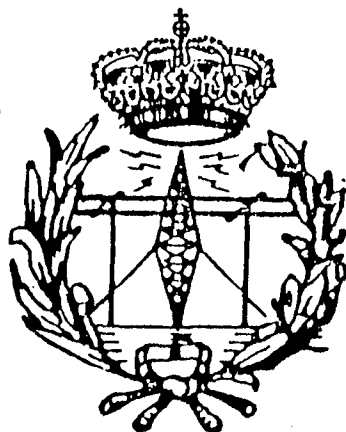


UNIVERSIDAD POLITECNICA
DE CANARIAS

ESCUELA UNIVERSITARIA
DE TELECOMUNICACIONES



TITULO: BASE DE DATOS PARA GRAFICOS POR ORDENADOR
Y MULTIMODEM F.S.K. PARA LA
RED TELEFONICA CONMUTADA

CARRERA: INGENIERIA DE TELECOMUNICACIONES

ESPECIALIDAD: EQUIPOS ELECTRONICOS

AUTOR: JORGE PORTILLO MENIZ

TUTOR: SEBASTIAN SUAREZ GIL

SEPTIEMBRE 1988

UNIVERSIDAD POLITECNICA DE CANARIAS
INGENIERIA TECNICA DE TELECOMUNICACIONES
PROYECTO FIN DE CARRERA

*TITULO: BASE DE DATOS PARA GRAFICOS POR ORDENADOR
Y MULTIMODEM F.S.K. PARA LA
RED TELEFONICA CONMUTADA.*

AUTOR:

TUTOR:

Jorge Portillo Meniz

Sebastián Suárez Gil

SEPTIEMBRE 1988

Dedico este trabajo a mi
esposa e hijos, Sin la
ayuda y comprensión que me
han dedicado, no habría
sido posible la realización
de este proyecto.

CONTENIDO

	PAGINA
INTRODUCCION	1
I. DR. LOGO	4
I.1. Dr. Logo, Programación Modular	5
I.2. Principales primitivas de Dr. Logo	13
II. BASE DE DATOS PARA EL CALCULO Y DIBUJO DE FIGURAS GEOMETRICAS	20
II.1. PRESENTACION	21
II.1.1. Módulo principal	22
II.1.2. Procedimientos	23
II.1.3. Listado de procedimientos	24
II.2. GEOMETRIA PLANA	28
II.2.1. Cálculo y dibujo de triángulos	32
II.2.1.1. Triángulo rectángulo	35
II.2.1.2. Ejemplos de aplicación y procedimientos	40
II.2.1.3. Triángulo isósceles	47
II.2.1.4. Ejemplos de aplicación y procedimientos	51
II.2.1.5. Triángulo equilátero	53
II.2.1.6. Ejemplos de aplicación y procedimientos	61
II.2.1.7. Triángulo cualquiera	67
II.2.1.8. Ejemplos de aplicación y procedimientos	75
II.2.2. Dibujo de polígonos regulares	89
II.2.2.1. Módulo y procedimientos	89
II.2.2.2. Ejemplos de aplicación	91
II.2.2.3. Listado de procedimientos	92
II.2.3. Dibujo de rectas	94
II.2.3.1. Módulo y procedimientos	94
II.2.3.2. Ejemplos de aplicación	97
II.2.3.3. Listado de procedimientos	99
II.2.4. Dibujo de curvas	101
II.2.4.1. Curvas trigonométricas	102
II.2.4.2. Ejemplos de aplicación	106
II.2.4.3. Otras curvas	110
II.2.4.4. Ejemplos de aplicación	112
II.2.4.5. Listado de procedimientos	116

II.3. GEOMETRIA DEL ESPACIO	124
II.3.1. Cálculo y dibujo del prisma recto	125
II.3.1.1. Módulo-(Dibujo)- y procedimientos	125
II.3.1.2. Ejemplos de aplicación	127
II.3.1.3. Módulo-(Cálculo)- y procedimientos	130
II.3.1.4. Ejemplos de aplicación	133
II.3.1.5. Listado de procedimientos	136
II.3.2. Cálculo y dibujo de la pirámide	143
II.3.2.1. Módulo-(Dibujo)- y procedimientos	144
II.3.2.2. Ejemplos de aplicación	145
II.3.2.3. Listado de procedimientos	147
II.3.2.4. Módulo-(Cálculo)- y procedimientos	149
II.3.2.5. Ejemplos de aplicación	154
II.3.2.6. Listado de procedimientos	156
II.3.3. Cálculo y dibujo del cubo	160
II.3.3.1. Módulo-(Dibujo)- y procedimientos	161
II.3.3.2. Ejemplos de aplicación	162
II.3.3.3. Módulo-(Cálculo)- y procedimientos	163
II.3.3.4. Ejemplos de aplicación	165
II.3.3.5. Listado de procedimientos	166
II.3.4. Cálculo y dibujo del octaedro	170
II.3.4.1. Módulo-(Dibujo)- y procedimientos	171
II.3.4.2. Ejemplos de aplicación	171
II.3.4.3. Módulo-(Cálculo)- y procedimientos	173
II.3.4.4. Ejemplos de aplicación	175
II.3.4.5. Listado de procedimientos	176
II.4. LISTADO DE LA BASE DE DATOS	181
II.5. PROGRAMA DEMOSTRACION	249
II.5.1. Listado del programa demostración	250

III. MULTIMODEM F.S.K. PARA LA RED TELEFONICA CONMUTADA	254
INTRODUCCION	255
III.1. ESPECIFICACIONES (Normas del CCITT)	256
III.2. INSTRUCCIONES DE OPERACION Y TEORIA DE FUNCIONAMIENTO	259
III.2.1. Instrucciones de operación	260
III.2.2. Teoria de funcionamiento	262
III.3. FUENTE DE ALIMENTACION	269
III.4. COMPONENTES	273
III.4.1. Lista de componentes	274
III.5. ESQUEMAS ELECTRICOS	276
III.6. PRESUPUESTO	283
APENDICE	287
Características de los componentes	

INTRODUCCION

Este Proyecto esta dividido en dos apartados. El apartado (II) es una base de datos para gráficos por ordenador y el apartado (III) un Módem F.S.K. para la red telefónica conmutada.

Aunque cada apartado por sí solo podría constituir materia suficiente como para elaborar un proyecto se han propuesto ambos por dos motivos.

El primer motivo ha sido que el proyecto no adoleciera de una materia tan representativa de la especialidad de Equipos Electrónicos como es la electrónica, ya que en principio la idea fue proponer sólomente la base de datos para gráficos.

El segundo motivo ha sido que ambos apartados pueden formar un sólo proyecto, dado que el acceso a la base de datos para otros terminales distantes se puede hacer a través del módem propuesto. De esta manera el acceso a la base de datos es mucho más amplio, permitiendo el uso a terminales no inteligentes mediante un programa de comunicaciones.

En el apartado (I) se da una introducción del lenguaje utilizado, sus principales características, estructuras de programación y primitivas más importantes.

El apartado (II) es una base de datos para gráficos. Esta base de datos está dedicada a los alumnos de BUP y COU. Con ella se pretende dar una aplicación práctica de los equipos informáticos para algo más que la informática en sí. Con este programa de aplicación los alumnos de matemáticas, a la vez que se familiarizan con el ordenador, pueden comprobar resultados de problemas y lo que es más importante, pueden ver las figuras geométricas sobre las que están trabajando.

Esta base de datos es fácilmente ampliable a cualquier tipo y nivel de estudio de matemáticas y de geometría en especial.

El estudiante que haga consulta a la base de datos podrá, con un aporte mínimo de datos, hallar todas las variables de la figura geométrica que desee y además el programa le dibujará la figura sobre la que está haciendo los cálculos. De esta manera podrá cotejar resultados y ver como es en realidad la figura del problema.

Esta base de datos se ha elaborado poniendo especial cuidado en la sencillez de su operación. Se han optimizado al máximo los recursos para que la ejecución de las órdenes sean lo más rápida posible. En todo momento se informa al usuario del programa, donde se encuentra y como trasladarse a otros módulos del programa. El usuario del programa podrá acceder a los diferentes módulos de dos maneras, cargando en el espacio de trabajo el módulo que necesite o a través del menú general.

Esta segunda manera ofrece el inconveniente de que el usuario tendrá que ir pasando por todo el árbol que compone el programa hasta llegar a la rama que quiere usar.

La estructura modular de Dr. Logo le hace especialmente útil para este tipo de programa. La ampliación de la base de datos es muy sencilla y sólo está limitada por la memoria del soporte físico que utilicemos para almacenarla.

En el apartado (III) se encuentra el Multimodem F.S.K. para la red telefónica conmutada. En el diseño de este módem se ha puesto especial cuidado en el abaratamiento del coste.

Algunos de los componentes que se han utilizado en su diseño, por su novedad, aún no están en el mercado nacional. El uso de estos componentes y el propio diseño hacen del Multimodem un aparato muy versátil y barato.

Este módem dispone de varias velocidades y protocolos, para cambiar de módem se actúa sobre un conmutador.

A través de este conmutador se hace la programación para los diferentes modos de que dispone.

INTRODUCCION

A DR. LOGO

Introducción a Dr. Logo

El lenguaje Dr. Logo fué desarrollado por un equipo de científicos de la informática dirigidos por Seymour Papert, del Massachussets Institute of Technology (MIT). Su objetivo era desarrollar un lenguaje con el que fuera fácil experimentar y que facilitara la enseñanza de los conceptos matemáticos y de programación de ordenadores.

Una característica de LOGO es su "tortuga"; un objeto móvil que se controla mediante instrucciones sencillas. Esta característica de LOGO se ha hecho tan popular que otros lenguajes de programación, en sus versiones modernas, la han incluido para facilitar la explotación de su capacidad gráfica. Generalmente la "tortuga" consiste en un carácter que aparece en la pantalla. Papert y su equipo le dieron el nombre de tortuga para ayudar a los alumnos a identificar el objeto al que le estaban dando instrucciones.

Los diseñadores de LOGO pensaron en él como una herramienta educativa "sin limite inferior ni superior", y, de hecho, desde los dibujos de alumnos de Preescolar hasta los gráficos más complejos en tres dimensiones y desde los programas elementales de listas hasta los sistemas expertos (programas que hacen deducciones lógicas), todo tiene cabida en LOGO.

primitivas

Las instrucciones básicas de Dr. Logo son las llamadas "primitivas"; sus efectos están predefinidos. Hay en total unas doscientas primitivas, para control de la tortuga, realización de operaciones matemáticas y proceso de la información. Sus nombres constituyen el vocabulario de Dr. Logo.

procedimientos

Los procedimientos son instrucciones de Dr. Logo, análogas en todos los aspectos a las primitivas con la diferencia de que son elaboradas por el usuario.

La definición de un procedimiento tiene la siguiente forma:

```
to nombre-procedimiento
serie de instrucciones (en una o varias líneas)
end
```

Como nombre del procedimiento se puede poner cualquier palabra que no sea el nombre de ninguna primitiva y que no empiece con un número. Una vez definido el procedimiento queda almacenado en el espacio de trabajo de Dr. Logo, el cual reconoce el nombre del procedimiento como nueva palabra de su vocabulario.

Desde este momento Dr. Logo tratará este procedimiento como si fuera una más de sus primitivas. Con la definición de nuevos procedimientos Dr. Logo puede llegar a ser un lenguaje más potente, capaz de hacer todo tipo de cometidos, tantos como el usuario halla definido previamente.

Tanto las primitivas como los procedimientos se ejecutan con sólo escribir sus nombres. Hay primitivas que para ejecutarse necesitan de una entrada además del nombre de la primitiva, cualquier operación aritmética o lógica sería un ejemplo de primitiva con entrada.

De la misma forma se pueden definir procedimientos con o sin entradas, un texto puede ser un ejemplo de procedimiento sin entrada, un procedimiento que halle la superficie de una figura geométrica puede ser un ejemplo de procedimiento con entrada, la entrada serían los datos necesarios para hallar la superficie de esa figura.

La definición de procedimiento con entrada tiene la siguiente forma:

```
to nombre-procedimiento :e1 :e2...:en
serie de instrucciones
end
```

variables

La entrada del procedimiento se denomina "variable local" y Dr. Logo sólo la utiliza dentro de ese procedimiento.

Las variables en Logo pueden ser locales y globales. En la definición del procedimiento con estradas vimos un ejemplo de variable local.

Las variables locales sólo pueden ser utilizadas por un procedimiento y por los procedimientos que éste invoque. En cuanto concluye la ejecución del procedimiento, Dr. Logo ya no puede decirnos cual es el valor de la variable, los valores no se almacenan en memoria.

Dr. Logo puede manejar variables globales. En las variables globales, una vez asignadas un valor con la instrucción "make", se conserva hasta que otra instrucción lo modifique.

Estas variables pueden figurar como entrada a otros procedimientos y ser utilizadas en las definiciones de procedimientos. Este tipo de variable no se menciona en la primera línea de la definición, por el contrario se definen a lo largo del procedimiento o incluso pueden definirse en un procedimiento que sea sólo de definiciones de variables globales.

Una vez ejecutado el procedimiento donde se encuentren definidas variables globales, los valores de estas se almacenan en posiciones discretas de memoria para poder ser consultadas en cualquier momento durante la ejecución del resto de los procedimientos.

Para saber el valor almacenado de cualquier variable se escribe :nombre-variable y al pulsar return nos devolverá el último valor almacenado en memoria de esa variable.

Para hacer cualquier tipo de operación aritmética o lógica con los valores almacenados de las variables, basta con poner sus nombres y al operar se hará con los valores que tengan almacenados en memoria.

Ejemplo: :V1/:V2 dara el cociente de los ultimos valores almacenados en memoria de las variables V1 y V2.

módulos

Una vez se tengan definidos todos los procedimientos que se necesiten para una labor determinada se puede comenzar a construir un Módulo. Un Módulo lo compone uno o más procedimientos unidos de manera que realicen una labor predefinida. A este Módulo se le da un nombre y se guarda en el soporte de memoria para llamarle cuando se necesite hacer la labor que realiza. Cada vez que se cargue un Módulo en el espacio de trabajo de Logo, junto con él se cargarán todos los procedimientos utilizados para su elaboración.

Se puede elaborar un Módulo en el que utilicemos, además de primitivas y procedimientos, otros Módulos ya definidos. De esta manera se puede llegar a cualquier grado de complejidad en la elaboración de un Módulo de programación. Un programa en Dr. Logo puede estar formado por uno o más Módulos, interconectados entre si con el software apropiado para hacer funcionar al conjunto de forma coherente.

Para que un Módulo se ejecute, desde el espacio de trabajo de Logo, primero hay que cargarlo, desde el soporte físico de memoria. Al igual que las primitivas y los procedimientos, los Módulos de programación pueden estar definidos con o sin entradas.

En Dr. Logo el Módulo principal de un programa se puede definir como el primero que llama durante su ejecución a otros Módulos que se encuentren en el cuerpo de su definición. Estos Módulos a su vez pueden invocar a otros en su ejecución y así hasta que la capacidad de memoria del espacio de trabajo de Logo lo permita.

De esta manera se consigue que un programa (procedimiento), que en otro lenguaje podría ser de cientos de líneas se pueda reducir en Logo a tres o cuatro palabras de dos o tres letras cada una. El programa se puede dividir en tantos módulos como su correcta ejecución aconseje.

Al elaborar programas que sean muy largos es muy importante tener el espacio de trabajo de Logo con el menor número posible de Módulos a la vez. Para conseguir esto se conectan los Módulos por medio de ordenes, que serán pequeños procedimientos, de manera que nunca halla más de un determinado número de Módulos en el espacio de trabajo. Estas ordenes harán que cuando se ha terminado con un Módulo, se borre del espacio de trabajo y seguidamente se cargue el siguiente que se va a ejecutar.

De esta manera la velocidad de proceso de la información por parte de Dr. Logo es mucho mayor, y los programas se ejecutarán con mucha más rapidez y eficacia.

editor de Dr. Logo

Dr. Logo dispone de un editor de textos desde donde se pueden elaborar los procedimientos que defina el usuario.

Para la edición se escribe ed "nombre-de-procedimiento +Rtrn. Con esta operación se entra en el editor de Logo.

Cuando se quiera editar un procedimiento ya confeccionado, se escribe ed "n-d-p +Rtrn y se entra en el cuerpo del procedimiento.

LAS PRIMITIVAS

DE DR. LOGO

PRINCIPALES PRIMITIVAS

DE DR. LOGO

PRIMITIVAS DE TORTUGA GRAFICA

.....

bk n.....(backward,atrás). Mueve la tortuga 'n' pasos hacia atrás.

fd n.....(forward,delante). Mueve la tortuga 'n' pasos hacia delante.

home.....Lleva la tortuga a la posición [0 0] (centro de la pantalla gráfica) y rumbo 0 (norte).

ht.....(hide turtle, ocultar tortuga). Hace la tortuga invisible, con lo que los dibujos se realizan a mayor velocidad.

lt n.....(left, izquierda). Gira el rumbo de la tortuga 'n' grados hacia la izquierda.

pd.....(pen down,bajar pluma). Baja la pluma de la tortuga para que ésta dibuje al moverse.

pe.....(pen erase,pluma de borrar). Hace que la tortuga borre todo lo que encuentra dibujado en su camino.

pu.....(pen up, subir pluma). Sube la pluma de la tortuga para que esta no dibuje al moverse.

rt n.....(right, derecha). Gira el rumbo de la tortuga 'n' grados hacia la derecha.

seth n.....(set heading, establecer rumbo). Orienta la tortuga con el rumbo absoluto especificado por 'n' (en grados).

setpos[n m].....(set position, establecer posición). Lleva la tortuga al punto de coordenadas especificado por 'n' y 'm'.

setx n.....Lleva la tortuga al punto cuya coordenada X es 'n', sin modificar la coordenada Y.

sety n.....Lleva la tortuga al punto cuya coordenada Y es 'n', sin modificar la coordenada X.

st.....(show turtle, mostrar tortuga). Hace la tortuga visible, si estaba oculta.

tf.....(turtle facts, datos de tortuga). Escribe en la pantalla información relativa a la tortuga: posición, rumbo, visibilidad.

towards[n m].....(hacia). Gira la tortuga de forma que quede mirando hacia el punto de coordenadas 'n', 'm'.

PRIMITIVAS DE PANTALLA GRAFICA

clean.....(limpiar). Borra la pantalla gráfica sin afectar a la tortuga.

cs.....(clear screen, borrar pantalla). Borra la pantalla gráfica y restablece la situación inicial de la pantalla: posición [0 0] y rumbo 0º, pluma abajo.

dot[n m].....(punto). Dibuja un punto en la posición especificada por 'n' y 'm'.

fence.....(valla). Establece un límite que confina la posición de la tortuga a la pantalla gráfica visible.

fs.....(full screen, pantalla completa). Asigna la pantalla entera a los gráficos.

setscrunch n.....(setscrunch, definir relación de escalas). Hace referencia a la relación entre las escalas vertical y horizontal en la pantalla gráfica. 'n' puede tener cualquier valor comprendido entre 0.1 y 10. El valor inicial es 0.468.

setsplit n.....Especifica el número de líneas de texto que debe haber en la pantalla mixta.

sf.....(screen facts, datos de pantalla). Escribe en la pantalla toda la información relativa a la pantalla gráfica.

ss.....(split screen, pantalla mixta). Establece una pantalla para texto dentro de la pantalla gráfica.

window.....(ventana). Permite que la tortuga se salga de los límites de la pantalla gráfica visible después de una orden wrap o fence.

wrap.....(superponer borde con borde). Hace que la tortuga aparezca por el lado opuesto de la pantalla cuando sobrepasa algún borde.

PRIMITIVAS DE PANTALLA DE TEXTO

.....

ct.....(clear text, borrar texto). Borra la ventana en la que está actualmente el cursor; lleva el cursor al extremo superior izquierdo de la ventana.

cursor.....Da como salida una lista de coordenadas que consiste en los números de columna y fila de la posición actual del cursor en la ventana de texto.

pr[a b ...].....Escribe en la pantalla de texto los objetos especificados entre corchetes y efectua un retorno de carro después de escribir el último objeto.

setcursor[n m]....Lleva el cursor a la posición especificada por 'n' y 'm'.

show[a b ..].....(mostrar). Escribe en la pantalla de texto el objeto de entrada mantiene los corchetes y efectua un retorno de carro.

ts.....(text screen, pantalla de texto). Asigna la pantalla completa como pantalla de texto.

type[a b..].....(mecanografiar). Escribe en la pantalla los objetos de entrada, suprime los corchetes externos de las listas y no efectua un retorno de carro después de escribir el último objeto.

VARIABLES

.....

erall.....(erase all, borrar todo). Borra todas las variables y procedimientos del espacio de trabajo.

ern[a b ..].....(erase name, borrar nombre). Borra del espacio de trabajo las variables especificadas.

(local "a "b..)....Hace que las variables de entrada sean accesibles solamente al procedimiento actual y a los procedimientos por él invocados.

make "variable a....Asigna a la variable el valor 'a'.

nodes.....Da el número de nodos que quedan libres en el espacio de trabajo.

po "a.....(print out, listar). Da el valor de la variable especificada.

poall.....(print out all, listar todo). Escribe las definiciones de todos los procedimientos y los valores de todas las variables que hay en el espacio de trabajo.

pons.....(PRINT OUT NAMES, LISTAR NOMBRES). Da la lista de los nombres y variables de todas las variables que hay en el espacio de trabajo.

recycle.....Libera el mayor número posible de nodos y reorganiza el espacio de trabajo.

thing "a.....(cosa). Da el valor del objeto citado.

OPERACIONES ARITMETICAS

arctan n.....Da como salida el arco (en grados) cuya tangente es 'n'.

cos n.....Da como salida el coseno del ángulo 'n'(en grados).

sin n.....Da como salida el seno del ángulo 'n'(en grados).

int n.....Da como salida la parte entera del número 'n'.

quotient n m.....Da el cociente de la división entera de 'n' por 'm'.

random n.....Da un número entero aleatorio no negativo menor que el número 'n'.

remainder n m.....Da el resto de la división entera de 'n' por 'm'.

round n.....Da como salida el número 'n' redondeado al entero más próximo.

+ a b, a+b.....Da la suma de los números 'a' y 'b'.

- a b, a-b.....Da el resultado de restar 'b' de 'a'.

* a b, a*b.....Da el producto de los números 'a' y 'b'.

/a b, a/b.....Da el resultado de dividir 'a' por 'b'.

OPERACIONES LOGICAS

and a b.....Da como salida TRUE si los valores de las expresiones lógicas 'a' y 'b' son ambos verdadero, y FALSE en cualquier otro caso.

not a.....Da como salida TRUE si la expresión lógica 'a' es FALSA y FALSE en caso contrario.

or a b.....Da como salida FALSE si los valores de las expresiones lógicas 'a', 'b',..., son todos falso, y TRUE en cualquier otro caso.

= a b, a=b.....Da como salida TRUE si 'a' es igual a 'b', y FALSE en cualquier otro caso.

>a b, a>b.....Da como salida TRUE si 'a' es mayor que 'b' y FALSE en cualquier otro caso.

CONTROL DE EJECUCION

bye.....Retorna al sistema operativo.

co.....Reanuda la ejecución tras una pausa impuesta por programa.

error.....Da una lista cuyos elementos describen el error más reciente.

go "palabra.....Ejecuta la línea del procedimiento actual que está identificada por la etiqueta palabra.

if a b.....Ejecuta la lista de instrucciones 'b' si la expresión lógica 'a' tiene el valor TRUE. En caso contrario pasa a la siguiente instrucción.

label "palabra.....Identifica una línea para que pueda ser ejecutada por una instrucción go "palabra.

op a.....(output, salida). Convierte el objeto 'a' en la salida del procedimiento y da por terminada la ejecución de éste.

pause.....Suspende la ejecución del procedimiento actual para permitir la interacción con el intérprete o el editor. La ejecución se reanuda con co.

repeat n[lista].....Ejecuta la lista de instrucciones el número de veces especificado por 'n'.

run [lista].....Ejecuta la lista de instrucciones especificada.

stop.....Detiene la ejecución del procedimiento actual.

CAPTACION DE DATOS POR EL TECLADO

.....

rc.....(read character, leer carácter). Da como salida el próximo carácter que se introduzca por el teclado.

rl.....(read list, leer lista). Da como salida una lista que consiste en los próximos caracteres que se introduzcan por el teclado. La entrada termina con un retorno de carro.

rq.....(read quote, leer literalmente). Da como salida una palabra(o sucesión de palabras) que consiste en la línea que se va a introducir por el teclado. La entrada termina con un retorno de carro.

*El lenguaje Dr. Logo dispone de más primitivas que no se han mencionado aquí, las que se han expuesto son las de uso más general y por consiguiente las más utilizadas en este trabajo,

BASE DE DATOS

PARA GRAFICOS

II.1.- Presentación

El acceso a la base de datos se hace a través del módulo "bdatos". Al cargar este módulo pasan al espacio de trabajo los procedimientos 'bd', 'e', y 'p'. Al ejecutarse el procedimiento 'bd' sale en la pantalla el siguiente menú:



(fig. II.1)

Desde este menú podemos acceder a la geometría plana o a la geometría del espacio.

II.1.1.- Módulo principal

El módulo principal es "bdatos". Desde este módulo, y a través de la figura (II.1), accedemos a la base de datos.

El procedimiento 'bd' es el encargado del dibujo del menú de presentación.

Desde este menú podremos escoger una de las dos grandes ramas que componen la base de datos. Una de las ramas es la geometría plana y la otra es la geometría del espacio.

II.1.2.- Procedimientos

Al ejecutarse el procedimiento 'bd' se pasa a la pantalla un dibujo ya realizado y guardado en memoria. En 'bd' se ha utilizado este método para ganar rapidez en su ejecución, este método no es recomendable cuando se dispone de poca memoria, y aunque se disponga de suficiente memoria no se debe utilizar, a no ser que esté justificado el ahorro de tiempo.

Listado del procedimiento 'bd':

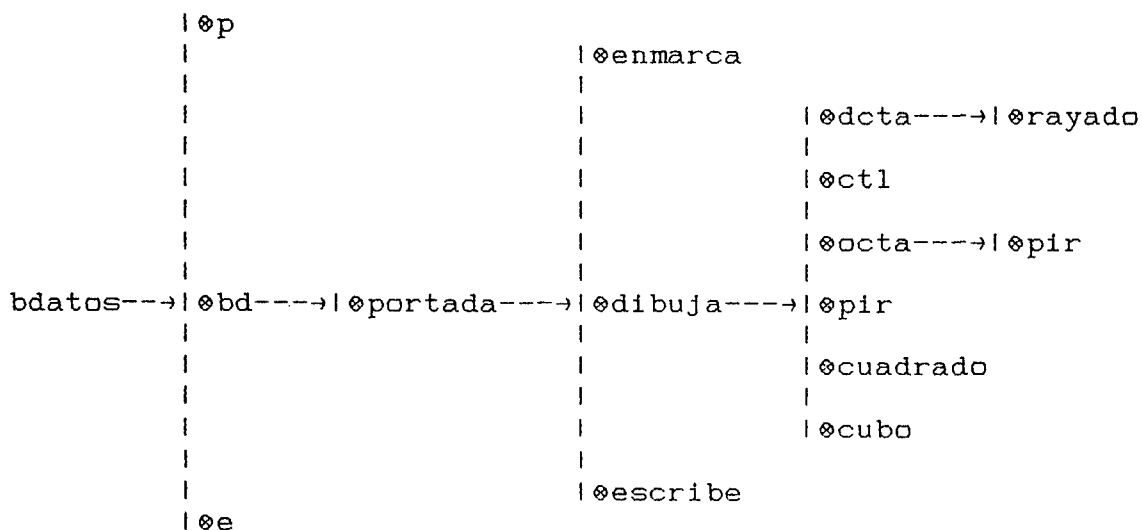
```
to bd
    setsplit 1 ct fs ht
    loadpic "portada
end
```

El procedimiento 'bd', dedica una línea para textos y el resto de la

pantalla mixta a gráficos(setsplit 1), seguidamente limpia la pantalla de texto(ct), dedica toda la pantalla a gráficos(fs), y oculta la tortuga de la pantalla(ht), posteriormente carga el dibujo del menú de presentación en la pantalla(loadpic "portada). El nombre del dibujo ("portada) es el mismo que el nombre del procedimiento que lo realiza, ('portada').

El procedimiento 'portada' no se carga en el espacio de trabajo porque ya se ha utilizado para el dibujo del menú y por lo tanto no es necesario para la operación del programa, aunque sí lo es para su elaboración.

Estructura del módulo "bdatos" y sus procedimientos



(fig. II.2.)

II.1.3.- Listado de procedimientos

to p

```
ct fs erall
load "plana
ts plana
```

end

to e

```
ct fs erall
load "espacio
ts espacio
```

end

to bd

```
setsplit 1 ct fs ht
loadpic "portada
```

end

to portada

```
enmarca
dibuja
escribe
```

end

to enmarca

```
setsplit 1 cs ct ht
pu home setpos [-285 109] pd ctl 565 140 20
pu home setpos [-280 114] pd ctl 555 130 0
pu home setpos [-320 -233] pd ctl 620 320 20
setcursor [33 5] pr [BASE DE DATOS PARA]
setcursor [38 7] pr [GRAFICOS] setcursor [9 11]
pr [PROGRAMA PARA EL CALCULO Y DIBUJO DE FIGURAS PLANAS]
pr [Y TRIDMIENSIONALES] setcursor [28 13]
pr [AUTOR: Jorge Portillo Meniz]
```

```

pu home setpos [-101 123] pd
repeat 77[fd 60 rt 90 pd fd 1 rt 90 px fd 60 lt 90 pd fd 1 lt 90 px]
pu home setpos [-295 55] pd
repeat 286[fd 25 rt 90 pd fd 1 rt 90 px fd 25 lt 90 pd fd 1 lt 90 px]
pu home setpos [-140 20] pd
repeat 111[fd 30 rt 90 pd fd 1 rt 90 px fd 30 lt 90 pd fd 1 lt 90 px]
end

```

to dibuja

```

ct pu home setpos [-190 150]
pd ctl 20 60 10
pu home setpos [195 190]
pd rt 20 octa 45 50 20
pu home setpos [-265 190]
pd ctl 40 18 20
pu home setpos [130 195]
pd cuadrado 30
pu home setpos [-130 190]
pd pir 55 40 20
pu home setpos [30 192]
pd cubo 30 150
pu home setpos [75 130]
pd dcta 100 70 30

```

end

to escribe

```

setcursor [19 17]
pr [PARA MAS INFORMACION DEL DIBUJO Y CALCULO DE LA]
setcursor [19 19]
pr [FIGURA QUE DESEE PULSE LAS TECLAS QUE SE INDICAN]
setcursor [38 22] pr [- - - -]
setcursor [10 25]
pr [1) GEOMETRIA PLANA.....]
setcursor [55 25] pr [ESCRIBA p Y PULSE RETURN.]
setcursor [10 27]
pr [2) GEOMETRIA DEL ESPACIO.....]
setcursor [55 27] pr [ESCRIBA e Y PULSE RETURN.]
setcursor [0 30]

```

end

```

to ctl :x :y :z
  repeat 2[fd :y rt 90 fd :x rt 90] fd :y
  rt 65 fd :z rt 25 fd :x rt 90 fd :y
  rt 65 fd :z rt 115 fd :y rt 65 fd :z
end

to cuadrado :lado
  repeat 4[fd :lado rt 90]
end

to raya :a
  repeat 6[pu fd :a/12 pd fd :a/12]
end

to dcta :a :c :alfa
  make "x :a*cos :alfa
  make "h :a*sin :alfa
  make "y :c-x
  make "beta arctan (:h/:y)
  make "b 90-:alfa
  make "d :h/sin :beta
  rt 90-:alfa fd :a rt :alfa+90 raya :h pu
  bk :h pd lt 90+:alfa rt :alfa+:beta
  fd :d rt 180-:beta fd :x+:y rt 90
end

to cubo :x :ang
  make "y :x make "z :x*sin :ang
  fd :y rt 90 fd :x rt 90 fd :y rt 90 fd :x
  rt 90 fd :y rt 90-:ang fd :z rt :ang fd :x
  rt 90 fd :y rt 90-:ang fd :z rt 90+:ang
  fd :y rt 90-:ang fd :z rt 90+:ang fd :y
  rt 90 fd :x rt 90 fd :y bk :y lt 90+:ang
  fd :z rt 90+:ang
end

```



```

to pir :x :y :ang
  make "alt :y*cos :ang
  make "z :x*sin :ang
  make "b :z*sin :ang
  make "a :z*cos :ang
  make "c (:x+:a)/2
  make "f :alt/:c
  make "e arctan :f
  make "d :alt/sin :e
  rt 90-:ang fd :z rt :ang fd :x rt 180-:ang
  fd :z rt :ang fd :x rt 180-:e fd :d
  make "s :alt-:b
  make "g arctan :s/:c
  make "h :c/cos :g
  rt :e+:g fd :h
  make "i (:x-:a)/2
  make "j arctan :i/:alt
  make "k :alt/cos :j
  bk :h rt 90-:g-:j fd :k
  make "m :alt-:b
  make "n arctan :i/:m
  make "o :i/sin :m
  bk :k rt :j+:n fd :o
  make "q 90-:n-:ang
  rt :q pu fd :z pd rt 90+:ang
end

```

```

to octa :x :y :ang
  pd pir :x :y :ang
  if :ang>0 [make "alfa :ang-180]
  if :ang<0 [make "alfa :ang+180]
  if :ang=0 [make "alfa :ang]
  pir :x :y :alfa
end

```

II.2.-Geometría Plana

Al escribir p y pulsar return, desde el menú principal, se ejecuta el procedimiento 'p', éste procedimiento borra el espacio de trabajo, carga el módulo "plana" y ejecuta el procedimiento 'plana' (ver apartado II.1.3.).

Al ejecutarse el procedimiento 'plana', sale en pantalla el siguiente menú:

GEOMETRIA DEL PLANO

**PARA MAS INFORMACION SOBRE EL DIBUJO Y CALCULO
DE LA FIGURA QUE DESEE PULSE LAS TECLAS SIGUIENTES**

a) TRIANGULOS..... ESCRIBA a Y PULSE RETURN
b) POLIGONOS..... ESCRIBA b Y PULSE RETURN
c) RECTAS..... ESCRIBA c Y PULSE RETURN
d) CURVAS..... ESCRIBA d Y PULSE RETURN

*** SI QUIERE VOLVER AL MENU ANTERIOR ESCRIBA bd Y PULSE RETURN ***

(fig. II.3.)

Desde este menú se tiene acceso al cálculo y dibujo de todas las figuras que se relacionan. Estas figuras están divididas en cuatro grupos y cada uno de estos grupos incluyen una gran variedad de casos particulares.

Al cargarse en el espacio de trabajo el módulo "plana" se cargan los procedimientos 'plana', 'a', 'b', 'c', 'd', 'bd', 'ventana' y 'pit'.

El procedimiento 'ventana' lo utiliza el procedimiento 'plana' para dibujar el marco de la figura II.2.1., el procedimiento 'bd' nos llevará al menú anterior y los procedimientos 'a'..'d' seleccionan un tipo determinado de figura.

.....
Listado de procedimientos - Módulo "plana" -
.....

```
to plana
  ventana setcursor [21 8]
  pr[PARA MAS INFORMACION SOBRE EL DIBUJO Y CALCULO]
  setcursor [19 10]
  pr [DE LA FIGURA QUE DESEE PULSE LAS TECLAS SIGUIENTES]
  setcursor [36 13] pr [ * * * * * ]
  setcursor [12 15] pr[a) TRIANGULOS.....]
  setcursor [52 15] pr[ESCRIBA a Y PULSE RETURN]
  setcursor [12 17] pr[b) CUADRILATEROS.....]
  setcursor [52 17] pr[ESCRIBA b Y PULSE RETURN]
  setcursor [12 19] pr[c) POLIGONOS.....]
  setcursor [52 19] pr[ESCRIBA c Y PULSE RETURN]
  setcursor [12 21] pr[d) RECTAS.....]
  setcursor [52 21] pr[ESCRIBA d Y PULSE RETURN]
  setcursor [12 23] pr[e) CURVAS.....]
  setcursor [52 23] pr[ESCRIBA e Y PULSE RETURN]
  setcursor [13 27]
  pr[*SI QUIERE VOLVER AL MENU ANTERIOR ESCRIBA bd Y PULSE RETURN*]
  setcursor [0 30] pit 2
```

end

```
to a
  erall ct fs
  load "triangulo
  ts triangulo
end
```

```

to pit :n
    repeat :n[type char 7]
end

to b
    erall ct fs
    load "poligono
    poligono
end

to c
    erall ct fs
    load "recta
    recta
end

to d
    erall ct fs
    load "curva
    curva
end

to cti :x :y :z
    repeat 2[fd :y rt 90 fd :x rt 90] fd :y
    rt 65 fd :z rt 25 fd :x rt 90 fd :y
    rt 65 fd :z rt 115 fd :y rt 65 fd :z
end

to bd
    erall ct fs
    load "bdatos
    bd
end

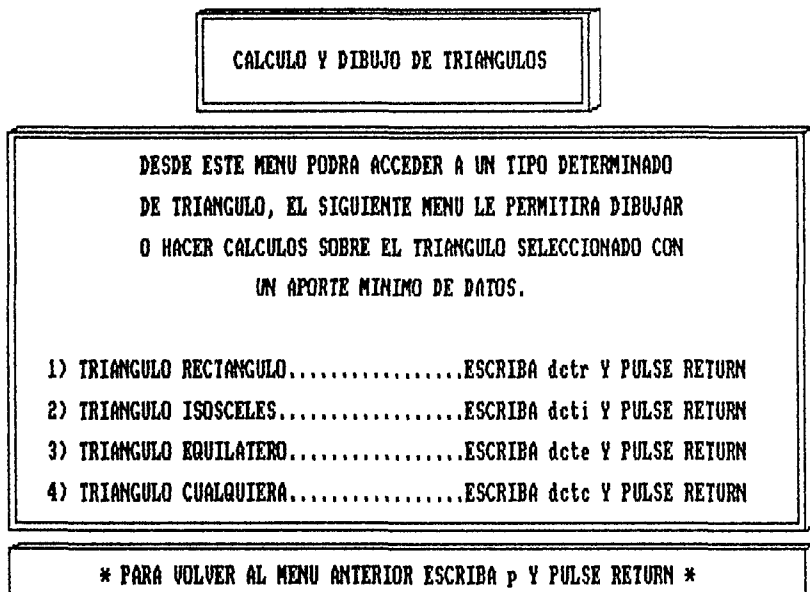
```

```
to ventana
  setsplit 2 ht ts ct pit 1
  pu home setpos [-103 185] pd ctl 195 40 0
  pu home setpos [-108 180] pd ctl 205 50 10
  pu home setpos [-300 -160] pd ctl 600 310 0
  pu home setpos [-305 -165] pd ctl 610 320 10
  pu home setpos [-305 -225] pd ctl 610 40 10
  setcursor [35 3] pr [GEOMETRIA DEL PLANO]
end
```

II.2.1.-Cálculo y dibujo de triángulos

Al seleccionar el apartado (a) de la figura (II.3.) se ejecuta el procedimiento 'a', este procedimiento borra el espacio de trabajo, carga el módulo "triángulo" y ejecuta el procedimiento 'triángulo'. Con este módulo se cargan los procedimientos 'dctr', 'dctc', 'dcti', 'dcte', 'ctl', 'marco', 'triángulo', 'pit' y 'p'.

El procedimiento 'triángulo' presenta el siguiente menú:



(fig. II.4.)

Desde este menú se tiene acceso al cálculo y dibujo de cualquier tipo de triángulo. En los siguientes apartados se explica detalladamente el desarrollo de cada uno de estos casos.

Listado de procedimientos - Módulo "triángulo" -

to triángulo

```
pit 1 marco setcursor [28 3]
pr [CALCULO Y DIBUJO DE TRIANGULOS] setcursor [19 8]
pr[DESDE ESTE MENU PODRA ECCEDER A UN TIPO DETERMINADO]
setcursor [19 10]
pr[DE TRIANGULO, EL SIGUIENTE MENU LE PERMITIRA DIBUJAR]
setcursor [19 12]
pr[O HACER CALCULOS SOBRE EL TRIANGULO SELECCIONADO CON]
setcursor [30 14] pr[UN APORTE MINIMO DE DATOS.]
setcursor [10 18] pr[1) TRIANGULO RECTANGULO.....]
setcursor [50 18] pr[ESCRIBA dctr Y PULSE RETURN]
setcursor [10 20] pr[2) TRIANGULO ISOSCELES.....]
setcursor [50 20] pr[ESCRIBA dcti Y PULSE RETURN]
setcursor [10 22] pr[3) TRIANGULO EQUILATERO.....]
setcursor [50 22] pr[ESCRIBA dcte Y PULSE RETURN]
setcursor [10 24] pr[4) TRIANGULO CUALQUIERA.....]
setcursor [50 24] pr[ESCRIBA dctc Y PULSE RETURN]
setcursor [15 28]
pr[*PARA VOLVER AL MENU ANTERIOR ESCRIBA p Y PULSE RETURN*]
setcursor [0 30] pit 2
```

end

to marco

```
ct clean setsplit 1
pu home setpos [-160 168] pd ctl 290 70 0
pu home setpos [-165 163] pd ctl 300 80 10
pu home setpos [-310 -185] pd ctl 610 330 10
pu home setpos [-305 -180] pd ctl 600 320 0
pu home setpos [-310 -240] pd ctl 610 40 10
```

end

to dctr

```
erall fs
load "dctr dctr"
```

end

to dctc

```
erall ct fs
load "dctc dctc"
```

end

```

to dcti
  erall ct fs
  load "dcti dcti
end

to dcte
  erall ct fs
  load "dcte dcte
end

to ctl :x :y :z
  repeat 2[fd :y rt 90 fd :x rt 90] fd :y
  rt 65 fd :z rt 25 fd :x rt 90 fd :y
  rt 65 fd :z rt 115 fd :y rt 65 fd :z
end

to pit :n
  repeat :n[type char 7]
end

to p
  erall fs
  load "plana plana
end

```


II.2.1.1.- Triángulo rectángulo

Al seleccionar el apartado (1) de la figura (II.4.), se ejecuta el procedimiento 'dctr' del módulo "triángulo".

```
Listado del procedimiento 'dctr':  to dctr
                                   erall fs
                                   load "dctr dctr
                                   end
```

Al cargarse el módulo "dctr" en el espacio de trabajo también se cargan los procedimientos 'ttr', 'dtr', 'ctr', 't', 'recuadro', 'dctr', 'salida', 'texto', 'ctl' y 'pit'.

La siguiente instrucción del procedimiento 'dctr', perteneciente al módulo "triangulo", es la de ejecutar el procedimiento 'dctr', esta vez el perteneciente al módulo "dctr".

DIBUJO DEL TRIANGULO RECTANGULO

PARA DIBUJAR UN TRIANGULO RECTANGULO ESCRIBA dtr Y A
CONTINUACION EL VALOR DE LA HIPOTENUSA Y EL ANGULO DE LA BASE.

EJEMPLO: ESCRIBA es..dtr 100 45 Y PULSE RETURN

EL PROGRAMA DIBUJARA UN TRIANGULO DE HIPOTENUSA = 100 Y EL
ANGULO DE LA BASE 45 °, LOS OTROS DOS LADOS Y ANGULOS LOS
CALCULA EL PROGRAMA ANTES DE HACER EL DIBUJO.

PARA EL CALCULO ESCRIBA ctr Y PULSE RTN, PARA EL MENU ANTERIOR t Y PULSE RTN

El procedimiento
'dctr' presenta en
pantalla el menú
de la fig. II.5.

(fig. II.5.)

El procedimiento 'dctr' utiliza para su ejecución, tres procedimientos previamente definidos, 'recuadro', 'texto' y 'salida'.

Desde la figura (II.5.) tenemos acceso al dibujo de triángulos rectángulos 'dtr', al menú anterior 't' y al módulo de cálculo de triángulos rectángulos 'ctr'.

Si elegimos la opción del cálculo, al escribir ctr y pulsar return se ejecuta el procedimiento 'ctr' del módulo "dctr", que es el que se encuentra en el espacio de trabajo hasta este momento.

Este procedimiento borra el espacio de trabajo y carga el módulo "ctr". Al cargarse este módulo se cargan los trece procedimientos que lleva en su declaración, estos procedimientos son: 'ttr', 'ctr', 'tr1', 'dctr', 'tr2', 'texto', 'variables', 'dibujo.2', 'ctl', 'pit', 'ventana', 'ventanal' y 'medidas'.

En este módulo también hay un procedimiento que se llama 'dctr', al igual que en el módulo "dctr" y en el módulo "triangulo". Aunque los tres tienen el mismo nombre, cada uno hace una labor diferente, ya que nunca están en el espacio de trabajo al mismo tiempo. El procedimiento 'dctr' del módulo "triangulo" sirve para trasladarnos al módulo "dctr", si escogemos la opción (1) de la figura (II.4.). El procedimiento 'dctr' del módulo "dctr" es el encargado de presentar en pantalla la figura (II.5.). El procedimiento 'dctr' perteneciente al módulo "ctr" nos lleva de vuelta al módulo "dctr".

Si tenemos el módulo "ctr" en el espacio de trabajo, al ejecutarse el procedimiento 'ctr' sale en pantalla el menú de la figura (II.6.).

Desde este menú tenemos acceso al cálculo de un triángulo rectángulo en los supuestos (1) y (2).

CALCULO DEL TRIANGULO RECTANGULO

PARA CALCULAR LAS VARIABLES DE UN TRIANGULO RECTANGULO A PARTIR DE UNOS DATOS MINIMOS, PULSE LAS TECLAS QUE SE INDICAN

- 1) UN CATETO Y UN ANGULO.....ESCRIBA tr1 (cateto) (angulo) Y PULSE RETURN
- 2) LA HIPOTENUSA Y UN ANGULO.....ESCRIBA tr2 (hipotenusa) (angulo) Y PULSE RETURN

POR EJEMPLO: CONOCE LA HIPOTENUSA (250) Y UN ANGULO (40 °) , (Caso 2 °) , EN ESTE SUPUESTO ESCRIBA..tr2 250 40 ..Y PULSE RETURN, A CONTINUACION SE BORRA LA PANTALLA Y EL PROGRAMA DIBUJARA EL TRIANGULO AL MISMO TIEMPO QUE DA LOS VALORES DE TODOS LOS PARAMETROS QUE DEFINEN ESE TRIANGULO

SI QUIERE VOLVER AL MENU ANTERIOR ESCRIBA dctr Y PULSE RETURN

El procedimiento 'ctr' presenta en pantalla el menú de la fig. II.6.

(fig. II.6.)

Si escogemos el supuesto (1), conocemos un cateto y un ángulo del triángulo, los cálculos los hace el procedimiento 'tr1'.

Durante la ejecución de 'tr1' se ejecutan los procedimientos previamente definidos, 'ventanal' y 'ttr'.

El procedimiento 'ttr' usa, a su vez, los procedimientos previamente definidos, 'variables' y 'medidas', 'ventanal' usa, en su ejecución el procedimiento previamente definido 'ventana' y este último utiliza 'pit'.

Por lo tanto el diagrama del procedimiento 'tr1' es:

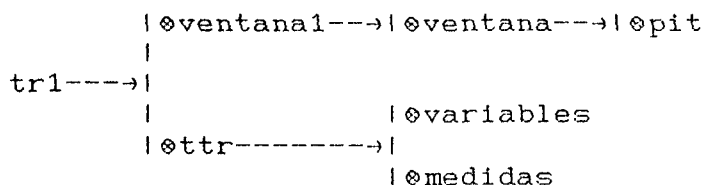


fig. (II.7.)

Al escribir tr1 con dos entradas y pulsar return, el procedimiento comprueba si el ángulo(2ª entrada) es mayor o igual a 90º, en este supuesto se indica al usuario del error y se suspende la ejecución del procedimiento. Si el ángulo es menor de 90º se ejecuta el procedimiento 'ventanal'.

El procedimiento 'ventanal', ejecuta el procedimiento 'ventana' y éste, a su vez, ejecuta el procedimiento 'pit', el procedimiento 'pit' emite un bip indicando que el programa ha comenzado a ejecutarse. Después del bip, el procedimiento 'ventana', divide la pantalla en tres apartados, mediante delimitadores gráficos. En este momento se concluye la ejecución del procedimiento 'ventana' y continua la ejecución del procedimiento 'ventanal'. El resto del procedimiento 'ventanal', dibujará un triángulo genérico en el apartado izquierdo, en éste momento concluye la

ejecución del procedimiento 'ventana1' y comienza a ejecutarse el procedimiento 'ttr'.

El procedimiento 'ttr' lleva dos entradas, la segunda entrada es la misma que tenía 'tr1', pero la primera entrada es la hipotenusa del triángulo y no un cateto como era en 'tr1', el cambio se hizo en el cuerpo de 'tr1', antes de ejecutar el procedimiento 'ventana1'. El motivo por el que se ha hecho de esta manera es para poder volver a utilizar este procedimiento que fué declarado en el módulo anterior, "dctr".

El procedimiento 'ttr', primero ejecuta el procedimiento 'variables', con este procedimiento se hallan todas las variables del triángulo objeto del cálculo y se almacenan en memoria. Una vez terminado el procedimiento 'variables', se ejecuta el procedimiento 'medidas', este procedimiento toma las medidas del triángulo objeto del cálculo y las reduce o aumenta de manera que encajen en el apartado derecho de la pantalla, reservado para su representación, además este procedimiento coloca la figura exactamente en el centro de este apartado.

Para terminar, el procedimiento 'tr1', dá el valor de todas las variables del triángulo en el apartado izquierdo de la pantalla y después de sonar dos bips concluye la ejecución de 'tr1'.

Si en la figura (II.6.) escogemos el supuesto (2), conocemos la hipotenusa y un ángulo, el proceso es similar al anterior. En este supuesto el procedimiento encargado del cálculo y dibujo es el 'tr2'.

II.2.1.2.- Ejemplos de aplicación y listado de procedimientos

Ejemplos de aplicación -(Dibujo de triángulos)-

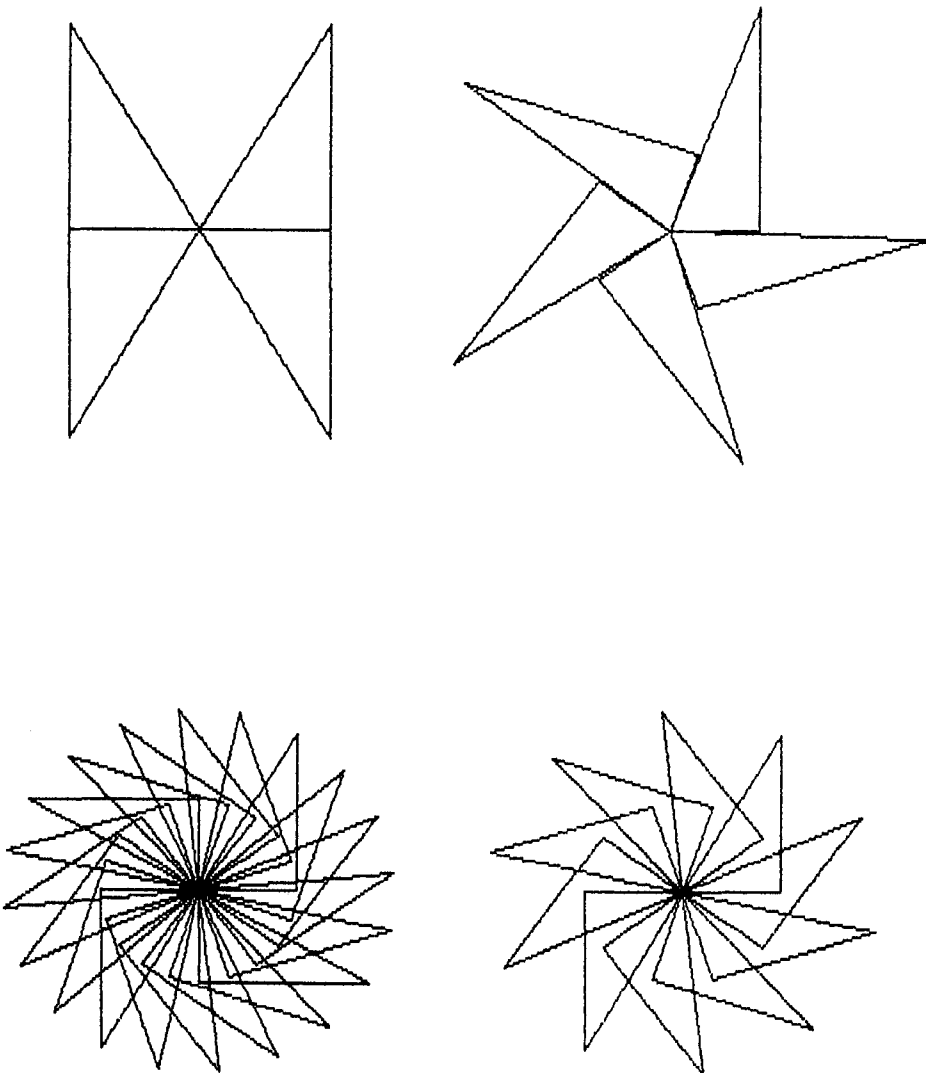
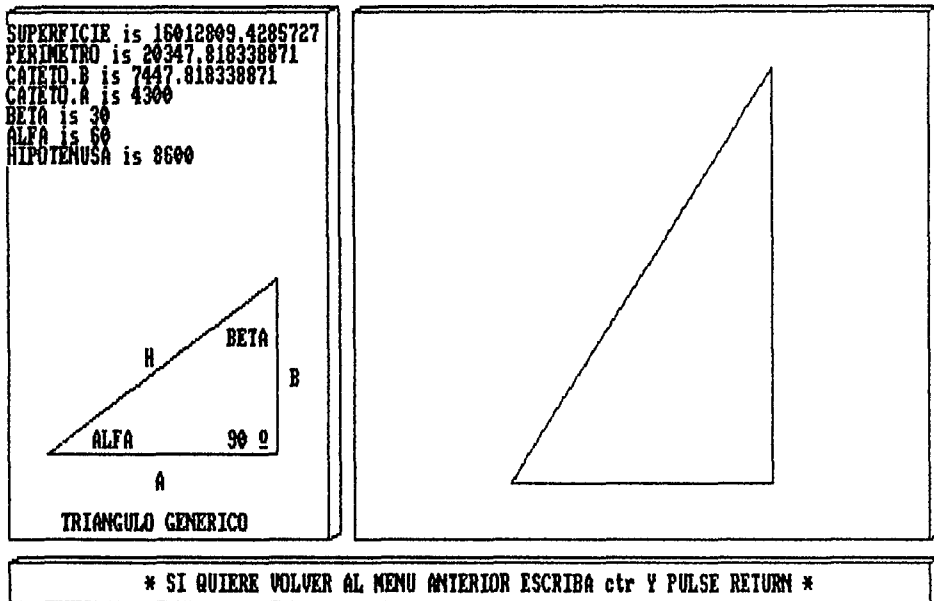


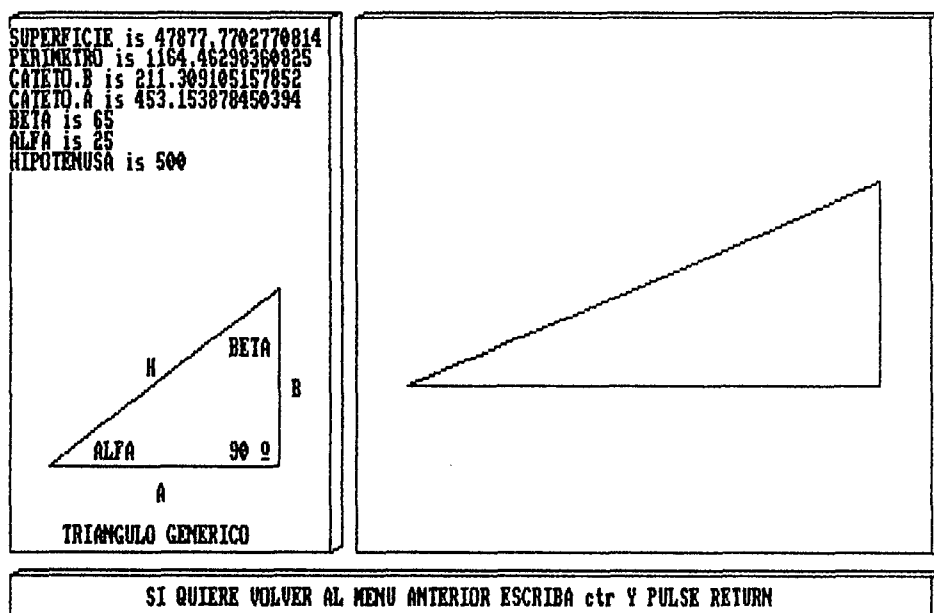
fig. (II.8.)

Ejemplos de aplicación -(Cálculo de triángulos)-



Ejemplo del caso
(1), con 'tr1'.

fig. (II.9.)



Ejemplo del caso
(2), con 'tr2'.

fig. (II.10.)

Listado de procedimientos -Módulo "dctr"-

```
to ttr :dato :ang
    setsplit 3 ht
    pd rt 90-:ang
    fd :dato rt 90+:ang
    fd :dato*sin :ang rt 90
    fd :dato*cos :ang rt 90
end
```

```
to dtr :dato :ang
    pu ts ct setsplit 3 ht
    pd rt 90-:ang
    fd :dato rt 90+:ang
    fd :dato*sin :ang rt 90
    fd :dato*cos :ang rt 90 setcursor [20 27]
    pr[SI QUIERE VOLVER AL MENU ANTERIOR ESCRIBA dctr Y PULSE RETURN]
    setcursor [17 29]
    pr[SI QUIERE EL DIBUJO SIN LEYENDAS ESCRIBA ttr EN VEZ DE DTR]
end
```

```
to ctr
    erall ct fs
    load "ctr ts ctr"
end
```

```
to t
    erall fs
    load "triangulo triangulo"
end
```

```
to recuadro
    ct clean ht pit 1
    pu home setpos [-155 190] pd ctl 295 60 10 25
    pu home setpos [-325 -145] pd ctl 655 325 10 25
    pu home setpos [-325 -225] pd ctl 655 70 10 25
end
```



```

to dctr
  recuadro
  texto
  salida
end

to salida
  setcursor [7 26]
  pr[PARA EL CALCULO ESCRIBA ctr Y PULSE RETURN, PARA MENU ANTERIOR t Y PULSE
RETURN]
  setcursor [0 30] pit 2
end

to texto
  setcursor [29 2]
  pr[DIBUJO DEL TRIANGULO RECTANGULO]
  setcursor [19 6]
  pr[PARA DIBUJAR UN TRIANGULO RECTANGULO ESCRIBA dtr Y A]
  setcursor [14 8]
  pr[CONTINUACION EL VALOR DE LA HIPOTENUSA Y EL ANGULO DE LA BASE.]
  setcursor [22 11]
  pr[EJEMPLO: ESCRIBA cs..dtr 100 45 Y PULSE RETURN]
  setcursor [16 16]
  pr[EL PROGRAMA DIBUJARA UN TRIANGULO DE HIPOTENUSA = 100 Y EL]
  setcursor [16 18]
  pr[ANGULO DE LA BASE 45º, LOS OTROS DOS LADOS Y ANGULOS LOS]
  setcursor [22 20]
  pr[CALCULA EL PROGRAMA ANTES DE HACER EL DIBUJO.]
end

to ctl :x :y :z :ang
  repeat 2[fd :y rt 90 fd :x rt 90] fd :y
  rt 90-:ang fd :z rt :ang fd :x rt 90 fd :y
  rt 90-:ang fd :z rt 90+:ang fd :y rt 90-:ang fd :z
end

to pit :n
  repeat :n[type char 7]
end

```

Listado de procedimientos - Módulo "ctr" -

```
to tr1 :a :b
  if (:b<90) [go "10]
  ts ct setcursor [25 16]
  pr[EL ANGULO NO PUEDE SER MAYOR O IGUAL A 90º] stop
label "10 make "dato :a/cos :b
  make "alfa :b
  ventanal ttr :dato :alfa
  ern[C1 C2 dato alfa dato.d]
  setcursor [0 1] pons
  setcursor [13 29]
  pr[SI QUIERE VOLVER AL MENU ANTERIOR ESCRIBA ctr Y PULSE RTN.]
  recycle setcursor [0 30] pit 2
end
```

```
to ventanal
  ventana pu home setpos [-330 -115] pd
  rt 50 fd 230 rt 130 fd 230*sin 40
  rt 90 fd 230*cos 40 rt 90
  setcursor [8 21] pr [ALFA]
  setcursor [21 16] pr [BETA]
  setcursor [21 21] pr [90º]
  setcursor [14 23] pr [A]
  setcursor [13 17] pr [H]
  setcursor [27 18] pr [B]
  setcursor [5 25] pr [TRIANGULO GENERICO]
end
```

```
to ventana
  cs ht setsplit 1 pit 1
  pu home setpos [-360 -185] pd ctl 245 440 10
  pu home setpos [-95 -185] pd ctl 440 440 10
  pu home setpos [-360 -240] pd ctl 705 35 10
end
```

```
to pit :n
  repeat :n[type char 7]
end
```

```

to ttr :dato :alfa
    variables medidas
    pd rt 90-:alfa fd :dato.d rt 90+:alfa
    fd (:dato.d*sin:alfa)
    rt 90 fd (:dato.d*cos :alfa) rt 90
end

to variables
    make "ALFA :alfa
    make "BETA 90-:alfa
    make "CATETO.A :dato*cos :alfa
    make "CATETO.B :dato*sin :alfa
    make "HIPOTENUSA :dato
    make "SUPERFICIE (:CATETO.A*:CATETO.B)/2
    make "PERIMETRO :HIPOTENUSA+:CATETO.A+:CATETO.B
end

to medidas
    make "dato.d 400
    make "C1 220-(:dato.d*cos :alfa)/2
    make "C2 220-(:dato.d*sin :alfa)/2
    pu setx :C1-95 sety :C2-185 pd
end

to tr2 :dato :alfa
    if (:alfa<90) [go "10]
    clean setcursor [20 15]
    pr[EL ANGULO NO PUEDE SER MAYOR O IGUAL A 90º] stop
label "10 ventanal variables
    medidas dibujo.2 ern [C1 C2 dato.d]
    setcursor [0 1] pons setcursor [13 28]
    pr[SI QUIERE VOLVER AL MENU ANTERIOR ESCRIBA ctr Y PULSE RTN.]
    setcursor [0 30] recycle pit 2
end

to dibujo.2
    rt 90-:alfa fd :dato.d rt 90+:alfa fd (:dato.d*sin :alfa)
    rt 90 fd (:dato.d*cos :alfa) rt 90
end

```

```

to clt :x :y :z
    repeat 2[fd :y rt 90 fd :x rt 90] fd :y
    rt 65 fd :z rt 25 fd :x rt 90 fd :y
    rt 65 fd :z rt 115 fd :y rt 65 fd :z
end

```

```

to ctr
    cs pit 1
    pu home setpos [-130 200] pd ctl 270 45 10
    pu home setpos [-335 -165] pd ctl 685 355 10
    pu home setpos [-335 -225] pd ctl 685 45 10
    texto
end

```

```

to texto
setcursor[30 2] pr[CALCULO DEL TRIANGULO RECTANGULO] setcursor [14 6]
pr[PARA CALCULAR LAS VARIABLES DE UN TRIANGULO RECTANGULO A PARTIR]
setcursor [19 8]
pr[DE UNOS DATOS MINIMOS, PULSE LAS TECLAS QUE SE INCICAN]
setcursor [6 11]
pr[1ø UN CATETO Y UN ANGULO.....] setcursor [39 11]
pr[ESCRIBA tr1 (cateto) (ángulo) Y PULSE RETURN] setcursor [6 13]
pr[2) LA HIPOTENUSA Y UN ANGULO.....] setcursor [39 13]
pr[ESCRIBA tr2 (hipotenusa) (ángulo) Y PULSE RETURN]
setcursor [9 17]
pr[POR EJEMPLO: CONOCE LA HIPOTENUSA (250) Y UN ANGULO (40º), (Caso 2º), EN]
setcursor [9 19]
pr[ESTE SUPUESTO ESCRIBA..tr2 250 40 Y PULSE RETURN, A CONTINUACION SE
BORRA] setcursor [9 21]
pr[LA PANTALLA Y EL PROGRAMA DIBUJARA EL TRIANGULO AL MISMO TIEMPO QUE DAI]
setcursor [17 23]
pr[LOS VALORES DE TODOS LOS PARAMETROS QUE DEFINEN EL TRIANGULO]
setcursor [15 27]
pr[SI QUIERE VOLCVER AL MENU ANTERIOR ESCRIBA dctr Y PULSE RETURN]
setcursor [0 30] pit 2
end

```

```

to dctr
    erall ct fs load "dctr dctr
end

```

II.2.1.3.- Triángulo isósceles

Al seleccionar el apartado (2) de la figura (II.4.), se ejecuta el procedimiento 'dcti' del módulo "triángulo".

```
Listado del procedimiento 'dcti':  to dcti
                                     erall fs
                                     load "dcti dcti
                                     end
```

Al cargarse el módulo "dcti" en el espacio de trabajo también se cargan los procedimientos 'dti', 'recuadro', 'dcti', 'cti', 't', 'dts', 'ctl' y 'pit'.

la siguiente instrucción del procedimiento 'dcti', perteneciente al módulo "triángulo", es la de ejecutar el procedimiento 'dcti', perteneciente al módulo "dcti".

DIBUJO DEL TRIANGULO ISOSCELES

PARA DIBUJAR UN TRIANGULO ISOSCELES TIENE QUE ESCRIBIR
dcti Y A CONTINUACION DAR LOS DATOS DEL TRIANGULO

EJEMPLO: ESCRIBA es..dcti 200 150 Y PULSE RETURN

EL PROGRAMA DIBUJARA UN TRIANGULO ISOSCELES DE 200 DE ALTURA
Y 150 DE BASE , LOS OTROS DOS LADOS Y ANGULOS LOS CALCULA EL
PROGRAMA ANTES DE DIBUJAR EL TRIANGULO

* PARA EL CALCULO ESCRIBA cti Y PULSE RETURN *
* PARA EL MENU ANTERIOR ESCRIBA t Y PULSE RETURN *

El procedimiento
'dcti' presenta en
pantalla el menú
de la fig. II.11.

(fig. II.11.)

Desde la figura (II.11.) tenemos acceso al dibujo de triángulos isósceles 'dti', al menú anterior 't' y al módulo de cálculo de triángulos isósceles 'cti'.

Si elegimos la opción del módulo anterior, al escribir t y pulsar return, se ejecuta el procedimiento 't'. El procedimiento 't', borra el espacio de trabajo, carga el módulo "triángulo", ejecuta el procedimiento 'triángulo' y presenta en pantalla la figura (II.4).

Si elegimos la opción del dibujo podremos dibujar cualquier triángulo isósceles o cualquier tipo de figura realizada con esa figura patrón. Para operar con buen rendimiento esta parte del programa es necesario estar familiarizado con las primitivas básicas de Dr. Logo. Sólo en el dibujo de figuras será necesario que el usuario de la base de datos conozca las instrucciones más importantes de Logo. En el resto del uso de esta base de datos no será necesario que el usuario tenga conocimientos de Logo.

Si elegimos la opción de cálculo, al escribir cti y pulsar return se ejecuta el procedimiento 'cti' del módulo "dcti", que es el que se encuentra en el espacio de trabajo en este momento.

Este procedimiento borra el espacio de trabajo y carga el módulo "cti". Al cargarse este módulo se cargan los quince procedimientos que lleva en su declaración. Estos procedimientos son: 'marco', 'cti', 'dcti', 'rayado', 'iso', 'tb2', 'tb1', 'final', 'ctl', 'medidas', 'variables', 'dibujo', 'pit', 'ventana' y 'ventana1'.

Si tenemos el módulo "cti" en el espacio de trabajo, al ejecutarse el procedimiento 'cti', sale en pantalla el menú de la figura (II.12.).

Desde este menú tenemos acceso al cálculo de un triángulo isósceles en los supuestos (1) y (2).

CALCULO DEL TRIANGULO ISOSCELES

PARA CALCULAR LAS VARIABLES DE UN TRIANGULO ISOSCELES A PARTIR DE UNOS DATOS MINIMOS, PULSE LAS TECLAS QUE SE INDICAN

- 1) LA BASE Y ANGULO OPUESTO.....ESCRIBA th1 (base) (angulo) Y PULSE RETURN
- 2) LA BASE Y ANGULO CONTIGUO.....ESCRIBA th2 (base) (angulo) Y PULSE RETURN

POR EJEMPLO: SI CONOCE LA BASE.. (250) Y EL ANGULO CONTIGUO... (40 °) (Caso 2 °) , EN ESTE CASO ESCRIBA...th2 250 40Y PULSE RETURN , A CONTINUACION SE BORRA LA PANTALLA Y EL PROGRAMA DIBUJARA EL TRIANGULO AL MISMO TIEMPO QUE DA LOS VALORES DE TODAS LAS VARIABLES

SI QUIERE VOLVER AL MENU ANTERIOR ESCRIBA dcti Y PULSE RETURN

El procedimiento 'cti' presenta en pantalla el menú de la fig. II.12.

(fig. II.12.)

Si escogemos el supuesto (1), conocemos la base y el ángulo opuesto, los cálculos los hace el procedimiento 'tb1'.

Durante la ejecución de 'tb1' se ejecutan los procedimientos previamente definidos, 'ventana1', 'variables', 'medidas', 'dibujo', 'final' y 'pit'.

El procedimiento 'tb1' comienza ejecutando el procedimiento 'ventana1', este procedimiento divide la pantalla en tres apartados mediante delimitadores gráficos, seguidamente se ejecuta 'variables', este procedimiento halla todas las variables del triángulo objeto del cálculo y las almacena en memoria.

Una vez terminado el procedimiento 'variables', se ejecuta el procedimiento 'medidas'. Este procedimiento toma las medidas del triángulo objeto del cálculo y las reduce o aumenta de manera que encajen en el apartado derecho de la pantalla reservado para su representación. Este procedimiento hace que la figura quede exactamente en el centro.

Por último se ejecuta el procedimiento 'dibujo', este procedimiento dibuja la figura objeto del cálculo.

Para terminar, el procedimiento 'final', da el valor de todas las variables del triángulo en el apartado izquierdo de la pantalla y después de sonar dos bips termina la ejecución de 'tb1'.

Si en la figura (II.12.) escogemos el supuesto (2), conocemos la base y el ángulo contiguo, el proceso es similar al anterior. En éste supuesto el procedimiento encargado del cálculo y dibujo es el 'tb2'.

La única diferencia entre 'tb2' y 'tb1' es que en 'tb2' tenemos que comprobar si el ángulo es mayor de 90° , en cuyo caso, se informa al usuario del error y se concluye el programa. Si no es mayor de 90° , se halla el ángulo opuesto a la base y una vez hallado éste ángulo, se trata el procedimiento como si fuera 'tb1' .

II.2.1.4.- Ejemplos de aplicación y listado de procedimientos

Ejemplos de aplicación - (Dibujo de triángulos)-

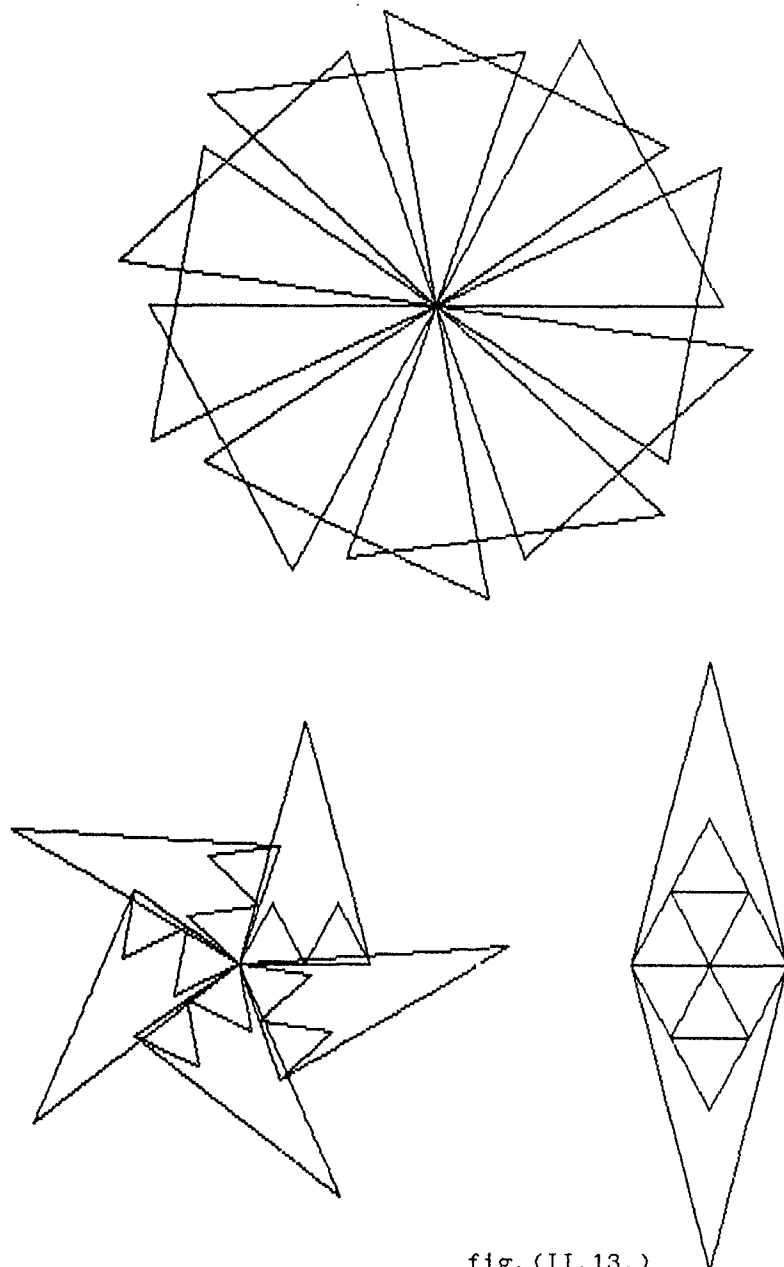
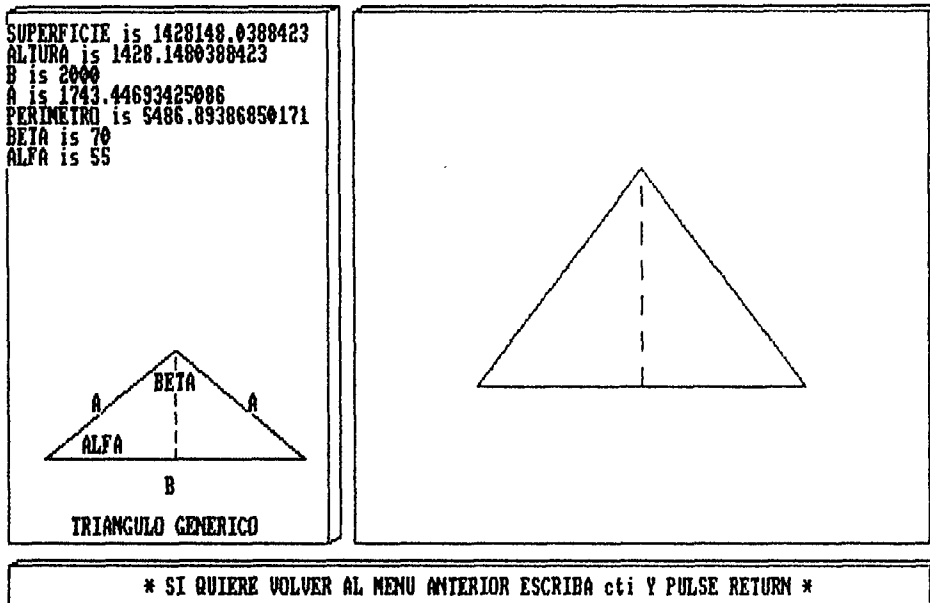


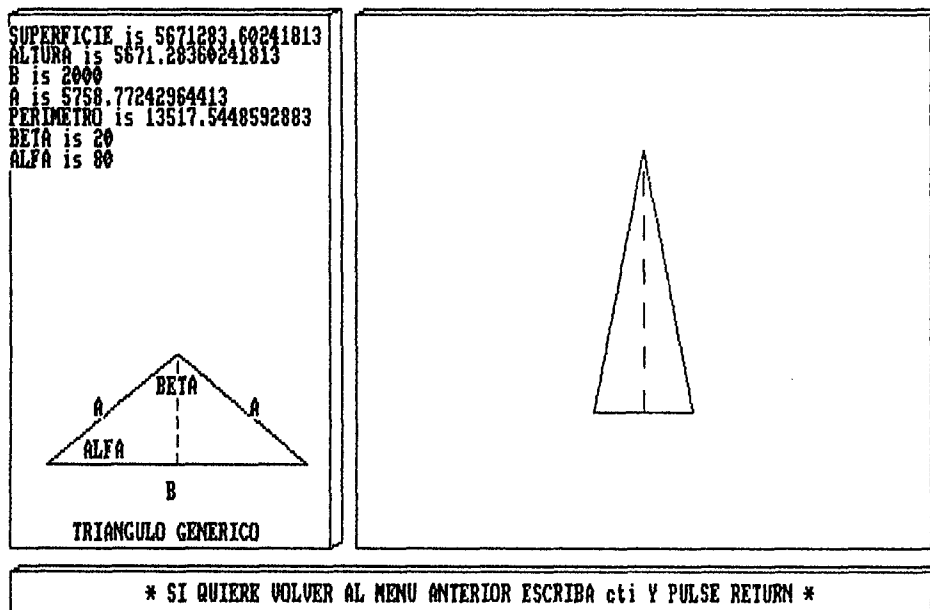
fig. (II.13.)

