

87
2es.



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

MÓDULO DE OSCILOSCOPIO CON PUNTA DE PRUEBA INALÁMBRICA

T E S I S

QUE PARA OBTENER EL TÍTULO DE:

INGENIERO MECÁNICO ELECTRICISTA
(ÁREA ELÉCTRICA ELECTRÓNICA)

P R E S E N T A
DE JESÚS TOLENTINO MANUEL ANTONIO

DIRECTOR: M. I. ROBERTO TOVAR MEDINA

CIUDAD DE MÉXICO, D.F.

1998

TESIS CON
FALLA DE ORIGEN

260168





Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas

Tesis Digitales

Restricciones de uso

DERECHOS RESERVADOS ©

PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

A mis padres:

Mis agradecimientos por todo su gran apoyo que siempre me brindaron para impulsarme a seguir siempre adelante aún en los momentos difíciles. Por darme la mejor de las herencias para la vida, ¡Gracias!.

A mis hermanos:

Agradezco todo su apoyo que tuve a lo largo de toda la carrera y en todo momento ese respaldo moral que nunca estuvo de más.

A mis amigos:

Doy mis agradecimientos por su apoyo incondicional que tuve durante mis estudios en la Universidad, así como en la elaboración de la tesis.

A mi asesor de tesis:

Agradezco su apoyo y adecuado asesoramiento, así como su paciencia en todo el desarrollo de la tesis.

Manuel Antonio de Jesús Tolentino

A mis Padres:

Les agradezco su apoyo incondicional tanto moral como económicamente. A mi Papá, por darme ese ejemplo de triunfador y por darme aquel gran consejo tan decisivo en mi vida. A mi Mamá, por su cuidado y atención cuando empezaba mis estudios.

A mis Hermanos:

Les agradezco por aquella competencia sana de ser buenos estudiantes. A Adriana por su ejemplo y ayuda cuando éramos niños. A Daniel por ser mi hermano inseparable de niños, compartiendo grandes momentos de juego.

A mis Tías:

Les agradezco por su comprensión y ayuda. A mi tía Irene, por creer y confiar en mí, para terminar mi carrera. A Hortensia por ayudarme con los trabajos y dibujos que me pedía en la Primaria y Secundaria.

A mi Director de Tesis:

Le agradezco por su asesoría y paciencia al ayudarnos a realizar este trabajo. Ing. Roberto Tovar, por su ánimo para que este trabajo no solamente sea una Tesis que se quede en la Universidad, sino que se conozca en otras partes.

A mi Universidad:

Le agradezco el haberme aceptado como parte de ella durante ocho años, desde el bachillerato. Por darme esas becas, que aproveché al máximo.

Gracias a Todos Ustedes
Julio Arturo Hernández Peña

A Elvia y J. Elías:

Por su cariño, comprensión, confianza, apoyo, sacrificios, consejos, ejemplo y por todos esos momentos de felicidad que siempre han procurado que tenga la familia.

Gracias papás

A Sara, Areli, Jael y Dámaris:

Por su comprensión, paciencia, simpatía y por todos esos momentos únicos que hemos vivido juntas.

Gracias hermanitas

A mis familiares:

Por todo el apoyo que he recibido de todos ustedes, no puedo mencionar nombres por que son muchos, pero ustedes saben quienes son.

Gracias a todos

A la Universidad Nacional Autónoma de México:

Por darme la oportunidad de estudiar una carrera profesional, por mi formación y por todo lo que he recibido de ella.

Gracias UNAM

A mis Profesores:

Por que de cada uno de ustedes aprendí valores diferentes.

Gracias profesores

Al Ing. Roberto Tovar Medina:

Por su asesoría, consejos, orientación y confianza en nosotros.

Gracias Ingeniero

La realización de este trabajo la he logrado con la ayuda de todos ustedes, por que directa o indirectamente me hicieron sentir motivada a seguir adelante hasta que por fin lo logré. A todos ustedes les dedico este trabajo, especialmente a mis padres, por que a ellos les debo mi felicidad, mis valores y mis logros.

Gracias

Lydia Noguerón Galicia

Índice

Objetivos	XIII
Capítulo 1: Introducción	1
Capítulo 2: Osciloscopio	7
2.1 Componentes de un Osciloscopio	8
2.1.1-Tubo de Rayos Catódicos	8
2.1.2 Graticula	12
2.1.3 Amplificadores Vertical y Horizontal	13
2.1.3.1 Amplificador Vertical	13
2.1.3.2 Amplificador Horizontal	15
2.1.4 Circuito de la Base de Tiempo	16
2.1.5 Generador de Barrido	16
2.1.6 Circuito de Disparo	17
2.1.7 Selector de Entrada	19
2.1.8 Atenuador de Entrada	19
2.2 Señales de Entrada al Osciloscopio	21
2.3 Modo de Despliegue de una Señal en un Osciloscopio	24
2.4 Ancho de Banda de un Osciloscopio	25

Módulo de Osciloscopio

2.5 Controles Básicos del Osciloscopio	26
2.6 Osciloscopio con Memoria	30
2.7 Osciloscopio de Muestreo	31
2.8 Osciloscopios de Almacenamiento Digital	32

Capítulo 3: Señales Eléctricas

3.1 Señales de Corriente Continua	33
3.2 Señales de Corriente Alterná	34
3.2.1 Señales Analógicas	35
3.2.2 Señales Digitales	35
3.3 Frecuencia y Fase de las Ondas Senoidales	36
3.4 Valor Promedio y Valor Cuadrático Medio	39
3.4.1 Valor Promedio	39
3.4.2 Valor Cuadrático Medio	41

Capítulo 4: Convertidores de Señales

4.1 Introducción	45
4.2 Proceso de Muestreo	48
4.3 Convertidores Analógico-Digitales	52
4.4 Convertidores Digital-Analógico	58
4.5 Circuitos de Captura y Mantenimiento	61
4.6 Multiplexores Analógicos	62
4.7 Interface entre Convertidores y Microcomputadoras	64

4.8 Realización de Conversiones por Programa	67
Capítulo 5: Transmisión y Recepción de Información	
5.1 Transmisión de Datos en Serie y en Paralelo	70
5.2 Modos de Comunicación	72
5.2.1 Modo Simplex	72
5.2.2 Modo Dúplex	72
5.2.3 Modo Semi-Dúplex	73
5.3 Modos de Transmisión	73
5.3.1 Transmisión Asíncrona	74
5.3.2 Transmisión Síncrona	74
5.4 Tasas de Transmisión: Bits/seg y Bauds/seg	75
5.4.1 Tasa de Transmisión de Bits/Seg	75
5.4.2 Tasa de Transmisión de Bauds/Seg	76
5.5 Medios de Comunicación	76
5.5.1 Líneas de Dos Alambres	77
5.5.2 Líneas de Par Trenzado	78
5.5.3 Cable Coaxial	79
5.5.4 Fibra Óptica	80
5.5.5 Luz Infrarroja	82
5.5.6 Radio	82
5.5.7 Microondas	83
5.5.8 Satelital	83
5.6 Efectos Adversos en la Transmisión	86

Módulo de Osciloscopio

5.6.1 Atenuación	86
5.6.2 Ancho de Banda Limitado	86
5.6.3 Distorsión	87
5.6.4 Ruido	88
5.6.5 Interferencia	89
5.7 Modulación	90
5.7.1 Tipos de Modulación	90
5.7.1.1 Modulación en Amplitud	91
5.7.1.2 Modulación en Frecuencia	93
5.7.1.3 Modulación en Fase	94
5.7.2 Demodulación	95
5.7.3 Factor de Modulación y Porcentaje de Modulación	96
5.7.4 Sobremodulación	97
5.7.5 Índice de Modulación	98
5.8 Filtros	99
Capítulo 6: Detección y Corrección de Errores en los Datos	101
6.1 Introducción	101
6.2 Conceptos Básicos de Codificación y Decodificación	104
6.2.1 Modelo General de un Sistema con Control de Errores	104
6.2.2 Códigos de Bloque	104
6.2.3 Códigos de Árbol	107
6.2.4 Códigos Convolutivos	107
6.2.5 Distancia de Hamming	107

6.2.6 Errores Aleatorios y Ráfaga	108
6.3 Códigos Lineales	109
6.3.1 Definición y Codificación	109
6.3.2 Detección y Corrección de Errores	113
6.4 Códigos de Bloque	116
6.4.1 Códigos Cíclicos	116
6.4.1.1 Definición y Codificación	116
6.4.1.2 Decodificación de Códigos	120
6.4.2 Códigos BCH	121
6.4.3 Códigos Reed-Solomon	122
6.5 Códigos BCH	123
6.5.1 Conceptos Básicos	124
6.5.1.1 Campos Finitos	124
6.5.1.2 Polinomio Primitivo	126
6.5.2 Diseño de Códigos BCH	131
6.5.3 Detección y Corrección de Errores en Códigos BCH	133
6.5.3.1 Cálculo del Síndrome	133
6.5.3.2 Algoritmo Euclideo	134
6.5.3.3 Localización de la Posición del Error	135
Capítulo 7: Microcontrolador 68HC11	139
7.1 Modos de Operación	141
7.1.1 Modo Single Chip	141
7.1.2 Modo Expanded Nonmultiplexed	141

Módulo de Osciloscopio

7.1.3 Modo Special Bootstrap	142
7.1.4 Modo Special Test	142
7.2 Pines del MCU68HC11F1	143
7.3 Mapa de Registros del MCU68HC11F1	145
7.3.1 Registros de Control y Estatus	146
7.3.2 Registro de Mapeo I/O (INIT)	147
7.4 Puertos de Entrada/Salida	147
7.4.1 Puerto A	148
7.4.1.1 Registro de Datos del Puerto A (PORTA)	148
7.4.1.2 Registro de Dirección de Datos del Puerto A (DDRA)	149
7.4.2 Puertos B y F	150
7.4.3 Puertos C y G	150
7.4.4 Puerto D	150
7.4.5 Puerto E	151
7.5 Convertidor Analógico a Digital	151
7.5.1 Asignación de Canal	152
7.5.1.1 Operación de Un Canal	153
7.5.1.2 Operación de Múltiples Canales	154
7.5.2 Registro de Control/Estatus del A/D	154
7.5.3 Registros de Resultados del A/D	155
7.5.4 Inicio de la Conversión A/D y Selección del Reloj	156
7.6 Temporizador Programable	157
7.6.1 Función de Captura de Entrada	157
7.6.2 Función de Comparación de Salida	159

7.6.2.1 Registro para Forzar la Comparación	160
7.6.2.2 Registro 1 de la Máscara de comparación de Salida	161
7.6.2.3 Registro 1 de Datos de Comparación de Salida	161
7.6.3 Registro de Conteo del Temporizador	162
7.6.3.1 Registro 1 de Control del Temporizador	162
7.6.3.2 Registro 1 de la Máscara de Interrupción del Temporizador	163
7.6.3.3 Registro 1 de la Bandera de Interrupción del Temporizador	163
7.6.3.4 Registro 2 de la Máscara de Interrupción del Temporizador	164
7.6.3.5 Registro 2 de la Bandera de Interrupción del Temporizador	166
7.6.4 Acumulador de Pulsos	167
7.6.4.1 Registro de Control del Acumulador de Pulsos	168
7.6.4.2 Registro de Conteo del Acumulador de Pulsos	169
7.7 Interface de Comunicaciones Serial	170
7.7.1 Formato de los Datos	170
7.7.2 Operación de la Transmisión	171
7.7.3 Registros SCI	171
7.7.3.1 Registro Serial de Comunicación de Datos	171
7.7.3.2 Registro de Control de Comunicaciones Serial (SCCR1)	172
7.7.3.3 Registro 2 de Control de Comunicaciones Serial (SCCR2)	173
7.7.3.4 Registro de Estatus de Comunicaciones Serial (SCSR)	175
7.7.3.5 Registro de Tasa de Baudaje (BAUD)	177
Capítulo 8: Generalidades del Lenguaje de Programación Visual Basic	179
8.1 Historia	180

Módulo de Osciloscopio

8.2 Creación de Aplicaciones con Visual Basic	182
8.2.1 Proyectos	183
8.2.2 Formas	183
8.2.3 Caja de Herramientas	185
8.2.4 Propiedades	189
8.2.5 Programación	193
8.2.6 Menús	195
8.2.6.1 Ventana Menu Design	195
Capítulo 9: Desarrollo del Módulo de Osciloscopio	199
9.1 Introducción	199
9.2 Captura, Procesamiento y Transmisión de la Señal	202
9.2.1 Especificaciones de la Información a Transmitir	203
9.2.1.1 Byte de Dato (\$0100)	205
9.2.1.2 Byte de Información (\$0101)	205
9.2.1.3 Estado del MCU68HC11F1 mientras Recibe una Tarea	206
9.2.2 Teclado de la Punta Inalámbrica	208
9.2.3 Sistema de Acondicionamiento de la Señal	213
9.2.3.1 Detección de Valores Negativos y Generación del Bit de Signo	214
9.2.3.2 Detección del Rango de Atenuación/Amplificación	216
9.2.3.3 Atenuación/Amplificación de la Señal	221
9.2.3.4 Conversión de las Señales Negativas a Señales Positivas	224
9.2.3.5 Generación de los Bits para Indicar la Atenuación	225
9.2.3.6 Otras Consideraciones del Circuito de Acondicionamiento	226

9.2.4	Captura de una Señal Eléctrica	228
9.2.4.1	Conversión A/D de una Señal Eléctrica	228
9.2.5	Codificación de la Información a Transmitir	231
9.2.5.1	Diseño del Código	231
9.2.5.2	Rutina de Codificación	236
9.2.6	Sistema de Transmisión	241
9.2.6.1	Modulación de la Información	242
9.2.6.2	Transmisión de la Información	244
9.2.6.3	Transmisor	247
9.3	Recepción, Procesamiento y Graficación de la Señal	250
9.3.1	Sistema de Recepción	251
9.3.1.1	Receptor	251
9.3.1.2	Recepción de la Información	253
9.3.2	Sistema de Detección y Corrección de Errores	258
9.3.3	Control del Programa "Módulo de Osciloscopio"	274
9.3.3.1	Programa "Módulo de Osciloscopio"	275
9.3.3.2	Control Remoto del Programa	279
9.3.4	Graficación de la Señal	310
Capítulo 10:	Resultados	313
Capítulo 11:	Conclusiones	319
Apéndice A:	Puertos de la Computadora Personal	321
A.1	Puerto Paralelo	321

Módulo de Osciloscopio

A.2 Puerto Serie	325

Apéndice B: Simuladores Detectores y Correctores de Errores **331**

B.1 Simulador GF(2⁵)	332

B.1.1 Diseño del código BCH para el Simulador GF(2 ⁵)	332

B.1.2 Desarrollo del Simulador GF(2 ⁵)	333

B.1.2.1 Codificador	334

B.1.2.2 Detector / Corrector de Errores	336

B.1.2.3 Decodificación	339

B.1.3 Funcionamiento del Simulador	340

B.2 Simulador GF(2⁶)	346

B.2.1 Desarrollo del Simulador	347

B.2.1.1 Función Btndfcn	347

B.2.1.2 Procedimiento que Lee la Información y la Codifica	347

B.2.1.3. Procedimiento que Transmite la Información	349

B.2.1.4 Procedimiento que Recibe la Palabra Codificada	349

B.2.1.5 Función Btndcers	350

B.2.2 Funcionamiento del Programa	351

Apéndice C: Códigos del Módulo de Osciloscopio **359**

C.1 Código Fuente del Programa del MCU68HC11	359

C.2 Código del Programa "Módulo de Osciloscopio"	364

Apéndice D: Hojas de Especificación **371**

D.1 Hojas de Especificación de los ILEDs	371
D.1.1 Máximos Rangos Absolutos	371
D.1.2 Características Electroópticas	372
D.1.3 Otras Características	372
D.2 Especificaciones del Receptor/Demodulador GP1U52X	373
D.2.1 Características del GP1U52X	374
D.2.2 Especificaciones Eléctricas del GP1U52X	374
D.3 Especificaciones Eléctricas del MAX232CPE	375
D.4 Especificaciones Eléctricas de los Transistores 2N2222A	376
D.5 Especificaciones de los Multiplexores TC4052 y TC4053	376
D.5.1 Características del TC4052 y TC4053	378
D.5.2 Máximos Rangos Absolutos	378
D.5.3 Características Eléctricas	378
D.6 Especificaciones de los Operacionales CA3130 y CA3140	379
D.6.1 Características del CA3130 y CA3140	380
D.6.2 Características Eléctricas	381
D.6.3 Máximos Rangos Absolutos	381
D.7 Especificaciones del Amplificador Operacional TL074	381
D.7.1 Características del TL074	382
D.7.2 Rangos Máximos	382
D.7.3 Características Eléctricas	383
D.8 Especificaciones de Contador HCF4516	384
D.8.1 Características del HCF4516	384

Módulo de Osciloscopio

D.8.2 Características Eléctricas	385
D.9 Especificaciones de los CI 74HC08, 74HC32 y 74HC04	385
D.9.1 Características del 74HC08, 74HC32 y 74HC04	386
D.9.2 Características Eléctricas	386
D.10 Especificaciones Eléctricas del MCU68HC11F1	387
D.10.1 Máximos Rangos	387
D.10.2 Características Eléctricas DC	388
D.10.3 Control de Tiempos	389
D.10.4 Tiempos de los Puertos Periféricos	390
D.10.5 Características del Convertidor Analógico/Digital	390
D.10.6 Tiempos del Bus	392
D.10.7 Direcciones de Vectores	393
D.10.8 Nombres de los Pines	394

Apéndice E: Circuitos Eléctricos

E.1 Circuito del Acondicionador de la Señal Eléctrica	395
E.2 Circuito Eléctrico del Transmisor	397
E.3 Circuito Eléctrico del Receptor	398
E.4 Parámetros del Módulo de Osciloscopio	400

Bibliografía

	401

Objetivos

Los objetivos a desarrollar en el presente trabajo son:

- ♦ Diseñar e implementar una punta inalámbrica, para lograr la transmisión digital de una señal eléctrica.
- ♦ Desarrollar un circuito de adquisición y acondicionamiento para una señal eléctrica, como parte de la punta inalámbrica.
- ♦ Diseñar e implementar un subsistema de autorrango para el circuito de acondicionamiento de la señal eléctrica adquirida.
- ♦ Desarrollar e implementar un pequeño teclado para controlar la captura de la señal eléctrica y manipular remotamente un programa graficador de dicha señal.
- ♦ Implementar un transmisor eficiente, para lograr la transmisión inalámbrica.
- ♦ Implementar un sistema capaz de recibir eficientemente la señal digital transmitida, de tal forma que una PC la pueda interpretar.
- ♦ Elaborar un programa en un lenguaje de Microsoft Windows para graficar la forma de onda de la señal digital transmitida, como una interface gráfica fácil de usar y agradable para el usuario.

Capítulo 1:

Introducción

En muchas ocasiones, las personas requieren de hacer mediciones de eventos o cosas, como por ejemplo el tiempo en que tarda una persona en llegar a un destino, la distancia que se recorre entre dos puntos, la velocidad que puede alcanzar un avión, el peso de un objeto, etc.. Durante mucho tiempo las personas se han abocado a hacer instrumentos que midan directamente esos eventos o cosas, como por ejemplo el reloj, la cinta métrica, la báscula, etc.. Pero no nada más se han inventado instrumentos sino que también se ha establecido un Sistema Internacional de Unidades, el cual define un estándar en las unidades de medición, teniendo como finalidad mantener una coherencia en las mediciones. Como por ejemplo utilizar los metros con lo kilogramo y segundos.

Con el uso de la electricidad y por ende la utilización de aparatos que funcionan con ella, hubo la necesidad de crear nuevos instrumentos que ayudarán a medir corrientes y voltajes de dichos aparatos. Dos de los aparatos más importantes para medir estos fenómenos eléctricos son el multímetro y el osciloscopio. El primero como su nombre lo indica se utiliza para medir muchas variables como voltajes, corrientes, resistencias, y los más modernos también miden frecuencia, temperatura, capacitancias e inductancias. Este dispositivo tiene una pantalla en donde muestra el

resultado de las mediciones. También cuenta con escalas, para poder medir una gran variedad de magnitudes, sin embargo aunque este dispositivo es muy útil en el análisis y diseño de sistemas eléctricos, no es suficiente contar con él, porque a veces se requiere ver las formas de onda de las señales. El osciloscopio se utiliza precisamente para ver la forma de onda de las señales por medio de una pantalla de rayos catódicos, que permite visualizar formas de ondas de las señales eléctricas de los sistemas, es decir nos permite ver la magnitud del voltaje en un determinado instante de tiempo. En muchas ocasiones se requiere contar con esta información para saber si nuestro sistema está funcionando correctamente y poder determinar en donde se encuentran fallas eléctricas, o qué dispositivo o subsistema se necesita adaptarle para que funcione correctamente.

La tecnología de los osciloscopios ha progresado, ya que por ejemplo, antes contaban con muchas perillas que permitían hacer los ajustes convenientes para poder observar las ondas eléctricas. En los nuevos y modernos osciloscopios, estos ajustes (o por lo menos la mayoría de ellos) los hacen automáticamente, dejando la opción de que el usuario pueda modificarlos, pero ya no con perillas, sino con botones, que permiten la entrada de funciones digitales, las cuales dan una mayor precisión en las medidas, además de que los usuarios ya no tienen que hacer las mediciones a simple vista, sino que los osciloscopios entregan las mediciones que el usuario desea numéricamente. Esto es gracias a que los nuevos osciloscopios cuentan con su propio microprocesador para procesar las señales leídas y proporcionan gran cantidad de información aparte de la gráfica. Además a muchos de ellos ya se les puede conectar una impresora o ellos mismos cuentan con su propia impresora integrada. También hay módulos de osciloscopios que ya interactúan con la computadora, pero esto es a través

de un puerto paralelo. Es indudable que estos avances tecnológicos tienen repercusiones en el costo de dichos aparatos, ya que en cuanto más completo está un osciloscopio, éste se vuelve más caro para adquirirlo.

Considerando lo anterior surge la idea de hacer un Módulo de Osciloscopio, no para hacer un aparato tecnológicamente poderoso, sino para hacer un sistema práctico y barato para que lo usaran estudiantes y profesores, ya que durante las carreras de Ingeniería en Electrónica, Telecomunicaciones o Computación un estudiante requiere de utilizar un osciloscopio para realizar algunas de sus prácticas o proyectos escolares y en muchas ocasiones le es muy difícil hacer mediciones en la escuela, porque en ocasiones no se tiene el tiempo necesario para realizar cómodamente mediciones o porque cuando se aproxima una fecha de revisión de proyectos, por lo general, los osciloscopios son insuficientes. Otro inconveniente es que en la mayoría de las prácticas se pide entregar las gráficas observadas en el osciloscopio y es tedioso, incómodo e impreciso estar copiando las gráficas manualmente, puesto que en la mayoría de los casos los osciloscopios no cuentan con un módulo de impresión.

Nosotros sabíamos que el hacer un osciloscopio no era lo más apropiado, ya que hay muchos osciloscopios en el mercado y el tratar de hacer uno iba a ser costoso y al final no obtendríamos nada innovador. Al enfrentarnos a esta problemática surgió la idea de utilizar la computadora como herramienta para el despliegue de la gráfica de la señal.

No sería muy conveniente tener la computadora al lado de los aparatos o dispositivos cuyas señales eléctricas iban a ser medidas, ya que por lo general al tener una computadora al lado de dichos dispositivos, ésta les podría generar una señal de ruido o ser muy estorbosa para el espacio donde generalmente se hacen las mediciones. Nuevamente nos enfrentamos a otro inconveniente, fue entonces cuando surgió la idea de tener alejada la computadora con respecto a la mesa de mediciones.

Por lo anterior se determinó que nuestro osciloscopio iba a consistir de dos partes una parte donde se capturaría la señal eléctrica a medir y otra donde se desplegaría dicha señal. Aquí observamos que la comunicación entre ambas partes podría ser de forma alámbrica o inalámbricamente, por lo que se decidió que lo más apropiado era tener una comunicación inalámbrica, ya que por lo general una conexión alámbrica a veces no es muy práctica, ya que el cable que existe de por medio no iba a estar fijo, sino que se tendría que estar arrastrando cada vez que se hiciera una medición de una señal eléctrica, además de que sería muy estorboso y probablemente causaría molestias con otras personas.

Una de las ventajas que vimos en el hacer el módulo de osciloscopio con una computadora fue de que los alumnos e investigadores por lo general tienen acceso a una computadora personal. Cabe recordar que en los últimos años la computadora se ha convertido en una herramienta indispensable en la formación profesional de una persona por el sin número de tareas que puede realizar.

Al emprender este proyecto se buscaba darle aplicación a conocimientos aprendidos en la escuela, como por ejemplo el manejo de convertidores analógicos a digitales, microcontroladores, amplificadores operacionales utilizados como comparadores, circuitos de lógica TTL y switches analógicos por el aspecto de hardware y conocimientos de programación por el aspecto de software.

Una de las ideas que siempre se tuvo al hacer este proyecto fue la de tratar de hacer un módulo que fuera fácil de manejar, entender y adquirir por estudiantes, profesores o investigadores, es decir que pensamos en hacer un módulo práctico y sobre todo barato.

Como la manera de transmitir la información de la señal eléctrica a medir iba a ser de forma inalámbrica debíamos poner énfasis en la integridad de los datos recibidos. Para ello se tenía que implementar un esquema de detección y corrección de errores de los datos.

Para la adquisición de la señal se decidió implementar una red atenuadora, en la cual se propuso que la selección del escalamiento se realizaría en forma automática para obtener un valor escalado de la señal lo más apropiado posible. Para obtener el escalamiento automático teníamos dos opciones: una era el realizarlo por medio de software a través del microcontrolador, el cual también se usaría para realizar la conversión digital-analógica y codificación de la señal, y la otra opción era el realizar la selección automática por medio de hardware, la cual se eligió debido a que el

realizarlo por medio del microcontrolador haría más lento el proceso de conversión y codificación de la señal adquirida.

Por otra parte ya que la adquisición de la señal sería por medio de una punta inalámbrica se decidió implementar un circuito de adquisición y acondicionamiento de la señal sencillo para evitar tener una punta de tamaño muy grande y por consiguiente difícil de manejar.

Uno de los puntos que siempre tendríamos en la mente es que a mayor eficiencia de aplicación es más caro el desarrollo de un prototipo. Esto era un gran dilema en la planeación del presente trabajo ya que nosotros no debíamos alejarnos de la idea de que el sistema debería ser barato y muy accesible para las personas que lo fueran a utilizar. Tampoco podíamos desarrollar un sistema en el que no garantizáramos seguridad en la información, por querer hacer algo sencillo y barato. Por otra parte nosotros pensamos en que este sistema lo podrían construir muchas de las personas que cuentan con un microcontrolador, un convertidor analógico digital, una PC y sobre todo un lenguaje de programación que permita hacer gráficas.

Actualmente la mayoría de las aplicaciones corren bajo un ambiente de interface gráfica; en el caso de las Computadoras Personales en Microsoft Windows. Tal vez a muchas personas les desagrada el uso y manejo de Microsoft Windows pero hay que ser realistas, es lo más fácil de encontrar ya sea en una escuela, en una casa o en un laboratorio.

Capítulo 2: Osciloscopio

El Osciloscopio es un instrumento muy versátil y útil que se emplea para la medición y análisis de formas de onda y otros fenómenos en los circuitos electrónicos. Con el Osciloscopio se pueden hacer mediciones de voltajes AC o DC y periodos de señales, además se pueden hacer algunas mediciones indirectas de corrientes de AC o DC, frecuencias, relaciones de fase, y características de señales eléctricas como, tiempos de subida, tiempos de bajada, sonidos, etc.. El osciloscopio es básicamente un graficador X-Y muy rápido que despliega una señal contra otra o contra el tiempo. También el Osciloscopio puede dar una representación visual de muchos fenómenos dinámicos por medio de transductores que convierten corriente, presión, esfuerzo temperatura, aceleración y otras cantidades físicas a voltajes.

El eje X (eje horizontal) es un voltaje generado internamente en forma de una rampa lineal en función del tiempo, el cual mueve el punto luminoso periódicamente de izquierda a derecha sobre la pantalla. El voltaje a examinar se aplica en el eje Y (entrada vertical) y mueve el punto hacia arriba y hacia abajo de acuerdo con el valor instantáneo del voltaje de entrada. El punto moviéndose a gran velocidad, traza sobre la pantalla una gráfica que muestra las variaciones de un voltaje de entrada en función del tiempo.

2.1 Componentes de un Osciloscopio

Existen gran variedad de Osciloscopios en el mercado, pero los más sencillos nunca han cambiado en su constitución. A continuación se describirán las partes fundamentales de un Osciloscopio Básico, sin considerar que en la actualidad existen unos modernos Osciloscopios que por ejemplo son controlados por medio de un microprocesador, no siendo éste esencial para todos los Osciloscopios. El Osciloscopio tradicional se compone de 6 módulos, los cuales son:

- ◆ Tubo de Rayos Catódicos
- ◆ Grátícula
- ◆ Amplificadores Vertical y Horizontal
- ◆ Circuito de la Base de Tiempo
- ◆ Generador de Barrido
- ◆ Circuito de Disparo (Trigger)
- ◆ Selector de Entrada
- ◆ Atenuador de Entrada

2.1.1 Tubo de Rayos Catódicos

El Tubo de Rayos Catódicos (TRC) es como un envase de vidrio sellado herméticamente, con un cañón electrónico y un sistema de deflexión montados dentro de uno de los extremos del envase y por el otro extremo una pantalla fluorescente. En la Figura 2.1 se muestra el TRC con algunos de sus componentes. El envase se deja al alto vacío, para que un fino haz de electrones producido dentro del tubo se esparza por la colisión con moléculas de gas que se encuentre en su camino. Es por esto que el TRC produce un haz de electrones enfocado agudamente y acelerado a

gran velocidad. Este haz de electrones enfocado y acelerado viaja desde su fuente (el cañón electrónico), hasta el otro extremo del TRC, donde golpea el material fluorescente depositado en la cara interior del TRC (la pantalla) con la suficiente energía para producir un pequeño punto luminoso sobre ella. Mientras viaja desde su fuente hasta la pantalla, el haz electrónico pasa entre un conjunto de placas de deflexión horizontales y verticales.

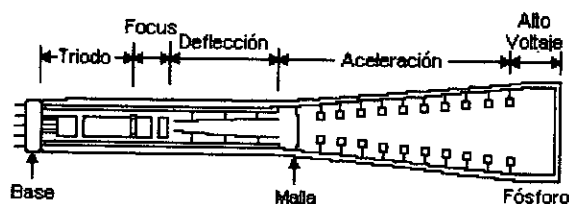


Figura 2.1: Tubo de Rayos Catódicos

La función del cañón de electrones es producir un haz de electrones. El cañón consiste en un cátodo termoiónico (un cátodo hecho de un material que emite electrones cuando se calienta), varios electrodos aceleradores y controles para foco e intensidad. Cuando el cátodo se calienta a alta temperatura, empieza a emitir electrones. Algunos de estos electrones pasan a través de un pequeño orificio en la rejilla de control de intensidad que rodea al cátodo. Si se aplica un voltaje negativo a esta rejilla, solamente pueden pasar a través del hueco un número limitado de electrones. Este número se puede controlar variando la magnitud del voltaje. La intensidad del punto luminoso donde el haz golpea la pantalla depende del número de electrones en el haz.

Los electrones emitidos se comprimen por medio de los ánodos aceleradores y el foco del cañón para formar el haz comprimido. Para lograr esta compresión se utiliza un esquema de enfocamiento electrostático en el TRC del Osciloscopio. El mismo campo electrostático también dirige los electrones a lo largo del eje del haz y los acelera hacia la pantalla fluorescente.

Después de dejar el cañón electrónico, el haz enfocado y acelerado pasa a través de dos conjuntos de placas deflectoras. Si no hay una diferencia de voltaje entre las placas, el haz continúa derecho y golpea la pantalla en su centro. Si existe una diferencia de voltaje entre uno o los dos conjuntos de placas, el haz se desvía de su camino recto. La cantidad de desviación se determina por medio de las magnitudes de las diferencias de potencial. En los Osciloscopios comunes, se debe aplicar una diferencia de potencial entre 20 y 50 Volts más para desviar el punto un centímetro.

Los dos conjuntos de placas deflectoras se colocan perpendicularmente uno del otro para que puedan controlar independientemente el haz tanto en la dirección horizontal como vertical. Por ejemplo, el voltaje aplicado a las placas deflectoras verticales cambia la dirección del haz de electrones únicamente en la dirección vertical. El haz puede desviar hacia arriba o hacia abajo dependiendo de la polaridad de estas placas. Si el voltaje de la placa superior se hace positivo con respecto al de la placa inferior, el haz cargado negativamente será atraído por la placa superior y desviado hacia arriba. De forma similar, un voltaje aplicado a las placas horizontales desviará el haz hacia la izquierda o la derecha.

La Pantalla Fluorescente del TRC esta cubierta con fósforo. El material más usado es la Willemita (orto-silicato de zinc), que emite la luz verde. En realidad, existen otros materiales usados en el recubrimiento de la pantalla fluorescente, que emiten diferentes colores y de distintas características en su persistencia, como por ejemplo el óxido de zinc, para una luz de color azul; silicato de zinc y berilio para una luz amarilla.

En el punto donde el haz de electrones golpea la pantalla, el fósforo emite una luz visible (un punto luminoso). EL fósforo continúa emitiendo luz por un corto periodo de tiempo, después de que el haz ha dejado de golpearlo. Si el haz de electrones se mueve repetidamente a través de la pantalla a lo largo del mismo camino y si la acción se repite rápidamente la imagen que se forma en al pantalla del Osciloscopio aparecerá como una línea sólida.

La persistencia del foco es la longitud de tiempo que toma la intensidad del punto luminoso para decrecer hasta por un 10% de su brillantes original. El valor de la persistencia varía de acuerdo con el tipo de fósforo. En otras palabras, la persistencia es la duración de la fosforescencia de la imagen la cual varía con respecto al material y a la cantidad de energía del haz electrónico.

Cuando el haz de electrones golpea la pantalla se genera, tanto calor, como temperatura. (80 % de la energía del haz electrónico se convierte en calor y 20 % en luz visible). Por lo general se trata de evitar que el haz queme y produzca huecos en la capa de fósforo, manteniendo la intensidad del haz en un valor pequeño.

Los Osciloscopios cuentan con un enrejado o malla en la pantalla, la cual es un conjunto de líneas verticales y horizontales que se encuentran dibujadas permanentemente sobre una mica de la pantalla del TRC. Esta malla sirve como una escala para la medición de las cantidades mostradas por el TRC. En la mayoría de los Osciloscopios las líneas de la malla se encuentran espaciadas un centímetro.

2.1.2 Grátícula

La Grátícula es una rejilla de líneas que se usa como una escala cuando se hacen mediciones de tiempo y amplitud con un Osciloscopio. La grátícula está dibujada en una mica que se pone por delante de la cara del TRC. El despliegue producido por el rayo de electrón y la grátícula colocados en el mismo plano eliminará inexactitudes de medición, llamados errores paralelos. Los errores paralelos ocurren cuando el trazo y la grátícula están en planos diferentes, con respecto a la posición de los ojos del observador.

Aunque se pueden usar grátículas de diferentes tamaños, generalmente están marcados en un patrón de 10x8 cuadros, es decir, hay ocho divisiones principales verticales y diez divisiones principales horizontales. Las etiquetas en los controles del panel principal de un Osciloscopio, siempre se refieren a las divisiones principales. Las divisiones principales son marcadas en pulgadas o centímetros, siendo los centímetros los más ampliamente usados. Las marcas pequeñas en las líneas centrales de la grátícula representan las divisiones menores o subdivisiones. Además en la grátícula estándar algunas grátículas incluyen marcas para las divisiones del tiempo de subida.

2.1.3 Amplificadores Vertical y Horizontal

Los voltajes aplicados a las entradas de un Osciloscopio son tan pequeños que no son capaces de provocar una deflexión apreciable en el haz de electrones si se aplicaran directamente a las placas deflectoras del TRC. Los voltajes de entrada se introducen primero en amplificadores que incrementan su magnitud.

Los amplificadores verticales y horizontales incrementan las señales eléctricas pequeñas suficientemente para deflexionar el haz electrónico, ya sea vertical u horizontalmente a través del TRC. Normalmente un TRC requiere un voltaje diferencial entre sus placas verticales de 100V si el haz va a ser deflexionado sobre la gráticula del Osciloscopio. Los niveles de entrada que originan tal deflexión pueden ser del orden de los 10mV o menos, por lo que se requiere una ganancia considerablemente grande en los amplificadores.

2.1.3.1 Amplificador Vertical

El amplificador vertical consiste de varias etapas, con ganancia o sensibilidad fijas, normalmente expresada en términos del factor de deflexión (V/div). El amplificador conserva sus características de manipulación de la señal con la selección apropiada del atenuador de entrada. Con el atenuador en su posición más sensible, la ganancia total del amplificador corresponde a la lectura más baja del selector de Volts/div.

El Amplificador Vertical generalmente esta constituido de dos circuitos principales: el preamplificador y el amplificador vertical principal. El preamplificador es la etapa de entrada, la cual consiste a menudo de un seguidor Fet con una impedancia de entrada alta que virtualmente aísla el amplificador del atenuador. El amplificador vertical principal consiste de una etapa controladora de amplificación y una etapa de salida en push-pull que entrega dos señales de voltaje iguales pero de polaridad opuesta para alimentar las placas de deflexión vertical del TRC. Los circuitos de doble efecto casi siempre se usan en los amplificadores verticales y horizontales porque mejoran la linealidad de la deflexión del TRC. El amplificador vertical también incluye un circuito para centrar el haz y amplificadores adicionales para impulsar la línea de retardo vertical.

Cabe mencionar que el amplificador es el principal factor en determinar la sensibilidad y ancho de banda de un Osciloscopio. En general una sensibilidad grande, la cual se expresa en términos de voltaje por centímetro de la deflexión vertical a la frecuencia de banda media, se obtiene de acuerdo al ancho de banda. La cantidad de amplificación que cada amplificador suministra se selecciona por medio de los controles de sensibilidad (V/div) del Osciloscopio. Para hacer versátil el Osciloscopio, el control de sensibilidad se diseña para que suministre una amplia selección de niveles de amplificación.

2.1.3.2 Amplificador Horizontal

El amplificador horizontal tiene básicamente dos propósitos:

- 1) Cuando el Osciloscopio se usa en el modo ordinario de operación para desplegar una señal aplicada a la entrada vertical, el amplificador horizontal amplifica la señal de salida del generador de barrido.
- 2) Cuando el Osciloscopio se usa en su modo X-Y, la señal aplicada a la terminal de entrada horizontal se amplificará por medio del amplificador horizontal.

Cuando el Osciloscopio se usa en el modo ordinario de operación, los requerimientos de ganancia y ancho de banda para el amplificador horizontal no son tan rigurosos como los del amplificador vertical. Aunque el amplificador vertical es capaz de reproducir fielmente pequeñas amplitudes, señales de alta frecuencia con rápidos tiempos de levantamientos, el amplificador horizontal se requiere para proporcionar solo una reproducción fiel de la señal de barrido, la cual tiene una amplitud relativamente grande y un tiempo de levantamiento más lento.

Como con el amplificador vertical, la etapa final del amplificador horizontal es un amplificador push-pull (meter-sacar). El amplificador horizontal tiene como primera etapa, una red atenuadora, la cual reduce por medio de un divisor de voltajes, la amplitud de la señal de entrada horizontal a un nivel igual a la sensibilidad de el amplificador horizontal.

2.1.4 Circuito de la Base de Tiempo

La aplicación más común de un Osciloscopio consiste en observar la variación de una señal con respecto al tiempo. Para hacer esto se debe aplicar a las placas horizontales del Osciloscopio un voltaje que haga la posición horizontal del haz proporcional al tiempo. Además, este voltaje se debe aplicar repetidamente para que el haz pueda trazar el mismo camino varias veces de tal forma que el punto móvil aparezca como una línea sólida. Finalmente, este voltaje se debe sincronizar con la señal que se observa para que el mismo camino se repita siempre y resulte una imagen estable en la pantalla. El circuito de la Base de Tiempo del Osciloscopio tiene el trabajo de producir tal señal de voltaje sincronizada y repetida.

2.1.5 Generador de Barrido

La señal generada por la base de tiempo se llama la señal de barrido o diente de sierra o señal rampa. La perilla de esta señal mueve horizontalmente el haz a través de la pantalla. La forma de un ciclo de la señal de barrido se muestra en la Figura 2.2, donde V_H es el voltaje aplicado a las placas horizontales del TRC. El punto del haz electrónico está localizado en el extremo izquierdo de la pantalla cuando $t=0$, el voltaje creciente de la señal de barrido moverá el punto horizontalmente a través de la pantalla. Al final de T_1 segundos, el punto se habrá movido a través de la totalidad de la longitud de la pantalla. Durante el tiempo de T_1 a T_2 , V_H decrecerá hasta cero y el punto retornará rápidamente al extremo izquierdo. Puesto que V_H incrementa linealmente con el tiempo desde $t=0$ hasta $t=T_1$, la posición del punto durante este intervalo de tiempo será proporcional al tiempo transcurrido desde el comienzo de la señal de barrido. El interruptor denominado tiempo/div, colocado al frente del

Osciloscopio, determina cuanto tiempo le toma a la señal de barrido mover el punto a través de una división de la pantalla. Si no hay una señal externa aplicada a las placas verticales, la señal de barrido obligará al punto a trazar una línea horizontal en la pantalla del Osciloscopio. Si hay una señal en la entrada vertical, la señal de barrido hará que el Osciloscopio despliegue un trazo de Voltaje contra Tiempo.

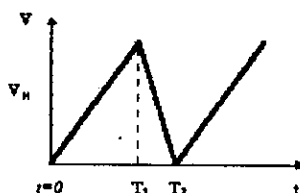


Figura 2.2: Ciclo de la Señal de Barrido

2.1.6. Circuito de Disparo

Se indica cómo se efectúa el disparo mediante una señal exterior. El disparo inicia simultáneamente el barrido y el estímulo. La respuesta provocada es “desplazada” durante el intervalo de barrido. Si el disparo es repetitivo, las respuestas son superpuestas y las diferencias entre la amplitud o retardo de la respuesta son fácilmente discernibles. Del mismo modo la propia señal puede iniciar el barrido.

Cuando el nivel del potencial alcanza el valor elegido para el nivel de disparo, un pulso de disparo es generado e inicia el barrido. El nivel de disparo puede ser ajustado a cualquier valor ya sea positivo o negativo, de modo que el barrido pueda ser iniciado en cualquier porción de la forma de onda de la señal; esta técnica se le denomina disparo interno. En este ejemplo es aparente que la frecuencia de barrido es

un submúltiplo de la frecuencia de la señal periódica y que sucesivas señales de la imagen son superpuestas. En cada uno de los ejemplos anteriores un barrido producido por el pulso de disparo deberá ser usado. Para este modo de operación, el voltaje de la placa deflectora se mantiene a un valor constante, hasta que el barrido sea iniciado por el disparo. Después de que el barrido se inicia, las condiciones originales son restauradas hasta que aparece otro pulso de disparo.

En algunos Osciloscopios de calidad, se puede seleccionar el tipo de señal de disparo a usar. Existen métodos adecuados para seleccionarla. Para disparar el barrido desde alguna fuente externa con una forma de onda que esté relacionada con la señal a observar se conecta la señal externa al conector de entrada del disparo y se sitúa el selector de disparo en la posición Ext. AC o Ext DC. El disparo externo proporciona grandes ventajas con respecto a otros modos de disparo en ciertas condiciones. Con el método de disparo exterior la señal permanece esencialmente constante en forma y amplitud. Es posible obtener sin distorsión la forma de onda de una señal, así como también su amplificación para cada etapa del circuito sin modificar los controles de "disparo" para cada observación. También pueden verse el intervalo de tiempo y las relaciones de fase entre las formas de onda a diferentes puntos del circuito. Por ejemplo si la señal de disparo se produce con la forma de onda de la señal de entrada del circuito, la relación de fase y tiempo de las formas de onda en cada punto del circuito son automáticamente comparadas con la señal de entrada que aparece sobre la pantalla del TRC.

2.1.7 Selector de Entrada

El selector de entrada consiste de un interruptor de tres posiciones: AC (Corriente Alterna), GND (Tierra) y DC (Corriente Directa). Colocando el selector de entrada en la posición AC se acopla capacitivamente la señal de voltaje al atenuador. El capacitor bloquea las componentes de DC de la señal de entrada y permite únicamente que la componente de AC entren al amplificador. Esta característica es útil para medir voltajes de AC que tienen superpuestos otros voltajes de DC o voltajes de polarización. Por otra parte, colocando el selector de entrada en la posición DC se conecta la señal de voltaje directamente al atenuador de tal forma que ambas componentes de DC y AC se aplican al amplificador. Éste modo de medición es particularmente importante para determinar valores instantáneos totales de voltaje. La conexión a GND (tierra) del selector de entrada, disponible en algunos Osciloscopios, es una posición intermedia entre DC y AC que por seguridad elimina cualquier carga almacenada en el atenuador de entrada aterrizándolo momentáneamente cuando se pasa del modo AC al de DC.

2.1.8 Atenuador de Entrada

El atenuador de entrada consiste de un conjunto de divisores de voltaje RC, controlados desde el panel frontal del Osciloscopio por medio del control Volts/div. Éste selector está calibrado en términos del factor de deflexión (V/div), normalmente en la secuencia 1-2-5. Un rango típico del atenuador es 0.1, 0.2, 0.5, 1, 2, 5, 10, 20 y 50 Volts por División, con la máxima atenuación es este último rango.

Para asegurar la operación lineal del Osciloscopio sobre un rango específico de frecuencia (el ancho de banda típico es de DC a 25 MHz), la atenuación de la señal de entrada debe ser independiente de la frecuencia, y esto requiere del llamado atenuador compensado. Con el interruptor del atenuador en la posición superior, la señal de entrada V_i , se conecta directamente a la entrada del amplificador vertical, sin atenuación, es decir si se conecta directamente a la posición 0.1 V/div o máxima sensibilidad del sistema de deflexión vertical.

Con el interruptor en la posición más baja, una red atenuadora $R_a C_a$ está conectada al circuito y una división de voltaje toma lugar, entonces, el voltaje de salida V_o es proporcional a la razón entre la impedancia de entrada del amplificador y la impedancia total del circuito. En el atenuador compensado esta relación de impedancias se conserva constante e independiente de la frecuencia de la señal de voltaje, ajustando C_a de tal forma que la constante de tiempo $R_a C_a$ es igual a la constante $R_i C_i$. Dicho puente está balanceado cuando $R_a C_i = R_i C_a$ o cuando $R_a C_a = R_i C_i$.

En la conexión X-Y no hay corriente y por lo tanto se puede omitir en el circuito atenuador. Por consiguiente, el voltaje de salida está determinado por el divisor de voltaje resistivo, y es igual a (independientemente de la frecuencia de la señal de la entrada):

$$V_o = \frac{R_i}{R_i + R_a} V_i$$

Para balancear el puente y compensar el atenuador, un método práctico consiste en la aplicación de una señal de prueba de onda cuadrada (calibrador) a la entrada del atenuador y ajustar C_a hasta que el voltaje de salida observado en la

pantalla del TRC es una réplica exacta de la señal de entrada. Con una señal de calibración aplicada a la entrada y Ca ajustado correctamente, el voltaje de salida es una onda cuadrada.

Una sobrecompensación ocurre cuando Ca es muy grande y se produce un sobresalto en el pulso, mientras que a altas frecuencias cualquier onda sinusoidal aparece más larga. La subcompensación ocurre cuando Ca es muy pequeño y esto ocasiona esquinas redondeadas en la forma de onda del calibrador, mientras la señal a altas frecuencias sufre una excesiva atenuación.

2.2 Señales de Entrada al Osciloscopio

La gran mayoría de los Osciloscopio cuentan con amplificadores directamente acoplados. El acoplamiento se refiere a un amplificador que es capaz de amplificar tanto señales de DC como de AC. Sin embargo existe un límite máximo de frecuencias por encima de las cuales un amplificador directamente acoplado no es efectivo.

Existen dos tipos de entradas a través de las cuales una señal se puede conectar a un Osciloscopio: la Entrada Simple y la Entrada Diferencial. Sus nombres se derivan del tipo de amplificador al cual están conectadas.

La entrada diferencial tiene dos terminales además de la terminal de tierra para cada canal amplificador. Un ejemplo de este tipo de entrada se muestra en el canal amplificador vertical del Osciloscopio. Con una entrada diferencial, el voltaje

entre dos puntos no aterrizados de un circuito se puede medir sin necesidad de "flotar" el Osciloscopio. Simplemente se conecta cada punto a una de las dos terminales de entrada diferencial y el amplificador electrónicamente sustrae los niveles de voltaje aplicados a las dos terminales y despliega la diferencia en la pantalla. Además son reducidas las interferencias no deseadas y los ruidos en la señal usando el amplificador diferencial. Un amplificador diferencial es más complejo que un amplificador simple, por eso los Osciloscopios diseñados con entrada diferencial son más costosos.

Para medir un voltaje con relación a tierra a través de una entrada diferencial basta utilizar una de las terminales (y el terminal a tierra). La otra terminal se deja sin conectar e internamente su circuito es aterrizado. En este tipo de aplicación la entrada diferencial se emplea como una entrada simple.

Ya que la señal se conecta al Osciloscopio a través de sus entradas se puede escoger si se alimenta una parte de la señal o toda ella a las placas del TRC. Por ejemplo, la señal puede contener tanto componentes de DC como de AC. Los controles de muchos Osciloscopios permiten escoger que parte de los componentes de la señal se aplicaran a la señal del circuito amplificador.

Cuando en el interruptor de acople de la entrada vertical se selecciona la posición DC, la terminal de entrada acopla (conecta) toda la señal al amplificador vertical (tanto la componente AC como de DC). Por otra parte, si se escoge la posición AC la conexión de entrada al amplificador contiene ahora un capacitor en serie. Este capacitor aparece como un circuito abierto a las componentes de DC, y por tanto las bloquea y evita que entren al circuito amplificador.

De la misma forma, componentes de AC de muy bajas frecuencias también se bloquean (por debajo de 2-10 Hz, dependiendo del Osciloscopio). Entonces una señal que contiene ambas componentes aplicada a una entrada con acople de AC será mostrada en la pantalla sin sus componentes de DC y de muy bajas frecuencias.

El acople de AC en la entrada se utiliza únicamente cuando una señal de alta frecuencia se quiere ver sin sus componentes de DC. Esto es muy útil si la magnitud de la componente de AC de una señal es mucho menor que su nivel de DC. Eliminando la componente de DC, la parte de AC de una señal se puede observar más fácilmente porque cualquier cambio en la mayor componente DC puede colocar toda la señal fuera de la pantalla .

La posición GND del interruptor aterriza el circuito del amplificador. No convierte la terminal de entrada en un punto de tierra para una señal de entrada. El propósito de esta posición es remover cualquier voltaje aplicado a las placas de deflexión del TRC y permitir que el haz de electrones se pueda centrar. Esta opción es útil si se quiere centrar el haz sin necesidad de más controles del Osciloscopio. Si la entrada al Osciloscopio es una entrada diferencial, normalmente se tiene un interruptor de acople de entrada para cada uno de las terminales no aterrizados.

Algunos Osciloscopios tienen amplificadores que no son directamente acoplados. En estos Osciloscopios, señales de AC o componentes de DC no se pueden observar. Únicamente existen acople de AC para las señales de entrada y no se necesita interruptor de acople de entrada.

2.3 Modo de Despliegue de una Señal en un Osciloscopio

Existen dos formas básicas en las cuales un Osciloscopio muestra una señal eléctrica, el modo X-Y y el modo de barrido de disparo. El modo X-Y despliega la gráfica de la variación de dos señales externas, una contra la otra. El modo de barrido de disparo muestra la variación de una señal externa a medida que varía con el tiempo. Éste último es el modo más utilizado.

Cuando se usa en el Osciloscopio el modo X-Y, una señal externa aplicada a las placas horizontales controla la deflexión del haz electrónico hacia la derecha e izquierda (dirección X). Simultáneamente, otra señal externa aplicada a las placas verticales, mueve el haz hacia arriba y abajo (dirección Y). Si ambas señales cambian lentamente con el tiempo, el trazo en la pantalla será un punto luminoso que se mueve lentamente. Si las dos señales son periódicas, el punto móvil trazará el mismo camino consecutivamente. Las señales con frecuencias de 50 Hz o más hacen aparecer el trazo como una línea sólida debido a la persistencia de la visión y del fósforo. Las figuras que forman el despliegue de las señales en éste modo de operación se conocen como Figuras de Lissajous.

En el modo de Barrido de Disparo, la base de tiempo genera una onda de barrido a intervalos exactos de tiempo. La onda en diente de sierra causa que el haz de electrones barra horizontalmente la pantalla a una figura uniforme. Simultáneamente, la señal de entrada vertical defleca el haz electrónico hacia arriba y abajo. Puesto que la posición horizontal del haz es proporcional al tiempo, el trazo resultante es la gráfica de la variación de señal de entrada vertical con respecto al

tiempo. Para que ésta imagen tenga una forma estable en la pantalla, cada señal de barrido debe empezar en el mismo punto de la señal de la entrada vertical. Además para que la señal aparezca como una línea sólida la señal de barrido se debe repetir al menos 50 veces por segundo. La señal de barrido o la longitud de tiempo que toma el haz para barrer horizontalmente una división de la pantalla del Osciloscopio se determina por medio del circuito del generador de barrido.

2.4 Ancho de Banda de un Osciloscopio

El ancho de banda del Osciloscopio es el punto en el cual la señal se reduce a -3 db con respecto a un punto de referencia en el rango de baja frecuencia. Ya que la señal del amplificador vertical se incrementa en frecuencia, el amplificador del Osciloscopio debe disminuir la amplitud. Una disminución controlada de la ganancia es necesaria para proporcionar la respuesta adecuada del amplificador vertical a señales complejas. Los Osciloscopios que tienen una respuesta de frecuencia del amplificador vertical con una reducción mayor de los 6 db por octava, no proporcionan adecuadamente los componentes de alta frecuencia de las formas de onda complejas. Los Osciloscopios con escasa atenuación a las altas frecuencias tienden a deformar la señal de ondas cuadradas.

Una señal reducida a -3 db no es la mitad del voltaje aplicado. a -3 db, es igual a 0.707 del valor de referencia. Una señal reducida en amplitud a -3 db (debido al incremento de la frecuencia) tiene un gran desfaseamiento, con respecto al punto de referencia normalmente cerca de los 45 grados. Algunas veces, el ancho de banda del amplificador vertical de un Osciloscopio es especificado con alguna

limitación en su función. Esta especificación podría ser de 10 MHz a 4 cm. de deflexión y 8 MHz a deflexión total. Dicha especificación indica que el Osciloscopio no puede usarse en el Ancho de Banda total. Ésta especificación se usa para los Osciloscopios de estado sólido, que están limitados en la capacidad de alimentación de las placas. Los Osciloscopios modernos no se especifican de esta manera, algunos Osciloscopios tienen acoplamientos de corriente alterna y de corriente directa y utilizan un selector de posiciones.

Si un Osciloscopio se usa en el modo AC, éste tiene una respuesta mayor en su ancho de banda en el punto de -3 db. Ésto se debe a la pendiente de la curva de respuesta del amplificador y a una disminución del ancho de banda de las respuestas bajas en el punto de -3 db, debido a un capacitor de acoplamiento de entrada. El límite de baja frecuencia (-3 db) cuando se encuentra acoplado en AC, es de 2 a 10 Hz. Cuando se acopla a AC, el voltaje (entre 600 y 400 Volts) que puede aplicarse al capacitor se debe especificar. Esta especificación es para el pico de AC mas el componente de DC.

2.5 Controles Básicos del Osciloscopio

Un Osciloscopio tiene algunos controles que se requieren para manipular las características del haz. A continuación se describe la función de cada uno de ellos. Los nombres varían dependiendo de cada modelo de un Osciloscopio, pero en general son los mismos nombres.

Encendido (Power): Enciende o apaga el Osciloscopio, ya conectado.

Intensidad (Intensity): Controla el brillo del trazo del Osciloscopio. Esta perilla suministra una conexión a la rejilla de control de intensidad del cañón electrónico del TRC. Cuando la perilla se voltea en el sentido de las manecillas del reloj, el voltaje repelente de la rejilla se disminuye y más electrones pueden emerger del orificio en la rejilla del cátodo para formar el haz. Un mayor número de electrones en el haz origina un punto más brillante en la pantalla. Se debe tener cuidado y evitar que el haz electrónico produzca puntos permanentes (queme). en la pantalla. Un punto estacionario se debe conservar el punto en movimiento. Si un halo aparece alrededor del punto, la intensidad es muy alta. Antes de encender el Osciloscopio, hay que bajar la intensidad.

Foco (Focus): El control del Foco está conectado al ánodo del cañón electrónico, el cual comprime el haz de electrones emergentes en un punto fino. Cuando este control se ajusta, el trazo sobre la pantalla del Osciloscopio, se hace tan fino y bien definido como es posible.

Astigmatismo (Astigmatism): Este control trabaja en conjunto con el control de foco para mejorar el foco. En algunos Osciloscopios hay una perilla para el astigmatismo, pero en otros este control requiere un destornillador o está colocado dentro del chasis para su ajuste.

Posición: Las perillas de posición se utilizan para trasladar el trazo o el centro de la imagen mostrada alrededor de la pantalla. Para controlar esta posición se hace por medio de 2 controles: el control de *Posición Horizontal (Horizontal Position)* y el control de *Posición Vertical (Vertical Position)*. Los controles ajustan los voltajes de DC aplicados a las placas de deflexión del TRC. Cuando los controles de

posición horizontal y vertical son puestos en su posición intermedia. la deflexión de los voltajes se divide de igual manera entre ambas mitades de los potenciómetros, por lo tanto no existe deflexión de el rayo; y éste simplemente viaja a lo largo del eje del TRC golpeando el centro de la pantalla.

Iluminación de la Escala (Scale Illumination): Suministra la iluminación a la escala.

Las líneas grabadas de la escala se hacen brillantes por medio de luz aplicada desde los extremos y bordes de la pantalla, evitándose así destellos que interfieren con la señal mostrada.

Sensibilidad Vertical o Volts/div: Determina el valor necesario del voltaje que se debe aplicar a la entrada vertical para deflectar el haz, una división (centímetro) Este control conecta un atenuador por pasos o un potenciómetro al amplificador del Osciloscopio y permite controlar la sensibilidad vertical con pasos discretos.

Ajuste Volts/div: Permite una variación suave de la sensibilidad vertical. Esta perilla se debe colocar en la posición calibrada para que de esta forma la sensibilidad vertical sea igual a la marcada por el interruptor.

Tiempo de Barrido o Tiempo/div. (Time/div): Controla el tiempo que toma el punto para moverse horizontalmente a través de una división de la pantalla cuando se usa el modo de barrido de disparo. Un valor muy pequeño indica un tiempo muy corto de barrido.

Ajuste Tiempo/Div: Este control permite escoger una escala de barrido continua sin calibrar (Tiempo/div). En algunos Osciloscopios este ajuste no existe.

Fuente de Disparo o Selector de Sincronismo (Trigger Source o Sync Selector):

Selecciona la fuente de la señal de disparo. Utilizando este control, uno escoge el tipo de señal que se emplea para sincronizar el barrido horizontal con la señal de entrada vertical. La posible selección normalmente incluye:

- 1) *Interno (Internal).*- La salida del amplificador vertical se usa para disparar el barrido. Con esto se logra que la señal de entrada controle el disparo. Para la mayoría de las aplicaciones este tipo de disparo es el apropiado.
- 2) *Línea (Line).*- Esta posición selecciona el voltaje de la línea de 60 Hz como la señal de disparo. Es útil cuando la frecuencia de la señal de entrada está relacionada con la frecuencia de la línea.
- 3) *Externa (External).*- Cuando se usa esta posición, se debe aplicar una señal externa para disparar la señal de barrido. Esta señal se debe conectar a la entrada marcada como External Trigger Input. La señal de disparo externa debe tener una frecuencia compatible con la señal de entrada vertical para obtener una imagen estable.

Amplificador de Barrido (Sweep Magnifier): Este control permite disminuir el tiempo por división de la onda de barrido. Sin embargo, la reducción es acompañada por una ampliación de una porción de la onda de barrido en vez de tener que cambiar la constante de tiempo del circuito de tiempo que la genera.

Pendiente de Disparo (Trigger Slope): Este interruptor determina si el circuito de pulsos en la base de tiempo responderá a una señal de disparo de pendiente positiva o negativa.

2.6 Osciloscopio con Memoria

La mayoría de las aplicaciones del Osciloscopio Convencional son limitadas por la persistencia del fósforo en el TRC, ya que se tiene que observar la señal en tiempo real, haciendo al mismo tiempo imposible la observación de eventos que ocurren una sola vez y muy cercanos uno del otro. A pesar de que esos eventos se pueden grabar fotográficamente, esto puede resultar costoso, además de que pueden consumir tiempo. El avance tecnológico que han tenido los Osciloscopio han dado lugar al Osciloscopio con Memoria, el cual ha hecho posible extender el tiempo de retención de la señal en el TRC, para que se pueda analizar con detalle el despliegue de la señal, ya que ésta se almacena en una memoria interna del Osciloscopio.

Existen dos tipos de Osciloscopios de almacenamiento, uno es usando un diseño especialmente para el TRC y otro es el que hace uso de técnicas digitales. Para poder mantener el despliegue del TRC de almacenamiento por un tiempo prolongado, éste utiliza dos cañones de electrones. Un cañón es de escritura, como cualquier otro cañón. El otro cañón es de riego, el cual bombardea uniformemente la pantalla del TRC con electrones de baja energía. Las partículas de fósforo colisionadas por esos electrones, toman un nivel bajo de carga, sin embargo las partículas no energizadas se mantienen en un estado "sin carga". Cuando el trazo de una gráfica va a ser grabado el cañón de escritura se enciende y los electrones de alta energía se impactan con la pantalla formando una imagen. Las partículas de fósforo son impactadas por los electrones de alta energía que poseen una carga considerablemente grande, por lo tanto y adicionalmente, los electrones del cañón esparcidor son atraídos,

con los cuales se constituye la imagen. Cuando se quiere borrar la imagen desplegada, se aterriza la pantalla de fósforo, la cual remueve el exceso de carga.

2.7 Osciloscopio de Muestreo

La relación ganancia-ancho de banda del amplificador vertical, limita el rango de frecuencia de señales que se pueden desplegar en un Osciloscopio convencional. Si señales con frecuencias arriba de la frecuencia a la que se pueden ver con un Osciloscopio convencional son repetitivas, estas se pueden retrasar muchas miles de veces por medio del muestreo.

La técnica de muestreo utilizada para observar señales eléctricas repetitivas de muy alta frecuencia, con un Osciloscopio es mostrada la siguiente Figura 2.3. El muestreo de la señal de entrada se toma de ciclos sucesivos, con una muestra tomada por ciclo y cada muestra un poco retrasada con respecto a la muestra precedente. La forma de onda de la señal reconstruida, la cual es desplegada sobre la pantalla de TRC, es una forma de onda compuesta de las muestras tomadas de los ciclos sucesivos de la señal de entrada.

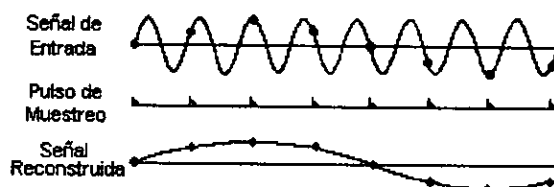


Figura 2.3: Principio de Muestreo

2.8 Osciloscopios de Almacenamiento Digital

El Osciloscopio de Almacenamiento convencional, se ha usado por varios años para retener formas de onda o señales transitorias (temporales). Sin embargo, las imágenes guardadas se desaparecen gradualmente y no se pueden llamar nuevamente en éste tipo de instrumentos.

Los Osciloscopios de Almacenamiento Digital, proporcionan grandes beneficios cuando se compara con los Osciloscopios analógicos. Por ejemplo, el despliegue digital nunca se desvanece o desaparece. Por otra parte el Osciloscopio de almacenamiento digital se puede conectar con calculadoras de escritorio o microcomputadoras para obtener una gran variedad de mediciones rápidas y precisas.

Comparada a la técnica usada en Osciloscopios de almacenamiento analógico en el despliegue de señales por un período prolongado de tiempo, los TRC de los Osciloscopios de almacenamiento digital son mucho más simples. Los Osciloscopios de Almacenamiento Digital usan un TRC convencional sin los cañones de esparcimiento o riego que usa el Osciloscopio de almacenamiento analógico. Los Osciloscopios de Almacenamiento Digital incorporan un circuito de memoria para guardar información acerca de la señal. La forma de onda se lee repetidamente de la memoria para formar el despliegue sobre la pantalla del TRC.

Capítulo 3:

Señales Eléctricas

Las corrientes y voltajes de la energía eléctrica tienen diversas características como son la amplitud, formas de onda, frecuencia y periodo. En la electrónica, las características de la corriente y el voltaje son de mucha importancia, ya que por medio de éstas se puede transmitir información. Las corrientes o voltajes que transmiten información se conocen generalmente como señales eléctricas.

Cualquier estudio de la electrónica debe incluir una descripción y un análisis de los tipos de señales más comunes, debido a que son ellas las que transportan la información dentro de los aparatos electrónicos.

Existen dos tipos de señales: las señales de corriente continua y las señales de corriente alterna.

3.1 Señales de Corriente Continua

Las señales de corriente continua mantienen un solo valor de amplitud todo el tiempo. Es decir que si observamos una gráfica de voltaje o corriente en el tiempo observaremos una línea recta todo el tiempo. La Figura 3.1 muestra un ejemplo que representa gráficamente de corriente continua:

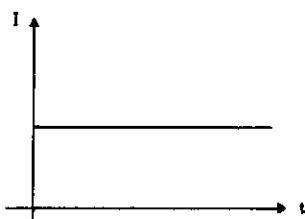


Figura 3.1: Ejemplo de Corriente Continua

Las señales de corriente continua tienen muchas aplicaciones prácticas, la mayoría de sus aplicaciones se encuentran en sistemas digitales.

Los voltajes de corriente continua se usan ampliamente para suministrar potencia, de hecho, la mayor parte de la potencia suministrada a los circuitos electrónicos de un sistema, es de corriente continua.

3.2 Señales de Corriente Alterna

Las señales de corriente alterna se utilizan bastante para transmitir o producir información, los conceptos de corriente alterna son los mismos sin importar la aplicación que se le dé. La corriente alterna siempre varía en magnitud, tiempo y forma y éstas son afectadas por inductores y capacitores. La corriente alterna tiene como propiedades básicas frecuencia, fase y amplitud. Es posible representarla gráficamente por medio de formas de onda. Las gráficas de la Figura 3.2 muestran las formas de onda que puede formar la corriente alterna:

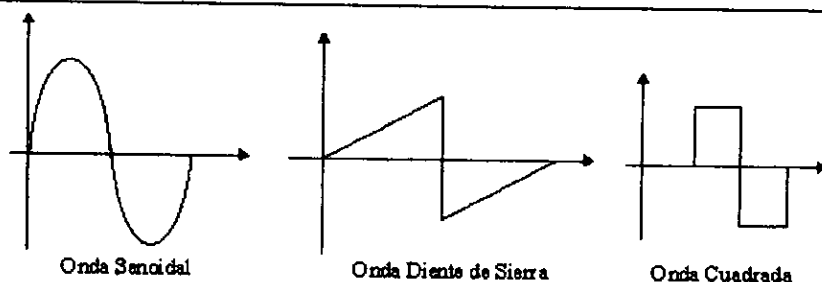


Figura 3.2: Formas de Onda de Señales de Corriente Alterna

Las señales de corriente alterna pueden ser señales analógicas o señales digitales.

3.2.1 Señales Analógicas

Las señales analógicas son análogas a la señal física que representan. La magnitud de una señal analógica puede tomar cualquier valor; es decir que la amplitud de una señal analógica presenta una variación continua sobre su intervalo de actividad. Los circuitos que procesan estas señales se les conoce como circuitos analógicos.

3.2.2 Señales Digitales

Otra forma de representar una señal es por medio de una sucesión de números, cada uno de ellos representa la magnitud de la señal en un instante de tiempo. A estas señales se les conoce como señales digitales. Las señales digitales se definen solamente en los instantes de muestreo. No son señales continuas, si no señales

de tiempo discretas. Los circuitos que procesan señales digitales se les conoce como circuitos digitales. La computadora digital esta constituida por una gran cantidad de circuitos digitales. El procesamiento digital de las señales ha llegado a ser ampliamente usado, debido a los grandes avances que se han hecho en el diseño y en la fabricación de los circuitos digitales. Además con los circuitos digitales es mucho más fácil trabajar con señales digitales que con analógicas.

El procesamiento digital de señales es económico y confiable, y además permite que se realicen una gran variedad de funciones, las cuales serían imposibles o poco prácticas llevarlas a cabo por medios analógicos. Sin embargo la mayoría de las señales en el mundo físico son analógicas, y por ello existen tareas de procesamiento que se realizan mejor mediante circuitos analógicos.

3.3 Frecuencia y Fase de las Ondas Senoidales

Como ya se mencionó, las formas de onda de las señales se analizan y se miden en muchas aplicaciones eléctricas. Cuando una señal de corriente alterna varía repetitivamente en forma continua (independientemente de la forma de repetición), a la onda se le llama Onda Periódica. La forma de onda periódica que con más frecuencia se encuentra en los sistemas eléctricos es la senoidal. Sus características son comunes a las de muchas otras ondas periódicas. La Figura 3.3 muestra un ejemplo de una senoide y su expresión matemática es:

$$v = V_0 \sin \omega t = V_0 \sin(2\pi ft)$$

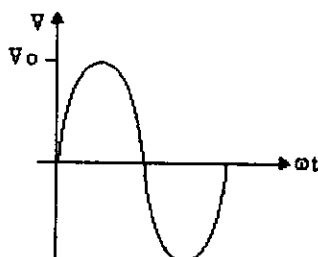


Figura 3.3: Onda Senoidal

La amplitud de la onda senoidal, y de cualquier onda cuadrada, es el valor máximo de la función y en este caso está dado por V_0 , en la ecuación anterior. La frecuencia f se define como el número de ciclos que pasan en un segundo. De acuerdo con lo anterior la frecuencia es medida en ciclos por segundo o hertz (Hz). El tiempo transcurrido durante un ciclo de onda se llama periodo, T . La frecuencia y el periodo de la onda son inversos es decir:

$$f = \frac{1}{T}$$

Además, un ciclo de una onda abarca 2π radianes. Si se multiplican 2π por la frecuencia, se obtiene la frecuencia angular (ω) de la onda senoidal. Lo anterior es expresado:

$$\omega = 2\pi f = \frac{2\pi}{T}$$

Las unidades de ω son radianes por segundo. En la Figura 3.4 se grafican dos ondas senoidales de frecuencias iguales en un mismo eje de tiempo y desfasadas en el tiempo una de la otra.

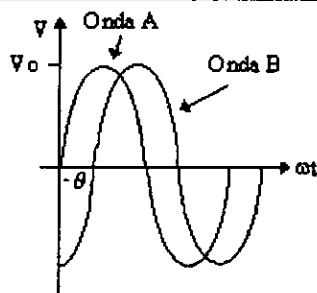


Figura 3.4: Ondas senoidales desfasadas en el Tiempo

Las ecuaciones de estas dos ondas difieren en el valor de sus ángulos de fase. El concepto de ángulo de fase implica la comparación de dos cantidades relativas, es decir, si se define la onda *A* como teniendo un ángulo de fase cero, su ecuación es:

$$v = V_0 \sin \omega t$$

Entonces, la onda *B* tendrá un ángulo θ de fase, que indica cuanto se encuentra desfasada la onda *B* de la onda *A*, en el tiempo. Si el primer valor de la onda *B* se presenta después del primer valor de la onda *A*, se dice que la onda *B* sigue a la onda *A*, o esta atrasada con respecto a la onda *A* y viceversa. En la figura anterior se puede decir que la onda *B* está atrasada cada θ grados con respecto a la onda *A*. Entonces la ecuación de la onda *B* se escribe:

$$v = V_0 \sin(\omega t - \theta)$$

donde el signo menos de θ indica que la onda de la ecuación de la onda *B* está atrasada un ángulo θ con respecto a la onda de la ecuación de la onda *A*.

3.4 Valor Promedio y Valor Cuadrático Medio

En señales de corriente continua, es fácil medir la corriente o calcular la energía disipada por los componentes de algún circuito, solo basta con hacer una medición en un instante de tiempo. En las señales de corriente alterna, como su amplitud varia en el tiempo, no es suficiente medir solamente, la corriente o la energía disipada, en un instante de tiempo, puesto que no sería posible determinar todo lo que se debe conocer sobre la señal. Sin embargo, es posible determinar la forma de onda de una señal de corriente alterna y también es posible determinar algunos de sus valores característicos. Estos valores se pueden emplear para comparar la efectividad de varias formas de onda con respecto a la de otra forma, o para predecir los efectos que tendrá una forma de onda determinada sobre el circuito al que se aplica.

Los valores característicos que se emplean con mayor frecuencia en ondas de señales de corriente alterna son sus valores promedio y su valor medio cuadrático (rms).

3.4.1 Valor Promedio

El valor promedio de una onda de corriente alterna, que varia a lo largo de un periodo T es el valor que tendría una corriente directa si se suministrara una cantidad igual de carga en el mismo periodo T . Matemáticamente el valor promedio se obtiene dividiendo el área bajo la curva de la onda en un periodo T , entre el tiempo del periodo, es decir:

$$A_{prom} = \frac{\text{Area bajo la curva}}{\text{Longitud del Periodo (segundos)}}$$

.....(3.1)

donde A_{prom} es el valor promedio.

En forma más general, esta expresión se define:

$$A_{prom} = \frac{1}{T} \int_0^T f(t) dt$$

.....(3.2)

donde T es la longitud del periodo de la curva y $f(t)$ es la ecuación de la forma de onda.

Por ejemplo si se desea calcular el valor promedio de la Figura 3.5, entonces se calcula el área bajo la curva y se calcula el valor promedio mediante la ecuación 3.1.

$$A_{prom} = \frac{(60 \times 1) + (0) - (10 \times 3)}{5} = 6$$

Otro ejemplo seria calcular el valor promedio de la senoide que se muestra en la Figura 3.6.

Para calcular el valor promedio se utiliza la función que describe a una senoide en la ecuación 3.2.

$$\begin{aligned}
 A_{\text{prom}} &= \frac{1}{T} \int_0^T A_0 \sin \frac{2\pi}{T} t \, dt = \frac{A_0}{T} \int_0^T \sin \frac{2\pi}{T} t \, dt \\
 &= -\frac{TA_0}{2\pi T} \left[\cos \frac{2\pi t}{T} \right]_0^T \\
 &= \frac{A_0}{2\pi} [(1) - (+1)] \\
 &= 0
 \end{aligned}$$

Por medio de este ejemplo se demuestra que el valor promedio de una onda puramente senoidal es cero.

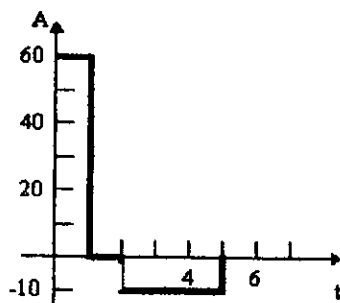


Figura 3.5: Ejemplo del Valor Promedio

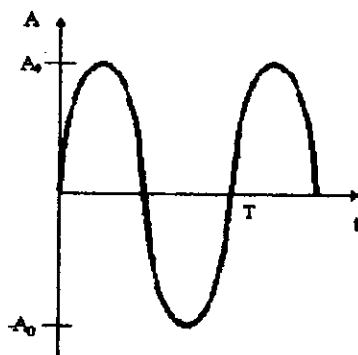


Figura 3.6: Ejemplo del Valor Promedio para una Senoidal

3.4.2 Valor Cuadrático Medio

Otro valor característico de una onda de corriente alternas es el valor Cuadrático Medio (RMS; Root Mean Square). Este valor se utiliza con más frecuencia que el valor promedio para describir las señales eléctricas, debido a que el valor promedio de las ondas simétricas es cero, y esto no es una información muy útil sobre la propiedad de una señal. Por el contrario, el valor RMS de cualquier onda nunca es cero.

El valor RMS de una onda se relaciona con su capacidad de suministrar energía; de acuerdo con ésta, a veces al valor RMS también se le denomina valor efectivo. El valor RMS es el valor de una señal de corriente continua que entregaría la misma cantidad de potencia si se sustituyera por la señal de corriente alterna en cuestión.

La potencia se define como el producto de la corriente y el voltaje::

$$P = V I$$

además, de la ley de ohm sabemos que:

$$V = R I$$

$$I = \frac{V}{R}$$

por lo tanto sustituyendo la última ecuación en la ecuación 3.3 obtenemos:

$$P = \frac{V^2}{R}$$

si sabemos que el voltaje es una función en el tiempo entonces la potencia quedaría definida como:

$$P = \frac{f(t)^2}{R}$$

entonces el valor promedio de la potencia es:

$$P = \frac{1}{RT} \int_0^T f(t)^2 dt$$

donde R es la resistencia a la que se le está suministrando la energía.

Por otro lado el valor del voltaje RMS es el valor equivalente al valor del voltaje de una señal de corriente continua de igual potencia, así que la potencia, para el valor RMS, se podría definir de la siguiente forma:

$$P = \frac{A_{rms}^2}{R}$$

igualando las ecuaciones 3.5 y 3.6 obtenemos:

$$\frac{A_{rms}^2}{R} = \frac{1}{RT} \int_0^T f(t)^2 dt$$

despejando el valor RMS :

$$A_{rms} = \sqrt{\frac{1}{T} \int_0^T f(t)^2 dt}$$

Comparando la ecuación 3.2, que define al valor promedio, y la ecuación anterior, que define el valor RMS se deduce que:

$$A_{rms} = \sqrt{\text{promedio}[f(t)^2]}$$

$$A_{rms} = \sqrt{\frac{\text{área bajo la curva}[f(t)^2]}{T}}$$

Para ejemplificar lo anterior se calcula el valor RMS de la Figura 3.6.

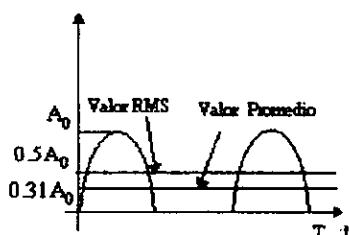
$$A_{rms} = \sqrt{\frac{1}{T} \int_0^T [f(t)]^2 dt} = \sqrt{\frac{1}{T} \int_0^T A_0^2 \left(\sin \frac{2\pi}{T} t \right)^2 dt}$$

$$= \sqrt{\frac{A_0^2}{T} \left[\frac{t}{2} - \frac{T \sin(4\pi/T)t}{4\pi} \right]_0^T} = \frac{A_0}{\sqrt{2}}$$

$$= \frac{A_0}{\sqrt{2}} = 0.707 A_0$$

así que el valor RMS de una senoidal es su amplitud entre raíz de dos.

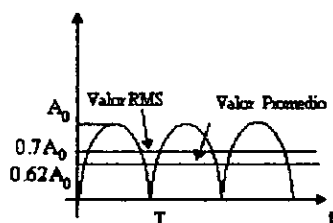
A continuación se muestran algunas formas de onda de señales eléctricas (Figura 3.7), que se presentan con frecuencia en sistemas eléctricos. Para cada una de ellas se dan su valor RMS y su valor promedio con respecto a sus amplitudes pico. La onda senoidal también se presenta con frecuencia, sin embargo no se incluye a continuación puesto que ya se examinó.



Media Onda Senoidal Rectificada

$$\text{Valor Promedio} = 0.31 A_0$$

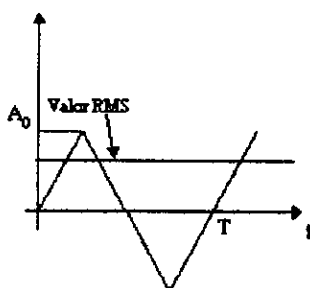
$$\text{Valor RMS} = 0.5 A_0$$



Onda Senoidal Completa Rectificada

$$\text{Valor Promedio} = \frac{2}{\pi} A_0 = 0.636 A_0$$

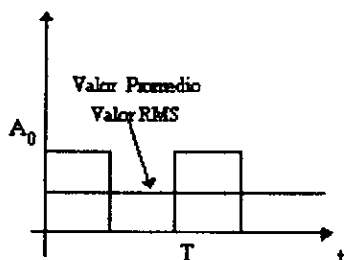
$$\text{Valor RMS} = \frac{A_0}{\sqrt{2}}$$



Onda Triangular

$$\text{Valor Promedio} = 0$$

$$\text{Valor RMS} = \sqrt{\frac{A_0^2}{3}}$$



Onda Cuadrada

$$\text{Valor Promedio} = \frac{A_0}{2}$$

$$\text{Valor RMS} = \frac{A_0}{2}$$

Figura 3.7: Valores Promedio y RMS de Señales Eléctricas

Capítulo 4:

Convertidores de Señales

4.1 Introducción

La utilización de microprocesadores como dispositivos de control de sistemas reales hace necesario un intercambio de información entre dichos sistemas y la microcomputadora. Estas informaciones pueden ser señales digitales, o señales analógicas, las cuales deben ser convertidas en señales eléctricas mediante transductores y posteriormente en señales digitales mediante dispositivos que realicen la conversión analógica a digital para que el microprocesador pueda realizar el algoritmo residente en su memoria.

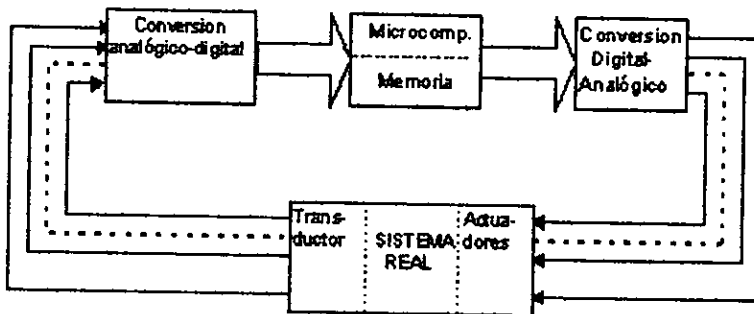


Figura 4.1: Control de un Sistema Real Mediante una Microcomputadora

Se pueden distinguir en el proceso de control tres partes:.

Adquisición de datos: Este sistema consiste en un circuito de acondicionamiento de la señal analógica (normalmente, un amplificador y un filtro).

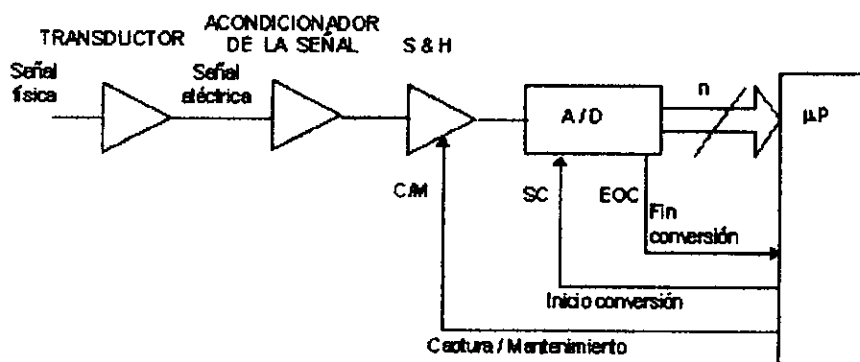


Figura 4.2: Sistema de Adquisición de Datos de una Única Entrada Analógica

Un Circuito de Captura y Mantenimiento (Sample & Hold): Almacena el nivel analógico que está en un determinado instante en su entrada y lo mantiene mientras se realiza la conversión.

El convertidor analógico-digital: Este convierte el nivel analógico, en una palabra digital de n bits, la cual, una vez terminada la conversión se almacena en la memoria de la microcomputadora.

Cuando son varios los transductores que se deben leer secuencialmente, se puede utilizar un selector o multiplexor analógico entre los acondicionadores de señal y los circuitos de captura y mantenimiento.

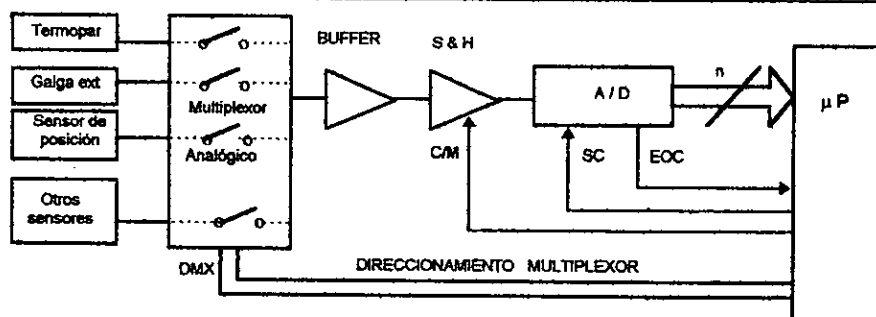


Figura 4.3: Adquisición de Datos Mediante Multiplexor Analógico

El multiplexor es direccionado de tal manera que cada entrada analógica se conecta al convertidor secuencialmente. Este método permite una economía del espacio y del costo de la circuitería, ya que con un solo circuito de captura y mantenimiento y un convertidor analógico-digital se pueden tratar gran cantidad de entradas analógicas, a costa de reducir la velocidad de adquisición de datos en relación a la que se obtendría utilizando tantos sistemas sencillos como entradas hubiera.

