



UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE INGENIERÍA ELÉCTRICA

DISEÑO, IMPLEMENTACIÓN Y PRUEBAS DE UN SISTEMA DE OSCILADOR LOCAL Y SU SISTEMA DE CONTROL

MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL ELÉCTRICO

ÁLVARO ISAAC MORALES CABRERA

PROFESOR GUÍA:
WALTER MAX-MOERBECK ASTUDILLO

MIEMBROS DE LA COMISIÓN:
MAXIMILIANO PRIETO CARVAJAL
FRANCISCO RIVERA SERRANO

Este trabajo ha sido parcialmente financiado por la Agencia Nacional
de Investigación y Desarrollo (ANID) a través de
su fondo Quimal ASTRO21-0010

SANTIAGO DE CHILE

2024

RESUMEN DE LA MEMORIA PARA OPTAR
AL TÍTULO DE INGENIERO CIVIL ELÉCTRICO
POR: ÁLVARO ISAAC MORALES CABRERA
FECHA: 2024
PROF. GUÍA: WALTER MAX-MOERBECK

DISEÑO, IMPLEMENTACIÓN Y PRUEBAS DE UN SISTEMA DE OSCILADOR LOCAL Y SU SISTEMA DE CONTROL

En radioastronomía, disciplina en la que se estudian fenómenos y objetos celestes mediante la recepción de radiación electromagnética en la banda de ondas de radio, el oscilador local es un componente fundamental que se utiliza dentro del sistema de recepción del radiotelescopio. Las ondas de radio captadas pueden tener frecuencias muy altas, muchas de ellas encontrándose en el rango de microondas. Tales frecuencias no son adecuadas para el procesamiento directo, y por ende se requiere de una conversión hacia un rango más bajo. Es por este motivo que se requiere de un oscilador, el cual genera una señal a frecuencia estable, conocida y que es mezclada con la onda incidente. Este proceso, denominado conversión de frecuencia, entrega una señal de frecuencia menor y adecuada para el posterior procesamiento y análisis.

El oscilador local forma parte del amplio conjunto de instrumentación que se puede encontrar dentro del sistema de recepción de un radiotelescopio. En complemento de otros componentes de radiofrecuencias, es posible la recepción y procesamiento de un amplio rango de señales de la banda de observación. Sobre lo mismo, es de particular interés el radiotelescopio denominado Telescopio de Ondas Milimétricas del Sur, *Southern Millimeter Wave Telescope SMWT*, por sus siglas en inglés, ubicado en Cerro Calán y que forma parte de la Universidad de Chile. La importancia yace en que se encuentra en marcha el plan de actualización de la instrumentación del radiotelescopio. Con esto, se logrará su operación en la banda de 67–116 GHz y, además, en lugar de utilizar nitrógeno líquido, se integrará un ciclo cerrado para la etapa de criogenia.

En consecuencia, el presente trabajo de título se enmarca en el contexto de la actualización de la instrumentación del radiotelescopio *SMWT*, en específico de su oscilador local. Desde su puesta en marcha en 1982 hasta el día de hoy, su oscilador se ha visto modificado en función de los requerimientos y especificaciones de la investigación que debe ser llevada a cabo. Para este caso, se diseñó e implementó el nuevo oscilador local, en conjunto con su sistema de control, mediante el cual se puede medir la potencia de salida, sintonizar valores específicos de frecuencia, y mantener el monitoreo constante de los dispositivos.

Se logró implementar el nuevo oscilador local en conjunto de su sistema de control, y con ello se comprobó su funcionamiento. Para esto, se diseñó una interfaz de línea de comandos que establece la comunicación con los componentes del oscilador, y permite sintonizar una frecuencia para el rango de 36 – 60 GHz. Además, se incorporó el sistema de medición de potencia, el cual entrega la potencia de salida del oscilador local, con una precisión de cuatro decimales. De esta forma, es posible obtener y guardar información sobre el estado de los componentes del oscilador en términos de voltaje, potencia, temperatura y frecuencia.

*La paz interior es la mejor armadura
para enfrentar los desafíos de la vida.*

Agradecimientos

Para toda mi familia. A mis padres Pamela y Rodolfo. Gracias por enseñarme todo lo que sé y por siempre velar que nada falte. Todos mis logros son también parte de ustedes, y ambos pueden sentirse merecedores de este proceso. Gracias a mi hermana Pauli, siempre has sido una tremenda compañera y amiga. Que nadie detenga tus planes, vas a ser y eres la mejor. No dudes en contar conmigo, hasta en lo más mínimo que necesites.

Gracias a mi abuelita Lucía, que siempre has estado presente en cada etapa de nuestras vidas, y por recibirnos con los brazos abiertos cada domingo en tu hogar con cientos de anécdotas. Gracias a mi tía Fabiola, por tus comidas ricas, por tu alegría y forma de ser. Estaré eternamente agradecido por tus consejos y enseñanzas. Y para mi abuelita Rosa. Eres una persona inigualable que llevaré en mi corazón para toda mi vida. Eres la pionera en formar parte de esto.

Para Valeria, Godoy, Nacha, Amanda y Reed, mis amigos/as de la vida. Han sido 10 o 20 años de un sin fin de experiencias. Gracias por ser parte y estar presentes en lo más mínimo. Sin duda agradezco que hayan estado cada vez que lo he necesitado. Mil gracias por ayudar a deshacerme de mi armadura.

A Gabriel, mi Profe de Saxo. Gracias por enseñar y mostrarme lo bonito que puede llegar a ser tocar el Saxofón, gracias por siempre incluirme en todas las actividades. Sin duda pude encontrar algo más que un simple pasatiempo. Que vengan más clases y dúos!

A los Boludos. Fue poco el tiempo que coincidimos, pero siento que los conozco de toda la vida. Tengo los mejores recuerdos de nuestra pasada por París. Que se repitan todos esos viajes, caminatas, comidas, charlas y salidas que hicimos. Espero volver a encontrarnos nuevamente por Madrid!

Gracias Salma por todo el apañe y experiencias que hemos tenido durante este último tiempo, estoy muy contento de habernos encontrado. Orgulloso de ti y todos tus logros. Eres y serás la mejor enfermera.

Al equipo de Calán y sobre todo al Laboratorio, quienes son un tremendo grupo humano y en todo momento me recibieron con las puertas abiertas. En especial a Max. Gracias por todo el tiempo y paciencia que tuviste siempre para ayudar y explicarme cada vez que lo necesité. Tenemos que sacar algún tema de jazz pronto! Por último, agradezco también a mi comisión, sobre todo a Walter por confiarme esta parte del proyecto.

Gracias a todos/as por aportar con su granito de arena.

Tabla de Contenido

1. Introducción	1
1.1. Motivación y Formulación del Problema	1
1.2. Objetivos del Trabajo de Título	2
1.2.1. Objetivo General	2
1.2.2. Objetivos Específicos	2
2. Marco Teórico y Estado del Arte	4
2.1. Marco Teórico	4
2.1.1. Rol de la Astronomía	4
2.1.1.1. Radiación Electromagnética	4
2.1.1.2. Detección de Señales en la Banda de Microondas	5
2.1.1.3. Importancia de la Banda 67-116 GHz	6
2.1.2. Receptor Heterodino	7
2.1.2.1. Ancho de Banda de RF e IF	8
2.1.2.2. Sobre el Ruido	8
2.1.2.3. Amplificador de Bajo Ruido	9
2.1.2.4. Mezclador	9
2.1.2.5. Oscilador Local	10
2.1.2.6. Multiplicador de Frecuencia	10
2.1.2.7. Atenuador	10
2.1.2.8. Acoplador Direccional	11
2.1.2.9. Detector de Potencia de Banda Ancha	11
2.1.3. Una Breve Historia del Radiotelescopio Mini	11
2.1.3.1. Oscilador <i>Gunn</i> , el Primer Gran Cambio	12
2.1.3.2. El Segundo Cambio: Diodos <i>Schottky</i>	12
2.1.4. Especificaciones Técnicas del SMWT	12
2.1.4.1. Sistema Óptico	13
2.1.4.2. Sistema Receptor	14
2.1.4.3. Sistema de Control	18
2.2. Estado del Arte	20
2.2.1. ALMA - Banda 2 y 3	20
2.2.2. Otros Radiotelescopios	20
2.2.3. Receptores Heterodinios Modernos	22
2.2.3.1. Fuentes de Oscilador Local	22
2.2.4. Sintetizadores	22
3. Diseño e Implementación	24

3.1.	Metodología	24
3.2.	Diseño del Oscilador Local y su Sistema de Control	27
3.2.1.	Componentes de Radiofrecuencias	28
3.2.1.1.	Sintetizador	28
3.2.1.2.	Multiplicador	39
3.2.1.3.	Atenuador Variable	42
3.2.1.4.	Acoplador Direccional	43
3.2.1.5.	Detector de Potencia de Banda Ancha UD	43
3.2.2.	Componentes del Sistema de Control	45
3.2.2.1.	Raspberry Pi	45
3.2.2.2.	Placa de Control Raspberry Pi - Oscilador Local	45
3.2.2.3.	Módulo ADS1115	48
3.2.2.4.	Convertor de Nivel	50
3.2.2.5.	Relé	51
3.3.	Implementación del Oscilador Local y su Sistema de Control	52
3.3.1.	Funcionamiento del Sistema de Control	52
3.3.1.1.	Componente de <i>Software</i>	52
3.3.1.2.	Componente de <i>Hardware</i>	58
3.3.1.3.	Sistema de Medición de Potencia	63
3.3.2.	Configuración de Mediciones del Oscilador Local	66
3.3.2.1.	Banda 36-50 GHz.	67
3.3.2.2.	Banda 50-60 GHz.	69
4.	Resultados y Discusión	72
4.1.	Sistema de Control	72
4.1.1.	Medición de Potencia	72
4.1.2.	Sintonización de Frecuencia	74
4.2.	Oscilador Local	76
4.2.1.	Sensibilidad del Detector UD	77
4.2.1.1.	Calibración	77
4.2.2.	Mediciones de Potencia del Sistema para 36-50 GHz	80
4.2.3.	Mediciones de Potencia del Sistema para 50-60 GHz	82
4.2.4.	Sensibilidad del Sistema de Medición de Potencia	84
5.	Conclusiones	86
5.1.	Trabajo Futuro	87
	Bibliografía	88
	Anexos	90
A.	Configuración Inicial del Raspberry Pi del Sistema de Control del Oscilador Local para el <i>SMWT</i>	90
A.1.	Instalación Raspberry Pi OS	90
A.2.	Configuración de Conexión Wi-Fi	91
A.3.	Configuración SSH	92
A.4.	Encontrar la IP del Raspberry Pi	92
A.5.	Configuración de IP estática	93
B.	Documentación del Oscilador Local	95

B.1.	Google Drive con Documentos	95
B.2.	Estructura del Repositorio Instmini	95
C.	Códigos en Python y C	97
D.	Control de Frecuencia en Oscilador Local Actual	116
E.	Placa de Control RPI-LO, Versión 2	118

Índice de Tablas

2.1.	Líneas de emisión para la banda 67 – 116 GHz de interés para la investigación con el <i>SMWT</i>	7
2.2.	Principales especificaciones técnicas del radiotelescopio.	13
2.3.	Otros instrumentos de interés que operan en bandas similares a la de 67 – 116 GHz.	21
2.4.	Principales características de Sintetizadores de <i>Micro Lambda Wireless INC</i>	23
3.1.	Equipos de radiofrecuencias y sistema de control utilizados.	24
3.2.	Equipos disponibles en el Laboratorio utilizados durante las pruebas.	24
3.3.	Información sobre pines de control para la programación del sintetizador.	29
3.4.	Especificaciones de RF para el sintetizador [30].	30
3.5.	Especificaciones eléctricas para el sintetizador [30].	30
3.6.	Especificaciones mecánicas para el sintetizador [30].	31
3.7.	Condiciones de operación del sintetizador.	31
3.8.	Significado de cada bit de estado del sintetizador.	31
3.9.	Información del sintetizador que será monitoreada.	36
3.10.	Especificaciones del multiplicador X4.	39
3.11.	Especificaciones del atenuador variable.	42
3.12.	Especificaciones del acoplador direccional.	43
3.13.	Especificaciones del detector de potencia de banda ancha UD.	44
3.14.	Tabla de calibración del detector de potencia de banda ancha UD.	44
3.15.	Especificaciones del módulo <i>ADS1115</i>	49
3.16.	Especificaciones del módulo conversor de nivel.	51
3.17.	Especificaciones del módulo relé de 1 canal.	52
3.18.	Comandos disponibles para ser enviados en <i>rpiserver</i>	54
4.1.	Resultados estadísticos para las mediciones de potencia con el Modelo 1.	78
4.2.	Resultados estadísticos para las mediciones de potencia para la banda de 33.5 – 50 GHz, con el Modelo 2.	82
4.3.	Resultados estadísticos para las mediciones de potencia para la banda de 50 – 64 GHz, con el Modelo 2.	83
D.1.	Definiciones de variables de interés para el control de frecuencia.	117

Índice de Ilustraciones

2.1.	Espectro Electromagnético [3].	5
2.2.	Transmisión de la atmósfera terrestre para la radiación electromagnética [4].	6
2.3.	Diagrama de un receptor heterodino [6].	8
2.4.	Diagrama comúnmente utilizado que representa un acoplador direccional [6].	11
2.5.	Diagrama general del radiotelescopio Mini [14].	13
2.6.	Esquema de una antena Cassegrain. En (a) se muestra la vista lateral del trazo del rayo. En (b) se muestra la vista frontal [16].	14
2.7.	Diagrama del <i>front-end</i> y LNA.	15
2.8.	Diagrama actual de la placa del oscilador local [17].	16
2.9.	Espectro para la molécula de CO dentro de Orión KL [18].	18
2.10.	Interfaz de comunicación para la operación del radiotelescopio.	19
2.11.	Instalación y montaje del radiotelescopio Mini [12].	19
2.12.	Antenas de ALMA sobre el Llano de Chajnantor [19].	20
2.13.	Ejemplos de otros radiotelescopios que operan en bandas similares a la de 67–116 GHz.	21
3.1.	Principales equipos del <i>SMWT</i>	25
3.2.	Flujo de trabajo para las etapas establecidas en la metodología seguida.	25
3.3.	Diagrama del nuevo oscilador local.	27
3.4.	Diagrama de funcionamiento de conectores dentro del sintetizador [30].	28
3.5.	Interfaz de configuración para el sintetizador MLSP-4016.	29
3.6.	Diseño en PCB (izq.) y esquemático (der.) de la placa adaptadora de pines.	32
3.7.	Vista frontal (izq.) y trasera (der.) de la placa adaptadora de pines.	33
3.8.	Reconocimiento por parte del <i>Raspberry Pi</i> , del sintetizador como <i>HID</i>	33
3.9.	Diagrama de flujo de <i>synthesizer.c</i>	35
3.10.	Monitoreo de temperatura y bits de estado en <i>synthesizer.c</i>	37
3.11.	Demostración de envío de comandos y recepción de respuestas en <i>synthesizer.c</i>	38
3.12.	Multiplicador X4 activo de frecuencia.	39
3.13.	Configuración de mediciones del multiplicador para la banda de 36 – 50 GHz.	40
3.14.	Resultado de la multiplicación en la banda de 36 – 50 GHz.	40
3.15.	Configuración de mediciones del multiplicador para la banda de 50 – 60 GHz.	41
3.16.	Resultado de la multiplicación en la banda de 50 – 60 GHz.	42
3.17.	Atenuador variable de perilla.	42
3.18.	Acoplador direccional.	43
3.19.	Detector de Potencia de Banda Ancha UD.	44
3.20.	<i>Raspberry Pi</i> 4B.	45
3.21.	Diseño en PCB y esquemático de la placa de control del LO.	46
3.22.	Vista frontal de la placa de control del LO.	47
3.23.	Vista trasera de la placa de control del LO.	48

3.24.	Módulo conversor análogo digital ADS1115.	49
3.25.	Lectura de voltaje de RIGOL y <i>ADS1115</i> entre 0 V y 4 V para distintos pasos.	50
3.26.	Módulo conversor de nivel.	51
3.27.	Módulo relé.	52
3.28.	Diagrama de flujo de <i>rpiserver</i>	53
3.29.	Respuesta del comando “V” en <i>rpiserver</i>	54
3.30.	Respuesta del comando “P” en <i>rpiserver</i>	55
3.31.	Respuesta del comando “S” y “T” en <i>rpiserver</i>	55
3.32.	Respuesta del comando “Q” y “C” en <i>rpiserver</i>	56
3.33.	Monitoreo de frecuencia, voltaje y potencia en <i>rpiserver</i>	56
3.34.	Ejemplo de envío y recepción de frecuencia.	57
3.35.	Configuración de mediciones del sistema de control para el oscilador local.	59
3.36.	<i>Raspberry Pi</i> bajo el <i>hat</i> de la placa de control para el oscilador local.	60
3.37.	Ejemplo de enclavamiento y desenclavamiento de la señal y estado del relé, en físico.	60
3.38.	Ejemplo de enclavamiento y desenclavamiento de la señal y estado del relé, en la terminal.	61
3.39.	Medición de la señal de control cuando el sintetizador se encuentra en condiciones de operación.	62
3.40.	Ejemplo en la terminal del <i>Raspberry Pi</i> del funcionamiento del sistema de control para el oscilador local.	63
3.41.	Modos de conexión entre detector y conversor.	64
3.42.	Ajuste polinómico e interpolación PCHIP de los datos de la tabla 3.14.	65
3.43.	Gráfico de voltaje y potencia entre 0 V y 4 V para distintas frecuencias.	66
3.44.	Configuración de mediciones para la cadena de oscilador local.	67
3.45.	Diagrama de la configuración de mediciones utilizada para la banda 36 – 50 GHz del oscilador local.	68
3.46.	Configuración de mediciones utilizada para la banda 36 – 50 GHz para las pruebas del oscilador local.	69
3.47.	Diagrama de la configuración de mediciones utilizada para la banda 50 – 60 GHz del oscilador local.	70
3.48.	Configuración de mediciones utilizada para la banda 50 – 60 GHz para las pruebas del oscilador local.	71
4.1.	Medición de voltaje, con y sin atenuador, para la banda de 33.5 – 50 GHz.	73
4.2.	Medición de voltaje, con y sin atenuador, para la banda de 50 – 60 GHz.	74
4.3.	Terminal de cliente y servidor ante el envío y recepción de frecuencia.	75
4.4.	Señales generadas por el oscilador local.	76
4.5.	Potencia medida, con y sin atenuador, en analizador de señales PXA y detector UD.	77
4.6.	Ajuste polinómico e interpolación PCHIP con el Modelo 2.	80
4.7.	Potencia obtenida en Analizador PXA y Detector UD, con y sin atenuador, para la banda de 33.5 – 50 GHz.	81
4.8.	Potencia obtenida en Analizador PXA y Detector UD, con y sin atenuador, para la banda de 50 – 64 GHz.	83
4.9.	Atenuación y Potencia, para una frecuencia de 50 GHz.	84
A.1.	Configuraciones iniciales para <i>Raspberry Pi Imager</i>	91
A.2.	Configuración del archivo “wpa_supplicant.conf” para el <i>Raspberry Pi</i>	91

A.3.	<i>Software Angry IP Scanner</i> , donde se muestra la dirección IP asignada al <i>Raspberry Pi</i>	92
A.4.	Configuración de parámetros para dirección IP fija en <i>Raspberry Pi</i>	94
B.1.	Diagrama de la distribución del repositorio <i>Instmini</i>	96
E.1.	Vista 3D frontal y trasera de la Placa RPI-LO, versión 2.	118
E.2.	PCB y esquemático de la Placa RPI-LO, versión 2.	118

Capítulo 1

Introducción

1.1. Motivación y Formulación del Problema

Los cielos Chilenos, en particular los del Norte de Chile, son una zona privilegiada y de suma relevancia para la observación astronómica. Esto ha posibilitado la instalación de telescopios de gran envergadura como, por ejemplo, los de Cerro Tololo y Cerro Pachón (Observatorio AURA), La Silla y Paranal (Observatorio ESO) y en el Llano de Chajnantor (ALMA), que día a día contribuyen a la comunidad con grandes descubrimientos. Dentro de las ramas de observación, el énfasis del presente trabajo de título está en la radioastronomía, que corresponde a un área de la astronomía que se encarga de estudiar los fenómenos y objetos del universo. Esto, a diferencia de otras áreas como la de infrarrojo u óptico, lo realiza a través de la recepción de radiación electromagnética en la región de radio del espectro electromagnético.

En particular, el presente trabajo de título concierne al *Southern Millimeter Wave Telescope* (*SMWT* por sus siglas en inglés). Este radiotelescopio, con el prefijo “radio” ya que se encarga de observar la región de radio del espectro electromagnético, fue inicialmente instalado en el Observatorio Interamericano de Cerro Tololo en 1982. Posteriormente, en 2004, el receptor fue trasladado al Observatorio Astronómico Nacional en Cerro Calán y, en 2010, se construye un nuevo edificio que alberga por completo al telescopio. En la actualidad, el *SMWT* forma parte del Laboratorio de Ondas Milimétricas (*Millimeter-Wave Laboratory*, *MWL* por sus siglas en inglés) del Departamento de Astronomía de la Universidad de Chile. El radiotelescopio se encuentra en proceso de actualización y modernización de sus equipos e instrumentos con el propósito de ser utilizado en docencia e investigación en la Banda 2 y 3 (67 – 116 GHz).

Dentro de los planes de actualización del radiotelescopio, conocido comúnmente como “Mini” debido a su pequeño tamaño, es crucial la incorporación e integración de un nuevo oscilador local. El oscilador juega un rol fundamental dentro del procesamiento de la señal recibida, puesto que contribuye a la conversión de frecuencia hacia rangos menores dentro de los cuales se puede encausar la onda. La integración del oscilador no solamente sería un paso más que cumplir dentro de la lista de etapas de la actualización, sino que también contribuye de manera directa a la generación de instrumentación adecuada y capaz de competir con sus similares. A pesar de que el camino para convertirse en una opción viable de ser llevada a cabo a gran escala, como en los telescopios de procedencia extranjera que se sitúan

en los observatorios del Norte de Chile, puede ser largo, el presente trabajo corresponde a una muestra de que a nivel nacional existen las condiciones para generar tecnología de punta.

Para la elaboración del nuevo oscilador local, es fundamental el estudio tanto del área de radioastronomía como de la instrumentación utilizada en radiotelescopios en la banda de interés. Dentro de esta última, resulta importante la comprensión de receptores heterodinos, los cuales son sistemas de recepción implementados en instrumentos que reciben ondas de radio. Con esto, se tienen conocimientos base para lograr una adecuada descripción de los componentes. Del mismo modo, se debe diseñar el sistema de control y de medición de potencia del oscilador local. Particularmente, el nuevo oscilador está basado en un sintetizador de frecuencia, en adición de un multiplicador, atenuador variable, detector de potencia de banda ancha y el respectivo sistema de control y comunicación.

En conclusión, el principal resultado que apunta la presente memoria, corresponde a la implementación del oscilador local y del diseño de su sistema de control y medición de potencia. Para esto, fue necesario el diseño, pruebas e implementación de tal sistema, el cual cumple con los requerimientos específicos de los dispositivos que lo componen. En términos de *hardware*, el sistema de control se compone por un *Raspberry Pi* más una placa personalizada encargada de medir la potencia del oscilador y generación de la señal de control de enclavamiento. En el lado de *software*, se utiliza una interfaz de línea de comandos para el monitoreo, medición de potencia y sintonización de frecuencia.

1.2. Objetivos del Trabajo de Título

1.2.1. Objetivo General

Las labores fueron desarrolladas en el Laboratorio de Ondas Milimétricas de la Universidad de Chile, con principal presencia del *SMWT*. Bajo este contexto, el objetivo general del presente trabajo consiste en el diseño, pruebas e implementación del sistema de control del nuevo oscilador local. Para esto, fue necesario la elaboración del sistema de control, más la medición de potencia y componentes adicionales y, posteriormente, la integración en el oscilador local, de tal modo que se pudo probar la cadena completa.

1.2.2. Objetivos Específicos

A continuación, se definen los cuatro objetivos específicos que se han planteado para el presente proyecto.

- Diseño y pruebas del sistema de control para el oscilador local.
- Medición de la potencia de salida del oscilador local con el sistema de control.
- Implementación del sistema de control en el oscilador local.
- Implementación y pruebas del oscilador local.

Tras el planteamiento del contexto y de los objetivos, es importante mencionar la estructura que presenta el actual informe. Este se compone por cinco capítulos: Introducción, Marco

Teórico y Estado del Arte, Diseño e Implementación, Resultados y Discusión y Conclusiones.

En el Capítulo 1, se realizó la Introducción al presente trabajo. Se comentó la motivación y formulación del problema, además del planteamiento de los objetivos. En el Capítulo 2, se aborda el Marco Teórico y Estado del Arte del presente tema. En particular, se introducen los principales conceptos relacionados a la radioastronomía, del radiotelescopio Mini y de receptores heterodinos. En segundo lugar, se comenta el Estado del Arte para describir y comparar referencias sobre instrumentación en radiotelescopios que se asemejen al caso actual.

En el Capítulo 3, se describe el Diseño e Implementación del sistema de control y del oscilador local. En este, se comenta la metodología, los componentes y funcionamiento del sistema, además de enseñar las configuraciones realizadas para las mediciones. En el Capítulo 4, en Resultados y Discusión, se muestran y comentan los resultados obtenidos, tanto para el sistema de control, como para el oscilador local a lo largo de toda la banda de frecuencias: 36 – 60 GHz. Finalmente, en el Capítulo 5, de Conclusiones, se da el cierre a los temas presentados y con ello, se verifica el cumplimiento de los objetivos planteados en función de los resultados obtenidos.

Es importante mencionar que se cuenta con una sección de Anexos, en la cual se describen en detalle cinco aspectos de gran relevancia. En el Anexo A, se muestran los pasos a seguir para la configuración inicial del *Raspberry Pi*. En el Anexo B, se describe la documentación reunida para el oscilador local y la estructura del repositorio *Github* en el que se trabajó. En el Anexo C, se enseñan los códigos de Python y C con los que se trabajó. En el Anexo D, se muestra el control de frecuencia que es utilizado en el oscilador local actual y, finalmente, en el Anexo E, se muestra el nuevo diseño de la placa para el sistema de control del oscilador local. Con esta información, se facilita la operación del oscilador local y la continuación del proyecto de modernización del *SMWT*.

Capítulo 2

Marco Teórico y Estado del Arte

2.1. Marco Teórico

2.1.1. Rol de la Astronomía

La astronomía ha estado ligada al ser humano desde tiempos de antaño. Corresponde a una de las ciencias naturales con más historia y se ve reflejado en los diversos registros de las civilizaciones más antiguas, los cuales hoy en día se pueden asociar a la visión e interpretación que en ese entonces se le daba a los objetos y fenómenos del cielo. La investigación astronómica ha cambiado en diversas ocasiones los paradigmas sobre la humanidad y el rol, si es que tiene, el ser humano en el universo y es mediante tales descubrimientos el cómo se ha entendido y dimensionado el tamaño de los objetos que se encuentran en nuestro entorno.

Naturalmente, la astronomía comienza desde observaciones a ojo desnudo, ya que se limita a la radiación que es posible percibir a través de nuestros ojos. Así fue como se asentaron las bases de los desarrollos de las leyes que describen el movimiento de los planetas y cuerpos celestes. Mediante observaciones se logró comprobar la teoría. En sus inicios, el principal medio de observación correspondía al ojo humano y, con el transcurso del tiempo, fue ayudado por dispositivos mecánicos para medir posiciones de objetos celestes en el cielo. El telescopio fue inventado en el siglo XVII y en 1609, Galileo Galilei realizó sus primeras observaciones con este instrumento [1]. Únicamente en las últimas décadas comienza el desarrollo de múltiples instrumentos electrónicos, como receptores, dedicados al estudio de la radiación electromagnética proveniente del espacio.

2.1.1.1. Radiación Electromagnética

La luz visible constituye únicamente una fracción diminuta del espectro electromagnético, como se muestra en la figura 2.1. Dentro de este espectro, se encuentran diversos tipos de ondas electromagnéticas, caracterizadas por sus diferentes frecuencias y longitudes de onda. Debido al fenómeno de dualidad onda-partícula, las ondas electromagnéticas exhiben comportamientos tanto de onda como de partícula. Además, su energía se relaciona con la frecuencia, lo que les permite emitir y absorber energía en frecuencias específicas, dando lugar a líneas espectrales distintivas [2].

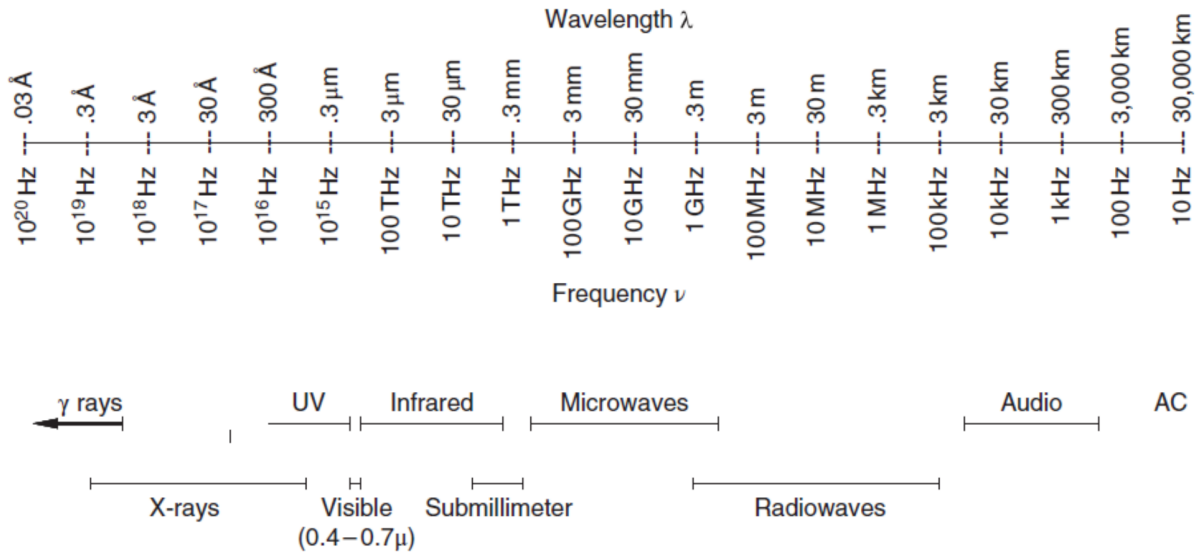


Figura 2.1: Espectro Electromagnético [3].

Los cuerpos celestes pueden producir radiación por varios mecanismos como, por ejemplo, radiación termal en el caso de un cuerpo negro, y es por medio de esta radiación que es posible obtener su información. Además, si el cuerpo se encuentra en movimiento y emite un fotón cuya frecuencia es de ν_0 , el mismo fotón será recibido por un observador, en reposo con respecto al cuerpo emisor, con otra frecuencia ν [3]. Este fenómeno se conoce como Efecto Doppler y dependiendo del caso en estudio, se puede caracterizar la velocidad radial con la que se aleja o acerca una fuente, produciendo un corrimiento en las líneas espectrales que se observen.

El mecanismo de radiación y Efecto Doppler fue inicialmente aplicado a la astronomía óptica, pudiéndose analizar la composición química y temperatura de objetos celestes, además de poder determinar el movimiento de cuerpos masivos ya sea si se están acercando (corrimiento al azul) o alejando (corrimiento al rojo). El estudio hacia otras frecuencias se considera un hito reciente puesto que solamente en 1931, el ingeniero en telecomunicaciones Karl Jansky, descubrió una fuente de ruido que interfería en las comunicaciones de radio. En su momento, sus aportes no fueron considerados y, tras casi 20 años, se confirmó que el ruido que Jansky había captado, provenía desde el centro de la Vía Láctea. Con esto, el auge de la radioastronomía, donde se cubre un rango de frecuencias que va de unos pocos megahertz (longitud de onda $\lambda = 100$ m) hasta al menos 300 GHz ($\lambda = 1$ mm), es inminente ya que permite extender el espectro electromagnético observable por grandes órdenes de magnitud [1].

2.1.1.2. Detección de Señales en la Banda de Microondas

Afortunadamente, desde la superficie de la tierra, la atmósfera es transparente a las ondas de radio. No obstante, no ocurre lo mismo para otras bandas de frecuencia, como el infrarrojo o ultravioleta, ya que existe un porcentaje alto de absorción por parte de la atmósfera. La observación de radio tiene mayores grados de libertad con respecto a la ubicación del telescopio, pero siempre se mantiene alejado de centros urbanos para así mantener la aislación de señales de transmisión de radio, televisión e internet. El intervalo en el cual se hace posible la observación de ondas de radio se conoce como Ventana de Radio. Cubre longitudes de

onda entre $\lambda \approx 20\text{m}$ ($\nu \approx 15\text{MHz}$, bajas frecuencias) hasta $\lambda \approx 0.2\text{mm}$ ($\nu \approx 1.5\text{THz}$, altas frecuencias). Estos límites no son fijos ya que depende de variaciones tanto de altitud, posición geográfica y tiempo [4].

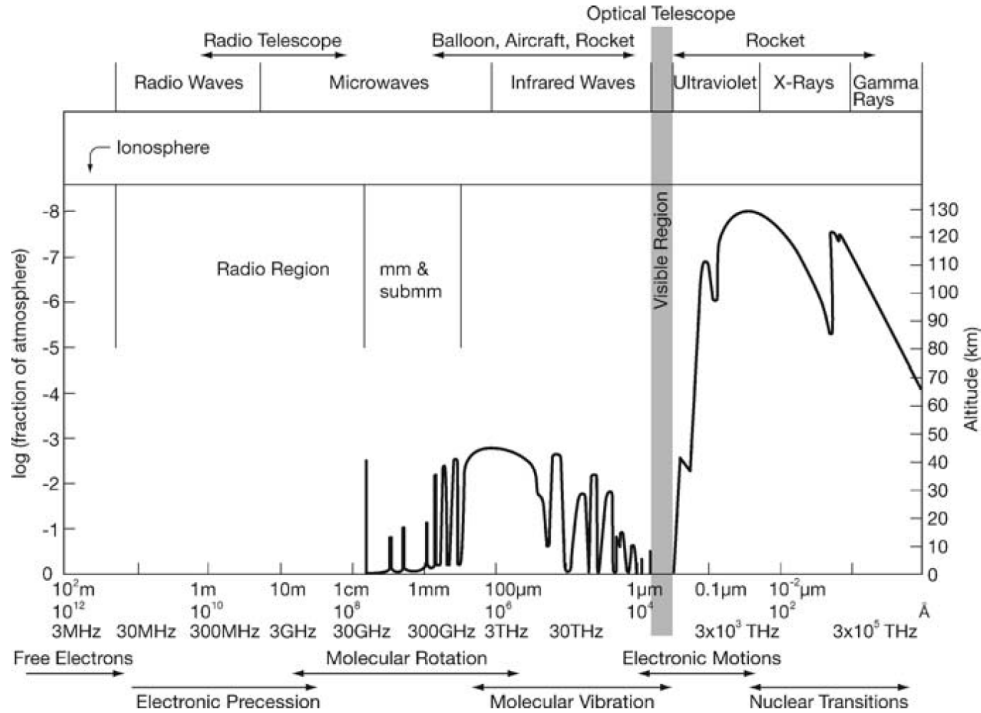


Figura 2.2: Transmisión de la atmósfera terrestre para la radiación electromagnética [4].

La curva de la figura 2.2 muestra, en el lado derecho, la altura desde el nivel del mar en donde es posible observar radiación proveniente desde el espacio. Es importante notar que solo dos ventanas son observables a nivel del mar: la ventana de radio y visible. En el lado izquierdo, se muestra la altura en la atmósfera, en términos de la fracción que atraviesa la radiación, en la cual la radiación es atenuada.

Las señales que provienen desde el espacio, recorren distancias considerables antes de llegar a la tierra, produciendo que la potencia de estas ondas sea muy baja. Es por esto que, por un lado, se hace necesaria instrumentación adecuada para la recepción de señales en tales niveles, a la par de una antena receptora con altos niveles de directividad y resolución angular.

