



UNIVERSIDAD NACIONAL DE MAR DEL PLATA

FACULTAD DE INGENIERÍA

DEPARTAMENTO DE ELECTRÓNICA

PROYECTO DE GRADO PARA OBTENER EL TÍTULO DE INGENIERO EN ELECTRÓNICA

MEDIDOR DE IMPEDANCIAS PARA VHF

AUTOR: NAHUEL GASPAR DURÉ

DIRECTOR: Dr.Ing. ALEJANDRO JOSÉ URÍZ

MAR DEL PLATA, ARGENTINA

AÑO 2022



RINFI se desarrolla en forma conjunta entre el INTEMA y la Biblioteca de la Facultad de Ingeniería de la Universidad Nacional de Mar del Plata.

Tiene como objetivo recopilar, organizar, gestionar, difundir y preservar documentos digitales en Ingeniería, Ciencia y Tecnología de Materiales y Ciencias Afines.

A través del Acceso Abierto, se pretende aumentar la visibilidad y el impacto de los resultados de la investigación, asumiendo las políticas y cumpliendo con los protocolos y estándares internacionales para la interoperabilidad entre repositorios



Esta obra está bajo una [Licencia Creative Commons Atribución-
NoComercial-CompartirIgual 4.0 Internacional](https://creativecommons.org/licenses/by-nc-sa/4.0/).



UNIVERSIDAD NACIONAL DE MAR DEL PLATA

FACULTAD DE INGENIERÍA

DEPARTAMENTO DE ELECTRÓNICA

PROYECTO DE GRADO PARA OBTENER EL TÍTULO DE INGENIERO EN ELECTRÓNICA

MEDIDOR DE IMPEDANCIAS PARA VHF

AUTOR: NAHUEL GASPAR DURÉ

DIRECTOR: Dr.Ing. ALEJANDRO JOSÉ URÍZ

MAR DEL PLATA, ARGENTINA

AÑO 2022

Resumen

El medidor de impedancias es un instrumento que permite caracterizar de forma eficiente cualquier tipo de carga conectada a lo largo de un rango de frecuencia. Esta impedancia va a depender del tipo de componente. Con este instrumento es posible determinar los elementos parásitos de la impedancia y ver su modelo equivalente en alta frecuencia. Esto último, se puede lograr a partir de la gráfica de magnitud y fase de la impedancia medida con el instrumento.

El objetivo de este proyecto es la realización de un medidor de impedancias de bajo costo que pueda calcular la impedancia de cualquier carga. Esto incluye inductancias, capacitancias, etc en un amplio rango de frecuencias. Para poder visualizar la impedancia, se deberá desarrollar una aplicación para PC que lea los datos a través de una placa de sonido USB. Esta última, se usará para proteger la entrada de audio de la PC.

Índice General

Resumen	2
Índice de figuras	6
1. Introducción	10
1.1. Origen del problema.....	10
1.2. Definición de impedancia.....	10
1.3. Métodos de medición de impedancia.....	11
1.3.1. Método del Puente.....	11
1.3.2. Método resonante.....	12
1.3.3. Método $I - V$	13
1.3.4. Método RF $I - V$	13
1.3.5. Método de analizador de red.....	14
1.3.6. Método del puente auto balanceado.....	15
1.3.6.1 Funcionamiento.....	15
1.4. Objetivos.....	17
1.4.1. Objetivo general	17
1.4.2. Objetivos de desarrollo.....	17
2. Diseño del firmware del sistema	18
2.1. Introducción.....	20
2.2. Configuración de los registros del PIC.....	22
2.2.1. Oscilador.....	22
2.2.2. Multiplicador de frecuencia PLL.....	26
2.2.3. Configuración de los periféricos.....	26
2.2.4. Módulo UART.....	32
2.2.4.1. Transmisión.....	32
2.2.4.2. Recepción.....	34
2.2.4.3. Generador de tasa de baudios.....	36
2.2.5. Módulo puerto serie síncrono maestro (MSSP)	37

2.2.5.1. Bus I2C.....	37
2.2.5.2. Modo maestro.....	38
2.2.5.2.1. Configuración inicial del modo maestro.....	39
2.2.5.2.2. Operaciones en modo maestro I2C.....	40
2.3. Configuración de los registros del Si5351.....	43
2.3.1. Introducción.....	43
2.3.2. Funcionamiento.....	43
2.3.2.1. Sintetizador de frecuencias (PLL).....	44
2.3.2.2. Divisor multifraccionario de salida (OMD).....	44
2.3.2.3. Configuración de los registros.....	44
2.3.3. Configuración del Si5351.....	45
2.3.3.1. Configuración de los registros asociados al procedimiento de programación del Si5351.....	45
3. Diseño e Implementación del hardware y software del sistema	54
3.1. Introducción.....	54
3.2. Diseño de la placa de adquisición del PIC18F2550.....	54
3.2.1. Diseño e implementación del hardware del PIC18F2550.....	56
3.3. Diseño e implementación de los filtros pasa bajos de RF.....	57
3.3.1. Caracterización de la señal de salida del Si5351.....	57
3.3.2. Requisitos de diseño de los filtros de RF.....	59
3.3.3. Implementación y validación del funcionamiento de los filtros pasa bajos de RF.....	62
3.4. Diseño e implementación del medidor de Impedancias.....	69
3.4.1. Diseño de la primera etapa de RF del medidor de impedancias.....	69
3.4.1.1. Mezclador SA612A.....	70
3.4.1.2. Puente de impedancia.....	71
3.4.1.3. Resistencias de sensado.....	72
3.4.1.4. Adaptación de impedancias.....	75
3.4.1.5. Determinación del rango dinámico del SA612A.....	79
3.4.1.6. Resistencias de compensación.....	80
3.4.1.7. Diseño de las protecciones para el integrado SA612A.....	81

3.4.2.	Diseño de la segunda etapa de audio del medidor de impedancias.....	82
3.4.2.1.	Configuración de los filtros elípticos de 8vo orden MAX7400.....	82
3.4.2.2	Diodos de protección para entrada de línea.....	84
3.4.2.3	Verificación del filtrado de la señal de RF a salida de los filtros MAX7400.....	85
3.4.3.	Diseño del pcb del medidor de impedancias.....	87
3.4.3.1.	Reglas de diseño aplicadas.....	87
3.4.3.2.	Diseño del camino de VCC.....	91
3.4.3.3.	Medidor de impedancias físico.....	91
3.5.	Diseño del Software del sistema.....	92
3.5.1.	Interfaz gráfica de usuario (GUI).....	92
3.5.2.	Diagrama de flujo del código funcional de la interfaz gráfica de usuario (GUI).....	94
4.	Validación del prototipo	95
4.1.	Medición sobre carga de 47 Ω	95
4.2.	Medición sobre carga de 100 pF.....	98
4.3.	Medición sobre carga de 56 Ω en serie con una carga de 82 pF.....	99
5.	Conclusiones y trabajo futuro	100
5.1.	Conclusiones.....	100
5.2.	Trabajo futuro.....	100
	Bibliografía	102

Índice de Figuras

1.1. Representación gráfica en forma rectangular de la impedancia.....	10
1.2. Circuito método del puente.....	11
1.3. Circuito método resonante.....	12
1.4. Circuito método I – V.....	13
1.5. Circuito método RF I – V baja impedancia.....	14
1.6. Circuito método RF I – V alta impedancia.....	14
1.7. Circuito método de analizador de red.....	15
1.8. Circuito método del puente auto balanceado.....	16
2.2. Diagrama en bloques del sistema final a implementar.....	20
2.3. Diagrama de flujo del código funcional del PIC18F2550.....	21
2.4. Registro de control del oscilador.....	22
2.5. Registro de configuración CONFIG1L.....	23
2.6. Registro de configuración CONFIG1H.....	23
2.7. Trayecto de la señal de reloj.....	24
2.8. Configuración de oscilador externo como oscilador primario.....	24
2.9. Selección de componentes para el oscilador primario.....	25
2.10. Tipos de osciladores.....	25
2.11. Estructura interna de interrupción del microcontrolador.....	27
2.12. Trayecto de la señal para poder generar la interrupción del PIC18F2550	28
2.13. Registro de control de interrupción.....	29
2.14. Registro de control de reinicio.....	30
2.15. Registro de habilitación de interrupción de periféricos.....	31
2.16. Registro de bandera de interrupción de periféricos.....	32
2.17. Diagrama en bloques del procedimiento de transmisión de la UART.....	33
2.18. Registro de control y estado de transmisión.....	34

2.19. Diagrama en bloques del procedimiento de recepción de la UART.....	34
2.20. Registro de control y estado de recepción.....	35
2.21. Configuración de tasa de baudios.....	36
2.22. Diagrama en bloques del modo I2C.....	37
2.23. Registro de control SSPCON1.....	39
2.24. Registro de estado SSPSTAT.....	39
2.25. Registro de control SSPCON2.....	42
2.26. Diagrama en bloques de los distintos componentes del Si5351.....	43
2.27. Diagrama en bloques de la estructura interna del Si5351.....	44
2.28. Procedimiento de programación I2C del Si5351.....	45
2.29. Registro de verificación de estado del dispositivo.....	46
2.30. Registro de control de habilitación de salida.....	46
2.31. Registro de control de CLK0.....	47
2.32. Registro de control de CLK1.....	48
2.33. Registro de configuración de capacitancia.....	48
2.34. Registro de fuente de entrada para el PLL.....	49
2.35. Registro de control de CLK0.....	50
2.36. Registro de control de CLK1.....	52
2.37. Registro de reinicio del PLL.....	53
2.38. Registro de habilitación de salida.....	53
3.1. Diagrama en bloques del sistema completo.....	54
3.2. Esquemático de la placa de adquisición del PIC18F2550.....	55
3.3. Características del PIC18F2550.....	55
3.4. Diseño del hardware de la placa de adquisición del PIC18F2550.....	56
3.5. Placa de adquisición del PIC18F2550.....	57
3.6. Señal de salida (CLK0) de 1 MHz.....	58
3.7. Espectro de la señal de salida (CLK0) de 1 MHz.....	58
3.8. Diseño de filtro Butterworth de 5to orden.....	60
3.9. Respuesta en frecuencia del filtro de RF en magnitud.....	61
3.10. Respuesta en frecuencia del filtro de RF en fase.....	61

3.11. Diseño del esquemático de los filtros de RF.....	62
3.12. Diseño del pcb de los filtros de RF.....	62
3.13. Filtros de RF.....	62
3.14. Parámetros “S” del primer filtro de RF.....	63
3.15. Parámetros “S” del segundo filtro de RF.....	64
3.16. Comparación de la respuesta en frecuencia y fase de ambos filtros.....	65
3.17. Señal de 30 MHz a la entrada del filtro de RF.....	66
3.18. Señal de 30 MHz a la salida del filtro de RF.....	66
3.19. Contenido espectral de la señal de 30 MHz a la entrada del filtro de RF.....	67
3.20. Contenido espectral de la señal de 30 MHz a la salida del filtro de RF.....	68
3.21. Esquemático de la primera etapa de RF del medidor de impedancias.....	69
3.22. Diagrama en bloques del SA612A.....	70
3.23. Circuito equivalente del SA612A.....	71
3.24. Sistema de medición de impedancia.....	72
3.25. Punto de intercepción de tercer orden y compresión.....	73
3.26. Esquemático del modelo de adaptación de impedancias.....	75
3.27. Impedancia equivalente del bloque Zeq.....	75
3.28. Determinación del valor de la Zeq.....	76
3.29. Simulación para obtener la Zeq.....	77
3.30. Respuesta en frecuencia de la impedancia de entrada del SA612A.....	77
3.31. Cálculo de los valores de la red T.....	78
3.32. Esquemático del puente de medición de impedancias del SA612A.....	79
3.33. Simulación del puente de medición de impedancias.....	80
3.34. Sistema de medición de impedancia incógnita.....	81
3.35. Protecciones del integrado SA612A.....	82
3.36. Esquemático de la segunda etapa de audio del medidor de impedancias.....	83
3.37. Respuesta del filtro elíptico de 8vo orden MAX7400.....	83
3.38. Esquema de configuración del MAX7400.....	84
3.39. Nivel de voltaje máximo para la señal de audio en la entrada de línea.....	85
3.40. Características eléctricas del diodo 1N5819.....	85
3.41. Señal de RF a la entrada del filtro MAX7400.....	86

3.42. Señal de audio a la salida del filtro MAX7400.....	86
3.43. Diseño del pcb de la capa superior del medidor de impedancias.....	87
3.44. Rutas donde viajan las señales de RF.....	88
3.45. Diseño de pista de 50 Ω	89
3.46. Distancia entre dos rutas adyacentes.....	90
3.47. Diseño de pcb de la capa inferior del medidor de impedancias.....	91
3.48. Medidor de impedancias capa superior.....	91
3.49. Medidor de impedancias capa inferior.....	92
3.50. Diseño de la interfaz gráfica de usuario.....	93
3.51. Diagrama funcional de la GUI.....	94
4.1. Banco de medición del sistema completo.....	95
4.2. Procedimiento de medición para una carga de 47 Ω	96
4.3. Magnitud de la impedancia de 47 Ω versus frecuencia.....	97
4.4. Fase de la impedancia de 47 Ω versus frecuencia.....	97
4.5. Magnitud de la impedancia del capacitor de 100 pF versus frecuencia.....	98
4.6. Fase de la impedancia del capacitor de 100 pF versus frecuencia.....	98
4.7. Magnitud de la carga de 56 Ω en serie con carga una de 82 pF versus frecuencia.....	99
4.8. Fase de la carga de 56 Ω en serie con una carga de 82 pF versus frecuencia.....	99

Capítulo 1

Introducción

1.1. Origen del problema

La impedancia es un parámetro de suma importancia a la hora de evaluar el funcionamiento de un componente o circuito. Hoy en día existen diversos circuitos electrónicos que trabajan en alta frecuencia. Cada uno de ellos está formado por elementos tales como: bobinas, capacitores y resistencias. Conocer la impedancia de un componente en un rango de frecuencia, da indicios de la calidad de fabricación del componente y del comportamiento que exhibe. Por ejemplo, en los filtros pasivos es crucial que exhiban el comportamiento deseado. Esto no sucede a menudo debido a las propiedades parásitas de los componentes asociados. Otro ejemplo importante es la impedancia característica de un circuito de RF y la adaptación de impedancias. Es importante que el circuito este adaptado para que no haya onda reflejada. También, otro ejemplo son los cables. Es importante determinar la impedancia característica de estos, ya que, por lo general, cualquier circuito de RF que conecte distintos elementos mediante cables, tendrá un cambio en su impedancia. Por último, existen en las comunicaciones, osciladores como así también sintonizadores de radio cuyo buen funcionamiento dependerá en parte de la calidad del capacitor variable.

1.2. Definición de Impedancia

La impedancia se define como la oposición a la CA (corriente alterna) a una frecuencia dada cuando pasa a través de un dispositivo o circuito electrónico. La impedancia se utiliza para caracterizar componentes y circuitos electrónicos. En la figura 1.1, se observa la representación gráfica en forma rectangular de la impedancia.

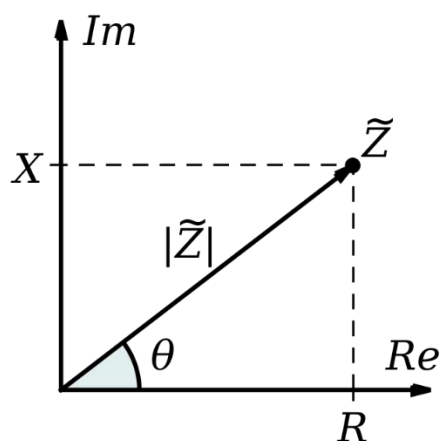


Figura 1.1: Representación gráfica en forma rectangular de la impedancia

De la figura 1.1, se puede obtener la ecuación rectangular que representa a la impedancia. Esta se representa en la siguiente ecuación:

$$\tilde{Z} [\Omega] = R [\Omega] + j \times X [\Omega]$$

Donde R y X son la parte real e imaginaria respectivamente de la impedancia. R representa la resistencia y X la reactancia de la impedancia. También se puede representar la misma en forma polar, esto se logra mediante la siguiente ecuación:

$$\tilde{Z} = |Z| \angle \theta$$

Donde |Z| representa la relación, en módulo, entre la amplitud de la tensión aplicada al circuito y la corriente que fluye del mismo. Esto último, se lo puede representar en la siguiente ecuación:

$$|Z| = \frac{|V|}{|I|}$$

A su vez, θ representa el desfase entre la tensión aplicada al circuito y la corriente que fluye del mismo. El ángulo θ se lo puede representar mediante la siguiente ecuación:

$$\theta = \theta_V - \theta_I$$

Si se tiene como dato del gráfico rectangular R y X, se puede obtener |Z| y θ a través de las siguientes ecuaciones:

$$|Z| = \sqrt{R^2 + X^2} \quad [\Omega]$$

$$\theta = \tan^{-1} \left(\frac{X}{R} \right) \quad [^\circ]$$

La naturaleza de la reactancia X puede ser inductiva o capacitiva. Ambas, se pueden calcular según las siguientes ecuaciones:

$$X_L [\Omega] = 2 \times \pi * f [\text{Hz}] \times L [\text{Hy}]$$

$$X_C [\Omega] = \frac{1}{2 \times \pi \times f [\text{Hz}] \times C [\text{F}]}$$

Dónde:

- X_L representa la reactancia inductiva, f es la frecuencia de la señal de tensión aplicada y L es el valor de la inductancia.
- X_C representa la reactancia capacitiva, f es la frecuencia de la señal de tensión aplicada y C es el valor de la capacitancia.

1.3. Métodos de Medición de Impedancia

Existen diferentes métodos de medición de impedancia basados en varios parámetros, como la frecuencia de operación, la precisión de la medición, el rango de medición y la facilidad de realizar la medición. Estos métodos se los conocen como: método de puente, método resonante, método I-V, método RF I-V, método de análisis de red y método de puente de equilibrio automático. Además de estos, el medidor LCR se utiliza para medir la impedancia.

A continuación, se detallarán los métodos de medición de impedancia y sus diferencias con respecto a la frecuencia, ventajas y desventajas. Según lo siguiente, se pueden seleccionar las técnicas de medición de impedancia de elección para la medición de baja y alta frecuencia.

1.3.1. Método del Puente

En la figura 1.1, se puede observar el circuito que representa el método del puente para el cálculo de la impedancia incógnita Z_x . Cuando no circula corriente por el detector "D", el valor de la impedancia Z_x se puede obtener mediante la siguiente ecuación:

$$Z_x = \frac{Z_1}{Z_2} \times Z_3$$

Las ventajas de usar un medidor de impedancias basado en este método son:

- Buena exactitud en la medición (0,1% típico)
- Gran ancho de banda (0 a 300 MHz) si se usan diferentes tipos de puente
- Bajo costo

Las desventajas de usar un medidor de impedancias basado en este método son:

- Se necesita balancear el puente de forma manual
- Si se usa un solo instrumento el ancho de banda se ve reducido

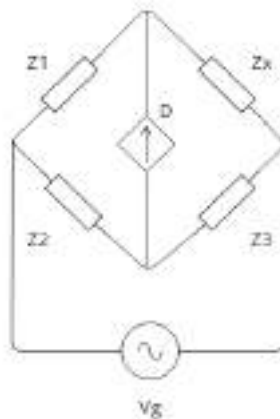


Figura 1.2: Circuito método del puente

1.3.2. Método resonante

En la figura 1.3, se puede observar el circuito que representa el método resonante. Se ajusta el capacitor variable para obtener la frecuencia de resonancia de la impedancia incógnita ($L_x + R_x$). El factor de calidad "Q" del capacitor variable se obtiene a través del voltímetro conectado en sus extremos. Se pueden obtener valores de "Q" de hasta 300. El rango del instrumento es de 10 kHz hasta 70 MHz.

Las ventajas de usar un medidor de impedancias basado en este método son:

- Buen factor de calidad Q

Las desventajas de usar un medidor de impedancias basado en este método son:

- Necesita ser sintonizado hasta la frecuencia de resonancia
- Precisión de medición de baja impedancia

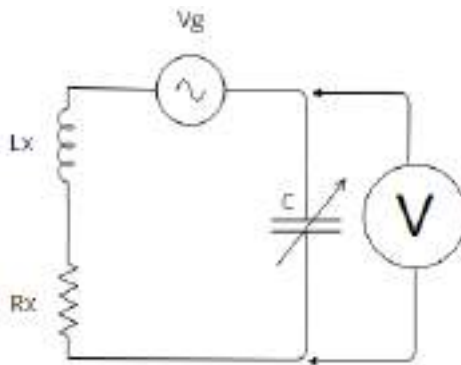


Figura 1.3: Circuito método resonante

1.3.3. Método I – V

En la figura 1.4, se puede observar el circuito que representa el método I – V. La impedancia Z_x se puede determinar mediante el valor de voltaje que sensa el voltímetro “V1” y la corriente “I” que circula por el circuito. Para ello, se coloca una resistencia de valor muy chica en comparación con la impedancia a medir. De esta forma, se puede determinar la corriente que circula en el circuito. En la práctica se suele utilizar un transformador en vez de una resistencia en serie “R”, para prevenir los efectos que causa colocar una resistencia en el circuito. Sin embargo, esto último trae aparejado una limitación en el rango de frecuencia. La impedancia Z_x puede determinarse mediante la siguiente ecuación:

$$Z_x = \frac{V_1}{I} = \frac{V_1}{V_2} \times R$$

Las ventajas de usar un medidor de impedancias basado en este método son:

- Permite la medición de dispositivos conectados a tierra

Las desventajas de usar un medidor de impedancias basado en este método son:

- El rango de frecuencia del instrumento si se usa un transformador en vez de R1. Este rango quedaría acotado de 10 kHz hasta 100 MHz.

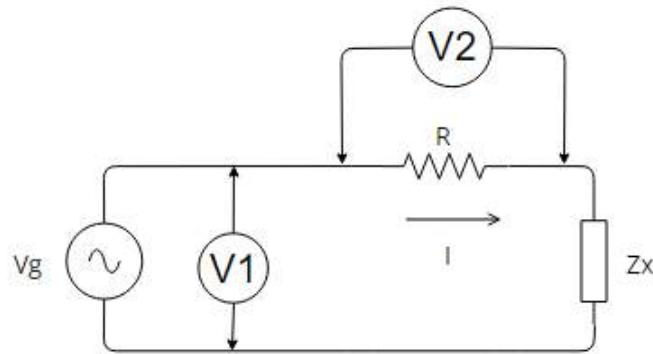


Figura 1.4: Circuito método I – V

1.3.4. Método RF I – V

Este método utiliza el mismo principio básico de medición que el método I-V. Permite medir la impedancia de alta frecuencia mediante el uso de un circuito que adapta la impedancia característica de un cable coaxial de alta frecuencia (50Ω) y un conector coaxial de alta frecuencia (50Ω). Hay dos tipos de circuitos, uno representa el medidor de tensión y el otro el medidor de corriente. El primero se usa para medir impedancias bajas y el segundo para medir impedancias altas. En las figuras 1.5 y 1.6, se puede observar el primero y segundo circuito respectivamente. Para cada uno de ellos, se puede determinar la impedancia Z_x mediante las siguientes ecuaciones:

$$Z_x \text{ (baja impedancia)} = \frac{2R}{\frac{V_2}{V_1} - 1}$$

$$Z_x \text{ (baja impedancia)} = \frac{R}{2} \times \left(\frac{V_1}{V_2} - 1 \right)$$

Las ventajas de usar un medidor de impedancias basado en este método son:

- Buena exactitud de medición (1 % típico)
- Gran ancho de banda (1 MHz a 3 GHz)

La desventaja de usar un medidor de impedancias basado en este método es:

- Ancho de banda limitado si se usa un transformador de baja pérdida en reemplazo de la resistencia R

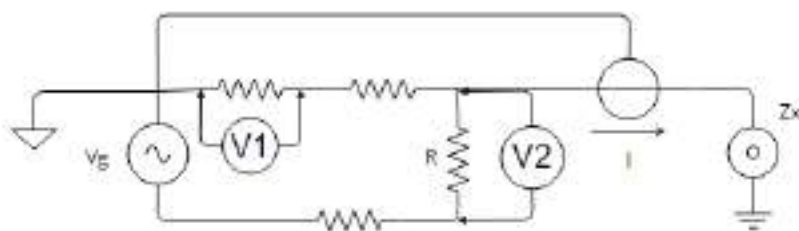


Figura 1.5: Circuito método RF I – V baja impedancia

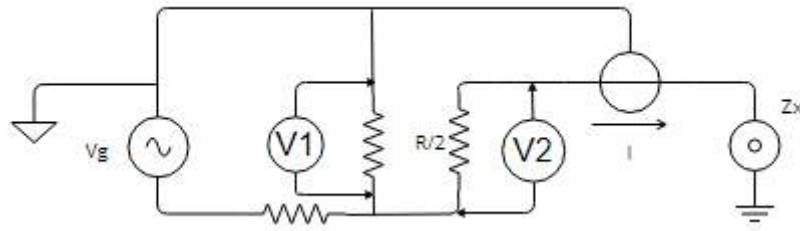


Figura 1.6: Circuito método RF I – V alta impedancia

1.3.5. Método del analizador de red

En la figura 1.7, se puede observar el circuito que representa el método de analizador de red. Este método se basa en la obtención del coeficiente de reflexión. Mide la relación entre la señal incidente y la señal reflejada. Usa un acoplador direccional o puente para detectar la señal reflejada y el analizador de red es usado para suministrar y medir las señales.

Las ventajas de usar un medidor de impedancias basado en este método son:

- Amplio rango de cobertura de frecuencia. Funciona en baja frecuencia en el orden de los Hertz y en alta frecuencia en el orden de los mega Hertz
- Buena exactitud de medición cuando la impedancia incógnita está cerca de la impedancia característica

Las desventajas de usar un medidor de impedancias basado en este método son:

- Es necesario recalibrar cuando cambia la frecuencia de barrido
- El rango de medición de impedancia es angosto

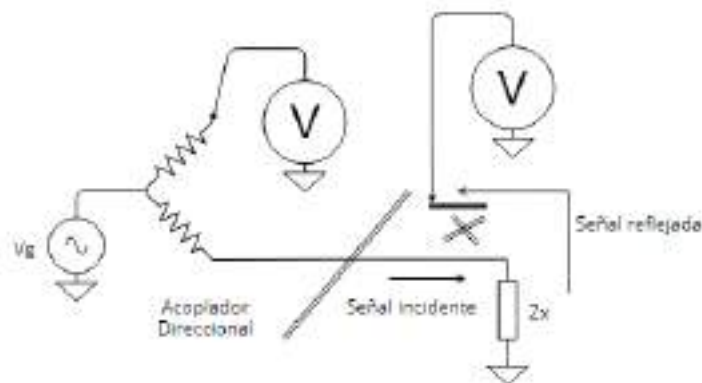


Figura 1.7: Circuito método de analizador de red

1.3.6. Método del puente auto balanceado

En la figura 1.8, se puede observar el circuito que representa el método del puente auto balanceado. El método de puente de auto balanceado se utiliza comúnmente en los instrumentos modernos de medición de impedancia de baja frecuencia.

Su rango de frecuencia operativa se ha ampliado hasta 120 MHz. Básicamente, para medir la impedancia compleja del DUT es necesario medir el voltaje de la señal de prueba aplicada al DUT y la corriente que fluye a través de él.

Los instrumentos de puente de auto balanceado para la medición de impedancia de baja frecuencia (por debajo de 100 kHz) generalmente emplean un circuito convertidor I - V simple (un amplificador operacional con un circuito de retroalimentación negativa)

1.3.6.1 Funcionamiento

La corriente de la señal de prueba (I_x) circula a través del dispositivo bajo prueba y también circula hacia el convertidor I - V. El amplificador operacional del convertidor I - V hace que la misma corriente que I_x circule a través de la resistencia (R_r) en el circuito de retroalimentación negativa. Dado que la corriente de retroalimentación (I_r) es igual a la corriente de entrada (I_x), esta circula a través de R_r y el potencial en el punto denotado como bajo (Low) se conduce automáticamente a cero voltios. Por lo tanto, se lo llama masa virtual. El voltaje de salida del convertidor IV (V_r) está representado por la siguiente ecuación:

$$(1) V_r = I_r \times R_r = I_x \times R_r$$

De donde,

$$(2) I_x = \frac{V_x}{Z_x}$$

Si se reemplaza (1) en (2) y se despeja Z_x , se obtiene:

$$Z_x = \frac{V_x}{I_x} \times R_r = \frac{V_x}{V_r} R_r$$

Las ventajas de usar un medidor de impedancias basado en este método son:

- Cubre una amplia gama de frecuencias (20 Hz a 120 MHz)
- Proporciona un amplio rango de medición de impedancia (Es utilizado por muchos medidores LCR)

Las desventajas de usar un medidor de impedancias basado en este método son:

- La exactitud de medición de impedancia en alta frecuencia no es buena debido a la limitación que exhibe el amplificador I - V.

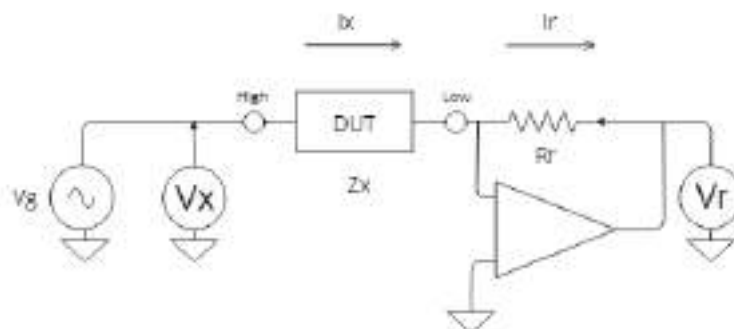


Figura 1.8: Circuito método del puente auto balanceado

1.4. Objetivos

1.4.1 Objetivo general

Diseñar un instrumento que pueda medir la magnitud y fase de una impedancia de forma precisa. El mismo deberá contar con una interfaz gráfica de usuario para visualizar la magnitud y fase versus la frecuencia. A su vez, deberá poder controlar mediante la interfaz el barrido de frecuencia del instrumento. Por último, deberá tener la opción de calibrar el instrumento con la selección de algún valor de referencia.

1.4.2 Objetivos de desarrollo

Desarrollar todos los componentes que componen el firmware del sistema. Esto consta de:

- Configurar la señal de reloj del microcontrolador PIC18F2550 haciendo uso del IDE MPLABX y el compilador XC8.
- Configurar el módulo I2C para controlar el Si5351.
- Configurar el módulo RS232 para hacer uso de interrupción por llegada de caracteres.
- Configurar los registros de programación del Si5351 para obtener la frecuencia de señal de reloj deseada.

Desarrollar todos los componentes que componen el hardware del sistema. Esto consta de:

- Dos filtros pasa bajos para obtener la armónica fundamental de la señal de reloj.
- Dos redes de atenuación del tipo "T" para adecuar los niveles de potencia de RF de la señal de reloj.
- Dos mezcladores NE612AN para sensar la tensión y corriente de la impedancia conectada.
- Dos filtros de FI para obtener las dos señales de 1kHz.
- Una placa de sonido para proteger la entrada de audio del PC.

Desarrollar todos los componentes que componen el software del sistema. Esto consta de:

- Configurar la interfaz de usuario para que pueda enviar la frecuencia deseada mediante el puerto serie del pc al microcontrolador.
- Configurar la interfaz de usuario para que pueda decodificar las dos señales provenientes de la placa de sonido USB.
- Configurar la interfaz de usuario para que pueda realizar la calibración del instrumento en el rango de frecuencia deseado.

Capítulo 2

Diseño del firmware del sistema

En este proyecto se realizará un medidor de impedancias para alta frecuencia. Este medidor será capaz de caracterizar una impedancia incógnita en magnitud y fase en el rango de 30 a 50 MHz. Para poder llevar esto a cabo, se deberá medir la corriente y la tensión de la impedancia que se conecte. Con esto, se podrá determinar el valor de la impedancia. Para poder determinar que valor tiene la impedancia en amplitud y fase, se deberá usar un generador que pueda generar una señal del tipo sinusoidal. La señal de este generador deberá ser de frecuencia variable, en el rango de 30 a 50 MHz, para poder aplicarse a la impedancia conectada. Una vez establecido esto último, se deberá usar un dispositivo que funcione como “voltímetro” y otro que funcione como “amperímetro”. Estos medidores deberán tener un ancho de banda que cumpla con el rango de frecuencias del proyecto para poder medir la impedancia correctamente. Adicionalmente, una vez que se miden las dos señales de RF, se deberá realizar una conversión de estas dos señales a un nivel de audio (1 kHz), para que se puedan procesar mediante la entrada de audio de una PC. Durante esta conversión, será necesario que las dos señales no pierdan su característica de módulo y fase. Por último, se deberá realizar una interfaz visual de usuario que muestre una gráfica de la magnitud y fase de la impedancia versus la frecuencia. En la figura 2.1, se puede observar un diagrama en bloques del sistema genérico a implementar.

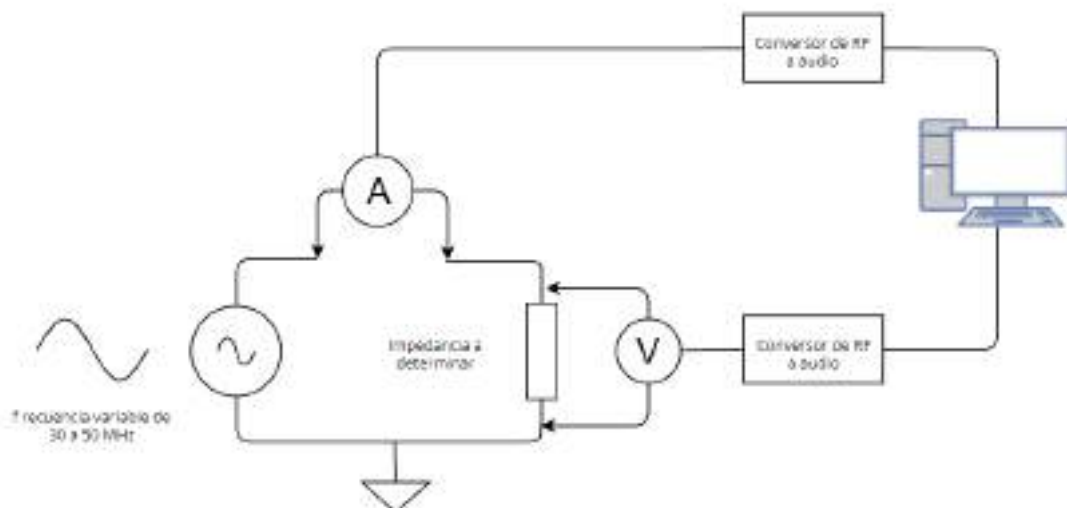


Figura 2.1: Diagrama en bloques del sistema genérico a implementar

En la figura 2.2, se puede observar el diagrama en bloques del sistema final a implementar. Este diagrama se diseñó a partir del diagrama de la figura 2.1, para poder satisfacer las funciones que deben cumplir los componentes de dicha figura. Se seleccionaron los siguientes componentes principales:

- La señal de frecuencia variable se generará mediante el generador de señal Si5351. La ventaja de usar este generador es que posee tres salidas. Para este proyecto, se necesitará una salida adicional para que se pueda usar como señal de oscilador local para los mezcladores SA612.
- Como voltímetro y amperímetro se usará el mezclador SA612A. A su vez, cada uno cumplirá la función de convertor de RF a audio.
- Se usarán dos filtros de audio para obtener las señales de audio de corriente y de tensión. Esto es necesario porque los mezcladores SA612A generan una señal de RF y una señal de audio a la salida.

A continuación, se llevará a cabo una explicación más detallada a nivel técnico del funcionamiento de los distintos bloques que componen la figura 2.2:

- Mediante la interfaz visual de usuario se selecciona la frecuencia deseada con la que se quiere caracterizar la impedancia "Zx" de la figura.
- La frecuencia deseada se recibe mediante una interrupción de caracteres en el PIC18F2550.
- Se programa la frecuencia de la señal de onda cuadrada de las dos salidas CLK0 y CLK1 del Si5351. CLK0 se programa con la frecuencia que se recibió y CLK1 con la frecuencia que se recibió más 1 kHz.
- Las dos señales del Si5351 son filtradas y se obtienen dos señales sinusoidales.
- Se reducen en amplitud y se adaptan las dos señales provenientes del filtrado para el correcto funcionamiento de los integrados SA612A.
- Los mezcladores SA612A se encargan de medir una tensión y corriente proporcional a la tensión y corriente aplicada en la impedancia incógnita Zx. Mediante estos dos parámetros se calcula el valor de amplitud y fase de la Zx.
- Los filtros de FI se encargan de filtrar la señal de salida de los mezcladores para quedarse únicamente con la señal de audio (1kHz).
- Los diodos de protección se usan para proteger el nivel máximo soportado por la entrada de línea de la placa de sonido USB.
- La placa de sonido USB se usa para proteger la PC de una posible ruptura por sobrevoltaje. A su vez, se encarga de realizar la conversión analógica digital con una frecuencia de muestreo de 48 kHz.