Cuando es necesario que una cierta cantidad de señales analógicas sean leídas simultáneamente, cada canal debe tener su propio circuito de captura y mantenimiento, los cuales son a continuación seleccionados secuencialmente por medio de un multiplexor para su conversión digital por medio de un convertidor analógico-digital.

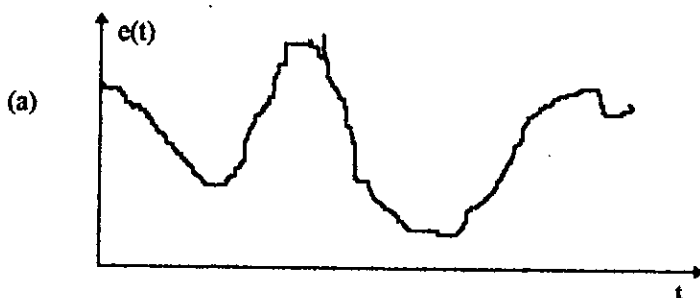
4.2 Proceso de Muestreo

Las señales analógicas que se pueden obtener normalmente procedentes de transductores presentan un dominio temporal y un gran margen de valores continuos.

Esta misma señal muestreada, es de valores continuos, pero el dominio temporal es discreto, o sea, en el tiempo $(t_{k+1} - t_k)$, el valor de la señal es constante.

Esto es, la misma señal digitalizada, la cual presenta un dominio temporal y un margen de valores discretos; sólo puede tomar valores que se encuentren a una distancia entre ellos igual a $m(v_{i+1} - v_i)$, $m=1,2,3,\dots$

La Figura 4.4 nos muestra la gráfica de la señal analógica procedente de un transductor (a); la gráfica de misma señal muestreada (b); la gráfica de la misma señal pero digitalizada (c); impulsos de muestreo (d).



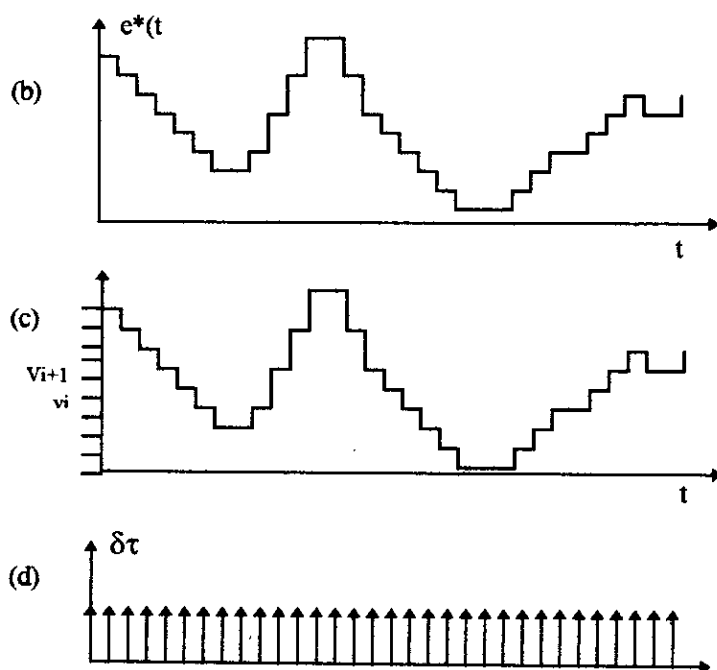


Figura 4.4: a) Señal Analógica Procedente de un Transductor ; b) Señal Muestreada; c) Señal Digitalizada; d) Impulsos de Muestreo.

La relación entre las tres señales anteriores puede representarse según el diagrama de la Figura 4.5.

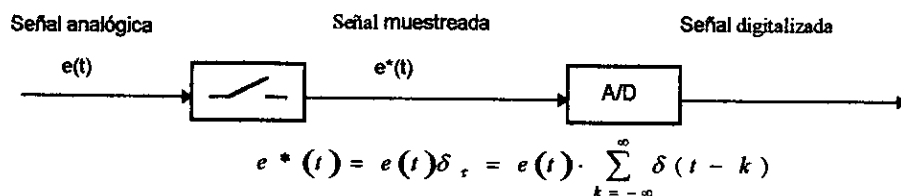


Figura 4.5: Diagrama del Proceso de Muestreo

La señal analógica continua, $e(t)$, mediante un muestreador $\delta\tau$ se convierte en una señal muestreada $e^*(t) = e(t) \cdot \delta\tau$, la cual a su vez se transforma en una señal digitalizada mediante un convertidor analógico-digital.

En cada uno de los procesos que sufre la señal analógica, se pierde información debido a la discretización que se realiza; se puede reducir el error en la señal muestreada haciendo el período de muestreo suficientemente pequeño, pero éste queda limitado por la anchura de banda del muestreador real, o sea por el tiempo mínimo necesario para muestrear.

El análisis de Fourier de una señal analógica (descomposición en series de Fourier si ésta es periódica o la integral de Fourier en caso contrario) indica la composición espectral dicha señal. Los coeficientes de las series de senos y cosenos dan información respecto a la amplitud y la fase de cada frecuencia. El teorema de Nyquist o del muestreo expresa el límite inferior de la velocidad de muestreo, conocida la máxima componente frecuencial de la señal de entrada, como la correspondiente al doble de dicha frecuencia. Esta frecuencia denominada a menudo frecuencia de Nyquist, es la mínima con la que puede llegarse a reconstruir la señal inicial de entrada.

La Figura 4.6 muestra el error de solapamiento producido al muestrear las dos señales senoidales de frecuencias f_1 y f_2 , con un período de muestreo T_m superior al período de muestreo máximo, $T_{mmax} = (1/2) (1/f_2) = 1/2f_2$.

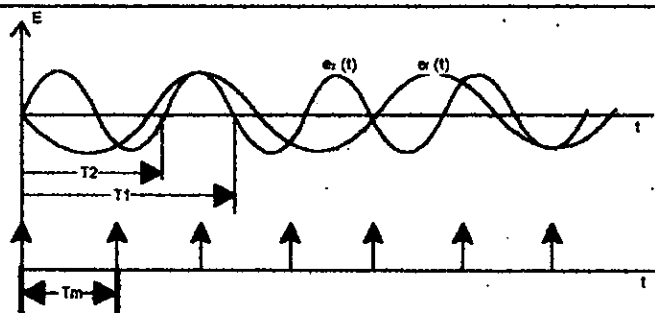


Figura 4.6: Error de Solapamiento en el Muestreo

En los sistemas reales se suele escoger una frecuencia de muestreo (especificada en muchos casos como la correspondiente a la frecuencia a la cual el sistema puede muestrear el dato, realizar la conversión y preparar el sistema para realizar una nueva conversión) de un valor comprendido entre 5 y 10 veces superior a la máxima componente frecuencial significativa de la señal de entrada.

En el caso de que el sistema de adquisición no admita una frecuencia de muestreo suficientemente alta frente a una señal analógica de entrada que presente elevadas componentes frecuenciales (en muchos de los casos debidas al ruido superpuesto a la señal analógica) debe recurrirse a la realización de un filtrado previo de dicha señal antes de introducirla en el muestreador, el cual, aunque no elimina totalmente el ruido de altas frecuencias, los puede atenuar suficientemente para que no afecten al resultado de la conversión. En muchos de los casos es conveniente la realización de un filtrado digital de la señal digitalizada una vez introducida en la microcomputadora.

4.3 Convertidores Analógico-Digitales

Un convertidor analógico-digital es un dispositivo que recibe una señal de entrada, V_{ent} , y la transforma en una palabra digital, P_d , con una precisión y resolución dadas, mediante una comparación con una tensión de referencia, V_{ref} . Para un convertidor analógico-digital ideal, la palabra digital está relacionada con la señal de entrada por la siguiente expresión:

$$P_D = \frac{V_{ent}}{V_{ref}} = \sum_{i=1}^n \frac{\delta_i}{2^i}, \delta_i = \{0,1\}$$

Este cociente se realiza en binario, por lo que el resultado exacto sería de un número infinito de bits, pero se comete un error de cuantificación al tener en cuenta únicamente un número limitado de bits, el cual viene determinado por la capacidad del convertidor A/D. Si este es de n bits, se realiza una cuantificación de la señal analógica en 2^n niveles, el error máximo cometido es de la mitad de la distancia entre dos niveles, quedando la expresión:

$$\varepsilon_{max} = \frac{1}{2} \cdot \frac{V_{ref}}{2^n}$$

Los convertidores A/D admiten señales analógicas de entrada, de corriente o de tensión, de una única polaridad (de márgenes 0 y +V) o bipolar (de márgenes +V y -V).

La Tabla 4.1 indica alguno de los códigos usuales de la salida de los convertidores A/D y el valor del escalado correspondiente a la salida digital.

Las características generales para los distintos tipos de convertidores dependen en gran medida de la forma en que se realiza la conversión.

Unipolar binaria	$V_s = V_{ref} \left[\sum_{i=1}^n \frac{\delta_i}{2^i} \right]; \delta_i \in \{1,0\}$	11111111 $\rightarrow$ 9.961 V
Unipolar binaria	$V_{max} = V_{ref} \left[1 - \frac{1}{2^n} \right]$	00000000 $\rightarrow$ 0 V
Bipolar binaria	$V_s = V_{ref} \left[\sum_{i=1}^n \left(\frac{\delta_i}{2^{i-1}} \right) - 1 \right]$	11111111 $\rightarrow$ 9.922 V
Bipolar binaria	$V_{max\_posit} = V_{ref} \left[1 - \frac{1}{2^{n-1}} \right]$	10000000 $\rightarrow$ 0 V
Bipolar binaria	$V_{max\_neg} = -V_{ref}$	00000000 $\rightarrow$ -10 V
Complemento a uno	$V_s = V_{ref} \left[\sum_{i=2}^n \left(\frac{\delta_i}{2^{i-1}} \right) - \delta_1 + \frac{\delta_1}{2^{n-1}} \right]$	01111111 $\rightarrow$ 9.922 V
Complemento a uno	$V_{max\_posit} = V_{ref} \left(1 - \frac{1}{2^{n-1}} \right)$	00000000 11111111 $\rightarrow$ 0 V
Complemento a uno	$V_{max\_neg} = -V_{ref} \left(1 - \frac{1}{2^{n-1}} \right)$	10000000 $\rightarrow$ -9.922 V
Complemento a dos	$V_s = V_{ref} \left[\sum_{i=2}^n \left(\frac{\delta_i}{2^{i-1}} \right) - \delta_1 \right]$	01111111 $\rightarrow$ 9.222 V
Complemento a dos	$V_{max\_posit} = V_{ref} \left(1 - \frac{1}{2^{n-1}} \right)$	00000000 $\rightarrow$ 0 V
Complemento a dos	$V_{max\_neg} = -V_{ref}$	10000000 $\rightarrow$ -10 V

NOTA: para 8 bits y $V_{ref} = 10$ V

Tabla 4.1: Códigos Mas Usuales en los Convertidores A/D

Una primera clasificación se puede basar en si la conversión se realiza directamente o se realiza una transformación de la señal de entrada en una variable intermedia que luego se transforma en digital. El convertidor paralelo compara simultáneamente la señal analógica de entrada con una serie de niveles de referencia correspondientes a los valores del escalado cuantificados utilizando por lo tanto, tantos comparadores como niveles se desea obtener. El tiempo de conversión es

muy pequeño, ya que es del orden de nano segundos, pero el número elevado de componentes hace que el sistema sea costoso y en general de pocos bits. No se acostumbra utilizar microprocesadores estándar debido a la disparidad entre el tiempo de conversión y el tiempo de ciclo.

Entre los convertidores indirectos que transforman la señal de entrada en una variable temporal como los utilizados con microprocesadores son los de simple y doble rampa así como los de aproximaciones sucesivas. En la Figura 4.7 se muestra el esquema de este tipo de convertidores.

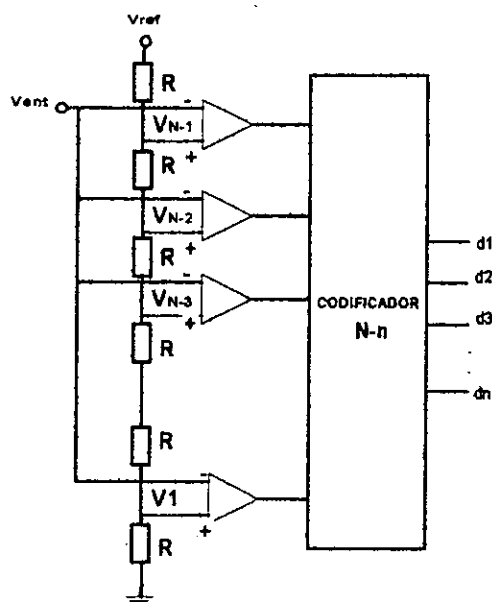


Figura 4.7: Convertidor Analógico-Digital Paralelo

El convertidor A/D de simple rampa integra una tensión de referencia hasta que la rampa generada alcanza un nivel igual a la tensión de entrada.

Durante este tiempo un contador cuenta los impulsos generados por un oscilador patrón. El resultado del contador es proporcional a la tensión analógica de entrada, por lo que, escogiendo adecuadamente la frecuencia del oscilador y la tensión de referencia, se obtiene directamente la salida digitalizada, como se muestra en Figura 4.8.

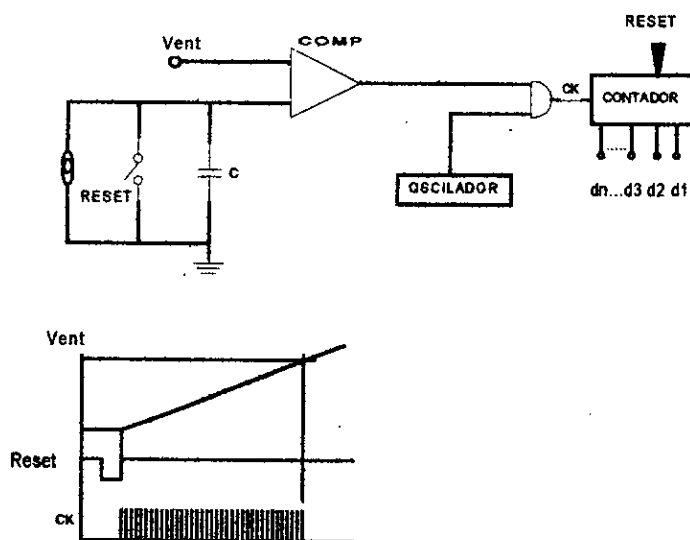


Figura 4.8: Convertidor A/D de Simple Rampa

El tiempo de conversión es largo y la precisión que se puede obtener no es muy buena debido a las variaciones de la capacidad C , de la corriente I_d y de la frecuencia del oscilador a lo largo del tiempo.

El convertidor A/D de doble rampa soluciona en gran medida los condicionamientos de precisión anteriores debido a que al realizar dos rampas consecutivas se compensan los errores producidos por las derivadas de la capacidad y la frecuencia. La primera rampa se realiza integrando la entrada durante un tiempo fijo, la segunda rampa se obtiene integrando una tensión de referencia. El tiempo de integración de la segunda rampa depende del nivel que se ha conseguido durante la primera integración por lo que durante este tiempo un contador cuenta los impulsos de un oscilador patrón. El tiempo máximo de conversión es el necesario para contar $(2)(2^n)$ impulsos de reloj siendo n el número de bits de la salida digital.

El convertidor A/D de aproximaciones sucesivas hace una optimización del tiempo de conversión mediante la realización de un algoritmo de aproximación. Utiliza un convertidor D/A, un comparador y un registro de aproximaciones sucesivas (SAR, Successive Approximation Register).

El algoritmo consiste en la aproximación bit por bit de la salida empezando por el bit de mayor peso (MSB), que corresponde a la mitad de la máxima excursión de tensión:

$$V_{ref}(1/2) + V_{ent}.$$

Este bit valdrá 1 ó 0 si el nivel de la señal de entrada está por encima ó por debajo de este valor. Para realizar la comparación de tensión de entrada con una tensión que vale $\delta_1 \cdot V_{ref} \cdot 1/2 + V_{ref} \cdot 1/4$, siendo δ_1 el valor obtenido en el primer bit; se asignará a δ_2 el valor 1 ó 0 con el mismo criterio anterior para seguir

realizando el algoritmo de aproximación hasta llegar al bit de menor peso (LSB). El número de iteraciones es igual al número de bits del convertidor. Este método es conceptualmente muy sencillo, ya que permite realizar la conversión con rapidez, no exige una realización muy costosa y su tiempo de conversión es del mismo orden que el tiempo de ciclo de los microprocesadores estándar.

Existen otros convertidores A/D que por diversos motivos (costo, fiabilidad o aplicación específica) no son ampliamente utilizados.

El convertidor A/D de descarga presenta una estructura tal que la carga resultante de la entrada se transfiere a un condensador; a continuación este se descarga a corriente constante al mismo tiempo que un contador acumula los impulsos procedentes de un oscilador patrón, el resultado del cual es el valor de la medida.

El convertidor de triple rampa tiene una estructura y algunas características en cuanto a fiabilidad, estabilidad, similares al de doble rampa pero consigue aumentar la velocidad de conversión al realizar la integración de referencia en dos etapas: en la primera etapa aproxima los bits de mayor peso y a continuación realiza la aproximación de los bits de menor peso. El convertidor de seguimiento (tracking) o modulador delta va resiguiendo constantemente la señal de entrada mediante incrementos o decrementos consecutivos del valor digital (modulación delta). El control del signo de los incrementos consecutivos se realiza en función de la respuesta de un comparador que tiene en sus entradas, la señal a digitalizar y la señal analógica correspondiente a la señal digitalizada, procedente de un convertidor digital-

analógico que tiene incorporado. La respuesta de este convertidor es rápida siempre y cuando las variaciones de la señal analógica de entrada sean pequeñas.

El convertidor tensión-frecuencia consiste en la conversión de la señal de tensión de entrada en una frecuencia proporcional a dicha tensión, que posteriormente se convierte en digital mediante la acumulación de los impulsos en un contador que tiene una ventana abierta durante un tiempo fijo. Es especialmente utilizado para la adquisición remota de datos y para sistemas en los que se desea aislar la sección analógica de entrada y la sección digital, pues sólo es necesario una señal de comunicación entre las partes analógica (remota) y la digital.

4.4 Convertidores Digital-Analógico

Un convertidor digital-analógico (D/A) es un dispositivo que recibe información digital en forma de una palabra de n bits, y la transforma en una señal analógica. La transformación se realiza mediante una correspondencia entre 2^n combinaciones binarias posibles en la entrada y 2^n tensiones (o corrientes) discretas obtenidas a partir de la tensión de referencia V_{ref} . La señal analógica así obtenida no es una señal continua, sino que se obtiene un número discreto de escalones a consecuencia de la discretización de la entrada. El proceso consiste en hacer una ponderación de cada bit y sumar el resultado de todas las ponderaciones multiplicadas por su valor (0 o 1).

$$V_{sal} = \sum_{i=1}^n V_{pond,i} \cdot \delta_i \cdot \delta = \{0,1\}$$

La ecuación anterior sugiere una realización inmediata: el convertidor digital-analógico de resistencias ponderadas. Este convertidor utiliza un sumador de n canales, siendo n el número de bits, seleccionándose cada canal o no, según el valor del bit correspondiente (0 ó 1). En el caso de un convertidor lineal, el valor de la resistencia de cada canal $R = 2^i \cdot R$.

$$V_{sal} = -R \sum_{i=1}^n \delta_i \cdot \frac{V_{ref}}{2^i R} = -V_{ref} \sum_{i=1}^n \delta_i \cdot \frac{1}{2^i}$$

Este circuito tiene el inconveniente de que necesita resistencias de gran precisión de valores distintos. Al mismo tiempo, los conmutadores de canal deben tener unos valores de resistencia de conducción y de bloqueo en unos márgenes suficientemente alejados de los valores de las resistencias del circuito para que no interfieran en la conversión. En la Figura 4.9 se muestra el diagrama correspondiente.

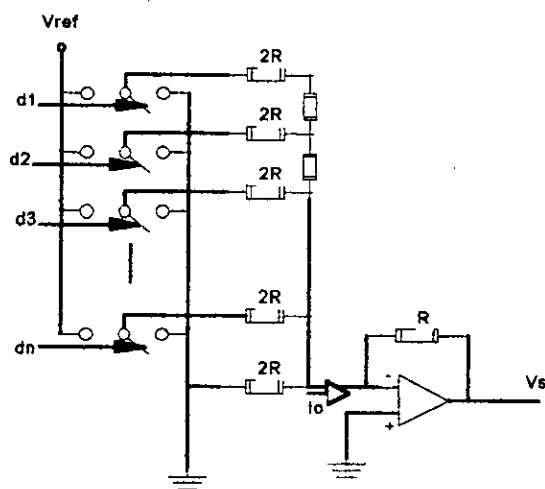


Figura 4.9: Convertidor D/A de Resistencias Ponderadas

El convertidor digital-analógico por red de resistencias $R-2R$, soluciona los problemas expuestos anteriormente al utilizar resistencias con sólo dos valores; la corriente que circula por las resistencias con sólo dos valores; la corriente que circula por las resistencias $2R$, está en progresión geométrica de razón $1/2$ debido a que en cada punto de unión de las resistencias $R-R$, la impedancia del circuito es igualmente R . Se obtiene una corriente a la entrada del amplificador operacional suma de las corrientes de cada rama, la cual produce una tensión analógica de salida V_{sal} .

El convertidor digital-analógico por modulación de anchura de impulso es un convertidor que en una primera etapa convierte el código digital en una señal pulsante de anchura controlada y en una segunda etapa y mediante un filtro de paso bajo, convierte esta señal pulsante en una señal analógica. En la Figura 4.10 se puede observar la estructura de este convertidor:

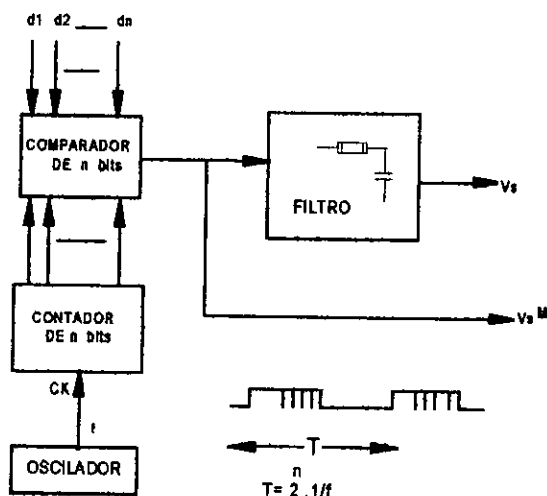


Figura 4.10: Convertidor Digital-Analógico de Modulación de Anchura de Impulso

Un comparador de n bits compara el código digital con el valor de un contador de n bits que se va incrementando con los impulsos de un oscilador patrón. El período de la señal modulada es por lo tanto, el tiempo necesario para contar 2^n impulsos, mientras que la anchura de la señal modulada es igual al tiempo necesario para contar tantos impulsos como indica la palabra digital de entrada. El filtrado de salida implica una velocidad de conversión lenta, pero este convertidor es estructuralmente monotónico.

4.5 Circuitos de Captura y Mantenimiento

Los circuitos de captura y mantenimiento (Sample & Hold) son elementos analógicos equivalentes a los registros capturadores (latches) digitales. Se utilizan para muestrear una señal analógica y mantener el nivel de dicha señal en un medio de almacenamiento (normalmente un condensador) para proceder a su medida o a su procesamiento en un sistema.

El funcionamiento de un circuito de captura y mantenimiento ideal muestra que éste sigue la entrada durante el período de muestreo y luego mantiene el último valor analógico de tensión, cuando conmuta al modo de mantenimiento. Tanto el tiempo de captura como el tiempo de mantenimiento dependen del valor del condensador de almacenamiento. Si se desea obtener un circuito S & H que tenga un tiempo de captura muy pequeño deberá utilizarse un condensador de baja capacidad, mientras si se desea que la señal capturada se mantenga durante un largo tiempo, deberá utilizarse un condensador grande.

4.6 Multiplexores Analógicos

Un multiplexor analógico es un dispositivo que tiene un conjunto de entradas, una salida y una serie de conmutadores analógicos direccionables que van seleccionando aleatoriamente o secuencialmente cada una de las entradas y las conecta a la salida. En cada instante sólo se cierra un conmutador y la secuencia de muestreo se realiza mediante un circuito de control o por la microcomputadora directamente.

Un multiplexor ideal presenta un tiempo de conmutación de cada canal y una impedancia de paso nulos, no introduce distorsión en la señal y cuando un canal está abierto, hay aislamiento completo entre su entrada y la salida. Los conmutadores analógicos están basados en dispositivos de efecto de campo. Los multiplexores reales presentan una capacidad parásita de entrada, CE que contribuye a limitar la anchura de banda de funcionamiento, como se muestra en la Figura 4.11.

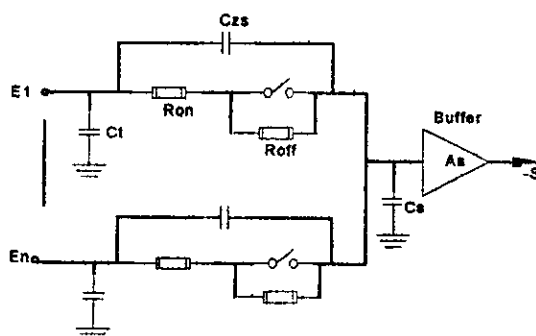


Figura 4.11: Circuito Equivalente de un Multiplexor Analógico

Cuando un conmutador está abierto, presenta una impedancia de paso muy elevada (del orden de 10^3 a $10^6 \Omega$) mientras que cuando está cerrado, es de

algunos pocos ohmios (de 30 a 150 Ω típicamente). Según sea la configuración utilizada, los multiplexores pueden conmutar señales unipolares o diferenciales. El modo unipolar se utiliza en aplicaciones en las que el nivel es significativamente superior a las tensiones en modo común presentes en el sistema.

Cuando las medidas son de bajo nivel y puede sumarse ruido en modo común, la entrada diferencial es la más adecuada, (como se muestra en la Figura 4.12) debido a la utilización de la capacidad de rechazo en modo común (CMR) del amplificador, aunque se reduce a la mitad la capacidad de canales de entrada.

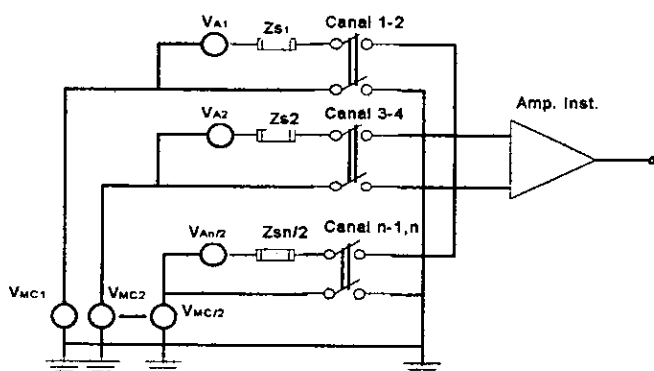


Figura 4.12: Multiplexor Analógico en Modo Diferencial

La configuración cuasidiferencial (Figura 4.13) permite mantener la capacidad total de canales de entrada y al mismo tiempo mantiene el rechazo de la señal en modo común, pero sólo se puede utilizar en los casos en que las señales se obtienen en unas condiciones de ruido comunes.

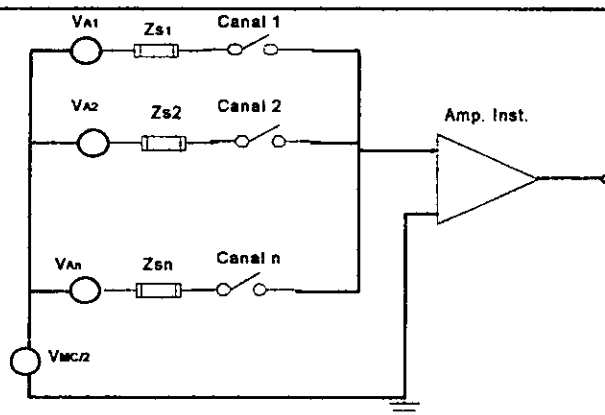


Figura 4.13: Multiplexor Analógico en Modo Cuasidiferencial

La utilización de un amplificador de instrumentación a la alta impedancia de entrada (superior a 10^8 ohmios) colocado a la salida del multiplexor permite minimizar los errores a las impedancias finitas de conducción de los canales del selector analógico.

4.7 Interface entre Convertidores y Microcomputadoras

La interface entre un convertidor y una microcomputadora se realiza mediante una interconexión multifacética que contiene tanto hardware como software. Al mismo tiempo, la interface utiliza señales de control, de direccionamiento y de datos.

Las señales de direccionamiento seleccionan el convertidor o el canal de entrada o el de salida en sistemas multicanal. Las señales de control son las que

inician la conversión, señalan el fin de conversión, indican si el convertidor está ocupado mientras que las transferencias de información entre el microprocesador y el convertidor se realiza mediante las señales de datos.

La transferencia de datos entre un microprocesador y un convertidor analógico-digital puede estar controlada por el micro o por el convertidor. En el primer caso, los datos se transfieren entre un registro asociado al convertidor y el registro acumulador u otro registro interno del microprocesador, según se utilice una instrucción de E/S o la transferencia sea mapeada en memoria. En uno y otro caso, el microprocesador debe esperar, bien sea mediante un bucle de espera o en estado de WAIT, a que el convertidor acabe la conversión para entrar el dato válido. Cuando es el convertidor A/D el que inicia la transferencia de datos se utilizan los métodos de transferencia por interrupción, por acceso directo a memoria (Direct Memory Access, DMA) o mediante memoria auxiliar.

Transferencia por interrupción: El microprocesador selecciona el canal de entrada, inicia el proceso de conversión y luego prosigue con su tarea. Al acabarse la conversión, la señal EOC (End of Conversion) interrumpe al microprocesador. Si la interrupción es vectorizada, el convertidor envía la dirección de la rutina de servicio, mientras que si no es vectorizada, el microprocesador debe examinar todos los convertidores hasta saber cuál ha sido el que la ha enviado, para poder así servirlo. Esta transferencia no necesita una circuitería compleja, es útil cuando los convertidores presentan un tiempo de conversión elevado, pero el programa interruptivo debe estar alojado en la memoria.

La transferencia por acceso directo a memoria permite al convertidor su funcionamiento continuo. Después de cada conversión, la señal EOC inicia la transferencia de la palabra convertida a la memoria principal. Pero sólo puede realizarse la transferencia durante el tiempo en que el bus de datos y de direcciones en el microprocesador no está utilizado, o sea provocando el estado de retención (Hold) del microprocesador, o por medio del robo del ciclo. Este método es especialmente útil cuando hay gran cantidad de convertidores; se necesita un controlador de bus, aumentándose la complejidad de la circuitería necesaria, pero no se pierde tiempo de procesador ni se necesita programa de gestión de la conversión, puesto que el conversor está en funcionamiento continuo o sea que al acabar una conversión y después de almacenarla en la memoria, vuelve a autoiniciarse la conversión.

La transferencia mediante memoria auxiliar es similar al caso anterior, pero los datos obtenidos se guardan en una memoria auxiliar y el procesador los lee de allí. Se introducen retrasos (incluso de un periodo de iteración) debido a que las señales que se memorizan son muestreadas sólo una vez en cada periodo de iteración.

La transferencia entre un microprocesador y un convertidor digital analógico se inicia mediante el envío de la palabra o palabras digitales (en caso de que el convertidor sea de mayor número de bits que el bus del microprocesador), al registro del convertidor. Puede utilizarse también la estructura de mapeado de memoria ya que el registro se comportará como una (o varias) posiciones de memoria de sólo escritura, permitiendo así una programación sencilla y más potente, pues se podrán utilizar todas las instrucciones de referencia a memoria para realizar la transferencia al convertidor.

Según sea la estructura utilizada, habrá un mayor peso en la componente de hardware o en la de software. Son de uso bastante común las tarjetas preparadas para interconectar en el bus estándar de algunas microcomputadoras, que incluyen un sistema de adquisición de datos, un conjunto de convertidores y la parte que asegura la compatibilidad entre las alimentaciones y niveles lógicos de las señales digitales que se intercambian. Los convertidores discretos directamente acoplables al bus de los microprocesadores acostumbran a utilizar el estado de alta impedancia en el registro de salida. Pueden ser monolíticos o híbridos, consiguiéndose en los primeros una gran flexibilidad de aplicación, mientras que los híbridos suelen tener unas prestaciones superiores en cuanto a calidad y precisión.

4.8 Realización de Conversiones por Programa

En aplicaciones donde la velocidad de conversión no es muy importante y además, el microprocesador no está demasiado ocupado, éste puede tomar parte en la realización de conversiones controladas por programa.

Aunque hay varios caminos posibles para realizar la conversión A/D por software, la técnica de aproximaciones sucesivas parece ser la más adecuada en la mayoría de los casos, ya que ofrece una velocidad de conversión razonable utilizando un mínimo de circuitería adicional, consistente en un convertidor D/A y un comparador.

Es conveniente la utilización de un circuito de captura y mantenimiento que congele la señal de entrada mientras se realiza la conversión. El algoritmo de conversión puede ocupar entre 20 y 40 palabras de memoria, dependiendo sobre todo, de la facilidad del microprocesador concreto, para testar bits. El comparador, compara la señal de entrada congelada con la salida del convertidor digital analógico y su resultado (0 si es menor, 1 si es mayor) retorna al procesador a través de un "port" o a través de un bit de prueba. El microprocesador va enviando al convertidor D/A las palabras de aproximación, hasta que la conversión se ha realizado.

La incorporación de posibilidades adicionales al circuito básico, tales como autoescalado, salida analógica, corrección automática de derivas, detección de max-min, etc. no exige grandes modificaciones ni en el diseño de la circuitería ni en el programa de gestión.

Capítulo 5:

Transmisión y Recepción de Información

A través del tiempo se han desarrollado muchas formas de comunicarse remotamente entre las personas. Desde los primeros inventos, como el telégrafo, hasta los modernos satélites se han mejorado muchas cosas. Uno de los puntos que más se ha desarrollado es que antes se requería forzosamente de señales eléctricas que viajaran a través de cables que no podían conectar dos dispositivos a gran distancia a menos que fuera a través de repetidoras. Ahora, gracias a la tecnología uno puede comunicarse a grandes distancias sin necesidad de utilizar forzosamente señales eléctricas, ni cables y con gran seguridad de que lleguen íntegras a su destino. Aunque se sigan utilizando repetidoras para enlazar dos puntos distantes, éstas se han desarrollado enormemente, como por ejemplo el uso de un satélite para que la información enviada a miles de kilómetros distantes, llegue en cuestión de segundos.

El esquema básico de comunicación sigue siendo el mismo: la fuente de información, el transmisor, el receptor y el canal de comunicación. Las fuentes de Información se integran, ya sea por señales sonoras (voz, música, etc.), señales visuales (fotografía, textos, etc.), señales almacenadas (unidades magnéticas, discos, etc.),

señales físicas (temperatura, calor, ozono, etc.). El transmisor, ejecuta las funciones de amplificación, filtrado y modulación. En el canal de comunicaciones se producen agentes adversos como ruido, interferencia y distorsión. El receptor, se producen las funciones de amplificación, filtrado y demodulación.

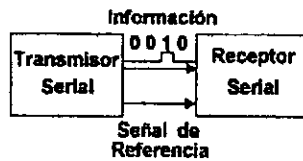
En este capítulo se dará una descripción de los modos de comunicación, las formas y medios de comunicación que se utilizan actualmente, las formas de modulación y las dificultades que se presentan al recibir un mensaje.

5.1 Transmisión de Datos en Serie y en Paralelo

En la Transmisión Serial, cada símbolo se transmite consecutivamente, uno después de otro. Por ejemplo el número 0010 debe ser transmitido en secuencia, con quizás una pequeña pausa entre cada bit. En la Figura 5.1 se muestra el Modo de Transmisión Serial.

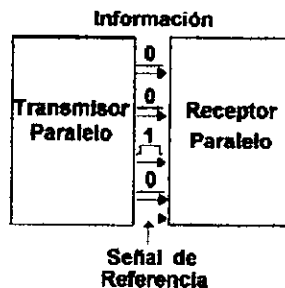
En la transmisión en paralelo, la información se envía simultáneamente. Para hacer esto se requiere de varios canales de transmisión en paralelo. Por ejemplo, para transmitir el número 0010, se deben separar cada bit para transmitirse. En la Figura 5.2 se muestra el Modo de Transmisión Paralelo.

Cabe resaltar, que la transmisión en serie requiere un solo canal de transmisión, mientras que la transmisión en paralelo requiere de varios canales de transmisión. La transmisión en paralelo permite que la información sea transmitida en mucho menor tiempo, pero en cambio se requieren muchos canales de transmisión.

*Figura 5.1: Transmisión Serial*

Las transmisiones en serie las realiza una red telefónica, un modem serial (transmitiendo bits uno después de otro), entre otros. La transmisión serial se utiliza para la línea telefónica y para la conexión de ésta a la computadora. La transmisión en paralelo se utiliza entre varios de los componentes internos de una computadora. El medio de transmitir datos dentro de la computadora es a través de los buses, los cuales llevan los bits para formar una palabra binaria.

Para poder convertir datos en serie a datos en paralelo se requiere de una interface de entrada/salida con la computadora conocido como Transmisor-Receptor Universal Asíncrono (UART). El UART es un circuito que convierte la información serial a paralelo para poder hacer una transmisión asíncrona.

*Figura 5.2: Transmisión en Paralelo*

5.2 Modos de Comunicación

Existen básicamente tres modos de comunicación: Modo Simplex, Modo Dúplex y Modo Semi-dúplex. Estas se refieren a la direccionalidad de comunicación entre dos puntos de comunicación.

5.2.1 Modo Simplex

La Comunicación en modo Simplex es una ruta de comunicación que habilita sólo un camino, es decir, que la transmisión trabaja solamente en una dirección todo el tiempo. Por ejemplo, la transmisión de una señal de televisión, donde la estación televisora transmite su información pero no hay manera de que el público receptor a través de sus televisiones, pueda enviar información por el mismo medio de transmisión a la estación televisora.

5.2.2 Modo Dúplex

La Comunicación en modo Dúplex es una ruta de comunicación que habilita dos caminos simultáneamente, es decir que en este modo se permite la transmisión y recepción al mismo tiempo. Un circuito de comunicaciones que permite simultáneamente la comunicación en dos caminos, debe ser un dispositivo Dúplex Completo. Un ejemplo de circuito de transmisión Dúplex es la red de telefonía de voz, en donde tanto el emisor como el receptor pueden entablar una comunicación sin necesidad de utilizar otra vía de comunicación, ni intercalar los tiempos de conversación.

5.2.3 Modo Semi-Dúplex

El modo de Comunicación Semi-Dúplex se da cuando la comunicación de dos lugares se obtiene invirtiendo la dirección de comunicación en un circuito, esto es que la comunicación de dos lugares tiene que ser switchheada o intercalada. Un ejemplo de este modo de transmisión lo tenemos en la comunicación de las aeronaves con la torre de control o en la radio de banda citadina (CD), en los cuales para hablar se requiere hacer un switcheo en los aparatos, es decir, primero una persona (Transmisor) debe transmitir su información mientras que la otra persona recibe (Receptor) la información. Posteriormente hacen un cambio de papeles donde ahora el transmisor se convierte en receptor y el receptor se convierte en transmisor. Un Transmisor no puede ser Receptor al mismo tiempo, ni un Receptor puede ser Transmisor al mismo tiempo.

5.3 Modos de Transmisión

Para que una transmisión sea correctamente decodificada, los caracteres recibidos, por ejemplo de un modem, deben saber cuando ha empezado una transmisión, para que sus circuitos de reloj interno puedan ser sincronizados apropiadamente con los datos transmitidos. Esta sincronización debe hacerse, ya que sin ella, no habría manera de identificar el inicio y fin de una señal transmitida, ni mucho menos podría hacer una buena identificación de bits. Además la computadora debe prepararse para poder recibir datos. Si no estuviera preparada la computadora para empezar a recibir información, ésta no sería capaz de identificar por sí misma, en qué momento atiende el puerto de recepción de información.

5.3.1 Transmisión Asíncrona

En la Transmisión Asíncrona los intervalos de tiempo entre los caracteres transmitidos deben ser de diferente longitud. Los bits de comienzo y final son transmitidos antes y después de cada caracter para sincronizar al reloj receptor. En la Figura 5.3, se muestra un ejemplo de modo de transmisión Asíncrona.

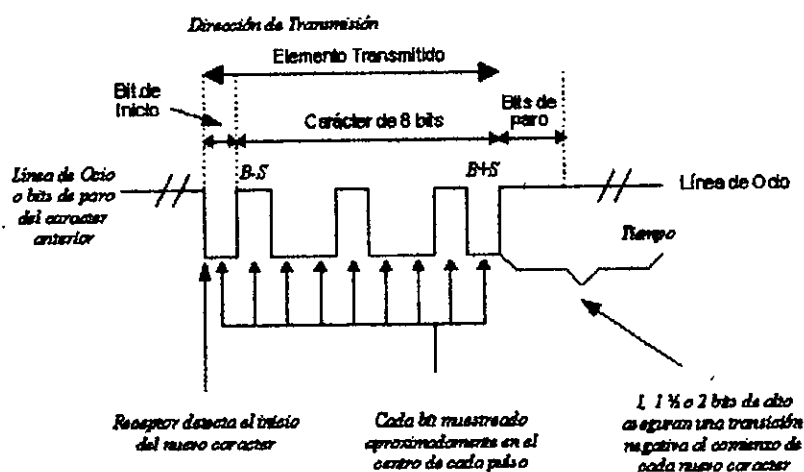


Figura 5.3: Modo de Transmisión Asíncrona

5.3.2 Transmisión Síncrona

La Transmisión Síncrona es el proceso por el cual los caracteres de datos se transmiten en un formato, con el transmisor y receptor estando en sincronía. No hay bits de comienzo, ni de final, por lo que esta técnica de transmisión es mucho más eficiente que la Transmisión Asíncrona. La Transmisión Síncrona es la más usada, por ser más segura en la integridad de la información. En la Figura 5.4, se muestra un ejemplo de modo de transmisión Síncrona.

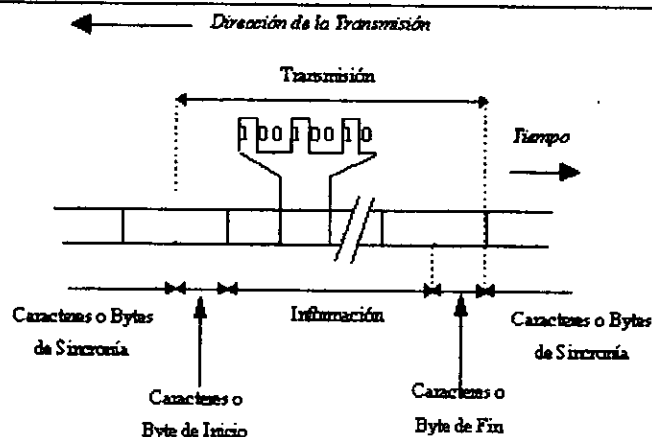


Figura 5.4: Modo de Transmisión Sincrónica

5.4 Tasas de Transmisión: Bits/seg y Bauds/seg

Existen dos formas de medir la cantidad de información digital que se puede transmitir en cierto intervalo de tiempo, generalmente en segundos. A esto se conoce como Tasa de Transmisión y se puede hacer de dos formas en bits por segundos y Bauds por segundos.

5.4.1 Tasa de Transmisión de Bits/Seg

La Tasa de Transmisión de Bits/Seg. es la velocidad a la cual se transmiten y/o reciben datos. Esta tasa de velocidad de Bits es expresada en bits por segundo (bps o b/s). Una buena Tasa de Transmisión puede ser de 10 caracteres por segundo. Si once bits se utilizan para cada caracter, esto corresponde a una velocidad de transmisión de datos de 110 b/s.

5.4.2 Tasa de Transmisión de Bauds/Seg

La tasa de transmisión Bauds es otro término que se utiliza para describir la velocidad de transmisión. Se refiere a la velocidad a la cual los elementos de la señal usada para representar bits son transmitidos. Un elemento de la señal es una pequeña onda de un periodo en la cual la amplitud, la frecuencia o la fase pueden o no pueden representar un bit. Por ejemplo si el elemento de la señal puede tomar una de dos amplitudes, entonces un "1" o un "0" se puede codificar, de acuerdo al valor de la amplitud. Suponiendo que cada elemento de la señal puede tomar una de cuatro amplitudes, entonces cada amplitud puede corresponder a un par específico de dos bits; si hay cuatro amplitudes, entonces hay cuatro posibles combinaciones de bits (00, 01, 10, 11). En este caso, la tasa de velocidad de bits debe ser dos veces la tasa de velocidad Baud. Este tipo de codificación se usa para tasas de bits mayores a 1200 b/s.

La tasa de velocidad de bits se utilizan por usuarios de computadoras, mientras que la tasa de velocidad de Baud la manejan los especialistas en telecomunicaciones.

5.5 Medios de Comunicación

El medio de Comunicación es la forma de enlazar dos dispositivos de comunicación para el intercambio de información. Dichos dispositivos pueden estar distantes o relativamente cerca entre sí. El medio de transmisión puede desde un simple cable hasta señales electromagnéticas que viajan por la atmósfera por medio de antenas receptoras y transmisoras. A continuación se dará alguna explicación de las características de algunos medios de comunicación.

5.5.1 Líneas de Dos Alambres

En una Línea de dos Alambres cada alambre está aislado uno del otro con un material dieléctrico. Este tipo de línea es utilizada para conectar equipos de menos de 50 m. de distancia uno del otro, siempre y cuando se utilice una de tasa de transmisión adecuada (menos de 19.2 Kbps). La señal (generalmente un nivel de voltaje o corriente relativo con respecto a un nivel de referencia) se aplica en un alambre mientras que en el otro se aplica el nivel de referencia (generalmente tierra).

Bajo el mismo principio de la línea de dos alambres se puede hacer una conexión que generalmente ocupa muchos alambres en donde para cada alambre hay una señal conservando un alambre para el nivel de referencia común (tierra). Estos alambres son conjuntados en un solo cable conocido como cable plano. En la Figura 5.5 se muestra este tipo de cable.

Con este tipo de línea se debe tener cuidado para impedir el acoplamiento cruzado de señales eléctricas entre alambres adyacentes en el mismo cable. Esto es conocido como interferencia y se debe al acoplamiento capacitivo entre los dos alambres. Además su estructura lo hace susceptible a la intromisión de señales de ruido de otras fuentes de señales eléctricas provocados por radiación electromagnética. El principal problema con la interferencia de señales es que ellas se generan en un solo alambre (por ejemplo, el alambre de la señal) creando una diferencia de señal adicional entre los dos alambres. Ya que el receptor opera normalmente usando las diferencia de señales entre los dos alambres, ésta puede provocar una interpretación errónea de la señal combinada recibida (señal más ruido).

Todos esos factores contribuyen a las longitudes limitadas de la línea y las tasas de transmisión de bits que se pueden usar.

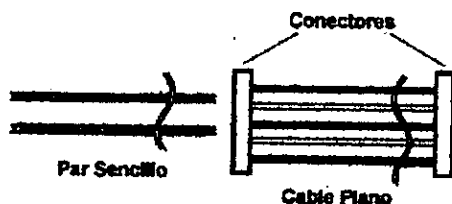


Figura 5.5: Líneas de dos alambres

5.5.2 Líneas de Par Trenzado

Las líneas de Par Trenzado son mucho más inmunes al ruido y consisten de un par de alambres que están entrelazados (trenzados) entre sí. La proximidad del alambre de la señal y el de referencia significa que alguna señal de interferencia se recoge por ambos alambres reduciendo su efecto al hacer la diferencia de señales. Además, si muchos pares trenzados se conjuntan bajo el mismo cable, el trenzado de cada par con el cable reduce aún más la interferencia. Un esquema de una línea de par trenzado se muestra en la Figura 5.6.

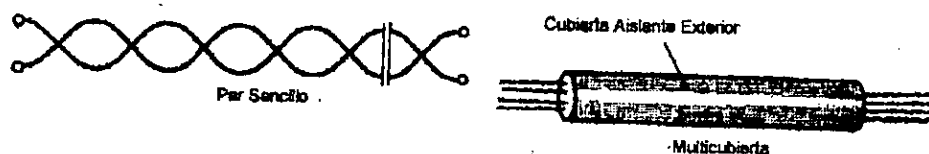


Figura 5.6: Par Trenzado sin Blindaje

Las líneas de par trenzado son las más recomendadas para tasas de transmisión de bits del orden de 1 Mbps en distancias cortas (menos de 100 m) y tasas de transmisión de bits más bajas para distancias más largas. Estas líneas son conocidas

como Pares Trenzados sin Blindaje y se utilizan ampliamente en las redes telefónicas. Con varios cables de par trenzado formando una malla blindada, se pueden reducir aun más los efectos de interferencias. Estos últimos cables se conocen como Pares Trenzados Blindados y se muestran en la Figura 5.7.

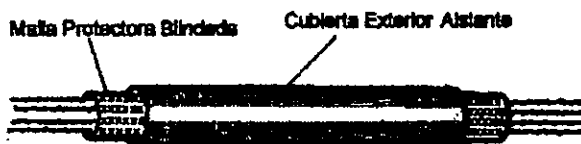


Figura 5.7: Par Trenzado Blindado

Un factor limitante de las líneas de par trenzado es lo que se conoce como efecto de despelleje, esto es que cuando se incrementa la tasa de transmisión de bits (y frecuencia) de la señal transmitida, el flujo de corriente en los alambres tiende sólo hacia afuera de la superficie de el alambre, de este modo se usa menos la sección de cruce permitido. El incremento de resistencia eléctrica de los alambres para señales de frecuencia altas permite una atenuación más alta. Además a frecuencias altas, de más de 1 MBps, la potencia de señal se pierde más por los efectos de radiación.

5.5.3 Cable Coaxial

En la Figura 5.8 se muestra que los alambres de la señal y referencia toman la forma de un conductor de centro sólido corriendo concéntricamente (coaxialmente) dentro de un conductor circular sólido exterior (o trenzado). Idealmente el espacio entre los dos conductores deberían de ser llenados con aire, pero en la práctica esto es llenado normalmente con un dieléctrico aislando el material con un sólido o una estructura de panal.

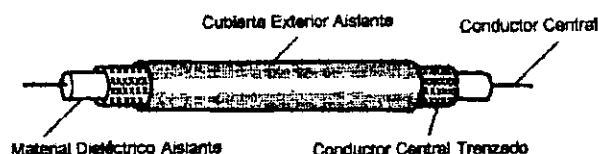


Figura 5.8: Cable Coaxial

El centro del conductor aislado de las señales de interferencia externas y además sólo hay las pérdidas mínimas como resultado de la radiación electromagnética y el efecto de despelleje. El cable coaxial se puede usar con diferentes tipos de señales, pero generalmente con señales digitales arriba de 10 Mbps para distancias de algunos cientos de metros (o más grandes con la modulación). La geometría del cable coaxial reduce considerablemente los distintos efectos limitantes de transmisión que es, la frecuencia de señal máxima, y por lo tanto las tasas de transmisión de información, que se pueden transmitir usando un conductor sólido (generalmente cobre).

5.5.4 Fibra Óptica

El cable de Fibra Óptica difiere de los anteriores, ya que su información se transmite en forma de rayos fluctuantes de luz en una fibra de vidrio y no como una señal eléctrica en un alambre. Las ondas de luz tienen mucho más ancho de banda que las señales eléctricas permitiendo al cable de Fibra Óptica alcanzar tasas de transmisión de cientos de megabits por segundo. Además las ondas de luz son inmunes a las interferencias electromagnéticas. Consecuentemente el cable de Fibra

Óptica es extremadamente utilizado para la transmisión digital de información con tasas de transmisión de bits más bajas en entornos ruidosos eléctricamente. También se utiliza el cable de Fibra Óptica, porque ya que es difícil intervenirlo físicamente. Un cable de fibra óptica consiste de una sola fibra de vidrio para cada señal a ser transmitida, contenida en una capa protectora, la cual también blindaje a la fibra de cualquier luz externa. Un diagrama de este tipo de cable se muestra en la Figura 5.9.

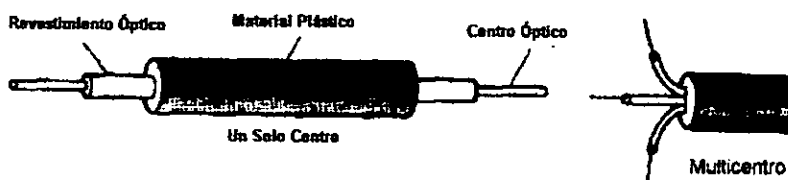


Figura 5.9: Cables de Fibra Óptica

La forma de generar la señal de luz es por medio de un transmisor óptico el cual convierte las señales eléctricas normales, como se usa en un DTE (Equipo Terminal de Datos que puede ser una computadora digital, una impresora, un teclado, un ratón, etc., cuya función es producir o recibir una señal digital que representa cierta información). Similarmente un receptor óptico se usa para convertir las señales ópticas a señales eléctricas. Generalmente el transmisor utiliza un Diodo Emisor de Luz (LED) o un Diodo de Inyección Láser (ILD) para hacer la operación de conversión, mientras el receptor utiliza un Foto-Diodo sensible a la luz o un Foto-Transistor. La fibra consiste de dos partes: el núcleo de vidrio y un recubrimiento de vidrio con un índice refractivo muy bajo. La propagación de luz a lo largo del núcleo de fibra óptica en uno de tres caminos depende del tipo y anchura del material usado.

5.5.5 Luz Infrarroja

La Transmisión de datos por medio de la luz infrarroja, se utiliza poco, debido a que los dispositivos a comunicar deben estar cercanos uno del otro (por lo general en un mismo cuarto), ya que como el medio de transmisión es la luz infrarroja cualquier obstáculo en el viaje de ésta, hará que la información se distorsione o simplemente se pierda. Para distancias muy cercanas, de menos de cinco metros, es seguro, además de que es el medio de transmisión mas económico. La transmisión de datos por medio de la luz infrarroja se utiliza más frecuente en controles remotos para el funcionamiento de televisiones, radios, calculadoras al transferir programas, etc..

Al utilizar la transmisión por medio de luz infrarroja, se requiere de un diodo emisor de luz infrarroja como transmisor y un fototransistor de luz infrarroja como receptor. Los diodos emisores de luz infrarroja tiene el mismo principio de funcionamiento de un diodo emisor de luz normal. Existen en el mercado una gran variedad de fotoreceptores de luz infrarroja, incluso generalmente vienen integrados con un amplificador y un demodulador. Las tasas de transmisión de bits que se pueden lograr con este medio son muy bajas, ya que no se pueden utilizar frecuencias muy altas porque el espectro de la luz infrarroja no lo permite.

5.5.6 Radio

La transmisión de radio de bajas frecuencias se utiliza mucho en lugar de la comunicación vía cables, además se usan para aplicaciones que requieren una amplia cobertura de área hacia una alta densidad de usuarios. La cobertura de área de cada transmisor es restringida (por límites de potencia de salida) por lo que ésta

proporciona sólo canales suficientes para soportar la carga total en esa área. La cobertura amplia se alcanza organizando muchos transmisores de radio, en donde cada transmisor opera usando una banda diferente de frecuencias de sus vecinos. Como el campo de cobertura de cada transmisor es limitada, es posible volver a usar su banda de frecuencia en otras partes de la red. Normalmente sus tasas de transmisión de bits son de aproximadamente de 10 Kbits/seg, mucho más baja que con cableado fijo.

5.5.7 Microondas

Las comunicaciones con microondas se usan frecuentemente cuando los medios de transmisión físicos no son prácticos para instalar o su instalación es muy cara, por ejemplo a través de un río o quizá un pantano o desierto. Como los rayos de microondas colimados viajan a través de la atmósfera terrestre, ésta puede ser interferida por factores tales como edificios y condiciones climáticas adversas. Por otro lado, con una comunicación satelital, los rayos viajan principalmente a través del espacio y es por lo tanto menos propenso a tales efectos. Sin embargo, las comunicaciones de microondas directas a través de la atmósfera terrestre se pueden usar relativamente sobre distancias de más de 50 Km.

5.5.8 Satelital

En la Transmisión satelital, los datos se transmiten por medio de ondas electromagnéticas a través del espacio. Por medio de un rayo de microondas se envía información que se transmite a un satélite desde la tierra. Este rayo se recibe en el satélite y se retransmite hacia destinos predeterminados utilizando una antena

direccional y un circuito a bordo conocido como transponder. Un solo satélite tiene muchos transponder, en donde cada uno convierte una banda de frecuencias en particular. Un canal de satélite típico tiene un gran ancho de banda (500 MHz) y puede proporcionar muy altas tasas de transmisión de bits, usando la técnica de multicanalización. La capacidad total del canal se divide en un número de subcanales, los cuales pueden soportar grandes tasas de transmisión de bits.

Los satélites usados para propósitos de comunicaciones son normalmente geoestacionarios, es decir que el satélite orbita a la tierra una vez cada 24 horas en sincronismo con la rotación de la tierra. La órbita del satélite se escoge para proporcionar una ruta de comunicación directa entre las estaciones transmisoras y las estaciones receptoras. El grado de la colimación de los rayos de microondas retransmitidos por el satélite puede ser grande (para que la señal se pueda recoger en un área geográfica muy amplia) o se puede enfocar para que la señal se pueda recoger sobre una área limitada. La potencia de la señal es muy grande, permitiendo a los receptores de diámetro muy pequeño (conocidos como Terminales de Apertura Muy Pequeña o VSAT), saber como usar las antenas parabólicas. Un sistema típico de satélites se muestra en la Figura 5.10. En esta Figura sólo se muestra una ruta de transmisión unidireccional. La ruta dúplex es la más usada. Los canales asociados con cada estación terrestre operan a diferentes frecuencias.

Otras configuraciones comunes involucran una estación terrestre central que se comunica con un número de estaciones terrestres VSAT distribuidas en diferentes regiones. Normalmente un DTE se conecta a cada VSAT y éste se puede comunicar con una computadora central, como se muestra en la Figura 5.11.

Normalmente las centrales emiten a todos las VSAT's en una sola frecuencia, mientras que en dirección inversa cada VSAT transmite a diferente frecuencia.

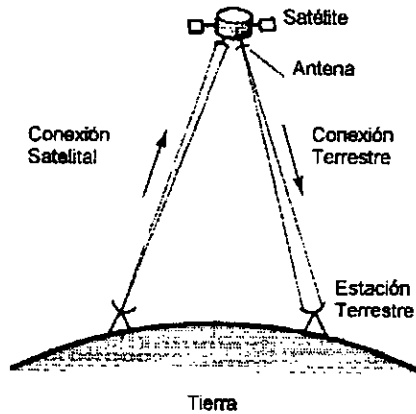


Figura 5.10: Transmisión Satelital punto a punto

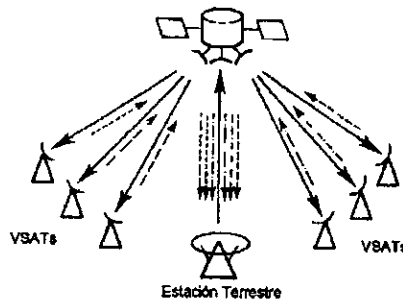


Figura 5.11: Transmisión Satelital Multipunto

5.6 Efectos Adversos en la Transmisión

Una señal transmitida siempre se afecta por la atenuación, el ancho de banda, la distorsión y el ruido. Todos ellos siempre están presentes y producen un efecto combinado. A continuación se describen cada uno de ellos.

5.6.1 Atenuación

La Atenuación de una señal se produce porque la amplitud de la señal transmitida se decrementa a lo largo del medio de transmisión. En la Figura 5.12 se muestra el efecto de Atenuación. Siempre se debe considerar la atenuación del medio de transmisión, por lo que se debe establecer un límite para la longitud del cable y asegurar así que el receptor pueda detectar e interpretar la señal atenuada recibida. Si el cable es muy largo, se requieren uno o más amplificadores (conocidos como repetidores) a lo largo del cable para restablecer la señal recibida su nivel original. Con los amplificadores la señal atenuada se incrementa en función de su frecuencia, ya que siempre una señal está compuesta de un rango de frecuencias, y por tal razón los amplificadores se diseñan para amplificar diferentes señales de frecuencia. Los ecualizadores se emplean para igualar la atenuación de una banda de frecuencias.

5.6.2 Ancho de Banda Limitado

Una señal transmitida se compone de un gran número de componentes de frecuencias, pero sólo se reciben los componentes que están dentro del ancho de banda del medio de transmisión. Esto confirma que la amplitud de cada componente de frecuencia hace disminuir la señal digital con el incremento de

frecuencia. En resumen, un ancho de banda grande del medio de transmisión hace pasar componentes de frecuencia grandes y por lo tanto se recibe una reproducción más fiel de la señal original. En la Figura 5.12 se muestra el efecto del Ancho de Banda Limitado. Para determinar la máxima tasa de información (datos) de un medio de transmisión en función de su ancho de banda se emplea la fórmula establecida por Nyquist

$$C = 2B \log_2 M \text{ bps}$$

donde C es la máxima tasa de transferencia de datos de una línea o sistema, $2B$ es la máxima tasa de datos (asumiendo que sólo existen dos niveles por elemento de señal y que un sistema o línea de transmisión tiene un ancho de banda de B Hertz) y M son los niveles por elemento de señal.

5.6.3 Distorsión

La tasa de propagación de una señal sinusoidal a lo largo de una línea de transmisión varía con respecto a la frecuencia de la señal. Consecuentemente cuando se transmite una señal, los distintos componentes de frecuencia hacen que la señal llegue al receptor con variaciones de retardo, resultando una distorsión de la señal recibida. Para una señal digital, la suma de los incrementos de la distorsión da como resultado que la tasa de bits de los datos transmitidos se incremente y algunos de los componentes de frecuencia asociados con cada transición de bits se retrasan y empiezan a interferir con otros componentes de frecuencia asociados con un bit de retardo. La distorsión se conoce también como interferencia intersimbólica y hacen que varíen los instantes de la transición de bits de la señal recibida. En la Figura 5.12 se muestra el efecto de Distorsión.

5.6.4 Ruido

Normalmente ante la ausencia de alguna señal, una línea de transmisión o canal debería tener presente una señal eléctrica cero, pero en la realidad siempre hay perturbaciones en la línea, incluso cuando no exista señal alguna que esté siendo transmitida. Esto se conoce como nivel de ruido de la línea. Existe otro tipo de ruido, el cual es producido por la temperatura que conserva el cable, pero ésta es muy difícil de controlar, ya que para hacerlo se debe de poner a 0° Kelvin la temperatura del cable. Cuando una señal transmitida se atenúa, el nivel de ruido se reduce. En la Figura 5.12 se muestra el efecto de Ruido. Un parámetro importante que está asociado con un medio de transmisión, es el ratio de potencia en una señal recibida S , para la potencia en el nivel de ruido N . El ratio de ruido S/N se conoce como el ratio señal a ruido y normalmente se expresa en decibeles o dB de la siguiente manera:

$$\frac{S}{N} = 10 \log_{10} \left(\frac{S}{N} \right) \text{ db}$$

Aquí se ve que un ratio de ruido grande S/N significa una señal de potencia relativa al nivel de ruido prevaleciente, resultando una señal de calidad buena. Contrariamente, un ratio de ruido pequeño S/N significa una señal de calidad baja. Para señales digitales, la máxima tasa de información (datos) teórica de un medio de transmisión está relacionado al ratio S/N y puede estar determinado usando una fórmula atribuida Shannon y Hartley, conocida como Ley Shannon-Hartley, la cual establece que :

$$C = B \log_2 \left(1 + \frac{S}{N} \right) \text{ bps}$$

donde C , es la tasa de información en bps, B es el ancho de banda de la línea (o sistema) en Hz, S es la potencia de la señal en watts y N es la potencia del ruido aleatorio en watts.

5.6.5 Interferencia

Es la contaminación de las señales externas producidas por señales con las mismas características y la provoca el hombre.

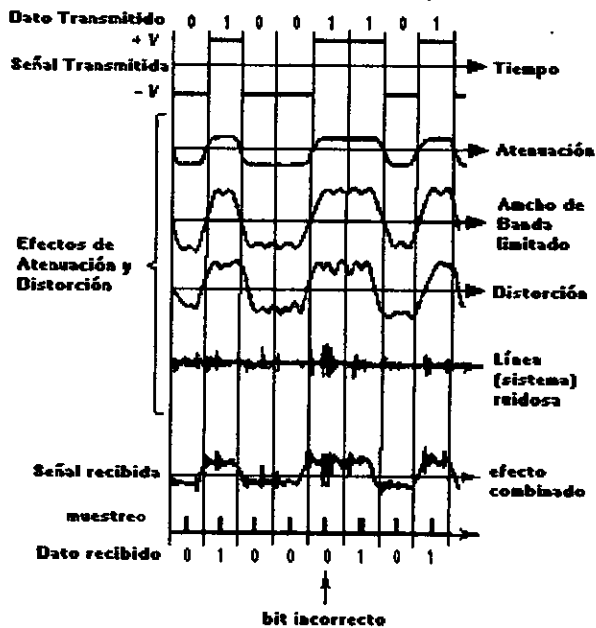


Figura 5.12: Efectos Adversos en la Transmisión

5.7 Modulación

Para que se pueda transmitir varias señales sobre un mismo medio de comunicación al mismo tiempo, las señales de cada señal se deben mover en el espectro de frecuencias y no se confundan entre sí. Esto se lleva a cabo usando otra señal de alta frecuencia que lleva la señal de información (de baja frecuencia) en ella, a otro rango de frecuencias. A este proceso, se le conoce como modulación.

La señal que lleva la información se conoce como señal portadora. Algunas propiedades de la portadora varían en sincronía con la señal de información. La señal portadora es una onda senoidal a la frecuencia de la portadora. Con la ecuación

$$\text{Voltaje instantáneo (e)} = E_{c(\max)} \sin(2\pi f_c t) + \theta$$

se muestra que la onda senoidal tiene tres características que se pueden alterar las cuales son: E_c (voltaje portador), f_c (frecuencia portadora) y θ (ángulo de fase).

Un aspecto muy importante que hay que tomar en cuenta en la modulación, es el alcance que se quiera tener en la transmisión influyendo así, el tamaño de la antena, ya que éste es inversamente proporcional a la frecuencia radiada.

5.7.1 Tipos de Modulación

Existen tres tipos básicos de modulación que se emplean para la transmisión, las cuales son: Modulación de Amplitud (AM), Modulación de Frecuencia (FM) y Modulación de Fase (PM). El principio general de cada modulación se explica a continuación.

5.7.1.1 Modulación en Amplitud

La Modulación de Amplitud (AM) es un incremento o decremento de el voltaje portador (E_c), con el resto de los factores como constantes. Para ejemplificar el proceso de modulación en la Figura 5.13 se muestran las etapas de la Modulación en Amplitud. Nótese que durante el tiempo (a), la señal del mensaje tiene una amplitud cero y la señal portadora tiene una amplitud pico de E_c . En el tiempo (b) la señal de mensaje es positiva, con un valor de $E_c + E_s$. En el tiempo (c) la señal de mensaje cruza por cero y la amplitud pico de la portadora regresa a un nivel igual a E_c . En el tiempo (d) la señal mensaje está en el pico más negativo, $-E_s$, y el tamaño de la señal portadora se ha decrementado a un valor de $E_c - E_s$. Durante el tiempo (e) la señal mensaje regresa a cero, y el voltaje pico de la portadora regresa otra vez a E_c . En la Figura 5.13 también se muestra que la frecuencia de la portadora es constante mientras su amplitud cambia de acuerdo con la forma del tamaño y frecuencia de la señal de mensaje, y que la frecuencia de la portadora es mucho más grande que la frecuencia del mensaje.

En la Modulación en Amplitud debe considerarse que:

- 1) El cambio de la amplitud de la portadora es causa de, y directamente proporcional a, la polaridad y amplitud de la señal de mensaje.
- 2) El nivel cero de la señal de mensaje corresponde al nivel pico de la señal portadora sin modular. Cuando la señal de mensaje pasa a ser positiva, la señal portadora se hace más grande en amplitud que la portadora sin modular por un valor igual a la señal de mensaje pico. Cuando la señal de mensaje moduladora pasa a ser negativa, el nivel pico de la portadora se

hace menor que el nivel pico de la señal portadora sin modular por un valor igual a la señal de mensaje pico.

- 3) Los picos positivos y negativos de la señal portadora modulada son imágenes reflejadas de las otras por las condiciones de $E_{\max}$ y $E_{\min}$.
- 4) Cuando el nivel pico de la señal de mensaje es igual a el nivel pico de la señal portadora, se dice que pasa a ser modulada por completo. Esto se conoce como modulación al 100 %. En (b) la amplitud total $E_c + E_s = 2E_c$, y en el punto (d) la amplitud total es $E_c - E_s = 0$. las condiciones para la modulación al 100 %.

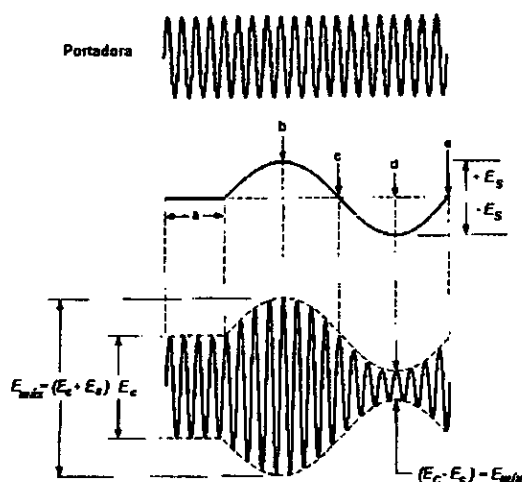


Figura 5.13: Etapas de una Modulación por Amplitud

En la Modulación en Amplitud digital, conocida como ASK (Amplitude Shift Keying; Modulación por Corrimiento de Amplitud), el nivel o amplitud de una frecuencia se intercala entre dos niveles de voltaje, a una tasa de

transmisión determinada por la señal a transmitir. Una frecuencia (portadora) se selecciona para estar en el rango aceptable de frecuencias que el medio de transmisión maneje. Este tipo de modulación, se afecta al variar la señal por atenuación, provocada por las condiciones de propagación de diferentes medios de transmisión. En la Figura 5.14, se muestra una comparación de las diferentes formas de modulación digital.

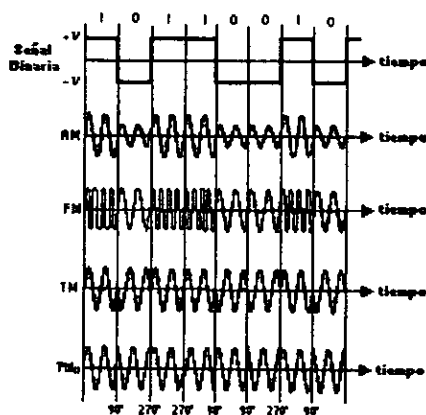


Figura 5.14: Comparación de las diferentes formas de modulación digital

5.7.1.2 Modulación en Frecuencia

La Modulación de Frecuencia (FM) es un cambio en la frecuencia portadora (f_c), con el resto de los factores como constantes. En la Modulación de Frecuencia digital, la frecuencia de una señal portadora de amplitud fija se cambia de acuerdo a la cadena binaria que se transmitirá. Como solamente se requieren dos frecuencias para los datos binarios, este tipo de modulación se conoce también como FM digital o Modulación FSK (Frequency Shift Keying; Modulación por Corrimiento de Frecuencia). Este es el método que se usa más frecuente para bajas tasas de

transmisión de bits de entre trescientos a mil doscientos Baudios. En la Figura 5.14, se muestra una comparación de las diferentes formas de modulación digital.

5.7.1.3 Modulación en Fase

La Modulación por Fase (PM) es un cambio de fase en el ángulo de fase de la portadora (θ). El ángulo de fase no puede cambiar sin también hacer un cambio en la frecuencia. Por lo tanto, la modulación de fase es en realidad una segunda forma de modulación de frecuencia. En la Modulación de Fase digital la frecuencia y amplitud de la señal portadora permanece constante mientras su fase se cambia para cada cambio de bit de datos transmitidos. La manera de usar la Modulación por Fase es con dos señales portadoras fijas (representando a los números binarios 1 y 0) en donde la diferencia de fase es de 180° (equivalente a una señal inversa) entre ellas. A esto se le conoce como Modulación PSK (Phase Shift Keying; Modulación por Corrimiento de Fase). Este tipo de modulación de fase debe mantener una señal portadora de referencia en el receptor con la cual se compara la fase de la señal recibida (esto se conoce como PSK coherente de fase). En la práctica esto requiere de un circuito de demodulación compleja. También este tipo de modulación es muy susceptible a los cambios de fase aleatorios en la señal transmitida. Consecuentemente una manera alternativa de Modulación de PSK se usa a menudo en el empleo de cambios de fase para cada transición de bit determinado por el estado del siguiente bit a ser transmitido relativo con el bit actual. Un cambio de fase de 90° relativo a la señal actual indica que un 0 binario se está transmitiendo, mientras un cambio de fase de 270° indica un 1 binario. De esta forma el circuito demodulador sólo necesita determinar la magnitud de cada cambio de fase aún para el valor absoluto. Este tipo de modulador de fase se conoce

como PSK diferencial. En la Figura 5.14, se muestra una comparación de las diferentes formas de modulación digital.

5.7.2 Demodulación

Existen tres aspectos que se deben considerar para el proceso de la demodulación, es especial de la demodulación digital, que son los siguientes:

- 1) Cuando una señal portadora de una frecuencia fija f_c se modula por una segunda señal de frecuencia fija f_m , se producen un número de componentes de frecuencia adicionales conocido como bandas laterales, por ejemplo con AM sólo se producen dos bandas laterales, en $f_c + f_m$ y en $f_c - f_m$, cada una conteniendo una fracción de potencia conformada en la portadora. Cabe señalar que estas son las bandas laterales que contienen la información f_m requerida. Para la FM y PM, muchas bandas laterales se producen en múltiples f_m de la portadora ($f_c + f_m, f_c + 2f_m$, etc.).
- 2) Usando Análisis de Fourier se puede demostrar que una onda cuadrada se forma de un número infinito de componentes de frecuencia senoidales. Esa señal cuadrada consta de una frecuencia fundamental f_n , igual a la mitad de la tasa de transmisión de bits en hertz y múltiplos de esta frecuencia ($3f_n, 5f_n, 7f_n$, etc.) conocidos como armónicas.
- 3) Cuando un conjunto de datos en binario se transmite, los patrones de bit cambian constantemente. Por lo tanto la frecuencia fundamental (y las armónicas asociadas) también cambian constantemente. Por un lado, el conjunto de datos debe ser una señal cuadrada mientras que por otro lado ésta debe de ser una señal de frecuencia DC.

Se puede concluir que la señal producida después de modular una señal portadora senoidal con una onda cuadrada, se hace por medio de la portadora más un número infinito posibles de componentes de frecuencia los cuales contienen la información requerida. Como la mayor parte de la potencia está contenida en la frecuencia fundamental del conjunto de bits a modular y en las bandas laterales primarias de la señal modulada resultante, es posible determinar la información transmitida al detectar solo una banda limitada de frecuencias para cada lado de la portadora y asegurar que esta banda abarque las bandas laterales primarias producidas por la modulación de frecuencia máxima. Esto está determinado por la tasa de transmisión de bits usada.

5.7.3 Factor de Modulación y Porcentaje de Modulación

Una señal portadora por sí mismo no es modulable o es una modulación en cero, y su amplitud es constante. Cuando la amplitud de la señal portadora se cambia, el grado de cambio es una medida del porcentaje de modulación, por ejemplo el cambio de la mitad de amplitud de la portadora (pico a pico) es una modulación al 50%. Así el 100 % de modulación puede ser descrito como un cambio de amplitud que ocasiona que el voltaje de la portadora se haga cero en un punto de la onda y se haga el doble de voltaje de la portadora sin modular en otro punto de la onda.

El Factor de Modulación (m) se obtiene observando la señal modulada en un osciloscopio y registrando los valores pico a pico de la señal en su cresta y en su valle:

$$m = \frac{E_{\max} - E_{\min}}{E_{\max} + E_{\min}} \quad \dots \text{ cuando se ve la señal modulada}$$

$$m = \frac{E_{\text{pico del mensaje}}}{E_{\text{pico de la portadora}}} \quad \dots \text{ cuando los dos voltajes sin modular se conocen}$$

Lo más común es que se conozca el voltaje de la portadora y se ponga un porcentaje de modulación. El despliegue de la señal en un osciloscopio mostrará los valores $E_{\max}$ y $E_{\min}$ de la portadora sin modular. Aunque muchas veces se utilicen los intentos de prueba y error, la cresta y valle de la señal puede ser predeterminada por:

$$E_{\max} = E_c (1 + m)$$

y

$$E_{\min} = E_c (1 - m)$$

Generalmente se recomienda mantener la modulación de la portadora entre un 85% y un 95%, sin exceder el 100% de modulación. La señal de la portadora sobremodulada distorsiona el mensaje y reduce la salida de transmisión.

5.7.4 Sobremodulación

Consideremos que una señal para modular es una rampa comenzando ésta en menos uno e incrementándose linealmente a más uno. Multiplexando esto por el portador nos da una señal de portador modulado. El problema ocurre cuando la señal se demodula, la envolvente positiva de la portadora modulada no se parece al mensaje. Este problema se puede corregir agregando al mensaje una constante de valor DC, para que la señal resultante no sea negativa. Al hacer esto, la envolvente de la porción positiva modulada es idéntica a la señal del mensaje.

5.7.5 Índice de Modulación

El problema de la sobremodulación se puede prevenir agregando una constante en términos de DC, a la señal del mensaje al momento de modular, para que la forma de onda resultante sea siempre positiva. Esto es, $m(t) + DC$, donde: $m(t)$ es el mensaje y DC es el valor constante DC.

Como el corrimiento de DC se aproxima a la magnitud del máximo pico negativo en la señal del mensaje, la portadora modulada se aproximará a una sobremodulación. Si el corrimiento de DC y el pico máximo negativo son iguales, la portadora modulada está exactamente en el borde de la señal sobremodulada. Así, el ratio del valor absoluto del máximo pico negativo al valor del corrimiento de DC es una medida de como la portadora modulada está siendo sobremodulada. Este ratio es conocido como el índice de modulación.

Se asume que la señal modulada es normalizada para que su magnitud sea menor o igual a uno, o bien, $|m(t)| \leq 1$. En este caso, un corrimiento de DC de uno asegurará que ocurra la sobremodulación, es decir que para evitar la sobremodulación se tiene que $DC \geq |m(t)|_{\max}$. El índice de modulación será el valor máximo de la magnitud de $m(t)$. Esto es, $I.M. = |m(t)|_{\max} / DC$. Si la señal del mensaje es una señal sinusoidal con amplitud máxima de m , donde $|m| \leq 1$, y si entrada a el modulador es

$$m \sin(2\pi f t) + 1,$$

la amplitud de m de la señal del mensaje sinusoidal puede ser el índice de modulación.

5.8 Filtros

Algunos dispositivos electrónicos se diseñan para no tener respuestas de frecuencias no planas. Un grupo de frecuencias, conocido como banda de frecuencias, se debe amplificar más que otras. En el caso extremo una banda de frecuencias debe eliminar a las otras. Una señal que pasa a través de un dispositivo, conocido como filtro debe tener una banda completa de frecuencias eliminadas o reducidas en amplitud, dependiendo de la respuesta de frecuencia del filtro.

El efecto de ganancia de un filtro sobre una señal de entrada es una multiplicación del espectro de frecuencia de la señal de entrada por la repuesta de frecuencia del filtro. Si el filtro sólo pasa las frecuencias altas, el efecto sobre la señal de salida es eliminar las frecuencias bajas de la señal de entrada. Algebraicamente, el espectro de salida $O(f)$ iguala al producto del espectro de entrada $I(f)$ por la respuesta en frecuencia del filtro $H(f)$.

$$O(f) = H(f) \cdot I(f)$$

Un filtro que pasa sólo las frecuencia bajas es llamado filtro paso-bajas, uno que pasa sólo las frecuencias altas se conoce como filtro paso-altas, un filtro que elimina frecuencias bajas y altas, y pasa sólo las frecuencias de la parte media del rango es conocido como filtro paso-bandas, el filtro que pasa frecuencias bajas y altas y elimina las frecuencias de la parte media de la banda se conoce como filtro de rechaza-bandas. La frecuencia en la cual un filtro tiene una transición entre pasar y detener frecuencias se le conoce como frecuencia de corte.

Capítulo 6:

Detección y Corrección de Errores en los Datos

6.1 Introducción

Una de las grandes ventajas de las señales digitales es que son más inmunes al ruido que las analógicas. Esto se debe en primer lugar a que el detector digital de datos sólo puede detectar si cada símbolo es un 1 o es un 0, en el caso de que el ruido sea muy bajo los datos pueden ser detectados perfectamente.

Cuando existe demasiado ruido el detector puede equivocarse al tomar la decisión de la señal, es decir puede cambiar 0s por 1s o viceversa. Para corregir esto es necesario codificar los datos. Esto significa agregar más símbolos, a los símbolos de la información (redundancia), que sirvan de chequeadores en la verificación de la información, es decir, esta información redundante debe tener ciertas características, de tal forma, que ayude a detectar errores y en algunos ocasiones corregirlos. Esto hará que la probabilidad de error disminuya.

Actualmente las técnicas de codificación para el control de errores, son muy usadas en los sistemas digitales, ya que son fáciles de implementarse debido a

los avances en la tecnología electrónica y porque la cantidad de datos digitales transmitidos o almacenados deben tener una gran confiabilidad.

Cuando se diseña un código para el control de errores en un sistema de comunicación digital se deben hacer algunas consideraciones. Primero debemos saber qué tipo de error está ocurriendo en el sistema. Los errores se clasifican en tres tipos que son: errores aleatorios, errores ráfaga y errores byte. Los errores aleatorios pueden ocurrir en cada símbolo. Los errores ráfaga ocurren intensivamente en un periodo de datos. Y los errores byte ocurren en un pequeño bloque de datos. En algunos casos pueden presentarse dos o los tres tipos de errores. Para cada tipo de error existe un código que puede detectar efectivamente errores y/o corregirlos. No necesariamente un código diseñado para detectar y/o corregir un tipo de error es efectivo para otro tipo.

Los códigos pueden ser largos o cortos, esto depende de su capacidad para corregir errores. Generalmente los códigos largos son lentos en su transmisión pero son muy efectivos en la corrección y detección de errores, además de ser lentos en la transmisión también son más complejos en su codificación y decodificación. Los códigos cortos pueden codificarse, transmitirse y decodificarse más rápidamente pero su probabilidad de error es mayor. En la Tabla 6.1 se muestran algunos códigos de control de errores que pueden ser usados dependiendo del tipo de error.

Existen tres esquemas, en el control de errores, que también se deben considerar en el diseño de un código. Estos esquemas son: FEC (Forward Error Correction) el decodificador corrige cualquier error de acuerdo a una regla dada, AEQ (Automatic Repeat Request) cuando el decodificador detecta un error hace una petición

de retransmisión de datos y en el tercero se hace una estimación del verdadero valor de datos erróneos utilizando características estadísticas de los datos.

Tipo de Error	Códigos	Tamaño
Aleatorios	Código convolucional Self-orthogonal	Corto ↑ ↓ Largo
	Códigos BCH	
	Códigos convolucionales con decodificador Viterbi	
	Códigos convolucionales con decodificación secuencial	
	Códigos concatenados (Reed-Solomon + convolucional)	
Ráfaga	Códigos Iwadare	Corto ↑ ↓ Largo
	Códigos Fire	
	Códigos Reed-Solomon	
	Códigos Reed-Solomon dobles	
Byte	Códigos Reed-Solomon	

Tabla 6.1: Códigos para el Control de Errores Típicos

El esquema de control de errores es determinado por los requerimientos de veracidad en los datos, la permisividad de tiempo de retardo del receptor del decodificador de datos, la cantidad de hardware que puede ser puesta para mandar y recibir, etc.

En este capítulo se darán algunas definiciones de códigos y la teoría básica de algunos de ellos así como su definición y codificación. Sólo se introducirá el esquema de decodificación para los códigos Reed-Solomon y BCH.

6.2 Conceptos Básicos de Codificación y Decodificación

6.2.1 Modelo General de un Sistema con Control de Errores

En un sistema de comunicaciones digitales con control de error antes de ser transmitido el mensaje pasa por un codificador y después de ser recibido pasa por un decodificador que obtiene un valor estimado del mensaje enviado. La Figura 6.1 muestra un diagrama de bloque de lo antes mencionado.