Ejemplos de aplicación -(Cálculo de triángulos)-



Ejemplo con
'tb1'.

fig. (II.14.)



Ejemplo con
'tb2'.

fig. (II.15.)

Listado de procedimientos - Módulo "dcti"-

```
to dcti
  recuadro setcursor [30 2]
  pr[DIBUJO DEL TRIANGULO ISOSCELES] setcursor [19 6]
  pr[SI QUIERE DIBUJAR UN TRIANGULO ISOSCELES TIENE QUE ESCRIBIR]
  setcursor [22 8]
  pr[dti Y A CONTINUACION DAR LOS DATOS DEL TRIANGULO]
  setcursor [22 11]
  pr [EJEMPLO: ESCRIBA cs..dti 200 150 Y PULSE RETURN]
  setcursor [15 14]
  pr[EL PROGRAM,A DIBUJARA UN TRIANGULO ISOSCELES DE 200 DE ALTURA]
  setcursor [15 16]
  pr[Y 150 DE BASE, LOS OTROS DOS LADOS Y ANGULOS LOS CALCULA EL]
  setcursor [26 18]
  pr[PROGRAMA ANTES DE DIBUJAR EL TRIANGULO]
  setcursor [22 23]
  pr[*PARA EL CALCULO ESCRIBA cti Y PULSE RETURN*]
  setcursor [20 25]
  pr[*PARA EL MENU ANTERIOR ESCRIBA t Y PULSE RETURN*]
  setcursor [0 30] pit 2
end
```

```
to cti
  erall ct fs
  load "cti ts cti"
end
```

```
to t
  erall ct fs
  load "triángulo triángulo"
end
```

```
to recuadro
  ct clean ht setsplit 2 pit 1
  pu home setpos [-140 200] pd cti 295 45 10
  pu home setpos [-290 -105] pd cti 600 290 10
  pu home setpos [-290 -190] pd cti 600 70 10
end
```

```

to cti :x :y :z
    repeat 2[fd :y rt 90 fd :x rt 90] fd :y
    rt 65 fd :z rt 25 fd :x rt 90 fd :y
    rt 65 fd :z rt 115 fd :y rt 65 fd :z
end

```

```

to pit :n
    repeat :n type char 7]
end

```

```

to dts :a :b
    make "ALFA arctan :b/(2*:a)
    make "c :a/cos :ALFA
    make "BETA 90-:ALFA
    rt 90-:BETA fd :c rt 2*:BETA fd :c
    rt 90+:ALFA fd :b rt 90
end

```

```

to dti :a :b
    dts :a :b
    setcursor [16 28]
    pr[PARA VOLVER AL MENU ANTERIOR ESCRIBA dcti Y PULSE RTN]
    setcursor [15 29]
    pr[SI QUIERE EL DIBUJO SIN LEYENDAS ESCRIBA dts EN VEZ DE dti]
    setcursor [0 30]
end

```

.....
Listado de procedimientos - Módulo "cti" -


```

to cti
    (Igual que 'cti' del módulo "dcti")
end

```

```

to pit
    (Igual que 'pit' del módulo "dcti")
end

```

```

to ventanal
    ventana pu home setpos [-332 -115] pd
    iso 200 42
    setcursor [14 18] pr [BETA]
    setcursor [7 21] pr [ALFA]
    setcursor [15 23] pr [B]
    setcursor [8 19] pr [A]
    setcursor [23 19] pr [A]
    setcursor [6 25] pr [TRIANGULO GENERICO]
end

```

```

to ventana
    cs ht setsplit 1 pit 1
    pu home setpos [-360 -185] pd ctl 245 440 10
    pu home setpos [-95 -185] pd ctl 440 440 10
    pu home setpos [-360 -240] pd ctl 705 35 10
end

```

```

to variables
    make "B :b
    make "A :B/(2*cos :ALFA)
    make "ALTURA :A*sin :ALFA
    make "SUPERFICIE (:B*:ALTURA)/2
    make "PERIMETRO :A+:A+:B
end

```

```

to dcti
    erall ct fs
    load "dcti dcti
end

```

```

to cti
    marco texto setcursor [14 27]
    pr[SI QUIERE VOLVER AL MENU ANTERIOR ESCRIBA dcti Y PULSR RTN]
    setcursor [0 30] pit 2
end

```

```

to rayado :a
    repeat 6[pu fd (:a/12) pd fd (:a/12) pd]
end

```

```

to marco
    ct cs ht setsplit 2 pit 1 setcursor [30 2]
    pr[CALCULO DEL TRIANGULO ISOSCELES] pu home setpos [-130 202]
    pd ctl 265 38 12 pu home setpos [-335 -165] pd ctl 680 355 12
    pu home setpos [-335 -225] pd ctl 680 45 12
end

```

```

to texto
    setcursor [10 5]
    pr[PARA CALCULAR LAS VARIABLES DE UN TRIANGULO ISOSCELES A PARTIR DE UNOS]
    setcursor [22 7]
    pr[DATOS MINIMOS, PULSE LAS TECLAS QUE SE INDICAN]
    setcursor [8 10]
    pr[1) LA BASE Y ANGULO OPUESTO.....]
    setcursor [41 10]
    pr[ESCRIBA tb1 (base) (angulo) Y PULSE RETURN]
    setcursor [8 12]
    pr[2) LA BASE Y ANGULO CONTIGUO.....]
    setcursor [41 12]
    pr[ESCRIBA tb2 (base) (angulo) Y PULSE RETURN]
    setcursor [10 16]
    pr[EJEMPLO: SI CONOCE LA BASE.. (250) Y EL ANGULO CONTIGUO...(40º)]
    setcursor [10 18]
    pr[(Caso 2º), EN ESTE CASO ESCRIBA ..tb2 250 40... Y PULSE RETURN, A]
    setcursor [10 20]
    pr[CONTINUACION SE BORRA LA PANTALLA Y EL PROGRAMA DIBUJARA EL TRIANGULO]
    setcursor [16 22]
    pr[AL MISMO TIEMPO QUE DA LOS VALORES DE TODAS LAS VARIABLES]
end

```

```

to final
    setcursor [0 1] pons setcursor [13 28]
    pr[*SI QUIERE VOLVER AL MENU ANTERIOR ESCRIBA cti Y PULSE RTN*]
    setcursor [0 30] pit 2
end

```

```

to iso :b :al
    make "a :b/(2*cos :al)
    make "h :a*sin :al
    make "be 180-2*:al
    rt 90-:al fd :a rt 2*:al+:be/2 rayado :h
    pu bk :h pd lt :be/2 fd :a rt 180-:al fd :b
    rt 90 ern [a h be]
end

```

```

to tbi :b :beta
    make "BETA :beta make "ALFA (180-:BETA)/2
    ventanal variables medidas dibujo final
end

```

```

to tb2 :b :alfa
    if (:alfa<90) [go "1] cs setcursor [25 15] pit 1
    pr[EL ANGULO CONTIGUO DE UN TRIANGULO] setcursor [33 17]
    pr[NO PUEDE SER MAYOR DE 90º] setcursor [0 30] pit 2 stop
label "1 make "ALFA :alfa make "BETA 180-2*:ALFA
    ventanal variables medidas dibujo final
end

```

```

to medidas
    make "M 400 if :A>ALTURA [make "Co :M/:A go "1]
    make "Co :M/:ALTURA
label "1 make "Ad :Co*:A make "Bd :Co*:B make "ALTURAd :Co*:ALTURA
    make "C1 220-:B/2-100 make "C2 220-:ALTURA/2-185
    pu setx :C1 sety :C2 pd
end

```

```

to dibujo
    rt 90-:ALFA fd :Ad rt 2*:ALFA+:BETA/2
    rayado :ALTURAd pu bk :ALTURAd pd lt :BETA/2
    fd :Ad pd rt 180-:ALFA fd :Ed rt 90
    ern [Ad Bd ALTURAd C1 C2 Co M]
end

```

II.2.1.5.- Triángulo equilátero

Al seleccionar el apartado (3) de la figura (II.4.), se ejecuta el procedimiento 'dcte' del módulo "triángulo".

```
Listado del procedimiento 'dcte':  to dcte
                                     erall fs
                                     load "dcte dcte
                                     end
```

Al cargarse el módulo "dcte" en el espacio de trabajo también se cargan los procedimientos 'recuadro', 'dcte', 'cte', 't', 'dte', 'ctl' y 'pit'.

La siguiente instrucción del procedimiento 'dcte', perteneciente al módulo "triangulo", es la de ejecutar el procedimiento 'dcte', esta vez el perteneciente al módulo "dcte".

DIBUJO DEL TRIANGULO EQUILATERO

PARA EL DIBUJO DE UN TRIANGULO EQUILATERO TIENE QUE ESCRIBIR dte
Y A CONTINUACION LA MEDIDA DE UNO DE SUS LABOS

EJEMPLO: ESCRIBA cs...dte 200 Y PULSE RETURN

EL PROGRAMA DIBUJARA UN TRIANGULO EQUILATERO DE LADO = 200

* PARA EL CALCULO ESCRIBA cte Y PULSE RETURN *

* PARA EL MENU ANTERIOR ESCRIBA t Y PULSE RETURN *

El procedimiento 'dcte' presenta en pantalla el menú de la fig. II.16.

(fig. II.16.)

Desde la figura (II.16.) se tiene acceso al dibujo de triángulos equiláteros 'dte', al menú anterior 't' y al módulo de cálculo de triángulos equiláteros 'cte'.

Si se elige la opción del módulo anterior, al escribir t y pulsar return, se ejecuta el procedimiento 't'. El procedimiento 't', borra el espacio de trabajo, carga el modulo "triángulo" y ejecuta el procedimiento 'triángulo' presentando en pantalla la figura (II.4).

Si elegimos la opción del dibujo se podrá dibujar cualquier triángulo equilátero o cualquier tipo de figura que contenga esa figura patrón.

Si elegimos la opción del cálculo, al escribir cte y pulsar return se ejecuta el procedimiento 'cte' del módulo "dcte", que es el que se encuentra en el espacio de trabajo en este momento.

Este procedimiento borra el espacio de trabajo y carga el módulo "cte". Al cargarse este módulo se cargan los trece procedimientos que lleva en su declaración, estos procedimientos son; 'rayado', 'marco', 'salida', 'texto', 'cte', 'dcte', 'te', 'ctl', 'variables', 'medidas', 'pit', 'tg' y 'ventana'.

Si el módulo "cte" ocupa el espacio de trabajo, al ejecutarse el procedimiento 'cte' sale en pantalla el menú de la figura (II.17.).

Desde este menú se tiene acceso al cálculo de un triángulo equilátero y al menú anterior, de dibujo de triángulos.

CALCULO DEL TRIANGULO EQUILATERO

AL SER LOS TRES ANGULOS Y LADOS IGUALES SOLO ES NECESARIO SABER LO QUE MIDE
UNO DE LOS LADOS PARA PODER HALLAR TODAS LAS VARIABLES DEL TRIANGULO
ESCRIBA te (lado) Y PULSE RETURN...A CONTINUACION SE BORRARA LA PANTALLA
Y EL PROGRAMA LE DIBUJARA EL TRIANGULO A LA VEZ QUE DA TODAS LAS MEDIDAS
EJEMPLO: te 450

* SI QUIERE VOLVER AL MENU ANTERIOR ESCRIBA DCTE Y PULSE RETURN *

El procedimiento 'cte' presenta en pantalla el menú de la fig. II.17.

(fig. II.17.)

Durante la ejecución de 'te' se ejecutan los procedimientos previamente definidos, 'ventana', 'variables', 'medidas', 'rayado' y 'pit'.

II.2.1.6.- Ejemplos de aplicación y listado de procedimientos

Ejemplos de aplicación -(Dibujo de triángulos)-

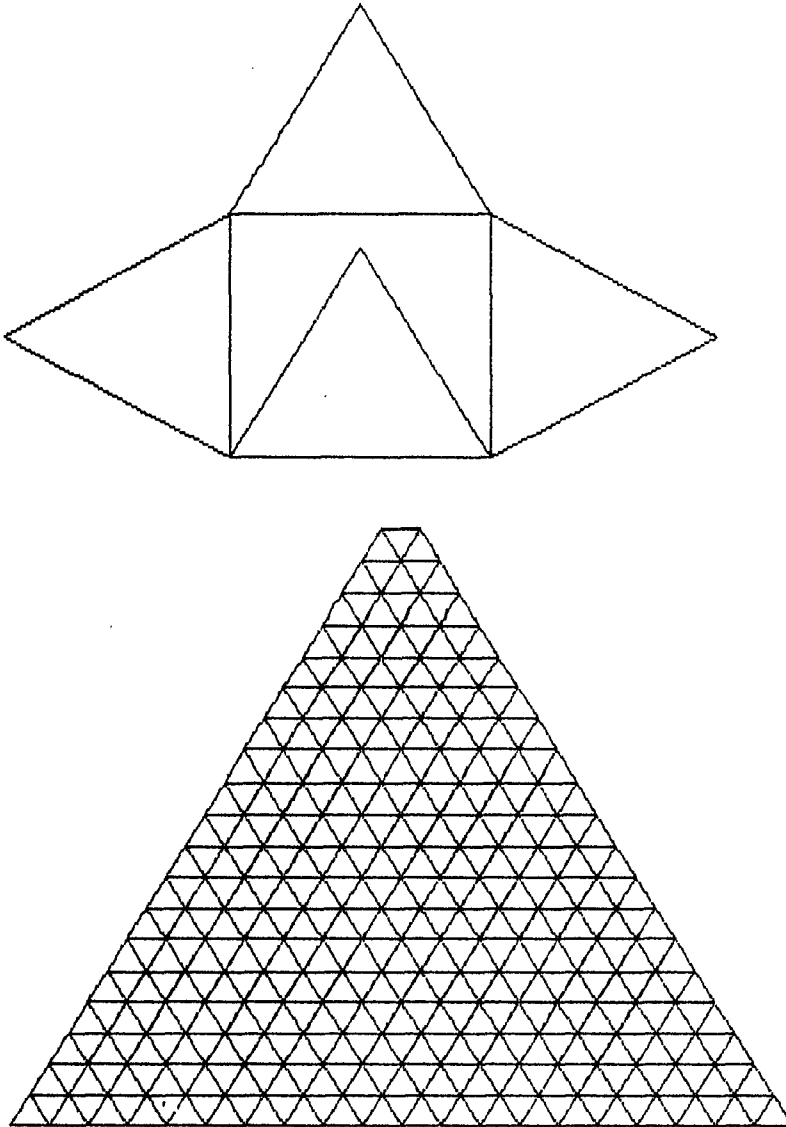
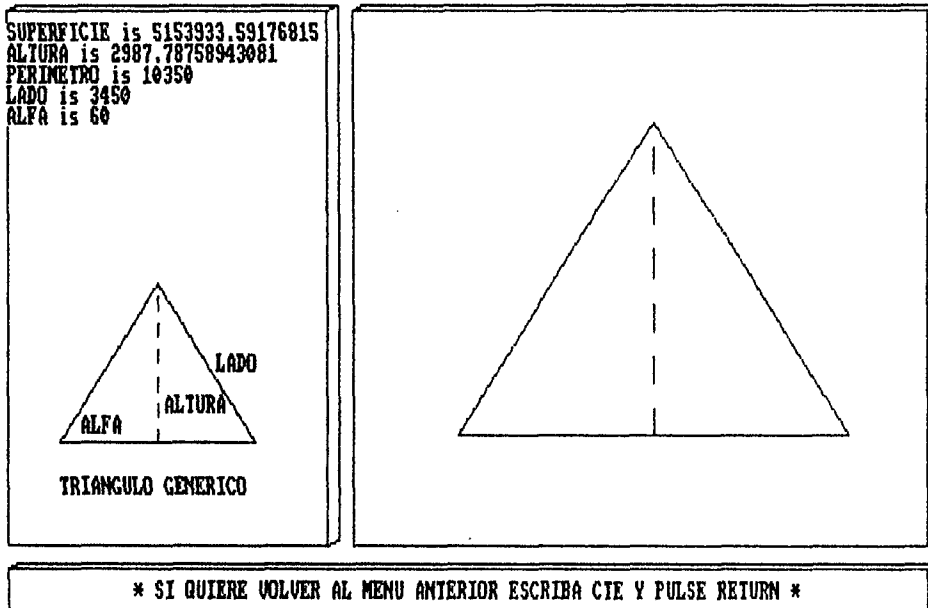


fig. (II.18.)

Ejemplos de aplicación - (Cálculo de triángulos) -



Ejemplo con
'te'.

fig. (II.19.)

Listado de procedimientos - Módulo "dcte" -

to dcte

```
recuadro setcursor [31 1]
pr[DIBUJO DEL TRIANGULO EQUILATERO] setcursor [14 6]
pr[PARA EL DIBUJO DE UN TRIANGULO EQUILATERO TIENE QUE ESCRIBIR dte]
setcursor [21 8]
pr[Y A CONTINUACION LA MEDIDA DE UNO DE SUS LADOS]
setcursor [22 12]
pr [EJEMPLO: ESCRIBA cs..dte 200 Y PULSE RETURN]
setcursor [16 15]
pr[EL PROGRAMA DIBUJARA UN TRIANGULO EQUILATERO DE LADO = 200]
setcursor [20 20]
pr[* PARA EL CALCULO ESCRIBA cte Y PULSE RETURN *]
setcursor [18 22]
pr[* PARA EL MENU ANTERIOR ESCRIBA t Y PULSE RETURN *]
setcursor [0 29] pit 2
```

end

```
to cte
  erall ct fs
  load "cte cte
end
```

```
to t
  erall ct fs
  load "triángulo triángulo
end
```

```
to recuadro
  ct clean ht setsplit 2 pit 1
  pu home setpos [-140 219] pd ctl 295 40 10
  pu home setpos [-290 -55] pd ctl 590 235 10
  pu home setpos [-290 -135] pd ctl 590 65 10
end
```

```
to ctl :x :y :z
  repeat 2[fd :y rt 90 fd :x rt 90] fd :y
  rt 65 fd :z rt 25 fd :x rt 90 fd :y
  rt 65 fd :z rt 115 fd :y rt 65 fd :z
end
```

```
to pit :n
  repeat :n[type char 7]
end
```

```
to dte :a
  rt 30 repeat 2[fd :a rt 120] fd :a rt 90
end
```

Listado de procedimientos - Módulo "cte" -

```
to ctl
  (Igual que 'ctl' del módulo "dcte")
end

to pit
  (Igual que 'pit' del módulo "dcte")
end

to ventana
  cs ht setsplit 1 pit 1
  pu home setpos [-360 -185] pd ctl 245 440 10
  pu home setpos [-95 -185] pd ctl 440 440 10
  pu home setpos [-360 -240] pd ctl 705 35 10
end

to variables
  make "LADO :a
  make "ALTURA :a*sin 60
  make "ALFA 60
  make "SUPERFICIE (:a*:ALTURA)/2
  make "PERIMETRO :LADO
end

to medidas
  make "A1 :LADO
  make "ALT. :ALTURA
  make "C 300/:A1
  make "A1 :A1*:C make "ALT. :ALT.*:C ern [C]
end

to cte
  ts marco texto salida
end
```

```

to rayado :a
    repeat 6[pu fd :a/12 pd fd :a/12]
end

```

```

to dcte
    erall ct fs
    load "dcte dcte
end

```

```

to marco
    pit 1 setsplit 1 cs ht pu home setpos [-130 205] pd
    ct1 270 33 10 pu home setpos [-335 -201 pd ct1685 210 10
    pu home setpos [-335 -75] pd ct1 685 40 10
    setcursor [30 2] pr[CALCULO DEL TRIANGULO EQUILATERO]
end

```

```

to salida
    setcursor [13 18]
    pr[SI QUIERE VOLVER AL MENU ANTERIOR ESCRIBA dcte Y PULSE RETURN]
    setcursor [0 28] pit 2
end

```

```

to texto
    setcursor [7 6]
    pr[AL SER LOS TRES ANGULOS Y LADOS IGUALES SOLO ES NECESARIO SABER LO QUE]
    setcursor [11 8]
    pr[UNO DE LOS LADOS PARA PODER HALLAR TODAS LAS VARIABLES DEL TRIANGULO]
    setcursor [9 10]
    pr[ESCRIBA te (lado) Y PULSE RETURN..A CONTINUACION SE BORRARA LA PANTALLA]
    setcursor [9 12]
    pr[Y EL PROGRAMA DIBUJARA EL TRIANGULO A LA VEZ QUE DA TODAS LAS MEDIDAS]
    setcursor [34 14]
    pr [EJEMPLO: te 450]
end

```

```
to tg
  ct cs ht pu home setpos [-320 -100] pd
  rt 30 fd 150 rt 150 rayado 150*sin 60 pu
  bk 150*sin 60 pd lt 30 fd 150 rt 120 fd 150
  rt 90 setcursor [7 20] pr [ALFA]
  setcursor [20 17] pr [LADO] setcursor [15 19]
  pr [ALTURA] setcursor [5 23] pr [TRIANGULO GENERICO]
  setcursor [0 30]
end
```


II.2.1.7.- Triángulo cualquiera

Al seleccionar el apartado (4) del menú de la figura (II.4.), se ejecuta el procedimiento 'dctc' del módulo "triángulo".

```
Listado del procedimiento 'dctc':  to dctc  
  
                                     erall fs load "dctc dctc  
  
                                     end
```

Al cargarse el módulo "dctc" en el espacio de trabajo también se cargan los procedimientos 'recuadro', 'dctc', 'dct1', 'dct2', 't', 'ctl' y 'pit'.

La siguiente instrucción del procedimiento 'dctc', perteneciente al módulo "triángulo", es la de ejecutar el procedimiento 'dctc' perteneciente al módulo "dctc".

DIBUJO Y CALCULO DE UN TRIANGULO CUALQUIERA

A TRAVES DE ESTE MENU PUEDE ACCEDER AL CALCULO Y DIBUJO DE CUALQUIER TRIANGULO.
PARA ELLO DEBE APORTAR LOS DATOS MINIMOS NECESARIOS PARA EL CALCULO DE LA FIGURA
Y POSTERIOR DIBUJO POR EL PROGRAMA.

- * SI CONOCE DOS LADOS Y UN ANGULO.....ESCRIBA dct1 Y PULSE RETURN
- * SI CONOCE UN LADO Y DOS ANGULOS.....ESCRIBA dct2 Y PULSE RETURN

ESTOS SON LOS DATOS MINIMOS NECESARIOS PARA EL CALCULO DE UN
TRIANGULO, SI CONOCE MAS DATOS NO SE LOS DE AL ORDENADOR, EL
PROGRAMA CALCULA LOS DATOS RESTANTES ANTES DE DIBUJAR.

* PARA VOLVER AL MENU ANTERIOR ESCRIBA t Y PULSE RETURN *

El procedimiento 'dctc' presenta en pantalla el menú de la fig. II.20.

(fig. II.20.)

Desde el menú de la figura (II.20.) se tiene acceso al dibujo y cálculo de un triángulo cualquiera en dos supuestos. El primer supuesto es que se conocen dos lados y el ángulo que forman. El segundo supuesto es que se conocen dos ángulos y uno de los lados. También se tiene acceso al módulo anterior.

Si se elige la opción del módulo anterior, al escribir t y pulsar return, se ejecuta el procedimiento 't'. El procedimiento 't', borra el espacio de trabajo, carga el módulo "triángulo" y ejecuta el procedimiento 'triángulo'. Se retorna al menú de la fig. (II.4).

Si se elige la opción del dibujo y cálculo de un triángulo cualquiera, en el primer supuesto, se escribe dct1 y se pulsa return. Al escribir dct1 y pulsar return, se ejecuta el procedimiento 'dct1' perteneciente al módulo "dcto". Este procedimiento borra el espacio de trabajo y carga el módulo "dct1". La siguiente instrucción del procedimiento 'dct1', perteneciente al módulo "dcto", es la de ejecutar el procedimiento 'dct1' del módulo "dct1".

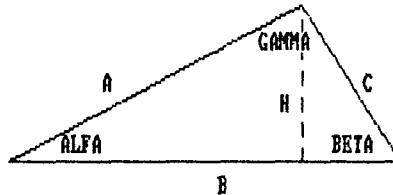
Al cargarse en el espacio de trabajo el módulo "dct1" también se cargan los trece procedimientos que lleva en su declaración. Estos procedimientos son: 'dct1', 'rayado', 'dcto', 'dt', 'd', 'ctcl', 'dts', 'dibujo', 'ctl', 'recuadro', 'variables', 'dib' y 'pit'.

Al ejecutarse el procedimiento 'dct1' del módulo "dct1" sale en pantalla el menú de la figura (II.21.). Desde este menú se tiene acceso al dibujo de un triángulo cualquiera en el supuesto (1), al módulo de cálculo de un triángulo cualquiera en el supuesto (1) y al módulo anterior.

*** D I B U J O ***

PARA DIBUJAR UN TRIANGULO DEL QUE SE CONOCEN DOS LADOS Y EL ANGULO QUE FORMAN, TIENE QUE ESCRIBIR , dt (lado1) (lado2) (angulo) Y PULSAR RTN. EJEMPLO: SI ESCRIBE dt 260 300 30 Y PULSA RTN EL PROGRAMA DIBUJARA UN

TRIANGULO COMO EL DE LA PANTALLA, CON A = 260 - B = 300 Y ALFA = 30 EL RESTO DE LOS DATOS LOS CALCULA EL PROGRAMA ANTES DE DIBUJAR.



PRUEBE EL EJEMPLO: es.... dt 240 310 35

*** PARA EL CALCULO ESCRIBA ctc1 Y PULSE RETURN ***

*** PARA VOLVER AL MENU ANTERIOR ESCRIBA dctc Y PULSE RETURN ***

El procedimiento 'dct1' presenta en pantalla el menú de la figura (II.21.).

fig. (II.21.)

Si se elige la opción de dibujo de un triángulo cualquiera podremos dibujar cualquier triángulo del que conozcamos dos lados y el ángulo que forman. El procedimiento encargado del dibujo es 'dt' (del módulo "dct1"). Este procedimiento tiene tres entradas que son los datos que conocemos.

Al escribir dt con las tres entradas y pulsar rtn. se ejecuta el procedimiento previamente definido 'dts', con esto concluye el dibujo del triángulo, el resto del procedimiento esta dedicado a escribir notas de aclaración al pie de la pantalla.

El procedimiento 'dts' tiene las mismas entradas que el procedimiento 'dt', con ellas ejecuta los procedimientos 'variables' y 'dib'.

El procedimiento 'variables' calcula todos los lados y ángulos del triángulo y los guarda en memoria. El procedimiento 'db' dibuja el triángulo con los datos calculados previamente por 'variables', aquí termina la ejecución del procedimiento 'dts'. Si se quiere dibujar sin notas al pie de la pantalla bastará con poner dts en vez de dt, de esta forma el dibujo concluye con la ejecución de 'dts'.

Si se elige la opción del módulo anterior, al escribir dctc y pulsar rtn. se borra el espacio de trabajo y se ejecuta el procedimiento 'dctc' del módulo "dct1". El procedimiento 'dctc' nos llevará de vuelta al menú de la figura (II.20.).

Si se elige la opción de cálculo de un triángulo cualquiera, se escribe ctcl y se pulsa return, esta operación ejecuta el procedimiento 'ctcl' del módulo "dct1". Este procedimiento borra el espacio de trabajo y carga el módulo "ctcl" .

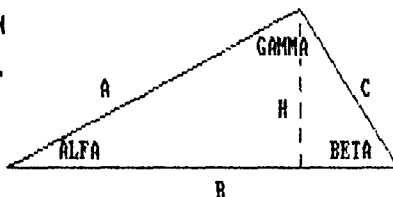
Con el módulo 'ctcl' se podrán hallar todas las variables de cualquier triángulo del que se conozcan dos lados y el ángulo que forman, al mismo tiempo el programa dibujará el triángulo objeto del cálculo.

Al cargarse el módulo "ctc1" se cargan los quince procedimientos que lleva en su declaración. Estos procedimientos son: 'rayado', 'dt', 'ct1', 'pit', 'ventana', 'vent', 'texto', 'variables', 'medida', 'ctc1', 'tg', 'dibujo', 'dib', 'var' y 'dct1'.

Al elegirse la opción del cálculo, se ejecuta el procedimiento 'ctc1' del módulo "dct1". Este procedimiento primero borra el espacio de trabajo después carga el módulo "ctc1" y por último ejecuta el procedimiento 'ctc1' del módulo "ctc1". Al ejecutarse el procedimiento 'ctc1' del módulo "ctc1" sale en pantalla el menú de la figura (II.22.).

*** C A L C U L O ***

LOS DATOS QUE CONOCE SON DOS LADOS Y EL ANGULO QUE FORMAN. CON ESTOS DATOS HALLARA DE FORMA IMMEDIATA LOS OTROS DOS ANGULOS, EL OTRO LADO LA SUPERFICIE, EL PERIMETRO Y LA ALTURA.
 EJEMPLO: SI LAS MEDIDAS DE LOS LADOS SON 260 , 300 Y EL ANGULO 30 º, ESCRIBA.....
 ..dt 260 300 30 Y PULSE RETURN
 EL PROGRAMA DIBUJARA EL TRIANGULO DE LA PANTALLA. LOS DATOS QUE APORTAMOS CORRESPONDEN EN EL DIBUJO CON (A , B y ALFA) .
 EL RESTO DE LOS DATOS LOS CALCULA EL PROGRAMA Y LOS MUESTRA EN PANTALLA.



El procedimiento 'ctc1' presenta el menú de la figura (II.22.).

*** PARA VOLVER AL MENU ANTERIOR ESCRIBIR dct1 Y PULSAR RETURN ***

fig. (II.22.)

Desde el menú de la figura (II.22.) se tiene acceso al menú anterior y al cálculo de cualquier triángulo en el supuesto (1).

Si se elige la opción del menú anterior, al escribir `dct1` y pulsar `return`, se ejecuta el procedimiento '`dct1`' del módulo "`ctc1`". Este procedimiento borra el espacio de trabajo y nos lleva al menú de la figura (II.21.).

Si se elige la opción del cálculo, bastará con escribir `dt` con las tres entradas (los dos lados y el ángulo) y al pulsar `return` se ejecutará el procedimiento que hace el cálculo de todas las variables de ese triángulo. Al mismo tiempo dibujará en la pantalla el triángulo objeto del cálculo. En la ejecución del procedimiento '`dt`' están incluidos seis procedimientos previamente definidos. Estos procedimientos se ejecutan en el siguiente orden: '`ventana`' - '`variables`' - '`medida`' - '`dibujo`' - '`var`' y '`pit`'.

El procedimiento '`ventana`' divide la pantalla en tres apartados mediante delimitadores gráficos. Una vez ejecutado el procedimiento '`ventana`' se ejecuta el procedimiento '`variables`', este procedimiento toma las entradas del procedimiento '`dt`' y con ellas calcula todos los datos restantes del triángulo objeto del cálculo y los guarda en memoria.

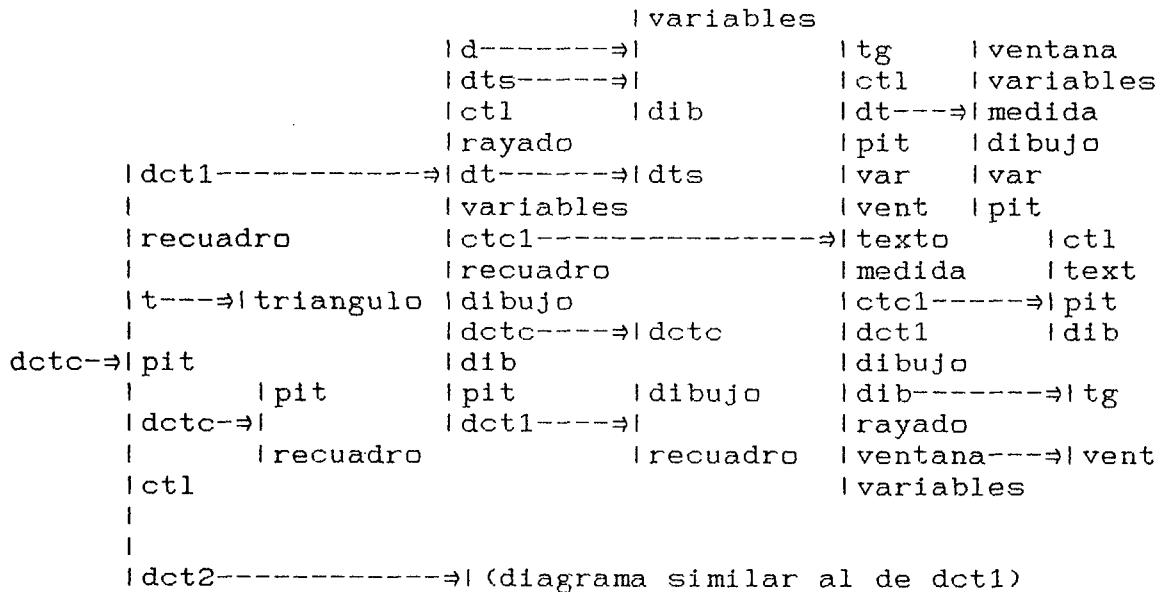
El procedimiento '`medida`' se encarga de reducir o aumentar el tamaño del triángulo para que quepa y quede exactamente en el centro del apartado que se ha reservado para su representación.

Las nuevas medidas del triángulo se guardan en memoria y el procedimiento 'dibujo' se encarga del dibujo del triángulo.

El procedimiento 'var' se ejecutará si el ángulo (α) es menor de 90°, en cuyo caso hará cambios en los valores de los ángulos (β) y Gamma. Estos cambios vienen originados por la forma en que están calculados: Dependiendo del signo que tenga la expresión [$A-(B/\cos \alpha)$] hay que hacer los cambios.

El procedimiento 'dt' concluye haciendo una reorganización de la memoria del espacio de trabajo y borrando de memoria todos los valores de todas las variables que se han utilizado durante la ejecución del procedimiento.

Diagrama modular del dibujo y cálculo de un triángulo cualquiera en el supuesto (1):



Si se elige la opción (2) del menú de la figura (II.20.), (conocemos dos ángulos y uno de sus lados), al escribir `dct2` y pulsar `return`, se ejecuta el procedimiento '`dct2`' del módulo "`dctc`".

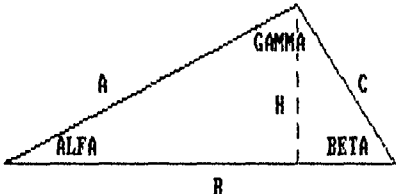
Este procedimiento borra el espacio de trabajo, carga el módulo "`dct2`" y ejecuta el procedimiento '`dct2`' del módulo "`dct2`".

El procedimiento '`dct2`' del módulo "`dct2`" presenta en pantalla el menú de la figura (II.23.).

Desde este momento el proceso que se sigue para el dibujo y cálculo del triángulo es similar al caso (1), la única diferencia es que al calcular las variables se parte de unos datos diferentes.

*** D I B U J O ***

PARA DIBUJAR UN TRIANGULO DEL QUE CONOCE DOS ANGULOS Y UN LADO ESCRIBA..
`..dt (angulo1) (angulo2) (lado) ...Y PULSE RETURN.`
 EJEMPLO:SI ESCRIBE `dt 30 60 260` Y PULSA RETURN EL PROGRAMA DIBUJARA
 UN TRIANGULO COMO EL DE LA PANTALLA
 CON ALFA = 30 , BETA = 60 y A = 260
 EL RESTO DE LOS DATOS LOS CALCULA EL
 PROGRAMA ANTES DE DIBUJAR.



PRUEBE EL EJEMPLO: es.... `dt 40 50 250 ...`

*** PARA EL CALCULO ESCRIBA `ctc2` Y PULSE RETURN ***
*** PARA VOLVER AL MENU ANTERIOR ESCRIBA `dctc` Y PULSE RETURN ***

El procedimiento '`dct2`' presenta el menú de la figura (II.23)

fig. (II.23)

II.2.1.8.- Ejemplos de aplicación y listado de procedimientos

Ejemplos de aplicación -(Dibujo de triángulos)-

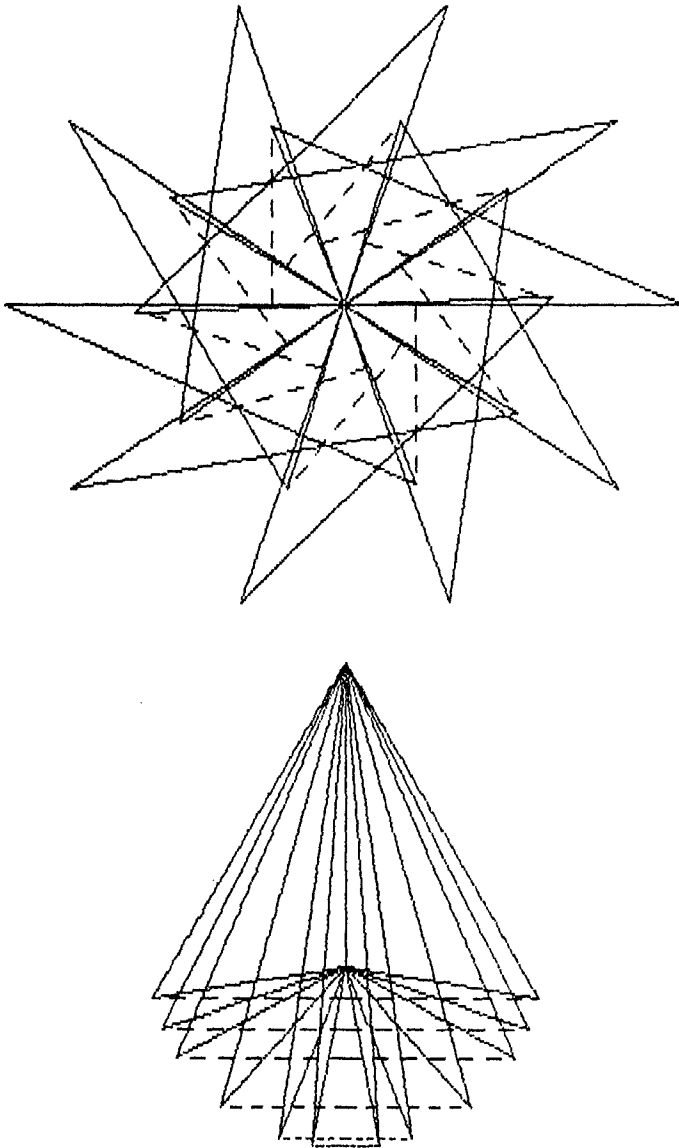
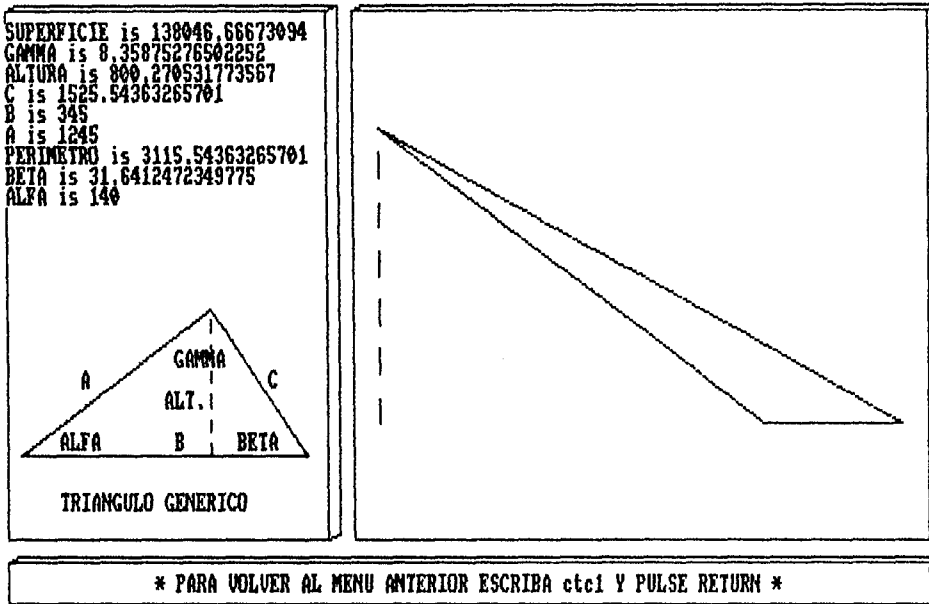


fig. (II.24.)

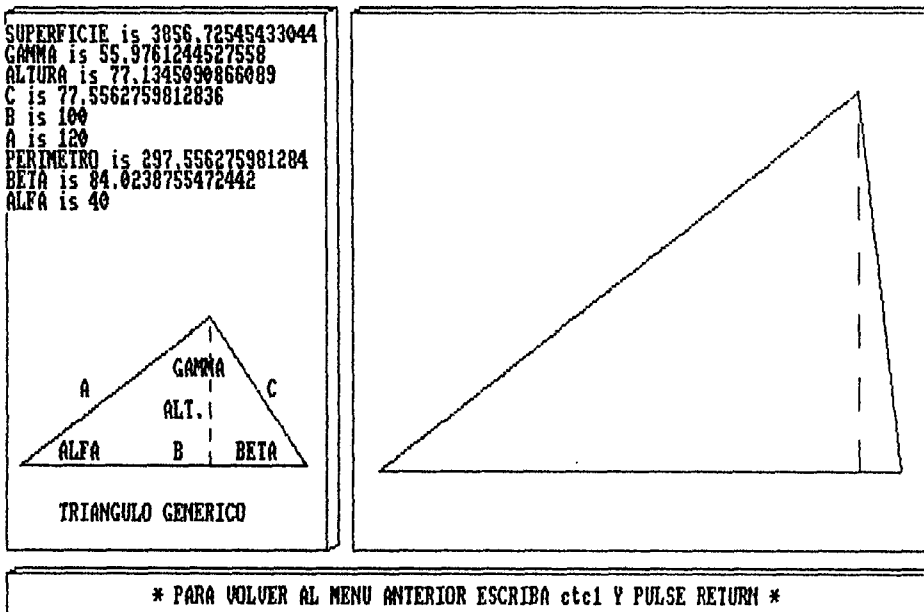
Ejemplos de aplicación - (Cálculo de triángulos) -



Ejemplo con

dt : A : B : α

fig. (II.25.)



Ejemplo con

dt : A : B : α

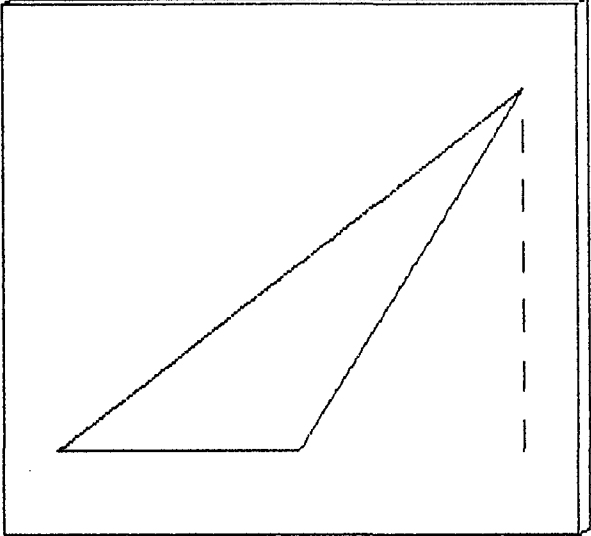
fig. (II.26.)

```

SUPERFICIE is 671450.824747026
GAMMA is 20
ALTURA is 1478.41142416
C is 1707.12249006342
B is 908.340958104442
A is 2300
PERIMETRO is 4915.46345616786
BETA is 120
ALFA is 40

```

(TRIANGULO GENERICO)



Ejemplo con

$$dt : \alpha : B : A$$

* PARA VOLVER AL MENU ANTERIOR ESCRIBA ctc2 Y PULSE RETURN *

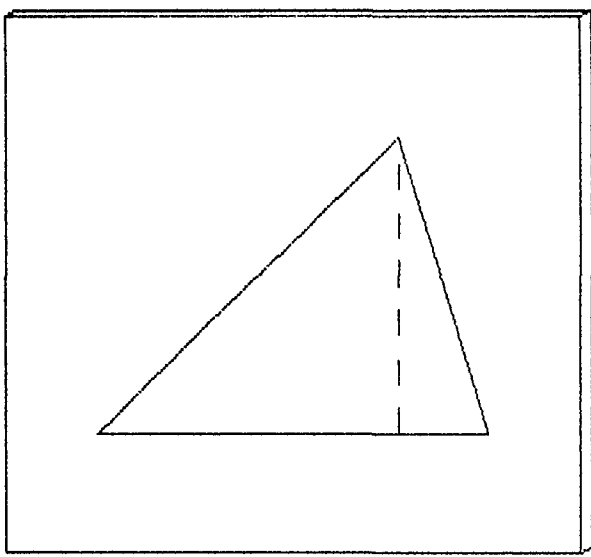
fig. (II.27.)

```

SUPERFICIE is 4982.37516213054
GAMMA is 60
ALTURA is 89.1981337070465
C is 92.7927687811254
B is 111.714784941446
A is 124
PERIMETRO is 328.507553722572
BETA is 74
ALFA is 46

```

(TRIANGULO GENERICO)



Ejemplo con

$$dt : \alpha : \beta : A$$

* PARA VOLVER AL MENU ANTERIOR ESCRIBA ctc2 Y PULSE RETURN *

fig. (II.28.)

Listado de procedimientos - Módulo "dctc"-
.....

```
to dctc
recuadro setcursor [25 2]
pr[DIBUJO Y CALCULO DE UN TRIANGULO CUALQUIERA] setcursor [6 6]
pr[A TRAVES DE ESTE MENU PUEDE ACCEDER AL CALCULO Y DIBUJO DE CUALQUIER
TRIANGULO.] setcursor [5 8]
pr[TIENE QUE APORTAR LOS DATOS MINIMOS NECESARIOS PARA EL CALCULO DE LA
FIGURA] setcursor [25 10]
pr [Y POSTERIOR DIBUJO POR EL PROGRAMA.] setcursor [6 13]
pr[SI CONOCE DOS LADOS Y UN ANGULO.....]
setcursor [57 13]
pr[ESCRIBA dct1 Y ESCRIBA RETURN] setcursor [6 15]
pr[* SI CONOCE UN LADO Y DOS ANGULOS.....]
setcursor [57 15]
pr[ESCRIBA dct2 Y PULSE RETURN] setcursor [14 18]
pr[ESTOS SON LOS DATOS MINIMOS NECESARIOS PARA EL CALCULO DE UN]
setcursor [14 20]
pr[TRIANGULO, SI CONOCE MAS DATOS NO SE LOS DE AL ORDENADOR, EL]
setcursor [14 22]
pr[PROGRAMA LOS CALCULA ANTES DE DIBUJAR.] setcursor [15 27]
pr[* PARA VOLVER AL MENU ANTERIOR ESCRIBA t Y PULSE RETURN*]
setcursor [0 30] pit 2
end
```

```
to dct1
    erall ct fs load "dct1 dct1"
end
```

```
to dct2
    erall ct fs load "dct2 dct2"
end
```

```
to recuadro
    pit 1 ct clean ht setsplit 2 pu home setpos [-175 200]
    ctl 370 40 0 pu home setpos [-180 195] pd ctl 380 50 10
    pu home setpos [-350 -175] pd ctl 700 350 10
    pu home setpos [-350 -225] pd ctl 700 40 10
end
```

```
to t
  erall fs load "triángulo triángulo"
end
```

```
to ctl :x :y :z
  repeat 2[fd :y rt 90 fd :x rt 90] fd :y rt 65 fd :z rt 25
  fd :x rt 90 fd :y rt 65 fd :z rt 115 fd :y rt 65 fd :z
end
```

```
to pit :n
  repeat :n[type chat 7]
end
```

.....
 Listado de procedimientos - Módulo "dct1" -


```
to dct1
  pit 1 ht recuadro dibujo setcursor [37 1] pr[*D I B U J O *]
  setcursor [9 5]
  pr[PARA DIBUJAR UN TRIANGULO DEL QUE SE CONOCEN DOS LADOS Y EL ANGULO QUE]
  setcursor [9 7]
  pr[FORMAN, TIENE QUE ESCRIBIR, dt (lado1) (lado2) (angulo) Y PULSAR RTN.]
  setcursor [9 9]
  pr[EJEMPLO: SI ESCRIBE dt 260 300 30 Y PULSA RTN. EL PROGRAMA DIBUJARA UN]
  setcursor [5 12] pr[TRIANGULO COMO EL DE LA PANTALLA,]
  setcursor [5 14] pr[CON A = 260 B = 300 y ALFA = 30]
  setcursor [5 16] pr[EL RESTO DE LOS DATOS LOS CALCULA]
  setcursor [5 18] pr[EL PROGRAMA ANTES DE DIBUJAR.]
  setcursor [9 21] pr[PRUEBE EL EJEMPLO: es... dt 240 310 35]
  setcursor [19 25] pr[*PARA EL CALCULO ESCRIBA ctc1 Y PULSE RETURN*]
  setcursor [13 27]
  pr[*PARA VOLVER AL MENU ANTERIOR ESCRIBA dcte Y PULSE RETURN*]
  setcursor [0 30] pit 2
end
```

```
to rayado :a
  repeat 6[pu fd :a/12 pd fd :a/12]
end
```

```
to dctl
    erall ct fs load "dctl dctl"
end
```

```
to dt :a :b :alfa
    dts :a :b :alfa
    setsplit 2 setcursor [13 27]
    pr[*PARA VOLVER AL MENU ANTERIOR ESCRIBA dctl Y PULSE RETURN*]
    setcursor [10 29]
    pr[*SI QUIERE EL DIBUJO SIN NOTAS AL PIE ESCRIBA dts EN VEZ DE dt*]
end
```

```
to d :a :b :alfa
    variables dib
end
```

```
to ctcl
    erall ct fs load "ctcl ctcl"
end
```

```
to dts :a :b :alfa
    variables dib setsplit 1
end
```

```
to dibujo
    ct pu home ht sety -50 pd d 260 300 30
    setcursor [54 14] pr[A] setcursor [65 19] pr[B]
    setcursor [79 14] pr[C] setcursor [50 17] pr[ALFA]
    setcursor [76 17] pr[BETA] setcursor [62 12] pr[GAMMA]
    setcursor [71 15] pr[H]
end
```

```
to pit :n
    repeat :n[type char 7]
end
```

```

to ctl :x :y :z
    repeat 2[fd :y rt 90 fd :x rt 90] fd :y rt 70 fd :z
    rt 20 fd :x rt 90 fd :y rt 70 fd :z rt 110 fd :y rt 70 fd :z
end

```

```

to recuadro
    ct cs pu ho me setpos [-85 217] pd ctl 160 38 15
    pu home setpos [-340 -130] pd ctl 670 325 15
    pu home setpos [-340 -220] pd ctl 670 70 15
end

```

```

to variables
    ht make "x :a*cos :alfa
    make "ALTURA :a*sin :alfa
    make "y :b-:x
    make "BETA arctan (:ALTURA/:y)
    make "GAMMA 180-:alfa-:BETA
    make "C :ALTURA/sin :BETA
end

```

```

to dib
    rt 90 - :alfa fd :a rt :alfa + 90 rayado :ALTURA pu bk :ALTURA
    pd lt 90 - :BETA fd :C rt 180 - :BETA fd :x + :y rt 90
end

```

.....
Listado de procedimientos - Módulo "ctl1" -
.....

```

to rayado :a
(Igual que en el módulo "dct1")
end

```

```

to ctl :x :y :z
(Igual que en el módulo "dct1")
end

```

```

to pit :n
(Igual que en el módulo "dct1")
end

```

```

to dt :a :b :alfa
  ventana variables medida dibujo if :C < 0 [make "C :C*-1]
  make "PERIMETRO :A + :B + :C if :ALFA < 90 [var]
  ern [x M Co d1 y a1 b1 c1] setcursor [14 28]
  pr[*PARA VOLVER AL MENU ANTERIOR ESCRIBA ctcl Y PULSE RTN.*]
  setcursor [0 1] pons recycle setcursor [0 30]
  ern [SUPERFICIE PERIMETRO ALTURA A B C ALFA BETA GAMMA: pit 2
end

```

```

to ventana
  vent pu home setpos [-350 -115] pd tg 190 220 40
  setcursor [5 21] pr[ALFA] setcursor [22 21] pr [BETA]
  setcursor [16 17] pr [GAMMA] setcursor [7 18]
  pr[TRIANGULO GENERICO] setcursor [25 18] pr [C]
  setcursor [15 19] pr [ALT] setcursor [0 30]
end

```

```

to vent
  cs ht setsplit 1 pit 1 pu home setpos [-360 -185]
  pd ctl 245 440 10 pu home setpos [-95 -185] pd
  ctl 440 440 10 pu home setpos [-360 -240] pd ctl 705 35 10
end

```

```

to texto
  setcursor [5 5]
  pr[LOS DATOS QUE CONOCE SON DOS LADOS Y EL ANGULO QUE FORMAN, CON ESTOS]
  setcursor [5 7]
  pr[DATOS HALLARA DE FORMA INMEDIATA LOS OTROS DOS ANGULOS, EL OTRO LADO]
  setcursor [5 9]
  pr[LA SUPERFICIE, EL PERIMETRO Y LA ALTURA.] setcursor [5 11]
  pr[EJEMPLO: SI LAS MEDIDAS DE LOS LADOS SON:] setcursor [5 13]
  pr[260, 300 Y EL ANGULO 30º, ESCRIBA.....] setcursor [5 15]
  pr[.dt 260 300 30... Y PULSE RETURN] setcursor [5 17]
  pr[EL PROGRAMA DIBUJARA EL TRIANGULO DE] setcursor [5 19]
  pr[LA PANTALLA, LOS DATOS QUE APORTAMOS] setcursor [5 21]
  pr[CORRESPONDEN EN EL DIBUJO CON (A, B Y ALFA).] setcursor [5 23]
  pr[EL RESTO DE LOS DATOS LOS CALCULA EL PROGRAMA Y LOS MUESTRA EN PANTALLA]
  setcursor [13 27]
  pr[*PARA VOLVER AL MENU ANTERIOR ESCRIBA dct1 Y PULSE RETURN*]
  setcursor [0 29]
end

```



```

to variables
    make "x :a*cos :alfa make "ALTURA :a*sin :alfa
    make "y :b-x make "ALFA :alfa make "BETA arctan (:ALTURA/:y)
    make "GAMMA 180-:alfa-:BETA make "SUPERFICIE :b*:ALTURA/2
    make "A :a make "B :b make "C :ALTURA/sin :BETA
end

```

```

to medida
    make "D -:A * cos :alfa if :alfa < 90 [make "D 0]
    make "D1 -:B + :A * cos :alfa if :alfa > 90 [make "D1 0]
    if and (:alfa<90) (:D1 < 0) [make "D 0 make "D1 0]
    make "D2 :D+:D1+:B
    make "M 400 make "a1 :a make "b1 :B make "c1 :C make "d1 :ALTURA
    if :D2>d1 [make "Co :M/:D2 go "1] make "Co :M/:d1
label "1 make "a1 :a1*:Co make "b1 :b1*:Co make "c1 :c1*:Co
    make "d1 :d1*:Co make "D2 :D2*:Co make "D1 :D1*:Co make "D :D*:Co
end

```

```

to ctcl
    pit 1 cs setsplit 2 pu home setpos [-100 217] pd ctl 170 40 10
    pu home setpos [-340 -170] pd ctl 660 370 10 pu home
    setpos [-340 -225] pd ctl 660 40 10 setcursor [35 1]
    pr[* C A L C U L O *] dib texto pit 2
end

```

```

to tg :uno :dos :tres
    make "x :uno*cos :tres make "ALTURA :uno*sin :tres
    make "y :dos-x make "BETA arctan (:ALTURA/:y)
    make "GAMMA 180-:tres-:BETA make "C :ALTURA/sin :BETA
    rt 90-:tres fd :uno rt :tres+90 rayado :ALTURA pu
    bk :altura PD lt 90-:BETA fd :C rt 180-:BETA fd :x+:y
    rt 90 ern [x ALTURA y BETA GAMMA C]
end

```

```

to dibujo
    make "C1 125-:D2/2+:D make "C2 25-:d1/2
    pu setx :C1 sety :C2 pd ern [D D1 D2 C1 C2]
    rt 90-:alfa fd :a1 rt :alfa+90 rayado :d1 pd bk :d1
    pd lt 90 - :BETA fd :c1 rt 180-:BETA fd :b1 rt 90
end

```

```

to dib
  ct pu home ht sety -50 pd tg 260 300 30 setcursor [54 14]
  pr[A] setcursor [65 19] pr[B] setcursor [79 14] pr[C]
  setcursor [50 17] pr[ALFA] setcursor [76 17] pr[BETA]
  setcursor [62 12] pr[GAMMA] setcursor [71 15] pr[H]
end

to var
  if :b/cos :alfa < :a [make "BETA 180+:BETA make "GAMMA :GAMMA-180]
end

to dct1
  erall ct fs load "dct1 dct1
end

```

Listado de procedimientos - Módulo "dct2" -


```

to rayado :a
  (Igual que el del módulo "ctc1")
end

to dctc
  erall ct fs load "dctc dctc
end

to dt :alfa :beta :a
  dts :alfa :beta :a setcursor [11 28]
  pr[*PARA VOLVER AL MENU ANTERIOR ESCRIBA dct2 Y PULSE RETURN*]
  setcursor [10 29]
  pr[*SI QUIERE EL DIBUJO SIN NOTAS AL PIE ESCRIBA dts EN VEZ DE dt*]
end

```

```

to ctc2
    erall ct fs load "ctc2 ctc2
end

```

```

to dibujo
    ct pu home sety -50 pd dts 30 60 260 setcursor [54 14]
    pr[A] setcursor [65 19] pr [B] setcursor [79 14] pr [C]
    setcursor [50 17] pr [ALFA] setcursor [76 17] pr [BETA]
    setcursor [69 12] pr [GAMMA] setcursor [71 15] pr [H]
end

```

```

to dct2
    pit 1 ct cs ht setsplit 1 pu home setpos [-75 220] pd ctl 140 35 15
    pu home setpos [-340 -130] pd ctl 670 325 15
    pu home setpos [-340 -220] pd ctl 670 70 15
    dibujo setcursor [37 1] pr[* D I B U J O *] setcursor [5 5]
    pr[PARA DIBUJAR UN TRIANGULO DEL QUE CONOCE DOS ANGULOS Y UN LADO ESCRIBA.]
    setcursor [5 7]
    pr[..dt (angulo1) (angulo2) (lado)... Y PULSE RETURN.]
    setcursor [5 9]
    pr[EJEMPLO:....SI ESCRIBE dt 30 60 260 Y PULSA RETURN EL PROGRAMA DIBUJARA]
    setcursor [5 11]
    pr[UN TRIANGULO COMO EL DE LA PANTALLA] setcursor [5 13]
    pr[CON ALFA = 30, BETA = 60 Y A = 260] setcursor [5 15]
    pr[EL RESTO DE LOS DATOS LOS CALCULA EL] setcursor [5 17]
    pr[PROGRAMA ANTES DE DIBUJAR.] setcursor [5 21]
    pr[PRUEBE EL EJEMPLO: cs...dt 40 50 250...] setcursor [19 25]
    pr[*PARA EL CALCULO ESCRIBA ctc2 Y PULSE RETURN*] setcursor [13 27]
    pr[*PARA VOLVER AL MENU ANTERIOR ESCRIBA dctc Y PILSE RETURN*]
    setcursor [0 30] pit 2
end

```

```

to pit :n
    repeat :n[type char 7]
end

```

```

to ctl :x :y :z
    (Igual que en el módulo "dct2")
end

```

```

to dts :alfa :beta :a
  if :alfa+:beta>180 [cs setcursor [10 15] pr[LA SUMA DE LOS TRES
  ANGULOS DE UN TRIANGULO ES 180º.....:alfa+:beta<180] stop]
  ht make "x :a*cos :alfa if :x<0 [make "x 0]
  make "ALTURA :a*sin :alfa make "GAMMA 180-:alfa-:beta
  make "C :ALTURA/sin :beta make "y :C*cos :beta
  if :x=0 [make "y :y-:a*cos 180-:alfa]
  rt 90-:alfa fd :a rt :alfa+90 rayado :ALTURA pu bk :ALTURA
  pd lt 90-:beta fd :C rt 180-:beta fd :x+:y rt 90
  ern [x ALTURA GAMMA C y] pit1
end

```

.....
 Listado de procedimientos - Módulo "ctc2" -


```

to medida
  make "M 400 make "a1 :a make "b0 :b make "c1 :c make "d1 :ALTURA
  make "b2 :A*cos 180-:alfa if :b2<0 [make "b2 :b2*-1]
  if :alfa>90 [make "b0 :c1*cos :beta]
  make "b3 :C*cos 180-:beta
  if :b0>:d1 [make "C0 :M/:b0 go "1] make "Co :M/:d1
label "1 make "a1 :a1*:Co make "b1 :b1*:Co make "c1 :c1*:Co
  make "d1 :d1*:Co make "b2 :b2*:Co make "b3 :b3*:Co
end

```

```

to rayado :a
  (Igual que en el módulo anterior)
end

```

```

to d :uno :dos :tres
  (Igual que 'tg' en el módulo "ctc1")
end

```

```

to tri
  ct pu home ht sety -50 pd d 260 300 30 setcursor [54 14] pr[A]
  setcursor [65 19] pr[B] setcursor [79 14] pr[C]
  setcursor [50 17] pr[ALFA] setcursor [76 17] pr[BETA]
  setcursor [69 12] pr[GAMMA] setcursor [71 15] pr[H]
end

```

```

to ctc2
pit 1 cs ht setsplit 1 pu home setpos [-95 215> pd ctl 170 40 10
pu home setpos [-340 -170] pd ctl 680 370 10 pu home
setpos [-340 -225] pd ctl 680 40 10 tri setcursor [35 1]
pr[*C A L C U L O *] setcursor [5 5]
pr[LOS DATOS QUE CONOCE SON DOS ANGULOS Y UN LADO. CON ESTOS DATOS PODRA]
setcursor [5 7]
pr[HALLAR DE FORMA INMEDIATA LOS OTROS DOS LADOS EL OTRO ANGULO LA ALTURA]
setcursor [5 9]
pr[LA SUPERFICIE Y EL PERIMETRO. EJEMPLO:] setcursor [5 11]
pr[SI LOS VALORES DE LOS ANGULOS SON 30º] setcursor [5 13]
pr[Y 60º Y EL LADO 260, ESCRIBA.....] setcursor [5 15]
pr[dt 30 60 260..... Y PULSE RETURN.] setcursor [5 17]
pr[EL PROGRAMA DIBUJARA EL TRIANGULO DE] setcursor [5 19]
pr[LA PANTALLA. LOS DATOS QUE CONOCEMOS] setcursor [5 21]
pr[CORRESPONDEN EN EL DIBUJO CON (ALFA,BETA Y A.)] setcursor [5 23]
pr[EL RESTO DE LOS DATOS LOS CALCULA EL PROGRAMA Y SE LOS MUESTRA EN
PANTALLA.] setcursor [13 27]
pr[*PARA VOLVER AL MENU ANTERIOR ESCRIBA dct2 Y PULSE RETURN*]
setcursor [0 30] pit 2
end

```

```

to dt :alfa :beta :a
ventana variables medida if :alfa<90 [make "b2 0]
if :beta<90 [make "b3 0]
make "C1 125-:b1/2+:b2/2-:b3/2 make "C2 35-:d1/2 pu home
setx :C1 sety :C2 pd
rt 90-:alfa fd :a1 rt :alfa+90 rayado :d1 pu bk :d1
pd lt 90-:beta fd :c1 rt 180-:beta fd :b1 rt 90
if :C<0 [make "C :C*-1]
make "SUPERFICIE :B*:ALTURA/2
make "PERIMETRO :a+:B+:C
make "ALFA :alfa make "BETA :beta make "A :a
ern [b0 b2 x y C1 C2 Co M a1 b1 c1 d1] setcursor [14 28]
pr[PARA VOLVER AL MENU ANTERIOR ESCRIBA ctc2 Y PULSE RTN.]
setcursor [0 1] pons ern [SUPERFICIE GAMMA ALFA BETA A B C PERIMETRO
ALTURA] setcursor [0 30 ] recycle pit 2
end

```

```

to variables
make "x :a*cos :alfa make "ALTURA :a*sin :alfa
make "GAMMA 180-:alfa-:beta make "C :ALTURA/sin :beta
make "y :C*cos :beta make "B :x+:y make "A :a
end

```

```
to dct2
    erall ct fs load "dct2 dct2
end
```

```
to pit :n
(Igual que en el módulo anterior)
end
```

```
to ctl :x :y :z
(Igual que en el módulo anterior)
end
```

```
to gen
    pu home setpos [-350 -185] pd d 150 220 40 setcursor [5 20] pr[ALFA]
    setcursor [21 20] pr[BETA] setcursor [13 17] pr[GAMMA]
    setcursor [5 18] pr[A] setcursor [15 22] pr[B] setcursor [4 23]
    pr[TRIANGULO GENERICO] setcursor [24 18] pr[C]
    setcursor [12 19] pr [ALT.] setcursor [0 30]
end
```

```
to ventana
    ct ht setsplit 1 pit 1 pu home setpos [-360 -185]
    pd ctl 245 440 10 pu home setpos [--95 -185] pd
    ctl 440 440 10 pu home setpos [-360 -240] pd
    ctl 705 35 10 gen setcursor [0 30]
end
```

II.2.2.- Dibujo de polígonos regulares

Al seleccionar el apartado (2) de la figura (II.3.), se ejecuta el procedimiento 'b' del módulo "plana".

```
Listado del procedimiento 'b':      to b
                                     erall fs
                                     load "poligono poligono
                                     end
```

Al cargarse el módulo "polígono" en el espacio de trabajo también se cargan los procedimientos 'pol', 'recuadro', 'p', 'polígono', 'pl', 'pols', 'ctl' y 'pit'.

La siguiente instrucción del procedimiento 'b' es la de ejecutar el procedimiento 'polígono'.

*** POLIGONOS ***

PARA DIBUJAR CUALQUIER POLIGONO BASTA CON APORTAR COMO
DATOS EL NUMERO DE LADOS Y LA MEDIDA DE UNO DE LOS LADOS
EJEMPLO: ESCRIBA ...pol 125 6 .. Y PULSE RETURN

EL PROGRAMA DIBUJARA UN HEXAGONO DE LADO = 125

*** SI QUIERE VOLVER AL MENU ANTERIOR ESCRIBA p Y PULSE RETURN ***

El procedimiento 'polígono' presenta en pantalla el menú de la fig. II.29.

(fig. II.29.)

Desde la figura (II.29.) tenemos acceso al dibujo de cualquier polígono 'pol' y al menú anterior 'p' .

Si elegimos la opción del módulo anterior, al escribir p y pulsar return, se ejecuta el procedimiento 'p'. El procedimiento 'p', borra el espacio de trabajo, carga el módulo "plana" y ejecuta el procedimiento 'plana' presentando en pantalla la figura (II.3).

Si elegimos la opción del dibujo de polígonos tendremos la posibilidad de hacer dibujos con cualquier polígono regular.

Antes de terminar el procedimiento de dibujo de polígonos ('pol') escribe dos notas al pie del dibujo. En estas notas se informa al usuario como trasladarse a otros módulos del programa. Si el usuario prefiere hacer el dibujo sin estas notas, tendrá que escribir -pols- en lugar de pol.

II.2.2.2.- Ejemplos de aplicación y listado de procedimientos

Ejemplos de aplicación - (Dibujo de polígonos)-

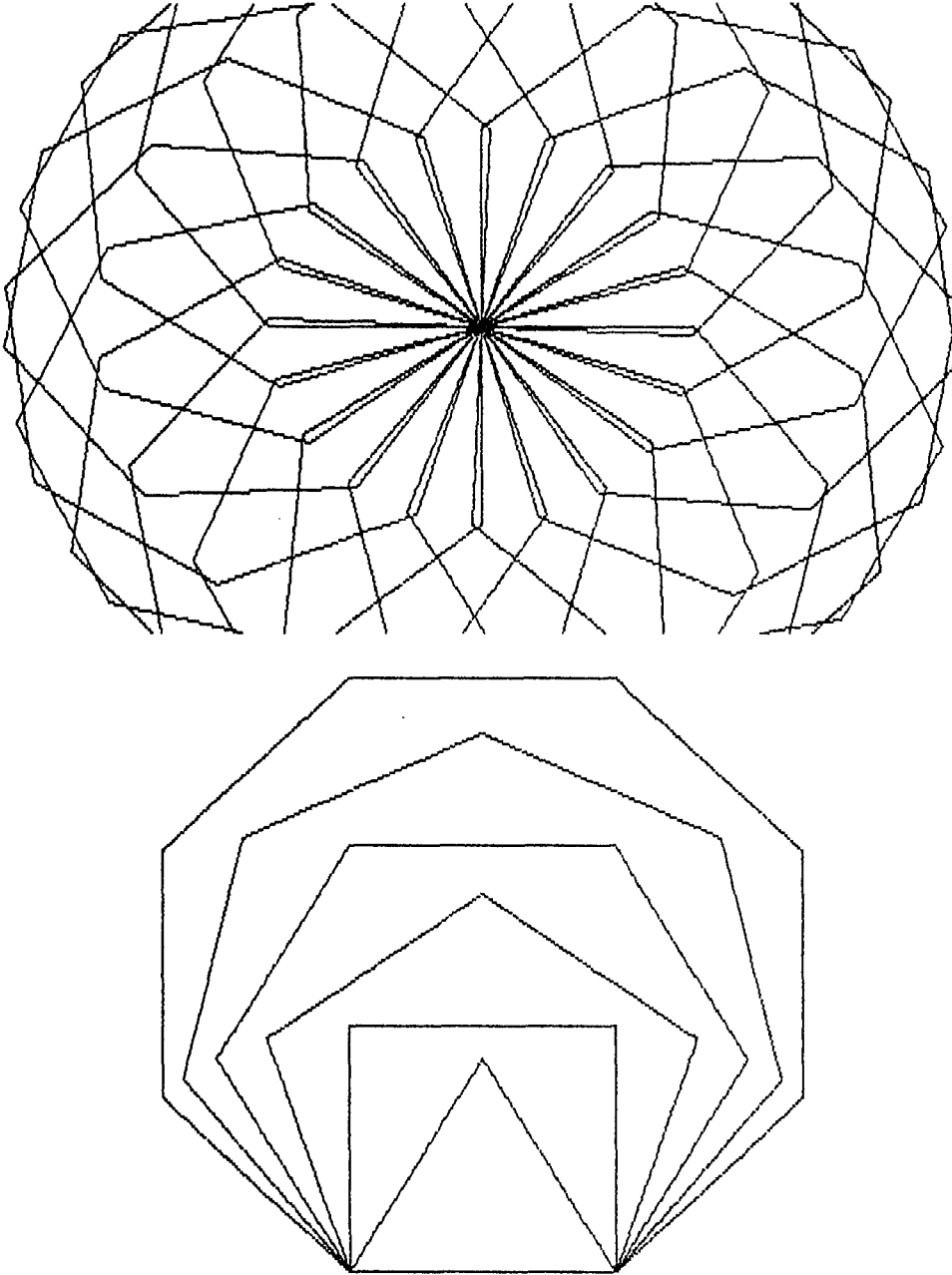


fig. (II.30.)

II.2.2.3.- Listado de procedimientos - Módulo "polígono"-

```
to ctl :x :y :z
    repeat 2[fd :y rt 90 fd :x rt 90] fd :y
    rt 65 fd :z rt 25 fd :x rt 90 fd :y
    rt 65 fd :z rt 115 fd :y rt 65 fd :z
end

to pit :n
    repeat :n[type char 7]
end

to pol :lado :n
    pols :lado :n
    setcursor [10 28]
    pr[SI QUIERE EL DIBUJO SIN NOTAS AL PIE ESCRIBA pols EN VEZ DE pol]
    setcursor [15 29]
    pr[PARA VOLVER AL MENU ANTERIOR ESCRIBA pl Y PULSE RETURN]
    setcursor [0 30]
end

to recuadro
    setsplit 2 ht ts ct pit 1 pu home setpos [-103 185] pd
    ctl 195 40 0 pu home setpos [-108 180] pd ctl 205 50 10
    pu home setpos [-300 -120] pd ctl 600 270 0 pu home
    setpos [-305 -125] ps ctl 610 280 10 pu home setpos [-305 -195]
    pd ctl 610 50 10 setcursor [34 3] pr[* P O L I G O N O S *]
end

to p
    erall ct fs load "plana plana
end

to pl
    poligono
end
```

```

to poligono
  recuadro setcursor [17 10]
  pr[ PARA DIBUJAR CUALQUIER POLIGONO BASTA CON APORTAR COMO]
  setcursor [16 12]
  pr[ DATOS EL NUMERO DE LADOS Y LA MEDIDA DE UNO DE LOS LADOS]
  setcursor [20 14]
  pr[ EJEMPLO: ESCRIBA ..pol 125 6.. Y PULSE RETURN.]
  setcursor [20 18]
  pr[ EL PROGRAMA DIBUJARA UN EXAGONO DE LADO = 125]
  setcursor [13 25]
  pr[ * SI QUIERE VOLVER AL MENU ANTERIOR ESCRIBA p Y PULSE RETURN *]
  setcursor [0 30] pu home pd pit 2
end

```

```

to pols :lado :n
  make "ang (:n-2)*180/:n
  repeat :n[fd :lado rt 180-:ang] ern [ang]
end

```

II.2.3.- Dibujo de rectas

Al seleccionar el apartado (3) de la figura (II.3.), se ejecuta el procedimiento 'c' del módulo "plana".

```
Listado del procedimiento 'c':      to c
                                     erall fs
                                     load "recta recta
                                     end
```

Al cargarse el módulo "recta" en el espacio de trabajo también se cargan los procedimientos 'tri', 'div', 'ejes', 'pit', 'recuadro', 'p', 're', 'recta' y 'ctl'.

