

ESCUELA POLITECNICA NACIONAL
FACULTAD DE INGENIERIA ELECTRICA
ESPECIALIZACION DE ELECTRONICA Y TELECOMUNICACIONES

DISEÑO Y CONSTRUCCION DE UN SISTEMA DE
ADQUISICION DE DATOS Y CONTROL DEL ROTOR PARA
OBTENER CONFIGURACIONES DE RADIACION DE
ANTENAS.

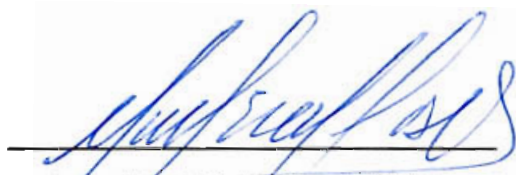
TESIS PREVIA A LA OBTENCION DEL TITULO DE
INGENIERO EN ELECTRONICA Y TELECOMUNICACIONES

SORAYA LUCIA SINCHE MAITA
OSWALDO VINICIO POLO GONZALEZ

QUITO, MARZO DE 1999

CERTIFICACION:

CERTIFICO QUE LA PRESENTE TESIS
HA SIDO DESARROLLADA POR LA
SRTA. SORAYA LUCIA SINCHE MAITA
Y EL SR. OSWALDO VINICIO POLO
GONZALEZ

A handwritten signature in blue ink, appearing to read 'Mario Cevallos V.', is written over a horizontal line.

ING. MARIO CEVALLOS V.

AGRADECIMIENTO:

A los Ingenieros: Mario Cevallos, Carlos Egas, Ramiro Morejón, Fernando Vásquez, Pablo Hidalgo, Tarquino Sánchez, Efrén Díaz, a los Señores: Carlos Herrera, Santiago Yépez y al Instituto Geofísico de la E.P.N. por la ayuda prestada en la realización del presente trabajo.

A nuestros amigos, compañeros y a todas aquellas personas que nos apoyaron durante el desarrollo de esta tesis.

DEDICATORIA

A mis padres, a quienes respeto y admiro por todo el sacrificio y la ayuda que me brindaron durante toda mi carrera, a mis hermanos por su comprensión y apoyo.

----- A Oswaldo por ser la persona con la cual he compartido los buenos y malos momentos a lo largo de todo este tiempo, por la paciencia y empeño puesto para culminar con este trabajo.

DEDICATORIA

A María Auxiliadora por ayudarme en los momentos difíciles, y a las personas que más admiro, mis Padres.

...

INDICE

CAPITULO I:

1. FUNDAMENTOS TEÓRICOS.

1.1. Patrones de radiación de una antena	1
1.1.1. Diagrama rectangular	1
1.1.2. Diagrama polar	2
1.1.3. Planos principales de radiación	2
1.1.3.1. Plano E	3
1.1.3.2. Plano H	3
1.2. Diagramas teóricos de radiación de antenas típicas	4
1.2.1. Dipolo corto	5
1.2.2. Dipolo largo	7
1.2.3. Arreglo de dos elementos	8

CAPITULO II:

2. REQUERIMIENTOS DE HARDWARE DEL SISTEMA A IMPLEMENTAR.

2.1. Sistema radioeléctrico de transmisión	10
2.1.1. Generación de la señal de prueba	10
2.1.2. Oscilador de RF y modulador	10
2.1.3. Amplificador de potencia	11
2.1.3.1. Diseño del amplificador	11
2.1.3.1.1. Red de acoplamiento de la etapa de entrada.....	14
2.1.3.1.2. Red de acoplamiento de la etapa de salida	17
2.1.3.2. Fuente de polarización variable	19
2.1.3.3. Diagrama esquemático del amplificador	20
2.1.4. Antena de transmisión	21

2.1.4.1. Balun	22
2.2. Sistema radio eléctrico para recepción	24
2.2.1. Antena de recepción	25
2.2.2. Acoplador de impedancias	25
2.2.3. Mezclador y oscilador	26
2.2.4. Amplificador de frecuencia intermedia	26
2.3. Control del motor	26
2.3.1. Control de arranque y parada	28
2.3.1.1. Funcionamiento del circuito	30
2.3.2. Control de sentido de giro	31
2.3.3. Sensor de posición.	31
2.3.4. Fuentes de polarización	32
2.3.5. Circuito de señalización y temporización	33
2.3.5.1. Circuito de señalización	33
2.3.5.2. Circuito de temporización	33

CAPITULO III:

3. ADQUISICIÓN, PROCESAMIENTO DE DATOS Y PRESENTACIÓN EN PANTALLA. REQUERIMIENTOS DEL SOFTWARE PARA EL SISTEMA

3.1. Adquisición y procesamiento de datos en base al Ezkit_Lite de Analog Device.	37
3.1.1. Descripción general del Ezkit_lite	38
3.1.2. Adquisición de datos	41
3.1.3. Procesamiento de la señal de prueba	41
3.1.4. Software implementado en el Ezkit_lite	42
3.1.4.1. Software para el posicionamiento del rotor de antena...	42	
3.1.4.1.1. Inicialización del ADSP2181 y configuración del AD1847	44

3.1.4.2. Software para la adquisición de datos.....	50
3.2. Software para la presentación de resultados en pantalla en base al lenguaje de programación Visual Basic 5.0.....	54
3.2.1. Generalidades del lenguaje de programación	54
3.2.2. Diseño del proyecto para la presentación del lóbulo de radiación	55
3.2.2.1. Diseño de la pantalla de presentación	56
3.2.2.2. Diseño de la pantalla principal	57

CAPITULO IV:

4. ANALISIS DE RESULTADOS, CONCLUSIONES Y RECOMENDACIONES.

4.1. Análisis de los resultados obtenidos en el sistema de adquisición de datos implementado	66
4.2. Conclusiones y Recomendaciones	74

ANEXOS

ANEXO A: Características eléctricas

ANEXO B: Manual de Usuario

ANEXO C: Listado de los programas implementados en el Ezkit_lite

ANEXO D: Código del programa CONFRAD

INTRODUCCION

INTRODUCCION

En la actualidad existe una gran variedad de sistemas de comunicación, los cuales tienen diversas aplicaciones y cuyo objetivo común es la transmisión de información de un lugar a otro. La transmisión de señales dependerá de varios aspectos como son distancia, medios de propagación, calidad de los equipos utilizados para la transmisión y recepción, entre otros.

Uno de estos sistemas es el denominado sistema radioeléctrico, el cual utiliza como medio de propagación el medio geográfico (atmósfera), mediante ondas radioeléctricas, definidas por la UIT (Unión Internacional de Telecomunicaciones) como ondas electromagnéticas que se propagan en el espacio sin guía artificial y como límite superior de frecuencia se fija convencionalmente, en 300GHz.

La información a ser transmitida en este tipo de sistemas tiene en sus puntos terminales de transmisión y recepción antenas, cuyas características deben ser adecuadamente calibradas para un buen desempeño del sistema.

Una de estas características es el lóbulo de radiación, el cual es una respuesta de la distribución espacial de la energía radiada por la antena.

La necesidad de tener un sistema que nos permita en forma experimental determinar el lóbulo de radiación de una antena, nos ha llevado a implementar un sistema automático, de fácil uso, el cual cumplirá con este propósito.

El objetivo del presente trabajo es obtener en forma experimental la representación gráfica del lóbulo de radiación de una antena en términos del valor del campo eléctrico y su posición angular.

Para este propósito se ha diseñado un bloque de transmisión del cual se obtiene una señal modulada en AM; un bloque de recepción en donde se procesa la información enviada por el transmisor en un bloque de adquisición de datos basados en el Ezkit_lite de Analog Devices, que nos permite procesar la información para luego ser enviada al computador a través del puerto serial.

Simultáneamente se tiene un motor que hará girar a la antena de donde se obtendrá la posición angular mediante un sensor óptico ubicado en el rotor del mismo, esta información ingresa al Ezkit_lite, desde donde se realiza el control del hardware que hace girar a la antena cuyas características de radiación se desean evaluar.

Todo este sistema es controlado desde un computador utilizando para este propósito Visual Basic 5.0. Con el cual se obtienen los diagramas de radiación experimentales de la antena que llamaremos antena de prueba.

En el Capítulo I se describen los modos de propagación de las ondas electromagnéticas así como los patrones de radiación de las antenas y las ecuaciones utilizadas para simular los mismos, lo que nos servirá como fundamento teórico para el desarrollo de nuestro trabajo.

En el Capítulo II se encuentran los requerimientos del sistema, diseño e implementación del hardware utilizado para la transmisión y recepción de la señal, así como para el control del motor.

El Capítulo III consta del desarrollo del software utilizado para el procesamiento y adquisición de datos, control del motor y graficación de los lóbulos de radiación.

En el Capítulo IV se realiza un análisis comparativo de los resultados obtenidos experimentalmente y los resultados obtenidos en base a la aplicación de desarrollos matemáticos, llegándose finalmente a determinar las conclusiones y recomendaciones del presente trabajo, el mismo que servirá de base a posteriores trabajos de investigación referentes a estos temas.

CAPITULO I

FUNDAMENTOS TEORICOS

1.1. PATRONES DE RADIACION DE UNA ANTENA.

Se entiende por patrón de radiación la distribución espacial de la energía radiada por una antena presentada en un diagrama polar o rectangular. Dicho diagrama es de gran importancia ya que nos permite conocer las características de las antenas: ganancia, directividad, relación delante/atrás, número de lóbulos secundarios, etc.

Los patrones de radiación pueden representarse en diversos sistemas de coordenadas, pero por lo común se puede obtener suficiente información si se los representa en coordenadas polares o rectangulares.

1.1.1. Diagrama Rectangular.

En este tipo de diagrama se fija el valor normalizado de la intensidad de campo en el eje vertical, y la posición angular en el eje horizontal.

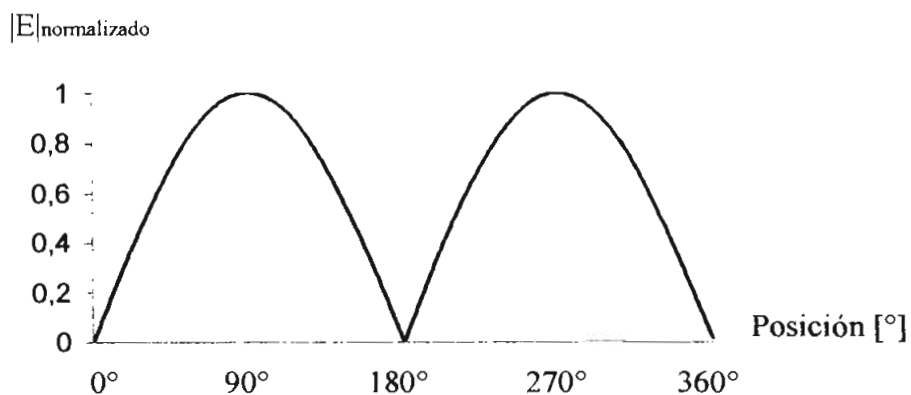


Figura 1.1. Diagrama Rectangular para un dipolo corto

1.1.2. Diagrama Polar.

En este diagrama, el valor de referencia de 0° se ubica en un eje y el valor normalizado de la intensidad de campo se ubica en forma radial con relación al centro del sistema de coordenadas y la posición angular se obtiene con respecto al eje de referencia de 0° .

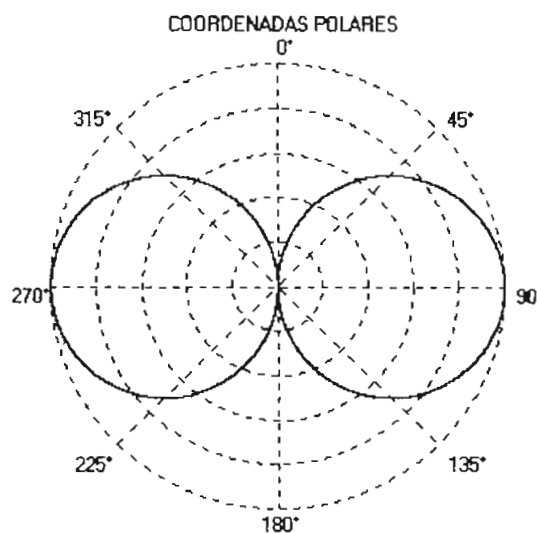


Figura 1.2. Diagrama polar para un dipolo corto

1.1.3. Planos de Radiación.

A grandes distancias desde la fuente de radiación, las ondas irradiadas pueden ser consideradas como planas, en este caso las ondas electromagnéticas son transversales, con los campos E y H ortogonales.

Los campos E y H están en fase. El campo eléctrico oscila en el plano XY y el campo magnético en el plano XZ.

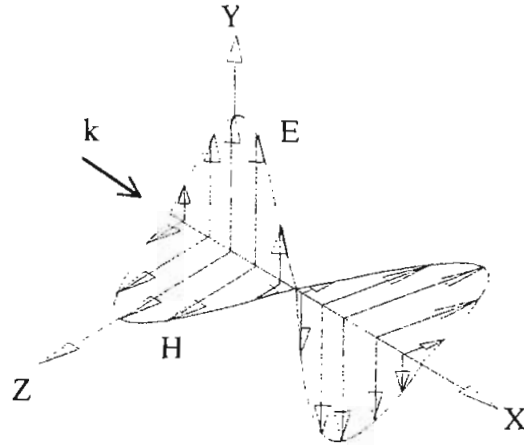


Figura 1.3. Campo eléctrico y magnético en una onda electromagnética plana.

Una antena tiene dos planos principales de radiación: plano E y plano H.

1.1.3.1. Plano E.

El plano E es aquel que es paralelo al vector de campo eléctrico y pasa a través de la antena en la dirección de máxima radiación.

1.1.3.2. Plano H.

El plano H es perpendicular al plano E y también pasa a través de la antena en la dirección de máxima radiación; se lo puede definir también como el plano que contiene al vector de campo magnético y la dirección de máxima radiación.

Otro de los parámetros de mucha importancia es el tipo de polarización de la antena, que está determinada por la orientación de las líneas de fuerza del campo eléctrico radiado o recibido por la antena. La polarización de una antena en una dirección específica está definida como el lugar geométrico de la variación temporal del campo E que produce una antena. Esta puede ser: lineal, circular o elíptica. La primera de ellas a su vez puede ser diagonal, vertical y horizontal, lo que dependerá de la orientación del campo eléctrico con relación al elemento radiante.

La polarización es horizontal cuando el campo E es paralelo a la superficie de la tierra, y es vertical cuando es perpendicular a la superficie de la tierra.

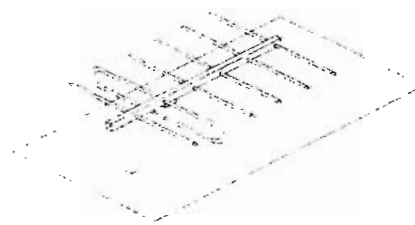


Figura 1.4. Antena Yagui polarizada horizontalmente

1.2. Diagramas Teóricos de Radiación de Antenas Típicas.

Una antena es un elemento que sirve para emitir o recibir ondas electromagnéticas.

Una antena puede ser considerada como una fuente de radiación puntual para campo lejano. Entendiéndose por campo lejano o región Fraunhofer, la región que se halla separada de la antena una distancia dada por:

$$R > 2 L^2 / \lambda$$

donde: L = la mayor longitud de la antena.

λ = longitud de onda.

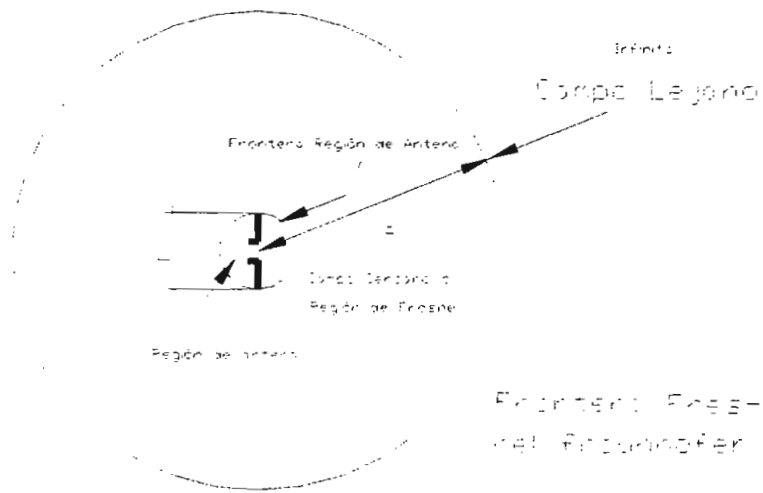


Figura 1.5. Regiones de campo para una antena.

Esta consideración se la realiza para que la distribución de la energía radiada desde la antena sea independiente de la distancia del punto de observación.

1.2.1. Dipolo Corto.

Está formado por dos conductores pequeños, con excitación central simétrica, cuya longitud para propósitos prácticos es menor que la cuarta parte de la longitud de onda de la radiación. La corriente que circula por esta antena tiene una distribución lineal.

Basados en análisis matemáticos se llega a obtener la ecuación de campo lejano en valor absoluto, que nos permite obtener el patrón de radiación de este tipo de dipolo.

$$E = \frac{30\pi I H \text{ sen } \theta}{\lambda R}$$

Donde:

E = intensidad de campo eléctrico (V/m)

I = intensidad de corriente (A)

H = semilongitud del dipolo ó L = 2H (m)

θ = ángulo entre el eje de la antena y la dirección de radiación (rad)

λ = longitud de onda (m)

R = distancia desde el dipolo al punto de observación (m)

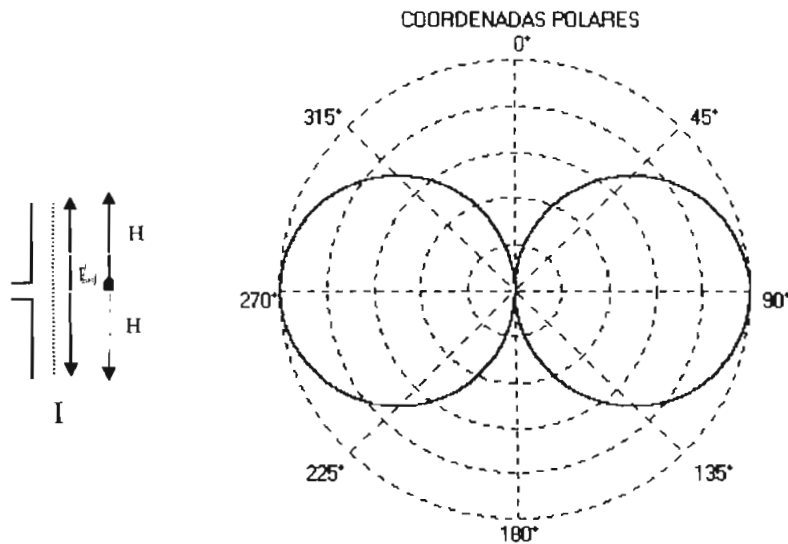


Figura 1.6. Diagrama polar para un dipolo de $L = 0.125\lambda$

1.2.2. Dipolo Largo.

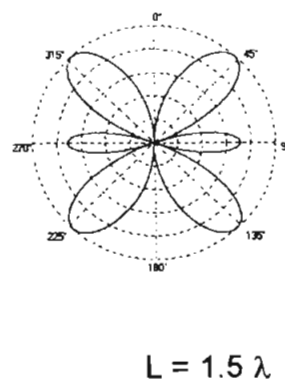
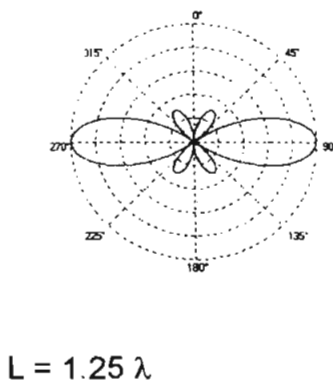
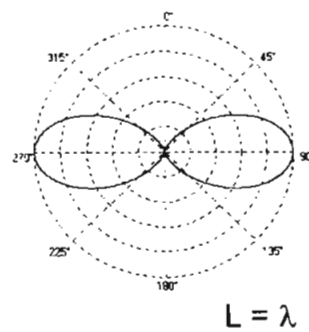
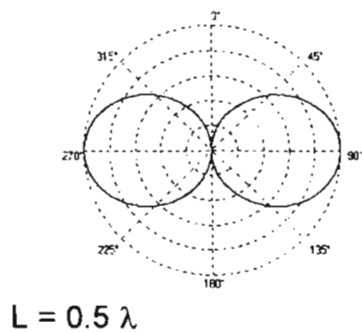
Se puede considerar un dipolo largo a aquel cuya longitud es mayor que $\lambda/2$. La corriente que circula por este tipo de antena tiene una distribución sinusoidal.

El campo lejano en valor absoluto en un dipolo largo está dado por la siguiente expresión:

$$E = \frac{60 I}{R} \left(\frac{\cos(\beta H \cos \theta) - \cos(\beta H)}{\sin \theta} \right)$$

Donde: $\beta =$ constante de desviación de fase $= 2\pi/\lambda$.

A continuación se presentan varios gráficos de los lóbulos de radiación de dipolos largos.



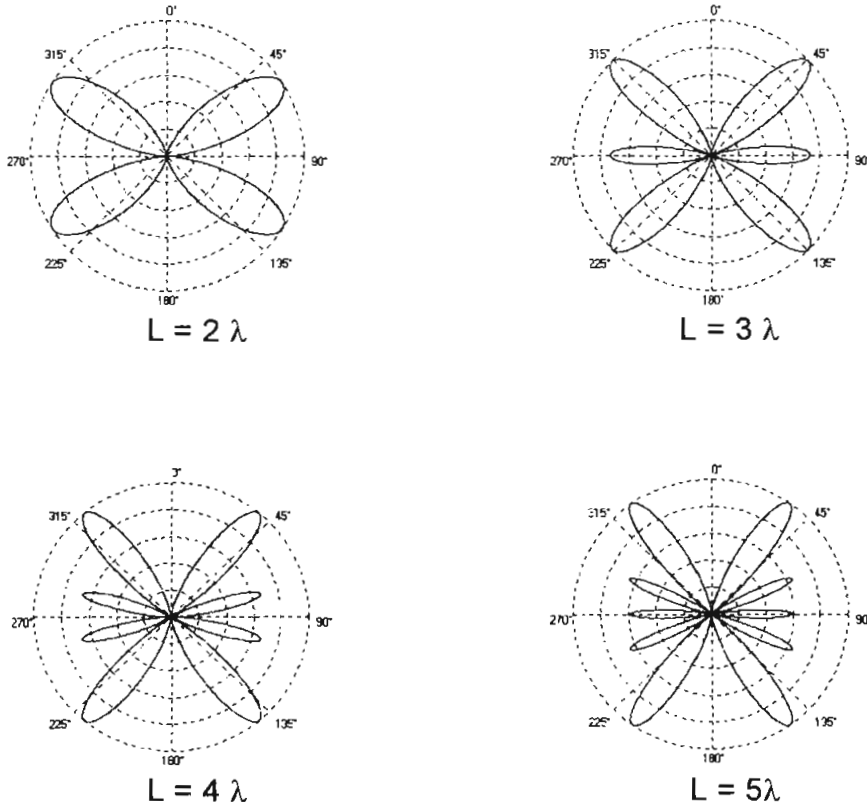


Figura 1.7. Diagramas polares de algunos dipolos.

1.2.3. Arreglo de dos elementos.

En un arreglo de dos elementos iguales en longitud y con iguales corrientes de excitación, el diagrama de radiación depende de dos variables: la distancia entre los elementos y el desfase entre las corrientes que alimentan a los elementos.

El campo lejano está dado por:

$$E = 2E_1 \cos\left(\frac{\beta d \cos \theta \pm \alpha}{2}\right)$$

Donde : E_1 = campo radiado por cada elemento

d = distancia entre las antenas

α = desfase entre las señales

Todas las ecuaciones descritas serán utilizadas para realizar la graficación simulada de los lóbulos de radiación.

$$d = 0.5\lambda$$

$$\alpha = 90^\circ$$

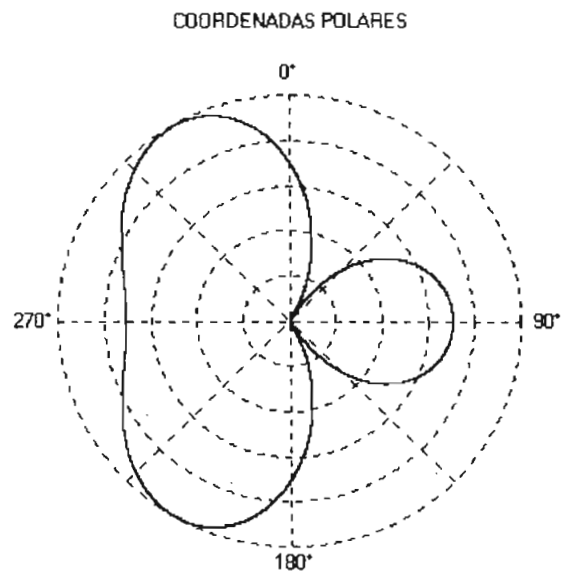


Figura 1.8. Diagrama polar de un arreglo de dos elementos

CAPITULO II

REQUERIMIENTOS DE HARDWARE DEL SISTEMA A IMPLEMENTAR

2.1. Sistema Radioeléctrico de Transmisión

El sistema radioeléctrico de transmisión consta de un conjunto de equipos que permiten adecuar la información a ser transmitida. El equipo de transmisión a ser implementado se presenta en el siguiente diagrama de bloques:

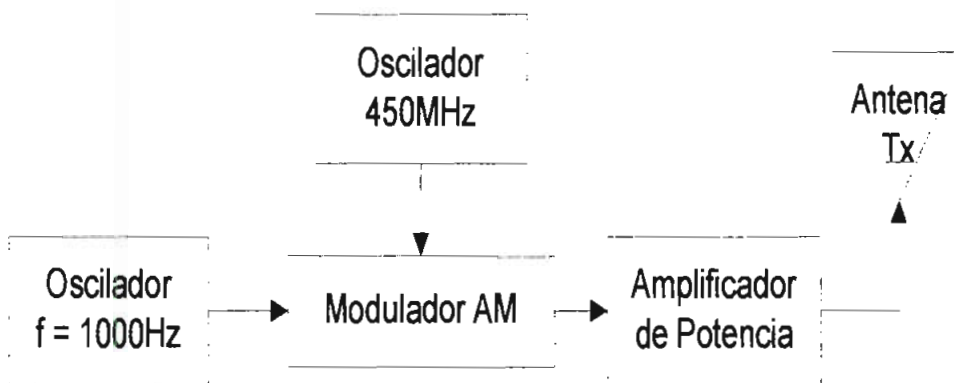


Figura 2.1. Sistema de Transmisión.

2.1.1. Generación de la señal de prueba.

La información a ser enviada es una onda senoidal, que se la obtiene del oscilador tipo 1214A de General Radio existente en el laboratorio, el cual brinda salidas senoidales de 1KHz y de 400Hz. Se ha escogido la señal de 1 KHz por las facilidades que presenta el muestreo de la señal en la etapa de adquisición de datos.

2.1.2. Oscilador de RF y Modulador.

Para la realización de dichas funciones se ha utilizado el oscilador modelo 1362 existente en el laboratorio, el cual trabaja en un rango de frecuencias de 220

a 920 MHz, provee a la salida una potencia máxima de 200mW y una impedancia de salida de 50Ω . Este equipo nos permite modular en amplitud la información.

2.1.3. Amplificador de Potencia.

Debido a que la señal a ser transmitida se puede ver afectada por señales ajenas a nuestro sistema, se requiere que la misma tenga un nivel de potencia suficientemente fuerte lo cual no se puede conseguir con los equipos existentes en el laboratorio por lo que se procedió a amplificar la señal.

La señal es amplificada a 1W, para lo cual utilizamos el transistor 2N5944 (ECG392) cuyo rango de operación está entre 400 a 960 MHz.

2.1.3.1. Diseño del Amplificador.

Debido a que la frecuencia de trabajo es de 450MHz, el diseño se lo realizará en base a las técnicas del Microstrip Line, ya que estas nos facilitan la implementación de los valores requeridos para ciertos elementos que se consideran en el diseño.

Para poder iniciar el diseño con la técnica del Microstrip debemos conocer las características de la baquelita a ser utilizada. Para ello se procedió a medir de forma indirecta el valor de la permitividad, obteniendo como resultado el valor de 2.

Además se debe conocer los siguientes parámetros:¹

W = ancho del microstrip

h = espesor del dieléctrico

t = espesor del cobre

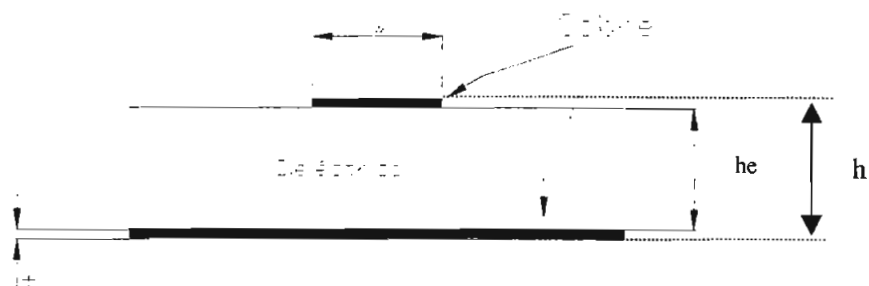


Figura 2.2. Sección transversal de Microstrip

$$1\text{mils} = 39.37\text{mm}$$

$$W = 5\text{ mm} = 196.85\text{ mils}$$

$$h = 1.5\text{mm} = 59.055\text{ mils}$$

$$t = 1.4\text{ mils}$$

$$h_e = h - 2t$$

$$h_e = 59.055 - 2 \times 1.4$$

$$h_e = 56.255\text{mils}$$

$$W_{eff} = W + \frac{t}{\pi} \left(\ln \left(\frac{2h_e}{t} + 1 \right) \right)$$

$$W_{eff} = 196.85 + \frac{1.4}{\pi} \left(\ln \left(\frac{2 \times 56.255}{1.4} + 1 \right) \right)$$

$$W_{eff} = 198.81\text{mils}$$

¹ TAM Alam, Principles of Microstrip Design, June 1988

La longitud de onda en este tipo de líneas está afectada por el ancho de la cinta, dando lugar a la relación:

$$\lambda = \frac{c}{\sqrt{\epsilon_{eff}} f}$$

Donde: c = velocidad de la luz

f = frecuencia de operación

ϵ_{eff} = permitividad relativa efectiva

$$\epsilon_{eff} = \frac{\epsilon_r + 1}{2} + \frac{\epsilon_r - 1}{2} \left[\left(1 + \frac{12 h}{W_{eff}} \right)^{-\frac{1}{2}} \right]$$

$$\epsilon_{eff} = \frac{2 + 1}{2} + \frac{2 - 1}{2} \left[\left(1 + \frac{12 \times 59.055}{198.81} \right)^{\frac{1}{2}} \right]$$

$$\epsilon_{eff} = 1.734$$

$$\lambda = \frac{300 \times 10^6}{\sqrt{1.734} \times 450 \times 10^6} \text{ (m)}$$

$$\lambda = 0.506 \text{ m}$$

$$Z_0 = \frac{377}{\sqrt{\epsilon_{eff}} \left[\frac{W_{eff}}{h} + 1.393 + 0.667 \ln \left(\frac{W_{eff}}{h} + 1.444 \right) \right]}$$

$$Z_0 = \frac{377}{\sqrt{1.734} \left[\frac{198.81}{59.055} + 1.393 + 0.667 \ln \left(\frac{198.81}{59.055} + 1.444 \right) \right]}$$

$$Z_0 = 49.4 \Omega$$

Debido a que la impedancia obtenida se acerca al valor ideal de 50Ω el error cometido al trabajar con una impedancia característica de 50Ω en el diseño del amplificador es mínimo.

El amplificador se diseña en clase C, debido a la eficiencia de esta configuración. Para el diseño del mismo se necesita conocer la impedancia que presentan el generador y la carga que se conectan a la entrada y a la salida del amplificador respectivamente. Como por lo general estas impedancias no son iguales a las impedancias de entrada y salida que presenta el transistor, se requiere la implementación de redes transformadoras de impedancia para obtener máxima transferencia de potencia.

Los valores de impedancias obtenidos de las hojas de especificaciones del transistor 2N5944 para 450 MHz son:

$$Z_{in} = 1.2 + j0.66 \Omega$$

$$Z_{out} = 5.3 - j11 \Omega$$

2.1.3.1.1. Red de Acoplamiento de la etapa de entrada

Para realizar el acoplamiento se utilizó una red PI, la que nos da flexibilidad a través de los capacitores variables utilizados.

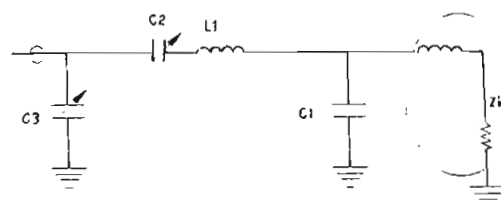


Figura 2.3. Red de acoplamiento de entrada

Ubicamos la impedancia de entrada en la carta de Smith (punto A), obtenemos la admitancia de la misma (punto B), a través de un círculo de conductancia constante se llega hasta el punto C, para lo cual necesitamos añadir una susceptancia B1

$$jB_1 = (Y_C - Y_B) Y_0$$

$$jB_1 = [(31 + j 8) - (31 - j 17.58)] Y_0$$

$$jB_1 = j 25.58 Y_0$$

$$jB_1 = j 0.5116 \Omega$$

$$B_1 = 2\pi f C_1$$

$$C_1 = \frac{B_1}{2\pi f}$$

$$C_1 = \frac{0.5116}{2\pi \times 450 \times 10^6}$$

$$C_1 = 180.94 \text{ pF}$$

Luego el valor de la admitancia del punto C lo transformamos a su equivalente de impedancia (punto D) y nos movemos en un círculo de resistencia constante hasta el punto E, obteniendo como resultado una reactancia X1, la misma que equivale a movernos una determinada longitud eléctrica en la carta de Smith.

$$\frac{l_1}{\lambda_{eff}} = \frac{l_E}{\lambda_{eff}} + \frac{l_D}{\lambda_{eff}}$$

$$\frac{l_1}{\lambda_{eff}} = 0.001 + 0.03$$

$$\frac{l_1}{\lambda_{eff}} = 0.031$$

$$l_1 = 0.031 \times 50.6 \text{ cm}$$

$$l_1 = 1.5686 \text{ cm}$$

Obtenemos el correspondiente valor de admitancia del punto E (punto F), y manteniendo un círculo de conductancia constante cortamos el círculo unitario desplazado (punto G), obteniendo el valor de susceptancia B_2 .

$$jB_2 = (Y_G - Y_F) Y_0$$

$$jB_2 = [(0.8 - j 0.4) - (0.8 - j 5)] Y_0$$

$$jB_2 = j 4.6 Y_0$$

$$jB_2 = j 0.092 \Omega$$

$$C_2 = \frac{B_2}{2 \pi f}$$

$$C_2 = \frac{0.092}{2 \pi \times 450 \times 10^6}$$

$$C_2 = 32.53 \text{ pF}$$

El valor de la admitancia del punto G lo transformamos a su equivalente de impedancia (punto H) el cual corta el círculo unitario. Posteriormente nos movemos hasta el centro de la Carta de Smith terminando el acoplamiento, para lo cual se tiene que agregar un valor de reactancia X_3 .

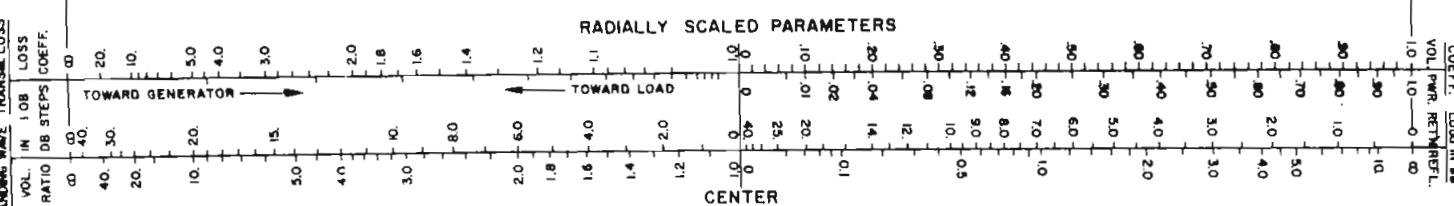
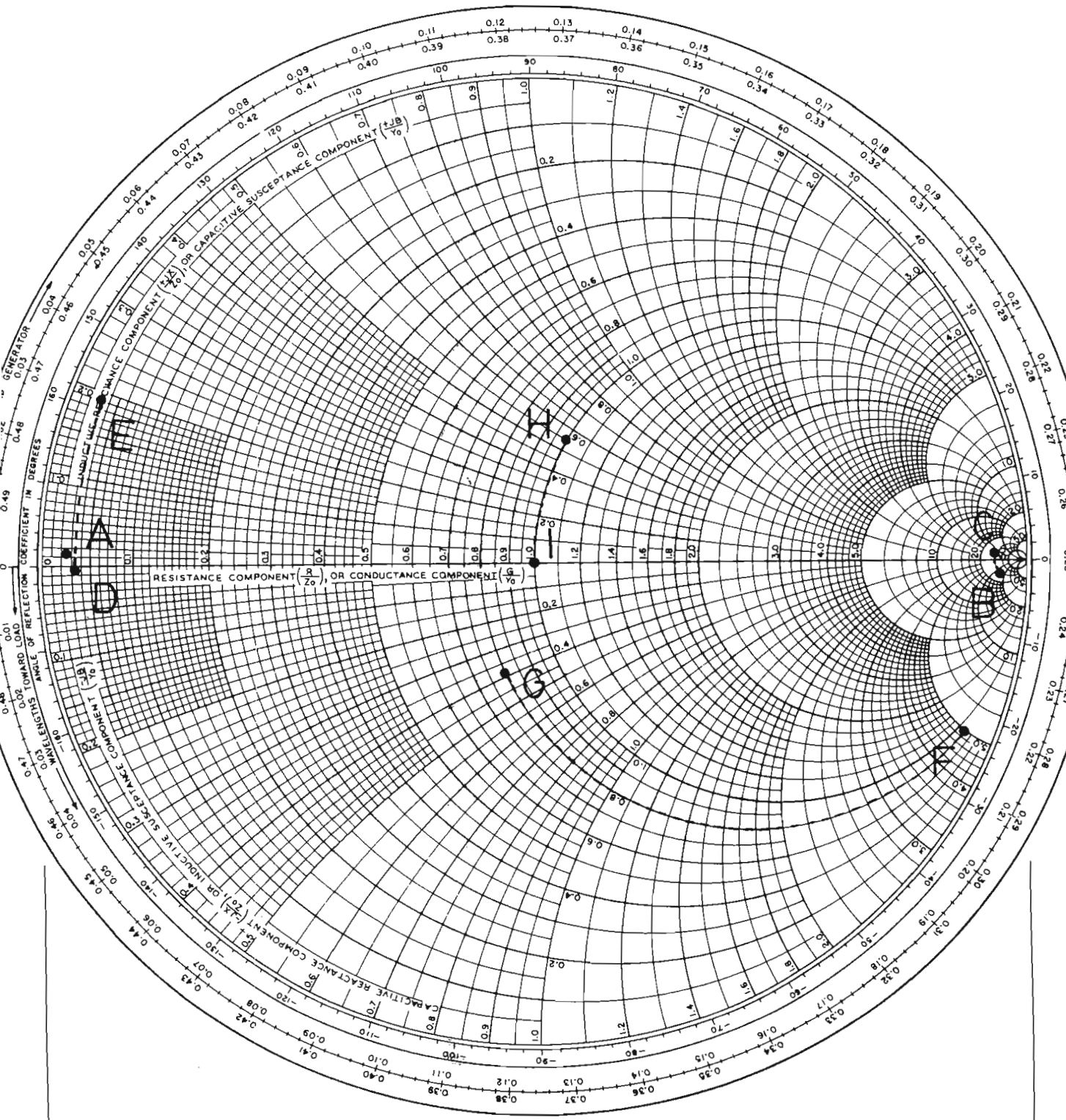
$$jX_3 = (Z_I - Z_H) Z_0$$

$$jX_3 = [1 - (1 + j 0.55)] Z_0$$

$$jX_3 = -j 0.55 Z_0$$

$$jX_3 = -j 27.5 \Omega$$

IMPEDANCE OR ADMITTANCE COORDINATES



$$C_3 = \frac{1}{2 \pi f X_3}$$

$$C_3 = \frac{1}{2 \pi \times 450 \times 10^6 \times 27.5}$$

$$C_3 = 12.86 \text{ pF}$$

2.1.3.1.2. Red de Acoplamiento de la etapa de Salida

Para realizar el acoplamiento se utilizó una red L, la que nos permite acoplar la red del lado de baja impedancia al de alta impedancia.

