

**DISEÑO E IMPLEMENTACION DE UN ENTRENADOR EN MODULACION PSK
Y QAM PARA EL AREA DE COMUNICACIONES ELECTRONICAS DE LA
TECNOLOGICA DE BOLIVAR**

GERMAN HERNANDEZ CASTILLO

**UNIVERSIDAD TECNOLOGICA DE BOLIVAR
FACULTAD DE INGENIERIA ELECTRICA, ELECTRONICA Y MECATRONICA
CARTAGENA DE INDIAS**

2004

**DISEÑO E IMPLEMENTACION DE UN ENTRENADOR EN MODULACION PSK
Y QAM PARA EL AREA DE COMUNICACIONES ELECTRONICAS DE LA
TECNOLOGICA DE BOLIVAR**

GERMAN HERNANDEZ

Trabajo de grado presentado para
optar al título de Ingeniero Electrónico

DIRECTOR

Oscar Santos

Ingeniero Electrónico

**UNIVERSIDAD TECNOLOGICA DE BOLIVAR
FACULTAD DE INGENIERIA ELECTRICA, ELECTRONICA Y MECATRONICA**

CARTAGENA

2004

Cartagena, 16 de Agosto de 2003

Señores:
COMITÉ DE PROYECTOS DE GRADO
Facultad De Ingeniería Eléctrica, Electrónica Y Mecatrónica
Universidad Tecnológica de Bolívar
L.C.

Apreciados Señores:

Por medio de la presente me permito hacer la entrega formal del proyecto de grado titulado “DISEÑO E IMPLEMENTACION DE UN ENTRENADOR EN MODULACION PSK Y QAM PARA EL AREA DE COMUNICACIONES ELECTRONICAS DE LA TECNOLOGICA DE BOLIVAR” que será presentado por el estudiante GERMAN DUVAN HERNANDEZ CASTILLO, el cual estuvo bajo mi supervisión como asesor del mismo para optar el titulo de INGENIERO ELECTRÓNICO.

Atentamente,

OSCAR SANTOS
Ingeniero Electrónico / Director del proyecto

Cartagena, 15 de Marzo de 2004

Señores:

COMITÉ DE PROYECTO DE GRADO

Facultad de Ingeniería Eléctrica y Electrónica
Corporación Universitaria Tecnológica de Bolívar
L.C.

Apreciados Señores:

Hago entrega formal de mi proyecto de grado titulado “DISEÑO E IMPLEMENTACION DE UN ENTRENADOR EN MODULACION PSK Y QAM PARA EL AREA DE COMUNICACIONES ELECTRONICAS DE LA TECNOLOGICA DE BOLIVAR”, como requisito para optar el título de INGENIERO ELECTRÓNICO.

Esperamos que éste cumpla con las normas y requerimientos exigidos por la facultad.

Atentamente,

GERMAN DUVAN HERNANDEZ CASTILLO
Código 9904359

Nota de aceptación

Presidente del jurado

Jurado

Jurado

Ciudad y fecha

...A mi familia quien me apoyo en los instantes más difíciles y a todos aquellos profesionales que brindaron sin costo alguno parte de su conocimiento...

Agradezco a todos aquellos especialistas que, a veces, sin considerar el factor monetario colaboraron con la realización de este proyecto; no solo a nivel local, sino a nivel internacional; expertos de USA., Europa, Brasil, que orientaron en todo momento no solo por norma de trabajo sino por fraternidad profesional

CONTENIDO

1. Introducción.....	22
2. Marco Teórico.....	24
2.1 Modulación por desplazamiento de fase	25
2.1.1 Teoría de la modulación por desplazamiento de fase.....	25
2.1.2 <i>PSK (Phase Shift Keying)</i>	27
2.1.3 <i>QPSK</i> . Desplazamiento de fase en cuadratura.....	29
2.7 Experiencias a realizar.....	38
3. Resultados de la investigación	42
4. Elementos a utilizar.....	43
5. Desarrollo	46
5.1 Estructura en bloques funcionales del entrenador.....	46
5.1.1 Fuente de Alimentación.....	46
5.1.2. Sección de modulación y demodulación.....	60
5.1.3 El <i>PIC16F877</i>	83
5.1.4 Base de Tiempo. Generador de Reloj.....	103
5.1.5 Teclado de Usuario. Interfase teclas de usuario.....	106
5.1.6 Puerto E/S. <i>USART/RS-232-C</i> y <i>MAX232</i>	106
5.1.7 Módulos de Radio.....	108
5.1.8 Visualizador <i>LCD OPTREX DMC-50968</i>	112

6. Conclusiones	115
7. Observaciones y recomendaciones.....	116
Bibliografía.....	209
Anexos ..	117

LISTA DE ANEXOS

Anexo A. Características Mecánicas del Prototipo	117
Anexo B. <i>ASIC</i> . Circuito Integrado de Aplicación Específica	120
Anexo C. Tutorial <i>UART</i>	133
Anexo D. La codificación <i>Reed Solomon</i>	148
Anexo E. Código <i>PN</i> . Código de Seudo Ruido	159
Anexo F. <i>Scrambler</i>	165
Anexo G. Programa del <i>PIC 16F877</i> implementado en lenguaje ensamblador ...	171
Anexo H. selección de la frecuencia de reloj	205
Anexo I. Comentarios	207

LISTA DE FIGURAS

Figura 1. Diagrama de bloques simplificado de un modulador <i>PSK</i>	28
Figura 2. Diagrama de constelación <i>QPSK</i>	30
Figura 3. Practicas a realizar con el entrenador	40
Figura 4. Diagrama de bloques del entrenador	46
Figura 5. <i>PCB</i> de la fuente de alimentación.....	49
Figura 6. Vista posterior en 3D de la <i>PCB</i> de la fuente de alimentación	50
Figura 7. Vista superior de la <i>PCB</i> de la fuente de alimentación armada.....	50
Figura 8. Dimensiones de la caja del entrenador	51
Figura 9. Circuito de la fuente de alimentación con carga <i>Thévenin</i> equivalente ..	54
Figura 10. Circuito equivalente sin diodo <i>Zener</i>	56
Figura 11. Esquema de la fuente de alimentación.....	58
Figura 12. Vista superior de la <i>PCB</i> del entrenador	59
Figura 13. Imagen de la <i>PCB</i> del entrenador	59
Figura 14. Diagrama de bloques del <i>S1109CR</i>	61
Figura 15. Circuito de interfase recomendado por <i>INTEL</i>	64
Figura 16. Opciones del microcontrolador	65
Figura 17. Diagrama de bloques del desmodulador	74
Figura 18. Características mecánicas del <i>ASIC S1109CR</i>	82
Figura 19. Vista frontal definitiva del entrenador	85

Figura 20. Mensaje de solicitud de dirección.....	87
Figura 21. Mensaje de error de registro	88
Figura 22. Bloque funcional del puerto <i>USART</i> en modo asíncrono	90
Figura 23. Diagrama de bloques de la sección transmisora del <i>USART</i> en modo asíncrono	93
Figura 24. Diagrama de bloques del receptor <i>USART</i>	98
Figura 25. Diagrama de bloques del programa del <i>PIC16F877</i>	101
Figura 26. Diagrama de bloques del divisor programable	105
Figura 27. Generador de Reloj implementado con el <i>74LS04</i>	105
Figura 28. El <i>MAX232</i>	107
Figura 29. Aspecto real de los módulos <i>RX2/TX2</i> de <i>Radiometrix</i>	108
Figura 30. Diagrama de bloques del <i>TX2</i>	110
Figura 31. Diagrama de bloques del <i>RX2</i>	111
Figura 32. Dimensiones físicas de los módulos <i>RX2/TX2</i>	111
Figura 33. Vista frontal del panel de indicadores	112
Figura 34. Disposición de terminales en el bus del <i>OPTREX DMC-50968</i>	113
Figura 1A. Vista Frontal.....	118
Figura 1B. Un circuito integrado a) encapsulado <i>PGA</i> b) el chip de silicón	120
Figura 1D. Diagrama de bloques simplificado de un sistema con codificación <i>Reed</i> <i>Solomon</i>	148
Figura 2A. Vista Posterior	118
Figura 2B. Pastilla de un <i>ASIC</i> basado en celdas (<i>CBIC</i>).....	125

Figura 1E. Espectro de frecuencia con código <i>PN</i>	162
Figura 1F. Ejemplo de <i>scrambling</i> y <i>descrambling</i> en forma serie.....	165
Figura 2D. Palabra clave <i>Reed Solomon</i>	150
Figura 2E. Secuencia <i>m</i>	164
Figura 2F. Ejemplo de <i>scrambling</i> en forma paralela.....	166
Figura 3A. Vista Superior.....	119
Figura 3B. Diseño simplificado de una celda estándar.....	126
Figura 3D. Palabra clave <i>Reed Solomon</i>	154
Figura 3F. Módulo <i>Scrambler</i> usado por el <i>S1109CR</i>	166
Figura 4A. Modulo Receptor Individual.....	119
Figura 4B. Asignación de rutas del <i>CBIC</i> descrito en la figura 39.	127
Figura 4D. Arquitectura general para decodificar códigos <i>Reed Solomon</i>	155
Figura 4F. <i>Scrambler</i> serie descrito en la figura 1F como referencia.....	167
Figura 5B. Pastilla de un arreglo de puertas acanalado.....	130
Figura 5F. Resultado de las operaciones.....	169
Figura 6B. Pastilla de un arreglo de puertas sin canal.....	131
Figura 7B. Una pastilla de arreglo de puertas incrustadas o estructuradas.....	131
Figura 8B. Pastilla de un dispositivo lógico programable (<i>PLD</i>).....	132

LISTA DE TABLAS

Tabla 1. Resultados para cada fase en un sistema <i>QPSK</i>	35
Tabla 2. Totalización de los consumos de corriente máximos	48
Tabla 3. Magnitudes de los valores de voltaje especificados	48
Tabla 4. Disipación de potencia en los reguladores integrados	58
Tabla 5. Cuadro sinóptico de las funciones más representativas del <i>S1109CR</i>	63
Tabla 6. Características mecánicas de los encapsulados <i>MQFP</i>	69
Tabla 7. Campos de datos de los registros de configuración.....	70
Tabla 8. Valor de <i>FCWSEL</i> utilizado para configuración por defecto	71
Tabla 9. Opciones de cerrado de datos.....	71
Tabla 10. Opciones del bus de datos múltiples para codificación de <i>bit</i>	72
Tabla 11. Parámetros del desmodulador o <i>scrambler</i>	73
Tabla 12. Parámetros del codificador <i>Reed-Solomon</i>	75
Tabla 13. Opciones de trazador de mapa de <i>bit</i>	76
Tabla 14. Control de codificador diferencial.....	76
Tabla 15. Codificación diferencial <i>QPSK</i> y desplazamiento de fase	77
Tabla 16. Selecciones de trazador de mapa de símbolos.....	77
Tabla 17. Trazador de mapa de símbolos.....	78
Tabla 18. Opciones de configuración del filtro <i>FIR</i>	78
Tabla 19. Almacenamiento de coeficientes del filtro <i>FIR</i>	79

Tabla 20. Control de nivel de la señal de interpolación	79
Tabla 21. Control de inversión de la señal	80
Tabla 22. Selección del <i>FCW</i>	80
Tabla 23. Características estándares de los encapsulados <i>MQFP</i>	81
Tabla 24. Características relevantes de los registros que intervienen en la determinación de la frecuencia en baudios de la transferencia de información en el modo asíncrono del <i>USART</i>	93
Tabla 25. Características más significativas de los registros asociados con la transmisión asíncrona	95
Tabla 26. Registro <i>TXSTA</i>	95
Tabla 27. Registro <i>RCSTA</i>	98
Tabla 28. Características más relevantes de los registros que intervienen en la recepción asíncrona	100
Tabla 29. Configuración de los registros para el modo básico	102
Tabla 1C. Asignaciones de señales en el estándar <i>RS-232-C</i>	141

GLOSARIO DE TERMINOS

ASIC: *Circuito Integrado de Aplicación Específica.* Circuito integrado especializado optimizado para una función particular

CODIGO REED SOLOMON: código de corrección de errores usado en equipos de comunicación inalámbrica y dispositivos de almacenamiento

CODIGOS EN BLOQUE: aquellos que separan el flujo de datos en bloques de k *bits* y a los cuales se le agregan $(n-k)$ *bits* de chequeo. En la literatura, esto es referenciado como un bloque de código (n,k) . Por ejemplo un código $(15,11)$ tiene 15 palabras de código, cuatro *bits* de paridad y el resto de *bits* de datos

CRITERIO DE BARKHAUSEN. Criterio que describe las condiciones necesarias para que un circuito oscile. Originalmente fue desarrollado para determinar la frecuencia de oscilación a usar en un radio transmisor

DATAPATH: bus de datos de señales múltiples.

DIAGRAMA DE BLOQUES: representación gráfico de un conjunto de ecuaciones, funciones estadísticas, procedimientos o estructuras que forman parte de un sistema **DC (Direct Current):** corriente directa o corriente continua (CC). Desde que tienen el mismo significado, cualquiera de los dos términos se usa a lo largo de texto

FPGA: *Field Programable Gate Array.* Matriz de Puertas de Campo Programable

LCD: Visualizador de cristal líquido. Unidad de presentación visual alfanumérica basada en la polarización de elementos ópticos mediante *polaroides* y aplicación de corriente eléctrica

LED: *Diodo Emisor de Luz.* Elemento semiconductor similar a los diodos convencionales pero cuyas propiedades físicas le permite emitir energía en forma luminosa en vez de calorífica como sucede con los diodos comunes

MQFP: *METRIC QUAD FLAT PACK.* Encapsulado plano de cuadrángulo métrico

NRZ: código de no retorno a cero. En este tipo de *codificación de línea*, el pulso binario se mantiene por el total de tiempo de *bit*.

PSEUDO RUIDO: (*PSUDO NOISE*). Un código pseudo ruido es aquel que tiene un espectro similar a una secuencia aleatoria de bits pero es determinísticamente generado.

PRV: (*PEAK-REVERSE-VOLTAGE*). Voltaje de pico inverso: máximo voltaje en polarización inversa que puede soportar un diodo

RS-232-C: especificación de la *EIA* también reflejo del estándar v.24 de la *CCITT* utilizado ampliamente para la comunicación asíncrona o *UART* (sin necesidad de señal de reloj para la sincronización) a través de los puertos seriales de un ordenador

SCRAMBLER: dispositivo que traspone o invierte señales o también codifica un mensaje al transmisor para hacer el mensaje incomprensible a un receptor no equipado con un apropiado *set* de dispositivos de *descrambling*

SMD: (*SURFACE MOUNTING DEVICE*). Siglas inglesas de dispositivo de montaje superficial. Dispositivo en que las terminales no atraviesan la tarjeta de circuito impreso sino que van montados en su superficie a través de un compendio de técnicas especiales entre las que destacan, flujo de aire caliente y aplicación de luz infrarroja

TTL: *Lógica Transistor Transistor*. Tipo de circuitos integrados para aplicaciones digitales cuyas compuertas están construidas a base de transistores bipolares (por lo menos originalmente) en toda su estructura y que forman parte de las familias lógicas más usadas

UART: *Universal Asynchronous Receiver Transmitter*. Siglas de Transmisor Receptor Asíncrono Universal

VHDL: estándar de la *IEEE*. Lenguaje de simulación lógica digital, soportado por la mayoría de los proveedores *EDA*. No es muy diferente a *Verilog* pero este además soporta modelado a nivel de sistema y la simulación es mucho más integral que la de *Verilog*. El aprendizaje de *VHDL* es más largo y el código menos simple de aplicar. Algunos ingenieros de hardware tienen que dominar ambos lenguajes debido al incremento en uso de los Bloques de propiedad intelectual (*IP*), los cuales pueden estar no escritos en su lenguaje favorito

VERILOG: estándar de la *IEEE*. Lenguaje de simulación lógica digital, soportado por la mayoría de los proveedores *EDA*, usado para diseñar circuitos *ASIC* y para la simulación de sistemas

RESUMEN

El *entrenador en modulación digital* es un equipo creado, en síntesis, para desarrollar prácticas en el laboratorio de comunicaciones. Esta construido en base a tecnologías contemporáneas, algunas trabajadas en el transcurso de la carrera como los microcontroladores *PIC*, pantallas *LCD*, etc.. y otras que requirieron amplia investigación, las cuales realizan funciones estudiadas solo teóricamente, como la modulación digital (*QPSK*, *BPSK* y *16QAM*) y algunas mas, ni siquiera son tocadas por los libros didácticos empleados en la formación como la *Codificación Diferencial*, el *Scrambler*, la *Codificación Reed Salomón*, etc., conceptos que se expanden en los anexos del documento escrito. La razón es que estas funciones son integradas por un chip *VLSI* estilo *ASIC* desarrollado por *Stanford Telecom* y fabricado por *INTEL* el cual salió al mercado desde 1999. El entrenador en modulación digital, trabaja con tecnologías bastante elaboradas; en cuanto a escala de integración ; como el ya citado *STEL-1109*, circuito *MQFP* de 80 terminales corazón de este, el cual pertenece al campo de las tecnologías *VLSI* y como se ve en el *anexo B*, pertenece al conjunto de los *ASIC's*. Representa una ayuda para el estudiante del área de comunicaciones teniendo en cuenta los equipos con los que este contaba al inicio del proyecto.

Las otras tecnologías empleadas ya son mas comunes: microcontroladores *PIC*, pantallas *LCD*, *PCB*'s en doble faz de fibra de vidrio, módulos de radio híbridos, reguladores electrónicos, etc.

El entrenador está en capacidad de producir modulación digital en *BPSK*, *QPSK* y *16QAM*. Cada una de estas consiste en un tipo de modulación digital, las dos primeras son modulación de fase, mientras que la última es una técnica llamada modulación de amplitud en cuadratura. La modulación por desplazamiento de fase consiste en esto; la fase de la señal varía de acuerdo con un *set* de datos de entrada; en *BPSK*, que es la mas sencilla de todas, la fase varia entre 0 y 180 grados de acuerdo a si la entrada es un 0 binario o si es un 1 binario.

Para los otros tipos de modulación *PSK*, mientras mas cambios de fases hay, mas eficiente es, pero también es mayor la probabilidad de error al interpretar la señal, todo esto en función de los mecanismos de sincronización y detección de errores empleados. *QAM* difiere de *PSK* en que además de la fase de la señal portadora, también la magnitud cambia.

1. INTRODUCCION

El desarrollo práctico de los temas de la asignatura, es una herramienta básica como complemento de los recursos bibliográficos y la asesoría de los docentes para lograr las metas propuestas en esta área. La tendencia del mundo de las tecnologías electrónicas es ir optimizando cada vez más los sistemas de comunicaciones, diseñando e implementando modernas y potentes versiones de estos que mejoran las ya existentes y que les brindan mayor confiabilidad a sus usuarios. Factores como el desarrollo de la tecnología y la revolución de las comunicaciones hacen que los sistemas que son hoy de innovación, mañana se conviertan en sistemas atrasados y obsoletos, obligando a un constante cambio y a que los individuos estén a la vanguardia y desarrollen nuevos métodos que permitan solucionar los nuevos inconvenientes que se presentan a diario.

Los continuos avances han llevado a las instituciones de educación superior que tienen dentro de sus programas académicos carreras como la ingeniería electrónica, a crear una estructura interna que le permita mejorar la calidad de los futuros profesionales, lo cual hace importante la implementación de la modulación digital como una práctica de laboratorio para que los estudiantes apliquen los conceptos impartidos en el área de las comunicaciones electrónicas.

La Universidad Tecnológica de Bolívar es reconocida en la región por la calidad de la educación que se le brinda a cada uno de sus estudiantes, esto acompañado de modernas tecnologías y los medios que esta ofrece, la cual la ha llevado a estar entre las mejores universidades de la región, compitiendo con las mejores instituciones del país. Considerando lo anterior se pretende diseñar e implementar un modulador demodulador para que los estudiantes tengan una importante herramienta de instrucción que les permita aumentar aun más sus conocimientos en las técnicas de modulación digital en este caso la *BPSK*, *QPSK* y *16QAM*

2. MARCO TEORICO

Un *ASIC* realiza los tipos de modulación ya mencionados con las opciones de incorporar técnicas y mejoras recientes añadidas a la modulación digital como aleatorización y codificación. Este esta controlado e interfasado con el *PC* o con el usuario mediante un microcontrolador *PIC16F877*. Para la transmisión por radio, se utilizan módulos certificados para tal efecto, el transmisor recibe la señal modulada digitalmente y la transmite a un apropiado receptor. La frecuencia de portadora es de menos de *100 KHz*. Todo esto para garantizar la “observabilidad” de las formas de onda por parte del usuario y la correcta transmisión por los radio módulos que cortan en la entrada con un filtro pasabajos de 2º orden a *100 KHz*.

El modulo de *RF (FM-SAW)* hace la transmisión a frecuencias de *UHF* lo que garantiza una excelente transferencia de energía a través de dispositivos totalmente certificados por laboratorios Europeos. Para la practica cableada, una toma de cable coaxial debidamente adaptada, toma las muestras sin la etapa de radio; el respectivo par de conectores *BNC*, uno empotrado en el entrenador y otro adaptado al cable, están debidamente instalados. Fue necesario adquirir adicionalmente un módulo receptor para las transmisiones de radio del equipo en cuestión. Una interfase de control digital mediante un *PC* se añadió para facilitar el

manejo del equipo por parte de sus usuarios. Se implementan dos prácticas que definen las experiencias a realizar, estas se subdividen y arrojan finalmente cuatro laboratorios, como se indica mas adelante en experiencias a realizar.

2.1 MODULACION POR DESPLAZAMIENTO DE FASE (PSK)

2.1.1 TEORÍA DE LA MODULACIÓN POR DESPLAZAMIENTO DE FASE: una onda coseno puede ser representada por la ecuación

$$v(t) = V \cdot \cos(\omega t + \Omega) \quad (1)$$

Donde $v(t)$ es el valor instantáneo del voltaje de la onda, V es la amplitud de la onda, ω es la frecuencia angular de la onda, t es el tiempo y Ω es el desplazamiento de fase con respecto a una fase de referencia. En esta expresión las variables independientes son V , ω y Ω . Si cualquiera de estas variables varía en respuesta a otra señal, se dice que la onda está siendo modulada. Siendo V la amplitud, ω la frecuencia y Ω la fase de la onda, variando en el tiempo los valores de V se produce modulación de amplitud o *AM*, variando en el tiempo los valores de ω se produce modulación de frecuencia o *FM* y variando Ω se produce la modulación de fase o *PM*. La onda que se modula se conoce como portadora. Con

modulación de fase, la fase instantánea depende de la onda moduladora y está dada por:

$$q(t) = q + k.v_m(t) \quad (2)$$

Donde

- q es la fase de referencia de la portadora
- $v_m(t)$ es el voltaje modulador
- k es una constante (expresada en radianes/voltio).

Sea

$$v_m(t) = V_m.\text{sen}(w_m t) \quad (3)$$

Tomando $W = q(t)$ y reemplazando la expresión (2) en la (1) y luego sustituyendo (3), se obtiene la forma general para una portadora sinusoidal (coseno) con fase modulada por una onda sinusoidal (seno) moduladora

$$v(t) = V \cos(\mathbf{w}t + q + kv_m(t))$$

$$v(t) = V \cos(\mathbf{w}t + q + k.V_m.\text{sen}(w_m t))$$

$$v(t) = V \cos(\mathbf{w}t + q + \mathbf{m}.\text{sen}(w_m t))$$

Donde m es el *Índice de Modulación*. Generalmente se toma como referencia la fase de la portadora, por ejemplo $q = 0$, lo que da:

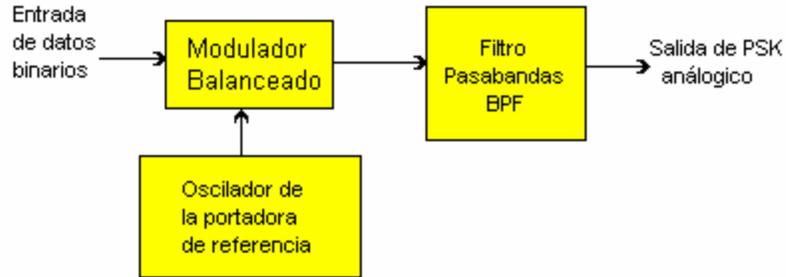
$$v(t) = V \cdot \cos(\omega t + m \cdot \text{sen}(\omega_m t))$$

Cuando la señal moduladora es una onda binaria, la señal transmitida es conmutada directamente de una fase a otra. El sistema se describe como *Modulación por Desplazamiento de Fase (PSK)*. Como la frecuencia es la velocidad de cambio de fase, modulación de frecuencia y fase son muy similares. En realidad, a menudo se denominan conjuntamente modulación de ángulo

2.1.2 PSK (PHASE SHIFT KEYING): es una forma de modulación de fase donde la onda moduladora es un torrente de datos digitales. En *PSK*, la señal transmitida tiene amplitud y frecuencia constantes, pero su fase, con respecto a una referencia, está directamente relacionada con el valor de una señal de datos binarios. Para una señal binaria de dos niveles, el desplazamiento de fase puede hacerse igual a 180 grados, esto es un desplazamiento de fase de 90 grados desde la referencia. En ese caso, el proceso de modulación es exactamente equivalente a *ASK* con portadora suprimida. El desplazamiento de fase también puede ser menor a 90 grados, como se estudiará en esta clase. Hay una variedad de métodos para producir una onda *PSK*. En el diagrama en bloques de abajo

se muestra un método común cuando se emplean desplazamientos de fase de 180 grados:

Figura 1. Diagrama de bloques simplificado de un modulador PSK



Tomassi, Wayne. Sistemas de Comunicaciones Electrónicas. Naucalpan de Juarez. Prentice Hall Hispanoamericana, 1996. 858 páginas.

El desplazamiento de fase no tiene que ser 180 grados, aunque en la práctica se emplea frecuentemente puesto que permite la máxima separación de los estados digitales.

Dos de los tipos más populares de circuitos los cuales son útiles sobre un amplio rango de frecuencias (hasta 4 Ghz, o cerca) son el *modulador balanceado* empleando diodos *Schottky*, o *switches FET* de doble puerta. A frecuencias relativamente bajas, *PSK* puede ser producido digitalmente, en vez de usar moduladores balanceados. Esto se hace empleando un registro de desplazamiento con el reloj dado por un oscilador a la frecuencia de la portadora Si se toman

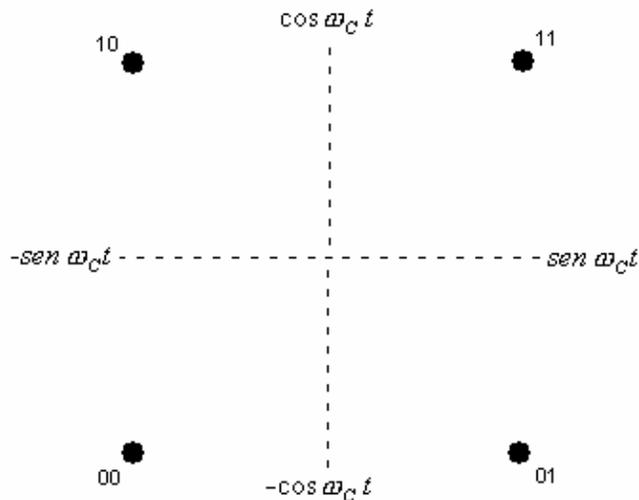
derivaciones en varios puntos a lo largo del registro de desplazamiento, se producen fases diferentes. Se puede emplear un *switch* para seleccionar las dos fases requeridas para los estados 1 y 0. Si el oscilador de reloj se hace variable, el desplazamiento de fase también puede hacerse variable.

Desplazamiento de Fase 90 grados: cuando el desplazamiento de fase entre los dos estados que representan el 1 y el 0 binario es de 90 grados, la onda de la señal transmitida será de la forma siguiente: No hay componente de *DC* en la onda moduladora, por lo tanto no habrá componente de frecuencia portadora (f_c) en la onda *PSK* resultante. Así, con desplazamientos de fase de 90 grados, la portadora es suprimida. No hay diferencia en la onda transmitida entre *PSK* 90 grados y *ASK* con portadora suprimida.

2.1.3 QPSK. DESPLAZAMIENTO DE FASE EN CUADRATURA. El desplazamiento de fase en cuadratura (*QPSK*) es una extensión del método de *PSK* simple. En *QPSK* la señal puede tomar uno de los cuatro ángulos de fase posibles, mutuamente en cuadratura, cada uno corresponden a una condición de entrada de datos particular. Sea el formato de datos *NRZ* en el cual cada palabra es dividida en pares de *bit* en lugar de *bits* individuales. Hay cuatro modos posibles de paridad binaria 1 y 0. Estos son: 00 01 10 11. Cualquier palabra de información con un número par de *bits* puede ser representado por una combinación de estos

pares de *bits*. Uno de los cuatro ángulos de fase es asignado a cada uno de estos pares de *bits*. *QPSK* comparado con el *BPSK* ofrece el doble de *bits* de información por portadora de cambio de fase, encuentra amplia aplicación en sistemas de transmisión de datos de portadora modulada de alta velocidad. Por ejemplo, si la tasa de transferencia de datos tiene 9600 *bits* por segundo la tasa de señalización de línea de transmisión tendrá 4800 pares de *bits* por segundo y así estará a 4800 baudios. Esto significa que el ancho de banda requerido por cualquier tasa de transferencia de datos dada será aproximadamente la mitad para *QPSK* comparada con *BPSK*. Típicamente, las cuatro fases elegidas para *QPSK* son ± 45 grados y ± 135 grados. Cada uno de estos es asignado a un par de *bits* (*dibit*). El diagrama muestra un posible patrón *dibit*, frecuentemente referido como una constelación debido a su forma de estrella.

Figura 2. Diagrama de constelación *QPSK*



La expresión para este *QPSK* será:

$$s(t) = \cos(\mathbf{w}_c t + q)$$

Donde

$$q = +\frac{A}{4}, -\frac{A}{4}, +\frac{3A}{4}, -\frac{3A}{4}$$

Por expansión trigonométrica, esto puede estar escrito:

$$s(t) = A \cdot \cos \mathbf{w}_c t + B \cdot \text{sen } \mathbf{w}_c t$$

Donde *A* y *B* son los valores dados correspondientes a los cuatro posibles ángulos.

