

UNIVERSIDAD NACIONAL DE TUCUMÁN
FACULTAD DE CIENCIAS EXACTAS Y TECNOLOGÍA
DEPARTAMENTO DE ELECTRICIDAD, ELECTRÓNICA
Y COMPUTACIÓN

Trabajo de Graduación, Informe final:

**“SINTETIZADOR DE FRECUENCIAS PARA
MICROONDAS EN BANDA KU (10,6 A 11,8 [GHZ])”**

Autor:

CANGEMI, José Ignacio

CX04-1904-6

Ingeniería Electrónica, Plan 1991 (Mod. 2004)

Tutor:

Ing. ARIAS, Néstor E.

Fecha de presentación: Año 2009, diciembre

Agradecimientos

Éste trabajo es el resultado de años de trabajo y esfuerzo personal, pero que no hubiese sido posible sin el apoyo incondicional de muchos que quizás sin saberlo son parte de esto. Por eso quiero agradecer a:

Ingeniero Néstor Arias, mi tutor, profesor y mentor, por el apoyo y motivación a realizar éste proyecto con acierto.

Al ingeniero Juan Ise, que siempre estuvo presente para brindarme una mano y consejo.

A la empresa DIGICOM y en particular al ingeniero Marcelo Mitre que siempre me abrió las puertas de su empresa, y a Ramón Cardoso, por su ayuda desinteresada en el armado de uno de los circuitos de éste proyecto, ya que sin su gran experiencia no se podría haber concretado exitosamente.

A mi familia, por brindarme la oportunidad de tener una educación universitaria, que es un corolario de las enseñanzas recibidas con amor a lo largo de toda mi vida.

A Romina, mi compañera, que siempre está a mi lado, por su apoyo incondicional en la realización de éste y muchos otros proyectos que hemos imaginado juntos.

A todos mis amigos y compañeros, porque siempre estuvieron ahí.

Y finalmente, a Canal 8 de Tucumán, que es desde hace cuatro años una segunda casa en la cual he crecido profesional y humanamente, de la mano de mis compañeros de trabajo y de mi jefe y amigo el ingeniero Gastón Acosta, por su gran generosidad.

Índice

Sintetizador de frecuencias para microondas en banda Ku(10,6 a 11,8 [Ghz]).....	1
1 Descripción general del proyecto.....	1
1.1 Objetivo del proyecto.....	1
1.2 Posibles soluciones.....	2
1.3 Solución adoptada.....	2
1.4 Esquema del PLL.....	4
1.5 Descripción de las partes del equipo.....	5
1.5.1 Sintetizador.....	5
1.5.2 Oscilador.....	9
1.5.3 Microcontrolador.....	11
2 Circuito esquemático.....	12
2.1 Sintetizador y VCO.....	12
2.2 Microcontrolador y display.....	13
3 Cálculos.....	14
3.1 Filtro de lazo del PLL.....	14
3.2 Potencia de entrada al sintetizador.....	17
3.3 Cálculo de líneas de transmisión y adaptación de impedancia de entrada.....	18
3.3.1 Diseño de la línea de transmisión en la banda de 10,6 a 11,8 [Ghz].....	18
3.3.2 Diseño de la línea de transmisión en la banda de 2,65 a 2,95 [Ghz].....	21
3.4 Adaptación de impedancia de entrada en el sintetizador	22
3.5 Adaptación de impedancia en el diseño actual.....	23
4 Programación del microcontrolador.....	24
4.1 Inicialización del protocolo I ² C en el microcontrolador.....	24
4.1.1 Configurar los pines 3 y 4 del PORTC como entrada.....	24
4.1.2 Configurar en modo Master.....	24
4.1.3 Configurar el bitrate del protocolo.....	25
4.1.4 Desactivar slew-rate.....	25
4.2 Lectura y escritura del bus I ² C.....	25
4.2.1 Acknowledge.....	25
5 Mediciones.....	26
5.1 Medición con el banco de microondas.....	26
5.2 Medición del espectro de frecuencias.....	26
6 Conclusiones.....	28
ANEXO – Código fuente del programa para el microcontrolador.....	30

Sintetizador de frecuencias para microondas en banda Ku(10,6 a 11,8 [Ghz])

1 Descripción general del proyecto

1.1 Objetivo del proyecto

El siguiente proyecto se realizó para cubrir principalmente las necesidades del Laboratorio de Telecomunicaciones de la Facultad de Ciencias Exactas de la Universidad Nacional de Tucumán de disponer de un generador de microondas en la banda X, **es decir de 8 a 12 [Ghz]**, sintetizado y estable en frecuencia para utilizarlo tanto en investigación como en docencia.

El proyecto de tener un oscilador en microondas programable, surgió como una necesidad de tener una fuente estable y de frecuencia conocida, ya que para desarrollar una línea de trabajos en microondas se necesita disponer de generación de éstas señales. A medida que avanzó la investigación de cómo desarrollar éste proyecto surgieron diversas líneas de desarrollo a seguir, como así también las dificultades asociadas a trabajar a tan alta frecuencia, como por ejemplo conseguir materiales dieléctricos necesarios para los circuitos impresos, o la disponibilidad de instrumental.

Se pudo conseguir buenos resultados con los instrumentos disponibles en el laboratorio, y se dejan sentadas las bases para continuar realizando desarrollos relacionados con éste proyecto, como por ejemplo la modulación de señales en esta banda de frecuencias y trabajos en radar doppler.

Para seguir el estudio y desarrollo de éste proyecto se ha consultado diferentes fuentes bibliográficas que tratan diversos temas, como la teoría general de Phase Lock Loop (PLL), referencia de programación de lenguaje ANSI C, notas de aplicación de sintetizadores de frecuencia y especificaciones de protocolos de comunicación. Las referencias a ellas se encuentran citadas al final de éste informe.

1.2 Posibles soluciones

En un comienzo se analizó y se ensayó las diferentes maneras de alcanzar el objetivo del proyecto, las diferentes tecnologías disponibles en el país y las posibilidades de importar algún componente desde el exterior. Las soluciones analizadas fueron:

- Usar un oscilador controlado por tensión en la banda de 1 [Ghz] a 1,3 [Ghz] (se probó con un VCO Motorola MC12149) aportado al Laboratorio por la empresa DIGICOM SRL, con multiplicadores contruidos con transistores JFET y filtros de salida en cada etapa.
- Utilizar un oscilador controlado por tensión en la banda de 2 [Ghz] de Minicircuits JTOS-2000A disponible en el laboratorio y multiplicadores no lineales a la salida.
- Importar un oscilador controlado por tensión en la banda de 10,6[Ghz] a 11,8 [Ghz], con salidas en $F_o/2$ y $F_o/4$ de la empresa Hittite, el HMC534LP5.

En todos los casos se ha utilizado para controlar el VCO y formar un PLL completo un sintetizador de frecuencias SP5769 de Zarlink, el cual trabaja con frecuencias de hasta 3[Ghz] y la configuración del mismo se realiza a través de un puerto I²C integrado.

1.3 Solución adoptada

La línea con la que se eligió continuar fue fue la última, es decir importar un VCO HMC534LP5 de Hittite, y aprovechar su salida en $F_o/4$, lo que da un rango de frecuencias de 2,65[Ghz] a 2,95[Ghz] en la entrada del sintetizador SP5769.

El equipo completo está compuesto por un microcontrolador, que envía la configuración necesaria al sintetizador y colecta datos de estado del mismo por el bus I²C y muestra información sobre un display, y un PLL que esta integrado por un sintetizador de frecuencias y un oscilador controlado por tensión.

El microcontrolador lee la entrada de dos pulsadores y con ésta información selecciona la frecuencia que va a programar sobre los divisores del sintetizador, y luego éste lee el estado del mismo y cuando se enclavado en frecuencia muestra un mensaje de enclavamiento y la frecuencia seleccionada. Por el contrario si hubo enclavamiento en frecuencia muestra un mensaje de error.

Para realizar las mediciones de frecuencia se ha utilizado el banco de microondas que dispone el Laboratorio, preparado para trabajar entre 8 y 12 [Ghz] con guías de onda, un diodo detector y un microamperímetro. Además para observar el espectro de frecuencias generadas, se utilizó un analizador de espectro que alcanza los 2,7[Ghz] en conjunto con un LNB de banda Ku que tiene un oscilador local de 9,75 [Ghz] y convierte el espectro al rango de 950 [Mhz] a 2150[Mhz].

Cabe destacar, que la sección del microcontrolador, es decir el display, las teclas de selección y el microcontrolador, pueden ser reemplazados a futuro por otra configuración diferente, con otra tecnología o simplemente por una PC que puedan correr el protocolo I²C sobre un puerto externo.

