que si el modelo proviene de la vida real se hace más complejo y tiene dificultades con problemas que involucren de 100 a 200 ciudades [6].

1.4.3.2 Métodos de Branch and Cut

Utilizan técnicas de programación entera combinando Branch and Bound con planos de corte, para ir eliminando restricciones. Entre las ventajas que presentan, está que son muy eficientes pueden resolver problemas de miles de ciudades (hasta 5000). Las desventajas fundamentales que presenta consisten en que es de difícil implementación y que es extremadamente dependiente del tipo de problema (para algunos funciona muy bien, pero para otros no) [6].

1.4.3.3 Teniendo en cuenta el número de secuencia en que la ciudad es visitada

Otra opción sería no agregar las U_i^7 que impiden la formación de subtours⁸ y luego de una corrida inicial, se le agregan restricciones con las U_i que sean necesarias para romper los subtours⁸ que se hayan formado, esta opción simplifica inicialmente el problema y permite una solución completa, pero es muy trabajosa y depende de la complejidad del modelo planteado (en algunos casos habrá que agregar todas las U_i por lo que el problema demora mucho en resolverse) [6].

1.4.3.4 Programación dinámica

La resolución de problemas mediante programación dinámica consiste en reducir el tiempo de ejecución de un algoritmo mediante subproblemas, pero no es factible esta solución en el caso específico de las perforaciones en el circuito impreso pues, pocas veces están dispersas, sino que siguen un orden continuo, por tanto si se divide en grupos toma más tiempo cruzar de un grupo a otro, lo mismo sucede con los procesos de pintar y fresar. (Ver anexo 2)

1.5 Ejemplo de utilización del algoritmo del viajante para el perforado de placas

Un ejemplo que ilustra lo antes mencionado y que guarda estrecha relación con el tema en cuestión es que el griego Vangelis F. Magiron utilizó el problema del viajante de comercio para el perforado de plaquetas de circuitos impresos, la tarea que se realizaba era la perforación de la plaqueta en las

Capítulo 1: Fundamentación Teórica

posiciones correspondientes a los pins de diversos componentes electrónicos, que luego serían soldados. En el caso en cuestión, se disponía de una maquinaria programable a la cual se le podía indicar el ruteo a seguir y las perforaciones a realizar. Se había detectado que la situación antes de la modelización representaba “un cuello de botella” en la producción ya que la máquina empleaba más tiempo en el movimiento lateral de la perforadora que en la perforación misma. Para corregir eso se desarrolló un modelo basado en el problema del viajante, que permitió encontrar un ruteo eficiente de la cabeza perforadora minimizando el tiempo total utilizado.

En este problema, como en la mayoría de los ejemplos que utilizan el problema del viajante, se emplearon heurísticas para su resolución. Primero determinando las coordenadas de los puntos que había que perforar, y tomando las distancias rectilíneas satisfaciendo la desigualdad triangular; luego se escribió un algoritmo que implementaba una heurística de construcción, seguida de una heurística de mejoramiento intercambiando en tres ejes, borrando en tres ejes y agregando otros tres [6].

El algoritmo fue codificado en BASIC y inicialmente tardaba 40 min con un recorrido de 100 orificios. El algoritmo se acercaba al óptimo con una diferencia del 15% en el peor de los casos y de un 10% promedio. Luego se programó en lenguaje de máquina aumentando la velocidad del algoritmo para converger a una solución óptima. Tan exitoso fue el experimento, que en junio de 1989 se comenzó a comercializar bajo el nombre de OPTODROME [6].

1.6 Metodología de desarrollo

No existe una metodología de desarrollo universal. Cada equipo de desarrollo selecciona la metodología que es más conveniente a utilizar según las características del sistema.

Para el desarrollo del sistema se decidió utilizar Xtreme Programing (XP), una metodología de desarrollo ligera basada en cuatro valores fundamentales la comunicación, retroalimentación, simplicidad y coraje que tienen como principal objetivo el aumento de la productividad a la hora de desarrollar los programas. Esta nace de la mano de Kent Beck como un proceso de creación de software diferente al convencional.

Capítulo 1: Fundamentación Teórica

Este tipo de metodología también conocida como ágil realiza la importancia de la programación del código más que la documentación del proceso por esto constituye una de las metodologías de desarrollo de software más exitosas de la actualidad.

1.6.1 ¿Por qué XP?

A continuación se muestran varias razones que conllevaron a la selección de esta metodología.

- ❖ Proyecto pequeño: XP fue creada para ser utilizada en proyectos pequeños.
- ❖ Equipo de desarrollo pequeño: XP plantea la programación en equipos pequeños (de 2 a 12 estudiantes) por lo que los miembros del equipo pueden intercambiar responsabilidades en un momento determinado.
- ❖ No existe un contrato previo especificando tiempo, recursos y alcance: Para el desarrollo del sistema no se dispone de un contrato con un presupuesto ni un alcance previamente definidos, puesto que es un proyecto para el uso interno de la empresa.
- ❖ Los requisitos del sistema cambian frecuentemente: Posibilita que el cliente pueda ir cambiando de opinión durante el desarrollo del sistema en cuanto al alcance del mismo.
- ❖ Aparición de riesgos en el desarrollo: Como se cuenta con un corto período de tiempo para la entrega del producto final y cambios frecuentes en los requerimientos pueden aparecer incontables riesgos en el mismo pero XP está diseñada para mitigarlos.
- ❖ Propiedad colectiva del código: XP plantea que todos los programadores pueden realizar cambios en cualquier parte del código en cualquier momento.
- ❖ Dentro de las prácticas que presenta XP se encuentran la de Entregas Pequeñas y Cliente en el Lugar lo que aumenta la retroalimentación entre este y el equipo de desarrollo, permitiendo satisfacer todas sus necesidades.

1.7 Herramientas y tecnologías

1.7.1 Lenguaje de programación

C# es un lenguaje de propósito general, orientado a objetos, sencillo pero a la vez poderoso. Combina los mejores elementos de múltiples lenguajes de amplia difusión como C++, Java, Visual Basic o Delphi, siendo el código autocontenido, o sea no necesita de ficheros adicionales al código fuente y el tamaño de los tipos de datos es fijo e independiente del compilador, facilitando la portabilidad del código.

C# soporta todas las características propias del paradigma de programación orientada a objetos y para facilitar la migración de programadores, C# mantiene una sintaxis muy similar a C, C++ o Java permitiendo incluir directamente en código escrito en C# fragmentos de código escrito en estos lenguajes.

Se seleccionó este lenguaje principalmente a solicitud del cliente, además por las disímiles ventajas que presenta.

1.7.2 Plataforma de desarrollo

Se seleccionó para el desarrollo de la aplicación la plataforma Microsoft Visual Studio.NET según el lenguaje de programación anteriormente especificado, ya que este fue diseñado específicamente para ser utilizado en la plataforma.NET.

Visual Studio .NET posee gestión automática de memoria, mediante un recolector de basura que evita que el programador tenga que tener en cuenta cuando ha de destruir los objetos que dejen de serle útiles, aumenta de un modo extraordinario la productividad de los desarrolladores, además de poseer editores de códigos mejorados. En la plataforma .NET las versiones nuevas de las DLLs pueden coexistir con las viejas, simplificando la instalación y desinstalación del software, eliminando así el "Infierno de las DLLs".

1.8 Conclusiones

En el presente capítulo se han dado a conocer los principales conceptos relacionados con la creación de circuitos impresos y las mesas de coordenadas o plotter, haciendo una valoración de las tendencias actuales referente al tema en cuestión.

Se ha realizado un análisis de las herramientas que se van a tener en cuenta para la implementación del sistema, así como la metodología de desarrollo que se utilizará y el lenguaje de programación, siendo estos aspectos de vital importancia para el desarrollo del sistema

Se llevó a cabo además un estudio de los principales algoritmos para encontrar el camino mínimo entre dos puntos y así dar solución a la problemática planteada.

2.1 Introducción

El presente capítulo tiene como objetivo realizar un estudio de las características del sistema que se va a desarrollar, además de llevar a cabo un análisis crítico del sistema actual y argumentar el flujo de los procesos involucrados en el campo de acción. Se plantean además los distintos requisitos funcionales del sistema y se describe el algoritmo que se empleará para desarrollar el mismo teniendo en cuenta las restricciones de hardware donde correrá la aplicación y el tamaño de la placa.

2.2 Flujo de los procesos involucrados en el campo de acción

Para realizar todo el proceso de optimización de trayectorias en el fichero de tipo Gerber primeramente se carga el fichero, dada las características presentes en el mismo se pasa al proceso de optimización a través del algoritmo específico para cada una de las funcionalidades de la mesa de coordenadas, ya sea para barrenar, fresar o pintar. Mostrando al usuario cómo está conformado el fichero antes de ser optimizado para verificar si es el correcto, además el valor de la distancia que será recorrida por el cabezal de la mesa para luego establecer una comparación con la mostrada luego de ser optimizado. Se procede a salvar el fichero el cual contendrá todas las instrucciones, ordenadas, que la mesa de coordenadas deberá realizar.

2.3 Análisis crítico del funcionamiento del sistema actual

Existen diversos pasos para la creación de circuitos impresos, cada uno con sus características pero que siguen el mismo objetivo, resulta difícil encontrar una máquina que realice este proceso completamente ya que actualmente en la creación de los circuitos impresos pueden intervenir disímiles maquinarias, además de que existen técnicas más avanzadas para su creación.

Capítulo 2: Características del Sistema

Convirtiéndose así la mesa de coordenadas en un mecanismo único, ya que ha sido adaptada para realizar completamente el proceso. Para esto es necesario el fichero de tipo Gerber, con todas las instrucciones para el funcionamiento del cabezal en la mesa para cada uno de los procesos ejecutados por la misma, pero actualmente este movimiento debe ganar en rapidez ya que provoca que se pierda tanto tiempo, como las materias primas utilizadas, por lo que se hace necesario desarrollar un algoritmo para optimizar el fichero que cumpla con dichos requerimientos.

2.4 Objeto de automatización

El proceso de creación de los circuitos impresos en el ICIMAF se lleva a cabo hace más de veinte años. Debido a los adelantos tecnológicos que se evidencian a diario la interfaz de hardware ha sido objeto de varios procesos de modernización pero no se ha logrado que el proceso se ejecute eficientemente, por lo que se hace necesario ajustar el fichero estándar Gerber, para lograr minimizar el costo de recursos empleados.

El fichero contiene todas las instrucciones que la máquina debe ejecutar, pero actualmente haciendo referencia al proceso de barrenado específicamente, cuando esta taladra un agujero no sigue un orden lógico para su ejecución por lo que provoca un elevado costo de recursos, siendo necesario que el mismo se rija por un criterio como puede ser la menor distancia entre los puntos. Además se evidenció en el proceso de pintar en el momento de rellenar un hueco, pintaba el borde de este y en vez de continuar con el relleno del mismo se dirigía a pintar otra línea y volvía de nuevo a terminar el relleno.

2.5 Características de la propuesta del sistema

El fichero estándar Gerber es el encargado de describir el funcionamiento del cabezal en la mesa de coordenadas en cuestión, para los distintos procesos que se llevan a cabo tales como fresar, barrenar o pintar. Por tanto se hace necesario un algoritmo que permita reordenar de manera óptima las instrucciones del fichero Gerber, de modo que los movimientos del cabezal de la mesa de coordenadas se realicen con mayor eficiencia.

Para dar solución a la problemática expuesta anteriormente se desarrollará una aplicación encargada de generar dichas trayectorias teniendo en cuenta minimizar el costo de recursos, mostrando además la

Capítulo 2: Características del Sistema

distancia que posteriormente será recorrida por el cabezal de la mesa antes de ser optimizada y después de realizar dicha operación.

2.5.1 Algoritmo para dar solución a la problemática planteada

Para concretar una solución teniendo en cuenta las características que se presentan en el problema, se llega a la conclusión de que el algoritmo idóneo para implementar el sistema a consecuencia de la longitud del fichero que representa el tamaño de la placa, así como las restricciones de hardware donde se ejecutará la aplicación es el problema del Viajante de Comercio mediante la solución del vecino más cercano, conocido como Nearest Neighbor.

A continuación se muestran las características de dicho algoritmo, teniendo en cuenta las distancias recorridas.

- ❖ Es una solución rápida pero no siempre óptima.
- ❖ El coste del algoritmo es n^2 donde n es el número de puntos a visitar.
- ❖ Partiendo de un nodo inicial, iterativamente busca el nodo con el que tenga menor distancia, o si existen varios, escoge uno de ellos aleatoriamente, así sucesivamente hasta recorrer todos los nodos una sola vez.
- ❖ El problema desarrollado comienza insertando las distancias entre los distintos nodos del grafo.

A continuación se muestra el pseudocódigo del algoritmo para optimizar los barrenos:

```
PROCEDURE Optimizar_Barrenos (): void
BEGIN
IF (lista_barrenos.Vacia) THEN
mensaje de error
ELSE
var: Barrenados
Arreglar_Listado(0,0)
WHILE NOT(lista_barrenos.Vacia) DO
bar:= listado_auxiliar.Obtener(listado_auxiliar.Longitud() - 1)
```

```
Arreglar_listado(ejeX:int,ejeY:int)
END
Sustituir_Listado()
END

PROCEDURE Arreglar_Listado(x: int, y: int)
BEGIN
pos:= -1
result:= 10000
FOR i:=0 TO lista_barrenos.Longitud() DO
IF (Calcular_Distancia(x1: int, y1: int, x2: int, y2: int) < result)
BEGIN
pos:= i
result:= Calcular_Distancia(x1: int, y1: int, x2: int, y2: int)
END
END
bar:= lista_barrenos.Obtener(pos: int)
lista_barrenos.Eliminar(pos: int)
listado_auxiliar.Adicionar(bar: Barrenados)
END
```

2.6 Requerimientos

Los requerimientos son una condición o capacidad que debe estar presente en un sistema o componentes de sistema para satisfacer un contrato, estándar, especificación u otro documento formal.

Los requerimientos pueden dividirse en requerimientos funcionales y requerimientos no funcionales. Los requerimientos funcionales definen las funciones que el sistema será capaz de realizar. Describen las transformaciones que el sistema realiza sobre las entradas para producir salidas.

Capítulo 2: Características del Sistema

Los requerimientos no funcionales tienen que ver con características que de una u otra forma puedan limitar el sistema, como por ejemplo, el rendimiento (en tiempo y espacio), interfaces de usuario, fiabilidad (robustez del sistema, disponibilidad de equipo), mantenimiento, seguridad, portabilidad, estándares, etc.

2.6.1 *Requerimientos funcionales*

El sistema debe ser capaz de:

R1: Gestionar Fichero.

- R1.1: Mostrar asistente para cargar el fichero.
- R1.2: Leer la información contenida en el fichero.
- R1.3: Mostrar los datos contenidos en el fichero.
- R1.4: Calcular la trayectoria que describe el fichero y Mostrar el resultado.
- R1.5: Salvar el fichero después de optimizado.

R2: Optimizar Fichero.

- R2.1: Seleccionar Opción para optimizar el fichero cargado.
- R2.2: Realizar las operaciones para el optimizado del fichero.
- R2.3: Ajustar los datos obtenidos al formato del fichero deseado.
- R2.4: Calcular la trayectoria que describe el fichero optimizado y Mostrar el resultado.
- R2.5: Mostrar el fichero optimizado.

2.6.2 *Requerimientos no funcionales*

Requerimientos de hardware:

- ❖ Pentium IV, procesador 2.2 GHZ, 512 megabytes (MB) de memoria RAM o más.

Requerimientos de software:

- ❖ Sistema operativo Windows XP, 2000 o superior
- ❖ . Net frameWork. 2.0

Diseño e implementación:

- ❖ Aplicación desarrollada en Visual Studio.Net 2005.
- ❖ Utilización de C# como lenguaje de desarrollo.

Apariencia o interfaz externa:

- ❖ Diseño sencillo y amigable.

Usabilidad:

- ❖ El sistema podrá ser usado por cualquier persona que posea conocimientos básicos en el manejo de un ordenador.
- ❖ Permitir un fácil manejo, con la capacidad de brindar confianza y seguridad en su uso

2.7 Conclusiones

Mediante el presente capítulo se presentaron las distintas características y funcionalidades que debe poseer el sistema a desarrollar, realizando un análisis crítico del sistema actual, así como un análisis de todos los procesos involucrados en el campo de acción, mostrando además cada uno de los requerimientos que el sistema debe poseer. Se presenta el algoritmo idóneo de acuerdo con las características del sistema y del hardware de la máquina donde estará instalada la aplicación.

Capítulo 3: Exploración y Planificación.

3.1 Introducción

En el presente capítulo se presentará un análisis de las fases de exploración y planificación de acuerdo con la metodología de desarrollo propuesta para la implementación del sistema. Dando a conocer además cada uno de los artefactos generados durante el desarrollo de estas, llevando a cabo una descripción de los mismos.

3.2 Fase de exploración

En esta fase, los clientes plantean a grandes rasgos las historias de usuario que son de interés para la primera entrega del producto. Al mismo tiempo el equipo de desarrollo se familiariza con las herramientas, tecnologías y prácticas que se utilizarán en el proyecto. La fase de exploración toma de pocas semanas a pocos meses, dependiendo del tamaño y familiaridad que tengan los programadores con la tecnología [7].

3.2.1 Historias de usuario

Las historias de usuario (HU) son la técnica utilizada en XP para especificar los requisitos del software. En ellas se describe brevemente las características que el sistema debe poseer, sean requisitos funcionales o no funcionales. Cada historia de usuario debe ser lo suficientemente comprensible y delimitada. Estas se escriben desde la perspectiva del cliente aunque los desarrolladores pueden brindar también su ayuda en la identificación de las mismas [7].

Durante la fase de exploración se identificaron cuatro HU las cuales se muestran a continuación.

Tabla 1- Historia de usuario Cargar fichero estándar Gerber.

Historia de Usuario	
Número: 1	Nombre: Cargar fichero estándar Gerber
Usuario: Administrador	

Capítulo 3: Exploración y Planificación

Prioridad en negocio: Alta	Riesgo de desarrollo: Bajo
Puntos estimados: 1	Iteración asignada: 1
Descripción: Para realizar este caso de uso (CU) el usuario debe seleccionar la operación de cargar fichero que se puede dividir en tres tipos tales como barrenado, plumilla o fresado.	
Observaciones:	

Tabla 2- Historia de usuario Optimizar fichero barrenado.

Historia de Usuario	
Número: 2	Nombre: Optimizar fichero barrenado
Usuario: Administrador	
Prioridad en negocio: Media	Riesgo de desarrollo: Medio
Puntos estimados: 4	Iteración asignada: 2
Descripción: A partir de que se cargue el fichero, permite optimizar dicho fichero en dependencia de las instrucciones que posea el mismo.	
Observaciones:	

Tabla 3- Historia de usuario Optimizar fichero fresado.

Historia de Usuario	
Número: 3	Nombre: Optimizar fichero fresado
Usuario: Administrador	
Prioridad en negocio: Media	Riesgo de desarrollo: Medio
Puntos estimados: 2	Iteración asignada: 2
Descripción: A partir de que se cargue el fichero, permite optimizar dicho fichero en dependencia de las instrucciones que posea el mismo.	
Observaciones:	

Capítulo 3: Exploración y Planificación

Tabla 4- Historia de usuario Optimizar fichero líneas.