La siguiente instrucción del procedimiento 'c' es la de ejecutar el procedimiento 'recta'.

*** DIBUJO DE RECTAS ***

**PARA DIBUJAR CUALQUIER RECTA BASTA CON APORTAR COMO
DATOS EL VALOR DEL TERMINO INDEPENDIENTE Y LA PENDIENTE**

**EJEMPLO: EN LA ECUACION $y = 65 + 1.2 x$
LA PENDIENTE VALE (+ 1.2) Y EL TERMINO INDEPENDIENTE (65)
PARA DIBUJAR ESTA RECTA HAY QUE ESCRIBIR.. re 65 1.2 .. Y PULSAR RTN.
SI ESCRIBE ejes ANTES DE DIBUJAR LA RECTA
LA DIBUJARA SOBRE UNOS EJES CARTESIANOS**

*** SI QUIERE VOLVER AL MENU ANTERIOR ESCRIBA p Y PULSE RETURN ***

El procedimiento 'recta' presenta en pantalla el menú de la figura II.31.

fig. (II.31.)

Desde la figura (II.31.) se tiene acceso al dibujo de cualquier recta 're' y al menú anterior 'p' .

Si se elige la opción del menú anterior, al escribir p y pulsar return, se ejecuta el procedimiento 'p'. El procedimiento 'p', borra el espacio de trabajo, carga el módulo "plana" y ejecuta el procedimiento 'plana' presentando en pantalla la figura (II.3).

Si se elige la opción del dibujo de rectas tendremos la posibilidad de representar cualquier recta de la que conozcamos el término independiente y la pendiente.

Si se quiere que la recta esté sobre unos ejes de coordenadas, primero se ejecuta el procedimiento 'ejes'. Este procedimiento hace el dibujo de unos ejes cartesianos tabulados.

Al escribir ejes y pulsar return sale en pantalla la figura (II.32.); sobre esta figura se pueden dibujar las rectas que se necesiten representar.

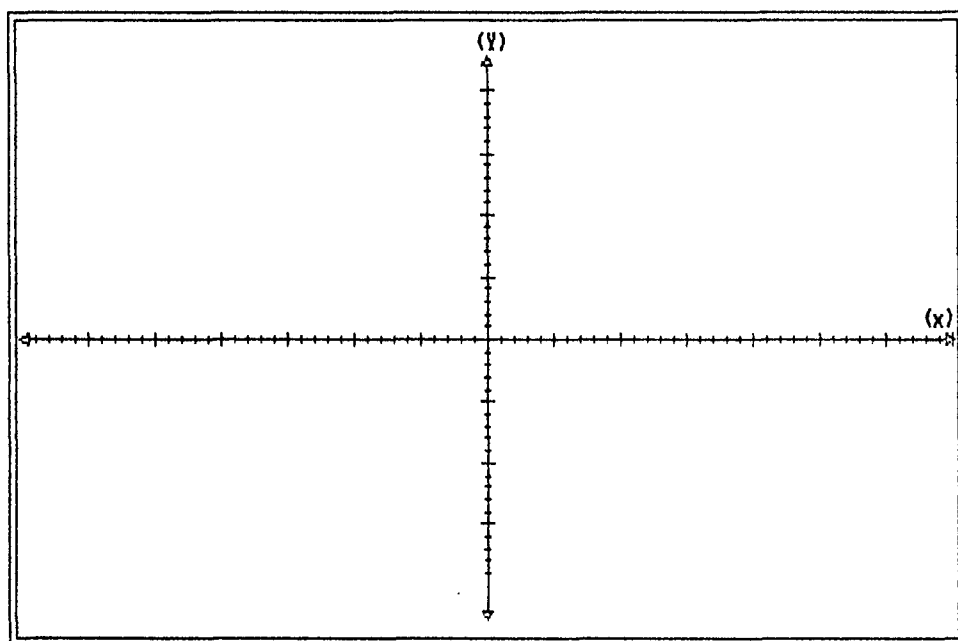
El procedimiento 're' hace el dibujo de la recta teniendo en cuenta el espacio reservado para su representación en la fig.(II.32.).

Como datos se tienen la pendiente y el término independiente. Con la pendiente se calcula el ángulo ($\text{ángulo} = \arctan \text{pendiente}$). Este será el ángulo que forma la recta con la abcisa.

Con estos datos se hacen los cálculos necesarios para que la recta no mida más de 470 entre sus extremos(en vertical). La horizontal está limitada por el mismo procedimiento que la dibuja, y está limitada en 325.

Estos limites son de la pantalla y no de las rectas que se puedan representar.

Cuando el término independiente sea mayor o menor de 580 la recta se saldrá del espacio reservado para su representación en pantalla. El usuario del programa no podrá visualizar estas rectas, aunque el dibujo se realiza por el programa de forma virtual.



El procedimiento 'ejes' muestra en pantalla el dibujo la figura (II.32.).

fig. (II.32.)

II.2.3.2.- Ejemplos de aplicación y listado de procedimientos

Ejemplos de aplicación - (Dibujo de rectas)-

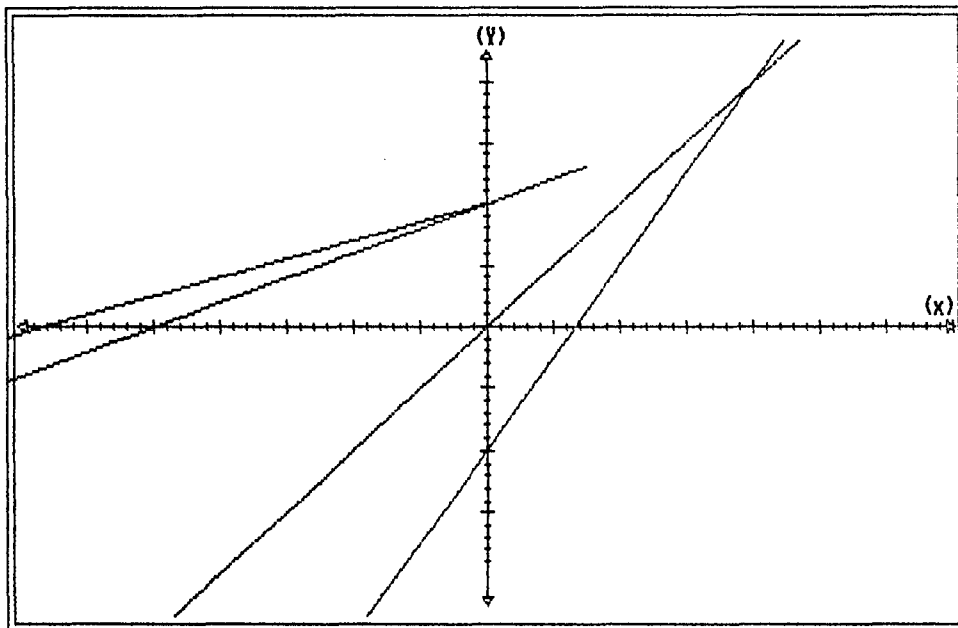


fig. (II.33.)

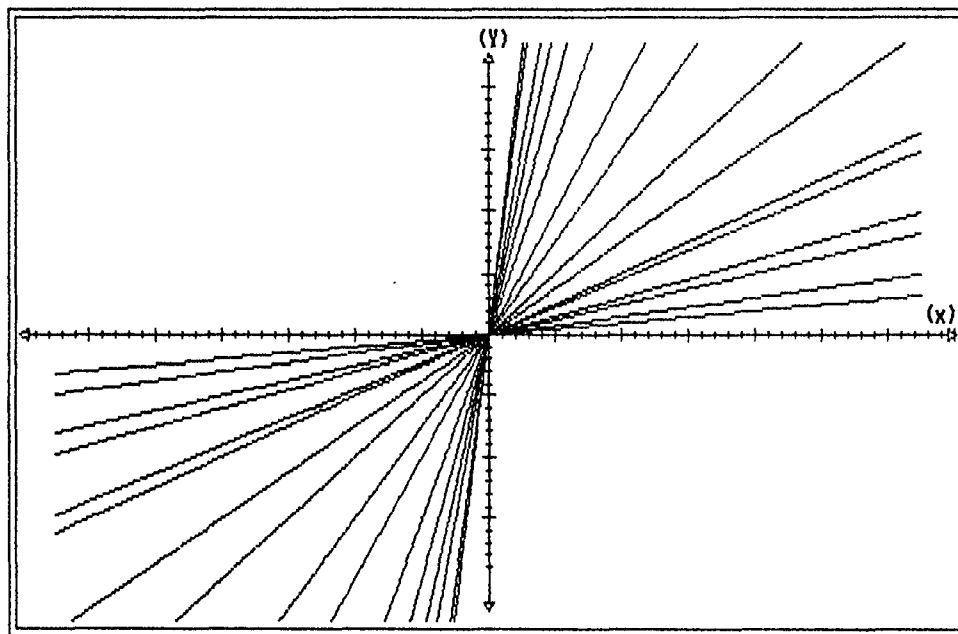


fig. (II.34.)

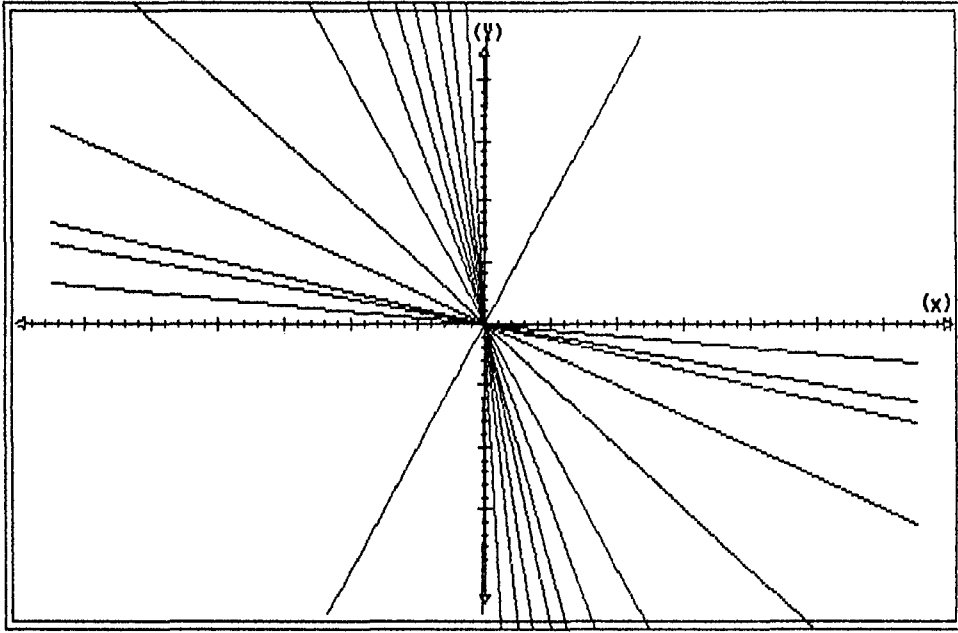


fig. (II.35.)

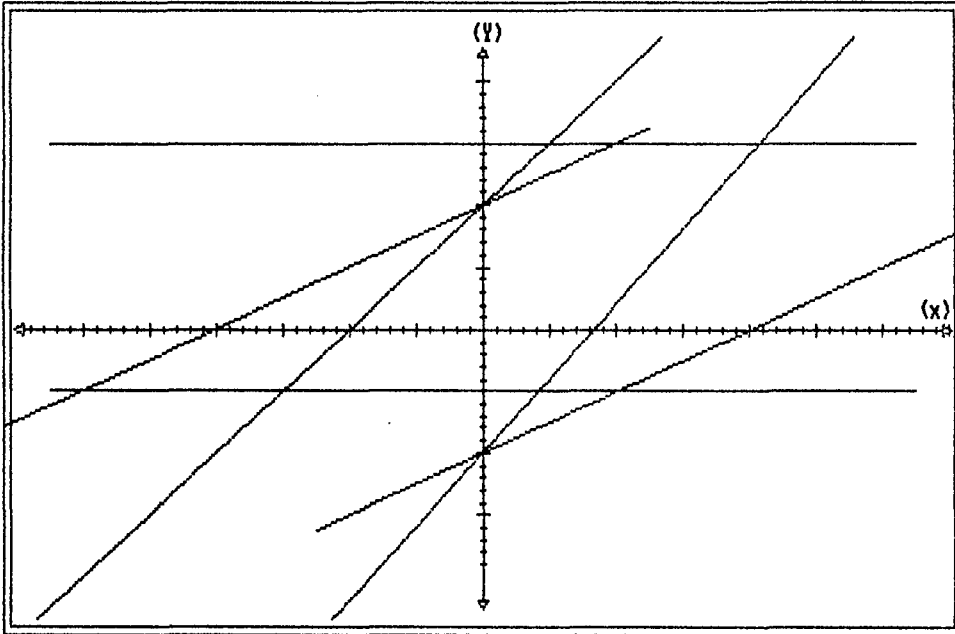


fig. (II.36.)

II.2.3.3.- Listado de procedimientos - Módulo "recta"-

```
to ctl :x :y :z
    repeat 2[fd :y rt 90 fd :x rt 90] fd :y
    rt 65 fd :z rt 25 fd :x rt 90 fd :y
    rt 65 fd :z rt 115 fd :y rt 65 fd :z
end

to pit :n
    repeat :n[type char 7]
end

to p
    erall ct fs load "plana plana
end

to tri :lado
    rt 30 fd :lado repeat 2[rt 120 fd :lado]
end

to div
    pu home setpos [-350 0] pd
    repeat 14[repeat 4[rt 90 pu fd 10 pd lt 90 pd fd 2 bk 4 fd 2] rt 90 fd 10
    lt 90 fd 5 bk 10 fd 5] pu setpos [0 -200] pd lt 90
    repeat 8[repeat 4[rt 90 pu fd 10 lt 90 pd fd 2 bk 4 fd 2]rt 90 fd 10 lt 90
    fd 5]
end

to ejes
    pu home clean setcursor [0 30] pit 1 setpos [-360 -247] pd setsplit 1
    repeat 2[fd 509 rt 90 fd 719 rt 90] pu home setpos [-355 -242] pd
    repeat 2[fd 499 rt 90 fd 709 rt 90] pu home
    pd fd 220 lt 90 fd 4 rt 90 tri 8 rt 180 fd 4 rt 90 fd 440
    lt 90 fd 4 rt 90 tri 8 setcursor [44 1] pr(Y)
    setcursor [86 14] pr (X) setcursor [0 30] pu home pd div pit 2
end
```

```

to recuadro
  setsplit 2 ht ts ct pit 1 pu home setpos [-95 185]
  pd ctl 190 40 0 pu home setpos [-105 180] pd ctl 208 50 10
  pu home setpos [-300 -125] pd ctl 595 275 0 pu home
  setpos [-305 -130] pd ctl 605 285 10 pu home setpos [-305 -195]
  pd ctl 605 50 10 setcursor [35 3]
  pr [* DIBUJO DE RECTAS *]
end

```

```

to re :a :m
  pu home setsplit 1
  make "M 470 make "y :a-325*:m make "ag arctan :m
  make "d 325/cos :ag
  if :ag=0 [make "b 0 make "C3 0 make "C1 2*:a go "1]
  make "C 2*:d*sin :ag make "C1 0 make "C3 0
  if :C>:M [make "C1 :C-:M make "C2 :C1/sin :ag]
  if :C>:M [make "C3 :C2*cos :ag make "d :d-:C2/2]
  make "b (:a*cos :ag)/sin :ag
label "1 setx -325-:b+C3/2 sety :y-:a+:C1/2 pd
rt 90-:ag fd 2*:d setcursor [0 30] pit 1
end

```

```

to recta
  recuadro setcursor [18 8]
  pr[PARA DIBUJAR CUALQUIER RECTA BASTA CON APORTAR COMO]
  setcursor [16 10]
  pr[DATOS EL VALOR DEL TERMINO INDEPENDIENTA Y LA PENDIENTE]
  setcursor [24 13]
  pr[POR EJEMPLO: EN LA ECUACION Y = 65 + 1.2 X] setcursor [15 15]
  pr[LA PENDIENTE VALE (+1.2) Y EL TERMINO INDEPENDIENTE (65)]
  setcursor [10 17]
  pr[PARA DIBUJAR ESTA RECTA HAY QUE ESCRIBIR re 65 1.2 Y PULSAR RTN.]
  setcursor [23 19] pr[SI ESCRIBE ejes ANTES DE DIBUJAR LA RECTA]
  setcursor [24 21] pr[LA DIBUJARA SOBRE UNOS EJES CARTESIANOS]
  setcursor [14 25]
  pr[* SI QUIERE VOLVER AL MENU ANTERIOR ESCRIBA p Y PULSE RETURN *]
  setcursor [0 30] pu home pd pit 2
end

```

II.2.4.- Dibujo de curvas

Al seleccionar el apartado (4) de la figura (II.3.), se ejecuta el procedimiento 'd' del módulo "plana".

El procedimiento 'd' carga el módulo "curva" en el espacio de trabajo, con él también se cargan los procedimientos 'curva', 'pit', 'recuadro', 'p', 'a', 'b' y 'ctl'.

La siguiente instrucción del procedimiento 'd' es la de ejecutar el procedimiento 'curva'.

*** DIBUJO DE CURVAS ***

PARA EL DIBUJO DE LOS DIFERENTES TIPOS DE CURVAS PULSE LAS
TECLAS QUE SE INDICAN

a) CURVAS TRIGONOMETRICAS.....ESCRIBA a Y PULSE RTN.
b) OTRAS CURVAS.....ESCRIBA b Y PULSE RTN.

NOTA: ANTES DE DIBUJAR LAS DIFERENTES CURVAS ESCRIBA ejes Y PULSE RTN.

*** PARA VOLVER AL MENU ANTERIOR ESCRIBA p Y PULSE RTN ***

El procedimiento 'curva' presenta en pantalla el menú de la figura II.37.

(fig. II.37.)

Desde la figura (II.37.) se tiene acceso al dibujo de curvas trigonométricas('a') , otras curvas('b') y al menú anterior('p').

Si se elige la opción del menú anterior, al escribir p y pulsar return, se ejecuta el procedimiento 'p'. El procedimiento 'p', borra el espacio de trabajo, carga el módulo "plana" y ejecuta el procedimiento 'plana' presentando en pantalla la figura (II.3).

II.2.4.1.- Curvas trigonométricas.

Si se elige la opción de curvas trigonométricas, al escribir a y pulsar return, se ejecuta el procedimiento 'a'. Este procedimiento borra el espacio de trabajo, carga el módulo "curvaa" y ejecuta el procedimiento 'curvaa', presentando en pantalla la figura (II.38.).

* CURVAS TRIGONOMETRICAS *

SENO (x) : PARA EL DIBUJO DE ESTA CURVA ESCRIBA seno Y DOS ENTRADAS (AMPLITUD MAXIMA Y FCIA.) . EJEMPLO... seno 100 4 ..Y PULSE RTN.
COSENO (x) : PARA EL DIBUJO DE ESTA CURVA ESCRIBA coseno Y DOS ENTRADAS (AMPLITUD MAXIMA Y FCIA.) . EJEMPLO... coseno 140 6 ..Y PULSE RTN.
TANGENTE (x) : PARA EL DIBUJO DE ESTA CURVA ESCRIBA tang Y DOS ENTRADAS (VALOR MAXIMO Y DEL EJE (x) DONDE COMIENZA A DIBUJAR) . (EJEMPLO...tang 100 -270 ..Y PULSE RETURN)
COTANGENTE (x) :PARA EL DIBUJO DE ESTA CURVA ESCRIBA cotang Y DOS ENTRADAS (AMPLITUD MAXIMA Y FCIA.) . EJEMPLO...cotang 120 -300 ..Y PULSE RTN.)
SECANTE (x) : PARA EL DIBUJO DE ESTA CURVA ESCRIBA sec Y DOS ENTRADAS (AMPLITUD MAXIMA Y FCIA.) . EJEMPLO..sec 80 2 .Y PULSE RTN.
COSECANTE (x) : PARA EL DIBUJO DE LA CURVA ESCRIBA cosec Y DOS ENTRADAS (AMPLITUD MAXIMA Y FCIA.) . EJEMPLO..cosec 55 2 .Y PULSE RTN

* PARA VOLVER AL MENU ANTERIOR ESCRIBA c Y PULSE RTN *

El procedimiento "curvaa" presenta en pantalla el menú de la figura II.38.

fig. (II.38.)

Al cargarse el módulo "curvaa" se cargan los procedimientos 'ejes', 'tri', 'div', 'pit', 'ctl', 'c', 'a', 'recuadro', 'curva1', 'curva2', 'curvaa', 'seno', 'coseno', 'tang', 'cotang', 'sec' y 'cosec'.

Con este módulo se pueden realizar dibujos de todas las curvas trigonométricas básicas. Los dibujos van sobre unos ejes tabulados. Los ejes están divididos por unas marcas. La distancia entre las marcas pequeñas es de 10 y entre las divisiones mayores es de 50 (hay 5 pequeñas entre las mayores).

El dibujo de estos ejes lo ejecuta el procedimiento 'ejes'.

Para la elaboración de los procedimientos que hacen los dibujos de las curvas se han utilizado tres métodos. En el caso de las trigonométricas se han utilizado dos métodos diferentes.

En los dos métodos utilizados en las curvas trigonométricas, primero se calcula el valor de la función para un valor de (x) y posteriormente se dibuja un punto en el plano $(X-Y)$ cuyas coordenadas son el valor de (x) para el que ha sido calculada la función y el valor que ha dado la función en ese punto.

La diferencia entre los dos métodos utilizados está en el algoritmo utilizado para calcular los valores sucesivos donde hemos acotado la función para su representación.

Excepto las funciones `-tangente-` y `-cotangente-` el método utilizado es el de hacer un bucle con dos contadores. Estos dos contadores van aumentando el valor de la coordenada (x) y el valor del ángulo. Cuando (x) vale cero se da por terminado el dibujo.

Ejemplo: - Listado del procedimiento 'seno' -

```
to seno :a :b
  make "x -360 make "alfa 360/:b [Se inicializan los contadores]
label "1 make "y :a*sin (:b*:alfa) [Se calcula el valor de Y(x)]
  dot list :x :y dot list -:x -:y[Se dibuja el punto y su simétrico]
  make "x :x+2 make "alfa :alfa+1[Se incrementan los contadores]
  if :x>0 [recycle stop] go "1 [Se controla la ejecución]
end
```

La amplitud máxima la da el valor de (:a) y la frecuencia el valor de (:b). Debido a la simetría de la curva $\text{seno}(x)$ sólo es necesario el cálculo de la mitad de los puntos para poder representar toda la curva.

Por este motivo, cuando el valor de (x) es mayor de cero se hace una reorganización de la memoria(`recycle`) y se concluye el procedimiento(`stop`).

Para el dibujo de la `-tangente-` y la `-cotangente-` se ha utilizado el método de la recursividad. Este método consiste en ejecutar el procedimiento a sí mismo variando alguna o todas las variables locales. De esta manera se consigue el mismo efecto que haciendo un bucle.

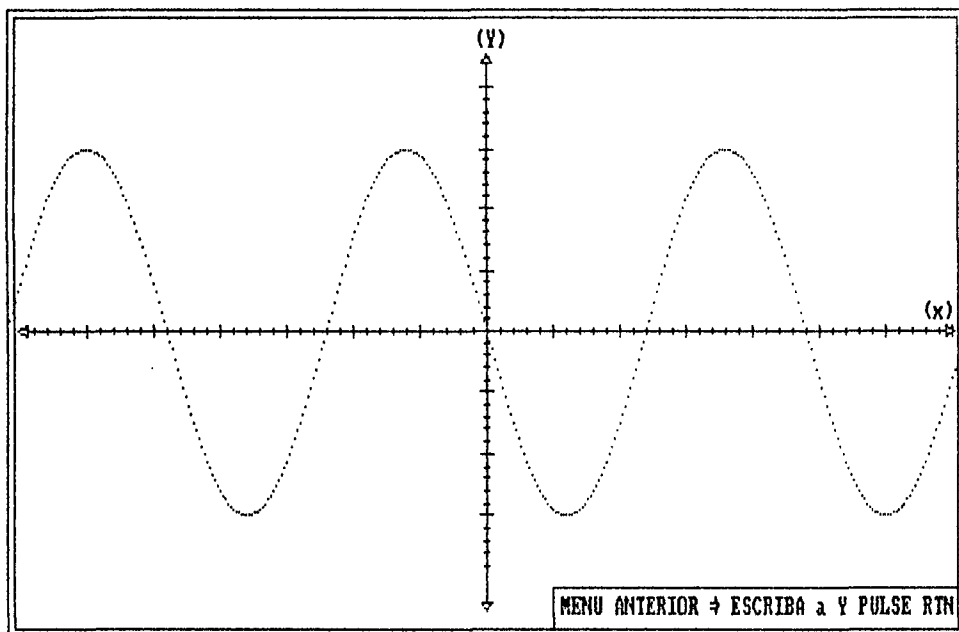
Ejemplo: - Listado del procedimiento 'tangente' -

```
to tang :a :x
  if :x>0 [stop] [Se controla la ejecución]
  if or (:x=-90) (:x=-270) [make "x :x+1] [Estos valores dan ∞]
label "1 make "y :a*sin (:x)/cos (:x) [Se calcula Y(x)]
  if :y>240 [make "x :x+2 if or :x=-270 :x=-90 [make "x :x+1] go "1
  [Si Y(x) > 240 no se representa y se calcula el próximo valor]
  dot list :x :y dot list -:x -:y [Se representa los puntos]
  tang :a :x+2 [(recursividad)Se ejecuta de nuevo cambiando la (x)]
end
```

Para tener el espacio de trabajo lo más libre posible durante la ejecución de las curvas trigonométricas, se hace lo siguiente:

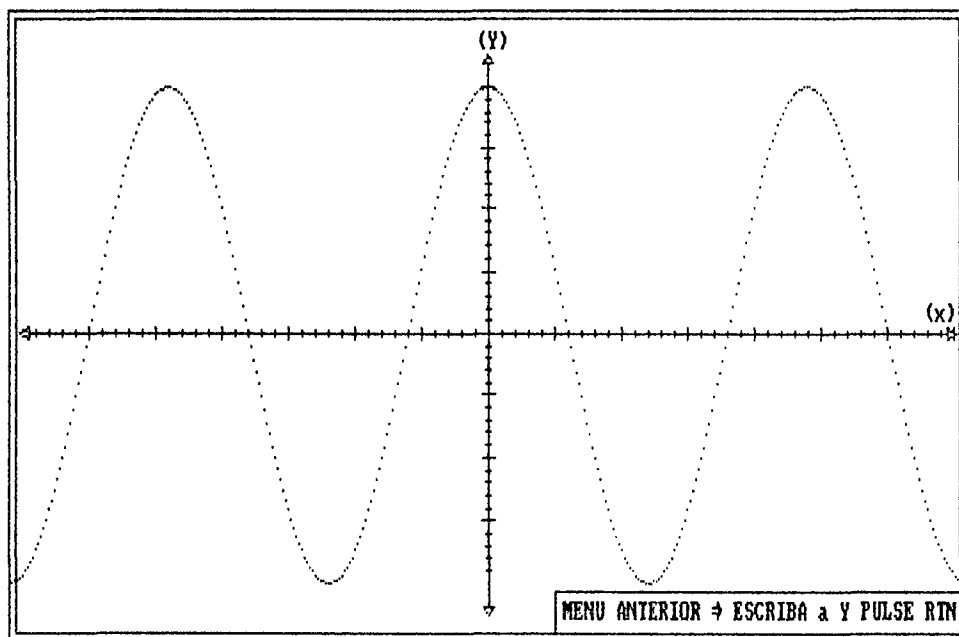
Al ejecutar el procedimiento 'ejes', antes de dibujar la curva, se borran todos los procedimientos de texto y se informa al usuario que para volver al menú anterior escriba - a - y pulse return. Con esta operación se cargan en el espacio de trabajo los procedimientos que anteriormente borró 'ejes' y se ejecuta el procedimiento 'curvaa'.

II.2.4.2.- Ejemplos de aplicación (- Dibujo de curvas -)



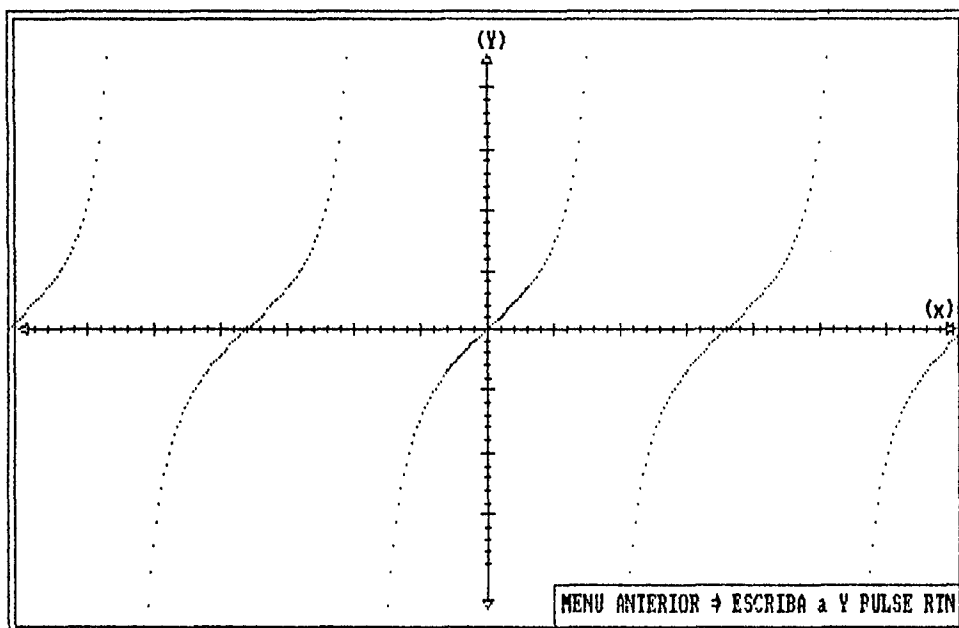
Ejemplo de
seno (x).

fig. (II.39.)



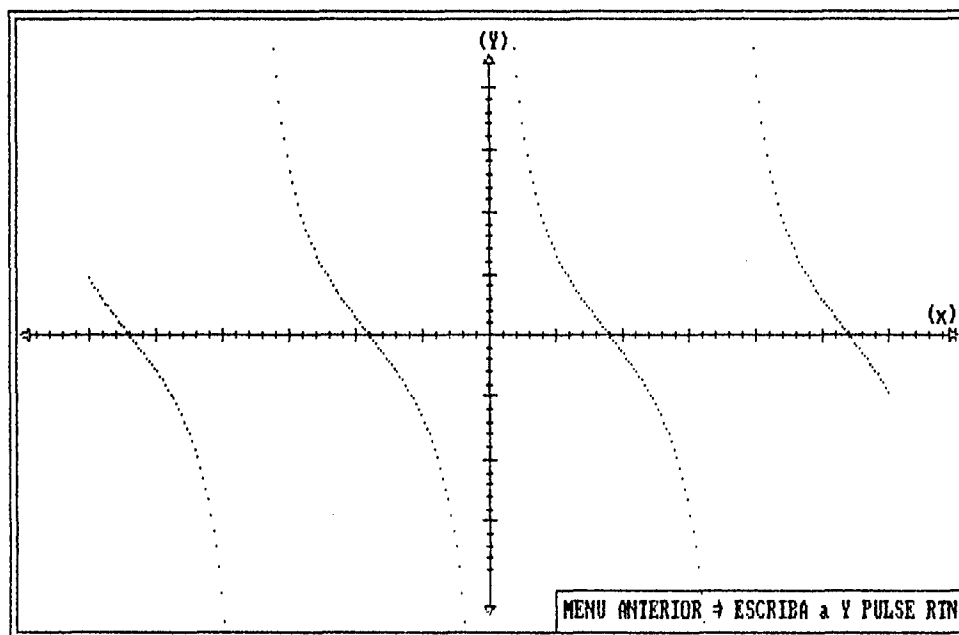
Ejemplo de
cos (x).

fig. (II.40.)



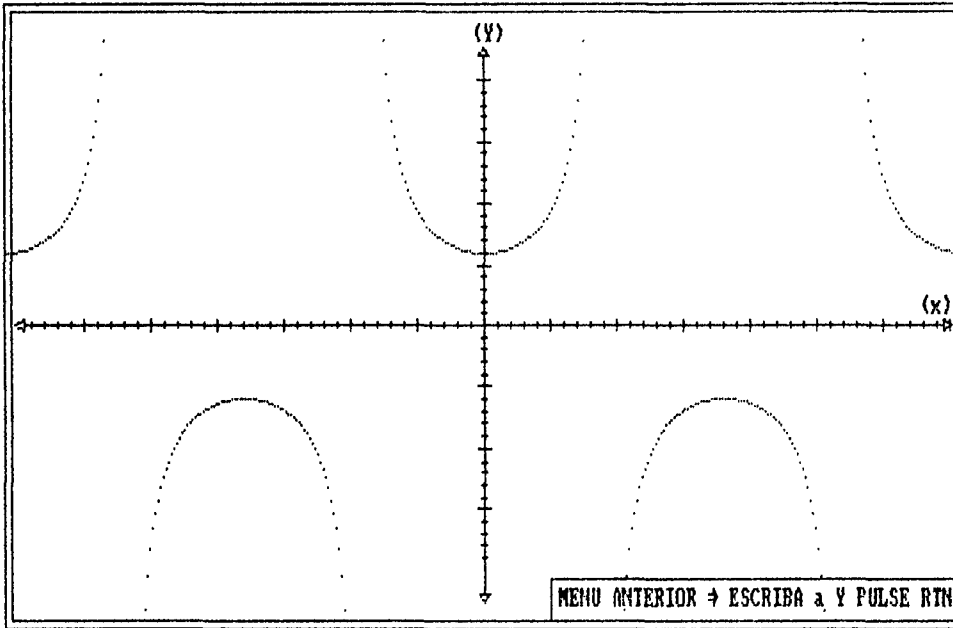
Ejemplo de
tangente(x).

fig. (II.41.)



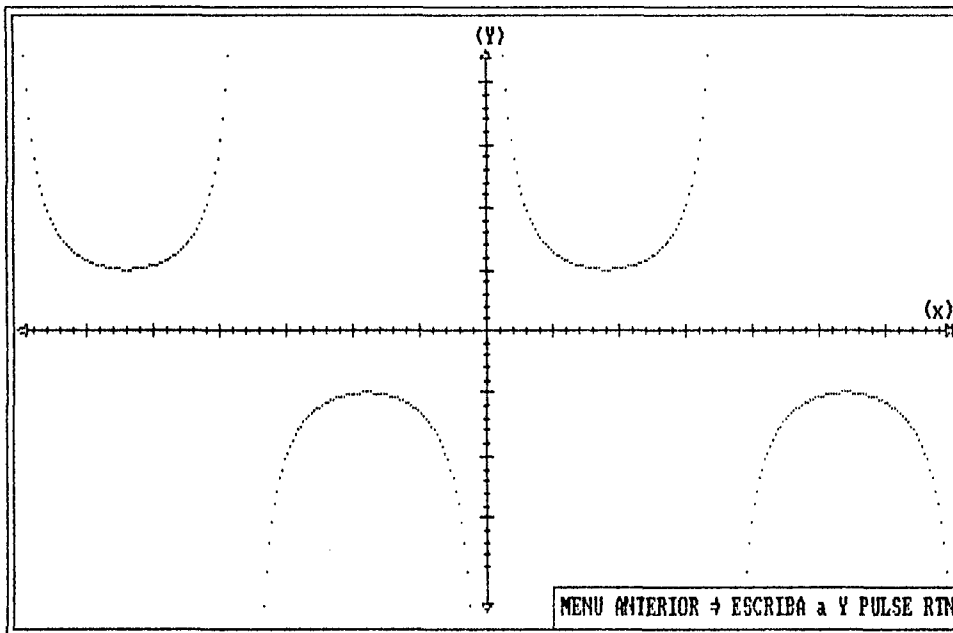
Ejemplo de
cotang(x).

fig. (II.42)



Ejemplo de
secante (x) .

fig. (II.43.)



Ejemplo de
cosecante (x) .

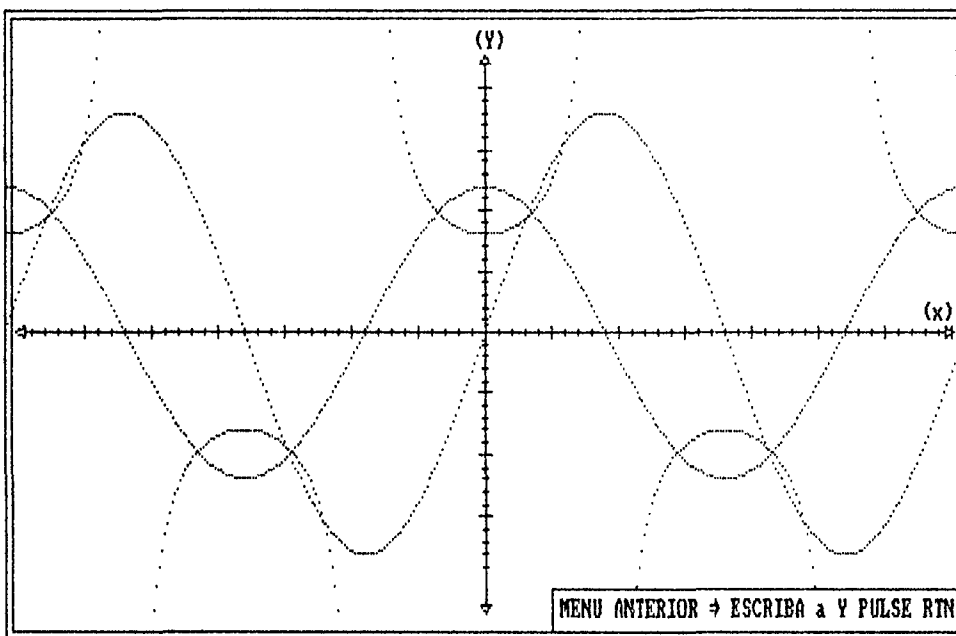
fig. (II.44.)

En el siguiente ejemplo están representadas las curvas seno(x), coseno(x) y secante(x). En el dibujo se puede apreciar la diferencia de fase entre las curvas seno(x) y coseno(x), así como la diferencia de amplitudes. Estas curvas concuerdan con los ejemplos siguientes:

seno 180 2

coseno 120 2

sec 80 2



Ejemplo de
coseno(x),
secante(x)
y seno(x).

fig. (II.45)

II.2.4.3.- Otras curvas

Si se elige la opción (2) del menú de la figura II.38. se carga el módulo "curvab". Al cargarse este módulo se cargan los procedimientos 'tri', 'div', 'ejes', 'pit', 'ctl', 'c', 'a', 'recuadro', 'curva1', 'curva2', 'curvab', 'sa', 'oc', 'ot', 'cir', 'pby' y 'pbx'.

La próxima instrucción ejecuta el procedimiento 'curvab'. Este procedimiento presenta en pantalla la figura (II.46.). Esta figura es el menú de acceso al dibujo de varias curvas.

*** DIBUJO DE CURVAS ***

PARABOLA: LA ECUACION GENERAL ES $(Y = A * X^2 + B * X + C)$
PARA DIBUJAR CUALQUIER PARABOLA SE ESCRIBE pby Y TRES ENTRADAS. LAS
ENTRADAS SON LOS TERMINOS (A - B - C) . PARA INVERTIR LOS EJES PONER phx
(EJEMPLO: ESCRIBA.. pby 1.e-02 0.5 -60 ..Y PULSE RETURN)

CIRCUNFERENCIA: LA ECUACION GENERAL ES $(X^2 + Y^2 = R^2)$
PARA DIBUJAR CUALQUIER CIRCUNFERENCIA SE ESCRIBE cir Y TRES ENTRADAS
LAS ENTRADAS SON EL RADIO Y LAS COORDENADAS (XY) DEL CENTRO.
(EJEMPLO: ESCRIBA.. cir 80 -100 50 .. Y PULSE RETURN)

DIAGRAMA ESPECTRAL SENO (x) / (x) : SU ECUACION ES $Y = \text{SENO}(x) / (x)$
PARA REPRESENTAR ESTA CURVA SE ESCRIBE sa Y TRES ENTRADAS. LAS
ENTRADAS SON: AMPLITUD MAXIMA, FCIA. Y ANGULO (entre -360 y 0)
(EJEMPLO: ESCRIBA sa 150 5 -200 Y PULSE RETURN)

ONDA CUADRADA: ESCRIBA oc Y TRES ENTRADAS, (AMPLITUD, FCIA. Y OFFSET)
(EJEMPLO: ESCRIBA oc 100 5 60 Y PULSE RTN.)

ONDA TRIANGULAR: ESCRIBA ot Y TRES ENTRADAS, (AMPLITUD, FCIA. Y OFFSET)
(EJEMPLO: ESCRIBA ot 150 7 -50 Y PULSE RTN.)

*** PARA VOLVER AL MENU ANTERIOR ESCRIBA c Y PULSE RTN ***

El procedimiento
'curvab' presenta
el menú de la
figura II.46.

fig. (II.46.)

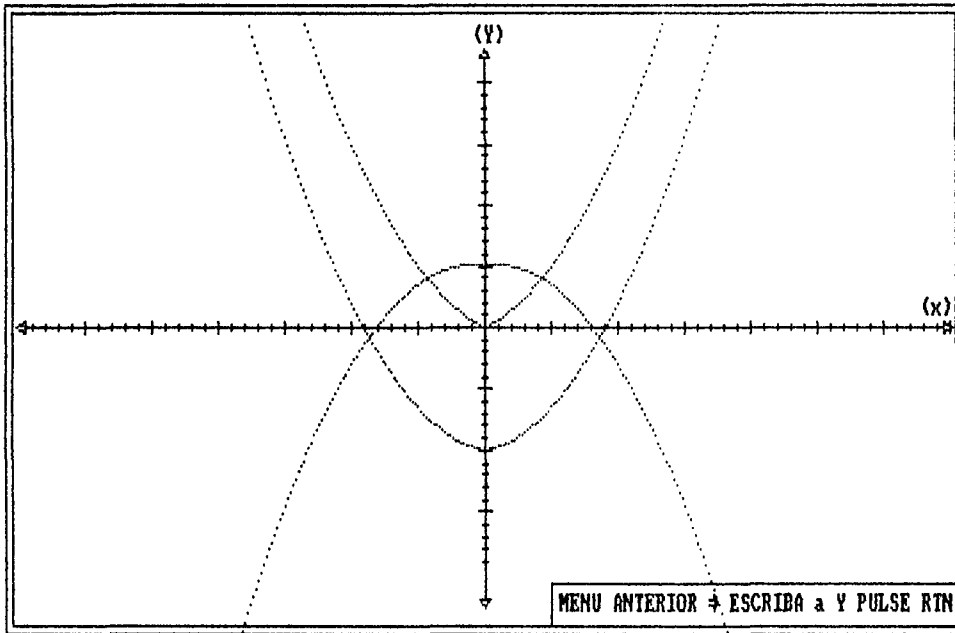
Desde el menú de la figura II.46 se tiene acceso al dibujo de cualquier parábola, circunferencia, onda cuadrada, onda triangular y diagrama espectral de $\text{seno}(x)/(x)$.

El tercer método para el dibujo de curvas es el utilizado en la elaboración de las ondas rectangulares y triangulares. Este método es mucho más rápido que los otros dos y el dibujo de las curvas es prácticamente instantáneo. Para el dibujo se utiliza la tortuga. Se puede utilizar este método porque no es necesario el uso de funciones en el cuerpo del procedimiento. En las otras curvas es inevitable el usar el método de cálculo de puntos para su posterior representación.

- Listado del procedimiento 'ot' (dibujo de la onda triangular) -

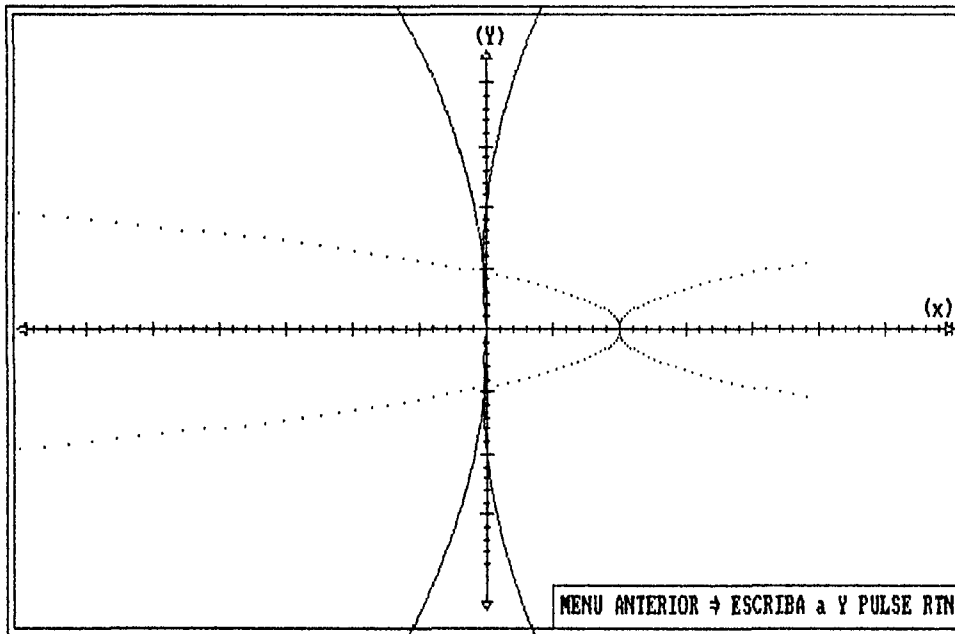
```
to ot :y :f :y1
  make "alfa arctan (2*:f*:y/325)
  [Se calcula el ángulo de la pendiente de la onda]
  make "a :y/sin :alfa
  [Se calcula el lado del triángulo]
  pu home setx -325 sety :y1 pd
  [Se situa la tortuga para comenzar a dibujar]
  repeat :f[rt 90-:alfa fd :a rt 2*:alfa fd 2*:a lt 2*:alfa
  fd :a lt 90-:alfa]
  [Se dibuja la señal triangular]
end
```

II.2.4.4.- Ejemplos de aplicación - Dibujo de curvas -



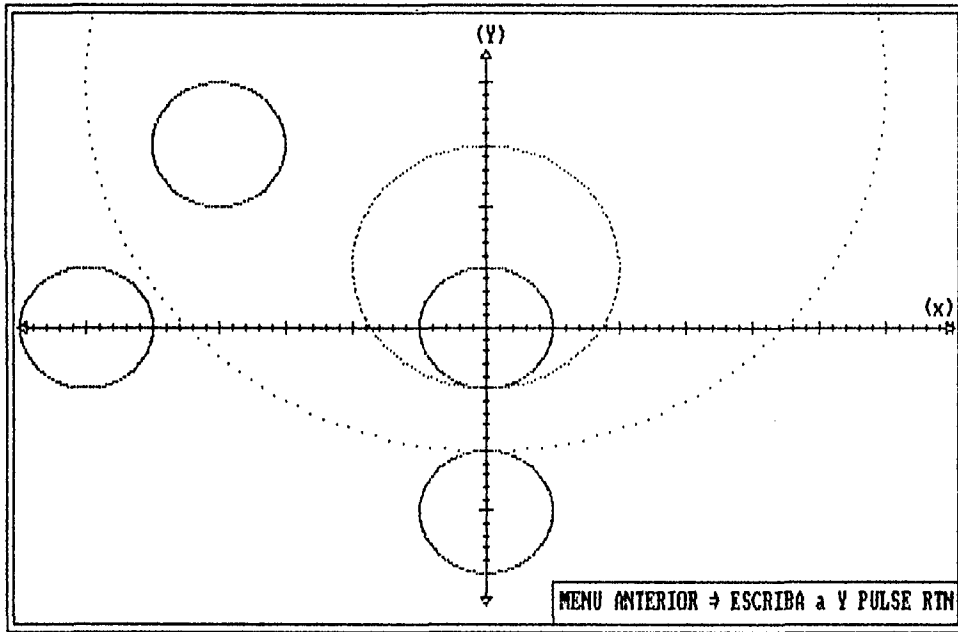
Ejemplo de
parábola (pby)

fig. (II.47.)



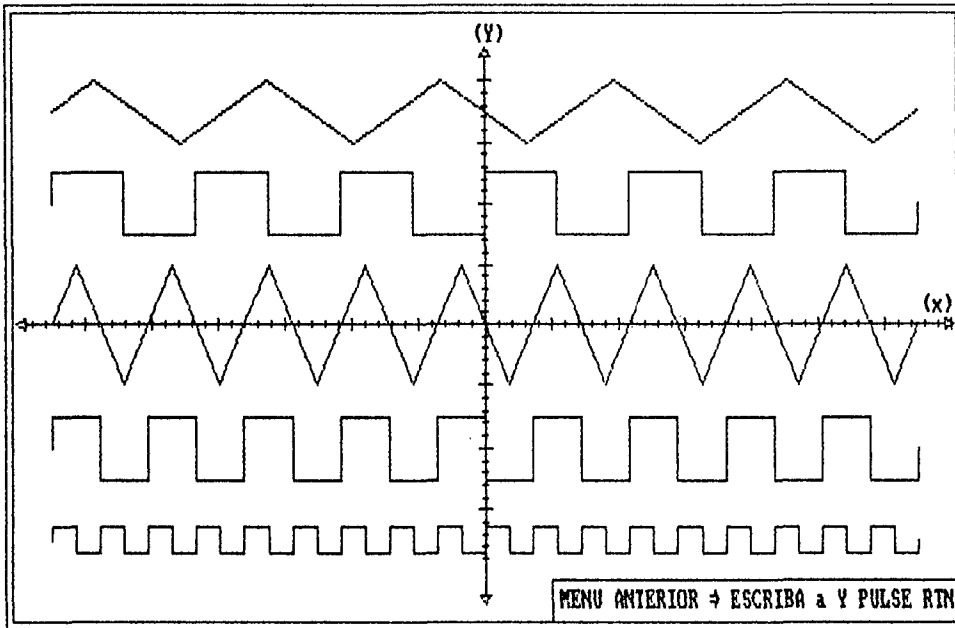
Ejemplo de
parábola con
inversión de
ejes (pbx).

fig. (II.48.)



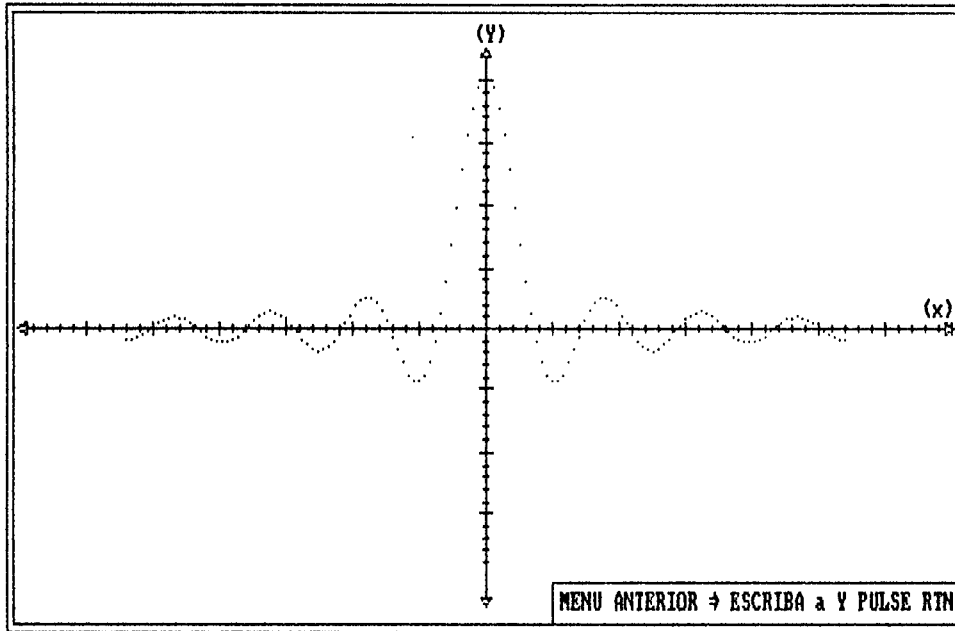
Varios Ejemplos
del dibujo de
circunferencias.

fig. (II.49.)



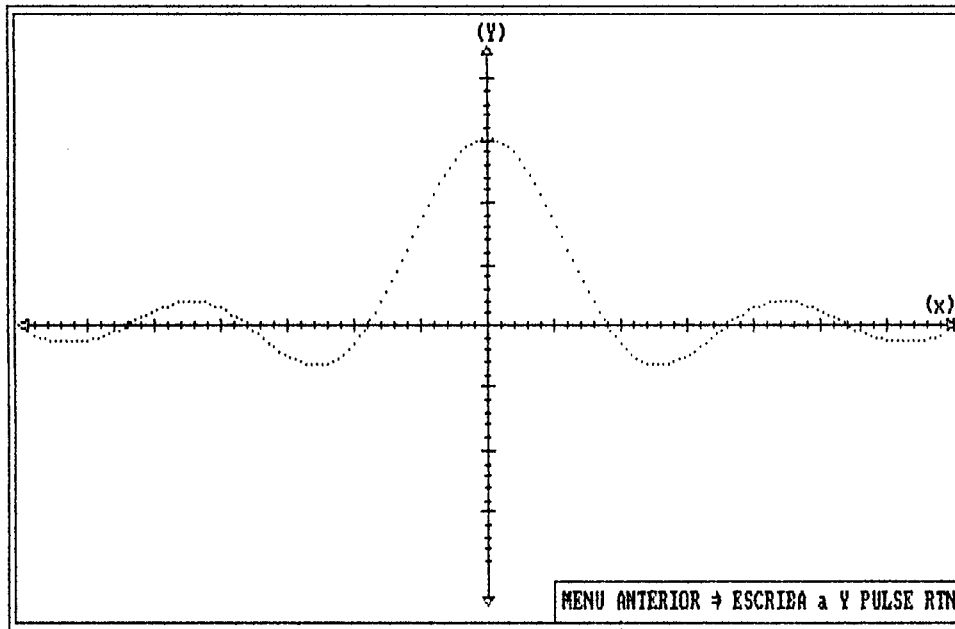
Ejemplos de la
onda cuadrada y
triangular.

fig. (II.50.)



Ejemplo de
 $\text{seno}(x)/x$.

fig. (II.51.)



Ejemplo de
 $\text{seno}(x)/x$.

fig. (II.52.)

En la fig. (II.53) se representan ejemplos de la parábola, recta y circunferencia. En el dibujo se pueden apreciar los puntos de corte de las diferentes curvas. Estos puntos de corte coinciden con los hallados de forma analítica. Los ejemplos utilizados son los siguientes:

```
re 0 1 re 100 0  
cir 120 100 100 cir 80 -250 0  
pby 0.01 0.1 -60
```

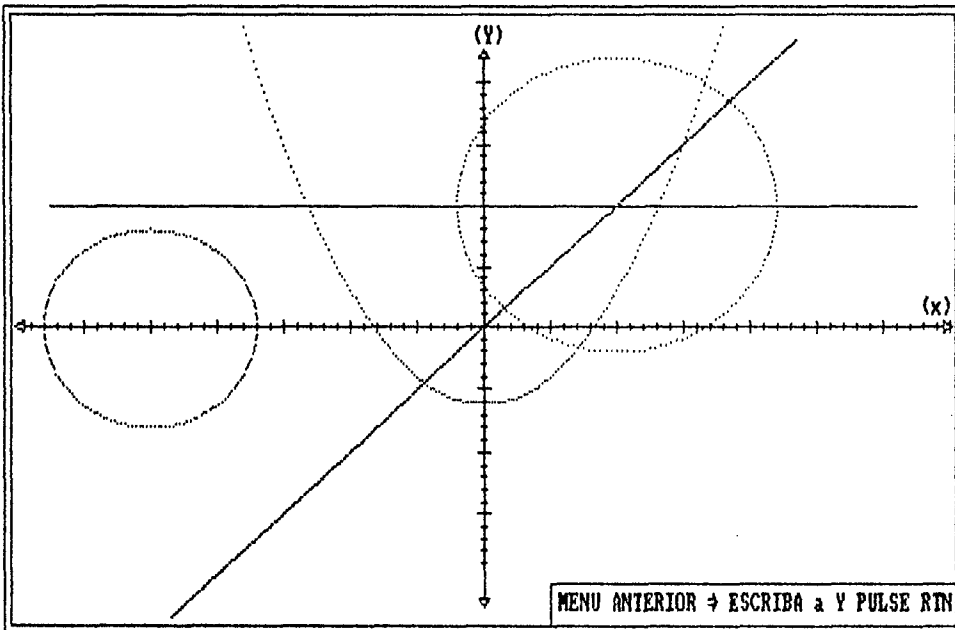


fig. (II.53.)

II.2.4.5.- Listado de procedimientos

Módulo "curva"

```
to ctl :x :y :z
  repeat 2[fd :y rt 90 fd :x rt 90] fd :y
  rt 65 fd :z rt 25 fd :x rt 90 fd :y
  rt 65 fd :z rt 115 fd :y rt 65 fd :z
end

to pit :n
  repeat :n[type char 7]
end

to p
  erall ct fs load "plana plana
end

to a
  erall ct fs load "curvaa curvaa
end

to b
  erall ct fs load "curvab curvab
end

to recuadro
  setsplit 2 ht ts ct pit 1 pu home setpos [-100 200] pd
  ctl 185 40 0 pu home setpos [-110 195] pd ctl 203 50 10
  pu home setpos [-300 -135] pd ctl 595 310 0 pu home
  setpos [-304 -140] pd ctl 605 320 10 pu home setpos [-305 -210]
  pd ctl 605 50 10 setcursor [34 2] pr [* DIBUJO DE CURVAS *]
end
```

```

to curva
recuadro setcursor [15 8]
pr [PARA EL DIBUJO DE LOS DIFERENTES TIPOS DE CURVAS PULSE LAS]
setcursor [34 10] pr [TECLAS QUE SE INDICAN]
setcursor [10 14] pr [a) CURVAS TRIGONOMETRICAS.....]
setcursor [57 14] pr [ESCRIBA a Y PULSE RTN.]
setcursor [10 17] pr [b) OTRAS CURVAS.....]
setcursor [57 17] pr [ESCRIBA b Y PULSE RTN.] setcursor [9 21]
pr [NOTA: ANTES DE DIBUJAR LAS DIFERENTES CURVAS ESCRIBA ejes Y PULSE RTN.]
setcursor [17 26]
pr [* PARA VOLVER AL MENU ANTERIOR ESCRIBA p Y PULSE RTN *]
pit 2 setcursor [0 30]
end

```

Módulo "curvaa"

.....

```

to ejes
er "curvaa er "curva1 er "curva2 er "recuadro pu home clean
setcursor [0 30] pit 1 setpos [-360 -247] pd
repeat 2[fd 509 rt 90 fd 719 rt 90] pu home setpos [-355 -242]
pd repeat 2[fd 499 rt 90 fd 709 rt 90] pu hoe setpos [50 -242]
pd fd 35 rt 90 fd 305 pu home pd fd 220 lt 90 fd 4 rt 90 tri 8
rt 180 rt 4 rt 90 fd 440 lt 90 fd 4 rt 90 tri 8 bk 4 lt 90 fd 220
rt 90 fd 345 lt 90 fd 4 rt 90 tri 8 bk 4 lt 90 fd 690 lt 90 fd 4
rt 90 tri 8 setcursor [52 28]
pr [MENU ANTERIOR → ESCRIBA a Y PULSE RTN] setcursor [44 1]
pr [(Y)] setcursor [86 14] pr [(X)] setcursor [0 30]
pu home pd div recycle pit 2
end

```

```

to tri :l
rt 30 fd :l repeat 2[rt 120 fd :l]
end

```

```

to pit :n
repeat :n[type char 7]
end

```

```

to div
  pu home setpos [-350 0] pd
  repeat 14[repeat 4[rt 90 pu fd 10 pd lt 90 fd 2 bk 4 fd 2]
  rt 90 pu fd 10 lt 90 pd fd 5 bk 10 fd 5]
  pu setpos [0 -200] pd lt 90
  repeat 8[repeat 4[rt 90 pu fd 10 lt 90 pd fd 2 bk 4 fd 2]
  rt 90 fd 10 lt 90 fd 5 bk 10 fd 5]
end

to ctl :x :y :z
  repeat 2[fd :y rt 90 fd :x rt 90] fd :y rt 65 fd :z rt 25
  fd :x rt 90 fd :y rt 65 fd :z rt 115 fd :y rt 65 fd :z
end

to c
  erall ct fs load "curva curva
end

to a
  ct fs load "curvaa curvaa
end

to curvaa
  recuadro curva1 curva2
end

to recuadro
  setsplit 1 ht ts ct pit 1 pu home setpos [-125 185] pd
  ctl 215 40 0 pu home setpos [-135 180] pd ctl 232 50 10
  pu home setpos [-300 185] pd ctl 595 345 0 pu home
  setpos [-305 -190] pd ctl 605 355 10 pu home
  setpos [-305 -240] pd ctl 605 40 10 setcursor [30 3]
  pr [*CURVAS TRIGONOMETRICAS*]
end

```

```

to curva1
  setcursor [10 7]
  pr [SENO(X): PARA EL DIBUJO DE ESTA CURVA ESCRIBA seno Y DOS ENTRADAS]
  setcursor [10 8]
  pr [(AMPLITUD MAXIMA Y FCIA.).EJEMPLO...seno 100 4.. Y PULSE RTN.]
  setcursor [10 10]
  pr [COSENO(X): PARA EL DIBUJO DE ESTA CURVA ESCRIBA coseno Y DOS
  ENTRADAS] setcursor [10 11]
  pr [(AMPLITUD MAXIMA Y FCIA.). EJEMPLO..coseno 140 6..Y PULSE RTN.]
  setcursor [10 13]
  pr [TANGENTE(X): PARA EL DIBUJO DE ESTA CURVA ESCRIBA tang Y DOS]
  setcursor [10 14]
  pr [ENTRADAS (VALOR MAXIMO Y DEL EJE (X) DONDE COMIENZA A DIBUJAR).]
  setcursor [23 15]
  pr [(EJEMPLO...tang 100 -270.. Y PULSE RETURN]
end

```

```

to curva2
  setcursor [10 17]
  pr [COTANGENTE(X): PARA EL DIBUJO DE ESTA CURVA ESCRIBA cotang Y DOS]
  setcursor [10 18]
  pr [ENTRADAS (VALOR MAXIMO Y DEL EJE (X) DONDE COMIENZA A DIBUJAR).]
  setcursor [23 19]
  pr [(EJEMPLO...cotang 120 -300 ..Y PULSE RTN.))
  setcursor [10 21]
  pr [SECANTE(X): PARA EL DIBUJO DE ESTA CURVA ESCRIBA sec Y DOS ENTRADAS]
  setcursor [10 22]
  pr [(AMPLITUD MAXIMA Y FCIA.). EJEMPLO ..sec 80 2..Y PULSE RTN.]
  setcursor [COSECANTE(X): PARA EL DIBUJO DE LA CURVA ESCRIBA cosec Y DOS
  ENTRADAS] setcursor [10 25]
  pr [(AMPITUD MAXIMA Y FCIA.). EJEMPLO cosec 55 2.. Y PULSE RTN.]
  setcursor [17 28]
  pr [* PARA VOLVER AL MENU ANTERIOR ESCRIBA c Y PULSE RTN *]
  pit 2 setcursor [0 30]
end

```

```

to seno :a :b
  make "x -360 make "alfa 360/:b
label "1  make "y :a*sin (:b*:alfa) dot list :x :y
  dot list -:x -:y make "x :x+2 make "alfa :alfa+1
  if :x>0 [recycle stop] go "1
end

```

```

to coseno :a :b
  make "x -360 make "alfa 360/:b
label "1  make "y :a*cos (:b*:alfa)
          if or quotient :b 2 < :b/2 :b=1 [make "y -:y]
          dot list :x :y dot list -:x -:y
          make " x :x+2 make "alfa :alfa+1
          if :x>0 [recycle stop] go "1
end

```

```

to tang :a :x
  if :x>0 [stop]
  if or :x= -90 :x= -270 [make "x :x+1]
label "1  make "y :a*sin (:x)/cos (:x)
          if :y>240 [make "x :x+2 if or :x= -90 :x= -270 [make "x :x+1]
          go "1] dot list :x :y dot list -:x -:y tang :a :x+2
end

```

```

to cotang :a :x
  if :x>0 [stop]
  if or :x< -90 :x< -270 [make "x :x+1]
label "1  make "y :a*cos (:x)/sin (:x)
          if :y>240 [make "x :x+2 if or :x= -270 :x= -90 [make "x :x+1]
          go "1] dot list :x :y dot list -:x -:y cotang :a :x+2
end

```

```

to sec :a :b
  make "x -359 make "alfa :x/:b
label "1  if or :x= -90 :x= -180 [make "x :x+1] make "y :a/cos (:b*:alfa)
          if or quotient :b 2 < :b/2 :b=1 [make "y -:y]
          if :y>245 [make "x :x+2 make "alfa :alfa+1 go "1]
          dot list :x :y dot list -:x :y make "x :x+2
          make "alfa :alfa+1 if :x>0 [recycle stop] go "1
end

```

```

to cosec :a :b
  make "x -359 make "alfa :x/:b
label "1  if or :x= -90 :x= -270 [make "x :x+1] make "y :a/sin (:b*:alfa)
          if :y>250 [make "x :x+2 make "alfa :alfa+1 go "1]
          dot list :x :y dot list -:x -:y make "x :x+2
          make "alfa :alfa+1 if :x>0 [recycle stop] go "1
end

```

Módulo "curvab"

El listado de los procedimientos 'tri', 'pit', 'c', 'div' y 'ctl' son iguales que en el módulo anterior y el procedimiento 'ejes' sólo difiere en la primera instrucción, que en este caso es (er "curvab) en lugar de (er "curvaa).

```
to a
  ct fs load "curvabb curvab
end
```

```
to curvab
  recuadro curva1 curva2
end
```

```
to recuadro
  setsplit 1 ht ts ct pit 1 pu home setpos [-100 200] pd
  ctl 185 40 0 pu home setcursor [34 2]
  pr [*DIBUJO DE CURVAS*] setpos [-110 195] pd ctl 203 50 10
  pu home setpos [-300 -185] pd ctl 595 360 0 pu home
  setpos [-305 -190] pd ctl 605 370 10 pu home
  setpos [-305 -240] pd ctl 605 40 10
end
```

```
to pby :a :b :c
  make "x -300
label "1
  make "y :a*:x*:x+:b*:x+:c
  if :y>250 [make "x :x+2 go "1]
  if :x>0 [recycle stop]
  dot list :x :y dot list -:x :y
  make "x :x+2 go "1
end
```

```

to pbx :a :b :c
    Igual que 'pby' excepto que en lugar de
    (dot :x :y dot -:x :y) se pone
    (dot :y :x dot :y -:x)
end

to cir :r :x :y
    make "a 0
label "1
    make "y1 :r*sin :a make "x1 :r*cos :a
    dot list :x+:x1 :y+:y1 dot list :x-:x1 :y+:y1
    dit list :x-:x1 :y-:y1 dot list :x+:x1 :y-:y1
    if :a>90 [recycle stop] make "a :a+2 go "1
end

to sa :a :b :x
    if :x=0 [make "x 0.01] if :x= -180 [recycle]
    make "A (sin :b*:x)/(cos :b*:x) make "y 58*:a*:A
    dot list :x :y dot list -:x :y
    if :x>0 [recycle stop] sa :a :b :x+4
end

to oc :y :f :y1
    make "x 325/:f pu home setx -325 sety :y1 pd ht
    repeat :f[fd :y rt 90 fd :x rt 90 fd 2*:y lt 90 fd :x lt 90 fd :y]
end

to ot :y :f :y1
    make "alfa arctan (2*:f*:y/325) make "a :y/sin :alfa
    pu home setx -325 sety :y1 pd
    repeat :f[rt 90-:alfa fd :a rt 2*:alfa fd 2*:a lt 2*:alfa fd :a
    lt 90-:alfa]
end

```



```

to curva1
  setcursor [16 6]
  pr[PARABOLA: LA ECUACION GENERAL ES (Y= A * X2 + B * X + C)]
  setcursor [12 7]
  pr[PARA DIBUJAR CUALQUIER PARABOLA SE ESCRIBE pby Y TRES ENTRADAS, LAS]
  setcursor [9 8]
  pr[ENTRADAS SON LOS TERMINOS (A-B y C). PARA INVERTIR LOS EJES PONER pbx]
  setcursor [17 9]
  pr[(EJEMPLO: ESCRIBA .pby 0.01 0.5 -60. Y PULSE RETURN)]
  setcursor [CIRCUNFERENCIA: LA ECUACION GENERAL ES (X2 + Y2 = R2)]
  setcursor [12 12]
  pr[PARA DIBUJAR CUALQUIER CIRCUNFERENCIA SE ESCRIBE cir Y TRES ENTRADAS]
  setcursor [15 13]
  pr [LAS ENTRADAS SON EL RADIO Y LAS COORDENADAS (X-Y) DEL CENTRO.]
  setcursor [18 14]
  pr[(EJEMPLO: ESCRIBA .cir 80 -50 80. Y PULSE RETURN)]
end

```

```

to curva2
  setcursor [9 16]
  pr[DIAGRAMA ESPECTRAL SENO(X)/(X): SU ECUACION ES Y = SENO(X)/(X)]
  setcursor [12 17]
  pr[PARA REPRESENTAR ESTA CURVA SE ESCRIBE sa Y TRES ENTRADAS, LAS]
  setcursor [12 18]
  pr[ENTRADAS SON: AMPLITUD MAXIMA, FCIA. Y ANGULO. (entre -360 y 0)]
  setcursor [21 19]
  pr[(EJEMPLO: ESCRIBA .sa 150 5 -200 Y PULSE RETURN)]
  setcursor [11 21]
  pr[ONDA CUADRADA: ESCRIBA oc Y TRES ENTRADAS, (AMPLITUD, FCIA. Y OFFSET)]
  setcursor [23 22]
  pr[(EJEMPLO: ESCRIBA oc 100 5 60 Y PULSE RTN.)]
  setcursor [10 24]
  pr[ONDA TRIANGULAR: ESCRIBA ot Y TRES ENTRADAS, (AMPLITUD FCIA. Y OFFSET)]
  setcursor [23 25]
  pr[(EJEMPLO: ESCRIBA ot 150 7 -50 Y PULSE RTN.)]
  setcursor [17 28]
  pr[*PARA VOLVER AL MENU ANTERIOR ESCRIBA c Y PULSE RTN. *]
  pit 2 setcursor [0 30]
end

```

II.3.- Geometría del espacio.

Al seleccionar el apartado (2) de la figura (II.1.), se ejecuta el procedimiento 'e' del módulo "bdatos".

Este procedimiento carga el módulo "espacio". Al cargarse el módulo "espacio" también se cargan los procedimientos 'espacio', 'prisma', 'piramide', 'cubo', 'ctl', 'bd', 'ventana' y 'pit'.

La siguiente instrucción del procedimiento 'e' es la de ejecutar el procedimiento 'espacio'.

DIBUJO Y CALCULO DE FIGURAS TRIDIMENSIONALES

CON ESTE MENU PUEDE ACCEDER AL CALCULO Y DIBUJO
DE LAS FIGURAS QUE SE RELACIONAN A CONTINUACION

- 1) PRISMA.....ESCRIBA prisma Y PULSE RETURN.
- 2) PIRAMIDE.....ESCRIBA piramide Y PULSE RETURN.
- 3) CUBO.....ESCRIBA cubo Y PULSE RETURN.
- 4) OCTAEDRO.....ESCRIBA octaedro Y PULSE RETURN.

* SI QUIERE VOLVER AL MENU ANTERIOR ESCRIBA bd Y PULSE RETURN *

El procedimiento 'espacio' presenta en pantalla el menú de la fig. II.54.

(fig. II.54.)

Desde la figura (II.54.) se tiene acceso al dibujo y cálculo de cualquier prisma, pirámide, cubo, octaedro y al menú anterior.

Si se elige la opción del módulo anterior, al escribir bd y pulsar return, se ejecuta el procedimiento 'bd'. El procedimiento 'bd', borra el espacio de trabajo, carga el módulo "bdatos" y ejecuta el procedimiento 'bd' presentando en pantalla la figura (II.1).

- P R I S M A -

II.3.1.- Cálculo y dibujo del prisma recto

II.3.1.1.- Módulo - Dibujo - y procedimientos

Si en la figura II.54 elegimos la opción (1), al escribir prisma y pulsar return, se ejecuta el procedimiento 'prisma' del módulo "espacio".

El procedimiento 'prisma' borra el espacio de trabajo y carga el módulo "prisma".

Con este módulo se puede dibujar cualquier prisma y también se puede acceder al módulo de cálculo del prisma.

Con el módulo "prisma" se cargan los procedimientos 'sp', 'cp', 'ctl', 'pri', 'pris', 'recuadro', 'pit', 'prisma' y 'pris'.

La proxima instrucción es la de ejecutar el procedimiento 'prisma' del módulo "prisma". Al ejecutarse este procedimiento sale en pantalla el dibujo de la figura (II.55.). Este menú informa al usuario como dibujar un prisma y como trasladarse al módulo anterior y al módulo de cálculo del prisma.

DIBUJO Y CALCULO DEL PRISMA RECTO

PARA DIBUJAR CUALQUIER PRISMA VISTO DESDE CUALQUIER ANGULO TIENE QUE ESCRIBIR `pri` SEGUIDO DE CUATRO ENTRADAS, LAS CUATRO ENTRADAS SON LAS MEDIDAS DE LOS TRES LADOS Y DEL ANGULO DE VISION.

EJEMPLO: ESCRIBA `pri 50 200 40 20` Y PULSE `RTN`. EL PROGRAMA DIBUJA UN PRISMA DE 50 DE BASE 200 DE ALTURA 40 DE ANCHO Y VISTO DESDE UN ANGULO DE 20 °.

(PRUEBE EL EJEMPLO `cs.,pri 50 50 40 20` .Y PULSE `RTN`.)

* PARA EL CALCULO DEL PRISMA ESCRIBA `cp` Y PULSE `RTN`. *

* PARA VOLVER AL MENU ANTERIOR ESCRIBA `sp` Y PULSE `RTN`. *

fig. (II.55.)