2.1.1.3. Importancia de la Banda 67-116 GHz

Gran parte de los conocimientos que hoy en día existen sobre la estructura de la Vía Láctea, provienen de observaciones de radio, en particular del monóxido de carbono (CO). Esto se debe a que esta molécula es una buena trazadora del hidrógeno molecular (H_2), la cual es una molécula simétrica y no tiene un dipolo eléctrico permanente y, por ende, no es fácil de excitar para emitir radiación. De hecho, una parte de estos estudios fueron realizados desde el hemisferio sur, con el radiotelescopio (*SMWT*) en el Observatorio Interamericano Cerro Tololo [5].

Inicialmente, el *SMWT*, también conocido como “Mini” debido a su tamaño, fue dise-

ñado para observar las líneas de $^{12}\text{C}^{16}\text{O}$ a 115 GHz y $^{13}\text{C}^{16}\text{O}$ a 110 GHz. Con posteriores modificaciones en sus componentes, se logró ampliar el rango de observación a la banda de 84 – 116 GHz, para que también fuese posible la observación de la molécula HCN. Con el presente trabajo de título, el rango será nuevamente modificado a 67 – 116 GHz, lo cual es compatible con el interés de instrumentación y científico. Dentro de este último rango, las líneas espectrales de interés de las moléculas se muestran en la siguiente tabla 2.1.

Tabla 2.1: Líneas de emisión para la banda 67 – 116 GHz de interés para la investigación con el *SMWT*.

Fórmula Química	Frecuencia [GHz]	Fórmula Químca	Frecuencia [GHz]
^{12}CO	115.271	DCN	72.431
^{13}CO	110.201	N_2H^+	93.171
C^{18}O	109.782	H38 α	115.27
HCO^+	89.189	H39 α	106.74
H^{13}CO^+	86.754	H40 α	99.02
HCN	88.632	H41 α	92.03
H^{13}CN	86.340	H42 α	85.69
CS	97.981	H43 α	79.91
HC_3N	100.076	H44 α	74.65
CN	113.144	H45 α	69.83
DCO^+	72.039		

Tanto para el rango de 67 – 116 GHz, como para el estudio de frecuencias mayores y/o menores, es necesario de una instrumentación adecuada. Es por esto que, a continuación, se presentan las principales características y funcionamiento de un receptor de radio, como el usado en el *SMWT*.

2.1.2. Receptor Heterodino

La onda que es recibida desde el espacio es transformada en una señal eléctrica por un receptor, los cuales son muy similares a los utilizados en telecomunicaciones. Estos dispositivos se denominan receptores (súper) heterodinos, o receptores coherentes, y el diagrama con sus principales componentes puede ser encontrado en la figura 2.3. La principal función de un receptor heterodino, consiste en convertir la señal recibida en una de baja frecuencia, conservando la información de la fase de la onda original. La amplitud, a pesar de que pueda ser amplificada por el receptor, puede ser recuperada a través de la calibración del instrumento y, de esta forma, es posible conocer la intensidad de la fuente.

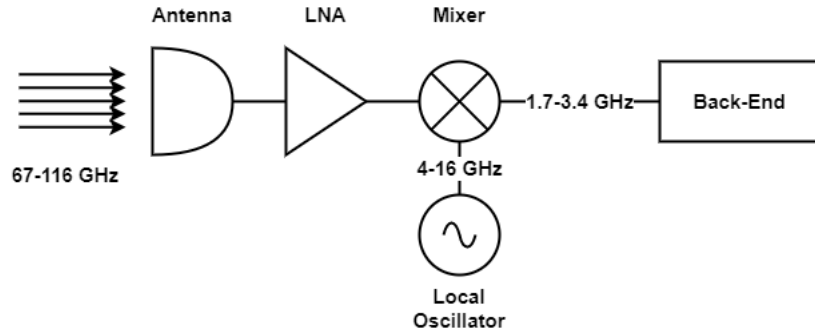


Figura 2.3: Diagrama de un receptor heterodino [6].

La radiación incidente (señal de radiofrecuencia, *radio frequency*, *RF* por sus siglas en inglés) es colectada por la antena, donde es confinada por una guía de onda. Luego, la señal es dirigida por un amplificador de bajo ruido (*low noise amplifier*, LNA por sus siglas en inglés) que amplifica la señal, generando ruido muy bajo. Posteriormente, en el mezclador (*mixer* en inglés), se mezcla la señal amplificada con una de referencia generada por el oscilador local (*local oscillator*, LO por sus siglas en inglés) y produce una señal con frecuencia igual a la operación de frecuencias entre el LO y la señal RF. El resultado, conocido como frecuencia intermedia (*Intermediate Frequency*, por sus siglas *IF* en inglés), entra a la parte trasera para extraer la información deseada en función de cada observación [6].

2.1.2.1. Ancho de Banda de RF e IF

El ancho de banda es el tamaño, en frecuencia, de la banda que se puede observar. Con esto, el ancho de banda *RF* corresponde al ancho de banda que el telescopio puede observar, antes de la conversión a frecuencias bajas. Asimismo, el ancho de banda *IF* es el tamaño de la banda luego de la conversión y que es posteriormente digitalizada.

Este parámetro es uno de los aspectos fundamentales de un receptor y determina los requisitos tecnológicos que se deben utilizar en la parte delantera. Para este tipo de aplicaciones, de radioastronomía, se busca tener un gran rango, de modo que se permita la visualización de un mayor número de líneas espectrales o ancho de banda en el continuo.

2.1.2.2. Sobre el Ruido

En gran parte de los procesos de observación radioastronómica, los sistemas de recepción procesan señales muy débiles y que pueden ser alteradas por el ruido intrínseco que se produce por los componentes. En sistemas típicos de microondas, la señal de entrada atraviesa todos los dispositivos que componen la cadena y, debido a esto, la relación de señal-a-ruido puede verse degradada. Si se conoce la temperatura de ruido de cada etapa individual, entonces se puede determinar la temperatura de ruido del sistema conectado en cascada. La expresión que representa este parámetro se muestra en la ecuación 2.1.

$$T_{sistema} = T_1 + \frac{T_2}{G_1} + \frac{T_3}{G_1 G_2} + \dots + \frac{T_n}{\prod G_i} \quad (2.1)$$

De la ecuación 2.1, T_i y G_i corresponden a la temperatura de ruido equivalente y ganancia, respectivamente, de la etapa i [6]. De esta forma, se busca que la primera fase sea enfriada,

utilizando nitrógeno (77° C) o helio (4° C) para disminuir el ruido de los dispositivos, puesto que es la etapa en la que domina el ruido de la cadena de componentes [7].

2.1.2.3. Amplificador de Bajo Ruido

Otra aspecto importante que se observa de la figura 2.3 y de la ecuación 2.1, corresponde a la importancia de la primera etapa que se realiza en el LNA. Los dispositivos más cercanos a la antena son los que mayor ruido generan, por lo que el LNA debe ser escogido de manera tal que su impacto en el sistema sea bajo. La elección se encuentra en función de dos factores: la tecnología disponible de los elementos circuitales y el diseño del amplificador que genere el menor ruido.

Dependiendo de la tecnología disponible, el dispositivo contiguo a la antena puede ser tanto un LNA como un Transductor de Modos Ortogonales (*Orthomode Transducer*, OMT por sus siglas en inglés), el cual permite estudiar por separado las distintas polarizaciones de la onda electromagnética. Usualmente, el amplificador se encuentra dentro de la etapa de criogenia, de tal modo que el ruido producido por sus componentes sea minimizado.

2.1.2.4. Mezclador

Un mezclador presenta presenta dos entradas. Una de ellas puede ser una señal generada por un LO, con frecuencia f_{LO} , y se puede representar como $v_{LO}(t) = \cos(2\pi f_{LO}t)$. En la otra entrada, se puede ingresar la señal capturada con frecuencia f_{RF} , que es expresada de forma similar: $v_{RF}(t) = \cos(2\pi f_{RF}t)$. De esta forma, a la salida del mezclador se encuentra el producto de ambas señales:

$$\begin{aligned} v_{IF}(t) &= K \cdot v_{LO}(t) \cdot v_{RF}(t) = K \cdot \cos(2\pi f_{LO}t) \cdot \cos(2\pi f_{RF}t) \\ &= \frac{K}{2} \cdot [\cos(2\pi(f_{LO} - f_{RF})t) + \cos(2\pi(f_{LO} + f_{RF})t)] \end{aligned} \quad (2.2)$$

En la ecuación 2.2, K es una constante que representa las pérdidas en el mezclador. De esta forma, la señal RF generada está dada por una señal que contiene frecuencias con la suma y diferencia de las frecuencias de las señales entrantes. Lo anteriormente descrito representa a un mezclador ideal, sin embargo, en la realidad, se deben incorporar otros factores en el análisis, como las características no lineales de voltaje y corriente de los elementos circuitales. Un ejemplo de comportamiento no lineal, se puede encontrar en un diodo *Schottky*, donde la corriente i que circula puede ser expresada en series de potencias [6].

Un posible tipo de mezclador es el mezclador de rechazo de imagen, en donde es posible trabajar de manera independiente las bandas superior e inferior (*Upper Sideband (USB)* y *Lower Sideband (LSB)*, respectivamente), con respecto al oscilador local.

Un mezclador sub armónico, o *sub-harmonic mixer*, es también utilizado para transformar una frecuencia de radio a una de frecuencia intermedia. La principal diferencia con un mezclador, corresponde a la frecuencia de operación necesaria del LO. Mientras que para realizar la conversión de frecuencia, un mezclador produce la señal de frecuencia intermedia: $f_{IF} = f_{LO} \pm f_{RF}$, un mezclador sub armónico genera la misma señal IF, pero la frecuencia del LO es una fracción de la frecuencia requerida para el mezclador. La expresión viene dada por: $f_{IF} = N \cdot f_{LO} \pm f_{RF}$, la cual representa un mezclador sub armónico de orden N [6].

2.1.2.5. Oscilador Local

Corresponde a un dispositivo electrónico que genera una señal de frecuencia constante y estable en sistemas de comunicación, como radios, televisores, transmisores y receptores. El propósito principal del oscilador local, es generar una señal de frecuencia precisa que se mezcle con una señal de entrada en el proceso de modulación o demodulación. Esto se logra mediante la mezcla de la señal del LO con la señal de entrada en un mezclador, que produce una señal con frecuencia igual a la suma y la diferencia de frecuencias de las dos señales.

En los receptores, el oscilador local se utiliza para sintonizar diferentes frecuencias de radio. Al ajustar la frecuencia del oscilador local, se puede seleccionar la frecuencia específica deseada para la recepción. Por otro lado, en los transmisores, el oscilador local se utiliza para generar la portadora en la modulación de la señal de entrada. Es importante que el oscilador local genere una señal de frecuencia estable y precisa para evitar interferencias y distorsiones en el proceso de modulación o demodulación. Para ello, se utilizan componentes electrónicos especializados, como cristales piezoeléctricos o circuitos de oscilador controlados por voltaje (VCO, por sus siglas en inglés).

2.1.2.6. Multiplicador de Frecuencia

Corresponde a un dispositivo que genera una señal de salida cuya frecuencia es un múltiplo de la frecuencia de entrada. El multiplicador activo de frecuencia típicamente consta de etapas de amplificación y no linealidad controlada. La señal de entrada, que tiene una frecuencia específica, se amplifica y luego se somete a un proceso no lineal que genera armónicos de la frecuencia original. Estos armónicos se filtran y seleccionan para obtener la frecuencia deseada multiplicada [6].

A diferencia de los multiplicadores de frecuencia pasivos, que utilizan dispositivos no lineales como diodos para realizar la multiplicación, los multiplicadores activos emplean amplificadores activos, como transistores bipolares o de efecto de campo (FET). Estos amplificadores activos permiten una mayor eficiencia y flexibilidad en el diseño.

2.1.2.7. Atenuador

Su principal función consiste en disminuir el nivel de potencia de una señal eléctrica, sin distorsionarla. Esto suele ser logrado mediante tres formas diferentes: por dispersión, reflexión o absorción. De esta forma, la señal atenuada, a pesar de tener un nivel de potencia menor, mantiene la misma fase y frecuencia de la señal de entrada.

Existen diferentes tipos de atenuadores, los cuales utilizan distintas técnicas para lograr su funcionalidad. Los atenuadores fijos tienen definido por fábrica el nivel de atenuación y este no puede ser cambiado. Los atenuadores variables ofrecen un nivel de ajuste que puede ser modificado. Y, finalmente, los atenuadores programables son dispositivos digitales que pueden ser programados para entregar diferentes niveles de atenuación.

2.1.2.8. Acoplador Direccional

Corresponde a un dispositivo pasivo de microondas y suele ser usado en técnicas de división o combinación de potencia. En el presente caso, el dispositivo será utilizado para dividir la señal de entrada en dos señales de salida, con menor potencia.

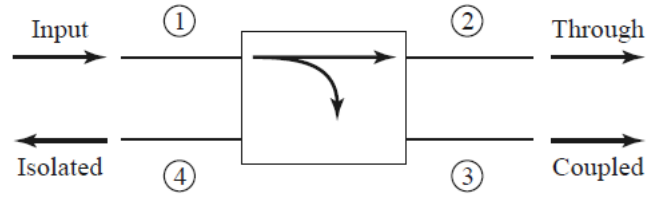


Figura 2.4: Diagrama comúnmente utilizado que representa un acoplador direccional [6].

En la figura 2.4, se muestra el diagrama que representa un acoplador direccional y sus cuatro terminales. Para la caracterización de un acoplador direccional, se suelen utilizar cuatro parámetros. El factor de acoplamiento, que indica la fracción de la potencia de salida que es acoplada en el puerto de salida. La directividad corresponde a una medida de la capacidad del acoplador de aislar las ondas, hacia adelante y hacia atrás. El aislamiento se define como la potencia que se entrega al puerto desacoplado. Finalmente, las pérdidas de inserción, indican la potencia de entrada que es ingresada al puerto pasante, disminuida por la potencia entregada a los puertos acoplados y aislados [6].

2.1.2.9. Detector de Potencia de Banda Ancha

Este tipo de detectores suelen utilizar diodos Schottky, debido a que por su tipo de juntura, puede ser operado a altas frecuencias. Con esto, el detector utiliza una señal RF de entrada y la convierte en una potencia de salida, la cual suele ser representada por voltaje continuo [6].

En el presente trabajo de título, se trabajó con un detector de potencia de banda ancha. Este dispositivo está diseñado para operar en un intervalo de 40 – 60 GHz y, a su salida, entrega un voltaje negativo en función de su sensibilidad y frecuencia de entrada. En secciones posteriores del presente informe, se comentarán con profundidad las características de este detector [8].

2.1.3. Una Breve Historia del Radiotelescopio Mini

En la década de los 70 y 80, el Dr. Patrick Thaddeus y su equipo de la Universidad de Columbia desarrollaron el radiotelescopio *Millimeter-Wave Telescope*, *MWT* por sus siglas en inglés, para caracterizar el monóxido de carbono (CO) en el espacio [9]. Este instrumento, operado en la Universidad de Columbia y en el *Harvard-Smithsonian Center for Astrophysics*, obtuvo datos significativos que buscaban trazar las nubes moleculares en CO en toda la Vía Láctea. Para ser completado, se construyó un telescopio gemelo llamado *Southern Millimeter-Wave Telescope*, *SMWT* por sus siglas en inglés, y fue instalado en el Observatorio Cerro Tololo en 1982. En conjunto, estos telescopios realizaron el estudio galáctico más extenso del CO interestelar. Sin embargo, en 1998, el telescopio en Cerro Tololo cesó operaciones, siendo posteriormente trasladado al Laboratorio de Ondas Milimétricas en Cerro Calán, sede del Observatorio Astronómico Nacional de Chile, en donde fue modernizado y utilizado para

estudios acotados de nubes moleculares en la Galaxia.

2.1.3.1. Oscilador *Gunn*, el Primer Gran Cambio

En 2005, se trasladó el receptor al Laboratorio de Ondas Milimétricas de Cerro Calán y se da inicio a una actualización de la instrumentación obsoleta. Walter Max-Moerbeck, estudiante de ingeniería eléctrica, reemplazó el oscilador Klystron por uno Gunn y se amplió el rango de observaciones a frecuencias entre 85 y 116 GHz [10]. Además, en 2006, Nicolás Reyes, otro estudiante, integró un amplificador *HEMT*, proporcionado por el *California Institute of Technology*, con lo que se mejoró la ganancia y se redujo la temperatura de ruido [11]. En 2009, se evaluó la opacidad del cielo de Santiago, resultando favorable para observaciones radioastronómicas en Cerro Calán. Luego de trasladar la antena del Mini desde Cerro Tololo, se realizó el montaje y la construcción del edificio. En 2010, Pablo Vásquez, otro estudiante, llevó a cabo la caracterización y puesta en marcha del Mini, que volvió a operar después de más de 10 años y, con ello, se convirtió en una herramienta central para la docencia, además de ser una excelente plataforma de desarrollos de instrumentación e investigación astronómica [12].

2.1.3.2. El Segundo Cambio: Diodos *Schottky*

A pesar de la exitosa puesta en marcha del radiotelescopio, se identificaron áreas de mejora. Primero, se incorporó un triplicador de frecuencia, basado en diodos *Schottky*, y se amplió el ancho de banda operacional aunque con baja eficiencia. Para compensar esto, se añadió un amplificador de potencia [13]. Además, se estudió el patrón de radiación mediante un oscilador YIG y un multiplicador para mejorar la estabilidad y control de la frecuencia generada [14]. En segundo lugar, dos tesis de doctorado del Laboratorio de Ondas Milimétricas contribuyeron al desarrollo. Una enfocada en la digitalización de instrumentación radioastronómica, diseñando un espectrómetro con separación de banda lateral digital (2SB) para un alto cociente de rechazo de 40 dB [15]. La otra, se centró en el diseño, construcción y pruebas del receptor con separación de banda lateral, implementado en 2014 para mejorar el sistema receptor [16].

2.1.4. Especificaciones Técnicas del SMWT

A continuación, se describen las especificaciones técnicas del radiotelescopio. Es importante destacar que pueden existir, y existirán, cambios en los sistemas. En síntesis, se tienen 3 grandes áreas: sistema óptico, procesamiento de señales y sistema computacional. A continuación, en la figura 2.5 se muestra el diagrama general del radiotelescopio [14], en la cual se pueden apreciar los métodos de comunicación entre los diferentes elementos en conjunto con los principales componentes. Posteriormente, se comentará cada sistema del *SMWT*.

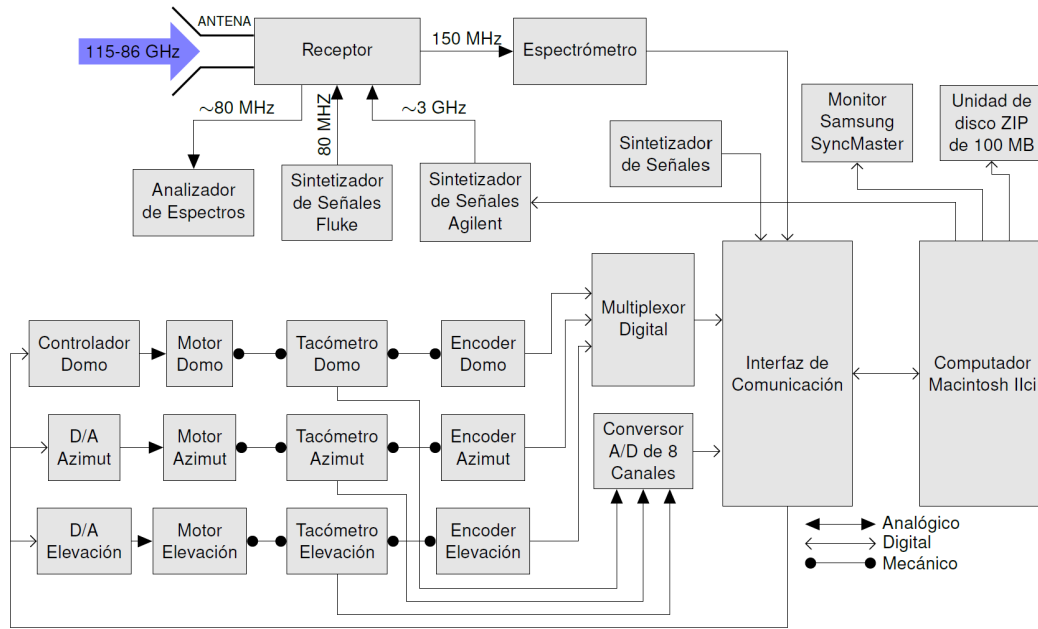


Figura 2.5: Diagrama general del radiotelescopio Mini [14].

Además, los principales parámetros y especificaciones del Mini se muestran en la siguiente tabla 2.2.

Tabla 2.2: Principales especificaciones técnicas del radiotelescopio.

Parámetro	Valor
Antena	1.2m Cassegrain
Precisión Superficial	20 μ m
HPBW	8.8 minutos de arco (a 115.3 GHz) ^a
SLL	Menor a 18 dB ^a
Eficiencia del haz	82% ^a
Ancho de banda	128 MHz, 25.6 MHz
Resolución Espectral	1 MHz, 0.1 MHz
Temperatura de ruido	380 K (SSB)
Razón f/D equivalente	3.79

^a Reportados en [14].

2.1.4.1. Sistema Óptico

El sistema óptico de la antena corresponde a un reflector tipo Cassegrain. La antena consiste en un reflector primario parabólico de 1.2 m y un secundario hiperbólico de 18.8 cm con su respectivo soporte y estructura de alineamiento. El patrón de radiación de la antena de campo lejano fue medido a una distancia de 4 km de Cerro Calán, específicamente en Cerro Manquehue, y fue realizado por el estudiante P. Astudillo [14] en 2013. A continuación, en la figura 2.6, se muestra un diagrama esquemático de un reflector tipo Cassegrain.

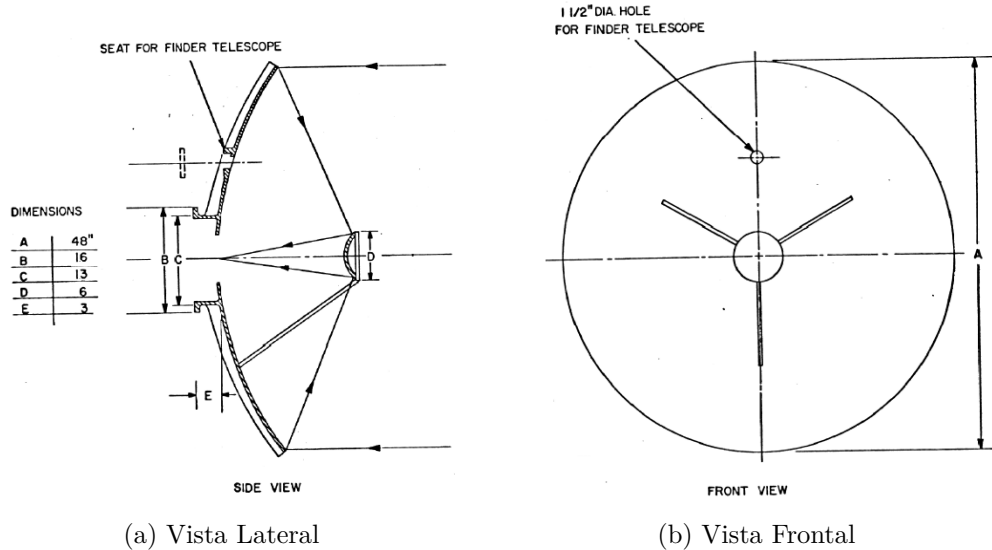


Figura 2.6: Esquema de una antena Cassegrain. En (a) se muestra la vista lateral del trazo del rayo. En (b) se muestra la vista frontal [16].

Tanto el reflector primario como secundario están contruidos de una sola pieza de aluminio con las dimensiones especificadas, las cuales son profundizadas en [14]. La bocina está compuesta de dos piezas de cobre y se conecta al receptor mediante un transductor de guía de onda, el cual realiza la transición de una guía de onda circular a una rectangular mediante cuatro secciones consecutivas [12]. La antena se encuentra montada sobre un eje horizontal, cuyo sistema utiliza motores de accionamiento directo para los ejes de azimut y elevación, permitiendo un movimiento de 5 grados por segundo. Esta velocidad facilita la metodología de alternancia de posición, una técnica de observación que alterna la fuente de interés con una región sin emisión para conocer la respuesta del instrumento. Los ejes son monitorizados por tacómetros y codificadores de posición de 16 bits con una precisión de 0.33 minutos de arco, siendo controlados por una computadora [12].

2.1.4.2. Sistema Receptor

El sistema de recepción se compone de dos grandes secciones: parte delantera y parte trasera, o en inglés *front-end* y *back-end*, respectivamente. La primera engloba los procesos que ocurren en el receptor del radiotelescopio, donde se actúa como el frente inicial, que a su vez puede ser divididas en tres bloques: amplificador de bajo ruido (LNA), oscilador local (LO) y bloque de frecuencia intermedia (IF). En la segunda etapa, se encuentra la parte trasera, la cual a la vez se compone del espectrómetro. A continuación, se describirá el funcionamiento actual de cada uno de los bloques nombrados.

- **Front-End**

El receptor recibe directamente la señal de RF. El principal objetivo de esta sección corresponde a realizar una conversión de frecuencia y entregar un valor de salida que pueda ser operado en el procesamiento posterior. Para el radiotelescopio Mini, el receptor se compone de 3 grandes bloques:

1. **Primer bloque: Amplificador de Bajo Ruido**

Actualmente, el LNA se ubica después de la antena y corresponde a una de las primeras etapas que recibe la señal incidente. En conjunto con el mezclador, estos dos componentes se sitúan dentro de la etapa de criogenia. Sin embargo, con los cambios que serán implementados en el futuro, el amplificador es precedido por un OMT más dos cadenas, una por cada polarización, para el acondicionamiento de la señal. La nueva etapa de parte delantera que será puesta en marcha en el *SMWT*, se muestra en la siguiente figura 2.7.

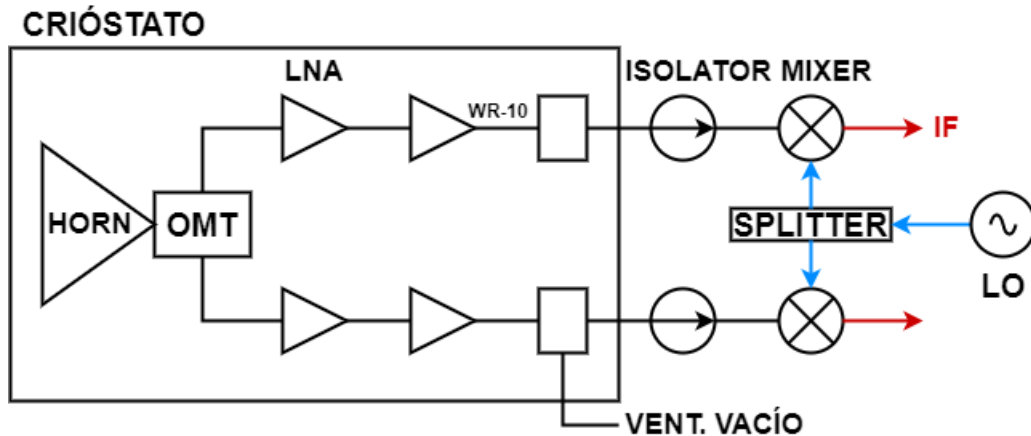


Figura 2.7: Diagrama del *front-end* y LNA.

2. Segundo bloque: Oscilador Local

Debido a los avances tecnológicos, se ha actualizado el funcionamiento y componentes del oscilador local. Como se mencionó, en lugar de utilizar un oscilador Gunn, se integra el tipo YIG el cual opera en un rango de frecuencias de 12 – 20 GHz. El principal motivo de este cambio se debe a la integración de mezcladores subarmónicos en la etapa de recepción, trabajo que se encuentra con mayor detalle en [13]. De esta forma, los requerimientos del oscilador YIG deben seguir especificaciones técnicas diferentes como, por ejemplo, que la interfaz de salida debe ser una guía de onda rectangular estándar WR-19 (Banda U), de modo que el LO debe operar entre 40 – 60 GHz. Además, se debe mantener un sistema de control de frecuencia para ajustar tanto el tono como el corrimiento de frecuencia producido por Efecto Doppler. En la figura 2.8, se muestra el diagrama de la placa del oscilador con el YIG integrado, circuito que actualmente se encuentra operando en el radiotelescopio.

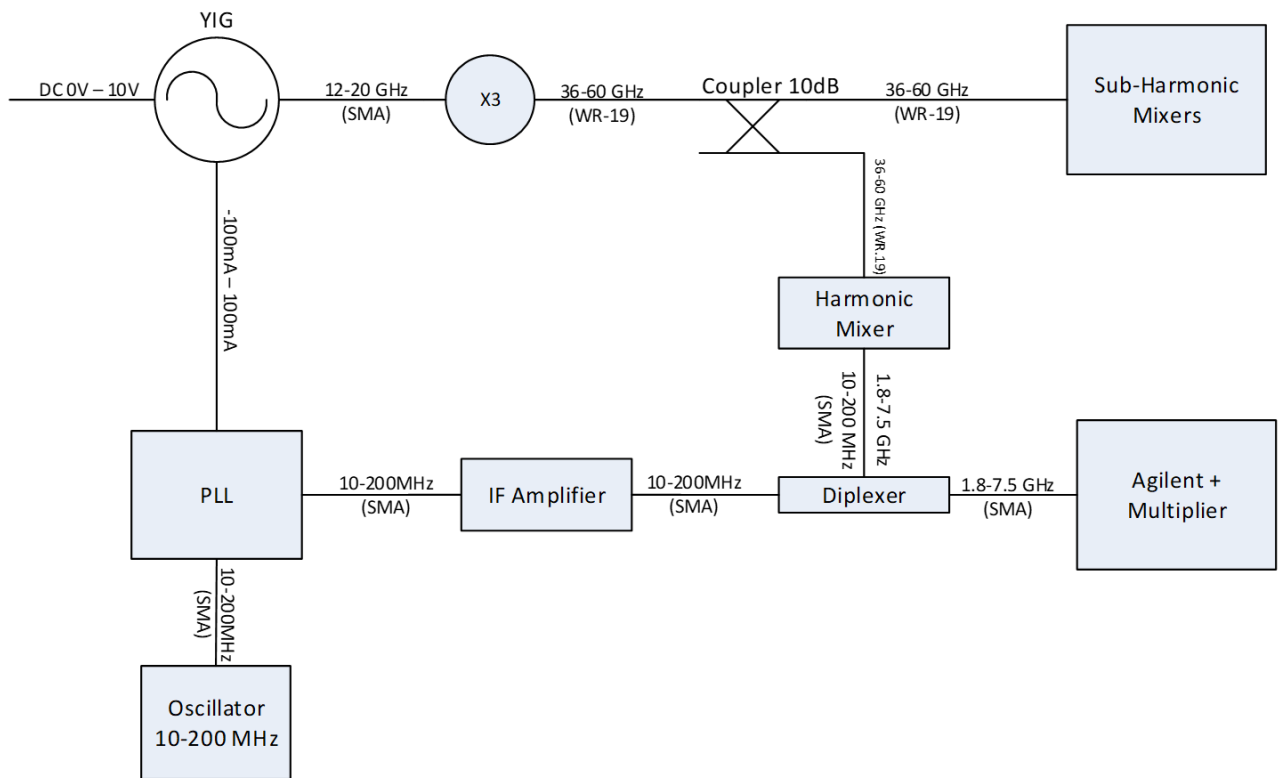


Figura 2.8: Diagrama actual de la placa del oscilador local [17].

El esquema está basado en un Estabilizador Heterodino, el cual permite tener un lazo de control utilizando un mezclador, además de tener un sistema de control que opera en bajas frecuencias (menores a los GHz). De esta forma, a continuación, se procede a describir el funcionamiento de los subsistemas del diagrama anterior.

El Oscilador YIG se caracteriza por generar un tono puro y por tener una relación lineal entre el voltaje de control y la frecuencia que produce. El modelo utilizado para este diseño tiene un corrimiento de frecuencia de 20 MHz, lo que no permite un control exacto de frecuencia y por ende surge la necesidad de un sistema de control. Para esto, se utiliza el control fino de frecuencia del oscilador, *FM Coil* en inglés, para generar un lazo de seguimiento de fase (*Phase-locked loop*, por sus siglas en inglés PLL).

El Multiplicador se requiere para que los mezcladores, que operan en una frecuencia de 36 – 60 GHz, sean compatibles con la frecuencia de salida del oscilador, correspondiente a 12 – 20 GHz. En consecuencia, el multiplicador X3 cumple el rol de intermediario entre el oscilador y mezcladores, teniendo que, además, satisfacer los requisitos de conexión para el lado de baja frecuencia (SMA), con el de alta (WR-19), como también cumplir con la potencia de operación de los mezcladores. Finalmente, para la operación del multiplicador, se añade un atenuador fijo de 10 dB a la entrada, más un atenuador variable en la salida y que entrega una potencia de 6 – 9 dBm. De esta forma, la potencia de salida del multiplicador es regulada y entregada a los mezcladores.

El Mezclador Armónico mezcla una muestra de la señal del oscilador local con una de las armónicas de una señal de referencia, generando una señal de frecuencia intermedia, que entra al circuito del PLL. En el diseño de este mezclador, se requiere de la incorporación de elementos adicionales, los cuales se encuentran también en la figura 2.8:

- (i) El acoplador direccional toma una muestra de la señal del oscilador local y la utiliza para crear el lazo de control para el estabilizador heterodino. Dentro del diseño con el que se encuentra, el acoplador de 10 dB se sitúa entre el multiplicador y el atenuador variable.
- (ii) El diplexor es utilizado para separar la señal de referencia del mezclador y la señal de frecuencia intermedia. Con esto, se satisface el requisito de dos terminales del mezclador armónico. Finalmente, como el diplexor no elimina la corriente DC producida por el mezclador, se añade un atenuador de 1 dB que actúa como carga a tierra.

El amplificador, denominado amplificador IF, se utiliza para la señal de frecuencia intermedia producida por el mezclador puesto que posee muy baja intensidad. De esta forma, para cumplir con los requisitos de potencia para que el circuito PLL detecte la señal, se cuenta con dos amplificadores conectados en cascada.

El Circuito PLL que se utiliza, se puede resumir en que, el detector de fase compara la fase del oscilador local con la de una señal de referencia estable, donde la salida dependerá de la diferencia de estas señales. El resultado generado será filtrado por el filtro de bucle, o *loop filter*, y que a la vez entrega un voltaje de control, el cual cambia la frecuencia hasta que se reduzca la diferencia de fase entre la señal de referencia con la del oscilador local. El ciclo se repite hasta que la diferencia se hace mínima y el oscilador se estabiliza, momento en el cual se dice que está enclavado o *locked*. Cuando esto suceda, se generará un voltaje de salida de 5 V que indica que el dispositivo está enclavado [17].

3. Tercer bloque: Frecuencia Intermedia

La etapa de frecuencia intermedia corresponde a una de las últimas del procesamiento de la señal original. Este bloque está encargado de adquirir la señal entregada por los mezcladores sub armónicos ($\sim 1000\text{MHz}$), realizar otro proceso de conversión de frecuencia y de amplificación, de manera tal que la muestra ($\sim 150\text{MHz}$) pueda ser procesada por el *back-end*. Además, se cuenta con aisladores que previenen posibles reflexiones de potencia entre este bloque y el anterior.

- ***Back-end***

1. **Espectrómetro**

La señal del radiotelescopio es procesada por un espectrómetro desarrollado por R. Finger, durante su tesis de doctorado. Este dispositivo, basado en avances tecnológicos digitales, utiliza un *Hardware* Informático de Arquitectura Abierta Configurable, *Reconfigurable Open Architecture Computing Hardware*, ROACH por sus siglas en inglés, para implementar un híbrido IF digital y un espectrómetro digital. Esta mejora aumenta la cantidad de canales disponibles de 512 a 2048. El resultado combinado de *front-end* y *back-end* se presenta en perfiles de velocidad, gráficos que representan la velocidad radial de la fuente celeste en el eje horizontal y la temperatura del radiotelescopio en el eje vertical. Estos perfiles son cruciales para estudiar el movimiento de gases en nubes moleculares o galaxias, así como para visualizar características de líneas de emisión o absorción, considerando el desplazamiento Doppler [15, 16]. El resultado de una observación, por ejemplo de la molécula de CO dentro de la región Orión KL, se muestra en la siguiente figura 2.9.