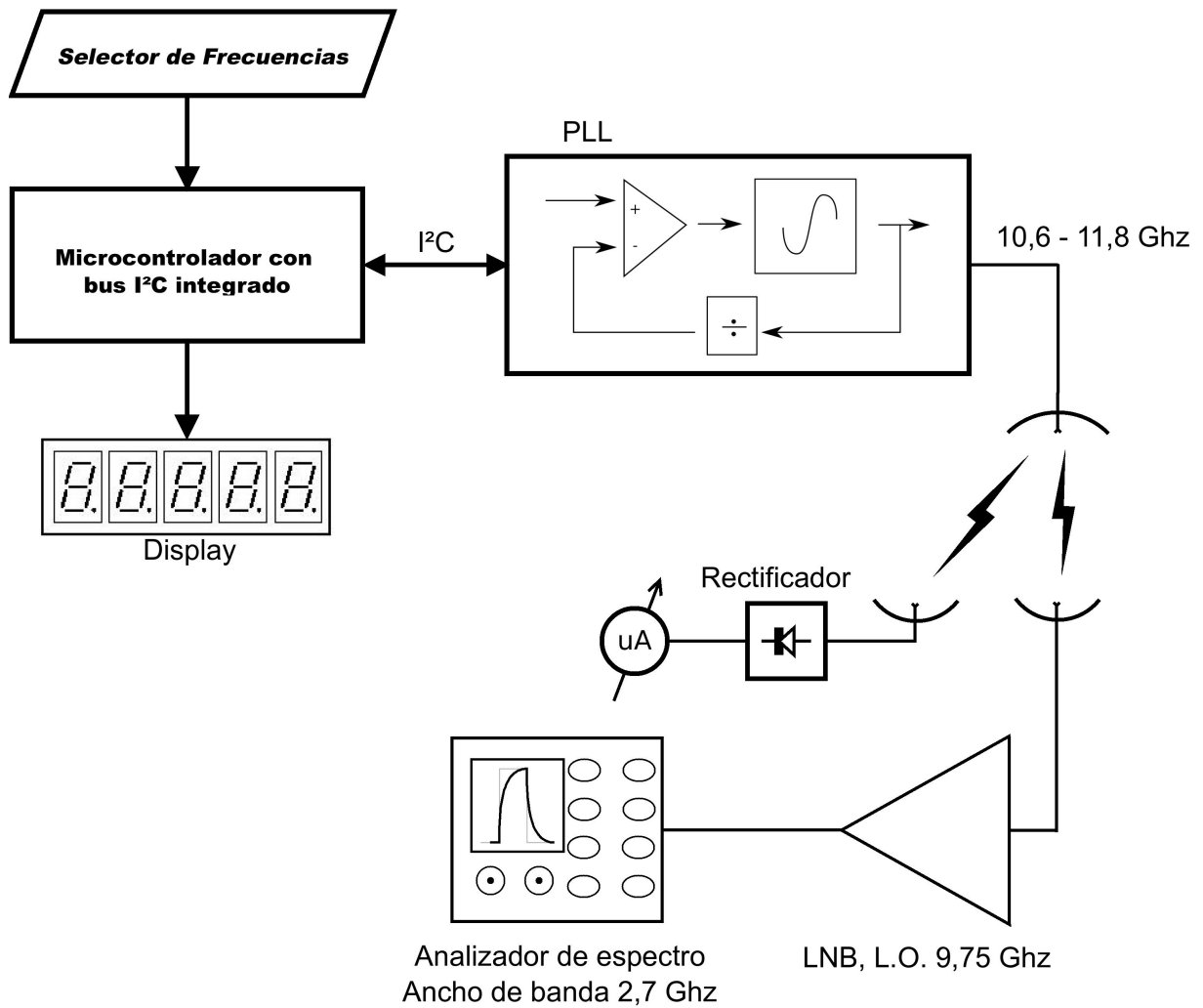


Fig. 1: Esquema general del equipo completo

1.4 Esquema del PLL

El circuito principal de éste proyecto es la sección de RF, es decir el PLL. Para la mejor comprensión del funcionamiento de éste exponemos un diagrama esquemático general sobre el cual trabajaremos en adelante.

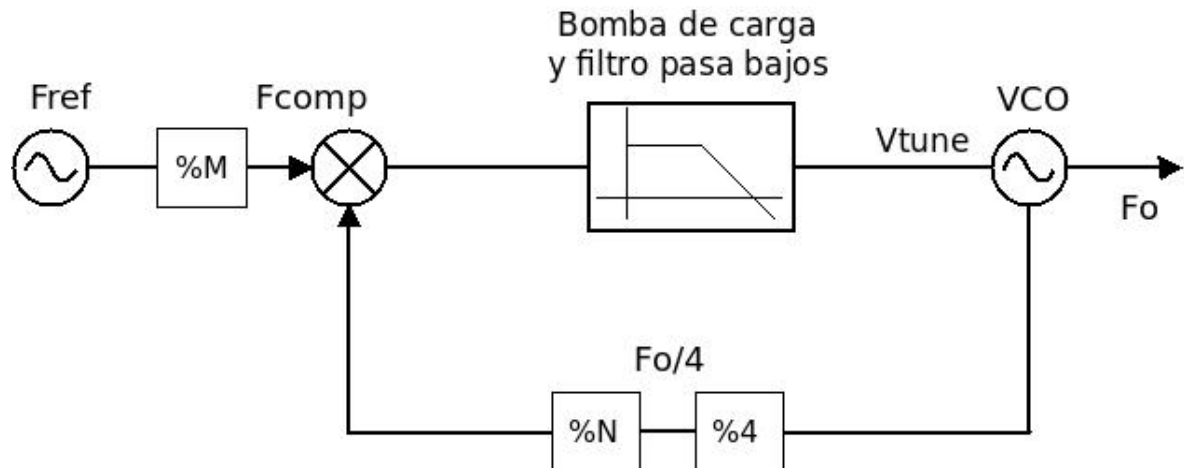


Fig. 2: Esquema general del PLL

El PLL está conformado por un oscilador a frecuencia F_{ref} , ésta frecuencia se divide por un divisor programable por M y se obtiene la frecuencia de comparación F_{comp} , que es comparada en frecuencia y fase con la salida del VCO dividida en un divisor por 4 y otro programable por N. La salida del comparador son pulsos de corriente proporcionales al error de comparación, que son integrados por el filtro del lazo para luego entrar sobre la tensión de control del VCO.

La impedancia de entrada del dispositivo está especificada en la hoja de datos del mismo en un diagrama de Smith y reproducido a continuación en la figura 5. Como se puede apreciar es necesario realizar una adaptación de impedancias para obtener máxima transferencia de energía desde el VCO al sintetizador, ésta adaptación es explicada en la sección de cálculos.

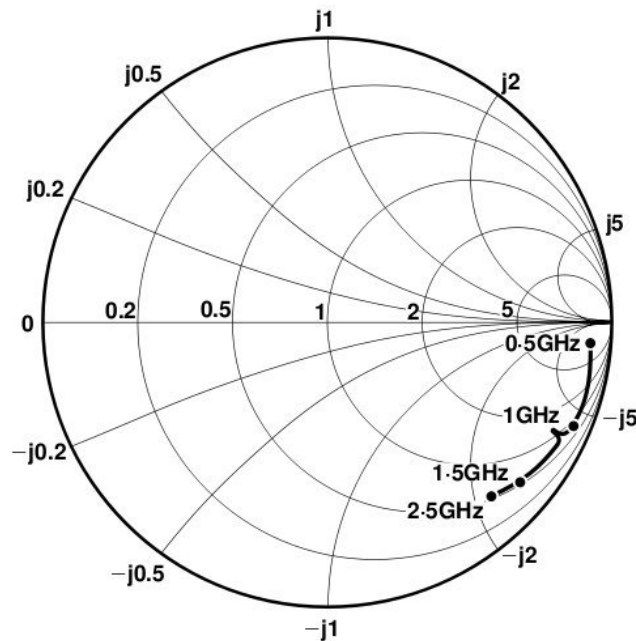


Fig. 5: Impedancia de entrada de RF de SP5769,
 $Z_0=50[\Omega]$

La salida del divisor programable es aplicada directamente al comparador de fase, en donde es comparada tanto en fase como en frecuencia con la frecuencia de comparación. Ésta última es obtenida de un oscilador a integrado con un cristal externo, luego ésta la frecuencia de referencia es dividida hasta la frecuencia de comparación por un divisor programable con 16 valores diferentes, detallados en la figura 7.

La salida del comparador está conectada a una bomba de carga la cual alimenta el filtro de lazo del PLL, y a un amplificador que controla un transistor externo que, utilizado junto con el filtro, integra los pulsos de la bomba de carga para controlar el varactor del VCO a utilizar.

El sintetizador SP5769 se programa vía un bus I²C, con velocidad clock de hasta 400[Khz], con niveles lógicos de tensión desde 3,3[V] hasta 5V. La selección de los niveles se realiza a través de P3, como es necesario en éste caso se lo deja a circuito abierto seleccionando la lógica de 5[V]. El protocolo de comunicación sigue el estándar descrito por Philips para el protocolo I²C[2], como un dispositivo esclavo dentro del bus.



Puede recibir, como enviar datos, en modo escritura y lectura respectivamente. El modo se selecciona con el **LSB** del byte de dirección que envía el Master del bus, si es un nivel lógico bajo (0[V]) selecciona el modo escritura, y si es alto (5[V]) selecciona el modo lectura.

La dirección a la que responde el dispositivo es programada vía el pin Address (10) con diferentes tensiones fijas, en nuestro caso para simplificar el circuito elegimos la dirección 01 dejando éste pin a circuito abierto.

La figura 6 muestra los bytes de configuración de escritura del VCO con los valores necesarios para este proyecto.

M S B						L S B				
Byte 1	Dirección	1	1	0	0	0	1	0	0	A
Byte 2	Divisor prog.	0	2 ¹⁴	2 ¹³	2 ¹²	2 ¹¹	2 ¹⁰	2 ⁹	2 ⁸	A
Byte 3	Divisor prog.	2 ⁷	2 ⁶	2 ⁵	2 ⁴	2 ³	2 ²	2 ¹	2 ⁰	A
Byte 4	Control	1	0	0	0	R 3	R 2	R 1	R 0	A
Byte 5	Control	0	0	0	0	0	0	0	0	A

Fig. 6: Bytes de configuración de escritura SP5769

Aquí se configura: Modo Test Off, Corriente de la bomba de carga $I_{cp} = \pm 155[\mu A]$, pin de REF/COMP en alta impedancia, y todos los puertos, P3 a P0 a 0[V].

Los valores R3, R2, R1 y R0 son los bits de configuración del divisor de la frecuencia de referencia, la figura 7 contiene los diferentes valores para éste divisor programable.

R3	R2	R1	R0	Divisor de referencia
0	0	0	0	2
0	0	0	1	4
0	0	1	0	8
0	0	1	1	16
0	1	0	0	32
0	1	0	1	64
0	1	1	0	128
0	1	1	1	256
1	0	0	0	24
1	0	0	1	5
1	0	1	0	10
1	0	1	1	20
1	1	0	0	40
1	1	0	1	80
1	1	1	0	160
1	1	1	1	320

Fig. 7: Configuración divisor de referencia

La frecuencia resultante de dividir la frecuencia de referencia en éste divisor, es la frecuencia de comparación que tendrá el comparador de fase-frecuencia como entrada, y también es la separación del ancho del canal.

		MSB						LSB		
Byte 1	Dirección	1	1	0	0	0	1	0	1	A
Byte 2	Status	POR	FL	0	0	0	0	0	0	A

Fig. 8: Bytes en modo lectura del SP5769

En el modo lectura, el dispositivo recibe el byte con la dirección y el bit de escritura/lectura en estado alto y devuelve un byte de status. El mismo está especificado en la figura 8.