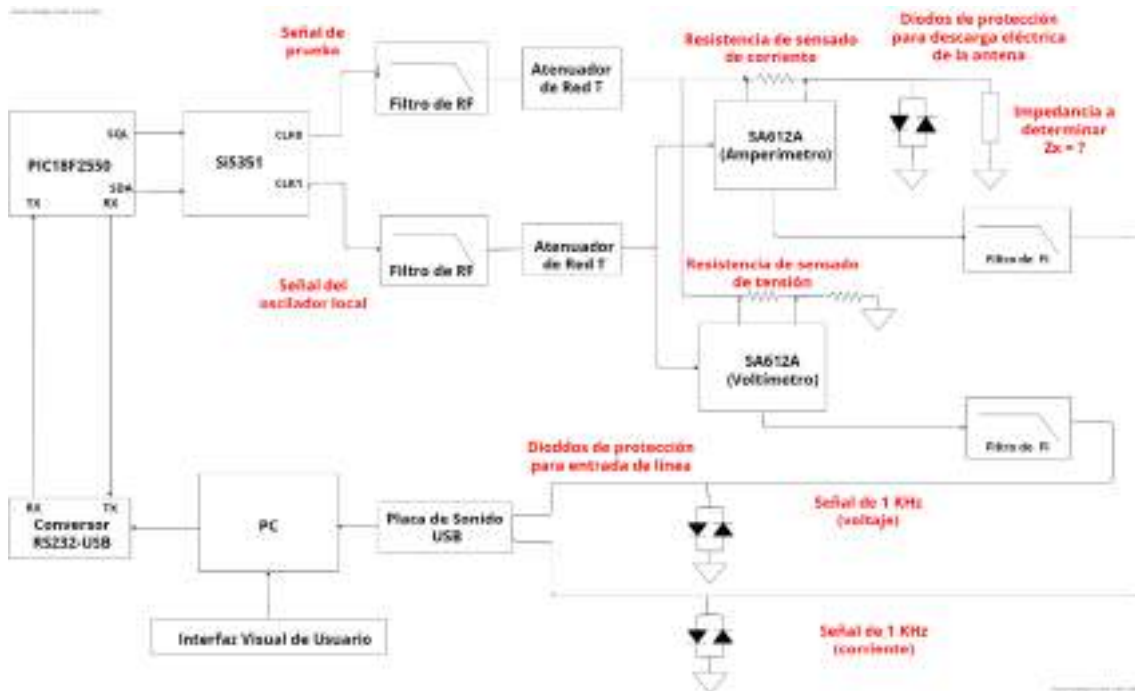


Figura 2.2: Diagrama en bloques del sistema final a implementar

2.1. Introducción

En este capítulo se abordará el diseño del código funcional para el microcontrolador PIC18F2550 y el generador de señal de reloj Si5351. Se demostrará paso a paso como configurar cada registro asociado al código funcional. El firmware se realizó con el IDE MPLABX y con el compilador XC8. Adicionalmente, se detallará el procedimiento de programación de los registros del Si5351 para obtener la frecuencia de la señal de reloj deseada. En la figura 2.3, se puede observar el diagrama de flujo del código funcional del PIC.

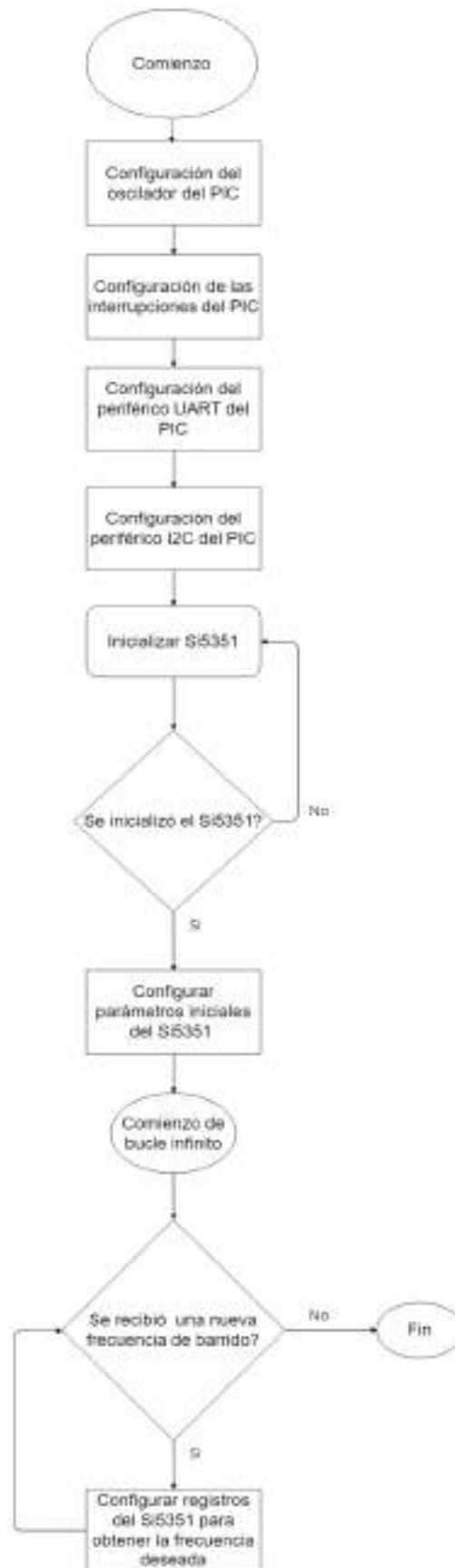


Figura 2.3: Diagrama de flujo del código funcional del PIC 18F2550

2.2. Configuración de los registros del PIC

2.2.1. Oscilador

El funcionamiento del oscilador del PIC18F2550 es controlado por registros de configuración y registros de control. Los registros de configuración, CONFIG1L y CONFIG1H, seleccionan el modo de oscilador y las opciones del USB prescaler/postscaler. Estos parámetros conocidos como bits de configuración son seteados en el momento en que la memoria de programa del PIC es grabada con un programador y quedan guardados hasta que el PIC es reprogramado.

El registro OSCCON permite seleccionar el modo activo de oscilador. Dicho registro es usado ante todo para controlar el cambio de fuente de reloj en los modos de alimentación controlada (son 7 modos, los cuales proveen una gran variedad de opciones de alimentación del PIC para conservar la energía).

El registro OSCTUNE es usado para afinar la fuente de reloj del reloj interno del PIC. No será explicado en el presente escrito, ya que, se utilizará como fuente de reloj principal del PIC un oscilador externo.

En la figura 2.4, se puede observar la configuración del registro OSCCON.

REGISTER 2-2: OSCCON: OSCILLATOR CONTROL REGISTER

RW-0	RAW-1	RW-0	RAW-0	R ¹¹	R-0	RW-0	RW-0
IDLEN	IRCF2	IRCF1	IRCF0	OSTS	IOFS	SCS1	SCS0
bit 7							bit 0

Legend:
 R = Readable bit W = Writable bit U = Unimplemented bit, read as '0'
 -n = Value at POR '1' = Bit is set '0' = Bit is cleared x = Bit is unknown

bit 7 **IDLEN:** Idle Enable bit
 1 = Device enters Idle mode on SLEEP instruction
 0 = Device enters Sleep mode on SLEEP instruction

bit 6-4 **IRCF2:IRCF0:** Internal Oscillator Frequency Select bits
 111 = 8 MHz (INTOSC drives clock directly)
 110 = 4 MHz
 101 = 2 MHz
 100 = 1 MHz⁽¹⁾
 011 = 500 kHz
 010 = 250 kHz
 001 = 125 kHz
 000 = 31 kHz (from either INTOSC/256 or INTRC directly)⁽²⁾

bit 3 **OSTS:** Oscillator Start-up Time-out Status bit⁽¹⁾
 1 = Oscillator Start-up Timer time-out has expired; primary oscillator is running
 0 = Oscillator Start-up Timer time-out is running; primary oscillator is not ready

bit 2 **IOFS:** INTOSC Frequency Stable bit
 1 = INTOSC frequency is stable
 0 = INTOSC frequency is not stable

bit 1-0 **SCS1:SCS0:** System Clock Select bits
 1x = Internal oscillator
 01 = Timer1 oscillator
 00 = Primary oscillator

Note: 1: Depends on the state of the IESO Configuration bit.
 2: Source selected by the INTSRC bit (OSCTUNE<7>), see text.
 3: Default output frequency of INTOSC on Reset.

Figura 2.4: Registro de control del oscilador

En las figuras 2.5 y 2.6, se puede observar la configuración de los registros CONFIG1L y CONFIG1H. Adicionalmente, si se analizan los bits de configuración FOSC3-FOSC0 de la figura 2.6, se puede observar que existen doce modos de configuración para el tipo de oscilador.

REGISTER 25-1: CONFIG1L: CONFIGURATION REGISTER 1 LOW (BYTE ADDRESS 300000h)

U-0	U-0	R/P-0	R/P-0	R/P-0	R/P-0	R/P-0	R/P-0
—	—	USBIV	CPUDIV1	CPUDIV0	PLLDIV2	PLLDIV1	PLLDIV0
bit 7							bit 0

Legend:

R = Readable bit P = Programmable bit U = Unimplemented bit, read as '0'
 -n = Value when device is unprogrammed u = Unchanged from programmed state

bit 7-6 **Unimplemented:** Read as '0'

bit 5 **USBIV:** USB Clock Selection bit (used in Full-Speed USB mode only; UCFG:FSEN = 1)
 1 = USB clock source comes from the 96 MHz PLL divided by 2
 0 = USB clock source comes directly from the primary oscillator block with no postscale

bit 4-3 **CPUDIV1:CPUDIV0:** System Clock Postscale Selection bits
For XT, HS, EC and ECIO Oscillator modes:
 11 = Primary oscillator divided by 4 to derive system clock
 10 = Primary oscillator divided by 3 to derive system clock
 01 = Primary oscillator divided by 2 to derive system clock
 00 = Primary oscillator used directly for system clock (no postscale)
For XTPLL, HSPLL, ECPLL and ECPIO Oscillator modes:
 11 = 96 MHz PLL divided by 5 to derive system clock
 10 = 96 MHz PLL divided by 4 to derive system clock
 01 = 96 MHz PLL divided by 3 to derive system clock
 00 = 96 MHz PLL divided by 2 to derive system clock

bit 2-0 **PLLDIV2:PLLDIV0:** PLL Prescaler Selection bits
 111 = Divide by 12 (48 MHz oscillator input)
 110 = Divide by 10 (40 MHz oscillator input)
 101 = Divide by 8 (24 MHz oscillator input)
 100 = Divide by 5 (20 MHz oscillator input)
 011 = Divide by 4 (16 MHz oscillator input)
 010 = Divide by 3 (12 MHz oscillator input)
 001 = Divide by 2 (8 MHz oscillator input)
 000 = No prescale (4 MHz oscillator input drives PLL directly)

Figura 2.5: Registro de configuración CONFIG1L

REGISTER 25-2: CONFIG1H: CONFIGURATION REGISTER 1 HIGH (BYTE ADDRESS 300001h)

R/P-0	R/P-0	U-0	U-0	R/P-0	R/P-1	R/P-0	R/P-1
IESO	FCMEN	—	—	FOSC3 ⁽¹⁾	FOSC2 ⁽¹⁾	FOSC1 ⁽¹⁾	FOSC0 ⁽¹⁾
bit 7							bit 0

Legend:

R = Readable bit P = Programmable bit U = Unimplemented bit, read as '0'
 -n = Value when device is unprogrammed u = Unchanged from programmed state

bit 7 **IESO:** Internal/External Oscillator Switchover bit
 1 = Oscillator Switchover mode enabled
 0 = Oscillator Switchover mode disabled

bit 6 **FCMEN:** Fail-Safe Clock Monitor Enable bit
 1 = Fail-Safe Clock Monitor enabled
 0 = Fail-Safe Clock Monitor disabled

bit 5-4 **Unimplemented:** Read as '0'

bit 3-0 **FOSC3:FOSC0:** Oscillator Selection bits⁽¹⁾
 111x = HS oscillator, PLL enabled (HSPLL)
 110x = HS oscillator (HS)
 1011 = Internal oscillator, HS oscillator used by USB (INTHS)
 1010 = Internal oscillator, XT used by USB (INTXT)
 1001 = Internal oscillator, CLKO function on RA6, EC used by USB (INTCKO)
 1000 = Internal oscillator, port function on RA6, EC used by USB (INTIO)
 0111 = EC oscillator, PLL enabled, CLKO function on RA6 (ECPLL)
 0110 = EC oscillator, PLL enabled, port function on RA6 (ECPIO)
 0101 = EC oscillator, CLKO function on RA6 (EC)
 0100 = EC oscillator, port function on RA6 (ECIO)
 001x = XT oscillator, PLL enabled (XTPLL)
 000x = XT oscillator (XT)

Nota 1: The microcontroller and USB module both use the selected oscillator as their clock source in XT, HS and EC modes. The USB module uses the indicated XT, HS or EC oscillator as its clock source whenever the microcontroller uses the internal oscillator.

Figura 2.6: Registro de configuración CONFIG1H

En la figura 2.7, se puede observar el diagrama de la señal de reloj para la configuración correcta del PIC. Se debe trazar el camino correcto de forma tal que quede bien configurada la señal de reloj que le llega a la CPU y los periféricos.

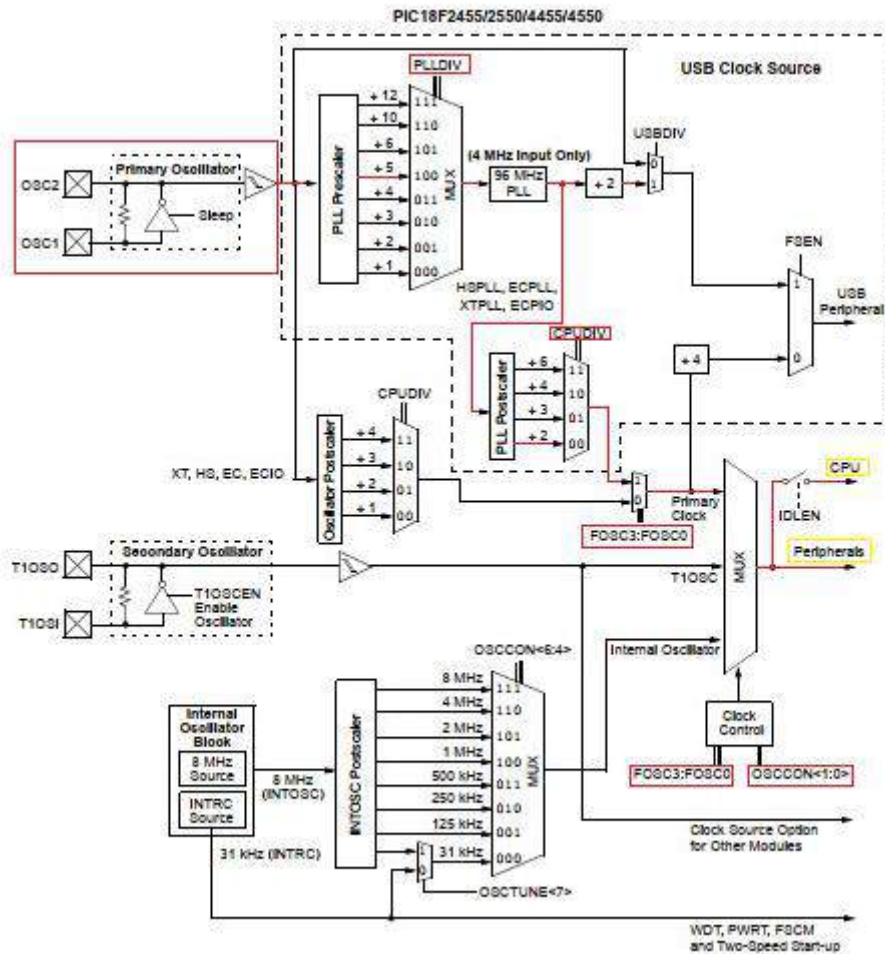


Figura 2.7: Trayecto de la señal de reloj

A partir de la figura 2.7, se debe configurar cada registro marcado en rojo. El primer bloque que se debe tener en cuenta es el oscilador primario. Este se compone de un oscilador a cristal o resonador cerámico. El cristal debe colocarse de forma externa en conjunto con dos capacitores a masa. En la figura 2.8, se puede observar esto último.

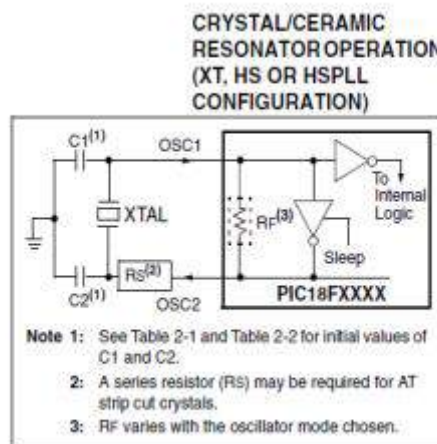


Figura 2.8: Configuración de oscilador externo como oscilador primario

En la figura 2.9, se puede observar los valores de los capacitores C1 y C2 para una correcta oscilación del oscilador primario.

Osc Type	Crystal Freq	Typical Capacitor Values Tested:	
		C1	C2
XT	4 MHz	27 pF	27 pF
HS	4 MHz	27 pF	27 pF
	8 MHz	22 pF	22 pF
	20 MHz	15 pF	15 pF

Figura 2.9: Selección de componentes para el oscilador primario

En este proyecto se optó por utilizar un cristal de 20 MHz con sus respectivos capacitores C1 y C2 acorde a la figura 2.9.

El oscilador primario se configura mediante firmware escribiendo en el registro OSCCON en formato binario de la siguiente forma:

$$(1) \text{ OSCCON} = \text{xxxxxx00}$$

Esta configuración de la ecuación (1) se puede ver en la figura 2.4 en los dos bits menos significativos. Si se configura de acuerdo con (1), el multiplexor (MUX) selecciona la fuente de reloj primaria. Una vez configurado el reloj como fuente de reloj primaria, se debe optar por elegir el tipo de reloj. En la figura 2.10, se puede ver los tipos de osciladores configurables. Cabe destacar que cuando se usa este tipo de oscilador las posibles configuraciones de este son: XT, XTPLL, HS y HSPLL.

Nº	Fuente de reloj	NOMBRE	TIPO DE OSCILADOR/ES
1	Cristal o Resonador	XT	Crystal/Resonator
2	Cristal o Resonador	XTPLL	Crystal/Resonator con PLL activado
3	Cristal o Resonador	HS	High-Speed Crystal/Resonator
4	Cristal o Resonador	HSPLL	High-Speed Crystal/Resonator con PLL activado
5	Señal externa	EC	External Clock como fuente de clock para la CPU con salida de Frecuencia/4 en RA6
6	Señal externa	ECIO	External Clock como fuente de clock para la CPU con pin de Input/Output en RA6
7	Señal externa	ECPLL	External Clock con PLL activado con salida de Frecuencia/4 en RA6
8	Señal externa	ECPIO	External Clock con PLL activado, con pin de Input/Output en RA6
9	Oscilador Interno y Cristal	INTHS	Internal Oscillator como fuente de clock para la CPU, oscilador HS como fuente de oscilador para el módulo USB
10	Oscilador Interno y Cristal	INTXT	Internal Oscillator como fuente de clock para la CPU, oscilador XT como fuente de oscilador para el módulo USB
11	Oscilador Interno y Señal externa	INTIO	Internal Oscillator como fuente de clock para la CPU, señal externa EC como fuente de oscilador para el módulo USB, con pin de Input/Output digital en el pin RA6
12	Oscilador Interno y Señal externa	INTCKO	Internal Oscillator como fuente de clock para la CPU, señal externa EC como fuente de oscilador para el módulo USB, con salida de frecuencia dividida por 4 (FOSC/4) en RA6

Figura 2.10: Tipos de osciladores

En este proyecto se utilizó el N°4, es decir, el oscilador HSPLL.

Para poder elegir este tipo de reloj se tiene que configurar en formato binario el registro CONFIG1H de la siguiente forma:

$$(2) \text{ CONFIG1H} = \text{xxxx1111}$$

2.2.2. Multiplicador de frecuencia PLL

Los PIC18F2550 incluyen un circuito PLL (Phase Locked Loop). Este ha sido previsto específicamente para aplicaciones USB con osciladores de baja velocidad y también para ser usado como fuente de reloj para el CPU del microcontrolador. En el presente no se hará uso del módulo USB 2.0. Se hace uso del oscilador HSPLL para alimentar la CPU y los periféricos a la máxima frecuencia de reloj. Si se sigue el trayecto de la figura 2.7, se puede ver que una vez que la señal sale de oscilador primario pasa por un PLL Prescaler, cuyo nombre del registro es PLLDIV. Este se debe configurar de forma tal de obtener 4 MHz a la entrada del PLL. En este proyecto se utilizó un cristal de 20 MHz, por lo tanto, se debe configurar el registro CONFIG1L de la siguiente forma:

$$(3) \text{ xxxxx100}$$

De esta forma, el prescaler se configura de forma tal que:

$$\text{Frecuencia entrada PLL} = \frac{20\text{MHz}}{5} = 4\text{MHz}$$

A la salida del PLL se obtiene una frecuencia de 96 MHz. Si se continua el camino de la señal se observa que se encuentra un PLL Postscaler. Este registro se llama CPUDIV y determina la frecuencia final que le llega a la CPU del microcontrolador como así también a sus periféricos, de los cuáles, en este proyecto se utilizaron la UART y I2C. Para configurar este registro se debe escribir en el registro CONFIG1L de la siguiente forma:

$$(4) \text{ xxx00xxx}$$

Los dos bits en cero me determinan la división que realiza el registro. En este caso se configura de forma tal que:

$$\text{Frecuencia CPU} = \frac{\text{Frecuencia de salida del PLL}}{2} = \frac{96\text{Mhz}}{2} = 48\text{Mhz}$$

Por último, siguiendo la figura 2.7, se debe configurar el reloj que se quiere usar (primario, interno o externo al microcontrolador). Este paso se explicó en el presente previamente. Se configuró de forma tal de usar un oscilador primario y que trabaje en el modo HSPLL. Una vez realizado esto, se trazó el camino correspondiente hasta llegar a la frecuencia de la CPU del microcontrolador.

2.2.3. Configuración de los periféricos

En este proyecto se hizo uso de una interrupción por recepción de caracteres, es decir a través del puerto serie (UART), para controlar por medio de una PC la frecuencia de barrido del Si5351. A su vez, para poder programar el generador de señal de reloj (SI5351) a través de sus registros, se tuvieron que programar los mismos mediante el protocolo de comunicación I2C.

En la figura 2.11, se detalla la estructura interna de interrupción del microcontrolador.

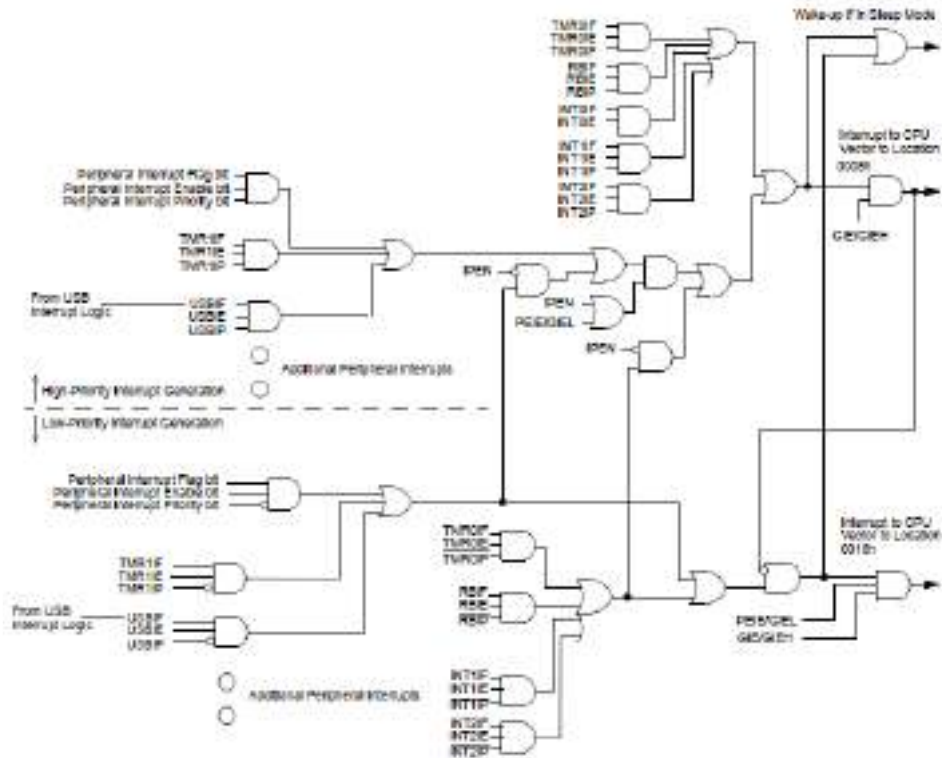


Figura 2.11: Estructura interna de interrupción del microcontrolador

En la figura 2.11, se puede observar todos los bits de configuración como entradas a las compuertas. Estos bits se configuran a través de los registros de control de interrupción. Adicionalmente, se puede observar el trayecto marcado en rojo de la señal.

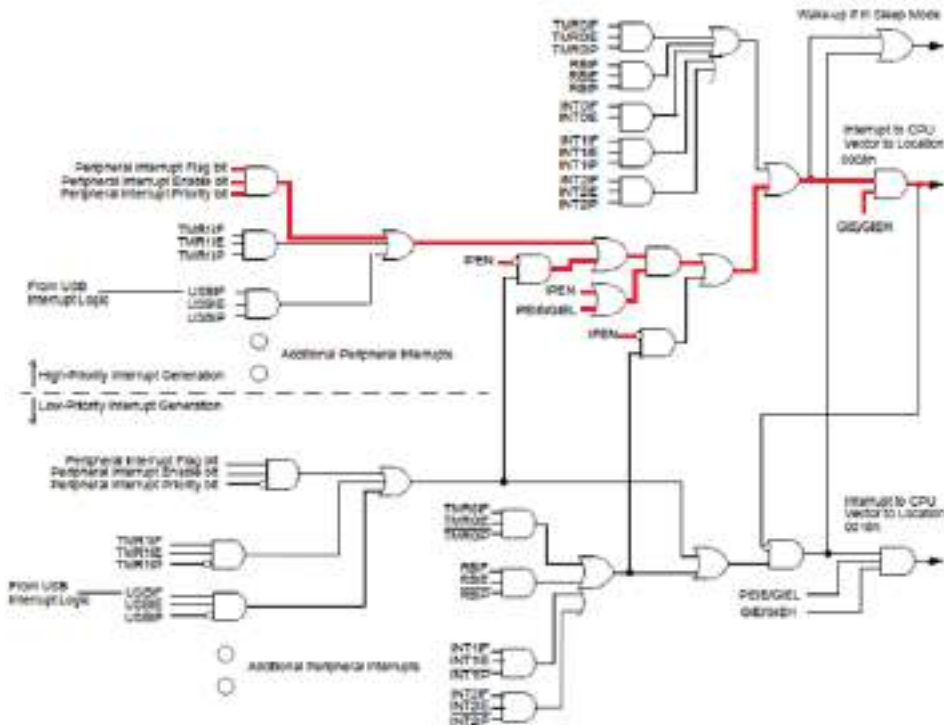


Figura 2.12: Trayecto de la señal para poder generar la interrupción del PIC18F2550

Para poder generar la interrupción en la dirección de memoria de programa 0008h, se debe seguir el trayecto de la figura 2.12. Para ello, se deben configurar los siguientes registros:

a) Registro INTCON

En la figura 2.13, se puede observar que los dos bits del registro INTCON corresponden al bit de habilitación de interrupciones globales y al bit de habilitación de interrupciones por periféricos. Estos bits se habilitaron para poder generar la interrupción de la UART.

Este registro se configuró de la siguiente forma:

$$INTCON = 0b11xxxxxx$$

REGISTER 9-1: INTCON: INTERRUPT CONTROL REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-x
GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF ⁽¹⁾
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7	GIE/GIEH: Global Interrupt Enable bit <u>When IPEN = 0:</u> 1 = Enables all unmasked Interrupts 0 = Disables all Interrupts <u>When IPEN = 1:</u> 1 = Enables all high-priority Interrupts 0 = Disables all Interrupts
bit 6	PEIE/GIEL: Peripheral Interrupt Enable bit <u>When IPEN = 0:</u> 1 = Enables all unmasked peripheral Interrupts 0 = Disables all peripheral Interrupts <u>When IPEN = 1:</u> 1 = Enables all low-priority peripheral Interrupts (if GIE/GIEH = 1) 0 = Disables all low-priority peripheral Interrupts
bit 5	TMR0IE: TMR0 Overflow Interrupt Enable bit 1 = Enables the TMR0 overflow Interrupt 0 = Disables the TMR0 overflow Interrupt
bit 4	INT0IE: INT0 External Interrupt Enable bit 1 = Enables the INT0 external interrupt 0 = Disables the INT0 external Interrupt
bit 3	RBIE: RB Port Change Interrupt Enable bit 1 = Enables the RB port change Interrupt 0 = Disables the RB port change Interrupt
bit 2	TMR0IF: TMR0 Overflow Interrupt Flag bit 1 = TMR0 register has overflowed (must be cleared in software) 0 = TMR0 register did not overflow
bit 1	INT0IF: INT0 External Interrupt Flag bit 1 = The INT0 external Interrupt occurred (must be cleared in software) 0 = The INT0 external Interrupt did not occur
bit 0	RBIF: RB Port Change Interrupt Flag bit ⁽¹⁾ 1 = At least one of the RB7:RB4 pins changed state (must be cleared in software) 0 = None of the RB7:RB4 pins have changed state

Figura 2.13: Registro de control de interrupción

b) Registro RCON

En la figura 2.14, se puede observar el registro de control de reinicio. En este proyecto no se hizo uso de prioridad en las interrupciones, ya que, se utilizó una sola interrupción.

Este registro se configuró de la siguiente forma:

$$RCON = 0b0xxxxxxx$$

REGISTER 9-10: RCON: RESET CONTROL REGISTER

R/W-0	R/W-1 ⁽¹⁾	U-0	R/W-1	R-1	R-1	R/W-0 ⁽²⁾	R/W-0
IPEN	SBOREN	—	RI	TO	PD	POR	BOR
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

- bit 7 **IPEN:** Interrupt Priority Enable bit
1 = Enable priority levels on interrupts
0 = Disable priority levels on interrupts (PIC16CXXX Compatibility mode)
- bit 6 **SBOREN:** BOR Software Enable bit⁽¹⁾
For details of bit operation, see Register 4-1.
- bit 5 **Unimplemented:** Read as '0'
- bit 4 **RI:** RESET Instruction Flag bit
For details of bit operation, see Register 4-1.
- bit 3 **TO:** Watchdog Time-out Flag bit
For details of bit operation, see Register 4-1.
- bit 2 **PD:** Power-Down Detection Flag bit
For details of bit operation, see Register 4-1.
- bit 1 **POR:** Power-on Reset Status bit⁽²⁾
For details of bit operation, see Register 4-1.
- bit 0 **BOR:** Brown-out Reset Status bit
For details of bit operation, see Register 4-1.

Figura 2.14: Registro de control de reinicio

c) Registro PIE1

En la figura 2.15, se puede observar el registro de habilitación de interrupción de periféricos número uno. En este proyecto, se habilitó el bit de interrupción por recepción de caracteres y se deshabilitó el bit de interrupción por transmisión de la USART.

Este registro se configuró de la siguiente forma:

$$PIE1 = 0bxx10xxxx$$

REGISTER 9-6: PIE1: PERIPHERAL INTERRUPT ENABLE REGISTER 1

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
SPPIE ⁽¹⁾	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7	SPPIE: Streaming Parallel Port Read/Write Interrupt Enable bit ⁽¹⁾ 1 = Enables the SPP read/write interrupt 0 = Disables the SPP read/write interrupt
bit 6	ADIE: A/D Converter Interrupt Enable bit 1 = Enables the A/D interrupt 0 = Disables the A/D interrupt
bit 5	RCIE: EUSART Receive Interrupt Enable bit 1 = Enables the EUSART receive interrupt 0 = Disables the EUSART receive interrupt
bit 4	TXIE: EUSART Transmit Interrupt Enable bit 1 = Enables the EUSART transmit interrupt 0 = Disables the EUSART transmit interrupt
bit 3	SSPIE: Master Synchronous Serial Port Interrupt Enable bit 1 = Enables the MSSP interrupt 0 = Disables the MSSP interrupt
bit 2	CCP1IE: CCP1 Interrupt Enable bit 1 = Enables the CCP1 interrupt 0 = Disables the CCP1 interrupt
bit 1	TMR2IE: TMR2 to PR2 Match Interrupt Enable bit 1 = Enables the TMR2 to PR2 match interrupt 0 = Disables the TMR2 to PR2 match interrupt
bit 0	TMR1IE: TMR1 Overflow Interrupt Enable bit 1 = Enables the TMR1 overflow interrupt 0 = Disables the TMR1 overflow interrupt

Figura 2.15: Registro de habilitación de interrupción de periféricos

d) Registro PIR1

En la figura 2.16, se puede observar el registro de bandera de interrupción de periféricos. En este registro, el bit RCIF se configuró inicialmente de forma tal que el búfer de recepción (RCREG) se encuentre vacío. Esto último se lo conoce como “limpiar la bandera”. Adicionalmente, este bit se debe poner en cero cada vez que llega una trama completa de caracteres. De esta forma, se le indica a la PC que puede enviar otro carácter a través del puerto serie (UART) sin que haya errores de desborde o sobrescritura de datos. El bit TXIF se configura inicialmente como cero, es decir, el búfer de transmisión de la UART se encuentra lleno. Este bit se debe poner en uno, cada vez que se envía una trama de caracteres desde el microcontrolador hacia la PC. De esta forma, al igual que en la recepción de caracteres, se evita el desborde o la sobrescritura de datos.

Este registro se configuró de la siguiente forma:

$$PIR1 = 0bxx01xxxx$$

REGISTER 9-4: PIR1: PERIPHERAL INTERRUPT REQUEST (FLAG) REGISTER 1

R/W-0	R/W-0	R-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0
SPPIF ⁽¹⁾	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7	<p>SPPIF: Streaming Parallel Port Read/Write Interrupt Flag bit⁽¹⁾</p> <p>1 = A read or a write operation has taken place (must be cleared in software)</p> <p>0 = No read or write has occurred</p>
bit 6	<p>ADIF: A/D Converter Interrupt Flag bit</p> <p>1 = An A/D conversion completed (must be cleared in software)</p> <p>0 = The A/D conversion is not complete</p>
bit 5	<p>RCIF: EUSART Receive Interrupt Flag bit</p> <p>1 = The EUSART receive buffer, RCREG, is full (cleared when RCREG is read)</p> <p>0 = The EUSART receive buffer is empty</p>
bit 4	<p>TXIF: EUSART Transmit Interrupt Flag bit</p> <p>1 = The EUSART transmit buffer, TXREG, is empty (cleared when TXREG is written)</p> <p>0 = The EUSART transmit buffer is full</p>
bit 3	<p>SSPIF: Master Synchronous Serial Port Interrupt Flag bit</p> <p>1 = The transmission/reception is complete (must be cleared in software)</p> <p>0 = Waiting to transmit/receive</p>
bit 2	<p>CCP1IF: CCP1 Interrupt Flag bit</p> <p><u>Capture mode:</u></p> <p>1 = A TMR1 register capture occurred (must be cleared in software)</p> <p>0 = No TMR1 register capture occurred</p> <p><u>Compare mode:</u></p> <p>1 = A TMR1 register compare match occurred (must be cleared in software)</p> <p>0 = No TMR1 register compare match occurred</p> <p><u>PWM mode:</u></p> <p>Unused in this mode.</p>
bit 1	<p>TMR2IF: TMR2 to PR2 Match Interrupt Flag bit</p> <p>1 = TMR2 to PR2 match occurred (must be cleared in software)</p> <p>0 = No TMR2 to PR2 match occurred</p>
bit 0	<p>TMR1IF: TMR1 Overflow Interrupt Flag bit</p> <p>1 = TMR1 register overflowed (must be cleared in software)</p> <p>0 = TMR1 register did not overflow</p>

Figura 2.16: Registro de bandera de interrupción de periféricos

2.2.4. Módulo UART

2.2.4.1. Transmisión

La operación de este módulo se basa en tres registros:

- TXSTA: registro de control y estado de transmisión
- RCSTA: registro de control y estado de recepción
- BAUDCON: registro de control de tasa de baudios

En la figura 2.17, se puede observar el procedimiento de transmisión del módulo UART:

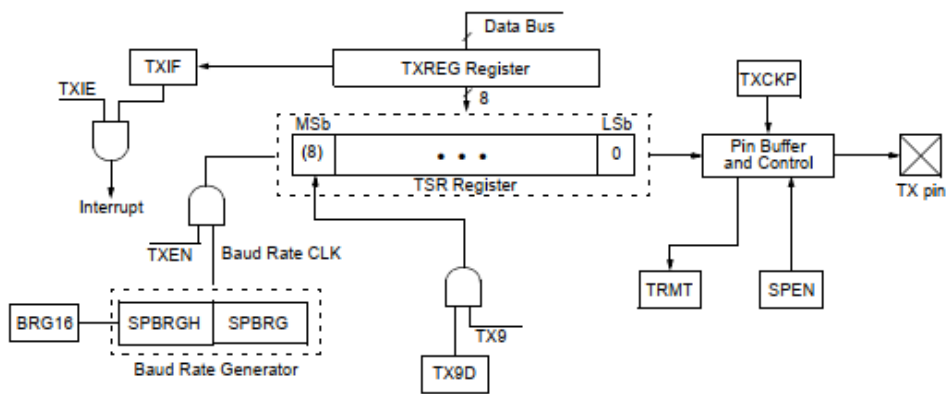


Figura 2.17: Diagrama en bloques del procedimiento de transmisión de la UART

El procedimiento de transmisión sin interrupción de la UART es el siguiente:

- Se configuran los bits BRGH y BRG16 para obtener la tasa de baudios deseada para la transmisión.
- Se habilita el puerto serie colocando en binario un uno en el bit SPEN, de esta forma, se configuran los pines de transmisión (Tx) y recepción (Rx) de la UART.
- Se configura el bit SYNC en cero para que el modo de transmisión sea asincrónico.
- Se habilita la transmisión colocando en binario un uno en el bit TXEN.
- Se carga el dato en el registro TXREG.