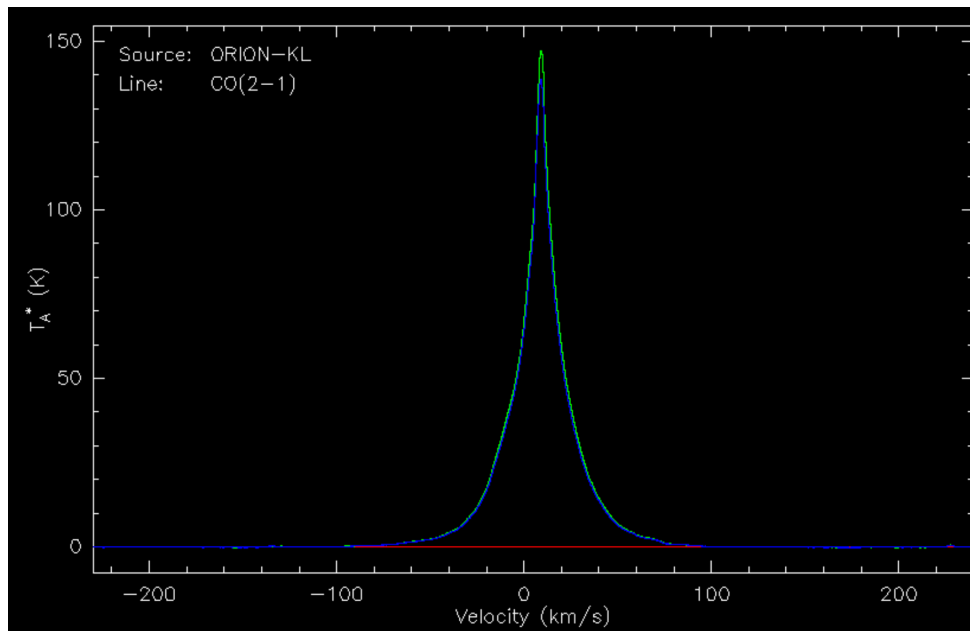


Figura 2.9: Espectro para la molécula de CO dentro de Orión KL [18].

2.1.4.3. Sistema de Control

- **Sistema Computacional**

Para que se realice la comunicación del computador con el resto de los equipos que operan en el telescopio, se utiliza una interfaz de comunicación, especialmente diseñada para este tipo de implementaciones. El programa utiliza dos ventanas principales: la ventana *Command* se utiliza para ingresar los comandos que especifican las tareas por realizar, mientras que en la ventana *Monitor* se visualiza la información del estado actual del Mini [12]. En la figura 2.10 se muestra la interfaz del programa de control.

LST	SOURCE	FOLDER	OBSERVER	SCAN#	UT DATE	UTC			
00:00:00	MAD	AEN_scans	AEN	1	3/08/11	00:00:00			
What, me worry?									
R. A.		DECLINATION		AZIMUTH		OBS MODE			
ONPOS	19:19:04	14:09:46	*	COMMAND	180.00	45.00	Switched		
OFFPOS	19:02:36	17:22:37		ACTUAL	180.00	45.00	FET: OFF		
HOME POS	0:00:00	0:00:00		ERROR	0.00	0.00	EFFICIENCY %		
GALACTIC:	48.999	0.000		DOME	180.00		99 99		
ARC	BW	ARC	BW	RMS	FWHM	INTEG	ELAPSED	ON-OFF	CALIB
OFFSET	0.0	0.0 W	0.0	0.0 N	0.00	1.3	10:00	0:00	0:15 0:01
DMAP	LIST?	I	IMAX	J	JMAX	BW	COS?	HOM?	GRID
0.0	N	1	1	1	1	0.125	Y	N	1
OFFS: 0 from none					SMPL: none				
FREQUENCY			CALIBRATION						
RESTFREQ	LINE	HARM	T CAL	T WAT	TAU W	T REC			
115271.201	12 CO	23	252 K	270 K	0.291	250 K			
SYNTH SIG	VLSR	VANT	CHPCR	<TP>	<DP>	TDOME			
103.095214	0.0	-41.4	1.01	0 K	0.00 K	117 F			
SYNTH REF	OFFRQ	DELTA	CALOK	TPOK	DPOK	T AMB			
103.095214	0.0	0.25	5 %	10.0	10.0	60 F			
						STARTUP: 3/24 18:09 UT			
						DEFAULTS			
						ORDER MARKS VELINT			
						1 1 -10.0 10.0			

Figura 2.10: Interfaz de comunicación para la operación del radiotelescopio.

Dentro de esta interfaz, el enfoque se encuentra en la sección de “*frequency*”, la cual es controlada por un *script* de control denominado “*frequency.c*” y puede ser encontrada en mayor detalle en la sección D del Anexo.

- **Montura**

El telescopio tiene una montura de tipo elevación-azimut. Posee un eje horizontal que controla el azimut, el cual a su vez está montado sobre otro eje vertical que controla la elevación. A su vez, el domo es operado directamente por el sistema computacional mencionado en el apartado anterior. De esta forma, la posición del telescopio es codificada por 16 bits en cada eje, lo cual otorga una precisión de 0.33 minutos de arco, además de que posee un control de velocidad de rotación para ambos movimientos. La instalación y montaje del Mini se muestran en la siguiente figura 2.11 [12].



(a) Montaje Mini Parte 1.



(b) Montaje Mini Parte 2.

Figura 2.11: Instalación y montaje del radiotelescopio Mini [12].

2.2. Estado del Arte

2.2.1. ALMA - Banda 2 y 3

El Gran Arreglo Milimétrico/submilimétrico Largo de Atacama, *Atacama Large Millimeter/submillimeter Array*, por sus siglas en inglés ALMA, recibe señales que cubren el rango de longitudes de ondas asociadas a frecuencias sub y milimétricas. Cuenta con 66 antenas, las cuales están equipadas con detectores sensibles a 10 tipos de bandas diferentes. En particular, las Bandas 2 y 3, que operan a 67 – 116 GHz y 84 – 116 GHz, respectivamente, cubren la misma región que el Mini. En Banda 2, ALMA espera capturar información relacionada a las propiedades del polvo en gran variedad de ambientes astronómicos como, por ejemplo, en discos planetarios en formación, galaxias y moléculas pesadas en la Vía Láctea, como es el caso de la línea de congelamiento, o *snow line* en inglés, del monóxido de carbono. Por el otro lado, en Banda 3, se puede observar la estructura a pequeña escala de nubes gaseosas, siendo relevante para la búsqueda de nuevas moléculas y la detección de gas molecular en discos que rodean estrellas jóvenes [19].



Figura 2.12: Antenas de ALMA sobre el Llano de Chajnantor [19].

En ALMA, la implementación de la Banda 2 fue la última en ser realizada. Durante este proceso, la posibilidad de combinar en un solo receptor la Banda 2 y 3 (67 – 116 GHz), toma gran importancia ya que abre la posibilidad de operar sistemas que cuenten con un gran ancho de banda. Por ejemplo, con la realización de este trabajo, se logrará que el *SMWT* opere para el mismo rango de frecuencia, lo cual va de la mano con el interés científico y el área de instrumentación [20].

2.2.2. Otros Radiotelescopios

Los radiotelescopios se encuentran dispersos por diversas zonas de la Tierra. A pesar de que muchos de gran importancia científica se sitúan en Chile, también se pueden encontrar

otros instrumentos en prácticamente todos los continentes. Cada uno de estos opera a la frecuencia a la que fue diseñado, pero por motivos del presente trabajo, se acotó la búsqueda para la banda de 67 – 116 GHz. En la siguiente tabla 2.3, se puede encontrar un resumen sobre otros radiotelescopios ubicados en diferentes países y que operan en bandas similares a la mencionada anteriormente

Tabla 2.3: Otros instrumentos de interés que operan en bandas similares a la de 67 – 116 GHz.

Radiotelescopio	Ubicación	Frecuencia (GHz)	Receptor	Antena (m)
ATCA [21]	Australia	1.1 – 105	HEMT	22 ^a
MOPRA [22]	Australia	16 – 116	HEMT	22
Nobeyama [23]	Japón	1 – 150	SIS	45
Metsähovi [24]	Finlandia	80 – 115	Schottky ^b	14
IRAM [25]	España	80 – 370	SIS	30
Green Bank [26]	EEUU	0.1 – 116	SIS / HEMT ^c	100
CfA [27]	EEUU	115	SIS	1.2

^a Corresponde a un arreglo de 6 antenas del mismo tamaño. ^b Cuenta con un receptor para cada banda. ^c No se especifica con claridad cuál es el receptor. Se hace referencia a la Banda 2 y 3 de ALMA.

En la figura 2.13, se pueden observar diferentes imágenes de otros radiotelescopios.

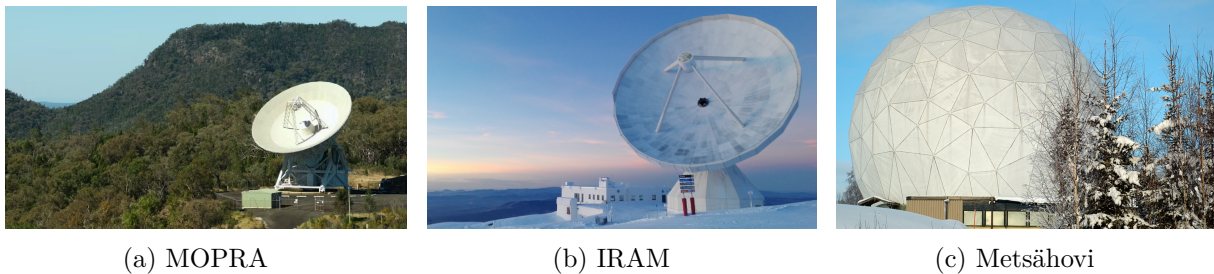


Figura 2.13: Ejemplos de otros radiotelescopios que operan en bandas similares a la de 67 – 116 GHz.

En conclusión, se encuentra una clasificación estándar del tipo de receptor que se utiliza en telescopios, la cual va en función del rango de frecuencia en el que se opera y el punto en común de los *front-ends* yace en la generación de un bajo ruido. En el área, se identifican los siguientes dispositivos [4], donde ν hace referencia a la frecuencia:

- Para $\nu < 115$ GHz, se utilizan Transistores de Alta Movilidad de Electrones, *High Electron Mobility Transistors*, *HEMT* por sus siglas en inglés, y Circuitos Integrados de Microondas Monolíticos, *Microwave Monolithic Integrated Circuits*, *MMIC* por sus siglas en inglés.
- Para $72 < \nu < 800$ GHz, se utilizan Mezcladores Superconductores, *Superconducting Mixers*, *SIS* por sus siglas en inglés. Estos dispositivos proveen poco ruido en el rango milimétrico y submilimétrico, tienen mayor sensibilidad que los mezcladores *Schottky*, requieren menor potencia proveniente del oscilador local pero deben ser enfriados a 4 K.

- Para $\nu > 900$ GHz, se utilizan Bolómetros de Electrones Calientes, *Hot Electron Bolometers*, *HEB* por sus siglas en inglés. Estos elementos son encontrados en sistemas que observan líneas espectrales a $\lambda < 0.3$ mm, lo cual abarca pequeños rangos de infrarrojo.

2.2.3. Receptores Heterodinos Modernos

Desde la introducción del proceso de heterodinación en 1902 y su aplicación en radioastronomía, el desarrollo tecnológico ha sido crucial para modernizar los receptores utilizados en detectores de microondas y radio [28]. Los receptores heterodinos modernos, basados en detectores superconductores y amplificadores semiconductores mejorados, alcanzan niveles de sensibilidad cercanos al límite teórico [29]. Dada la débil potencia de las señales incidentes, la mejora de la sensibilidad es fundamental, abordando problemas como la absorción atmosférica y el ruido del receptor. Además, para enfrentar el tiempo de exposición, se emplean enfoques como los arreglos de píxeles en el plano focal, (*focal plane arrays*), para reducir el tiempo de observación proporcional al número de píxeles, y receptores heterodinos de multi-banda, que utilizan un multiplexor para procesar señales RF en sub-bandas, siendo útiles en la observación de múltiples líneas espectrales simultáneamente, como en el caso de cuásares y nubes moleculares, entre otros casos.

2.2.3.1. Fuentes de Oscilador Local

En el rango de operación de ondas milimétricas y submilimétricas, se puede hacer uso directo de las señales generadas por sintetizadores comerciales. En el rango de pocos GHz hasta ~ 100 GHz, los osciladores locales del tipo YIG son implementados en los receptores, mientras que para rangos superiores a 100 GHz, se prefiere el uso de osciladores del tipo Gunn [4].

En ALMA, se utiliza un enfoque completamente diferente. Una señal de microondas con frecuencia estable para el rango 30-900 GHz debe ser distribuida a lo largo de más de 10 km a cada receptor. Para ello, primero se producen dos señales láser moduladas que se llevan a los receptores mediante fibras ópticas. En los receptores, las señales láser se mezclan para producir una señal de microondas, la cual finalmente se multiplica y amplifica para producir la frecuencia LO necesaria.

2.2.4. Sintetizadores

De la compañía *Micro Lambda Wireless INC*, se encuentra una variedad de sintetizadores que dependen del uso que se quiera dar del dispositivo. En forma general, los sintetizadores abarcan un rango de 50 MHz a 20 GHz y se basan en osciladores del tipo YIG, los mismos que se encuentran actualmente en el Mini. Además, se reduce considerablemente el tamaño de los dispositivos electrónicos del sistema y se entrega un ruido de fase muy bajo, debido a que cuentan con un sistema de *PLL* interno. Es por esto que son ideales para ser utilizados como el principal oscilador local en un sistema receptor, pero también puede implementarse en convertidores de frecuencia y equipos de prueba.

La Serie *MLSP* está basada en osciladores YIG y provee 1 kHz de resolución de frecuencia sobre el rango de 600 MHz a 20 GHz. Los niveles de potencia varían entre 8 y 13 dBm, con una velocidad de afinación igual a 5 mSec y cuenta con dos tipos de interfaces para su implementación: opción con conexión USB en la cual se puede utilizar el *software* propietario

y la opción de conexión serial (*five wire* en inglés) para la programación personalizada del dispositivo. En la siguiente tabla 2.4 se muestra un resumen de las principales características del dispositivo *MLSP-4016* y otros modelos que cubren frecuencias similares.

Tabla 2.4: Principales características de Sintetizadores de *Micro Lambda Wireless INC*

Modelo	MLSP-4016	MLSP-6018	MLSP-8020
Frecuencia de Salida	4 – 16 GHz	6 – 18 GHz	8 – 20 GHz
Potencia Mínima de Salida	10 dBm	10 dBm	8 dBm
Tamaño de Paso Mínimo	1 kHz	1 kHz	1 kHz
Variación de Potencia sobre Freq/Temp	5 dB	6 dB	6 dB
Armónicos	-12 dBc	-12 dBc	-12 dBc

Todas las series de sintetizadores cuentan con diversas opciones para la operación del oscilador local y dependen de las referencias externas e internas. En la opción A, se tiene una referencia externa que varía entre 1 – 200 MHz pero no existe referencia interna. En la opción B, se tienen ambas referencias y se opera entre y 1 – 200 MHz. Finalmente, en la opción C ocurre lo contrario a la opción A y se opera con referencia interna a 100 MHz. Estas tres configuraciones son las principales funciones de oscilador pero, aún así, existen otras diferentes que pueden ser encontradas en la guía de funcionamiento del dispositivo (ver Anexo B).

En el presente caso, se utiliza el modelo *MLSP-4016* ya que es compatible con la nueva configuración que se desea entregar al receptor del Mini. En particular, se cuenta con la opción de configuración B, que indica que necesita ser operado con referencia externa e interna, más la opción D, el cual señala que los conectores del sintetizador se encuentran en la parte frontal. Además, existen otros ejemplares que pueden ser utilizados en otras aplicaciones. Dentro de estos, se pueden encontrar los modelos *MLMS*, *MLVS* y *ML5G*, siendo todos útiles para la incorporación en sistemas de microondas y radiofrecuencias.

Capítulo 3

Diseño e Implementación

3.1. Metodología

En el presente trabajo se desarrolló, probó e implementó el sistema de control y, además, se diseñó e implementó el sistema de medición de potencia para el oscilador local para el sistema de recepción del *Southern Millimeter Wave Telescope*. Para esto, por el lado de radiofrecuencias, se trabajó con un diseño de *hardware* preestablecido con los componentes que se muestran en la tabla 3.1. Para la parte de control, se diseñó e implementó un sistema que fuese compatible con el diseño de *hardware* mencionado anteriormente. Los elementos del sistema de control son mostrados en la tabla 3.1.

Tabla 3.1: Equipos de radiofrecuencias y sistema de control utilizados.

Dispositivos de Radiofrecuencia	Dispositivos del Sistema de Control
<i>Sintetizador MLSP-4016</i>	<i>Raspberry Pi</i>
Multiplicador	Placa de Control <i>RPI</i> - LO
Atenuador Variable	Módulo ADS1115
Acoplador Direccional	Convertor de Nivel
Detector de Potencia de Banda Ancha UD	Relé

También, es importante mencionar los equipos disponibles en el Laboratorio de Ondas Milimétricas, que fueron utilizados para las pruebas realizadas, tanto con el sistema de control de manera independiente, como con el oscilador local. En la siguiente tabla 3.2 se muestran los equipos utilizados.

Tabla 3.2: Equipos disponibles en el Laboratorio utilizados durante las pruebas.

Equipo de Laboratorio	Modelo	Especificaciones
Analizador de Señales CXA	N9000A	9 kHz - 26.5 GHz
Generador de Señales Analógicas	E8257D	250 kHz - 20 GHz
Analizador de Señales PXA	N9030A	3 Hz - 50 GHz
Fuente de Alimentación DC Programable	DP932A	-
Generador de Formas de Onda Arbitrario	33220A	20 MHz
Módulo Extensor de Banda V	M1970V	50 – 75 GHz

Además, los equipos presentes en el *Southern Millimeter Wave Telescope*, fueron fundamentales para el análisis y entendimiento del funcionamiento actual del oscilador local. Así, se logró comprender el flujo de operación que actualmente se ejecuta, en términos de *software* y *hardware*, del sistema de recepción y en particular, del oscilador local. El radiotelescopio y algunos de los equipos que se utilizan para su operación, son mostrados en la figura 3.1.



(a) Radiotelescopio Mini (b) Equipos del SMWT

Figura 3.1: Principales equipos del SMWT.

De esta forma, a continuación, se muestra la estructura de la metodología que fue seguida a lo largo del trabajo realizado. En el diagrama que se observa en la figura 3.2, se establece una serie de etapas, en las que se resume y se describe el flujo de trabajo.

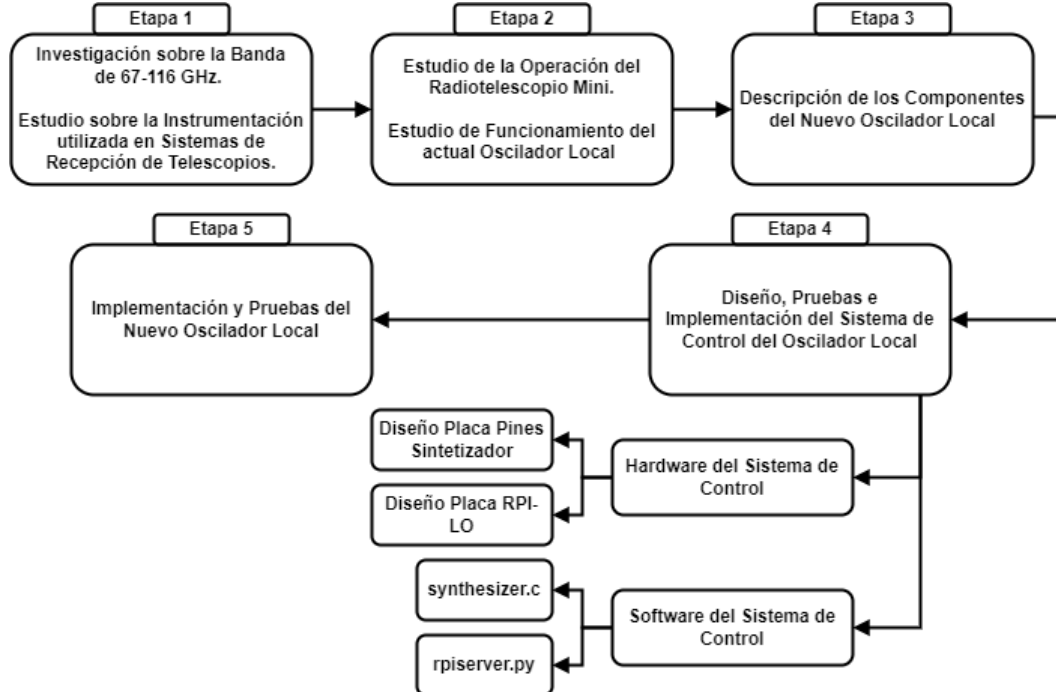


Figura 3.2: Flujo de trabajo para las etapas establecidas en la metodología seguida.

Con esto, se entrega una descripción sobre el desarrollo del trabajo y con ello, de las

labores realizadas. A continuación, se describen las etapas mencionadas en la figura 3.2.

i) **Etapa 1: Contextualización Astronómica para la banda de 67-116 GHz.**

Dentro de la primera etapa, se realizó una breve investigación de la nueva banda en la que operará el *SMWT*, más la instrumentación que es utilizada en radiotelescopios. Con ello, se acotó el problema y se enfatizó en las tecnologías que puedan cumplir los requisitos de la nueva banda.

ii) **Etapa 2: Estudio del Funcionamiento Actual del Radiotelescopio.**

En esta etapa, se realizó un estudio de cada uno de los subsistemas del *SMWT*, con principal énfasis en el bloque de recepción y en particular, del oscilador local. Con esto, no solamente se logró comprender el funcionamiento actual de los equipos que son utilizados para la operación del radiotelescopio, sino que también se estudiaron a profundidad los conceptos que existen detrás de los mecanismos.

iii) **Etapa 3: Descripción de los Nuevos Componentes del Oscilador Local.**

Durante esta etapa, se trabajó en la descripción de los dispositivos de radiofrecuencias y del sistema de control. Para esto, se realizó una revisión de la hoja de datos de cada uno de los equipos utilizados, de donde se extrajo la información de interés para el diseño del sistema de control. Con esto, se identifican las tres principales funciones que el nuevo sistema de control del oscilador local debe cumplir:

- Generar una señal de control que indique el enclavamiento de la señal generada por el oscilador local.
- Monitorizar el estado de los componentes. En específico, se debe medir la potencia de salida del sistema.
- Establecer la comunicación con los dispositivos que componen el oscilador local, de tal modo que se pueda definir la frecuencia de la señal que debe ser generada.

iv) **Etapa 4: Diseño, Pruebas e Implementación del Sistema de Control del Oscilador Local.**

A lo largo de esta etapa, el trabajo que fue realizado se dividió en dos. El componente de *hardware*, que también es dividido en dos, está conformado por la placa adaptadora de pines, que es utilizada para la comunicación y alimentación del sintetizador, más la placa de control *Raspberry Pi-LO*, en donde se integran los dispositivos del sistema de control. El segundo componente, de *software*, también se divide en dos: el programa *synthesizer*, que se encarga de manejar la comunicación y monitoreo del sintetizador *MLSP-4016*, y el programa *rpiserver.py* que ejecuta el sistema de control.

v) **Etapa 5: Implementación y Pruebas del Nuevo Oscilador Local.**

En la última etapa, se realizó la integración del sistema de control realizado en la etapa anterior, con el resto de los componentes de radiofrecuencias. Con esto, se buscó la verificación conjunta del funcionamiento de la totalidad del sistema y, con ello, la solución de detalles que fueron corregidos. Cabe destacar que, por motivos que serán mencionados posteriormente, los resultados de la implementación son mostrados para dos configuraciones de medición diferentes: para la banda de 36 – 50 GHz y para 50 – 60 GHz.

3.2. Diseño del Oscilador Local y su Sistema de Control

El nuevo oscilador local que será implementado dentro del radiotelescopio Mini, presenta grandes diferencias con respecto a su versión anterior. El nuevo diagrama para el oscilador local se muestra en la siguiente figura 3.3.

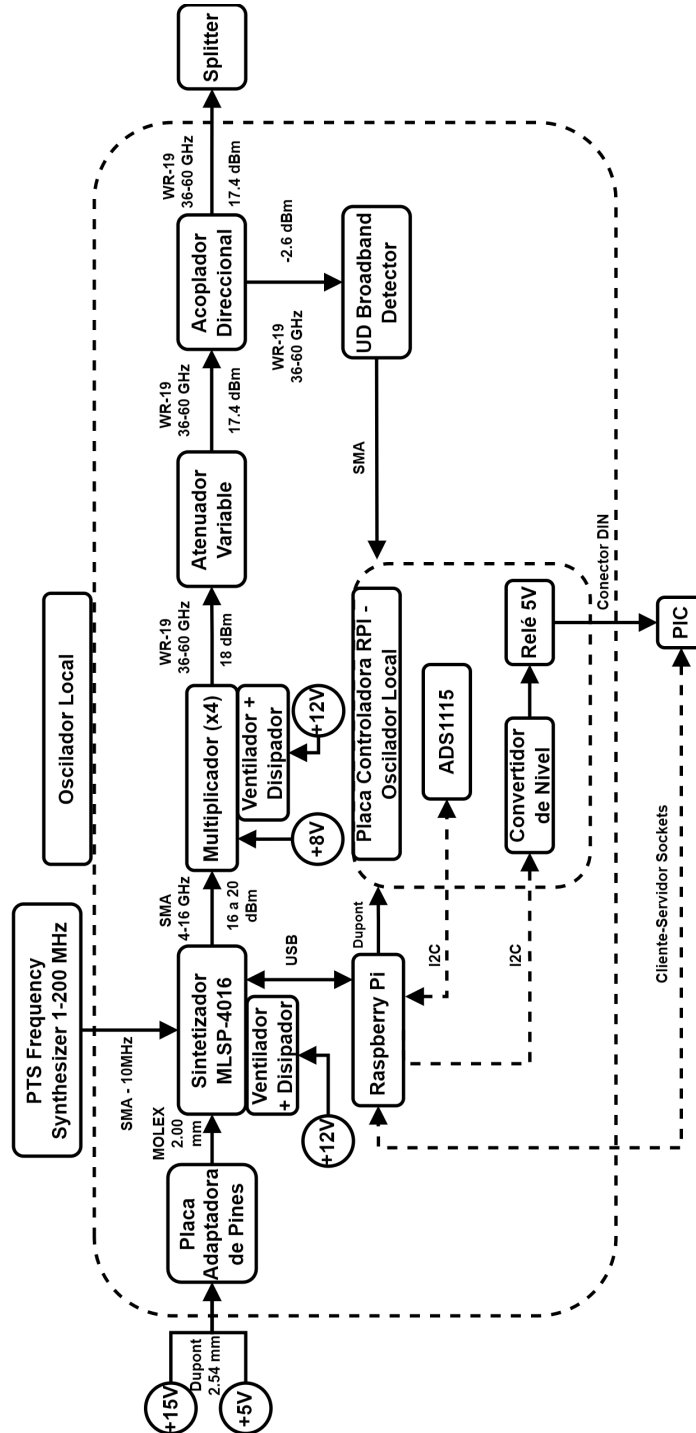


Figura 3.3: Diagrama del nuevo oscilador local.

De la figura 3.3, se observa que el sintetizador, que es alimentado por la Placa Adaptadora de Pines, cumple la función de generar una señal entre 4 y 16 GHz y que, además, utiliza una referencia externa de 10 MHz, pero es posible utilizar una señal dentro del intervalo de 1 – 100 MHz, con potencia de entrada de 0 ± 3 dBm. Posteriormente, la señal es multiplicada por cuatro, luego es atenuada y pasada hacia el acoplador direccional. En este último, la señal multiplicada pasa a la siguiente etapa fuera del oscilador local pero, además, se toma una muestra para la medición de su potencia. Asimismo, la lectura del valor de la potencia es obtenida por el sistema de control, el cual, además, está encargado de la comunicación y sintonización de la señal que es generada con el sintetizador.

En el presente capítulo se describe el diseño, en conjunto con las especificaciones, de los componentes tanto del oscilador local como el de su sistema de control. Además, se muestra la implementación de ambos elementos. El diseño del oscilador local se ve reflejado en el diagrama de la figura 3.3, por lo que es sumamente relevante mencionar los componentes de este. En consecuencia, se pueden realizar dos clasificaciones: componentes de radiofrecuencias y del sistema de control.

3.2.1. Componentes de Radiofrecuencias

3.2.1.1. Sintetizador

El sintetizador utilizado corresponde al modelo *MLSP-4016*, de la compañía *Micro Lambda Wireless Inc.* El dispositivo cuenta con cuatro tipos de conectores, como se puede ver en la siguiente figura 3.4. En ella, se muestra un diagrama que explica el funcionamiento de los conectores dentro del sintetizador. *J1* posee la entrada de alimentación de 5 V y 15 V, además de proveer la comunicación serial mediante el protocolo de cinco cables, o *five wire* en inglés. En paralelo, *J2* corresponde a la conexión *USB* con la que cuenta el sintetizador, *J3* es la señal de referencia externa y, finalmente, *J4* es la señal de salida de dispositivo. Como se puede observar en la misma figura 3.4, para controlar el sintetizador, se acepta la comunicación tanto con *USB* como con *five wire*. En el presente caso, se utilizó la opción *USB* para diseñar la interfaz de línea de comandos.

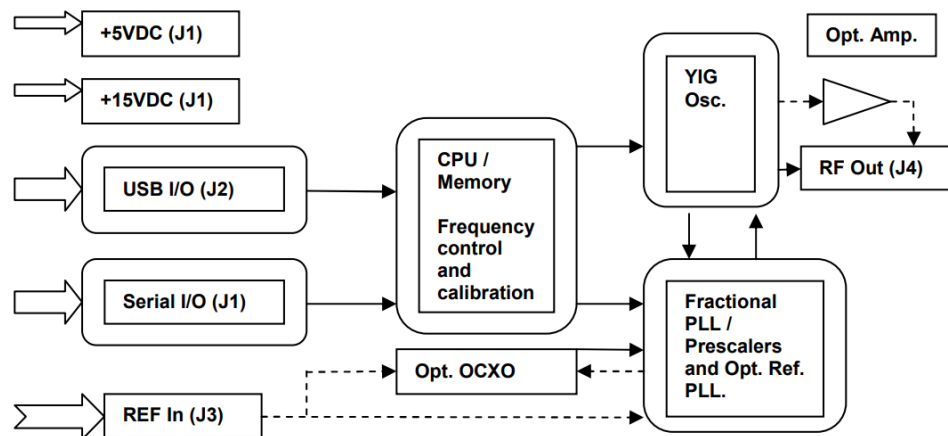


Figura 3.4: Diagrama de funcionamiento de conectores dentro del sintetizador [30].

Por el lado de *software*, la interfaz *USB* brinda al usuario un programa propietario que

puede ser utilizado en dispositivos *Windows*. El sintetizador puede ser controlado de manera remota a través de la siguiente interfaz de configuración, mostrada en la figura 3.5. Sin embargo, es importante señalar que no se cuenta con dispositivos *Windows* dentro del radiotelescopio y, además, el sistema de control, regido por un *Raspberry Pi*, tampoco cuenta con este sistema operativo. En consecuencia, este programa será únicamente utilizado como referencia de la información básica que entrega el sintetizador, pero no será usado en el sistema de control.

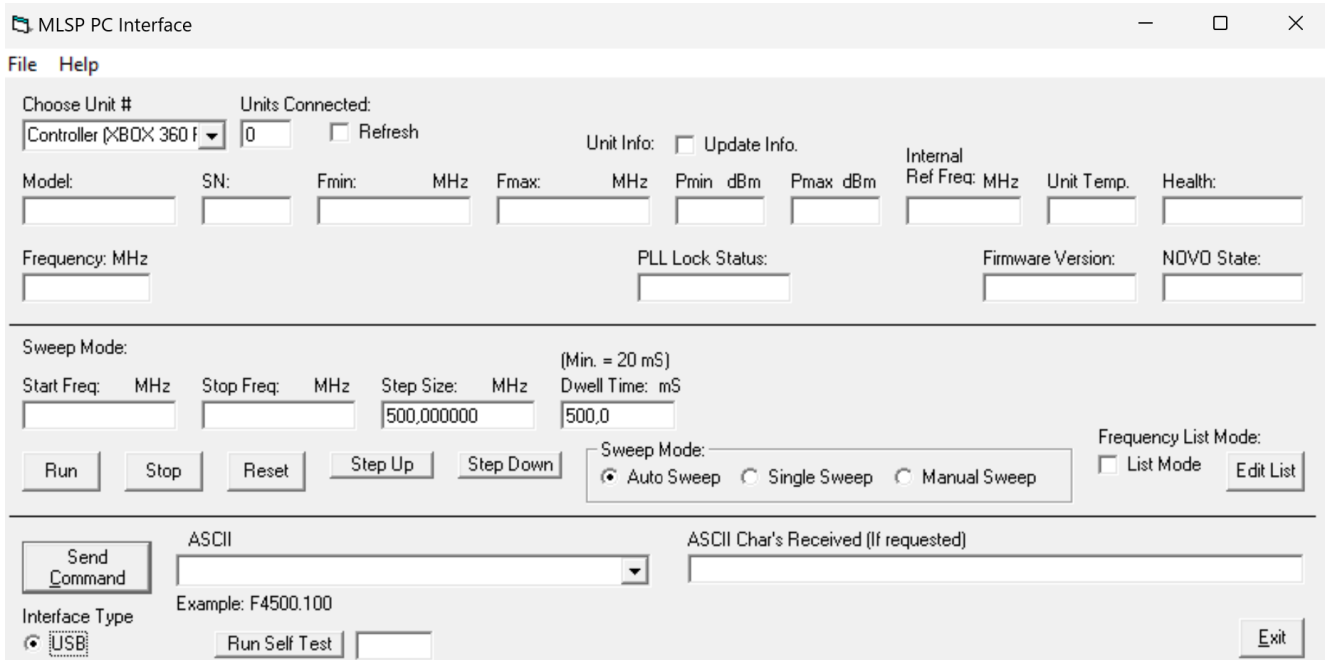


Figura 3.5: Interfaz de configuración para el sintetizador MLSP-4016.

Un método de comunicación aceptado por el sintetizador corresponde a un bus serial de 5 cables, o *5 wire serial bus*, el cual se encuentra disponible dentro del conector *J1*. Los cinco pines de control se muestran en la siguiente tabla 3.3.

Tabla 3.3: Información sobre pines de control para la programación del sintetizador.

Pin	Input/Output	Descripción
J1 - Pin 11	Input	<i>Select</i> . Habilita el desplazamiento de los datos hacia el buffer interno de comandos.
J1 - Pin 9	Input	<i>Clock</i> . La información es enviada al dispositivo en el flanco ascendente.
J1 - Pin 10	Output	<i>Data In</i> . Pin de entrada de información, que es enviada desde el <i>MSB</i> (bit más significativo).
J1 - Pin 12	Output	<i>Busy</i> . Línea que es utilizada para indicar que el sintetizador está procesando otros comandos.
J1 - Pin 14	Output	<i>Data Out</i> . Línea que es usada para enviar información interna del sintetizador al usuario.

El puerto serial de comunicación del sintetizador, de tipo *CMOS/TTL*, se asemeja al

protocolo *Serial Peripheral Interface* (SPI), utilizado para la comunicación entre microcontroladores y dispositivos periféricos [31]. Sus pines (*Clock*, *Data In/Out*, *Select*) siguen un formato similar al estándar, con una velocidad máxima de reloj de 250 kHz. El pin *Busy* actúa como monitor del estado del dispositivo, aunque puede ser omitido en la programación si hay suficiente tiempo entre comandos. Todos los comandos se envían y reciben en formato *ASCII* [32].

Por otro lado, la comunicación por *USB* es otro método aceptado por el sintetizador. El equipo es reconocido como un dispositivo de interacción humana, *HID* por sus siglas en inglés, por lo que se escogió este método de comunicación en lugar del bus serial de 5 cables. De esta forma, se diseñó una interfaz para realizar el control y comunicación con el sintetizador, cuyo objetivo fue únicamente simular el programa facilitado por el fabricante. El algoritmo implementado se denominó *synthesizer* y será descrito posteriormente.