El bit POR en estado alto indica que la alimentación estuvo por debajo de los 3[V], lo que puede indicar que recién inicia y los datos programados pueden estar corruptos.

El bit FL en estado alto, indica que el dispositivo esta enclavado en fase, y en bajo no lo está.

1.5.2 Oscilador

Se ha elegido trabajar con el VCO HMC534LP5E de Hittite[3]. Este dispositivo está realizado con tecnología MMIC e integra resonadores, dispositivos de resistencia negativa, diodos varactores, y tiene salidas de $F_o/2$ y $F_o/4$, ésta última pasando a través de un divisor.

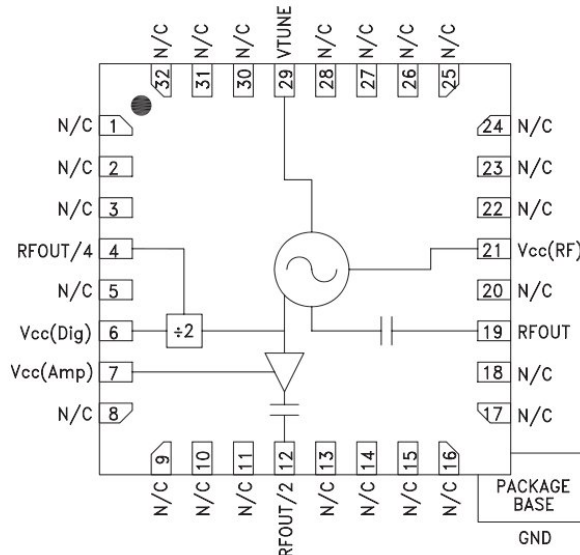


Fig. 9: Diagrama esquemático y de conexiones del VCO HMC534LP5E

La potencia típica de salida es de 11[dBm] y de -6[dBm] en $F_o/4$ cuando está alimentado con una fuente de 5V. Se puede evitar el consumo de potencia innecesaria apagando el amplificador de salida de $F_o/2$, y la alimentación del prescaler de $F_o/4$. En esta aplicación, sólo utilizará las salidas principal y la salida de $F_o/4$, por lo que el pin de alimentación del amplificador de $F_o/2$ lo dejamos a circuito abierto para economizar consumo.

Como es notorio en el diagrama de conexiones, y la hoja de datos del oscilador, el empaquetado del circuito se llama QFN 5x5, que es un diseño sin patas de montaje superficial de 25[mm²], y la conexión de masa de este tipo de empaquetado está sobre la superficie de la parte inferior del mismo, ocupando la mayor parte de ésta.

La respuesta en frecuencia del oscilador está representando en la curva de la figura 10, y la de la salida de $F_o/4$ en la figura 11.

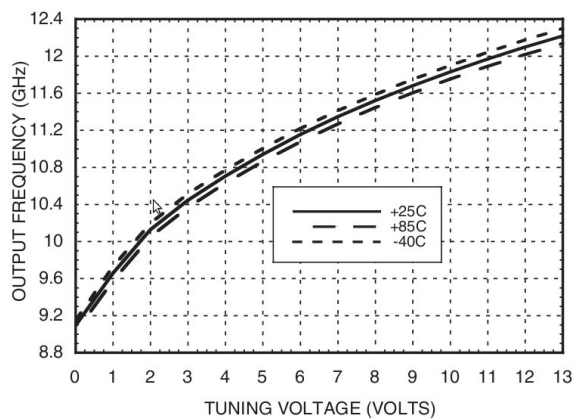


Fig. 10: Frecuencia de salida vs. Tensión de sintonía, $V_{cc}=5[V]$

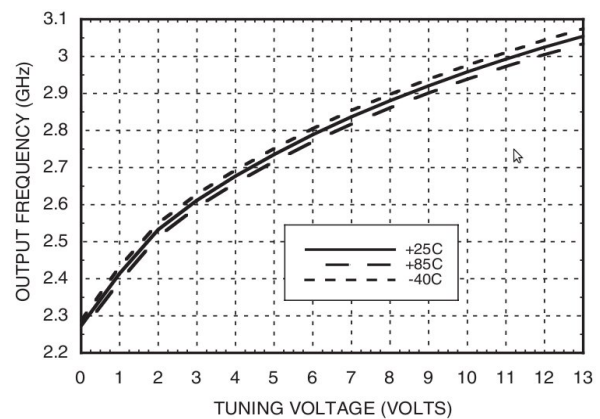


Fig. 11: Frecuencia de salida/4 vs. Tensión de sintonía, $V_{cc}=5[V]$

La salida de RF está desacoplada de continua a través de un capacitor, pero no es así con la salida del prescaler de $F_o/4$, por lo tanto es necesario agregar antes de la adaptación de impedancias un capacitor lo suficientemente grande para no modificar la red de adaptación. Observando el circuito de prueba propuesto por Hittite, en la salida del prescaler se colocará un capacitor de 1[nF] para desacoplar la tensión continua sobre la entrada del sintetizador.

El diagrama esquemático utilizado ésta aplicación, basado en lo propuesto por el fabricante es el siguiente:

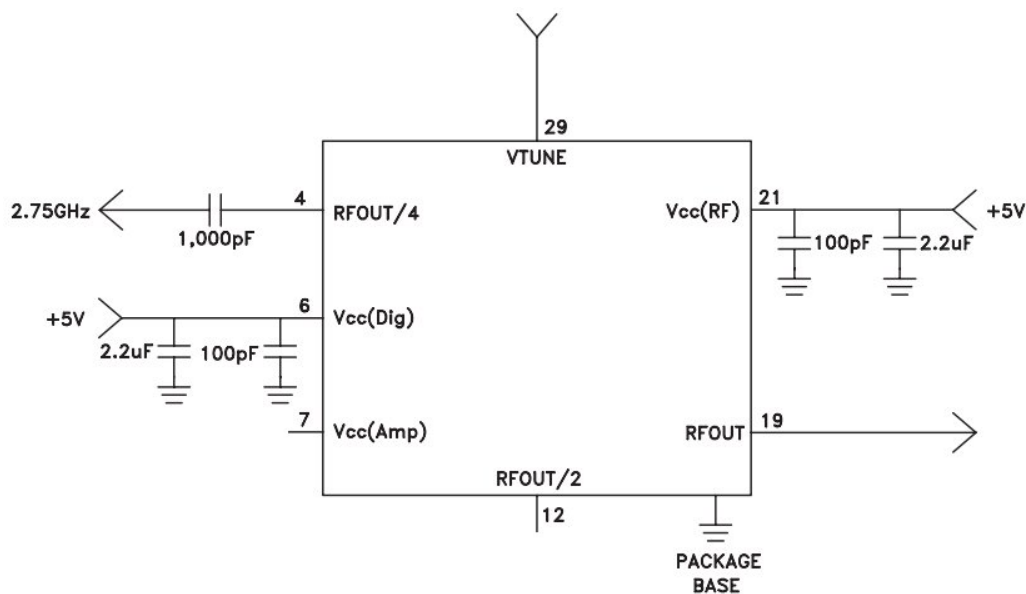


Fig. 12: Esquema de conexión del HMC534LP5

1.5.3 Microcontrolador

Para poder programar el sintetizador se puede utilizar un microcontrolador con al menos dos entradas/salidas digitales y programar dentro del mismo el protocolo I²C según las especificaciones de Philips, o utilizar un microcontrolador con un módulo I²C integrado.

Se ha seleccionado un microcontrolador de Microchip de 8 bits, el 16F873A[4], que tiene un módulo I²C integrado, además 3 puertos de E/S, que pueden ser utilizados en diferentes configuraciones, conversor A/D de 10-bit hasta 5 canales, 2 comparadores analógicos, y un puerto serial asincrónico universal.

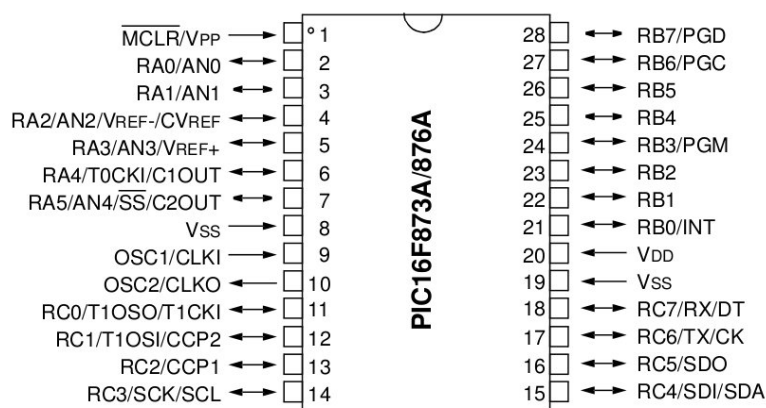


Fig. 13: Esquema de conexiones del microcontrolador
PIC16F873A

Se ha utilizado Hi-Tech C como compilador de C, lenguaje con el que se programó el microcontrolador, utilizando como entorno de programación la herramienta Microchip, MPLab.

El programa principal utilizado se encuentra en el anexo del documento, pero detallaremos a continuación las operaciones básicas que debe realizar el mismo:

- Configurar el módulo I²C como en Master mode
- Establecer la velocidad de datos en 100[Khz]
- Utilizar dos puertos del microcontrolador para obtener una salida visual de la información del programa utilizando una matriz de 6 módulos de 7 segmentos.
- Enviar y recibir los bytes de configuración y de status del sintetizador
- Leer un puerto digital para seleccionar entre 2 frecuencias preestablecidas para probar el funcionamiento del sistema.

2 Circuito esquemático

2.1 Sintetizador y VCO

A continuación se detalla el circuito esquemático de conexiones del sintetizador y el VCO, como se puede apreciar están los ya definidos los valores de los componentes, **excepto C1, C2 y R1, que son los componentes del filtro de lazo**, que serán detallados en la sección de cálculos.

