

01170



**UNIVERSIDAD NACIONAL AUTONOMA
DE MEXICO**

**DIVISION DE ESTUDIOS DE POSGRADO
FACULTAD DE INGENIERIA**

**Diseño e Implementación de un Transceiver
Inalámbrico en RF para Sismología**

TESIS

Que para obtener el grado de

MAESTRO EN INGENIERIA ELECTRICA

(AREA: TELECOMUNICACIONES)

Presenta:

Ing. Henry David Kemper Castro

Director de Tesis: M.I. Juan Carlos Roa Beiza

Asesor: M.I. Nicolás Kemper Valverde

México D.F. -- 2005



m. 342239



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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

AGRADECIMIENTOS

Doy Gracias a Dios Jehová por la suerte de tener a Trini y Grima como mis padres que siempre me brindaron su amor, ejemplo, comprensión, su apoyo y confianza a lo largo de mi vida personal y profesional.

A mis hermanos Walter, Norma y Gladis que gracias a su cariño, preocupación, respeto y apoyo pude seguir adelante y así poder concluir cada etapa iniciada en mi vida, y sobre todo en la culminación de esta tesis.

A mis tíos Nicolás y Vilma por todo el cariño y apoyo brindado en mis estudios de maestría, y que junto a Paola son mi segunda familia en estas tierras mexicanas.

Al M.I. Juan Carlos Roa Beiza por el apoyo y orientación brindada en la elaboración de mi tesis. Así como a todos mis maestros de la maestría que me ayudaron en mi formación profesional.

A toda mi familia, amigos y colegas que al leer estas líneas se sienten contentos e identificados con mi titulación.

CONTENIDO

CONTENIDO

	Pág.
Resumen	03
Introducción	07
1. Planteamiento del Problema	13
1.1. Antecedentes	13
1.2. Panorama Actual	17
1.3. Planteamiento del Problema	18
1.4. Propuesta de Solución al Problema	19
2. Diseño e Implementación del Transceiver Inalámbrico de Datos en RF	25
2.1 Transceiver Inalámbrico de Datos en RF	25
2.1.1 Descripción General	25
2.1.2 Principales Módulos de Comunicaciones en RF en el Mercado Actual	26
2.1.3 Transceiver Inalámbrico Híbrido de Amplificador Secuenciado (HAS)	28
2.1.3.1 Teoría de Operación	28
2.1.3.2 Flujo de la Señal en el Receptor	29
2.1.3.3 Flujo de la Señal en el Transmisor	31
2.1.3.4 Requerimientos de Potencia	33
2.1.3.5 La Antena – Puerto de Entrada/Salida en RF	34
2.1.3.6 El Generador de Pulsos	35
2.1.3.7 Filtro Pasa Bajos	36
2.1.3.8 Acoplamiento en Banda Base	38
2.1.3.9 Diseño del Transceiver HAS en su Configuración OOK/ASK	40
2.1.3.10 Módulo Transceiver Híbrido DR3000	41
2.1.3.10.1 Configuración OOK a 19200 bps	44
2.1.3.11 Salida de Datos	45

2.2	Módulo Transmisor Digital en RF con base en el Microcontrolador	
	PIC16F873	46
2.2.1	Descripción del Módulo Transmisor Digital en RF	46
2.2.2	Consideraciones del Diseño del Módulo Transmisor digital en RF..	47
2.2.3	Flujo de los Datos en el Transmisor Digital en RF	49
2.2.4	Rutinas del Programa de Control del Módulo Transmisor	52
2.2.4.1	Rutina del Programa Principal: <i>Principal</i>	54
2.2.4.2	Rutina de Lectura de Datos recibidos: <i>Rutina_RX</i>	57
2.2.4.3	Rutina de Codificación: <i>ConByte12</i>	59
2.2.4.4	Rutina de Interrupción: <i>RutinaInterrupción</i>	62
2.2.4.5	Rutina de Configuración de Parámetros del Transmisor: <i>Set_Parametros_Com</i>	65
2.3	Módulo Receptor Digital en RF con base en el Microcontrolador	
	PIC16F873	69
2.3.1	Descripción del Módulo Receptor Digital en RF	69
2.3.2	Flujo de los Datos del Módulo Receptor Digital en RF	70
2.3.3	Rutinas del Programa de Control del Módulo Receptor Digital en RF	71
2.3.3.1	Rutina Principal del Programa: <i>PPrincipal</i>	73
2.3.3.2	Rutina de Decodificación: <i>Decodificar</i>	76
2.3.3.3	Rutina de Transmisión de Datos decodificados: <i>Tx_RS232</i>	79
2.3.3.4	Rutina de Configuración de Parámetros del Receptor: <i>Set_Parametros_Com</i>	79
2.3.3.5	Rutina de Interrupción del Módulo Receptor Digital en RF: <i>RutinaInterrupcion</i>	80

	Pág.
2.4 Integración de los Módulos Transmisor y Receptor en RF	82
2.4.1 Descripción de los Módulos Transmisor y Receptor Inalámbrico en RF	82
2.4.2 Implementación de los Módulos Transceivers Inalámbricos en RF	85
3. Pruebas y Resultados	91
3.1 Ambiente de Desarrollo MPLAB IDE de Microchip	91
3.1.1 Simulación de los Módulos Transmisor y Receptor	93
3.1.2 Monitoreo y Depuración del Disparo de Lectura en el Receptor	93
3.2 Pruebas de Transmisión y Recepción	96
3.2.1 Pruebas de Lectura con un Programa Terminal de PC	96
3.2.2 Pruebas de Transmisión y Recepción de datos asíncronos	100
3.3 Pruebas de Transmisión y Recepción con Registrador Sismológico ...	104
3.3.1 Prueba 1	104
3.3.2 Prueba 2	108
3.3.3 Prueba 3	109
3.3.4 Prueba 4	114
3.3.4 Prueba de Alcance de Comunicación	118
4. Conclusiones y Recomendaciones	121
4.1 Conclusiones	121
4.2 Recomendaciones	122
Apéndices	
A. Microcontrolador PIC16F873	127
B. Rutinas Importantes implementados en los Microcontroladores PIC	131
Bibliografía	149

RESUMEN

RESUMEN

Se realizó el diseño e implementación de un transceiver de datos como propuesta de solución al enlace en RF de estaciones sísmicas digitales remotas con una estación central. Usando un módulo comercial, un transceiver DR3000 de RF Monolithics, se diseñó las interfaces en hardware y software que permitan la transmisión de datos digitales procedentes de un registrador sísmico. Los programas de control del módulo transceiver y los algoritmos de codificación y empaquetado residen en un microcontrolador PIC16F873A. En este prototipo se acopló una antena de $\frac{1}{4}$ de onda obteniendo alcance en línea de vista directa de aproximadamente 92 m. Para mayores distancias se tiene que agregar una etapa de potencia. Los resultados demuestran la factibilidad del diseño del transceiver.

SUMMARY

It was made the design and implementation of transceiver of data as propose of solution to the link remote in RF of the digital seismic stations with a central station. Using a commercial module, the transceiver DR3000 of RF Monolithics, the interfaces were designed so much in hardware and software that allow a data transmission coming from a seismic recorder. The control programs of the module to transceiver and the algorithms of codification and packed resides in a microcontroller PIC16F87A. In this prototype an antenna of $\frac{1}{4}$ of wave was matched, and obtained it a link in direct line of approximately 92 m. For greater distances must add a power stage. The results demonstrate the feasibility of the design of the transceiver.

INTRODUCCION

INTRODUCCION

El presente trabajo de tesis se centró en el diseño e implementación de cada una de las interfaces en hardware y software que conforman un transceiver (transceptor) inalámbrico en radiofrecuencia (RF), para ser aplicado al campo de la instrumentación sísmica. El transceiver constituye una solución puntual en la necesidad existente de enlazar estaciones sísmicas digitales remotas con un puesto central de administración de la Red Sismotelemétrica SISMEX, que cuenta con mayor capacidad de procesamiento, análisis y almacenamiento de la señal sísmica.

El transceiver permite interrogar a registradores sísmicos, para verificar y modificar la configuración del registrador sísmico y así poder descargar sus archivos que contienen la información sísmica detectada y almacenada en su memoria. Igualmente, el transceiver puede generar el canal de enlace para una transmisión de la señal sísmica en tiempo real a la estación central.

Las señales sísmicas se caracterizan por ser de bajas frecuencias (en el rango de 0.01 a 20 Hz) y configurando el registrador sísmico con una tasa de muestreo de 100 muestras por segundo (mps) es más que suficiente para obtener parámetros de

estudio y análisis de algún evento telúrico procedente de la estación sísmica remota; es decir con una velocidad de enlace de 9600 bits por segundo (bps) es factible la transmisión de la señal sísmica de 3 canales sin pérdida de la señal del lado del receptor o puesto central.

El transceiver inalámbrico en RF, posee cuatro partes importantes: (1) El módulo transceiver, que recibe y transmite los datos binarios en banda base, usando una modulación digital y portadora en el rango de frecuencia de 900 MHz. (2) Una interface RS232C, que nos permite conectar el puerto serial de cualquier registrador sísmico o un dispositivo serial de propósito general. (3) El módulo transmisor, que recibe datos del registrador, los codifica y agrupa dentro de un paquete de datos, y (4) El módulo receptor, que recibe el paquete de datos, verifica un encabezado y revisa si existe algún error, si todo está correcto lo decodifica y lo transmite en formato RS232C al equipo destino.

La implementación ha sido delimitada para un enlace de corto alcance (3 a 60 metros), que proporcione una operación robusta, bajo consumo de potencia a un bajo costo de implementación. Al final se propone parte de la solución de una siguiente fase que puede pasar del prototipo a una versión comercial, donde se agrega la etapa de Potencia, permitiendo una distancia mayor de enlace.

El presente trabajo de tesis se expone en cuatro capítulos y dos apéndices. En el Capítulo I, se describe la concepción del tema de tesis, es decir el planteamiento del problema y una estrategia de trabajo con objetivo de solucionar el problema.

En el Capítulo II, se describe primeramente el dispositivo de comunicación en RF utilizado en el diseño. Teniendo como base este medio de enlace y el puerto de comunicaciones RS232 del otro extremo se diseña cada una de las partes necesarias para los módulos transmisor y receptor que componen el transceiver inalámbrico en RF.

En el Capítulo III, se reportan las pruebas iniciales de los módulos de comunicación en RF y su integración con un microcontrolador, formando un transmisor y receptor. Las

pruebas iniciales de los módulos transmisor y receptor se realizaron conectando ambos módulos por medio de un cable, esto logro depurar muchas rutinas implementadas en los microcontroladores. También se muestran los resultados obtenidos conectando un registrador sísmico.

En el Capítulo IV, se mencionan las conclusiones y recomendaciones para personas que sigan en este campo del conocimiento de comunicaciones inalámbricas aplicadas al campo de Sismología..

En los Apéndices se resume información del Microcontrolador PIC16F873, así como las rutinas importantes que conforman las interfaces en software de los módulos transmisor y receptor que conforman el transceiver.

CAPITULO I

Planteamiento del Problema

1 PLANTEAMIENTO DEL PROBLEMA

1.1 Antecedentes

Este trabajo de tesis constituye una propuesta de solución a una necesidad de comunicación en radiofrecuencia (RF) con registradores sísmicos digitales remotos, pudiendo también ser utilizado con otros tipos de dispositivos seriales de propósito general. El proyecto se consolidó durante mi estancia en la Coordinación de Ingeniería Sismológica del Instituto de Ingeniería de la Universidad Nacional Autónoma de México. Institución que administra desde 1973 una red sísmica, denominada SISMEEX, que se encarga de monitorear la actividad sísmica del Valle de México ^[1]. Actualmente, esta red ya quedó rezagada tecnológicamente debido a su diseño original, basada en electrónica analógica. Los actuales registradores sismológicos digitales son incompatibles con los dispositivos de comunicación de enlace en RF que actualmente posee SISMEEX, por ello este proyecto de tesis se originó como una propuesta de solución que contempla la modernización de SISMEEX en la etapa de comunicación en RF.

La Red SISMEX está compuesta de nueve estaciones remotas ubicadas estratégicamente en diferentes puntos del Valle de México, las estaciones remotas se hayan enlazadas a la estación central del Instituto de Ingeniería a través de radios analógicos en configuración transmisor-receptor, es decir, con una comunicación unidireccional en la banda de los 470 MHz. Estos radios proporcionan canales de voz (100Hz a 4000Hz), y usan modulación FM para poder enviar la información sísmica hacia el puesto central de la red ^[1], tal como se muestra en la Fig. 1.1.1.



Fig. 1.1.1 Estaciones Remotas de la Red SISMEX ^[1]

No obstante, actualmente la Coordinación de Ingeniería Sismológica cuenta con registradores sismológicos digitales, dispositivos electrónicos que cuentan con interfaces distintas y no pueden ser integrados de manera óptima y eficiente a la actual Red SISMEX. Estos nuevos registradores sismológicos pueden ser configurados para realizar un funcionamiento autónomo fuera de línea o en el mejor de los casos configurar un sistema de comunicación en tiempo real hacia una estación central de registro y control. Actualmente, la mayoría de registradores sismológicos comerciales proporcionan un puerto de comunicaciones RS232, canal de comunicación por donde podemos transmitir la información de los datos digitales mediante un dispositivo de comunicación inalámbrico digital denominado TRANSCEIVER.

Tal como se muestra en la Fig. 1.1.2, la configuración típica de una estación sismológica remota moderna, se compone de un registrador sismológico digital, sensores uniaxiales ó triaxiales, módulo de energía y la referencia precisa de tiempo de un GPS. Se le denomina estación remota, porque se encuentra instalado en lugares inhóspitos para evitar registrar el ruido generado por el hombre.

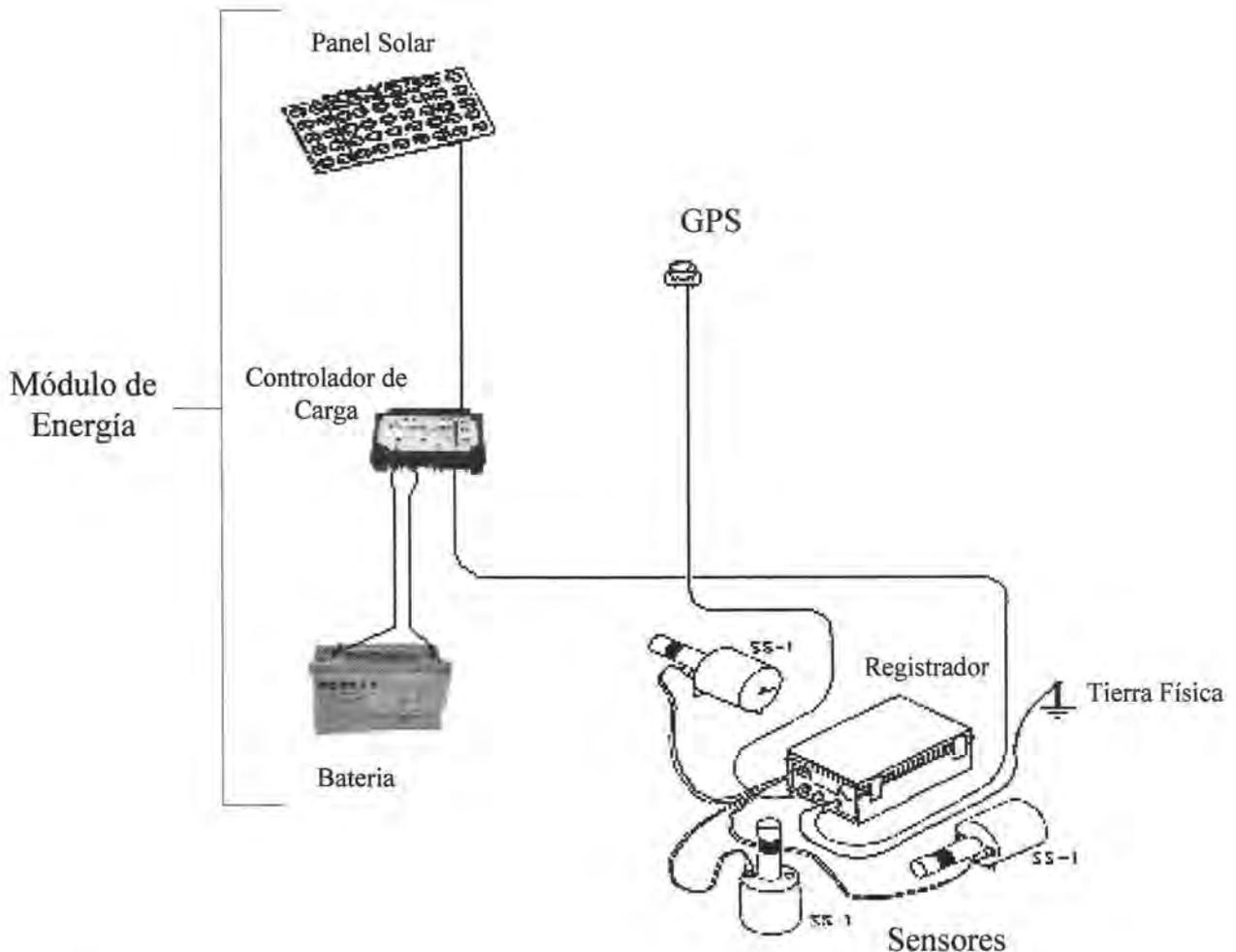


Fig. 1.1.2 Configuración de una Estación Sismológica Remota (Estación Remota)

La estación remota esta configurada en el modo de funcionamiento autónomo, es decir, el registrador cuenta con un sistema de alimentación, detección, adquisición y un algoritmo de disparo para el almacenamiento de los eventos sísmicos en su memoria interna. Por tanto, para realizar el análisis de los eventos sísmicos, es

necesario trasladarse a la estación remota y descargar los datos contenidos en la memoria del registrador, evitando de este modo la saturación de la memoria de almacenamiento del dispositivo. Pero, como mencionamos anteriormente estos registradores sismológicos tienen la capacidad de ser enlazados a una central de registro, agregando a esta estación sismológica remota una opción de comunicación, en nuestro caso particular, un equipo de comunicación en RF a través del puerto de comunicaciones RS232. El equipo de comunicación permite monitorear la estación remota desde un lugar central de registro, afinar los parámetros de funcionamiento del registrador, así como también permite descargar los datos de los eventos sismológicos, sin tener la necesidad de estar trasladándose a la estación remota, lo que constituye un atractivo en sus costos comparado con otras tecnologías como telefonía celular o satelital.

Aprovechando esta experiencia en la instrumentación sísmica tanto en la Red SISMEX, así como en la manipulación de registradores sismológicos actuales completamente digitales, concluimos que era viable la implementación de un transceiver personalizado que cubra con nuestras necesidades de comunicación en RF, y sea una propuesta de desarrollo tecnológico para la red SISMEX.

Los avances tecnológicos en las diferentes áreas de la ingeniería propician perspectivas viables de desarrollos tecnológicos, es por ello que se pensaron en mejoras en el sistema de la Red SISMEX con el objetivo de integrar a la red los registradores sismológicos digitales. Los desarrollos tecnológicos en países como Latinoamérica son factibles, sobre todo que la industria de las comunicaciones inalámbricas en radiofrecuencia ha sido impulsada por el perfeccionamiento en la fabricación de circuitos digitales en radiofrecuencia, integración de circuitos a gran escala, y otras técnicas de miniturización, obteniéndose módulos pequeños de radiofrecuencia que generan el canal de comunicación con un gran ancho de banda en aplicaciones de corto o gran alcance. Estos módulos de radiofrecuencia baratos y muy confiables, se vienen comercializando libremente para ser usados por los diseñadores, cuentan con líneas de entrada y salida de datos digitales en banda base, es decir, son dispositivos con arquitectura abierta, característica que puede ser

aprovechada en la integración con otros módulos de control para el diseño personalizado de sistemas de comunicación inalámbrico en radiofrecuencia.

1.2 Panorama Actual

En los inicios de las comunicaciones inalámbricas en RF se transmitía diversos tipos de información analógica, tal como voz, video, etc. Posteriormente las diversas fuentes de información se fueron digitalizando impulsadas por la mejora de la información y su procesamiento digital. Actualmente se está convirtiendo en un estándar el transmitir información digital.

Continuamente están apareciendo diversas áreas de interés que usan tecnologías de comunicaciones inalámbricas digitales en RF, es decir las comunicaciones inalámbricas en RF están en pleno auge, llevados de la mano de la investigación con intereses en aplicaciones comerciales. El avance tecnológico en esta área conllevó grandes perspectivas en aplicaciones de enlace en corto o gran alcance, incrementando considerablemente su ancho de banda y seguridad, tan comparable como las redes cableadas. La diversificación de las aplicaciones es motivada por un mercado competitivo, estos productos comerciales los podemos encontrar en sistemas de seguridad para hogares, basados en video y voz inalámbricos, redes LANs inalámbricas, telefonía celular, sistemas de comunicaciones inalámbricas personales, etc. Además, actualmente se está implementando la siguiente generación en sistemas digitales de comunicación, que permitirá la convergencia de los servicios, es decir realizar aplicaciones que involucren voz, video y datos digitales a altas velocidades (servicios multimedia en terminales móviles).

Un aspecto importante que nos permite implementar un transceiver inalámbrico en RF es la fabricación y comercialización de circuitos integrados orientados a generar un canal de comunicación en RF, con una arquitectura abierta y pines digitales en banda base, de fácil integración en nuestras aplicaciones particulares. Estos circuitos integrados cada vez más mejorados y a un menor costo, proporcionan un valor alto en la frecuencia de la portadora, tamaño más pequeño, aunado a su creciente ancho de banda y mejora en reducción del ruido existente en el espacio aéreo.

Existen productos comerciales que proporcionan un enlace de comunicación en RF con interface RS-232C, siendo compatibles en la integración con registradores digitales remotos, formando un enlace con la estación remota, con el objetivo de formar una red sismotelemétrica administradas con una computadora central, generando una Red Sísmica. Las ventajas de un enlace inalámbrico son innumerables, resaltando que se puede interrogar al instrumento, configurar sus parámetros de funcionamiento y descargar sus datos almacenados.

Debemos considerar las características mas importantes para el campo sismológico a considerar son: potencia de transmisión, distancia de enlace, frecuencia de portadora, modos de transmisión half-duplex y full-duplex, velocidades de comunicación; y sobre todo que sean compatibles con la alimentación de la batería de 12 voltios y la interface RS-232C de tres hilos del registrador sismológico.

1.3 Planteamiento del Problema

La instrumentación sismológica en México se enfrenta a una incertidumbre cuando se decide conectar estaciones remotas con una estación central para formar una red sismotelemétrica. Esta incertidumbre es generada por la poca asesoría recibida de parte de los distribuidores de equipos comerciales aunado a un mayor problema que es el bajo presupuesto de implementación y mantenimiento del sistema de comunicación en RF. Razón por la cual existen en México estaciones remotas autónomas y sin medio de comunicación, tal como ocurre con la Red Acelerográfica de la Ciudad de México ^[2].

SISMEX no es ajeno a esta realidad, y este trabajo de tesis plantea una solución viable, que es el diseño e implementación de un transceiver inalámbrico de datos en RF; generando así, nuestra propia tecnología en comunicaciones de RF, logrando implementar una red sismotelemétrica personalizada a nuestras necesidades y con bajo costo de implementación, de mantenimiento preventivo y correctivo de nuestra red sísmica, tal como se muestra en la Fig. 1.3.1. Cabe recalcar que el planteamiento de solución al problema no incluye el programa de adquisición y administración de la

Red Sísmica, solo se limita a generar una comunicación en RF desde el puesto central con cada una de las estaciones remotas de manera transparente con una interface serial RS232, que lo poseen tanto el registrador sísmológico como la PC de la estación central.

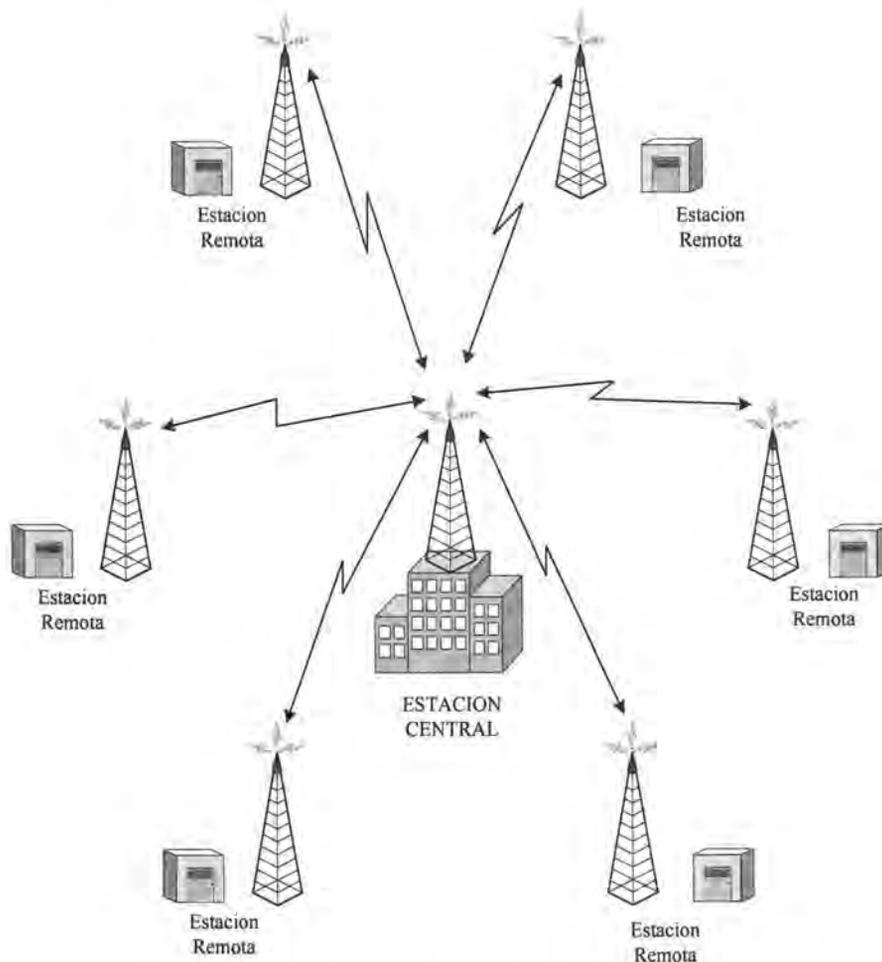


Fig. 1.3.1 Enlace en Radio Frecuencia de una Red Sísmotelemétrica

1.4 Propuesta de solución al problema

Este proyecto de tesis tiene como objetivo el evaluar, diseñar e implementar un transceiver inalámbrico de datos en RF, teniendo como propósito enlazar registradores sísmológicos remotos con una estación central, tal como se muestra en la Fig. 1.3.1. En la etapa inicial, se evaluará los diversos dispositivos comerciales que podrían servir en la implementación del transceiver. Luego de definir el módulo RF con

el cual trabajar se diseñarán las interfaces necesarias para acoplar cada uno de sus puertos o terminales con un microcontrolador PIC. Y finalmente planear el diseño de los programas y rutinas de control que residirán en el microcontrolador, para cada uno de los módulos del transceiver como se describe en la Fig. 1.4.1, tanto para el transmisor como para el receptor.

Utilizando diferentes herramientas en software y dispositivos electrónicos existentes en el mercado a un costo bastante razonable y con arquitecturas abiertas a la integración, es factible el diseñar un transceiver inalámbrico de datos en RF para redes sismológicas. En redes sismológicas se requiere una velocidad de transferencia relativamente baja, es decir, para un registrador sísmico de 24 bits, configurado a 100 muestras por segundo (mps) de sus tres canales, es suficiente configurar su velocidad serial a 9600 bits por segundo (bps). Es decir, podemos implementar sin ningún inconveniente diversos transceivers en RF con características similares a los productos comerciales, pero a un bajo costo de implementación y con un funcionamiento personalizado acorde a nuestras necesidades. Además podemos reducir drásticamente los tiempos de mantenimiento correctivo de los enlaces de comunicación.

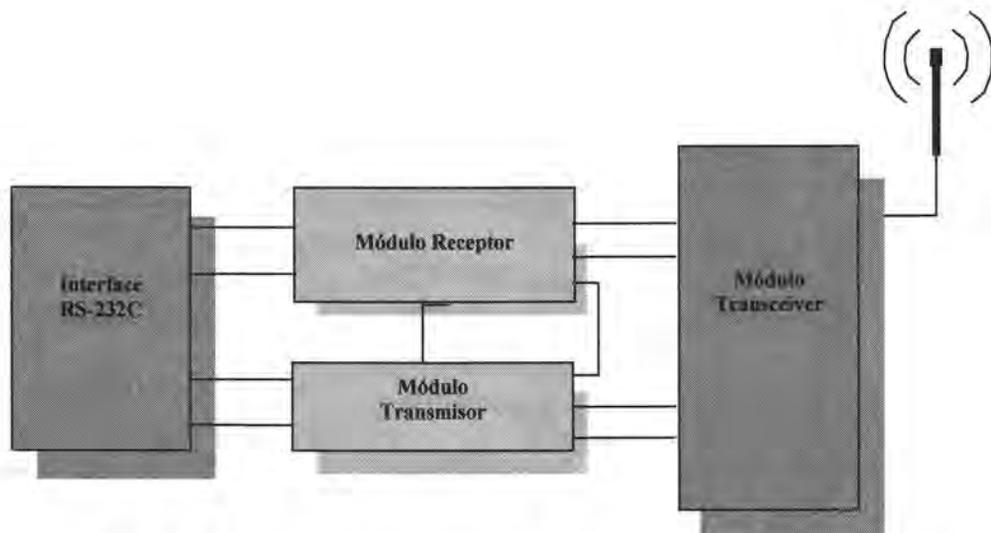


Fig. 1.4.1 Diagrama de Bloques del Transceiver Inalámbrico

La implementación del diseño en el trabajo de tesis ha sido delimitada para un enlace de corto alcance entre unos 10 a 60 metros, que proporcione una operación robusta, bajo consumo de potencia. Para que en su siguiente fase de implementación comercial se agregue la etapa de Potencia, permitiendo alcanzar una distancia mayor de enlace.

Como se muestra en la Fig. 1.4.1, el transceiver inalámbrico de datos en RF propuesto, está dividido en cuatro partes importantes:

- 1) **La interface RS-232C** que nos permite conectar al puerto serial RS-232 de cualquier registrador o dispositivo serial con de propósito general. Este módulo es transparente para los demás módulos, sólo se requiere configurar los parámetros que fijan el formato de datos, su velocidad y protocolo del puerto, los que deben coincidir con los parámetros del registrador o dispositivo serial a conectar.
- 2) **El módulo Receptor** recibe los paquetes de datos provenientes del Módulo Transceiver, verifica que es un paquete para tal unidad y chequea si existe algún error para luego desempaquetarlo y enviar los datos en formato RS-232C al dispositivo destino.
- 3) **El módulo Transmisor** recibe datos del registrador conectado al puerto RS-232, los codifica y agrupa en un paquete, agrega una cabecera y genera un código de chequeo de error para que sea procesada por el módulo receptor de otro transceiver.
- 4) **El módulo Transceiver** recibe y transmite los datos binarios, usando una modulación digital con una portadora en la frecuencia de 916.5 MHz.

En la Fig. 1.4.2, se presenta un diagrama de bloques de una de las implementaciones que nos permitirán evaluar el diseño en hardware y sobre todo el software programado en cada uno de los microcontroladores que conforman los módulos transmisor y receptor. Se usará una computadora en primera instancia en lugar de los registradores o algún dispositivo serial, porque nos permitirá cambiar, afinar y

visualizar el funcionamiento de nuestros algoritmos de control y funcionamiento de cada uno de los módulos transmisor y receptor de un transceiver.

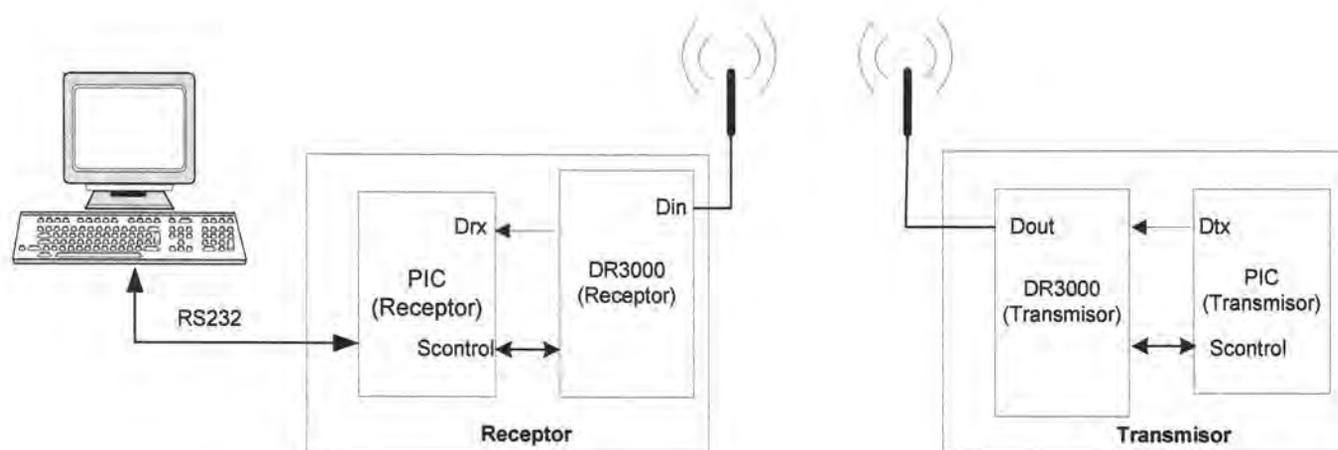


Fig. 1.4.2 Módulo de Evaluación del Transceiver inalámbrico de Datos

CAPITULO II

Diseño e Implementación del Transceiver Inalámbrico de Datos en RF

2 DISEÑO E IMPLEMENTACION DEL TRANSCEIVER INALAMBRICO DE DATOS EN RF

2.1 Transceiver Inalámbrico de Datos en RF

2.1.1 Descripción General

El avance en las comunicaciones inalámbricas establece nuevos requisitos para los módulos transceivers de Comunicación Inalámbrica de Datos en RF. Es decir, las frecuencias de operación son cada vez más altas, bajo consumo de potencia y un alto grado de integración, requiriendo para su implementación el diseño con técnicas muy diferentes a las técnicas clásicas de diseño en RF. Pero hay que recalcar que existiendo un alto grado de integración y la notable reducción del consumo de potencia que requieren los componentes digitales, aún, se están mejorando las técnicas que reduce las limitaciones físicas de la topología de los transceiver, generando un mejor acoplamiento de la parte digital con la parte analógica en radiofrecuencia del puerto de la antena. La tecnología CMOS ayuda a diseñar multicaminos de tecnologías del transmisor y receptor que conforma el transceiver, lográndose transceivers con frecuencias de operación en el rango de 1 a 2 GHz.

Entre las ventajas en el uso de estos módulos de comunicación en RF, podemos mencionar:

- Se requiere un conocimiento básico y poca experiencia en RF.
- No se requieren de equipos de medida y prueba que sean complejos y costosos.
- Son viables en producciones a pequeña escala.
- Interfaces simples dentro de los diseños.
- Bajos costos.
- No requiere procesos de sintonización.
- Reducción significativa del tiempo de desarrollo y comercialización.
- Ideal para probar conceptos de comunicaciones inalámbricas.

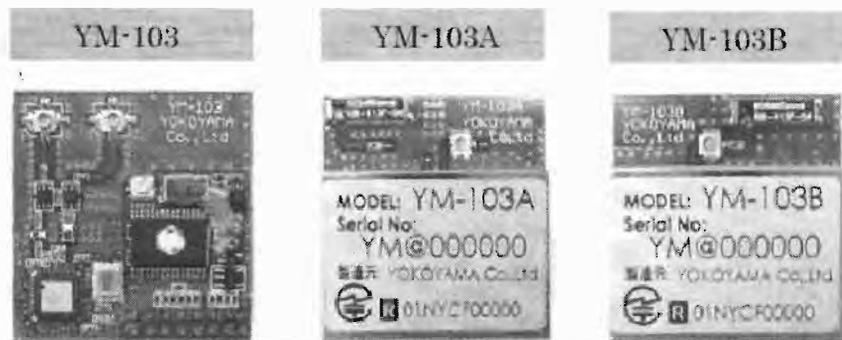
2.1.2 Principales Módulos de Comunicaciones en RF en el Mercado Actual

En el mercado actual existen diversas empresas que producen módulos de comunicación en radiofrecuencia, entre las más destacadas podemos mencionar:

- A la empresa japonesa YOKOYAMA ^[3], que produce módulos de comunicación inalámbrica a la frecuencia de 2.4 GHz Spread Spectrum Radio, que son sistemas PLL controlados con VCO, con una velocidad máxima de 160 kbps, con un alcance corto (20 – 100 metros) y con 41 canales. Estos productos son interesantes por su tecnología Spread Spectrum y vienen integrados con interfaces para modem o RS232C en un tamaño reducido, tal como se observa en la Fig. 2.1.2.1, facilitando la integración en equipos y sistemas remotos con estas interfaces estándares.
- A la empresa CHIPCON ^[4], que produce módulos de comunicación inalámbrica con rangos de frecuencia inferiores a 1 GHz y en el rango de 2.4 GHz. Con características sobresalientes de poder programar su frecuencia de operación, bajo consumo de potencia, y soportan velocidades desde 9.6 kbps hasta 153.6 kbps en frecuencias inferiores a 1 GHz y hasta velocidades desde 153.6 a 1 Mbps en frecuencias en el rango de 2.4 GHz. Además

tienen modelos que incluyen memoria buffer en su canal de transmisión y recepción.

- A la empresa Canadiense ABACOM Technologies [5]. Poseen módulos Transceivers FM, para comunicaciones half-duplex que soportan velocidades desde 9.6 kbps hasta 100 kbps, con un alcance de enlace hasta 220 metros. Con salida de detección de portadora, e interface directa a lógica CMOS. Tiene versiones a 433.92 MHz y 914.5 MHz. Además de módulos transceivers AM de hasta 4800 bps.



(a)



(b)

Fig. 2.1.2.1 Productos en módulos de RF de (a) Yokohama [3] y (b) Abacom [5]

- La Compañía TOSHIBA ^[6]. Que oferta módulos transceivers en el rango de frecuencia de 240 a 450 MHz, con alta sensibilidad de calidad industrial, proporciona una salida bastante limpia, que evita necesidad de filtrado adicional, capacidades de comunicación AM y FM, requiere pocos componentes externos.

2.1.3 Transceiver Inalámbrico Híbrido de Amplificador Secuenciado (HAS)

El transceiver HAS TR1000, es un dispositivo producido por la empresa RF Monolithic (RFM) ^[7], fue diseñado específicamente para aplicaciones de enlace inalámbricos en RF de corto alcance. Las funciones críticas de RF están contenidas en el híbrido, que simplifica y agiliza su integración en alguna aplicación de interés particular, en este caso un transceiver digital que permita enlazar dos puertos RS-232C de tres hilos, de forma inalámbrica en RF.

Como se mencionó en la sección 1.4, se implementará el transceiver inalámbrico de datos en RF, conformado por cada una de las etapas que se muestra en el diagrama de bloques de la Fig. 1.4.1. Empezaremos por describir el módulo del transceiver en RF, porque es el corazón de todo el diseño, encargado de generar el canal de comunicación y en base a éste módulo se diseñaron las etapas de transmisión y recepción.

2.1.3.1. Teoría de Operación

El conjunto de características únicas del transceiver HAS es posible debido a la arquitectura de su sistema, tal como lo muestra el diagrama de bloques en la Fig. 2.1.3.1.1. La parte esencial del transceiver es la sección del receptor con amplificador-secuenciado, que proporciona una ganancia de detección estable de más de 90 dB, sin ningún blindado o desacoplamiento especial. La estabilidad es lograda distribuyendo la ganancia total de RF sobre el tiempo ^[8].

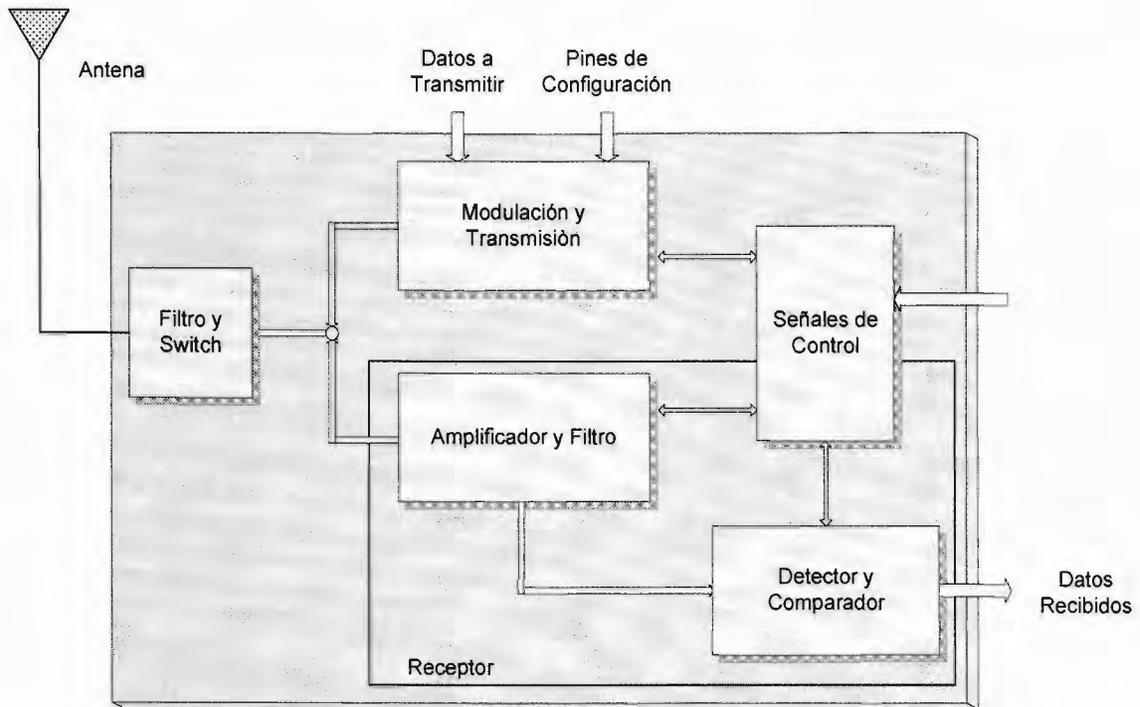


Fig. 2.1.3.1.1 Diagrama de bloques del Transceiver Inalámbrico HAS

2.1.3.2. Flujo de la Señal en el Receptor

Como se muestra en la Fig. 2.1.3.2.1, la señal RF entrante primeramente pasa a través del filtro de onda acústica superficial (SAW) de banda angosta para luego ser aplicada al Amplificador de Radiofrecuencia 1 (RFA1). Un generador de pulsos enciende al RFA1 por $0.5 \mu\text{s}$. La señal amplificada del RFA1 pasa por la línea de retardo SAW de $0.5 \mu\text{s}$, para luego ingresar al Amplificador de Radiofrecuencia 2 (RFA2). RFA1 es apagada y RFA2 es encendida por $0.55 \mu\text{s}$, amplificando un poco más la señal de RF. El bloque Generador de Pulsos & Alimentación de Amplificadores de RF, enciende y apaga independientemente a RFA1 y RFA2. Un efecto del filtrado de la línea de retardo SAW es el ensanchamiento del pulso de señal de RFA1. Como se muestra en el diagrama de tiempos de la Fig. 2.1.3.2.1, RFA1 y RFA2 nunca están encendidos en el mismo tiempo, asegurando una excelente estabilidad del receptor [7].