Una vez realizado este procedimiento, el registro TXREG transfiere el byte de datos al registro de desplazamiento de transmisión (TSR), este se vacía y el bit de bandera (TXIF) se activa. Esta bandera indica que la transmisión se completó, es decir, el byte se envió.

En la figura 2.18, se puede ver el registro de control y estado de transmisión. En este proyecto se usó la comunicación RS232 con ocho bits de transmisión en modo asincrónico. Se seleccionó una tasa de baudios configurada en alta velocidad. El bit TRMT se usa como bit de sólo lectura para comprobar mediante firmware que ya se envió la trama de datos y el búfer TSR se encuentra libre. El bit TXEN se configura en uno para poder habilitar la transmisión.

Los bits de este registro se configuraron de la siguiente forma:

$$TXSTA = 0bx010x1xx$$

REGISTER 20-1: TXSTA: TRANSMIT STATUS AND CONTROL REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-1	R/W-0
CSRC	TX9	TXEN ⁽¹⁾	SYNC	SENDB	BRGH	TRMT	TX9D
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

- bit 7 **CSRC**: Clock Source Select bit
Asynchronous mode:
 Don't care.
Synchronous mode:
 1 = Master mode (clock generated internally from BRG)
 0 = Slave mode (clock from external source)
- bit 6 **TX9**: 9-Bit Transmit Enable bit
 1 = Selects 9-bit transmission
 0 = Selects 8-bit transmission
- bit 5 **TXEN**: Transmit Enable bit⁽¹⁾
 1 = Transmit enabled
 0 = Transmit disabled
- bit 4 **SYNC**: EUSART Mode Select bit
 1 = Synchronous mode
 0 = Asynchronous mode
- bit 3 **SENDB**: Send Break Character bit
Asynchronous mode:
 1 = Send Sync Break on next transmission (cleared by hardware upon completion)
 0 = Sync Break transmission completed
Synchronous mode:
 Don't care.
- bit 2 **BRGH**: High Baud Rate Select bit
Asynchronous mode:
 1 = High speed
 0 = Low speed
Synchronous mode:
 Unused in this mode.
- bit 1 **TRMT**: Transmit Shift Register Status bit
 1 = TSR empty
 0 = TSR full
- bit 0 **TX9D**: 9th bit of Transmit Data
 Can be address/data bit or a parity bit.

Figura 2.18: Registro de control y estado de transmisión

2.2.4.2. Recepción

En la figura 2.19, se puede observar el procedimiento de recepción del módulo UART:

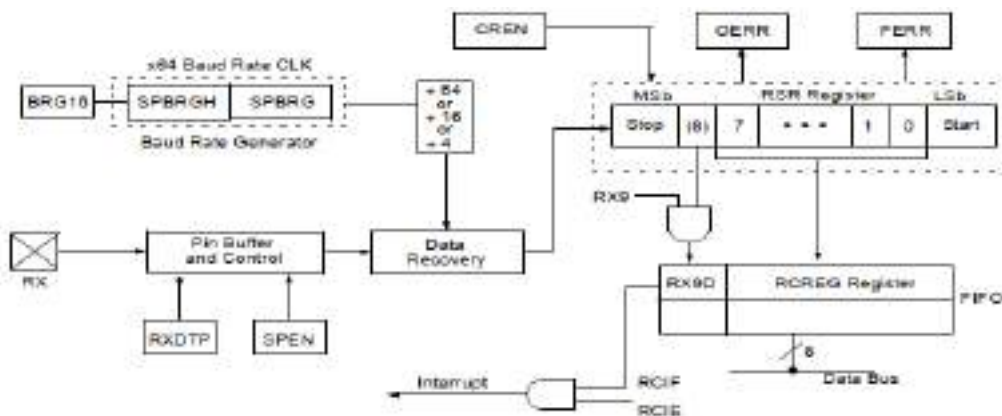


Figura 2.19: Diagrama en bloques del procedimiento de recepción de la UART

El procedimiento de recepción de caracteres con interrupción de la UART es el siguiente:

- Se configuran los bits BRGH y BRG16 para obtener la tasa de baudios deseada para la transmisión.
- Se habilita el puerto serie colocando en binario un uno en el bit SPEN, de esta forma, se configuran los pines de transmisión (Tx) y recepción (Rx) de la UART.
- Se habilita la interrupción por recepción de caracteres. Esto se hace colocando en binario un uno en el bit RCIE del registro PIE1.
- Se habilita la recepción continua de datos. Se coloca en uno el bit CREN del registro RCSTA.
- El bit de bandera RCIF se pondrá en uno automáticamente cuando se complete la recepción de la trama entera y se generará una interrupción.
- Por último, debe leer el registro RCSTA. Si ocurre algún error se debe deshabilitar la recepción de caracteres poniendo en cero el bit CREN del registro RCSTA

En la figura 2.20, se pueden observar el registro de control y estado de recepción. En base al procedimiento de configuración de recepción de caracteres de la UART, el registro RCSTA se configuró de la siguiente forma:

$$RCSTA = 0b10x1xxxx$$

REGISTER 20-2: RCSTA: RECEIVE STATUS AND CONTROL REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-0	R-0	R-x
SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7	SPEN: Serial Port Enable bit 1 = Serial port enabled (configures RX/DT and TX/CK pins as serial port pins) 0 = Serial port disabled (held in Reset)
bit 6	RX9: 9-Bit Receive Enable bit 1 = Selects 9-bit reception 0 = Selects 8-bit reception
bit 5	SREN: Single Receive Enable bit <u>Asynchronous mode:</u> Don't care. <u>Synchronous mode – Master:</u> 1 = Enables single receive 0 = Disables single receive This bit is cleared after reception is complete. <u>Synchronous mode – Slave:</u> Don't care.
bit 4	CREN: Continuous Receive Enable bit <u>Asynchronous mode:</u> 1 = Enables receiver 0 = Disables receiver <u>Synchronous mode:</u> 1 = Enables continuous receive until enable bit CREN is cleared (CREN overrides SREN) 0 = Disables continuous receive
bit 3	ADDEN: Address Detect Enable bit <u>Asynchronous mode 9-bit (RX9 = 1):</u> 1 = Enables address detection, enables Interrupt and loads the receive buffer when RSR=8 is set 0 = Disables address detection, all bytes are received and ninth bit can be used as parity bit <u>Asynchronous mode 8-bit (RX9 = 0):</u> Don't care.
bit 2	FERR: Framing Error bit 1 = Framing error (can be updated by reading RCREG register and receiving next valid byte) 0 = No framing error
bit 1	OERR: Overrun Error bit 1 = Overrun error (can be cleared by clearing bit CREN) 0 = No overrun error
bit 0	RX9D: 9th bit of Received Data This can be address/data bit or a parity bit and must be calculated by user firmware.

Figura 2.20: Registro de control y estado de recepción

2.2.4.3. Generador de tasa de baudios

Se usa para configurar la velocidad de transmisión y recepción de caracteres, es decir, la velocidad de la comunicación serial entre el microcontrolador y la PC. En el presente se configuró la velocidad de transmisión en 115200 baudios. Esta corresponde a una velocidad de transmisión de 115200 bits / segundo. Además, se optó por utilizar ocho bits de transmisión. Esto se configura colocando un cero en el bit BRG16 del registro BAUDCON. El modo de transmisión se configuró de forma asincrónica. Esto último, se lleva a cabo colocando un cero en el bit SYNC del registro TXSTA. En la figura 2.21, se puede observar la fórmula a utilizar para configurar la velocidad de baudios deseada:

Configuration Bits			BRG/EUSART Mode	Baud Rate Formula
SYNC	BRG16	BRGH		
0	0	0	8-bit/Asynchronous	$F_{osc}/[64(n+1)]$
0	0	1	8-bit/Asynchronous	$F_{osc}/[16(n+1)]$
0	1	0	16-bit/Asynchronous	$F_{osc}/[16(n+1)]$
0	1	1	16-bit/Asynchronous	$F_{osc}/[16(n+1)]$
1	0	x	8-bit/Synchronous	$F_{osc}/[4(n+1)]$
1	1	x	16-bit/Synchronous	$F_{osc}/[4(n+1)]$

Figura 2.21: Configuración de tasa de baudios

En la figura 2.21, se puede observar, remarcada en rojo, la fórmula para configurar la velocidad de comunicación serial. En este proyecto se configuró de la siguiente forma:

$$\text{tasa de baudios deseada} = \frac{F_{osc}}{16 \times (n + 1)}$$

Dónde n corresponde al registro SPBRG. Este valor se debe calcular y configurar mediante firmware para obtener la tasa de baudios deseada. En este proyecto se calculó de la siguiente forma:

$$115200 \frac{\text{bits}}{\text{s}} = \frac{48000000 \text{ Hz}}{16 \times (SPBRG + 1)}$$

Despejando,

$$SPBRG = \frac{48000000 \text{ Hz}}{16 \times 115200 \frac{\text{bits}}{\text{s}}} - 1 = 25,042$$

Mediante firmware se redondea al valor más entero más cercano, en este caso:

$$SPBRG = 25$$

Se vuelve a calcular la tasa de baudios para este nuevo valor:

$$\text{Tasa de baudios calculada} = 115384,62$$

El error que se comete al redondear el valor del registro se puede calcular también de la siguiente forma:

$$\text{Error} = \frac{\text{tasa de baudios calculada} - \text{tasa de baudios deseada}}{\text{tasa de baudios deseada}} = \frac{184,62}{115200} = 0,16 \%$$

2.2.5. Módulo puerto serie síncrono maestro (MSSP)

Este módulo es una interfaz serie que sirve para comunicarse con otros dispositivos periféricos o microcontroladores. Estos dispositivos periféricos pueden ser memorias EEPROMs, registros de desplazamientos, etc.

El módulo MSSP puede configurarse en uno de los siguientes modos:

- Interfaz de periféricos serie (SPI)
- Circuito inter – integrado (I2C)

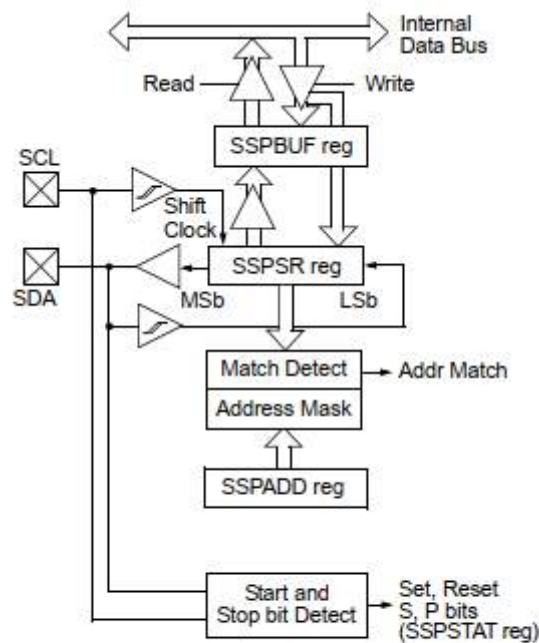
En este proyecto se utilizó el bus I2C para programar el generador de señal de reloj SI5351.

2.2.5.1. Bus I2C

La operación de este módulo se basa en los siguientes registros:

- Registro de control uno (SSPCON 1)
- Registro de control dos (SSPCON 2)
- Registro de estado (SSPSTAT)
- Registro de búfer de transmisión / recepción en serie (SSPBUF)
- Registro de desplazamiento (SSPSR)
- Registros de dirección (SSPADD)

En la figura 2.22, se puede observar el diagrama en bloques del modo I2C:



Note: Only port I/O names are used in this diagram for the sake of brevity. Refer to the text for a full list of multiplexed functions.

Figura 2.22: Diagrama en bloques del modo I2C

Los registros SSPCON1, SSPCON2 y SSPSTAT son de control y estado del bus I2C. Estos se pueden acceder, leer y escribir. Con respecto al registro SSPSTAT, los seis bits más bajos son de sólo lectura y los dos más altos se pueden acceder, leer y escribir. El registro SSPSR no es accesible por el usuario. Este se usa para desplazar los datos de entrada o salida. El registro búfer de datos SSPADD se usa para escribir o leer datos. Cuando el modo I2C se configura como esclavo, este contiene la dirección del esclavo y cuando este se configura como maestro, los siete bits más bajos contienen el valor correspondiente a la configuración del generador de velocidad en baudios. En operaciones de recepción de datos el registro SSPSR y SSPBUF actúan como doble búfer, es decir, cuando el registro SSPSR recibe un byte completo lo transfiere al registro SSPBUF y se produce la interrupción activando la bandera SSPIF. Por lo contrario, en operaciones de transmisión de datos escribir un byte en el registro SSPBUF implicará escribir al mismo tiempo en el registro SSPSR.

2.2.5.2. Modo maestro

En este modo el maestro (PIC18F2550) genera todos los pulsos de reloj serie y las condiciones de comienzo (Start) y parada (Stop). La transmisión finaliza con una condición de parada o con una repetición de comienzo. La repetición de comienzo es el comienzo del próximo dato a transferir, por lo tanto, el bus I2C aún no se libera.

El generador de tasa de baudios se usa para configurar la frecuencia de reloj para el terminal SCL del microcontrolador.

2.2.5.2.1 Configuración inicial del modo maestro

- 1) Se habilita el puerto serie síncrono maestro (I2C). Para ello se debe acceder al registro SSPCON1, mediante firmware, escribiendo un uno en el bit SSPEN. Una vez habilitado este, se configuran los pines SDA y SCL del PIC18F2550. Esto último se realiza accediendo al registro TRIS, con esto, se configuran los dos pines como entrada.
- 2) Se configura el microcontrolador para que trabaje como maestro en el modo I2C. Se debe configurar los cuatro bits más bajos (SSPM) del registro SSPCON1 para que trabaje como maestro y genere la señal de reloj adecuada.
- 3) Se configura el bit SMP del registro SSPSTAT para que el maestro trabaje a velocidad estándar, es decir 100 kHz.
- 4) Se configura el generador de tasa de baudios (BRG). Este se realiza escribiendo en el registro SSPADD el valor correspondiente para obtener la velocidad de trabajo deseada.

En las figuras 2.23 y 2.24, se puede observar el registro de control SSPCON1 y el registro de estado SSPSTAT respectivamente. Estos dos registros se configuraron, en base al procedimiento descrito anteriormente, de la siguiente forma:

$$SSPCON1 = 0bxx1x1000$$

$$SSPSTAT = 0b0xxxxxx$$

$$SSPADD = \frac{FOSC}{4 * (Clock + 1)} = \frac{48000000 \text{ Hz}}{4 * (100000 \text{ Hz} + 1)} = 119$$

REGISTER 19-4: SSPCON1: MSSP CONTROL REGISTER 1 (I²C™ MODE)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
WCOL	SSPOV	SSPEN	CKP	SSPM0	SSPM2	SSPM1	SSPM3
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-1 = Value at POR	'1' = Bit is set	0 = Bit is cleared	x = Bit is unknown

bit 7 **WCOL:** Write Collision Detect bit
In Master Transmit mode:
 1 = A write to the SSPBUF register was attempted while the PC conditions were not valid for a transmission to be started (must be cleared in software)
 0 = No collision
In Slave Transmit mode:
 1 = The SSPBUF register is written while it is still transmitting the previous word (must be cleared in software)
 0 = No collision
In Receive mode (Master or Slave mode):
 This is a "don't care" bit.

bit 6 **SSPOV:** Receive Overflow Indicator bit
In Receive mode:
 1 = A byte is received while the SSPBUF register is still holding the previous byte (must be cleared in software)
 0 = No overflow
In Transceive mode:
 This is a "don't care" bit in Transceive mode.

bit 5 **SSPEN:** Master Synchronous Serial Port Enable bit
 1 = Enables the serial port and configures the SDA and SCL pins as the serial port pins⁽¹⁾
 0 = Disables serial port and configures these pins as I/O port pins⁽²⁾

bit 4 **CKP:** SCK Release Control bit
In Slave mode:
 1 = Release clock
 0 = holds clock low (clock stretch), used to ensure data setup time
In Master mode:
 Unused in this mode.

bit 3-0 **SSPM0:SSPM3:** Master Synchronous Serial Port Mode Select bits
 1111 = I²C Slave mode, 10-bit address with Start and Stop bit interrupts enabled⁽³⁾
 1110 = I²C Slave mode, 7-bit address with Start and Stop bit interrupts enabled⁽³⁾
 1011 = I²C Firmware Controlled Master mode (Slave ID)⁽⁴⁾
 1000 = I²C Master mode, clock = F_{osc}/4 * (SSPAD0 + 1)^(2,5)
 0111 = I²C Slave mode, 10-bit address⁽³⁾
 0110 = I²C Slave mode, 7-bit address⁽³⁾

Figura 2.23: Registro de control SSPCON1

REGISTER 19-3: SSPSTAT: MSSP STATUS REGISTER (I²C™ MODE)

R/W-0	R/W-0	R-0	R-0	R-0	R-0	R-0	R-0
SRP	CKE	DA	RF1	RF0	RAV ⁽¹⁾	UA	BF
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-1 = Value at POR	'1' = Bit is set	0 = Bit is cleared	x = Bit is unknown

bit 7 **SRP:** Slew Rate Control bit
In Master or Slave mode:
 1 = Slew rate control disabled for standard speed mode (100 kHz and 1 MHz)
 0 = Slew rate control enabled for High-Speed mode (400 kHz)

bit 6 **CKE:** SMBus Select bit
In Master or Slave mode:
 1 = Enable SMBus specific inputs
 0 = Disable SMBus specific inputs

bit 5 **DA:** Data/Address bit
In Master mode:
 Reserved.
In Slave mode:
 1 = Indicates that the last byte received or transmitted was data
 0 = Indicates that the last byte received or transmitted was address

bit 4 **RF:** Stop bit⁽³⁾
 1 = Indicates that a stop bit has been detected last
 0 = Stop bit was not detected last

bit 3 **RF0:** Start bit⁽³⁾
 1 = Indicates that a Start bit has been detected last
 0 = Start bit was not detected last

bit 2 **RAV:** Read/Write Information bit⁽¹⁾
In Slave mode:
 1 = Read
 0 = Write
In Master mode:
 1 = Transmit is in progress
 0 = Transmit is not in progress

bit 1 **UA:** Update Address bit (10-bit Slave mode only)
 1 = Indicates that the user needs to update the address in the SSPADD register
 0 = Address does not need to be updated

bit 0 **BF:** Buffer Full Status bit
In Transmit mode:
 1 = SSPBUF is full
 0 = SSPBUF is empty
In Receive mode:
 1 = SSPBUF is full (does not include the ACK and Stop bits)
 0 = SSPBUF is empty (does not include the ACK and Stop bits)

Figura 2.24: Registro de estado SSPSTAT

2.2.5.2.2 Operaciones en modo maestro I2C

Escritura de datos:

- 1) Se genera la condición de comienzo. Para ello, se debe escribir un uno en el bit SEN del registro SSPCON2. Esto genera que se active el bit de bandera de interrupción SSPIF. El usuario debe borrar el bit de bandera de interrupción mediante firmware una vez que la condición de comienzo ya se envió.
- 2) El módulo MSSP espera un tiempo de comienzo antes de continuar.
- 3) Se carga en el registro SSPBUF la dirección física del esclavo que se quiere enviar. Esta dirección está compuesta por ocho bits (un byte). Los siete bits más significativos son para indicar la dirección y el último, en este caso, se usa para indicar la operación de escritura.
- 4) La dirección física se desplaza través del pin SDA hasta que los ocho bits son transmitidos.
- 5) Se debe esperar el ACK del esclavo. El módulo MSSP del microcontrolador pone en uno el bit ACKSTAT del registro SSPCON2.
- 6) El módulo MSSP genera una interrupción al final del noveno pulso de reloj. Coloca el bit de la bandera de interrupción SSPIF en uno. El usuario debe borrar el bit de bandera de interrupción mediante firmware.
- 7) Se carga en el registro SSPBUF la dirección de memoria del esclavo que se quiere enviar. Esta dirección está compuesta por ocho bits (un byte).
- 8) La dirección de memoria se desplaza través del pin SDA hasta que los ocho bits son transmitidos.
- 9) Se debe esperar el ACK del esclavo. El módulo MSSP del microcontrolador pone en uno el bit ACKSTAT del registro SSPCON2.
- 10) El módulo MSSP genera una interrupción al final del noveno pulso de reloj. Coloca el bit de la bandera de interrupción SSPIF en uno. El usuario debe borrar el bit de bandera de interrupción mediante firmware.
- 11) Se carga en el registro SSPBUF el dato que se quiere enviar. Este dato está compuesto por ocho bits (un byte).
- 12) El dato se desplaza través del pin SDA hasta que los ocho bits son transmitidos.
- 13) Se debe esperar el ACK del esclavo. El módulo MSSP del microcontrolador pone en uno el bit ACKSTAT del registro SSPCON2.
- 14) El módulo MSSP genera una interrupción al final del noveno pulso de reloj. Coloca el bit de la bandera de interrupción SSPIF en uno. El usuario debe borrar el bit de bandera de interrupción mediante firmware.
- 15) Se genera la condición de parada. Para ello, se debe escribir un uno en el bit PEN del registro SSPCON2. Esto genera que se active el bit de bandera de interrupción SSPIF una vez que la condición de parada se completa.

Lectura de datos:

- 1) Se genera la condición de comienzo. Para ello, se debe escribir un uno en el bit SEN del registro SSPCON2. Esto genera que se active el bit de bandera de interrupción SSPIF. El usuario debe borrar el bit de bandera de interrupción mediante firmware una vez que la condición de comienzo ya se envió.
- 2) El módulo MSSP espera un tiempo de comienzo antes de continuar.
- 3) Se carga en el registro SSPBUF la dirección física del esclavo que se quiere enviar. Esta dirección está compuesta por ocho bits (un byte). Los siete bits más significativos son para indicar la dirección y el último, en este caso, se usa para indicar la operación de escritura.
- 4) La dirección física se desplaza través del pin SDA hasta que los ocho bits son transmitidos.
- 5) Se debe esperar el ACK del esclavo. El módulo MSSP del microcontrolador pone en uno el bit ACKSTAT del registro SSPCON2.
- 6) El módulo MSSP genera una interrupción al final del noveno pulso de reloj. Coloca el bit de la bandera de interrupción SSPIF en uno. El usuario debe borrar el bit de bandera de interrupción mediante firmware.
- 7) Se carga en el registro SSPBUF la dirección de memoria del esclavo que se quiere enviar. Esta dirección está compuesta por ocho bits (un byte).
- 8) La dirección de memoria se desplaza través del pin SDA hasta que los ocho bits son transmitidos.
- 9) Se debe esperar el ACK del esclavo. El módulo MSSP del microcontrolador pone en uno el bit ACKSTAT del registro SSPCON2.
- 10) El módulo MSSP genera una interrupción al final del noveno pulso de reloj. Coloca el bit de la bandera de interrupción SSPIF en uno. El usuario debe borrar el bit de bandera de interrupción mediante firmware.
- 11) Se genera la repetición de comienzo. Se debe escribir un uno en el bit RSEN del registro SSPCON2. Esto genera que se active el bit de bandera de interrupción SSPIF. El usuario debe borrar el bit de bandera de interrupción mediante firmware una vez que la condición de repetición de comienzo ya se envió.
- 12) Se carga en el registro SSPBUF la dirección física del esclavo que se quiere enviar. Esta dirección está compuesta por ocho bits (un byte). Los siete bits más significativos son para indicar la dirección y el último, en este caso, se usa para indicar la operación de lectura.
- 13) La dirección física se desplaza través del pin SDA hasta que los ocho bits son transmitidos.
- 14) Se debe esperar el ACK del esclavo. El módulo MSSP del microcontrolador pone en uno el bit ACKSTAT del registro SSPCON2.
- 15) El módulo MSSP genera una interrupción al final del noveno pulso de reloj. Coloca el bit de la bandera de interrupción SSPIF en uno. El usuario debe borrar el bit de bandera de interrupción mediante firmware.
- 16) Se habilita la recepción de datos por parte del esclavo. Se debe poner en uno el bit RCEN del registro SSPCON2.
- 17) Se debe esperar a que la bandera de estado BF se ponga en cero. Esta bandera se pone en uno cuando una dirección o dato se carga en el registro SSPBUF desde el registro SSPSR. Se pone en cero cuando el registro SSPBUF es leído.

- 18) Se almacena el dato recibido en un variable. Para ello, se debe asignar dicha variable al registro SSPBUF.
- 19) Si es el último dato recibido, el usuario debe generar el bit de reconocimiento de datos por parte del maestro hacia el esclavo. Se escribe un uno en el bit de reconocimiento de datos ACKDT del registro SSPCON2.
- 20) Se genera la condición de parada. Se escribe un uno en el bit de parada STOP del registro SSPCON2.
- 21) Se lee la variable en la cual se almacenó el dato mediante firmware.

En la figura 2.25, se puede observar los bits de configuración del registro asociado con el procedimiento de escritura y de lectura de datos.

REGISTER 19-5: SSPCON2: MSSP CONTROL REGISTER 2 (I²C™ MASTER MODE)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
GCEN	ACKSTAT	ACKDT ⁽¹⁾	ACKEN ⁽²⁾	RCEN ⁽²⁾	PEN ⁽²⁾	RSEN ⁽²⁾	SEN ⁽²⁾
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7	GCEN: General Call Enable bit (Slave mode only) Unused in Master mode.
bit 6	ACKSTAT: Acknowledge Status bit (Master Transmit mode only) 1 = Acknowledge was not received from slave 0 = Acknowledge was received from slave
bit 5	ACKDT: Acknowledge Data bit (Master Receive mode only) ⁽¹⁾ 1 = Not Acknowledge 0 = Acknowledge
bit 4	ACKEN: Acknowledge Sequence Enable bit ⁽²⁾ 1 = Initiate Acknowledge sequence on SDA and SCL pins and transmit ACKDT data bit. Automatically cleared by hardware. 0 = Acknowledge sequence Idle
bit 3	RCEN: Receive Enable bit (Master Receive mode only) ⁽²⁾ 1 = Enables Receive mode for I ² C 0 = Receive Idle
bit 2	PEN: Stop Condition Enable bit ⁽²⁾ 1 = Initiate Stop condition on SDA and SCL pins. Automatically cleared by hardware. 0 = Stop condition Idle
bit 1	RSEN: Repeated Start Condition Enable bit ⁽²⁾ 1 = Initiate Repeated Start condition on SDA and SCL pins. Automatically cleared by hardware. 0 = Repeated Start condition Idle
bit 0	SEN: Start Condition Enable/Stretch Enable bit ⁽²⁾ 1 = Initiate Start condition on SDA and SCL pins. Automatically cleared by hardware. 0 = Start condition Idle

Figura 2.25: Registro de control SSPCON2

2.3. Configuración de los registros del Si5351

2.3.1 Introducción

El Si5351 es un generador de reloj configurable mediante el protocolo I2C. Es ideal para reemplazar cristales, osciladores de cristal, VCXO, PLLs, etc. Consta de dos sintetizadores (PLL), tres divisores de salida, uno por canal de salida y una etapa de salida por canal, todos controlados a través del bus I2C. También se incorpora un circuito oscilador común (OSC), que funciona a 25 MHz o 27 MHz, pero necesita un cristal externo. Adicionalmente, este generador de señal de reloj puede generar cualquier frecuencia con resolución de 1 Hz hasta 200 MHz en cada uno de sus salidas con error de 0 ppm. En la figura 2.26, se puede observar el diagrama en bloques de los distintos componentes internos que conforman el Si5351.

En este proyecto, se usarán las salidas CLK0 y CLK1 en el rango de 30 a 50 MHz. Estas dos salidas presentan una señal del tipo onda cuadrada cuya amplitud es de 3,3 Vpp. Ambas señales deberán ser filtradas, mediante un filtro pasa bajos, para quedarse solamente con la armónica fundamental. Una vez filtradas, la señal CLK0 se usará para caracterizar la impedancia incógnita conectada en el medidor de impedancias y la señal CLK1 se usará como oscilador local para los mezcladores SA612A. Esta última estará 1 kHz por encima de la señal de caracterización de impedancia (CLK0) durante todo el barrido de frecuencia.

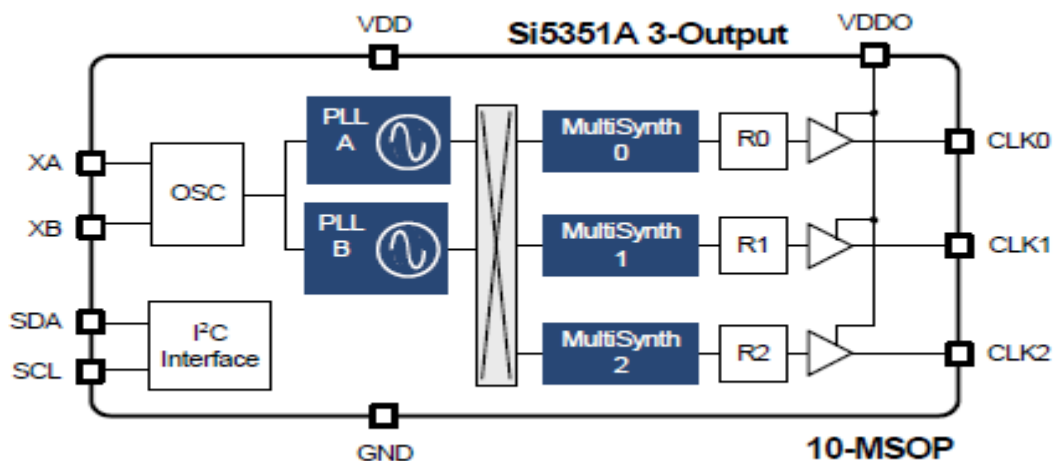


Figura 2.26: Diagrama en bloques de los distintos componentes del Si5351

2.3.2 Funcionamiento

En la figura 2.27, se puede observar el funcionamiento de forma más detallada de los bloques de la estructura interna del Si5351.

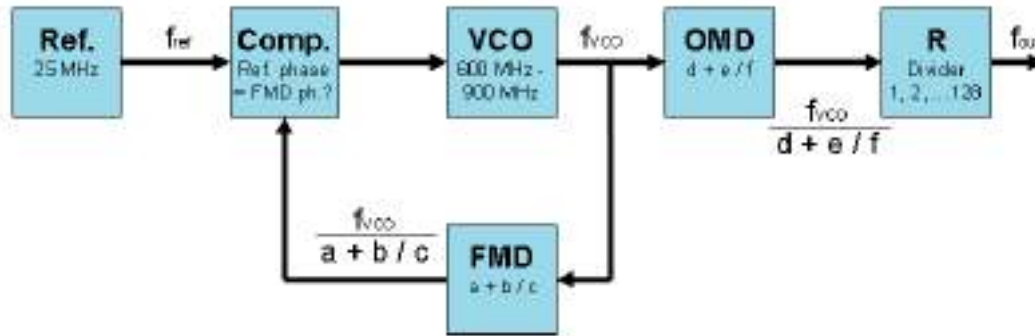


Figura 2.27: Diagrama en bloques de la estructura interna del Si5351

2.3.2.1 Sintetizador de frecuencias (PLL)

Un sintetizador de radiofrecuencia se encarga de generar frecuencias a partir de una frecuencia de referencia. En este caso se usa un cristal de 25 MHz como referencia.

Un sintetizador consta de tres circuitos electrónicos principales: un comparador de fase, un VCO y un divisor de realimentación (FMD). Para cada ajuste programado diferente del FMD, el VCO generará una frecuencia diferente.

2.3.2.2 Divisor multifraccionario de salida (OMD)

Para mejorar la resolución de frecuencia, desde pasos amplios hasta pasos más finos, el sintetizador en el Si5351A es seguido por un circuito divisor, que la empresa SiLabs denomina divisor de salida múltiple (OMD). Además, ha agregado una segunda etapa divisoria, llamada R, que puede dividir aún más la frecuencia generada por el sintetizador.

El inconveniente de un divisor de salida es que la frecuencia de salida (f_{out}) será menor que la frecuencia de VCO (f_{vco}). En el Si5351A hay una división mínima de cuatro. Esto significa que la frecuencia VCO es siempre al menos cuatro veces mayor que la frecuencia de salida.

Pero en lugar de tener pasos de frecuencia amplios, por ejemplo 25 kHz, los pasos pueden ser mucho más bajos, como por ejemplo 1 Hz. Todo depende de cómo estén configuradas las relaciones FMD y OMD.

2.3.2.3 Configuración de los registros

La ecuación matemática que relaciona la frecuencia de salida con la frecuencia de entrada, basándose en la figura 2.27, es la siguiente:

$$(1) f_{out} = \frac{f_{ref} \times FMD}{R * OMD} = \frac{f_{vco}}{R * OMD}$$

Dónde FMD corresponde al divisor de realimentación y OMD al divisor de salida múltiple. Sus ecuaciones son las siguientes:

$$(2) FMD = \frac{f_{vco}}{f_{ref}} = \frac{f_{vco}}{f_{xtal}} = a + \frac{b}{c}$$

$$(3) \text{ OMD} = \frac{f_{\text{vco}}}{f_{\text{out}}} = d + \frac{e}{f}$$

La parte entera de la ecuación del divisor de realimentación es “a” y la parte fraccionaria es “b/c”. Para poder programar el divisor se debe elegir la frecuencia del oscilador controlado por voltaje (f_{vco}) adecuada. La frecuencia del VCO solo puede ir desde 600 MHz a 900 MHz para su correcto funcionamiento. La frecuencia de referencia (f_{ref}) está fijada a la frecuencia del cristal, es decir, 25 MHz. La relación fraccional FMD también tiene un rango de validez. Esta va desde $15 + 0/1,048,575$ hasta $90 + 0/1,048,575$. Es decir, la parte entera “a” puede tomar valores enteros en el rango de 15 a 90. El numerador de la parte fraccionaria “b” va desde 0 hasta 1048575 y el denominador “c” va desde 1 hasta 1048575.

2.3.3 Configuración del Si5351

Para poder obtener la frecuencia de reloj de salida deseada se debe programar el dispositivo mediante firmware. En la figura 2.28, se puede observar el procedimiento para llevarlo a cabo.

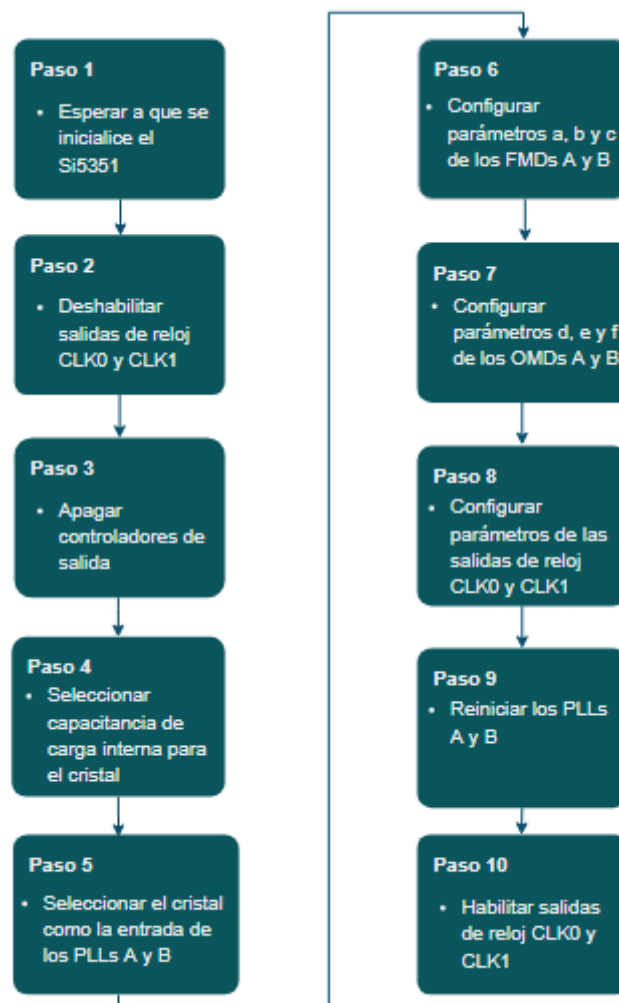


Figura 2.28: Procedimiento de programación I2C del Si5351

2.3.3.1 Configuración de los registros asociados al procedimiento de programación del Si5351

Paso 1: Se debe leer mediante el protocolo I2C el bit SYS_INIT del registro de verificación de estado del Si5351. Si este se encuentra en uno, indica que la inicialización del sistema se encuentra completa y se puede proceder con el paso dos. En la figura 2.29, se puede observar este registro.