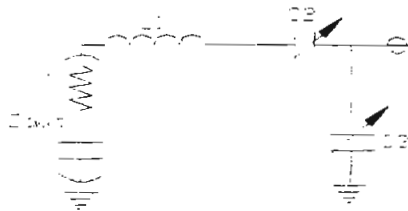


Figura 2.5. Red de acoplamiento de salida

Ubicamos la impedancia de salida en la carta de Smith (punto A), a través de un círculo de resistencia constante llegando hasta el punto B, para lo cual se necesita añadir una reactancia X_1' , que equivale a desplazarnos una longitud eléctrica igual a:

$$\frac{l_1'}{\lambda_{eff}} = \frac{l_A}{\lambda_{eff}} + \frac{l_B}{\lambda_{eff}}$$

$$\frac{l_1'}{\lambda_{eff}} = 0.0345 + 0.0775$$

$$\frac{l_1'}{\lambda_{eff}} = 0.112$$

$$l_1' = 0.112 \times 50.6 \text{ cm}$$

$$l_1' = 5.6 \text{ cm}$$

Con un desplazamiento hasta el punto C en un círculo de resistencia constante se obtiene como resultado la reactancia X_2' .

$$jX_2' = (Z_C - Z_B) Z_0$$

$$jX_2' = [(0.106 + j 0.3) - (0.106 + j 0.53)] Z_0$$

$$jX_2' = -j 0.23 Z_0$$

$$jX_2' = -j 11.5 \Omega$$

$$C_2' = \frac{1}{2\pi f X_2'}$$

$$C_2' = \frac{1}{2\pi \times 450 \times 10^6 \times 11.5}$$

$$C_2' = 30.75 \text{ pF}$$

Se encuentra el correspondiente valor de admitancia del punto C (punto D), el cual corta al círculo unitario. Para posteriormente realizar un desplazamiento hasta el centro de la Carta de Smith mediante un valor de admitancia B_3 .

$$jB_3 = (Y_E - Y_D) Y_0$$

$$jB_3 = [1 - (1 - j 2.98)] Y_0$$

$$jB_3 = j 2.98 Y_0$$

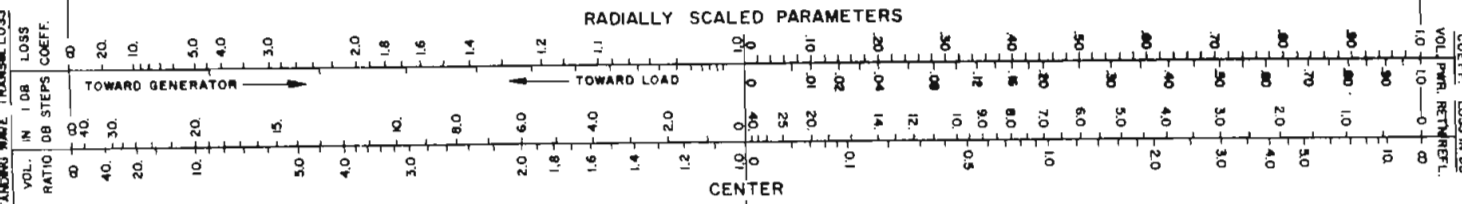
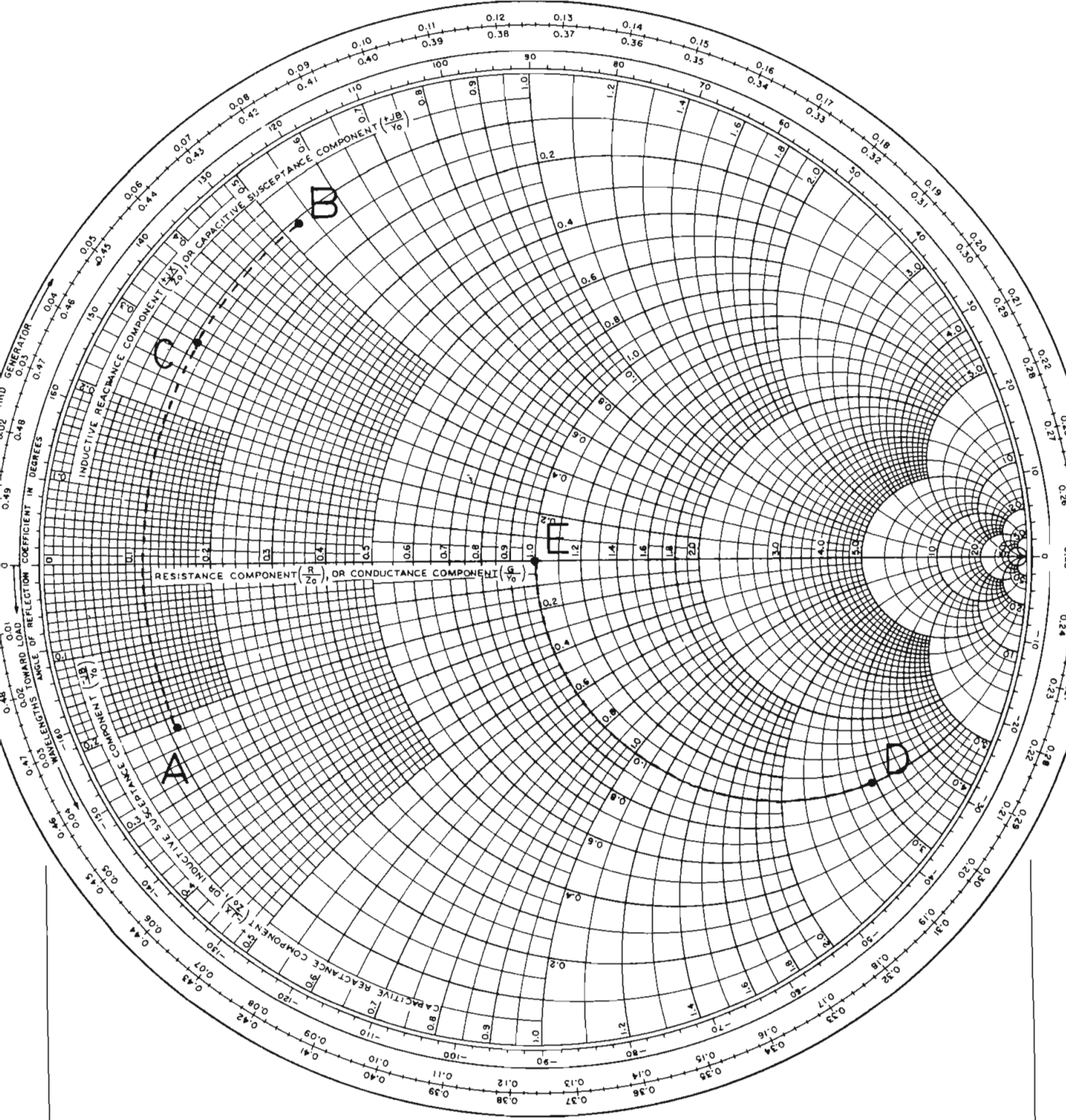
$$jB_3 = j 0.0596 \Omega$$

NAME
SMITH CHART FORM 530I-7560-N

TITLE RED DE ACOPLAMIENTO DE SALIDA
GENERAL RADIO COMPANY, WEST CONCORD, MASSACHUSETTS

DWG. NO.
DATE

IMPEDANCE OR ADMITTANCE COORDINATES



$$C_3 = \frac{B_3}{2\pi f}$$

$$C_3 = \frac{0.0596}{2\pi \times 450 \times 10^6}$$

$$C_3 = 21 pF$$

2.1.3.2. Fuente de polarización variable.

La fuente de alimentación necesaria para que opere el transistor 2N5944 es de 12 voltios DC (11.5 voltios valor utilizado) con una corriente máxima de 240 mA, por lo que se implementó una fuente variable en base al LM317TW (ECG956) que soporta hasta 1.5 Amperios y nos permite variar el voltaje desde 1.2 a 37 voltios²

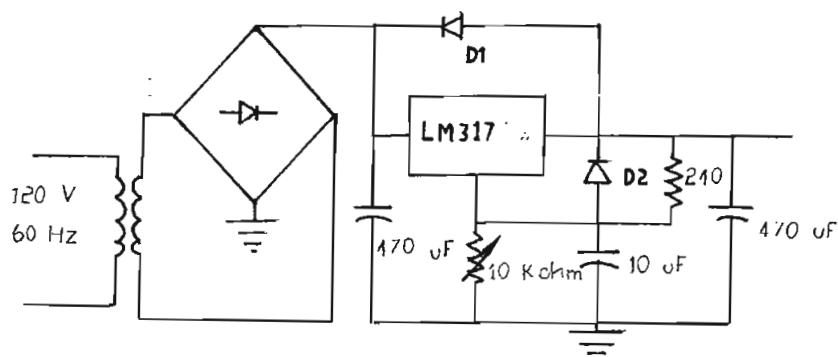


Figura 2.7. Fuente variable de 7 a 15 Voltios

A continuación se presenta el circuito esquemático del amplificador implementado, así como también el del circuito impreso.

² Esquema basado en el libro de Boylestad-Nashelsky, ELECTRONICA TEORIA DE CIRCUITOS, 4ta ed, pág 740.

2.1.3.3. Diagrama esquemático del amplificador implementado utilizando la técnica del Strip Line.

Amplificador de Potencia en base al transistor 2N5944

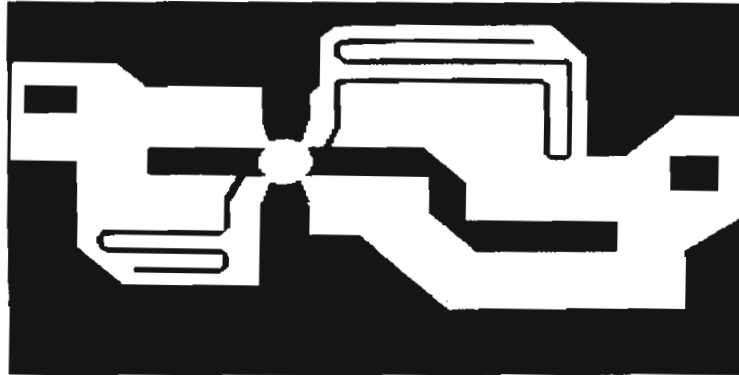
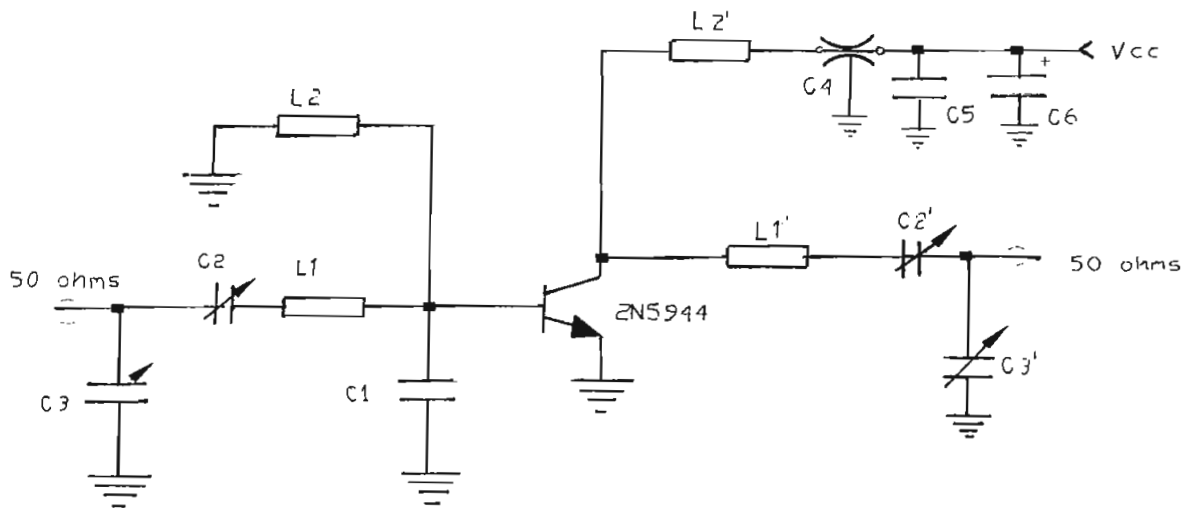


Figura 2.8.1. Circuito impreso implementado.



- | | |
|---|-------------------------------|
| C1 = 180 pF | L1 = 15.68 mm x 5 mm |
| C2 = 9 - 50 pF | L1' = 56 mm x 5 mm |
| C3 = 3 - 30 pF | L2 = 80mm x 2 mm |
| C2' = 9 - 50 pF | L2' = 110 mm x 2 mm |
| C3' = 3-30 pF | |
| C4 = capacitor de pasc Baquelita de 2 oz de cobre | |
| C5 = 0.1 uF | y de permitividad relativa= 2 |
| C6 = 1 uF | |

Figura 2.8.2. Circuito esquemático del Amplificador de potencia a 450 MHz.

2.1.4. Antena de Transmisión

El elemento radiador que se ha escogido para la etapa de transmisión es una antena Yagui, la cual está diseñada con las siguientes características:

- Número de elementos: 6
- Ganancia: 10dB
- Frecuencia de trabajo: 450MHz
- Radiador : dipolo doblado
- Acoplamiento a 50Ω

La longitud de onda con la cual se va trabajar es:

$$\lambda = \frac{c}{f}$$
$$\lambda = \frac{300}{450} \text{ cm}$$
$$\lambda = 66.67 \text{ cm}$$

Para la realización del diseño se tomo como base una tabla⁽¹⁾ con valores experimentales de antenas Yagui de 6 elementos y se obtuvo los siguientes resultados:

$$\text{Longitud del elemento reflector} = \lambda/2 \Rightarrow 33.33 \text{ cm}$$

$$\text{Longitud del elemento radiador} = 0.95 \lambda/2 \Rightarrow 31.67 \text{ cm}$$

$$\text{Longitud del primer elemento director} = 0.91 \lambda/2 \Rightarrow 30.33 \text{ cm}$$

$$\text{Longitud del segundo elemento director} = 0.875 \lambda/2 \Rightarrow 29.2 \text{ cm}$$

¹ EDITEC, Práctica de la construcción e instalación de antenas de FM y Televisión

Longitud del tercer elemento director = $0.84 \lambda/2 \Rightarrow 28$ cm

Longitud del cuarto elemento director = $0.81 \lambda/2 \Rightarrow 27$ cm

La distancia de separación entre elementos es igual a $\lambda/4$ que para nuestro caso equivale a 16.66 cm, los tubos que se utilizaron para elaborar los elementos tienen un diámetro de 7.9mm y el boom se lo realizó con un tubo de sección cuadrangular de 2.1cm de lado.

Considerando que la antena presenta en sus terminales de alimentación una salida balanceada de 300Ω y el equipo al cual se va a conectar tiene una salida desbalanceada de 50Ω , se necesita un sistema adaptador de impedancias.

2.1.4.1. Balun

Los transformadores o adaptadores de impedancias (balun) son necesarios para que la señal que sale del amplificador de potencia llegue a la antena con las menores pérdidas posibles.

Existen muchas maneras de construir baluns, por lo que el método escogido dependerá de: la frecuencia de operación, el grado de desequilibrio de impedancias y la intensidad de la señal.

De acuerdo a su respuesta de frecuencia estos pueden ser: Baluns lineales o de banda estrecha y baluns de banda ancha.

Para nuestro propósito en el caso de la antena de transmisión se ha escogido el balun de banda estrecha, el cual se halla formado de secciones de línea de transmisión de una determinada longitud de onda.

Debido a que la impedancia que presenta el equipo de transmisión es 50Ω desbalanceados y la antena presenta en sus terminales de alimentación una impedancia de 300Ω balanceados, el balun a ser diseñado tendrá una relación de transformación de 1: 6 (desbalanceado: balanceado). Para obtener esta relación de transformación se lo diseño bajo el siguiente esquema:

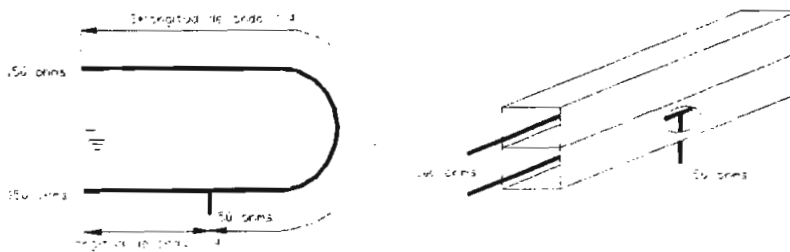


Figura 2.9. Esquema de un balun con relación de transformación 1:6

El segmento de línea de longitud $\lambda/2$ realiza una transformación de impedancias de relación 1:4, por lo que la impedancia a la salida de este segmento es de 200Ω balanceados. El segmento de longitud $\lambda/4$ realiza un acoplamiento de relación 1:1.5 (balanceado – balanceado), obteniendo de esta manera una impedancia de 300Ω .

El segmento de longitud $\lambda/4$ tiene una impedancia característica Z_0 igual a:

$$Z_0 = \sqrt{150 \times 100} \qquad Z_0 = 122.47\Omega$$

El balun diseñado como se muestra en la figura 2.9 utiliza dos conductores paralelos que se mantienen fijos a determinada distancia por medio de espaciadores aislantes los cuales se encuentran ubicados al interior de un tubo cuadrado de aluminio de 2.1cm por lado. El diámetro del conductor a ser utilizado se lo calcula de la siguiente manera:

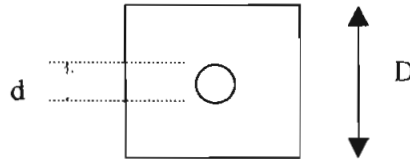


Figura 2.10. Sección transversal del balun implementado.⁴

$$Z_0 \approx (138 \log_{10} \rho + 6.48 - 2.34 A - 0.48 B - 0.12 C) e^{-1.2}$$

$$\rho = D / d$$

$$A = \frac{1 + 0.405 \rho^{-4}}{1 - 0.405 \rho^{-4}}; B = \frac{1 + 0.163 \rho^{-8}}{1 - 0.163 \rho^{-8}}; C = \frac{1 + 0.067 \rho^{-12}}{1 - 0.067 \rho^{-12}}$$

Para obtener una impedancia de 122.47Ω , es necesario que $\rho = 6.13$ aproximadamente (datos obtenidos experimentalmente). Para cumplir con esta relación y conociendo que $D = 2$ cm se obtiene que el conductor ha ser utilizado es el AWG #8.

2.2. Sistema Radioeléctrico para Recepción

El diagrama de bloques utilizado para el sistema de recepción es el siguiente:

⁴ ITT Reference Data For Radio Engineers, fifth edition, page 22-22

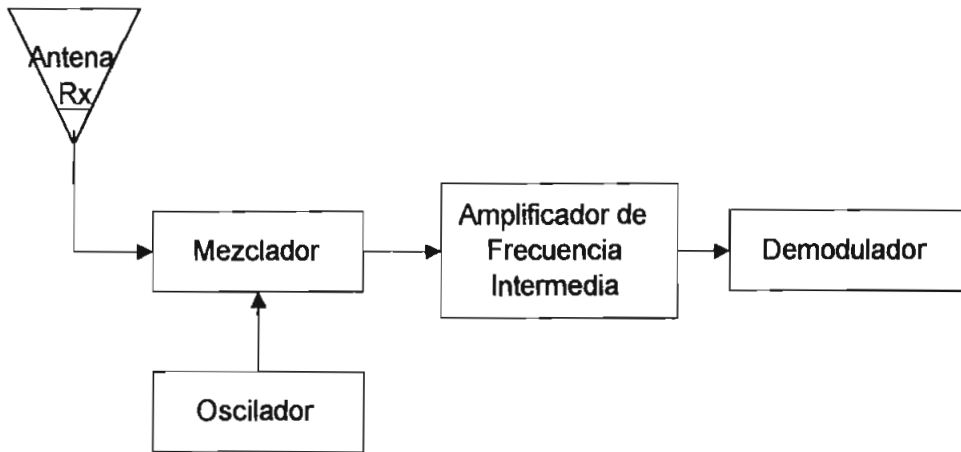


Figura 2.11. Sistema de Recepción

2.2.1. Antena de Recepción

Como antena receptora patrón se utilizó un dipolo doblado $\lambda/2$ para la frecuencia de 450MHz. La impedancia de salida de esta antena es de 300Ω . Cabe señalar que se puede colocar cualquier tipo de antena que responda a la frecuencia de trabajo y se encuentre acoplada a los equipos de laboratorio utilizados. Además se debe tomar en cuenta el nivel máximo de la señal captada para que el equipo no se sature.

Al igual que la antena de transmisión, se necesita de un sistema de adaptación de impedancias, con la misma relación de transformación.

2.2.2. Acoplador de Impedancias

La señal que llega a la antena es por lo general débil por tanto se debe transferir esta señal con la menor pérdida de energía. Para ello se recurre a un

transformador de impedancias el cual se diseñó con el mismo principio del equipo de transmisión, con la diferencia que se utilizó una línea de transmisión cuya impedancia característica es de 50Ω (RG58).

2.2.3. Mezclador y Oscilador

Para mezclar las señales se dispone del mezclador 874-MRAL el cual trabaja dentro de un rango de frecuencia de 10MHz a 900MHz, al cual se ingresa la señal proveniente de la antena y la señal del oscilador 1362, obteniéndose así la señal que va a entrar al amplificador de frecuencia intermedia.

2.2.4. Amplificador de Frecuencia Intermedia

Para este propósito utilizamos el equipo amplificador 1236I-F de General Radio existente en el laboratorio, al cual se ingresa la señal proveniente del mezclador obteniendo a su salida la señal demodulada, la cual posteriormente ingresará al sistema de adquisición de datos.

2.3. Control del Motor

Para la graficación del lóbulo de radiación se requiere tener como información la posición angular de la antena, para esto se dispone en el laboratorio de un rotor de antena, el cual tiene para el efecto un motor de inducción monofásico de fase partida.

El rotor tiene un sistema reductor de velocidad y un tope en base a un electroimán el que para nuestro propósito deberá estar activado continuamente.

Este motor esta formado por dos devanados, los que tienen igual número de espiras e igual sección del hilo del cual están hechos, por lo que presentan la misma impedancia en sus terminales. Para el arranque utiliza un condensador permanente.

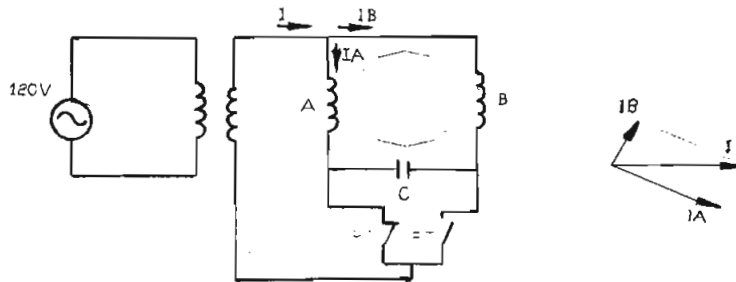


Figura 2.12. Esquema de motor

Las características del motor son:

Voltaje de alimentación = 24V

Corriente máxima = 2.25 A

Condensador = 130MFD 110/115 VAC

Las características del tope son:

Voltaje de alimentación = 24 V

Corriente máxima = 3.25 A

Para realizar el control del motor se utilizan las banderas programables PF0, PF3 y PF4 del microcontrolador ADSP 2181, las mismas que sirven para el arranque, cambio del sentido de giro y activación del tope del motor respectivamente.

2.3.1. Control de Arranque y Parada

El arranque y parada del motor se lo realiza a través del Triac TIC226M (ECG5609). Para disparar al Triac, se utiliza el Optotriac MOC3021, el cual también permite aislar el circuito de control del circuito de potencia. El triac que

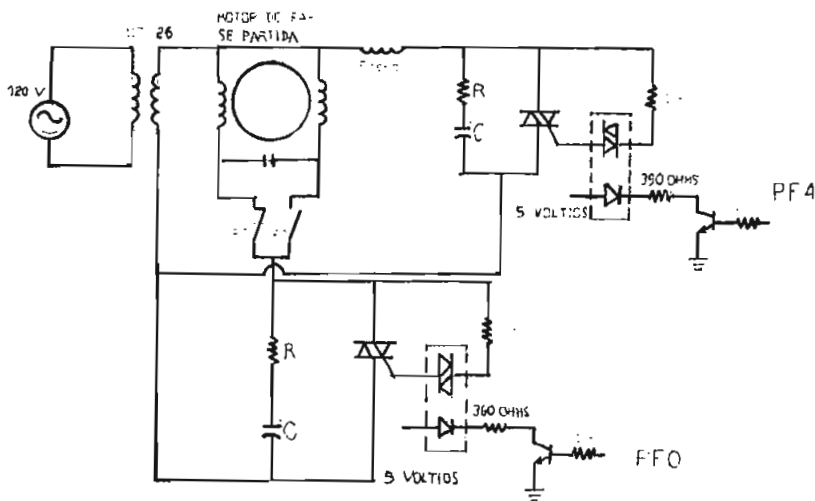


Figura 2.13. Circuito de control para el arranque del motor y activación del tope.

forma parte del optotriac se conecta a la línea de alimentación de voltaje a través de una resistencia limitadora de corriente R_1 , la que debe cumplir con las siguientes condiciones:

$$V = \text{Voltaje pico de alimentación del motor } 24\sqrt{2}V.$$

$$V_{opt} = \text{Voltaje sobre el triac del MCO3021 } 3 V$$

V_{GT1} = Voltaje compuerta – Terminal 1 2.5 V

I_{GTmax} = Corriente máxima que soporta el triac (MOC3021) 100mA

I_{GTmin} = Corriente mínima para que se active el triac (MOC3021) 10mA

$$R_1 \geq \frac{V - V_{opt} - V_{GT1}}{I_{GTmax}} \qquad R_1 \leq \frac{V - V_{opt} - V_{GT1}}{I_{GTmin}}$$
$$R_1 \geq \frac{33.94 - 3 - 2.5}{100 \times 10^{-3}} \Omega \qquad R_1 \leq \frac{33.94 - 3 - 2.5}{10 \times 10^{-3}} \Omega$$
$$R_1 \geq 284.41 \Omega \qquad R_1 \leq 2844.1 \Omega$$

El valor de resistencia escogido es de $1K\Omega$. La potencia disipada por R_1 esta dada por:

$$P_{disipadaR_1} = I_{rms} \times V_{rms} = \frac{V_{rms}^2}{R_1}$$
$$P_{disipadaR_1} = \frac{(24 - 3 - 2.5)^2}{1000} w$$
$$P_{disipadaR_1} = 0.35 w$$

La corriente que pasa por el led del MOC3021 es limitada por la resistencia R_2 , su valor se obtiene a continuación:

$$R_2 = \frac{V_{CC} - V_{led} - V_{CEsat}}{I_{led}}$$
$$R_2 = \frac{5 - 1.2 - 0.2}{10m} \Omega$$
$$R_2 = 360 \Omega$$

Se escoge $R_2 = 390 \Omega$

Para que el rotor pueda girar libremente es necesario quitar el tope que tiene el mismo, el circuito para accionarlo es igual que el utilizado para el arranque del motor, ya que este se alimenta con el mismo valor de voltaje.

2.3.1.1. Funcionamiento del circuito.

De la bandera programable PF4 del microcontrolador ADSP2181 se tiene la señal de control para la activación del tope, la que ingresa a través de una resistencia de $1K\Omega$ a la base del transistor NPN (Q1), el cual se activa cuando la señal que entra es 1L, esto polariza al led del optotriac MOC3021 y a su vez activa al triac interno del mismo produciendo el disparo del triac T1 con lo que se activa el tope.

La señal de arranque del motor es dada por la bandera programable PF0 que ingresa por una resistencia de $1K\Omega$ a la base del transistor NPN (Q2), la que polariza al led del optotriac MOC3021 y a su vez activa al triac interno del mismo produciendo el disparo del triac T2 con lo que inicia el arranque del motor.

Para el apagado del motor se necesita únicamente que la señal de las banderas PF0 y PF4 estén en 0L, con lo que el transistor NPN entra en corte, abriendo el circuito de disparo del triac deteniendo al motor.

Los triacs T1 y T2 cuentan con protecciones contra sobrevoltaje, para evitar disparos involuntarios.

2.3.2. Control del Sentido de Giro

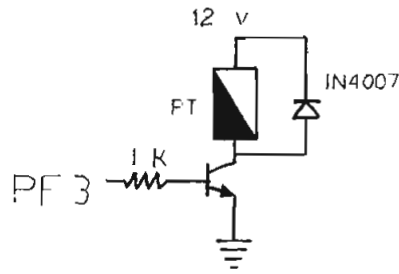


Figura 2.14. Circuito implementado para cambiar el sentido de giro

La bandera PF3 es utilizada para cambiar el sentido de giro del rotor, la cual pone en saturación al transistor Q3, activando el relé. Este relé dispone de un contacto abierto y un contacto cerrado. Tiene conectado en los extremos de la bobina un diodo dando un camino para la circulación de corriente almacenada en la misma cuando el transistor Q3 entra en corte.

2.3.3. Sensor de Posición

Para obtener la posición de la antena se ha utilizado el sensor óptico H13B1, el cual fue adaptado al eje del rotor. El circuito implementado es el siguiente:

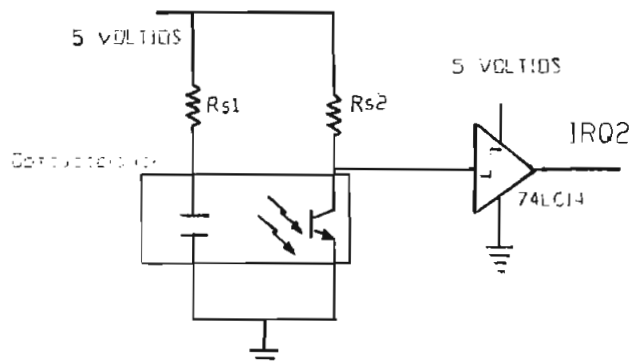


Figura 2.15. Circuito sensor de posición

Las resistencias R_{S1} y R_{S2} limitan corriente en el diodo y en el transistor respectivamente.

Los valores de las resistencias R_{S1} y R_{S2} se los calcula de la siguiente forma:

$$\begin{aligned} R_{S1} &= \frac{V_{CC} - V_{led}}{I_{led}} & R_{S2} &= \frac{V_{CC} - V_{CE}}{I_C} \\ R_{S1} &= \frac{(5 - 1.6)V}{10mA} & R_{S2} &= \frac{(5 - 1)V}{4mA} \\ R_{S1} &= 340\Omega & R_{S2} &= 1K\Omega \end{aligned}$$

Por tanto los valores escogidos son: $R_{S1} = 330\Omega$ y $R_{S2} = 1K\Omega$.

Debido a que el tren de pulsos obtenido presenta distorsiones, se utilizó una compuerta lógica con retención (Schmitt trigger). La señal optimizada ingresa a la interrupción IRQ2 del microprocesador ADSP2181 a una frecuencia de 58Hz.

2.3.4. Fuentes de Polarización

Para la implementación del circuito de control se requiere de dos fuentes de voltaje: 12V y 5V. Estas fuentes se basaron en los circuitos integrados LM7812 y LM7805 que soportan hasta 1 Amperio ya que la corriente total consumida por los circuitos utilizados es de 100 mA aproximadamente. Requieren una entrada mínima de 14.6 V y 7.3 V respectivamente. Este requerimiento se lo ha cubierto con un transformador de dos salidas (13 y 15 Vrms), cuyas señales son rectificadas por puentes de diodos.

Para mantener el voltaje de DC y filtrar cualquier variación de voltaje de alta frecuencia se colocó los capacitores de 680uF y 470uF.

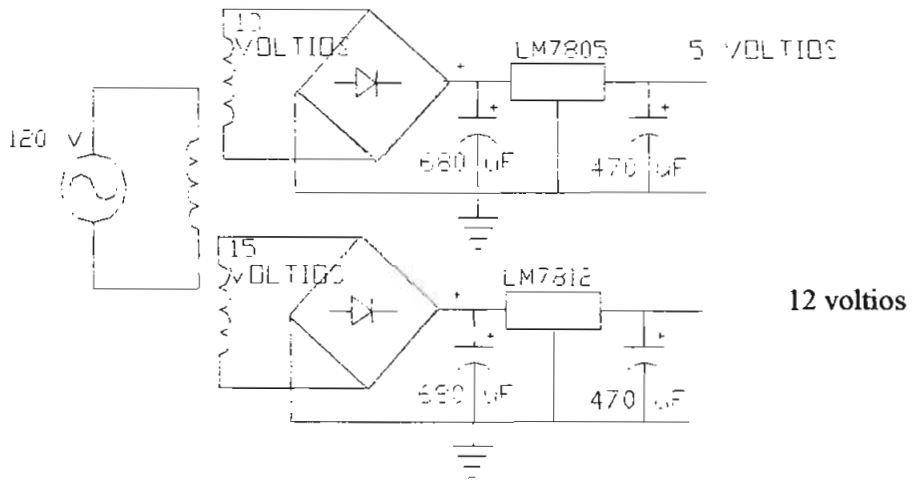


Figura 2.16. Fuentes de Polarización

2.3.5. Circuito de señalización y temporización.

2.3.5.1. Circuito de Señalización

Es necesario que el equipo disponga de un sistema de señalización para que el usuario pueda conocer el estado del mismo, el circuito implementado consta de cuatro leds indicadores del encendido del motor, sentido de giro, transmisión de datos y reset; los mismos que son activados a través de transistores que funcionan como interruptores.

2.3.5.2. Circuito de Temporización

En caso de que el enlace se pierda, es necesario tener un sistema que permita finalizar todo el proceso de adquisición de datos, ya que de no ser así, se corre el riesgo de no obtener los resultados esperados.

El circuito de temporización está diseñado como un monoestable, para que cuando transcurra el tiempo de 1 minuto realice un reset del equipo.

La implementación de este circuito se lo hace en base a un contador modulo 15 (74LS393), al cual ingresa una señal de reloj de frecuencia 0.25Hz que es generada por una compuerta inversora CMOS Smith trigger (74HC14). La señal proveniente del PF0 (Ezkit_lite) activa el proceso de temporización al ingresar al pin 2 (CLR) del contador pasando previamente por una compuerta inversora.

Como se puede observar en la Figura 2.18 la señal de salida del circuito de temporización ingresa a una compuerta OR, así como también la señal proveniente del PF7 del Ezkit_It (reset por software) la salida de esta compuerta activa un Relé que permite resetear el equipo.

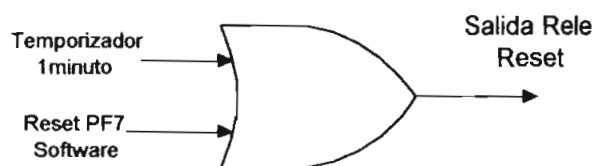


Figura 2.18. Compuerta OR de reset

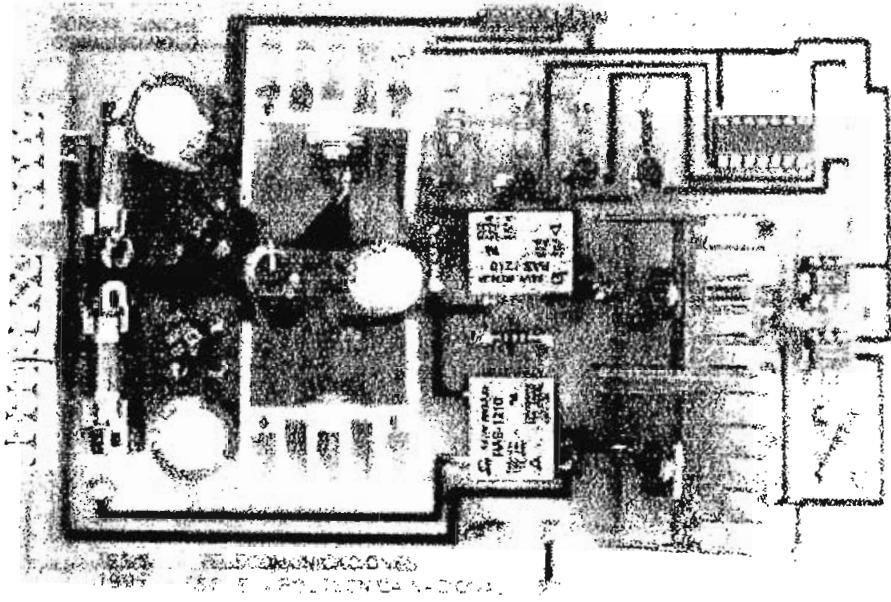


Figura 2.19. Tarjeta implementada para el control del rotor



Figura 2.20. Tarjeta para la señalización y temporización

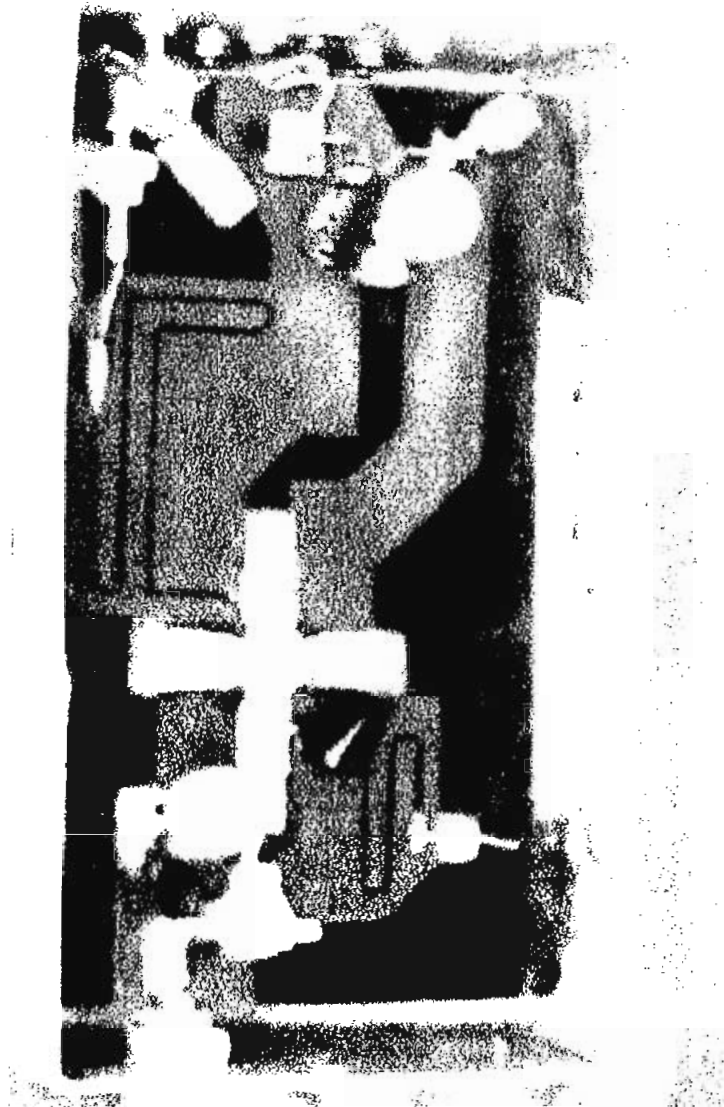


Figura 2.21. Amplificador de potencia de 1w a 450MHz

CAPITULO III

ADQUISICION, PROCESAMIENTO
DE DATOS Y PRESENTACION EN
PANTALLA. REQUERIMIENTOS
DEL SOFTWARE PARA EL
SISTEMA

Para poder obtener la configuración de radiación de una antena en forma experimental se debe conocer el valor del campo eléctrico en función de la posición angular, en el capítulo anterior se describió el hardware utilizado para este propósito, pero como lo que se obtiene es una señal analógica, esta debe digitalizarse para posteriormente ser procesada y enviada al computador por puerto serial.

Para convertir la señal analógica de prueba de 1 KHz a digital se requiere de un conversor A/D, el cual debe ser capaz de muestrear mínimo a 2 KHz, según el teorema de Nyquist. La señal digitalizada debe ser procesada, para lo cual, se requiere un microprocesador que sirva para este propósito, de tal manera que logre tomar y procesar todos los datos enviados por el conversor.

Simultáneamente, debe ser capaz de sensar la posición del motor que hace girar a la antena de prueba. Una vez adquiridos los datos, se realiza la conversión de señales TTL provenientes del microprocesador a niveles RS232 para el envío de datos hacia el computador, donde es adecuada para ser presentada en forma gráfica al usuario.

3.1. Adquisición y procesamiento de datos en base al Ezkit_Lite de Analog Device.

Todos los requerimientos citados anteriormente se los ha cubierto con el módulo de desarrollo que ofrece Analog Device denominado Ezkit_Lite, basado

en el microprocesador ADSP2181¹, el cual permite procesar digitalmente señales analógicas dentro del rango de frecuencias de audio.

3.1.1. Descripción general del Ezkit_Lite.

El módulo de desarrollo Ezkit_Lite básicamente incluye:

- ◆ DSP (ADSP2181) de 33 MIPS.
- ◆ Puerto de sonido estéreo de 16 bits AD1847
- ◆ Interfaz RS232
- ◆ Socket de memoria EPROM
- ◆ Pulsantes de acceso a interrupciones.
- ◆ Regulador de voltaje.
- ◆ Conectores para expansión.
- ◆ Uso de jumpers para configuración.

Este módulo viene con su propio software para la realización de aplicaciones DSP.

A continuación se muestra el módulo de desarrollo:

¹ Mayor información referirse a la Tesis del Ing. Fernando Vázquez. Implementación de un laboratorio básico de DSP

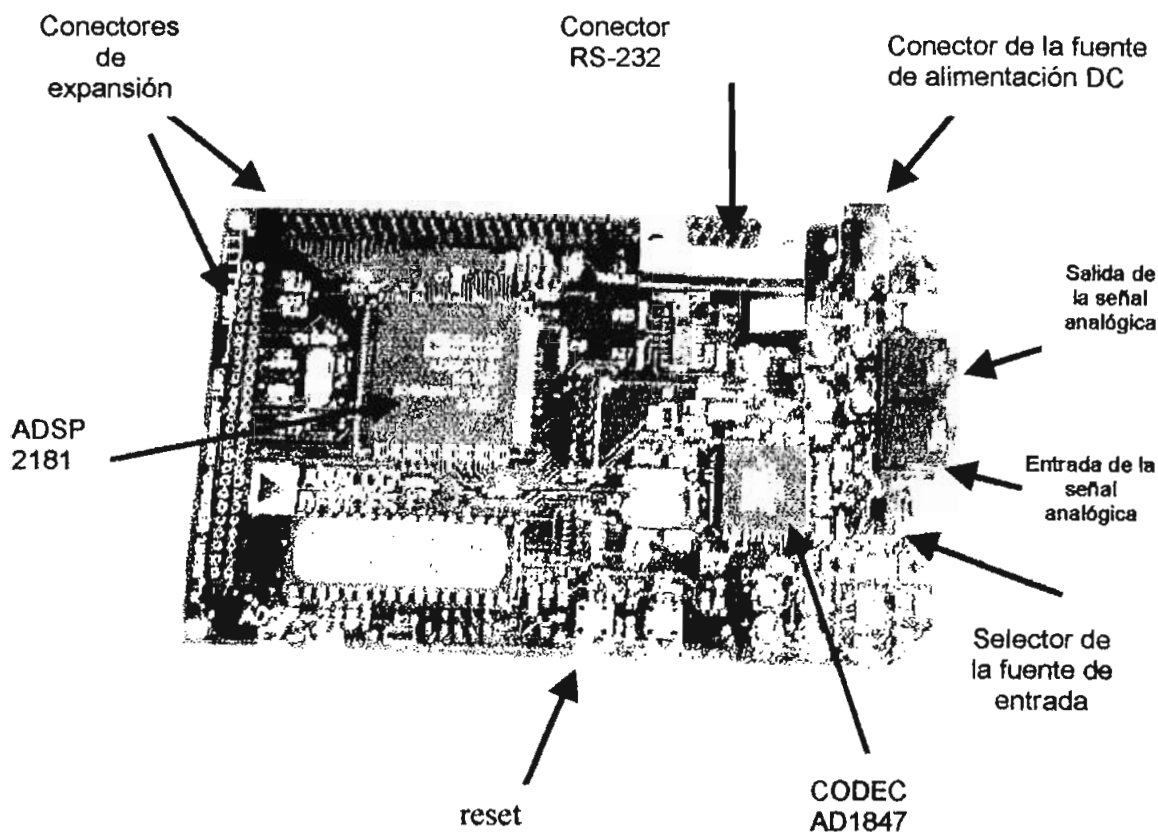


Figura 3.1. Módulo de desarrollo Ezkit_Lite

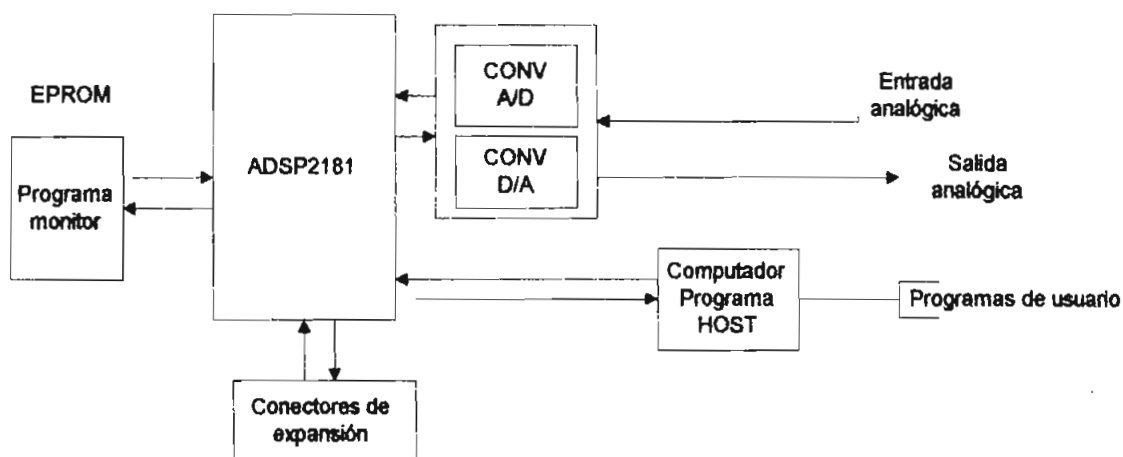


Figura 3.2. Diagrama de bloques del módulo de desarrollo.