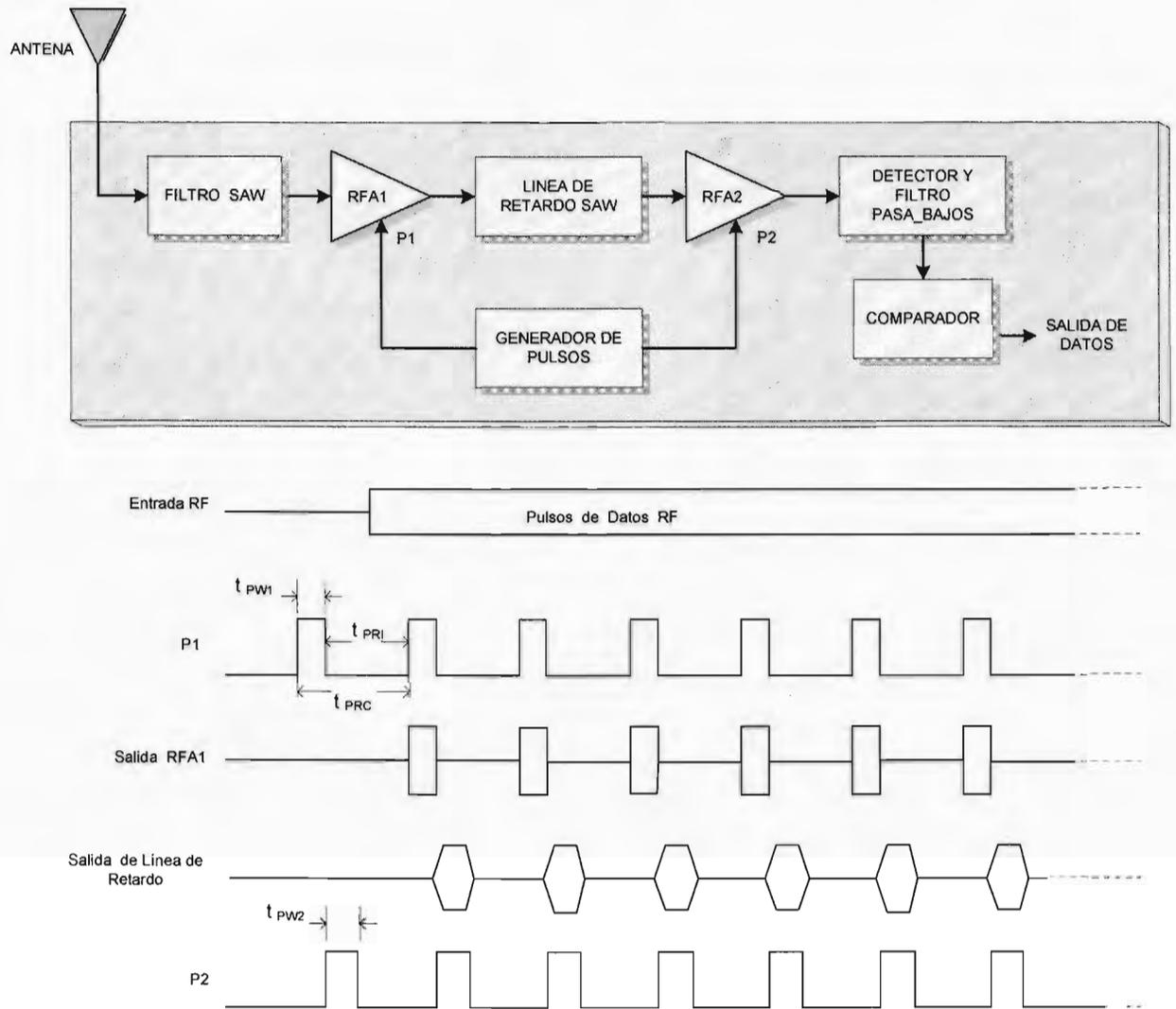


Fig. 2.1.3.2.1 Diagrama de bloques del Receptor HAS & Diagrama de Tiempos [8]

Dado que un receptor con amplificador-secuenciado es intrínsecamente un receptor de muestreo, el ciclo de tiempo total entre el inicio de una secuencia de encendido del RFA1 y el inicio de la siguiente secuencia de encendido del RFA1 debe ser fijado para muestrear al menos 10 veces el pulso más angosto de los datos en RF.

Los amplificadores RFA1 y RFA2 detectan el inicio de la saturación cambiando la ganancia a su valor mas bajo El rechazo combinado del filtro SAW RF y la línea de retardo SAW proporcionan un excelente rechazo global del receptor [7]. Se logra aproximadamente 100 dB de rango dinámico en el receptor [7].

La salida analógica del filtro pasa bajos es amplificada e ingresada al módulo Comparador que la transforma en una cadena de datos digitales, además de proporcionar mejor desempeño en condiciones de señales con ruido. El umbral del módulo comparador permite una compensar la sensibilidad del receptor y la densidad del ruido de salida en la condición en que no exista señal. Se debe fijar el umbral a 0, para mayor sensibilidad. Es necesario considerar una interface en hardware o software para procesar este ruido presente continuamente en el receptor. Podemos incrementar el umbral pero a costa de perder sensibilidad y evitar falsos disparos o glitches de datos en el pin del procesamiento de la señal recibida. Así, un umbral de 25 mV proporciona una buena compensación con los glitches y baja sensibilidad para un ancho de banda de 48 KHz (datos a 19.2 kpbs NRZ). Considerar la siguiente expresión para calcular el umbral a un diferente ancho de banda [7].

$$V = 3.6 * (f_{LPF})^{1/2} \quad (\text{Ec. 2.1})$$

donde V está en mV y f_{LPF} está en KHz

2.1.3.3. Flujo de la Señal en el Transmisor

El transmisor consiste de un oscilador de línea de retardo SAW seguido por un buffer amplificador modulado. El filtro SAW suprime los armónicos transmitidos a la antena. Notamos en la Fig. 2.1.3.3.1, que algunos dispositivos SAW usados en el amplificador secuenciado del receptor son reutilizados por el transmisor.

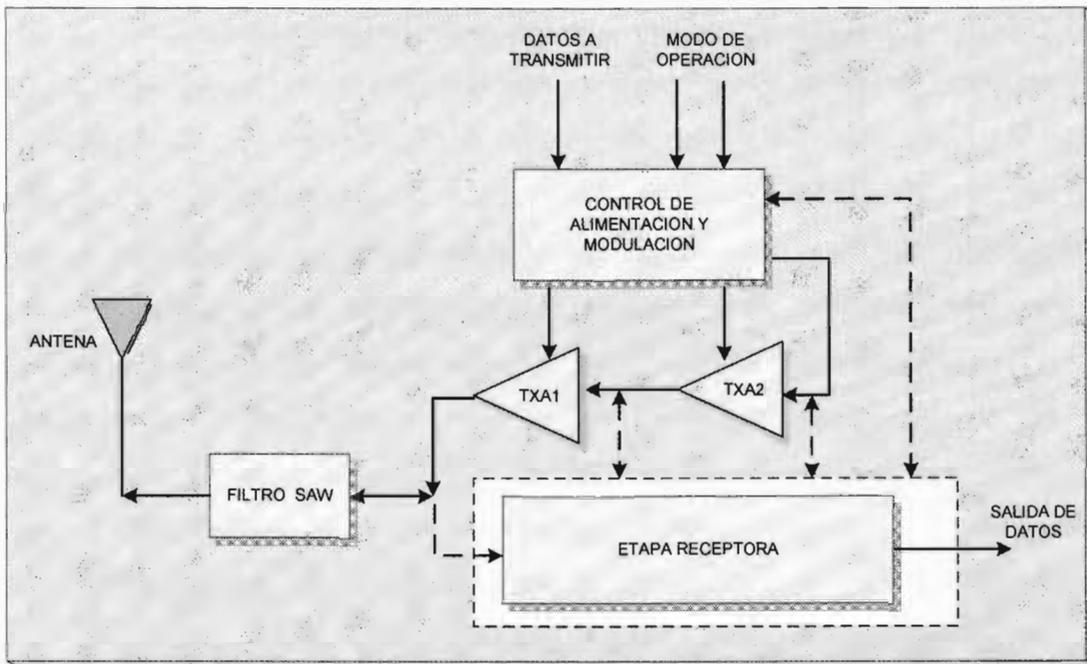


Fig. 2.1.3.2.1 Diagrama de bloques del Transmisor HAS [8]

El transmisor soporta dos formatos de modulación, modulación encendido-apagado (OOK), y la modulación de desplazamiento en amplitud (ASK). Cuando se selecciona la modulación OOK, la salida del transmisor se apaga completamente entre pulsos de datos "1". Cuando se selecciona la modulación ASK, un pulso "1" es representado por la transmisión de un nivel alto de potencia, y un "0" es representado por la transmisión de un nivel bajo de potencia. La modulación OOK provee ahorro de potencia. La modulación ASK debe ser usada para datos de alta velocidad (pulsos de datos de menos de 30 μ s). La modulación ASK también reduce los efectos de algunos tipos de interferencia y permite que los pulsos transmitidos controlen el ancho de banda de modulación.

El transceiver tiene 4 modos de operación, y es controlado por el bloque Modulation&Bias Control, teniendo como entradas los pines del control CNTRL0 y CNTRL1, que configuran el modo de operación del transceiver, tal como lo muestra la Tabla 2.1.3.3.1. CNTRL1 y CNTRL0 son entradas compatibles CMOS. Estas entradas siempre deben estar conectadas a un estado lógico [7].

Tabla 2.1.3.3.1 Modos de Operación del Transceiver

CNTRL0	CNTRL1	Función
1	1	Modo Receptor
0	1	Modo Transmisor ASK
1	0	Modo Transmisor OOK
0	0	Modo Baja potencia ("sleep")

Cuando se selecciona un tipo de modulación, los RFAs del receptor son apagados. En el modo OOK, el amplificador oscilador de línea de retardo TXA1 y el amplificador de buffer de salida TXA2 son apagados cuando el voltaje de entrada en el pin TXMOD es menor de 220 mV. En el modo OOK, la velocidad de los datos es limitada por los tiempos de encendido y apagado del oscilador de la línea de retardo. En el modo ASK, TXA1 está continuamente encendida, y la salida de TXA2 es modulada por la corriente de entrada del pin TXMOD.

La potencia de salida del transmisor RF es proporcional a la corriente de entrada sobre el pin TXMOD. Una resistencia en serie con el pin TXMOD ajusta el pico de la potencia del transmisor. La máxima potencia de salida requiere una corriente de entrada entre 300 y 450 μ A.

2.1.3.4. Requerimientos de Potencia

VCC1 proporciona voltaje de alimentación para la salida del amplificador del transmisor y para la circuitería del receptor en banda base. VCC1 debe ser conectado a través de una ferrita de desacoplamiento para RF que es desviado por un capacitor de RF del lado de la fuente de alimentación. VCC2 proporciona voltaje de alimentación positiva para la sección del receptor en RF y el oscilador del transmisor. VCC2 debe ser desviado con un capacitor en RF (puede ser un capacitor de tantalio de 10 μ F) y aislado de la fuente positiva por medio de un resistor de 100 ohm. El rango del voltaje de la fuente de alimentación para su funcionamiento esta entre 2.7 y 3.5 Vdc. El rizo de la

fuentes de alimentación debe ser menor que 10 mVpp. Los requerimientos en el desacoplamiento de la fuente de alimentación son menos rigurosos cuando solo se utiliza el transceiver como receptor.

El transceiver funciona con datos de baja velocidad en un rango limitado de temperatura (-10 °C a 85°C) con una fuente de voltaje debajo de 2.5 Vdc. Se recomienda una velocidad máxima de 9600 bps (mínimo ancho de pulso de señal de 104 μ s) para su funcionamiento en bajo voltaje. El pico de la potencia transmitida varía aproximadamente 2 dB en el rango de fuente de voltaje de 3.5 a 2.5 Vdc, es decir se genera menos potencia a un voltaje de 2.5 Vdc [7].

2.1.3.5. La Antena - Puerto de Entrada/Salida en RF

El pin de Entrada/Salida en RF del transceiver, RFIO, permite conectar a la entrada del filtro SAW y debe ser acoplado a la impedancia de la antena para una correcta operación del transceiver.

Los elementos externos en RF necesarios para el transceiver son la antena y componentes de acoplamiento. La antena debe presentar una impedancia en el rango de 35 a 72 Ω para poder ser acopladas al pin RFIO por medio de una bobina en serie y una de desacoplamiento tal como se muestra en la Fig. 2.1.3.3.1, logrando una buena protección de descarga electrostática - ESD, requiriendo un camino de retorno a tierra del pin RFIO, usando un choque RF de desviación al pin GND1 o al pin GND3. Para agregar más protección ESD, las antenas montadas externamente deben tener una cubierta de aislamiento.

Para una impedancia de antena de 50 Ω se requiere una inductancia en serie de 10 nH y una inductancia de desviación de 100 nH. El choque ESD debe formar parte del circuito de acoplamiento de la antena. Es recomendable combinar las bobinas que acoplen 50 ohm a la entrada del transceiver con el circuito de acoplamiento de la antena para reducir el número de componentes.

2.1.3.6 El Generador de Pulsos

Como se mencionó anteriormente, los RFAs del receptor son controlados por el bloque Pulse Generator & RF Amplifier Bias, que a su vez son controlados por los pines de entrada PRATE, PWIDTH y la señal Power Down Control proveniente del bloque Modulation & Bias Control. Se desactivan los RFAs del receptor por medio de la señal Power Down Control, que es generada en los modos de transmisión y baja potencia.

Velocidad y Ancho de los Pulsos

EL generador de pulsos tiene dos modos de operación, en aplicaciones de baja velocidad de datos (baja corriente) y en aplicaciones de alta velocidad de datos (alta sensibilidad).

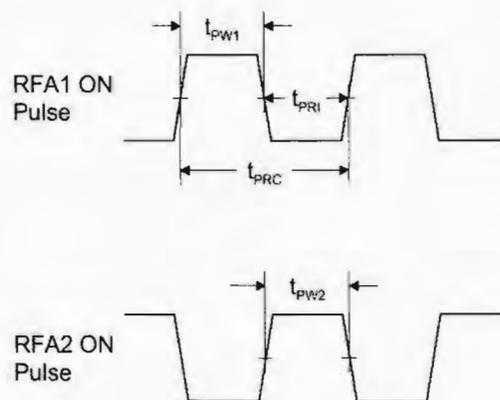


Fig. 2.1.3.6.1 Diagrama de Tiempos del Generador de Pulsos [8]

Velocidad Baja de Datos

Haremos referencia a la Fig. 2.1.3.6.1. El intervalo de tiempo t_{PRI} es fijado por la resistencia PRATE (R_{PR}). El intervalo t_{PRI} puede ser ajustado entre 0.1 a 5 μs con la resistencia R_{PR} en el rango de 51 $K\Omega$ hasta 2000 $K\Omega$. El valor de R_{PR} es dado por la ecuación (Ec.) 2.2, que se muestra a continuación [7].

$$R_{PR} = 404 * t_{PRI} + 10.5 \quad (\text{Ec. 2.2})$$

donde t_{PRI} está en μs , y R_{PR} en $K\Omega$.

El intervalo de tiempo t_{PW1} es fijado por la resistencia R_{PW} (t_{PW2} es fijado en 1.1 veces el t_{PW1}). El t_{PW1} puede ser ajustado entre 0.55 y 1 μs con la resistencia R_{PW} en el rango de 200 $K\Omega$ hasta 390 $K\Omega$. El valor de R_{PW} es definido por la Ec. 2.3 [7].

$$R_{PW} = 404 * t_{PW1} - 18.6 \quad (\text{Ec. 2.3})$$

donde t_{PW1} está en μs , y R_{PW} en $K\Omega$.

Se recomienda resistencias con tolerancia en su valor del $\pm 5\%$. Es importante mantener la capacitancia total entre tierra, V_{cc} y este nodo menos de 5 pF para mantener la estabilidad.

Velocidad Alta de Datos

Cuando se conecta el Pin PWIDTH a V_{cc} por medio de una resistencia de 1 $M\Omega$, los RFAs funcionan a un duty cycle del 50%-50%, facilitando su funcionamiento con datos de alta velocidad. En este caso, el intervalo de tiempo t_{PRC} es controlado por la resistencia R_{PR} , dentro del rango de 0.1 a 1.1 μs usando un valor de la resistencia R_{PR} en el rango de 11 $K\Omega$ hasta 220 $K\Omega$. El valor de R_{PR} esta definida por la Ec. 2.4 [7].

$$R_{PR} = 198 * t_{PRC} - 8.51 \quad (\text{Ec. 2.4})$$

donde t_{PRC} está en μs y R_{PR} en $K\Omega$.

Debe usarse una resistencia con una tolerancia en su valor del $\pm 5\%$.

Para señales de pulso con un mínimo ancho entre 8.7 y 17.4 μs , t_{PRC} debe ser fijado a 0.87 μs . Este valor proporciona una razón de muestreo nominal de 10 muestreos para un pulso de señal de 8.7 μs , y se aprovecha del ensanchamiento del pulso a través de una línea de retardo SAW para proporcionar una ganancia RF cercana a la óptima.

2.1.3.7 Filtro Pasa Bajos

El filtro usado por transceiver HAS es de tres polos. Tal como se observa en la Fig. 2.1.3.2.1, la resistencia R_{LPF} permite ajustar el ancho de banda del filtro pasa bajos del receptor. El valor de la resistencia puede oscilar entre 330 $K\Omega$ y 820 Ω , proporcionando

un valor del ancho de banda del filtro, f_{LPF} entre 4.5 KHz y 1.8 MHz. El valor de la resistencia R_{LPF} es determinado por la Ec. 2.5 [7]:

$$R_{LPF} = \frac{1445}{f_{LPF}} \quad (\text{Ec. 2.5})$$

donde R_{LPF} está en $K\Omega$ y f_{LPF} en KHz.

Debe usarse una resistencia con una tolerancia de su valor del $\pm 5\%$, proporcionan un f_{LPF} en el rango de f_{LPF} y $1.3 * f_{LPF}$, con algunas variaciones con el voltaje de la fuente de alimentación, temperatura, etc. El pico de corriente disponible en el pin RXDATA aumenta en proporción con el ancho de banda del filtro.

Selección del Ancho de Banda

Cuando usamos el DS2 o el DS1 sin umbral, se recomienda usar la Ec. 2.6 para encontrar el valor del ancho de banda [7].

$$f_{LPF} = \frac{1000}{SP_{MIN}} \quad (\text{Ec. 2.6})$$

donde f_{LPF} está en KHz y SP_{MIN} es el mínimo ancho del pulso de señal en μs

Para aplicaciones donde se requieran un receptor con mayor sensibilidad, f_{LPF} debe ser reducido a un 75% del valor anterior, así se utilice una codificación Manchester o simbolización de 12 bits.

El ancho de banda recomendado cuando sólo se usa un umbral en DS1 es calculado mediante la Ec. 2.7 [7]:

$$f_{LPF} = \frac{2500}{SP_{MIN}} \quad (\text{Ec. 2.7})$$

2.1.3.8 Acoplamiento en Banda Base

El pin BBOUT proporciona la salida del receptor en banda base. Este pin se conecta al pin CMPIN a través de un capacitor cerámico de acoplamiento, C_{BBO} , que va hacia el DS1 y DS2. La constante de tiempo t_{BBO} para esta conexión se calcula mediante la Ec. 2.8 [7]:

$$t_{BBO} = 0.064 * C_{BBO} \quad (\text{Ec. 2.8})$$

donde t_{BBO} está en μs y C_{BBO} en pF

Se debe usar un C_{BBO} , con una tolerancia en su valor del $\pm 10\%$. La constante de tiempo puede variar entre t_{BBO} y $1.8 * t_{BBO}$, con algunas variaciones con el valor del voltaje de la fuente de alimentación, temperatura, etc.

Cuando el transceiver está en baja potencia o en modo transmisor, la impedancia de salida del Pin BBOUT es muy alta. Esta característica ayuda a conservar la carga del capacitor de acoplamiento para reducir al mínimo el tiempo de estabilización del DS1 y DS2, cuando el transceiver regresa al modo receptor.

El valor correcto del capacitor de acoplamiento en banda base depende del máximo ancho de pulso que puede ocurrir en la señal recibida. El máximo ancho de pulso depende alternadamente, de la cadena de datos codificada, la velocidad de los datos, y la máxima longitud de un nivel lógico que puede ocurrir en la cadena de los datos. Si no se usa alguna codificación, el máximo ancho de un pulso es igual al periodo de bit multiplicada por la máxima longitud. Si se usa la codificación de byte a 12 bits, el máximo ancho de pulso es cuatro veces el periodo de bit. Para una codificación Manchester, el máximo ancho del pulso es un periodo de bit.

La constante de tiempo t_{BBO} se calcula mediante la Ec. 2.9, y debe ser seleccionada de tal manera que no decaiga más del 20 % durante un evento con un ancho de pulso máximo [7], así:

$$t_{BBO} = 4.48 * SP_{MAX} \quad (\text{Ec. 2.9})$$

donde t_{BBC} y el máximo ancho del pulso de la señal, SP_{MAX} están en μs

Despejando C_{BBO} de la Ec. 2.8, tenemos:

$$C_{BBO} = 15.625 * t_{BBC} \quad (\text{Ec. 2.10})$$

donde t_{BBC} está en μs y C_{BBO} está en pF, o

Reemplazando t_{BBC} de la Ec. 2.9 en la Ec. 2.10, tenemos:

$$C_{BBO} = 70 * SP_{MAX} \quad (\text{Ec. 2.11})$$

donde SP_{MAX} está en μs y C_{BBO} está en pF

Se debe respetar un preámbulo del paquete de datos igual a 1.6 veces t_{BBC} , que fija C_{BBO} a un voltaje del 80% del valor óptimo de comparación de los DSs. Usando una codificación Manchester, esto sería un byte. Usando la codificación de byte a 12-bits, esto sería cuatro bytes. Intentando transmitir datos no codificados con SP_{MAX} de 16 bits o más requiere un preámbulo de instrucción grande, impracticable. Esto es una razón importante para la codificación de los datos.

Niveles de Salida de la Señal en Banda Base

El pin BBOOUT puede ser conectado a algún proceso externo de recuperación de datos (DSP, etc.). Cuando los RFAs del receptor están funcionando a un duty cycle de 50%-50%, la señal BBOOUT varía aproximadamente 10 mV/dB, con un nivel de señal de hasta 685 mVpp. Para duty cycles más bajos, la razón mV/dB y el nivel de la señal pico a pico son proporcionalmente menores. La señal detectada está montada sobre un nivel de 1,1 Vdc que varía un poco con la fuente de voltaje, temperatura, etc. La señal del pin BBOOUT es acoplada al pin CMPIN o a un proceso externo de recuperación de datos por medio de un condensador en serie. La impedancia nominal de salida del pin BBOOUT es 1 K Ω . Se recomienda un valor una impedancia de carga entre 50 K Ω y 500 K Ω en paralelo con un condensador de valor no mayor a 10 pF.

Cuando un proceso externo de recuperación de datos es usado con AGC, BBOUT debe ser acoplado a un proceso externo de recuperación de datos y al pin CMPIN por condensadores de acoplamiento en serie independientes. La señal AGC Reset es derivada de la circuitería del bloque Peak Detector que utiliza la señal aplicada al pin CMPIN.

2.1.3.9 Diseño del Transceiver HAS en su Configuración OOK/ASK

El Transceiver HAS puede ser configurado e integrado en diferentes requerimientos de aplicación. La configuración mostrada en la Fig. 2.1.3.9.1 es conveniente para transmisión de datos OOK con un pulso mínimo de 30 μs o datos ASK con un mínimo ancho de pulso de 8.7 μs .

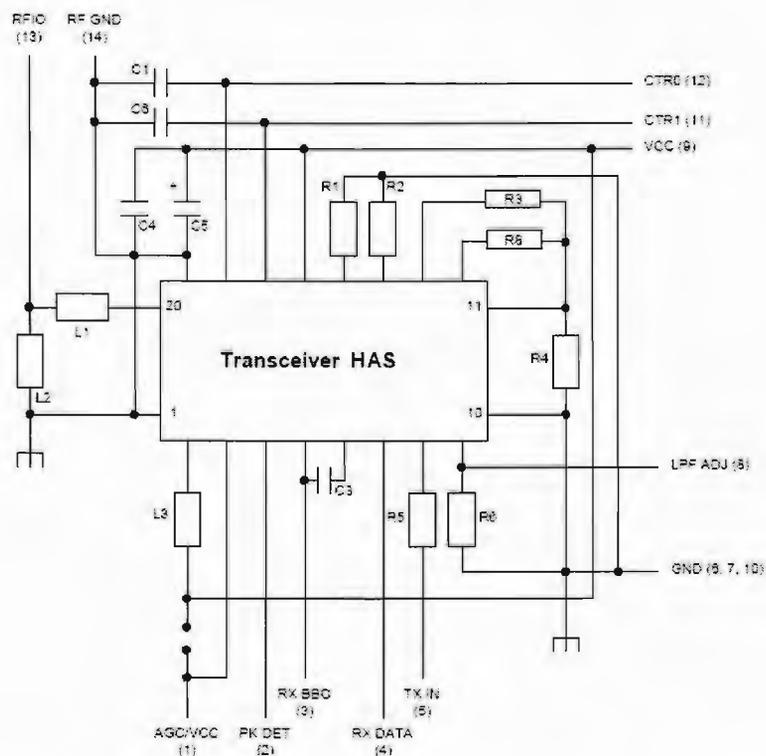


Fig. 2.1.3.9.1 Integrando el TR1000 en un Módulo Transceiver [8]

2.1.3.10 Módulo Transceiver Híbrido DR3000

La Fig. 2.1.3.10.1, muestra la implementación física del Transceiver en la configuración de la Fig. 2.1.3.9.1, particularmente en su modo OOK/ASK, además se muestra la descripción de sus pines para ser integrados en el diseño particular en la Tabla 2.1.3.10.2.

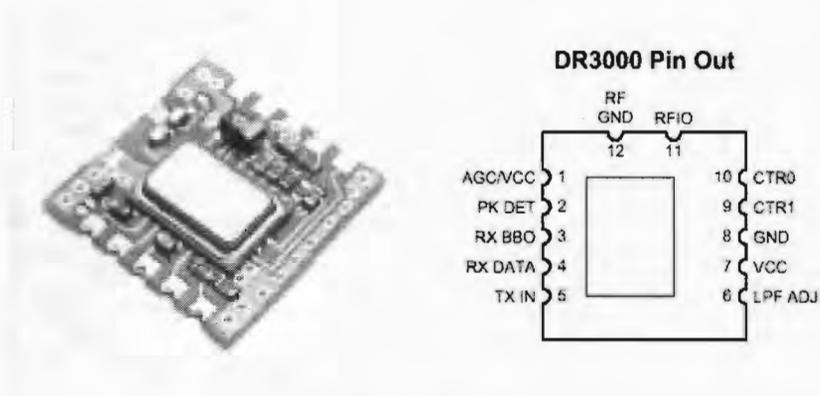


Fig. 2.1.3.10.1 Configuración OOK/ASK y Designación de Pines [7]

Los valores usados en la configuración de la Fig. 2.1.3.9.1 se enumeran en la Tabla 2.1.3.10.1.

Tabla 2.1.3.10.1 Valores de elementos usados en el DR3000

Referencia	Valor
C1, C4, C6	100 pF \pm 10%
C3	0.1 μ F \pm 10%
C5	4.7 μ F \pm 10%
R1	270 K Ω \pm 5%
R2	330 K Ω \pm 5%
R3	10 K Ω \pm 5%
R4	100 K Ω \pm 5%
R5	4.7 K Ω \pm 5%
R6	330 K Ω \pm 5%
L1	10 nH \pm 5%
L2	100 nH \pm 5%

Tabla 2.1.3.10.2 Descripción de pines del Módulo DR3000

Pin	Nombre	Descripción
1	AGC/VCC	Este va conectado directamente al pin AGCCAP del transceiver TR1000, y es deshabilitado si se conecta a VCC
2	PK DET	Este pin controla la operación del pico de detección. Va conectado del pin PKDET del transceiver TR1000
3	RX BBO	Este pin puede ser usado para alimentar a un proceso de recepción de datos externo. La impedancia nominal de salida de este pin es de 1 K. Este pin es el pin BBOUT del transceiver TR1000. Si se modifica el valor de C3 para soportar una mayor velocidad de datos y/o diferente estructura de codificación .
4	RX DATA	Este pin es el pin RXDATA del transceiver TR1000. Este pin puede manejar una carga paralela de 10 pF, 500 K
5	TX IN	Este pin es el pin TXMOD del transceiver TR1000 a través de una resistencia de 4.7 K en el modulo DR3000.
6	LPF ADJ	Este pin es el ajuste del ancho de banda del filtro pasa bajo, es conectado directamente al pin LPFADJ del transceiver TR1000. Una resistencia externa puede ser agregada entre el pin 6 y tierra para soportar mayores velocidades
7	VCC	Es el voltaje de alimentación positivo para el módulo DR3000. El voltaje de operación es entre 2.7 y 3.5 Vdc. También es posible usar el pin 1 como el VCC de entrada.
8	GND	Este el pin de retorno del voltaje de alimentación.
9	CTR1	Este pin junto con CTR0 es uno de los dos pines para seleccionar el modo de operación del transceiver. Este pin debe tener siempre un nivel lógico. Después de alimentar este pin se eleva hasta 2.7 Vdc (modo receptor). Después de esto cualquier modo puede ser seleccionado.
10	CTR0	Este pin junto con TRL1 es uno de los dos pines para seleccionar el

		modo de operación del transceiver. Este pin debe tener siempre un nivel lógico. Después de alimentar este pin se eleva hasta 2.7 Vdc (modo receptor). Después de esto cualquier modo puede ser seleccionado.
11	RFIO	Es el pin de Entrada/Salida de RF. Se implementó un acoplamiento para una carga de 50 ohm (antena).
12	RF GND	Este pin es la tierra (retorno) de RF que es usada en conjunto del pin RFIO.

De manera similar como se indicó en el apartado 2.1.3.3, para seleccionar el Modo de funcionamiento del módulo DR3000 usamos dos terminales, tal como se muestra en la Tabla 2.1.3.10.3.

Tabla 2.1.3.10.3 Modos de Funcionamiento del DR3000

CTR0	CTR1	Modo
0	0	Sleep (stanby)
0	1	Transmisión ASK
1	0	Transmisión OOK
1	1	Receptor

La Fig. 2.1.3.10.2, se muestran dos ejemplos de aplicación del módulo DR3000, para una velocidad de 2.4 kbps y 19.2 kbps.

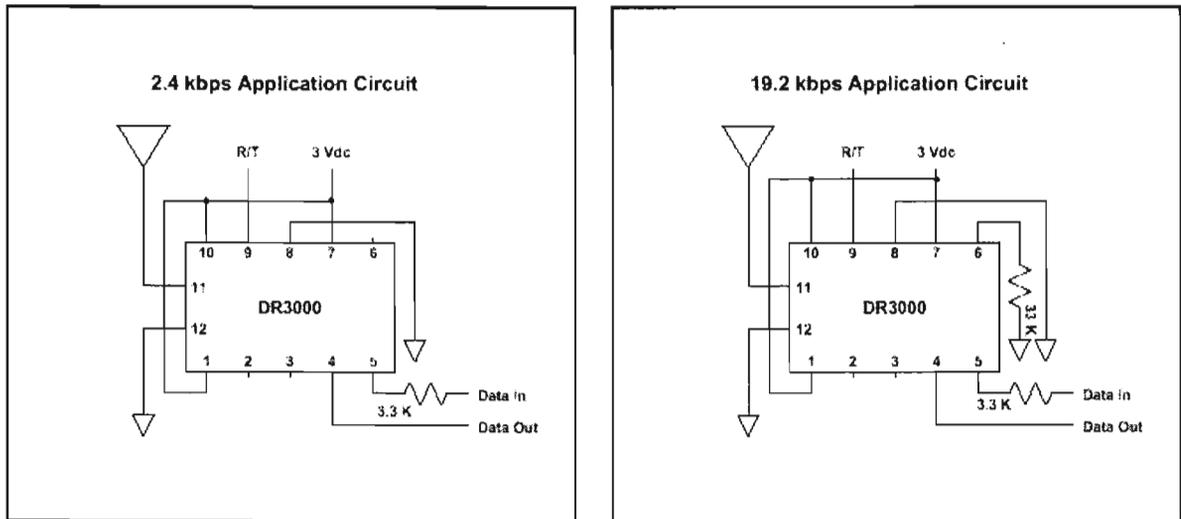


Fig. 2.1.3.10.2 Configuración de 2.4 y 19.2 kbps [7]

2.1.3.10.1 Configuración OOK a 19200 bps

En esta configuración podemos hallar los valores de acuerdo a valores de los componentes mostrados en la Tabla 2.1.3.10.1.

La velocidad de 19200 está dentro del rango de baja velocidad. Usando el valor de la resistencia $R_2 = R_{PR} = 270 \text{ K}\Omega$ en la Ec. 2.2 obtenemos:

$$t_{PRI} = \frac{R_{PR} - 10.5}{404} = \frac{270 - 10.5}{404} = 0.6423 \text{ } \mu\text{S}$$

El valor del tiempo t_{PRI} está en el rango permitido de 0.1 y 5 μs

Usando el valor de la resistencia $R_1 = R_{PW} = 330 \text{ K}\Omega$ en la Ec. 2.3 obtenemos:

$$t_{PW1} = \frac{R_{PW} - 18.6}{404} = \frac{330 - 18.6}{404} = 0.8628 \text{ } \mu\text{S}$$

El valor del tiempo t_{PW1} está en el rango permitido de 0.55 y 1 μs

Con referencia en la Fig. 2.1.3.6.1, obtenemos el tiempo t_{PRC} :

$$t_{PRC} = t_{PRI} + t_{PW1} = 0.6423 + 0.8628 = 1.505 \text{ } \mu\text{S}$$

Usando el valor de la resistencia $R_6 = R_{LPF} = 330 \text{ K}\Omega$ en la Ec. 2.5 podemos hallar el ancho de banda del filtro pasa bajos del receptor así:

$$f_{LPF} = \frac{1445}{R_{LPF}} = \frac{1445}{330} = 4.379 \text{ KHz}$$

Hay que notar que este valor debe ser ajustado para permitir un mayor ancho de banda y permitir un enlace de 19200 bps, reemplazando en la Ec. 2.7 el valor de $SP_{MIN} = 1/\text{velocidad} = 1/19200$, obtenemos:

$$f_{LPF} = \frac{2500}{1/19200} = 48 \text{ KHz}$$

Para ajustar el ancho de banda del filtro es necesario conectar una resistencia en paralelo a la resistencia R_6 , que se tenga una resistencia equivalente igual a:

$$R_{LPF} = \frac{1445}{48} = 30.104 \text{ K}\Omega$$

Esto se logra conectando una resistencia entre los pines 9 y tierra del módulo DR3000, con un valor de $R = 33.13 \text{ K}\Omega$.

2.1.3.11 Salida de Datos

El Pin RXTADA es el pin de salida de los datos recibidos. RXTADA maneja una carga paralela de 10 pF, 500 K Ω . El pico de corriente disponible en este pin se incrementa con la frecuencia de corte del filtro pasa bajos del receptor. En los modos de transmisión o baja potencia, se pone en alta impedancia. Si se requiere, puede usarse una resistencia pull-up o pull-down de 1000 K Ω para fijar un nivel lógico cuando se pone en alta impedancia.

La Fig. 2.1.3.11.1 muestra dos opciones, un amplificador inversor y no-inversor para RXDATA, apropiados para manejar cargas inferiores a 10 K Ω . Tener presente que la comunicación a través de un transceiver HAS es no invertida, un dato de pulso positivo transmite una señal de radio (más grande) que sale del receptor como un pulso positivo. El amplificador inversor puede ser usado para invertir la señal RXDATA cuando sea necesario.

Receiver Output Buffers

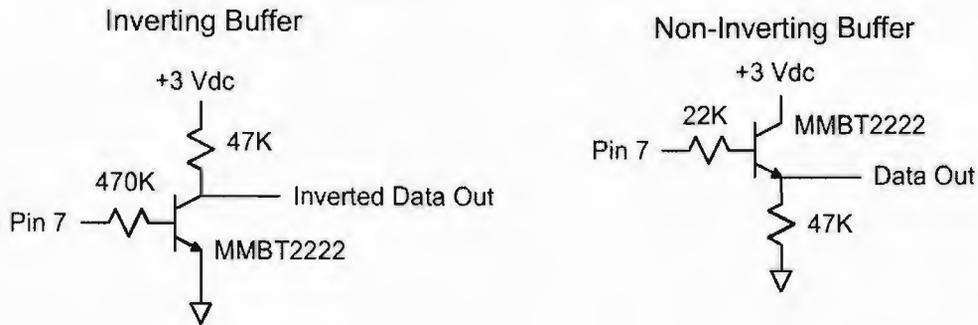


Fig. 2.1.3.11.1 Buffers a la Salida del Receptor ^[8]

2.2 Módulo Transmisor Digital en RF con base en el Microcontrolador PIC16F873

2.2.1 Descripción del Módulo Transmisor Digital en RF

El Módulo Transmisor digital en RF, tiene como función principal recibir los datos en formato RS232 provenientes de un dispositivo serial para luego codificarlos, agruparlos como un paquete de datos y finalmente ser transmitidos al receptor utilizando Radiofrecuencia.

Tal como se muestra en la Fig. 2.2.1.1, el diagrama de bloque del transmisor digital en RF, podemos ver que está formado por cuatro bloques importantes. El convertidor RS232-TTL que realiza el cambio de niveles de voltaje para que pueda ser compatible con el procesador digital. El procesador digital y codificador es el cerebro del transmisor digital en RF ya que realiza todas las funciones de control y flujo de los datos. La etapa de transmisión en RF recibe la señal digital en banda base para ser modulada y transmitida por el espacio aéreo que constituye el canal para el sistema de comunicación. Además, cuenta con un módulo que permite modificar la configuración del transmisor por medio de un programa Terminal de computadora.

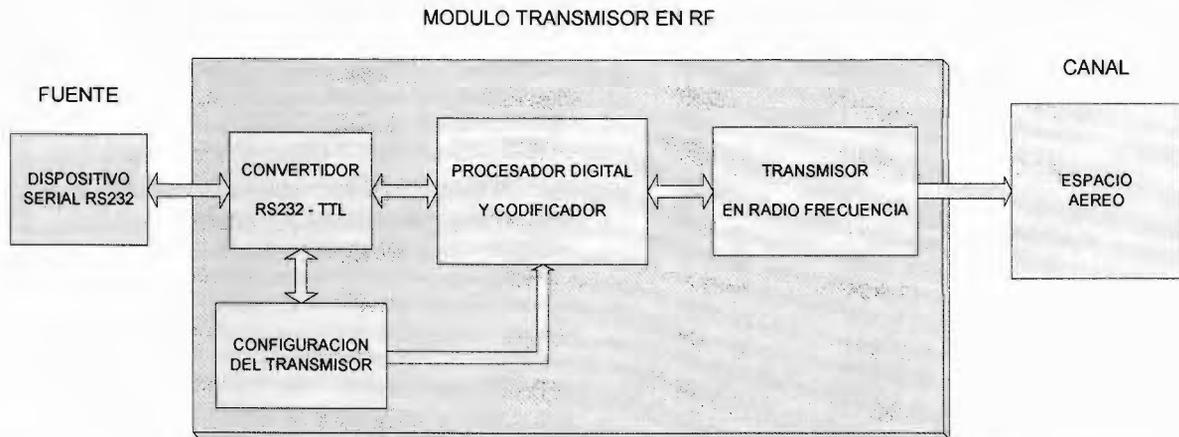


Fig. 2.2.1.1 Diagrama de Bloques del Transmisor Digital en RF

2.2.2. Consideraciones del Diseño del Módulo Transmisor digital en RF

El diseño del transmisor digital en RF se realiza teniendo como base al módulo transceiver DR3000 anteriormente mencionado. Al utilizar el módulo DR3000 en su configuración ASK/OOK y con una velocidad de sus datos en banda base a 19200 bps obliga al módulo Procesador Digital y Codificador a realizar una codificación de los bytes provenientes del dispositivo serial, esta codificación evita que se sature el transmisor o receptor por una cadena de ceros o unos no mayor de cuatro niveles iguales consecutivos. Estas son las características importantes a considerar y son los parámetros que definen el diseño del transmisor digital en RF.

Si nos enfocamos en la entrada del transmisor digital en RF, es decir al dispositivo serial RS232, tiene que manejar una velocidad inferior al módulo DR3000 y así evitar sobre escrituras de los datos empaquetados a ser transmitidos. En la Fig. 2.2.2.1 podemos ver los diagramas de tiempos del transmisor. Al considerar una velocidad de 9600 del dispositivo serial, entonces el tiempo necesario para recibir un byte, sería $(1/9600) * 10$ que es igual a $1041.67 \mu s$, este tiempo define la ubicación dentro del programa de control del módulo de Procesamiento Digital y Codificación, como se verá mas adelante en el apartado 2.2.4.

Podemos mencionar que el módulo de procesamiento digital y codificación lo conforma un microcontrolador PIC16F873 y sus programas de control del flujo de datos. Este microcontrolador trabaja con un oscilador de 20 MHz, y aunado a su arquitectura RISC, nos permite tener un ciclo de instrucción de 0.2 μ s, valor que nos permite diseñar rutinas de control bastante robustas, tal como a lo largo de este capítulo.

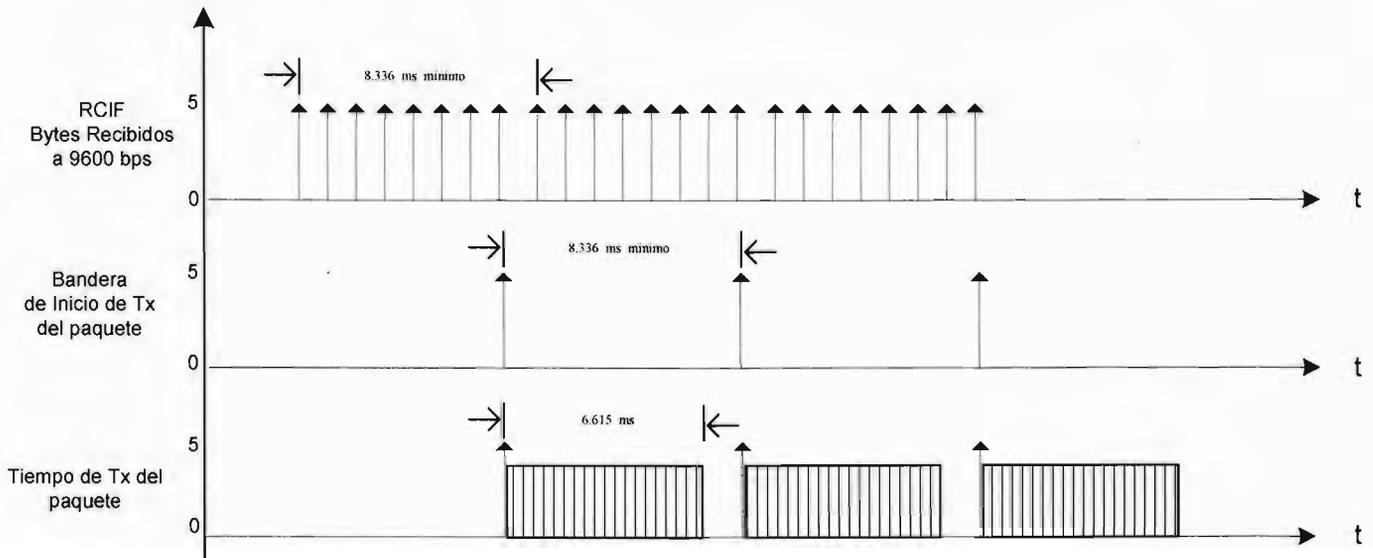


Fig. 2.2.2.1 Diagrama de Tiempos del Transmisor Digital en RF

Si consideramos que nuestro bloque de datos lo conforman 8 bytes del dispositivo serial, entonces para recibir cada bloque se tardaría como mínimo unos $8 \times 1041.6 \mu\text{s} = 8.33 \text{ ms}$. En nuestro diseño hay que considerar que antes de transmitir estos 8 bytes recibidos por medio del módulo DR3000, deben ser codificados y agrupados como un paquete de datos, se agrega 1 bit de inicio, una cabecera de 6 bits y dos bytes de Código de Detección de Errores codificados. La codificación realiza la conversión de cada byte en 12 bits. Es decir, por cada 8 bytes recibidos por el USART se transmiten 127 bits ($[2 \times (8+2)+1] \times 6+1$) de manera serial un pin del microcontrolador PIC al módulo DR3000. Si consideramos que el DR3000 transmite a una velocidad de 19200 bps, el tiempo necesario para transmitir el paquete completamente sería de $127/19200$ segundos = 6.615 ms. Es decir, con estos tiempos no podemos sobrescribir dos bloques continuos, tal como vemos en la Fig. 2.2.2.1.

Una última consideración de diseño del transmisor fue el eliminar al máximo el ruido del canal y así disminuir considerablemente la probabilidad de bit de error, esto se logró enviando una señal o tren de pulsos continuos, como se describe en el apartado 2.2.

2.2.3. Flujo de los Datos en el Transmisor Digital en RF

Los datos que fluyen en el módulo transmisor digital en RF, son controlados por un microcontrolador PIC16F873, este microcontrolador junto con sus dispositivos y periféricos externos necesarios como se muestra en la Fig. 2.2.3.1, fue programado para realizar todas las funciones del módulo Transmisor. La capacidad de memoria de programa del PIC permite que todas las rutinas del programa global residan en el mismo microcontrolador y así agilizar el acceso a datos y rutinas de control del módulo transmisor digital en RF.