La señal transmitida es por lo tanto la suma de dos ondas en cuadratura.

Demodulación de *QPSK*

La señal *QPSK* recibida deberá ser demodulada para producir los dos componentes de la señal transmitida. Esto es para combatir los problemas de ambigüedad de fase. Un detector de fase, o detector de producto, es un mezclador, cuya acción es la de multiplicación. En el detector, las dos señales a ser multiplicadas tendrán la misma frecuencia, así la salida del detector contendrá

componentes de frecuencia de suma y diferencia. Para lograr esto, se tiene que añadir el proceso de recuperación de portadora; la portadora recuperada, coherente en fase y frecuencia con la portadora de referencia, es usada entonces en los detectores de producto para obtener las componentes usadas en la demodulación. El componente de suma estará al doble de la frecuencia de portadora y el componente de diferencia estará a *DC*. Por ejemplo si las entradas al detector son $\cos \omega_c t + q$ y $\cos \omega_c t$ la salida será:

$$\cos(\omega_c t + q) \times \cos(\omega_c t) = 0.5 \cos(\omega_c t + q + \omega_c t) + 0.5 \cos(\omega_c t + q - \omega_c t)$$

$$\cos(\omega_c t + q) \times \cos(\omega_c t) = 0.5 \cos(2\omega_c t + q) + 0.5 \cos(q)$$

$$\cos(\omega_c t + q) \times \cos(\omega_c t) = 0.5 [\cos(2\omega_c t + q) + \cos(q)]$$

$$\cos(\omega_c t + q) \times \cos(\omega_c t) = 0.5 [\cos(2\omega_c t) \cdot \cos(q) - \text{sen}(2\omega_c t) \cdot \text{sen}(q) + \cos(q)]$$

Donde q es la fase de información transmitida. Un filtro pasa bajos es usado para atenuar el segundo armónico ($2\omega_c t$) haciendo cero o nulos los componentes de dicho armónico dejando como resultado:

$$V(o) = 0.5 \cdot \cos(q)$$

Donde q es el fase de desplazamiento modulada ($\pm A/4$, $\pm 3A/4$) y $V(o)$ es la salida de voltaje *DC* representando el apropiado *dibit*. Similarmente, $0.5 \cos(\pm 45 \text{ grados})$

= +0.35, y por lo tanto el detector no sabe si fue enviado +45 grados, o -45 grados. Igualmente, $0.5 \cos(\pm 135 \text{ grados}) = 0.35$, la misma ambigüedad existe.

Para resolver estas ambigüedades, un segundo detector operativo en cuadratura es requerido. Este puede ser logrado del circuito de doble Lazo de Costas.

Las entradas al segundo detector son $\cos \mathbf{w}_C t + q$ y $\sin \mathbf{w}_C t$ y la salida ser:

$$\cos(\mathbf{w}_C t + q) \times \sin(\mathbf{w}_C t) = 0.5 \sin(\mathbf{w}_C t + q + \mathbf{w}_C t) - 0.5 \sin(\mathbf{w}_C t + q - \mathbf{w}_C t)$$

$$\cos(\mathbf{w}_C t + q) \times \sin(\mathbf{w}_C t) = 0.5 \sin(2\mathbf{w}_C t + q) - 0.5 \sin(q)$$

$$\cos(\mathbf{w}_C t + q) \times \sin(\mathbf{w}_C t) = 0.5 \sin(2\mathbf{w}_C t) \cdot \cos(q) + 0.5 \sin(q) \cdot \cos(2\mathbf{w}_C t) - 0.5 \sin(q)$$

Donde q es la fase de información transmitida. De nuevo, un filtro pasa bajos es usado para atenuar el segundo armónico ($2\mathbf{w}_C t$), dejando:

$$V(o) = -0.5 \cdot \sin(q)$$

Donde q es el fase de desplazamiento modulada ($\pm A/4$, $\pm 3A/4$) y $V(o)$ es la salida de voltaje *DC* representando el apropiado *dibit*. Las salidas de los dos moduladores son multiplicadas juntas para producir la señal de control *VCO*, dando

$$0.5 [\cos(2\mathbf{w}_C t + q) + \cos(q)] \times 0.5 [\sin(2\mathbf{w}_C t + q) - \sin(q)] =$$

$$0.25 \left[\begin{aligned} &\cos(2\mathbf{w}_c t + q) \times \text{sen}(2\mathbf{w}_c t + q) - \cos(2\mathbf{w}_c t + q) \times \text{sen}(q) \\ &+ \cos(q) \text{sen}(2\mathbf{w}_c t + q) - \text{sen}(q) \times \cos(q) \end{aligned} \right] =$$

Expandiendo estos términos, usando la identidad:

$$\cos \mathbf{a} \times \text{sen} \mathbf{b} = 0.5 [\text{sen}(\mathbf{a} + \mathbf{b}) - \text{sen}(\mathbf{a} - \mathbf{b})]$$

Se obtiene:

$$= \frac{1}{8} \text{sen} 2(2\mathbf{w}_c t + q) - \frac{1}{8} [\text{sen}(2\mathbf{w}_c t + 2q) - \text{sen}(2\mathbf{w}_c t)] + \frac{1}{8} [\text{sen}(2\mathbf{w}_c t + 2q) + \text{sen}(2\mathbf{w}_c t)] - \frac{1}{8} \text{sen} 2q$$

Que se simplifica a:

$$= \frac{1}{8} [\text{sen} 2(2\mathbf{w}_c t + q) + \text{sen}(2q)]$$

$$= \frac{1}{8} [\text{sen}(4\mathbf{w}_c t + 2q) + \text{sen}(2q)]$$

Este contiene un termino $4\mathbf{w}_c t$, el cual será filtrado, resultando un termino *DC* proporcional al desplazamiento de fase el cual es usado para el control VCO.

Las señales *QPSK* recibidas son demoduladas en un circuito doble de *Lazo de Costas* para producir las dos señales de datos correspondientes a aquellas originalmente generadas en el transmisor. En operación normal la frecuencia del VCO es sincronizada con la de la señal entrante y produce dos salidas de onda cuadradas, mutuamente en cuadratura, a la frecuencia portadora. Cuando una

señal de onda de seno es multiplicada con una onda cuadrada de la misma frecuencia la salida contendrá una componente *DC* proporcional a la diferencia de fase entre las dos formas de onda. Por lo tanto, los multiplicadores 1 y 2 actúan como detectores de fase los cuales extraen las componentes de fase y cuadratura de la señal y producen dos formas de onda de datos correspondientes a las dos señales originalmente generadas en el transmisor.

Las salidas de los multiplicadores 1 y 2 son proporcionales a $\cos(i)$ y $\sin(i)$. Conocemos que el ángulo de fase de la señal entrante puede ser ± 45 grados, o ± 135 grados respecto a la referencia de transportador. En el receptor el VCO provee la señal de referencia y las salidas de los multiplicadores 1 y 2 tendrán las polaridades mostradas en la tabla 1:

Tabla 1. Resultados para cada fase en un sistema *QPSK*

Fase de la señal	Mult. 1 salida	Mult. 1 Polaridad	Mult. 2 salida	Mult. 2 polaridad
+45	$K\cos 45$	+	$K\sin 45$	+
+135	$K\cos 135$	-	$K\sin 135$	+
-135	$K\cos(-135)$	-	$K\sin(-135)$	-
-45	$K\cos(-45)$	+	$K\sin(-45)$	-

Se puede observar que las dos salidas producen diferentes combinaciones de polaridad por cada ángulo de fase. Estas combinaciones de polaridad corresponde al estado original *dibit* del transmisor. Ejemplo: +45 a 11, +135 a 01, -135 a 00 y -45 a 10.

Una de las salidas dará una forma de onda correspondiente a los patrones de bits *MS*, el otro dará los patrones de *bits LS**. El bucle tiene que permanecer en bloqueo mientras la señal de entrada cambia entre estas cuatro fases. Esto significa que el sistema tiene que poder bloquear a estas cuatro fases sin cambiar la frecuencia o fase del *VCO*.

Para lograr esto el conjunto de circuitos asociado con los multiplicadores 3 y 4 es requerido. Se ordena que una entrada a cada uno de estos multiplicadores sea hecha tan grande que cada multiplicador cambie efectivamente la polaridad de su otra entrada. La salida combinada de los multiplicadores 3 y 4 se aplica a la terminal de control del *VCO*. Recordando que el *VCO* es sincronizado a la frecuencia de la señal entrante y que un voltaje particular es requerido para mantener este sincronismo, puede verse que hay cuatro valores de fase para completar en los que el sistema puede bloquearse. Este da al circuito la habilidad para mantener una fase de referencia constante con los cambios de entrada entre sus cuatro posibles valores de fase. En *PSK* binario había doble ambigüedad

* MSB: bit mas significativo, como se detalla en el glosario

al definir la fase de referencia. Con *QPSK* hay una cuádruple ambigüedad. Para resolver esta ambigüedad una palabra de reconocimiento es transmitida periódicamente. Esta palabra tiene una combinación distintiva de 1's y 0's y puede ser usada para reconocer si cualquier *bit* ha sido invertido como resultado de bloqueo hacia una fase ambigua. Los *bits* en error pueden entonces ser corregidos.

El circuito integrado de aplicación específica o *ASIC*, a utilizar en este proyecto, esta diseñado pensando en las necesidades de reales del mercado, no fue originalmente fabricado con fines didácticos, pero por lo completo de su constitución, maneja muchos conceptos contemporáneos empleados en las comunicaciones digitales como la necesidad de enmascarar los datos, cubierta con *Scrambler* y Seudo Ruido; la de brindar una buena relación señal ruido y corregir errores en la transmisión lograda con *Reed Solomon*, filtros de interpolación, ajustes de ganancia y bs recursos con los que brinda fidelidad en la información procesada, que se obtiene trabajando a nivel digital y luego convirtiendo de digital a análogo con *DAC's* integrados. Seguidamente, además de hablar de un significado a usar ampliamente en el resto del documento, se tocan los conceptos antes nombrados para dar una idea de los modelos matemáticos de origen y el porque de su implementación

2.7 EXPERIENCIAS A REALIZAR

Las experiencias a realizar se dividen de acuerdo al tipo de proceso a estudiar en secciones sencillas para agilizar el proceso de aprendizaje en cada experiencia. Cada experiencia consiste de 4 etapas divididas en dos grupos. Un grupo controla la práctica desde el panel del entrenador con una configuración base y el otro conectado e un ordenador.

EXPERIENCIA 1: MODULACION Y TRANSMISION BPSK UTILIZANDO CABLE COAXIAL Se utiliza el entrenador para transmitir las señales en *BPSK* a través de cable coaxial. En la fase A una señal de prueba generada internamente en el *ASIC* por *Pseudo Ruido* es sometida a análisis, esto a través de los botones de usuario empotrados en el entrenador para seleccionar las codificaciones y otros parámetros como el *Scrambler*. En la fase B el alumno controla la configuración a través de software por el puerto serie de un ordenador del laboratorio. Para *BPSK* se cuenta opcionalmente con un modulo demodulador *BPSK* sencillo donde se puede recuperar la señal original

EXPERIENCIA 2: MODULACION Y TRANSMISION QPSK UTILIZANDO CABLE COAXIAL Para esta experiencia el alumno trabajará de forma similar a la experiencia 1, las diferencias residen en que la modulación a tratar será *QPSK*. Al

igual que en la primera experiencia, se trabaja una parte desde el panel del equipo y otra conectado a un computador del laboratorio. En todas estas experiencias, se espera que el alumno compruebe los efectos de las distintas modulaciones sobre la señal original

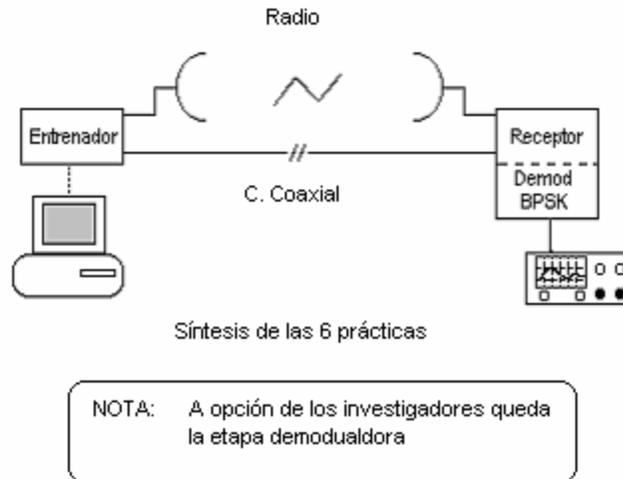
EXPERIENCIA 3: MODULACION Y TRANSMISION 16QAM UTILIZANDO CABLE COAXIAL Se trabaja como en las anteriores pero para transmisión *QAM*

EXPERIENCIA 4: MODULACION Y TRANSMISION BPSK UTILIZANDO MEDIO AEREO. La experiencia 4 es básicamente la experiencia 1 (tratamiento de *BPSK*) con la diferencia de que el medio a usar es aéreo

EXPERIENCIA 5: MODULACION Y TRANSMISION QPSK UTILIZANDO MEDIO AEREO. La transmisión a tratar será *QPSK* por medio aéreo

EXPERIENCIA 6: MODULACION Y TRANSMISION 16QAM UTILIZANDO MEDIO AEREO. Modulación *16QAM* por medio aéreo. Por la mayor complejidad de la salida en este tipo de modulación, el alumno tiene que someterse a interrogantes mas profundas. Uno de los aspectos complejos a tratar son los diagramas de constelación y fasoriales para esta modulación

Figura 3. Practicas a realizar con el entrenador



Al tener *BPSK*, *QPSK* y *16QAM* será posible que los estudiantes confirmen las ventajas de cada una sobre la anterior, como el ancho de banda de *bits*, la eficiencia y los baudios utilizados. Usualmente un módem se usa para transmitir información digital a través de una línea telefónica que usa par de cobre (módem: siglas de *modulador–demodulador*), y así conectarse a la Internet, una red corporativa o a cualquiera que sea accesible por línea telefónica o medio aéreo. Para transmitir esta información por canales tradicionalmente de voz, es necesario utilizar este proceso: Convertir una señal digital en una señal análoga para su transmisión por una canal típicamente de voz, con un ancho de banda de *Nyquist* de 8 KHz. Las modulaciones *PSK* y la *QAM* se subdividen en varios tipos a

distintas escalas de eficiencia como por ejemplo: *BPSK*, *QPSK*, *8PSK*; *QAM*, *16QAM*. Implementar un entrenador en el laboratorio donde el alumno pueda examinar cada una de las partes del proceso implícito y observar las distintas ventajas y desventajas de cada una tiene que dar como resultado un enriquecimiento en la formación del estudiante del área de Comunicaciones II. El estudiante no tiene que jugar con las piezas del entrenador, el fin didáctico de este es conocer el proceso y experimentar otros como la aleatorización de la información no tocar o desarmar el equipo: sin embargo un esquema de este se puede anexar a los laboratorios para darles una idea de lo que manipulan. Un equipo que desglosara el proyecto implicaría módulos desarmables donde cada alumno colocara sondas (*probes*) en las uniones. Esto se traduciría en diseño totalmente distinto. Como conclusión cualquier cambio de objetivos del proyecto se traduce en un cambio total del diseño. Como una observación importante se aclara que para el radio enlace usado en las prácticas a través de *RF* se uso otro tipo de modulación, se usaron módulos que emplean *FM SAW* ofreciendo varias ventajas sobre la idea original:

1. Módulos totalmente certificados de acuerdo a normas internacionales
2. Señales portadoras “observables” por el alumno usuario al no tener que trabajar con portadoras de muy alta frecuencia (*RF*)
3. Posibilidad de realizar las practicas con equipos ya existentes en el laboratorio y que no representan costo adicional

3. RESULTADOS DE LA INVESTIGACION

- Con la puesta en marcha de este proyecto se alcanzara afianzar más a los estudiantes en los conocimientos sobre las técnicas de modulación digital *PSK* y *QAM*.
- Los estudiantes del laboratorio de comunicaciones electrónicas podrán conocer el comportamiento de las señales tratadas en estas prácticas sobre los medios de transmisión utilizados para el desarrollo de la misma.
- Con la incorporación de herramientas informáticas el estudiante tendrá un elemento mas de apoyo para alcanzar los objetivos propuestos en el proceso enseñanza aprendizaje
- Adicionalmente y como valor agregado el entrenador ofrece la oportunidad de que el alumno conozca técnicas modernas de codificación y aleatorización como la codificación *Reed-Solomon* y el *Scrambler* respectivamente, las cuales se usan comercialmente para garantizar la integridad y eficiencia de los datos transmitidos de forma serial
- Se resalta la importancia de lo amistoso de la interfase de usuario y de lo manejable de la practica en el proceso de aprendizaje en detalles como la observabilidad de las señales digitales en la fijación de los parámetros de diseño

4. ELEMENTOS A UTILIZAR

MODULACIÓN Y TRANSMISIÓN DE SEÑALES BPSK, QPSK y 16QAM UTILIZANDO CABLE COAXIAL

- Entrenador
- Sección de cable coaxial
- Osciloscopio
- Guía de laboratorio

Objetivos de estas experiencias:

- Acercar al estudiante a una modulación digital real valiéndose del prototipo como herramienta profundizando con este, de una manera sencilla, en varios medios de transmisión para enriquecer aun mas la experiencia, utilizando además herramientas clásicas del laboratorio como el osciloscopio, cable coaxial y los *PC's*

MODULACIÓN Y TRANSMISIÓN DE SEÑALES BPSK, QPSK Y 16QAM POR RF

- Entrenador configurado para transmisión de radio
- Módulos transmisor y receptor (incluido en el proyecto)
- Osciloscopio (no incluido)
- Guía de laboratorio

Adicionalmente se propone un pequeño demodulador, el cual es opcional, en el proyecto

RECEPCIÓN DE SEÑALES BPSK, QPSK Y 16QAM A TRAVÉS DE CABLE COAXIAL

- Entrenador
- Sección de cable coaxial (incluido)
- Guía de laboratorio
- Demodulador del lado receptor del entrenador (opcional)

RECEPCIÓN DE SEÑALES BPSK, QPSK Y 16QAM POR RF

- Entrenador
- Módulos transmisor y receptor (incluido)
- Guía de laboratorio
- Demodulador del lado receptor del entrenador (opcional)

Originalmente se había hablado de varias propuestas; software para producir digitalmente la señal a demodulador y una etapa anexa de demodulación. La reorganización y optimización, además de una exhaustiva investigación en la etapa

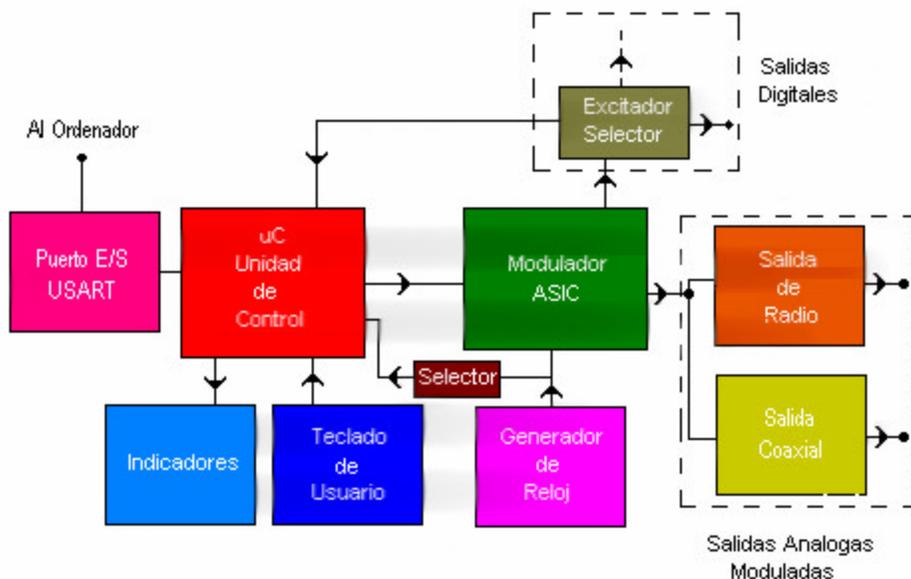
de diseño llega a la conclusión de que si existe ya un modulador no hay necesidad de implementar otro para las practicas y de que los costos de un equipo demodulador para todos los tipos de modulación propuestos, son extremadamente elevados. Entonces la transmisión y recepción se hace en una sola dirección, lo que bien justifica los costos y hace menos complicado el uso del entrenador por parte de sus usuarios finales.

El entrenador es básicamente un modulador, la demodulación se brinda como un valor agregado al proyecto para que el alumno tenga a oportunidad de experimentar el proceso en las dos direcciones; transmisión y recepción

5. DESARROLLO

5.1 ESTRUCTURA EN BLOQUES FUNCIONALES DEL ENTRENADOR

Figura 4. Diagrama de bloques del entrenador



En el dibujo se pueden identificar las diversas partes que se numeraran a continuación:

5.1.1 FUENTE DE ALIMENTACION. Aunque no aparezca en el diagrama, es la encargada de energizar con los potenciales adecuados las diversas partes del proyecto. Incluye las respectivas protecciones y reguladores. Esta compuesta por

tecnologías activas y pasivas: circuitos integrados en encapsulado *TO-220*, fusible, condensadores, aislamiento galvánico (transformador), resistores y transistores.

El diseño entero de la fuente es clásico; un transformador reductor, que proporciona aislamiento galvánico entre la red eléctrica y el circuito, de fabricación nacional con referencia *M-509 de MAGOM®*, que es capaz de proporcionar 6, 9, 12, 15 y 18 voltios. Le sigue un puente rectificador encapsulado de 1.5 Amperios con un *PRV* de unos 600 Voltios, el *W06M*, a la salida el respectivo condensador electrolítico para disminuir el voltaje de rizado a la entrada de los reguladores y así tener una atenuación en el voltaje de rizado aceptable y más que suficiente, este tiene un valor de unos 2200 μF ; con este valor se evita que la carga inicial del condensador dispare el fusible pues su constante de tiempo de carga no es lo suficientemente grande y recordando que todo condensador en condiciones iniciales se presenta como un cortocircuito. Un fusible situado antes del puente rectificador es la protección contra sobre corriente. Este es de acción rápida con una corriente nominal de 1 A. El máximo consumo de corriente teórico de la fuente viene determinado por la suma de consumos de corriente máximos individuales especificados para cada componente y se lista en la tabla siguiente

Tabla 2. Totalización de los consumos de corriente máximos

Dispositivo	Imáx	Cantidad	Consumo Total
<i>Microcontrolador</i>	<i>30 mA</i>	<i>1</i>	<i>30.00 mA</i>
<i>Modulador: S1109CR</i>	<i>30 mA</i>	<i>1</i>	<i>30.00 mA</i>
<i>TTL's Varios</i>	<i>30 mA</i>	<i>3</i>	<i>90.00 mA</i>
<i>LED's</i>	<i>25 mA</i>	<i>2</i>	<i>50.00 mA</i>
<i>Módulo de Radio</i>	<i>17 mA</i>	<i>1</i>	<i>17.00 mA</i>
<i>Generador del reloj</i>	<i>25 mA</i>	<i>1</i>	<i>25.00 mA</i>
<i>Panel visualizador LCD</i>	<i>30 mA</i>	<i>1</i>	<i>36.00 mA</i>
Totales			<i>278 mA</i>

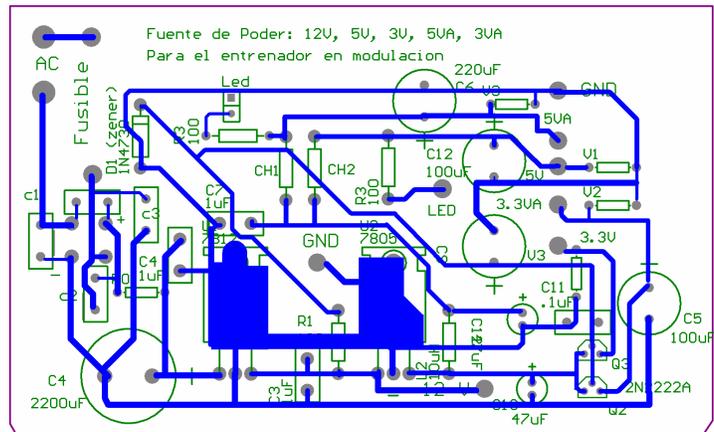
Los valores marcados en negrilla representan consumos de corriente específicamente en la carga del 7812 y no en la del 7805, o sea, que estas corrientes no entran en los cálculos de disipación de potencia de este último. La fuente tiene que cubrir las necesidades de todo el prototipo, por esto, de acuerdo a las necesidades de los distintos módulos, los voltajes necesarios a la salida de esta son:

Tabla 3. Magnitudes de los valores de voltaje especificados

V_{DD}	<i>5 voltios para el s1109cr, los TTL's y LED's</i>
V_{SS}	<i>3.3 voltios parte digital del s1109cr</i>
AV_{DD}	<i>5 voltios parte análoga del s1109cr</i>
AV_{SS}	<i>3,3 voltios parte análoga del s1109cr</i>
Voltaje fuente módulo de radio	<i>5 voltios</i>

Ya que la fuente proporciona alimentación a circuitos digitales en determinado caso muy sensibles a los transitorios como los basados en *MOS (CMOS-MOSFET)*, cada salida entre 3,3 voltios y 5 voltios tiene un pequeño *varistor* de bajo voltaje especialmente diseñado para estos propósitos. El diseño de la *PCB* para la fuente de alimentación se muestra a continuación en la figura 5 como se ve en el software *EDA Protel*

Figura 5. *PCB* de la fuente de alimentación bajo *Protel Design Explorer 99* ®



En la figura 6 se puede apreciar una vista en 3a dimensión generada virtualmente por el mismo *Protel*; y en la figura 7, ya se tiene una imagen de la vista superior de la misma *PCB* dentro de la caja

Figura 6. Vista posterior 3D de la *PCB* de la fuente de alimentación

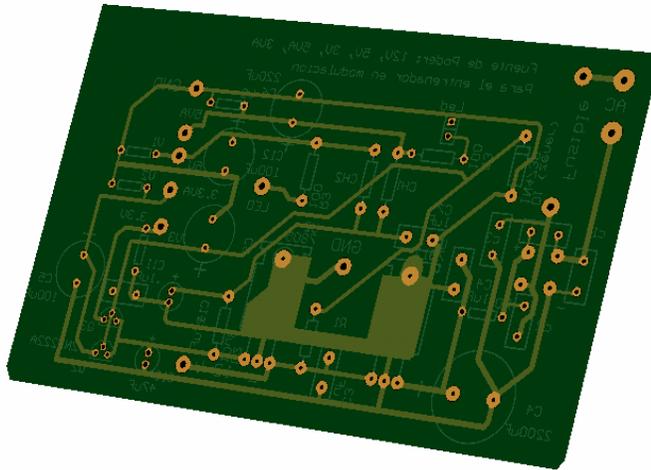


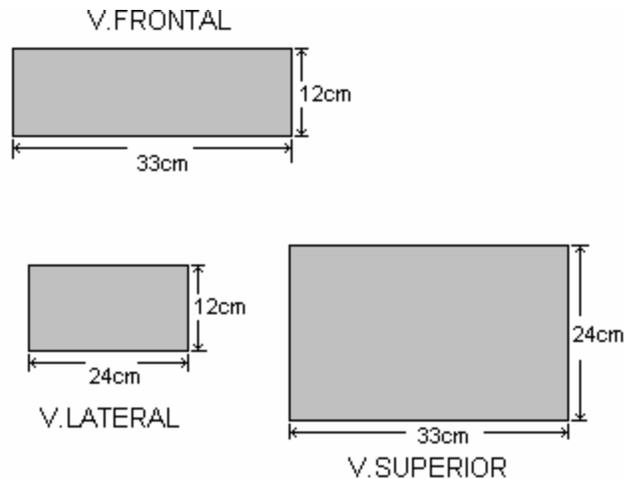
Figura 7. Vista superior de la *PCB* de la fuente de alimentación armada



Como lo muestra el trazado de la figura 5, solo hubo que trazar la capa inferior de la tarjeta, pues la densidad de componentes e interconexiones, lo que algunos expertos llaman la población de componentes sobre la *PCB*, no es mucha.