Historia de Usuario	
Número: 4	Nombre: Optimizar fichero líneas
Usuario: Administrador	
Prioridad en negocio: Media	Riesgo de desarrollo: Medio
Puntos estimados: 3	Iteración asignada: 2
Descripción: A partir de que se cargue el fichero, permite optimizar dicho fichero en dependencia de las instrucciones que posea el mismo.	
Observaciones:	

Tabla 5- Historia de usuario Mostrar fichero optimizado.

Historia de Usuario	
Número: 5	Nombre: Mostrar fichero optimizado
Usuario: Administrador	
Prioridad en negocio: Baja	Riesgo de desarrollo: Medio
Puntos estimados: 1	Iteración asignada: 3
Descripción: Permite mostrar al usuario el fichero después de ser optimizado.	
Observaciones:	

Tabla 6- Historia de usuario Salvar fichero.

Historia de Usuario	
Número: 6	Nombre: Salvar fichero estándar Gerber
Usuario: Administrador	
Prioridad en negocio: Baja	Riesgo de desarrollo: Medio

Capítulo 3: Exploración y Planificación

Puntos estimados: 2	Iteración asignada: 2
Descripción: Permite al usuario salvar el fichero que se ha optimizado para su posterior uso en la mesa de coordenadas.	
Observaciones:	

3.3 Planificación

En esta fase el cliente establece la prioridad de cada historia de usuario y se realiza una estimación del esfuerzo necesario para implementar cada una de ellas.

Las estimaciones de esfuerzo asociado a la implementación de las historias de usuario las establecen los programadores utilizando como medida el punto. Un punto, equivale a una semana ideal de trabajo, donde los programadores trabajan el tiempo planeado sin ningún tipo de interrupción [8].

3.3.1 Estimación de esfuerzo por Historias de usuario

Para el desarrollo del sistema se realizó una estimación del esfuerzo para cada una de las historias de usuario identificadas, alcanzando los resultados que se muestran en la siguiente tabla.

Tabla 7- Estimación de esfuerzo por Historia de usuario.

Historia de Usuario	Puntos estimados
Cargar fichero estándar Gerber	1
Optimizar fichero barreno	4
Optimizar fichero fresado	2
Optimizar fichero líneas	2
Mostrar el fichero optimizado	1
Salvar fichero estándar Gerber	2

3.3.2 Plan de iteraciones

Al identificarse cada una de las historias de usuario y la estimación del esfuerzo dedicado a la realización de cada una de estas se procede a la etapa de implementación del proyecto [8]. La realización del sistema se va a dividir en tres iteraciones las cuales se presentan a continuación.

3.3.3 Iteración 1

El principal objetivo de esta iteración lo constituye la implementación de las HU de mayor prioridad en este caso lo constituye la HU Cargar fichero estándar Gerber como la misma presenta un riesgo de desarrollo bajo se llevará a cabo además en esta iteración la implementación de la HU Optimizar fichero barreno. Al finalizar dicha iteración el cliente podrá verificar si el fichero se cargó correctamente y una de las funcionalidades del sistema.

3.3.4 Iteración 2

Durante el transcurso de esta iteración se implementarán las HU de mediana prioridad, se desarrollarán las HU Optimizar fichero fresado y Optimizar fichero líneas, al finalizar la misma se obtendrá una versión de prueba para verificar el buen funcionamiento del sistema incluyendo las funcionalidades implementadas en la iteración anterior, manteniendo así la retroalimentación con el usuario.

3.3.5 Iteración 3

Durante el transcurso de esta iteración se implementarán las HU de baja prioridad, al finalizar la misma se obtendrá una versión 1.0 del sistema final, agregando la funcionalidad de mostrar al usuario el fichero optimizado, además se podrá salvar el fichero para su posterior uso en la mesa de coordenadas. Como resultado de la presente iteración el sistema se pondrá a prueba para evaluar su desempeño.

3.3.6 Plan de duración de iteraciones

El plan de duración de cada una de las iteraciones surge como parte del ciclo de vida de un proyecto que utiliza XP, este plan de encarga de mostrar cada una de las historias de usuario que serán desarrolladas en cada una de las distintas iteraciones, así como el orden en que serán implementadas y la duración estimada para su realización.

Tabla 8- Plan de duración de las iteraciones.

Iteración	Orden de implementación las HU	Duración de la iteración
Iteración 1	1- Cargar fichero estándar Gerber	5 semanas

Capítulo 3: Exploración y Planificación

	2- Optimizar fichero barreno	
Iteración 2	1- Optimizar fichero fresado 2- Optimizar fichero líneas	4 semanas
Iteración 3	1- Mostrar fichero optimizado 2- Salvar fichero estándar Gerber	3 semanas

3.3.7 Plan de entregas del sistema

Al finalizar cada una de las iteraciones se desarrollarán releases del sistema en la fecha que se indica en la tabla a continuación, el plan de entregas del sistema es fundamental para la fase de implementación.

Tabla 9- Plan de Entregas.

Historias De Usuario	Final de la 1era Iteración 3era semana de marzo	Final de la 2da Iteración 4ta semana de abril	Final de la 3era Iteración 3era semana de mayo
Cargar fichero estándar Gerber Optimizar Fichero barreno	v0.1	v0.2	v1.0
Optimizar Fichero fresado Optimizar fichero líneas		v0.1	v1.0
Mostrar fichero optimizado Salvar fichero estándar Gerber			v1.0

3.4 Conclusiones

En el presente capítulo se abordó todo lo referente a las fases de exploración y planificación, dando a conocer una descripción de cada uno de los artefactos generados en cada etapa, mostrando además cada una de las iteraciones que se van a llevar a cabo, constituyendo así el plan de entregas de las distintas versiones del sistema, logrando una mejor planificación en cuanto a tiempo y recursos.

Capítulo 4: Implementación y Pruebas

4.1 *Introducción.*

El desarrollo es la parte más importante en el proceso de XP. Todos los trabajos tienen como objetivo que se programen lo más rápido posible, sin interrupciones y en dirección correcta.

Durante el transcurso de las iteraciones se realiza la implementación de las historias de usuario seleccionadas para cada una de estas. Al inicio de las mismas, se lleva a cabo una revisión del plan de iteraciones y se modifica de ser necesario [8].

Al finalizar cada iteración se debe obtener un producto final que debe ser mostrado a los clientes para que este sea probado y así mantener la retroalimentación entre estos y los desarrolladores.

En el presente capítulo se detallan las tres iteraciones que se van a llevar a cabo para la construcción del sistema, así como las tareas generadas por cada una de las HU y las pruebas de aceptación efectuadas en el sistema.

4.2 *Diseño del sistema*

XP no requiere de la representación del sistema mediante diagramas de clases utilizando UML (Lenguaje Unificado de Modelado) para el diseño de las aplicaciones se pone en práctica el uso de las tarjetas CRC (Clase, Responsabilidad y Colaboración) [7], constituyendo uno de los pasos más importantes que se llevan a cabo en un proyecto orientado a objetos, ya que son fáciles de usar y animan la interacción. Los diagramas también pueden ser utilizados siempre y cuando influyan en el mejoramiento de la comunicación.

Con el objetivo de una mejor comprensión de las distintas funcionalidades brindadas por la aplicación se muestra una tarjeta CRC para cada clase, con la finalidad de obtener un diseño simple y al mismo tiempo no incurrir en características que no son necesarias.

Capítulo 4: Implementación y Pruebas

Tabla 10- Tarjeta CRC Clase Barrenados.

Barrenados	
Responsabilidad	Colaboración
Devolver coordenadas de los puntos.	
Cambiar coordenadas de los puntos.	

Tabla 11- Tarjeta CRC Clase Líneas.

Líneas	
Responsabilidad	Colaboración
Devolver coordenadas de las líneas.	
Cambiar coordenadas de las líneas.	

Tabla 12- Tarjeta CRC Clase Control_Gerber.

Control_Gerber	
Responsabilidad	Colaboración
Llenar lista de barrenos	Barrenados
Obtener barreno	Barrenados
Devolver lista de barrenos	Barrenados
Arreglar listado	Barrenados
Optimizar barreno	Barrenados
Calcular distancia	
Sustituir listado	Barrenados

Tabla 13- Tarjeta CRC Clase Optimizador_Líneas.

Optimizador_Líneas	
Responsabilidad	Colaboración
Llenar lista de líneas	Líneas
Devolver lista de barrenos	Líneas

Capítulo 4: Implementación y Pruebas

Calcular punto más cercano	Líneas
Arreglar listado	Líneas
Optimizar listado	Líneas
Calcular distancia	
Sustituir listado	Líneas

En el transcurso de cada una de las iteraciones se lleva a cabo la implementación de las HU representadas, al inicio de cada una estas, se realiza una revisión del plan de iteraciones y se modifica de ser necesario.

A continuación se muestran las iteraciones planificadas para el desarrollo del sistema, obteniendo al finalizar el respectivo producto final.

4.3 Iteración 1

La primera iteración tuvo como objetivo desarrollar las HU de mayor prioridad, donde se carga el fichero estándar Gerber, mostrándolo al usuario para verificar si es el correcto, continuando con el proceso de optimizar el fichero para barrenar, mediante el algoritmo específico.

Tabla 14- Historias de usuario abordadas en la primera iteración.

Historia de Usuario	Estimación	Real
Cargar fichero estándar Gerber	1	1
Optimizar Fichero barreno	4	4
Total	5	5

4.3.1 Tareas de las Historias de usuario abordadas en la iteración.

Cargar fichero estándar Gerber

Tabla 15- Tarea # 1 de la Historia de usuario Cargar fichero estándar Gerber.

Tarea	
Número de tarea: 1	Número de la HU: 1
Nombre de tarea: Cargar fichero estándar Gerber	

Capítulo 4: Implementación y Pruebas

Tipo de tarea: Desarrollo	Puntos Estimados: 0.50
Fecha de inicio: 2 de febrero de 2009	Fecha de fin: 4 de febrero de 2009
Responsables: Jesús Borrego y Yusibel Cardo	
Descripción: Se lee del fichero de barreno las coordenadas de los puntos y se almacenan en variables del sistema.	

Tabla 16- Tarea # 2 de la Historia de usuario Cargar fichero estándar Gerber.

Tarea	
Número de tarea: 2	Número de la HU: 1
Nombre de tarea: Mostrar fichero estándar Gerber	
Tipo de tarea: Desarrollo	Puntos Estimados: 0.20
Fecha de inicio: 5 de febrero de 2009	Fecha de fin: 9 de febrero de 2009
Responsables: Jesús Borrego y Yusibel Cardo	
Descripción: Luego de cargar el fichero es mostrado al usuario para verificar que este sea el correcto.	

Tabla 17- Tarea # 3 de la Historia de usuario Cargar fichero estándar Gerber.

Tarea	
Número de tarea: 3	Número de la HU: 1
Nombre de tarea: Cálculo de distancia sin optimizar	
Tipo de tarea: Desarrollo	Puntos Estimados: 0.10
Fecha de inicio: 10 de febrero de 2009	Fecha de fin: 11 de febrero de 2009
Responsables: Jesús Borrego y Yusibel Cardo	
Descripción: Luego de cargar el fichero se lleva a cabo el cálculo de la trayectoria que describe el mismo sin optimizar.	

Tabla 18- Tarea # 4 de la Historia de usuario Cargar fichero estándar Gerber.

Tarea	
Número de tarea: 4	Número de la HU: 1
Nombre de tarea: Mostrar cálculo de distancia sin optimizar	
Tipo de tarea: Desarrollo	Puntos Estimados: 0.10

Capítulo 4: Implementación y Pruebas

Fecha de inicio: 12 de febrero de 2009	Fecha de fin: 13 de febrero de 2009
Responsables: Jesús Borrego y Yusibel Cardo	
Descripción: Luego de efectuar el cálculo de la trayectoria que describe el fichero sin optimizar es mostrado al usuario.	

Optimizar Fichero barreno

Tabla 19- Tarea # 1 de la Historia de usuario Optimizar fichero barreno.

Tarea	
Número de tarea: 1	Número de la HU: 2
Nombre de tarea: Optimizar fichero estándar Gerber (Barreno)	
Tipo de tarea: Desarrollo	Puntos Estimados: 3.8
Fecha de inicio: 16 de febrero de 2009	Fecha de fin: 12 de marzo de 2009
Responsables: Jesús Borrego y Yusibel Cardo	
Descripción: Utilización de algoritmos para ajustar el fichero de manera óptima atendiendo los requerimientos de hardware que se precisan.	

Tabla 20- Tarea # 2 de la Historia de usuario Optimizar fichero barreno.

Tarea	
Número de tarea: 2	Número de la HU: 2
Nombre de tarea: Cálculo de distancia optimizado	
Tipo de tarea: Desarrollo	Puntos Estimados: 0.1
Fecha de inicio: 13 de marzo de 2009	Fecha de fin: 16 de marzo 2009
Responsables: Jesús Borrego y Yusibel Cardo	
Descripción: Luego de cargar el fichero se lleva a cabo el cálculo de la trayectoria que describe el mismo luego de ser optimizado.	

Tabla 21- Tarea # 3 de la Historia de usuario Optimizar fichero barreno.

Tarea	
Número de tarea: 3	Número de la HU: 2

Capítulo 4: Implementación y Pruebas

Nombre de tarea: Mostrar cálculo de distancia optimizado	
Tipo de tarea: Desarrollo	Puntos Estimados: 0.1
Fecha de inicio: 17 de marzo de 2009	Fecha de fin: 18 de marzo 2009
Responsables: Jesús Borrego y Yusibel Cardo	
Descripción: Luego de efectuar el cálculo de la trayectoria que describe el fichero optimizado es mostrado al usuario.	

4.4 Iteración 2

Mediante el transcurso de la segunda iteración se desarrollaron las HU de mediana prioridad, en este caso optimizar el fichero de fresado y el fichero para pintar.

Tabla 22- Historias de usuario abordadas en la segunda iteración.

Historia de Usuario	Estimación	Real
Optimizar Fichero fresado	2	2
Optimizar fichero líneas	3	3
Total	5	5

Optimizar Fichero fresado

Tabla 23- Tarea # 1 de la Historia de usuario Optimizar fichero fresado.

Tarea	
Número de tarea: 1	Número de la HU: 3
Nombre de tarea: Optimizar fichero estándar Gerber (fresado)	
Tipo de tarea: Desarrollo	Puntos Estimados: 2
Fecha de inicio: 19 de marzo de 2009	Fecha de fin: 2 de abril de 2009
Responsables: Jesús Borrego y Yusibel Cardo	
Descripción: Utilización de algoritmos para ajustar el fichero de manera óptima atendiendo los requerimientos de hardware que se precisan.	

Optimizar fichero líneas

Capítulo 4: Implementación y Pruebas

Tabla 24- Tarea # 1 de la Historia de usuario Optimizar fichero líneas.

Tarea	
Número de tarea: 1	Número de la HU: 4
Nombre de tarea: Optimizar fichero estándar Gerber (pintado)	
Tipo de tarea: Desarrollo	Puntos Estimados: 3
Fecha de inicio: 3 de abril de 2009	Fecha de fin: 24 de abril de 2009
Responsables: Jesús Borrego y Yusibel Cardo	
Descripción: Utilización de algoritmos para ajustar el fichero de manera óptima atendiendo los requerimientos de hardware que se precisan.	

4.5 Iteración 3

Durante la presente iteración se implementaron las restantes historias de usuario que involucran Mostrar fichero optimizado y Salvar el fichero. Al finalizar dicha iteración se obtendrá un producto listo para su puesta en funcionamiento.

Tabla 25- Historias de usuario abordadas en la tercera iteración.

Historia de Usuario	Estimación	Real
Mostrar fichero optimizado	2	2
Salvar fichero estándar Gerber	1	1
Total	3	3

Mostrar fichero optimizado

Tabla 26- Tarea # 1 de la Historia de usuario Mostrar fichero optimizado.

Tarea	
Número de tarea: 1	Número de la HU: 5
Nombre de tarea: Ajustar fichero estándar Gerber	
Tipo de tarea: Desarrollo	Puntos Estimados: 1.5
Fecha de inicio: 27 de abril de 2009	Fecha de fin: 6 de mayo 2009
Responsables: Jesús Borrego y Yusibel Cardo	

Capítulo 4: Implementación y Pruebas

Descripción: Una vez optimizado el fichero se convierte al formato que interpreta la mesa de coordenadas.

Tabla 27- Tarea # 2 de la Historia de usuario Mostrar fichero optimizado.

Tarea	
Número de tarea: 2	Número de la HU: 5
Nombre de tarea: Mostrar fichero estándar Gerber optimizado	
Tipo de tarea: Desarrollo	Puntos Estimados: 0.5
Fecha de inicio: 7 de mayo de 2009	Fecha de fin: 12 de mayo 2009
Responsables: Jesús Borrego y Yusibel Cardo	
Descripción: Una vez optimizado el fichero es mostrado al usuario para verificar que este sea correcto.	

Salvar Fichero

Tabla 28- Tarea # 1 de la Historia de usuario Salvar fichero estándar Gerber.

Tarea	
Número de tarea: 1	Número de la HU: 6
Nombre de tarea: Salvar fichero estándar Gerber	
Tipo de tarea: Desarrollo	Puntos Estimados: 1
Fecha de inicio: 13 de mayo de 2009	Fecha de fin: 20 de mayo 2009
Responsables: Jesús Borrego y Yusibel Cardo	
Descripción: Se salva el fichero para su posterior uso en la mesa de coordenadas	

4.6 Pruebas

Uno de los pilares fundamentales de XP es el proceso de pruebas [7]. Anima a probar constantemente tanto como sea posible. Esto permite aumentar la calidad de los sistemas reduciendo el número de errores no detectados y disminuyendo el tiempo transcurrido entre la aparición de un error y su detección. También permite aumentar la seguridad de evitar efectos colaterales no deseados a la hora de realizar modificaciones y refactorizaciones.

Capítulo 4: Implementación y Pruebas

XP divide las pruebas del sistema en dos grupos [7]: pruebas unitarias, encargadas de verificar el código y diseñada por los programadores, y pruebas de aceptación o pruebas funcionales destinadas a evaluar si al final de una iteración se consiguió la funcionalidad requerida diseñadas por el cliente final.

4.7 Pruebas de Aceptación

Las pruebas de aceptación son más importantes que las pruebas unitarias dado que significan la satisfacción del cliente con el producto desarrollado y el final de una iteración y el comienzo de la siguiente [9].

Tabla 29- Pruebas de aceptación para la Historia de usuario Cargar fichero estándar Gerber.

Prueba de Aceptación	
Código: HU1_P1	Historia de Usuario: Cargar fichero estándar Gerber.
Nombre: Mostrar fichero sin optimizar	
Descripción: Prueba de funcionalidad para mostrar fichero sin optimizar.	
Condiciones de ejecución: Se debe cargar el fichero.	
Entrada / Pasos de Ejecución: Se accede a la opción del menú Abrir fichero y se muestra el fichero sin optimizar.	
Resultado esperado: El fichero es mostrado correctamente.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 30- Pruebas de aceptación para la Historia de usuario Cargar fichero estándar Gerber.

Prueba de Aceptación	
Código: HU1_P2	Historia de Usuario: Cargar fichero estándar Gerber.