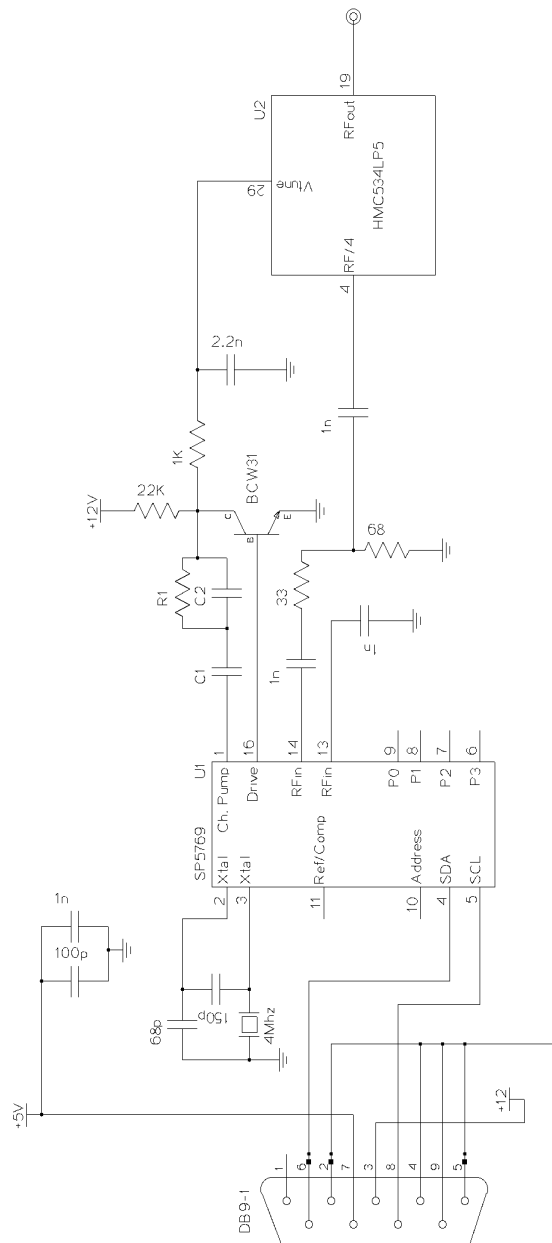


Fig. 14: Circuito esquemático PLL

2.2 Microcontrolador y display

El circuito esquemático siguiente, contiene la interconexión entre el microcontrolador, el display donde se muestra la información de la frecuencia seleccionada y el status de enclavamiento.

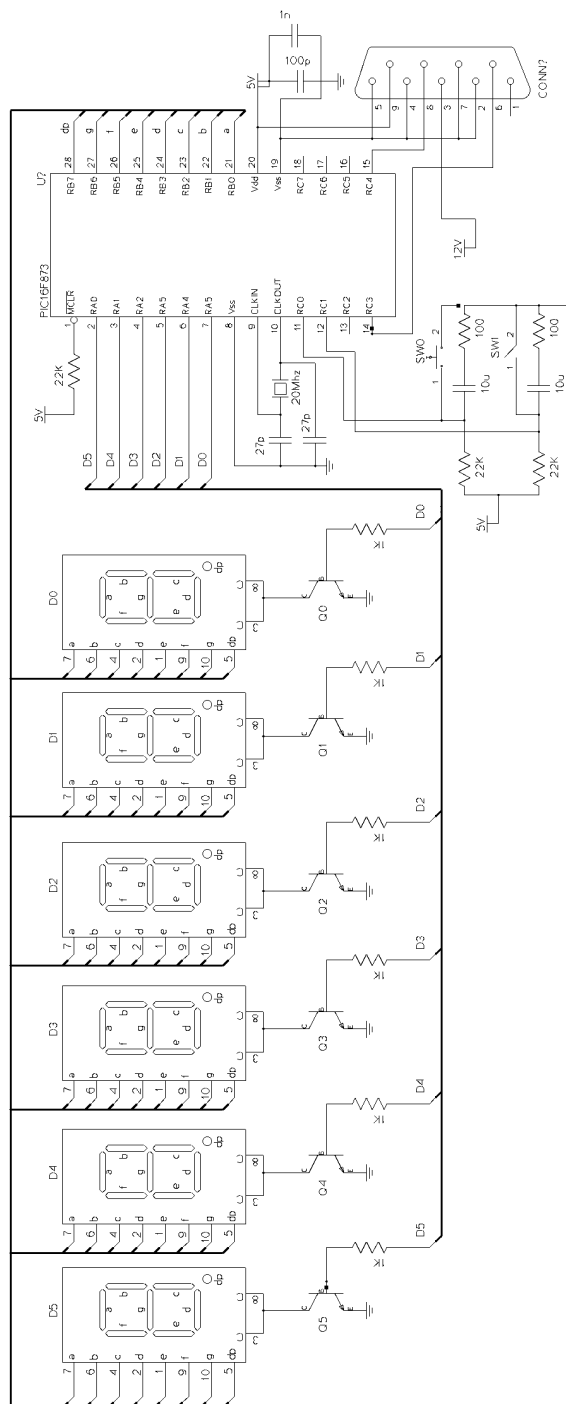


Fig. 15: Circuito esquemático microcontrolador y display

3 Cálculos

En ésta sección se explica los cálculos principales de diseño de las partes del circuito. Cabe destacar que en el diseño del equipo completo se tuvo en cuenta los circuitos de aplicación de cada uno de los componentes, facilitándose así el diseño.

3.1 Filtro de lazo del PLL

Esta es una de las partes más críticas en el diseño de un PLL, ya que si no es adecuado produciría retardos en el tiempo de enclavamiento, o en el peor de los casos no llegaría a enclavar. También es necesario para eliminar el ruido de fase en la salida.

Para realizar éstos cálculos se basó en una nota de aplicación de Zarlink[5], que explica las bases de diseño de sintetizadores para receptores de TV y satélite.

Definiremos a continuación el tipo de filtro a utilizar, las partes que lo componen, su función transferencia y la forma de calcularlo. El circuito del filtro a utilizar está sugerido por el fabricante en un circuito de aplicación del sintetizador y en la nota de aplicación mencionada anteriormente, y es el siguiente:

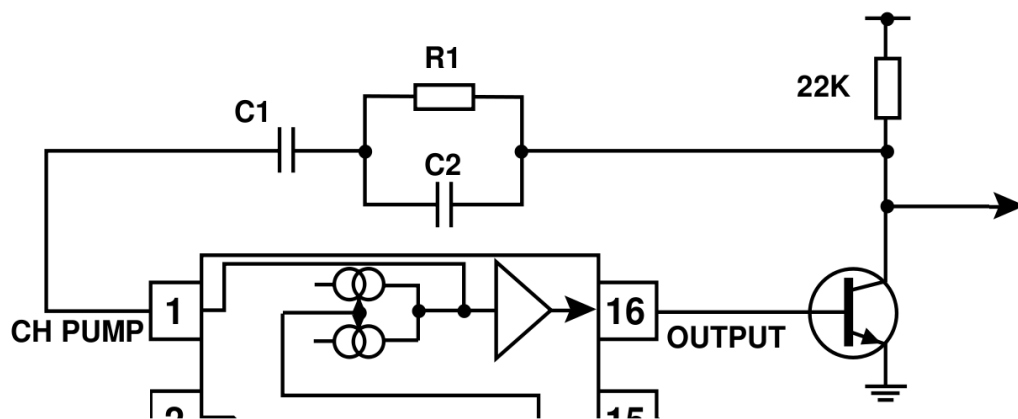


Fig. 16: Filtro propuesto por el fabricante

El filtro integra los pulsos de la bomba de carga, y el transistor alimentado con la tensión correcta para lograr la excursión en el rango de frecuencias del VCO. Éste circuito tiene equivalente que permite simplificar el análisis de la función transferencia.

El detector de fase/bomba de carga tiene como salida pulsos de corriente $I_{cp}[\mu A]$ de frecuencia F_{comp} , y de ancho proporcional al error de fase. Por lo tanto la ganancia del detector de fase en conjunto con la bomba de carga es:

$$Kd = \frac{I_{cp}}{2\pi} [\mu A/rad]$$

El filtro de lazo, es el que convierte los pulsos de la bomba de carga en una tensión

proporcional al error de fase, y tiene según el circuito equivalente de la figura 17, una función transferencia de segundo orden como la siguiente:

$$F(s) = \frac{(1 + sT_2)}{sT_1(1 + sT_3)} \quad \text{en donde} \quad \begin{aligned} T_1 &= C1 \\ T_2 &= (C1 + C2) R2 \\ T_3 &= C2 R1 \end{aligned}$$

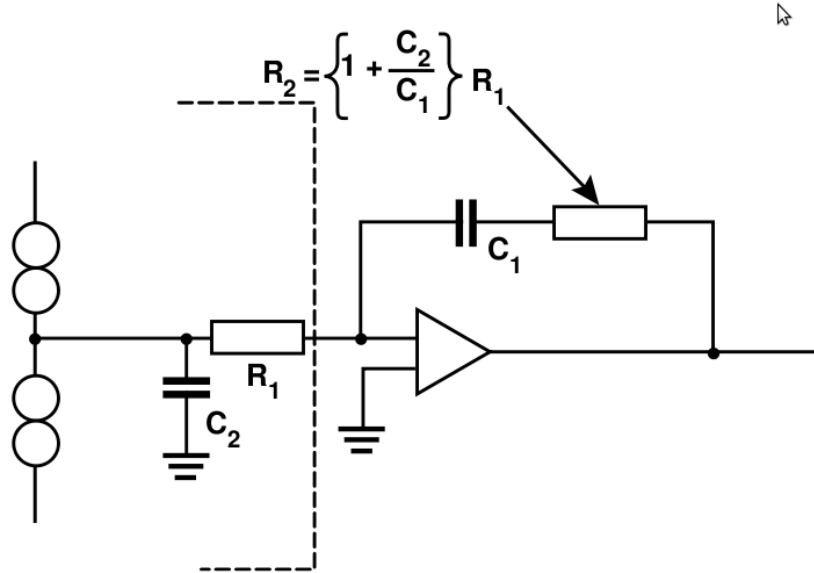


Fig. 17: Simplificación para calcular el filtro

Este filtro equivalente que se encuentra luego de R1, tiene una frecuencia natural ω_o y un factor de amortiguamiento ζ , que pueden seleccionarse de acuerdo a la aplicación del circuito. Luego de algunas simplificaciones de la función transferencia, encontramos los valores de C1, C2 y R2, con relación a la frecuencia natural y el factor de amortiguamiento:

$$\begin{aligned} C1 &= \frac{Kd Ko}{PN \omega_o^2} \\ C2 &= \frac{C1}{5} \\ R2 &= \frac{2\zeta}{\omega_o C1} \end{aligned}$$

En el grupo de ecuaciones anteriores se puede observar cuatro variables, P, N, Kd y Ko.