CARACTERÍSTICAS MECANICAS. Las dimensiones de la caja C-90 se muestran en la siguiente figura

Figura 8. Dimensiones de la caja del entrenador



PARÁMETROS DE DISEÑO. El regulador integrado de 12 voltios, el *LM7812*, que es un regulador de voltaje positivo estándar como todos los reguladores de la serie *LM78XX*, se encarga de proporcionar un voltaje estable con un 5% de tolerancia, con muy bajo porcentaje de rizado y limitación interna de corriente. Este voltaje se dedica al modulo de transmisión de radio que trabaja con voltajes de este rango, entre 5 voltios y 12 voltios. El voltaje obtenido del *LM7812* se suministra al regulador de 5 voltios *LM7805* en serie con este, el cual multiplica su eficiencia a la ya proporcionada por el *LM7812*. El *datasheet* de este y otros componentes pueden hallarse en los anexos del proyecto. Debido a la alta estabilidad del voltaje

a la salida del *LM7805*, para alimentar a la parte *CMOS* del circuito integrado *ASIC*, la cual consume 3.3 Voltios con un 10% de tolerancia, se dedico un regulador de voltaje serie a transistor con un diodo zener de unos 4 voltios, que a la salida de cada emisor de transistor y por la caída de voltaje en la unión base – emisor, popularmente conocida como potencial de barrera del diodo *, debe proporcionar aproximadamente los 3.3 voltios requeridos por *ASIC*. Como el fabricante del *ASIC* especifica un 10% de tolerancia en todos sus voltajes, entonces el voltaje de fuente en este caso debe estar entre 2,97 y 3,67 voltios. El cálculo de la malla se describe a continuación, pero primero se procede a calcular la carga alimentada por esta fuente. El integrado de aplicación especifica tiene un máximo consumo de corriente de 30 mA, este se tomará como máximo consumo de corriente lo que es un límite bastante extremo, pues el consumo de corriente de 30 mA esta especificado para todo el integrado, o sea, la entrada de 3.3 voltios (V_{DD}) y la entrada de 5 voltios (5Vdd), este caso en especial se refiere al V_{DD} del integrado. El cálculo de la carga en el peor de los casos (R_L mínimo) arroja

$$R_L = \frac{V_{dd}}{I_{m\acute{a}x}} = \frac{3.3 V}{30mA} = 110\Omega$$

Ya que se usan transistores *2N2222A* los que especifican una ganancia de corriente o ***b*** mínima de 150 para está corriente de colector (lo mismo se puede

* Malvino: "Principios de electrónica" Capitulo 1, teoría del diodo

deducir de observar las curvas características del transistor en su hoja de datos) y una máxima de 400. Con la β mínima y la R_L mínima se pueden calcular los parámetros necesarios en la base del transistor y así, la corriente que manejará el *zener* que permite conocer el valor de la potencia máxima que deberá disipar y el mínimo valor de la resistencia en serie con el *zener*. Para simplificar los cálculos se puede sustituir la base del transistor por su equivalente *Thévenin* y calcular con mayor facilidad descriptiva los valores en el regulador *zener*. Antes se puede confirmar la disipación de potencia el transistor para que no quede en tela de juicio el motivo de su elección para esta aplicación. La primera razón; es un componente muy comercial y accesible; la segunda es que por sus características eléctricas se amolda a las exigencias

POTENCIA DISIPADA POR CADA TRANSISTOR ASOCIADO AL S1109CR. Es el producto del voltaje en la unión colector–emisor por la corriente a través del transistor. De la tabla 2 se toma el máximo consumo de corriente para el ASIC. El V_{CE} se toma del voltaje aplicado a cada transistor en su colector, menos el que debe salir por cada emisor, para cada colector común.

$$V_{CE} = \text{Voltaje de colector} - \text{Voltaje de emisor} = 5 \text{ voltios} - 3.3 \text{ voltios} = 1.7 \text{ voltios}$$

$$I_c = \text{intensidad máxima consumida por la carga} = 30 \text{ mA}$$

Entonces:

$$P_D = V_{ce} * I_c = 1.7V * 30mA = 51mW$$

Que es menos del 10% de la máxima potencia que puede disipar el componente

CÁLCULO DEL EQUIVALENTE THÉVENIN VISTO POR EL REGULADOR ZENER. La carga vista por cada regulador zener puede ser llevada a su equivalente Thévenin para simplificar en forma gráfica el análisis. Para este efecto se toman los parámetros mas aproximados, como es el caso de la ganancia de corriente del transistor ($\beta=150$) en función del rango de corriente que va a manejar

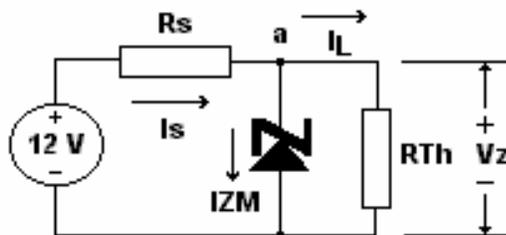
I_b = Corriente máxima en la base del transistor = $30mA/150 = 200 \mu A$

V_z = Voltaje proporcionado por el zener = 3,9 voltios

Entonces

$$R_{TH} = \frac{V_z}{I_b} = \frac{3.9V}{200 \mu A} = 19.5K\Omega$$

Figura 9. Circuito de la fuente de alimentación con carga *Thévenin* equivalente



El valor mínimo de la resistencia en serie viene determinado por la máxima corriente que el diodo puede soportar especificada como I_{ZM} . Esta se puede calcular de la potencia del diodo que por cuestiones de tamaño tendrá una potencia zener máxima (P_{ZM}) de 0,5 Vatios

$$I_{ZM} = \frac{P_{ZM}}{V_Z} = \frac{0.5 \text{ Vatios}}{3.9 \text{ Voltios}} = 128 \text{ mA}$$

Empleando la *Ley de Kirchhoff* de Corriente y haciendo un nodo en **a** se obtiene la expresión

$$I_s = I_{ZM} + I_L$$

$$I_s = 128 \text{ mA} + 200 \text{ mA}$$

Teniendo en cuenta que $I_L \ll I_{ZM}$ y es entonces despreciable

$$I_s \cong 128 \text{ mA}$$

La resistencia en serie mínima será entonces

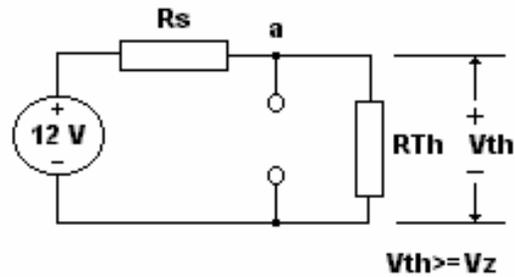
$$R_{s_{MINIMA}} = \frac{V_{ent} - V_z}{I_{L_{MAX}}}$$

$$R_{s_{MINIMA}} = \frac{12 \text{ Volts} - 3.9 \text{ Volts}}{128 \text{ mA}}$$

$$R_{s_{MINIMA}} \cong 64 \Omega$$

Por otro lado la R_s máxima será el valor para el cual el voltaje en el diodo zener todavía está en estado encendido. Esto se da cuando el voltaje aplicado al circuito *Thévenin* equivalente, sin el diodo zener, es mayor o igual al voltaje zener

Figura 10. Circuito equivalente sin diodo Zener



Tomando los nombres de las variables tratadas y despejando de la formula del divisor de voltaje se obtiene

$$R_s = \frac{R_{Th}(12V - V_z)}{V_z}$$

$$R_s = \frac{19.5K\Omega(12V - 3.9V)}{3.9V}$$

$$R_s = 40K\Omega$$

Se le asigna un valor comercial a la R_s que acepte las tolerancias de 5% que poseen tanto los reguladores de voltaje como las resistencias estándar y las posibles variaciones en la β del transistor. Entonces

$$R_s = 1K\Omega$$

Los cálculos anteriores se refieren específicamente al regulador zener serie que alimenta al s1109cr. Para la fuente de alimentación en general se puede predecir la disipación de potencia de acuerdo a los parámetros de máximo consumo de corriente de cada componente especificados en la tabla 2. Redondeando los valores por exceso, se toma una corriente total de 709 mA. Se puede calcular

entonces la disipación de potencia para cada regulador electrónico. El circuito rectificador incluido su filtro proporciona un voltaje *DC* de salida máximo de 18 Voltios

REGULADOR DE 12 VOLTIOS LM7812. El voltaje a través del regulador serie integrado *LM7812* viene dado por la diferencia del voltaje de entrada menos el voltaje de salida

$$V_{serie_1} = V_{ent} - V_{sal}$$

$$V_{serie_1} = 18V - 12V$$

$$V_{serie_1} = 6V$$

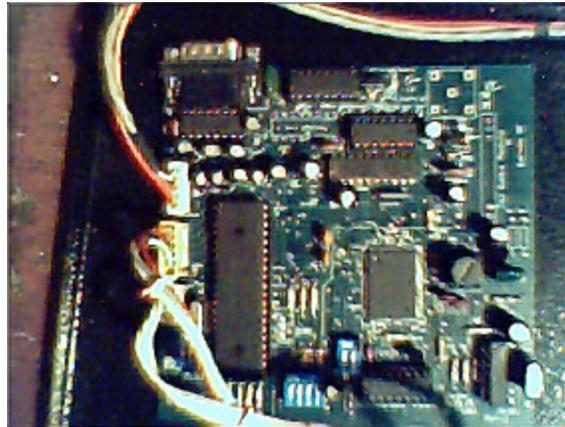
Entonces se puede calcular la potencia de disipación

$$P_{D_{LM7812}} = 6V * 278mA \cong 1.67W$$

REGULADOR DE 5 VOLTIOS LM7805. El voltaje a través del regulador serie integrado *LM7805* viene dado por la diferencia del voltaje de entrada, los 12 Voltios proporcionados por el *LM7812*, menos el voltaje de salida. La corriente a través de este es la corriente total menos la corriente de los *LED*'s que es manejada solo por el *7812* y otro *7805* dedicado a la sección de visualizadores

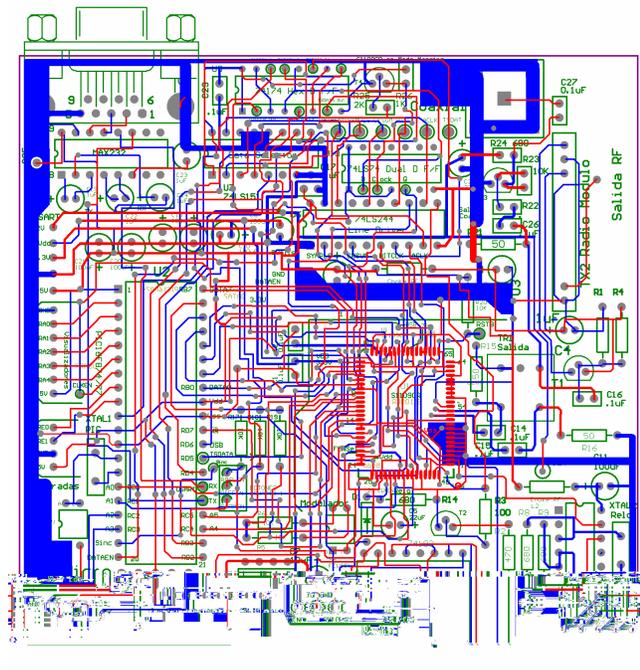
En la siguiente figura se muestra una vista superior de la *PCB* del entrenador armado y dispuesto en la caja

Figura 12. Vista superior de la *PCB* del entrenador



La *PCB* de la sección moduladora en estado definitivo se ve como en la figura 12

Figura 13. Imagen de la *PCB* del entrenador

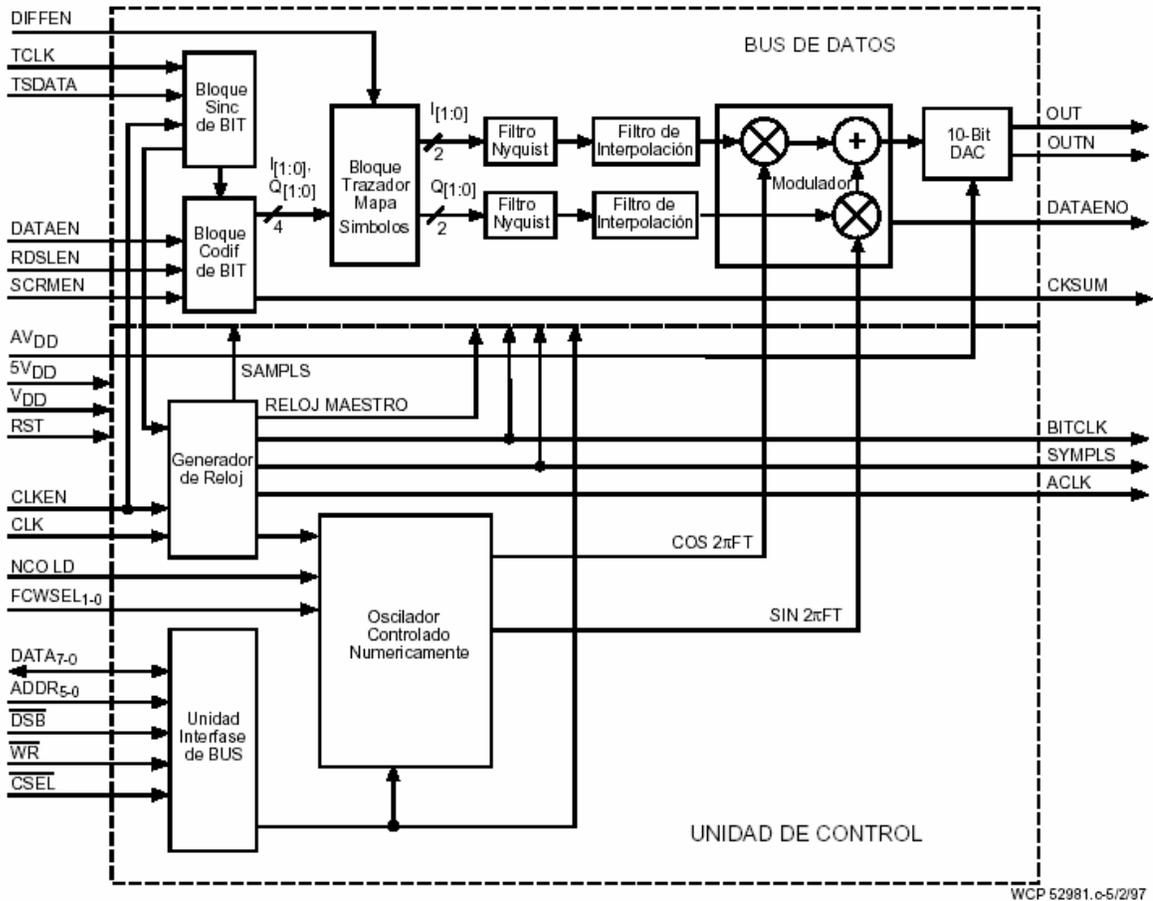


5.1.2. SECCIÓN DE MODULACIÓN. Con tecnologías integradas, en esta sección se realiza la modulación de información digital generada desde el software en el PC o internamente en el entrenador para su posterior transmisión por los diferentes medios

EL MODULADOR DIGITAL ASIC S1109CR. El *STEL-1109*[†] esta conformado por las secciones *Bus de Datos* y *Unidad de Control* descritas en la figura. EL *Bus de Datos* está conformado por un *Bloque Sincronizador de Bit*, un *Bloque Codificador de Bit* (Aleatorizador, Codificador *Reed Solomon* y dos multiplexadores...), un *Bloque Trazador de Mapa de Símbolos* (trazador de mapa de *bits*, codificador diferencial y trazador de mapa de símbolos...), dos canales (uno I y uno Q), un combinador y un DAC de 10 *bits*. Cada canal consiste de un filtro *Nyquist*, un filtro de interpolación y un modulador. La Unidad de Control esta conformada por una Unidad Interfase de Bus (*BIU*), un generador de reloj y un *NCO* (Oscilador Controlado Numéricamente). El *ASIC S1109CR* es un integrado diseñado por la *Stanford Telecom*, actualmente parte de *INTEL*, y usado en aplicaciones de modulación digital, como cable *modem*. Un "Cable Módem" es un dispositivo que permite tener acceso a datos a muy alta velocidad vía una red de *TV cable (CATV)*. Un *Cable Módem* típicamente tiene dos conexiones, una al cable coaxial de la empresa prestadora del servicio de *TV cable* y la otra a la computadora (PC).

[†] Se notará que se usan alternativamente las notaciones *STEL-1109* y *S1109CR*. Ambas denotan el mismo circuito integrado: mientras la primera es de *STANFORD TELECOM*, su fabricante original, la segunda es de *INTEL*, quien adquirió los derechos

Figura 14. Diagrama de bloques del S1109CR



Intel Corporation: Cable network operation. *STEL-1109 Data Sheet* [Ebook]. San José: California. Diciembre 15, 1999. p. 9

El S1109CR ofrece varias posibilidades interesantes, valores agregados para el estudiante y el usuario en general. Este se presenta en un encapsulado MQFP de 80 terminales, el cual va montado superficialmente sobre la PCB doble faz principal. Como todo SMD, merece una técnica de soldadura especial a la hora de montarse sobre la PCB.

UNIDAD DE CONTROL. El *S1109CR* se maneja mediante la configuración de 58 registros programables de 8 *bits* direccionados mediante los pines 8, 9, 10, 12, 13 y 14 que representan los 5 *bits* de direccionamiento de los registros, recordando que el numero de combinaciones posibles con valores binarios (*bits*) tiene que ser 58 por el numero de registros y que el numero mas cercano se logra con 6 *bits*, que dan la posibilidad de 64 combinaciones posibles

$$2^n \geq 58 \rightarrow n = 6$$

Entonces estos *bits* denominados *ADDR0* hasta *ADDR5* permiten seleccionar el registro a leer o modificar. La información concerniente a los registros ya sea para lectura o escritura se almacena a través de un buffer de datos de 8 *bits* (*DATA0* a *DATA7*). Otras terminales importantes de la unidad de control son \overline{DSB} , \overline{CSEL} y \overline{WR} . \overline{CSEL} , se usa para permitir o no operaciones de acceso al *S1109CR*: en estado alto se desactiva todo acceso al chip; en estado bajo se permiten todas las operaciones de acceso. Este solo afecta el acceso a los registros no al Bus de Datos. A través del *data strobe* \overline{DSB} es posible que los valores que estén en el bus de datos pasan al registro de configuración seleccionado mediante los *bits* de direccionamiento *ADDR*. La entrada selectora de lectura escritura \overline{WR} se usa para controlar la dirección del acceso a los registros de configuración, o sea, si es lectura o si es escritura. Cuando \overline{WR} es alto, el flanco ascendente de \overline{DSB} se usa para cargar los datos dentro del registro seleccionado en el bus de datos de 8 *bits*. Cuando \overline{WR} es bajo, se cargan los datos en el bus de datos al registro seleccionado. Esto se puede

apreciar fácilmente en los diagramas de temporización. Con la entrada \overline{RST} en estado bajo se borran todos los datos de configuración de los registros. El siguiente cuadro sinóptico define las funciones de control descritas anteriormente

Tabla 5. Cuadro sinóptico de las funciones más representativas del S1109CR

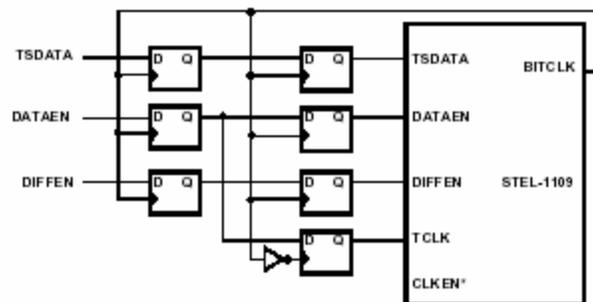
$ADDR_{5-0}$	Entrada	Seleccionar el registro a configurar o leer	Direcciona los registros
$DATA_{7-0}$	Entrada/Salida	Buffer de datos Entrada/Salida para el registro seleccionado a través de ADDR	Buffer I/O de los registros
\overline{DSB}	Entrada	Carga los datos del buffer de datos al registro o viceversa	Data Strobe negado
\overline{CSEL}	Entrada	Permite o no el acceso externo al s1109cr	Acceso a los registros
\overline{WR}	Entrada	Selecciona si se va a leer de los registros o se va a escribir sobre ellos. Utiliza al bus de datos como intermediario	Selección de lectura o escritura
\overline{RST}	Entrada	Limpia todos los registros del S1109CR	Reset negado de la configuración

En el diseño se pueden seleccionar 3 configuraciones básicas y mediante tres *switches* de usuario se le pueden agregar selectivamente Codificación *Reed Solomon*, *Scrambler* o Codificación Diferencial. El *S1109CR* tiene tres modos básicos de operación:

- Modo Maestro
- Modo Esclavo
- Modo de Prueba

Los tres modos difieren en cuanto a la sincronización y el último se caracteriza porque genera un código de Pseudo Ruido (*PN Code*) como datos a transmitir. El modo usado en este proyecto es modo maestro, entre otras cosas, porque la interfase requerida es más sencilla. La interfase recomendada por *INTEL* e incluida en el proyecto se consigue en la página 38 del *Datasheet* del integrado

Figura 15. Circuito de interfase *Modo Maestro* recomendado por *INTEL*



Intel Corporation, Op. Cit., p. 38

Esta interfase evidencia la principal característica del modo maestro: la sincronización se efectúa a través de los pulsos del *BITCLK*. Solo se necesitan 7 (siete) *flip-flop's* tipo *D* y un inversor que, en el caso del proyecto, se implementa a transistor. Por la sencillez de este último se omiten los detalles técnicos de su funcionamiento en consideración a los conocimientos que un alumno del área de comunicaciones electrónicas ya debe poseer.

OPCIONES DEL ENTRENADOR

Una vez debidamente programado el microcontrolador debe brindar las opciones descritas en la figura

Figura 16. Opciones del microcontrolador

- Modulación *BPSK* {
 - Scrambler*
 - Codificador Diferencial*
 - Reed Solomon*

- Modulación *QPSK* {
 - Scrambler*
 - Codificador Diferencial*
 - Reed Solomon*

- Modulación *16QAM* {
 - {
 - Scrambler*
 - Codificador Diferencial*
 - Reed Solomon*
 - {
 - Constelación natural*
 - Constelación Gray*
 - Constelación DAVIC*
 - Constelación Left*
 - Constelación Right*

Estas son posibles mediante la programación ordenada de los registros del *S1109CR* a través del *PIC 16F877*, el cual carga cada una de las configuraciones, es la interfase de usuario entre el entrenador y los pulsadores, y también, previa selección, la interfase de usuario entre el *PC* y el entrenador a través de su terminal *RS-232* o *USART*. Para programar el *S1109CR*, es necesario tener control sobre las direcciones de los registros a través de *ADDR₀₋₅* y además, cargar en estos para cada uno, los 8 *bits* de configuración. En total, se requiere un manejo de 14 *bits* para programar cada registro. El *PIC 16F877* dispone de cinco puertos *RA*, *RB*, *RC*, *RD*, *RE* que pueden ser debidamente programados como entradas para realizar todas las funciones de control del entrenador. De estos, el puerto A se configuró para controlar la interfase visualizadora, que es la encargada de tener al usuario al tanto del estado actual del integrado, Los 8 *bits* de puerto B son los encargados de cargar los valores al registro seleccionado mediante los 6 *bits* disponibles del puerto C. Los otros dos *bits* del puerto C se usan mediante previa configuración como los pines *RX* y *TX* del puerto *USART*, el mismo que se comunica con el *PC*. Ciertos *bits* de puerto son configurados como entradas para controlar mediante pulsadores en hardware de usuario, las configuraciones, realizadas corriendo el respectivo programa almacenado en el *PIC*. El *PIC 16F877* se amolda a las exigencias del entrenador y a las exigencias del diseño casi a la perfección. Además este contiene la interfase necesaria para tener un contacto dúplex con el ordenador *HOST*. Entre las múltiples posibilidades que ofrece el *S1109CR* están:

MODO DE TEST O DE PRUEBA. Una es que puede crear un flujo determinístico de *bits* de prueba; el *ASIC* internamente puede crear un *código PN* o de Pseudo Ruido, al cual monitorear después de pasar por el respectivo proceso de modulación. Este modo de *Test* elimina la necesidad de la programación previa y permite una prueba rápida por parte del usuario, ya sea el asistente de mantenimiento o el estudiante.

CODIFICADOR DIFERENCIAL. Un codificador diferencial permite mejoras en al tasa de *bits* transmitidos y mejora la confiabilidad de la transmisión. Como el modo de *Test*, es habilitable o deshabilitable en la programación, además este se puede habilitar con una terminal externa

CODIFICADOR REED-SOLOMON. Adicionalmente a la opción de codificación diferencial, se le puede habilitar codificación *Reed-Solomon*. Como en las anteriores es habilitable y deshabilitable en la programación[‡]

SCRAMBLER. Internamente se cuenta con una especie de aleatorizador de información serial, originalmente diseñado para evitar patrones espectrales que puedan generar símbolos redundantes en la transmisión, el *scrambler*.

[‡] En los anexos se explica con detenimiento cada una de estas técnicas de codificación y aleatorización

Todos estos son opcionales y pueden ser deshabilitados por completo para que la forma de onda a recibir en el extremo receptor de cualquiera de las prácticas, sea más interpretable por el alumno. Por la misma razón, la frecuencia a la cual el integrado generará la modulación se fija a 8 *MHz*. esta frecuencia es apta para ser interpretada en un osciloscopio estándar *FLUKE* como los del laboratorio, capaz de tomar muestras de hasta 20 *MHz*. (Algunos fabricantes no cumplen al 100% con el máximo de sus especificaciones). Los filtros *FIR* y *Nyquist* dentro del *ASIC* garantizan una señal transmitida de alta calidad. Así mismo un interpolador garantiza que los símbolos producidos en el *DAC* interno de 10 *bits*, no contengan espurias que puedan degenerar los datos transmitidos. Como se puede deducir, el *ASIC* procesa digitalmente la modulación, esto ofrece un resultado de alta calidad y además garantiza la durabilidad y precisión del equipo a lo largo del tiempo versus muchos equipos de laboratorio que presentan la desventaja de ser módulos análogos sencillos que degeneran rápidamente. Al proyecto se le añade una pequeña etapa de demodulación que permite el desarrollo de cuatro prácticas por parte del alumno.

El proyecto es un modulador, el demodulador garantiza la interpretación de la señal ya modulada. Ya que la parte mas significativa del proyecto es la modulación, el demodulador se efectúa con un resto de equipo que, como se prometió desde el principio, realiza demodulación *BPSK*. Con esto se pretende hacer el entrenador menos modular, mas compacto, solo se necesitan dos módulos físicos: el

modulador y demodulador integrados en el modulo principal y un modulo receptor que da al osciloscopio en el caso de utilizar ondas de radio, que además puede compactar el transmisor de los datos generados en el PC. A continuación se anexa una tabla con las dimensiones características de los encapsulados *MQFP* como el usado por el chip *S1109CR* (80 terminales *MQFP*)

Tabla 6. Características mecánicas de los encapsulados *MQFP*

MQFP NOMINAL PACKAGE DIMENSIONS (mm)								
Body Size	Body Thks	Lead Form	Standoff	Tip To Tip	Lead Count	JEDEC	Tray Matrix	Units Per Tray
10 x 10	2.00	1.60	0.15	13.2	44/52/64	MS-022	6 x 16	96
10 x 10	2.00	1.95	0.15	13.9	44/52/64	MO-112	6 x 16	96
14 x 14	2.00	1.60	0.15	17.2	52/64/80/100	MS-022	6 x 14	84
14 x 14	2.67	1.60	0.15	17.2	52/64/80/100	MS-022	6 x 14	84
14 x 14	2.67	1.95	0.15	17.9	52/64/80/100	MO-112	6 x 14	84
14 x 20	2.71	1.60	0.33	17.2 x 23.2	64/80/100/128	MS-022	6 x 11	66
14 x 20	2.71	1.95	0.23	17.9 x 23.9	64/80/100/128	MO-112	6 x 11	66
28 x 28	3.37	1.30	0.13	30.5	208/256	MS-029	3 x 8	24
28 x 28	3.37	1.30	0.33	30.5	120/128/144/160/208	MS-029	3 x 8	24
28 x 28	3.37	1.60	0.33	31.2	120/128/144/160/208	MS-022	3 x 8	24
32 x 32	3.40	1.30	0.38	34.5	240	MS-029	3 x 8	24
32 x 32	3.40	1.30	0.32	34.5	240	MS-029	3 x 8	24
40 x 40	3.80	1.30	0.43	42.5	304	MS-029	2 x 6	12

Amkor Technology, Inc. Metric Quad Flat Pack (MQFP). [online]; 2003. Disponible en <http://www.amkor.com/products/ProductFamilies.cfm>

REGISTROS DEL ASIC S1109CR. En esta sección se describen cada uno de los registros asociados al *S1109CR* y su respectiva función. En la hoja de información técnica o *datasheet* se pueden encontrar las distintas tablas con los respectivos registros de configuración, aquí se encuentran las mismas tablas con un pequeño recuento de los registros que contienen. En las siguientes tablas describen globalmente los campos de datos de los registros de configuración

Tabla 7. Campos de datos de los registros de configuración

Address (Hex)	Contents							
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
08 - 00	NCO ²¹							
28 - 09	FIR Filter Coefficients ¹⁸							
29	LSB Sampling Rate Control (see address 39 for MSB) ²⁰							
2A	Interpolation Filter Gain Control ¹⁹				Auxiliary Clock Rate Divider ²⁰			
2B	Set To Zero	Set To Zero	Interpolation Filt. Bypass ¹⁹		Set To One	Set To Zero	Invert I/Q Chan. ²⁰	
2C	TCLK Sel. ⁹	Set To Zero	Set To Zero	Set To Zero	MOD ¹⁴		FIR bypass ¹⁸	Set To Zero
2D	FZSINB ²¹	Bit Mapping ¹³			Set To One	Set To Zero	PN Code Sel. ⁹	PN On/Off ⁹
2E	Symbol Mapping ¹⁵			CLRFR ¹⁸				Bit Sync Re-arm ⁹
2F	Set To Zero							
32-30	SCRAMBLER Init Registers ¹¹							
35-33	SCRAMBLER Mask Registers ¹¹							
36	PPolynomial ¹³	BypassB ¹⁰	S-RS ¹⁰	Self-Sync ¹¹	T ¹²			
37	K ¹²							
38	DATAENBPB ¹⁰	DATAENSEL ¹⁰	RSENBPB ¹⁰	RSENSEL ¹⁰	SCRMENBPB ¹⁰	SCRMENSEL ¹⁰	DiHDCBPB ¹⁴	DiHDCSEL ¹⁴
39	Set To Zero	Set To Zero	TRLSBF ¹²	LDLSBF ¹²	MSB Sampling Rate Control (see address 29 for LSB) ²⁰			

Ibíd., p. 8

Los números a la derecha de cada definición de campo de Bit son las referencias a la página en el documento donde se explica con más detalle su función. En la ruta de datos se encuentran las opciones del candado de datos (*data latching*)

De aquí se puede ir tomando el *FCW A*, valor en el registro con el cual se determinará la frecuencia de la portadora, también se llenan *FCW B* y *FCW C* en caso de solicitar otros valores para pruebas. El valor del *FCW A* se calcula con información tomada del *datasheet* del *S1109CR* y se detalla mejor en el apéndice.

Según los cálculos el valor del FCW en decimal debe ser 524288 lo que representa en binario 1000000000000000000b, el llenado de los registros se describe con detalles en la siguiente tabla

Tabla 8. Valor de *FCWSEL* utilizado para configuración por defecto

FCWSEL ₁₀	FCWSEL Seleccionado	Bits 23 – 16 Registro 02h	Bits 15 – 8 Registro 01h	Bits 7 – 0 Registro 00h
00	FCW A	00001000	00000000	00000000

Con este se puede obtener una frecuencia de portadora de 447437 KHz, en el diseño también se consideró una frecuencia de reloj de 14.318 MHz

CERRADO DE DATOS (DATA LATCHING). El siguiente cuadro se muestran los registros asociados a este *data latching*

Tabla 9. Opciones de cerrado de datos

Data Source	Latched By	Register 2C Bit 7	Register 2D Bits 1,0	Mode Name
TSDATA	BITCLK	0	X,0	Master Mode
TSDATA	TCLK	1	X,0	Slave Mode
PN Code 10, 3	BITCLK	0	0,1	Test Mode
PN Code 23, 18	BITCLK	0	1,1	Test Mode

Intel Corporation, Op. Cit., p. 9

Las opciones a tomar de esta sección para el entrenador son:

- Datos transmitidos en Modo Maestro y

- Código Pseudo Ruido (*PN*) por defecto

El programa se divide entonces ya en dos bloques principales: con Código *PN* (por defecto) o con *TSDATA*.