En otros términos, las especificaciones en términos de *hardware* fueron divididas en tres clases, las cuales se mencionan a continuación.

- **Especificaciones de RF.** Los valores para los parámetros de radiofrecuencias se muestran a continuación en la tabla 3.4.

Tabla 3.4: Especificaciones de RF para el sintetizador [30].

Parámetro	Valor
Frecuencia de salida	4 – 16 GHz
Potencia mínima de salida	+10 dBm
Tamaño de paso mínimo	1 kHz
Tiempo de calentamiento	3 min.
<i>Load VSWR</i>	2.0 : 1
Ruido de fase a 100 Hz del portador	71 dBc/Hz
Ruido de fase a 1 kHz del portador	90 dBc/Hz
Ruido de fase a 10 kHz del portador	88 dBc/Hz
Ruido de fase a 100 kHz del portador	115 dBc/Hz
Ruido de fase a 1 MHz del portador	138 dBc/Hz

- **Especificaciones Eléctricas.** En la tabla 3.5, se detallan las condiciones de operación que debe cumplir, eléctricamente, el sintetizador para que pueda ser utilizado de manera correcta.

Tabla 3.5: Especificaciones eléctricas para el sintetizador [30].

Parámetro	Valor
Voltaje de Operación	$15V_{DC} \pm 0.5$ y $5V_{DC} \pm 0.25$
Corriente de Operación	1500 [mA] y 300 [mA]
Potencia Disipada	23 [W]
Temperatura	0° a +60° C

- **Especificaciones Mecánicas.** Finalmente, los aspectos mecánicos del sintetizador se muestran en la siguiente tabla 3.6.

Tabla 3.6: Especificaciones mecánicas para el sintetizador [30].

Parámetro	Valor
Peso	426g
Conector J1 - Macho	MOLEX # 35363 - 1460
Conector J1 - Hembra	MOLEX # 35507 - 1400
Conector J1 - <i>Crimp Pin</i>	MOLEX # 50212 - 8000
Conector J1 - Cable	A.W.G. 22-24
Conector J2 - Cable	USB Mini-B

Dentro de las condiciones de operación del sintetizador, se establece que las variables de temperatura y bits de estado son críticas y, por ende, se tiene la necesidad de conocer sus valores. Para esto, se define la siguiente tabla 3.7, donde se muestra la notación utilizada y los rangos de operación para cada una.

Tabla 3.7: Condiciones de operación del sintetizador.

	Notación	Rango de Operación
Temperatura	T	$0^{\circ}C \leq T \leq 60^{\circ}C$
Bits de Estado	$D_0 D_1 D_2 D_3 D_4 D_5 D_6 D_7$	11000011

Las condiciones de operación definidas para el sintetizador, se basan en que la temperatura se mantenga dentro del rango establecido y que los bits de estado sean iguales a los mostrados en la tabla 3.7. El significado de cada bit se muestra en la tabla 3.8.

Tabla 3.8: Significado de cada bit de estado del sintetizador.

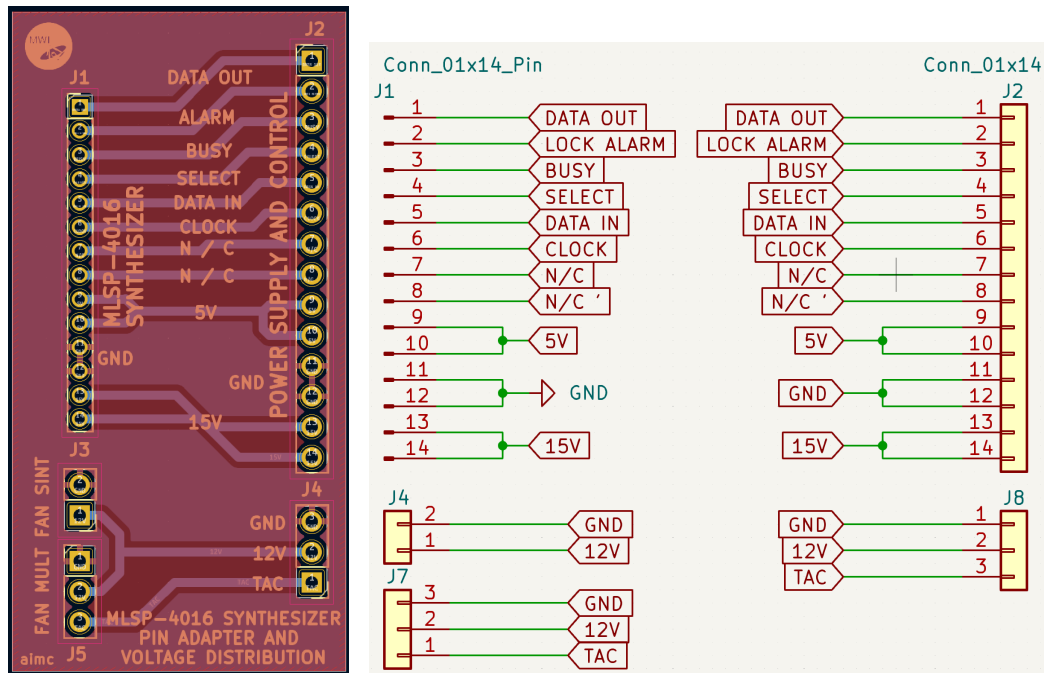
Número de Bit	Descripción	Valor
D_0	Enclavamiento 100 MHz	1 = Enclavado, 0 = Desenclavado
D_1	Enclavamiento del PLL de YIG	1 = Enclavado, 0 = Desenclavado
D_2	No ocupado	No ocupado
D_3	No ocupado	No ocupado
D_4	No ocupado	No ocupado
D_5	No ocupado	No ocupado
D_6	Estado Sintetizador	1 = Bien, 0 = Mal
D_7	Estado NOVO	1 = Enclavado, 0 = Desenclavado

- **Placa Adaptadora de Pines**

Para poner en marcha el sintetizador y comprobar su correcto funcionamiento, se diseñó e imprimió una placa personalizada, denominada “Placa Adaptadora de Pines”, la cual puede cambiar el paso de distintos conectores. El conector que es utilizado en el sintetizador corresponde al MOLEX #35363-1460, el cual cuenta con una separación entre

sus pines, también denominada como paso, de 2 mm. Como este tipo de conectores no son de fácil acceso, puesto que el estándar corresponde a un paso de 2.54 mm, se utilizó la placa diseñada para satisfacer los requerimientos de conectividad.

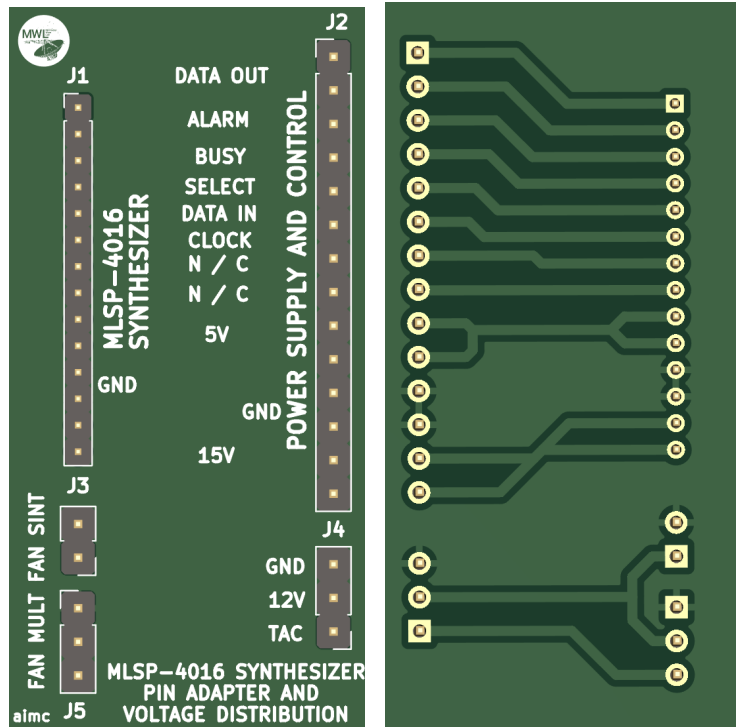
En el esquemático de la placa, se realiza la conexión desde el lado izquierdo para el conector del sintetizador y en el lado derecho, se dejan los pines estandarizados libres para que se puedan conectar libremente los pines que sean necesarios. Por otro lado, en los conectores *J3*, *J4* y *J5*, se añadió el espacio necesario para la conexión del ventilador y disipador de calor del sintetizador y del multiplicador. En las figuras 3.6.a y 3.6.b se muestra el diseño en PCB y esquemático de la placa adaptadora de pines, respectivamente.



(a) Diseño en PCB de la placa adaptadora de pines. (b) Diseño esquemático de la placa adaptadora de pines.

Figura 3.6: Diseño en PCB (izq.) y esquemático (der.) de la placa adaptadora de pines.

En las figuras 3.7.a y 3.7.b se muestra el diseño de la placa vista desde la parte frontal y trasera, respectivamente. En las imágenes se puede apreciar la diferencia de tamaño entre los conectores de distinto paso y donde, además, se identifica el dispositivo que debe ser conectado en cada sección de la placa.



(a) Vista frontal de la placa adaptadora de pines. (b) Vista trasera de la placa adaptadora de pines.

Figura 3.7: Vista frontal (izq.) y trasera (der.) de la placa adaptadora de pines.

Debido al factor de *software*, solo fue necesario utilizar los pines de voltaje de alimentación de 5 V y 15 V más las conexiones a tierra. Sin embargo, también se dejó a libre disposición los pines de control en caso de que fuese necesario implementar la comunicación mediante el bus serial de 5 cables.

• Interfaz de Control para Sintetizador

El sintetizador *MLSP-4016* es un dispositivo de interfaz humana: recibe entradas por parte de algún usuario, realiza cierto proceso definido y entrega salidas al mismo usuario. En la figura 3.8, se muestra la terminal del *Raspberry Pi* y, en él, el reconocimiento del sintetizador como dispositivo *HID*. En la segunda línea, el número *04d8* representa el identificador del vendedor, mientras que *003f* corresponde al identificador del producto.

```
pi@raspberrypi:~/instmini/lab_tests/comms $ lsusb
Bus 002 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 001 Device 016: ID 04d8:003f Microchip Technology, Inc. MLSP2208
Bus 001 Device 002: ID 2109:3431 VIA Labs, Inc. Hub
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
```

Figura 3.8: Reconocimiento por parte del *Raspberry Pi*, del sintetizador como *HID*.

El programa diseñado, *synthesizer*, hace uso de la librería *HIDAPI* [33] para comunicarse con el sintetizador, la cual es compatible con el sistema operativo del *Raspberry*

Pi: Linux. Además, el objetivo de *synthesizer* consiste únicamente en servir como una herramienta de control y monitoreo del sintetizador. También, es independiente de otros dispositivos y programas, por lo que presenta bastante utilidad para resolver posibles problemas sin entorpecer diferentes sistemas. Se buscó que el *software* fuera simple y que no ocupase grandes recursos de la memoria del *Raspberry Pi*.

El diseño se realizó de tal forma que cumpliera un rol similar al *software* del fabricante, de donde se rescataron dos principales funcionamientos. El primero, muestra en pantalla las principales propiedades del sintetizador: temperatura, estado de la referencia externa, interna y del *PLL*, auto pruebas que se pueden realizar internamente, como también voltajes que operan dentro del dispositivo. En el segundo, el programa está escuchando constantemente al usuario por el posible ingreso de comandos, en caso de que este desee consultar alguna información en particular. En particular, se puede definir la frecuencia de la señal que genera el sintetizador cuando el usuario ingrese el comando *fxxxx.xxxxx*.

El diagrama de flujo del funcionamiento completo del programa *synthesizer* se muestra en la figura 3.9, la cual puede ser explicada en cuatro etapas. Dentro de la primera, se realizan todas las instrucciones y verificaciones de conectividad con librerías y el sintetizador. En caso de que no se encuentren tales herramientas, el programa muestra el respectivo mensaje de error y termina el programa. En la segunda etapa, se espera hasta que el usuario ingrese alguno de los comandos disponibles dentro del manual de operación (ver Anexo B). Cuando lo realiza, se verifica si este corresponde al comando de salida y, en caso de coincidir, se pasa a la fase de cierre. Si no corresponde a la salida, y siempre que el comando exista, se procede a enviar el mensaje, esperar la respuesta del sintetizador y cuando esta exista, se muestra el resultado en la consola. En paralelo a este proceso, se ejecuta la tercera etapa, en donde se consulta de manera periódica el valor de la temperatura y de los bits de estado del sintetizador.

Durante las etapas anteriores, siempre se verifica el caso en que el usuario cierre el programa a través del comando de salida. Cuando esto ocurre, se pasa a la cuarta y última etapa en la cual se procede a la rutina de cierre. De esta forma, se envían de manera automática los comandos que aparecen en la tabla 3.9 con el objetivo de enseñar el último estado del sintetizador antes de cerrar la conexión.

Es importante mencionar que el programa *synthesizer.c* es utilizado únicamente para controlar el sintetizador de manera independiente. Esto quiere decir que, si se conecta el dispositivo al *Raspberry Pi*, es posible verificar su estado y control mediante el envío de los comandos disponibles en el manual de operación (ver Anexo B). El programa no forma parte del sistema de control del oscilador local, pero fue elaborado con el objetivo de simular la interfaz que provee el fabricante.

En síntesis, la interfaz de línea de comandos fue diseñada con el objetivo de establecer un vínculo de comunicación bidireccional con el sintetizador. El programa busca mantener la simpleza e interacción con el usuario, además de cumplir las funcionalidades más importantes que realiza el *software* del fabricante: envío de comandos, recepción de

respuestas y monitoreo del estado del sintetizador. A continuación se procede a comentar cada una de ellas.

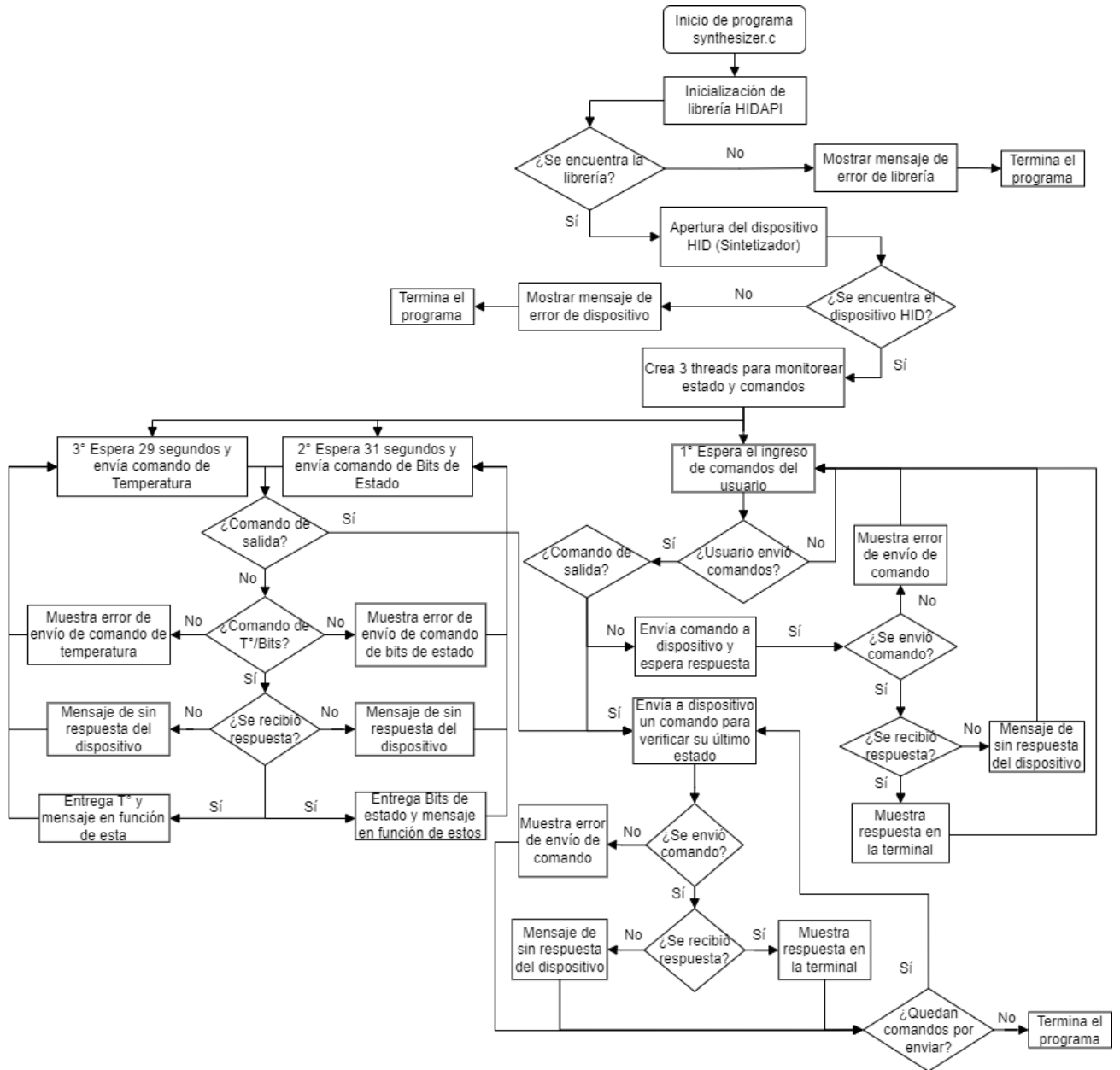


Figura 3.9: Diagrama de flujo de *synthesizer.c*.

(i) Monitoreo del estado del sintetizador

Dentro de las características que posee el sintetizador, se escogieron para ser monitoreadas periódicamente aquellas que puedan causar posibles fallos en el funcionamiento del dispositivo. En consecuencia, de todo el set de comandos disponibles dentro de la hoja de datos (ver Anexo B), se escogieron los que se muestran en la siguiente tabla 3.9:

Tabla 3.9: Información del sintetizador que será monitoreada.

Comando	Descripción
T	Temperatura
?	Bits de estados
V1	Voltaje del PLL
V2	Voltaje de Ref. Ext.
V3	Voltaje interno (+2.5 V)
V4	Voltaje interno (+3.3 V)
V5	Voltaje interno (+5.0 V)
V6	Voltaje interno (+6.75 V)
V7	Voltaje interno (+12.0 V)
V8	Voltaje interno (+13.5 V)
V9	Voltaje interno (−5.0 V)
R0010	Máxima temperatura alcanzada
R0013	Estado de la unidad
R0014	Estado de calibración
R0015	Auto prueba

De los comandos mencionados en la tabla 3.9, se definió que cada 29 y 31 segundos, se iban a mostrar en pantalla los valores para la temperatura y bits de estado, respectivamente, en conjunto con un mensaje personalizado que va en función de la respuesta del instrumento. Esto quiere decir que durante la ejecución del programa, cuando se esté esperando a que se ingresen nuevos comandos, se entregará la información de temperatura y bits de estado.

En la figura 3.10 se muestra una prueba realizada del funcionamiento del programa. Dentro de la terminal del *Raspberry Pi*, se ejecuta el *script* que contiene el programa y este inicia con la apertura del dispositivo *HID* e inmediatamente se ingresa el comando de la frecuencia de la señal que debe generar. Al instante, la terminal queda disponible para que el usuario envíe comandos en específico, mientras que de manera paralela, se inicia el proceso de monitoreo del estado del sintetizador.

El proceso envía de manera automática el comando de temperatura y espera hasta que el sintetizador responda, lo cual se produce sin interferir el posible ingreso de comandos. Luego de que se recibe la información, esta se muestra en la consola en conjunto con un mensaje que está en función del valor de la temperatura. Esto quiere decir que el mensaje dependerá si el dispositivo tiene altas o bajas temperaturas. Dos segundos después, el proceso se repite pero para el comando de bits de estado y también se muestra una nota con información detallada del significado de cada bit. El procedimiento sigue hasta que se envía el comando de salida “exit”. En resumen, se entregarán los datos de temperatura y bits de estado cada 29 y 31 segundos, respectivamente.

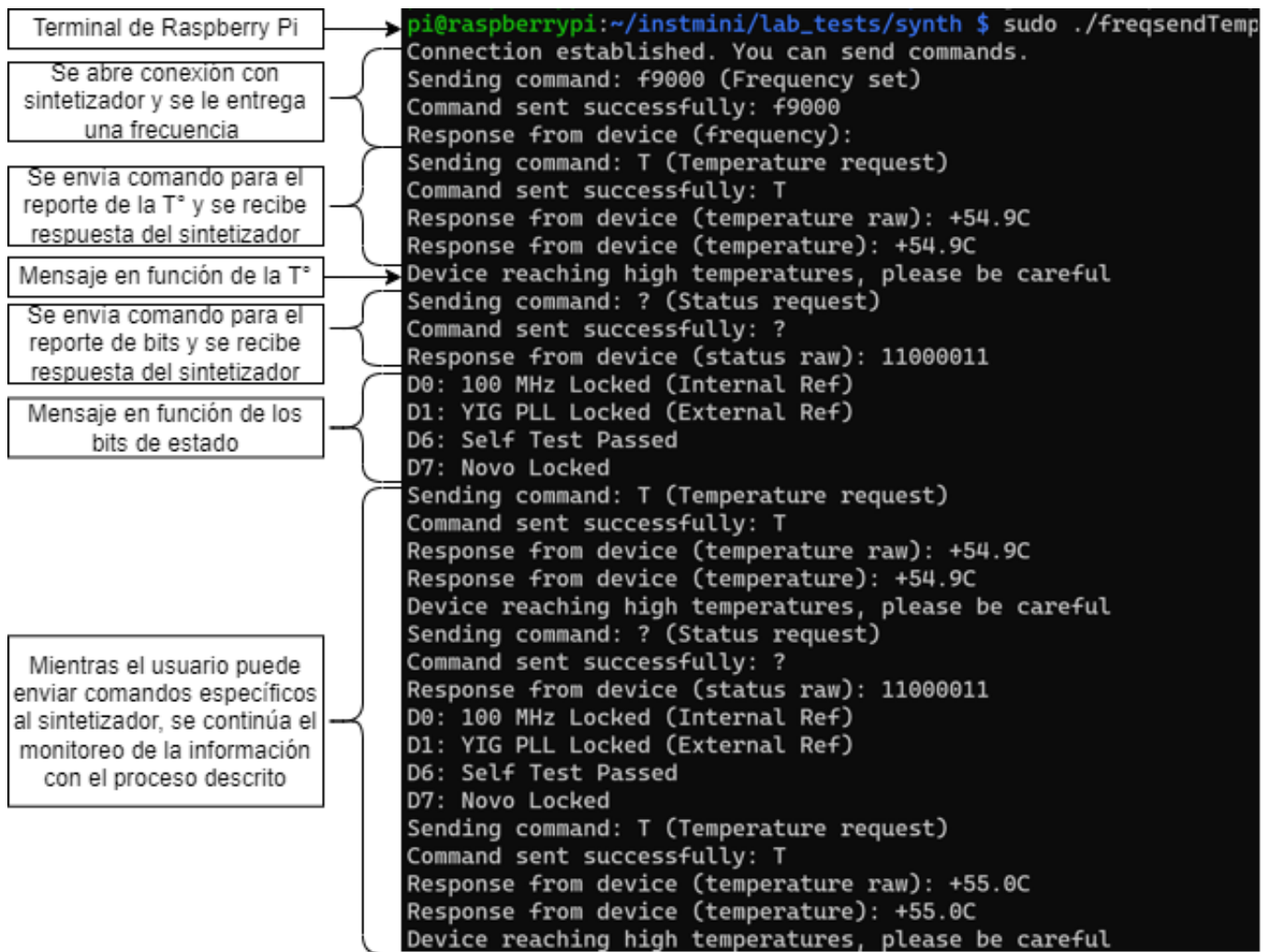


Figura 3.10: Monitoreo de temperatura y bits de estado en *synthesizer.c*.

Es importante mencionar que las variables monitoreadas pueden ser modificadas en función de lo que desee el usuario. Esto quiere decir que se pueden añadir otros comandos, o eliminar alguno de los escogidos. De la misma forma, el tiempo con el que se entregan tales datos puede aumentar o disminuir en base a los intereses del operador.

(ii) **Envío de comandos y recepción de respuestas**

La segunda funcionalidad que se implementó es sumamente relevante ya que con esta se logra la comunicación bidireccional con el sintetizador. De este modo, en la terminal del *Raspberry Pi*, el usuario puede consultar por información en específico del instrumento ingresando el respectivo comando. Asimismo, el operador puede definir de manera manual la frecuencia de la señal que desee generar.

Mientras se espera que se ingresen comandos, el monitoreo de la temperatura y bits de estado van a seguir ejecutándose de manera paralela y sin intervenir mutuamente. De esta forma, se logra simular el comportamiento de la interfaz de control proveniente del fabricante.

En consecuencia, el programa está disponible y a la espera del ingreso de comandos por parte del operador. En caso de que un mensaje sea enviado, se revisará si es que es el comando de salida “exit”, o si procede a enviarlo al sintetizador. Se verifican los casos de que el envío del mensaje o la recepción de la respuesta haya fallado y, en caso contrario, se muestra la respuesta en la consola del *Raspberry Pi*. Finalmente, con el objetivo de desgeneralizar las respuestas ante los comandos y de entregar mayor información, se han personalizado algunos mensajes que se enseñan en consola, como los de temperatura, frecuencia y bits de estado. Un ejemplo en el que se ilustra únicamente el envío de comandos y recepción de respuestas, se muestra en la siguiente figura 3.11.

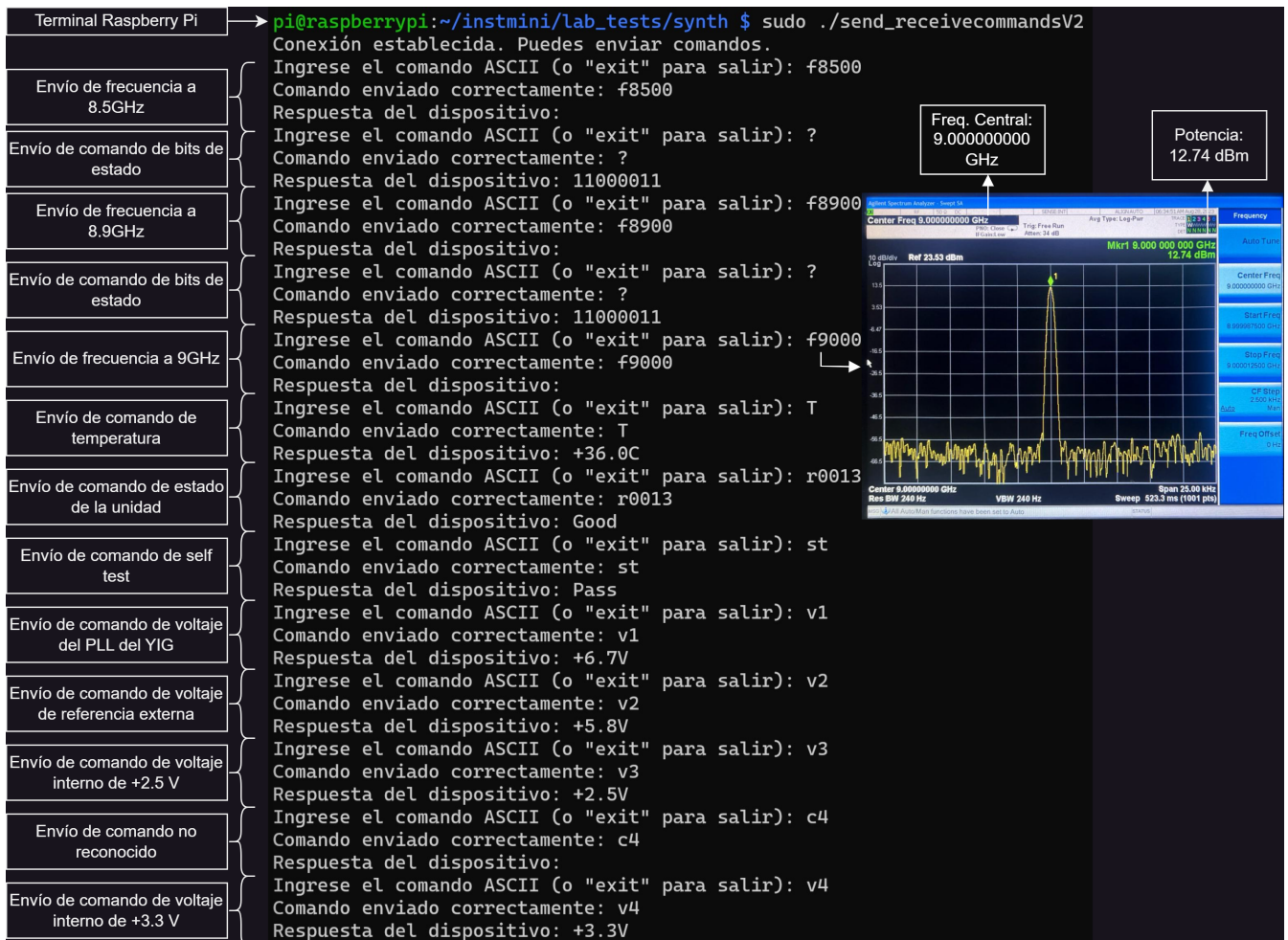


Figura 3.11: Demostración de envío de comandos y recepción de respuestas en *synthesizer.c*.

En la figura 3.11, se muestra la terminal del *Raspberry Pi* y la ejecución del programa. Se sigue con la apertura del dispositivo *HID* y, posteriormente, comienza el envío de los comandos, con lo que se logra la correcta recepción de los mensajes enviados por el sintetizador. Se comienza con el ingreso de una señal de frecuencia de 8.5 GHz y se procede con el comando de bits de estado “?”. Para el primer caso, es normal que no exista respuesta ya que el sintetizador no envía nada, mientras que en el segundo, se reciben correctamente los bits “11000011”. Se vuelve a

repetir el proceso, pero esta vez se define la frecuencia en 9.0 GHz y se logra visualizar el espectro de la señal generada en el dispositivo *PXA Signal Analyzer*. En la figura 3.11, se observa que la señal producida efectivamente tiene una frecuencia de 9 GHz, con 9 decimales de precisión y que, además, su potencia es de 12.74 dBm.

El proceso continúa con el envío de otro tipo de comandos con el fin de probar el comportamiento ante distintos mensajes. El procedimiento sigue hasta que se envía el comando de salida “exit”.

3.2.1.2. Multiplicador

Para el nuevo oscilador local que será operado en el Mini, se requiere de un multiplicador que cumpla las especificaciones y sea compatible con los demás dispositivos de radiofrecuencias. En este caso, la salida del sintetizador, que entrega entre 4 – 16 GHz, será la entrada al multiplicador que multiplicará cuatro veces la frecuencia del instrumento, teniendo como salida el rango de 36 – 60 GHz. Las especificaciones del multiplicador se muestran en la siguiente tabla 3.10.

Tabla 3.10: Especificaciones del multiplicador X4.

Modelo	SFA-363603418-19Sf-S1
Fabricante	<i>Sage Millimeter</i>
Frecuencia de entrada	9 – 15 GHz
Frecuencia de salida	36 – 60 GHz
Potencia de entrada	0 – 20 dBm
Potencia típica de salida	+18 dBm
Voltaje de operación	+8 V_{DC}
Corriente de operación	800 mA
Temperatura de operación	0 – 50 °C

En la siguiente figura 3.12 se muestra el multiplicador X4 que fue escogido.



Figura 3.12: Multiplicador X4 activo de frecuencia.

Para la prueba del funcionamiento del multiplicador, a su entrada se necesitó de un equipo que fuese capaz de generar una señal con frecuencia entre 9 GHz y 15 GHz, mientras que en su salida se compruebe que la señal ha sido multiplicada por el respectivo factor. El resultado de la señal multiplicada varía entre 36 y 60 GHz, pero como en el laboratorio no se cuenta

con algún equipo que sea capaz de analizar tal intervalo de manera simultánea, se realizaron las pruebas en dos bandas: para 36 – 50 GHz y 50 – 60 GHz.

(i) **Banda 36-50 GHz**

Para la primera banda, se utilizó el dispositivo generador de señales analógicas (250 kHz - 20 GHz) *Agilent E8257D* para la señal de entrada del intervalo de 9 – 12.5 GHz. Posterior a este, se encuentra el multiplicador, el cual entrega la señal con frecuencia en el intervalo de 36 – 50 GHz. Este resultado se analizó con el dispositivo analizador de señales (3 Hz - 50 GHz) *Agilent PXA*. En la siguiente figura 3.14, se muestra un diagrama con la configuración de mediciones utilizada para la medición realizada.

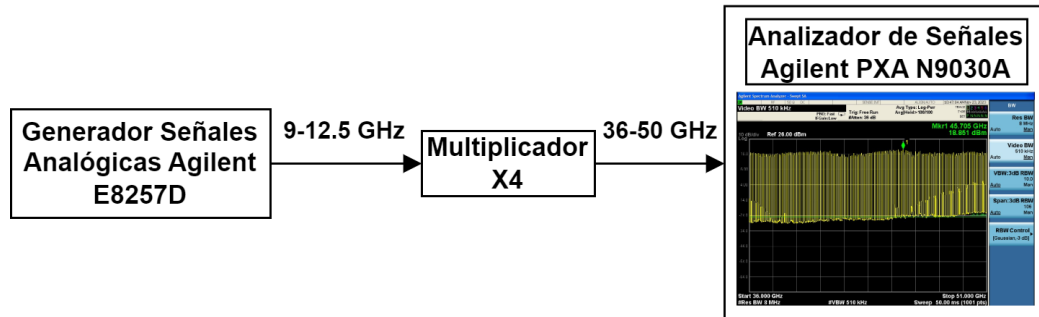


Figura 3.13: Configuración de mediciones del multiplicador para la banda de 36 – 50 GHz.

El resultado de la medición se muestra en la figura 3.14, en donde se observa un barrido completo con el intervalo inicial de 9 – 12.5 GHz y se puede notar que el rango de observación corresponde a 36 – 51 GHz. En la figura 3.14, cada máximo representa la señal multiplicada con cierta frecuencia y con potencia aproximada de salida de 19 dBm.

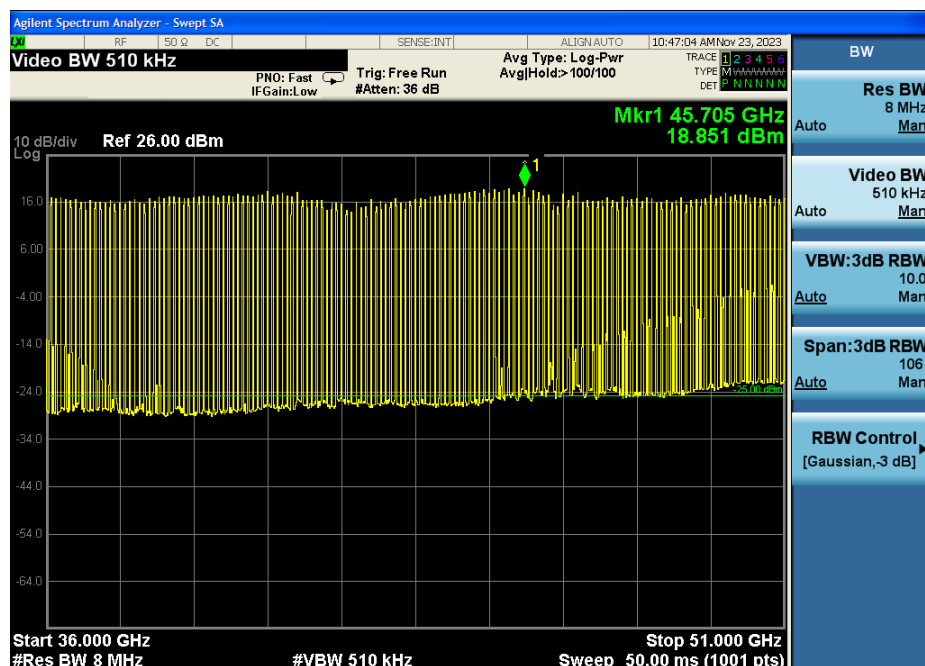


Figura 3.14: Resultado de la multiplicación en la banda de 36 – 50 GHz.