Register 0. Device Status								
Bit	D7	D6	D5	D4	D3	D2	D1	D0
Name	SYS_INIT	LOL_B	LOL_A	LOS			REVID[1:0]	
Type	R	R	R	R	R	R	R	

Reset value = 0000 0000

Bit	Name	Function
7	SYS_INIT	<p>System Initialization Status.</p> <p>During power up the device copies the content of the NVM into RAM and performs a system initialization. The device is not operational until initialization is complete. It is not recommended to read or write registers in RAM through the I²C interface until initialization is complete. An interrupt will be triggered (INTR pin = 0, Si5351C only) during the system initialization period.</p> <p>0: System initialization is complete. Device is ready. 1: Device is in system initialization mode.</p>
6	LOL_B	<p>PLL B Loss Of Lock Status.</p> <p>Si5351A/C only. PLLB will operate in a locked state when it has a valid reference from CLKIN or XTAL. A loss of lock will occur if the frequency of the reference clock forces the PLL to operate outside of its lock range or if the reference clock fails to meet the minimum requirements of a valid input signal as specified in the Si5351 data sheet. An interrupt will be triggered (INTR pin = 0, Si5351C) during a LOL condition.</p> <p>0: PLL B is locked. 1: PLL B is unlocked. When the device is in this state it will trigger an interrupt causing the INTR pin to go low (Si5351C only).</p>
5	LOL_A	<p>PLL A Loss Of Lock Status.</p> <p>PLL A will operate in a locked state when it has a valid reference from CLKIN or XTAL. A loss of lock will occur if the frequency of the reference clock forces the PLL to operate outside of its lock range as specified in the data sheet, or if the reference clock fails to meet the minimum requirements of a valid input signal as specified in the Si5351 data sheet. An interrupt will be triggered (INTR pin = 0, Si5351C only) during a LOL condition.</p> <p>0: PLL A is operating normally. 1: PLL A is unlocked. When the device is in this state it will trigger an interrupt causing the INTR pin to go low (Si5351C only).</p>
4	LOS	<p>CLKIN Loss Of Signal (Si5351C Only).</p> <p>A loss of signal status indicates if the reference clock fails to meet the minimum requirements of a valid input signal as specified in the Si5351 data sheet. An interrupt will be triggered (INTR pin = 0, Si5351C only) during a LOS condition.</p> <p>0: Valid clock signal at the CLKIN pin. 1: Loss of signal detected at the CLKIN pin.</p>
3:0	Reserved	Reserved.

Figura 2.29: Registro de verificación de estado del dispositivo

Paso 2: En la figura 2.30, se puede observar el registro de control de habilitación de salida del Si5351. Se debe escribir un uno en los dos bits menos significativos del registro tres del Si5351 y, de esta forma, se deshabilitan las salidas de reloj CLK0 y CLK1.

Register 3. Output Enable Control

Bit	D7	D6	D5	D4	D3	D2	D1	D0
Name	CLK7_OEB	CLK6_OEB	CLK5_OEB	CLK4_OEB	CLK3_OEB	CLK2_OEB	CLK1_OEB	CLK0_OEB
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Reset value = 0000 0000

Bit	Name	Function
7:0	CLKx_OEB	Output Disable for CLKx. Where x = 0, 1, 2, 3, 4, 5, 6, 7 0: Enable CLKx output. 1: Disable CLKx output.

Figura 2.30: Registro de control de habilitación de salida

Paso 3: En la figura 2.31 y 2.32, se puede observar el registro de control de la señal de salida CLK0 y CLK1 respectivamente del Si5351. Para deshabilitar estas salidas, se debe escribir un uno en los bits CLK0_PDN y CLK1_PDN de cada registro.

Register 16. CLK0 Control

Bit	D7	D6	D5	D4	D3	D2	D1	D0
Name	CLK0_PDN	MISO_INT	MISO_SRC	CLK0_INV	CLK0_SRC[1:0]		CLK0_DRV[1:0]	
Type	R/W	R/W	R/W	R/W	R/W		R/W	

Reset value = 0000 0000

Bit	Name	Function
7	CLK0_PDN	Clock 0 Power Down. This bit allows powering down the CLK0 output driver to conserve power when the output is unused. 0: CLK0 is powered up. 1: CLK0 is powered down.
6	MISO_INT	Multisynth 0 Integer Mode. This bit can be used to force MISO into integer mode to improve jitter performance. Note that the fractional mode is necessary when a delay offset is specified for CLK0. 0: MISO operates in fractional division mode. 1: MISO operates in integer mode.
5	MISO_SRC	Multisynth source select for CLK0. 0: Select PLLA as the source for Multisynth. 1: Select PLLB (Si5351A/C only) or VCXO (Si5351B only) Multisynth.
4	CLK0_INV	Output Clock 0 Invert. 0: Output Clock 0 is not inverted. 1: Output Clock 0 is inverted.
3:2	CLK0_SRC[1:0]	Output Clock 0 Input Source. These bits determine the input source for CLK0. 00: Select the XTAL as the clock source for CLK0. This option bypasses both synthesis stages (PLL/VCXO & Multisynth) and connects CLK0 directly to the oscillator which generates an output frequency determined by the XTAL frequency. 01: Select CLKIN as the clock source for CLK0. This bypasses both synthesis stages (PLL/VCXO & Multisynth) and connects CLK0 directly to the CLKIN input. This essentially creates a buffered output of the CLKIN input. 10: Reserved. Do not select this option. 11: Select Multisynth 0 as the source for CLK0. Select this option when using the Si5351 to generate free-running or synchronous clocks.
1:0	CLK0_DRV[1:0]	CLK0 Output Rise and Fall Time Drive Strength Control. 00: 2 mA 01: 4 mA 10: 6 mA 11: 8 mA.

Figura 2.31: Registro de control de CLK0

Register 17. CLK1 Control								
Bit	D7	D6	D5	D4	D3	D2	D1	D0
Name	CLK1_PDN	MS1_INT	MS1_SRC	CLK1_INV	CLK1_SRC[1:0]		CLK1_IDRV[1:0]	
Type	R/W	R/W	R/W	R/W	R/W		R/W	

Reset value = 0000 0000

Bit	Name	Function
7	CLK1_PDN	Clock 1 Power Down. This bit allows powering down the CLK1 output driver to conserve power when the output is unused. 0: CLK1 is powered up. 1: CLK1 is powered down.
6	MS1_INT	MultiSynth 1 Integer Mode. When the MS1 divider is an even integer, this bit can be used to force MS1 into Integer mode to improve jitter performance. Not valid for odd integers. Note that the fractional mode is necessary when a delay offset is specified for CLK1. 0: MS1 operates in fractional division mode. 1: MS1 operates in Integer mode.
5	MS1_SRC	MultiSynth Source Select for CLK1. 0: Select PLLA as the source for MultiSynth0. 1: Select PLLB (Si5351A/C only) or VCXO (Si5351B only) MultiSynth0.
4	CLK1_INV	Output Clock 1 Invert. 0: Output Clock 1 is not inverted. 1: Output Clock 1 is inverted.
3:2	CLK1_SRC[1:0]	Output Clock 1 Input Source. These bits determine the input source for CLK1. 00: Select the XTAL as the clock source for CLK1. This option by-passes both synthesis stages (PLL/VCXO & MultiSynth) and connects CLK1 directly to the oscillator which generates an output frequency determined by the XTAL frequency. 01: Select CLKIN as the clock source for CLK1. This by-passes both synthesis stages (PLL/VCXO & MultiSynth) and connects CLK1 directly to the CLKIN input. This essentially creates a buffered output of the CLKIN input. 10: Select MultiSynth 0 as the source for CLK1. 11: Select MultiSynth 1 as the source for CLK1. Select this option when using the Si5351 to generate free-running or synchronous clocks.
1:0	CLK1_IDRV[1:0]	CLK1 Output Rise and Fall time / Drive Strength Control. 00: 2 mA 01: 4 mA 10: 6 mA 11: 8 mA

Figura 2.32: Registro de control de CLK1

Paso 4: En la figura 2.33, se puede observar el registro de configuración de la capacitancia del cristal del Si5351. Se debe escribir un uno en los dos bits más significativos de este registro. Al colocar ambos bits en uno, se selecciona una capacitancia de 10 pF. De esta forma, se configura de forma correcta la frecuencia del oscilador del dispositivo, que es de 25 MHz.

Register 183. Crystal Internal Load Capacitance								
Bit	D7	D6	D5	D4	D3	D2	D1	D0
Name	XTAL_CL[1:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Reset value = 11xx xxxx

Bit	Name	Function
7:6	XTAL_CL[1:0]	Crystal Load Capacitance Selection. These 2 bits determine the internal load capacitance value for the crystal. See the Crystal Inputs section in the Si5351 data sheet. 00: Reserved. Do not select this option. 01: Internal CL = 6 pF. 10: Internal CL = 8 pF. 11: Internal CL = 10 pF (default).
5:0	Reserved	Bits 5:0 should be written to 010010b.

Figura 2.33: Registro de configuración de capacitancia

Paso 5: En la figura 2.34, se puede observar el registro de entrada del PLL. Para configurarlo correctamente, se debe seleccionar el cristal de 25 MHz como la frecuencia de reloj de referencia para la entrada del PLLA y PLLB. Para ello, se tiene que escribir un uno en los bits 2 y 3 del registro.

Register 15. PLL Input Source								
Bit	D7	D6	D5	D4	D3	D2	D1	D0
Name	CLKIN_DIV[1:0]				PLLB_SRC	PLLA_SRC		
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Reset value = 0000 0000

Bit	Name	Function
7:6	CLKIN_DIV[1:0]	CLKIN Input Divider. Valid PLL input range is 10-40 MHz. If CLKIN is > 40 MHz, CLKIN input divider, CLKIN_DIV, must be used to bring the PLL input within the 10-40 MHz range. 00b: Divide by 1. 01b: Divide by 2. 10b: Divide by 4. 11b: Divide by 8.
5:4	Reserved	Leave as default.
3	PLLB_SRC	Input Source Select for PLLB. 0: Select the XTAL input as the reference clock for PLLB. 1: Select the CLKIN input as the reference clock for PLLB (Si5351C only).
2	PLLA_SRC	Input Source Select for PLLA. 0: Select the XTAL input as the reference clock for PLLA. 1: Select the CLKIN input as the reference clock for PLLA (Si5351C only).
1:0	Reserved	Leave as default.

Figura 2.34: Registro de fuente de entrada para el PLL

Paso 6: Estos parámetros se configuran en base a la ecuación (2), en la cual, se puede determinar el valor de a, b y c. Una vez que se hallan estos valores, se configuran mediante firmware las siguientes ecuaciones:

$$(4) \text{MSN}_x\text{P1}[17:0] = 128 \times a + \text{Floor}\left(128 \times \frac{b}{c}\right) - 512$$

$$(5) \text{MSN}_x\text{P2}[19:0] = 128 \times b - c \times \text{Floor}\left(128 \times \frac{b}{c}\right)$$

$$(6) \text{MSN}_x\text{P3}[19:0] = c$$

La ecuación (4) corresponde a un número de 18 bits que es una representación codificada del parámetro “d” de la parte fraccional de la ecuación (2). El registro 30 representa los bits 0 a 7, el registro 29 representa los bits 8 a 15 y los dos bits menos significativos del registro 28 representan los bits 16 a 17

La ecuación (5) corresponde a un número de 20 bits que es una representación codificada del parámetro “e” de la parte fraccional de la ecuación (2). El registro 33 representa los bits 0 a 7, el registro 32 representa los bits 8 a 15 y los cuatro bits menos significativos del registro 31 representan los bits 16 a 19.

La ecuación (6) corresponde a un número de 20 bits que es una representación codificada del parámetro “c” de la parte fraccional de la ecuación (3). Los registros del Si5351 son de 8 bits. Este parámetro se separa en tres partes. El registro 27 representa los bits 0 a 7, el registro 26 los bits 8 a 15 y los cuatro bits más significativos del registro 31 representan los bits 16 a 19.

Paso 7:

Estos parámetros se configuran en base a la ecuación (3), en la cual, se puede determinar el valor de d, e y f. Una vez que se hallan estos valores, se configuran mediante firmware las siguientes ecuaciones:

$$(4) \text{MSN}_x_P1[17:0] = 128 \times d + \text{Floor}\left(128 \times \frac{e}{f}\right) - 512$$

$$(5) \text{MSN}_x_P2[19:0] = 128 \times e - f \times \text{Floor}\left(128 \times \frac{e}{f}\right)$$

$$(6) \text{MSN}_x_P3[19:0] = f$$

La ecuación (4) corresponde a un número de 18 bits que es una representación codificada del parámetro “d” de la parte fraccional de la ecuación (2). El registro 46 representa los bits 0 a 7, el registro 45 representa los bits 8 a 15 y los dos bits menos significativos del registro 44 representa los bits 16 a 17

La ecuación (5) corresponde a un número de 20 bits que es una representación codificada del parámetro “e” de la parte fraccional de la ecuación (2). El registro 49 representa los bits 0 a 7, el registro 48 representa los bits 8 a 15 y los cuatro bits menos significativos del registro 47 representa los bits 16 a 19.

La ecuación (6) corresponde a un número de 20 bits que es una representación codificada del parámetro “f” de la parte fraccional de la ecuación (3). Los registros del Si5351 son de 8 bits. Este parámetro se separa en tres partes. El registro 43 representa los bits 0 a 7, el registro 42 los bits 8 a 15 y los cuatro bits más significativos del registro 47 representa los bits 16 a 19.

Paso 8: En las figuras 2.35 y 2.36, se pueden observar los parámetros a configurar para el registro de control de salida de reloj cero (CLK0) y para el registro de control de salida de reloj uno (CLK1)

En este proyecto se configuraron los bits del registro CLK0 de forma tal que se encienda el controlador de CLK0, el divisor de salida múltiple sea fraccionario (OMD 0), la fuente de entrada al OMD 0 sea el PLLA, la salida de reloj CLK0 no esté invertida, la fuente de entrada para la salida CLK0 sea el OMD 0 y la corriente de salida sea de 8mA.

En este proyecto se configuraron los bits del registro CLK1 de forma tal que se encienda el controlador de CLK1, el divisor de salida múltiple sea fraccionario (OMD 1), la fuente de entrada al OMD 1 sea el PLLA, la salida de reloj CLK1 no esté invertida, la fuente de entrada para la salida CLK1 sea el OMD 1 y la corriente de salida sea de 8mA.

Register 16. CLK0 Control

Bit	D7	D6	D5	D4	D3	D2	D1	D0
Name	CLK0_PDN	MS0_INT	MS0_SRC	CLK0_INV	CLK0_SRC[1:0]		CLK0_IDRV[1:0]	
Type	R/W	R/W	R/W	R/W	R/W		R/W	

Reset value = 0000 0000

Bit	Name	Function
7	CLK0_PDN	Clock 0 Power Down. This bit allows powering down the CLK0 output driver to conserve power when the output is unused. 0: CLK0 is powered up. 1: CLK0 is powered down.
6	MS0_INT	MultiSynth 0 Integer Mode. This bit can be used to force MS0 into integer mode to improve jitter performance. Note that the fractional mode is necessary when a delay offset is specified for CLK0. 0: MS0 operates in fractional division mode. 1: MS0 operates in integer mode.
5	MS0_SRC	MultiSynth Source Select for CLK0. 0: Select PLLA as the source for MultiSynth0. 1: Select PLLB (SI5351A/C only) or VCXO (SI5351B only) MultiSynth0.
4	CLK0_INV	Output Clock 0 Invert. 0: Output Clock 0 is not inverted. 1: Output Clock 0 is inverted.
3:2	CLK0_SRC[1:0]	Output Clock 0 Input Source. These bits determine the input source for CLK0. 00: Select the XTAL as the clock source for CLK0. This option by-passes both synthesis stages (PLLVCXO & MultiSynth) and connects CLK0 directly to the oscillator which generates an output frequency determined by the XTAL frequency. 01: Select CLKIN as the clock source for CLK0. This by-passes both synthesis stages (PLLVCXO & MultiSynth) and connects CLK0 directly to the CLKIN input. This essentially creates a buffered output of the CLKIN input. 10: Reserved. Do not select this option. 11: Select MultiSynth 0 as the source for CLK0. Select this option when using the SI5351 to generate free-running or synchronous clocks.
1:0	CLK0_IDRV[1:0]	CLK0 Output Rise and Fall time / Drive Strength Control. 00: 2 mA 01: 4 mA 10: 6 mA 11: 8 mA

Figura 2.35: Registro de control de CLK0

Register 17. CLK1 Control								
Bit	D7	D6	D5	D4	D3	D2	D1	D0
Name	CLK1_PDN	MS1_INT	MS1_SRC	CLK1_INV	CLK1_SRC[1:0]		CLK1_IDRV[1:0]	
Type	R/W	R/W	R/W	R/W	R/W		R/W	

Reset value = 0000 0000

Bit	Name	Function
7	CLK1_PDN	Clock 1 Power Down. This bit allows powering down the CLK1 output driver to conserve power when the output is unused. 0: CLK1 is powered up. 1: CLK1 is powered down.
6	MS1_INT	MultiSynth 1 Integer Mode. When the MS1 divider is an even Integer, this bit can be used to force MS1 into Integer mode to improve jitter performance. Not valid for odd Integers. Note that the fractional mode is necessary when a delay offset is specified for CLK1. 0: MS1 operates in fractional division mode. 1: MS1 operates in Integer mode.
5	MS1_SRC	MultiSynth Source Select for CLK1. 0: Select PLLA as the source for MultiSynth0. 1: Select PLLB (SI5351A/C only) or VCXO (SI5351B only) MultiSynth0.
4	CLK1_INV	Output Clock 1 Invert. 0: Output Clock 1 is not inverted. 1: Output Clock 1 is Inverted.
3:2	CLK1_SRC[1:0]	Output Clock 1 Input Source. These bits determine the input source for CLK1. 00: Select the XTAL as the clock source for CLK1. This option by-passes both synthesis stages (PLL/VCXO & MultiSynth) and connects CLK1 directly to the oscillator which generates an output frequency determined by the XTAL frequency. 01: Select CLKIN as the clock source for CLK1. This by-passes both synthesis stages (PLL/VCXO & MultiSynth) and connects CLK1 directly to the CLKIN input. This essentially creates a buffered output of the CLKIN input. 10: Select MultiSynth 0 as the source for CLK1. 11: Select MultiSynth 1 as the source for CLK1. Select this option when using the SI5351 to generate free-running or synchronous clocks.
1:0	CLK1_IDRV[1:0]	CLK1 Output Rise and Fall time / Drive Strength Control. 00: 2 mA 01: 4 mA 10: 6 mA 11: 8 mA

Figura 2.36: Registro de control de CLK1

Paso 9: En la figura 2.37, se puede observar el registro de reinicio del PLL. Para configurar este registro, se debe escribir un uno en el bit siete para reiniciar el PLLB y un uno en el bit cinco para reiniciar el PLLA.

Register 177. PLL Reset								
Bit	D7	D6	D5	D4	D3	D2	D1	D0
Name	PLLB_RST		PLLA_RST					
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Reset value = 0000 0000

Bit	Name	Function
7	PLLB_RST	PLLB_Reset. Writing a 1 to this bit will reset PLLB. This is a self clearing bit.
6	Reserved	Leave as default.
5	PLLA_RST	PLLA_Reset. Writing a 1 to this bit will reset PLLA. This is a self clearing bit.
4:0	Reserved	Leave as default.

Figura 2.37: Registro de reinicio del PLL

Paso 10: En la figura 2.38, se puede observar el registro de habilitación de salida para las salidas de reloj CLK0 y CLK1. Para poder habilitarlas, se debe escribir un cero en los dos bits menos significativos de este registro.

Register 3. Output Enable Control								
Bit	D7	D6	D5	D4	D3	D2	D1	D0
Name	CLK7_OEB	CLK6_OEB	CLK5_OEB	CLK4_OEB	CLK3_OEB	CLK2_OEB	CLK1_OEB	CLK0_OEB
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Reset value = 0000 0000

Bit	Name	Function
7:0	CLKx_OEB	Output Disable for CLKx. Where x = 0, 1, 2, 3, 4, 5, 6, 7 0: Enable CLKx output. 1: Disable CLKx output.

Figura 2.38: Registro de habilitación de salida

Capítulo 3

Diseño e Implementación del hardware y software del sistema

3.1. Introducción

En este capítulo se describe el proceso de diseño del hardware y software del medidor de impedancias. Para el desarrollo del sistema completo de la figura 3.1, se debió desarrollar cuatro placas electrónicas. La primera hace referencia a la parte de electrónica digital programable y la segunda hace referencia a la parte de radiofrecuencia del proyecto. La primera placa está compuesta por el PIC18F2550, el generador de señal de reloj Si5351 y el conversor de USB a RS232. La segunda y tercera placa son idénticas y corresponden a dos filtros pasa bajos. La última placa corresponde a la parte del mezclador de radiofrecuencia y el filtro de audio de salida. En la figura 3.1, se puede observar el diagrama en bloques del sistema a implementar.

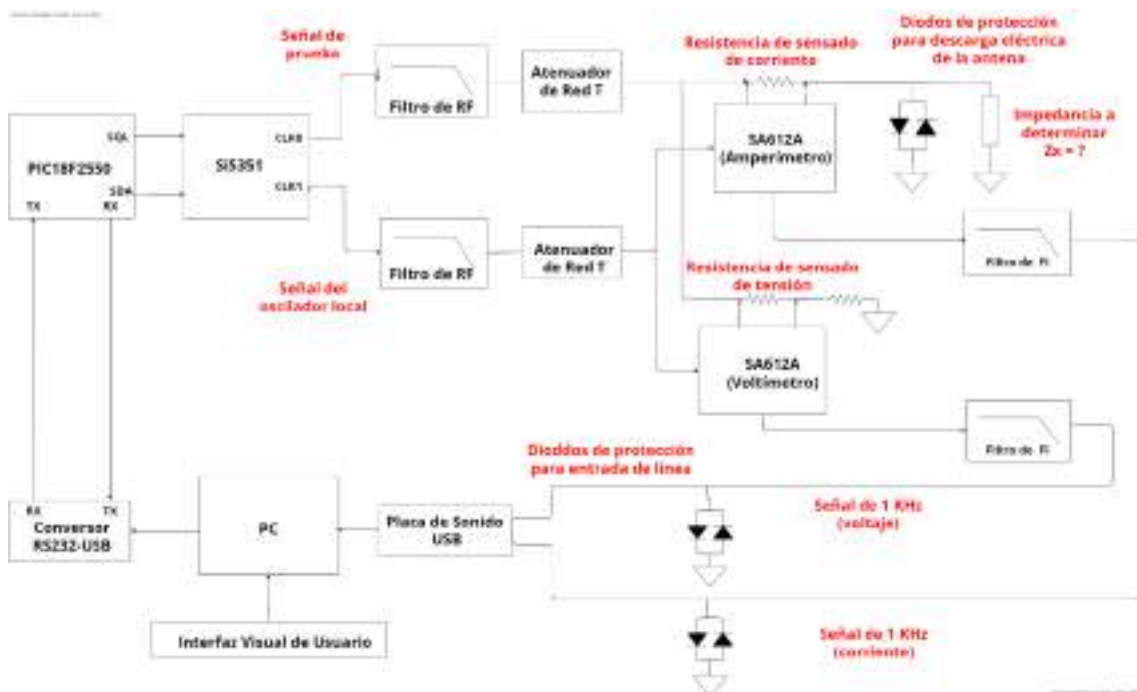


Figura 3.1: Diagrama en bloques del sistema completo

3.2. Diseño de la placa de adquisición del PIC18F2550

En la figura 3.2, se puede observar el esquemático empleado para el diseño de la placa de adquisición del PIC.

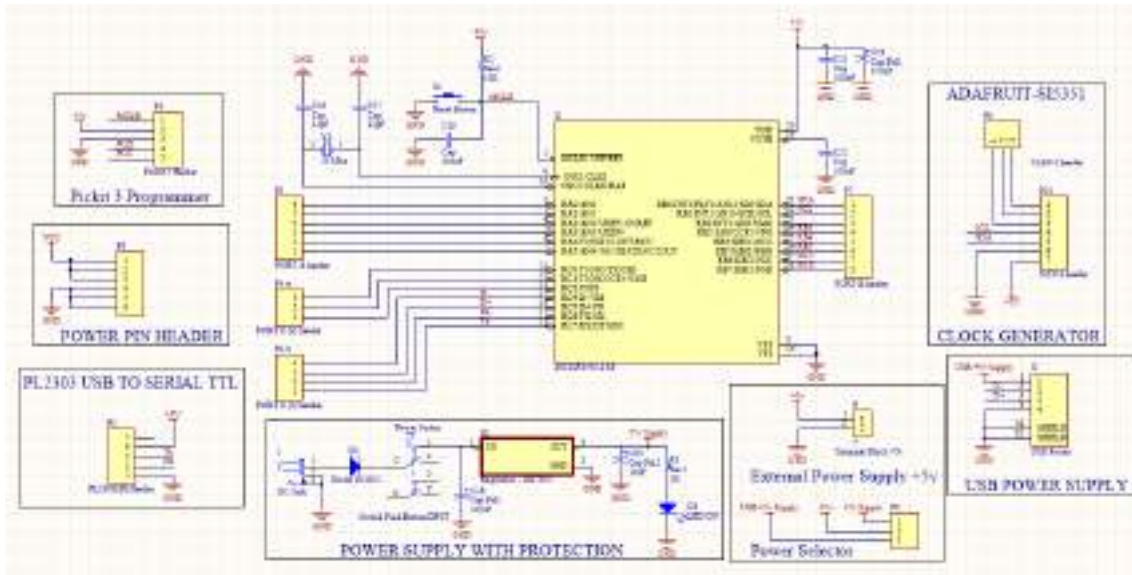


Figura 3.2: Esquemático de la placa de adquisición del PIC18F2550

Lo primero que se tuvo en cuenta a la hora de desarrollar esta placa fue la elección del microcontrolador a utilizar. Esto último, se tuvo en cuenta en base a todos los componentes del proyecto a utilizar. Como puede verse en el esquemático es necesario que el microcontrolador sea capaz de:

- Controlar el Si5351 mediante un módulo I2C.
- Enviar y recibir caracteres mediante puerto serie. Debe tener un módulo UART.
- Hacer uso de interrupciones para la recepción de caracteres.

Adicionalmente, en la figura 3.3, se puede observar las características principales del dispositivo.

PARÁMETROS	PIC18F2550
Tensión de Alimentación	5 V
Corriente máxima de entrada	250 mA
Corriente máxima de salida por todos los pines	200 mA
Corriente máxima de entrada en un pin I/O	25 mA
Corriente máxima de salida en un pin I/O	25 mA
Temperatura de almacenamiento	-65°C < T < +150°C
Potencia disipada total	1.0 W
Memoria FLASH	32768 bytes
Memoria SRAM	2048 bytes
Memoria EEPROM	256 bytes
Dimensiones	10.34 x 17.87 x 2.50 mm
Tipo de Encapsulado	PDIP

Figura 3.3: Características del PIC18F2550

La memoria de programa es de suma importancia para el tamaño del firmware. En este proyecto se consideró que como mínimo fuera necesario un microcontrolador con 32 kbyte de memoria de programa.

Una vez establecido el microcontrolador, se procedió a implementar la alimentación para el microcontrolador. Este trabaja con 5V. Se optó por utilizar tres fuentes de alimentación:

- Alimentación por USB
- Alimentación externa para hacer uso de una batería. Esto es para darle portabilidad al instrumento.
- Alimentación con regulador 7805 y protecciones.

3.2.1. Diseño e implementación del hardware del PIC18F2550

En la figura 3.4, se puede observar el diseño del hardware de la placa de adquisición del PIC18F2550. Luego, en la figura 3.5, se puede observar la plaqueta impresa con los componentes ya soldados.

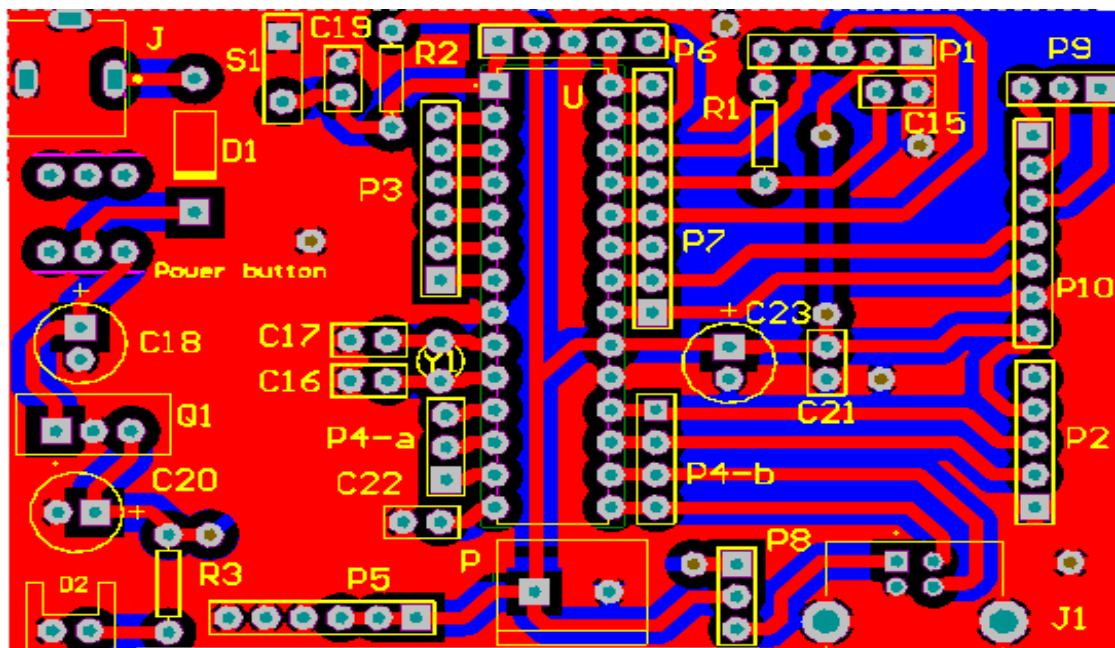


Figura 3.4: Diseño del hardware de la placa de adquisición del PIC18F2550

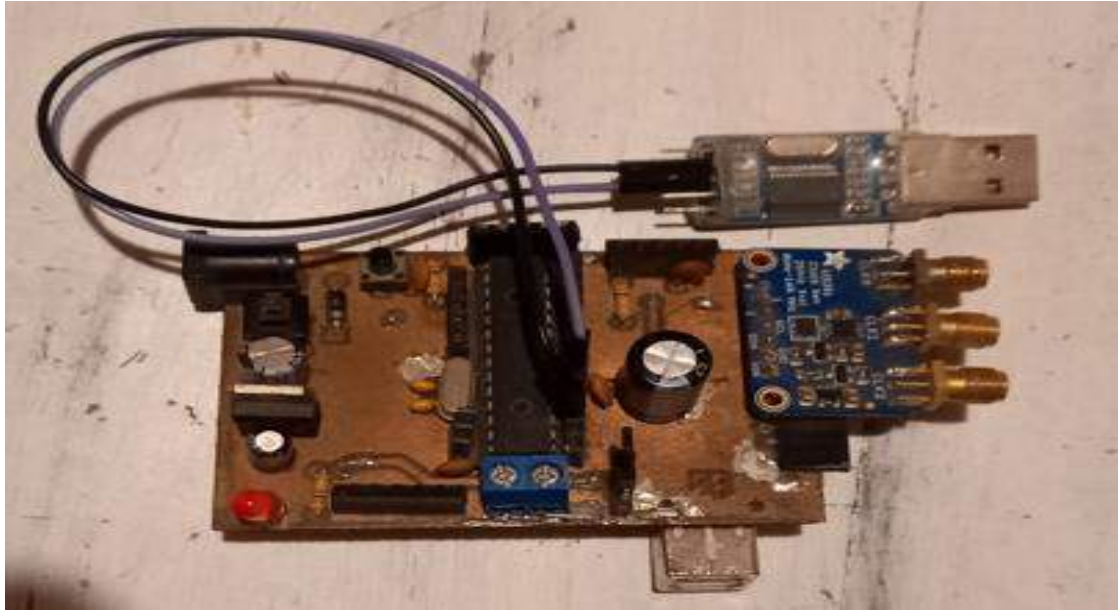


Figura 3.5: Placa de adquisición del PIC18F2550

3.3. Diseño e implementación de los filtros pasa bajos de RF

3.3.1. Caracterización de la señal de salida del Si5351

La señal de la salida del generador de onda cuadrada Si5351 es de 3,3 Vpp. Si se aplica la transformada de Fourier, se puede determinar las amplitudes de los armónicos. La fundamental, es decir, la primera armónica es la que se utilizará en este proyecto. Para ello, se deberán filtrar el resto de los armónicos para obtener una señal de prueba sinusoidal.

En la figura 3.6, se observa la señal de salida del Si5351 para una frecuencia de 1MHz. También, en la figura 3.7, se puede observar el análisis espectral de la señal mediante el analizador FFT del osciloscopio.