Existen tres modos de operación para el *data latching*: modo maestro o *master mode*, modo esclavo o *slave mode* y modo de prueba o *test mode*. En el modo maestro, los datos cargados externamente son cerrados por el *BITCLK* interno. En el modo esclavo los datos cargados externamente son cerrados por el *TCLK* externo. En cualquiera de los modos de prueba, un código de pseudo ruido o código *PN* es generado internamente y cerrado por el *BITCLK* interno

CONTROL DE BUSES DE DATOS. Los datos en el *S1109CR* pueden ser encaminados a través o alrededor de los distintos codificadores seleccionando los registros apropiados. La siguiente tabla es un sumario de los valores requeridos en los registros para obtener las distintas posibilidades de trazado de rutas

Tabla 10. Opciones del bus de datos para codificación de *bit*

Data Path	Register 36 Bits 6,5	Register 38 Bits 7-2
Data stopped (continuously)	X, X	01 XX XX
Data path on (continuously)	X, X	11 XX XX
Data path enabled by pin 18	X, X	X0 XX XX
Scrambler off (continuously)	X, X	XX XX 01
Scrambler on (continuously)	X, X	XX XX 11
Scrambler enabled by pin 32	X, X	XX XX X0
RS Encode off (continuously)	1, X	XX 01 XX
RS Encode on (continuously)	1, X	XX 11 XX
RS Encode enabled by pin 29	1, X	XX X0 XX
Scrambler then RS Encoder	1, 1	XX XX XX
RS Encoder then Scrambler	1, 0	XX XX XX
Bypass RS Encoder	0, X	XX XX XX

Ibíd. p. 11

Las opciones a tomar de esta sección para el entrenador son:

- Bus de multidatos habilitado por el pin 18
- *Scrambler* habilitado por el pin 32
- Codificación *RS* habilitada por el pin 29
- Codificador *RS* luego *Scrambler*

Las funciones son elegidas entonces por los interruptores externos asociados a estos pines.

PARAMETROS DEL SCRAMBLER. Como se había comentado el *S1109CR* posee internamente su *Scrambler*. El *scrambler* se define como un dispositivo que traspone, invierte señales o también, que codifica un mensaje al transmisor haciéndolo incomprensible para un receptor no equipado con un apropiado juego de dispositivos de *descrambling*

Tabla 11. Parámetros del desmodulador o *scrambler*

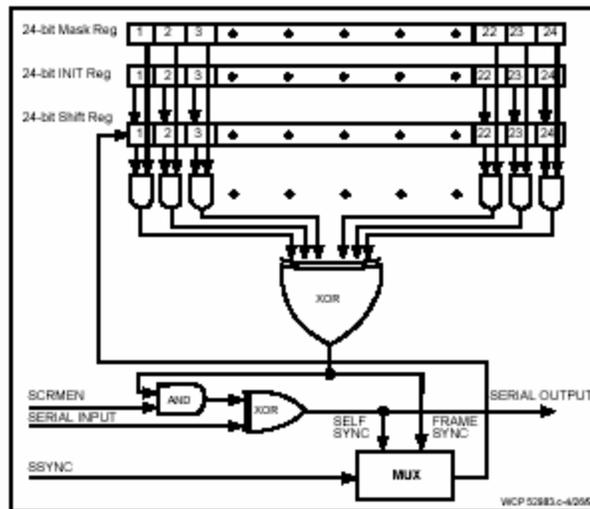
Parameter	Characteristic	Configuration Register Setting		
Generator Polynomial (Mask Reg)	$p(x) = c_{24}x^{24} + c_{23}x^{23} + \dots + c_1x + 1$ where c_i is a binary value (0, 1)	Register 35 Bit 7 to Bit 0 c_{24} to c_{17}	Register 34 Bit 7 to Bit 0 c_{16} to c_9	Register 33 Bit 7 to Bit 0 c_8 to c_1
Seed (INIT Reg)	Any 24 bit binary value, $s_{24:1}$	Register 32 Bit 7 to Bit 0 s_{24} to s_{17}	Register 31 Bit 7 to Bit 0 s_{16} to s_9	Register 30 Bit 7 to Bit 0 s_8 to s_1
Scrambler Type	Frame synchronized (sidestream)	Register 36 Bit 4 Set to zero		
Scrambler	Self-synchronized	Register 36 Bit 4 Set to one		

Ibíd., p. 11

Las opciones a tomar de esta sección para el entrenador son:

- Polinomio generador sencillo:
- Autosincronizado

Figura 17. Diagrama de bloques del desmodulador



Ibíd., p. 11

PARAMETROS DEL CODIFICADOR REED-SOLOMON

En la siguiente tabla se describen los parámetros del codificador *Reed-Solomon* como lo describe el *Datasheet* de *INTEL*

Tabla 12. Parámetros del codificador *Reed-Solomon*

Field Name	Configuration Register	Description
PP	36 _H (bit 7)	1-bit field for selecting Primitive Polynomial: $0 \Rightarrow p(x) = x^8 + x^4 + x^3 + x^2 + 1$ $1 \Rightarrow p(x) = x^8 + x^7 + x^2 + x + 1$
T	36 _H (bits 3-0)	4-bit field for setting Error Correction Capability. Programmable over the range of 1 to 10.
K	37 _H (bits 7-0)	8-bit field for setting User Data Packet Length (K) in bytes. Programmable over the range of 1 to (255 - 2T). [Net block length, $N = K + 2T$]
LDLSBF	39 _H (bit 4)	Determines whether the first bit of the serial input is to be the MSB (bit 4 = 0) or LSB (bit 4 = 1) of the byte applied to the RS Encoder.
TRLSBF	39 _H (bit 5)	Determines whether the MSB (bit 5 = 0) or LSB (bit 5 = 1) of the RS Encoder checksum byte is to be the first bit of the serial output data.
Notes: 1. GF (256). 2. Code generator polynomial 1 is used when PP=0: $G(x) = \prod_{i=120}^{119+2T} (x - \alpha^i) \Big _{\alpha = 02H}$ 3. Code generator polynomial 2 is used when PP=1. $G(x) = \prod_{i=0}^{2T-1} (x - \alpha^i) \Big _{\alpha = 02H}$		

Ibíd., p. 13

Las opciones a tomar de esta sección para el entrenador son:

- PP = 0
- T = 2 = 0010b
- LDLSBF = 0, MSB
- K seleccionable por el usuario

OPCIONES DE TRAZADOR DE MAPA DE BIT

Tabla 13. Opciones de trazador de mapa de *bit*

Mode	Bit-To-Symbol Mapping				Bit Mapping	Mod Mode
	b_0	b_1	b_2	b_3	Register 2D bits 6-4	Register 2C bits 3,2
BPSK	$I_1'' Q_1'' I_0'' Q_0''$	N/A	N/A	N/A	XXX	1X
QPSK	$I_1'' I_0''$	$Q_1'' Q_0''$	N/A	N/A	XX0	00
QPSK	$Q_1'' Q_0''$	$I_1'' I_0''$	N/A	N/A	XX1	00
16QAM	I_1''	I_0''	Q_1''	Q_0''	000	01
16QAM	Q_1''	Q_0''	I_1''	I_0''	001	01
16QAM	I_0''	I_1''	Q_0''	Q_1''	010	01
16QAM	Q_0''	Q_1''	I_0''	I_1''	011	01
16QAM	I_1''	Q_1''	I_0''	Q_0''	100	01
16QAM	Q_1''	I_1''	Q_0''	I_0''	101	01
16QAM	I_0''	Q_0''	I_1''	Q_1''	110	01
16QAM	Q_0''	I_0''	Q_1''	I_1''	111	01

Note: b_0 is the first serial data bit to arrive at the Bit Mapper

Ibíd., p. 14

Tabla 14. Control de codificador diferencial

Level/Value	Register 38 Bits 1,0
Encoding off (continuously)	0,1
Encoding on (continuously)	1,1
Encoding enabled by pin 70 high - enable the Differential Encoder low - disable the Differential Encoder	X,0

Ibíd., p. 14

La opción a tomar de esta sección para el entrenador es:

- Codificación diferencial activada por el *pin* 70

Tabla 15. Codificación diferencial *QPSK* y desplazamiento de fase

Current Input (IQ)	Current Output (IQ)	Next Output (IQ)	Phase Shift (degrees)
00	00	00	0
	01	01	-90 (CW)
	10	10	90 (CCW)
	11	11	180
01	00	01	-90 (CW)
	01	11	180
	10	00	0
	11	10	90 (CCW)
10	00	10	90 (CCW)
	01	00	0
	10	11	180
	11	01	90 (CCW)
11	00	11	180
	01	10	90 (CCW)
	10	01	-90 (CW)
	11	00	0

Ibíd., p. 15

Tabla 16. Selecciones de trazador de mapa de símbolos

Mapping Selection	Register 2E Bits 7-5
Natural	0XX
Gray	100
DAVIC	101
Left	110
Right	111

Ibíd., p. 16

Ya que estas selecciones de trazado son seleccionados por el usuario, los registros son programados de acuerdo a la solicitud de este.

Tabla 17. Trazador de mapa de símbolos

Natural Mapping (Bypass) I ₁ ' Q ₁ ' I ₀ ' Q ₀ '	Input Code				Output Code I ₁ Q ₁ I ₀ Q ₀
	Gray I ₁ ' Q ₁ ' I ₀ ' Q ₀ '	DAVIC I ₁ ' Q ₁ ' I ₀ ' Q ₀ '	Left I ₁ ' Q ₁ ' I ₀ ' Q ₀ '	Right I ₁ ' Q ₁ ' I ₀ ' Q ₀ '	
0000	0011	0011	0011	0011	0000
0001	0010	0001	0010	0001	0001
0010	0001	0010	0001	0010	0010
0011	0000	0000	0000	0000	0011
0100	0110	0110	0101	1010	0100
0101	0111	0111	0111	1011	0101
0110	0100	0100	0100	1000	0110
0111	0101	0101	0110	1001	0111
1000	1001	1001	1010	0101	1000
1001	1000	1000	1000	0100	1001
1010	1011	1011	1011	0111	1010
1011	1010	1010	1001	0110	1011
1100	1100	1100	1100	1100	1100
1101	1101	1110	1101	1110	1101
1110	1110	1101	1110	1101	1110
1111	1111	1111	1111	1111	1111

Ibíd., p. 18

Tabla 18. Opciones de configuración del filtro *FIR*

Mode	Gain	Register 2E Bits 4-1	Register 2C Bit 1
No FIR Filter	N/A	XXXX	1
16QAM	Unity	1010	0
BPSK/QPSK	Unity	0000	0
BPSK/QPSK	x2	1111	0
BPSK/QPSK	x3	1010	0

Ibíd., p. 18

Seleccionables por el usuario

Tabla 19. Almacenamiento de coeficientes del filtro *FIR*

MSB (Bits 9-8)	LSB (Bits 7-0)	Filter Taps
0A _H	09 _H	Taps 0 and 31
0C _H	0B _H	Taps 1 and 30
0E _H	0D _H	Taps 2 and 29
10 _H	0F _H	Taps 3 and 28
...
...
22 _H	21 _H	Taps 12 and 19
24 _H	23 _H	Taps 13 and 18
26 _H	25 _H	Taps 14 and 17
28 _H	27 _H	Taps 15 and 16

Note: For MSB storage, only bits 1-0 are used.

Ibíd., p. 18

Tabla 20. Control de nivel de la señal de interpolación

Gain Factor (Relative)	Filter Gain Control Register 2A Bits 7-4
2 ⁰	0 _H
2 ¹	1 _H
2 ²	2 _H
2 ³	3 _H
2 ⁴	4 _H
2 ⁵	5 _H
2 ⁶	6 _H
2 ⁷	7 _H
2 ⁸	8 _H
2 ⁹	9 _H
2 ¹⁰	A _H
2 ¹¹	B _H
2 ¹²	C _H
2 ¹³	D _H
2 ¹⁴	E _H
2 ¹⁵	F _H

Ibíd., p. 19

Tabla 21. Control de inversión de la señal

Output of Adder Block	Invert I/Q Channel Register 2B Bits 1,0
$\text{Sum} = I \cdot \cos(\omega t) + Q \cdot \sin(\omega t)$	0 0
$\text{Sum} = -I \cdot \cos(\omega t) + Q \cdot \sin(\omega t)$	0 1
$\text{Sum} = I \cdot \cos(\omega t) - Q \cdot \sin(\omega t)$	1 0
$\text{Sum} = -I \cdot \cos(\omega t) - Q \cdot \sin(\omega t)$	1 1

Ibíd., p. 20

Seleccionables por el usuario

El *FCW* es importante para la determinación de la frecuencia de la portadora del modulador. Se describe en la siguiente tabla

Tabla 22. Selección del *FCW*

FCWSEL ₁₋₀	FCW Selected	FCW Value Bits		
		23 - 16	15 - 8	7 - 0
00	FCW A	Register 02 _H Bits 7 - 0	Register 01 _H Bits 7 - 0	Register 00 _H Bits 7 - 0
01	FCW B	Register 05 _H Bits 7 - 0	Register 04 _H Bits 7 - 0	Register 03 _H Bits 7 - 0
10	FCW C	Register 08 _H Bits 7 - 0	Register 07 _H Bits 7 - 0	Register 06 _H Bits 7 - 0
11	Zero Frequency			

Ibíd., p. 22

Las opciones a tomar de esta sección para el entrenador son, colocar un valor por defecto que se ejecuta al arranque y dejar a disposición del usuario el que vuelva a escribir este registro de configuración con otros valores

REGISTROS CON VALOR PREDETERMINADO

Hay unos cuantos registros que solo se programan una vez y cuyo contenido no tiene porque variar, no importa la selección del usuario. Estos valores predeterminados se describen en el *Data Sheet* de *Intel* como *Set to 0* o *Set to 1*, dependiendo del caso; estos son empleados por el fabricante para procesos de control de calidad en la línea de producción

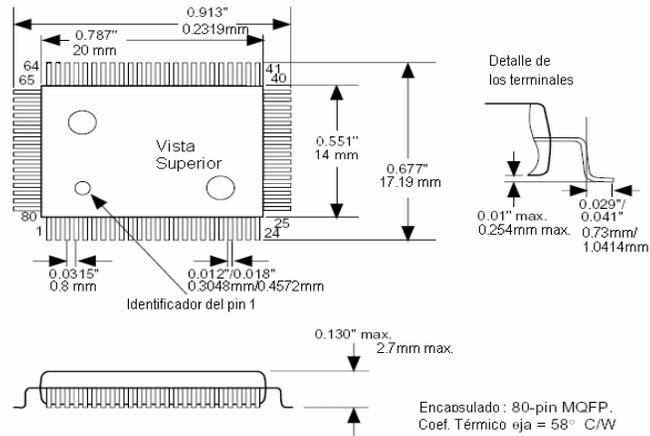
CARACTERISTICAS MECANICAS DEL S1109CR

Tabla 23. Características estándares de los encapsulados *MQFP* como el usado

Encapsulado x dimensiones (<i>Footprint</i>) (mm) PQFP 14 x 20 x 2.7 / (3.20 / 3.90)	Cantidad de terminales	Espaciado entre terminales (mm)	Tamaño de <i>Pad</i> disponible (mm)	Rango aceptable de talla muerta (mm)
	80	0.80	6.60 sq. - Cu	1.24 – 6.09
	100	0.65	6.60 sq – Cu 9.10 se. – Cu	1.24 – 6.09 3.74 – 8.59
	128	0.50	7.00 sq. – Cu 9.10 sq. – Cu 11.00 sq. - Cu	1.24 – 6.09 3.74 – 8.59 5.64 – 10.49

Estas y otras dimensiones se pueden apreciar con mayor claridad en la siguiente figura

Figura 18. Características mecánicas del ASIC S1109CR



Intel Corporation, Op. Cit., p. 39 (traducción del original)

5.1.3 UNIDAD DE CONTROL: EI PIC16F877. El microcontrolador *PIC16F877* es el que realiza las funciones de control del *ASIC*: configura el *ASIC* de acuerdo a las escogencias del usuario, gracias a que guarda el programa de configuración en su memoria interna y es capaz de intercambiar datos a través del puerto *USART* con un *PC* externo

El *PIC16F877* es un microcontrolador de la gama media de microcontroladores *MICROCHIPS*. Estos se pueden considerar como una combinación de las virtudes del *PIC16F84* con la inclusión de los recursos de los *PIC16C73* y *74*. Incorporan la memoria *FLASH* con una capacidad de 4 K y 8 K palabras de 14 *bits*, sin cambiar la estructura interna del procesador y conservando el mismo repertorio de instrucciones

El puerto A esta dedicado a manejar el panel de visualizadores, también llamado aquí interfase gráfica, pues es la encargada de poner al tanto al usuario del estado actual del entrenador. Al puerto A va conectado con o sin adaptadores al visualizador *LCD*, inicialmente estos 6 *bits* se iban a interconectar con un juego de *LED's* como indicadores. La mayoría de los visualizadores tienen un bus de datos de 8 *bits*, además de los terminales de alimentación pero también es cierto que estos *LCD's* pueden configurarse en la secuencia de inicio para manejar un bus de 4 *bits*. Los 8 *bits* del puerto B del microcontrolador RB_0-RB_7 (pines 33 al 40) son los encargados de intercambiar datos con el *ASIC*, estos se conectan con los 8 terminales $DATA_0-DATA_7$ del integrado modulador, terminales 76, 77, 78, 79, 2, 3, 4 y 5 respectivamente. Los 6 *bits* de direccionamiento se manejan a través del

puerto C del *µC*, terminales 15, 16, 17, 24 y 25. Como es clásico en montajes con microcontroladores el cristal oscilador que resuena a 4MHz, se encuentra al lado del integrado conectado a los terminales 15 y 16 del mismo. Este es opcional, con el *DIP Switch 1* se puede seleccionar entre si usar un cristal para el microcontrolador o trabajar con el mismo reloj principal del entrenador, o inclusive, llevar a lógica cero permanente la actividad de este reloj, colocando su salida a tierra. La primera parte entonces para dar paso al diseño de las funciones del microcontrolador es definir que instrucciones necesita ejecutar, estas dependen de las necesidades del *S1109CR*. Las funciones que tiene que desempeñar el *µC* se pueden clasificar en

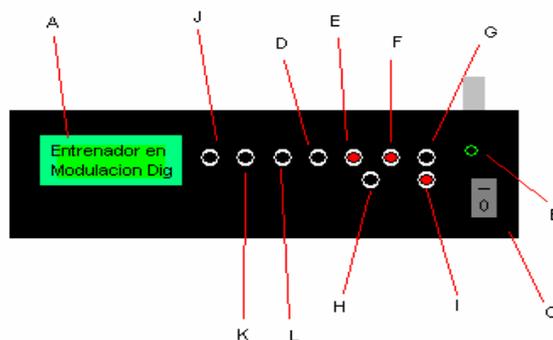
- ◆ Configurar el *ASIC* al encendido del equipo con una configuración base o por defecto y permitir una reconfiguración por parte del usuario en estado de funcionamiento
- ◆ Una vez configurado, estar disponible para comunicarse con un ordenador a través del puerto serie
- ◆ Visualizar tanto la configuración actual del *ASIC* como la acción que ejecuta en ese instante, ya sea a través del visualizador local o por mensajes enviados al *PIC*
- ◆ Recibir comandos del usuario y enviar mensajes a este ya sea a través del ordenador o de la interfase integrada. Esto incluye el manejo del puerto *USART*

El *PIC* es alimentado por los 5 voltios de la fuente y esta temporizado por un cristal de exactamente 4 MHz, oscilador de referencia estable y necesaria a la hora de fijar parámetros no variables como el *baud rate* del puerto serial y los retardos en el *firmware* para ciertas funciones de visualización.

FUNCIONES CONTROLADAS POR HARDWARE. BOTONES DE USUARIO

El diseño original del entrenador era mas rustico que el definitivo. Se consideraron varias opciones para el panel de botones de funciones dispuestos en la parte frontal de este. Gracias a ventajas como usar una pantalla *LCD* que dada su alta eficiencia por contar con un microcontrolador integrado y posibilidad de funcionar con un bus de solo 4 *bits*, permitió tener más salidas del microcontrolador disponibles, fue posible contar con hasta 7 entradas del *PIC* para recibir comandos. Es así que finalmente el panel de botones del entrenador quedo dispuesto como muestra la figura siguiente

Figura 19. Vista frontal definitiva del entrenador



En el panel se tienen las siguientes funciones y las de control están relacionadas con las entradas del *PIC* como se describe a continuación

Interruptor C. Encendido y apagado del entrenador. Es un interruptor de alto voltaje y corriente que energiza y desenergiza el entrenador a través de la conexión y desconexión de el transformador alimentador de la red de 120 Voltios *RMS*

Botón G: Escribir registro. Con este se predispone el equipo para recibir un comando de configuración de entrada binaria. Este comando de configuración consta de una dirección y un valor asociado. La dirección, es la dirección del registro a escribir, el valor, es el *byte* de configuración asociado a ese registro. Una vez presionado este botón el entrenador va a pedir los dos *bytes*, empezando por el de la dirección del registro para continuar con el valor asociado. Los valores del *byte* se insertan a través de los botones E y F los cuales aplican un 1 y un 0 respectivamente. Está conectado a la entrada RD4 del puerto D del PIC con su resistencia levantadora de 10 K Ω (*pull-up resistor*) en paralelo conectada a +5V

Botón D: Leer registro. A través del botón D se puede leer un registro del equipo para examinar así el valor de configuración contenido en este. Como en el caso anterior, se comienza insertando el *byte* de dirección, con el cual el microcontrolador busca en el *ASIC* para luego mostrar el valor de configuración

contenido en este. Al presionar el botón, se deja atrás el mensaje de configuración por defecto y aparece un mensaje solicitando la información anteriormente descrita

Está conectado a la entrada RE2 del puerto E del *PIC* con su resistencia levantadora de 10 K Ω (*pull-up resistor*) también conectada +5V

Figura 20. Mensaje de solicitud de dirección



Al mostrar este mensaje el equipo está listo para recibir, *bit a bit*, el número de la dirección del registro a explorar. Este se introduce desde el *LSB* o *bit* menos significativo, desplazándose gráficamente hacia la derecha (vea botones E y F)

Al terminar de introducir el *byte*, aparece por 3 segundos una pantalla informativa, mostrando el respectivo valor de dirección introducido pero convertido a decimal y el valor de configuración actual asociado a ese registro. En ninguno de los dos casos puede ser mayor a su equivalente decimal 58, pues nada más hay 58 registros a configurar. Si por error al usarlo se introduce un número binario cuyo

equivalente decimal sobrepasa los 58 registros, se dará un mensaje de error, como se muestra en la figura

Figura 21. Mensaje de error de registro



Botón F: Insertar 1 binario. Como se explico anteriormente, presionando este botón, se inserta un uno en el *byte* binario a insertar. Al llenar los 8 espacios del *byte* este automáticamente se almacena en el equipo. Esta conectado a la entrada RD3 del puerto D del microcontrolador fijado a lógica 1 por una resistencia levantadora de 10 K Ω (*pull-up resistor*)

Botón E: Insertar 0 binario. Presionando este botón, se inserta un cero en el *byte* binario a insertar. Al llenar los 8 espacios del *byte* este automáticamente se almacena en el equipo. Esta conectado a la entrada RD2 del puerto D del microcontrolador también con su resistencia levantadora de 10 K Ω (*pull-up resistor*)

Botón H: Borrar *bits*. Es el pulsador de “borrón y cuenta nueva”. Mediante este el usuario puede rectificar desde uno hasta la totalidad de *bits* digitados mientras

escribe con cualquiera de los comandos de lectura o escritura de registro. Esta conectado a la entada RA4 del puerto A del microcontrolador con su respectiva resistencia levantadora de $10K\Omega$ (*pull-up*)

Botón I: Reset General. Al presionar este pulsador, todos los registros del *S1109CR* y el *PIC* quedan a cero, lo mismo que si se hubieran escrito todos con el valor $0x00h$. Esta conectado directamente al *S1109CR* a la entrada *RST* o de *reset* la cual se lleva a estado alto en condiciones normales (botón no oprimido) por su respectiva resistencia de levantamiento de $10K\Omega$ (*pull-up*)

Botones J, K Y L. Botones para activar/desactivar las funciones y codificaciones posibles en el entrenador: J para Codificación diferencial, K para *Reed Solomon* y L para el *Scrambler*. Las 3 entradas tienen sus respectivas resistencias de *Pull-Up* de $10 K\Omega^{\S}$

CONFIGURACION COMO INTERFASE *UART*. Para ser totalmente compatible con el *PC*, el microcontrolador debe trabajar en modo asíncrono, esto se logra mediante la configuración de los respectivos registros. El *USART* también llamado *SCI*, puede funcionar como un sistema de comunicación *full duplex* o bidireccional asíncrono, adaptándose a multitud de periféricos y dispositivos como monitores y *PC*'s. También puede trabajar en modo síncrono unidireccional a *half duplex* para

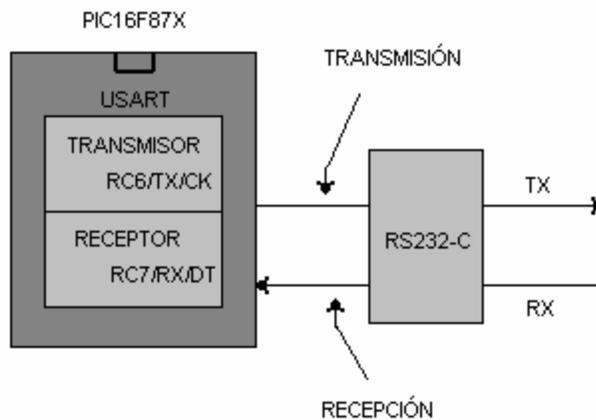
[§] Para circuitos CMOS es suficiente con una resistencia de $10K\Omega$ para establecer un nivel de lógica alto, por su alta impedancia de entrada

trabajar con periféricos como memorias, conversores, etc. Aunque para fines del proyecto el *PIC* debe trabajar como *UART* se procede a listar los tras modos posibles de trabajo:

- *Asíncrono (full duplex bidireccional)*
- *Síncrono Maestro (Half duplex, unidireccional)*
- *Síncrona Esclavo (Half duplex unidireccional)*

Un diagrama de bloques funcional de cómo realiza esto se muestra en la figura 22

Figura 22. Bloque funcional del puerto *USART* en modo asíncrono



Cabe recordar algunas características del protocolo *RS-232-C* como por ejemplo los dos *bits* básicos que forman una palabra de datos. Una palabra de datos o trama *RS-232-C EIA* esta formada básicamente un *bit* de inicio o *START BIT* y un *bit* de parada o *STOP BIT*, entre estos *bits* se encuentra el contenido, la

información en la palabra la cual suele estar conformada por 8 *bits* de datos, que pueden ser mas, para el *PIC16F877* está limitada a 9 *bits* máximo con consecuentes ligeras modificaciones en la configuración. Los cuatro bloques que configuran la arquitectura del *USART* en el *µC* son

- Circuito de muestreo
- Generador de baudios
- Transmisor asíncrono
- Receptor asíncrono

El circuito de muestreo actúa sobre la pata *RC7/RX/DT*, que es por donde se recibe el *bit* de información o control y se encarga de muestrear tres veces su valor, para decidir este por mayoría.

GENERADOR DE BAUDIOS. En el protocolo, la frecuencia en baudios a la que se realiza la transferencia se debe efectuar en un valor normalizado: 330, 600, 1.200, 2.400, 4.800, 9.600, 19.200, 38.400, etc. Para generar esta frecuencia, el *USART* dispone de un Generador de Frecuencia en Baudios, *BRG*, cuyo valor es controlado por el contenido grabado en el registro *SPBRG*. Además del valor *X* cargado en el registro la frecuencia de baudios del generador depende del *bit BRGH* del registro *TXSTA<2>*. En el caso de que *BRGH* sea 0 se trabaja en baja velocidad y si *BRGH* = 1 se trabaja en alta velocidad. De acuerdo a este *bit* se

obtendrá el valor de una constante k necesaria para la determinación de la frecuencia de funcionamiento

$$\text{Baudios} = \frac{F_{osc}}{k.(x+1)}$$

Donde

$Baudios$	=	frecuencia en Baudios
F_{osc}	=	frecuencia del reloj del PIC
x	=	valor cargado en el registro SPBRG
k	=	constante. Depende de BRGH en TXSTA<2>

Como se especifica al final la constante k depende del valor cargado en el *bit BRGH*: si $BRGH = 1$ entonces $K = 16$ y si $BRGH = 0$ se tiene $K = 64$. Despejando

$$x = \frac{F_{osc}}{(Baudios)(k)} - 1$$

En la tabla siguiente se presenta la estructura interna junto a otros datos de interés de los registros que intervienen en la determinación de la frecuencia en baudios a la que transfiere la información el *USART*

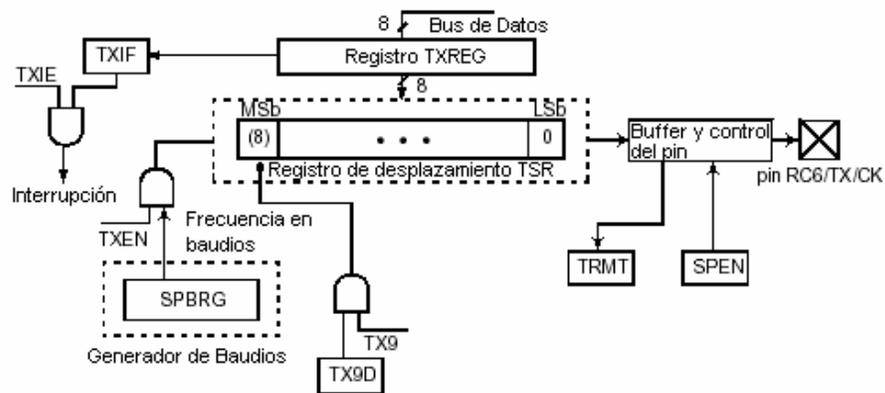
Tabla 24. Características relevantes de los registros que intervienen en la determinación de la frecuencia en baudios de la transferencia de información en el modo asíncrono del *USART*

Dir	Nombre	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0	Valor en por. Bor	Valor en el resto de Resets
98h	TXSTA	CSRC	TX9	TXEN	SYNC	--	BRGH	TRMT	TX9D	0000 -010	0000 -010
18h	RCSTA	SPEN	RX9	SREN	CREN	ADDE N	FERR	OERR	RX9D	0000 000x	0000 000x
99h	SPBRG	Registro Generador de Baudios							0000 0000	0000 0000	

Angulo Usategui José Ma, Romera Yesa Susana y Angulo Martinez Ignacio. Microcontroladores PIC: Diseño práctico de aplicaciones. Madrid, 2000, p. 194

EL TRANSMISOR ASINCRONO.