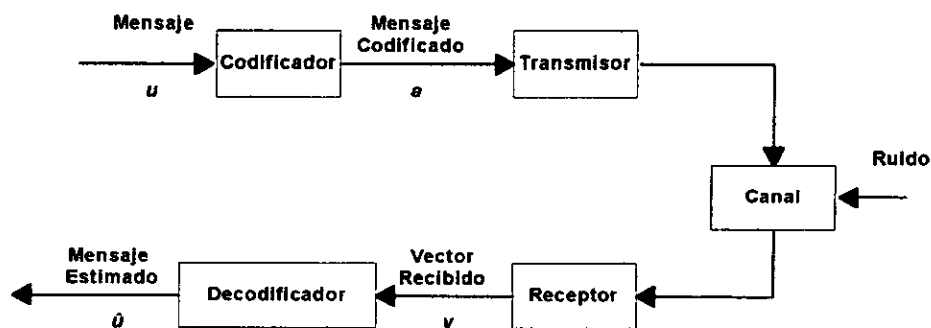


Figura 6.1: Diagrama de un Sistema de Comunicaciones Digitales con Control de Error.

6.2.2 Códigos de Bloque

Los códigos de bloques dividen el mensaje, que puede ser una secuencia continua de símbolos binarios, en bloques de longitud k , este puede ser denominado $\vec{u} = (u_1 u_2 \dots u_k)$ en donde u_i puede tomar los valores de 0 ó 1. El codificador manipula los bloques de información, dependiendo de las reglas de codificación del código, dando como resultado una palabra o vector de código, la cual

tienen una longitud n ($n > k$). La palabra código puede ser denominada $\bar{a} = (a_1 a_2 \dots a_k \dots a_n)$ en donde a_i puede ser 0 ó 1.

La palabra código es enviada por el canal de transmisión, en donde puede ser alterada por el ruido, una vez llegando al receptor es decodificada independientemente de las demás palabras de código.

A n se le conoce como longitud del código y a k como dimensión del código.

La razón k/n es llamada tasa de código y se denomina por R .

El conjunto de todas las posibles palabras que se pueden generar con un código de bloques se puede representar:

$$C = (\bar{a}_1, \bar{a}_2, \dots, \bar{a}_M), \text{ donde } M = 2^k$$

La señal recibida, posiblemente alterada por el ruido se denomina por $\bar{y} = (y_1, y_2, \dots, y_n)$ y se le llama vector recibido. El decodificador estima el vector enviado, el cual se denomina $\bar{a}$ y a partir de éste decodifica el mensaje. La palabra decodificada se denomina $\hat{u}$. En algunos casos el decodificador sólo decide si hay o no errores.

El vector recibido puede expresarse:

$$\bar{y} = \bar{x} + \bar{e}$$

.....6.1

donde $\bar{e} = (e_1, e_2, \dots, e_n)$ y representa el ruido que contiene el mensaje y es llamado vector de error.

Los métodos de decodificación se clasifican en hard decision decoding y en soft decision decoding.

Los códigos de hard decision decoding detectan si cada símbolo transmitido es un cero o un uno y si el vector recibido es una secuencia binaria. Y en los códigos de soft decision decoding, el receptor genera información analógica para cada símbolo recibido y el decodificador recupera el mensaje de esta señal analógica.

En hard decision decoding, la Ecuación 6.1 sigue las siguientes reglas:

$$\bar{a} + \bar{e} = \bar{y}$$

$$0 + 0 = 0$$

$$1 + 0 = 1$$

$$0 + 1 = 1$$

$$1 + 1 = 0$$

... 6.2

Esta regla puede ser implementada con una *OR* exclusiva.

Cada bit de e indica si ha ocurrido un error o no. Si el bit i es un 1 quiere decir que a ocurrido un error en la posición i de la palabra recibida; de lo contrario el bit es un 0.

6.2.3 Códigos de Árbol

Estos códigos manipulan la secuencia de símbolos de la información sin separarla en bloques, como los códigos anteriores. El codificador de un código de árbol procesa la información de forma continua y asocia a cada secuencia de información un código que contiene más símbolos.

Los códigos convolucionales pertenecen a los códigos de árbol. Estos códigos son fáciles de implementar en comparación con otros códigos de árbol.

6.2.4 Códigos Convolucionales

Cada código es generado por varios bloques de mensajes consecutivos. Un codificador convolucional consta, generalmente, de registros de corrimiento para retener el mensaje y circuitos lógicos para producir las salidas.

6.2.5 Distancia de Hamming

La distancia de Hamming entre los vectores $\bar{a} = (a_1 a_2 \dots a_n)$, que es la información codificada transmitida, y $y = (y_1 y_2 \dots y_n)$, que es el vector recibido, se denota por $d(a, y)$ y es el número de posiciones en las que x y y difieren. Esto se puede expresar de la siguiente forma:

$$d_H(a, y) = \sum_{i=1}^n \delta(a_i, y_i) ,$$

donde:

$$\delta(a_i, y_i) = \begin{cases} 0 & a_i = y_i \\ 1 & a_i \neq y_i \end{cases}$$

Por ejemplo, si enviamos un vector codificado $\bar{a} = (01110)$ y recibimos en vector $\bar{y} = (11100)$, difieren en dos bits por lo tanto $d_H(a, y) = 2$.

El *peso de Hamming* del vector a es denotado por $w_H(a)$ y es el número de unos que contiene el vector a . $w_H(a)$ es igual a la distancia de Hamming entre el vector a y el $\bar{0} = (00\dots 0)$, es decir:

$$w_H(\bar{a}) = d_H(\bar{a}, \bar{0})$$

La distancia mínima de un código, se denota por d , y es la mínima distancia de Hamming entre sus palabras de código $\bar{a}_1, \bar{a}_2, \dots, \bar{a}_M$.

$$\begin{aligned} d &= d_H(a_i, a_j) \\ i, j &= 1 \dots M \\ i &\neq j \end{aligned}$$

La distancia mínima de un código es un parámetro importante para saber su capacidad de detección y corrección de errores.

El mínimo peso de Hamming entre las palabras del código excepto con la palabra $\bar{0}$ es llamado peso mínimo del código.

6.2.6 Errores Aleatorios y Ráfaga

Los errores aleatorios ocurren en un bit transmitido y éste es independiente de los errores que puedan ocurrir en los bits vecinos. Si en número de

bits erróneos es t , entonces el peso de Hamming del vector de error es t y la distancia de Hamming entre el vector transmitido y el vector recibido es t .

Los errores ráfaga ocurren intensivamente en un periodo. El número de bits que hay en este periodo de errores, desde el primer error que ocurre hasta el último error, es llamado longitud ráfaga.

Un ejemplo de errores aleatorios podría ser:

$$\bar{e} = (0001000001)$$

el error ocurre en dos bits recibidos.

Un ejemplo de errores ráfaga es:

$$\bar{e} = (0101101000)$$

en donde su longitud ráfaga es seis. Este vector de error también puede ser considerado como un error aleatorio en cuatro bits.

6.3 Códigos Lineales

6.3.1 Definición y Codificación

Los códigos lineales son los más usados y reciben este nombre porque cumplen con las propiedades de linealidad.

Los códigos lineales dependen de una matriz generadora a la cual llamaremos G y una palabra de información $\bar{u}$. El código de $\bar{u}$, $\bar{a}$, se define por:

$$\bar{a} = \bar{u} \cdot G$$

G es una matriz de $k \times n$ con elementos en un Campo de Galios. Los k renglones de G son linealmente independientes por lo tanto el rango de G es k .

Un campo finito es un campo con un número finito de elementos. El número de elementos de un campo finito es llamado *orden del campo*. Un campo de orden q se denota por $GF(q)$, donde GF son las iniciales en inglés de campo de Galios (Galios field). q se expresa como $q = p^m$, donde p y m son enteros positivos.

El número de palabras codificadas que se pueden obtener es p^t . El número de símbolos de la palabra codificada, en este caso n , es llamado longitud del código y el número de símbolos de la palabra sin codificar, k , es llamado dimensión del código. Un código lineal de longitud n y dimensión k es llamado código (n, k)

Si la matriz generadora tiene en su primera parte la matriz identidad de $k \times k$, es decir:

$$G = \begin{pmatrix} 1 & 0 & \dots & 0 & -P_{1,1} & -P_{2,1} & \dots & -P_{n-k,1} \\ 0 & 1 & \dots & 0 & -P_{1,2} & -P_{2,2} & \dots & -P_{n-k,2} \\ \vdots & \vdots & \dots & \vdots & \vdots & \vdots & \dots & \vdots \\ 0 & 0 & \dots & 1 & -P_{1,k} & -P_{2,k} & \dots & -P_{n-k,k} \end{pmatrix}$$

entonces el código lineal es sistemático. Cuando se utilizan códigos sistemáticos los primeros k símbolos de la palabra codificada son iguales a la palabra sin codificar. En

este caso los primeros k símbolos del mensaje son llamados símbolos del mensaje o símbolos de información y los últimos $n - k$ son llamados símbolos chequeadores.

Los códigos lineales también pueden ser definidos por la siguiente ecuación:

$$H \cdot \bar{a}^T = 0$$

en donde H es una matriz de $(n - k) \times n$ con elementos en $GF(q)$ y $\bar{a}$ es la palabra a codificar. los $n - k$ renglones de H son linealmente independientes. H es llamada matriz chequeadora de paridad o matriz de paridad. Sustituyendo $\bar{a} = \bar{u} \cdot G$ en la ecuación anterior se obtiene:

$$G \cdot H^T = 0$$

La matriz de paridad asociada a la matriz generadora anterior es:

$$H = \begin{pmatrix} P_{1,1} & P_{1,2} & \dots & P_{1,k} & 1 & 0 & \dots & 0 \\ P_{2,1} & P_{2,2} & \dots & P_{2,k} & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \dots & \vdots & \vdots & \vdots & \dots & \vdots \\ P_{n-k,1} & P_{n-k,2} & \dots & P_{n-k,k} & 0 & 0 & \dots & 1 \end{pmatrix}$$

esta matriz tiene en su última parte una matriz identidad de $(n - k) \times (n - k)$. A esta forma se le llama forma canónica escalonada. Cuando se usa la matriz de paridad los símbolos chequeadores $a_{k+1}, a_{k+2}, \dots, a_n$ son expresados en términos de los símbolos del mensaje $(u_1 u_2 \dots u_k) = (a_1 a_2 \dots a_k)$ como sigue:

$$\begin{aligned} a_{k+1} &= -(P_{1,1}u_1 + P_{1,2}u_2 + \dots + P_{1,k}u_k) \\ a_{k+2} &= -(P_{2,1}u_1 + P_{2,2}u_2 + \dots + P_{2,k}u_k) \\ &\vdots \\ a_n &= -(P_{n-k,1}u_1 + P_{n-k,2}u_2 + \dots + P_{n-k,k}u_k) \end{aligned}$$

Es posible transformar una matriz de paridad a la forma canónica escalonada.

Ejemplo:

Considerando un código en GF(2) con una matriz generadora:

$$G = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 \end{pmatrix}$$

Este es un código sistemático ya que en su primera parte la matriz generadora contiene una matriz identidad de 2×2 . La matriz de paridad de este código es:

$$H = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix}$$

Las palabras que pueden ser enviadas son: (00), (01), (10), (11). Para obtener la palabra codificada de algunos de ellos basta con sustituir en $\bar{a} = \bar{u}G$ la matriz generadora y la palabra a codificar. Por ejemplo si la palabra a codificar es (11), su palabra codificada será:

$$\begin{aligned} \bar{a} &= \bar{u}G = (11) \cdot \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 \end{pmatrix} \\ &= (1101) \end{aligned}$$

o utilizando la matriz de paridad :

$$H \cdot \bar{a}^T = 0$$

$$\begin{pmatrix} 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{pmatrix} = \bar{0} \quad \dots\dots\dots 6.3.1$$

como los primeros k símbolos de la palabra codificada son iguales a los de la palabra sin codificar entonces:

$$a_1 = u_1$$

$$a_2 = u_2$$

y los demás símbolos son expresados en términos de los símbolos del mensaje despejándolos de la ecuación 6.3.1:

$$a_3 = u_1 + u_2$$

$$a_4 = u_2$$

donde $\bar{u} = (u_1, u_2) = (11)$.

El conjunto de palabras codificadas es:

$$C = \{0000, 0111, 1010, 1101\}$$

que corresponden a las palabras: $(00), (01), (10), (11)$ respectivamente.

6.3.2 Detección y Corrección de Errores

Para decodificar un código lineal es necesario calcular el síndrome que se denomina por s y se obtienen de:

$$s = H \cdot \bar{y}^T$$

donde H es la matriz de paridad y $\bar{y}$ es el vector recibido.

Consideran que el vector recibido es igual al vector enviado por el transmisor más el error encontrado en el canal de transmisión, es decir:

$$\bar{y} = \bar{a} + \bar{e}$$

entonces:

$$s = H\bar{y}^T = H(\bar{a} + \bar{e})^T = H\bar{e}^T$$

ya que $H\bar{a}^T = \bar{0}$. El síndrome sólo depende del vector de error $\bar{e}$.

Si $\bar{y}$ es una palabra codificada entonces:

$$s = H\bar{y}^T = 0$$

de lo contrario:

$$s = H\bar{y}^T \neq 0$$

Por medio de las expresiones anteriores es posible detectar si ocurrió algún error o no. Si el síndrome es igual con cero significa que no ocurrió ningún error, de otra forma significa que la palabra recibida no es una palabra de código por lo tanto es errónea.

Para detectar errores los decodificadores primero calculan el síndrome de $\bar{y}$. En caso de que haya ocurrido un error el síndrome es diferente de cero y entonces el decodificador detecta que ocurrió un error $\bar{e}$ y que el código transmitido fue $\bar{y} - \bar{e}$.

Generalmente se determina un conjunto de errores que pueden ocurrir durante la transmisión de la información y se almacenan en el decodificador. Si

el error no se encuentra dentro de este conjunto de errores el decodificador sólo indica que existe un error pero no lo puede corregir.

Para ejemplificar lo anterior consideremos un código de longitud $n=3$ en un $GF(2)$, suponiendo que este código sólo corrige errores simples, entonces los posibles errores que se pueden detectar son:

$$E = \{100, 010, 001\}$$

La matriz de paridad del código es:

$$H = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix}$$

El síndrome para cada vector de error debe ser distinto. Los síndromes de los vectores de error de este ejemplo son:

$$s_1 = H\bar{e}_1^T = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

$$s_2 = H\bar{e}_2^T = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

$$s_3 = H\bar{e}_3^T = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

Las palabras de código son: $C = \{000, 111\}$

Suponiendo que se manda la palabra $\bar{a} = 111$ y se recibe $\bar{y} = 101$ el síndrome para y es:

$$s = Hy^T = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

Comparando el síndrome de y con los síndromes de los posibles errores que pueden ocurrir observamos que el síndrome de y es igual a s_2 , por lo tanto se detecta que el error es $\bar{e}_2 = 010$. Para corregir el error simplemente tenemos que hacer la operación de Xor del vector de error e con el vector recibido y .

6.4 Códigos de Bloque

6.4.1 Códigos Cíclicos

6.4.1.1 Definición y Codificación

Una palabra codificada $\bar{a} = (a_0 a_1 \dots a_{n-1})$, donde $a_i \in GF(q)$ puede representarse por el polinomio de grado $n-1$ de x :

$$A(x) = a_0 + a_1x + a_2x^2 + \dots + a_{n-1}x^{n-1}$$

Un mensaje $u = (u_0 u_1 \dots u_{k-1})$ en donde $u_i \in GF(q)$, se puede representar por el polinomio de grado $k-1$:

$$U(x) = u_0 + u_1x + u_2x^2 + \dots + u_{k-1}x^{k-1}$$

Un código pseudocíclico es definido por un conjunto de polinomios de $GF(q)$ de grado menor o igual que $n-1$ que pueden ser divididos por:

$$G(x) = g_0 + g_1x + g_2x^2 + \dots + g_rx^r$$

donde $g_i \in GF(q)$ y $r = n - k$. $G(x)$ es llamado polinomio generador del código.

Para generar la palabra de código $A(x)$ del mensaje $U(x)$ se utiliza la relación:

$$A(x) = U(x)G(x)$$

El conjunto de polinomios $A(x)$ generados por la relación anterior es igual al conjunto de polinomios de grado menor que $n-1$ que son divisibles por $G(x)$. Por lo anterior el código dado por este codificador es un código pseudocíclico. Cualquier palabra codificada con este código pseudocíclico puede ser escrita como $U(x)G(x)$.

La matriz generadora G de un código pseudocíclico es una matriz de $k \times n$ de $GF(q)$ y se puede representar de la siguiente manera:

$$G = \begin{pmatrix} g_0 & g_1 & g_2 & \dots & g_r & 0 & 0 & \dots & 0 \\ 0 & g_0 & g_1 & g_2 & \dots & & g_r & 0 & \dots & 0 \\ & & \vdots & & & & & & & \\ 0 & 0 & 0 & \dots & g_0 & g_1 & g_2 & \dots & g_r \end{pmatrix}$$

Las componentes del vector codificado son:

$$\bar{a} = \bar{u}G$$

donde u es un vector de mensaje y sus elementos son los coeficientes de $A(x)$. El rango de G es k por lo tanto es un código lineal (n, k)

Un código lineal es llamado cíclico si algún corrimiento cíclico de alguna de sus palabras de código es también una palabra codificada. Es decir si tenemos un el vector:

$$\vec{a} = (a_0 a_1 a_2 \dots a_{n-1})$$

un corrimiento cíclico sería:

$$\vec{a}' = (a_{n-1} a_0 a_1 \dots a_{n-2})$$

El polinomio que representa $\vec{a}'$ es:

$$\begin{aligned} A'(x) &= xA(x) - a_{n-1}x^n + a_{n-1} \\ &= xA(x) - a_{n-1}(x^n - 1) \end{aligned}$$

donde $A(x)$ es el polinomio que representa a $\vec{a}$.

Si $A(x)$ es una palabra codificada de un código pseudocíclico y el polinomio generador $G(x)$ divide a $x^n - 1$, entonces $G(x)$ divide a $A'(x)$. Por lo anterior $A'(x)$ es también una palabra codificada de este código pseudo cíclico. Un código cíclico es un código pseudocíclico con un polinomio generado que divide a $x^n - 1$. Cualquier palabra codificada de un código cíclico puede ser escrita como $U(x) \cdot G(x)$.

Si $G(x)$ es el polinomio generador de un código cíclico de longitud n entonces:

$$H(x) = (x^n - 1) / G(x)$$

que es el polinomio checadore de paridad o polinomio de paridad. Si $A(x) = U(x)G(x)$ es una palabra codificada de este código entonces:

$$\begin{aligned} A(x)H(x) &= U(x)G(x)(x^n - 1) / G(x) \\ &= U(x)(x^n - 1) \end{aligned}$$

0

$$A(x)H(x) = 0 \bmod(x^n - 1)$$

Ejemplo: considerado un código pseudocíclico de longitud $n = 3$ en $GF(2)$ con un polinomio generador $G(x) = 1 + x$. El número de símbolos de mensaje es $k = n - r = 2$.

$U(x)$ es un polinomio de grado menor o igual que uno. Las palabras de código son:

$$0 \times G(x) = 0$$

$$1 \times G(x) = 1 + x$$

$$x \times G(x) = x + x^2$$

$$(1 + x) \times G(x) = 1 + x^2$$

La matriz generadora es:

$$G = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix}$$

Como el polinomio $G(x) = 1 + x$ divide a $x^3 - 1$ entonces es un código cíclico $x^3 - 1 = (1 + x)(1 + x + x^2)$. El polinomio checadore de paridad es $H(x) = 1 + x + x^2$.

Las palabras de código son:

$$C = \{000, 110, 011, 101\}$$

Como se puede observar las palabras de código están recorridas cíclicamente, excepto el cero). Este código es el código checadore de paridad de longitud tres. Generalmente un código checadore de paridad de cualquier longitud es un código cíclico con un polinomio generador $G(x) = 1 + x$.

6.4.1.2 Decodificación de Códigos

La palabra recibida $\bar{y} = (y_1 y_2 \dots y_n)$ y un vector de error $\bar{e} = (e_1 e_2 \dots e_n)$ pueden representarse por medio de los siguientes polinomios respectivamente:

$$Y(x) = y_0 + y_1 x + y_2 x^2 + \dots + y_{n-1} x^{n-1}$$

$$E(x) = e_0 + e_1 x + e_2 x^2 + \dots + e_{n-1} x^{n-1}$$

Se sabe que :

$$Y(x) = A(x) + E(x)$$

El polinomio checador de paridad $S(x)$ es definido por el residuo de $Y(x)/G(x)$, es decir:

$$S(x) = Y(x) \bmod G(x)$$

$S(x)$ es un polinomio de grado menor o igual que $r - 1$ en $GF(q)$. Cuando $A(x)$ es dividido por $G(x)$ se obtiene:

$$\begin{aligned} S(x) &= [A(x) + E(x)] \bmod G(x) \\ &= E(x) \bmod G(x) \end{aligned}$$

Por lo tanto $S(x)$ es determinado sólo por $E(x)$.

En el caso de solo detección de error, se debe determinar únicamente si o no $S(x)$ es cero, si es cero no hay error.

El método para la detección de error en un código cíclico es llamado CRC (Cyclic Redundancy Check, Checador de Redundancia Cíclica) o CRCC (Cyclic Redundancy Check Code, Código Checador de Redundancia Cíclica).

La detección de errores es llevada a cabo por la averiguación del polinomio de error $E(x)$ que corresponde al síndrome $S(x)$.

Suponiendo que un error simple $E(x) = 1$ ocurre en el canal. Entonces el síndrome para este error es:

$$\begin{aligned} S(x) &= E(x) \bmod G(x) \\ &= 1 \bmod G(x) \\ &= 1 \end{aligned}$$

Si el código empleado es capaz de corregir un error simple, entonces el vector de error correspondiente a $S(x) = 1$ debe ser $E(x) = 1$. Por lo tanto se debe decidir que el error $E(x) = 1$ ocurre si $S(x) = 1$.

Si se hace un corrimiento cíclico del vector $Y(x)$ $n - i$ bits, suponiendo que el error simple $E(x) = x^i$ ocurrió, el vector obtenido contendrá el error $E(x) = 1$ por lo tanto el síndrome para el corrimiento de $Y(x)$ es $S(x)=1$.

6.4.2 Códigos BCH

Los códigos BCH son de los más importantes códigos de detección y corrección de errores. Ellos fueron propuestos por Bose y Chaudhuri en 1960 y por Hocquenghem en 1959.

Un código cíclico de longitud n sobre $GF(q)$ es un código BCH de distancia diseñada δ si, para algún entero $b \geq 0$, el polinomio del código es:

$$G(x) = LCM \{M_b(x), M_{b+1}(x), \dots, M_{b+\delta-2}(x)\}$$

donde $M_i(x)$ es el mínimo polinomio de α^i sobre $GF(q)$. $\alpha^b, \alpha^{b+1}, \dots, \alpha^{b+\delta-2}$ son ceros de $G(x)$ (la definición del mínimo polinomio se dará más adelante). α es un elemento de $GF(q^m)$ y b es un entero positivo.

Un código BCH se llama primitivo si $n = q^m - 1$, y no primitivo si $n \neq q^m - 1$. Es asumido que un código BCH es primitivo a menos que se indique lo contrario. La dimensión de un código BCH primitivo está definida por:

$$k = n - \text{gra}[G(x)]$$

La distancia diseñada δ es la distancia mínima considerada para el diseño de un código BCH. Aún no se ha encontrado una expresión exacta para la mínima distancia de un código BCH, pero la aproximación que se obtiene es bastante ajustada.

Debido al importante papel que tienen los códigos BCH en los sistemas de comunicaciones digitales serán explicados más detalladamente en el siguiente capítulo.

6.4.3 Códigos Reed-Solomon

Fueron creados por Reed y Solomon en 1960. Son una clase especial de códigos BCH. Un código Reed-Solomon es un código BCH sobre $GF(q)$ de longitud

$n = q-1$, esto es que, $m = 1$. Por lo tanto el polinomio generador de un código Reed-Solomon de distancia diseñada δ es el polinomio de más bajo grado sobre $GF(q)$ teniendo a $\alpha^b, \alpha^{b+1}, \dots, \alpha^{b+\delta-2}$ como ceros, donde α es un elemento primitivo de $GF(q)$. Si el polinomio mínimo de α^j sobre $GF(q)$ es simplificado por $M_i(x) = x - \alpha^j$, el polinomio generado es:

$$G(x) = (x - \alpha^b)(x - \alpha^{b+1}) \dots (x - \alpha^{b+\delta-2})$$

Para los códigos Reed-Solomon la distancia diseñada coincide con la mínima distancia del código es decir $d = \delta$.

La dimensión del código es:

$$n - \text{gra}[G(x)] = (q - 1) - (d - 1) = q - d$$

Un código Reed-Solomon es un código $(q - 1, q - d)$ sobre $GF(q)$ con una mínima distancia d .

6.5 Códigos BCH

Antes de empezar con el diseño de códigos BCH introduciremos algunos conceptos importantes.

6.5.1 Conceptos Básicos

6.5.1.1 Campos Finitos

Un campo finito es un conjunto de elementos que pueden ser sumados, restados, multiplicados y divididos (excepto divididos por cero). Un campo es un conjunto de elementos que cumplen las siguientes reglas para cualquier elemento x , y o z :

- 1.- $x + y = y + x$
- 2.- $x + (y + z) = (x + y) + z$, $x(yz) = (xy)z$
- 3.- $x(y + z) = xy + xz$
- 4.- 0 debe existir tal que: $x + 0 = 0 + x = x$
- 5.- 1 debe existir tal que $x1 = 1x = x$
- 6.- $-x$ debe existir tal que: $(-x) + x = x + (-x) = 0$
- 7.- x^{-1} debe existir tal que: $x^{-1}x = xx^{-1} = 1$, para $x \neq 0$

El número de elementos de un campo finito es el orden del campo. Un campo del orden q es denotado por $GF(q)$. $GF(q)$ sólo existe cuando q puede ser expresado como $q = p^m$, donde p es un entero y m es un entero positivo. $GF(p^m)$ existe para cada p y m , en un campo con p^m elementos. p es llamada la característica del campo. Si un campo finito tiene la característica p entonces:

$$p\beta = \beta + \beta + \dots + \beta = 0, \quad (p \text{ veces})$$

para un elemento β en el campo. Si los elementos de $GF(q)$ pueden ser expresados como.

$$0, 1, \alpha, \alpha^2, \dots, \alpha^{q-2}$$

El elemento 1 puede ser escrito como α^0 . Por lo tanto todos los elementos de un campo finito excepto 0 son potencias de α . Las propiedades de α son:

$$\alpha^{q-1} = 1$$

$$\alpha^i \neq 1, \quad 1 \leq i \leq q-2$$

De tal manera que un elemento de un campo finito es llamado *elemento primitivo*. Una raíz de una ecuación $f(x) = 0$ es llamada cero de $f(x)$. Un elemento primitivo de $GF(p^m)$ es un cero de un polinomio primitivo de orden m . La multiplicación en un campo finito puede ser desarrollada mediante la siguiente formula:

$$\alpha^i \alpha^j = \alpha^{(i+j) \bmod (q-1)}$$

$f(\alpha) = 0$ donde $f(x)$ es un polinomio primitivo, un elemento de $GF(p^m)$ es expresado como un polinomio en α :

$$\alpha = a_0 + a_1\alpha + a_2\alpha^2 + \dots + a_{m-1}\alpha^{m-1}$$

$$a_i \in GF(p)$$

Un campo finito puede ser desarrollado usando la representación:

$$(a_0 + a_1\alpha + \dots + a_{m-1}\alpha^{m-1}) + (b_0 + b_1\alpha + \dots + b_{m-1}\alpha^{m-1}) = \\ = (a_0 + b_0) + (a_1 + b_1)\alpha + \dots + (a_{m-1} + b_{m-1})\alpha^{m-1}$$

donde $a_i + b_i$ pertenece a la adición p .

El campo finito más simple es $GF(2)$. En donde la adición y la multiplicación son llevadas a cabo en $GF(2)$ como sigue:

+	0	1
0	0	1
1	1	0

X	0	1
0	0	0
1	0	1

6.5.1.2 Polinomio Primitivo

Para cualquier elemento β excepto 0, existe un entero positivo e tal que:

$$\beta^e = 1$$

el mínimo e que satisface esta ecuación es llamado el orden de β . Si un elemento de $GF(p^m)$ tiene orden $p^m - 1$ es llamado *elemento primitivo*.

Si un polinomio en un campo finito no puede ser expresado como producto de dos polinomios o más es llamado polinomio irreducible. Todos los ordenes de los ceros de un polinomio irreducible son iguales. Si β que pertenece a $GF(p^m)$ es un cero de un polinomio irreducible $f(x)$ de grado r sobre $GF(p)$, entonces $\beta, \beta^p, \dots, \beta^{p^{r-1}}$ también son ceros de $f(x)$.

Un polinomio primitivo es aquel polinomio irreducible que tiene un elemento primitivo.

Algunos polinomios primitivos con coeficientes en $GF(2)$ son:

Grado	Polinomio primitivo
1	$1 + x$
2	$1 + x + x^2$
3	$1 + x + x^3$
4	$1 + x + x^4$
5	$1 + x^2 + x^5$
6	$1 + x + x^6$
7	$1 + x + x^7$
8	$1 + x^2 + x^3 + x^4 + x^8$

Un polinomio primitivo con coeficientes en $GF(p)$ y que tiene un elemento primitivo de $GF(p^m)$ como un cero tienen grado m .

Si el término de más alto grado de un polinomio es 1 es llamado *polinomio monico*.

Un polinomio mínimo sobre $GF(p)$ de β es el polinomio monico de más bajo grado $M(x)$ con coeficientes en $GF(p)$ tal que $M(\beta) = 0$, para un elemento β de $GF(p)$.

Un polinomio mínimo es irreducible y tiene menor o igual grado que m .

Si $M(x)$ es el polinomio mínimo de β , $M(x)$ es el polinomio mínimo de $\beta^p, \beta^{p^2}, \dots, \beta^{p^{m-1}}$ también. El mínimo polinomio de un elemento primitivo es un polinomio primitivo.

Las raíces de la ecuación:

$$x^q - x = 0$$

constituyen $GF(q)$. De este modo:

$$x^q - x = \prod_{\beta \in GF(q)} (x - \beta)$$

Si $M(x)$ es el polinomio mínimo de un elemento de $GF(q)$, $x^q - x$ se puede dividir por $M(x)$.

Para ejemplificar los conceptos anteriores consideremos el campo finito $GF(2^2)$ que tiene como elementos a:

$$0, 1, \alpha, \alpha^2$$

(ya que los elementos de un campo pueden ser expresados como $0, 1, \alpha, \dots, \alpha^{q-2}$) considerando que α es un cero del polinomio primitivo de orden 2 $f(x) = 1 + x + x^2$ (por que el campo es de 2^2), se tiene:

$$f(\alpha) = 1 + \alpha + \alpha^2 = 0$$

despejado α^2 :

$$\alpha^2 = \alpha + 1$$

Por lo tanto los elementos de $GF(2^2)$ son expresados como:

$$0 = 0$$

$$1 = 1$$

$$\alpha = \alpha$$

$$\alpha^2 = \alpha + 1$$

o también puede representarse en forma vectorial en donde sólo se ponen los coeficientes del polinomio representativo de cada elemento del campo, es decir:

$$0 : (00)$$

$$1 : (01)$$

$$\alpha : (10)$$

$$\alpha^2 : (11)$$

Un ejemplo para la adición es el siguiente:

$$1 + \alpha^2 = 1 + (\alpha + 1) = \alpha$$

y para la multiplicación:

$$\alpha^2 \alpha^2 = \alpha^{4 \bmod 3} = \alpha$$

$$\alpha\alpha^2 = \alpha^{3 \bmod 3} = 1$$

el último ejemplo prueba propiedad $\alpha^{q-1} = 1$.

Ahora consideremos el campo finito $GF(2^3)$ donde α es un cero del polinomio primitivo de orden 3 $f(x) = 1 + x + x^3$ (por que el campo es de 2^3), entonces:

$$f(\alpha) = 1 + \alpha + \alpha^3 = 0$$

$$\alpha^3 = \alpha + 1$$

por lo tanto sus elementos pueden ser representados :

<i>elemento</i>	<i>forma polinomial</i>	<i>forma vectorial</i>
0	0	(000)
1	1	(001)
α	α	(010)
α^2	α^2	(100)
α^3	$\alpha + 1$	(011)
α^4	$\alpha^2 + \alpha$	(110)
α^5	$\alpha^2 + \alpha + 1$	(111)
α^6	$\alpha^2 + 1$	(101)

Para el campo finito $GF(2^4)$ donde α es un cero del polinomio primitivo de orden 4 $f(x) = 1 + x + x^4$ (por que el campo es de 2^4), entonces:

$$f(\alpha) = 1 + \alpha + \alpha^4 = 0$$

$$\alpha^4 = \alpha + 1$$

sus elementos pueden ser representados :

<i>Elemento</i>	<i>Forma Polinomial</i>	<i>Forma Vectorial</i>
0	0	(0000)
1	1	(0001)
α	α	(0010)
α^2	α^2	(0100)
α^3	α^3	(1000)
α^4	$\alpha + 1$	(0011)
α^5	$\alpha^2 + \alpha$	(0110)
α^6	$\alpha^3 + \alpha$	(1100)
α^7	$\alpha^3 + \alpha + 1$	(1011)
α^8	$\alpha^2 + 1$	(0101)
α^9	$\alpha^3 + \alpha$	(1010)
α^{10}	$\alpha^2 + \alpha + 1$	(0111)
α^{11}	$\alpha^3 + \alpha^2 + \alpha$	(1110)
α^{12}	$\alpha^3 + \alpha^2 + \alpha + 1$	(1111)
α^{13}	$\alpha^3 + \alpha^2 + 1$	(1101)
α^{14}	$\alpha^3 + 1$	(1001)

El polinomio primitivo $f(x)$ es el mínimo polinomio de α y se puede denotar como $M_f(x)$, también es el polinomio mínimo de α^2 , α^4 y α^8 . Proveemos para α^8 que es un cero de $M_f(x)$:

$$\begin{aligned}
 M_1 &= 1 + \alpha^8 + (\alpha^8)^4 = 1 + \alpha^8 + \alpha^{32} \\
 &= 1 + (1 + \alpha^2) + \alpha^{32 \bmod 15} \\
 &= 1 + 1 + \alpha^2 + \alpha^2 = 0
 \end{aligned}$$

En la Tabla 6.2 se muestran los polinomios mínimos de los elementos de $GF(2^4)$

<i>Elementos</i>	<i>Polinomio Mínimo</i>
0	$M_{\infty}(x) = x$
1	$M_0(x) = x + 1$
$\alpha, \alpha^2, \alpha^4, \alpha^8$	$M_1(x) = x^4 + x + 1$
$\alpha^3, \alpha^6, \alpha^9, \alpha^{12}$	$M_3(x) = x^4 + x^3 + x^2 + x + 1$
α^5, α^{10}	$M_5(x) = x^2 + x + 1$
$\alpha^7, \alpha^{11}, \alpha^{13}, \alpha^{14}$	$M_7(x) = x^4 + x^3 + 1$

Tabla 6.2: Polinomios Mínimos de los Elementos de $GF(2^4)$

De la ecuación $x^9 - x = \prod_{\beta \in GF(q)} (x - \beta)$ se tiene:

$$x^{16} - x = M_{\infty}(x)M_0(x)M_1(x)M_3(x)M_5(x)M_7(x)$$

6.5.2 Diseño de Códigos BCH

En todos los ejemplos anteriores se ha trabajado con campos de base 2 por que nos serán útiles para el diseño de los códigos BCH. Sólo se trabajará con códigos BCH binarios ya que son los más usados y los más fáciles de implementar electrónicamente.

Para el diseño de los códigos BCH es necesario definir sobre qué campo se diseñará el código, su longitud n , el número de errores que será capaz de detectar y/o corregir y a partir de los datos anteriores se debe calcular el polinomio generador y la dimensión k del código.

Ejemplo 1.

Suponga que tiene que diseñar un código sobre $GF(2^4)$, que tenga la capacidad de corregir un error.

Como debe tener la capacidad de corregir un error su distancia diseñada δ debe ser igual a 3. Partiendo de la ecuación:

$$G(x) = LCM\{M_b(x), M_{b+1}(x), \dots, M_{b+\delta-2}(x)\}$$

sabemos que necesitamos los polinomios mínimos de α y α^2 . El polinomio mínimo de α es $M_1(x) = x^4 + x + 1$ y el de α^2 es el mismo (ver Tabla 6.2), por lo tanto el polinomio generador es:

$$G(x) = x^4 + x + 1$$

La longitud del código es $n = 2^4 - 1 = 15$ y la dimensión es:

$$k = n - \text{Grad}[G(x)] = 15 - 4 = 11$$

por lo tanto el código es un código (15, 11, 3).

Para la codificación de la información se hace el producto de $G(x) \cdot U(x)$.

Ejemplo 2.

Suponga que en vez de corregir un error debe corregir dos errores.

Como se deben de corregir dos errores la distancia diseñada δ debe ser igual a 5, por lo tanto se deben multiplicar los polinomios mínimos de α , α^2 , α^3 , y

α^4 para obtener el polinomio generador. El polinomio mínimo de α , α^2 y α^4 es $M_1(x)$ y el de α^3 es $M_3(x)$, por lo tanto el polinomio generador es:

$$G(x) = M_1(x)M_3(x) = (x^4 + x + 1)(x^4 + x^3 + x^2 + x + 1)$$

La longitud n del código es 15 y la dimensión k es 7, así que el código es un código (15, 7, 5).

6.5.3 Detección y Corrección de Errores en Códigos BCH

Existen varios métodos para la detección y corrección de errores, cualquier método básicamente sigue los siguientes tres pasos:

- 1.- Cálculo del síndrome
- 2.- Cálculo del polinomio localizador de error
- 3.- Cálculo de la posición donde se encuentran los errores.

Para el segundo paso utilizaremos el algoritmo Euclideo, ya que es un algoritmo general para el cálculo de polinomio localizador de errores cuando existen múltiples errores, sin embargo aunque es un método rápido en la detección de errores, no es recomendable para la detección de 1 a 3 errores por que existen algoritmos más sencillos y rápidos para estos casos.

6.5.3.1 Cálculo del Síndrome

Si la palabra recibida es:

$$\bar{y} = (y_1 y_2 \dots y_n)$$

su representación polinomial sería:

$$y(x) = y_1 + y_2x + y_3x^2 + \dots + y_nx^{n-1}$$

entonces el síndrome se define por:

$$S(z) = s_0 + s_1z + \dots + s_{2t-1}z^{2t-1}$$

donde t es el número de errores máximo que puede corregir el código y S_j está definida por:

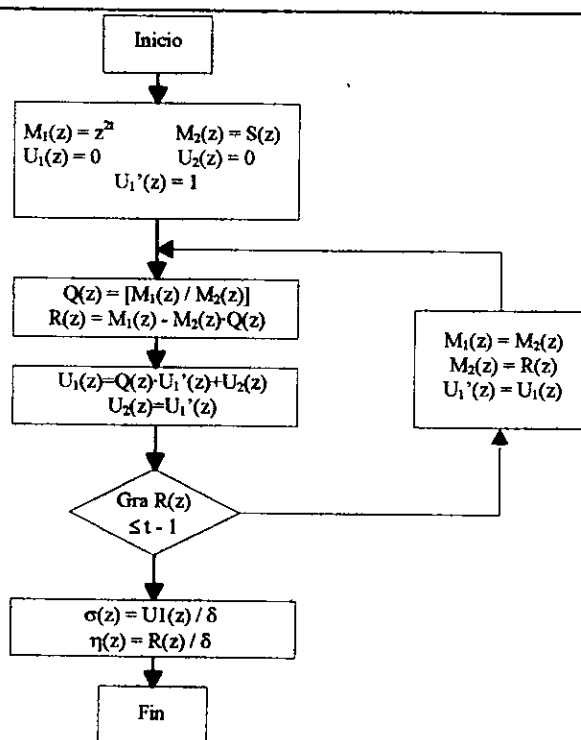
$$S_j = Y(\alpha^{j+\xi})$$

Todas las ecuaciones anteriores tienen un fundamento matemático que no se explicará a detalle ya que no es la finalidad del capítulo, pero si tiene alguna duda puede consultar la bibliografía correspondiente a códigos.

El polinomio Y se evalúa para todas las raíces del polinomio generador, para este caso las raíces del polinomio generador son $\alpha^\xi, \alpha^{\xi+1}, \dots, \alpha^{\xi+2t-1}$.

6.5.3.2 Algoritmo Euclideano

El algoritmo Euclidiano se muestra en la Figura 6.2. Por medio de este algoritmo obtenemos las ecuaciones $\eta(z)$ y $\sigma(z)$ las cuales solamente pueden ser obtenidas a partir de $S(z)$.



donde: $[]$ denotan el cociente de la división
 δ es el coeficiente de máximo grado de $U_1(z)$.

Figura 6.2: Algoritmo Euclideo

6.5.3.3 Localización de la Posición del Error

El polinomio localizador de error es $\sigma(z)$ el cual es obtenido mediante el algoritmo Euclideo y puede ser representado por:

$$\sigma(z) = \prod_{i \in \mathcal{E}} (z - \alpha_i)$$

Para la determinación de la posición del error se utiliza la *búsqueda de Chien* que consiste en evaluar $\sigma(z)$ con elementos de $GF(q)$ desde α_1 hasta α_n , de modo que los elementos que satisfacen la siguiente ecuación:

$$\sigma(\alpha_i) = 0$$

indican que existe un error en la posición i , donde:

$$\alpha_i = \alpha^{-(i-1)}, \quad 1 \leq i \leq n$$

Nota: $\alpha^j = \alpha^{(j \bmod q-1)+(q-1)}$ ya que no existen exponentes negativos dentro de un campo finito y de esta manera se siguen cumpliendo las propiedades de campos finitos.

Ejemplo.

Suponga que se usó un código binario BCH de longitud 15, con capacidad de corregir dos errores y recibió el siguiente vector:

$$001000100000000 = x^2 + x^6 = Y(x)$$

Paso 1. Se calculan los síndromes sobre $GF(16)$ y se obtienen los siguientes resultados:

$$s_0 = Y(\alpha) = \alpha^3$$

$$s_1 = Y(\alpha^2) = \alpha^6$$

$$s_2 = Y(\alpha^3) = \alpha^2$$

$$s_3 = Y(\alpha^4) = \alpha^{12}$$

así que:

$$s(z) = \alpha^{12}z^3 + \alpha^2z^2 + \alpha^6z + \alpha^3$$

Paso 2. Por medio del algoritmo Euclideo se procede a calcular el polinomio localizador del error.

Inicialización:

$$M_1(z) = z^4$$

$$M_2(z) = \alpha^{12}z^3 + \alpha^2z^2 + \alpha^6z + \alpha^3$$

$$U_1(z) = 0$$

$$U_1'(z) = 1$$

$$U_2(z) = 0$$

Ciclo 1:

$$Q(z) = \alpha^3z + \alpha^8$$

$$R(z) = \alpha^{13}z^2 + \alpha^8z + \alpha^{11}$$

$$U_1(z) = \alpha^3z + \alpha^8$$

$$U_2(z) = 1$$

¿es el grado de $R(z) \leq 1$? No

Preparando variables para el siguiente ciclo:

$$M_1(z) = \alpha^{12}z^3 + \alpha^2z^2 + \alpha^6z + \alpha^3$$

$$M_2(z) = \alpha^{13}z^2 + \alpha^8z + \alpha^{11}$$

$$U_1'(z) = \alpha^3z + \alpha^8$$

Ciclo 2:

$$Q(z) = \alpha^{14}z + \alpha^{14}$$

$$R(z) = \alpha^{12}$$

$$U_1(z) = \alpha^2z^2 + \alpha^{12}z + \alpha^9$$

$$U_2(z) = \alpha^3z + \alpha^8$$

¿Es el grado de $R(z) \leq 1$? Si, entonces el algoritmo termina y el polinomio localizador del error es:

$$\sigma(z) = (\alpha^2 z^2 + \alpha^{12} z + \alpha^9) / \alpha^2$$

$$\sigma(z) = z^2 + \alpha^{10} z + \alpha^7$$

Paso 3. Con la búsqueda de Chien se localizan las posiciones del error. El polinomio localizador del error solo es cero para los elementos α^{-6} y α^{-2} , es decir:

$$\sigma(\alpha^{-2}) = \sigma(\alpha^{-6}) = 0$$

esto significa que los errores se encuentran en las posiciones 3 y 7, por lo tanto la palabra enviada fue:

$$(0000000000000000) = 0$$

Capítulo 7:

Microcontrolador 68HC11

El MCU68HC11 es un microcontrolador de 8 bits con capacidad de soportar otros circuitos como periféricos. Tiene una velocidad nominal de 2 MHz. Sus principales características son:

- ◆ Sistema temporizador expandido a 16 bits, con 4 estados preescalares programables.
- ◆ Interface de Comunicaciones Serial (SCI), Sin Retorno a Cero (NRZ).
- ◆ Convertidor Analógico/Digital con ocho canales de 8 bits.
- ◆ Mecanismo protector de bloques para la EEPROM y el CONFIG.
- ◆ Bus Expanded Nonmultiplexed.
- ◆ Empaque de 68 pines.
- ◆ Modo de ahorro de energía por medio de las instrucciones STOP y WAIT.
- ◆ 64 Kb de memoria direccionable.
- ◆ Interface Periférica Serial (SPI).
- ◆ 512 bytes de EEPROM.
- ◆ Circuito acumulador de pulsos de 8 bits.
- ◆ 1024 bytes de RAM Estática (toda almacenada durante el Standby).
- ◆ Instrucciones Bit Test (Prueba de bits) y Branch (Saltos).
- ◆ Circuito de interrupción de tiempo real.
- ◆ Cuatro modos programables del Microcontrolador.
- ◆ Sistema Vigía (Watchdog) para la correcta operación de la computadora

La Figura 7.1 muestra el diagrama de bloques del MCU68HC11F1.

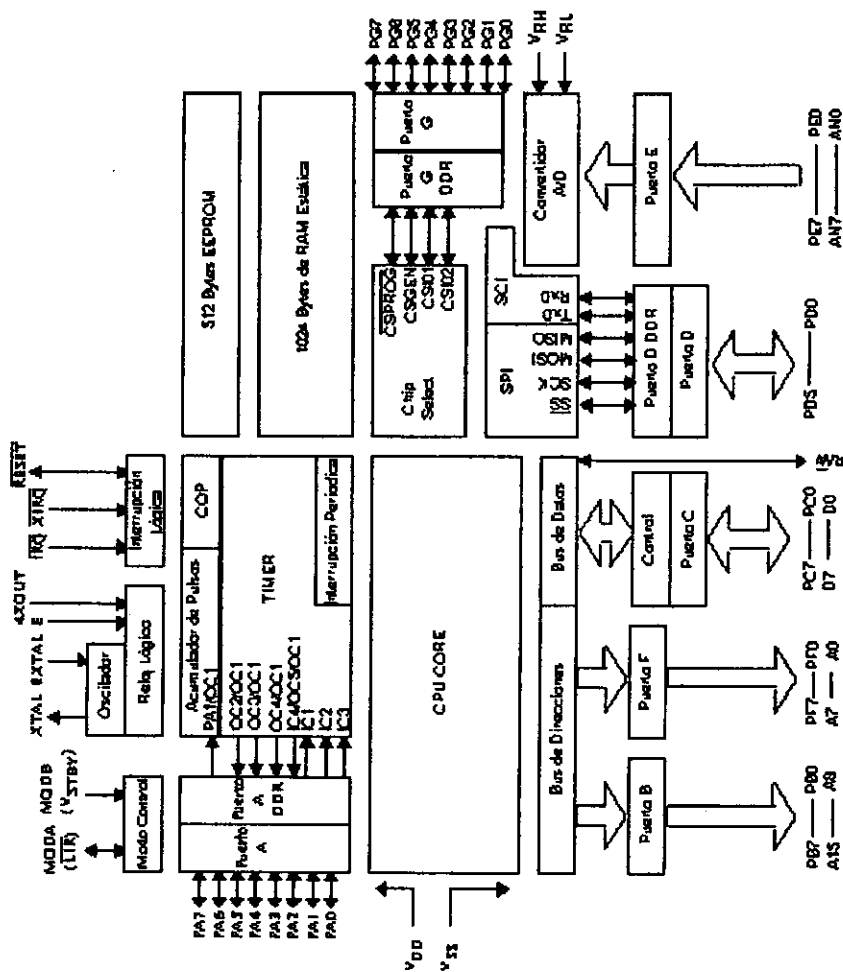


Figura 7.1: Diagrama de Bloques del MCU68HC11F1

7.1 Modos de Operación

El MCU68HC11F1 tiene dos modos normales de operación que son: Single-Chip y Expanded Nonmultiplexed; y dos modos especiales de operación que son: Special Bootstrap y Special Test. El MCU68HC11F1 tiene dos pines especiales, llamados MODA y MODB, para seleccionar una de las formas de operación. La forma de operación de acuerdo con los valores de los pines (MODA y MODB) se muestran en la Tabla 7.1

MODA	MODB	Modo Seleccionado
0	1	Single Chip
1	1	Expanded Nonmultiplexed
0	0	Special Bootstrap
1	0	Special Test

Tabla 7.1: Modos de Operación del MUC68HC11F1

7.1.1 Modo Single Chip

En el modo Single Chip, el MCU68HC11 funciona de forma autónoma y no tiene direccionamiento externo y los pines de los puertos B, C y F funcionan como entradas/salidas de propósito general. El software requerido para controlar el MCU68HC11F1 está contenido en la EEPROM y empieza a funcionar cuando se da un reset al MCU68HC11F1.

7.1.2 Modo Expanded Nonmultiplexed

En el modo Expanded Nonmultiplexed, el MCU68HC11F1 puede direccionar arriba de 64K bytes. Los bits de direccionamiento más significativos salen del los pines del puerto B, y menos significativos salen del puerto F. Los datos

bidireccionales aparecen en el puerto C. El pin lectura/escritura ($R/\overline{W}$) se usa para controlar la dirección de los datos transferidos por el bus del puerto C. Los pines de chip selects programables están dispuestos en los pines del puerto G.

7.1.3 Modo Special Bootstrap

Este modo es parecido al modo Single Chip. El programa residente bootloader, permite cargar un programa de longitud variable en la memoria RAM mediante la Interface de Comunicación Serial (SCI). El programa de control pasa a la RAM cuando una línea ociosa de menos de cuatro caracteres ocurre. En este modo todos los vectores de interrupciones son mapeados por RAM, por lo tanto el usuario puede efectuar brincos a la tabla de vectores. Este modo se puede usar para pruebas y diagnósticos en módulos completos y para programar la EEPROM.

7.1.4 Modo Special Test

Este modo es para hacer pruebas, sin embargo, también se puede usar para calibrar programas o para personalizar datos dentro de la EEPROM interna. Los 512 bytes de EEPROM se apagan inicialmente en este modo. El usuario puede tener acceso a un número de bits de control de prueba especiales. El reset y los vectores de interrupciones son rastreados externamente desde las localidades \$BFC0 - \$BFFF.

7.2 Pines del MCU68HC11F1

A continuación se describe las diferentes señales necesarias para los diferentes pines del MCU68HC11F1.

V_{DD} y V_{SS} : Por estos pines se proporciona la energía eléctrica al MCU68HC11F1. Por el pin V_{DD} entran $+5\text{ V} \pm 10\%$ y por V_{SS} es tierra (0 V).

Reset : Este pin de control bidireccional verificado bajo se usa como una entrada para inicializar el MCU68HC11F1 a un estado conocido. También se usa como una salida open-drain para indicar que se ha detectado un fallo interno en el monitor del reloj o en el circuito COP.

***XTAL* y *EXTAL*:** Estos pines proveen la interface, para un cristal o un reloj CMOS compatible, y así controlar el reloj interno del circuito generador. La frecuencia aplicada debe ser cuatro veces mas rápida que la tasa de reloj deseada.

***Reloj E*:** Este pin es la salida del generador de reloj E, el cual se puede usar como referencia de tiempo. La frecuencia de la salida del reloj E es una cuarta parte de la frecuencia de entrada a los pines XTAL y EXTAL.

***4XOUT*:** Este pin es la salida del buffer 4X del reloj (4 veces la frecuencia del reloj E), se puede usar para manejar la entrada del reloj de otro proceso. Esta salida se habilita fuera del reset y se puede deshabilitar por el bit CLK4X del registro OPT2. Se puede escribir en CLK4X una vez estando fuera de cada reset.

$\overline{IRQ}$: Este pin da la capacidad de producir una interrupción asíncrona en el MCU68HC11F1.

$\overline{XIRQ}$: Este pin da la capacidad para generar una interrupción asíncrona no mascarable para el MCU68HC11F1.

$\overline{MODA/LIR}$ y $\overline{MODB/V_{STBY}}$: Durante el reset, estos pines se usan para implementar los cuatro modos de operación. La salida $\overline{LIR}$ se puede usar como una ayuda para depurar una vez que el reset se complementa. El pin $\overline{LIR}$ es verificado bajo durante los primeros ciclos de reloj E de cada instrucción. La entrada V_{STBY} se usa para conservar la RAM mientras la energía está baja.

V_{RL} y V_{RH} : Estos pines dan la referencia de voltaje para el convertidor A/D.

$\overline{R/W}$: Este pin de salida se usa para controlar la dirección de la transferencia en el bus de datos externo en el modo Expanded Nonmultiplexed. Un nivel de voltaje bajo en este pin indica que los datos están siendo escritos para el bus de datos externo. Un nivel de voltaje alto en este pin indica que es programado un ciclo de lectura. Este pin se mantiene en alto en los modos Single Chip y Bootstrap.

Líneas de Entrada y Salida: Hay 54 líneas de entrada y salida las cuales están organizadas en seis puertos de 8 bits (los puertos A, B, C, E, F y G) y un puerto de 6 bits (el puerto D). La mayoría de esos puertos sirven para más de un propósito, dependiendo del modo de operación o la función periférica seleccionada.

7.3 Mapa de Registros del MCU68HC11F1

La Figura 7.2 muestra el mapa de memoria o registros para los cuatro modos en los que trabaja el MCU68HC11F1.

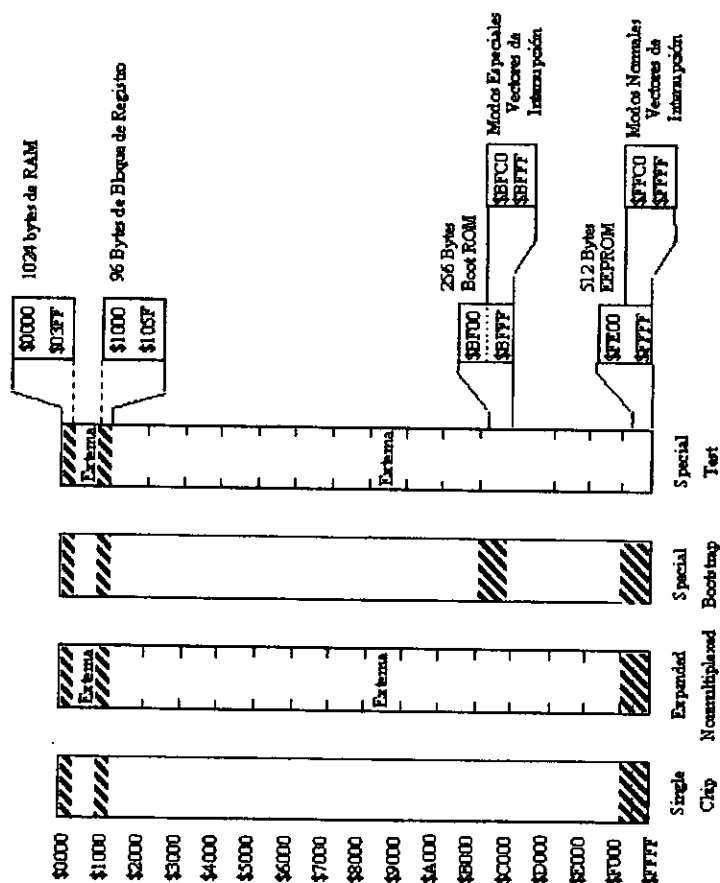


Figura 7.2: Mapa de Registros del MCU68HC11F1

La memoria consiste de 1Kb de RAM estática, 512 bytes de EEPROM, 96 registros bytes de estatus y control y 256 bytes de ROM bootloader (lo cuales están presentes en el modo Special Bootstrap). También cuenta con la capacidad de expandirse a 64Kb de memoria externa direccionable.

Con el registro INIT, el bloque de 96 registros (bytes) de control y estatus y el 1Kb de RAM estática son mapeables por 4Kb límites en la memoria. Sin embargo el reset se coloca en la RAM desde \$0000 hasta \$03FF y el espacio de registro desde \$1000 hasta \$105F, donde el 1 representa el valor decodificado de los cuatro bits de orden más bajo del registro INIT. La EEPROM se habilita por el bit EEON del registro CONFIG. En los modos Expanded Nonmultiplexed y Special Test el CONFIG se localiza desde \$xE00 hasta \$xFFF (donde x representa el valor de los cuatro bits mas significativos del registro CONFIG). En los modos Single Chip y Special Bootstrap, la EEPROM se localiza desde \$FE00 hasta \$FFFF. La ROM Bootstrap es mapeada por las localidades de memoria \$BF00-\$BFFF sobre el tránsito para el modo Special Bootstrap. Podrían crearse conflictos en el mapeo, si el bloque de registros toma prioridad sobre la RAM, y el Bootstrap ROM tiene prioridad sobre la EEPROM.

7.3.1 Registros de Control y Estatus

Los 96 registros de control y estatus se pueden relocar por cualquiera de los 4Kb límites en la memoria, pero normalmente se localizan en las direcciones \$1000-\$105F después del reset. Para indicar esta relocación las direcciones de cualquier registro remapeable deben comenzar con 1.

7.3.2 Registro de Mapeo I/O (INIT)

El registro INIT es un registro de propósito especial de 8 bits que se usa durante la inicialización para cambiar la localización de la RAM y el registro de control que hay tiene por default el MCU68HC11F1. Se puede escribir en él sólo una vez en los primeros 64 ciclos de reloj E después de un reset en los modos normales (Single Chip y Expanded Nonmultiplexed). Después, este registro es de sólo lectura.

	7	6	5	4	3	2	1	0	
\$103D	RAM3	RAM2	RAM1	RAM0	REG3	REG2	REG1	REG1	INIT
RESET:	0	0	0	0	0	0	0	1	

Cuando el registro INIT es puesto en \$01 por el reset, la dirección de comienzo por default para la RAM es \$0000 y \$1000 para los registros de control y estatus. Los bits RAM3-RAM0 especifican las direcciones de comienzo para el bloque de registros de control y estatus. En cada caso, los cuatro bits RAM o REG son los 4 bits más significativos de los 16 bits de direccionamiento de la RAM. Los 12 bits sobrantes expresan los tres caracteres finales de la dirección.

7.4 Puertos de Entrada/Salida

El MC68HC11F1 está equipado con seis puertos de Entrada/Salida de 8 bits (A, B, C, E, F y G) y un puerto de Entrada/Salida de 6 bits (D). Las funciones de los puertos de Entrada/Salida (B, C, F y G) están controladas por el modo particular de operación seleccionada. En los modos de Single Chip y Special Bootstrap ellos están configurados como puertos de datos de Entrada/Salida paralelos. En modos Expanded Nonmultiplexed y Special Test, dichos puertos y el pin $R/\overline{W}$ son configurados como un bus de expansión de memoria, con los puertos B y F como el bus de dirección, el puerto

C como el bus de datos, el pin $R/\overline{W}$ como dato de control de dirección del bus, y los cuatro bits más significativos del puerto G como selector de chip externo. El resto de los puertos no se afectan por los cambios de modos. Los puertos A, D y G se pueden usar como puertos de Entrada/Salida de propósito general, aunque cada uno tiene una función alternativa. El puerto E se puede usar para entradas estáticas de propósito general y/o entradas del canal del convertidor A/D. Los bits del puerto A controlan las funciones del temporizador. El puerto D maneja las funciones SPI (Interface Periférica Serial) y SCI (Interface de Comunicación Serial).

Cuando se usan las funciones de Entrada/Salida de propósito general, los puertos A, C, D y G están bajo el control de los registros de dirección de datos (DDR) y los registros de datos del puerto (PORT) A, C, D y G respectivamente.

7.4.1 Puerto A

El puerto A es un puerto de Entrada/Salida de propósito general de 8 bits con un registro de datos (PORTA) y un registro de dirección de datos (DDRA). Además, el puerto A se puede configurar para funciones de Captura de Entrada (IC) del Temporizador, funciones de Comparación de Salida (OC) del temporizador o la función del acumulador de pulsos.

7.4.1.1 Registro de Datos del Puerto A (PORTA)

El puerto A (PORTA) se puede leer en cualquier momento. Las entradas regresan el nivel de pin, mientras las salidas regresan el nivel de entrada del

pin controlador. Si se escribe, PORTA guarda el dato internamente. Este controla los pines solo si ellos están configurados como salidas. Las Escrituras a PORTA no cambian el estado de pin cuando los pines están configurados para Comparadores de Salida del Temporizador.

	7	6	5	4	3	2	1	0	
\$1000	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0	PORTA
RESET	0	0	0	0	0	0	0	0	
Fun. After.	PAI	OC2	OC3	OC4	OC5/IC4	IC1	IC2	IC3	
y/o	OC1	OC1	OC2	OC1	OC1	—	—	—	

7.4.1.2 Registro de Dirección de Datos del Puerto A (DDRA)

El temporizador fuerza a que el estado de Entrada/Salida sea una salida para cada bit del puerto asociado con un Comparador de Salida habilitado. En tal caso, los bits DDRA no se cambian, pero pierden el control inmediato del bit del puerto A (PORTA) asociado. El DDRA recupera el control del estado de Entrada/Salida del pin una vez que el Comparador de Salida de Temporizador asociado se deshabilita. Habilitando una función de captura de entrada no se fuerza la dirección de un pin del puerto A (PORTA) asociado.

	7	6	5	4	3	2	1	0	
\$1001	DDA7	DDA6	DDA5	DDA4	DDA3	DDA2	DDA1	DDA0	DDRA
RESET	0	0	0	0	0	0	0	0	

Poner un 1 en el correspondiente pin de Entrada/Salida del Puerto A se configura como salida y poner un 0 se configura como entrada solamente.

7.4.2 Puertos B y F

En el modo de Single Chip, todos los pines de los puertos B y F son pines de salida de propósito general. En el modo Expanded Nonmultiplexed, todos los pines de los puertos B y F funcionan como bits de direccionamiento del bus de dirección y el acceso a ambos puertos se trata como un acceso externo.

7.4.3 Puertos C y G

Los puertos C y G son puertos de Entrada/Salida de propósito general de 8 bits con un registro de datos y un registro de dirección de datos. En el modo de Single Chip, los pines de los puertos C y G son pines de Entrada/Salida de propósito general. En el modo Expanded Nonmultiplexed, los puertos C y G son buses de datos, y los accesos a ambos puertos se manejan como accesos externos. Los pines de datos bidireccional están controlados por la señal $R/\overline{W}$ en el modo Expanded Nonmultiplexed.

7.4.4 Puerto D

El puerto D es un puerto de propósito general de 6 bits con un registro de datos y un registro de dirección de datos. En todos los modos, los seis pines del puerto D se pueden usar para Entradas/Salidas de propósito general o para los subsistemas SCI (Interface de Comunicación Serial) y SPI (Interface Periférica Serial).

7.4.5 Puerto E

El puerto E se usa para entradas estáticas de propósito general (PE7-PE0) y/o entradas (AN7-AN0) para el convertidor Analógico/Digital (A/D) en todos los modos de operación. El puerto E no puede leer como entradas estáticas mientras ocurre una conversión A/D. Una lectura puede alterar una conversión que esté en progreso si ésta coincide con la porción de muestreo del ciclo de conversión.

	7	6	5	4	3	2	1	0	
\$100*	PE7	PE6	PE5	PE4	PE3	PE2	PE1	PE0	PORTE
RESET	0	0	0	0	0	0	0	0	
Fun. Alt.	AN7	AN6	AN5	AN4	AN3	AN2	AN1	AN0	

7.5 Convertidor Analógico a Digital

El MCU68HC11F1 contiene un convertidor A/D de muestreo de canal de 8 bits, de entradas multiplexadas, de aproximación sucesiva, y retención. Las líneas dedicadas V_{RL} y V_{RH} se proporcionan para la entrada de voltaje de referencia. Esos pines se usan en lugar de los pines de encendido del dispositivo, para incrementar la exactitud de la conversión A/D.

La conversión A/D de 8 bits del MCU68HC11F1 son precisos para con ± 1 LSB (Least Significant Bit, Bit Menos Significativo), usando el reloj E del sistema ($\pm \frac{1}{2}$ LSB errores de cuantización y $\pm \frac{1}{2}$ LSB los otros errores combinados). Cada conversión se completa en 32 ciclos de reloj del MCU. El bit CSEL del registro OPTION permite la selección de un oscilador interno RC que permite al A/D usarse con muy bajas tasas de ciclos de reloj del MCU68HC11F1. Un ciclo de conversión típica requiere de 16 μ s con un bus de frecuencia de 2 MHz.

El convertidor A/D es radiométrico sin códigos reservados. Un voltaje de entrada V_{RL} se convierte a \$00 y un voltaje de entrada igual a V_{RH} se convierte a \$FF, sin indicación de desbordamiento (overflow). La Figura 7.3 muestra la secuencia detallada para un conjunto de cuatro conversiones. esta secuencia comienza en el ciclo E de reloj después de una escritura en el registro Control/Status del A/D (ADCTL).

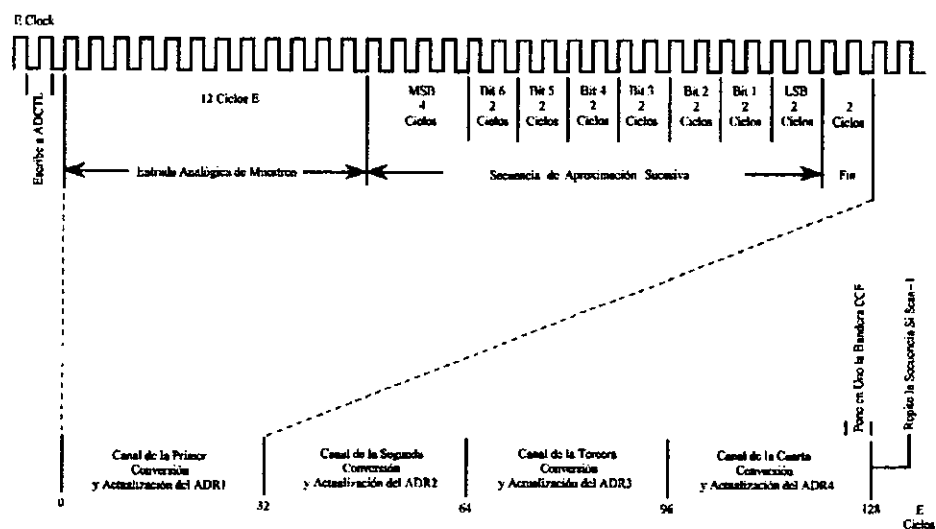


Figura 7.3: Secuencia de Conversión del A/D

7.5.1 Asignación de Canal

Un multiplexor permite al convertidor A/D seleccionar una de las 16 señales analógicas; 8 canales corresponden a las líneas de entrada del Puerto E en el MCU68HC11F1M 4 canales son para los puntos de referencia interna o funciones de prueba, y 4 canales son reservados para futuros usos. La Tabla 7.2 muestra como se seleccionan los canales por medio cuatro bits de control de selección de canal.

CD	CC	CB	CA	Señal de Canal	Resultado en ADRx si MULT = 1
0	0	0	0	AN0	ADR1
0	0	0	1	AN1	ADR2
0	0	1	0	AN2	ADR3
0	0	1	1	AN3	ADR4
0	1	0	0	AN4	ADR1
0	1	0	1	AN5	ADR2
0	1	1	0	AN6	ADR3
0	1	1	1	AN7	ADR4
1	0	0	0	Reservado	ADR1
1	0	0	1	Reservado	ADR2
1	0	1	0	Reservado	ADR3
1	0	1	1	Reservado	ADR4
1	1	0	0	V_{RH}^*	ADR1
1	1	0	1	V_{RL}^*	ADR2
1	1	1	0	$\frac{1}{2} V_{RH}^*$	ADR3
1	1	1	1	Reservado*	ADR4

* Este grupo de canales se usa durante las pruebas.

Tabla 7.2: Asignación de Canales del A/D

7.5.1.1 Operación de Un Canal

Hay dos variantes de la operación de un canal: en la primera (cuando SCAN = 0), el canal seleccionado se convierte cuatro veces consecutivas poniendo el primer resultado en el registro 1 de resultados del A/D (ADR1), el segundo en el ADR2, el tercero en el ADR3 y el cuarto en el ADR4 y cuando se ha completado la cuarta conversión, todas las actividades de conversión se detienen hasta que un nuevo comando de conversión se escribe en el ADCT; en la segunda variante (cuando SCAN = 1), las conversiones continúan haciéndose en el canal seleccionado con la quinta conversión poniéndose en ADR1, sobrescribiéndose en el primer resultado. La sexta conversión sobrescribe el contenido de ADR2, etc.

7.5.1.2 Operación de Múltiples Canales

Hay dos variantes en la operación multicanal: en la primera (cuando $SCAN = 0$), el grupo seleccionado de cuatro canales son convertidos, cada canal se convierte una vez donde el primer resultado se pone en el registro ADR1, el segundo en ADR2, etc., y después de que la cuarta conversión se completó, toda actividad de conversión se detiene hasta que un nuevo comando de conversión se escribe en el ADCTL; en la segunda (cuando $SCAN = 1$), las conversiones continúan haciéndose en el grupo de canales con la quinta conversión poniéndose en ADR1, sobrescribiéndose en el primer resultado. La sexta conversión sobrescribe el contenido de ADR2, etc.

7.5.2 Registro de Control/Estatus del A/D

Todos los bits de este registro se pueden leer o escribir, excepto el bit 7 (indicador de Estatus de sólo lectura) y el bit 6 (el cual siempre se lee como cero).

	7	6	5	4	3	2	1	0	
\$1030	CCF	0	SCAN	MULT	CD	CC	CB	CA	ADCTL
RESET	0	0	0	0	0	0	0	0	

CCF (Bandera de Conversiones Completamente realizadas): Este indicador de estatus de sólo lectura se pone en 1 cuando los cuatro registros de resultados contienen resultados válidos de comparación. Cada vez que el ADCTL es escrito, este bit es automáticamente puesto en 0, y se empieza una secuencia de conversión. En el modo continuo, las conversiones continúan de manera round-robin, y los registros de resultados continúan siendo actualizados con el dato actual aunque el bit CCF continúe en uno. El usuario debe escribir el registro ADCTL para inicializar la conversión. Para abortar una conversión en progreso, una escritura en el registro ADCTL inicializa una nueva secuencia de conversión inmediatamente.

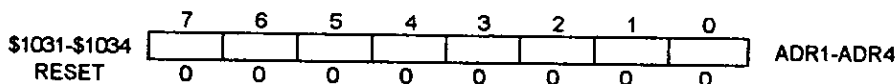
SCAN (Control de Revisión Continua): Cuando el bit de control es 0, las cuatro conversiones requeridas trabajan para llenar los cuatro registros de resultados. Cuando SCAN es 1, las conversiones continúan de manera round-robin y los registros de resultados se actualizan cuando llegan datos válidos.

MULT (Control de Múltiple-Canal/Único-Canal): Cuando este bit es 0, el sistema A/D está configurado para que se hagan cuatro conversiones consecutivas en el canal especificado por los bits de selección de canal (CD-CA). Cuando MULT es 1, el sistema A/D está configurado para hacer una conversión en cada uno de los cuatro canales, donde cada registro de resultado corresponde a un canal.

CD-CA (Selectores de Canal D-A): Estos 4 bits se usan para seleccionar uno de los 16 canales A/D. Cuando un modo multicanal se selecciona (MULT = 1), los dos bits menos significativos de selección de canal (CB y CA) no tienen sentido. Los bits CD y CC especifican que grupo de los cuatro canales están para ser convertidos.

7.5.3 Registros de Resultados del A/D

Los registros de resultados del A/D son registros de sólo lectura utilizados para mantener un resultado de conversión de 8 bits. La escritura a esos registros no tienen efecto. Los datos en los registro A/D son válidos cuando el bit CCF del ADCTL se pone en 1, indicativo de que una secuencia de conversión está completamente hecha. Por ejemplo, el resultado ADR1 es válido 33 ciclos después de escribir en el ADCTL.



7.5.4 Inicio de la Conversión A/D y Selección del Reloj

El inicio de una conversión A/D se controla por el bit 7 (ADPU) del registro de opciones (OPTION, \$1039). Cuando ADPU es 0, se deshabilita el sistema A/D y pero cuando es 1, se habilita. Se requiere un retardo aproximado de 100 μ s después de iniciar el convertidos A/D para estabilizar a los voltajes bias.

La selección de Reloj se controla por el bit 6 (CSEL) del registro de opciones (OPTION, \$1039). Cuando CSEL es 0, el sistema A/D utiliza el sistema de reloj E y cuando CSEL es 1, el sistema A/D utiliza una fuente de reloj interna RC que corre aproximadamente a 1.5 MHz. El reloj E del MCU68HC11F1 no es muy recomendable para manejar el sistema A/D si éste está operando abajo de los 750 kHz, para ello se debería seleccionar el reloj interno RC. Se requiere un retardo de 10 ms después de cambiar el CSEL de 0 a 1, para permitir que el oscilador RC empiece y el voltaje de bias interno se estabilice. El bit CSEL también habilita un oscilador RC por separado para poder manejar las burbujas de carga de la EEPROM.

Cuando el sistema A/D está operando con el reloj E del MCU68HC11F1, todas las operaciones de switcheo y comparadores están sincronizadas con los relojes del MCU68HC11F1. Esto permite que los resultados del comparador sean muestreados en cortos intervalos de tiempos de reloj para minimizar los errores de ruido. El oscilador interno RC es asíncrono al reloj del MCU68HC11F1, por lo que los ruidos afectan los resultados del A/D y sobretodo cuando CSEL es 1.

7.6 Temporizador Programable

Todas las funciones de tiempo están relacionadas a un contador de corrimiento libre de 16 bits. El contador de corrimiento libre se activa por la salida de un preescalador programable. El preescalador es activado por el reloj E y divide a este reloj por 1, 4, 8 o 16. Los bits de control del preescalador, se encuentran en el registro TMSK2 (Registro 2 de la Máscara de Interrupción del Temporizador) y sólo se pueden escribir una vez durante los primeros 64 ciclos de reloj E después de una inicialización del sistema (reset). El contador de libre corrimiento (registro TCNT) se puede leer por software en cualquier momento sin afectar su valor desde que éste se activa. El contador se pone en cero al inicializarse el sistema (reset), además es un registro de sólo lectura y se repite cada 65,536 cuentas. Cuando la cuenta cambia de \$FFFF a \$0000, el bit de la bandera de desbordamiento (overflow) del temporizador (TOF) del Registro 2 de la Bandera de Interrupción del Temporizador (TFLG2), se pone en 1. La bandera de desbordamiento también genera una interrupción interna si el bit de habilitación por Interrupción de Desbordamiento del Temporizador (TOI) de el Registro 2 de la Máscara de Interrupción del Temporizador (TMSK2) se pone en 1. El temporizador tiene 3 registros de Captura de Entrada, 4 de Comparación de Salida, 1 registro adicional que puede funcionar de ambas maneras, controlándolo por software.

7.6.1 Función de Captura de Entrada

Existen 3 registros de Captura de Entrada (IC) de sólo lectura de 16 bits (TIC1, TIC2 y TIC3) y 1 registro que puede servir ya sea como el cuarto registro de Captura de Entrada o el quinto registro de Comparación de Salida (TI4O5). Estos

registros no están inicializados al tener un reset del sistema. Cada registro de Captura de Entrada se usa para obtener el valor de contador de libre corrimiento cuando por alguno de los pines PA3-PA0 (entradas para IC4-IC1), se detecta una transición seleccionada de algún dispositivo externo. Una interrupción se puede generar cuando se detecta un flanco de captura de entrada. El momento de detección se puede leer del registro del temporizador apropiado como parte de la rutina de interrupción.

Los bits de control y estatus para implementar las funciones de Captura de Entrada están contenidas en los registros DDRA, PACTL, TCTL2, TMSK1 y TFLG1. Para configurar los bits 0, 1, 2, o 3 del puerto A como Capturas de Entrada, se deben poner en 0 los bits respectivos en DDRA, después de la inicialización del sistema (reset). Adicionalmente para habilitar PA3 como el cuarto registro de Captura de Entrada, el bit I4/O5 del Registro PACTL debe ser puesto en 1, pero si PA3 se quiere configurar como el quinto Comparador de Salida, después de la inicialización del sistema (reset), se debe poner en 0 el bit I4/O5. Si el bit DDA3 del registro DDRA se pone en 1 (configurando PA3 como una salida) habilitando el IC4, al escribir en PA3, se generan flancos en el pin, provocando así Capturas de Entradas. Cuando el registro TI4O5 está actuando como un IC4, éste no se puede escribir.

	7	6	5	4	3	2	1	0	
\$1021	EDG4B	EDG4A	EDG1B	EDG1A	EDG2B	EDG2A	EDG3B	EDG3A	TCTL2
RESET	0	0	0	0	0	0	0	0	

EDGxB y EDGxA (Control de Flancos para Captura de Entrada): Hay cuatro pares de esos bits. Cada par se pone en cero al inicializar el sistema (reset). Con estos bits se puede configurar el circuito detector de flancos para la Captura de Entrada a través de los IC4-IC1. La codificación está determinada por la Tabla 7.3.

EDGxB	EDGxA	Configuración
0	0	Captura Deshabilitada
0	1	Captura Sólo en Flancos de Subida
1	0	Captura Sólo en Flancos de Bajada
1	1	Captura en cualquier Flanco (subida o bajada)

Nota: El IC4 funciona sólo si el bit del registro del PACTL es 1.

Tabla 7.3: Configuración de los Flancos para la Captura de Entrada

7.6.2 Función de Comparación de Salida

Hay 4 registros de Comparación de Salida (OC) de lectura/escritura de 16 bits (TOC1, TOC2, TOC3 y TOC4) y 1 quinto registro (TI4O5) que puede funcionar bajo control de software tanto como IC4 como OC5. Cada uno de los registros de Comparación de Salida se en \$FFFF al inicializarse el sistema (reset). Un valor escrito en el registro de Comparación de Salida se compara con el valor del contador de libre corrimiento durante cada ciclo de reloj E. Si ocurre una igualdad, la bandera respectiva de Comparación de Salida se pone en 1 en el Registro 1 de Bandera de Interrupción del Temporizador (TFLG1), y se genera una interrupción, si esa interrupción particular se habilita en el Registro 1 de la Máscara de Interrupción del temporizador (TMSK1). Además de la interrupción, ocurre una acción específica en uno o varios pines de salida del temporizador. Para el OC5-OC2, la acción del pin se controla por pares de bits (OMx y OLx) en el Registro TCTL1. La acción de salida se toma en cada comparación exitosa, sin tener en cuenta que de todos modos la bandera OCxF en el Registro TFLG1 fue puesta previamente en 0.

El OC1 es diferente de los otros Comparadores de Salida ya que una comparación OC1 exitosa puede afectar a cualquiera de los otros cinco pines de comparación. La acción de salida OC1 se puede tomar cuando se genera una igualdad

y se controla por dos registros de cinco bits, el Registro de la Máscara 1 de Comparación de Salida (OC1M) y el Registro de Datos 1 de Comparación de Salida (OC1D). El OC1M especifica que salidas del puerto A están listas para usarse, y el OC1D especifica qué dato es colocado en esos pines del puerto.

Hasta la inicialización del sistema (reset), el bit I4/O5 de el PACTL se configura como OC5. La función del OC5 está habilitado cuando los bits OM5 y OL5 de el TCTL1 son apropiadamente puestos en 1. Aunque el DDRA configura todos los pines del puerto A como entradas fuera de la inicialización (reset), los pines del puerto A asignados a cada Comparación de Salida son forzados a ser salidas. Los bits del registro DDRA, no cambian, Los bits del DDRA se revierten para el control I/O una vez que el Comparador de Salida asociado se deshabilita.

Los bits de control y estatus para implementar las funciones de comparación de salida están contenidos en los Registros PACTL, CFORC, TCTL1, TMSK1 y el TFLG1.

7.6.2.1 Registro para Forzar la Comparación

Este registro de sólo escritura se usa para forzar anticipadamente las funciones de Comparación de Salida y no se recomienda para usarse con la función que conserva la Comparación de salida, porque una comparación normal ocurre antes o después de que el forzamiento pueda resultar en una operación no deseable.

	7	6	5	4	3	2	1	0	
\$100B	FOC1	FOC2	FOC3	FOC4	FOC5	0	0	0	CFORC
RESET	0	0	0	0	0	0	0	0	

FOC5-FOC1 (Acción de Forzamiento x de Comparación de Salida): Cuando el bit es 1, provoca la acción programada para que la Comparación de Salida x ocurra y la bandera x de Comparación de Salida (OCxF) del TFLG1 no se pone en 1. Pero cuando es 0 no afecta en nada.

7.6.2.2 Registro 1 de la Máscara de comparación de Salida

Este registro se usa con el OC1 para especificar los bits del puerto A que son afectados como resultado de una comparación OC1 exitosa. Los bits de el registro OC1M corresponden bit por bit con las líneas 7-3 del puerto A.

	7	6	5	4	3	2	1	0	
\$100C	OC1M7	OC1M6	OC1M5	OC1M4	OC1M3	0	0	0	OC1M
RESET	0	0	0	0	0	0	0	0	

OC1M7-OC1M3 (Máscaras de Comparación de Salida): Cuando el bit es 1 el respectivo OC1 está habilitado para controlar el correspondiente pin del puerto A; si el bit es 0, el respectivo OC1 está deshabilitado.

7.6.2.3 Registro 1 de Datos de Comparación de Salida

El OC1D se usa con OC1 para especificar el dato que está puesto en el pin del puerto A como resultado de una comparación OC1 exitosa. Cuando ocurre una comparación exitosa OC1, para cada bit que vale 1 en el OC1M, el correspondiente bit de dato en OC1D se puesto en el correspondiente bit del puerto A.

	7	6	5	4	3	2	1	0	
\$100D	OC1D7	OC1D6	OC1D5	OC1D4	OC1D3	0	0	0	OC1D
RESET	0	0	0	0	0	0	0	0	

7.6.3 Registro de Conteo del Temporizador

Este registro de sólo lectura de 16 bits contiene el valor preescalado del temporizador de 16 bits. Una lectura del contador completo primero debe direccionar el byte más significativo. Una lectura de esta dirección provoca que el byte menos significativo sea guardado en un buffer para el siguiente ciclo del CPU de tal forma que la lectura de un byte doble regrese el estado completo de 16 bits de el contador en el momento del ciclo de lectura del byte más significativo.

	15	14	13	12	11	10	9	8	
\$100E									TCNT
\$100F									TCNT
RESET	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	

7.6.3.1 Registro 1 de Control del Temporizador

Los bits de este registro se usan para especificar la acción tomada como resultado de una Comparación OCx exitosa.

	7	6	5	4	3	2	1	0	
\$1020	OM2	OL2	OM3	OL3	OM4	OL4	OM5	OL5	TCTL1
RESET	0	0	0	0	0	0	0	0	

OM2-OM5 (Modo de Salida).

OL2-OL5 (Nivel de Salida): Esos pares de bit de control se configuran para especificar que la acción tomada es el resultado de una comparación OCx exitosa. OC5 sólo funciona si el bit I4/O5 en el Registro PACTL es 0. La siguiente tabla muestra como configurar este registro.:

OMx	OLx	Acción hasta una Comparación Exitosa
0	0	Temporizador Desconectado de la Salida Lógica del Pin.
0	1	Retiene la Línea de Salida OCx.
1	0	Pone la Línea de Salida OCx en 0.
1	1	Pone la Línea de Salida OCx en 1.

7.6.3.2 Registro 1 de la Máscara de Interrupción del Temporizador

Este registro de 8 bits se usa para habilitar o deshabilitar las interrupciones de Captura de Entrada y Comparación de Salida del Temporizador.

	7	6	5	4	3	2	1	0	
\$1022	OC1I	OC2I	OC3I	OC4I	I4O5I	IC1I	IC2I	IC3I	TMSK1
RESET	0	0	0	0	0	0	0	0	

OCxI (Interrupción x de Comparación de Salida): Si el bit OCxF es 1, se requiere la secuencia de Interrupción, pero si es 0 se inhibe Interrupción.

ICxI (Interrupción x de Captura de Entrada): Si el bit ICxF es 1, se requiere la secuencia de Interrupción, pero si es 0 se inhibe Interrupción.

I4O5I (Interrupción de Captura de Entrada 4 o de Comparación de Salida 5):

Cuando el bit I4/O5 de PACTL es 1, el bit I4O5I funciona como el bit de interrupción IC4, pero cuando I4/O5 es 0, el bit I4O5I funciona como el bit de control de la interrupción OC5. Para ambos casos cuando el bit I4O5I es 1, se requiere la secuencia de Interrupción, pero cuando el bit I4O5I es 0 se inhibe la Interrupción.