Capítulo 4: Implementación y Pruebas

Nombre: Mostrar cálculo de la distancia sin optimizar.
Descripción: Prueba de funcionalidad para mostrar cálculo de la distancia sin optimizar.
Condiciones de ejecución: Se debe cargar el fichero.
Entrada / Pasos de Ejecución: Se accede a la opción del menú Abrir fichero, se muestra el fichero sin optimizar y el cálculo la distancia.
Resultado esperado: El cálculo de la distancia es mostrado correctamente.
Evaluación de la prueba: Prueba satisfactoria.

Tabla 31- Pruebas de aceptación para la Historia de usuario Mostrar fichero optimizado.

Prueba de Aceptación	
Código: HU1_P3	Historia de Usuario: Mostrar fichero optimizado.
Nombre: Mostrar cálculo de la distancia del fichero Gerber optimizado.	
Descripción: Prueba de funcionalidad para mostrar cálculo de la distancia sin optimizar.	
Condiciones de ejecución: Se debe cargar y optimizar el fichero.	
Entrada / Pasos de Ejecución: Se accede a la opción del menú Abrir fichero, luego seleccionar la opción de Optimizar fichero.	
Resultado esperado: El cálculo de la distancia es mostrado correctamente.	
Evaluación de la prueba: Prueba satisfactoria.	

Capítulo 4: Implementación y Pruebas

Tabla 32- Pruebas de aceptación para la Historia de usuario Mostrar fichero optimizado.

Prueba de Aceptación	
Código: HU1_P4	Historia de Usuario: Mostrar fichero optimizado.
Nombre: Mostrar fichero optimizado	
Descripción: Prueba de funcionalidad para mostrar fichero optimizado.	
Condiciones de ejecución: Se debe cargar y optimizar el fichero.	
Entrada / Pasos de Ejecución: Se accede a la opción de menú Abrir fichero, luego seleccionar la opción Optimizar fichero.	
Resultado esperado: El fichero es mostrado correctamente.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 33- Pruebas de aceptación para la Historia de usuario Salvar fichero estándar Gerber.

Prueba de Aceptación	
Código: HU1_P5	Historia de Usuario: Salvar fichero estándar Gerber.
Nombre: Salvar el fichero optimizado.	
Descripción: Prueba de funcionalidad para salvar el fichero optimizado.	
Condiciones de ejecución: Se debe cargar y optimizar el fichero, además ser mostrado al usuario.	
Entrada / Pasos de Ejecución: Se accede a la opción de menú Abrir fichero, luego seleccionar la opción Optimizar fichero y la opción Salvar fichero.	
Resultado esperado: El fichero se guarda correctamente.	
Evaluación de la prueba: Prueba satisfactoria.	

Capítulo 4: Implementación y Pruebas

UCI Universidad Católica
de Chile
Informáticas

Vicerrectoría de Formación
Dirección de Formación del Profesional
Departamento de Práctica Profesional

OPINIÓN DEL USUARIO DEL TRABAJO DE DIPLOMA

El Trabajo de Diploma, titulado "Desarrollo de una aplicación para la optimización de trayectorias en una mesa de coordenadas" fue realizado en el Instituto de Cibernética Matemática y Física (ICIMAF). Esta entidad considera que, en correspondencia con los objetivos trazados, el trabajo realizado le satisface

Totalmente
Parcialmente en un _____ %

Los resultados de este Trabajo de Diploma le reportan a esta entidad los beneficios siguientes (cuantificar):

El resultado obtenido permite aumentar la eficiencia en el proceso de fabricación de circuitos impresos, imprescindible para la obtención de los prototipos desarrollados en el ICIMAF como parte de los proyectos de investigación-desarrollo. Los métodos de optimización de trayectorias empleados logran una reducción importante del tiempo necesario para este proceso (alrededor de un 7% en pruebas preliminares) y por tanto un ahorro de recursos energéticos, tiempo de trabajo y menor desgaste del equipamiento.

Los resultados se presentan con la calidad requerida.

Y para que así conste, se firma la presente a los 1 días del mes de Junio del año 2009

MSc. Guillermo Alvarez Bestard
Representante de la entidad

Jefe del Departamento de Control Automático
Cargo


Firma



4.8 Pruebas de Unidad

Tabla 34- Pruebas de unidad a la Historia de usuario Cargar fichero estándar Gerber.

Prueba de unidad	
Historia de usuario:	Cargar fichero estándar Gerber
Componente:	Form1
Funcionalidad:	abrirFicheroToolStripMenuItem_Click
Objetivos:	Llenar la lista con los datos contenidos en el fichero.
Descripción:	Para realizar la prueba, se verificó que el fichero que se carga sea el correcto y que los datos queden guardados en la lista, mostrando al usuario el contenido del mismo y además el cálculo de la distancia recorrida sin optimizar.
Resultados esperados:	En caso de haber ocurrido algún error al cargar el fichero el sistema muestra un mensaje de error, sino llena la lista con los datos contenidos en el mismo y se muestra al usuario para verificar que es el deseado.
Observaciones:	

Tabla 35- Pruebas de unidad a la Historia de usuario Optimizar fichero barreno.

Prueba de unidad	
Historia de usuario:	Optimizar fichero barreno
Componente:	Control_Gerber
Funcionalidad:	Optimizar_Barrenos
Objetivos:	Optimizar el fichero
Descripción:	Para realizar la prueba se verifica que la lista contenga los datos del fichero, se realiza la optimización de la lista de acuerdo al cálculo de las distancias existentes entre cada barreno con tal de que se permita minimizar el recorrido del cabezal, ajustando el orden del listado a estas nuevas condiciones.

Capítulo 4: Implementación y Pruebas

Resultados esperados:	En caso de que la lista no contenga datos se muestra un mensaje de error, sino se devuelve la lista optimizada.
Observaciones:	

Tabla 36- Pruebas de unidad a la Historia de usuario Optimizar fichero líneas.

Prueba de unidad	
Historia de usuario:	Optimizar fichero líneas
Componente:	Optimizador_Lineas
Funcionalidad:	Optimizar_Lineas
Objetivos:	Optimizar el fichero
Descripción:	Para realizar la prueba se verifica que la lista contenga los datos del fichero, se realiza la optimización de la lista de acuerdo al cálculo de las distancias existentes entre cada línea con tal de que se permita minimizar el recorrido del cabezal siendo en algunos casos necesario invertir el criterio de inicio de línea, ajustando el orden del listado a estas nuevas condiciones.
Resultados esperados:	En caso de que la lista no contenga datos se muestra un mensaje de error, sino se devuelve la lista optimizada.
Observaciones:	

Tabla 37- Pruebas de unidad a la Historia de usuario Mostrar fichero optimizado.

Prueba de unidad	
Historia de usuario:	Mostrar fichero optimizado
Componente:	Form1
Funcionalidad:	Mostrar_Listado_Optimizado
Objetivos:	Mostrar los datos contenidos en el fichero optimizado.

Capítulo 4: Implementación y Pruebas

Descripción:	Para realizar la prueba se guardan los datos del fichero en una lista y se completa el formato de las coordenadas de acuerdo al estándar Gerber.
Resultados esperados:	Se muestra el fichero optimizado.
Observaciones:	

Tabla 38- Pruebas de unidad a la Historia de usuario Salvar fichero estándar Gerber.

Prueba de unidad	
Historia de usuario:	Salvar fichero estándar Gerber
Componente:	Form1
Funcionalidad:	salvarFicheroToolStripMenuItem_Click
Objetivos:	Salvar el fichero optimizado.
Descripción:	Para realizar la prueba, se verificó que la lista optimizada contenga datos, generando el fichero.
Resultados esperados:	Se salva el fichero optimizado.
Observaciones:	

4.9 Conclusiones

En el presente capítulo se llevaron a cabo las fases de implementación y prueba, se desarrollaron cada una de las tareas correspondientes para dar solución a las historias de usuario, además las pruebas de unidad y de aceptación para verificar las distintas funcionalidades del sistema que propicien al usuario conformidad y seguridad ante el mismo. Al finalizar el capítulo se cumple con la propuesta que trae consigo el presente trabajo de diploma.

5.1 *Introducción*

La estimación constituye una de las tareas de mayor importancia en la planificación de los proyectos de software. Aporta un número de horas aproximado que habrá que combinar con los recursos para obtener la planificación de actividades en el tiempo y establecer entonces los hitos del proyecto. En el presente capítulo se realiza una estimación del esfuerzo para la realización del sistema propuesto así como un estudio de la factibilidad para llevar a cabo el desarrollo del mismo.

5.2 *COCOMO II*

COCOMO II consiste básicamente en la aplicación de ecuaciones matemáticas sobre los Puntos de Función sin ajustar o la cantidad de líneas de código (SLOC, Source Lines Of Code) estimados para un proyecto. Estas ecuaciones se encuentran ponderadas por ciertos factores de costo (cost drivers) que influyen en el esfuerzo requerido para el desarrollo del software [11].

Proporciona una familia de modelos de estimación de coste de software cada vez más detallado y tiene en cuenta las necesidades de cada sector y el tipo de información disponible para sostener la estimación del coste del software [10].

5.3 *Características del sistema*

Para llevar a cabo la estimación del proyecto se deben de obtener los puntos de función desajustados, mediante las consultas, entradas y salidas externas del sistema así como los archivos lógicos internos y los de interfaz externa. Mediante las siguientes tablas se muestran las características del sistema.

5.3.1 Entradas externas

Las entradas externas están definidas como un proceso elemental mediante el cual ciertos datos cruzan la frontera del sistema desde afuera hacia adentro [11]. A continuación se muestran cada una de las entradas externas que presenta el sistema.

Tabla 39- Entradas externas.

Nombre de la entrada externa	Cantidad de ficheros	Cantidad de elementos de datos	Clasificación (simple, media, compleja)
Cargar fichero	1	1	simple
Total		1	simple

5.3.2 Salidas externas

Las salidas externas se pueden definir como un proceso elemental con componentes de entrada y de salida mediante el cual datos simples y datos derivados (esto es, datos que se calculan a partir de otros datos) cruzan la frontera del sistema desde adentro hacia afuera [11]. A continuación se muestran las salidas externas del sistema.

Tabla 40- Salidas externas.

Nombre de la salida externa	Cantidad de ficheros	Cantidad de elementos de datos	Clasificación (simple, media, compleja)
Mostrar fichero optimizado	1	1	simple
Salvar fichero optimizado	1	1	simple
Mostrar mensaje de error		1	simple
Total		3	simple

5.3.3 Consultas externas

Se definen como un proceso elemental con componentes de entrada y de salida donde un Actor del

Capítulo 5: Estudio de Factibilidad

Sistema rescata datos de uno o más Archivos Lógicos Internos o Archivos de Interfaz Externos. Los datos de entrada no actualizan ni mantienen ningún archivo (lógico interno o de interfaz externo) y los datos de salida no contienen datos derivados (es decir, los datos de salida son básicamente los mismos que se obtienen de los archivos). Dentro de éste tipo de transacción entran los listados y las búsquedas de los sistemas [11].

Tabla 41- Consultas externas.

Nombre de la consulta externa	Cantidad de ficheros	Cantidad de elementos de datos	Clasificación (simple, media, compleja)
Menú de Acciones	0	3	simple
Total		3	

5.3.4 Archivos lógicos internos

Los archivos lógicos internos no son más que datos relacionados lógicamente e identificables por el usuario, que residen enteramente dentro de los límites del sistema y se mantienen a través de las Entradas Externas [11].

Tabla 42- Archivos lógicos internos.

Nombre de los archivos lógicos internos	Cantidad de records	Cantidad de elementos de datos	Clasificación (simple, media, compleja)
Total		0	

5.3.5 Archivos de interfaz externos

Los archivos de interfaz externos se pueden definir como un grupo de datos relacionados lógicamente e identificables por el usuario, que se utilizan solamente para fines de referencia. Los datos residen enteramente fuera de los límites del sistema y se mantienen por las Entradas Externas de otras aplicaciones, es decir, cada Archivo de Interfaz Externo es un Archivo Lógico Interno de otra aplicación.

Capítulo 5: Estudio de Factibilidad

Tabla 43- Archivos de interfaz externos.

Nombre de los archivos lógicos internos	Cantidad de records	Cantidad de elementos de datos	Clasificación (simple, media, compleja)
Total		0	

5.3.6 Puntos de función sin ajustar

Tabla 44- Puntos de función sin ajustar.

Elementos	Simple		Medio		Complejo		Subtotal
	No.	Peso	No.	Peso	No.	Peso	
Entrada Externa	1	3	0	4	0	6	3
Salida Externa	2	4	0	5	0	7	8
Archivos de interfaz externos	0	3	0	4	0	6	0
Fichero Lógico Interno	0	7	0	10	0	15	0
Consulta Externa	3	5	0	7	0	15	15
Total (UFP)							26

5.4 Cálculo de instrucciones fuentes, esfuerzo, tiempo de desarrollo, cantidad de hombres y costo

Luego de obtener el total de puntos sin ajustar (UFP) mediante la siguiente fórmula, se procede a calcular la cantidad de instrucciones fuentes del sistema.

$$\text{SLOC} = \text{UFP} \times \text{ratio}$$

Donde:

SLOC: Cantidad de instrucciones fuentes.

UFP: Puntos de función sin ajustar.

Capítulo 5: Estudio de Factibilidad

ratio: Conversión de puntos de función desajustados a líneas de código para el lenguaje c#.

A partir de la ecuación anterior se obtiene el siguiente resultado:

$$\text{SLOC} = 26 \times 53$$

$$\text{SLOC} = 1378 = 1.378 \text{ KSLOC}$$

Tabla 45- Características generales del proyecto.

Características	Valor
Puntos de función sin ajustar	26
Lenguaje (c#)	53
Instrucciones fuentes por puntos de función	1378 SLOC
Instrucciones fuentes	1.378 KSLOC

5.4.1 Cálculo del esfuerzo nominal

El cálculo del esfuerzo nominal se toma como base en el método del diseño preliminar, anteriormente expuesto, así como para el modelo post arquitectura los cuales son definidos por COCOMO II.

$$\text{PM}_{\text{nominal}} = A \times (\text{Size})^B$$

$$\text{PM}_{\text{nominal}} = 2.94 * (1.378)^{1.01}$$

$$\text{PM}_{\text{nominal}} = 4.06 \text{ meses /hombre}$$

Donde:

PM_{nominal}: Esfuerzo nominal requerido en meses – hombre.

Size: Tamaño estimado del software, en Puntos de Función sin ajustar (KSLOC).

A: Constante (Valor: 2.94).

Capítulo 5: Estudio de Factibilidad

B: Constante denominada Factor escalar y su valor esta dado por la resultante de los aspectos positivos sobre los negativos que presenta el proyecto

$$B = 0.91 + 0.01 \times \sum (W_i) [11]$$

$$B = 0.91 + 0.01 \times 9.65$$

$$B = 1.01$$

Donde:

W_i : Indica las características del proyecto a través de variables escalares que especifican la complejidad y el entorno de desarrollo. Mediante la siguiente tabla se muestran cada uno de los valores.

Tabla 46- Factores de escala.

Nombre	Valor	Justificación
PREC	4.96	Actualmente no hay sistemas que cumplan con las funcionalidades del software a desarrollar
TEAM	1.10	La interacción del equipo es altamente cooperativa.
PMAT	1.56	Se cuenta con la experiencia necesaria para que el software cumpla con las funcionalidades requeridas.
FLEX	2.03	Cuenta con una alta flexibilidad
Total (Wi)	9.65	

5.4.1.2 Ajuste del esfuerzo nominal

Para ajustar el valor determinado anteriormente se deben de obtener los Multiplicadores de Esfuerzo (ME) que representan las características del proyecto y expresan su impacto en el desarrollo total del producto de software [11]. A continuación se muestra la tabla con cada uno de los valores correspondientes.

Tabla 47- Multiplicadores de esfuerzo.

Nombre	Valor	Justificación
SCED	1.00	Se requiere terminar el proyecto en el tiempo estimado.
PERS	1.00	Alta capacidad del personal.

Capítulo 5: Estudio de Factibilidad

FCIL	1.00	La utilización de entornos de desarrollo integrados, así como de herramientas CASE simples facilitan en gran medida el trabajo.
RUSE	1.00	Se pretende reutilizar una parte del código.
RCPX	1.74	La complejidad del sistema es alta
Total (EM)	1.74	

$$PM_{\text{ajustado}} = PM_{\text{nominal}} \times \prod (EM_i)$$

$$PM_{\text{ajustado}} = 4.06 \text{ meses/hombre} \times 1.74$$

$$PM_{\text{ajustado}} = 7.06 \text{ hombres/mes}$$

5.4.2 Cálculo del tiempo de desarrollo del software

La siguiente ecuación muestra el tiempo requerido para el proyecto.

$$TDEV = C \times (PM_{\text{ajustado}})^F$$

$$TDEV = 3.64 \times (7.06)^{0.26}$$

$$TDEV = 6.05$$

Donde:

C: Constante de valor 3.64.

$$F = D + 0.2 \times 0.01 \times \sum SF$$

$$F = 0.24 + 0.2 \times 0.01 \times 9.65$$

$$F = 0.26$$

Donde:

D: constante cuyo valor es 0.24

SF: valor de los factores de escala

5.4.3 Cálculo del costo total del proyecto

Para el cálculo del costo total correspondiente al proyecto en cuestión, COCOMO II define la siguiente ecuación:

$$C = CHM \times PM$$

$$C = 200 \times 7.06$$

$$C = 1412$$

Donde:

C : Costo total

CHM: Costo del salario de los obreros, que se calcula mediante la siguiente ecuación.

$$CHM = CH \times Sal$$

$$CHM = 2 \times 100$$

$$CHM = 200$$

Donde:

CH: Cantidad de personas destinadas al proyecto, se calcula a través de la siguiente ecuación.

$$CH = PM / TDEV$$

$$CH = 7.06 / 6.05 \approx 2$$

Donde:

Sal: Salario por cada trabajador.

Mediante la siguiente tabla se muestran los resultados obtenidos a través de los cálculos para determinar el esfuerzo y costo para llevar a cabo el desarrollo del sistema.

Tabla 48- Resultados.

	Valor
Esfuerzo	7.06 hombres/ mes

Capítulo 5: Estudio de Factibilidad

Tiempo de desarrollo	6.1 meses
Cantidad de hombres	2 hombres
Salario medio	\$ 100
Costo	\$ 1412

5.5 Beneficios tangibles e intangibles

El desarrollo del sistema propuesto brinda como principal beneficio al ICIMAF una solución a las limitantes presentadas debido al gasto innecesario de tiempo y recursos, a consecuencia de la falta de un algoritmo para optimizar las trayectorias del cabezal en la mesa de coordenadas a la hora de ejecutar las distintas funcionalidades del mismo

El presente sistema no está concebido como un producto de carácter comercial, pero este puede ser empleado en cualquier institución que lo necesite por los distintos beneficios que reporta, y en un futuro puede reportar beneficios monetarios a la empresa en cuestión si adquiere este carácter.

5.6 Análisis del costo

De acuerdo a los beneficios que reporta el desarrollo de un producto, justifica el costo que conlleva la realización del mismo. El sistema que se propone no reporta grandes gastos pues solo influye el salario de los desarrolladores, por lo cual se puede concluir que su implementación es factible, aunque se debe de tener en cuenta la plataforma sobre la cual correrá la aplicación requiere de una licencia para su uso.

5.7 Conclusiones

En el presente capítulo se realizó un análisis de factibilidad del sistema propuesto, teniendo en cuenta tanto los costos como los beneficios, tangibles e intangibles del mismo, llegando a la conclusión de que es viable su desarrollo.