P es el factor de división del divisor fijo en la salida del VCO, es decir P=4, N el valor del divisor programable del sintetizador, Kd la ganancia del detector de fase/bomba de carga y Ko la constante del VCO, éstos dos últimos los obtenemos de las especificaciones de cada dispositivo.

Resolviendo la ecuación de transferencia del filtro[5][6], encontramos la relación entre la frecuencia natural ω_o , el factor de amortiguamiento ζ , el margen de fase y el tiempo de

establecimiento (ts) de $\Delta\omega_o$ con su respectivo error ω_e :

$$\zeta = \frac{\tan \phi}{2\sqrt{1+\tan^2 \phi}} \quad \omega_o = \frac{-\ln \left[\frac{\omega_e}{\Delta \omega_o} \sqrt{1-\zeta^2} \right]}{\zeta t_s}$$

Con el grupo de ecuaciones anteriores y el establecimiento de algunos parámetros se puede lograr calcular el filtro de lazo necesario para la aplicación.

Se eligen entonces los siguientes valores:

$$\phi = 50$$

Se elige éste valor ya que se desea que el sistema sea estable, por lo que es recomendable trabajar con un valor mayor que 45° .

$$\omega_e = 200 [Hz] \quad t_s = 10 [ms] \quad \Delta \omega_o = 200 [Mhz]$$

Es decir una vez que se programe el sintetizador y éste cambie en una diferencia de 200[Mhz] el error de la salida sea 200[Hz] luego de 10[ms].

Con los valores anteriores se obtienen los siguientes resultados:

$$\zeta = 0,478 \quad \omega_o = 2919 [rad]; \quad f_o = 464,6 [Hz]$$

Luego de la hoja de datos del VCO: $K_o = 200 (2\pi) [Mhz/V]$, y de la configuración de I_{cp} en el sintetizador $K_d = \frac{I_{cp}}{2\pi} [\mu A/rad]$, $I_{cp} = 155 [\mu A]$, se obtienen los valores finales para los componentes del filtro:

$$C_1 = 44,15 [nF] \quad C_2 = 8,8 [nF] \quad R_2 = 7,413 [K \Omega]$$

Y adoptando valores normalizados, se elige:

$$C_1 = 47 [nF] \quad C_2 = 10 [nF] \quad R_2 = 15 [K \Omega] // 15 [K \Omega] = 7,5 [K \Omega]$$

3.2 Potencia de entrada al sintetizador

Es preciso verificar que los niveles de salida del VCO y los de entrada del sintetizador en condiciones de máxima transferencia de energía sean compatibles, es decir la potencia de salida del VCO debe ser mayor que la potencia de entrada mínima requerida en el sintetizador.

Los valores máximos y mínimos están detallados por los fabricantes en la hoja de datos del cada componente[1][3]. Los valores de entrada del sintetizador son los que se muestran en la figura 4 y los del VCO en la figura 18.

Parameter		Min.	Typ.	Max.	Units
Frequency Range	Fo	10.6 - 11.8			GHz
	Fo/2	5.3 - 5.9			GHz
Power Output	RFOUT	+9		+14	dBm
	RFOUT/2	+8		+14	dBm
	RFOUT/4	-9		-3	dBm

Fig. 18: Potencia de salida del VCO en sus diferentes salidas

Tomando entonces la potencia de salida mínima en la salida de RF/4 del VCO:

$$P_{o.min} = -9 [dBm] \Rightarrow V_{o.min} = 79,34 [mV] / 50 [\Omega]$$

y comparando con los valores de la figura 4, se puede observar claramente que:

$$V_{i.min} = 40 [mV] \Rightarrow V_{o.min} > V_{i.min}$$

por lo que se comprueba que se cumplen los niveles de señal necesarios, por lo que sólo resta adaptar la impedancia de entrada del sintetizador a 50[Ω].

3.3 Cálculo de líneas de transmisión y adaptación de impedancia de entrada

Aunque en el diseño final del proyecto actual no se ha utilizado líneas del tipo microstripline, guía de ondas coplanares debido a que no se pudo realizar el circuito de RF sobre una placa de doble cara de cobre con un dieléctrico de teflón por limitaciones tecnológicas en la fabricación de la misma, aún así cabe explicar el cálculo de éstas y de la adaptación de impedancias utilizando stubs, ya que se puede pensar en ésta realización en el mediano plazo.

3.3.1 Diseño de la línea de transmisión en la banda de 10,6 a 11,8 [Ghz]

Ésta línea es la que transporta la salida del VCO a un conector tipo SMA, en el cual se conecta la línea coaxial lleva la señal hasta una guía de onda conectada a una antena.

Se considera que, en el conector SMA se conecta una carga de $50[\Omega]$, por lo que se establece que la impedancia característica de la línea será también igual a $50[\Omega]$.

Observando el tipo de línea utilizado en la placa de prueba que ofrece el fabricante del VCO, se opta por utilizar una guía de onda coplanar con plano de tierra para ésta sección del circuito.

Los cálculos siguientes están basados en ecuaciones, desarrolladas en “Transmission Line Design Handbook ” de Wadell[7] y comprobados por simulación con el software Microwave Office.

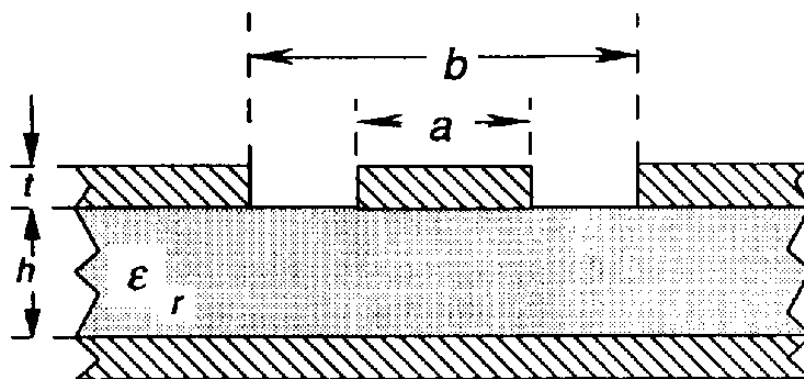


Fig. 19: Corte de una guía de onda coplanar

La impedancia característica está definida por:

$$Z_o = 60 \frac{\pi}{\sqrt{\epsilon_{eff}}} \frac{1}{\frac{k}{k'} + \frac{k_1}{k_1'}}$$

$$k = \frac{a}{b} \quad k' = \sqrt{1 - k^2}$$

$$k_1 = \frac{\tanh(\frac{\pi a}{4h})}{\tanh(\frac{\pi b}{4h})} \quad k_1' = \sqrt{1 - k_1^2}$$

$$\epsilon_{eff} = \frac{1 + \epsilon_r \frac{k' k_1}{k k_1'}}{1 + \frac{k' k_1}{k k_1'}}$$

Se ha utilizado en éste caso un placa laminada de cobre doble cara con dieléctrico de teflón GML1000 con las siguientes características:

$$h = 20 [\text{mils}] \quad t = 2 [\text{mils}] \quad \epsilon_r = 3,05 \quad \text{Constante en entre continua y } 10 [\text{Ghz}]$$

Por método de prueba y error o utilizando el siguiente programa en Matlab 8 para recorrer el una curva de impedancia basada en las ecuaciones anteriores:

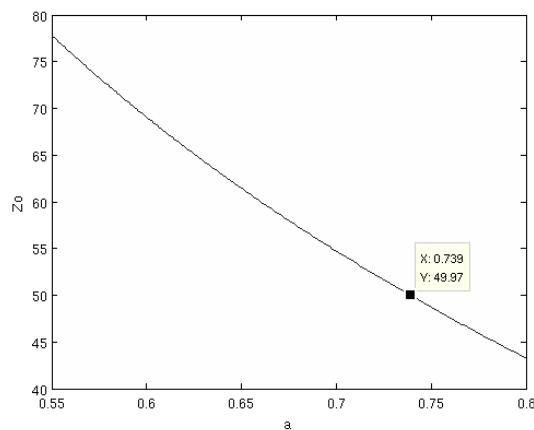
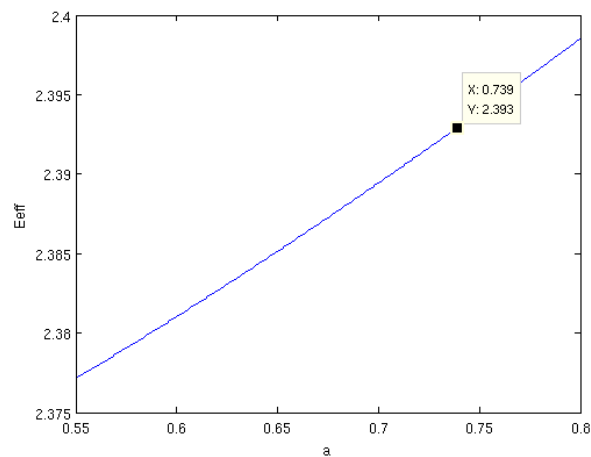


Fig. 20: Salida del programa en Matlab, Z_o vs. a

Fig. 21: E_{eff} vs. a

```

%%% Calculo de impedancia caracteristica
%%% para coplanar waveguides
%%% obtenemos valores en mm
Er=3.05;
a=0.55:0.001:0.8;
b(1:length(a))=1.2;
h(1:length(a))=0.020*25.4;

k=a./b;
kk=sqrt(1.-k.^2);
ARG_NUM=pi.*a./(4.*h);
ARG_DEN=pi.*b./(4.*h);
NUM=tanh(ARG_NUM);
DEN=tanh(ARG_DEN);
k1=NUM./DEN;
kk1=sqrt(1.-k1.^2);

Eeff=(1.+Er.*(kk.*k1./(k.*kk1)))./(1.+kk.*k1./(k.*kk1));
Z0=60*pi./((sqrt(Eeff)).*(k./kk)+k1./kk1);
figure(1);
plot(a,Z0);xlabel('a');ylabel('Zo')
figure(2);
plot(a,Eeff);xlabel('a');ylabel('Eeff')

```

Cuadro 1: Programa en Matlab 2008 para encontrar las dimensiones de una guía de onda coplanar con plano de tierra

Con éste programa obtenemos las siguientes dimensiones y características:

$$a=29[mils] \quad b=47[mils] \quad Z_o=50[\Omega] \quad \epsilon_{eff}=2,393 \Rightarrow fp=\frac{1}{\sqrt{\epsilon_{eff}}}=0,6464$$

$$\text{De lo que deducimos el valor de } \lambda_{cpw}=\frac{\lambda_o}{\sqrt{\epsilon_{eff}}}=\frac{3*10^8}{11,2*10^9}0,6464=0,0173[m]$$

tomando la frecuencia central en 11.2[Ghz], diseñar una línea de $n\lambda_{cpw}/2$ hasta el conector SMA.