7.6.3.3 Registro 1 de la Bandera de Interrupción del Temporizador

Este registro se usa para indicar la ocurrencia de eventos del sistema del Temporizador. Con el registro TMSK1, el TFLG1 permite al subsistema del temporizador operar en un sistema manejado por poléo o interrupción. Cada bit de el TFLG1 tiene un bit correspondiente en TMSK1 en la misma posición de bit.

	7	6	5	4	3	2	1	0	
\$1023	OC1F	OC2F	OC3F	OC4F	I4O5F	IC1F	IC2F	IC3F	TFLG1
RESET	0	0	0	0	0	0	0	0	

OCxF (Bandera x de Comparación de Salida): Los bits se ponen en 1 cada vez que el contador del temporizador es igual con un valor del registro x de Comparador de Salida y los mismos bits se ponen en 0 escribiendo un 1 en la posición de bit correspondiente.

I4O5F (Bandera de Captura de Entrada 4/Comparación de Salida 5): Este bit funciona como bandera ya sea para la Captura de Entrada 4 o la Comparación de Salida 5. La bandera se pone en 1 por hardware tanto para IC4 o OC5, dependiendo qué función está habilitada por el bit I4/O5 de PACTL en el momento de la interrupción. Este bit se pone en 1 al escribir un 1 en el bit 3.

ICxF (Bandera x de Captura de Entrada): Se pone un 1 cada vez que un flanco activo seleccionado se detecta en la línea de entrada ICx, esos bits se ponen en 0 escribiendo un 1 en la correspondiente posición de bit.

7.6.3.4 Registro 2 de la Máscara de Interrupción del Temporizador

Este registro se usa para controlar ya sea la secuencia de interrupción por hardware que es requerida, como el resultado de un bit de estatus que se pone en 1 en el registro de Bandera de Interrupción del Temporizador (TFLG1). Los dos bits del preescalador del Temporizador también se incluyen en este registro.

	7	6	5	4	3	2	1	0	
\$1024	TOI	RTI	PAOVI	PAI	0	0	PR1	PRO	TMSK2
RESET	0	0	0	0	0	0	0	0	

TOI (Interrupción de Desbordamiento del Temporizador Habilitado): Este bit se pone en 1 cuando se requiere la interrupción y el bit TOF de el TFLG2 es 1. Poniendo un 0 en este bit se deshabilita la interrupción.

RTII (Interrupción de Tiempo Real Habilitado): Este bit se pone en 1 cuando se requiere la interrupción y el bit RTIF de TFLG2 es 1. Cuando no se requiere de la interrupción este bit se pone en 0.

PAOVI. (Interrupción de Desbordamiento del Acumulador de Pulsos Habilitado): Este bit se pone en 1 cuando se requiere la interrupción y el bit PAOVF de TFLG2 es 1. Al poner un 0 en este bit se deshabilita la interrupción.

PAII (Interrupción del Acumulador de Pulsos Habilitado): Este bit se pone en 1 cuando se requiere la interrupción y el bit PAIF de TFLG2 es 1. Al poner un 0 en este bit se deshabilita la interrupción.

PR1 y PR0 (Selector del Preescalador del Temporizador): Esos bits se usan para seleccionar el preescalador y se pueden escribir solamente durante la inicialización. La escritura a esos bits se deshabilitan después de una escritura o después de 64 ciclos de reloj E después de la inicialización del sistema (reset).

PR1	PR0	Divide por
0	0	1
0	1	4
1	0	8
1	1	16

Tabla 7.4: Configuración para seleccionar el Preescalador del Temporizador

7.6.3.5 Registro 2 de la Bandera de Interrupción del Temporizador

Los bits de este registro indican la ocurrencia de los eventos del sistema del temporizador. Acoplados con los cuatro bits más significativos del TMSK2, los bits de el TFLG2 permite al subsistema del temporizador operar por poléo o por interrupción. Cada bit del TFLG2 corresponden al respectivo bit del TMSK2.

	7	6	5	4	3	2	1	0	
\$1025	TOF	RTIF	PAOVF	PAIF	0	0	0	0	TFLG2
RESET	0	0	0	0	0	0	0	0	

TOF (Desbordamiento del Temporizador): Este bit es 1 cada vez que el contador de libre corrimiento de 16 bits avanza de el valor \$FFFF - \$0000 y es 0 al escribir un 1 en el bit 7 de el TFLG2.

RTIF (Bandera de Interrupción de Tiempo Real): Este bit es 1 en cada flanco de subida del nivel seleccionado y es 0 al escribir un 1 en el bit 7 del TFLG2.

PAOVF (Bandera de Interrupción de Desbordamiento del Acumulador de Pulsos): Este bit es 1 cuando la cuenta en el acumulador de pulsos pasa de \$FF a \$00 y es 0 escribiendo un 1 en el bit 5 de el TFLG2.

PAIF (Bandera de Interrupción del Flanco de Entrada del Acumulador de Pulsos): Este bit es 1 cuando un flanco activo se detecta en el pin de entrada del PAI. En el modo eventual, el evento del flanco dispara a PAIF. En el modo de acumulador del temporizador controlado, el flanco de salida de la señal manipulada en la entrada PAI dispara el PAIF. Este bit se pone en 0 al escribir un 1 en el bit 4 del TFLG2.

7.6.4 Acumulador de Pulsos

El acumulador de Pulsos es un contador de 8 bits que puede operar en dos modos, dependiendo del estado del bit de control en el registro PACTL. Esos modos son el modo de conteo de eventos y el modo de acumulación de tiempos controlado. En el modo de conteo de eventos, el contador de 8 bits se activa para valores incrementales por medio de un pin externo. La máxima tasa de medición de tiempo para el modo de conteo de eventos externo es el reloj E dividido por dos. En el modo de acumulación de tiempo controlado, un reloj E de libre corrimiento dividido por 34 señales maneja el contador de 8 bits, pero sólo mientras el pin de entrada de PAI externa está activa.

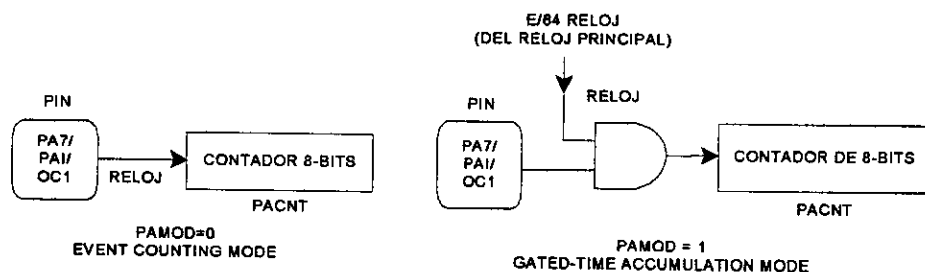


Figura 7.4: Modo de Operación del Acumulador de Pulsos

El acumulador de pulsos utiliza el bit 7 del puerto A (este pin también se puede usar como un pin de Entrada/Salida de propósitos generales o como un comparador de salida). Aún cuando el bit 7 del puerto A se configure como una salida, el pin continua manejando la entrada en el acumulador de pulsos.

7.6.4.1 Registro de Control del Acumulador de Pulsos

Tres bits de este registro controlan el sistema de acumulador de pulsos de 8 bits. Otro bit habilita, ya sea la función 5 de Comparador de Salida o la función 4 de Captura de Entrada. Los otros dos bits seleccionan la tasa para el sistema de interrupción en tiempo real.

	7	6	5	4	3	2	1	0	
\$1026	0	PAEN	PAMOD	PEDGE	0	I4/O5	RTR1	RTR0	PACTL
RESET	0	0	0	0	0	0	0	0	

PAEN (Sistema de Acumulador de Pulsos Habilitado): Si se pone un 1 en este bit se activa el Acumulador de Pulsos, pero si se pone un 0, se desactiva el Acumulador de Pulsos, ninguna bandera se pone en 1 y el contador se congela).

PAMOD (Modo del Acumulador de Pulsos): Si se pone en 1 este bit se activa el Modo de Acumulador de Tiempos Controlado, pero si se pone un 0m se activa el Modo de Contador de Eventos.

PEDGE (Control de Flancos del Acumulador de Pulsos): Dependiendo del estado de el bit PAMOD, este bit tiene diferentes funciones, tal como se muestra en la Tabla 7.5.

PAMOD	PEDGE	Acción en el Reloj
0	0	PAI Incrementa el Contador en el Flanco de Subida
0	1	PAI Incrementa el Contador en el Flanco de Bajada
1	0	Un 0 en el PAI Inhibe el Contéo
1	1	Un 1 en el PAI Inhibe el Contéo

Tabla 7.5: Configuración para el Control de Flancos del Acumulador de Pulsos

I4/O5 (Configuración del registro TI4O5 para IC u OC): Si se pone un 1 en este bit se habilita la función IC4, pero si se pone un 0 en este bit se habilita la función OC5.

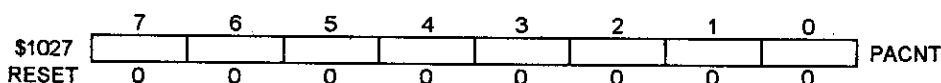
RTR1 y RTR0 (Tasa de Interrupción en Tiempo Real): El valor decodificado de esos pines seleccionan una de cuatro tasas para el circuito de interrupción periódica de tiempo real. Después del inicio (reset) del sistema, un periodo completo RTI transcurre antes de la primera interrupción RTI. Las tasas RTI se muestran en la Tabla 7.6.

RTR1	RTR0	Divide E por	XTAL= 2^{23}	XTAL = 8.0 MHz	XTAL = 4.9152 MHz	XTAL = 4.0 MHz	XTAL = 3.6864 MHz
0	0	2^{13}	2.91 ms	4.10 ms	ms	ms	ms
0	1	2^{14}	7.81 ms	ms	ms	ms	ms
1	0	2^{15}	15.62 ms	ms	ms	ms	ms
1	1	2^{16}	31.25 ms	ms	ms	ms	ms
		E -	2.1 MHz	2.0 MHz	1.2288 MHz	1.0 MHz	821.6 kHz

Tabla 7.6: Configuración de las Tasas RTI

7.6.4.2 Registro de Conteo del Acumulador de Pulsos

Este registro de Lectura/Escritura de 8 bits contiene la cuenta de eventos de entrada externos en la entrada PAI o la cuenta acumulada mientras la entrada PAI está activa en el modo de acumulación de tiempo controlado.



7.7 Interface de Comunicaciones Serial

La Interface de Comunicaciones Serial (SCI) se usa para que el MCU68HC11F1 pueda comunicarse con dispositivos periféricos que requieran de un formato asíncrono de datos consecutivos. El SCI usa el estándar de no retorno a cero (NRZ) con una velocidad de transmisión en baudios variable, derivada del cristal de reloj. La comunicación se lleva acabo usando los pines del puerto D. PD0 se usa para recibir datos (RxD), y PD1 para transmitir datos (TxD). El circuito generador de la tasa de baudaje contiene un programa preescalable y un divisor de frecuencia de la frecuencia E.

7.7.1 Formato de los Datos

El formato de los datos a transmitir en serie es el siguiente:

- 1) Una línea en estado alto para la recepción o transmisión de un mensaje.
- 2) Un bit de inicio (0 lógico) que es transmitido o recibido, indicando el comienzo de cada caracter.
- 3) Los datos que son transmitidos y recibidos por LSB (Bit Menos Significativo).
- 4) Un bit de paro (1 lógico) para indicar el fin de un marco (un marco consiste de un bit de inicio, un carácter de ocho o nueve bits y un bit de paro).
- 5) Un rompimiento, el cual se define como la transmisión o recepción de un cero lógico para múltiples marcos.

La selección de la longitud de la palabra se controla por el bit M del registro de control SCI 1 (SCCR1).

7.7.2 Operación de la Transmisión

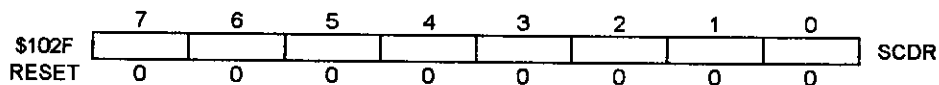
El transmisor de la SCI incluye un registro paralelo transmisor de datos llamado SCDR y un registro de corrimiento serial que pone datos desde el SCDR de forma consecutiva. El contenido del registro de corrimiento serial puede ser de sólo escritura a través del SCDR. Este sistema de doble buffer permite que un caracter salga mediante corrimientos, mientras que otro caracter espera en el SCDR para ser transferido dentro del registro de corrimiento serial. La salida del registro de corrimiento serial es por PD1. Mientras se está transmitiendo o el bit de transmisión (TE) esté habilitado, el registro 2 de Control de Comunicaciones Serial (SCCR2) estará en 1.

7.7.3 Registros SCI

Cuando se trabaja con la SCI, normalmente se utilizan 4 registros, con los cuales se establece diferentes características de los datos a transmitir o recibir. Además se utiliza otro registro que es donde se pone el dato a transmitir o el dato a recibir por medio del SCI.

7.7.3.1 Registro Serial de Comunicación de Datos

Este registro realiza dos funciones. recibe datos cuando es leído y transmite datos cuando se escribe en él.



7.7.3.2 Registro de Control de Comunicaciones Serial (SCCR1)

El registro SCCR1 provee el control de bits para determinar la longitud de la palabra y selecciona el método usado para la función Wakeup.

	7	6	5	4	3	2	1	0	
\$102C	R8	T8	0	M	WAKE	0	0	0	SCCR1
RESET	0	0	0	0	0	0	0	0	

R8 (Bit 8 del Dato Recibido): Este bit funciona como el noveno bit serial de datos recibido, cuando el sistema SCI se configura para operar con 9 bits de datos.

T8 (Bit 8 del Dato de Transmítido): Este bit funciona como el noveno bit de dato a transmitir, cuando el sistema SCI se configura para operar con 9 bits de datos.

M (Longitud del Caracter del SCI): Al poner un 1, se establece que el formato es del dato es de un bit de inicio, nueve bits de datos y un bit de paro. Si se pone un 0 se establece que el formato de datos es de un bit de inicio, ocho bits de datos y un bit de paro.

Wake (Método Seleccionado del Wakeup): Al poner un 1 se indica la Marca de Dirección, pero si se pone un 0m la línea está ociosa.

La función Wakeup reduce el servicio del SCI en muchos sistemas receptores. El software de cada receptor evalúa el o los primeros caracteres recibidos de cada mensaje. Si el mensaje se quiere poner en un receptor diferente, el SCI se puede poner en un modo suspendido, parando el resto del mensaje generando la petición para el servicio. Cuando un nuevo mensaje comienza, la lógica hace que los receptores se suspendan para reactivarse y evaluar el o los caracteres iniciales del nuevo mensaje.

Están disponibles dos métodos del Wakeup: un Wakeup de línea ociosa o un Wakeup de marca de dirección. En el Wakeup de línea ociosa, un receptor suspendido se activa tan pronto como la línea RxD comienza a trabajar. En el Wakeup de marca de dirección, un 1 en el MSB (bit más significativo) de un carácter se usa para indicar que el mensaje es una dirección que activa a todos los receptores suspendidos.

7.7.3.3 Registro 2 de Control de Comunicaciones Serial (SCCR2)

Este registro proporciona los bits de control que habilitan o deshabilitan las funciones individuales del SCI.

	7	6	5	4	3	2	1	0	
\$102D	TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK	SCCR2
RESET	0	0	0	0	0	0	0	0	

TIE (Habilitador de la Interrupción de Transmisión): Cuando este bit es 1 se hace la petición de interrupción del SCI, si el bit TDRE del SCSR es 1; pero cuando es 0 se deshabilita la interrupción.

TCIE (Habilitador de Interrupción de Transmisión Completa): Cuando este bit es 1 se hace la petición de interrupción del SCI, si está en 1 el bit TC del SCSR. Cuando este bit es 0 se deshabilita la interrupción.

RIE (Habilitador de la Interrupción de Recepción): Este bit se pone en 1 para solicitar la interrupción del SCI (si está en 1 el bit RDRF o el bit OR del SCSR). Este bit se pone en 0 para indicar que el registro de dato del receptor está lleno o para deshabilitar la interrupción de exceso de error (OR).

ILIE (Habilitador de la Interrupción Línea Ociosa): Este bit se pone en 1 para solicitar la interrupción SCI (si está en 1 el bit IDLE del SCSR). Este bit se pone en 0 para inhibir la interrupción de detección de una línea ociosa (IDLE).

TE (Habilitador del Transmisor): Se pone un 1 en este bit para indicar que la transmisión de la salida del registro de corrimiento es por la línea TxD, y se pone un 0 en este bit para indicar el revertimiento del pin PD1 de la Entrada/Salida de propósito general al terminar la transmisión actual.

RE (Habilitador del Receptor): Se pone un 1 en este bit para habilitar el Receptor, y se pone un 0 para deshabilitar el Receptor e inhibir las interrupciones RDRF, IDLE, OR, NF y FE.

RWU (Wakeup de Recepción): Cuando este bit se ponen en 1, este bit pone al receptor a descansar y habilita la función wakeup. Si el bit WAKE del SCCRL es un 0, el RWU se pone en 0 por la lógica del SCI después de recibir 10 (SCCR1 bit M = 0) u 11 (SCCR1 bit M=1) unos consecutivos. Si el WAKE se pone en 1, el RWU se pone en 0 por la lógica del SCI después de recibir una palabra de dato en la cual MBS es se pone en 1.

SBK (Paro de la Señal Enviada): Si este bit se pone en 1 o 0, el transmisor manda 10 u 11 ceros consecutivos, dependiendo del bit M del SCCR1, entonces el receptor está ocioso o mandando datos. Si el SBK queda en 1, el transmisor continúa mandando los marcos completos de ceros en grupos de 10 u 11 hasta que SBK se pone en 0.

7.7.3.4 Registro de Estatus de Comunicaciones Serial (SCSR)

Este registro proporciona entradas a los circuitos lógicos de interrupciones para la generación del sistema de interrupción del SCI.

	7	6	5	4	3	2	1	0	
\$102E	TDRE	TC	RDRF	IDLE	OR	NF	FE	0	SCSR
RESET	1	1	0	0	0	0	0	0	

TDRE (Registro transmisor de datos vacío): Este bit es 1 automáticamente mientras el contenido del SCDR es transferido al registro de corrimiento serial en una operación de transmisión. No se transmiten nuevos datos a menos que el registro SCSR sea leído antes de escribir al SCDR. Este bit es 0 al leer el SCSR después de una escritura al SCDR.

TC (Transmisión Completa): Este bit es 1 automáticamente cuando se completa todo el marco de datos o una condición de rompimiento de transmisión. Cuando TC es 1, la línea serial va desocupada (marca continua). TC es puesto en 0 por una lectura del SCSR seguida de una escritura a el SCDR.

RDRF (Registro Receptor de Datos Completo): Este bit es 1 automáticamente cuando un carácter es transmitido desde el registro de corrimiento serial a el SCDR en una operación de recepción. Este bit es 0 al leer el SCSR después de una lectura de SCDR.

IDLE (Detección de una Línea Ociosa): Este bit es 0 mientras el bit RWU de SCCR2 está en 1. Este bit es 1 automáticamente cuando la entrada serial recibida detecta una línea ociosa. Este bit se pone en 0 por una lectura del SCSR seguida de una lectura del SCDR.

OR (Exceso de Errores): Este bit es puesto en 1 automáticamente cuando un nuevo caracter está listo para ser transferido desde el registro de corrimiento del receptor hacia el SCDR y se ha completado el SCDR (se pone en 1 el bit RDRF). La transferencia de datos es 0 hasta que el bit. OR se pone en 0 por una lectura al SCSR seguida de una lectura al SCDR.

NF (Bandera de Ruido): Este bit es automáticamente puesto en 1 si hay ruido en alguno de los bits recibidos, incluyendo los bits de inicio y paro. Este bit no es puesto en 1 hasta que la bandera RDRF es 1. Este bit es 0 en caso de un exceso de errores y también es puesto en 0 por una lectura al SCSR seguida de una escritura al SCDR.

FE (Marco de Error): Este bit es 1 automáticamente cuando el bit de no paro se detecta en el caracter de dato recibido. El FE es puesto en 1 al mismo tiempo que la bandera RDRF es puesto en 1. Si el byte recibido causa exceso de errores y marco de errores, solo se reconoce el exceso de errores. Esta bandera se inhibe y se transfiere hasta que es 0. FE es puesto en 0 por una lectura al SCSR seguida de una escritura al SCDR.

7.7.3.5 Registro de Tasa de Baudaje (BAUD)

Este registro es para seleccionar diferentes tasas de baudaje que se usan como control para la recepción y transmisión.

	7	6	5	4	3	2	1	0	
\$102B	TCLR	0	SCP1	SCP0	RCKB	SCR2	SCR1	SCR0	BAUD
RESET	0	0	0	0	0	0	0	0	

TCLR (Limpiador del Contador de la Tasa de Baudaje): Este bit está deshabilitado y permanece en 0 en los modos de Special Test o Special Bootstrap. El reset pone en 0 este bit. Cuando se pone este bit en 1 el contador de la tasa de baudaje se reinicializa. Después de una lectura, este bit siempre regresa a 0 lógico.

SCP1 y SCP0 (Selección de la tasa de Baudaje Preescalador del SCI) Estos bits controlan un preescalador, el cual a través de su salida es la entrada de un segundo divisor que es controlado por los bits SCR2-SCR0 (Tabla 7.7).

RCKB (Reloj Verificador de la Tasa de Baudaje del SCI): Este bit se usa durante el modo Special Test para habilitar la OR exclusiva del reloj del receptor y del transmisor y se pueda manejar fuera del pin TxD. RCKB es 0 y no puede ser 1 en los modos normales de operación (Single Chip y Expanded Nonmultiplexed).

SCR2-SCR0 (Selección de la Tasa de Baudaje del SCI): Estos bits seleccionan la tasa de baudaje para la transmisión y recepción. La salida del preescalador seleccionada por SCP1 y SCP0 se divide por el conjunto de estos bits.

SC P1	SC P0	SC R2	SC R1	SC R0	Frecuencia del Cristal				
					2 ²² Hz	8 MHz	4.9152 MHz	4MHz	3.6864 MHz
					Tasa de Bandaaje [Band]				
0	0	0	0	0	131.072K	125.00k	76.80k	62.50k	57.60k
0	0	0	0	1	65.536k	62.50k	38.40k	31.25k	28.80k
0	0	0	1	0	32.768k	31.25k	19.20k	15.625k	14.40k
0	0	0	1	1	16.384k	15.625k	9600	7812.5	7200
0	0	1	0	0	8192	7812.5	4800	3906	3600
0	0	1	0	1	4096	3906	2400	1953	1800
0	0	1	1	0	2048	1953	1200	977	900
0	0	1	1	1	1024	977	600	488	450
0	1	0	0	0	43.691k	41.666k	25.60k	20.833k	19.20k
0	1	0	0	1	21.845k	20.833k	12.80k	10.417k	9600
0	1	0	1	0	10.923k	10.417k	6400	5208	4800
0	1	0	1	1	5461	5208	3200	2604	2400
0	1	1	0	0	2731	2604	1600	1302	1200
0	1	1	0	1	1365	1302	800	651	600
0	1	1	1	0	683	651	400	326	300
0	1	1	1	1	341	326	200	163	150
1	0	0	0	0	32.768k	31.250k	19.20k	15.625k	14.40k
1	0	0	0	1	16.384k	15.625k	9600	7812.5	7200
1	0	0	1	0	8192	7812.5	4800	3906	3600
1	0	0	1	1	4096	3906	2400	1953	1800
1	0	1	0	0	2048	1953	1200	977	900
1	0	1	0	1	1024	977	600	488	450
1	0	1	1	0	512	488	300	244	225
1	0	1	1	1	256	244	150	122	112.5
1	1	0	0	0	10.082k	9600(+0.16%)	5908	4800(+0.16%)	4431
1	1	0	0	1	5041	4800	2954	2400	2215
1	1	0	1	0	2521	2400	1477	1200	1108
1	1	0	1	1	1260	1200	738	600	554
1	1	1	0	0	630	600	369	300	277
1	1	1	0	1	315	300	185	150	138
1	1	1	1	0	158	150	92	75	69
1	1	1	1	1	79	75	46	38	35
					2.1 MHz	2 MHz	1.2288 MHz	1 MHz	921.6 kHz
					Frecuencia de Bus (Frecuencia E)				

Tabla 7.7: Tasa de Baudaje Definida por SCP1-SCP2, SCR2, SCR1, SCR0 y el Cristal de Frecuencia

Capítulo 8:

Generalidades del Lenguaje de Programación Visual Basic

Visual Basic es un sistema de programación para Computadoras Personales, que permite la creación de entornos gráficos en ambiente de Microsoft Windows para la realización de múltiples tipos de aplicaciones. Actualmente en el mercado existe hasta la versión 4.0. Este capítulo trata objetos de la Versión 3.0 y 4.0, y no se hace distinción de versiones, porque básicamente los objetos que se describen se manejan en ambas versiones.

Las versiones 3.0 y 4.0 corren en versiones de Windows 3.1x, Windows 95. Y Windows NT. La versión 4.0 salió en diferentes versiones como son la Professional y la Enterprise, además de que salieron dos modalidades de la versión 4.0: la versión 4.0 de 16 bits y la versión 4.0 de 32 bits.

A continuación se dará una reseña histórica del Basic, además de una explicación de las ventanas necesarias para hacer aplicaciones en Visual Basic y los objetos más comunes en Visual Basic

8.1 Historia

En los años 50's fueron creados los primeros lenguajes de programación, principalmente para resolver problemas matemáticos. Dichos lenguajes eran complejos y poco entendibles para los usuarios, sin embargo no representó mayor problema, puesto que ellos sólo eran científicos e investigadores, quienes en muchos casos eran los mismos que los habían inventado. A medida de que la tecnología informática se fue convirtiendo en una herramienta útil en las empresas, los complicados lenguajes llegaron a ser un obstáculo.

Para darle solución a este problema, a principios de los años 60's se desarrolló el lenguaje de programación Basic (acrónimo de Beginner's All-purpose Symbolic Instruction Code) en el Dartmouth College. En su versión original Basic era un lenguaje muy simple, diseñado especialmente para hacer una programación sencilla. Basic sirvió para que muchos programadores hicieran una gran variedad de programas.

El que Basic fuera simple lo hizo pequeño, siendo importante el tamaño cuando las computadoras también comenzaron a ser pequeñas. El MITS Altair (revolución de las microcomputadoras) apareció en 1975. Fue entonces cuando Bill Gates y Paul Allen, cofundadores de Microsoft, desarrollaron la versión de Basic para Altair; dicha versión funcionaba en los 4 Kb de RAM de los que disponía la computadora. Esta versión de Basic se convirtió en el producto más usado de la industria de las computadoras personales. Con el transcurso del tiempo BASIC se ha ido mejorando. Cuando surgió la computadora personal de IBM, el GWBASIC de

Microsoft estableció el estándar. Más adelante Microsoft desarrolló el QuickBasic que fue líder en tecnología de programación en los años 80's.

A fines de la década de los 80's, apareció tecnología de Interface Gráfica de Usuario (GUI). Por medio de una interface gráfica de usuario, es posible que las aplicaciones sean fáciles de aprender y de usar. En lugar de que el usuario aprenda una serie de comandos, simplemente tiene que seleccionar opciones de un menú con un clic de un botón del ratón. Diversas ventanas en la pantalla permiten que el usuario pueda ejecutar más de un programa a la vez. Cuando un programa necesita información o necesita que el usuario tome decisiones aparecen los cuadros de diálogo. Con este nueva interface gráfica de usuario, los entornos se volvieron maravillosos para el usuario, pero la programación se volvió más complicada, puesto que ahora se tenían que crear ventanas, menus, fuentes, cuadros de diálogo y una multitud de elementos, incluso para programas simples.

Cuando surgió Microsoft Windows, que es una plataforma para escribir aplicaciones gráficas, la programación todavía se volvió más complicada, por ejemplo, para mandar un mensaje a pantalla en cualquier lenguaje, sólo se empleaban cuatro líneas de código y un programa similar para Microsoft Windows requería dos o tres páginas de código y además implicaba aprender a controlar fuentes, menús, ventanas, memoria y otros recursos del sistema. Sin embargo la ventajas que Microsoft Windows le ofrecía a los usuarios finales era incuestionable.

Como solución a esta problemática en la programación, en 1991 Microsoft presentó Visual Basic. Visual Basic es un sistema de programación que hace que la complejidad de Windows sea más simple para los programadores. Visual Basic logra que los menús, fuentes, cuadros de diálogo, campos de texto con desplazamiento y algunas otras cosas se diseñen con facilidad, y con un requerimiento mínimo de código. Visual Basic es un lenguaje de programación orientada a eventos, el cual es un estilo de programación especialmente adaptado a las interfaces gráficas de usuario.

La programación orientada a eventos proporciona a los usuarios la libertad de determinar en cada momento, qué es lo que se quiere hacer, cosa que los lenguajes tradicionales no permiten. Con los lenguajes orientados a eventos el programador desarrolla programas que responden a las acciones del usuario como elegir un comando, hacer clic en una ventana, mover el ratón, etc.

8.2 Creación de Aplicaciones con Visual Basic

Los primeros pasos a seguir en la creación de una aplicación en Visual Basic son:

- ♦ Creación de un nuevo proyecto para organizar las partes de la aplicación
- ♦ Creación de una forma para cada ventana en la aplicación.
- ♦ Creación de un menú para la forma principal.

8.2.1 Proyectos

Un proyecto en Visual Basic es un conjunto de formas y módulos que se relacionan entre sí, para formar una aplicación. Los proyectos tienen una ventana que contienen una lista de todos los archivos necesarios para ejecutar la aplicación. En la ventana proyecto (Figura 8.1) se presentan el nombre del archivo, la lista de todas las formas, módulos (en la versión 3.0 se presentan también los controles de extensión de Visual Basic). También contiene dos botones que están etiquetados con View Form (ver forma) y View Code (ver código). El primero muestra el diseño de la forma seleccionada (este botón sólo está activo cuando está marcada una forma) y el segundo es para ver directamente la ventana de código.

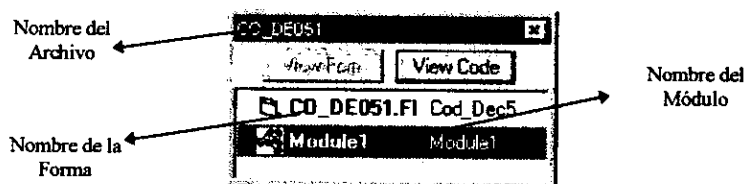


Figura 8.1: Ventana de Proyecto

8.2.2 Formas

El objeto Form (Forma) es una ventana o caja de diálogo que da presentación a la aplicación de una interface de usuario. También se puede definir una forma como un contenedor de controles. Existen dos tipos de formas: las formas MDI (Interfaz de Múltiples Documentos) y las normales. Ambas actúan de similar forma. En ambas se pueden introducir objetos para la realización de ciertas tareas.

Una forma MDI además de objetos, también puede contener formas hijas. Una forma MDI es una ventana que actúa como base de una aplicación y es el contenedor de formas hijas que tienen su propiedad MDIChild como verdadera. Sólo puede haber una única forma MDI, pero puede haber muchas formas MDI hijas. Si una forma MDI hija tiene menús, la barra del menú de la forma hija automáticamente reemplaza la barra de menú de la forma MDI cuando se activa la forma hija. Las formas MDI hijas minimizadas aparecen como un icono dentro de la forma MDI. Las formas MDI hijas son diseñadas independientemente de las formas MDI, pero están siempre en la forma MDI cuando se ejecuta.

Todos los objetos que contienen las formas se deben de insertar en la ventana de Forma o Form (Figura 8.2).

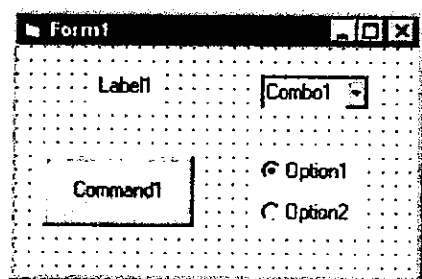


Figura 8.2: Ventana Form con algunos objetos insertados

8.2.3 Caja de Herramientas

La caja de herramientas Figura 8.3 muestra todos los controles estándar de Visual Basic más algunos controles (Custom Controls) que se pueden agregar al proyecto. Hay 21 herramientas básicas, las cuales se describen a continuación.

Pointer (Apuntador): Este control sirve para mover o modificar el tamaño de un control después de ser dibujado.

Picture Box (Caja de Dibujos): Se usa para desplegar imágenes gráficas, como gráficas obtenidas de algún método gráfico u otros controles.

Label (Etiqueta): Se usa para textos que no se cambian, como por ejemplo títulos de alguna gráfica.

Text Box (Caja de Texto): Se usa para textos, el cual el usuario puede aceptar o cambiar.

Frame (Marco): Se usa para crear un grupo de funciones o controles. Primero se crea el Frame y luego se dibujan los controles dentro de él.

Command Button (Botón de Comando): Se usan para crear botones en donde el usuario puede escoger alguno y así se ejecute cierta acción.

Check Box (Caja de Verificación): Se usa para cuando se tienen opciones que el usuario puede fácilmente seleccionar indicándolas como verdadera o falsa..

Option Button (Botón de Opciones): Se usa para cuando se tiene un grupo opciones y se quiere seleccionar sólo una.

Combo Box (Caja Combinada): Este objeto es un cuadro combinado formado por una lista y una caja de texto. El usuario puede seleccionar algo de la lista o introducir un valor en la caja de texto, dependiendo del número de la propiedad Style que se tenga.

List Box (Caja de Lista): Se usa para desplegar una lista de cosas, de la cual el usuario puede seleccionar una o varias. La lista puede ser recorrida, si tiene más cosas de las que pueden desplegarse a la vez.

Horizontal Scroll Bar (Barra de Desplazamiento Horizontal) y Vertical Scroll Bar (Barra de Desplazamiento Vertical): Se usan para aplicaciones por ejemplo, en donde se indique la posición en una escala o como un dispositivo de entrada o para indicar velocidad o cantidad.

Timer (Temporizador): Se usa para ejecutar cierto código durante intervalos regulares de tiempo.

Drive List Box (Caja de Lista de Unidades de Discos): Permite al usuario seleccionar una unidad de discos en tiempo de ejecución. Se utiliza este control para desplegar una lista de todas las unidades de la computadora.

Directory List Box (Caja de Lista de Directorios): Sirve para desplegar directorios y rutas en tiempo de ejecución. Se utiliza este control para desplegar una lista jerárquica de directorios.

File List Box (Caja de Lista de Archivos): Sirve para localizar archivos en un directorio específico, durante el tiempo de ejecución.. Se utiliza para desplegar una lista de archivos seleccionados bajo cierto formato, como pueden ser las extensiones de archivos.

Shape (Forma): Se usa para dibujar una variedad de figuras en las formas durante su diseño. Se pueden dibujar rectángulos, rectángulos con esquinas redondeadas, cuadros, cuadros con esquinas redondeadas, óvalos o círculos.

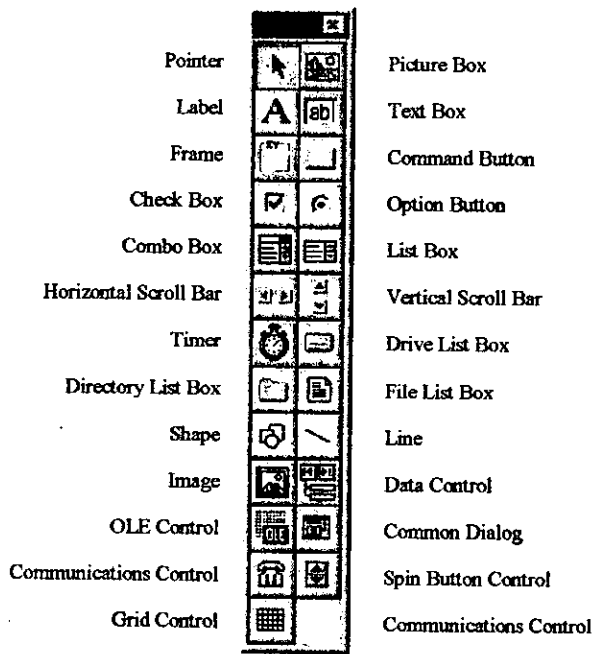


Figura 8.3: Herramientas de Visual Basic

Line (Línea): Se usa para dibujar una variedad de estilos de líneas en una forma cuando se está diseñando.

Image (Imagen): Se usa para desplegar en una forma una imagen gráfica, desde un bitmap, icono o metafile. Una imagen desplegada en un control image sólo puede ser de adorno y usa menos recursos que una caja picture.

Data Control (Control de Datos): Se usa para proveer el acceso a datos en una base de datos, es decir que el Data Control permite moverse de registro en registro para desplegar y manipular datos de registros de una tabla de base de datos.

OLE Control (Control OLE): Las siglas OLE significa Object Linking and Embedding (Enlace e Incrustación de Objetos). Este control se usa para proveer objetos ligando o insertándolos (OLE) de un servidor OLE dentro de la aplicación Visual Basic.

Common Dialog (Ventana de Dialogo Común): Se usa para crear tipos comunes de cajas de dialogo para abrir o cerrar archivos y escoger fuentes o colores.

Communications Control (Control de Comunicaciones): Este control proporciona comunicaciones seriales para las aplicaciones al transmitir y recibir datos a través del puerto serial.

Spin Button Control (Control de Botón de Giro): Es un control girador que puede usarse con otros controles para incrementar y decrementar números. También puede ser usado para desplazarse a través de un rango de valores o de una lista de términos.

Grid Control (Control de Celdas): Se usa para crear un hoja de calculo que contiene columnas y renglones de datos.

8.2.4 Propiedades

Las propiedades son los atributos de un objeto, es decir, las propiedades definen características de los objetos; tales como tamaño, color, localización. También definen el comportamiento, como si está habilitado o no, si está visible o no, etc. Cada objeto tiene una lista de propiedades las cuales se muestran en la ventana de propiedades. Las propiedades se pueden cambiar a la hora del diseño. Cuando se seleccionan varios objetos la ventana de propiedades muestra las propiedades que son comunes en los objetos que fueron seleccionados.

La ventana de propiedades (Figura 8.4) consiste de:

- 1) Object Box (Caja de Objeto), ésta se localiza abajo de la barra del título e identifica la forma o el control que actualmente se encuentra seleccionado.
- 2) Settings Box (Caja de Especificaciones), ésta se localiza abajo del Object Box y es donde se hace el cambio de la propiedad seleccionada.
- 3) Property List (Lista de Propiedades), ésta se localiza al pie de la ventana de propiedades y se forma por dos columnas que muestran todas las propiedades que se pueden cambiar a la hora del diseño.

A continuación se describen algunas propiedades que son comunes para muchos objetos.

BackColor (Color de Fondo): Determina el color de fondo de un objeto. y se aplica a: form, check box, combo box, command button, data control, directory list box, drive list box, file list box, grid, label, list box, OLE control, option button, picture box, shape, text box.

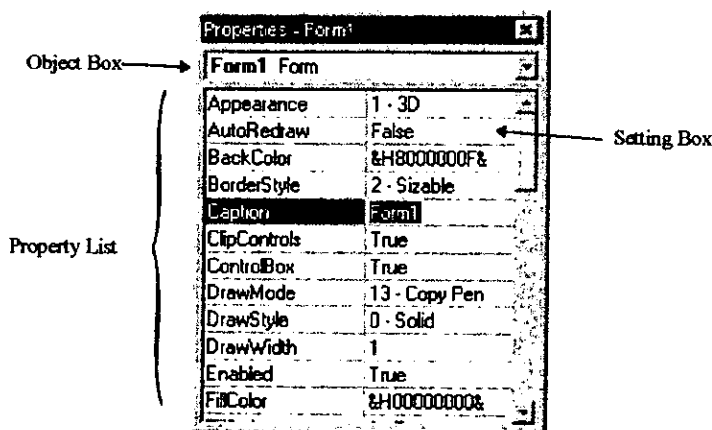


Figura 8.4: Ventana de Propiedades

ForeColor (Color Frontal): Determina el color del primer plano y se usa para desplegar texto y gráficas en un objeto.

Caption (Etiqueta): Se aplica a: form, MDI form, check box, command button, data control, frame, label, menu y option button. En formas determina el texto desplegado en la barra de título de la forma. Cuando la forma se minimiza, este texto se despliega debajo del icono de la forma. En controles determina el texto desplegado en o junto al control.

DragIcon (Icono de Arrastre): Se aplica a: check box, combo box, command button, data control, directory list box, drive list box, file list box, frame, grid, horizontal y vertical scroll bar, image, label, list box, OLE control, option button, picture box y text box. Determina el icono que va a ser mostrado como el indicador en una operación de agarre y arrastre.

Enabled (Habilitado): Se aplica a: form, MDI form, check box, combo box, command button, data control, directory list box, drive list box, file list box, frame, grid, horizontal y vertical scroll bar, image, label, list box, menu, option button, picture box, text box y timer. Determina si la forma o control puede responder a algún evento generado por el usuario.

FontName (Nombre de la Fuente de Letra): Se aplica a: form, check box, combo box, command button, common dialog, data control, directory list box, drive list box, file list box, frame, grid, label, list box, option button, picture box, Printer object y text box. Determina el tipo de letra utilizado para desplegar texto en un control, en un dibujo o en una operación de impresión.

FontSize (Tamaño de la Fuente de Letra): Se aplica a: form, check box, combo box, command button, common dialog, data control, directory list box, drive list box, file list box, frame, grid, label, list box, option button, picture box, printer object, text box. Determina el tamaño de la letra que será usada para desplegar texto en un control, un dibujo o una operación de impresión.

Index (Índice): Se aplica al check box, combo box, command button, common dialog, data control, directory list box, drive list box, file list box, frame, grid, horizontal

y vertical scroll bar, image, label, line, list box, menu, OLE control, option button, picture box, shape, text box y timer. Especifica el número que identifica únicamente a un control en un arreglo de controles. Está disponible sólo si el control es parte de un arreglo de controles. Es de sólo lectura a la hora de ejecución.

MousePointer (Apuntador del Mouse): Se aplica a: form, check box, combo box, command button, data control, directory list box, drive list box, file list box, frame, horizontal y vertical scroll bar, image, label, list box, option button, picture box, Screen object y text box. Determina el tipo de apuntador del mouse desplegado cuando el mouse se encuentra sobre una posición particular de una forma o control a la hora de la ejecución.

MultiLine (Multilínea): Se aplica sólo a text box y especifica si un text box puede aceptar y desplegar múltiples líneas de texto. Es de sólo lectura en la ejecución.

Name (Nombre): Se aplica a: todas las formas, controles y los objetos acceso a datos Database, Dynaset, Field, Index, QueryDef, Snapshot, Table y TableDef. Especifica el nombre usado en el código para identificar una forma, control u objeto de acceso a datos. No está disponible en la ejecución.

ScrollBar (Barra de Desplazamiento): Se aplica a MDI form, grid, text box. Especifica si un objeto tiene scroll vertical u horizontal y es de sólo lectura en la ejecución.

Style (Estilo): Se aplica sólo a combo box. Especifica el tipo de combo box y la porción de su caja de lista, es de sólo lectura a la hora de la ejecución.

TabIndex (Índice de Tabulación): Se aplica a: check box, combo box, command button, directory list box, drive list box, file list box, frame, grid, horizontal y vertical scroll bar, label, list box, option button, picture box y text box. Determina el orden del tab de un control con su forma padre.

Text (Texto): Se aplica a: combo box, list box y text box. Determina el texto contenido en el área de edición. Para el combo box y list box se determina las cosas seleccionadas en la caja de lista, el valor que regresa es equivalente al valor regresado por la expresión List(listindex). Es de sólo lectura a la hora de ejecución.

Visible (Visible): Se aplica a: form, MDI form, check box, combo box, command button, common dialog, data control, directory list box, drive list box, file list box, frame, grid, horizontal scroll bar, image, label, line, list box, menu, OLE control, option button, picture box, shape, text box y vertical scroll bar. Determina si un objeto es visible o no.

8.2.5 Programación

Es posible programar cada uno de los objetos que contiene el proyecto o la aplicación. La edición del código se hace en la ventana de código (Figura 8.5). Dicha ventana de código incluye:

- 1) Object Box (Caja de Objeto): Contiene la lista de las formas y controles que contiene la forma actual.

- 2) Procedure Box (Caja de Procedimientos): Cuando se está editando el código de una forma, la Procedure Box lista todos los eventos reconocidos por Visual Basic para la forma o control desplegado en la Object box. Cuando el Object Box está en general, el Procedure Box muestra todas las funciones generales que se utilizan en el proyecto.
- 3) Split Bar (Varra de División): Aquí se encuentra debajo de la barra de título hasta arriba del scroll bar vertical. Arrastrando hacia abajo se pueden ver diferentes partes del código al mismo tiempo. La información que aparece en la Object box y la Procedure box se aplica a el código de la ventana activa.

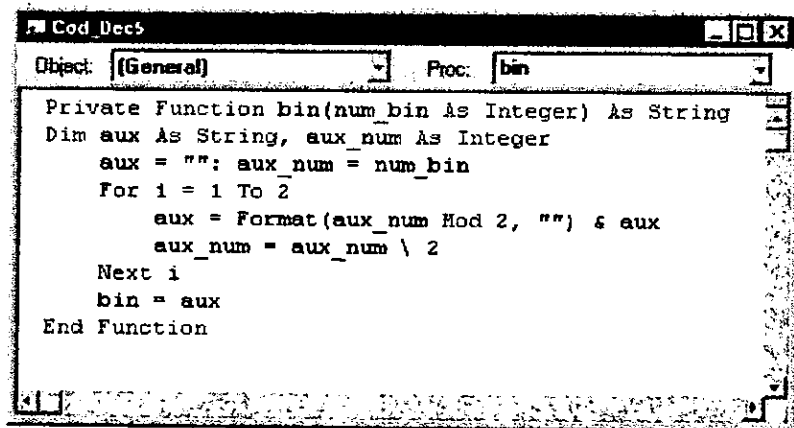


Figura 8.5: Ventana de Código

8.2.6 Menús

En Visual Basic es posible diseñar menús de control para las aplicaciones. Un menú puede contener comandos, títulos de submenús y barras separadoras. Cada menú creado puede tener cuatro niveles arriba de submenús.

Para crear una barra de menú se usa la ventana de Menu Design que despliega los nombres de menús disponibles para usar en una ventana activa. También se puede escribir código para hacer que el menú responda dinámicamente o para usar preferencias o condiciones de ejecución. Mientras se colocan algunas propiedades de control del menú en la ventana de Menu Design, todas las propiedades del menú están disponibles en la ventana de propiedades.

Cuando se crea una aplicación MDI (Interface Multiple-Documento), la barra de menú de la forma hija MDI remplazará la barra de menú de la forma MDI cuando la forma hija es activada.

8.2.6.1 Ventana Menu Design

La ventana de Menu Design es para crear menús de las aplicaciones, definiendo propiedades del menú. Esta ventana se muestra en la Figura 8.6 y consiste de las siguientes propiedades:

Caption (Título): contiene el nombre del menú o comando que aparecerá en la barra de menú.

Name (Nombre): es el nombre de control del menú. Un nombre de control es un identificador usado sólo para acceder el menú en código, éste no aparece en el menú.

Index (Índice): es el valor numérico que determinará la posición del control de menú en un arreglo de controles. Esta posición no está relacionada con la posición de la pantalla.

Shortcut (Teclas de Acceso Rápido): aquí se define una combinación de teclas (por ejemplo Alt + F1) para acceder rápidamente al control de menú requerido.

WindowList (Lista de Ventanas): esta opción se selecciona para que el menú de control contenga una lista de abrir formas MDI hijas en una aplicación MDI.

HelpContextID (Número de Identificación de Ayuda): único valor numérico para el contexto ID. Este valor se usa para encontrar la ayuda apropiada de un tópico en el identificador del archivo de ayuda por la propiedad HelpFile.

Checked (Seleccionado): esta opción se selecciona si se quiere que aparezca una marca inicialmente a la izquierda del término del menú.

Enabled (Habilitado): se selecciona si se quiere que en la ejecución responda o no (apareciendo con un color tenue) a los eventos.

Visible: se selecciona si se quiere que en la ejecución aparezca el control de menú.

Flechas Horizontales: se usan para cambiar el nivel de un menú a un nivel más alto o más bajo. Se pueden crear cuatro niveles de submenús.

Flechas Verticales: se usan para cambiar la posición del menú en el mismo nivel del menú.

List Box (Caja de Lista): es una lista jerárquica de elementos del menú. Los elementos submenú se sangran para indicar su nivel o posición jerárquica.

Botón Next (Siguiente): Mueve la selección a la siguiente línea.

Botón Insert (Insertar): Inserta una línea, antes de la línea actual. en cuadro de lista

Botón Delete (Borrar): Borra línea seleccionada.

Botón OK (Aceptar): Cierra la ventana de Menu Design y aplica todos lo cambios del menú a la última forma seleccionada. El menú está disponible a la hora del diseño, pero los eventos no se generan.

Botón Cancel (Cancelar): Cierra la ventana de Menu Design y cancela todos los cambios.

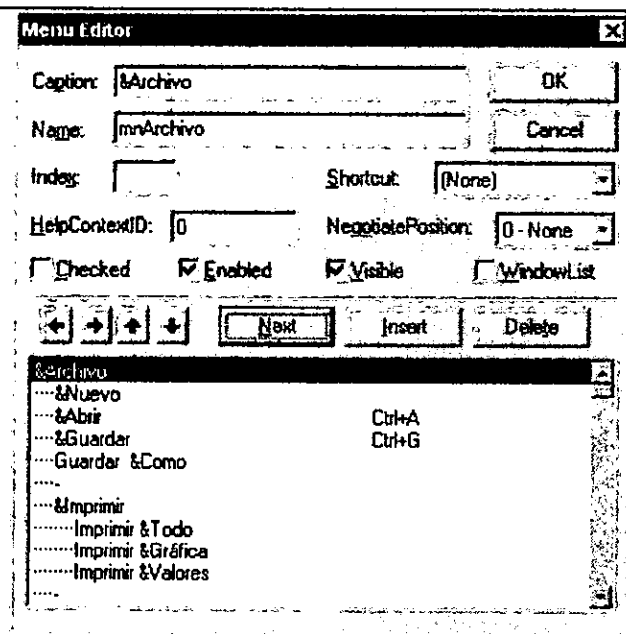


Figura 8.6: Ventana Menu Design

Capítulo 9:

Desarrollo del Módulo de Osciloscopio

9.1 Introducción

El tema principal del presente trabajo se enfoca especialmente en la transmisión y recepción de una señal eléctrica de forma inalámbrica, así como la graficación de esta señal por medio de un programa en una PC. Por eso titulamos este trabajo como “Módulo de Osciloscopio con Puntas Inalámbricas”.

El diagrama de bloques del Módulo de Osciloscopio se muestra en la Figura 9.1. En el presente capítulo se explica a detalle cómo se desarrollaron cada uno de los bloques.

Como se puede observar en el diagrama de bloques, el sistema se divide en dos partes; los bloques de la primera parte se encargan del acondicionamiento, procesamiento, codificación y transmisión de la señal eléctrica a analizar; y los de la segunda de la recepción, decodificación, procesamiento y graficación de dicha señal.

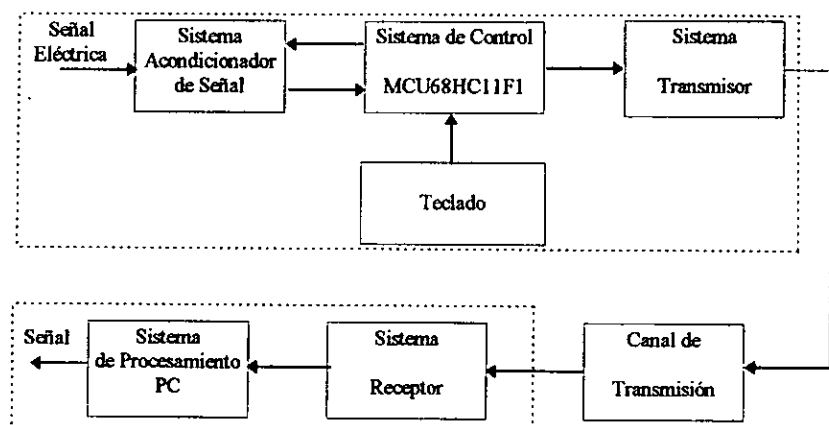


Figura 9.1: Diagrama de Bloques del Módulo de Osciloscopio

En cuanto a la primera parte del desarrollo del Módulo de Osciloscopio se diseñó y desarrolló un pequeño sistema electrónico utilizando el microcontrolador MCU68HC11F1 con el fin de controlar la captura de la señal eléctrica a analizar. En este sistema electrónico se implementó un pequeño teclado especial para controlar remotamente un programa en la PC. También se desarrolló un programa en ensamblador para poder indicar al microcontrolador en qué momento podía realizar la captura y transmisión de la señal eléctrica. Dentro de este programa se desarrolló una rutina que implementa un esquema de codificación para garantizar la integridad de los datos a transmitir.

Con respecto a la segunda parte había que desarrollar un programa en algún lenguaje de programación para poder graficar la señal eléctrica transmitida.

Dentro de este programa había que ver el esquema de detección y corrección de errores y decodificación de la señal recibida para posteriormente procesarla y graficarla.

El desarrollo del software del Módulo de Osciloscopio se hizo con los programas que se nombran abajo, y los códigos de los programas desarrollados se encuentran contenidos en el Apéndice C y en un disco anexo al trabajo.

- ♦ *IASM11 (Ensamblador Integrado) Versión 3.02m* de P&E Microcomputer Systems: Este programa es para editar y compilar programas en lenguaje ensamblador para poder programar el MCU68HC11F1 a través de su puerto serie. El programa crea un archivo con extensión “.S19”.
- ♦ *PCbug11 Versión 3.24a* de Motorola: Este programa se utiliza para cargar en la memoria RAM o EEPROM del MCU68HC11F1 el programa compilado con extensión “.S19”.
- ♦ *Visual Basic Versión 4.0 Edición Profesional*: Este más que un programa es un lenguaje de Programación y sirvió para hacer la interface gráfica que muestra la señal eléctrica capturada y las distintas ventanas de control de dicha señal.

9.2 Captura, Procesamiento y Transmisión de la Señal

La Captura, Procesamiento y Transmisión de la Señal se resume en un término: "Punta Inalámbrica". Este término se debe más que nada, a que la parte de captura de la Señal Eléctrica del Módulo de Osciloscopio envíe dicha señal muestreada de forma inalámbrica a una PC, para que en ésta se grafique. Es obvio que no se puede tener sólo las puntas de pruebas sin conectarse a algún sistema electrónico de captura y que éstas puedan transmitir, ya que se requiere de un circuito electrónico que procese la señal eléctrica y la transmita. Nosotros le denominamos "Punta Inalámbrica" a todo el módulo que abarca la captura de la señal eléctrica, la codificación y la modulación de la información para que se haga la transmisión inalámbrica.

En el desarrollo del módulo de punta inalámbrica se requiere del MCU68HC11F1 para poder controlar el momento de captura, la conversión de la señal analógica a digital, la codificación de la señal convertida a digital, la salida de dicha información procesada para que pueda ser transmitida de forma inalámbrica y la generación de una señal portadora que haga posible la transmisión inalámbrica. Además del MCU68HC11F1 se requiere de un sistema adecuador de la Señal Eléctrica que permita manejar señales dentro de un rango de voltajes máximos y mínimos establecidos con el MCU68HC11F1, al mismo tiempo debe servir como protección para el microprocesador. En cuanto al sistema de transmisión inalámbrica se eligieron ILEDs (Diodos Emisores de Luz Infrarroja) para transmitir la señal.

9.2.1 Especificaciones de la Información a Transmitir

Primeramente había que definir cómo se iba a manejar la información capturada. El Convertidor A/D del MCU68HC11F1 pone la señal capturada en un registro de 8 bits (1 byte), pero sólo es una muestra del voltaje que se está convirtiendo. La graficación de varias muestras se dibujaría sin poder determinar la escala de voltaje a la que fueron capturadas. Esto nos llevó a determinar que se utilizaría por lo menos otro byte, donde debería estar la información de la escala de voltaje, el signo de la muestra de voltaje, y la escala del periodo de captura. Para hacer esta consideración también se debía tomar en cuenta que estos bytes no se iban a transmitir directamente, sino que iban a tener que ser codificados, por lo que el número de bytes transmitidos se iba a incrementar.

Siguiendo el esquema de codificación BCH, el cual se menciona más adelante, nos indica que para codificar una palabra de 2 bytes (16 bits) se requería de al menos 8 bytes (de los cuales sólo se ocupan 63 bits de los 64) para garantizar una detección y corrección de un máximo de 11 bits erróneos por palabra. Si se quería disminuir la cantidad de bytes codificados garantizando una detección y corrección de un máximo de 5 bits erróneos por palabra, sólo se debían utilizar 11 bits de 16 (es decir de 2 bytes) para producir una palabra de 4 bytes codificados (de los cuales sólo se ocupan 31 bits de los 32). Esta segunda opción no nos daba mucho espacio para poder especificar la escala de voltaje, escala de periodo y signo de la señal muestreada. Es por eso que se optó por utilizar el Esquema de Detección y Corrección de Error donde se iba a producir una palabra de 8 bytes codificados, con lo que se podrían manejar

valores más grandes para representar la escala de voltaje y escala de periodo a la cual se muestréa la señal eléctrica.

Además de contener las muestras de voltaje con sus escalamientos, la palabra de 2 bytes sin codificar también puede contener la información con la que se controla el programa graficador del "Módulo de Osciloscopio". El sistema tiene que hacer remotamente el cambio de pantallas, las selección de opciones del programa y el momento de captura. Para hacer esto se requerían de sólo 3 bits de la palabra sin codificar, los cuales se controlarían por medio de un teclado especial.

Ya analizado que se requerían 2 bytes se procedió a especificar la manera de cómo se utilizarían en el MCU68HC11F1. Considerando que la RAM del MCU68HC11F1 empieza en la dirección \$0000 y que a partir de esta se almacenan los accesos internos de la pila del microprocesador, se decidió que los bytes de RAM a utilizar estarían a partir de la dirección \$0100. Bajo este análisis se especificó que los dos bytes requeridos iban a estar entonces en las direcciones \$0100 y \$0101. Cabe aclarar que las direcciones que se mencionen estarán referidas como direcciones hexadecimales y cuando se refieran a ellas siempre se antepone el signo \$.

Los 2 bytes de la palabra sin codificar se denominaron como byte de dato (al byte \$0100) y byte de información (al byte \$0101). El byte \$0100 puede tomar dos formatos, dependiendo de si se está transmitiendo un dato de alguna señal eléctrica muestreada o algún dato de combinación del teclado especial. A continuación se describe la función de cada bit de los bytes de la palabra sin codificar.

9.2.1.1 Byte de Dato (\$0100)

	7	6	5	4	3	2	1	0	
\$0100	D7	D6	D5	D4	D3	D2/TFu	D1/TAr	D0/TAb	DATO
RESET	0	0	0	0	0	0	0	0	

D7-D0 (Bits de Datos): Cuando el bit D/F del byte de Información (\$0101) es 1, estos bits llevan la información de la muestra de la Conversión A/D del Voltaje de la señal eléctrica que se acaba de capturar.

TFu (Tecla Función): Cuando el bit D/F del byte de Información (\$0101) es 0 este bit representa el estado del botón Función del teclado remoto especial.

TAr (Tecla Arriba): Cuando el bit D/F del byte de Información (\$0101) es 0 este bit representa el estado del botón Tecla_Arriba del teclado remoto especial.

TAAb (Tecla Abajo): Cuando el bit D/F del byte de Información (\$0101) es 0 este bit representa el estado del botón Tecla_Abajo del teclado remoto especial.

Con los bits TFu, TAr y TAb se hacen 7 posibles combinaciones con las que se controla el programa "Módulo de Osciloscopio".

9.2.1.2 Byte de Información (\$0101)

	7	6	5	4	3	2	1	0	
\$0101	D/F	S	V2	V1	V0	P1	P0	T	INFO
RESET	0	0	0	0	0	0	0	0	

D/F (Dato/Función): Cuando el bit está en 0 indica que se está transmitiendo una Muestra de la conversión A/D de la Señal Eléctrica recientemente capturada. Cuando este bit está en 1 indica que se está transmitiendo alguna combinación de teclas del teclado especial para controlar el programa "Módulo de Osciloscopio".

S (Signo): Este bit indica el Signo (con referencia en 0 V) de la Muestra de la conversión A/D del voltaje de la señal eléctrica recientemente capturada y que se va a transmitir. Cuando este bit es 1, indica que se envió una muestra de la señal con voltaje positivo o cero Volts, y cuando este bit vale 0, indica que se va a transmitir una muestra de la señal con voltaje negativo.

V2-V0 (Escala de Voltaje): Estos 3 bits indican el escalamiento de voltaje a la que viene la muestra de la conversión A/D de la señal eléctrica capturada..

P1-P0 (Escala de Periodo): Estos 2 bits indican el escalamiento de Periodo a la que viene la muestra de la conversión A/D de la señal eléctrica capturada.

T (Palabra Transmitida): Este bit siempre está en 0 a menos que se quiera transmitir información remotamente. Antes de transmitir información este bit cambia a 1, y una vez que ya se ha enviado la información este bit regresa a 0.

9.2.1.3 Estado del MCU68HC11F1 mientras Recibe una Tarea

Mientras el MCU68HC11F1 recibe alguna instrucción que permita la captura y procesamiento de alguna señal eléctrica, éste se encuentra en un ciclo preguntando siempre por el estado del bit 7 del Puerto G (\$1002), el cual es el indicador de requerimiento de conversión A/D de una señal eléctrica. Cabe especificar aquí que cuando se desea hacer una captura, se utiliza el teclado especial que se analizará en la siguiente sección. Este teclado especial funciona por interrupción, cuando se quiere hacer una captura de alguna señal eléctrica, es en el procedimiento de

esta interrupción donde se cambia el estado del bit 7 del Puerto G. En la Figura 9.2 se muestra el diagrama de flujo de lo que hace el MCU68HC11F1 mientras espera alguna instrucción y lo que hace cuando se detecta que se quiere capturar una señal eléctrica.

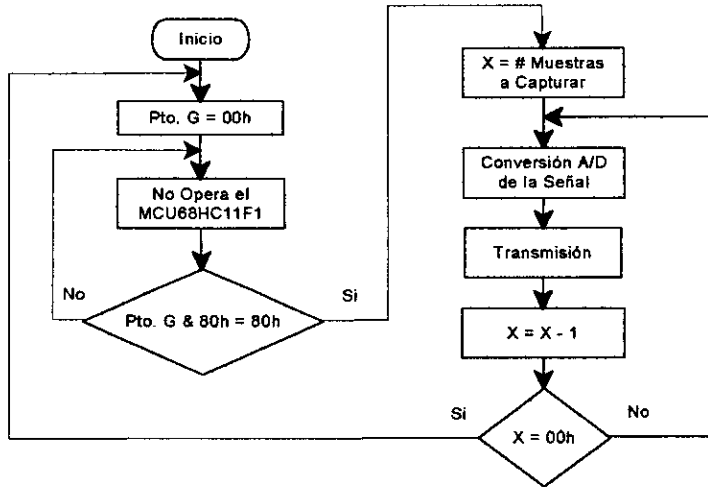


Figura 9.2: Diagrama de Flujo Mientras Recibe Alguna Orden del Teclado

Cuando el MCU68HC11F1 detecta que se quiere hacer una captura de señal eléctrica se carga en el Registro X el número de muestras que se quieren hacer de la señal eléctrica. Aquí no se especifican precisamente el número de muestras que se quieren realizar, pero entre mayor sea el número de muestras, mejor información de la señal se tendrá pero se tardará más tiempo en graficarse la señal en el programa “Módulo de Osciloscopio”, ya que el proceso de decodificación (detección y corrección de errores) es muy largo y cuando se hace un procesamiento de un gran número de muestras, la graficación de la señal eléctrica es muy lenta.

Una vez realizada la captura de las n muestras se le asigna un 0 al bit 7 del Puerto G. A continuación se muestra el código en lenguaje ensamblador que utiliza el MCU68HC11F1 para hacer esta tarea.

```
PR1:  NOP           ; Cuando MCU38HC11F1 está en este ciclo, verifica
      LDAA $1002    ; el bit 7 del Pto. G (bit que indica
      ANDA #$80     ; momento de conversión A/D de señal)
      CMPA #$80     ; Mientras verifica hace que
      BNE PR1       ; el MCU68HC11F1 no haga algo.
      LDX #$0014    ; Carga el valor de 0014h, correspondientes a las
                   ; 20 muestras.
PR2:  JSR CONAD     ; Brinca a la rutina de Conversión A/D de señal.
      JSR TRANS     ; Brinca a la rutina de Transmisión los valores
                   ; convertidos.
      DEX           ; Decrementa el Reg. X, (muestras a transmitir).
      BNE PR2       ; Si no es cero el decremento (Si aún hay muestras
                   ; a transmitir), brinca a la etiqueta PR2.
      LDAA #$00     ; Carga 00h en Reg. A para ponerlo en el bit 7 del
      STAA $1002    ; Pto. G (bit del momento de conv. A/D de señal).
      BRA PR1       ; Brinca a PR1.
```

9.2.2 Teclado de la Punta Inalámbrica

En este tema se define de qué forma se va a controlar la captura de la señal eléctrica a analizar y cómo se va a controlar remotamente el programa "Módulo de Osciloscopio". Para poder tener este control se requiere del desarrollo de un pequeño teclado especial, que consiste de 3 teclas. Con la combinación de las 3 teclas se obtiene 7 diferentes funciones, sin considerar que debe haber una combinación con la cual no se realizará nada, ya que es la que va a estar presente cuando ninguna tecla es presionada.

El estado del teclado va a ser sondeado por el MCU68HC11F1 a través de su pin externo de interrupción $\overline{\text{IRQ}}$ y del Puerto G configurado de tal forma que se pueda leer dicho estado. Como el pin externo de la interrupción $\overline{\text{IRQ}}$ es

verificado bajo, es decir se activa con una señal de voltaje cercana a los 0V, se hace un arreglo de tal forma que cuando alguna de las tres teclas se presiona, sola o en combinación con las otras teclas se mande una señal lógica de 0 al pin $\overline{\text{IRQ}}$. Es por eso que se utiliza una compuerta lógica AND, de tal forma que por medio de unos "Push-Buttons" siempre reciba una señal lógica de 1. El arreglo anteriormente mencionado se muestra en la Figura 9.3.

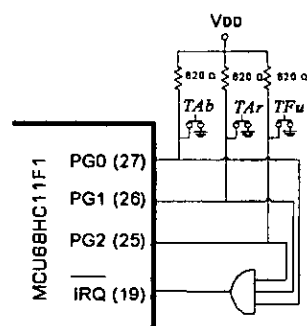


Figura 9.3: Diagrama Eléctrico del Teclado Remoto

Nótese que aquí se ocupan 3 pines del Puerto G del MCU68HC11F1 como entradas, por lo que su configuración se hace por medio de las siguientes líneas de ensamblador, donde para configurar el Puerto G se utiliza el registro \$1003. Poniendo un 1 lógico en un bit de ese registro, se indica que el correspondiente pin del Puerto G (\$1002) será un pin de salida, pero si se escribe un 0 lógico en ese mismo bit del registro \$1003 se indica que el correspondiente pin del Puerto G (\$1002) será un pin de entrada. Entonces, ya que se ocuparán los bits 0, 1 y 2 del Puerto G se configura este puerto para que dichos pines sean de lectura.

```
LDA #80h ; Se carga 80h, para configurar el Pto. G, donde
STAA $1003 ; sólo el bit 7 será de escr. y el resto de lect.
```

Además de la configuración del Puerto A al comienzo del programa se debe especificar a qué rutina brincar cuando se genera una interrupción por medio del $\overline{\text{IRQ}}$. Estas especificaciones se definen de la siguiente manera, notando que al final se debe limpiar la bandera de interrupción.

```
LDAA #$7E ; Carga 7Eh (correspondiente a la instrucción JMP)
STAA $EE ; en el 1er. byte del Vector IRQ (Vector de
          ; Interrupción).
LDX #INTEC ; Carga el valor de la Dirección donde comienza la
STX $EF ; rutina INTEC en sig. 2 bytes del Vector IRQ.
CLI ; Limpia la Máscara de Interrupción.
```

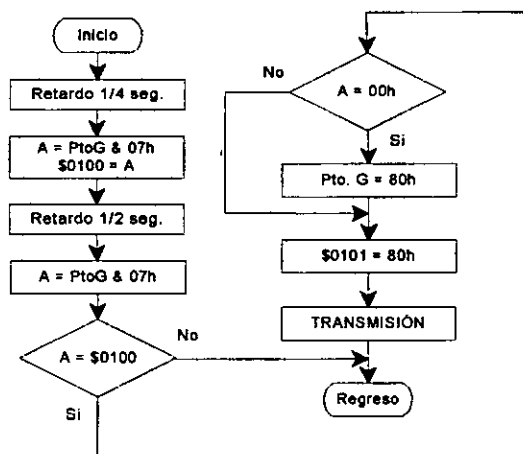


Figura 9.4: Diagrama de Flujo de la Interrupción por Teclado

El diagrama de Flujo de la Figura 9.4, muestra lo que se hace al generar una interrupción cuando se recibe una señal lógica de 0 en el pin $\overline{\text{IRQ}}$.

Al comienzo de la interrupción se hace un pequeño retardo de $\frac{1}{4}$ de segundo (250 ms.) para confirmar que la rutina de interrupción no se activó por un ruido eléctrico generado en el teclado. Después de este retardo se hace una lectura del Puerto G poniendo en 0 lógico los bits del 3 al 7, que son los que no tiene nada que ver con el teclado especial. La lectura de los tres bits del teclado especial se almacena en el Registro A. Después se hace una espera de $\frac{1}{2}$ segundos para volver a leer el Puerto G y compararlo con la lectura anterior. Con este último paso se garantiza que realmente se está presionando la tecla o teclas deseadas, esto es más que nada por si se presiona primero una tecla y después otra, siendo que en realidad se quieren presionar ambas haciendo una combinación distinta a sólo presionar una de ellas. A continuación se muestra el código para hacer la confirmación de la tecla presionada.

```
INTEC: PSHA      ; Se meten los valores del Registro A en la Pila.
      JSR RETAR   ; Retardo de 1/4 de Seg. (Brinca a RETAR).
      LDAA $1002  ; Carga en Reg. A, el valor del Pto. G, haciendo
      ANDA #$07   ; que sólo tome los bits 0, 1 y 2, por medio de
      STAA $0100  ; la AND, para posteriormente depositarlo en la
                  ; Dir. $0100 de RAM (Reg. de Datos).
      JSR RETAR   ; Hace un retardo de
      JSR RETAR   ; 1/2 seg.
      LDAA $1002  ; Carga (otra vez después de 1/2 seg.) en Reg. A,
      ANDA #$07   ; el valor del Pto. G haciendo que sólo tome los
      CMPA $0100  ; bits 0,1 y 2 (con una AND) y compararlo con el
                  ; valor de la Dir. $0100 de RAM (Reg. de Datos).
      BNE IT2     ; Si son diferentes ambos valores brinca a IT2.
```

La siguiente parte del diagrama es para identificar si se requiere hacer una conversión A/D de la señal eléctrica. Esta indicación se hace presionando los tres botones del teclado al mismo tiempo. Hay que recordar que el presionar un botón genera una señal lógica de 0, por lo que al presionar las tres teclas al mismo tiempo los bits 0, 1 y 2 del Puerto G registran un 0 lógico y al hacerle una AND de estos tres bits con 111 se obtiene que el resultado es 000. Si el resultado de esa AND en el registro A

Módulo de Osciloscopio

es 00h, asigna 1 al pin 7 del Puerto G, indicando así que se va a hacer una captura de la señal eléctrica. Si el resultado de la AND en el registro A es diferente de 00h, el pin 7 del Puerto G mantiene un 0. Una vez identificado si se presionaron las tres teclas al mismo momento para hacer una captura de señal eléctrica se pone en 1 el bit D/F del registro \$0101 (Byte de Información) para indicar que se presionó una de las 7 (de 8 posibles) combinaciones de las teclas. Una vez hecha la identificación de la combinación presionada se ejecuta el procedimiento de Transmisión, que es el que se encarga de llamar a su vez el procedimiento de codificación de los bytes \$0100 y \$0101 para después transmitir por el puerto serie del MCU68HC11F1 la información codificada. A continuación se muestra el código en Ensamblador para identificar la combinación de teclas presionadas y la llamada al procedimiento de transmisión que se encuentra en el procedimiento que se activa cuando se genera una interrupción por medio del pin de $\overline{\text{IRQ}}$.

```
CMPS #000 ; Se compara Reg. A con 00h (por si se presionaron
           ; a la vez, las 3 teclas del teclado remoto).
BNE IT1   ; Si son diferentes los valores (NO se presionaron
           ; las 3 teclas del teclado remoto), brinca a IT1.
LDAA #80  ; Carga 80h en el Reg. A, para dejarlo en el Pto. G
STAA $1002 ; e indicar que se inicie la conversión A/D.
IT1: LDAA #80 ; Carga 80h en el Reg. A, dejándolo en Dir. $0101
     STAA $0101 ; de RAM, (Byte de Información) para indicar que
           ; se presionó alguna tecla.
JSR TRANS ; Transmite el Dato que est. el Puerto Serie del
           ; MCU68HC11F1 (Brinca a TRANS).
IT2: PULA ; Saca el valor del Registro A de la Pila.
     RTI ; Retorno de Interrupción.
```

Las otras tareas que se realizan al presionar las otras combinaciones de teclas se definirán posteriormente, cuando se haga la explicación del procedimiento de Recepción del programa "Módulo de Osciloscopio".

9.2.3 Sistema de Acondicionamiento de la Señal

Una parte importante en el desarrollo de este proyecto es sin duda la obtención y acondicionamiento de la señal eléctrica ya que es parte fundamental en la adquisición de datos para ser obtenidos por el convertidor A/D del MCU68HC11F1.

El sistema de acondicionamiento de la señal eléctrica hace un autoescalamiento en la atenuación o amplificación por dos motivos: el primero es para proporcionar al convertidor A/D del MCU68HC11F1 voltajes pequeños, protegiéndolo de altos voltajes que lo dañen; y el segundo es para proporcionar voltajes comprendidos en unos valores ya establecidos para que el convertidor A/D pueda hacer conversiones dentro su rango.

Para la etapa de acondicionamiento de la señal eléctrica se necesitó de una red de atenuación/amplificación que acondicione la señal antes de entrar al convertidor A/D. Aunque para una mejor funcionalidad se propuso que el escalamiento en la red atenuadora se realizara automáticamente, es decir que se tuviera un circuito de auto-rango, por lo cual se pensó en implementar esta red por medio de comparadores realimentados por un convertidor A/D externo. Sin embargo por otro lado se tenía el convertidor A/D ya incluido en el MCU68HC11F1. Esto nos condujo finalmente a la elección del convertidor A/D del MCU68HC11F1, para tener menos componentes electrónicos y aprovechar los recursos que nos proporciona el mismo.

En cuanto a la realización del circuito de auto-rango, se utilizaron comparadores implementados con amplificadores operacionales y multiplexores analógicos, así como un contador y lógica combinacional adicional, para manejar el

escalamiento de la señal en la etapa de atenuación/amplificación, y de esta forma lograr enviar al convertidor A/D de MCU68HC11F1 valores de voltaje dentro del rango que maneja.

La estructura del circuito se puede resumir mediante un diagrama de bloques (Figura 9.5), donde se menciona por etapas el desarrollo del circuito de obtención y acondicionamiento de la señal.

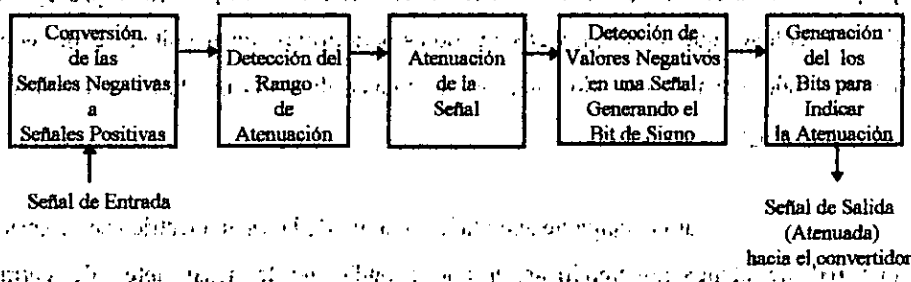


Figura 9.5: Diagrama de Bloques del Sistema de Acondicionamiento de la Señal Eléctrica

9.2.3.1 Detección de Valores Negativos y Generación del Bit de Signo

Para la etapa de detección de valores negativos y generación del bit de signo se introduce una señal eléctrica a la entrada al circuito comparador, el cual entrega a la salida un voltaje de $+V_{CC}$ (5 V.) que se interpreta como el bit de signo, el cual indica al MCU68HC11F1 si se trata de un valor de voltaje negativo o positivo de la señal de entrada. Esto se realiza mediante la implementación de un circuito comparador a través de un amplificador operacional (TL074), en su configuración no inversora.

En la Figura 9.6 se muestra el diagrama para la configuración del amplificador operacional (comparador no inversor).

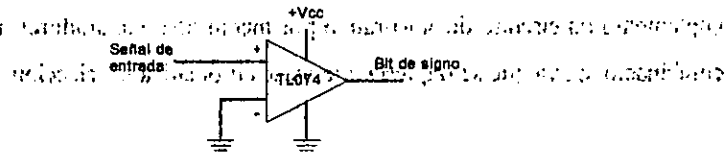


Figura 9.6: Comparador No-Inversor

Para la etapa de conversión de voltajes negativos a positivos, se necesita de un circuito auxiliar, ya que el convertidor A/D del MCU68HC11F1 sólo puede hacer una conversión A/D con voltajes positivos. Esto se logra conjuntamente con el bit de signo generado en la etapa mencionada anteriormente.

Por lo tanto se pensó emplear un circuito rectificador de onda completa, de tal manera que una señal AC o con valores negativos pueda ser rectificada y de esta forma obtener solo voltajes positivos. Aunque con este circuito se puede rectificar una señal eléctrica, también se tiene una atenuación inevitable para señales pequeñas o de poco voltaje, además de interferencias o problemas de ruido resultando no muy funcional. Por esta razón se implementó un circuito rectificador de onda completa de precisión, ya que en muchas de las ocasiones es necesario utilizar un Osciloscopio para analizar señales pequeñas.

9.2.3.2 Detección del Rango de Atenuación/Amplificación

En la parte de detección del rango de atenuación/amplificación se implementó un circuito de auto-rango por medio de comparadores, un contador y lógica combinacional, ya que se requería que fuese automática la selección.

En la implementación de los comparadores se utilizaron en primera instancia amplificadores operacionales JFET, para obtener precisión y velocidad en respuesta (Slew-Rate); tomando para esto la configuración básica de un comparador, como se muestra en la Figura 9.7.

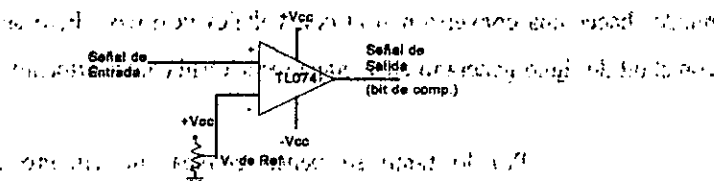


Figura 9.7: Configuración Básica de un Comparador

A la salida del amplificador se tiene un voltaje de $+V_{cc}$ (+5 V), que es el voltaje de polarización del amplificador operacional generando el bit de estado de la señal adquirida. Para obtener el bit de estado, se requiere previamente hacer un análisis de la señal con 2 comparadores, para que den la posición de la señal muestreada con un límite superior positivo (+1 V), un límite inferior positivo (+0.1 V), tomando el valor de DC de la señal rectificada, tal como se puede observar en la Figura 9.8.

Los comparadores del sistema acondicionador, se diseñaron para un voltaje de comparación de entre 0.1 y 1.0 V, ya que el convertidor A/D maneja un rango de voltaje de 0 a 5 V.

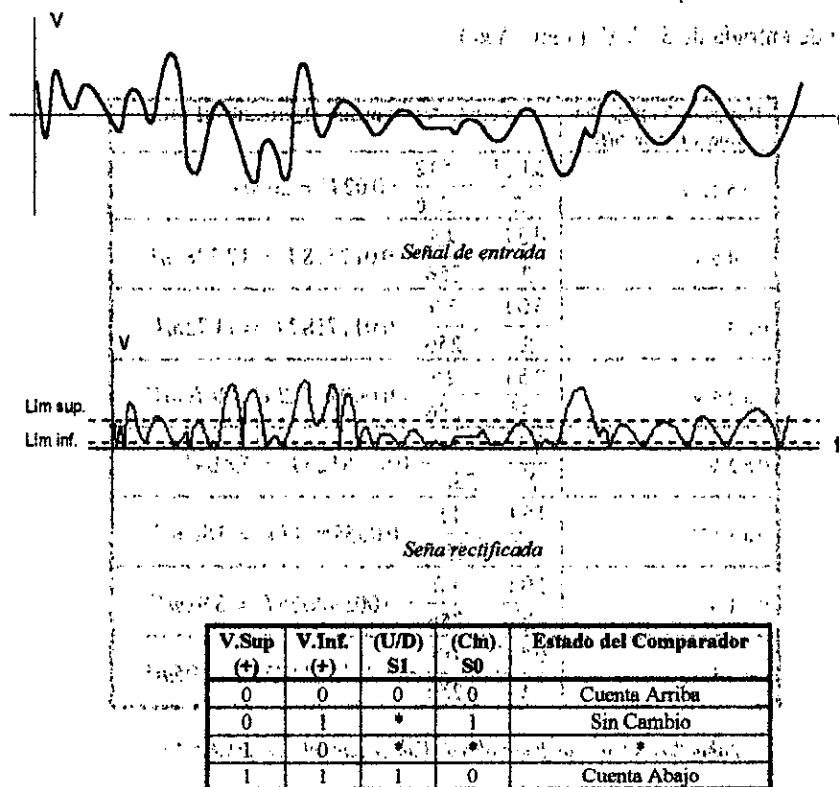


Figura 9.8: Análisis de la Señal Muestreada

En la Tabla 9.1 se encuentra el resultado de un análisis sobre los

diferentes intervalos de voltaje que se pueden manejar con el convertidor A/D del

MCU68HC11F1 y la resolución que se puede obtener en cada intervalo de voltaje.

Tomando en cuenta que el convertidor A/D tiene una resolución de 8 bits, y un voltaje máximo de entrada de 5.12 V. ($V_{RH} - V_{RL}$).

Rango de Voltaje del Convertidor A/D	Voltaje del bit menos significativo (LSB)
0 a 5.12 V	$\frac{5.12 V}{2^8} = \frac{5.12}{256} = 0.02 V = 20 mV$
0 a 4.5 V	$\frac{4.5 V}{2^8} = \frac{4.5}{256} = 0.017578 V = 17.578 mV$
0 a 3 V	$\frac{3.0 V}{2^8} = \frac{3.0}{256} = 0.0117187 V = 11.72 mV$
0 a 2.5 V	$\frac{2.5 V}{2^8} = \frac{2.5}{256} = 0.00976562 V = 9.76 mV$
0 a 2 V	$\frac{2 V}{2^8} = \frac{2}{256} = 0.0078125 V = 7.81 mV$
0 a 1.5 V	$\frac{1.5 V}{2^8} = \frac{1.5}{256} = 0.00585937 V = 5.86 mV$
0 a 1 V	$\frac{1.0 V}{2^8} = \frac{1.0}{256} = 0.00390625 V = 3.91 mV$
0 a 0.5 V	$\frac{0.5 V}{2^8} = \frac{0.5}{256} = 0.001953125 V = 1.95 mV$

Tabla 9.1: Análisis de Intervalos de Voltaje para el Convertidor A/D.

De las consideraciones anteriores tomando en cuenta una ventana de comparación, se estudiaron los posibles estados en que se podían encontrar los comparadores, definiendo el tipo de atenuación o amplificación que se manejaría dentro del sistema completo.

Para la etapa de detección del rango de atenuación o amplificación, primeramente se tuvo que hacer un análisis de cuantos rangos se podían manejar, así como los valores de atenuación o amplificación que conforman dicho rango. Esto queda

definido de acuerdo al convertidor A/D del MCU68HC11F1, con la resolución que nos brinda de 8 bits. Teniendo un voltaje mínimo de 3.91 mV, para activar el bit menos significativo (LSB), se toma una escala de voltaje de entrada del convertidor de 0 a 1 V, de tal forma que se pueda tener un voltaje adecuado y aceptable en la red de atenuación o amplificación.

Considerando el rango de voltaje de 0 a 1 V de la Tabla 9.1, se tiene un voltaje de 3.91 mV para el bit menos significativo, por lo que el convertidor efectuará la conversión analógica a digital como se muestra en la Tabla 9.2.