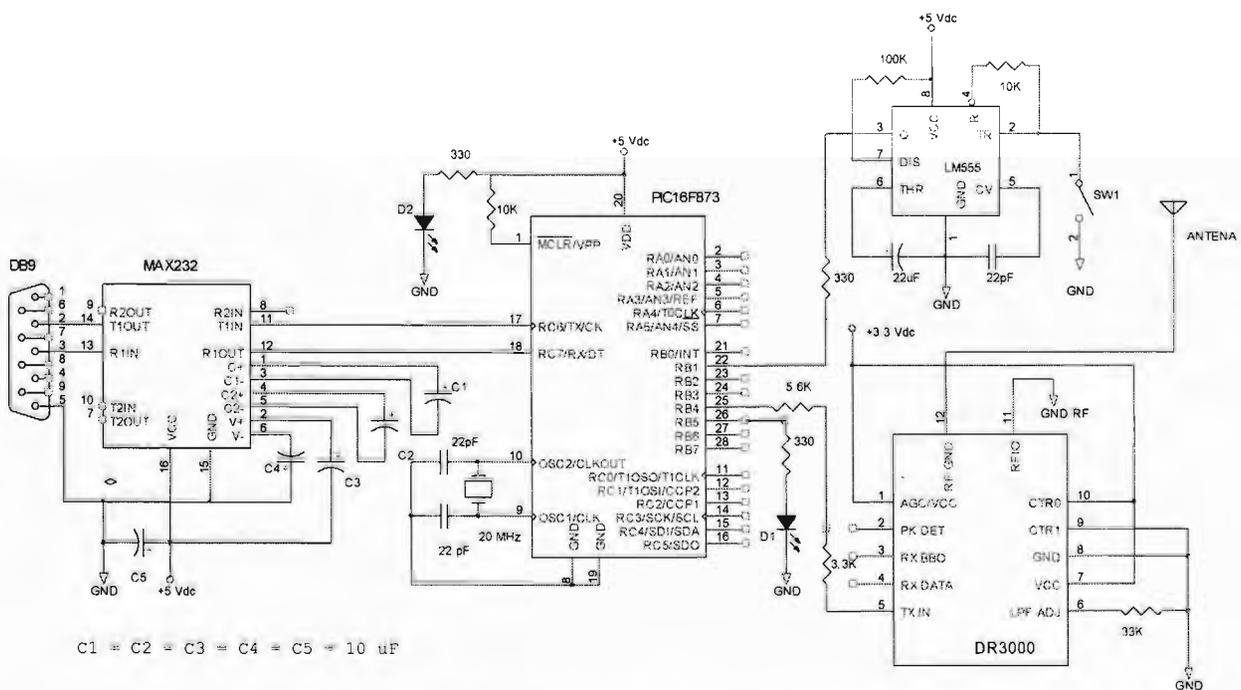


Fig. 2.2.3.1 Circuito del Módulo Transmisor Digital en RF

Como se observa en la Fig. 2.2.3.1, los bytes asíncronos recibidos en formato RS-232 por medio de la interface DB9, son convertidos a niveles TTL logrando así una compatibilidad con el microcontrolador PIC. Se utiliza el circuito integrado MAX232 ^[17], por su funcionamiento con sólo una fuente de +5 voltios y este chip internamente

genera los voltajes necesarios que exige el formato RS232 en ambos sentidos tal como se muestra en la Fig. 2.2.3.2a. Sólo se necesitan 3 líneas, las señales Tx, Rx y GND para lograr la comunicación asíncrona full-duplex sobre un cable dedicado.

Las líneas seriales convertidas a nivel TTL son conectadas a los pines del periférico USART del microcontrolador PIC. Hay que tener presente de conectar la línea de transmisión al pin de recepción del periférico USART – pin RC7/RX del PIC, y conectar la línea de recepción al pin de transmisión del periférico USART – pin RC6/TX del PIC. Como se definió previamente, los parámetros del puerto de comunicación que rigen esta conexión es de 9600 bps, datos asíncronos, 8 bits de datos, sin paridad, 1 bit de inicio y 1 bit de parada.

Los bytes recibidos por el microcontrolador PIC, son codificados y agrupados dentro de la memoria RAM del microcontrolador. Se implementó la codificación o transformación de cada byte recibido a un dato de 12 bits. Al completar el tamaño definido del bloque se activa un bit bandera que indica el inicio de la transmisión en banda base del paquete de datos una velocidad de 19200 bps que lo realiza el microcontrolador PIC por su pin RB4 hacia el terminal de entrada de datos del módulo DR3000. La cadena de bits que conforman el paquete de datos tiene que ser debidamente acoplada al módulo de transmisión en RF, esto se logra por medio de una resistencia, que reduce el nivel de corriente y voltaje al pin de entrada que soporta a lo mucho un nivel de 3.3 voltios.

El transmisor siempre esta enviando una onda cuadrada con una frecuencia periódica de 9600, es decir con un tiempo de bit equivalente a una velocidad de 19200 bps, esto se hace para identificar si existe un enlace en radio frecuencia con el receptor y para reducir el ruido del canal. Al dispararse la transmisión de información comienza a transmitir el paquete de datos y después de terminar se continua transmitiendo la onda cuadrada mencionada.

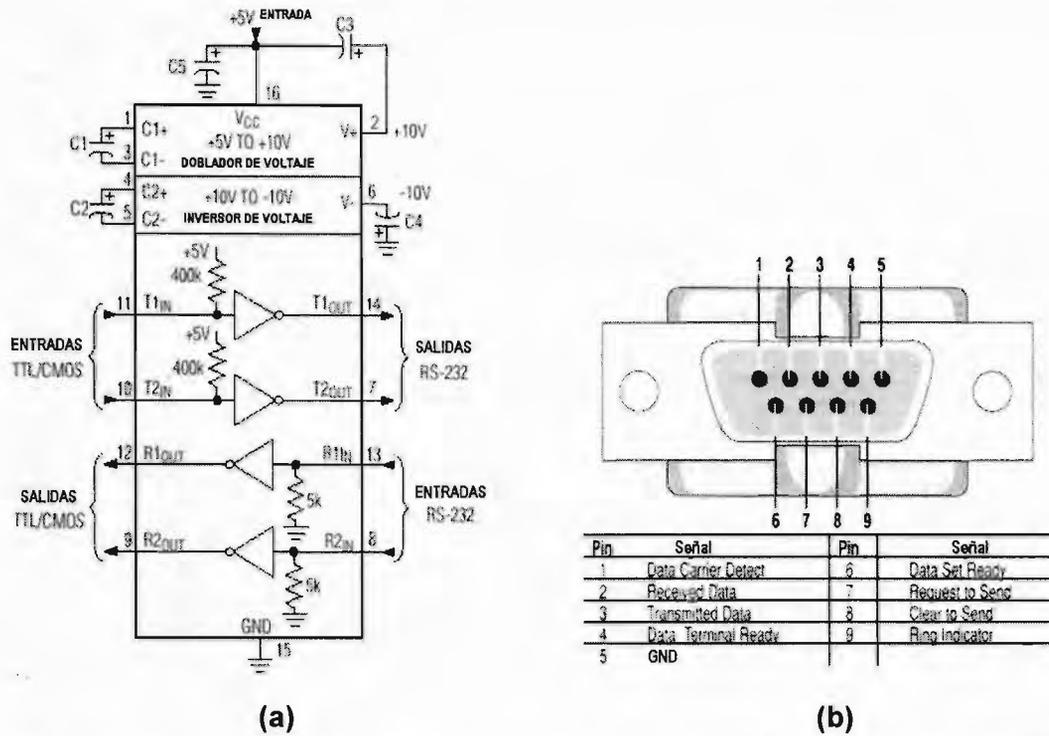


Fig. 2.2.3.2 (a) Circuito interno del Integrado MAX-232 ^[17] (b) Interface DB9 ^[17]

Como se observa en la Fig. 2.2.3.3, se implementó un pulsador que permite ingresar al modo de configuración, con la idea de cambiar la velocidad de comunicación con el dispositivo serial y algunos parámetros de funcionamiento del transceiver digital en RF. Para ello se utiliza el puerto serial RS232 de una PC y un programa de herramientas del sistema operativo Windows, el HyperTerminal y seleccionando por teclado las opciones que se presentan en la pantalla de interface de usuario del programa HyperTerminal.

Mas adelante, en el apartado 2.2.4.5, se verán mas detalles en la rutina Set_Parametros_Com. Para la implementación del generador del pulso se utilizó el circuito integrado LM555 en su configuración monoestable, tal como se muestra en la Fig. 2.2.3.3.

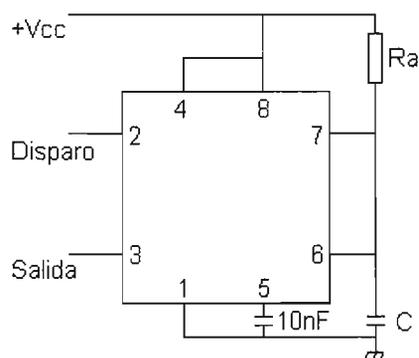


Fig. 2.2.3.3 Configuración del LM555 como monoestable

Se mantiene la señal de disparo en nivel alto (5V) para que la salida se mantenga en nivel bajo (0V), que es el estado de reposo. Al generarse un nivel bajo en el terminal de disparo se genera un nivel alto en la salida del LM555 por un tiempo determinado por la fórmula de la Ec. 2.12. ^[17]

$$T = 1.1 * Ra * C \quad (\text{Ec. 2.12})$$

Los valores usados para $Ra = 10 \text{ K}\Omega$ y para $C = 100 \mu\text{F}$, resultando en un tiempo activo del disparo $T = 3 \text{ seg}$, evitando así un rebote del switch de disparo al modo configuración. Luego de este tiempo T , la salida del LM555 regresa a su estado de reposo, es decir la salida se pone en nivel "0".

2.2.4. Rutinas del Programa de Control del Módulo Transmisor Digital en RF

Las rutinas del programa de control del módulo transmisor digital en RF residen en el microcontrolador PIC, en la memoria de programa. Después de evaluar los diversos modelos de microcontroladores de la gama media y con memoria de programa del tipo Flash se seleccionó el modelo PIC16F873A-20. Dentro de las características más importantes del PIC16F873, podemos mencionar que es un microcontrolador de 8 bits, tiene una arquitectura RISC, poseen diversos periféricos como el USART, SPI, convertidor A/D, etc. Al proporcionarle un reloj de cristal externo de 20 MHz se logra tener ciclos de instrucción de 0.2 μs , lo que nos permite implementar rutinas rápidas y acordes con el propósito de la comunicación.

Para la programación del microcontrolador se usó el kit de desarrollo de Microchip, el PICSTART PLUS. Este kit de desarrollo es un sistema de desarrollo que incluye el hardware externo o grabador y el software de desarrollo integrado MPLAB versión 5.1.

Como se muestra en la Fig. 2.2.4.1, el programa de control comienza, inicializando al microcontrolador, los pines de cada uno de sus puertos y definiendo el modo de funcionamiento del USART. Podemos dividir al programa de control en dos bloques importantes: el Programa Principal y la Rutina de Interrupción. El Programa Principal tiene la función de inicializar al microcontrolador del modo que queremos que funcione y de realizar tareas activadas ya sea por los periféricos o por la misma rutina de interrupción, aquí se encuentran las diversas tareas que no requieren de una atención con una interrupción. La Rutina de Interrupción se encarga de generar el tren de pulsos y de transmitir el paquete de datos cuando este listo, esta rutina de interrupción se activa en cada 52 μ s. Los detalles de las rutinas del programa de control se verán mas adelante en este mismo capítulo.

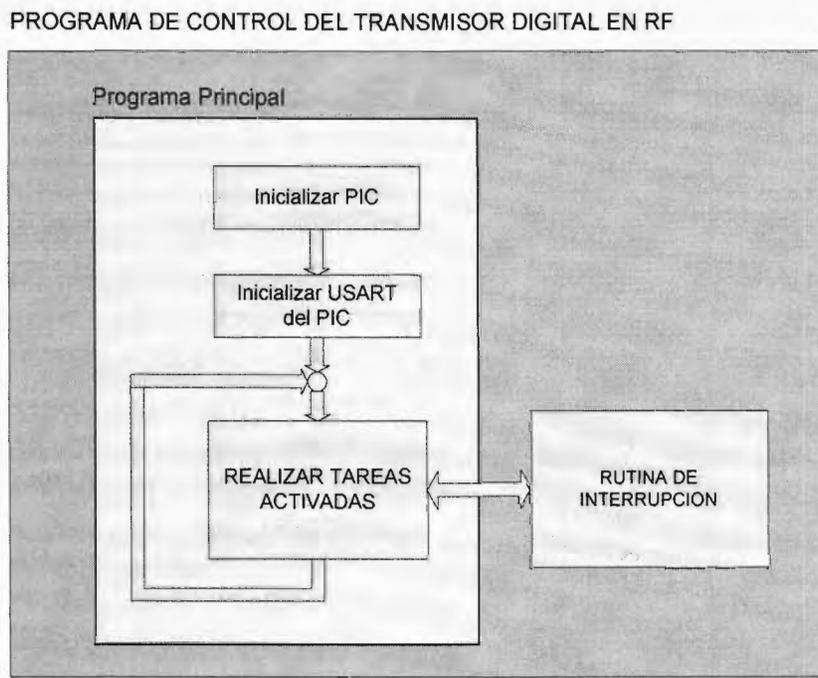
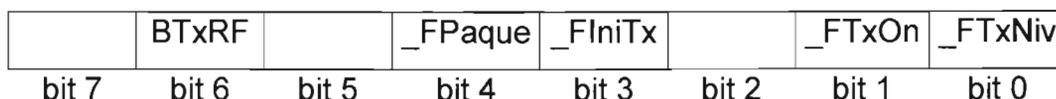


Fig. 2.2.4.1 Diagrama de Bloques del Programa de Control del Transmisor Digital en RF

Describimos una variable de control del programa en general, se trata de un registro de 8 bits:

ESTADOCOM, usado para monitorear el estado de la transmisión del paquete de datos al módulo de radio frecuencia, así:



BTxRF Contiene el valor del bit a transmitir

_FPaque Bandera que indica que se recibió un paquete de datos y están listos para transmitirse.

_FIniTx Bandera que indica que ya se transmitió el primer bit del paquete y que se debe continuar transmitiendo todo el bloque (paquete).

2.2.4.1. Rutina del Programa Principal: *Pprincipal*

Tal como se muestra en la Fig. 2.2.4.1.1, la rutina del programa principal se encarga de realizar la configuración del modo de funcionamiento del microcontrolador PIC así como la inicialización de los periféricos a usar en la transmisión digital de los datos recibidos en su interface RS232 hacia en espacio aéreo para que sea recibido por el receptor.

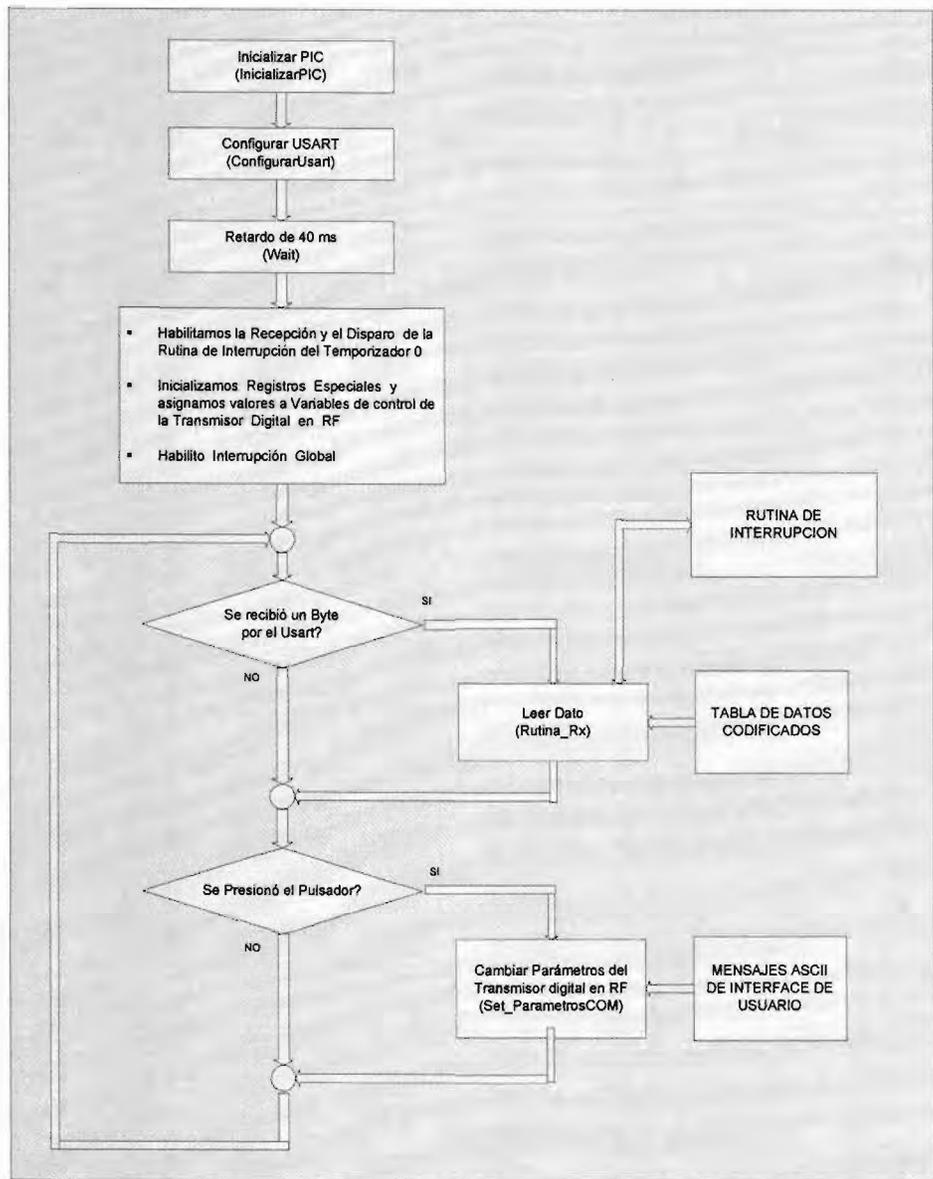


Fig. 2.2.4.1.1 Diagrama de Bloques de la Rutina Principal del Transmisor Digital en RF

Hay que mencionar que previamente a esta rutina se realizó la definición de la Tabla de Datos Codificados en memoria de programa del microcontrolador. Esta tabla es usada en el bloque Leer Dato por la rutina que codifica los datos recibidos (rutina ConByte12), tal como se mencionará más al detalle en este capítulo. Igualmente se define en memoria de programa del PIC, las cadenas de texto usadas para interactuar en la Interfaz de Usuario proporcionada por el programa Hyperterminal del sistema operativo Windows.

Continuaremos con la descripción detallada de cada una de las rutinas que conforman el programa del transmisor digital en RF. Para una mejor referencia se proporcionan las rutinas en assembler del microcontrolador PIC en el anexo B, al final de la redacción de esta tesis.

Como lo muestra el diagrama de la Fig. 2.2.4.1.1, se llama a la rutina *InicializarPIC* que se encarga de configurar los puertos A y E como terminales digitales, los terminales del puerto B se fijan como salidas digitales y se asignan el arreglo de resistencias pull-up. El terminal PB4, etiquetado como `_tx`, es usado para la transmisión serial del paquete de datos. Se configura la asignación del reloj interno y 2 como el divisor de frecuencia del Temporizador TMR0. Se configuran los terminales RC7/XR y RC6/TX del puerto C como entrada y salida, respectivamente, tal como se requiere el uso del periférico de comunicación serial (USART) del microcontrolador PIC.

La rutina *ConfigurarUsart* realiza la configuración y habilitación del puerto serial USART del microcontrolador a una velocidad de 9600 bps, 8 bits de datos, no paridad, 1 bit de inicio y 1 de parada, esta configuración tiene que ser la misma que el dispositivo serial que proporciona los datos, tal como se observa en la Fig. 2.2.1.1. Para definir la velocidad de comunicación del USART del microcontrolador PIC a 9600 bps, se requiere fijar el bit BRGH = 1 y el registro SPBRG = D'129'. Por diseño del software de control del puerto se deshabilitó las interrupciones de la transmisión y recepción.

Luego se llama a la rutina *Wait*, que genera un retardo de 40 ms aproximadamente, logrando estabilizar los voltajes del microcontrolador PIC y sus periféricos, sobre todo el USART. En general este retardo es recomendable para la correcta operación de los puertos del microcontrolador y condiciones de nuestra aplicación en particular.

Continuando con el programa principal se habilita la recepción del USART del PIC, esto es necesario porque se usa el poleo de la bandera la recepción para leer el dato recibido, mas adelante se describe más detalladamente en el programa de control del transceiver. En esta etapa es necesario reiniciar los valores de: las Banderas `_Fpaque` y `_FiniTx`; y las diversas variables usadas en el control de la secuencia de la rutina del

lazo Principal. Las variables más importante son CONTA_TP, que almacena el valor de bytes a recibir y agrupar como un paquete de datos; CCRH y CCRL que almacenan el valor del código de detección de errores.

Se habilita la interrupción del temporizador TMRO, encargada de generar el sincronismo para los datos transmitidos con un período de disparo de 52 μ s (que equivale a una velocidad de 19200 bps), esto se logra fijando el valor del registro TMR0 = 0x88.

Luego se habilita la interrupción Global del microcontrolador PIC para que pueda responder con cualquier interrupción habilitada, en nuestro caso a la interrupción del sobreflujo del contador del temporizador TMR0. Para esta rutina de interrupción se definen las variables usadas en la rutina de interrupción: FSR_INTER, que almacena el valor FSR usada en la rutina de interrupción; BITS_TX, que almacena el número de bits a transmitir de un byte; NNIBLES_TX, que almacena el número de nibbles que conforman el paquete de datos a transmitir.

2.2.4.2. Rutina de Lectura de Datos recibidos: *Rutina_RX*

Tal como se observa en las figuras Fig. 2.2.4.1 y Fig. 2.2.4.1.1, el lazo importante a considerar en la secuencia del programa principal, es el monitoreo de la bandera RCIF que nos indica si existe o no un dato recibido en el registro del USART del microcontrolador. Si RCIF=1 se llama a la rutina Rutina_RX, que se encarga de leer el byte recibido y lo almacena en la variable BYTE_RX. Este byte es usado por la rutina SumarCCR que se encarga de sumar los valores de los bytes recibidos en dos registros CCRH:CCRL, que serán incluidos en el paquete de datos, y que sirven como código de detección de errores del lado del receptor. Después se ejecuta la rutina ConByte12 que también usa el byte recibido para codificarlo y almacenarlo en memoria RAM como dos bytes.

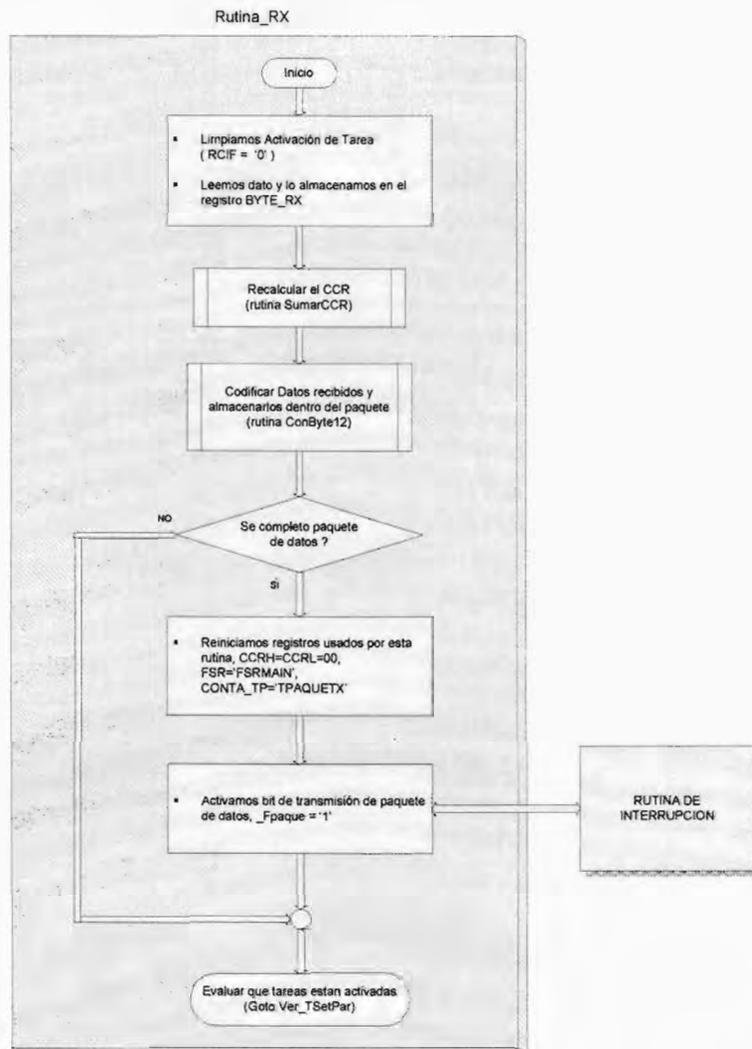


Fig. 2.2.4.2.1 Diagrama de Bloques de la Rutina: Rutina_RX

Hay que mencionar que el inicio del paquete de datos codificados es una cabecera, en nuestro caso particular es el valor 0x13. Esta etapa descrita se realiza hasta recibir 8 bytes del dispositivo serial, completando así nuestro tamaño de paquete de datos codificados en RAM del Banco 0, estos datos son copiados como imagen en el Banco 1 de la RAM para así estar listos para su transmisión. Esta rutina avisa a la rutina de interrupción que tiene un paquete de datos listos para ser transmitidos al fijar el bit `_Fpaque = '1'`, como se describe luego en la rutina de interrupción. Sólo mencionaremos que el transmisor digital en RF deja de transmitir el tren de pulsos y empieza a transmitir el contenido del paquete de datos codificados ubicados en RAM

del microcontrolador por el mismo Terminal, o sea por el pin RB4. La parte final del lazo realiza nuevamente la inicialización de las variables modificadas en todo este lazo del programa para nuevamente estar listos para seguir recibiendo bytes del dispositivo serial. El puntero del programa de control salta a la parte de evaluar si se disparó alguna otra tarea para ser realizada.

2.2.4.3. Rutina de Codificación: *ConByte12*

Esta rutina se encarga de convertir 1 byte recibido del dispositivo serial en un dato de 12 bits, que son almacenados en dos bytes de la memoria RAM del microcontrolador PIC, estos datos van formando parte del paquete de datos a ser transmitidos por RF.

Haciendo referencia a la Fig. 2.2.4.3.1, el byte recibido por el microcontrolador, se divide en 2 nibbles (1 nibble = 4 bits), el nibble menos significativo es codificado en un dato de 6 bits usando la Tabla 2.2.4.3.1, para luego ser almacenado en un byte de la memoria RAM que van formando el paquete de datos a transmitir. Análogamente se trabaja con el nibble más significativo. Hay que tener presente que de cada byte almacenado en la RAM sólo se transmiten los 6 bits menos significativos.

Esta codificación se hace para conservar el balance DC y AC de la señal y respetar el número máximo de 1s y 0s a ser transmitidos que soporta la configuración del módulo transceiver DR3000.

En el microcontrolador esto se logra accediendo a cada byte de la Tabla de Datos Codificados almacenados en memoria de programa del microcontrolador. El acceso se logra usando un nibble y el offset de la Tabla, estos valores son agregados al contador del programa del microcontrolador y usando una rutina de retorno de un valor de un byte que corresponde a la codificación del nibble. El detalle del programa en assembler se muestra en el Anexo B.

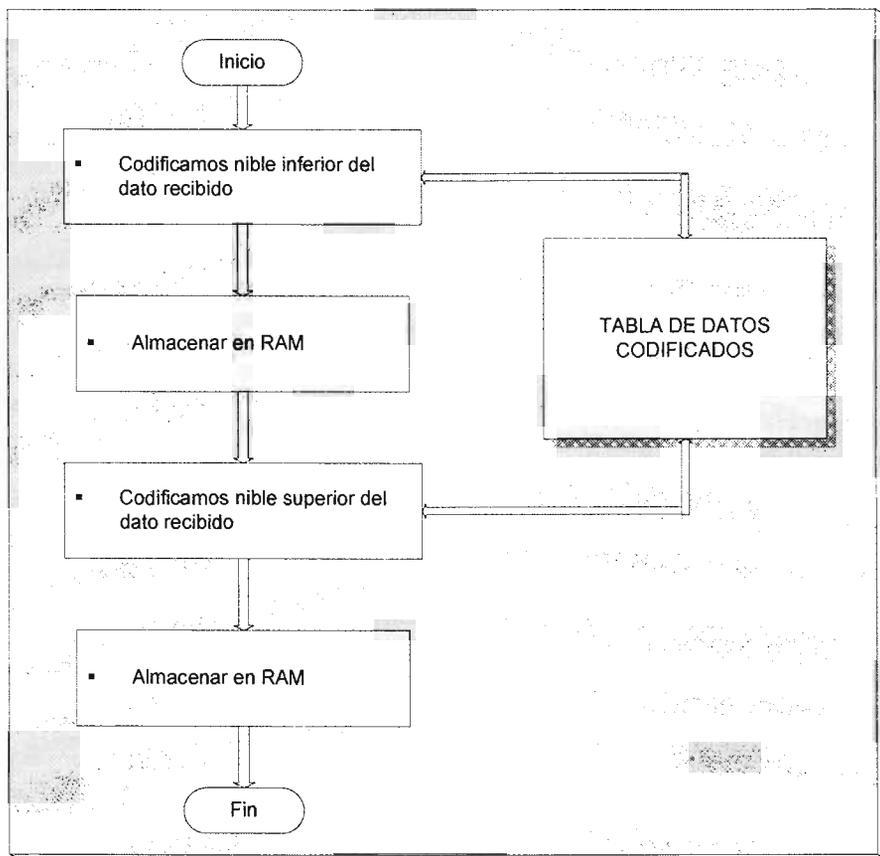


Fig. 2.2.4.3.1 Diagrama de Bloques de la Rutina: ConByte12

Los valores de la Tabla de codificación almacenada en memoria del programa que corresponden a cada valor de nibble se muestran en la siguiente Tabla 2.2.4.3.1.

Tabla 2.2.4.3.1 Correspondencia de Nibbles y su Codificación

Nibble	6 bits
0000	→ 0x15
0001	→ 0x31
0010	→ 0x32
0011	→ 0x23
0100	→ 0x34
0101	→ 0x25
0110	→ 0x26
0111	→ 0x07

1000	→	0x38
1001	→	0x29
1010	→	0x2A
1011	→	0x0B
1100	→	0x2C
1101	→	0x0D
1110	→	0x0E
1111	→	0x1C

Así, al transmitirse ocho caracteres A (código ASCII 0x41), las rutinas del programa Rutina_RX obtendrán un valor del CCR = 0x0208 (CCRL=08, CCRH=02). El valor hexadecimal 0x41, que corresponde al carácter A, se divide en dos nibbles "1" y "4". Si hacemos corresponder estos valores con un valor codificado usando la Tabla 2.2.4.3.1, resulta que el "1" se codifica como 0x31 y el "4" se codifica como 0x34. Al completar el bloque de datos, se codifica el CCR, es decir el CCRL se divide en "8" (que se codifica como 0x38) y en "0" (que se decodifica en 0x15); el CCRH se divide en "2" (que se codifica como 0x32) y en "0" (que se codifica como 0x15) Entonces, sabiendo que la dirección de los datos codificados se inicia en 0xA0, los datos ya codificados con su respectiva posición de memoria será:

A0: 0x13	A6: 0x34	AC: 0x34	B2: 0x15
A1: 0x31	A7: 0x31	AD: 0x31	B3: 0x32
A2: 0x34	A8: 0x34	AE: 0x34	B4: 0x15
A3: 0x31	A9: 0x31	AF: 0x31	B5:
A4: 0x34	AA: 0x34	B0: 0x34	B6:
A5: 0x31	AB: 0x31	B1: 0x38	B7:

Si se transmiten "WALTERKC" que son 8 bytes, el tamaño del paquete, se tiene un CCR = 0x025D (CCRL=5D, CCRH=02). Haciendo la correspondencia como el caso anterior, se tiene la siguiente tabla de datos ya codificados en su respectiva posición de memoria:

A0: 0x13	A6: 0x34	AC: 0x25	B2: 0x25
A1: 0x07	A7: 0x34	AD: 0x0B	B3: 0x32
A2: 0x25	A8: 0x25	AE: 0x34	B4: 0x15
A3: 0x31	A9: 0x25	AF: 0x23	B5:
A4: 0x34	AA: 0x34	B0: 0x34	B6:
A5: 0x2C	AB: 0x32	B1: 0x0D	B7:

2.2.4.4. Rutina de Interrupción : *RutinaInterrupcion*

Esta rutina de interrupción está configurada para activarse en cada desbordamiento del temporizador TMR0, cada 52 μ s. Esta rutina genera un tren de pulsos sobre el terminal RB4 del microcontrolador, y al existir un paquete de datos listos se empieza a transmitir por este mismo pin cada bit que compone el paquete de datos codificados sobre el módulo transceiver inalámbrico DR3000, tal como se observa en la Fig. 2.2.3.1.

Para generar un ciclo de interrupción del desbordamiento del temporizador TMR0 cada 52 μ s y con un reloj de cristal de 20 MHz, es necesario modificar el valor del registro TMR0 a un valor de 0x88. Este tiempo seleccionado corresponde al tiempo de bit de un enlace serial a una velocidad de 19200 bps.

Como podemos observar en la Fig. 2.2.4.4.1, su diagrama de bloques de esta rutina de interrupción, se inicia respaldando los valores de los registros W, STATUS y FSR, ya que estos registros son usados en la rutina del programa principal y su modificación alteraría drásticamente el correcto funcionamiento del módulo transmisor. También observamos que estos valores de estos registros son nuevamente recuperados al final de la rutina de interrupción.

Continuaremos con la descripción de cada uno de los pasos a seguir por el programa de interrupción para poder llevar acabo su función, los detalles del código en assembler se muestran en el Anexo B.

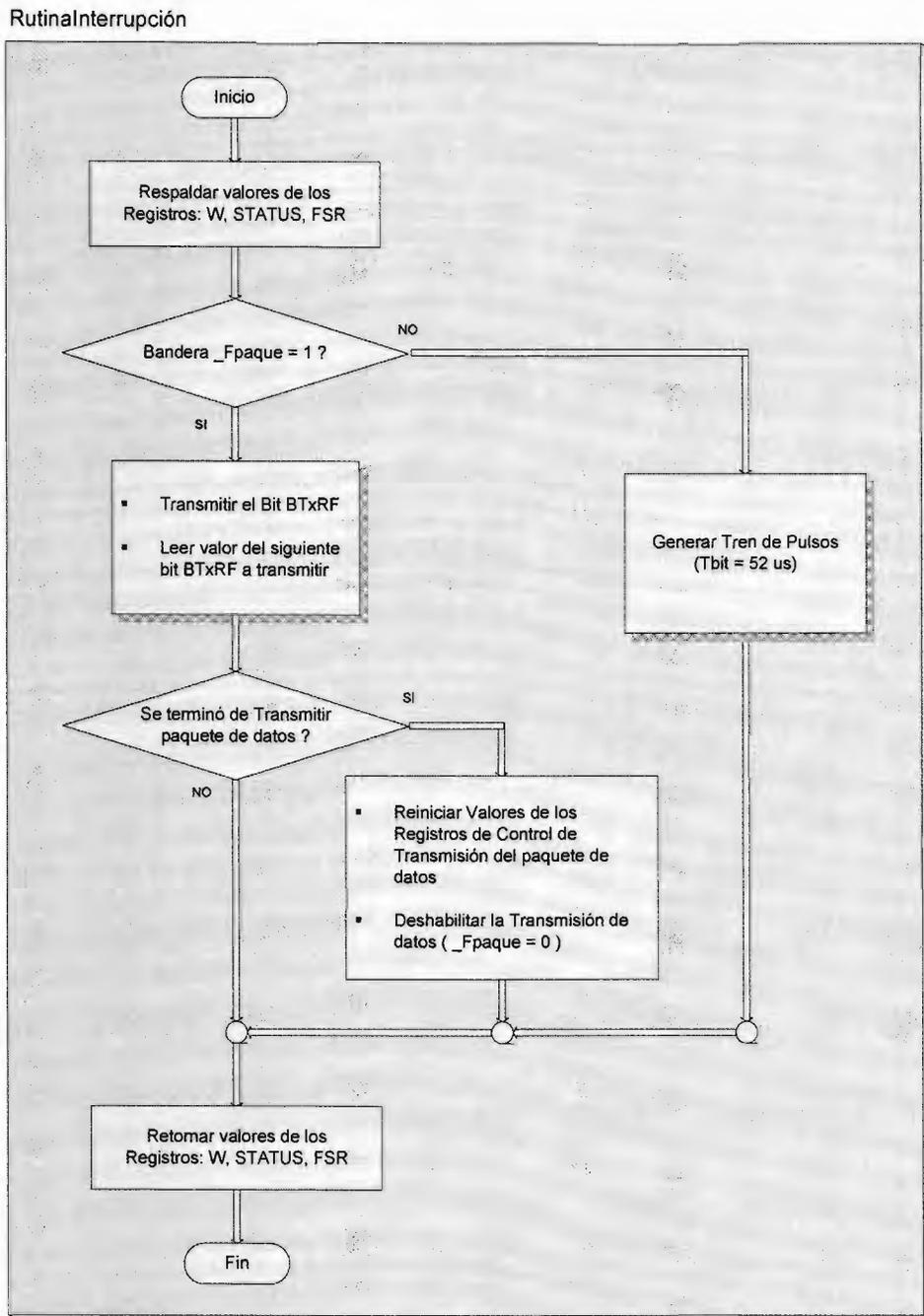


Fig. 2.2.4.4.1 Diagrama de Bloques de la Rutina de Interrupción: RutinaInterrupcion

Por lo general esta rutina se encarga de generar una onda cuadrada sobre el terminal PB4, esto nos permite enlazarnos con el módulo receptor y disminuir el ruido del canal de radiofrecuencia considerablemente. Hay que considerar que el valor de las banderas `_Fpaque` ó `_FlniTx` tienen el valor 0. Cuando la bandera `_Fpaque = 1`, nos indican que se tiene listo un bloque de datos a transmitir, se interrumpe la transmisión de la onda

cuadrada y empieza a transmitir el bloque de datos, se programó para que se mande una trama de 01 y así en el lado del receptor pueda ser observado el flanco de bajada y disparar el inicio de la lectura o recepción de los datos.

Luego de conservar la trama de la onda cuadrada, se transmite el bit "1", para luego fijar el bit `_FiniTX = 1`, que es la bandera que inicia la transmisión del paquete completo de datos. Se transmite el primer bit "1" con la intención de asegurar el disparo correcto de la lectura en el lado receptor, es decir generamos un nivel alto en el terminal PB4 un tiempo mínimo de 104 μ s. Y también el transmitir un primer bit ya conocido nos permite aprovechar esta secuencia para calcular el siguiente bit a transmitir en el siguiente disparo de la rutina de interrupción, es decir cuando la rutina de interrupción se dispare nuevamente ya se conocerá el bit a transmitir lo que evita un posible desplazamiento en los tiempos de bits.

Como se observa en el código assembler del Anexo B, se utilizan dos registros de control. El registro `BITS_TX` controla el número de bits a transmitir de la variable `BYTE_TX`, su valor inicial es de 6, los bits útiles de un byte del bloque de datos codificados útiles en nuestro algoritmo, se inicia con la transmisión del bit menos significativo de la variable `BYTE_TX`. Otro registro de control es la variable `NNIBLES_TX`, que almacena los nibbles que quedan por transmitir, su valor inicial depende del tamaño de bytes recibidos por el USART del microcontrolador.

Después de transmitir el paquete de datos se reinician los valores de las variables por esta interrupción, es decir se fija el valor de las variables: `FSR_INTER = 0xC0` y `NNIBLES_TX = 0x15`. Además de fijar las banderas `_FPaque = _FIniTx = 0`.

Nuevamente se reinicia con la generación de la onda cuadrada sobre el terminal RB4, y queda a la espera del siguiente paquete de datos a transmitir.

2.2.4.5. Rutina de Configuración de Parámetros del Transmisor: Set_Parametros_Com

Esta rutina del microcontrolador permite configurar parámetros del transmisor, su ejecución esta ligada a un switch que genera un pulso y activa esta rutina. Para generar el pulso se utilizó el circuito integrado LM555 en su configuración monoestable, tal como se muestra en la Fig. 2.2.3.3.

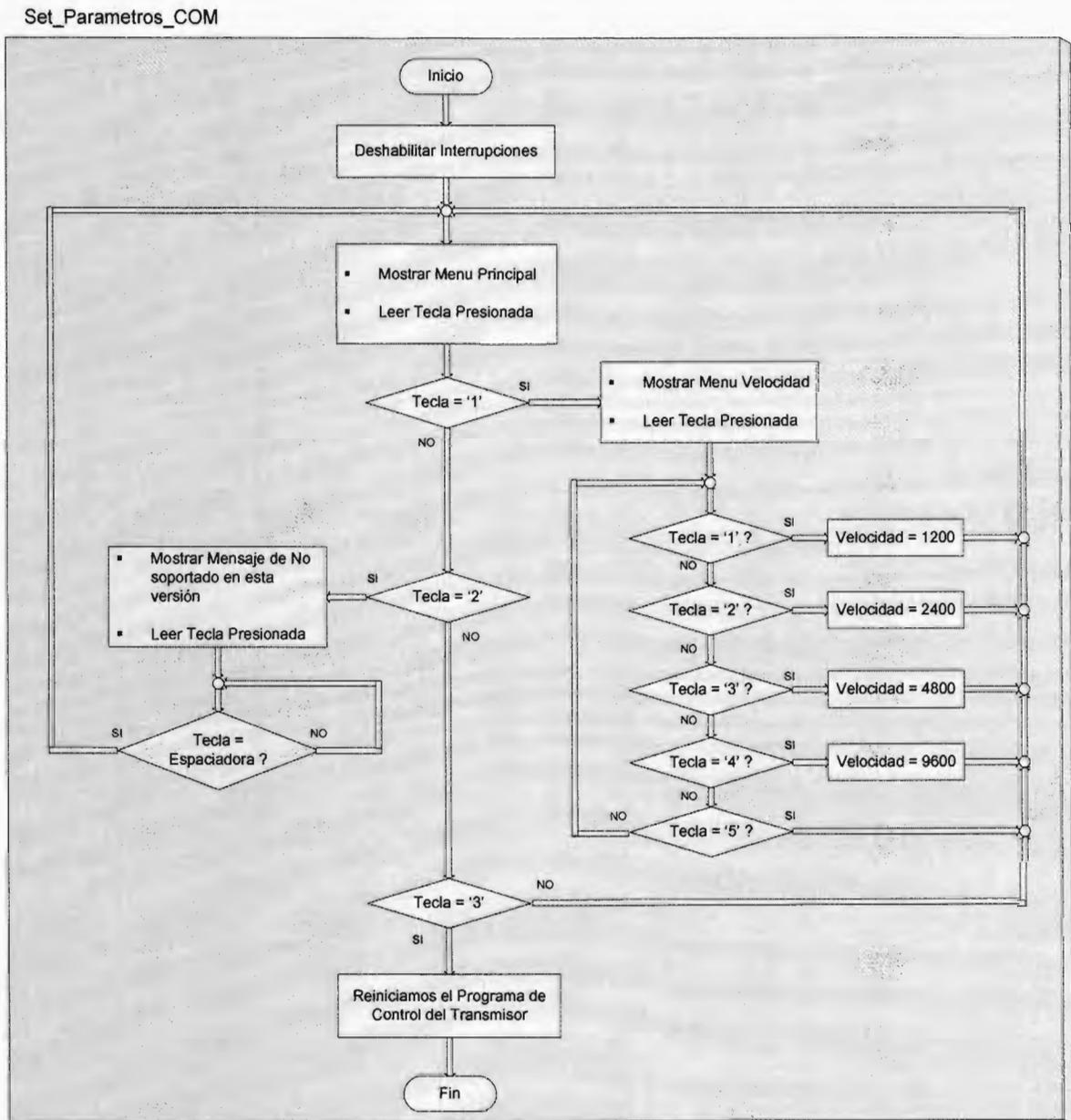


Fig. 2.2.4.5.1 Interface de Usuario del Hyperterminal con el Transceiver de Datos

Hay que recalcar que antes de presionar el pulsador se debe tener en cuenta en habilitar y configurar el Hyperterminal de Windows, como un enlace serie asíncrono RS232 con velocidad de 19200 bps, 8 bits de datos, sin paridad, 1 bit de inicio y 1 bit de parada. Lo requerimos para poder visualizar los mensajes que envía el microcontrolador y poder interactuar con el programa de configuración de parámetros.

El diagrama de bloques de esta rutina se muestra en la Fig. 2.2.4.5.1, hay que recalcar que el programa principal del transmisor está continuamente monitoreando la activación del pulso para ir al modo de Configuración y ejecutar la rutina Set_Parametros_COM. Primero realiza la desactivación de cualquier rutina de interrupción, es decir no se transmite el tren de pulsos o cualquier dato. Si el led D1 del diagrama circuital de la Fig. 2.2.3.1 está encendido, nos indica que estamos dentro del modo Configuración, al salir de la rutina el led se apagará.

Los valores modificados en este modo de configuración, se almacenaran en memoria EEPROM del microcontrolador y al reiniciar su ejecución funcionará con sus parámetros modificados.