El ADSP2181 contiene 12 fuentes de interrupción, de las cuales 4 son externas: IRQL0, IRQL1, IRQ2 y IRQE. La interrupción IRQ2 se la utiliza para

obtener la posición de la antena, mediante la señal generada por el sensor que se encuentra dentro del rotor. Se tiene la posibilidad de tener dos interrupciones externas más al reconfigurar el puerto serial 1, como IRQ0 y IRQ1, las cuales se utilizan para la transmisión serial desde el ADSP al computador en nuestro caso.

Además nos proporciona 13 banderas, de las cuales 3 pueden ser solo salidas y 8 pueden ser configuradas como entrada y salida; estas 8 banderas después de un Reset quedan por defecto configuradas como entradas. Son programadas con el uso de dos Registros que se hallan en las direcciones 0x3fe5 y 0x3fe6 de la memoria.

Se dispone de dos puertos seriales sincrónicos: SPORT0 y SPORT1, cada uno de los puertos tiene una interfaz serial de 5 pines:

NOMBRE DEL PIN	FUNCION
SCLK	Reloj serial
RFS	Recepción de trama de sincronización
TFS	Transmisión de trama de sincronización
DR	Recepción de datos seriales
DT	Transmisión de datos seriales

Tabla 3.1. Pines del interfaz RS-232

Debido a que los puertos seriales del ADSP2181 son sincrónicos y el puerto serial del computador es asincrónico, se realizó la interfaz mediante software, para lo cual se ha utilizado el SPORT1 configurado como interrupción externa (flag_in y flag_out). El ADSP2181 tiene la posibilidad de tener una señal de reloj programable para los SPORTs, para lo cual se utiliza la interrupción denominada TIMER.

3.1.2. Adquisición de datos

La señal a la salida del demodulador es ingresada al CODEC1847, el cual es configurado a la mínima frecuencia de muestreo. La señal digitalizada obtenida tiene una resolución de 16 bits e ingresa al ADSP2181 en una localidad de memoria, para posteriormente ser filtrada, eliminando así señales ajenas a la señal de prueba. A continuación se realiza un proceso de selección de muestras para determinar el nivel máximo dentro de un intervalo de tiempo, el cual es dado por el tren de pulsos del sensor.

3.1.3. Procesamiento de la señal de prueba.

El procesamiento de la señal de prueba utiliza un filtro digital pasabanda, del tipo FIR (Respuesta Impulsiva Finita), de orden 256, cuya descripción viene dada por:

$$y(n) = \sum_{k=0}^{N-1} h(n) \times x(n - k)$$

Donde : N = orden del filtro.
 $h(n)$ = coeficientes del filtro.
 $X(n)$ = señal de entrada a filtrarse.
 $Y(n)$ = señal de salida filtrada.

Los coeficientes utilizados en la implementación de este filtro fueron tomados del programa FIRDEMO que trae el Ezkit_Lite, para un filtro pasabanda. La banda de paso va de 800 Hz a 1200 Hz.

3.1.4. Software implementado en el Ezkit_Lite.

El software utilizado para el sistema de adquisición de datos está desarrollado en lenguaje ensamblador propio del DSP. Para ésta aplicación se desarrolló un programa que permite posicionar el rotor de antena y otro para la adquisición de datos.

3.1.4.1. Software para el posicionamiento del rotor de antena.

El posicionamiento del rotor de antena se lo realiza mediante el programa denominado POSICION.EXE, el cual es resultado del enlace de dos programas POSICION.DSP y UART.DSP. Este programa permite realizar un control de la posición en forma manual o automática desde el computador.

Para establecer la comunicación entre el PC y el Ezkit_Lite se utiliza como programa auxiliar el UART.DSP, el que nos permite realizar la conversión de puerto serial sincrónico a asincrónico. Este programa está compuesto de varias

subrutinas que realizan básicamente funciones como son: configurar el puerto de comunicaciones, enviar y recibir caracteres. El programa POSICION.DSP llama a estas subrutinas conforme a sus requerimientos, por ejemplo: cuando se quiere enviar un caracter desde el DSP al PC se llama a la subrutina Out_Char_AX1, mientras que para el caso inverso se llama a la subrutina Get_Char_AX1.

El control manual que realiza este programa, permite que el usuario dependiendo de la selección que realice haga girar al rotor en sentido horario o en sentido antihorario. El ángulo de desplazamiento de la antena es de 10 grados por cada pulsación, por lo que se requeriría de 36 pulsaciones para que de una vuelta completa.

El control automático permite ubicar a la antena de prueba en el punto donde el nivel de recepción de la señal es máximo, esto lo consigue girando 360° donde obtiene la posición del valor máximo de la señal, para posteriormente girar en sentido contrario y ubicarse en esta posición. El tiempo que le toma al rotor dar una vuelta completa es de 46 segundos

El programa terminará su ejecución cuando desde el computador se envíe la señal que activa la bandera que realiza el reset del equipo.

El diagrama general de bloques del programa implementado se muestra en la figura 3.3.

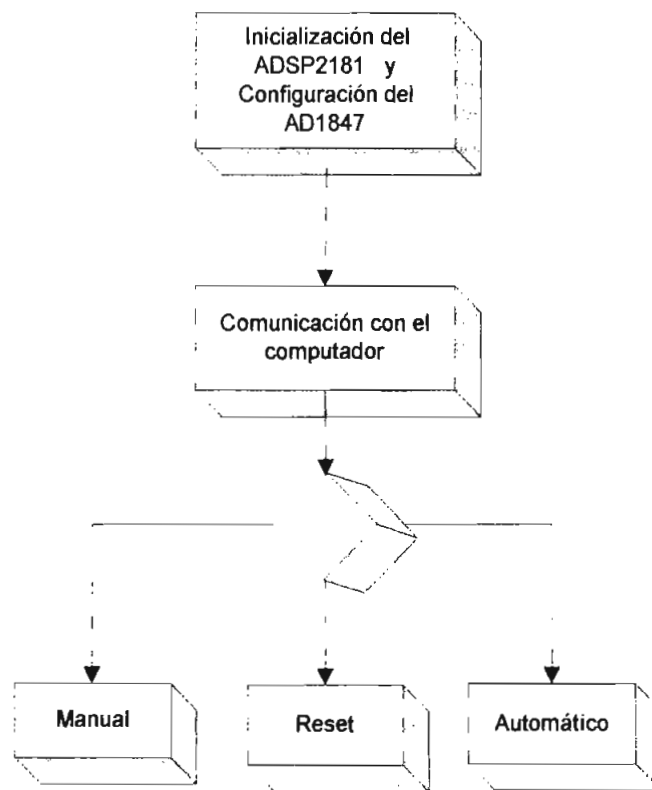


Figura 3.3. Diagrama de bloques del programa Posición.exe

3.1.4.1.1. Inicialización del ADSP2181 y configuración del AD1847.

El proceso de inicialización requiere configurar el modo de funcionamiento del ADSP2181 y del CODEC AD1847. La configuración del CODEC se la realiza en el ADSP2181 con los registros indexados que se disponen para este propósito. En estos registros es posible configurar parámetros como: frecuencia de muestreo, factor de ganancia de la señal, formato de datos para realizar la digitalización de la señal, habilitación y deshabilitación de los canales de entrada y salida de la señal analógica.

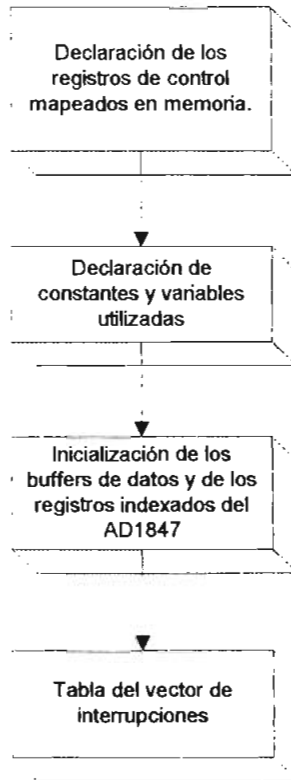
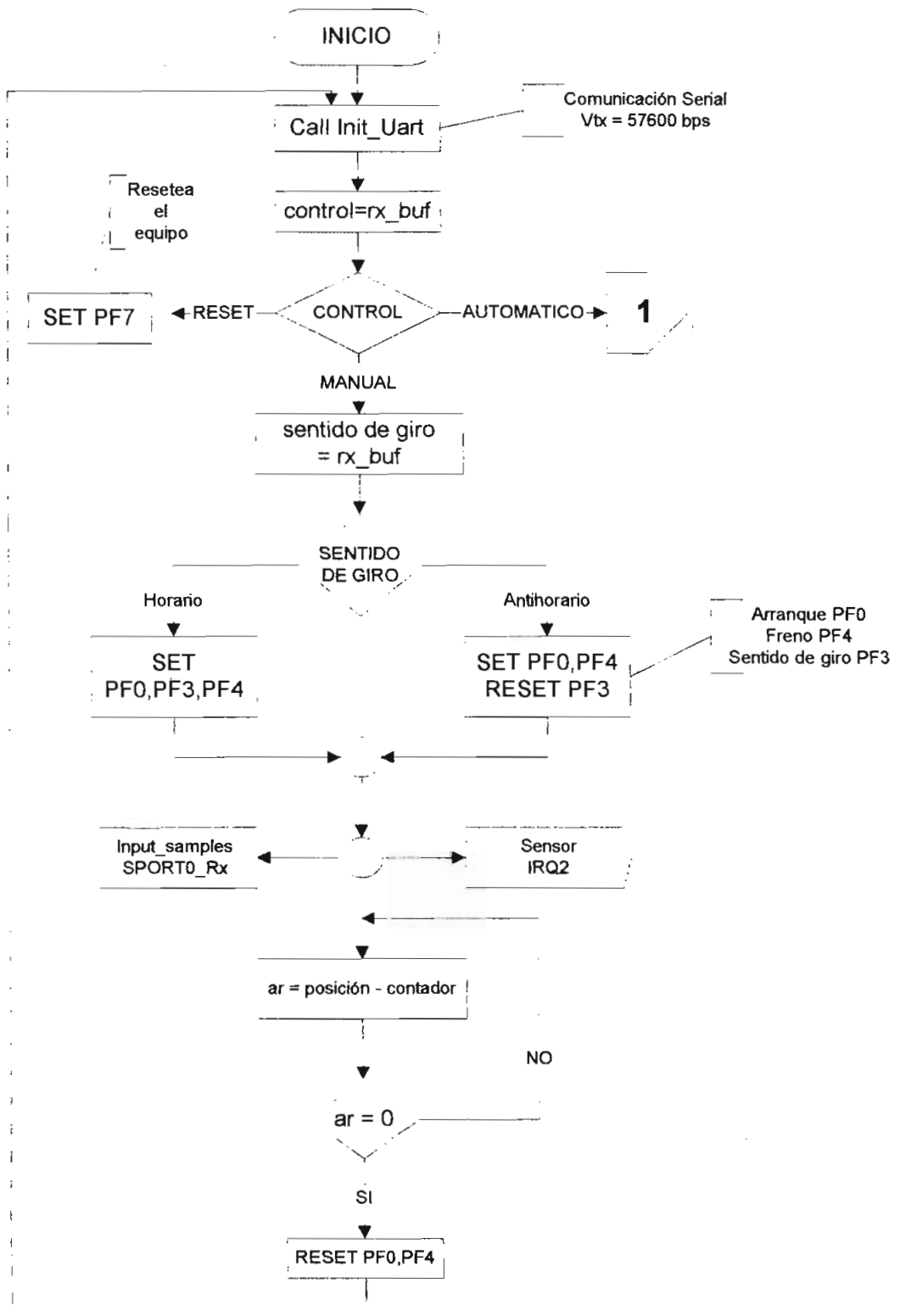


Figura 3.4. Esquema de la inicialización del ADSP2181 y el CODEC AD1847

Este procedimiento configura al AD1847 con una frecuencia de muestreo de 5.5 KHz, y una ganancia unitaria en la salida.

En la figura 3.5 se presenta el diagrama de flujo del programa posición.dsp. el mismo que después del proceso de inicialización descrito anteriormente, queda en un lazo de espera hasta que llegue un carácter, luego de lo cual, se lo somete a una comparación que dará paso a la selección de una de las opciones que dispone el programa.

Proceso para posicionar el rotor de antena



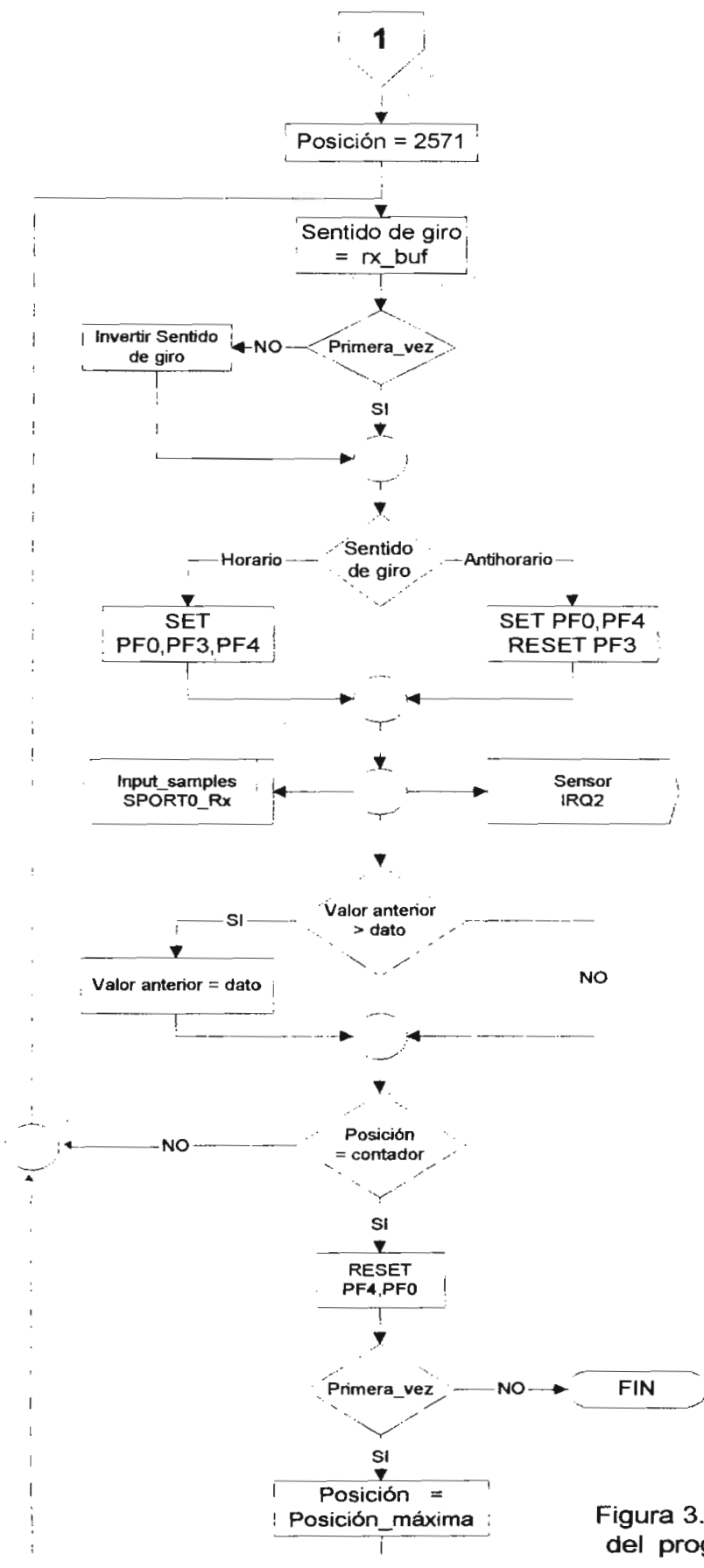


Figura 3.5. Diagrama de flujo del programa Posición.dsp

Si se escogió el modo manual, el programa se queda en un lazo de espera de un carácter que le indique el sentido de giro seleccionado, luego de lo cual arranca el rotor de antena e ingresa a un lazo mientras el valor guardado dentro de la variable posición sea igual al del contador, este valor es de 71 lo que permite que el rotor gire 10° cuantas veces el usuario lo desee. El contador es incrementado cuando se atiende a la interrupción IRQ2.

En caso de escoger el modo automático el proceso de selección del sentido de giro es similar al del control manual. El valor guardado en la variable posición es de 2571, lo que hace que el rotor gire 360° con el objeto de obtener la posición en donde la señal de recepción es máxima. Luego de obtener esta posición el rotor gira en sentido contrario al inicial hasta que el contador sea igual a este valor. El valor de posición en donde la señal es máxima se lo obtiene a través de una serie de comparaciones sucesivas realizadas cuando se atiende a la interrupción del IRQ2. Los datos digitalizados de la señal de prueba obtenidos desde el CODEC a través del SPORT0 son tomados cuando se atiende a la interrupción de recepción de este puerto.

Cuando se recibe desde el computador el carácter que da la orden de Reset se activa la bandera PF7, que resetea el equipo.

A continuación se presentan los diagramas de flujo de las llamadas a las principales interrupciones utilizadas en el programa

Llamada a la interrupción IRQ2 - Sensor

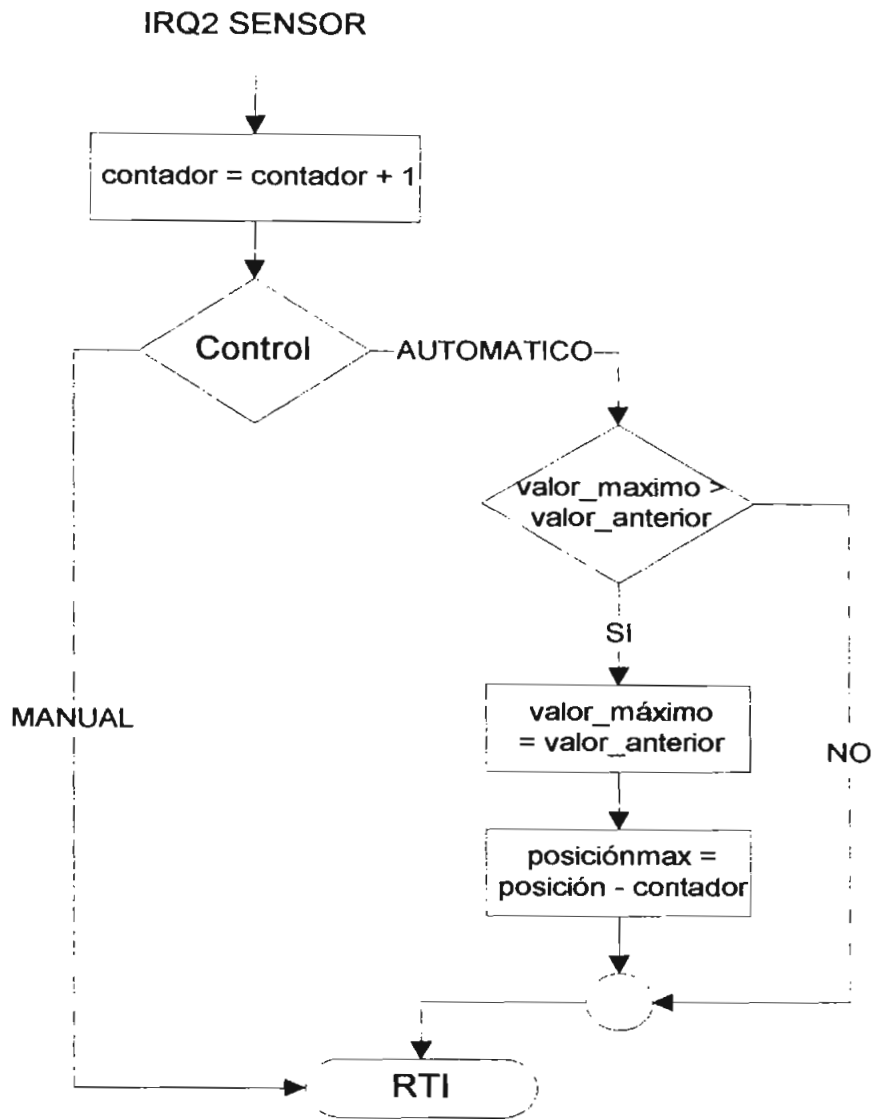


Figura 3.6. Diagrama de flujo de la atención a la interrupción IRQ2

Llamada a la interrupción del SPORT0 de Recepción

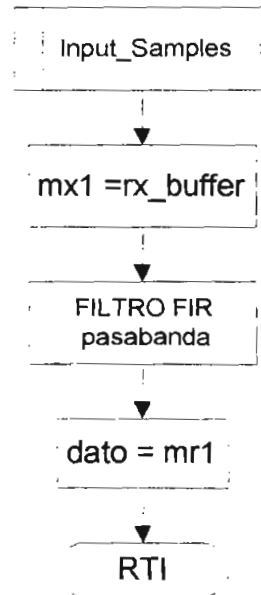


Figura 3.7. Diagrama de flujo de la atención a la interrupción Input_samples

3.1.4.2. Software para la Adquisición de datos

El programa implementado para la adquisición de datos es el denominado ADQUISIC.EXE, el cual como en el caso anterior, es el resultado de enlazar los programas Adquisición.dsp y UART.dsp.

Este programa está diseñado de tal manera que una vuelta completa del rotor, es suficiente para adquirir los datos que permiten obtener la configuración de radiación de la antena de prueba.

Luego del proceso de inicialización, el programa se mantiene en un lazo de espera hasta que llegue el carácter que le da la orden de arranque al rotor de antena. De la misma forma que el caso automático, la variable posición se almacena con un valor de 2571 para que el rotor gire 360 grados.

A medida que va girando el rotor se toma muestras de la señal, la frecuencia de muestreo utilizada para este proceso es de 5.5KHz, la que permite obtener un número representativo de datos. El número de datos seleccionados de todo este grupo son 2571, lo que implica que se está almacenando un dato por cada pulso del sensor, lo que equivale aproximadamente a tomar 7 datos por grado.

El conjunto de datos seleccionados tiene una precisión de 16 bits en formato 1.15, lo que implica tener un número con una cifra entera y 15 cifras decimales, llegando a obtener valores comprendidos en el rango de ± 1 voltio. En el sistema implementado no se necesita tener una precisión tan elevada, por lo que los datos son transformados a formato 1.7, es decir 1 entero y 7 cifras decimales. Esto se lo consigue a través de un proceso de desplazamiento, luego de lo cual se almacenan los resultados en una tabla dispuesta para el efecto siendo posteriormente enviados al computador.

Los datos son transmitidos a partir del valor máximo, para tal efecto se busca la ubicación dentro de la tabla del mismo, y se inicia la transmisión desde esta ubicación. Este proceso es muy simple debido a que el ADSP2181 nos da la posibilidad de almacenar datos en forma circular.

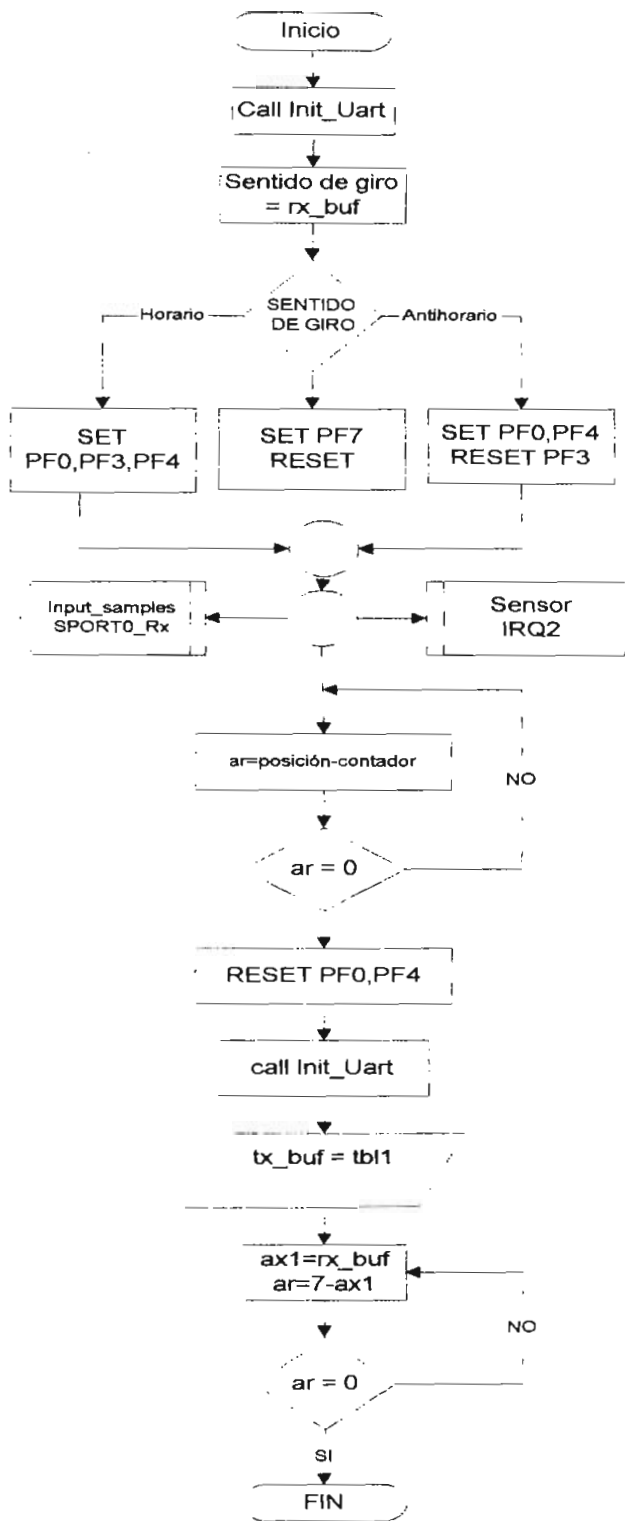


Figura 3.8. Diagrama de flujo del programa Adquisic.dsp.

Llamado a la interrupción IRQ2.

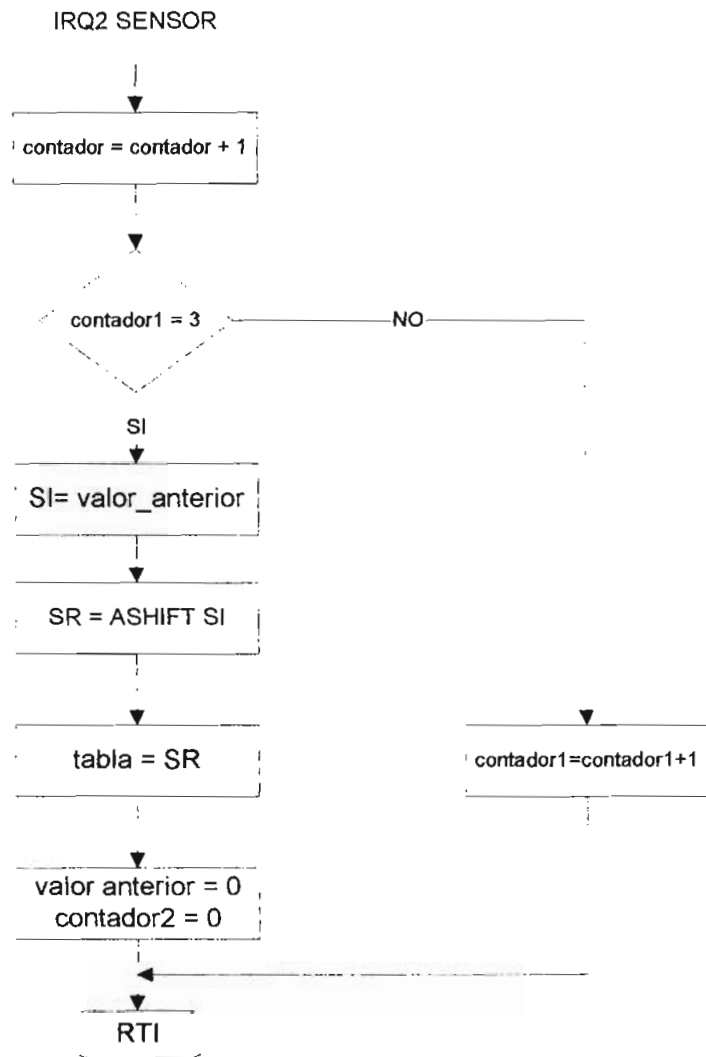


Figura 3.9. Diagrama de flujo de la atención a la interrupción sensor.

Dentro de este programa al igual que en el modo automático del programa posicion.dsp, se atiende a la interrupción del SPORT0 de transmisión (Input_samples) cuyo diagrama de flujo se lo presenta en el figura 3.7.

La implementación de estos programas en el módulo de desarrollo se la puede realizar sin necesidad de grabar los mismos en la memoria EPROM de

este, por lo que se ofrece flexibilidad para ejecutar varios programas independientes.

3.2. Software para la presentación de resultados en pantalla en base al lenguaje de programación Visual Basic 5.0.

La interfaz de usuario es una de las partes más importantes de una aplicación, debe ser amigable y de fácil utilización. Este objetivo se lo ha cubierto a través del lenguaje de programación Visual Basic 5.0, el cual permite desarrollar aplicaciones en ambiente Windows creando la interfaz visual con el usuario, Visual Basic es un lenguaje de fácil programación.

3.2.1. Generalidades del lenguaje de programación

Visual Basic proporciona un juego completo de herramientas que facilitan el desarrollo rápido de aplicaciones. Contiene centenares de instrucciones, funciones y palabras clave, muchas de las cuales están directamente relacionadas con la interfaz gráfica de Windows. Es una herramienta de programación muy potente y de fácil implementación.

La escritura de programas en Visual Basic involucra dos pasos:

- ◆ Programación visual.
- ◆ Programación del código.

Durante el primer paso se puede diseñar programas mediante las herramientas que vienen con Visual Basic. La programación visual no requiere escribir código, únicamente se debe operar y usar estas herramientas. En el paso de programación del código se escribe un programa que está compuesto de instrucciones en el lenguaje de programación Visual Basic.

3.2.2. Diseño del proyecto para la presentación del lóbulo de radiación

La interfaz visual entre el sistema de adquisición de datos y el usuario debe permitir:

- Establecer la comunicación entre el computador y el Ezkit_lite.
- Transformar los datos que vienen en formato ASCII a valores de voltaje.
- Presentar en pantalla la gráfica voltaje vs posición angular que corresponde a la configuración de radiación polar o rectangular.
- Guardar los valores en un archivo
- Imprimir los resultados obtenidos
- Abrir archivos existentes
- Simular configuraciones de radiación de antenas típicas
- Obtener ayuda sobre la utilización del programa

Para cumplir con estos requerimientos se ha desarrollado el programa denominado CONFRAD.EXE, cuyo diseño se lo realiza mediante la creación de un proyecto. Este proyecto se lo puede representar en forma general con el siguiente diagrama de bloques.

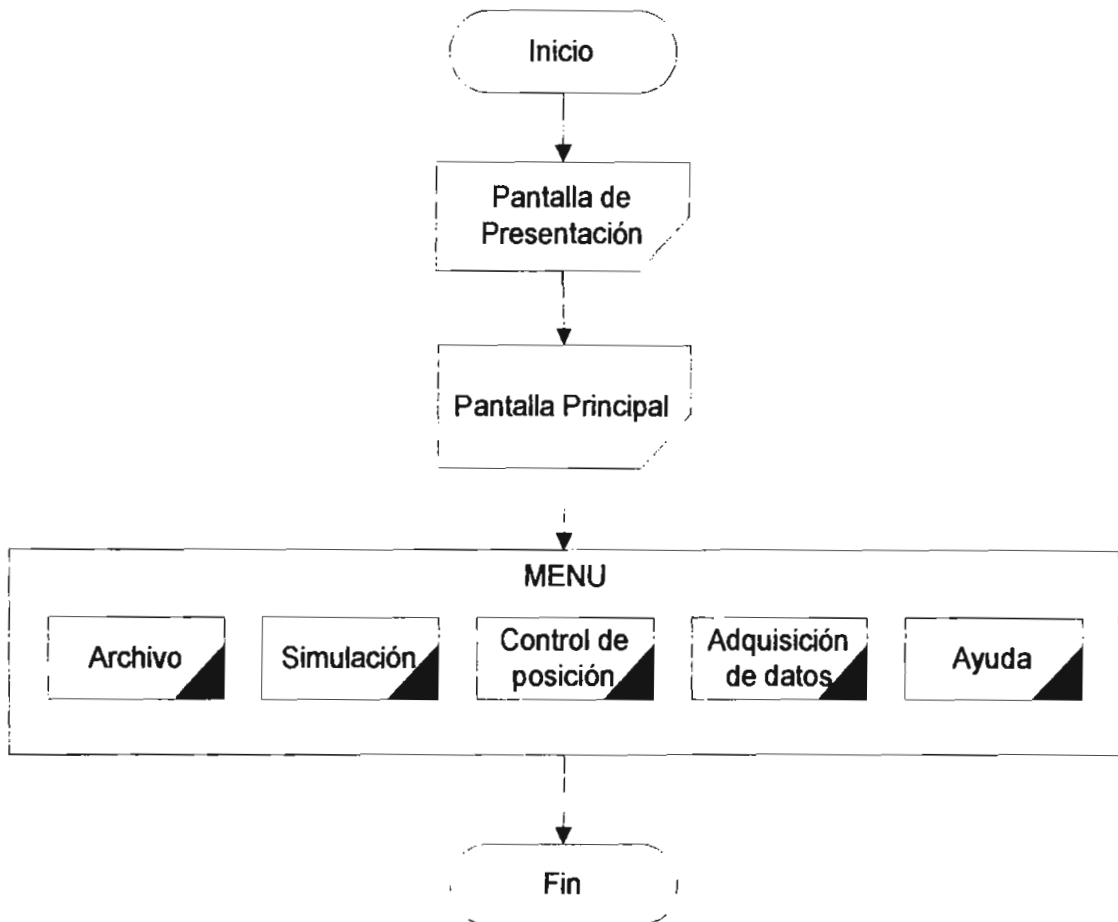


Figura 3.10. Diagrama del proyecto CONFRAD.vbp

3.2.2.1. Diseño de la pantalla de presentación

La pantalla de presentación del CONFRAD fue elaborada con la ayuda del programa Microsoft Power Point.

3.2.2.2. Diseño de la pantalla principal

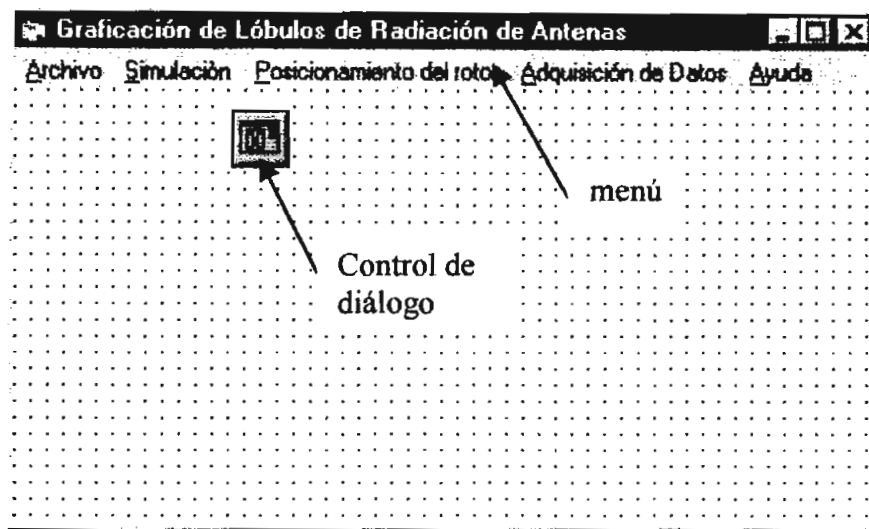


Figura 3.12. Pantalla principal del proyecto

En esta pantalla se ha creado una barra de menú la cual contiene las opciones que puede ejecutar este programa. Además dispone de un control de diálogo el que permite una interfaz visual entre Visual Basic y Windows.

A continuación se detallan cada una de las opciones de la barra de menú:

Menú Archivo

Permite abrir archivos existentes de configuraciones de radiación de antenas obtenidos por el CONFRAD, ofreciendo al usuario dos opciones: obtener información en modo texto (Datos) y obtener información en modo gráfico (Imagen).

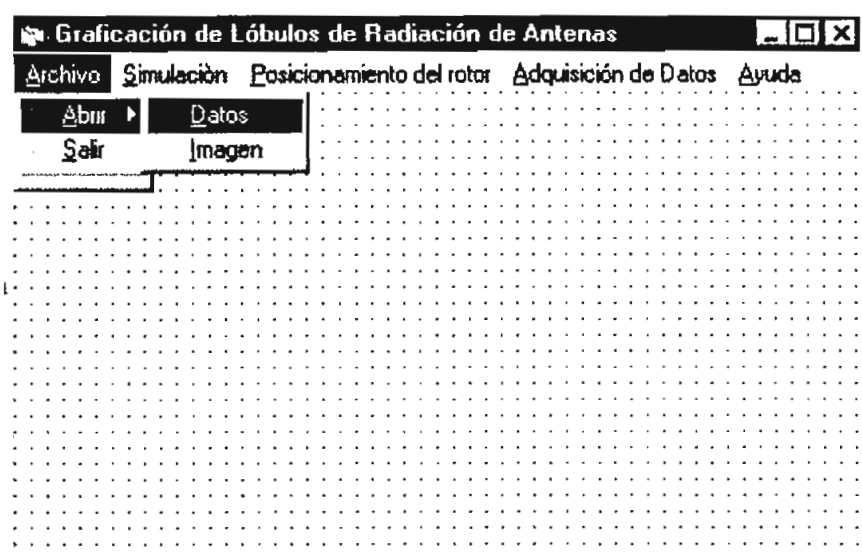


Figura 3.13. Opciones del menú Archivo.

Luego de seleccionar cualquiera de las dos opciones, a través del control de dialogo se presenta una pantalla, la cual permite escoger el archivo que se desea abrir.

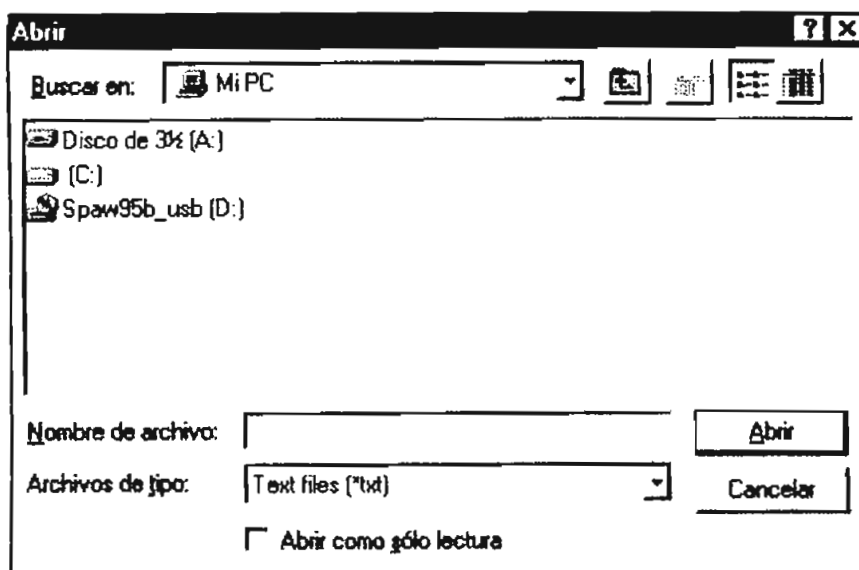


Figura 3.14. Pantalla para seleccionar el archivo

La información se presenta en modo texto en la siguiente pantalla.