3.3.2 Diseño de la línea de transmisión en la banda de 2,65 a 2,95 [Ghz]

Una segunda opción es utilizar el software para cálculo de líneas de transmisión que acompaña a el programa Microwave Office, llamado TXline. Los valores que resultan son bastantes aproximados al método utilizado en el punto anterior.

Para ésta línea utilizaremos microstripline para no tener longitudes grandes de línea.

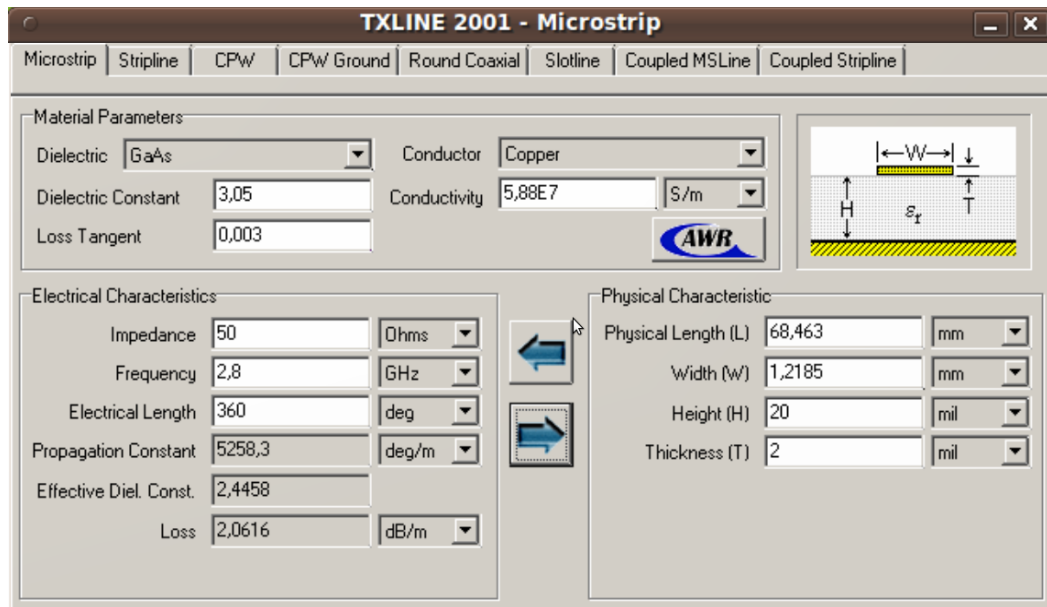


Fig. 22: Programa TXLINE para cálculo de líneas de transmisión

De aquí se obtienen los siguientes resultados:

$$W=1,2185[mm] \quad \lambda_{ms}=68,46[mm] \quad fp=\frac{1}{\sqrt{2,4458}}=0,6394$$

3.4 Adaptación de impedancia de entrada en el sintetizador

Utilizando los datos obtenidos en el punto 3.3.2 y una carta de Smith, podemos diseñar la adaptación tomando como impedancia a adaptar la dada para 2,5[Ghz] por el Zarlink, es decir $Z_i=25-j120[\Omega]$, entonces normalizando todo a $50[\Omega]$ el diagrama de adaptación nos queda:

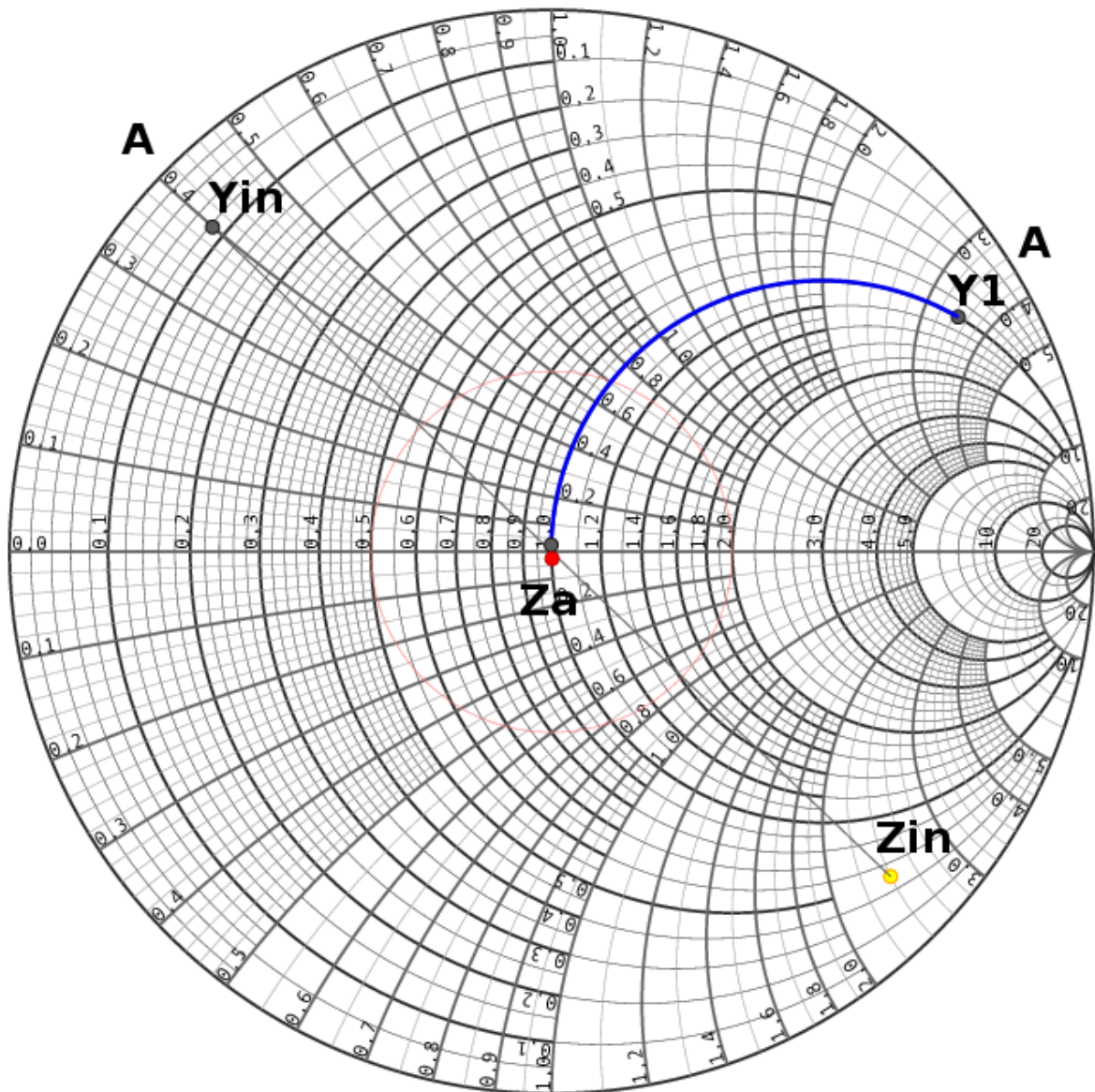


Fig. 23: Adaptación de impedancia con un stub, $Z_{in}=25-j120$

Utilizando la microstripline calculada, a una distancia $A=10,1[\text{mm}]$ se debe poner un stub a circuito abierto de longitud $l_s=20,2[\text{mm}]$.

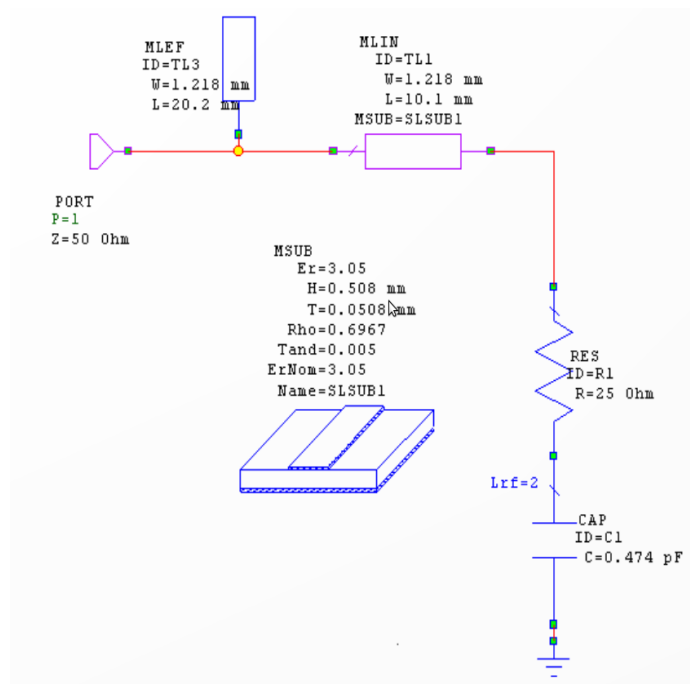


Fig. 24: Adaptación de impedancia de entrada del sintetizador con un stub en circuito abierto, $R1$ y $C1$ representan la impedancia de entrada del sintetizador

3.5 Adaptación de impedancia en el diseño actual

Al no tener éxito con la creación del circuito impreso en el laminado de teflón, se tuvo que encontrar una solución alternativa, más empírica que teórica.