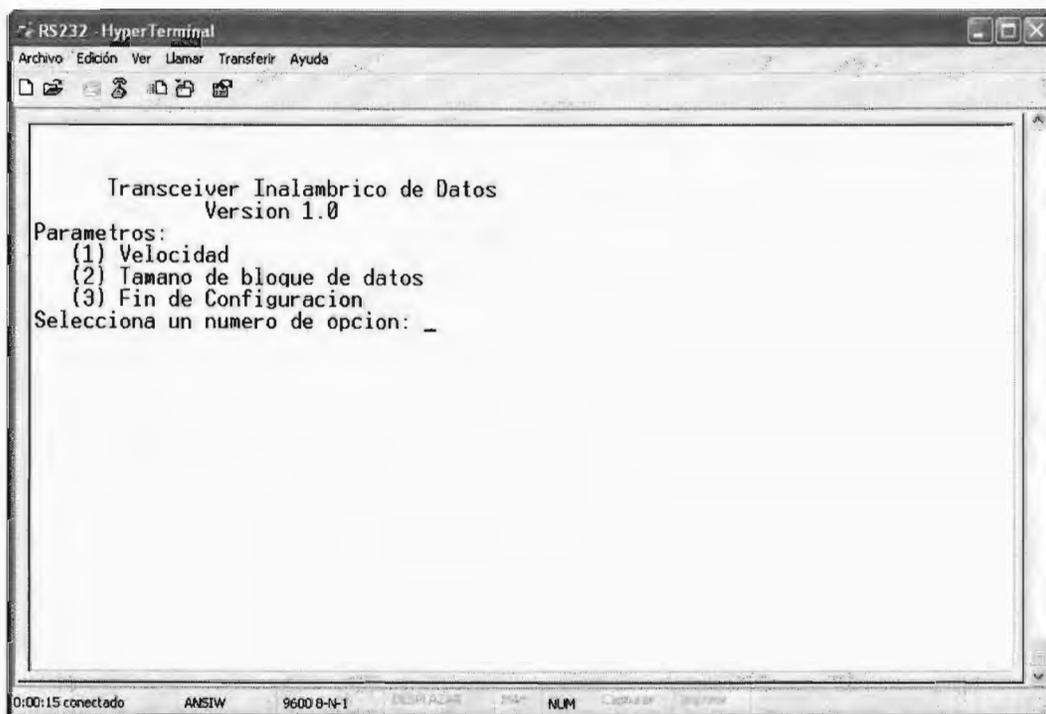


Fig. 2.2.4.5.2 Interface de Usuario del Hyperterminal con el Transceiver de Datos

En la Fig. 2.2.4.5.2, se muestra el menú principal que es presentado al usuario en la pantalla del Hyperterminal, este mensaje se visualiza al presionar el pulsador de modo de configuración. El programa espera presionar el número de las opciones presentadas, es este caso sólo tres teclas que corresponden a los números: 1, 2 y 3. Si presionamos alguna otra tecla el programa lo ignora.

No olvidemos que la interface de usuario del programa Hyperterminal proporciona por el puerto serial el código ASCII de la tecla presionada y este valor es procesado por el programa Set_Parametros_COM, tal como se muestra en el diagrama de flujo de la Fig. 2.2.4.5.1.

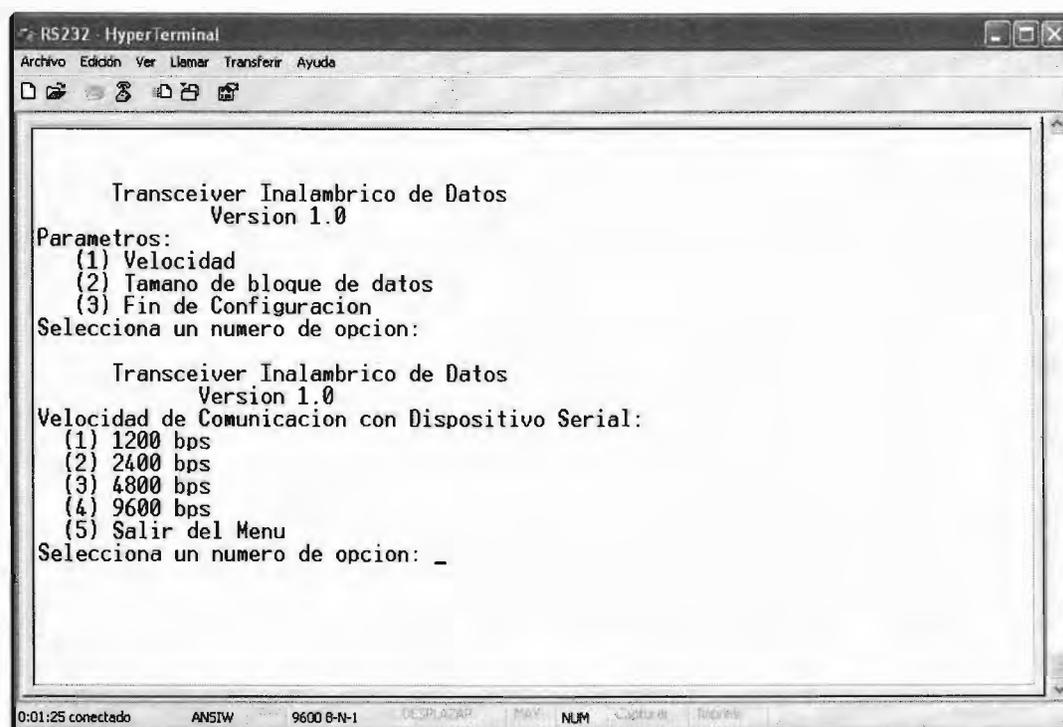


Fig. 2.2.4.5.3. Interface de Usuario del Hyperterminal con el Transceiver de Datos

Si presionamos la tecla del número 1, aparece otro menú, tal como se muestra en la Fig. 2.2.4.5.3, que nos permite cambiar la velocidad del dispositivo serial a ser conectado por el transmisor. Tenemos 4 velocidades soportadas por el transmisor digital en RF. Esta velocidad seleccionada no afecta a la velocidad que se fija al ingresar a este modo de Configuración del transmisor, esta velocidad siempre será de

19200 bps. Al presionar la tecla del número 5, regresamos al menú principal y no se genera ningún cambio en la velocidad.

La Fig. 2.2.4.5.4 muestra el mensaje al presionar la tecla del número 2, del menú principal. Esta opción permite cambiar el tamaño del paquete de datos, pero esta versión no la soporta, es por ello que después de enviar un mensaje espera hasta que se presione la tecla espaciadora y regresar a mostrar el menú principal.

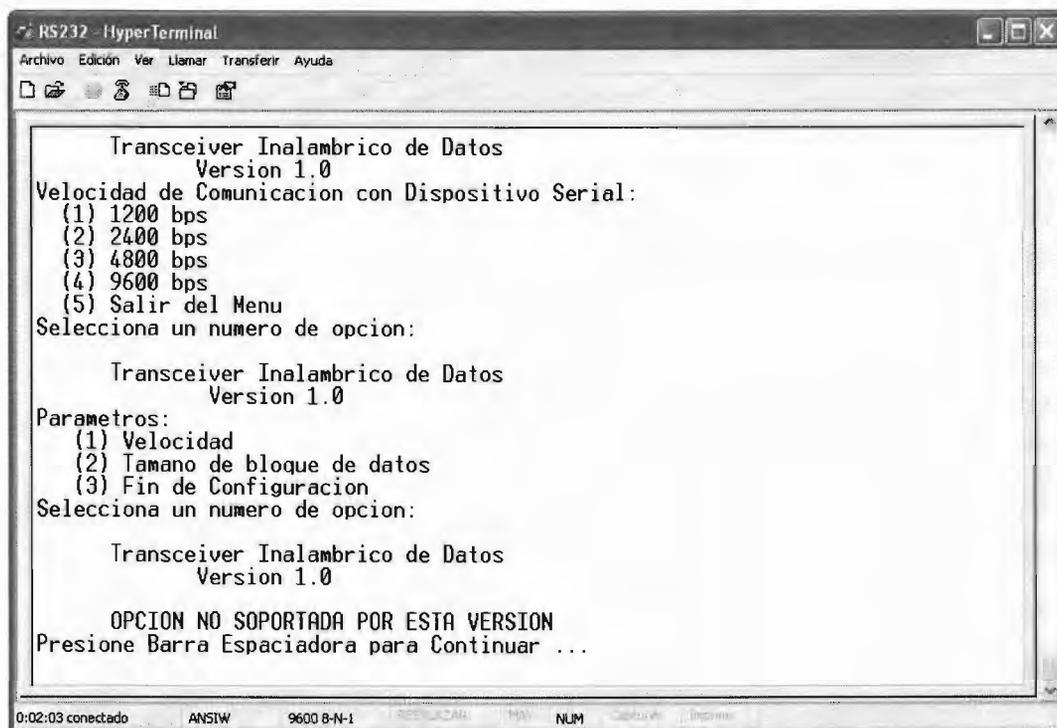


Fig. 2.2.4.5.4 Interface de Usuario del Hyperterminal con el Transceiver de Datos

En el menú principal al presionar la tecla 3, que corresponde a la opción “Salir del Menú”, apaga el led D1, lo que indica que se terminó el modo de configuración, así que genera un salto del puntero del programa y el programa se reinicia con los valores modificados.

2.3 Módulo Receptor Digital en RF con base en el Microcontrolador PIC16F873

2.3.1 Descripción del Módulo Receptor Digital en RF

El Módulo Receptor digital en RF, tiene como objetivo o función principal el de monitorear en el espacio aéreo el tren de pulsos proporcionados por el módulo transmisor digital en RF y al identificar una secuencia inicial de bits preestablecida, da inicio con la lectura de cada uno de los bits codificados que conforman el paquete de datos transmitidos. Estos datos recibidos son procesados en primera instancia verificando el correcto código de detección de errores y luego los decodificar y transmite a un dispositivo serial en formato RS232.

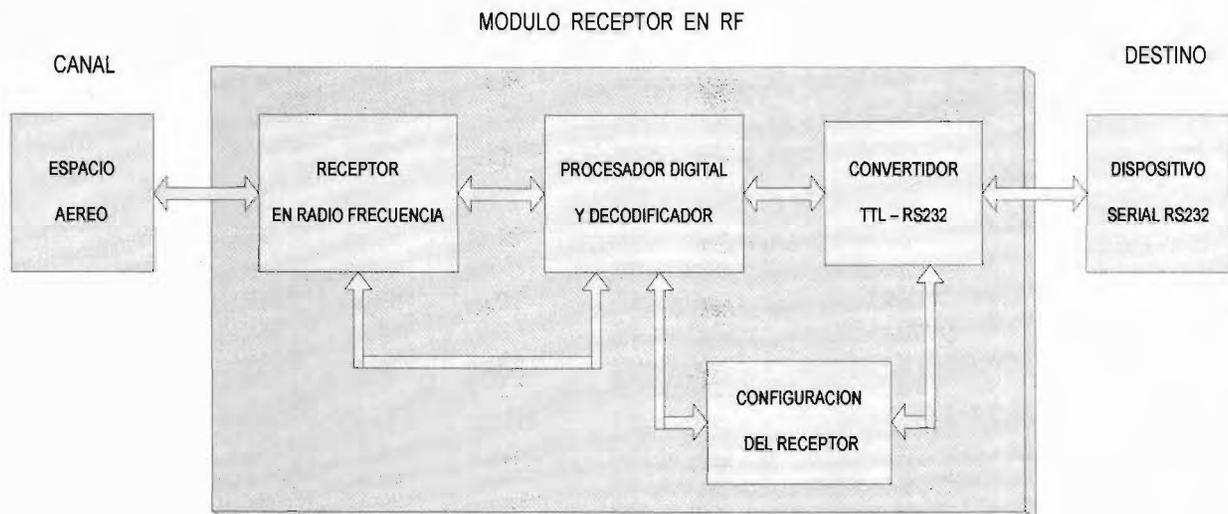


Fig. 2.3.1.1 Diagrama de Bloques de Receptor Digital en RF

Tal como se muestra en la Fig. 2.3.1.1, el diagrama de bloque del receptor digital en RF está formado por cuatro bloques importantes. La etapa de recepción en RF, recibe la portadora en la banda UHF del espacio aéreo, esta portadora contiene la información digital que conforma el paquete de datos transmitidos, esta etapa proporciona la señal digital en banda base para ser validada por la etapa del Procesador digital, que se encarga de validar el paquete de datos, decodificar y transmitir los datos serialmente en formato TTL. La etapa del Convertidor TTL-RS232 permite generar el formato RS232 y así poder ser compatible con el dispositivo serial RS232 destino. El Procesador Digital y

Decodificador es el cerebro del transmisor digital en RF ya que realiza todas las funciones de control y flujo de los datos. Similarmente como en el transmisor, el módulo receptor cuenta con un módulo que permite modificar la configuración del receptor por medio de un programa terminal de computadora, particularmente el Hyperterminal de la plataforma Windows.

2.3.2 Flujo de los Datos del Módulo Receptor Digital en RF

Tal como observamos en la Fig. 2.3.2.1, los datos seriales que fluyen en el módulo receptor son controlados por un microcontrolador PIC16F873, que es programado para realizar todas las funciones del módulo Receptor que se mencionan posteriormente en el apartado 2.3.3.

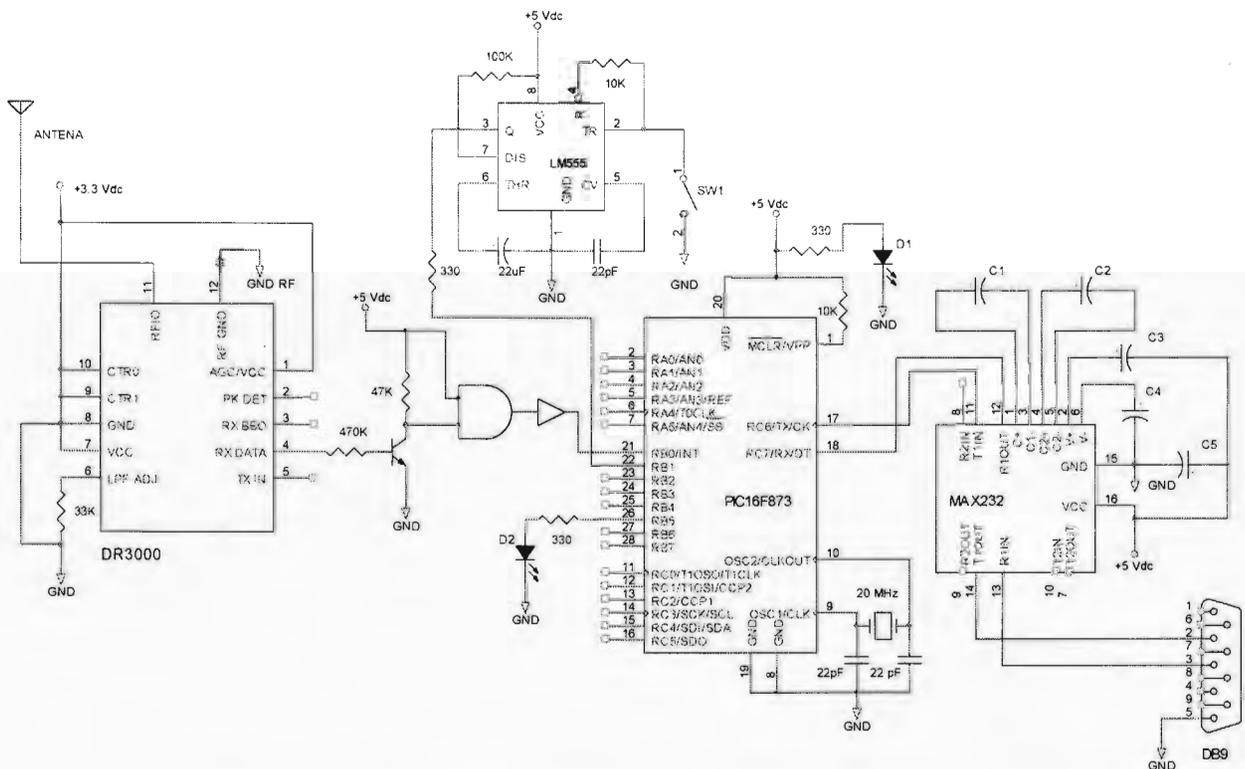


Fig. 2.3.2.1 Diagrama Circuital del Módulo Receptor Digital en RF

En la Fig. 2.3.2.1, los datos seriales en banda base proporcionados por el módulo DR3000 son monitoreados por el microcontrolador por medio del pin PB0/INT0, de interrupción externa y configurado para activarse en su flanco de subida. Hay que

resaltar que se realizó una etapa de acondicionamiento entre el módulo DR3000 y el microcontrolador PIC, por medio de una circuitería buffer y un inversor y así ajustar nuestras señales acorde con una correcta operación del receptor. Al activarse esta rutina de interrupción, se ejecuta una rutina que verifica la cadena de bits de cabecera y cuando ocurre una secuencia determinada comienza a grabar en RAM cada uno de los bits seriales transmitidos que constituyen el paquete de datos codificados. La secuencia preestablecida se compone de la secuencia de dos tiempos de bits activos o "1s" y una cabecera de 6 bits de valor 0x13. Luego de completar la lectura del paquete de datos codificados, se realizan las rutinas de decodificación y transmisión serial al dispositivo destino de la comunicación, para ello se utiliza el periférico UART del microcontrolador a la velocidad fijada por el sistema global y un convertidor de niveles TTL-RS232, un circuito integrado RS232 y una interface física serial DB9. Similarmente como en el transmisor se debe tener cuidado de conectar la señal del microcontrolador PIC de transmisión - pin RC6, con el pin de recepción de la interface DB9, es decir el pin 2 y la señal de recepción del microcontrolador PIC - pin RC7, con el pin de transmisión de la interface DB9, es decir el pin 3.

Una de las razones importantes del uso de una interrupción externa es porque nos permite la sincronización del reloj del receptor y así evitar desfasamientos en las lecturas de cada bit del bloque total de datos codificados.

Detalles del flujo y procesamiento de los datos se verán con detalle en las rutinas de control del receptor. Podemos mencionar que igual que en el transmisor se implementó un pulsador con las mismas características que permite modificar los parámetros de comunicación serial.

2.3.3. Rutinas del Programa de Control del Módulo Receptor Digital en RF

Las rutinas y programas de control del módulo receptor digital en RF residen en el PIC, que es un microcontrolador de 8 bits, de arquitectura RISC y que usa un cristal externo de 20 MHz, alcanzando con ello un ciclo de instrucción de 0.2 μ s, lo que nos permite implementar rutinas rápidas y acordes con el propósito particular de comunicación.

PROGRAMA DE CONTROL DEL RECEPTOR DIGITAL EN RF

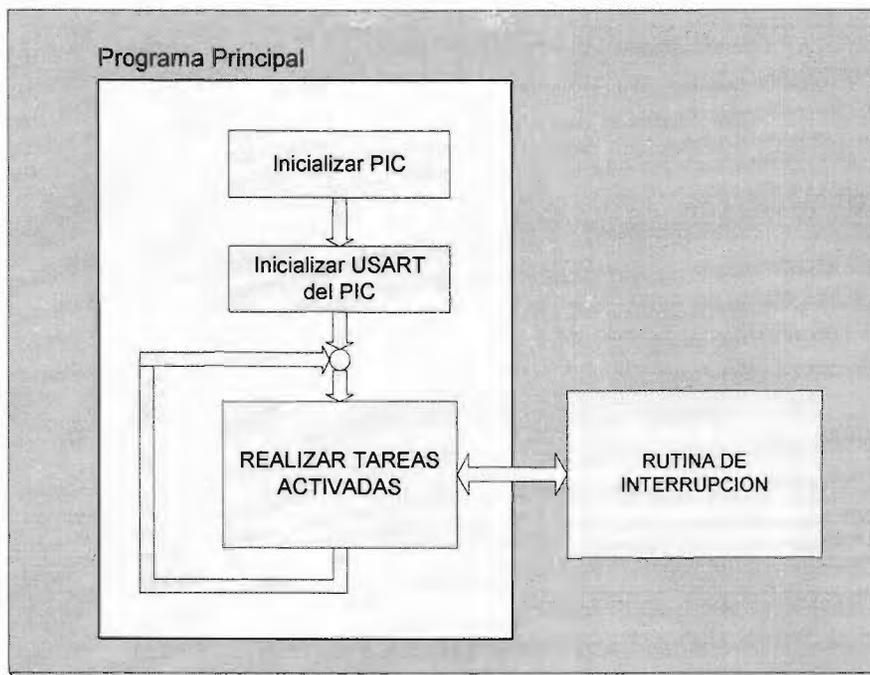


Fig. 2.3.3.1 Diagrama de Bloques del Programa de Control del Módulo Receptor Digital en RF

En la Fig. 2.3.3.1, se observa el diagrama de bloques del programa de control del receptor digital en RF, que básicamente se compone de dos bloques: Programa Principal y Rutina de Interrupción. El Programa Principal se encarga de inicializar el microcontrolador, y entre las configuraciones podemos mencionar que se define el acceso indirecto a los bancos 0 y 1 de la memoria RAM del microcontrolador; se deshabilita interrupciones de todos sus periféricos, solo se centra en atender la interrupción generada por el pin RB0/INT0 cuando ocurra un flanco de subida el cual dispara la activación de la interrupción del temporizador TMR0, encargada de leer el bloque de datos; se define el puerto A como un puerto digital de salida; se definen los terminales PB0, PB1 y PB2 como entradas digitales y el resto de terminales del puerto B como salidas digitales, además de habilitar el arreglo pull-up sobre este puerto. De igual manera se inicializa el periférico UART del microcontrolador PIC, a una velocidad de 9600 bps, 8 bits de datos, no paridad, 1 bit de inicio y 1 bit de parada, no se usan sus disparos de interrupción. El programa principal después de inicializar las variables

de control ingresa a un lazo y monitorea que tareas están activas y procede a realizarlas. Las tareas a realizar tienen una prioridad preestablecida y se inician con el disparo de una bandera generada en la rutina de interrupción. Las tareas disparadas por la interrupción son para decodificar y transmitir los datos decodificados de forma serial al dispositivo destino, detalles que se verán más adelante. La Rutina de Interrupción inicialmente se dispara con un flanco de subida en el Terminal RB0/INT, esta rutina dispara a la rutina de interrupción del temporizador TMR0 para generar el sincronismo necesario en la lectura de cada uno de los bits del paquete de datos, es aquí también donde se verifica la cabecera y si cumple se lee el bloque completo, caso contrario se deshabilita y activa nuevamente la interrupción externa INT0. En el caso de que la cabecera sea la correcta se termina de leer el paquete de datos, se activa una bandera que dispara otras rutinas en el programa principal, mayores detalles se verán en el apartado 2.3.3.1.

Además, de forma análoga como para el módulo transmisor digital en RF, se utilizó el kit de desarrollo de Microchip, el PICSTART PLUS. Este kit de desarrollo es un sistema de desarrollo que incluye el hardware externo del grabador y el software de desarrollo integrado MPLAB versión 5.1.

2.3.3.1 Rutina Principal del Programa: *Pprincipal*

Esta rutina del programa se encarga de realizar las configuraciones del modo de funcionamiento del microcontrolador PIC así como la inicialización de los periféricos a usar.

Previamente a esta rutina se realizó la definición de una Tabla para la Decodificación de los datos recibidos, además se definen cadenas de texto usadas como interface con el usuario, todos estos datos se almacenan en la memoria de programa del microcontrolador.

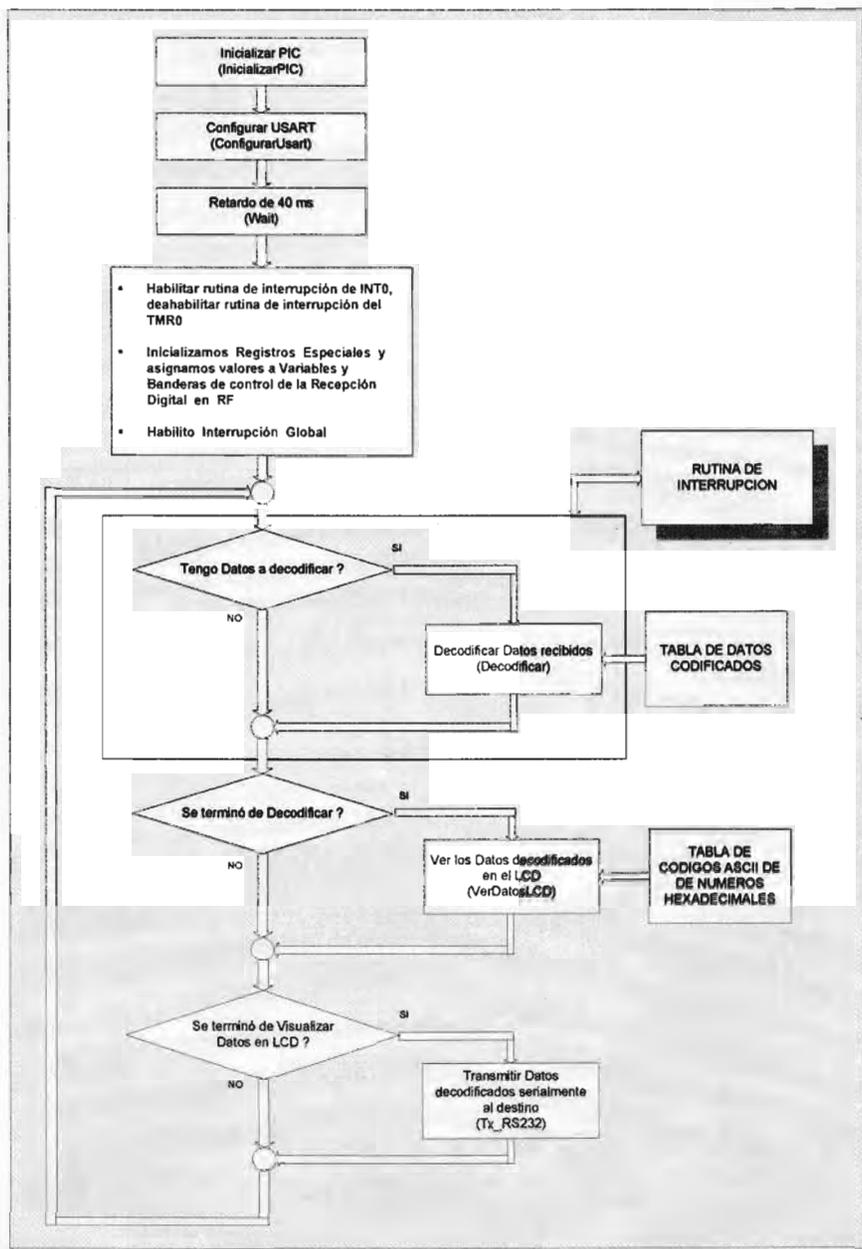


Fig. 2.3.3.1.1 Diagrama de Bloques de la Rutina del Programa Principal

Las siguientes descripciones hacen referencia al diagrama de flujo de la rutina del Programa principal mostrado en la Fig. 2.3.3.1.1. Se llama a la rutina *InicializarPIC* que se encarga de configurar los puertos A y E como terminales digitales, los terminales del puerto B como salidas digitales y asignarle el arreglo de resistencias pull-up. El terminal PB0, etiquetado como RxCa en el programa, es usado para la recepción serial del

paquete de datos. Se configura la asignación del reloj interno y divisor de frecuencia a 2 al Temporizador TMR0. Se configuran los terminales RC7/XR y RC6/TX del puerto C como entrada y salida, respectivamente, tal como se requiere el uso del periférico de comunicación serial (USART) del microcontrolador.

De manera similar que el módulo Transmisor Digital en RF, la rutina *ConfigurarUsart* para configurar y habilitar el puerto serial USART del microcontrolador a una velocidad de 9600 bps, 8 bits de datos, no paridad, 1 bit de inicio y 1 de parada. Para definir la velocidad de comunicación del USART del PIC se requiere definir el bit BRGH y el registro SPBRG, particularmente se seleccionó el bit BRGH=1 y se asignó el valor decimal D'129', lo que corresponde a una velocidad de 9600 bps. Por diseño del software de control del puerto se deshabilitó las interrupciones de la transmisión y recepción.

Luego se llama a la rutina *Wait*, que genera un retardo de 40 ms aproximadamente, logrando estabilizar los voltajes del microcontrolador y sus periféricos, es un retardo recomendable para la correcta operación de los puertos del microcontrolador.

Continuando con el programa principal se habilita la recepción del USART del PIC, habilitamos la rutina de interrupción ligada a la interrupción externa del pin PB0, deshabilitamos la rutina de interrupción del temporizador TMR0, esto es necesario para la correcta operación del diseño del programa de control.

Continuamos con la inicialización de valores a las variables FSR_DATOS, FSR_INTER, REGRX. Además fijamos a 0 las banderas F1isrT0 y F1isrT1 usadas en el lazo de la rutina de interrupción del temporizador TMR0, tal como veremos mas adelante. Se fijan a 0 las banderas T1_Deco, T2_VLCD y T3_RS232, banderas que controlan la realización de tareas relacionadas a la decodificación, visualizar datos decodificados en un LCD y transmitir los datos decodificados por el UART del PIC. La visualización en un LCD constituye una forma de monitorear los datos que llegan en el receptor y así depurar nuestra programación, pero el diseño final no contempla el incluir un LCD.

Luego se habilita la interrupción Global del microcontrolador PIC para que pueda responder con cualquier interrupción habilitada, en nuestro caso a la interrupción externa en el flanco de subida del terminal PB0.

El lazo importante a considerar en esta secuencia del programa es el monitoreo de las banderas del registro de tareas, etiquetadas como T1_Deco, T2_VLCD y T3_RS232, que realizan las rutinas Decodificar, VerDatosLCD y Tx_RS232, respectivamente. La ejecución de estas rutinas-tareas se realizan de manera consecutiva, es decir el termino de una de ellas genera el inicio de otra rutina, todo este lazo es habilitado desde la rutina de interrupción del temporizador TMR0 después de realizar la lectura completa del bloque de datos recibidos.

Cada tarea se encarga de deshabilitarse y reiniciar todos los registros de control utilizadas en cada rutina de programa de cada tarea. Al final cada tarea,

2.3.3.2 Rutina de Decodificación: *Decodificar*

El diagrama de flujo de esta rutina se muestra en la Fig. 2.3.3.2.1, y tiene como función el de decodificar cada nibble recibido por la rutina de Interrupción y validando el dato con el CCR enviado, si se detecta un error el bloque de datos se ignora y no transmite al dispositivo destino. Si el dato es verificado como válido se transmite al dispositivo serial en formato RS232.

Tal como apreciamos en el código ensamblador de la rutina Decodificar, que muestra la Fig. 2.3.3.2.2, se inicia limpiando la bandera T1_Deco (generada en la rutina de interrupción) y almacenamos el registro FSR del contexto de la rutina principal, porque será modificado tanto para acceder a los datos recibidos como a la dirección de la memoria RAM del microcontrolador PIC donde se almacenarán los datos ya decodificados. Luego es necesario reiniciar los registros: NN_DECO, que almacena el número de nibbles a decodificar, que componen el paquete de información; el registro FSR_DATOS que almacena el puntero de los datos decodificados a ser almacenados en el banco 1 de la memoria RAM. Hay que hacer notar que se utiliza principalmente el

acceso indirecto para poder acceder a cada nibble recibido en el bando 0 de la RAM y al lugar de almacenamiento de los bytes decodificados del banco 1.

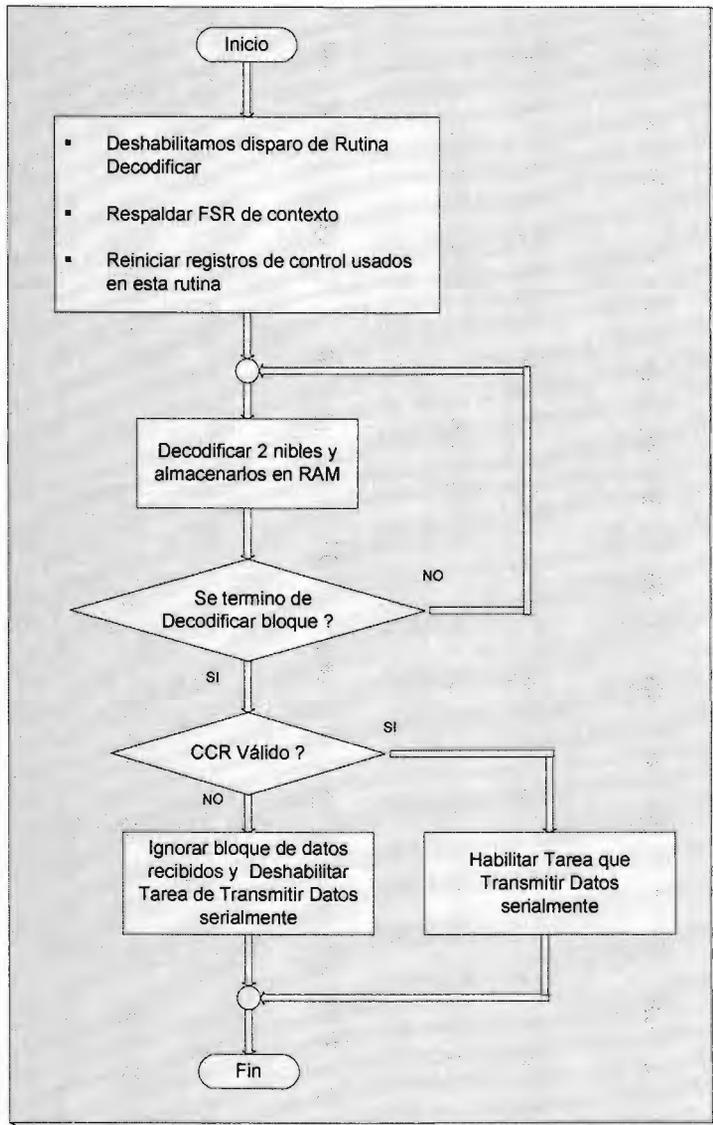


Fig. 2.3.3.2.1 Diagrama de Bloques de la Rutina de Decodificación

Se utiliza una subrutina ConvertirNibble dentro de la rutina Decodificar, que realiza la conversión de un byte a un nibble del registro decodificado. Básicamente hace la comparación con dos valores hexadecimales 0x15 y 0x0F que corresponden a los nibbles 0x00 y 0x0F, o en todo caso realiza el AND lógico con 0x0F. Y después de llamarla dos veces, se forma el byte de datos original del lado del transmisor. Esta

rutina tiene como parámetro de entrada el registro DECO_TMP que contiene el byte codificados por el transmisor (6 bits útiles), este registro es convertido a su nibble original y lo retorna en el mismo registro DECO_TMP, tal como lo muestra en la Fig. 2.3.3.2.2.

```

; Tarea: Decodificar
Decodificar
    BANK0
    BCF          T1_Deco      ; Deshabilito Rutina

Decodificar
    MOVF        FSR,W
    MOVWF       FSR_DECO
    MOVLW       VNN_DECO
    MOVWF       NN_DECO
    MOVLW       VFSR_DATOS  ;Puntero de FSR de Datos

Decodificados
    MOVWF       FSR_DATOS
    MOVLW       VFSRINT+1
    MOVWF       FSR

OtraDecodi
    MOVF        INDF,W
    MOVWF       DECO_TMP
    CALL        ConvertirNibble
    MOVF        DECO_TMP,W
    MOVWF       REGDECO

    INCF        FSR,F
    MOVF        INDF,W
    MOVWF       DECO_TMP
    CALL        ConvertirNibble

    SWAPF      DECO_TMP,W
    ADDWF      REGDECO,F
    INCF        FSR,F

; Grabamos en RAM dato Decodificado
GrabamosRAM
    MOVF        FSR,W
    MOVWF       FSR_TEMP
    MOVF        FSR_DATOS,W
    MOVWF       FSR
    MOVF        REGDECO,W
    MOVWF       INDF

    INCF        FSR_DATOS,F
    MOVF        FSR_TEMP,W
    MOVWF       FSR

; Terminamos de Decodificar?
    DECF        NN_DECO,F
    GOTO        OtraDecodi
    GOTO        Fin_RDecodi

Ver_AHORA
; Sumamos datos para formar el CCR
    MOVLW       VFSR_DATOS
    MOVWF       FSR
    MOVLW       0x08
    MOVWF       NN_DECO

Otra_SCCR
    MOVF        INDF,W
    CALL        SumarCCR
    INCF        FSR,F
    DECF        NN_DECO,F

    GOTO        Otra_SCCR
    GOTO        Fin_RDecodi

; Comprobamos CCR Transmido con el Recibido
    MOVLW       VFSR_DATOS+0x08
    MOVWF       FSR
    MOVF        INDF,W
    XORWF       CCRL,W      ; Comprobamos CCRL, son iguales?
    BTFSS      STATUS,Z    ; Si
    GOTO        Fin_NoDato ; No
    GOTO        CCRcorrectos

    INCF        FSR,F
    MOVF        INDF,W
    SUBWF       CCRH,W      ; Comprobamos CCRH, son iguales?
    BTFSS      STATUS,Z    ; Si
    GOTO        Fin_NoDato ; No
    GOTO        CCRcorrectos

CCRcorrectos
    BSF         T2_VLCD     ; Habilitamos Rutina de
    Visualizar Datos Recibidos
    GOTO        Fin_RDecodi

Fin_NoDato
    BCF         T2_VLCD     ; Deshabilito Rutina de
    Visualizar Datos Recibidos
    GOTO        Fin_RDecodi

; Habilitamos Rutina de
    BSF         T2_VLCD
    Visualizar Datos Recibidos
    MOVF        FSR_DECO,W  ; Regresamos valor FSR de lazo
    principal
    MOVWF       FSR
    GOTO        Ver_T2

;***** ConvertirNibble *****
; Rutina que identifica el nibble Tx y lo convierte en nibble
; menos significativo, Entrada y salida: DECO_TMP
ConvertirNibble
    BANK0
    MOVLW       0x15
    XORWF       DECO_TMP,W
    BTFSC      STATUS,Z
    GOTO        FijarReg1

    MOVLW       0x1C
    XORWF       DECO_TMP,W
    BTFSC      STATUS,Z
    GOTO        FijarReg2

    MOVLW       0x0F
    ANDWF      DECO_TMP,F
    GOTO        Fin_CN

FijarReg2
    MOVLW       0x0F
    MOVWF      DECO_TMP
    GOTO        Fin_CN

FijarReg1
    CLRF       DECO_TMP
    GOTO        Fin_CN

Fin_CN
    RETURN
    
```

Fig. 2.3.3.2.2 Código Asembler de la Rutina de Decodificación

Luego de terminar de grabar los datos decodificados, se encuentra la suma CCR que permite verificar con el CCR recibido y son diferentes, existe error y no se habilita la transmisión al dispositivo destino, para luego reiniciar todos los registros usados por esta rutina.

Si los bytes del CCR son iguales tanto del lado transmisor como receptor, se habilita la tarea Tx_RS232, poniendo a "1" la bandera T3_RS232, rutina que se encarga de transmitir los bytes decodificados, sin incluir el CCR.

2.3.3.3 Rutina de Transmisión de Datos decodificados: Tx_RS232

Esta rutina se encarga de transmitir los datos previamente decodificados y validados, hacia un dispositivo serial en formato RS232. Para ello el microcontrolador PIC usa el UART configurado e inicializado en rutinas previas que definen una velocidad del sistema o dispositivo a interconectar, con 8 bits de datos, sin paridad, un bit de inicio y un bit de parada.

Al inicio del programa se respalda el valor del FSR de contexto del programa principal, para luego modificarlo por el valor de la etiqueta VFSR_DATOS, dirección donde ya se encuentran los datos decodificados y listos para su transmisión. Los datos son transmitidos en bytes consecutivos, evitando se sobrescriban, para ello se polea el valor del bit TRMT. El detalle en código ensamblador de la rutina TX_RS232 se muestra en el Anexo B.

2.3.3.4. Rutina de Configuración de Parámetros del Receptor: Set_Parametros_Com

De manera similar que en el módulo transmisor digital en RF, esta rutina permite configurar parámetros del transmisor, su ejecución esta ligada a un switch que genera un pulso y activa esta rutina. Para mayor referencia podemos ver el apartado 2.2.4.5, Rutina para la Configuración de Parámetros del Transmisor: *Set_Parametros_Com*, ya que el hardware y el programa son iguales tanto para el transmisor como para el receptor digital en RF.

**ESTA TESIS NO SALE
DE LA BIBLIOTECA**

Igualmente se requiere habilitar y configurar el programa Hyperteminal de la plataforma Windows, como un enlace serie asíncrono RS232 con velocidad de 19200 bps, 8 bits de datos, sin paridad, 1 bit de inicio y 1 bit de parada. Lo requerimos para poder visualizar los mensajes que envía el microcontrolador y poder interactuar con el programa de configuración de parámetros.

2.3.3.5 Rutina de Interrupción del Módulo Receptor Digital en RF: *RutinaInterrupcion*

Esta rutina de interrupción se encuentra diseñada para responder a una interrupción externa en un flanco de subida del voltaje sobre el terminal RB0/INT0 y así sincronizar y habilitar la rutina de interrupción para el temporizador TMR0 que controlará los tiempos de lectura de los bits que componen el paquete de datos recibidos en el módulo DR300 del receptor. Para hacer la lectura completa del paquete de datos, debe cumplirse los requerimientos de bit de inicio y cabecera, de lo contrario se reinicia su modo de funcionamiento y sigue esperando un flanco de subida como al inicio.

La ISR de INT0 fija un tiempo inicial de activación de la ISR del TMR0 a 26 μ s, pero luego al ejecutarse la ISR del TMR0 se fija a un periodo constante de 52 μ s que corresponde a un tiempo de bit de una comunicación serial a 19200 bps.

El detalle en código ensamblador de la rutina de interrupción del módulo receptor digital en RF se encuentra en el Anexo B, que servirá de guía en la siguiente descripción de la rutina. Como podemos observar en su diagrama de flujo de la Fig. 2.3.3.5.1, la rutina de Interrupción inicia respaldando los valores de los registros de control dentro del contexto de la rutina Principal, cuya modificación alteraría drásticamente el correcto funcionamiento del módulo receptor, es decir los registros W, STATUS y FSR. Estos valores son nuevamente recuperados al final de la rutina de interrupción. De igual manera, se carga el valor del FSR usado en esta rutina de interrupción con el contenido del registro FSR_INTER.

Inicialmente la ISR del INT0 esta habilitada, luego de detectar el flanco de subida deshabilitamos esta interrupción y habilitamos la interrupción del temporizador TMR0 y que se dispare dentro de medio tiempo de bit, en este caso dentro de 26 μ s. Además se fija a "1" la bandera F1isrT0 que habilita la lectura del primer bit en la siguiente interrupción del TMR0.

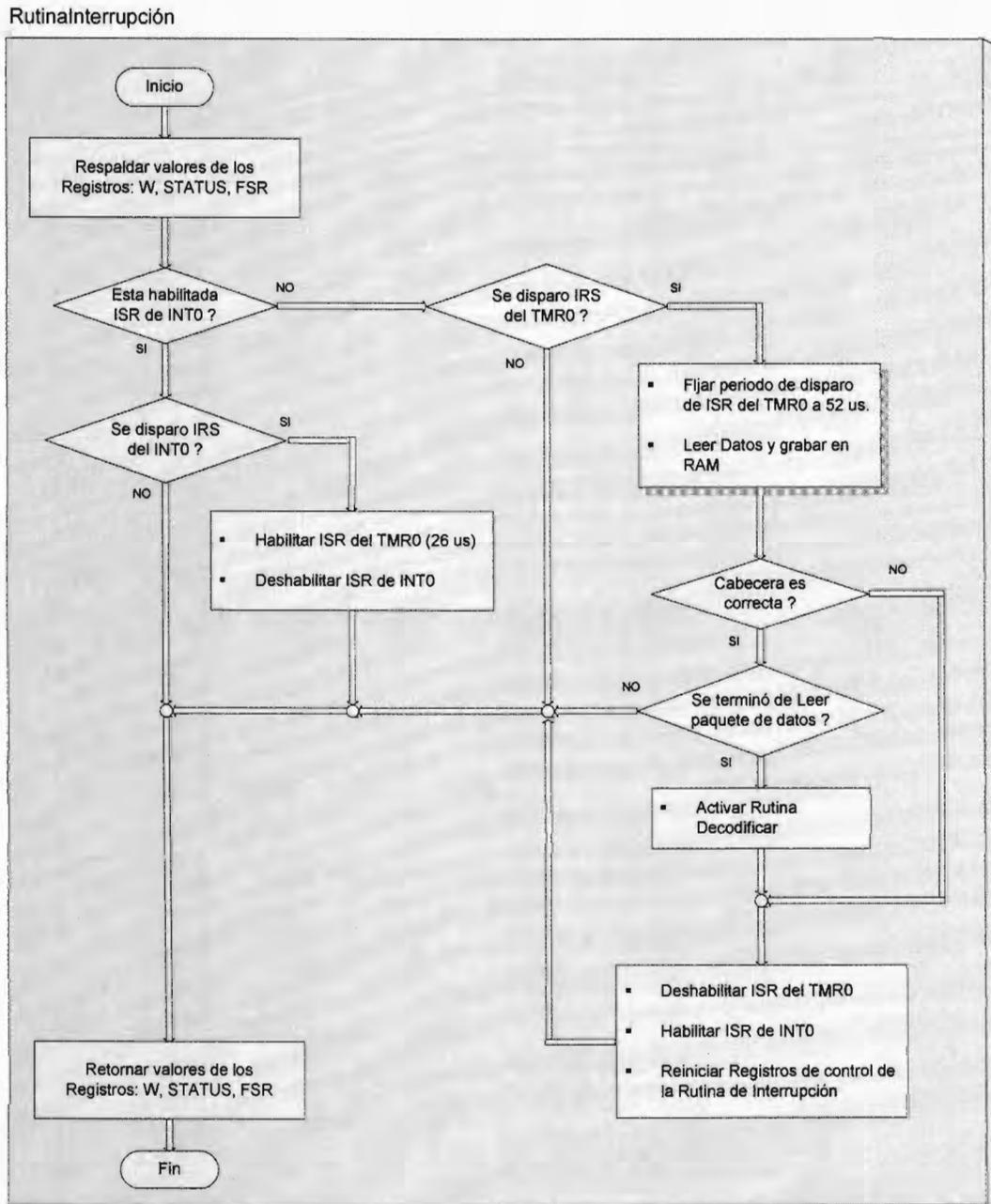


Fig. 2.3.3.5.1 Diagrama de Bloques de la Rutina de Interrupción del Módulo Receptor

En el siguiente disparo de la rutina de interrupción que corresponde a la ISR del TMR0, se lee el valor del bit en el terminal PB0, el cual debe ser "1", caso contrario deshabilita la interrupción generada por el TMR0 y habilita nuevamente la interrupción externa del terminal PB0. Si se lee dos "1" de manera consecutiva fija la bandera F1isrT0 = 0 y la bandera F2isrT0 = 1, lo que permite iniciar la lectura de los nibbles de la cadena de bits codificados provenientes del módulo de radiofrecuencia. Luego se reinician los valores de los registros CON_PRED, NNIBLES y REGRX. Hay que recalcar que se configura el TMR0 para que dispare en cada tiempo de bit, en este caso cada 52 μ s, para ello fijamos el valor del registro TMR0 = 0x88.