(ii) Banda 50-60 GHz

Para la segunda banda, se tuvo que trabajar con otra configuración de mediciones debido a la necesidad de análisis de una señal para un intervalo diferente. El inicio se mantiene intacto con respecto a lo mencionado anteriormente. Sin embargo, para que se observe correctamente el resultado de la multiplicación en el analizador de señales, se conectó luego del multiplicador, un acoplador direccional para disminuir la potencia (en 20 dB). Posteriormente, se utilizó un mezclador armónico para guía de onda *Agilent M1970V* que extiende la banda hasta 75 GHz. Con estos dos cambios, se cumplen las especificaciones que permiten la conexión al mismo analizador de señales *Agilent*. En la siguiente figura 3.15, se muestra la configuración de mediciones utilizada.

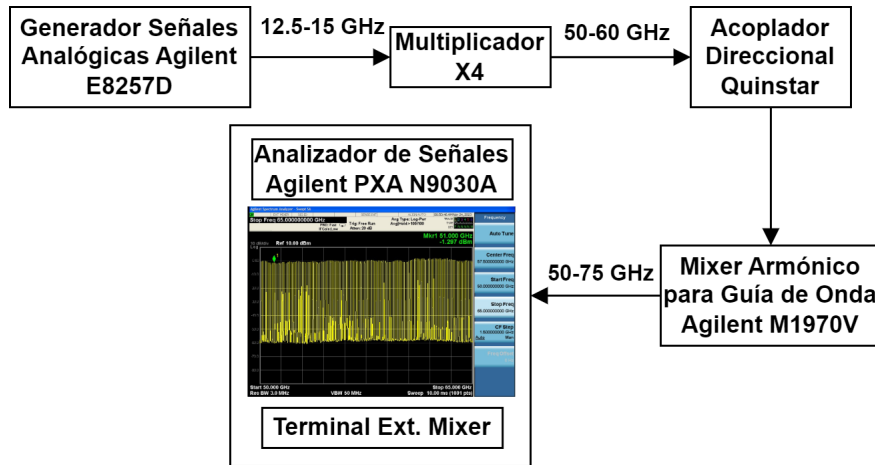


Figura 3.15: Configuración de mediciones del multiplicador para la banda de 50 – 60 GHz.

El resultado de la medición se muestra en la figura 3.16, en donde se observa un barrido completo con el intervalo inicial de 12.5 – 15 GHz y se puede notar que el rango de observación corresponde a 50 – 65 GHz. En la figura 3.16, cada máximo representa la señal multiplicada con cierta frecuencia y potencia, cuya información se puede ver en color verde. Con esto, se verifica el funcionamiento tanto del multiplicador como del acoplador direccional, puesto que este último corresponde al seleccionado para la implementación del oscilador local.

En la figura 3.16, se puede observar que la potencia de salida del multiplicador para una señal con frecuencia igual a 51 GHz, es de -1.297 dBm. Sin embargo, se debe recordar de que esta señal está atenuada en 20 dB por el acoplador direccional. De esta manera, la potencia de esta señal, sin la atenuación, equivale a 18.703 dBm.

Es importante mencionar que, para ambas bandas, también se analizó el comportamiento del multiplicador ante distintas potencias de entrada. Para esto, se fue cambiando la potencia de la señal generada con el generador de señales analógicas *Agilent*, de las figuras 3.13 y 3.15. Así, se encontró que, cuando la potencia de entrada al multiplicador era atenuada en 10 dB, haciéndola llegar hasta aproximadamente 5 dBm, se alcanza un punto óptimo en el cual se presenta la mayor potencia de salida del multiplicador.

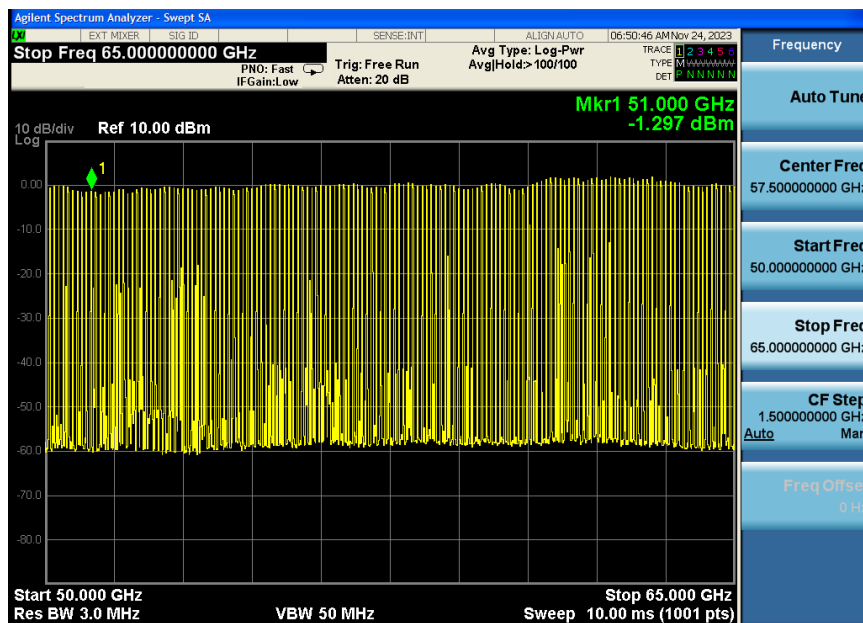


Figura 3.16: Resultado de la multiplicación en la banda de 50 – 60 GHz.

3.2.1.3. Atenuador Variable

El atenuador variable permite ajustar la potencia que sale desde el multiplicador, pudiendo así controlar este parámetro. Con esto, la potencia puede ser variada en función de la necesidad en el que el sistema se encuentre operando. En la tabla 3.11, se muestran las principales especificaciones del atenuador que fue seleccionado para el oscilador local.

Tabla 3.11: Especificaciones del atenuador variable.

Fabricante	<i>Aerowave</i>
Frecuencia	40 – 60 GHz
Pérdida de inserción	0.6 dB
Rango de Atenuación	0.6 – “∞” dB
Guía de Onda	WR-19

En la siguiente figura 3.17 se muestra el atenuador variable de perilla escogido.

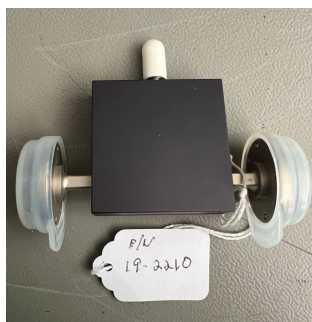


Figura 3.17: Atenuador variable de perilla.

3.2.1.4. Acoplador Direccional

El acoplador direccional permite obtener una muestra de la potencia de una señal de microondas, pero atenuada en cierto valor conocido. En este caso, una parte de la señal es dirigida al divisor, o *splitter*, y la segunda ingresa al detector de potencia de banda ancha atenuada en 20 dB. A partir de este último, se obtiene un voltaje que está definido por la sensibilidad del dispositivo. Finalmente, el voltaje pasa a un conversor análogo a digital para obtener muestras de la señal y mediante una equivalencia entre la sensibilidad y el voltaje, se muestra la potencia de la onda original. En la siguiente tabla 3.12 se muestran las especificaciones del dispositivo.

Tabla 3.12: Especificaciones del acoplador direccional.

Fabricante	<i>Qwinstar</i>
Frecuencia	40 – 60 GHz
Atenuación	20 dB
Guía de Onda	WR-19

En la siguiente figura 3.18 se muestra el acoplador direccional escogido.



Figura 3.18: Acoplador direccional.

3.2.1.5. Detector de Potencia de Banda Ancha UD

El detector recibe una muestra de la señal que proviene desde el acoplador direccional y, a su salida, entrega cierto nivel de voltaje en función de la potencia que fue recibida. En este caso, el detector se utilizó para la medición y control de la potencia que se entrega a la siguiente etapa del oscilador local. El dispositivo tiene como entrada una guía de onda y, a su salida, entrega el voltaje a través de un conector SMA. En la siguiente tabla 3.13 se muestran las especificaciones del detector.

Tabla 3.13: Especificaciones del detector de potencia de banda ancha UD.

Modelo	UD
Fabricante	<i>Pacific Millimeter</i>
Frecuencia	40 – 60 GHz
Sensibilidad Reportada	750 mV/mW ^a
Guía de Onda	WR-19
VSWR	2

^a El valor reportado es el mínimo que menciona el fabricante, pues este parámetro depende de la frecuencia y potencia de entrada.

Un resumen con la información de calibración del detector entregada por el fabricante, se muestra en la siguiente tabla 3.14. En esta, se observa la sensibilidad y voltaje como función de la frecuencia y potencia de entrada. Más información sobre el detector puede ser encontrada en el Anexo B.

Tabla 3.14: Tabla de calibración del detector de potencia de banda ancha UD.

Frecuencia (GHz)	Potencia (dBm)	Canal (V)	Sensibilidad (mV/mW)
40	-20.0	.01993	1993
45	-20.0	.01903	1902
50	-20.0	.02256	2256
55	-20.0	.02076	2076
60	-20.0	.01877	1876

La tabla 3.14 quiere decir que, para una potencia de entrada al detector, el voltaje de salida que muestra el dispositivo dependerá de la frecuencia y de la sensibilidad. Por ejemplo, si se tiene la potencia de entrada igual a -20 dBm con frecuencia de 50 GHz, la sensibilidad del detector será de 2256 mV/mW y en consecuencia la salida del detector será de $-.02256$ V. En la figura 3.19, se muestra el detector de potencia de banda ancha UD que fue escogido.



Figura 3.19: Detector de Potencia de Banda Ancha UD.

3.2.2. Componentes del Sistema de Control

3.2.2.1. Raspberry Pi

El eje central del sistema de control corresponde al *Raspberry Pi*. El pequeño computador cumple el rol de controlar y servir como el nexo de comunicación entre el usuario y sintetizador. Esto es realizado mediante una interfaz de línea de comandos diseñada para monitorear las principales características del sintetizador, como temperatura, voltajes, señales de referencia externa e interna y estado del *PLL*. Además, debe recibir desde el microcontrolador PIC, que es parte del sistema del control del telescopio, la frecuencia a la cual se debe definir la operación del sintetizador, lo que puede ser realizado mediante la conexión entre cliente-servidor de *sockets*.

Además, está encargada de realizar las mediciones de la potencia del oscilador local. Para esto, utiliza un conversor análogo a digital, el cual está encargado de recibir un voltaje proveniente desde el detector de potencia de banda ancha y, por las especificaciones del dispositivo, se puede realizar una equivalencia a potencia.

En paralelo a la medición de potencia y comunicación con el sintetizador, el *Raspberry Pi* debe generar una señal de 5 V para indicar el estado de enclavamiento de *PLL*. Para esto, consulta constantemente (cada dos segundos) el estado del sintetizador y obtiene información de los bits de estado del *PLL*. Las tres tareas mencionadas anteriormente son realizadas por el algoritmo diseñado, *rpiserver*, y será comentado en secciones posteriores al presente informe.

El modelo utilizado para el *Raspberry Pi* corresponde al 4B, presente en la siguiente figura 3.20.

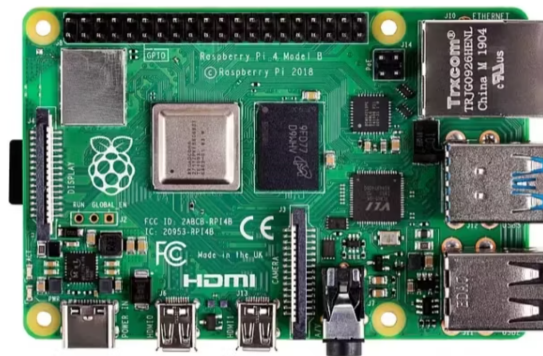
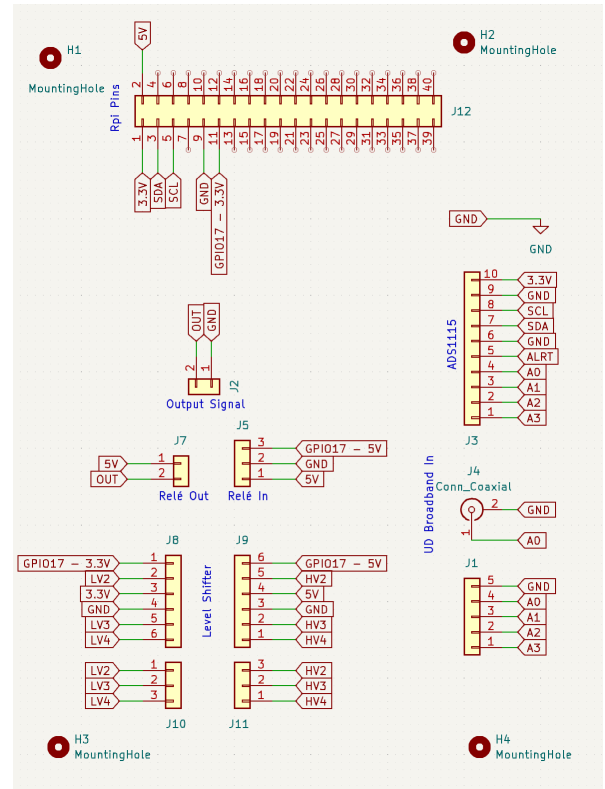
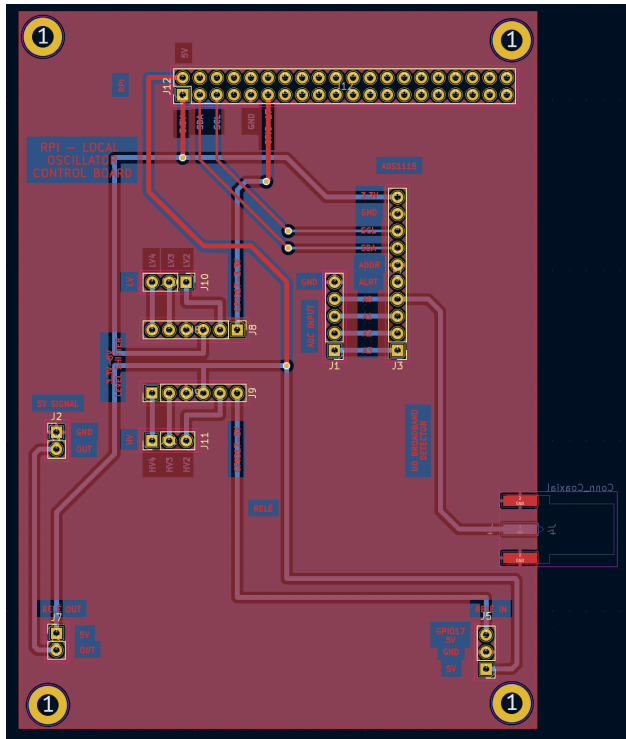


Figura 3.20: *Raspberry Pi* 4B.

3.2.2.2. Placa de Control Raspberry Pi - Oscilador Local

La placa de control del oscilador local con el *Raspberry Pi*, se diseñó con el objetivo de integrar los dos principales requisitos de *hardware* para el sistema de control: medición de potencia y generación de la señal de control. La primera versión de la placa, denominada como *RPI - Local Oscillator Control Board V1*, corresponde al diseño realizado del prototipo con el cual se trabajó en la *proto-board*. En las figuras 3.21.a y 3.21.b, se muestran los diseños en PCB y esquemáticos, respectivamente, de la placa de control.



(a) Diseño PCB de la placa de control del LO. (b) Diseño esquemático de la placa de control del LO.

Figura 3.21: Diseño en PCB y esquemático de la placa de control del LO.

La placa se compone de cuatro etapas diferentes, las cuales se describen a continuación:

1. Conexión con *Raspberry Pi*.

La primera etapa corresponde a las conexiones que se realizan directamente con el *Raspberry Pi*. Con el objetivo de minimizar los espacios que ocupan los dispositivos, la conexión entre los pines se realiza mediante el método de *hardware* sujeto en la parte superior, o *hardware attached on top* (HAT). El conector *J12* de la figura 3.22 muestra tal conexión. De los pines disponibles del *Raspberry Pi*, se utilizan los de alimentación 3.3 V, 5 V y *GND*, los del protocolo de comunicación *I²C* (*SCL* y *SDA*) y pin de propósito general de entrada/salida, *GPIO* por sus siglas en inglés. El resto de los pines quedan a libre disposición en caso de que sea necesario.

del módulo.

En la figura 3.22, el conector *J5* representa las conexiones de entrada del relé. De arriba hacia abajo, se tiene el pin de control *GPIO* del *Raspberry Pi*, luego de la etapa de subida de voltaje a 5 V, luego la conexión a tierra (*GND*) y, finalmente, el voltaje de operación del módulo. El conector *J7*, comenzando desde el pin superior, representa la conexión a *normalmente abierto* y que está conectado a los propios 5 V del *Raspberry Pi* para la generación del voltaje de tal magnitud. El pin inferior representa la señal de salida del relé y es la que estará controlada por *software* para indicar el estado del *PLL*.

Finalmente, el conector *J2* de la figura 3.22, entrega los 5 V generados por el relé. Los pines están conectados al mismo conector que actualmente entrega el voltaje necesario que indica al microcontrolador PIC el estado de enclavamiento del *PLL*.

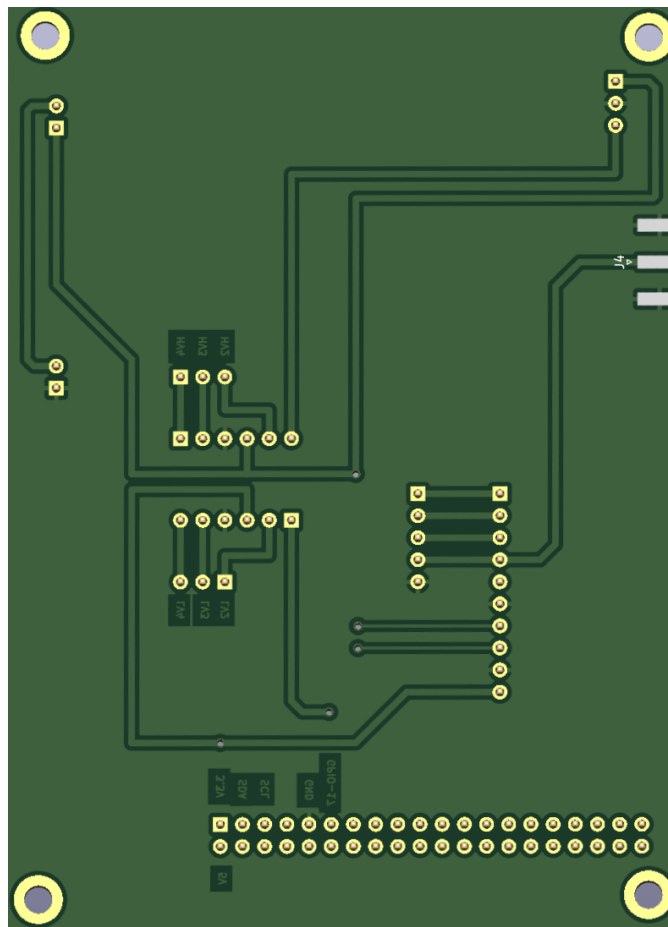


Figura 3.23: Vista trasera de la placa de control del LO.

3.2.2.3. Módulo ADS1115

Para la medición de la potencia que es entregada desde el atenuador variable, se utiliza el detector de potencia de banda ancha mencionado anteriormente. Este detector entrega un voltaje en función del parámetro de sensibilidad que se encuentra definido en el dispositivo. Para la lectura del voltaje, se utiliza un módulo conversor análogo a digital, el cual convierte

la señal analógica de voltaje entregada por el detector, a una señal digital que puede ser procesada por el *Raspberry Pi*.

En la siguiente figura 3.24 se muestra el módulo conversor analógico a digital *ADS1115*. Además, en los pines se pueden identificar aquellos correspondientes a los de alimentación (*VDD* y *GND*), a los del protocolo *I²C* (*SCL*, *SDA*, *ADDR* y *ALRT*), más las entradas analógicas (*A0* hasta *A3*).



Figura 3.24: Módulo conversor analógico digital ADS1115.

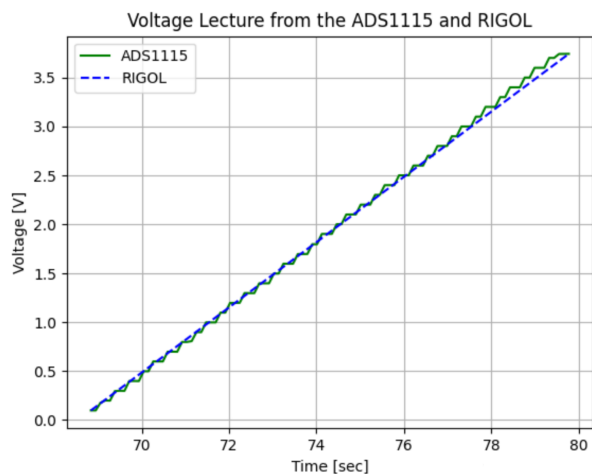
Se escogió el módulo *ADS1115* como el conversor analógico a digital, *ADC* por sus siglas en inglés, debido a que cuenta una precisión de 16 bits y que además es compatible su comunicación mediante la interfaz serial *I²C*. En la siguiente tabla 3.15, se muestran las principales características del dispositivo:

Tabla 3.15: Especificaciones del módulo *ADS1115*.

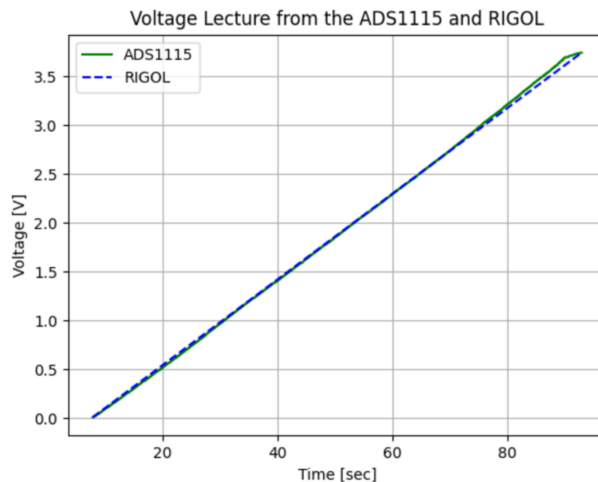
Modelo	ADS1115
Fabricante	<i>Texas Instruments</i>
Voltaje de operación	3.3 V
Interfaz comunicación	<i>I²C</i>
Precisión	16 Bits
Tasa de conversión	8 – 860 SPS
Voltaje de entrada	$\pm 6.144/PGA$
Amplificador de ganancia programada (PGA)	2/3, 1, 2, 4, 8, 16

Con el objetivo de medir la precisión del módulo *ADS1115*, se realizó una comparación para distintos pasos del voltaje entregado por la fuente de alimentación DC lineal programable, RIGOL, y el voltaje leído por el módulo.

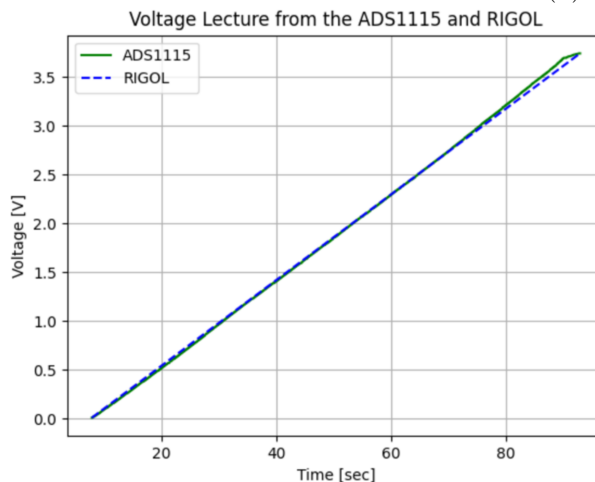
Para las mediciones, se escogió el intervalo entre 0 V y 4 V, lo cual es similar al rango de lectura del detector de potencia de banda ancha UD. Con esto, se escogieron tres pasos distintos: 0.1 V, 0.01 V y 0.001 V y los resultados de las mediciones y comparaciones se pueden observar en las figuras 3.25.a, 3.25.b y 3.25.c, respectivamente.



(a) Paso = 0.1 V.



(b) Paso = 0.01 V.



(c) Paso = 0.001 V.

Figura 3.25: Lectura de voltaje de RIGOL y *ADS1115* entre 0 V y 4 V para distintos pasos.

3.2.2.4. Conversor de Nivel

Para que sea factible el control del relé mediante el pin de *GPIO* del *Raspberry Pi*, es necesario la incorporación de un conversor de niveles lógicos. El pin de *GPIO* entrega un voltaje de 3.3 V, mientras que el pin de control del relé requiere de 5 V y en consecuencia, el conversor de nivel debe ser capaz de subir el voltaje de un nivel bajo hasta uno alto. En la siguiente tabla 3.16, se muestran las especificaciones del conversor de nivel para el sistema de control.

Tabla 3.16: Especificaciones del módulo conversor de nivel.

Modelo	BOB 12009
Fabricante	<i>Sparkfun</i>
Voltaje de nivel bajo	3.3 V
Voltaje de nivel alto	5 V
Número de canales	4
LV1	<i>GPIO17 - 3.3 V</i>
HV1	<i>GPIO17 - 5 V</i>

De esta manera, se eligió el conversor de nivel que se muestra en la figura 3.26. En ella, se pueden observar los niveles bajos y altos, denominados por *LV* y *HV*, respectivamente, más la conexión a tierra (*GND*). Además, los pines de conexión para los niveles bajos y altos (*LV1*, ..., *LV4* y *HV1*, ..., *HV4*), representan a las entradas y/o salidas del dispositivo de acorde a la funcionalidad que se le desee entregar.

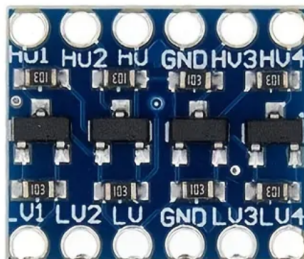


Figura 3.26: Módulo conversor de nivel.

3.2.2.5. Relé

Para generar la señal de 5 V que indica el estado del enclavamiento del oscilador local, se utiliza un relé que está controlado por el *Raspberry Pi*. El módulo relé de 1 canal integra dos led que indican tanto el encendido y apagado de todo el módulo, como el estado de la salida.

En la figura 3.27, se muestran los pines de conexión y led del módulo relé de 1 canal. Los pines de la derecha (de arriba hacia abajo) indican la señal de control, para este caso *GPIO17* transformado a 5 V, la conexión a tierra (*GND*) y el voltaje de alimentación de 5 V (*VCC*). El led rojo muestra el estado del módulo y el verde la señal de salida. Los pines de la izquierda (de arriba hacia abajo) significan normalmente abierto (*normally open*, en inglés), la señal de salida (*COMM*) y normalmente cerrado (*normally closed*, en inglés).



Figura 3.27: Módulo relé.

En resumen, el led verde está encendido y entrega 5 V a la salida cuando el *PLL* está enclavado y se apaga, y no entrega voltaje, cuando el *PLL* está desenclavado. En la siguiente tabla 3.17 se muestran las especificaciones para el relé de 1 canal que fue utilizado.

Tabla 3.17: Especificaciones del módulo relé de 1 canal.

Modelo Relé	SRD-05VDC-SL-C
Fabricante	<i>Songle</i>
Voltaje de operación	5 V
Número de canales	1
Voltaje de salida	5 V

3.3. Implementación del Oscilador Local y su Sistema de Control

3.3.1. Funcionamiento del Sistema de Control

El programa, denominado como *rpiserver*, corresponde a una interfaz de línea de comandos mediante la cual se puede controlar la frecuencia que se define en el sintetizador. Además, permite la medición de la potencia de salida del oscilador local y el constante monitoreo de los componentes del sistema.

El sistema de control que se diseñó para el oscilador local, se puede dividir en dos componentes. El primero, relacionado con el *software*, consiste en el funcionamiento del código de *rpiserver*, en donde se realizó la programación necesaria para el monitoreo de los dispositivos, sintonización de frecuencia y medición de potencia. El segundo componente, de *hardware*, se compone principalmente por la placa de control diseñada e impresa para el oscilador local.

3.3.1.1. Componente de *Software*

La figura 3.28 muestra el diagrama de flujo del programa *rpiserver*, el sistema de control del oscilador local. En primer lugar, cuando se ejecuta *rpiserver*, comienza la inicialización del servidor que está esperando que el cliente envíe el valor de frecuencia que desee sintonizar. En paralelo a la ejecución del servidor, se inicializan tres hilos que describen los procesos de manejo de comandos, monitoreo de componentes y manejo de frecuencia.

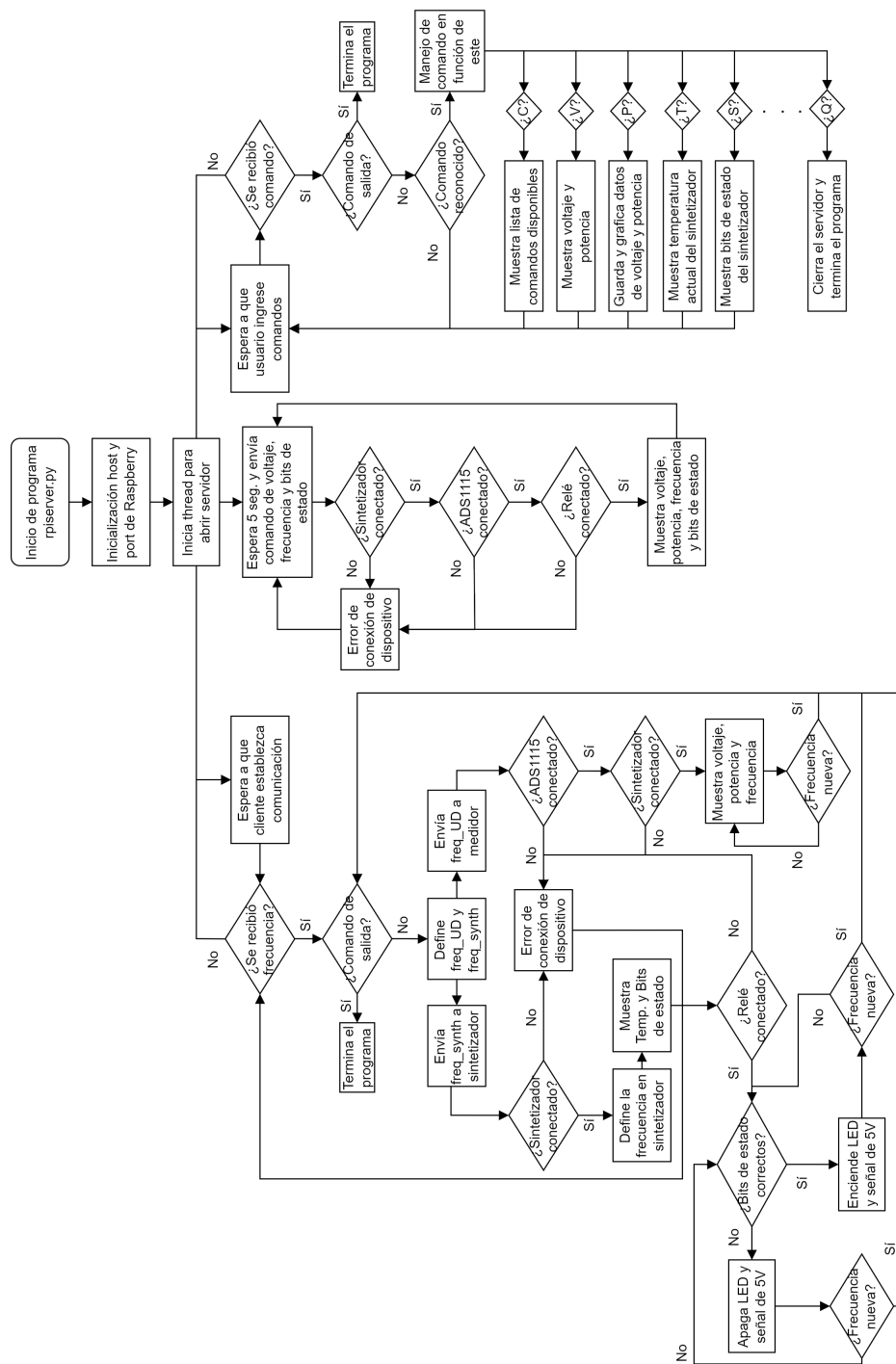


Figura 3.28: Diagrama de flujo de *rpiserver*.

- **Manejo de Comandos.** Para el primer hilo, el proceso se mantiene en espera mientras que el usuario ingresa comandos. Cuando lo realiza, se verifica que este no corresponda al comando de salida, y cuando no lo es, verifica que el comando esté definido y exista. Si no es el comando de salida y este es reconocido, se ejecuta el proceso relacionado al comando. En la primera versión de la implementación, se reconoce “C” para enseñar la lista de comandos disponibles, “V” para mostrar de manera inmediata el voltaje y

potencia, “P” para guardar y graficar los datos de voltaje y potencia, “T” para visualizar el valor actual de la temperatura del sintetizador, “S” para mostrar los bits de estado del sintetizador y “Q” para cerrar el servidor y terminar el programa.

Se desarrolló una primera versión en la cual se definió una lista con los mensajes disponibles para ser enviados. En la siguiente tabla 3.18, se muestran los comandos disponibles.

Tabla 3.18: Comandos disponibles para ser enviados en *rpiserver*.

Comando	Descripción
C	Mostrar comandos disponibles
V	Muestra las lecturas de frecuencia, voltaje y potencia
P	Guarda los datos de V y P en un CSV además de realizar un gráfico
T	Muestra la temperatura actual del sintetizador
S	Muestra los bits de estado del sintetizador
Q	Cierra el servidor y el programa

Dentro de la lista, el comando “V” muestra las lecturas del último valor de frecuencia que fue enviado al sintetizador y al sistema de medición de potencia, el cual es la multiplicación por 4 del valor anterior. Además, se muestra el valor de voltaje que es medido por el detector y la posterior transformación para el resultado de potencia. La respuesta típica que se muestra ante el envío del comando se entrega en la siguiente figura 3.29. Por un lado, se ve la frecuencia generada por el sintetizador equivalente a 12000 MHz y, por el otro, los datos de frecuencia, voltaje y potencia para el sistema de medición.

```

----- Información Sintetizador -----
Frecuencia Sintetizador: 12000.0 [MHz]
-----
--- Información UD Broadband Detector ---

Frecuencia: 48.0 [GHz]
Voltaje UD: 0.0004 [V]
Potencia: -37.4319 [dBm]
-----

```

Figura 3.29: Respuesta del comando “V” en *rpiserver*.

El comando “P” toma datos por un intervalo de tiempo definido, por ejemplo 10 seg., y genera un archivo con extensión *.csv* para guardarlos. Dentro de este, se guarda el tiempo, voltaje, potencia y frecuencia. Posteriormente, estos datos son mostrados en dos gráficos: Voltaje y Potencia, y Tiempo y Voltaje. El proceso para realizar la conversión se comenta posteriormente, mientras que la respuesta típica ante el comando “P” se muestra a continuación en la figura 3.30.

En la parte superior de la figura 3.30, se puede ver la terminal del *Raspberry Pi* cuando ejecuta el programa *rpiserver*. A la izquierda, se muestra el archivo que se genera, dentro del mismo directorio en el que se encuentra en *rpiserver*, de modo tal que se genera un

respaldo de los datos en caso de que sean necesarios. En el lado derecho, se muestran los gráficos que son generados con los datos. En el primer gráfico, se muestra Tiempo y Voltaje, donde las variaciones son producidas por el cambio de menos de 1 milésima. El segundo gráfico muestra el Voltaje y Potencia.



Figura 3.30: Respuesta del comando “P” en *rpiserver*.

Para los comandos “T” y “S”, la respuesta muestra los valores de temperatura y bits de estado actuales del sintetizador, respectivamente. Una típica respuesta ante el envío de tales comandos se muestra en la siguiente figura 3.31.

```

Command sent successfully: ?
Response from device (status raw): 11000011
D0: 100 MHz Locked (Internal Ref)
D1: YIG PLL Locked (External Ref)
D6: Self Test Passed
D7: Novo Locked
Sending command: T (Temperature request)
Command sent successfully: T
Response from device (temperature raw): +54.9C
Response from device (temperature): +54.9C
  
```

Figura 3.31: Respuesta del comando “S” y “T” en *rpiserver*.