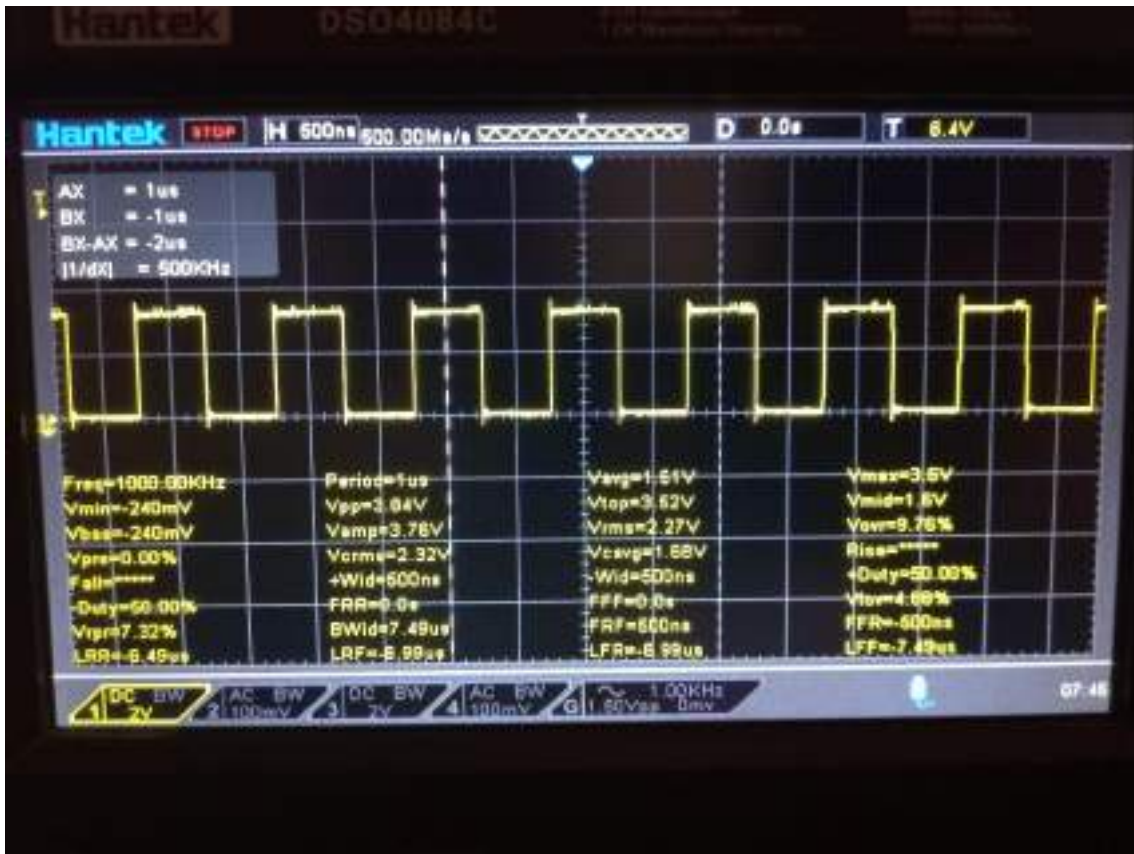


Figura 3.6: Señal de salida (CLK0) de 1 MHz

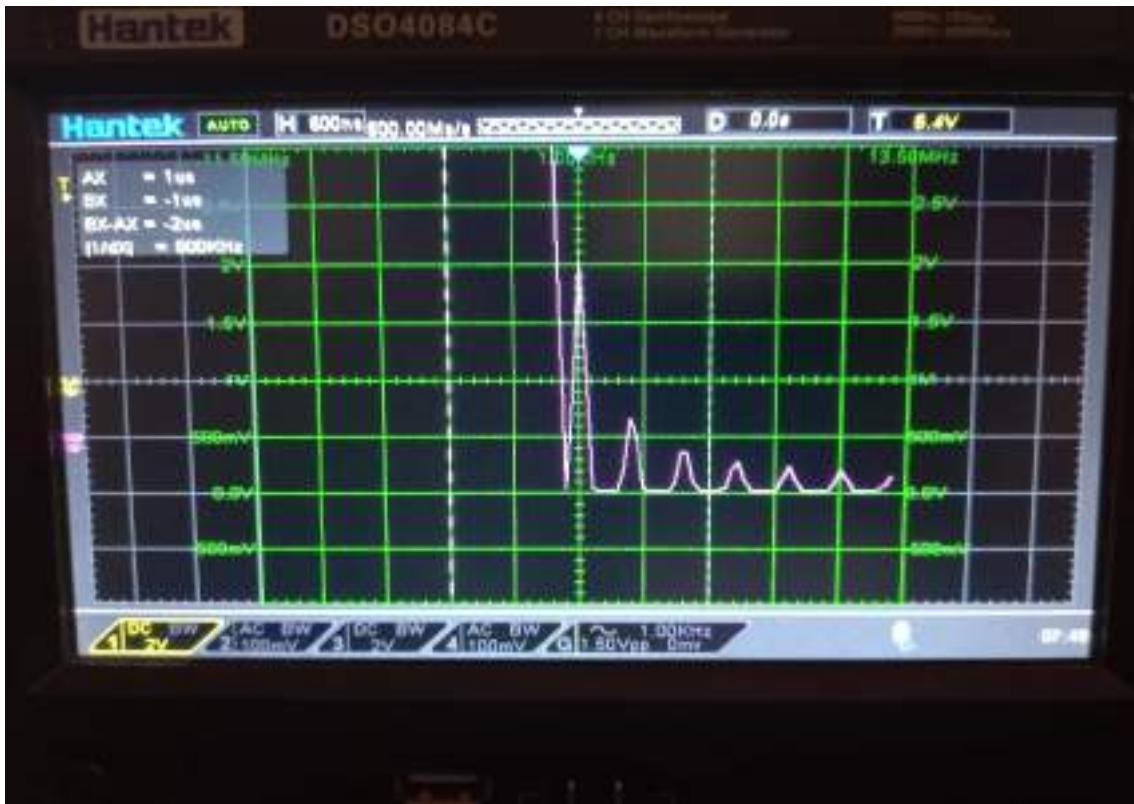


Figura 3.7: Espectro de la señal de salida (CLK0) de 1 MHz

En la figura 3.7, se puede observar que la amplitud RMS de la fundamental es de 2,1 V aproximadamente. Esto coincide con el análisis de Fourier. Dicho cálculo se puede observar en la siguiente ecuación:

$$A_1 \text{ (amplitud fundamental)} \\ = \frac{2 \times A_{pp} \text{ (amplitud pico a pico de la señal de onda cuadrada)}}{\pi}$$

$$A_1 = \frac{2 \times 3,3 \text{ V}}{\pi} \cong 2,1 \text{ V}$$

De la misma forma, se puede calcular la tercera armónica que aparece en 3 MHz (3fo). Se puede observar el cálculo en la siguiente ecuación:

$$A_3 \text{ (amplitud tercera armónica)} \\ = \frac{2 \times A_{pp} \text{ (amplitud pico a pico de la señal de onda cuadrada)}}{3 \times \pi}$$

$$A_3 = \frac{2 \times 3,3 \text{ V}}{3 \times \pi} \cong 0,7 \text{ V} \cong 700 \text{ mV}$$

El resto de los armónicos que aparecen en la figura 3.7, tienen los siguientes valores de amplitud RMS:

$$A_5 \cong 0,42 \text{ v}$$

$$A_7 \cong 0,3 \text{ v}$$

$$A_9 \cong 0,23 \text{ v}$$

$$A_{11} \cong 0,19 \text{ v}$$

$$A_{13} \cong 0,16 \text{ v}$$

$$A_{15} \approx 0,14 \text{ v}$$

Con estos valores se puede calcular la distorsión armónica total. La fórmula es la siguiente:

$$THD = \frac{\sqrt{\sum_{n=2}^{\infty} A_n \text{ (RMS)}}}{A_1}$$

$$THD = 0,449 \approx 45\%$$

3.3.2. Requisitos de diseño de los filtros de RF

En principio, el Si5351 puede generar una señal de onda cuadrada que va desde los 8 kHz hasta los 200 MHz. Debido a que genera una señal de onda cuadrada y no senoidal, se tuvo que hacer una limitación de banda. Esta limitación se debe a que, para poder caracterizar la impedancia incógnita en este proyecto, se tiene que aplicar una señal senoidal. El Si5351 contempla un amplio espectro de rango de frecuencia. En el presente trabajo su principal aplicación iba a ser para FM (frecuencia modulada). Debido al contexto de pandemia, el autor tuvo que hacer el diseño de los filtros en base al instrumental propio.

Este instrumental hace uso de un osciloscopio con un ancho de banda limitado a 80 MHz. En consecuencia, se decidió cambiar la limitación en banda para que el filtro corte en 52 MHz. De esta forma, el rango de trabajo del medidor de impedancias quedó definido de 30 a 50 MHz.

A modo de resumen, los filtros pasa bajos de radio frecuencia se diseñaron de forma tal que:

- La característica del filtro fuera del tipo Butterworth, para que no presente ripple en la banda de paso. Esto último, es fundamental para poder caracterizar la impedancia incógnita. Ya que, si la amplitud de la fundamental varía en el rango de frecuencia en el que funciona el instrumento, esto traerá aparejado una mala medición de la impedancia incógnita.
- El orden del filtro atenuara el tercer armónico presente en la onda cuadrada generada por el Si5351. Este armónico, es el más cercano a la fundamental y su vez, el de mayor amplitud. Se diseñó el filtro de RF de forma tal que atenúe por lo menos 20 dB el tercer armónico.
- El ancho de banda de estos se limite en 52 MHz. De esta forma, se pudo visualizar con el osciloscopio el comportamiento adecuado de los filtros en la banda de paso.

Para el diseño del filtro se usó la herramienta de simulación online “LC Filter Design Tool”. En la figura 3.8, se puede observar el diseño realizado mediante la herramienta.

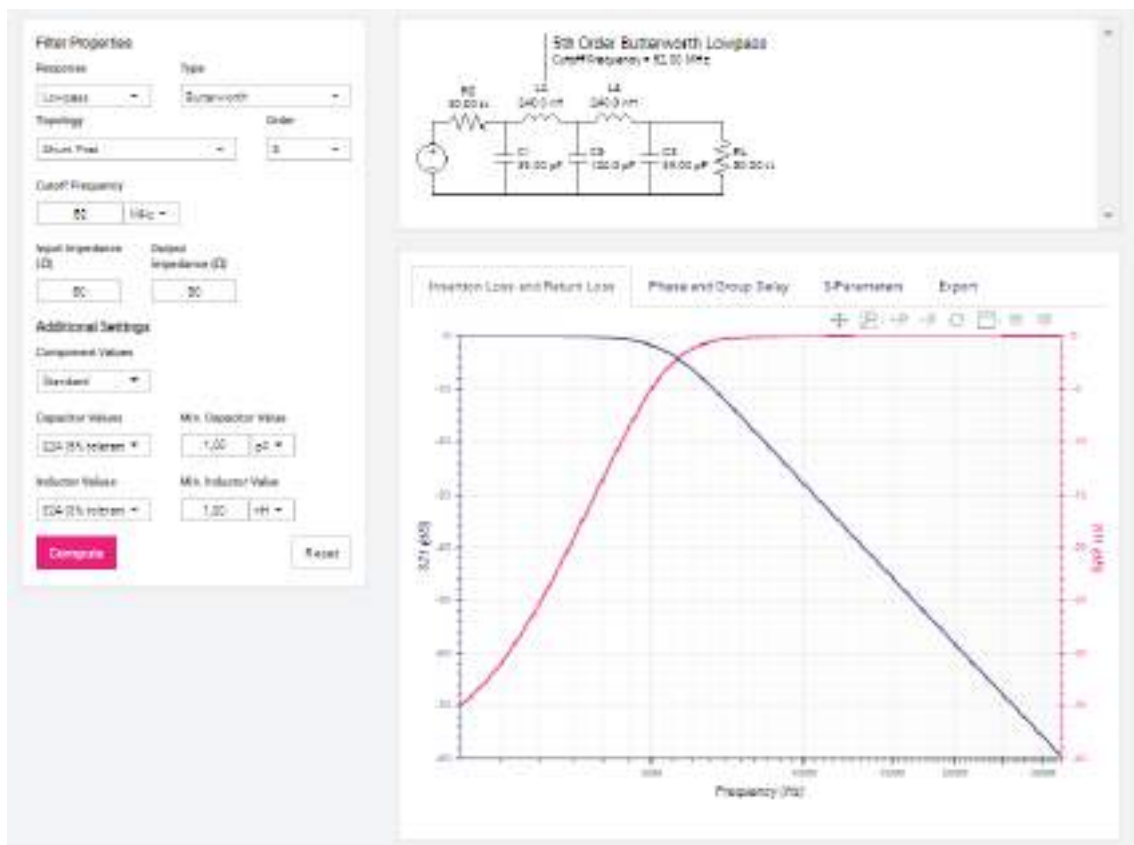


Figura 3.8: Diseño de filtro Butterworth de 5to orden

En las figuras 3.9 y 3.10, se puede observar la respuesta en frecuencia del filtro en magnitud y en fase respectivamente.

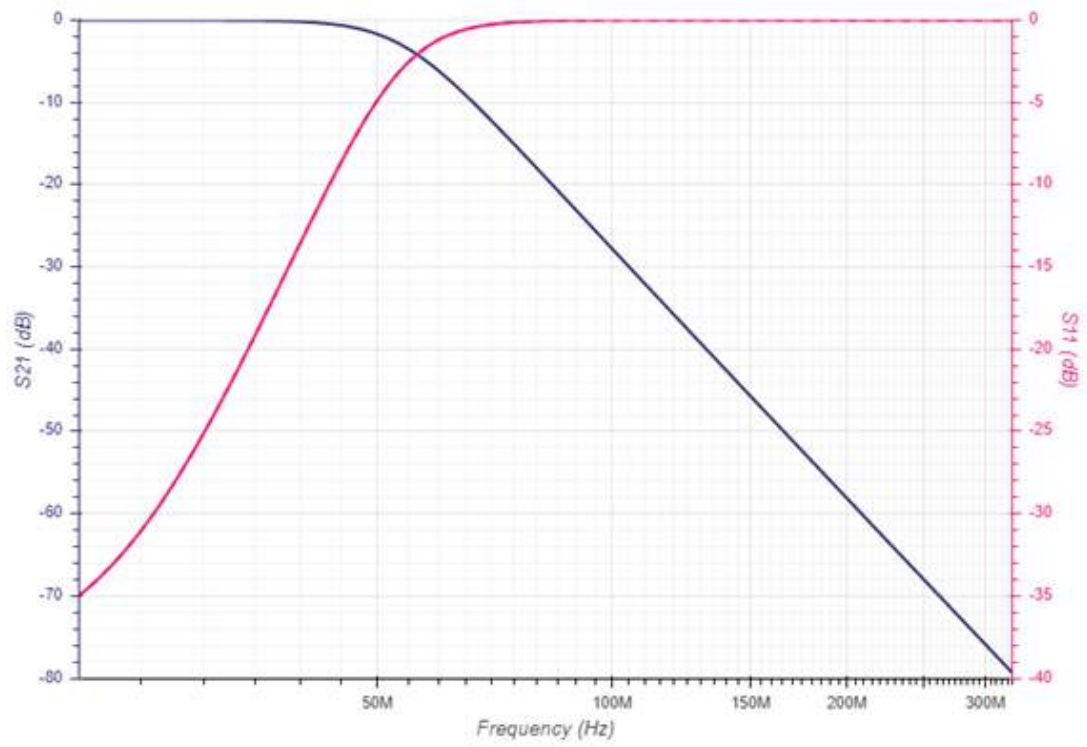


Figura 3.9: Respuesta en frecuencia del filtro de RF en magnitud

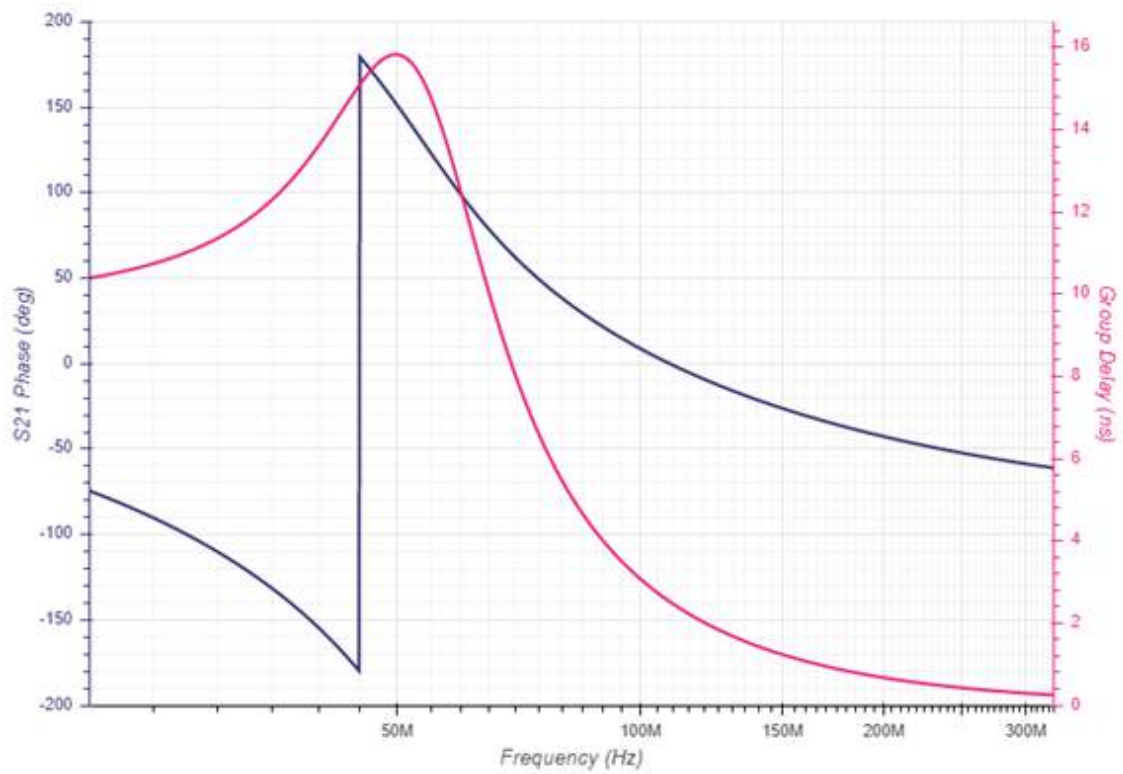


Figura 3.10: Respuesta en frecuencia del filtro de RF en fase

3.3.3. Implementación y validación del funcionamiento de los filtros pasa bajos de RF

En la figura 3.11, se puede observar el diseño del esquemático de los dos filtros pasa bajos de RF. En la figura 3.12, se puede observar el diseño del PCB de ambos filtros.

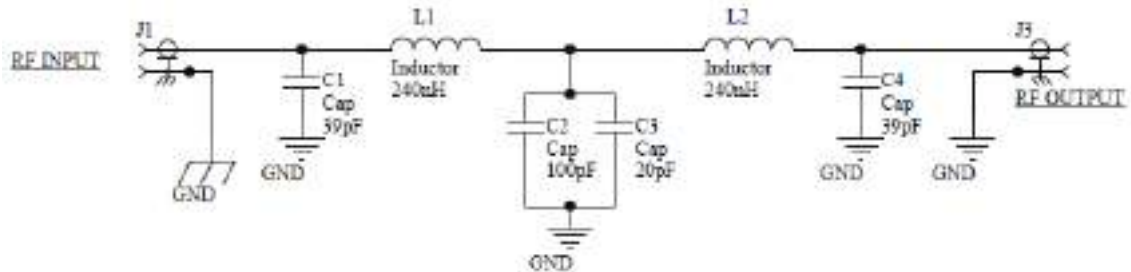


Figura 3.11: Diseño del esquemático de los filtros de RF

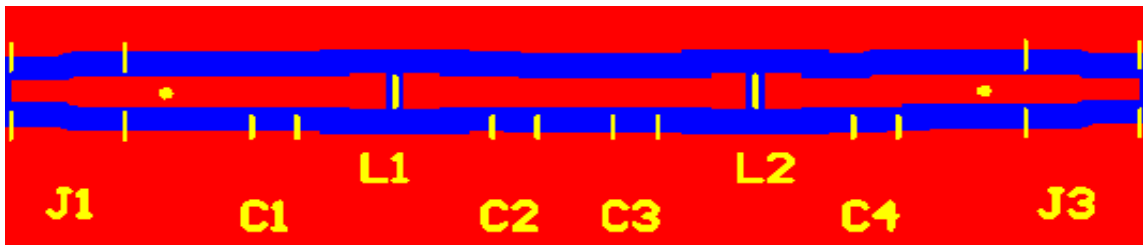


Figura 3.12: Diseño del pcb de los filtros de RF.

En la figura 3.13, se puede observar la implementación del circuito impreso de los filtros de RF con los componentes ya soldados.



Figura 3.13: Filtros de RF

Se verificó el funcionamiento del diseño de la figura 3.13 mediante el uso de la herramienta “NANO VNA”. Este dispositivo es un analizador de red vectorial que se encuentra en el laboratorio de comunicaciones (LAC) de la facultad de ingeniería de la UNMDP. Este instrumento tiene la ventaja, en comparación de un analizador de espectros, que puede dar información acerca de la característica de fase del sistema a medir. La información que registra este instrumento la guarda como un archivo “.txt”. En las figuras 3.14 y 3.15, se puede observar la información registrada por el dispositivo luego de realizar la medición de los filtros de RF.

#	MHz	S	MA	R	50	S21		S12		S22	
!	freq	S11			S21		S12		S22		
30.000	0.03128	126.30	0.93228	-98.96	0.93218	-98.95	0.03127	126.00			
33.367	0.07391	101.76	0.92108	-111.26	0.92094	-111.26	0.07379	101.91			
36.734	0.13037	82.84	0.90814	-124.15	0.90788	-124.14	0.13044	82.73			
40.101	0.20070	64.40	0.88951	-137.74	0.88971	-137.74	0.20086	64.41			
43.467	0.28504	45.69	0.86132	-152.14	0.86119	-152.15	0.28509	45.67			
46.834	0.37975	26.45	0.82076	-167.34	0.82052	-167.35	0.37982	26.41			
50.201	0.47944	6.91	0.76303	176.80	0.76327	176.81	0.47948	6.91			
53.568	0.57528	-12.73	0.68835	160.68	0.68827	160.70	0.57523	-12.69			
56.935	0.66054	-32.06	0.60116	145.01	0.60124	145.02	0.66055	-32.04			
60.302	0.73107	-50.66	0.51031	130.27	0.51030	130.30	0.73092	-50.67			
63.668	0.78712	-68.27	0.42423	116.91	0.42412	116.92	0.78679	-68.27			
67.035	0.82773	-84.70	0.34769	105.04	0.34784	105.04	0.82820	-84.72			
70.402	0.85770	-100.02	0.28394	94.63	0.28397	94.63	0.85760	-100.05			
73.769	0.87887	-114.27	0.23139	85.50	0.23147	85.51	0.87908	-114.27			
77.136	0.89411	-127.53	0.18858	77.72	0.18868	77.73	0.89437	-127.53			
80.503	0.90405	-139.88	0.15454	70.73	0.15462	70.73	0.90402	-139.90			
83.869	0.91051	-151.56	0.12719	64.49	0.12720	64.47	0.91077	-151.56			
87.236	0.91570	-162.59	0.10532	59.49	0.10530	59.47	0.91564	-162.57			
90.603	0.91888	-173.08	0.08768	54.40	0.08762	54.39	0.91899	-173.07			
93.970	0.92083	176.93	0.07322	49.74	0.07322	49.71	0.92117	176.92			
97.337	0.92307	167.37	0.06146	45.58	0.06145	45.58	0.92299	167.41			
100.704	0.92412	158.24	0.05170	41.66	0.05171	41.70	0.92464	158.26			
104.070	0.92582	149.47	0.04356	38.14	0.04360	38.09	0.92562	149.47			
107.437	0.92655	141.09	0.03679	34.98	0.03677	34.93	0.92670	141.09			
110.804	0.92730	132.97	0.03104	31.84	0.03101	31.84	0.92713	132.98			
114.171	0.92760	125.15	0.02622	29.25	0.02621	29.21	0.92771	125.14			
117.538	0.92836	117.59	0.02203	26.59	0.02208	26.74	0.92791	117.58			
120.905	0.92982	110.23	0.01856	24.46	0.01853	24.41	0.92977	110.26			

Figura 3.14: Parámetros “S” del primer filtro de RF

# MHz	S	MA	R	50	S21		S12		S22	
!	freq	S11	S21	S21	S12	S22	S12	S22	S12	S22
30.000	0.03112	138.14	0.93345	-99.38	0.93334	-99.34	0.03089	138.54		
33.367	0.07233	107.86	0.92220	-111.74	0.92181	-111.71	0.07193	107.75		
36.734	0.12808	86.62	0.90920	-124.66	0.90923	-124.65	0.12845	86.68		
40.101	0.19846	67.00	0.89059	-138.32	0.89079	-138.31	0.19876	67.00		
43.467	0.28357	47.45	0.86202	-152.84	0.86195	-152.81	0.28359	47.47		
46.834	0.37933	27.65	0.82161	-168.17	0.82133	-168.17	0.37929	27.66		
50.201	0.48028	7.66	0.76353	175.82	0.76322	175.85	0.48022	7.66		
53.568	0.57743	-12.31	0.68769	159.60	0.68735	159.60	0.57716	-12.30		
56.935	0.66363	-31.94	0.59934	143.77	0.59946	143.78	0.66391	-31.92		
60.302	0.73488	-50.75	0.50788	128.97	0.50780	128.98	0.73476	-50.72		
63.668	0.79115	-68.49	0.42134	115.55	0.42123	115.57	0.79115	-68.49		
67.035	0.83192	-85.01	0.34456	103.66	0.34461	103.68	0.83203	-85.00		
70.402	0.86111	-100.39	0.28101	93.25	0.28101	93.26	0.86103	-100.38		
73.769	0.88241	-114.62	0.22888	84.16	0.22887	84.19	0.88235	-114.61		
77.136	0.89725	-127.86	0.18643	76.42	0.18650	76.44	0.89736	-127.84		
80.503	0.90658	-140.21	0.15287	69.47	0.15283	69.47	0.90682	-140.20		
83.869	0.91323	-151.86	0.12586	63.24	0.12587	63.22	0.91374	-151.80		
87.236	0.91826	-162.82	0.10435	58.27	0.10431	58.28	0.91811	-162.83		
90.603	0.92082	-173.28	0.08698	53.19	0.08692	53.23	0.92073	-173.27		
93.970	0.92308	176.76	0.07274	48.56	0.07274	48.59	0.92292	176.79		
97.337	0.92473	167.27	0.06115	44.43	0.06115	44.45	0.92474	167.29		
100.704	0.92562	158.21	0.05162	40.58	0.05150	40.60	0.92399	158.18		
104.070	0.92636	149.42	0.04360	37.02	0.04359	37.03	0.92719	149.45		
107.437	0.92698	141.08	0.03690	33.89	0.03686	33.88	0.92719	141.09		
110.804	0.92794	133.00	0.03119	30.74	0.03124	30.83	0.92723	133.01		
114.171	0.92804	125.19	0.02645	28.18	0.02643	28.15	0.92821	125.21		
117.538	0.92854	117.66	0.02236	25.57	0.02239	25.64	0.92820	117.65		
120.905	0.92962	110.34	0.01888	23.42	0.01889	23.48	0.92942	110.37		

Figura 3.15: Parámetros “S” del segundo filtro de RF

En base a la información suministrada por el “NANO VNA”, se utilizaron las mediciones realizadas sobre el parámetro “S21” para recrear la respuesta en frecuencia y fase de ambos filtros. Para ello, se utilizó un programa en el lenguaje Python y se obtuvo la gráfica de la figura 3.16.

En la figura 3.16, se puede observar que el comportamiento en frecuencia y fase de los filtros es prácticamente idéntico. Esto es de suma importancia para poder realizar con buena exactitud la medición de la impedancia a determinar. Es fundamental que en el rango de frecuencia del instrumento (30-50 MHz) la armónica fundamental no sea atenuada y que, por lo menos, la tercera armónica tenga una atenuación mínima de 20 dB para la menor frecuencia de barrido del instrumento. Esto último se puede observar en las figuras 3.14 y 3.15, la atenuación de los filtros para la tercera armónica, cuya frecuencia es de 90 MHz, es la siguiente:

$$20 \times \log(0,08) = -21,94 \text{ dB}$$

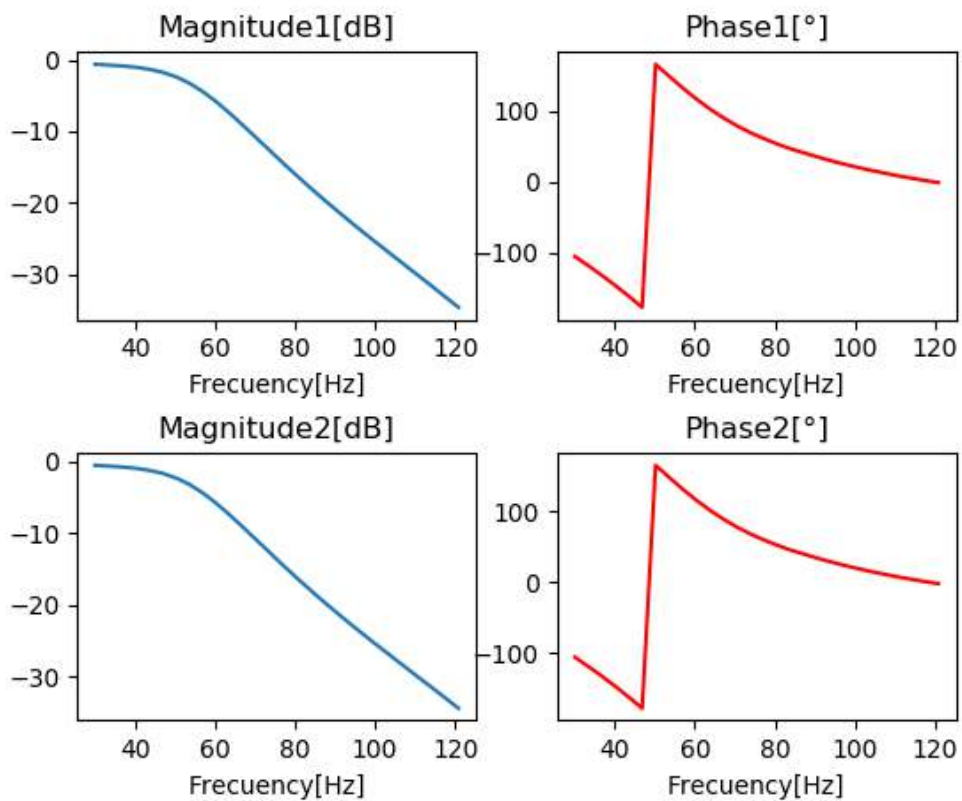


Figura 3.16: Comparación de la respuesta en frecuencia y fase de ambos filtros

También se verificó el comportamiento de los filtros con el osciloscopio “HANTEK DS04000C”. En las figuras 3.17 y 3.18, se puede observar la señal del generador de onda de cuadrada a la entrada y a la salida del filtro respectivamente. En la figura 3.18, se observa que la señal a la salida es del tipo sinusoidal. Esto da indicio de que el filtro atenuó los armónicos que corresponden. Además, se observa que la amplitud de la señal no se modificó prácticamente. Por lo tanto, el filtro tuvo el comportamiento adecuado en la banda de paso. En efecto, mantuvo la característica de máxima planicidad del filtro de tipo “Butterworth” del diseño.

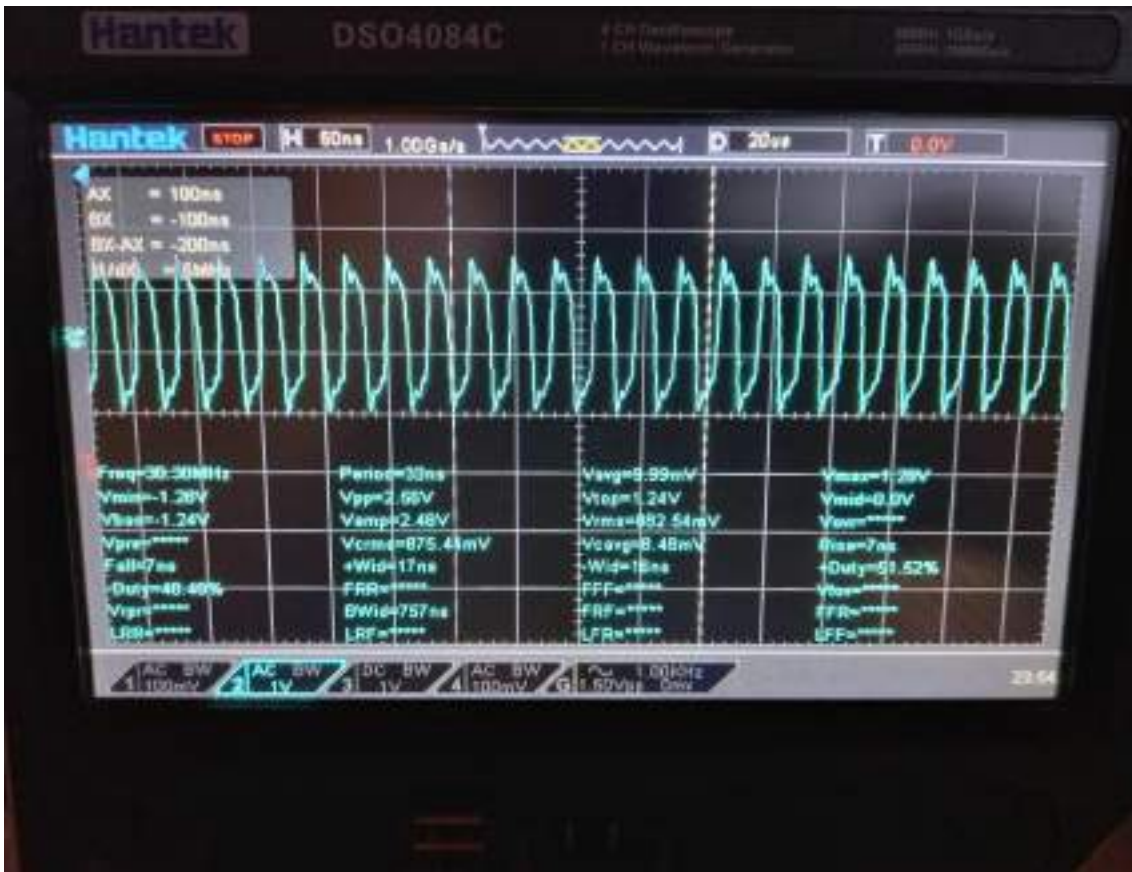


Figura 3.17: Señal de 30 MHz a la entrada del filtro de RF

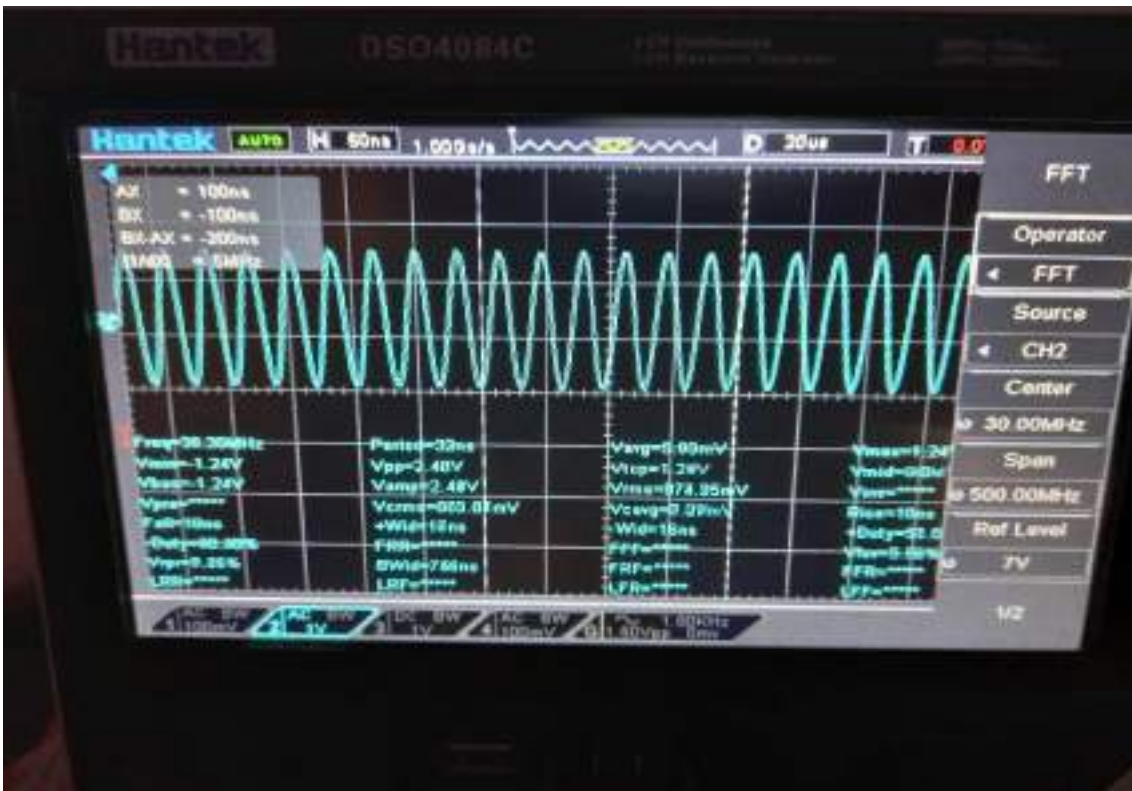


Figura 3.18: Señal de 30 MHz a la salida del filtro de RF

Adicionalmente, en las figuras 3.19 y 3.20, se pudo corroborar el contenido espectral de la señal a la entrada y a la salida del filtro de RF respectivamente, mediante el uso del analizador FFT del osciloscopio Hantek. Se observa que los armónicos presentes en la figura 3.20, se encuentran reducidos en amplitud más de diez veces su amplitud de la figura 3.19.



Figura 3.19: Contenido espectral de la señal de 30 MHz a la entrada del filtro de RF

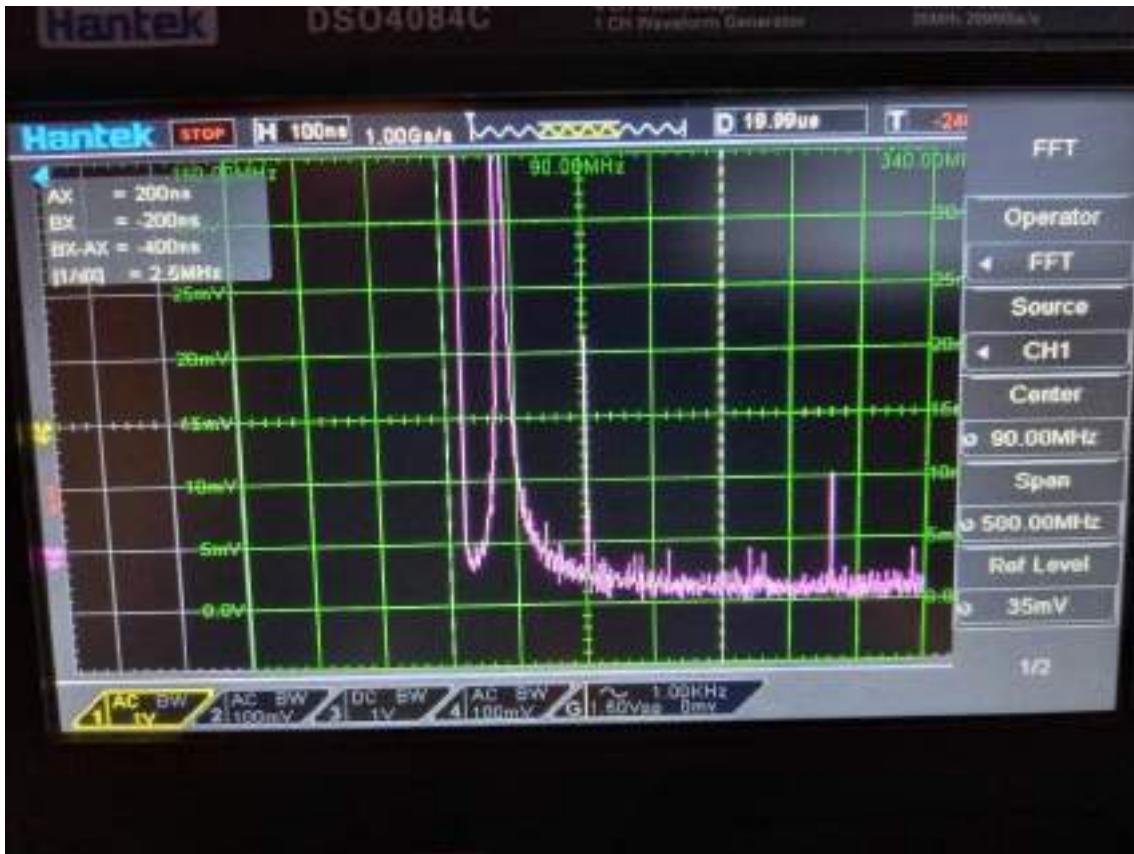


Figura 3.20: Contenido espectral de la señal de 30 MHz a la salida del filtro de RF

3.4.1.1. Mezclador SA612A

El SA612A es un mezclador doble balanceado monolítico VHF de baja potencia con oscilador y regulador de voltaje. Está diseñado para sistemas de comunicaciones de bajo costo y bajo consumo con frecuencias de señal de hasta 500 MHz y frecuencias de oscilador local tan altas como 200 MHz. El mezclador es una configuración de multiplicador del tipo de "Célula Gilbert" que proporciona una ganancia de 14 dB o más a 45 MHz. A continuación, se puede observar el diagrama de la configuración de pines del SA612A.