CONCLUSIONES GENERALES

A modo de colofón, se ha desarrollado una aplicación que permite optimizar los ficheros Gerber siguiendo un criterio de cercanía entre los elementos físicos que se describen en el contenido del mismo. Logrando así mejorar el rendimiento de la mesa de coordenadas durante el proceso de fabricación de circuitos impresos, lo cual implica una sensible disminución en el tiempo de utilización de la mesa y optimiza el recorrido del cabezal de la misma, con el consecuente ahorro de energía y contribuyendo al mantenimiento y conservación de las partes mecánicas de la misma.

RECOMENDACIONES

Se recomienda que, para próximas versiones de la aplicación, se tengan en cuenta los cambios de dirección del cabezal de la mesa de coordenadas; pues, según pruebas realizadas, se llegó a la conclusión de que estos influyen en el tiempo de funcionamiento de la misma. Además evaluar otras variantes de implementación del algoritmo del problema del viajante, específicamente Métodos de Branch and Cut, cuyo desempeño varía de una aplicación a otra.

BIBLIOGRAFÍA

1. **Sopena Mosnforte, Ramón, Vilar Samper, José y Martí-Bonmatí, Luis.** *Algoritmos en diagnóstico por la imagen.*
2. **Briz, Julián y Laso, Isidro.** *Internet y Comercio Electrónico.*
3. Escuela de Ingeniería de Sistemas y Computación. *Programación Extrema.* [En línea] [Citado el: 11 de Enero de 2009.] <http://eisc.univalle.edu.co/materias/WWW/material/lecturas/xp.pdf>.
4. Algoritmia.net. *Aplicaciones prácticas de los algoritmos.* [En línea] [Citado el: 23 de Octubre de 2008.] <http://www.algoritmia.net/articles.php?id=54>.
5. AstroMia. *Al-Khwarizmi, el álgebra y los algoritmos.* [En línea] [Citado el: 15 de Octubre de 2008.] <http://www.astromia.com/biografias/alkhwarizmi.htm>.
6. Centro de Especialización Profesional y Extensión Universitaria. *TÉCNICAS DE PROGRAMACIÓN.* [En línea] [Citado el: 20 de Octubre de 2008.] http://www.cepeu.edu.py/LIBROS_ELECTRONICOS_2/Los%20algoritmos.pdf.
7. Desarrolloweb.com. *Introducción a los algoritmos.* [En línea] [Citado el: 21 de Octubre de 2008.] <http://www.desarrolloweb.com/articulos/2143.php>.
8. Divide y vencerás. [En línea] [Citado el: 22 de Octubre de 2008.] <http://www.lcc.uma.es/~av/Libro/CAP3.pdf>.
9. Elaboración de circuitos impresos. [En línea] [Citado el: 05 de Enero de 2009.] <http://perso.wanadoo.es/jalons3/cursos/elabcomp.htm>.
10. learobotics. [En línea] [Citado el: 06 de Enero de 2009.] <http://www.learobotics.com/personal/juan/publicaciones/art4/hardware-libre.pdf>.
11. ISA-UMH Ingeniería de Sistemas y Automática. *MÉTODOS DE APRENDIZAJE INDUCTIVO.* [En línea] [Citado el 10 de Febrero de 2009.] http://isa.umh.es/asignaturas/aprendizaje/3_metodos.pdf.
12. Departamento de Matemática Aplicada I. *Geometría Computacional Tema 3: Problemas de proximidad.* [En línea] [Citado el 10 de Febrero de 2009.] <http://ma1.eii.us.es/Miembros/almar/docencia/gctem3ma.htm>.

13. Martin Fowler. *The New Methodology*. [En línea] [Citado el: 12 de Enero de 2009.] <http://martinfowler.com/articles/newMethodology.html>.
14. Desarrolloweb.com. *Qué es C#*. [En línea] [Citado el: 14 de Enero de 2009.] <http://www.desarrolloweb.com/articulos/561.php>.
15. **González Seco, José Antonio**. *El lenguaje de programación c#*.
16. Agapea. *Visual Studio.Net*. [En línea] [Citado el: 15 de Enero de 2009.] <http://www.agapea.com/libros/Visual-Studio-NET-isbn-8441513805-i.htm>.
17. **Teleformación**. *Conf#3 Flujo de trabajo de requerimientos*. [pdf] Curso 2007-2008.
18. Servicio de informática. *ALGORITMO de DIJKSTRA*. [En línea] [Citado el: 25 de Enero de 2009.] <http://www.alumnos.unican.es/uc900/Algoritmo.htm>.
19. Telematic services. *Algoritmo de Floyd*. [En línea] [Citado el: 26 de Enero de 2009.] <http://sertel.upc.es/redes/floyd.pdf>.
20. TRACER::UPC. *Algoritmos simples de Kruskal y Prim*. [En línea] [Citado el: 28 de Enero de 2009.] <http://tracer.lsi.upc.es/kp/SimpleAlgorithmsPage.htm>.
21. Alipi Style. *Algoritmos De Búsqueda*. [En línea] [Citado el: 27 de Enero de 2009.] <http://yobtrams.wordpress.com/2007/10/25/algoritmos-de-busqueda/>.
22. Departamento de Electrónica. *Algoritmos heurísticos*. [En línea] [Citado el: 29 de Enero de 2009.] <http://www.elo.utfsm.cl/~lsb/pascal/clases/cap25.pdf>.
23. SOCIEDAD ANDALUZA DE EDUCACIÓN MATEMÁTICA THALES. *El viajante de comercio mediante programación dinámica*. [En línea] [Citado el: 01 de Febrero de 2009.] http://thales.cica.es/rd/Recursos/rd99/ed99-0033-04/din_viajante.html.
24. Foros de la EUITIO. *TP:Viajante de comercio (II) - Backtracking*. [En línea] [Citado el: 03 de Febrero de 2009.] [http://euitio178.ccu.uniovi.es/wiki/index.php/TP:Viajante_de_comercio_\(II\)_-_Backtracking](http://euitio178.ccu.uniovi.es/wiki/index.php/TP:Viajante_de_comercio_(II)_-_Backtracking).
25. Foros de la EUITIO. *TP:Viajante de comercio - Algoritmos voraces*. [En línea] [Citado el: 03 de Febrero de 2009.] http://euitio178.ccu.uniovi.es/wiki/index.php/TP:Viajante_de_comercio_-_Algoritmos_voraces.
26. Willydev. *Métodologías ágiles para el desarrollo de software*. [En línea] [Citado el: 02 de Febrero de 2009.] <http://www.willydev.net/descargas/masyxp.pdf>.
27. Departamento de lenguajes y sistemas informáticos. *Tarjetas CRC*. [En línea] <http://lsi.ugr.es/~ig1/isoo/crc.pdf>.

BIBLIOGRAFÍA

28. XP. *A gentle introduction*. [En línea] [Citado el: 05 de Abril de 2009.]
<http://www.extremeprogramming.org>.
29. Departamento de lenguajes y sistemas informáticos. *PRUEBAS DEL SISTEMA EN PROGRAMACIÓN*. [En línea] [Citado el: 04 de Abril de 2009.]
http://www.lsi.us.es/~javierj/investigacion_ficheros/PSISEXTREMA.pdf.
30. Departamento de Ciencia de la Computación e Inteligencia Artificial. *Tema 2. Estimación de costes*. [En línea] [Citado el: 02 de Marzo de 2009.]
<http://www.dccia.ua.es/dccia/inf/asignaturas/GPS/archivos/Tema2.PDF>.
31. Organización y estructura de la información. *Primera etapa: Cálculo de los Puntos Función no ajustados*. [En línea] [Citado el: 02 de Marzo de 2009.]
<http://www.oei.eui.upm.es/Asignaturas/PInformaticos/ficheros/software/opcion1/pf1var1.html>.
32. Peiper. *Complejidad algorítmica*. [En línea] [Citado el: 15 de Febrero de 2009.]
<http://www.peiper.com.ar/edicion13/Complejidad.pdf>.
33. Esnips. Gerber RS-274X Format. [En línea] [Citado el: 10 de Enero de 2009.]
<http://www.esnips.com/doc/decfd35d-72bd-448b-a3f4-b3cc65ae494f/rs274xc>.
34. Reyes Chirino, Raymari; Díaz Rodríguez, Ariel. Evaluación de posibles elementos a considerar en la integración de los métodos de Boehm y Humphrey para la estimación de la duración de un proyecto de software para aplicaciones Multimedia”. Universidad de las Ciencias Informáticas. Ciudad Habana, 2008. 116 páginas.
35. **Teleformación**. *Conf#10 Algoritmos sobre Grafos I*. [pdf] Curso 2007-2008.
36. **Teleformación**. *Conf#11 Algoritmos sobre Grafos II*. [pdf] Curso 2007-2008.

REFERENCIAS BIBLIOGRÁFICAS

REFERENCIAS BIBLIOGRÁFICAS

1. Rodolfo Quispe- Otazu. *Qué es un algoritmo*. [En línea] [Citado el: 21 de 10 de 2008.] <http://www.rodolfoquispe.org/blog/que-es-un-algoritmo.php>.
2. **Helms, H.** *The McGraw-Hill Computer Handbook*. 1988.
3. Multipcb. *Fabricación de circuitos impresos - PCB - circuito impreso*. [En línea] [Citado el: 20 de 01 de 2009.] <http://www.multipcb.de/circuitos-impresos.html>.
4. Página de electrónica. *Circuito Impreso*. [En línea] [Citado el: 25 de 01 de 2009.] <http://electronica.webcindario.com/glosario/PCB.htm>.
5. **Grupo de Investigación y Desarrollo.** *DISEÑO Y FABRICACIÓN DE PLACAS DE CIRCUITO IMPRESO*. Ciudad Habana. 2000.
6. Facultad de Ingeniería Universidad de Buenos Aires. *El Problema del Viajante - Conceptos, Variaciones y Soluciones Alternativas*. [En línea] [Citado el 10 de Febrero de 2009.] <http://materias.fi.uba.ar/7114/Docs/ApunteViajante.pdf>.
7. **Beck, Kent.** *Extreme Programming Explained*. s.l. : Addison-Wesley, 2000.
8. **Beck, K. y Fowler, M.** *Planeando en programación extrema*. 2000.
9. **Ken Auer, Roy Miller.** *Extreme Programming Applied*. s.l. : Addison-Wesley Professional, 2001.
10. Escuela Superior de Ingeniería Informática. *TEMA 7: COCOMO II*. [En línea] [Citado el: 28 de Febrero de 2009.] http://trevinca.ei.uvigo.es/~cfajardo/Nueva_carpeta/presentaciones/cocomo2k.pdf.
11. Centros ITBA. *Estimación del esfuerzo basado en Casos de Uso*. [En línea] [Citado el: 28 de 02 de 2009.] 2- <http://www.centros.itba.edu.ar/capis/rtis/rtis-6-1/estimacion-del-esfuerzo-basada-en-casos-de-usos.pdf>.

GLOSARIO DE TÉRMINOS

1. **Disjuntos:** No tienen elementos en común.
2. **Fresar:** Cortar.
3. **Proteus, Orcad, Electronic Workbench, Circuit Maker:** Entornos integrados diseñados para la realización completa de proyectos de construcción de equipos electrónicos en todas sus etapas: diseño, simulación, depuración y construcción.
4. **Grafo dirigido:** Con aristas dirigidas.
5. **Grafo Ponderado:** Con pesos en las aristas.
6. **NP- hard :** NP: No Determinístico – difícil, es que no existe una solución en tiempo polinómico para el problema planteado.
7. **Ui :** Número de secuencia en la cual la ciudad i es visitada.
8. **Subtours:** Cuando se forman 2 caminos que debiera ser uno mismo.

Anexo 1 Mesa de coordenadas

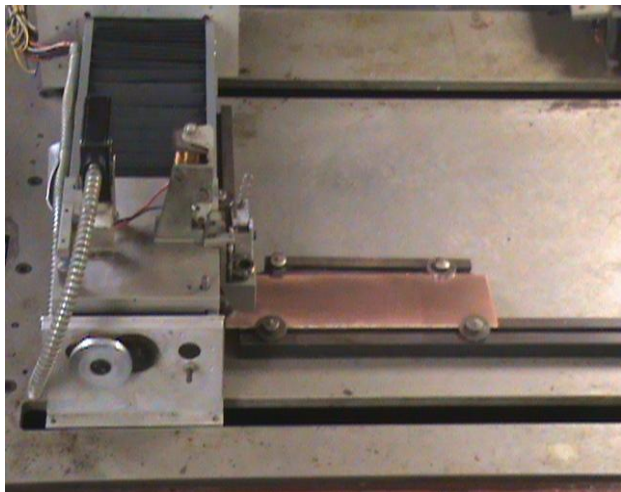


Figura 1 Mesa de coordenadas

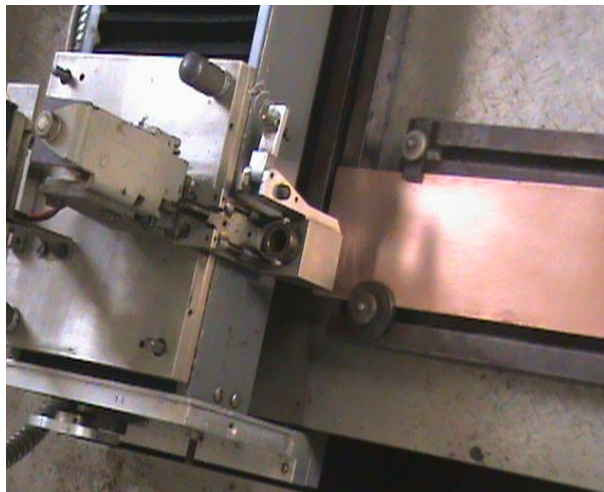


Figura 2 Mesa de coordenadas



Figura 3 Mesa de coordenadas

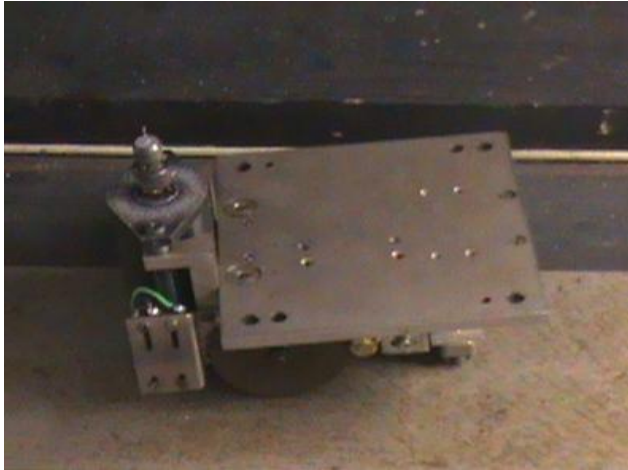


Figura 4 Herramienta para el barrenado o fresado



Figura 5 Cabezal portaplumas

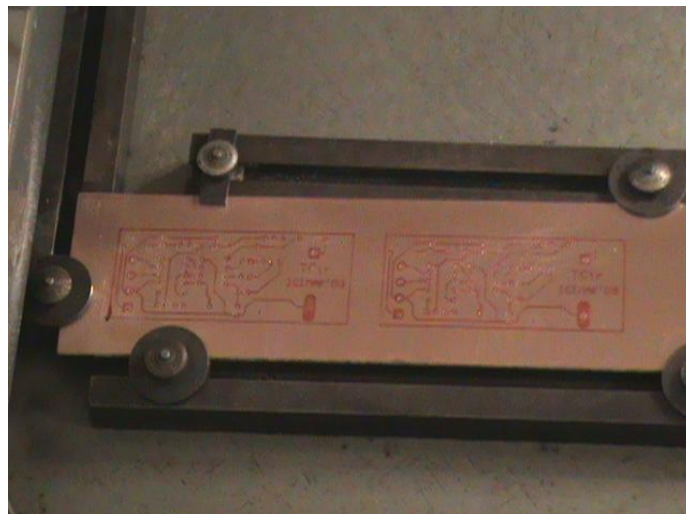


Figura 6 Proceso de pintar el circuito impreso antes y después de optimizado

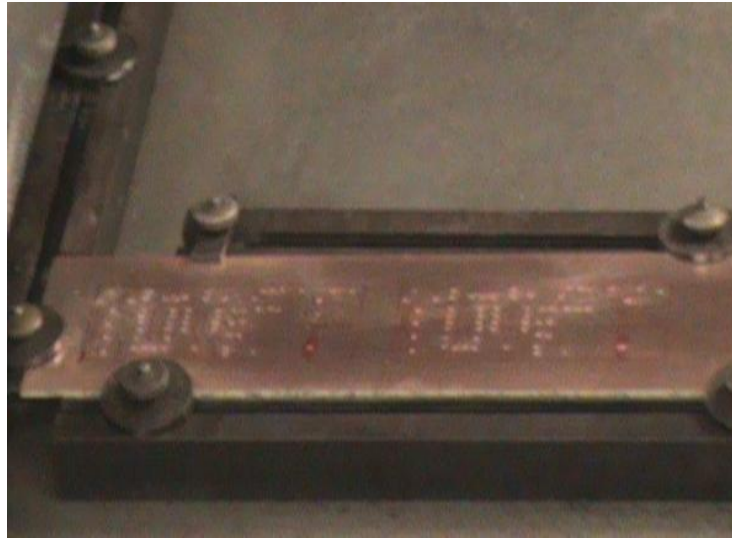


Figura 7 Proceso de barrenar el circuito impreso antes y después de optimizado

Anexo 2 Circuito impreso

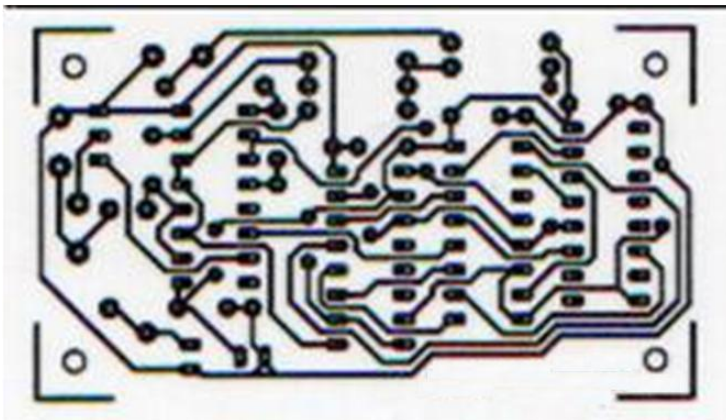
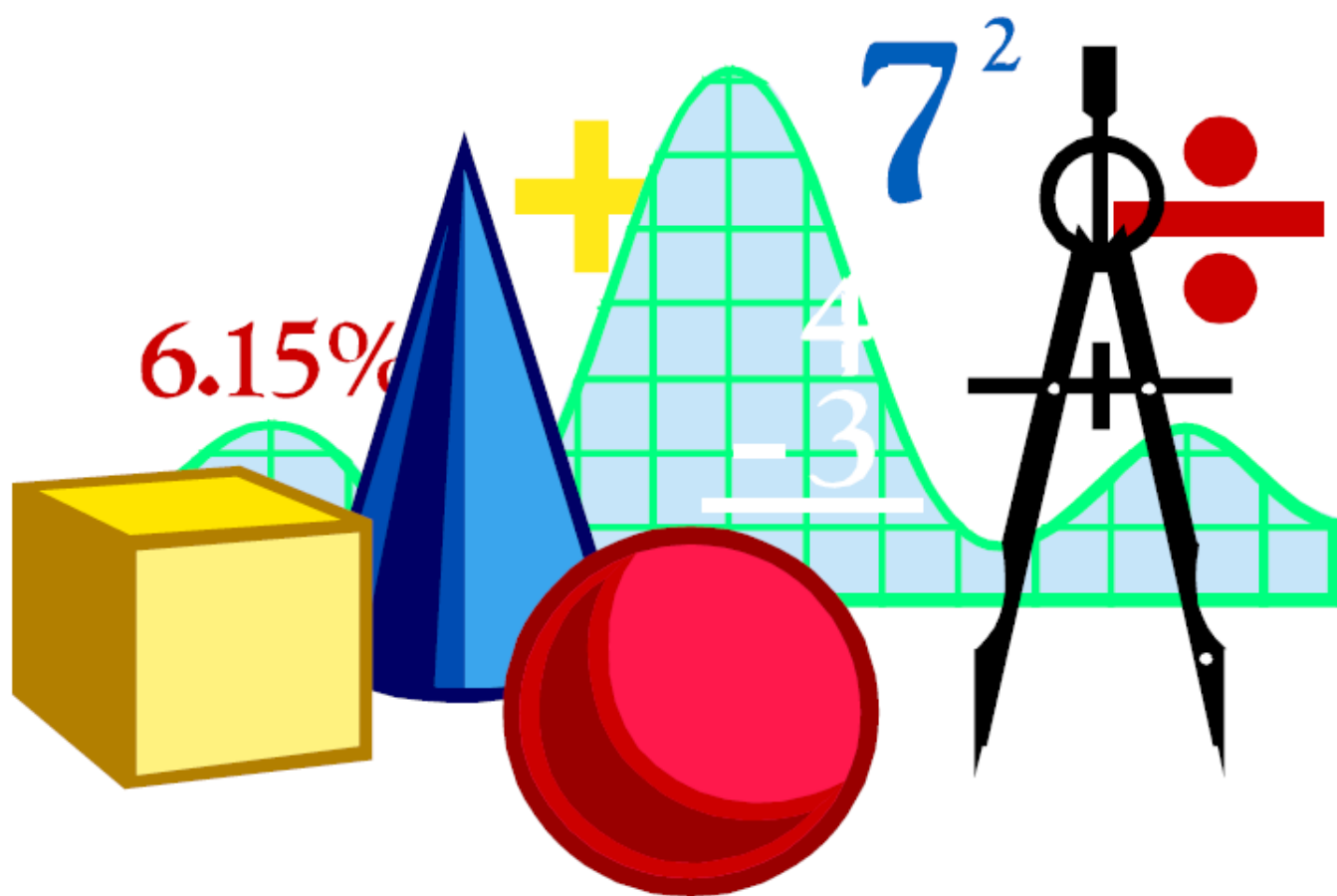