Voltaje del Convertidor A/D	Salida digital del Convertidor A/D
3.91 mV	00000001
7.82 mV	00000010
11.73 mV	00000011
15.64 mV	00000100
19.55 mV	00000101
M	M
1.0 V	11111111

Tabla 9.2: Voltajes de Conversión A/D

En el diseño de la punta inalámbrica se consideró manejar tres bits para "escalar" el voltaje que entra al convertidor A/D del MCU68HC11F1. Con esos tres bits se pueden tener ocho posibles estados, donde cuatro de ellos serían de atenuación y cuatro de amplificación, quedando establecidos como se muestra en la Tabla 9.3. En esta tabla tenemos una escala máxima de atenuación de 0.001 (1/1000), y una máxima amplificación de 10000, de tal forma que es posible manejar voltajes de 0.4 μ V (teóricamente), hasta voltajes de 1000 V en la señal de entrada. Dicho escalamiento da un gran margen de voltaje para el manejo de señales.

A	B	C	Factor de Atenuación
0	0	0	x 0.001 (1/1000)
0	0	1	x 0.01 (1/100)
0	1	0	x 0.1 (1/10)
0	1	1	x 1 (1/1)
1	0	0	x 10
1	0	1	x 100
1	1	0	x 1000
1	1	1	x 10000

Tabla 9.3: Factores de Atenuación/Amplificación

El circuito de auto-rango se implementó con un contador binario de cuatro bits up/down, diseñando un circuito binario para ocho estados que cambiara de acuerdo a las salidas de los comparadores. Los estados del contador se definieron en la Tabla 9.3, y el diagrama lógico se muestra en la Figura 9.9, en el que A, B y C son las entradas de control para los switches analógicos, tales que determinarán el factor de escalamiento de la señal de entrada.

La señal de reloj para el contador, se genera con el Comparador de Salida 3 del MCU68HC11F1, para proporcionar una señal cuadrada de 20 KHz. El código en lenguaje ensamblador para generar la señal cuadrada, se menciona en un tema mas adelante, cuando se genera una señal cuadrada a 40 KHz que se emplea en la modulación de la señal a transmitir.

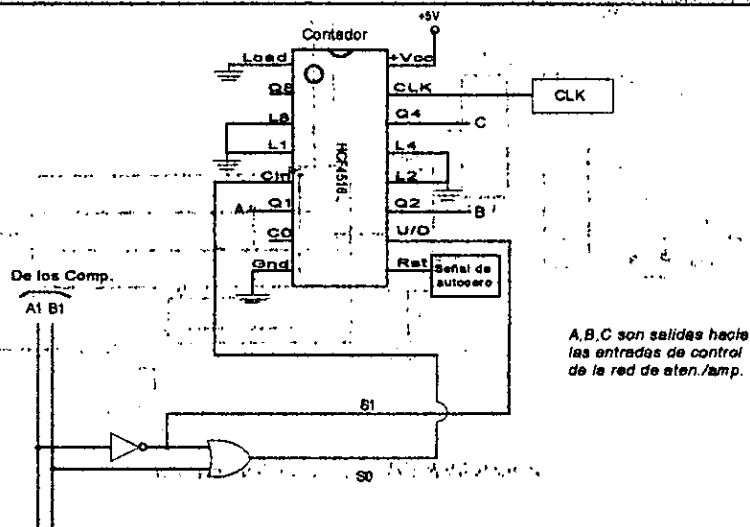


Figura 9.9 : Circuito Lógico para el Auto-Rango

9.2.3.3 Atenuación/Amplificación de la Señal

Para el circuito de atenuación/amplificación se hizo un análisis previo tomando en cuenta ocho escalas. Dicho análisis se realizó utilizando un amplificador operacional en su configuración de amplificador inversor, variando la ganancia del mismo mediante un arreglo de resistencias en décadas, es decir variando la atenuación o amplificación en un factor de 10 como se muestra en el diagrama de la Figura 9.10. En este diagrama se observa que en el arreglo del amplificador hay diodos tipo 1N914, que se utilizan como protección del circuito, ya que se manejan switches analógicos, los cuales cuando hay un cambio de un estado a otro, se tiene en la entrada un voltaje grande para señales grandes.

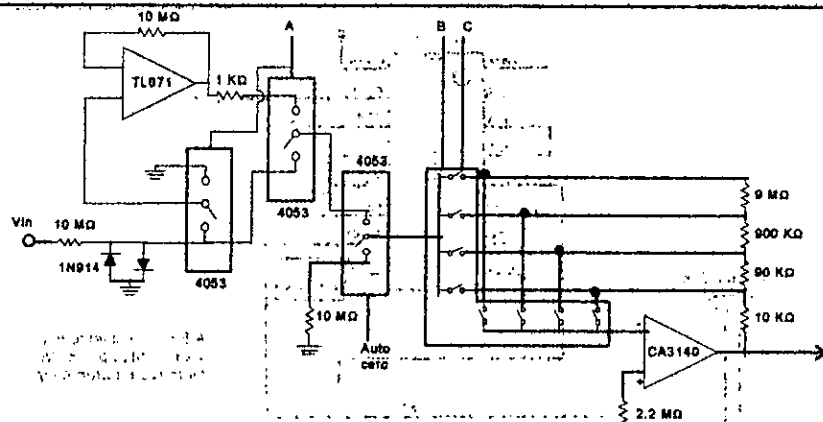


Figura 9.10: Red de Atenuación/Amplificación

En el circuito de auto-rango, fue necesario la utilización de switches analógicos que pudieran ser controlados digitalmente, por lo que en este caso se utilizó para la red atenuadora/amplificadora el C.I. 4052., el cual se conforma de un conjunto de cuatro switches analógicos y dos entradas de control, utilizados para el manejo de cuatro rangos de atenuación/amplificación con una impedancia total de $10\text{ M}\Omega$. Cuando funciona para atenuación, a la entrada de la red tenemos una resistencia de $10\text{ M}\Omega$; y cuando funciona para amplificación únicamente se cambia la resistencia de entrada del amplificador operacional a $1\text{ K}\Omega$ pero conservando la impedancia de entrada de $10\text{ M}\Omega$, esto se logra con un switch analógico independiente (C.I. 4053). De este último C.I. se obtiene una tercera línea de control, conformándose así las líneas de control ABC de la red amplificadora/atenuadora.

Las 3 líneas de selección son manejadas por medio del contador HCF4516, el cual es un contador binario de 4 bits up-down. Este contador a diferencia

del contador con dos relojes (74HC193), proporciona una mayor estabilidad en el proceso de selección del estado del contador. La habilitación del contador, así como el conteo del mismo, dependen de las ventanas de comparación (tal como se definió anteriormente en la detección de valores negativos) como se observa en la Figura 9.8:

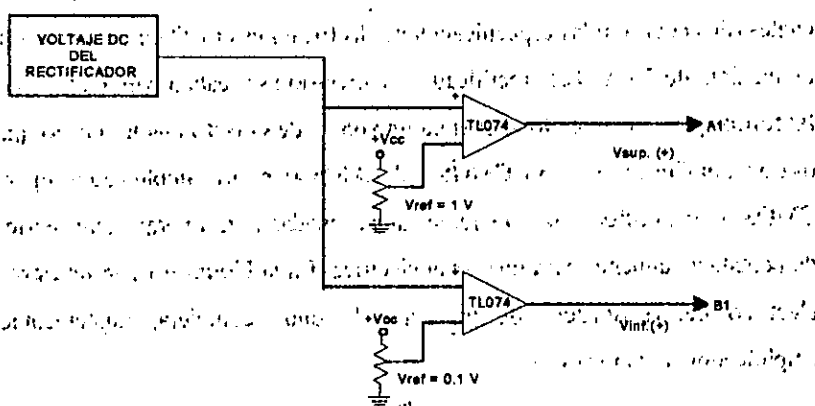


Figura 9.11: Comparadores para el Manejo de la Red de Atenuación/Amplificación

Para la atenuación automática de la señal se utilizan los circuitos comparadores retroalimentados por la red de atenuación/amplificación y la lógica binaria adicional, la cual se basa en los estados de los comparadores, como se muestra en la Figura 9.11. Para esto, el circuito básico es el contador como se mostró anteriormente en la Figura 9.9, en el cual se tienen 2 líneas de selección para que el contador cuente hacia arriba, hacia abajo o se quede sin cambio, es decir tendremos tres estados en la selección de auto-rango, como se mostró en la Tabla 9.3:

9.2.3.4 Conversión de las Señales Negativas a Señales Positivas

Debido a la necesidad de tener siempre valores positivos en la señal eléctrica adquirida, se implementó un rectificador de onda completa de precisión por medio de amplificadores operacionales de baja potencia y bajo voltaje, para cumplir con los requerimientos que demanda el convertidor A/D del MCU68HC11F1, uno de los cuales (de acuerdo a las especificaciones eléctricas) es el voltaje de entrada positivo y de un máximo de 5.1 V. Este rectificador de precisión se realizó con diodos tipo 1N914, de tal forma que en el circuito se tuviera un voltaje de Offset mínimo, por lo que también se usaron amplificadores tipo CMOS. El LMC660 es un amplificador operacional tipo CMOS, el cual ofrece las características adecuadas para operar como parte del sistema de acondicionamiento para una señal eléctrica. En la Figura 9.12 se muestra el diagrama eléctrico del rectificador de precisión de onda completa, implementado con los amplificadores operacionales.

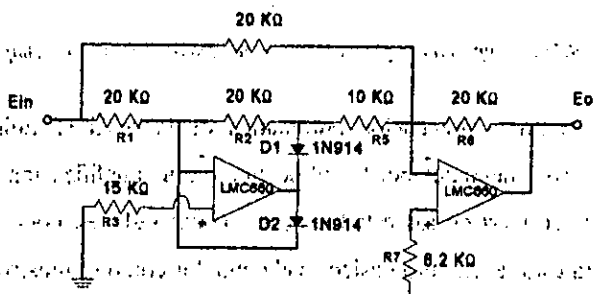


Figura 9.12: Rectificador de Onda Completa de Precisión

Un diseño previo del rectificador de precisión de onda completa utilizaba amplificadores operacionales comunes (que no eran tipo CMOS), pero causaban un voltaje de Offset más grande. También se hizo un circuito rectificador de

media onda de precisión, del cual se hicieron pruebas de funcionamiento, observándose que para voltajes muy pequeños (del orden de mV) se producían corrientes parásitas, y por lo tanto cambios en la señal de salida.

El circuito rectificador de onda completa de precisión utiliza C.I. de tecnología CMOS, porque brindan algunas ventajas cuando se están manejando señales de voltajes pequeños, además de que se pueden polarizar con $\pm 5\text{ V}$, es decir, se adaptan fácilmente a los circuitos digitales. También proporcionan a la entrada una impedancia muy alta, una ganancia grande a lazo abierto (alrededor de unos cientos de miles, como la de un amplificador operacional LM741), un voltaje de Offset del orden de mV, una rápida respuesta (slew-rate), la habilidad para manejar rápidamente altas frecuencias (ancho de banda grande) y cambios de amplitudes muy grandes, otorgando en general una buena eficiencia cuando se están manejando pequeños voltajes.

9.2.3.5 Generación de los Bits para Indicar la Atenuación

Los bits del rango de atenuación/amplificación se obtienen directamente del contador (controlado con la lógica binaria mencionada anteriormente), la cual es manejada por los comparadores que detectan el nivel de voltaje de entrada de retroalimentación, proveniente de la red atenuadora/amplificadora. El MCU68HC11F1 lee directamente estos bits y captura la señal eléctrica ya acondicionada, la cual tiene niveles de voltaje de entre 0 y 1 V (como ya se mencionó este rango de voltaje es para tener 3.91 mV en para el LSB).

9.2.3.6 Otras Consideraciones del Circuito de Acondicionamiento

Para la entrada de la señal eléctrica (en el caso de amplificación), fue necesario implementar un circuito amplificador de ganancia unitaria mediante un amplificador operacional para poder tener a la entrada una impedancia fija de $10M\Omega$. También para la entrada de voltaje a los comparadores fue necesario implementar un diodo y un capacitor que nos entregara el valor de DC (pico) de la señal rectificada, evitando de esta forma la inestabilidad en el sistema.

Cabe mencionar que se utilizaron circuitos integrados de tecnología JFET, para tener una mayor eficiencia en los circuitos implementados, teniendo menos caídas de voltaje (desbalance en la señal analógica adquirida) y dando la oportunidad de manejar altas velocidades en la transferencia de bits (básicamente en la parte de lógica digital del circuito de auto-rango).

Debido a la velocidad con que se manejan los datos en el circuito de auto-rango, fue necesario implementar un circuito de fase de auto-cero, el cual se necesita en el manejo del Offset de la señal eléctrica adquirida, ya que después de un pequeño intervalo de tiempo de funcionamiento del circuito se produce un pequeño desbalance en la señal eléctrica (offset del sistema), debido a las características de los circuitos integrados digitales, así como de los circuitos integrados lineales, que producen efectos notables de Offset en el circuito de acondicionamiento y adquisición de la señal eléctrica.

Ya que se está utilizando el convertidor A/D de aproximaciones sucesivas del MCU68HC11F1, se tienen dos formas para hacer un ajuste en la fase de

auto-cero. Una forma es ajustar por medio del convertidor A/D, de tal forma que se pueda obtener un resultado apropiado para el rango aceptable del convertidor; y la otra forma es hacer una corrección de la señal antes de iniciar la conversión A/D, es decir aproximar al valor real de la señal de entrada. Esto último se logra conectando el circuito a un capacitor en un tiempo muy corto, para que se cargue con el voltaje de Offset de la señal del circuito, y se descargue después en el circuito compensador, compensando de esta forma el Offset en la señal eléctrica que se entrega al convertidor A/D y tener como resultado el ajuste requerido en la señal adquirida, dando una mayor precisión en la adquisición de la señal eléctrica.

Cabe señalar que para protección del circuito se utilizó un arreglo con diodos tipo 1N914, de tal forma que limitara el voltaje o recortara la señal (en el caso de una señal alterna), así como para evitar daño en los C.I., como el multiplexor analógico y en los amplificadores operacionales, dado que en el momento de hacer los cambios de escala por medio de los switches analógicos se puede seleccionar la red de amplificación o bien la de atenuación, y en ese intervalo de tiempo de switcheo, se podrían causar daños a los C.I. digitales o lineales si no se tuvieran los diodos de protección.

Se propone manejar una frecuencia del orden de 100 KHz en el reloj que habilita al contador, para tener una buena velocidad de cambio de escala. Con esto se determina la velocidad de respuesta en el circuito de selección del rango de atenuación/amplificación.

9.2.4. Captura de una Señal Eléctrica

Como ya se mencionó antes, el MCU68HC11F1 tiene su propio convertidor A/D de 8 bits, el cual sólo puede convertir voltajes que estén entre los valores establecidos en V_H y V_L (pines 67 y 68 del MCU68HC11F1). Al pin del Convertidor A/D (pin 59) se le introduce una señal que tendrá un V_L de 0 V y un V_H de 1 V provenientes del Sistema de Acondicionamiento de la Señal Eléctrica.

9.2.4.1 Conversión A/D de una Señal Eléctrica

Como ya se mencionó en el Capítulo 7, el MCU68HC11F1 cuenta con 4 entradas para utilizar 4 diferentes convertidores A/D. Para este sistema sólo se va a ocupar un solo convertidor A/D, por lo cual se debe configurar el byte de Opción (\$1039) del MCU68HC11F1, por medio de las siguientes instrucciones:

```
LDAA #$80 ; Se establece que el MCU68HC11F1 va a utilizar el  
STAA $1039 ; Convertidor A/D configurándolo con el Registro  
de Opción de Configuración del Sist. (OPTION).
```

El sistema adecuador de señales antes mencionado proporciona al MCU68HC11F1 la Escala y el Signo del Voltaje que tiene la señal eléctrica al ser capturada. Bajo estas condiciones hay que programar el Puerto G del MCU68HC11F1 para que utilice sus pines 3, 4 y 5 como entradas para la Escala de Voltaje y el pin 6 como entrada para obtener el Signo del Voltaje (0 = positivo; 1 = negativo) de la señal eléctrica a analizar. El puerto G (\$1002) se programa para utilizar sus pines como entrada (asignándoles un 0 lógico) o como salidas (asignándoles un 1 lógico) a través del registro \$1003. A continuación se muestran las líneas para programar el Puerto G.

```
LDAA #$80 ; Se carga 80h, para configurar el Pto. G, donde  
STAA $1003 ; sólo el bit 7 será de escr. y el resto de lect.
```

Cabe señalar que el pines 7 del Puerto G se configura como salida, ya que dicho pin servirá para indicar al sistema de acondicionamiento de señales el momento en que tiene que adecuar la señal, por ejemplo, cuando dicho pin tenga un 1 lógico el de sistema de acondicionamiento de señales eléctricas ajustará la señal eléctrica para que el MCU68HC11F1 pueda hacer la conversión A/D, pero cuando tengo un 0 lógico, el sistema de acondicionamiento no funcionará. El diagrama de flujo de la Figura 9.13 muestra como se hace una conversión A/D de una señal eléctrica.

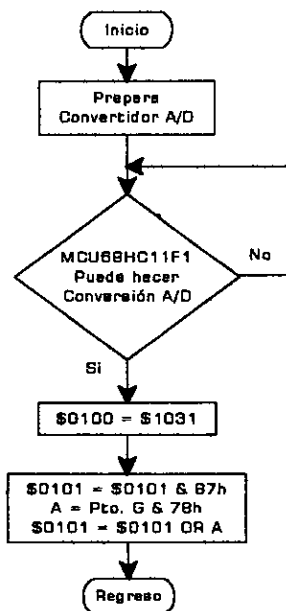


Figura 9.13: Diagrama de Flujo para Hacer la Conversión A/D

Aquí se observa que primero se prepara al MCU68HC11F1 para que empiece a funcionar el A/D para después verificar a través del registro \$1030 (registro de control del convertidor A/D) si ya puede realizar en ese momento una conversión

A/D. Ya que se ha verificado que se pueda hacer una conversión A/D se procede a leer el valor digital de dicha conversión y se deposita en el byte de Dato (\$0100). A continuación se muestran las instrucciones en lenguaje Ensamblador las instrucciones para hacer lo descrito anteriormente.

```

LDAA #$A0 ; Carga el Reg. A con A0h (1010 0000) para ponerlo
STAA $1030 ; en el Reg. de Control del A/D (ADCTL).
CN2: LDAA $1030 ; Carga el contenido del Reg. de Control del A/D
      ANDA #$80 ; (ADCTL) en el Reg. A para hacerle una AND con
      BEQ CN2   ; 80h y ver si ya se hizo la conversión A/D. Si
                ; no la ha hecho va a CN2.
      LDAA $1031 ; Carga el contenido del Reg. 1 de Resultados del
                ; A/D (ADRI) en el Reg. A.
      STAA $0100 ; Pone la muestra de la conversión A/D de la señal
                ; eléc. en el Reg. $0100 de RAM (Byte de Dato).
    
```

Ya que se tiene el valor digital de la Conversión A/D, se procede a tomar el valor de Escala y Signo de la señal eléctrica para depositarlo en el byte de Información (\$0101). Como no todo el byte de Información tiene datos de la Escala de Voltaje y Signo de la Señal eléctrica, se procede a hacer 0 los 4 bits que ocupan en dicho byte, pero sin perder el contenido de los otros bits de ese byte. Después se leen los pines 3, 4, 5 y 6 del Puerto G, que es donde llegará la información del Signo y la Escala de Voltaje de la Señal. Por último a través de una OR lógica se unen los valores leídos por el puerto G y el resto de bits del byte de información, para dejar el resultado en el Byte de Información. A continuación se muestra el código en ensamblador para hacer esto.

```

LDAA $0101 ; Carga en Reg. A la Dir. $0101 de RAM (Byte INFO)
ANDA #$87 ; para hacerle una AND con 87h y ponerlo otra
STAA $0101 ; vez en la Dir. $0101.
LDAA $1002 ; Carga en Reg. A el contenido del Pto. G que
ANDA #$78 ; contiene la Escala de Voltaje con signo de la
                ; señal eléctrica y le hace una AND con 78h.
ORAA $0101 ; Hace una OR del Registro A con la Dir. $0101
STAA $0101 ; poniendo así la Escala de Voltaje de la
                ; Conversión A/D en la Dir. $0101 (Byte INFO).
    
```

9.2.5 Codificación de la Información a Transmitir

9.2.5.1 Diseño del Código

Para que la información recibida fuera más confiable, se tomó la decisión de implementar un código detector y localizador de errores, en este caso se optó por el código BCH, ya que es uno de los más fáciles de implementar. Para implementar este código se requería de una dimensión de 16 bits, ya que ésta nos da la capacidad de incluir el dato de una Conversión A/D, su Escala de Voltaje, Signo de Voltaje y Periodo de una señal capturada, o bien la información del teclado remoto.

Los códigos que se aproximan más a dar una dimensión de 16 bits son los códigos de longitud de 31 bits, es decir los códigos sobre el $GF(2^5)$. Con el siguiente procedimiento (desarrollado en Visual Basic) se calcularon los polinomios mínimos de los elementos del $GF(2^5)$.

```
Sub BtnMP_Click ()
' Esta función calcula el polinomio mínimo del elemento a^p, donde a^p
' es un elemento de GF(2^5)
Dim i0, i1, i2, i3, i4, i5, i6 As Integer 'Coeficientes del polinomio.
Dim res As Integer ' Contendrá el res. de la evaluación.
Dim p As Integer ' Es el coeficiente del elemento a evaluar.
Dim aux As Integer ' Indica si algún coeficiente difiere de cero.
Static MP(7) As Integer ' Contiene los coeficientes.

p = InputBox("Coeficiente p" + Format(p, ""))
res = 1
For i6 = 0 To 1 ' Genera los coeficientes de X^6.
    MP(6) = i6
    For i5 = 0 To 1 ' los de X^5.
        MP(5) = i5
        For i4 = 0 To 1 ' los de X^4.
            MP(4) = i4
            For i3 = 0 To 1 'los de X^3.
                MP(3) = i3
                For i2 = 0 To 1 ' los de X^2.
                    MP(2) = i2
                    For i1 = 0 To 1 ' los de X.
                        MP(1) = i1
```

```

For i0 = 0 To 1 ' término indepen.
    MP(0) = i0
    res = -1 ' inicializa con cero la eval.
    If i0 = 1 Then
        res = 0
    End If
    aux = 0
    For j = 1 To 6
        If MP(j) Then
            res = suma(res, p * j)
        End If
        ' Hace la evaluación del polinomio.
        aux = aux + MP(j)
    Next j
    If (res = -1) And (aux > 0) Then
        Exit For
    End If
Next i0
If (res = -1) And (aux > 0) Then
    Exit For
End If
Next i1
If (res = -1) And (aux > 0) Then
    Exit For
End If
Next i2
If (res = -1) And (aux > 0) Then
    Exit For
End If
Next i3
If (res = -1) And (aux > 0) Then
    Exit For
End If
Next i4
If (res = -1) And (aux > 0) Then
    Exit For
End If
Next i5
If (res = -1) And (aux > 0) Then
    Exit For
End If
' sale si la evaluación fue igual a cero y alguno de los
' coeficientes es diferente de cero (para todos los if).
Next i6
Debug.Print "Polinomio Mínimo"
Debug.Print "p= "; p
For j = 0 To 6 ' Imprime el polinomio mínimo.
    Debug.Print "MP("; j; ")= "; MP(j)
Next j
End Sub

```

Lo que hace esta función es evaluar el elemento p para todos los polinomios de grado uno hasta grado seis, en cuanto alguna de estas evaluaciones es igual con cero y el polinomio tiene algún coeficiente diferente de cero, la función sale de todas los bucles a los que entró e imprime el polinomio resultante. Recuerde que el polinomio mínimo de un elemento es el polinomio de menor grado en donde al evaluar dicho elemento, es cero.

La función *suma* realiza la operación suma sobre el $GF(2^5)$. Esta función será explicada en la parte de decodificación, aunque no realiza la operación sobre el $GF(2^5)$, sino sobre el $GF(2^6)$, el funcionamiento es el mismo; lo único que varía, son los elementos. La función suma lleva como parámetros los coeficientes de los elementos a sumar. Al manejar sólo los coeficientes cuando se realiza la operación $p * j$, en realidad se está ejecutando la operación $(\alpha^p)^j$.

Los polinomios mínimos para algunos elementos de $GF(2^5)$ son que se muestran en la Tabla 9.4.

Elementos	Polinomio Mínimo
$\alpha, \alpha^2, \alpha^4, \alpha^8, \alpha^{16}$	$M_1(x) = x^5 + x^2 + 1$
$\alpha^3, \alpha^6, \alpha^{12}$	$M_3(x) = x^5 + x^4 + x^3 + x^2 + 1$
$\alpha^5, \alpha^9, \alpha^{10}$	$M_5(x) = x^5 + x^4 + x^2 + x + 1$
α^7, α^{14}	$M_7(x) = x^5 + x^3 + x^2 + x + 1$
α^{11}, α^{13}	$M_{11}(x) = x^5 + x^4 + x^3 + x + 1$
α^{15}	$M_{15}(x) = x^5 + x^3 + 1$

Tabla 9.4: Polinomios Mínimos de Algunos Elementos de $GF(2^5)$

A partir de la Tabla 9.4 se obtiene la Tabla 9.5.

Errores a Corregir	Distancia Mínima Diseñada (d)	Polinomio Generador	Grado del Polinomio Generador	Magnitud del Código (k)
1	3	$M_1(x)$	5	26
2	5	$M_1(x)M_3(x)$	10	21
3	7	$M_1(x)M_3(x)M_5(x)$	15	16
4	9	$M_1(x)M_3(x)M_5(x)M_7(x)$	20	11
5	11	$M_1(x)M_3(x)M_5(x)M_7(x)$	20	11
6	13	$M_1(x)M_3(x)M_5(x)M_7(x)M_{11}(x)$	25	6
7	15	$M_1(x)M_3(x)M_5(x)M_7(x)M_{11}(x)$	25	6
8	17	$M_1(x)M_3(x)M_5(x)M_7(x)M_{11}(x)M_{13}(x)$	30	1
Longitud $n = 31$				

Tabla 9.5: Polinomios Generadores de los Códigos del $GF(2^5)$

No hay que olvidar que la magnitud de un código, es la longitud menos el grado del polinomio generador. El polinomio generador es la multiplicación de los polinomios mínimos de α hasta α^{d-1} .

Observando la Tabla 9.5, nos damos cuenta que el código que nos permite una dimensión de 16 bits, sólo tiene la capacidad de corregir tres errores. Este código es el binario BCH (31, 16, 7), donde dicho código no nos será muy útil puesto que no nos permite corregir un número considerable de errores.

Ahora se diseñarán algunos códigos de longitud 63, es decir los códigos que operan sobre $GF(2^6)$. De la misma forma que se obtuvieron los polinomios mínimos para algunos elementos de $GF(2^5)$, se calculan algunos para $GF(2^6)$, los cuales se muestran en la Tabla 9.6.

Elementos	Polinomio Mínimo
$\alpha, \alpha^2, \alpha^4, \alpha^8, \alpha^{16}$	$M_1(x) = x^6 + x + 1$
$\alpha^3, \alpha^6, \alpha^{12}$	$M_3(x) = x^6 + x^4 + x^2 + x + 1$
$\alpha^5, \alpha^{10}, \alpha^{17}, \alpha^{20}$	$M_5(x) = x^6 + x^5 + x^2 + x + 1$
α^7, α^{14}	$M_7(x) = x^6 + x^5 + 1$
α^9, α^{18}	$M_9(x) = x^6 + x^5 + 1$
α^{11}, α^{22}	$M_{11}(x) = x^6 + x^5 + x^3 + x^2 + 1$
α^{13}, α^{19}	$M_{13}(x) = x^6 + x^4 + x^3 + x + 1$
α^{15}	$M_{15}(x) = x^6 + x^5 + x^4 + x^2 + 1$
α^{21}	$M_{21}(x) = x^6 + x + 1$

Tabla 9.6: Polinomios Mínimos de Algunos Elementos de $GF(2^6)$

A partir de la Tabla 9.6 se obtiene la 9.7.

Errores a Corregir	Distancia Mínima Diseñada (d)	Polinomio Generador	Grado del P.G.	k
1	3	$M_1(x)$	6	57
2	5	$M_1(x)M_3(x)$	12	51
3	7	$M_1(x)M_3(x)M_5(x)$	18	45
4	9	$M_1(x)M_3(x)M_5(x)M_7(x)$	24	39
5	11	$M_1(x)M_3(x)M_5(x)M_7(x)M_9(x)$	27	36
6	13	$M_1(x)M_3(x)M_5(x)M_7(x)M_9(x)M_{11}(x)$	33	30
7	15	$M_1(x)M_3(x)M_5(x)M_7(x)M_9(x)M_{11}(x)M_{13}(x)$	39	24
8	17	$M_1(x)M_3(x)M_5(x)M_7(x)M_9(x)M_{11}(x)M_{13}(x)M_{15}(x)$	45	18
9	19	$M_1(x)M_3(x)M_5(x)M_7(x)M_9(x)M_{11}(x)M_{13}(x)M_{15}(x)$	45	18
10	21	$M_1(x)M_3(x)M_5(x)M_7(x)M_9(x)M_{11}(x)M_{13}(x)M_{15}(x)$	45	18
11	23	$M_1(x)M_3(x)M_5(x)M_7(x)M_9(x)M_{11}(x)M_{13}(x)M_{15}(x)M_{21}(x)$	47	16

Longitud $n = 63$

Tabla 9.7: Polinomios Generadores de Algunos Códigos del $GF(2^6)$

El código que nos permite tener una dimensión de 16 bits es el binario BCH (63, 16, 23). Este código nos da la capacidad de corregir un máximo de 11 errores, por lo que es el código factible para el sistema de detección y corrección de errores del Módulo de Osciloscopio.

La función generadora resultante es:

$$G(x) = x^{47} + x^{46} + x^{43} + x^{42} + x^{40} + x^{39} + x^{36} + x^{33} + x^{32} + x^{27} + x^{25} + x^{24} + x^{23} + x^{22} + x^{20} + x^{19} + x^{18} + x^{16} + x^{13} + x^{12} + x^{11} + x^9 + x^8 + x^5 + x^3 + x + 1$$

Para codificar la información sólo hay que multiplicar el polinomio asociado a la información por el polinomio generador. Los polinomios también pueden representarse como vectores de elementos binarios, de esta manera se considerarán para la codificación.

9.2.5.2 Rutina de Codificación

La rutina de codificación se encarga de multiplicar el polinomio generador por la información a transmitir, esta rutina forma parte de lo que es, el sistema de control y obviamente está desarrollada en lenguaje ensamblador para el MCU68HC11F1. Para que se entienda mejor qué es lo que hace esta función se explicará gráficamente lo que realiza con un pequeño ejemplo.

Suponga que la función generadora es $G(x) = x^2 + x + 1$, así que el vector asociado es $G = (111)$, ahora suponga que la información a codificar es $u = (10)$, la multiplicación se efectúa de la siguiente manera:

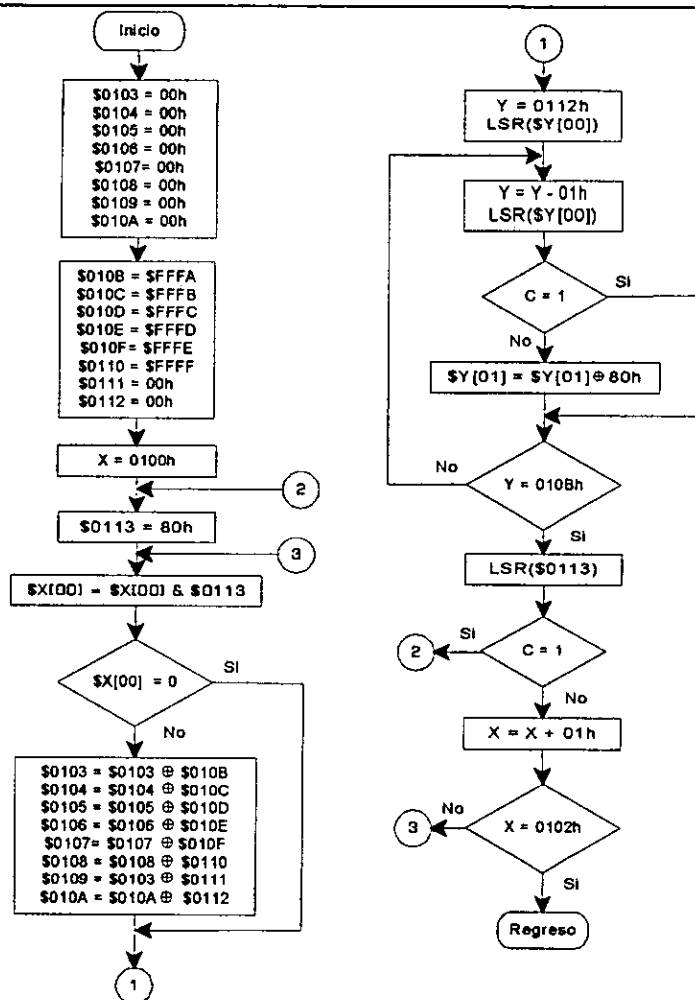
G:	1	1	1	
u:	X	1	0	1
Producto Parcial 1:	1	1	1	
Producto Parcial 2:		0	0	0
Producto Parcial 3:	Exor		1	1
Producto Total:	1	1	0	1

Observe que el *Producto Parcial 3*, son dos corrimientos a la derecha del vector generador. Para examinar el vector a codificar se emplea un vector testigo que al principio se le asigna un 1 en el bit más significativo, después se hacen corrimientos hacia la derecha, y con una *AND* se sabe si existe un 1 o un 0 en las diferentes posiciones del vector. El vector generador real se guarda en seis bytes. En un solo vector se almacenan los productos parciales, donde este vector abarca ocho bytes. El producto total es la palabra codificada.

Al principio, el vector que contiene los productos parciales es de la misma longitud que el del producto total, donde en los primeros bytes se le asigna el mismo valor que el de la función generadora y al resto del vector se le asignan ceros. Cada que se efectúa un producto por 0 o por 1, se hace un corrimiento hacia la derecha de este vector; para el caso de que sea un producto por 1, se hace un EXOR de este vector con el vector del producto total. En la Figura 9.14, se muestra el diagrama de flujo de la codificación.

La rutina de codificación se divide en varios procedimientos. El primer procedimiento se encarga de inicializar con ceros desde la direcciones \$0103 a la \$010A, que es en donde se almacenará la palabra codificada. Dicho procedimiento es el siguiente:

```
CODIF: LDY #$0103 ; Se carga en Reg. Y la 1er. Dir. donde estar la
          ; palabra a codificar.
CD1:  LDAA #$00 ; El contenido de las 8 Direcciones
      STAA $00,Y ; desde la Dirección $0103 hasta
      INY ; la Dirección $010A se limpia,
      CPY #$010B ; es decir se inicializarán
      BNE CD1 ; con cero.
```



Nota: El símbolo @ significa la operación EXOR. Lo que se encuentra dentro de los paréntesis [], son índices de los registros de la memoria. La operación LSR es un corrimiento a la derecha del registro.

Figura 9.14: Algoritmo de la Rutina de Codificación

El siguiente procedimiento copia desde la dirección \$010B a la \$0110, el contenido desde la dirección \$FFFA a la \$FFFF respectivamente. Estas direcciones contienen los bits correspondientes a los coeficientes del polinomio generador del código.

```

LDX # $FFFA ; Se carga en Reg. X la 1er. Dir. (Memoria EEPROM)
              ; de 6 del Vector Generador de Código,
CD4: LDAA $00,X ; Se carga uno a uno el contenido de las
      STAA $00,Y ; Direcciones apuntadas por X .
      INX       ; a las Direcciones apuntadas por Y,
      INY       ; es decir, se copia el contenido desde las Dir.
      CPY # $0111 ; $FFFA hasta. $FFFF a las Dir. de RAM desde
      BNE CD4    ; $010B hasta la Dir. $0110.

```

La multiplicación de la palabra a codificar se hace bit a bit, con la palabra generadora de código. Por medio de un byte testigo se recorre la palabra a codificar. Para saber en donde existe un 1, se hace una *AND* del byte testigo con el byte que se está analizando de la palabra a codificar. El byte testigo se encuentra en la dirección \$0113. Se inicializa con 80h, para asignar un 1 al bit más significativo. Después por medio de un corrimiento hacia la derecha y haciendo operaciones *AND* se analiza la palabra a transmitir. Como la palabra a transmitir utiliza dos bytes el procedimiento anterior se hace dos veces.

Cada vez que se detecta un 1 en la información se hace una *EXOR* de los bytes que contienen el vector código con los bytes que contienen el vector generador. El resultado se deposita en el vector código. Lo anterior significa la multiplicación del vector generador por 1. La primera vez que se ejecuta la rutina, el vector código es 0. En caso de que se detecte un 0, significa que se multiplicará el vector generador por un 0. Después el programa hace un corrimiento a la derecha del vector generador, y vuelve a evaluar la palabra a codificar. Esta operación se realiza

hasta que evalúa todos los bits de la palabra a codificar. Cuando la rutina termina, la palabra codificada se encuentra desde la dirección \$0103 a la \$010A. A continuación se muestra la rutina de codificación, descrita anteriormente.

```

LDAA #$00 ; Se carga 00h en el Registro A. dejándolo en las
STAA $0111 ; Direcciones $0111 y $0112, ya que es necesario
STAA $0112 ; que inicien en cero la codificación.
LDX #$0100 ; Se Carga en X la Dir. $0100, que es a partir de
           ; de donde se encuentran los 2 bytes a codificar.
CD5: LDAA #$80 ; Carga 80h en el Reg. A para dejarlo en la Dir.
STAA $0113 ; $0113 la cual servir como byte testigo de la
           ; codificación.

COD: LDAA $00,X ; Se carga en Reg. A el contenido de Dir. apuntada
ANDA $0113 ; por Reg. X, y hace una AND con el contenido de
BEQ CORR ; Dir. $0113 (Byte Testigo). Si el bit 6 es 1,
           ; brinca a CORR (hace corrimiento a la Derecha).
LDY #$0103 ; Se carga en Y la Dir. $0103 del Vector Generador.
CD2: LDAA $00,Y ; Carga en Reg. A el valor del byte apuntado por el
EORA $08,Y ; Reg. Y (Vector Generador) haciéndole un EXOR
STAA $00,Y ; con otro byte apuntado por Y (Vector de Código)
           ; poniendo el resul. en el byte del Vector Gener.
INY ; Incrementa Reg. Y hasta que sea igual a la Dir.
CPY #$010B ; $010B, mientras hace el resto EXOR con los
BNE CD2 ; bytes del Vector Generador de Código y Código.
CORR: LDY #$0112 ; Carga en Y la 1er. Dir. del Vector Generador,
LSR $00,Y ; para hacer un corrimiento hacia la derecha de
           ; dicho Vector.
CD3: DEY ; Decrementa Y, y hace corrimiento hacia la derecha
LSR $00,Y ; del contenido del byte apuntado por Reg Y.
BCC NOACA ; Si no hubo acarreo en el corrimiento va a NOACA.
LDAA #$80 ; Carga 80h en A para hacer un EXOR con siguiente
EORA $01,Y ; byte dejando el resultado en la Dir. apuntada
STAA $01,Y ; por Reg Y.
NOACA: CPY #$010B ; El corrimiento de todos los bytes del Vector
BNE CD3 ; Generador termina cuando el Reg. Y (como
           ; apuntador) es igual a la Dir. $010B.
           ; /****
LSR $0113 ; Hace un corrimiento del byte testigo para
           ; preguntar por el
BCC COD ; siguiente byte. En caso de que sea cero el
           ; resultado
INX ; Incrementa el la Dir. apuntada por X (Dir. de la
CPX #$0102 ; del corrimiento pregunta si ha terminado de
           ; codificar.
BNE CD5 ; Si la codificación no ha terminado brinca a CD5.
RTS ; Termina la codificación y la palabra codificada
           ; est en las Dir. $0103-$010A, siendo el byte
           ; menos significativo $0103.

```

9.2.6 Sistema de Transmisión

La transmisión inalámbrica se diseñó para que fuera a través de leds de luz infrarroja, ya que consideramos que éstos son el medio más económico que hay en el mercado y son fáciles de implementar.

Primero había que definir por donde saldría la información procesada (la codificación de la conversión A/D de la señal eléctrica capturada o la codificación de la combinación de teclas que se presionaron). La decisión fue fácil, ya que sólo hay dos formas de sacar información del MCU68HC11F1; una es por alguno de los puertos paralelos y la otra es por el puerto serial. Si se utilizaba algún puerto paralelo, había que utilizar por lo menos un led de luz infrarroja para cada uno de los ocho bits del puerto. Esto no sería práctico, puesto que se pensaba utilizar un esquema en el cual la información transmitida llegaría por varias direcciones. Lo anterior nos condujo a utilizar simplemente el Puerto Serial, como lo hace cualquier medio de comunicación remota de las Computadoras actuales.

Una vez definida la salida de la señal procesada a través del puerto serial era necesario determinar la frecuencia a la que iba a ser modulada la señal que se transmitiría. Dicha modulación tendría garantizar la seguridad en la transmisión, considerando que los Leds de luz infrarroja son muy vulnerables al ruido y son de muy poco alcance. En el mercado se consiguió el GPIU52X que es un Detector/Demodulador de luz infrarroja, como los que se usan en las televisiones y aparatos de sonido que se operan a control remoto. Según sus hojas de especificación la frecuencia a la que reciben sin ningún problema la señal modulada es de 40 KHz.

Considerando que con el MCU68HC11F1 se puede generar una señal cuadrada a cierta frecuencia a través de los Comparadores de Salida (OC), se concluyó que se podía obtener la señal portadora a 40 KHz, por medio de un OC. El mensaje sería la información codificada que sale por el Puerto Serial.

9.2.6.1 Modulación de la Información

Como ya se mencionó anteriormente la información procesada saldría por el Puerto Serie del MCU68HC11F1, para ello había que especificar una tasa de transmisión del puerto serie, la cual tendría que ser proporcional a la escala de la frecuencia central del detector/demodulador de luz infrarroja; por eso se seleccionó una velocidad de transmisión de 4800 bps. Además había que especificar que el MCU68HC11F1 sólo iba a transmitir y no a recibir. A continuación se muestran las instrucciones en ensamblador de esta configuración.

```
LDAA #$31    ; Se establece que el MCU68HC11F1 tendrá una veloc.  
STAA $102B   ; de transmisión de 4800 BPS, a través del Reg.  
             ; de Control de Bauds del SCI (BAUD).  
LDAA #$08    ; Se establece que el MCU68HC11F1 sólo va a  
STAA $102D   ; transmitir, configurándolo con el Registro 2 de  
             ; Control SCI (SCCR2).
```

Para poder generar la señal portadora de 40 KHz, como ya se mencionó, había que utilizar un Comparador de Salida del MCU68HC11F1, en este caso se utilizó, el Comparador de Salida 2 (OC2). El utilizar un OC implica utilizar un offset de comparación, para generar una interrupción cada vez que el contador de libre corrimiento del MCU68HC11F1 sea igual al valor del offset. Cada vez que se genera una interrupción sale una señal verificada alta por el pin 6 del puerto A, si previamente

se configuró dicho pin para ser utilizado como OC2. Esta señal verificada alta da un ciclo de trabajo del 50 %. La fórmula para calcular el offset es:

$$\text{Offset} = \frac{T_I}{T_E}$$

donde T_I es el tiempo que envía un 1 lógico por OC2 y T_E es el periodo de reloj al que trabaja el MCU68HC11F1. El MCU68HC11F1 fue configurado para que tuviera una frecuencia de trabajo de 2 MHz, por lo que T_E es:

$$T_E = \frac{1}{2 \text{ MHz}} = 0.5 \mu\text{s}$$

Como se quiere una frecuencia de 40 KHz con un ciclo de trabajo de 50% se tiene que:

$$T = \frac{1}{f} = \frac{1}{40 \text{ KHz}} = 25 \mu\text{s}$$

por lo tanto T_I es

$$T_I = \frac{T}{2} = \frac{0.5 \mu\text{s}}{2} = 12.5 \mu\text{s}$$

Entonces el offset es:

$$\text{Offset} = \frac{T_I}{T_E} = \frac{12.5 \mu\text{s}}{0.5 \mu\text{s}} = 25 = 19h$$

A continuación se muestra el código en ensamblador configurar el OC2 del MCU68HC11F1.


```
LDAA #$7E ; Carga el valor 7Eh (OpCode de la instrucción JMP)
STAA $DC ; en el 1er.byte del Pseudovector del Comparador
; 2 de Salida del Temporizador.
LDX #INPUL ; Carga el valor de la Dirección donde comienza la
STX $DD ; rutina INPUL en los otros dos bytes del
; Pseudovector del TOC2.
LDX #$0019 ; Carga el valor del Offset, el cual es de 40 KHz,
STX $00FE ; poniéndolo en la Dir. $00FE de RAM y en la Dir.
STX $1018 ; del Reg. 2 del Comparador de Salida (TOC2).
LDAA #$40 ; Carga en Reg. A 40h para configurar el Comparador
STAA $1020 ; de Salida 2 a través del Registro 1 de Control
STAA $1022 ; del Temporizador (TCTL1), del Reg. 1 de la
STAA $1023 ; Máscara de Interrup. del Temporizador (TMSK1) y
; del Reg. 1 de la Bandera de Interrup. del
; Temporizador (TFLG1).
CLI ; Limpia la Máscara de Interrupción.
```

A continuación se muestra el código en ensamblador para generar la señal portadora de 40KHz.

```
INPUL: LDD $1018 ; Carga valor del Reg. 2 del Comparador de Salida
ADDD $00FE ; (TOC2) y suma el Offset de 40 KHz (almacenado
STD $1018 ; en Dir. $00FE de RAM) para que luego se deje en
; en el Reg. 2 del Comparador de Salida (TOC2).
LDAA #$40 ; Carga 40h en Reg. A para indicar que se hizo
STAA $1023 ; la interrupción, por medio de Reg. 1 de Bandera
; de Interrupción del Temporizador (TFLG1).
```

9.2.6.2 Transmisión de la Información

En las secciones anteriores se explicó cómo se iba a codificar la información y cómo se iba a transmitir. En la Figura 9.15 se muestra un diagrama del procedimiento que se ocupa para poner la información codificada en el Puerto Serie del MCU68HC11F1.

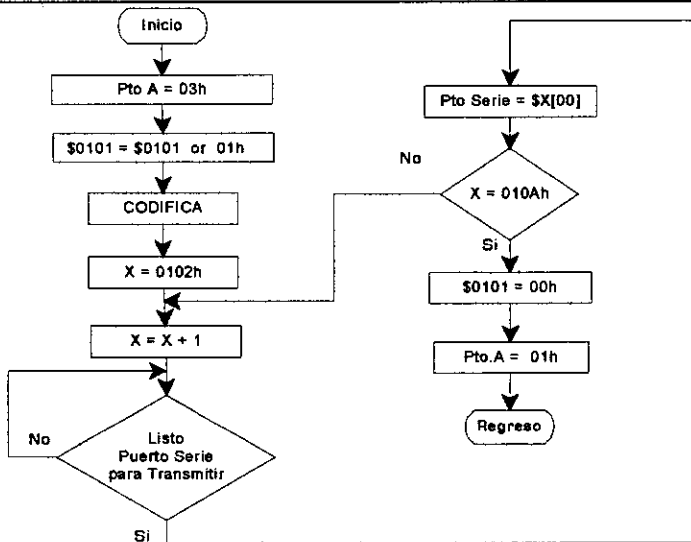


Figura 9.15: Diagrama de Flujo para Poner la Información Codificada en el Puerto Serie

Primero se sacan dos señales verificadas altas por los pines menos significativos del Puerto A; por medio del pin 0 se indica si el programa está funcionando y por medio del pin 1 se indica si el MCU68HC11F1 está ejecutando el procedimiento de transmisión. Después se pone un 1 lógico en el bit 0 (T) del Byte \$0101 (Byte de información) haciendo una operación OR con 01h, ya que este bit servirá en el programa donde se reciba la información, para saber si la palabra recibida es una palabra enviada. Posteriormente codifica la palabra, generando así ocho bytes como ya fue explicado a detalle en la sección de codificación de la información. Las instrucciones en lenguaje ensamblador para hacer esto son las siguientes.

```

LDAA #03      ; Pone 03h en Pto. A, indicación del inicio de
STAA $1000    ; trans. de palabra codif., por el Pto. Serie.
LDAA $0101    ; Carga en Reg. A lo que tiene la Dir. $0101 (Byte
ORAA #01      ; de Información) de RAM para hacerle una OR con
STAA $0101    ; 01h, (asegura así que se transmite un dato) y
  
```

Módulo de Osciloscopio

```
JSR CODIF ; después se pone otra vez en Dir. $0101 de RAM.  
; Brinca a la Rutina CODIF de Codificación.
```

El siguiente paso es poner cada uno de los ocho bytes que representan la información codificada de los byte \$0100 y \$0101, en el puerto serie del MCU68HC11F1. Cabe recordar que estos ocho bytes se encuentran desde la dirección \$0103 hasta la \$010A de RAM. Cuando el MCU68HC11F1 envía un byte por su Puerto Serie, espera hasta que se pueda realizar la transmisión de otro byte, por lo que se debe actualizar la bandera de "dato enviado". A continuación se muestra el código en ensamblador que pone cada uno de los ocho bytes codificados en el puerto serie.

```
LDX #$0102 ; Carga 0102h (correspondiente a Dir. $0102 de RAM)  
; en el Registro X.  
TR1: INX ; Incrementa el valor del Reg. X (apuntador de Dir.  
; de RAM).  
TR2: LDAA $02E ; Carga en Reg. A el valor del Reg. de Estado del  
ANDA #$40 ; SCI (SCSR), para hacerle una AND con 40h. El 6  
BEQ TR2 ; indica, si ya se puede transmitir, si no se  
; puede todavía transmitir, brinca a TR2.  
LDAA $00,X ; Carga en Reg. A, lo que tenga la primer. Dir.  
STAA $02F ; Indexada de RAM, apuntada por el Reg. X, y se  
; pone en Reg. de Datos SCI (byte a transmitir).  
CPX #$010A ; Compara 0106h (Dirección $0106 de RAM) con el  
BNE TR1 ; contenido del Reg. X (apuntador de RAM), si son  
; diferentes ambos valores, brinca a TR1.
```

Después se limpia el byte \$0101 (byte de información), por si la punta inalámbrica transmite alguna palabra, para que no sea tomado en cuenta por el programa de recepción, ya que el bit 0 (T) del byte \$0101 está en 0. Por último se indica poniendo un 0 lógico en el pin 1 del Puerto A, que ya se terminó de ejecutar el procedimiento de transmisión.

```
LDAA #$00 ; Carga en Reg. A 00he para ponerlo en Dir. $0101  
STAA $0101 ; (Byte INFO) asegurando así que transmitió la  
; palabra.  
LDAA #$01 ; Pone 01h en Pto. A, indicación del fin de transm.  
STAA $1000 ; de la palabra codificada, por el Pto. Serie.
```

9.2.6.3 Transmisor

Puesto que la información que se transmite es por medio de luz infrarroja, es necesario polarizar unos ILEDs (Diodos Emisores de Luz Infrarroja) con el mismo voltaje de polarización del MCU68HC11F1, el cual es de 5 V. Las hojas de especificaciones de los ILEDs que se utilizan, indican que la máxima corriente en sentido directo es de 50 mA y el voltaje en sentido directo es de 1.2 V. Como la idea original era que ILEDs transmitieran en todas direcciones la información, tal vez se ocuparían más de dos ILEDs, lo que traería como consecuencia que el MCU68HC11F1 no trabajara adecuadamente ya que los ILEDs (los cuales estarían conectados al puerto serie del microprocesador) consumirían mucha corriente. Por eso se pensó en utilizar unos transistores como controladores de corriente de estos. Otra de las razones de utilizar controladores, fue que cuando el MCU68HC11F1 termina de enviar información por su puerto serie, éste mantiene una señal verificada alta (1 lógico), aunque no esté realizando algún proceso en ese momento, por lo que los ILEDs estarían encendidos. Los controladores de corriente están conectados en una configuración de inversores, los cuales se muestra en la Figura 9.16.

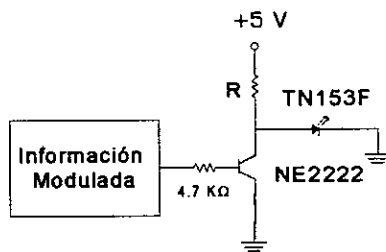


Figura 9.16: Polarización del Controlador de los ILEDs

donde:

$$R = \frac{5V - 1.2V}{0.05A} = 76\Omega$$

Ya determinado el valor de la resistencia para encender con la máxima corriente en sentido directo el ILED, había que ver cuál es el espectro de luz infrarroja radiada. Este espectro se determinó para dos dimensiones, midiendo la distancia contra grados, de luz infrarroja radiada. En la Figura 9.17 se muestra el resultado de este análisis.

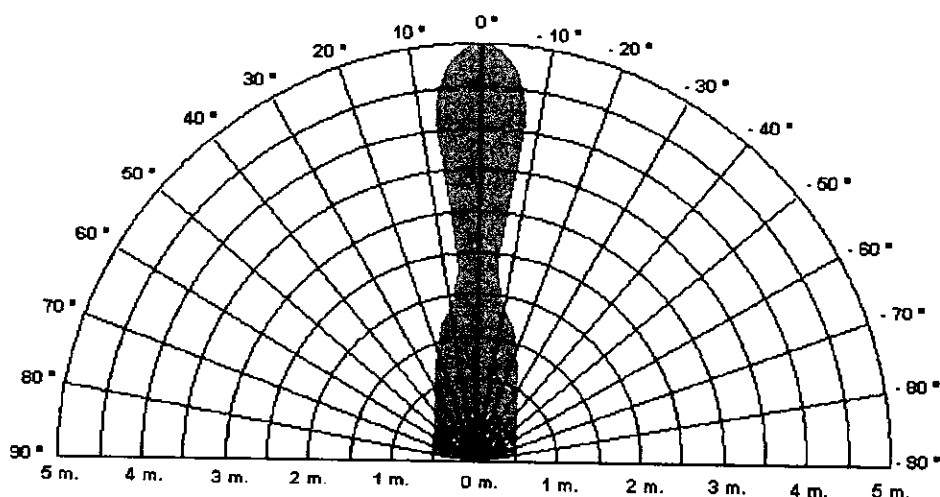


Figura 9.17: Espectro de Radiación de Luz Infrarroja de un ILED

Al analizar este espectro nos damos cuenta que tiene un muy buen alcance, de hasta cinco metros, pero su ángulo de radiación no es muy bueno ya que durante esos cinco metros casi siempre radia a 20°. Haciendo un cálculo estimado de si se quisiera radiar luz infrarroja en 180° planos, se requerirían de por lo menos 8

ILEDs, y si se quisiera radiar en 360° planos, tal vez se requerirían de hasta 16 ILEDs. Como la idea original era radiar a todas partes, se pensó por lo menos en radiar a 360° planos pero utilizando 8 ILEDs como máximo. El poner 8 leds como máximo implicaba colocarlos a cada 45° . Haciendo un análisis del espectro de radiación de luz infrarroja de 3 leds separados 45° entre ellos, y utilizando la configuración electrónica antes mencionada con una corriente máxima en sentido directo de 50 mA y un voltaje en sentido directo de 1.2 V donde R tendría que ser de 28Ω , nos daba el espectro de radiación que se muestra en la Figura 9.18.

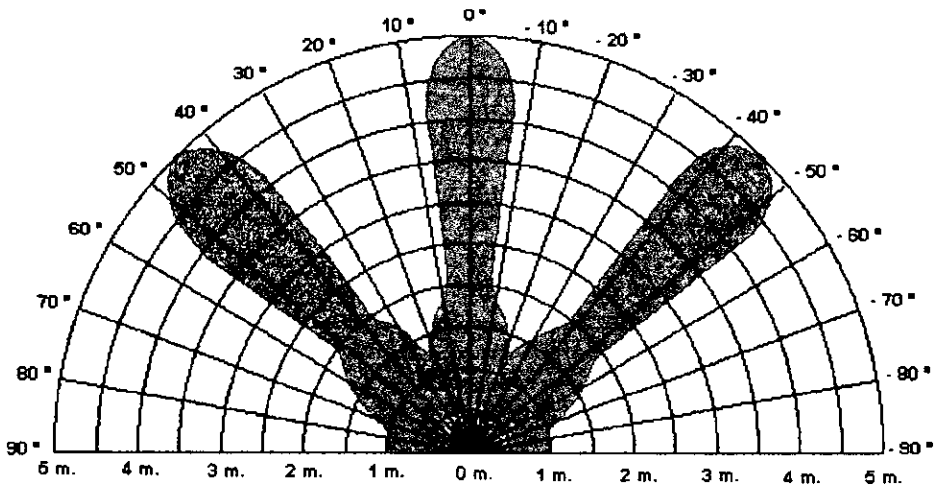


Figura 9.18: Espectro de Radiación Luz Infrarroja de Tres ILEDs

La Figura 9.18 muestra que el sistema de ILEDs separados a 45° uno del otro, garantiza una buena radiación antes de 1.5 m. de distancia, pero después se vuelve demasiado direccional, es decir que en la separación de 45° desde 1.5 m. hasta 4.5 m., se radia luz infrarroja aproximadamente en 15° (7.5° corresponde al ILED

referenciado a 0° y 7.5° del ILED referenciado a 45°), mientras que aproximadamente 30° ya no radia (el espacio entre los dos ILEDs). El poner más ILEDs para cubrir ese espacios vacíos de 30° , implica más estorbo en la "Punta Inalámbrica", por lo que se decidió dejarlo con 8 ILEDs, aunque sea muy direccional. En el mercado existen otros ILEDs cuyo ángulo de radiación de luz infrarroja es de hasta 60° , pero no tienen un buen alcance. En la Figura 9.19 se muestra el esquema electrónico empleado para utilizar 8 ILEDs, ya conectados al Puerto Serial del MCU68HC11F1 y a la señal moduladora de 40 KHz generada por él mismo.

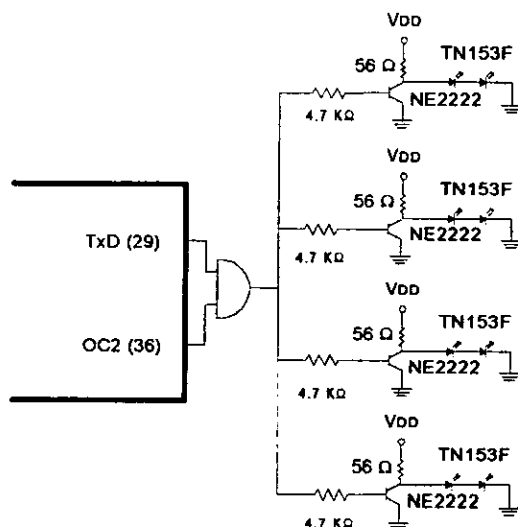


Figura 9.19: Esquema Electrónico de los 8 ILEDs Conectados al Puerto Serial del MCU68HC11F1

9.3 Recepción, Procesamiento y Graficación de la Señal

La Recepción, Procesamiento y Graficación es la segunda parte del desarrollo del Módulo de Osciloscopio, la cual se refiere al sistema electrónico del receptor/demodulador de luz infrarroja (que pondrá en alguno de los dos puertos seriales externos de una computadora PC, los datos transmitidos); al sistema detector de errores; y a la interpretación de la palabra, ya sea como una muestra componente de una gráfica de alguna señal eléctrica transmitida o una función para el control del programa "Módulo de Osciloscopio".

9.3.1 Sistema de Recepción

El sistema de recepción consiste del Hardware implementado para demodular la señal a 40 KHz transmitida por la punta inalámbrica y ponerla en alguno de los dos puertos seriales externos de la PC; y del Software desarrollado para leer del puerto serie, la información demodulada e interpretar si es un dato válido o no, que se transmitió por la punta inalámbrica.

9.3.1.1 Receptor

Para que la información transmitida inalámbricamente llegue a ser procesada por una computadora PC, el receptor se compone de un receptor/demodulador de luz infrarroja conocido comercialmente como GPIU52X, un inversor, un circuito integrado MAX232 y un DB9-hembra (estos dos últimos conocidos como sistema RS232, que sirve de interface entre el receptor y la computadora.

El GPIU52X (receptor/demodulador de luz infrarroja) es ampliamente utilizado en los aparatos electrónicos de control remoto y consiste de un amplificador y un filtro pasobandas de 40 KHz $\pm$ 4 KHz (el cual demodula las señales transmitidas a esta frecuencia), entre otras cosas. En el Apéndice D, se mencionan las características del GPIU52X. La salida del receptor/demodulador, actúa como un seguidor, por lo que hay que invertir la señal demodulada, ya que como se recordará en la configuración del transmisor, la información se invierte antes de transmitirse. Es por esta razón que se pone un inversor a la salida del receptor/demodulador.

Una vez invertida la señal recibida y demodulada, se procede a pasarla a través de un circuito integrado conocido como MAX232, porque tanto el GPIU52X como el inversor proporcionan sus salida en un nivel TTL o CMOS (0 y 5 V) mientras que la computadora maneja niveles de 10 V y -10 V por sus puertos seriales externos.

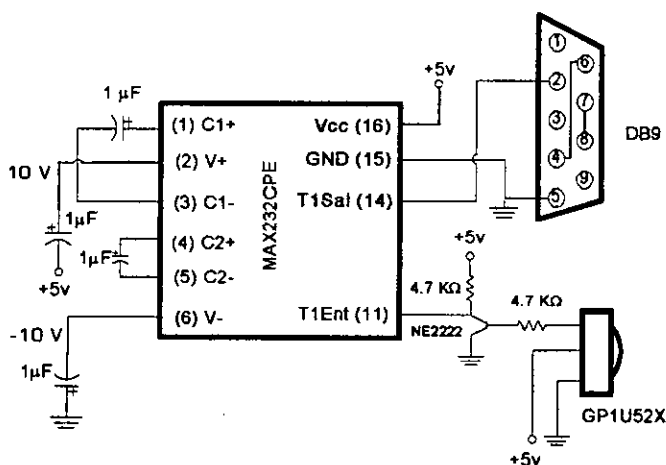


Figura 9.20: Esquema Electrónico del Receptor

Por último se requiere de un cable con un conector tipo DB-9 hembra en un extremo para conectar el receptor a los puertos seriales 1 o 2 de la computadora (si los puertos seriales son de 25 pines, se utiliza un convertidor a 25 pines). En la Figura 9.20 se muestra el esquema electrónico del receptor.

9.3.1.2 Recepción de la Información

Teniendo el receptor funcionando, se hizo una rutina para obtener la información demodulada por el puerto serie de la PC. El puerto serie se configura a 4800 BPS, puesto que como se recordará el MCU68HC11F1 proporciona la información procesada, a través de su puerto a esa misma velocidad. A continuación se muestra la rutina que configura el puerto serie a una velocidad de recepción de 4800 BPS, sin paridad, con 8 bits de datos y con 1 bit de paro.

```
Sub ConfiguraPuertoSerie(ByVal pto As Integer)
' Inicializa parámetros del Puerto Serial a utilizar.

Comm1.CommPort = pto ' Asigna el Puerto Serial requerido.
Comm1.InBufferSize = 7168 ' Asigna buffer de 7 Kb al Puerto Serie.
Comm1.PortOpen = True ' Abre el puerto
Comm1.Settings = "4800,N,8,1" ' 4800 bps, sin paridad, 8 datos y
                             ' 1 bit de paro.
Comm1.InputLen = 0 ' Para indicar al buffer de lectura cuando se
                  ' está usando el Puerto Serial como entrada.
End Sub
```

En esta rutina se observa que el buffer del puerto serie se programa para almacenar aproximadamente 7 KB de las palabras codificadas recibidas, ya que como se utilizaba una interface gráfica en el programa "Módulo de Osciloscopio" y el proceso de decodificación es a veces tardado, se podrían perder algunas palabras.

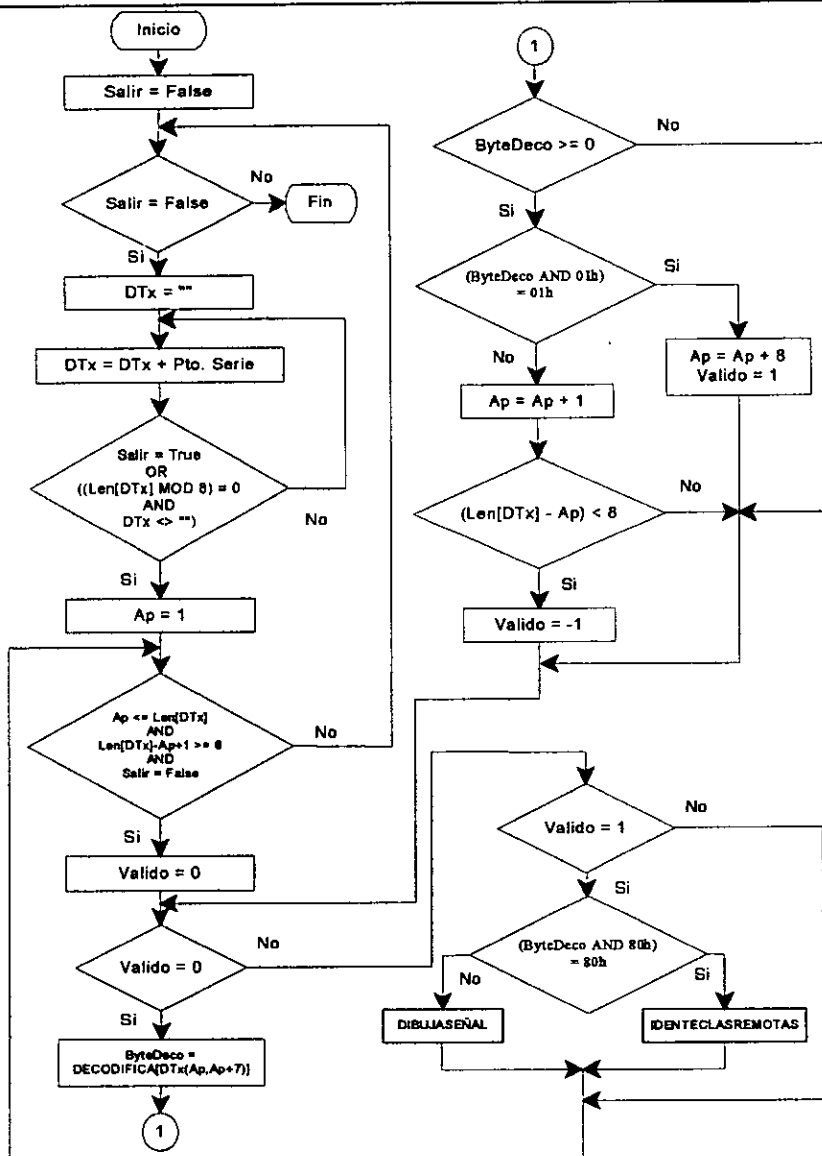


Figura 9.21: Diagrama de Flujo de la Rutina de Recepción

El siguiente paso es hacer una rutina de recepción que obtenga la información contenida en el puerto serie de la PC. En la Figura 9.21 se muestra el diagrama de flujo de esta rutina contenida en el programa "Módulo de Osciloscopio".

Hay que hacer notar que el programa "Módulo de Osciloscopio" sólo está dedicado a recibir la información que se le envía a la computadora inalámbricamente, por lo que mientras se está ejecutando, siempre va a estar en esta rutina de recepción. Esto no quiere decir que el usuario no pueda ejecutar otras aplicaciones en Microsoft Windows, ya que a través de la función *DoEvents* de Visual Basic, se logra que la PC no sólo se dedique a la ejecución del programa "Módulo de Osciloscopio". Además, si se quiere salir del programa se cuenta con la opción "Salir" en varias partes.

En el diagrama de flujo se muestra que la rutina de recepción tiene un ciclo que siempre se ejecuta hasta que se reciba la orden de salir del programa. Mientras se da esta orden, se ejecuta otro ciclo donde se recibe la información, la cual se va almacenado en el buffer del puerto serie. Una vez que se lee el buffer del puerto serie, se borran los bytes leídos, por lo que difícilmente se llenará. Es necesario utilizar una variable auxiliar tipo cadena, denominada DTx, la cual contendrá la información leída del buffer para su procesamiento. El ciclo donde se lee permanentemente el buffer del puerto serie terminará hasta que se reciba la orden de salir o bien que DTx tenga almacenados ocho o más bytes recibidos. Como se recordará, la información codificada, viene en ocho bytes. A continuación se muestra el código que se ejecuta mientras se recibe la orden de salir o se reciben paquetes de ocho bytes codificados.

Módulo de Osciloscopio

```
Salir = False ' No hay salida del Programa.
Do While Salir = False ' Mientras no se quiera salir del Programa.
    Dtx = ""
    Do ' Mientras no reciba Bytes o no se quiera salir del progr.
        Dummy = DoEvents() ' Hace que Microsoft Windows pueda
                            ' procesar otros eventos.
        Dtx = Dtx + Comm1.Input ' Lee Puerto Serial.
    Loop Until (((Len(Dtx) Mod 8) = 0) And Len(Dtx)) Or Salir
    ...
Loop
Comm1.PortOpen = False ' Cierra asignación del Puerto Serial.
End ' Finaliza el programa.
```

Cabe hacer notar que DTx se limpia antes de entrar al ciclo donde lee la información contenida en el buffer del puerto serie. Esto se hace porque se supone que al estar haciendo el primer ciclo, ya se utilizó DTx. Los puntos suspensivos (...) indican que hay más código ahí. Las últimas dos líneas de código se ejecutan cuando se recibe la orden de salir, para indicar a la PC que ya no se utilizará el puerto serie y se termine el programa.

Existe dos formas de verificar si se recibieron correctamente los bytes: una es a través del procedimiento de detección y corrección de errores que está en la función *Decodificar* y la otra forma es revisando el Byte de Información. En la siguiente sección, se explica la función *Decodificar*, por lo que ahora sólo se mencionará, que si la palabra fue corregida (si es el caso) y decodificada exitosamente, dicha función regresa la palabra ya decodificada en un número entero largo (de 4 bytes), de lo contrario regresa un número negativo.

La segunda forma consiste en verificar el estado lógico del bit 0 (T) del Byte de Información, donde este bit es 1, cada vez que se transmite información desde la Punta Inalámbrica. Entonces, si no fuera 1 al decodificarse la palabra, se

decodificaron 8 bytes que no eran los correctos, por lo que hay que hacer un corrimiento de un byte (por medio de *Ap*) del contenido en *DTx* y volver a decodificar el paquete de 8 bytes hasta que esté en 1 ese bit. Si no se llegara a formar un paquete de 8 bytes contenidos en *DTx* al estar haciendo corrimientos, se indica por medio de la bandera *Válido* poniéndole un número negativo, que no se hizo una decodificación de la palabra. Si se encontró que el bit 0 (*T*) está en 1, la bandera *Válido* toma el valor de 1. El código que a continuación se muestra, realiza lo antes descrito.

```

Ap = 1 ' inicio de 8 Bytes para identificar bit de Transmisión.
Do While (Ap <= Len(Dtx)) And ((Len(Dtx) - Ap + 1) >= 8) And
(Salir = False)
    Válido = 0 ' No se valida par de Bytes.
    Do While Válido = 0 'Mientras no exista par de Bytes Válidos.
        ByteDeco = Decodificar(Mid$(Dtx, Ap, 8))
        ' Decodifica paquete de ocho bytes recibidos.
        If ByteDeco >= 0 Then ' Si NO se detectó error al Decodif.
            If (ByteDeco And &H1) = &H1 Then
                ' Si se detecta bit de transmisión.
                Válido = 1 ' Par de Bytes recibidos Correctos.
                Ap = Ap + 8 ' Avanza paquete de 8 bytes.
            Else ' Si NO se ha detectado bit de Transmisión.
                Ap = Ap + 1 ' Avanza un byte.
                If ((Len(Dtx) - Ap) < 8) Then
                    ' Si ya no hay bytes Acumulados.
                    Válido = -1 ' NO se detectó bit de Transmisión.
                End If
            End If
        End If
    End If
Loop

```

Cuando *Válido* es 1, se procede a identificar si se presionó alguna combinación de teclas de la Punta Inalámbrica, para realizar cierta función en el programa o si es una muestra de la gráfica de alguna señal eléctrica. Hay que recordar que si es 1 el bit 7 (*D/F*) del Byte de Información, la palabra transmitida contiene alguna de las 7 combinaciones de teclas de la Punta Inalámbrica; pero si ese bit es 0,

entonces la palabra, contiene una muestra de la gráfica de alguna señal eléctrica. A continuación se muestra el código, que efectúa lo mencionado

```
If Válido = 1 Then ' Si existe par de bytes válidos.  
  If (ByteDeco And &H80) = &H80 Then  
    ' Si se detectó que se utilizó el teclado remoto.  
    IdemTeclasRemotas ByteDeco  
  Else ' Si se detectó una Muestra de señal eléctrica.  
    DibujaSeñal frmOsciloscopio, ByteDeco, 1  
  End If  
End If
```

9.3.2 Sistema de Detección y Corrección de Errores

Parte del sistema de detección y corrección de errores, se encuentra en la codificación y la otra parte en la decodificación. Cuando la PC recibe los datos, antes de ser procesados, pasan por una rutina llamada *Decodificar*, la cual se encarga de detectar y corregir errores, además de decodificar la información. *Decodificar* recibe una cadena de 8 caracteres y regresa un número entero largo de cuatro bytes. En esta sección nos enfocaremos a explicar las funciones *GenerarElementos*, *Suma* y *Decodificar* que en conjunto logran la decodificación.

La rutina *GenerarElementos*, genera los elementos de $GF(2^6)$, y se ejecuta cuando inicia el programa "Módulo de Osciloscopio".

El polinomio primitivo de grado 6 de $GF(2)$ es: $1 + x + x^6$, suponiendo que α es raíz del polinomio, entonces $\alpha^6 = \alpha + 1$, de acuerdo con lo estudiado en el Capítulo 6. Expresando lo anterior en forma vectorial, $\alpha^6 = 000011$. Con lo anterior se obtiene la Tabla 9.8, donde se observa que para generar un elemento

hay que hacer un corrimiento hacia la izquierda del elemento anterior y cuando existe un acarreo se debe hacer un EXOR del resultado del corrimiento del elemento con 000011, ya que es el vector asociado al polinomio generador del campo. Los fundamentos de lo anterior fueron explicados en el Capítulo 6.

Elem.	Vector	Elem.	Vector	Elem.	Vector
α^0	000001	α^{22}	110101	α^{43}	110111
α^1	000010	α^{23}	101001	α^{44}	101101
α^2	000100	α^{24}	010001	α^{45}	011001
α^3	001000	α^{25}	100010	α^{46}	110010
α^4	010000	α^{26}	000111	α^{47}	100111
α^5	100000	α^{27}	001110	α^{48}	001101
α^6	000011	α^{28}	011100	α^{49}	011010
α^7	000110	α^{29}	111000	α^{50}	110100
α^8	001100	α^{30}	110011	α^{51}	101011
α^9	011000	α^{31}	100101	α^{52}	010101
α^{10}	110000	α^{32}	001001	α^{53}	101010
α^{11}	100011	α^{33}	010010	α^{54}	010111
α^{12}	000101	α^{34}	100100	α^{55}	101110
α^{13}	001010	α^{35}	001011	α^{56}	011111
α^{14}	010100	α^{36}	010110	α^{57}	111110
α^{15}	101000	α^{37}	101100	α^{58}	111111
α^{16}	010011	α^{38}	011011	α^{59}	111101
α^{17}	100110	α^{39}	110110	α^{60}	111001
α^{18}	001111	α^{40}	101111	α^{61}	110001
α^{19}	011110	α^{41}	011101	α^{62}	100001
α^{20}	111100	α^{42}	111010	α^{63}	000001
α^{21}	111011				

Tabla 9.8: Elementos del $GF(2^6)$

La rutina *GenerarElementos* guarda en un arreglo global de enteros denominado *A*, los elementos generados. A continuación se muestra su código.

```
Sub GenerarElementos ()
' Este procedimiento genera los elementos del  $GF(2^6)$ .

Dim i As Integer ' Contador
```


Módulo de Osciloscopio

```
A(1) = 1
For i = 2 To (2 ^ 6)
    A(i) = A(i - 1) * 2
    If A(i) >= (2 ^ 6) Then
        A(i) = (A(i) - (2 ^ 6)) Xor 3
    End If
Next i
End Sub
```

El elemento uno, que es α^0 se almacena en $A(1)$, el elemento cero no es almacenado, pero se representará con α^{-1} . Para generar el resto de los elementos se hace un ciclo con un *For*, en donde al elemento se le asigna el corrimiento a la izquierda del elemento anterior. Nótese que los elementos se almacenan con el equivalente decimal del número binario que lo representa. Para hacer un corrimiento a la izquierda se multiplica al elemento anterior por 2. Si el elemento es mayor que 2^6 significa que ocurrió un acarreo, así que se elimina el acarreo al elemento, restándole 2^6 y se hace un *EXOR* con el decimal del binario que representa al polinomio generador del campo, en este caso el número 3.

La función *Suma*, se utiliza frecuentemente dentro de *Decodificar*, y realiza la operación suma sobre el campo $GF(2^6)$; además lleva como parámetros los exponentes de los elementos a sumar, que son *sum1* y *sum2*. La primer sentencia *if* y la segunda verifican que los elementos se encuentren dentro del campo, si no es así, se les asigna su equivalente, que sí pertenece a él. La tercer sentencia *if* verifica si los elementos son 0; si así fuera entonces regresa un 0, de lo contrario continúa la evaluación con otra sentencia *if*, la cual si *sum1* es 0, entonces regresa a *sum2* o viceversa si *sum2* es 0, regresa a *sum1*. Si no se ha cumplido ninguno de los tres casos anteriores, hace la *EXOR* de los elementos, que como se recordará se encuentran en el arreglo *A*. Si el resultado de la operación *EXOR* es 0, regresa un -1, ya que la función

así como recibe exponentes regresa exponentes; pero si es diferente de 0, busca el resultado a lo largo del arreglo A, hasta que lo encuentra, para que la función regrese el valor del exponente correspondiente a ese resultado. El código de la función suma se muestra a continuación.

```
Function Suma (sum1 As Integer, sum2 As Integer) As Integer
' Esta función realiza la operación suma sobre GF(2^6).
Dim aux As Integer, aa As Integer
If sum1 > 63 Then ' Pone dentro del campo a sum1.
    sum1 = sum1 Mod 63
End If
If sum2 > 63 Then ' Pone dentro del campo a sum2.
    sum2 = sum2 Mod 63
End If
If (sum1 = -1) And (sum2 = -1) Then
    Suma = -1 ' Si los dos son ceros regresa un cero.
Else
    If (sum1 = -1) Then 'Si un sumando es 0 regresa el valor del
        Suma = sum2      ' otro.
    Else
        If (sum2 = -1) Then
            Suma = sum1
        Else
            aux = A(sum1 + 1) Xor A(sum2 + 1) ' Suma los sumandos.
            If aux = 0 Then
                Suma = -1 ' Si la suma es 0 el exponente es -1.
            Else
                For aa = 1 To 64 ' Busca el equivalente.
                    If aux = A(aa) Then
                        Suma = aa - 1 ' Asigna el exponente.
                        Exit For
                    End If
                Next aa
            End If
        End If
    End If
End If
End Function
```

El diagrama de flujo de la función *Decodificar* se muestran en la Figura 9.22.

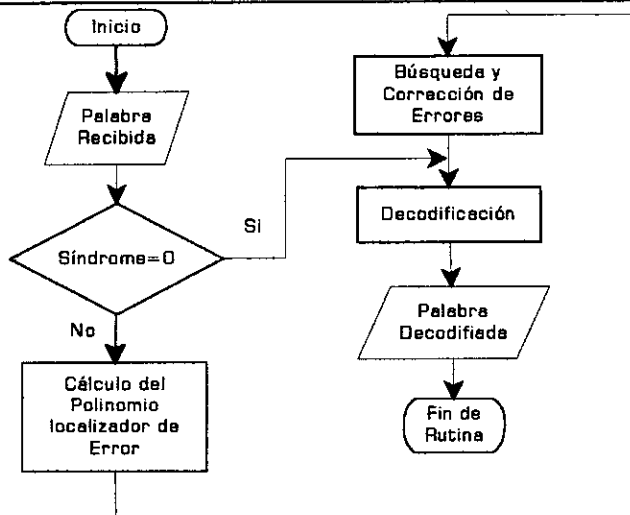


Figura 9.22: Diagrama de Flujo de la función Decodificar

La declaración de la función *Decodificar* y sus variables es la siguiente:

```

Function Decodificar(pa As String) As Long

Static R(30) As Integer ' Exponentes de los coeficientes de R(z).
Static M2(40) As Integer ' Exponentes de los coeficientes de M2(z).
Static M1(40) As Integer ' Exponentes de los coeficientes de M1(z).
Static U1(40) As Integer ' Exponentes de los coeficientes de U1(z).
Static U1_(40) As Integer ' Exponentes de los coeficientes de U1'(z).
Static U2(40) As Integer ' Exponentes de los coeficientes de U2(z).
Static Q(30) As Integer ' Exponentes de los coeficientes de Q(z).
Static S(22) As Integer ' Exponentes de los coeficientes de S(z).
' de los coeficientes de S(z).
Static recib(8) As Integer ' Arreglo donde se guardará en enteros
' los bytes de la palabra recibida.
Static gra_z As Integer ' Grado de la variable z.
Static val_z As Integer ' Valor de la evaluación de Ö(z).
Static Gen(2) As Integer ' Coeficientes de la
Static rres(8) As Integer ' Residuo parcial de la decodificación.
Dim lon_R As Integer ' Longitud de R(z)
Dim gra_R As Integer ' Grado de R(z).
Dim lon_M2 As Integer ' Longitud de M2(z).
  
```

```

Dim gra_M2 As Integer      ' Grado de M2(z).
Dim lon_M1 As Integer      ' Longitud de M1(z).
Dim gra_M1 As Integer      ' Grado de M1(z).
Dim lon_U1 As Integer      ' Longitud de U1(z).
Dim gra_U1 As Integer      ' Grado de U1(z).
Dim lon_U1_ As Integer     ' Longitud de U1'(z).
Dim gra_U1_ As Integer     ' Grado de U1'(z).
Dim lon_U2 As Integer      ' Longitud de U2(z).
Dim gra_U2 As Integer      ' Grado de U2(z).
Dim lon_Q As Integer       ' Longitud de Q(z).
Dim gra_Q As Integer       ' Grado de Q(z).
Dim n As Integer           ' Veces que se realiza el algoritmo Euclideano.
Dim d As Integer           ' Exponente del coef. de mayor grado de z de U1(z).
Dim sindro As Integer      ' Variable que indica si todos los
                          ' síndromes fueron cero.
Dim num_err As Integer     ' Número de errores detectados.
Dim word_corr As Integer   ' Byte que servirá para recorrer los bits
                          ' de los bytes que contienen la palabra recibí().
Dim dec As Long            ' Valor de la palabra decodificada.

```

donde *pa* es el parámetro que contiene la palabra recibida, la cual es de tipo string de 8 bytes.

Para facilitar los cálculos se convierte la cadena a ocho enteros por medio de las siguientes instrucciones:

```

' Convierte la palabra recibida que es un string de longitud 8
' a 8 enteros y los guarda en el arreglo recibí().
For i = 0 To 7
    recibí(i) = Asc(Mid$(pa, i + 1, 1))
Next i

```

Los exponentes de los síndromes son calculados con la siguiente rutina.

```

' Cálculo de los exponentes de los coeficientes del polinomio
' síndrome s(z).

'Cálculo del polinomio síndrome. Se evalúa el polinomio asociado
' a la palabra recibida para las raíces de G(x).
word_corr = 128 ' Le pone un uno al bit más significativo y al
For j = 1 To 22 ' resto cero.
    s(j) = -1 ' Inicializa la evaluación en cero.

```

Módulo de Osciloscopio

```
If recibi(0) And word_corr Then ' Pregunta si el termino
    s(j) = 0 ' independiente es uno.
End If
word_corr = word_corr \ 2 'Hace n corrimiento hacia la derecha.
For i = 0 To 7 ' Evaluación de los ocho bytes.
    For k = 0 To 7 ' Evaluación de cada bit.
        If Not ((i = 0 And k = 0) Or (i = 7 And k = 7)) Then
            If recibi(i) And word_corr Then ' Pregunta si el
                ' bit i*8+k es un uno.
                s(j) = Suma(s(j), (j * ((i * 8) + k))) 'suma el ele.
            End If
            word_corr = word_corr \ 2 ' Hace un corri. a la dere.
        End If
    Next k
    word_corr = 128 ' Hace uno el bit más significativo
    Next i ' y lo demás cero.
    word_corr = 128
    If s(j) <> -1 Then ' Si el síndrome calculado es dif. de cero.
        sindro = sindro + 1
    End If
Next j
```

Esta rutina consiste en evaluar cada raíz del polinomio generador en el polinomio asociado a la palabra recibida. La palabra recibida convertida a enteros, se encuentra en el arreglo *recibi*. Para evaluar el polinomio de la palabra recibida, debe evaluarse bit a bit cada uno de los bytes (que conforman la palabra recibida que se encuentran en el arreglo *recibi*), utilizando el byte testigo *word_corr* para ello. A este byte testigo se le aplican corrimientos para después hacerle una AND con el byte correspondiente. Si el resultado de la AND es un 1, la raíz del polinomio generador que se está evaluando, se eleva al exponente de la variable del polinomio. No hay que olvidar que sólo se manejan los exponentes de los elementos. El exponente de cada raíz del polinomio generador, se representa por j y los exponentes de la variable del polinomio de la palabra recibida son representados por $i * 8 + k$, la variable i indica el byte del vector recibido que se está analizando, y la variable k indica cada bit de ese byte. En el arreglo de enteros s , se almacenan los exponentes de los resultado de las

evaluaciones de cada raíz. Estos resultados equivalen a los exponentes de los coeficientes del polinomio $s(z)$, el cual es necesario para el algoritmo Euclideo.