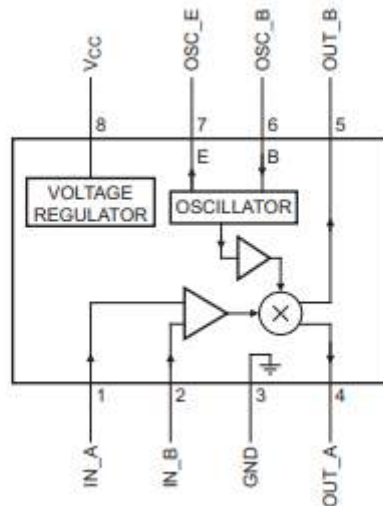


Figura 3.22: Diagrama en bloques del SA612A

Se eligió este mezclador en el presente trabajo ya que cubre las especificaciones del ancho de banda en el cual trabaja el generador de señal del reloj Si5351.

El SA612A está compuesto por una celda de Gilbert, un oscilador/búfer y una red de polarización con compensación de temperatura, como se muestra en la figura 3.23. La celda de Gilbert es un amplificador diferencial (entradas pines 1 y 2) que impulsa una celda de conmutación balanceada. La etapa de entrada diferencial proporciona ganancia y determina la cifra de ruido y el rendimiento del manejo de la señal del sistema.

Este mezclador está diseñado para un rendimiento óptimo de bajo consumo. El punto de intercepción de tercer orden es de aproximadamente -15 dBm (es decir, aproximadamente +5 dBm de intercepción de salida debido a la ganancia de RF). Se debe contemplar esta limitación de la señal, por lo tanto, se debe escalar la entrada del mezclador adecuadamente.

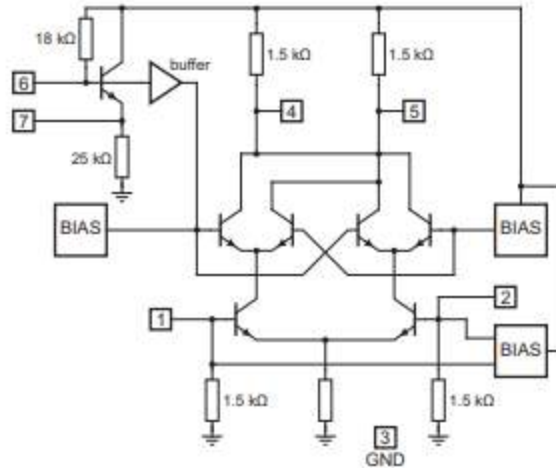


Figura 3.23: Circuito equivalente del SA612A

- Consideraciones:
 - i. Las entradas de RF (pines IN_A e IN_B) están polarizadas internamente. Son simétricas. La impedancia de entrada de corriente alterna equivalente es de aproximadamente $1,5 \text{ k}\Omega // 3 \text{ pF}$. Los pines IN_A e IN_B se pueden usar indistintamente, pero no deben tener una polarización de corriente continua externa.
 - ii. Las salidas del mezclador (pines OUT_A y OUT_B) también están polarizadas internamente. Cada salida está conectada al suministro positivo interno mediante una resistencia de $1,5 \text{ k}\Omega$. Esto permite la terminación de salida directa pero también permite una salida balanceada.
 - iii. La señal del oscilador local externo debe ser de 200 mV (pico a pico) mínimo hasta 300 mV (pico a pico) máximo. Esta señal se debe aplicar a través de un capacitor de acople en el pin 6 (OSC_B).

3.4.1.2. Puente de impedancia

En la figura 3.24, se observa el sistema de medición que se utilizó para obtener la impedancia incógnita. Los bloques que corresponde a los sensores hacen referencia a los mezcladores SA612A. Un mezclador se usa para determinar una tensión proporcional a la tensión aplicada a la carga "Z" y el otro para determinar una tensión proporcional a la corriente que circula por dicha carga. En la imagen se puede considerar a:

$$I_1 = I_2 = I_3 = I_4 \cong 0 \text{ A}$$

Esta aproximación se debe básicamente a que las entradas "IN_A" y "IN_B" de los integrados poseen una impedancia de entrada de $1,5\text{ k}\Omega // 3\text{ pF}$. La cual, es mucho mayor a las resistencias de sensado de tensión y corriente, R_4 y R_1 respectivamente. Por lo tanto, se puede considerar que la corriente que circula por la carga denominada " I_z " es aproximadamente igual a la corriente que circula por la resistencia R_1 . De la misma forma, la corriente que circula por R_4 es aproximadamente la misma que circula por R_5 .

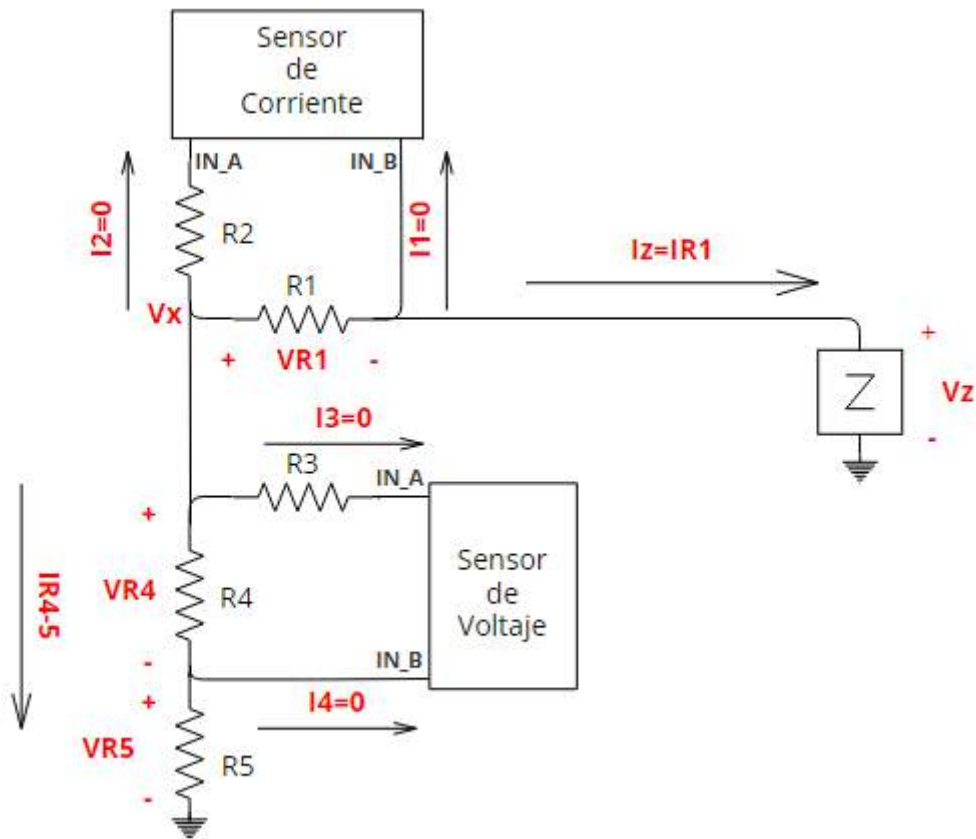


Figura 3.24: Sistema de medición de impedancia

3.4.1.3. Resistencias de sensado

En este proyecto se utilizaron los valores de resistencias:

$$R_1 = R_4 = 22\Omega$$

Estos valores no se eligieron de forma arbitraria, se debió tener en cuenta principalmente el rango dinámico en el cual funciona el integrado SA612A. En la figura 3.25, se puede observar el rango dinámico del SA612A.

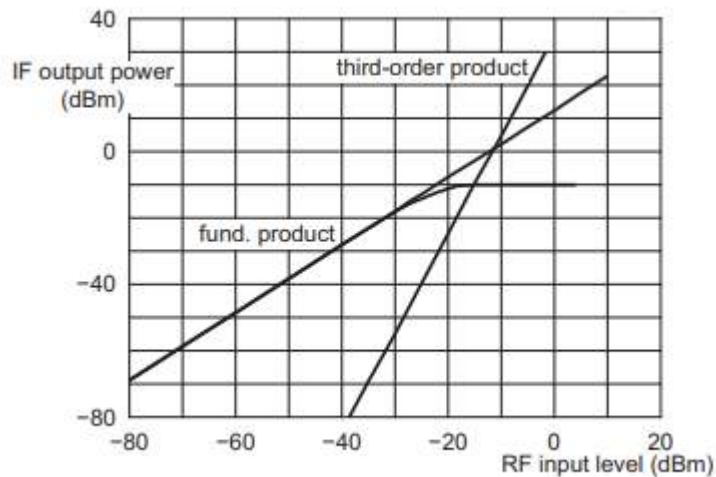


Figura 3.25: Punto de intercepción de tercer orden y compresión

De la figura 3.25 se puede observar que:

- El punto de intercepción de tercer orden es aproximadamente -15 dBm
- El punto de compresión en el cual el mezclador pierde su linealidad es aproximadamente -25 dBm

Adicionalmente, en la hoja de datos del SA612A se especifica que es capaz de recibir una señal de -119 dBm con una relación de señal a ruido de 12dB.

También, se tuvo en cuenta la amplitud pico a pico de la señal de salida. Es decir, la potencia de salida de frecuencia intermedia que se puede ver en la figura 3.25. Esto se debe a que posterior a la etapa de RF, se realizó un filtrado de audio para obtener la señal de frecuencia intermedia (FI) de 1kHz, para luego procesarla y realizar la adquisición a través de una placa de sonido USB. Esta placa posee una entrada de línea que soporta como máximo 0,91 V de amplitud pico a pico. Por lo tanto, se consideró esto último.

De estas consideraciones, se optó por trabajar entre -60 y -30 dBm el nivel de RF de la señal de entrada que mide el dispositivo. Con esto último, se seleccionaron los valores de resistencias R1 y R4 de forma tal que:

- El mezclador tenga suficiente sensibilidad.
- El mezclador no sature, es decir, se encuentre dentro de su rango dinámico.
- La amplitud pico a pico de la señal de salida sea menor a 0,91 V.

Para poder determinar el valor de la impedancia incógnita “Z”, se debe obtener Vz y Iz. Luego, se puede hallar la impedancia en módulo de la siguiente forma:

$$|Z| = \frac{|Vz|}{|Iz|}$$

La ecuación que define el valor de “Z” se obtiene a partir de las tensiones diferenciales que mide cada integrado (VR1 y VR4). A continuación, se muestra el planteo algebraico:

$$(a) IZ = IR1 = \frac{VR1}{R1} = \frac{VR1}{22 \Omega}$$

$$IR4 = \frac{VR4}{R4} = \frac{VR4}{22 \Omega}$$

$$VR5 = VX - VR4 = IR4 \times R5 = IR4 \times 100 \Omega$$

$$VX - VR4 = \frac{VR4}{22 \Omega} \times 100 \Omega$$

$$VX = \frac{VR4}{22 \Omega} \times 100 \Omega + VR4$$

$$VX = \left(\frac{100}{22} + 1 \right) \times VR4$$

$$(b) VX = \frac{122}{22} \times VR4$$

Si reemplaza (b) en (c) y (a) en (d),

$$(c) VZ = VX - VR1 = \frac{122}{22} \times VR4 - VR1$$

$$(d) VZ = IZ \times Z = \frac{VR1}{22 \Omega} \times Z$$

Se iguala (c) con (d),

$$\frac{122}{22} \times VR4 - VR1 = \frac{VR1}{22 \Omega} \times Z$$

Se despeja el valor de Z,

$$Z = \frac{22 \Omega}{VR1} \times \left(\frac{122}{22} \times VR4 - VR1 \right)$$

$$(e) |Z| = 122 \Omega \times \frac{VR4}{VR1} - 22 \Omega$$

Esta última ecuación, se implementará mediante software más adelante en el presente escrito. De esta forma, se podrá visualizar el valor de la impedancia incógnita en función de la frecuencia.

3.4.1.4. Adaptación de impedancias

Este punto fue de suma importancia en el presente proyecto. Esto se debe a que cumplió un rol fundamental en adaptar la impedancia de 50Ω de entrada a la impedancia equivalente del esquema de medición de impedancia. Con esto último, se logró que no haya onda reflejada. También, se utilizó esta adaptación para atenuar el nivel de voltaje de la señal de prueba de forma tal que los mezcladores trabajen en la zona lineal.

La señal de prueba es una señal sinusoidal de RF cuya amplitud y frecuencia dependen del Si5351. En este proyecto se utilizó un filtro de RF para obtener la primera armónica (A1), es decir, la fundamental. Esta tiene una amplitud de aproximadamente 2,1 volts. Se partió de este último valor para poder determinar la red T de atenuación y adaptación que mejor se ajuste. En la figura 3.26, se puede observar el esquemático de la adaptación de impedancias.

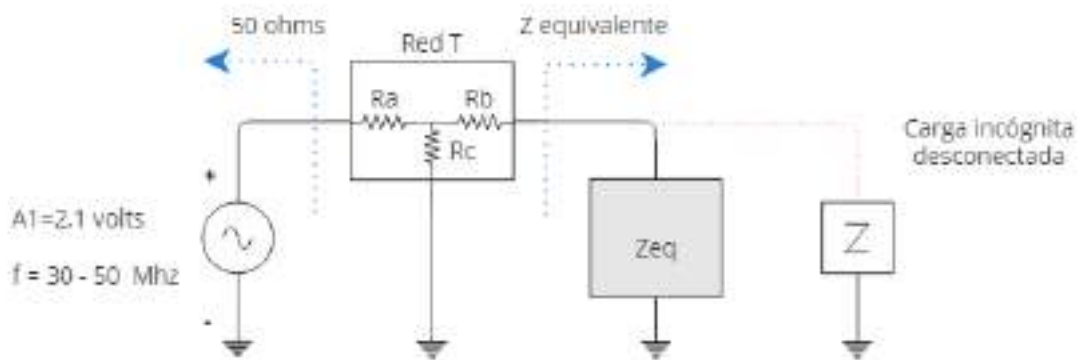


Figura 3.26: Esquemático del modelo de adaptación de impedancias

En la figura 3.27, se puede observar la impedancia equivalente que representa el bloque “Zeq”.

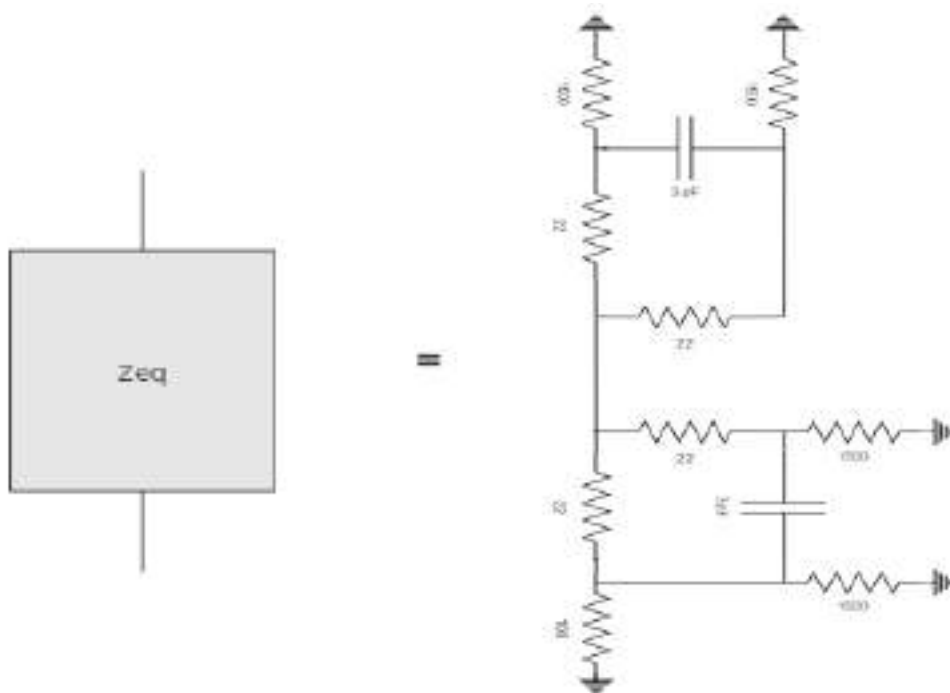


Figura 3.27: Impedancia equivalente del bloque Z_{eq}

Para poder hallar el valor de la impedancia equivalente se utilizó el software de simulación NL5. En la figura 3.28, se aplicó un generador con una tensión de prueba a la figura 3.27.

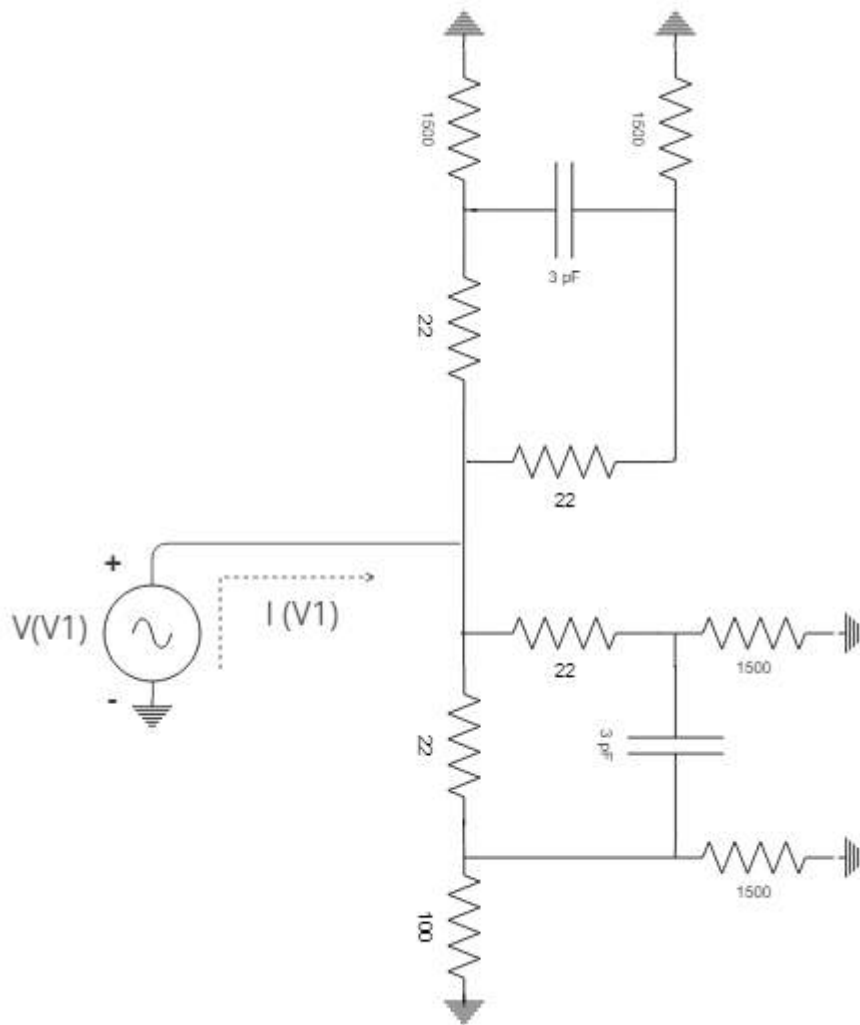


Figura 3.28: Determinación del valor de la Z_{eq}

En la figura 3.29, se puede observar la simulación de las señales $V(V1)$ y $I(V1)$ en el rango de funcionamiento del instrumento (30 a 50 MHz).

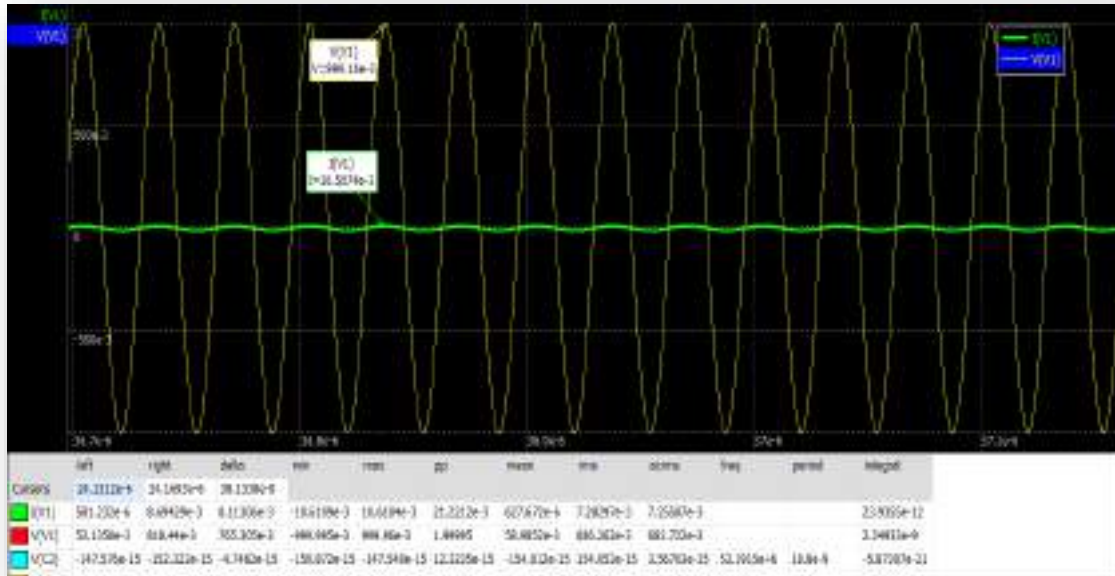


Figura 3.29: Simulación para obtener la Z_{eq}

Luego de la simulación, la impedancia equivalente puede calcularse a partir del valor pico de cada señal.

$$Z_{equivalente} = \frac{999,16 \text{ mV}}{10,5074 \text{ mA}} \cong 94 \Omega$$

En la figura 3.29, se puede observar que la tensión y la corriente están en fase. Esto da un indicio que la capacidad de 3 pF equivalente en la impedancia de entrada de cada mezclador no aporta fase en el rango de frecuencia del instrumento de 30 a 50 MHz. En la figura 3.30, se muestra el diagrama de bode de la respuesta en frecuencia para cada capacidad. Se observa que el aporte de fase del capacitor es considerable cuando la frecuencia es mayor a 100 MHz.

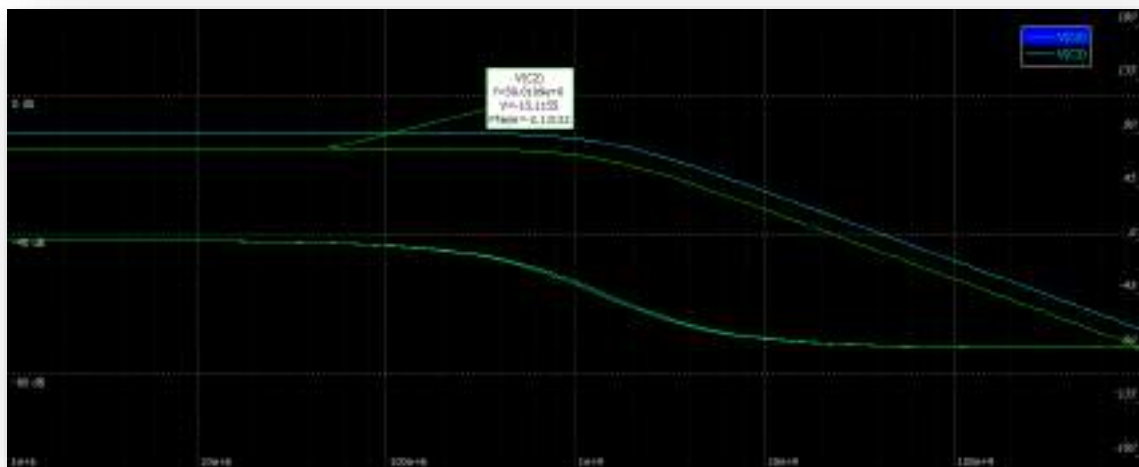


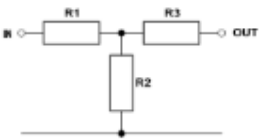
Figura 3.30: Respuesta en frecuencia de la impedancia de entrada del SA612A

Por último, se calcularon los valores de la red T de la figura 3.26 mediante el uso de una aplicación online “T Attenuator Calculator”. Esto último se puede observar en la figura 3.31. En el diseño de este proyecto se utilizaron los valores de resistencias de la serie E12. En efecto, quedaron definidos como:

$$R1 = 39 \Omega$$

$$R2 = 15 \Omega$$

$$R3 = 82 \Omega$$

Enter values and press ENTER or click the CALCULATE button			
	Attenuation:	<input type="text" value="20"/>	dB
	Input Impedance:	<input type="text" value="50"/>	Ohms
	Output Impedance:	<input type="text" value="94"/>	Ohms
	<input type="button" value="Calculate"/>		
Perfect values			
R1	<input type="text" value="37.1603"/>		Ohms
R2	<input type="text" value="13.8498"/>		Ohms
R3	<input type="text" value="82.0492"/>		Ohms

E12 Values		E12 Performance	
R1	<input type="text" value="39"/> Ohm	Attenuation	<input type="text" value="19.834"/> dB
R2	<input type="text" value="15"/> Ohm	Z _{in}	<input type="text" value="52.822"/> Ohm
R3	<input type="text" value="82"/> Ohm	Z _{out}	<input type="text" value="94.837"/> Ohm

Figura 3.31: Cálculo de los valores de la red T

Se diseñó la red T con una atenuación de 20 dB aproximadamente para lograr que la tensión que mida cada mezclador se encuentre dentro de su rango dinámico. Como se comentó anteriormente, luego de que la señal de onda cuadrada proveniente del Si5351 es filtrada, se obtiene una señal sinusoidal cuya amplitud es de aproximadamente 2,1 V. Luego esta señal pasa por la red T y se logra que la amplitud se reduzca aproximadamente diez veces. De esta forma se obtiene una señal cuya amplitud pico es de aproximadamente 0,21 V.

3.4.1.5. Determinación del rango dinámico del SA612A

En la figura 3.33, se utilizó el software “TopSpice” para simular cómo se comporta el sistema de medición de la figura 3.24 en base a el comportamiento lineal esperado y la exactitud del instrumento en el cálculo del valor de la impedancia incógnita. Mediante la simulación se puede calcular cual es el rango de impedancia en módulo que es capaz de medir el instrumento haciendo énfasis en su rango dinámico. En efecto, el dispositivo de medición entra en saturación a partir de un nivel de RF de su señal de entrada en aproximadamente -25 dBm. En base a esto último, se puede hallar la amplitud pico máxima diferencial que este es capaz de medir sin saturar. Cuanto menor es la impedancia conectada, mayor es la corriente que circula en el puente de medición y, por lo tanto, mayor es la tensión diferencial que miden los mezcladores entre sus entradas (2,4) y (6,5) identificadas en la figura 3.32. Si se tiene en cuenta esto último, la impedancia mínima que el instrumento es capaz de medir sin perder exactitud es 10 Ω . En la figura 3.33, se puede observar la simulación.

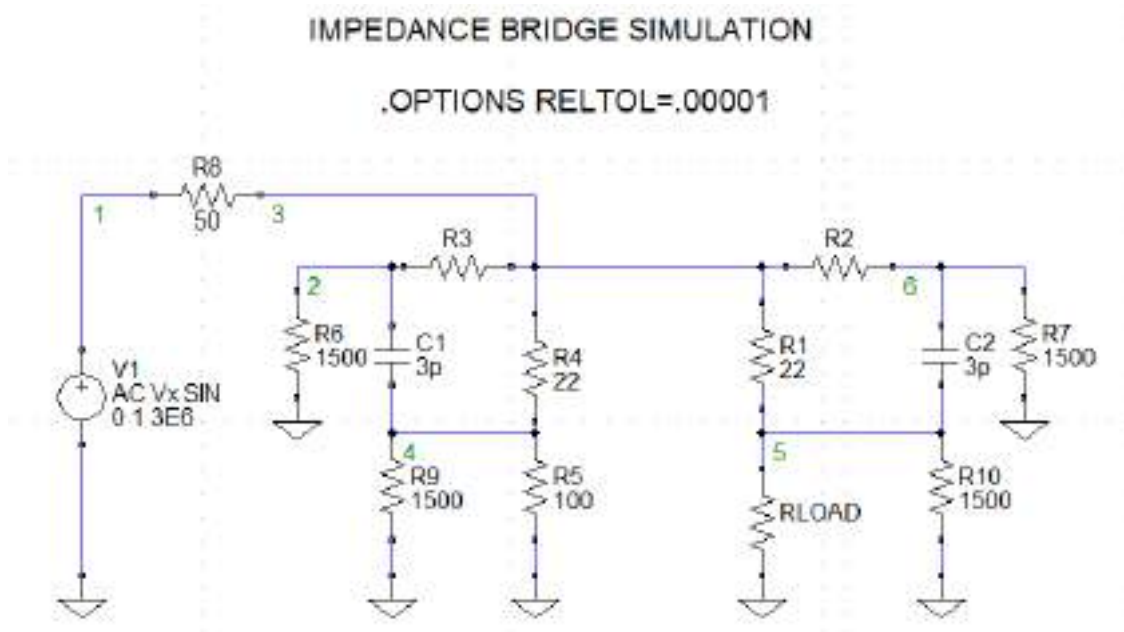


Figura 3.32: Esquemático del puente de medición de impedancias del SA612A

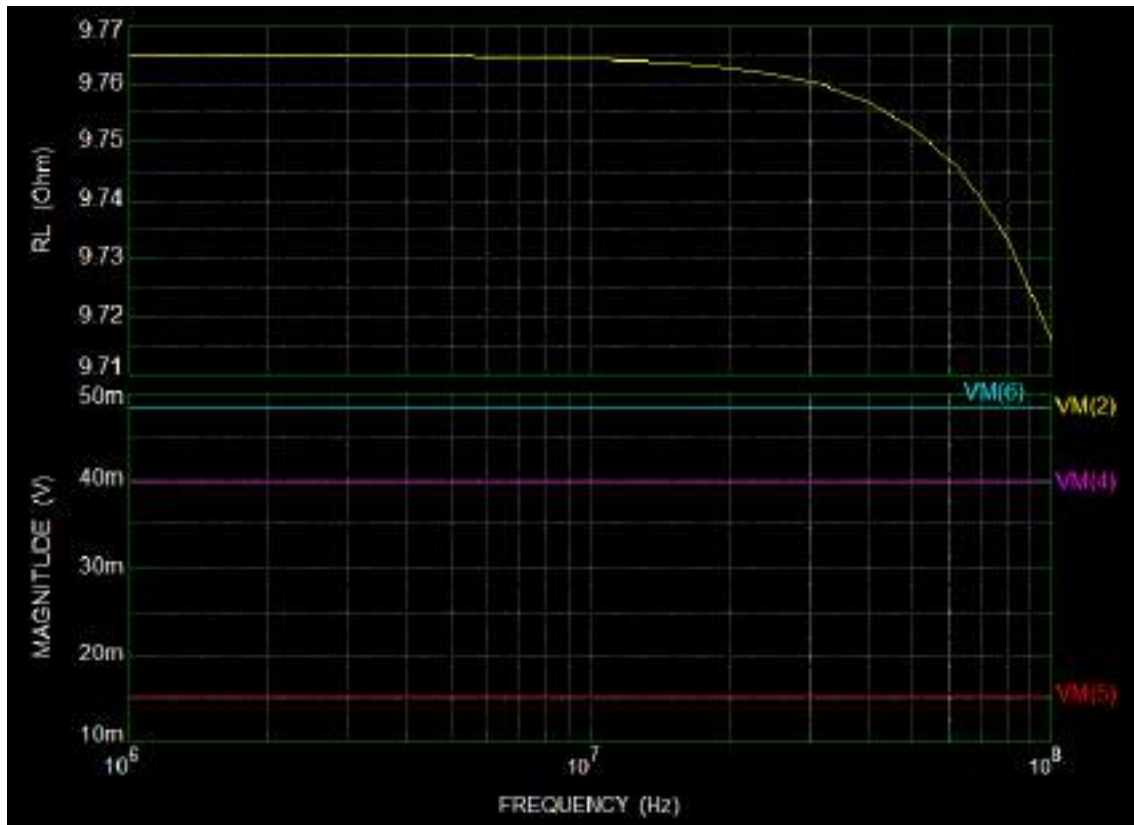


Figura 3.33: Simulación del puente de medición de impedancias

3.4.1.6. Resistencias de compensación

Las resistencias “R2” y “R3” de la figura 3.34, se diseñaron para lograr una mayor exactitud en la medición cuando se miden valores grandes de la impedancia incógnita (Z).

Inicialmente, el cálculo para determinar el valor de “Z” se desarrolló en la ecuación (e) del punto 3.4.1.3. del presente informe. En esta ecuación, se consideró que las corrientes denotadas como: “I2”, “I1”, “I3” e “I4” eran aproximadamente cero amperes. Esta aproximación a medida que el valor de la impedancia incógnita aumenta en magnitud pierde exactitud en la medición.

En principio, se consideró que R2 y R3 no eran necesarios en el esquema de medición, pero luego se llevó a cabo un análisis más detallado. Este consistió en analizar hasta que valor de impedancia el instrumento era capaz de medir con buena exactitud.

Debido a que la impedancia de entrada en cada mezclador (IN_A y IN_B) es de 1,5 k Ω , esto genera que haya una caída de tensión diferencial aplicada a la entrada del mezclador. Esto se debe a que una corriente pequeña circulará por la entrada de cada mezclador. Si la corriente que circula por las entradas de cada mezclador (I2, I1, I3 y I4) es muy chica en comparación con la corriente que circula por las resistencias de sensado R2 y R4, esto no generará error considerable en la medición de la impedancia. Pero a medida que la impedancia “Z” aumenta, la corriente que circula por el sistema de medición se hará cada vez más pequeña, esto traerá aparejado un error si no se incluyen en el diseño de medición a las resistencias de compensación R2 y R3.

Si ahora se colocan en las entradas IN_A de los mezcladores una resistencia cuyo valor es igual a la resistencia de sensado, esto genera que desde el nodo "Vx" se vea la misma resistencia para cada lado. Por lo tanto, la misma corriente circulará por R2 y R1, y de forma análoga, por R3 y R4. Esto hace que la tensión en la entrada IN_A sea igual a la tensión en la entrada IN_B para ambos mezcladores. En efecto, la tensión diferencial se cancela logrando así una mayor exactitud en el instrumento para valores de "Z" elevados.

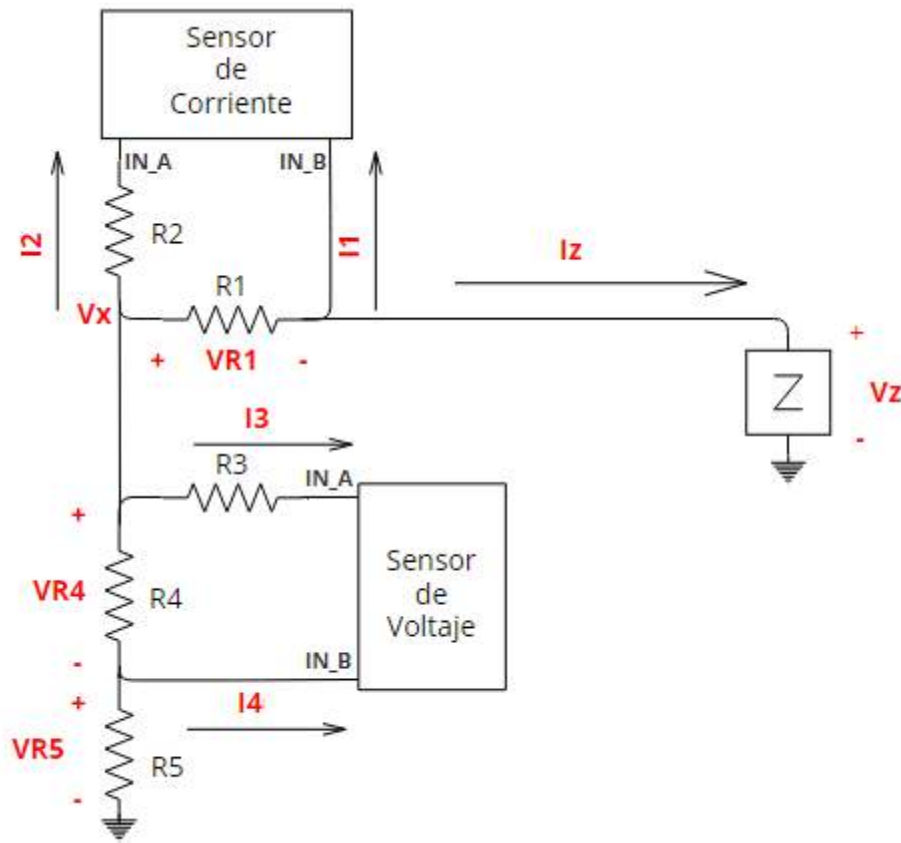


Figura 3.34: Sistema de medición de impedancia incógnita

3.4.1.7. Diseño de las protecciones para el integrado SA612A

En la figura... se observan distintos componentes conectados a la fuente de alimentación del mezclador. La resistencia de 22 Ω actúa como si fuera un "fusible" si se produce un cortocircuito en la fuente de 5 V. En la hoja de datos del SA612A se especifica una corriente de fuente típica de 2,4 mA. Si se produce un corto en la fuente, se obtiene que:

$$I_{\text{fuente}} = \frac{V_{\text{colector}}}{1500 \Omega} = \frac{3.6 \text{ v}}{1500 \Omega} = 2,4 \text{ mA}$$

Debido a que la resistencia de 22 Ω es mucho menor que la resistencia de colector de 1500 Ω, si se produce un corto y circula una corriente muy grande, esta hará que la resistencia de 22 Ω disipe una potencia mayor a 1 w. Con esto último, la misma se quemará abriendo el circuito y, en consecuencia, protegerá el integrado de una posible ruptura.