Figura 23. Diagrama de bloques de la sección transmisora del *USART* en modo asíncrono



Microchip Technology Inc. PIC 16F87X Data Sheet. 2001 [eBook]. [citado 11-11-2003], p. 99

Disponible en <http://www.microchip.com/download/lit/pline/picmicro/families/16f87x/30292c.pdf>

El dato que se desea transmitir por el *USART* transmisor de la figura se deposita en el registro *TXREG* y a continuación se traspassa al registro de desplazamiento *TSR*, que va sacando los *bits* secuencialmente y a la frecuencia establecida. Además se añaden los *bits* de inicio y de parada. El *USART* receptor recibe uno a uno los *bits*, elimina los de control y los de información una vez llenado el registro de desplazamiento *RSR* los traslada automáticamente al registro *RCREG*, donde quedan disponibles para su posterior procesamiento.

El núcleo del sistema está constituido por el registro de desplazamiento *TSR*, que obtiene es dato desde el registro *TXREG* y luego lo va desplazando y sacando *bit* a *bit*, en serie, por el pin *RC6/TX/CK*. El primer *bit* que sale es el de menos peso. El dato a transmitir se carga por software en *TXREG* y se transfiere al *TSR* cuanto se haya transmitido el *bit* de parada del dato anterior. La transferencia entre los dos registros en un ciclo y entonces, el señalizador *TXIF* se pone a 1, para advertir que el registro de transmisión se ha vaciado. También en este momento puede producirse una interrupción si se ha habilitado programando el *bit* *TXIE* = 1 en el registro *PIE1<4>*. Cuando se escribe otro dato sobre *TXREG*, el señalizador *TXIF* se pone a 0. El *bit* *TRMT* sirve para indicar el estado del registro *TSR* y vale 1 cuando esta vacío

Tabla 25. Características más significativas de los registros asociados con la transmisión asíncrona

Dirección	Nombre	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0	Valor en Por. Bor	Valor en resto de resets
8Ch	PIE1	PSPIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
0Ch	PIR1	PSPIF	ADIF	RCIF	TXIF	SPPIF	CCPIE	TMR2IF	TMR1IF	0000 0000	0000 0000
98h	TXSTA	CSRC	TX9	TXEN	SYNC	---	BRGH	TRMT	TX9D	0000 -010	0000 - 010
19h	TXREG	Registro de transmisión del USART								0000 0000	0000 0000
18h	RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 000x	0000 000x
99h	SPBRG	Registro Generador de Baudios								0000 0000	0000 0000

Angulo Usategui José Ma, Romera Yesa Susana y Angulo Martinez Ignacio, Op. Cit., p. 197

Tabla 26. Registro *TXSTA*

CSRC	TX9	TXEN	SYNC	---	BRGH	TRMT	TX9D
------	-----	------	------	-----	------	------	------

Ibíd., p. 197

CSRC Bit de selección del reloj

Modo Asíncrono: no influye

Modo Síncrono:

1 = Modo maestro (reloj generado internamente desde BRG)

0 = Modo esclavo (reloj generado por una fuente externa)

TX9. Habilita el noveno *bit* de transmisión

1 = Selecciona la transmisión de 9 *bits*

0 = Selecciona transmisión de 8 *bits*

TXEN

1 = Transmisión activada

0 = Transmisión desactivada

SYNC

1 = modo síncrono

0 = modo asíncrono

BRGH. Bit de selección de la velocidad de baudios

Modo asíncrono:

1 = alta velocidad

0 = baja velocidad

Modo Síncrono: no se usa

TRMT. Bit de estado del registro de desplazamiento de transmisión

1 = TSR vacío

0 = TSR no vacío

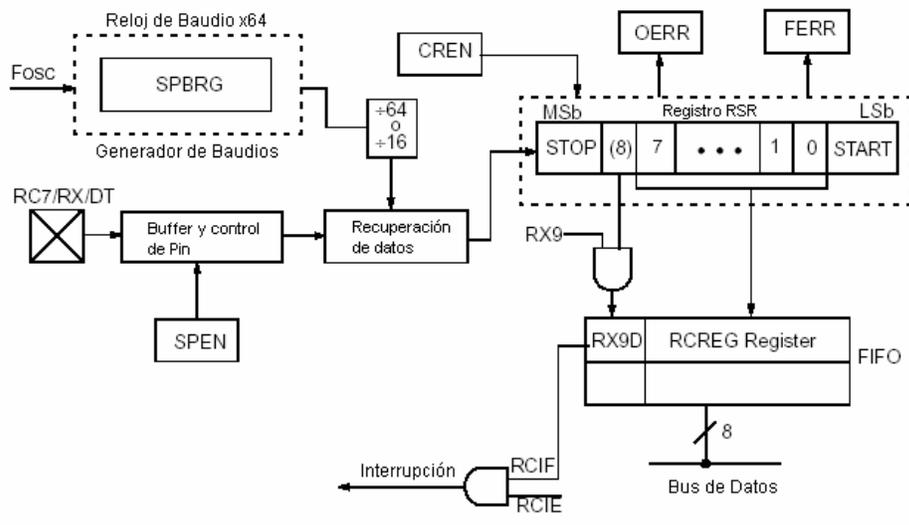
TX9D. Bit 9 del dato a transmitir (puede ser el *bit* de paridad)

CONFIGURACION COMO RECEPTOR ASINCRONO. Los datos se reciben en serie, *bit a bit*, por la pata** *RC7/RX/DT* y se van introduciendo secuencialmente en el registro de desplazamiento *RSR* que funciona a una frecuencia 16 veces mas rápida que la de trabajo. Cuando el dato consta de 9 *bits* hay que programar el *bit RX9 = 1* y el noveno *bit* de información se colocará en el *bit RX9D* del registro del *RCSTA*. Obsérvese en la figura el control sobre el noveno *bit* con las puertas de control y las señales que se aplican (*ADDEN = 1*). Cuando *CREN = 1* en el registro *RCSTA<4>*, se habilita la recepción. Si un procesador maestro intenta enviar información a uno de los esclavos, primero envía un byte de dirección que identifica al destinatario. El byte de dirección se identifica porque el *bit RX9D* que llega vale 1. Si el *bit ADDEN = 1* en el esclavo se ignoran todos los *bits* de datos. Pero si el noveno *bit* que se recibe vale 1, quiere decir que se trata de una dirección y el esclavo provocará una interrupción, y se transferirá el contenido del registro *RSR* al buffer de recepción.

Tras la interrupción, el esclavo examina la dirección y si coincide con la suya pone *ADDEN = 0* para poder recibir datos del maestro. Si *ADDEN = 1* como los datos son ignorados, el *bit* de parada no se carga en *RSR*, por lo que este hecho no produce interrupción.

** El termino PIN de origen de habla inglesa se refiere a las pequeñas patas de los integrados, que los españoles llaman "patitas", es un anglicismo usado en este documento algo popular a veces usar "pin" en vez de pata o patita

Figura 24. Diagrama de bloques del receptor *USART*



Microchip Technology Inc., Op. Cit., p. 101

Tabla 27. Registro *RCSTA*

SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D
------	-----	------	------	-------	------	------	------

Angulo Usategui José Ma, Romera Yesa Susana y Angulo Martinez Ignacio, Op. Cit., p. 197

SPEN. Habilitación del puerto serie

1 = Puerto serie habilitado (se configuran RC7/RX/DT y RC6/TX/CK)

0 = Puerto serie deshabilitado

RX9. Habilita el *bit* 9 de recepción

1 = Selecciona recepción de 9 *bits*

0 = Selecciona recepción de 8 *bits*

SREN. Configura la recepción sencilla

Modo Asíncrono: no influye

Modo síncrono maestro:

1 = Habilita reopción sencilla

0 = Deshabilita recepción sencilla

Modo síncrono esclavo: no se utiliza

CREN. Configura la recepción continua

Modo asíncrono:

1 = habilita modo de recepción continua

0 = Deshabilita recepción continua

Modo Síncrono

1 = Habilita recepción continua hasta que el *bit* CREN es borrado

0 = Deshabilita recepción continua

ADDEN. Detección de dirección

Modo asíncrono con 9 *bits* (RX9 = 1):

1 = activa la detección de dirección, activa la interrupción y descarga del buffer de recepción al activarse RSR<8>

0 = desactiva la detección de dirección, todos los *bits* son recibidos y el *bit* 9 puede ser utilizado como paridad

FERR. Bit de error de trama

1 = error de trama (puede ser actualizado leyendo RCREG y recibiendo el siguiente dato válido)

0 = no hay error de trama

OERR. Bit de error de sobrepasamiento u *overflow*

1 = error de sobrepasamiento (puede ser borrado escribiendo un 0 en el *bit* CREN)

0 = no hay error de sobrepasamiento

RX9D. Bit 9 del dato recibido (puede ser el *bit* de paridad)

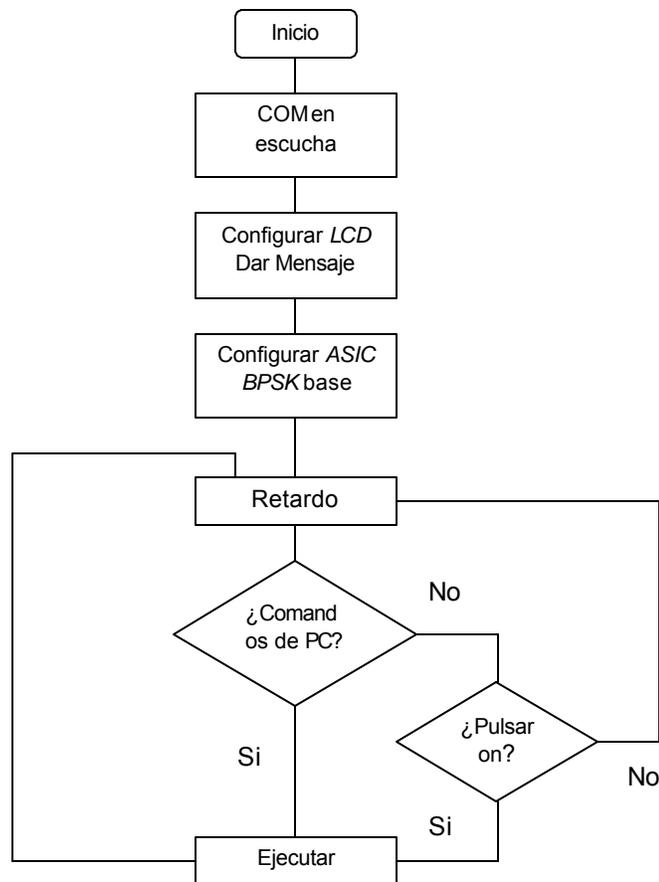
Tabla 28. Características más relevantes de los registros que intervienen en la recepción asíncrona

Dirección	Nombre	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0	Valor en Por. Bor	Valor en resto de resets	
8Ch	PIE1	PSPIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000	
0Ch	PIR1	PSPIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000	
98h	TXSTA	CSRC	TX9	TXEN	SYNC	---	BRGH	TRMT	TX9D	0000 - 010	0000 - 010	
18h	RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 0000	0000 0000	
1Ah	RCREG	Registro de Recepción del USART									0000 000x	0000 000x
99h	SPBRG	Registro Generador de Baudios									0000 0000	0000 0000

Angulo Usategui José Ma, Romera Yesa Susana y Angulo Martínez Ignacio, Op. Cit., p. 200

El programa que realiza estas funciones es bastante extenso y el número de subrutinas es amplio, el código de este en lenguaje ensamblador se encuentra en los anexos del trabajo. Resumiendo, el algoritmo descrito tiene un diagrama de bloques como el descrito en la siguiente figura

Figura 25. Diagrama de bloques del programa del *PIC16F877*



La configuración por defecto implica cargar los 58 registros del *S1109CR* con unos valores predeterminados. Antes de programar cualquier configuración se definen entonces los valores de los registros como a continuación

CONFIGURACIÓN BASE: MODO BPSK. En la tabla 29 se describen los valores cargados en los registros del *STEL-1109* por defecto como rutina del *PIC*: los no descritos tienen 00h o 00000000₂ como valor de configuración

Tabla 29. Valores de configuración por defecto

<i>Dirección de Registro</i>	<i>Dirección en hexadecimal</i>	<i>Valor de configuración</i>
2	02h	00001000 ₂
5	05h	00010000 ₂
9	09h	00000001 ₂
41	29h	11111111 ₂
42	2Ah	00111111 ₂
43	2Bh	00001000 ₂
44	2Ch	00001010 ₂
45	2Dh	10001010 ₂
53	35h	01000001 ₂
54	36h	00000012 ₂
55	37h	00000001 ₂
56	38h	00000000
57	39h	00011112 ₂

Estos valores de configuración conducen al siguiente resultado:^{††}

FCWA = 000010000000000000000000; FCWB = 000100000000000000000000;
 N = 111111111111; K = 00000001; T = 1; PN = OFF CODIGO X,X; GANANCIA DEL FILTRO DE INTERPOLACION = 8; DIVISOR DEL RELOJ AUXILIAR = 0; MODO BPSK; TCLK = 0; COEFICIENTE DE FILTRO FIR = 1; DATAPATH ACTIVADO POR LOS TRES PULSADORES A LA IZQUIERDA DEL ENTRENADOR; FILTRO FIR DESACTIVADO; ETAPAS FILTRO DE INTERPOLACION = 3; CONSTELACION BPSK NORMAL; MASCARA DEL SCRAMBLER

^{††}. Los *bits* marcados con negrilla son valores predeterminados. El color relaciona cada resultado con sus bits de configuración

5.1.4 BASE DE TIEMPO – GENERADOR DE RELOJ. El oscilador que genera la base de tiempo con la que funcionan los elementos del entrenador (principalmente el *S1109CR*) esta representado por un circuito *Pierce* con *TTL* que genera una frecuencia de 14.318 MHz. Las características de este oscilador son una frecuencia bastante estable y que necesita un transistor con una ganancia de voltaje elevada (alrededor de 70) para un montaje discreto. El primer parámetro, frecuencia bastante estable, es apropiado para los requerimientos del circuito. Existen opciones más económicas a gran escala y versátiles pero no accesibles en Colombia, como el oscilador *CY2071A* que con 5 voltios genera hasta 100 MHz. Por la inaccesibilidad al componente en el comercio local y hasta nacional se optó por fabricar un generador de reloj alternativo a base de *TTL*. El *TTL* a pesar de consumir más corriente que el *CMOS* es mas apto a la hora de trabajar a frecuencias superiores a 1 MHz. Estos pueden trabajar hasta a 40 MHz sin problemas. Aún así no fue posible conseguir circuitos más específicos para la generación del reloj, como por ejemplo, el *74124* que es un *VCO* muy apto para el uso. Se optó entonces por usar compuertas *TTL* para implementar el generador de reloj. Estas por presentar características como inversión de fase (inversores) ganancia de corriente y hasta cierta ganancia de voltaje (de compuerta a compuerta) pueden cumplir fácilmente con criterios como el de *Barkhausen*. El diseño utiliza un *74HC04* que no son más que 6 inversores contenidos en un chip. La ventaja de este *TTL* sobre la mayoría de los diseños *Pierce* con *FET*

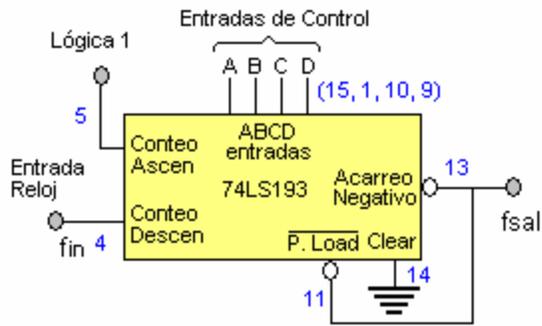
es que resulta menos voluminoso al no usar inductores ni acoples inductivos voluminosos, reduciendo el uso de componentes pasivos difíciles de implementar como son los inductores. Siempre se cuida además que la máxima excursión de salida no exceda los 5 voltios para los circuitos con $V_{DD} = 5$ Voltios y 3.3 voltios para los circuitos con $V_{DD} = 3.3$ Voltios. Estas precauciones se toman a la salida de un circuito integrado *TTL* divisor programable o *Divide by N* con un N binario de 4 *bits* el cual puede dividir la frecuencia del reloj de acuerdo a la siguiente expresión

$$f_{out} = \frac{f_{in}}{1A + 2B + 4C + 8D}$$

Que es lo mismo que dividir la frecuencia de entrada entre el equivalente decimal formado por los dígitos binarios ABCD. Cabe recordar que el microcontrolador trabaja con un cristal de 4MHz, independiente del reloj del *STEL-1109*. No existen grandes problemas por la diferencia de frecuencias entre dispositivos ya que el proceso de configuración del *ASIC* es totalmente asíncrono, es controlado por el *PIC*. El proceso que necesita sincronización es el de envío de datos del *PIC* al *ASIC* y sin embargo, ya existen dentro de este un bloque de sincronización de *bit* que en el montaje esta auxiliado por el circuito de candados o *LATCH's* recomendado por los mismos ingenieros de la *INTEL* para trabajo en modo maestro. Se escogió el *Master Mode* porque exige menos software y maneja las entradas que el usuario va a utilizar en las prácticas. Como referencia un ciclo de instrucción en el *PIC* es 4 veces un ciclo del oscilador, o sea que para el *PIC* usado con un cristal de 4 MHz, las instrucciones tienen un periodo de 1 μ s, lo cual cumple aun con *Nyquist* al considerar el muestreo de datos efectuado por el *ASIC* si se

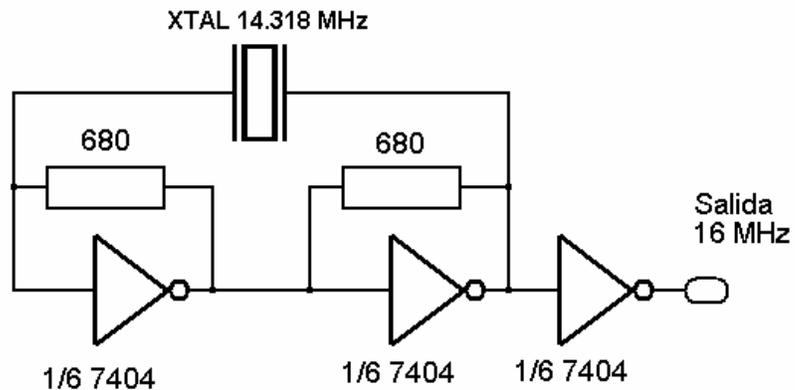
considera que este trabaja a mínimo unos 2 MHz. El divisor programable se describe a continuación

Figura 26. Diagrama de bloques del divisor programable



Ya en el diseño la segunda condición se logra con dos diodos de silicio en serie orientados a la entrada del reloj (*CLK*) del *S1109CR*. La versatilidad ofrecida por el divisor programable permitió hacer ciertas pruebas de diseño adicionales en el proyecto terminado

Figura 27. Generador de Reloj implementado con el *74HC04*

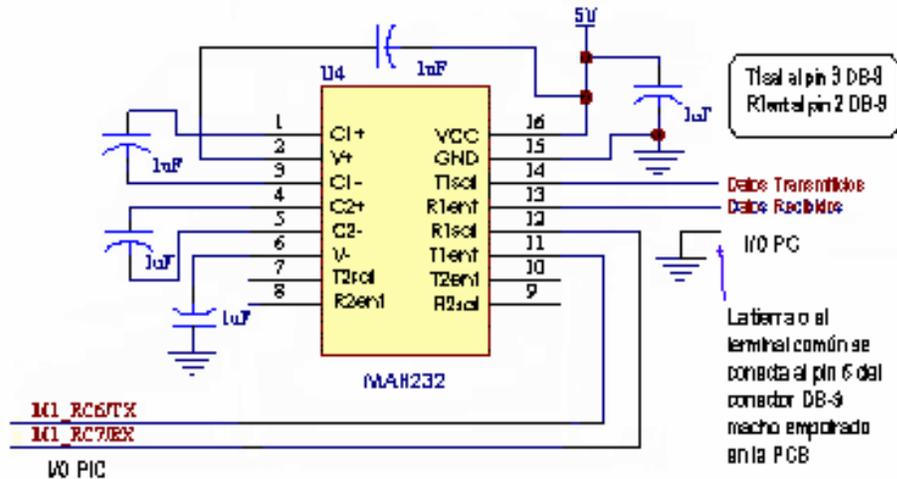


5.1.5 TECLADO DE USUARIO. INTERFASE TECLAS DE USUARIO. Funciones como el encendido y el apagado, o la selección de si se va a modular o demodular quedan aquí definidas, no solamente por el *PC*, algo que lo hace más independiente de este. Tecnología: *CMOS* o *TTL drivers* o decodificadores de teclado, teclado de membrana o pulsadores

5.1.6 PUERTO E/S: CONECTOR DB-9, USART/RS-232-C y MAX232. El *µC* *PIC16F877*, como se detalla en las sección dedicada al microcontrolador, tiene incorporada la habilidad de manejar el puerto serie *USART* en modo síncrono o asíncrono. Para la interconexión con el *PC* se trabaja en modo asíncrono, necesitándose un pin del *PIC* como transmisor o *TX* (PIN 26) y una como receptor o *RX* (PIN 25) los cuales van dirigidos al integrado *MAX232*; este toma los valores de voltaje del *µC* y los convierte a los respectivos valores de voltaje, $\pm 10V$, para cumplir con los requerimientos *NRZ* del protocolo *RS-232-C*. Para completar el conjunto, una tercera es la de tierra (*GND*), la que hace comunes los potenciales del *PC* y del entrenador. Un *generador de baudio* en el *µC*, determina la velocidad de la transferencia de datos, donde el valor máximo a escoger es 9600 baudios. La conexión física al *PC* se hace a través de un conector *DB-9* macho empotrado en la *PCB* y con salida externa al entrenador a través de un cable *BUS*. La configuración del conector para este propósito es sencilla.

En la siguiente gráfica se describe el *MAX232* y las tres terminales usadas necesarias para, con estas, poder comunicarse con el *PC*.

Figura 28. El *MAX232*

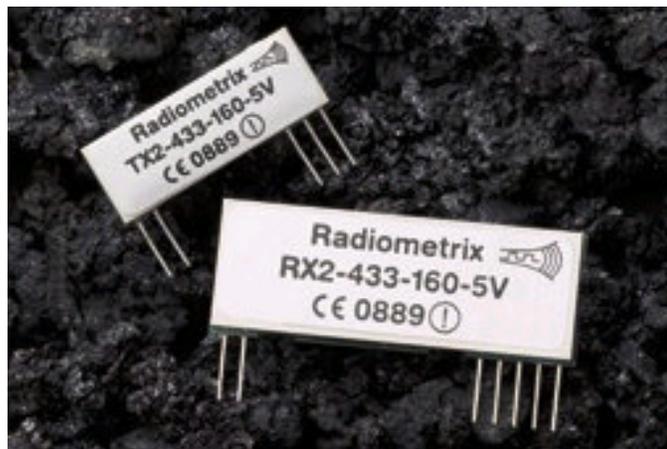


- Para el *MAX232* los condensadores de 1uF toman el valor de 0.1uF
- Los terminales *R0C* y *TX* corresponden a los pines 26 y 25 del *uC* respectivamente

Hay que destacar que el conector *DB-9* empleado cumple con la distribución de señales para *RS-232C*, significa esto que para conectarse con el *PC*, el cable serie debe estar cruzado, convirtiéndose en modem nulo. Cruzar un cable que solo contiene un par útil y una tierra común es muy sencillo: la pata 2 de uno (*RX*) se conecta a la pata 3 del otro (*TX*) y viceversa, la tierra permanece igual (pata 5 a pata 5).

5.1.7 ETAPA DE RADIO: MODULOS RADIOMETRIX. Uno de los elementos de la práctica es la transmisión de información mediante ondas de radio, para tal efecto se escogieron los módulos mas compactos, modernos y eficientes que se pudieron conseguir en el mercado. Los módulos británicos de marca *Radiométrix* permiten comunicación entre dos puntos a una distancia mínima de 300 metros utilizando modulación *FM SAW*, esto significa que para transmitir por radio se hizo necesario aplicar a la ya obtenida modulación digital otra modulación que permitiera la inmediata conversión de la frecuencia de 80 Khz. a una frecuencia en la banda *UFH* de 433 MHz que viaja muy fácilmente por el aire.

Figura 29. Aspecto real de los módulos *RX2/TX2* de *Radiometrix*



Radiometrix Ltd. TX2 & RX2 Data Sheet [online], 2001. Issue G. [Citado 10-12-2002].

Disponible en <http://www.artbrno.cz/Radiometrix/dsheets/tx2rx2.pdf> de forma gratuita

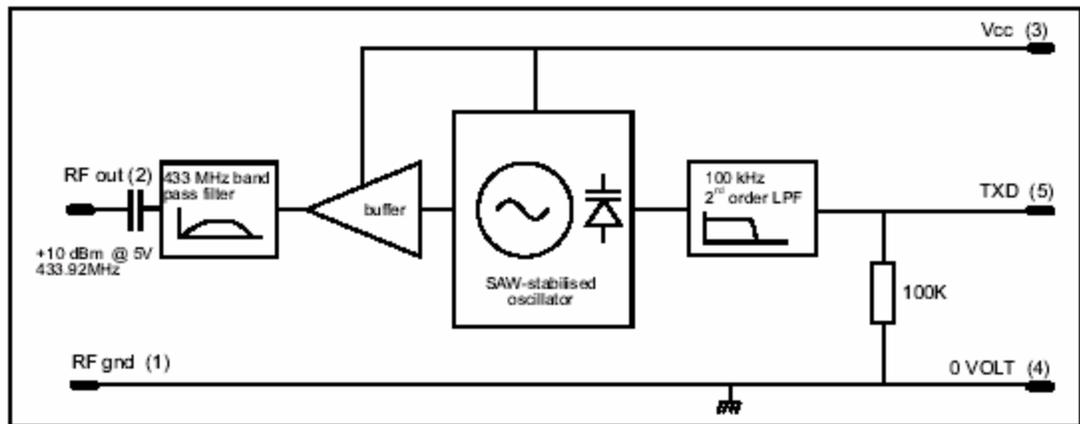
El *ASIC* es muy eficiente, este puede trabajar con tasas de *bits* mayores de 20 Mbps a altas frecuencias de reloj. Entre otras cosas a la hora de transmitir por radio, esta tasa se reduce a la máxima tasa ofrecida por el transmisor utilizado que es de 40 Kbps, que es más que suficiente para aplicaciones didácticas de laboratorio. Los módulos utilizados son los transmisores y receptores miniatura de la serie *TX2/RX2*, específicos para enlaces de datos. Entre sus características se encuentran:

- Certificados por la *CE* por cuerpo independiente notificado
- Verificados para cumplir con el radio estándar de armónicos EN 300 220-3 por prueba en laboratorio acreditado
- Verificados para cumplir con estándar *EMC* de armónicos En 301 489-3 por prueba en laboratorio acreditado
- Tasas de datos de hasta 160 Kbps (los usados alcanzan hasta 40 Kbps)
- Rango de uso de hasta 300 metros
- Totalmente apantallados
- Versiones de hasta 433.92 MHz

Estos cumplen con las normas *EMC* europeas de minimización de radiaciones espurias y susceptibilidades y ofrecen muchas posibilidades de aplicación. Los estándares exactos usados en el entrenador son los *TX2-433-40-5V* y *RX2-433-40-5V* que ofrecen una frecuencia base de portadora *FM SAW* de 433 MHz con una

tasa de *bits* de 40 Kbps y un voltaje de alimentación típico de 5 voltios. Los certificados *CE* respectivos de cada uno se anexan al final del proyecto. A continuación se anexa el diagrama de bloques del *TX2*

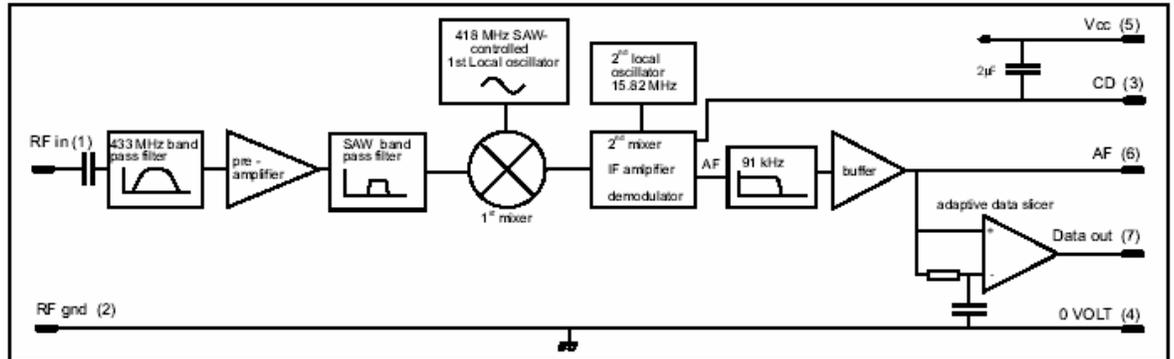
Figura 30. Diagrama de bloques del *TX2*



Ibíd., p. 2

Se pueden apreciar las distintas partes; el filtro pasabajos de 100KHz para reducir el espectro de entrada, el modulador simbolizado por un varactor, que es parte de la mayoría de los osciladores controlados por voltaje utilizados para modulación y un filtro pasabanda al final para evitar distorsión del espectro. En la siguiente figura se aprecia el diagrama de bloques del receptor *RX2* de *Radiometrix*

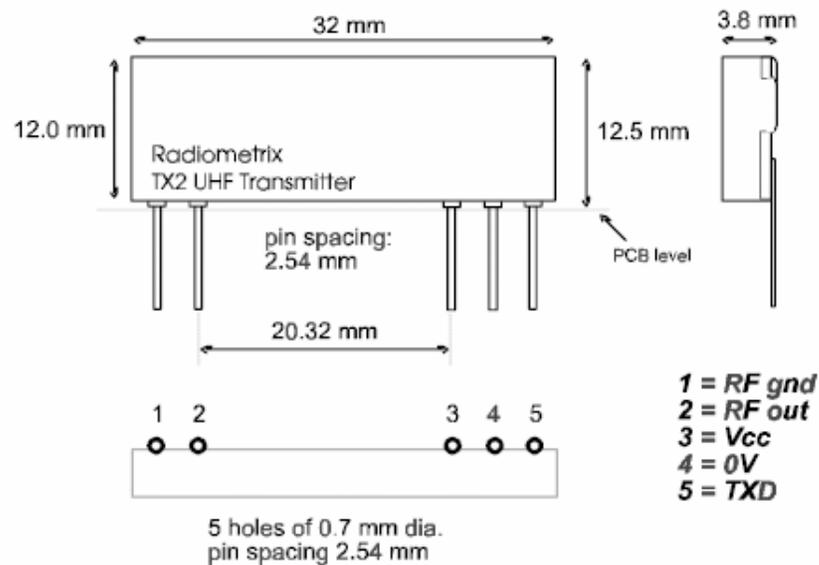
Figura 31. Diagrama de bloques del RX2



Ibíd., p. 3

Todo esto en un juego de módulos miniatura certificados muy prácticos y fáciles de montar. A continuación se presentan las características mecánicas de los mismos

Figura 32. Dimensiones físicas de los módulos RX2/TX2



Ibíd., p. 3

Tecnología: Módulos comerciales certificados de radio con tecnología contemporánea: Módulos *Radiométrix* (Inglaterra)

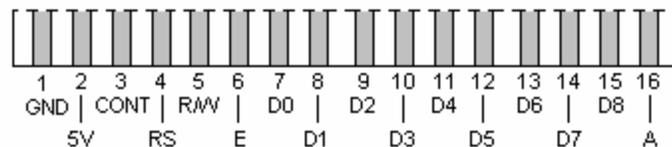
5.1.8 VISUALIZADOR: LCD OPTREX DMC-50968. Para que el usuario tenga una idea de las funciones que el equipo esta ejecutando y si esta recibiendo o no comandos, un visualizador de cristal liquido de 2 líneas por 16 caracteres se coloca al frente del entrenador conformando el panel visualizador del sistema. El modulo es clásico en montajes con microcontroladores, adaptada a las necesidades del entrenador y apto para tal efecto.