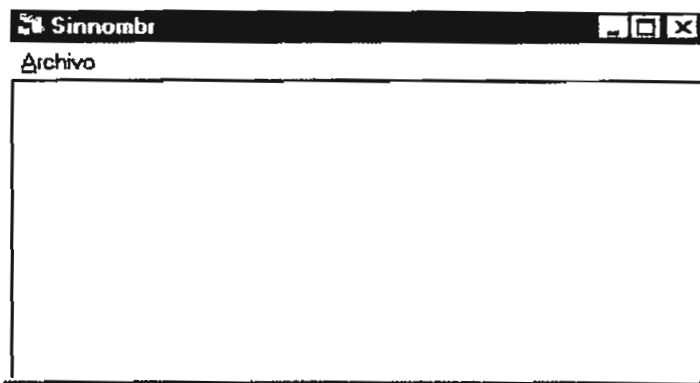


Figura 3.15. Pantalla para abrir archivo de datos

Con la opción de imagen y luego de seleccionado el archivo, se llama a la función `AbrirArchivoImagen()` que se encuentra en el módulo 2, la que se encarga de presentar la pantalla mostrada en la figura 3.21, en donde al presionar el botón `Graficar`, se inicia con la lectura del archivo para posteriormente ser presentado en pantalla a través de la instrucción `PSET`.

Menú Simulación

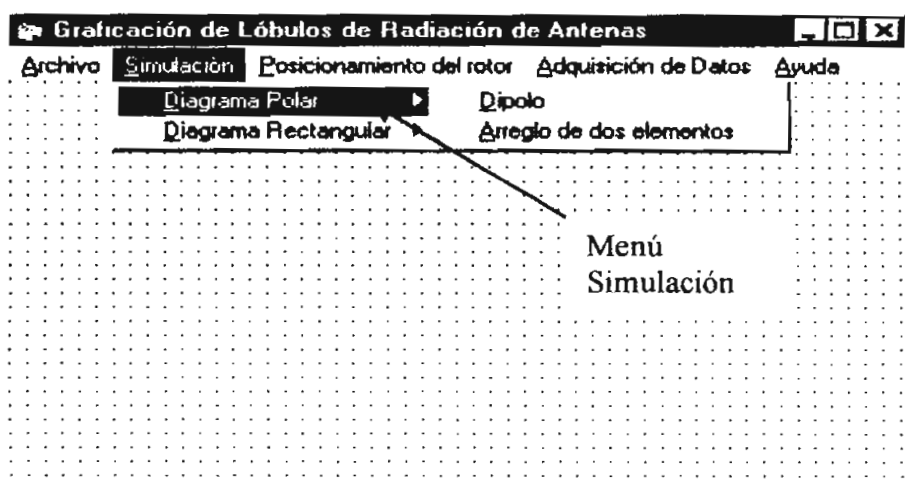


Figura 3.16. Opciones del menú simulación

Permite simular configuraciones de radiación de: antenas dipolo y arreglos de dos elementos en forma rectangular o polar. A continuación se presentan las formas implementadas para este propósito:

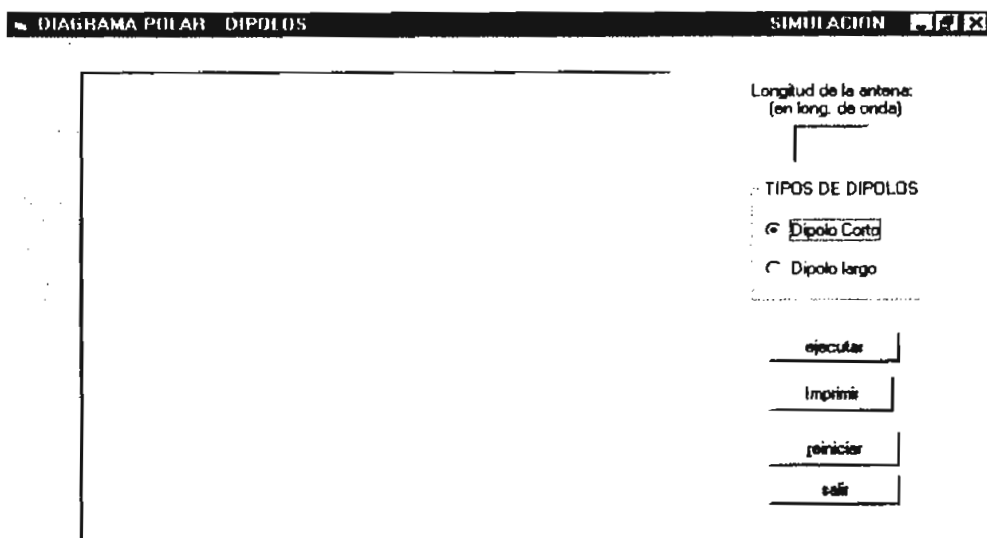


Figura 3.17. Pantalla Diagrama de radiación dipolo

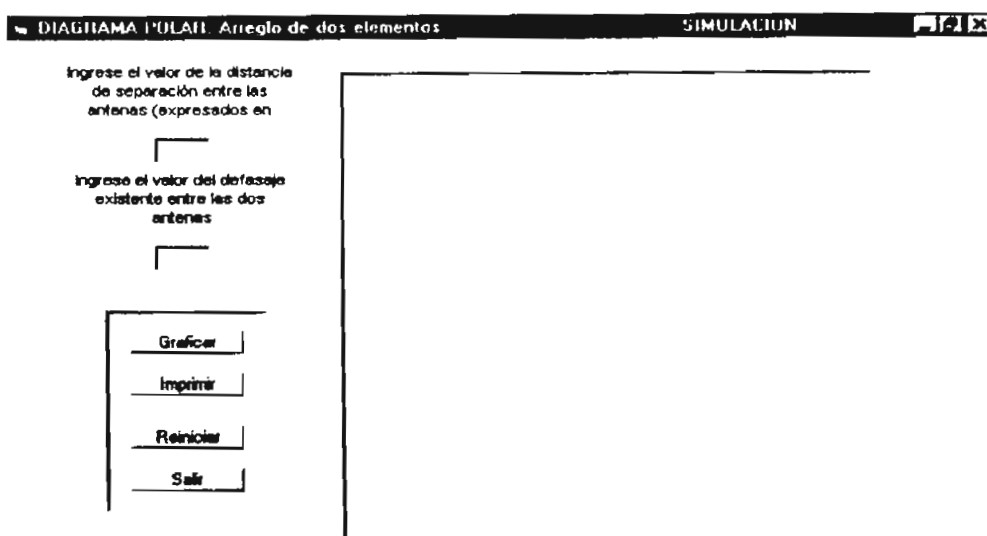


Figura 3.18. Pantalla diagrama de radiación arreglo de dos elementos.

En la pantalla principal al escoger la opción de simulación en coordenadas polares o rectangulares tanto para dipolos como para arreglos de dos elementos, se inicia un proceso de cálculo tras hacer click en el botón graficar, basado en las expresiones matemáticas indicadas en el capítulo I. Los valores ingresados de longitud del dipolo en caso de haber escogido la opción *dipolo* o el valor de la separación entre los elementos y el ángulo de desfase en caso de haber escogido la opción *arreglo de dos elementos* son parámetros utilizados para efectuar estos cálculos. El resultado se lo presenta en forma gráfica a través de la instrucción PSET.

Menú Posicionamiento del rotor

Al seleccionar esta opción se ingresa a la siguiente pantalla:

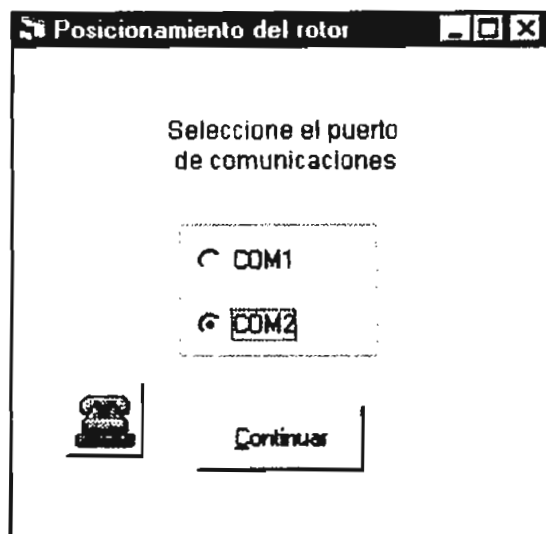


Figura 3.19. Pantalla del control de posición del rotor de antena, configuración del puerto utilizado.

En esta pantalla se selecciona el puerto de comunicaciones a utilizarse, y se llama a la aplicación del Ezkit_Lite, donde se escoge el archivo posición.exe. Luego de descargado este programa se cierra esta aplicación y se presenta la siguiente pantalla:

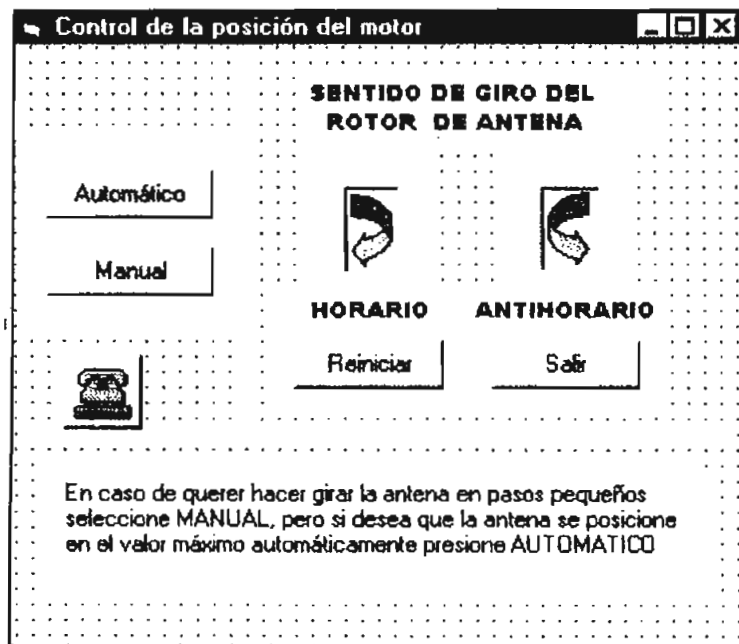


Figura 3.20. Pantalla del control de posición del rotor de antena

En esta pantalla se presentan dos opciones Manual y Automático. Cuando se realiza click en el botón Manual, se abre el puerto de comunicaciones y se lo configura para una velocidad de transmisión de 57600 bps, luego de lo cual se envía el carácter "3" que indica al Ezlit_lite que se seleccionó esta opción, y se desactiva el botón del modo automático. En este punto, el usuario está listo para hacer girar a la antena dependiendo en cual de las flechas realice el click, consiguiendo con ello moverla en sentido horario o antihorario. Para esto se envía al Ezkit_lite el caracter "9" o "1" respectivamente.

Si el click se lo realiza en el botón Automático, se envían dos caracteres, el primero que indica al Ezkit_lite que se escogió el modo automático y el segundo que le indica el sentido de giro. La selección del sentido de giro se la realiza de tal forma que no le permite girar más de 360° en un mismo sentido

El botón reiniciar permite volver a escoger cualquiera de las dos opciones descritas anteriormente.

El botón salir envía al Ezkit_lite el caracter "7", con el cual resetea el equipo, finalizando el control de posición de la antena.

Menú Adquisición de Datos

Al ingresar en este menú se presenta la pantalla de configuración del puerto del punto anterior, luego de lo cual se le pide al usuario que ingrese el nombre del archivo con el cual desea que se guarden los datos adquiridos. A continuación se llama a la aplicación del Ezkit_Lite en donde se descarga el programa Adquisic.exe. Terminado este proceso y luego de haber cerrado la aplicación del Ezkit_Lite se ingresa a la siguiente pantalla mostrada en la figura 3.21.

En esta pantalla se selecciona los valores inicial y final de la posición angular que se desea graficar. A continuación se selecciona el tipo de coordenadas en que se presentará la gráfica.

Al presionar el botón graficar se inicia el proceso de comunicación con el Ezkit_Lite enviando el carácter que indica el sentido en el que debe girar el rotor, quedándose en un lazo de espera hasta que el Ezkit_Lite envíe el carácter "2" que le indica que inicia la transmisión de datos. Los datos recibidos son transformados a su equivalente en voltaje y normalizados, para posteriormente ser almacenados en el archivo escogido por el usuario.

Cada vez que ingresa un dato en el PC, se envía un carácter que indica al Ezkit_Lite que el dato fue recibido, y que está listo para recibir el siguiente dato, con lo cual se asegura que los datos que le llegan son los correctos.

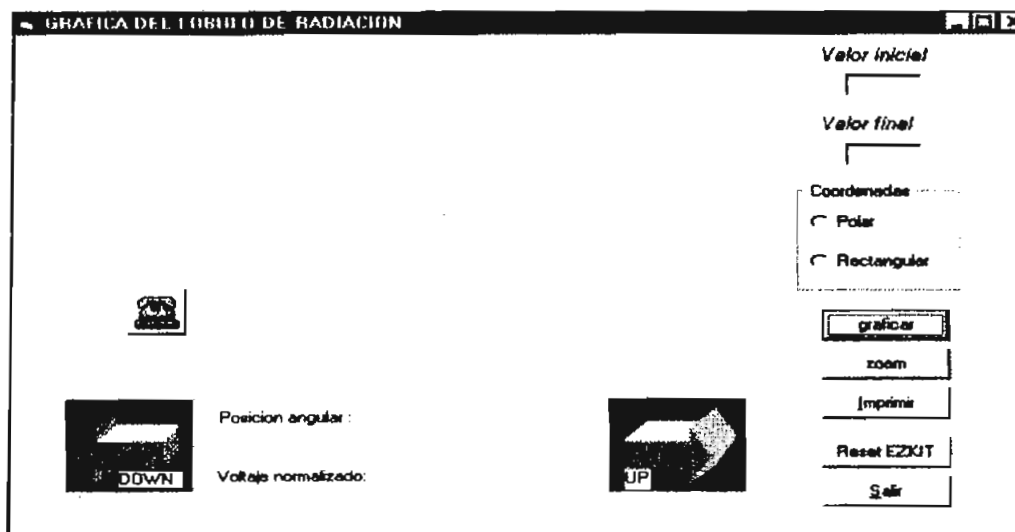


Figura 3.21. Pantalla para la graficación de configuraciones de radiación.

En cada pantalla, se tiene la opción de imprimir las configuraciones de radiación, para esto se utiliza el comando Printer.

El código de programa se encuentra en el Anexo D.

El programa además de las pantallas dispone de dos módulos principales módulo1.bas y módulo2.bas.

Módulo1: contiene declaraciones de variables utilizadas en forma común por algunas pantallas. Permite además adecuar los valores adquiridos en el puerto serial a valores de voltaje.

Módulo2: Permite abrir archivos en modo texto a través de `AbrirArchivo()`, y en modo gráfico por medio de `AbrirArchivolmagen()`.

CAPITULO IV

ANALISIS DE RESULTADOS,
CONCLUSIONES Y
RECOMENDACIONES

4.1. Análisis de resultados obtenidos en el sistema de adquisición de datos implementado.

Como la antena de recepción utilizada para la realización de las pruebas fue un dipolo $\lambda/2$, se facilita la comparación de la configuración de radiación experimental con los datos simulados

Para obtener configuraciones de radiación experimentales de antenas se debieron tomar en cuenta las siguientes consideraciones:

- ◆ Distancia de separación entre antenas.
- ◆ Nivel de señal máximo que ingresa al sistema de adquisición.

La distancia de separación para que la antena de prueba cumpla con la condición de campo lejano es:

$$R > \frac{2L^2}{\lambda}$$

Como la antena de prueba es un dipolo de longitud $\lambda/2$, el $R > 0.33$ metros, por lo que la distancia escogida para las pruebas fue 10 veces mayor a este valor, es decir 3 metros.

El nivel máximo de la señal de entrada con el que se obtuvo un resultado óptimo en las configuraciones de radiación fue de 1 voltio pico ya que sobrepasado este nivel se distorsionan por las limitaciones existentes en el módulo de desarrollo Ezkit_Lite.

Cuando la antena de prueba está polarizada horizontalmente, la configuración de radiación en coordenadas polares de un dipolo $\lambda/2$ obtenida por el sistema de adquisición en forma experimental, es la presentada en la figura 4.1 y en forma teórica en la figura 4.2.

Al comparar estas figuras se puede observar que en el caso del gráfico teórico el nivel máximo de señal se halla en 90° , mientras que en el experimental este nivel está en 0° ; esto se debe a que para el caso experimental siempre el nivel de referencia se lo toma en 0° que corresponde al nivel máximo de la señal. Para este caso el nivel de 0° del gráfico experimental debe coincidir con nivel máximo de la señal en el caso teórico para que las gráficas puedan ser comparadas.

Como se puede observar la configuración de radiación experimental obtenida por el sistema de adquisición de datos se asemeja a la teórica, lo que nos indica que los datos obtenidos son correctos, con lo que se cumple el objetivo de esta tesis. Una de las causas de las diferencias existentes al realizar el análisis comparativo, se debe a que los equipos de laboratorio utilizados ya no funcionan en forma óptima, lo que produce alteraciones en la señal que ingresa al sistema de adquisición de datos. Además se tiene que los lóbulos no son simétricos, esto se debe a factores tales como:

- La forma como está sujetado el balun a la antena y el eje sobre el cual gira la misma.

COORDENADAS POLARES

Dipolo 1/2 longitud de onda $f=450\text{MHz}$

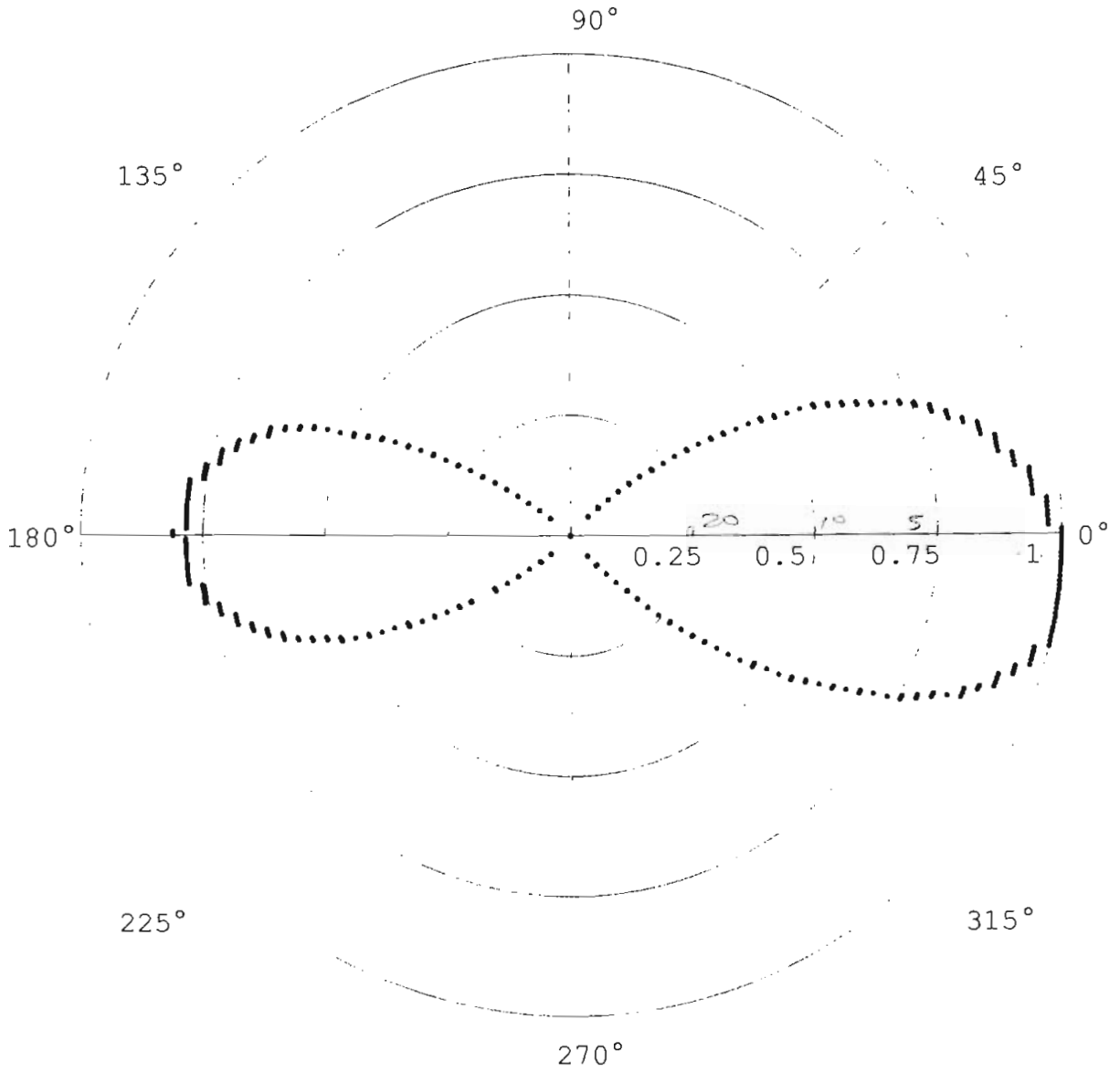
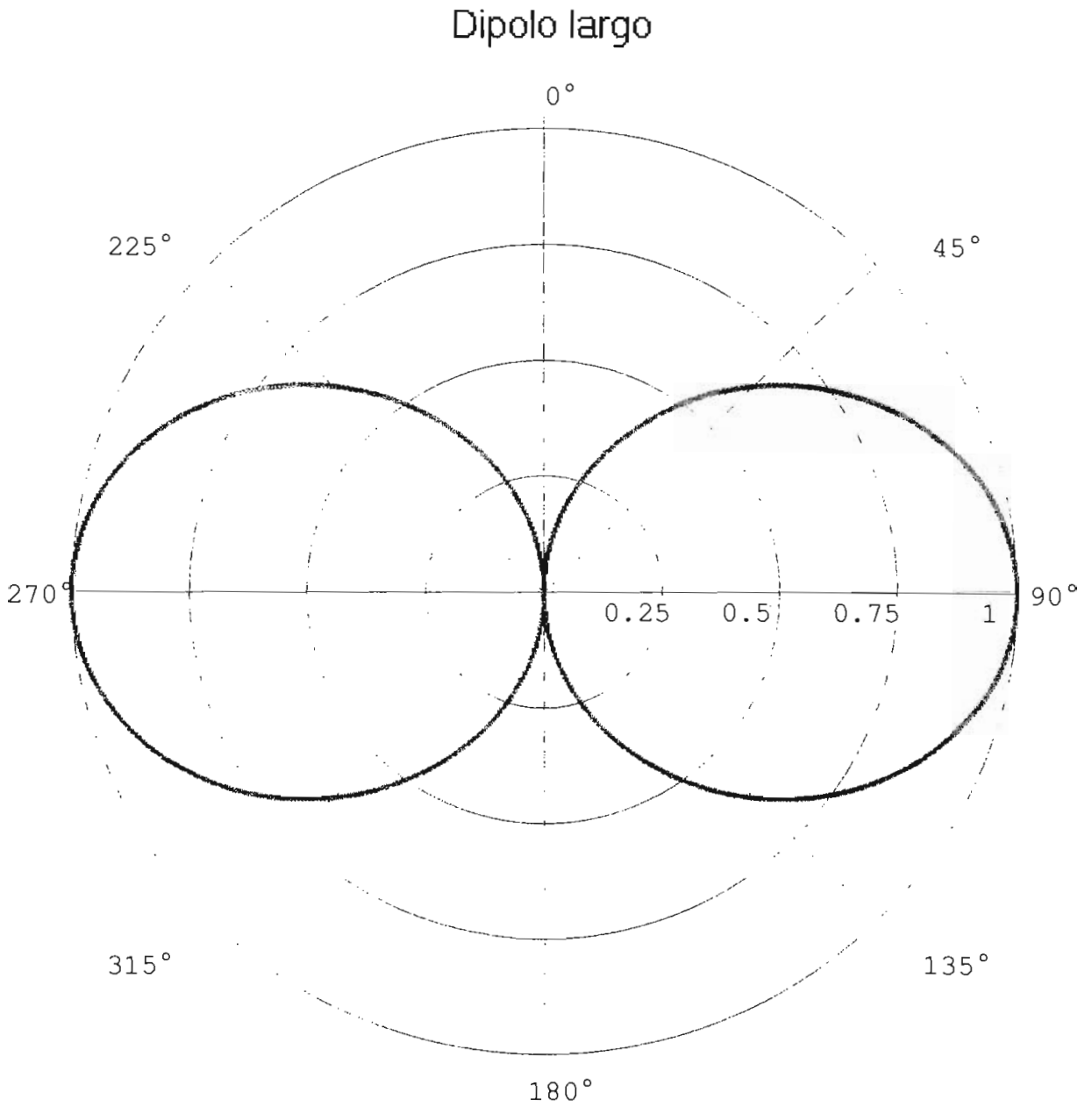


Figura 4.1. Diagrama lineal de la antena de prueba

COORDENADAS POLARES



Longitud del dipolo =0,5

Figura 4.2. Diagrama lineal de la simulación

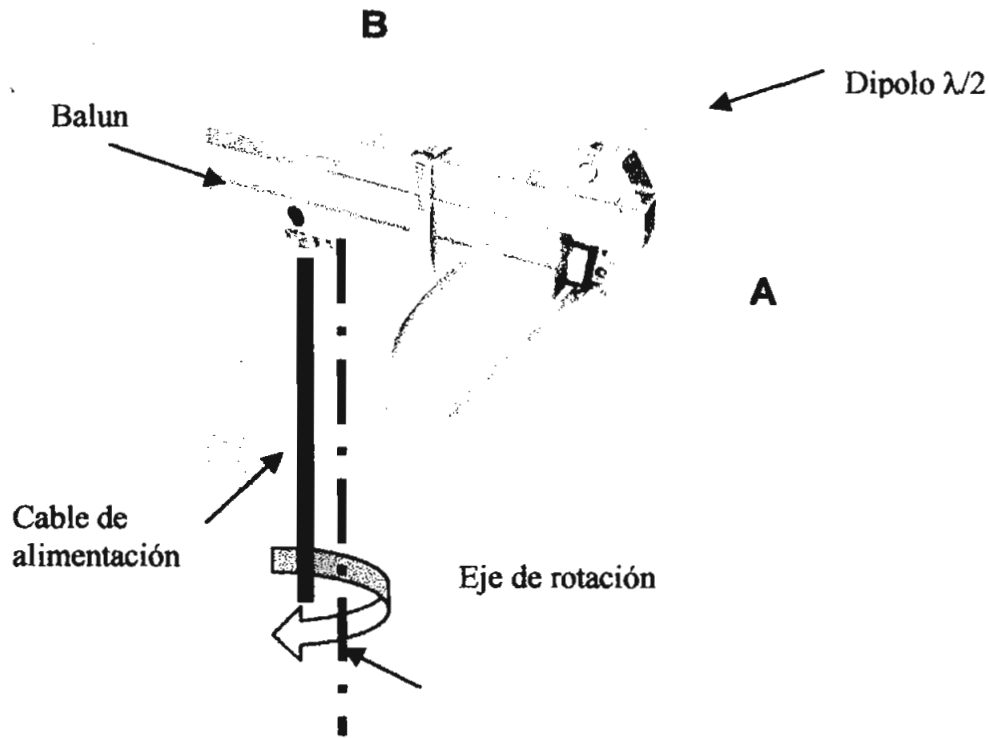


Figura 4.3. Antena dipolo utilizada para las pruebas

En la cara A la señal de recepción va a ser mayor que la cara B, debido a que el balun absorbe parte de la energía que se recibe.

- La distorsión de la señal debido a las reflexiones que existen en el ambiente donde se realizaron las pruebas.

Al comparar los gráficos en coordenadas rectangulares, se puede distinguir un mayor número de ceros en el caso experimental, esto se debe a que para valores de señal muy pequeños el sistema los redondea a cero, este inconveniente se puede reducir aumentando la sensibilidad del equipo. Este

mismo efecto no se lo observa en coordenadas polares por la forma como se grafican éstas (figura 4.4 y figura 4.5).

Además se realizaron pruebas con una antena Yagui de 6 elementos, obteniéndose como resultado las configuraciones de radiación en coordenadas polares y rectangulares mostradas en las figuras 4.6 y 4.7 respectivamente, donde se observa la directividad de la antena.

COORDENADAS RECTANGULARES

Dipolo 1/2 longitudes de onda $f=450\text{MHz}$

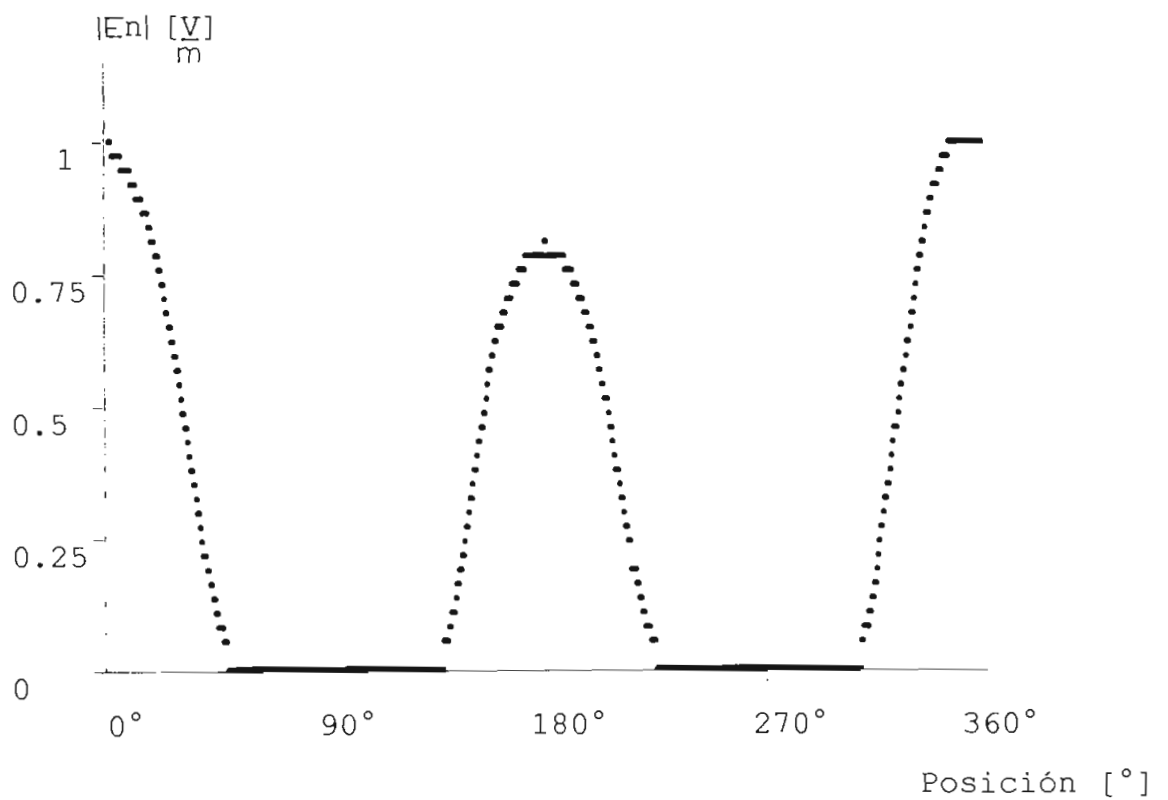


Figura 4.4. Diagrama lineal de la antena de prueba

COORDENADAS RECTANGULARES

Dipolo Largo

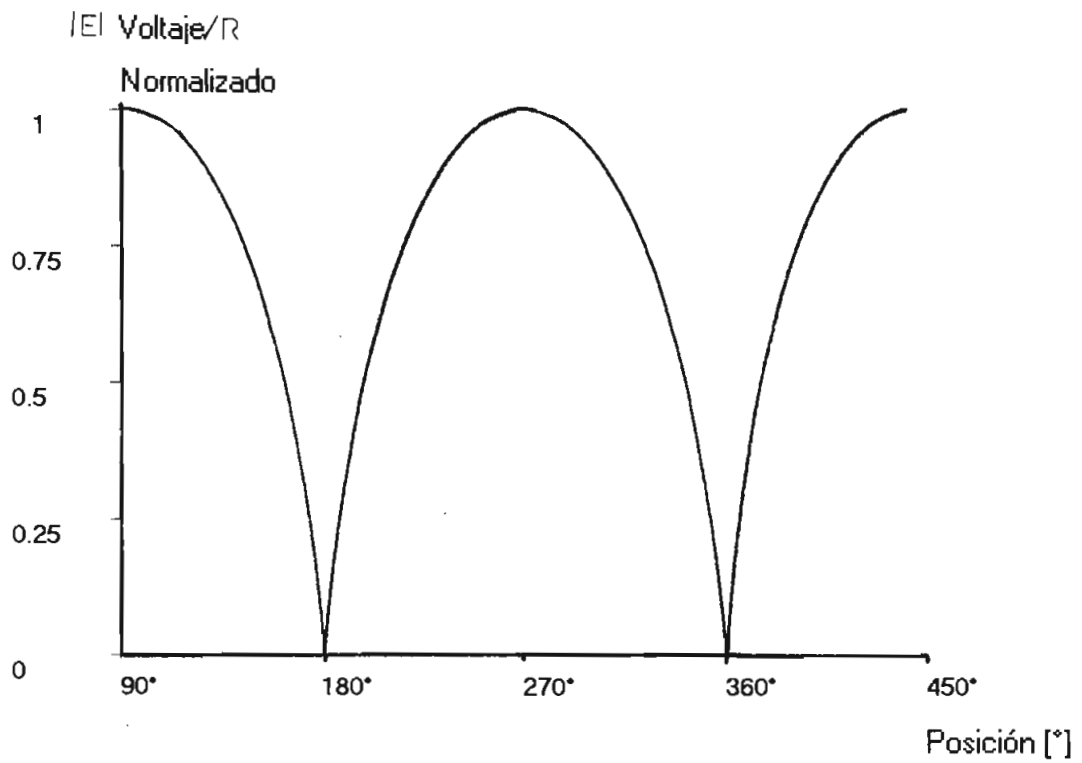


Figura 4.5. Diagrama lineal simulado

COORDENADAS POLARES

Antena Yagui de 6 elementos

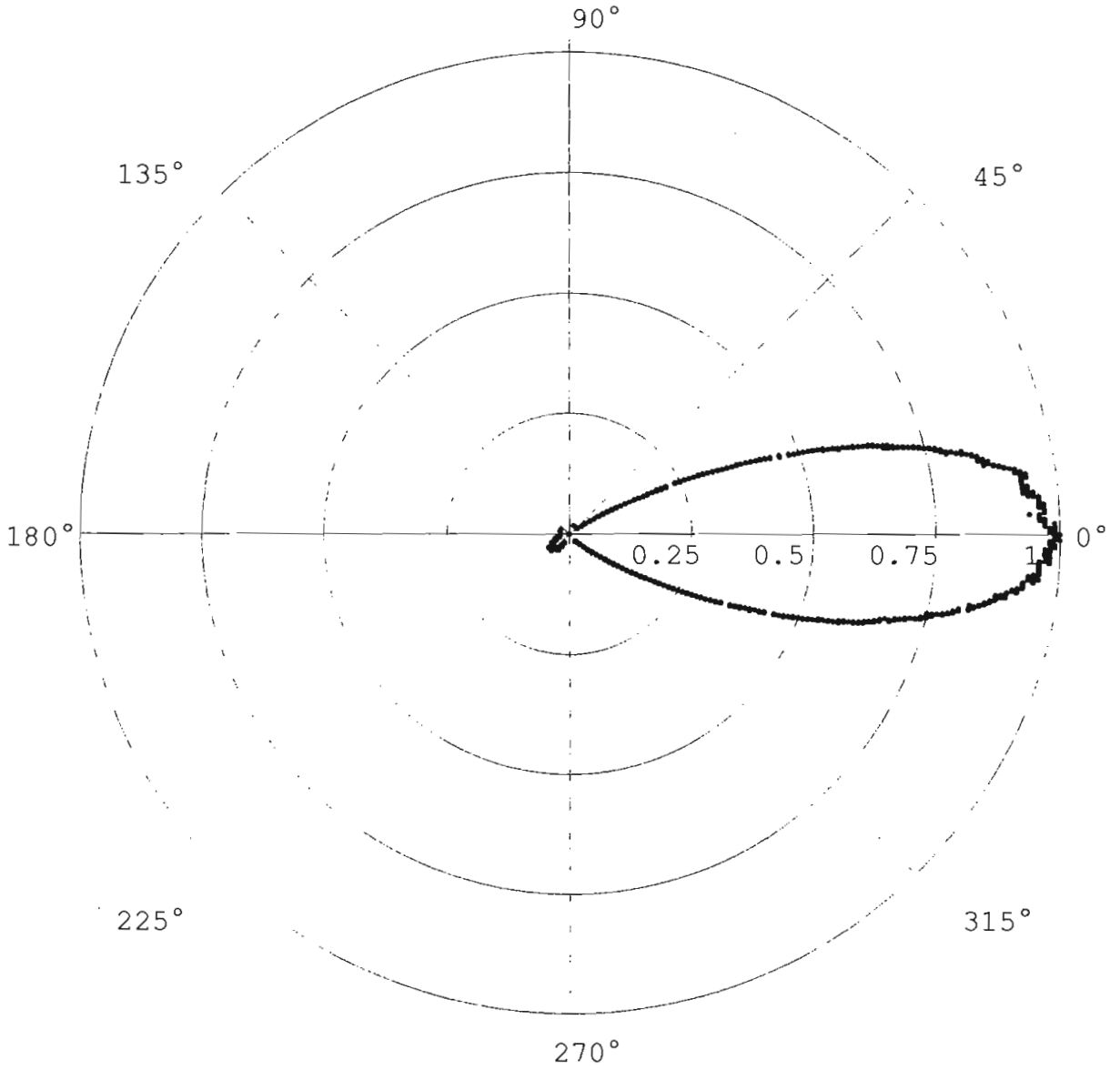


Figura 4.6. Diagrama lineal de la antena de prueba

COORDENADAS RECTANGULARES

Antena YAGUI de 6 elementos

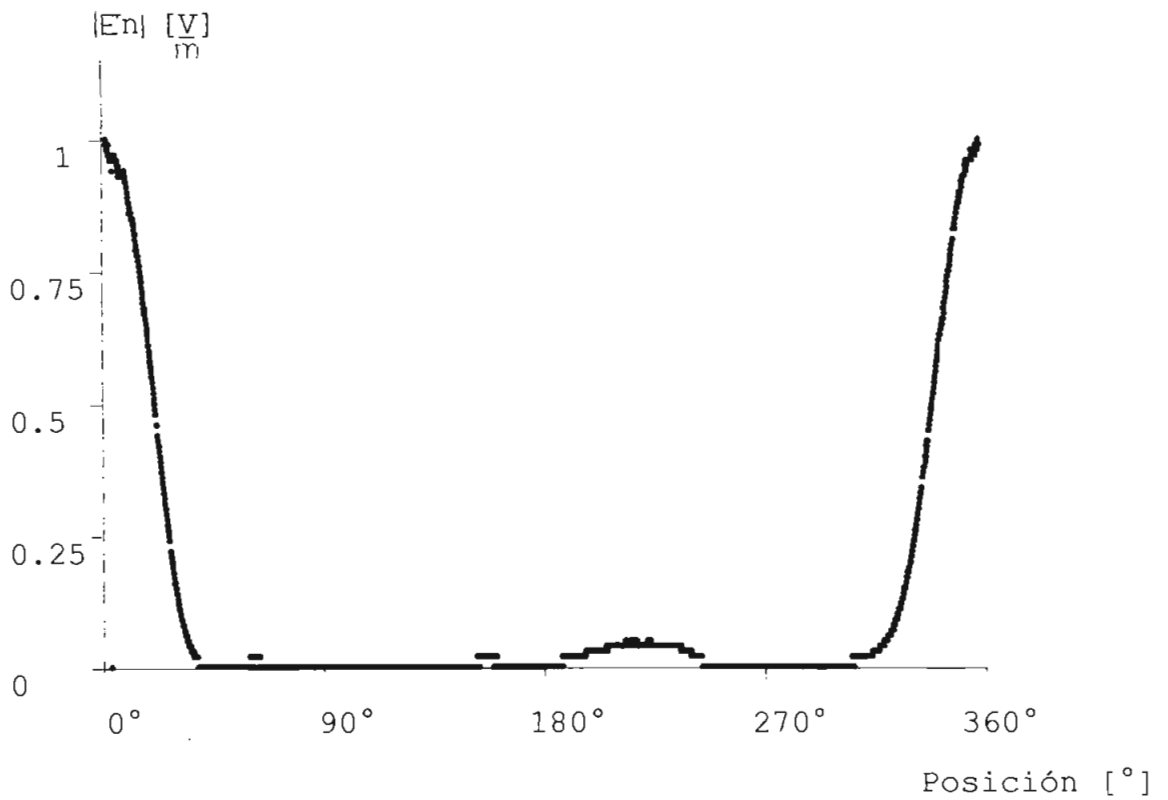


Figura 4.7. Diagrama lineal de la antena de prueba

4.2. CONCLUSIONES Y RECOMENDACIONES

CONCLUSIONES

Luego de haber culminado con el presente trabajo de tesis se puede concluir:

- Los resultados obtenidos por el sistema implementado son satisfactorios, por cuanto las configuraciones de radiación experimentales de las antenas de prueba se asemejan a las configuraciones teóricas.
- El medio en que se realizan las pruebas influye directamente en la obtención de los resultados, debido a que cualquier elemento que se encuentre en las cercanías de la antena de prueba causará reflexiones, refracciones y absorción de energía.
- Las técnicas de Microstrip facilitaron el diseño e implementación del amplificador de potencia. Es interesante apreciar que un segmento de línea de transmisión puede comportarse como un capacitor o inductor lo que mejora en gran medida el rendimiento del circuito.
- En nuestro país no se dispone en el mercado de las herramientas ni elementos necesarios para trabajar en el rango de las radiofrecuencias, por lo que este trabajo estuvo limitado a los pocos recursos disponibles.

- Los resultados se optimizaron al utilizar el módulo de desarrollo Ezkit_lite, ya que permitió adecuar y procesar digitalmente la señal obtenida del sistema radioeléctrico de recepción.
- El software y hardware del sistema de adquisición de datos, podrá ser utilizado en el mundo de las telecomunicaciones.
- El *sistema de adquisición de datos*, será un aporte al equipamiento del laboratorio de Propagación y Antenas, a la vez que servirá de apoyo y ayuda didáctica a profesores y alumnos de la Facultad de Ingeniería Eléctrica de la Escuela Politécnica Nacional.
- El lenguaje de programación Visual Basic que se utilizó, permitió realizar aplicaciones en ambiente Windows, haciendo la interfaz visual amigable al usuario.