La bobina de 4,7 uH se encarga de filtrar el ruido proveniente del PIC. De esta forma, se evita que afecte el rendimiento del mezclador.

Los capacitores de desacople se usan para mantener una fuente de tensión estable dentro del rango recomendado del integrado y además eliminan cualquier señal de alta frecuencia que entre en la fuente de alimentación. Básicamente, elimina esta última presentando un camino de baja impedancia a tierra. Se usaron tres capacitores de distinto valor para poder tener un mayor rango de frecuencia, en el cual, el arreglo de banco de capacitores presente en su totalidad un camino de baja impedancia.

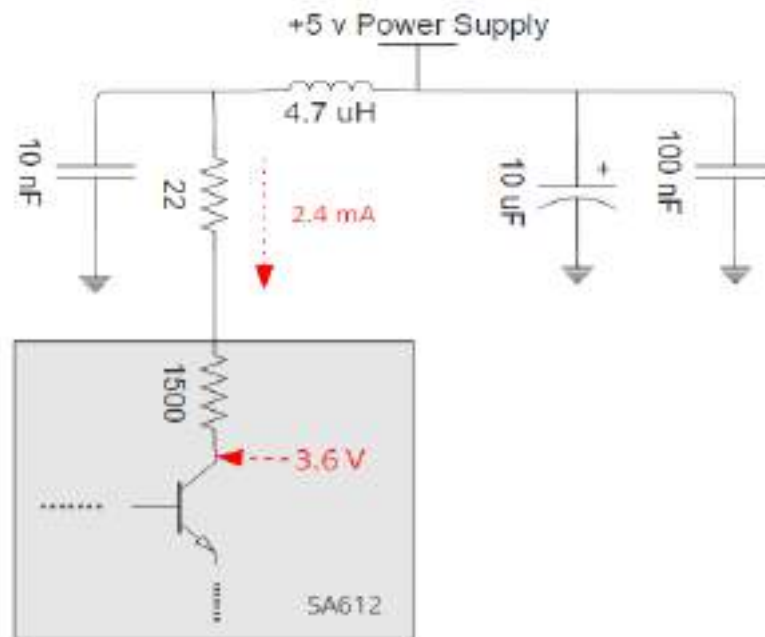


Figura 3.35: Protecciones del integrado SA612A

3.4.2. Diseño de la segunda etapa de audio del medidor de impedancias

3.4.2.1 Configuración de los filtros elípticos de octavo orden MAX7400

En la figura 3.36, se muestra el diagrama esquemático de la segunda etapa de audio del medidor de impedancias. Esta etapa se encarga de filtrar la señal de RF a la salida del mezclador SA612, para obtener las dos señales de audio de tensión y corriente de 1 kHz. Para poder llevar esto último a cabo, se usaron los filtros MAX7400. Estos filtros son del tipo elíptico pasa bajos de octavo orden que proporcionan una atenuación pronunciada con un buen rechazo en la banda de parada. Funcionan con una relación de frecuencia de reloj a corte de 100:1 y la frecuencia de corte máxima es de 10 kHz. En la figura 3.37, se puede observar la respuesta del filtro MAX7400.

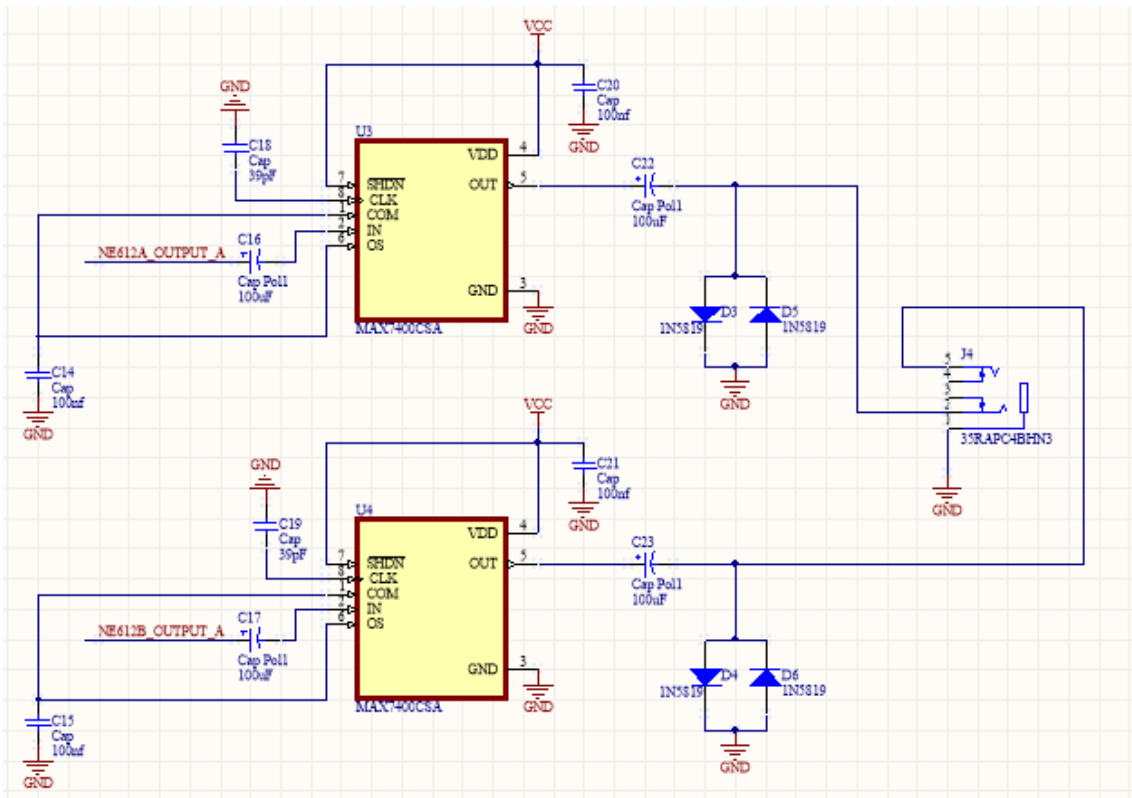


Figura 3.36: Esquemático de la segunda etapa de audio del medidor de impedancias

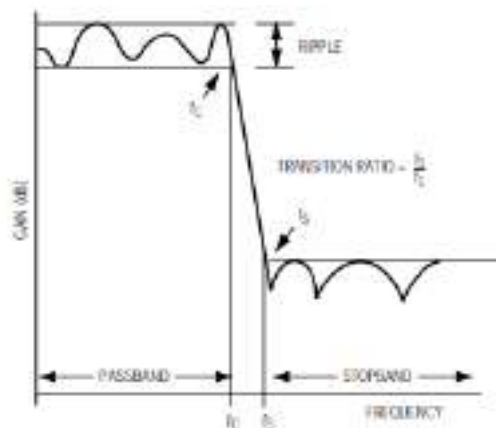


Figura 3.37: Respuesta del filtro elíptico de 8vo orden MAX7400

Para diseñar la frecuencia de corte se hizo uso del oscilador interno del integrado. Esta frecuencia se obtiene a través de la siguiente ecuación especificada en la hoja de datos del MAX7400:

$$F_{\text{oscilador}} \text{ (kHz)} = \frac{38 \times 10^3}{C_{\text{oscilador}} \text{ (pF)}}$$

En este proyecto se seleccionó un capacitor de 39 pF para configurar la frecuencia del oscilador. De esta forma se obtuvo:

$$F_{\text{oscilador}} (\text{kHz}) = 974,35 \text{ kHz}$$

Luego, la frecuencia de corte del filtro quedó definida como:

$$F_{\text{corte}} = \frac{F_{\text{oscilador}}}{100} = 9,74 \text{ kHz}$$

En la figura 3.38, se observa el esquemático para la correcta configuración del integrado. En la señal de reloj denotada como "CLOCK" se coloca el capacitor de 39 pF a masa para obtener la frecuencia de corte deseada.

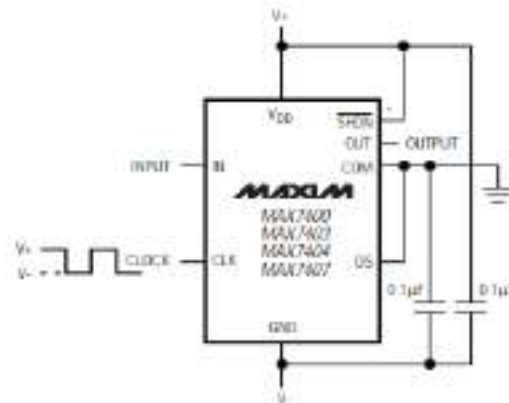


Figura 3.38: Esquema de configuración del MAX7400

3.4.2.2 Diodos de protección para entrada de línea

Estos diodos se usan para proteger la entrada de línea de la placa de sonido USB. Para la selección del diodo adecuado se tuvo en cuenta el máximo nivel de señal de entrada que puede soportar la entrada de línea de la placa de sonido USB. En la figura 3,39, se observa que la señal de entrada de línea para el consumidor soporta como máximo una tensión de aproximadamente 0,9 Vpp (-10 dBV). En base a esto último, se decidió utilizar el diodo 1N5819. Este diodo es del tipo Schottky, por lo tanto, presenta una tensión de forward (V_f) muy pequeña. En la figura 3.40, se observa las características eléctricas del componente. El diodo entra en conducción para una tensión de forward de 0,34 V. Debido a esto último y a la configuración antiparalelo de la figura 3.36 en la entrada de línea, si la señal supera los $\pm 0,34$ V de amplitud pico, los diodos entrarán en conducción y recortarán la amplitud de la señal. De esta forma, protegerán de una posible ruptura la placa de sonido USB.

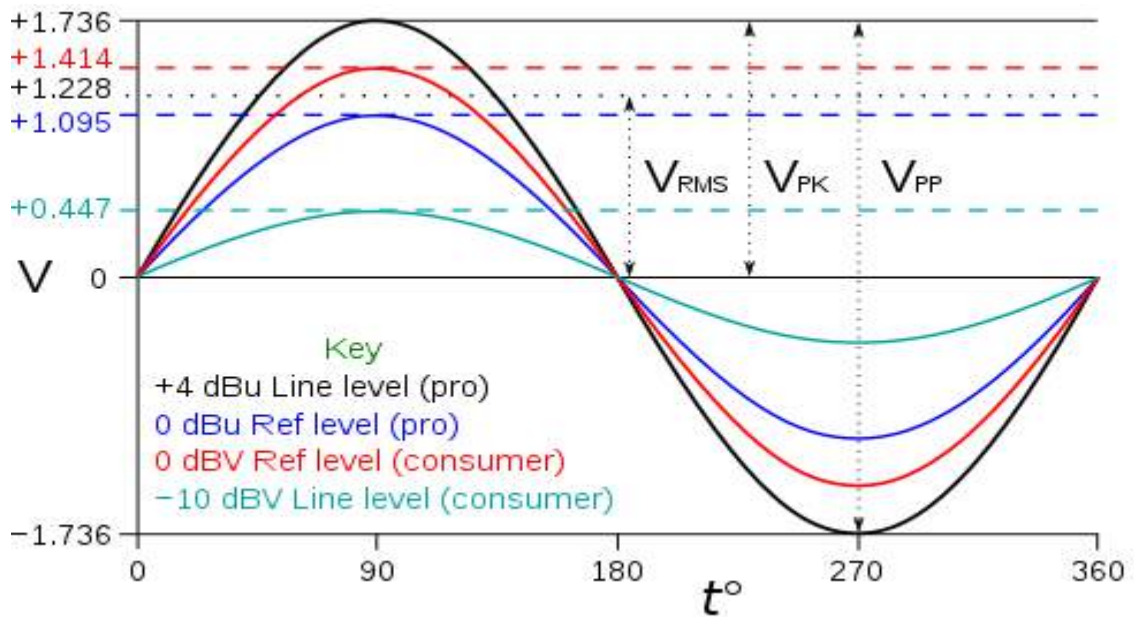


Figura 3.39: Nivel de voltaje máximo para la señal de audio en la entrada de línea

Characteristic	Symbol	1N5817	1N5818	1N5819	Unit	
Maximum Instantaneous Forward Voltage (Note 2)	($I_F = 0.1 \text{ A}$)	0.32	0.33	0.34	V	
	($I_F = 1.0 \text{ A}$)	0.45	0.55	0.5		
	($I_F = 3.0 \text{ A}$)	0.75	0.875	0.9		
Maximum Instantaneous Reverse Current @ Rated dc Voltage (Note 2)	I_R	($T_L = 25^\circ\text{C}$)	1.0	1.0	1.0	mA
		($T_L = 100^\circ\text{C}$)	10	10	10	

Figura 3.40: Características eléctricas del diodo 1N5819

3.4.2.3 Verificación del filtrado de la señal de RF a salida de los filtros MAX7400

En la figura 3.41, se puede observar la señal a la salida del mezclador SA612A o a la entrada del filtro MAX7400. Esta señal, para el ejemplo de la figura 3.41, está compuesta por la señal de RF de 30,001 MHz más la señal de audio de 1kHz. En la figura 3.42, se observa la señal de la figura 3.41 luego de ser filtrada por el filtro MAX7400. Se puede observar que esta señal es puramente sinusoidal sin presencia de otros armónicos.



Figura 3.41: Señal de RF a la entrada del filtro MAX7400

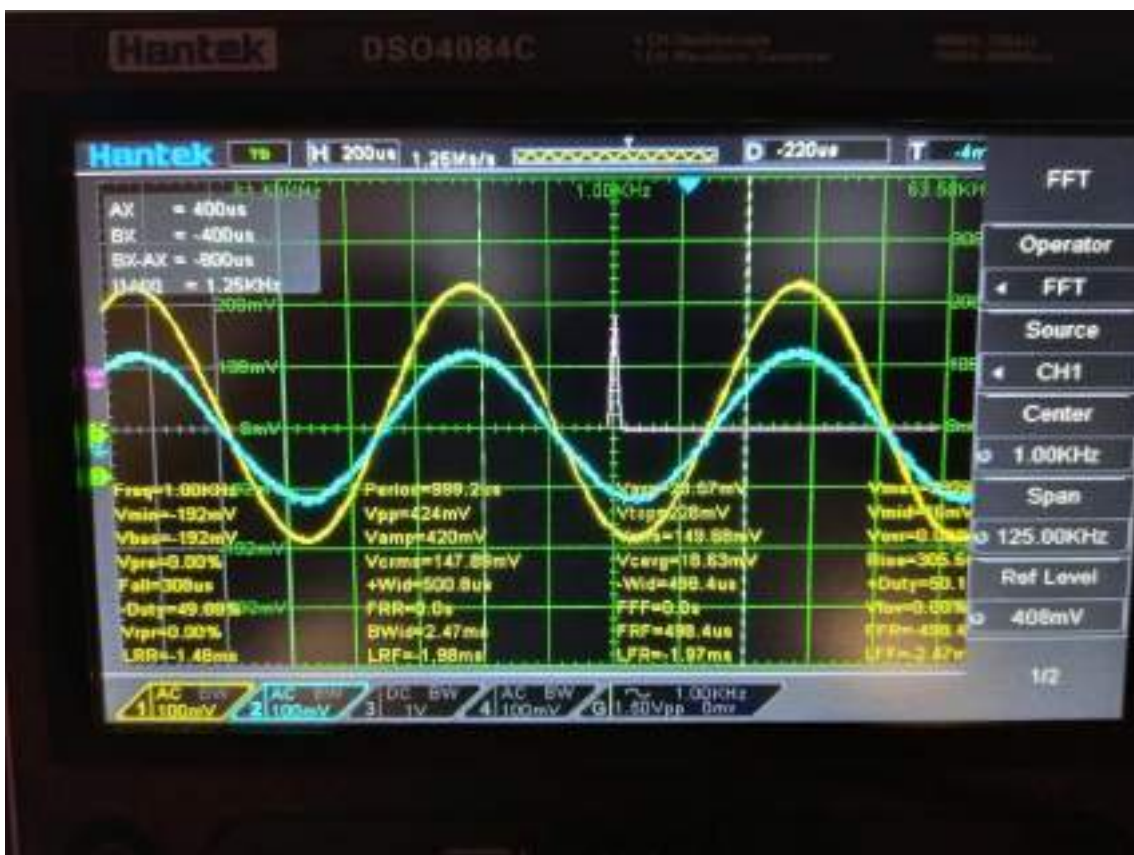


Figura 3.42: Señal de audio a la salida del filtro MAX7400

3.4.3. Diseño del pcb del medidor de impedancias

En la figura 3.43, se observa el diseño de pcb para la etapa de RF y audio del medidor de impedancias. Para el desarrollo de la primera etapa se tuvo en cuenta las reglas de diseño para señales de alta velocidad.

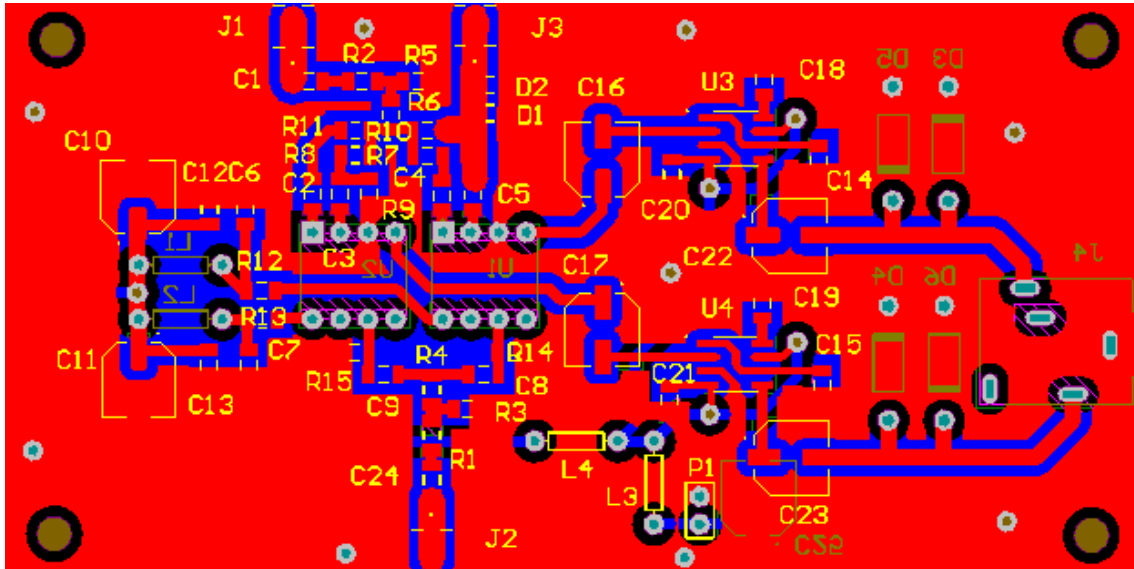


Figura 3.43: Diseño del pcb de la capa superior del medidor de impedancias

3.4.3.1. Reglas de diseño aplicadas

- Integridad de la señal: Indica la longitud máxima que debe tener una pista para que no haya onda reflejada. Cuando el tiempo de viaje de la señal a lo largo de la longitud de ida y vuelta es igual o mayor que el tiempo de crecimiento de la señal, el diseño puede sufrir reflexiones. Para evitar esto se usa la regla de un tercio del tiempo de crecimiento de la señal, la cual indica que si la ruta (pista) dura más de un tercio del tiempo de subida de la señal pueden ocurrir reflexiones. En este caso la ruta pasa a ser una línea de transmisión. Esto último, se puede observar en la siguiente ecuación:

$$L_R \geq \frac{T_R}{3} \times v_p = \frac{T_R}{3} \times \frac{c}{\sqrt{\epsilon_R}}$$

Donde,

v_p = velocidad de propagación de la señal en el medio de transmisión

$$c = \text{velocidad de la luz} \cong 3 \times 10^8 \frac{m}{s} = 300 \frac{mm}{ns}$$

ϵ_R = constante dieléctrica del material

T_R = tiempo de crecimiento de la señal en ns

L_R = longitud de la ruta en mm

Para este proyecto se utilizó una placa cuyo material es FR4. La constante dieléctrica de este material es de aproximadamente 4,6. Luego la velocidad de propagación de la señal en este medio es de:

$$v_p = \frac{c}{\sqrt{\epsilon_R}} = \frac{300 \frac{mm}{ns}}{\sqrt{4,6}} = 139,87 \frac{mm}{ns}$$

El tiempo de subida o crecimiento es el intervalo de tiempo que tarda la señal en pasar del 10% de su valor final al 90% de su valor final. Este tiempo se lo puede relacionar con el ancho de banda de la señal mediante la siguiente fórmula:

$$T_R = \frac{0,35}{AB}$$

En este proyecto, se diseñó el sistema de medición de impedancias para que trabaje entre 30 y 50 MHz de barrido. Por lo tanto, se debe considerar el tiempo de crecimiento más chico para el adecuado diseño de las pistas.

$$T_R = \frac{0,35}{50 \text{ MHz}} = 7 \text{ ns}$$

Luego, la longitud de pista en la cual aparecen los efectos de degradación de la señal es:

$$L_R = \frac{T_R}{3} \times v_p = 7 \text{ ns} \times \frac{139,87 \frac{mm}{ns}}{3} = 326,36 \text{ mm} = 32,63 \text{ cm}$$

En la figura 3.44, se observa una porción de la figura 3.41 donde viaja la señal de radiofrecuencia. La entrada designada "J1" corresponde a un conector sma. En este se conecta la señal de prueba que caracteriza la impedancia conectada en el conector sma designado como "J3". En la entrada designada como "J2" se conecta otro conector sma que corresponde a la señal del oscilador local. Luego de cada conector se diseñaron las pistas, cuyas longitudes no superan los 12 cm.

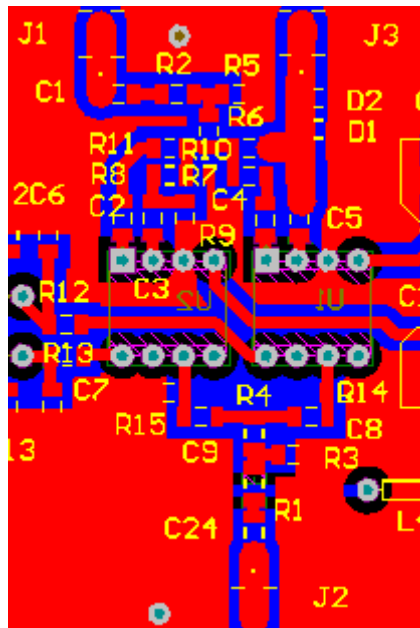


Figura 3.44: Rutas donde viajan las señales de RF

- Enrutamiento de impedancia controlada: Esta regla hace referencia a la adaptación de impedancias para evitar reflexiones en una línea de transmisión. En la figura 3.44, se observan los conectores sma designados como “J1” y “J2”. Las pistas que les siguen se diseñaron de forma tal que adapten los 50 Ω provenientes de la señal de cada filtro de RF terminado en 50 Ω . La pista que le sigue al conector “J3” se diseñó también con una impedancia característica de 50 Ω , por si desea conectar una antena para medir.

Esta adaptación se logró llevar a cabo mediante el uso de la herramienta “Saturn PCB Design”. En la figura 3.45, se puede ver el diseño de pista de 50 Ω deseado en base al ancho de la pista, separación entre la pista y los planos de masas adyacentes, la altura entre la pista y el plano de masa inferior y, por último, la constante dieléctrica del sustrato de la placa.



Figura 3.45: Diseño de pista de 50 Ω

- Interferencia: Debido a que parte de la energía en una señal de alta velocidad viaja a través del material que rodea la ruta, es inevitable que parte de esa energía se acople a rutas adyacentes. Para reducir la radiación electromagnética emitida por la señal que viaja por una pista se usa la topología coplanar. En este caso las líneas de campo electromagnética se cierran con el plano de masa adyacente a la pista. Si se tienen dos pistas diferentes en paralelo se debe usar la regla de diseño conocida como “3-W”. Esta regla hace énfasis en que, si se tienen dos pistas en paralelo, la distancia que tiene que haber entre ellas desde un centro de la pista hasta el centro de la otra, debe ser como mínimo de tres veces el ancho de la pista. De esta forma, se evita que la radiación electromagnética emitida por la señal de una ruta cree interferencia en la otra ruta.

En la figura 3.46, se puede observar el uso adecuado de esta regla para el diseño de dos caminos de RF en paralelo. Se diseñó la distancia entre el camino de RF del conector “J1” y el camino de RF del conector “J2” para que sea de 17 mm.

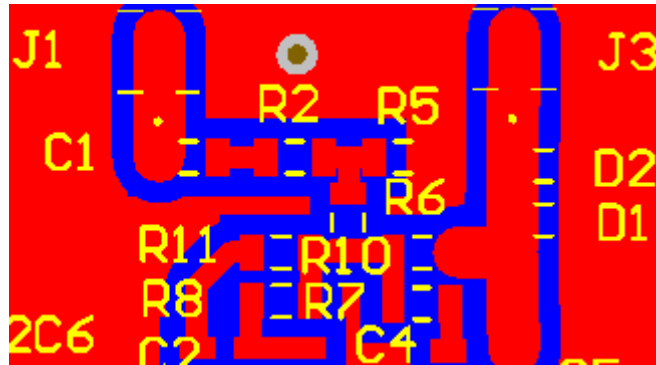


Figura 3.46: Distancia entre dos rutas adyacentes

- Espaciamiento entre vías: Los planos de masa del pcb deben estar todos unidos para evitar que actúen como “stubs” o segmentos de línea de transmisión. Dicho de otra manera, deben ser de baja impedancia en todas las frecuencias operativas del pcb, de lo contrario ya no se verán como tierra, sino que comienzan a verse reactivos a las señales que circulan en ellos. La forma de controlar que esto último no suceda es mediante vías que unen los dos planos de masa. La distancia entre vías debe ser de un octavo de longitud de onda o menor. Esto se debe a que cualquier pieza de cobre, como el plano de masa, comienza a comportarse como un “stub” cuando su longitud se acerca a un octavo de longitud de onda.

Para poder determinar la distancia de espaciamiento de las vías que unen los dos planos de masa se debió utilizar la siguiente fórmula:

$$\lambda = \frac{V_p}{f} = \frac{c}{f \times \sqrt{\epsilon_R}}$$

El medidor de impedancias barre de 30 a 50 MHz. Por lo tanto, se diseñó el espaciamiento con el mayor valor en frecuencia del instrumento. La constante dieléctrica relativa del material FR4 es de aproximadamente 4,6.

$$\lambda = \frac{300 \times 10^8 \text{ m/s}}{50 \times 10^6 \text{ Hz} \times \sqrt{4,6}} = 279,75 \text{ cm}$$

Luego, el espaciamiento entre vías debe ser menor o igual a:

$$\frac{\lambda}{8} = 34,96 \text{ cm}$$

Se decidió diseñar la distancia de espaciamiento de las vías en 30 cm, para tener en cuenta algún margen de error que pudiera haber en la constante dieléctrica del material fabricado.

3.4.3.2. Diseño del camino de VCC

En la figura 3.47, se observa el diseño de pcb de la capa superior del medidor de impedancias. En ella se encuentra el diseño de los caminos de la alimentación para los integrados.

Se usó una configuración del tipo “estrella” para rutear la tensión de VCC a cada integrado. La bobina L3 filtra cualquier ruido que haya en la señal de VCC. Adicionalmente, se agregó la bobina L4 para evitar que el camino de RF de la señal del oscilador local de la otra capa irradie el camino de VCC.

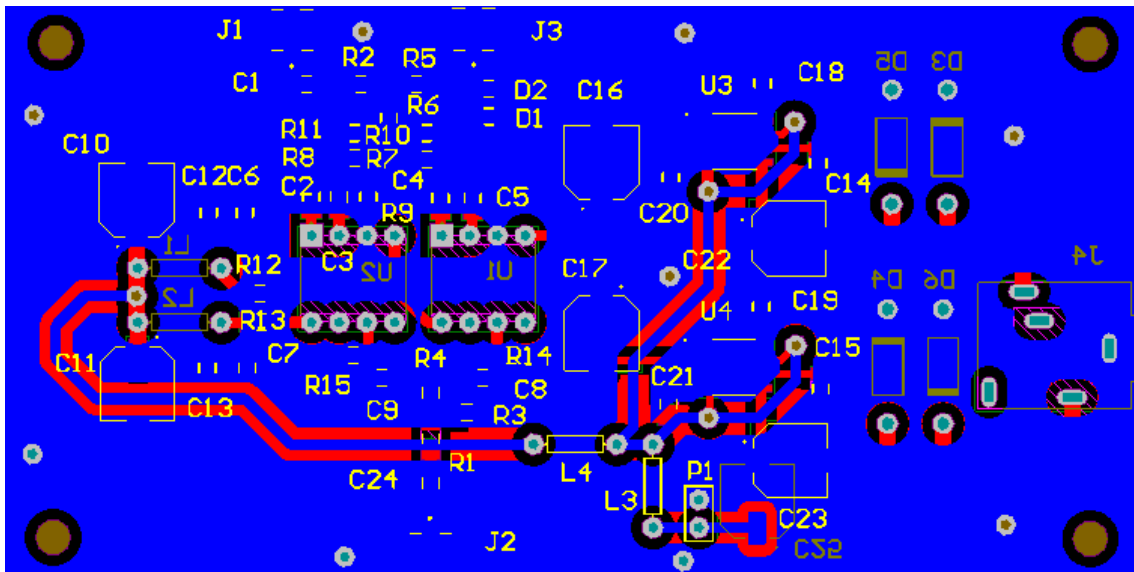


Figura 3.47: Diseño de pcb de la capa inferior del medidor de impedancias

3.4.3.3. Medidor de impedancias físico

En las figuras 3.46 y 3.47, se puede observar el circuito impreso del medidor de impedancias con los componentes ya soldados en cada capa de la placa.

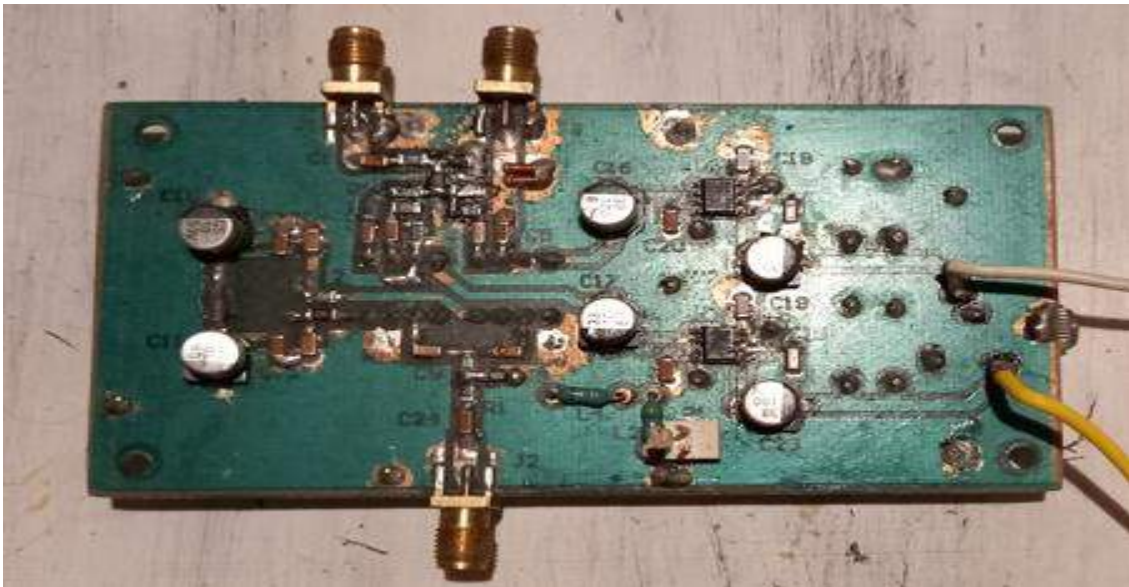


Figura 3.48: Medidor de impedancias capa superior

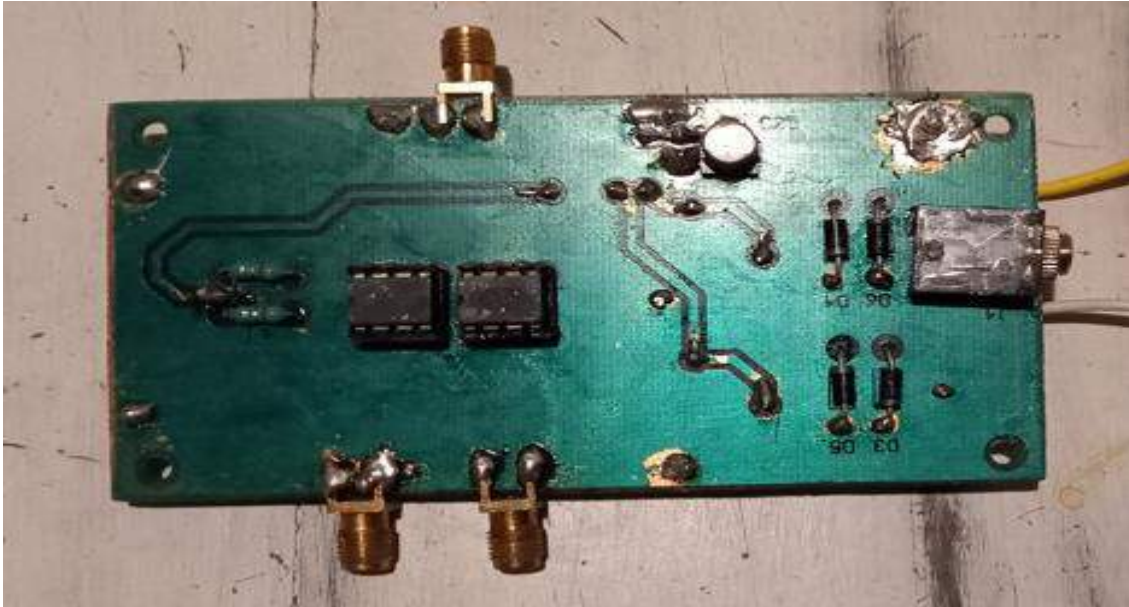


Figura 3.49: Medidor de impedancias capa inferior

3.5. Diseño del Software del sistema

3.5.1. Interfaz gráfica de usuario (GUI)

La aplicación cuenta con una sencilla interfaz gráfica que permite al usuario elegir la frecuencia inicial de barrido, el delta de frecuencia y la frecuencia final. Si se conoce el valor del componente a medir o algún valor cercano, se puede colocar el valor correspondiente para poder realizar la calibración del instrumento.

Para el desarrollo de la GUI se utilizó el software “QT Designer”. Se usó “PyQt5” para poder hacer uso de la biblioteca gráfica “QT” “en el lenguaje “Python”. Por último, se usó el IDE “Spyder” para el desarrollo funcional del código.

En la figura 3.48, se puede observar la interfaz gráfica desarrollada en “QT Designer”. La misma consta de diferentes widgets que se fueron arrastrando desde el panel de la izquierda (Widget Box) hacia la “Main Window”. Esta última, en programación orientada a objetos, corresponde a la clase principal que contiene todos los objetos dentro de ella. A continuación, se listan los widgets utilizados en el desarrollo de la interfaz:

- **QGroupBox:** Este widget proporciona un marco de cuadro de grupo con un título. Se usa como contenedor para agrupar distintos widgets dentro de él. En este diseño se utilizaron dos cuadros. Uno se lo nombró como “Parámetros de Barrido” y el otro como “Calibración de Impedancia”.
- **QTextEdit:** Esta clase proporciona un widget que se utiliza para editar y mostrar texto sin formato y enriquecido. En este diseño se utilizaron seis widgets de esta clase.
- **QLabel:** Este widget se usa como etiqueta. En este proyecto se usaron siete de este tipo. Estos se nombraron como “Frecuencia Inicial [MHz]”, “Delta de Frecuencia [KHz]”, “Frecuencia Final [MHz]”, “Valor R1”, “Valor R2”, “Valor C1”, “Seleccionar el tipo de componente a medir” y “Seleccionar el o los valores del componente”.