Si la variable *sindro* es diferente de 0, significa que existen errores y es necesario aplicar el algoritmo euclideo para detectar en donde se encuentran y poder corregirlos.

De los polinomios utilizados sólo se guardan los exponentes de los coeficientes, su longitud y su grado, es decir si la longitud de M_2 es 3, su grado 2 y sus exponentes 2, -1 y 0, el polinomio que se está representando es: $M_2(z) = \alpha^2 z^2 + 1$ (recuerde que el exponente negativo representa un 0). Los exponentes de los polinomios se guardan en los arreglos que llevan su nombre, es decir los exponentes de los coeficientes de $M_2(z)$ se guardan en el arreglo $M2$. La longitud y grado del polinomio se almacenan en las variables *lon_<polinomio>* y *gra_<polinomio>* respectivamente, por ejemplo, para el caso anterior, la longitud de $M2$ se almacena en la variable *lon_M2* y el grado en *gra_M2*.

Con las siguientes sentencias se asignan los valores iniciales a los polinomios $M_1(z)$, $U_1(z)$, $M_2(z)$, $U_2(z)$ y $U_1'(z)$, además se inicializa la variable *n*, que funge como contador para informar cuantas veces se efectúa el ciclo del algoritmo.

```
If sindro <> 0 Then ' Si alguno de los síndromes fue <> 0
                    ' efectúa el algoritmo Euclideo.
    ' Hace la asignación  $M2(2)=S(z)$ 
    lon_M2 = 22
    gra_M2 = 21
    f = 1
    ' Descarta los coeficientes de las variables de mayor orden que
    ' sean igual a 0 hasta que encuentra uno diferente de 0.
    Do While (lon_M2 > 1 And s(23 - f) = -1)
```

Módulo de Osciloscopio

```
    gra_M2 = gra_M2 - 1
    lon_M2 = lon_M2 - 1
    f = f + 1
Loop
For i = f To 22
    M2(i - f + 1) = s(22 - i + 1)
Next i

' Hace la asignación M1(z)=z^2t=z^22.
lon_M1 = 1
gra_M1 = 22
M1(1) = 0

' Asigna U1(z)=0.
U1(1) = -1
lon_U1 = 1
gra_U1 = 0

' Asigna U1'(z)=1.
U1_(1) = 0
lon_U1_ = 1
gra_U1_ = 0

' Asigna U2(z)=0.
U2(1) = -1
lon_U2 = 1
gra_U2 = 0

' Veces que se ejecuta el ciclo
n = 1
```

El ciclo del algoritmo Euclideano se efectúa hasta que el grado del polinomio $R(z)$ es menor o igual al número máximo de errores a corregir menos uno. Para nuestro caso particular el algoritmo se ejecuta hasta que el grado de $R(z)$ es menor o igual a 10.

Las siguientes líneas de código efectúan la operación $M_1(z) \div M_2(z)$, asignándole el cociente a $Q(z)$ y el residuo a $R(z)$. Cuando el grado de $M_1(z)$ es menor al grado de $M_2(z)$, se le asigna a $Q(z)$ un 0 y a $R(z)$ el polinomio $M_1(z)$, terminando así

el proceso de la división. De lo contrario primero se verifica que el polinomio $M_2(z)$ no sea 0.

```
' División de los polinomios. Q(Z)=[M1(z)/M2(z)]
If gra_M1 < gra_M2 Then
    Q(1) = -1
    lon_Q = 1
    gra_Q = 0
    For i = 1 To lon_M1
        R(i) = M1(i)
    Next i
    lon_R = lon_M1
    gra_R = gra_M1
Else
    If lon_M2 > 1 And M2(1) > -1 Then
```

Si la longitud de $M_1(z)$ es menor o igual a la longitud de $M_2(z)$, a la longitud de $M_1(z)$ se le incrementa la diferencia de grados de los dos polinomios y a los nuevos elementos les asigna un 0. Por ejemplo, si el grado de $M_1(z)$ es 4, su longitud 2 y los exponentes de sus coeficientes son 3 y 2, entonces el polinomio representado es: $M_1(z) = \alpha^3 x^4 + \alpha^2 x^3$; y si tenemos que el grado de $M_2(z)$ es 2, su longitud 3 y los exponentes de sus coeficientes 0, 0 y 1 entonces el polinomio representado es $M_2(z) = z^2 + z + \alpha$; así que lo hace esta rutina es asignarle a $M_1(z)$ el polinomio: $\alpha^3 x^4 + \alpha^2 x^3 + 0x^2 + 0x + 0$.

```
If lon_M1 <= lon_M2 Then
    For ii = lon_M1 + 1 To lon_M2 + (gra_M1 - gra_M2)
        M1(ii) = -1
    Next ii
    lon_M1 = lon_M2 + (gra_M1 - gra_M2)
End If
```

Posteriormente se les asignan al grado y la longitud de $R(z)$, el grado y la longitud de $M_1(z)$ respectivamente, puesto que el residuo inicialmente es $M_1(z)$.

```
gra_R = gra_M1
lon_R = lon_M1
```


Por medio del apuntador i , se verifican los elementos de $M_1(z)$ y se le asignan valores a los elementos de $Q(z)$. La división se efectúa en la forma tradicional, por ejemplo:

$$Q(z) = \frac{z + 1}{\frac{\alpha z^2 + \alpha^6}{\alpha z^2 + \alpha z}} = \frac{\alpha^6 z + 1}{\alpha^6 z + \alpha^6}$$

$$R(z) = \frac{\alpha^6 z + \alpha^6}{\alpha}$$

Como el programa sólo almacena los exponentes de los coeficientes, esto se representaría de la siguiente manera (en este ejemplo los elementos de los vectores se encuentran separados por comas):

$$Q(z) = \frac{1,6}{0,0 \mid \frac{1,0,0}{1,1}} = \frac{6,1}{6,6}$$

$$R(z) = \frac{1}{1}$$

Lo mostrado arriba se efectúa con el siguiente código, el cual se ejecuta mientras que el grado de $R(z)$ es mayor o igual al grado de $M_2(z)$, es decir cuando ya no es posible dividir el residuo entre $M_2(z)$. La primer sentencia *if* calcula el cociente, mientras que la segunda sentencia *if* verifica que el exponente no sea negativo y en caso de serlo le asigna su equivalente positivo. El ciclo *for* se utiliza para calcular los exponentes de los coeficientes del polinomio residuo, los cuales se almacenan en el arreglo $M1$.

```
i = 1
Do While ((gra_R >= gra_M2) And (lon_R > 0))
  If M1(i) = -1 Then
    Q(i) = -1
  Else
    Q(i) = M1(i) - M2(1)
    If Q(i) < 0 Then
```

```

        Q(i) = (Q(i) Mod 63) + 63
    End If
End If
For j = 2 To lon_M2
    If (Q(i) <> -1) And (M2(j) <> -1) Then
        M1(i + j - 1) = Suma(M1(i + j - 1), Q(i) +
                                M2(j))
    End If
Next j
gra_R = gra_R - 1
lon_R = lon_R - 1
i = i + 1
Loop
lon_Q = i - 1
gra_Q = gra_M1 - gra_M2

```

Con el siguiente código se eliminan los coeficientes ceros de las variables de mayor orden hasta donde el coeficiente sean diferente de cero, es decir, si los tres coeficientes de las variables de mayor orden de algún polinomio son 0, se eliminan esos tres elementos. Posteriormente se le asigna el residuo que se encuentra en $M1(z)$ a $R(z)$.

```

f = 1
Do While ((M1(f + lon_Q) = -1) And (gra_R > 0))
    gra_R = gra_R - 1
    f = f + 1
Loop
For ii = f + lon_Q To lon_M1
    R(ii - (f + lon_Q) + 1) = M1(ii)
Next ii
lon_R = lon_R - f + 1

```

Si $M_2(z)$ es 0, a $Q(z)$ le asigna 1 y a $R(z)$ le asigna $M_1(z)$.

```

Else
    lon_Q = 1
    gra_Q = 0
    Q(1) = 0
    For i = 1 To lon_M1
        R(i) = M1(i)
    Next i
    lon_R = lon_M1
    gra_R = gra_M1
End If
End If
' Fin de la división.

```

A continuación se hace la operación $Q(z)*U'(z)$ y el resultado se almacena en $U_1(z)$. Este resultado es su solución parcial, ya que después se le suma $U_2(z)$, siguiendo el algoritmo Euclideo. La multiplicación se efectúa de una forma muy sencilla, primero se calcula la longitud y el grado del polinomio resultante, después se inicializa $U_1(z)$ con 0, se calculan los productos parciales y se suman a los obtenidos anteriormente hasta que se obtiene el producto total.

```
' Multiplicación de los polinomios Q(z)*U1'(z).
lon_U1 = lon_Q + lon_U1 - 1
gra_U1 = gra_Q + gra_U1
For kk = 1 To lon_U1
    U1(kk) = -1
Next kk
For i = 1 To lon_U1
    If {U1(i) <> -1} Then
        For j = 1 To lon_Q
            If {Q(j) <> -1} Then
                U1(i + j - 1) = Suma(U1(j + i - 1), Q(j) +
U1(i))
            End If
        Next j
    End If
Next i
' Fin de la multiplicación.
```

A continuación al polinomio $U_1(z)$ que vale $Q(z)*U_1'(z)$, se le suma el polinomio $U_2(z)$. Si el grado de $U_1(z)$ es mayor al grado de $U_2(z)$, entonces primero se ajustan las longitudes de tal forma que sean iguales, asignándoles 0 a los nuevos elementos de ellos; por último se efectúa la suma. Para el caso contrario, en donde el grado de $U_2(z)$ es mayor al grado de $U_1(z)$ se hace la mismo pero ahora el resultado se obtiene en $U_2(z)$, y después de la suma a $U_1(z)$ se le asigna lo que contiene $U_2(z)$.

```
' Suma de los polinomios. U1(z) = Q(z)*U1'(z)+U2(z).
If gra_U1 > gra_U2 Then
    If lon_U1 < lon_U2 Then
        For i = lon_U1 + 1 To lon_U2
            U1(i) = -1
        Next i
        lon_U1 = lon_U2
```

```

End If
For i = lon_U1 + 1 To lon_U2 + (gra_U1 - gra_U2)
    U1(i) = -1
Next i
For i = 1 To lon_U2
    U1(i + (gra_U1 - gra_U2)) = Suma(U1(i + (gra_U1 -
gra_U2)), U2(i))
Next i
Else
If lon_U2 < lon_U1 Then
    For i = lon_U2 + 1 To lon_U1
        U2(i) = -1
    Next i
    lon_U2 = lon_U1
End If
For i = lon_U2 + 1 To lon_U1 + (gra_U1 - gra_U2)
    U2(i) = -1
Next i
For i = 1 To lon_U1
    U2(i + (gra_U2 - gra_U1)) = Suma(U2(i + (gra_U2 -
gra_U1)), U1(i))
Next i
For i = 1 To lon_U2
    U1(i) = U2(i)
Next i
lon_U1 = lon_U2
gra_U1 = gra_U2
End If
' Fin de la suma de polinomios.

```

Siguiendo con el algoritmo Euclideo, si el grado de $R(z)$ es mayor a 1, se hacen las asignaciones $U_2(z) = U_1(z)$, $M_1(z) = M_2(z)$, $M_2(z) = R(z)$ y $U_1'(z) = U_1(z)$, terminando así el algoritmo Euclideo.

```

If gra_R > 1 Then
    ' Hace U2(z)=U1'(z).
    For i = 1 To lon_U1_
        U2(i) = U1_(i)
    Next i
    lon_U2 = lon_U1_
    gra_U2 = gra_U1_

    ' Hace M1(z)=M2(z).
    For i = 1 To lon_M2
        M1(i) = M2(i)
    Next i

```

Módulo de Osciloscopio

```
lon_M1 = lon_M2
gra_M1 = gra_M2

' Hace M2(z)=R(z).
For i = 1 To lon_R
    M2(i) = R(i)
Next i
lon_M2 = lon_R
gra_M2 = gra_R

' Hace U1'(z)=U1(z).
For i = 1 To lon_U1
    U1_(i) = U1(i)
Next i
lon_U1_ = lon_U1
gra_U1_ = gra_U1

n = n + 1
End If
```

El polinomio localizador del error es $U_1(z)/\delta$, donde δ representa el coeficiente de mayor grado de $U_1(z)$. El exponente del coeficiente de mayor orden, está almacenado en $U_1(1)$. Con el siguiente ciclo *For* se hace la operación $U_1(z)/\delta$.

```
' Le asigna a d el coeficiente de la variable de mayor orden
' de U1.
If U1(1) <> -1 Then
    d = U1(1)
Else
    d = 0
End If

'Hace U1(z)=U1(z)/d=sigma(z).
For i = 1 To lon_U1
    If U1(i) <> -1 Then
        U1(i) = (U1(i) - d) Mod 63
        If U1(i) < 0 Then
            U1(i) = U1(i) + 63
        End If
    End If
End If
Next i
```

Posteriormente se hace la búsqueda de Chien y se corrigen los errores detectados, obteniendo así la palabra corregida.

```

' Se hace la búsqueda de Chien para detectar los errores.
For i = 0 To 62
    val_z = -1
    gra_z = gra_U1
    For j = 1 To lon_U1 ' Evalúa el elemento alfa^(-i) en U1(z).
        If gra_z > 0 Then
            temp = ((-i * gra_z) Mod 63)
            If temp < 0 Then
                temp = temp + 63
            End If
            If U1(j) <> -1 Then
                val_z = Suma(val_z, temp + U1(j)) ' U1(alfa^-i).
            End If
        Else
            If U1(j) <> -1 Then ' Caso del termino independiente.
                val_z = Suma(val_z, U1(j))
            End If
        End If
        gra_z = gra_z - 1
    Next j
    If val_z = -1 Then ' Si es cero hay un error en posición i.
        ' Hace la corrección del error.
        recibi(i \ 8) = recibi(i \ 8) Xor 2 ^ (7 - (i Mod 8))
        num_err = num_err + 1 ' Número de errores encontrados.
    End If
Next i
End If

```

El último paso es hacer la decodificación, la cual consiste en dividir el polinomio de la palabra recibida entre el polinomio generador. Este es el proceso inverso de la codificación, por lo tanto el proceso es una división binaria.

```

' Decodificación
If (Not (sindro <> 0)) Or ((sindro <> 0) And (num_err <> 0)) Then
    word_corr = 128
    Gen(0) = 212: Gen(1) = 220 ' Coeficientes de la fun. generadora.
    For i = 0 To 1
        rres(i) = recibi(i)
    Next i
    k = 0
    dec = 0
    For i = 0 To 15
        If rres(k) And word_corr Then
            dec = dec + (2 ^ (15 - i)) ' Información decodificada.
            For j = 0 To 1
                rres(j) = rres(j) Xor Gen(j) ' Residuo
            Next j
        End If
        k = k + 1
    Next i
End If

```

Módulo de Osciloscopio

```
End If
word_corr = word_corr / 2 ' Corrimiento a la derecha.
If i = 7 Then
    k = k + 1 ' Pasa al siguientes byte de Gen.
    word_corr = 128
End If

' Corrimiento a la derecha de la función generadora.
Gen(1) = Gen(1) \ 2
If (Gen(0) Mod 2) Then
    Gen(1) = Gen(1) Xor (128)
End If
Gen(0) = Gen(0) \ 2
Next i
Else
    dec = -88 ' Indica si el error no se pudo corregir.
End If
Decodificar = dec ' Regresa la palabra decodificada.
```

9.3.3 Control del Programa "Módulo de Osciloscopio"

El programa "Módulo de Osciloscopio" se puede controlar remotamente a través de los 3 botones de la punta inalámbrica y con los cuales se pueden hacer hasta 7 posibles combinaciones. Hay que recordar que normalmente serían 8 combinaciones, pero una es la que siempre está presente cuando no se ha presionado algún botón. También es importante hacer ver que la información de la combinación de los botones que se presionaron está contenida en los bits 0 (TAb o Tecla-Abajo), 1 (TAr o Tecla-Arriba) y 2 (TFu o Tecla-Función) del Byte de Datos. Cada uno de estos 3 bits contienen el estado eléctrico de cada uno de los tres botones. Cabe recordar también que un botón tiene un estado verificado alto mientras no se presione, así por ejemplo si se presionan simultáneamente las teclas TAr y TAb la combinación lógica que forman los bits 0, 1 y 2 del Byte de Datos es 100b, pero si se presiona sólo TFu la combinación que forman esos tres bits es 011b. La función que realiza alguna combinación de esos tres botones en el programa "Módulo de

Osciloscopio”, dependerá de la Ventana que se encuentre activa y de las funciones que se puedan realizar para dicha ventana.

El programa “Módulo de Osciloscopio” no sólo se puede controlar remotamente, sino que funciona como cualquier otra aplicación de Microsoft Windows, donde hay un Menú y algunos controles activador por el teclado normal o un Mouse.

9.3.3.1 Programa “Módulo de Osciloscopio”

El programa “Módulo de Osciloscopio” se diseñó para que tuviera los controles básicos de un Osciloscopio: El control de la Escala de Voltaje, el control de la Escala de Tiempo, el control de Disparo de dos señales capturadas, los controles para mover las líneas de referencia de los canales y del tiempo, los controles para ver el tipo de señal eléctrica que se captura (señal AC o DC) y por supuesto la grátícula, que es donde se desplegarán las señales eléctricas capturadas. También se pensó en tener funciones comunes de un programa de computadora, como son el abrir o guardar un archivo donde estén contenidos los datos de una señal eléctrica capturada con la punta inalámbrica. Además cuenta con la opción para imprimir una señal eléctrica capturada incluyendo la información característica (Voltaje, Voltaje RMS, Frecuencia, Periodo, etc.) de la señal.

Los controles del programa se encuentran en 7 Ventanas (las cuales se controlan internamente a través de la variable *Vent*): Ventana de Canal A (*Vent* = 1), Ventana de Canal B (*Vent* = 2), Ventana de Tiempo (*Vent* = 3), Ventana de Trigger (*Vent* = 4), Ventana Pantalla (*Vent* = 5), Ventana de Comandos (*Vent* = 6) y Ventana

Módulo de Osciloscopio

de Información ($V_{ent} = 0$). La Ventana de Información no se activa directamente, sino a través de los botones de Información (I) contenidos en las subventanas A, B y T de la Ventana Pantalla; además tiene dos presentaciones: una cuando se activa con los botones de información (I) de las subventanas A y B, y la otra cuando se activa presionando el botón de Información (I) de la subventanas T. En la Figura 9.23 se muestra la forma principal del programa “Módulo de Osciloscopio” con las 6 Ventanas Principales (Canal A, Canal B, Tiempo, Trigger, Pantalla y Comandos) y sus respectivos controles.

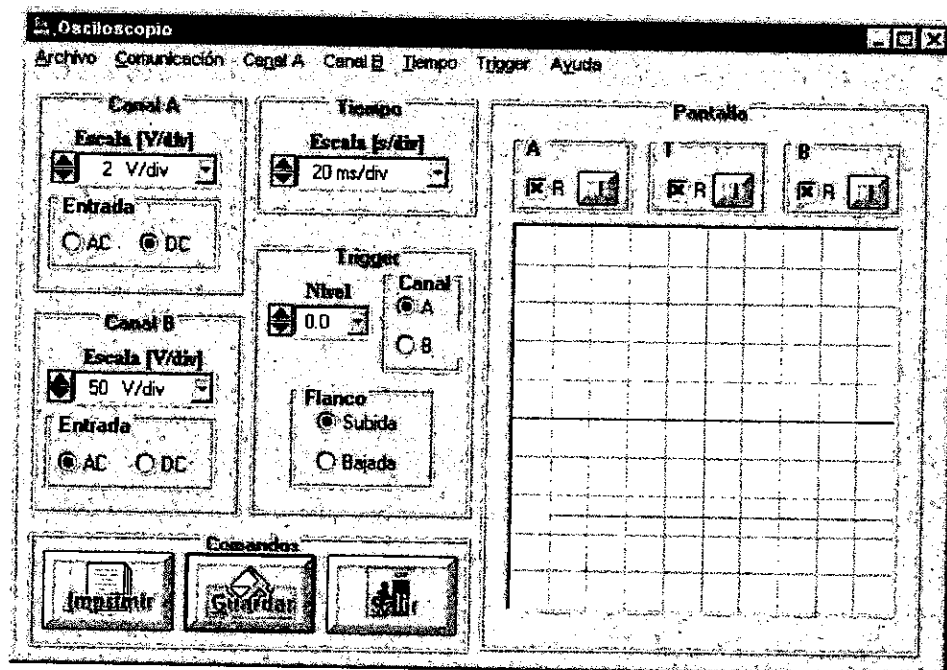


Figura 9.23: Forma Principal del Programa “Módulo de Osciloscopio”

Las Ventanas de Canal A y Canal B, cuentan con:

- ♦ Escala de Voltaje: Este control actúa como un Zoom vertical de la señal eléctrica capturada, donde se atenúa o se amplifica el voltaje de la gráfica.
- ♦ Entrada: Este control indica si la señal graficada corresponde a una señal AC o a una señal DC.

La Ventana de Tiempo, cuenta sólo con:

- ♦ Escala de Tiempo: Este control actúa como un Zoom horizontal donde se prolonga o acorta el tiempo de la gráfica de la señal.

La Ventana de Trigger, cuenta con:

- ♦ Nivel: Este control sirve para que una señal capturada AC se empiece a graficar en cierta división de la grátícula de la Ventana Pantalla con respecto al eje horizontal central (nivel 0.0), es decir el nivel de disparo tiene valores desde la posición 5.0 hasta la posición -5.0, correspondientes a la división +5 y la división -5 de la grátícula.
- ♦ Canal de Disparo: Este control sirve para indicar sobre cuál de las dos señales (si es que se tienen dos señales graficadas) se hace el disparo.
- ♦ Flanco de Disparo: Con este control se indica si el disparo es sobre el flanco de subida o el de bajada de una señal AC graficada.

La Ventana de Pantalla cuenta con:

- ◆ **Controles de Referencia (R):** Este control sirve para hacer visibles o no las líneas de referencias del Canal A, Canal B o Tiempo, según sea el caso.
- ◆ **Botones de Información (I):** Estos botones activan la Ventana de Información para ver la información de las características eléctricas y del Tiempo, de las señales graficadas.
- ◆ **Líneas de Referencia:** Las líneas de Referencia del Canal A y Canal B, desplazan verticalmente las señales eléctricas graficadas por el Canal A y por el Canal B (según sea el caso) y la línea de referencia del Tiempo desplaza horizontalmente ambas señales. Estos desplazamientos se pueden hacer sólo si hay alguna señal ya capturada y si están visibles las líneas de referencia.

La Ventana de Comandos, cuenta con:

- ◆ **Botón de Imprimir:** Con este control se imprime la gráfica y valores eléctricos de alguna señal eléctrica ya capturada y graficada.
- ◆ **Botón de Guardar:** Con este control se almacenan las muestras de una señal eléctrica ya capturada.
- ◆ **Botón de Salir:** Con este control se sale de programa.

La Ventana de Información sólo cuenta con

- ◆ **Botón de Regresar:** sirve para regresar a la Ventana Pantalla.

9.3.3.2 Control Remoto del Programa

Cada vez que se controla remotamente el programa, se tienen que revisar los estados de los bits 0 (TAb), 1 (TAr) y 2 (TFu) del Byte de Datos ya decodificado. Como se mencionó anteriormente cuando alguno de esos 3 botones es presionado su respectivo bit registra un 0 lógico. Para revisar los estados lógicos de esos bits, se hace con las siguientes instrucciones:

```
Dato = (Dato And &H700) \ &H100 ' Separa el Byte que contiene la
                                ' combinación de Teclas.

Select Case Dato
  Case &H0: Tecl_Fu_Ar_Ab ' Teclas Función + Arriba + Abajo.
  Case &H1: Tecl_Fu_Ar   ' Teclas Función + Arriba.
  Case &H2: Tecl_Fu_Ab   ' Teclas Función + Abajo.
  Case &H3: Tecl_Fu      ' Tecla Función.
  Case &H4: Tecl_Ar_Ab   ' Teclas Arriba + Abajo.
  Case &H5: Tecl_Ar      ' Tecla Arriba.
  Case &H6: Tecl_Ab      ' Tecla Abajo.
End Select
```

donde el Byte de Datos está representado por la variable Dato, descartando primero el Byte de Información y después los bits 7, 6, 5, 4 y 3 del Byte de Datos. Posteriormente se hace la identificación de cada una de las 7 combinaciones que se pueden hacer con las 3 teclas de la Punta Inalámbrica. La única combinación que sólo tiene una función común en todas las Ventanas del Programa, es la que se registra cuando se presionan las tres teclas al mismo tiempo, con lo cual se indica que se va a hacer una captura y transmisión de una señal eléctrica con la Punta Inalámbrica.

El programa maneja dos formas de ver las Ventanas: Maximizada (ideal cuando se usa la Punta Inalámbrica) o Minimizada (ideal cuando se está analizando en la PC, una señal eléctrica ya capturada). Internamente estas dos formas de ver las ventanas se controlan con la variable *Gigante*; cuando *Gigante = True*, las

ventanas se ven maximizadas, y cuando *Gigante = False*, las ventanas se ven minimizadas. Los controles de las ventanas pueden cambiar sus valores a través de la Punta Inalámbrica, sólo cuando dichas ventanas están maximizadas. Al iniciarse el programa, todas las ventanas están minimizadas. En la Tabla 9.9 se muestran cada una de las funciones que se realizan cuando todas las ventanas están minimizadas y se usa el teclado de la Punta inalámbrica.

Teclas	Función
F + Ar + Ab	Maximiza la Ventana de Pantalla para graficar la señal eléctrica recientemente capturada..
F + Ar	Cambia el focus ciclicamente por cada una de las 6 Ventanas activas y minimizadas en el siguiente orden: Canal A, Canal B (para estas dos Ventanas el Focus se pone en el <i>ComboBox</i> de Escala de Voltaje), Tiempo (Focus en el <i>ComboBox</i> de Escala de Tiempo), Trigger (Focus en el <i>ComboBox</i> de Nivel de Disparo), Pantalla (Botón de Información de Canal A) y Comandos (Botón de Imprimir).
F + Ab	Cambia el focus ciclicamente por cada una de las 6 Ventanas activas y minimizadas en el siguiente orden: Comandos (Botón de Imprimir), Pantalla (Botón de Información de C. A), Trigger (Focus en el <i>ComboBox</i> de Nivel de Disparo), Tiempo (Focus en el <i>ComboBox</i> de Escala de Tiempo), Canal B y Canal A (para estas dos Ventanas el Focus se pone en el <i>ComboBox</i> de Escala de Voltaje).
F	Sin función.
Ar	Maximiza la Ventana que tiene el Focus en ese momento. El Focus estará para las ventanas de Canal A, Canal B en el <i>ComboBox</i> de Escala de Voltaje, para la Ventana de Tiempo en el <i>ComboBox</i> de Escala de Tiempo, para la Ventana de Trigger en el <i>ComboBox</i> del Nivel de Disparo, para la Ventana de Pantalla en el Botón de Información de C. A y para la Ventana de Comandos en el botón de Imprimir.
Ab	Maximiza la Ventana que tiene el Focus en ese momento. El Focus estará para las ventanas de Canal A, Canal B en el <i>ComboBox</i> de Escala de Voltaje, para la Ventana de Tiempo en el <i>ComboBox</i> de Escala de Tiempo, para la Ventana de Trigger en el <i>ComboBox</i> del Nivel de Disparo, para la Ventana de Pantalla en el Botón de Información de C. A y para la Ventana de Comandos en el botón de Imprimir.
Ar + Ab	Sin función.

Tabla 9.9: Funciones Realizables cuando Todas las Ventanas están Minimizadas

Cuando se captura una señal eléctrica con la punta inalámbrica (presionando Fu, Ar y Ab) en la PC se realiza el proceso de graficación de una nueva señal, maximizando la Ventana Pantalla, aunque haya otra Ventana ya maximizada.

Esta función es la de más alta prioridad y funciona de la misma forma para todas las ventanas. A continuación se muestra el código que se ejecuta cuando se presionan las teclas Fu + Ar + Ab de la Punta Inalámbrica.

```

If (Vent <> 5) Or (Gigante = False) Then ' Si no está activa Vent.
' de Pantalla o bien que estén Minimizadas todas las ventanas.
Gigante = True ' Maximizar Ventanas.
Vent = 5 ' Ventana de Pantalla a Maximizar.
Va = 5 ' Ventana de Pantalla anteriormente maximizada.
frmVentana(0).Visible = False ' Desaparece Ventana de
' Información por si previamente se veía.
AgrandaForma True ' Maximiza forma del Programa.
ReDefineVentana Gigante, Vent ' Maximiza Ventana de Pantalla.
RedefineRejillas frmOsciloscopio ' Dibuja Gráticula y Gráficas
' en la Ventana Pantalla.
End If
frmOsciloscopio.Refresh ' Refresca objetos de la Forma del Progr.
MuestrasGrafi = 0 ' Ninguna muestra Graficada.
SR.TotalMuest = -1 ' Ninguna muestras recibida remotamente.
Pantalla.Cls ' Limpia cualquier señal previamente dibujada.

```

Primeramente, se determina si se trata de alguna de las Ventana principales distinta a la Ventana Pantalla o bien que sea alguna de las 6 ventanas principales que esté minimizada, para cuatro cosas: maximizar la Ventana Pantalla y asignarla como ventana activa; asignar a *Va* (Ventana Anterior) que la última ventana maximizada es la Ventana Pantalla; amplía la forma del programa “Módulo de Osciloscopio”, para hacer una buena distribución de la gráticula y los controles de la Ventana Pantalla; y dibuja con dimensiones más grandes tanto las rejillas de la gráticula como los controles de la Ventana Pantalla. Por último ya hecha la verificación se procede a limpiar la gráticula (denominado internamente como *Pantalla*, el cual es un objeto *PictureBox* de Visual Basic) e inicializa las variables que se utilizan al graficar las muestras de una señal eléctrica transmitida.

A continuación se muestra el código que se ejecuta en la PC, cuando se presionan simultáneamente en la Punta Inalámbrica las teclas Fu + Ar.

```
If Gigante = False Then ' Si las Ventanas están Minimizadas.  
Vent = Vent + 1 ' Incrementa contador Ventanas.  
Select Case Vent ' Selección de Ventana a tratar minimizada.  
Case 1 To 4 ' Ventanas de Can. A, Can. B, Tiempo o Trigger.  
cmbEscala(Vent).SetFocus ' Pone Focus en algún ComboBox.  
Case 5 ' Ventana de Pantalla.  
btnInfo(1).SetFocus ' Focus en Botón de Información de A.  
Case 6 ' Ventana de Comandos.  
btnComando(1).SetFocus ' Focus en Botón de Imprimir.  
Case 7 ' Si incremento fue 7.  
Vent = 1 ' Asigna Ventana a tratar la de Canal A.  
cmbEscala(1).SetFocus ' Pone Focus en ComboBox de Vent. C. A.  
End Select  
End If
```

Inicialmente, se identifica si están minimizadas las 6 Ventanas principales. Si es así, incrementa la variable *Vent* (la cual representa cada una de las 7 ventanas). A través de la sentencia *Select-Case*, se asigna el Focus a un control característico de la Ventana apuntada por *Vent*. Hay que notar, que si el incremento de *Vent* es 7, se hace *Vent* = 1 y se asigna el Focus al control de la Escala de Voltaje de la Ventana de Canal A. El código que se emplea cuando se presionan simultáneamente las teclas Fu + Ab es prácticamente el mismo, sólo que en vez de incrementar en 1 la variable *Vent*, se decrementa 1.

Las siguientes líneas de código se ejecutan, cuando se presiona en la Punta Inalámbrica, la tecla Ar o bien la tecla Ab.

```
Func = 0 ' Sin control.  
Gigante = True ' Modo Maximizado.  
ReDefineVentana Gigante, Vent ' Se maximiza ventana requerida.  
Select Case Vent ' Selección de Ventana a maximizar.  
Case 1 To 4 ' Vent. a ver son C. A, C. B, Tiempo o Trigger.  
cmbEscala(Vent).SetFocus ' Pone Focus en algún ComboBox.  
Case 5 ' Ventana a Ver es Pantalla.  
AgregaForma True ' Maximiza Forma del Progr.
```

```

RedefineRejillas frmOsciloscopio ' Dibuja Graticula y
                                ' Gráficas en graticula de la Ventana Pantalla.
btnInfo(1).SetFocus ' Focus en botón de Información de A.
Case 6 ' Si ventana a ver es Comandos.
    btnComando(1).SetFocus ' Pone Focus en Botón de Imprimir.
End Select
Va = Vent ' Ventana Maximizada a dejar de ver = Ventana actual.

```

Primero se indica a través de la variable *Func*, que ningún control de ventana está activo. Después se indica que la ventana activa estará maximizada, la cual a través del procedimiento *RedefineVentana*, se cambian las dimensiones de la Ventana y sus controles, para que se vean más grandes, ocupando casi por completo toda la pantalla del monitor. Posteriormente se asegura que tenga el Focus un control característico de la ventana activa. Por último a través de la variable *Va*, se registra cual es la ventana activa y maximizada, para futuros procesos.

En la Figura 9.24 se muestra la Ventana de Canal A ya maximizada, con el Cuadro de Selección Cyan en el control de la Escala de Voltaje.

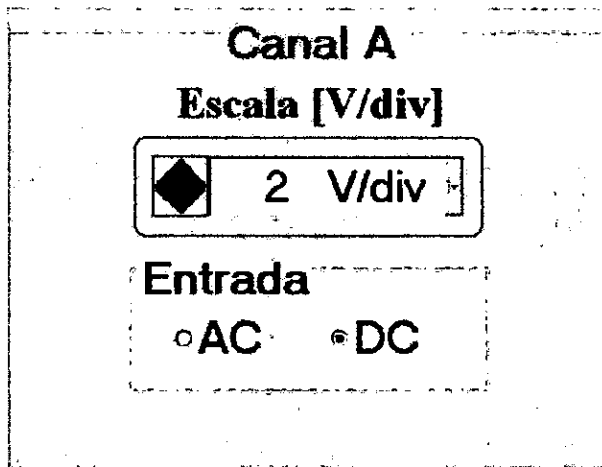


Figura 9.24: Ventana Maximizada de Canal A

En la Tabla 9.10 se describen cada una de las funciones cuando se presiona alguna combinación de teclas y se tiene maximizada la Ventana de Canal A o la Ventana de Canal B.

Teclas	Tarea
F + Ar + Ab	Maximiza la Ventana de Pantalla para graficar la señal eléctrica capturada.
F + Ar	Teniendo el Cuadro de Selección Cyan en el <i>ComboBox</i> de Escala de Voltaje, se incrementa en tres el índice para mostrar un valor de Escala de Voltaje más grande. Teniendo el Cuadro de Selección Cyan en el <i>Frame</i> de Entrada (AC o DC), se selecciona AC si estaba seleccionado DC o se selecciona DC si estaba seleccionado AC.
F + Ab	Teniendo el Cuadro de Selección Cyan en el <i>ComboBox</i> de Escala de Voltaje, se decrementa en tres el índice para mostrar un valor de Escala de Voltaje más pequeño. Teniendo el Cuadro de Selección Cyan en el <i>Frame</i> de Entrada (AC o DC), se selecciona AC si estaba seleccionado DC o se selecciona DC si estaba seleccionado AC.
F	Si no se ve en ninguna parte de la Ventana de Canal A o Canal B (según la que este maximizada en ese momento) el Cuadro de Selección Cyan, este aparece en el <i>ComboBox</i> de Escala de Voltaje. Teniendo el Cuadro de Selección Cyan en el <i>ComboBox</i> de Escala de Voltaje, se cambia dicho Cuadro de Selección Cyan al <i>Frame</i> de Entrada (AC o DC). Teniendo el Cuadro de Selección Cyan en el <i>Frame</i> de Entrada (AC o DC), desaparece el Cuadro de Selección Cyan de la Ventana de Canal A o Canal B (según la que esté maximizada en ese momento).
Ar	Teniendo el Cuadro de Selección Cyan en el <i>ComboBox</i> de Escala de Voltaje, se incrementa en uno el índice para mostrar un valor de Escala de Voltaje más grande. Teniendo el Cuadro de Selección Cyan en el <i>Frame</i> de Entrada (AC o DC), se selecciona AC si estaba seleccionado DC o se selecciona DC si estaba seleccionado AC.
Ab	Teniendo el Cuadro de Selección Cyan en el <i>ComboBox</i> de Escala de Voltaje, se decrementa en uno el índice para mostrar un valor de Escala de Voltaje más pequeño. Teniendo el Cuadro de Selección Cyan en el <i>Frame</i> de Entrada (AC o DC), se selecciona AC si estaba seleccionado DC o se selecciona DC si estaba seleccionado AC.
Ar + Ab	Teniendo el Cuadro de Selección Cyan en el <i>ComboBox</i> de Escala de Voltaje, se abre o se cierra este objeto como cuando se presiona la tecla F4 del teclado normal para abrir cualquier <i>ComboBox</i> en el ambiente Windows. Teniendo el Cuadro de Selección Cyan en el <i>Frame</i> de Entrada (AC o DC), se minimiza la Ventana de Canal A o Canal B (según la que esté maximizada en ese momento) para tener la Forma Principal del Programa "Módulo de Osciloscopio" con todas las Ventanas minimizadas. Si no se ve en ninguna parte de la Ventana de Canal A o Canal B (según la que esté maximizada en ese momento) el Cuadro de Selección Cyan, se minimizan estas Ventanas para tener la Forma Principal del Programa "Módulo de Osciloscopio" con todas las Ventanas minimizadas.

Tabla 9.10: Funciones Realizables cuando las Ventanas de Canal A o B están Maximizadas

El código que a continuación se muestra, se ejecuta cuando se tiene maximizadas las Ventanas de Canal A, o Canal B, o Tiempo o Trigger y se presionan simultáneamente las teclas Fu + Ar de la Punta Inalámbrica.

```

If Func = 1 Then ' Si el Focus está en algún ComboBox de
    ' Escala de Voltaje, Escala de Tiempo o Nivel.
    opci = opci - 3 ' Decrementa 3 Opci (índice de ComboBox).
    If opci < 0 Then ' Si Opci < primer Elemento de ComboBox.
        opci = 0 ' Asigna primer elemento de ComboBox.
    End If
    cmbEscala(Vent).ListIndex = opci ' Asigna como índice del
    ' ComboBox nuevo valor de Opci.
Else ' Si Focus está en algún OptionButton de Entradas para
    ' Can A., Can. B, Canal de Trigger o Flanco de Trigger.
    opci = opci + 1 ' Incrementa Opci (índice del ComboBox).
    If (opci = 3) Then ' Si decremento fue tres.
        opci = 1 ' Asigna el primer OptionButton.
    End If
    If Vent = 1 Then ' Si la Ventana es Canal A.
        ActualizaBotónOpción opci ' OptionButtons AC o DC.
    Else ' Si Ventana es Canal B o Trigger.
        ActualizaBotónOpción opci + Vent + (2 * (Func - 2))
        ' OptionButtons AC o DC (A. B), Sub, Baj, A o B (trig).
    End If
End If

```

Como se puede observar en la Tabla 9.10 y en el código anterior, al presionar simultáneamente las teclas Fu + Ar de la Punta Inalámbrica, y el programa “Módulo de Osciloscopio” tiene la Ventana de Canal A, Canal B, Tiempo o Trigger activa maximizada, se presentan dos casos. El primer caso sucede cuando el Cuadro de Selección Cyan está alrededor ya sea, del control de la Escala de Voltaje (Ventanas de Canal A o B maximizadas), o de la Escala de Tiempo (Ventana de Tiempo maximizada), o del Nivel de Trigger (Ventana de Trigger maximizada). Como se puede observar estos controles de las ventanas están hechos con el objeto *ComboBox* de Visual Basic, y se controlan a través de la variable *Func*, asignándoles siempre a los 4 objetos, *Func = 1*. Después de esta asignación, se decrementa en 3 el apuntador *opci*,

el cual apunta a los valores contenidos en el control de la Escala de Voltaje (para la Ventanas de Canal A y B), mostrando así, un valor de la Escala de Voltaje más grande; si la Ventana de Tiempo está activa, el decremento de *opci* mostraría un valor de la Escala de Tiempo más grande; si la Ventana de Trigger está activa, el decremento de *opci* mostraría un valor del Nivel de Disparo más grande. Antes de mostrar el valor al que apunta *opci*, se debe verificar que dicho decremento no fue menor a un límite inferior, en este caso es 0; si fue menor al límite inferior, entonces se hace $opci = 0$. Ahora sí, después de hacer esta verificación, se muestra el nuevo valor de la Escala de Voltaje (si la Ventana del Canal A o B está activa), o de la Escala de Tiempo (si la Ventana de Tiempo está activa), o del Nivel de Disparo (si la Ventana de Trigger está activa). El segundo caso se presenta cuando el Cuadro de Selección Cyan está en la subventana que tiene los controles AC o DC (si la Ventana del Canal A o B está activa), o en la subventana que tiene los controles del Canal de Disparo, o en la subventana que tiene los controles del Flanco de Disparo (estas dos subventanas están contenidas en la Ventana de Trigger). Primero se incrementa *opci* en 1, que en este caso actúa como apuntador de los controles de esas subventanas. Ya que dichas subventanas sólo tienen dos controles como máximo, cuando el incremento es 3 se hace $opci = 1$, queriendo decir con esto que apuntará al primer control de la subventana. Después del incremento, a través del procedimiento *ActualizaBotónOpción* se selecciona el control referido por *opci*.

Cuando la Ventana de Canal A, o Canal B, o Tiempo, o Trigger está maximizada y se presionan simultáneamente $Fu + Ab$ en la Punta Inalámbrica, se presentan también dos casos muy similares. La diferencia del primer caso radica en que en lugar de decrementar *opci* en 3, se incrementa en 3; si el incremento fue mayor

a un límite superior (el cual depende del máximo número de valores que tenga el control que se está refiriendo), se asigna a *opci* el valor de ese límite superior. La diferencia del segundo caso radica en que en lugar de incrementar *opci* en 1 se decrementa; si el decremento de *opci* es 0, se hace $opci = 2$, para así apuntar al siguiente control de la subventana a la que se está refiriendo en ese momento.

Cuando la Ventana de Canal A, o Canal B, o Tiempo, o Trigger está maximizada y se presiona desde la Punta Inalámbrica la tecla Ar: también se tienen dos casos, casi iguales a los que se presentan cuando desde la Punta Inalámbrica se presionan simultáneamente las teclas Fu + Ar y está maximizada alguna de esas Ventanas. La única diferencia se presenta en el primer caso, ya que en lugar de decrementar en 3 *opci*, se decrementa en 1, teniendo el mismo efecto ya descrito.

Cuando se presiona desde la Punta Inalámbrica la tecla Ab y se tiene maximizada la Ventana de Canal A, o Canal B, o Tiempo, o Trigger, se presentan también dos casos, semejantes a los que se tienen cuando desde la Punta Inalámbrica se presionan simultáneamente las teclas Fu + Ab y está maximizada alguna de esas Ventanas. La única diferencia se tiene en el primer caso, puesto que en lugar de incrementar *opci* en 3, se incrementa en 1, haciendo lo mismo que ya describió.

Cuando se presiona desde la Punta Inalámbrica la tecla Fu y se tiene maximizada la Ventana de Canal A, o Canal B, o Tiempo, o Trigger, se presentan tres casos, los cuales hacen rotar por los controles de esas Ventanas, al Cuadro de Selección Cyan y así poder manipular cada uno de estos controles. El control de la rotación se

hace a través de la variable *Func*. A continuación se muestra el código con el cual se lleva el control de la rotación.

```

FuA = Func ' Función Anterior = Función.
Func = Func + 1 ' Incrementa Función.
If ((Vent = 1 Or Vent = 2) And Func = 3) Or (Vent = 3 And Func = 2)
Or ((Vent = 4 Or Vent = 6) And Func = 4) Or (Vent = 5 And Func = 7) Then
    ' Si Ventana de Canal A o Canal B y el incremento fue 3 o
    ' Ventana de Tiempo y el incremento fue 2 o
    ' Ventana de Trigger o Comandos y el incremento fue 4 o
    ' Ventana de Pantalla y el incremento fue 7.
    Func = 0 ' Ninguna Función Asignada (Cuadro de Selec. Cyan).
End If
    
```

Primero se registra a través de la variable *FuA*, que control o subventana tiene al Cuadro de Selección Cyan alrededor de él. Posteriormente se incrementa *Func*, que actúa como apuntador del control o subventana que debe tener alrededor el Cuadro de Selección Cyan. Cada Ventana tiene un número máximo de controles y subventanas a las que puede apuntar *Func* o bien que pueden tener el Cuadro de Selección Cyan, por ejemplo para la Ventana de Canal A y Canal B, *Func* sólo puede apuntar al Control de la Escala de Voltaje y a la Subventana del Tipo de Entrada de Señal, por lo que el valor máximo de *Func* es 2; para la Ventana de Tiempo, *Func* sólo puede apuntar al control de la Escala de Tiempo, por lo que el valor máximo de *Func* es 1; para la Ventana de Trigger, *Func* puede apuntar al control del Nivel de Disparo, o a la Subventana del Canal de Disparo, o la Subventana del Flanco de Disparo, por lo que el máximo valor de *Func* es 3; para la Ventana de Pantalla, *Func* apunta a los Botones de Información de la señal del Canal A, Canal B o Tiempo, o bien puede apuntar a los controles que hacen ver las líneas de referencia de los Canales A, B o Tiempo, por lo que en esta Ventana el valor máximo de *Func* es 6; para la Ventana de Comandos, *Func* apunta a los Botones de Imprimir, Guardar o Salir, por lo que el valor máximo de *Func* para

esta ventana es de 3. Por esta razón se debe verificar que el incremento de *Func* no haya pasado el valor Máximo de cada ventana, y si fuera así *Func* tendría que ser 0, o lo que es lo mismo, el Cuadro de Selección Cyan debe desaparecer de la Ventana maximizada en cuestión, a través de la siguiente instrucción:

```
shpCuadro(Vent).Visible = False ' Desaparece respectivo
' Cuadro de Selección Cyan (según la ventana requerida).
```

donde *shpCuadro* es el nombre del control que representa el Cuadro de Selección Cyan. Una vez determinado el valor de *Func*, (considerando que los controles de las Ventanas de Canal A, Canal B, Tiempo y Trigger son muy semejantes), se procede a obtener los valores del control que tiene a su alrededor, el Cuadro de Selección Cyan, a través de las siguientes instrucciones.

```
If Func = 1 Then ' Si es Primer Función.
    opci = cmbEscala(Vent).ListIndex ' Opci = al índice
    ' mostrado en el ComboBox.
    cmbEscala(Vent).SetFocus 'Focus en repec. ComboBox.
    RedefineCuadroFunc spnEscala(Vent) ' Redefine objeto de
    ' SpinButton (incremento sensible).
Else ' Si Función es > 1.
    opci = IdenOpciSelec() ' Opci = índice de OptionButton
    ' seleccionado.
    opRefer(opci).SetFocus ' Focus en Nvo. OptionButton.
    If (Vent = 4) And (Func = 2) Then
        ' Si Ventana es Trigger y Función = 2.
        RedefineCuadroFunc frmControl(3) ' Redefine Frame de
        ' Canal de Disparo.
    Else
        RedefineCuadroFunc frmControl(Vent) ' Redefine Frame
        ' de Entrada para Vent. C.A o C.B o Frame
        ' de Flanco de Disparo para Vent. Trig.
    End If
End If
shpCuadro(Vent).Visible = True ' Hace visible respectivo
' Cuadro de Selección Cyan.
```

Como se puede observar en la Tabla 9.10 y en el código anterior, al presionar la tecla *Fu* de la Punta Inalámbrica, y el programa “Módulo de Osciloscopio” tiene la Ventana de Canal A, Canal B, Tiempo o Trigger activa maximizada, se presentan dos casos. El

primer caso sucede cuando el Cuadro de Selección Cyan está alrededor ya sea, del control de la Escala de Voltaje (Ventanas de Canal A o B maximizadas), o de la Escala de Tiempo (Ventana de Tiempo maximizada), o del Nivel de Trigger (Ventana de Trigger maximizada). Primero se debe asignar a *opci* (apuntador de índices de los controles antes mencionados), el valor de índice que tiene dicho control, para darle el Focus a ese control. Por último se dimensiona a través de la función *RedefineCuadroFunc*, el Cuadro de Selección Cyan con respecto al objeto *SpinButton*, el cual hace incrementos o decrementos en 1 del valor del objeto. El segundo caso ocurre cuando el Cuadro de Selección Cyan está alrededor de la Subventana del Tipo de Señal Eléctrica que se captura (para las Ventanas de Canal A o Canal B maximizadas), o la del Canal de Disparo, o la del Flanco de Disparo (estas dos para la Ventana de Trigger maximizada). Primeramente se asigna a *opci* el valor de índice del control que está seleccionado en alguna de las subventanas antes mencionadas, a través de la función *IdenOpciSelec()*, para después asignarle a ese control el Focus. Por último se dimensiona el Cuadro de Selección Cyan con respecto a la subventana de Tipo de Señal de Entrada (para las Ventanas de Canal A o B), las de Canal de Disparo o Flanco de Disparo (para la Ventana de Trigger), a través de la función *RedefineCuadroFunc*. Para ambos casos, al final se hace visible (si no lo estaba) el Cuadro de Selección Cyan.

Como se puede observar en la Tabla 9.10, al presionar simultáneamente las teclas Ar + Ab de la Punta Inalámbrica, y el programa “Módulo de Osciloscopio” tiene la Ventana de Canal A, Canal B, Tiempo o Trigger activa maximizada, se presentan dos casos: el primero actúa como si se presionara la tecla de Enter del teclado de la PC, para aceptar ciertos parámetros de alguna ventana de

opciones de un Software para Windows; el segundo caso actúa como cuando se tiene algún *ComboBox* de Windows y se presiona F4 para abrirlo o cerrarlo. El código que se muestra a continuación corresponde al primer caso, en donde el Cuadro de Selección Cyan está alrededor de los Controles de Escala de Voltaje (para las Ventanas de Canal A o Canal B), o el de Escala de Tiempo (para la Ventana de Tiempo), o el de Nivel de Disparo (para la Ventana de Trigger).

```
If (Func = 1) And (Vent < 5) Then ' Cuando el cuadro de Selección
    ' Cyan está en ComboBox de Vent. C. A, C. B, Tiempo o Trigger.
    cmbEscala(Vent).SetFocus ' Focus en ComboBox de Vent. Actual.
    SendKeys "{F4}", True ' Emula el presionar F4 con la cual en el
        ' ambiente de Windows se abre o cierran los ComboBox.
End If
```

Lo que se hace en esta parte, es asignar el Focus al respectivo control de las ventanas antes mencionadas, para después ejecutar por software el presionar la tecla F4 del teclado normal. Cuando se hace esto se abre o se cierra el objeto *ComboBox*.

Las siguientes líneas de código corresponde al segundo caso.

```
AgrandaForma False ' Tamaño Normal de la forma.
Gigante = False ' Ventana Minimizada.
ReDefineVentana Gigante, Vent ' Minimiza la ventana actual.
Select Case Vent
    Case 1 To 4 ' Si es Vent. de C. A, C. B, Tiempo o Trig.
        shpCuadro(Vent).Visible = False ' Respec. Cuadro Cyan.
        cmbEscala(Vent).SetFocus ' Focus en Respec. ComboBox.
    Case 5 ' Si la Ventana es Pantalla.
        RedefineRejillas frmOsciloscopio ' Dibuja Gráticula y
            ' Gráficas de la Ventana Pantalla.
        shpCuadro(11).Visible = False ' Cuadro Sel. Cyan de A.
        shpCuadro(12).Visible = False ' Cuadro Sel. Cyan de B.
        shpCuadro(13).Visible = False ' Cuadro Sel. Cyan de T.
        btnInfo(1).SetFocus ' Focus en Botón de Informa. de A.
    Case 6
        shpCuadro(Vent).Visible = False ' Cuadro Selec. Cyan.
        btnComando(1).SetFocus ' Focus en botón de Imprimir.
End Select
Va = 0 ' Ninguna Ventana anterior maximizada.
Func = 0 ' Ninguna Función Activa.
```


Módulo de Osciloscopio

```
For i = 1 To 6 ' Para las 6 Vent. habilita Frames de Vent.  
    frmVentana(i).Enabled = True ' por si no están activas.  
Next i
```

Este código funciona con todas las Ventanas, excepto la Ventana de Información. Primero se reduce la forma del programa, minimizando la ventana maximizada por medio del procedimiento *ReDefineVentana* y establecer que ya no hay ventanas maximizadas. Posteriormente se desaparece el Cuadro de Selección Cyan (*shpCuadro*) para establecer el Focus en el control de la Escala de Voltaje (si está activa la Ventana del Canal A o Canal B), o en el control de la Escala de Tiempo (si está activa la Ventana de Tiempo), o en el control de Nivel (si está activa la Ventana de Trigger), o en el Botón de Información del Canal A (si está activa la Ventana de Pantalla), o en el Botón de Imprimir (si está activa la Ventana de Comandos). Por último se habilitan cada una de las 6 Ventanas principales ya minimizadas y se establece que no hay ningún objeto con el Cuadro de Selección Cyan (*Func = 0*) además de que no hay Ventanas Maximizadas (*Va=0*), para otros procesos.

En la Figura 9.25 se muestra la Ventana de Tiempo Maximizada con el Cuadro de Selección Cyan en el control de Escala de Tiempo.

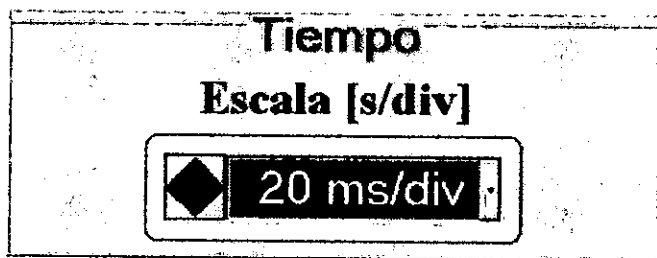


Figura 9.25: Ventana de Tiempo Maximizada

En la Tabla 9.11 se describen las funciones que se pueden ejecutar con el teclado remoto cuando se presiona alguna combinación de botones y la Ventana de Tiempo está Maximizada.

Teclas	Tarea
F + Ar + Ab	Maximiza la Ventana de Pantalla para graficar la señal eléctrica capturada.
F + Ar	Teniendo el Cuadro de Selección Cyan en el <i>ComboBox</i> de Escala de Tiempo, se incrementa en tres el índice para mostrar un valor de Escala de Tiempo más grande.
F + Ab	Teniendo el Cuadro de Selección Cyan en el <i>ComboBox</i> de Escala de Tiempo, se decrementa en tres el índice para mostrar un valor de Escala de Tiempo más pequeño.
F	Si no se ve en ninguna parte de la Ventana de Tiempo el Cuadro de Selección Cyan, este aparece en el <i>ComboBox</i> de Escala de Tiempo. Teniendo el Cuadro de Selección Cyan en el <i>ComboBox</i> de Escala de Tiempo, desaparece el Cuadro de Selección Cyan de la Ventana de Tiempo.
Ar	Teniendo el Cuadro de Selección Cyan en el <i>ComboBox</i> de Escala de Tiempo, se incrementa en uno el índice para mostrar un valor de Escala de Tiempo más grande.
Ab	Teniendo el Cuadro de Selección Cyan en el <i>ComboBox</i> de Escala de Tiempo, se decrementa en uno el índice para mostrar un valor de Escala de Tiempo más pequeño.
Ar + Ab	Teniendo el Cuadro de Selección Cyan en el <i>ComboBox</i> de Escala de Tiempo, se abre o se cierra este objeto como cuando se presiona la tecla F4 del teclado normal para abrir cualquier <i>ComboBox</i> en el ambiente Windows. Si no se ve en ninguna parte de la Ventana de Tiempo el Cuadro de Selección Cyan, se minimiza esta Ventana para tener la Forma Principal del Programa "Módulo de Osciloscopio" con todas las Ventanas minimizadas.

Tabla 9.11: Funciones Realizables cuando las Ventana de Tiempo está Maximizada

En la Figura 9.26 se muestra la Ventana de Trigger Maximizada.

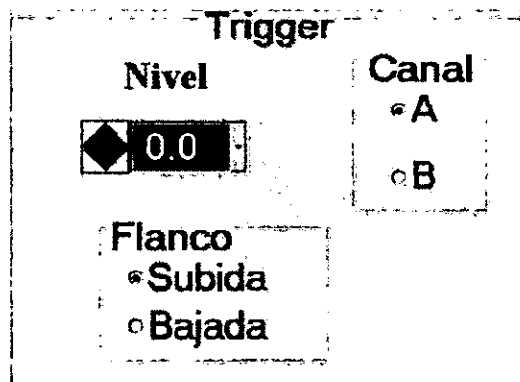


Figura 9.26: Ventana de Trigger Maximizada

En la Tabla 9.12 se describen las funciones que se pueden ejecutar con el teclado remoto cuando se presiona alguna combinación de botones y la Ventana de Trigger está Maximizada.

Teclas	Tarea
F + Ar + Ab	Maximiza la Ventana de Pantalla para graficar la señal eléctrica capturada.
F + Ar	Teniendo el Cuadro de Selección Cyan en el <i>ComboBox</i> de Nivel de Disparo, se incrementa en tres el índice para mostrar un valor de Nivel de Disparo más grande. Teniendo el Cuadro de Selección Cyan en el <i>Frame</i> de Canal Disparo (A o B), se selecciona A, si estaba seleccionado B o se selecciona B, si estaba seleccionado A. Teniendo el Cuadro de Selección Cyan en el <i>Frame</i> de Flanco de Disparo (Subida o Bajada), se selecciona Subida si estaba seleccionado Bajada o se selecciona Bajada si estaba seleccionado Subida.
F + Ab	Teniendo el Cuadro de Selección Cyan en el <i>ComboBox</i> de Nivel de Disparo, se decrementa en tres el índice para mostrar un valor de Nivel de Disparo más pequeño. Teniendo el Cuadro de Selección Cyan en el <i>Frame</i> de Canal de Disparo (A o B), se selecciona A si estaba seleccionado B o se selecciona B si estaba seleccionado A. Teniendo el Cuadro de Selección Cyan en el <i>Frame</i> de Flanco de Disparo (Subida o Bajada), se selecciona Subida si estaba seleccionado Bajada o se selecciona Bajada si estaba seleccionado Subida.
F	Si no se ve en ninguna parte de la Ventana de Trigger el Cuadro de Selección Cyan, éste aparece en el <i>ComboBox</i> de Nivel de Disparo. Teniendo el Cuadro de Selección Cyan en el <i>ComboBox</i> de Nivel de Disparo, se cambia dicho Cuadro de Selección al <i>Frame</i> de Canal de Disparo (A o B). Teniendo el Cuadro de Selección Cyan en el <i>Frame</i> de Canal de Disparo (A o B), se cambia el Cuadro de Selección Cyan al <i>Frame</i> de Flanco de Disparo (Subida o Bajada). Teniendo el Cuadro de Selección Cyan en el <i>Frame</i> de Flanco de Disparo (Subida o Bajada), desaparece el Cuadro de Selección Cyan de la Ventana de Trigger.
Ar	Teniendo el Cuadro de Selección Cyan en el <i>ComboBox</i> de Nivel de Disparo, se incrementa en uno el índice para mostrar un valor de Nivel de Disparo más grande. Teniendo el Cuadro de Selección Cyan en el <i>Frame</i> de Canal (A o B), se selecciona A si estaba seleccionado B o se selecciona B si estaba seleccionado A. Teniendo el Cuadro de Selección Cyan en el <i>Frame</i> de Flanco de Disparo (Subida o Bajada), se selecciona Subida si estaba seleccionado Bajada o se selecciona Bajada si estaba seleccionado Subida.
Ab	Teniendo el Cuadro de Selección Cyan en el <i>ComboBox</i> de Nivel de Disparo, se decrementa en uno el índice para mostrar un valor de Nivel de Disparo más pequeño. Teniendo el Cuadro de Selección Cyan en el <i>Frame</i> de Canal de Disparo (A o B), se selecciona A si estaba seleccionado B o se selecciona B si estaba seleccionado A. Teniendo el Cuadro de Selección Cyan en el <i>Frame</i> de Flanco de Disparo (Subida o Bajada), se selecciona Subida si estaba seleccionado Bajada o se selecciona Bajada si estaba seleccionado Subida.
Ar + Ab	Teniendo el Cuadro de Selección Cyan en el <i>ComboBox</i> de Nivel de Disparo, se abre o se cierra este objeto como cuando se presiona la tecla F4 del teclado normal para abrir cualquier <i>ComboBox</i> en el ambiente Windows. Teniendo el Cuadro de Selección Cyan en el <i>Frame</i> de Canal de Disparo (Subida o Bajada) o en el <i>Frame</i> de Flanco de Disparo (Subida o Bajada), se minimiza la Ventana de Trigger para tener la Forma Principal del Programa "Módulo de Osciloscopio" con todas las Ventanas minimizadas. Si no se ve en ninguna parte de la Ventana de Trigger el Cuadro de Selección Cyan, se minimiza esta Ventana para tener la Forma Principal del Programa "Módulo de Osciloscopio" con todas las Ventanas minimizadas.

Tabla 9.12: Funciones Realizables cuando las Ventana de Trigger está Maximizada

Las 7 combinaciones de teclas que se pueden generar con los tres botones del teclado remoto especial cuando están Maximizadas la Ventana de Tiempo o la Ventana de Trigger ejecutan funciones muy semejantes de cuando se hacen uso del teclado especial cuando está maximizada la Ventana de Canal A, por lo tanto, el código de programación que atiende a cada combinación es el mismo que se explicó anteriormente, sólo que ahora refiriéndose a las Ventanas de Tiempo o a la de Trigger..

En la Figura 9.27 se muestra la Ventana de Pantalla cuando está maximizada y el cuadro de Selección Cyan está en el control de Referencia de la Subventana B. En esta figura también se muestra la gráfica de una señal eléctrica.

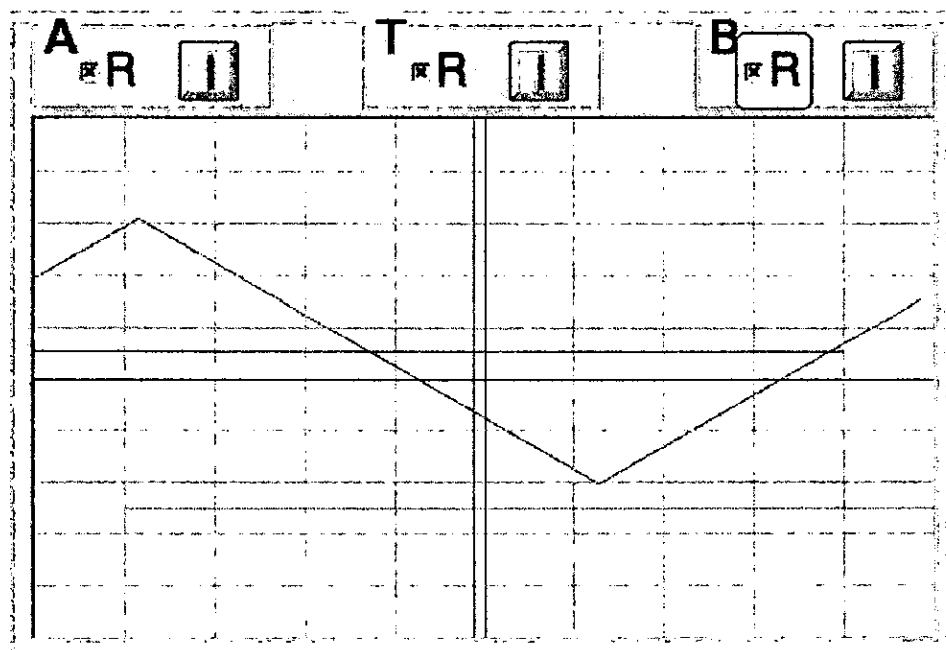


Figura 9.27: Ventana Pantalla Maximizada

En la Tabla 9.13 se describen las funciones que se pueden ejecutar con el teclado remoto cuando se presiona alguna combinación de botones y la Ventana Pantalla está Maximizada.

Teclas	Tarea
F + Ar + Ab	Maximiza la Ventana de Pantalla para graficar la señal eléctrica capturada..
F + Ar	Teniendo el Cuadro de Selección Cyan en el <i>CheckBox</i> de Referencia de A o B, se sube una División de la Graticula la Línea de Referencia del Canal A o Canal B (según donde esté el Cuadro Color Cyan. Teniendo el Cuadro de Selección Cyan en el <i>CheckBox</i> de Referencia de T, se mueve a la derecha una División de la Graticula la Línea de Referencia del Tiempo.
F + Ab	Teniendo el Cuadro de Selección Cyan en el <i>CheckBox</i> de Referencia de A o B, se baja una División de la Graticula la Línea de Referencia del Canal A o Canal B (según donde esté el Cuadro Color Cyan. Teniendo el Cuadro de Selección Cyan en el <i>CheckBox</i> de Referencia de T, se mueve a la izquierda una División de la Graticula la Línea de Referencia del Tiempo.
F	Si no se ve en ninguna parte de la Ventana de Pantalla el Cuadro de Selección Cyan, éste aparece en <i>CheckBox</i> de Referencia de A. Si el Cuadro de Selección está en el <i>CheckBox</i> de Referencia de A pasa al Botón de Información de A. Si el Cuadro de Selección está en el botón de Información de A, pasa al <i>CheckBox</i> de Referencia de B. Si el Cuadro de Selección está en el <i>CheckBox</i> de Referencia de B pasa al Botón de Información del B. Si el Cuadro de Selección está en el botón de Información de B, pasa al <i>CheckBox</i> de Referencia de T. Si el Cuadro de Selección está en el <i>CheckBox</i> de Referencia de T pasa al Botón de Información del T. Si el Cuadro de Selección está en el Botón de Información de T desaparece el Cuadro de Selección Cyan.
Ar	Teniendo el Cuadro de Selección Cyan en el <i>CheckBox</i> de Referencia de A o B, se sube una Subdivisión de la Graticula la Línea de Referencia del Canal A o Canal B (según donde esté el Cuadro Color Cyan. Teniendo el Cuadro de Selección Cyan en el <i>CheckBox</i> de Referencia de T, se mueve a la derecha una Subdivisión de la Graticula la Línea de Referencia del Tiempo.
Ab	Teniendo el Cuadro de Selección Cyan en el <i>CheckBox</i> de Referencia de A o B, se baja una Subdivisión de la Graticula la Línea de Referencia del Canal A o Canal B (según donde esté el Cuadro Color Cyan. Teniendo el Cuadro de Selección Cyan en el <i>CheckBox</i> de Referencia de T, se mueve a la izquierda una Subdivisión de la Graticula la Línea de Referencia del Tiempo.
Ar + Ab	Teniendo el Cuadro de Selección Cyan en algún de los tres <i>CheckBox</i> de Referencia (A, B o T) se selecciona o no dicho objeto, como si se hiciera <i>Click</i> en él con el Mouse. Teniendo el Cuadro de Selección Cyan en alguno de los tres botones de Información (A, B o T) se ejecuta su procedimiento como si se hiciera <i>Click</i> en él con el Mouse. Si no se ve en ninguna parte de la Ventana de Pantalla, se minimiza dicha Ventana para tener la Forma Principal del Programa "Módulo de Osciloscopio" con todas las Ventanas minimizadas.

Tabla 9.13: Funciones Realizables cuando las Ventana Pantalla está Maximizada

Las tareas y procedimientos que se ejecutan cuando se tiene maximizada la Ventana de Pantalla y se presionan simultáneamente las Teclas Fu + Ar + Ab del teclado remoto especial, son los mismos que se ejecutan cuando se presiona esa combinación de teclas y se tiene maximizada la Ventana de Canal A..

Cuando se tiene maximizada la Ventana de Pantalla, el Cuadro de Selección Cyan está alrededor de los controles para ver las líneas de referencia y se presionan simultáneamente los botones Fu + Ar de la Punta Inalámbrica se ejecuta el siguiente código:

```

If (Vent = 5) And (Func Mod 2) And (ChkR((Func + 1) \ 2).Value =
True) Then
    ' Si la Ventana es Pantalla y la Función corresponda a los
    ' CheckBox de A, B o T con valor verdadero.
    If (Func = 1) Or (Func = 3) Then ' Si es CheckBox de A o B.
        PLR = LinPosición((Func + 1) \ 2).Y1 - 10 ' Incrementa
        ' Subdivisión de Línea Refer. de C. A o C. B.
        If (PLR < 10) Then ' Si Lín. Ref. < Mínima Coordenada Y.
            PLR = 10 ' Asigna Mínima coordenada Y.
        End If
        LinPosición((Func + 1) \ 2).Y1 = PLR ' Nva. Pos. de Línea
        LinPosición((Func + 1) \ 2).Y2 = PLR ' Ref. de C.A o C.B.
        NvosValRef (Func + 1) \ 2 ' Actualiza Menús de Posición.
    End If
    If (Func = 5) Then ' Si es CheckBox de T (Tiempo).
        PLR = LinPosición(3).X1 + (ResoluciónX / 10)
        ' Incrementa Subdivisión de Línea Referencial de Tiempo.
        If (PLR > (ResoluciónX * 10) - (ResoluciónX / 10)) Then
            ' Si Opci > Máxima Coordenada X.
            PLR = (ResoluciónX * 10) - (ResoluciónX / 10)
            ' Asigna Máxima coordenada X.
        End If
        LinPosición(3).X1 = PLR ' Nueva posición de Línea
        LinPosición(3).X2 = PLR ' Referencial del Tiempo.
        NvosValRef 3 ' Actualiza Menú de Posición de Tiempo.
    End If
End If

```

Cabe aclarar que los controles de selección (*chkR*) sirven para hacer visibles o no, las líneas de referencia (*LinPosición*). Tanto los controles como las líneas de referencias son arreglos de objetos. Los índices de los controles de selección son 1, 3 y 5 que corresponden al Canal A, Canal B y Tiempo respectivamente, mientras que los índices de las líneas de referencia son 1, 2, y 3 que corresponden al Canal A, Canal B y Tiempo respectivamente. Como se puede observar en la Tabla 9.13 y en el código anterior, al presionar dichas teclas de la Punta Inalámbrica, se presentan dos casos. El primer caso sucede cuando el Cuadro de Selección Cyan está alrededor ya sea, del control de selección del Canal A [*chkR(1)*] y se desplaza hacia arriba una división de la grátula, la línea de referencia del Canal A [*LinPosición(1)*]; o bien, el Cuadro de Selección Cyan está alrededor del control de selección del Canal B [*chkR(3)*], y se desplaza hacia arriba una división de la grátula, la línea de referencia del Canal B [*LinPosición(2)*]. Para ello, se hace una resta de la posición actual de la línea de referencia menos el valor relativo en pixels de una división vertical; si la diferencia es menor al valor mínimo que puede tener la línea de referencia del Canal en cuestión en la grátula, a dicha línea de referencia se le asigna ese valor mínimo. El segundo caso sucede cuando el Cuadro de Selección Cyan está alrededor del control de selección [*chkR(5)*] del Tiempo, y se desplaza una división de la grátula, la línea de referencia del Tiempo [*LinPosición(3)*]. Para ello, se hace una suma de la posición actual de la línea de referencia más el valor relativo en pixels de una división horizontal; si la suma es mayor al valor máximo que puede tener la línea de referencia del Tiempo en la grátula, a dicha línea de referencia se le asigna ese valor máximo. La variable *PLR* sirve como auxiliar, mientras que el procedimiento *NvosValRef* sirve para actualizar el menú que se relaciona a la línea de referencia que se trata. Para ambos casos, el

desplazamiento de líneas sólo se puede hacer cuando el control de selección tiene un estado de selección verdadero [*chkR(n).value = True*].

Cuando la Ventana de Pantalla está maximizada, el Cuadro de Selección Cyan está alrededor de los controles para ver las líneas de referencia y se presionan simultáneamente *Fu + Ab* en la Punta Inalámbrica, se presentan también dos casos muy similares. La diferencia del primer caso radica en que en lugar de hacer una resta de la posición actual de la línea de referencia, se hace una suma de dicha posición mas el valor relativo en pixels de una división vertical; si la suma es mayor al valor máximo que puede tener la línea de referencia del Canal en cuestión en la gráticula, a dicha línea de referencia se le asigna ese valor máximo. La diferencia del segundo caso radica en que en lugar de hacer una suma de la posición actual de la línea de referencia, se hace una resta de dicha posición, menos el valor relativo en pixels de una división horizontal; si la suma es menor al valor mínimo que puede tener la línea de referencia del Tiempo en la gráticula, a dicha línea de referencia se le asigna ese valor mínimo.

Cuando se presiona la tecla *Ar* o la Tecla *Ab* de la Punta Inalámbrica, se tiene maximizada la Ventana de Pantalla y el Cuadro de Selección está alrededor de un Control de Selección para ver alguna línea de referencia, se presentan casos muy parecidos a los descritos anteriormente cuando se presionan simultáneamente las teclas *Fu + Ar* o las teclas *Fu + Ab*. La diferencia radica en el caso de las líneas de referencia de los Canales *A* o *B*, que en lugar de sumar o restar divisiones verticales de la gráticula, ahora se suma o restan subdivisiones verticales de la gráticula. En el caso de la línea de referencia del Tiempo, la diferencia radica, que en lugar de sumar o restar

divisiones horizontales de la gráticula, ahora se suma o restan subdivisiones horizontales de la gráticula.

Cuando se presiona la tecla Fu de la Punta Inalámbrica y se tiene maximizada la Ventana de Pantalla, se ejecuta el siguiente código que sirve para rotar el Cuadro de Selección Cyan a través de los controles de selección y los botones de información.

```
If ((FuA) \ 2) Then ' Si ya había Función Anterior.
    shpCuadro(((FuA) \ 2) + 10).Visible = False
    ' Desaparece cuadro de Selec. Cyan de A, B o T.
End If
If (Func Mod 2) Then ' Si Función Actual corresponde a
    ' Checkbox de A, B, o T.
    RedefineCuadroFunc chkR((Func + 1) \ 2)
    ' Redimensiona el CheckBox requerido.
Else ' Si Función Actual corresponde a algún Botón de
    ' de información de A, B, o T.
    RedefineCuadroFunc btnInfo((Func + 1) \ 2)
    ' Redimensiona botón de información requerido.
End If
shpCuadro(((Func + 1) \ 2) + 10).Visible = True
    ' Hace Visible Cuadro de Selección Cyan para indicar
    ' al objeto (CheckBox o Botón de Inf.) requerido.
```

El código anterior no está completo, ya que el principio es muy parecido al código que se explicó cuando se tenía maximizada la Ventana de Canal A y se presionaba la tecla Fu del teclado remoto especial. Hay que recordar que primero se registra en *FuA* el índice de objetos que tiene el Cuadro de Selección Cyan ($FuA = Fu$) y después se hace un incremento del índice de objetos ($Fu = Fu + 1$). En este código, primero se pregunta si antes de hacer el incremento del índice de objetos, el Cuadro de Selección Cyan no estaba alrededor de un botón de Información (de Canal A, Canal B o Tiempo), ya que si fuera así, tendría que desaparecer dicho Cuadro de Selección de la subventana activa. El hecho de desaparecer el Cuadro de Selección, es porque cada

subventana (del Canal A, Canal B y Tiempo) de la Ventana de Pantalla tiene su propio Cuadro de Selección Cyan, es decir en la Ventana de Pantalla hay tres Cuadros de Selección Cyan, mientras que en las otras seis ventanas sólo hay un Cuadro de Selección Cyan por ventana. Después se procede a identificar si el incremento del apuntador de objetos (*Func*), corresponde a poner el Cuadro de Selección Cyan alrededor de un botón de Información o alrededor de un objeto de selección de las líneas de referencia, para así darle las dimensiones correctas al Cuadro de Selección Cyan, a través del procedimiento *RedefineCuadroFunc*. Por último se hace visible dicho cuadro de Selección.

Cuando se presionan simultáneamente las teclas Ar + Ab de la Punta Inalámbrica y se tiene maximizada la Ventana de Pantalla, se ejecuta el siguiente código, con el cual se actúa como si se presionara la tecla Enter del teclado de la PC y ejecutar así el evento *Click* del Mouse que actúa sobre los objetos de selección de las líneas de referencia y los botones de información.

```
If Func Mod 2 Then ' Si es algún CheckBox de Referencias.
    chkR((Func + 1) \ 2).Value = Not (chkR((Func + 1) \ 2).Value)
                                ' Cambia el valor del CheckBox.
    LinPosición(Func + 1) \ 2.Visible = chkR(Func + 1) \ 2.Value
Else ' Si es algún botón de Información.
    btnInfo_Click ((Func + 1) \ 2) ' Ejecuta el procedim.
                                ' del botón de Información A o B.
End If
```

Primero se identifica sobre qué objeto está el Cuadro de Selección Cyan; si está alrededor del objeto de selección de las líneas de referencia, dicho objeto ve como se selecciona o deja de estar seleccionado. Si el valor del objeto de selección de las líneas de referencias tiene un valor verdadero, dicha línea de referencia se hace

visible, pero si tiene un valor falso, la línea de referencia se hace invisible. Si el Cuadro de Selección está alrededor del botón de información se ejecuta el procedimiento del evento de *Click* de dicho botón. Este botón es un arreglo de botones, que corresponden a las subventanas de Canal A, Canal B y Tiempo. A continuación se muestra el código que se ejecuta con el evento de *Click* del botón de información que está rodeado por el Cuadro de Selección Cyan.

```
Vent = 5 ' Ventana de Pantalla.
If Gigante Then ' Si está maximizada Ventana Pantalla.
    Func = Index * 2 ' Asigna qué botón se presionó.
    shpCuadro(11).Visible = False ' Desaparece todos los Cuadros de
    shpCuadro(12).Visible = False ' Selec. Cyan de los Frames A, B
    shpCuadro(13).Visible = False ' y T de la Ventana Pantalla.
    shpCuadro(Index + 10).Visible = True ' Aparece Cuadro Cyan.
    RedefineCuadroFunc btnInfo(Index) ' Dimensiona el Cuadro Cyan.
End If
frmVentana(0).Visible = True ' Hace visible Ventana de Información.
RedefineVentana Gigante, Index + 10 'Para Frames de Vent. Pantalla.
B = True
If Index = 3 Then ' Si es el Botón Información de Tiempo.
    B = False
End If
For i = 0 To 4
    ' Para los Cuadros de Textos y Etiquetas de Voltaje, Voltaje RMS
    ' Frecuencia, Periodo, Fase, Escala y Referencia
    ' correspondientes a la Ventana de Información.
    lblInfo(i).Visible = B
    txtInfo(i).Visible = B
Next i
CargaInformación ' Carga Información en los Cuadros de Textos.
```

Este procedimiento tiene como parámetro la variable *Index*, ya que los 3 botones de información son un arreglo del objeto *CommandButton* de Visual Basic; para referirse al botón de Información del Canal A, *Index* es 1; para referirse al botón de Información del Canal B, *Index* es 2; para referirse al botón de Información del Tiempo, *Index* es 3. Como la Ventana de Información sólo aparece cuando se presiona alguno de estos botones, lo primero que se hace es indicar que esta ventana es

hija de la Ventana de Pantalla. Cabe recordar que este procedimiento no sólo se puede activar cuando la ventana de Pantalla esté maximizada, sino que se puede activar cuando no hay ninguna ventana maximizada. Si está alguna ventana maximizada, se debe confirmar qué botón se presionó, es decir qué botón de información debe tener alrededor el Cuadro de Selección Cyan. Posteriormente se desaparecen los tres Cuadros de Selección de la Ventana Pantalla y sólo se deja el de la subventana donde está el botón presionado. Estando o no el modo maximizado se debe aparecer la Ventana de Información y ubicar cada uno de los objetos que la componen a través del procedimiento *ReDefineVentana*. Cabe señalar que cuando se presiona el botón de información del tiempo, sólo aparecen las etiquetas y textos correspondientes a la Escala de Tiempo y la Posición de la Línea de Referencia del Tiempo, mientras que cuando se presionan el Botón de Información del Canal A o del Canal B aparecen las etiquetas y textos correspondientes al Voltaje, Voltaje RMS, Frecuencia, Fase y Periodo (si la señal tuviera esos parámetros) de una señal eléctrica, además de la Escala de Voltaje y Posición de la Línea de Referencia. Por eso cuando se presiona el Botón de Información del Tiempo sólo se ven dos etiquetas y dos textos, mientras que cuando se presiona el Botón de Información de Canal A o del Canal B, se muestran siete etiquetas y siete textos. Por último se llenan los textos con la información requerida a través del procedimiento *CargaInformación*. En la Figura 9.28 se muestra la Ventana de Información Maximizada, con el Cuadro de Selección Cyan en el Botón de Regresar.

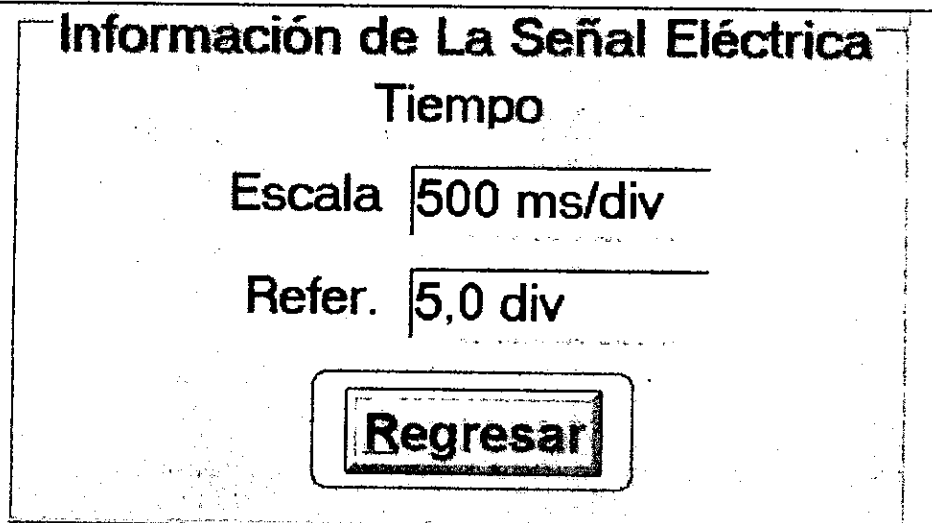


Figura 9.28: Ventana de Información Maximizada

Hay que señalar que cuando se controla remotamente el programa lo único que se puede hacer cuando está esta ventana maximizada de Información es regresar a la Ventana de Pantalla o Capturar una nueva señal eléctrica. Por lo tanto sólo actúan dos combinaciones de teclas las cuales se describen en la Tabla 9.14.

Teclas	Función
F + Ar + Ab	Maximiza la Ventana de Pantalla para graficar la señal eléctrica capturada..
F + Ar	Sin función.
F + Ab	Sin función.
F	Sin función.
Ar	Sin función.
Ab	Sin función.
Ar + Ab	Si está la Ventana de Información Maximizada se ejecuta el su procedimiento como si se hiciera Click en él con el Mouse. Si no se ve en ninguna parte de la Ventana de Comandos el Cuadro de Selección Cyan, se minimiza dicha Ventana para tener la Forma Principal del Programa "Módulo de Osciloscopio" con todas las Ventanas minimizadas.

Tabla 9.14: Funciones Realizables cuando las Ventana de Información está Maximizada

Cuando se presionan simultáneamente las teclas Ar + Ab en la Punta Inalámbrica y la Ventana de Información está maximizada se presiona por Software el Botón de Regresar de la Ventana de Información por medio del evento *Click* del Botón con la siguiente instrucción:

```
btnRegresar_Click ' Ejecuta procedimiento del botón de Regresar.
```

Este procedimiento a su vez ejecuta el siguiente código.

```
frmVentana(5).Enabled = True ' Habilita ventana de Pantalla.  
frmVentana(0).Visible = False ' Desaparece Ventana de Información.  
btnInfo(Canal).SetFocus ' Foco al botón de Información requerido.  
Vent = 5 ' Ventana de Pantalla.
```

Primero se habilita la Ventana de Pantalla y desaparece la Ventana de Información. Después se asigna el Focus al botón de Información que se presionó. Por último se confirma como ventana activa a la Ventana de Pantalla.

En la Figura 9.29 se muestra la Ventana de Comandos cuando está maximizada y el Cuadro de Selección Cyan se encuentra alrededor del Botón Guardar, el cual sirve para almacenar una señal eléctrica en un archivo.



Figura 9.29: Ventana de Comandos Maximizada

En la Tabla 9.15 se describen las funciones que se pueden ejecutar con el teclado remoto cuando se presiona alguna combinación de botones y la Ventana de Comandos está maximizada.