RECOMENDACIONES

- Recomendamos que las posteriores aplicaciones e investigaciones que se realicen, relacionadas con este tema de tesis, utilicen como soporte el presente trabajo y busquen otras alternativas acordes con las nuevas tecnologías.
- La Facultad de Ingeniería Eléctrica debe actualizar a la biblioteca con información sobre nueva tecnología existente para radiofrecuencia, así como de manuales técnicos sobre RF.
- Para que estos proyectos tengan una mayor acogida entre los estudiantes la Escuela Politécnica Nacional debe implementar un rubro para investigación dirigido a este propósito.
- Debido a que los equipos del laboratorio utilizados en el sistema radioeléctrico de transmisión y recepción existentes han sobrepasado el límite de su vida útil, se recomienda realizar proyectos destinados a reemplazar los mismos, tomando en cuenta los nuevos avances tecnológicos.
- Es necesario que el equipo que donamos a nuestra facultad, sea utilizado en los trabajos teórico-prácticos que se desarrollen en el laboratorio de Propagación y Antenas.

BIBLIOGRAFIA

- ANALOG DEVICES, Ez-kit Lite Reference Manual, First Edition, Canadá 1995.
- ANALOG DEVICES, Adsp-2100 Family User's Manual, Third Edition, Massachusetts 1995.
- BALANIS Constantine, Antenna Theory. Analysis and Design, Harper & Row, New York 1982.
- BERRAL Isidoro, Instalación de Antenas de TV, editorial Paraninfo, España 1996.
- CEVALLOS Mario, Diseño de Amplificadores para UHF y Microondas utilizando parámetros S, E.P.N., Quito – Ecuador 1984
- COMMUNICATIONS TRANSISTOR CORPORATION, Solid Circuits, California 1973.
- DWORSKY Lawrence, Modern Transmission Line Theory and Applications, New York 1979.
- EDITEC, Práctica de la Construcción e Instalación de Antenas de FM y TV, España 1970.
- GENERAL RADIO, Catalog 78, Massachusetts - USA 1978
- GIBILISCO Stan, Electrónica. Diccionario Enciclopédico, Tomo III, McGRAW-HILL, México 1994.
- GUREWICH Ori, Aprendiendo Visual Basic 5.0. en 21 días, Prentice Hall, México 1997.
- GUILBERT CH, La práctica de antenas, 2da edición, MARCOMBO, España 1973.

- ITT, Reference Data for Radio Engineers, fifth edition, United States of America 1973.
- JASIK H, JOHNSON R, Antenna Engineering Handbook, Second Edition, McGRAW-HILL, New York 1961.
- JORDAN / BALMAIN, Electromagnetic Waves and Radiating Systems, Prentice-Hall, United States of America 1968.
- KRAUS John, Antennas, Edition McGRAW-HILL, New York 1950
- NICOLALDE Edison, Diseño y Construcción de un Sistema de Adquisición de datos para obtener el Diagrama de Radiación de una Antena, E.P.N., Tesis de Grado Quito 1992.
- OPPENHEIM Alan, Señales y Sistemas, Prentice Hall, México 1983
- SCHELKUNOFF Sergei, Antennas theory and practice, John Wiley & Sons, New York 1952.
- TOCCI Ronald, Sistemas Digitales, 5ta edición, Prentice-Hall, México 1993.

ANEXOS

ANEXO A

CARACTERISTICAS ELECTRICAS

2N5944
2N5945
2N5946

The RF Line

NPN SILICON RF POWER TRANSISTORS

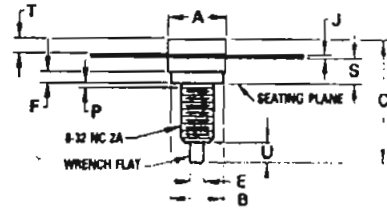
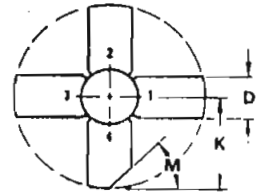
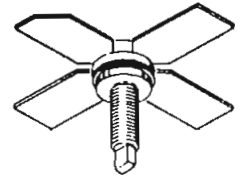
... designed for 7.0 to 15 Volts, UHF large signal amplifier applications required in industrial and commercial FM equipment operating in the 400 to 960 MHz range.

- Specified 12.5 Voff, 470 MHz Characteristics —
 Power Output = 2.0 W — 2N5944
 4.0 W — 2N5945
 10 W — 2N5946

 Minimum Gain = 9.0 dB — 2N5944
 8.0 dB — 2N5945
 6.0 dB — 2N5946

 Efficiency = 60% Minimum
- Characterized with series equivalent large-signal impedance parameters

2.0, 4.0, 10 W - 470 MHz
 RF POWER TRANSISTORS
 NPN SILICON



DIM	MILLIMETERS		INCHES	
	MIN	MAX	MIN	MAX
A	7.06	7.26	0.278	0.286
B	6.20	6.50	0.244	0.256
C	14.99	16.51	0.590	0.650
D	5.46	5.96	0.215	0.235
E	1.40	1.65	0.055	0.065
F	1.52	—	0.060	—
J	0.08	0.17	0.003	0.007
K	11.05	—	0.435	—
M	45° NOM	—	45° NOM	—
P	—	1.27	—	0.050
S	3.00	3.25	0.118	0.128
T	1.40	1.77	0.055	0.070
U	2.92	3.68	0.115	0.145

MAXIMUM RATINGS

Rating	Symbol	2N5944	2N5945	2N5946	Unit
*Collector-Emitter Voltage	V _{CEO}	16			Vdc
*Collector-Base Voltage	V _{CBO}	36			Vdc
*Emitter-Base Voltage	V _{EBO}	4.0			Vdc
*Collector Current - Continuous	I _C	0.4	0.8	2.0	Adc
*Total Device Dissipation @ T _C = 25°C(1)	P _D	5.0	15	37.5	Watts
Derate above 25°C		28.5	85.5	214	mW/°C
*Storage Temperature Range	T _{stg}	-65 to +200			°C
Stud Torque(2)		5.5			in-lbs.

*Indicates JEDEC Registered Data
 (1) These devices are designed for RF operation. The total device dissipation rating applies only when the devices are operated as RF amplifiers.
 (2) For repeated assembly use 5 in lbs.

STYLE 1:
 PIN 1. EMITTER
 2. BASE
 3. EMITTER
 4. COLLECTOR

CASE 244-04

2N5944, 2N5945, 2N5946

*ELECTRICAL CHARACTERISTICS ($T_C = 25^\circ\text{C}$ unless otherwise noted)

Characteristic	Symbol	Min	Typ	Max	Unit
OFF CHARACTERISTICS					
Collector-Emitter Breakdown Voltage ($I_C = 50 \text{ mA dc}, I_B = 0$)	$V_{(BR)CEO}$	16	—	—	Vdc
($I_C = 100 \text{ mA dc}, I_B = 0$)	2N5944	16	—	—	
($I_C = 200 \text{ mA dc}, I_B = 0$)	2N5945	16	—	—	
Collector-Emitter Breakdown Voltage ($I_C = 50 \text{ mA dc}, V_{BE} = 0$)	$V_{(BR)CES}$	36	—	—	Vdc
($I_C = 100 \text{ mA dc}, V_{BE} = 0$)	2N5944	36	—	—	
($I_C = 200 \text{ mA dc}, V_{BE} = 0$)	2N5945	36	—	—	
Emitter-Base Breakdown Voltage ($I_E = 1.0 \text{ mA dc}, I_C = 0$)	$V_{(BR)EBO}$	4.0	—	—	Vdc
($I_E = 2.0 \text{ mA dc}, I_C = 0$)	2N5944	4.0	—	—	
($I_E = 4.0 \text{ mA dc}, I_C = 0$)	2N5945	4.0	—	—	
Collector Cutoff Current ($V_{CE} = 15 \text{ Vdc}, V_{BE} = 0, T_C = 55^\circ\text{C}$)	I_{CES}	—	0.2	10	mA dc
	2N5944	—	0.5	20	
	2N5945, 2N5946	—	—	—	
Collector Cutoff Current ($V_{CB} = 15 \text{ Vdc}, I_E = 0$)	I_{CBO}	—	—	1.0	mA dc
	2N5944, 2N5945	—	—	2.0	
	2N5946	—	—	—	

ON CHARACTERISTICS

DC Current Gain ($I_C = 100 \text{ mA dc}, V_{CE} = 5.0 \text{ Vdc}$)	h_{FE}	20	80	—	
($I_C = 200 \text{ mA dc}, V_{CE} = 5.0 \text{ Vdc}$)	2N5944	20	80	—	
($I_C = 500 \text{ mA dc}, V_{CE} = 5.0 \text{ Vdc}$)	2N5945	20	80	—	
	2N5946	20	80	—	

DYNAMIC CHARACTERISTICS

Output Capacitance ($V_{CB} = 12.5 \text{ Vdc}, I_E = 0, f = 1.0 \text{ MHz}$)	C_{ob}	—	11	15	pF
	2N5944	—	18	25	
	2N5945	—	38	45	
	2N5946	—	—	—	

FUNCTIONAL TEST (Figures 20 and 21)

Common-Emitter Amplifier Power Gain ($V_{CC} = 12.5 \text{ Vdc}, P_{out} = 2.0 \text{ W}, I_C(\text{max}) = 267 \text{ mA dc}, f = 470 \text{ MHz}$)	G_{PE}	9.0	10	—	dB
($V_{CC} = 12.5 \text{ Vdc}, P_{out} = 4.0 \text{ W}, I_C(\text{max}) = 533 \text{ mA dc}, f = 470 \text{ MHz}$)	2N5944	8.0	9.0	—	
($V_{CC} = 12.5 \text{ Vdc}, P_{out} = 10 \text{ W}, I_C(\text{max}) = 1.33 \text{ A dc}, f = 470 \text{ MHz}$)	2N5945	6.0	7.0	—	
Collector Efficiency ($V_{CC} = 12.5 \text{ Vdc}, P_{out} = 2.0 \text{ W}, I_C(\text{max}) = 240 \text{ mA dc}, f = 470 \text{ MHz}$)	η	60	—	—	%
($V_{CC} = 12.5 \text{ Vdc}, P_{out} = 4.0 \text{ W}, I_C(\text{max}) = 500 \text{ mA dc}, f = 470 \text{ MHz}$)	2N5944	60	—	—	
($V_{CC} = 12.5 \text{ Vdc}, P_{out} = 10 \text{ W}, I_C(\text{max}) = 1.3 \text{ A dc}, f = 470 \text{ MHz}$)	2N5945	60	—	—	
	2N5946	60	—	—	

*Indicates JEDEC Registered Data

These devices are available in various packages, such as a studless stripline package, TO-206AD (TO-39) and also in chip form on beryllium oxide carriers for hybrid assemblies.
For further information, contact your nearest Motorola representative or the factory representative.

2N5944, 2N5945, 2N5946

2N5944 TYPICAL PERFORMANCE DATA

FIGURE 1 - SERIES EQUIVALENT IMPEDANCE

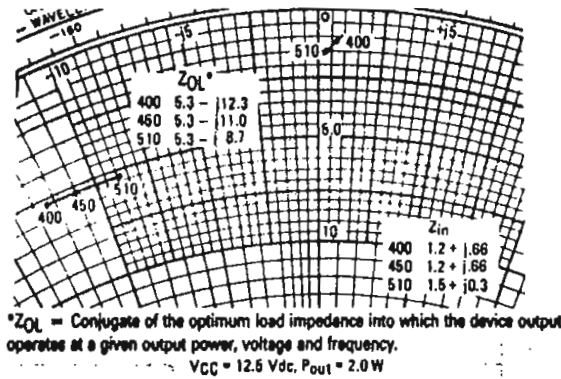


FIGURE 2 - OUTPUT POWER versus SUPPLY VOLTAGE

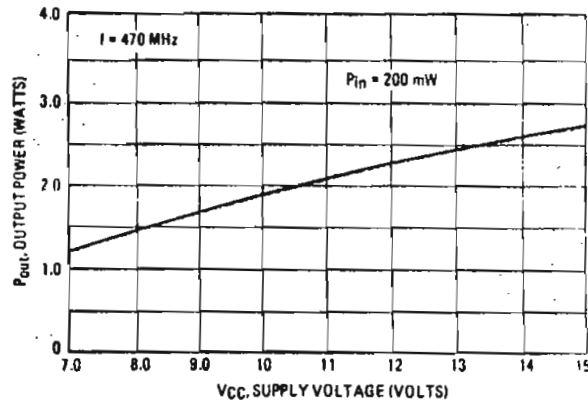


FIGURE 3 - OUTPUT POWER versus INPUT POWER

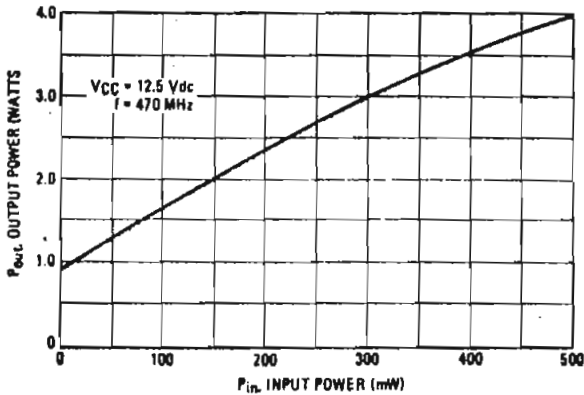


FIGURE 4 - OUTPUT POWER versus FREQUENCY

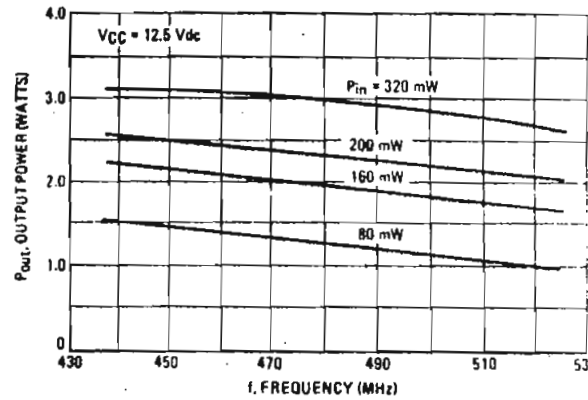


FIGURE 5 - OUTPUT POWER versus INPUT POWER

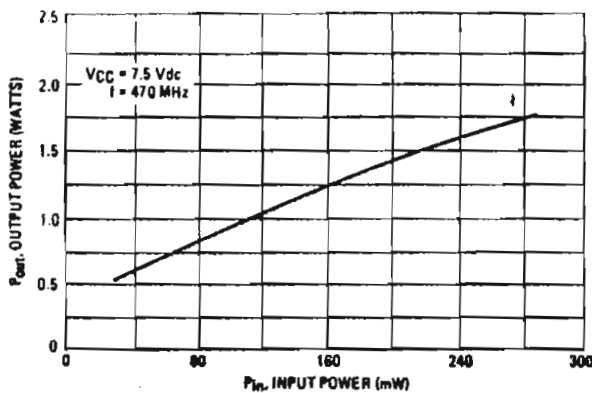
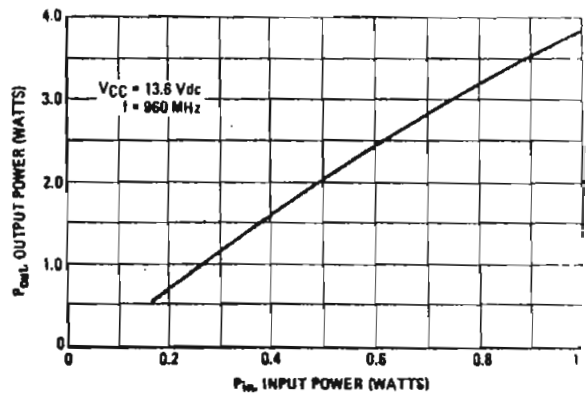
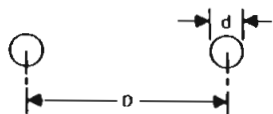


FIGURE 6 - OUTPUT POWER versus INPUT POWER



Open 2-wire line in air (See also Fig. 23).

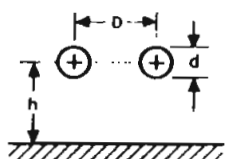


$$Z_0 = 120 \cosh^{-1}(D/d)$$

$$\approx 276 \log_{10}(2D/d)$$

$$\approx 120 \log_e(2D/d)$$

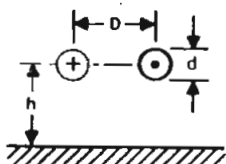
Wires in parallel, near ground



For $d \ll D, h$,

$$Z_0 = (69/\epsilon^{1/2}) \log_{10}\{(4h/d)[1+(2h/D)^2]^{1/2}\}$$

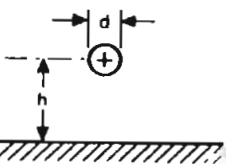
Balanced, near ground



For $d \ll D, h$,

$$Z_0 = (276/\epsilon^{1/2}) \log_{10}\{(2D/d)[1+(D/2h)^2]^{-1/2}\}$$

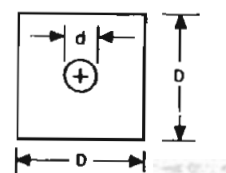
Single wire, near ground



For $d \ll h$,

$$Z_0 = (138/\epsilon^{1/2}) \log_{10}(4h/d)$$

Single wire, square enclosure



$$Z_0 \approx [138 \log_{10} \rho + 6.48 - 2.34A - 0.48B - 0.12C] \epsilon^{-1/2}$$

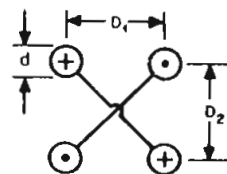
where $\rho = D/d$

$$A = (1 + 0.405\rho^{-4}) / (1 - 0.405\rho^{-4})$$

$$B = (1 + 0.163\rho^{-8}) / (1 - 0.163\rho^{-8})$$

$$C = (1 + 0.067\rho^{-12}) / (1 - 0.067\rho^{-12})$$

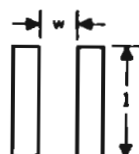
I. Balanced 4-wire



For $d \ll D_1, D_2$,

$$Z_0 = (138/\epsilon^{1/2}) \log_{10}\{(2D_2/d)[1+(D_2/D_1)^2]^{-1/2}\}$$

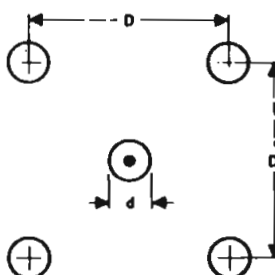
J. Parallel-strip line



$w/l < 0.1$

$$Z_0 \approx 377(w/l)$$

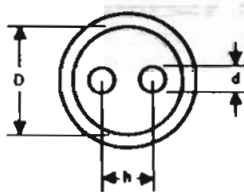
K. Five-wire line



For $d \ll D$,

$$Z_0 = (173/\epsilon^{1/2}) \log_{10}(D/0.933d)$$

L. Wires in parallel—sheath return



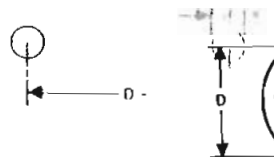
For $d \ll D, h$,

$$Z_0 = (69/\epsilon^{1/2}) \log_{10}[(\nu/2\sigma^2)(1-\sigma^4)]$$

$$\sigma = h/D$$

$$\nu = h/d$$

D. Coaxial 2-wire line in air

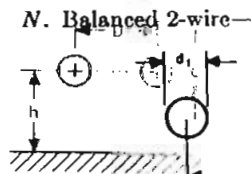


$$Z_0 = 120 \cosh^{-1}(D/d)$$

$$\approx 276 \log_{10}(2D/d)$$

$$\approx 120 \log_e(2D/d)$$

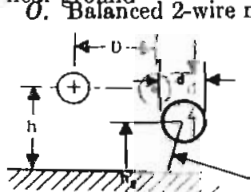
E. Wires in parallel, near ground



For $d \ll D, h$,

$$Z_0 = (60/\epsilon^{1/2}) \log_{10}\{(4h/d)[1+(2h/D)^2]^{1/2}\}$$

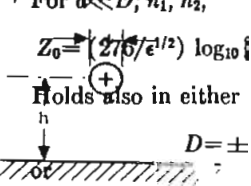
F. Balanced, near ground



For $d \ll D, h$,

$$Z_0 = (276/\epsilon^{1/2}) \log_{10}\{(2D/d)[1+(D/2h)^2]^{-1/2}\}$$

G. Single wire, near ground



For $d \ll D, h_1, h_2$,

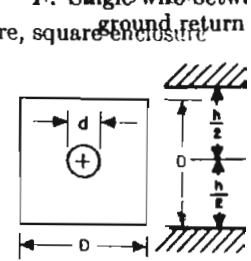
$$Z_0 = (276/\epsilon^{1/2}) \log_{10}\{(2D/d)[1+(D/2h)^2]^{-1/2}\}$$

Holds also in either

$$D = \pm$$

$$h_1 = h_2$$

H. Single wire, square enclosure



For $d/h < 0.75$,

$$Z_0 \approx [138 \log_{10} \rho + 6.48 - 2.34A - 0.48B - 0.12C] \epsilon^{-1/2}$$

Principles of Microstrip Design

By Alam Tam
American Microwave Technology

Microstrip is becoming a more important and popular transmission media for use in the RF and microwave industry, based upon its electrical, cost, size, manufactureability and reproducibility performance. All RF, microwave and high speed digital circuit designers should become familiar with microstrip design techniques to better their job skills. This microstrip design tutorial will enhance your understanding of the advantages, electrical characteristics and physical properties through practical applications, specific equations, useful tables and design examples.

Microstrip is a transmission line which carries the RF signal from one point to another with a unique characteristic impedance. The characteristic impedance is determined by the selected width, plus the board or substrate material used. The physical cross-sectional construction of a microstrip above the dielectric medium is shown in Figure 1. The high frequency energy will produce a skin effect on the microstrip conductor. The current density around the conductor is then concentrated on a sheet where a skin thickness deep on the surface will be exposed to the electric field.

The transverse electric (TE) mode defines that a transmission have magnetic field components in the direction of energy flow yet the electric field is everywhere transverse. On the other hand, transverse magnetic (TM) mode defines that the transmission have the electric field components in the direction of energy flow yet the magnetic field is everywhere transverse. They are referred to as "higher order" modes for microwave transmission.

As a practical recommendation, the physical separation between the metallic cover and the actual microstrips on the substrate should be at least 10 times the substrate thickness. To avoid the higher waveguide mode excitation within the enclosure, the physical width of the housing in which the microstrip are situated, should not be more than one half of the free space wavelength at the operating frequency.

Microstrip design has proven to be a very mature technology. It provides wide operating bandwidth, good miniaturization, tremendous weight reduction, satisfactory thermal characteristics, reasonable RF power handling, ease of component integration and mass production. It is being widely used in many military and spaceborne programs. Historically, the reliability performance of microstrip designs has been successful, as predicted.

Review of Fundamental Transmission Line Equations

Since the electric and magnetic field lines and the ground plane are not entirely contained in the substrate as shown in Figure 2, the propagating mode along the microstrip is not purely transverse electromagnetic (TEM) but quasi-TEM. Assuming it is quasi-TEM, the phase velocity in the microstrip is given by:

$$V_p = C/\sqrt{\epsilon_{eff}} \quad (1)$$

The wavelength in the microstrip line is given by:

$$\lambda_g = V_p/f \quad (2a)$$

$$\text{Where, } \lambda_g = \lambda_0/\sqrt{\epsilon_{eff}} = C/f/\sqrt{\epsilon_{eff}} \quad (2b)$$

λ_g = microstrip wavelength

λ_0 = electrical wavelength in the free air

The characteristic impedance of the transmission line is also given by:

$$Z_0 = 1 / V_p(C), \text{ where } C = 3 \times 10^8 \text{ m/s} \quad (3)$$

Determination of Characteristic Impedance (Z_0)

The Wheeler and Schneider formulas shall be considered because they provide accuracy better than 2 percent.

For $w/h < 1$,

$$Z_0 = \frac{60}{\sqrt{\epsilon_{eff}}} \ln(8h/W_e + 2.5W_e/h), \text{ where:} \quad (4a)$$

$$\epsilon_{eff} = \frac{\epsilon_r + 1}{2} + \frac{\epsilon_r - 1}{2} \left[\frac{1}{1 + 12h/W_e} \right]^{-0.5} + 0.04(1 - W_e/h)^2 \quad (4b)$$

For $w/h > 1$,

$$Z_0 = \frac{377}{\sqrt{\epsilon_{eff}} [W_e/h + 1.393 + 0.667 \ln(W_e/h + 1.444)]} \quad (5a)$$

$$W_e = W + \frac{t}{\pi} \left(\ln \frac{2h_e}{t} + 1 \right) \text{ Effective microstrip width} \quad (5b)$$

$$h_e = h - (2t) \text{ Effective thickness of the substrate} \quad (5c)$$

$$\epsilon_{eff} = \frac{\epsilon_r + 1}{2} + \frac{\epsilon_r - 1}{2} \left[\frac{1}{1 + 12h/W_e} \right]^{-0.5} \text{ Effective dielectric constant} \quad (5d)$$

Example #1: What is the characteristic impedance of an 80 mil wide microstripline on a 1/32" thick Teflon fiberglass board with a dielectric constant of 2.55?

$$\epsilon_{eff} = 1.78 + 0.775 (0.425) = 2.1$$

$$W_e = 80 + 2 = 82 \text{ mil}$$

$$h_e = 31 - 2(1.4) = 28 \text{ mil}$$

@ $t = 1.4$ mil thickness for 1 oz. copper clad material

$W_e/h_e = 82/31 = 2.65$, which is greater than 1 in this case:

$$\text{Thus, } Z_0 = \frac{377}{\sqrt{2.1} [82/31 + 1.393 + 0.667 \ln(82/31 + 1.444)]} = \frac{377}{7.22} = 52 \text{ Ohms}$$

An alternate solution can be achieved by using the graphical design approach, referring to Figure 3. Draw a vertical line starting at $w/h = 2.65$ axis until it intercepts with the $\epsilon = 2.55$ curve. At the same point, make a line in parallel with the w/h axis until it reaches the vertical Z_0 axis. Therefore, the new characteristic impedance can be easily found and the result should be fairly close to that calculated in Example #1.

Electrical length of an inductive microstrip is determined by:

$$l = \frac{L \times V_p}{Z_0} \quad (6)$$

ANEXO B

MANUAL DE USUARIO

MANUAL DE USUARIO

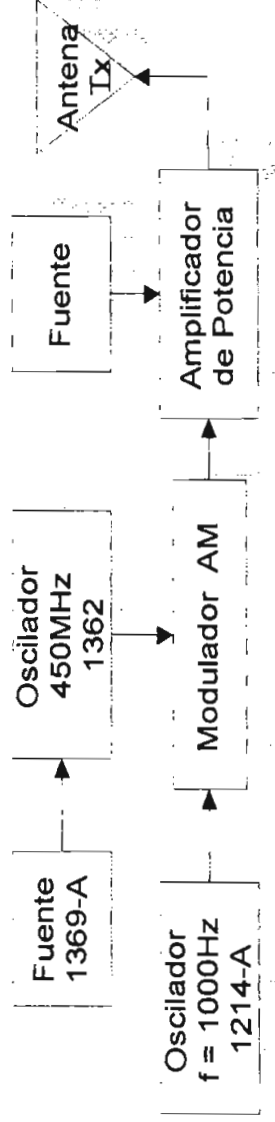
Este manual le permite conocer los equipos necesarios para la implementación de todo este sistema y la forma de conectarlos.

- Equipos a utilizarse
- Sistema Radioeléctrico de Tx
- Sistema Radioeléctrico de Rx
- Sistema de Adquisición de datos
- Recomendaciones

Equipos a utilizarse

- Oscilador de 1 KHz, de General Radio modelo 1214A.
- 2 Osciladores de RF General Radio modelo 1362.
- Fuente de alimentación de General Radio 1369-A.
- Amplificador de potencia y fuente de alimentación.
- Antena de transmisión.
- Antena de recepción.
- Mezclador de General Radio modelo 874 MRAL.
- Amplificador de IF.
- Sistema de adquisición de datos.
- Osciloscopio.
- Computador.

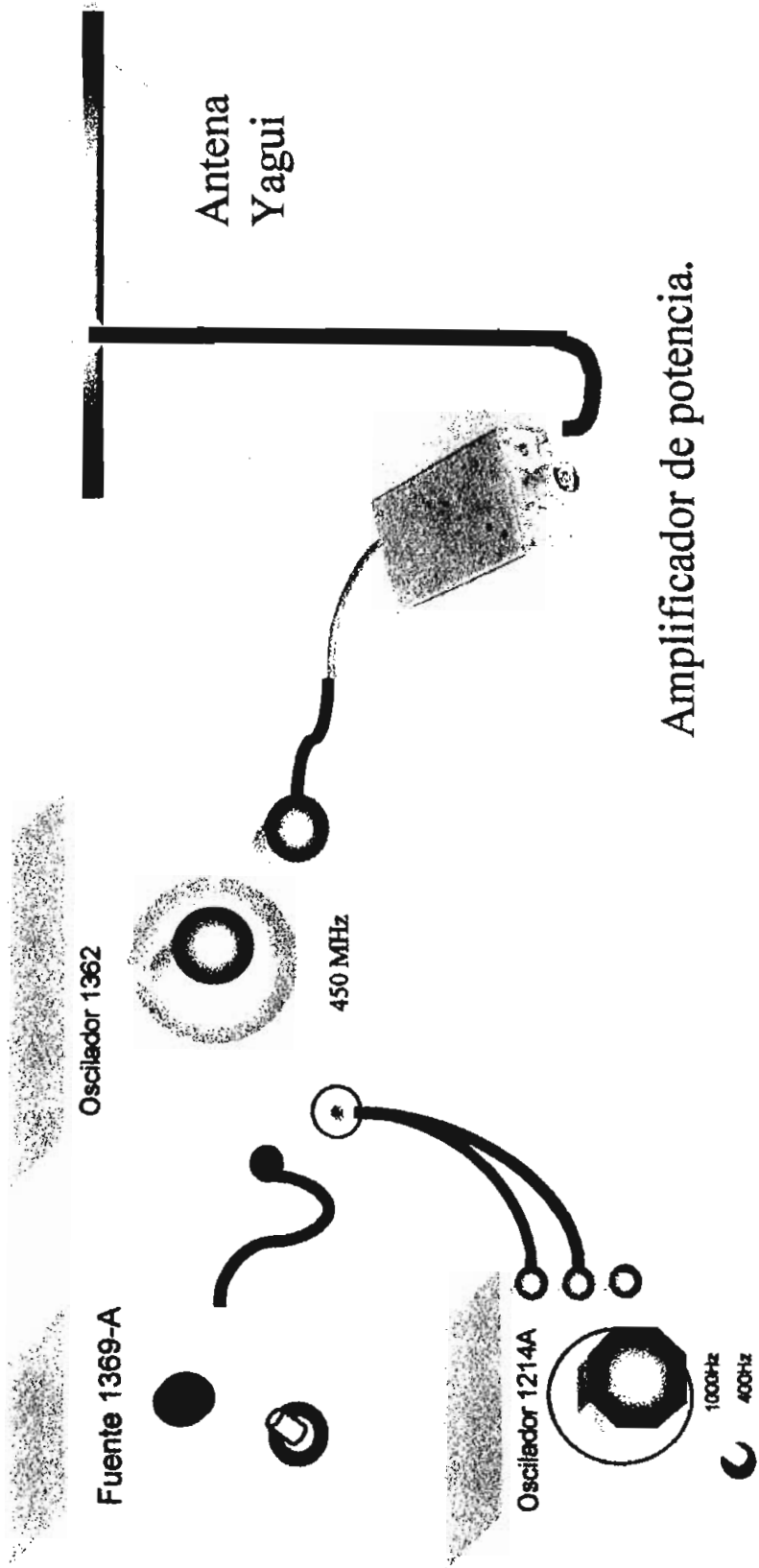
Sistema Radioeléctrico de Tx



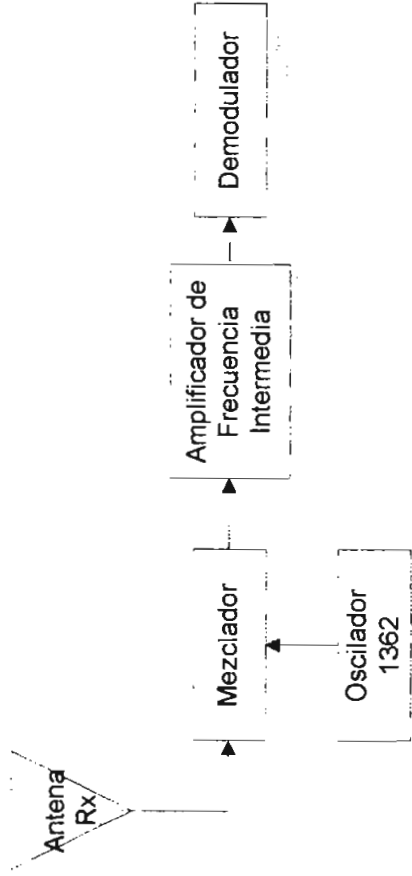
El sistema radioeléctrico de transmisión se lo puede representar con este diagrama de bloques. Los equipos utilizados son:

- Oscilador de 1KHz de General Radio 1214-A
- Oscilador y modulador de RF de General Radio modelo 1362.
- Fuente de alimentación de General Radio 1369A
- Amplificador de potencia y fuente de alimentación.
- Antena de transmisión.

Cont... Sistema Radioeléctrico de Tx



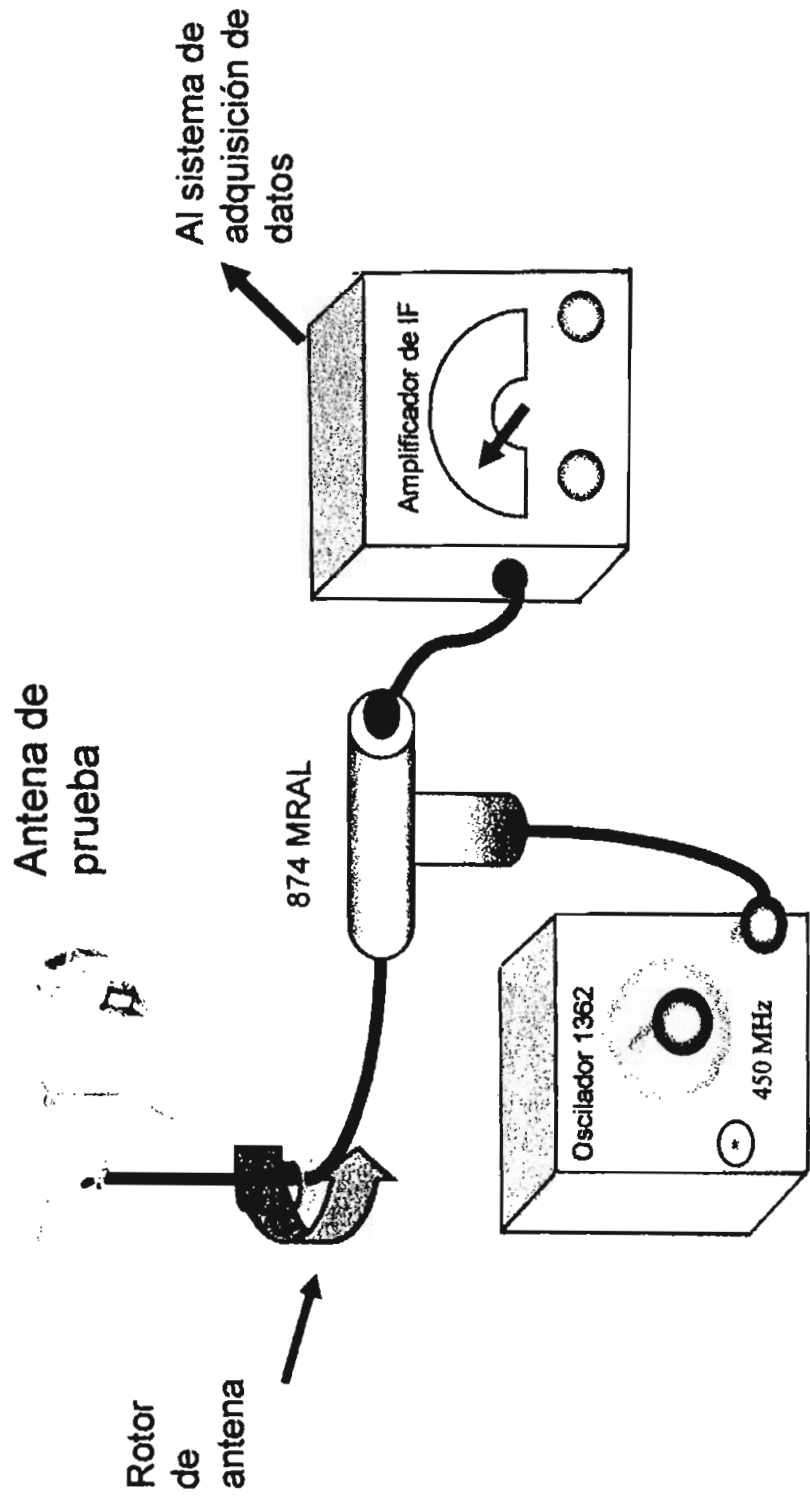
Sistema Radioeléctrico de Rx



El diagrama de conexión para el sistema radioeléctrico de Rx se muestra en la figura. Los equipos utilizados son:

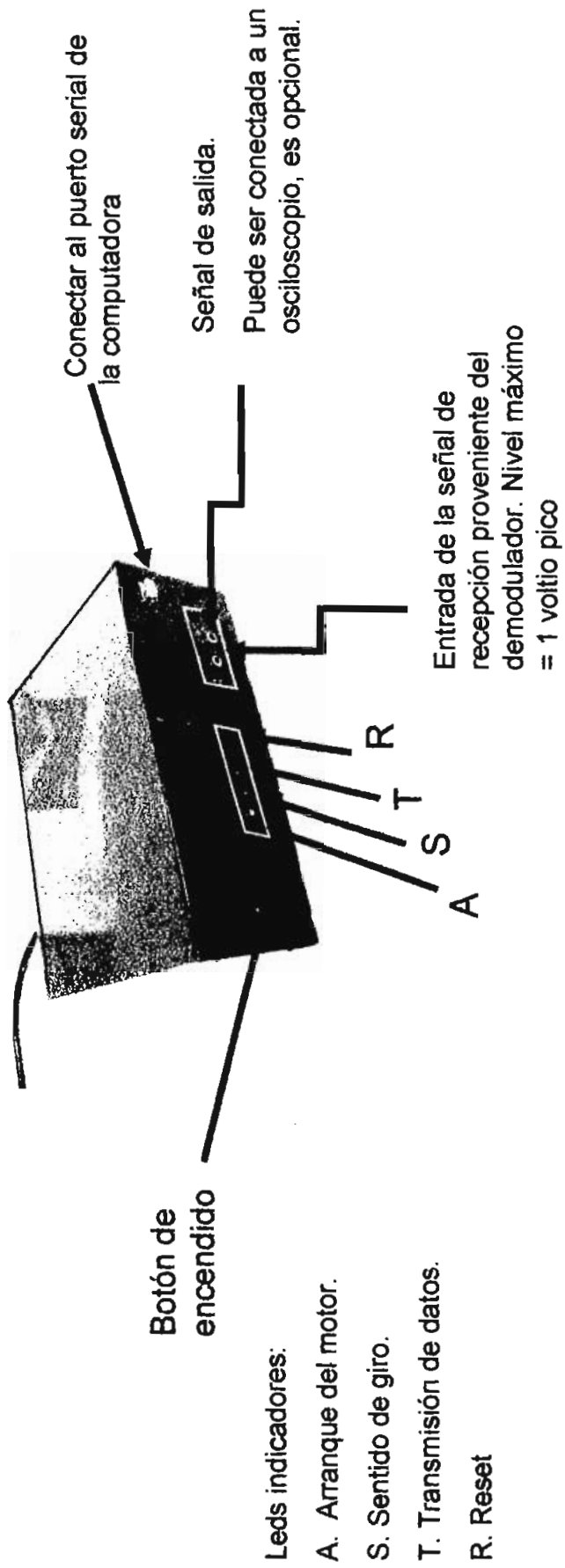
- Antena de Recepción.
- Mezclador de General Radio modelo 874 MRAL.
- Amplificador de frecuencia intermedia y demodulador.