Después de leer el primer nibble este es comparado con la cabecera, si son iguales continúa con la lectura de los bits seriales de los datos codificados. El valor del registro CON_PRED controla el número de lectura de cada nibble codificado, el valor del registro NNIBLES controla el número total de nibbles a leer.

Después de terminar de leer todo el paquete de datos, se reinician los valores del contenido de los registros de control del programa del módulo receptor, tal como estaban antes del disparo del ISR de INT0 y queda a la espera de otro ciclo de lectura del paquete de datos. Se fijan las banderas F1isrT0 = F2isrT0 = 0. Al final de la lectura del paquete de datos se fija la bandera de tarea T1_Deco = 1, lo que habilita la ejecución de la rutina Decodificar.

2.4 Integración de los Módulos Transmisor y Receptor en RF

2.4.1 Descripción de los Módulos Transmisor y Receptor Inalámbrico en RF

Después de diseñar por separado cada módulo de comunicación, se requiere integrar los módulos transmisor y receptor que conforman en transceiver, tal como se mostró en la Fig. 1.4.1. Para ello el transceiver se fija su funcionamiento en el modo Receptor y sólo cuando se tiene datos a transmitir se cambia al modo Transmisor, con la condición que cada modo no será interrumpido hasta completar su tarea. El diagrama de flujo de las rutinas que controlan esta integración se muestra en la Fig. 2.4.1.1, donde apreciamos que en la Rutina Principal genera la activación del modo Transmisor o

Receptor. Después que se habilita el modo de funcionamiento, la Rutina de Interrupción genera todas las rutinas ya sea en modo Transmisor o Receptor, tal como se describe en los apartados 2.2 y 2.3.

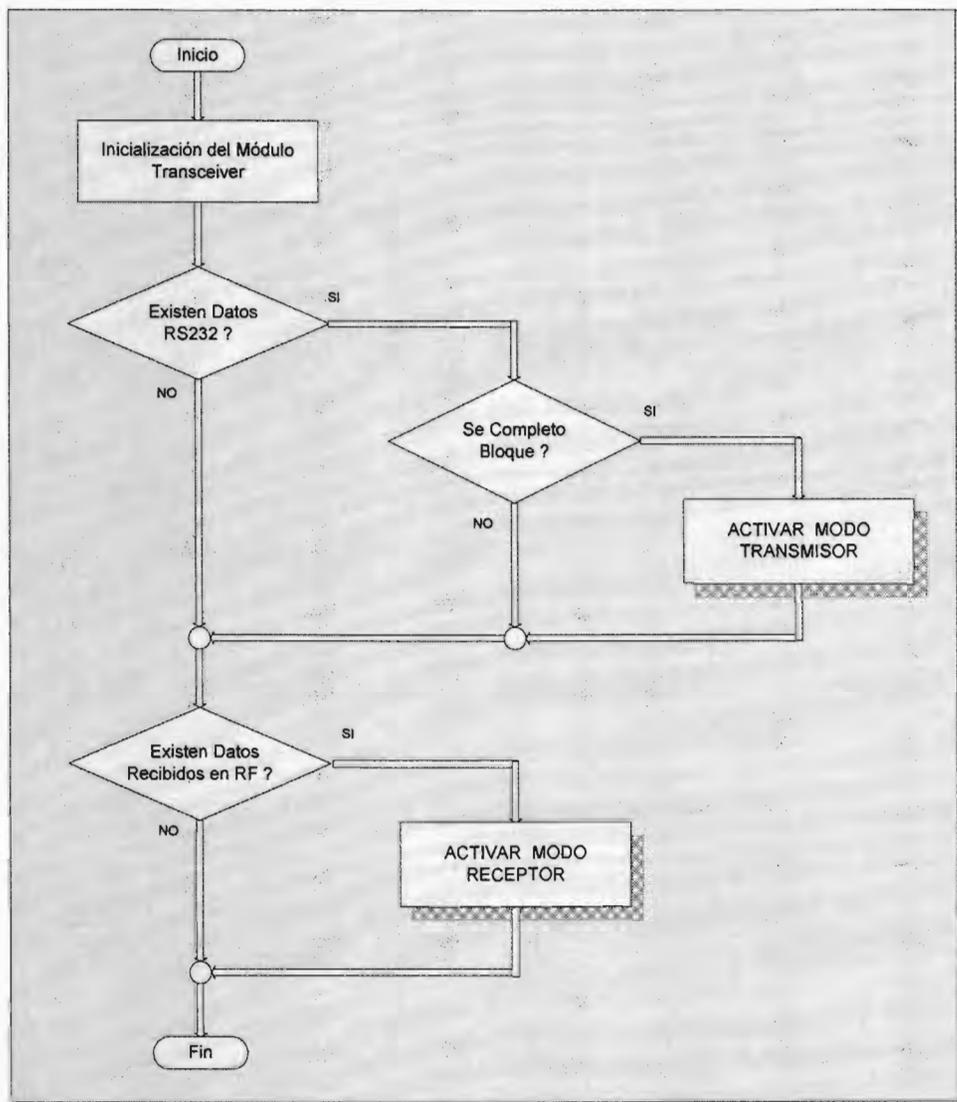


Fig. 2.4.1.1 Diagrama de Bloques de la Rutina Principal del Transceiver Digital en RF

Para el modo Transmisor se agrega dentro del paquete de datos a transmitir un preámbulo de 4 bytes con valor hexadecimal 0x55, para generar un balance en AC y DC del lado del Receptor. Después de transmitir se regresa que al modo por defecto que es el modo Receptor.

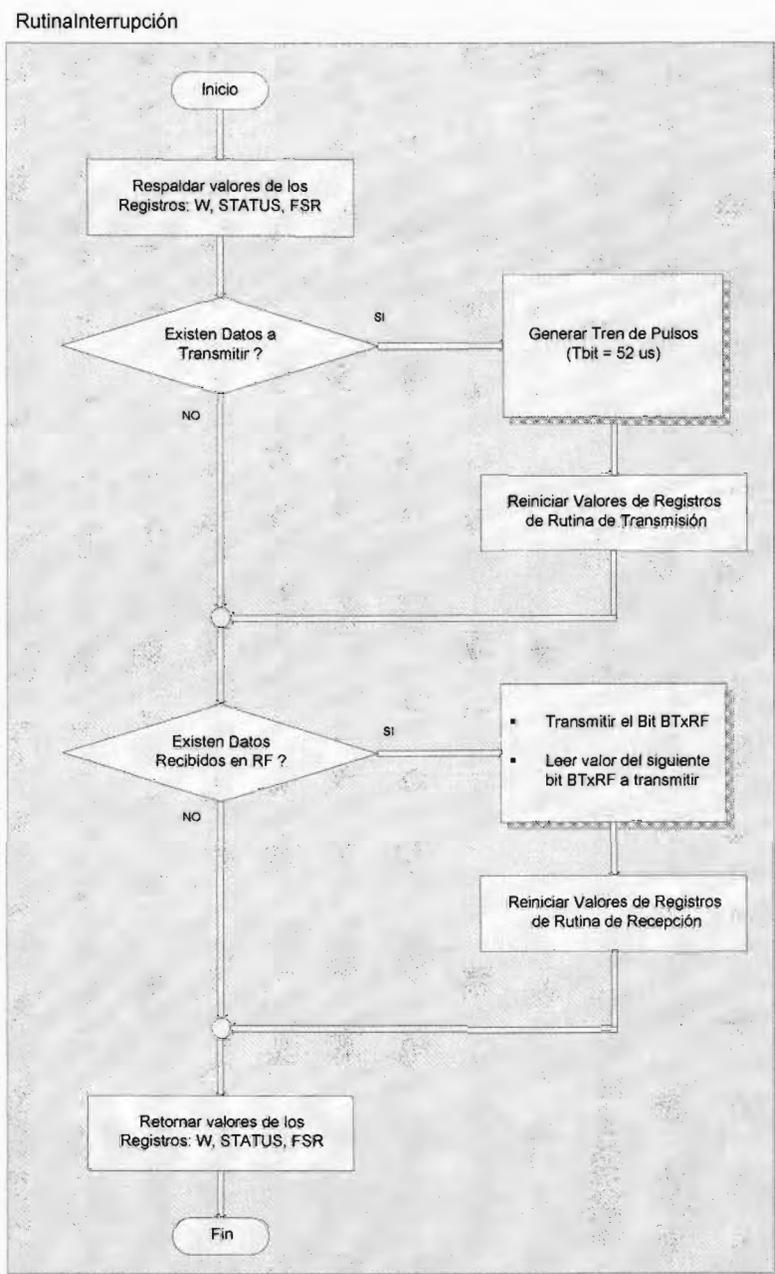


Fig. 2.4.1.2 Diagrama de Bloques de la Rutina de Interrupción del Transceiver Digital en RF

2.4.2 Implementación de los Módulos Transceivers Inalámbricos en RF

Los transceivers fueron implementados integrando los módulos transmisor y receptor mostrado en los diagramas de las Figuras 2.2.3.1 y 2.3.2.1 , tal como se muestran en las Figuras . Igualmente, se realizó la conexión de una antena de celular en el puerto de entrada/salida de RF del módulo DR3000. Se requirió agregarle un plano de tierra para un mejor enlace en RF, la implementación del módulo transmisor y receptor se muestran en las figuras 2.4.2.1 y 2.4.2.2.

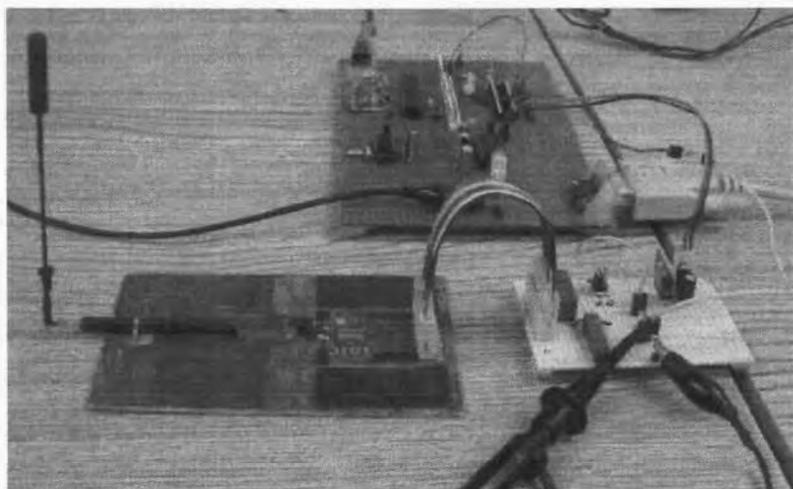


Fig. 2.4.2.1 Implementación del Transceiver - Transmisor

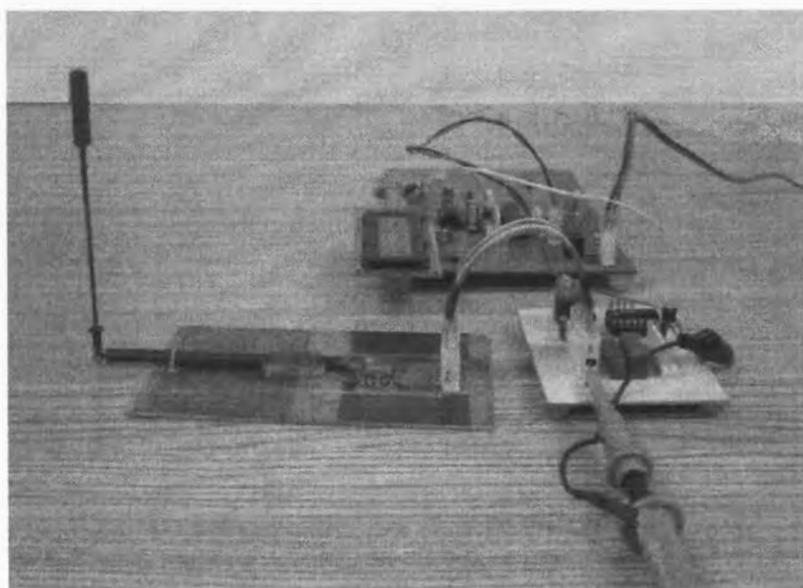


Fig. 2.4.2.2 Implementación del Transceiver - Receptor

Se realizó una prueba que nos asegure una correcta implementación de los módulos transceivers, esto se logró transmitiendo un tren de pulsos por el módulo transceiver transmisor y observar la señal que se recibe del lado del módulo transceiver receptor. Tal como se muestra en la Fig. 2.4.2.3, se conecta dos canales de un osciloscopio para poder ver las señales a monitorear. Se observa en la Fig. 2.4.2.4 el trazo del transceiver transmisor y del transceiver receptor, es decir, la onda transmitida (trazo superior) y la onda recibida (trazo inferior).

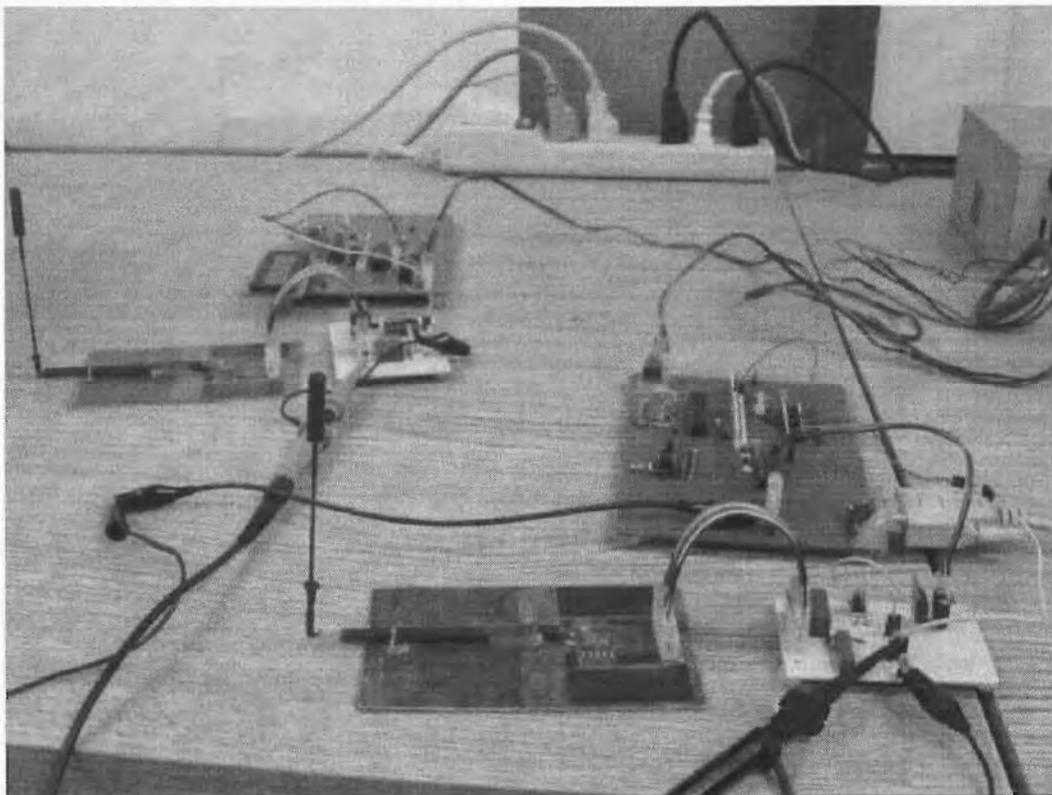


Fig. 2.4.2.3 Enlace de los módulos Transceivers

Este monitoreo se realizó a la entrada del modulo DR3000 del transceiver transmisor y la salida del módulo DR3000 del transceiver receptor.

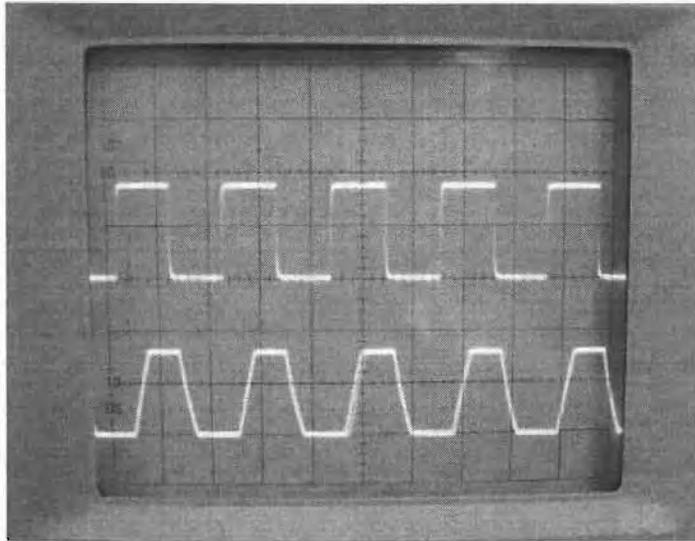


Fig. 2.4.2.4 Monitoreo de señal Transmitida (pulsos superiores) y Recibida (pulsos inferiores)

CAPITULO III

Pruebas y Resultados

3 PRUEBAS Y RESULTADOS DEL TRANSCEIVER DIGITAL EN RF

3.1 Ambiente de Desarrollo MPLAB IDE de Microchip

El MPLAB IDE versión 5.1, es una herramienta de desarrollo proporcionada por el fabricante de los microcontroladores PIC. Es un ambiente de desarrollo integrado que nos permite editar, depurar, compilar y generar un archivo de código máquina, un archivo hex. Este archivo de extensión hex puede ser usado por el simulador del MPLAB para su depuración. El archivo hex depurado es utilizado por el grabador para descargar o programar los PICs. El MPLAB funciona con grabadores fabricados por Microchip, en este caso se utilizó el PICStar Plus, que se muestra en la Fig. 3.1.2 ^[12].

Cada uno de los diagramas de flujos considerados en el capítulo anterior fueron simulados usando el MPLAB. Tal como se observa en la Fig. 3.1.1, el MPLAB proporciona una serie de ventanas que nos permiten ver el contenido de todas las ubicaciones de memoria del programa y de datos. Igualmente, el MPLAB tiene la opción de ejecutar el programa por medio de pasos, puntos de detección, puntos de

disparo estándar y complejos, herramientas que fueron utilizadas para la evaluación de todo el programa que compone el transceiver ^{[11][12]}. Observamos en la parte de la izquierda el código fuente en ensamblador que puede ser simulado y así depurar cada parte de las rutinas que componen el transceiver. Podemos definir pantallas para observar el contenido de los registros de la RAM, mostrados en una ventana del lado derecho, con color azul; igualmente podemos ver diversas variables de interés, así como la ventana de entradas asíncronas y contenido de la pila con las direcciones de retorno de rutinas.

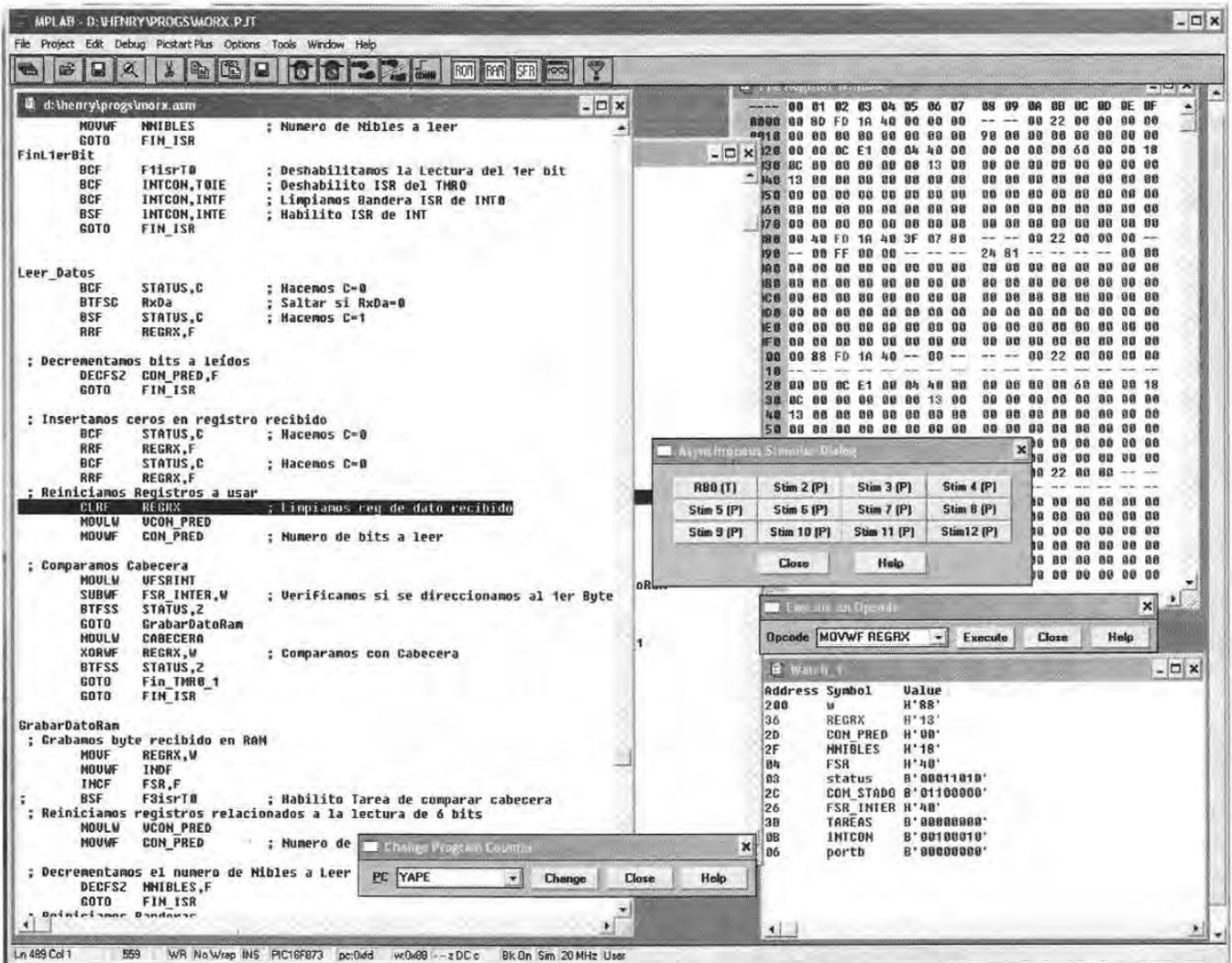


Fig. 3.1.1 Pruebas de Simulación del Microcontrolador en Ambiente de Desarrollo MPLAB

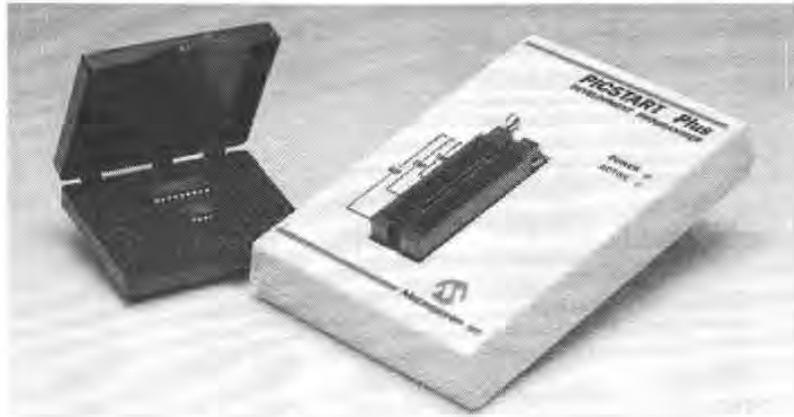


Fig. 3.1.2 Programador PICStart Plus

3.1.1 Simulación de los Módulos Transmisor y Receptor

El simulador del MPLAB fue una herramienta básica para depurar cada una de las rutinas explicadas en los apartados 2.2.4 y 2.3.3 . Las rutinas más importantes de ambos módulos lo constituyen las Rutinas de Interrupción. Para el caso de la Rutina de Interrupción del módulo Transmisor se depuró los períodos de disparo en la transmisión del tren de pulsos, así como se pudo monitorear los valores de cada uno de los registros y el estado de los bits de control de transmisión del paquete de datos. En el caso de la Rutina de Interrupción del módulo Receptor se utilizó la entrada asíncrona del MPLAB que modificaba el estado del pin RB0 que disparaba el inicio de la lectura del paquete de datos, esta rutina se disparaba cada 2 tiempos de bits cuando existía una cadena de bits de ceros y unos.

3.1.2 Monitoreo y Depuración del Disparo de Lectura en el Receptor

Considerando que el transmisor envía un tren de pulsos con un tiempo de bit de 52 μ s, se debe afinar el valor necesario en el registro del temporizador TRM0 del microcontrolador que controla el tiempo de la lectura de cada bit serial en el receptor. El registro TMR0 al cambiar su valor de 0xFF a 0x00 genera una interrupción, acción que aprovechamos para ligarla con una rutina. El tiempo de desbordamiento del TMR0 se calcula mediante la fórmula de la Ec. 3.1 ^[10].

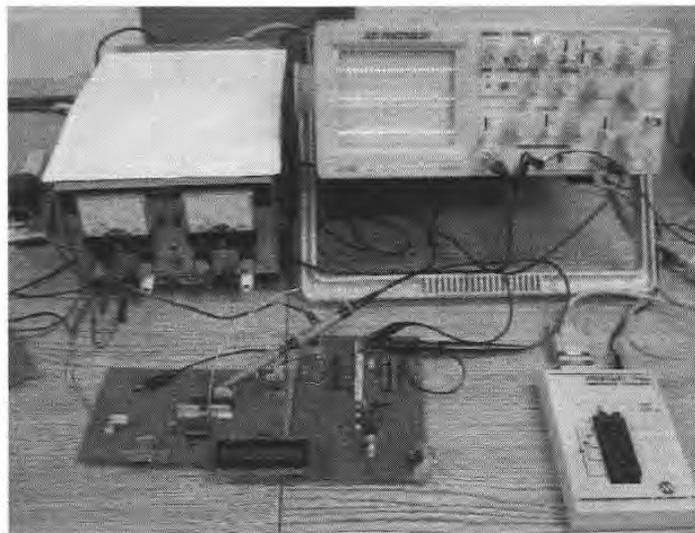
$$T_{int} = \frac{8 * (255 - TMR0 + 1)}{f_{osc}} \quad (\text{Ec. 3.1})$$

donde:

T_{int} = tiempo de interrupción del desbordamiento del TMR0

f_{osc} = frecuencia del oscilador del microcontrolador

Así para generar un período de disparo de la interrupción del TMR0 de 52 μ s, se debe cambiar el valor del registro TMR0 a un valor de 0X7E, esto debe de cambiarse cada rutina de interrupción. A este tiempo debe de considerarse las instrucciones que lleva a cabo antes de realizar esta rutina, que en este caso en particular es aproximadamente 5 μ s. Los cambios que implican en reprogramar el microcontrolador del módulo receptor se comprueban con el osciloscopio, tal como se observa en la Fig. 3.1.2.1, logrando afinar de una manera correcta estos valores. El valor hallado para el registro TMR0 es 0x8A.



(a)



(b)

Fig. 3.1.2.1 (a) y (b) Pruebas de depuración del Módulo Receptor

En la Fig. 3.1.2.2, se observan las ondas recibidas y los pulsos de disparo de lectura de la cadena de bits que conformarían la cadena de bits, por cuestiones de verificación, en esta figura sólo se activaron 11 lecturas de bits. En el osciloscopio, la onda superior es el tren de pulsos generados por el receptor y la onda inferior son los disparos de lectura, generados por el receptor. Hay que recalcar que existe un tiempo inicial de medio tiempo de bit, lo que nos ayuda a ubicar la lectura del bit, justo al centro de cada uno de los bits seriales en el pin RB0.

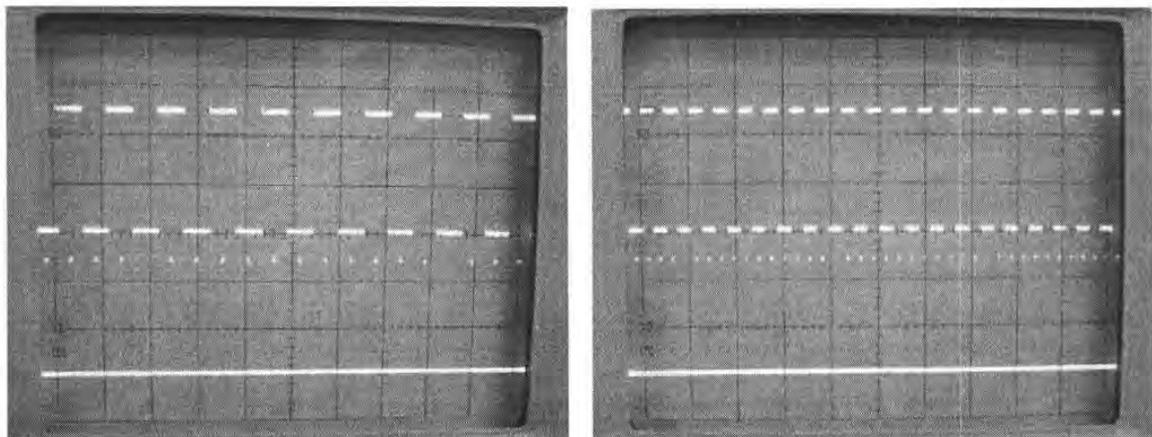


Fig. 3.1.2.2 Disparos de Lectura en el Módulo Receptor

3.2 Pruebas de Transmisión y Recepción

En este apartado se presentan diversas figuras que muestran las pruebas en hardware y software del transceiver.

Hay que considerar que las pruebas se realizaron sobre la implementación de los transceivers tal como se describe en el apartado 2.4.2.

3.2.1 Pruebas de Lectura con un Programa Terminal de PC

El módulo transmisor fue programado para que esté enviando el tren de pulsos con un ancho de pulso de 52 μ s. El receptor fue programado para leer el estado de cada bit en su pin digital, el receptor primeramente detectaba el flanco de subida, accionando una interrupción del microcontrolador. Posteriormente el programa del receptor lee el valor de cada bit serial y lo va ingresando en un byte, para luego enviarlo por el puerto serial para ser mostrado en un programa Terminal, que visualiza los datos en formato ASCII estándar, tal como se muestra en la Fig. 3.2.1.1.

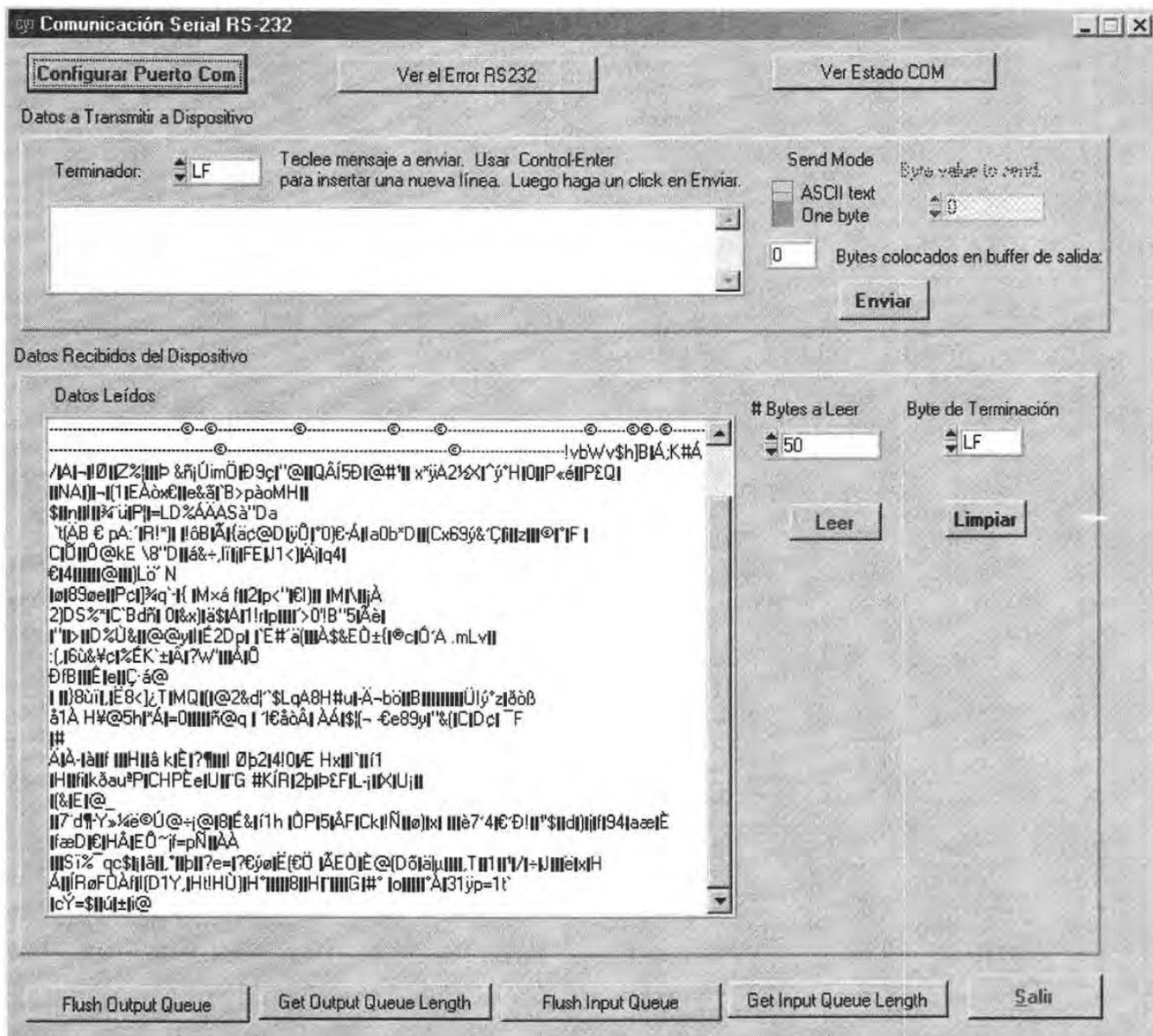


Fig. 3.2.1.1 Lectura de datos recibidos, pantalla 1

Estos resultados sólo permitieron comprobar la rutina del receptor que realizaba la lectura de los datos que presentaba el módulo DR3000, en los valores recibidos no se puede leer ningún patrón de interés. Afinando el tiempo de disparo de lectura en el receptor, cambiando el valor del registro TMR0 a un valor hexadecimal 8E, obtuvimos la lectura con un cierto patrón, tal como se muestra en la Fig. 3.2.1.2.

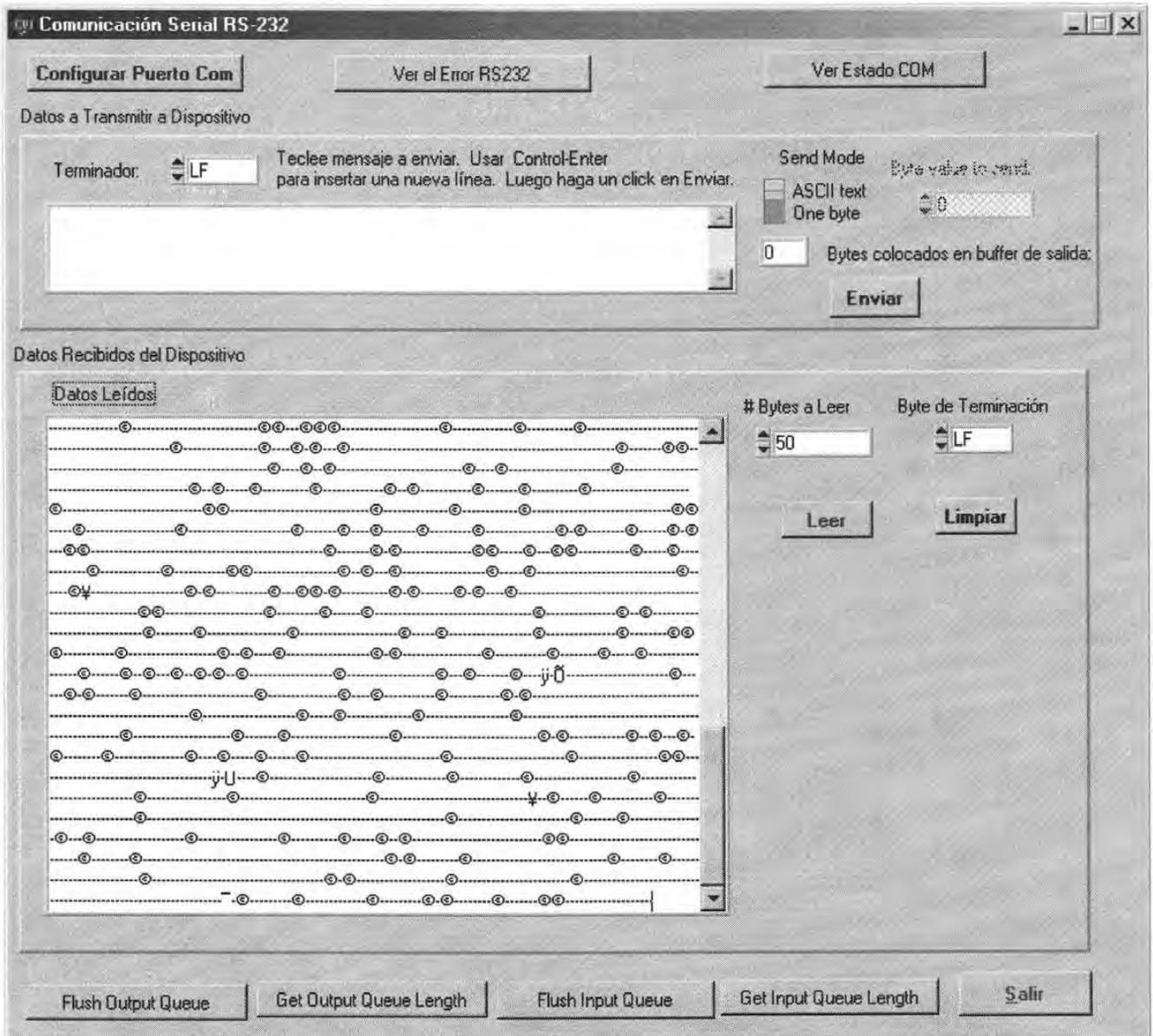


Fig. 3.2.1.2 Lectura de datos recibidos, pantalla 2

Podemos notar que si leemos correctamente el dato tendríamos un valor hexadecimal 0xAA, que es un caracter no imprimible y variable en otro estándar, así que cambiamos al flanco de bajada de lectura de los datos. Tal como se observa en la Fig 3.2.1.3, se obtiene ya casi un patrón, que se trata de la letra R, que corresponde al valor hexadecimal 0x52, lo cual nos indica que existe un pequeño desfase en el período de lectura del receptor, ya que en la lectura correcta sería 0x55 que corresponde a la letra U.

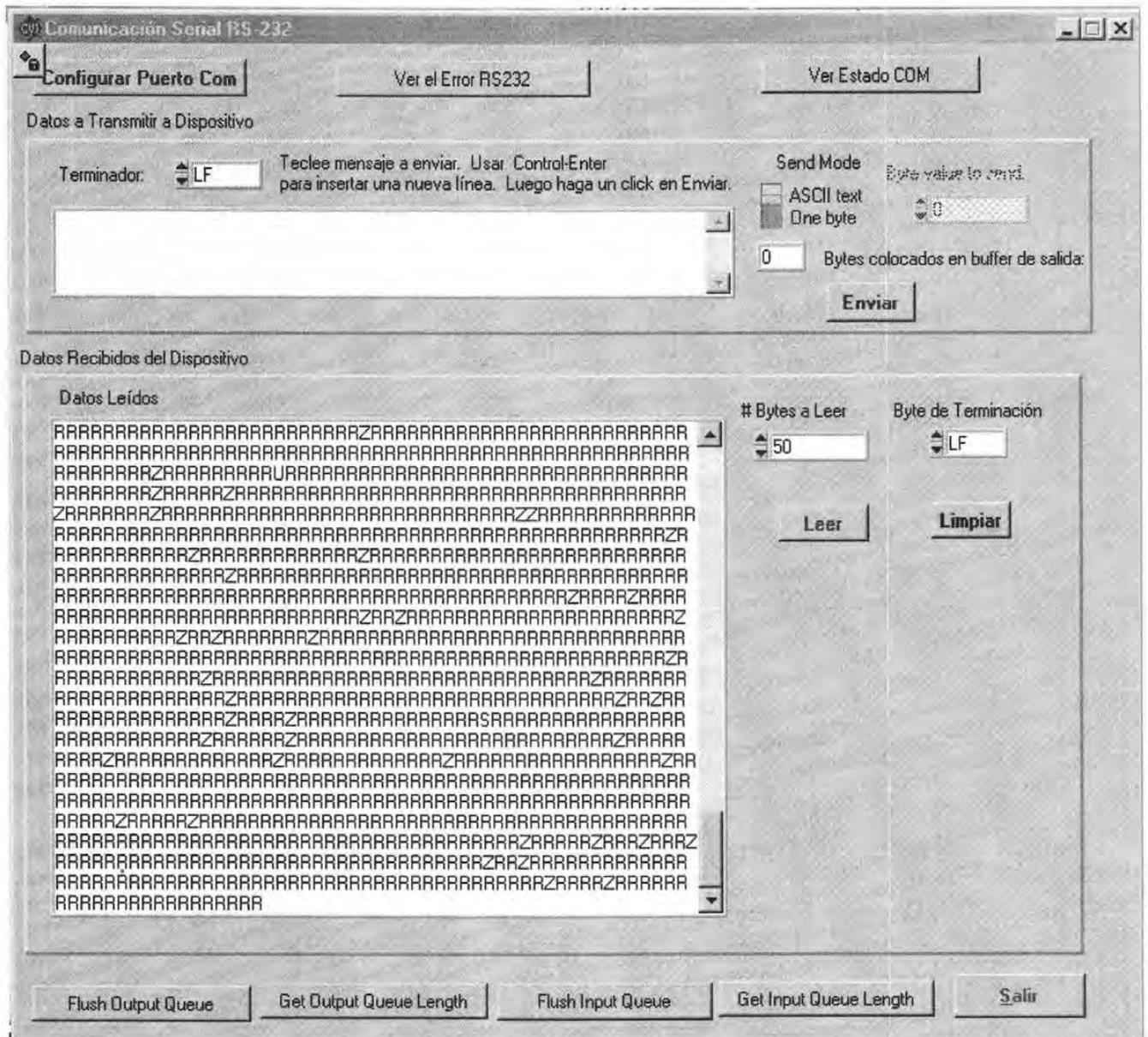


Fig. 3.2.1.3 Lectura de datos recibidos, pantalla 3

Después de seguir con las afinaciones tanto en el tiempo de retardo inicial como en la rutina de interrupción, al priorizar la interrupción que controla el inicio de la lectura de cada bit que conforma el byte de datos a leer, podemos ver en la Fig. 3.2.1.4, la correcta lectura del carácter U, que es justamente lo que se está transmitiendo en una cadena de bits al activarse la lectura en un flanco de bajada, es decir las rutinas de transmisión y recepción están afinadas.