Figura 8 Circuito impreso

Anexo 3. Manual Gerber

Gerber RS-274X Format



User's Guide

© Copyright 1998 Barco Graphics, Gent, Belgium, and Barco Gerber Systems, South Windsor, CT, USA.

All rights reserved. This material, information and instructions for use contained herein are the property of Barco Graphics N.V. and Barco Gerber Systems. The material, information and instructions are provided on an AS IS basis without warranty of any kind. There are no warranties granted or extended by this document. Furthermore Barco Graphics N.V. and Barco Gerber Systems do not warrant, guarantee or make any representations regarding the use, or the results of the use of the hardware, software or the information contained herein. Neither Barco Graphics nor Barco Gerber Systems shall be liable for any direct, indirect, consequential or incidental damages arising out of the use or inability to use the software or the information contained herein.

The information contained herein is subject to change without notice. Revisions may be issued from time to time to advise of such changes and/or additions.

No part of this document may be reproduced, stored in a data base or retrieval system, or published, in any form or in any way, electronically, mechanically, by print, photoprint, microfilm or any other means without prior written permission from Barco Graphics N.V. or Barco Gerber Systems.

This document supersedes all previous dated versions.

All product names are trademarks or registered trademarks of their respective owners. Correspondence regarding this publication should be forwarded to:

Engineering Services
Barco Gerber Systems Corporation
30 South Satellite Road
South Windsor, CT 06074

Printed in the USA

Published by
Barco Gerber Systems Corporation
Engineering Services Department

RS-274X Format User's Guide
Part Number 414-100-014 Rev C
September 21, 1998

Preface

The information in this Guide was published previously as the *Gerber Format Guide*, which documented *RS-274X format*, also called *Gerber Format*, for both vector and raster devices. Acknowledging that vector plotting is rapidly becoming an outdated technology, this Guide describes use of RS-274X only in raster applications and eliminates codes that pertain to vector applications. This does **not** imply that existing files that include codes and parameters not described in this edition of the Guide will not work. The intent of this Guide is to describe and document uses of RS-274X format.

Contents

INTRODUCTION	1
Who should use this Guide?	1
How to use this Guide	2
Related documentation	2
Additional copies of the Guide	2
Where to go for help	2
RULES AND GUIDELINES.....	3
File structure	3
Data blocks	3
Layers	3
Data types	4
RS-274X parameters	4
Directive parameters	5
Image parameters	5
Aperture parameters	5
Layer-specific parameters	6
Miscellaneous parameters	6
Standard RS-274D Codes and Coordinate Data	7
General File Preparation Guidelines	8
RS-274X Parameter Guidelines	8
Table 1 RS-274X Parameter Order of Entry	9
RS-274D Data Guidelines	10
Table 2 RS-274D Code Order of Entry	10
Sample Files	10
Example 1	10
Example 2	11
REFERENCE.....	15
RS-274X parameters	15
AD - Aperture Definition	16
AM - Aperture Macro	19
AS - Axis Select	25
FS - Format Statement	26
IF - Include File	28
IJ - Image Justify	29
IN - Image Name	30
IO - Image Offset	31
IP - Image Polarity	32
IR - Image Rotate	33
KO - Knockout	34
LN - Layer Name	36
LP - Layer Polarity	37
MI - Mirror Image	38
MO - Mode	39
OF - Offset	40
PF - Plot Film	41

SF - Scale Factor.....	42
SR - Step and Repeat	43
STANDARD RS-274D CODES.....	45
D Codes.....	45
Table 4 D Codes	45
G Codes.....	46
Table 5 G Codes	46
Linear Interpolation (G01, G10, G11, G12)	47
Circular Interpolation (G02, G03, G74, G75)	47
Multiquadrant (360°) Circular Interpolation (G74, G75)	48
Polygon Area Fill (G36, G37)	49
M Codes.....	50
GLOSSARY.....	51
INDEX.....	53

Introduction

Gerber data is a simple, generic means of transferring printed circuit board information to a wide variety of devices that convert the electronic PCB data to artwork produced by a photoplotter. Virtually every PCB CAD system generates Gerber data because all photoplotters read it. It is a software structure consisting of X,Y coordinates supplemented by commands that define where the PCB image starts, what shape it will take, and where it ends. In addition to the coordinates, Gerber data contains aperture information, which defines the shapes and sizes of lines, holes, and other features.

Gerber Format, which is the format in which Gerber data is expressed, actually is a family of data formats that are subsets of EIA Standard RS-274D. *Extended Gerber Format*, which is also called *RS-274X*, provides enhancements that handle polygon fill codes, positive/negative image compositing, and custom apertures, and other features. RS-274X also encapsulates the aperture list in the header of the Gerber data file and therefore allows files to pass from one system to another without the need to re-input the aperture table. RS-274X produces a variety of Gerber data called *X data*.

RS-274X is a *superset* of the EIA Standard RS-274D format. RS-274X supports some of the parameter data codes (G codes) and aperture codes (D codes) contained in RS-274D, as well as codes referred to as *mass parameters*. Mass parameters are plot parameters that define characteristics that can affect an entire plot, or only specific parts of the plot, called *layers*. Mass parameters extend the capabilities of Gerber Format. Their presence makes the Gerber data *X data*.

RS-274X is maintained by Gerber Systems Corporation (GS), a leading supplier of CAD/CAM systems, large-area plotting systems, and precision cutting systems since 1965.

Who should use this Guide?

In order to use this Guide, you should have a fundamental understanding of PCB fabrication or PCB design and laser plotting concepts. This Guide is intended for use by:

- PCB designers preparing data for conversion to RS-274X
- PCB fabricators creating or using Gerber data files
- Developers of software applications using RS-274X data

How to use this Guide

You will find the following sections in this Guide:

Rules and Guidelines explain file content and structure and outlines rules and guidelines for creating a correct RS-274X file. It also contains a sample file.

Reference defines use and constraints on use of every RS-274X parameter and data code currently supported. Parameters and data codes are described separately. Both are presented in alphanumerical order.

You will also find a **Glossary** and **Index** at the end of the Guide.

Related documentation

This Guide assumes you are familiar with Electronic Industries Association EIA Standard RS-274D. You can obtain a copy of this standard from the Electronic Industries Association, Engineering Department, 2001 Eye Street NW, Washington, DC 20006 USA.

Where to go for help



Should you need assistance, contact the Baraco Gerber Systems Corporation Technical Assistance Center by telephone 8 a.m. to 5 p.m. (eastern time) at (860) 291-7016, by fax at (860) 291-7021.

Rules and Guidelines

This section provides background information, describes organization, and presents guidelines for use of RS-274X. For detailed descriptions of use of individual codes and parameters, see *Reference*, page 15.

File structure

An *RS-274X plot file* is a file consisting of RS-274X parameters and standard RS-274D codes which, when correctly interpreted, result in an image that may be displayed or plotted.

Data blocks

The file is composed of a number of *data blocks* containing parameters and codes. Each data block is delimited by an end-of-block character, typically an asterisk (*).

Each data block may contain one or more parameters or codes. For example:

```
X0Y0D02*  
X50000Y0D01*
```

Layers

One or more data blocks may be grouped into a *layer* of information that describes part of a graphic image. In RS-274X context, a layer is a *named information component of the image* composed of one or more data blocks. Each layer may have characteristics, such as name, polarity, and interpolation mode, that differ from other layers of information. In addition, an individual layer may be “knocked out” of the surrounding graphic image, and may be repeated and/or rotated individually.

Note: A layer must not be confused with a PCB layer. A PCB layer has a physical definition and might be compared to a physical plane. An RS-274X layer is simply a group of data blocks that may be manipulated collectively and separately from other layers.

Data types

An RS-274X file may contain the following kinds of data appearing in the following general order:

1. RS-274X Parameters

RS-274X parameters are also called *mass parameters* or *extended Gerber format*. The inclusion of these parameters in the file makes the plot file RS-274X, or *X data*, instead of standard RS-274D.

2. Standard RS-274D Codes

Standard RS-274D codes were once called *word address format*. They consist of:

- one-character **function codes** such as G codes, D codes, M codes, etc. Function codes were the *words* of the old terminology. They describe how coordinate data associated with them should be interpreted (such as linear or circular interpolation), how the imaging device should move (light source on or off), and more.
- **coordinate data** define points to which the imaging device must move. The coordinate data represented the *address* of the old terminology. X,Y coordinate data describe linear positions. I, J coordinates define arcs.

RS-274X parameters

RS-274X parameters define characteristics that apply to an entire plot or to a single layer, depending on the parameter's position in the file and whether it generates a new information layer in the file (as, for example, layer-specific parameters do). RS-274X parameters consist of two alpha characters followed by one or more optional modifiers.

RS-274X parameters are delimited by a *parameter delimiter*, typically a percent (%) sign. Because parameters are also contained in a data block, they are also delimited by an end-of-block character. For example,

```
%FSLAX23Y23*%
```

This parameter is a Format Statement (FS) describing how the coordinate data in the file should be interpreted (in this case, 4.2 format for both X and Y coordinates). It is delimited by an end-of block character (*) as well as by a parameter delimiter (%).

RS-274X parameters may be grouped according to the scope of their function in the file. The groups should appear in the file in the following order:

1. **Directive parameters** control overall file processing.
2. **Image parameters** supply information about an entire image.
3. **Aperture parameters** describe the shape of lines and components throughout the file.
4. **Layer-specific parameters** describe processing of one or more data layers.
5. **Miscellaneous parameters** provide capabilities that do not fall into the above groups.

RS-274X parameters are generally placed at the beginning of the file in the order shown above. Certain parameters, such as the layer-specific parameters, may be embedded within the file.

The next sections describe each parameter type.

Directive parameters

Directive parameters control overall file processing. They include:

AS	Axis Select
FS	Format Statement
MI	Mirror Image
MO	Mode of units
OF	Offset
SF	Scale Factor

As a general rule, directive parameters should be placed at the beginning of the file. For simplicity sake, you should use each directive parameter only once in a file, although it is not illegal to use a directive parameter more than once. Each directive parameter controls processing until another like it is encountered.

When used more than once in a file, subsequent directive parameters may be embedded anywhere in the standard RS-274D code data or grouped with other layer-specific parameters.

Directive parameters do not generate a new layer.

Image parameters

Image parameters supply information about the entire (composite) image. Image parameters include:

IJ	Image Justify
IN	Image Name
IO	Image Offset
IP	Image Polarity
IR	Image Rotation
PF	Plotter Film

Image parameters should be used only once in a file and should be placed at the beginning of the file. If an image parameter occurs more than once in a file, the last one encountered will be the operative parameter.

Aperture parameters

Vector plotters control the width and shape of features by projecting light through a series of openings, or apertures, in a rotating wheel. Each position on the wheel is identified by a unique D code. When the D code appears in the data, the wheel rotates to the referenced position for exposure.

Unlike a vector device, a raster device has no apertures and therefore requires a description of the aperture geometry to create the required lines and other features. The aperture parameters provide the description.

The aperture parameters include:

AD	Aperture Description
AM	Aperture Macro

In general, aperture parameters apply to an entire file. An exception is an embedded AD parameter, which will generate a new layer if it redefines a D code previously used in the image data.

Note: Generating a new layer may result in unanticipated results because it causes certain RS-274D values (such as interpolation mode) to be reset.

The AM parameter describes a special aperture by using the following set of predefined aperture shapes to describe an aperture:

- Circle
- Line (vector)
- Line (center)
- Line (lower left)
- Outline
- Polygon
- Moiré
- Thermal

See the AM command description, page 19, for more information.

Layer-specific parameters

Layer-specific parameters supply information for the processing of one or more *information* layers (not to be confused with board layers). They may be used more than once in a file. Layer-specific parameters always generate a new layer and should be placed at the beginning of the new layer. If not repeated for a given layer, the previous layer-specific parameters remain in effect.

The layer-specific parameters include:

KO	Knockout
LN	Layer Name
LP	Layer Polarity
SR	Step and Repeat

Note: Generating a new layer may result in unanticipated results because it causes certain RS-274D values (such as interpolation mode) to be reset.

Miscellaneous parameters

There is a single miscellaneous parameter:

IF	Include File
----	--------------

The IF parameter is used to include (nest) external files in a file.

Standard RS-274D Codes and Coordinate Data

Standard RS-274D codes (D codes, G codes, M codes, etc.) specify how the coordinate data should be manipulated. Each code applies to coordinate data located in the same data block as the code (that is, between EOB characters). It also applies to coordinate data following it until another code of the same type is encountered, or until a new layer is generated. This continuing action is referred to as *modal*.

For example, G02 specifies clockwise, single-quadrant circular interpolation and is modal. All coordinate data following it will be considered clockwise arc data until another interpolation code is encountered, or until a new layer is generated. When a new layer is generated, interpolation will be reset to linear (G01).

Like parameters, standard RS-274D codes may be grouped according to function. They generally appear in the file in the following sequence:

1. **N codes** (sequence numbers) are similar to line numbers and may be assigned to data blocks to simplify organization. Sequence numbers may be 0 to 99999. N codes are not necessary.
2. **G codes** (general functions) specify how to interpolate and move to the coordinate locations following the code until changed or until a new layer is generated (modal).
3. **D codes** (plot functions) select and control tools, specify line type, etc.
4. **M codes** (miscellaneous functions) perform a variety of functions such as program stop and origin specification.

Standard RS-274D codes are described in detail starting on page 45.

Coordinate Data

Coordinate data includes:

- **X,Y data** define linear positions along the X and Y axes.
- **I,J data** define arcs.

For example,

X200Y200D02*

This data block directs the plotter to move in a positive direction to coordinate location 0.2,0.2 (assuming leading zeroes are omitted) with the light source off (tool up). Additional X,Y coordinate data positions following this code will also cause motion with the light source off until a different code is encountered.

Absolute versus relative data

Depending on the preceding FS parameter in the file, coordinate data may be defined either relative to the plot origin (that is, as absolute values) or relative to the last coordinate position (that is, as incremental data).

Numerical precision

Coordinate data may be expressed in inches or millimeters to ± 6.6 decimal places (that is, up to six integer digits and six fractional digits). Unless preceded by a “-”, the direction is assumed to be “+”.

Axis assignment

The coordinate axes may be assigned to any physical plotter axes using the AS parameter, but typically the A plotter axis is assigned to X and the B axis to Y.

General File Preparation Guidelines

Follow these guidelines when preparing RS-274X data:

- Use data blocks in a way that organizes the file visually for easy reading.
- Enter all codes and parameters in upper case.
- Use file names that comply with the system file naming conventions. DOS and therefore Windows 3.1 files names are limited to eight characters. UNIX systems are case-sensitive.
- End every data block with an end-of-block character, typically *. For example,
X0000Y5000D02*
- Do not break a line within a block.
- End every file with an end-of-program code (M00 or M02).

RS-274X Parameter Guidelines

- Begin and end RS-274X parameter data with a parameter delimiter, typically %. The parameter delimiter must immediately follow the end-of-block without intervening spaces. For example,
%ASAXBY*%
- Parameters may be entered singly or grouped between delimiters, up to a maximum of 4096 characters between delimiters. A maximum of 80 characters between delimiters is recommended. Always consider readability. For example,
%SFA1.0B1.0*ASAXBY*%
- Line breaks are permitted between parameters to improve readability. For example,
%SFA1.0B1.0*
ASAXBY*%
- Use an explicit decimal point with all numerical values associated with a parameter. If the decimal point is omitted, an integer value is assumed.
- Express numerical values in the units defined by the MO code in the file (inches or millimeters).

- In general, enter RS-274X parameters at the beginning of the file (or at the beginning of the layer for layer-specific parameters). Enter them in the order shown in Table 1.

Note: When RS-274X parameters are embedded in the RS-274D data, all data preceding the parameter will be processed before the data blocks affected by the embedded parameters are interpreted.

Table 1 RS-274X Parameter Order of Entry

Parameter		Function	Comments	Default
Required	Optional			
	AS	Axis select	Single use recommended. When used more than once, enter these parameters at the beginning of a layer. These codes do not generate a new layer.	A=X, B=Y
FS		Format statement		
	MI	Mirror image		No mirror
	MO	Mode (inch or millimeter units)		Inch
	OF	Offset		A=0, B=0
	SF	Scale Factor		A=1.0, B=1.0
	IJ	Image Justify	Use only once at the beginning of the file.	No justification
IN		Image Name		
	IO	Image Offset		A=0, B=0
	IP	Image Polarity		Positive
	IR	Image Rotation		0
	PF	Plotter Film		
AD		Aperture Description	May be used singly or may be layer-specific. Enter these parameters at the beginning of the file or layer.	
	AM	Aperture Macro		
	LN	Layer Name		
	LP	Layer Polarity		Positive
	KO	Knockout		Off
	SR	Step and Repeat		A=1, B=1
	RO	Rotate		No rotation

RS-274D Data Guidelines

Follow these guidelines when preparing RS-274D data:

- Enter functions codes and coordinate data following the RS-274X parameters.
- Function codes apply to coordinate data in the same block as well as to subsequent coordinate data. They do not affect coordinate data preceding the block in which they occur.
- Enter function codes in the file in the order shown in Table 2.