El comando “C” muestra la lista de comandos disponibles y que son reconocidos por el programa, mientras que “Q” cierra el servidor, termina la ejecución del programa y deja

la terminal disponible del *Raspberry Pi*. Un ejemplo ante el envío de los comandos, se muestra en la siguiente figura 3.32.

```
C
'V' -> Muestra valores de Voltaje [V] y Potencia [dBm]
'P' -> Guarda los datos de V y P en un CSV además de graficarlos
'T' -> Muestra la Temperatura actual del Sintetizador
'S' -> Muestra los Bits de estado del Sintetizador
'Q' -> Cerrar el Servidor
Ingrese un comando: Q
Finalizando Programa...
Programa Finalizado
pi@raspberrypi:~/instmini/lab_tests/comms $
```

Figura 3.32: Respuesta del comando “Q” y “C” en *rpiserver*.

- **Monitoreo del Estado de los Componentes.** Para el segundo hilo, se muestra, de manera periódica, la información relacionada al voltaje y potencia del sistema detector de potencia, la frecuencia que actualmente se encuentra sintonizada en el sintetizador y los bits de estado relacionados al estado y enclavamiento del dispositivo generador de señales. En caso de que algún componente (ya sea sintetizador, módulo *ADS1115* y relé) se encuentre desconectado, el programa muestra el respectivo mensaje del error en la conexión.

Este subproceso se encarga de consultar de manera periódica el estado de los componentes. En primera instancia, se escogió que las variables a ser monitoreadas son la frecuencia de operación, el voltaje que entrega el detector de potencia de banda ancha UD y la potencia de la muestra que es extraída desde el acoplador direccional.

El período definido, que también puede ser cambiado en función del interés del usuario, corresponde a 2 segundos. En la siguiente figura 3.33, se muestra un ejemplo de funcionamiento dentro de *rpiserver* y el monitoreo de la frecuencia, voltaje y potencia.

```
pi@raspberrypi:~/instmini/lab_tests/comms $ sudo python3 rpiserver.py
Error al abrir el dispositivo: open failed
Error al abrir el dispositivo: open failed
Error al abrir el dispositivo: open failed
Ingrese un comando:
Servidor escuchando en 169.254.81.30:8000

Frecuencia: 0 [GHz]
Voltaje UD: 0.0013 [V]
Potencia: -47.3366 [dBm]
Error al comunicarse con el dispositivo. Por favor verifique la conexión.

Frecuencia: 0 [GHz]
Voltaje UD: 0.0014 [V]
Potencia: -46.9258 [dBm]
Error al comunicarse con el dispositivo. Por favor verifique la conexión.

Frecuencia: 0 [GHz]
Voltaje UD: 0.0014 [V]
Potencia: -46.9258 [dBm]
Error al comunicarse con el dispositivo. Por favor verifique la conexión.
```

Figura 3.33: Monitoreo de frecuencia, voltaje y potencia en *rpiserver*.

- **Manejo de Frecuencia.** Finalmente, el tercer hilo, corresponde y se ejecuta cuando el cliente envía algún valor de frecuencia. Mientras que esto no suceda, las variables “*freq_synth*” y “*freq_UD*” se mantienen nulas y por ende los valores para voltaje, potencia y bits de estado dejan de tener validez. Además, se monitorea el ingreso del comando de salida, mediante el cual se termina la ejecución del programa. Cuando esto no sucede y se detecta un valor válido de frecuencia, se comienza con la definición de las dos variables mencionadas anteriormente. Con esto, por un lado, se define la frecuencia de la señal que es generada por el sintetizador y, por el otro lado, se realiza la conversión a potencia. El proceso puede ser dividido en dos etapas.

(i) **Cliente: Envío de frecuencia.**

En primer lugar, el proceso comienza cuando el cliente envía algún valor de frecuencia al servidor. Para las pruebas realizadas, el cliente fue otro computador que se conectó mediante *sockets* al servidor. En el caso de la integración al sistema de recepción del radiotelescopio, el cliente es el microcontrolador *PIC*.

La frecuencia, definida por código dentro del programa *client_rpi.py* de la sección C.4 del Anexo, simula el valor que es obtenido dentro del control de frecuencia de la sección D del Anexo, la cual depende de la fuente de observación. Posteriormente, la variable es enviada por *sockets* al servidor e inicia la segunda etapa del programa. En las siguientes figuras 3.34.a y 3.34.b, se muestra un ejemplo del envío y recepción de frecuencia, respectivamente.

```
pi@raspberrypi:~/instmini/lab_tests/comms $ sudo python3 client_rpi.py
Frecuencia enviada 12000000000 [Hz]
Respuesta del servidor:
  Frecuencia seteada en Sintetizador: 12000.0 [MHz]
  Frecuencia en Medidor de Potencia 48.0 [GHz]
```

(a) Ejemplo de envío de frecuencia por parte del cliente.

```
pi@raspberrypi:~/instmini/lab_tests/comms $ sudo python3 rpiserver.py
Error al abrir el dispositivo: open failed
Error al abrir el dispositivo: open failed

Servidor escuchando en 169.254.81.30:8000
Error al abrir el dispositivo: open failed
Ingrese un comando:

Frecuencia: 0 [GHz]
Voltaje UD: 0.0015 [V]
Potencia: -46.5506 [dBm]
Error al comunicarse con el dispositivo. Por favor verifique la conexión.
----- Información del Servidor -----
Conexión entrante de 169.254.81.30:36968
-----
Frecuencia recibida: 12000000000.0 [Hz]
Frecuencia Sintetizador: 12000.0 [MHz]
Frecuencia en Medidor de Potencia: 48.0 [GHz]
```

(b) Ejemplo de recepción de frecuencia por parte del servidor.

Figura 3.34: Ejemplo de envío y recepción de frecuencia.

Las imágenes fueron obtenidas de la terminal del *Raspberry Pi*, donde en la parte superior de las figuras se puede ver la ejecución de los programas *rpiserver* y *client_rpi*. Es importante notar que cuando en las simulaciones se muestra el men-

saje “Error al abrir el dispositivo: open failed”, es una situación normal. Esto se debe a que las pruebas se realizaron con el sintetizador desconectado.

(ii) **Servidor: Procesamiento de Frecuencia.**

Como se ha mencionado anteriormente, cuando el servidor recibe algún valor de frecuencia, define dos variables: “*freq_UD*” y “*freq_synth*”, las cuales abren los procesos para el sistema de medición de potencia. El procesamiento que ocurre con la frecuencia que se envía al sistema de medición de potencia, es simple, puesto que debe únicamente multiplicar el valor para transformarlo a GHz. De esta forma, se tiene:

$$freq_{UD} = frecuencia \cdot 4 / (10^9) \quad (3.1)$$

Donde *frecuencia* es el valor de la frecuencia enviada por el cliente y “*freq_UD*” es la frecuencia que se envía al sistema de medición de potencia. El valor se multiplica por 4 para simular el multiplicador y posteriormente, se transforma a GHz. Por otro lado, la frecuencia que se envía al sintetizador, denominada como “*freq_synth*”, debe transformarse a MHz para que sea válida como entrada.

$$freq_{synth} = frecuencia / (10^6) \quad (3.2)$$

Luego de que se define “*freq_synth*”, el sintetizador genera la señal con igual frecuencia a la que fue enviada. Posteriormente, se muestran en la interfaz los valores que indican la temperatura y bits de estado del sintetizador. Con estos, se mantiene un monitoreo periódico con el objetivo de verificar que estos se encuentren dentro de los rangos establecidos, los cuales se muestran en la tabla 3.7.

Cuando se muestra en la consola de manera periódica la información de temperatura y bits, como se observa en la figura 3.31, no solamente se realiza para que el usuario lo pueda comprobar, sino que de manera interna se comprueba que las variables estén dentro del rango establecido. En el caso positivo, se procede a encender el relé y, en consecuencia, se prende el led y se genera la señal de control de 5 V. Mientras no se reciba una nueva frecuencia y los bits de estado no cambien, el led y la señal de 5 V se mantendrá sin cambios. En caso contrario, ya sea que se reciba una nueva frecuencia o que los bits de estado cambien, se detiene la generación de la señal y, por ende, la observación se finaliza.

3.3.1.2. Componente de *Hardware*

La segunda parte del sistema de control del oscilador local puede ser descrita por el *hardware*, es decir, los elementos físicos y tangibles que lo componen. Los dispositivos utilizados corresponden al *Raspberry Pi*, Módulo *ADS1115*, Convertidor de Nivel y Relé. Las pruebas previas a la implementación fueron realizadas con la configuración de mediciones de la figura 3.35. En ella, se pueden observar los cuatro componentes principales, mientras que las etiquetas simulan la salida del detector de potencia de banda ancha UD y la salida del relé con la señal de control. Además, es importante mencionar que el sintetizador (mostrado en el lado izquierdo de la figura 3.35) es controlado mediante la interfaz *rpiserver*.

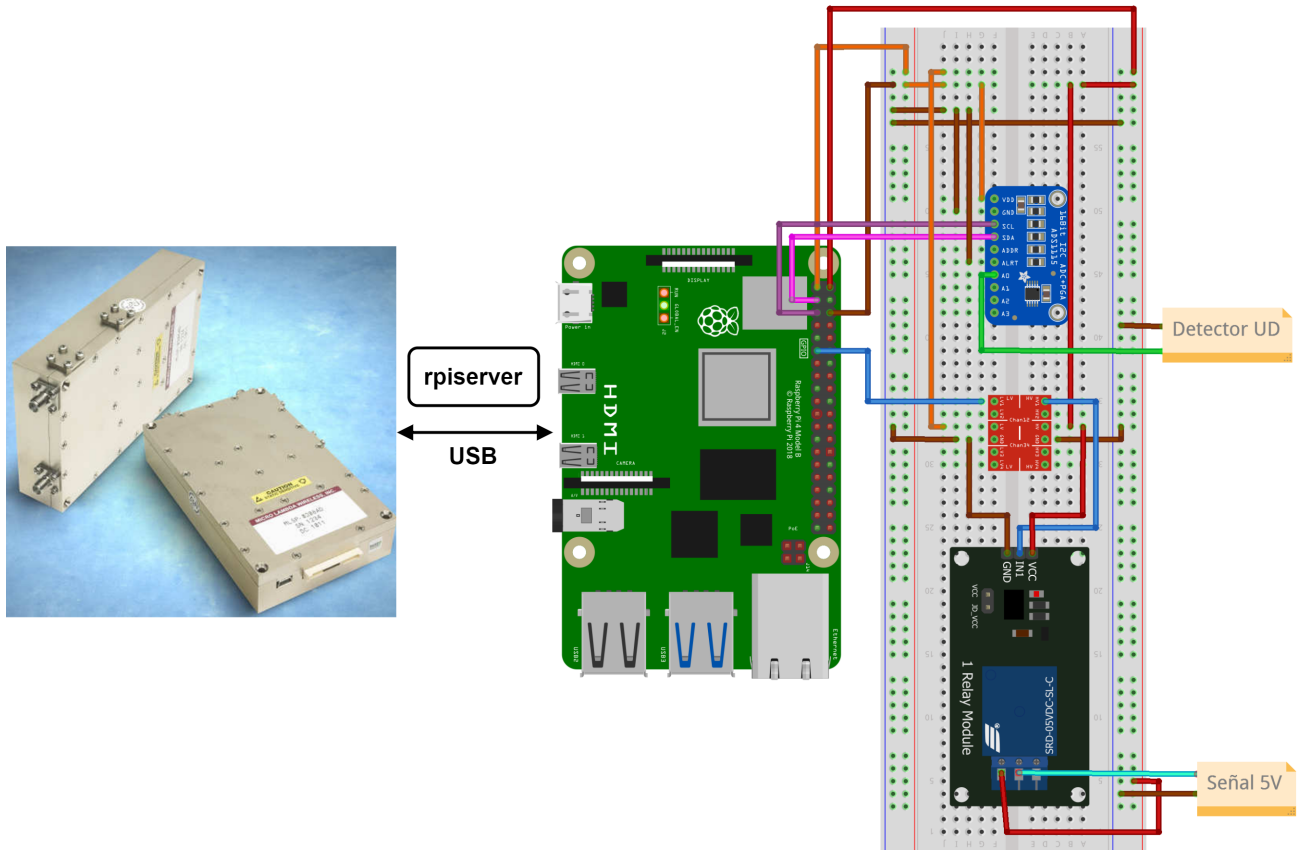


Figura 3.35: Configuración de mediciones del sistema de control para el oscilador local.

Con el funcionamiento exitoso del circuito de la figura 3.35, se trabajó con la placa de control del oscilador local mostrada en la figura 3.36. Por otro lado, se simuló la señal de salida del detector de potencia de banda ancha UD con el dispositivo generador de formas de onda arbitrarias *Agilent 33220 A*. Para la verificación de la señal de control de 5 V, se comprobó de manera visual el estado del led del relé, más la lectura con multímetro de la salida.

Además del generador arbitrario de formas de onda *Agilent*, se utilizó una fuente de alimentación DC programable *RIGOL* para satisfacer la alimentación de 5 V y 15 V del sintetizador. Este último, se encuentra conectado a la placa adaptadora de pines, para cumplir con la alimentación y también al *Raspberry Pi*, para establecer la comunicación. Finalmente, en la parte superior del microcomputador se encuentra conectada la placa del sistema de control para el oscilador local, en donde se pueden encontrar los dispositivos del sistema de control.

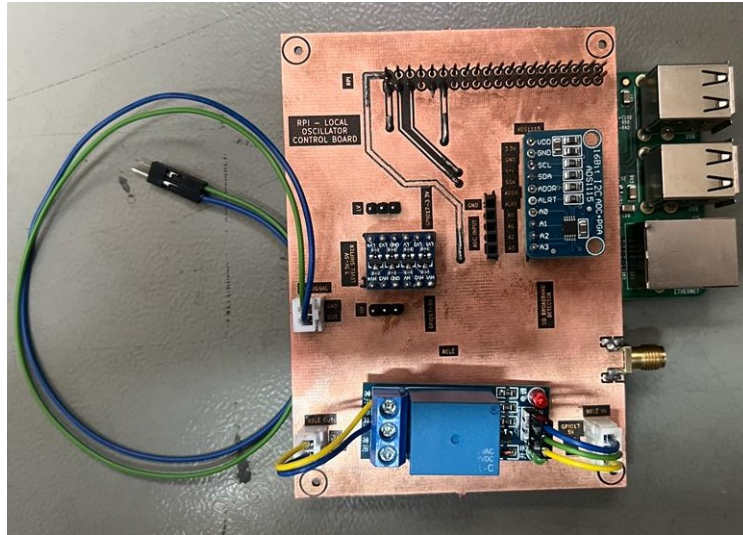
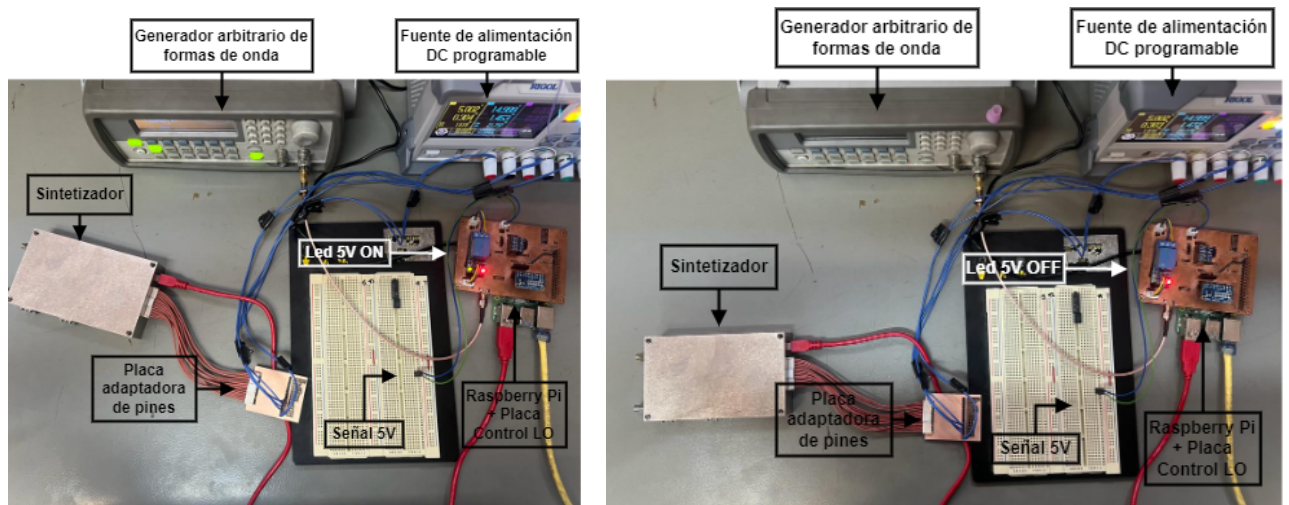


Figura 3.36: *Raspberry Pi* bajo el *hat* de la placa de control para el oscilador local.

- (i) **Señal de Control.** Para la primera prueba, se comprobó que cuando el sintetizador se encuentre en su estado óptimo y las referencias externas, internas y PLL del YIG, se encuentren enclavadas, entonces se encienda el relé y se genere la señal de control de 5 V. De la misma forma, se verificó que cuando las condiciones no se cumplieran, la señal no fuese generada.

Así, cuando los anteriores requerimientos se cumplen, la respuesta ante los bits de estado será: “11000011”. Esto se observa de manera física en la figura 3.37.a, en donde se muestra el encendido del led verde del relé y la medición de 5 V. Por el otro lado, cuando las condiciones no se cumplen, se muestra en la figura 3.37.b que el led está apagado y no se mide el voltaje de la señal de salida.



(a) Ejemplo de enclavamiento y encendido de relé.

(b) Ejemplo de desenclavamiento y apagado de relé.

Figura 3.37: Ejemplo de enclavamiento y desenclavamiento de la señal y estado del relé, en físico.

Los resultados anteriores se pueden ver reflejados en *rpiserver* cuando los bits de estado son correctos e incorrectos en las figuras 3.38.a y 3.38.b, respectivamente.

```
pi@raspberrypi:~/instmini/lab_tests/comms $ sudo python3 rpiserver.py
Conexión establecida con el dispositivo USB.

Servidor escuchando en 169.254.81.30:8000
Error al abrir el dispositivo: open failed
Ingrese un comando: Conexión establecida con el dispositivo USB.

Frecuencia: 0 [GHz]
Voltaje UD: 0.0016 [V]
Potencia: -46.2051 [dBm]
----- Información Sintetizador -----
Comando de solicitud de temperatura enviado: T
Respuesta inválida del dispositivo (temperatura): +26.8C
----- Información Sintetizador -----
Comando de solicitud de bits de estado enviado: ?
Bits de estado del sintetizador: 11000011
B0: 100 MHz Locked (Internal Ref)
B1: YIG PLL Locked
B6: Self Test Passed
B7: NOVO locked (External Ref)
RELÉ LED ON => +5V

Frecuencia: 0 [GHz]
Voltaje UD: 0.0013 [V]
Potencia: -47.3366 [dBm]
----- Información Sintetizador -----
Comando de solicitud de temperatura enviado: T
Respuesta inválida del dispositivo (temperatura): +27.0C
----- Información Sintetizador -----
Comando de solicitud de bits de estado enviado: ?
Bits de estado del sintetizador: 11000011
B0: 100 MHz Locked (Internal Ref)
B1: YIG PLL Locked
B6: Self Test Passed
B7: NOVO locked (External Ref)
RELÉ LED ON => +5V
```

(a) Ejemplo de enclavamiento y encendido de relé.

```
pi@raspberrypi:~/instmini/lab_tests/comms $ sudo python3 rpiserver.py
Servidor escuchando en 169.254.81.30:8000
Conexión establecida con el dispositivo USB.
Error al abrir el dispositivo: open failed
Ingrese un comando: conexión establecida con el dispositivo USB.

Frecuencia: 0 [GHz]
Voltaje UD: 0.0016 [V]
Potencia: -46.2051 [dBm]
----- Información Sintetizador -----
Comando de solicitud de temperatura enviado: T
Respuesta inválida del dispositivo (temperatura): +29.1C
----- Información Sintetizador -----
Comando de solicitud de bits de estado enviado: ?
Bits de estado del sintetizador: 11000010
B0: 100 MHz Locked (Internal Ref)
B1: YIG PLL Locked
B6: Self Test Passed
B7: NOVO Unlocked (External Ref)
Revise Bits de estado de Sintetizador
RELÉ LED OFF => 0V

Frecuencia: 0 [GHz]
Voltaje UD: 0.0013 [V]
Potencia: -47.3366 [dBm]
----- Información Sintetizador -----
Comando de solicitud de temperatura enviado: T
Respuesta inválida del dispositivo (temperatura): +29.3C
----- Información Sintetizador -----
Comando de solicitud de bits de estado enviado: ?
Bits de estado del sintetizador: 11000010
B0: 100 MHz Locked (Internal Ref)
B1: YIG PLL Locked
B6: Self Test Passed
B7: NOVO Unlocked (External Ref)
Revise Bits de estado de Sintetizador
RELÉ LED OFF => 0V
```

(b) Ejemplo de desenclavamiento y apagado de relé.

Figura 3.38: Ejemplo de enclavamiento y desenclavamiento de la señal y estado del relé, en la terminal.

En la figura 3.39, se observa en el multímetro la lectura del voltaje de la señal de control de 5 V. Además, a la derecha de la figura se observa el relé con el led de color verde encendido, el cual indica la generación de la señal.

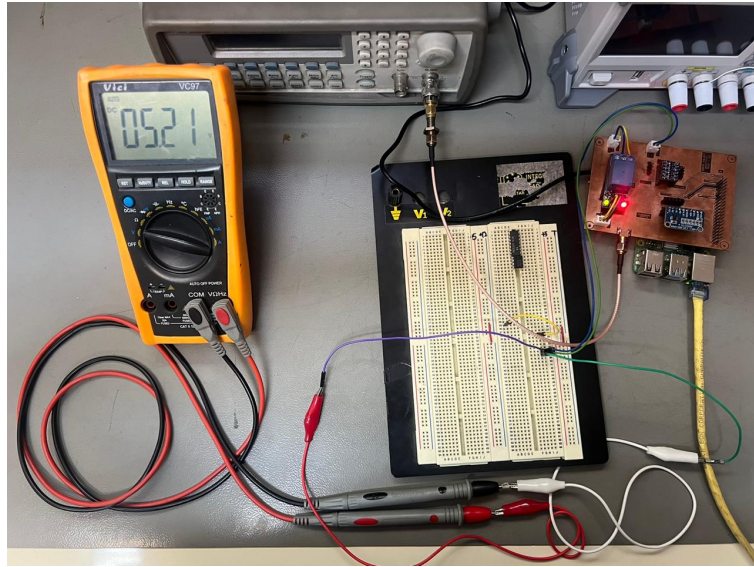


Figura 3.39: Medición de la señal de control cuando el sintetizador se encuentra en condiciones de operación.

- (ii) **Medición de Potencia.** Para la segunda prueba, en la que se verifica la medición de potencia del sistema de control, se inicia el procedimiento con la definición de la frecuencia con la que se va a operar. En este caso en particular, la frecuencia enviada por el usuario fue de 12 GHz y el resultado de la multiplicación, por cuatro, se muestra en la primera línea de la figura 3.40. En la segunda línea se utilizó un voltaje arbitrario, que posteriormente fue cambiando, que simula la lectura del detector de potencia de banda ancha UD, mientras que en la tercera se enseña el resultado de la conversión realizada hacia potencia.

Posteriormente, se muestra el estado del sintetizador en términos de temperatura, la cual se encuentra dentro del rango de operación aceptable, y los bits de estado con la respectiva información de las referencias externas e internas, PLL del YIG y estado del sintetizador. Finalmente, como el sistema se encuentra enclavado y dentro de los rangos de operación, se puede ver reflejado el mensaje que indica el encendido del led del relé y la generación de 5 V para la señal de control.

Además, en los dos bloques inferiores restantes, se pueden ver los mismos datos que fueron mencionados anteriormente, pero para otros voltajes y potencias.

```

Frecuencia: 48.0 [GHz]
Voltaje UD: 0.3384 [V]
Potencia: -23.0462 [dBm]
----- Información Sintetizador -----
Comando de solicitud de temperatura enviado: T
Respuesta inválida del dispositivo (temperatura): +34.2C
----- Información Sintetizador -----
Comando de solicitud de bits de estado enviado: ?
Bits de estado del sintetizador: 11000011
B0: 100 MHz Locked (Internal Ref)
B1: YIG PLL Locked
B6: Self Test Passed
B7: NOVO locked(External Ref)
RELÉ LED ON => +5V

Frecuencia: 48.0 [GHz]
Voltaje UD: 0.8980 [V]
Potencia: -18.8075 [dBm]
----- Información Sintetizador -----
Comando de solicitud de temperatura enviado: T
Respuesta inválida del dispositivo (temperatura): +34.4C
----- Información Sintetizador -----
Comando de solicitud de bits de estado enviado: ?
Bits de estado del sintetizador: 11000011
B0: 100 MHz Locked (Internal Ref)
B1: YIG PLL Locked
B6: Self Test Passed
B7: NOVO locked(External Ref)
RELÉ LED ON => +5V

Frecuencia: 48.0 [GHz]
Voltaje UD: 0.6745 [V]
Potencia: -20.0504 [dBm]
----- Información Sintetizador -----
Comando de solicitud de temperatura enviado: T
Respuesta inválida del dispositivo (temperatura): +34.6C
----- Información Sintetizador -----
Comando de solicitud de bits de estado enviado: ?
Bits de estado del sintetizador: 11000011
B0: 100 MHz Locked (Internal Ref)
B1: YIG PLL Locked
B6: Self Test Passed
B7: NOVO locked(External Ref)
RELÉ LED ON => +5V

```

Figura 3.40: Ejemplo en la terminal del *Raspberry Pi* del funcionamiento del sistema de control para el oscilador local.

3.3.1.3. Sistema de Medición de Potencia

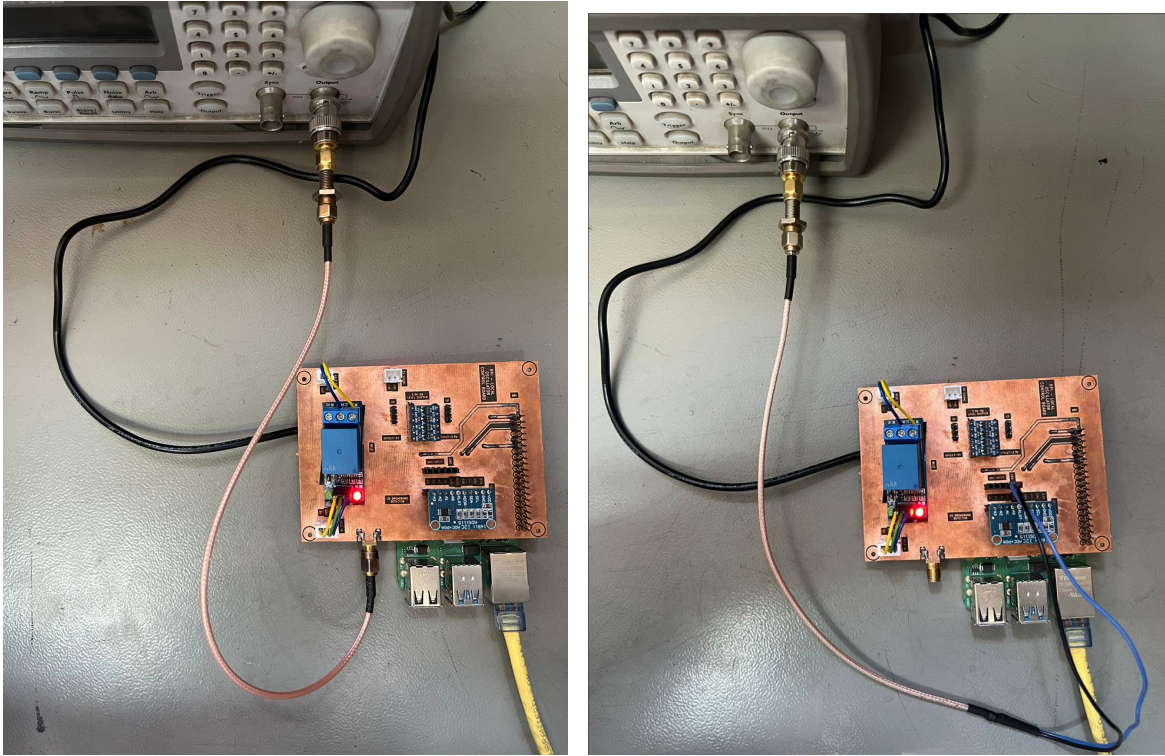
El sistema de medición de potencia entrega de manera periódica el valor de la potencia del sistema, la cual depende de manera directa del valor del voltaje que es entregado por el detector de potencia de banda ancha, además de la frecuencia a la cual se está operando. De esta forma, se puede describir en dos etapas:

i) Detector de Potencia de Banda Ancha y Módulo ADS1115.

La primera etapa describe el componente de *hardware* del sistema. El detector de potencia de banda ancha UD recibe en guía de onda, la muestra de la señal que proviene desde el acoplador direccional. El dispositivo entrega una señal de voltaje que depende de la frecuencia y potencia.

La señal de voltaje analógica es muestreada por el módulo conversor análogo a digital *ADS1115*, y la conversión de voltaje a potencia es realizada por *software* dentro del

Raspberry Pi. La conexión física entre el detector y el conversor puede ser realizada en dos formas diferentes que provee la placa de control del oscilador local diseñada (ver figura 3.22): *SMA-Dupont* y *SMA-SMA*. En las siguientes figuras 3.41.a y 3.41.b, se muestra la conexión *SMA-SMA* y *SMA-Dupont*, respectivamente, entre el conversor y detector.



(a) Conexión SMA-SMA entre detector y conversor. (b) Conexión SMA-Dupont entre detector y conversor.

Figura 3.41: Modos de conexión entre detector y conversor.

ii) Conversión de Voltaje a Potencia.

La segunda etapa describe el componente de *software* del sistema de medición de potencia. La conversión, que es realizada en el programa *ads1115.py* de la sección del Anexo C.1, inicia definiendo la configuración del conversor. Para esto, define la ganancia del módulo en 1, lo cual implica un amplio rango de detección de ± 4.096 V pero menor capacidad de detecciones de pequeñas variaciones de voltaje. Además, de los cuatro canales de entrada disponibles en el conversor, se utiliza el canal A0 y se deja a libre disposición la utilización del resto de los tres canales.

Para realizar la conversión de voltaje a potencia, se utilizó el proceso contrario mostrado con la tabla 3.14. Esto quiere decir que, a partir del voltaje que es obtenido desde el detector UD más la frecuencia de operación que es conocida, se calcula el valor de sensibilidad y en consecuencia, la potencia que fue ingresada. Este valor puede ser comparado mediante el Analizador de Señales *PXA*, el cual entrega la potencia de salida del acoplador direccional sin ser atenuada en 20 dB (ver figura 3.3).

Debido a que la sensibilidad se vuelve un parámetro fundamental en la conversión de voltaje a potencia, es importante la generación de un modelo que pueda calcular este parámetro en función de los datos de calibración de la tabla 3.14. Para esto, se realizó una comparación entre dos tipos de interpolación de los datos: ajuste polinómico e interpolación PCHIP [34], el cual es un método de interpolación cúbica por tramos de Hermite en el que se conserva la monotonicidad y la forma de los datos. En la siguiente figura 3.42 se muestra la comparación realizada.

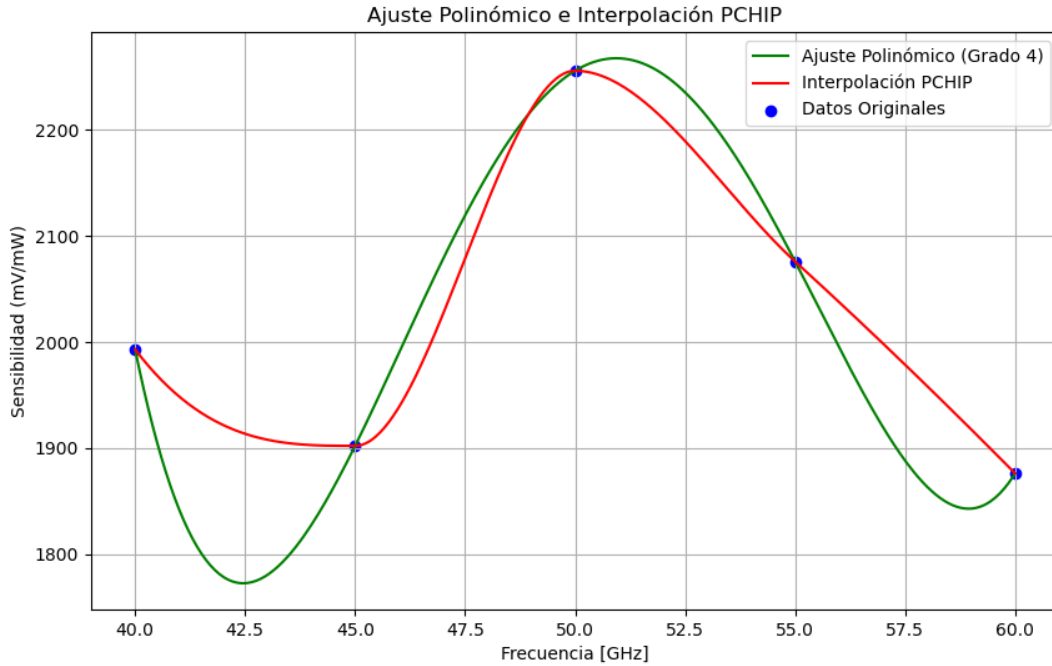


Figura 3.42: Ajuste polinómico e interpolación PCHIP de los datos de la tabla 3.14.

Con el gráfico de la figura 3.42, se concluye que para los datos de calibración, la curva de ajuste a utilizar corresponde a la generada por la interpolación PCHIP. El descarte del ajuste polinómico se debe a que genera puntos máximos y mínimos locales que no son deseados. Con la interpolación realizada, es posible obtener la sensibilidad a partir del valor conocido de frecuencia. De esta forma, se define el parámetro *sensibilidad* como sigue:

$$data_{freq} = [40, 45, 50, 55, 60] \quad (3.3)$$

$$data_{sens} = [1993, 1902, 2256, 2076, 1876] \quad (3.4)$$

$$sensibilidad = PchipInterpolator(data_{freq}, data_{sens}) \quad (3.5)$$

Si se considera que la *PchipInterpolator* es una función de la librería *scipy.interpolate*, $data_{freq}$ corresponde a los valores de frecuencia y $data_{sens}$ los valores de sensibilidad, se puede definir posteriormente la conversión hacia potencia como sigue:

$$potencia_{mW} = (voltaje \cdot 1000) / sensibilidad \quad (3.6)$$

$$potencia_{dBm} = 10 \cdot \log(potencia_{mW}) \quad (3.7)$$

Donde *voltaje* es el valor enviado por el detector UD y leído por el conversor ADC, y las variables *potencia_{mW}* y *potencia_{dBm}* corresponden a la potencia en *miliwatts* y en *dBm*, respectivamente.

A partir del modelo presentado, se logró obtener para distintos valores de frecuencia, la relación entre voltaje y potencia que sería leído por el conversor entre el intervalo de 0 y 4 V. En la siguiente figura 3.43 se muestra el gráfico de voltaje y potencia, entre 0 V y 4 V para distintas frecuencias.