La prueba se realizó utilizando un laminado de fibra de vidrio (FR4) de 0,8[mm] de espesor para la realización del circuito impreso de RF.

Para evitar pérdida de potencia en la salida del VCO, ya que éste dieléctrico tiene altas pérdidas en el orden de los 10[Ghz], se acortó la línea de salida lo máximo posible, es decir se ubicó el pin del conector SMA lo más próximo a la soldadura de la salida del VCO.

Por otro lado se tomó la misma adaptación de impedancia de entrada del sintetizador que se encuentra en el circuito de prueba en la hoja de datos del mismo.

Con éstas modificaciones se han realizado mediciones muy satisfactorias.

4 Programación del microcontrolador

El microcontrolador PIC16F873A tiene incluido un módulo serial síncrono, y éste puede funcionar en dos modos distintos, Serial Peripheral Interface (SPI) o Inter-Integrated Circuit(I²C). En el modo I²C puede actuar tanto como Master del bus o Slave.

Se utilizará en éste caso el protocolo I²C para programar el sintetizador del sistema, por lo que se debe configurar el módulo serial de manera tal que la comunicación sea correcta.

El protocolo I²C utiliza sólo dos pines del microcontrolador ellos son:

- Serial clock (SCL) – Pin RC3
- Serial data (SDA) – Pin RC4

4.1 Inicialización del protocolo I²C en el microcontrolador

El módulo serial para trabajar en I²C utiliza seis registros del microcontrolador:

- Registros de control SSPCON y SSPCON2
- Registro de estado SSPSTAT
- Búfer de transmisión/recepción (SSPBUF)
- Registro de dirección (SSPADDD)

Para inicializar el módulo correctamente se deben seguir los siguientes pasos:

- 1) Configurar los pines 3 y 4 del PORTC como entrada
- 2) Configurar el módulo en modo Master I²C.
- 3) Configurar el bitrate del protocolo
- 4) Desactivar control de slew-rate

4.1.1 Configurar los pines 3 y 4 del PORTC como entrada

Para realizar éste paso es necesario setear `TRISC=TRISC&12`

4.1.2 Configurar en modo Master.

El modo Master I²C se selecciona configurando el registro SSPCON, el valor necesario para habilitar el módulo serial y el modo de operación del mismo, en éste caso es necesario escribir el registro `SSPCON=28H`

4.1.3 Configurar el bitrate del protocolo

El valor del bitrate está dado por la siguiente ecuación: $clock = \frac{F_{osc}}{4 * (SSPADD + 1)}$
por lo que para setear un clock de 100[Khz] con un cristal de 4[Mhz] en el microcontrolador resulta SSPADD=49.

4.1.4 Desactivar slew-rate

Se debe desactivar el control de slew-rate porque se está utilizando un bitrate de 100[Khz] en el bus I²C y éste control no es necesario para ésta velocidad, para poder realizar esto es necesario escribir un 1 en el bit SMP del registro SSPSTAT.

4.2 Lectura y escritura del bus I²C

Básicamente el módulo integrado al microcontrolador, realiza la comunicación casi automáticamente, pero es necesario tener en cuenta los flags que se debe controlar en cada lectura/escritura realizada.

La escritura sobre el bus se realiza enviando una condición de Start sobre el bus, ésta condición la realiza el Master, en este caso el microcontrolador, y se logra escribiendo el registro SSPBUF la dirección a donde se quiere comunicar en poniendo el último bit de ese byte en estado bajo, para indicar que se va a escribir sobre esa dirección, luego espera un acknowledge por parte del sintetizador.

Para lograr la lectura del estado del sintetizador el procedimiento es parecido, es decir primero el microcontrolador envía el byte de dirección con el último bit en estado alto y espera un acknowledge, una vez aceptado éste por el sintetizador, en el microcontrolador se debe setear el flag RCEN en estado alto, para que el módulo comience el ciclo de lectura de un byte.

4.2.1 Acknowledge

Cada byte transmitido correctamente por el bus I²C, debe estar seguido de un acknowledge por parte del receptor para indicar la correcta recepción del byte. Tanto para el sintetizador cuando recibe los bytes de configuración, como para el microcontrolador cuando recibe el byte de status.

Si esta secuencia no es realizada como está definida en la especificación del protocolo, los dispositivo generan una condición de stop interna y la escritura o lectura puede no haber sido correcta, por lo que es aconsejable reiniciar el proceso de lectura/escritura.

5 Mediciones

Se realizaron dos tipos de mediciones, una con el banco de microondas y otra con un LNB y un analizador de espectro de hasta 2,7[Ghz].

5.1 Medición con el banco de microondas

Ésta fue la primera medición realizada, y si bien no es cuantitativa respecto al espectro exacto que se está generando, nos da precisamente la frecuencia que se está generando. Para ésta medición se trabajó con guías de onda, atenuadores calibrados, antenas, un ondámetro y un frecuencímetro, ambos de cavidad resonante.

Para medir la frecuencia que emite el PLL se utiliza un diodo rápido como detector que rectifica la señal de RF que recibe una antena, ésta señal rectificada se la mide con un microamperímetro que da un nivel de continua indicando que hay señal de RF. Cuando se ajusta el ondámetro, se encuentra un punto donde la cavidad está en resonancia a la frecuencia que se emite y la mayor parte de la energía es absorbida por la cavidad, por lo que en el detector se halla un mínimo en la lectura. Es en éste punto que se toma lectura de la longitud de onda en la guía observando el valor en el tornillo micrométrico del ondámetro, y con simples cálculos se obtiene la frecuencia a la cual se está transmitiendo.

5.2 Medición del espectro de frecuencias

Éstas mediciones sirvieron para ver indirectamente el espectro que se genera con el PLL. Para ello se utiliza un LNB de la empresa Chaparral que trabaja en banda Ku, de 10,7 a 12,75 [Ghz] y tiene una salida de frecuencia intermedia entre 950[Mhz] y 2,15[Ghz], por lo que es observable con el analizador de espectros. El LNB amplifica y mezcla la frecuencia de entrada con una frecuencia de oscilador local de 9,75 [Ghz], entregando en la salida el espectro de la diferencia de frecuencias, por lo tanto las frecuencias recomendadas como límites mínimo y máximo para generar son $F_{min}=10,7$ [Ghz] y $F_{max}=11,9$ [Ghz].

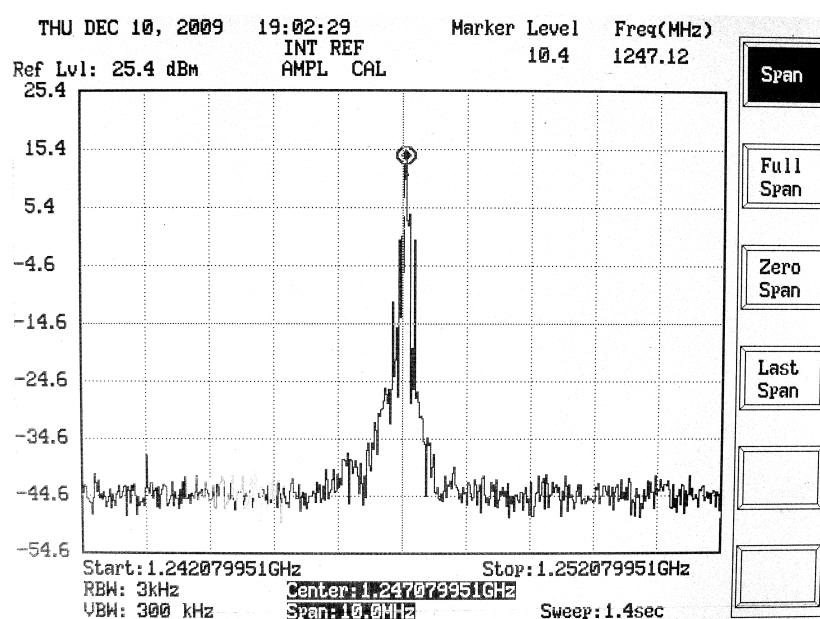


Fig. 25: Medición de la frecuencia de 11[Ghz]

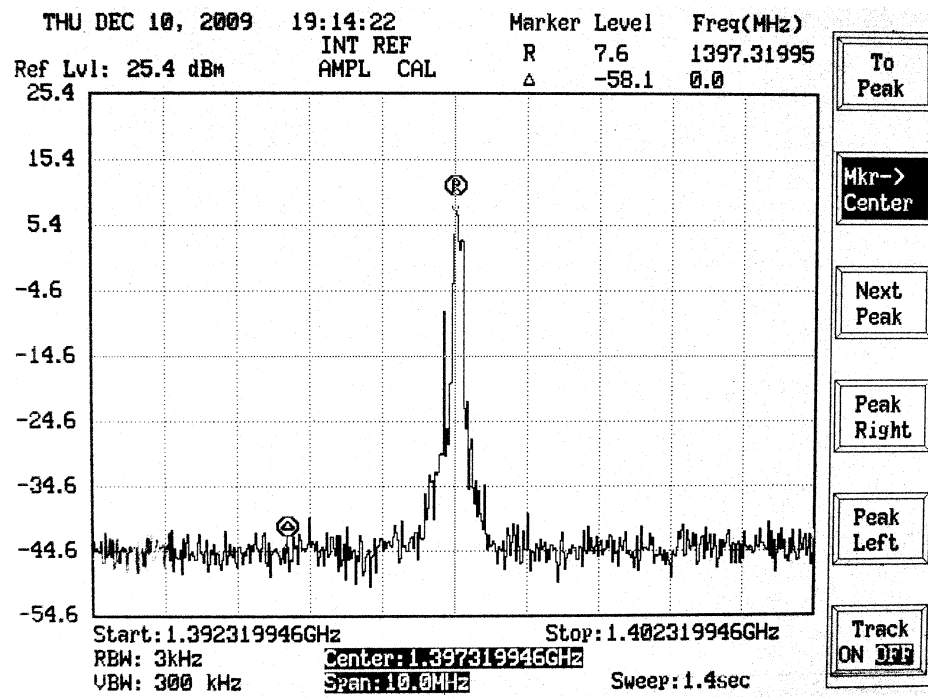


Fig. 26: Medición de la frecuencia de 11,5[Ghz]

En la figura 25 se puede observar una de las mediciones realizadas cuando se programó el PLL para que trabaje a una frecuencia de 11[Ghz]. La frecuencia marcada es 1247[Mhz] aproximadamente, y si a ésta se le suma el valor de la frecuencia del oscilador local del LNB, se obtiene la frecuencia recibida en la antena, es decir 10,997[Ghz]. Hay un error de 3[Mhz], el cual es atribuible a la estabilidad del oscilador local del LNB, como lo especifica el fabricante.