Table 2 RS-274D Code Order of Entry

Code	Function	Comments
N	Sequence number	Optional
G	General functions: linear interpolation, circular interpolation, polygon area fill, etc.	Once encountered, remains in effect until countermanded.
D	Aperture or tool assignment; line/flash control	Once encountered, remains in effect until countermanded.
M	Miscellaneous function: program stop or end.	Every file must end with M00 or M02.

Sample Files

The examples on these pages illustrate the use of both mass parameters and standard RS-274D codes.

Example 1

Example 1 illustrates a single layer image.

*G04 EXAMPLE 1: 2 BOXES

%FSLAX23Y23*%

Format statement - leading zeroes omitted,
absolute coordinates, X2.3, Y2.3.

%MOIN*%

Set units to inches.

%OFA0B0*%

No offset

%SFA1.0B1.0*%

Scale factor is A1, B1

%ADD10C,0.010*%

Define aperture D-code 10 - 10 mil circle

%LNBOXES*%

Name layer "BOXES".

G54D10

X0Y0D02*X5000Y0D01*

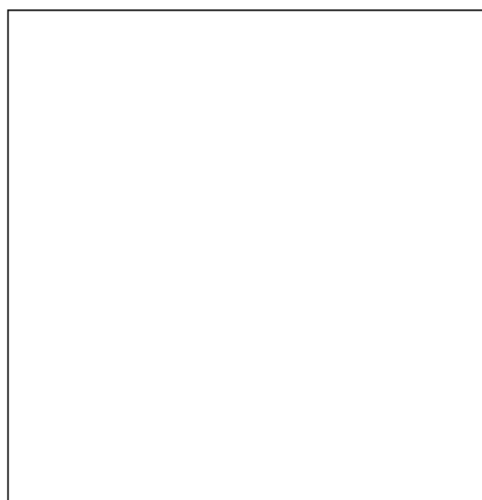
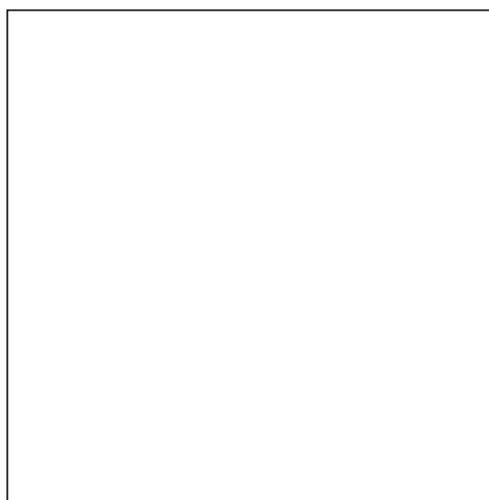
X5000Y5000D01*X0Y5000D01*X0Y0D01*

RS-274D data

X6000Y0*X11000Y0D01*

X11000Y5000D01*X6000Y5000D01*
 X6000Y0D01*D02*
 M02*

End of data



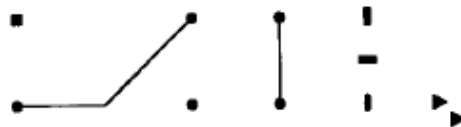
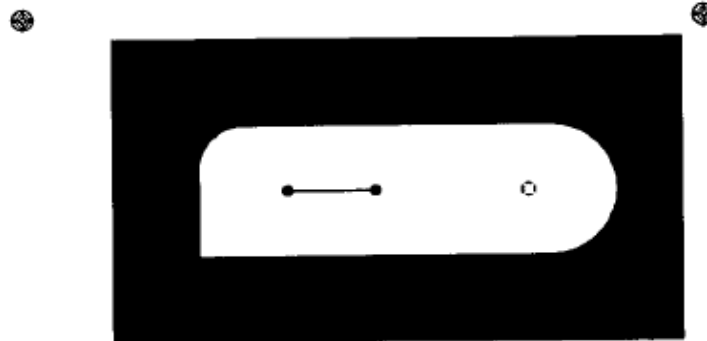
Example 2

Example 2 illustrates RS-274-X data.

%ASAXBY*	Axis Select, A=X, B=Y
FSLAX23Y23*	Format Statement, Leading zeros omitted, absolute data, 2 integer digits and 3 fractional digits
MIA0B0*	Mirror about the specified axis; 0=no, 1=yes
MOIN*	Mode inches
OFA0B0*	Offset 0
SFA1.0B1.0*%	Scale Factor
%IJALBL*	Image justify
INXTEST*	Image name
IOA0B0*	Image offset
IPPOS*	Image Polarity
IR0*%	Image Rotation
G04 Define Apertures*	Comment
%AMTARGET125*	Aperture Macro
6,0,0,0.125,.01,0.01,3,0.003,0.150,0*%	Moiré Description
%AMTHERMAL80*	Aperture Macro
7,0,0,0.080,0.055,0.0125,45*%	Thermal Description
%ADD10C,0.01*	Aperture Description, D10 is a circular aperture with 0.01" diameter
ADD11C,0.06*	Aperture Description, D11 is a circular aperture with 0.06" diameter
ADD12R,0.06X0.06*	Aperture Description, D12 is a rectangular aperture, 0.06" X 0.06"
ADD13R,0.04X0.100*	Aperture Description, D13 is a rectangular aperture, 0.04" X 0.100"
ADD14R,0.100X0.04*	Aperture Description, D14 is a rectangular aperture, 0.100" X 0.04"
ADD15O,0.04X0.100*	Aperture Description, D15 is a obround aperture, 0.04" X 0.100"
ADD16P,0.100X3*	Aperture Description, D16 is a 3 sided polygon 0.100" overall size
ADD17P,0.100X3*	Aperture Description, D17 is a 3 sided polygon 0.100" overall size
ADD18TARGET125*	Aperture Description, D18 is a special aperture called "TARGET"

ADD19THERMAL80*%	Aperture Description, D19 is a special aperture called "THERMAL"
%LNXTTEST1*	Layer Name XTEST1
LPD*	Layer Polarity Dark
SRX1Y1I0J0**%	Step and Repeat set to 1 X 1 (Not Required)
G54D10*	Aperture select
G01X0Y250D02*	Linear move with light off
X0Y0D01*	Linear move with light on
X250Y0D01*	Linear move with light on
X1000Y1000D02*	Linear move with light off
X1500D01*	Linear move with light on
X2000Y1500*	Notice since D01 is modal it does not need to be repeated
X2500D02*	Notice since the X & Y commands are modal, Y is not repeated
Y1000D01*	Here, X is not repeated and uses its previous value of 2.500"
D02*	Light off no move
G54D11*	New aperture selected
G55X1000Y1000D03*	G55 prepares for flash It is not necessary. D03 is the flash command.
X2000D03*	Y value does not change
X2500D03*	This method reduces the size of the file
Y1500D03*	Here, X does not change from previous value
X2000D03*	Flash
G54D12*	New aperture select
X1000Y1500D03*	Move to (1.0, 1.5) and flash
G54D13*	New aperture select
X3000Y1500D03*	Move and flash
G54D14*	New aperture select
Y1250D03*	Move and flash
G54D15*	New aperture select
Y1000D03*	Move and flash
G54D10*	New aperture select
G01X3750Y1000D02*	Linear move, light off. Start point of the following arc command
G75*	Sets the mode to 360 degree circular interpolation
G03X3750Y1000I250J0D01*	Move from start point above to end point drawing a complete circle
G54D16*	New aperture select
G55X3400Y1000D03*	Flash
G54D17*	New aperture select
G55X3500Y900D03*	Flash
G54D10*	New aperture select
G36*	Start Polygon fill
G01X500Y2000D02*	
Y3750D01*	
X3750*	
Y2000*	
X500*	
X500Y2000D02*	
G37*	End Polygon fill
G54D18*	New aperture select
G55X0Y3875D03*	Flash
X3875Y3875D03*	Flash
%LNXTTEST2*	Layer Name
LPC**%	Layer Polarity clear
G36*	Start Polygon fill
G01X1000Y2500D02*	
Y3000D01*	
G74*	Quadrant arc
G02X1250Y3250I250J0D01*	Clockwise arc move with radius .25"
G01X3000*	Complete 90 degree arc
G75*	360 degree arc mode

G02X3000Y2500I0J-375D01*	Clockwise arc move with radius .375"
G01X1000*	Linear move light on
X1000Y2500D02*	Linear move light off
G37*	End Polygon fill
%LNXTEST3*	Layer Name
LPD*%	Layer Polarity Dark
G54D10*	New aperture select
X1500Y2875D02*	
X2000D01*	
D02*	
G54D11*	
X1500Y2875D03*	
X2000D03*	
G54D19*	New aperture select
X2875Y2875D03*	
M02*	End of file



Reference

RS-274X parameters

This section describes every RS-274X parameter supported at time of publication. They are arranged in alphabetical order. Standard RS-274D code descriptions begin on page 45.

Each parameter description illustrates the parameter data block format, explains each parameter modifier, lists restrictions, and gives an example.

The data block format illustration uses the following notation conventions:

%Parameter code <required modifiers>[optional modifiers]*%

where:

Parameter code	is the 2-character code (AD, AM, FS, etc.)
<required modifiers>	must be entered to complete definition
[optional modifiers]	may be required depending on the required modifiers

AD Aperture Definition

The AD parameter is used to describe apertures (D codes) used in the RS-274X file. All apertures used in an RS-274X file must be described in terms of shape and size for the file to be interpreted correctly. The AD parameter must precede use of the associated aperture D-code. A definition remains in effect until redefined.

Two kinds of apertures may be used in an RS-274X file: *standard* apertures and *special* apertures.

Standard apertures

The AD parameter identifies standard apertures by D-code number and describes them in terms of shape (circular, rectangular, obround, or polygonal) and size (diameter if round, height and width if rectangular or obround, outside dimension and number of sides if polygonal). Apertures may be solid or open (that is, with a hole) and are always centered.

Special apertures

The AD parameter is also used to assign a D-code to a file containing an aperture description. The aperture description file may be a .mac file created by the AM (Aperture Macro) parameter or a .des file created by an Aperture Editor such as the Gerber GPC Aperture Editor. See the AM parameter description for further information on aperture macros.

AD parameter syntax rules

- Like other mass parameters, begin and end each parameter block with a parameter delimiter (typically %).
- Within the AD parameter block, separate each modifier by an X.
- Dimensions must be positive and will be rounded to the resolution of the output device.
- The various plotters and output devices may permit different D-code ranges, but the range must not exceed 10 to 999.

Data Block Format

```
%ADD<D-code number><aperture type>,<modifier>[X<modifier>]*%
```

where:

ADD

<D-code number>

<aperture type>,<modifier>[X<modifier>]

the AD parameter and D (for D-code)

the D-code number being defined (10 - 999)

the aperture descriptions. **<aperture type>** may be one of the following:

- a standard aperture (C, R, O, P, or T) with modifiers (see below). Modifiers depend on aperture type. Use **X** to separate each modifier. All dimensions are positive and will be rounded to the resolution of the output device.

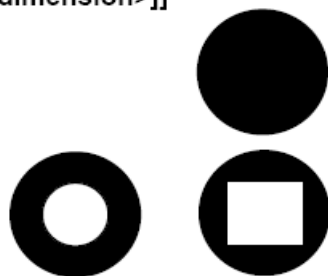
AD
Aperture Definition

- a file name containing the aperture description (.des file)
- an aperture macro name previously defined by the AM parameter (.mac file)

Note: Be sure to use the units (inches or millimeters) specified by the MO parameter for all modifiers.

Standard apertures:

C, <outside diameter>[X<X-axis hole dimension>[X<Y-axis hole dimension>]]



Circle. To define a solid aperture, enter only the diameter. To define a hole, enter one dimension for a round hole, two for a rectangle. The hole must fit within the aperture. For a square hole, X must equal Y. Both aperture and hole will be centered. For example,

`%ADD10C,.05X0.025*%`

D-code 10 is a .05 circle with a .025 round hole in the center.

R, <X-axis dimension>X<Y-axis dimension>[X<X-axis hole dimension>X<Y-axis hole dimension>]



Rectangle or square. May be solid or open. If the X axis dimension equals the Y dimension, the aperture will be square. To define a solid aperture, enter only the X and Y dimensions; omit the hole dimensions. To define a hole, enter one dimension for a round hole, two for a rectangle. The hole must fit within the aperture. Both rectangle and hole will be centered. For example,

`%ADD22R,0.020X0.040*%`

D-code 22 is a .02 x .04 solid rectangle.

O, <X-axis dimension>X<Y-axis dimension>[X<X-axis hole dimension>[X<Y-axis hole dimension>]]



Obround (oval). May be solid or open. If the X dimension is larger than Y, the shape will be horizontal. If the X dimension is smaller than Y, the shape will be vertical. To define a solid aperture, enter only the X and Y dimensions; omit the hole dimensions. To define a hole, enter one hole dimension for a round hole, two for a rectangular or square hole. If open, the hole must fit within the aperture. For example,

`%ADD22O,0.020X0.04X0.005X0.010*%`

D-code 22 is a vertical obround .02 wide x .04 high with a .05 x .01 rectangular hole.

AD
Aperture Definition

P, <outside dimension>X<number of sides>[X<degrees of rotation>[X<X-axis hole dimension>X<Y-axis hole dimension>]]



Regular polygon. May be solid or open. To define a solid aperture, enter only the outside dimension and number of sides (3 to 12). The first point is located on the X axis. May be rotated $\pm 360^\circ$ from the X-axis. If open, the hole must fit within the outside dimension. *Note: If you use the hole dimension modifiers, you must enter a rotation (even if it is 0).* For example,

`%ADD17Diamond,.030X4X0.0*%`

D-code 17 is a polygon within an outside dimension of .03, 4 sides, with no center hole.

Examples

`%ADD10C,.025*%`

Define D-code 10: 25 mil round

`%ADD22R,.050X.050X.027*%`

Define D-code 22: 50 mil square with 27 mil round hole

`%ADD57O,.030X.040X.015*%`

Define D-code 57: obround 30 x 40 mil with 15 mil round hole

`%ADD30P,.016X6*%`

Define D-code 30: polygon (hexagon), 16 mil outside dimension with 6 sides

`%ADD15CIRC*%`

Define D-code 15: a special aperture described by aperture macro CIRC defined previously by an aperture macro

AM Aperture Macro

The AM parameter is used to define named apertures (sometimes called *special apertures*) in *aperture macro* format consisting of building blocks called *primitives*. The named aperture macros may be used in AD parameter descriptions just like the standard apertures (that is, circle, rectangle, obround, polygon, and thermal). Every non-standard aperture must be described before the D-code associated with it occurs in the file.

Special apertures offer two advantages over standard apertures:

- They allow multiple shapes called primitives to be combined in a single aperture, which permits creation of unusual or complicated apertures.
- They need not be centered.
- Aperture macro modifiers may be variable. Variable modifiers are supplied by the AD parameter that references the aperture macro.
- An aperture macro variable may be a numerical function of another macro variable (+, -, etc.).

Aperture macro contents

An aperture macro contains the following elements:

- aperture macro name
- one or more of the seven aperture primitives, each identified by a primitive number (see Table 3 below for a description of the primitives)
- primitive modifiers that describe the primitive in terms of exposure, position, dimensions, etc.
- variable primitive modifiers to be supplied by the AD parameter
- optional embedded comment blocks
- numerical operators

AM parameter syntax rules

- Like other mass parameters, begin and end each parameter block with a parameter delimiter (typically %).
- Within the AM parameter block, separate each primitive and modifier group by an end-of-block character (typically *).
- Within each primitive group, separate modifiers by commas.
- Modifiers may be absolute values, such as 0, 1, 2, or 9.05, or they may be variable modifiers to be supplied by the AD parameter when it refers to the aperture macro.
- Identify variable modifiers to be supplied by the AD parameter as \$*n* where *n* indicates the order in which the modifier is expected in the AD parameter. \$1

AM Aperture Macro

would be the first variable modifier expected in the AD parameter, \$2 the second, and so on, numbering sequentially from left to right. If an absolute value is entered instead of a variable, the variables shift right. For example, if an absolute value is entered for the first variable, the next variable becomes \$1 even though it is the second modifier of the primitive.

- The interpretation of each modifier differs for each primitive. See Table 3 on the next page for a full explanation of aperture macro primitives and modifiers.
- Do not begin a variable primitive modifier with a minus sign (for example, -\$1). To indicate negative, precede the variable with 0 (for example, **0-\$1**).
- Start optional comment strings with a leading 0 (for example, ***0 THIS IS A COMMENT***).
- Position and dimensions are expressed in the units specified by the MO parameter. Decimal points are permitted.
- Use only the following numerical operators with variable modifiers:

Operator	Function
+	add
-	subtract
/	divide
x	multiply
=	equate
<i>n</i>	numerical factor

- Make sure the aperture macro file name matches the aperture macro name and that it has a .mac extension.

Data Block Format

```
%AM<aperture macro name>* <primitive number>,<modifier$1>,<modifier$2>,<...>]* [<primitive number> [<modifiers>]]* ...*%
```

where:

AM

the AM parameter

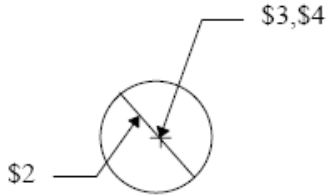
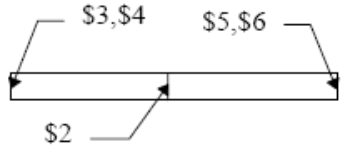
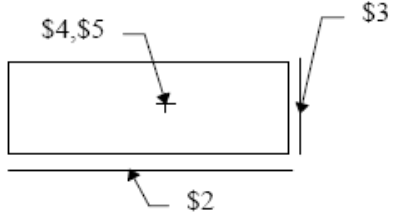
<aperture macro name>*

the name to be used in the AD parameter

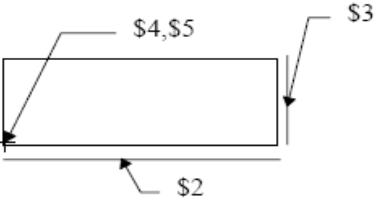
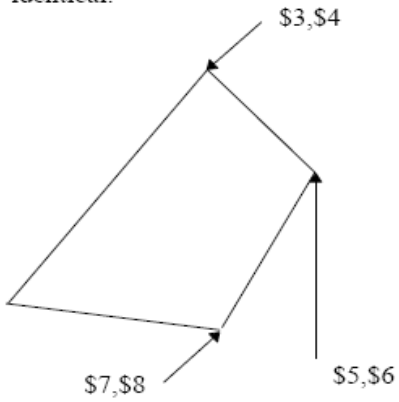
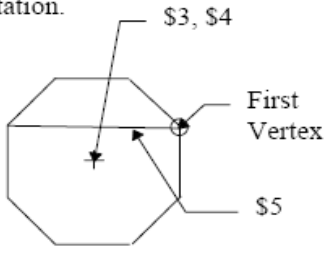
<primitive number>,<modifier\$1>,<modifier\$2>,<modifier\$3>,...*

the primitive number with modifiers. The primitive number identifies the geometry (outline, polygon, etc.). The modifiers differ with the various primitive numbers. Use either actual values (for example, 0.050 for a width) or a variable placeholder (for example, \$1 for exposure on/off).