Teclas	Función
F + Ar + Ab	Maximiza la Ventana de Pantalla para graficar la señal eléctrica capturada.
F + Ar	Sin función.
F + Ab	Sin función.
F	Teniendo el Cuadro de Selección Cyan en los botones (CommandButton) de Imprimir o Guardar, pasa al botón de Guardar o Salir respectivamente. Teniendo el Cuadro de Selección Cyan en el botón de Salir, dicho Cuadro de Selección Cyan desaparece. Si no se ve el Cuadro de Selección Cyan en algún botón lo aparece en el primer botón (Imprimir).
Ar	Sin función.
Ab	Sin función.
Ar + Ab	Teniendo el Cuadro de Selección Cyan en alguno de los tres botones (Imprimir, Guardar o Salir) se ejecuta su procedimiento como si se hiciera Click en él con el Mouse. Si no se ve en ninguna parte de la Ventana de Comandos el Cuadro de Selección Cyan, se minimiza dicha Ventana para tener la Forma Principal del Programa "Módulo de Osciloscopio" con todas las Ventanas minimizadas.

Tabla 9.15: Funciones Realizables cuando las Ventana de Comandos está Maximizada

Cuando se presiona la Tecla Fu de la Punta Inalámbrica y se tiene la Ventana de Comandos maximizada, se rota el Cuadro de Selección Cyan por los botones: Imprimir, Guardar y Salir. Estos tres botones forman un arreglo del objeto *CommandButton* de Visual Basic, en donde el Botón de Imprimir tiene el índice 1; el Botón de Guardar tiene el índice 2; y el Botón de Salir tiene el índice 3. A continuación se muestra el código que se ejecuta cuando se presiona remotamente dicha tecla:

```
btnComando(Func).SetFocus ' Focus en Botón requerido.
RedefineCuadroFunc btnComando(Func) ' Redimensiona botón
' de Imprimir, Guardar o Salir.
shpCuadro(Vent).Visible = True ' Hace Visible Cuadro de
' Selección Cyan (indicador de objeto requerido).
```

La primera parte de este procedimiento se omite ya que se explicó cuando se presionaba la tecla Fu de la Punta Inalámbrica y está maximizada la Ventana de Canal A. Hay que señalar que este código sólo se ejecuta cuando $Func \geq 0$. Primero se asigna el Focus al botón cuyo índice es igual al apuntador de botones de la Ventana de Comandos. Después se dimensiona el Cuadro de Selección Cyan con respecto al Botón que tiene ahora el Focus. Por último se hace visible el Cuadro de Selección alrededor del Botón con el Focus.

Cuando se presionan simultáneamente las teclas Ar + Ab de la Punta Inalámbrica, el Cuadro de Selección Cyan está aldedero de alguno de los tres botones y está maximizada la Ventana de Comandos se ejecuta por Software el evento *Click* del botón a través de la siguiente instrucción:

```
btnComando_Click (Func)      ' Ejecuta el procedimiento del
                              ' botón de Información T.
```

Este procedimiento del evento de *Click* ejecuta el siguiente código:

```
Vent = 6 ' Ventana de Comandos
Select Case Index ' Identifica botón (Imprimir, Guardar o Salir).
Case 1 ' Si se presionó el botón "Imprimir"
    If SR.TotalMuest >= 0 Then ' Si hay muestras recibidas.
        frmOsciloscopio.Caption = "Imprimiendo" ' Indica Impresión.
        Intro = Chr(10) + Chr(13) ' Define la Tecla Enter.
        ' Reconoce Valores del Canal A y del Canal B.
        Vlr = "    CANAL A                                CANAL B" +
Intro
        Vlr = Vlr + "Voltaje      = " + Voltaje(1) + "          "
        Vlr = Vlr + "Voltaje      = " + Voltaje(2) + Intro
        Vlr = Vlr + "Voltaje RMS = " + VoltRMS(1) + "          "
        Vlr = Vlr + "Voltaje RMS = " + VoltRMS(2) + Intro
        Vlr = Vlr + "Frecuencia  = " + Frecuencia(1) + "          "
        Vlr = Vlr + "Frecuencia  = " + Frecuencia(2) + Intro
        Vlr = Vlr + "Periodo     = " + Periodo(1) + "          "
        Vlr = Vlr + "Periodo     = " + Periodo(2) + Intro
        Vlr = Vlr + "Fase        = " + Fase(1) + "          "
        Vlr = Vlr + "Fase        = " + Fase(2) + Intro
```


Módulo de Osciloscopio

```
Vlr = Vlr + "Escala      = " + Format(cmbEscala(1).TEXT,
"0.0") + "
Vlr = Vlr + "      Escala      = " + cmbEscala(2).TEXT +
Intro
Vlr = Vlr + "Referencia = " + Format(SR.PEA + (SR.PDA / 10),
"0.0") + "
Vlr = Vlr + "Referencia = " + Format(SR.PEB + (SR.PDB / 10),
"0.0") + Intro

' Reconoce Valores del Tiempo.
Vlr = Vlr + Intro + "      TIEMPO " + Intro
Vlr = Vlr + "Escala      = " + cmbEscala(3).TEXT + Intro
Vlr = Vlr + "Referencia = " + Format(SR.PET + (SR.PDT / 10),
"0.0") + Intro

RedefineRejillas Graticula ' Dibuja Grat. y Gráf. en Impr.
Printer.Print Vlr ' Imprime Valores de la(s) gráfica(s).
Graticula.PrintForm ' Imprime Gráficas de Señales Eléct.
Printer.EndDoc ' Finaliza impresión.
frmOsciloscopio.Caption = "Módulo de Osciloscopio"
End If
Case 2 ' Si se Presionó el botón "Guardar"
If SR.TotalMuest >= 0 Then ' Si hay muestras recibidas.
frmOsciloscopio.Caption = "Guardando" ' Indica que Guarda.
Open Rut2 & ObtenIndiceArch() For Random As #1 Len =
LONG_LINEA
' Abre el archivo binario.
Put #1, 1, SR ' Guarda todo el registro.
Close #1 ' Cierra el archivo.
frmOsciloscopio.Caption = "Módulo de Osciloscopio"
End If
Case 3 ' Si se presionó el botón "Salir"
Salir = True ' Bandera de Salida.
End Select
If Gigante Then ' Si la Ventana está Maximizada.
Func = 0 ' No hay Función Seleccionada.
Gigante = False ' Ventana Minimizada.
shpCuadro(6).Visible = False ' Desaparece Cuadro Cyan.
ReDefineVentana Gigante, 6 ' Redimensiona Ventana de Comandos.
End If
```

Inicialmente se asegura que la ventana activa es la de Comandos, para después identificar qué botón se presionó. En caso de que se haya presionado el Botón de Imprimir (con el cual se imprimen tanto la gráfica como los valores característicos de la señal eléctrica capturada), primero se averigua si se han recibido

las muestras de alguna señal eléctrica; si efectivamente ya se recibieron las muestras de alguna señal eléctrica, se procede a acumular en una variable de tipo string, todos los valores característicos que se obtienen de la(s) señal(es) eléctrica(s) de ambos canales. Cada valor característicos (voltaje, voltaje RMS, frecuencia, fase, periodo, escala de voltaje y posición de la línea de referencia, en caso de que existan) se obtienen con las funciones diseñadas para darlos como un string. Después se dibujan las señales eléctricas dibujadas en la grátula de la Ventana Pantalla, pero en otra grátula de las mismas características y que está en la forma *Grátula*, del programa "Módulo de Osciloscopio". Por último se imprimen los valores y la forma.

En caso de que se haya presionado el Botón de Guardar se procede a identificar si se recibieron muestras de alguna señal eléctrica; si efectivamente ya se recibieron muestras de la señal eléctrica se procede a almacenar en un archivo de tipo binario el registro que contiene la señal eléctrica y las características de la gráfica. La declaración de este registro es el siguiente:

```
Type Señal ' Se define el tipo Señal para poder almacenar en un
archivo
    ' la señal eléctrica muestreada.
Muestra(1000) As Integer ' Muestras capturadas de la Señal Eléct.
TotalMuest As Integer ' Contador de Muestras Recibidas.
Formato As String * 3 ' Formato de los Archivos "*.osc".
EA As Integer ' Indice de Escala Canal A.
PEA As Integer ' Indice de Posición Divisional del Canal A.
PDA As Integer ' Indice de Posición Subdivisional del Canal A.
EB As Integer ' Indice de Escala Canal B.
PEB As Integer ' Indice de Posición Divisional del Canal B.
PDB As Integer ' Indice de Posición Subdivisional del Canal B.
ET As Integer ' Indice de Escala Tiempo.
PET As Integer ' Indice de Posición Divisional del Tiempo.
PDT As Integer ' Indice de Posición Subdivisional del Tiempo.
NST As Integer ' Indice de Signo del Nivel del Trigger.
NET As Integer ' Indice de Nivel Entero del Trigger.
NDT As Integer ' Indice de Nivel Decimal del Trigger.
End Type
```

En este registro están contenidos la información del número de muestras recibidas, la palabra (compuesta del Byte de Dato y del Byte de Información), el formato del archivo, las posiciones de las líneas de referencia, y los índices de los controles de Escala de Voltaje (para las Ventanas de Canal A o Canal B), Escala de Tiempo (para la Ventana de Tiempo) y Nivel de Disparo (para la Ventana de Trigger).

En caso de que se haya presionado el Botón de Salir, se activa la bandera de salida como verdadera, para indicar al control del programa (contenido en el procedimiento *Recepción*), que el programa debe finalizar.

Una vez identificado qué botón se presionó y la Ventana de Comandos está maximizada, se desaparece el Cuadro de Selección Cyan, y se asigna $Func = 0$ para indicar ningún botón de la Ventana de Comandos tiene a su alrededor el Cuadro de Selección Cyan. Además se minimiza dicha ventana y se regresa a su posición original a través del procedimiento *ReDefineVentana*.

9.3.4 Graficación de la Señal

Para lograr la graficación de las muestras transmitidas inalámbricamente de la señal eléctrica se desarrolló un procedimiento para ubicar un punto correspondiente a la muestra en la pantalla, o para unir por medio de una línea a dos puntos correspondientes a dos muestras transmitidas; entonces este procedimiento debe de recibir tres parámetros. El primer parámetro corresponde a la forma en la que se encuentra la gráticula donde se va a graficar, ya que como se recordará existen dos gráticulas, una es para la impresión y otra para la graficación. El segundo parámetro es

la muestra transmitida (*Valor*), la cual es una variable de tipo Long, ya que ésta contiene el Byte de Dato y el Byte de Información. El tercer parámetro (*Tipo*) es para identificar si se va a almacenar o no, en un arreglo, la palabra transmitida, ya que no siempre se va a graficar por primera vez una gráfica, sino que posiblemente se tenga que redibujar la gráfica.

Lo primero que se hace en el procedimiento es refrescar el dibujo realizado en la graticula, cada vez que se esté dibujando la gráfica sobre cada división o media división de la graticula. El código se muestra a continuación:

```
If (MuestrasGrafic Mod (ResoluciónX \ 2)) = 0 Then
    ' Si es el la mitad de una División.
    Pantalla.Refresh ' Actualiza el la gráfica de la Señal Eléc.
End If
```

El siguiente paso es identificar si se quiere o no almacenar en un arreglo (del cual es donde se saca cada muestra a graficar) las muestras de la Señal Eléctrica recibida.

```
If Tipo Then ' Si se quiere almacenar la Señal muestreada.
    SR.Muestra(MuestrasGrafic) = Valor - 32768
    SR.TotalMuest = SR.TotalMuest + 1 ' Muestras recibidas
End If
```

Posteriormente se procede a determinar cuáles van a ser los puntos de referencia X y Y (dependiendo de las Línea de Referencia del Canal A o B y la del Tiempo), es decir a partir de dónde se van a dibujar de la gráfica. Las siguientes líneas muestran la determinación de los puntos de referencia para dibujar una señal capturada por el Canal A.

```
XIni = LinPosición(3).X1 - ((ResoluciónX * 10) / 2)
YIni = LinPosición(1).Y1
```

Lo siguiente es separar en una variable de dos bytes la palabra recibida, para posteriormente obtener el valor del voltaje absoluto (Byte de Dato), el signo, la Escala de Voltaje y la Escala de Periodo (estos tres contenidos en el Byte de Información). A continuación se muestra el código para hacer esto.

```
Vallar = SR.Muestra(MuestrasGrafic) + 32768 ' Suma Mantisa de Negat.  
VolAbs = Vallar \ &H100 ' Voltaje Absoluto.  
If (Vallar And &H40) = &H40 Then ' Signo de la muestra.  
    Signo = -1 ' Negativo  
Else  
    Signo = 1 ' Positivo    End If  
EscVol = 10 ^ (((Vallar And &H38) \ &H8) - 3) ' Escala de Voltaje.  
EscPer = (ResoluciónX / 10) ^ (((Vallar And &H6) \ &H2) - 2)  
    ' Escala del Periodo.
```

Después se dibuja la muestra en la grátícula, determinando si se trata de un punto inicial o la continuación de otras muestras ya dibujadas. Por último se incrementa la variable global con la que se lleva el conteo de las muestras graficadas.

```
If MuestrasGrafic = 0 Then ' Si es la primer muestra.  
    Forma.Pantalla.CurrentX = XIni + (EscPer) ' Punto inicial en X.  
    Forma.Pantalla.CurrentY = YIni - (VolAbs * EscVol * Signo)  
' Punto ini. Y.  
Else ' Si no es la primer muestra.  
    Forma.Pantalla.Line -(XIni + (MuestrasGrafic * EscPer), YIni -  
(VolAbs * EscVol * Signo))  
End If  
MuestrasGrafic = MuestrasGrafic + 1 ' Muestras Graficadas.
```

Capítulo 10:

Resultados

El Módulo de Osciloscopio se compone de cuatro sistemas: el sistema acondicionador de la señal eléctrica, el sistema de detección y corrección de errores, el sistema de transmisión y el sistema receptor. Para el sistema de detección y corrección de errores y el sistema de transmisión, la utilización del MCU68HC11F1 es muy importante, ya que es donde se realizan todos los procesos de conversión A/D, la modulación y la codificación, además nos permite un ahorro de espacio y peso, lo cual hace que nuestra punta de prueba sea lo suficientemente ligera.

El sistema acondicionador de la señal eléctrica a su vez se compone de un subsistema de auto-rango, el cual se conforma de un circuito de auto escalamiento para la adquisición de la señal eléctrica. Este último está constituido por un circuito que detecta el nivel de voltaje y una red atenuadora/amplificadora la cual maneja la ganancia del voltaje.

Durante el desarrollo del trabajo se hicieron pruebas de funcionalidad de la red atenuadora, donde se observó que se tenía un voltaje de offset mínimo, el cual fue compensado a través de una fase de auto-cero, además del ajuste previo del offset en el amplificador de instrumentación CA3140. En el rectificador de onda completa de

precisión (también llamado circuito de valor absoluto) se usaron amplificadores operacionales tipo CMOS (LMC660) de instrumentación, para tener una mejor precisión.

En el caso de los comparadores la respuesta de los amplificadores operacionales es buena, ya que se utilizaron amplificadores JFET (TL074), y la señal de comparación es el voltaje de DC proveniente de la señal rectificada, evitando de esta forma la inestabilidad del sistema, al hacer la adquisición de la señal eléctrica.

Las pruebas en la red atenuadora/amplificadora hechas para señales de DC fueron satisfactorias, dando la ganancia o atenuación necesaria para entregarla al convertidor A/D. En dichas pruebas hechas con señales de AC, se pudo observar un funcionamiento estable del sistema, aunque para voltajes muy pequeños, (del orden de 0.1 mV) no se presentaba muy clara la señal.

Después de varias pruebas con el circuito completo del sistema acondicionador de la señal eléctrica, en la fase de auto-cero implementada, se observó que era también viable hacer un ajuste del offset para los circuitos que lo necesitaban, como en los amplificadores operacionales y específicamente el CA3140 que maneja el escalamiento de la señal eléctrica, por lo cual fue necesario hacer un reajuste del offset.

El sistema de detección y corrección de errores es transparente para el usuario, ya que no es una parte física del Módulo de Osciloscopio. Este sistema se implementó para asegurar que la información transmitida fuera lo más inmune posible al ruido. Este sistema se divide en tres procesos: la codificación, la

corrección/detección de errores y la decodificación. El primero de ellos se ejecuta antes de hacer la transmisión de la señal y los otros dos después de la recepción de la señal. El programa que hace el proceso de codificación se encuentra almacenado en el microcontrolador de la punta inalámbrica (MCU68HC11F1) y el procedimiento que hace los otros dos procesos se encuentra dentro del programa Modulo de Osciloscopio.

Este sistema de detección y corrección de errores se constituye por un código BCH binario, debido a que estos códigos son los más utilizados dentro de las comunicaciones digitales, por lo tanto los más conocidos, derivando así, que exista mucha documentación acerca de ellos, además de algoritmos que nos permiten llegar a una detección y corrección de errores más rápida. El algoritmo Euclidiano junto con la búsqueda de Chien permiten detectar y corregir errores rápidamente en la información, aunque es notable el retardo que genera la ejecución de estos procesos en nuestro sistema.

Para la selección del código binario BCH que se utilizaría primero se probó con códigos sobre $GF(2^5)$. Al no encontrar un código que nos permitiera corregir un número considerable de errores por palabra, cuando se tenía una longitud de 16 bits, se optó por probar con los códigos BCH sobre $GF(2^6)$. De esta forma se seleccionó el código binario BCH (63,16,23), puesto que este código nos da la capacidad de corregir hasta 11 errores por palabra, además se consideró que de esta forma se lograba que la información fuera confiable.

Aunque la detección y corrección de errores alentan el procesamiento de la información, se decidió implementarlo al Módulo de Osciloscopio ya que es un instrumento de medición y debe ser lo más exacto y confiable posible.

El sistema transmisor se conforma por el Comparador de Salida OC2 del MCU68HC11F1 (con el que se genera una portadora de 40 KHz), el Puerto Serie del MCU68HC11F1 (donde sale la información procesada a transmitir) y los ILEDs con sus respectivos controladores de voltaje y corriente. Al estar haciendo pruebas de transmisión con los ILEDs, se observó que estos eran muy direccionales, pero de muy buen alcance, ya que su espectro de radiación así era. Ningún ILED probado, cumplía con lo que queríamos, un alcance de por lo menos 5 m. y una amplitud de radiación de por lo menos 60°.

Cabe señalar que una peculiaridad que tiene el Módulo de Osciloscopio, es que se puede cambiar la transmisión inalámbrica por una transmisión alámbrica, aunque esto no se haya planteado como objetivo del trabajo, pero da una seguridad aun mayor en la transmisión.

Por ultimo tenemos el sistema receptor, el cual está constituido por el receptor de luz infrarroja GP1U52X (que recibe y demodula la señal transmitida), un MAX232 (que acopla el voltaje TTL/CMOS del GP1U52X a los voltajes de ± 9 V. del puerto serie de la PC) y por supuesto la PC. La PC se encarga de ejecutar nuestra aplicación gráfica que es la interface entre el usuario y la punta de prueba inalámbrica. Esta interface corre bajo ambiente Microsoft Windows, permitiendo así que su uso sea muy fácil. La aplicación gráfica puede ser operada con la punta inalámbrica o con el

teclado y el Mouse de la PC. Para navegar por las ventanas de la aplicación por medio de la punta inalámbrica se utiliza tres teclas que tiene la punta, las cuales se combinan entre ellas para hacer diferentes operaciones, esto nos permite tener pocas teclas y no utilizar tanto espacio.

Cuando se presionan las tres teclas simultáneamente, la punta inalámbrica comienza a capturar una señal eléctrica y la transmite, mientras que en la aplicación se maximiza la Ventana de Pantalla donde se encuentra la gráticula y en donde se va a graficar la señal que se está capturando. En el momento en que la aplicación recibe la orden de que va a recibir información, procede a ejecutar la detección y corrección de errores, para posteriormente hacer la decodificación. Una vez hecha la decodificación, verifica que el dato decodificado sea un dato válido ya que podría ser ruido. Por último determina si el dato enviado es una muestra de la señal eléctrica y así graficarla en la gráticula. Este proceso se ejecuta hasta que termina de recibir información que representa la señal eléctrica capturada.

Existen algunas consideraciones que no son satisfactorias como fueron planeadas en un principio, pero son susceptibles de ser mejorados:

- ♦ El sistema de acondicionamiento funciona sin problemas cuando procesa señales de muy baja frecuencia o de DC, ya que el sistema queda limitado al ancho de banda de los circuitos lineales (1 MHz), como los amplificadores operacionales; además la inmunidad al ruido de estos no es muy buena. No obstante, este circuito puede operar a una frecuencia de 54 KHz (ancho de banda del Sistema Adecuador) sin provocar problemas de inestabilidad en el sistema, ni problemas de offset.

- ♦ El transmisor empleado con 8 ILEDs es muy direccional, es decir que solo apuntando adecuadamente al receptor la información se transmite correctamente.

Consideramos que si se emplea un arreglo de 16 ILEDs se puede corregir en buena parte este problema, pero ahora se tendría el inconveniente de que tal vez el transmisor crezca en tamaño.

- ♦ Cuando se tienen que procesar muchas muestras transmitidas de la captura de una señal eléctrica, el proceso de decodificación empleado hace que se alente un poco la graficación de dicha señal.

- ♦ Existen problemas al recibir la señal por el Puerto Serie con la versión 4.0 de Microsoft Visual Basic, mientras que con la versión 3.0 no hubo problemas, por lo que se hicieron dos programas, uno para cada versión de Microsoft Visual Basic.

En este capítulo se muestran los resultados de la implementación de los programas de captura y procesamiento de señales eléctricas. Se muestran los resultados de la implementación de los programas de captura y procesamiento de señales eléctricas. Se muestran los resultados de la implementación de los programas de captura y procesamiento de señales eléctricas.

En este capítulo se muestran los resultados de la implementación de los programas de captura y procesamiento de señales eléctricas. Se muestran los resultados de la implementación de los programas de captura y procesamiento de señales eléctricas. Se muestran los resultados de la implementación de los programas de captura y procesamiento de señales eléctricas.

Capítulo 11: Conclusiones

Consideramos que el objetivo principal del presente trabajo se cumplió, el cual era construir una punta inalámbrica que pudiera capturar y transmitir señales eléctricas para que en una computadora PC se graficara y se pudiera analizar con detalle, incluso se pudiera imprimir y almacenar en un archivo. Además se logró esto con un bajo costo, considerando que ya se tiene acceso a una computadora con facilidad.

Como en todas las primeras versiones de los sistemas, siempre hay que hacer mejoras, y no tanto porque no funcionen. Cabe señalar que si en el presente trabajo se hacen estas mejoras, habría que romper con uno de los objetivos primordiales, el cual es que este sistema debe ser barato en su construcción. Estas mejoras implicarían utilizar algunos circuitos integrados sofisticados que lo encarecerían, no obstante, se ha considerado que el Módulo de Osciloscopio requiere de algunas, como son:

- ◆ Cambiar el codificador-decodificador hecho por software a uno hecho por hardware (utilizado para garantizar la integridad de los datos), o bien hacer una rutina en lenguaje ensamblador (la cual sea llamada por el programa del Módulo de Osciloscopio), o bien utilizar tablas auxiliares para que la realización

de las operaciones de multiplicación y suma del campo sobre el que opera el código, sea más rápido en el proceso de detección de errores.

- ◆ Implementar un circuito con un convertidor A/D de alta velocidad, del tipo de convertidores en paralelo (flash converter), de tal forma que se pudiera hacer más rápida la comparación de la señal analógica de entrada, y por lo tanto, hacer la adquisición de señales de alta frecuencia sin problemas de inestabilidad en el sistema o ruido en la señal.
- ◆ Hacer un transmisor con señal de radio, utilizando señales a diferente frecuencia o amplitud, para manejar mas de un canal en el programa Módulo de Osciloscopio.
- ◆ Cambiar la forma de introducir la información a la PC, ya que si se quiere hacer una Módulo de Osciloscopio de "n" canales habría que utilizar "n" Puertos Seriales y si se quiere seguir utilizando un transmisor hecho con ILEDs, se tendría que utilizar "n" receptores-demoduladores de luz infrarroja. Para este caso la solución también sería utilizar señal de radio, para enviar inalámbricamente varias señales a diferente frecuencia. Este cambio implicaría utilizar el Puerto Paralelo, ya que sólo así pueden introducirse distintas señales seriales al mismo tiempo. Pero si se llegara a ocupar el puerto paralelo, se tendrían dificultades cuando se quisiera imprimir, ya que el puerto paralelo estaría ya ocupado por el receptor-demodulador.
- ◆ Utilizar Visual Basic como Front-End, mientras que los procesos de interpretación, decodificación y detección/corrección de errores de la palabra recibida se harían en lenguaje C o ensamblador.

Apéndice A:

Puertos de la

Computadora Personal

Los puertos de comunicación de las computadoras personales hacen que éstas se comuniquen con el mundo exterior a través de dichos puertos, en otras palabras estos puertos permiten la entrada y salida de datos. Los puertos están integrados en la unidad de sistema. Los datos que salen y entran de la computadora son manejados por el procesador.

Los puertos más comunes de una computadora son cinco: el puerto de teclado, el puerto de video, el puerto de ratón, el puerto paralelo y el puerto serie. A continuación se dará una breve información de los puertos Serial y Paralelo de la computadora personal.

A.1 Puerto Paralelo

El puerto paralelo está diseñado para pasar datos en grupos de ocho bits. Actualmente es la conexión más común para la impresora.

La interfaz original contaba con un contacto de 36 patas, ahora la interfaz emplea un contacto de 25 patas con escudo D (DB25). El nuevo contacto tiene menos conductores de tierra, lo cual no afecta para nada a la operación de la interfaz.

Este puerto es llamado paralelo porque la información pasa a través de cables paralelo, es decir, los ocho bits de información se desplazan juntos, mientras que por el puerto serie cada bit de información, viaja uno por uno, es decir bit a bit. La Figura A.1 muestra el contacto de cable DB25 del puerto paralelo.

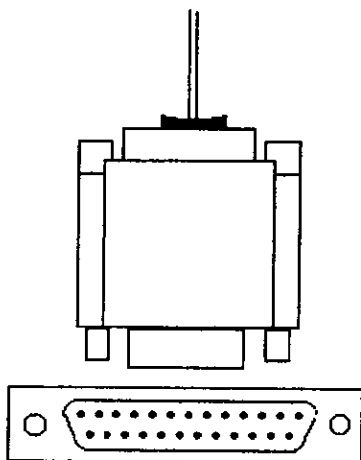


Figura A.1.- Contacto de cable DB25 del Puerto Paralelo

Debido a que la mayoría de las veces el puerto paralelo se emplea para conectar la impresora a la computadora, las señales que se citan en la Tabla A.1, son las que envía o recibe la computadora de la impresora. El puerto paralelo también puede emplearse para aceptar información de algún aparato externo.

Nombre de la señal	Pin	Nombre de la señal	Pin
-STROBE	1	+ Select	13
Dato 0	2	- Auto FDXT	14
Dato 1	3	-Error	15
Dato 2	4	-Init	16
Dato 3	5	-Slotin	17
Dato 4	6	Ground (Tierra)	18
Dato 5	7	Ground (Tierra)	19
Dato 6	8	Ground (Tierra)	20
Dato 7	9	Ground (Tierra)	21
-ACK (Reconocimiento: enterada)	10	Ground (Tierra)	22
Busy (ocupada)	11	Ground (Tierra)	23
Paper (sin papel)	12	Ground (Tierra)	24
		Ground (Tierra)	25

Tabla A.1.- Conexiones de los pines del Puerto Paralelo

A continuación se describe cada pin de conexión del puerto paralelo:

Líneas de Tierra: Son ocho porque a cada línea de información le corresponde una línea de tierra.

Señal STROBE: Indica a la impresora que el flujo de información está completo y que puede imprimir un caracter. El signo negativo de la señal significa que el pulso de STROBE es negativo. La señal del STROBE baja sucede cuando la impresora termina de enviar un byte de información.

Señales de Información: Son los ocho bits de un byte de información.

Señal ACK (Reconocimiento; enterada): indica a la computadora que la impresora ya esta lista para recibir más información. Mientras la señal ACK sea un nivel de voltaje alto, la computadora no envía nueva información. Cuando el nivel de voltaje es bajo, la computadora manda más información hacia la impresora.

Señal Busy (ocupado): indica a la computadora que está ocupada. La computadora espera a que el primer buffer se vacíe para enviar más información.

Señal Paper (sin papel): le indica a la computadora que no envíe más información, porque la impresora no está lista para recibirla.

Señal +Select: informa a la computadora si la impresora está en línea o no. Cuando la impresora no está en línea, no puede recibir caracteres de la computadora.

Señal -Auto FDXT: controla la manera en que la impresora maneja una nueva línea. La impresora puede adelantar la cabeza a la siguiente línea cuando regresa el carro. Cuando la computadora envía una señal de voltaje baja, la impresora agrega un alimentador de línea al carácter para que regrese el carro. Cuando la impresora recibe un nivel de voltaje bajo hace un retorno de carro.

Señal -Error: Indica cualquier otro error de la impresora.

Señal -Init: Se envía por la computadora para controlar la impresora, por medio de esta señal la computadora restablece los parámetros originales de la impresora, con la finalidad de que la configuración del último programa se aplique el siguiente trabajo de impresión.

Señal -Slctin (Select Input): Indica a la computadora si la impresora está lista para aceptar información (en línea o fuera de línea). Cuando esta señal baja, la impresora puede aceptar información. Cuando está alta, significa que la impresora no está en línea y no puede aceptar información.

Las conexiones en paralelo tienen la gran ventaja de ser más rápidas, debido a que pueden mandar varios bits de información al mismo tiempo; teniendo un alcance máximo de 3 a 4 metros. Teóricamente el puerto paralelo puede enviar medio millón de caracteres por segundo a través del cable. Una impresora puede seguir este ritmo durante un periodo corto de tiempo, después se rezaga y es entonces cuando la computadora detiene la transmisión ya que la línea de ocupado está encendida.

A.2 Puerto Serie

El puerto serie es un puerto de muchas aplicaciones en el cual se pueden conectar diversos dispositivos. Este puerto pasa sólo un bit de datos en un momento determinado. Algunas veces este puerto también es llamado puerto RS-232, haciendo referencia al número de especificaciones técnicas que definió primero a las interfaces seriales. Existe una gran variedad de periféricos diseñados para ser conectados por el puerto serie, siendo uno de los más comunes el módem. Existen todavía computadoras en donde se conecta el ratón por el puerto serie. El puerto serie es un puerto de comunicación estándar, la mayoría de las computadoras personales cuentan con dos de estos puertos.

Como su nombre lo indica por medio del puerto serie la información fluye bit a bit. Esto hace que la información viaje más despacio, pero permite comunicación a mayor distancia. Es posible emplearse una línea serial de 15 metros o más sin que se afecte la comunicación.

El serial estándar RS-232 requiere 25 líneas, pero con este estándar se pueden usar menos líneas. En la siguiente Figura A.2 se muestra el contacto de 9 pines con concha tipo D, que suele usarse actualmente para el puerto COMM 1 en la mayoría de las PC. Con el puerto serial COMM 2 se emplea el contacto DB 25, que es más antiguo. Sin embargo no tiene conectadas los 25 pines por completo.

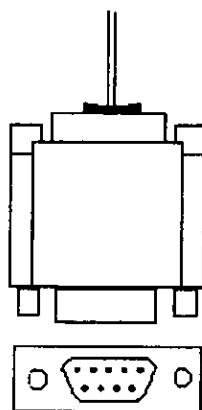


Figura A.2.- Contacto Serial de 9 pines con concha tipo D

Las líneas más usadas de los contactos DB 25 y DB 9 son los que se muestran en la Tabla A.2.

Señal	Nombre	Pin DB9	Pin DB25
DCD	Data Carrier Detect	1	8
RX	Receive Data	2	3
TX	Transmit Data	3	2
DTR	Data Terminal Ready	4	20
GND	Signal Ground	5	7
DSR	Data Set Ready	6	6
RTS	Request to Send	7	4
CTS	Clear to Send	8	5
IR	Ring Indicator	9	22

Tabla A.2.- Conexiones de los pines del Puerto Serie

A continuación se describe cada Línea del Puerto Serie:

Línea DCD (Data Carrier Detect): En ésta línea se envía una señal que indica que habrá comunicación entre sí.

Línea RX (Receive Data): A través de ésta línea se recibe la información que fue enviada remotamente por un DTE (Data Terminal Equipment / Equipo Terminal de Datos), por ejemplo la computadora, o un DCE (Data Communications Equipment / Equipo de Comunicación de Datos), por ejemplo un modem.

Línea TX (Transmit Data): Esta línea es el canal por el cual la computadora o el DCE remoto transmite la información. Existen dos líneas de información una es para enviar y otra para transmitir información.

Línea DTR (Data Terminal Ready): Esta línea lleva una señal positiva desde el dispositivo DTE hasta el dispositivo remoto, para indicar que hay un dispositivo DTE conectado al cable serial. Junto con esta línea va la línea DSR. En el caso de que dos dispositivos se estén comunicando, estas dos líneas deben estar altas, cada una indicando a la otra que está lista para la comunicación.

Línea GND (Signal Ground): Esta es línea que tiene la señal de tierra.

Línea DSR (Data Set Ready): Esta es la otra parte del DTR. El DSR lleva la señal de un DCE que le indica al dispositivo DTR que está en línea y está lista para comunicarse.

Línea RTS (Request to Send): Esta es la línea que lleva la señal que permite a dos equipos conectados entre sí, poderse comunicar, cuando el Receptor esté listo para

recibir información. El DTE controla a la línea RTS, mientras que el DCE controla a la línea CTS.

Línea CTS (Clear to Send): Esta línea es la pareja de la línea RTS. En muchos enlaces de comunicaciones, la información no fluye a través de la conexión serial a menos que se tenga un nivel de voltaje alto en estas dos líneas, pero hay excepciones, como por ejemplo cuando se conecta a la computadora otra computadora remota, donde en dicho enlace se pueden usar o no las líneas CTS y RTS.

Línea RI (Ring Indicator): Esta línea, en el caso del módem, le indica a la computadora que acaba de entrar una llamada. Esto permite que el software de comunicación sepa qué tiene que hacer una conexión de módem.

Cuando un cable se conecta a un DTE, en este caso la computadora, se supone que el otro extremo se conectará a un DCE, podría ser un módem. Un DCE tiene diferentes asignaciones para las pines seriales, de manera que la línea TX del DTE se conecta automáticamente al pin RX de un DCE. De manera similar, la línea TX del DCE se une al pin RX del DTE.

La línea RTS de un DTE debe conectarse a la línea CTS del DCE, ya que la línea de solicitud de enviar (RTS: Request to Send line) debe ser contestada como Listo para poder enviar (CTS: Clear to Send). Las líneas DTR y DSR también deben conectarse entre sí, y la línea DCD de un DTE por lo general se conecta a la línea DTR de un DCE.

La conexión cruzada se efectúa cuando se usa un aparato DTE y un DCE. Para el caso de conectar dos DTE, como el de unir dos computadoras a través de la línea serial para intercambiar archivos, se debe utilizar un cable nulo de módem, es decir un cable que no efectúe la conexión cruzada. En la Figura A.3 se muestra cómo está trenzado este cable.

El número de conexiones que se utilizan, dependen del uso que se le da a la línea serial. Hace tiempo, la línea serial solía ser de tres cables: RX, TX y tierra. Actualmente, el uso más común del puerto serie se utiliza principalmente para conectar la computadora a un módem o a otro equipo remoto. Como no siempre se sabe qué aparato se conectará, se acostumbra conectar por lo menos las líneas que se muestran en la Tabla A.2. Con estas nueve líneas se pueden conectar un módem, una impresora serial, un graficador y otros aparatos seriales comunes.

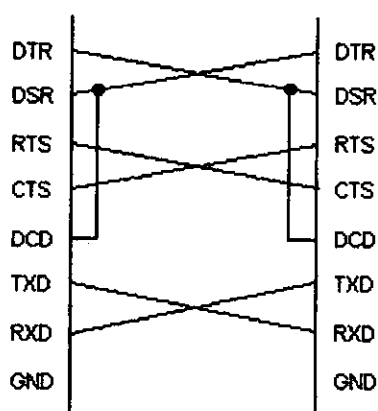


Figura A.3.- Conexión cruzada del Puerto Serie

Para que se pueda establecer una comunicación entre un dispositivo externo y la computadora, a través del puerto serie, se requiere de un enlace de comunicaciones y un programa de soporte de comunicaciones. El puerto serie se comunica con un dispositivo externo en un formato de comunicaciones asíncronas, esto elimina la necesidad de una señal de reloj común entre los dos dispositivos. Este formato permite que la comunicación sea muy sencilla.

Apéndice B:

Simuladores Detectores y Correctores de Errores

Para mostrar como es que se puede implementar el esquema de detección y corrección de errores utilizando códigos BCH, se desarrollaron dos simuladores.

El primer simulador a partir de un vector dado u codifica la información, después da la opción de introducir errores, los cuales por medio del esquema de detección y corrección de errores son corregidos. Este simulador trabaja sobre un $GF(2^5)$, para diferenciarlo del otro lo llamaremos Simulador $GF(2^5)$.

El segundo simulador contiene los esquemas de detección y corrección de errores que se utilizaron en el módulo de osciloscopio. Este simulador se desarrolló con la finalidad de ver que los esquemas estuvieran funcionando correctamente antes de implementarlos al módulo de osciloscopio. Como las rutinas de detección y corrección son las mismas que se utilizaron en el módulo de osciloscopio ya fueron explicadas en el capítulo de desarrollo, por lo tanto, en esta sección ya no se profundizará en ella. Al segundo simulador lo llamaremos simulador $GF(2^6)$ por que el código BCH que utiliza trabaja sobre un $GF(2^6)$.

B.1 Simulador $GF(2^5)$

B.1.1 Diseño del código BCH para el Simulador $GF(2^5)$

Se busca que el código corrija el mayor número de errores posibles en una palabra sin codificar de 8 bits. Se escogió trabajar en un campo de $GF(2^5)$, ya que este campo nos permite tener una longitud de 31 bits y esto nos da mayor capacidad en la corrección de errores. No fue conveniente desarrollar el código BCH sobre $GF(2^4)$ puesto que no nos permitía corregir aproximadamente un cincuenta por ciento, si se requiere de una dimensión de ocho bits. La Tabla B.1 muestra los polinomios mínimos de algunos elementos de $GF(2^5)$.

Elementos	Polinomio Mínimo
$\alpha, \alpha^2, \alpha^4, \alpha^8, \alpha^{16}$	$M_1 = x^5 + x^2 + 1$
$\alpha^3, \alpha^6, \alpha^{12}$	$M_2 = x^5 + x^4 + x^3 + x^2 + 1$
$\alpha^5, \alpha^9, \alpha^{10}$	$M_3 = x^5 + x^4 + x^2 + x + 1$
α^7, α^{14}	$M_4 = x^5 + x^3 + x^2 + x + 1$
α^{11}, α^{13}	$M_5 = x^5 + x^4 + x^3 + x + 1$
α^{15}	$M_6 = x^5 + x^3 + 1$

Tabla B.1: Algunos Polinomios Mínimos de $GF(2^5)$

De la Tabla B.1 se deriva la Tabla B.2.

El código que más se aproxima a tener una dimensión de 8 bits es el código (31,11,11). La dimensión del código es más grande que la requerida, si se diseñara un código con dimensión más pequeña no cubriríamos nuestras necesidades. El polinomio generador es el siguiente:

$$G(x) = M_1(x)M_2(x)M_3(x)M_4(x)$$

El simulador se diseñó a partir del polinomio generador anterior y trabaja sobre un $GF(2^5)$.

Errores a Corregir	Distancia Diseñada δ	Polinomio Generador $G(x)$	Dimensión k
1	3	$M_1(x)$	26
2	5	$M_1(x)M_2(x)$	21
3	7	$M_1(x)M_2(x)M_3(x)$	16
4	9	$M_1(x)M_2(x)M_3(x)M_4(x)$	11
5	11	$M_1(x)M_2(x)M_3(x)M_4(x)$	11
6	13	$M_1(x)M_2(x)M_3(x)M_4(x)M_5(x)$	6
7	15	$M_1(x)M_2(x)M_3(x)M_4(x)M_5(x)$	6
8	17	$M_1(x)M_2(x)M_3(x)M_4(x)M_5(x)M_6(x)$	1

Longitud $n = 31$

Tabla B.2: Polinomios Generadores

B.1.2 Desarrollo del Simulador $GF(2^5)$

El simulador está dividido en tres bloques: el codificador, el detector/corrector de errores y el decodificador, como se muestra en la Figura B.1.

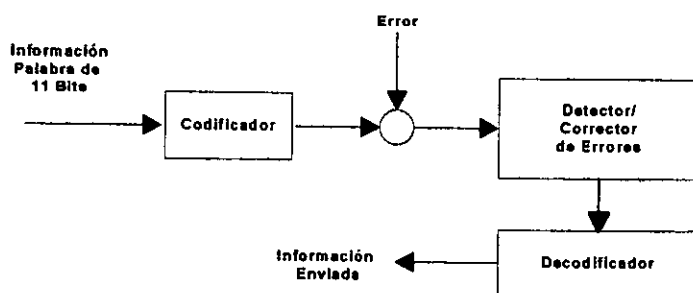


Figura B.1: Diagrama de Bloques del Simulador

B.1.2.1 Codificador

Como ya se había mencionado para codificar un vector u de información es necesario multiplicar su polinomio representativo por el polinomio generador.

Para hacer la multiplicación de dichos polinomios sólo es necesario guardar los coeficientes de ambos. El grado y la longitud de los polinomios es constante por lo tanto no es necesario almacenar estos valores.

Se debe multiplicar cada elemento de $u(x)$ por cada uno de los de $g(x)$, y después se deben sumar todos los coeficientes de la variable x del mismo grado.

El vector $u(x)$ es introducido por el usuario, más adelante se mostrará como es que el usuario lo introduce. El vector $g(x)$ es constante y está almacenado en el programa.

El programa del simulador se desarrolló en el lenguaje de programación Visual Basic, así que las rutinas que se presentarán estarán hechas en dicho lenguaje. La rutina que codifica la información se llama *BtnCdfcn* y consta de las siguientes líneas:

```
For i = 0 To 30
    c(i) = 0
Next i

For i = 1 To 11
    LblInfo(i).Enabled = False
    If LblInfo(i).Caption = "1" Then
        For j = 0 To 20
            c(i + j - 1) = c(i + j - 1) Xor g(j)
        Next j
    End If
Next i
```

```
End If  
Next i
```

En su primera parte la rutina inicializa con ceros el arreglo *c* que es en donde se guardará el código. Los bucles For utiliza el arreglo *c* y *g*, donde *g* es el arreglo que contiene los coeficientes del polinomio generador. El bucle exterior se efectúa once veces por que la dimensión es once y el interno se efectúa veintiún veces por que el grado del polinomio generador es veinte. El vector *u* es leído por el *Caption* del objeto *LblInfo* ya que ahí es donde el usuario pone la información que desea codificar.

Cuando existe un uno en el *Caption* de *LblInfo(i)* se hace un XOR con *g(i)*, ya que de lo contrario sería una multiplicación por cero y no alteraría lo que se tiene almacenado en $c(i + j - 1)$. *i* y *j* representan la potencia de la variable *x* en los polinomios *u(x)* y *g(x)* respectivamente, es por ello que se hace la suma de ellos para saber a qué potencia corresponde el producto y sumarlo en la posición correspondiente. Se les resta uno por que el vector *u* comienza en uno, si empezara en cero esta resta no sería necesaria, claro que el bucle iría de cero a diez. Note que aquí la información se procesa utilizando arreglos, mientras que en el módulo de Osciloscopio los coeficientes de los vectores se almacenaban en una sola variable de tipo entero.

Las siguientes instrucciones muestra dos arreglos de etiquetas, *LblCod* y *LblCw*, donde se muestra el código. *LblCw* tiene la opción de modificarse, esto con la finalidad de que el usuario pueda introducir errores.

```
Btncodfcn.Enabled = False  
LblEtCod.Visible = True
```

```
BtnDefcn.Enabled = True
For i = 0 To 30
    LblCod(i + 1).Visible = True
    LblCw(i + 1).Visible = True
    LblCw(i + 1).Caption = Format(c(i), "")
    LblCod(i + 1).Caption = Format(c(i), "")
Next i
LblEtPalRec.Visible = True
Btndcers.Enabled = True
```

B.1.2.2 Detector / Corrector de Errores

La rutina de Detección/Corrección de Errores utiliza varias funciones además de que necesita algunas constante y variables que deben estar definidas antes de ser ejecutada.

Como ya se había dicho todos los elementos de $GF(2^5)$ pueden representarse como potencias de α excepto el cero, es por ello que para representarlo se utiliza el exponente -1. A cada elemento se le asocia un vector binario de longitud cinco, esta información se genera con la siguiente rutina:

```
Private Sub Form_Load()
    a(1) = 1
    For i = 2 To (2 ^ 5)
        a(i) = a(i - 1) * 2
        If a(i) >= (2 ^ 5) Then a(i) = (a(i) - (2 ^ 5)) Xor 5
    Next i

    g(20) = 1: g(18) = 1: g(17) = 1: g(13) = 1: g(10) = 1
    g(9) = 1: g(7) = 1: g(6) = 1: g(4) = 1: g(2) = 1
    g(0) = 1
End Sub
```

Donde a $a(i)$ se le asigna el vector del elemento de $GF(2^5)$ que le corresponde a α^{i-1} . Observe que esta función es muy similar a la empleada en el

Módulo de Osciloscopio con sus variantes correspondientes. $g(i)$ contiene los coeficientes del polinomio $g(x)$.

El Detector/Corrector de Errores también utiliza la función suma que tiene dos parámetros de entrada, sum1 y sum2, al igual que la función anterior, esta función también es utilizada en el Módulo de Osciloscopio, con sus variantes correspondientes, por lo tanto no se detallará. La función es la siguiente:

```
Private Function suma(sum1 As Integer, sum2 As Integer) As Integer
If sum1 > 31 Then sum1 = sum1 Mod 31
If sum2 > 31 Then sum2 = sum2 Mod 31
If (sum1 = -1) And (sum2 = -1) Then
    suma = -1
Else
    If (sum1 = -1) Then
        suma = sum2
    Else
        If (sum2 = -1) Then
            suma = sum1
        Else
            aux = a(sum1 + 1) Xor a(sum2 + 1)
            If aux = 0 Then
                suma = -1
            Else
                For aa = 1 To 32
                    If aux = a(aa) Then
                        suma = aa - 1
                        Exit For
                    End If
                Next aa
            End If
        End If
    End If
End If
End Function
```

Leer la palabra recibida, calcular el polinomio del síndrome, desarrollo del algoritmo Euclideo y detectar y corregir errores, son los procedimientos que realiza el decodificador.

El procedimiento que se emplea para leer la palabra recibida es la siguiente:

```
BtnDefcn.Enabled = False
Rem Leer palabra recibida
For i = 1 To 31
    LblCw(i).Enabled = False
    If LblCw(i).Caption = "1" Then
        c(i - 1) = 1
    Else
        c(i - 1) = 0
    End If
Next i
```

note que al arreglo *c* se le asignan nuevos valores los cuales pueden ser los mismos o pueden estar alterados por el usuario. Los procedimientos utilizados para la detección y corrección de errores son iguales a los desarrollados en el Capítulo 10, por lo tanto en esta sección ya no se explicarán.

La última parte de la rutina de *Detección / Corrección de Errores* es la siguiente:

```
If num_err Then
    For i = 0 To 30
        LblPalCorr(i + 1).Visible = True
        LblPalCorr(i + 1).Caption =
            Format(c(i), "")
    Next i
    LblEtPalCorr.Visible = True
    LblEtPalCorr.Caption = "Palabra
        Corregida"
    BtnDefcn.Enabled = True
Else
    LblPalCorr(0).Visible = True
    LblPalCorr(0).Caption = "NO ES POSIBLE"
```

```

CORREGIR LA PALABRA RECIBIDA"
LblEtPalCorr.Visible = True
LblEtPalCorr.Caption = "Ocurrieron más
errores de los que se pueden corregir en la
transmisión"
BtnDcfcn.Enabled = False
End If
End If
Btndcers.Enabled = False
BtnIni0.Enabled = True
BtnIni1.Enabled = True
End Sub

```

en la cual si se detectaron errores visualiza la palabra corregida con el título de "Palabra Corregida" y habilita el botón de Decodificación el cual más adelante se muestra. En el caso de que no se detecte ningún error significa que ocurrieron más errores de los que se podían corregir ya que si entró a esta rutina quiere decir que si existen errores, por que el síndrome es diferente de cero, por lo tanto manda un mensaje advirtiéndolo. Se deshabilita el botón de decodificación ya que la decodificación si se efectúa en este caso será errónea. Y saliendo de esta sentencia el programa de *Detección /Corrección de Errores* termina.

B.1.2.3 Decodificación

Para decodificar la palabra recibida y corregida, si es el caso, se divide el polinomio asociado a la palabra recibida y corregida entre el polinomio generador. La subrutina empleada es *BtnDcfcn*, que es la siguiente:

```

Private Sub BtnDcfcn_Click()
Static u(11) As Integer
Dim pal As String
Rem decodificación
For i = 0 To 10
If c(i) = 1 Then
u(i) = 1
For j = 0 To 20

```



```
        c(i + j) = c(i + j) Xor g(j)
    Next j
Else
    u(i) = 0
End If
Next i
pal = ""
For i = 0 To 10
    LblPalDec(i + 1).Visible = True
    If u(i) Then LblPalDec(i + 1).Caption = 1
Else LblPalDec(i + 1).Caption = 0
Next i
BtnDcfcn.Enabled = False
LblEtPalDec.Visible = True
BtnIni0.Enabled = True
BtnIni1.Enabled = True
End Sub
```

En donde se utilizan 2 bucles *Fors* anidados, el externo es para recorrer los coeficientes de la palabra recibida y corregida y el interno es para recorrer los coeficientes del polinomio generador. Se utiliza otro bucle *for* para visualizar la palabra decodificada, además de que se hacen visibles otros botones de control necesarios.

En cuanto el programa regresa la palabra decodificada termina y es posible hacer otro proceso o salir de él.

B.1.3 Funcionamiento del Simulador

La primer ventana que presenta el simulador es la que se muestra en la Figura B.2, en donde los botones que dicen Inicializar con 0's e Inicializar con 1's sirven para poner en las casillas donde va la información ceros o unos respectivamente. Cuando se presiona el botón Inicializar con 0's todos los bits de información toman el valor de cero y cuando se presiona el botón Inicializar con 1's todos los bits de

información toman el valor de uno. En el caso de que no se quieran introducir sólo unos o sólo ceros, por medio del ratón se pueden cambiar los valores, presionando una vez con el botón izquierdo sobre el cuadro donde se desea modificar la información y en ese instante el bit se complementa, es decir si la casilla tiene un uno al apuntar con el ratón dentro de ella y hacer un clic con el botón izquierdo la casilla tendrá ahora un cero.

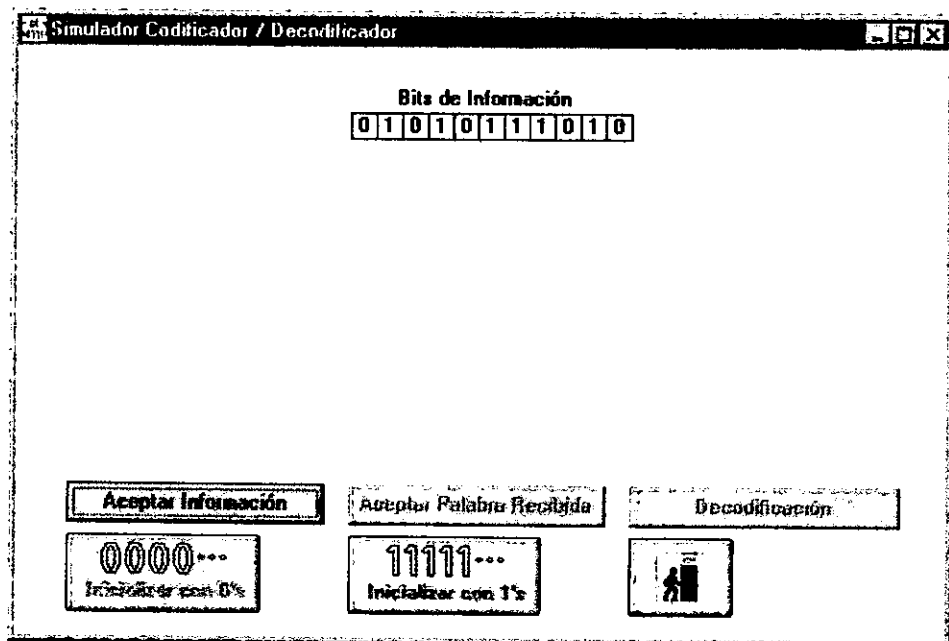


Figura B.2: Primer Ventana del Simulador GF(2⁵)

Cuando la información está lista para ser codificada se presiona el botón aceptar información, después de hacer esto el programa nos muestra la ventana que se muestra en la Figura B.3.

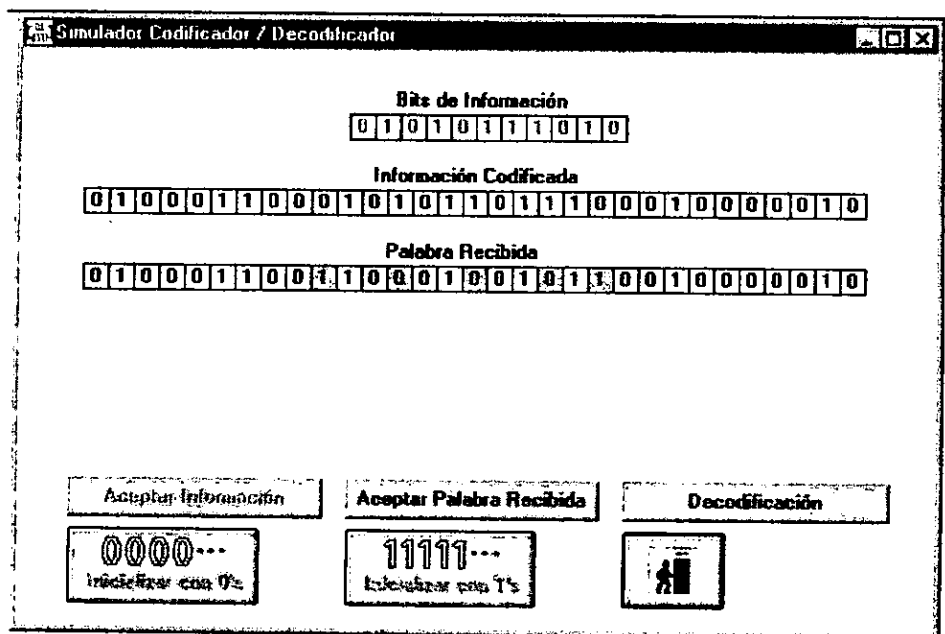


Figura B.3: Ventana que Muestra la Simulación de la Palabra Recibida

Como se puede observar nos aparece un arreglo de cuadros en donde tenemos la información codificada y otro en donde también tenemos la información codificada, pero éste tiene una etiqueta que dice Palabra Recibida, en este arreglo de cuadros el usuario puede modificar la información de tal forma que se simule que ocurrieron errores durante la transmisión. La forma de alterar este vector es igual a la de introducir la información, con el ratón apuntando en la casilla en donde se desea corregir el valor del bit se presiona el botón izquierdo una vez y la información es corregida al mismo tiempo que el fondo del cuadro se ilumina de rojo para resaltar el error. En caso de que se desee quitar el error el procedimiento es el mismo, al presionar el botón izquierdo del ratón apuntando en una casilla ya iluminada, la

información es corregida y la casilla vuelve a ser blanca. Después de editar la palabra recibida se presiona el botón que dice "Aceptar Palabra Recibida". Dentro de este botón se encuentra la subrutina *Btndcers*, la cual se encarga de detectar y corregir errores por medio del algoritmo Euclideano.

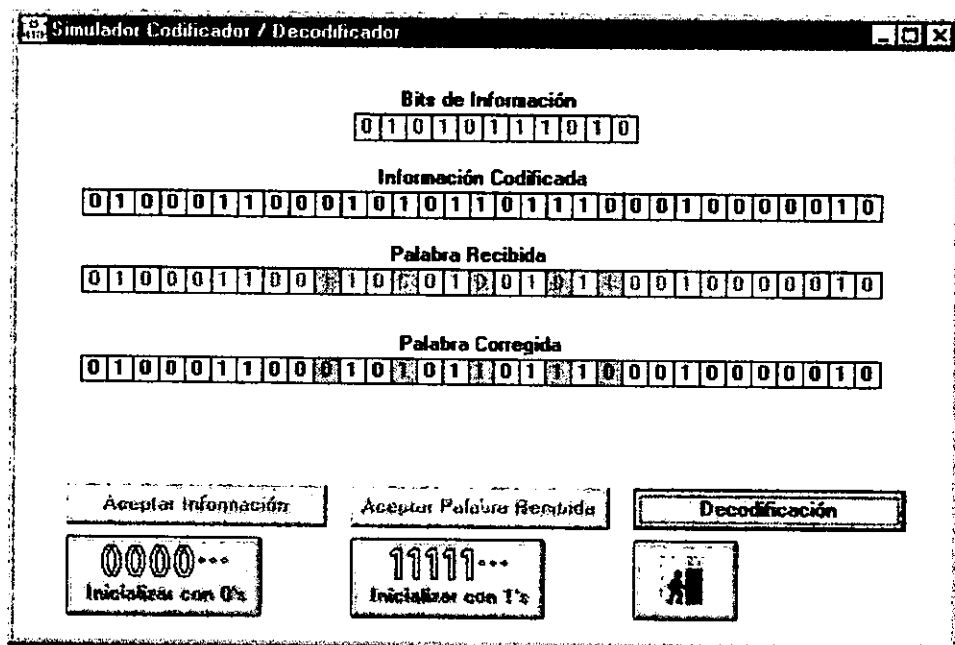


Figura B.4: Ventana que Muestra la Palabra Corregida

Al terminar de ejecutarse la rutina de detección y corrección de errores se nos muestra una ventana, la cual se muestra en la Figura B.4, en donde se observa la palabra corregida, para el ejemplo que estamos viendo como si ocurrieron errores se muestra el arreglo de cuadros con la palabra corregida, con un fondo obscuro

se resaltan los cuadros en donde se corrigieron los errores. En el caso de que no hubiera habido errores sólo se manda un mensaje de que no ocurrieron errores.

Después se habilitan los botones de *Decodificación*, *Inicializar con 1's* e *Inicializa con 0's*, es decir que el usuario tiene las opciones de decodificar la información corregida o en caso de que no le interese la decodificación puede iniciar nuevamente el programa ya sea inicializando la información con sólo unos o sólo ceros o salir del programa. El usuario puede salir del programa en cualquier momento que lo desee ya que este botón de *Salida* está habilitado siempre durante la ejecución del programa.

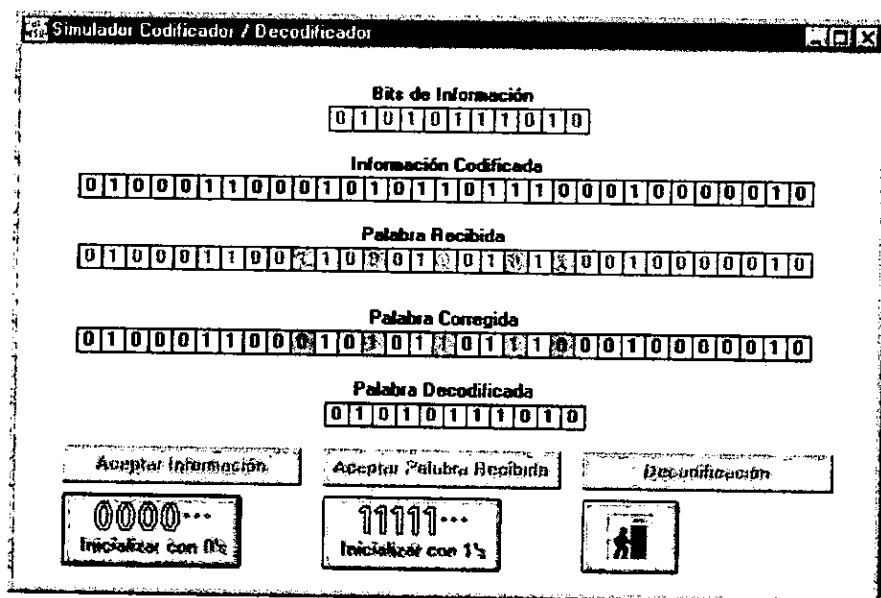


Figura B.5: Ventana que Muestra la Palabra Decodificada

El botón *Decodificación* contiene la rutina BtnDfcfn que se encarga de decodificar la información y mostrarla. Después de que se ejecuta la rutina BtnDfcfn el usuario puede volver a ejecutar el programa o salir de él. La ventana que se muestra después de ejecutar dicha rutina es la de la Figura B.5.

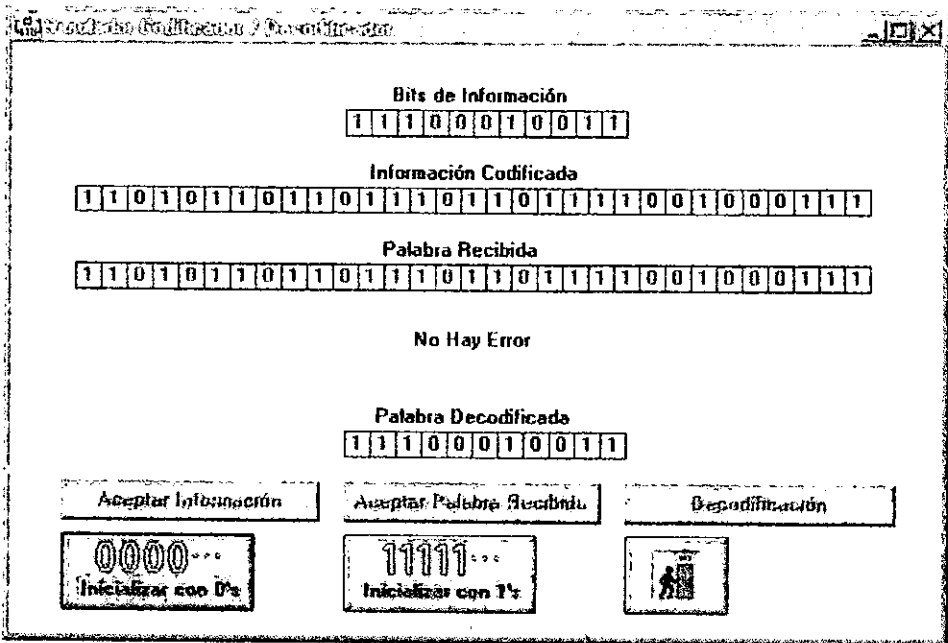


Figura B.6: Ventana que se Muestra Cuando no hay Error

En la Figura B.6 se muestra la pantalla de un ejemplo donde no se introducen errores, con la finalidad de mostrar qué hace el programa en el caso de no detectar errores. Se introdujo la información, después se codificó, no se editó la palabra recibida, es decir, que se simula que no ocurrieron errores durante la transmisión, al

aceptar la palabra recibida la rutina de Detección y Corrección de errores al no detectar ningún error manda el mensaje "No Hay Error" y por último se muestra la información decodificada que es la misma que se introdujo al principio.

Por medio del diseño de este simulador se observó cómo es posible adaptar el algoritmo Euclidiano para detectar y corregir errores utilizando un código BCH. En la siguiente sección se hace una implementación similar de un código BCH, pero que trabaja sobre otro GF , dicha implementación es la que se utilizó para la parte de codificación dentro del módulo de osciloscopio.

B.2 Simulador $GF(2^6)$

Como ya se había mencionado este segundo simulador se desarrolló con la finalidad de ver que los algoritmos de detección y corrección de errores estuvieran funcionando correctamente antes de implementarlos al módulo de osciloscopio. El diseño del código, como la mayoría de las rutinas no se incluirán en esta sección puesto que son las mismas que se explicaron en el capítulo de desarrollo.

Para el módulo de osciloscopio se diseñó un código BCH sobre $GF(2^6)$ con capacidad de corregir 11 errores. La longitud del código es $n = 63$ y la dimensión es de $k = 16$. La codificación es realizada con el microcontrolador MCU68HC11F1 y también con la PC, esto es con la finalidad de hacer una comparación y ver que la codificación se estuviera efectuando adecuadamente por medio del microcontrolador.

B.2.1 Desarrollo del Simulador

La rutina de codificación desarrollada en lenguaje ensamblado es la misma rutina explicada en el Capítulo 10.

B.2.1.1 Función *Btncdfcn*

Esta función se encarga de leer la información introducida por el usuario, codificarla, transmitir la información sin codificar al microcontrolador y recibir la palabra codificada por el microcontrolador y mostrarla. A continuación se desglosan cada uno de los procedimientos mencionados y se da una breve explicación de cada uno de ellos. Como puede observar se hacen dos codificaciones la que hace el microcontrolador y la que hace el programa, esto fue con la finalidad de comparar las palabras y observar si el programa en ensamblador estaba funcionando correctamente.

La definición de las variables locales utilizadas es la siguiente::

```
Dim num0 as Integer ' Byte menos significativo de la información.
Dim num1 as Integer ' Byte más significativo de la información.
Dim Recib(8) as Integer ' Palabra codificada por el MC68HC11F1.
```

B.2.1.2 Procedimiento que Lee la Información y la Codifica

El procedimiento que lee la información introducida por el usuario y la codifica es el siguiente:

```
Rem Codificación
  BtnIni1.Enabled = False
  BtnIni0.Enabled = False
  For i = 0 To 62
    c(i) = 0
  Next i
  num0 = 0: num1 = 0
  For i = 1 To 16
    LblInfo(i).Enabled = False
```



```

    If LblInfo(i).Caption = "1" Then
        If num0 + (2 ^ (16 - i)) > 255 Then
            num1 = num1 + (2 ^ (8 - i))
        Else
            num0 = num0 + (2 ^ (16 - i))
        End If
        For j = 0 To 47
            c(i + j - 1) = c(i + j - 1) Xor g(j)
        Next j
    End If
Next i

Btncdfcn.Enabled = False
LblEtCod.Visible = True
LblCod(i).Visible = True
For i = 0 To 62
    LblCod(i).Visible = True
    LblCod(i).Caption = Format(c(i), "")
Next i

```

En sus dos primeras líneas deshabilita los botones de *Inicializar con 1's* e *Inicializar con 0's*. El primer bucle *For* se encarga de inicializar el arreglo en donde se almacenará la palabra codificada. El siguiente bucle *For* se encarga de ir almacenando la información en dos bytes *num1* que es el más significativo y *num0* que es el menos significativo, es decir que *num0* contendrá los ocho bits menos significativos y *num1* contendrá los 8 bits más significativos. La información binaria que introduce el usuario con esta rutina se está almacenando en forma decimal en las variables *num1* y *num0*.

Dentro del segundo ciclo *For* existe un bucle *For* anidado, por medio de este se va realizando la codificación y se va almacenando en el arreglo *c*. El arreglo *g* contiene el vector generador de código. Las últimas líneas se encargan de poner en pantalla la palabra codificada.

B.2.1.3. Procedimiento que Transmite la Información

El procedimiento que transmite la información es el siguiente:

```
Btndcers.Enabled = True
Comm1.Output = Chr$(num1) + Chr$(num0)
```

La información se transmite en código ASCII, es decir, se transmiten los caracteres correspondientes en el código ASCII a los números que contiene las variables *num1* y *num0*.

B.2.1.4 Procedimiento que Recibe la Palabra Codificada

Las siguientes instrucciones se encargan de leer la palabra codificada por el MC68HC11F1, la cual es recibida por el puerto serie. En la variable *palab* se almacena la palabra codificada recibida, la cual se encuentra en código ASCII.

```
palab = ""
Do ' Verifica que el dato esté en el puerto serie
  Dummy = DoEvents()
  palab = palab + Comm1.Input ' Lee bytes acumulados en el puerto
                             ' serie
Loop Until ((Len(palab) Mod 8) = 0) And Len(palab)
```

La información recibida que se encuentra en código ASCII es convertida a decimal por medio de las siguientes instrucciones:

```
For i = 0 To 7
  recib(i) = Asc(Mid$(palab, i + 1, 1))
Next i
```

Por último se manda la información a pantalla, con las siguientes instrucciones:

```
For i = 1 To 63
  LblCw(i).Visible = True
Next i
```

Módulo de Osciloscopio

```
For i = 0 To 7
    word_corr = 1
    For j = 0 To 7
        If i < 7 Then
            LblCw(((i + 1) * 8) - j).Caption = IIf(recib(i) And
                word_corr, "1", "0")
            word_corr = word_corr * 2
        Else
            word_corr = word_corr * 2
            LblCw(63 - j).Caption = IIf(recib(i) And word_corr, "1",
                "0")
            If j = 6 Then Exit For
        End If
    Next j
Next i
LblEtRec_Micro.Visible = True
End Sub
```

El primer bucle hace visible el arreglo de etiquetas que muestran los bits de la palabra codificada por el MC68HC11F1. El siguiente bucle *For* se encarga de analizar el arreglo *recib*, para mandar unos o ceros según el análisis hecho. Se utiliza la variable *word_corr* para recorrer cada uno de los bits de cada elemento del arreglo *recib*. Al multiplicar por dos se hace un corrimiento hacia la izquierda, si se observa el número en binario y por medio de una AND se sabe si el bit en la posición $((i + 1) * 8) - j$ es un uno o un cero. El último byte de la palabra codificada es tratado de forma especial ya que su bit menos significativo no cuenta, puesto que la longitud de la palabra codificada es de 63 bits.

B.2.1.5 Función *Btndcers*

Btndcers es la función que lee la palabra recibida por el MC68HC11F1 y corregida por el usuario y de decodificar la información. Esta función ejecuta otra función llamada *Decod* que es la función que decodifica. La función *Decod* recibe un parámetro de tipo *string*, así que la función *Btndcers* antes de llamarla tiene

que transformar la palabra codificada a tipo *string*. La función *Btndecers* es la siguiente:

```
Private Sub Btndcers_Click()
Dim Dec As String, Num_Rec As Long
For i = 1 To 63
    LblCw(i).Enabled = False
Next i

palab = ""
For i = 0 To 7
    num0 = 0
    For j = 0 To 7
        If ((j + (8 * i)) + 1) < 64 Then
            If LblCw((j + (8 * i)) + 1).Caption = "1" Then num0 = num0 +
                (2 ^ (7 - j))
        End If
    Next j
    palab = palab + Chr$(num0)
Next i
Dec = Decod(palab)
End Sub
```

Lo que hace el segundo bucle *For* es leer la palabra recibida y modificada por el usuario de un arreglo de etiquetas, que después será mostrado. Cada bit leído es multiplicado por una potencia de dos, dependiendo de la posición de donde se encuentra, y adicionado a una variable llamada *num0*, para obtener el equivalente decimal. La palabra codificada es agrupada de 8 en 8 bits, para que forme bytes que serán almacenado en una variable de tipo cadena llamada *palab*. *palab* almacena el código ASCII correspondiente a cada número decimal de la palabra codificada.

B.2.2 Funcionamiento del Programa

Al ejecutarse el programa se muestra la pantalla de la Figura B.7. El arreglo de cuadros que está etiquetado como "Bits de Información" recibe la información que se desea codificar. La forma de introducir la información es igual a la

del simulador anterior, teniéndose también las opciones de inicializar la información con ceros o con unos. Cuando la información está lista se presiona el botón de *Aceptar Información* el cual codifica la información, la transmite sin codificarla, recibir la información codificada por el micro y muestra en pantalla la información codificada por el microcontrolador y por el programa. La función que realiza todos estos procedimientos es *Btncdfcn*.

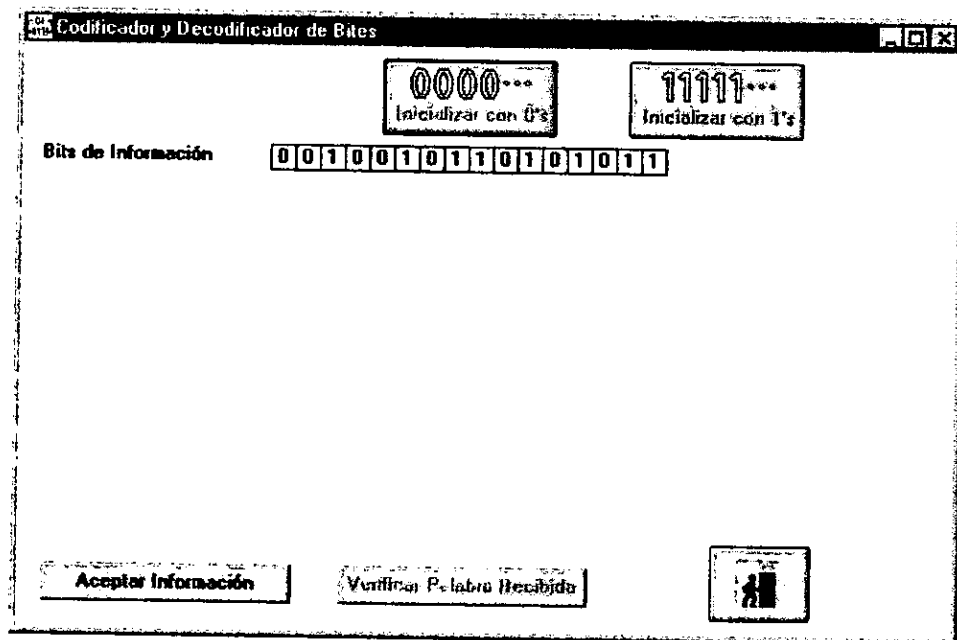
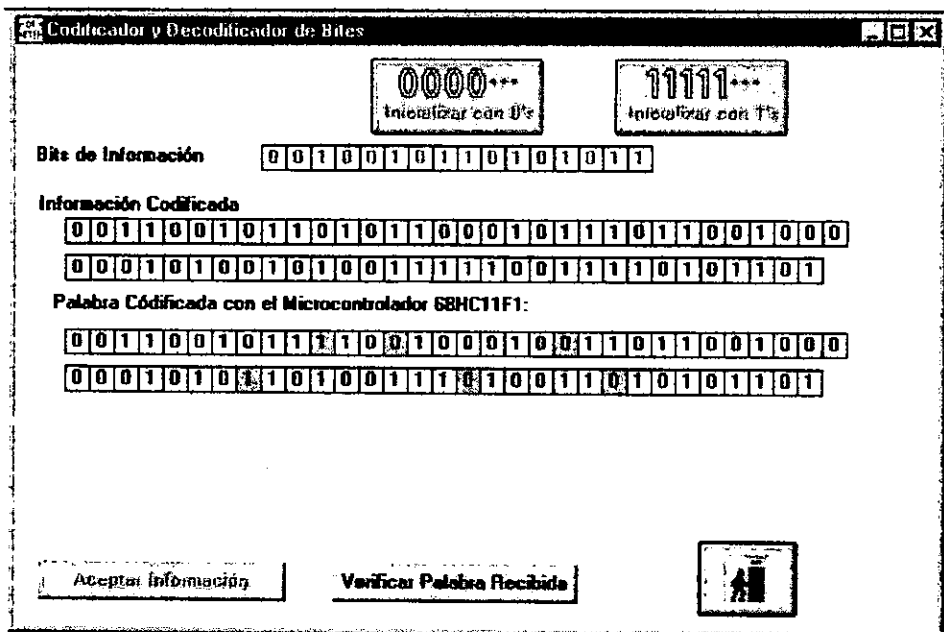


Figura B.7: Ventana del Simulador GF(2⁶)

Al terminar de ejecutarse dicha rutina el programa muestra los arreglos de cuadros que contienen la información codificada por el programa y por el microcontrolador como se observa en la Figura B.8.

Figura B.8: Ventana del Simulador GF(2⁶)

Es posible alterar la información codificada con el micro, esto es con la finalidad de mostrar que la rutina de detección y corrección de errores está funcionando correctamente. Esto se debe a que la transmisión y recepción de información con el microcontrolador es alámbricamente y por ello la probabilidad de que ocurran errores en dicho proceso es mínima, es por ello que se le da la oportunidad al usuario de introducir errores, para que observe que el Detector Corrector de Errores está funcionando correctamente. Para el ejemplo, mostrado los cuadros que se ven sombreados es en donde se introdujeron errores. Los errores se introducen por medio del ratón de la misma forma que se hace con el simulador anterior.

Al presionar el botón de *Verificar Palabra Recibida* el programa analiza la información mostrada en el arreglo de cuadros que está etiquetado con "Palabra Codificada con el Microcontrolador 68HC11F1".

El botón *Verificar Palabra Recibida* contiene la función *Bndcers* la cual contiene a la función *Decod* que es la que se encarga de analizar la palabra recibida para detectar si hay o no errores y en el caso de haberlos corregirlos. Después de que se ejecutan estas dos funciones en el caso de que hubieran ocurrido errores se muestra en pantalla un arreglo de cuadros que contienen la palabra corregida, así como se muestra en la Figura B.9, en el caso contrario se indica que no hay errores y más abajo se muestra la información decodificada.



Figura B.9: Ventana de Resultados

Los cuadros que se ven sombreados son los cuadros en donde se encuentran los bits que fueron corregidos. Por último el usuario puede salir del programa y o volver a empezar.

Es importante mencionar que antes de ejecutar el programa el microcontrolador debe estar conectado a la PC y además debe de estarse ejecutando el programa codificador hecho en ensamblador.

A continuación se muestra otro ejemplo en el cual no se introducen errores y tampoco se produjeron errores en la recepción. La información introducida es la que se muestra en la Figura B.10.

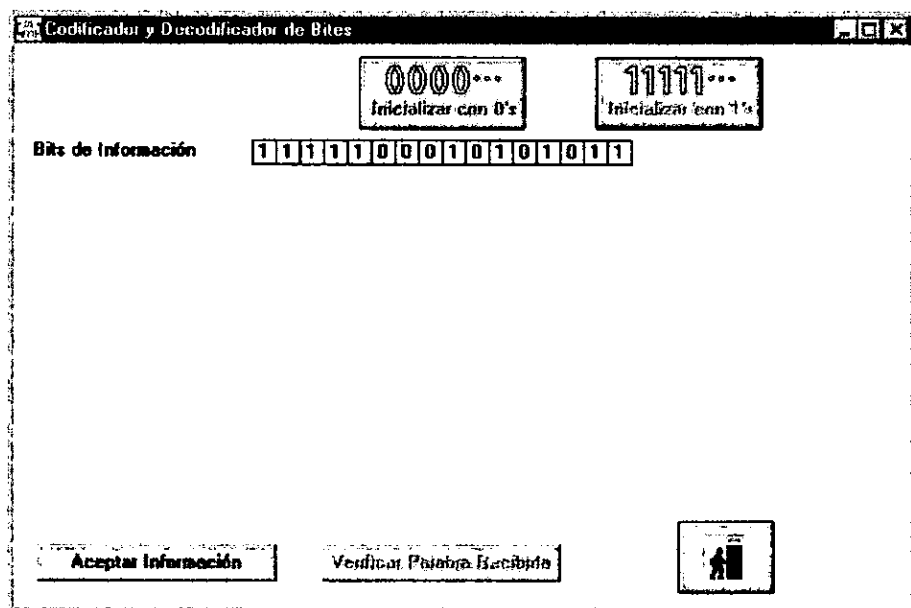


Figura B.10: Ventana con Otro Ejemplo

La información codificada por el microcontrolador y el programa es la que se muestra en la Figura B.11.



Figura B.11: Información Codificada por el MCU68HC11F1 y por el Simulador

Ahora no se introdujeron errores y además como se observa la información enviada por el microcontrolador es igual a la codificada por el programa, es decir que no ocurrieron errores durante la transmisión.

Se oprime el botón *Verificar Palabra Recibida* sin alterar la palabra recibida, el programa analiza la palabra recibida y al no detectar errores manda un

mensaje a pantalla que dice: "No Hay Error", después decodifica la información se manda a pantalla como se muestra en la Figura B.12.

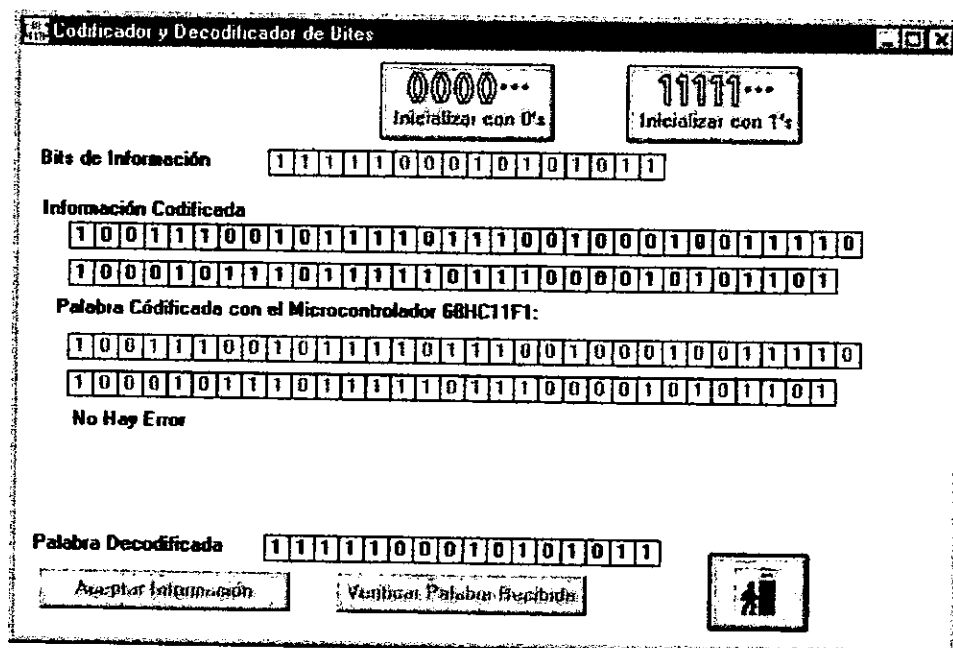


Figura B.12: Ventana que Muestra la Información Decodificada

Este programa se desarrolló con la finalidad de observar que el diseño del código y los algoritmos utilizados funcionaran correctamente. Una vez que se observó que funcionaban correctamente se procedió a implementar las funciones dentro del Módulo de Osciloscopio.