II.3.1.2.- Ejemplos de aplicación - Dibujo de prismas -

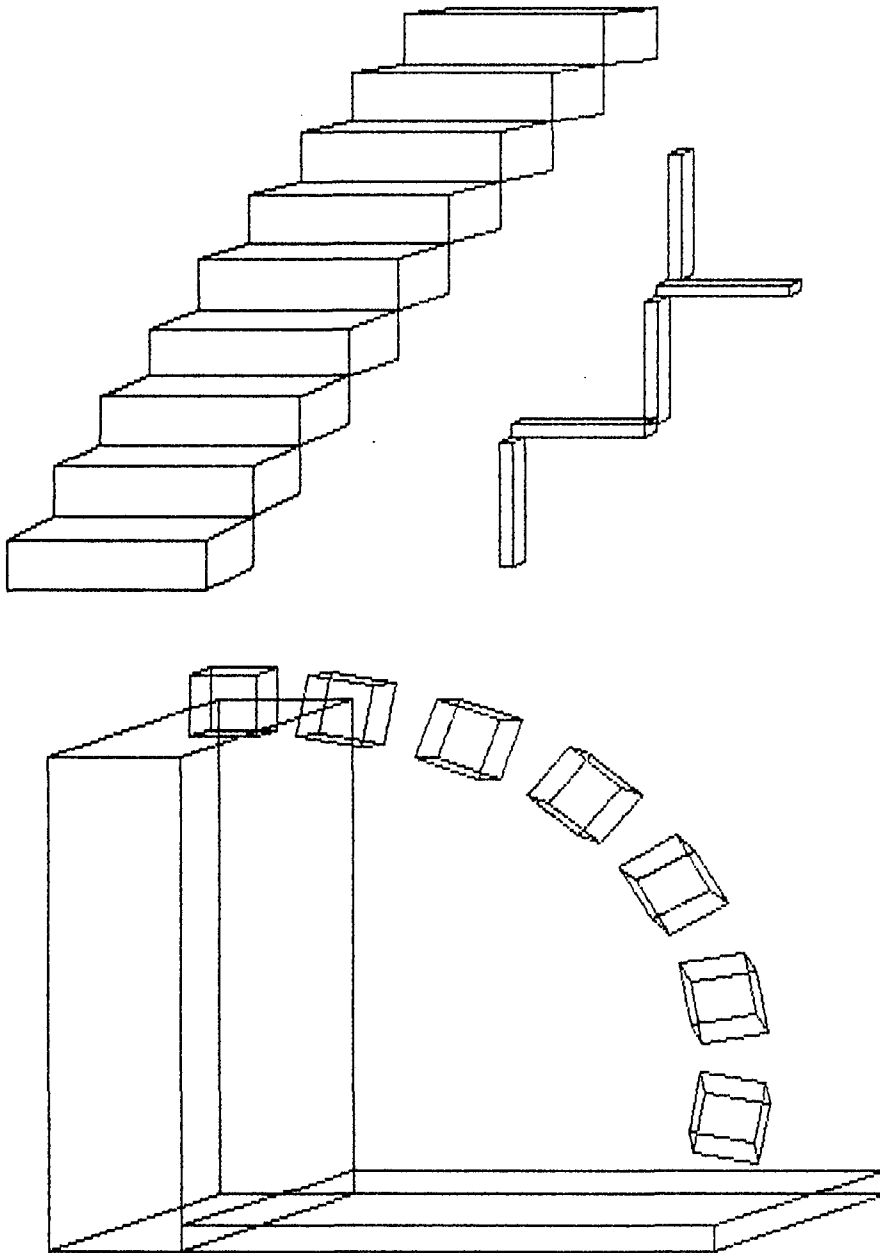


fig. (II.56.)

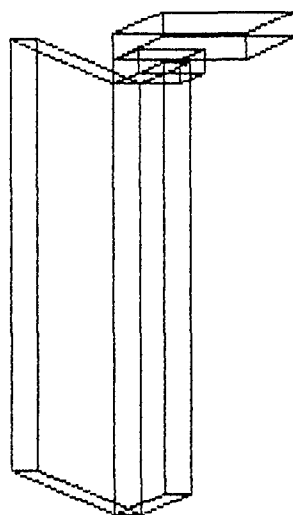
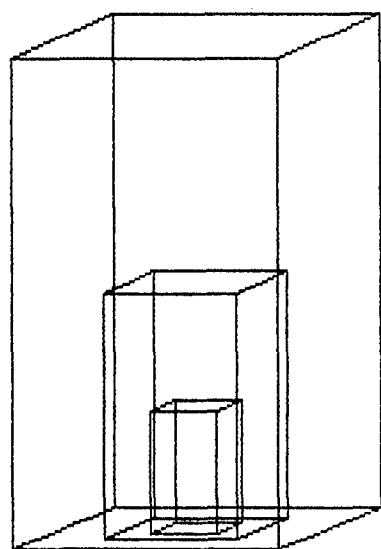
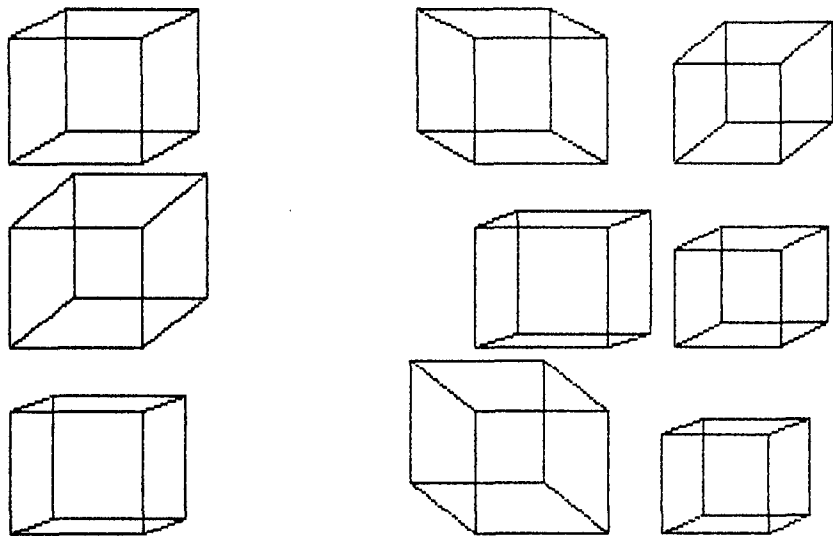


fig (II.57.)

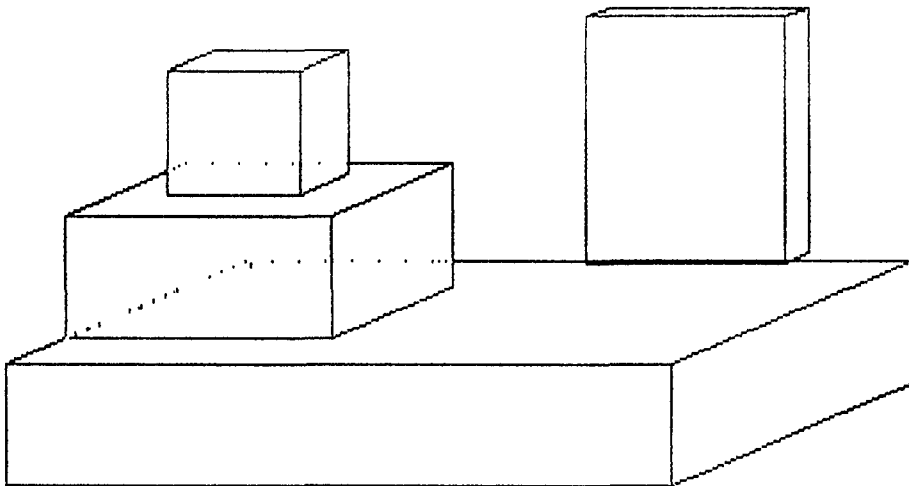
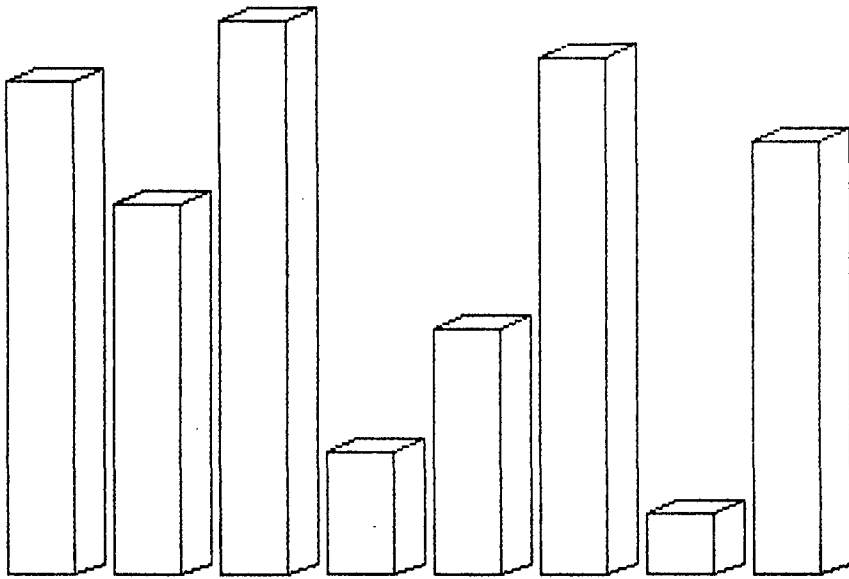


fig. (II.58.)

II.3.1.3.- Cálculo del prisma

Si se elige la opción del cálculo en la figura (II.55.), al escribir cp y pulsar return, se ejecuta el procedimiento 'cp' del módulo "prisma".

El procedimiento 'cp' borra el espacio de trabajo y carga el módulo "cprisma" con sus quince procedimientos.

La siguiente instrucción ejecuta el procedimiento 'cprisma'. El procedimiento 'cprisma' presenta en pantalla la figura (II.59.).

CALCULO DEL PRISMA RECTO

PARA HACER CALCULOS SOBRE UN PRISMA RECTO ESCRIBA cp Y TRES ENTRADAS.
LAS TRES ENTRADAS SON LAS MEDIDAS DE LA BASE LA ALTURA Y EL ANCHO.

AL ESCRIBIR cp (TRES ENTRADAS) Y PULSAR RTN. EL PROGRAMA DIBUJA EL PRISMA OBJETO DEL CALCULO AL MISMO TIEMPO QUE PRESENTA EN PANTALLA LOS VALORES DE (SUPERFICIE LATERAL Y TOTAL, VOLUMEN Y DIAGONALES) .

(PRUEBE EL EJEMPLO ..cp 100 200 100 ..Y PULSE RTN.)

* SI QUIERE VOLVER AL MENU ANTERIOR ESCRIBA pri Y PULSE RTN. *

El procedimiento 'cprisma' presenta la figura (II.59.).

fig. (II.59.)

Desde la figura (II.59) se tiene acceso al módulo anterior ('prisma') y al cálculo del prisma ('cp').

Si se elige la opción del módulo anterior, al escribir prisma y pulsar return se ejecuta el procedimiento 'prisma' del módulo "cprisma". Este procedimiento borra el espacio de trabajo, carga de nuevo el módulo "prisma" y presenta en pantalla la figura (II.59.).

Si se elige la opción del cálculo, al escribir cp con tres entradas y pulsar return, se ejecuta el procedimiento 'cp'.

El procedimiento 'cp' lleva en su declaración, además de las primitivas necesarias, siete procedimientos previamente definidos. Por orden de ejecución los procedimientos son: pit-variabler-ventana-variablen-xyz-medidas-dibujo y pit.

El procedimiento 'variabler' calcula las variables reales, del prisma objeto del cálculo, en función de los datos de entrada.

El procedimiento 'ventana' dibuja los delimitadores gráficos para el posterior dibujo de la figura y representación de los valores de sus parámetros.

El procedimiento 'variablen' calcula las medidas de la figura que se va a representar en pantalla, y las guarda en memoria.

El procedimiento 'xyz' representa, en la parte inferior del apartado reservado para el dibujo, los valores de los datos que se dieron al inicio del procedimiento.

El procedimiento 'medidas' hace un cambio en las medidas de la figura. Este procedimiento hace en la figura el efecto de un zoom, de manera que siempre ocupe el mismo espacio reservado para su representación.

El procedimiento 'dibujo' dibuja la figura objeto del cálculo.

Una vez terminado el procedimiento 'dibujo' se borran de memoria todas las variables que se calcularon durante la ejecución de 'cp', excepto las que se representan en pantalla. Estas son las variables que calcula el procedimiento 'variabler'. Con una reorganización de la memoria y dos bips (pit) se da por terminado el cálculo y dibujo del prisma.

El procedimiento 'cp' tiene el siguiente arbol de procedimientos:

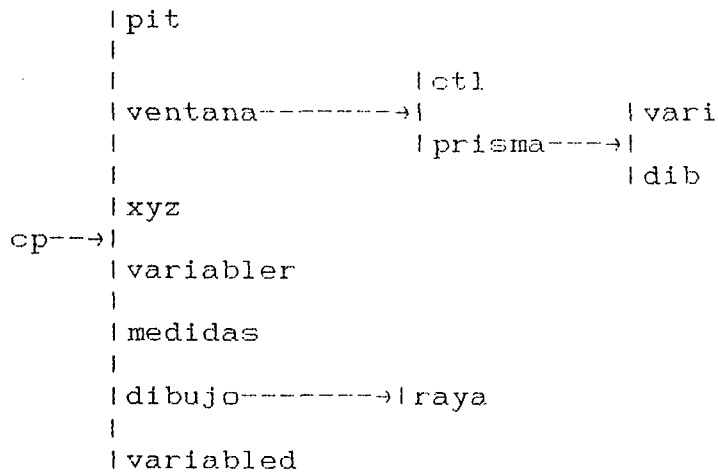
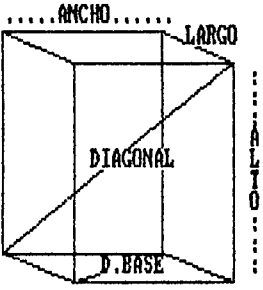
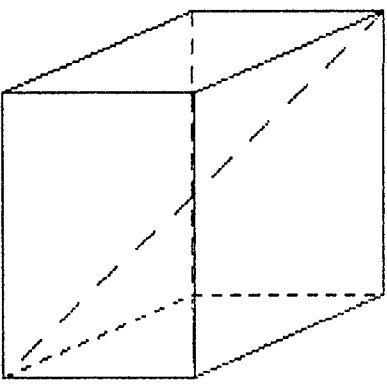


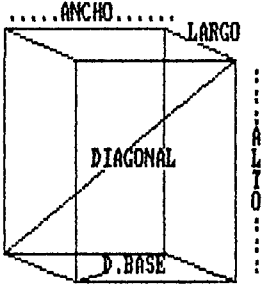
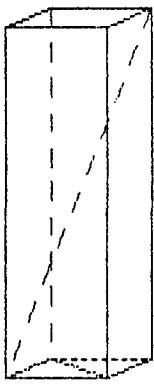
fig. (II.60.)

II.3.1.4.- Ejemplos de aplicación - Cálculo de prismas -

<p>AREA.BASE is 1488 DIAGONAL is 77.0818900860546 VOLUMEN is 58032 AREA.L is 6720 AREA.L is 3744 D.BASE is 66.4844699672922</p>  <p>(PRISMA GENERICO)</p>	 <p>PRISMA OBJETO DEL CALCULO</p> <p>ANCHO = 24 ALTO = 39 LARGO = 62</p>
<p>* PARA VOLVER AL MENU ANTERIOR ESCRIBA cprisma Y PULSE RTN. *</p>	

Ejemplo con.....
 cp 24 39 62.

fig.(II.61.)

<p>AREA.BASE is 3484000 DIAGONAL is 7059.28035917315 VOLUMEN is 22792328000 AREA.L is 52552656 AREA.L is 45584656 D.BASE is 2652.29670073679</p>  <p>(PRISMA GENERICO)</p>	 <p>PRISMA OBJETO DEL CALCULO</p> <p>ANCHO = 1742 ALTO = 6542 LARGO = 2000</p>
<p>* PARA VOLVER AL MENU ANTERIOR ESCRIBA cprisma Y PULSE RTN. *</p>	

Ejemplo con.....
 cp 1742 6542 2000

fig.(II.62.)

AREA.BASE is 875000
 DIAGONAL is 1443.9884471699
 VOLUMEN is 157500000
 AREA.T is 2650000
 AREA.L is 900000
 D.BASE is 1432.69565847331

.....ANCHO..... LARGO
 ALTO
 DIAGONAL
 D.BASE
 (PRISMA GENERICO)

PRISMA OBJETO DEL CALCULO ANCHO = 1250
 ALTO = 180
 LARGO = 700

Ejemplo con.....
 cp 1250 180 700

* PARA VOLVER AL MENU ANTERIOR ESCRIBA cprisma Y PULSE RTN. *

fig. (II.63.)

AREA.BASE is 4
 DIAGONAL is 10.3925622325449
 VOLUMEN is 40
 AREA.T is 88
 AREA.L is 80
 D.BASE is 2.82842717315266

.....ANCHO..... LARGO
 ALTO
 DIAGONAL
 D.BASE
 (PRISMA GENERICO)

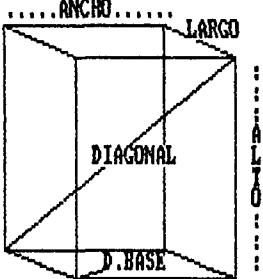
PRISMA OBJETO DEL CALCULO ANCHO = 2
 ALTO = 10
 LARGO = 2

Ejemplo con.....
 cp 2 10 2.

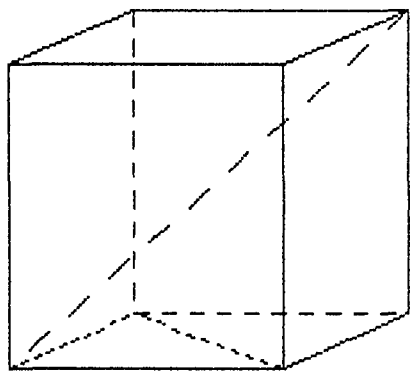
* PARA VOLVER AL MENU ANTERIOR ESCRIBA cprisma Y PULSE RTN. *

fig. (II.64.)

AREA BASE is 7.5e-02
 DIAGONAL is 0.492468962651
 VOLUMEN is 2.25e-02
 AREA.T is 0.45
 AREA.L is 0.3
 D.BASE is 0.390521777671453



(PRISMA GENERICO)



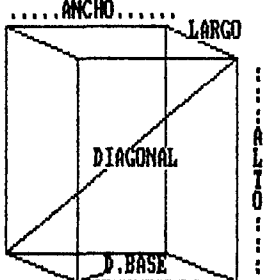
PRISMA OBJETO DEL CALCULO
 ANCHO = 0.25
 ALTO = 0.3
 LARGO = 0.3

Ejemplo con.....
cp 0.25 0.30 0.30

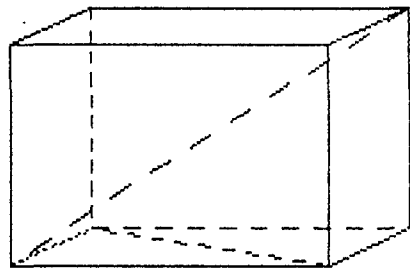
* PARA VOLVER AL MENU ANTERIOR ESCRIBA cprisma Y PULSE RTN. *

fig. (II.65.)

AREA BASE is 36000000
 DIAGONAL is 17000.4876763752
 VOLUMEN is 864000000000
 AREA.T is 624000000
 AREA.L is 432000000
 D.BASE is 14422.6759990385



(PRISMA GENERICO)



PRISMA OBJETO DEL CALCULO
 ANCHO = 12000
 ALTO = 9000
 LARGO = 8000

Ejemplo con.....
cp 12000 9000 8000

* PARA VOLVER AL MENU ANTERIOR ESCRIBA cprisma Y PULSE RTN. *

fig. (II.66.)

II.3.1.5.- Listado de procedimientos

Módulo "espacio"

```
to espacio
  ventana setcursor [21 4]
  pr [DIBUJO Y CALCULO DE FIGURAS TRIDIMENSIONALES] setcursor [19 9]
  pr [CON ESTE MENU PUEDE ACCEDER AL CALCULO Y DIBUJO] setcursor [19 11]
  pr [DE LAS FIGURAS QUE SE RELACIONAN A CONTINUACION] setcursor [6 14]
  pr [1) PRISMA.....] setcursor [50 14]
  pr [ESCRIBA prisma Y PULSE RETURN.] setcursor [6 16]
  pr [2) PIRAMIDE.....] setcursor [50 16]
  pr [ESCRIBA piramide Y PULSE RETURN.] setcursor [ 6 18]
  pr [3) CUBO.....] setcursor [50 18]
  pr [ESCRIBA cubo Y PULSE RETURN.] setcursor [6 20]
  pr [3) OCTAEDRO.....] setcursor [50 20]
  pr [ESCRIBA octaedro Y PULSE RETURN.] setcursor [11 27]
  pr [*SI QUIERE VOLVER AL MENU ANTERIOR ESCRIBA bd Y PULSE RETURN*]
  setcursor [11 28] pit 2
end
```

```
to ventana
  setsplit 2 ht ts cs pu home pit 1 setpos [-210 165]
  pd ctl 385 45 10 pu home setpos [-205 170] pd
  ctl 375 35 0 pu home setpos [-340 -160] pd
  ctl 650 300 10 pu home setpos [-335 -155] pd
  ctl 640 290 0 pu home setpos [-340 -225] pd ctl 650 45 10
end
```

```
to prisma
  erall ct fs load "prisma prisma"
end
```

```
to piramide
  erall ct fs load "piram piram"
end
```

```
to cubo
  erall ct fs load "cubo cubo"
end
```

```
to octaedro
    erall ct fs load "oct oct"
end
```

```
to bd
    ct fs load "bdatos bd"
end
```

```
to pit :n
    repeat :n[type char 7]
end
```

```
to ctl :x :y :z
    repeat 2 [fd :y rt 90 fd :x rt 90] fd :y rt 65 fd :z rt 25
    fd :x rt 90 fd :y rt 65 fd :z rt 115 fd :y rt 65 fd :z
end
```

Módulo "prisma"

.....

```
to prisma
    recuadro setcursor [10 8]
    pr[PARA DIBUJAR CUALQUIER PRISMA VISTO DESDE CUALQUIER ANGULO TIENE QUE]
    setcursor [10 10]
    pr[ESCRIBIR pri SEGUIDO DE CUATRO ENTRADAS, LAS CUATRO ENTRADAS SON LAS]
    setcursor [19 12]
    pr[MEDIDAS DE LOS TRES LADOS Y DEL ANGULO DE VISION.] setcursor [8 16]
    pr[EJEMPLO: ESCRIBA pri 50 200 40 20 Y PULSE RTN. EL PROGRAMA DIBUJA UN]
    setcursor [10 18]
    pr[PRISMA DE 50 DE BASE 200 DE ALTURA 40 DE ANCHO Y VISTO DESDE UN ANGULO
    DE 20 9] setcursor [17 21]
    pr[PRUEBE EL EJEMPLO cs..pri 50 50 40 20..Y PULSE RTN.]) setcursor [16 26]
    pr[*PARA EL CALCULO DEL PRISMA ESCRIBA cp Y PULSE RTN.*] setcursor [15 28]
    pr[*PARA VOLVER AL MENU ANTERIOR ESCRIBA sp Y PULSE RTN.*]
    setcursor [0 30] pit 2
end
```

```
to sp
    erall ct fs load "espacio espacio"
end
```

to recuadro

```
ct ts pit 1 setsplit 1 ht pu home setpos [-155 180] pd
ctl 295 53 10 pu home setpos [-150 185] pd ctl 285 43 0
pu home setpos [-330 -157] pd ctl 660 320 10 pu home
setpos [-330 -240] pd ctl 660 70 10 setcursor [28 3]
pr[DIBUJO Y CALCULO DEL PRISMA RECTO]
```

end

to ctl :x :y :z

```
repeat 2[fd :y rt 90 fd :x rt 90] fd :y rt 65 fd :z rt 25
fd :x rt 90 fd :y rt 65 fd :z rt 115 fd :y rt 65 fd :z
```

end

to pris :x :y :z :ag

```
make "z :z*sin :ag fd :y repeat 2[fd :y rt 90 fd :x rt 90]
fd :y rt 90-:ag fd :z rt :ag fd :x rt 90 fd :y rt 90-:ag
fd :z rt 90+:ag fd :y rt 90-:ag fd :z rt 90+:ag fd :y
rt 90 fd :x rt 90 fd :y bk :y lt 90+:ag fd :z rt 90+:ag
```

end

to pit :n

```
repeat :n[type char 7]
```

end

to cp

```
erall ct fs load "cprisma cprisma
```

end

to pri :x :y :z :ag

```
pris :x :y :z :ag ct setcursor [10 28]
```

```
pr[SI QUIERE EL DIBUJO SIN LEYENDAS AL PIE -ESCRIBA pris EN VEZ DE pri-]
```

```
setcursor [15 29]
```

```
pr[PARA VOLVER AL MENU ANTERIOR ESCRIBA prisma Y PULSE RTN.]
```

end

Módulo "cprisma"

```
to cprisma
  ts ct pit 1 setsplit 3 ht pu home setpos [-120 184] pd ctl 220 45 10 20
  pu home setpos [-115 189] pd ctl 210 35 0 0 pu home setpos [-305 -130]
  pd ctl 590 295 10 20 pu home setpos [-305 -190] pd ctl 590 40 10 20
  setcursor [32 3 ] pr[*CALCULO DEL PRISMA RECTO] setcursor [10 8]
  pr[PARA HACER CALCULOS SOBRE UN PRISMA RECTO ESCRIBA cp TRES ENTRADAS.]
  setcursor [11 10]
  pr [LAS TRES ENTRADAS SON LAS MEDIDAS DE LA BASE LA ALTURA Y EL ANCHO.]
  setcursor [10 13]
  pr[AL ESCRIBIR cp (TRES ENTRADAS) Y PULSAR RTN. EL PROGRAMA DIBUJA EL]
  setcursor [10 15]
  pr[PRISMA OBJETO DEL CALCULO AL MISMO TIEMPO QUE PRESENTA EN PANTALLA]
  setcursor [10 17]
  pr[LOS VALORES DE (SUPERFICIE LATERAL Y TOTAL, VOLUMEN Y DIAGONALES).]
  setcursor [17 20]
  pr[ (PRUEBE EL EJEMPLO..cp 100 200 100..YPULSE RTN.)] setcursor [13 25]
  pr[*SI QUIERE VOLVER AL MENU ANTERIOR ESCRIBA pri Y PULSE RTN.*]
  pit 2 setcursor [0 30]
end
```

```
to ventana
  cs ht setsplit 1 pu setpos [-360 -180] pd ctl 242 435 10 30
  pu home setpos [-95 -180] pd ctl 440 435 10 30] pu home
  setpos [-90 -175] pd ctl 430 425 0 0 pu home setpos [-360 -245]
  pd ctl 705 50 10 30 pu home setpos [-355 -240] pd ctl 695 40 0 0
  pu home setpos [-330 -110] pd prisma 120 180 60 -25
  setcursor [4 10] pr[.....ANCHO.....] setcursor [21 11]
  pr [LARGO] setcursor [27 13] pr[:] setcursor [27 14] pr[:]
  setcursor [27 15] pr[:] setcursor [27 16] pr[A]
  setcursor [27 17] pr[L] setcursor [27 18] pr[T]
  setcursor [27 19] pr [0] setcursor [27 20] pr [:]
  setcursor [27 21] pr [:] setcursor [27 22] pr [:]
  setcursor [12 17] pr [DIAGONAL] setcursor [13 22]
  pr [D.BASE] setcursor [6 24] pr [PRISMA GENERICO]
end
```

```
to pit :n
  repeat :n[type char 7]
end
```

```

to ctl :x :y :z :ag
  repeat 2[fd :y rt 90 fd :x rt 90] fd :y rt 90-:ag fd :z rt :ag
  fd :x rt 90 fd :y rt 90-:ag fd :z rt 90+:ag fd :y rt 90-:ag fd :z
end

```

```

to prisma :x :y :z :ag
  vari dib ern [m D ALTURA altura gamma a tita beta]
end

```

```

to vari
  make "a :z*cos :ag make "altura :z*sin :ag
  make "beta arctan :altura/(:x-:a)
  make "gamma 180-:ag-:beta make "m :altura/sin :beta
  make "tita arctan (:altura+:y)/(:x+:a)
  make "D (:altura+:y)/sin :tita make "ALTURA :y
end

```

```

to dib
  ctl :x :y :z :ag rt 90+:ag fd :y rt 90 fd :x rt 90 fd :y
  bk :y lt 90+:ag fd :z rt 180 fd :z rt 180-:gamma fd :m
  rt :gamma+:ag fd :x rt 180-:tita fd :D
end

```

```

to pri
  erall ct fs load "prisma prisma
end

```

```

to raya :a
  repeat 10 [pu fd :a/20 pd fd :a/20]
end

```

```

to xyz
  setcursor [38 23] pr [PRISMA OBJETO DEL CALCULO]
  setcursor [67 22] pr [ANCHO =] setcursor [75 22] pr :x
  setcursor [67 23] pr [ALTO =] setcursor [75 23] pr :y
  setcursor [67 24] pr [LARGO =] setcursor [75 24] pr :z/sin 25
end

```

```

to cp :x :y :z
  ts pit 1 make "ag 25 variabler make "z :z*sin 25
  ventana variablen xyz medidas dibujo
  ern [m ag al all D.MA D.ME C M dm beta betal tita
  a gamma altura d.p.  $\beta$ ] setcursor [0 1] pons
  recycle pit 2 setcursor [0 30]
end

```

```

to medidas
  make "M 300 make "X (:x+:a) make "Y (:y+:altura)
  make "dm :D.MA if :Y>:X [make "C :M/:Y go "1] make "C :M/:X
label "1 make "x :x*:C make "y :y*:C make "dm :dm*:C make "z :z*:C
  make "m :m*:C make "X :X*:C make "Y :Y*:C
end

```

```

to variabler
  make "al arctan :z/:x
  make "all arctan (:y*cos :al)/:x
  make "D.BASE :x/cos :al
  make "DIAGONAL :D.BASE/cos :all
  make "AREA.BASE :x*:z make "AREA.L (:x+:y)*4
  make "AREA.T :AREA.L+2*:AREA.BASE
  make "ALTURA :y
  make "VOLUMEN :AREA.BASE*:ALTURA
end

```

```

to variablen
  make "a :z*cos :ag make "altura :z*sin :ag
  make "betal arctan :altura/(:x+:a)
  make "beta arctan :altura/(:x-:a)
  make "gamma 180-:ag-:beta
  make "m :altura/sin :beta
  make "tita arctan (:altura+:y)/(:x+:a)
  make "d.p. :altura/(sin (arctan (:altura/(:x+:a))))
  make " $\beta$  arctan :y/:d.p.
  make "D.ME :y/sin : $\beta$ 
  make "D.MA (:altura+:y)/sin :tita
end

```

to dibujo

```
pu home setx 130-:X/2 sety 60-:Y/2 pd ern [X Y]
repeat 2[fd :y rt 90 fd :x rt 90] fd :y rt 65 fd :z rt :ag
fd :x rt 90 fd :y rt 65 fd :z rt 115 fd :y rt 65 fd :z
fd 115 raya :y rt 90 raya :x rt 90 raya :y pu bk :y
pd lt 115 raya :z rt 180 pu fd :z pd rt 180-:gamma
raya :m rt :gamma+:ag fd :x rt 180-:tita raya :dm
ct setcursor [13 28]
pr[*PARA VOLVER AL MENU ANTERIOR ESCRIBA cprisma Y PULSE RTN.*]
```

end

- P I R A M I D E -

II.3.2.- Cálculo y dibujo de la Pirámide

Al seleccionar desde el menú -Geometría del Espacio- la opción número dos (Pirámides), se ejecuta el procedimiento "pirámide".

```
Listado del procedimiento "pirámide":      to pirámide
                                             erall ct fs
                                             load "piram
                                             piram
                                             end
```

El procedimiento 'piramide' borra toda la memoria del espacio de trabajo(erall), borra las líneas de texto de la pantalla mixta(ct), y en la siguiente orden dedica toda la pantalla a los gráficos(fs).

Con esta operación se ha preparado la pantalla para recibir la carga del módulo "piram"(load "piram), después de la carga del módulo "piram" se ejecuta el procedimiento 'piram'.

Al cargarse el módulo "piram" se cargan los procedimientos 'piram', 'sp', 'ctl', 'cpi' y 'pir'. El nombre del procedimiento 'piram' coincide con el nombre del módulo "piram".

II.3.2.1.- Módulo-(Dibujo) y procedimientos

Después de cargar el módulo "piram", la siguiente orden es la de ejecutar el procedimiento 'piram'. Al ejecutarse 'piram' sale en pantalla el siguiente menú:

PROCEDIMIENTO PIRAM

DIBUJO Y CALCULO DE PIRAMIDES

CON ESTE MODULO PODRA DIBUJAR CUALQUIER PIRAMIDE VISTA DESDE CUALQUIER ANGULO.

EJEMPLO: SI ESCRIBE pir 150 200 20

EL PROGRAMA DIBUJARA UNA PIRAMIDE CUADRANGULAR DE 150 DE BASE Y 200 DE ALTURA VISTA DESDE UN ANGULO DE 20 ° POR LA DERECHA DE LA PANTALLA, SI PUNE -20 ° LA VERA POR LA IZQUIERDA, ASIMISMO LA ALTURA DISMINUIRA DE FORMA PROPORCIONAL AL AUMENTO DEL ANGULO DE VISION.

PRUEBE EL EJEMPLO: es pir 150 200 20

(es DELANTE DE pir BORRA LA PANTALLA ANTES DE EMPEZAR A DIBUJAR)

* PARA VOLVER AL MENU ANTERIOR ESCRIBA sp Y PULSE RETURN *

* PARA EL CALCULO ESCRIBA cpi Y PULSE RETURN *

fig. (II.67.)

DIAGRAMA MODULAR

"Módulo principal"

↑

↓

'pir' ↔ ("piram") ↔ 'sp'

↑

↓

"cpi"

fig. (II.68.)

Desde este menú se puede volver al menú anterior, dibujar pirámides o trasladarse al menú de cálculo de pirámides.

Con el procedimiento 'pir' se puede dibujar cualquier pirámide vista desde cualquier ángulo.

Con el procedimiento 'sp' se borra todo lo que tiene el espacio de trabajo y se vuelve al menú anterior.

Con el procedimiento 'cpi' se borra todo lo que hay en el espacio de trabajo y se carga el módulo "cpiram".

II.3.2.2.-Ejemplos de aplicación - Dibujo de pirámides -

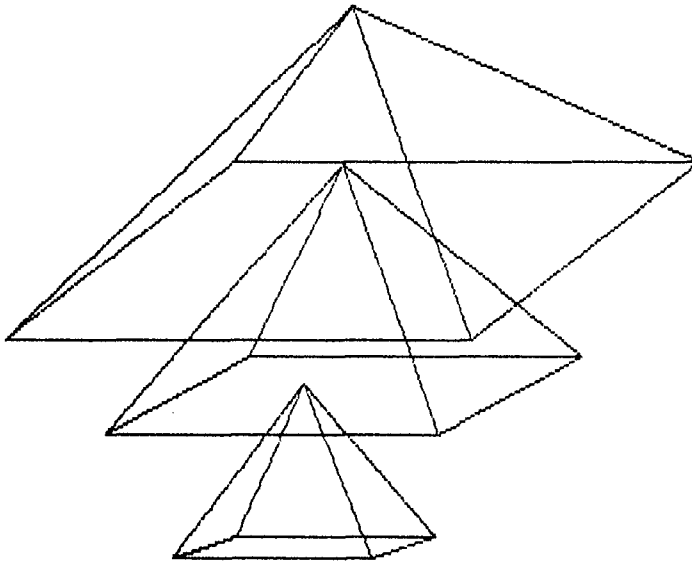


fig. (II.69.)

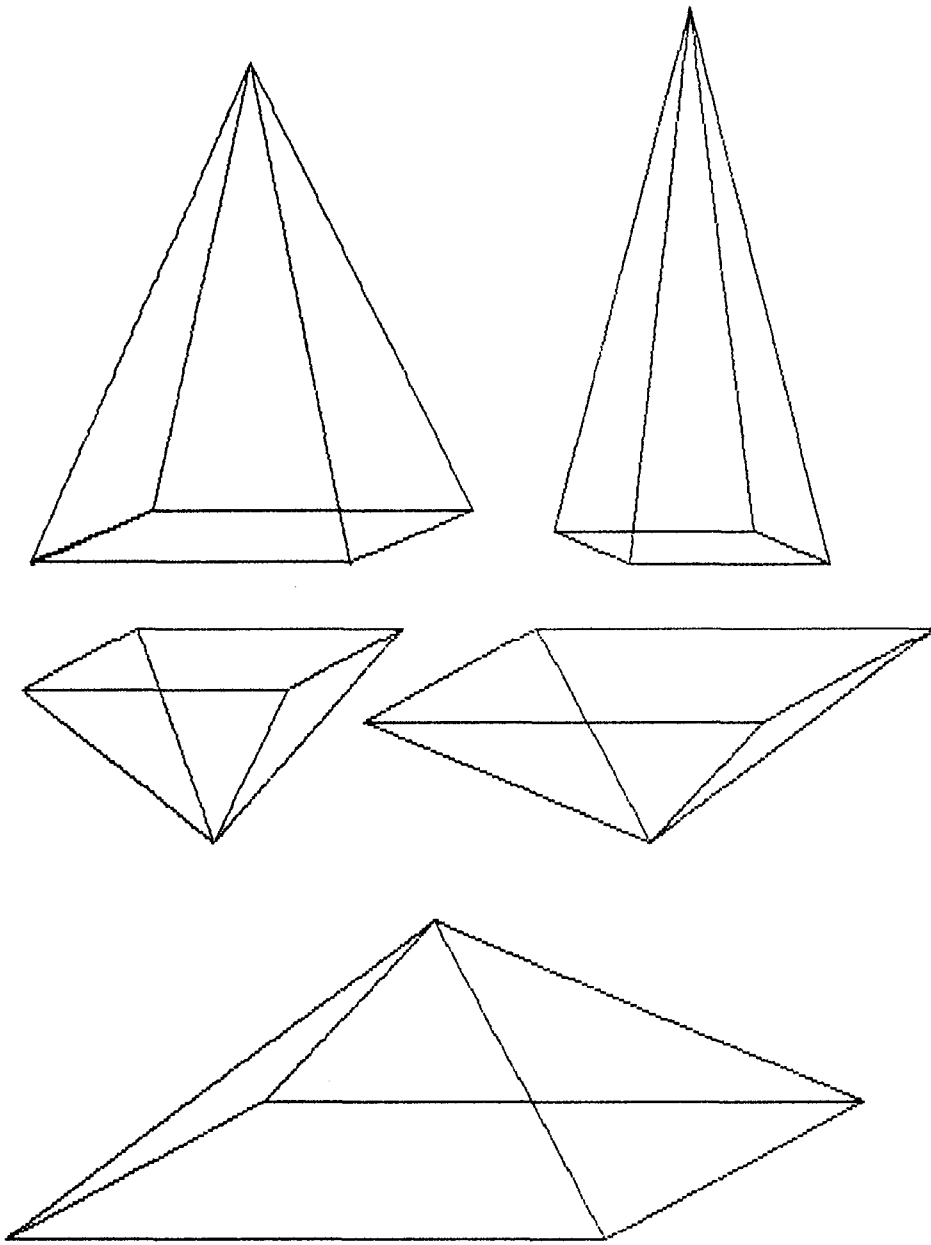


fig. (II.70.)

II.3.2.3.- Listado de procedimientos

Módulo "piram"

```
to sp
  erall ct fs
  load "espacio espacio
end
```

```
to ctl :x :y :z
  repeat 2[fd :y rt 90 fd :x rt 90] fd :y
  rt 65 fd :z rt 25 fd :x rt 90 fd :y rt 65
  fd :z rt 115 fd :y rt 65 fd :z rt 115
end
```

```
to pir :x :y :ag
  setsplit 2
  make "y :y*cos :ag make "z :x*sin :ag make "b :z*sin :ag
  make "a :z*cos :ag make "f (2*:y)/(x+:a) make "e arctan :f
  make "d :y/sin :e rt 90-:ag fd :z rt :ag fd :x
  rt 180-:ag fd :z rt :ag fd :x rt 180-:e fd :d make "f :y-:b
  make "g arctan (2*:f)/(x+:a) make "h (x+:a)/(2*cos :g)
  rt :e+:g fd :h make "j arctan ((x-:a)/(2*:y)) make "k :y/cos :j
  pu bk :h pd rt 90-:g-:j fd :k make "m :y-:b
  make "n arctan (x-:a)/(2*:m) make "o (x-:a)/(2*sin :n) pu bk :k
  pd rt :j+:n fd :o make "q 90-:n-:ag rt :q fd :z rt 90+:ag
  recycle ct setcursor [10 29]
  pr [SI QUIERE VOLVER AL MENU ANTERIOR ESCRIBA piram Y PULSE RETURN]
end
```

```
to cpi
  erall ct fs
  load "cpiram cp
end
```

```

to piram
  ct setsplit 2 ht pu home setpos [-150 200] pd ctl 300 40 0
  pu home setpos [-155 195] pd ctl 310 50 10
  pu home setpos [-300 -100] pd ctl 610 275 10
  pu home setpos [-300 210] pd ctl 610 80 10 setcursor [31 2]
  pr[DIBUJO Y CALCULO DE PIRAMIDES] setcursor [25 6]
  pr[CON ESTE MODULO PODRA DIBUJAR CUALQUIER] setcursor [25 7]
  pr[PIRAMIDE VISTA DESDE CUALQUIER ANGULO] setcursor [27 9]
  pr[EJEMPLO: SI ESCRIBE pir 150 200 20] setcursor [10 11]
  pr[EL PROGREMA DIBUJARA UNA PIRAMIDE CUADRANGULAR DE 150 DE BASE
    y 200 DE] setcursor [10 12]
  pr[ALTURA VISTA DESDE UN ANGULO DE 20º POR LA DERECHA DE LA PANTALLA,
    SI] setcursor [10 13]
  pr[PONE -20º LA VERA POR LA IZQUIERDA, ASIMISMO LA ALTURA DISMINUYE DE]
  setcursor [10 14]
  pr[FORMA PROPORCIONAL AL AUMENTO DEL ANGULO DE VISION.]
  setcursor [26 17]
  pr[PRUEBE EL EJEMPLO: cs pir 150 200 20] setcursor [13 19]
  pr[(cs DELANTE DE pir BORRA LA PANTALLA ANTES DE EMPEZAR A DIBUJAR)]
  setcursor [22 26]
  pr[*PARA EL CALCULO ESCRIBA cpi Y PULSE RETURN*] setcursor [0 29]
end

```

II.3.2.4.- Módulo -(Cálculo)- y procedimientos

Al ejecutarse el procedimiento 'cpi' desde el módulo "cpiram", se borra el espacio de trabajo y se carga el módulo "cpiram".

```
Listado del procedimiento 'cpi':      to cpi
                                         erall ct fs
                                         load "cpiram cp
                                         end
```

El procedimiento 'cpi' borra el espacio de trabajo(erall), borra las líneas de texto en la pantalla mixta(ct), dedica toda la pantalla a los gráficos(fs), carga el módulo "cpiram"(load "cpiram) y ejecuta el procedimiento 'cp'(cp).

Con el módulo "cpiram" pasan al espacio de trabajo todos los procedimientos que lleva en su declaración.

Junto con el módulo "cpiram" se cargan los siguientes procedimientos: 'variabler', 'medidas', 'cpir', 'pir', 'dibujo', 'cp', 'variabled', 'dpir', 'ctl', 'ventana', 'raya' y 'pit'.

De estos procedimientos sólo tres son ejecutables por el usuario del programa. El resto de los procedimientos de este módulo no son reconocidos por el usuario.

Los procedimientos ejecutables son 'cp', 'cpir' y 'dpir'. El procedimiento 'cp' muestra en pantalla el menú de la figura (II.xx).

PROCEDIMIENTO CP

*** CALCULO DE PIRAMIDES ***

CON ESTE MODULO PODRA HACER CALCULOS SOBRE CUALQUIER PIRAMIDE CUADRANGULAR

EJEMPLO: SI ESCRIBE cpir 150 200

* EL PROGRAMA HARA LOS CALCULOS PARA OBTENER EL VOLUMEN LA SUPERFICIE Y TODAS LAS VARIABLES DE UNA PIRAMIDE DE 150 DE BASE Y 200 DE ALTURA.

* AL MISMO TIEMPO , EL PROGRAMA DIBUJARA LA PIRAMIDE Y MOSTRARA EN LA PANTALLA EL RESULTADO DE LOS CALCULOS REALIZADOS.

PRUEBE EL EJEMPLO: cpir 150 200

*** PARA VOLVER AL MENU ANTERIOR ESCRIBA dpir Y PULSE RETURN ***

fig. (II.71.)

DIAGRAMA MODULAR

(Módulo pral.)

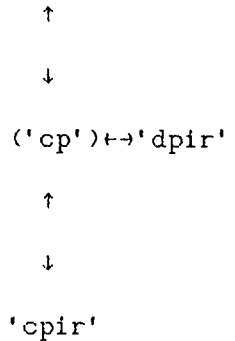


fig. (II.72.)

Desde este menú se puede pasar al menú anterior(dpir) o pasar al cálculo de pirámides(cpir). Si se ejecuta 'dpir' se borra el espacio de trabajo y se carga el módulo "cpiram". Si por el contrario se ejecuta 'cpir', se pone en marcha el procedimiento de cálculo de pirámides.

Al escribir cpir con dos entradas(variables locales), que son la medida de la base y altura de la pirámide, se ejecuta el procedimiento 'cpir'.

En la elaboración de este procedimiento se han utilizado, además de las primitivas necesarias, seis procedimientos previamente definidos.

Los procedimientos utilizados en 'cpir' son: 'ventana', 'pit', 'variabled', 'medidas', 'dibujo' y 'variabler', en este mismo orden.

El procedimiento 'ventana' divide la pantalla en tres apartados por medio de delimitadores gráficos. Cada apartado se dedica a un resultado del procedimiento 'cpir'.

El apartado izquierdo es para la indicación de las variables de la pirámide y para el dibujo de una pirámide genérica que sirve de guía al usuario del programa, el apartado derecho es para el dibujo de la pirámide objeto del cálculo y el apartado inferior es para informar al usuario de cómo trasladarse a otros módulos del programa.

En la elaboración del procedimiento 'ventana' también se han utilizado además de las primitivas necesarios procedimientos previamente definidos. Estos procedimientos son 'ctl' y 'pir'. El procedimiento 'ctl' dibuja los delimitadores gráficos en la pantalla y el procedimiento 'pir' dibuja la pirámide genérica en el apartado izquierdo de la pantalla.

En la elaboración de 'pir' también se ha utilizado un procedimiento previamente definido, esta vez se utiliza un procedimiento que se volverá a usar en la ejecución del procedimiento 'cpir', se trata del procedimiento 'variabled'. Este procedimiento calcula las medidas de todas las líneas y ángulos que delimitan la pirámide, por lo tanto hay que usarlo de nuevo para el dibujo de la pirámide objeto del cálculo.

Una vez terminado 'ventana' la pantalla está preparada y suena un bip producido por el procedimiento 'pit' que es el siguiente en ejecutarse.

El próximo procedimiento es 'variable', una vez terminado estarán almacenadas en memoria todas las medidas de la pirámide que se va a dibujar.

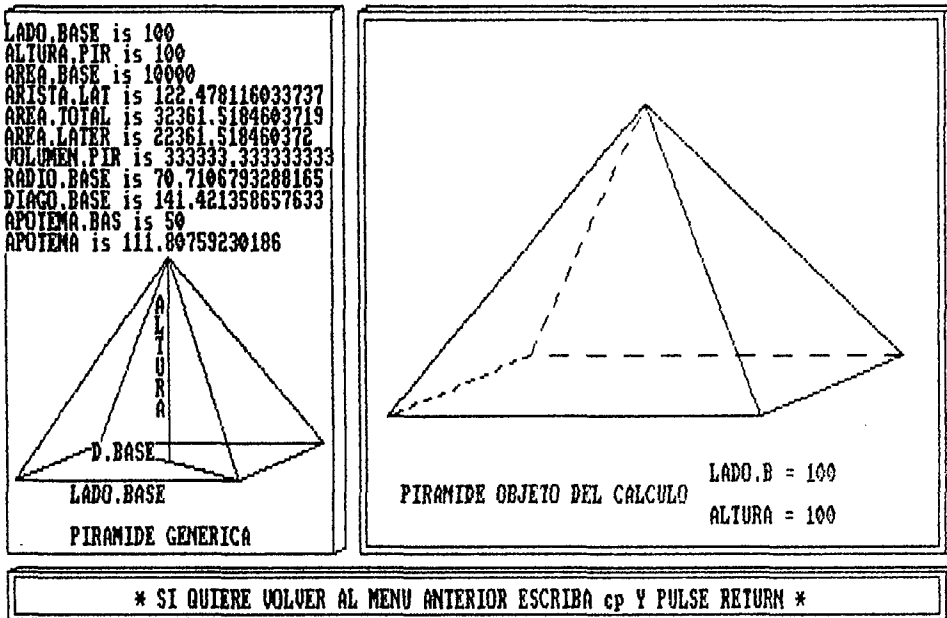
Esta pirámide se tiene que dibujar en el apartado derecho y este apartado al igual que toda la pantalla, tiene unas medidas determinadas. Por lo tanto, hay que preparar estas medidas para que independientemente del tamaño real de la pirámide siempre ocupe el mismo espacio reservado en la pantalla.

El procedimiento encargado de preparar el tamaño de la pirámide para su dibujo es 'medidas'. Este procedimiento reduce o aumenta el tamaño de las medidas en función de una constante que es la medida del espacio reservado para su representación en la pantalla.

Una vez terminado el procedimiento 'medidas' se ejecuta 'dibujo'. Este procedimiento dibuja la pirámide objeto del cálculo en el espacio reservado a tal fin. En la elaboración de este procedimiento se ha usado otro procedimiento previamente definido. Se trata del procedimiento 'raya' que es el encargado de dibujar las partes ocultas de la figura.

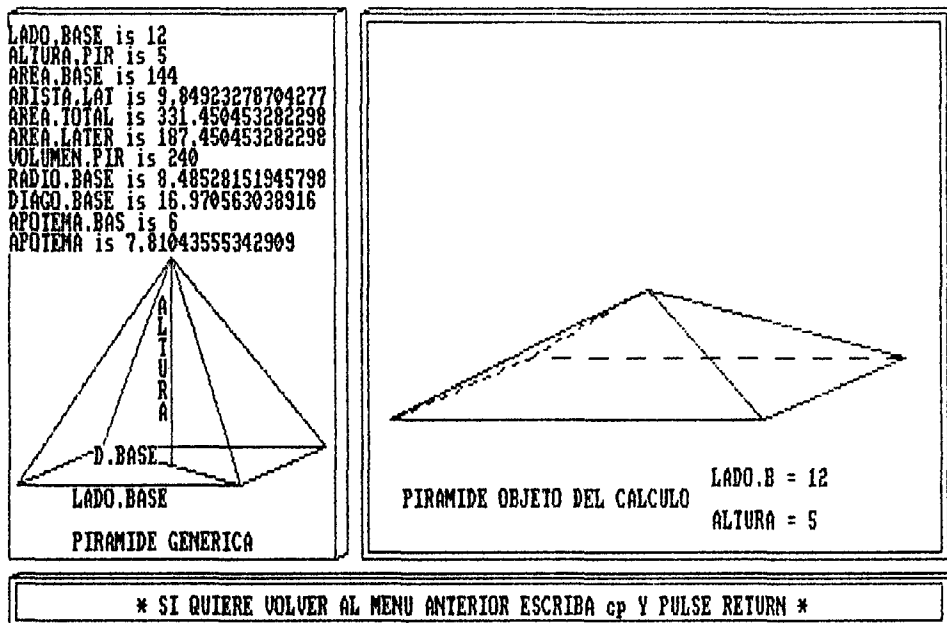
Por último han de calcularse las medidas reales de la pirámide para así poder hallar todas sus variables (volumen, superficie, etc.). De esta labor se encarga el procedimiento 'variable'. Una vez terminado presenta todos los resultados en el apartado izquierdo de la pantalla.

II.3.2.5.-Ejemplos de aplicación - Cálculo de pirámides -



Ejemplo con...
cpir 100 100.

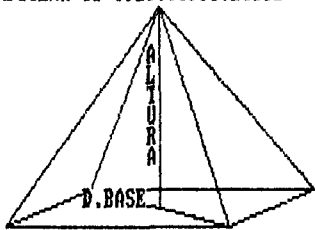
fig. (II.74.)



Ejemplo con...
cpir 12 5.

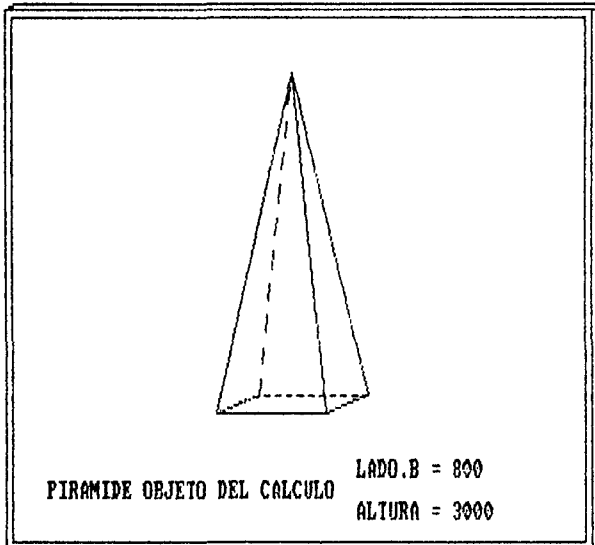
fig. (II.75.)

LADO.BASE is 800
 ALTURA.PIR is 3000
 AREA.BASE is 640000
 ARISTA.LAT is 3052.96905136838
 AREA.TOTAL is 5482656.58742963
 AREA.LATER is 4842656.58742963
 VOLUMEN.PIR is 640000000
 RADIO.BASE is 565.685434630532
 DIAGO.BASE is 1131.37086926106
 APOTEMA.BAS is 400
 APOTEMA is 3026.66036714352



LADO.BASE

PIRAMIDE GENERICA



PIRAMIDE OBJETO DEL CALCULO

LADO.B = 800

ALTURA = 3000

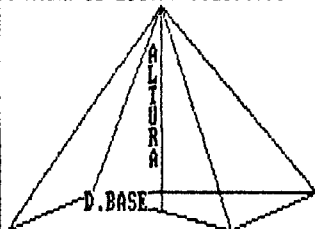
* SI QUIERE VOLVER AL MENU ANTERIOR ESCRIBA ep Y PULSE RETURN *

Ejemplo con....

cpir 800 3000.

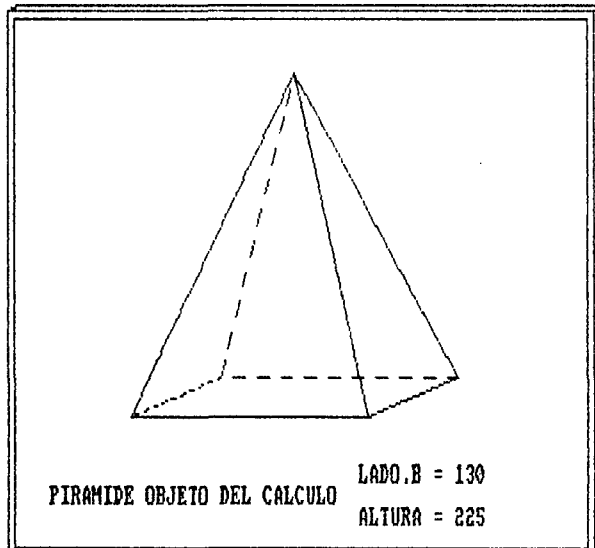
fig. (II.76.)

LADO.BASE is 130
 ALTURA.PIR is 225
 AREA.BASE is 16900
 ARISTA.LAT is 243.059900716566
 AREA.TOTAL is 77793.1317456034
 AREA.LATER is 60893.1317456034
 VOLUMEN.PIR is 1267500
 RADIO.BASE is 91.9238831274614
 DIAGO.BASE is 183.847766254923
 APOTEMA.BAS is 65
 APOTEMA is 234.204352867705



LADO.BASE

PIRAMIDE GENERICA



PIRAMIDE OBJETO DEL CALCULO

LADO.B = 130

ALTURA = 225

* SI QUIERE VOLVER AL MENU ANTERIOR ESCRIBA ep Y PULSE RETURN *

Ejemplo con....

cpir 130 225.

fig. (II.77.)

II.3.2.6.-Listado de procedimientos

Módulo "cpiram"

```
to cp
  ct cs ht setsplit 2 pu home setpos [-125 200] pd ctl 235 40 0
  pu home setpos [-130 195] pd ctl 245 50 10
  pu home setpos [-305 -120] pd ctl 615 295 10
  pu home setpos [-305 -180] pd ctl 615 50 10
  setcursor [32 2]
  pr[*CALCULO DE PIRAMIDES*]
  setcursor [8 7]
  pr[CON ESTE MODULO PODRA HACER CALCULOS SOBRE CUAQUIER PIRAMIDE
  CUADRANGULAR] setcursor [27 10]
  pr[EJEMPLO: SI ESCRIBE cpir 150 200]
  setcursor [10 12]
  pr[*EL PROGRAMA HARA LOS CALCULOS PARA OBTENER EL VOLUMEN LA SUPERFICIE Y]
  setcursor [12 14]
  pr[TODAS LAS VARIABLES DE UNA PIRAMIDE DE 150 DE BASE Y 200 DE ALTURA.]
  setcursor [10 16]
  pr[* AL MISMO TIEMPO, EL PROGRAMA DIBUJARA LA PIRAMIDE Y MOSTRARA EN LA]
  setcursor [12 18]
  pr[PANTALLA EL RESULTADO DE LOS CALCULOS REALIZADOS.]
  setcursor [27 20]
  pr[PRUEBE EL EJEMPLO: cpir 150 200]
  setcursor [ 16 24]
  pr[* PARA VOLVER AL MENU ANTERIOR ESCRIBA dpir Y PULSE RETURN *]
  setcursor [0 29] pit 2
end

to cpir :x :y
  ventana pit 1
  variabled medidas
  dibujo pit 1 variabler
  ern [C c l m D z k j h e n o p q a b c d g]
  setcursor [0 1] pons setcursor [12 28]
  pr[SI QUIERE VOLVER AL MENU ANTERIOR ESCRIBA cp Y PULSE RETURN]
  setcursor [0 30]
  recycle pit 2
end
```

to ventana

```
cs ht setcursor [0 30] setsplit 1
pu home setpos [-360 -185] pd ctl 245 440 10
pu home setpos [-95 -185] pd ctl 440 440 10
pu home setpos [-90 -180] pd ctl 430 430 0
pu home setpos [-360 -240] pd ctl 705 40 10
pu home setpos [-355 -235] pd ctl 695 30 0
pu home setpos [-354 -115] pd pir 168 200
setcursor [14 14] pr[A]
setcursor [14 15] pr[L]
setcursor [14 16] pr[T]
setcursor [14 17] pr[U]
setcursor [14 18] pr[R]
setcursor [14 19] pr[A]
setcursor [8 21] pr[D. BASE]
setcursor [6 23] pr[LADO. BASE]
setcursor [6 25] pr[PIRAMIDE GENERICA]
setcursor [37 23] pr[PIRAMIDE OBJETO DEL CALCULO]
setcursor [66 22] pr[LADO. B =] setcursor [75 22] pr :x
setcursor [66 24] pr[ALTURA =] setcursor [75 24] pr :y
```

end

to ctl :x :y :z

```
repeat 2[fd :y rt 90 fd :x rt 90] fd :y
rt 65 fd :z rt 25 fd :x rt 90 fd :y
rt 65 fd :z rt 115 fd :y rt 65 fd :z
```

end

to variabled

```
make "y :y*cos 25 make "z :z*sin 25
make "b :z*sin 25 make "a :z*cos 25
make "c (:x+:a)/2
make "e arctan :y/:c
make "d :y/sin :e
make "g arctan (:y-:b)/:c
make "h :c/cos :g
make "j arctan (:x-:a)/(2*:y)
make "k :y/cos :j
make "m :y-:b
make "n arctan (:x-:a)/2*:m)
make "o (:x-:a)/(2*sin :n)
make "p 90-:n make "q :p-25
make "D :b/(sin arctan :b/(:x-:a))
```

end

```

to pit :n
  repeat :n[type char 7]
end

```

```

to pir :x :y
  variabled
  rt 65 fd :z rt 25 fd :x rt 155 fd :z
  rt 25+arctan :b/(:x-:a) fd :D bk :D
  lt arctan :b/(:x-:a) fd :x rt 180-:e
  fd :d rt :e+:g fd :h bk :h rt 90-:g-:j
  fd :k bk :k rt :j fd :y-(:b/2)
  bk :y-(:b/2) rt :n fd :o rt :q
end

```

```

to medidas
  if :x>:y [make "C 280/:x go "10] make "C 280/:y
label "10 make "z :z*:C make "x :x*:C make "d :d*:c
  make "h :h*:C make "k :k*:C make "o :o*:C
end

```

```

to dibujo
  make "c1 120-(:x+:z*cos 25)/2 pu home
  setx :c1 sety -75 pd rt 65 raya :z rt 25
  raya :x rt 155 fd :z rt 25 fd :x rt 180-:e
  fd :d rt :e+:g fd :h pu bk :h pd rt 90-:g-:j
  fd :k pu bk :k pd rt :j+:n raya :o rt :q
end

```

```

to raya :a
  repeat 10[pu fd :a/20 pd fd :a/20]
end

```

```

to dpir
  ct fs erall
  load "piram piram
end

```

```

to variabler
  make "x :x/:C
  make "y :y/cos 25
  make "DIAG.BAS :x/ cos 45
  make "ARISTA.L :y/sin arctan ((2*:y)/:DIAG.BAS)
  make "ALTURA.P :y
  make "AREA.BAS :x*:x
  make "ALTURA.T :y/sin arctan ((2*:y)/:x)
  make "AREA.LAT 2*:ALTURA.T*:x
  make "AREA.TOT :AREA.BAS+:AREA.LAT
  make "LADO.BAS :x
  make "VOLUME.P (:AREA.BAS*:ALTURA.P)/3
end

```

II.3.3.- Cálculo y dibujo del cubo

Al seleccionar el apartado (3) de la figura (II.54.), se escribe cubo y se pulsa return. Con esta operación se ejecuta el procedimiento 'cubo' del módulo "espacio".

Este procedimiento borra el espacio de trabajo y carga el módulo "cubo". Con el módulo "cubo" se cargan los procedimientos 'pit', 'cubo', 'recuadro', 'sp', 'cu', 'cus', 'ctl' y 'cc'.

La siguiente instrucción del procedimiento 'cubo', perteneciente al módulo "espacio", es la de ejecutar el procedimiento 'cubo' (perteneciente al módulo "cubo").

DIBUJO Y CALCULO DEL CUBO

PARA DIBUJAR CUALQUIER CUBO VISTO DESDE CUALQUIER ANGULO TIENE QUE
ESCRIBIR `cu` SEGUIDO DE DOS ENTRADAS. LAS DOS ENTRADAS SON LAS
MEDIDAS DEL LADO Y DEL ANGULO DE VISION.

EJEMPLO: ESCRIBA `cu 100 30` Y PULSE `RTH`. EL PROGRAMA DIBUJA UN CUBO
DE LADO = 100 Y VISTO DESDE UN ANGULO DE 30 °.

(PRUEBE EL EJEMPLO `cs..cu 120 30` .Y PULSE `RTH`.)

* PARA EL CALCULO DEL CUBO ESCRIBA `cc` Y PULSE `RTH`. *

* PARA VOLVER AL MENU ANTERIOR ESCRIBA `sp` Y PULSE `RTH`. *

El procedimiento
'cubo' presenta en
pantalla la figura
(II.78.).

(fig. II.78.)

II.3.3.1.- Módulo - Dibujo - y procedimientos

Desde la figura (II.78.) se tiene acceso al módulo anterior (sp), al cálculo del cubo (cc) y al dibujo del cubo (cu).

Si se elige la opción del módulo anterior, al escribir sp y pulsar return, se ejecuta el procedimiento 'sp'. Este procedimiento borra el espacio de trabajo, carga el módulo "espacio" y ejecuta el procedimiento 'espacio' (este procedimiento presenta en pantalla la figura II.54.).

Si se elige la opción del dibujo, se puede hacer el dibujo de cualquier cubo. El procedimiento que dibuja el cubo tiene dos variables locales. Las dos variables son la medida de la arista y la del ángulo de visión del cubo. Con este procedimiento la diversidad de figuras que se pueden representar son prácticamente todas las que el usuario pueda imaginar.

II.3.3.2.- Ejemplos de aplicación - Dibujo del cubo -

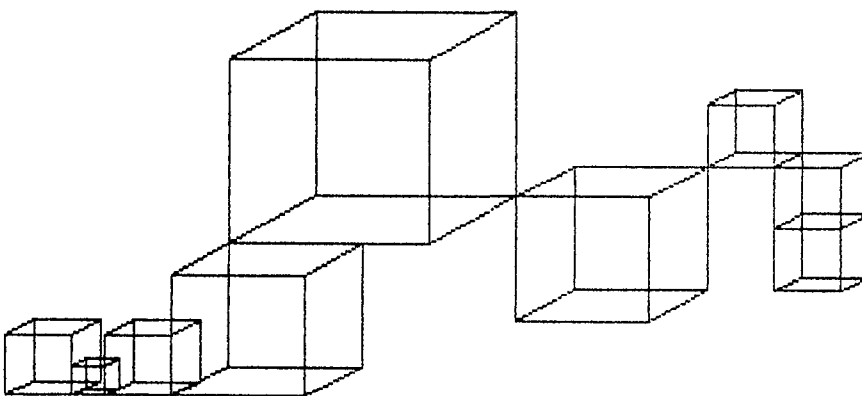


fig. (II.79.)

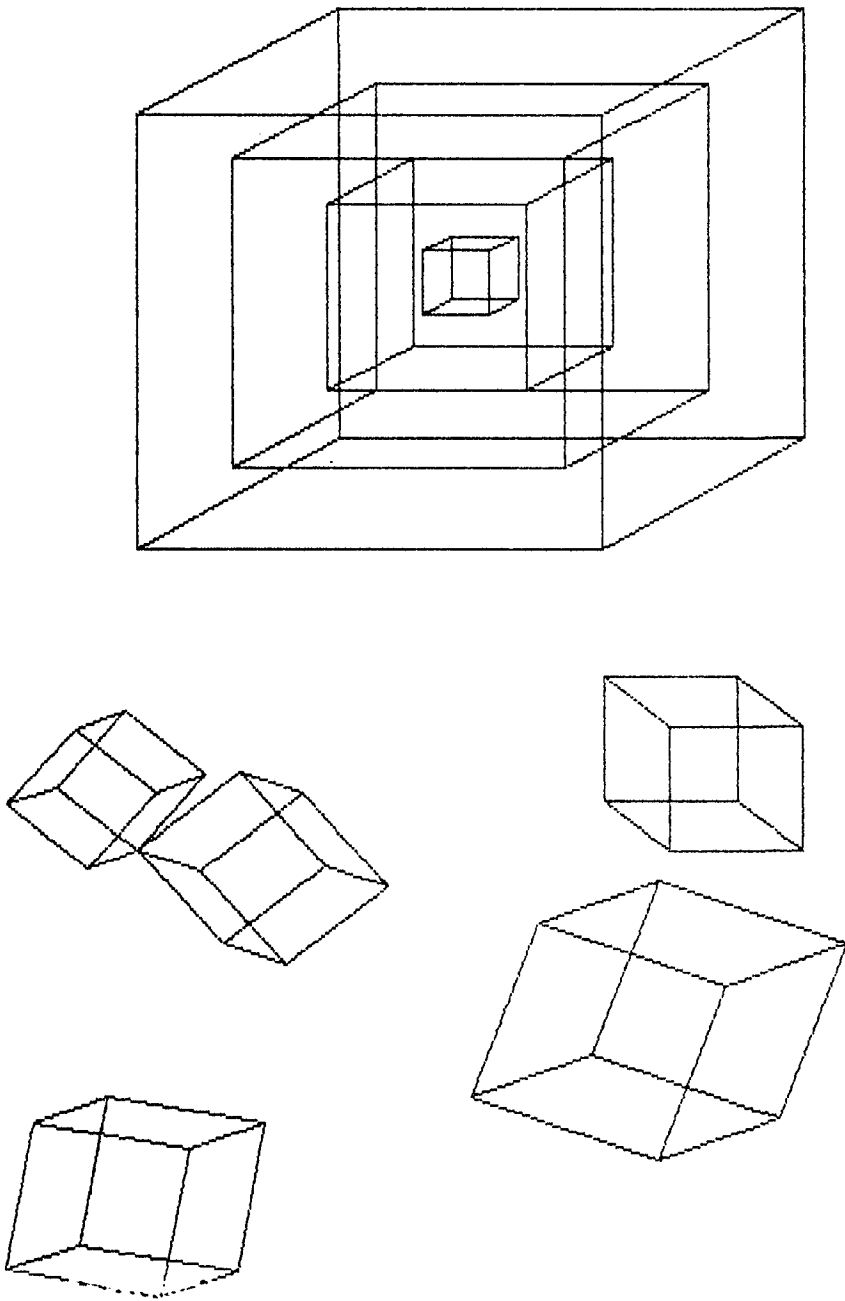


fig. (II.80.)

II.3.3.3.- Módulo - Cálculo - y procedimientos

Si se elige la opción del cálculo del cubo desde la figura (II.78.), al escribir cc y pulsar return, se ejecuta el procedimiento 'cc' del módulo "cubo". Este procedimiento borra el espacio de trabajo y carga el módulo de cálculo del cubo("ccubo"). La próxima instrucción ejecuta el procedimiento 'ccubo'. Este procedimiento presenta en pantalla el menú de la figura (II.81.). Al cargarse en el espacio de trabajo el módulo "ccubo" también se cargan los procedimientos 'ventana', 'vari', 'medidas', 'variabler', 'xyz', 'variabled', 'ctl', 'dib', 'raya', 'dibujo', 'pit', 'cc', 'ccubo', 'cubo' y 'ccv'.

CALCULO DEL CUBO

PARA HACER CALCULOS SOBRE UN CUBO ESCRIBA cc Y UNA ENTRADA.
LA ENTRADA ES LA MEDIDA DE LA ARISTA.

AL ESCRIBIR cp (UNA ENTRADA) Y PULSAR RTN. EL PROGRAMA DIBUJA EL
CUBO OBJETO DEL CALCULO AL MISMO TIEMPO QUE PRESENTA EN PANTALLA
LOS VALORES DE (SUPERFICIE LATERAL Y TOTAL, VOLUMEN Y DIAGONALES)

(PRUEBE EL EJEMPLO .. cc 100 ..Y PULSE RTN.)

* SI QUIERE VOLVER AL MENU ANTERIOR ESCRIBA cubo Y PULSE RTN. *

El procedimiento
'ccubo' presenta
la fig.(II.81.).

fig. (II.81.)

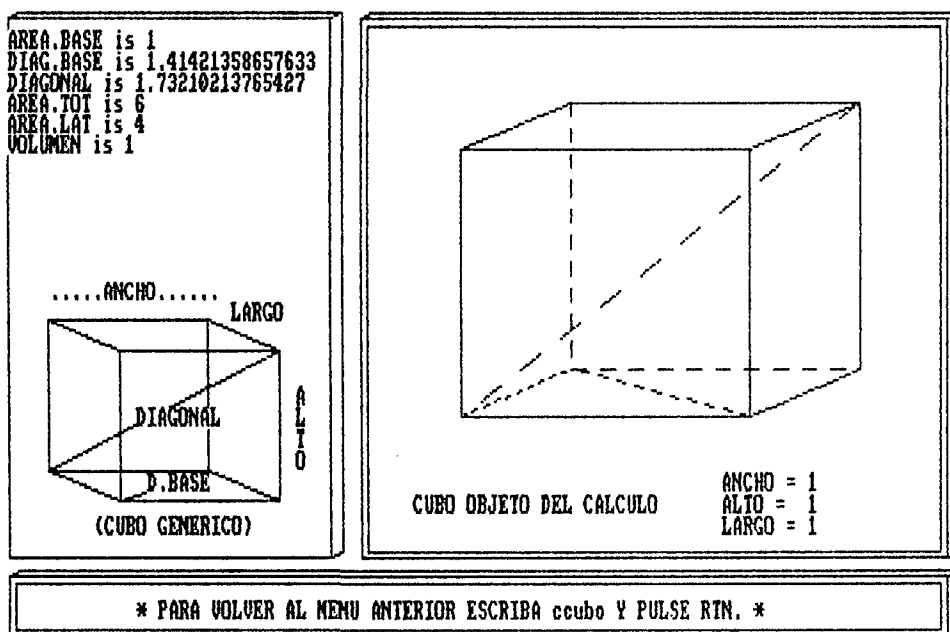
En la elaboración de este módulo se han aprovechado prácticamente todos los procedimientos del módulo "cprisma".

La consideración que se ha hecho es que se trata con un prisma que tiene todas sus aristas iguales; de esta manera la figura tratada es un cubo. Estos procedimientos están pensados para trabajar con las tres variables de las aristas del prisma, pero con una pequeña modificación que se ha introducido al comienzo del procedimiento del cálculo 'cc', se han aprovechado todos los procedimientos.

La modificación consiste en asignar el valor de la arista del cubo a las variables (y) y (z). Posteriormente se hace uso de todos los procedimientos del cálculo del prisma.

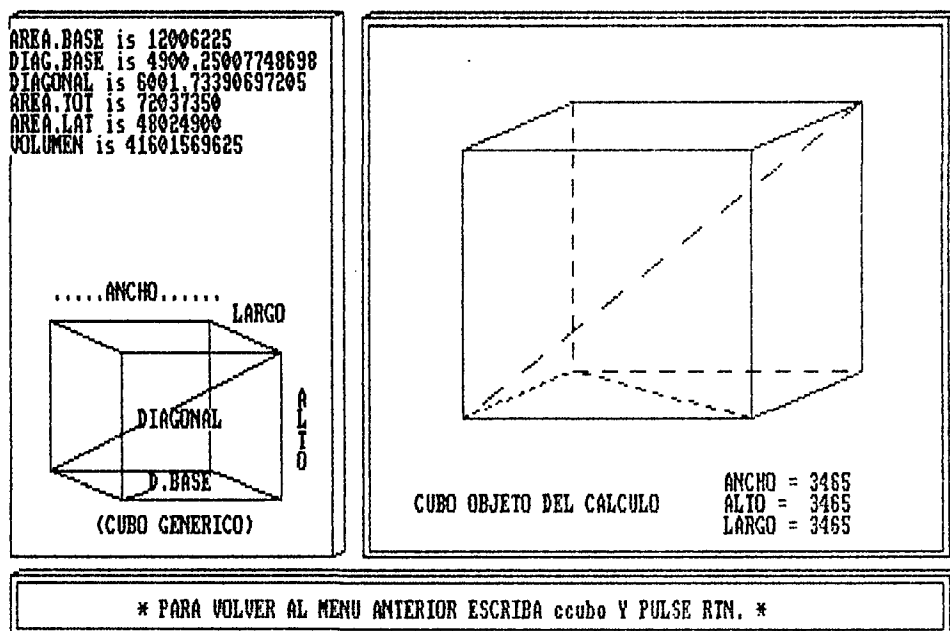
Estos procedimientos serían más simples si se hubiesen hecho pensados para el cálculo del cubo, pero el hecho de que ya estaban confeccionados justifica su uso.