Cont... Sistema Radioeléctrico de Rx

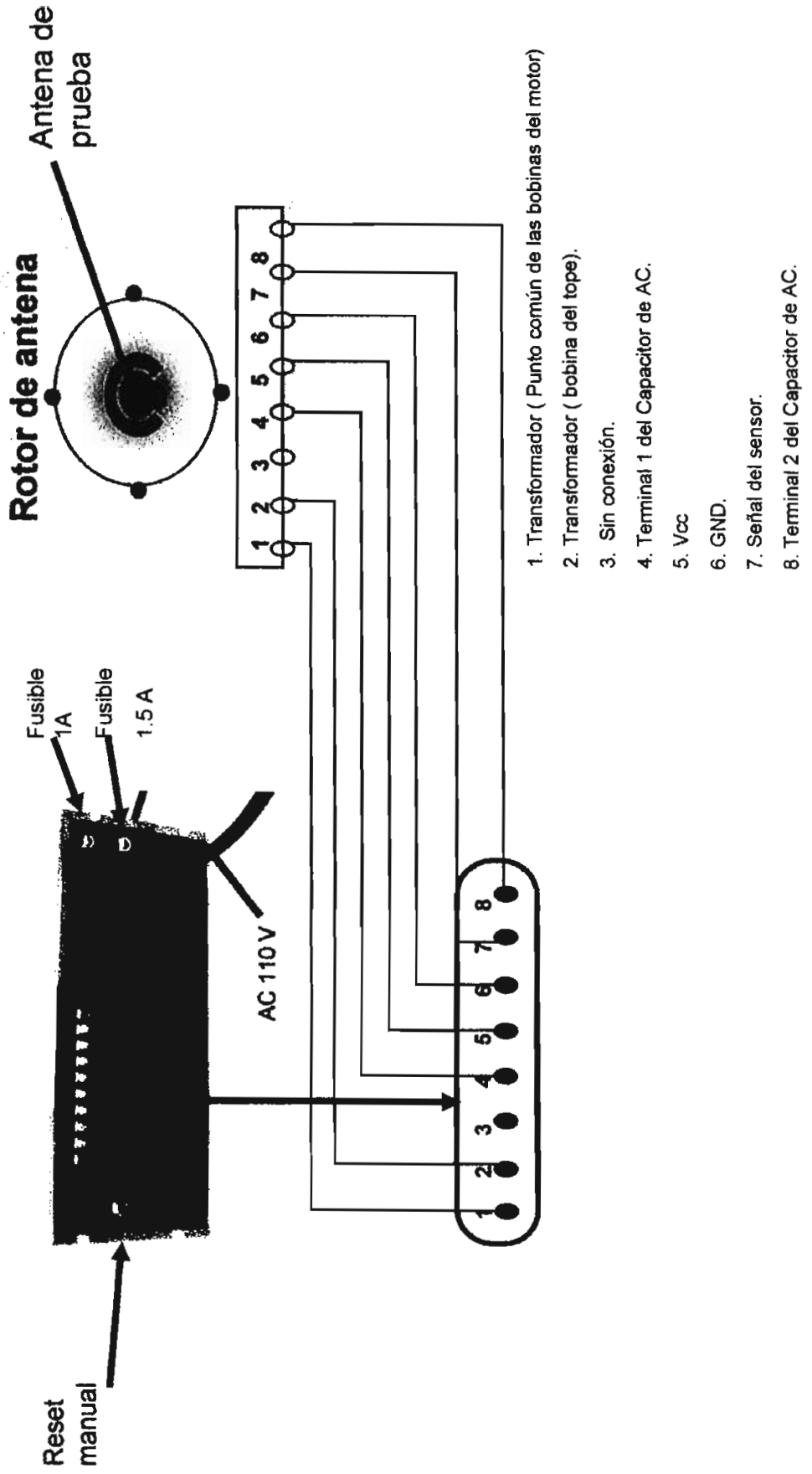


Sistema de adquisición de datos

El equipo de adquisición de datos debe ser conectado de la siguiente forma:

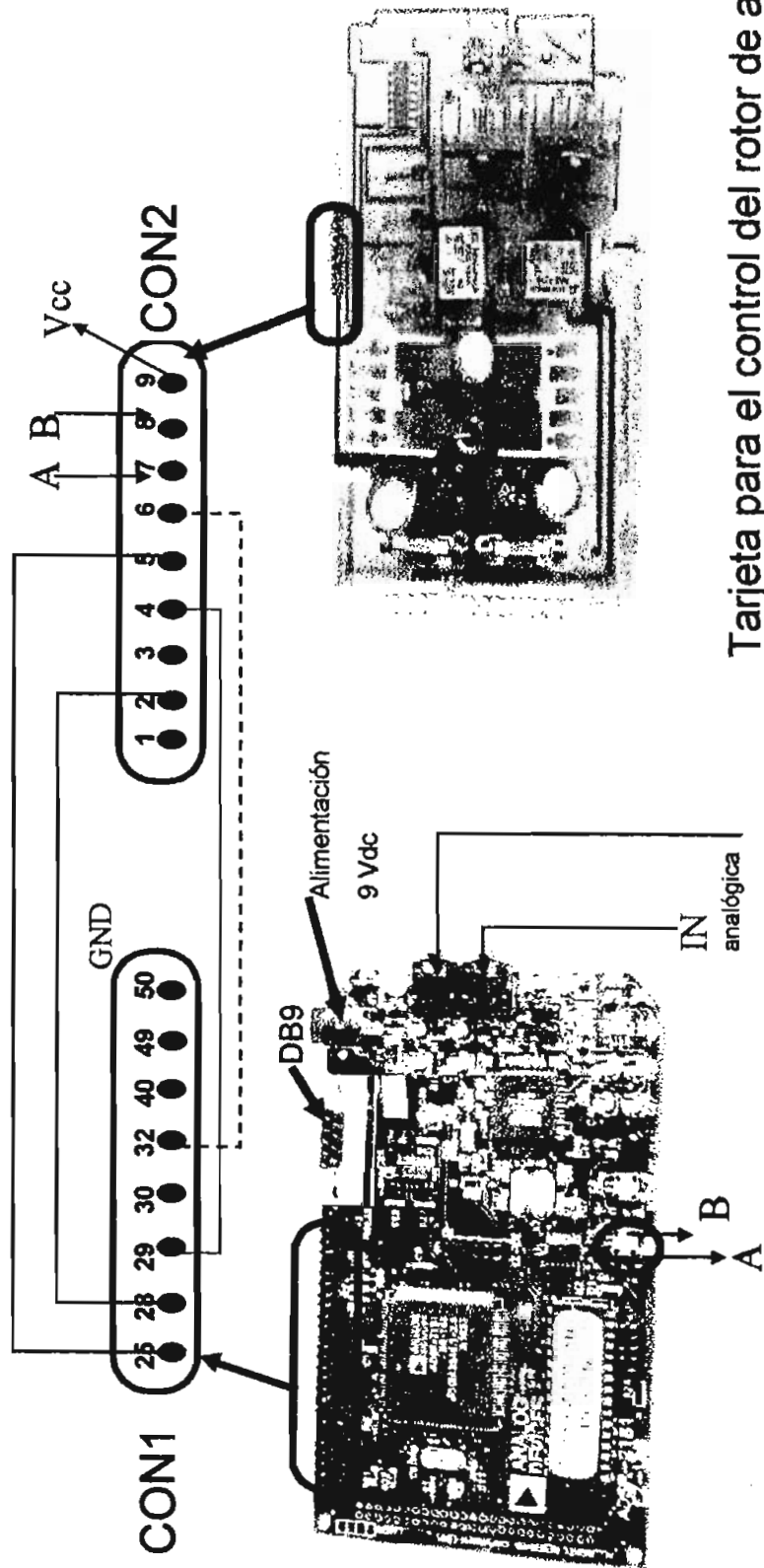


Cont... Sistema de adquisición de datos

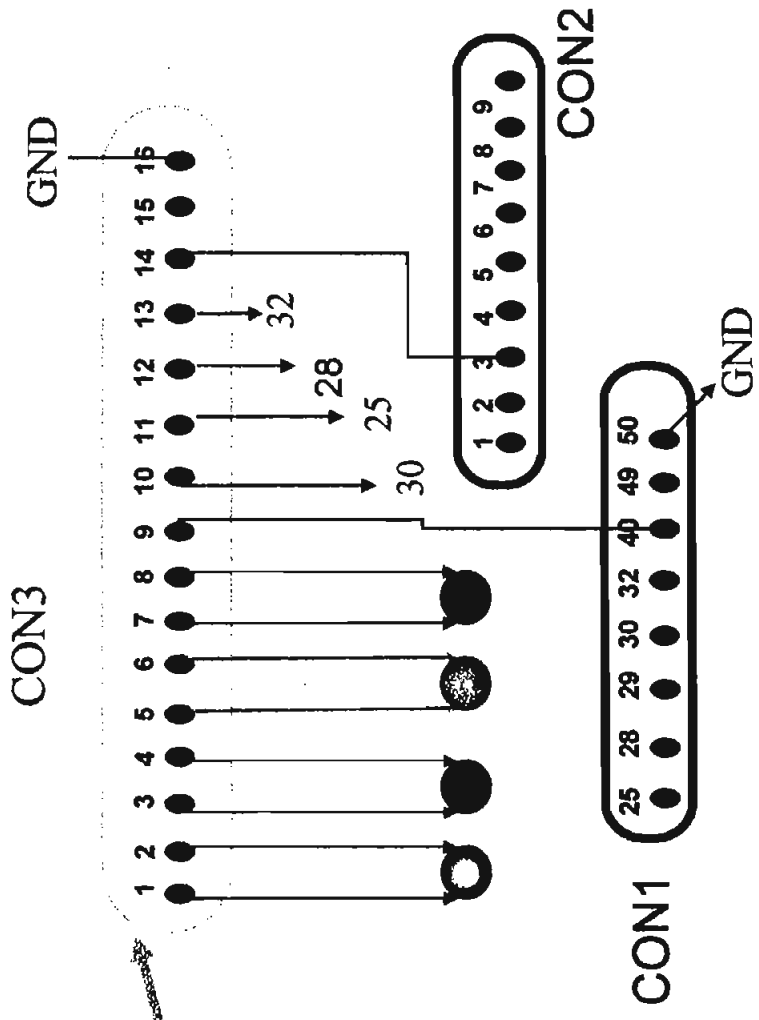
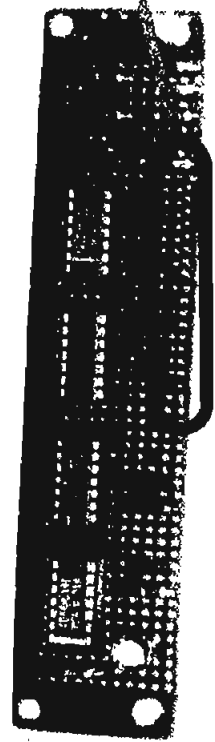


Cont... Sistema de adquisición de datos

Conexiones internas del sistema de adquisición de datos.



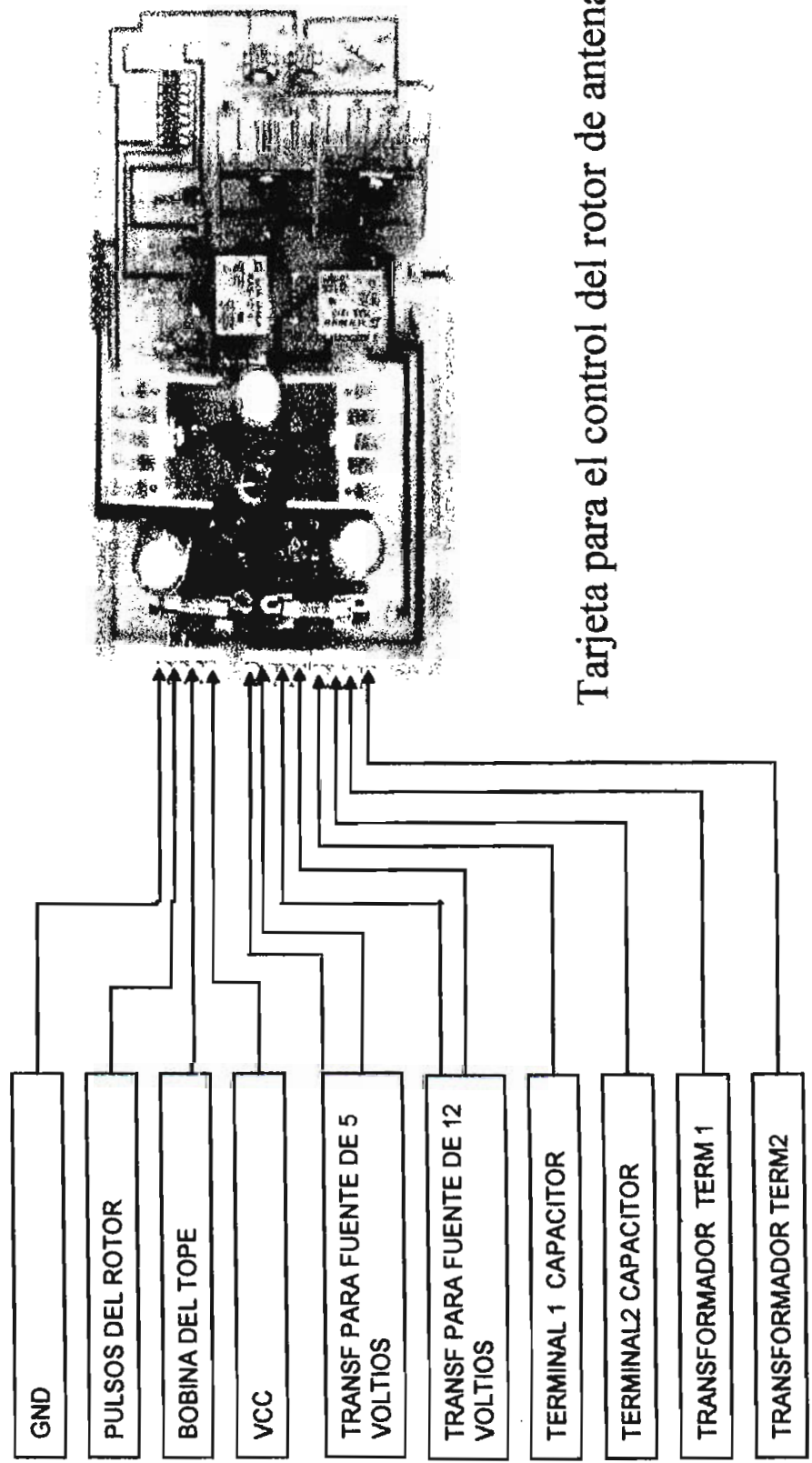
Cont... Sistema de adquisición de datos



Tarjeta de señalización y temporización.

Sistema de adquisición de datos

Cont...



Tarjeta para el control del rotor de antena.

Recomendaciones

Para obtener resultados óptimos al utilizar el sistema de adquisición de datos se recomienda:

- Verificar si todas las conexiones están de acuerdo al manual de usuario.
- Verificar que la señal de prueba sea de 1 KHz.
- Antes de ingresar la señal al equipo de adquisición de datos, debe asegurarse que el nivel de ésta no sea mayor a 1 Voltio pico.
- Asegurarse de que las antenas cumplan con la condición de campo lejano.
- Verificar que la antena de transmisión y la antena de recepción tengan el mismo tipo de polarización.
- Para facilitar la utilización integral del sistema se recomienda ingresar a los tutoriales diseñados para este propósito que se encuentran en el menú de ayudas del CONFRAD.

Manual de Usuario del CONFRAD

● Archivo

CONFRAD, es un sistema que permite obtener configuraciones de radiación de Antenas

● Simulación

● Control de Posición

● Adquisición de Datos

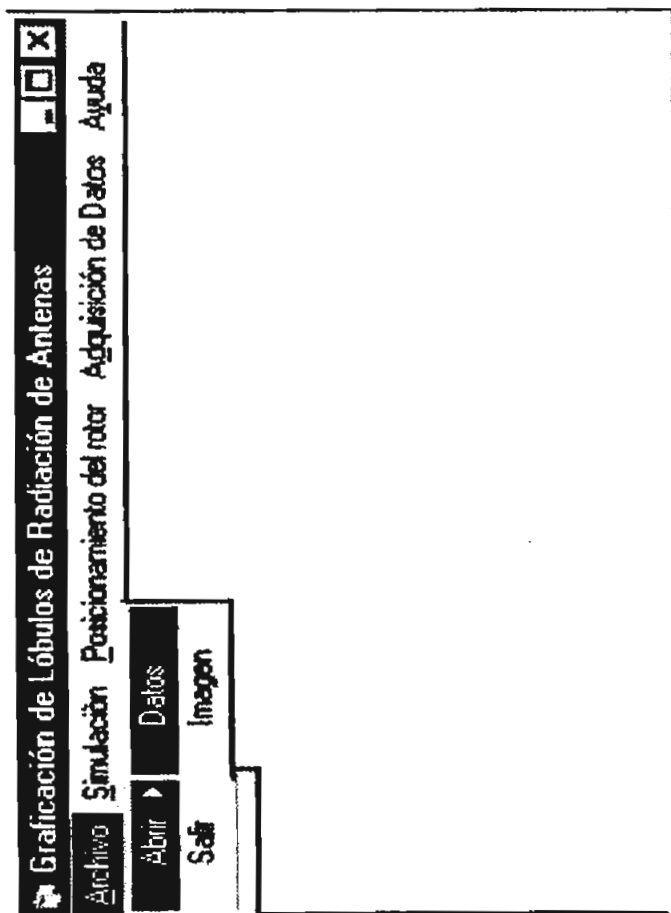
● Ayuda

Archivo

- Abrir: permite abrir archivos existentes (*.cfr) y presentarlos como datos o imágenes

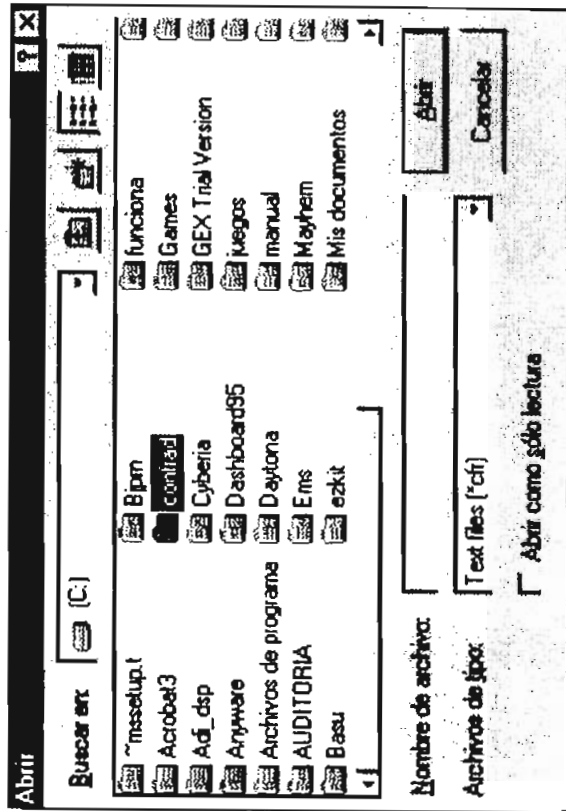


- Salir: termina la ejecución del CONFRAD.



Imagen

Al escoger esta opción se muestra la pantalla que se encuentra a la derecha y se procede a escoger el archivo



Cont...

Imagen

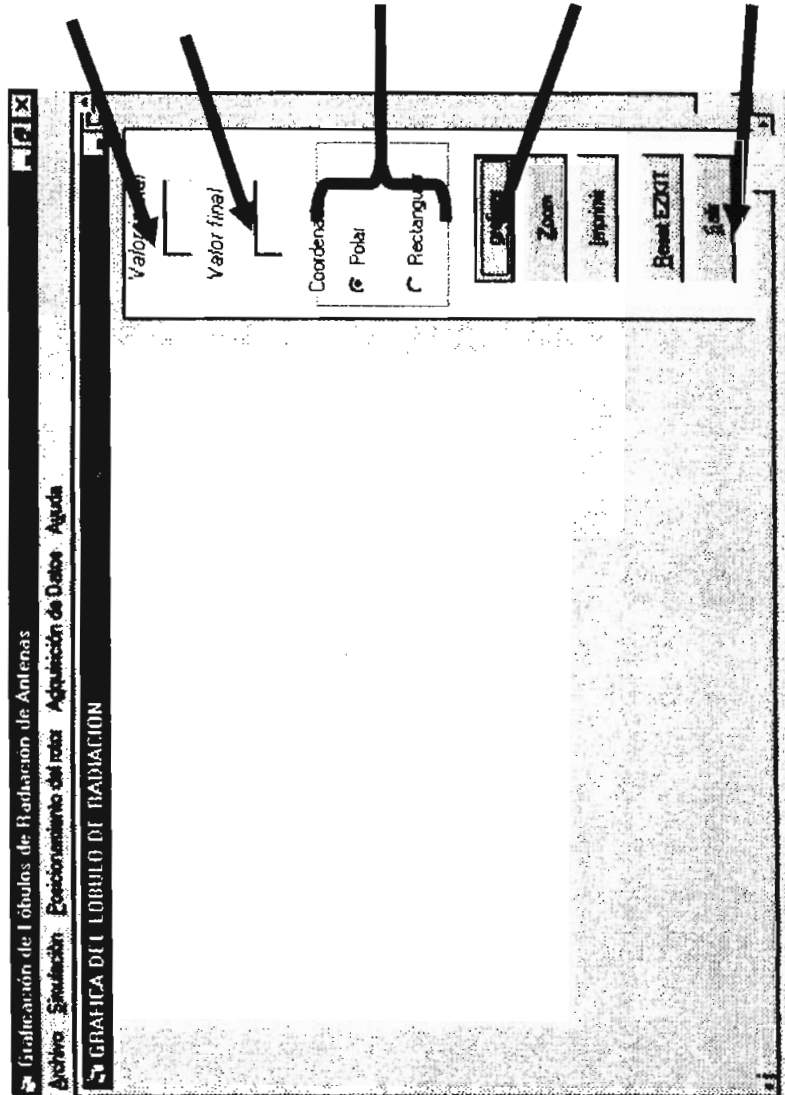
Ingrese el valor inicial en grados desde el cual desea que empiece a graficar

Ingrese el valor final en grados hasta donde desea que grafique

Seleccione el tipo de coordenadas en las que desea que se presente el gráfico

Haga click para obtener la configuración de radiación

Salir de esta opción



Cont...

Imagen

Gráfico de Voltaje Normalizado (Vn [V]) vs Posición [°].

COORDENADAS RECTANGULARES

Posición angular: 0,5607477

Voltaje normalizado: 1

Valor inicial: 0

Valor final: 360

Coordenadas: Polar, Rectangular

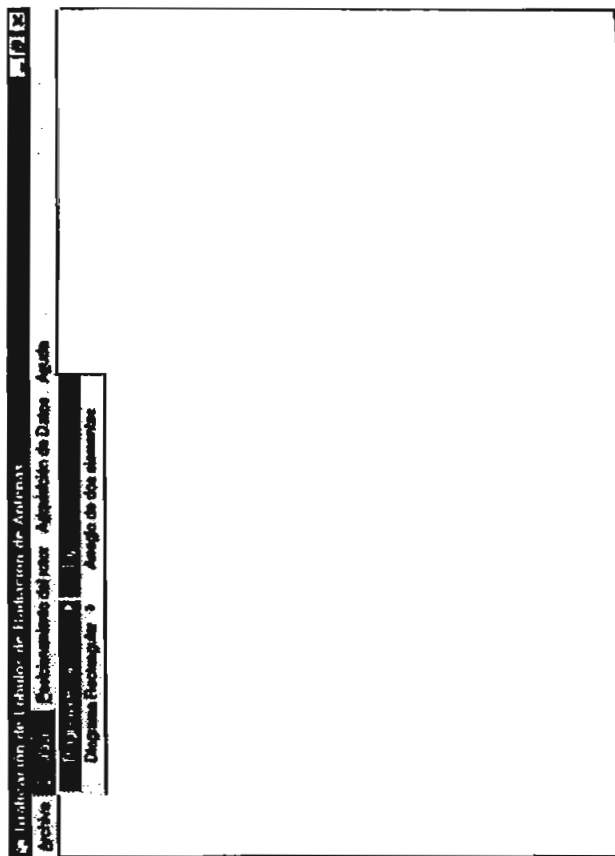
Botones: graficar, Zoom, Imprimir, Ejecutar, Salir

Annotations:

- Down: al hacer click se obtiene información de voltaje y posición de la gráfica
- Up: al hacer click se obtiene información de voltaje y posición de la gráfica
- Posición angular
- Voltaje normalizado
- Zoom: permite presentar el gráfico en el intervalo de posición escogido
- Imprimir: permite imprimir la configuración de radiación

Simulación

Permite simular las configuraciones de radiación de dipolos y arreglos de dos elementos tanto en coordenadas polares como en rectangulares



■ Dipolo

■ Arreglo de dos elementos

Dipolo

The screenshot shows a software window titled "Posicionamiento del rotar" with a menu bar containing "Archivo", "Simulación", "Posicionamiento del rotar", "Ajustación de Datos", and "Ayuda". The main area is divided into two sections:

- Left Section:** A polar plot titled "COORDENADAS POLARES" showing a radiation pattern with lobes along the 0° and 180° axes. The plot includes concentric dashed circles and radial lines labeled at 0°, 45°, 90°, 135°, 180°, 225°, 270°, and 315°.
- Right Section:** A control panel with the following elements:
 - A text field labeled "Longitud de la antena: (en long. de onda)" containing the value "1,125".
 - A section titled "TIPOS DE DIPOLOS" with two radio buttons: "Dipolo Corto" (unselected) and "Dipolo largo" (selected).
 - Four buttons: "Graficar", "Imprimir", "parar", and "salir".

Five black arrows point from text instructions to the corresponding UI elements:

- "Se debe ingresar el valor de la longitud del dipolo en longitudes de onda" points to the "Longitud de la antena" field.
- "Escoger el tipo de dipolo" points to the "Dipolo largo" radio button.
- "Presenta en pantalla la configuración de radiación del dipolo seleccionado" points to the "Graficar" button.
- "Imprime la imagen" points to the "Imprimir" button.
- "Reinicia la simulación" points to the "parar" button.
- "Finaliza la simulación" points to the "salir" button.

Arreglo de dos elementos

Se debe ingresar la distancia de separación entre dos elementos en longitudes de onda

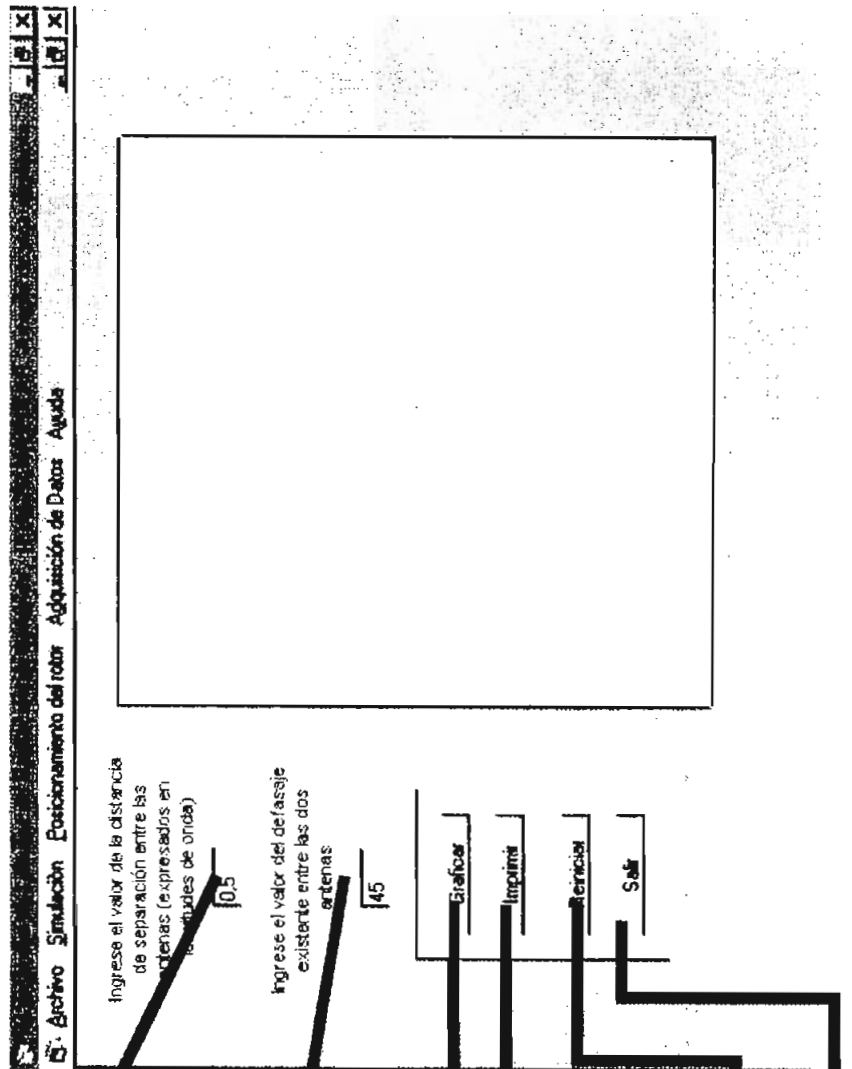
Ingrese el ángulo de desfase existente entre los elementos.

Presenta en pantalla la configuración de radiación.

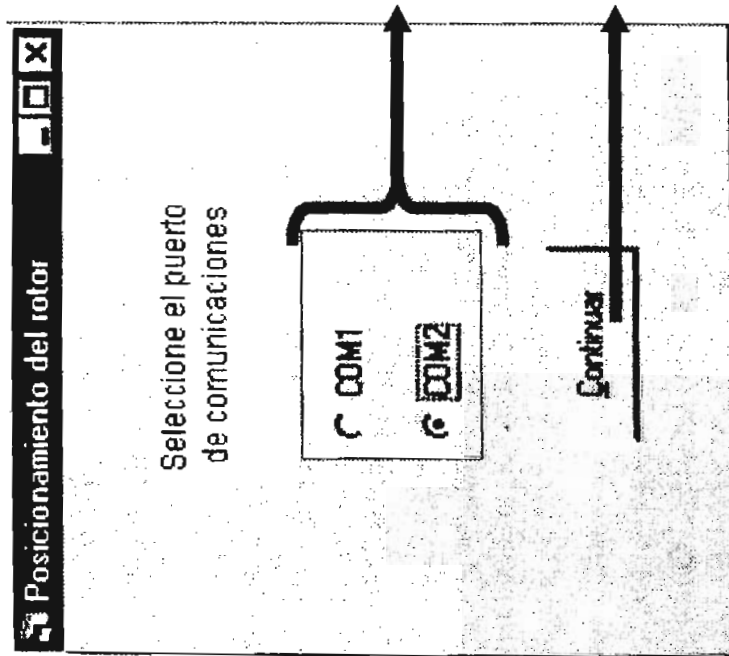
Imprime la imagen obtenida

Reinicia la simulación

Sale de la simulación.



Control de posición del rotor



Cont... Control de posición del ...

Se ingresa al programa EZ-KIT Lite Monitor

En el menú Loading se selecciona la opción Download user program...

Download user programs

File View Demo Loading Options Help
Download user program and Go...
Download user program
Go...
Upload data memory...
Upload program memory...
Download data memory...
Download program memory...

Si ya descargo el programa y lo cerró, haga click en la flecha

1. Seleccione **LOADING** en la barra de menus
2. Haga click en la opción **Download user program...**
3. Seleccione el archivo **Posicion.exe** y presione **Aceptar**
4. Cierre el programa **Ezkit - Lite Monitor**

GUIA



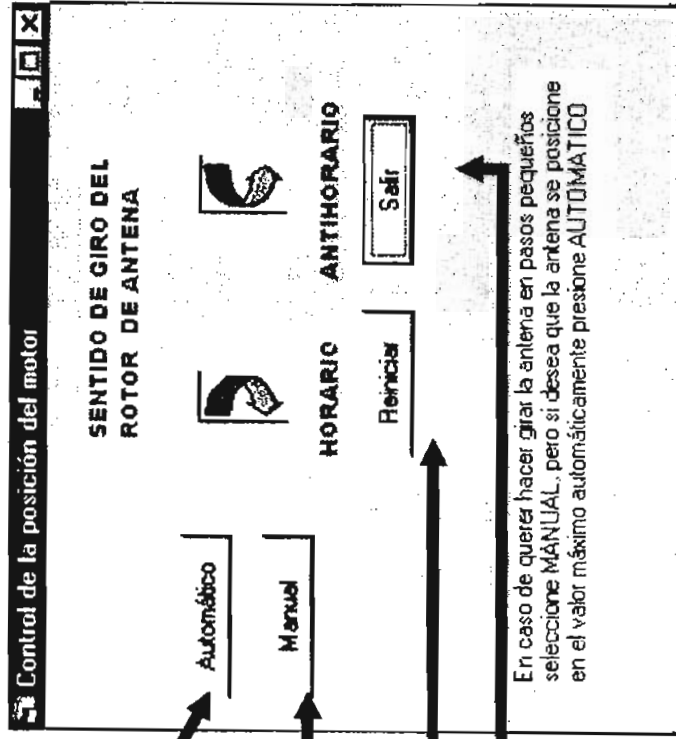
Cont... Control de posición ...

Al hacer click en Automático, el rotor se ubica en la posición donde es máximo el nivel de la señal captada y se sale de esta pantalla

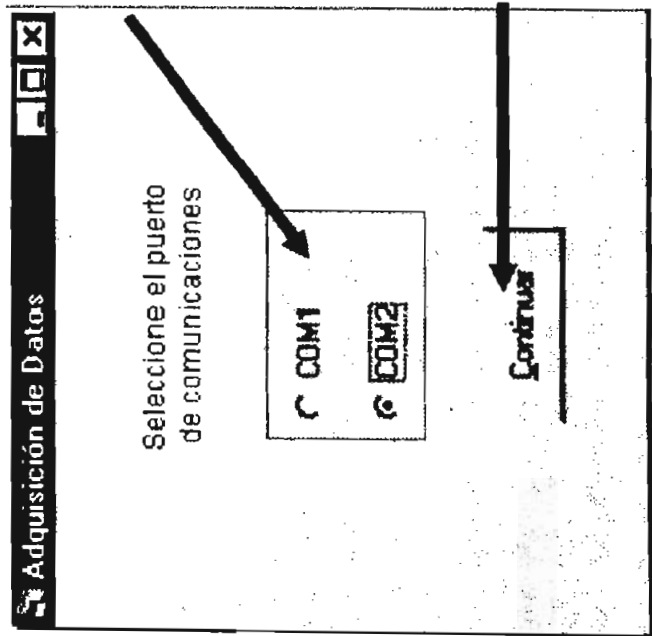
Al hacer click en manual se activa el sentido horario y antihorario y se deshabilita la opción automática.

Permite reiniciar el control activando las dos opciones nuevamente: manual y automático

Salir del control de posición



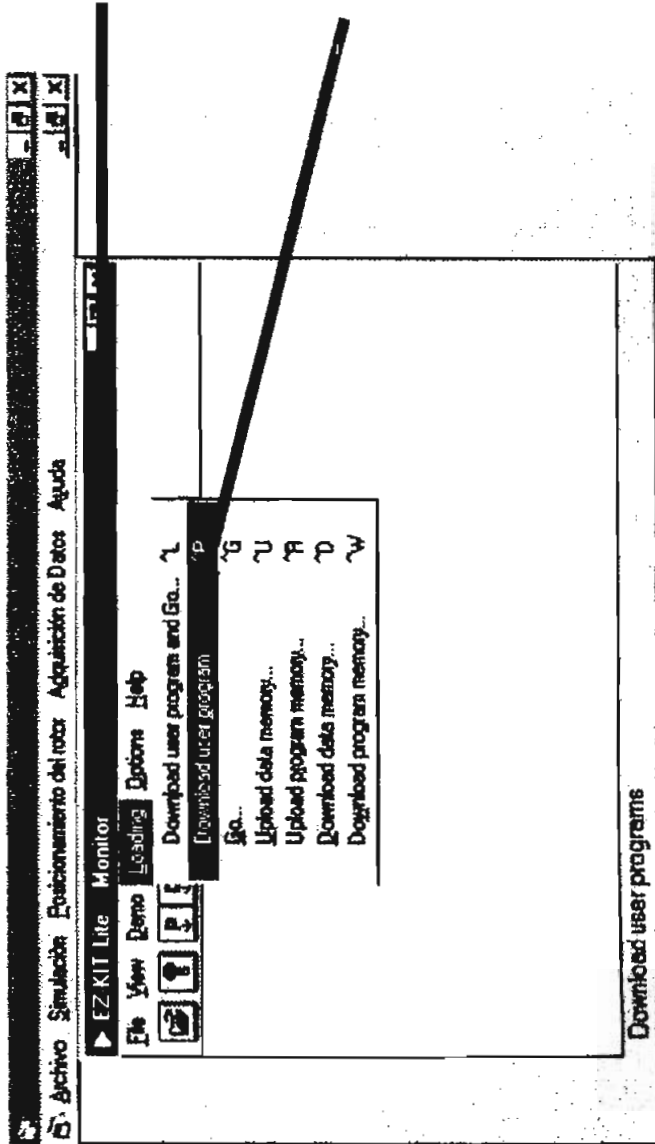
Adquisición de datos



Seleccione el puerto de comunicaciones con el que se trabaja.

Haga click en continuar para seguir con el proceso de adquisición.

Cont... Adquisición de datos



Se ingresa al programa
EZ-KIT Lite Monitor

En el menú Loading se
selecciona la opción
Download user program...



Si ya descargo el
programa y lo
cierro haga click
en la flecha

1. Seleccione LOADING en la barra de menús
2. Haga click en la opción Download user program...
3. Seleccione el archivo Adquisic.exe y presione Aceptar
4. Cierre el programa Ezkit - Lite Monitor

GUIA

Cont... Adquisición de datos

Adquisición de Datos

Los datos a adquirirse van a ser guardados en un archivo de texto. Ingrese el nombre del archivo con el cual usted quiere que se guarde.

Debe ingresar el nombre del archivo con el cual quiere que se guarde los datos

Haga click para continuar

Continuar

Cont... Adquisición de datos

The screenshot shows the 'Adquisición de Datos' software interface. The title bar reads 'Gráfica de Lóbulos de Radiación de Antenas' and the menu bar includes 'Archivo', 'Simulación', 'Posicionamiento del rotor', and 'Adquisición de Datos'. The main menu is 'GRATICA DEL LOBULO DE RADIACION'. The interface features several input fields and buttons: 'Valor inicial', 'Valor final', a 'Coordenadas' section with radio buttons for 'Polar' and 'Rectangular', a 'Gráfico' button, a 'Zoom' button, an 'Imprimir' button, a 'Reset EZKIT' button, and a 'Salir' button. Arrows from the text on the right point to these specific elements.