Figura 33. El visualizador *OPTREX DMC-50968*



PARÁMETROS DE DISEÑO: El *DMC-50968* es un visualizador *LCD* genérico, pues los retardos para el encendido y la distribución de terminales son bastante estándares y comunes a todos los *Display's* basados en el microcontrolador *Hitachi HD44780*. Como característica principal de este están una buena relación costo beneficio, que no se consigue hoja de información técnica de este en todo *internet* y que no hay respuesta alguna de soporte técnico de *OPTREX* para este modelo. Hay que conformarse entonces con el *DataSheet* genérico ofrecido por *OPTREX* y arriesgarse a probar que las entradas son como se muestran a continuación

Figura 34. Disposición de terminales en el bus del *OPTREX DMC-50968*



Los terminales del *OPTREX DMC-50968* están dispuestos de la forma descrita en el dibujo de la sección cortada de la tira de bus y son 16, donde la mayoría tiene 15, pues el terminal 16 es para conectar el ánodo de la luz de respaldo opcional a esta pantalla. Esta última es un arreglo de *LED's* alimentados por una fuente de 120 voltios, los cuales en casi todos los casos se obtienen de un circuito convertidor *DC a DC* conectado a la fuente de 5 voltios.

La configuración de los respectivos valores, retardos y comandos enviados a través del *PIC* y totalmente tratadas por software, permiten la muestra de los mensajes específicos del entrenador, algunos mostrados en la sección de botones del panel del entrenador y totalmente descritos en la *Guía de Usuario*

6. CONCLUSIONES

La investigación demuestra que trabajar con tecnologías de montaje superficial y altas escalas de integración si es posible; inclusive muchas de estas tienen técnicas tan sencillas de programación y acceso como las de un *PIC* microcontrolador; direccionamiento, temporización, control: el manejo de periféricos y temporización fue tan complicado como lo fue el de la pantalla *LCD*, pues como es bien sabido, también esta cuenta con su microcontrolador incorporado el cual requiere un manejo dedicado de tiempos de acceso y terminales de datos.

Los costos de este proyecto excedieron los topes originalmente fijados, suponiendo que se adquiriera un integrado adicional para efectuar todas las demodulaciones necesarias, la inversión para adquirirla hubiera agregado unos US\$500.00 al costo del proyecto, algo así como un millón y medio de pesos mas; entonces es implementable pero también costoso en etapa de investigación y desarrollo

7. OBSERVACIONES Y RECOMENDACIONES

Las siguientes observaciones se presentan como apoyo a posteriores trabajos de investigación en el área de aplicación e integración *ASIC*:

- Se pudo implementar el *ASIC MQFP* sobre una *PCB* doble faz de fibra de vidrio de alta calidad, necesaria por la alta densidad de pistas e interconexiones del integrado y sus dispositivos anexos
- El montaje se hizo inicialmente con una pistola *blower Hewlett Packard*, en emergencia inclusive con caudín de 20 vatios
- EL soporte técnico de *INTEL* estuvo disponible todo el tiempo; para el mercadeo, la comunicación constante y los precios competitivos por medios electrónicos de *AVNET* fue indispensable, pues la compra internacional de integrados se hace generalmente al mayor, vendiendo hasta millares como tope mínimo. *AVNET* ofrece la posibilidad de comerciar por unidades con precios aceptables
- El *PIC* ofreció toda la flexibilidad necesaria: los archivos se compilaron con *MPLAB 6*, el *PIC* se escribió con *Pony Programmer* y el hardware es del programador serial *Si Prog v2.2* disponible por Internet de múltiples fuentes

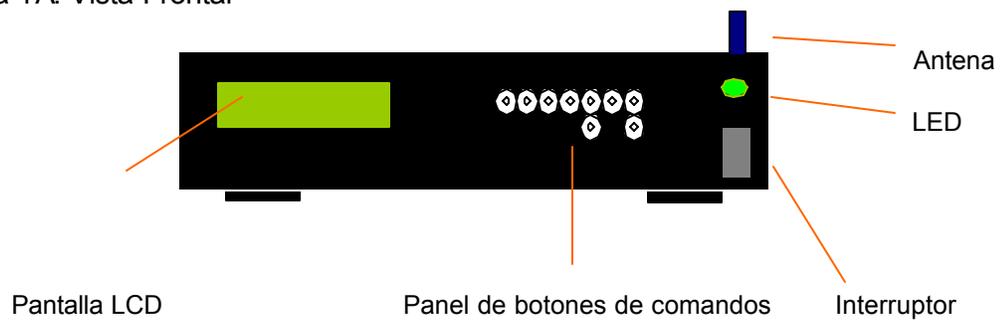
ANEXO A. CARACTERISTICAS MECANICAS DEL PROTOTIPO

El entrenador se emparedó en una caja metálica de color negro con dimensiones apropiadas, la C90. En Bogota se descubrió mercado de tecnologías y cajas importadas que ofrecían una solución dinámica e innovadora, pero se omitieron pues no se consiguieron fondos adicionales para realizar la innovación, Sin embargo, si la institución ofrece en el futuro apoyo para este tipo de detalles, serán recibidas con gusto. La apariencia externa del proyecto es la de una caja rectangular. Entre las ventajas de la caja metálica están:

- Blindaje total contra interferencias, con el debido aterrizaje a tierra del chasis
- Resistencia a los impactos y caídas. Un mal posible en equipo de laboratorio estudiantil

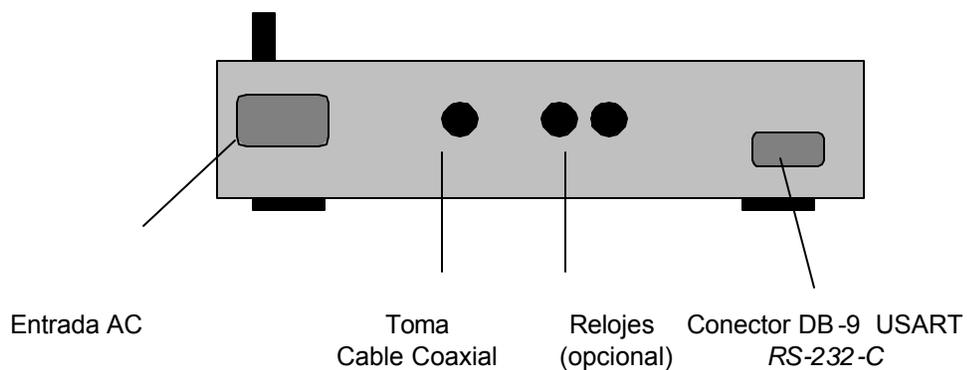
Las claras desventajas de la caja metálica se evidencian en la transmisión de radiofrecuencias. El apantallamiento ofrecido por el metal también introduce una atenuación al transmisor de radio

Figura 1A. Vista Frontal



En esta vista frontal el alumno accede directamente a el interruptor necesario para encender el entrenador, los botones de comandos, los cuales, cambian manualmente ciertas funciones y la pantalla *LCD* que debe dar los mensajes respectivos con cada comando, el *LED* de encendido que avisa cuando el prototipo esta encendido. Se puede observar también la pequeña antena para la transmisión por radio

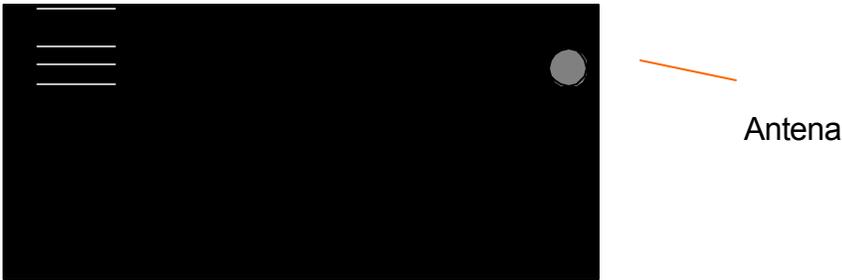
Figura 2A Vista Posterior



En la vista posterior se pueden observar los conectores necesarios para permitir el flujo de información tanto del entrenador hacia un PC remoto o hacia el visualizador usado, un osciloscopio, o inclusive la antena de radio que servirá para propagar la información a través del aire. La entrada de línea de corriente alterna, es el conector para al cable de

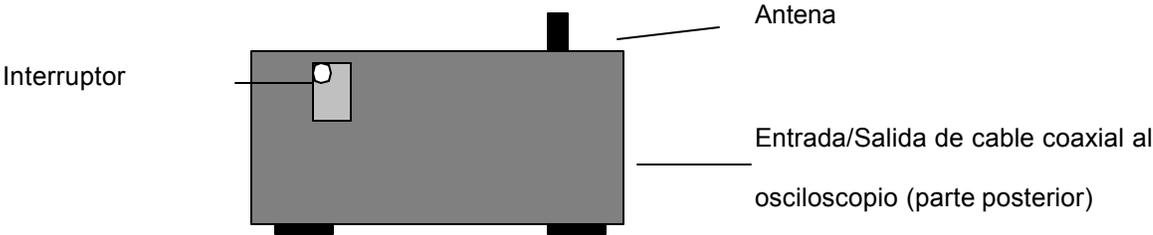
alimentación del equipo a 120V y 60Hz. Opcionalmente se propone una entrada externa para el reloj del *S1109CR*. Se pueden usar otros elementos para poder observar los diagramas fasoriales y de constelación respectivos a cada tipo de modulación, siendo ya estos equipos *adicionales y ajenos al proyecto*. Una vista superior del entrenador se tiene a continuación

Figura 3A. Vista Superior



El modulo receptor individual se vería así:

Figura 4A. Modulo Receptor Individual^{##}

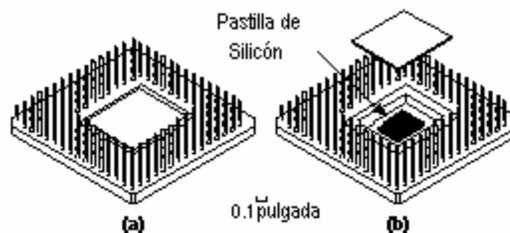


^{##} En este equipo, la parte demoduladora pasó a ser opcional después de los resultados de exhaustiva investigación y desarrollo

ANEXO B. ASIC. CIRCUITO INTEGRADO DE APLICACION ESPECIFICA

Un ASIC es un circuito integrado de aplicación específica. Antes de contestar la pregunta de que significa es bueno echarle un vistazo a la evolución del chip de silicón o circuito integrado (IC). En la figura se puede observar un encapsulado de circuito integrado (es una matriz de pines cuadriculados o más fielmente *Pin-Grid-Array* o *PGA*), está bocabajo, los pines atraviesan los orificios de la PCB. Comúnmente se le denomina *chip* al encapsulado, pero como se puede observar en la figura, el chip de silicón (a menudo llamado pastilla) está montado en la cavidad sellado sobre la cubierta. Un encapsulado *PGA* está usualmente hecho de un material cerámico pero son comunes los encapsulados plásticos.

Figura 1B. Un circuito integrado a) encapsulado *PGA* b) el chip de silicón o pastilla está bajo la cubierta del encapsulado



Application-Specific Integrated Circuits Copyright. Addison Wesley Longman, Inc., 1997.

Disponible en <http://www-ee.eng.hawaii.edu/~msmith/ASICs/HTML/Book/CH01>

Las dimensiones físicas de una pastilla de silicón varían desde unos pocos milímetros a más de una pulgada sobre una cara, pero a menudo se miden las dimensiones de un *IC* por el número de puertas lógicas o el número de transistores que este contiene. Como unidad de medida, *una puerta equivalente* corresponde a una puerta NAND de dos entradas (un circuito que ejecuta la función lógica $F = \overline{A \bullet B}$) A menudo se usa solo el término puerta en lugar de puerta equivalente cuando se miden las dimensiones de un chip, no se debe confundir esta con la compuerta de un transistor *FET* o *MOS*. Por ejemplo, un integrado con 100 mil puertas contiene el equivalente a 100.000 puertas *NAND* de dos entradas. La industria del silicón ha evolucionado desde los tempranos 1970 y madurado rápidamente desde entonces. La pequeña escala de integración *SLI* contenía de 1 a 10 puertas lógicas implicando unas pocas decenas de transistores. La mediana escala de integración o *MSI* incrementó el rango de lógica disponible a contadores y similares. La gran escala de integración o *LSI* encapsuló funciones lógicas más complejas, como el primer microprocesador. La era de la muy alta escala de integración o *VLSI* ofrece ahora microprocesadores de 64 *bits* completos con memoria cache y unidades aritméticas de punto flotante, más de un millón de transistores sobre una sola pieza de silicón. Las primeras tecnologías usadas en los circuitos integrados empleaban transistores bipolares y eran *TTL* o *ECL*. A pesar de que fue anterior a la bipolar, la tecnología *MOS* era difícil de fabricar debido a problemas con la interconexión de óxido. Esos problemas fueron resueltos, hoy en día los *CMOS* necesitan poco enmascaramiento, son más

compactos y consumen menos energía que los integrados bipolares equivalentes, además los integrados a base de *CMOS* son más baratos que los conformados por bipolares. A inicios de los 80's las puertas de aluminio fueron remplazadas por puertas de polisilicón pero el nombre *MOS* permaneció. La implementación de polisilicón permitió la creación de dos tipos de transistores: *MOS de canal n* y *MOS de canal p* en un solo integrado, y consecuentemente la tecnología *MOS* complementario o *CMOS*. Además el *CMOS* sobre el anterior *NMOS* facilitó el proceso de fabricación haciéndolos mas pequeños y redujo el consumo de energía. Hay cuatro transistores *CMOS* por cada puerta *NAND* de dos entradas en un integrado (también en una *NOR* de dos entradas) de modo que para convertir de puertas a transistores nada mas hay que multiplicar el número de puertas por 4. También se puede medir un *IC* por la más pequeña *dimensión característica* (apenas la mitad de la longitud del transistor mas pequeño) impresa sobre el integrado.

Las dimensiones de los transistores se miden en micrones (1 micrón es la millonésima parte de un metro), Actualmente se usan transistores *BiCMOS* (combinación de bipolar con *CMOS*) los cuales permiten trabajar con voltajes mas elevados. Aun así el volumen de la producción de *CMOS* es el más dominante obre las otras tecnologías pues es, ante todo, más económico de producir. Con el nacimiento de las tecnologías *VLSI* en los 80's los ingenieros comenzaron a hacer avances significativos de diseño que fueron personalizados de acuerdo a las

necesidades. Sin embargo, hay situaciones en las que no es favorable personalizar todos los *IC*'s en un sistema microelectrónico. La primera conferencia respecto a este segmento de la industria que crecía tan rápidamente fue la *IEEE Custom Integrated Circuits Conference (CICC)*, las posteriores a esta conferencia anual se referencian al diseño de integrados personalizados. Actualmente se tiene la *IEEE International ASIC Conference*, la cual sigue los avances en tecnología *ASIC* separadamente de otros tipos de integrados hechos a la medida. Como regla general para identificar un *ASIC*, si se puede conseguir en una guía de referencia este probablemente no sea un *ASIC*, aunque hay algunas excepciones.

TIPOS DE *ASIC*'S. Los circuitos integrados son hechos sobre una delgada oblea de silicón, con cada capa conteniendo cientos de pastillas. Los transistores y las conexiones están hechos de algunas capas (usualmente entre 10 y 15 capas distintas) construidas una encima de otra. Cada mascara de capa sucesiva tiene una estructura que se define usando una mascara similar a una superficie fotográfica de vidrio. La primera media docena de capas definen los transistores. La última docena de capas define las conexiones metálicas entre transistores. Un *IC* totalmente personalizado incluye algunas celdas hechas a la medida y todas las mascaras de capas personalizadas. Un ejemplo de integrado totalmente hecho a la medida o personalizado es un microprocesador. El tiempo de manufactura (no incluye el tiempo de diseño) es de típicamente 8 semanas. Estos integrados son proyectados para una aplicación específica, de modo que algunos de ellos pueden

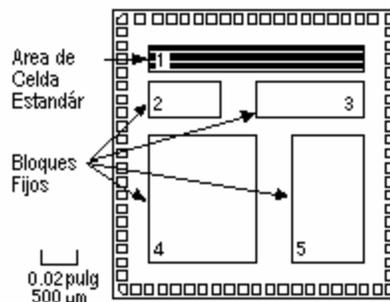
llamarse *totalmente personalizados*. Los integrados *ASIC* semipersonalizados tienen celdas prediseñadas y algunas mascararas de capas hechas a la medida. Usando celdas prediseñadas de una *librería de celdas* se facilita la labor del diseñador. Hay dos tipos de integrados *ASIC* semipersonalizados a cubrir: *ASIC's* basados en celdas estándar y *ASIC's* basados en arreglos de puertas. Seguidamente se describen los *ASIC's* programables, para los cuales todas las celdas lógicas son prediseñadas y ninguna de las mascararas de capas son personalizadas. Existen dos tipos de *ASIC's* programables los dispositivos lógicos programables o *PLD's* y los miembros mas recientes de la familia, los arreglos de puertas de campo programable o *FPGA*.

ASIC'S TOTALMENTE HECHOS A LA MEDIDA. En un *ASIC* totalmente personalizado o hecho a la medida el ingeniero diseña alguna o todas las celdas lógicas, circuitos, o disposiciones específicamente para un *ASIC* dado. Esto significa que no usa celdas prediseñadas porque no son lo suficientemente rápidas, consumen mucha corriente o porque no existen en una biblioteca de celdas predefinida. Por esta última razón puede ser necesario hacer un diseño totalmente nuevo

ASIC'S BASADOS EN CELDAS ESTANDAR. Un *ASIC* basado en celdas (*cell-based IC*, o *CBIC*) usa celdas lógicas prediseñadas (por ejemplo; puertas *AND*, puertas *OR*, multiplexadores y *flip-flop's*) conocidas como celdas estándar. El

termino *CBIC* puede aplicarse a cualquier *IC* que usa celdas pero es más aceptado que el *ASIC* basado en celdas o *CBIC* signifique *ASIC*'s basados en celdas estándar. Las áreas de celdas estándar (también llamadas bloques flexibles) en un *CBIC* están construidas de filas de celdas estándar — como un muro hecho de ladrillos. Las celdas de área estándar se pueden usar en combinación con celdas prediseñadas más grandes, quizás microcontroladores o hasta microprocesadores, conocidas como megaceldas. Las megaceldas también son llamadas megafunciones, bloques totalmente personalizados, macros de nivel de sistema (*SLM*'s), bloques fijos, núcleos, o bloques funcionales estándar (*FSB*'s).

Figura 2B. Pastilla de un *ASIC* basado en celdas (*CBIC*) con un área de celda estándar (un bloque flexible) conjuntamente con cuatro bloques fijos. El bloque fijo contiene filas de celdas estándar. Este es el aspecto de la pastilla de la sección b) en la figura anterior bajo un microscopio de baja potencia. Los pequeños cuadrados alrededor de los bordes son *bounding pads*, o pequeñas trazas de cobre sobre el sustrato a la cual se unirán los conductores



Ibíd.

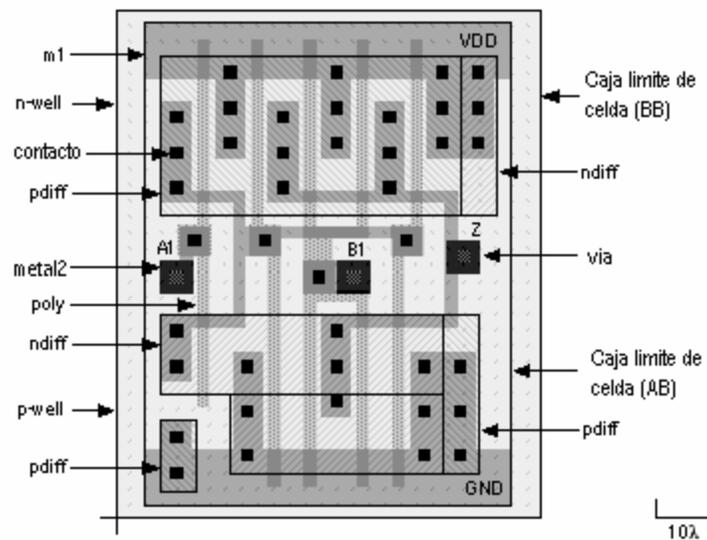
La principal ventaja de usar *CBIC*'s es que los diseñadores ahorran tiempo, dinero, y se reduce el riesgo al usar una librería de celda estándar prediseñada, previamente probada y caracterizada. Adicionalmente cada celda puede ser optimizada por separado. La figura describe un *CBIC*, los principales beneficios de usar este tipo de *ASIC* son

- ◆ Todas las mascararas de capa son hechas a la medida – transistores e interconexiones
- ◆ Los bloques hechos a la medida se pueden incrustar
- ◆ El tiempo de manufactura es de alrededor de 8 semanas

Las células estándar se diseñan de forma que se acomoden tal como ladrillos en un muro. La figura describe un ejemplo de una celda estándar simple (es simple en el sentido de que no esta maximizada en densidad — pero si es ideal para mostrar su construcción interna) Los buses de potencia y tierra (*VDD* y *GND* o *VSS*) corren horizontalmente sobre las líneas de metal dentro de las celdas.

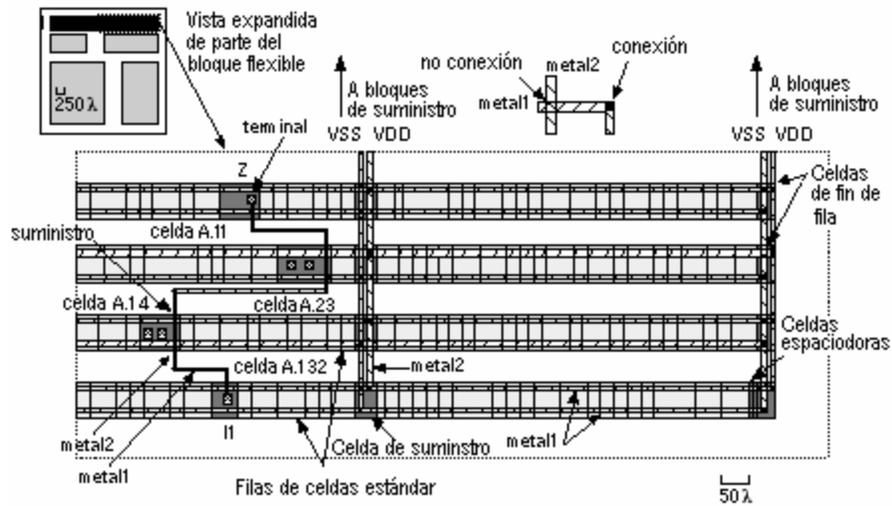
Figura 3B. Diseño simplificado de una celda estándar. Esta podría tener aproximadamente 25 micrones sobre un *ASIC* con $\lambda(\text{lambd}) = 0.25$ micrones (un micrón es igual a 10^{-6} m). Las celdas estándar se apilan como ladrillos en un muro. Los limites del estuche (AB) definen los “bordes” del ladrillo. La diferencia de la caja límite BB menos la AB es el área de superposición entre los ladrillos, Las fuentes de poder (*VDD* y *GND*) corren horizontalmente dentro de una celda estándar sobre una capa de metal que se sitúa

alrededor de las capas de transistores. Cada patrón diferentemente sombreado y etiquetado representa una capa distinta. La celda estándar tiene conectores en el centro (los tres cuadrados etiquetados A1, B1 y Z) que le permiten conectarse a otras. El diseño se dibujó usando *ROSE*. Un editor simbólico de diseño creado por *Rockwell* y *Compass* y entonces importado dentro del proyecto de investigación de *Tanner L-Edit*



Ibíd.

Figura 4B. Asignación de rutas del *CBIC* descrito en la figura 39. El uso de celdas estándar conformadas regularmente desde una librería tales como la de la figura anterior, permite a los *ASIC* ser diseñados automáticamente. Este *ASIC* usa dos capas de metal separadas e interconectadas (metal1 y metal2) desplazándose en ángulos rectos entre si (como pistas en un circuito impreso). Las interconexiones entre celdas lógicas usan espacios (llamados canales) entre filas de celdas. Los *ASIC's* pueden tener tres o mas capas de metal permitiendo a las filas de celdas tocarse con la interconexión que pasa sobre estas



Ibíd.

El uso de células estándar puede no hacer el diseño de ASIC más eficiente para lógicas que operan con señales múltiples a través de un bus de datos — un *datapath* (*DP*) —. Algunas compañías de librerías ASIC proporcionan un compilador de *datapath* que automáticamente genera la *trayectoria de datos lógicos* (*logic datapath*). Una librería de trayectoria de datos generalmente contiene celdas tales como sumadores, sustractores, multiplicadores, y unidades aritméticas y lógicas simples (*ALU's*). Al conectar celdas de trayectoria de datos para formar una trayectoria de datos, usualmente (pero no siempre) se obtienen diseños más rápidos y densos que al usar celdas estándar o arreglos de puertas. Las celdas estándar y las librerías de arreglos de puertas pueden contener cientos de celdas lógicas diferentes, incluyendo funciones combinacionales (puertas *NAND*, *NOR*, *AND*, *OR*) con entradas múltiples, así como candados y *flip-flop's* con combinaciones diferentes de *reset*, *preset* y opciones de temporización. La

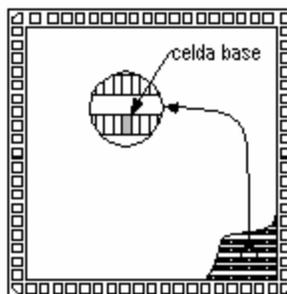
compañía de librería *ASIC* le suministra a los diseñadores un *data book* u hoja de información técnica como artículo o forma electrónica con todas las descripciones funcionales e información de temporización para cada elemento de la librería.

ASIC'S BASADOS EN ARREGLOS DE PUERTAS. En un arreglo de puertas (algunas veces abreviado *GA* por sus siglas en ingles) o *ASIC* basado en arreglo de puertas, los transistores están predefinidos en la oblea de silicón. El patrón predefinido de transistores sobre un arreglo de puertas es el *arreglo base* y elemento mas pequeño producido para hacer el arreglo base (como baldosas en el piso) es la *celda base* (algunas veces llamada celda primitiva). El diseñador solo define las mejores capas de metal, los cuales definen la interconexión entre transistores, usando mascararas hechas a la medida. Para definir este arreglo de puertas de otro modo a menudo se le llama *Masked Gate Array (MGA)* o arreglo de puertas enmascarado. El diseñador escoge de una librería de arreglo de puertas de entre variadas celdas lógicas prediseñadas y precharacterizadas. Las celdas lógicas en una librería de arreglo de puertas son frecuentemente llamadas *macros*. La razón de esto es que el diseño de puertas base es el mismo para cada celda lógica y solo se personaliza la interconexión (dentro de las celdas y entre estas) a fin de que exista analogía entre macros de arreglo de puertas y una macro de software. Se pueden completar los pasos de difusión que forman los transistores y entonces almacenar las obleas (por esta razón algunas veces se le dice al arreglo de puertas *arreglo predifundido*). Cuando por un *MGA* solo son exclusivas las

interconexiones de metal, se pueden usar las obleas almacenadas para diferentes usuarios de acuerdo a sus necesidades. Usando obleas prefabricadas hasta los pasos de metalización se reduce el tiempo necesario para hacer un *MGA*, el punto de cambio, desde unos pocos días o a lo sumo varias semanas. Los costos de todos los pasos iniciales de fabricación de un *MGA* son distribuidos entre cada usuario y esto reduce el costo de un *MGA* comparado con un diseño totalmente hecho a la medida o un diseño *ASIC* de celda estándar. Existen actualmente los siguientes diferentes tipos de *ASIC's MGA* o *ASIC's* basados en arreglos de puertas:

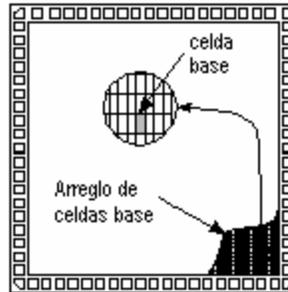
- Arreglos de puertas acanalados
- Arreglos de puertas sin canal
- Arreglos de puertas estructurados

Figura 5B. Pastilla de un arreglo de puertas acanalado. Los espacios entre filas de la celda base están repartidos para interconexión



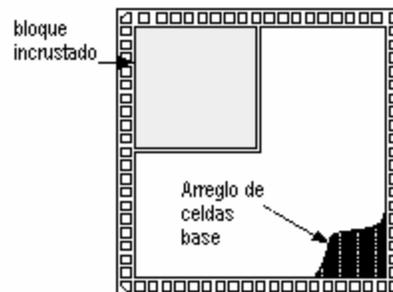
Ibíd.