II.3.3.4.- Ejemplos de aplicación - Cálculo del cubo -



Ejemplo con
cc 1.

fig. (II.82.)



Ejemplo con
cc 3465.

fig. (II.83.)

II.3.3.5.- Listado de procedimientos

Módulo "cubo"

```
to sp
  erall ct fs load "espacio espacio"
end
```

```
to ctl :x :y :z
  repeat 2[fd :y rt 90 fd :x rt 90] fd :y
  rt 65 fd :z rt 25 fd :x rt 90 fd :y
  rt 65 fd :z rt 115 fd :y rt 65 fd :z
end
```

```
to pit :n
  repeat :n[type char 7]
end
```

```
to recuadro
  cs ts pit 1 setsplit 1 ht pu home setpos [-140 180] pd
  ctl 250 53 10 pu home setpos [-135 185] pd ctl 240 43 0
  pu home setpos [-310 -157] pd ctl 595 320 10 pu home
  setpos [-310 -240] pd ctl 595 70 10 setcursor [31 3]
  pr [DIBUJO Y CALCULO DEL CUBO]
end
```

```
to cu :x :ag
  cus :x :ag ct setcursor [9 28]
  pr [SI QUIERE EL DIBUJO SIN LEYENDAS AL PIE - ESCRIBA cus EN
  VEZ DE cu -] setcursor [14 29]
  pr [PARA VOLVER AL MENU ANTERIOR ESCRIBA cubo Y PULSE RTN.]
end
```

```
to cc
  erall ct fs load "ccubo ccubo"
end
```

```

to cus :x :ag
    make "z :x*sin :ag
    repeat 4[fd :x rt 90] fd :x rt 90-:ag fd :z rt :ag
    fd :x rt 90 fd :x rt 90-:ag fd :z rt 90+:ag fd :x
    rt 90-:ag fd :z rt 90+:ag repeat 2[fd :x rt 90]
    fd :x bk :x lt 90+:ag fd :z rt 90+:ag
end

```

```

to cubo
    recuadro setcursor [10 8]
    pr [PARA DIBUJAR CUALQUIER CUBO VISTO DESDE CUALQUIER ANGULO TIENE QUE]
    setcursor [12 10]
    pr [ESCRIBIR cu SEGUIDO DE DOS ENTRADAS. LAS DOS ENTRADAS SON LAS]
    setcursor [22 12]
    pr [MEDIDAS DEL LADO Y DEL ANGULO DE VISION.] setcursor [10 16]
    pr [EJEMPLO: ESCRIBA cu 100 30 Y PULSE RTN. EL PROGRAMA DIBUJA UN CUBO]
    setcursor [20 18]
    pr [DE LADO = 100 Y VISTO DESDE UN ANGULO DE 30 °.]
    setcursor [19 21]
    pr [(PRUEBE EL EJEMPLO es...cu 120 30.. Y PULSE RTN.)]
    setcursor [17 26]
    pr [*PARA EL CALCULO DEL CUBO ESCRIBA cc Y PULSE RTN.*]
    setcursor [15 28]
    pr [*PARA VOLVER AL MENU ANTERIOR ESCRIBA sp Y PULSE RTN.*]
    setcursor [0 30] pit 2
end

```

Módulo "ccubo"

Los procedimientos 'variabler', 'medidas', 'xyz', 'variablen', 'ctl', 'pit', 'raya' y 'dibujo' son los mismos que tiene el módulo "cprisma".

```

to vari
    make "a :z*cos :ag make "altura :z*sin :ag
    make "beta arctan :altura/(:x-:a)
    make "gamma 180-:ag-:beta make "m :altura/sin :beta
    make "tita arctan (:altura+:y)/(:x+:a)
    make "D (:altura+:y)/sin :tita make "ALTURA :y
end

```

to ccubo

```
ts ct pit 1 setsplit 3 ht pu home setpos [-110 184] pd
ctl 190 45 0 20 pu home setpos [-105 189] pd ctl 180 35 0 0
pu home setpos [-305 -130] pd ctl 590 295 10 20 pu home
setpos [-305 -190] pd ctl 590 40 10 20 setcursor [35 3]
pr [CALCULO DEL CUBO] setcursor [14 8]
pr [PARA HACER CALCULOS SOBRE UN CUBO ESCRIBA cc Y UNA ENTRADA.]
setcursor [23 10]
pr [LA ENTRADA ES LA MEEDIDA DE LA ARISTA.] setcursor [12 13]
pr [AL ESCRIBIR cp (UNA ENTRADA) Y PULSAR RTN. EL PROGRAMA DIBUJA
EL] setcursor [12 15]
pr [CUBO OBJETO DEL CALCULO AL MISMO TIEMPO QUE PRESENTA EN]
PANTALLA] setcursor [12 17]
pr [LOS VALORES DE (SUPERFICIE LATERAL Y TOTAL, VOLUMEN Y DIAG.)]
setcursor [20 20]
pr [(PRUEBE EL EJEMPLO..cc 100.. Y PULSE RTN.)]
setcursor [13 25]
pr [*SI QUIERE VOLVER AL MENU ANTERIOR ESCRIBA cubo Y PULSE RTN.*]
pit 2 setcursor [0 30]
```

end

to ventana

```
cs ht setsplit 1 pit 1
pu home setpos [-360 -180] pd ctl 242 435 10 30
pu home setpos [-95 -180] pd ctl 440 435 10 30
pu home setpos [-90 -175] pd ctl 430 425 0 0
pu home setpos [-360 -240] pd ctl 705 35 10 30
pu home setpos [-355 -240] pd ctl 695 40 0 0
pu home setpos [-330 -110] pd ccv 120 120 60 -25
setcursor [4 13] pr [.....ANCHO.....] setcursor [21 14]
pr [LARGO] setcursor [27 18] pr [A] setcursor [27 19]
pr [L] setcursor [27 20] pr [T] setcursor [27 21]
pr [O] setcursor [12 19] pr [DIAGONAL] setcursor [13 22]
pr [D.BASE] setcursor [8 24] pr [(CUBO GENERICO)]
```

end

to cubo

```
erall ct fs load "cubo cubo
```

end

```

to cc :x
  ts pit 1 make "ag 25 make "y :x make "z :x variabler
  make "z :x*sin 25
  ventana variabler xyz medidas dibujo
  ernfm z y ag all D.MA D.ME C M  $\beta$  beta betal tita a gamma altura d.p.]
  setcursor [0 1] pons recycle pit 2 setcursor [0 30]
end

```

```

to ccv :x :y :z :ag
  vari dib ern [m D ALTURA altura gamma a tita betal]
end

```

```

to dib
  ctl :x :y :z :ag rt 90+:ag fd :y rt 90 fd :x rt 90 fd :y
  bk :y lt 90+:ag fd :z rt 180 fd :z rt 180-:gamma fd :m
  rt :gamma+:ag fd :x rt 180-:tita fd :D
end



```

II.3.4.- Cálculo y dibujo del octaedro

Al seleccionar el apartado (4) del menú de la figura (II.54.), se escribe octaedro y se pulsa return. Con esta operación se ejecuta el procedimiento 'octaedro' del módulo "espacio".

Este procedimiento borra el espacio de trabajo y carga el módulo "octa". Con el módulo "octa" se cargan los procedimientos 'pir', 'sp', 'ot', 'pit', 'ctl', 'octa', 'ots' y 'oct'.

la siguiente instrucción del procedimiento 'octaedro' es la de ejecutar el procedimiento 'octa'.



DIBUJO Y CALCULO DEL OCTAEDRO

PARA EL DIBUJO DE UN OCTAEDRO ESCRIBA `ot` Y TRES ENTRADAS. LAS TRES ENTRADAS SON LAS MEDIDAS DE LA BASE, ALTURA Y ANGULO DE VISION DE UNA DE LAS PIRAMIDES QUE LO FORMAN. AL SER LA FIGURA SIMETRICA, BASTA CON DAR LAS MEDIDAS DE LA PIRAMIDE PARA EL DIBUJO DEL OCTAEDRO.

EJEMPLO: ESCRIBA `es.. ot 120 100 25` Y PULSE `RIN`.

* PARA EL MENU ANTERIOR ESCRIBA `sp` Y PULSE `RIN`. *

* PARA EL CALCULO ESCRIBA `oct` Y PULSE `RIN`. *

El procedimiento 'octa' presenta en pantalla la figura (II.84.).

(fig. II.84.)

II.3.4.1.- Módulo - Dibujo - y procedimientos

Desde la figura (II.84.) se tiene acceso al módulo anterior (sp), al cálculo del octaedro (oct) y al dibujo del octaedro (ot).

Si se elige la opción del módulo anterior, al escribir sp y pulsar return, se ejecuta el procedimiento 'sp'. Este procedimiento borra el espacio de trabajo, carga el módulo "espacio" y ejecuta el procedimiento 'espacio'.

II.3.4.2.- Ejemplos de aplicación - Dibujo del octaedro -

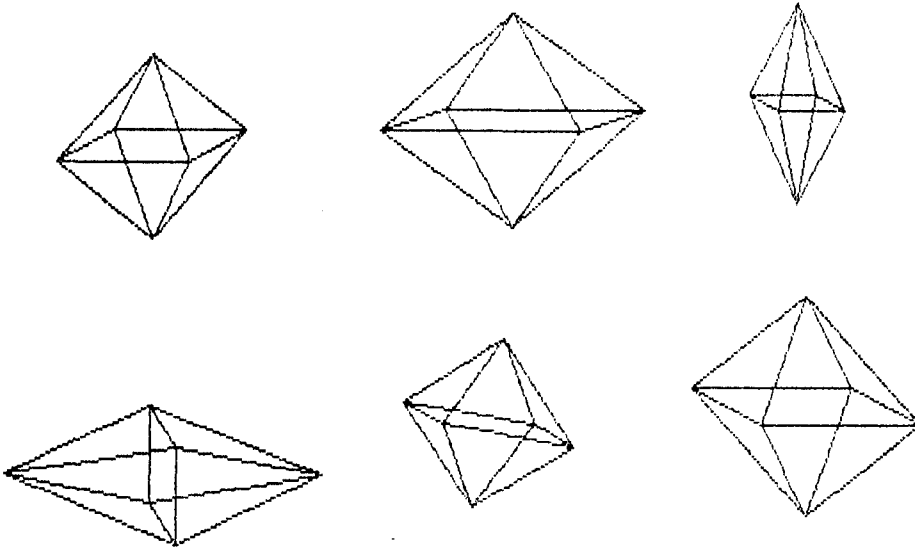


fig. (II.85.)

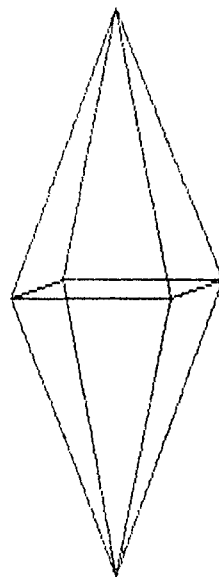
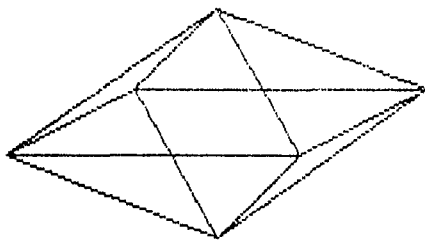
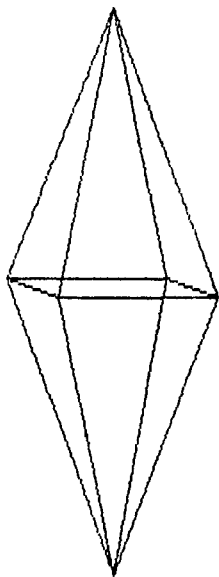
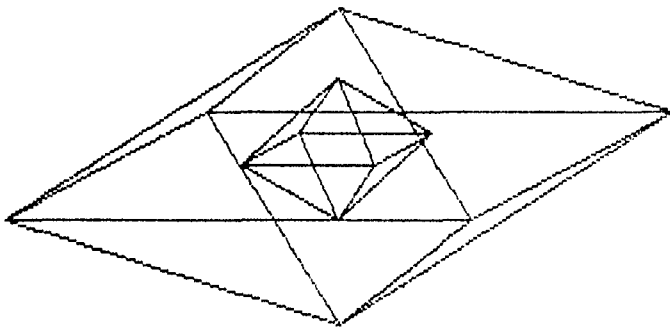


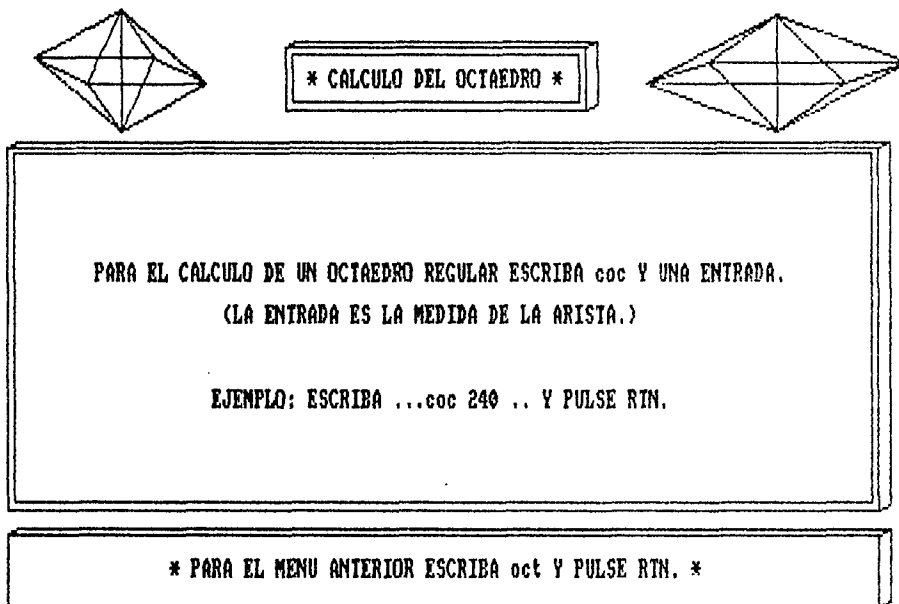
fig. (II.86.)

II.3.4.3.- Módulo - Cálculo - y procedimientos

Al escribir oct y pulsar return desde la figura (II.84.), se ejecuta el procedimiento 'oct' del módulo "octa". Este procedimiento carga el módulo "cocta" y ejecuta el procedimiento 'cocta' del mismo módulo.

Con el módulo "cocta" se cargan los procedimientos 'pir', 'oct', 'pit', 'ctl', 'ventana', 'cocta', 'ot', 'coc', 'variabler', 'medidas', 'variabled', 'raya', 'dibujo1' y 'dibujo2'.

El procedimiento 'cocta' presenta en pantalla el menú de la fig. (II.87.).



El procedimiento 'cocta' presenta el menú de la figura (II.87.).

fig. (II.87.)

Con este procedimiento se pueden hacer cálculos sobre un octaedro regular del que se conoce la arista.

Al escribir coc con una entrada y pulsar return, se ejecuta el procedimiento 'coc'.

Para su ejecución se utilizan los procedimientos previamente definidos: 'ventana', 'variabler', 'variabled', 'medidas', 'dibujo1', 'dibujo2' y 'pit'.

En la ejecución del procedimiento 'coc' se ejecutan los procedimientos antes mencionados en el siguiente orden:

El procedimiento 'ventana', como en otros módulos, hace una división de la pantalla gráfica.

El procedimiento 'variabler' calcula las variables reales del octaedro objeto del cálculo, y las guarda en memoria.

El procedimiento 'variabled' calcula las variables del octaedro para su dibujo en pantalla.

El procedimiento 'medidas' reduce o aumenta las medidas de las variables de dibujo.

El procedimiento 'dibujo1' dibuja la parte superior del octaedro objeto del cálculo y la diagonal del mismo.

El procedimiento 'dibujo2' dibuja la parte inferior del octaedro objeto del cálculo. Con dos bips que produce el procedimiento 'pit' se concluye el cálculo y dibujo del octaedro.

II.3.4.4.- Ejemplos de aplicación - Cálculo del octaedro -

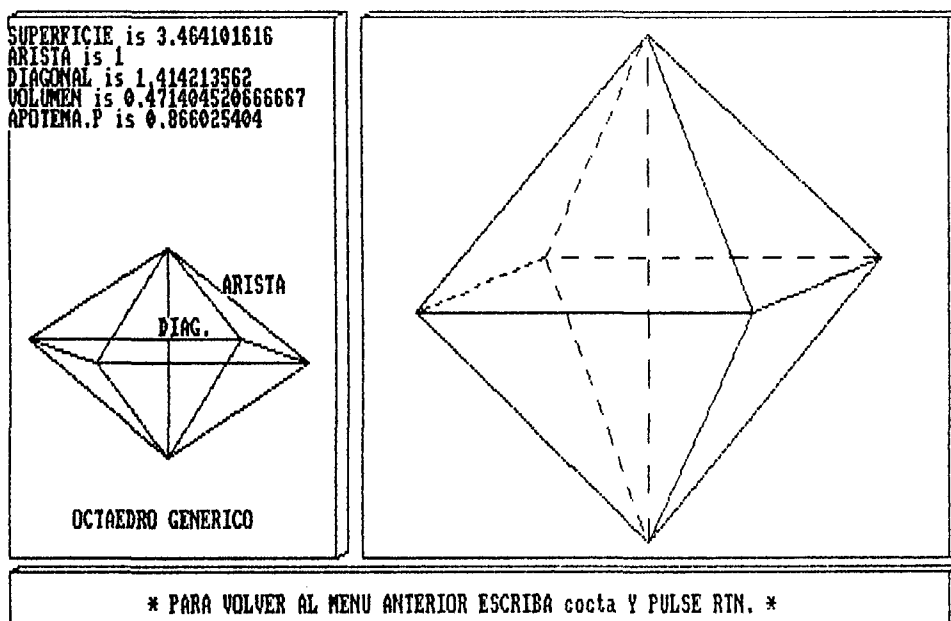


fig. (II.88.)

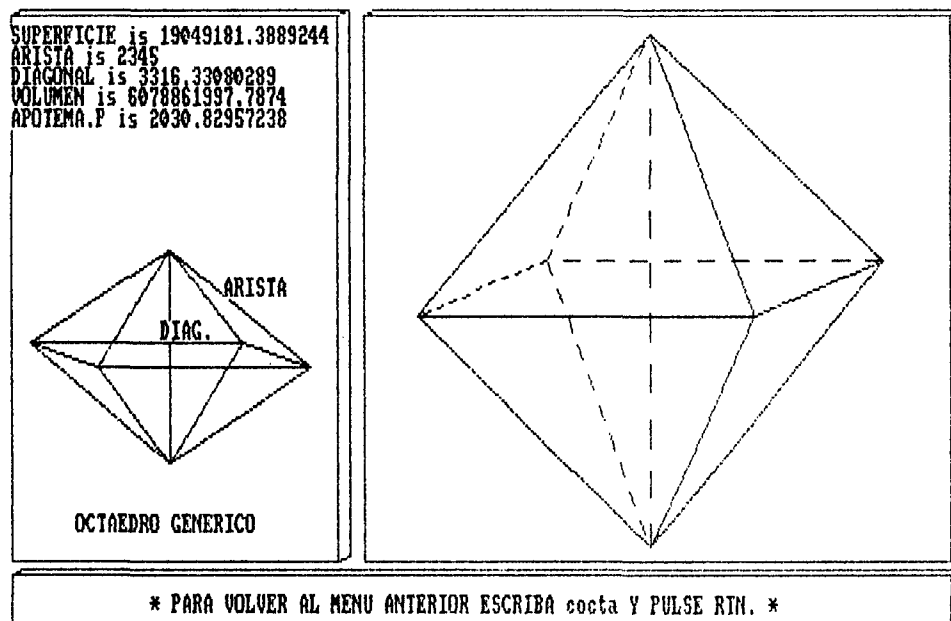


fig. (II.89.)

II.3.4.5.- Listado de procedimientos

Módulo "octa"

```
to sp
  erall ct fs load "espacio espacio"
end

to ctl :x :y :z
  repeat 2[fd :y rt 90 fd :x rt 90] fd :y
  rt 65 fd :z rt 25 fd :x rt 90 fd :y
  rt 65 fd :z rt 115 fd :y rt 65 fd :z
end

to pit :n
  repeat :n[type char 7]
end

to ot :x :y :angulo
  ots :x :y :angulo
  setcursor [10 27]
  pr [SI QUIERE EL DIBUJO SIN LEYENDAS AL PIE ESCRIBA ots EN VEZ DE ot]
  setcursor [14 28]
  pr [PARA VOLVER AL MENU ENTERIOR ESCRIBA octa Y PULSE RETURN]
end

to ots :x :y :angulo
  pd pir :x :y :angulo
  rt 90-:angulo fd :x*sin :angulo rt :angulo fd :x rt 90
  pd pir :x :y :angulo
end

to oct
  erall ct fs load "cocta cocta"
end
```

```

to pir :x :y :angulo
      (El listado de este procedimiento es el mismo
      que el listado de 'pir' en el módulo "piram").
end

```

```

to octa
  setsplit 1 ts ct pit 1 cs ht setcursor [29 3]
  pr [DIBUJO Y CALCULO DEL OCTAEDRO] pu home setpos [-150 180]
  pd ctl 270 50 10 pu home setpos [-145 185] pd
  ctl 260 40 0 pu home setpos [180 210] pd ots 120 50 20
  pu home setpos [-300 215] pd ots 50 50 -30 pu home
  setpos [-330 -160] pd ctl 655 310 10 pu home setpos [-330 -240]
  pd ctl 655 68 10 pu home setpos [-325 -155] pd ctl 645 300 0
  setcursor [10 10]
  pr [PARA EL DIBUJO DE UN OCTAEDRO ESCRIBA ot Y TRES ENTRADAS. LAS TRES]
  setcursor [9 12]
  pr [ENTRADAS SON LAS MEDIDAS DE LA BASE, ALTURA Y ANGULO DE VISION DE UNA]
  setcursor [9 14]
  pr [DE LAS PIRAMIDES QUE LO FORMAN. AL SER LA FIGURA SIMETRICA, BASTA CON]
  setcursor [14 16]
  pr [DAR LAS MEDIDAS DE LA PIRAMIDE PARA EL DIBUJO DEL OCTAEDRO.]
  setcursor [18 19]
  pr [EJEMPLO: ESCRIBA cs. ot120 100 25 Y PULSE RTN.] setcursor [17 26]
  pr [* PARA EL MENU ANTERIOR ESCRIBA sp Y PULSE RTN.*]
  setcursor [19 28]
  pr [* PARA EL CALCULO ESCRIBA oct Y PULSE RTN.*]
  pit 2 setcursor [0 30]
end

```

Módulo "cocta"

```

to ventana
  cs ht setsplit 1 pit 1
  pu home setpos [-360 -185] pd ctl 245 440 10
  pu home setpos [-95 -185] pd ctl 440 440 10
  pu home setpos [-360 -240] pd ctl 705 43 10
  pu home setpos [-295 30] pd ot 160 100 -20
  setcursor [6 24] pr [OCTAEDRO GENERICO]
  setcursor [20 13] pr [ARISTA] setcursor [14 15]
  pr [DIAG.] setcursor [0 30]
end

```

```

to pit :n
    repeat :n[type char 7]
end

```

```

to ctl :x :y :z
    repeat 2[fd :y rt 90 fd :x rt 90] fd :y
    rt 65 fd :z rt 25 fd :x rt 90 fd :y
    rt 65 fd :z rt 115 fd :y rt 65 fd :z
end

```

```

to pir :x :y :angulo
    (El listado es el mismo
     que en el módulo "piram".)
end

```

```

to oct
    erall ct fs load "octa octa
end

```

```

to cocta
    ts ct setsplit 1 pit 1 cs ht setcursor [32 3]
    pr [*CALCULO DEL OCTAEDRO*] pu home setpos [-120 180]
    pd ctl 225 50 10 pu home setpos [-115 185] pd
    ctl 215 40 0 pu home setpos [150 200] pd ot 150 60 20
    pu home setpos [-270 200] pd ot 90 70 -30 pu home
    setpos [-330 -140] pd ctl 655 290 10 pu home setpos [-325 -135]
    pd ctl 645 280 0 setcursor [12 12]
    pr [PARA EL CALCULO DE UN OCTAEDRO REGULAR ESCRIBA coc Y UNA ENTRADA.]
    setcursor [23 18]
    pr [EJEMPLO: ESCRIBA ...coc 240...Y PULSE RETURN.]
    setcursor [19 26]
    pr [* PARA EL MENU ANTERIOR ESCRIBA oct Y PULSE RTN. *]
    pit 2 setcursor [0 30]
end

```

```

to ot :x :y :angulo
    pd pir :x :y :angulo rt 90-:angulo fd :x*sin :angulo
    rt :angulo fd :x rt 90 pd pir :x :y :angulo
end

```



```

to coc :a
  pit 1 ventana make "x :a make "y :a
  variabler variabler medidas pu home
  setx 120-:X/2 sety 240-:Y/2 pd dibujo1
  pu rt 65 fd :x*sin 25 rt 25 fd :x rt 90 dibujo2
  ern [al M alt X C Y b c d e f g h i j k m n o p q s x z]
  setcursor [13 28]
  pr [*PARA VOLVER AL MENU ANTERIOR ESCRIBA cocta Y PULSE RTN.*]
  setcursor [0 1] pons recycle setcursor [0 30] pit 2
end

```

```

to variabler
  make "APOTEMA.P :a*1.732050808/2
  make "SUPERFICIE 2*:a*:a*1.732050808
  make "VOLUMEN :a*:a*:a*1.414213562/3
  make "DIAGONAL :a*1.414213562
  make "ARISTA :a
end

```

```

to medidas
  make "M 350 make "Y 2*:alt make "X :a+:z*cos 25
  if :Y>:X [make "C :M/:Y] make "C :M/:X
  make "x :a*:C make "z :z*:C make "Y :Y*:C make "X :X*:X
  make "d :d*:C make "h :h*:C make "k :k*:C make "o :o*:C
  make "b :b*:C make "m :m*:C
end

```

```

to variabler
  make "alt :y*cos 25 make "z :x*sin 25 make "b :z*sin 25
  make "al :z*cos 25 make "c (:x+:al)/2 make "f :alt/:c
  make "e arctan :f make "d :alt/sin :e make "s :alt-:b
  make "g arctan :s/:c make "h :c/cos :g
  make "i (:x-:al)/2 make "j arctan :i/:alt
  make "k :alt/cos :j make "m :alt-:b make "n arctan :i/:m
  make "o :i/sin :n make "p 90-:n make "q :p-25
end

```

```

to raya :a
  repeat 10[pu fd :a/20 pd fd :a/20]
end

```

to dibujo1

```
rt 65 raya :z rt 25 raya :x rt 155 fd :z rt 25 fd :x
rt 180-:e fd :d rt :e+90 raya 2*:m+:b pu bk 2*:m+:b
pd lt 90-:g fd :h bk :h rt 90-:g-:j fd :k bk :k
rt :j+:n yara :o rt :q pu fd :z rt 115
```

end

to dibujo2

```
rt 65 fd :z rt 25 fd :x rt 155 fd :z rt 25 fd :x
rt 180-:e pd fd :d rt :e+g fd :h bk :h rt 90-:g-:j
raya :k pu bk :k pd rt :j+:n fd :o rt :q pu fd :z pd rt 115
```

end

*LISTADO DE LA
BASE DE DATOS*

Módulo "bdatos"

```
to p
  ct fs erall
  load "plana
  ts plana
end
```

```
to e
  ct fs erall
  load "espacio
  ts espacio
end
```

```
to bd
  setsplit 1 ct fs ht
  loadpic "portada
end
```

```
to portada
  enmarca
  dibuja
  escribe
end
```

```
to enmarca
  setsplit 1 cs ct ht pu home setpos [-285 109] pd ctl 565 140 20
  pu home setpos [-280 114] pd ctl 555 130 0 pu home setpos [-320 -233]
  pd ctl 620 320 20 setcursor [33 5] pr [BASE DE DATOS PARA]
  setcursor [38 7] pr [GRAFICOS] setcursor [9 11]
  pr [PROGRAMA PARA EL CALCULO Y DIBUJO DE FIGURAS PLANAS Y TRIDIMENSION]
  setcursor [28 13] pr [AUTOR: Jorge Fortillo Meniz]
  pu home setpos [-101 123] pd
  repeat 77[fd 60 rt 90 pd fd 1 rt 90 px fd 60 lt 90 pd fd 1 lt 90 px]
  pu home setpos [-295 55] pd
  repeat 286[fd 25 rt 90 pd fd 1 rt 90 px fd 25 lt 90 pd fd 1 lt 90 px]
  pu home setpos [-140 20] pd
  repeat-111[fd 30 rt 90 pd fd 1 rt 90 px fd 30 lt 90 pd fd 1 lt 90 px]
end
```

```

to dibuja
  ct pu home setpos [-190 150]
  pd ctl 20 60 10
  pu home setpos [195 190]
  pd rt 20 octa 45 50 20
  pu home setpos [-265 190]
  pd ctl 40 18 20
  pu home setpos [130 195]
  pd cuadrado 30
  pu home setpos [-130 190]
  pd pir 55 40 20
  pu home setpos [30 192]
  pd cubo 30 150
  pu home setpos [75 130]
  pd dcta 100 70 30
end

```

```

to escribe
  setcursor [19 17]
  pr [PARA MAS INFORMACION DEL DIBUJO Y CALCULO DE LA]
  setcursor [19 19]
  pr [FIGURA QUE DESEE PULSE LAS TECLAS QUE SE INDICAN]
  setcursor [38 22] pr [- - - -]
  setcursor [10 25]
  pr [1) GEOMETRIA PLANA.....]
  setcursor [55 25] pr [ESCRIBA p Y PULSE RETURN.]
  setcursor [10 27]
  pr [2) GEOMETRIA DEL ESPACIO.....]
  setcursor [55 27] pr [ESCRIBA e Y PULSE RETURN.]
  setcursor [0 30]
end

```

```

to ctl :x :y :z
  repeat 2[fd :y rt 90 fd :x rt 90] fd :y
  rt 65 fd :z rt 25 fd :x rt 90 fd :y
  rt 65 fd :z rt 115 fd :y rt 65 fd :z
end

```

```

to cuadrado :lado
  repeat 4[fd :lado rt 90]
end

```

```

to raya :a
  repeat 6[pu fd :a/12 pd fd :a/12]
end

```

```

to dcta :a :c :alfa
  make "x :a*cos :alfa
  make "h :a*sin :alfa
  make "y :c-x
  make "beta arctan (:h/:y)
  make "b 90-:alfa
  make "d :h/sin :beta
  rt 90-:alfa fd :a rt :alfa+90 raya :h pu
  bk :h pd lt 90+:alfa rt :alfa+:beta
  fd :d rt 180-:beta fd :x+:y rt 90
end

```

```

to cubo :x :ang
  make "y :x make "z :x*sin :ang
  fd :y rt 90 fd :x rt 90 fd :y rt 90 fd :x
  rt 90 fd :y rt 90-:ang fd :z rt :ang fd :x
  rt 90 fd :y rt 90-:ang fd :z rt 90+:ang
  fd :y rt 90-:ang fd :z rt 90+:ang fd :y
  rt 90 fd :x rt 90 fd :y bk :y lt 90+:ang
  fd :z rt 90+:ang
end

```

```

to pir :x :y :ang
  make "alt :y*cos :ang
  make "z :x*sin :ang make "b :z*sin :ang
  make "a :z*cos :ang make "c (:x+a)/2
  make "f :alt/:c make "e arctan :f
  make "d :alt/sin :e rt 90-:ang
  fd :z rt :ang fd :x rt 180-:ang
  fd :z rt :ang fd :x rt 180-:e fd :d
  make "s :alt-b make "g arctan :s/:c make "h :c/cos :g
  rt :e+g fd :h make "i (:x-a)/2
  make "j arctan :i/:alt make "k :alt/cos :j
  bk :h rt 90-:g-:j fd :k make "m :alt-b
  make "n arctan :i/:m make "o :i/sin :m
  bk :k rt :j+:n fd :o make "q 90-:n-:ang
  rt :q pu fd :z pd rt 90+:ang
end

```

```
to octa :x :y :ang
  pd pir :x :y :ang
  if :ang>0 [make "alfa :ang-180]
  if :ang<0 [make "alfa :ang+180]
  if :ang=0 [make "alfa :ang]
  pir :x :y :alfa
end
```

Módulo "plana"

```
to plana
  ventana setcursor [21 8]
  pr[PARA MAS INFORMACION SOBRE EL DIBUJO Y CALCULO]
  setcursor [19 10]
  pr [DE LA FIGURA QUE DESEE PULSE LAS TECLAS SIGUIENTES]
  setcursor [36 13] pr [*****]
  setcursor [12 15] pr(a) TRIANGULOS.....]
  setcursor [52 15] pr[ESCRIBA a Y PULSE RETURN]
  setcursor [12 17] pr(b) CUADRILATEROS.....]
  setcursor [52 17] pr[ESCRIBA b Y PULSE RETURN]
  setcursor [12 19] pr(c) POLIGONOS.....]
  setcursor [52 19] pr[ESCRIBA c Y PULSE RETURN]
  setcursor [12 21] pr(d) RECTAS.....]
  setcursor [52 21] pr[ESCRIBA d Y PULSE RETURN]
  setcursor [12 23] pr(e) CURVAS.....]
  setcursor [52 23] pr[ESCRIBA e Y PULSE RETURN]
  setcursor [13 27]
  pr[*SI QUIERE VOLVER AL MENU ANTERIOR ESCRIBA bd Y PULSE RETURN*]
  setcursor [0 30] pit 2
end

to a
  erall ct fs
  load "triangulo
  ts triangulo
end

to pit :n
  repeat :n[type char 7]
end

to b
  erall ct fs
  load "poligonos
  poligonos
end
```



```
to c
  erall ct fs
  load "rectas
  rectas
end
```

```
to d
  erall ct fs
  load "curvas
  curvas
end
```

```
to ctl :x :y :z
  repeat 2[fd :y rt 90 fd :x rt 90] fd :y
  rt 65 fd :z rt 25 fd :x rt 90 fd :y
  rt 65 fd :z rt 115 fd :y rt 65 fd :z
end
```

```
to bd
  erall ct fs
  load "bdatos
  bd
end
```

```
to ventana
  setsplit 2 ht ts ct pit 1
  pu home setpos [-103 185] pd ctl 195 40 0
  pu home setpos [-108 180] pd ctl 205 50 10
  pu home setpos [-300 -160] pd ctl 600 310 0
  pu home setpos [-305 -165] pd ctl 610 320 10
  pu home setpos [-305 -225] pd ctl 610 40 10
  setcursor [35 3] pr [GEOMETRIA DEL PLANO]
end
```

Módulo "triángulo"

to triángulo

```
pit 1 marco setcursor [28 3]
pr [CALCULO Y DIBUJO DE TRIANGULOS] setcursor [19 8]
pr[DESDE ESTE MENU PODRA ECEDER A UN TIPO DETERMINADO]
setcursor [19 10]
pr[DE TRIANGULO, EL SIGUIENTE MENU LE PERMITIRA DIBUJAR]
setcursor [19 12]
pr[O HACER CALCULOS SOBRE EL TRIANGULO SELECCIONADO CON]
setcursor [30 14] pr[UN APORTE MINIMO DE DATOS.]
setcursor [10 18] pr[1) TRIANGULO RECTANGULO.....]
setcursor [50 18] pr[ESCRIBA dctr Y PULSE RETURN]
setcursor [10 20] pr[2) TRIANGULO ISOSCELES.....]
setcursor [50 20] pr[ESCRIBA dcti Y PULSE RETURN]
setcursor [10 22] pr[3) TRIANGULO EQUILATERO.....]
setcursor [50 22] pr[ESCRIBA dcte Y PULSE RETURN]
setcursor [10 24] pr[4) TRIANGULO CUALQUIERA.....]
setcursor [50 24] pr[ESCRIBA dctc Y PULSE RETURN]
setcursor [15 28]
pr[*PARA VOLVER AL MENU ANTERIOR ESCRIBA p Y PULSE RETURN*]
setcursor [0 30] pit 2
```

end

to marco

```
ct clean setsplit 1
pu home setpos [-160 168] pd ctl 290 70 0
pu home setpos [-165 163] pd ctl 300 80 10
pu home setpos [-310 -185] pd ctl 610 330 10
pu home setpos [-305 -180] pd ctl 600 320 0
pu home setpos [-310 -240] pd ctl 610 40 10
```

end

to dctr

```
erall fs
load "dctr dctr"
```

end

to dctc

```
erall ct fs
load "dctc dctc"
```

end

```
to dcti
    erall ct fs
    load "dcti dcti
end
```

```
to dcte
    erall ct fs
    load "dcte dcte
end
```

```
to ctl :x :y :z
    repeat 2[fd :y rt 90 fd :x rt 90] fd :y
    rt 65 fd :z rt 25 fd :x rt 90 fd :y
    rt 65 fd :z rt 115 fd :y rt 65 fd :z
end
```

```
to pit :n
    repeat :n[type char 7]
end
```

```
to p
    erall fs
    load "plana plana
end
```

Módulo "dctr"

```
to ttr :dato :ang
    setsplit 3 ht
    pd rt 90-:ang
    fd :dato rt 90+:ang
    fd :dato*sin :ang rt 90
    fd :dato*cos :ang rt 90
end

to dtr :dato :ang
    pu ts ct setsplit 3 ht
    pd rt 90-:ang
    fd :dato rt 90+:ang
    fd :dato*sin :ang rt 90
    fd :dato*cos :ang rt 90 setcursor [20 27]
    pr[SI QUIERE VOLVER AL MENU ANTERIOR ESCRIBA dctr Y PULSE RETURN]
    setcursor [17 29]
    pr[SI QUIERE EL DIBUJO SIN LEYENDAS ESCRIBA ttr EN VEZ DE DTR]
end

to ctr
    erall ct fs
    load "ctr ts ctr"
end

to t
    erall fs
    load "triangulo triangulo"
end

to recuadro
    ct clean ht pit 1
    pu home setpos [-155 190] pd ctl 295 60 10 25
    pu home setpos [-325 -145] pd ctl 655 325 10 25
    pu home setpos [-325 -225] pd ctl 655 70 10 25
end
```

```

to dctr
  recuadro
  texto
  salida
end

to salida
  setcursor [7 26]
  pr[PARA EL CALCULO ESCRIBA ctr Y PULSE RETURN, PARA MENU ANTERIOR t Y PULSE
  RETURN]
  setcursor [0 30] pit 2
end

to texto
  setcursor [29 2]
  pr[DIBUJO DEL TRIANGULO RECTANGULO]
  setcursor [19 6]
  pr[PARA DIBUJAR UN TRIANGULO RECTANGULO ESCRIBA dtr Y A]
  setcursor [14 8]
  pr[CONTINUACION EL VALOR DE LA HIPOTENUSA Y EL ANGULO DE LA BASE.]
  setcursor [22 11]
  pr[EJEMPLO: ESCRIBA cs..dtr 100 45 Y PULSE RETURN]
  setcursor [16 16]
  pr[EL PROGRAMA DIBUJARA UN TRIANGULO DE HIPOTENUSA = 100 Y EL]
  setcursor [16 18]
  pr[ANGULO DE LA BASE 45º, LOS OTROS DOS LADOS Y ANGULOS LOS]
  setcursor [22 20]
  pr[CALCULA EL PROGRAMA ANTES DE HACER EL DIBUJO.]
end

to ctl :x :y :z :ang
  repeat 2[fd :y rt 90 fd :x rt 90] fd :y
  rt 90-:ang fd :z rt :ang fd :x rt 90 fd :y
  rt 90-:ang fd :z rt 90+:ang fd :y rt 90-:ang fd :z
end

to pit :n
  repeat :n[type char 7]
end

```

Módulo "ctr"

```
to tr1 :a :b
  if (:b<90) [go "10]
  ts ct setcursor [25 16]
  pr[EL ANGULO NO PUEDE SER MAYOR O IGUAL A 90º] stop
label "10 make "dato :a/cos :b
  make "alfa :b
  ventanal ttr :dato :alfa
  ern[C1 C2 dato alfa dato.d]
  setcursor [0 1] pons
  setcursor [13 29]
  pr[SI QUIERE VOLVER AL MENU ANTERIOR ESCRIBA ctr Y PULSE RTN.]
  recycle setcursor [0 30] pit 2
end
```

```
to ventanal
  ventana pu home setpos [-330 -115] pd
  rt 50 fd 230 rt 130 fd 230*sin 40
  rt 90 fd 230*cos 40 rt 90
  setcursor [8 21] pr [ALFA]
  setcursor [21 16] pr [BETA]
  setcursor [21 21] pr [90º]
  setcursor [14 23] pr [A]
  setcursor [13 17] pr [H]
  setcursor [27 18] pr [B]
  setcursor [5 25] pr [TRIANGULO GENERICO]
end
```

```
to ventana
  cs ht setsplit 1 pit 1
  pu home setpos [-360 -185] pd ctl 245 440 10
  pu home setpos [-95 -185] pd ctl 440 440 10
  pu home setpos [-360 -240] pd ctl 705 35 10
end
```

```
to pit :n
  repeat :n[type char ?]
end
```

```

to ttr :dato :alfa
    variables medidas
    pd rt 90-:alfa fd :dato.d rt 90+:alfa
    fd (:dato.d*sin:alfa)
    rt 90 fd (:dato.d*cos :alfa) rt 90
end

to variables
    make "ALFA :alfa
    make "BETA 90-:alfa
    make "CATETO.A :dato*cos :alfa
    make "CATETO.B :dato*sin :alfa
    make "HIPOTENUSA :dato
    make "SUPERFICIE (:CATETO.A*:CATETO.B)/2
    make "PERIMETRO :HIPOTENUSA+:CATETO.A+:CATETO.B
end

to medidas
    make "dato.d 400
    make "C1 220-(:dato.d*cos :alfa)/2
    make "C2 220-(:dato.d*sin :alfa)/2
    pu setx :C1-95 sety :C2-185 pd
end

to tr2 :dato :alfa
    if (:alfa<90) [go "10]
    clean setcursor [20 15]
    pr[EL ANGULO NO PUEDE SER MAYOR O IGUAL A 90º] stop
label "10 ventana1 variables
    medidas dibujo.2 ern [C1 C2 dato.d]
    setcursor [0 1] pons setcursor [13 28]
    pr[SI QUIERE VOLVER AL MENU ANTERIOR ESCRIBA ctr Y PULSE RTN.]
    setcursor [0 30] recycle pit 2
end

to dibujo.2
    rt 90-:alfa fd :dato.d rt 90+:alfa fd (:dato.d*sin :alfa)
    rt 90 fd (:dato.d*cos :alfa) rt 90
end

```

```

to ctl :x :y :z
    repeat 2[fd :y rt 90 fd :x rt 90] fd :y
    rt 65 fd :z rt 25 fd :x rt 90 fd :y
    rt 65 fd :z rt 115 fd :y rt 65 fd :z
end

```

```

to ctr
    cs pit 1
    pu home setpos [-130 200] pd ctl 270 45 10
    pu home setpos [-335 -165] pd ctl 685 355 10
    pu home setpos [-335 -225] pd ctl 685 45 10
    texto
end

```

```

to texto
setcursor[30 2] pr[CALCULO DEL TRIANGULO RECTANGULO] setcursor [14 6]
pr[PARA CALCULAR LAS VARIABLES DE UN TRIANGULO RECTANGULO A PARTIR]
setcursor [19 8]
pr[DE UNOS DATOS MINIMOS, PULSE LAS TECLAS QUE SE INCICAN]
setcursor [6 11]
pr[1] UN CATETO Y UN ANGULO.....] setcursor [39 11]
pr[ESCRIBA tr1 (cateto) (ángulo) Y PULSE RETURN] setcursor [6 13]
pr[2] LA HIPOTENUSA Y UN ANGULO.....] setcursor [39 13]
pr[ESCRIBA tr2 (hipotenusa) (ángulo) Y PULSE RETURN]
setcursor [9 17]
pr[POR EJEMPLO: CONOCE LA HIPOTENUSA (250) Y UN ANGULO (40º), (Caso 2º), EN]
setcursor [9 19]
pr[ESTE SUPUESTO ESCRIBA..tr2 250 40 Y PULSE RETURN, A CONTINUACION SE
BORRA] setcursor [9 21]
pr[LA PANTALLA Y EL PROGRAMA DIBUJARA EL TRIANGULO AL MISMO TIEMPO QUE DA]
setcursor [17 23]
pr[LOS VALORES DE TODOS LOS PARAMETROS QUE DEFINEN EL TRIANGULO]
setcursor [15 27]
pr[SI QUIERE VOLCVER AL MENU ANTERIOR ESCRIBA dctr Y PULSE RETURN]
setcursor [0 30] pit 2
end

```

```

to dctr
    erall ct fs load "dctr dctr
end

```


Módulo "dcti"

```
to dcti
  recuadro setcursor [30 2]
  pr[DIBUJO DEL TRIANGULO ISOSCELES] setcursor [19 6]
  pr[SI QUIERE DIBUJAR UN TRIANGULO ISOSCELES TIENE QUE ESCRIBIR]
  setcursor [22 8]
  pr[dti Y A CONTINUACION DAR LOS DATOS DEL TRIANGULO]
  setcursor [22 11]
  pr [EJEMPLO: ESCRIBA cs..dti 200 150 Y PULSE RETURN]
  setcursor [15 14]
  pr[EL PROGRAM, A DIBUJARA UN TRIANGULO ISOSCELES DE 200 DE ALTURA]
  setcursor [15 16]
  pr[Y 150 DE BASE, LOS OTROS DOS LADOS Y ANGULOS LOS CALCULA EL]
  setcursor [26 18]
  pr[PROGRAMA ANTES DE DIBUJAR EL TRIANGULO]
  setcursor [22 23]
  pr[*PARA EL CALCULO ESCRIBA cti Y PULSE RETURN*]
  setcursor [20 25]
  pr[*PARA EL MENU ANTERIOR ESCRIBA t Y PULSE RETURN*]
  setcursor [0 30] pit 2
end

to cti
  erall ct fs
  load "cti ts cti"
end

to t
  erall ct fs
  load "triángulo triángulo"
end

to recuadro
  ct clean ht setsplit 2 pit 1
  pu home setpos [-140 200] pd cti 295 45 10
  pu home setpos [-290 -105] pd cti 600 290 10
  pu home setpos [-290 -190] pd cti 600 70 10
end
```

```

to ctl :x :y :z
    repeat 2[fd :y rt 90 fd :x rt 90] fd :y
    rt 65 fd :z rt 25 fd :x rt 90 fd :y
    rt 65 fd :z rt 115 fd :y rt 65 fd :z
end

```

```

to pit :n
    repeat :n[type char 7]
end

```

```

to dts :a :b
    make "ALFA arctan :b/(2*:a)
    make "c :a/cos :ALFA
    make "BETA 90-:ALFA
    rt 90-:BETA fd :c rt 2*:BETA fd :c
    rt 90+:ALFA fd :b rt 90
end

```

```

to dti :a :b
    dts :a :b
    setcursor [16 28]
    pr[PARA VOLVER AL MENU ANTERIOR ESCRIBA dti Y PULSE RTN]
    setcursor [15 29]
    pr[SI QUIERE EL DIBUJO SIN LEYENDAS ESCRIBA dts EN VEZ DE dti]
    setcursor [0 30]
end

```

Módulo "cti"

```

to ctl
    (Igual que 'ctl' del módulo "dcti")
end

```

```

to pit
    (Igual que 'pit' del módulo "dcti")
end

```

```

to ventanal
    ventana pu home setpos [-332 -115] pd
    iso 200 42
    setcursor [14 18] pr [BETA]
    setcursor [7 21] pr [ALFA]
    setcursor [15 23] pr [B]
    setcursor [8 19] pr [A]
    setcursor [23 19] pr [A]
    setcursor [6 25] pr [TRIANGULO GENERICO]
end

to ventana
    cs ht setsplit 1 pit 1
    pu home setpos [-360 -185] pd ctl 245 440 10
    pu home setpos [-95 -185] pd ctl 440 440 10
    pu home setpos [-360 -240] pd ctl 705 35 10
end

to variables
    make "B : b
    make "A : B/(2*cos :ALFA)
    make "ALTURA : A*sin :ALFA
    make "SUPERFICIE (:B*:ALTURA)/2
    make "PERIMETRO :A+:A+:B
end

to dcti
    erall ct fs
    load "dcti dcti
end

to cti
    marco texto setcursor [14 27]
    pr[SI QUIERE VOLVER AL MENU ANTERIOR ESCRIBA dcti Y PULSR RTN]
    setcursor [0 30] pit 2
end

```

```
to rayado :a
    repeat 6[pu fd (:a/12) pd fd (:a/12) pd]
end
```

```
to marco
    ct cs ht setsplit 2 pit 1 setcursor [30 2]
    pr[CALCULO DEL TRIANGULO ISOSCELES] pu home setpos [-130 202]
    pd cti 265 38 12 pu home setpos [-335 -165] pd cti 680 355 12
    pu home setpos [-335 -225] pd cti 680 45 12
end
```

```
to texto
    setcursor [10 5]
    pr[PARA CALCULAR LAS VARIABLES DE UN TRIANGULO ISOSCELES A PARTIR DE UNOS]
    setcursor [22 7]
    pr[DATOS MINIMOS, PULSE LAS TECLAS QUE SE INDICAN]
    setcursor [8 10]
    pr[1) LA BASE Y ANGULO OPUESTO.....]
    setcursor [41 10]
    pr[ESCRIBA tb1 (base) (angulo) Y PULSE RETURN]
    setcursor [8 12]
    pr[2) LA BASE Y ANGULO CONTIGUO.....]
    setcursor [41 12]
    pr[ESCRIBA tb2 (base) (ángulo) Y PULSE RETURN]
    setcursor [10 16]
    pr[EJEMPLO: SI CONOCE LA BASE.. (250) Y EL ANGULO CONTIGUO... (40º)]
    setcursor [10 18]
    pr[(Caso 2º), EN ESTE CASO ESCRIBA ..tb2 250 40... Y PULSE RETURN, A]
    setcursor [10 20]
    pr[CONTINUACION SE BORRA LA PANTALLA Y EL PROGRAMA DIBUJARA EL TRIANGULO]
    setcursor [16 22]
    pr[AL MISMO TIEMPO QUE DA LOS VALORES DE TODAS LAS VARIABLES]
end
```

```
to final
    setcursor [0 1] pons setcursor [13 28]
    pr[*SI QUIERE VOLVER AL MENU ANTERIOR ESCRIBA cti Y PULSE RTN*]
    setcursor [0 30] pit 2
end
```

```

to iso :b :al
    make "a :b/(2*cos :al)
    make "h :a*sin :al
    make "be 180-2*:al
    rt 90-:al fd :a rt 2*:al+:be/2 rayado :h
    pu bk :h pd lt :be/2 fd :a rt 180-:al fd :b
    rt 90 ern [a h be]
end

to tb1 :b :beta
    make "BETA :beta make "ALFA (180-:BETA)/2
    ventanal variables medidas dibujo final
end

to tb2 :b :alfa
    if (:alfa<90) [go "1] cs setcursor [25 15] pit 1
    pr[EL ANGULO CONTIGUO DE UN TRIANGULO] setcursor [33 17]
    pr[NO PUEDE SER MAYOR DE 90°] setcursor [0 30] pit 2 stop
label "1 make "ALFA :alfa make "BETA 180-2*:ALFA
    ventanal variables medidas dibujo final
end

to medidas
    make "M 400 if :A>ALTURA [make "Co :M/:A go "1]
    make "Co :M/:ALTURA
label "1 make "Ad :Co*:A make "Bd :Co*:B make "ALTURAd :Co*:ALTURA
    make "C1 220-:B/2-100 make "C2 220-:ALTURA/2-185
    pu setx :C1 sety :C2 pd
end

to dibujo
    rt 90-:ALFA fd :Ad rt 2*:ALFA+:BETA/2
    rayado :ALTURAd pu bk :ALTURAd pd lt :BETA/2
    fd :Ad pd rt 180-:ALFA fd :Bd rt 90
    ern [Ad Bd ALTURAd C1 C2 Co M]
end

```

Módulo "dcte"

```
to dcte
  recuadro setcursor [31 1]
  pr[DIBUJO DEL TRIANGULO EQUILATERO] setcursor [14 6]
  pr[PARA EL DIBUJO DE UN TRIANGULO EQUILATERO TIENE QUE ESCRIBIR dte]
  setcursor [21 8]
  pr[Y A CONTINUACION LA MEDIDA DE UNO DE SUS LADOS]
  setcursor [22 12]
  pr [EJEMPLO: ESCRIBA cs..dte 200 Y PULSE RETURN]
  setcursor [16 15]
  pr[EL PROGRAMA DIBUJARA UN TRIANGULO EQUILATERO DE LADO = 200]
  setcursor [20 20]
  pr[* PARA EL CALCULO ESCRIBA cte Y PULSE RETURN *]
  setcursor [18 22]
  pr[* PARA EL MENU ANTERIOR ESCRIBA t Y PULSE RETURN *]
  setcursor [0 29] pit 2
end

to cte
  erall ct fs load "cte cte"
end

to t
  erall ct fs load "triángulo triángulo"
end

to recuadro
  ct clean ht setsplit 2 pit 1
  pu home setpos [-140 219] pd ctl 295 40 10
  pu home setpos [-290 -55] pd ctl 590 235 10
  pu home setpos [-290 -135] pd ctl 590 65 10
end

to ctl :x :y :z
  repeat 2[fd :y rt 90 fd :x rt 90] fd :y
  rt 65 fd :z rt 25 fd :x rt 90 fd :y
  rt 65 fd :z rt 115 fd :y rt 65 fd :z
end
```

```
to pit :n
  repeat :n[type char 7]
end
```

```
to dte :a
  rt 30 repeat 2[fd :a rt 120] fd :a rt 90
end
```

Módulo "cte"

```
to ctl
  (Igual que 'ctl' del módulo "dcte")
end
```

```
to pit
  (Igual que 'pit' del módulo "dcte")
end
```

```
to ventana
  cs ht setsplit 1 pit 1
  pu home setpos [-360 -185] pd ctl 245 440 10
  pu home setpos [-95 -185] pd ctl 440 440 10
  pu home setpos [-360 -240] pd ctl 705 35 10
end
```

```
to variables
  make "LADO :a
  make "ALTURA :a*sin 60
  make "ALFA 60
  make "SUPERFICIE (:a*:ALTURA)/2
  make "PERIMETRO :LADO
end
```

```

to medidas
    make "A1 :LADO
    make "ALT. :ALTURA
    make "C 300/:A1
    make "A1 :A1*:C make "ALT. :ALT.*:C ern [C]
end

```

```

to cte
    ts marco texto salida
end

```

```

to rayado :a
    repeat 6[pu fd :a/12 pd fd :a/12]
end

```

```

to dcte
    erall ct fs
    load "dcte dcte
end

```

```

to marco
    pit 1 setsplit 1 cs ht pu home setpos [-130 205] pd
    ctl 270 33 10 pu home setpos [-335 -20] pd ctl 685 210 10
    pu home setpos [-335 -75] pd ctl 685 40 10
    setcursor [30 2] pr[CALCULO DEL TRIANGULO EQUILATERO]
end

```

```

to salida
    setcursor [13 18]
    pr[SI QUIERE VOLVER AL MENU ANTERIOR ESCRIBA dcte Y PULSE RETURN]
    setcursor [0 28] pit 2
end

```



```

to texto
setcursor [7 6]
pr[AL SER LOS TRES ANGULOS Y LADOS IGUALES SOLO ES NECESARIO SABER LO QUE]
setcursor [11 8]
pr[UNO DE LOS LADOS PARA PODER HALLAR TODAS LAS VARIABLES DEL TRIANGULO]
setcursor [9 10]
pr[ESCRIBA te (lado) Y PULSE RETURN..A CONTINUACION SE BORRARA LA PANTALLA]
setcursor [9 12]
pr[Y EL PROGRAMA DIBUJARA EL TRIANGULO A LA VEZ QUE DA TODAS LAS MEDIDAS]
setcursor [34 14]
pr [EJEMPLO: te 450]
end

```

```

to tg
ct cs ht pu home setpos [-320 -100] pd
rt 30 fd 150 rt 150 rayado 150*sin 60 pu
bk 150*sin 60 pd lt 30 fd 150 rt 120 fd 150
rt 90 setcursor [7 20] pr [ALFA]
setcursor [20 17] pr [LADO] setcursor [15 19]
pr [ALTURA] setcursor [5 23] pr [TRIANGULO GENERIC0]
setcursor [0 30]
end

```

Módulo "dctc"

```
to dctc
recuadro setcursor [25 2]
pr[DIBUJO Y CALCULO DE UN TRIANGULO CUALQUIERA] setcursor [6 6]
pr[A TRAVES DE ESTE MENU PUEDE ACCEDER AL CALCULO Y DIBUJO DE CUALQUIER
TRIANGULO.] setcursor [5 8]
pr[TIENE QUE APORTAR LOS DATOS MINIMOS NECESARIOS PARA EL CALCULO DE LA
FIGURA] setcursor [25 10]
pr [Y POSTERIOR DIBUJO POR EL PROGRAMA.] setcursor [6 13]
pr[SI CONOCE DOS LADOS Y UN ANGULO.....]
setcursor [57 13]
pr[ESCRIBA dct1 Y ESCRIBA RETURN] setcursor [6 15]
pr[* SI CONOCE UN LADO Y DOS ANGULOS.....]
setcursor [57 15]
pr[ESCRIBA dct2 Y PULSE RETURN] setcursor [14 18]
pr[ESTOS SON LOS DATOS MINIMOS NECESARIOS PARA EL CALCULO DE UN]
setcursor [14 20]
pr[TRIANGULO, SI CONOCE MAS DATOS NO SE LOS DE AL ORDENADOR, EL]
setcursor [14 22]
pr[PROGRAMA LOS CALCULA ANTES DE DIBUJAR.] setcursor [15 27]
pr[* PARA VOLVER AL MENU ANTERIOR ESCRIBA t Y PULSE RETURN*]
setcursor [0 30] pit 2
end
```

```
to dct1
    erall ct fs load "dct1 dct1"
end
```

```
to dct2
    erall ct fs load "dct2 dct2"
end
```

```
to recuadro
    pit 1 ct clean ht setsplit 2 pu home setpos [-175 200]
    ctl 370 40 0 pu home setpos [-180 195] pd ctl 380 50 10
    pu home setpos [-350 -175] pd ctl 700 350 10
    pu home setpos [-350 -225] pd ctl 700 40 10
end
```

```
to t
  erall fs load "triángulo triángulo"
end
```

```
to ctl :x :y :z
  repeat 2[fd :y rt 90 fd :x rt 90] fd :y rt 65 fd :z rt 25
  fd :x rt 90 fd :y rt 65 fd :z rt 115 fd :y rt 65 fd :z
end
```

```
to pit :n
  repeat :n[type chat 7]
end
```

Módulo "dct1"

```
to dct1
  pit 1 ht recuadro dibujo setcursor [37 1] pr[*D I B U J O *]
  setcursor [9 5]
  pr[PARA DIBUJAR UN TRIANGULO DEL QUE SE CONOCEN DOS LADOS Y EL ANGULO QUE]
  setcursor [9 7]
  pr[FORMAN, TIENE QUE ESCRIBIR, dt (lado1) (lado2) (angulo) Y PULSAR RTN.]
  setcursor [9 9]
  pr[EJEMPLO: SI ESCRIBE dt 260 300 30 Y PULSA RTN. EL PROGRAMA DIBUJARA UN]
  setcursor [5 12] pr[TRIANGULO COMO EL DE LA PANTALLA,]
  setcursor [5 14] pr[CON A = 260 B = 300 y ALFA = 30]
  setcursor [5 16] pr[EL RESTO DE LOS DATOS LOS CALCULA]
  setcursor [5 18] pr[EL PROGRAMA ANTES DE DIBUJAR.]
  setcursor [9 21] pr[PRUEBE EL EJEMPLO: es... dt 240 310 35]
  setcursor [19 25] pr[*PARA EL CALCULO ESCRIBA ctc1 Y PULSE RETURN*]
  setcursor [13 27]
  pr[*PARA VOLVER AL MENU ANTERIOR ESCRIBA detc Y PULSE RETURN*]
  setcursor [0 30] pit 2
end
```

```
to rayado :a
  repeat 6[pu fd :a/12 pd fd :a/12]
end
```

```
to dctc
    erall ct fs load "dctc dctc"
end
```

```
to dt :a :b :alfa
    dts :a :b :alfa
    setsplit 2 setcursor [13 27]
    pr[*PARA VOLVER AL MENU ANTERIOR ESCRIBA dct1 Y PULSE RETURN*]
    setcursor [10 29]
    pr[*SI QUIERE EL DIBUJO SIN NOTAS AL PIE ESCRIBA dts EN VEZ DE dt*]
end
```

```
to d :a :b :alfa
    variables dib
end
```

```
to ctcl
    erall ct fs load "ctcl ctcl"
end
```

```
to dts :a :b :alfa
    variables dib setsplit 1
end
```

```
to dibujo
    ct pu home ht sety -50 pd d 260 300 30
    setcursor [54 14] pr[A] setcursor [65 19] pr[B]
    setcursor [79 14] pr[C] setcursor [50 17] pr[ALFA]
    setcursor [76 17] pr[BETA] setcursor [62 12] pr[GAMMA]
    setcursor [71 15] pr[H]
end
```

```
to pit :n
    repeat :n[type char 7]
end
```

```

to ctl :x :y :z
  repeat 2[fd :y rt 90 fd :x rt 90] fd :y rt 70 fd :z
  rt 20 fd :x rt 90 fd :y rt 70 fd :z rt 110 fd :y rt 70 fd :z
end

```

```

to recuadro
  ct cs pu ho me setpos [-85 217] pd ctl 160 38 15
  pu home setpos [-340 -130] pd ctl 670 325 15
  pu home setpos [-340 -220] pd ctl 670 70 15
end

```

```

to variables
  ht make "x :a*cos :alfa
  make "ALTURA :a*sin :alfa
  make "y :b-:x
  make "BETA arctan (:ALTURA/:y)
  make "GAMMA 180-:alfa-:BETA
  make "C :ALTURA/sin :BETA
end

```

```

to dib
  rt 90 - :alfa fd :a rt :alfa + 90 rayado :ALTURA pu bk :ALTURA
  pd lt 90 - :BETA fd :C rt 180 - :BETA fd :x + :y rt 90
end

```

Módulo "ctl1"

```

to rayado :a
  (Igual al del módulo "dct1")
end

```

```

to ctl :x :y :z
  (Igual al del módulo "dct1")
end

```

```

to pit :n
  (Igual al del módulo "dct1")
end

```

```

to dt :a :b :alfa
  ventana variables medida dibujo if :C < 0 [make "C :C*-1]
  make "PERIMETRO :A + :B + :C if :ALFA < 90 [var]
  ern [x M Co dl y a1 b1 c1] setcursor [14 28]
  pr[*PARA VOLVER AL MENU ANTERIOR ESCRIBA ctc1 Y PULSE RTN.*]
  setcursor [0 1] pons recycle setcursor [0 30]
  ern [SUPERFICIE PERIMETRO ALTURA A B C ALFA BETA GAMMA.: pit 2
end

```

```

to ventana
  vent pu home setpos [-350 -115] pd tg 190 220 40
  setcursor [5 21] pr[ALFA] setcursor [22 21] pr [BETA]
  setcursor [16 17] pr [GAMMA] setcursor [7 18]
  pr[TRIANGULO GENERICO] setcursor [25 18] pr [C]
  setcursor [15 19] pr [ALT] setcursor [0 30]
end

```

```

to vent
  cs ht setsplit 1 pit 1 pu home setpos [-360 -185]
  pd ctl 245 440 10 pu home setpos [-95 -185] pd
  ctl 440 440 10 pu home setpos [-360 -240] pd ctl 705 35 10
end

```

```

to texto
  setcursor [5 5]
  pr[LOS DATOS QUE CONOCE SON DOS LADOS Y EL ANGULO QUE FORMAN, CON ESTOS]
  setcursor [5 7]
  pr[DATOS HALLARA DE FORMA INMEDIATA LOS OTROS DOS ANGULOS, EL OTRO LADO]
  setcursor [5 9]
  pr[LA SUPERFICIE, EL PERIMETRO Y LA ALTURA.] setcursor [5 11]
  pr[EJEMPLO: SI LAS MEDIDAS DE LOS LADOS SON:] setcursor [5 13]
  pr[260, 300 Y EL ANGULO 30º, ESCRIBA.....] setcursor [5 15]
  pr[..dt 260 300 30.... Y PULSE RETURN] setcursor [5 17]
  pr[EL PROGRAMA DIBUJARA EL TRIANGULO DE] setcursor [5 19]
  pr[LA PANTALLA, LOS DATOS QUE APORTAMOS] setcursor [5 21]
  pr[CORRESPONDEN EN EL DIBUJO CON (A, B Y ALFA).] setcursor [5 23]
  pr[EL RESTO DE LOS DATOS LOS CALCULA EL PROGRAMA Y LOS MUESTRA EN PANTALLA]
  setcursor [13 27]
  pr[*PARA VOLVER AL MENU ANTERIOR ESCRIBA dct1 Y PULSE RETURN*]
  setcursor [0 29]
end

```

```

to variables
  make "x :a*cos :alfa make "ALTURA :a*sin :alfa
  make "y :b-x make "ALFA :alfa make "BETA arctan (:ALTURA/:y)
  make "GAMMA 180-:alfa-BETA make "SUPERFICIE :b*:ALTURA/2
  make "A :a make "B :b make "C :ALTURA/sin :BETA
end

to medida
  make "D -:a*cos :alfa if :alfa<90 [make "D 0]
  make "D1 -:B+A*cos :alfa if :alfa>90 [make "D1 0]
  if and (:alfa<90) (:D1<0) [make "D 0 make "D1 0]
  make "D2 :D+:D1+:B
  make "M 400 make "a1 :a make "b1 :b make "c1 :C make "d1 :ALTURA
  if :D2>d1 [make "Co :M/:D2 go "1] make "Co :M/:d1
label "1 make "a1 :a1*:Co make "b1 :b1*:Co make "c1 :c1*:Co make "d1 :d1*:Co
  make "D :D*:Co make "D1 :D1*:Co make "D2 :D2*:Co
end

to ctcl
  pit 1 cs setsplit 2 pu home setpos [-100 217] pd ctl 170 40 10
  pu home setpos [-340 -170] pd ctl 660 370 10 pu home
  setpos [-340 -225] pd ctl 660 40 10 setcursor [35 1]
  pr[* C A L C U L O *] dib texto pit 2
end

to tg :uno :dos :tres
  make "x :uno*cos :tres make "ALTURA :uno*sin :tres
  make "y :dos-x make "BETA arctan (:ALTURA/:y)
  make "GAMMA 180-:tres-BETA make "C :ALTURA/sin :BETA
  rt 90-:tres fd :uno rt :tres+90 rayado :ALTURA pu
  bk :altura PD lt 90-:BETA fd :C rt 180-:BETA fd :x+:y
  rt 90 ern [x ALTURA y BETA GAMMA C]
end

to dibujo
  make "C1 125-:D2+:d make "C2 25-:d1/2
  pu setx :C1 sety :C2 pd ern [D D1 D2 C1 C2]
  rt 90-:alfa fd :a1 rt :alfa+90 rayado :d1 pd bk :d1
  pd lt 90 - :BETA fd :c1 rt 180-:BETA fd :b1 rt 90
end

```

```
to dib
  ct pu home ht sety -50 pd tg 260 300 30 setcursor [54 14]
  pr[A] setcursor [65 19] pr[B] setcursor [79 14] pr[C]
  setcursor [50 17] pr[ALFA] setcursor [76 17] pr[BETA]
  setcursor [62 12] pr[GAMMA] setcursor [71 15] pr[H]
end

to var
  if :b/cos :alfa < :a [make "BETA 180+:BETA make "GAMMA :GAMMA-180]
end
```

```
to dct1
  erall ct fs load "dct1 dct1"
end
```

Módulo "dct2"

```
to rayado :a
  (Igual al del módulo "ctc1")
end
```

```
to dctc
  erall ct fs load "dctc dctc"
end
```

```
to dt :alfa :beta :a
  dts :alfa :beta :a setcursor [11 28]
  pr[*PARA VOLVER AL MENU ANTERIOR ESCRIBA dct2 Y PULSE RETURN*]
  setcursor [10 29]
  pr[*SI QUIERE EL DIBUJO SIN NOTAS AL PIE ESCRIBA dts EN VEZ DE dt*]
end
```



```

to ctc2
    erall ct fs load "ctc2 ctc2
end

```

```

to dibujo
    ct pu home sety -50 pd dts 30 60 260 setcursor [54 14]
    pr[A] setcursor [65 19] pr [B] setcursor [79 14] pr [C]
    setcursor [50 17] pr [ALFA] setcursor [76 17] pr [BETA]
    setcursor [69 12] pr [GAMMA] setcursor [71 15] pr [H]
end

```

```

to dct2
    pit 1 ct cs ht setsplit 1 pu home setpos [-75 220] pd ctl 140 35 15
    pu home setpos [-340 -130] pd ctl 670 325 15
    pu home setpos [-340 -220] pd ctl 670 70 15
    dibujo setcursor [37 1] pr[* D I B U J O *] setcursor [5 5]
    pr[PARA DIBUJAR UN TRIANGULO DEL QUE CONOCE DOS ANGULOS Y UN LADO ESCRIBA.]
    setcursor [5 7]
    pr[..dt (angulo1) (angulo2) (lado)... Y PULSE RETURN.]
    setcursor [5 9]
    pr[EJEMPLO:....SI ESCRIBE dt 30 60 260 Y PULSA RETURN EL PROGRAMA DIBUJARA]
    setcursor [5 11]
    pr[UN TRIANGULO COMO EL DE LA PANTALLA] setcursor [5 13]
    pr[CON ALFA = 30, BETA = 60 Y A = 260] setcursor [5 15]
    pr[EL RESTO DE LOS DATOS LOS CALCULA EL] setcursor [5 17]
    pr[PROGRAMA ANTES DE DIBUJAR.] setcursor [5 21]
    pr[PRUEBE EL EJEMPLO: cs...dt 40 50 250...] setcursor [19 25]
    pr[*PARA EL CALCULO ESCRIBA ctc2 Y PULSE RETURN*] setcursor [13 27]
    pr[*PARA VOLVER AL MENU ANTERIOR ESCRIBA dctc Y PILSE RETURN*]
    setcursor [0 30] pit 2
end

```

```

to pit :n
    repeat :n[type char 7]
end

```

```

to ctl :x :y :z
    (Igual al del módulo "dct2")
end

```

```

to dts :alfa :beta :a
  if :alfa+:beta>180 [cs setcursor [10 15] pr[LA SUMA DE LOS TRES
  ANGULOS DE UN TRIANGULO ES 180º.....:alfa+:beta<180] stop]
  ht make "x :a*cos :alfa if :x<0 [make "x 0]
  make "ALTURA :a*sin :alfa make "GAMMA 180-:alfa-:beta
  make "C :ALTURA/sin :beta make "y :C*cos :beta
  if :x=0 [make "y :y-:a*cos 180-:alfa]
  rt 90-:alfa fd :a rt :alfa+90 rayado :ALTURA pu bk :ALTURA
  pd lt 90-:beta fd :C rt 180-:beta fd :x+:y rt 90
  ern [x ALTURA GAMMA C y] pit1
end

```

Módulo "ctc2"

```

to medida
  make "M 400 make "a1 :a make "b0 :b make "c1 :c make "d1 :ALTURA
  make "b2 :A*cos 180-:alfa if :b2<0 [make "b2 :b2*-1]
  if :alfa>90 [make "b0 :c1*cos :beta]
  make "b3 :C*cos 180-:beta
  if :b0>:d1 [make "C0 :M/:b0 go "1] make "Co :M/:d1
label "1 make "a1 :a1*:Co make "b1 :b1*:Co make "c1 :c1*:Co
  make "d1 :d1*:Co make "b2 :b2*:Co make "b3 :b3*:Co
end

```

```

to rayado :a
(Igual al del módulo anterior)
end

```

```

to d :uno :dos :tres
(Igual a 'tg' en el módulo "ctc1")
end

```

```

to tri
  ct pu home ht sety -50 pd d 260 300 30 setcursor [54 14] pr[A]
  setcursor [65 19] pr[B] setcursor [79 14] pr[C]
  setcursor [50 17] pr[ALFA] setcursor [76 17] pr[BETA]
  setcursor [69 12] pr[GAMMA] setcursor [71 15] pr[H]
end

```

```

to ctc2
pit 1 cs ht setsplit 1 pu home setpos [-95 215] pd ctl 170 40 10
pu home setpos [-340 -170] pd ctl 680 370 10 pu home
setpos [-340 -225] pd ctl 680 40 10 tri setcursor [35 1]
pr[*C A L C U L O *] setcursor [5 5]
pr[LOS DATOS QUE CONOCE SON DOS ANGULOS Y UN LADO. CON ESTOS DATOS PODRA]
setcursor [5 7]
pr[HALLAR DE FORMA INMEDIATA LOS OTROS DOS LADOS EL OTRO ANGULO LA ALTURA]
setcursor [5 9]
pr[LA SUPERFICIE Y EL PERIMETRO. EJEMPLO:] setcursor [5 11]
pr[SI LOS VALORES DE LOS ANGULOS SON 30º] setcursor [5 13]
pr[Y 60º Y EL LADO 260, ESCRIBA.....] setcursor [5 15]
pr[dt 30 60 260..... Y PULSE RETURN.] setcursor [5 17]
pr[EL PROGRAMA DIBUJARA EL TRIANGULO DE] setcursor [5 19]
pr[LA PANTALLA. LOS DATOS QUE CONOCEMOS] setcursor [5 21]
pr[CORRESPONDEN EN EL DIBUJO CON (ALFA,BETA Y A).] setcursor [5 23]
pr[EL RESTO DE LOS DATOS LOS CALCULA EL PROGRAMA Y SE LOS MUESTRA EN
PANTALLA.] setcursor [13 27]
pr[*PARA VOLVER AL MENU ANTERIOR ESCRIBA dct2 Y PULSE RETURN*]
setcursor [0 30] pit 2
end