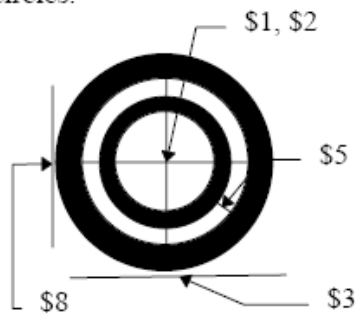
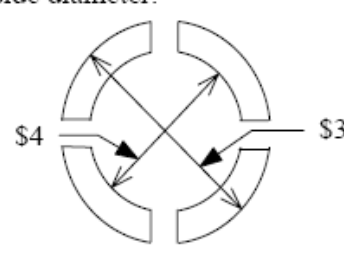
AM
Aperture Macro
Table 3 Aperture macro primitives

Primitive number	Description	Variable Modifiers	Description
1	Circle 	\$1	Exposure on/off 0 = OFF (laser off, no exposure) 1 = ON (laser on, image exposed) 2 = reverse current exposure state
		\$2	Diameter
		\$3	X center position
		\$4	Y center position
2 or 20	Line (vector): a line defined by width, and beginning and end points. The line ends are rectangular. 	\$1	Exposure on/off 0 = OFF (laser off, no exposure) 1 = ON (laser on, image exposed) 2 = reverse current exposure state
		\$2	Line width
		\$3	X start point
		\$4	Y start point
		\$5	X end point
		\$6	Y end point
		\$7	Rotation in degrees (+ = counterclockwise, - = clockwise)
21	Line (center): a centered rectangle defined by width, height, and center point. The end points are rectangular. 	\$1	Exposure on/off 0 = OFF (laser off, no exposure) 1 = ON (laser on, image exposed) 2 = reverse current exposure state
		\$2	Rectangle width
		\$3	Rectangle height
		\$4	X center point
		\$5	Y center point
		\$6	Rotation in degrees (+ = counterclockwise, - = clockwise)

AM
Aperture Macro

22	<p>Line (lower left): a rectangle defined by width, height, and the lower left point. The end points are rectangular.</p> 	\$1	Exposure on/off 0 = OFF (laser off, no exposure) 1 = ON (laser on, image exposed) 2 = reverse current exposure state
		\$2	Width
		\$3	Height
		\$4	X lower left point
		\$5	Y lower left point
		\$6	Rotation in degrees (+ = counterclockwise, - = clockwise)
3	End of file	none	Must be used to end .des files.
4	<p>Outline: an open or closed shape defined by a start point, n additional points (up to 50), and the X,Y coordinates that define them. For a closed shape, the first and last points must be identical.</p> 	\$1	Exposure on/off 0 = OFF (laser off, no exposure) 1 = ON (laser on, image exposed) 2 = reverse current exposure state
		\$2	n , the number of points in the outline
		\$3	X start point
		\$4	Y start point
		\$5	X point #1
		\$6	Y point #1
		\$7	X point #2
		\$8, etc.	Y point #2. Continue as needed.
		\$9 or the last number used	Rotation in degrees (+ = counterclockwise, - = clockwise)
5	<p>Polygon: a closed, symmetrical, centered shape defined by n vertices (3 to 10 inclusive), a center point, diameter, and rotation.</p> 	\$1	Exposure on/off 0 = OFF (laser off, no exposure) 1 = ON (laser on, image exposed) 2 = reverse current exposure state
		\$2	number of vertices (integer)
		\$3	X center point
		\$4	Y center point
		\$5	Diameter
		\$6	Rotation in degrees (+ = counterclockwise, - = clockwise)

**AM
Aperture Macro**

<p>6</p> <p>Moiré: a cross hair centered on n concentric circles defined by the center point, outside diameter, line thickness, and gap between circles.</p> 	\$1	X center point
	\$2	Y center point
	\$3	Outside diameter
	\$4	Circle line thickness
	\$5	Gap between circles
	\$6	number of circles
	\$7	Cross hair thickness
	\$8	Cross hair length
	\$9	Rotation in degrees (+ = counterclockwise, - = clockwise)
<p>7</p> <p>Thermal: a cross hair centered on a circle defined by outside and inside diameter.</p> 	\$1	X center point
	\$2	Y center point
	\$3	Outside diameter
	\$4	Inside diameter
	\$5	Cross hair thickness
	\$6	Rotation in degrees (+ = counterclockwise, - = clockwise)

Example 1

`%AMDONUT*1,1,$1,$2,$3*1,0,$4,$2,$3*%` Define an aperture macro named DONUT consisting of two concentric circles:

1,1,\$1,\$2,\$3 Circle (1), exposure on (1), diameter (\$1), X center (\$2), Y center (\$3) all to be supplied by AD parameter

1,0,\$4,\$2,\$3 Circle (1), exposure off (0), diameter (\$4, different from first circle), X center and Y center (\$2 and \$3, same as first circle)

AM Aperture Macro

The AD parameter using this macro might look like the following:

```
%ADD32DONUT,0.100X0X0X0.080*%
```

Define D-code 32 to be aperture macro **DONUT**. The diameter of the first circle will be 0.100. The center of both circles will be at 0,0. The diameter of the second circle will be 0.080.

\$1 = 0.100
\$2 = 0
\$3 = 0
\$4 = 0.080

Example 2

```
%AMDONUT*1,1,$1,$2,$3*$1=$2+0.030*1,0,$1-$4,$2,$3*%
```

Define an aperture macro named **DONUT** consisting of two concentric circles with diameter of the second circle defined as a function of the diameter and center point of the first:

1,1,\$1,\$2,\$3 Circle (1), exposure on (1), diameter (\$1), and center point X,Y (\$2, \$3) to be defined in the AD parameter

\$1=\$2+0.030 Define a variable to be used to calculate the diameter of second circle to be a function of the diameter and center point X coordinate of the first.

1,0,\$1-\$4,\$2,\$3 Circle (1), exposure off (0), diameter (\$1-\$4), and center point X,Y (\$2, \$3, same as the first circle).

The ADD parameter using this aperture macro might look like:

```
%ADD33DONUT,0.020X0X0X0.014*%
```

Define D-code 33 to be aperture macro **DONUT**. The diameter of the first circle 0.020. The center of both circles will be 0,0. The diameter of the second circle will be ((0 + 0.030) - 0.014).

Example 3

```
%AMDONUT*1,1,0.100X0X0*1,0,0.080X0X0*%
```

Define an aperture macro named **DONUT** consisting of two concentric circles, using primitive modifiers.

```
%ADD32DONUT*%
```

The resulting AD command only needs to reference the aperture macro name.

AS
Axis Select

The AS parameter is used to assign any two data axes to the output device A or B axes.

Data Block Format

AS A<X or Y>B<X or Y>*

where:

A and B are output device axes

X and Y are data axes

Default

AXBY

Example

%ASAYBX*% Assign the X axis data to the output device B axis and the Y axis data to the output device A axis.

FS Format Statement

The FS parameter is used to define the format of the input coordinate data and to define the allowable N, G, D, and M code lengths. It should be the first RS-274X parameter in the file. It is recommended that only one be used in the file. It is usually the first parameter.

The FS parameter allows you to specify the following format characteristics:

- Number of integer and decimal places in coordinate data (coordinate format)
- Zero omission (leading or trailing zeroes omitted)
- Absolute or incremental coordinate notation
- Sequence number (N-code) length
- General function code (G codes) length
- Draft code (D code) length
- Miscellaneous code (M code) length

Note: Decimal point programming is not supported.

Coordinate format

Coordinate format specifies how many integer and how many decimal places to expect in the coordinate data. For example, 2.3 format specifies two integer and three decimal places. A maximum of six integer and six decimal places may be specified (999999.999999). Different formats may be defined for the X and Y axes.

Zero omission

Zero omission compresses data by omitting either leading or trailing zeroes from coordinate values. Any given string of digits may be interpreted very differently depending on the zero omission specification. Coordinate format also affects how zero omission is interpreted.

Leading zero omission eliminates all zeroes that precede non-zero digits but retains following zeroes. For example, with 2.3 coordinate format, 15 would be interpreted as 0.015.

Note: Use leading zero omission for NO ZEROES OMITTED files.

Trailing zero omission eliminates all zeroes following non-zero digits but retains preceding zeroes. For example, with 2.3 coordinate format, 15 would be interpreted as 15.000.

Absolute or incremental notation

Coordinate values may be expressed as either absolute distances from a fixed 0,0 point or as relative distances from the preceding coordinate position.

RS-274D code lengths

The FS parameter can be used to specify length limits for the following standard RS-274D codes:

- N Sequence number
- G General function
- D Plot function
- M Miscellaneous function

These codes are described starting on page 45.

Data Block Format

```
%FS<L or T><A or I>[Nn][Gn]<Xn><Yn>[Dn][Mn]
```

where:

FS	The FS parameter
<L or T>	Use L to omit leading zeroes. Use T to omit trailing zeroes.
<A or I>	Use A for absolute coordinate values. Use I for incremental coordinate values.
[Nn], [Gn], [Dn], and [Mn]	Enter the code and an integer length limit, for example, N2 to specify two-digit sequence codes.
<Xn> and <Yn>	Enter X or Y and the number of integer and decimal places in the coordinate data for each axis, for example, X23 for X-axis data with two integer and three decimal places (99.999). 6.6 is maximum. The X and Y axes may have different values.

Example

```
%FSLAX25Y25*%
```

Coordinate data will have leading zeros omitted (L) and be expressed as absolute (A) positions with two integer and five decimal places in both axes (X25Y25).

IF
Include File

The IF parameter is used to identify an external file to be included in the RS-274X file. The files referenced by the IF parameter will be interpreted exactly as if they were included at the point of reference in the RS-274X file. Included files may also contain IF parameters, up to 10 levels of nesting.

The IF parameter is often used to include an external aperture file containing AD and AM parameters that describe the apertures used in the RS-274X file, sometimes referred to as an "external" aperture list. The IF parameter can also be used to include external data files, which allows you to merge multiple data files. Included files simplify the creation of standard plot sequences such as title blocks and coupons.

Data Block Format

<code>%IF<filename.ext>*%</code>
--

Examples

<code>%IFCOUPON3.GBR*%</code>	Include file COUPON3.GBR.
<code>%IFCIRCL.mac*%</code>	Include aperture macro file CIRCL.mac.
<code>%IFAPT004.des*%</code>	Include aperture description file APT004.des.

IJ Image Justify

The IJ parameter is used to override the absolute data coordinates for final placement of the image on the output device. The image may be centered or may be placed at an absolute position relative to the lower left of the platen.

Note: When centered, the pixel coordinates for the platen reside in the first quadrant (+X and +Y). X and Y are positive numbers, greater than zero and less than the platen size.

When more than one IJ parameter appear in the data, the final entry encountered is the one used.

Data Block Format

```
%IJ[A<parameter>B<parameter>][<offset>]*%
```

where:

IJ	the Image Justify parameter
A	the plotter A axis justification
<parameter>	L left or lower justification (default) C center justification
B	The plotter B axis justification
<parameter>	L left or lower justification (default) C center justification
<offset>	the starting position offset relative to 0,0

Default

None

Examples

%IJ*%	Left justify in X and lower justify in Y.
%IJAC*%	Center justify in X, lower justify in Y.
%IJACB.100*%	Center justify in X, offset .1 units in Y.
%IJALB.10*%	Left justify in X, offset .1 units in Y.
%IJB.100*%	Same as previous example.
%IJA1B1*%	Offset image 1 unit in X and Y.

IN
Image Name

The IN parameter is used to assign a name of up to 77 alphanumeric characters to the entire image of the RS-274X file. Information layers may also be named; see the LN parameter.

Data Block Format

<code>%IN<character string>*%</code>
--

where:

`<character string>` up to 77 alphanumeric characters except the asterisk (*).

Examples

`%INSOLDERMASK*%`

`%INPANEL_1*%`

IO Image Offset

The IO parameter is used to offset an image from the 0,0 point. The offset is expressed as an increment in the units defined by the MO parameter along the output device A and B axis. The AS parameter is used to correlate data axes with output device axes. The offset may be different for each axis and may be entered for a single axis.

Data Block Format

```
%IOA<±n>B<±n>*%
```

where:

- IO** the Image Offset parameter
- A<±n>** the offset along the output device A axis
- B<±n>** the offset along the output device B axis

Default

```
%IOA0B0*%
```

Examples

- %IOA1.0B1.5*%** Offset the image 1.0 units along the A axis and 1.5 units along the B axis from 0,0.
- %IOB5.0*%** Offset the image 5.0 units along the B axis from 0,0.

IP Image Polarity

The IP parameter is used to specify the positive or negative polarity of the entire file image. This *image polarity* differs from *layer polarity*, which is specified by the LP parameter and which applies only to one or more data layers of the entire image.

Data Block Format

```
%IP<POS or NEG>*%
```

where:

IP the IP parameter
<POS or NEG> Use **POS** for positive polarity, **NEG** for negative polarity.

Default

```
%IPPOS*%
```

Example

```
%IPNEG*%              Output the entire image with negative polarity.
```

IR
Image Rotation

The IR parameter is used to rotate the entire image counterclockwise in 90° increments about the 0,0 coordinate. All apertures follow the rotation. If you do not use the IR parameter, 0° rotation is assumed.

Data Block Format

%IR<90 or 180 or 270>

where:

IR the IR parameter
<90 or 180 or 270> Enter the desired value.

Default

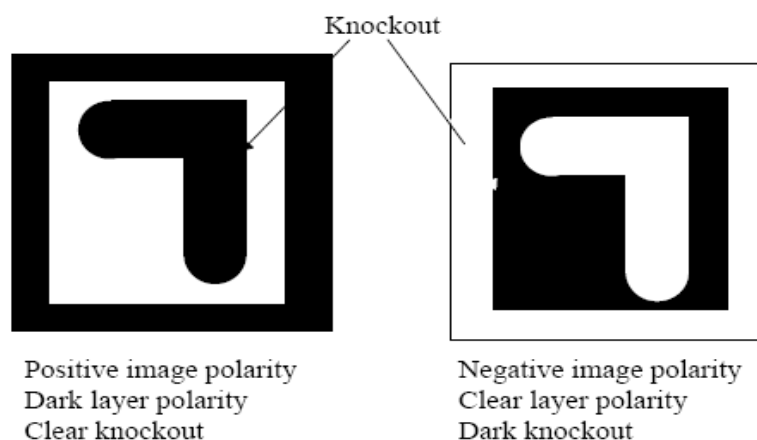
%IR0*%

Example

%IR90*% Rotate the entire image 90° counterclockwise.

KO Knockout

The KO parameter is used to specify that a rectangular region of the image will have the opposite polarity of the image or layer in which it occurs (either clear or dark), making it a "knockout" from the surrounding region.



KO is typically used to :

- set the initial background polarity of the final image
- set travel extent along the Y axis
- knock out a region around a component
- disable knockout on the current data layer

A knockout can be defined in one of two ways:

- lower left point and height and width
- border width around a component

When a knockout is defined around a component, the knockout is applied to all data following the KO parameter until the knockout is disabled. To disable a previously defined knockout, enter KO with no modifiers.

Data Block Format

```
%KO[C or D][X<coordinate>Y<coordinate>I<width>J<height> ]or [K<border dimension>]
```

where:

KO the KO parameter

C or D Enter **C** for clear or **D** for dark. To create a knockout defined by the data extents, do not enter modifiers. To disable a previously enabled knockout, enter neither **C** nor **D**.

X<lower left X coordinate>Y<lower left Y coordinate>I<width>J<height>

Use this modifier to define the knockout by a lower left point, width, and height.

K<border dimension> Use this modifier to define the knockout as a border around a component. Express the dimension in units specified by the MO parameter.

Examples

%KODX0I0I20J26*% Create a dark knockout that extends from 0,0 to 20,26. This in effect sets the Y travel extent.

%KOCK.050*% Create a clear knockout .050 units around all sides of the data that follows the KO parameter until the knockout is disabled.

%KOD*% Make the region defined by the extents of the data following the command dark.

%KO*% Disable a previously enabled knockout.



LN
Layer Name

The LN parameter is used to assign a name of up to 77 alphanumeric characters to the information layer that follows the parameter in the RS-274X file. Entire image files may also be named; see the IN parameter.

Data Block Format

<code>%LN<character string>*%</code>
--

where:

`<character string>` up to 77 alphanumeric characters except the asterisk (*).

Examples

`%LNSOLDERMASK*%`

`%LNINTERNAL_VCC*%`

LP Layer Polarity

The LP parameter is used to specify the positive or negative polarity of the information layer or layers following it. This *layer polarity* differs from *image polarity*, which is specified by the IP parameter and which applies to the entire image. The layer polarity applies to all data following the LP parameter until another LP parameter is encountered.

Data Block Format

```
%LP<C or D>*%
```

where:

LP the IP parameter
<C or D> Use **C** for clear polarity, **D** for dark polarity.

Default

```
%LPD*%
```

Example

```
%LPC*%                      Make all succeeding data clear.
```


MI Mirror Image

The MI parameter is used to turn mirror imaging either on or off. When on, all A- and/or B-axis data following the parameter will be mirrored (that is, inverted or multiplied by -1) until another MI command is used. Notice that mirroring A-axis data flips the image about the B axis. Mirroring B-axis data flips the image about the A axis.

Note: MI does not mirror special apertures.

The AS parameter is used to correlate the X and Y axes with the output device A and B axes.

Data Block Format

<code>%MI[A<0 or 1>][B<0 or 1>]*%</code>
--

where:

- | | |
|------------------------|--|
| MI | the MI parameter |
| A<0 or 1> | To invert A-axis data (flipping the image about the B-axis), enter A1. To disable, enter A0. |
| B<0 or 1> | To invert B-axis data (flipping the image about the A-axis), enter B1. To disable, enter B0. |

Default

`%MIA0B0*%`

Example

`%MIA0B1*%` Disable mirroring of A-axis data. Invert B-axis data, flipping the image about the A-axis.

**MO
Mode**

The MO parameter specifies that dimension data should be interpreted as inches or millimeters. Integer and decimal place format is specified by the FS parameter. Inches are assumed if units are not specified.

Data Block Format

```
%MO<IN or MM>*%
```

where:

MO the MO parameter

<IN or MM> Enter **IN** to specify inches. Enter **MM** to specify millimeters.

Default

```
%MOIN*%
```

Example

```
%MOIN*% Dimension data will be expressed in inches.
```

OF Offset

The OF parameter is used to offset the final image up to ± 99999.99999 units from the imaging device 0,0 point. The data may be offset along the imaging device A or B axis, or both. Values used with the OF parameter are expressed in units specified by the MO parameter, are always absolute, and are used primarily with absolute coordinate data. Incremental coordinate data may be offset simply by moving the imaging device to the desired offset position before starting the plot. The FS parameter specifies whether the data is absolute or incremental.

If an embedded FS parameter changes the format from absolute to incremental, the OF parameter value is saved and reinstated another FS parameter returns the format to absolute.

Data Block Format

```
%OF[A< $\pm$ offset value>][B< $\pm$ offset value>]
```

where:

OF	the OF parameter
A \pm n	Offset along the A axis. Use 5.5 format.
B \pm n	Offset along the B axis. Use 5.5 format.