Otra medición, pero en ésta ocasión la frecuencia seleccionada es 11,15[Ghz], se puede ver en la figura 26.

También es destacable que el nivel de señal obtenido es 50[dB] aproximadamente superior al ruido.

6 Conclusiones

Luego de haber concretado el objetivo de éste proyecto exitosamente, se han dejado sentadas las bases para continuar una línea de investigación en microondas, disponiendo el Laboratorio de Telecomunicaciones de la Facultad de Ciencias Exactas, de un generador de frecuencias en microondas programable.

Se ha podido comprobar el perfecto funcionamiento del PLL, en su programación, enganche y potencia de salida.

Se debe pensar ahora en las posibilidades de ampliación de éste proyecto, como por ejemplo la modulación para comunicaciones y la implementación de un módulo receptor, realizando ensayos de ancho de banda y diferentes tipos de modulación, y el control del PLL a través de un PC.

Bibliografía

- [1]Zarlink Semiconductors, "SP5769 Datasheet", Zarlink Semiconductors, pp. 1-6, Mayo 2002
- [2]Philips Semiconductors, "The I²C Specification", Philips Semiconductors, pp. 3-40, año 2000
- [3]Hititte Microwave Corp., "HMC534LP5 Datasheet", Hititte Microwave Corp., pp. 1-8, año 2006
- [4]Microchip Technology Inc., "PIC 16F87XA Datasheet - 28/40/44- Pin Enhaced Flash Microcontrolers", Microchip Technology Inc., pp. 1-234, julio 2003
- [5]Zarlink Semiconductors, "AN168, TV/Satellite Synthesisers - Basic Design Guidelines", Zarlink Semiconductors, pp. 2-9, año 1995
- [6]V. F. Kroupa, "Phase Lock Loops and Frequency Synthesis",Wiley & Sons, cap. 1 y 2 , año 2003
- [7]Wadel, Brian C., "Transmission Line Design Handbook",Artech House Inc., pp. 79 y 80, año 1991

ANEXO – Código fuente del programa para el microcontrolador

archivo i2c.c

```
#include <htc.h>
```

```
#include "i2c.h"
```

```
void init_i2c(void) {
```

```
    PORTA=0;
```

```
    PORTB=0;
```

```
    PORTC=0;
```

```
    ADCON1=6;
```

```
    TRISA=0;
```

```
    TRISB=0;
```

```
    TRISC=0;
```

```
    TRISC3=1;
```

```
    TRISC4=1;
```

```
    SSPCON=0x38;
```

```
    SSPCON2=0;
```

```
    SSPADD=49; //49=> 100Khz          99->50Khz          12-> 384Khz
```

```
    CKE=0;
```

```
    SMP=1;
```

```
    BCLIF=0;
```

```
    SSPIF=0;
```

```
    OPTION=0x7F;
```

```
    GIE=0;
```

```
    SSPIE=1;
```

```
    PEIE=1;
```

```
}
```

```
void i2c_waitForIdle(void) {
```

```
    while ((RW & (SSPCON2 & 0x1F))!=0) {}
```

```
    _delay(12*(_XTAL_FREQ/4000000)); //Mientras exista start, restart, stop,ack, rcv o send o haya una tx,  
    espera
```

```
}
```

```
void i2c_start(void) {
```

```
    i2c_waitForIdle();
```

```
    SEN=1;
```

```
}
```

```
void i2c_repstart(void) {
```

```
    i2c_waitForIdle();
```

```
    RSEN=1;
```

```
}
```

```
void i2c_stop(void) {
```

```
    i2c_waitForIdle();
```

```
    PEN=1;
```

```
}
```

```
int i2c_read(unsigned char ack){
```

```
    unsigned char i2cReadData;
```

```
    i2c_waitForIdle();
```

```
    RCEN=1;
```

```
    i2c_waitForIdle();
```

```

    i2cReadData=SSPBUF;
    i2c_waitForIdle();
    if (ack){
        ACKDT=0;
    }
    else
    {
        ACKDT=1;
    }
    ACKEN=1;
    return (i2cReadData);
}

unsigned char i2c_write(unsigned char i2cWriteData) {
    i2c_waitForIdle();
    SSPBUF=i2cWriteData;
    return(!ACKSTAT);
}

```

archivo i2c.h

```

#ifndef _I2C_H_
#define _I2C_H_

```

```

#ifndef _XTAL_FREQ
#define _XTAL_FREQ 4000000
#endif

```

```

extern void          init_i2c(void);
extern void          i2c_waitForIdle(void);
extern void          i2c_start(void);
extern void          i2c_repstart(void);
extern void          i2c_stop(void);
extern int           i2c_read(unsigned char);
extern unsigned char i2c_write(unsigned char);

```

```

#endif          /* _I2C_H_ */

```

archivo main.c

```

#define _XTAL_FREQ 20000000
#include <htc.h>
#include <math.h>
#include <pic.h>
#include "i2c.h"

```

```

__CONFIG(HS & WDTDIS & PWRTDIS & BORDIS & LVPDIS & DUNPROT & WRTEN & DEBUGDIS &
UNPROTECT);

```

```

//definicion de caracteres para display
//const int Dis_E=0x79; const int Dis_I=0x30;
//const int Dis_n=0x54; const int Dis_t=0x46;

```

```
//unsigned char Dis_b=0x7C; unsigned char Dis_y=0xE6;
//unsigned char Dis_1=0x06; unsigned char Dis_2=0x9B;
//unsigned char Dis_3=0x4F; unsigned char Dis_4=0x66;
//unsigned char Dis_A=0x77; unsigned char Dis_d=0x5E;
//unsigned char Dis_r=0x50; unsigned char Dis_o=0x5C;
//unsigned char Dis_C=0x3C; unsigned char Dis_k=0x74;
//unsigned char Dis_P=0x73; unsigned char Dis_L=0x38;
//unsigned char Dis_S=0x6D;
```

```
//mensajes
```

```
unsigned char data1=0x4F;
unsigned char data2=0xB9;
unsigned char data3=0x84;
unsigned char data4=0x00;
```

```
unsigned char data5=0x4C;
unsigned char data6=0xBE;
unsigned char data7=0x84;
unsigned char data8=0x00;
```

```
unsigned char addr=0xC2;
unsigned char addrREAD=0xC3;
```

```
const int Elnit[] = {0x00,0x30,0x54,0x30,0x46,0x00}; // Dis_E, Dis_I, Dis_n, Dis_I, Dis_t,0};
const int EAddr[] = {0x79,0x77,0x5E,0x5E,0x50,0x79};
const int EByte1[] = {0x79,0x7C,0xE6,0x46,0x79,0x06};
const int EByte2[] = {0x79,0x7C,0xE6,0x46,0x79,0x9B};
const int EByte3[] = {0x79,0x7C,0xE6,0x46,0x79,0x4F};
const int EByte4[] = {0x79,0x7C,0xE6,0x46,0x79,0x66};
const int TodoOK[] = {0x5C,0x74,0x00,0x00,0x00,0x00};
const int Stop[] = {0x6D,0x46,0x5C,0x73,0x00,0x00};
const int por[] = {0x73,0x5C,0x50,0x00,0x00,0x00};
const int nopor[] = {0x54,0x5C,0x73,0x5C,0x50,0x00};
const int pll[] = {0x73,0x38,0x38,0x00,0x00,0x00};
const int noload[] = {0x54,0x5C,0x38,0x5C,0x3C,0x74};
```

```
const int * pinit;
const int * paddr;
const int * pbyte1;
const int * pbyte2;
const int * pbyte3;
const int * pbyte4;
const int * pok;
const int * pstop;
const int * ppor;
const int * pnopor;
const int * pll;
const int * pnoload;
```

```
void display(const int *err) {
    int i,j;
    PORTA=0;
    for(i=0;i<10;i++)
        _delay(20*(_XTAL_FREQ/4000));

    for(j=0;j<50;j++) {
        for (i=0; i<=5; i++) {
            PORTA=(int) pow(2,i);
```

```

        PORTB=err[i];
        _delay(_XTAL_FREQ/4000);
    }
}

```

```

void main() {
    int h;
    int status;

    pinit=&EInit[0];
    paddr=&EAddr[0];
    pbyte1=&EByte1[0];
    pbyte2=&EByte2[0];
    pbyte3=&EByte3[0];
    pbyte4=&EByte4[0];
    pok=&TodoOK[0];
    pstop=&Stop[0];
    ppor=&por[0];
    pnopor=&nopor[0];
    ppll=&pll[0];
    pnolock=&nolock[0];

    init_i2c();
    display(pinit);
    for(h=0;h<200;h++) {
        _delay(20*(_XTAL_FREQ/4000));
    }
}

```

```

i2c_start();

```

```

if(!i2c_write(addr)) {
    display(paddr);
    i2c_stop();
    display(pstop);
    asm("goto 0x00");
}

```

```

if(!i2c_write(data5)) {
    display(pbyte1);
    i2c_stop();
    display(pstop);
    asm("goto 0x00");
}

```

```

if(!i2c_write(data6)) {
    display(pbyte2);
    i2c_stop();
    display(pstop);
    asm("goto 0x00");
}

```

```

if(!i2c_write(data7)) {
    display(pbyte3);
    i2c_stop();
    display(pstop);
    asm("goto 0x00");
}

```

```

if(!i2c_write(data8))
    display(pbyte4);
i2c_stop();
display(pstop);
SSPIF=0;
display(pok);

i2c_start();
if(!i2c_write(addrREAD)) {
    display(paddr);
    i2c_stop();
    display(pstop);
    asm("goto 0x00");
}

status=i2c_read(1);

i2c_stop();

if((status>>7) & 0x01)
    display(ppor);
else
    display(pnopor);

if((status>>6) & 0x01){
    while(1){
        display(ppll);
    }
}
else {
    display(pnolock);
}
}

```