Figura 6B. Pastilla de un arreglo de puertas sin canal o arreglo de mar de puertas (SOG). El área de la pastilla está completamente rellena con un arreglo de celdas base (el arreglo base)



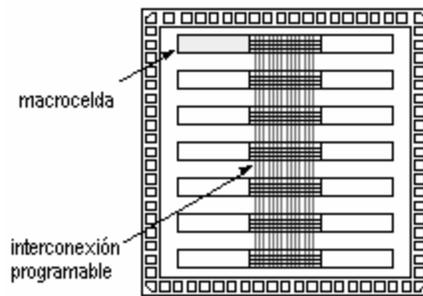
Ibíd.

Figura 7B. Una pastilla de arreglo de puertas incrustadas o estructuradas describiendo un bloque incrustado en el borde superior izquierdo (por ejemplo, una memoria de acceso aleatorio estática o *RAM* estática). El resto de la pastilla esta rellena con un arreglo de celdas base.



Ibíd.

Figura 8B. Pastilla de un dispositivo lógico programable (PLD). Las macroceldas consisten de lógica de arreglo programable seguida por un *flip-flop* o candado. Las macroceldas se conectan usando un gran bloque de interconexión programable



Ibíd.

ANEXO C. TUTORIAL *UART*

El Controlador Universal Asíncrono Receptor/Transmisor (*UART* por sus siglas inglesas) es el componente clave del subsistema de comunicaciones serie de un computador. El *UART* toma *bytes* de datos y transmite los *bits* individuales en forma secuencial. En el destino, un segundo *UART* reensambla los *bits* en *bytes* completos. La transmisión de comunicación serie es comúnmente usada con *modems* y para comunicaciones no conectadas a una red; entre computadores, terminales y otros dispositivos. Existen dos formas primarias de transmisión serie: Síncrona y Asíncrona. Dependiendo de los modos soportados por el hardware, el nombre del subsistema de comunicación usualmente incluirá una *A* si este soporta comunicaciones Asíncronas, y una *S* si soporta comunicaciones síncronas.

Ambas formas se describen a continuación. Algunos acrónimos comunes de origen ingles son:

- *UART Universal Asynchronous Receiver/Transmitter*
- *USART Universal Synchronous-Asynchronous Receiver/Transmitter*

TRANSMISION SERIAL SINCRONA. Las comunicaciones de transmisión serie síncronas requieren que el emisor y el receptor compartan un reloj uno con otro, o que el emisor proporcione un *strobe* u otra señal de temporización de modo que el receptor sepa cuando “leer” el siguiente *bit* de los datos. En la mayoría de las formas de comunicación serial síncrona, si no existen datos disponibles en un determinado instante para transmitir, se debe enviar un carácter de relleno en su lugar, de modo que siempre se esté transmitiendo información. La comunicación síncrona es usualmente mas eficiente porque solo se transmiten *bits* de datos entre emisor y receptor y puede ser mas costosa si se requiere cableado y circuitos extra para compartir una señal de reloj entre el emisor y el receptor. Una forma de transmisión síncrona es usada con impresoras y discos fijos, componentes en los que la información es enviada sobre un *set* de de conductores mientras un reloj o *strobe* se envía sobre un conductor diferente. Las impresoras y los discos fijos son dispositivos normalmente no serie, porque la mayoría de los estándares de las interfaces de discos fijos emiten una palabra de datos entera por cada señal de reloj o *strobe* a través del uso de un alambre separado para cada *bit* de la palabra. En la industria de los *PC*'s, estos se conocen como dispositivos paralelos. El hardware estándar de comunicaciones series en el *PC* no soporta operaciones síncronas. Este modo se describe solo para propósitos de comparación

TRANSMISION SERIE ASINCRONA. Las transmisiones asíncronas permiten que los datos sean transmitidos sin que el emisor tenga que enviar una señal de reloj al

receptor. En lugar de esto el emisor y el receptor deben estar de acuerdo, por anticipado, en cuanto a parámetros de temporización; además se añaden *bits* especiales a cada palabra los cuales se usan para sincronizar las unidades emisoras y receptoras. Cuando una palabra es dada al *UART* para transmisiones asíncronas un *bit* llamado "*bit* de inicio" (*Start bit*) se agrega al inicio de cada palabra que va ser transmitida. El *bit* de inicio se usa para alertar al receptor que una palabra de datos esta a punto de ser enviada y para forzar al reloj en el receptor a sincronizarse con el reloj del transmisor. Esos dos relojes deben de ser lo suficientemente precisos para no tener deriva de frecuencia por mas de un 10% durante la transmisión de los *bits* remanentes en la palabra (esta necesidad se fijó en los días de los teletipos mecánicos y se logra con facilidad en los equipos modernos).

Después del *bit* de inicio, se envían los *bits* individuales de la palabra de datos, enviando primero el *bit* menos significativo (*LSB*). Cada *bit* en la transmisión es enviado exactamente por el mismo monto de tiempo que cualquiera de los otros, y el receptor inspecciona el alambre a aproximadamente la mitad periodo asignado a cada *bit* para determinar si este *bit* es un 1 o un 0. Por ejemplo, si toma 2 segundos enviar cada *bit*, el receptor examinará la señal para ver si es un 1 o un 0 después de pasado un segundo, entonces este esperará 2 segundos para luego examinar el valor del siguiente *bit* y así sucesivamente. El emisor no sabe que el receptor ha inspeccionado el valor del *bit*. El emisor solo sabe que el reloj dice que

hay que iniciar la transmisión del siguiente *bit* de la palabra. Cuando la totalidad de la palabra de datos ha sido enviada, el transmisor puede agregar un *Bit de Paridad* que este mismo genera. El *bit* de paridad puede ser usado por el receptor para efectuar un chequeo de error sencillo. Entonces el transmisor genera el menos un *Bit de Parada (Stop Bit)*. Cuando el receptor ha tomado todos los *bits* en la palabra de datos, puede chequear los bis de paridad (ambos emisor y receptor deben estar de acuerdo en cuanto a como usar los *Bits de Paridad*), y entonces el receptor inspecciona buscando un *Bit de Parada*. Si el *Bit de Parada* no aparece como se espera entonces el *UART* considera que la palabra de datos esta distorsionada y reportará un error de sincronización de trama a el procesador *host* de donde se lee la palabra. La causa más usual de errores de sincronización de trama es que los relojes del emisor y el receptor no corren a la misma velocidad, o que la señal es interrumpida. Independientemente de cómo los datos sean recibidos, correctamente o no, el *UART* automáticamente descarta los *bits* de inicio, paridad y parada. Si el emisor y el receptor se configuran de forma idéntica, esos *bits* no pasan al *host*. Si hay otra palabra lista para la transmisión, el *Bit de Inicio* de la nueva palabra puede enviarse inmediatamente se reciba el *Bit de Parada* de la palabra anterior. Como los datos asíncronos son “autosincronizados”, al no existir datos a transmitir, la línea de transmisión se puede poner inactiva.

OTRAS FUNCIONES *UART*. Adicionalmente al trabajo de convertir datos de paralelo a serie para transmisión y de serie a paralelo en la recepción, el *UART*

usualmente proporciona circuitos adicionales para señales que pueden ser posteriormente usadas para indicar el estado del medio de transmisión y regular el flujo de datos en el evento de que el equipo remoto no este preparado para aceptar mas datos, Por ejemplo cuando el dispositivo conectado al *UART* es un *modem*, el *modem* puede detectar la presencia de una portadora en la línea telefónica mientras que el PC es capaz de darle instrucciones al *modem* para resetearse o no tomar llamadas, por la subida o bajada de una o mas de estas señales extras. La función de cada una de esas señales adicionales esta definida en el estándar EIA *RS-232-C*

LOS ESTANDARES *RS-232-C* Y V.24. En la mayoría de los sistemas de computadoras, el *UART* esta conectado a la circuitería que genera señales que cumplen con la especificación *EIA RS-232-C*. Existe también un estándar de la *CCITT* llamado V.24 que refleja las especificaciones incluidas en *RS-232-C*

ASIGNACIONES DE BIT DEL *RS-232-C* (MARCAS Y ESPACIOS). En *RS-232-C* un valor de 1 es llamado marca y un valor de 0 es llamado espacio. Cuando una línea de comunicación está inactiva, se dice que la línea esta “marcando” o transmitiendo valores continuos de 1. El *Bit de Inicio* siempre tiene un valor de 0 (un espacio). El *Bit de Parada* siempre tiene un valor de 1 (una marca). Esto significa que siempre habrá una transición de una marca (1) a un espacio (0) en la línea, al inicio de cada palabra, cuando se transmiten múltiples palabras de

seguido. Esto permite que el emisor y el receptor puedan volver a sincronizar sus relojes independientemente del contenido de los *bits* de datos que están siendo transmitidos. El tiempo de inactividad entre los *bits* de inicio y de parada no tienen que ser un múltiplo exacto (incluyendo el cero) de la tasa de *bits* del enlace de comunicaciones, pero la mayoría de los *UART* son diseñados de esta forma por su sencillez. En *RS-232-C* la señal de marca (un 1) se representa con un voltaje de entre -2 voltios y -12 voltios CC y una señal de espacio (un 0) se representa por un voltaje de entre 0 y +12 voltios CC, el receptor se supone permite alguna perdidas de voltaje en cables largos. Algunos transmisores en dispositivos de baja potencia (como computadores portátiles) algunas veces usan solo +5 voltios CC y -5 voltios CC^{§§}, pero esos valores son aun aceptables para cables de corta longitud

RS-232-C SEÑAL DE PAUSA. *RS-232-C* también especifica una señal llamada una pausa, la cual es causada por el envío continuo de valores de espacio (sin *bits* de inicio o parada). Cuando no hay electricidad presente en el circuito de datos, se considera que la línea esta enviando pausa. La señal de pausa debe tener una duración mayor que el tiempo que toma enviar un *byte* completo más los *bits* de inicio, parada y paridad. La mayoría de los *UART*'s pueden distinguir entre un error de sincronización de trama y una pausa, pero si el *UART* no puede hacer esto, se puede usar a detección de errores de trama para identificarlas. En los días de los teletipos, cuando numerosas impresoras alrededor de los Estados unidos se

^{§§} El *MAX232* convierte los voltajes *TTL* del PC (0 a +5V) a voltajes bipolares de +10V y -10V para . cumplir con este protocolo

interconectaron en serie (tales como los servicios de noticias), cualquier unidad podía causar una pausa, abriendo temporalmente el circuito entero de modo que no fluyera corriente. Esto se usó para permitir que una localidad con noticias urgentes pausara alguna otra que estuviera transmitiendo en ese momento. En los sistemas modernos existen dos tipos de señales de pausa. Si la pausa dura más de 1,6 segundos, significa que hay una *Ruptura de Datos* y esto limita al computador remoto a responder a esta señal. Algunas veces esta forma de pausa de datos se usa como señal de atención o interrupción y algunas veces se acepta como forma sustituta para el carácter *ASIC CONTROL—C*. Las marcas y espacios son también equivalentes a “agujeros” y “no agujeros” en sistemas de cinta de papel.

RS-232-C. DISPOSITIVOS DCE Y DTE. El *RS-232-C* define dos tipos de equipos: El equipo terminal de datos (*DTE*) y el equipo portador de datos (*DCE*). Usualmente el dispositivo *DTE* es el terminal (o computador) y el *DCE* es el *modem*. A través de la línea telefónica y al otro lado y en el otro extremo de una conversación, el *modem* es también un dispositivo *DCE*. El dispositivo *DCE* recibe señales sobre los terminales en que el dispositivo *DTE* transmite, y viceversa. Cuando dos dispositivos que son ambos *DTE* o *DCE* se conectan entre sí sin un *modem* o medio traductor similar entre ellos, se debe usar un *modem* nulo. El *modem* nulo redispone eléctricamente el cableado de modo que la salida del transmisor se conecten eléctricamente a la entrada del receptor del otro dispositivo

y viceversa. Traducciones similares se realizan sobre todas las señales de control de modo que cada dispositivo vea que se está comunicando con señales *DCE* (o *DTE*) de otro dispositivo. El número de señales generadas por los dispositivos *DCE* y *DTE* no son simétricas. El dispositivo *DTE* genera menos señales para el dispositivo *DCE* que las que el *DTE* recibe del *DCE*.

ASIGNACIONES DE PIN DEL RS-232-C. La especificación *EIA RS-232-C* (y la equivalente *ITU V.24*) exige un conector de 25 pines (usualmente un *DB25*) y define la aplicación de la mayoría de los pines en ese conector. En el computador personal *IBM* y sistemas similares se ofrecen un subconjunto de señales *RS-232-C* a través de un conector de 9 pines (*DB9*). Las señales que no son incluidas en el conector del *PC* tratan de mantener una operación síncrona y este modo de transmisión no es soportado por el *UART* que *IBM* seleccionó para su uso en el *PC IBM*. Dependiendo del fabricante del computador un *DB25* un *DB9* o ambos tipos de conectores se pueden usar para las comunicaciones *RS-232-C*. El *PC IBM* también usa un *DB25* para la interfase paralela de la impresora, lo cual causa confusión. Seguidamente se muestra una tabla de las asignaciones de señales *RS-232-C* en los conectores *DB25* y *DB9*.

Tabla 1C. Asignaciones de señales en el estándar RS-232-C

Pin en el DB25 RS-232-C	Pin en el DB9 del IBM PC	Símbolo en el circuito EIA	Símbolo en el circuito CCITT	Nombre Común	Fuente de señal	Descripción
1	-	AA	101	PG/FG	-	Trama/tierra de protección
2	3	BA	103	TD	DTE	Transmitir datos
3	2	BB	104	RD	DCE	Datos recibidos
4	7	CA	105	RTS	DTE	Petición de permiso para transmitir
5	8	CB	106	CTS	DCE	Permitido transmitir
6	6	CC	107	DSR	DCE	Preparado para ser transmitido
7	5	AV	102	SG/GND	-	Tierra de señal
8	1	CF	109	DCD/SD	DCE	Detector de portadora
9	-	-	-	-	-	Reservado para prueba
10	-	-	-	-	-	Reservado

						para prueba
11	-	-	-	-	-	Reservado para prueba
12	-	CI	122	SRLSD	DCE	Señal secundaria de línea del receptor
13	-	SCB	121	SCTS	DCE	Permitido transmitir secundario
14	-	SBA	118	STD	DTE	Transmitir datos secundario
15	-	DB	114	TSET	DCE	Temporizador elemento transmisor de señal
16	-	SBB	119	SRD	DCE	Petición de permiso para transmitir secundaria
17	-	DD	115	RSET	DCE	Elemento temporizador señal receptor
18	-	-	141	LOOP	DTE	Lazo cerrado

						local
19	-	SCA	120	SRS	DTE	Petición de permiso para transmitir secundaria
20	4	CD	108.2	DTR	DTE	Terminal de datos listo
21	-	-	-	RDL	DTE	Lazo cerrado digital remoto
22	9	CE	125	RI	DCE	Indicador de llamada telefónica entrante
23	-	CH	111	DSRS	DTE	Selector de tasa de señal de datos
24	-	DA	113	TSET	DTE	Elemento temporizador señal de transmisión
25	-	-	142	-	DCE	Modo de Prueba

Frank Durda IV. Serial and *UART* Tutorial [online]. FreeBSD.org, 1996. [Citado el 3 de Marzo de 2003]. Disponible en http://www.freebsd.org/doc/en_US.ISO8859-1/articles/serial-uart/, de forma gratuita

BITS, BAUDIOS Y SIMBOLOS. Un baudio es la medida de la velocidad de transmisión en una comunicación asíncrona. Debido a los avances en las tecnologías de comunicación por *modem*, este término es frecuentemente maltratado cuando se describen las tasas de datos en los dispositivos más modernos. Tradicionalmente una tasa de datos representa el número de *bits* que se están transmitiendo en un instante sobre el medio, no el monto de datos que se están moviendo en ese instante de un dispositivo *DTE* a otro. El conteo de baudio incluye los *bits* del encabezado (*overhead*) Inicio, Parada y Paridad que son generados por el *UART* emisor y removidos por el *UART* receptor. Esto significa que la palabra de datos de 7 *bits* realmente toma 10 *bits* para ser completamente transmitida. Consecuentemente, un *modem* capaz de mover 300 *bits* por segundo de un lugar a otro puede mover solamente 30 palabras de 7 *bits* si se usa paridad y están presentes los *bits* de inicio y parada. Si se usan palabras de datos de 8 *bits* y también se usan *bits* de paridad, la tasa de datos salta a 27.27 palabras por segundo, porque ahora se toman 11 *bits* para enviar palabras de 8 *bits* y el *modem* todavía sigue enviando 300 *bits* por segundo. La formula para convertir *bytes* por segundo en tasa de baudio y viceversa fue sencilla hasta que surgieron los *modem's* con corrección de errores. Estos *modem's* reciben el flujo de *bits* serie del *UART* en el computador *host* (aun cuando se usan *modem's* internos los datos siguen serializados) y convierten los *bits* otra vez en *bytes*. Esos *bytes* son entonces combinados dentro de paquetes y enviados sobre la línea telefónica usando un método síncrono de transmisión. Esto significa que los *bits* de parada,

paridad e inicio agregados por el *DTE* (el computador) son removidos por el modem antes de la transmisión. Cuando esos *bytes* son recibidos por el modem remoto, este añade los *bits* de inicio, parada y paridad a las palabras, convirtiéndolas a un formato serial y entonces enviándolas al *UART* receptor en el computador remoto, quien quita los *bits* de inicio, parada y paridad. La razón de esas conversiones extra es que estos dos *modem's* pueden efectuar *corrección de errores*, lo cual significa que el modem receptor es capaz de solicitar al modem transmisor reenviar un bloque de datos que no fue recibido con la suma de chequeo (*checksum*) correcta. Este chequeo es manejado por los *modem's* y los dispositivos *DTE* usualmente desconocen la ocurrencia del proceso. Por la remoción de los *bits* de inicio, parada y paridad los *bits* de datos adicionales que los dos *modem's* deben compartir entre ellos para efectuar corrección de errores están en la mayoría de los casos ocultos a la tasa de transmisión efectiva vista por el equipo *DTE* emisor y receptor. Por ejemplo, si un modem envía diez palabras de 7 *bits* a otro modem sin incluir los *bits* de inicio, parada y paridad, el modem emisor podrá agregar 30 *bits* de su propia información que el modem receptor puede usar para corrección de errores, sin impactar la velocidad de transmisión real de los datos. El uso del término baudio es mucho más confuso para los *modem's* que efectúan compresión. Una palabra sencilla de 8 *bits* pasada a través de una línea telefónica puede representar una docena de palabras que fueron transmitidas por el modem emisor. El modem receptor volverá a expandir los datos a su contenido original y los pasará al *DTE* receptor. Los *modem's* modernos también poseen

buffer's que permiten que la tasa de *bits* moviéndose a través de la línea telefónica (*DCE a DCE*) tener una velocidad diferente que la de los *bits* que se mueven a través del *DTE* y el *DCE* sobre ambos finales de la conversación. Normalmente la velocidad entre el *DTE* y el *DCE* es mas alta que la de *DCE a DCE* por el uso de compresión en los *modem's*. Debido al número de *bits* necesarios para describir un *byte* cambiado durante la travesía entre las dos máquinas mas las diferentes velocidades de *bits* por segundo que se encuentran presentes en los enlaces *DTE-DCE* y *DCE-DCE*, el uso del termino baudio para describir las velocidades de comunicación en conjunto causan confusiones y puede dar una idea falsa de la verdadera velocidad de transmisión. Entonces *bits* por segundo (*bps*) es el termino mas correcto a usar para describir la tasa de transmisión vista desde la interfase *DCE a DCE* y los términos baudio o *bit* por segundo son aceptables cuando la conexión esta hecha entre dos sistemas con una conexión cableada, o si el *modem* en uso no ejecuta compresión o corrección de errores. Los modernos *modem's* de alta velocidad (2400, 9600, 14,400 y 19,200 bps) en realidad siguen operando a o por debajo de los 2400 baudios, o mas precisamente, 2400 símbolos por segundo. Un *modem* de alta velocidad es capaz de codificar mas *bits* de datos dentro de cada símbolo usando una técnica llamada Relleno de Constelación (*Constellation Stuffing*), la cual se usa porque la tasa de *bits* por segundo efectiva es más alta, pero el *modem* continúa operando dentro del ancho de banda limitado que proporciona el sistema telefónico. Los *modem's* que operan a velocidades de 28,800 y más tienen tasas de símbolo variables, pero la técnica es la misma.

EL UART DEL COMPUTADOR PERSONAL IBM. Iniciando con el computador personal o *PC*, *IBM* seleccionó el *UART* de la *National Semiconductor INS8250* para su uso en el adaptador Serie/Paralelo de este. Las subsecuentes generaciones de computadores *IBM* compatibles y otros proveedores continuaron usando el *INS8250* o versiones mejoradas de la familia de *UART*'s de la *National Semiconductor*. Existen distintas versiones del *INS8250* (*INS8250*, *INS8250B*, *INS8250A*, *INS82C50A*, *NS16450*, *NS16C450*, *NS16550*, *NS16550A*, *PC16550D*) las cuales no difieren mucho del original. Al pasar de los años el *8250*, *8250A*, *16450* y el *16550* han sido licenciados o copiados por otros proveedores de chips; en el caso del *8250*, *8250A* y el *16450*, el circuito exacto (la megacelda) fue licenciada a algunos proveedores entre los que están *Western Digital* e *Intel*.

ANEXO D. LA CODIFICACION REED SOLOMON

Los códigos *Reed-Solomon* son códigos de corrección de errores basados en bloque con un amplio rango de aplicaciones en comunicaciones digitales y almacenamiento de información. Los códigos *Reed-Solomon* son usados para corregir errores en algunos sistemas, incluidos:

- ◆ Dispositivos de almacenamiento (cintas, *CD's*, *DVD*, códigos de barras, etc.)
- ◆ Comunicaciones inalámbricas o móviles (celulares, enlaces microondas, etc.)
- ◆ Comunicaciones vía satélite
- ◆ Televisión digital / *DVB*
- ◆ *Modem* de alta velocidad tales como *ADSL*, *xDSL*, etc.

Figura 1D. Diagrama de bloques simplificado de un sistema con codificación *Reed Solomon*



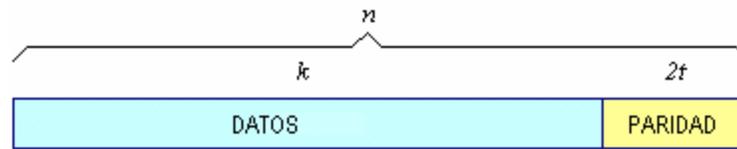
Riley, Martyn Riley y Richardson, Iain. Reed-Solomon Codes. An introduction to Reed-Solomon codes: principles, architecture and implementation. [online]. 1998. Disponible en http://www.4i2i.com/reed_solomon_codes.htm.

El codificador *Reed Solomon* toma un bloque de datos digitales y agrega *bits* redundantes extra. Los errores ocurren durante la transmisión o almacenamiento por un sinnúmero de razones (por ejemplo ruido o interferencia, rayones en un *CD*, etc.). El decodificador *Reed Solomon* procesa cada bloque e intenta corregir errores y recuperar los datos originales. El número y tipo de errores pueden ser corregidos dependiendo de las características del código usado.

PROPIEDADES DE LOS CODIGOS REED SOLOMON. Los códigos *Reed Solomon* son un subconjunto de códigos *BCH* y son códigos en bloque lineal. Un código *Reed Solomon* está especificado por $RS(n,k)$ con símbolos *s-bit*. Esto significa que el codificador toma k símbolos de datos de s *bits* cada uno y agrega $n - k$ símbolos de paridad para hacer una palabra clave de n símbolos. Hay $n - k$ símbolos de paridad de s *bits* cada uno. Un decodificador *Reed Solomon* puede corregir hasta t símbolos conteniendo errores en una palabra clave, donde $2t = n - k$.

La siguiente figura (Figura 2D) describe una palabra clave típica de un código *Reed Solomon* (este es conocido como un código sistemático porque los datos a la izquierda no cambian y solo se le agregan símbolos de paridad)

Figura 2D. Palabra clave *Reed Solomon*



Ibíd.

Ejemplo: Un código *Reed Solomon* muy popular es $RS(255,223)$ con 8 símbolos. Cada palabra clave contiene 255 *bytes*, de los cuales 223 *bits* son datos y 32 *bits* son paridad. Para este código:

$$n = 255, \quad k = 223, \quad s = 8$$

$$2t = 32 \quad t = 16$$

El decodificador puede corregir cualquier error de 16 símbolos en la palabra clave. P. ej: errores de hasta 16 *bits* en cualquier parte de la palabra clave se pueden corregir automáticamente. Determinando un tamaño de símbolo s , la longitud máxima de la palabra clave (n) para un código *Reed Solomon* es $n = 2^s - 1$. Por ejemplo la longitud máxima de un código con símbolos de 8 *bits* ($s = 8$) es de 255 *bytes*. Los códigos *Reed Solomon* pueden ser acortados (conceptualmente) haciendo 0 un número de símbolos de datos en el codificador, no transmitiendo estos, y reinsertándolos en el decodificador. Ejemplo: el código (255,223) descrito arriba puede ser acortado a (200,168). El codificador toma un bloque de 168 *bytes* de datos, agrega (conceptualmente) 55 *bytes* cero, crea una palabra clave

(255, 223), transmite solo los 168 *bytes* de datos y 32 *bytes* de paridad. La cantidad de “potencia” de computo requerida para codificar y decodificar el código *Reed Solomon* es relativo al número de símbolos de paridad por palabra clave. Un valor grande de t significa que se puede corregir un gran número de errores pero también que requiere mas potencia de cómputo que un valor de t pequeño

ERRORES DE SIMBOLO. Un error de símbolo ocurre cuando 1 *bit* en un símbolo esta errado o cuando todos los *bits* en un símbolo están errados. Ejemplo: *RS(255,223)* puede corregir 16 errores de símbolo. En el peor de los casos, pueden ocurrir errores de 16 *bits*, cada uno en un símbolo separado (*byte*) de modo que el decodificador corrige errores de 16 *bits*. En el mejor de los casos, ocurren errores de 16 *bits* completos de modo que el decodificador corrige errores de 16×8 *bits*. Los códigos *Reed Solomon* son particularmente buenos para corregir errores de ráfaga de impulsos (*Burst*) (donde una serie de *bits* en la palabra clave se reciben con error)

DECODIFICACION. El procedimiento algebraico de la decodificación *Reed Solomon* puede corregir errores y cancelaciones. Una cancelación ocurre cuando se conoce la posición de un símbolo errado. Un decodificador puede corregir hasta t errores o hasta $2t$ cancelaciones. La información cancelada puede a menudo ser proporcionada por el demodulador en un sistema de comunicación digital, si por ejemplo, el demodulador “embandera” los símbolos recibidos que probablemente

contienen errores. Cuando una palabra clave es decodificada, se dan tres posibles resultados

1. Si $2s + r < 2t$ (s errores, r cancelaciones) entonces la palabra clave transmitida originalmente siempre se recuperará, de otro modo
2. El decodificador detectará que no puede recuperar la palabra clave original e indica este hecho, o
3. El decodificador trabajara equivocadamente y recuperará una palabra clave incorrecta sin indicación alguna.

La probabilidad de cada una de las tres posibilidades depende del código *Reed Solomon* particular y sobre el número de distribución de errores

GANANCIA DE CODIFICACION. La ventaja de usar códigos *Reed Solomon* es que la probabilidad de un error remanente en los datos decodificados es (usualmente) mucho mas baja que la probabilidad de un error si no se usara *Reed Solomon*. Esto a menudo se describe como ganancia de codificación.

Ejemplo: un sistema de codificación digital se diseña para operar con una tasa de error de *bit* (*BER*) de 10^{-9} , si por ejemplo, no más que 1 en 10^9 *bits* se reciben en error. Esto se puede lograr aumentando la potencia del transmisor o agregando *Reed Solomon* (u otro tipo de corrección de errores por reenvío – *Forward Error*

Correction –). *Reed Solomon* le permite al sistema lograr este *BER* objetivo con una baja potencia de salida en el transmisor. El “ahorro” de potencia aportado por *Reed Solomon* (en decibeles) es la ganancia de código

CAMPO ARITMÉTICO FINITO (GALOIS). Los códigos *Reed Solomon* se basan en un área matemática especializada conocida como *Campos Galois* o *Campos Finitos*. Un campo finito tiene la propiedad de que las operaciones aritméticas sobre los elementos del dominio siempre tienen un resultado en el dominio. Un codificador o decodificador *Reed Solomon* necesita llevar a cabo esas operaciones aritméticas. Estas requieren un hardware especial o implementar funciones por software

GENERADOR POLINÓMICO. Una palabra clave *Reed Solomon* se genera usando un polinomio especial. Todas las palabras clave válidas son exactamente divisibles por el generador polinómico. La forma general del generador polinómico es:

$$g(x) = (x - \alpha^i)(x - \alpha^{i+1}) \dots (x - \alpha^{i+2t})$$

Y la palabra clave se construye usando

$$c(x) = g(x).i(x)$$

Donde $g(x)$ es el generador polinómico, $i(x)$ es el bloque de información, $c(x)$ es una palabra clave válida y a es referenciado como un elemento primitivo del dominio. Ejemplo: Generador para $RS(255,249)$

$$g(x) = (x - \alpha^0)(x - \alpha^1)(x - \alpha^2)(x - \alpha^3)(x - \alpha^4)(x - \alpha^5)$$

$$g(x) = x^6 + g_5x^5 + g_4x^4 + g_3x^3 + g_2x^2 + g_1x^1 + g_0$$

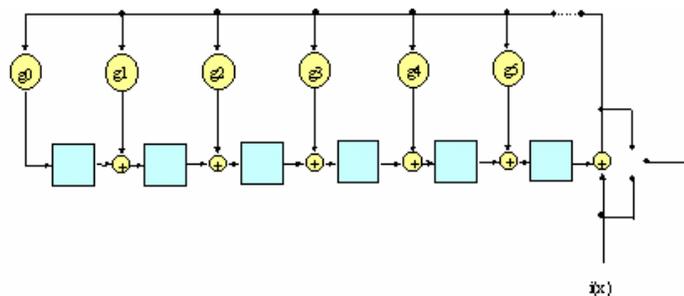
ARQUITECTURAS PARA CODIFICAR Y DECODIFICAR CODIGOS REED-SOLOMON.