Default

```
%OFA0B0*%
```

Example

```
%OFA1.0B1.0*%                    Offsets the plot 1 unit from 0,0 along both the A and B axes.
```

PF
Plot Film

The PF parameter is used to indicate to the operator the film (or other media) to be used to image the data file.

Data Block Format

`%PF<name>*%`

where:

PF the PF parameter

<name> up to 20 alphanumeric characters; asterisk (*) is an illegal character

SF
Scale Factor

The SF parameter is used to specify a scale factor of from 0.0001 to 999.99999 for the output device A- and/or B-axis data. The factor may be different for each axis. All data following the parameter will be multiplied by the factor until another SF parameter is encountered. The AS parameter is used to correlate the X and Y data axes with the imaging device A and B axes.

Data Block Format

```
%SF[A<factor>][B<factor>]*%
```

where:

SF	the SF parameter
A<factor>	the A-axis data factor
B<factor>	the B-axis data factor

Default

```
%A1B1*%
```

Example

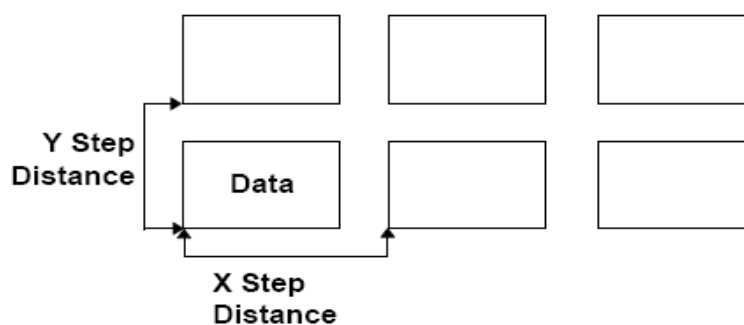
```
%SFA.5B3*%
```

 Multiply A-axis data by .5, B-axis data by 3.

SR Step and Repeat

The SR parameter is used to duplicate the data following the parameter a specific number of times (repeats) at a regular intervals (steps). The number of times the image is repeated and the space between repeats may be specified independently for X and Y data.

When entered without parameters, it is also used to disable a previous SR parameter.



Data Block Format

```
%SR[X<number of repeats>][Y<number of repeats>][I<X-axis step>][J<Y-axis step>]
```

where:

SR	the SR parameter
X<number of repeats>	the number of times the data will be repeated along the X-axis
Y<number of repeats>	the number of times the data will be repeated along the Y-axis
I<X-axis step distance>	the distance between the X-axis repeats
J<Y-axis step distance>	the distance between the Y-axis repeats

Default

```
%SRX1Y1I0J0*%
```

Examples

```
%SRX4I5.0J2*%
```

Repeat the image four times along the X axis with 5.0 units from one step to the next. The J modifier will be ignored because no Y repeats were specified.

SR
Step and Repeat

%SRX2Y3I2.0J3*%

Repeat the image twice along the X axis and three times along the Y axis. X-axis repeats will be spaced 2.0 units apart. Y-axis repeats will be spaced three units apart.

%SR*%

Disable a previous SR parameter.

Standard RS-274D Codes

This section describes standard RS-274D codes (D codes, G codes, and M codes) that are applicable to raster output.

D Codes

D codes (draft codes) select apertures and determine whether the feature described should be imaged as a line or “flashed”. Table 4 lists supported D codes.

Table 4 D Codes

Code	Function	Comments
D01 (D1)	Draw line, exposure on	You cannot draw using an aperture defined by an aperture macro (AM parameter). These apertures can only be flashed.
D02 (D2)	Exposure off	
D03 (D3)	Flash aperture	D03 remains in effect until a new layer is encountered.
D10-D999	Select an aperture defined by an AD parameter.	

G Codes

G codes are general function codes. They specify how the coordinate data should be interpolated (linear or circular), turn the Polygon Area Fill feature on and off (see page 49 for more information on Polygon Area Fill), and can also be used to specify absolute or incremental format. Table 5 lists supported G codes.

Table 5 G Codes

Code	Function	Comments
G00	Move	Affected by Polygon Area Fill (see page 49.)
G01	Linear interpolation (1X scale)	See page 47.
G02	Clockwise circular interpolation	See page 47.
G03	Counterclockwise circular interpolation	
G04	Ignore data block	
G10	Linear interpolation (10X scale)	See page 47.
G11	Linear interpolation (0.1X scale)	
G12	Linear interpolation (0.01X scale)	
G36	Turn on Polygon Area Fill	See page 49.
G37	Turn off Polygon Area Fill	See page 49.
G54	Tool prepare	Usually precedes an aperture D-code
G70	Specify inches	See also MO parameter.
G71	Specify millimeters	See also MO parameter.
G74	Disable 360° circular interpolation (single quadrant)	See <i>Circular Interpolation</i> , page 47.
G75	Enable 360° circular interpolation (multiquadrant)	See <i>Circular Interpolation</i> , page 47.
G90	Specify absolute format	See also FS parameter.
G91	Specify incremental format	See also FS parameter.

Linear Interpolation (G01, G10, G11, G12)

Linear interpolation plots a straight line from the present position to the X,Y coordinate specified by the data block.

Data Block Format

G01 X±m.n Y±m.n Dnn

where:

G01 Specifies linear interpolation

X±m.n Y±m.n Defines the line end point

Dnn D-code (exposure on or off)

Circular Interpolation (G02, G03, G74, G75)

There are two types of circular interpolation: single quadrant (90°) and multi-quadrant (360°). Single quadrant interpolation produces an arc. Multi-quadrant interpolation can produce arcs that are larger than 90° and also circles.

Single Quadrant Circular Interpolation (G74)

G02 and G03 specify single quadrant (90°) circular interpolation; G74 disables it. Single quadrant circular interpolation plots an arc within one quadrant (90°). Single quadrant arcs must fit entirely within the quadrant in which they begin. A separate data block is required for each quadrant. A minimum of four data blocks is required to plot a circle.

Data Block Format

Gnn X±m.n Y±m.n Im.n Jm.n Dnn

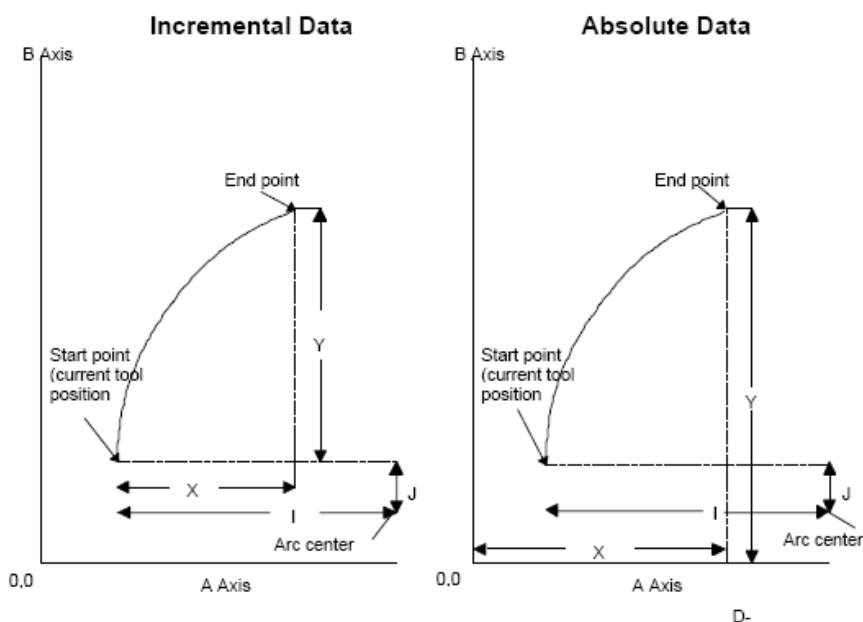
where:

Gnn G02 specifies clockwise circular interpolation
G03 specifies counterclockwise circular interpolation

X±m.n Y±m.n Defines the arc end point. These variables are in the format defined by the format statement (FS parameter). A sign is optional.

Im.n Jm.n Defines the incremental distance between the arc start point and the center measured parallel to the X and Y axes respectively. Notice that these numbers are unsigned values. The direction to the center is determined implicitly.

Dnn D-code (exposure on or off)



Multiquadrant (360°) Circular Interpolation (G74, G75)

A data block containing only G75 specifies 360° circular interpolation, which plots arcs in more than one quadrant using only one data block. Every block following a G75 code will be interpreted as 360° interpolation until a G74 is encountered. The I and J variables will be considered signed. If no sign is present, the circle will be in a positive direction from the start point.

A G74 code turns 360° multiquadrant circular interpolation off, reverting to single quadrant interpolation. To turn circular interpolation off and revert to linear interpolation, use G01.

Data Block Format

Gnn X±m.n Y±m.n ±Im.n ±Jm.n Dnn
--

where:

Gnn G74 turns off 360° circular interpolation
 G75 turns on 360° circular interpolation

X±m.n Y±m.n Defines the arc end point

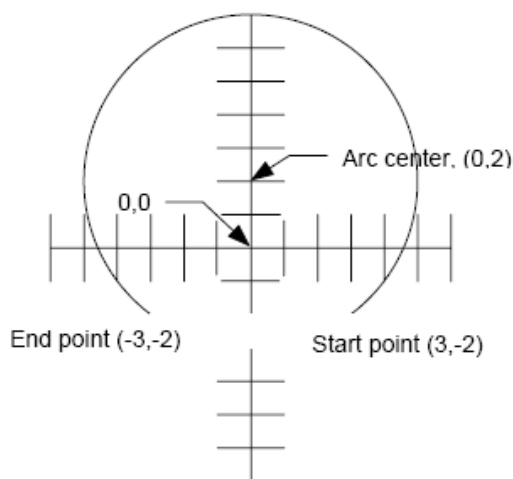
±Im.n J±m.n Defines the distance between the arc start point and the center measured parallel to the X and Y axes respectively. These variables are always incremental values in the format defined by the format statement (FS parameter). A sign is optional.

Dnn D-code (exposure on or off)

An example of multiquadrant interpolation is shown on the next page.

```
%FSLAX43Y43*%
G75*
G01X3000Y-2000D02*
G03X-3000Y-2000I-3000J4000D01
G01*
```

DESIRED PLOT

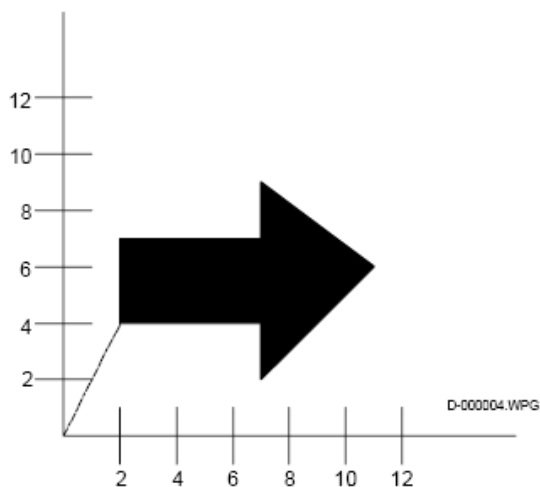


Polygon Area Fill (G36, G37)

G36 and G37 provide a more efficient means of filling closed polygons than stroke fill. When these codes are used, the filled area is defined simply by its closed outline. Stroke fill is an inefficient method of filling a polygon.

G36 turns on polygon area fill. G37 turn it off. There are no variables or apertures. Following a G36 and before G37, all lines drawn with D01 are considered edges of the polygon. D02 closes and fills the polygon.

```
G36*
X2Y4D02*
X7D01*
Y2D01*
X11Y6D01*
X7Y9D01*
X7D01*
X2D01*Y4D01*G37*
```



M Codes

M-codes identify the end of a file. Three M codes are commonly used:

M00	Program stop
M01	Optional stop
M02	End of program

Glossary

ABSOLUTE POSITION: Position expressed as a distance from the 0,0 point in the data.

APERTURE: Previously, an opening in a wheel through which light passed to expose film. Currently a D code assignment and description of geometry that determines the shape of a feature.

APERTURE MACRO: A mass parameter that describes the geometry of a special aperture and assigns it to a D code.

APERTURE PARAMETER: A mass parameter (AD or AM) that assigns an aperture description to a D code.

CIRCULAR INTERPOLATION: Specifies that the data should be interpreted as arcs; may be single-quadrant or multiquadrant.

COMPOSITE IMAGE: The entire image, including all information layers.

COORDINATE DATA: X,Y position data that describes placement of features in the image.

D CODES: Draft (tool) RS-274D codes. They specify tool exposure action (line draw or flash).

DELIMITER: A character that indicates the beginning and end of a mass parameter.

DIRECTIVE PARAMETER: A mass parameter that controls overall file processing.

EXTENDED GERBER FORMAT: Gerber data that includes mass parameters.

FUNCTION CODES: G codes, D codes, M codes that are part of RS-274D.

G CODES: General function RS-274D codes. They specify interpolation, polygon area fill, etc.

GERBER DATA: Data expressed in Gerber Format.

GERBER FORMAT: A subset of RS-274D Word Address Format that is the universal plotter language; may also contain mass parameters, whose presence make it Extended Gerber Format, or RS-274X.

IMAGE PARAMETER: A parameter that supplies information about an entire image.

KNOCKOUT: A rectangular region about an information layer whose polarity is the opposite of the layer polarity.

LAYER: A named information component of Gerber data that may be treated as a unit, for example, rotated or repeated; has no relationship to a physical PCB layer.

LAYER-SPECIFIC PARAMETER: A mass parameter that applies to a single information layer (for example KO, LN, LP, and SR).

LINEAR INTERPOLATION: Specifies that the data should be interpreted as straight lines.

MASS PARAMETERS: Commands or codes that may be embedded in Gerber Data that specify how the data should be processed.

MULTI-QUADRANT INTERPOLATION: Specifies that the data should be interpreted as arcs that can extend into more than one quadrant, up to 360°).

NEGATIVE: An artwork in which the intended conductive pattern is transparent to light and the areas to be free from conductive material are opaque.

NUMERICAL PRECISION: The number of integer and decimal places used to express a number.

POLARITY: Describes whether the circuitry will be imaged as dark on a clear background (positive) or clear on a dark background (negative). Polarity may be applied to an entire image or to a single layer.

POLYGON AREA FILL: A feature that provides a more efficient means of creating solid (filled) polygons than stroke fill.

RELATIVE POSITION: Position expressed as a distance from the last position.

RS-274D: Electronics Industries Association (EIA) standard data format; a superset of Gerber Format.

RS-274X: Extended Gerber Format, that is, Gerber Format with mass parameters.

SINGLE QUADRANT INTERPOLATION: Specifies that the data should be interpreted as an arc that must fit entire within a single quadrant (90°).

STEP AND REPEAT: A method by which successive exposures of a single image are made to produce a multiple image production master.

STROKE FILL: An inefficient means of creating solid (filled) polygons by “painting” the area.

X DATA: Gerber data that includes mass parameters.

Index

A

absolute data coordinates	
override by IJ	29
Absolute notation	26
ABSOLUTE POSITION	51
absolute values	7
AD	6, 16
AM	6, 16, 19
APERTURE	51
Aperture Defintion	16
Aperture Description	6
Aperture Editor	16
Aperture Macro	6, 19, 51
APERTURE PARAMETER	51
Aperture parameters	5
apertures	5
special	16
standard	16
arc	47
arcs	7
AS	5, 25
Assistance	2
Axis assignment	8
Axis Select	5, 25

B

Bulletin Board Service (BBS)	2
------------------------------	---

C

center image	29
Circle	6, 17, 47, 48
circular interpolation	4, 47, 51
COMPOSITE IMAGE	51
coordinate data	4, 7, 51
format	26

D

D codes	7, 16, 26, 45, 51
data block	3, 7, 8
format	15
maximum length	8
Data types	4
Decimal point programming	26

DELIMITER	51
DIRECTIVE PARAMETER	51
Directive parameters	5

E

Electronic Industries Association	2
end-of-block character	8
end-of-program code	8
English units	
specifying	39
<i>Extended Gerber Format</i>	1, 4, 51

F

file naming conventions	8
File structure	3
Format Statement	4, 5, 26
FS	5, 26
function codes	4, 51
functions codes	10

G

G codes	7, 26, 46, 51
G01, G10, G11, G12	47
G02, G03, G74, G75	47
G36, G37	49
G74	47
G74, G75	48
General File Preparation Guidelines	8
general function codes	46
general functions	7
Gerber data	1, 51
Gerber Format	1, 51
Gerber GPC Aperture Editor	16
Glossary	51

I

I,J data	7
IF	6, 28
IJ	5, 29
Image Justify	5, 29
Image Name	5, 30
Image Offset	5, 31
IMAGE PARAMETER	5, 51
image placement	29

Image Polarity	5, 32
Image Rotation	5, 33
IN	5, 30
Inches	
specifying	39
Include File	6
incremental data	7
Incremental notation	26
IO	5, 31
IP	5, 32
IR	5, 33
J	
justify image	29
K	
Knockout	6, 34, 51
KO	6, 34
L	
Layer	52
generated by mass parameters	4, 6
Layer Name	6, 36
Layer Polarity	6, 37
Layers	3
LAYER-SPECIFIC PARAMETER	6, 52
Leading zero omission	26
Line (center)	6, 21
Line (lower left)	6, 22
Line (vector)	6, 21
Linear Interpolation	47, 52
LN	6, 36
LP	6, 37
M	
M codes	7, 26, 50
M00 or M02	8
Mass parameters	4, 52
Metric units	
specifying	39
MI	5, 38
Millimeters	
specifying	39
Mirror Image	5, 38
Miscellaneous parameters	6
MO	5, 39
modal	7
Mode	39
Mode of units	5
Moiré	6, 23
Multiquadrant (360°) Circular Interpolation ...	48, 52
N	
N codes	7
Name	
image	30
layer	36
NEGATIVE	52
NO ZEROES OMITTED	26
notation	26
Numerical precision	8, 52
O	
Obround (oval)	17
OF	5, 40
Offset	5, 40
offset image	31
Order of entry	
RS-274D	10
RS-274X	9
Ordering information	2
Outline	6, 22
P	
Parameter delimiter	4, 8
Parameters	
placement of	5
PF	5, 41
Plot Film	5, 41
polarity	34, 52
image	32
layer	37
Polygon	6, 22
polygon area fill	1, 49, 52
primitives	19
R	
raster device	5
Rectangle or square	17
Regular polygon	18
RELATIVE POSITION	52
Rotate	
image	33
RS-274D	1, 2, 7, 52
code length	27
Data Guidelines	10
order of entry	10
RS-274X	1, 4, 8, 15, 52
defaults	9
order of entry	9
Parameter Guidelines	8
position in file	9
required and optional	9
S	
Sample Files	10
Scale Factor	5, 42
Sequence number	7, 26
SF	5, 42
Single Quadrant Circular Interpolation	47, 52
Special apertures	16, 19
SR	6, 43
Standard apertures	16, 17
Standard RS-274D Codes	4, 45

Step and Repeat6, 43, 52
STROKE FILL52

T

Technical Assistance Center2
Thermal.....6, 23
Trailing zero omission26

U

Units.....39
upper case
 required for entry8

V

Vector plotters 5

W

Web Page 2
word address format..... 4

X

X data 1, 52
X,Y data 7

Z

Zero omission..... 26

BARCO

GERBER SYSTEMS

Part Number 414 100 014 C

September 21, 1998