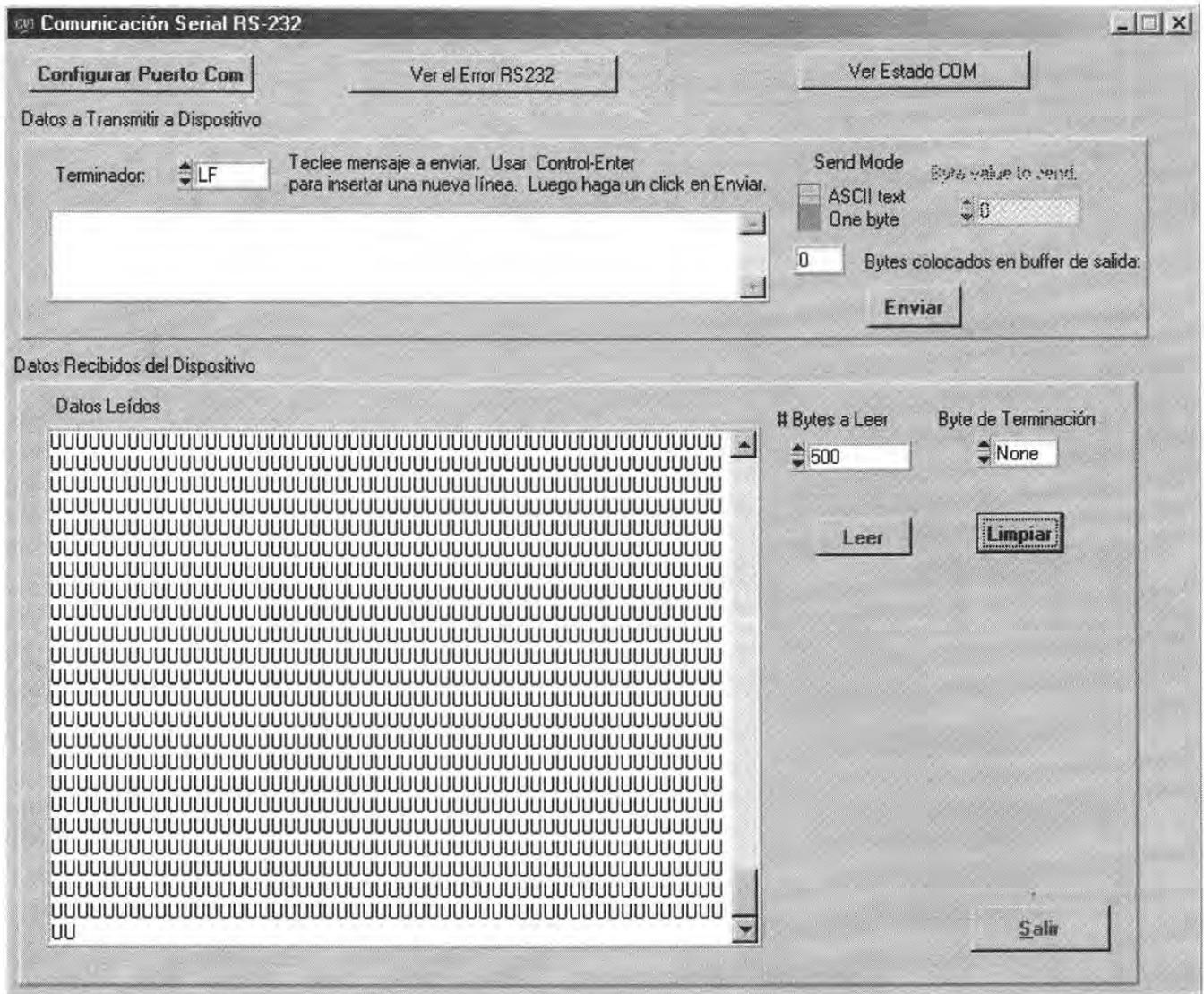


Fig. 3.2.1.4 Lectura de datos recibidos, pantalla 4

3.2.2 Pruebas de Transmisión y Recepción de datos asíncronos

La Fig. 3.2.2.1 muestra la señal del transmisor en el pin de entrada del módulo DR3000, con tiempos de bit para el 0, $t_0 = 52 \mu\text{s}$ y un tiempo de bit para el 1, $t_1 = 53.5 \mu\text{s}$. Este tren de pulsos viaja por el espacio y es recibido por otro módulo DR3000 y después de pasar por un buffer y un inversor se obtiene la onda mostrada en la parte inferior de esta misma figura, notamos un desfase de aproximadamente $23 \mu\text{s}$ ($t_d = 23\mu\text{s}$), característica que no importa para nuestro sistema de comunicación asíncrona. Podemos ver que se obtienen en el receptor un tiempo de bit $t_0 = 63 \mu\text{s}$ y un $t_1 = 45$

μs . Estos resultados nos indican que debemos afinar estos valores de tiempos de bits para poder reducir los errores de lectura en el receptor ya que en el receptor se espera una onda cuadrada parecida a la que se transmite.

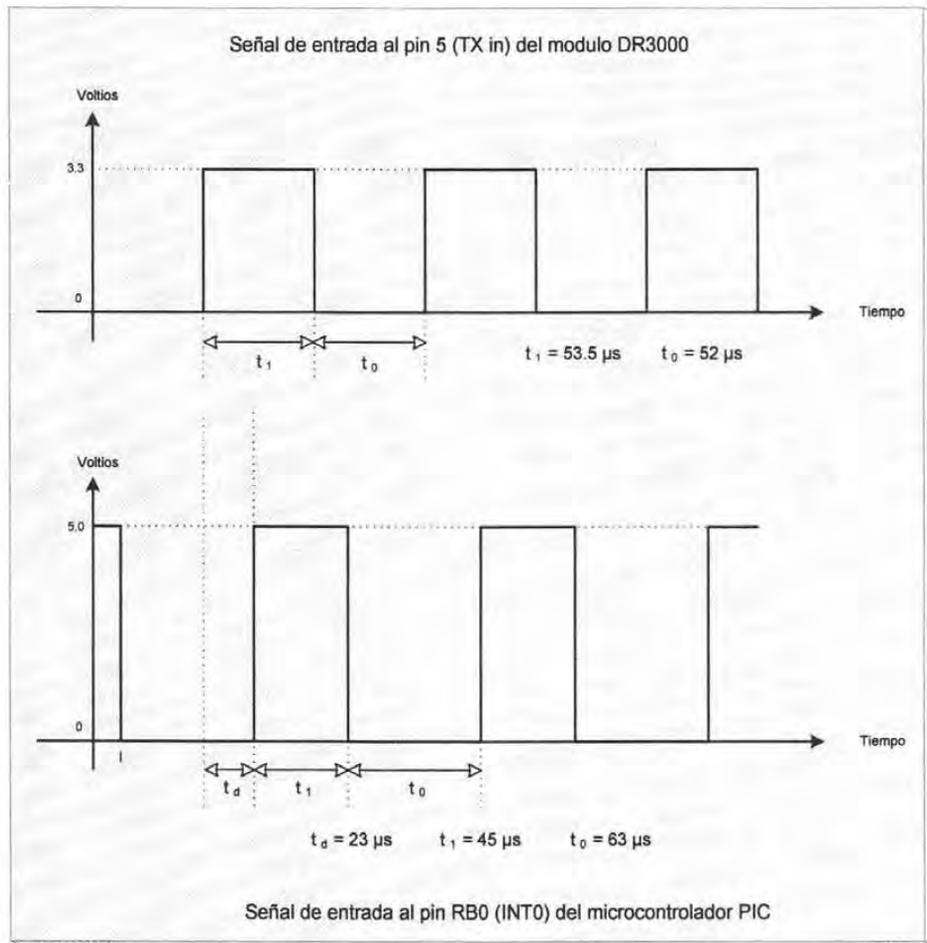


Fig. 3.2.2.1 Diagramas de tiempo de señal Transmitida y Señal recibida por los microcontroladores

En la Fig. 3.2.2.2, se observan las señales en el osciloscopio de la transmisión que corresponde a la señal superior y la recepción que corresponde a la señal recibida. Las escalas de la Fig. 3.2.2.2a tiene una escala vertical de 2 V/recuadro y en una escala horizontal de 20 μs /recuadro, mientras que en la Fig. 3.2.2.2b tiene una escala vertical de 2 V/recuadro y en una escala horizontal de 100 μs /recuadro.

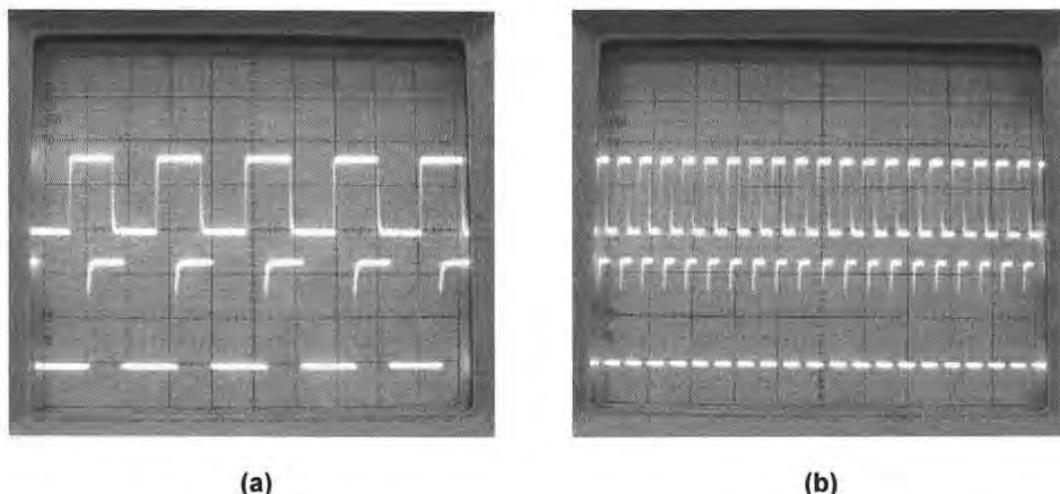


Fig. 3.2.2.2 Tren de pulsos en los microcontroladores de la: (a) la transmisión y (b) la recepción

Con estos valores de tiempos de bits descritos anteriormente se realizaron algunas pruebas que se muestran en la Tabla 3.2.2.1.

Tabla 3.2.2.1 Pruebas de Recepción con un bloque de datos de 7 bytes

Se Transmitió 7 valores:	Valores recibidos								
	Cabecera	Datos							CCR
0x2D	1010011	0x2E	0x96	0x97	0x97	0x97	0xC3	0xC3	0xDC
0x2D	1010011	0x2E	0x06	0x97	0x97	0x97	0x83	0xC3	0xDC
0x2D	1010011	0x2E	0x86	0x97	0x97	0x97	0x87	0xC3	0xDC
0x2E	1010011	0x2F	0x8C	0x9E	0x9E	0x9E	0xC6	0xF7	0xDA
0x2E	1010011	0x2F	0x86	0x9E	0x9E	0x9E	0xC6	0xF7	0xDA

Una primera estrategia de corregir las señales en el receptor es afinar los tiempos de bit en el transmisor y así intentar hacer parecer a la onda que idealmente se esperaba, aunque en el transmisor los tiempos de bits no estén muy parecidos.

Analizando la cadena de bits transmitida y la cadena de bits recibida podemos notar que existe un desplazamiento de los tiempos de lectura en el receptor con respecto a los tiempos de bits en la transmisión. Con los valores de la tabla anterior no podemos

decir si existe un adelanto o retraso en los tiempos de lectura en el receptor ya que según los datos existen las dos posibilidades.

Según nuestro punto de vista pudimos plantear una estrategia de afinar los valores de los tiempos de bits en el receptor. Iniciamos por redefinir el valor del primer retardo al inicio de la lectura de los datos que aproximadamente sería de medio tiempo de bit. Podemos observar en el osciloscopio que la señal del tren de pulsos del transmisor corresponde a una onda cuadrada de aproximadamente 52 μ s, tiempo que corresponde al ciclo de desbordamiento del temporizador TMR0 del microcontrolador fijado a un valor de 0x8A. Es decir, que para generar un retardo con este mismo temporizador debemos redefinir el valor del registro TMR0 que estaba definido en un valor de 0xD7, este valor fue modificado a un TMR0 = 0xC6. Con estos cambios se observó que se empezó a leer correctamente los primeros valores de datos del paquete transmitido, pero se seguía teniendo errores de recepción.

Tabla 3.2.2.2 Pruebas de Recepción al transmitir un bloque de datos

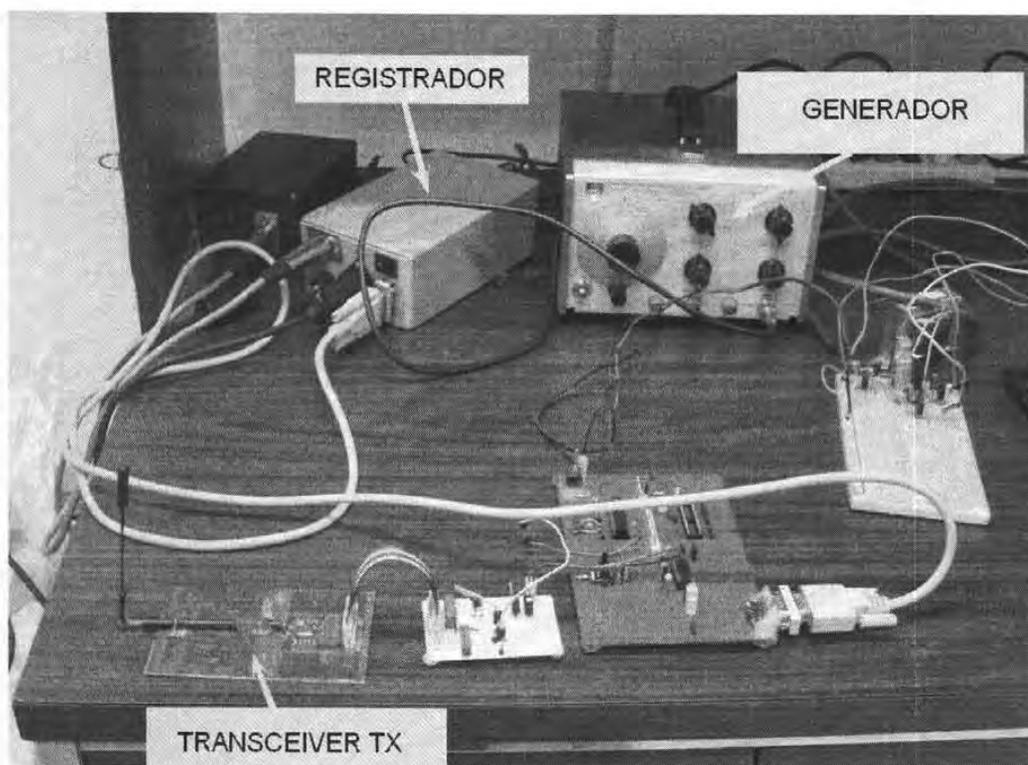
Se Transmitió:	Valores recibidos									
	Cabecera	Datos							CCR	
"ABCDEFGH"	1010011	0x41	0x42	0x43	0x44	0x45	0x46	0x47	0xDC	✓
"1234567"	1010011	0x31	0x32	0x33	0x34	0x35	0x36	0x37	0x6C	✓
"abcdefg"	1010011	0x61	0x62	0x63	0x64	0x65	0x66	0x67	0xBC	✓
"1A2B3C4"	1010011	0x31	0x41	0x32	0x42	0x33	0x43	0x34	0x0F	✗
"1A2B3C4"	1010011	0x31	0x41	0x32	0x42	0x33	0x43	0x34	0x33	✗
"1A2B3C4"	1010011	0x31	0x41	0x32	0x42	0x33	0x43	0x34	0x90	✓
"1A2B3C4"	1010011	0x31	0x41	0x32	0x42	0x33	0x43	0x34	0x90	✓
{0,0,0,0,0,0}	1010011	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	✓
{0,0,0,0,0,0}	1010011	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	✓
{1,1,1,1,1,1}	1010011	0x01	0x01	0x01	0x01	0x01	0x01	0x01	0x07	✓
{1,1,1,1,1,1}	1010011	0x01	0x01	0x01	0x01	0x01	0x01	0x01	0x07	✓
{0,1,2,3,4,5,6}	1010011	0x00	0x01	0x02	0x03	0x04	0x05	0x06	0x15	✓
{6,5,4,3,2,1,0}	1010011	0x01	0x01	0x01	0x01	0x01	0x01	0x01	0x15	✓
{0,2,4,8,16,32,64}	1010011	0x00	0x02	0x04	0x08	0x10	0x20	0x40	0x7E	✓
{64,32,16,8,4,2,0}	1010011	0x40	0x20	0x10	0x08	0x04	0x02	0x00	0x7E	✓
{3,16,25,61,77,129,205}	1010011	0x03	0x10	0x19	0x3D	0x4D	0x81	0xCD	0x04	✓
{205,129,77,61,25,16,3}	1010011	0xCD	0x81	0x4D	0x3D	0x19	0x10	0x03	0x04	✓
"1A2B3C4"	1010011	0x31	0x41	0x32	0x42	0x33	0x43	0x34	0x90	✓
"1A2B3C4"	1010011	0x31	0x41	0x32	0x42	0x33	0x43	0x34	0x90	✓

Analizando los bits de la cadena recibida, bit a bit, notamos que tenían un cierto retraso en los tiempos de bits de lectura, entonces se modificó el valor del período de lectura de cada bit en el receptor. Este periodo de lectura nuevamente es controlado por el valor del registro TMR0, se modificó su valor que tenía de 0x8B, por otro valor, es decir se fijó TMR0 = 0x8D. Esta modificación afinó nuestros resultados, lográndose una recepción bastante buena, tal como se muestran los resultados obtenidos de diferentes valores a ser transmitidos. En la primera columna de la Tabla 3.2.2.2, se describe los valores transmitido que se componen de cadenas encerrados entres dobles comillas y valores decimales encerrados en corchetes. Se muestra la cabecera recibida que como vemos no sufre ninguna modificación, luego se describen cada uno de los valores en formato hexadecimal que lee el receptor, podemos notar en la tabla que se tuvieron dos errores generados por leer un diferente CCR transmitido.

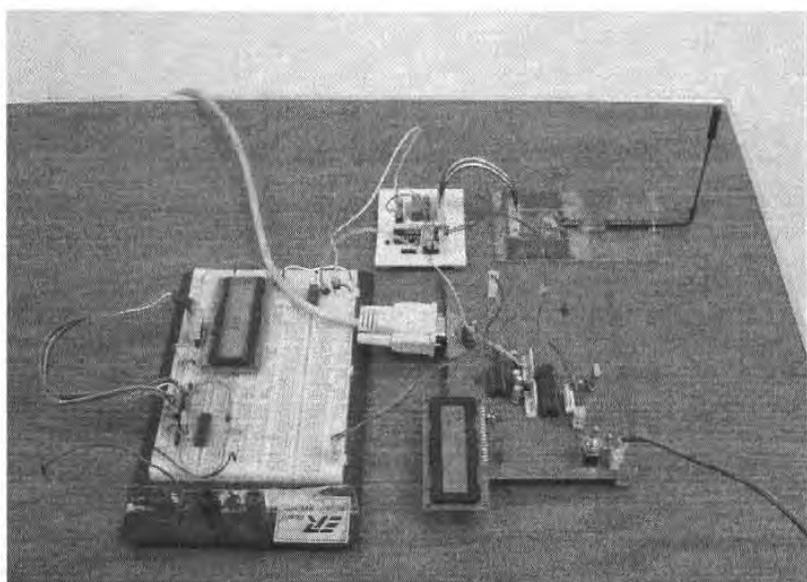
3.3 Pruebas de Transmisión y Recepción con un Registrador Sismológico

3.3.1 Prueba 1

Las siguientes pruebas se realizaron con el registrador desarrollado en el Instituto de Ingeniería como tesis de maestría ^[15], que se compone de un digitalizador y un software de adquisición de eventos que reside en una computadora. Como señal de entrada al digitalizador se conectó un generador de onda. En las fotos de la Fig. 3.3.1.1, podemos observar el registrador, el generador de señales y el módulo transmisor digital en RF interconectados para así poder transmitir la señal digitalizada del generador de señal. El generador de señales proporciona señales de baja frecuencia, característica de la señal sísmica con amplitudes bipolares de +5 Voltios. En esta prueba se configuró el tamaño del paquete de datos a 2 bytes y la velocidad del registrador y de enlace con la PC a 19200 bps. El digitalizador se configuró que realice un muestreo de 200 muestras por segundo (mps). Las pruebas realizadas anteriormente corresponde a una distancia de separación de ambos módulos de aproximadamente 1 m.



(a)



(b)

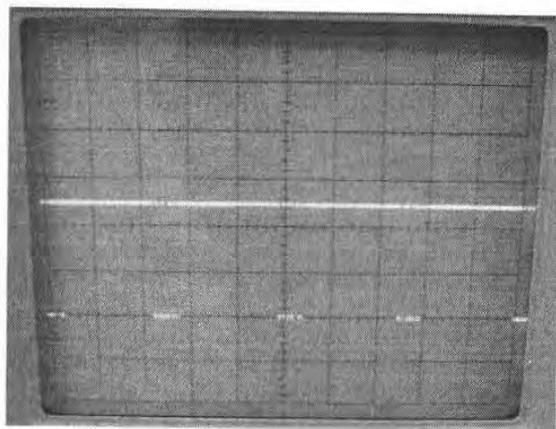
Fig. 3.3.1.1 Pruebas con un registrador sísmico: (a) Módulo Transmisor y (b) Módulo Receptor

En la Fig. 3.3.1.2. podemos observar más al detalle la interconexión del receptor con el software de adquisición y registro de la señal recibida por su puerto serial RS232. En este caso el software se configuró para visualización y registro de la señal de manera continua, que nos puede servir en un futuro para su análisis de los valores recibidos.

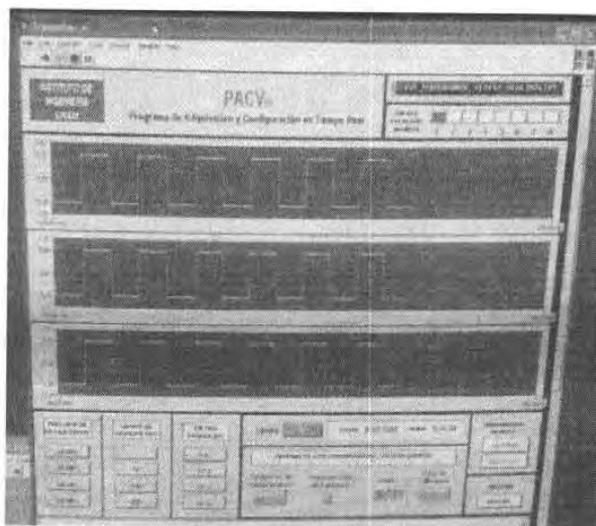
En la Fig. 3.3.1.3, podemos ver en el osciloscopio la señal del pin RB6, de salida del puerto UART del microcontrolador PIC del receptor, que es la señal a transmitirse a la PC donde reside el software de adquisición y registro; esta señal tiene un período de aproximadamente $5 \mu\text{s}$ que corresponde al muestreo de 200 mps, ya que el digitalizador apenas termina de realizar la digitalización transmite sus 2 bytes de la señal digitalizada. Igualmente estos 2 bytes fueron configurados en el sistema transmisor-receptor como tamaño de bloque del paquete de datos, es por eso que al recibirlos inmediatamente lo transmite a la PC o dispositivo serial destino.



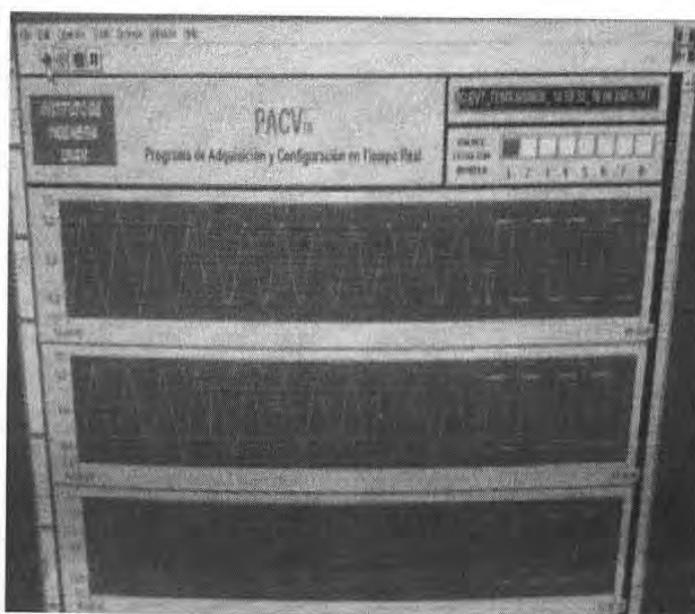
Fig. 3.3.1.2 Módulo de Recepción



(a)



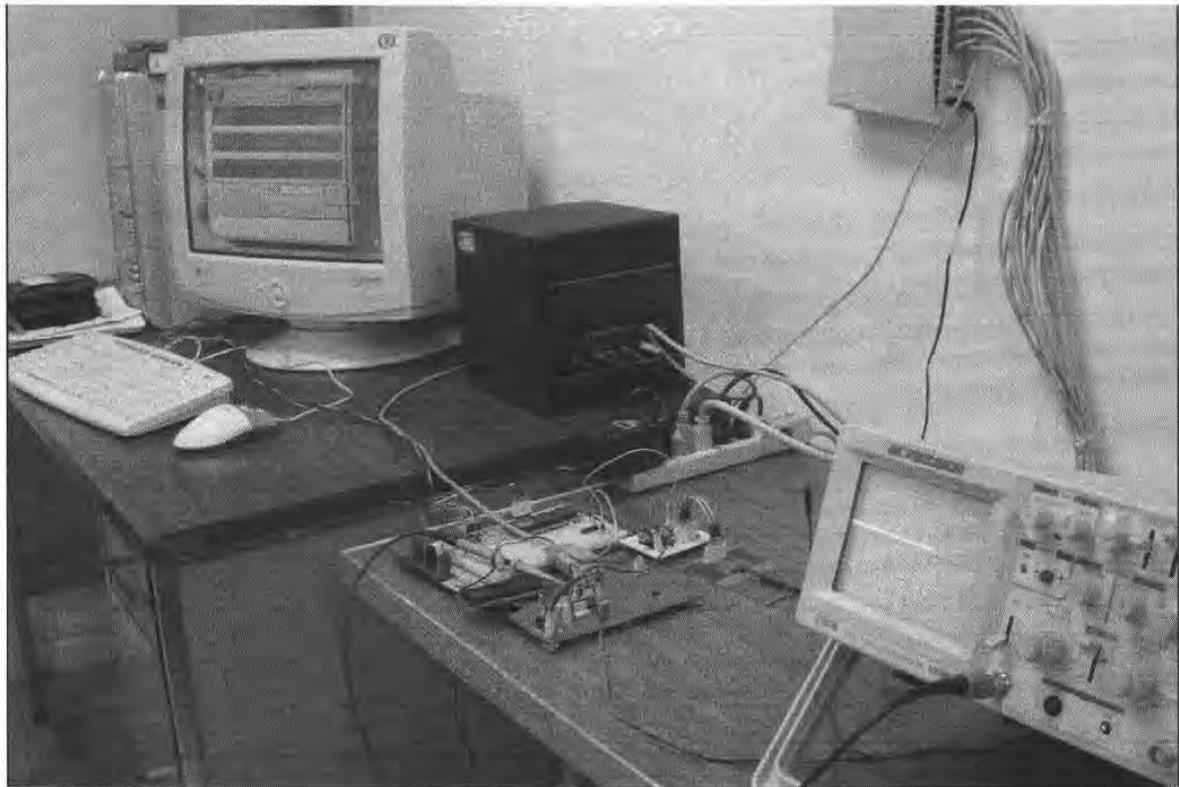
(b)



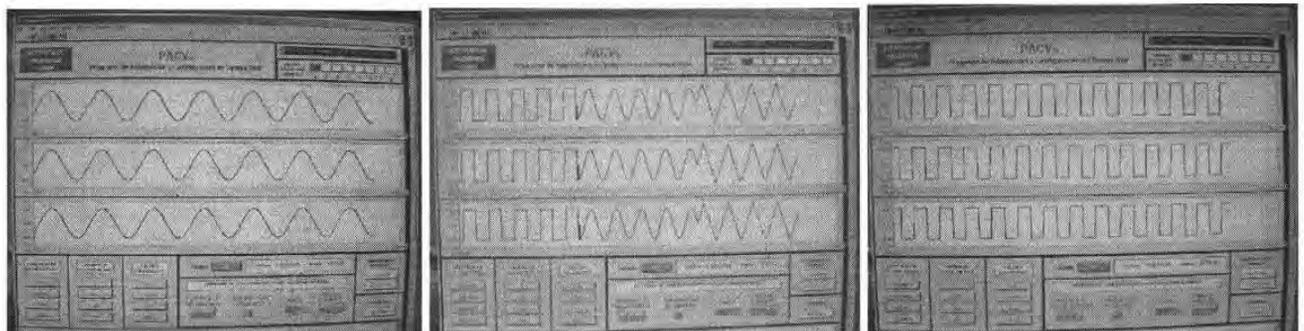
(c)

Fig. 3.3.1.3 Señales Recibidas, (a) Señal del pin RB6, (b) y (c) señal monitoreada

3.3.2 Prueba 2



(a)



(b)

(c)

(d)

Fig. 3.3.2.1 Módulo de Recepción (a) Receptor, (b) Señal del pin RB6, (c) y (d) señal monitoreada

Luego de verificar la correcta operación del transmisor y receptor se optó por hacer unas pruebas con una distancia de separación mayor que la anterior, en este caso se separaron el transmisor y receptor una distancia de aproximadamente 10 m, que es donde se encuentra nuestro servidor del

laboratorio, computadora que se le instaló el software de monitoreo y adquisición igual que en la PC de la prueba anterior. Para poder visualizar las ondas que proporciona el sistema proveniente del generador, se pueden observar en las ondas recibidas en la Fig. 3.3.2.1. En las ondas inferiores podemos ver el tipo de señal recibida y podemos verificar cualitativamente que no existe error de transmisión ni de recepción. Se modificó el fondo del programa para poder ver mejor la señal recibida. Este sistema se dejó funcionando por más de 12 horas y no presentó ningún error en su funcionamiento global.

3.3.3 Prueba 3

Fijando los valores en los registros:

En ISR de INT0, con TMR0 = D5

En ISR del TMR0, con TMR0 = 8A

Obtenemos los diagramas de tiempos mostrados en la Fig. 3.3.3.1. En el pin RB0 leemos una onda cuadrada con los tiempos $t_1 = 56 \mu\text{s}$, $t_0 = 52 \mu\text{s}$. Monitoreando los tiempos de lectura de bit en el pin RB0, obtenemos un $t_{\text{bit}/2} = 24 \mu\text{s}$, $t_{\text{lectura}} = 54 \mu\text{s}$.

Tomando como referencia la condición de balance en AC y DC del DR3000, e implementada en la codificación del módulo transmisor podemos tener una aproximado para los tiempos de lectura de bit como un promedio de t_1 y t_0 . Obteniendo los siguientes cálculos:

$$t_{\text{lectura}} = (t_1 + t_0)/2 = (56 \mu\text{s} + 52 \mu\text{s})/2 = 54 \mu\text{s}$$

$$t_{\text{bit}/2} = t_{\text{lectura}}/2 = 27 \mu\text{s}$$

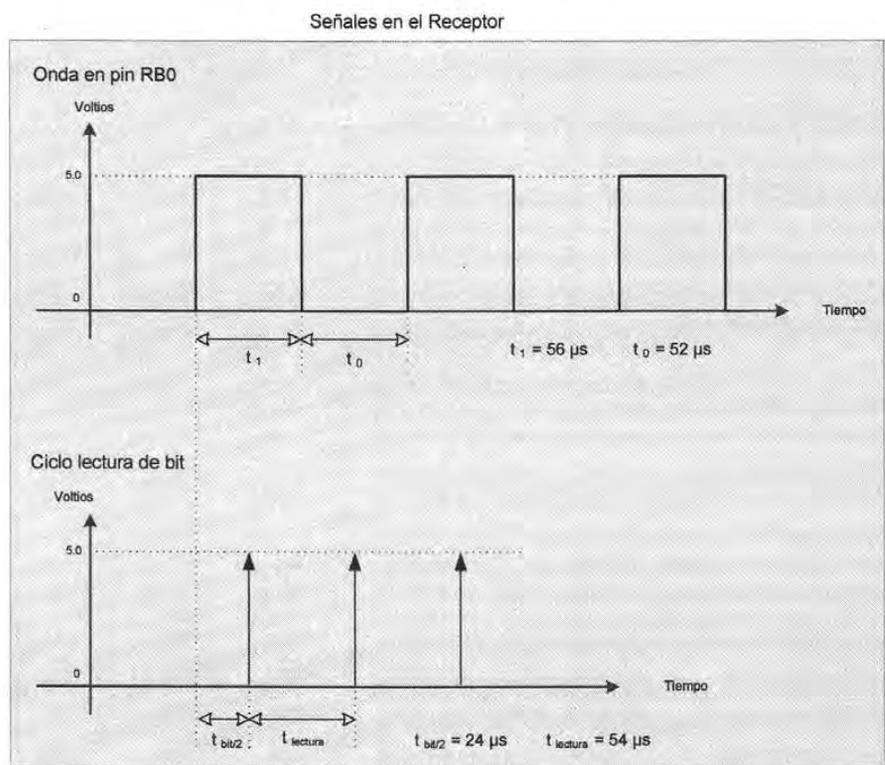


Fig. 3.3.3.1 Señales en el Módulo Receptor Digital en RF

Es decir el único valor a modificar sería el tiempo $t_{\text{bit}/2}$, fijando el valor del registro $\text{TMR0}=0\text{xCE}$, dentro del ISR de INT0.

Después de visualizar 5 disparos de lectura de bit consecutivos se observa un desfase de aproximadamente $4 \mu\text{s}$. Entonces, se disminuirá el valor del registro TMR0 de su valor actual dentro de la ISR del TMR0 , fijamos $\text{TMR0}=0\text{x8C}$.

Conclusión:

Dentro del ISR de INT0 se fijó el valor de $\text{TMR0}=0\text{xCE}$

Dentro del ISR del TMR0 se fijó el valor de $\text{TMR0}=0\text{x8C}$

Las pruebas se realizaron con el registrador [5], usando como señal a digitalizar una onda senoidal, obteniéndose la siguiente tabla de datos enviados y recibidos por el registrador y recibidos en una PC, usando los módulos Transmisor y

Receptor Digital en RF, se obtuvo en la Tabla 3.3.3.1, vemos que se reduce grandemente los errores obtenidos en pruebas anteriores.

Tabla 3.3.3.1 Valores Transmitidos y Recibidos por el registrador, usando los transceivers, Prueba 3

DATOS TRANSMITIDOS Y RECIBIDOS UTILIZANDO MODULOS TRANSMISOR Y RECEPTOR															
N	D Tx	D Rx	Dato	N	D Tx	D Rx	Dato	N	D Tx	D Rx	Dato	N	D Tx	D Rx	Dato
1	857	857	✓	237	1370	1370	✓	473	601	601	✓	709	14936	14936	✓
2	15706	15706	✓	238	9567	9567	✓	474	8795	8795	✓	710	10840	10840	✓
3	7512	7512	✓	239	13659	13659	✓	475	12892	12892	✓	711	12891	12891	✓
4	3417	3417	✓	240	11614	11614	✓	476	2651	2651	✓	712	4700	4700	✓
5	5469	5469	✓	241	7514	7514	✓	477	6746	6746	✓	713	601	601	✓
6	1373	1373	✓	242	15711	15711	✓	478	1626	1626	✓	714	15451	15451	✓
7	6493	6493	✓	243	861	861	✓	479	5726	5726	✓	715	15452	15452	✓
8	2397	2397	✓	244	9054	9054	✓	480	3675	3675	✓	716	7258	7258	✓
9	4446	4446	✓	245	9051	9051	✓	481	7775	7775	✓	717	7260	7260	✓
10	15967	15967	✓	246	9049	9049	✓	482	8541	8541	✓	718	7257	7257	✓
11	11867	11867	✓	247	9051	9051	✓	483	12638	12638	✓	719	15448	15448	✓
12	13917	13917	✓	248	9054	9054	✓	484	10591	10591	✓	720	15449	15449	✓
13	9822	9822	✓	249	858	858	✓	485	14681	14681	✓	721	604	604	✓
14	14936	14936	✓	250	15710	15710	✓	486	9565	9565	✓	722	8798	8798	✓
15	10841	10841	✓	251	11613	11613	✓	487	13662	13662	✓	723	12888	12888	✓
16	12895	12895	✓	252	3417	3417	✓	488	11612	11612	✓	724	2654	2654	✓
17	4702	4702	✓	253	5467	5467	✓	489	7517	7517	✓	725	6750	6750	✓
18	8794	8794	✓	254	9560	9560	✓	490	15707	15707	✓	726	1629	1629	✓
19	600	600	✓	255	14684	14684	✓	491	863	863	✓	727	5725	5725	✓
20	15452	15452	✓	256	2393	2393	✓	492	9049	9049	✓	728	3677	3677	✓
21	7258	7258	✓	257	4441	4441	✓	493	9051	9051	✓	729	7769	7769	✓
22	7256	7256	✓	258	350	350	✓	494	9055	9055	✓	730	347	347	✓
23	7262	7262	✓	259	11871	11871	✓	495	9053	9053	✓	731	4447	4447	✓
24	7261	7261	✓	260	3676	3676	✓	496	9052	9052	✓	732	10588	10588	✓
25	15454	15454	✓	261	9819	9819	✓	497	862	862	✓	733	14682	14682	✓
26	15451	15451	✓	262	14939	14939	✓	498	15709	15709	✓	734	9564	9564	✓
27	8794	8794	✓	263	6744	6744	✓	499	7516	7516	✓	735	5471	5471	✓
28	4699	4699	✓	264	2648	2648	✓	500	3421	3421	✓	736	3423	3423	✓
29	2650	2650	✓	265	4702	4702	✓	501	13660	13660	✓	737	7516	7516	✓
30	6746	6746	✓	266	8792	8792	✓	502	9564	9564	✓	738	15709	15709	✓
31	1624	1624	✓	267	15453	15453	✓	503	14682	14682	✓	739	861	861	✓
32	5720	5720	✓	268	7263	7263	✓	504	10584	10584	✓	740	9054	9054	✓
33	3676	3676	✓	269	7258	7258	✓	505	4447	12639	*	741	9049	9049	✓
34	7772	7772	✓	270	7260	7260	✓	506	346	346	✓	742	9049	9049	✓
35	347	347	✓	271	7262	7262	✓	507	7772	7772	✓	743	9049	9049	✓
36	4443	4443	✓	272	7258	7258	✓	508	13919	13919	✓	744	9049	9049	✓
37	10584	10584	✓	273	7263	7263	✓	509	5720	5720	✓	745	861	861	✓
38	14682	14682	✓	274	602	602	✓	510	14941	14941	✓	746	15706	15706	✓
39	9560	9560	✓	275	8796	8796	✓	511	6744	6744	✓	747	7516	7516	✓
40	5471	5471	✓	276	4697	4697	✓	512	12891	12891	✓	748	3419	3419	✓
41	3421	3421	✓	277	12891	12891	✓	513	4697	4697	✓	749	5467	5467	✓
42	11615	11615	✓	278	10847	10847	✓	514	607	607	✓	750	9566	9566	✓
43	15708	15708	✓	279	14939	14939	✓	515	600	600	✓	751	14686	14686	✓
44	858	858	✓	280	9817	9817	✓	516	15450	15450	✓	752	10586	10586	✓
45	9048	9048	✓	281	13915	13915	✓	517	7261	7261	✓	753	12632	12632	✓
46	9049	9049	✓	282	7770	7770	✓	518	7262	7262	✓	754	8540	8540	✓
47	9053	9053	✓	283	345	345	✓	519	7257	7257	✓	755	7771	7771	✓
48	9049	9049	✓	284	4442	4442	✓	520	7262	7262	✓	756	3677	3677	✓
49	9050	9050	✓	285	2398	2398	✓	521	15448	15448	✓	757	9823	9823	✓
50	863	863	✓	286	6495	6495	✓	522	15455	15455	✓	758	1624	1624	✓
51	856	856	✓	287	1375	1375	✓	523	607	607	✓	759	6748	6748	✓
52	7517	7517	✓	288	5469	5469	✓	524	4697	4697	✓	760	2650	2650	✓
53	3423	3423	✓	289	3418	3418	✓	525	12893	12893	✓	761	4703	4703	✓
54	13661	13661	✓	290	11615	11615	✓	526	10844	10844	✓	762	8798	8798	✓
55	9563	9563	✓	291	15708	15708	✓	527	14942	14942	✓	763	604	604	✓
56	14683	14683	✓	292	862	862	✓	528	9822	9822	✓	764	15452	15452	✓
57	6492	6492	✓	293	9052	9052	✓	529	13913	13913	✓	765	15448	15448	✓
58	12637	12637	✓	294	9053	9053	✓	530	11869	11869	✓	766	7258	7258	✓
59	8537	8537	✓	295	4952	4952	✓	531	346	346	✓	767	7258	7258	✓
60	15966	15966	✓	296	9049	9049	✓	532	4446	4446	✓	768	7257	7257	✓
61	11868	11868	✓	297	9048	9048	✓	533	2398	2398	✓	769	7263	7263	✓
62	5721	5721	✓	298	861	861	✓	534	6494	6494	✓	770	15453	15453	✓