```

```

to dt :alfa :beta :a
ventana variables medida if :alfa<90 [make "b2 0]
if :beta<90 [make "b3 0]
make "C1 125-:b1/2+:b2/2-:b3/2 make "C2 35-:d1/2 pu home
setx :C1 sety :C2 pd
rt 90-:alfa fd :a1 rt :alfa+90 rayado :d1 pu bk :d1
pd lt 90-:beta fd :c1 rt 180-:beta fd :b1 rt 90
if :C<0 [make "C :C*-1]
make "SUPERFICIE :B*:ALTURA/2
make "PERIMETRO :a+:B+:C
make "ALFA :alfa make "BETA :beta make "A :a
ern [b0 b2 x y C1 C2 Co M a1 b1 c1 d1] setcursor [14 28]
pr[PARA VOLVER AL MENU ANTERIOR ESCRIBA ctc2 Y PULSE RTN.]
setcursor [0 1] pons ern [SUPERFICIE GAMMA ALFA BETA A B C PERIMETRO
ALTURA] setcursor [0 30 ] recycle pit 2
end

```

```

to variables
make "x :a*cos :alfa make "ALTURA :a*sin :alfa
make "GAMMA 180-:alfa-:beta make "C :ALTURA/sin :beta
make "y :C*cos :beta make "B :x+:y make "A :a
end

```

```
to dct2
    erall ct fs load "dct2 dct2
end
```

```
to pit :n
(Igual al del módulo anterior)
end
```

```
to ctl :x :y :z
(Igual al del módulo anterior)
end
```

```
to gen
    pu home setpos [-350 -185] pd d 150 220 40 setcursor [5 20] pr[ALFA]
    setcursor [21 20] pr[BETA] setcursor [13 17] pr[GAMMA]
    setcursor [5 18] pr[A] setcursor [15 22] pr[B] setcursor [4 23]
    pr[TRIANGULO GENERICO] setcursor [24 18] pr[C]
    setcursor [12 19] pr [ALT.] setcursor [0 30]
end
```

```
to ventana
    ct ht setsplit 1 pit 1 pu home setpos [-360 -185]
    pd ctl 245 440 10 pu home setpos [-95 -185] pd
    ctl 440 440 10 pu home setpos [-360 -240] pd
    ctl 705 35 10 gen setcursor [0 30]
end
```

Módulo "polígono"

```
to ctl :x :y :z
    repeat 2[fd :y rt 90 fd :x rt 90] fd :y
    rt 65 fd :z rt 25 fd :x rt 90 fd :y
    rt 65 fd :z rt 115 fd :y rt 65 fd :z
end

to pit :n
    repeat :n[type char 7]
end

to pol :lado :n
    pols :lado :n
    setcursor [10 28]
    pr[SI QUIERE EL DIBUJO SIN NOTAS AL PIE ESCRIBA pols EN VEZ DE pol]
    setcursor [15 29]
    pr[PARA VOLVER AL MENU ANTERIOR ESCRIBA pl Y PULSE RETURN]
    setcursor [0 30]
end

to recuadro
    setsplit 2 ht ts ct pit 1 pu home setpos [-103 185] pd
    ctl 195 40 0 pu home setpos [-108 180] pd ctl 205 50 10
    pu home setpos [-300 -120] pd ctl 600 270 0 pu home
    setpos [-305 -125] ps ctl 610 280 10 pu home setpos [-305 -195]
    pd ctl 610 50 10 setcursor [34 3] pr[* P O L I G O N O S *]
end

to p
    erall ct fs load "plana plana
end

to pl
    poligono
end
```

```

to poligono
  recuadro setcursor [17 10]
  pr[FARA DIBUJAR CUALQUIER POLIGONO BASTA CON AFORTAR COMO]
  setcursor [16 12]
  pr[DATOS EL NUMERO DE LADOS Y LA MEDIDA DE UNO DE LOS LADOS]
  setcursor [20 14]
  pr[EJEMPLO: ESCRIBA ..pol 125 6.. Y PULSE RETURN.]
  setcursor [20 18]
  pr[EL PROGRAMA DIBUJARA UN EXAGONO DE LADO = 125]
  setcursor [13 25]
  pr[* SI QUIERE VOLVER AL MENU ANTERIOR ESCRIBA p Y PULSE RETURN *]
  setcursor [0 30] pu home pd pit 2
end

```

```

to pols :lado :n
  make "ang (:n-2)*180/:n
  repeat :n[fd :lado rt 180-:ang] ern [ang]
end

```

Módulo "recta"

```
to ctl :x :y :z
    repeat 2[fd :y rt 90 fd :x rt 90] fd :y
    rt 65 fd :z rt 25 fd :x rt 90 fd :y
    rt 65 fd :z rt 115 fd :y rt 65 fd :z
end

to pit :n
    repeat :n[type char 7]
end

to p
    erall ct fs load "plana plana
end

to tri :lado
    rt 30 fd :lado repeat 2[rt 120 fd :lado]
end

to div
    pu home setpos [-350 0] pd
    repeat 14[repeat 4[rt 90 pu fd 10 pd lt 90 pd fd 2 bk 4 fd 2] rt 90 fd 10
    lt 90 fd 5 bk 10 fd 5] pu setpos [0 -200] pd lt 90
    repeat 8[repeat 4[rt 90 pu fd 10 lt 90 pd fd 2 bk 4 fd 2]rt 90 fd 10 lt 90
    fd 5]
end

to ejes
    pu home clean setcursor [0 30] pit 1 setpos [-360 -247] pd setsplit 1
    repeat 2[fd 509 rt 90 fd 719 rt 90] pu home setpos [-355 -242] pd
    repeat 2[fd 499 rt 90 fd 709 rt 90] pu home
    pd fd 220 lt 90 fd 4 rt 90 tri 8 rt 180 fd 4 rt 90 fd 440
    lt 90 fd 4 rt 90 tri 8 setcursor [44 1] pr[(Y)]
    setcursor [86 14] pr [(X)] setcursor [0 30] pu home pd div pit 2
end
```

```

to recuadro
  setsplit 2 ht ts ct pit 1 pu home setpos [-95 185]
  pd ctl 190 40 0 pu home setpos [-105 180] pd ctl 208 50 10
  pu home setpos [-300 -125] pd ctl 595 275 0 pu home
  setpos [-305 -130] pd ctl 605 285 10 pu home setpos [-305 -195]
  pd ctl 605 50 10 setcursor [35 3]
  pr [* DIBUJO DE RECTAS *]
end

```

```

to re :a :m
  pu home setsplit 1
  make "M 470 make "y :a-325*:m make "ag arctan :m
  make "d 325/cos :ag make "C 2*:d*sin :ag
  if :ag=0 [make "b 0 make "C3 0 make "C1 2*:a go "1]
  make "C1 0 make "C3 0
  if :C>:M [make "C1 :C-:M make "C2 :C1/sin :ag]
  if :C>:M [make "C3 :C2*cos :ag make "d :d-:C2/2]
  make "b (:a*cos :ag)/sin :ag
label "1 setx -325-:b+C3/2 sety :y-:a+:C1/2 pd
  rt 90-:ag fd 2*:d setcursor [0 30] pit 1
end

```

```

to recta
  recuadro setcursor [18 8]
  pr[PARA DIBUJAR CUALQUIER RECTA BASTA CON APORTAR COMO]
  setcursor [16 10]
  pr[DATOS EL VALOR DEL TERMINO INDEPENDIENTA Y LA PENDIENTE]
  setcursor [24 13]
  pr[POR EJEMPLO: EN LA ECUACION Y = 65 + 1.2 X] setcursor [15 15]
  pr[LA PENDIENTE VALE (+1.2) Y EL TERMINO INDEPENDIENTE (65)]
  setcursor [10 17]
  pr[PARA DIBUJAR ESTA RECTA HAY QUE ESCRIBIR re 65 1.2 Y PULSAR RTN.]
  setcursor [23 19] pr[SI ESCRIBE ejes ANTES DE DIBUJAR LA RECTA]
  setcursor [24 21] pr[LA DIBUJARA SOBRE UNOS EJES CARTESIANOS]
  setcursor [14 25]
  pr[* SI QUIERE VOLVER AL MENU ANTERIOR ESCRIBA p Y PULSE RETURN *]
  setcursor [0 30] pu home pd pit 2
end

```


Módulo "curva"

```
to ctl :x :y :z
    repeat 2[fd :y rt 90 fd :x rt 90] fd :y
    rt 65 fd :z rt 25 fd :x rt 90 fd :y
    rt 65 fd :z rt 115 fd :y rt 65 fd :z
end

to pit :n
    repeat :n[type char 7]
end

to p
    erall ct fs load "plana plana
end

to a
    erall ct fs load "curvaa curvaa
end

to b
    erall ct fs load "curvab curvab
end

to recuadro
    setsplit 2 ht ts ct pit 1 pu home setpos [-100 200] pd
    ctl 185 40 0 pu home setpos [-110 195] pd ctl 203 50 10
    pu home setpos [-300 -135] pd ctl 595 310 0 pu home
    setpos [-304 -140] pd ctl 605 320 10 pu home setpos [-305 -210]
    pd ctl 605 50 10 setcursor [34 2] pr [* DIBUJO DE CURVAS *]
end
```

```

to curva
recuadro setcursor [15 8]
pr [PARA EL DIBUJO DE LOS DIFERENTES TIPOS DE CURVAS PULSE LAS]
setcursor [34 10] pr [TECLAS QUE SE INDICAN]
setcursor [10 14] pr [a) CURVAS TRIGONOMETRICAS.....]
setcursor [57 14] pr [ESCRIBA a Y PULSE RTN.]
setcursor [10 17] pr [b) OTRAS CURVAS.....]
setcursor [57 17] pr [ESCRIBA b Y PULSE RTN.] setcursor [9 21]
pr [NOTA: ANTES DE DIBUJAR LAS DIFERENTES CURVAS ESCRIBA ejes Y PULSE RTN.]
setcursor [17 26]
pr [* PARA VOLVER AL MENU ANTERIOR ESCRIBA p Y PULSE RTN *]
pit 2 setcursor [0 30]
end

```

Módulo "curvaa"

```

to ejes
er "curvaa er "curva1 er "curva2 er "recuadro pu home clean
setcursor [0 30] pit 1 setpos [-360 -247] pd
repeat 2[fd 509 rt 90 fd 719 rt 90] pu home setpos [-355 -242]
pd repeat 2[fd 499 rt 90 fd 709 rt 90] pu hoe setpos [50 -242]
pd fd 35 rt 90 fd 305 pu home pd fd 220 lt 90 fd 4 rt 90 tri 8
rt 180 rt 4 rt 90 fd 440 lt 90 fd 4 rt 90 tri 8 bk 4 lt 90 fd 220
rt 90 fd 345 lt 90 fd 4 rt 90 tri 8 bk 4 lt 90 fd 690 lt 90 fd 4
rt 90 tri 8 setcursor [52 28]
pr [MENU ANTERIOR → ESCRIBA a Y PULSE RTN] setcursor [44 1]
pr [(Y)] setcursor [86 14] pr [(X)] setcursor [0 30]
pu home pd div recycle pit 2
end

```

```

to tri :l
rt 30 fd :l repeat 2[rt 120 fd :l]
end

```

```

to pit :n
repeat :n[type char 7]
end

```

```

to div
  pu home setpos [-350 0] pd
  repeat 14[repeat 4[rt 90 pu fd 10 pd lt 90 fd 2 bk 4 fd 2]
  rt 90 pu fd 10 lt 90 pd fd 5 bk 10 fd 5]
  pu setpos [0 -200] pd lt 90
  repeat 8[repeat 4[rt 90 pu fd 10 lt 90 pd fd 2 bk 4 fd 2]
  rt 90 fd 10 lt 90 fd 5 bk 10 fd 5]
end

to ctl :x :y :z
  repeat 2[fd :y rt 90 fd :x rt 90] fd :y rt 65 fd :z rt 25
  fd :x rt 90 fd :y rt 65 fd :z rt 115 fd :y rt 65 fd :z
end

to c
  erall ct fs load "curva curva
end

to a
  ct fs load "curvaa curvaa
end

to curvaa
  recuadro curva1 curva2
end

to recuadro
  setsplit 1 ht ts ct pit 1 pu home setpos [-125 185] pd
  ctl 215 40 0 pu home setpos [-135 180] pd ctl 232 50 10
  pu home setpos [-300 185] pd ctl 595 345 0 pu home
  setpos [-305 -190] pd ctl 605 355 10 pu home
  setpos [-305 -240] pd ctl 605 40 10 setcursor [30 3]
  pr [*CURVAS TRIGONOMETRICAS*]
end

```

```

to curval
  setcursor [10 7]
  pr [SENO(X): PARA EL DIBUJO DE ESTA CURVA ESCRIBA seno Y DOS ENTRADAS]
  setcursor [10 8]
  pr [(AMPLITUD MAXIMA Y FCIA.).EJEMPLO...seno 100 4.. Y PULSE RTN.]
  setcursor [10 10]
  pr [COSENO(X): PARA EL DIBUJO DE ESTA CURVA ESCRIBA coseno Y DOS
  ENTRADAS] setcursor [10 11]
  pr [(AMPLITUD MAXIMA Y FCIA.). EJEMPLO..coseno 140 6..Y PULSE RTN.]
  setcursor [10 13]
  pr [TANGENTE(X): PARA EL DIBUJO DE ESTA CURVA ESCRIBA tang Y DOS]
  setcursor [10 14]
  pr [ENTRADAS (VALOR MAXIMO Y DEL EJE (X) DONDE COMIENZA A DIBUJAR).]
  setcursor [23 15]
  pr [(EJEMPLO...tang 100 -270.. Y PULSE RETURN]
end

```

```

to curva2
  setcursor [10 17]
  pr [COTANGENTE(X): PARA EL DIBUJO DE ESTA CURVA ESCRIBA cotang Y DOS]
  setcursor [10 18]
  pr [ENTRADAS (VALOR MAXIMO Y DEL EJE (X) DONDE COMIENZA A DIBUJAR).]
  setcursor [23 19]
  pr [(EJEMPLO...cotang 120 -300 ..Y PULSE RTN.))]
  setcursor [10 21]
  pr [SECANTE(X): PARA EL DIBUJO DE ESTA CURVA ESCRIBA sec Y DOS ENTRADAS]
  setcursor [10 22]
  pr [(AMPLITUD MAXIMA Y FCIA.). EJEMPLO ..sec 80 2..Y PULSE RTN.]
  setcursor [COSECANTE(X): PARA EL DIBUJO DE LA CURVA ESCRIBA cosec Y DOS
  ENTRADAS] setcursor [10 25]
  pr [(AMPITUD MAXIMA Y FCIA.). EJEMPLO cosec 55 2.. Y PULSE RTN.]
  setcursor [17 28]
  pr [* PARA VOLVER AL MENU ANTERIOR ESCRIBA c Y PULSE RTN *]
  pit 2 setcursor [0 30]
end

```

```

to seno :a :b
  make "x -360 make "alfa 360/:b
label "1  make "y :a*sin (:b*:alfa) dot list :x :y
          dot list -:x -:y make "x :x+2 make "alfa :alfa+1
          if :x>0 [recycle stop] go "1
end

```

```

to coseno :a :b
  make "x -360 make "alfa 360/:b
label "1   make "y :a*cos (:b*:alfa)
           if or quotient :b 2 < :b/2 :b=1 [make "y -:y]
           dot list :x :y dot list -:x -:y
           make "x :x+2 make "alfa :alfa+1
           if :x>0 [recycle stop] go "1
end

to tang :a :x
  if :x>0 [stop]
  if or :x= -90 :x= -270 [make "x :x+1]
label "1   make "y :a*sin (:x)/cos (:x)
           if :y>240 [make "x :x+2 if or :x= -90 :x= -270 [make "x :x+1]
           go "1] dot list :x :y dot list -:x -:y tang :a :x+2
end

to cotang :a :x
  if :x>0 [stop]
  if or :x< -90 :x< -270 [make "x :x+1]
label "1   make "y :a*cos (:x)/sin (:x)
           if :y>240 [make "x :x+2 if or :x= -270 :x= -90 [make "x :x+1]
           go "1] dot list :x :y dot list -:x -:y cotang :a :x+2
end

to sec :a :b
  make "x -359 make "alfa :x/:b
label "1   if or :x= -90 :x= -180 [make "x :x+1] make "y :a/cos (:b*:alfa)
           if or quotient :b 2 < :b/2 :b=1 [make "y -:y]
           if :y>245 [make "x :x+2 make "alfa :alfa+1 go "1]
           dot list :x :y dot list -:x :y make "x :x+2
           make "alfa :alfa+1 if :x>0 [recycle stop] go "1
end

to cosec :a :b
  make "x -359 make "alfa :x/:b
label "1   if or :x= -90 :x= -270 [make "x :x+1] make "y :a/sin (:b*:alfa)
           if :y>250 [make "x :x+2 make "alfa :alfa+1 go "1]
           dot list :x :y dot list -:x -:y make "x :x+2
           make "alfa :alfa+1 if :x>0 [recycle stop] go "1
end

```

Módulo "curvab"

El listado de los procedimientos 'tri', 'pit', 'c', 'div' y 'ctl' son iguales que en el módulo anterior y el procedimiento 'ejes' sólo difiere en la primera instrucción, que en este caso es (er "curvab) en lugar de (er "curvaa).

```
to a
  ct fs load "curvabb curvab
end
```

```
to curvab
  recuadro curva1 curva2
end
```

```
to recuadro
  setsplit 1 ht ts ct pit 1 pu home setpos [-100 200] pd
  ctl 185 40 0 pu home setcursor [34 2]
  pr [*DIBUJO DE CURVAS*] setpos [-110 195] pd ctl 203 50 10
  pu home setpos [-300 -185] pd ctl 595 360 0 pu home
  setpos [-305 -190] pd ctl 605 370 10 pu home
  setpos [-305 -240] pd ctl 605 40 10
end
```

```
to pby :a :b :c
  make "x -300
label "1
  make "y :a*:x*:x+:b*:x+:c
  if :y>250 [make "x :x+2 go "1]
  if :x>0 [recycle stop]
  dot list :x :y dot list -:x :y
  make "x :x+2 go "1
end
```

```
to pbx :a :b :c
```

```
    Igual que 'pby' excepto que en lugar de  
    (dot :x :y dot -:x :y) se pone  
    (dot :y :x dot :y -:x)
```

```
end
```

```
to cir :r :x :y
```

```
    make "a 0
```

```
label "1
```

```
    make "y1 :r*sin :a make "x1 :r*cos :a  
    dot list :x+:x1 :y+:y1 dot list :x-:x1 :y+:y1  
    dit list :x-:x1 :y-:y1 dot list :x+:x1 :y-:y1  
    if :a>90 [recycle stop] make "a :a+2 go "1
```

```
end
```

```
to sa :a :b :x
```

```
    if :x=0 [make "x 0.01] if :x= -180 [recycle]  
    make "A (sin :b*:x)/(b*:x) make "y 58*:a*:A  
    dot list :x :y dot list -:x :y  
    if :x>0 [recycle stop] sa :a :b :x+4
```

```
end
```

```
to oc :y :f :y1
```

```
    make "x 325/:f pu home setx -325 sety :y1 pd ht  
    repeat :f[fd :y rt 90 fd :x rt 90 fd 2*:y lt 90 fd :x lt 90 fd :y]
```

```
end
```

```
to ot :y :f :y1
```

```
    make "alfa arctan (2*:f*:y/325) make "a :y/sin :alfa  
    pu home setx -325 sety :y1 pd  
    repeat :f[rt 90-:alfa fd :a rt 2*:alfa fd 2*:a lt 2*:alfa fd :a  
    lt 90-:alfa]
```

```
end
```

```

to curval
  setcursor [16 6]
  pr[PARABOLA: LA ECUACION GENERAL ES (Y= A * X2 + B * X + C)]
  setcursor [12 7]
  pr[PARA DIBUJAR CUALQUIER PARABOLA SE ESCRIBE pby Y TRES ENTRADAS, LAS]
  setcursor [9 8]
  pr[ENTRADAS SON LOS TERMINOS (A-B y C). PARA INVERTIR LOS EJES PONER pbx]
  setcursor [17 9]
  pr[ EJEMPLO: ESCRIBA ..pby 0.01 0.5 -60.. Y PULSE RETURN)]
  setcursor [CIRCUNFERENCIA: LA ECUACION GENERAL ES (X2 + Y2 = R2)]
  setcursor [12 12]
  pr[PARA DIBUJAR CUALQUIER CIRCUNFERENCIA SE ESCRIBE cir Y TRES ENTRADAS]
  setcursor [15 13]
  pr [LAS ENTRADAS SON EL RADIO Y LAS COORDENADAS (X-Y) DEL CENTRO.]
  setcursor [18 14]
  pr[ EJEMPLO: ESCRIBA..cir 80 -50 80.. Y PULSE RETURN)]
end

```

```

to curva2
  setcursor [9 16]
  pr[DIAGRAMA ESPECTRAL SENO(X)/(X): SU ECUACION ES Y = SENO(X)/(X)]
  setcursor [12 17]
  pr[PARA REPRESENTAR ESTA CURVA SE ESCRIBE sa Y TRES ENTRADAS, LAS]
  setcursor [12 18]
  pr[ENTRADAS SON: AMPLITUD MAXIMA, FCIA. Y ANGULO. (entre -360 y 0)]
  setcursor [21 19]
  pr[ EJEMPLO: ESCRIBA..sa 150 5 -200 Y PULSE RETURN)]
  setcursor [11 21]
  pr[ONDA CUADRADA: ESCRIBA oc Y TRES ENTRADAS, (AMPLITUD, FCIA. Y OFFSET)]
  setcursor [23 22]
  pr[ EJEMPLO: ESCRIBA oc 100 5 60 Y PULSE RTN.))]
  setcursor [10 24]
  pr[ONDA TRIANGULAR: ESCRIBA ot Y TRES ENTRADAS, (AMPLITUD FCIA. Y OFFSET)]
  setcursor [23 25]
  pr[ EJEMPLO: ESCRIBA ot 150 7 -50 Y PULSE RTN.))]
  setcursor [17 28]
  pr[*PARA VOLVER AL MENU ANTERIOR ESCRIBA c Y PULSE RTN. *]
  pit 2 setcursor [0 30]
end

```


Módulo "espacio"

```
to espacio
  ventana setcursor [21 4]
  pr [DIBUJO Y CALCULO DE FIGURAS TRIDIMENSIONALES] setcursor [19 9]
  pr [CON ESTE MENU PUEDE ACCEDER AL CALCULO Y DIBUJO] setcursor [19 11]
  pr [DE LAS FIGURAS QUE SE RELACIONAN A CONTINUACION] setcursor [6 14]
  pr [1) PRISMA.....] setcursor [50 14]
  pr [ESCRIBA prisma Y PULSE RETURN.] setcursor [6 16]
  pr [2) PIRAMIDE.....] setcursor [50 16]
  pr [ESCRIBA piramide Y PULSE RETURN.] setcursor [ 6 18]
  pr [3) CUBO.....] setcursor [50 18]
  pr [ESCRIBA cubo Y PULSE RETURN.] setcursor [6 20]
  pr [3) OCTAEDRO.....] setcursor [50 20]
  pr [ESCRIBA octaedro Y PULSE RETURN.] setcursor [11 27]
  pr [*SI QUIERE VOLVER AL MENU ANTERIOR ESCRIBA bd Y PULSE RETURN*]
  setcursor [11 28] pit 2
end

to ventana
  setsplit 2 ht ts cs pu home pit 1 setpos [-210 165]
  pd ctl 385 45 10 pu home setpos [-205 170] pd
  ctl 375 35 0 pu home setpos [-340 -160] pd
  ctl 650 300 10 pu home setpos [-335 -155] pd
  ctl 640 290 0 pu home setpos [-340 -225] pd ctl 650 45 10
end

to prisma
  erall ct fs load "prisma prisma
end

to piramide
  erall ct fs load "piram piram
end

to cubo
  erall ct fs load "cubo cubo
end
```

```
to octaedro
    erall ct fs load "oct oct
end
```

```
to bd
    ct fs load "bdatos bd
end
```

```
to pit :n
    repeat :n[type char 7]
end
```

```
to ctl :x :y :z
    repeat 2 [fd :y rt 90 fd :x rt 90] fd :y rt 65 fd :z rt 25
    fd :x rt 90 fd :y rt 65 fd :z rt 115 fd :y rt 65 fd :z
end
```

Módulo "prisma"

```
to prisma
    recuadro setcursor [10 8]
    pr[PARA DIBUJAR CUALQUIER PRISMA VISTO DESDE CUALQUIER ANGULO TIENE QUE]
    setcursor [10 10]
    pr[ESCRIBIR pri SEGUIDO DE CUATRO ENTRADAS, LAS CUATRO ENTRADAS SON LAS]
    setcursor [19 12]
    pr[MEDIDAS DE LOS TRES LADOS Y DEL ANGULO DE VISION.] setcursor [8 16]
    pr[EJEMPLO: ESCRIBA pri 50 200 40 20 Y PULSE RTN. EL PROGRAMA DIBUJA UN]
    setcursor [10 18]
    pr[PRISMA DE 50 DE BASE 200 DE ALTURA 40 DE ANCHO Y VISTO DESDE UN ANGULO
    DE 20 9] setcursor [17 21]
    pr[(PRUEBE EL EJEMPLO cs..pri 50 50 40 20..Y PULSE RTN.)] setcursor [16 26]
    pr[*PARA EL CALCULO DEL PRISMA ESCRIBA cp Y PULSE RTN.*] setcursor [15 28]
    pr[*PARA VOLVER AL MENU ANTERIOR ESCRIBA sp Y PULSE RTN.*]
    setcursor [0 30] pit 2
end
```

```
to sp
    erall ct fs load "espacio espacio
end
```

to recuadro

```
ct ts pit 1 setsplit 1 ht pu home setpos [-155 180] pd
ctl 295 53 10 pu home setpos [-150 185] pd ctl 285 43 0
pu home setpos [-330 -157] pd ctl 660 320 10 pu home
setpos [-330 -240] pd ctl 660 70 10 setcursor [28 3]
pr[DIBUJO Y CALCULO DEL PRISMA RECTO]
```

end

to ctl :x :y :z

```
repeat 2[fd :y rt 90 fd :x rt 90] fd :y rt 65 fd :z rt 25
fd :x rt 90 fd :y rt 65 fd :z rt 115 fd :y rt 65 fd :z
```

end

to pris :x :y :z :ag

```
make "z :z*sin :ag fd :y repeat 2[fd :y rt 90 fd :x rt 90]
fd :y rt 90-:ag fd :z rt :ag fd :x rt 90 fd :y rt 90-:ag
fd :z rt 90+:ag fd :y rt 90-:ag fd :z rt 90+:ag fd :y
rt 90 fd :x rt 90 fd :y bk :y lt 90+:ag fd :z rt 90+:ag
```

end

to pit :n

```
repeat :n[type char 7]
```

end

to cp

```
erall ct fs load "cprisma cprisma
```

end

to pri :x :y :z :ag

```
pris :x :y :z :ag ct setcursor [10 28]
```

```
pr[SI QUIERE EL DIBUJO SIN LEYENDAS AL PIE -ESCRIBA pris EN VEZ DE pri-]
```

```
setcursor [15 29]
```

```
pr[PARA VOLVER AL MENU ANTERIOR ESCRIBA prisma Y PULSE RTN.]
```

end

Módulo "cprisma"

```
to cprisma
  ts ct pit 1 setsplit 3 ht pu home setpos [-120 184] pd ctl 220 45 10 20
  pu home setpos [-115 189] pd ctl 210 35 0 0 pu home setpos [-305 -130]
  pd ctl 590 295 10 20 pu home setpos [-305 -190] pd ctl 590 40 10 20
  setcursor [32 3 ] pr[*CALCULO DEL PRISMA RECTO] setcursor [10 8]
  pr[PARA HACER CALCULOS SOBRE UN PRISMA RECTO ESCRIBA cp TRES ENTRADAS.]
  setcursor [11 10]
  pr [LAS TRES ENTRADAS SON LAS MEDIDAS DE LA BASE LA ALTURA Y EL ANCHO.]
  setcursor [10 13]
  pr[AL ESCRIBIR cp (TRES ENTRADAS) Y PULSAR RTN. EL PROGRAMA DIBUJA EL]
  setcursor [10 15]
  pr[PRISMA OBJETO DEL CALCULO AL MISMO TIEMPO QUE PRESENTA EN PANTALLA]
  setcursor [10 17]
  pr[LOS VALORES DE (SUPERFICIE LATERAL Y TOTAL, VOLUMEN Y DIAGONALES).]
  setcursor [17 20]
  pr[ (PRUEBE EL EJEMPLO..cp 100 200 100..YPULSE RTN.)] setcursor [13 25]
  pr[*SI QUIERE VOLVER AL MENU ANTERIOR ESCRIBA pri Y PULSE RTN.*]
  pit 2 setcursor [0 30]
end
```

```
to ventana
  cs ht setsplit 1 pu setpos [-360 -180] pd ctl 242 435 10 30
  pu home setpos [-95 -180] pd ctl 440 435 10 30] pu home
  setpos [-90 -175] pd ctl 430 425 0 0 pu home setpos [-360 -245]
  pd ctl 705 50 10 30 pu home setpos [-355 -240] pd ctl 695 40 0 0
  pu home setpos [-330 -110] pd prisma 120 180 60 -25
  setcursor [4 10] pr[.....ANCHO.....] setcursor [21 11]
  pr [LARGO] setcursor [27 13] pr[:] setcursor [27 14] pr[:]
  setcursor [27 15] pr[:] setcursor [27 16] pr[A]
  setcursor [27 17] pr[L] setcursor [27 18] pr[T]
  setcursor [27 19] pr [O] setcursor [27 20] pr [:]
  setcursor [27 21] pr [:] setcursor [27 22] pr [:]
  setcursor [12 17] pr [DIAGONAL] setcursor [13 22]
  pr [D.BASE] setcursor [6 24] pr [PRISMA GENERICO]
end
```

```
to pit :n
  repeat :n type char 7]
end
```

```

to ctl :x :y :z :ag
  repeat 2[fd :y rt 90 fd :x rt 90] fd :y rt 90-:ag fd :z rt :ag
  fd :x rt 90 fd :y rt 90-:ag fd :z rt 90+:ag fd :y rt 90-:ag fd :z
end

```

```

to prisma :x :y :z :ag
  vari dib ern [m D ALTURA altura gamma a tita beta]
end

```

```

to vari
  make "a :z*cos :ag make "altura :z*sin :ag
  make "beta arctan :altura/(:x-:a)
  make "gamma 180-:ag-:beta make "m :altura/sin :beta
  make "tita arctan (:altura+:y)/(:x+:a)
  make "D (:altura+:y)/sin :tita make "ALTURA :y
end

```

```

to dib
  ctl :x :y :z :ag rt 90+:ag fd :y rt 90 fd :x rt 90 fd :y
  bk :y lt 90+:ag fd :z rt 180 fd :z rt 180-:gamma fd :m
  rt :gamma+:ag fd :x rt 180-:tita fd :D
end

```

```

to pri
  erall ct fs load "prisma prisma
end

```

```

to raya :a
  repeat 10 [pu fd :a/20 pd fd :a/20]
end

```

```

to xyz
  setcursor [38 23] pr [PRISMA OBJETO DEL CALCULO]
  setcursor [67 22] pr [ANCHO =] setcursor [75 22] pr :x
  setcursor [67 23] pr [ALTO =] setcursor [75 23] pr :y
  setcursor [67 24] pr [LARGO =] setcursor [75 24] pr :z/sin 25
end

```

```

to cp :x :y :z
  ts pit 1 make "ag 25 variabler make "z :z*sin 25
  ventana variabled xyz medidas dibujo
  ern [m ag al all D.MA D.ME C M dm beta beta1 tita
  a gamma altura d.p. β] setcursor [0 1] pons
  recycle pit 2 setcursor [0 30]
end

```

```

to medidas
  make "M 300 make "X (:x+:a) make "Y (:y+:altura)
  make "dm :D.MA if :Y>:X [make "C :M/:Y go "1] make "C :M/:X
label "1 make "x :x*:C make "y :y*:C make "dm :dm*:C make "z :z*:C
  make "m :m*:C make "X :X*:C make "Y :Y*:C
end

```

```

to variabler
  make "al arctan :z/:x
  make "all arctan (:y*cos :al)/:x
  make "D.BASE :x/cos :al
  make "DIAGONAL :D.BASE/cos :all
  make "AREA.BASE :x*:z make "AREA.L (:x+:y)*4
  make "AREA.T :AREA.L+2*:AREA.BASE
  make "ALTURA :y
  make "VOLUMEN :AREA.BASE*:ALTURA
end

```

```

to variabled
  make "a :z*cos :ag make "altura :z*sin :ag
  make "beta1 arctan :altura/(:x+:a)
  make "beta arctan :altura/(:x-:a)
  make "gamma 180-:ag-:beta
  make "m :altura/sin :beta
  make "tita arctan (:altura+:y)/(:x+:a)
  make "d.p. :altura/(sin (arctan (:altura/(:x+:a))))
  make "β arctan :y/:d.p.
  make "D.ME :y/sin :β
  make "D.MA (:altura+:y)/sin :tita
end

```

to dibujo

```
pu home setx 130-:X/2 sety 60-:Y/2 pd ern [X Y]
repeat 2[fd :y rt 90 fd :x rt 90] fd :y rt 65 fd :z rt :ag
fd :x rt 90 fd :y rt 65 fd :z rt 115 fd :y rt 65 fd :z
fd 115 raya :y rt 90 raya :x rt 90 raya :y pu bk :y
pd lt 115 raya :z rt 180 pu fd :z pd rt 180-:gamma
raya :m rt :gamma+:ag fd :x rt 180-:tita raya :dm
ct setcursor [13 28]
pr[*PARA VOLVER AL MENU ANTERIOR ESCRIBA oprisma Y PULSE RTN.*]
```

end

Módulo "piram"

```
to sp
  erall ct fs
  load "espacio.espacio"
end
```

```
to ctl :x :y :z
  repeat 2[fd :y rt 90 fd :x rt 90] fd :y
  rt 65 fd :z rt 25 fd :x rt 90 fd :y rt 65
  fd :z rt 115 fd :y rt 65 fd :z rt 115
end
```

```
to pir :x :y :ag
  setsplit 2
  make "y :y*cos :ag make "z :x*sin :ag make "b :z*sin :ag
  make "a :z*cos :ag make "f (2*:y)/(x+:a) make "e arctan :f
  make "d :y/sin :e rt 90-:ag fd :z rt :ag fd :x
  rt 180-:ag fd :z rt :ag fd :x rt 180-:e fd :d make "f :y-:b
  make "g arctan (2*:f)/(x+:a) make "h (x+:a)/(2*cos :g)
  rt :e+:g fd :h make "j arctan ((x-:a)/(2*:y)) make "k :y/cos :j
  pu bk :h pd rt 90-:g-:j fd :k make "m :y-:b
  make "n arctan (x-:a)/(2*:m) make "o (x-:a)/(2*sin :n) pu bk :k
  pd rt :j+:n fd :o make "q 90-:n-:ag rt :q fd :z rt 90+:ag
  recycle ct setcursor [10 29]
  pr [SI QUIERE VOLVER AL MENU ANTERIOR ESCRIBA piram Y PULSE RETURN]
end
```

```
to cpi
  erall ct fs
  load "cpiram cp"
end
```



```

to piram
ct setsplit 2 ht pu home setpos [-150 200] pd ctl 300 40 0
pu home setpos [-155 195] pd ctl 310 50 10
pu home setpos [-300 -100] pd ctl 610 275 10
pu home setpos [-300 210] pd ctl 610 80 10 setcursor [31 2]
pr[DIBUJO Y CALCULO DE PIRAMIDES] setcursor [25 6]
pr[CON ESTE MODULO PODRA DIBUJAR CUALQUIER] setcursor [25 7]
pr[PIRAMIDE VISTA DESDE CUALQUIER ANGULO] setcursor [27 9]
pr[EJEMPLO: SI ESCRIBE pir 150 200 20] setcursor [10 11]
pr[EL PROGREMA DIBUJARA UNA PIRAMIDE CUADRANGULAR DE 150 DE BASE y 200 DE]
setcursor [10 12]
pr[ALTURA VISTA DESDE UN ANGULO DE 20º POR LA DERECHA DE LA PANTALLA, SI]
setcursor [10 13]
pr[PONE -20º LA VERA POR LA IZQUIERDA, ASIMISMO LA ALTURA DISMINUIRA DE]
setcursor [10 14]
pr[FORMA PROPORCIONAL AL AUMENTO DEL ANGULO DE VISION.] setcursor [26 17]
pr[PRUEBE EL EJEMPLO: cs pir 150 200 20] setcursor [13 19]
pr[(cs DELANTE DE pir BORRA LA PANTALLA ANTES DE EMPEZAR A DIBUJAR)]
setcursor [22 26]
pr[*PARA EL CALCULO ESCRIBA cpi Y PULSE RETURN*] setcursor [0 29]
end

```

Módulo "cpiram"

```

to cp
ct cs ht setsplit 2 pu home setpos [-125 200] pd ctl 235 40 0
pu home setpos [-130 195] pd ctl 245 50 10
pu home setpos [-305 -120] pd ctl 615 295 10
pu home setpos [-305 -180] pd ctl 615 50 10
setcursor [32 2]
pr[*CALCULO DE PIRAMIDES*]
setcursor [8 7]
pr[CON ESTE MODULO PODRA HACER CALCULOS SOBRE CUAQUIER PIRAMIDE
CUADRANGULAR] setcursor [27 10]
pr[EJEMPLO: SI ESCRIBE cpir 150 200]
setcursor [10 12]
pr[*EL PROGRAMA HARA LOS CALCULOS PARA OBTENER EL VOLUMEN LA SUPERFICIE Y]
setcursor [12 14]
pr[TODAS LAS VARIABLES DE UNA PIRAMIDE DE 150 DE BASE Y 200 DE ALTURA.]
setcursor [10 16]
pr[* AL MISMO TIEMPO, EL PROGRAMA DIBUJARA LA PIRAMIDE Y MOSTRARA EN LA]
setcursor [12 18]
pr[PANTALLA EL RESULTADO DE LOS CALCULOS REALIZADOS.]
setcursor [27 20]
pr[PRUEBE EL EJEMPLO: cpir 150 200] setcursor [16 24]
pr[* PARA VOLVER AL MENU ANTERIOR ESCRIBA dpir Y PULSE RETURN *]
setcursor [0 29] pit 2
end

```

```

to cpir :x :y
  ventana pit 1
  variabled medidas
  dibujo pit 1 variabler
  ern [C c1 m D z k j h e n o p q a b c d g]
  setcursor [0 1] pons setcursor [12 28]
  pr[SI QUIERE VOLVER AL MENU ANTERIOR ESCRIBA cp Y PULSE RETURN]
  setcursor [0 30]
  recycle pit 2
end

```

```

to ventana
  cs ht setcursor [0 30] setsplit 1
  pu home setpos [-360 -185] pd ctl 245 440 10
  pu home setpos [-95 -185] pd ctl 440 440 10
  pu home setpos [-90 -180] pd ctl 430 430 0
  pu home setpos [-360 -240] pd ctl 705 40 10
  pu home setpos [-355 -235] pd ctl 695 30 0
  pu home setpos [-354 -115] pd pir 168 200
  setcursor [14 14] pr[A]
  setcursor [14 15] pr[L]
  setcursor [14 16] pr[T]
  setcursor [14 17] pr[U]
  setcursor [14 18] pr[R]
  setcursor [14 19] pr[A]
  setcursor [8 21] pr[D.BASE]
  setcursor [6 23] pr[LADO.BASE]
  setcursor [6 25] pr[PIRAMIDE GENERICA]
  setcursor [37 23] pr[PIRAMIDE OBJETO DEL CALCULO]
  setcursor [66 22] pr[LADO.B =] setcursor [75 22] pr :x
  setcursor [66 24] pr[ALTURA =] setcursor [75 24] pr :y
end

```

```

to ctl :x :y :z
  repeat 2[fd :y rt 90 fd :x rt 90] fd :y
  rt 65 fd :z rt 25 fd :x rt 90 fd :y
  rt 65 fd :z rt 115 fd :y rt 65 fd :z
end

```

```

to variabled
  make "y :y*cos 25 make "z :z*sin 25
  make "b :z*sin 25 make "a :z*cos 25
  make "c (:x+:a)/2 make "e arctan :y/:c
  make "d :y/sin :e make "g arctan (:y-:b)/:c
  make "h :c/cos :g
  make "j arctan (:x-:a)/(2*:y)
  make "k :y/cos :j make "m :y-:b
  make "n arctan (:x-:a)/2*:m)
  make "o (:x-:a)/(2*sin :n)
  make "p 90-:n make "q :p-25
  make "D :b/(sin arctan :b/(:x-:a))

```

end

```

to pit :n
  repeat :n[ type char 7]

```

end

```

to pir :x :y
  variabled
  rt 65 fd :z rt 25 fd :x rt 155 fd :z
  rt 25+arctan :b/(:x-:a) fd :D bk :D
  lt arctan :b/(:x-:a) fd :x rt 180-:e
  fd :d rt :e+:g fd :h bk :h rt 90-:g-:j
  fd :k bk :k rt :j fd :y-(:b/2)
  bk :y-(:b/2) rt :n fd :o rt :q

```

end

```

to medidas
  if :x>:y [make "C 280/:x go "10] make "C 280/:y
  label "10 make "z :z*:C make "x :x*:C make "d :d*:c
  make "h :h*:C make "k :k*:C make "o :o*:C

```

end

```

to dibujo
  make "c1 120-(:x+:z*cos 25)/2 pu home
  setx :c1 sety -75 pd rt 65 raya :z rt 25
  raya :x rt 155 fd :z rt 25 fd :x rt 180-:e
  fd :d rt :e+:g fd :h pu bk :h pd rt 90-:g-:j
  fd :k pu bk :k pd rt :j+:n raya :o rt :q

```

end

```
to raya :a
  repeat 10[pu fd(:a/20) pd fd (:a/20)]
end
```

```
to dpir
  ct fs erall
  load "piram piram
end
```

```
to variabler
  make "x :x/:C
  make "y :y/cos 25
  make "DIAG.BAS :x/ cos 45
  make "ARISTA.L :y/sin arctan ((2*:y)/:DIAG.BAS)
  make "ALTURA.P :y
  make "AREA.BAS :x*:x
  make "ALTURA.T :y/sin arctan ((2*:y)/:x)
  make "AREA.LAT 2*:ALTURA.T*:x
  make "AREA.TOT :AREA.BAS+:AREA.LAT
  make "LADO.BAS :x
  make "VOLUME.P (:AREA.BAS*:ALTURA.P)/3
end
```

Módulo "cubo"

```
to sp
  erall ct fs load "espacio espacio"
end

to ctl :x :y :z
  repeat 2[fd :y rt 90 fd :x rt 90] fd :y
  rt 65 fd :z rt 25 fd :x rt 90 fd :y
  rt 65 fd :z rt 115 fd :y rt 65 fd :z
end

to pit :n
  repeat :n[type char 7]
end

to recuadro
  cs ts pit 1 setsplit 1 ht pu home setpos [-140 180] pd
  ctl 250 53 10 pu home setpos [-135 185] pd ctl 240 43 0
  pu home setpos [-310 -157] pd ctl 595 320 10 pu home
  setpos [-310 -240] pd ctl 595 70 10 setcursor [31 3]
  pr [DIBUJO Y CALCULO DEL CUBO]
end

to cu :x :ag
  cus :x :ag ct setcursor [9 28]
  pr[SI QUIERE EL DIBUJO SIN LEYENDAS AL PIE ESCRIBA cus EN VEZ DE cu]
  setcursor [14 29]
  pr[PARA VOLVER AL MENU ANTERIOR ESCRIBA cubo Y PULSE RTN.]
end

to cc
  erall ct fs load "ccubo ccubo"
end
```

```

to cus :x :ag
    make "z :x*sin :ag
    repeat 4[fd :x rt 90] fd :x rt 90-:ag fd :z rt :ag
    fd :x rt 90 fd :x rt 90-:ag fd :z rt 90+:ag fd :x
    rt 90-:ag fd :z rt 90+:ag repeat 2[fd :x rt 90]
    fd :x bk :x lt 90+:ag fd :z rt 90+:ag
end

```

```

to cubo
    recuadro setcursor [10 8]
    pr [PARA DIBUJAR CUALQUIER CUBO VISTO DESDE CUALQUIER ANGULO TIENE QUE]
    setcursor [12 10]
    pr [ESCRIBIR cu SEGUIDO DE DOS ENTRADAS. LAS DOS ENTRADAS SON LAS]
    setcursor [22 12]
    pr [MEDIDAS DEL LADO Y DEL ANGULO DE VISION.] setcursor [10 16]
    pr [EJEMPLO: ESCRIBA cu 100 30 Y PULSE RTN. EL PROGRAMA DIBUJA UN CUBO]
    setcursor [20 18]
    pr [DE LADO = 100 Y VISTO DESDE UN ANGULO DE 30 °.]
    setcursor [19 21]
    pr [(PRUEBE EL EJEMPLO es...cu 120 30.. Y PULSE RTN.))
    setcursor [17 26]
    pr [*PARA EL CALCULO DEL CUBO ESCRIBA cc Y PULSE RTN.*]
    setcursor [15 28]
    pr [*PARA VOLVER AL MENU ANTERIOR ESCRIBA sp Y PULSE RTN.*]
    setcursor [0 30] pit 2
end

```

Módulo "cubo"

Los procedimientos 'variabler', 'medidas', 'xyz', 'variabled', 'ctl', 'pit', 'raya' y 'dibujo' son los mismos que tiene el módulo "cprisma".

```

to vari
    make "a :z*cos :ag make "altura :z*sin :ag
    make "beta arctan :altura/(:x-:a)
    make "gamma 180-:ag-:beta make "m :altura/sin :beta
    make "tita arctan (:altura+:y)/(:x+:a)
    make "D (:altura+:y)/sin :tita make "ALTURA :y
end

```

```

to ccubo
  ts ct pit 1 setsplit 3 ht pu home setpos [-110 184] pd
  ctl 190 45 0 20 pu home setpos [-105 189] pd ctl 180 35 0 0
  pu home setpos [-305 -130] pd ctl 590 295 10 20 pu home
  setpos [-305 -190] pd ctl 590 40 10 20 setcursor [35 3]
  pr [CALCULO DEL CUBO] setcursor [14 8]
  pr [PARA HACER CALCULOS SOBRE UN CUBO ESCRIBA cc Y UNA ENTRADA.]
  setcursor [23 10]
  pr [LA ENTRADA ES LA MEEDIDA DE LA ARISTA.] setcursor [12 13]
  pr [AL ESCRIBIR cp (UNA ENTRADA) Y PULSAR RTN. EL PROGRAMA DIBUJA
  EL] setcursor [12 15]
  pr [CUBO OBJETO DEL CALCULO AL MISMO TIEMPO QUE PRESENTA EN]
  PANTALLA] setcursor [12 17]
  pr [LOS VALORES DE (SUPERFICIE LATERAL Y TOTAL, VOLUMEN Y DIAG.)]
  setcursor [20 20]
  pr [(PRUEBE EL EJEMPLO..cc 100.. Y PULSE RTN.)]
  setcursor [13 25]
  pr [*SI QUIERE VOLVER AL MENU ANTERIOR ESCRIBA cubo Y PULSE RTN.*]
  pit 2 setcursor [0 30]
end

```

```

to ventana
  cs ht setsplit 1 pit 1
  pu home setpos [-360 -180] pd ctl 242 435 10 30
  pu home setpos [-95 -180] pd ctl 440 435 10 30
  pu home setpos [-90 -175] pd ctl 430 425 0 0
  pu home setpos [-360 -240] pd ctl 705 35 10 30
  pu home setpos [-355 -240] pd ctl 695 40 0 0
  pu home setpos [-330 -110] pd ccv 120 120 60 -25
  setcursor [4 13] pr [.....ANCHO.....] setcursor [21 14]
  pr [LARGO] setcursor [27 18] pr [A] setcursor [27 19]
  pr [L] setcursor [27 20] pr [T] setcursor [27 21]
  pr [O] setcursor [12 19] pr [DIAGONAL] setcursor [13 22]
  pr [D.BASE] setcursor [8 24] pr [(CUBO GENERICO)]
end

```

```

to cubo
  erall ct fs load "cubo cubo
end

```

```

to cc :x
  ts pit 1 make "ag 25 make "y :x make "z :x variabler
  make "z :x*sin 25
  ventana variabler xyz medidas dibujo
  ern[m z y ag all D.MA D.ME C M  $\beta$  beta beta1 tita a gamma altura d.p.]
  setcursor [0 1] pons recycle pit 2 setcursor [0 30]
end

```

```

to ccv :x :y :z :ag
  vari dib ern [m D ALTURA altura gamma a tita beta]
end

```

```

to did
  ctl :x :y :z :ag rt 90+:ag fd :y rt 90 fd :x rt 90 fd :y
  bk :y lt 90+:ag fd :z rt 180 fd :z rt 180-:gamma fd :m
  rt :gamma+:ag fd :x rt 180-:tita fd :D
end

```


Módulo "octa"

```
to sp
  erall ct fs load "espacio espacio"
end

to ctl :x :y :z
  repeat 2[fd :y rt 90 fd :x rt 90] fd :y
  rt 65 fd :z rt 25 fd :x rt 90 fd :y
  rt 65 fd :z rt 115 fd :y rt 65 fd :z
end

to pit :n
  repeat :n[type char 7]
end

to ot :x :y :angulo
  ots :x :y :angulo
  setcursor [10 27]
  pr [SI QUIERE EL DIBUJO SIN LEYENDAS AL PIE ESCRIBA ots EN VEZ DE ot]
  setcursor [14 28]
  pr [PARA VOLVER AL MENU ENTERIOR ESCRIBA octa Y PULSE RETURN]
end

to ots :x :y :angulo
  pd pir :x :y :angulo
  rt 90-:angulo fd :x*sín :angulo rt :angulo fd :x rt 90
  pd pir :x :y :angulo
end

to oct
  erall ct fs load "cocta cocta"
end
```

```

to pir :x :y :angulo
                                (El listado de este procedimiento es el mismo
                                que el listado de 'pir' en el módulo "piram").
end

```

```

to octa
setsplit 1 ts ct pit 1 cs ht setcursor [29 3]
pr [DIBUJO Y CÁLCULO DEL OCTAEDRO] pu home setpos [-150 180]
pd ctl 270 50 10 pu home setpos [-145 185] pd
ctl 260 40 0 pu home setpos [180 210] pd ots 120 50 20
pu home setpos [-300 215] pd ots 50 50 -30 pu home
setpos [-330 -160] pd ctl 655 310 10 pu home setpos [-330 -240]
pd ctl 655 68 10 pu home setpos [-325 -155] pd ctl 645 300 0
setcursor [10 10]
pr [PARA EL DIBUJO DE UN OCTAEDRO ESCRIBA ot Y TRES ENTRADAS. LAS TRES]
setcursor [9 12]
pr [ENTRADAS SON LAS MEDIDAS DE LA BASE, ALTURA Y ANGULO DE VISION DE UNA]
setcursor [9 14]
pr [DE LAS PIRAMIDES QUE LO FORMAN. AL SER LA FIGURA SIMETRICA, BASTA CON]
setcursor [14 16]
pr [DAR LAS MEDIDAS DE LA PIRAMIDE PARA EL DIBUJO DEL OCTAEDRO.]
setcursor [18 19]
pr [EJEMPLO: ESCRIBA cs. ot120 100 25 Y PULSE RTN.] setcursor [17 26]
pr [* PARA EL MENU ANTERIOR ESCRIBA sp Y PULSE RTN.*]
setcursor [19 28]
pr [* PARA EL CALCULO ESCRIBA oct Y PULSE RTN.*]
pit 2 setcursor [0 30]
end

```

Módulo "cocta"

```

to ventana
cs ht setsplit 1 pit 1
pu home setpos [-360 -185] pd ctl 245 440 10
pu home setpos [-95 -185] pd ctl 440 440 10
pu home setpos [-360 -240] pd ctl 705 43 10
pu home setpos [-295 30] pd ot 160 100 -20
setcursor [6 24] pr [OCTAEDRO GENERICO]
setcursor [20 13] pr [ARISTA] setcursor [14 15]
cr [DIAG.] setcursor [0 30]
end

```

```

to pit :n
    repeat :n[ type char 7]
end

```

```

to ctl :x :y :z
    repeat 2[fd :y rt 90 fd :x rt 90] fd :y
    rt 65 fd :z rt 25 fd :x rt 90 fd :y
    rt 65 fd :z rt 115 fd :y rt 65 fd :z
end

```

```

to pir :x :y :angulo
    (El listado es el mismo
     que en el módulo "piram".)
end

```

```

to oct
    erall ct fs load "octa octa
end

```

```

to cocta
    ts ct setsplit 1 pit 1 cs ht setcursor [32 3]
    pr [*CALCULO DEL OCTAEDRO*] pu home setpos [-120 180]
    pd ctl 225 50 10 pu home setpos [-115 185] pd
    ctl 215 40 0 pu home setpos [150 200] pd ot 150 60 20
    pu home setpos [-270 200] pd ot 90 70 -30 pu home
    setpos [-330 -140] pd ctl 655 290 10 pu home setpos [-325 -135]
    pd ctl 645 280 0 setcursor [12 12]
    pr [PARA EL CALCULO DE UN OCTAEDRO REGULAR ESCRIBA coc Y UNA ENTRADA.]
    setcursor [23 18]
    pr [EJEMPLO: ESCRIBA ...coc 240...Y PULSE RETURN.]
    setcursor [19 26]
    pr [* PARA EL MENU ANTERIOR ESCRIBA oct Y PULSE RTN. *]
    pit 2 setcursor [0 30]
end

```

```

to ot :x :y :angulo
    pd pir :x :y :angulo rt 90-:angulo fd :x*sin :angulo
    rt :angulo fd :x rt 90 pd pir :x :y :angulo
end

```

```

to coc :a
  pit 1 ventana make "x :a make "y :a
  variabler variablen medidas pu home
  setx 120-:X/2 sety 240-:Y/2 pd dibujo1
  pu rt 65 fd :x*sin 25 rt 25 fd :x rt 90 dibujo2
  ern [al M alt X C Y b c d e f g h i j k m n o p q s x z]
  setcursor [13 28]
  pr [*PARA VOLVER AL MENU ANTERIOR ESCRIBA cocta Y PULSE RTN.*]
  setcursor [0 1] pons recycle setcursor [0 30] pit 2
end

```

```

to variabler
  make "APOTEMA.P :a*1.732050808/2
  make "SUPERFICIE 2*:a*:a*1.732050808
  make "VOLUMEN :a*:a*:a*1.414213562/3
  make "DIAGONAL :a*1.414213562
  make "ARISTA :a
end

```

```

to medidas
  make "M 350 make "Y 2*:alt make "X :a+:z*cos 25
  if :Y>:X [make "C :M/:Y] make "C :M/:X
  make "x :a*:C make "z :z*:C make "Y :Y*:C make "X :X*:X
  make "d :d*:C make "h :h*:C make "k :k*:C make "o :o*:C
  make "b :b*:C make "m :m*:C
end

```

```

to variablen
  make "alt :y*cos 25 make "z :x*sin 25 make "b :z*sin 25
  make "al :z*cos 25 make "c (:x+:al)/2 make "f :alt/:c
  make "e arctan :f make "d :alt/sin :e make "s :alt-:b
  make "g arctan :s/:c make "h :c/cos :g
  make "i (:x-:al)/2 make "j arctan :i/:alt
  make "k :alt/cos :j make "m :alt-:b make "n arctan :i/:m
  make "o :i/sin :n make "p 90-:n make "q :p-25
end

```

```

to raya :a
  repeat 10{pu fd :a/20 pd fd :a/20}
end

```

to dibujo1

```
rt 65 raya :z rt 25 raya :x rt 155 fd :z rt 25 fd :x
rt 180-:e fd :d rt :e+90 raya 2*:m+:b pu bk 2*:m+:b
pd lt 90-:g fd :h bk :h rt 90-:g-:j fd :k bk :k
rt :j+:n yara :o rt :q pu fd :z rt 115
```

end

to dibujo2

```
rt 65 fd :z rt 25 fd :x rt 155 fd :z rt 25 fd :x
rt 180-:e pd fd :d rt :e+:g fd :h bk :h rt 90-:g-:j
raya :k pu bk :k pd rt :j+:n fd :o rt :q pu fd :z pd rt 115
```

end

PROGRAMA

DEMOSTRACION

PROGRAMA DEMOSTRACION

II.5.- Programa demostración

Este programa permite hacer una presentación práctica de la base de datos.

Con el módulo "demo", que consta de un procedimiento igualmente llamado 'demo', se hace la demostración práctica de toda la base de datos.

En cada módulo utilizado en este programa demostración se ha incluido un procedimiento. Este procedimiento lleva en su declaración las ordenes para utilizar los procedimientos usados en el programa "demo" y además las instrucciones para conectar con el siguiente módulo.

Una vez conectados con el siguiente módulo y desde el procedimiento del módulo anterior se manda una última orden; ésta es la de ejecutar el procedimiento homólogo.

Todos estos procedimientos de ejecución del programa demostración llevan el nombre 'demon', donde la 'n' es un número que indica el orden de ejecución dentro del programa demostración.

El módulo que inicia la ejecución del programa demostración es "demo". Al cargar este módulo en el espacio de trabajo y ejecutarlo comienza el programa demostración.

II.5.1.- Listado del programa demostración

```
to demo
  pit 20 load "bdatos bd recycle retardo 10
  erall ct fs load "plana demo1
end

to demo1
  plana recycle retardo 20 erall ct fs load "triangulo demo2
end

to demo2
  triangulo recycle retardo 20 erall ct fs load "ctc1 demo3
end
```

```

to demo3
    ctcl retardo 10 recycle dt 12 7 38 retardo 20
    dt 3450 625 120 erall ct fs load "plana demo4
end

to demo4
    plana retardo 20 recycle erall ct fs load "recta demo5
end

to demo5
    recta retardo 10 recycle ejes re 100 1.2 re -100 -0.8
    re 150 0 re -100 1.3 retardo 30 erall ct fs load "curvaa demo6
end

to demo6
    curvaa retardo 10 ejes er "ejes er "eje recycle
    sec 40 2 pit 2 erall ct fs load "curvab demo7
end

to demo7
    curvab retardo 10 recycle ejes pbx -0.02 0.02 0
    pit 1 cir 100 150 0 pit 1 retardo 10
    eje ot 25 10 200 pit 1 oc 15 20 135 pit 1 oc 75 6 0
    pit 1 ot 50 4 -140 pit 1 retardo 10
    erall ct fs load "espacio demo8
end

to demo8
    espacio retardo 10 erall ct fs load "prisma demo9
end

```



```

to demo9
  prisma retardo 10 cs setsplit 2 pu home setpos [-300 -200]
  pd ht ctt 400 10 200 25 fd 10 pris 40 240 30/sin 25 25
  re 65 fd 170 lt 65 pris 40 240 30/sin 25 25 rt 65 bk 170
  rt 25 fd 360 lt 90 pris 40 240 30/sin 25 25 rt 65 pu fd 170
  lt 65 pd pris 40 240 30/sin25 25 pu home setpos [-300 -200] pd
  fd 250 ctt 400 10 200 25 fd 10 pris 40 100 30/sin 25 25 rt 65
  fd 170 lt 65 pris 40 100 30/sin 25 25 25 rt 65 bk 170 rt 25
  fd 360 lt 90 pris 40 100 30/ 25 25 rt 65 pu fd 170 lt 65
  pd pris 40 100 100/sin 25 25 pu setpos [-300 -200] pd fd
  360 ctt 400 10 200 25 retardo 10 erall ct fs
  load "cprisma demo10

```

end

```

to demo10
  er "cprisma recycle cp 1 1 1 retardo 10 cp 3200 6000 4500
  retardo 10 erall ct fs load "cpiram demo11

```

end

```

to demo11
  er "cp ts ct recycle cpir 1 1 pit2 retardo 10
  cpir 1500 3400 3400 pit 20 erall recycle

```

end

DIAGRAMA DEL PROGRAMA DEMO

El programa - DEMO - tarda en ejecutarse 18 minutos. Este programa comienza con la ejecución del módulo "demo". A partir de la ejecución de este módulo, se van interconectando entre sí los diferentes módulos de la base de datos. Esta conexión se logra a través de los procedimientos 'demo' incluidos en los módulos utilizados.

Los módulos que se utilizan en el programa demostración están interconectados como se indica en la figura (II.90.).

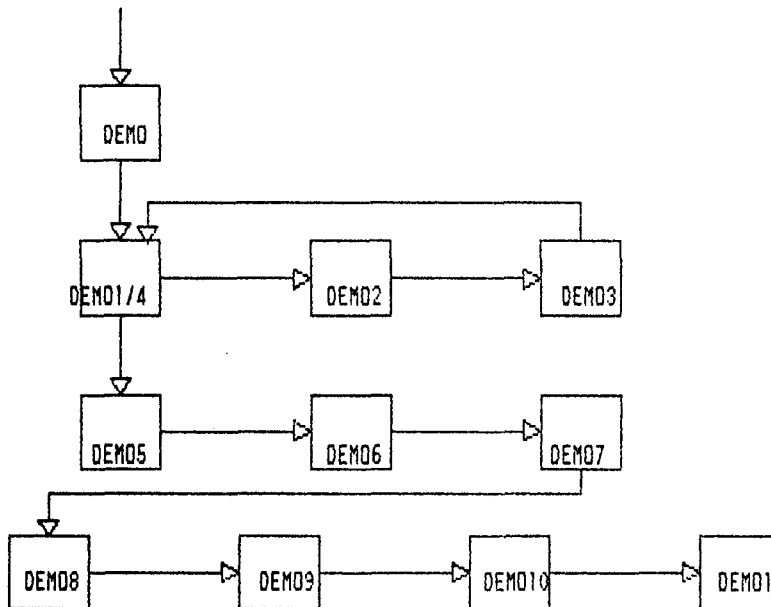


fig. (II.90.)

*MODEM F. S. K. PARA LA
RED TELEFONICA CONMUTADA*

INTRODUCCION

Debido a que la red Telefónica conmutada está pensada para la transmisión de señales analógicas, la información digital que se transmita a través de esta red está expuesta a perturbaciones que introducirían muchos errores. La mayoría de las centrales telefónicas de conmutación utilizan para el proceso de la información recibida del abonado elementos mecánicos (combinadores, buscadores, selectores, relés etc.), que introducirían una gran cantidad de errores en una transmisión digital. Por lo tanto la transmisión de información digital a través de esta red ha de hacerse utilizando un modem.

El tipo de modulación que utiliza este modem es de F.S.K. (Frequency Shift Keying), los unos y ceros recibidos en TD el Modem se traducirán en dos frecuencias, una para el uno y otra distinta para el cero lógicos, haciendo la operación inversa en la cadena de recepción.

Este modem está basado en el chip (Am7910), básicamente un modem. El trabajo, en lo que a diseño se refiere, se ha reducido a la lógica de control del aparato, la fuente de alimentación y todo el soporte del mismo.

Con este diseño se ha logrado un aparato muy versátil y barato. Puede trabajar en (simplex) o (duplex) y tiene varias velocidades y protocolos de comunicación. Se han incluido las más usuales, como son 300 600 y 1200 baudios. El chip permite la programación de más formas de transmisión, pero el hecho de abaratar al máximo el coste del aparato hizo que se usaran las más usuales.

ESPECIFICACIONES

III.1.- Normas del CCITT que cumple

Consultado el tomo VIII fascículo VIII.1 del CCITT (Comité Consultivo Internacional Telegráfico y Telefónico), el Multimodem cumple las recomendaciones V.21 en todos sus modos y parte de las recomendaciones V.23. El acceso al modem desde el terminal de datos es a través de un interfase RS232C.

Recomendación V.21 (CCITT).-

Modem duplex hasta 300 baudios normalizado para uso en la red telefónica conmutada.

Modo 1.- Se considera el modo (1) el que transmite en las dos frecuencias bajas y recibe en las altas. Para el uno y cero lógicos en la transmisión las frecuencias son 980 Hz. y 1180 Hz. respectivamente, mientras que para la recepción las frecuencias son 1650 Hz. y 1850 Hz.

Modo 2.- En este modo se utilizan para la transmisión las frecuencias altas y para la recepción las bajas.

En esta misma recomendación se aclara que el abonado que origina la llamada ha de utilizar el modo (1), dejando al abonado llamado el modo (2). De esta manera no habrá error en la comunicación por haber seleccionado ambos el mismo modo.

Recomendación V.23 (CCITT)

Modem hasta 600/1200 baudios normalizado para el uso en la red telefónica conmutada.

Esta recomendación tiene dos modos ambos Half Duplex con la posibilidad de un canal de retorno a 75 baudios, canal que se usa para la convalidación de datos recibidos protegiendo de esta manera contra errores los datos recibidos.

Modo 1.- La velocidad de transmisión en este modo es de hasta 600 baudios. Para este modo las frecuencias en línea son de 1300 Hz. y 1700 Hz. correspondiendo al uno y cero lógicos en la entrada de datos correspondiente (pin 2 del conector (cannon 25 pines) del RS232).

Modo 2.- La velocidad de transmisión para este modo es de hasta 1200 baudios y las frecuencias en línea 1300 Hz. y 2100 Hz. correspondientes al uno y cero lógicos respectivamente.

Actuando sobre el conmutador S1 se seleccionan los diferentes modos de transmisión y protocolo de que consta el modem.

En las posiciones 1 y 2 el modem está trabajando en full duplex, a una velocidad de hasta 300 baudios y cumpliendo la norma V.21 del CCITT.

En la posición 3 el modem trabaja en simplex hasta una velocidad de 600 baudios.

En la posición 4 el modem trabaja en simplex hasta una velocidad de 1200 baudios.

En las posiciones 3 y 4 se cumple la norma V.23 del CCITT en estos modos, no en su totalidad, ya que no se incluye la posibilidad de un canal de retorno.

*INSTRUCCIONES DE OPERACION
Y TEORIA DE FUNCIONAMIENTO*

III.2.1.- Instrucciones de operación

La parte frontal el multimodem tiene tres conmutadores y cinco indicadores ópticos.

El conmutador S1 tiene 4 posiciones. Este conmutador es para la selección de las velocidades y protocolos del modem.

La posición (1) selecciona la velocidad de 300 baudios (originate) y la posición (2) la misma velocidad (answer); ambas posiciones son Full duplex y cumplen la norma V.21 del CCITT.

La posición (3) selecciona la velocidad de 1200 baudios; su conexión es simplex.

La posición (4) selecciona la velocidad de 600 baudios y conexión simplex.

Hay cuatro indicadores ópticos (leds) que indican en todo momento la evolución de las señales más importantes, como son: (TD), (RD), (CTS) y (RTS).

El conmutador S2 es para conectar el modem a la red telefónica, mientras este conmutador este off el modem no está conectado a la red.

El conmutador S3 es para conectar la alimentación al aparato, tiene un indicador óptico (neon) que indica la correcta aplicación de la red eléctrica.

En la parte trasera del multimodem hay tres conectores. Uno de ellos es para la conexión a la red eléctrica. Este lleva un fusible en la misma base del conector.

El segundo conector es para la conexión a la red telefónica; este lleva dos hilos y acoplan el modem a la red por medio de un trnaformador.

El tercer conector es para la conexión con el terminal de datos. Se trata del conector Cannon de 25 pines; los niveles eléctricos y longitud del cable cumplen la norma RS-232C en la que se encuentran incluidas la V.24 y V.28 del CCITT.

TEORIA DE FUNCIONAMIENTO

III.2.2.- Teoria de funcionamiento

PROGRAMACION DEL MODEM:

Con este conmutador se hace la programación de las entradas de control del modem. El circuito de la figura (III.1.) proporciona las combinaciones que nos interesan con un mínimo de componentes.

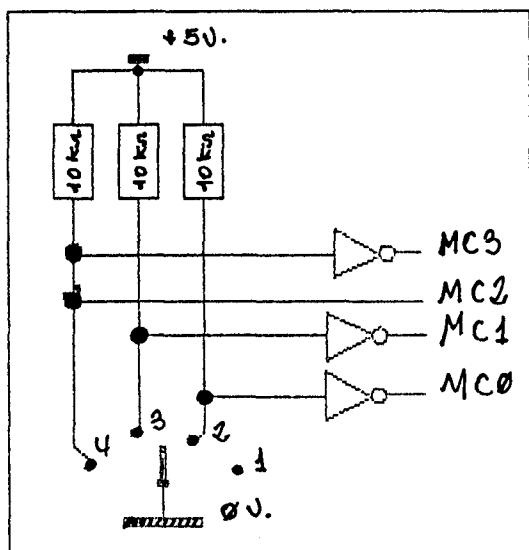


figura (III.1.)

Con sólo tres resistencias y un C.I. de inversores (MC14069), se hace toda la programación del modem. Para simplificar el diseño del circuito impreso del modem las resistencias van montadas sobre el mismo conmutador.

Posición de S1	Código de control				
	MC4	MC3	MC2	MC1	MC0
1	0	0	1	0	0
2	0	0	1	0	1
3	0	0	1	1	0
4	0	1	0	0	0

INTERFASES (RS232C/TTL) Y (TTL/RS232C):

RS232C/TTL.- La transmisión del ordenador hacia el Modem se hace a través de esta interfase. Este circuito pasa los niveles lógicos (RS232C), +12V. y -12V. a niveles lógicos (TTL), 0V. y +5V, y los invierte.

El circuito utilizado es el MC1489 que corresponde al IC2 del esquema eléctrico del modem .

TTL/RS232C.- La transmisión de la información desde el Modem al ordenador se hace a través de los circuitos MC1488. Cada uno de ellos consta de 4 drivers que convierten los niveles lógicos TTL a los niveles lógicos RS232C. e invierten la polaridad.

Para mantener un Slew-Rate de salida de 30V. por μ seg. o menor, el fabricante aconseja poner a las salidas unas capacidades de 300 pF.

LOGICA DE CONTROL:

De una manera muy simple se ha conseguido inhibir la salida cuando el terminal de control CTS este a nivel alto.

Cuando este terminal este a nivel alto, implicará que el modem no acepta datos para transmitir desde el terminal de datos, ya sea porque está recibiendo datos en un modo simplex o por cualquier otro motivo controlado.

Para asegurar esta situación se ha puesto el diodo D1 entre los terminales de TD y CTS. Cuando CTS este a nivel alto el terminal TD también lo estará. Así cualquier dato que venga por este terminal quedará eliminado entre D1 y R1.

RESET DEL MODEM:

Al conectar el modem se le da un reset automático. Este reset pone toda la circuiteria interior del (Am/910) a cero para comenzar a ser programada.

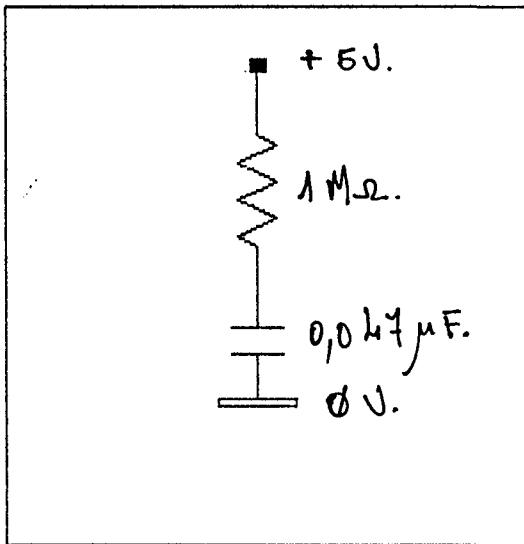


fig. (III.2.)

Integrando ambos miembros resulta:

$$\Rightarrow \ln [5-V_c(t)] - \ln K = -t/R \times C$$

aplicando log. a ambos miembros, resulta:

$$5 = I(t) \times R + V_c(t) \quad (1)$$

$I(t) = C \times dV_c(t)/dt$ sustituyendo en (1) resulta:

$$5 = C \times R \times dV_c(t)/dt + V_c(t) \quad (2)$$

operando resulta:

$$dV_c(t)/dt = [5 - V_c(t)] / R \times C \Rightarrow$$

$$dV_c(t) = [1/R \times C] \times [5 - V_c(t)] dt$$

haciendo el siguiente cambio:

$$[d(5 - V_c(t))] = - dV_c(t) \quad \text{resulta:}$$

$$d[5 - V_c(t)] / [5 - V_c(t)] = -1/R \times C dt$$

$$\ln [5 - V_c(t)] = - [1/R \times C] \times (t + K_a) \Rightarrow$$

$$\ln [5 - V_c(t)] / k = -t/R \times C$$

$$V_c(t) = 5 - K \times \text{EXP}(t/R \times C)$$

Cálculo de K para $t=0$;

$V_c(0) = 5 - K \Rightarrow K = 5 - V_c(0)$ como hemos considerado condiciones iniciales nulas tenemos que $V_c(0) = 0$, y por lo tanto la expresión que da el tiempo de carga y descarga del condensador es:

$$V_c(t) = 5 * [1 - \text{EXP}(-t/R \times C)]$$

Para $V_c(t) = 2.5$ Voltios se reconoce el 1 lógico en la entrada del Am7910, por lo tanto el tiempo de carga será $2.5 = 5 [1 - \text{EXP}(-t/R \times C)]$; de esta expresión sale que el tiempo de carga es $R \times C \times (0,69314718)$.

Con los valores de $1 \text{ M}\Omega$ y $0.047 \mu\text{F}$. da un tiempo de carga de 32 mseg. Con un tiempo de 32 mseg. queda todo reseteado. La puerta más lenta conmutaría mucho más rápido, por lo tanto da tiempo para que cualquier tecnología reconozca el cero aplicado al RESET.

SALIDA A LINEA:

La señal de salida aparece en (TC) del IC1, de aquí pasa directamente al primario del transformador de línea, induciéndose en el secundario, desde donde se conecta a la línea telefónica.

Antes de llegar al primario del transformador de línea, la señal se limita por los diodos (D2, D3, D4 y D5). Una vez limitada por los diodos la señal se pasa por el divisor de tensión que componen la R2 y la resistencia del primario del transformador.

Estos componentes se han calculado de forma que la potencia entregada a la línea telefónica no sea nunca mayor de 1 mW. (norma V.21 del CCITT); esta potencia pasada a unidades relativas equivale a 0 dbm. (dbs. referidos a 1 mW.).

(La R2 = 39 Ω y la Rp = 34 ; VR2+VRp = 1.4 V. como máximo, por el limitador)

La tensión en Rp es $Rp \times 1.4 / (R2 + Rp) = (0.68 \text{ Voltios})$. Los .68 Voltios reflejados en el secundario (1:1 y 600 Ω) equivalen a un consumo de (0.77 mW.). Este consumo está por debajo del mW. permitido.

Para caso de avería el transformador de línea lleva un (safe-fuse); este fusible cortará la transmisión cuando la intensidad a través del devanado sea superior a la permitida. La señal de salida está presente en todo momento en el terminal de recepción del modem. Esta señal no interfiere en la correcta interpretación por parte del Am7910 debido a que tan sólo se transparentarán las frecuencias que tenga asignadas para el cero y el uno en ese momento, y estas frecuencias no pueden ser las de transmisión.

SEÑAL DE LLEGADA:

La señal de recepción que aparece en línea queda limitada a una tensión $V = (V\alpha + Vz)$. La señal se induce en el primario limitándose de nuevo con R2 y los diodos D2, D3, D4 y D5, a una tensión $V_{ppmax.} = 2.4 \text{ V}$.

Si por el contrario la señal que aparece en línea es muy débil, se amplificará en IC5 hasta un nivel máximo que permiten los limitadores.

El condensador CL1 aísla el transformador de la tensión continua de alimentación del aparato telefónico (48 Voltios). El condensador CL2 aísla al limitador de recepción de esta misma tensión.