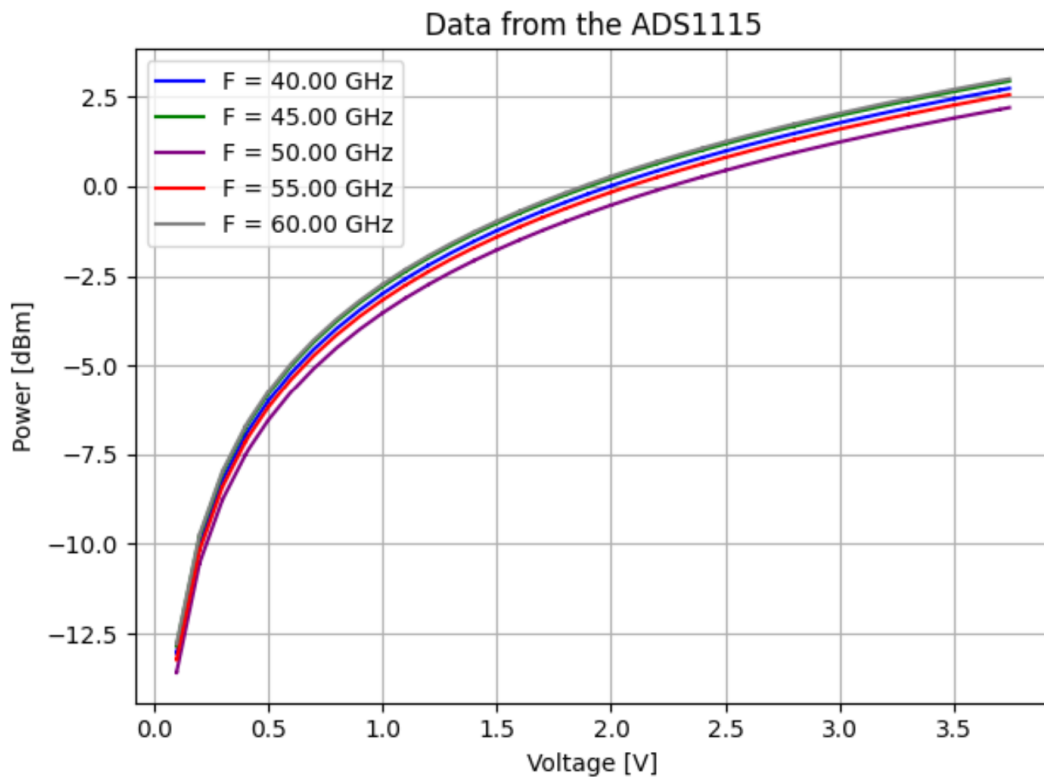


Figura 3.43: Gráfico de voltaje y potencia entre 0 V y 4 V para distintas frecuencias.

En resumen, a partir de la interpolación realizada de los datos de calibración, se puede obtener la sensibilidad como función de la frecuencia. Posteriormente, si se aplica la sensibilidad obtenida a la lectura de voltaje extraída del módulo *ADS1115*, se puede realizar la equivalencia hacia potencia.

3.3.2. Configuración de Mediciones del Oscilador Local

Es importante describir la configuración de las mediciones que fueron realizadas. Es por esto que se destaca que la toma de datos fue realizada en dos intervalos diferentes debido a

la restricción de equipos que se encuentran en el laboratorio. De esta forma, a continuación se comentan las especificaciones para cada banda.

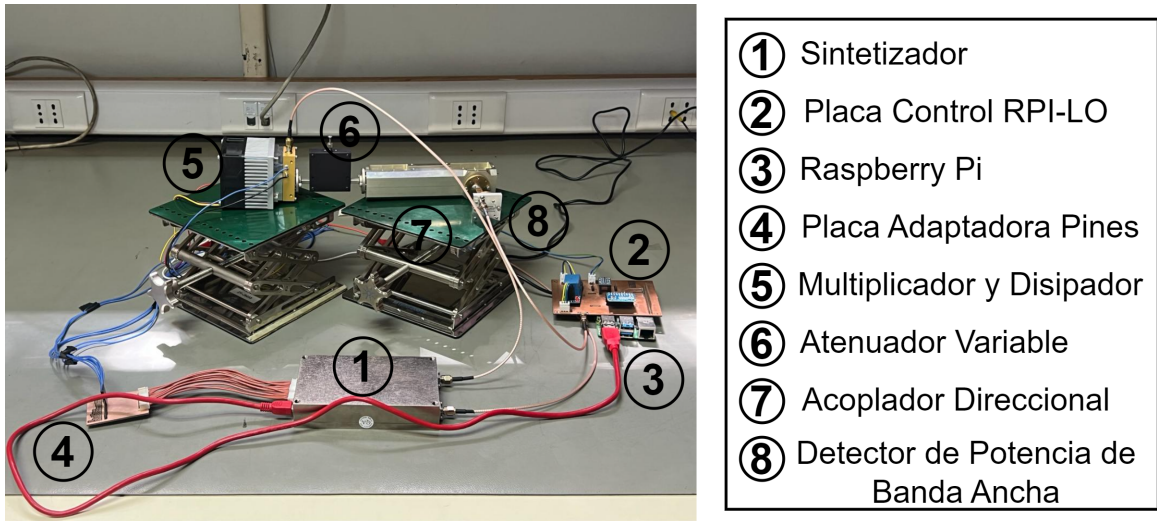


Figura 3.44: Configuración de mediciones para la cadena de oscilador local.

3.3.2.1. Banda 36-50 GHz.

Para la primera banda, la configuración de las mediciones, o *setup* en inglés, se muestra en la figura 3.45. Para las pruebas, se utilizaron dos fuentes de alimentación *RIGOL*. Una de ellas se conecta a la placa adaptadora de pines para cumplir con los 5 V y 15 V necesarios para alimentar al sintetizador. La restante, se dirige directamente con 8 V para la operación del multiplicador, más 12 V para su ventilador y disipador.

Una de las salidas del acoplador direccional se dirige a una transición de guía de onda *WR-22* a SMA. Allí, se realiza la conexión con el Analizador de Señales PXA para visualizar la señal resultante de todo el proceso previo. Es importante destacar que se utiliza la referencia de 10 MHz presente dentro del mismo analizador, como referencia externa del sintetizador. Finalmente, en la salida acoplada, se conectó una guía de onda *WR-19* hacia el detector de potencia de banda ancha UD.

Es relevante mencionar que entre el sintetizador y el multiplicador, se encuentra un atenuador fijo de 10 dB. Este último se sitúa en el diagrama de configuraciones de ambas bandas y su incorporación a la cadena de oscilador local es un caso de análisis, tal como fue mencionado en la sección 3.2.1.2. En consecuencia, los resultados que se muestran en el siguiente capítulo estudian el efecto que tiene la inclusión del atenuador fijo.

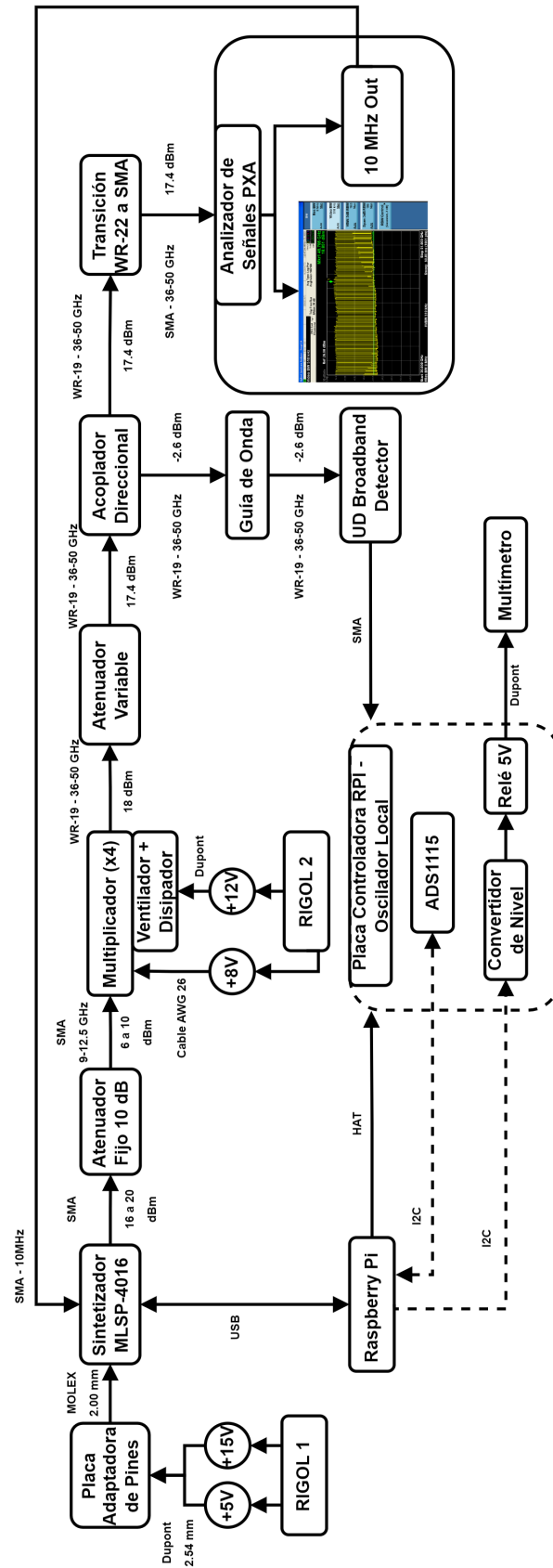


Figura 3.45: Diagrama de la configuración de mediciones utilizada para la banda 36 – 50 GHz del oscilador local.

En la siguiente figura 3.46, se muestra de manera física la configuración de mediciones para la banda de 36 – 50 GHz.

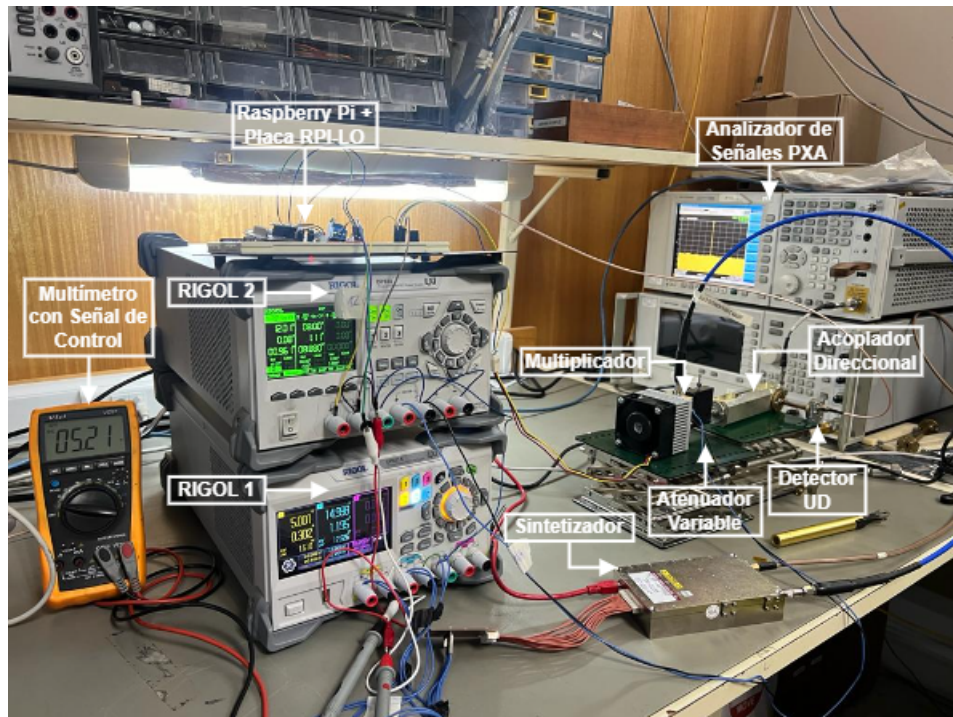


Figura 3.46: Configuración de mediciones utilizada para la banda 36 – 50 GHz para las pruebas del oscilador local.

3.3.2.2. Banda 50-60 GHz.

Para la segunda banda, la configuración de mediciones realizada se muestra en la figura 3.47. Si se compara con el diagrama de la figura 3.3 de la banda anterior, se desprende que se cambia la transición *WR-22* a SMA, por un módulo extensor. Además, se incluye otro atenuador variable para limitar la potencia que es ingresada al extensor. De este modo, la potencia que ingresa tanto al extensor como al analizador de señales PXA, queda limitada por este último atenuador.

El módulo extensor corresponde al mismo utilizado para la medición realizada con el multiplicador en la sección 3.12. De esta forma, es posible utilizar la entrada para la terminal del módulo extensor presente en el analizador de señales PXA, donde, además, se mantiene la referencia de 10 MHz conectada al sintetizador. En consecuencia, con el módulo se puede visualizar la banda desde 50 GHz a 75 GHz, cumpliéndose los requisitos necesarios para el análisis.

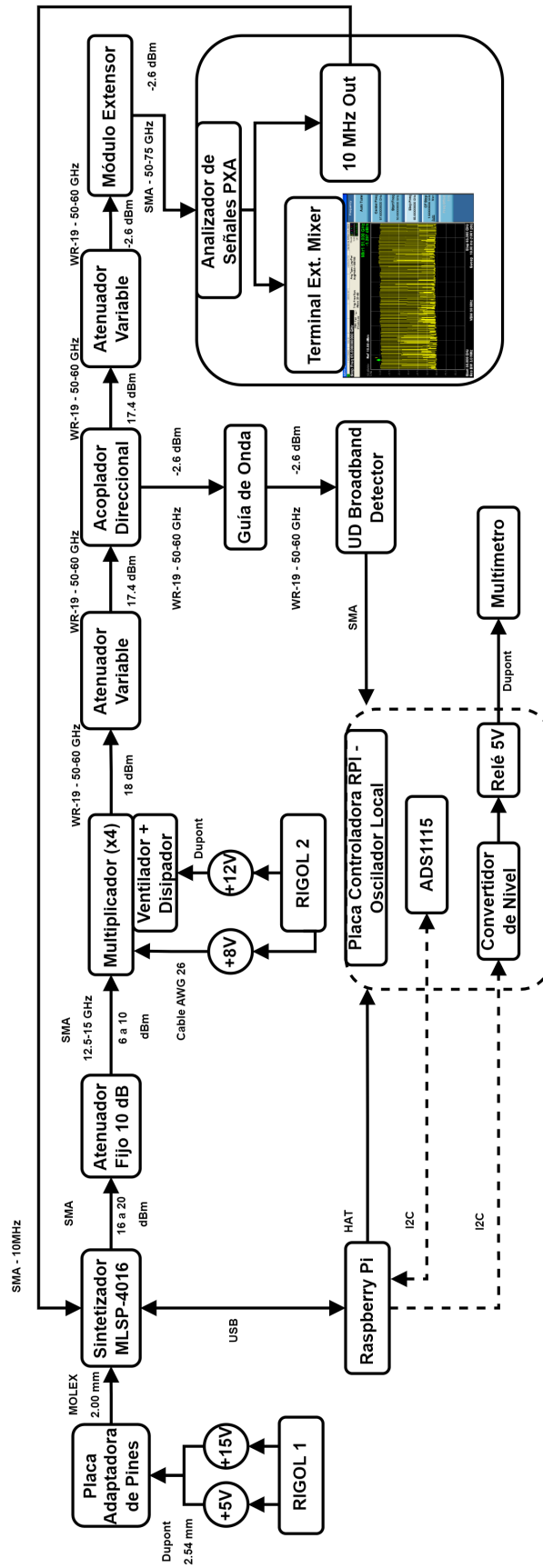


Figura 3.47: Diagrama de la configuración de mediciones utilizada para la banda 50 – 60 GHz del oscilador local.

Para este caso, debido a las limitaciones de potencia tanto del analizador de señales PXA como del módulo extensor, el atenuador variable que fue incorporado disminuye la potencia en 20 dB. De esta forma, la potencia que se muestra en el analizador de señales también está atenuada en 20 dB como el detector de potencia de banda ancha. Así, el resultado de potencia que se lee en ambos dispositivos (sistema de medición de potencia y analizador de señales PXA) debe ser idéntica.

En la siguiente figura 3.48, se muestra de manera física la configuración de mediciones para la presente banda.

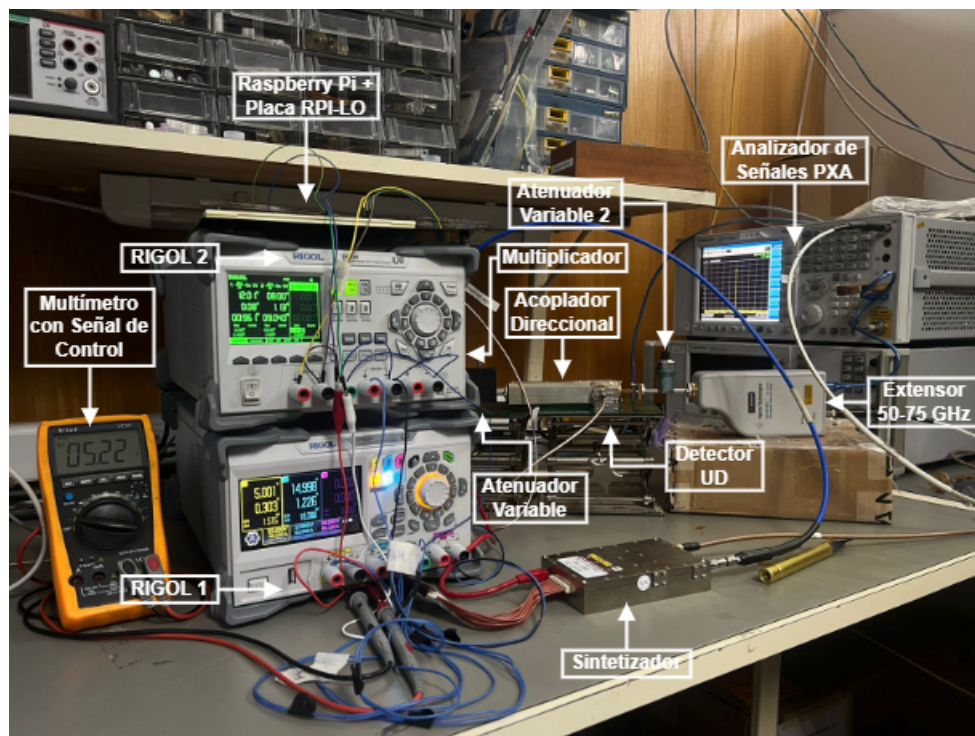


Figura 3.48: Configuración de mediciones utilizada para la banda 50 – 60 GHz para las pruebas del oscilador local.

Capítulo 4

Resultados y Discusión

En el presente capítulo se muestran y se discuten los resultados obtenidos, tanto para la implementación del sistema de control, como del oscilador local. Para cada una de las bandas, se incluyó el atenuador fijo de 10 dB debido a que se maximiza la potencia de salida del multiplicador para su banda de funcionamiento (ver sección 3.2.1.2). Es importante mencionar que el atenuador fijo se encontraba en la salida del sintetizador y antes del multiplicador (ver figuras 3.3.2.1 y 3.3.2.2). En resumen, se tiene un atenuador fijo (10 dB) y otro variable para el diagrama de configuraciones de la banda 36 – 50 GHz, mientras que para la banda de 50 – 60 GHz se tiene el mismo atenuador fijo, pero dos atenuadores variables.

En primer lugar, se muestran los resultados del funcionamiento del sistema de control. Para esto, se enseñan las mediciones obtenidas dentro del sistema de medición de potencia, y su correcta lectura de voltaje desde el detector de potencia de banda ancha. Además, se verifica que tanto en la ejecución de *rpiserver* como en la salida del sintetizador, se genere la frecuencia sintonizada. Por el lado del oscilador local, se muestran los resultados obtenidos a partir de la nueva calibración realizada. A partir de esta, se enseñan las mediciones de potencia de salida del sistema.

4.1. Sistema de Control

En términos del funcionamiento del sistema de control a lo largo de la implementación del oscilador local, se comentan a continuación los resultados obtenidos sobre el comportamiento de las etapas de medición de potencia y de sintonización de frecuencia. Además, dentro de esta última, se suman los resultados para el monitoreo del estado de los componentes.

Es importante recordar que, para ambas bandas, una de las principales variables de interés fue el comportamiento del multiplicador tanto con el atenuador fijo, como sin este dispositivo. Es por esto que, en los análisis que se muestran a continuación, se compara el comportamiento del sistema para ambos casos.

4.1.1. Medición de Potencia

Para la conversión de voltaje a potencia, es sumamente relevante el correcto funcionamiento del detector de potencia de banda ancha UD y del módulo conversor análogo a digital. Con esto, se asegura que los datos serán correctamente reunidos por el sistema de control. De la misma forma, el modelo que se utiliza para la equivalencia entre voltaje y potencia

es de gran importancia ya que, con este, es posible entregar al usuario el correcto valor de potencia de salida del oscilador local.

Se logró llevar a cabo el funcionamiento del sistema de medición de potencia a través del módulo conversor análogo a digital y del detector de potencia de banda ancha UD. Con esto, se logró medir el voltaje que es entregado por el detector, a lo largo de las bandas de 33.5 – 50 GHz y 50 – 64 GHz. Los resultados de las mediciones para la primera banda, los cuales fueron obtenidos con la configuración de mediciones de la figura 3.46, se muestran en el gráfico 4.1.

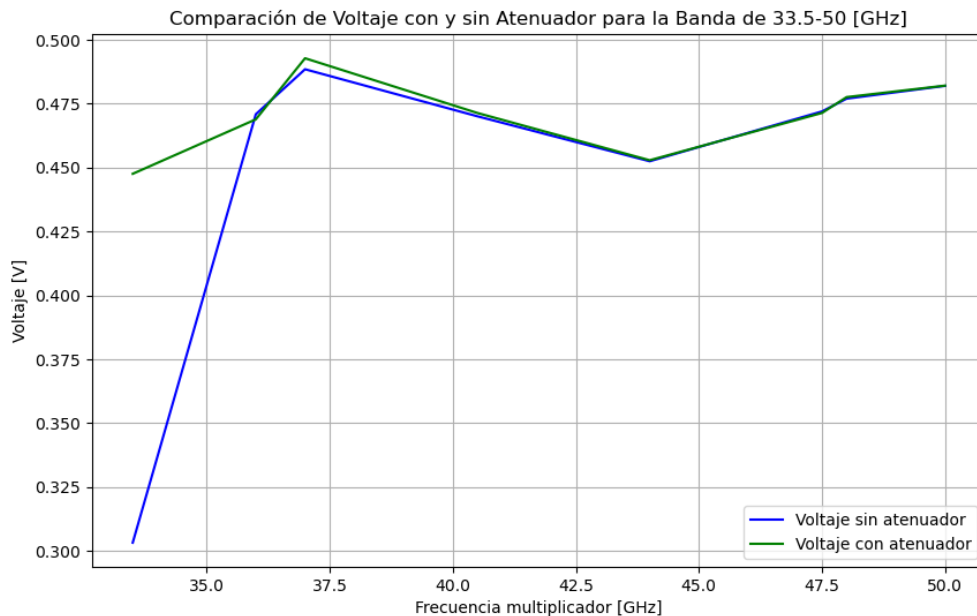


Figura 4.1: Medición de voltaje, con y sin atenuador, para la banda de 33.5 – 50 GHz.

En el gráfico 4.1, en el eje x se tiene la frecuencia del multiplicador, mientras que en el y, el voltaje leído por el módulo *ADS1115*. Las curvas muestran el comportamiento del voltaje, con y sin atenuador (fijo de 10 dB) en la configuración de mediciones. Así, se observa que las mediciones son poco afectadas por la presencia del componente y la baja medida del voltaje para 33.5 GHz, se debe a que se escapa del rango de operación del multiplicador (ver tabla 3.10).

Con ambos casos mostrados en la figura 4.1, se verifica que tanto el módulo conversor análogo a digital como el detector de potencia de banda ancha se encuentran operando correctamente. Además, muestra un valor de voltaje que varía entre 0.3 V y 0.48 V para el caso sin atenuador y, con el dispositivo, entre 0.45 V y 0.49 V.

Por otro lado, en la figura 4.2, se muestran las mediciones obtenidas para la banda de 50 – 64 GHz. En el eje x, se tiene la frecuencia del multiplicador, mientras que en el y, está el voltaje leído por el módulo *ADS1115*. Con esto, las curvas, muestran el comportamiento del voltaje, con y sin atenuador, en la configuración de mediciones de la figura 3.48.

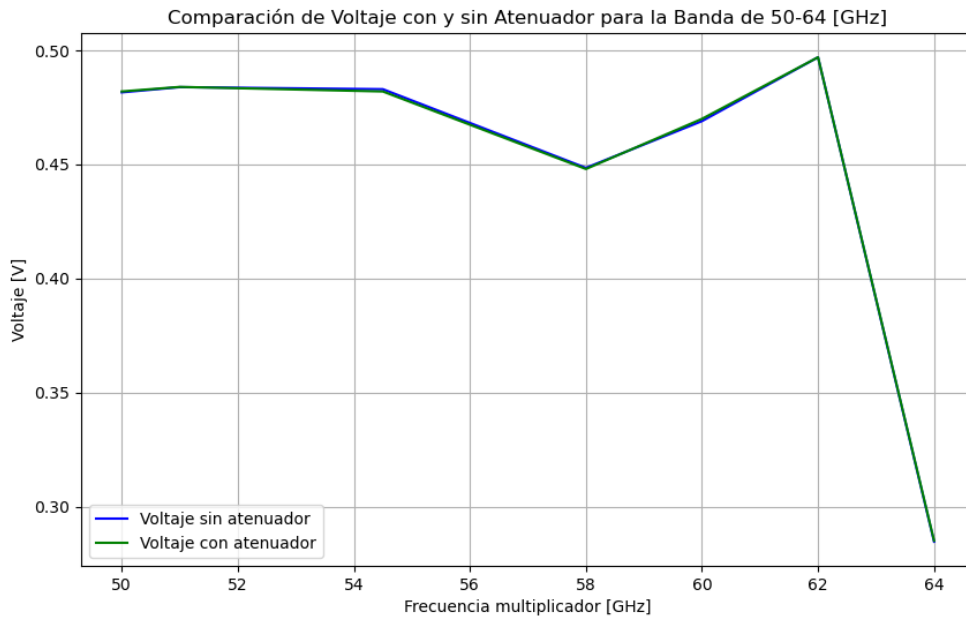


Figura 4.2: Medición de voltaje, con y sin atenuador, para la banda de 50 – 60 GHz.

Del gráfico 4.2, también se encuentra la comparación del voltaje que se mide con y sin atenuador. Se observa que las curvas azul (sin atenuador) y verde (con atenuador), son prácticamente idénticas y, por ende, las mediciones de voltaje son similares. Con ello, se infiere que la presencia del atenuador fijo dentro de la configuración de mediciones, no influye en la medición del voltaje. Sin embargo, al igual que en la figura 4.1, también se encuentra una caída en el nivel del voltaje para frecuencias superiores a 62 GHz, debido a que se escapa del rango de operación del multiplicador (ver tabla 3.10).

4.1.2. Sintonización de Frecuencia

Para la verificación del funcionamiento, se comprobó que el valor específico de frecuencia que era enviado desde el lado del cliente, no solamente fuera recibido en el servidor, o la terminal de la interfaz de *rpiserver*, sino que también fuera posible la visualización del espectro de la señal generada por el sintetizador, dentro del analizador de señales PXA.

En primer lugar, en la siguiente figura 4.3, se muestra el resultado que se obtiene cuando se define el valor de la frecuencia. A la izquierda, se muestran distintos casos en que se envió de manera correcta el mensaje, mientras que a la derecha, se enseña la respuesta que se entrega en la terminal de la interfaz de *rpiserver*. De esta forma, es importante notar que en el servidor (lado derecho de la figura 4.3), se reciben y se definen correctamente las frecuencias que serán enviadas al sistema de medición de potencia y al sintetizador.

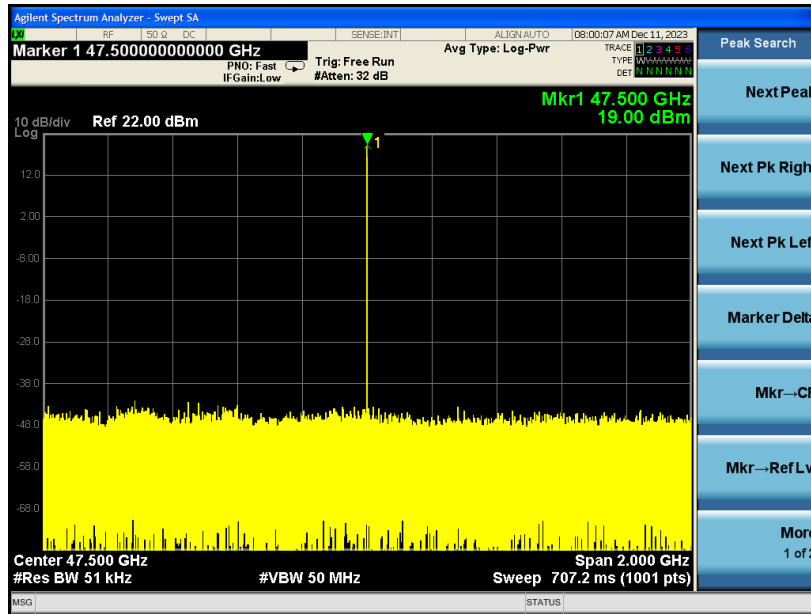
Cliente	Servidor
<pre> pi@raspberrypi:~/instmini/lab_tests/comms \$ sudo python3 client_rpi.py Frecuencia enviada 8375000000 [Hz] Respuesta del servidor: Frecuencia seteada en Sintetizador: 8375.0 [MHz] Frecuencia en Medidor de Potencia 33.5 [GHz] pi@raspberrypi:~/instmini/lab_tests/comms \$ sudo python3 client_rpi.py Frecuencia enviada 9250000000 [Hz] Respuesta del servidor: Frecuencia seteada en Sintetizador: 9250.0 [MHz] Frecuencia en Medidor de Potencia 37.0 [GHz] pi@raspberrypi:~/instmini/lab_tests/comms \$ sudo python3 client_rpi.py Frecuencia enviada 10125000000 [Hz] Respuesta del servidor: Frecuencia seteada en Sintetizador: 10125.0 [MHz] Frecuencia en Medidor de Potencia 40.5 [GHz] pi@raspberrypi:~/instmini/lab_tests/comms \$ sudo python3 client_rpi.py Frecuencia enviada 11000000000 [Hz] Respuesta del servidor: Frecuencia seteada en Sintetizador: 11000.0 [MHz] Frecuencia en Medidor de Potencia 44.0 [GHz] pi@raspberrypi:~/instmini/lab_tests/comms \$ sudo python3 client_rpi.py Frecuencia enviada 11875000000 [Hz] Respuesta del servidor: Frecuencia seteada en Sintetizador: 11875.0 [MHz] Frecuencia en Medidor de Potencia 47.5 [GHz] </pre>	<pre> ----- Información del Servidor ----- Conexión entrante de 192.168.2.123:41038 ----- Frecuencia recibida: 9250000000.0 [Hz] Frecuencia Sintetizador: 9250 [MHz] Frecuencia en Medidor de Potencia: 37.0 [GHz] ----- Información Sintetizador ----- f9250 Frecuencia del sintetizador fijada en: 9250 [MHz] Frecuencia enviada al sintetizador: f9250 Frecuencia del sintetizador fijada en: ----- Información Sintetizador ----- Comando de solicitud de temperatura enviado: T Respuesta inválida del dispositivo (temperatura): +46.3C ----- Información Sintetizador ----- Comando de solicitud de bits de estado enviado: ? Bits de estado del sintetizador: 11000011 B0: 100 MHz Locked (Internal Ref) B1: YIG PLL Locked B6: Self Test Passed B7: NOVO Locked (External Ref) RELÉ LED ON => +5V </pre> <pre> ----- Información del Servidor ----- Conexión entrante de 192.168.2.123:46986 ----- Frecuencia recibida: 11875000000.0 [Hz] Frecuencia Sintetizador: 11875 [MHz] Frecuencia en Medidor de Potencia: 47.5 [GHz] ----- Información Sintetizador ----- f11875 Frecuencia del sintetizador fijada en: 11875 [MHz] Frecuencia enviada al sintetizador: f11875 Frecuencia del sintetizador fijada en: ----- Información Sintetizador ----- Comando de solicitud de temperatura enviado: T Respuesta inválida del dispositivo (temperatura): +48.3C ----- Información Sintetizador ----- Comando de solicitud de bits de estado enviado: ? Bits de estado del sintetizador: 11000011 B0: 100 MHz Locked (Internal Ref) B1: YIG PLL Locked B6: Self Test Passed B7: NOVO Locked (External Ref) RELÉ LED ON => +5V </pre>

Figura 4.3: Terminal de cliente y servidor ante el envío y recepción de frecuencia.

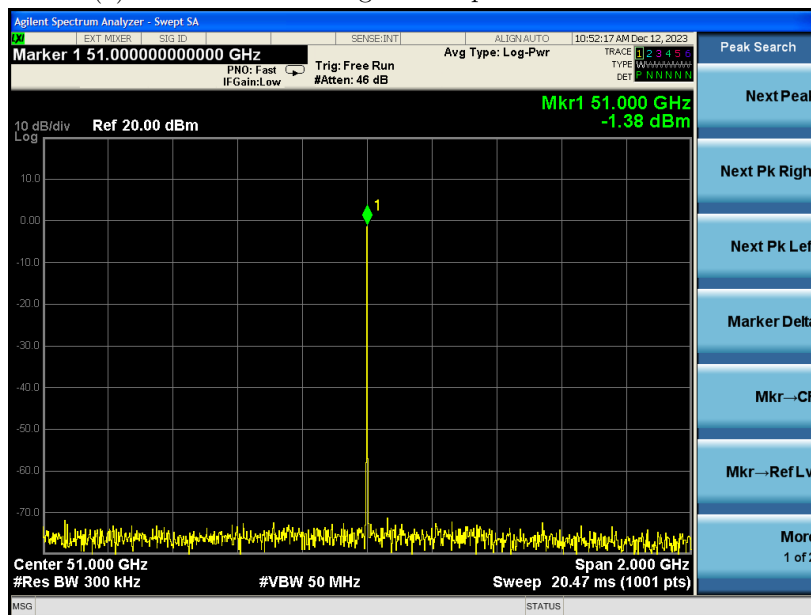
De esta forma, se observa en la figura 4.3 que, por ejemplo, si se envía desde el cliente 11875000000 Hz como frecuencia, entonces en el servidor se recibe el mismo valor en GHz, en MHz y multiplicado por cuatro. Este último está representado por “Frecuencia en Medidor de Potencia”, ya que es el valor de frecuencia que es ingresado al sistema de medición de potencia. Luego, como el envío y recepción de la frecuencia se realiza de manera exitosa, se comprueba en el analizador de señales el resultado de toda la cadena de oscilador local.

Además, es importante destacar de la figura 4.3, que en el lado del servidor *rpiserver*, se observa el estado de los componentes del sistema de control y del oscilador local. Con esto, se logra comprobar la temperatura y bits de estado del sintetizador.

En segundo lugar, en la figura 4.4, se muestra el espectro de la señal en el analizador de señales PXA, que es generada por el sintetizador. En la figura 4.4.a, se observa el espectro de la señal que fue producida en el ejemplo de la figura 4.3, la cual es de 47.5 GHz y tiene una potencia de 19 dBm. En la figura 4.4.b, se muestra una señal con frecuencia de 51.0 GHz que fue producida por el sintetizador con una señal de 12.75 GHz. En este caso, la potencia es de -1.38 dBm ya que fue atenuada 20 dB por el atenuador de la figura 3.47, el cual se encuentra entre el acoplador direccional y el módulo extensor.



(a) Señal de 47.5 GHz generada por el oscilador local.



(b) Señal de 51 GHz generada por el oscilador local.

Figura 4.4: Señales generadas por el oscilador local.

Con esto, se logró comprobar que la frecuencia que es enviada desde el cliente, es recibida correctamente tanto en el servidor, en donde se define el valor que se ingresa al sistema de medición de potencia (ver figura 4.3), como en el sintetizador, en donde se comprueba en el analizador de señales PXA que la frecuencia es multiplicada de manera correcta (ver figura 4.4).

4.2. Oscilador Local

Dentro de las pruebas que fueron realizadas con el oscilador local, es importante volver a destacar la restricción relacionada a los equipos presentes en el laboratorio. Como no se

cuenta con un analizador de señales para el rango de 36 – 60 GHz, fue necesario separar en dos bandas diferentes: 36 – 50 GHz y 50 – 60 GHz. Consecuentemente, la configuración de las mediciones es levemente diferente para cada banda y puede ser encontrada en las figuras 3.45 y 3.47.