ARQUITECTURA DEL CODIFICADOR. Los $2t$ símbolos de paridad en una palabra clave sistemática *Reed Solomon* se dan por:

$$p(x) = i(x) \cdot x^{n-k} \text{ mod } g(x)$$

El siguiente diagrama, descrito en la figura 3D, describe un arquitectura para un codificador sistemático $RS(255,249)$

Figura 3D. Palabra clave *Reed Solomon*

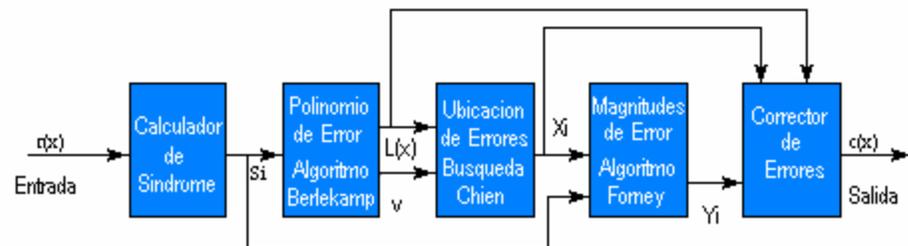


Ibíd.

Cada uno de los 6 registros posee un símbolo (8 *bits*). Los operadores aritméticos llevan a cabo la suma o multiplicación de campo finito sobre un símbolo completo

ARQUITECTURA DEL DECODIFICADOR. En la siguiente figura se describe una arquitectura general para decodificar códigos *Reed Solomon*

Figura 4D. Arquitectura general para decodificar códigos *Reed Solomon*



Ibíd.

Leyenda

- $r(x)$ palabra clave recibida
- Si síndromes
- $L(x)$ polinomio localizador de errores
- Xi localizaciones de error
- Yi magnitudes de error
- $c(x)$ palabra clave recuperada
- v número de errores

La palabra clave recibida $r(x)$ es la palabra clave transmitida original $c(x)$ más los errores:

$$r(x) = c(x) + e(x)$$

Un decodificador *Reed Solomon* intenta identificar la posición y magnitud de hasta t errores (hasta $2t$ borrones) y corregir los errores o borrones

CALCULO DE SINDROME. Este cálculo es similar al cálculo de paridad: Una palabra clave *Reed Solomon* tiene $2t$ síndromes que dependen solo de los errores (no de la palabra clave transmitida). Los síndromes pueden ser calculador por la sustitución de las raíces $2t$ del generador de polinomio $g(x)$ dentro de $r(x)$

HALLANDO LA LOCALIZACION DE LOS ERRORES DE SIMBOLO. Esto involucra resolver ecuaciones simultáneas con t incógnitas. Un algoritmo rápido ampliamente usado es el *Algoritmo Forney*

IMPLEMENTACION DE CODIFICADORES Y DECODIFICADORES REED SOLOMON. Se pueden implementar ya sea por software o por hardware de aplicación especial

IMPLEMENTACION BASADA EN HARDWARE. Existen un número de implementaciones comerciales basadas en hardware. Algunos sistemas existentes

usan circuitos integrados producidos a gran escala que codifican y decodifican códigos *Reed Solomon* (Como el *S1109R*). Estos tienden a soportar un cierto grado de programabilidad (por ejemplo $RS(255,k)$ donde t tiene de 1 a 16 símbolos). Una tendencia reciente es hacia el *VHDL* o diseños *Verilog* (núcleos lógicos o núcleos de propiedad intelectual), Estos tienen un numero de ventajas sobre los *IC's* estándar. Un núcleo lógico puede ser integrado con un *VHDL* u otros componentes y sintetizado a un *FPGA* o *ASIC*. Esto facilita también los diseños de Sistemas en un solo chip donde los múltiples módulos pueden combinarse en un único circuito integrado.

Dependiendo de los volúmenes de producción, los núcleos lógicos pueden a menudo ofrecer sistemas significativamente menos costosos que los circuitos integrados estándar. A través del uso de núcleos lógicos el diseñador evita el potencial necesario para hacer una compra de por vida de un circuito integrado *Reed Solomon*

IMPLEMENTACION BASADA EN SOFTWARE. Hasta hace poco, la implementación por software en tiempo real requería mucha potencia computacional para todos excepto para los códigos *Reed Solomon* más simples (códigos con pequeños valores de t). La mayor dificultad en implementar códigos *Reed Solomon* en software que los procesadores de aplicación general no soportan las operaciones aritméticas de los Campos *Galois*. Por ejemplo, implementar una multiplicación en el Campo *Galois* por software requiere una

prueba de 0, dos tablas logarítmicas de búsqueda, un modulo sumador y tabla antilogarítmica de búsqueda. Sin embargo, un diseño cuidadoso conjuntamente con incrementos en el rendimiento del procesador da como resultado que las implementaciones en software puedan operar a tasas de datos relativamente altas.

ANEXO E. CODIGO PN. CODIGO DE PSEUDO RUIDO

Un código de pseudo ruido es un código generado determinísticamente el cual permite que, al ser mezclado con la información a transmitir, quede enmascarada, dando la impresión de ser ruido. Además permite que después del proceso de demodulación, la información sea recuperada en un receptor apropiado. Un código de pseudo ruido obra como un ruido, tal como un ruido portador (pero determinista), usado para la propagación de la energía de la señal. Es importante la selección de un buen código porque el tipo y la longitud del código determinan los límites de la capacidad del sistema. La secuencia de código *PN* o es un pseudo ruido o una secuencia pseudo aleatoria de 1's y 0's, pero no una secuencia aleatoria real (puesto que es periódica). Las secuencias aleatorias no pueden ser previstas. La autocorrelación de un código *PN* tiene propiedades similares a las del ruido blanco

- Seudo Aleatorio. No aleatorio, pero se ve aleatorio para el usuario quien no conoce el código
- Determinístico, señal periódica que es conocida tanto para el transmisor como para el receptor. Lo largo del periodo del código de propagación *PN*, lo cercano que sea transmitida la señal sea una onda binaria aleatoria real. determina lo difícil que sea detectar esta
- Propiedades estadísticas del ruido blanco muestreado

Longitud

- ◆ Código corto: La misma secuencia PN para cada símbolo de datos ($N_c.T_c = T_s$)
- ◆ Código largo: El código de la secuencia PN es mucho mas largo que el símbolo de datos, de modo que un trozo de patrón distinto se asocia a cada símbolo ($N_c.T_c \gg T_s$)

PROPIEDADES DE LAS SECUENCIAS PN .

PROPIEDAD DE BALANCE: En cada periodo de la secuencia el número de 1's binarios difiere del número de 0's binarios un dígito como máximo (para N_{cc} impar)

$$p_n = +1 +1 +1 -1 +1 -1 -1 \rightarrow \Sigma = +1$$

Cuando se esta modulando una portadora con una secuencia de codificación PN , el balance uno cero (componente DC) puede limitar el grado de supresión de portadora alcanzable, puesto que la supresión de portadora es dependiente de la simetría de la señal modulante

DISTRIBUCION DE LA LONGITUD DE CARRERA. Una carrera es una secuencia de un solo tipo de dígitos binarios. En medio de las carreras de unos y

ceros en cada periodo es deseable que alrededor de la mitad de las carreras de cada tipo tengan una longitud de 1, alrededor de 1/4 una longitud de 2 1/8 una longitud de 3 y así sucesivamente

AUTOCORRELACION. El origen del nombre pseudo ruido es que la señal digital tiene una función de correlación la cual es muy similar a la de la señal de ruido blanco: impulsos análogos. La función de autocorrelación para la secuencia periódica PN se define como el número de concordancias menos el número de discordancias en un término por la comparación de términos de un periodo completo de la secuencia con un desplazamiento cíclico (posición τ) de la secuencia en si misma

$$Ra(\mathbf{t}) = \int_{-Nc.Tc/2}^{Nc.Tc/2} pn(t).pn(t + \mathbf{t})dt$$

Es mejor que $Ra(\mathbf{t})$ no sea mas largo que un conteo si no esta sincronizado ($\mathbf{t} = 0$). Para secuencias PN la autocorrelación tiene un pico máximo solo para una sincronización perfecta de dos secuencias idénticas (similar al ruido blanco). La sincronización del receptor se basa en esta propiedad

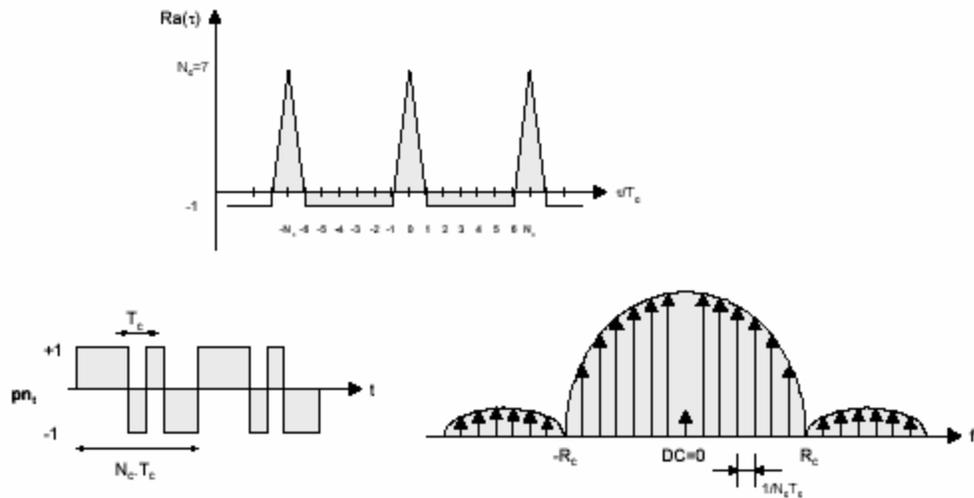
ESPECTRO DE FRECUENCIA. Debido a la naturaleza periódica de la secuencia PN el espectro de frecuencia tiene líneas espectrales las cuales se hacen mas

cercanas entre si incrementando la longitud de las secuencia N_c . Cada línea esta adicionalmente cubierta por un *scrambling* de datos, lo cual propaga cada línea espectral y adicionalmente rellena entre las líneas haciendo el espectro más aproximadamente continuo. La componente DC es determinada por el balance entre ceros y unos de la secuencia PN

Figura 1E. Espectro de frecuencia con código PN

$$\begin{array}{r}
 pn(0) = +1 +1 +1 -1 +1 -1 -1 \\
 \underline{pn(0) = +1 +1 +1 -1 +1 -1 -1} \\
 \quad +1 +1 +1 +1 +1 +1 +1 = \Sigma = 7 = Ra(\tau=0)
 \end{array}$$

$$\begin{array}{r}
 pn(0) = +1 +1 +1 -1 +1 -1 -1 \\
 \underline{pn(1) = +1 +1 -1 +1 -1 -1 +1} \\
 \quad +1 +1 -1 -1 -1 +1 -1 = \Sigma = -1 = Ra(\tau=1)
 \end{array}$$



J., Meel. Sirius Communications. Paper: "Spread Spectrum (SS)". [online]. Bélgica. SSS Online, 1999. Ultima revisión: octubre 20 de 2003. Disponible de forma gratuita en www.sss-mag.com/pdf/Ss_jme_denayer_intro_print.pdf

CORRELACION DE CRUCE. La correlación por cruce describe la interferencia entre códigos pn_i y pn_j

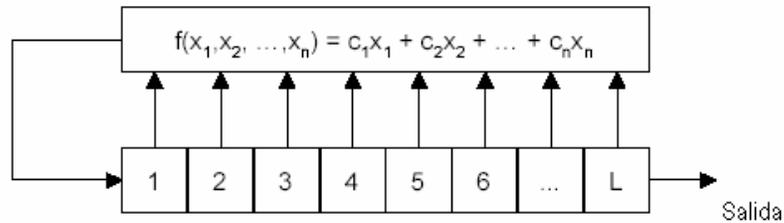
$$R_c(\mathbf{t}) = \int_{-N_c T_c / 2}^{N_c T_c / 2} pn_i(t) \cdot pn_j(t + \mathbf{t}) dt$$

La correlación por cruce es la medida de la concordancia entre dos códigos diferentes pn_i y pn_j . Cuando la correlación por cruce $R_c(\mathbf{t})$ es 0 para todos los \mathbf{t} , los códigos son llamados ortogonales. En *CDMA* múltiples usuarios ocupan el mismo ancho de banda de *RF* y simultáneamente transmiten. Cuando los códigos de los usuarios son ortogonales, no hay interferencia entre los usuarios después de la *despropagación* y se protege la privacidad de cada usuario. En la practica los códigos no son perfectamente ortogonales, por lo tanto la distorsión por cruce entre los códigos de usuarios introduce un degradación en el rendimiento (se incrementa la potencia del ruido después de la *despropagación*), lo cual limita el número máximo de usuarios simultáneos.

Tipos

Secuencia m: Un simple registro de desplazamiento generador (*SSRG*) tiene las señales de retroalimentación retornadas a una entrada del registro de desplazamiento (línea de retardo). El *SSRG* es lineal si la función de retroalimentación se puede expresar como una suma modulo 2 (XOR)

Figura 2E. Secuencia m



Ibíd. p. 15

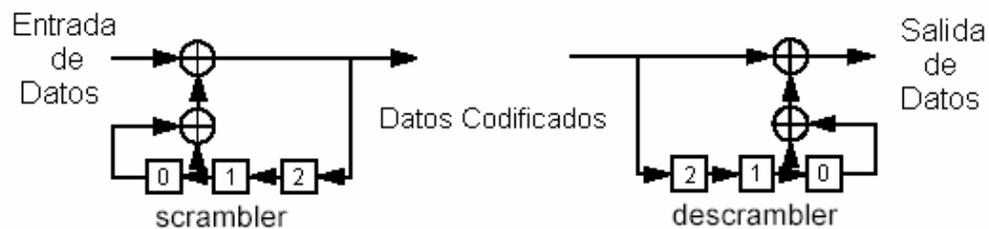
La función de retroalimentación $f(x_1, x_2, \dots, x_n)$ es una suma modulo 2 de los contenidos x_i de las celdas del registro de desplazamiento con c_i la existencia la conexión de los coeficientes de retroalimentación ($c_i = 0 =$ abierto, $c_i = 1 =$ conectado). Un *SSRG* con L *flip-flop's* produce secuencias que dependen de la longitud del registro L , las conexiones de la toma de retroalimentación y las condiciones iniciales. Cuando el periodo (longitud) de la secuencia es exactamente $N_c = 2^L - 1$, la secuencia *PN* es llamada una *secuencia de longitud máxima* o simplemente una *secuencia m*.

ANEXO F. SCRAMBLER

El común denominador en esta investigación lleva a que el *scrambler* es un tipo de codificador usado en una transmisión determinada para que solamente un receptor debidamente equipado pueda interpretar la información transmitida, haciéndola ininteligible para receptores que no. En todo caso, la codificación se hace en función de un generador polinómico que es distinto dependiendo de si el tipo de *scrambling* utilizado es serie o paralelo

PRINCIPIO DEL SCRAMBLING⁹⁹.

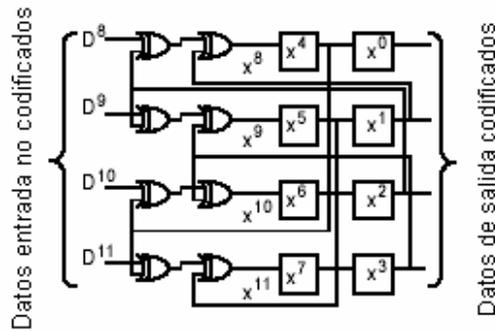
Figura 1F. Ejemplo de *scrambling* y *descrambling* en forma serie



Walter, Rick y Dugan, Richard. Presentación de la Plenaria del congreso de la IEEE en *Kauai*: Low overhead coding proposal for 10Gb/s serial links [online]. *Kauai*, 1999. Disponible en http://grouper.ieee.org/groups/802/3/10G_study/public/nov99/walker_1_1199.pdf

⁹⁹ De la presentación de la Plenaria del congreso de la IEEE en *Kauai*, Noviembre de 1999: "Low overhead coding proposal for 10Gb/s serial links" por Rick Walter y Richard Dugan

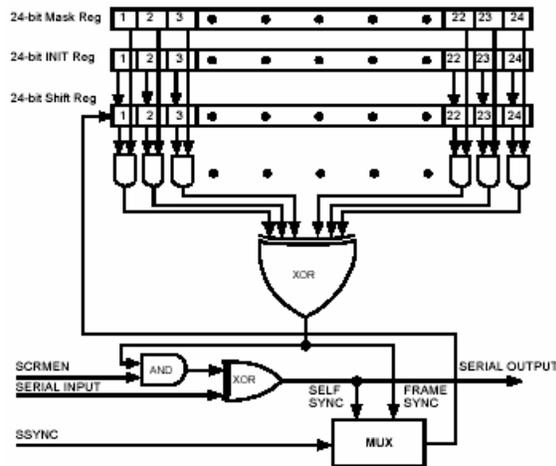
Figura 2F. Ejemplo de *scrambling* en forma paralela



Ibíd.

En el *S1109CR* el *scrambler* se utiliza para aleatorizar la información de modo que los patrones repetitivos de datos en la entrada no ocasionen componentes espectrales no deseados

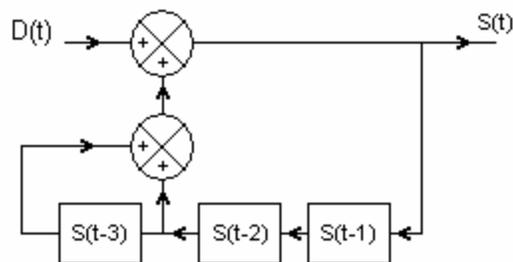
Figura 3F. Módulo *Scrambler* usado por el *S1109CR*



INTEL CORPORATION. Op. Cit., p. 11

RELACIÓN ENTRE EL MODO SERIE Y EL MODO PARALELO. Tomando el circuito de *scrambling* descrito anteriormente en la figura 16, el *scrambler* serie se define claramente en la figura 19, la cual contiene un diagrama de bloques un poco mas detallado.

Figura 4F. *Scrambler* serie descrito en la figura 1F como referencia



Walter, Rick y Dugan, Richard. Op.Cit.

Donde $D(t)$ son los datos de entrada, $S(t)$ son los correspondientes datos de salida en un tiempo t , donde $t = 0,1,2,3...$ a medida que el tiempo se incrementa. Los rectángulos contienen las señales en los candados (*latches*), o también, un retardo de un ciclo de reloj. Una ecuación recursiva describiendo el *scrambler* es

$$S(t) = S(t - 3) + S(t - 2) + D(t)$$

Donde “+” denota suma complemento a 2 (función OR exclusiva), la misma que se emplea en los puntos de suma de la figura. Es conveniente reorganizar los términos y agregar un compensador para hacer todos los índices positivos, obteniendo

$$(I) S(3) = S(0) + S(1) + D(3)$$

$$(II) D(3) = S(3) + S(1) + S(0) \text{ (la suma y la resta son equivalentes)}$$

Esto corresponde a un *scrambler* con un polinomio típicamente marcado como

$$X^3 + X^1 + X^0$$

Ahora se paraleliza la función para hacer un *scrambler* de 4 *bits* de ancho. Quitando los paréntesis se tiene ahora que $S(4) = S4$. Se toman los cuatro *bits* de datos de $D5$ a $D8$ y se generan los *bits* codificados de $S5$ a $S8$ en una sencilla operación sincronizada. De la relación de recursión (I)

$$S5 = S2 + S3 + D5$$

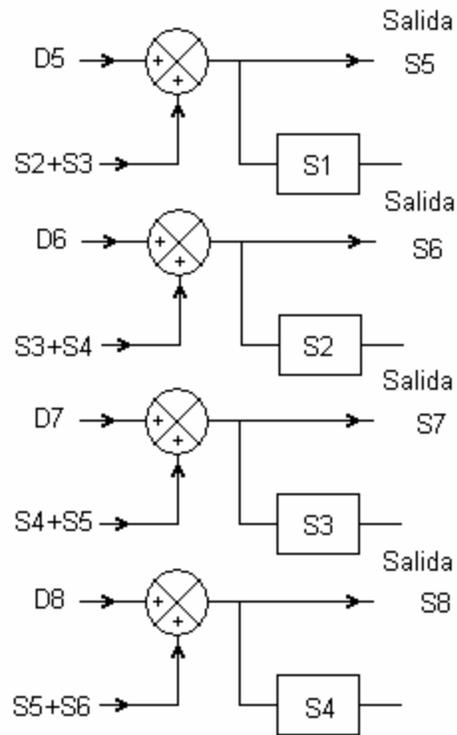
$$S6 = S3 + S4 + D6$$

$$S7 = S4 + S5 + D7$$

$$S8 = S5 + S6 + D8$$

Una posible implementación de este resultado se presenta en el diagrama de bloques descrito en la figura 5F

Figura 5F. Resultado de las operaciones



Ibíd.

De modo que, $S5$ a $S8$ se generan en un ciclo de reloj a partir de $D5$ a $D8$. Se usa un banco de candados para almacenar los valores computados previamente de $S1$ a $S4$ y entonces usarlos en la relación de recurrencia. El ajuste crítico se da en los cómputos de $S7$ y $S8$. $S7$ por ejemplo se define como

$$S7 = S5 + S4 + D7$$

S5 no esta disponible en los candados de almacenamiento, debe ser generado por

$$S5 = S3 + S2 + D5$$

De modo que la verdadera ecuación para S7 es

$$S7 = (S3 + S2 + D5) + S4 + D7$$

La cual puede ser implementada fácilmente en el diagrama anterior con puertas *XOR* de 3 entradas con dos etapas en cascada. La ampliación de este ejemplo a $X^{58} + X^{19} + 1$ con un bus de datos de extensión 64 es deducible. Fuera de las consideraciones del estado del arte de este trabajo de investigación, ya que el prototipo objeto de este estudio también trabaja conectado con un ordenador, parece imprescindible anexar algo de la teoría referente al puerto *UART*, el puerto serie que a través del protocolo *RS-232-C* va a permitir la comunicación entre estas dos máquinas

**ANEXO G. PROGRAMA DEL PIC 16F877 IMPLEMENTADO EN LENGUAJE
ENSAMBLADOR**

ANEXO H. SELECCIÓN DE LA FRECUENCIA DE RELOJ

“Un reloj maestro de 100KHz podría ser demasiado bajo. No podemos garantizar que este trabaje. Se emplea la siguiente ecuación:

$$\text{MasterClock} = 4 * \text{SymbolRate} * \text{Rel. Interpolación } (I)$$

Donde la Interpolación es un entero y no puede ser menor que 1 y la frecuencia de salida (portadora) es menor al 40% de la del master clock.

Saludos,

Chuck Frank

Intel Corporation Inc.”¹⁰

La respuesta de *Franck Chuck*, es consecuencia de las inquietudes manifestadas por las frecuencias de trabajo que el *ASIC STEL-1109* podía manejar. A pesar de haber trabajado todo el tiempo con la incertidumbre de si el integrado podría trabajar a frecuencias menores, vendría casi al final de las pruebas una respuesta de otro especialista de soporte técnico, *Kevin Zhang*:

¹⁰ Traducción de uno de los e-mail's recibidos en el transcurso de la investigación

“..If your master clock is at least 12.5MHz, it should work just fine. When it gets lower, the performance will be degraded. And when it gets very low, the chip may stop functioning. I can't really tell what's threshold, and it's not defined.

Wish you a great holiday!

Kevin”

Por seguridad, se escogió un reloj de 14.318 MHz, estable como la estabilidad del cristal que proporciona esta frecuencia de resonancia.

La frecuencia de portadora se preconfigura a 20 KHz en el encendido situándola dentro del ancho de banda del equipo y haciéndola fácil de muestrear. La curva de respuesta en frecuencia se describe en la guía de laboratorio 1 y en el boletín técnico

ANEXO I. COMENTARIOS

En respuesta a las inquietudes presentadas por los calificadoros de este proyecto se anexa esta hoja con las aclaraciones respectivas acerca del manejo de las interfaces para el entrenador.

La interfase del entrenador con el computador esta hecha a través de un lenguaje visual – *Delphi 7 empresarial* - bajo entorno Windows (*GUI*). Hablando en porcentajes, el 90% del proceso digital lo realizara el entrenador, el 10% restante será la ejecución de comandos e interpretación de los mismos por parte del computador. Por esta misma razón se planteo en principio que el computador podía ser opcional.

Los encargados del proyecto en todo caso aclaran que el desarrollo y evaluación de este software no están dentro del área del proyecto, bien pudiera omitirse o ser trabajo encargado a terceras personas más aptas y preparadas en programación. En el futuro se podrían diseñar otras aplicaciones sobre la misma base

La *board* se asemeja a las *boards* de evaluación presentadas por *INTEL*, pero con marcadas diferencias.

Para comenzar, el diseño de esta no se baso en *boards* de este tipo, como la *STEL 1209*. El diseño de este entrenador trata de trabajar con frecuencias mas bajas que las usuales para este integrado, pues es para un fin didáctico. En lo posible, el entrenador es transparente al usuario. En el *datasheet* de la *board* de evaluación del *STEL-1209* se encontró sí, una interfase software muy amigable que se usó como referencia para el desarrollo del software necesario para el proyecto. De ninguna manera este software es una copia del primero, pues ya que el hardware es distinto, se requiere un software distinto para controlarlo a través del *PC*. Para desarrollar el proyecto se pusieron a prueba conocimientos en hardware por tecnologías como *ASIC*, microcontroladores, *TTL's*, módulos híbridos, protocolo *USART*, dispositivos *SMD*, programación en *assembler* y uso de *MPLAB 6 IDE*, programación en *Delphi*, manejo general de protocolos de controladores y buses (*STEL-1109*, *LCD*, *PIC*); el trabajo de investigación requirió consultar muchas fuentes y hasta otros especialistas en otros lugares del mundo, como los ingenieros de *INTEL*, asistentes técnicos del *S1109CR* y especialistas del Reino Unido y Brasil representantes de *Radiometrix*, foros online en Internet, grupos de noticias, libros especializados, etc.

Amplia información adicional y otros documentos sobre la investigación inherente al proyecto están compartidos públicamente en

<http://www.geocities.com/tesiscutb2003>

BIBLIOGRAFIA

Tomasi, Wayne. Sistemas de Comunicaciones Electrónicas, segunda edición en español. México: Editorial *Prentice Hall Latinoamericana*, 1996. 858 páginas, ISBN 0-13-220021-X

L. Morris, Robert and R. Millar, John. Designing with TTL Integrated Circuits International Student Edition. Tokyo: McGraw-Hill Kogakusha, Ltd., 1971. 322 páginas

Angulo Usategui, José Maria; Romero Yesa, Susana y Angulo Martínez, Ignacio. Microcontroladores PIC: Diseño Practico Aplicaciones. Madrid: McGraw-Hill/ Interamericana de España, S.A.U. 2000. 232 páginas

C. Dorf, Richard. Circuitos Eléctricos: introducción al análisis y diseño. Bogota: Alfaomega S.A., 2000. 1124 páginas

Walter, Rick y Dugan, Richard. Presentación de la Plenaria del congreso de la IEEE en *Kauai*: Low overhead coding proposal for 10Gb/s serial links [online].

Kauai, 1999. Disponible en

http://grouper.ieee.org/groups/802/3/10G_study/public/nov99/walker_1_1199.pdf

Riley, Martyn Riley y Richardson, Iain. Reed-Solomon Codes. An introduction to Reed-Solomon codes: principles, architecture and implementation. [online]. 1998. [citado 13-02-2003]. Disponible en http://www.4i2i.com/reed_solomon_codes.htm

Frank Durda IV. Serial and *UART* Tutorial [online]. FreeBSD.org, 1996. [Citado el 3 de Marzo de 2003]. Disponible en http://www.freebsd.org/doc/en_US.ISO8859-1/articles/serial-uart/, de forma gratuita

J., Meel. Sirius Communications. Paper: "Spread Spectrum (SS)" [online]. Bélgica. SSS Online, 1999. Actualizado octubre 20 de 2003. [citado 2 de abril de 2003] Disponible en www.sss-mag.com/pdf/Ss_jme_denayer_intro_print.pdf, de forma gratuita

Escudero Aldarriaga, Alvaro; Bustos Muñoz, Jorge; Rojas Quintana, Mauricio. FEC (Forward Error Correction) y Código Reed-Solomon [online]. Concepción. Universidad de Concepción, 2001. Actualizado 20-11- 2002 [Citado 18 de enero 2003]. En http://www.die.udec.cl/~comdatos/trabajos/Codigos_FEC_salomon.pdf

Microchip Technology Inc. MPASM USER'S GUIDE with MPLINK and MPLIB , [online], 1999. Actualizado 11-04-2003. [Citado 10 de octubre de 2003]. Disponible en <http://www.microchip.com/Download/tools/picmicro/code/mpasm/33014g.pdf> de forma gratuita

Microchip Technology Inc. Implementing a Table Read, [online], 1999. Actualizado 11-04-2003. [Citado 04 de octubre de 2003]. Disponible en <http://www.microchip.com/download/appnote/pic16/00556e.pdf> de forma gratuita

Radiometrix Ltd.TX2 & RX2 Data Sheet [online], 2001. Issue G. [Citado 10-12-2002]. Disponible en <http://www.artbrno.cz/Radiometrix/dsheets/tx2rx2.pdf> de forma gratuita