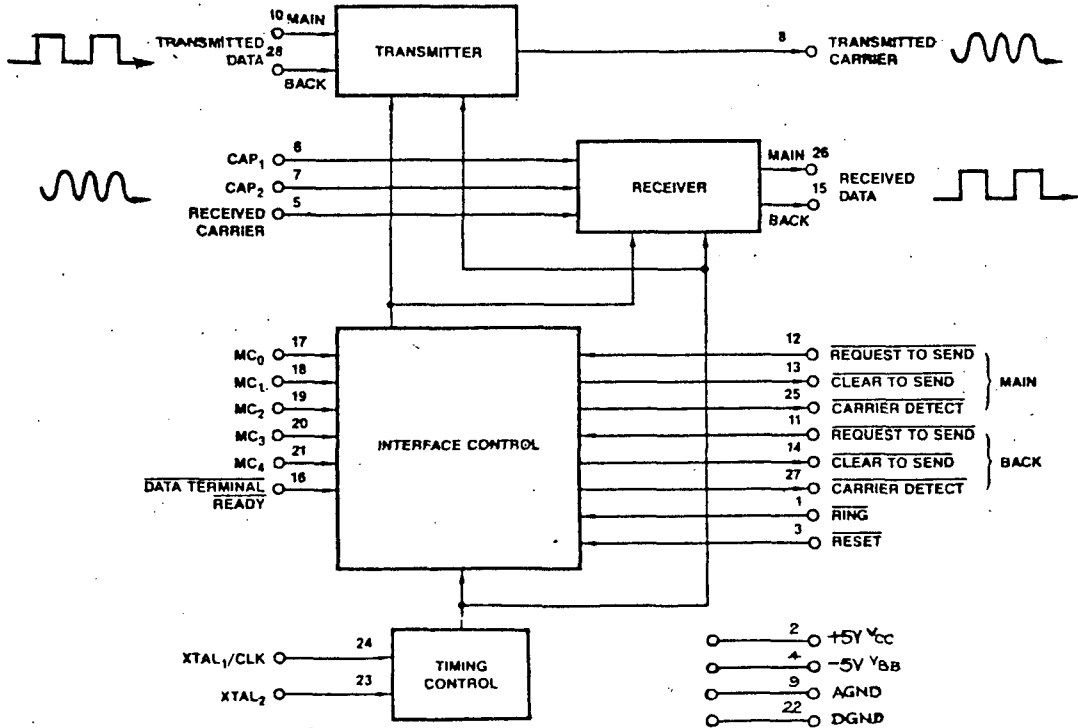
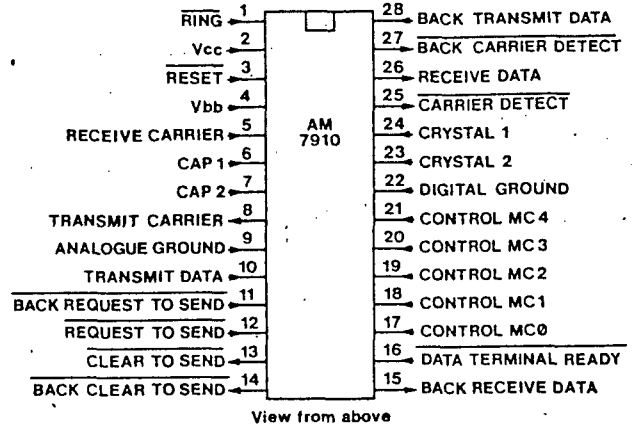
- DIAGRAMA DE BLOQUES -

AM7910 MODEM

A single chip modem capable of receiving and transmitting in full duplex, half duplex or simplex at 1200, 600, 300 or 75 bauds at the frequencies of Bell or CCITT standards. The chip performs modulation, demodulation and performs all necessary filtering with its own internal digital filters. The device runs from a single 2.4576 MHz crystal (FY81C) and has all the essential terminal control signals to RS232C/V24 as well as auto answer.

Characteristics (typical)

Supply voltage pin 2 (V _{CC}):	+5V ±5%
pin 4(V _{BB}):	-5V ±5%
pin 22:	0V (digital ground)
pin 9:	0V ±50mV (analogue ground)
Output high voltage:	>2.4V at -50µA
Output low voltage:	<0.4V at 2mA
Input high voltage:	2V to V _{CC}
Input low voltage:	-0.5V to +0.8V
Supply current (V _{CC}):	<125mA
(V _{BB}):	<25mA
Analogue input resistance:	>50kΩ
Operating input signal:	-1.6V min, +1.6V max
Analogue output voltage:	-1.1V min, +1.1V max into 600Ω



Order As YH93B (AM7910) £44.95 TQ 5

SEMICONDUCTORS SEMICONDUCTORS SEMICONDUCTORS SEMICONDUCTORS SEMICONDUCTORS SEMICONDUCTORS SEMICONDUCTORS SEMICONDUCTORS SEMICONDUCTORS SEMICONDUCTORS SEMI

fig. (III.3.)

FUENTE DE
ALIMENTACION

III.3.- Fuente de alimentación

Para la operación del modem se necesitan cuatro tensiones. Las tensiones son ± 5 Voltios y ± 12 Voltios. La exigencia de la precisión en estas tensiones viene dada por el circuito Am7910. Este circuito es el más exigente, la alimentación del Am7910 es de ± 5 Voltios ± 5 %.

Los ± 12 Voltios tienen un gran margen de precisión. Si se tiene en cuenta que las tensiones RS232C, van desde -3 Voltios hasta -12 Voltios y desde $+3$ Voltios hasta $+12$ Voltios, entonces los márgenes que marcan la precisión de estas tensiones son los márgenes que de el fabricante del interfase MC1488P (se alimenta con ± 12 V.).

Por lo tanto el diseño de la fuente está sujeto a estas premisas; un ± 5 % en las tensiones de ± 5 Voltios y un gran margen, en lo que a tensión se refiere, en los ± 12 Voltios.

El circuito Am7910 reconoce un cero con tensiones de 400 mV. y menores, por lo tanto y en principio, el rizado tiene que ser menor que 400 mV., y de hecho es mucho menor de 400 mV.

Con todas estas consideraciones se ha hecho el diseño de la fuente de alimentación. Esta fuente de alimentación se ha diseñado no con la pretensión de que sea una buena fuente, sino que prestando las exigencias que requieren las tensiones del aparato que se va a alimentar, sea lo más barata posible.

En el diseño se han utilizado; Un transformador, dos diodos zener, un rectificador, dos reguladores, dos resistencias y los filtros necesarios (condensadores). Con estos componentes se ha conseguido una fuente de ± 5 y ± 12 Voltios, con unas prestaciones más que suficientes para el aparato que alimenta.

El consumo total no llega a 600 mA. (entre las cuatro tensiones), pero debido a que el precio de los componentes es prácticamente el mismo, los reguladores de ± 5 Voltios el transformador y el rectificador son de 1 Amperio.

Utilizando estos componentes no habrá problemas de agitación térmica, ésta podría dañar los componentes e introducir señales parásitas en los circuitos que alimenta.

Las limitaciones de las tensiones de ± 5 Voltios están resueltas con la utilización de los reguladores 7805C y 7905C. Estos reguladores aseguran unas tensiones de ± 5 Voltios ± 4 %. Con los condensadores de 4700 μ F. que se han escogido como filtro a la salidas del rectificador, se consigue reducir el rizado de la fuente a menos de 50 mV.

Teoría de funcionamiento.-

La tensión de salida del secundario es de 2×9 Voltios; el punto medio se une y se considera masa y los extremos se llevan al rectificador.

A la salida del rectificador hay ± 8.5 Voltios. Estos ± 8.5 Voltios se ven incrementados en $\pm V \times 1.41$ por el condensador del filtro (el condensador se carga a V_{efc}). De estas tensiones se sacan los ± 5 y los ± 12 Voltios.

Los ± 12 Voltios de salida se aplican a una resistencia en serie con un zener de 12 Voltios. Del punto medio de la resistencia y el zener se sacan los 12 Voltios que utilizaremos para la alimentación, esta misma tensión se lleva a la entrada de un regulador de +5 Voltios y la salida de este regulador es la tensión de +5 Voltios. En las tensiones negativas las conexiones son las mismas, pero teniendo en cuenta la polaridad del zener, los condensadores electrolíticos y las patillas del regulador.

De esta manera los ± 12 Voltios están regulados ante pequeñas subidas de tensión de red, aunque la potencia que pueden desarrollar estos ± 12 Voltios es muy limitada, ya que en realidad son ± 9 Voltios el resto hasta los ± 12 Voltios lo dan la carga de los condensadores.

LISTA DE
COMPONENTES

III.4.1.- Lista de componentes

TARJETA DEL MODEM

<u>COMPONENTE</u>		<u>CODIGO FABRICANTE</u>	<u>SUMINISTRADOR</u>
<u>CIRCUITOS INTEGRADOS</u>			
IC-1	(Modem FSK)	Am7910	M. Elect.
IC-2	(interfase)	MC-1489N	"
IC-3	(interfase)	MC-1488N	"
IC-4	(inversores)	CD-4069CN	Local
IC-5	(Amp. Op.)	UA741CP	"
<u>RESISTENCIAS Y CRISTALES</u>			
XTAL	(cristal C.)	2.4576 Mhz.	E.SD. (Md.)
R1-8	(Resistencia)	carbon 10K Ω - $\frac{1}{4}$ W	Local
R9	(Resistencia)	carbon 1M Ω - $\frac{1}{4}$ W	"
R10	(Resistencia)	carbon 100 Ω - $\frac{1}{4}$ W	"
R11	(Resistencia)	carbon 600 Ω - $\frac{1}{4}$ W	"
R12,13	(Resistencia)	carbon 22K Ω - $\frac{1}{4}$ W	"
R14,15	(Resistencia)	carbon 39 Ω - $\frac{1}{4}$ W	"
<u>DIODOS</u>			
D1-5	(Diodos)	1N-4148	Local
D6,7	(Zener)	1N752-A	

CONDENSADORES

C1-6	(Cond. Ceramico)	300 pF.	Local
C7,8	(Cond. mica)	22 pF.	"
C9	(Cond. Ceramico)	0.047 μ F.	"
C10	(Cond. mica)	2000 pF.	"
C11,12	(Cond. Papel)	0.08 μ F.	"

TRANSFORMADOR

T1	(Transformador)	T. 1/1 IMP=600 Ω	Maplin Elect.
----	-----------------	-------------------------	---------------

FUENTE DE ALIMENTACION

COMPONENTES

T1	Transformador	120,220/(2x9)	Local
PR1	Fuente Rect.	WO-5 1 A.	Maplin Elect.
IC1	Regulador	μ A7805UC 5V.	"
IC2	Regulador	μ A7905UC -5V.	"
C1,2	Cond. Elect.	4700 μ F.	Local
C3,4	Cond. Elect.	220 μ F.	"
C5-8	Cond. ceramico	4.7 nF.	"

ESQUEMAS

ELECTRICOS

III.5.- Esquemas eléctricos

En este apartado se incluyen los siguientes dibujos:

fig. (III.4.).- Este dibujo es de la placa del modem vista desde la parte de las pistas. Esta xerocopia se hizo durante el proceso de construcción del modem, antes de taladrar los agujeros para los componentes.

fig. (III.5.).- Este dibujo corresponde a la misma placa del modem, esta vez vista desde la cara donde irán los componentes.

fig. (III.6.).- Este dibujo es el esquema eléctrico de la fuente de alimentación. Donde dice conector de alimentación, se trata de un conector de cinco pines que une las tarjetas del modem y de la fuente de alimentación.

fig. (III.7.).- Este dibujo es la placa de la fuente de alimentación, vista desde la parte de las pistas.

fig. (III.8.).- Este dibujo es el esquema eléctrico del Multimodem.

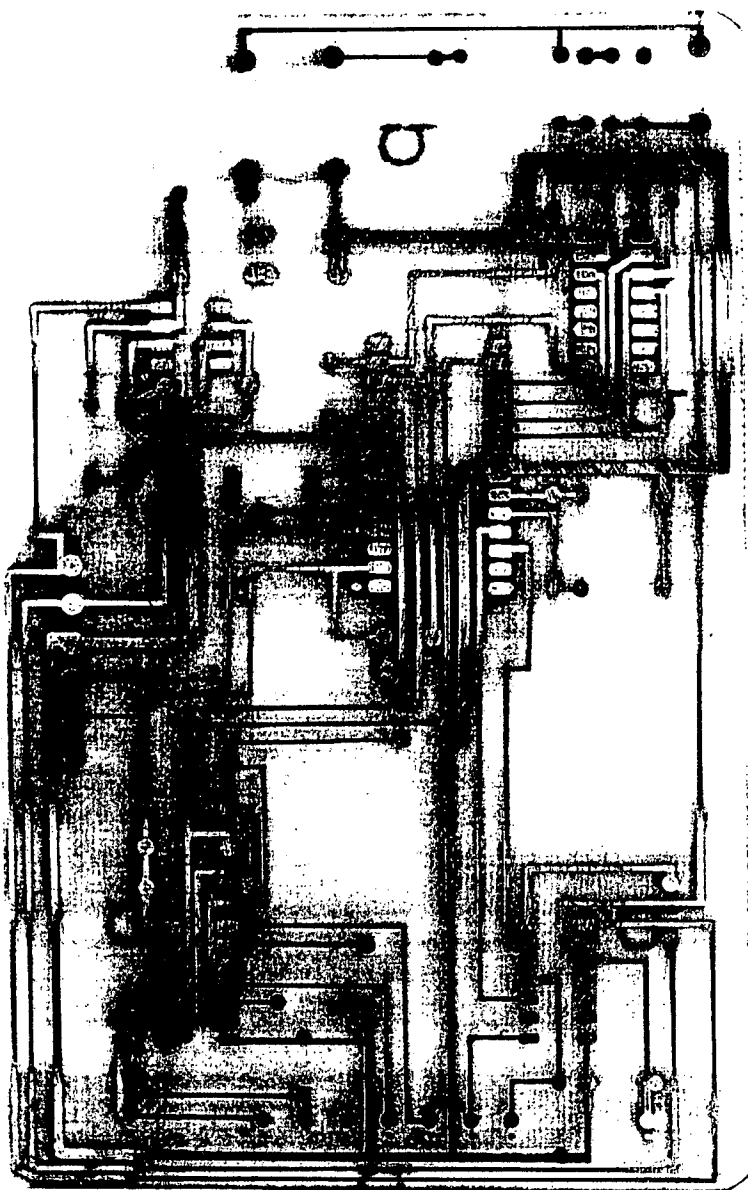
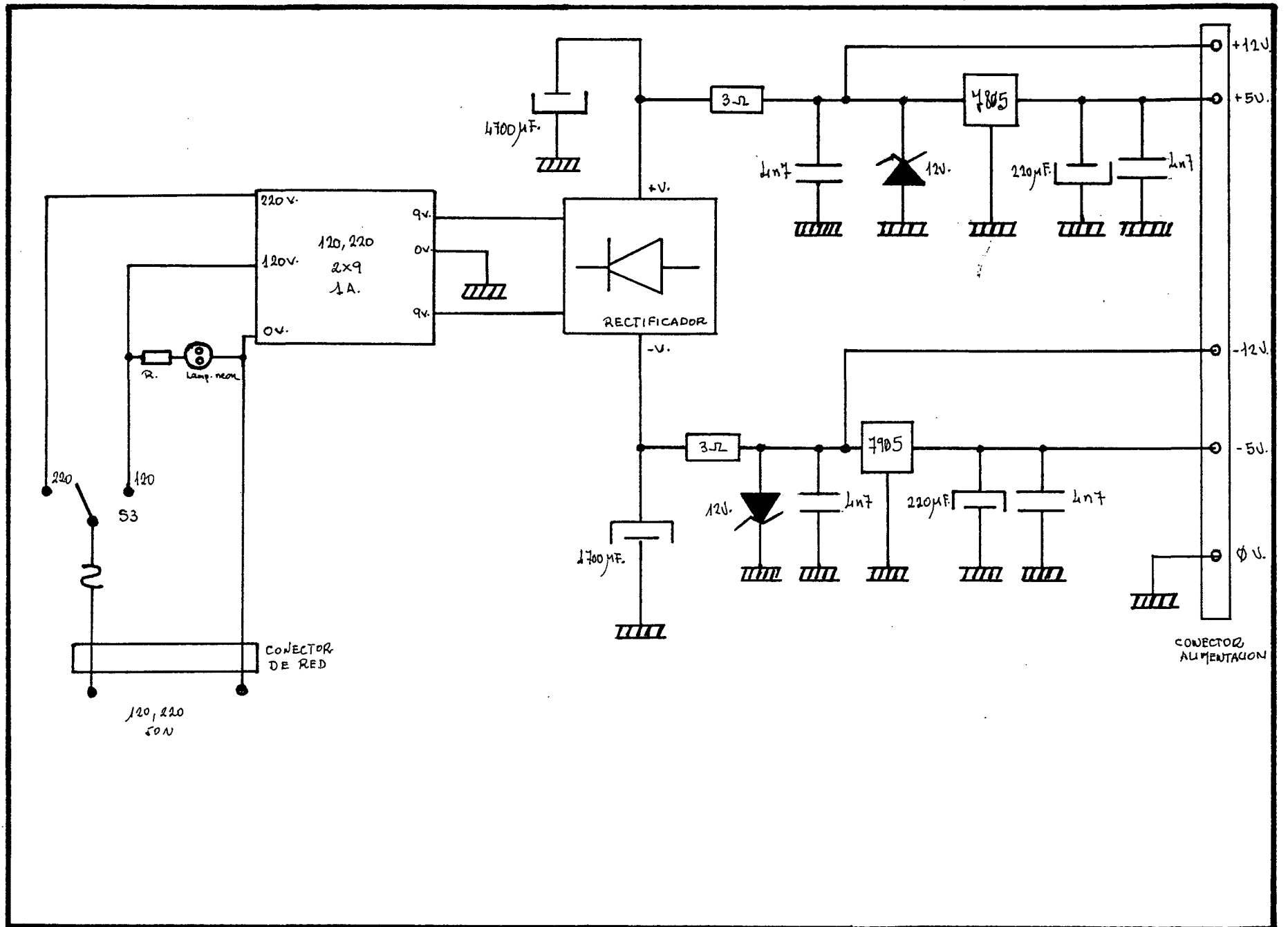


fig. (III.4.) CTO. IMPRESO DEL MODEM (CARA DE PISTA).



fig. (III.5.) CTO. IMPRESO DEL MODEM (CARA DE COMP.).

FIG. (III.6.) FUENTE DE ALIMENTACION.



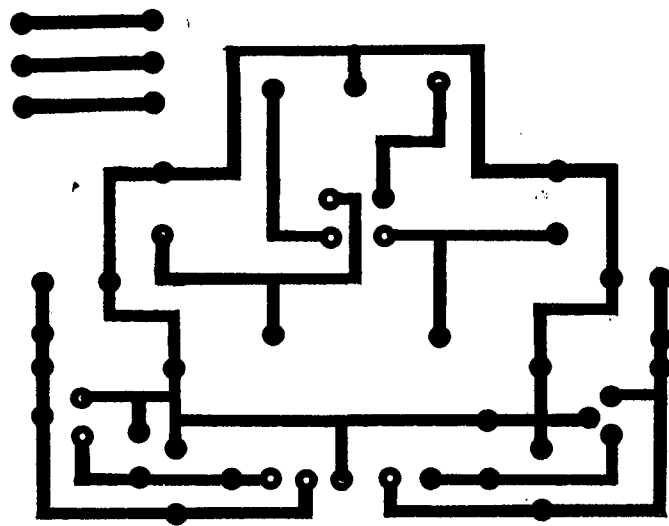
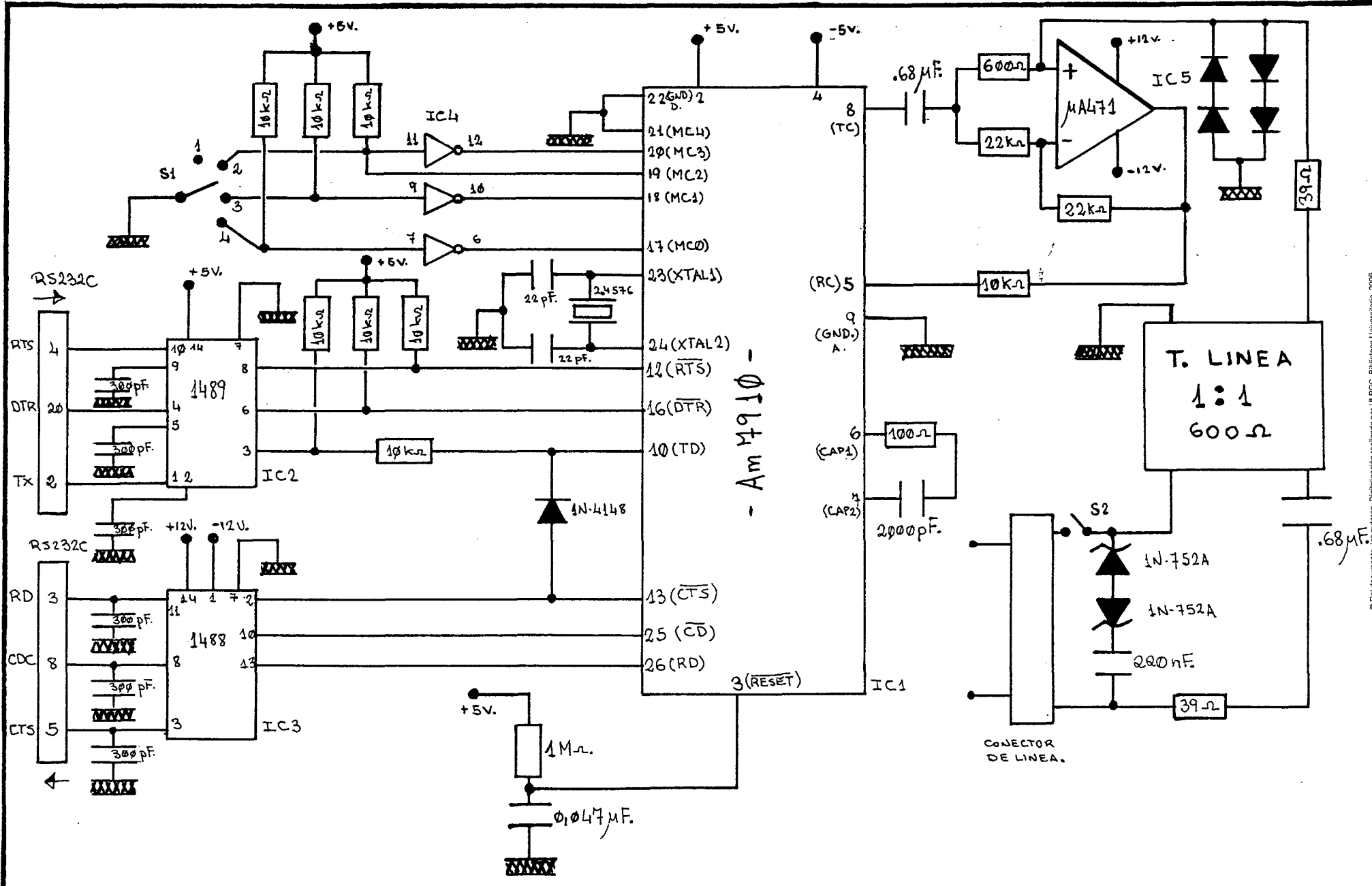


fig. (III.7.) CTO. IMPRESO FUENTE DE ALIMENTACION.

Figura (III.8.) ESQUEMA ELECTRICO DEL MULTIMODEM.



PRESUPUESTO

III.6.- Presupuesto del modem y de la base de datos.-

Precio de los componentes utilizados:

Placa de la fuente de alimentación:

Transformador	720 R	Local.
disipador	70 R	"
condensadores (4700 μ F.)	400 R	"
condensadores (220 μ F.)	200 R	"
reguladores	140 R	M. Elec.
rectificador	60 R	"
cond. ceramicos	40 R	"
placa	100 R	"
D. zener	60 R	"
conector red	220 R	"
interruptor de red	110 R	"
TOTAL COMPONENTES...	2020 R	

Placa del modem:

Transformador linea	930 R	M. Elec.
Am7910	2956 R	"
MC1489	120 R	"
MC1488	120 R	"
MC4069CN	75 R	"
μ A741CP	75 R	"
Cristal	215 R	E. S. D. (M)
Resistencias	300 R	Local
diodos	60 R	"
diodos Z.	90 R	"
condensadores	420 R	"
conectores a F.A.	160 R	M. Elec.
interruptor S1	120 R	"
interruptor S2	110 R	"
placa	100 R	"
TOTAL COMPONENTES...	5851 R	

Para atacar las placas se ha utilizado el atacador rápido aristón, un litro de este atacador tiene un precio de 816 R en el mercado local.

Para el dibujo de las placas se ha utilizado un rotulador FCB-pen, cuyo precio es de 120 R en (Map.-Elect.).

La caja donde se aloja el modem tiene un precio de 2200 R, y los tacos de goma que la apoyan 60 R. Los conectores Cannon y de red telefónica tienen unos precios de 420 R y 40 R respectivamente. El cable de red tiene un precio de 230 R. Con el atacador hay para unas 10 placas, y sólo se ha usado para dos, por lo tanto la parte proporcional que le corresponde a nuestro presupuesto es de ≈164 R.

Gastos materiales totales:

Componentes fuente alimentación:	2020 R
Componentes modem:	5851 R
materiales construcción placas	284 R
Caja y conectores	2720 R

GASTOS TOTALES...	10875 R

Si este aparato se fabrica en serie por una firma, hay que tener en cuenta, para su precio final, una gran serie de gastos y circunstancias.

En este supuesto los materiales se verían muy rebajados de precio, pero tendríamos que tener en cuenta toda la serie de gastos a que está sometida una empresa.

Tendríamos que tener en cuenta: La amortización de la maquinaria de la empresa, el alquiler del solar donde se encuentra o en su defecto la amortización del mismo, los salarios de los trabajadores y los gastos de seguridad social, los gastos de mantenimiento, los gastos de consumo de energía y mobiliario, los gastos de administración de la empresa y los gastos de gestión y suministro del producto terminado.

A todos estos gastos tendríamos que hallarles la parte proporcional que le corresponde a uno de nuestros aparatos, para ello se verá la cantidad que se factura cada mes y se le hallarán los gastos proporcionales a un aparato.

De esta manera se podría hacer un supuesto práctico real del precio en el mercado de este modem. Por tratarse de un prototipo no se tienen datos suficientes para hacer todas estas consideraciones, pero investigando en el mercado actual se puede ver aproximadamente que el precio de este aparato sería menor que la mayoría y presta más utilidades.

Presupuesto de la base de datos.-

En este presupuesto no hay gastos materiales, sólo el desgaste del equipo informático que se ha utilizado para su elaboración y el soporte físico (disco) para su almacenaje.

En realidad de esta parte del proyecto no podría dar un presupuesto y ser objetivo en el mismo. Este trabajo ha sido el resultado de casi tres años de investigación en Dr. Logo y muchas horas de programación.

APENDICE

LINEAR INTEGRATED CIRCUITS

TYPES μ A741M, μ A741C GENERAL-PURPOSE OPERATIONAL AMPLIFIERS

BULLETIN NO. DL-S 11363, NOVEMBER 1970—REVISED OCTOBER 1979

- Short-Circuit Protection
- Offset-Voltage Null Capability
- Large Common-Mode and Differential Voltage Ranges
- No Frequency Compensation Required
- Low Power Consumption
- No Latch-up

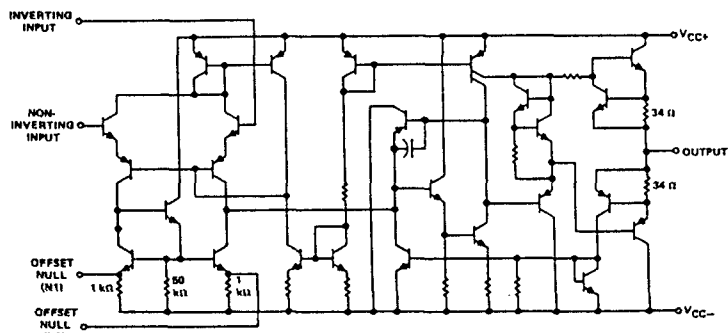
description

The μ A741 is a general-purpose operational amplifier featuring offset-voltage null capability.

The high common-mode input voltage range and the absence of latch-up make the amplifier ideal for voltage-follower applications. The device is short-circuit protected and the internal frequency compensation ensures stability without external components. A low-value potentiometer may be connected between the offset null inputs to null out the offset voltage as shown in Figure 2.

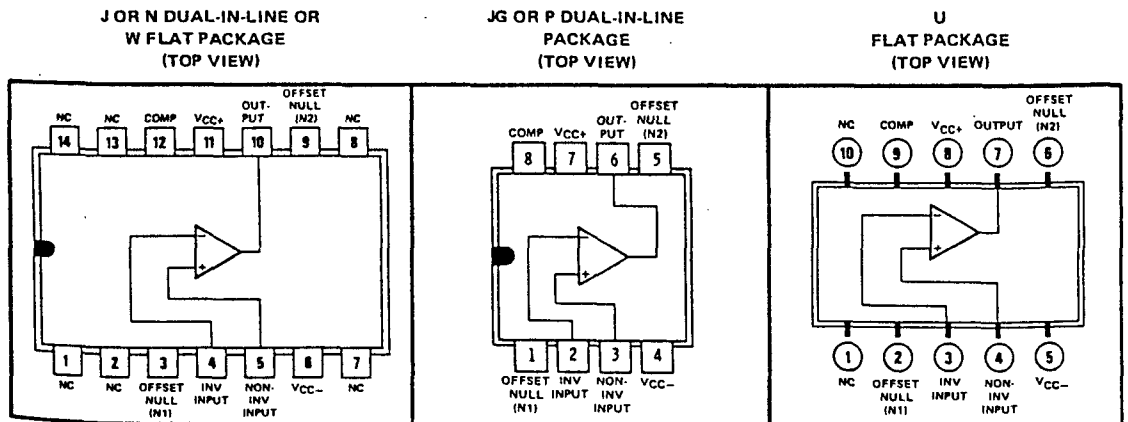
The μ A741M is characterized for operation over the full military temperature range of -55°C to 125°C ; the μ A741C is characterized for operation from 0°C to 70°C .

schematic



Resistor values shown are nominal

terminal assignments



NC—No internal connection

Copyright © 1979 by Texas Instruments Incorporated

TEXAS INSTRUMENTS
INCORPORATED

POST OFFICE BOX 225012 • DALLAS, TEXAS 75285

TYPES μ A741M, μ A741C

GENERAL-PURPOSE OPERATIONAL AMPLIFIERS

absolute maximum ratings over operating free-air temperature range (unless otherwise noted)

	μ A741M	μ A741C	UNIT
Supply voltage V_{CC+} (see Note 1)	22	18	V
Supply voltage V_{CC-} (see Note 1)	-22	-18	V
Differential input voltage (see Note 2)	± 30	± 30	V
Input voltage (either input, see Notes 1 and 3)	± 15	± 15	V
Voltage between either offset null terminal (N1/N2) and V_{CC-}	± 0.5	± 0.5	V
Duration of output short-circuit (see Note 4)	unlimited	unlimited	
Continuous total power dissipation at (or below) 25°C free-air temperature (see Note 5)	500	500	mW
Operating free-air temperature range	-55 to 125	0 to 70	°C
Storage temperature range	-65 to 150	-65 to 150	°C
Lead temperature 1/16 inch (1,6 mm) from case for 60 seconds	J, JG, U, or W package		300
Lead temperature 1/16 inch (1,6 mm) from case for 10 seconds	N or P package		260

- NOTES:
- All voltage values, unless otherwise noted, are with respect to the midpoint between V_{CC+} and V_{CC-} .
 - Differential voltages are at the noninverting input terminal with respect to the inverting input terminal.
 - The magnitude of the input voltage must never exceed the magnitude of the supply voltage or 15 volts, whichever is less.
 - The output may be shorted to ground or either power supply. For the μ A741M only, the unlimited duration of the short-circuit applies at (or below) 125°C case temperature or 75°C free-air temperature.
 - For operation above 25°C free-air temperature, refer to Dissipation Derating Curves, Section 2. In the J and JG packages, μ A741M chips are alloy-mounted; μ A741C chips are glass-mounted.

electrical characteristics at specified free-air temperature, $V_{CC+} = 15$ V, $V_{CC-} = -15$ V

PARAMETER	TEST CONDITIONS†	μ A741M			μ A741C			UNIT
		MIN	TYP	MAX	MIN	TYP	MAX	
V_{IO}	Input offset voltage	$R_S < 10$ k Ω	25°C	1	5	1	6	mV
$\Delta V_{IO(adj)}$	Offset voltage adjust range		25°C	± 15		± 15		mV
I_{IO}	Input offset current		25°C	20	200	20	200	nA
			Full range	500		300		
I_{IB}	Input bias current		25°C	80	500	80	500	nA
			Full range	1500		800		
V_{ICR}	Common-mode input voltage range		25°C	± 12	± 13	± 12	± 13	V
			Full range	± 12		± 12		
V_{OPP}	Maximum peak-to-peak output voltage swing	$R_L = 10$ k Ω	25°C	24	28	24	28	V
		$R_L > 10$ k Ω	Full range	24		24		
		$R_L = 2$ k Ω	25°C	20	26	20	26	
		$R_L > 2$ k Ω	Full range	20		20		
A_{VD}	Large-signal differential voltage amplification	$R_L > 2$ k Ω , $V_O = \pm 10$ V	25°C	50	200	20	200	V/mV
			Full range	25		15		
r_i	Input resistance		25°C	0.3	2	0.3	2	M Ω
r_o	Output resistance	$V_O = 0$ V, See Note 6	25°C	75		75		Ω
C_i	Input capacitance		25°C	1.4		1.4		pF
CMRR	Common-mode rejection ratio	$R_S < 10$ k Ω	25°C	70	90	70	90	dB
			Full range	70		70		
k_{SVS}	Supply voltage sensitivity ($\Delta V_{IO}/\Delta V_{CC}$)	$R_S < 10$ k Ω	25°C	30	150	30	150	μ V/V
I_{OS}	Short-circuit output current		25°C	± 25	± 40	± 25	± 40	mA
			Full range	3.3		3.3		
I_{CC}	Supply current	No load,	25°C	1.7	2.8	1.7	2.8	mA
		No signal	Full range	3.3		3.3		
P_D	Total power dissipation	No load,	25°C	50	85	50	85	mW
		No signal	Full range	100		100		

† All characteristics are specified under open-loop operation. Full range for μ A741M is -55°C to 125°C and for μ A741C is 0°C to 70°C.

NOTE 6: This typical value applies only at frequencies above a few hundred hertz because of the effects of drift and thermal feedback.

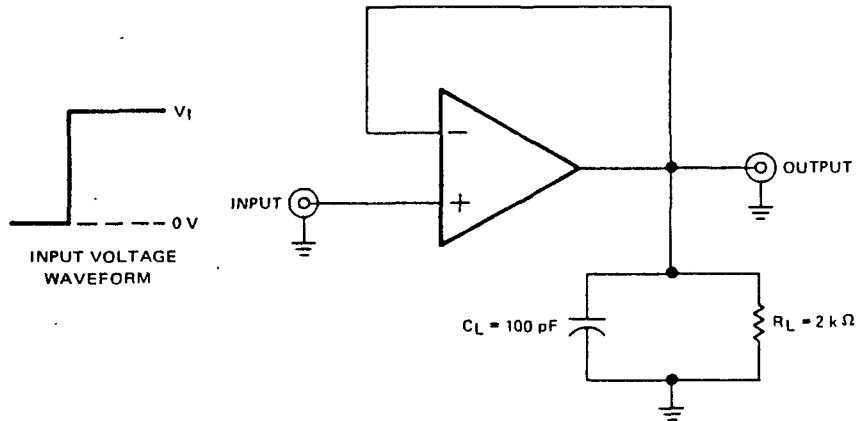
TYPES μ A741M, μ A741C

GENERAL-PURPOSE OPERATIONAL AMPLIFIERS

operating characteristics, $V_{CC+} = 15\text{ V}$, $V_{CC-} = -15\text{ V}$, $T_A = 25^\circ\text{C}$

PARAMETER	TEST CONDITIONS	μ A741M			μ A741C			UNIT
		MIN	TYP	MAX	MIN	TYP	MAX	
t_r	Rise time	$V_I = 20\text{ mV}$, $R_L = 2\text{ k}\Omega$, $C_L = 100\text{ pF}$, See Figure 1						μs
	Overshoot factor	5%						
SR	Slew rate at unity gain	$V_I = 10\text{ V}$, $R_L = 2\text{ k}\Omega$, $C_L = 100\text{ pF}$, See Figure 1						$\text{V}/\mu\text{s}$

PARAMETER MEASUREMENT INFORMATION



TEST CIRCUIT

FIGURE 1—RISE TIME, OVERSHOOT, AND SLEW RATE

TYPICAL APPLICATION DATA

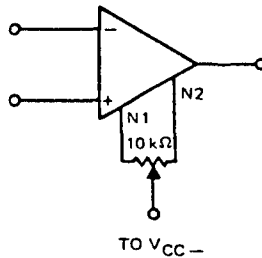


FIGURE 2—INPUT OFFSET VOLTAGE NULL CIRCUIT

TYPES μ A741M, μ A741C GENERAL-PURPOSE OPERATIONAL AMPLIFIERS

TYPICAL CHARACTERISTICS

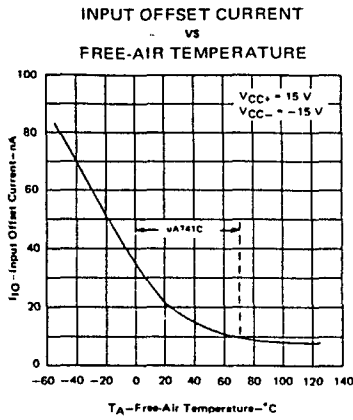


FIGURE 3

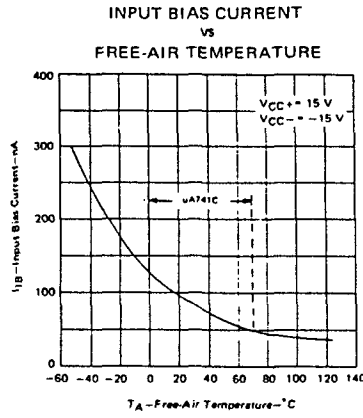


FIGURE 4

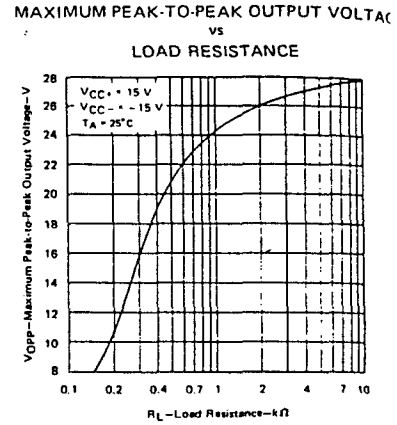


FIGURE 5

MAXIMUM PEAK-TO-PEAK OUTPUT VOLTAGE VS FREQUENCY

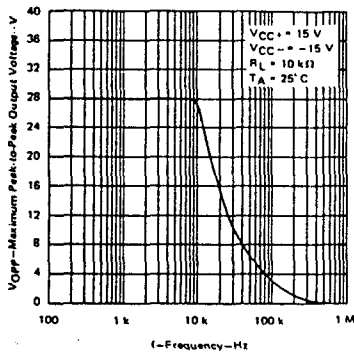


FIGURE 6

OPEN-LOOP LARGE-SIGNAL DIFFERENTIAL VOLTAGE AMPLIFICATION VS SUPPLY VOLTAGE

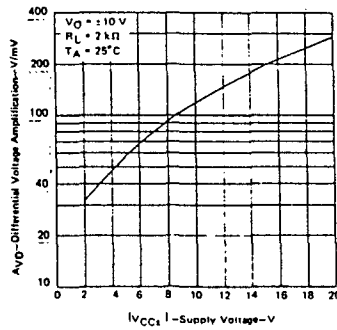


FIGURE 7

OPEN-LOOP LARGE-SIGNAL DIFFERENTIAL VOLTAGE AMPLIFICATION VS FREQUENCY

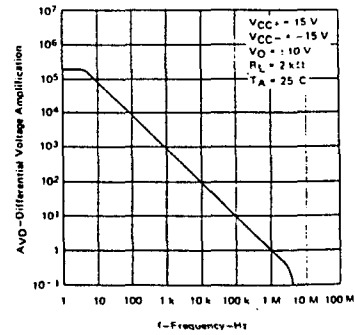


FIGURE 8

COMMON-MODE REJECTION RATIO VS FREQUENCY

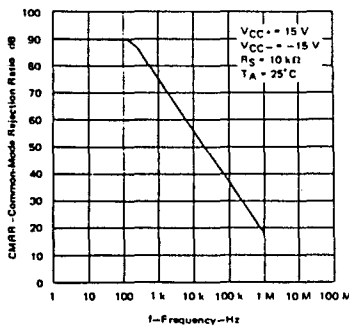


FIGURE 9

OUTPUT VOLTAGE VS ELAPSED TIME

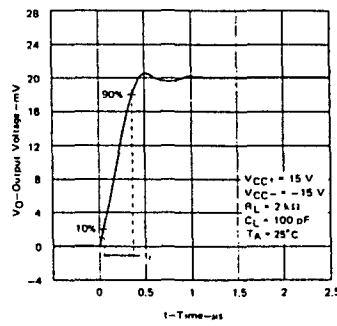


FIGURE 10

VOLTAGE-FOLLOWER LARGE-SIGNAL PULSE RESPONSE

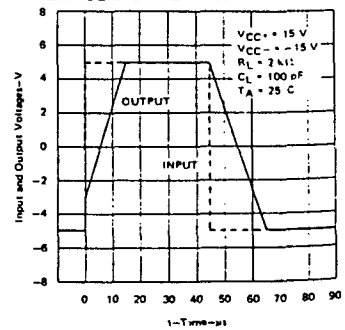


FIGURE 11



MC14069UB

HEX INVERTER

The MC14069UB hex inverter is constructed with MOS P-channel and N-channel enhancement mode devices in a single monolithic structure. These inverters find primary use where low power dissipation and/or high noise immunity is desired. Each of the six inverters is a single stage to minimize propagation delays.

- Quiescent Current = 0.5 nA typ/pkg @ 5 Vdc
- Noise Immunity = 45% of V_{DD} typ
- Supply Voltage Range = 3.0 Vdc to 18 Vdc
- Capable of Driving Two Low-Power TTL Loads, One Low-Power Schottky TTL Load or Two HTL Loads Over the Rated Temperature Range
- Double Diode Protection on All Inputs
- Pin-for-Pin Replacement for CD4069UB
- Meets JEDEC UB Specifications

CMOS SSI

(LOW-POWER COMPLEMENTARY MOS)

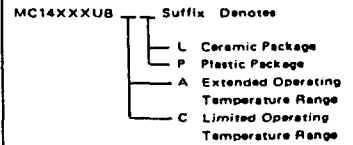
HEX INVERTER



L SUFFIX
CERAMIC PACKAGE
CASE 632

P SUFFIX
PLASTIC PACKAGE
CASE 646

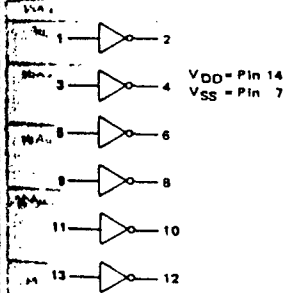
ORDERING INFORMATION



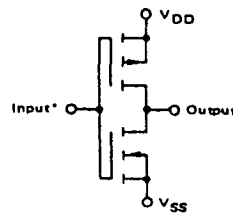
MAXIMUM RATINGS (Voltages referenced to V_{SS})

Rating	Symbol	Value	Unit
DC Supply Voltage	V _{DD}	-0.5 to +18	Vdc
Input Voltage, All Inputs	V _{in}	-0.5 to V _{DD} + 0.5	Vdc
DC Current Drain per Pin	I	10	mAdc
Operating Temperature Range — AL Device	T _A	-55 to +125	°C
		-40 to +85	°C
Storage Temperature Range	T _{stg}	-65 to +150	°C

LOGIC DIAGRAM



CIRCUIT SCHEMATIC (1/6 OF CIRCUIT SHOWN)

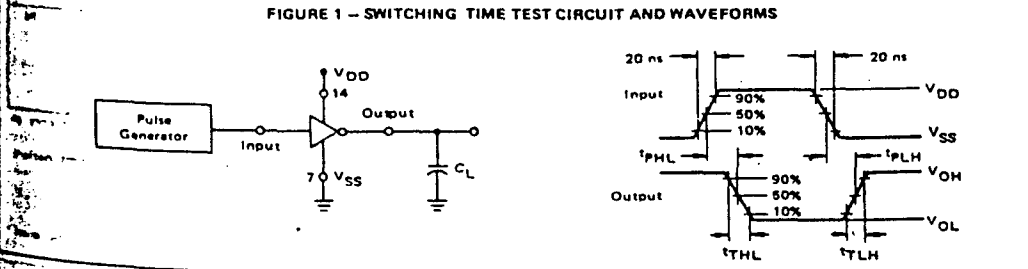


*Double diode protection on all inputs not shown.

This device contains circuitry to protect the inputs against damage due to high static voltages or electric fields; however, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum rated voltages to this high impedance circuit. For proper operation it is recommended that V_{in} and V_{out} be constrained to the range V_{SS} < V_{in} or V_{out} < V_{DD}.

Unused inputs must always be tied to an appropriate logic voltage level (e.g., either V_{SS} or V_{DD}).

FIGURE 1 — SWITCHING TIME TEST CIRCUIT AND WAVEFORMS



MC14069UB

ELECTRICAL CHARACTERISTICS

Characteristic	Symbol	V _{DD} Vdc	T _{low} *		25°C			T _{high} *		Unit	
			Min	Max	Min	Typ	Max	Min	Max		
Output Voltage V _{in} V _{DD} or 0	"0" Level V _{OL}	5.0	-	0.05	-	0	0.05	-	0.05	Vdc	
		10	-	0.05	-	0	0.05	-	0.05		
		15	-	0.05	-	0	0.05	-	0.05		
V _{in} 0 or V _{DD}	"1" Level V _{OH}	5.0	4.95	-	4.95	5.0	-	4.95	-	Vdc	
		10	9.95	-	9.95	10	-	9.95	-		
		15	14.95	-	14.95	15	-	14.95	-		
Input Voltage** (V _O = 4.5 Vdc) (V _O = 9.0 Vdc) (V _O = 13.5 Vdc)	"0" Level V _{IL}	5.0	-	1.0	-	2.25	1.0	-	1.0	Vdc	
		10	-	2.0	-	4.50	2.0	-	2.0		
		15	-	2.5	-	6.75	2.5	-	2.5		
	"1" Level V _{IH}	5.0	4.0	-	4.0	2.75	-	4.0	-	Vdc	
		10	8.0	-	8.0	5.50	-	8.0	-		
		15	12.5	-	12.5	8.25	-	12.5	-		
Output Drive Current (AL Device) Source	V _{OH} = 2.5 Vdc V _{OH} = 4.6 Vdc V _{OH} = 9.5 Vdc V _{OH} = 13.5 Vdc	5.0	-3.0	-	-2.4	-4.2	-	-1.7	-	mA _{dc}	
		5.0	-0.64	-	-0.51	-0.88	-	-0.36	-		
		10	-1.6	-	-1.3	-2.25	-	-0.9	-		
		15	-4.2	-	-3.4	-8.8	-	-2.4	-		
	V _{OL} = 0.4 Vdc V _{OL} = 0.5 Vdc V _{OL} = 1.5 Vdc	Sink I _{OL}	5.0	0.64	-	0.51	0.88	-	0.36	-	mA _{dc}
		10	1.6	-	1.3	2.25	-	0.9	-		
15	4.2	-	3.4	8.8	-	2.4	-	-			
Output Drive Current (CL/CP Device) Source	V _{OH} = 2.5 Vdc V _{OH} = 4.6 Vdc V _{OH} = 9.5 Vdc V _{OH} = 13.5 Vdc	5.0	-2.5	-	-2.1	-4.2	-	-1.7	-	mA _{dc}	
		5.0	-0.52	-	-0.44	-0.88	-	-0.36	-		
		10	-1.3	-	-1.1	-2.25	-	-0.9	-		
		15	-3.6	-	-3.0	-8.8	-	-2.4	-		
	V _{OL} = 0.4 Vdc V _{OL} = 0.5 Vdc V _{OL} = 1.5 Vdc	Sink I _{OL}	5.0	0.52	-	0.44	0.88	-	0.36	-	mA _{dc}
		10	1.3	-	1.1	2.25	-	0.9	-		
15	3.6	-	3.0	8.8	-	2.4	-	-			
Input Current (AL Device)	I _{in}	15	-	±0.1	-	±0.00001	±0.1	-	±1.0	μA _{dc}	
Input Current (CL/CP Device)	I _{in}	15	-	±0.3	-	±0.00001	±0.3	-	±1.0	μA _{dc}	
Input Capacitance (V _{in} = 0)	C _{in}	-	-	-	-	5.0	7.5	-	-	pF	
Quiescent Current (AL Device) (Per Package)	I _{DD}	5.0	-	0.25	-	0.0005	0.25	-	7.5	μA _{dc}	
		10	-	0.50	-	0.0010	0.50	-	15		
		15	-	1.00	-	0.0015	1.00	-	30		
Quiescent Current (CL/CP Device) (Per Package)	I _{DD}	5.0	-	1.0	-	0.0005	1.0	-	7.5	μA _{dc}	
		10	-	2.0	-	0.0010	2.0	-	15		
		15	-	4.0	-	0.0015	4.0	-	30		
Total Supply Current***† (Dynamic plus Quiescent, Per Gate) (C _L = 50 pF)	I _T	5.0	I _T = (0.3 μA/kHz) f + I _{DD} /6							μA _{dc}	
10	I _T = (0.6 μA/kHz) f + I _{DD} /6										
15	I _T = (0.9 μA/kHz) f + I _{DD} /6										
Output Rise and Fall Times** (C _L = 50 pF) †TLH, †THL = (1.35 ns/pF) C _L + 33 ns ‡TLH, ‡THL = (0.60 ns/pF) C _L + 20 ns §TLH, §THL = (0.40 ns/pF) C _L + 20 ns	†TLH, †THL	5.0	-	-	-	100	200	-	-	ns	
		10	-	-	-	50	100	-	-		
		15	-	-	-	40	80	-	-		
Propagation Delay Times** (C _L = 50 pF) †PLH, †PHL = (0.90 ns/pF) C _L + 20 ns ‡PLH, ‡PHL = (0.36 ns/pF) C _L + 22 ns §PLH, §PHL = (0.26 ns/pF) C _L + 17 ns	†PLH, †PHL	5.0	-	-	-	65	125	-	-	ns	
		10	-	-	-	40	75	-	-		
		15	-	-	-	30	55	-	-		

*T_{low} = -55°C for AL Device, -40°C for CL/CP Device

T_{high} = +125°C for AL Device, +85°C for CL/CP Device

†To calculate total supply current at loads other than 50 pF:

$$I_T(C_L) = I_T(50 \text{ pF}) + 6 \times 10^{-3}(C_L - 50) V_{DD} f$$

where: I_T is in μA (per package), C_L in pF, V_{DD} in Vdc, and f kHz is input frequency.

**The formulas given are for the typical characteristics only at 25°C.

†Noise immunity specified for worst-case input combination.

Noise Margin for both "1" and "0" level =

0.5 Vdc min @ V_{DD} = 5.0 Vdc

1.0 Vdc min @ V_{DD} = 10 Vdc

1.0 Vdc min @ V_{DD} = 15 Vdc

INTERFACE CIRCUITS

TYPE SN75188 QUADRUPLE LINE DRIVER

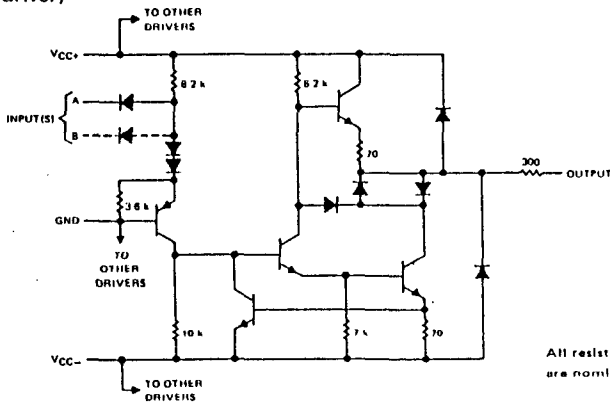
BULLETIN NO. DLS 7711874, SEPTEMBER 1973--REVISED JANUARY 1977

- Meets Specifications of EIA RS-232C
- Designed to be Interchangeable with Motorola MC1488
- Current-Limited Output . . . 10 mA Typical
- Power-Off Output Impedance . . . 300 Ω Min
- Slew Rate Control by Load Capacitor
- Flexible Supply Voltage Range
- Input Compatible with Most TTL and DTL Circuits

description

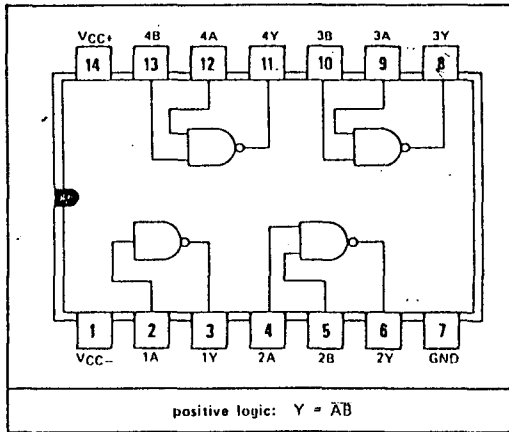
The SN75188 is a monolithic quadruple line driver designed to interface data terminal equipment with data communications equipment in conformance with the specifications of EIA Standard RS-232C with a diode in series with each supply-voltage terminal as shown under typical applications. The device is characterized for operation from 0°C to 75°C.

schematic (each driver)



All resistor values shown are nominal and in ohms.

J OR N
DUAL-IN-LINE PACKAGE
(TOP VIEW)



FUNCTION TABLE

A	B	Y
H	H	L
L	X	H
X	L	H

H = high level, L = low level,
X = irrelevant

absolute maximum ratings over operating free-air temperature range (unless otherwise noted)

Supply voltage V_{CC+} at (or below) 25°C free-air temperature (see Notes 1 and 2)	15 V
Supply voltage V_{CC-} at (or below) 25°C free-air temperature (see Notes 1 and 2)	-15 V
Input voltage range	-15 V to 7 V
Output voltage range	-15 V to 15 V
Continuous total dissipation at (or below) 25°C free-air temperature (see Note 2)	1 W
Operating free-air temperature range	0°C to 75°C
Storage temperature range	-65°C to 175°C
Lead temperature 1/16 inch from case for 60 seconds: J package	300°C
Lead temperature 1/16 inch from case for 10 seconds: N package	260°C

NOTES: 1. All voltage values are with respect to the network ground terminal.

2. For operation above 25°C free-air temperature, refer to the Maximum Supply Voltage Curve, Figure 6, and the Dissipation Derating Curves in the Thermal Information Section, which begins on page 21. In the J package, SN75188 chips are glass-mounted.

TYPE SN75188 QUADRUPLE LINE DRIVER

REVISED JANUARY 1977

electrical characteristics over operating free-air temperature range, $V_{CC+} = 9\text{ V}$, $V_{CC-} = -9\text{ V}$
(unless otherwise noted)

PARAMETER	TEST CONDITIONS	MIN	TYP [†]	MAX	UNIT	
V_{IH} High-level input voltage		1.9			V	
V_{IL} Low-level input voltage			0.8		V	
V_{OH} High-level output voltage	$V_{IL} = 0.8\text{ V}$, $R_L = 3\text{ k}\Omega$	$V_{CC+} = 9\text{ V}$, $V_{CC-} = -9\text{ V}$	6	7	V	
		$V_{CC+} = 13.2\text{ V}$, $V_{CC-} = -13.2\text{ V}$	9	10.5		
V_{OL} Low-level output voltage	$V_{IH} = 1.9\text{ V}$, $R_L = 3\text{ k}\Omega$	$V_{CC+} = 9\text{ V}$, $V_{CC-} = -9\text{ V}$	-7	-6	V	
		$V_{CC+} = 13.2\text{ V}$, $V_{CC-} = -13.2\text{ V}$	-10.5	-9		
I_{IH} High-level input current	$V_I = 5\text{ V}$			10	μA	
I_{IL} Low-level input current	$V_I = 0$			-1	-1.6	mA
$I_{OS(H)}$ Short-circuit output current at high level *	$V_I = 0.8\text{ V}$, $V_O = 0$	-6	-10	-12	mA	
$I_{OS(L)}$ Short-circuit output current at low level *	$V_I = 1.9\text{ V}$, $V_O = 0$	6	10	12	mA	
r_o Output resistance, power off	$V_{CC+} = 0$, $V_{CC-} = 0$, $V_O = -2\text{ V to } 2\text{ V}$	300			Ω	
I_{CC+} Supply current from V_{CC+}	$V_{CC+} = 9\text{ V}$, No load	All inputs at 1.9 V	15	20	mA	
	$V_{CC+} = 12\text{ V}$, No load	All inputs at 1.9 V	19	25		
		All inputs at 0.8 V	5.5	7		
	$V_{CC+} = 15\text{ V}$, No load, $T_A = 25^\circ\text{C}$	All inputs at 1.9 V	34			
All inputs at 0.8 V		12				
I_{CC-} Supply current from V_{CC-}	$V_{CC-} = -9\text{ V}$, No load	All inputs at 1.9 V	-13	-17	mA	
	$V_{CC-} = -12\text{ V}$, No load	All inputs at 1.9 V	-18	-23		
		All inputs at 0.8 V	-0.015			
	$V_{CC-} = -15\text{ V}$, No load, $T_A = 25^\circ\text{C}$	All inputs at 1.9 V	-34			
All inputs at 0.8 V		-2.5				
P_D Total power dissipation	$V_{CC+} = 9\text{ V}$, $V_{CC-} = -9\text{ V}$, No load			333	mW	
	$V_{CC+} = 12\text{ V}$, $V_{CC-} = -12\text{ V}$, No load			576		

[†]All typical values are at $T_A = 25^\circ\text{C}$.

*Not more than one output should be shorted at a time.

NOTE 4: The algebraic convention where the more positive (less negative) limit is designated as maximum is used in this data sheet for logic voltage levels only, e.g., if -6 V is a maximum, the typical value is a more negative voltage.

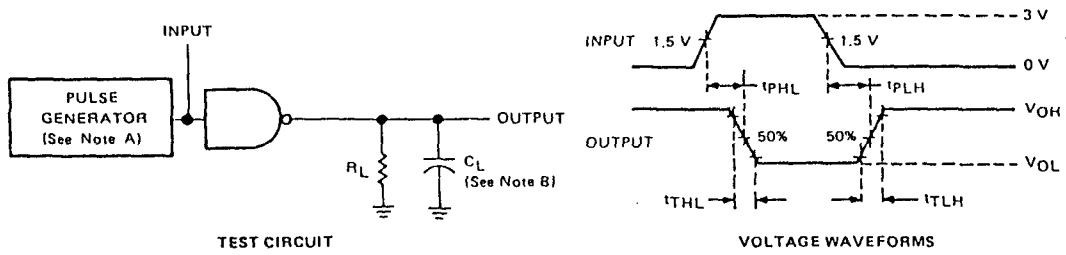
switching characteristics, $V_{CC+} = 9\text{ V}$, $V_{CC-} = -9\text{ V}$, $T_A = 25^\circ\text{C}$

PARAMETER	TEST CONDITIONS	MIN	TYP	MAX	UNIT
t_{PLH} Propagation delay time, low-to-high-level output	$R_L = 3\text{ k}\Omega$, $C_L = 15\text{ pF}$, See Figure 1	220	350		ns
t_{PHL} Propagation delay time, high-to-low-level output		100	175		ns
t_{TLH} Transition time, low-to-high-level output [‡]	See Figure 1	55	100		ns
t_{THL} Transition time, high-to-low-level output [‡]		45	75		ns
t_{TLH} Transition time, low-to-high-level output [§]	$R_L = 3\text{ k}\Omega$ to $7\text{ k}\Omega$, $C_L = 2500\text{ pF}$, See Figure 1	2.5			μs
t_{THL} Transition time, high-to-low-level output [§]		3.0			μs

[‡]Measured between 10% and 90% points of output waveform.

[§]Measured between $+3\text{ V}$ and -3 V points on the output waveform (EIA RS-232C conditions)

PARAMETER MEASUREMENT INFORMATION



NOTE: A. The pulse generator has the following characteristics: $t_w = 0.5 \mu s$, PRR = 1 MHz, $Z_0 = 50 \Omega$.
B. C_L includes probe and jig capacitance.

FIGURE 1—PROPAGATION AND TRANSITION TIMES

TYPICAL CHARACTERISTICS

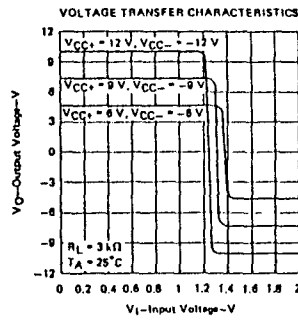


FIGURE 2

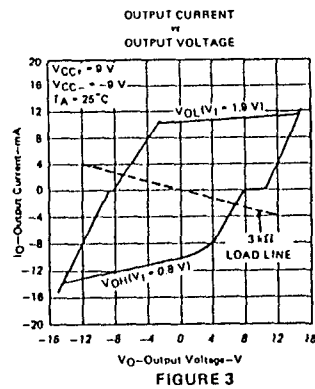


FIGURE 3

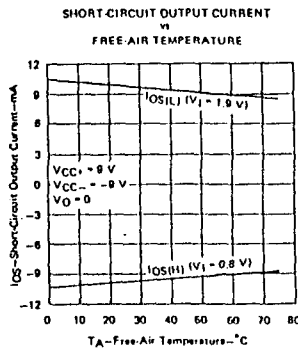


FIGURE 4

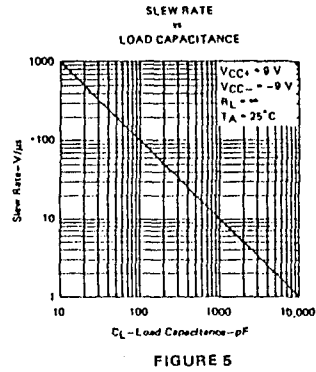


FIGURE 5

TYPE SN75188
QUADRUPLE LINE DRIVER

THERMAL INFORMATION
MAXIMUM SUPPLY VOLTAGE
vs
FREE-AIR TEMPERATURE

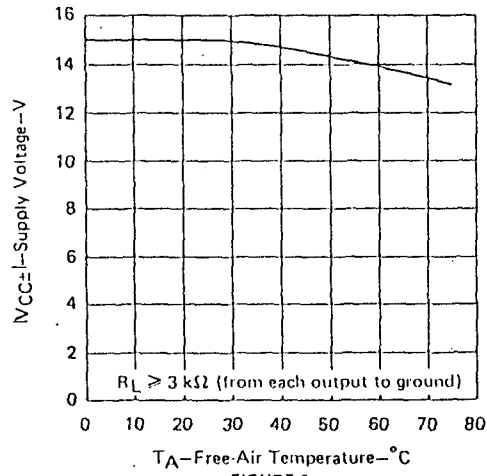


FIGURE 6

TYPICAL APPLICATION DATA

Diodes placed in series with the V_{CC+} and V_{CC-} leads will protect the SN75188 in the fault condition where the device outputs are shorted to $\pm 15 \text{ V}$ and the power supplies are at low voltage and provide low-impedance paths to ground.

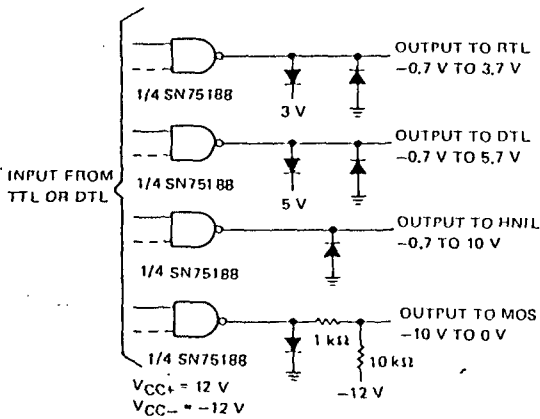


FIGURE 7—LOGIC TRANSLATOR APPLICATIONS

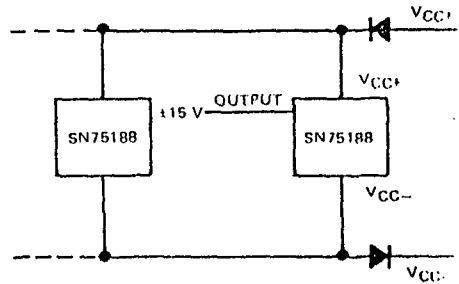
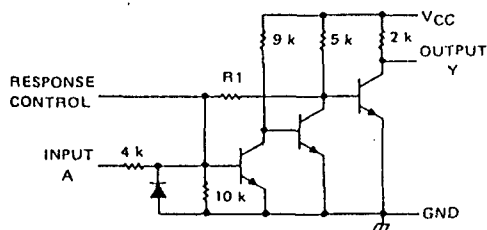


FIGURE 8—POWER SUPPLY PROTECTION TO MFI POWER-OFF FAULT CONDITIONS OF EIA STANDARD RS-232C

- Input Resistance . . . 3 kΩ to 7 kΩ
- Input Signal Range . . . ±30 V
- Fully Interchangeable with Motorola MC1489, MC1489A
- Operates From Single 5-V Supply
- Built-in Input Hysteresis (Double Thresholds)
- Response Control Provides:
 Input Threshold Shifting
 Input Noise Filtering
- Satisfies Requirements of EIA RS-232-C

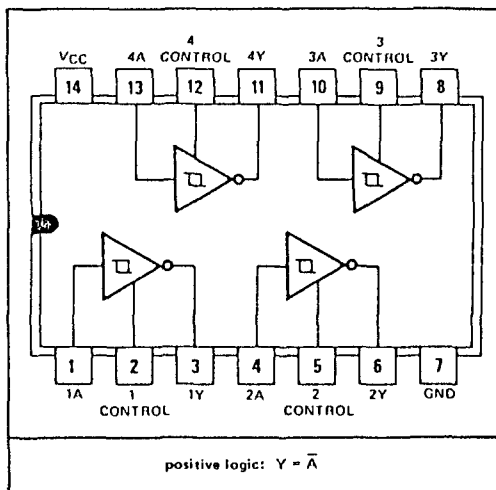
schematic (each receiver)



	SN75189	SN75189A
R1	10 k	2 k

Resistor values shown are nominal and in ohms.

JORN
DUAL-IN-LINE PACKAGE (TOP VIEW)



description

The SN75189 and SN75189A are monolithic quadruple line receivers designed to satisfy the requirements of the standard interface between data terminal equipment and data communication equipment as defined by EIA Standard RS-232C. A separate response control terminal is provided for each receiver. A resistor or a resistor and bias voltage can be connected between this terminal and ground to shift the input threshold voltage levels. An external capacitor can be connected from this terminal to ground to provide input noise filtering.

absolute maximum ratings at 25°C free-air temperature (unless otherwise noted)

Supply voltage, VCC (see Note 1)	10 V
Input voltage	±30 V
Output current	20 mA
Continuous total dissipation at (or below) 25°C free-air temperature (see Note 2)	1 W
Operating free-air temperature range	0°C to 75°C
Storage temperature range	-65°C to 175°C
Lead temperature 1/16 inch from case for 60 seconds: J package	300°C
Lead temperature 1/16 inch from case for 10 seconds: N package	260°C

NOTES: 1. Voltage values are with respect to the network ground terminal.
2. For operation above 25°C free-air temperature, refer to Dissipation Derating Curves in the Thermal Information section, which starts on page 21. In the J package, SN75189 and SN75189A chips are glass-mounted.

TYPES SN75189, SN75189A QUADRUPLE LINE RECEIVERS

electrical characteristics over operating free-air temperature range, $V_{CC} = 5V \pm 1\%$, (unless otherwise noted)

PARAMETER	TEST FIGURE	TEST CONDITIONS†	SN75189			SN75189A			UNIT
			MIN	TYP‡	MAX	MIN	TYP‡	MAX	
V_{T+} Positive-going threshold voltage	1		1	1.5	1.75	1.9	2.25	V	
V_{T-} Negative-going threshold voltage	1		0.75	1.25	0.75	0.97	1.25	V	
V_{OH} High-level output voltage	1	$V_I = 0.75V$, $I_{OH} = -0.5mA$	2.6	4	5	2.6	4	5	V
		Input open, $I_{OH} = -0.5mA$	2.6	4	5	2.6	4	5	V
V_{OL} Low-level output voltage	1	$V_I = 3V$, $I_{OL} = 10mA$	0.2	0.45		0.2	0.45	V	
I_{IH} High-level input current	2	$V_I = 25V$	3.6	8.3	3.6	8.3		mA	
		$V_I = 3V$	0.43	0.43	0.43	0.43		mA	
I_{IL} Low-level input current	2	$V_I = -25V$	-3.6	-8.3	-3.6	-8.3		mA	
		$V_I = -3V$	-0.43	-0.43	-0.43	-0.43		mA	
I_{OS} Short-circuit output current	3		-3		-3			mA	
I_{CC} Supply current	2	$V_I = 5V$, Outputs open	20	26	20	26		mA	

†All characteristics are measured with the response control terminal open.

‡All typical values are at $V_{CC} = 5V$, $T_A = 25^\circ C$.

switching characteristics, $V_{CC} = 5V$, $T_A = 25^\circ C$

PARAMETER	TEST FIGURE	TEST CONDITIONS	MIN	TYP	MAX	UNIT
t_{PLH} Propagation delay time, low-to-high-level output	4	$C_L = 15pF$, $R_L = 3.9k\Omega$		25	85	ns
t_{PHL} Propagation delay time, high-to-low-level output		$C_L = 15pF$, $R_L = 390\Omega$		25	60	ns
t_{TLH} Transition time, low-to-high-level output		$C_L = 15pF$, $R_L = 3.9k\Omega$		120	175	ns
t_{THL} Transition time, high-to-low-level output		$C_L = 15pF$, $R_L = 390\Omega$		10	20	ns

PARAMETER MEASUREMENT INFORMATION§

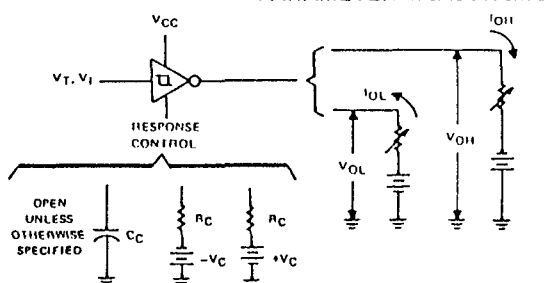


FIGURE 1— V_{T+} , V_{T-} , V_{OH} , V_{OL}

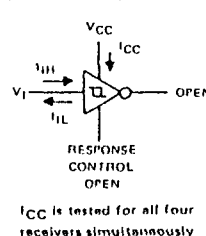


FIGURE 2— I_{IH} , I_{IL} , I_{CC}

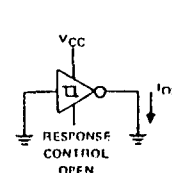
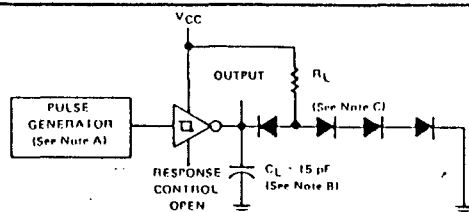


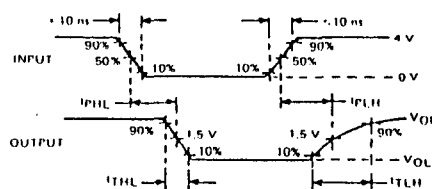
FIGURE 3— I_{OS}



TEST CIRCUIT

- NOTES: A. The pulse generator has the following characteristics: $Z_{out} \approx 50\Omega$, $t_w = 500ns$.
B. C_L includes probe and jig capacitance.
C. All diodes are 1N3064 or equivalent.

FIGURE 4—SWITCHING TIMES

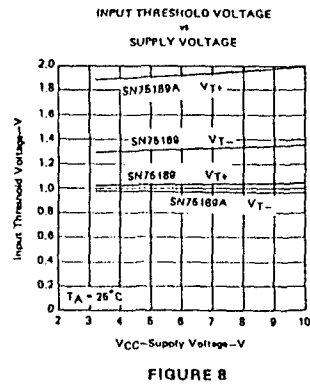
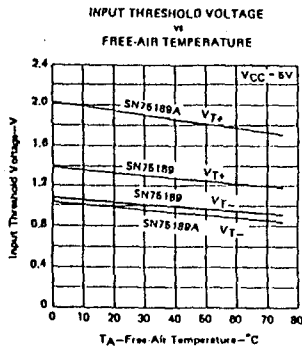
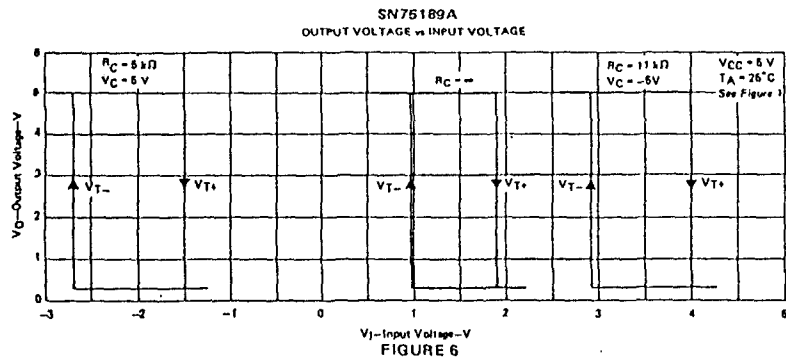
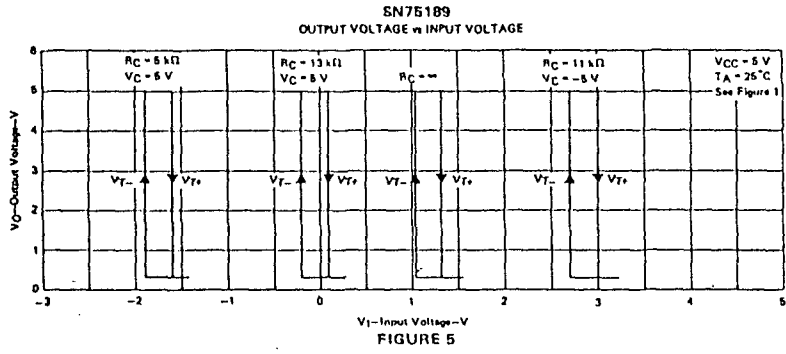


VOLTAGE WAVEFORMS

§Arrows indicate actual direction of current flow. Current into a terminal is a positive value.