Apéndice C:

Códigos del Módulo de Osciloscopio

C.1 Código Fuente del Programa del MCU68HC11

A continuación se muestra el código fuente en ensamblador que utiliza el MCU68HC11F1 para capturar la señal, hacer la conversión A/D, codificarla y transmitirla inalámbricamente. Este código se conforma 8 de partes: *Inicialización del MCU68HC11F1, Rutina Principal, Subrutina de la Interrupción IRQ (Cuando se Presiona alguna Tecla de la Punta Inalámbrica), Subrutina de Interrupción para Generar una Señal Portadora de 40 KHz, Subrutina de Transmisión, Subrutina de Codificación, Subrutina de Retardo de 1/4 de Segundo y Subrutina de Conversión A/D.*

```
; ***** INICIALIZACIÓN DEL MC68HC11F1 *****
ORG $FE00 ; Primera Dirección de la EEPROM.
LDAA #$80 ; Se carga 80h, para configurar el Pto. G, donde
STAA $1003 ; sólo el bit 7 será de escr. y el resto de lect.
LDAA #$DF ; Se carga FBh, para configurar el Pto. A, donde,
STAA $1001 ; el bit 2 ser de lectura, los demás de escrit.
LDAA #$01 ; Carga 01h en el Reg. A, para ponerlo en el Pto. A
STAA $1000 ; (Indicador de ya se cargó el prog. en Memoria).
LDAA #$31 ; Se establece que el MCU68HC11F1 tendrá una veloc.
STAA $102B ; de transmisión de 4800 BPS, a través del Reg.
; de Control de Bauds del SCI (BAUD).
LDAA #$08 ; Se establece que el MCU68HC11F1 sólo va a
STAA $102D ; transmitir, configurándolo con el Registro 2 de
; Control SCI (SCCR2).
LDAA #$7E ; Carga 7Eh (correspondiente a la instrucción JMP)
```

Módulo de Osciloscopio

```

STAA SEE      ; en el 1er. byte del Vector IRQ (Vector de
               ; Interrupción).
LDX #INTEC    ; Carga el valor de la Dirección donde comienza la
STX SEF       ; rutina INTEC en sig. 2 bytes del Vector IRQ.
CLI           ; Limpia la Máscara de Interrupción.

LDAA #S80     ; Se establece que el MCU68HC11F1 va a utilizar
STAA $I039    ; Convertidor A/D configurándolo con el Registro
               ; de Opción de Configuración del Sist. (OPTION).
LDAA #$7E     ; Carga el valor 7Eh (OpCode de la instrucción JMP)
STAA $DC      ; en el 1er. byte del Pseudovector del Comparador
               ; 2 de Salida del Temporizador.
LDX #INPUL    ; Carga el valor de la Dirección donde comienza la
STX $DD       ; rutina INPUL en los otros dos bytes del
               ; Pseudovector del TOC2.
LDX #$0100    ; Carga el valor del Offset, el cual es de 40 KHz,
STX $00FE     ; poniéndolo en la Dir. $00FE de RAM y en la Dir.
STX $1018     ; del Reg. 2 del Comparador de Salida (TOC2).
LDAA #$40     ; Carga en Reg. A 40h para configurar el Comparador
STAA $I020    ; de Salida 2 a través del Registro 1 de Control
STAA $I022    ; del Temporizador (TCTL1), del Reg. 1 de la
STAA $I023    ; Máscara de Interrup. del Temporizador (TMSK1) y
               ; del Reg. 1 de la Bandera de Interrup. del
               ; Temporizador (TFLG1).
CLI           ; Limpia la Máscara de Interrupción.

; ***** RUTINA PRINCIPAL *****
PR1: NOP      ; Cuando MCU38HC11F1 está en este ciclo, verifica
LDAA $I002    ; el bit 7 del Pto. G (bit que indica
ANDA #$80     ; momento de conversión A/D de señal)
CMPA #$80     ; Mientras verifica hace que
BNE PR1       ; el MCU68HC11F1 no haga algo.
LDX #$0064    ; Carga el valor de 0014h, correspondientes a las
               ; 20 muestras.
PR2: LDAA #$05 ; Fija una Escala Normal del Periodo de la Señal
STAA $0101    ; Eléctrica.
JSR CONAD     ; Brinca a la rutina de Conversión A/D de señal.
JSR TRANS     ; Brinca a la rutina de Transmisión los valores
               ; convertidos.
DEX           ; Decrementa el Reg. X, (muestras a transmitir).
BNE PR2       ; Si no es cero el decremento (Si aún hay muestras
               ; a transmitir), brinca a la etiqueta PR2.
LDAA #$00     ; Carga 00h en Reg. A para ponerlo en el bit 7 del
STAA $I002    ; Pto. G (bit del momento de conv. A/D de señal).
BRA PR1       ; Brinca a PR1.

; ***** SUBROUTINA DE INTERRUPTIÓN DEL IRQ (CUANDO SE *****
; ***** PRESIONA ALGUNA TECLA DEL TECLADO REMOTO) *****

```

```

INTEC: PSHA ; Se meten los valores del Registro A en la Pila.
      JSR RETAR ; Retardo de 1/4 de Seg. (Brinca a RETAR).
      LDAA $1002 ; Carga en Reg. A, el valor del Pto. G, haciendo
      ANDA #$07 ; que sólo tome los bits 0, 1 y 2, por medio de
      STAA $0100 ; la AND, para posteriormente depositarlo en la
                  ; Dir. $0100 de RAM (Reg. de Datos).

      JSR RETAR ; Hace un retardo
      JSR RETAR ; de 1/2 Seg.
      LDAA $1002 ; Carga (otra vez después de 1/2 seg.) en Reg. A,
      ANDA #$07 ; el valor del Pto. G haciendo que sólo tome los
      CMPA $0100 ; bits 0,1 y 2 (con una AND) y compararlo con el
                  ; valor de la Dir. $0100 de RAM (Reg. de Datos).

      BNE IT2 ; Si son diferentes ambos valores brinca a IT2.
      CMPA #$00 ; Se compara Reg. A con 00h (por si se presionaron
                  ; a la vez, las 3 teclas del teclado remoto).

      BNE IT1 ; Si son diferentes los valores (NO se presionaron
                  ; las 3 teclas del teclado remoto), brinca a IT1.
      LDAA #$80 ; Carga 80h en el Reg. A, para dejarlo en el Pto. G
      STAA $1002 ; e indicar que se inicie la conversión A/D.
IT1:   LDAA #$80 ; Carga 80h en el Reg. A, dejándolo en Dir. $0101
      STAA $0101 ; de RAM, (Byte de Información) para indicar que
                  ; se presionó alguna tecla.

      JSR TRANS ; Transmite el Dato que est el Puerto Serie del
                  ; MCU68HC11F1 (Brinca a TRANS).
IT2:   PULA ; Saca el valor del Registro A de la Pila.
      RTI ; Retorno de Interrupción.
  
```

; ***** SUBROUTINA DE INTERRUPCIÓN PARA GENERAR *****
 ; ***** UNA SEÑAL PORTADORA DE 40KHz *****

```

INPUL: LDD $1018 ; Carga valor del Reg. 2 del Comparador de Salida
      ADDD $00FE ; (TOC2) y suma el Offset de 40 KHz (almacenado
      STD $1018 ; en Dir. $00FE de RAM) para que luego se deje en
                  ; en el Reg. 2 del Comparador de Salida (TOC2).
      LDAA #$40 ; Carga 40h en Reg. A para indicar que se hizo
      STAA $1023 ; la interrupción, por medio de Reg. 1 de Bandera
                  ; de Interrupción del Temporizador (TFLG1).
      RTI ; Retorno de Interrupción.
  
```

; ***** SUBROUTINA DE TRANSMISIÓN *****

```

TRANS: PSHA ; Se meten los valores de los Registros A y X
      PSXH ; en la Pila.
      LDAA #$03 ; Pone 03h en Pto. A, indicación del inicio de
      STAA $1000 ; trans. de palabra codif., por el Pto. Serie.
      JSR RETAR
      LDAA $0101 ; Carga en Reg. A lo que tiene la Dir. $0101 (Byte
      ORAA #$01 ; de Información) de RAM para hacerle una OR con
      STAA $0101 ; 01h, (asegura así que se transmite un dato) y
                  ; después se pone otra vez en Dir. $0101 de RAM.
  
```

Módulo de Osciloscopio

```

JSR CODIF ; Brinca a la Rutina CODIF de Codificación.
LDX #0102 ; Carga 0102h (correspondiente a Dir. $0102 de RAM),
; en el Registro X.
TR1: INX ; Incrementa el valor del Reg. X (apuntador de Dir.
; de RAM).
TR2: LDAA $102E ; Carga en Reg. A el valor del Reg. de Estado del
ANDA #S40 ; SCI (SCSR), para hacerle una AND con 40h. El 6
BEQ TR2 ; indica, si ya se puede transmitir, si no se
; puede todavía transmitir, brinca a TR2.
LDAA $00,X ; Carga en Reg. A, lo que tenga la primer. Dir.
STAA $102F ; Indexada de RAM, apuntada por el Reg. X, y se
; pone en Reg. de Datos SCI (byte a transmitir).
CPX #010A ; Compara 0106h (Dirección $0106 de RAM) con el
BNE TR1 ; contenido del Reg. X (apuntador de RAM), si son
; diferentes ambos valores, brinca a TR1.
LDAA #$00 ; Carga en Reg. A 00he para ponerlo en Dir. $0101
STAA $0101 ; (Byte INFO) asegurando así que transmitió la
; palabra.
LDAA #$01 ; Pone 01h en Pto. A, indicación del fin de transm.
STAA $1000 ; de la palabra codificada, por el Pto. Serie.
PULX ; Saca los valores de los Registros A y X
PULA ; de la Pila.
RTS ; Retorno de Rutina.

```

; ***** SUBROUTINA DE CODIFICACIÓN *****

```

CODIF: LDY #0103 ; Se carga en Reg. Y la 1er. Dir. donde estar la
; palabra a codificar.
CD1: LDAA #$00 ; El contenido de las 8 Direcciones
STAA $00,Y ; desde la Dirección $0103 hasta
INY ; la Dirección $010A se limpia,
CPY #010B ; es decir se inicializarán
BNE CD1 ; con cero.
LDX #FFFA ; Se carga en Reg. X la 1er. Dir. (Memoria EEPROM)
; de 6 del Vector Generador de Código,
CD4: LDAA $00,X ; Se carga uno a uno el contenido de las
STAA $00,Y ; Direcciones apuntadas por X .
INX ; a las Direcciones apuntadas por Y,
INY ; es decir, se copia el contenido desde las Dir.
CPY #0111 ; $FFFA hasta. $FFFF a las Dir. de RAM desde
BNE CD4 ; $010B hasta la Dir. $0110.
LDAA #$00 ; Se carga 00h en el Registro A. dejándolo en las
STAA $0111 ; Direcciones $0111 y $0112, ya que es necesario
STAA $0112 ; que inicien en cero la codificación.
LDX #0100 ; Se Carga en X la Dir. $0100, que es a partir de
; de donde se encuentran los 2 bytes a codificar.
CD5: LDAA #$80 ; Carga 80h en el Reg. A para dejarlo en la Dir.
STAA $0113 ; $0113 la cual servir como byte testigo de la
; codificación.
COD: LDAA $00,X ; Se carga en Reg. A el contenido de Dir. apuntada
ANDA $0113 ; por Reg. X, y hace una AND con el contenido de

```

```

BEQ CORR ; Dir. $0113 (Byte Testigo). Si el bit 6 es 1,
; brinca a CORR (hace corrimiento a la Derecha).
LDY #$0103 ; Se carga en Y la Dir. $0103 del Vector Generador.
CD2: LDAA $00,Y ; Carga en Reg. A el valor del byte apuntado por el
EORA $08,Y ; Reg. Y (Vector Generador) haciéndole un EXOR
STAA $00,Y ; con otro byte apuntado por Y (Vector de Código)
; poniendo el resul. en el byte del Vector Gener.
INY ; Incrementa Reg. Y hasta que sea igual a la Dir.
CPY #$010B ; $010B, mientras hace el resto EXOR con los
BNE CD2 ; bytes del Vector Generador de Código y Código.
CORR: LDY #$0112 ; Carga en Y la 1er. Dir. del Vector Generador,
LSR 00,Y ; para hacer un corrimiento hacia la derecha de
; dicho Vector.
CD3: DEY ; Decrementa Y, y hace corrimiento hacia la derecha
LSR 00,Y ; del contenido del byte apuntado por Reg Y.
BCC NOACA ; Si no hubo acarreo en el corrimiento va a NOACA.
LDAA #$80 ; Carga 80h en A para hacer un EXOR con siguiente
EORA $01,Y ; byte dejando el resultado en la Dir. apuntada
STAA $01,Y ; por Reg Y.
NOACA: CPY #$010B ; El corrimiento de todos los bytes del Vector
BNE CD3 ; Generador termina cuando el Reg. Y (como
; apuntador) es igual a la Dir. $010B.
; /****
LSR $0113 ; Hace un corrimiento del byte testigo para
; preguntar por el
BCC COD ; siguiente byte. En caso de que sea cero el
; resultado
INX ; Incrementa el la Dir. apuntada por X (Dir. de la
CPX #$0102 ; del corrimiento pregunta si ha terminado de
; codificar.
BNE CD5 ; Si la codificación no ha terminado brinca a CD5.
RTS ; Termina la codificación y la palabra codificada
; est en las Dir. $0103-$010A, siendo el byte
; menos significativo $0103.

```

```

; ***** SUBROUTINA DE RETARDO DE 1/4 DE SEGUNDO *****
RETAR: PSHX ; Se mete el Valor del Registro X en la Pila
LDX $FFFF ; Carga el Valor de FFFFh para que se decremente
RT1: NOP ; en uno, dentro de un ciclo, hasta que sea cero.
DEX ; el tiempo aproximado que tarda ese ciclo es de
BNE RT1 ; 1/4 de segundo (250 ms.).
PULX ; Saca de la Pila el valor del Registro X.
RTS ; Retorno de Rutina.

```

```

; ***** SUBROUTINA DE CONVERSIÓN A/D *****
CONAD: PSHA ; Mete el Valor del Registro A en la Pila.
LDAA $8A0 ; Carga el Reg. A con A0h (1010 0000) para ponerlo
STAA $1030 ; en el Reg. de Control del A/D (ADCTL).

```

Módulo de Osciloscopio

```
CN2:  LDAA $1030 ; Carga el contenido del Reg. de Control del A/D
      ANDA #$80  ; (ADCTL) en el Reg. A para hacerle una AND con
      BEQ CN2    ; 80h y ver si ya se hizo la conversión A/D. Si
                ; no la ha hecho va a CN2.
      LDAA $1031 ; Carga el contenido del Reg. 1 de Resultados del
                ; A/D (ADR1) en el Reg. A.
      STAA $0100 ; Pone la muestra de la conversión A/D de la señal
                ; eléc. en el Reg. $0100 de RAM (Byte de Dato).
      LDAA $1002 ; Carga en Reg. A el contenido del Pto. G que
      ANDA #$78  ; contiene la Escala de Voltaje con signo de la
                ; señal eléctrica y le hace una AND con 78h.
      ORAA $0101 ; Hace una OR del Registro A con la Dir. $0101
      STAA $0101 ; poniendo así la Escala de Voltaje de la
                ; Conversión A/D en la Dir. $0101 (Byte INFO).
      PULA      ; Saca de la Pila el último valor para dejarlo
                ; en el Registro A.
      RTS       ; Retorno de Rutina
```

C.2 Código del Programa "Módulo de Osciloscopio"

En esta sección se muestran las Formas utilizadas para el programa Módulo de Osciloscopio, al igual que algunas de las propiedades de los objetos contenidos en ellas. El Código Fuente completo hecho en Visual Basic 4.0 y con el que se manipulan todos estos objetos, viene en un disco (adicional del trabajo) con el nombre de CODIGOVB.TXT.

La forma principal del programa Módulo de Osciloscopio se muestra en la Figura C.1, junto con una numeración correspondiente a la indicación de todos sus objetos utilizados en ella. Los nombres y algunas de las propiedades de los objetos de la forma principal del programa Módulo de Osciloscopio se muestran en la Tabla C.1 (En esta tabla no se mencionan todos los nombres de las líneas que conforman la grátícula, ya que son muy comunes a los que sí se mencionan).

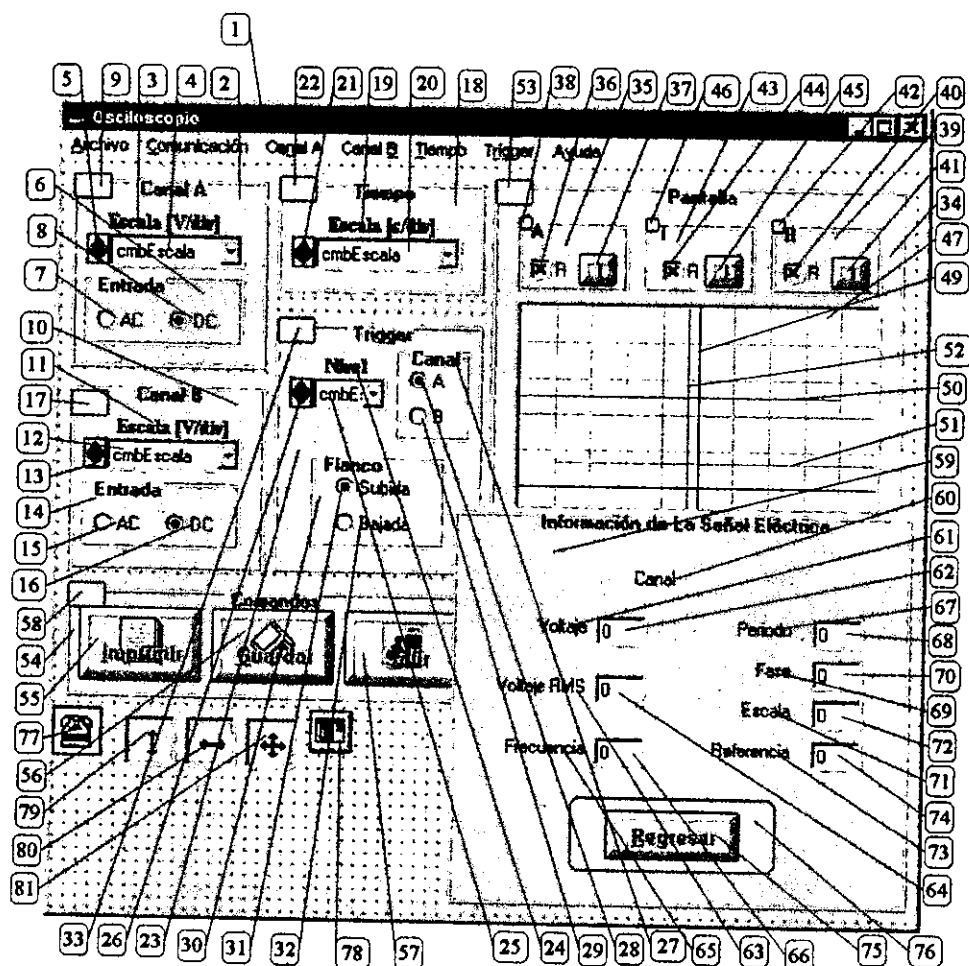


Figura C.1: Forma Principal del Programa "Módulo de Osciloscopio"

Módulo de Osciloscopio

N°	Objeto	Nombre	Propiedad
1	Form	frmOsciloscopio	Caption = "Osciloscopio"
2	SSFrame	frmVentana(1)	Caption = "Canal A"
3	Label	lblEscala(1)	Caption = "Escala [V/div]"
4	ComboBox	cmbEscala(1)	List = ""
5	SpinButton	spnEscala(1)	
6	SSFrame	frmControl(1)	Caption = "Entrada"
7	SSOption	opRefer(1)	Caption = "AC"
8	SSOption	opRefer(2)	Caption = "DC"
9	Shape	shpCuadro(1)	
10	SSFrame	frmVentana(2)	Caption = "Canal B"
11	Label	lblEscala(2)	Caption = "Escala [V/div]"
12	ComboBox	cmbEscala(2)	List = ""
13	SpinButton	spnEscala(2)	
14	SSFrame	frmControl(2)	Caption = "Entrada"
15	SSOption	opRefer(3)	Caption = "AC"
16	SSOption	opRefer(4)	Caption = "DC"
17	Shape	shpCuadro(2)	
18	SSFrame	frmVentana(3)	Caption = "Tiempo"
19	Label	lblEscala(3)	Caption = "Escala [s/div]"
20	ComboBox	cmbEscala(3)	List = ""
21	SpinButton	spnEscala(3)	
22	Shape	shpCuadro(3)	
23	SSFrame	frmVentana(4)	Caption = "Trigger"
24	Label	lblEscala(4)	Caption = "Nivel"
25	ComboBox	cmbEscala(4)	List = ""
26	SpinButton	spnEscala(4)	
27	SSFrame	frmControl(3)	Caption = "Canal"
28	SSOption	opRefer(5)	Caption = "A"
29	SSOption	opRefer(6)	Caption = "B"
30	SSFrame	frmControl(4)	Caption = "Flanco"
31	SSOption	opRefer(7)	Caption = "Subida"
32	SSOption	opRefer(8)	Caption = "Bajada"
33	Shape	shpCuadro(4)	
34	SSFrame	frmVentana(5)	Caption = "Pantalla"
35	SSFrame	frmControl(5)	Caption = "A"
36	SSCheck	chkR(1)	Caption = "R"
37	SSCommand	btnInfo(1)	Caption = "I"
38	Shape	shpCuadro(11)	
39	SSFrame	frmControl(6)	Caption = "B"
40	SSCheck	chkR(2)	Caption = "R"
41	SSCommand	btnInfo(2)	Caption = "I"
42	Shape	shpCuadro(12)	

Tabla C.1: Propiedades de los Objetos de la Forma Principal del Programa "Módulo de Osciloscopio"

Continuación de las propiedades de los objetos de la forma principal.

Nº	Objeto	Nombre	Propiedad
43.	SSFrame	frmControl(7)	Caption = "T"
44.	SSCheck	chkR(3)	Caption = "R"
45.	SSCommand	btnInfo(3)	Caption = "T"
46.	Shape	shpCuadro(13)	
47.	PictureBox	Pantalla	
48.	Line	LinRej(4)	
49.	Line	LinRej(13)	
50.	Line	LinPosición(1)	
51.	Line	LinPosición(2)	
52.	Line	LinPosición(3)	
53.	Shape	shpCuadro(5)	
54.	SSFrame	frmVentana(6)	Caption = "Comandos"
55.	SSCommand	btnComando(1)	Caption = "&Imprimir"
56.	SSCommand	btnComando(2)	Caption = "&Guardar"
57.	SSCommand	btnComando(3)	Caption = "&Salir"
58.	Shape	shpCuadro(6)	
59.	SSFrame	frmVentana(0)	Caption = "Información de La Señal Eléctrica"
60.	Label	lblCanal	Caption = ""
61.	Label	lblInfo(0)	Caption = "Voltaje"
62.	TextBox	txtInfo(0)	Text = ""
63.	Label	lblInfo(1)	Caption = "Voltaje RMS"
64.	TextBox	txtInfo(1)	Text = ""
65.	Label	lblInfo(2)	Caption = "Frecuencia"
66.	TextBox	txtInfo(2)	Text = ""
67.	Label	lblInfo(3)	Caption = "Periodo"
68.	TextBox	txtInfo(3)	Text = ""
69.	Label	lblInfo(4)	Caption = "Fase"
70.	TextBox	txtInfo(4)	Text = ""
71.	Label	lblInfo(5)	Caption = "Escala"
72.	TextBox	txtInfo(5)	Text = ""
73.	Label	lblInfo(6)	Caption = "Referencia"
74.	TextBox	txtInfo(6)	Text = ""
75.	SSCommand	btnRegresar	Caption = "&Regresar"
76.	Shape	shpCuadro(0)	
77.	MSComm	Comm1	
78.	CommonDialog	CrnnDialog	
79.	PictureBox	PicFlechas(0)	
80.	PictureBox	PicFlechas(1)	
81.	PictureBox	PicFlechas(2)	

Tabla C.1: Continuación

La forma principal maneja un sistema de menús, los cuales se mencionan en el código fuente del programa. En la Tabla C.2 se muestran todas las opciones de menús disponibles con su nombre y la propiedad Caption.

Nombre	Caption (Propiedad)
mnArchivo	"&Archivo"
mnArchivoNvo	"&Nuevo"
mnArchivoAbr	"&Abrir"
mnArchivoGdr	"&Guardar"
mnArchivoImp	"&Imprimir"
mnArchivoImT	"Imprimir & Todo"
mnArchivoImG	"Imprimir & Gráfica"
mnArchivoImV	"Imprimir & Valores"
mnArchivoSlr	"&Salir"
mnComunica	"&Comunicación"
mnComPS(1)	"Puerto Serie &1"
mnComPS(2)	"Puerto Serie &2"
mnComSon	"&Sonido"
mnCnlA	"Ca&nal A"
mnCnlEscA	"&Escala"
mnCnlEscAVal(0)	"50 V/div"
mnCnlEscAVal(1)	"20 V/div"
mnCnlEscAVal(20)	"10 μ V/div"
mnCnlPscEntA	"&Posición Divisional"
mnCnlPscEntAVal(0)	"0 div"
mnCnlPscEntAVal(9)	"9 div"
mnCnlPscDecA	"Po&sición Subdivisional"
mnCnlPscDecAVal(0)	"0 subdiv"
mnCnlPscDecAVal(9)	"9 subdiv"
mnCnlAA(1)	"&AC"
mnCnlAA(2)	"&DC"
mnCnlB	"Canal &B"
mnCnlEscB	"&Escala"
mnCnlEscBVal(0)	"50 V/div"
mnCnlEscBVal(1)	"20 V/div"
mnCnlEscBVal(20)	"10 μ V/div"
mnCnlPscEntB	"&Posición Divisional"
mnCnlPscEntBVal(0)	"0 div"
mnCnlPscEntBVal(9)	"9 div"
mnCnlPscDecB	"Po&sición Subdivisional"

Tabla C.2: Opciones del Muni del Programa "Módulo de Osciloscopio"

Continuación de las opciones de menú del programa "Módulo de Osciloscopio".

Nombre	Caption (Propiedad)
mnCnlPscDecBVal(0)	"0 subdiv"
mnCnlPscDecBVal(9)	"9 subdiv"
mnCnlBB(1)	"&AC"
mnCnlBB(2)	"&DC"
mnTiempo	"&Tiempo"
mnTiempoEsc	"&Escala"
mnTiempoEscVal(0)	"500 ms/div"
mnTiempoEscVal(1)	"200 ms/div"
mnTiempoEscVal(26)	"1 ns/div"
mnTiempoPscEnt	"&Posición Divisional"
mnTiempoPscEntVal(0)	"0 div"
mnTiempoPscEntVal(9)	"9 div"
mnTiempoPscDec	"Po&sición Subdivisional"
mnTiempoPscDecVal(0)	"0 subdiv"
mnTiempoPscDecVal(9)	"9 subdiv"
mnTrigger	"T&trigger"
mnTriggerCnl(1)	"Canal &A"
mnTriggerCnl(2)	"Canal &B"
mnTriggerDsp(1)	"&Subida"
mnTriggerDsp(2)	"Ba&jada"
mnTriggerNvl	"&Nivel"
mnTriggerNvlSig	"Signo"
mnTriggerNvlSigVal(0)	"Positivo"
mnTriggerNvlSigVal(1)	"Negativo"
mnTriggerNvlEnt	"Entero"
mnTriggerNvlEntVal(0)	"0"
mnTriggerNvlEntVal(5)	"5"
mnTriggerNvlDec	"Decimal"
mnTriggerNvlDecVal(0)	".0"
mnTriggerNvlDecVal(9)	".9"
mnAyuda	"A&yuda"
mnAyudaPgr	"&Uso de Programas"
mnAyudaTecMen	"&Teclas del Menú"
mnAyudaTecRem	"Teclado &Remoto"
mnAyudaAdP	"&Acerca del Programa"

Tabla C.2: Continuación

No se mencionan todas las opciones de menú de las Escalas de Voltajes del Canal A y B, de las Escalas de Tiempo, del Nivel de Trigger y de las posiciones divisionales y subdivisionales de las líneas de referencia de Canal A, Canal B y Tiempo, ya que todas son muy comunes a las que sí se mencionan.

En la Figura C.2 se muestra la forma con la que se manda impresión las señales eléctricas capturadas. Esta forma tiene el nombre Gráticula y utiliza los mismos nombres de los objetos de la gráticula contenida en la forma principal frmOsciloscopio.

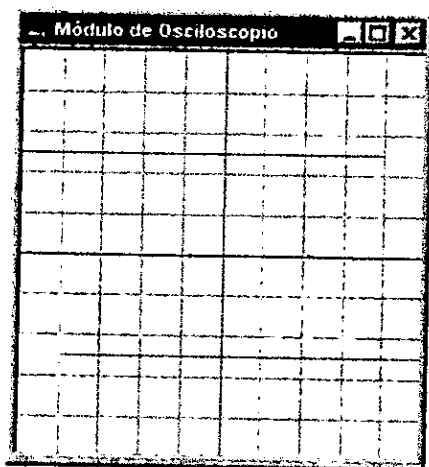


Figura C.2: Forma "Gráticula" para Imprimir las Señales Eléctricas

Apéndice D:

Hojas de Especificación

D.1 Hojas de Especificación de los ILEDs

Las siguientes Tablas muestran los valores eléctricos que manejan los ILEDs utilizados generalmente para los aparatos de Control Remoto y cuyos nombres comerciales son TN130 (3 mm.) y TN153 (5 mm.).

D.1.1 Máximos Rangos Absolutos

Considerando la Temperatura ambiente $T_a = 25^{\circ}\text{C}$, los máximos rangos absolutos para los ILEDs se muestran en la Tabla D.1.

Parámetro	Símbolo	Rango Máximo	Unidad
Potencia de Disipación	Pd	100	mW
Corriente en Sentido Directo	IF	50	mA
Corriente Máxima en Sentido Directo	IP	1.2	A
Voltaje Inverso	VR	5	V
Temperatura de Operación	Topr	-40 ~ +85	°C

Tabla D.1: Máximos Rangos de los ILEDs TN130 y TN153.

D.1.2 Características Electroópticas

Considerando la Temperatura ambiente $T_a = 25\text{ }^{\circ}\text{C}$, las características electroópticas se muestran en la Tabla D.2.

Parámetro	Símbolo	Típica	Máxima	Unidad	Condiciones
Voltaje en Sentido Directo	VF	1.2	1.5	V	IF = 20 mA
Corriente Inversa	IR	—	10	μA	VR = 5V
Capacitancia	Co	90	—	PF	V = 0, f = 1MHz
Longitud de Onda	P	940	—	nm	IF = 20 mA
Ancho de Banda Espectral		50	—	nm	IF = 20 mA
Ángulo	$2\theta\frac{1}{2}$	30	—	Deg	

Tabla D.2: Características Electroópticas de los ILEDs TN130 y TN153.

D.1.3 Otras Características

Algunas otras características que deben considerarse al usar los ILEDs se muestran en la Tabla D.3.

Nombre Comercial	Intensidad de Radiación (PO) mW/sr 20 mA		Lente	Tamaño
	mínima	Típica		
TN130F	2.6	3.6	Transparente	3 mm.
TN130BF	2.6	3.6	Azul-Claro	3 mm.
TN153F	3.7	7.2	Transparente	5 mm.
TN153BF	3.7	7.2	Azul-Claro	5 mm.

Tabla D.3: Características Generales de los ILEDs TN130 y TN153.

D.2 Especificaciones del Receptor/Demodulador GP1U52X

El Receptor/Demodulador de luz infrarroja utilizado es el GP1U52X el cual es un circuito integrado consistente de un circuito detector infrarrojo diseñado para usarse para los controles remotos en televisiones, videograbadoras y radios. Es de bajo costo.

El GP1U52X utiliza un pin fotodiodo que tiene su sensibilidad pico cercano al rango de infrarrojos. Los filtros bloquean la luz para reducir o eliminar falsas operaciones provocadas por otras fuentes de luz.

La salida del fotodiodo alimenta al preamplificador/alimentador para proporcionar una señal limpia al resto del circuito. El filtro PasoBandas rechaza todas las señales fuera de la banda $40\text{KHz} \pm 4\text{KHz}$. La señal restante se va al demodulador, integrador y al comparador. Al final la salida es una señal limpia de portadora. En la Figura D.1 se muestra el Diagrama de Bloques del GP1U52X.

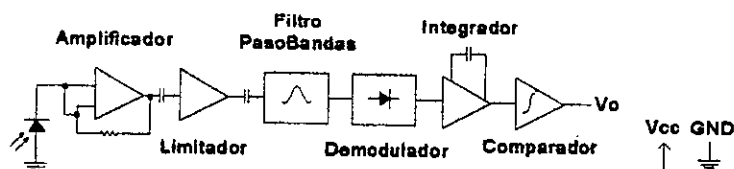


Figura D.1: Diagrama de Bloques del Receptor/Demodulador de Luz Infrarroja GP1U52X

D.2.1 Características del GP1U52X

El GP1U52X tiene las siguientes características:

- ♦ Rechaza todas las fuentes no moduladas a 40 KHz para una operación segura.
- ♦ Tamaño compacto para aplicaciones miniatura.
- ♦ Opera desde 5V, haciendo fáciles conexiones directas a componentes TTL y CMOS.
- ♦ Protegido con carcasa para proporcionar inmunidad total de ruido externo inducido por campos magnéticos..
- ♦ Construido con filtro pasobajas en la conexión de polarización ayuda a aislar el circuito de ruidos en la polarización.

D.2.2 Especificaciones Eléctricas del GP1U52X

Las especificaciones eléctricas del GP1U52X se muestran en la Tabla

D.4.

Parámetros	Valores	Unidad
Voltaje Máximo de Polarización	6.3	V
Voltaje de Operación Recomendado	5.0 $\pm$ 0.3	V
Disipación de Corriente (ICC)	5.0	mA
Frecuencia Central del PasoBandas	40	KHz
Ancho de Banda -3dB de 40KHz	4	KHz
PasoBandas Infrarrojo	980 $\pm$ 100	nm

Tabla D.4: Especificaciones del GP1U52X.

D.3 Especificaciones Eléctricas del MAX232CPE

El MAX232CPE es un convertidor de voltaje para poder manejar el Estándar del RS232 que utiliza la computadora a través de su Puerto Serie. Este Puerto Serie maneja Voltajes de -10 V y 10 V. los cuales representan un 1 y un 0 lógicos respectivamente. Entonces el MAX232CPE convierte 5 V y 0 V a -10 V y 10 V respectivamente. En la Figura D.2 se muestra la configuración Eléctrica del MAX232CPE para obtener los voltajes que posteriormente se mencionan.

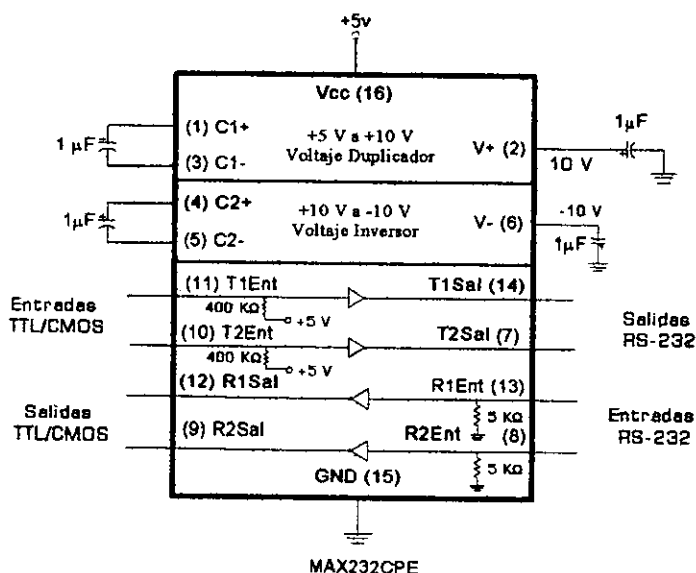


Figura D.2: Configuración Eléctrica del MAX232CPE para Obtener Voltajes RS232 y TTL/CMOS

A este circuito integrado se le deben conectar 4 capacitores de 1 µF: el primero entre los pines C1+ y C1-, el segundo entre los pines C2+ y C2-, el tercero entre el pin V- y Tierra y el cuarto entre V+ y Tierra. Una vez conectado los

capacitores como se especifica en el Diagrama Eléctrico se obtienen los siguientes niveles de voltaje: T1Ent y T2Ent aceptan niveles TTL/CMOS, T1Sal y T2Sal proporcionan niveles de ± 10 VDC o RS-232, R1Ent y R2Ent aceptan niveles RS-232 y R1Sal y R2Sal proporcionan niveles TTL/CMOS. Si se quiere transmitir información a través del MAX232CPE, se utilizan los pines que se refieren a T, pero si se quiere recibir información se utilizan los pines referidos con R. El Circuito Integrado hace su propio duplicador de voltaje y su inversor de voltaje, los son distribuidos internamente.

D.4 Especificaciones Eléctricas de los Transistores 2N2222A

Las características eléctricas del transistor 2N2222A se muestran en la Tabla D.5.

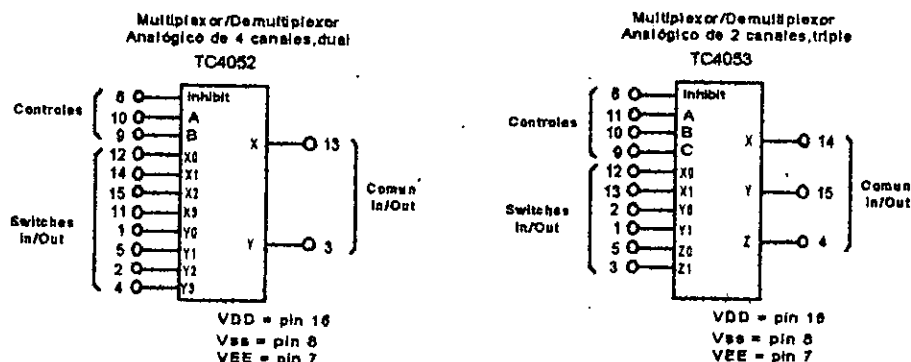
Parámetro	Símbolo	Valor	Unidad
Voltaje Colector a Base	BV_{CBO}	75	V
Voltaje Colector a Emisor	BV_{CEO}	40	V
Voltaje Base a Emisor	BV_{EBO}	6	V
Corriente Máxima del Colector	I_C	0.8	A
Potencia Máxima de Disipación	P_D	0.5 ($T_A = 25^\circ\text{C}$)	W
Frecuencia	f_t	300	MHz
Máxima Ganancia de Corriente	h_{FE}	300 typ	
Mínima Ganancia de Corriente	h_{FE}	100 typ	

Tabla D.5: Características Eléctricas del Transistor 2N2222A

D.5 Especificaciones de los Multiplexores TC4052 y TC4053

Los multiplexores analógicos TC4052 y TC4053 son switches analógicos controlados digitalmente. Los dos dispositivos se caracterizan por tener baja

impedancia (ON), y muy baja corriente de fuga (OFF). Se puede lograr el manejo de señales analógicas hasta el rango completo de la fuente de voltaje. En la Figura D.3 se muestran los esquemas de los multiplexores analógicos.



Nota: Las entradas de control son referenciadas a Vss; las entradas y salidas analógicas referenciadas a VEE; donde $VEE \leq Vss$.

Figura D.3: Esquemas de los Multiplexores Analógicos TC4052 y TC4053

En la Tabla D.6 se muestra la Tabla de Verdad que se obtienen de los Multiplexores Analógicos TC4052 y TC4053.

Entradas de Control				Switches en Estado ON					
Inhibit	Selección			TC4052		TC4053			
	C	B	A						
0	0	0	0	Y0	X0	Z0	Y0	X0	
0	0	0	1	Y1	X1	Z0	Y0	X1	
0	0	1	0	Y2	X2	Z0	Y1	X0	
0	0	1	1	Y3	X3	Z0	Y1	X1	
0	1	0	0	-	-	Z1	Y0	X0	
0	1	0	1	-	-	Z1	Y0	X1	
0	1	1	0	-	-	Z1	Y1	X0	
0	1	1	1	-	-	Z1	Y1	X1	
1	*	*	*	None		None			

*Nota: * no importa el estado*

Tabla D.6: Tabla de Verdad de los Multiplexores Analógicos TC4052 y TC4053

D.5.1 Características del TC4052 y TC4053

Las características generales de los multiplexores analógicos son:

- ♦ Diodo de protección sobre todas las entradas.
- ♦ Rango de voltaje de la fuente = 3.0 Vdc a 18 Vdc.
- ♦ Rango de voltaje analógico ($V_{DD} - V_{EE}$) = 3 a 18 V.
- ♦ Características de transferencia linealizadas.
- ♦ Bajo ruido -12nV/ciclo, $f = 1$ KHz típico.

D.5.2 Máximos Rangos Absolutos

Los máximos rangos absolutos de los amplificadores operacionales TC4052 y TC4053 se muestran en la Tabla D.7.

Símbolo	Parámetros	Valores	Unidad
V_{DD}	Fuente de Voltaje DC (Referenciado a V_{EE})	-0.5 a +18.0	V
V_{in} V_{out}	Entrada o Salida de Voltaje (Referenciado a V_{SS} para el Control de Entradas y V_{EE} para Switch I/O).	-0.5 a $V_{DD}+0.5$	V
I_{in}	Corriente de Entrada para el Pin de Control.	+10	mA
I_{sw}	Corriente a través del Switch.	+25	mA
P_D	Disipación de Potencia por Empaquetado.	500	mW
t_{stg}	Temperatura de Almacenamiento	-65 a +150	°C

Tabla D.7: Máximos Rangos Absolutos de los Multiplexores Analógicos TC4052 y TC4053

D.5.3 Características Eléctricas

Las características eléctricas de los multiplexores analógicos TC4052 y TC4053 se muestran en la Tabla D.8.

Característica	Símbolo	Condiciones de Prueba	Valores	Unl.
Rango de la Fuente de Voltaje DC.	V_{DD}	$V_{DD} - 3 \geq V_{SS} \geq V_{EE}$	3 a 18	V
Corriente Total de la Fuente	I_D	$T_a = 25^\circ\text{C}$	$0.07 \mu\text{A/KHz} + I_{DD}$	μA
Tiempo de Propagación	t_{PLH}, t_{PHL}	$V_{dc} = 5\text{ V}$	30	ns
Distorsión de la Segunda Armónica	-	$V_{dc} = 10\text{ V}$	0.07	%
Ancho de Banda	BW	$V_{dc} = 10\text{ V}$	17	MHz
Atenuación a través del Canal Off	-	$V_{dc} = 10\text{ V}$	-50	dB
Control de Entradas -Inhibit, A, B, C				
Voltaje de entrada de Nivel Bajo	V_{IL}	$V_{DD} = 5\text{ V}$	2.25	V
Voltaje de Entrada de Nivel Alto	V_{IH}	$V_{DD} = 5\text{ V}$	2.75	V
Capacitancia de Entrada	C_{in}	-	5.0	pF
In/Out en los Switches y Out/In Comunes -X, Y, Z				
Voltaje Pico-Pico Recomendado	V_{io}	Canal On u Off	V_{DD}	V_{RP}
Voltaje de Offset de Salida	V_{OO}	$V_{in} = 0\text{ V}$, sin carga	10	μV
Resistencia de ON	R_{on}	$V_{switch} \leq 500\text{ mV}$	250	Ω
Δ Resistencia (ON) entre Cualquiera de los dos Canales.	ΔR_{on}	-	25	Ω
Capacitancia del Switch I/O	C_{io}	Inhibit = V_{DD}	10	pF
Capacitancia Común O/I	C_{vi}	Inhibit = V_{DD}	60	pF

Tabla D.8: Características Eléctricas de los Multiplexores Analógicos TC4052 y TC4053

D.6 Especificaciones de los Operacionales CA3130 y CA3140

Los C.I. CA3130 y CA3140 son amplificadores operacionales que combinan las ventajas de transistores de alto voltaje PMOS con transistores bipolares de alto voltaje en un chip monolítico. En forma general el amplificador operacional CA3140 es una versión mejorada del CA3130 en cuanto a desempeño.

El amplificador operacional CA3130 y CA3140 BiMos se caracterizan debido a que tienen una compuerta de protección con transistores MOS/FET (PMOS) en la entrada del circuito, lo que les da una muy elevada impedancia de entrada, muy baja corriente de entrada y alta velocidad de desempeño.

Estos amplificadores operacionales son internamente compensados en fase para ofrecer una operación estable en operación como seguidor de ganancia unitaria, y adicionalmente se tiene acceso a las terminales para un capacitor externo suplementario, si se desea una estabilidad en frecuencia. Las terminales son provistas para uso en aplicaciones que requieren un voltaje de offset nulo. En la Figura D.4 se muestra el diagrama correspondiente.

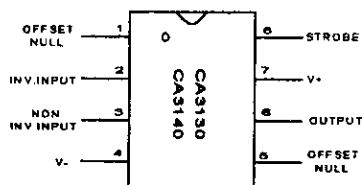


Figura D.4: Diagrama de los Amplificadores Operacionales CA3130 y CA3140

D.6.1 Características del CA3130 y CA3140

Los Amplificadores Operacionales CA3130 y CA2140 tienen las siguientes características.

- ◆ Estado de Entrada MOS/FET .
- ◆ Muy Alta Impedancia de Entrada (Z_{in}), 1.5 T Ω .
- ◆ Muy Baja Corriente de Entrada (I_i) , 10 pA a ± 15 V.
- ◆ Bajo Voltaje de Offset de Entrada (V_{io}) , 2 mV max.
- ◆ Internamente Compensado.
- ◆ Operación desde 4 V hasta 44 V con una Fuente Dual o Sencilla.
- ◆ Caracterizado para una Operación de ± 15 V y para Sistemas TTL con Operación de 4 V.

- ♦ Ancho de Banda de 4.5 MHz con Ganancia Unitaria con una Fuente de $\pm 15V$, y 3.7 MHz con Fuente de 4 V.
- ♦ Alta Velocidad de Respuesta de 9 V/ μs .
- ♦ Estado de Salida Compensado.

D.6.2 Características Eléctricas

Las principales características eléctricas de los Amplificadores Operacionales CA3130 y CA3140 se muestran en la Tabla D.9.

Característica	Condiciones de prueba	Valores	Unidad
Resistencia de Ajuste del Voltaje de Offset de Entrada.	Vdc = $\pm 15V$	4.7	K Ω
Resistencia de Entrada	Ta = 25 °C	1.5	T Ω
Resistencia de Salida	Ta = 25 °C	60	Ω
Capacitancia de Entrada	Ta = 25 °C	4	pF
Voltaje de Ruido en la Entrada	BW= 140 KHz	48	μV
Ganancia de Ancho de Banda	-	4.5	MHz
Velocidad de Respuesta	-	9	V/ μs
Ganancia en Señal Grande	Vdc = $\pm 15V$	100	dB
Corriente de Entrada	Vdc = $\pm 15V$	10	nA
Máxima Salida de Voltaje	Vdc = $\pm 15V$	-4.4	V
Dispositivo de Disipación, Pd	Vdc = $\pm 15V$	120	mW

Tabla D.9: Características Eléctricas de los Amplificadores Operacionales CA3130 y CA3140

D.6.3 Máximos Rangos Absolutos

Los máximos rangos absolutos se muestran en la Tabla D.10.

Parámetros	Valores	Unidad
Fuente de Voltaje DC (entre V- y V+)	36	V
Voltaje de Entrada en Modo Diferencial	+8 a -5	V
Corriente de Entrada	1	mA
Dispositivo de Disipación	630	mW
Rango de Temperatura	-55 a 125	°C

Tabla D.10: Máximos Valores Absolutos de los Amplificadores Operacionales CA3130 y CA3140

D.7 Especificaciones del Amplificador Operacional TL074

El C.I. TL074 es un conjunto de cuatro amplificadores operacionales de bajo ruido con entrada JFET. Estos amplificadores operacionales combinan dos tecnologías lineales en un solo circuito integrado monolítico. Cada amplificador operacional está internamente compensado para bajos voltajes de offset. En la Figura D.5 se muestra el diagrama del Amplificador Operacional TL074.

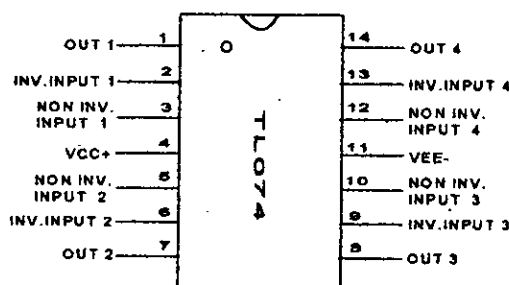


Figura D.5: Diagrama del Amplificador Operacional TL074

D.7.1 Características del TL074

Las características del amplificador operacional TL074 son:

- ♦ Bajo Voltaje de Ruido de Entrada (18 nV/MHz).
- ♦ Baja Distorsión Armónica (0.01%).
- ♦ Baja Corriente de Offset y Entrada de Bias.
- ♦ Alta Impedancia de Entrada ($10^{12} \Omega$).
- ♦ Alta Velocidad de Respuesta (13 V/ μ s).
- ♦ Amplia Ganancia de Ancho de Banda (4.0 MHz)
- ♦ Fuente de Corriente Baja (1.4 mA por Amp).

D.7.2 Rangos Máximos

En la Tabla D.11 se muestran los rangos máximos absolutos del amplificador operacional TL074.

Parámetros	Símbolo	Valores	Unidad
Fuente de Voltaje DC	V_{CC}	+18	V
	V_{EE}	-18	V
Voltaje de Entrada Diferencial	V_{ID}	+30	V
Rango de Voltaje de Entrada	V_{ER}	+15	V
Disipación de Potencia	P_D	680	mW
Rango de Temperatura de Operación	T_A	0 a 70	°C

Tabla D.11: Máximos Valores Absolutos de los Amplificadores Operacionales CA3130 y CA3140

D.7.3 Características Eléctricas

En la Tabla D.12 se muestran las principales características eléctricas del amplificador operacional TL074.

Característica	Símbolo	Valores	Unidad
Voltaje de Offset de Entrada	V_{IO}	3	mV
Coefficiente de Voltaje de Offset/Temperatura	Δ_{IO}/T	10	$\mu V/^{\circ}C$
Corriente de Bias ($V_{cm} = 0$)	I_B	30	pA
Corriente de Offset de Entrada	I_{IO}	5	pA
Resistencia de Entrada	R_i	10^{12}	Ω
Ganancia de Voltaje de Señal Grande	A_{VOL}	150	V/mV
Fuente de Corriente de Cada Amplificador	PSRR	1.4	mA
Ancho de Banda de Ganancia Unitaria	BW	4.0	MHz
Velocidad de Respuesta ($V_m = 10$ V, $R_L = 2$ K Ω)	SR	13	V/ μ s

Tabla D.12: Características Eléctricas del Amplificador Operacional TL074

D.8 Especificaciones de Contador HCF4516

El C.I. HCF4516 es un circuito contador CMOS funcionalmente equivalente al 74LS191 (TTL). Este es un contador binario de 4 bits de tipo Up-Down (Arriba-Abajo), además puede ser puesto en cascada utilizando una lógica interna de carry/borrow (acarreo/préstamo). En operación normal, Carry In (Cin), Reset (Rst) y Load se mantienen en estado bajo. El diagrama básico del contador HCF4516 se muestra en la Figura D.6.

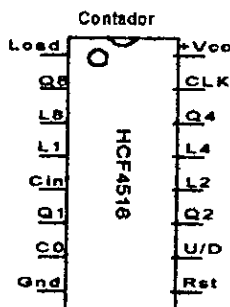


Figura D.6: Diagrama del Contador HCF4516

D.8.1 Características del HCF4516

Las principales características del contador HCF4516 son:

- ◆ Frecuencia Máxima del Reloj: $f = 3 \text{ MHz}$ con Fuente de 10 V; y $f = 1.5 \text{ MHz}$ con Fuente de 5 V.
- ◆ Corriente Total de la Fuente con una Velocidad del Reloj de 1 MHz: $I = 0.4 \text{ mA}$ con Fuente de 5 V; $I = 0.8 \text{ mA}$ con Fuente de 10 V.
- ◆ El Rizo del Reloj y Tiempo de Caída deberá ser más Rápido que $5 \mu\text{s}$.

D.8.2 Características Eléctricas

Las características eléctricas del contador HCF4516 se muestran en la Tabla D.13.

Parámetros	Símbolo	Valores	Unidad
Fuente de Voltaje DC	V_{DD}	3 a 10	V
Nivel de Voltaje Alto	V_H	V_{DD}	V
Nivel Bajo de Voltaje	V_L	0 a 0.5	V
Disipación de Potencia	P_D	0.1	mW
Retardo de Propagación	t_r	25	ns
Añanico de Salida	-	50	-
Margen de Ruido	-	3	V

Tabla D.13: Características Eléctricas del Contador HCF4516

D.9 Especificaciones de los CI 74HC08, 74HC32 y 74HC04

Las compuertas AND (74HC08), OR (74HC32) e INVERSOR (74HC04) son dispositivos de tecnología CMOS de alta velocidad y compatibles con la familia TTL. En la Figura D.7 se muestran los diagramas correspondientes. En las Tabla D.14 se muestra los estados lógicos para los C.I. 74HC08 y 74HC32.

A	B	74HC08	74HC32
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	1

Nota: Para el 74HC08 $C = A \cdot B$, para el 74HC32 $C = A + B$

Tabla D.14: Tabla Lógica para los C.I. 74HC08 y 74HC32

Para el CI 74HC04 su tabla de verdad se muestra en la Tabla D.15.

A	74HC04
0	1
1	0

Tabla D.15: Tabla Lógica para el C.I. 74HC04

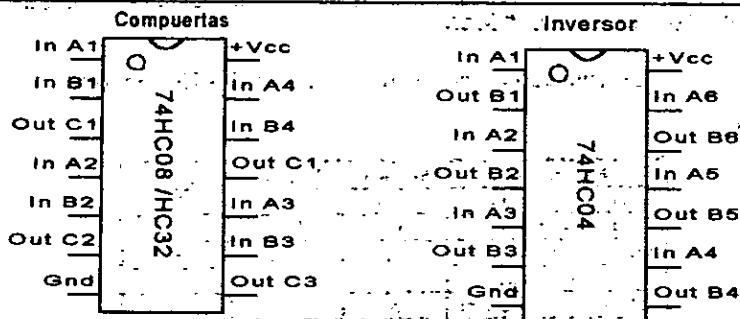


Figura D.7: Diagrama de los C.I. 74HC08(AND), 74HC32(OR) y 74HC04 (Inversor)

D.9.1 Características del 74HC08, 74HC32 y 74HC04

Algunas características de los CI 74HC08, 74HC32 y 74HC04 son:

- ♦ Los Cambios Lógicos de Alto a Bajo se Presentan con una Aceptable Inmunidad al Ruido.
- ♦ Los Circuitos Trabajan sobre un Amplio Rango en la Fuente de Voltaje, Típicamente de +13 V a +15 V.
- ♦ Las Entradas para todos los Dispositivos son Circuitos Abiertos y así son muy Fáciles de Manejar.

D.9.2 Características Eléctricas

Las características eléctricas de los C.I. 74HC08, 74HC32 y 74HC04 se muestran en la Tabla D.16.

Parámetros	Símbolo	Valores	Unidad
Fuente de Voltaje DC	V_{DD}	3 a 10	V
Nivel de Voltaje Alto	V_H	V_{DD}	V
Nivel Bajo de Voltaje	V_L	0 a 0.5	V
Disipación de Potencia	P_D	0.1	mW
Retardo de Propagación	t_t	25	ns
Abanico de Salida	-	50	-
Margen de Ruido	-	3	V

Tabla D.16: Características Eléctricas del los C.I. 74HC08, 74HC32 y 74HC04

D.10 Especificaciones Eléctricas del MCU68HC11F1

El microprocesador MCU68HC11G1 tiene un circuito protector contra grandes voltajes estáticos o campos eléctricos. Pero a pesar de este circuito protector se deben tener las precauciones necesarias para evitar que el microprocesador se queme o no llegue a operar adecuadamente.

D.10.1 Máximos Rangos

El microprocesador MCU68HC11F1 es un circuito electrónico con el cual deben tenerse ciertas precauciones antes de polarizarlo. En la Tabla D.17 se muestran los rangos máximo del MCU68HC11F1.

Rango	Símbolo	Valor	Unidad
Voltaje de Alimentación	V_{DD}	-0.3 a +7.0	V
Voltaje de Entrada	V_{IN}	-0.3 a +7.0	V
Rango de Temperatura Ambiental de Operación	T_A	T_L a T_H -40 a 85	°C
Rango de Temperatura de Conservación	T_{stg}	-55 a 150	°C
Drenaje de Corriente por Pin ⁽¹⁾ Excluyendo V_{DD} , V_{SS} , V_{RH} y V_{RL}	I_D	25	mA
Resistencia Térmica	θ_A	50	°C/W

Nota: (1) Un pin a la vez, observando los máximos límites de disipación de potencia.

Tabla D.17: Máximos Rangos del MCU68HC11F1

En la Tabla D.18 se muestran las características Eléctricas de DC del MCU68HC11F1, considerando que $V_{DD} = 5.0 V_{DC} \pm 10\%$, $V_{SS} = 0 V_{DC}$ y $T_A = T_L \text{ a } T_H$:

Características	Símb.	Mín.	Máx.	Unl.
Voltaje de Salida $I_{L_{max}} = \pm 10.0 \mu A$ ⁽¹⁾ Todas las salidas excepto RESET y MODA	V_{OL} V_{OH}	— $V_{DD}-0.1$	0.1 —	V
Voltaje Alto de Salida $I_{L_{max}} = -0.8mA, V_{DD} = 4.5 V$ ⁽²⁾ Todas las salidas excepto RESET, XTAL y MODA	V_{OH}	$V_{DD}-0.8$	—	V
Voltaje Bajo de Salida $I_{L_{max}} = 1.6 mA$ Todas las salidas excepto XTAL	V_{OL}	—	0.4	V
Voltaje Alto de Entrada Todas las salidas excepto RESET	V_{IH}	$0.7 \cdot V_{DD}$	V_{DD} V_{DD}	V
Voltaje Bajo de Entrada Todas las entradas	V_{IL}	$0.2 \cdot V_{DD}$	$0.2V_{DD}$	V
Puerto E/S, Salida 3 Estad. PA7-PA0, PC7-PC0, PD5-PD0, PG7-PG0 $V = V_{IH}$ ó V_{IL} MODA LIR, RESET	I_{OZ}	—	± 10	μA
Corriente de Entrada ⁽²⁾ $V_{in} = V_{DD}$ o V_{SS} IRQ, XIRQ $V_{in} = V_{DD}$ o V_{SS} MODB V_{stdy}	I_{in}	— —	± 1 ± 10	μA
Voltaje de RAM en Standby Suspendido	V_{SB}	4.0	V_{DD}	V
Corriente de RAM en Standby Suspendido	I_{SB}	—	20	μA
Corriente de Alimentación Total ⁽³⁾ RUN Modo Solo Chip Modo No-Multiplexado Expandido WAIT Modo Solo Chip Modo No-Multiplexado Expandido STOP (Sin Reloj) Modo de Solo Chip	I_{DD} W_{DD} S_{DD}	— — —	15 27 6 10 100	mA mA mA mA μA
Capacitancia de Entrada PE7-PE0, IRQ, XIRQ, EXTAL PA7-PA0, PC7-PC0, PD5-PD0, PG7-PG0, MODA LIR, RESET	C_{in}	— —	8 12	pF
Disipación de Potencia Modo de Solo Chip Modo No-Multiplexado Expandido	P_D	— —	85 150	mW

Notas 1) La especificación V_{OH} para RESET y MODA no se aplican porque son pines de compuerta abierta.

Las especificaciones V_{OH} no se aplican a los puertos C, D y G en el Modo OR alámbrado.

2) Ver las especificaciones del A/D para las corrientes de fuga del Puerto E.

3) Todos los puertos se configuran como entradas:

$V_{IL} = 0.2$; $V_{IH} = V_{DD} - 0.2V$; No hay cargas dc; XTAL se controla con una onda cuadrada; y $t_{\text{osc}} = 476.5 \text{ ns}$.

Tabla D.18: Características Eléctricas de DC del MCU68HC11F1

D.10.3 Control de Tiempos

En la Tabla D.19 se muestra el Control de Tiempos del MCU68HC11F1, considerando sólo una frecuencia de reloj E de 2.0 MHz (para frecuencias de 1.0 MHz y 2.1 MHz se utilizan otros parámetros).

Características		Simb.	2.0 MHz		Unid.	
		.	Mín	Máx		
Frecuencia de Operación		f_0	dc	2.0	MHz	
Periodo del Reloj E		t_{cyt}	500	-	ns	
Frecuencia del Cristal		f_{XTAL}	-	8.0	MHz	
Frecuencia del Oscilador Externo		$4f_0$	dc	8.0	MHz	
Tiempo de Instalación del Control de procesador		$t_{PCS} = 1.4t_{Cic} - 50 \text{ ns}$	t_{PCS}	75	-	ns
Ancho de Pulso de Entrada del Reset ⁽¹⁾		(Para garantizar el Vector de Reset Externo) (Tiempo de entrada mínimo; se limpia con reset interno)	PW _{RSTL}	8	-	t_{cyt}
				1	-	
Tiempo de Instalación en el Modo de Programación		t_{MPS}	2	-	t_{cyt}	
Tiempo de Retención en el Modo de Programación		t_{MPH}	0	-	ns	
Ancho de Pulso de Interrupción Flanco, IRQ Modo Sensitivo		$PW_{IRQ} = t_{Cic} + 20 \text{ ns}$	PW _{IRQ}	520	-	ns
Tiempo de Inicio en la Espera de Recuperación		t_{WRS}	-	4	t_{cyt}	
Tiempo Ancho de Pulso de Entrada de Captura, Entrada del Acumulador de Pulsos		$PW_{TDM} = t_{Cic} + 20 \text{ ns}$	PW _{TDM}	520	-	ns

Nota: (1) El RESET no se puede reconocer durante el primer ciclo de reloj, ya que éste se mantiene bajo. La circuitería interna controla el pin verificado bajo para los cuatro ciclos de reloj.

Tabla D.19: Control de Tiempos del MCU68HC11F1

D.10.4 Tiempos de los Puertos Periféricos

En la Tabla D.20 se muestran los Tiempos de los Puertos Periféricos, considerando sólo una frecuencia de reloj E de 2.0 MHz (para frecuencias de 1.0 MHz y 2.1 MHz se utilizan otros parámetros).

Características	Simb.	2.0 MHz		Unid.
		Mín	Máx	
Frecuencia de Operación (Frecuencia del Reloj E)	f_o	2.0	2.0	MHz
Periodo del Reloj E	t_{cyt}	500	-	ns
Tiempo de Instalación de Datos Periféricos (Lectura del MCU por los puertos A, C, D, E y G).	t_{PDH}	100	100	ns
Tiempo de Retención de Datos Periféricos (Lectura del MCU por los puertos A, C, D, E y G).	t_{PDH}	50	-	ns
Tiempo de Retardo, Escritura de Datos Periféricos (MCU escribe al puerto A) (MCU escribe a los puertos B, C, F y G) $t_{PDW} = 1.4t_{cyt} + 100ns$	t_{PDW}	- -	200 225	ns

Tabla D.20: Tiempos de los Puertos Periféricos del MCU68HC11F1

D.10.5 Características del Convertidor Analógico/Digital

En la Tabla D.21 se muestran las características del convertidor A/D del MCU68HC11F1, considerando $V_{DD} = 5.0 V_{dc} \pm 10\%$, $V_{SS} = 0 V_{dc}$, $T_A = T_L$ a T_H , $750 \text{ KHz} \leq E \leq 2.1 \text{ MHz}$.

Características	Parámetro	Mín.	Abs.	Máx.	Unid.
Resolución	Números de bits resueltos por el A/D.	-	8	-	Bits
No-Linealidad	Máxima Desviación de las Características de Transferencias Ideales del A/D.	-	-	$\pm 1/2$	LSB
Error Cero	Diferencia entre la salida de un A/D Ideal y un A/D Actual para un Voltaje de Entrada Cero.	-	-	$\pm 1/2$	LSB
Error de Escala Completa	Diferencia entre la salida de un A/D Ideal y un A/D Actual para un Voltaje de Entrada Completa.	-	-	$\pm 1/2$	LSB
Error de Desajuste Total	Máxima Suma de No-Linealidad, Error Cero y Error de Escala Completa.	-	-	$\pm 1/2$	LSB

Características del Convertidor Analógica/Digital (Continuación)

Error de Cuantización	Incertidumbre, debido a la resolución del Convertidor.			$\pm \%$	LSB
Exactitud Absoluta	Diferencia entre el Voltaje de Entrada Actual y el equivalente de Escala Completa del Código de Salida Binario. Todas las Fuentes de Error Incluidas: a) Reloj E. b) Un Oscilador RC Interno.			± 1 ± 2	LSB
Rango de Conversión	Rango de Voltaje de Entrada Analógica.	V_{RL}		V_{RH}	V
V_{RH}	Máximo Voltaje de Referencia Analógica. ⁽¹⁾	V_{RL}		$V_{DD}+0.1$	V
V_{RL}	Mínimo Voltaje de Referencia Analógica. ⁽²⁾	$V_{SS}-0.1$		V_{RH}	V
ΔV_R	Mínima Diferencia entre V_{RH} y V_{RL} . ⁽²⁾	3			V
Tiempo de Conversión	Tiempo Total para hacer una Sola Conversión Analógica a Digital: a) Reloj E. b) Un Oscilador RC Interno.		32	$t_{cl} + 32$	t_{cl} μs
Monotonocidad	El Resultado de la Conversión Nunca se Decrementa con un Incremento en el Voltaje de Entrada y No Tiene Códigos Faltantes.		Garantizada		
Lectura de Entrada Cero	Resultado de la Conversión cuando $V_{in}=V_{RL}$.	00			HEX
Lectura de Escala Completa	Resultado de la Conversión cuando $V_{in}=V_{RH}$.			FF	HEX
Tiempo de Adquisición de Muestreo	Tiempo de Muestreo en la Adquisición de la Entrada Analógica: a) Reloj E. b) Un Oscilador RC Interno.		12	12	t_{cyc} μs
Capacitancia de Retención de Muestreo	Capacitancia de Entrada Durante el Muestreo: PE7-PE0.		20 (Typ)		μF
Fuga de Entrada	Fuga de Entrada en los Pines A/D: PE7-PE0 V_{RL} , V_{RH}			400 1.0	nA μA

Notas:

- (1) Fuentes de Impedancias más grandes de 10 K Ω afectarán la exactitud de las mediciones, debido principalmente a fugas de la entrada.
- (2) Funciona verificado abajo de 2.5 V ΔV_R , pero la exactitud está probada y garantizada con $\Delta V_R=5V \pm 10\%$.

Tabla D.21: Características del Convertidor A/D del MCU68HC11F1

D.10.6 Tiempos del Bus

En la Tabla D.22 se muestran los Tiempos del Bus del MCU68HC11F1, considerando $V_{DD} = 5.0 V_{dc} \pm 10\%$, $V_{SS} = 0 V_{dc}$, $T_A = T_L$ a T_H .

N°.	Características	Símb.	2.1 MHz		Unid.
			Min	Max	
1	Tiempo del Ciclo ⁽¹⁾	t_{CYC}	476	-	ns
2	Ancho de Pulso, E bajo ($1.2 t_{CYC} - 23$ ns)	PW_{EL}	215	-	ns
3	Ancho de Pulso, E alto ($1.2 t_{CYC} - 23$ ns)	PW_{EH}	210	-	ns
4	Tiempo E de levantamiento y Caída	t_{pH}	-	20	ns
9	Tiempo de retención de las direcciones ($1.8 t_{CYC}$)	t_{AR}	60	-	ns
11	Tiempo de retardo de las direcciones ($1.8 t_{CYC} + 60$ ns)	t_{AD}	-	120	ns
12	Tiempo Válido de las direcciones ($PW_{EL} - t_{AD}$)	t_{AV}	95	-	ns
17	Tiempo de Instalación de Datos de Lectura	t_{DSR}	30	-	ns
18	Tiempo de Retención de Datos de Lectura	t_{DRR}	0	-	ns
19	Tiempo de Retardo de Datos de Escritura	t_{DOW}	-	40	ns
21	Tiempo de Retención de Datos de Escritura	t_{DRW}	30	-	ns
29	Tiempo de Acceso a las Direcciones del MPU ($t_{AV} + t_p + PW_{EL} - t_{DSR}$) ⁽¹⁾	t_{ACCA}	295	-	ns
39	Tiempo de Inicio en la Escritura de Datos ($PW_{EH} - t_{DOW}$) ⁽¹⁾	t_{DSW}	170	-	ns
50	Tiempo E de Retardo Válido del Chip Select	t_{ECSO}	-	50	ns
51	Tiempo E de Acceso Válido del Chip Select ($PW_{EH} - t_{ECSO} - t_{DSR}$) ⁽¹⁾	t_{ECSA}	130	-	ns
52	Tiempo de Retención del Chip Select	t_{CR}	0	20	ns
54	Tiempo de Retardo Válido de Direcciones del Chip Select ($14 t_{CYC} + 40$ ns)	t_{ACSD}	-	160	ns
55	Tiempo de Acceso Válido a las Direcciones del Chip Select ($t_{CYC} - t_{ACSD} - t_p$) ⁽¹⁾	t_{ACSA}	267	-	ns
56	Tiempo de Selección Válida para el Chip Select	t_{AVCS}	0	-	ns
57	Tiempo de Tres Estados Válido para Direcciones de Datos	t_{AVDZ}	-	10	ns

Nota: (1) Indica un parámetro afectado por la extensión del tiempo. Sumando $n < t_{CYC}$ al valor del parámetro donde $n=1, 2, 3$ (dependiendo de los valores escritos en el registro CSSTRH).

Tabla D.22: Tiempos de Bus del MCU68HC11F1

D.10.7 Direcciones de Vectores

En la Tabla D.23 se muestran todas las direcciones de los vectores que tiene el MCU68HC11F1 en el modo Bootstrap. Cuando se quiera hacer una interrupción en el MCU68HC11F1 se debe hacer referencia a la dirección del vector donde se quiera generar la interrupción..

Direcciones	Vectores
00C4	SCI
00C7	SPI
00CA	Borde de Entrada del Acumulador de Pulso
00CD	Desbordamiento del Acumulador de Pulso
00D0	Desbordamiento del Temporizador
00D3	Salida del Comparador 5 del Temporizador y Entrada de Captura 4
00D6	Salida del Comparador 4 del Temporizador
00D9	Salida del Comparador 3 del Temporizador
00DC	Salida del Comparador 2 del Temporizador
00DF	Salida del Comparador 1 del Temporizador
00E2	Entrada de Captura 3 del Temporizador
00E5	Entrada de Captura 2 del Temporizador
00E8	Entrada de Captura 1 del Temporizador
00EB	Interrupción Tiempo-Real
00EE	IRQ
00F1	XIRQ
00F4	SWI
00F7	Opcode Ilegal
00FA	Falla del COP
00FD	Monitor del Reloj
BF00 (Boot)	Reset

Tabla D.23: Direcciones de Vectores del MCU68HC11F1

D.10.8 Nombres de los Pines

En la Tabla D.24 se muestran los nombres asignados a cada uno de los 68 pines del MCU68HC11F1.

Pín	Nombre	Pín	Nombre	Pín	Nombre	Pín	Nombre
1	V _{ss}	18	XIRQ	35	PA7/OC1/ PAI	52	PF6/A6
2	MODB/V _{STBY}	19	IRQ	36	PA6/OC1/OC2	53	PF5/A5
3	MODA/LIR	20	PG7/CSPRG	37	PA5/OC2/OC3	54	PF4/A4
4	E	21	PG6/CSGEN	38	PA4/OC1/OC4	55	PF3/A3
5	R/W	22	PG5/CSIO1	39	PA3/OC1/IC4/ OC5	56	PF2/A2
6	EXTAL	23	PG4/CSIO2	40	PA2/IC1	57	PF1/A1
7	XTAL	24	PG3	41	PA1/IC2	58	PF0/A0
8	4XOUT	25	PG2	42	PA0/IC3	59	PE0/AN0
9	PC0/D0	26	PG1	43	PB7/A15	60	PE4/AN4
10	PC1/D1	27	PG0	44	PB6/A14	61	PE1/AN1
11	PC2/D2	28	PD0/RxW	45	PB5/A13	62	PE5/AN5
12	PC3/D3	29	PD1/TxD	46	PB4/A12	63	PE2/AN2
13	PC4/D4	30	PD2/MISO	47	PB3/A11	64	PE6/AN6
14	PC5/D5	31	PD3/MOSO	48	PB2/A10	65	PE3/AN3
15	PC6/D6	32	PD4/SCK	49	PB1/A9	66	PE7/AN7
16	PC7/D7	33	PD5/SS	50	PB0/A8	67	V _{RL}
17	RESET	34	V _{DD}	51	PF7/A7	68	V _{RH}

Tabla D.24: Pines del MCU68HC11F1

Apéndice E: Circuitos Eléctricos

E.1 Circuito del Acondicionador de la Señal Eléctrica

En el circuito del acondicionador de la señal eléctrica, el amplificador operacional de la red atenuadora cambia de ganancia de acuerdo a la señal de entrada, variando la resistencia de entrada así como la de retroalimentación por medio de la lógica de control.

El circuito rectificador de precisión se implementó con amplificadores operacionales de ganancia unitaria. El offset del circuito se manejó mediante una fase de auto-cero del sistema para obtener la compensación requerida.

Los dispositivos utilizados, en su mayoría son tipo CMOS, para tener una menor disipación de potencia, y una mayor eficiencia del circuito en conjunto. En la Figura E.1 se muestra el diagrama completo del circuito acondicionador de la señal eléctrica.

El módulo de

osciloscopio

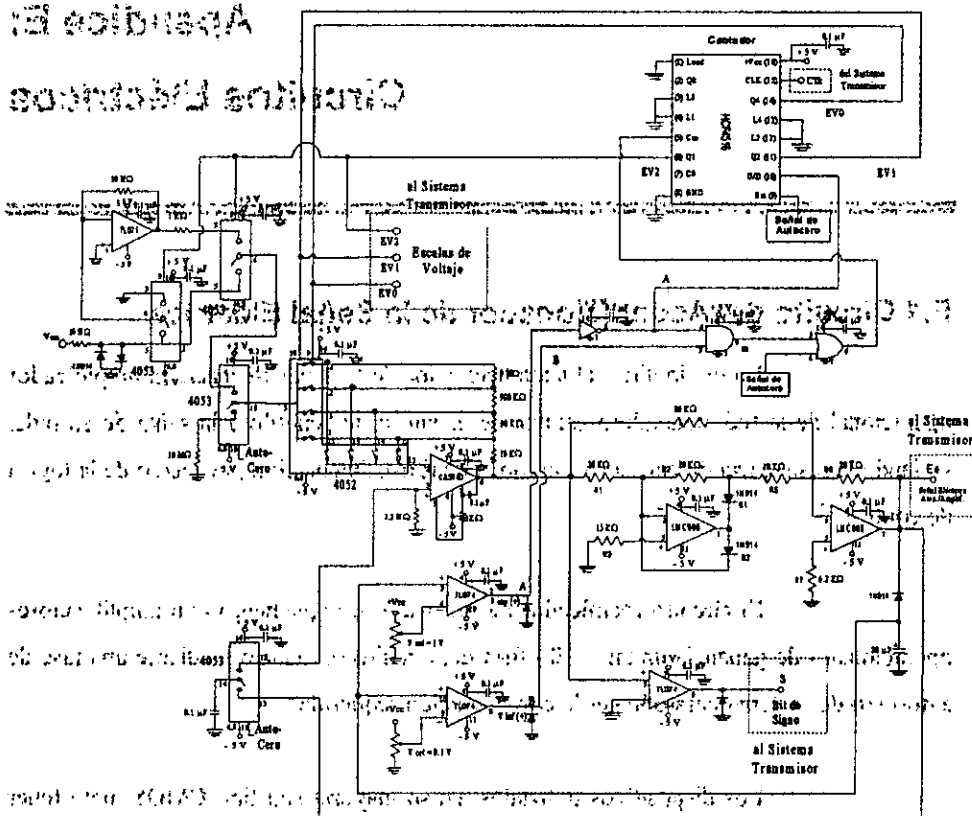


Figura E.1: Diagrama del Circuito Acondicionador de la Señal Eléctrica

E.2 Circuito Eléctrico del Transmisor

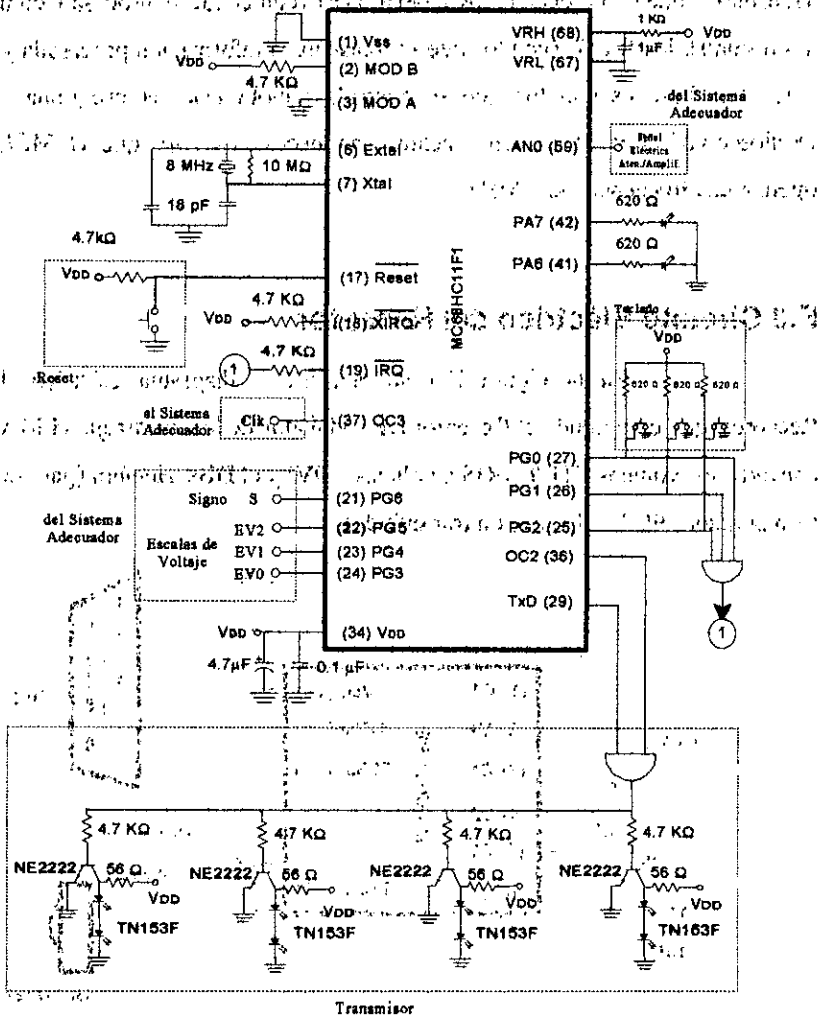


Figura E.2: Diagrama del Circuito Transmisor

En la Figura E.2 se muestra el diagrama eléctrico del Sistema Transmisor, que comprende el MCU68HC11F1 (con el que se procesa y codifica la señal a transmitir), los LEDs (con los que se transmite la información procesada y codificada) y los tres botones con los que se controla remotamente el programa "Módulo de Osciloscopio". Los otros componentes son necesarios para que el MCU68HC11F1 opere a una frecuencia de 2 MHz.

E.3 Circuito Eléctrico del Receptor

En la Figura E.3 se muestra el diagrama eléctrico del Circuito Receptor, que comprende el Receptor/Modulador de luz infrarroja, el MAX232 (para convertir los voltajes TTL/CMOS a voltajes $\pm 9V$) y el DB9-Hembra (para que entre a la Computadora PC la información transmitida).

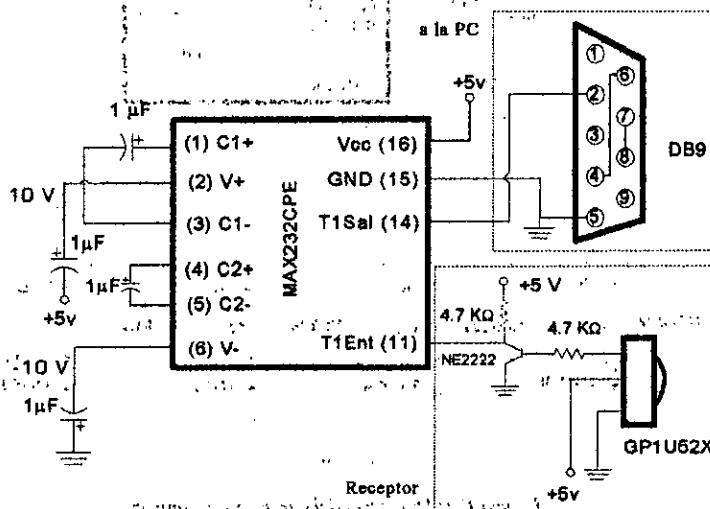


Figura E.3: Diagrama del Circuito Receptor

E.4 Parámetros del Módulo de Osciloscopio

Cabe recordar que el Módulo de Osciloscopio consta del Sistema Adecuador de la Señal Eléctrica, el Sistema Codificador/Decodificador, el Sistema Receptor y el Sistema Transmisor. Haciendo una recopilación de las características de cada sistema se obtienen los siguientes parámetros.

- ◆ Polarización del Sistema Adecuador de la Señal Eléctrica: $\pm 5\text{ V}$.
- ◆ Impedancia de Entrada: $10\text{ M}\Omega$
- ◆ Número de Bits de Control para el Auto-rango: 3 bits.
- ◆ Máximo Voltaje de Entrada: 1000 V .
- ◆ Mínimo Voltaje de Entrada: $3.9\text{ }\mu\text{V}$.
- ◆ Corriente Máxima de Entrada: $100\text{ mA @ }1000\text{ V}$.
- ◆ Corriente Mínima de Entrada: $400\text{ }\mu\text{A @ }3.9\text{ }\mu\text{V}$.
- ◆ Velocidad de Operación del Sistema Adecuador: 54 KHz .
- ◆ Escalas de Atenuación del Sistema Adecuador: 4 ($\times 1$, $\times 0.1$, $\times 0.01$ y $\times 0.001$).
- ◆ Escalas de Amplificación del Sistema Adecuador: 4 ($\times 10$, $\times 100$, $\times 1000$ y $\times 10000$).
- ◆ Resolución del Convertidor A/D: 8 bits.
- ◆ Frecuencia de Muestreo: 350 Hz .
- ◆ Código empleado para la Detección/Corrección de Errores: BCH Binario (63, 16, 23).
- ◆ Número de Errores detectados en una palabra codificada de 63 bits: 11 errores.
- ◆ Polarización del Transmisor: 5 V .
- ◆ Medio de Transmisión: Luz Infrarroja.
- ◆ Máximo Alcance de Transmisión: 5 m .
- ◆ Ángulo Promedio de Radiación: 7° .

- ◆ **Polarización del Receptor:** *500 V/div, 100 V/div, 50 V/div, 20 V/div, 10 V/div, 5 V/div, 2 V/div, 1 V/div, 500 mV/div, 200 mV/div, 100 mV/div, 50 mV/div, 20 mV/div, 10 mV/div, 5 mV/div, 2 mV/div, 1 mV/div, 500 μ V/div, 200 μ V/div y 100 μ V/div).*
- ◆ **Ventajas del Software:** *Almacenamiento e Impresión de la Señal Eléctrica.*

Bibliografía

1. *Backwith, Tomas G. y Marangoni, Roy D.*
Mechanical Measurements
Addison-Wesley
2. *Morris, Alain S.*
Principles of Measurement and Instrumentation
Prentice Hall
3. *Wolf, Stanley*
Guía para mediciones Electrónicas y Prácticas de Laboratorio
Prentice Hall
4. *Zapata Ferrer, Angel R.*
El Osciloscopio y sus Aplicaciones
Limusa
5. *Arcila Rodríguez, Wilbert y Vidal Macedo, J. Lino*
Apuntes de Análisis de Circuitos Eléctricos (Primera Parte)
Facultad de Ingeniería Universidad Nacional Autónoma de México
1985
6. *Mileaf, Harry*
Curso Práctico de Electrónica, Volumen 1
Ediciones Ciencia y Técnica, S.A.
Limusa
1990

7. *Sedra, A. y Smith, K.C.*
Dispositivos Electrónicos y Amplificación de Señales
McGraw-Hill
1989
8. *Wolf, Stanley y Smith, Richard F. M.*
Guía para Mediciones Eléctricas y Prácticas de Laboratorio
Prentice-Hall
1992
9. *Interconexión de Periféricos a Microprocesadores*
Serie Mundo Electrónico
Publicaciones Marcombo
1984
10. *Jung, Walter G.*
IC Op-Amp CookBook
Segunda Edición, 1981
11. *Lancaster, Don*
CMOS CookBook
Editorial Howard W. Sams & Company,
Segunda Edición, 1988
12. *Nashelsky, Louis y Boylestad, Robert*
Electrónica Teoría de Circuitos
Editorial Prentice Hall
Cuarta Edición, 1989
13. *Morris Mano, M.*
Diseño Digital
Editorial Prentice Hall
Primera Edición, 1987

14. *Berlin, Howard M.*
Op-Amp Circuits and Principles
Editorial SAMS
Primera edición, 1991
15. *Berlin, Howard M. y Getz, Frank C.*
Fundamentals of Amplifiers & Linear Integrated Circuits
Editorial Maxwell MacMillan
Tercera Edición
16. *Coughlin, Robert F. y F., Frederick*
Operational Amplifiers & Linear Integrated Circuits
Editorial Prentice Hall
17. *Couch, Leon W.*
Digital and Analog Communication Systems
Macmillan
Cuarta Edición
18. *Halsall, Fred*
Data Communications, Computer Networks and Open Systems
Addison-Wesley
Tercera Ediciónm 1992
19. *Martin, James*
Telecommunications and the Computer
Prentice-Hall
Segunda Edición, 1976
20. *Noll, A. Michael*
Introduction to Telecommunications Electronics
Artech House
1988

21. *Schoenbeck, Robert J.*
Electronic Communications: Modulation and Transmission
Merrill Publishing Company
1988
22. *Adámek, Jiri*
Foundations of Coding: Theory and Applications of Error-Correcting Codes with an Introduction to Cryptography and Information Theory
Awiley-Interscience Publication; John Wiley & Sons
23. *Anderson, John B. y Mohan, Seshadri*
Source and Channel Coding: An Algorithmic Approach
Kluwer Academic Publishers
1991
24. Essentials of Error Control Coding Techniques
Editado por Hideki Imai
Academic Press
1990
25. M68HC11: Reference Manual
Motorola
1991
26. MC68HC11 PCbug11: User's Manual
Motorola
Primera Edición, 1991
27. MC68HC11F1: High-Density Complementary Metal Oxide Semiconductor (HCMOS) Microcontroller Unit
Motorola
1989

-
28. *Savant, C. J.; Roden, Martin S. y Carpenter, Gordon L.*
Diseño Electrónico: Circuitos y Sistemas
Addison-Wesley Iberoamericana
Segunda Edición, 1992
 29. *Ross, Nelson*
Guía Completa de Visual Basic para Windows: Una Introducción Práctica a la programación con Windows
McGraw-Hill
1993
 30. *Heyman, Mark Steven*
La Esencia de Visual Basic 4
Sams Publishing, Prentice-Hall
1995
 31. Programmer's Guide
Microsoft Visual Basic; Programming System for Windows, Versión 4.0
Operating Environment
Microsoft Corporation
1995
 32. Professional Features; Custom Control Reference
Microsoft Visual Basic; Programming System for Windows, Versión 4.0
Professional Edition
Microsoft Corporation
1995
 33. Language Reference
Microsoft Visual Basic; Programming System for Windows, Versión 4.0
Microsoft Corporation
1995
 34. *Norton, Peter*
Toda la PC: Todo lo que siempre ha deseado saber sobre su PC
Prentice Hall
5ª Edición, 1993

35. *Foster, Dennis L.*

El IBM PS/2. Descripción, uso y aplicaciones.

McGraw-Hill

1988