Capítulo 3 : Pruebas y Resultados del Transceiver Digital en RF

63	1631	1631	✓	299	15709	15709	✓	535	1369	1369	✓	771	607	607	✓
64	6751	6751	✓	300	7518	7518	✓	536	5468	5468	✓	772	4702	4702	✓
65	2655	2655	✓	301	11608	11608	✓	537	3422	3422	✓	773	12889	12889	✓
66	12888	12888	✓	302	3416	3416	✓	538	11613	11613	✓	774	10842	10842	✓
67	8795	8795	✓	303	5464	5464	✓	539	15708	15708	✓	775	14938	14938	✓
68	606	606	✓	304	1368	1368	✓	540	856	856	✓	776	1627	1627	✓
69	15453	15453	✓	305	6488	6488	✓	541	9048	9048	✓	777	13918	13918	✓
70	7263	7263	✓	306	12637	12637	✓	542	9050	9050	✓	778	11866	11866	✓
71	7260	7260	✓	307	8543	8543	✓	543	4952	4952	✓	779	15963	15963	✓
72	7258	7258	✓	308	15963	15963	✓	544	4952	4952	✓	780	8541	8541	✓
73	7262	7262	✓	309	11868	11868	✓	545	9050	9050	✓	781	12639	12639	✓
74	7263	7263	✓	310	13912	13912	✓	546	861	861	✓	782	6488	6488	✓
75	15453	15453	✓	311	9816	9816	✓	547	856	856	✓	783	1374	1374	✓
76	601	601	✓	312	6751	6751	✓	548	15704	15704	✓	784	9567	9567	✓
77	8799	8799	✓	313	10844	10844	✓	549	11614	11614	✓	785	3418	3418	✓
78	4703	4703	✓	314	12892	12892	✓	550	3420	3420	✓	786	11609	11609	✓
79	10844	10844	✓	315	8793	8793	✓	551	5466	5466	✓	787	7515	7515	✓
80	6747	6747	✓	316	603	603	✓	552	1374	1374	✓	788	15711	15711	✓
81	1626	1626	✓	317	15453	15453	✓	553	6489	6489	✓	789	9048	9048	✓
82	5725	5725	✓	318	15452	15452	✓	554	12639	12639	✓	790	9048	9048	✓
83	11866	11866	✓	319	7259	7259	✓	555	4440	4440	✓	791	9053	9053	✓
84	15960	15960	✓	320	7258	7258	✓	556	15963	15963	✓	792	9055	9055	✓
85	8536	8536	✓	321	7257	7257	✓	557	11869	11869	✓	793	9053	9053	✓
86	12638	12638	✓	322	7261	7261	✓	558	13913	13913	✓	794	859	859	✓
87	10587	10587	✓	323	15449	15449	✓	559	9822	9822	✓	795	15711	15711	✓
88	1368	1368	✓	324	601	601	✓	560	6751	6751	✓	796	7515	7515	✓
89	9567	9567	✓	325	8797	8797	✓	561	10842	10842	✓	797	11615	11615	✓
90	13661	13661	✓	326	12890	12890	✓	562	12890	12890	✓	798	3420	3420	✓
91	3419	3419	✓	327	2649	2649	✓	563	8799	8799	✓	799	5465	5465	✓
92	7516	7516	✓	328	6745	6745	✓	564	605	605	✓	800	1373	1373	✓
93	15710	15710	✓	329	1630	1630	✓	565	600	600	✓	801	6489	6489	✓
94	861	861	✓	330	5725	5725	✓	566	15450	15450	✓	802	2398	2398	✓
95	9053	9053	✓	331	3675	3675	✓	567	7263	7263	✓	803	4442	4442	✓
96	9055	9055	✓	332	15960	15960	✓	568	7257	7257	✓	804	15967	15967	✓
97	4952	4952	✓	333	351	351	✓	569	7257	7257	✓	805	7772	7772	✓
98	9051	9051	✓	334	12632	12632	✓	570	7259	7259	✓	806	13913	13913	✓
99	9054	9054	✓	335	10589	10589	✓	571	15448	15448	✓	807	9821	9821	✓
100	857	857	✓	336	14687	14719	*	572	600	600	✓	808	14937	14937	✓
101	15705	15705	✓	337	9562	9562	✓	573	8793	8793	✓	809	10844	10844	✓
102	7516	7516	✓	338	13658	13658	✓	574	4699	4699	✓	810	12894	12894	✓
103	3419	3419	✓	339	11612	11612	✓	575	2649	2649	✓	811	8799	8799	✓
104	13656	13656	✓	340	7517	7517	✓	576	6745	6745	✓	812	8792	8792	✓
105	9566	9566	✓	341	15711	15711	✓	577	1630	1630	✓	813	600	600	✓
106	14686	14686	✓	342	861	861	✓	578	5726	5726	✓	814	7263	7263	✓
107	10584	10584	✓	343	9050	9050	✓	579	3678	3678	✓	815	7261	7261	✓
108	4443	4443	✓	344	9049	9049	✓	580	7769	7769	✓	816	7258	7258	✓
109	350	350	✓	345	9055	9055	✓	581	351	351	✓	817	7262	7262	✓
110	7772	7772	✓	346	9051	9051	✓	582	12632	12632	✓	818	7261	7261	✓
111	3672	3672	✓	347	9048	9048	✓	583	10588	10588	✓	819	15452	15452	✓
112	5720	5720	✓	348	859	859	✓	584	14680	14680	✓	820	602	602	✓
113	14943	14943	✓	349	15709	15709	✓	585	9564	9564	✓	821	8798	8798	✓
114	6744	6744	✓	350	7516	7516	✓	586	13658	13658	✓	822	4703	4703	✓
115	2649	2649	✓	351	3419	3419	✓	587	3421	3421	✓	823	2648	2648	✓
116	4697	4697	✓	352	13660	13660	✓	588	7518	7518	✓	824	10845	10845	✓
117	8793	8793	✓	353	9564	9564	✓	589	15707	15707	✓	825	1624	1624	✓
118	600	600	✓	354	14686	14686	✓	590	861	861	✓	826	9817	9817	✓
119	15452	15452	✓	355	2399	2399	✓	591	9050	9050	✓	827	3674	3674	✓
120	7257	7257	✓	356	4445	4445	✓	592	9054	9054	✓	828	7772	7772	✓
121	7262	7262	✓	357	8536	8536	✓	593	9051	9051	✓	829	345	345	✓
122	7262	7262	✓	358	7770	7770	✓	594	9055	9055	✓	830	4443	4443	✓
123	7259	7259	✓	359	3676	3676	✓	595	9050	9050	✓	831	10588	10588	✓
124	15448	15448	✓	360	5724	5724	✓	596	861	861	✓	832	14682	14682	✓
125	600	600	✓	361	1628	1628	✓	597	15711	15711	✓	833	9564	9564	✓
126	8794	8794	✓	362	6746	6746	✓	598	7513	7513	✓	834	5471	5471	✓
127	4697	4697	✓	363	2649	2649	✓	599	11614	11614	✓	835	3417	3417	✓
128	12891	12891	✓	364	12888	12888	✓	600	13662	13662	✓	836	7512	7512	✓
129	10841	10841	✓	365	8797	8797	✓	601	9563	9563	✓	837	15706	15706	✓
130	14942	14942	✓	366	600	600	✓	602	14682	14682	✓	838	862	862	✓
131	9821	9821	✓	367	15448	15448	✓	603	10590	10590	✓	839	9048	9048	✓
132	13915	13915	✓	368	7259	7259	✓	604	12634	12634	✓	840	9049	9049	✓
133	11869	11869	✓	369	7257	7257	✓	605	8542	8542	✓	841	4956	4956	✓
134	344	344	✓	370	7260	7260	✓	606	15964	15964	✓	842	9053	9053	✓
135	4440	4440	✓	371	7261	7261	✓	607	3675	3675	✓	843	9049	9049	✓
136	2397	2397	✓	372	15452	15452	✓	608	5722	5722	✓	844	859	859	✓
137	6489	6489	✓	373	600	600	✓	609	1626	1626	✓	845	15711	15711	✓
138	1373	1373	✓	374	603	603	✓	610	6745	6745	✓	846	7513	7513	✓

139	5470	5470	✓	375	4698	4698	✓	611	2649	2649	✓	847	11610	11610	✓
140	3420	3420	✓	376	12891	12891	✓	612	12892	12892	✓	848	13657	13657	✓
141	11615	11615	✓	377	10846	10846	✓	613	8793	8793	✓	849	9563	9563	✓
142	7519	7519	✓	378	14940	14940	✓	614	606	606	✓	850	14687	14687	✓
143	862	862	✓	379	9816	9816	✓	615	15454	15454	✓	851	6488	6488	✓
144	9048	9048	✓	380	13914	13914	✓	616	7259	7259	✓	852	12635	12635	✓
145	9054	9054	✓	381	11865	11865	✓	617	7262	7262	✓	853	8538	8538	✓
146	9053	9053	✓	382	15966	15966	✓	618	7256	7256	✓	854	15962	15962	✓
147	9051	9051	✓	383	8537	8537	✓	619	7259	7259	✓	855	11868	11868	✓
148	9052	9052	✓	384	2396	2396	✓	620	15452	15452	✓	856	5725	5725	✓
149	861	861	✓	385	6492	6492	✓	621	15453	15453	✓	857	1627	1627	✓
150	858	858	✓	386	1372	1372	✓	622	603	603	✓	858	6749	6749	✓
151	15708	15708	✓	387	9567	9567	✓	623	4700	4700	✓	859	10840	10840	✓
152	11613	11613	✓	388	3420	3420	✓	624	12893	12893	✓	860	12889	12889	✓
153	3418	3418	✓	389	11610	11610	✓	625	2655	2655	✓	861	8799	8799	✓
154	5466	5466	✓	390	7513	7513	✓	626	6745	6745	✓	862	605	605	✓
155	1369	1369	✓	391	860	860	✓	627	1629	1629	✓	863	15452	15452	✓
156	6489	6489	✓	392	859	859	✓	628	5723	5723	✓	864	7261	7261	✓
157	2396	2396	✓	393	9054	9054	✓	629	11866	11866	✓	865	7258	7258	✓
158	4440	4440	✓	394	9055	9055	✓	630	15966	15966	✓	866	7260	7260	✓
159	15967	15967	✓	395	9051	9051	✓	631	8543	8543	✓	867	7257	7257	✓
160	11867	11867	✓	396	9051	9051	✓	632	2392	2392	✓	868	7259	7259	✓
161	13916	13916	✓	397	9048	9048	✓	633	6492	6492	✓	869	15450	15450	✓
162	1631	1631	✓	398	15711	15711	✓	634	14685	14685	✓	870	604	604	✓
163	14938	14938	✓	399	7515	7515	✓	635	9563	9563	✓	871	8799	8799	✓
164	10841	10841	✓	400	11615	11615	✓	636	13658	13658	✓	872	12894	12894	✓
165	12891	12891	✓	401	3420	3420	✓	637	3423	3423	✓	873	2655	2655	✓
166	4700	4700	✓	402	5466	5466	✓	638	7518	7518	✓	874	6751	6751	✓
167	8792	8792	✓	403	1369	1369	✓	639	15705	15705	✓	875	1629	1629	✓
168	15453	15453	✓	404	6494	6494	✓	640	862	862	✓	876	5727	5727	✓
169	15452	15452	✓	405	2398	2398	✓	641	9049	25433	*	877	11868	11868	✓
170	7261	7261	✓	406	4444	4444	✓	642	9051	9051	✓	878	15962	15962	✓
171	7258	7258	✓	407	15965	15965	✓	643	9053	9053	✓	879	351	351	✓
172	7258	7258	✓	408	11871	11871	✓	644	9051	9051	✓	880	12634	12634	✓
173	7257	7257	✓	409	13917	13917	✓	645	9052	9052	✓	881	10587	10587	✓
174	15453	15453	✓	410	9821	9821	✓	646	862	862	✓	882	14685	14685	✓
175	602	602	✓	411	14938	14938	✓	647	7519	7519	✓	883	9565	9565	✓
176	8797	8797	✓	412	10841	10841	✓	648	11611	11611	✓	884	13657	13657	✓
177	12892	12892	✓	413	12895	12895	✓	649	3422	3422	✓	885	11612	11612	✓
178	2654	2654	✓	414	4702	4702	✓	650	5465	5465	✓	886	7513	7513	✓
179	6746	6746	✓	415	8796	8796	✓	651	1371	1371	✓	887	15711	15711	✓
180	1630	1630	✓	416	15453	15453	✓	652	14684	14684	✓	888	863	863	✓
181	5722	5722	✓	417	15448	15448	✓	653	2395	2395	✓	889	9050	9050	✓
182	3678	3678	✓	418	7262	7262	✓	654	4446	4446	✓	890	9054	9054	✓
183	7774	7774	✓	419	7258	7258	✓	655	345	345	✓	891	9055	9055	✓
184	351	351	✓	420	7258	7258	✓	656	7770	7770	✓	892	9053	9053	✓
185	12636	12636	✓	421	7263	7263	✓	657	13919	13919	✓	893	9048	9048	✓
186	10588	10588	✓	422	15454	15454	✓	658	9819	9819	✓	894	858	858	✓
187	14681	14681	✓	423	602	602	✓	659	14941	14941	✓	895	15706	15706	✓
188	9566	9566	✓	424	8794	8794	✓	660	10847	10847	✓	896	11615	11615	✓
189	5471	5471	✓	425	4699	4699	✓	661	2648	2648	✓	897	3421	3421	✓
190	3421	3421	✓	426	2650	2650	✓	662	4702	4702	✓	898	5471	5471	✓
191	7516	7516	✓	427	10847	10847	✓	663	8796	8796	✓	899	9562	9562	✓
192	15705	15705	✓	428	14941	14941	✓	664	15455	15455	✓	900	14686	14686	✓
193	861	861	✓	429	9823	9823	✓	665	7259	7259	✓	901	2399	2399	✓
194	859	859	✓	430	13919	13919	✓	666	7261	7261	✓	902	4443	4443	✓
195	9055	9055	✓	431	7770	7770	✓	667	7260	7260	✓	903	349	349	✓
196	9055	9055	✓	432	346	346	✓	668	7257	7257	✓	904	7769	7769	✓
197	9055	9055	✓	433	4443	4443	✓	669	7261	7261	✓	905	3676	3676	✓
198	9050	9050	✓	434	2395	2395	✓	670	15452	15452	✓	906	5724	5724	✓
199	861	861	✓	435	14684	14684	✓	671	600	600	✓	907	14939	14939	✓
200	15705	15705	✓	436	9560	9560	✓	672	8798	8798	✓	908	10847	10847	✓
201	7513	7513	✓	437	5465	5465	✓	673	4701	4701	✓	909	2649	2649	✓
202	11608	11608	✓	438	3417	3417	✓	674	2652	2652	✓	910	4703	4703	✓
203	13658	13658	✓	439	7514	7514	✓	675	10846	10846	✓	911	8794	8794	✓
204	9566	9566	✓	440	15706	15706	✓	676	14942	14942	✓	912	606	606	✓
205	14686	14686	✓	441	862	862	✓	677	9822	9822	✓	913	15450	15450	✓
206	10589	10589	✓	442	863	863	✓	678	13919	13919	✓	914	7257	7257	✓
207	12638	12638	✓	443	9050	9050	✓	679	7772	7772	✓	915	7262	7262	✓
208	8538	8538	✓	444	9051	9051	✓	680	15967	15967	✓	916	7257	7257	✓
209	15960	15960	✓	445	9055	9055	✓	681	4442	4442	✓	917	7259	7259	✓
210	3679	3679	✓	446	9050	9050	✓	682	2395	2395	✓	918	15450	15450	✓
211	5726	5726	✓	447	857	857	✓	683	6491	6491	✓	919	600	600	✓
212	1624	1624	✓	448	856	856	✓	684	9560	9560	✓	920	8792	8792	✓
213	6745	6745	✓	449	7517	7517	✓	685	5470	5470	✓	921	4698	4698	✓
214	2653	2653	✓	450	11610	11610	✓	686	13663	13663	✓	922	2648	2648	✓

215	4701	4701	✓	451	13659	13659	✓	687	11613	11613	✓	923	10843	10843	✓
216	8798	8798	✓	452	9561	9561	✓	688	15704	15704	✓	924	14941	14941	✓
217	602	602	✓	453	1368	1368	✓	689	858	858	✓	925	9818	9818	✓
218	15448	15448	✓	454	10587	10587	✓	690	9048	9048	✓	926	13917	13917	✓
219	7257	7257	✓	455	12635	12635	✓	691	9049	9049	✓	927	11871	11871	✓
220	7262	7262	✓	456	8537	8537	✓	692	4952	4952	✓	928	344	344	✓
221	7256	7256	✓	457	15964	15964	✓	693	9055	9055	✓	929	8543	8543	✓
222	7260	7260	✓	458	3679	3679	✓	694	9049	9049	✓	930	2398	2398	✓
223	7257	7257	✓	459	13912	13912	✓	695	9048	9048	✓	931	6492	6492	✓
224	15450	15450	✓	460	1631	1631	✓	696	860	860	✓	932	1370	1370	✓
225	605	605	✓	461	6745	6745	✓	697	7517	7517	✓	933	5464	5464	✓
226	4700	4700	✓	462	2655	2655	✓	698	11614	11614	✓	934	3416	3416	✓
227	12889	12889	✓	463	12888	12888	✓	699	3416	3416	✓	935	11613	11613	✓
228	10844	10844	✓	464	8799	8799	✓	700	5468	5468	✓	936	15704	15704	✓
229	6749	6749	✓	465	601	601	✓	701	1368	1368	✓	937	856	856	✓
230	1627	1627	✓	466	15449	15449	✓	702	6488	6488	✓	938	863	863	✓
231	13916	13916	✓	467	7263	7263	✓	703	2392	2392	✓	939	9049	9049	✓
232	11870	11870	✓	468	7257	7257	✓	704	8541	8541	✓	940	4952	4952	✓
233	15965	15965	✓	469	7258	7258	✓	705	15965	15965	✓	941	4952	4952	✓
234	8541	8541	✓	470	7257	7257	✓	706	3679	3679	✓	942	9053	9053	✓
235	2392	2392	✓	471	7259	7259	✓	707	13912	13912	✓	943	861	861	✓
236	10587	10587	✓	472	15453	15453	✓	708	1629	1629	✓	944	860	860	✓

3.3.4 Prueba 4

Utilizando la misma configuración y distancia de separación similar a la Prueba 3, se reprograma un nuevo algoritmo en la recepción, que genera la activación de la rutina de interrupción de INT0, al recibir correctamente la cabecera del paquete de datos. La ISR de INT0 realiza la tarea de modificar el valor del registro TMR0, y así sincronizar la lectura de los bits del paquete de datos, que corresponden a una onda senoidal. Esta sincronización dentro de la tarea de leer los bits de datos generan una corrección del desfase del tiempo de bit correspondiente a una velocidad de 19200 bps. El tamaño de paquete se limitó a 2 bytes. Según los datos obtenidos tal como lo muestra en la Tabla 3.3.4.1, podemos afirmar que este método nos permitirá la eliminación de errores recibidos y nos plantea la posibilidad de aumentar el número de bytes del paquete de datos de transferencia.

Esta nueva versión de la rutina de lectura de la cadena de bits es importante porque permite un máximo desfase acumulado de 8 bits, debido a la codificación balanceada, logrando entonces limitar el desfase a 8 veces el desfase de cada bit.

Tabla 3.3.4.1 Valores Transmitidos y Recibidos por el registrador, usando los transceivers, Prueba 4

DATOS TRANSMITIDOS Y RECIBIDOS UTILIZANDO MODULOS TRANSMISOR Y RECEPTOR																
N	D Tx	D Rx	Dato	N	D Tx	D Rx	Dato	N	D Tx	D Rx	Dato	N	D Tx	D Rx	Dato	
1	5468	5468	✓	247	11613	11613	✓	493	858	858	✓	739	9055	9055	✓	
2	1374	1374	✓	248	3418	3418	✓	494	15710	15710	✓	740	9050	9050	✓	
3	6489	6489	✓	249	5470	5470	✓	495	11615	11615	✓	741	862	862	✓	
4	2396	2396	✓	250	1373	1373	✓	496	3417	3417	✓	742	15708	15708	✓	
5	8539	8539	✓	251	6494	6494	✓	497	5467	5467	✓	743	7516	7516	✓	
6	15963	15963	✓	252	2398	2398	✓	498	1371	1371	✓	744	3419	3419	✓	
7	11868	11868	✓	253	4444	4444	✓	499	6495	6495	✓	745	5471	5471	✓	
8	13918	13918	✓	254	15965	15965	✓	500	2395	2395	✓	746	1371	1371	✓	
9	9822	9822	✓	255	11867	11867	✓	501	4445	4445	✓	747	14684	14684	✓	
10	14938	14938	✓	256	13917	13917	✓	502	344	344	✓	748	2395	2395	✓	
11	2655	2655	✓	257	9820	9820	✓	503	7770	7770	✓	749	4443	4443	✓	
12	12894	12894	✓	258	14937	14937	✓	504	13919	13919	✓	750	345	345	✓	
13	4696	4696	✓	259	10846	10846	✓	505	9819	9819	✓	751	7770	7770	✓	
14	8792	8792	✓	260	2648	2648	✓	506	14943	14943	✓	752	13919	13919	✓	
15	600	600	✓	261	4698	4698	✓	507	10846	10846	✓	753	5720	5720	✓	
16	7263	7263	✓	262	607	607	✓	508	12895	12895	✓	754	14939	14939	✓	
17	7260	7260	✓	263	600	600	✓	509	4701	4701	✓	755	10843	10843	✓	
18	7260	7260	✓	264	15452	15452	✓	510	8792	8792	✓	756	2650	2650	✓	
19	7261	7261	✓	265	7259	7259	✓	511	600	600	✓	757	4698	4698	✓	
20	7261	7261	✓	266	7258	7258	✓	512	15450	15450	✓	758	8796	8796	✓	
21	15454	15454	✓	267	7261	7261	✓	513	7259	7259	✓	759	600	600	✓	
22	604	604	✓	268	7262	7262	✓	514	7262	7262	✓	760	15448	15448	✓	
23	8795	8795	✓	269	15452	15452	✓	515	7257	7257	✓	761	7261	7261	✓	
24	4703	4703	✓	270	604	604	✓	516	7259	7259	✓	762	7262	7262	✓	
25	2652	2652	✓	271	8798	8798	✓	517	7263	7263	✓	763	7258	7258	✓	
26	6748	6748	✓	272	4699	4699	✓	518	604	604	✓	764	7261	7261	✓	
27	1626	1626	✓	273	2654	2654	✓	519	8794	8794	✓	765	15452	15452	✓	
28	5724	5724	✓	274	6744	6744	✓	520	4702	4702	✓	766	15455	15455	✓	
29	3678	3678	✓	275	14943	14943	✓	521	2650	2650	✓	767	8792	8792	✓	
30	7775	7775	✓	276	9823	9823	✓	522	10841	10841	✓	768	4700	4700	✓	
31	8540	8540	✓	277	3673	3673	✓	523	14938	14938	✓	769	2648	2648	✓	
32	12632	12632	✓	278	7769	7769	✓	524	9819	9819	✓	770	10846	10846	✓	
33	10588	10588	✓	279	349	349	✓	525	3676	3676	✓	771	14941	14941	✓	
34	14680	14680	✓	280	12636	12636	✓	526	7772	7772	✓	772	9817	9817	✓	
35	9564	9564	✓	281	2399	2399	✓	527	346	346	✓	773	13917	13917	✓	
36	13660	13660	✓	282	14682	14682	✓	528	4445	4445	✓	774	11871	11871	✓	
37	3417	3417	✓	283	9564	9564	✓	529	2395	2395	✓	775	346	346	✓	
38	7514	7514	✓	284	5467	5467	✓	530	14680	14680	✓	776	4446	4446	✓	
39	15709	15709	✓	285	3422	3422	✓	531	1371	1371	✓	777	2394	2394	✓	
40	861	861	✓	286	11615	11615	✓	532	5469	5469	✓	778	6493	6493	✓	
41	9052	9052	✓	287	15704	15704	✓	533	3422	3422	✓	779	1369	1369	✓	
42	9053	9053	✓	288	861	861	✓	534	11615	11615	✓	780	5469	5469	✓	
43	4952	4952	✓	289	9054	9054	✓	535	15706	15706	✓	781	3420	3420	✓	
44	9053	9053	✓	290	9053	9053	✓	536	860	860	✓	782	11613	11613	✓	
45	9050	9050	✓	291	9055	9055	✓	537	861	861	✓	783	15704	15704	✓	
46	859	859	✓	292	9049	9049	✓	538	9051	9051	✓	784	860	860	✓	
47	15709	15709	✓	293	9049	9049	✓	539	9054	9054	✓	785	9048	9048	✓	
48	7513	7513	✓	294	857	857	✓	540	9049	9049	✓	786	9053	9053	✓	
49	3419	3419	✓	295	860	860	✓	541	9049	9049	✓	787	9055	9055	✓	
50	13662	13662	✓	296	7515	7515	✓	542	857	857	✓	788	9055	9055	✓	
51	9563	9563	✓	297	3423	3423	✓	543	856	856	✓	789	9051	9051	✓	
52	14685	14685	✓	298	13659	13659	✓	544	7519	7519	✓	790	863	863	✓	
53	10590	10590	✓	299	9567	9567	✓	545	11609	11609	✓	791	860	860	✓	
54	12638	12638	✓	300	14685	14685	✓	546	13663	13663	✓	792	15706	15706	✓	
55	349	349	✓	301	10587	10587	✓	547	5464	5464	✓	793	11612	11612	✓	
56	7771	7771	✓	302	12637	12637	✓	548	14687	14687	✓	794	3418	3418	✓	
57	3673	3673	✓	303	8541	8541	✓	549	6490	6490	✓	795	9567	9567	✓	
58	5726	5726	✓	304	15962	15962	✓	550	12635	12635	✓	796	1373	1373	✓	
59	1626	1626	✓	305	3679	3679	✓	551	8537	8537	✓	797	10591	10591	✓	
60	6749	6749	✓	306	5721	5721	✓	552	15964	15964	✓	798	2392	2392	✓	
61	2654	2654	✓	307	1625	1625	✓	553	11864	11864	✓	799	4444	4444	✓	
62	4699	4699	✓	308	6750	6750	✓	554	13914	13914	✓	800	15965	15965	✓	
63	8797	8797	✓	309	2651	2651	✓	555	1625	1625	✓	801	11869	11869	✓	
64	604	604	✓	310	12890	12890	✓	556	6747	6747	✓	802	13914	13914	✓	
65	15450	15450	✓	311	8797	8797	✓	557	10844	10844	✓	803	1627	1627	✓	
66	15448	15448	✓	312	603	603	✓	558	12889	12889	✓	804	14936	14936	✓	
67	7262	7262	✓	313	15454	15454	✓	559	4696	4696	✓	805	10842	10842	✓	
68	7257	7257	✓	314	7257	7257	✓	560	601	601	✓	806	12893	12893	✓	
69	7261	7261	✓	315	7262	7262	✓	561	15451	15451	✓	807	4700	4700	✓	
70	7257	7257	✓	316	7256	7256	✓	562	7263	7263	✓	808	607	607	✓	

Capítulo 3 : Pruebas y Resultados del Transceiver Digital en RF

71	15449	15449	✓	317	7257	7257	✓	563	7256	7256	✓	809	15455	15455	✓
72	8792	8792	✓	318	15448	15448	✓	564	7257	7257	✓	810	7257	7257	✓
73	8799	8799	✓	319	15451	15451	✓	565	7257	7257	✓	811	7262	7262	✓
74	12892	12892	✓	320	605	605	✓	566	7259	7259	✓	812	7260	7260	✓
75	10842	10842	✓	321	8795	8795	✓	567	15451	15451	✓	813	7258	7258	✓
76	6751	6751	✓	322	12893	12893	✓	568	606	606	✓	814	7263	7263	✓
77	1627	1627	✓	323	2655	2655	✓	569	8793	8793	✓	815	15454	15454	✓
78	13912	13912	✓	324	6750	6750	✓	570	12888	12888	✓	816	604	604	✓
79	11870	11870	✓	325	1629	1629	✓	571	2655	2655	✓	817	8793	8793	✓
80	15966	15966	✓	326	5725	5725	✓	572	6750	6750	✓	818	4703	4703	✓
81	8539	8539	✓	327	11866	11866	✓	573	1630	1630	✓	819	2653	2653	✓
82	12637	12637	✓	328	15962	15962	✓	574	5721	5721	✓	820	6750	6750	✓
83	10591	10591	✓	329	8538	8538	✓	575	3677	3677	✓	821	1626	1626	✓
84	1368	1368	✓	330	12638	12638	✓	576	7775	7775	✓	822	9823	9823	✓
85	5468	5468	✓	331	6488	6488	✓	577	347	347	✓	823	3677	3677	✓
86	3416	3416	✓	332	14683	14683	✓	578	12638	12638	✓	824	7773	7773	✓
87	11609	11609	✓	333	9565	9565	✓	579	10585	10585	✓	825	345	345	✓
88	7517	7517	✓	334	13662	13662	✓	580	14683	14683	✓	826	12634	12634	✓
89	15705	15705	✓	335	11614	11614	✓	581	9566	9566	✓	827	10588	10588	✓
90	863	863	✓	336	7517	7517	✓	582	13657	13657	✓	828	14682	14682	✓
91	9054	9054	✓	337	856	856	✓	583	11612	11612	✓	829	9564	9564	✓
92	9053	9053	✓	338	863	863	✓	584	7512	7512	✓	830	13656	13656	✓
93	4952	4952	✓	339	9052	9052	✓	585	15710	15710	✓	831	3423	3423	✓
94	9050	9050	✓	340	9055	9055	✓	586	859	859	✓	832	7516	7516	✓
95	859	859	✓	341	9051	9051	✓	587	9050	9050	✓	833	15709	15709	✓
96	858	858	✓	342	9053	9053	✓	588	9053	9053	✓	834	859	859	✓
97	15710	15710	✓	343	863	863	✓	589	4952	4952	✓	835	9052	9052	✓
98	7514	7514	✓	344	857	857	✓	590	9051	9051	✓	836	9053	9053	✓
99	3421	3421	✓	345	15710	15710	✓	591	9048	9048	✓	837	9055	9055	✓
100	5467	5467	✓	346	7514	7514	✓	592	859	859	✓	838	9053	9053	✓
101	1375	1375	✓	347	3418	3418	✓	593	15709	15709	✓	839	9048	9048	✓
102	14680	14680	✓	348	13660	13660	✓	594	11615	11615	✓	840	859	859	✓
103	2398	2398	✓	349	9564	9564	✓	595	3419	3419	✓	841	15710	15710	✓
104	4440	4440	✓	350	14684	14684	✓	596	13660	13660	✓	842	7513	7513	✓
105	348	348	✓	351	2393	2393	✓	597	1375	1375	✓	843	3421	3421	✓
106	7772	7772	✓	352	4445	4445	✓	598	14685	14685	✓	844	13657	13657	✓
107	13918	13918	✓	353	345	345	✓	599	10590	10590	✓	845	9565	9565	✓
108	9819	9819	✓	354	7774	7774	✓	600	4447	4447	✓	846	14685	14685	✓
109	14942	14942	✓	355	3672	3672	✓	601	8536	8536	✓	847	10589	10589	✓
110	10845	10845	✓	356	9823	9823	✓	602	7773	7773	✓	848	12634	12634	✓
111	2650	2650	✓	357	1624	1624	✓	603	3672	3672	✓	849	347	347	✓
112	4701	4701	✓	358	6744	6744	✓	604	5724	5724	✓	850	7775	7775	✓
113	8792	8792	✓	359	2650	2650	✓	605	14943	14943	✓	851	3673	3673	✓
114	15455	15455	✓	360	4701	4701	✓	606	6750	6750	✓	852	5724	5724	✓
115	15452	15452	✓	361	603	603	✓	607	2653	2653	✓	853	1630	1630	✓
116	7261	7261	✓	362	604	604	✓	608	4699	4699	✓	854	6748	6748	✓
117	7257	7257	✓	363	15452	15452	✓	609	8793	8793	✓	855	2649	2649	✓
118	7258	7258	✓	364	7259	7259	✓	610	15455	15455	✓	856	4703	4703	✓
119	7258	7258	✓	365	7261	7261	✓	611	15452	15452	✓	857	8794	8794	✓
120	15450	15450	✓	366	7256	7256	✓	612	7263	7263	✓	858	605	605	✓
121	600	600	✓	367	7257	7257	✓	613	7260	7260	✓	859	15449	15449	✓
122	8792	8792	✓	368	15452	15452	✓	614	7257	7257	✓	860	7263	7263	✓
123	4698	4698	✓	369	15451	15451	✓	615	7261	7261	✓	861	7258	7258	✓
124	2650	2650	✓	370	8792	8792	✓	616	7261	7261	✓	862	11359	11359	✓
125	6744	6744	✓	371	4702	4702	✓	617	15453	15453	✓	863	7262	7262	✓
126	14943	14943	✓	372	12895	12895	✓	618	8792	8792	✓	864	7259	7259	✓
127	9819	9819	✓	373	10845	10845	✓	619	4700	4700	✓	865	15449	15449	✓
128	13919	13919	✓	374	14938	14938	✓	620	12891	12891	✓	866	603	603	✓
129	7770	7770	✓	375	9822	9822	✓	621	10841	10841	✓	867	8797	8797	✓
130	346	346	✓	376	13915	13915	✓	622	14938	14938	✓	868	12891	12891	✓
131	4446	4446	✓	377	11869	11869	✓	623	9817	9817	✓	869	2651	2651	✓
132	2395	2395	✓	378	344	344	✓	624	13913	13913	✓	870	14940	14940	✓
133	6491	6491	✓	379	4444	4444	✓	625	11869	11869	✓	871	1625	1625	✓
134	1375	1375	✓	380	2397	2397	✓	626	344	344	✓	872	13914	13914	✓
135	5471	5471	✓	381	6490	6490	✓	627	4440	4440	✓	873	11868	11868	✓
136	3421	3421	✓	382	1374	1374	✓	628	2394	2394	✓	874	15966	15966	✓
137	11609	11609	✓	383	5470	5470	✓	629	10591	10591	✓	875	8539	8539	✓
138	15708	15708	✓	384	3418	3418	✓	630	1370	1370	✓	876	12639	12639	✓
139	857	857	✓	385	11615	11615	✓	631	5470	5470	✓	877	6490	6490	✓
140	9050	9050	✓	386	15708	15708	✓	632	3420	3420	✓	878	1370	1370	✓
141	9052	9052	✓	387	860	860	✓	633	11613	11613	✓	879	5466	5466	✓
142	9055	9055	✓	388	9052	9052	✓	634	7519	7519	✓	880	13662	13662	✓
143	9051	9051	✓	389	9054	9054	✓	635	858	858	✓	881	11610	11610	✓
144	9050	9050	✓	390	9053	9053	✓	636	857	857	✓	882	7518	7518	✓
145	9052	9052	✓	391	4952	4952	✓	637	9050	9050	✓	883	15711	15711	✓
146	856	856	✓	392	9053	9053	✓	638	9049	9049	✓	884	9048	9048	✓

147	15704	15704	✓	393	9048	9048	✓	639	9051	9051	✓	885	9050	9050	✓
148	11612	11612	✓	394	858	858	✓	640	9049	9049	✓	886	9053	9053	✓
149	13662	13662	✓	395	15708	15708	✓	641	861	861	✓	887	4952	4952	✓
150	5464	5464	✓	396	11614	11614	✓	642	860	860	✓	888	9054	9054	✓
151	1368	1368	✓	397	13663	13663	✓	643	15706	15706	✓	889	9048	9048	✓
152	10587	10587	✓	398	5470	5470	✓	644	11615	11615	✓	890	857	857	✓
153	12637	12637	✓	399	1374	1374	✓	645	3416	3416	✓	891	15706	15706	✓
154	8543	8543	✓	400	6490	6490	✓	646	5467	5467	✓	892	7512	7512	✓
155	15961	15961	✓	401	2396	2396	✓	647	1370	1370	✓	893	3417	3417	✓
156	11864	11864	✓	402	8537	8537	✓	648	6491	6491	✓	894	5465	5465	✓
157	13914	13914	✓	403	15966	15966	✓	649	2396	2396	✓	895	1369	1369	✓
158	1629	1629	✓	404	11870	11870	✓	650	8539	8539	✓	896	6495	6495	✓
159	6747	6747	✓	405	13916	13916	✓	651	348	348	✓	897	2393	2393	✓
160	10844	10844	✓	406	9818	9818	✓	652	11870	11870	✓	898	4441	4441	✓
161	12894	12894	✓	407	14940	14940	✓	653	13918	13918	✓	899	346	346	✓
162	4700	4700	✓	408	10841	10841	✓	654	9818	9818	✓	900	11871	11871	✓
163	602	602	✓	409	12892	12892	✓	655	14937	14937	✓	901	3672	3672	✓
164	15451	15451	✓	410	4696	4696	✓	656	10841	10841	✓	902	9817	9817	✓
165	15448	15448	✓	411	607	607	✓	657	12895	12895	✓	903	14939	14939	✓
166	7261	7261	✓	412	15451	15451	✓	658	8795	8795	✓	904	10843	10843	✓
167	11359	11359	✓	413	15452	15452	✓	659	603	603	✓	905	2648	2648	✓
168	7258	7258	✓	414	7258	7258	✓	660	600	600	✓	906	4702	4702	✓
169	7259	7259	✓	415	7257	7257	✓	661	15448	15448	✓	907	8794	8794	✓
170	15454	15454	✓	416	7261	7261	✓	662	7261	7261	✓	908	15451	15451	✓
171	606	606	✓	417	7259	7259	✓	663	11359	11359	✓	909	15452	15452	✓
172	8795	8795	✓	418	15453	15453	✓	664	7258	7258	✓	910	7257	7257	✓
173	12890	12890	✓	419	606	606	✓	665	7263	7263	✓	911	7257	7257	✓
174	2654	2654	✓	420	8794	8794	✓	666	15450	15450	✓	912	7256	7256	✓
175	6748	6748	✓	421	4703	4703	✓	667	606	606	✓	913	7261	7261	✓
176	1625	1625	✓	422	2649	2649	✓	668	607	607	✓	914	15449	15449	✓
177	5722	5722	✓	423	6746	6746	✓	669	4703	4703	✓	915	602	602	✓
178	3675	3675	✓	424	1624	1624	✓	670	2654	2654	✓	916	8796	8796	✓
179	7775	7775	✓	425	5724	5724	✓	671	6746	6746	✓	917	4697	4697	✓
180	8542	8542	✓	426	3678	3678	✓	672	1628	1628	✓	918	2652	2652	✓
181	12638	12638	✓	427	7771	7771	✓	673	5720	5720	✓	919	10841	10841	✓
182	10588	10588	✓	428	345	345	✓	674	3674	3674	✓	920	14939	14939	✓
183	14685	14685	✓	429	12636	12636	✓	675	7769	7769	✓	921	9819	9819	✓
184	9563	9563	✓	430	10590	10590	✓	676	346	346	✓	922	3672	3672	✓
185	13658	13658	✓	431	14686	14686	✓	677	4446	4446	✓	923	11867	11867	✓
186	11608	11608	✓	432	9561	9561	✓	678	2399	2399	✓	924	349	349	✓
187	7518	7518	✓	433	13656	13656	✓	679	14682	14682	✓	925	4441	4441	✓
188	15707	15707	✓	434	3419	3419	✓	680	9560	9560	✓	926	2398	2398	✓
189	857	857	✓	435	7516	7516	✓	681	5467	5467	✓	927	6495	6495	✓
190	9048	9048	✓	436	15705	15705	✓	682	3419	3419	✓	928	9560	9560	✓
191	9049	9049	✓	437	857	857	✓	683	11615	11615	✓	929	5466	5466	✓
192	4952	4952	✓	438	9054	9054	✓	684	15710	15710	✓	930	3422	3422	✓
193	9051	9051	✓	439	9054	9054	✓	685	860	860	✓	931	7512	7512	✓
194	863	863	✓	440	9051	9051	✓	686	9054	9054	✓	932	15706	15706	✓
195	860	860	✓	441	9049	9049	✓	687	9050	9050	✓	933	857	857	✓
196	15705	15705	✓	442	9048	9048	✓	688	4956	4956	✓	934	863	863	✓
197	7516	7516	✓	443	861	861	✓	689	9050	9050	✓	935	9051	9051	✓
198	3423	3423	✓	444	856	856	✓	690	9053	9053	✓	936	9054	9054	✓
199	13658	13658	✓	445	7517	7517	✓	691	858	858	✓	937	4952	4952	✓
200	9564	9564	✓	446	11612	11612	✓	692	860	860	✓	938	9049	9049	✓
201	14680	14680	✓	447	13661	13661	✓	693	7515	7515	✓	939	859	859	✓
202	10586	10586	✓	448	9562	9562	✓	694	11612	11612	✓	940	15711	15711	✓
203	4443	4443	✓	449	14686	14686	✓	695	13663	13663	✓	941	7515	7515	✓
204	347	347	✓	450	10589	10589	✓	696	9567	9567	✓	942	11614	11614	✓
205	7771	7771	✓	451	12634	12634	✓	697	1372	1372	✓	943	13657	13657	✓
206	3672	3672	✓	452	8538	8538	✓	698	10590	10590	✓	944	5466	5466	✓
207	5724	5724	✓	453	7771	7771	✓	699	12637	12637	✓	945	1372	1372	✓
208	1624	1624	✓	454	3673	3673	✓	700	8542	8542	✓	946	6490	6490	✓
209	6750	6750	✓	455	5726	5726	✓	701	15964	15964	✓	947	12635	12635	✓
210	2653	2653	✓	456	1629	1629	✓	702	11864	11864	✓	948	8537	8537	✓
211	4697	4697	✓	457	6745	6745	✓	703	5721	5721	✓	949	15961	15961	✓
212	8794	8794	✓	458	2654	2654	✓	704	1625	1625	✓	950	11866	11866	✓
213	601	601	✓	459	4703	4703	✓	705	6747	6747	✓	951	13914	13914	✓
214	15454	15454	✓	460	8798	8798	✓	706	2649	2649	✓	952	1629	1629	✓
215	7259	7259	✓	461	606	606	✓	707	12892	12892	✓	953	6751	6751	✓
216	7262	7262	✓	462	15450	15450	✓	708	8795	8795	✓	954	2655	2655	✓
217	7256	7256	✓	463	7261	7261	✓	709	601	601	✓	955	12894	12894	✓
218	7259	7259	✓	464	7262	7262	✓	710	15450	15450	✓	956	4696	4696	✓
219	7261	7261	✓	465	7258	7258	✓	711	7263	7263	✓	957	606	606	✓
220	15455	15455	✓	466	7261	7261	✓	712	7260	7260	✓	958	15453	15453	✓
221	8792	8792	✓	467	7263	7263	✓	713	7260	7260	✓	959	7261	7261	✓
222	4696	4696	✓	468	15451	15451	✓	714	7257	7257	✓	960	7260	7260	✓

223	12895	12895	✓	469	605	605	✓	715	7263	7263	✓	961	7257	7257	✓
224	10845	10845	✓	470	4698	4698	✓	716	15452	15452	✓	962	7258	7258	✓
225	6751	6751	✓	471	12894	12894	✓	717	601	601	✓	963	7261	7261	✓
226	9817	9817	✓	472	10844	10844	✓	718	8797	8797	✓	964	15449	15449	✓
227	13913	13913	✓	473	6747	6747	✓	719	12892	12892	✓	965	601	601	✓
228	11870	11870	✓	474	1631	1631	✓	720	10844	10844	✓	966	8793	8793	✓
229	344	344	✓	475	13916	13916	✓	721	6745	6745	✓	967	4703	4703	✓
230	8543	8543	✓	476	3679	3679	✓	722	1629	1629	✓	968	2655	2655	✓
231	2398	2398	✓	477	15962	15962	✓	723	5725	5725	✓	969	6745	6745	✓
232	10591	10591	✓	478	8541	8541	✓	724	11866	11866	✓	970	1630	1630	✓
233	1369	1369	✓	479	12638	12638	✓	725	15960	15960	✓	971	5724	5724	✓
234	9563	9563	✓	480	6488	6488	✓	726	8538	8538	✓	972	11864	11864	✓
235	3418	3418	✓	481	1368	1368	✓	727	12633	12633	✓	973	7775	7775	✓
236	11613	11613	✓	482	5466	5466	✓	728	10587	10587	✓	974	8542	8542	✓
237	7519	7519	✓	483	13662	13662	✓	729	14683	14683	✓	975	12634	12634	✓
238	856	856	✓	484	11610	11610	✓	730	5464	5464	✓	976	10589	10589	✓
239	863	863	✓	485	7515	7515	✓	731	13662	13662	✓	977	14681	14681	✓
240	9050	9050	✓	486	15707	15707	✓	732	11614	11614	✓	978	9561	9561	✓
241	9054	9054	✓	487	861	861	✓	733	7513	7513	✓	979	13660	13660	✓
242	9053	9053	✓	488	9054	9054	✓	734	15709	15709	✓	980	11610	11610	✓
243	9051	9051	✓	489	9051	9051	✓	735	861	861	✓	981	7512	7512	✓
244	9052	9052	✓	490	4956	4956	✓	736	9054	9054	✓	982	15711	15711	✓
245	15711	15711	✓	491	9051	9051	✓	737	9055	9055	✓	983	862	862	✓
246	15710	15710	✓	492	9052	9052	✓	738	9053	9053	✓				

3.3.5 Pruebas de Alcance de Comunicación

Esta prueba se realizó con el objetivo de medir el alcance máximo de enlace de comunicación con vista directa entre los transceivers. Dentro de los datos del fabricante menciona que se puede lograr un alcance de aproximadamente 101 metros con línea de vista libre. En pruebas realizadas pudimos lograr un enlace a 89 metros y 91 metros, pudiendo concluir que se podría ampliar este alcance usando una antena direccionable como una antena del tipo yagui.

CAPITULO IV

Conclusiones y Recomendaciones

4. Conclusiones y Recomendaciones

4.1 Conclusiones

- 1) Las comunicaciones digitales por RF constituyen una técnica interesante y atrayente por sus notables ventajas y múltiples aplicaciones de enlace con diversos equipos remotos.
- 2) Es factible el diseñar e implementar transceivers usando la tecnología de RF, con el objetivo de enlazar registradores sísmicos remotos con una estación central.
- 3) Para el usuario de registradores sísmicos es transparente usar los transceiver, es decir, los programas proporcionados por el fabricante del registrador se usan de una manera similar como si estuvieran directamente conectados con un cable.
- 4) El transceiver inalámbrico en RF implementado, de corto alcance puede ser configurado para una instrumentación de áreas pequeñas, con un radio de

aproximadamente 100 m de vista libre, en lugares como terrenos, casas y edificios. Logrando instalar una red sismotelemétrica de corto alcance.

- 5) Los microcontroladores PIC de la gama media soportaron los programas y rutinas de los transceivers, lo que nos asegura mejoras en el uso de PICs de la serie 18.
- 6) Al tener en la estación central la señal de la estación remota puede ser fácilmente identificada e integrada dentro de una red Sísmica para su gestión de la información.

4.2 Recomendaciones

- 1) Se pueden realizar aplicaciones de enlaces en RF dentro de otros campos de aplicación. Podemos mencionar estaciones climatológicas o metereológicas, o unidades y dispositivos de control remoto, o dataloggers que requieren tener un enlace y formar parte de un sistema o red.
- 2) Transceiver implementado puede ser mejorado en su operación separando cada módulo transmisor y receptor en un microcontrolador separado, resultando una mejor respuesta de funcionamiento. En todo caso, implementar los algoritmos del transceiver en un microprocesador.
- 3) Para lograr mayor alcance se debe agregar un módulo de potencia al equipo de comunicación. Existen en el mercado diversos fabricantes de módulos amplificadores en RF, que proporcionan potencias de 1, 3, 5 Watts, logrando incrementar el alcance de enlace de comunicación punto a punto dentro del rango de 10 a 50 Km. Además, hay que considerar dentro del diseño un switch de antena, para poder amplificar la onda transmitida y no la recibida, tal como se muestra en la Fig. 4.2.1.