- QPushButton: Este widget proporciona un botón de comando. En este diseño se usó uno solo etiquetado como “Comenzar barrido”.
- QTabWidget: Esta clase proporciona una pila de widgets con pestaña. En este proyecto se usó una sola nombrada como “Impedancia Magnitud e Impedancia Fase”.
- QGraphicsView: Esta clase proporcionar un widget para poder visualizar un gráfico en dos dimensiones. En este proyecto se usaron dos de estos dentro de la “QTabWidget”.

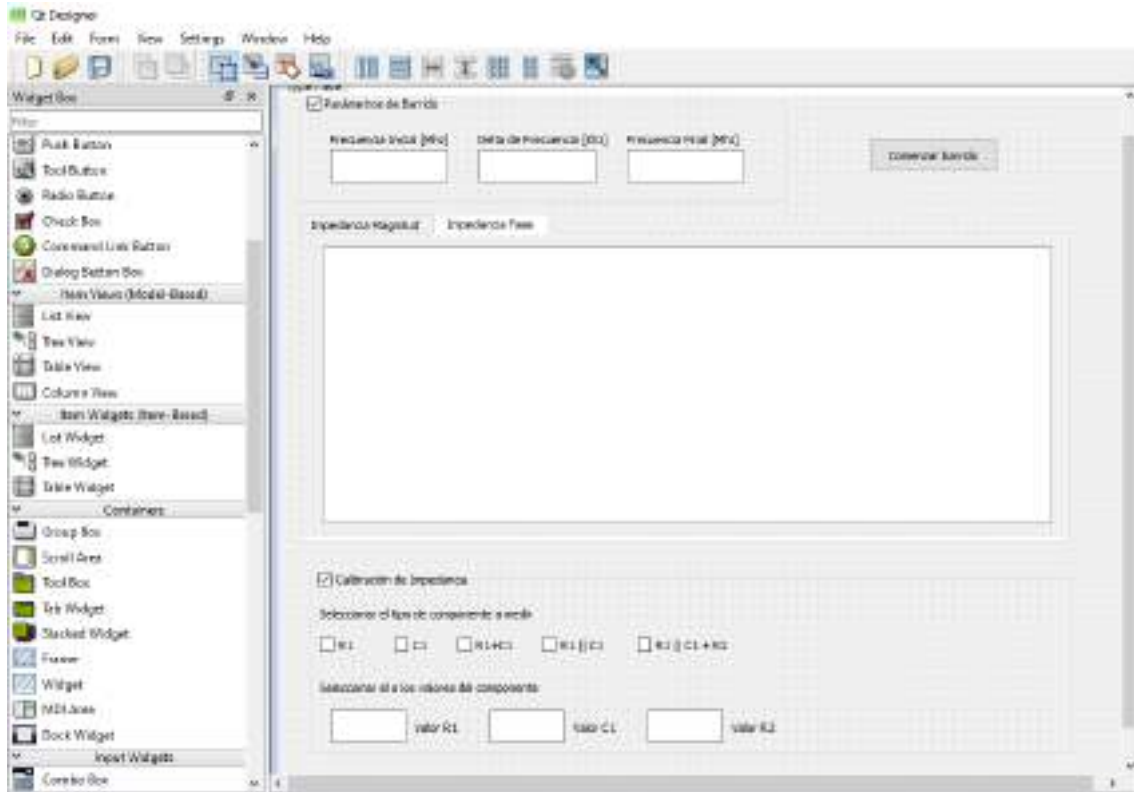


Figura 3.50: Diseño de la interfaz gráfica de usuario

Para poder trabajar en Python sobre esta interfaz, se debió realizar una conversión de formato del archivo de este diseño “.ui” a un código “.py”. La ventaja de hacer esto es que se puede realizar cambios posteriores en el diseño de la interfaz sin alterar el código funcional del programa. Esta conversión se logra mediante el comando:

```
pyuic5 -x ejemplo.ui -o ejemplo.py
```

Este comando se ejecuta en la terminal “Anaconda Prompt”. Para ello se debe situar en el directorio donde se encuentra el archivo de la GUI.

Para realizar el código funcional se usaron las siguientes librerías:

- PySerial
- Wave
- PyAudio
- Numpy
- Matplotlib
- Scipy

3.5.2. Diagrama de flujo del código funcional de la interfaz gráfica de usuario (GUI)

En la figura 3.49, se puede observar el diagrama de flujo del código funcional de la GUI.

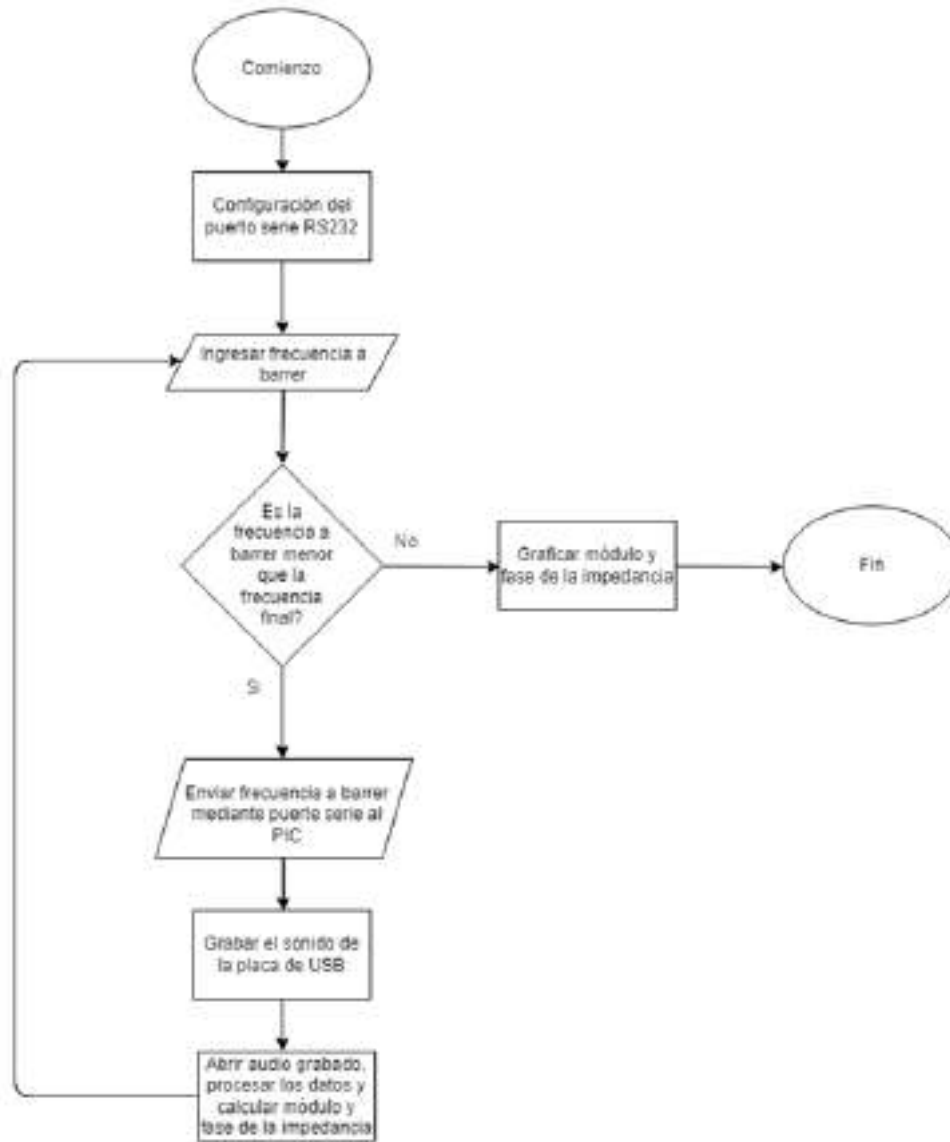


Figura 3.51: Diagrama funcional de la GUI

Capítulo 4

Validación del prototipo

En este capítulo se realizarán las validaciones del diseño del medidor de impedancias. Se demostrará el proceso de medición y se verificará, mediante la interfaz visual, el correcto funcionamiento del instrumento. Para ello, se realizará la medición con tres tipos diferentes de cargas:

- Carga resistiva de 47Ω
- Carga capacitiva de 100 pF
- Carga resistiva de 56Ω en serie con carga capacitiva de 82 pF

En la figura 4.1, se puede observar el banco de medición del sistema completo.



Figura 4.1: Banco de medición del sistema completo

4.1 Medición sobre carga de 47Ω

En la figura 4.2 se puede observar el procedimiento de medición para el cálculo de la resistencia de 47Ω . La interfaz es amigable y de fácil uso. Primero se colocan los parámetros de barrido deseado y, adicionalmente, si se conoce el valor de la impedancia a medir o un valor cercano, se puede tildar la opción "R1" y colocar el valor en el cuadro que corresponde. Por último, se hace clic en el botón "Comenzar barrido". Para este ejemplo, se decidió graficar la carga en todo el ancho de banda del instrumento (30 a 50 MHz) con un paso de 1000 kHz entre cada barrido.

Parámetros de Barrido

Frecuencia Inicial [MHz] Delta de Frecuencia [kHz] Frecuencia Final [MHz]

30 3000 50

Calibración de Impedancia

Seleccionar el tipo de componente a medir:

R1 C1 R1+C1 R1||C1 R1||C1+R2

Seleccionar el o los valores del componente:

Valor R1 Valor C1 Valor R2

Figura 4.2: Procedimiento de medición para una carga de 47 Ω

En las figuras 4.3 y 4.4, se puede observar el resultado de la medición en amplitud y fase respectivamente sobre la carga de 47 Ω .

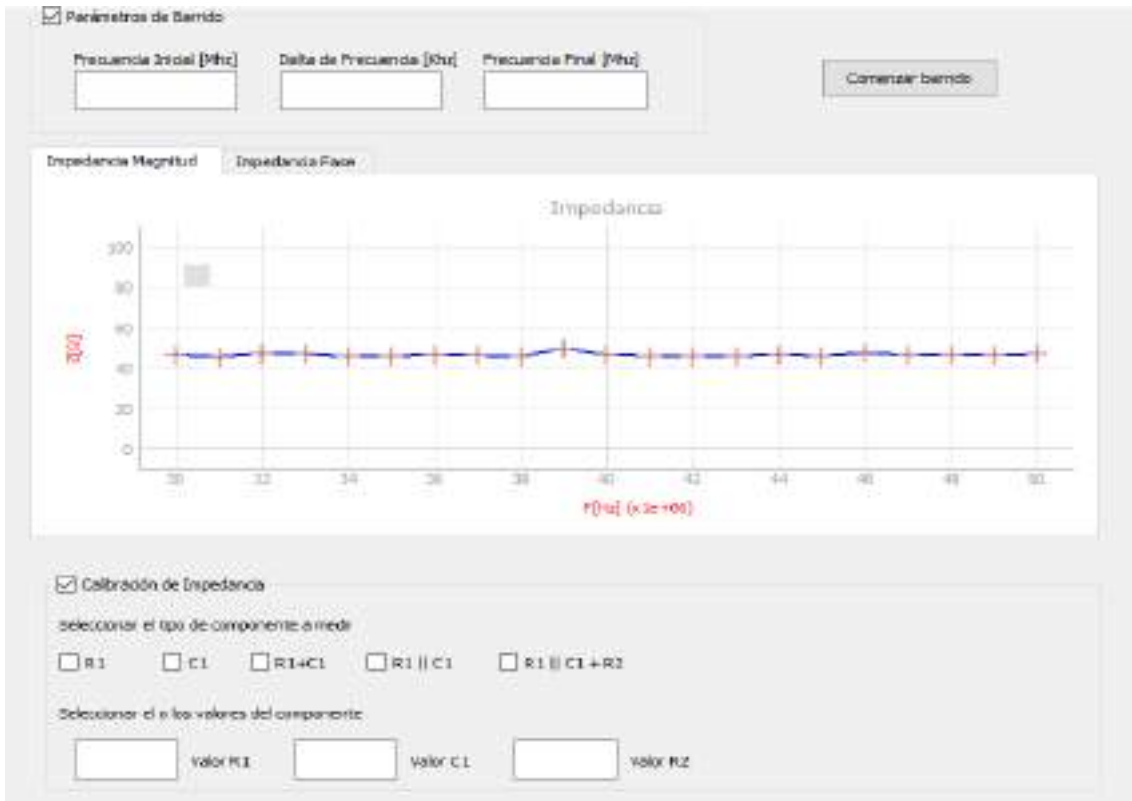


Figura 4.3: Magnitud de la impedancia de 47Ω versus frecuencia.

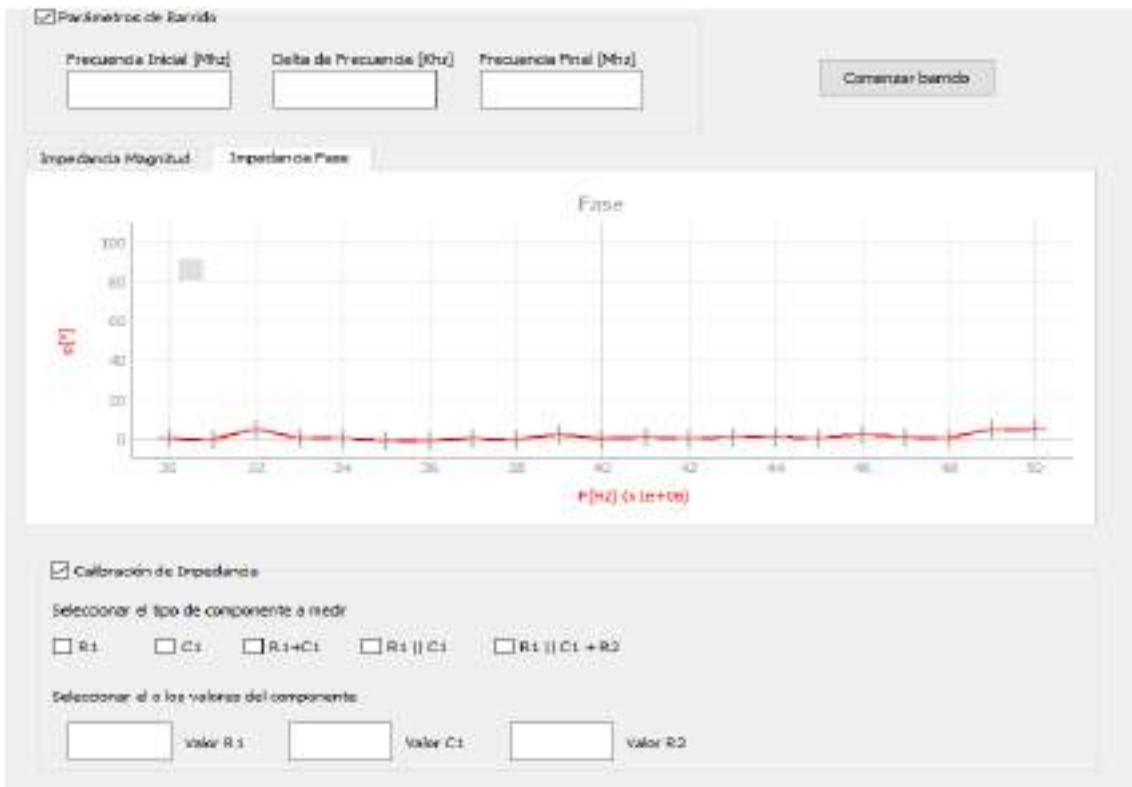


Figura 4.4: Fase de la impedancia de 47Ω versus frecuencia.

4.2 Medición sobre carga de 100 pF

En las figuras 4.5 y 4.6, se puede observar el resultado de la medición en amplitud y fase respectivamente sobre la carga de 100 pF.

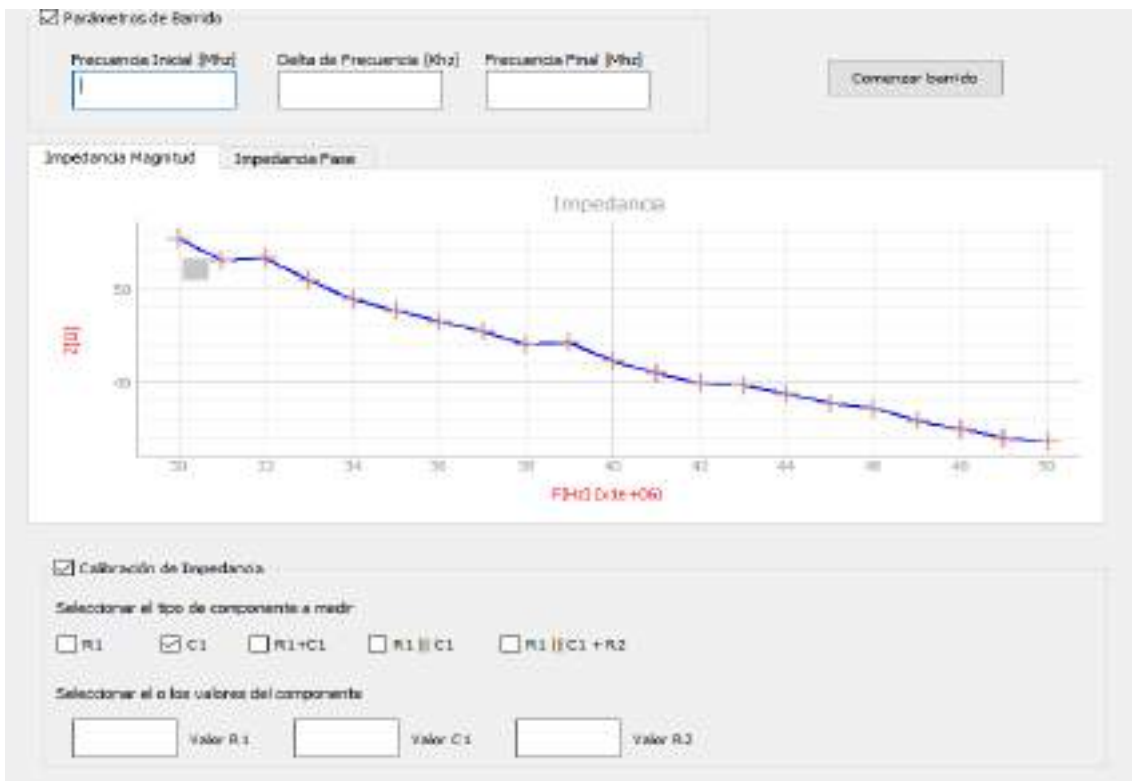


Figura 4.5: Magnitud de la impedancia del capacitor de 100 pF versus frecuencia.

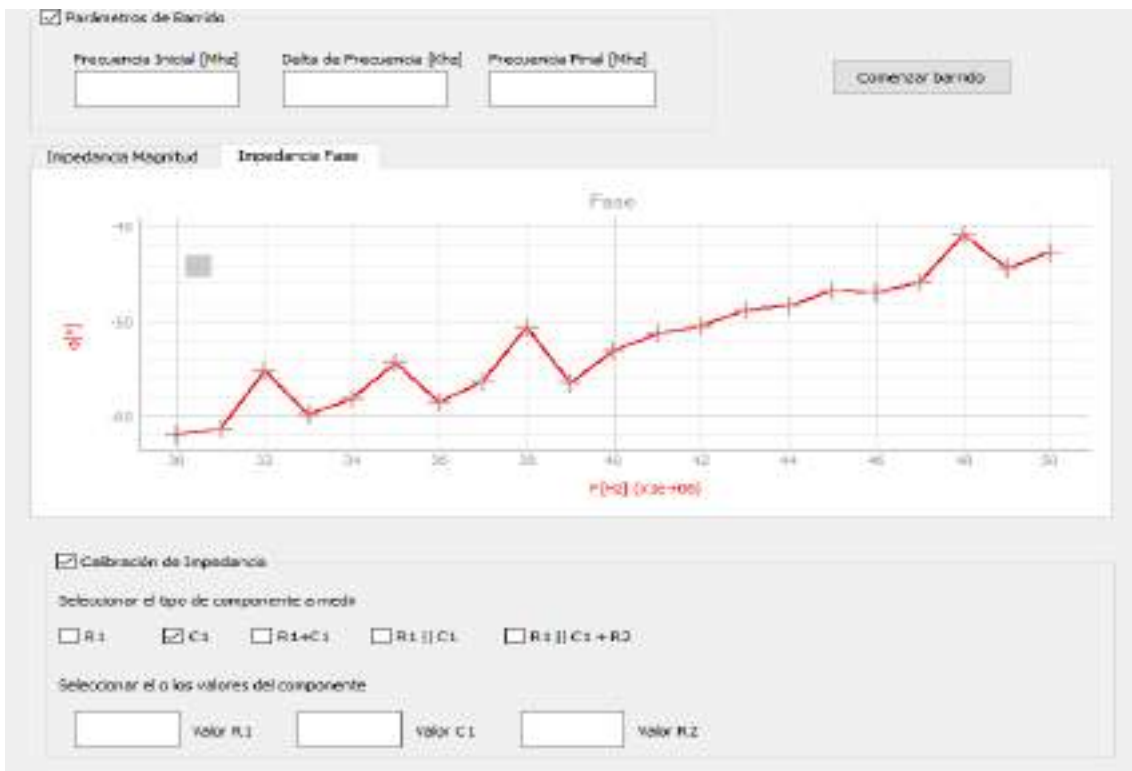


Figura 4.6: Fase de la impedancia del capacitor de 100 pF versus frecuencia.

4.3 Medición sobre carga de 56Ω en serie con una carga de 82 pF

En las figuras 4.7 y 4.8, se puede observar el resultado de la medición en amplitud y fase respectivamente sobre la carga de 56Ω en serie con una carga de 82 pF

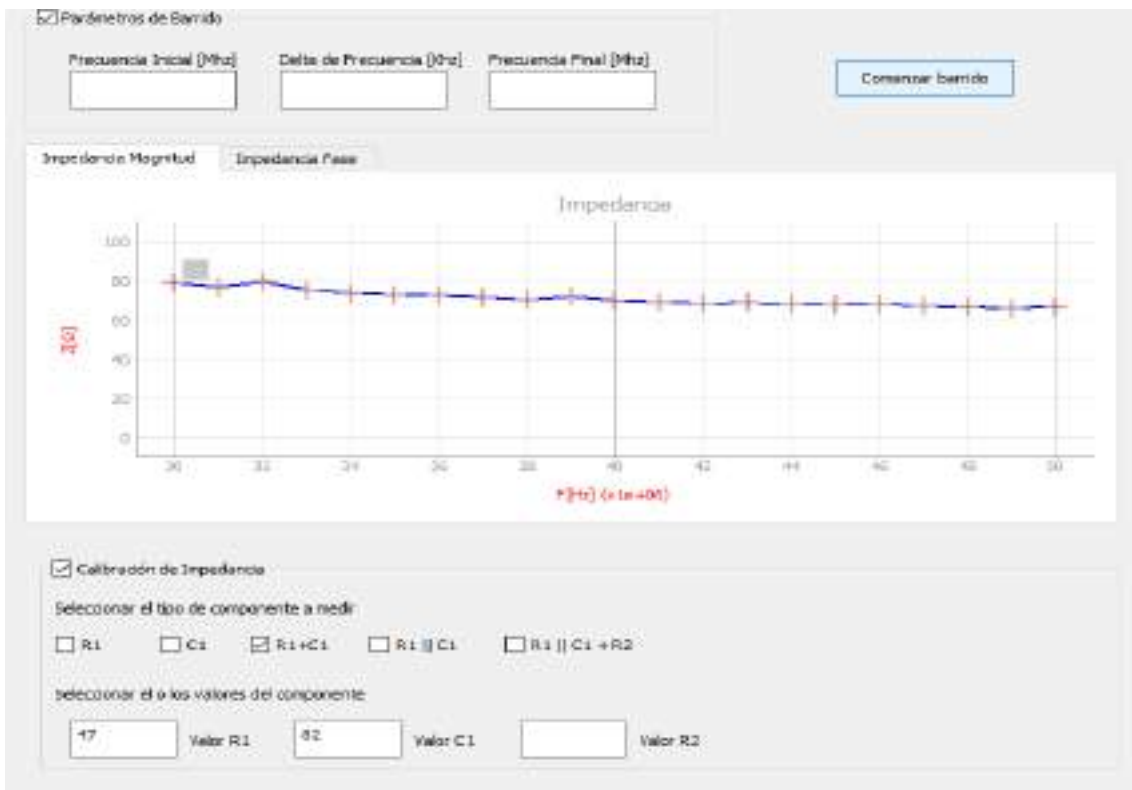


Figura 4.7: Magnitud de la carga de 56Ω en serie con una carga de 82 pF versus frecuencia

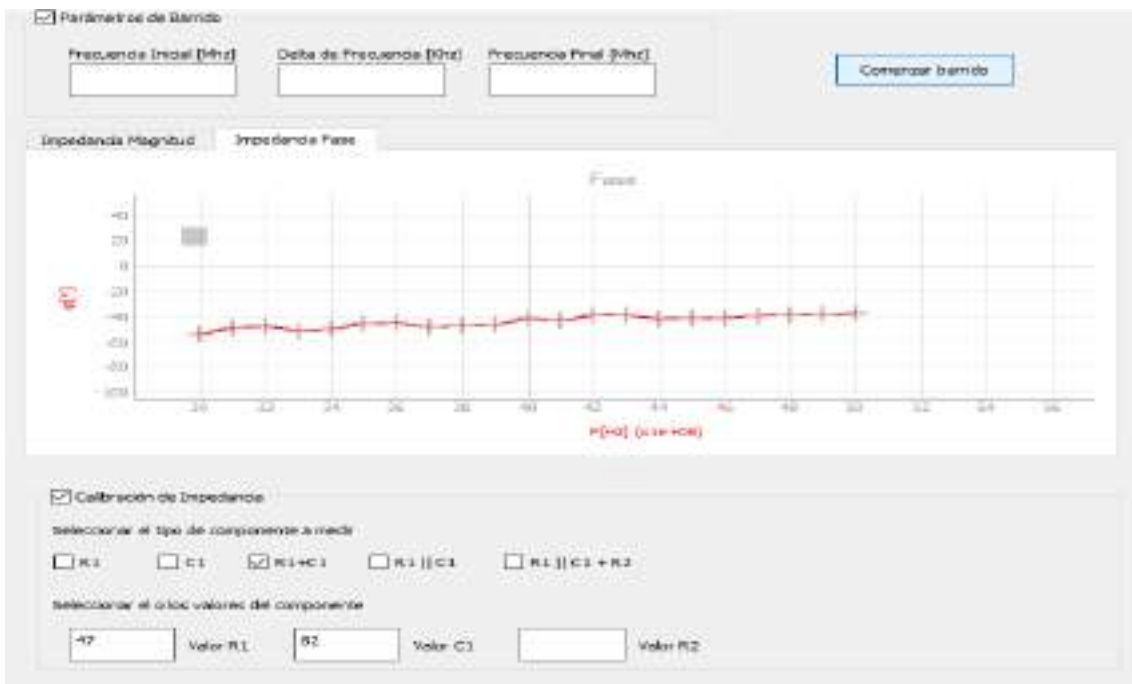


Figura 4.8: Fase de la carga de 56Ω en serie con una carga de 82 pF versus frecuencia

Capítulo 5

Conclusiones y trabajo futuro

5.1. Conclusiones

Se lograron los objetivos planteados. Se diseñó y construyó un medidor de impedancias en cuanto a su firmware, hardware y software.

Se pudo lograr construir el prototipo con un costo relativamente bajo.

Se lograron diseñar las placas de RF del circuito impreso mediante las reglas de diseño para señales de alta velocidad. En consecuencia, las pistas de RF no introdujeron fase considerable que pudiesen afectar la exactitud en la medición del medidor de impedancias.

En base a las simulaciones del diseño de los filtros de RF y las mediciones realizadas, se logró que los filtros tengan un comportamiento idéntico en cuanto a amplitud y fase, y, además, se cumplieron las condiciones de diseño.

Se logró sincronizar el barrido de frecuencia del instrumento mediante la interfaz gráfica de usuario (GUI), el microcontrolador (PIC), el generador de señal de reloj Si5351 y la adquisición de la señal de la placa de sonido USB.

Se logró realizar la calibración del prototipo mediante software. Debido a esto, se obtuvo una buena exactitud en la precisión de medición del instrumento.

Se hicieron modificaciones en el instrumento debido al contexto de la pandemia COVID – 19. En principio, el instrumento iba a cumplir la función de medidor de impedancias / analizador de antenas para el rango de frecuencia de FM (88 – 108 MHz). Las modificaciones se realizaron sobre el ancho de banda del instrumento. Se tuvo que reducir la frecuencia de corte de los filtros de RF para poder medir el funcionamiento del prototipo con el osciloscopio del autor, cuyo ancho de banda estaba limitado en 80 MHz.

5.2. Trabajo futuro

El Si5351 puede generar una señal de onda cuadrada cuya frecuencia puede variar de 8 kHz hasta 200 MHz. En este proyecto debido al contexto de pandemia no se pudo aprovechar el rango completo del instrumento. A continuación, se listan una serie de consideraciones para realizar mejoras sobre el medidor de impedancias:

- Diseñar un filtro de mayor orden con caída más abrupta. Con esto se lograría tener una mayor banda de paso para el barrido de frecuencia del instrumento.
- Diseñar distintos filtros con un selector previo. De esta forma el usuario pueda hacer uso de todo el rango de frecuencia del Si5351, basta con seleccionar el filtro adecuado para el rango deseado.

- Diseñar el instrumento con un microcontrolador que soporte el protocolo de comunicación I²S. Este protocolo sirve para interconectar circuitos de audio digital. Con este nuevo microcontrolador se podría reemplazar la placa de sonido USB de este proyecto por un códec (circuito integrado) de audio.
- Diseñar la plaqueta del medidor de impedancias con un pcb de cuatro capas. Con esto se lograrían las siguientes mejoras:
 - I. Proporcionar un plano de potencia completo. Se reduce la inductancia de las conexiones entre la fuente de alimentación y sus cargas.
 - II. Reducir la separación entre el plano de potencia y el plano de tierra, con esto se logra minimizar el área del bucle y, por lo tanto, los campos magnéticos generados asociados a las corrientes de potencia y de retorno.
 - III. Reducir la separación entre la capa de señal y el plano de retorno. Nuevamente, se reduce el área del bucle para cada ruta de señal y sus emisiones de campo magnético.
- Diseñar el instrumento para que trabaje también como analizador de antenas. Para ello, se deberán hacer ajustes sobre la interfaz visual para poder mostrar gráficas adicionales como, por ejemplo:
 - I. Diagrama de Smith
 - II. ROE

Bibliografía

- [1] *NE602/NE612 info.* (s/f). Qsl.net. Recuperado el 6 de junio de 2022, de <https://www.qsl.net/kf4trd/ne602.html>
- [2] (S/f). Modernhamguy.com. Recuperado el 6 de junio de 2022, de <https://modernhamguy.com/active-mixers-in-rf-design/>
- [3] electronics notes. (s/f). What is a Gilbert Cell Mixer » Electronics Notes. Electronics-notes.com. Recuperado el 6 de junio de 2022, de <https://www.electronics-notes.com/articles/radio/rf-mixer/gilbert-cell-rf-mixer.php>
- [4] electronics notes. (s/f-a). Double balanced mixer: Theory circuit operation » electronics notes. Electronics-notes.com. Recuperado el 6 de junio de 2022, de <https://www.electronics-notes.com/articles/radio/rf-mixer/double-balanced-mixer.php>
- [5] Feranec, R. [RobertFeranec]. (2015, enero 19). High speed PCB design rules (lesson 4 of advanced PCB layout course). Youtube. <https://www.youtube.com/watch?v=BIHLMQ2HO1w&list=RDCMUCJQkHVpk3A8bgDmPIJJOA>
- [6] Feranec, R. [RobertFeranec]. (2019, abril 2). Simple way to Calculate Impedance, Current, Crosstalk. Youtube. https://www.youtube.com/watch?v=O4T9y_aItA&list=RDCMUCJQkHVpk3A8bgDmPIJJOA&index=10
- [7] Feranec, R. [RobertFeranec]. (2014, julio 8). Length Matching in Altium. Youtube. <https://www.youtube.com/watch?v=bzRDfcJlIlgA&list=RDCMUCJQkHVpk3A8bgDmPIJJOA&index=3>
- [8] Fitzpatrick, M. (2019, agosto 20). Embedding PyQtGraph (or any other custom PyQt5 widgets) from Qt Designer. Python GUIs. <https://www.learnpyqt.com/courses/qt-creator/embed-pyqtgraph-custom-widgets-qt-app/>
- [9] Fitzpatrick, M. (2019b, octubre 12). Plotting in PyQt5 — Using PyQtGraph to create interactive plots in your apps. Python GUIs. <https://www.learnpyqt.com/courses/graphics-plotting/plotting-pyqtgraph/>
- [10] Profe, H. (2018, agosto 30). Primeros pasos en PyQt 5 y Qt Designer: Programas gráficos con Python. Medium. <https://medium.com/@hektorprofe/primeros-pasos-en-pyqt-5-y-qt-designer-programas-gr%C3%A1ficos-con-python-6161fba46060>
- [11] PyQtGraph: gráficas tiempo real con Python. (2019, abril 3). Lab. Gluón. <https://www.laboratoriogluon.com/pyqtgraph-graficas-tiempo-real-con-python/>

- [12] (S/f-c). Qsl.net. Recuperado el 6 de junio de 2022, de <https://www.qsl.net/w2aew/youtube/longtailedpair.pdf>
- [13] PCB layout & high speed design using Altium Designer. (s/f). Altium.Com. Recuperado el 6 de junio de 2022, de <https://www.altium.com/documentation/altium-designer/high-speed-design-in-altium-designer-ad>
- [14] (S/f-d). Qsl.net. Recuperado el 6 de junio de 2022, de <https://www.qsl.net/w2aew/youtube/GilbertCell.pdf>
- [15] w2aew [w2aew]. (2015, diciembre 21). #223: Basics of the Gilbert Cell | analog multiplier | mixer | modulator. Youtube. <https://www.youtube.com/watch?v=7nmmb0pqTU0>
- [16] Analog Devices, Inc. [analogdevicesinc]. (2014, enero 27). High speed and RF design considerations. Youtube. <https://www.youtube.com/watch?v=6jrVZu7eqiw>
- [17] Feranec, R. [RobertFeranec]. (2019b, mayo 1). Understanding PCB Layout Essential - Change the way how you look at the tracks. Youtube. <https://www.youtube.com/watch?v=EUAJbNnqhLs>
- [18] Carr, J. J. (s/f). Lb3hc.net. Recuperado el 6 de junio de 2022, de <http://www.lb3hc.net/wp-content/uploads/2019/12/UsingTheNe602.pdf>
- [19] (S/f-e). Blogspot.com. Recuperado el 6 de junio de 2022, de <http://audio-cfp.blogspot.com/2014/04/nivel-de-senal-microfono-y-linea.html>
- [20] Pic18f4550 i2c. (s/f). Electronicwings.Com. Recuperado el 6 de junio de 2022, de <https://www.electronicwings.com/pic/pic18f4550-i2c>
- [21] Si5351A. (s/f). Rfzero.Net. Recuperado el 6 de junio de 2022, de <https://www.rfzero.net/tutorials/si5351a/>
- [22] Pic18f4550 usart. (s/f). Electronicwings.Com. Recuperado el 6 de junio de 2022, de <https://www.electronicwings.com/pic/pic18f4550-usart>
- [23] (S/f-f). Qwe.wiki. Recuperado el 6 de junio de 2022, de https://es.qwe.wiki/wiki/Line_level
- [24] Magdy, K. (2018, julio 12). Interrupts in PIC microcontrollers. DeepBlue. <https://deepbluembedded.com/interrupts-in-pic-microcontrollers/>
- [25] T attenuator calculator. (s/f). M0ukd.Com. Recuperado el 6 de junio de 2022, de <https://m0ukd.com/calculators/t-attenuator-calculator/>
- [26] RF tools. (s/f). Rf-Tools.Com. Recuperado el 6 de junio de 2022, de <https://rf-tools.com/lc-filter/>

- [27] Via spacing on high-performance PCBs. (s/f). Pcbway.com. Recuperado el 6 de junio de 2022, de https://www.pcbway.com/blog/Engineering_Technical/Via_spacing_on_high_performance_PCBs.html
- [28] (S/f-g). W5big.com. Recuperado el 6 de junio de 2022, de https://w5big.com/QST_Article.pdf
- [29] (S/f-h). Qsl.net. Recuperado el 6 de junio de 2022, de <https://www.qsl.net/k5bcq/Kits/Kits.html>
- [30] Manually generating an Si5351 register map for 10-MSOP and 20-QFN devices. (2021). Skyworksinc.com. <https://www.skyworksinc.com/-/media/Skyworks/SL/documents/public/application-notes/AN619.pdf>
- [31] (S/f-i). Nxp.com. Recuperado el 6 de junio de 2022, de <https://www.nxp.com/docs/en/data-sheet/SA612A.pdf>
- [32] 阻抗测量仪器-阻抗测量手册 impedance measurement handbook (2). (s/f). 知乎专栏 . Recuperado el 7 de junio de 2022, de <https://zhuanlan.zhihu.com/p/191621735>
- [33] Unidos, H. E. (2021, abril 24). What is an Impedance Meter? Hioki USA. <https://hiokiusa.com/es/learning/test-tools/impedance-meters/>
- [34] Visual paradigm online - suite of powerful tools. (s/f). Visual-Paradigm.Com. Recuperado el 7 de junio de 2022, de <https://online.visual-paradigm.com/>