Ingrese el valor inicial en grados desde el cual desea que empiece a graficar

Ingrese el valor final en grados hasta donde desea que grafique

Seleccione el tipo de coordenadas en las que desea que se presente el gráfico

Haga click para obtener la configuración de radiación

Salir de esta opción

Cont... Adquisición de datos

Archivo Simulador Posicionamiento del motor Adquisición de Datos Ayuda

COORDENADAS POLARES

Valor inicial: 0
Valor final: 360

Coordenadas:
 Polar
 Rectangular

graficas
Zoom
Imprimir
Resetea el equipo
Salir

Posicion angular: 1,421495
Voltaje normalizado: 1

Down: al hacer click se obtiene información de voltaje y posición de la gráfica

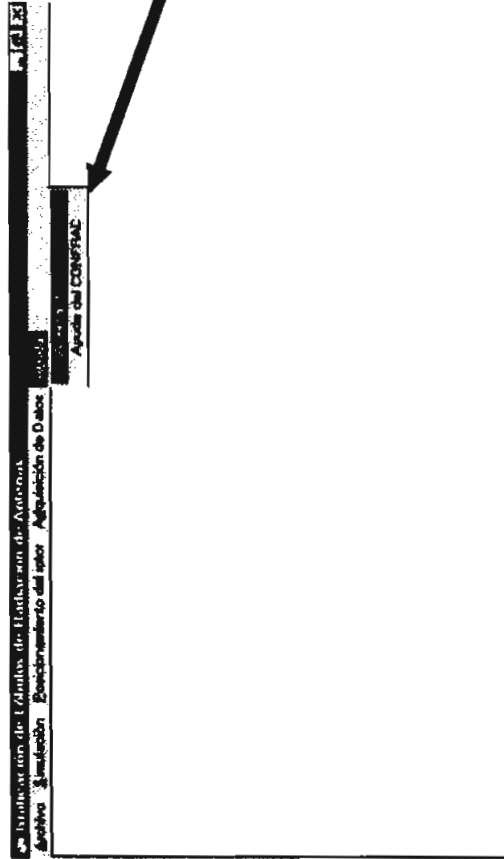
Up: al hacer click se obtiene información de voltaje y posición de la gráfica

Zoom: permite presentar el gráfico en el intervalo de posición escogido

Imprimir: permite imprimir la configuración de radiación

Resetea el equipo

Ayuda



Menú de la ayuda

ANEXO C

LISTADO DE LOS
PROGRAMAS
IMPLEMENTADOS EN EL
EZKIT_LITE

Apéndice C : LISTADO DE LOS PROGRAMAS IMPLEMENTADOS EN EL EZKIT_LITE

C.1. Listado del programa Adquisic.dsp

```
.module/ram/abs=0      Programa que permite adquirir y procesar datos;
.const IDMA=           0x3fe0;
.const BDMA_BIAD=      0x3fe1;
.const BDMA_BEAD=      0x3fe2;
.const BDMA_BDMA_Ctrl= 0x3fe3;
.const BDMA_BWCOUNT=   0x3fe4;
.const PFDATA=         0x3fe5;
.const PFTYPE=         0x3fe6;
.const SPORT1_Autobuf= 0x3fef;
.const SPORT1_RFSDIV=  0x3ff0;
.const SPORT1_SCLKDIV= 0x3ff1;
.const SPORT1_Control_Reg= 0x3ff2;
.const SPORT0_Autobuf= 0x3ff3;
.const SPORT0_RFSDIV=  0x3ff4;
.const SPORT0_SCLKDIV= 0x3ff5;
.const SPORT0_Control_Reg= 0x3ff6;
.const SPORT0_TX_Channels0= 0x3ff7;
.const SPORT0_TX_Channels1= 0x3ff8;
.const SPORT0_RX_Channels0= 0x3ff9;
.const SPORT0_RX_Channels1= 0x3ffa;
.const TSCALE=         0x3ffb;
.const TCOUNT=       0x3ffc;
.const TPERIOD=        0x3ffd;
.const DM_Wait_Reg=    0x3ffe;
.const Sys_Ctrl_Reg=   0x3fff;

.external init_uart;
.external turn_rx_on;
.external turn_rx_off;
.external out_char_ax1;
.external get_char_ax1;
.external process_a_bit;
.external baud_period;

.const N      =      256;
.const tabla  =      2571;

.var/dm/ram/circ rx_buf[3];    { Status + L data + R data }
.var/dm/ram/circ tx_buf[3];    { Cmd + L data + R data }
.var/dm/ram/circ init_cmds[13]; { 13 registros indexados }

.var/pm/ram/circ fir_coefs[N];
.var/dm/ram/circ delay_data[N];
.var/dm/ram/circ tbl1[tabla];

.var/dm stat_flag;
```

```
.var/dm fin_flag;
.var/dm giro;
.var/dm contador;
.var/dm posicion;
.var/dm dato;
.var/dm valanterior;
.var/dm maxinicio;
.var/dm maxfinal;
.var/dm posini;
.var/dm cntmedio;
```

```
.init tx_buf:      0xc000, 0x0000, 0x0000;
.init init_cmds:  0xc002, 0xc102, 0xc288, 0xc388,
                  0xc488, 0xc588, 0xc680, 0xc780,
                  0xc850, 0xc909, 0xca00, 0xcc40,
                  0xcd00;
```

```
*****
{
                          Tabla del vector interrupciones
*****
}*****
```

```
    jump start; rti; rti; rti;      /* reset */
    jump sensor; rti; rti; rti;    /* IRQ2 */
    rti; rti; rti; rti;           /* IRQL1 */
    rti; rti; rti; rti;           /* IRQL0 */
    ar = dm(stat_flag);
    ar = pass ar;
    if eq rti;
    jump next_cmd;                 /* SPORT0 TX */
    jump input_samples; rti; rti; rti; /* SPORT0 RX */
    rti; rti; rti; rti;           /* IRQE */
    rti; rti; rti; rti;           /* BDMA */
    rti; rti; rti; rti;           /* IRQ1 */
    rti; rti; rti; rti;           /* IRQ0 */
    jump process_a_bit; rti; rti; rti; /* TIMER */
    rti; rti; rti; rti;           /* POWER DOWN */
```

START:

```
.init fir_coefs:< c:\adi_dsp\21xx\bin\fir3.dat >;
i2=^delay_data;m2=1;l2=%delay_data;
i6=^fir_coefs;m6=1;l6=%fir_coefs;
AR=189; {velocidad de transmisión del puerto serial 57600bps}
DM(Baud_Period) = AR;
CALL Init_UART;
imask= 0x0031;
CNTR = 15000;
DO XLOOP UNTIL CE;
XLOOP: NOP;
CNTR =15000;
DO YLOOP UNTIL CE;
YLOOP: NOP;
CALL Turn_RX_On;
```

```

ax0=0;
dm(giro) = ax0;
dm(contador) = ax0;
ax0=2572;
dm(posicion)=ax0;

cntr = 1;
DO MLOOP until ce;
    CALL Get_Char_AX1;
    dm(giro)=ax1;
MLOOP: NOP;

ax0 = 0x0037;
    ay0 = dm(giro);
    ar = ax0 - ay0;
if ne jump continuar;

ax0 = 0x7bff;
    dm(0x3fe6)=ax0;
ax0 = 0x0080;
    dm(0x3fe5)=ax0;

continuar:
    ax0 = 0x0039;
    ay0 = dm(giro);
    ar = ax0 - ay0;
    if eq jump arranque;

PF0:          { arranque del motor }
ax0 = 0x7bff;
    dm(0x3fe6)=ax0;
    ax0 = 0x0000;
    dm(0x3fe5)=ax0;
    ax0 = 0x0010;    { PF4 }
    dm(0x3fe5)=ax0;
    ax0 = 0x0011;    { PF0 y PF4 }
    dm(0x3fe5)=ax0;
    jump start1;

arranque:
ax0 = 0x7bff;
    dm(0x3fe6)=ax0;
    ax0 = 0x0008;
    dm(0x3fe5)=ax0;
    ax0 = 0x0018;    { PF4 }
    dm(0x3fe5)=ax0;
    ax0 = 0x0019;    { PF0 y PF4 }
    dm(0x3fe5)=ax0;
    JUMP START1;

```

```

{*****
Configuración del CODEC y de ADSP2181
*****}

```

start1:

```

i0 = ^rx_buf;
l0 = %rx_buf;
i1 = ^tx_buf;
l1 = %tx_buf;
i3 = ^init_cmds;
l3 = %init_cmds;
m1 = 1;
ax0 = 0x287;
dm (SPORT0_Autobuf) = ax0;
ax0 = 0;
dm (SPORT0_RFSDIV) = ax0;
dm (SPORT0_SCLKDIV) = ax0;
ax0 = 0x860f;
dm (SPORT0_Control_Reg) = ax0;
ax0 = 0x7;
dm (SPORT0_TX_Channels0) = ax0;
dm (SPORT0_TX_Channels1) = ax0;
dm (SPORT0_RX_Channels0) = ax0;
dm (SPORT0_RX_Channels1) = ax0;
ax0 = 0xff;
dm (DM_Wait_Reg) = ax0;
ax0 = 0x1000;
dm (sys_ctrl_reg) = ax0;
ifc = 0xff;
nop;
icntl = 0;
mstat = 0x40;

ax0 = 1;
dm(stat_flag) = ax0;
imask = b#0001000000;    {tx. de SPORT0 }
ax0 = dm (i1, m1);
tx0 = ax0;

```

check_init:

```

ax0 = dm (stat_flag);
af = pass ax0;
if ne jump check_init;
ay0 = 2;

```

check_aci1:

```

ax0 = dm (rx_buf);
ar = ax0 and ay0;
if eq jump check_aci1;

```

check_aci2:

```

ax0 = dm (rx_buf);
ar = ax0 and ay0;
if ne jump check_aci2;
idle;

```

```

ay0 = 0xbf3f;
ax0 = dm (init_cmds + 6);
ar = ax0 AND ay0;
dm (tx_buf) = ar;
idle;
ax0 = dm (init_cmds + 7);
ar = ax0 AND ay0;
dm (tx_buf) = ar;
idle;
ifc = 0xff;
nop;
ax0 = 0;
dm(fin_flag) = ax0;
icntl = 0x04;
nop;

```

muestreo:

```

I5 = ^TBL1;
L5 = %TBL1;
m4 = 1;
cntr=2572;
do cero1 until ce;
ax0 = 0x0000;
cero1: dm(i5,m4)=ax0;
I5 = ^TBL1;
L5 = %TBL1;
m4 = 1;
do laz2 until eq;
imask = b#1000100000;
AR=dm(dato);
ax0 = ar;
ay0=dm(valanterior);
ar=ax0-ay0;
if ge jump siga;
jump salto;
siga: dm(valanterior)= ax0;
salto: ax0=dm(posicion);
ay0=dm(contador);
ar = ax0 - ay0;
laz2: nop;

```

parar:

```

ax0 = 0x7bff;
dm(0x3fe6)=ax0;
ax0 = 0x0000; { PF0 y PF4 }
dm(0x3fe5)=ax0;
ax0 = 0;
dm(contador) = ax0;

```

transmitir:

```

ax0=dm(0x3fe5);
ay0=0x0020;
ar = ax0 or ay0;

```

```

dm(0x3fe5)=ar;
ifc = 0x00ff;
nop;
imask = b#0000000001;
ax0 = 0;
dm(Sys_Ctrl_Reg) = ax0;
ax0=0;
dm(contador)=ax0;
dm(posini)=ax0;
dm(maxinicio)=ax0;
dm(maxfinal)=ax0;
i5 = ^tbl1;
l5 = %tbl1;
m4 = 1;
cntr = tabla;
do buscar until ce;
ax0 = dm(i5,m4);
ay0 = dm(maxinicio);
ar = ax0 - ay0;
if lt jump realizar;
dm(maxinicio)=ax0;
ax0=dm(contador);
dm(posini)=ax0;
realizar: nop;
ar=dm(contador);
ar=ar+1;
dm(contador)=ar;
buscar: nop;
ax0 = 0;
dm(contador)=ax0;
l5=^tbl1;
l5 = %tbl1;
m4 = 1;

do posic until eq;
ax0 =dm(l5,m4);
ar = dm(contador);
ar = ar+1;
dm(contador)=ar;
ax0 = dm(posini);
ay0 = dm(contador);
ar = ax0 - ay0;

posic: nop;
AR=189;
DM(Baud_Period) = AR;
CALL Init_UART;
imask = 0x0031;
CNTR = 15000;
DO XLOOP1 UNTIL CE;
XLOOP1: NOP;
CNTR = 15000;
DO YLOOP1 UNTIL CE

```

```

YLOOP1: NOP;
        CALL Turn_RX_On;
        cntr = 1;
        do envio until ce;
            ax1 = 0x0032;
            call out_char_ax1;
envio: nop;

        cntr = 2571;
        DO MLOOP1 until ce;
            ax1 = DM(i5,m4);
            CALL Out_Char_AX1;
espere:  nop;
        call get_char_ax1;
        ay0 = ax1;
        ax0 = 0x0035;
        ar = ax0 - ay0;
        if ne jump espere;
MLOOP1: NOP;
espere1:  nop;
        call get_char_ax1;
        ay0 = ax1;
        ax0 = 0x0032;
        ar = ax0 - ay0;
        if ne jump espere1;
            ax0 = 0x7bff;
            dm(0x3fe6)=ax0;
            ax0 = 0x0080;    { PF7 RESET }
            dm(0x3fe5)=ax0;

rts;

input_samples:
        ena sec_reg; {realiza el filtrado de la señal de entrada}
        mx1 = dm(rx_buf + 2); { canal derecho }
        dm(i2,m2)=mx1;
        CNTR=N-1;
        mr = 0, mx0 = dm(i2,m2), my0 = pm(i6,m6);
        do convolucion until ce;
convolucion:  mr=mr+mx0*my0(ss),mx0=dm(i2,m2),my0=pm (i6,m6);
            mr=mr+mx0*my0(md);
            if mv sat mr;
            dm(tx_buf + 2)=mr1;
            dm(dato) = mr1;

rti;

```

```

{*****
Atención a la interrupción IRQ2: cuenta el número de pulsos enviados por el sensor de
posición y guarda el valor de voltaje en cada pulso.
*****}

```

```

sensor:  ax0=0;
        ar = dm(contador);
        ar = ar + 1;

```



```

dm(contador)=ar;
  SI = dm(valanterior);
  SR = ASHIFT SI BY -24 (HI);
  ay0 = 0x00ff;
  ar = sr0 and ay0;
  if eq jump nzero;
  dm(i5,m4) = ar;
  jump final1;
nzero:  ax0 = 0x0001;
  dm(i5,m4) = ax0;
  final1:  ay1 = 0;
  dm(valanterior) = ay1;
rti;

```

```

next_cmd:                                     {configuración del CODEC AD1847}

```

```

  ena sec_reg;
  ax0 = dm (i3, m1);    { init_cmds }
  dm (tx_buf) = ax0;    { tx_buf  }
  ax0 = i3;
  ay0 = ^init_cmds;
  ar = ax0 - ay0;
  if gt rti;
  ax0 = 0xaf00;
  dm (tx_buf) = ax0;
  ax0 = 0;
  dm (stat_flag) = ax0;
  rti;
.ENDMOD;

```

C.2. Listado del programa Posicion.dsp

```
.module/ram/abs=0      Programa que permite posicionar el rotor de antena en modo
                        automático o manual;

.const IDMA=           0x3fe0;      {ubicación de los registros para la
.const BDMA_BIAD=      0x3fe1;      configuración del ADSP2181}
.const BDMA_BEAD=     0x3fe2;
.const BDMA_BDMA_Ctrl= 0x3fe3;
.const BDMA_BWCOUNT=  0x3fe4;
.const PFDATA=        0x3fe5;
.const PFTYPE=        0x3fe6;
.const SPORT1_Autobuf= 0x3fef;
.const SPORT1_RFSDIV= 0x3ff0;
.const SPORT1_SCLKDIV= 0x3ff1;
.const SPORT1_Control_Reg= 0x3ff2;
.const SPORT0_Autobuf= 0x3ff3;
.const SPORT0_RFSDIV= 0x3ff4;
.const SPORT0_SCLKDIV= 0x3ff5;
.const SPORT0_Control_Reg= 0x3ff6;
.const SPORT0_TX_Channels0= 0x3ff7;
.const SPORT0_TX_Channels1= 0x3ff8;
.const SPORT0_RX_Channels0= 0x3ff9;
.const SPORT0_RX_Channels1= 0x3ffa;
.const TSCALE=        0x3ffb;
.const TCOUNT=       0x3ffc;
.const TPERIOD=       0x3ffd;
.const DM_Wait_Reg=   0x3ffe;
.const Sys_Ctrl_Reg=  0x3fff;

.external init_uart;
.external tum_rx_on;
.external tum_rx_off;
.external out_char_ax1;
.external get_char_ax1;
.external process_a_bit;
.external baud_period;
.const N = 256;
.var/dm/ram/circ rx_buf[3];
.var/dm/ram/circ tx_buf[3];
.var/dm/ram/circ init_cmds[13];
.var/pm/ram/circ fir_coefs[N];
.var/dm/ram/circ delay_data[N];
.var/dm stat_flag;
.var/dm fin_flag;
.var/dm giro;
.var/dm sen_giro;
.var/dm contador;
.var/dm valor;
.var/dm valanterior;
.var/dm comprobar;
.var/dm numero_de_vez;
.var/dm posicion;
.var/dm posicion1;
```

```

.var/dm maximo;
.var/dm bandera;
.var/dm bandera1;
.var/dm bandera3;
.var/dm manual;
.var/dm manual1;
.var/dm dato;

.init tx_buf:    0xc000, 0x0000, 0x0000;
.init init_cmds: 0xc004, 0xc104, 0xc288, 0xc388,
                0xc488, 0xc588, 0xc680, 0xc780,
                0xc851, 0xc909, 0xca00, 0xcc40,
                0xcd00;

```

```

{*****
                                     Tabla del vector interrupciones
*****}

```

```

    jump start; rti; rti; rti;      /* reset */
    jump sensor; rti; rti; rti;    /* IRQ2 */
    rti; rti; rti; rti;           /* IRQL1 */
    rti; rti; rti; rti;           /* IRQL0 */
    ar = dm(stat_flag);            /* SPORT0 */
    ar = pass ar;
    if eq rti;
    jump next_cmd;                 /* SPORT0 TX */
    jump input_samples; rti; rti; rti; /* SPORT0 RX */
    rti; rti; rti; rti;           /* IRQE */
    rti; rti; rti; rti;           /* BDMA */
    rti; rti; rti; rti;           /* IRQ1 */
    rti; rti; rti; rti;           /* IRQ0 */
    jump process_a_bit; rti; rti; rti; /* TIMER */
    rti; rti; rti; rti;           /* POWER DOWN */

```

```

START:      { inicio del programa}
            .init fir_coefs:< c:\adi_dsp\21xx\bin\fir3.dat>;
            i2=^delay_data;m2=1;l2=%delay_data;
            i6=^fir_coefs;m6=1;l6=%fir_coefs;

            AR=189;      {velocidad de transmisión del puerto serial 57600bps}

            DM(Baud_Period) = AR;
            CALL Init_UART;
            CNTR = 15000;
            DO XLOOP UNTIL CE;
XLOOP:     NOP;
            CNTR =15000;
            DO YLOOP UNTIL CE;
YLOOP:     NOP;
            CALL Turn_RX_On;
            ax0=0;      {enceramiento de las variables}
            dm(giro) = ax0;
            dm(sen_giro) = ax0;

```

```

dm(posicion1) = ax0;
dm(contador) = ax0;
dm(valanterior) = ax0;
dm(numero_de_vez) = ax0;
dm(maximo)=ax0;
dm(bandera1) = ax0;
dm(bandera3)=ax0;
dm(manual)=ax0;
dm(manual1)=ax0;
dm(dato) = ax0;

cntr = 1;
DO MLOOP until ce;
    CALL Get_Char_AX1;
    dm(manual)=ax1;
MLOOP:    NOP;
ax0 = 0x0035;      { 5 automatico , posicion=2571 }
ay0 = dm(manual);  { 3 manual, posicion = 20 }
ar = ax0 - ay0;
if ne jump posi;
    ax0=2571;
    dm(posicion)=ax0;
    cntr = 1;
    DO MLOOP1 until ce;
        CALL Get_Char_AX1;
        dm(giro)=ax1;
MLOOP1:    NOP;
ax0 = 0x0039;
ay0 = dm(giro);
ar = ax0 - ay0;
if eq jump arranque;

PF0:    nop;
ax0 = 0x0001;
dm(sen_giro)=ax0;
ax0 = 0x7bff;
dm(0x3fe6)=ax0;
ax0 = 0x0000;
dm(0x3fe5)=ax0;
ax0 = 0x0010;    { PF4 }
dm(0x3fe5)=ax0;
cntr = 7500;
do tope1 until ce;
tope1:    nop;
ax0 = 0x7bff;
dm(0x3fe6)=ax0;
ax0 = 0x0011;    { PF0 y PF4 }
dm(0x3fe5)=ax0;
jump start1;

arranque:
ax0 = 0x7bff;
dm(0x3fe6)=ax0;

```

```

    ax0 = 0x0008;
    dm(0x3fe5)=ax0;
    ax0 = 0x0018;    { PF4 }
    dm(0x3fe5)=ax0;
    cntr = 7500;
    do tope until ce;
tope:   nop;
        ax0 = 0x7bff;
        dm(0x3fe6)=ax0;
        ax0 = 0x0019;    { PF0 y PF4 }

        jump START1;

posi:
    ax0 = 0x0037;
    ay0 = dm(manual);    { Si es 7 se resetea el ezkit }
    ar = ax0 - ay0;
    if ne jump salt;
        ax0 = 0x7bff;
        dm(0x3fe6)=ax0;
        ax0 = 0x0080;
        dm(0x3fe5)=ax0;
salt:
    ax0=71;                {permite girar al rotor en pasos de 10°}
    dm(posicion)=ax0;
    cntr = 1;
    DO MLOOP2 until ce;
        CALL Get_Char_AX1;
        dm(giro)=ax1;
    MLOOP2: NOP;
    ax0=0x0003;
    dm(manual1)=ax0;
    jump start1;

```

```

{*****
Configuración del CODEC y de ADSP2181
*****}

```

```

start1:
    i0 = ^rx_buf;
    l0 = %rx_buf;
    i1 = ^tx_buf;
    l1 = %tx_buf;
    i3 = ^init_cmds;
    l3 = %init_cmds;
    m1 = 1;
    ax0 = 0x287;
    dm (SPORT0_Autobuf) = ax0;
    ax0 = 0;
    dm (SPORT0_RFSDIV) = ax0;
    dm (SPORT0_SCLKDIV) = ax0;
    ax0 = 0x860f;
    dm (SPORT0_Control_Reg) = ax0;
    ax0 = 0x7;

```

```

dm (SPORT0_TX_Channels0) = ax0;
dm (SPORT0_TX_Channels1) = ax0;
dm (SPORT0_RX_Channels0) = ax0;
dm (SPORT0_RX_Channels1) = ax0;
ax0 = 0xff;
dm (DM_Wait_Reg) = ax0;
ax0 = 0x1000;
dm (sys_ctrl_reg) = ax0;
ifc = 0xff;
nop;
icntl = 0;
mstat = 0x40;
ax0 = 1;
dm(stat_flag) = ax0;
imask = b#0001000000;
ax0 = dm (i1, m1);
tx0 = ax0;
check_init:
    ax0 = dm (stat_flag);
    af = pass ax0;
    if ne jump check_init;
    ay0 = 2;
check_aci1:
    ax0 = dm (rx_buf);
    ar = ax0 and ay0;
    if eq jump check_aci1;

check_aci2:
    ax0 = dm (rx_buf);
    ar = ax0 and ay0;
    if ne jump check_aci2;
    idle;
    ay0 = 0xbf3f;
    ax0 = dm (init_cmds + 6);
    ar = ax0 AND ay0;
    dm (tx_buf) = ar;
    idle;
    ax0 = dm (init_cmds + 7);
    ar = ax0 AND ay0;
    dm (tx_buf) = ar;
    idle;
    ifc = 0xff;
    nop;
    ax0 = 0;
    dm(fin_flag) = ax0;
    icntl = 0x04;
    nop;

muestreo:
    ax0=0x0003;
    ay0=dm(manual1);
    ar=ax0-ay0;
    if ne jump auto;

```

```

ax0=0x0001;
dm(numero_de_vez)=ax0;
ax0=dm(posicion);
dm(posicion1)=ax0;
nop;

```

auto:

```

ax0=0x0001;
dm(bandera3)=ax0;
ar=dm(numero_de_vez);
ar = pass ar;
if eq jump seguir;
  ax0=dm(posicion1);
  dm(posicion)=ax0;
  nop;
  ax0=0x0003;
  ay0=dm(manual1);
  ar=ax0-ay0;
  if ne jump autom1;
    ax0=0x0034;
    ay0=dm(giro);
    ar=ax0-ay0;
    if eq jump PF01;
      jump arranq;

```

autom1:

```

ax0=dm(sen_giro);
ay0=0x0001;
ar=ax0 and ay0;
if eq jump PF01;

```

arranq:

```

ax0 = 0x7bff;
dm(0x3fe6)=ax0;
ax0 = 0x0008;
dm(0x3fe5)=ax0;
ax0 = 0x0018;    { PF3,PF4 }
dm(0x3fe5)=ax0;
cntr = 7500;
do tope0 until ce;

```

tope0:

```

nop;
ax0 = 0x7bff;
dm(0x3fe6)=ax0;
ax0 = 0x0019;    { PF0,PF3 y PF4 }
dm(0x3fe5)=ax0;
jump seguir;

```

PF01:

```

ax0 = 0x7bff;
dm(0x3fe6)=ax0;
ax0 = 0x0000;
dm(0x3fe5)=ax0;
ax0 = 0x0010;    { PF4 }
dm(0x3fe5)=ax0;
cntr = 7500;
do tope01 until ce;

```

```

tope01:  nop;
         ax0 = 0x7bff;
         dm(0x3fe6)=ax0;
         ax0 = 0x0011;    { set PF0 y PF4 arranca y activa el tope del motor }
         dm(0x3fe5)=ax0;
         jump seguir;

seguir:
         cntr=N;
         do cero until ce;
cero:    dm(i2,m2)=0;

         do laz2 until eq;
         imask = b#1000100000;
         ena sec_reg;
aux1:    ax0= dm(dato); { canal derecho }
         ar = abs ax0;
         ax0 = ar;
         ay0 = 0x003f;
         ar = ax0 - ay0;
         if gt jump continue;
         jump aux1;
continue: dm(i2,m2)=mx1;          {realiza el filtrado de la señal de entrada}
          CNTR=N-1;
          mr=0,mx0=dm(i2,m2),my0=pm(i6,m6);
          do convolucion until ce;
          convolucion: mr=mr+mx0*my0(ss),mx0=dm(i2,m2),my0=pm(i6,m6);
          mr=mr+mx0*my0(rnd);
          ax1=mx1;
          dm(tx_buf + 2)=ax1;

          AR=AX1;
          ax0 = ar;
          ay0=dm(valanterior);
          ar=ax0-ay0;
          if gt jump siga;
          jump salto;
siga:    dm(valanterior)= ax0;
salto:   nop;
         ax0 =dm(posicion);
         ay0 = dm(contador);
         ar = ax0 - ay0;          {compara el número de pulsos con el valor}
         laz2:  nop;              {que se encuentra en dm(posicion)}

parar:
         ax0 = 0x7bff;
         dm(0x3fe6)=ax0;
         ax0 = 0x0000;    { PF0 y PF4 }
         dm(0x3fe5)=ax0;
         CNTR = 15000;
         DO ESPERA1 UNTIL CE;
ESPERA1: NOP;

```



```

        ax0 = 0;
        dm(contador) = ax0;
        ar=dm(numero_de_vez);
        ar=ar+1;
        dm(numero_de_vez)=ar;
        ax0=ar;
        ay0=0x0002;
        ar=ax0 - ay0;
        if ne jump muestreo;
transmitir:  nop;
        ax0=0x0001;
        ay0=dm(bandera3);
        ar=ax0-ay0;
        if eq jump final1;
                jump start;
final1:     nop;
rts;

```

```

{*****
Atención a la interrupción del SPORT0 de Recepción: guarda el valor del buffer de
recepción en la variable dato.
*****}

```

```

input_samples:
        mx1 = dm(rx_buf + 2); { canal derecho }
        dm(dato) = mx1;
rti;

```

```

{*****
Atención a la interrupción IRQ2: cuenta el número de pulsos enviados por el sensor de
posición.
*****}

```

```

sensor:
        ay0 = dm(bandera);
        ax0 = 0x0005;
        ar = ax0 - ay0;
        if ne jump tran;
                ifc = 0x00ff;
                nop;
                icntl=0;
                imask = 0x0001;
                nop;
                jump transmitir;

```

```

tran:
        ar = dm(contador);
        ar = ar + 1;
        dm(contador)=ar;
        ax0=dm(valanterior);
        ay0=dm(maximo);
        ar=ax0-ay0;
        if lt jump prosiga;

```

```
dm(maximo)=ax0;
ax0=2571;
ay0=dm(contador);
ar=ax0-ay0;
dm(posicion1)=ar;
```

prosga:

```
ay0 = dm(bandera1);
ax0 = 0x0001;
ar = ax0 - ay0;
if ne jump final;
    SI = dm(valanterior);
    SR = ASHIFT SI BY -24 (HI);
    ay0 = 0x00ff;
    ar = sr0 and ay0;
    dm(i5,m4) = ar;
```

final:

```
ay1 = 0;
dm(valanterior) = ay1;
```

rti;

next_cmd:

{configuración del CODEC AD1847}

```
ena sec_reg;
ax0 = dm (i3, m1); { init_cmds }
dm (tx_buf) = ax0; { tx_buf }
ax0 = i3;
ay0 = ^init_cmds;
ar = ax0 - ay0;
if gt rti;
    ax0 = 0xaf00;
    dm (tx_buf) = ax0;
    ax0 = 0;
    dm (stat_flag) = ax0;
```

rti;

.ENDMOD;

C.3. Listado del programa UART.dsp

```
.module/ram          UART;

.const tx_num_of_bits = 10;
.const rx_num_of_bits = 8;
.const RX_BIT_ADD = 0x0100;
.const TX_BIT_ADD = 0xfe00;
.const PERIOD =      1144;
.const TSCALE=      0x3ffb;
.const TCOUNT=     0x3ffc;
.const TPERIOD=     0x3ffd;
.const DM_Wait_Reg= 0x3ffe;
.const Sys_Ctrl_Reg= 0x3fff;

.entry init_uart;
.entry out_char_ax1;
.entry get_char_ax1;
.entry turn_rx_on;
.entry turn_rx_off;
.entry process_a_bit;

.global flag_rx_ready;
.global baud_period;

.var flag_tx_ready;
.var flag_rx_ready;
.var flag_rx_stop_yet;
.var flag_rx_no_word;
.var flag_rx_off;
.var timer_tx_ctr;
.var timer_rx_ctr;
.var user_tx_buffer;
.var user_rx_buffer;
.var internal_tx_buffer;
.var internal_rx_buffer;
.var bits_left_in_tx;
.var bits_left_in_rx;
.var baud_period;

init_uart:
    ax0=0;
    dm(TSCALE)=ax0;
    ax0=dm(baud_period);
    dm(TCOUNT)=ax0;
    dm(TPERIOD)=ax0;
    ax0=0;
    dm(Sys_Ctrl_Reg)=ax0;

    ax0=1;
    dm(flag_tx_ready)=ax0;
    dm(flag_rx_ready)=ax0;
```

```
dm(flag_rx_stop_yet)=ax0;
dm(flag_rx_no_word)=ax0;
dm(flag_rx_off)=ax0;
```

```
set flag_out;
ifc=0x003f;
nop;
imask=b#000001;
ena timer;
rts;
```

process_a_bit:

```
ena sec_reg;
ax0=dm(flag_tx_ready);
ar=pass ax0;
if ne jump receiver;
```

{ _____ Sección de Transmisión _____ }

```
ay0=dm(timer_tx_ctr);
ar=ay0-1;
dm(timer_tx_ctr)=ar;
if ne jump receiver;
```

```
sr1=dm(internal_tx_buffer);
sr=lshift sr1 by -1 (hi);
dm(internal_tx_buffer)=sr1;
ar=pass sr0;
if ge reset flag_out;
if lt set flag_out;
toggle fl1;
ay0=3;
dm(timer_tx_ctr)=ay0;
```

```
ay0=dm(bits_left_in_tx);
ar=ay0-1;
dm(bits_left_in_tx)=ar;
if gt jump receiver;
```

```
ax0=1;
dm(flag_tx_ready)=ax0;
```

{ _____ Sección de Recepción _____ }

receiver:

```
ax0=dm(flag_rx_off);
ar=pass ax0;
if ne rti;
```

```
ax0=dm(flag_rx_stop_yet);
ar=pass ax0;
if ne jump rx_test_busy;
```

```
ay0=dm(timer_rx_ctr);
```

```

ar=ay0-1;
dm(timer_rx_ctr)=ar;
if ne rti;

ax0=1;
dm(flag_rx_stop_yet)=ax0;
dm(flag_rx_ready)=ax0;

ax0=dm(internal_rx_buffer);
dm(user_rx_buffer)=ax0;

ax0=0;
dm(flag_rx_no_word)=ax0;
rti;

rx_test_busy:
ax0=dm(flag_rx_ready);
ar=pass ax0;
if eq jump rx_busy;

if flag_in jump rx_exit;

ax0=0;
dm(flag_rx_ready)=ax0;
dm(internal_rx_buffer)=ax0;

ax0=4;
dm(timer_rx_ctr)=ax0;

ax0=rx_num_of_bits;
dm(bits_left_in_rx)=ax0;
rx_exit:
rti;

rx_busy:
ay0=dm(timer_rx_ctr);
ar=ay0-1;
dm(timer_rx_ctr)=ar;
if ne rti;

rcv:
ax0=3;
dm(timer_rx_ctr)=ax0;

ay0=RX_BIT_ADD;
ar=dm(internal_rx_buffer);
if not flag_in jump pad_zero;
ar=ar+ay0;

pad_zero:
sr=lshift ar by -1 (lo);
dm(internal_rx_buffer)=sr0;

```

```
ay0=dm(bits_left_in_rx);
ar=ay0-1;
dm(bits_left_in_rx)=ar;
if gt rti;
```

```
ax0=3;
dm(timer_rx_ctr)=ax0;
ax0=0;
dm(flag_rx_stop_yet)=ax0;
rti;
```

invoke_UART_transmit:

```
ax0=3;
dm(timer_tx_ctr)=ax0;
```

```
ax0=tx_num_of_bits;
dm(bits_left_in_tx)=ax0;
sr1=0;
sr0=TX_BIT_ADD;
ar=dm(user_tx_buffer);
sr=sr or lshift ar by 1 (lo);
dm(internal_tx_buffer)=sr0;
```

```
ax0=0;
dm(flag_tx_ready)=ax0;
rts;
```

get_char_ax1:

```
ax0=dm(flag_rx_no_word);
ar=pass ax0;
if ne jump get_char_ax1;
```

```
ax1=dm(user_rx_buffer);
ax0=1;
dm(flag_rx_no_word)=ax0;
```

rts;

out_char_ax1:

```
ax0=dm(flag_tx_ready);
ar=pass ax0;
if eq jump out_char_ax1;
dm(user_tx_buffer)=ax1;
call invoke_UART_transmit;
rts;
```

turn_rx_on:

```
ax0=0;
dm(flag_rx_off)=ax0;
rts;
```

turn_rx_off:

```
ax0=1;
```

```
dm(flag_rx_off)=ax0;  
rts;
```

```
.endmod;
```

ANEXO D

CODIGO DEL PROGRAMA
CONFRAD

Apéndice D : Código del programa CONFRAD

D.1. Código para pantallas de presentación

```
Private Sub cmdcontinuar_Click()  
frmpresent2.Show  
Unload Me  
End Sub
```

```
Private Sub cmdcontinuar_Click()  
frmMDI.Show  
Unload Me  
End Sub
```

D.2. Código de programa forma principal

```
Option Explicit  
Private Sub MDIForm_Load()  
ChDir App.Path  
ReDim Document(1)  
    ReDim FState(1)  
    FState(1).Dirty = False  
End Sub
```

```
Private Sub MDIForm_Unload(Cancel As Integer)  
    If Not AnyPadsLeft() Then  
        End  
    End If  
End Sub
```

```
Private Sub mnuayuda_Click()  
    CMDialog1.HelpFile = "c:\confrad\confrad.hlp"  
    CMDialog1.HelpCommand = cdlHelpContents  
    CMDialog1.ShowHelp  
End Sub
```

```
Private Sub mnuadatos_Click()  
    AbrirArchivo  
End Sub
```

```
Private Sub mnuaimagen_click()  
    intabrir = 1  
    AbrirArchivolimagen  
End Sub
```

```
Private Sub mnucontrolmanual_Click()  
    posgraf = 1  
    frmprad.Show  
End Sub
```

```
Private Sub mnuRecentFile_Click(index As Integer)
    OpenFile (mnuRecentFile(index).Caption)
GetRecentFiles
End Sub
```

```
Private Sub mnuadqu_Click()
    posgraf = 0
    intabrir = 0
    frmprad.Show
End Sub
```

```
Private Sub mnuasalir_Click()
    Unload Me
End Sub
```

```
Private Sub mnusddarr_Click()
    frmarrreglo.Show
End Sub
```

```
Private Sub mnusdipolo_Click()
    frmdipolo.Show
End Sub
```

```
Private Sub mnusdrarr_Click()
    frmrearrreglo.Show
End Sub
```

```
Private Sub mnusdrdipolo_Click()
    frmrectdip.Show
End Sub
```

D.3. Código de programa para abrir archivos

```
Option Explicit
Private Sub Form_Load()
    Me.Top = 0
    Me.Left = 0
    Me.Width = 6000
    Me.Height = 4000
End Sub
```

```
Private Sub Form_Resize()
    Text1.Height = ScaleHeight
    Text1.Width = ScaleWidth
End Sub
```

```
Private Sub mnuFontName_Click(index As Integer)
    Text1.FontName = mnuFontName(index).Caption
End Sub
```

```
Private Sub mnuAbrirArchivo_Click()
    AbrirArchivo
End Sub
```

D.4. Código de programa para la simulación de dipolos en coordenadas polares.

```
Option Explicit
Dim Angrad, Pi, AngFi1(), AngFi2(), X0, Y0, Xpos(), Ypos(), L, P(), intq(), g(), i
Dim X, Y, bandera, strMensaje, strbandera1
Dim intKeyAscii As Integer
Dim comparar, comparar1
Dim N As Integer

Private Sub cmdejecutar_Click()
Do While bandera = 0
  If txtlongitud.Text = "" Then
    strMensaje = "No se ha ingresado el valor de la longitud de la antena"
    MsgBox strMensaje, 16
    Exit Sub
  Else
    If IsNumeric(txtlongitud.Text) Then
      L = txtlongitud.Text
      If L <= 0 Then
        txtlongitud.Text = ""
        strMensaje = "Este no es un valor de longitud permitido"
        MsgBox strMensaje, 16
        Exit Sub
      End If
      If optcorto.Value = -1 Then
        If L > 0.25 Then
          txtlongitud.Text = ""
          strMensaje = "Este no es un valor de longitud de un dipolo corto"
          MsgBox strMensaje, 16
          Exit Sub
        End If
      End If
      If optlargo.Value = -1 Then
        If L < 0.25 Then
          txtlongitud.Text = ""
          strMensaje = "Este valor es el de un dipolo corto"
          MsgBox strMensaje, 16
          Exit Sub
        End If
      End If
      bandera = 1
    Else
      strMensaje = "El dato ingresado como longitud no es un valor numérico"
      MsgBox strMensaje, 16
      Exit Sub
    End If
  End If
Loop
picboxpantalla.DrawWidth = 1
picboxpantalla.DrawStyle = 2
picboxpantalla.DrawMode = 1
For i = 0 To 2000 Step 400
  picboxpantalla.Circle (2800, 2500), i, QBColor(3)
Next i
picboxpantalla.Line (800, 2500)-(4800, 2500), QBColor(3)
```

```

picboxpantalla.Line (2800, 500)-(2800, 4500), QBColor(3)
picboxpantalla.Line (1385, 1086)-(4214, 3914), QBColor(3)
picboxpantalla.Line (1385, 3914)-(4214, 1086), QBColor(3)
picboxpantalla.CurrentX = 1850
picboxpantalla.CurrentY = 50
picboxpantalla.FontSize = 8
picboxpantalla.Print "COORDENADAS POLARES"
ReDim Xpos(360)
ReDim Ypos(360)
ReDim AngFi1(360)
If optcorto.Value = -1 Then
  If L > 0.25 Then
    txtlongitud.Text = ""
    strMensaje = "Este valor es el de un dipolo largo"
    MsgBox strMensaje, 16
    Exit Sub
  End If

  For i = 0 To 360 Step 0.01
    Angrad = i * Pi / 180
    AngFi1(i) = Abs(Sin(Angrad))
    Xpos(i) = AngFi1(i) * Sin(Angrad)
    Ypos(i) = AngFi1(i) * Cos(Angrad)
    X = X0 + 2000 * Xpos(i)
    Y = Y0 - 2000 * Ypos(i)
    picboxpantalla.PSet (X, Y)
  Next i
Else
  If optlargo.Value = -1 Then
    If L < 0.26 Then
      txtlongitud.Text = ""
      strMensaje = "Este valor es el de un dipolo corto"
      MsgBox strMensaje, 16
      Exit Sub
    End If
  End If

  lblongitud.Visible = -1
  txtlongitud.Visible = -1
  Call Iniciar
  ReDim Xpos(360)
  ReDim Ypos(360)
  ReDim AngFi2(360)
  comparar = 0
  For i = 1 To 360 Step 0.05
    Angrad = i * Pi / 180
    AngFi2(i) = Abs((Cos((Pi * L) * Cos(Angrad)) - Cos(Pi * L)) / Cos(Angrad + (Pi / 2)))
    If AngFi2(i) > comparar Then
      comparar = AngFi2(i)
    End If
  Next i

  For i = 1 To 360 Step 0.05
    Angrad = i * Pi / 180

```

```

        AngFi2(i) = (Abs((Cos((Pi * L) * Cos(Angrad)) - Cos(Pi * L)) / Cos(Angrad + (Pi / 2)))) /
        comparar
        Xpos(i) = AngFi2(i) * Sin(Angrad)
        Ypos(i) = AngFi2(i) * Cos(Angrad)
        X = X0 + 2000 * Xpos(i)
        Y = Y0 - 2000 * Ypos(i)
        picboxpantalla.PSet (X, Y)
    Next i
    comparar = 0
End If
End Sub

Private Sub cmdreiniciar_Click()
bandera = 0
txtlongitud = ""
picboxpantalla.Cls
End Sub

Private Sub cmdsalir_Click()
Unload Me
End Sub
Private Sub Form_Load()
Pi = 3.14159265359
X0 = 2800
Y0 = 2500
lblongitud.Visible = -1
txtlongitud.Visible = -1
bandera = 0
End Sub

Private Sub txtlongitud_KeyPress(intKeyAscii As Integer)
If (intKeyAscii = 13) Then
End If
End Sub

```

D.5. Código de programa para simular dipolos en coordenadas rectangulares

```

Option Explicit
Dim Angrad, Angrad1, Pi, L, valor1(), AngFi1(), W(), z(), AngFi2(), X0, Y0, Xpos(),
Ypos(), i, j
Dim X, Y, bandera, strMensaje, strbandera1, comparar1, cuenta, cuenta1, cuenta2
Dim intKeyAscii, N As Integer

Private Sub cmdgraficar_Click()
Do While bandera = 0
    If txtlongitud.Text = "" Then
        strMensaje = "No se ha ingresado el valor de la longitud de la antena"
        MsgBox strMensaje, 16
        Exit Sub
    Else
        If IsNumeric(txtlongitud.Text) Then
            L = txtlongitud.Text
            bandera = 1
        Else

```

```

        strMensaje = "El dato ingresado como longitud no es un valor numérico"
        MsgBox strMensaje, 16
        Exit Sub
    End If
End If
Loop
picboxrecdip.Line (500, 3200)-(5100, 3200), QBColor(3)
picboxrecdip.Line (500, 800)-(500, 3200), QBColor(3)
picboxrecdip.CurrentX = 200
picboxrecdip.CurrentY = 250
picboxrecdip.Print "Voltaje"
picboxrecdip.CurrentX = 200
picboxrecdip.CurrentY = 500
picboxrecdip.Print "Normalizado"
picboxrecdip.CurrentX = 4500
picboxrecdip.CurrentY = 3300
picboxrecdip.Print "Posición"
picboxrecdip.CurrentX = 4500
picboxrecdip.CurrentY = 3500
picboxrecdip.Print "Angulo"
If optcorto.Value = -1 Then
    ReDim Xpos(450)
    ReDim Ypos(450)
    ReDim AngFi1(450)

    For i = 90 To 450 Step 0.01
        Angrad = i * Pi / 180
        j = i - 90
        Angrad1 = j * Pi / 180

        AngFi1(i) = Abs(Sin(Angrad))
        Xpos(j) = Angrad1
        Ypos(i) = AngFi1(i)
        X = X0 + 700 * Xpos(j)
        Y = Y0 - 2100 * Ypos(i)
        picboxrecdip.PSet (X, Y)
    Next i
Else

    Lbllongitud.Visible = -1
    txtlongitud.Visible = -1
    Call Iniciar
    ReDim Xpos(720)
    ReDim Ypos(720)
    ReDim AngFi2(720)
    ReDim valor1(720)
    comparar1 = 0
    cuenta = 0
    cuenta1 = 0
    For i = 0 To 360 'Step 0.01
        Angrad = i * Pi / 180
        valor1(i) = Abs((Cos((Pi * L) * Cos(Angrad)) - Cos(Pi * L)) / Cos(Angrad + (Pi / 2)))
        If valor1(i) > comparar1 Then
            comparar1 = valor1(i)
            cuenta1 = cuenta

```

```

End If
    cuenta = cuenta + 1
Next i

For i = cuenta1 To cuenta1 + 360 Step 0.03
    j = i - cuenta1
    Angrad = i * Pi / 180
    Angrad1 = j * Pi / 180
    AngFi2(i) = Abs((Cos((Pi * L) * Cos(Angrad)) - Cos(Pi * L)) / Cos(Angrad + (Pi / 2)))
    / comparar1
    Xpos(j) = Angrad1
    Ypos(i) = AngFi2(i)
    X0 = 145
    X = (X0 + 200 * Xpos(j)) * 3.5
    Y = Y0 - 2100 * Ypos(i)
    picboxrecdip.PSet (X, Y)
Next i
End If
End Sub

Private Sub cmdreiniciar_Click()
bandera = 0
txtlongitud = ""
picboxrecdip.Cls
End Sub

Private Sub cmdsalir_Click()
Unload Me
End Sub

Private Sub Form_Load()
Pi = 3.14159265359
X0 = 500
Y0 = 3180
Lbllongitud.Visible = -1
txtlongitud.Visible = -1
bandera = 0
End Sub

```

D.6. Código de programa para simular arreglos de dos elementos en coordenadas polares.

```

Option Explicit
Dim Angrad, Pi, Angulo, A, D, X0, Y0, Xpos(), Ypos(), valor(), L, i
Dim X, Y, bandera, strMensaje, strbandera1, comparar
Dim intKeyAscii As Integer

Public Sub Iniciar( )
Do While bandera = 0
    If txtd.Text = "" Then
        strMensaje = "No se ha ingresado el valor de la longitud de la antena"
        MsgBox strMensaje, 16
        Exit Sub
    Else