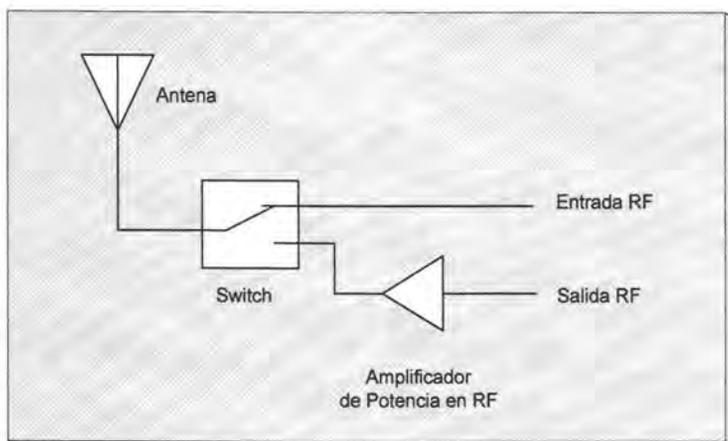


Fig. 4.2.1. Amplificador y Switch en RF

4) Se puede realizar una etapa de empaquetado de antes en la etapa previa a la codificación, es decir de los bytes recibidos por el puerto RS232, con el objetivo de poder ampliar la cantidad de datos a ser transmitidos (ampliar el ancho de banda del enlace).

5) Podemos ampliar la velocidad del transceiver utilizando un módulo de RF Monolithics (DR4000) ^[7], que actualmente soportan una velocidad de hasta 115.2 kbps. Igualmente se tendría que utilizar un microcontrolador o microprocesador con un oscilador mucho mayor que el de 20 MHz utilizado en este proyecto de tesis.

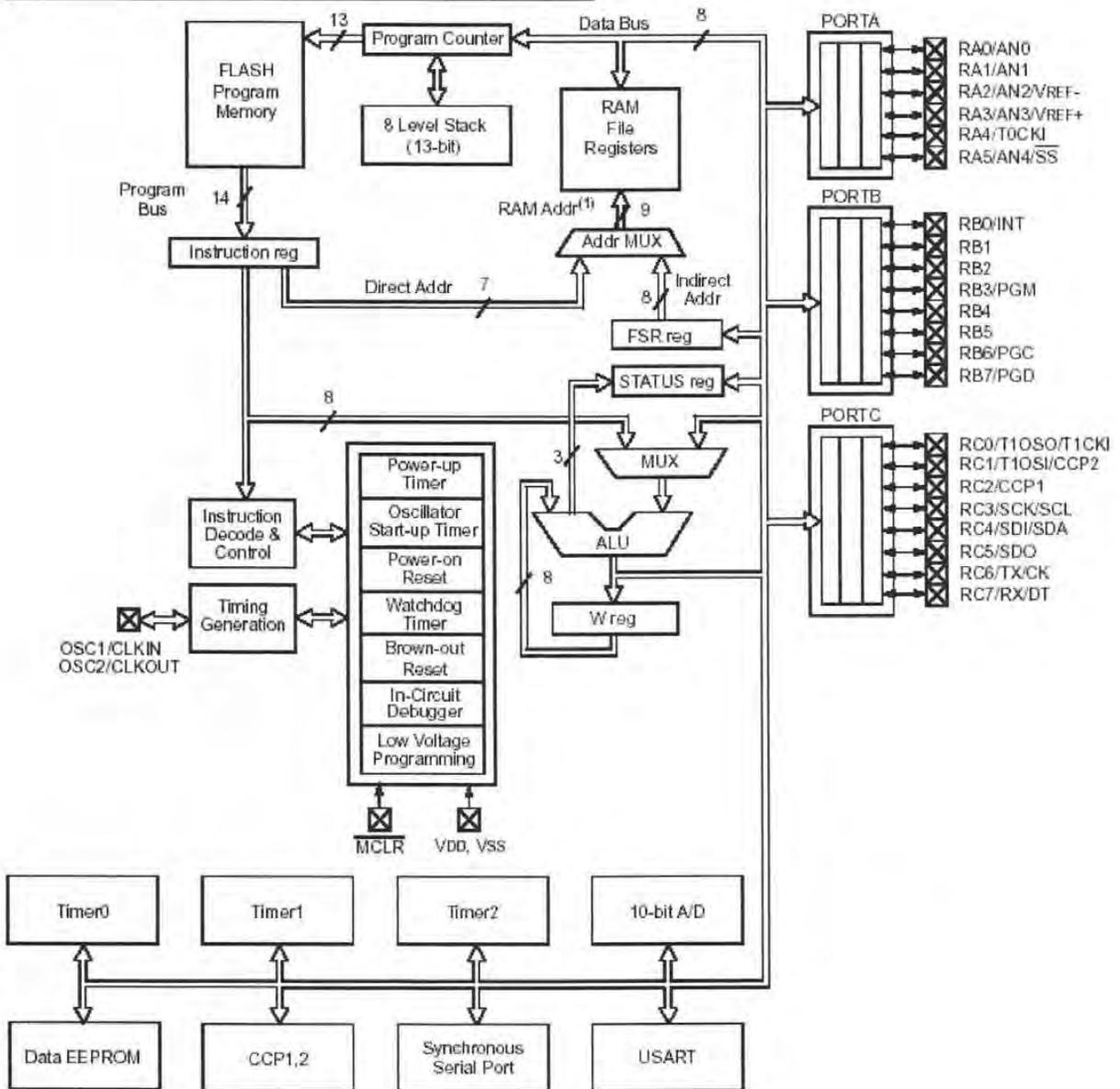
6) En la estación central se puede personalizar e implementar algún programa de adquisición y registro de los archivos de eventos; y configurándolo como una Red Sismotelemétrica. Se recomienda usar el programa Seislog, programa que brinda un buen funcionamiento para este fin.

APENDICES

A. MICROCONTROLADOR PIC16F873A-20

A1. Diagrama de Bloques del Microcontrolador PIC16F873

Device	Program FLASH	Data Memory	Data EEPROM
PIC16F873	4K	192 Bytes	128 Bytes



A2. Distribución de los Registros en Memoria RAM

Dirección		Dirección		Dirección		Dirección	
Direc. Indir. (*)	00h	Direc. Indir. (*)	80h	Direc. Indir. (*)	100h	Direc. Indir. (*)	180h
TMR0	01h	OPTION_REG	81h	TMR0	101h	OPTION_REG	181h
PCL	02h	PCL	82h	PCL	102h	PCL	182h
STATUS	03h	STATUS	83h	STATUS	103h	STATUS	183h
FSR	04h	FSR	84h	FSR	104h	FSR	184h
PORTA	05h	TRISA	85h		105h		185h
PORTB	06h	TRISB	86h	PORTB	106h	TRISB	186h
PORTC	07h	TRISC	87h		107h		187h
PORTD ⁽¹⁾	08h	TRISD ⁽¹⁾	88h		108h		188h
PORTE ⁽¹⁾	09h	TRISE ⁽¹⁾	89h		109h		189h
PCLATH	0Ah	PCLATH	8Ah	PCLATH	10Ah	PCLATH	18Ah
INTCON	0Bh	INTCON	8Bh	INTCON	10Bh	INTCON	18Bh
PIR1	0Ch	PIE1	8Ch	EEDATA	10Ch	ECON1	18Ch
PIR2	0Dh	PIE2	8Dh	EEADR	10Dh	ECON2	18Dh
TMR1L	0Eh	PCON	8Eh	EEDATH	10Eh	Reserved ⁽²⁾	18Eh
TMR1H	0Fh		8Fh	EEADRH	10Fh	Reserved ⁽²⁾	18Fh
T1CON	10h		90h		110h		190h
TMR2	11h	SSPCON2	91h				
T2CON	12h	PR2	92h				
SSPBUF	13h	SSPADD	93h				
SSPCON	14h	SSPSTAT	94h				
CCPR1L	15h		95h				
CCPR1H	16h		96h				
CCP1CON	17h		97h				
RCSTA	18h	TXSTA	98h				
TXREG	19h	SPBRG	99h				
RCREG	1Ah		9Ah				
CCPR2L	1Bh		9Bh				
CCPR2H	1Ch		9Ch				
CCP2CON	1Dh		9Dh				
ADRESH	1Eh	ADRESL	9Eh				
ADCON0	1Fh	ADCON1	9Fh				
	20h		A0h		120h		1A0h
Registros de Propósito General		Registros de Propósito General		Accesos a 20h-7Fh		Accesos a A0h - FFh	
96 Bytes		96 Bytes					
	7Fh		FFh		16Fh 170h		1EFh 1F0h
Banco 0		Banco 1		Banco 2		Banco 3	

Localizaciones de Memoria de Datos no implementada, se leen como '0'

* Registros no físicos

Notas:

- 1: Registros no implementados en el PIC16F873
- 2: Registros reservados, mantener estos registros en ceros.

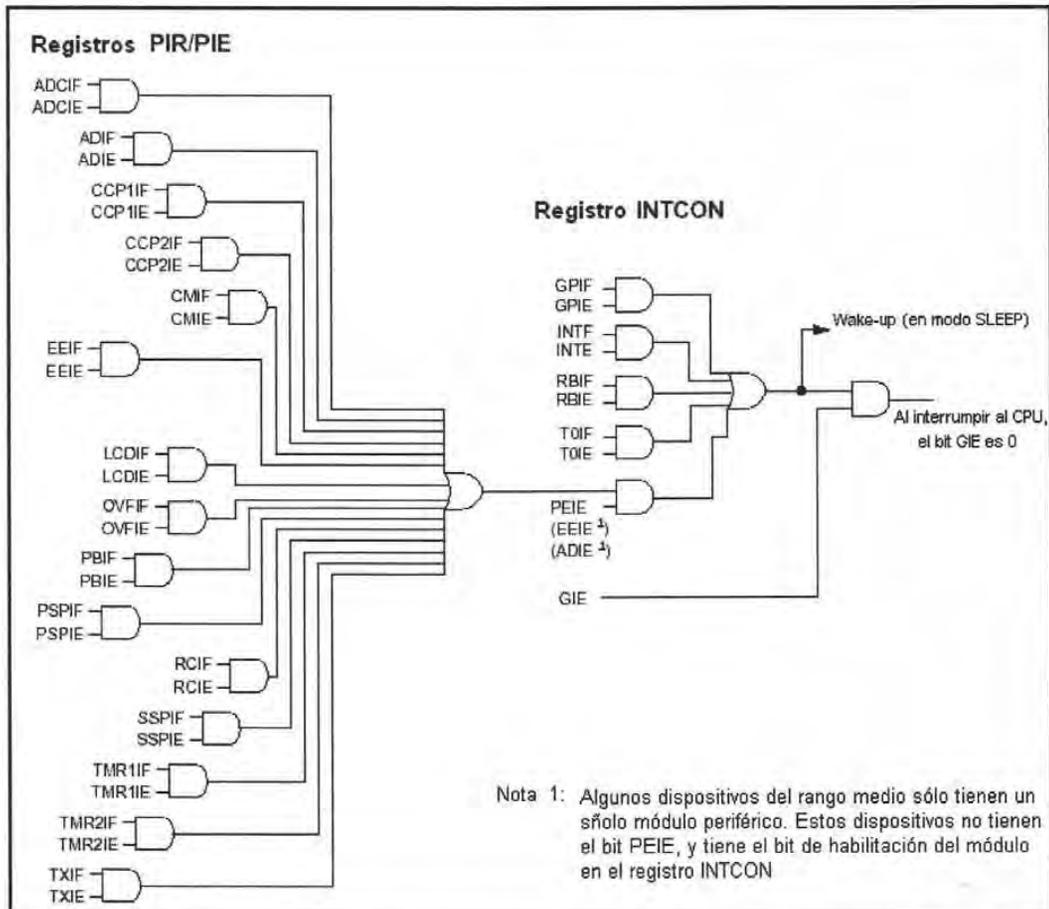
A4. Conjunto de Instrucciones del PIC16F87X

Nemónico, Operando	Descripción	Ciclos	Código de 14-Bits		Estado Afectado	Notas
			MSb	LSb		
Operaciones de Registros Orientadas a 1 Byte						
ADDWF	f,d	Sumar W y f	1	00 0111 dfff ffff	C,DC,Z	1,2
ANDWF	f,d	AND de W y f	1	00 0101 dfff ffff	Z	1,2
CLRF	f	Limpiar f	1	00 0001 1fff ffff	Z	2
CLRW	-	Limpiar W	1	00 0001 0xxx xxxx	Z	
COMF	f,d	Complemento de f	1	00 1001 dfff ffff	Z	1,2
DECF	f,d	Decrementar f	1	00 0011 dfff ffff	Z	1,2
DECFSZ	f,d	Decrementar f, saltar si es 0	1 (2)	00 1011 dfff ffff		1,2,3
INCF	f,d	Incrementar f	1	00 1010 dfff ffff	Z	1,2
INCFSZ	f,d	Incrementar f, saltar si es 0	1 (2)	00 1111 dfff ffff		1,2,3
IORWF	f,d	OR inclusivo de W con f	1	00 0100 dfff ffff	Z	1,2
MOVF	f,d	Mover f	1	00 1000 dfff ffff	Z	1,2
MOVWF	f,d	Mover W a f	1	00 0000 1fff ffff		
NOP	-	No operación	1	00 0000 0xx0 0000		
RLF	f,d	Rotar a la izquierda por el carry	1	00 1101 dfff ffff	C	1,2
RRF	f,d	Rotar a la derecha por el carry	1	00 1100 dfff ffff	C	1,2
SUBWF	f,d	Substraer W de f	1	00 0010 dfff ffff	C,DC,Z	1,2
SWAPF	f,d	Cambiar nibbles de f	1	00 1110 dfff ffff		1,2
XORWF	f,d	OR exclusivo de W y f	1	00 0110 dfff ffff	Z	1,2
Operaciones Orientadas a Bits de Registros						
BCF	f,b	Hacer 0 el bit b de f	1	01 00bb bfff ffff		1,2
BSF	f,b	Hacer 1 el bit b de f	1	01 01bb bfff ffff		1,2
BTFSC	f,b	Examinar bit de f, saltar si es 0	1 (2)	01 10bb bfff ffff		3
BTFSS	f,b	Examinar bit de f, saltar si es 1	1 (2)	01 11bb bfff ffff		3
Operaciones de Constantes y de Control						
ADDLW	k	Sumar a W la constante k	1	11 111x kkkk kkkk	C,DC,Z	
ANDLW	k	AND de W y constante k	1	11 1001 kkkk kkkk	Z	
CALL	k	Llamar a subrutina	2	10 0kkk kkkk kkkk		
CLRWDT	-	Limpiar el Watchdog Timer	1	00 0000 0110 0100	TO,PD	
GOTO	k	Saltar a la dirección	2	10 1kkk kkkk kkkk		
IORLW	k	OR inclusivo de W y constante k	1	11 1000 kkkk kkkk	Z	
MOVLW	k	Mover constante k a W	1	11 00xx kkkk kkkk		
RETFIE	-	Regresar de Interrupción	2	00 0000 0000 1001		
RETLW	k	Regresar la constante k en W	2	11 01xx kkkk kkkk		
RETURN	-	Regresar de subrutina	2	00 0000 0000 1000		
SLEEP	-	Ir al modo de espera	1	00 0000 0110 0011	TO,PD	
SUBLW	k	Substraer constante k de W	1	11 110x kkkk kkkk	C,DC,Z	
XORLW	k	OR exclusivo de W y constante k	1	11 1010 kkkk kkkk	Z	

Notas:

- 1: Cuando se modifica un registro de I/O como una función de el mismo (e.g. MOVF PORTB,1), el valor usado será el valor presente en los pines.
- 2: Si esta instrucción es ejecutada sobre el registro TMR0 (y, donde es aplicable, d=1), el preescalador será limpiado si es asignado al módulo del temporizador 0.
- 3: Si el Contador de Programa (PC) es modificado, o una condición de salto es verdadera, la instrucción requiere 2 ciclos. El segundo ciclo es ejecutado como un NOP.

A3. Lógica de los Bits de solicitud de Rutina de Interrupción



B. Rutinas Principales del Módulo Transceiver

B.1. Rutinas del Transmisor Inalámbrico Digital en RF

B.1.1. Rutina: Rutina_RX

```

; ***** Rutina_RX *****
; Se encarga de recibir los bytes del periférico UART, codificarlo y almacenarlo en RAM
; Después de completar el bloque de datos habilitar la transmisión del paquete de datos
Rutina_RX
    BANK0
    BCF    PIR1,RCIF        ; Limpiamos Bandera
    MOVF   RCREG,W         ; Leemos Dato y lo almacenamos
    MOVWF  BYTE_RX
    CALL   SumarCCR        ; Banco 0
    CALL   ConByte12       ; Banco 0

    DECFSZ CONTA_TP,F      ; Se Recibieron el paquete de datos?
    GOTO   Fin_RutRX      ; NO
; Codificamos el CCR
    MOVF   CCRL,W         ; SI
    MOVWF  BYTE_RX
    CALL   ConByte12       ; Banco 0
    MOVF   CCRH,W
    MOVWF  BYTE_RX
    CALL   ConByte12       ; Banco 0
; Hacemos una copia espejo del paquete de datos al Banco 1
    MOVLW  NNIBLES
    MOVWF  CONTA_TP
    MOVLW  FSRMAIN        ; Posición del paquete de bytes codificados a Tx
    MOVWF  FSR
MirrorDatos
    MOVF   INDF,W         ; leemos dato
    BSF    FSR,0x07
    MOVWF  INDF           ; en Banco 1
    BCF    FSR,0x07
    INCF   FSR,F
    DECFSZ CONTA_TP,F
    GOTO   MirrorDatos
; Reiniciamos Registros
    CLRF   CCRL           ; Reseteamos los bytes del CCR
    CLRF   CCRH
    MOVLW  FSRMAIN        ; Posición del paquete de bytes codificados a Tx
    MOVWF  FSR
    MOVLW  0x13           ; Valor Pre Datos, conservar balance DC
    MOVWF  INDF           ; en Banco 0
    INCF   FSR,F
    MOVLW  TPAQUETX      ; Inicializamos tamaño de datos del Paquete
    MOVWF  CONTA_TP
; Habilito Tx por RF
    BSF    _FPaque
Fin_RutRX
    GOTO   Ver_TSetPar

```

B.1.2. Rutina: ConByte12

```

.***** ConByte12 *****
;
; Convierte 1 byte a 12 bits codificados
; entrada: BYTE_RX y FSR           ; salida: NIB_L, NIB_H y ++ ++FSR
; t = 6 uS aprox.
ConByte12
    CALL    CarValNible1
    BANK0
    MOVWF  NIB_L
    SWAPF  BYTE_RX,F
    CALL    CarValNible1
    BANK0
    MOVWF  NIB_H
    MOVF   NIB_L,W           ; Almacenamos Nibles codificados en RAM
    MOVWF  INDF
    INCF   FSR,F
    MOVF   NIB_H,W
    MOVWF  INDF
    INCF   FSR,F           ; FSR guarda direccion de tabla de almacenamiento
    RETURN

```

```

.***** RUTINAS: CarValNible *****
;
; Recupera el Nible Codificado de la memoria de Programa
; Parte de la rutina de codificar 1 byte a 12 bits
CarValNible1
    BANK0
    CLRF   PCLATH
    MOVF   BYTE_RX,W
    ANDLW  0x0F
    ADDLW  TablaCod
    MOVWF  PCL
    RETURN
; No es necesario, pero lo usamos por sintaxis de rutinas

```

B.1.3. Rutina: TxMsg232

```

.***** TxMsg232 *****
;
; Transmite los mensajes de memoria del programa,
; usados en la Configuración del Transceiver y el Hypertérminoal
TxMsg232
    BANK0
LeerAsciiLetra
    CALL    LeerLetra
    MOVWF  LETRA_MSG
    ANDLW  0xFF           ; Probamos si la letra=0x00, Fin de Mensaje
    BTFSS  STATUS,Z
    GOTO   TxLetra
    GOTO   Fin_TxMsg232
TxLetra
    MOVF   LETRA_MSG,W
    MOVWF  TXREG
Reg_Vacio
    BANK1
    BTFSS  TXSTA,TRMT     ; Esperamos que Tx Byte
    GOTO   Reg_Vacio

```

```

    BANK0
    INCF    OFFSET_MSG,F
    GOTO    LeerAsciiLetra
Fin_TxMsg232
    RETURN

```

```

; ***** LeerLetra *****
LeerLetra
    BANK0
    MOVF    PCLATH_TMP,W
    MOVWF   PCLATH           ; Modificamos el PCLATCH
    MOVF    OFFSET_MSG,W
    MOVWF   PCL
    RETURN ; No es necesario, pero lo usamos por sintaxis de rutinas

```

B.1.4 Rutina: Set_ParametrosCOM

```

; ***** Set_ParametrosCOM *****
; Usando la Interface del HyperTerminal de Windows, cambiamos
; parametros de operacion del transceiver
Set_ParametrosCOM
    BANK0
    BSF     L_SPar           ; Led de Configuracion Activo
; Deshabilito Interrupción Global
    BCF     INTCON,GIE

; Deshabilito la Rx
    BANK0
    BCF     RCSTA,CREN      ; Deshabilito la Rx

; Habilito la Tx
    BANK1
    BSF     TXSTA,TXEN      ; Habilito la Tx

; Habilito la Rx
    BANK0
    BSF     RCSTA,CREN      ; Habilito la Rx

; ----- Menu 1 -----
MenuPrincipal
; Imprimir Menu 1
    CALL    Impri_Menu1

; Leemos opcion seleccionada:
Presiono_Tecla
    BANK0
    BTFSS   PIR1,RCIF       ; RCIF = 0? No se Presiono tecla?
    GOTO    Presiono_Tecla

; Leer Opcion Seleccionada
LeerTecla
    BCF     PIR1,RCIF       ; Limpiamos Bandera
    MOVF    RCREG,W         ; Leemos Dato y lo almacenamos
    MOVWF   BYTE_RX
; Comparamos con "1"

```

```

    MOVLW    0x31
    SUBWF    BYTE_RX,W
    BTFSC    STATUS,Z
    GOTO     Op1_Me1

; Comparamos con "2"
    MOVLW    0x32
    SUBWF    BYTE_RX,W
    BTFSC    STATUS,Z
    GOTO     Op2_Me1

; Comparamos con "3"
    MOVLW    0x33
    SUBWF    BYTE_RX,W
    BTFSC    STATUS,Z
    GOTO     Op3_Me1
    GOTO     Presiono_Tecla

;----- Opcion 1 Menu 1 -----
; Rutina que ejecuta al seleccionar la Opcion 1 del Menu 1
Op1_Me1
; Imprimir Menu 2
    CALL     Impri_Menu2

; Leemos opcion seleccionada:
Presiono_Tecla1
    BANK0
    BTFSS    PIR1,RCIF          ; RCIF = 0? No se Presiono tecla?
    GOTO     Presiono_Tecla1

; Leer Tecla
LeerTecla2
    BANK0
    BCF     PIR1,RCIF          ; Limpiamos Bandera
    MOVF    RCREG,W           ; Leemos Dato y lo almacenamos
    MOVWF   BYTE_RX

; Comparamos con "1"
    MOVLW    0x31
    SUBWF    BYTE_RX,W
    BTFSC    STATUS,Z
    GOTO     Op1_Me2

; Comparamos con "2"
    MOVLW    0x32
    SUBWF    BYTE_RX,W
    BTFSC    STATUS,Z
    GOTO     Op2_Me2

; Comparamos con "3"
    MOVLW    0x33
    SUBWF    BYTE_RX,W
    BTFSC    STATUS,Z
    GOTO     Op3_Me2
; Comparamos con "4"
    MOVLW    0x34

```

```

SUBWF  BYTE_RX,W
BTFSC  STATUS,Z
GOTO   Op4_Me2

```

; Comparamos con "5"

```

MOVLW  0x35
SUBWF  BYTE_RX,W
BTFSC  STATUS,Z
GOTO   Op5_Me2

```

```

GOTO   Presiono_Tecla1

```

```

CLRF   TEMP_2

```

O_Retar

```

CALL   Wait
DECFSZ TEMP_2,F
GOTO   O_Retar
GOTO   Pprincipal

```

```

;-----
; Rutina que ejecuta al seleccionar la Opcion 1 del Menu 2
Op1_Me2

```

```

BANK0
MOVLW  b1200
MOVWF  DATOSE           ; Fijamos a velocidad de "1200 bps"
GOTO   FIN_MENU2

```

; Rutina que ejecuta al seleccionar la Opcion 2 del Menu 2

Op2_Me2

```

BANK0
MOVLW  b2400
MOVWF  DATOSE           ; Fijamos a velocidad de "2400 bps"
GOTO   FIN_MENU2

```

; Rutina que ejecuta al seleccionar la Opcion 3 del Menu 2

Op3_Me2

```

BANK0
MOVLW  b4800
MOVWF  DATOSE           ; Fijamos a velocidad de "4800 bps"
GOTO   FIN_MENU2

```

; Rutina que ejecuta al seleccionar la Opcion 4 del Menu 2

Op4_Me2

```

BANK0
MOVLW  b9600
MOVWF  DATOSE           ; Fijamos a velocidad de "9600 bps"
GOTO   FIN_MENU2

```

; Rutina que ejecuta al seleccionar la Opcion 5 del Menu 2

Op5_Me2

```

GOTO   MenuPrincipal
GOTO   FIN_MENU2

```

FIN_MENU2

```

BANK0
MOVLW  0x10

```

```

MOVWF  DIRECC
CALL   Escribir_EEPROM
BANK0
INCF   DIRECC,F
CALL   Escribir_EEPROM
GOTO   Op1_Me1

; ---- Opcion 2 Menu 1 -----
; Rutina que ejecuta al seleccionar la Opcion 1 del Menu 1
Op2_Me1
; Imprimir Menu 3
CALL   Impri_Menu3

; Leemos opcion seleccionada:
Presiono_Tecla2
BANK0
BTFS   PIR1,RCIF      ; RCIF = 0? No se Presiono tecla?
GOTO   Presiono_Tecla2
; Leer Tecla
LeerTecla3
BANK0
BCF    PIR1,RCIF      ; Limpiamos Bandera
MOVF   RCREG,W        ; Leemos Dato y lo almacenamos
MOVWF  BYTE_RX

; Comparamos con tecla "ESC"
MOVLW  0x20
SUBWF  BYTE_RX,W
BTFS   STATUS,Z
GOTO   MenuPrincipal
GOTO   Presiono_Tecla2

; ---- Opcion 3 Menu 1 -----
; Rutina que ejecuta al seleccionar la Opcion 1 del Menu 1
Op3_Me1
CALL   Impri_Menu4      ; Mensaje de Fin de Configuracion
GOTO   FIN_MENUS

FIN_MENUS
; Deshabilito la Tx
BANK1
BCF    TXSTA,TXEN      ; Deshabilito la Tx
BANK0
BCF    RCSTA,CREN      ; Deshabilito la Rx
MOVLW  0x00
MOVWF  PCLATH
BANK0
BCF    L_SPar          ; Led de Configuracion Activo
GOTO   Pprincipal

```

B.1.5. Rutina: RutinaInterrupcion

```

***** RutinaInterrupcion *****
; Realiza la transmisión del tren de Pulsos y los Datos Codificados
; usa un tiempo de bit de aproximadamente 52 useg
RutinaInterrupcion
; Almacenamos registros W y STATUS
    BANK0
    MOVWF    W_TEMP
    SWAPF   STATUS,W
    MOVWF   STATUS_TEMP
    SWAPF   FSR,W
    MOVWF   FSR_TMP
    MOVF    FSR_INTER,W
    MOVWF   FSR

    BTFSC   INTCON,T0IF      ; ISR de TMR0 se disparó?
    GOTO    ISR_TMR0        ; Si
    GOTO    FIN_ISR         ; No
;-----
ISR_TMR0
    BANK0
    BCF     INTCON,T0IF     ; Limpiamos Flag ISR TRM0
    MOVLW  0x8D             ; (FF-C7+1)*2*0.2E-6 + 3.2 uS
    MOVWF  TMR0            ; 12 ciclos

    BTFSC  _FlnITx         ; Saltar si _FlnITx=0?
    GOTO   Trans_PaqueRF

    BTFSC  _FPaquete       ; Bandera de Paquete listo = 0?
    GOTO   Trans_1erBit

Seguir_RTMR0
;Generamos Tren de Pulsos
    BTFSC  _tx             ; Saltar si tx=0
    GOTO   Hacer0tx
Hacer1tx
    NOP                    ;igualar tiempo de bits a 51.6 uSeg
    BSF    _tx
    GOTO   SaltoSRT0
Hacer0tx
    BCF    _tx
SaltoSRT0
    GOTO   FIN_ISR

Trans_1erBit
; Tx el 1er Bit del paquete de datos
    BANK0
    BTFSS  _tx             ; Saltar si _tx = 1
    GOTO   Seguir_RTMR0
    BSF    _tx
    BSF    _FlnITx        ; Hacemos esta bandera = 1
    MOVF   INDF,W
    MOVWF  BYTE_TX
    BCF    BTxRF
    RRF    BYTE_TX,F
    BTFSC  STATUS,C

```

```

    BSF      BTxRF
    BCF      _FPaque      ; Deshabilitamos Tx 1er Bit
Fin_Tx1erBit
    GOTO     FIN_ISR

Trans_PaqueRF      ; Estamos listos para Tx el paquete por RF
    BANK0
    BTFSS    BTxRF      ; Tx bit BTxRF al pin_tx
    GOTO     TxRF0
    BSF      _tx
    GOTO     SaltoTxRF1
TxRF0
    BCF      _tx
SaltoTxRF1
; Calculos del sgte BTxRF
    BCF      BTxRF
    RRF      BYTE_TX,F
    BTFSC    STATUS,C
    BSF      BTxRF
    DECFSZ   BITS_TX,F
    GOTO     FIN_ISR
; Vamos por el sgte byte
    MOVLW   VBITSTX      ; why ya se calculo su 1er bit separadamente
    MOVWF   BITS_TX
    INCF    FSR,F        ; el sgte byte a Tx
    MOVF    INDF,W
    MOVWF   BYTE_TX
    BCF     BTxRF        ; Evaluamos valor del bit
    RRF     BYTE_TX,F
    BTFSC   STATUS,C
    BSF     BTxRF
    DECFSZ  NNIBLES_TX,F
    GOTO    FIN_ISR
; Reiniciamos Banderas y Registros
    BCF     _FinITx      ; Se termino de Tx Paquete
    BCF     _FPaque
    MOVLW   VFSRINT      ; Cargamos valor de FSR usado en Interrupcion
    MOVWF   FSR
    MOVLW   NNIBLES
    MOVWF   NNIBLES_TX
    GOTO    FIN_ISR
; -----
FIN_ISR
    BANK0
    MOVF    FSR,W
    MOVWF   FSR_INTER    ; Guardamos FSR de Interrupcion
    SWAPF   FSR_TMP,W
    MOVWF   FSR
    SWAPF   STATUS_TEMP,W
    MOVWF   STATUS
    SWAPF   W_TEMP,F
    SWAPF   W_TEMP,W
    RETFIE

```

B.2. Rutinas del Receptor Inalámbrico Digital en RF

B.2.1. Rutina: Decodificar

```

;***** Decodificar *****
; Decodifica los datos Codificados recibidos por RF
Decodificar
    BANK0
    BCF     T1_Deco           ; Deshabilito Rutina Decodificar
    MOVF   FSR,W
    MOVWF  FSR_DECO
    MOVLW  VNN_DECO
    MOVWF  NN_DECO
    MOVLW  VFSR_DATOS       ; Puntero de FSR de Datos Decodificados
    MOVWF  FSR_DATOS
    MOVLW  VFSRINT+1
    MOVWF  FSR

OtraDecodi
    MOVF   INDF,W
    MOVWF  DECO_TMP
    CALL   ConvertirNible
    MOVF   DECO_TMP,W
    MOVWF  REGDECO

    INCF   FSR,F
    MOVF   INDF,W
    MOVWF  DECO_TMP
    CALL   ConvertirNible

    SWAPF  DECO_TMP,W
    ADDWF  REGDECO,F
    INCF   FSR,F

; Grabamos en RAM dato Decodificado
GrabamosRAM
    MOVF   FSR,W
    MOVWF  FSR_TEMP
    MOVF   FSR_DATOS,W
    MOVWF  FSR
    MOVF   REGDECO,W
    MOVWF  INDF

    INCF   FSR_DATOS,F
    MOVF   FSR_TEMP,W
    MOVWF  FSR

; Terminamos de Decodificar?
    DECFSZ NN_DECO,F
    GOTO   OtraDecodi
    GOTO   Fin_RDecodi

Ver_AHORA
; Sumamos datos para formar el CCR
    MOVLW  VFSR_DATOS
    MOVWF  FSR

```

```

        MOVLW    0x08
        MOVWF   NN_DECO
Otra_SCCR
        MOVF    INDF,W
        CALL   SumarCCR
        INCF   FSR,F
        DECFSZ NN_DECO,F
        GOTO   Otra_SCCR
        GOTO   Fin_RDecodi

; Comprobamos CCR Transmitido con el Recibido
        MOVLW   VFSR_DATOS+0x08
        MOVWF   FSR
        MOVF    INDF,W
        XORWF   CCRL,W           ; Comprobamos CCRL, son iguales?
        BTFSS  STATUS,Z         ; Si
        GOTO   Fin_NoDato       ; No
        GOTO   CCRcorrectos

        INCF   FSR,F
        MOVF    INDF,W
        SUBWF   CCRH,W           ; Comprobamos CCRH, son iguales?
        BTFSS  STATUS,Z         ; Si
        GOTO   Fin_NoDato       ; No

CCRcorrectos
        BSF    T2_VLCD           ; Habilitamos Rutina de Visualizar Datos Recibidos
        GOTO   Fin_RDecodi

Fin_NoDato
        BCF    T2_VLCD           ; Deshabilito Rutina de Visualizar Datos Recibidos

Fin_RDecodi
        BSF    T3_RS232         ; Habilito la Rutina de Transmision Serial
;        BSF    T2_VLCD         ; Habilitamos Rutina de Visualizar Datos Recibidos
        MOVF   FSR_DECO,W       ; Regresamos valor FSR de lazo principal
        MOVWF  FSR
        GOTO   Ver_T2
    
```

B.2.2. Rutina: VerDatosLCD

```

; ***** VerDatosLCD *****
; Tarea 2
VerDatosLCD
        BANK0
        MOVF   FSR,W
        MOVWF  FSR_LCD         ; Grabamos FSR de contexto

        MOVLW  0x80           ; Direccionamos a la 1ra linea del LCD
        CALL  I_LCD

VerLCD_1
        MOVLW  0x02
        MOVWF  CNT_LCD
        MOVLW  VFSR_DATOS
        MOVWF  FSR
    
```

```

SgVDLCD1
  MOVF    INDF,W
  MOVWF   DATOSLCD
  CALL    Byte_en_LCD
  INCF    FSR,F           ; Incrementamos puntero de datos
  DECFSZ  CNT_LCD,F
  GOTO    SgVDLCD1
  GOTO    Fin_VerDLCD

SgVDLCD2
  MOVF    INDF,W
  MOVWF   DATOSLCD
  CALL    Byte_en_LCD
  INCF    FSR,F           ; Incrementamos puntero de datos
  DECFSZ  CNT_LCD,F
  GOTO    SgVDLCD2

Fin_VerDLCD
; Valores de Fin de Tarea
  MOVF    FSR_LCD,W      ; Regresamos valor FSR de lazo principal
  MOVWF   FSR
  BCF     T2_VLCD        ; Desactivo Rutina VerDatosLCD
  BSF     T3_RS232       ; Activo Rutina Tx_RS232
  GOTO    Ver_T3
    
```

B.2.2. Rutina: Tx_RS232

```

; ***** TX_RS232 *****
; Transmitir serialmente Datos recibidos
; Tarea 3
Tx_RS232
; Habilitamos la Tx
  BANK1
  BSF     TXSTA,TXEN

  BANK0
  BCF     T3_RS232       ; Desactivo Rutina Tx_RS232
  MOVF    FSR,W
  MOVWF   FSR_TX232     ; Grabamos FSR de contexto
  MOVLW   VFSR_DATOS
  MOVWF   FSR

  MOVLW   0x02
  MOVWF   CNT_LCD

OtraTx1
  BANK0
  MOVF    INDF,W
  MOVWF   TXREG
  INCF    FSR,F

Se_TxByte
  BANK1
  BTFSS   TXSTA,TRMT
  GOTO    Se_TxByte
  BANK0
    
```

```

DECFSZ  CNT_LCD,F
GOTO    OtraTx1
MOVWF   FSR_TX232,W
MOVWF   FSR           ; Regresamos valor FSR de contexto
GOTO    Ver_T4

```

B.2.4. Rutina: ConvertirNibble

```

.***** ConvertirNibble *****
; Rutina que identifica el nibble Tx y lo convierte en nibble
; menos significativo, Entrada y salida: DECO_TMP
ConvertirNibble
    BANK0
    MOVLW 0x15
    XORWF DECO_TMP,W
    BTFSC STATUS,Z
    GOTO  FijarReg1

    MOVLW 0x1C
    XORWF DECO_TMP,W
    BTFSC STATUS,Z
    GOTO  FijarReg2

    MOVLW 0x0F
    ANDWF DECO_TMP,F
    GOTO  Fin_CN

FijarReg2
    MOVLW 0x0F
    MOVWF DECO_TMP
    GOTO  Fin_CN

FijarReg1
    CLRF  DECO_TMP
    GOTO  Fin_CN

Fin_CN
    RETURN

```

B.2.5. Rutina: RutinaInterrupcion

```

.***** RutinaInterrupcion *****
; Definicion de Rutinas de Interrupcion
RutinaInterrupcion
; Almacenamos registros W y STATUS
    BANK0
    MOVWF W_TEMP
    SWAPF STATUS,W
    MOVWF STATUS_TEMP
    SWAPF FSR,W
    MOVWF FSR_TEMP
    MOVF  FSR_INTER,W
    MOVWF FSR

    BTFSS INTCON,INTE ; Saltar si INTE = 1?

```

```

GOTO    PREG_IRS_TMR0    ; Si
                                ; No

BTFSC   INTCON,INTF      ; ISR de INT se disparó?
GOTO    ISR_INT0         ; Si
GOTO    FIN_ISR          ; No

PREG_IRS_TMR0
BTFSS   INTCON,T0IE      ; Saltar si T0IE = 1
GOTO    FIN_ISR          ;

BTFSC   INTCON,T0IF      ; ISR de TMR0 se disparó?
GOTO    ISR_TMR0         ; Si
GOTO    FIN_ISR          ; No

ISR_INT0
BANK0
BCF     INTCON,INTF      ; Limpiamos Bandera ISR de INT0
BCF     INTCON,INTE      ; Deshabilito ISR de INT

MOVLW   0xC6              ; activacion de ISR del TMR0 a 26 us
MOVWF   TMR0
BCF     INTCON,T0IF      ; Limpiamos Flag ISR TRM0
BSF     INTCON,T0IE      ; Habilito ISR del TMR0

BSF     F1isrT0          ; F1isrT0 = 0, asegurarnos el flujo de control
MOVLW   0x02
MOVWF   CON_PRED        ; Leer dos "1", dispara lectura de bloque de datos

GOTO    FIN_ISR

ISR_TMR0
BANK0
BCF     INTCON,T0IF      ; Limpiamos Flag ISR TRM0
MOVLW   0x8D              ; (FF-C7+1)*2*0.2E-6 + 3.2 uS
MOVWF   TMR0

BTFSC   F2isrT0
GOTO    Leer_Datos       ; Leemos el paquete de datos

BTFSC   F1isrT0
GOTO    Leer_1erBit      ; Leemos el primer bit de los datos
GOTO    FIN_ISR

Leer_1erBit
BTFSS   RxDa              ; RxDa = 1?
GOTO    FinL1erBit       ; No
GOTO    Seguir_LDatos    ; Si

Seguir_LDatos
DECFSZ  CON_PRED,F
GOTO    FIN_ISR
BCF     F1isrT0          ; Deshabilitamos la Lectura del 1er bit
BSF     F2isrT0          ; Habilitamos la Lectura del paquete de datos
CLRF    REGRX            ; Limpiamos reg de dato recibido
MOVLW   VCON_PRED
MOVWF   CON_PRED        ; Numero de bits a leer

```

```

    MOVLW   VNNIBLES
    MOVWF   NNIBLES           ; Numero de Nibles a leer
    GOTO    FIN_ISR

FinL1erBit
    BCF     F1isrT0           ; Deshabilitamos la Lectura del 1er bit
    BCF     INTCON,T0IE       ; Deshabilito ISR del TMR0
    BCF     INTCON,INTF       ; Limpiamos Bandera ISR de INT0
    BSF     INTCON,INTE       ; Habilito ISR de INT
    GOTO    FIN_ISR

Leer_Datos
    BCF     STATUS,C          ; Hacemos C=0
    BTFSC   RxDa              ; Saltar si RxDa=0
    BSF     STATUS,C          ; Hacemos C=1
    RRF     REGRX,F

; Decrementamos bits leidos
    DECFSZ  CON_PRED,F
    GOTO    FIN_ISR

; Insertamos ceros en registro recibido
    BCF     STATUS,C          ; Hacemos C=0
    RRF     REGRX,F
    BCF     STATUS,C          ; Hacemos C=0
    RRF     REGRX,F

; Comparamos Cabecera
    MOVLW   VFSRINT
    SUBWF   FSR_INTER,W      ; Verificamos si direccionamos al 1er Byte
    BTFSS   STATUS,Z
    GOTO    GrabarDatoRam
    MOVLW   CABECERA
    XORWF   REGRX,W          ; Comparamos con Cabecera
    BTFSS   STATUS,Z
    GOTO    Fin_TMR0_1
    GOTO    GrabarDatoRam

GrabarDatoRam
; Grabamos byte recibido en RAM
    MOVF    REGRX,W
    MOVWF   INDF
    INCF    FSR,F
;    BSF     F3isrT0           ; Habilito Tarea de comparar cabecera
; Reiniciamos registros relacionados a la lectura de 6 bits
    CLRF    REGRX           ; Limpiamos reg de dato recibido
    MOVLW   VCON_PRED
    MOVWF   CON_PRED       ; Numero de bits a leer

; Decrementamos el numero de Nibles a Leer
    DECFSZ  NNIBLES,F
    GOTO    FIN_ISR

; Reiniciamos Banderas
    BSF     T1_Deco          ; Activo Tarea de Decodificación
    MOVLW   VFSRINT
    MOVWF   FSR             ; Cargamos valor de FSR usado en Interrupcion
                                ; que luego se carga en el registro FSR_INTER

```

```

Fin_TMR0_1
  BCF   INTCON,T0IE      ; Deshabilita ISR de TRM0
  BCF   INTCON,INTF      ; Limpio Flag IRS de INT
  BSF   INTCON,INTE      ; Habilitamos IRS de INT
  BCF   F1isrT0           ; Deshabilito Banderas del ISR del TMR0
  BCF   F2isrT0
  GOTO  FIN_ISR
  
```

```

FIN_ISR
  BANK0
  MOVF  FSR,W
  MOVWF FSR_INTER        ; Guardamos FSR de Interrupcion
  SWAPF FSR_TEMP,W
  MOVWF FSR
  SWAPF STATUS_TEMP,W
  MOVWF STATUS
  SWAPF W_TEMP,F
  SWAPF W_TEMP,W
  RETFIE
  
```

BIBLIOGRAFIA

Bibliografía

- [1] El Sistema de Información Sismotelemétrica de México (SISMEX), Avances hasta 1987
Miguel Torres Noguez y Enrique Mena Sandoval
Proyecto 6746, Instituto de Ingeniería – UNAM, 1988
- [2] Red Acelerográfica de la Ciudad de México
http://www.cires.org.mx/es_racm_main.php4
- [3] Yokohama Co. Ltd.
<http://www.yokoyama-mfg.co.jp>
- [4] ChipCon AS
<http://www.chipcon.com>
- [5] ABACOM Technologies
<http://www.abacom-tech.com>
- [6] Toshiba
<http://www.toshiba.com>
- [7] RF Monolithic, Inc.
<http://www.rfm.com>
- [8] ASH Transceiver Designer's Guide
Updated 2000.01.17
RFM RF Monolithics, Inc.
- [9] ASH Transceiver *Software* Designer's Guide
Working Draft - 2000.08.20
RFM RF Monolithics, Inc.

- [10]** PIC16F87X Data Sheet
28/40-Pin 8-Bit CMOS FLASH Microcontroller
DS30292C, 216 páginas.
2001, Microchip Technology Incorporated, impreso en U.S.A.

- [11]** MPASM USER'S GUIDE
With MPLINK and MPLIB
Microchip Technology Incorporated. DS33014G. USA 1999.

- [12]** MPLAB
IDE, SIMULATOR, EDITOR – USER'S GUIDE
Microchip Technology Incorporated. DS51025D. USA 2000.

- [13]** Motorola RF Monolithic Integrated Circuits
Motorola, Inc. 1997. SG381/D

- [14]** Transmisión Digital
Dogan & Osman
Segunda Edición. McGraw-Hill Book Company.
United Status 1989

- [15]** Diseño y Realización de un Registrador Sismológico Digital Uniaxial
Walter Kemper Castro. Tesis de Maestría. UNAM 2003.

- [16]** Introducción a los Sistema de Comunicación
Ferrel G. Stremier
Tercera Edición, 1993. Traducción de Gonzalo Isaac Duchén Sánchez
Addison-Wesley Iberoamericana, S.A. Estado Unidos