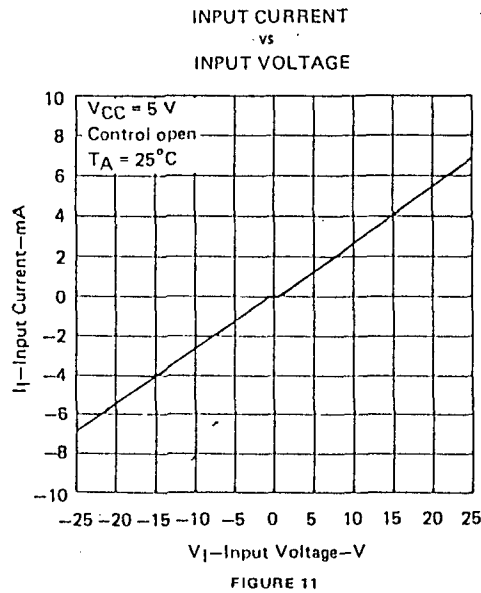
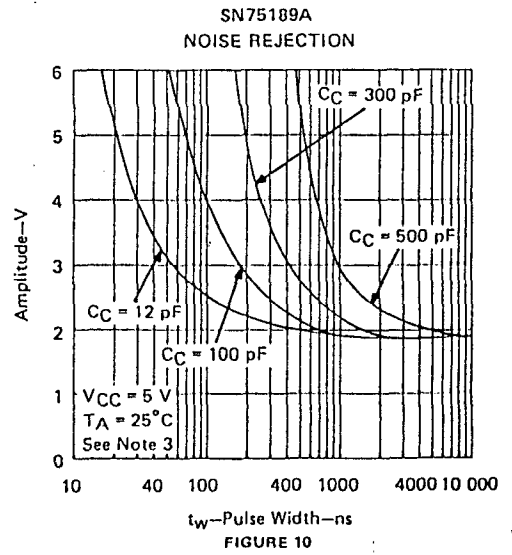
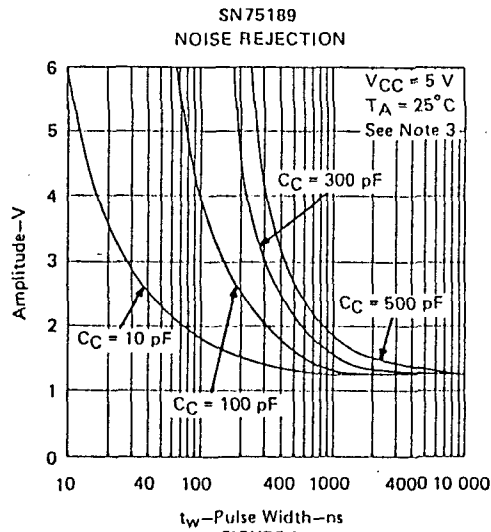
TYPES SN75189, SN75189A QUADRUPLE LINE RECEIVERS

TYPICAL CHARACTERISTICS



TYPES SN75189, SN75189A QUADRUPLE LINE RECEIVERS

TYPICAL CHARACTERISTICS



NOTE 3: This figure shows the maximum amplitude of a positive-going pulse that, starting from zero volts, will not cause a change of the output level.

DISTINCTIVE CHARACTERISTICS

- Complete FSK MODEM in a 28-pin package – just add line interface
- Compatible with Bell 103/113/108, Bell 202, CCITT V.21, CCITT V.23 specifications
- No external filtering required
- All digital signal processing, digital filters and ADC/DAC included on-chip
- Includes essential RS-232/CCITT V.24 handshake signals
- Auto-answer capability
- Local copy/test modes
- 1200 bps full duplex on 4-wire line
- Pin-programmable mode section

GENERAL DESCRIPTION

The Am7910 is a single-chip asynchronous Frequency Shift Keying (FSK) voiceband modem. It is pin selectable for baud rates of 300, 600 or 1200 bits per second and is compatible with the applicable Bell and CCITT recommended standards for 103/113/108, 202, V.21 and V.23 type modems. Five mode control lines select a desired modem configuration.

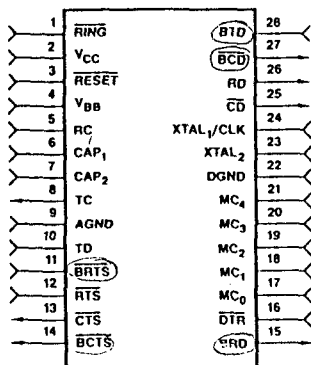
Digital signal processing techniques are employed in the Am7910 to perform all major functions such as modulation, demodulation and filtering. The Am7910 contains on-chip analog-to-digital and digital-to-analog converter circuits to minimize the external components in a system. This device includes the essential RS-232/CCITT V.24 terminal control signals with TTL levels.

Clocking can be generated by attaching a crystal to drive the internal crystal oscillator or by applying an external clock signal.

A data access arrangement (DAA) or acoustic coupler must provide the phone line interface externally.

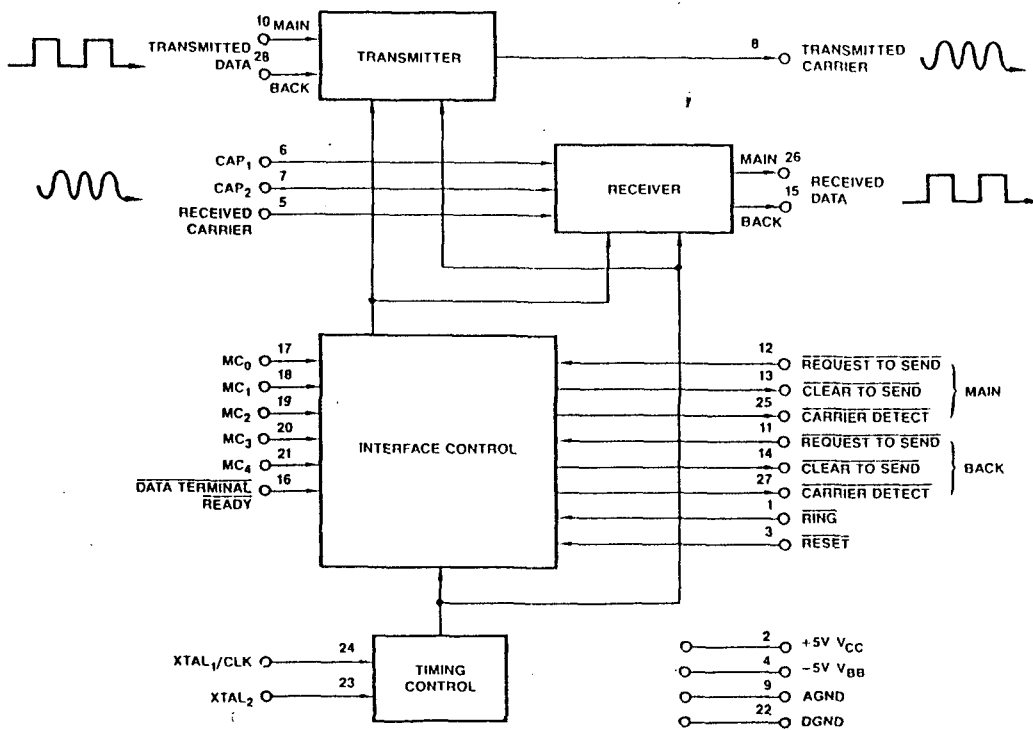
The Am7910 is fabricated using N-channel MOS technology in a 28-pin package. All the digital input and output signals (except the external clock signal) are TTL compatible. Power supply requirements are ± 5 volts.

Figure 1. Connection Diagram



01238C-1

Figure 2. Am7910 Block Diagram



NO INVENTAR

INTERFACE SIGNAL DESCRIPTION

MC₀-MC₄ (CONTROL INPUTS)

These five inputs select one of thirty-two modem configurations according to the Bell or CCITT specifications listed in Table 1. Only 19 of these 32 modes are actually available to the user.

Modes 0-8 are the normal operation modes. The 1200 Baud modes can be selected with or without a compromise equalizer.

Modes 16-25 permit loop back of the Am7910 transmitter and receiver. No internal connection is made. The user must externally connect the TRANSMITTED CARRIER pin (Figure 3) to the RECEIVED CARRIER pin if analog loopback is required. For digital loopback, external connection of RECEIVED DATA and TRANSMITTED DATA is required. Whenever a mode in this group is selected, the effect is to set all transmit and receive filters to the same channel frequency band so that loopback can be performed.

Modes 9-15 and 26-31 are reserved and should not be used.

DATA TERMINAL READY (DTR)

A LOW level on this input indicates the data terminal desires to send and/or receive data via the modem. This signal is gated with all other TTL inputs and outputs so that a low level enables all these signals as well as the internal control logic to function. A HIGH level disables all TTL I/O pins and the internal logic.

REQUEST TO SEND (RTS)

A LOW level on this input instructs the modem to enter transmit mode. This input must remain LOW for the duration of data transmission. The signal has no effect if DATA TERMINAL READY is HIGH (disabled). A HIGH level on this input turns off the transmitter.

CLEAR TO SEND (CTS)

This output goes LOW at the end of a delay initiated when REQUEST TO SEND goes LOW. Actual data to be transmitted should not be presented to the TRANSMITTED DATA input until a LOW is indicated on the CLEAR TO SEND output. Normally the user should force the TD input HIGH whenever CTS is off (HIGH). This signal never goes LOW as long as DTR is HIGH (disabled). CLEAR TO SEND goes HIGH at the end of a delay initiated when REQUEST TO SEND goes HIGH.

CARRIER DETECT (CD)

A LOW on this output indicates that a valid carrier signal is present at the receiver and has been present for at least a time, t_{CDON}, where t_{CDON} depends upon the selected modem configuration (Table 2). A HIGH on this output signifies that no valid carrier is being received and has not been received for a time, t_{CDOFF}. CARRIER DETECT remains HIGH when DTR is HIGH. Values for t_{CDON} and t_{CDOFF} are configuration dependent and are listed in Table 2.

TRANSMITTED DATA (TD)

Data bits to be transmitted are presented on this input serially; HIGH (mark) corresponds to logic 1 and LOW (space) corresponds to logic 0. This data determines which frequency appears at any instant at the TRANSMITTED CARRIER output pin (Table 2). No signal appears at the TRANSMITTED CARRIER output unless DTR is LOW and RTS is LOW.

RECEIVED DATA (RD)

Data bits demodulated from the RECEIVED CARRIER input are available serially at this output; HIGH (mark) indicates logic 1 and LOW (space) indicates logic 0. Under the following conditions this output is forced to logic 1 because the data may

1. When CARRIER DETECT is HIGH
2. During the internal squelch delay at half-duplex line turn around (202/V.23 modes only)
3. During soft carrier turnoff at half-duplex line turn around (202 mode only)
4. When DTR is HIGH
5. When RTS ON and BRTS OFF in V.23/202 modes only
6. During auto-answer sequence

BACK REQUEST TO SEND (BRTS)

Since the 1200 bps modem configurations, Bell 202 and CCITT V.23, permit only half duplex operation over two-wire lines, a low baud rate "backward" channel is provided for transmission from the main channel receiver to the main channel transmitter. This input signal (BRTS) is equivalent to REQUEST TO SEND for the main channel, except it belongs to the backward channel. Note that since the Am7910 contains a single transmitter, RTS and BRTS should not be asserted simultaneously. BRTS is meaningful only when a 202 or V.23 mode is selected by MC₀-MC₄. In all other modes it is ignored.

For V.23 mode the frequency appearing at the transmitted carrier (TC) output pin is determined by a MARK or SPACE at the back transmitted data (BTD) input (Table 2).

For 202 mode a frequency of 387Hz appears at TC when BRTS is LOW and BTD is HIGH. No energy (0.0 volts) appears at TC when BRTS is HIGH. BTD should be fixed HIGH for 202 back channel transmission. The signal, BRTS, then is equivalent to the signal, Secondary Request-to-Send, for 202 S/T modems, or Supervisory Transmitted Data for 202 C/D modems.

BACK CLEAR TO SEND (BCTS)

This line is equivalent to CLEAR TO SEND for the main channel, except it belongs to the back channel. BCTS is meaningful only when a V.23 mode is selected by MC₀-MC₄. This signal is not used in Bell 202 back mode.

BACK CARRIER DETECT (BCD)

This line is equivalent to CARRIER DETECT for the main channel, except it belongs to the backward channel. BCD is meaningful only when a 202 or V.23 mode is selected by MC₀-MC₄. For V.23 back channel mode, BCD turns on when either the MARK or SPACE frequency appears with sufficient level at the received carrier (RC) input.

For 202 back channel mode, BCD turns on in response to a 387Hz tone of sufficient level at the RC input. In this case BCD is equivalent to the signal, Secondary Received Line Signal Detector, for 202 S/T modems, or Supervisory Received Data for 202 C/D modems.

BACK TRANSMITTED DATA (BTD)

This line is equivalent to TRANSMITTED DATA for the main channel, except it belongs to the back channel. BTD is meaningful only when a 202 or V.23 mode is selected by MC₀-MC₄. For 202 back transmission of on/off keying, BTD should be fixed at a HIGH level.

BACK RECEIVED DATA (BRD)

This line is equivalent to RECEIVED DATA (except clamping) for the main channel, except it belongs to the back channel. BRD is meaningful only when a V.23 mode is selected by MC₀-MC₄. Under the following conditions this output is forced HIGH:

1. BRD HIGH
2. DTR HIGH
3. V.21/103 mode
4. During auto-answer
5. When BRTS ON and RTS OFF in V.23 modes only

TRANSMITTED CARRIER (TC)

This analog output is the modulated carrier to be conditioned and sent over the phone line.

RECEIVED CARRIER (RC)

This input is the analog signal received from the phone line. The modem extracts the information contained in this modulated carrier and converts it into a serial data stream for presentation at the RECEIVED DATA (BACK RECEIVED DATA) output.

RING

This input signal permits auto-answer capability by responding to a ringing signal from a data access arrangement. If a ringing

signal is detected ($\overline{\text{RING}}$ LOW) and $\overline{\text{DTR}}$ is LOW, the modem begins a sequence to generate an answer tone at the TC output

XTAL₁, XTAL₂

Master timing of the modem is provided by either a crystal connected to these two inputs or an external clock inserted into XTAL₁. The value of the crystal or the external clock frequency must be 2.4576MHz \pm 0.1%.

V_{CC}

+5 volt power supply. (\pm 5%)

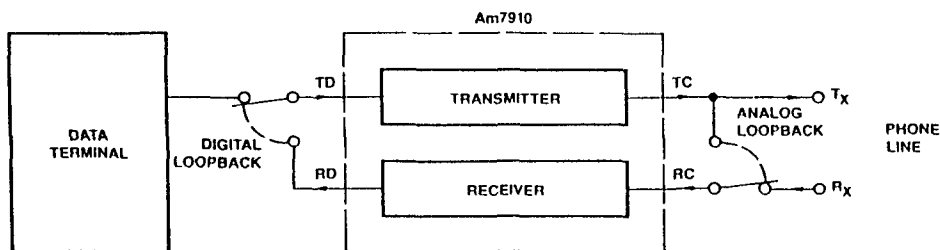
V_{BB}

-5 volt power supply. (\pm 5%)

TABLE 1.

MC ₄	MC ₃	MC ₂	MC ₁	MC ₀	
0	0	0	0	0	Bell 103 Originate 300bps full duplex
0	0	0	0	1	Bell 103 Answer 300bps full duplex
0	0	0	1	0	Bell 202 1200bps half duplex
0	0	0	1	1	Bell 202 with equalizer 1200bps half duplex
0	0	1	0	0	CCITT V.21 Orig 300bps full duplex
0	0	1	0	1	CCITT V.21 Ans 300bps full duplex
0	0	1	1	0	CCITT V.23 Mode 2 1200bps half duplex
0	0	1	1	1	CCITT V.23 Mode 2 with equalizer 1200bps half duplex
0	1	0	0	0	CCITT V.23 Mode 1 600bps half duplex
0	1	0	0	1	Reserved
0	1	0	1	0	
0	1	0	1	1	
0	1	1	0	0	
0	1	1	0	1	
0	1	1	1	0	
0	1	1	1	1	
1	0	0	0	0	
1	0	0	0	1	Bell 103 Ans loopback
1	0	0	1	0	Bell 202 Main loopback
1	0	0	1	1	Bell 202 with equalizer loopback
1	0	1	0	0	CCITT V.21 Orig loopback
1	0	1	0	1	CCITT V.21 Ans loopback
1	0	1	1	0	CCITT V.23 Mode 2 main loopback
1	0	1	1	1	CCITT V.23 Mode 2 with equalizer loopback
1	1	0	0	0	CCITT V.23 Mode 1 main loopback
1	1	0	0	1	CCITT V.23 Back loopback
1	1	0	1	0	Reserved
1	1	0	1	1	
1	1	1	0	0	
1	1	1	0	1	
1	1	1	1	0	
1	1	1	1	1	

Figure 3. Loopback Configurations



DGND

Digital signal ground pin.

AGND

Analog signal ground pin (for TRANSMITTED CARRIER and RECEIVED CARRIER).

CAP₁, CAP₂

Connection points of external capacitor/resistor required for proper operation of on-chip analog-to-digital converter.

Recommended values are: $C = 2000\text{pF} \pm 10\%$,
 $R = 100\Omega \pm 10\%$.

RESET

This input signal is for a reset circuit which operates in either of two modes. It automatically resets when power is applied to the device, or it can be activated by application of an external active low TTL pulse.

THEORY OF OPERATION

The Am7910 MODEM consists of three main sections, shown in the block diagram of Figure 2 – Transmitter, Receiver, and Interface Control.

TRANSMITTER (Modulator)

The transmitter, shown in Figure 4, receives binary digital data from a source such as a UART and converts the data to an analog signal using frequency shift keying (FSK) modulation. This analog signal is applied to the phone line through a DAA or acoustic coupler. FSK is a modulation technique which encodes one bit per baud. A logic one applied to the TRANSMITTED DATA (TD) input causes a sine wave at a given frequency to appear at the analog TRANSMITTED CARRIER (TC) output. A logic zero applied to input TD causes a sine wave of a different frequency to appear at the TC output. As the data at the TD input switches between logical one and zero, the TC output switches between the two frequencies. In the Am7910 this switching between frequencies is phase continuous. The frequencies themselves are digitally synthesized sine functions.

The frequencies for each modem configuration available in the Am7910 are listed in Table 3a.

The process of switching between two frequencies as in FSK generates energy at many more frequencies than the two used in the modulation. All the transmitted information can be recovered from a frequency band B Hz wide, where B is the bit rate or maximum rate of change of the digital data at the TD input. This band is centered about a frequency, f_C ,

$$\text{where } f_C = f_1 + (f_2 - f_1)/2$$

(f_1 = lower of two FSK frequencies)

(f_2 = higher of two FSK frequencies)

In addition to this primary information band, there exist side bands containing redundant information. It is desirable to attenuate these bands for two reasons:

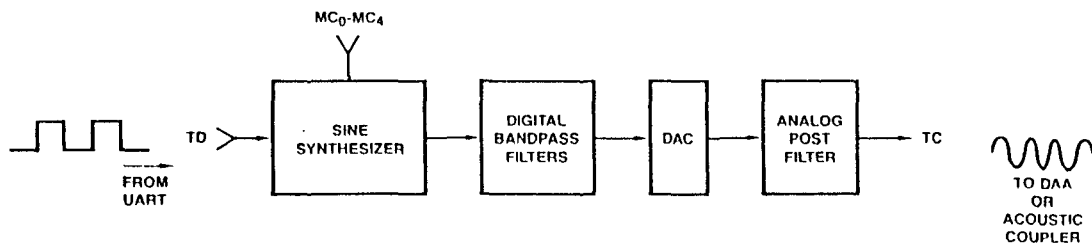
1. The phone companies have specifications on the amount of energy allowed in certain frequency bands on the line.
2. If two independent information channels are present simultaneously on the line (e.g. 300 bps full duplex or 1200 bps half duplex with back), the redundant transmitter components may fall in the frequency band of the local receiver channel and interfere with detection. In the Am7910 these redundant and undesirable components are attenuated by digital bandpass filters.

Following the digital bandpass filters, the filtered FSK signal is converted to an analog signal by an on-chip DAC operating at a high sample rate. This analog FSK signal is finally smoothed by a simple on-chip analog low pass filter.

RECEIVER (Demodulator)

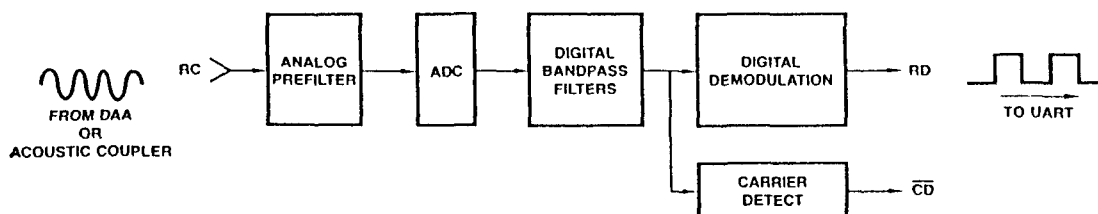
A simplified block diagram of the Am7910 FSK receiver is shown in Figure 5. Data transmitted from a remote site modem over the phone line is an FSK-modulated analog carrier. This carrier is applied to the RECEIVED CARRIER (RC) pin via a DAA or acoustic coupler. The first stage of the demodulator is a simple on-chip analog low pass anti-alias filter. The output of this is converted into digital form and filtered by digital bandpass filters to improve the signal to noise ratio and reject other independent channel frequencies associated with the phone line in the case of full duplex configuration. The bandpass filtered output is digitally demodulated to recover the binary data. A carrier detect signal is also digitally extracted from the received line carrier to indicate valid data.

Figure 4. Transmitter Block Diagram



01238C

Figure 5. Receiver Block Diagram



01238C

INTERFACE CONTROL

This section controls the handshaking between the modem and the local terminal. It consists primarily of delay generation counters, two state machines for controlling transmission and reception, and mode control decode logic for selecting proper transmit frequencies and transmit and receive filters according to the selected modem type. Inputs and outputs from this section are as follows:

REQUEST TO SEND (Main and Back)
CLEAR TO SEND (Main and Back)
CARRIER DETECT (Main and Back)
RING
MCO-MC4
DATA TERMINAL READY

Internal logic clamps protocol signals to different levels under certain conditions (e.g., initial conditions).

When Bell 103/113 and V.21 modem configurations are selected, the back channel signals are non-functional.

Figures 8 and 9 depict the sequencing of the two state machines. State machine 1 implements main or back channel transmission and the auto-answer sequence. State machine 2 implements reception on main or back channel.

The state machine powers on to the state labelled INITIAL CONDITIONS. Handshake signals are set to or assumed to be the levels listed in Table 2. The machine then waits for DATA TERMINAL READY (DTR) to be turned on. Whenever DTR is turned to the OFF state from an ON condition, each state machine and external signals return to the initial conditions within 25 microseconds. After DTR is turned ON the Am7910

becomes operational as a modem and the state machines proceed as depicted in the flowcharts.

The definitions of the terms Full Duplex and Half Duplex used in these flowcharts are depicted below (Figs. 6 and 7). "Full Duplex" applies to all 103/113, V.21 modes. "Half Duplex" applies to 202 and V.23, both forward and backward channel.

Full Duplex: Data can be transmitted and received simultaneously at a rate of 300 baud. Two independent 300Hz channels are frequency multiplexed into the 3000Hz bandwidth of the phone line. The Am7910 configurations for the Bell 103/113 and CCITT V.21 can be operated full duplex.

Half Duplex: In half duplex with back channel, the modem may transmit at 1200/600 baud and receive at 5/75 baud. Alternatively it may transmit at 5/75 baud and receive at 1200/600 baud. Examples are Bell 202 and CCITT V.23.

TABLE 2.
INITIAL CONDITIONS

Data Terminal Ready (\overline{DTR})	OFF
Request to Send (\overline{RTS})	OFF
Clear to Send (\overline{CTS})	OFF
Transmitted Data (TD)	Ignored
Back Channel Request to Send (\overline{BRTS})	OFF
Back Channel Clear to Send (\overline{BCTS})	OFF
Back Channel Transmitted Data (BTD)	Ignored
Ring (\overline{RING})	OFF
Carrier Detect (\overline{CD})	OFF
Received Data (RD)	MARK
Back Channel Carrier Detect (\overline{BCD})	OFF
Back Channel Received Data (BRD)	MARK

Figure 6. Full Duplex

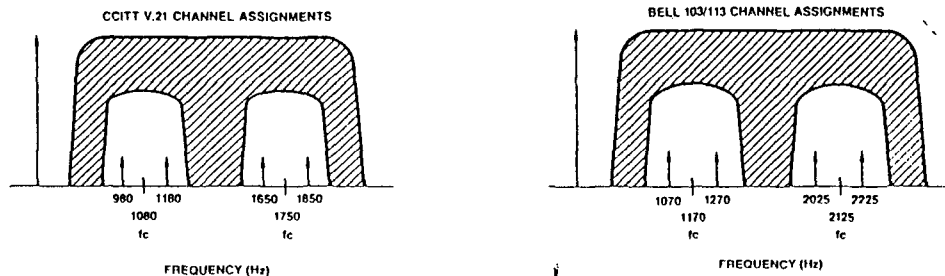
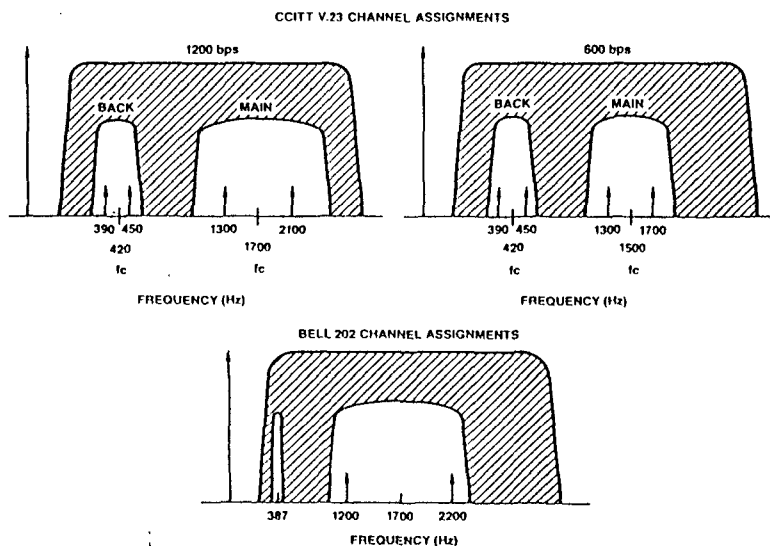


Figure 7. Half Duplex



012

012.

TABLE 3(a). FREQUENCY PARAMETERS

Modem	Baud Rate (BPS)	Duplex	Transmit Frequency		Receive Frequency		Answer Tone Freq Hz
			Space Hz	Mark Hz	Space Hz	Mark Hz	
Bell 103 Orig	300	Full	1070	1270	2025	2225	-
Bell 103 Ans	300	Full	2025	2225	1070	1270	2225
CCITT V.21 Orig	300	Full	1180	980	1850	1650	-
CCITT V.21 Ans	300	Full	1850	1650	1180	980	2100
CCITT V.23 Mode 1	600	Half	1700	1300	1700	1300	2100
CCITT V.23 Mode 2	1200	Half	2100	1300	2100	1300	2100
CCITT V.23 Mode 2 Equalized	1200	Half	2100	1300	2100	1300	2100
Bell 202	1200	Half	2200	1200	2200	1200	2025
Bell 202 Equalized	1200	Half	2200	1200	2200	1200	2025
CCITT V.23 Back	75	-	450	390	450	390	-
Bell 202 Back	5	-	*	*	**	**	-

- * (BRTS LOW) and (BTD HIGH): 387Hz at TC **387Hz at RC: BCD LOW
- * (BRTS HIGH) or (BTD LOW): 0 volts at TC **No 387Hz at RC: BCD HIGH
- * Meets new CCITT R20 frequency tolerance.

Frequency tolerance is less than ±0.4Hz with 2.4576MHz Crystal. Except Bell 202 which is +1Hz (1200 Hz, mark)

TABLE 3(b). TIMING PARAMETERS (Refer to Figures 10, 11 and 12 for Timing Diagrams)

Symbol	Description	Bell 103 Orig	Bell 103 Ans	CCITT V.21 Orig	CCITT V.21 Ans	CCITT V.23 Mode 1	CCITT V.23 Mode 2	CCITT V.23 Mode 2 EQ	Bell 202	Bell 202 EQ	CCITT V.23 Back	Bell 202 Back	Units
t _{RC(On)}	Request-to-Send to Clear-to-Send ON Delay	208.3	208.3	400	400	208.3	208.3	208.3	183.3	183.3	-	-	msec ±0.3%
t _{RC(Off)}	Request-to-Send to Clear-to-Send OFF Delay	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	-	-	msec ±0.25%
t _{BC(On)}	Back Channel Request-to-Send to Clear-to-Send ON Delay	-	-	-	-	-	-	-	-	-	82.3	-	msec ±0.64%
t _{BC(Off)}	Back Channel Request-to-Send to Clear-to-Send OFF Delay	-	-	-	-	-	-	-	-	-	0.5	-	msec ±25%
t _{CD(On)}	Carrier Detect ON Delay	94-106	94-106	301-312	301-312	11.4-15.4	11.4-15.4	11.4-15.4	18-22	18-22	-	-	msec
t _{CD(Off)}	Carrier Detect OFF Delay	21-40	21-40	21-40	21-40	5.4-13.3	5.4-13.3	5.4-13.3	12.4-23.4	12.4-23.4	-	-	msec
t _{BCD(On)}	Back Channel Carrier Detect ON Delay	-	-	-	-	-	-	-	-	-	17-25	17-25	msec
t _{BCD(Off)}	Back Channel Carrier Detect OFF Delay	-	-	-	-	-	-	-	-	-	21-38	21-38	msec
t _{AT}	Answer Tone Duration	-	1.9	-	3.0	3.0	3.0	3.0	1.9	1.9	-	-	sec ±0.44%
t _{SIL}	Silence Interval before Transmission	1.3	1.3	1.9	1.9	1.9	1.9	1.9	1.3	1.3	-	-	sec ±0.64%
t _{SQ}	Receiver Squelch Duration	-	-	-	-	156.3	156.3	156.3	156.3	156.3	-	-	msec ±3.3%
t _{STO}	Transmitter Soft Turn-Off Duration	-	-	-	-	-	-	-	24	24	-	-	msec ±2.3%
t _{RI}	Minimum RI Low Duration	-	25	-	25	25	25	25	25	25	-	-	µs

CALL ESTABLISHMENT

Before two modems can exchange data, an electrical connection through the phone system must be established. Although it may assist in call establishment, a modem typically does not play a major role. A call may be originated manually or automatically and it may be answered manually or automatically.

Manual Calling – Manual calling is performed by a person who dials the number, waits for an answer, then places the calling modem into data transmission mode.

Automatic Calling – Automatic calling is typically performed by an automatic calling unit (ACU) which generates the appropriate dialing pulse or dual-tone sequence required to call the remote (called) modem. The ACU also has the ability to detect an answer tone from the called modem and place the calling modem into data transmission mode.

Manual Answering – Manual answering is performed by a person who hears the phone ring, lifts the receiver, causes the called modem to send an answer tone to the calling modem, and places the called modem into data transmission mode.

Automatic Answering – Automatic answering is performed by a called modem with a data access arrangement (DAA). The DAA detects a ringing signal, takes the phone circuit off-hook (corresponding to lifting the receiver) and instructs the called modem to commence the auto-answer sequence. Next the called modem sends out silence on the line, followed by an answer tone. When this tone is detected by the calling modem, the connection is considered to have been established.

The Am7910 provides assistance for automatic answering through the RING signal as follows. Observe the upper right-hand portion of Figure 8(a). Assume that DATA TERMINAL READY (DTR) has recently been asserted to cause exit from the initial conditions. Note that if DTR remains OFF, RING is ignored. Assume also that \overline{RTS} and \overline{BRTS} are OFF and that the mode control lines (MCO-MC4) select a normal modem configuration, not a loopback mode. Automatic answering is initiated by receipt of a LOW level at the RING input, causing entrance to the auto-answer sequence depicted in Figure 8(c).

The Am7910 outputs silence (0.0 volts) at its TRANSMITTED CARRIER (TC) output for a time, t_{SIL} , followed by the answer tone for a time, t_{AT} . The CARRIER DETECT (CD) pin is clamped OFF and the RECEIVED DATA (RD) signal is therefore clamped to a MARK (HIGH) during the auto-answer sequence. Upon completion of the answer tone, CD is released. If the mode lines (MCO-MC4) select a 202 or V.23 mode, the transmit filters are set to the forward channel and the receive filters are set to the back channel during the auto answer sequence.

At the end of the auto-answer sequence, return is made to point A in the loop at the upper right-hand portion of Figure 8(a). Note that since the answer flag has been set, the auto-answer sequence cannot be entered again unless DTR is first turned OFF, then ON. At this point the phone line connection has been established and data transmission or reception may begin.

The RING input may be activated from a conditioned DAA Ring Indicator output for automatic answering or it may be activated by a switch for manual answering. Tying RING HIGH will disable the auto-answer function of the Am7910.

DATA TRANSMISSION

Full Duplex

Following call establishment, full duplex data transmission can be started by either the called or calling modem. In other words, if the connection has been established and the modem is looping through point A in Figure 8(a), it no longer matters which is the

called and which is the calling modem. Data transmission is initiated by asserting REQUEST TO SEND (RTS). At this time the TRANSMITTED DATA (TD) input will be released and a modulated carrier can appear at the TRANSMITTED CARRIER (TC) output. Following a delay, t_{RCON} , CLEAR TO SEND (CTS) will turn ON. At this time, data may be transmitted through the TC input. It is a common protocol for the user to always present a MARK at the TD input before \overline{RTS} is asserted and during the t_{RCON} delay.

Data transmission continues until \overline{RTS} is turned OFF. Following a short delay, t_{RCOFF} , CTS turns OFF. As soon as \overline{RTS} goes OFF the TD input is ignored and the TC output is set to 0.0 volts (silence). After CTS turns OFF, the state machine returns to point A in Figure 8(a).

Half Duplex

When a half duplex mode is selected (202 or V.23), data transmission can be either on the main channel at 1200/600 baud or on the back channel at 5/75 baud. In normal half duplex operation a single modem is either transmitting on the main and receiving on the back channel or vice versa. In the Am7910 control of the transmitter and receiver filters to the proper channel is performed by \overline{RTS} . When \overline{RTS} is asserted, the transmitter filters and synthesizer are set to transmit on the main channel; the receive filters are set to receive on the back channel. Therefore whenever \overline{RTS} is on, \overline{BRTS} should not be asserted since the transmitter cannot be used for the back channel. When \overline{RTS} is OFF and a half duplex mode is selected, the transmitter filters and synthesizer are set to the back channel; the receiver filters are set to the main channel. If \overline{RTS} and \overline{BRTS} are asserted simultaneously, \overline{RTS} will take precedence. However, if \overline{BRTS} is asserted before \overline{RTS} and the back channel data transmission sequence has been entered (Figure 8(b)), \overline{RTS} will be ignored until \overline{BRTS} is turned OFF.

The state machine sequences for main and back channel transmission differ slightly and are depicted in Figure 8. Assume the state machine is idling through point A in Figure 8(a).

Main Channel

This transmission sequence is entered if a 202 or V.23 mode is selected and \overline{RTS} is asserted. Since the receiver is now forced to the back channel, the RECEIVED DATA (RD) signal is clamped to a MARK; and the CARRIER DETECT signal is clamped OFF. The TRANSMITTED DATA input (TD) is released and a carrier appears at the TRANSMITTED CARRIER output which follows the MARK/SPACE applied to TD. \overline{RTS} turning ON initiates a delay, t_{RCON} , at the end of which the CLEAR TO SEND (CTS) output goes LOW. When CTS goes LOW data may be transmitted through input TD. Data transmission continues until \overline{RTS} is turned OFF. At this time several events are initiated. First a delay, t_{RCOFF} , is initiated at the end of which CTS turns OFF. The TD input is ignored as soon as \overline{RTS} goes OFF. If a 202 mode is selected, a soft turn-off tone appears at the TC output for a time t_{STO} , followed by silence (0.0 volts). For both 202 and V.23 modes a squelch period, t_{SQ} , is initiated when \overline{RTS} goes OFF. During this period the CD output is clamped OFF, forcing the RD output to a MARK condition. The squelch period begins as soon as \overline{RTS} goes OFF and thus overlaps both t_{RCOFF} and t_{STO} . At the end of the squelch period, the state machine returns to the idling loop at point A in Figure 8(a).

The reasons for squelch and soft-turnoff are as follows:

Soft Turn-Off: When \overline{RTS} is turned OFF at the end of a message transients occur which may cause spurious space signals to be received at a remote modem. During soft turn-off the modem transmits a soft carrier frequency for a period, t_{STO} , after \overline{RTS} is

turned OFF. This results in a steady MARK on the RECEIVED DATA (RD) line of the remote modem.

Squelch: The local receiver must be turned OFF after $\overline{\text{RTS}}$ is OFF, until the start of carrier detect, so that line transients are not demodulated. The process of disabling the receiver after $\overline{\text{RTS}}$ is turned OFF is called squelching.

Back Channel

This transmission sequence, shown in Figure 8(b), is entered if a 202 or V.23 mode is selected, $\overline{\text{RTS}}$ is OFF, and $\overline{\text{BRTS}}$ is asserted. The BACK CARRIER DETECT ($\overline{\text{BCD}}$) output is forced OFF and the BACK RECEIVED DATA (BRD) output is clamped to a MARK. The BACK TRANSMITTED DATA input (BTD) is released and a carrier appears at the TC output which follows the MARK/SPACE applied to BTD. Turning ON $\overline{\text{BRTS}}$ initiates a delay, t_{BRCON} , at the end of which the BACK CLEAR TO SEND ($\overline{\text{BCTS}}$) output goes LOW. When $\overline{\text{BCTS}}$ goes LOW data may be transmitted through input BTD. Data transmission continues until $\overline{\text{BRTS}}$ is turned OFF. The input BTD is immediately ignored and the TC output is silenced (set to 0.0 volts). Following a short delay, t_{BRCOFF} , the output $\overline{\text{BCTS}}$ goes OFF. The signals $\overline{\text{BCD}}$ and BRD are released and the state machine returns to idle at point A of Figure 8(a).

In 202 back channel mode, BTD should be tied HIGH. Then $\overline{\text{BRTS}}$ controls the ON/OFF keying modulation. When $\overline{\text{BRTS}}$ is LOW, 387Hz appears at the TC output; when $\overline{\text{BRTS}}$ is HIGH, 0 volts appears at TC.

DATA RECEPTION

Data reception is controlled by state machine 2 and depicted in Figure 9. At power on the machine enters initial conditions and remains there until DTR is asserted. It then loops until either CARRIER DETECT ($\overline{\text{CD}}$) or BACK CARRIER DETECT ($\overline{\text{BCD}}$) occurs.

Full Duplex

In full duplex data reception, CARRIER DETECT may appear at any time after the phone connection has been established. Reception is independent of transmission. When the receiver detects a valid carrier for at least a time, t_{CDON} , the output $\overline{\text{CD}}$ is turned ON, the RECEIVED DATA (RD) output is released, and valid data can be obtained at RD. Data is received until the receiver detects loss of carrier for at least a time, t_{CDOFF} . At this time the $\overline{\text{CD}}$ output is turned OFF and RD is clamped to a MARK. The state machine returns to the idle loop at point E.

Half Duplex

As discussed in the data transmission section above, when a half duplex mode has been selected, the signal $\overline{\text{RTS}}$ controls whether the main channel is transmitting or receiving. The back channel can only do the opposite from the main. If $\overline{\text{RTS}}$ is OFF, then CARRIER DETECT may be asserted and the data reception sequence is identical to that discussed above for full duplex reception. As long as $\overline{\text{RTS}}$ remains OFF, BACK CARRIER DETECT will never be asserted. If $\overline{\text{RTS}}$ is ON, then CARRIER DETECT will never be asserted. Instead the receiver will look for a valid carrier in the back channel frequency band. If a valid carrier exists for at least a time, t_{BCDON} , the output BACK CARRIER DETECT ($\overline{\text{BCD}}$) is turned ON, the BACK RECEIVED DATA (BRD) output is released and valid data can be obtained at BRD. Data is received until the receiver detects loss of back channel received signal for at least time, t_{BCDOFF} . At this time the $\overline{\text{BCD}}$ output is turned OFF. Data output, BRD, is clamped to a MARK if a V.23 mode is selected. For 202 back channel mode, $\overline{\text{BCD}}$ represents the received data. The BRD output can be ignored. The state machine returns to the idle loop at point E.

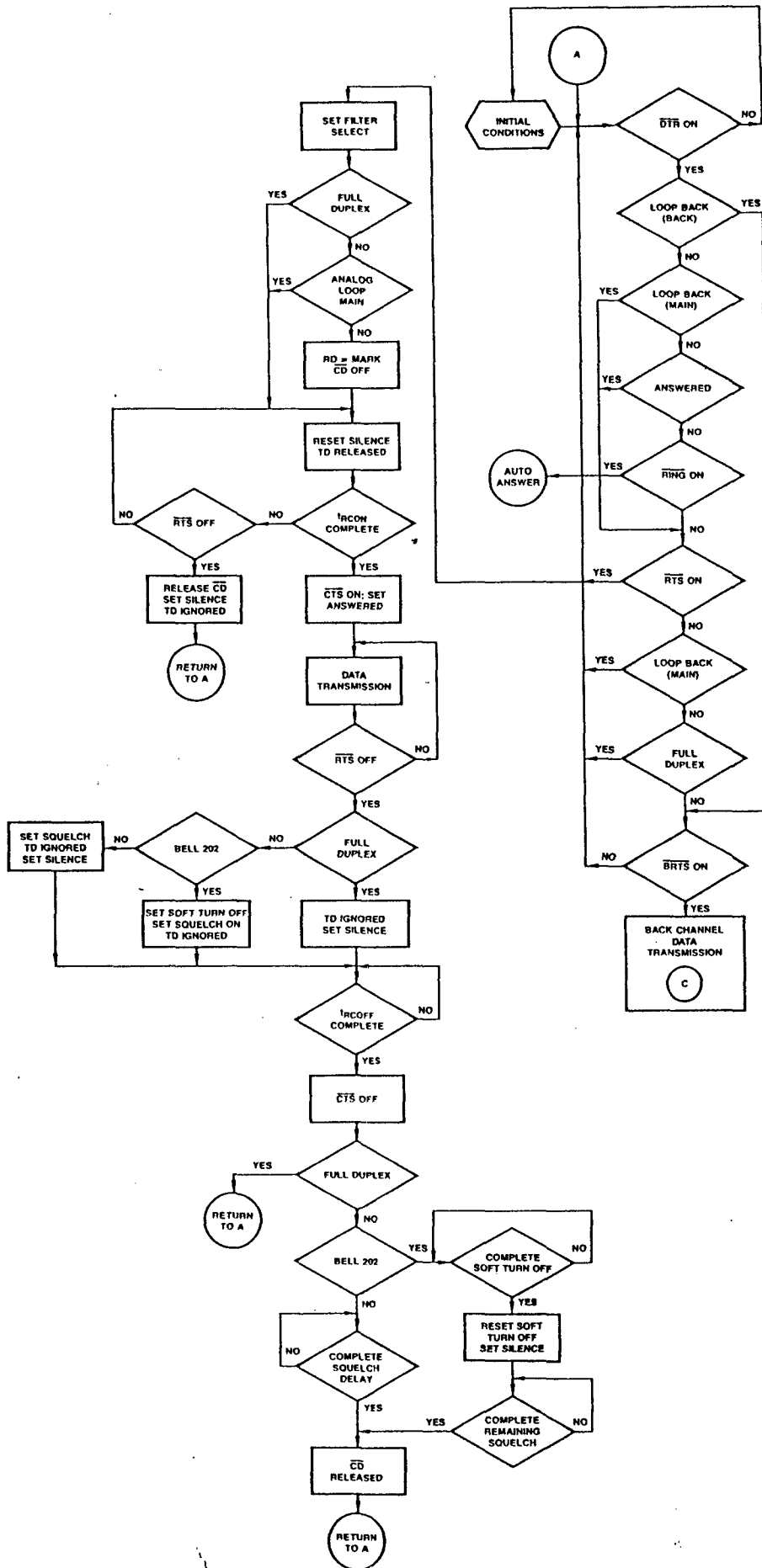
LOOPBACK

Ten modes exist to allow both analog and digital loopback for each modem specification met by the Am7910. When a loopback mode is selected, the signal processing (filters, etc.) for both the transmitter and receiver is set to process the same channel or frequency band. This allows the analog output, TRANSMITTED CARRIER, and the analog input, RECEIVED CARRIER, to be connected for local analog loopback. Alternatively the digital data signals, TD and RD or BTD and BRD, can be connected externally, allowing a remote modem to test the local modem with its digital data signals looped back.

When a loopback mode is selected, the state machine sequences are altered slightly. First, auto-answer is disabled. Second, if a half duplex loopback mode is selected (202 or V.23), the local CARRIER DETECT/ $\overline{\text{BCD}}$ is not forced OFF when $\overline{\text{RTS}}/\overline{\text{BRTS}}$ is asserted.

The 202 and V.23 main loopback modes allow use in a 4-wire configuration at 1200 bps.

Figure 8(a). Transmit Main Channel State Diagram



9

Figure 8(b). Transmit Back Channel State Diagram

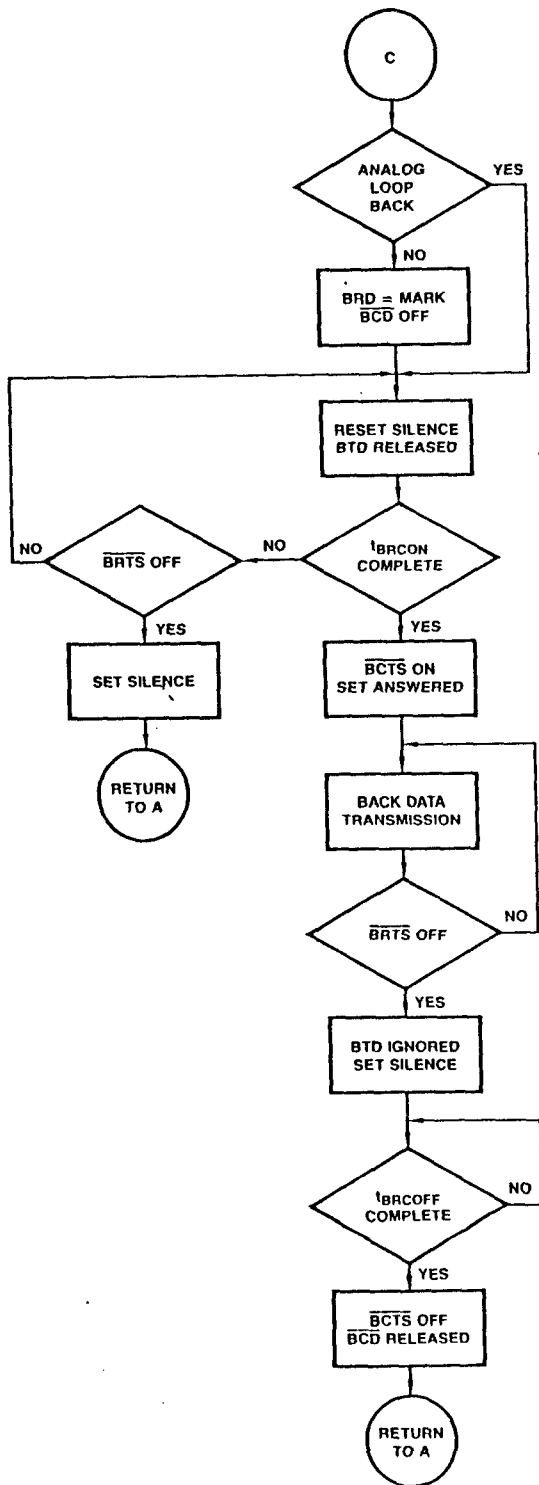
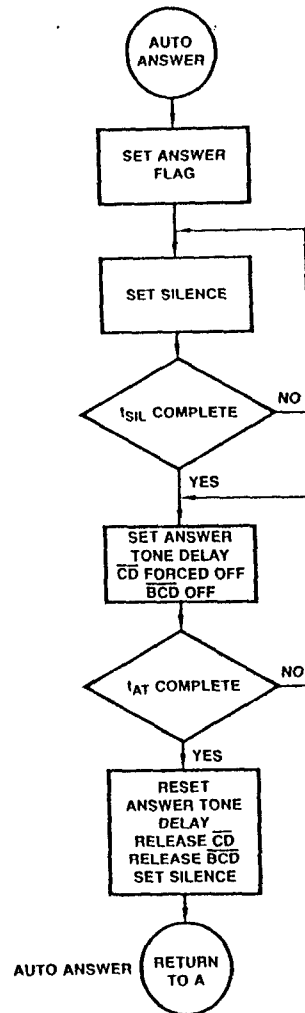
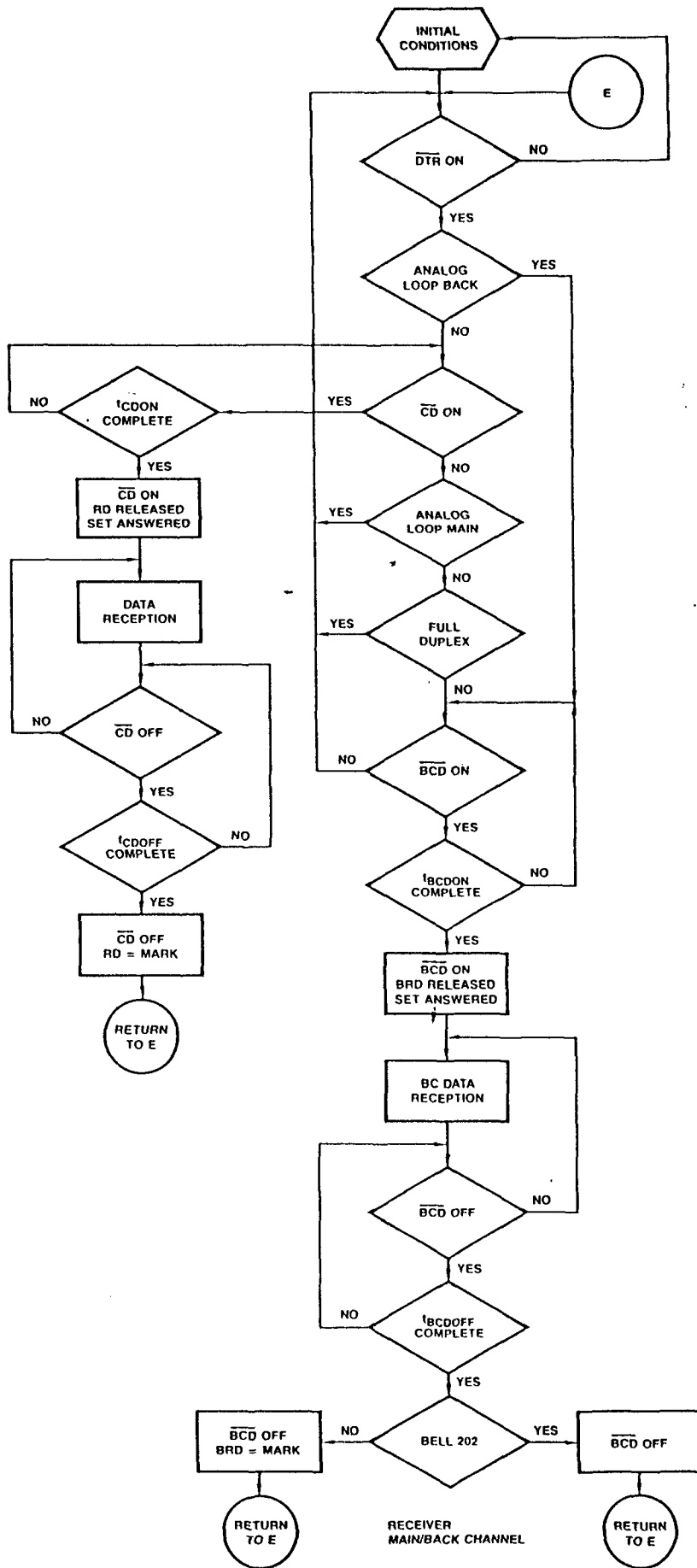


Figure 8(c). Auto Answer State Diagram



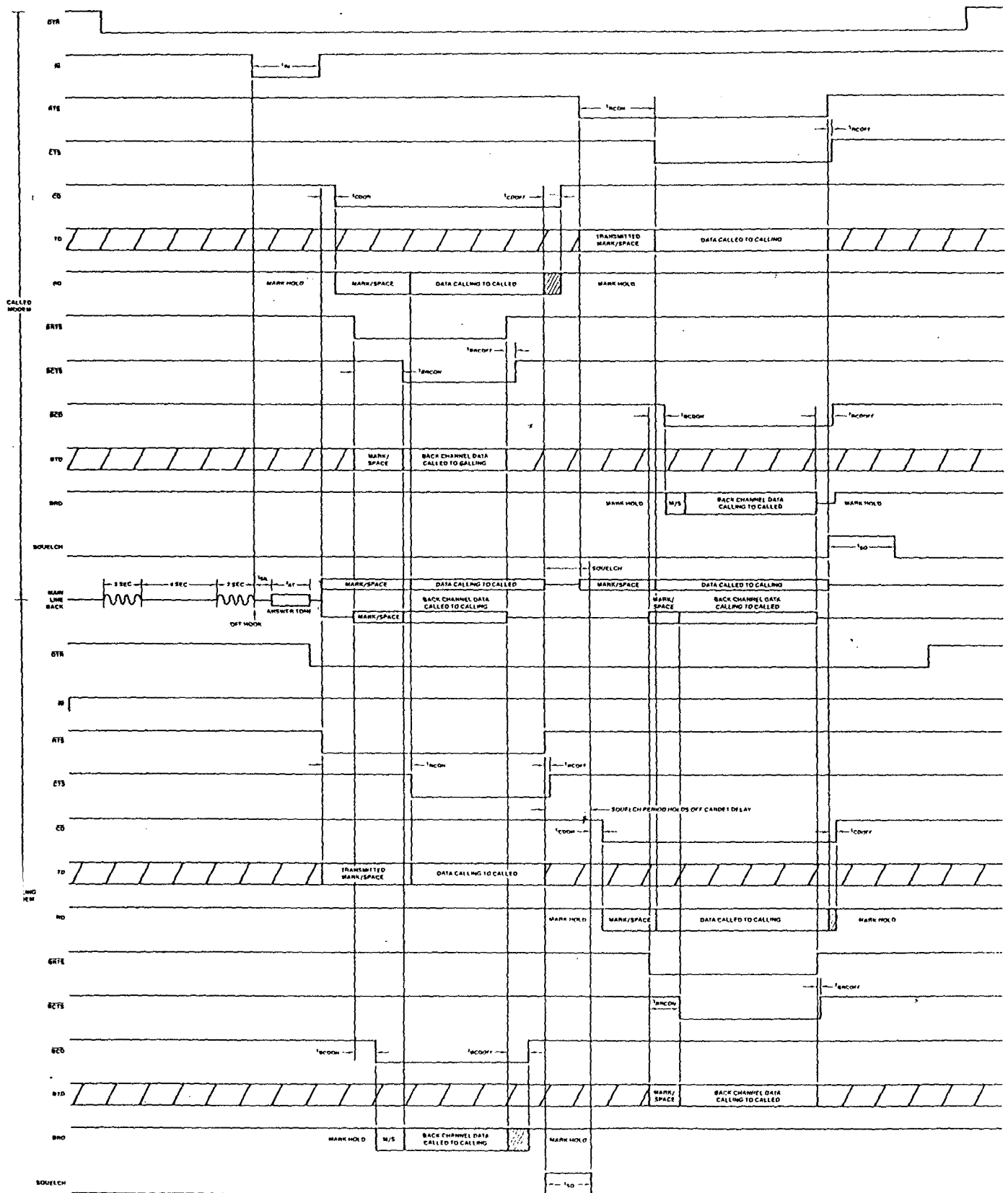
16

Figure 9. Receiver Main/Back Channel State Diagram



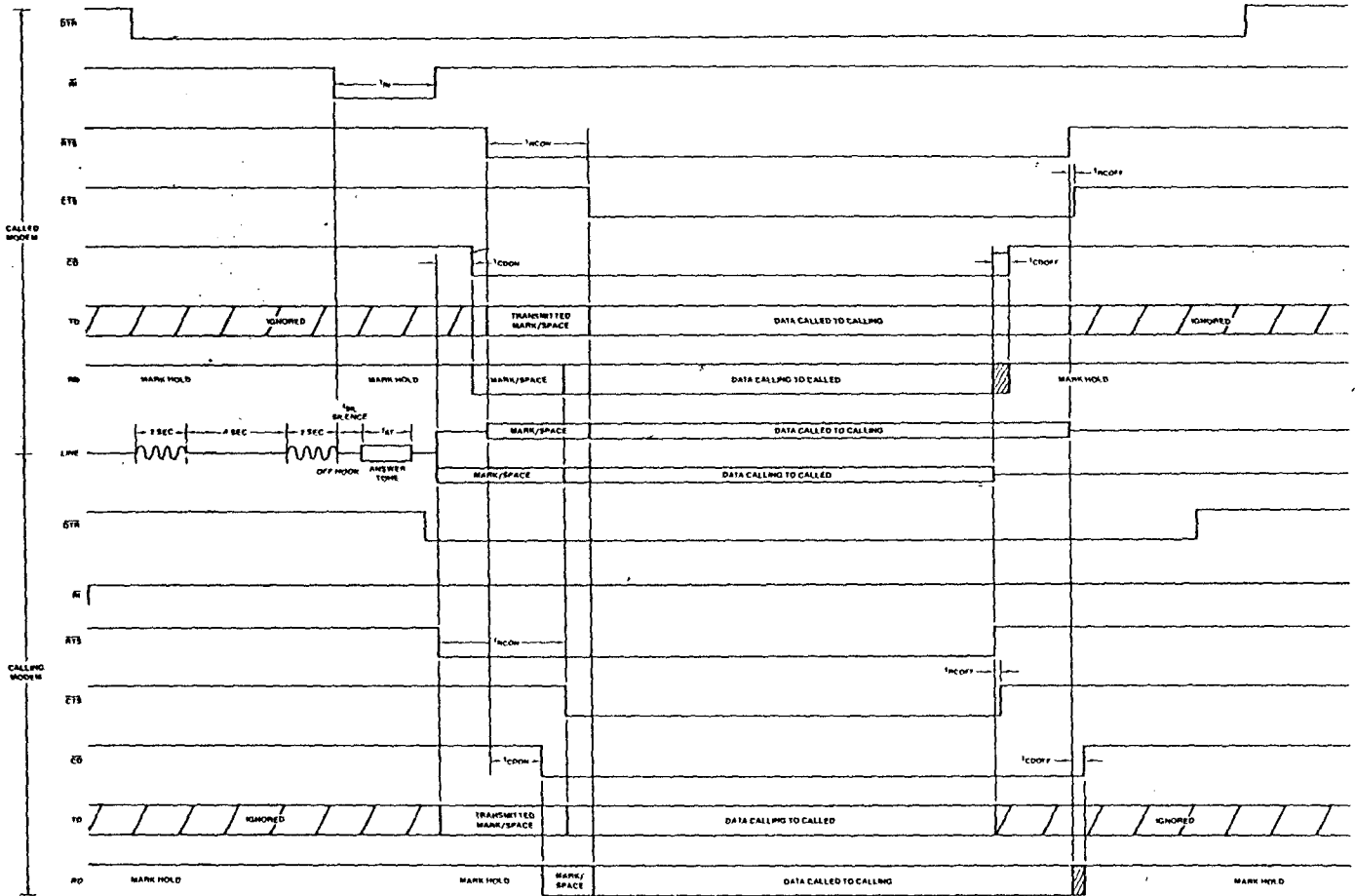
© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2008

Figure 11. CCITT V.23 Handshake Timing



© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2006

Figure 12. BELL 103/CCITT V.21 Handshake Timing



© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2008

CLOCK GENERATION

Master timing of the modem is provided by either a crystal connected to the XTAL₁ and XTAL₂ inputs or an external clock applied to the XTAL₁ input.

Crystal

When a crystal is used it should be connected as shown in Figure 13. The crystal should be a parallel resonance type, and its value must be 2.4576MHz ±.01%. A list of crystal suppliers is shown below.

External Clock

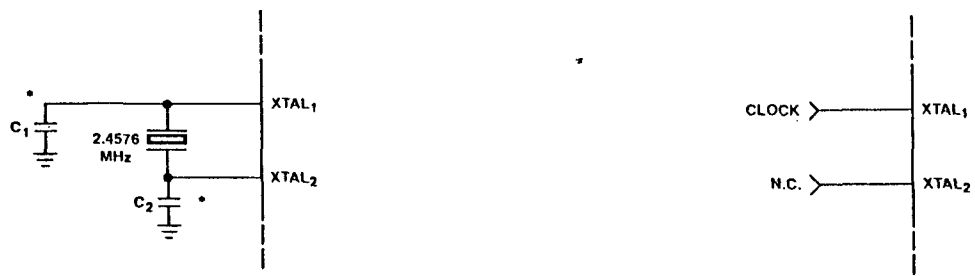
This clock signal could be derived from one of several crystal-driven baud rate generators. It should be connected to the XTAL₁ input and the XTAL₂ input must be left floating. The timing parameters required of this clock are shown in Figure 13 and the values are listed in Table 4.

Figure 13. Clock Generation

Crystal Information (f_C = 2.4576MHz)

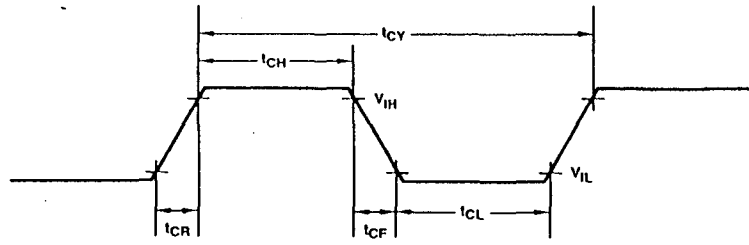
Manufacturer	P/N	C ₁	C ₂
M-Tron	MP-2	20pF	20pF
Monitor Products	MM-33	20pF	20pF

Note: Rise time of V_{CC} must be greater than 5msec to insure proper crystal oscillator start-up.



*Capacitors values vary with different crystal manufacturers.

(a)



(b)

01238C-15

TABLE 4. CLOCK PARAMETERS

Symbol	Parameters	Min	Typ	Max	Units
t _{CY}	Clock Period	406.86	406.9	406.94	ns
t _{CH}	Clock High Time	165			ns
t _{CL}	Clock Low Time	165			ns
t _{CR}	Clock Rise Time			20	ns
t _{CF}	Clock Fall Time			20	ns

PRELIMINARY

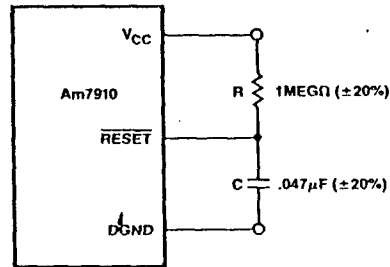
POWER ON RESET

The reset circuit operates in either of two modes.

Automatic Reset

In this mode an internal reset sequence is automatically entered when power is applied to the device. One resistor and one capacitor must be connected externally as shown in Figure 14. Values shown will work with most power supplies. Power supply (V_{CC}) rise time should be less than one half the RC time constant.

Figure 14. Automatic Reset



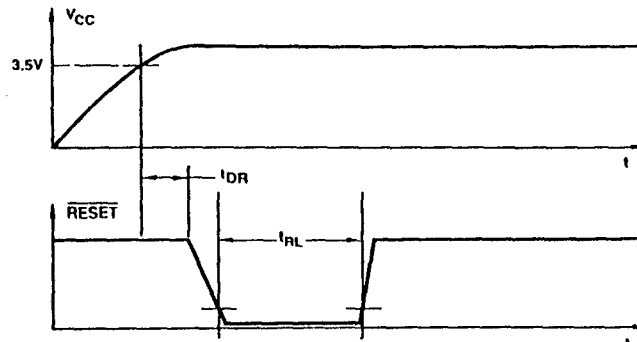
01238C-15

External Reset

In this mode the device may be forced into the reset sequence by application of an active LOW pulse applied to the RESET input. The reset must not be applied until the V_{CC} supply has reached at least 3.5V. Timing is diagrammed in Figure 15.

Figure 15. External Reset

TIMING DIAGRAMS



t_{DR} = delay from the time V_{CC} reaches 3.5V and the falling edge of RESET signal ($>1\mu s$)

t_{RL} = RESET LOW duration time ($>t_{MCK} = 406ns$)

01238C-17

**NOMINAL PERFORMANCE SPECIFICATIONS
TRANSMITTER (All Modem Types)**

Input Data Format: Serial, asynchronous, standard TTL levels
Modulation Technique:

Binary, phase-coherent Frequency Shift Keying (FSK)

TC Output Level: -3dBm into 600Ω

Frequency Accuracy:

- ±0.4Hz all modems except Bell 202 (mark)
- +1.0Hz Bell 202 (mark)

Harmonics: -45dB from fundamental for single tones

Delay uncertainty for TD logic input change to TC frequency change: ≤8.3μs

Out-of-band energy: See Figure 16

RECEIVER

Output Data Format: Serial, asynchronous, TTL levels

Demodulation Technique: Differential FM Detection

Sensitivity at Receiver Input: 0dBm to -48dBm

Frequency Deviation Tolerance: ±16Hz

Carrier Detect Threshold:

- ON >-43dBm ±1dB
- OFF <-48dBm ±1dB

TEST MEASUREMENT SETUP

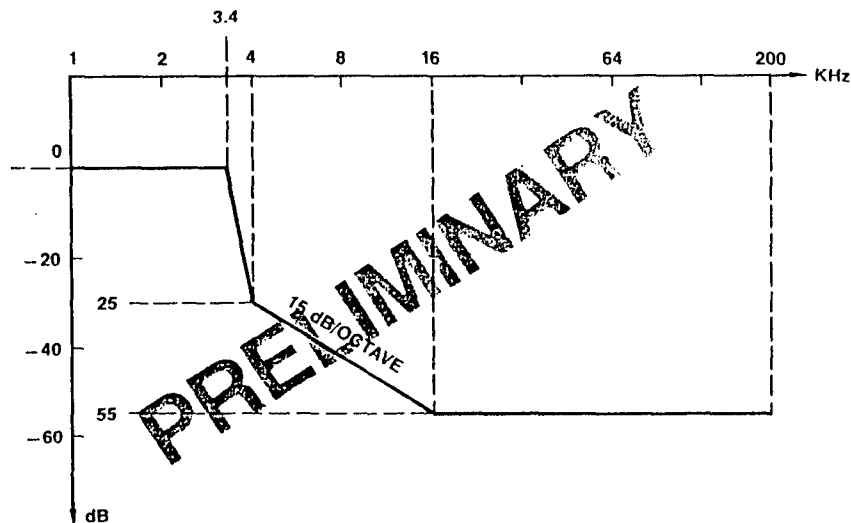
Am7910 performance is characterized using the test equipment setup shown in Figure 17. The HP1645A data error analyzer is used to generate 511-bit pseudo random binary sequences (PRBS) at D_{OUT} for testing the modem. The 1645A also receives and analyzes the 511-bit digital pattern at D_{IN} after it has progressed around the test loop. A reference transmitter

converts the digital sequence generated by the HP1645A into an FSK signal. The FSK signal is typically adjusted to different levels from -12 to -45dBm. The level-adjusted FSK signal or incident signal then passes through three pieces of equipment which comprise the telephone line simulator. The Wandel and Golterman TLN-1 and DLZ-4 simulate amplitude and group delay characteristics typical of a wide variety of phone lines. Line perturbations, such as amplitude hits and phase hits, may be injected by the Bradley 2A/2B.

The summing amplifier which drives the modem under test has three inputs. One of these inputs is the incident FSK signal which has been passed through a simulated phone line. The second input is from an optionally filtered noise source in order to simulate noise conditions which may be encountered on phone lines. The third input is from the transmitter of the Am7910 under test. This third input simulates the adjacent channel signal seen at the input of the Am7910 receiver due to the duplexer used on 2-wire lines. If 4-wire testing is being performed, the adjacent channel would not normally be included.

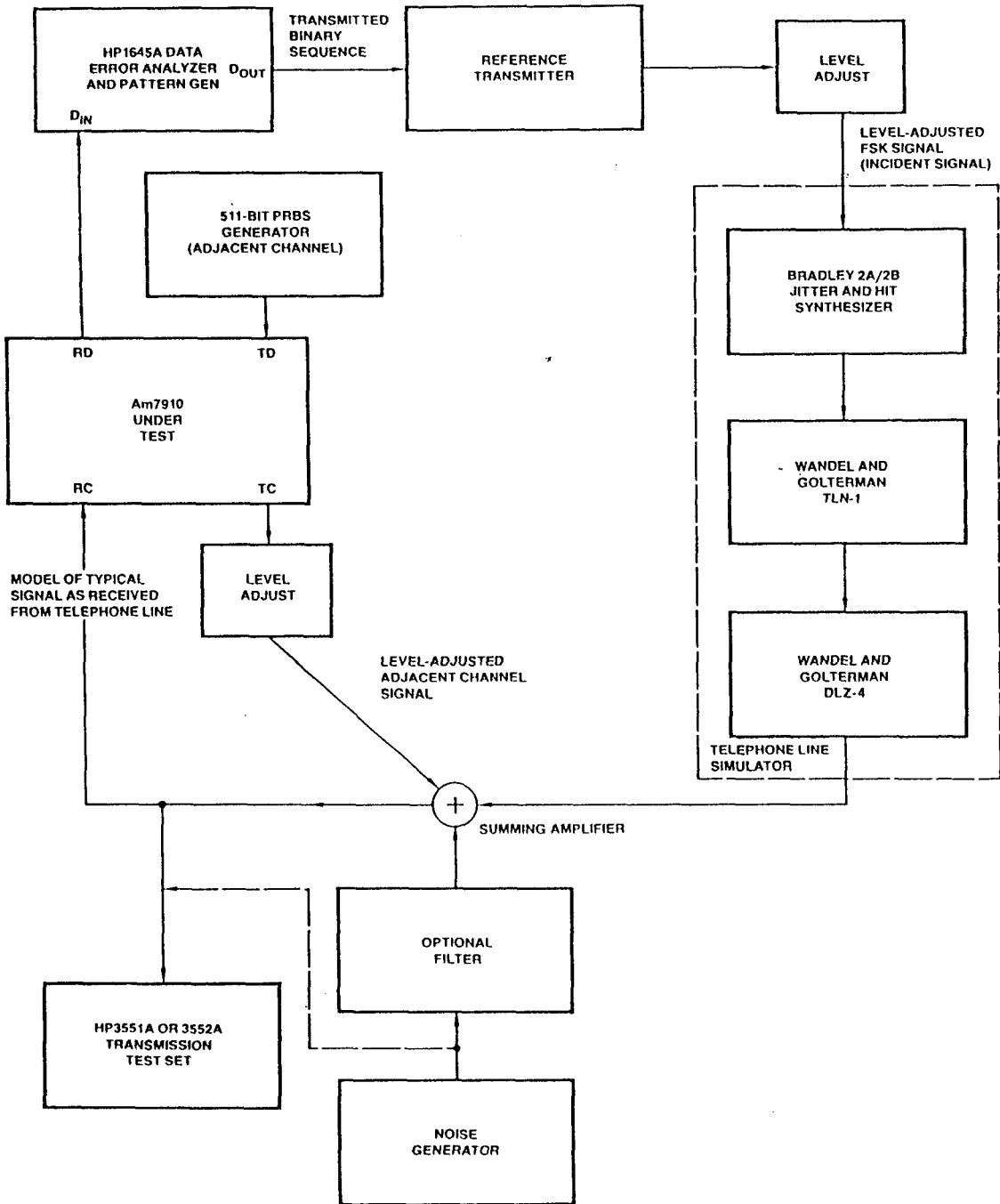
The HP3551A or HP3552A Transmission Test Set is used for measuring various levels which the modem under test is to receive. The levels of each of the three inputs to the summing amplifier should be measured independently of the other two inputs. For instance, the incident signal level should be measured by the transmission test set with no adjacent channel or noise present. The dashed line from the noise generator shows that the noise may or may not be measured at the output of the noise generator, depending on whether or not an optional filter is used, or on the characteristics of the filter.

Figure 16. Out-of-Band Transmitter Energy



17

Figure 17. BER and Distortion Measurement Test Setup



MAXIMUM RATINGS

Storage Temperature	-65 to +125°C
Ambient Temperature under Bias	0 to +70°C
V _{CC} with Respect to V _{DGND}	+6V/-4V
V _{BB} with Respect to V _{DGND}	-6V/+4V
All Signal Voltages with Respect to V _{DGND}	±5V

The products described by this specification include internal circuitry designed to protect input devices from damaging accumulations of charge. It is suggested, nevertheless, that conventional precautions be observed during storage, handling and use in order to avoid exposure to excessive voltages.

Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

OPERATING RANGE

Ambient Temperature	V _{CC}	V _{BB}	V _{AGND}	V _{DGND}
0°C ≤ T _A ≤ +70°C	+5.0V ±5%	-5.0V ±5%	0V ±50mV	0V

CHARACTERISTICS

Digital Inputs: TD, \overline{RTS} , MC₀-MC₄, \overline{DTR} , \overline{RING} , BTD, \overline{BRTS}
 Digital Outputs: RD, \overline{CTS} , \overline{CD} , BRD, BCTS, BCD

Parameters	Description	Test Conditions	Min	Typ	Max	Unit
V _{OH}	Output HIGH Voltage	I _{OH} = -50μA, C _{LD} = 50pF	2.4			Volts
V _{OL}	Output LOW Voltage	I _{OL} = +2mA, C _{LD} = 50pF			0.4	Volts
V _{IH}	Input HIGH Voltage		2.0		V _{CC}	Volts
V _{IL}	Input LOW Voltage		-0.5		0.8	Volts
V _{IHC}	External Clock Input HIGH (XTAL ₁)		3.8			Volts
V _{ILC}	External Clock Input LOW (XTAL ₁)		0.5		0.8	Volts
V _{IHR}	External Reset Input HIGH (\overline{RESET})		0.5		V _{CC}	Volts
V _{ILR}	External Reset Input LOW (\overline{RESET})		0.5		0.8	Volts
I _{IL}	Digital Input Leakage Current	0 ≤ V _{IN} ≤ V _{CC}	0		+10	μA
I _{CC}	V _{CC} Supply Current				125	mA
I _{BB}	V _{BB} Supply Current				25	mA
C _{OUT}	Output Capacitance	f _C = 1.0MHz		5	15	pF
C _{IN}	Input Capacitance	f _C = 1.0MHz		5	15	pF
Analog Input (RC):						
R _{IN}	Input Resistance	-1.6V < V _{RC} < +1.6V	50			Kohms
V _{RC}	Operating Input Signal		-1.6		+1.6	Volts
V _{RCOS}	Allowed DC Input Offset	REF V _{AGND}	-30		+30	mV
Analog Output (TC):						
V _{TC}	Output Voltage	R _L = 600Ω	-1.1		+1.1	Volts
V _{TCOS}	Output DC Offset			±200		mV