```

```

If IsNumeric(txtlongitud.Text) Then
    D = txtlongitud.Text
    bandera = 1
Else
    strMensaje = "El dato ingresado como longitud no es un valor numérico"
    MsgBox strMensaje, 16
    Exit Sub
End If
End If
If txtalfa.Text = "" Then
    strMensaje = "No se ha ingresado el valor de la longitud de la antena"
    MsgBox strMensaje, 16
    Exit Sub
Else
    If IsNumeric(txtalfa.Text) Then
        A = txtalfa.Text
        bandera = 1
    Else
        strMensaje = "El dato ingresado como longitud no es un valor numérico"
        MsgBox strMensaje, 16
        Exit Sub
    End If
End If
Loop
End Sub

Option Explicit
Dim D, Alfa, Angulo, valor(), Angrad, Pi, AngFi1(), X0, Y0, X, Y, Xpos(), Ypos()
Dim bandera, i

Private Sub cmdgraficar_Click()
    comparar = 0
    Do While bandera = 0
        If txtlongitud.Text = "" Then
            strMensaje = "No se ha ingresado el valor de la longitud de la antena"
            MsgBox strMensaje, 16
            Exit Sub
        Else
            If IsNumeric(txtlongitud.Text) Then
                D = txtlongitud.Text
                bandera = 1
            Else
                strMensaje = "El dato ingresado como longitud no es un valor numérico"
                MsgBox strMensaje, 16
                Exit Sub
            End If
        End If

        If txtalfa.Text = "" Then
            strMensaje = "No se ha ingresado el valor del angulo de desfase de la antena"
            MsgBox strMensaje, 16
            Exit Sub
        Else
            If IsNumeric(txtalfa.Text) Then
                A = txtalfa.Text
            End If
        End If
    End Do
End Sub

```



```

    If IsNumeric(txtlongitud.Text) Then
        D = txtlongitud.Text
        bandera = 1
    Else
        strMensaje = "El dato ingresado como longitud no es un valor numérico"
        MsgBox strMensaje, 16
        Exit Sub
    End If
End If
If txtalfa.Text = "" Then
    strMensaje = "No se ha ingresado el valor de la longitud de la antena"
    MsgBox strMensaje, 16
    Exit Sub
Else
    If IsNumeric(txtalfa.Text) Then
        A = txtalfa.Text
        bandera = 1
    Else
        strMensaje = "El dato ingresado como longitud no es un valor numérico"
        MsgBox strMensaje, 16
        Exit Sub
    End If
End If
Loop
End Sub

Option Explicit
Dim D, Alfa, Angulo, valor(), Angrad, Pi, AngFi1(), X0, Y0, X, Y, Xpos(), Ypos()
Dim bandera, i

Private Sub cmdgraficar_Click()
    comparar = 0
    Do While bandera = 0
        If txtlongitud.Text = "" Then
            strMensaje = "No se ha ingresado el valor de la longitud de la antena"
            MsgBox strMensaje, 16
            Exit Sub
        Else
            If IsNumeric(txtlongitud.Text) Then
                D = txtlongitud.Text
                bandera = 1
            Else
                strMensaje = "El dato ingresado como longitud no es un valor numérico"
                MsgBox strMensaje, 16
                Exit Sub
            End If
        End If
    End While

    If txtalfa.Text = "" Then
        strMensaje = "No se ha ingresado el valor del angulo de desfase de la antena"
        MsgBox strMensaje, 16
        Exit Sub
    Else
        If IsNumeric(txtalfa.Text) Then
            A = txtalfa.Text

```

```
End Sub
```

```
Private Sub cmdreinicio_Click()
```

```
bandera = 0
```

```
txtd = ""
```

```
txtalfa = ""
```

```
picboxarreglo.Cls
```

```
End Sub
```

```
Private Sub cmdsalir_Click()
```

```
Unload Me
```

```
End Sub
```

```
Private Sub Form_Load()
```

```
Pi = 3.14159265359
```

```
X0 = 2500
```

```
Y0 = 2800
```

```
bandera = 0
```

```
End Sub
```

```
Private Sub txtalfa_KeyPress(intKeyAscii As Integer)
```

```
If (intKeyAscii = 13) Then
```

```
End If
```

```
End Sub
```

```
Private Sub txtd_KeyPress(intKeyAscii As Integer)
```

```
If (intKeyAscii = 13) Then
```

```
End If
```

```
End Sub
```

D.7. Código para simulación de arreglos de dos elementos en coordenadas rectangulares

```
Option Explicit
```

```
Dim Angrad, Pi, Angulo, A, Dis, X0, Y0, Xpos(), Ypos(), valor(), L, i
```

```
Dim X, Y, bandera, strMensaje, strbandera1, comparar
```

```
Dim intKeyAscii As Integer
```

```
Private Sub cmdgraficar_Click()
```

```
Do While bandera = 0
```

```
    If IsNumeric(txtdistancia.Text) Then
```

```
        Dis = txtdistancia.Text
```

```
        bandera = 1
```

```
    Else
```

```
        strMensaje = "El dato ingresado como longitud no es un valor numérico"
```

```
        MsgBox strMensaje, 16
```

```
        Exit Sub
```

```
    End If
```

```
    If IsNumeric(Txtalfa.Text) Then
```

```
        A = Txtalfa.Text
```

```
        bandera = 1
```

```
    Else
```

```

        strMensaje = "El dato ingresado como longitud no es un valor numérico"
        MsgBox strMensaje, 16
        Exit Sub
    End If
Loop
ReDim Xpos(360)
ReDim Ypos(360)
ReDim valor(360)
picboxrecarreglo.Line (200, 3200)-(5000, 3200), QBColor(3)
picboxrecarreglo.Line (200, 800)-(200, 3200), QBColor(3)
picboxrecarreglo.CurrentX = 200
picboxrecarreglo.CurrentY = 250
picboxrecarreglo.Print "Voltaje"
picboxrecarreglo.CurrentX = 200
picboxrecarreglo.CurrentY = 500
picboxrecarreglo.Print "Normalizado"
picboxrecarreglo.CurrentX = 4300
picboxrecarreglo.CurrentY = 3300
picboxrecarreglo.Print "Posición"
picboxrecarreglo.CurrentX = 4300
picboxrecarreglo.CurrentY = 3500
picboxrecarreglo.Print "Angulo"
For i = 0 To 360 Step 1
    Angrad = i * Pi / 180
    Angulo = A * Pi / 360
    valor(i) = Abs(Cos((Pi * Dis * Cos(Angrad)) + Angulo))
    If valor(i) > comparar Then
        comparar = valor(i)
    End If
Next i
For i = 0 To 360 Step 0.01
    Angrad = i * Pi / 180
    Angulo = A * Pi / 360
    valor(i) = Abs(Cos((Pi * Dis * Cos(Angrad)) + Angulo)) / comparar
    Xpos(i) = i
    Ypos(i) = valor(i)
    X = X0 + 13 * Xpos(i)
    Y = Y0 - 2200 * Ypos(i)
    picboxrecarreglo.PSet (X, Y)
Next i
End Sub

Private Sub cmdreiniciar_Click()
bandera = 0
txtdistancia = ""
Ttxtalfa = ""
picboxrecarreglo.Cls
End Sub

Private Sub cmdsalir_Click()
Unload Me
End Sub

Private Sub Form_Load()
Me.Top = 0

```

```

Me.Left = 0
Me.Width = 9200
Me.Height = 7000
Pi = 3.14159265359
X0 = 200
Y0 = 3200
bandera = 0
End Sub

```

```

Private Sub txtalfa_KeyPress(intKeyAscii As Integer)
    If (intKeyAscii = 13) Then
        End If
    End Sub

```

```

Private Sub txtd_KeyPress(intKeyAscii As Integer)
    If (intKeyAscii = 13) Then
        End If
    End Sub

```

D.8. Código de programa para escoger puerto de comunicaciones

```

Private Sub cmdcontinuar_Click()
    If optcom1.Value = -1 Then
        intcom = 1
    Else
        intcom = 2
    End If

```

```

Dim retval
If posgraf = 0 Then
    lblprograma.Caption = "Adquisic.exe"
Else
    lblprograma.Caption = "Posicionar.exe"
End If

```

```

frmprad.WindowState = 2
retval = Shell("C:\adi_dsp\21xx\ezkit\2181\ezkitapp.EXE", 1)
cmdcontinuar.Visible = False
picboxflecha.Visible = True
fraport.Visible = False
optcom1.Visible = False
optcom2.Visible = False
lblmensaje.Visible = False
End Sub

```

```

Private Sub picboxflecha_Click()
    frmprad.WindowState = 0
    Unload frmprad
    If posgraf = 0 Then
        frmprevio.Show
    Else
        frmcontrolmanual.Show
    End If
End Sub

```

```
MSComm1.InputLen = 1
MSComm1.Output = "7"
MSComm1.InBufferCount = 0
MSComm1.PortOpen = False
End Sub
```

```
Private Sub cmdautomático_Click()
Dim strg, strh As String
cmdmanual1.Enabled = False
MSComm1.CommPort = intcom
MSComm1.Settings = "57600,N,8,1"
MSComm1.PortOpen = True
MSComm1.InputLen = 1
MSComm1.Output = "5"
MSComm1.InBufferCount = 0
MSComm1.PortOpen = False
Open "c:\confrad\giro" For Input As #1
    strg = Input(1, #1)
Close #1
strh = strg
If strg = "1" Then
    Open "c:\confrad\giro" For Output As #1
        Write #1, 9
    Close #1
Else
    Open "c:\confrad\giro" For Output As #1
        Write #1, 1
    Close #1
End If
MSComm1.CommPort = intcom
MSComm1.Settings = "57600,N,8,1"
MSComm1.PortOpen = True
MSComm1.InputLen = 1
MSComm1.Output = strh
MSComm1.InBufferCount = 0
MSComm1.PortOpen = False
End Sub
```

```
Private Sub cmdreiniciar_Click()
cmdautomático.Enabled = True
cmdmanual1.Enabled = True
picboxhorario.Enabled = False
picboxantih.Enabled = False
End Sub
```

```
Private Sub cmdsalir_Click()
MSComm1.CommPort = intcom
MSComm1.Settings = "57600,N,8,1"
MSComm1.PortOpen = True
MSComm1.InputLen = 1
MSComm1.Output = "7"
MSComm1.InBufferCount = 0
MSComm1.PortOpen = False
Unload Me
End Sub
```

```

Private Sub cmdmanual1_Click()
lblhorario.BackColor = &H800000
lblantih.BackColor = &H800000
cmdautomático.Enabled = False
picboxhorario.Enabled = True
picboxantih.Enabled = True
MSComm1.CommPort = intcom
MSComm1.Settings = "57600,N,8,1"
MSComm1.PortOpen = True
MSComm1.InputLen = 1
MSComm1.Output = "3"
MSComm1.InBufferCount = 0
MSComm1.PortOpen = False
End Sub

```

```

Private Sub Form_Load()
Me.Top = 1500
Me.Left = 2500
Me.Width = 5500
Me.Height = 5000
picboxhorario.Enabled = False
picboxantih.Enabled = False
End Sub
Private Sub Optantihorario_Click()
    intmanualgiro = "1"
End Sub

```

```

Private Sub Opthorario_Click()
    intmanualgiro = "9"
End Sub

```

```

Private Sub picboxantih_Click()
Dim intmanualgiro As String
intmanualgiro = "4"
MSComm1.CommPort = intcom
MSComm1.Settings = "57600,N,8,1"
MSComm1.PortOpen = True
MSComm1.InputLen = 1
MSComm1.Output = intmanualgiro
MSComm1.InBufferCount = 0
MSComm1.PortOpen = False
End Sub

```

```

Private Sub picboxhorario_Click()
Dim intmanualgiro As String
intmanualgiro = "6"
MSComm1.CommPort = intcom
MSComm1.Settings = "57600,N,8,1"
MSComm1.PortOpen = True
MSComm1.InputLen = 1
MSComm1.Output = intmanualgiro
MSComm1.InBufferCount = 0
MSComm1.PortOpen = False
End Sub

```

D.10. Código de programa previo a la adquisición de datos

```
Private Sub Continuar_Click()
If txtname.Text = "" Then
    strMensaje = "Ingrese el nombre del archivo"
    MsgBox strMensaje, 16
    Exit Sub
Else
    stropenfilename1 = txtname.Text & ".cfr"
    Open stropenfilename1 For Append As #1
        If FileLen(stropenfilename1) > 0 Then
            Close #1
            strMensaje = "Este archivo ya existe, ingrese otro nombre"
            MsgBox strMensaje, 16
            Exit Sub
        Else
            End If
        Close #1
    End If
    frmgrafica.Show
    Unload Me
End Sub

Private Sub Form_Load()
Me.Top = 1000
Me.Left = 2500
Me.Width = 4000
Me.Height = 4000
End Sub
```

D.11. Código de programa para la adquisición de datos y obtención de configuraciones de radiación.

```
Option Explicit
Dim Angrad, Pi, radio(), X, Y, posicion(), i, D, A, posangular
Dim intzoomin, intzoomout, j, intaux As Integer
Dim strcad, strc, strE, strg, strMensaje, g1, strbandera1, strvalor As String
Dim intb1, k, intcont, intcont2, intcontador, intso, intq, intzoomdif, intzoomres,
intcoordenadas, intzoom As Integer

Private Sub cmdgraf1_Click()
Dim strg2, strh, strm1, strvalor As String
Dim bc, Ac, Dc, k, m, intaux, intaux1 As Integer
Dim msg, halfwidth, halfheight
Dim resultado As Double
Dim a1, B1c, i, j As Double
Dim X0, Y0
Dim posiciontext As TextBox
intcontador = 0
cmdresetezkit.Enabled = False
Open stropenfilename1 For Input As #1
Do While Not EOF(1)
    Line Input #1, strg2
    resultado = Val(strg2)
```

```

    If resultado > intnormalizar Then
        intnormalizar = resultado
    Else
    End If
        intcontador = intcontador + 1
Loop
Close #1
    intso = intcontador
    X0 = 600
    Y0 = 3500
    strbandera1 = 0
    intcoordenadas = 2
espere: If optpolar.Value = -1 Then
    intcoordenadas = 1
    Else
        If optrectangular.Value = -1 Then
            intcoordenadas = 0
        Else
            GoTo espere
        End If
    End If
End If

Do While strbandera1 = 0
    If TXT1.Text = "" Then
        strMensaje = "No se ha ingresado el valor de la longitud de la antena"
        MsgBox strMensaje, 16
        Exit Sub
    Else
        If IsNumeric(TXT1.Text) Then
            d1 = TXT1.Text
            strbandera1 = 1
        Else
            strMensaje = "El dato ingresado como longitud no es un valor numérico"
            MsgBox strMensaje, 16
            Exit Sub
        End If
    End If
    If TXT2.Text = "" Then
        strMensaje = "No se ha ingresado el valor del angulo de desfase de la antena"
        MsgBox strMensaje, 16
        Exit Sub
    Else
        If IsNumeric(TXT2.Text) Then
            A = TXT2.Text
            strbandera1 = 1
        Else
            strMensaje = "El dato ingresado como longitud no es un valor numérico"
            MsgBox strMensaje, 16
            Exit Sub
        End If
    End If
Loop
intzoomin = d1

```



```

intzoomin = Val(intzoomin)
intzoomin = intzoomin * intso / 360
intzoomout = A
intzoomout = Val(intzoomout)
intzoomout = intzoomout * intso / 360
intb1 = 5000 / (intzoomout - intzoomin + 1) ' 0.0000921195785
j = 0
ReDim radio(intso)
ReDim posicion(intso)
ReDim Angrad(intso)
AutoRedraw = True
Open stropenfilename1 For Input As #1
Do While Not EOF(1)
For i = 0 To intso - 1 Step 1
    Line Input #1, g1
    If IsNull(g1) Then
        g1 = "0"
    End If
    D = Val(g1)
    D = D / intnormalizar 'normalización
    intaux = j
    j = j + intb1
intzoomdif = intzoomin - i
intzoomres = intzoomout - i
    If intzoomdif <= 0 Then
        If intzoomres >= 0 Then
            picboxcursor.DrawWidth = 2
            picboxcursor.FontSize = 2
            If intoordenadas = 0 Then
                picboxcursor.PSet ((X0 + intaux - (intzoomin * intb1) + intb1), (Y0 - D *
                2800)), QBColor(0)
            picboxcursor.Line (560, 3510)-(5500, 3510), , B
            picboxcursor.Line (560, 450)-(560, 3510), , B
            picboxcursor.CurrentX = 500
            picboxcursor.CurrentY = 3600
            picboxcursor.FontBold = True
            picboxcursor.Print TXT1
            picboxcursor.CurrentX = 5500
            picboxcursor.CurrentY = 3600
            picboxcursor.Print TXT2
            picboxcursor.CurrentX = 100
            picboxcursor.CurrentY = 100
            picboxcursor.Print "Vn [V]"
            picboxcursor.CurrentX = 5000
            picboxcursor.CurrentY = 3820
            picboxcursor.Print "Posición [°]"
            picboxcursor.CurrentX = 1500
            picboxcursor.CurrentY = 100
            picboxcursor.FontSize = 10
            picboxcursor.Print "COORDENADAS RECTANGULARES"
        Else
            X0 = 3000
            Y0 = 2000
            Angrad = i * Pi * 2 / intso
            radio(i) = D * Cos(Angrad)
        End If
    End If
Next i
Close #1

```

```

    posicion(i) = D * Sin(Angrad)
    X = X0 + 1800 * radio(i)
    Y = Y0 - 1800 * posicion(i)
    picboxcursor.Circle (3000, 2000), 1800, QBColor(7)
    picboxcursor.Circle (3000, 2000), 1200, QBColor(7)
    picboxcursor.Circle (3000, 2000), 600, QBColor(7)
    picboxcursor.Line (1200, 2000)-(4800, 2000), QBColor(7)
    picboxcursor.Line (3000, 200)-(3000, 3800), QBColor(7)
    picboxcursor.Line (1727, 727)-(4273, 3273), QBColor(7)
    picboxcursor.Line (1727, 3273)-(4273, 727), QBColor(7)
    picboxcursor.DrawWidth = 2
    picboxcursor.PSet (X, Y), QBColor(0)
    picboxcursor.FontSize = 10
    picboxcursor.FontBold = True
    picboxcursor.Print "COORDENADAS POLARES"
End If
Else
End If
Else
End If
Next i
Loop
Close #1
AutoRedraw = False
    lblposangular.Visible = True
    picboxd.Visible = True
    picboxi.Visible = True
    lblpvoltaje.Visible = True
    lblPangular.Visible = True
    lblPosVoltaje.Visible = True
    lblfondo.Visible = True
    posangular = intzoomin
End Sub

Private Sub cmdimprimir_Click()
Dim strg2, strh, strm1, strvalor As String
Dim bc, Ac, Dc, k As Integer
Dim msg, halfwidth, halfheight
Dim resultado As Double
Dim a1, B1c, i, j As Double
Dim X0, Y0
Dim intaux, intaux1 As Integer
Dim m As Integer

Printer.ScaleLeft = -((Printer.Width - frmgrafica.Width) / 0.6)
Printer.ScaleTop = -((Printer.Height - frmgrafica.Height) / 2)
If optpolar = -1 Then
    Printer.Circle (2800, 2500), 4000, QBColor(3)
    Printer.Circle (2800, 2500), 3000, QBColor(3)
    Printer.Circle (2800, 2500), 2000, QBColor(3)
    Printer.Circle (2800, 2500), 1000, QBColor(3)
    Printer.Line (-1200, 2500)-(6800, 2500), QBColor(3)
    Printer.Line (2800, -1600)-(2800, 6500), QBColor(3)
    Printer.Line (-15, -314)-(5614, 5314), QBColor(3)
    Printer.Line (-15, 5314)-(5614, -314), QBColor(3)

```

```

Else
  Printer.Line (-550, 5010)-(6000, 5010), , B
  Printer.Line (-550, 400)-(-550, 5010), , B
End If
ReDim Xpos(360)
ReDim Ypos(360)
ReDim AngFi1(360)
intcontador = 0
Open stropenfilename1 For Input As #1
Do While Not EOF(1)
  Line Input #1, strg2
  resultado = Val(strg2)
  If resultado > intnormalizar Then
    intnormalizar = resultado
  Else
    End If
  intcontador = intcontador + 1
Loop
Close #1
  intso = intcontador
  X0 = 600
  Y0 = 3500
  strbandera1 = 0
  intcoordenadas = 2
espere: If optpolar.Value = -1 Then
  intcoordenadas = 1
  Else
  If optrectangular.Value = -1 Then
    intcoordenadas = 0
  Else
    GoTo espere
  End If
End If
intb1 = 5000 / intso
j = 0
ReDim radio(intso)
ReDim posicion(intso)
ReDim Angrad(intso)

Open stropenfilename1 For Input As #1
Do While Not EOF(1)
  For i = 0 To intso - 1 'Step 1
    Line Input #1, g1
    If IsNull(g1) Then
      g1 = "0"
    End If
    D = Val(g1)
    D = D / intnormalizar 'normalización
    intaux = j
    j = j + intb1
    If intcoordenadas = 0 Then
      X0 = -400
      Y0 = 2500
      Printer.PSet ((X0 + intaux - (intzoomin * intb1) + intb1) * 1.3, (Y0 - D * 2000) *
      2), QBColor(0)
    End If
  Next i
End Do

```

```

Else
    X0 = 2800
    Y0 = 2500
    Angrad = i * Pi * 2 / intso
    radio(i) = D * Cos(Angrad)
    posicion(i) = D * Sin(Angrad)
    X = X0 + 4000 * radio(i)
    Y = Y0 - 4000 * posicion(i)
    Printer.DrawWidth = 10
    Printer.DrawStyle = 3
    Printer.PSet (X, Y), QBColor(0)
End If
Next i
Loop
Close #1
Printer.EndDoc
End Sub

Private Sub cmdzoom_Click()
    intzoom = 1
    frmgrafica.Cls
    TXT1.Text = ""
    TXT2.Text = ""
    picboxd.Visible = False
    picboxi.Visible = False
    lblposangular.Visible = False
    lblpvoltaje.Visible = False
    lblPangular.Visible = False
    lblPosVoltaje.Visible = False
    lblfondo.Visible = False
    picboxcursor.Cls
    fraopcion.Visible = True
    strbandera3 = 2
End Sub

Private Sub cmdgraf_Click()
    Dim strcad, strc, strE, strg, strm1, strvalor As String
    Dim bc, Ac, Dc, k, m, intaux, intaux1 As Integer
    Dim D, A, d1, i, j, X0, Y0
    Dim resultado, a1, B1c As Double
    Dim posiciontext As TextBox
    If strbandera3 = 0 Then
        X0 = 400
        Y0 = 2000
        Open "c:\confrad\giro" For Input As #1
            strg = Input(1, #1)
        Close #1
        strh = strg
        If strg = "1" Then
            Open "c:\confrad\giro" For Output As #1
                Write #1, 9
            Close #1
        Else
            Open "c:\confrad\giro" For Output As #1
                Write #1, 1
            Close #1
        End If
    End If

```

```

Close #1
End If

MSComm1.CommPort = intcom
MSComm1.Settings = "57600,N,8,1"
MSComm1.PortOpen = True
MSComm1.InputLen = 1
MSComm1.Output = strh
MSComm1.InBufferCount = 0
esperar:
  Do
    DoEvents
    Loop Until MSComm1.InBufferCount >= 1
    strm1 = MSComm1.Input
    If strm1 = "2" Then
      GoTo continuar
    Else
      GoTo esperar
    End If
continuar:
intcontador = 0
For i = 1 To 2571
Open stropenfilename1 For Append As #1
Ac = Len(strE)
  bc = 1
  Dc = 1
  While bc <= Ac
    strc = Left(strE, Dc)
    strE = Right(strE, Ac - bc)
    bc = bc + 1
    If IsNull(strc) Then
      strc = 0
    End If
    k = Asc(strc)
    resultado = transforma(k)
    Write #1, resultado
    If resultado > intnormalizar Then
      intnormalizar = resultado
    Else
      End If
    intcontador = intcontador + 1
    intaux = j
    j = j + B1c
  Wend
  Close #1
  MSComm1.Output = "5"
Next i
MSComm1.Output = "5"
MSComm1.PortOpen = False
Else
End If
intso = intcontador
X0 = 600
Y0 = 2000
strbandera1 = 0

```

```

intcoordenadas = 2
espere: If optpolar.Value = -1 Then
    intcoordenadas = 1
    Else
        If optrectangular.Value = -1 Then
            intcoordenadas = 0
        Else
            GoTo espere
        End If
    End If
End If
Do While strbandera1 = 0
    If TXT1.Text = "" Then
        strMensaje = "No se ha ingresado el valor de la longitud de la antena"
        MsgBox strMensaje, 16
        Exit Sub
    Else

        If IsNumeric(TXT1.Text) Then
            d1 = TXT1.Text
            strbandera1 = 1
        Else
            strMensaje = "El dato ingresado como longitud no es un valor numérico"
            MsgBox strMensaje, 16
            Exit Sub
        End If
    End If
    If TXT2.Text = "" Then
        strMensaje = "No se ha ingresado el valor del angulo de desfase de la antena"
        MsgBox strMensaje, 16
        Exit Sub
    Else
        If IsNumeric(TXT2.Text) Then
            A = TXT2.Text
            strbandera1 = 1
        Else
            strMensaje = "El dato ingresado como longitud no es un valor
            numérico"
            MsgBox strMensaje, 16
        End If
    End If
End If
Loop
intzoomin = d1
intzoomin = Val(intzoomin)
intzoomin = intzoomin * intso / 360
intzoomout = A
intzoomout = Val(intzoomout)
intzoomout = intzoomout * intso / 360
intb1 = 5500 / (intzoomout - intzoomin + 1)
j = 0
ReDim radio(intso)
ReDim posicion(intso)
ReDim Angrad(intso)
If strh = 9 Then

```

```

    intgiroinicio = 0
    intgirofin = intso - 1
    intpaso = 1
Else
    intgiroinicio = intso - 1
    intgirofin = 0
    intpaso = -1
End If

ReDim radio(intso)
ReDim posicion(intso)
ReDim Angrad(intso)
AutoRedraw = True

Open stropenfilename1 For Input As #1
Do While Not EOF(1)
For i = intgiroinicio To intgirofin Step intpaso
    Line Input #1, g1
    If IsNull(g1) Then
        g1 = "0"
    End If
    D = Val(g1)
    D = D / intnormalizar 'normalización
    intaux = j
    j = j + intb1
    intzoomdif = intzoomin - i
    intzoomres = intzoomout - i
    If intzoomdif <= 0 Then
        If intzoomres >= 0 Then
            picboxcursor.DrawWidth = 2
            picboxcursor.FontSize = 2
            If intcoordenadas = 0 Then
                picboxcursor.PSet ((X0 + intaux - (intzoomin * intb1) + intb1), (Y0 - D *
                2800)), QBColor(0)
                picboxcursor.Line (560, 3510)-(5500, 3510), , B
                picboxcursor.Line (560, 450)-(560, 3510), , B
                picboxcursor.CurrentX = 500
                picboxcursor.CurrentY = 3600
                picboxcursor.FontBold = True
                picboxcursor.Print TXT1
                picboxcursor.CurrentX = 5500
                picboxcursor.CurrentY = 3600
                picboxcursor.Print TXT2
                picboxcursor.CurrentX = 100
                picboxcursor.CurrentY = 100
                picboxcursor.Print "Vn [V]"
                picboxcursor.CurrentX = 5000
                picboxcursor.CurrentY = 3820
                picboxcursor.Print "Posición [°]"
                picboxcursor.CurrentX = 1500
                picboxcursor.CurrentY = 100
                picboxcursor.FontSize = 10
                picboxcursor.Print "COORDENADAS RECTANGULARES"
            Else
                X0 = 3000
            End If
        End If
    End If
Next i
Close #1

```

```

Y0 = 2000
Angrad = i * Pi * 2 / intso    'i*pi/180
radio(i) = D * Cos(Angrad)
posicion(i) = D * Sin(Angrad)
X = X0 + 1800 * radio(i)
Y = Y0 - 1800 * posicion(i)
    picboxcursor.Circle (3000, 2000), 1800, QBColor(7)
    picboxcursor.Circle (3000, 2000), 1200, QBColor(7)
    picboxcursor.Circle (3000, 2000), 600, QBColor(7)
    picboxcursor.Line (1200, 2000)-(4800, 2000), QBColor(7)
    picboxcursor.Line (3000, 200)-(3000, 3800), QBColor(7)
    picboxcursor.Line (1727, 727)-(4273, 3273), QBColor(7)
    picboxcursor.Line (1727, 3273)-(4273, 727), QBColor(7)
    picboxcursor.DrawWidth = 2
    picboxcursor.PSet (X, Y), QBColor(0)
    picboxcursor.CurrentX = 2950
    picboxcursor.CurrentY = 15
    picboxcursor.Print "0°"
    picboxcursor.CurrentX = 2850
    picboxcursor.CurrentY = 3810
    picboxcursor.Print "180°"
    picboxcursor.CurrentX = 750
    picboxcursor.CurrentY = 1815
    picboxcursor.Print "270°"
    picboxcursor.CurrentX = 4900
    picboxcursor.CurrentY = 1815
    picboxcursor.Print "90°"
    picboxcursor.CurrentX = 4300
    picboxcursor.CurrentY = 600
    picboxcursor.Print "45°"
    picboxcursor.CurrentX = 4300
    picboxcursor.CurrentY = 3300
    picboxcursor.Print "135°"
    picboxcursor.CurrentX = 1300
    picboxcursor.Print "225°"
    picboxcursor.CurrentX = 1300
    picboxcursor.CurrentY = 600
    picboxcursor.Print "315°"
    picboxcursor.CurrentX = 100
    picboxcursor.CurrentY = 100
    picboxcursor.FontSize = 10
    picboxcursor.FontBold = True
    picboxcursor.Print "COORDENADAS"
    picboxcursor.CurrentX = 100
    picboxcursor.CurrentY = 350
    picboxcursor.Print "POLARES"
End If
Else
End If
Else
End If
Next i
Loop
Close #1
AutoRedraw = False

```



```
lblposangular.Visible = True
picboxd.Visible = True
picboxi.Visible = True
lblpvoltaje.Visible = True
lblPangular.Visible = True
lblPosVoltaje.Visible = True
lblfondo.Visible = True
posangular = intzoomin
```

```
End Sub
```

```
Private Sub cmdresetezkit_Click()
    MSComm1.CommPort = intcom
    MSComm1.Settings = "57600,N,8,1"
    MSComm1.PortOpen = True
    MSComm1.InputLen = 1
    MSComm1.Output = "2"
    MSComm1.PortOpen = False
End Sub
```

```
Private Sub cmdsalir_Click()
    Unload Me
End Sub
```

```
Private Sub Form_Load()
    optpolar.Value = -1
    If intabrir = 0 Then
        cmdgraf.Visible = True
        cmdgraf1.Visible = False
    Else
        cmdgraf.Visible = False
        cmdgraf1.Visible = True
    End If
    strbandera3 = 0
    intrnormalizar = 0
    intcont = 0
    intcont2 = 0
    Pi = 3.14159
    intzoom = 0
    Me.Top = 0
    Me.Left = 0
    grd.ColWidth(0) = 1000
    grd.ColWidth(1) = 1200
    grd.ColWidth(2) = 5000
    datos(0) = 2 ^ -7
    datos(1) = 2 ^ -6
    datos(2) = 2 ^ -5
    datos(3) = 2 ^ -4
    datos(4) = 2 ^ -3
    datos(5) = 2 ^ -2
    datos(6) = 2 ^ -1
    datos(7) = -1

    Me.Height = 6500
```

```

Me.Width = 10000
  lblposangular.Visible = False
  lblpvoltaje.Visible = False
  lblPangular.Visible = False
  lblPosVoltaje.Visible = False
  lblfondo.Visible = False
  picboxd.Visible = False
  picboxi.Visible = False
End Sub

```

```

Private Sub picboxd_Click() ' permite las lecturas de voltaje en forma descendente
Dim X0, Y0
posangular = CInt(posangular)
  If posangular < intzoomout Then
    posangular = posangular + 1
  Else
  End If

```

```

  lblposangular.Caption = posangular * 360 / intso
X0 = 600
Y0 = 3500
intso = intcontador
intb1 = 5000 / (intzoomout - intzoomin + 1)
j = 0

```

```

Open stropenfilename1 For Input As #1
Do While Not EOF(1)
  For i = 0 To intso - 1
    Line Input #1, g1
    If IsNull(g1) Then
      g1 = "0"
    End If
    D = Val(g1)
    D = D / intnormalizar 'normalización
    intaux = j
    j = j + intb1
    If posangular = intzoomin Then GoTo siga1

```

```

  If i = posangular - 1 Then
    If intcoordenadas = 0 Then
      X0 = 600
      Y0 = 3500
      picboxcursor.DrawWidth = 5
      picboxcursor.FontSize = 4
      picboxcursor.PSet ((X0 + intaux - (intzoomin * intb1) + intb1), (Y0 - D *
        2800)), &HFFFFFF
    Else
      X0 = 3000
      Y0 = 2000
      Angrad = i * Pi * 2 / intso 'i*pi/180
      radio(i) = D * Cos(Angrad)
      posicion(i) = D * Sin(Angrad)
      X = X0 + 1800 * radio(i)
      Y = Y0 - 1800 * posicion(i)
      picboxcursor.DrawWidth = 5

```

```
picboxcursor.FontSize = 4
picboxcursor.PSet (X, Y), &HFFFFFFF
```

```
End If
End If
```

```
If i = posangular - 1 Then
  If intcoordenadas = 0 Then
    X0 = 600
    Y0 = 3500
    picboxcursor.DrawWidth = 2
    picboxcursor.FontSize = 2
    picboxcursor.PSet ((X0 + intaux - (intzoomin * intb1) + intb1), (Y0 - D *
      2800)), QBColor(0)
  Else
    X0 = 3000
    Y0 = 2000
    Angrad = i * Pi * 2 / intso 'i*pi/180
    radio(i) = D * Cos(Angrad)
    posicion(i) = D * Sin(Angrad)
    X = X0 + 1800 * radio(i)
    Y = Y0 - 1800 * posicion(i)
    picboxcursor.DrawWidth = 2
    picboxcursor.FontSize = 2
    picboxcursor.PSet (X, Y), QBColor(0)
```

```
End If
End If
```

```
sigla1:
```

```
If i = posangular Then
  If intcoordenadas = 0 Then
    X0 = 600
    Y0 = 3500
    picboxcursor.DrawWidth = 5
    picboxcursor.FontSize = 4
    picboxcursor.PSet ((X0 + intaux - (intzoomin * intb1) + intb1), (Y0 - D *
      2800)), QBColor(12)
    lblPosVoltaje.Caption = D
  Else
    X0 = 3000
    Y0 = 2000
    Angrad = i * Pi * 2 / intso 'i*pi/180
    radio(i) = D * Cos(Angrad)
    posicion(i) = D * Sin(Angrad)
    X = X0 + 1800 * radio(i)
    Y = Y0 - 1800 * posicion(i)
    picboxcursor.DrawWidth = 5
    picboxcursor.FontSize = 4
    picboxcursor.PSet (X, Y), QBColor(12)
    lblPosVoltaje.Caption = D
```

```
End If
Else
End If
```

```
Next i
```

```

Loop
Close #1
End Sub

```

```

Private Sub picboxi_Click() ' permite realizar lecturas de voltaje en forma ascendente
Dim X0, Y0
intso = intcontador
intb1 = 5000 / (intzoomout - intzoomin + 1) '
j = 0
  If posangular > intzoomin Then
    posangular = posangular - 1
  End If
  lblposangular.Caption = posangular * 360 / intso
Open stropenfilename1 For Input As #1
Do While Not EOF(1)
  For i = 0 To intso - 1 'Step 1
    Line Input #1, g1
    If IsNull(g1) Then
      g1 = "0"
    End If
    D = Val(g1)
    D = D / intnormalizar
    intaux = j
    j = j + intb1
    If posangular = intzoomout Then GoTo siga
    If i = posangular + 1 Then
      If intcoordenadas = 0 Then
        X0 = 600
        Y0 = 3500
        picboxcursor.DrawWidth = 5
        picboxcursor.FontSize = 4
        picboxcursor.PSet ((X0 + intaux - (intzoomin * intb1) + intb1), (Y0 - D *
          2800)), &HFFFFFF
      Else
        X0 = 3000
        Y0 = 2000
        Angrad = i * Pi * 2 / intso 'i*pi/180
        radio(i) = D * Cos(Angrad)
        posicion(i) = D * Sin(Angrad)
        X = X0 + 1800 * radio(i)
        Y = Y0 - 1800 * posicion(i)
        picboxcursor.DrawWidth = 2
        picboxcursor.FontSize = 2
        picboxcursor.PSet (X, Y), &HFFFFFF
      End If
    End If
  End If
  If i = posangular + 1 Then
    If intcoordenadas = 0 Then
      X0 = 600
      Y0 = 3500
      picboxcursor.DrawWidth = 2
      picboxcursor.FontSize = 2
      picboxcursor.PSet ((X0 + intaux - (intzoomin * intb1) + intb1), (Y0 - D *
        2800)), QBColor(0)
    Else

```

```

        X0 = 3000
        Y0 = 2000
        Angrad = i * Pi * 2 / intso 'i*pi/ 180
        radio(i) = D * Cos(Angrad)
        posicion(i) = D * Sin(Angrad)
        X = X0 + 1800 * radio(i)
        Y = Y0 - 1800 * posicion(i)
        picboxcursor.DrawWidth = 2
        picboxcursor.FontSize = 2
        picboxcursor.PSet (X, Y), QBColor(0)
    End If
End If
siga:
    If i = posangular Then
        If intcoordenadas = 0 Then
            X0 = 600
            Y0 = 3500
            picboxcursor.DrawWidth = 5
            picboxcursor.FontSize = 4
            picboxcursor.PSet ((X0 + intaux - (intzoomin * intb1) + intb1), (Y0 - D * 2800)),
                QBColor(12)
            lblPosVoltaje.Caption = D
        Else
            X0 = 3000
            Y0 = 2000
            Angrad = i * Pi * 2 / intso 'i*pi / 180
            radio(i) = D * Cos(Angrad)
            posicion(i) = D * Sin(Angrad)
            X = X0 + 1800 * radio(i)
            Y = Y0 - 1800 * posicion(i)
            picboxcursor.DrawWidth = 2
            picboxcursor.FontSize = 2
            picboxcursor.PSet (X, Y), QBColor(12)
            lblPosVoltaje.Caption = D
        End If
    Else
    End If
Next i
Loop
Close #1
End Sub
Private Sub lblposangular_Click()
    lblposangular.AutoSize = True
End Sub

```

D.12. Código de programa de los módulos

MODULO-1

```

Option Explicit
Type FormState
    Deleted As Integer
    Dirty As Integer
    Color As Integer

```

```

End Type
Global datos(0 To 7) As Double
Global intso, intzoomin, intzoomout As Integer
Global posgraf, intgiroinicio, intpaso, intgirofin As Integer
Global strbandera3, intcom, intabrir As Integer
Global intrnormalizar
Global strh, d1 As String
Public FState() As FormState
Public Document() As New frmArchivo
Public grafico() As New frmgrafica
Public Const ThisApp = "MDINote"
Public Const ThisKey = "Recent Files"
Public Function transforma(A As Integer) As Double
Dim base2(0 To 7), base3(0 To 7) As Integer
Dim i, j As Integer
i = 3
j = 0
While A >= 2
    i = A Mod 2
    A = Int(A / 2)
    base3(j) = i
    j = j + 1
    If A < 2 Then
        base3(j) = A
    End If
Wend
If j < 7 Then
    j = j + 1
    base3(j) = 0
End If
transforma = 0
i = 0
While i < 8
    transforma = transforma + (base3(i) * datos(i))
    i = i + 1
Wend
End Function

```

Módulo-2

```

Option Explicit
Global stropenfilename1 As String
Sub AbrirArchivo()
    Dim intRetVal
    On Error Resume Next
    Dim strOpenFileName As String
    frmMDI.COMDialog1.FileName = ""
    frmMDI.COMDialog1.ShowOpen
    If Err <> 32755 Then ' El usuario eligió Cancelar.
        strOpenFileName = frmMDI.COMDialog1.FileName
        ' Si el archivo es mayor de 65K, no puede
        ' ser abierto, de modo que cancela la operación.
        If FileLen(strOpenFileName) > 65000 Then
            MsgBox "El archivo es muy grande para abrirlo."
        End If
    End If
Exit Sub

```

```

        End If
        OpenFile (strOpenFileName)
    End If
End Sub

Function GetFileName(FileName As Variant)
    On Error Resume Next
    frmMDI.COMDialog1.FileName = FileName
    frmMDI.COMDialog1.ShowSave
    If Err <> 32755 Then ' El usuario eligió Cancelar.
        GetFileName = frmMDI.COMDialog1.FileName
    Else
        GetFileName = ""
    End If
End Function

Function OnRecentFilesList(FileName) As Integer
    Dim i ' Variable contador.
    For i = 1 To 4
        If frmMDI.mnuRecentFile(i).Caption = FileName Then
            OnRecentFilesList = True
            Exit Function
        End If
    Next i
    OnRecentFilesList = False
End Function

Sub OpenFile(FileName)
    Dim flIndex As Integer
    On Error Resume Next
    ' Abre el archivo seleccionado.
    Open FileName For Input As #1
    If Err Then
        MsgBox "Imposible abrir el archivo: " + FileName
        Exit Sub
    End If
    ' Cambia el puntero del mouse a reloj de arena.
    Screen.MousePointer = 11
    ' Modifica el título del formulario y lo presenta.
    flIndex = FindFreeIndex()
    Document(flIndex).Tag = flIndex
    Document(flIndex).Caption = UCase(FileName)
    Document(flIndex).Text1.Text = Input(LOF(1), 1)
    FState(flIndex).Dirty = False
    Document(flIndex).Show
    Close #1
    ' Restablece el puntero del mouse.
    Screen.MousePointer = 0
End Sub

Sub AbrirArchivolImagen()
    Dim intRetVal
    On Error Resume Next
    frmMDI.COMDialog1.FileName = ""
    frmMDI.COMDialog1.ShowOpen

```

```
If Err <> 32755 Then ' El usuario eligió Cancelar.
    stropenfilename1 = frmMDI.CMDialog1.FileName
    ' Si el archivo es mayor de 65K, no puede
    ' ser abierto, de modo que cancela la operación.
    If FileLen(stropenfilename1) > 65000 Then
        MsgBox "El archivo es muy grande para abrirlo."
        Exit Sub
    End If
frmgrafica.Show
End If
End Sub
```