Para cada una de las configuraciones de mediciones de las bandas, la potencia leída por el detector UD corresponde a la potencia entregada por la salida acoplada del acoplador direccional, que está atenuada en 20 dB. En el caso de la banda de 36 – 50 GHz, la potencia que entrega el analizador de señales PXA es disminuida en 20 dB para compararla con la potencia del detector UD. Para el caso de la banda de 50 – 60 GHz, la potencia entregada por el analizador se encuentra disminuida en 20 dB, como se muestra en la configuración de mediciones de la figura 3.47, por lo que no es necesario volver a disminuirla. Además, la medición de potencia entregada por el oscilador local corresponde a la que se obtiene desde el analizador de señales PXA y que proviene desde la salida restante del acoplador direccional (ver figura 3.3).

4.2.1. Sensibilidad del Detector UD

4.2.1.1. Calibración

A partir de los resultados obtenidos con el modelo de conversión de voltaje a potencia presentados en la sección 3.3.1.3, se logró medir y comparar la potencia del analizador de señales PXA y detector de potencia de banda ancha UD, pero estos presentaron un significativo nivel de diferencia, alrededor de 5 dB. Con el “Modelo 1”, aquel presentado en la sección 3.3.1.3, se midió la potencia para el rango de frecuencias 33.5 GHz y 48 GHz, tanto con el atenuador (fijo de 10 dB) como sin él. En la siguiente figura 4.5, se muestran los resultados de las mediciones obtenidas.

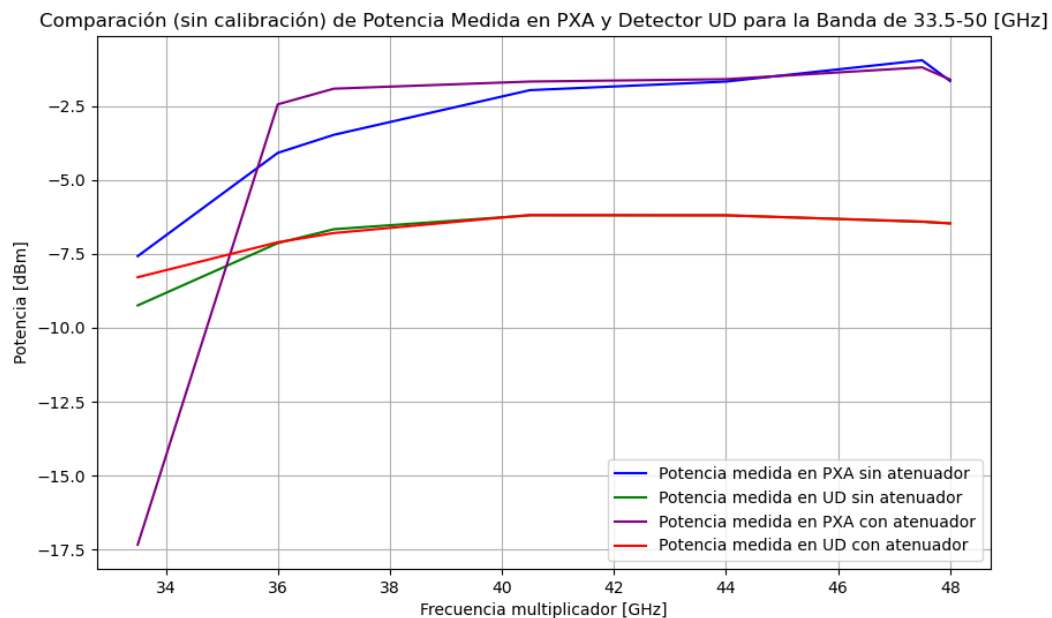


Figura 4.5: Potencia medida, con y sin atenuador, en analizador de señales PXA y detector UD.

En el gráfico de la figura 4.5, se tiene en el eje x la frecuencia del multiplicador, mientras que en el eje y se observa el nivel de potencia medido tanto en el analizador de señales PXA como en el detector de potencia de banda ancha. Las curvas de color azul y verde muestran las mediciones de potencia, sin atenuador, en el analizador de señales PXA y detector UD, respectivamente. A partir de estas, se observa que existe una diferencia, aproximadamente de 5 dB, entre las mediciones. De la misma forma, se puede llegar a la misma conclusión si es que se toman en cuenta las curvas de color morado y rojo, las cuales representan las mediciones de potencia en el analizador PXA y detector UD, respectivamente.

Por otro lado, si se observan las curvas de potencia medida en el analizador PXA, con y sin atenuador, representadas por el color morado y azul, respectivamente, se concluye que la presencia del atenuador no influye puesto que las curvas son similares. De la misma forma, si se toman en cuenta las curvas de potencia medida en el detector UD, con y sin atenuador, representadas por color rojo y verde, respectivamente, se llega a la misma conclusión mencionada anteriormente.

En la siguiente tabla 4.1, se muestran los valores para el promedio, media, desviación estándar y varianza de las mediciones. La tabla muestra la diferencia de medición de potencia entre cuatro diferentes casos. Por ejemplo, el valor de 0.9 dB equivale al promedio de la diferencia de potencia entre la medición obtenida en el analizador PXA con atenuador y de la medición obtenida en el analizador PXA sin atenuador.

Tabla 4.1: Resultados estadísticos para las mediciones de potencia con el Modelo 1.

Diferencia de Medición de Potencia	Potencia [dB]			
	Promedio	Media	Desviación Estándar	Varianza
Analizador PXA con atenuador y Analizador PXA sin atenuador	0.9	-0.08	3.67	13.51
Detector UD con atenuador y Detector UD sin atenuador	-0.12	-0.001	0.34	0.11
Analizador PXA sin atenuador y Detector UD sin atenuador	3.38	4.23	1.19	1.42
Analizador PXA con atenuador y Detector UD con atenuador	2.81	4.66	4.84	23.44

Como se observa en la tabla 4.1, el promedio de la diferencia de medición de potencia con el analizador de señales PXA, con y sin atenuador, es de 0.9 dB, mientras que para el caso del detector de potencia de banda ancha UD, es de $-0,12$ dB. Debido a esto, no existen cambios significativos en la medición de potencia cuando se incluye o no el atenuador, tanto para el analizador de señales PXA como para el detector de potencia de banda ancha UD.

Por otro lado, en la misma tabla 4.1, se observa que el promedio de la diferencia de medición de potencia entre el analizador PXA y el detector UD, sin atenuador es de 3.38 dB, mientras que para el caso con atenuador, es de 2.81 dB. Debido a que el promedio de la

diferencia de potencia para cada caso, que debería ser idéntica, es mayor a 2 dB, se decidió realizar una nueva calibración y por ende, un nuevo modelo para realizar la conversión de voltaje a potencia.

La nueva conversión, denominada como “Modelo 2” utiliza nuevos datos de frecuencia y sensibilidad para estimar la potencia. En lugar de utilizar los datos del fabricante, que fueron mostrados en $data_{freq}$ y $data_{sens}$ de la sección 3.3.1.3, se definió un intervalo de frecuencias y, con esto, se recopilaron los datos de voltaje y potencia en el analizador PXA. Los datos reunidos se muestran a continuación:

$$data_{freq2}[GHz] = [36, 40, 45, 50, 52.5, 55, 57.5, 60] \quad (4.1)$$

$$data_{volt-ud}[V] = [0.468, 0.474, 0.462, 0.483, 0.478, 0.475, 0.454, 0.471] \quad (4.2)$$

$$data_{pot-pxa-dbm}[dBm] = [16.22, 17.63, 18.48, 18.77, 19.2, 19.3, 18.67, 18.89] \quad (4.3)$$

Los datos que se muestran en 4.1, indican los valores de frecuencia que fueron tomados. En 4.2, se encuentran los valores de voltaje que fueron obtenidos desde el detector UD y en 4.3, se tienen los valores de potencia, en dBm, que fueron tomados desde el analizador de señales PXA. Con estos datos, se realizó el proceso inverso de las ecuaciones mostradas en la sección 3.3.1.3, para la transformación de voltaje a sensibilidad. Así, la conversión queda en función de la siguiente ecuación:

$$pot_{dBm-20}[dBm] = data_{pot-pxa-dbm} - 20 \quad (4.4)$$

$$pot_{mW}[mW] = 10^{(pot_{dBm-20}/10)} \quad (4.5)$$

$$sensibilidad[mV/mW] = (data_{volt-ud} \cdot 1000) / pot_{mW} \quad (4.6)$$

En 4.4, se tienen los valores de potencia, en dBm, disminuidos en 20 dB. En 4.5, se indican los valores de potencia en mW y, finalmente, en 4.6, se encuentran los valores de sensibilidad. Con esto, se vuelve a realizar el ajuste polinómico e interpolación PCHIP de los datos, pero ahora con la nueva sensibilidad y frecuencia. Con estos nuevos valores, el resultado de la nueva equivalencia entre frecuencia y sensibilidad, se muestra en la siguiente figura 4.6.

En la figura 4.6, en el eje x se observa la frecuencia, mientras que en el eje y, se tiene la sensibilidad. Con esto, los puntos de color azul corresponden a los datos originales obtenidos y con ellos, se realiza una comparación entre un ajuste polinómico de grado cuatro e interpolación PCHIP, lo cual se observa en las curvas de color verde y rojo, respectivamente.

En consecuencia, si se comparan los valores de la tabla 4.1 del Modelo 1, con los resultados obtenidos de la banda de 33.5 – 50 GHz de la tabla 4.2, con el Modelo 2, se logra observar que el cambio de modelos realizado mejora la medición de la potencia, haciendo que la diferencia entre las mediciones de los equipos sea menor.

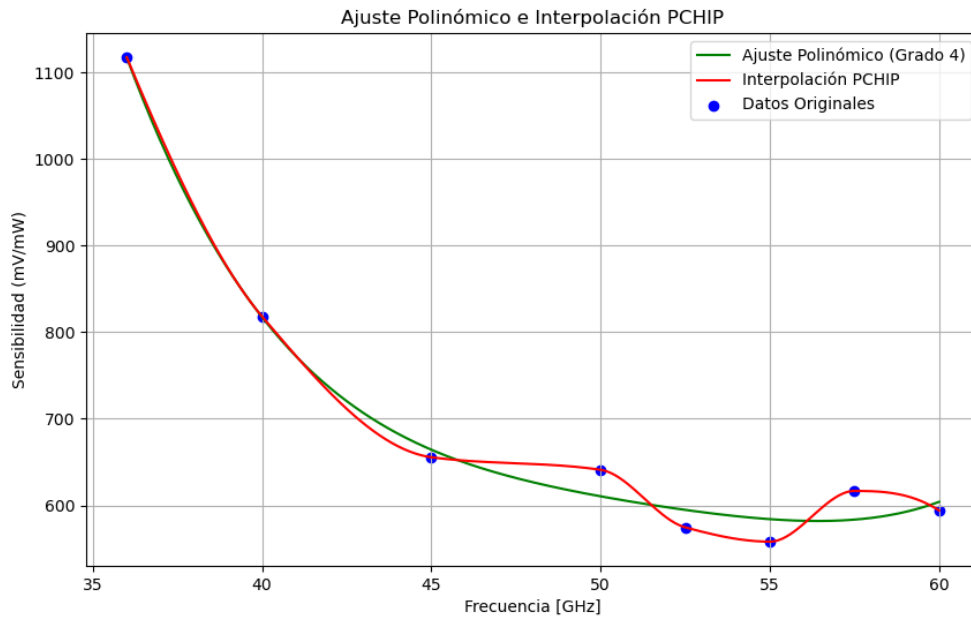


Figura 4.6: Ajuste polinómico e interpolación PCHIP con el Modelo 2.

Para el caso de la diferencia de medición de potencia en el analizador de señales PXA, con y sin atenuador, se logra disminuir el promedio desde 0.9 dB hasta 0.87 dB, mientras que en el detector UD, aumenta levemente desde -0.12 dB hasta -0.14 dB. Con esto, se sigue manteniendo que el atenuador no influye en la salida de la potencia del multiplicador. Por otro lado, para el caso de la diferencia de medición de potencia entre el analizador de señales PXA y el detector UD, sin atenuador, se logra disminuir el promedio de la diferencia de potencias desde 3.38 dB hasta 0.04 dB. Mientras que cuando se tiene el atenuador, el promedio de la diferencia de potencias disminuye desde 2.81 dB hasta -0.97 dB.

Con lo anterior, se logra que el sistema de medición de potencia tenga una mejor precisión y entregue un valor más acercado a la realidad, es decir, que la diferencia de potencia entre el analizador de señales y el detector UD, sea nula.

4.2.2. Mediciones de Potencia del Sistema para 36-50 GHz

Para la toma de datos de la banda actual, se modificó levemente el rango de sintonización a 33.5 – 50 GHz, debido a que de esta forma, se visualiza un mayor rango de frecuencias que puede generar el sintetizador. Los valores de potencia recopilados corresponden a aquellos entregados por el analizador de señales PXA y detector UD, para los casos con y sin atenuador. En la siguiente figura 4.7 se muestran los resultados obtenidos.

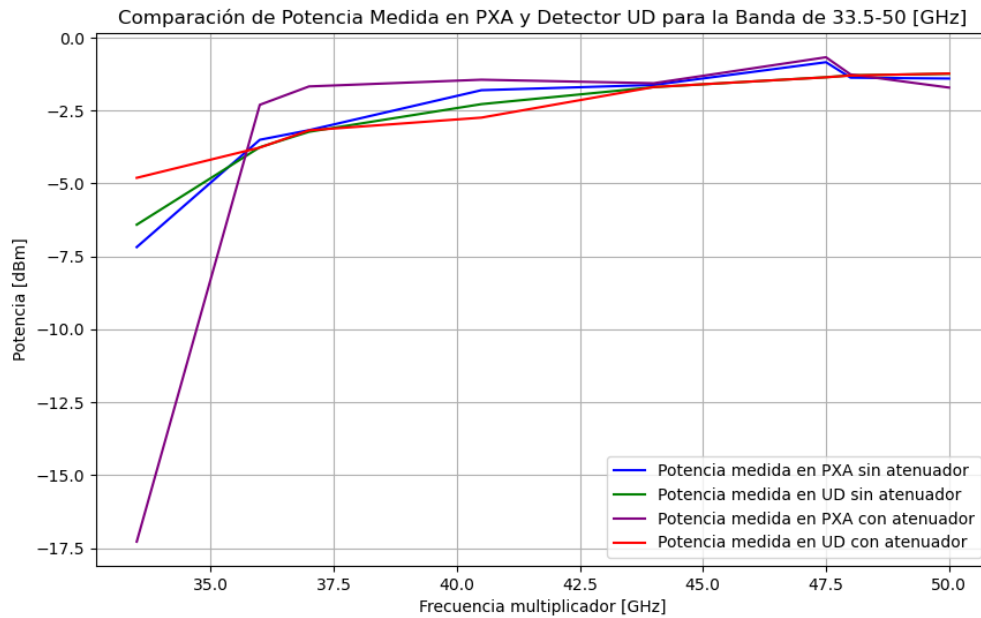


Figura 4.7: Potencia obtenida en Analizador PXA y Detector UD, con y sin atenuador, para la banda de 33.5 – 50 GHz.

En el gráfico de la figura 4.7, en el eje x se tiene la frecuencia del multiplicador, mientras que en el y, se tiene la potencia medida en el analizador de señales PXA y en el detector UD. Además, las curvas de color azul y morado muestran la potencia medida en el analizador PXA, sin y con atenuador, respectivamente. En las curvas de color verde y rojo, se observa la potencia medida en el detector UD, sin y con atenuador, respectivamente.

En primer lugar, se observa que la inclusión del atenuador fijo de 10 dB no afecta en gran medida a la potencia de salida del sistema, tanto para los casos del analizador PXA como en el detector UD. En ambos casos, se observa que la diferencia de potencia es menor a 1 dB, lo cual es similar a lo presentado en la tabla 4.1. Para el segundo caso, de la diferencia de medición de potencia entre el analizador PXA y detector UD, tanto sin y con atenuador, el valor también es menor a 1 dB y, por ende, se concluye que la medición tiene mayor precisión con el Modelo 2 que con el Modelo 1.

De la figura 4.7, es importante destacar que la potencia obtenida en la curva morada, para 33.5 GHz, es de -17.5 dB, ya que tal valor de frecuencia se encuentra fuera del rango de operación del multiplicador. En la siguiente tabla 4.2 se muestra el promedio, media, desviación estándar y varianza de los resultados obtenidos.

Tabla 4.2: Resultados estadísticos para las mediciones de potencia para la banda de 33.5 – 50 GHz, con el Modelo 2.

Diferencia de Medición de Potencia	Potencia [dB]			
	Promedio	Media	Desviación Estándar	Varianza
Analizador PXA con atenuador y Analizador PXA sin atenuador	0.87	-0.14	3.53	12.47
Detector UD con atenuador y Detector UD sin atenuador	-0.14	-0.0006	0.57	0.32
Analizador PXA sin atenuador y Detector UD sin atenuador	0.04	0.06	0.38	0.14
Analizador PXA con atenuador y Detector UD con atenuador	-0.97	0.41	4.39	19.35

De la tabla anterior, la diferencia que existe entre las medidas de potencia del analizador PXA, con y sin atenuador, no es considerable para concluir que el dispositivo afecta al sistema. Sin embargo, debido a los gráficos 4.2 y 4.5, se observa que el funcionamiento del multiplicador en los extremos de la banda, es mejor sin atenuador.

4.2.3. Mediciones de Potencia del Sistema para 50-60 GHz

Para la toma de datos de la banda actual, se modificó levemente el intervalo de frecuencias a 50 – 64 GHz, ya que de esta forma se visualiza un mayor rango de frecuencias que puede generar el sintetizador. Los valores de potencia recopilados corresponden a aquellos entregados por el analizador de señales PXA y detector de potencia de banda ancha UD, para los casos con y sin atenuador. Además, es importante mencionar que las lecturas del analizador fueron atenuadas en 20 dB por el segundo atenuador variable de la figura 3.47. En la siguiente figura 4.8, se observa el resultado de las mediciones.

Del gráfico 4.8, se observa que las curvas morada y azul, que representan la lectura de potencia en el analizador PXA con y sin atenuador (fijo de 10 dB), respectivamente, permanecen prácticamente idénticas a lo largo de toda la banda. Como máximo, la diferencia es menor a 0.06 dB. Lo mismo ocurre para el caso del Detector UD, donde las curvas verde y roja resultan prácticamente idénticas. Como máximo, la diferencia es menor a 0.002. Además, estas curvas son idénticas debido a que el segundo atenuador variable (ver figura 3.47) no afecta al detector UD.

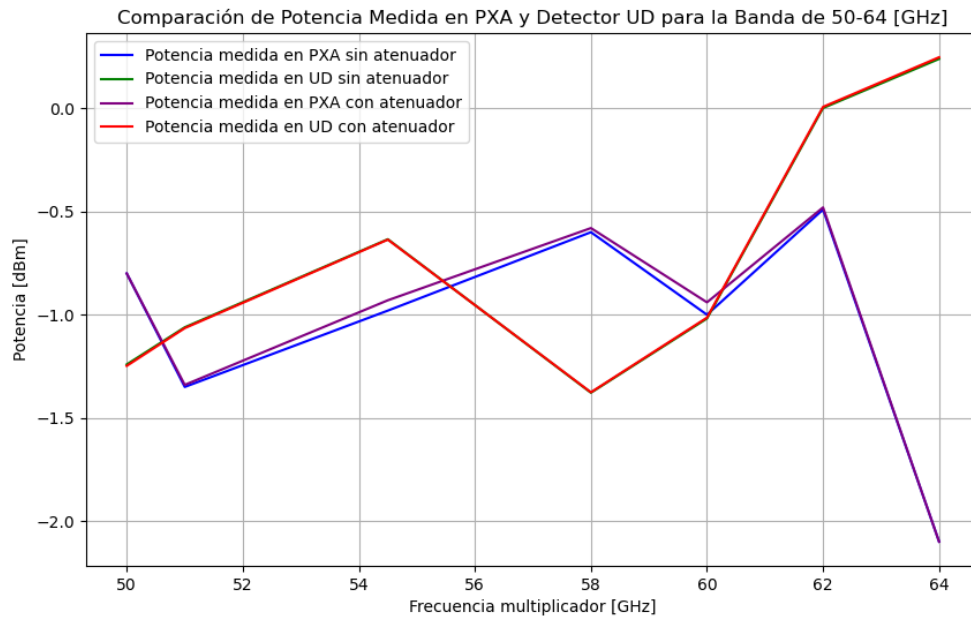


Figura 4.8: Potencia obtenida en Analizador PXA y Detector UD, con y sin atenuador, para la banda de 50 – 64 GHz.

La comparación de las mediciones entre los equipos analizador de señales PXA y detector UD, presentan diferencias significativamente leves. En la siguiente tabla 4.3, se muestra el promedio, media, desviación estándar y varianza de los resultados obtenidos, en donde se visualizan de mejor manera las diferencias mencionadas.

Tabla 4.3: Resultados estadísticos para las mediciones de potencia para la banda de 50 – 64 GHz, con el Modelo 2.

Diferencia de Medición de Potencia	Potencia [dB]			
	Promedio	Media	Desviación Estándar	Varianza
Analizador PXA con atenuador y Analizador PXA sin atenuador	-0.02	-0.01	0.02	0.0004
Detector UD con atenuador y Detector UD sin atenuador	-0.001	-0.002	0.005	$2.54 \cdot 10^{-5}$
Analizador PXA sin atenuador y Detector UD sin atenuador	-0.31	-0.28	0.92	0.85
Analizador PXA con atenuador y Detector UD con atenuador	-0.29	-0.27	0.93	0.87

A partir de la tabla 4.3, se logra confirmar lo mencionado anteriormente. Se observa que el promedio de la diferencia de potencia en el analizador PXA, con y sin atenuador, es de -0.02 dB, mientras que para el detector UD, es de -0.001 dB. De esta forma, la inclusión del atenuador no es relevante para el sistema.

Luego, se observa que el promedio de la diferencia de potencia entre el analizador PXA y detector UD, ambos sin atenuador, es de -0.31 dB. Para el caso del analizador PXA y detector UD pero con atenuador, el promedio de la diferencia es de -0.29 dB. De esto, se destaca que la diferencia para ambos casos es cercana a cero, lo que equivale decir que, la potencia estimada por el sistema de medición de potencia es cercana a la que realmente se entrega desde el oscilador local. Además, si se analiza el efecto del atenuador, se observa que la diferencia entre los promedios es de 0.02 dB. Es por esto que nuevamente se concluye que la inclusión del atenuador fijo de 10 dB no es considerable.

4.2.4. Sensibilidad del Sistema de Medición de Potencia

Una de las mediciones de potencia que se realizó, fue el análisis de la sensibilidad del sistema de medición de potencia. Para esto, se fijó una frecuencia en el sintetizador y, luego, se recopilaron datos sobre la potencia leída en el analizador de señales PXA y en el detector de potencia de banda ancha. Allí, a través del atenuador variable de la figura 3.47 que se encuentra luego del multiplicador y antes del acoplador direccional, se fue atenuando la potencia de la señal en una unidad.

La toma de datos se realizó con la configuración de mediciones de la figura 3.47, la frecuencia fue fijada en 50 GHz y se atenuó la potencia entre un rango de $0 - 20$ dB. A continuación, en la figura 4.9, se muestra el gráfico de sensibilidad que fue obtenido.

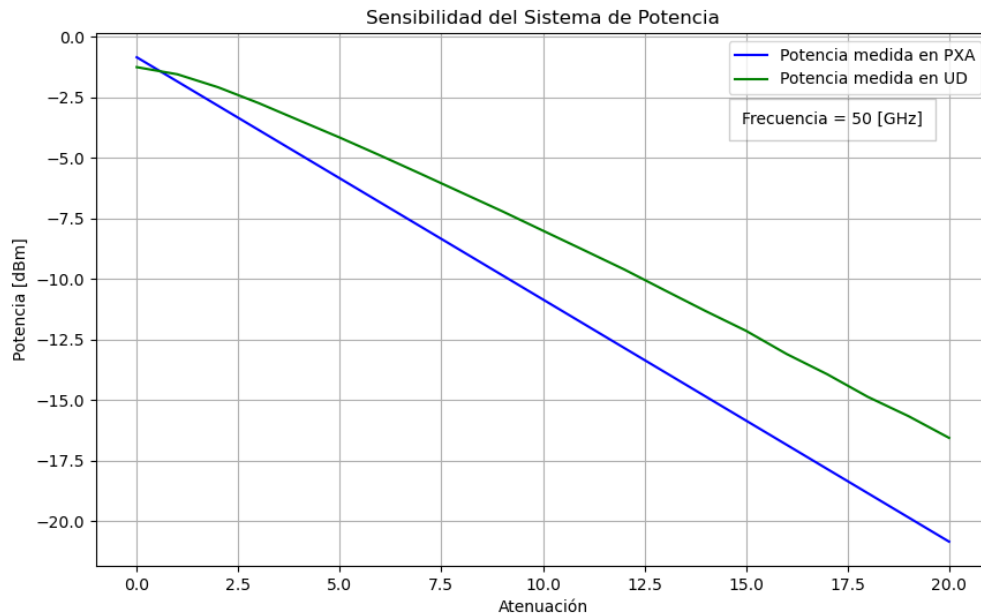


Figura 4.9: Atenuación y Potencia, para una frecuencia de 50 GHz.

En la figura 4.9, se tiene que el eje x representa el nivel de atenuación y el eje y, representa la potencia medida. Además, se puede observar que la curva de color azul representa la potencia medida en el analizador de señales PXA, mientras que la de verde es la potencia

medida en el detector de potencia de banda ancha UD.

El resultado esperado de la presente prueba, consiste en que ambas curvas deben ser idénticas, puesto que tal es el objetivo del sistema de medición de potencia. Esto quiere decir que la potencia estimada por el sistema de medición debe ser similar a la que entrega el oscilador local. Sin embargo, se puede observar de la figura 4.9, que las curvas comienzan a separarse con una diferencia mayor 2 dB a partir de un nivel de atenuación mayor a 6 dB. Para un nivel de atenuación de 20 dB, se observa que la diferencia entre las mediciones es de 4.2 dB.

Además, se observa que la curva del detector UD tiene un comportamiento decreciente a medida que la atenuación aumenta. También, se observa que entre el intervalo de atenuación de 0 y 2.5 dB, el detector UD es menos sensible ante las variaciones de atenuación y que existe una menor diferencia de potencia, de alrededor de 0.6 dB, entre la potencia medida en el analizador de señales PXA y detector UD. Esto quiere decir que, cuando la atenuación es menor a 2.5 dB, se tiene como máximo una diferencia de 0.6 dB para la estimación de la potencia que entrega el oscilador local. Sin embargo, a medida que se aumenta la atenuación, los niveles de precisión disminuyen hasta casi 4.2 dB.

Capítulo 5

Conclusiones

En la presente memoria, se trabajó en el sistema de control y medición de potencia de un oscilador local, con el objetivo de que posteriormente sea integrado en el bloque de recepción del *Southern Millimeter-Wave Telescope*. Esto permite el cumplimiento de una de las etapas del proyecto, el cual busca actualizar gran parte de la instrumentación que se encuentra operando actualmente en el radiotelescopio.

El nuevo oscilador local se conforma tanto por componentes de radio frecuencias, como elementos del sistema de control. Dentro de la primera categoría, se encuentra un sintetizador, multiplicador, atenuador variable, acoplador direccional y detector de potencia de banda ancha UD. Para la segunda categoría, los componentes del sistema de control, se utilizó un *Raspberry Pi* y se diseñó una placa personalizada que integra diferentes dispositivos de electrónica. De esta forma, el sistema fue implementado mediante la placa y una interfaz de línea de comandos, la cual establece una comunicación bidireccional con el sintetizador, el monitoreo del estado de los componentes y la medición de la potencia de salida del sistema.

El presente trabajo se dividió en tres etapas. En primer lugar, se realizó la descripción de los elementos del oscilador local, en donde se tienen componentes de radio frecuencias y del sistema de control. Por un lado, esto es sumamente relevante para la generación de documentación de cada dispositivo, puesto que se entrega material disponible para que el trabajo futuro sea más completo y fluido. Por otro lado, el conocimiento del funcionamiento y parámetros de operación de los instrumentos utilizados resulta valioso tanto para el diseño de las etapas posteriores, como la posible incorporación de otros equipos.

A lo largo de la segunda etapa, se realizó el diseño, implementación y pruebas del sistema de control del oscilador local. En términos de *software*, se llevaron a cabo dos interfaces, siendo *synthesizer.c* para el control e interacción en específico con el sintetizador, mientras que con *rpiserver.py* se permite el control, monitoreo y medición de potencia del oscilador local. Desde el punto de vista de *hardware*, se diseñaron dos placas personalizadas. Una de ellas permite la alimentación y comunicación con el sintetizador, además de distribuir voltaje para su ventilación y la del multiplicador. La otra de ellas, integra y minimiza el espacio que utilizan todos los componentes del sistema de control.

Dentro de la última etapa, se llevó a cabo la implementación del oscilador local en conjunto de su sistema de control. En términos del control, se logró el correcto funcionamiento

del sistema de medición de potencia y su adquisición de voltaje, además de la sintonización de frecuencia y monitoreo de los componentes. En consecuencia, se confirma la elección de los componentes escogidos: módulo *ADS1115*, conversor de nivel y relé. Con esto, se entrega un dispositivo capaz de obtener datos y trabajar con ellos a partir del detector de potencia de banda ancha, además de que sea posible establecer una comunicación bidireccional con el sintetizador.

Por el lado del oscilador local, se logró que este generara las frecuencias deseadas con suma precisión, siendo análoga a la que entrega el sintetizador. Con esto, el oscilador local cumple el rango de frecuencias definido inicialmente: 67 – 116 GHz. Además, es posible medir la potencia de salida del oscilador mediante el sistema de control y medición de potencia que fue implementado. Si bien el valor de potencia que se calcula presenta diferencias (ver figura 4.9) con respecto a la potencia real, se mejoró la precisión en base a los modelos de estimación realizados y, por ende, se logró reducir la diferencia en 5 dB. Con esto, es relevante mencionar que es posible seguir mejorando la precisión de la estimación de la potencia mediante la toma de nuevos datos a distintas frecuencias y el ajuste de las curvas en base a la información recolectada.

En conclusión, con el trabajo realizado se lograron cumplir los objetivos planteados en el inicio. Por su parte, en el presente informe se entrega documentación actualizada y de suma relevancia para el historial del *SMWT*. Se recopiló información que refleja antecedentes generales de los bloques actuales del radiotelescopio y, en particular, de su oscilador local. Con esto, el presente documento se constituye de información sumamente relevante para la comprensión del funcionamiento, dispositivos y operación del nuevo oscilador local y de su sistema de control.

5.1. Trabajo Futuro

Tras la finalización del trabajo relacionado a la presente memoria, es importante mencionar que los objetivos planteados en el inicio fueron completados. Aún así, se encuentra un margen de posibles mejoras y, de esta manera, se plantean las siguientes tareas que se pueden realizar con posterioridad:

- Probar la nueva placa del sistema de control RPI-LO del Anexo E.
- Afinar las mediciones de potencia del detector UD. Como existe un margen de mejora para la conversión de voltaje a potencia, es posible elaborar un nuevo modelo con mayor precisión.
- Diseñar la placa de oscilador local para el nuevo bloque de recepción del *SMWT*, la cual abarca los componentes de radio frecuencias y de su sistema de control.
- Integrar el oscilador local al sistema de control del radiotelescopio. Con esto se espera que exista una comunicación fluida y continua para que se sintonice la frecuencia deseada.
- Ampliar el alcance del sistema de control del oscilador local. Para esto, se debe incorporar la retroalimentación que conlleva la experiencia de usuario con la interfaz. Así, se planea añadir nuevos comandos con nuevas funcionalidades, u otros elementos para el monitoreo.

Bibliografía

- [1] H. Karttunen, P. Kröger, H. Oja, M. Poutanen y K. J. Donner, *Fundamental Astronomy*. Springer-Verlag: Springer, 5 ed., 2006.
- [2] M. Albert, *Quantum Mechanics*. Mineola, New York: Mineola, N.Y.: Dover Publications, 1 ed., 1999.
- [3] R. Feynman, “The Feynman Lectures on Physics.” Disponible en <https://www.feynmanlectures.caltech.edu/> (24/01/2024).
- [4] T. L. Wilson, K. Rohlfs y S. Hüttemeister, *Tools of Radio Astronomy*. Springer-Verlag: A&A Library, 5 ed., 2009.
- [5] L. Bronfman, H. Alvarez, R. S. Cohen y P. Thaddeus, “*A deep CO survey of molecular clouds in the southern Milky Way*,” 1989.
- [6] D. M. Pozar, *Microwave Engineering*. Wiley: John Wiley & Sons, Inc, 4 ed., 2012.
- [7] A. R. Thompson, J. M. Moran and G. W. Swenson Jr, *Interferometry and Synthesis in Radio Astronomy*. Springer Open: A&A Library, 3 ed., 2017.
- [8] Pacific Millimeter, “Broadband Detectors.” Disponible en <http://www.pacificmillimeter.com/Detectors.html> (14/12/2023).
- [9] Harvard Smythsonian Center for Astrophysics, “Millimeter-Wave Group.” Disponible en <https://lweb.cfa.harvard.edu/mmw/index.html> (16/05/2023).
- [10] W. Max-Moerbeck, “*Implementación de un Oscilador Gunn en un Receptor a 115GHz, para fines Radioastronómicos*,” Memoria para optar al título de Ingeniero Civil Electricista. Universidad de Chile, 2005.
- [11] N. Reyes, “*Integración física de un amplificador de bajo ruido a un receptor en 100 GHz*,” Memoria para optar al título de Ingeniero Civil Electricista. Universidad de Chile, 2006.
- [12] P. Vásquez, “*Instalación y puesta en marcha del radiotelescopio Mini*,” Memoria para optar al título de Ingeniero Civil Electricista. Universidad de Chile, 2011.
- [13] D. Monasterio, “*Diseño de un triplicador de frecuencia de 35 a 105 GHz basado en diodos schottky*,” Memoria para optar al título de Ingeniero Civil Electricista. Universidad de Chile, 2012.
- [14] P. Astudillo, “*Medición del patrón de radiación del telescopio Mini*,” Memoria para optar al título de Ingeniero Civil Electricista, 2014.
- [15] R. Finger, “*Design and construction of a digital sideband separating spectrometer for the 1.2-meter Southern Radio Telescope*,” PhD. Thesis, 2013.
- [16] R. Rodríguez, “*Design, construction and testing of a 2SB receiver for the Southern Millimeter-Wave Telescope*,” PhD. Thesis, 2015.

- [17] D. Monasterio, “Documentación técnica de la placa de oscilador local para el nuevo receptor del radiotelescopio mini.” Documento Interno.
- [18] APEX, “SHeFI Comparison Spectra.” Disponible en <https://www.apex-telescope.org/heterodyne/shfi/calibration/standards/> (16/05/2023).
- [19] ESO Website, “ALMA Receiver Bands.” Disponible en <https://www.eso.org/public/teles-instr/alma/receiver-bands/> (16/05/2023).
- [20] D. Monasterio, “*Development of new heterodyne-downconverter architectures for the next generation of radio astronomy instrumentation*,” PhD. Thesis, 2022.
- [21] P. Mirtschin, “Australia Telescope Compact Array.” Disponible en <http://astro.vaporia.com/start/atca.html#:~:text=It%20consists%20of%20six%202022,resolution%20direct%20ion%20and%20wavelength>. (16/05/2023).
- [22] P. Mirtschin, “Mopra Telescope.” Disponible en <https://www.narrabri.atnf.csiro.au/mopra/obsinfo.html> (16/05/2023).
- [23] K. Muraoka et al., “Nobeyama Radio Observatory.” Disponible en <https://ui.adsabs.harvard.edu/abs/2013ASPC..476..417M/abstract> (16/05/2023).
- [24] Metsähovi Radio Observatory, “Metsähovi Radio Observatory Receiver.” Disponible en <https://www.metsahovi.fi/receiver/> (16/05/2023).
- [25] IRAM, “30 meter Telescope - IRAM.” Disponible en <https://iram-institute.org/observatories/30-meter-telescope/> (16/05/2023).
- [26] R. D. Norrod, “Green Bank Telescope.” Disponible en <https://www.gb.nrao.edu/4mm/ProjectBook/Chapter3/chapter3.shtml> (16/05/2023).
- [27] Millimeter-Wave Telescope Group, “1.2 Meter Millimeter-Wave Telescope.” Disponible en <https://web.cfa.harvard.edu/mmw/mini.html> (16/05/2023).
- [28] NASA, “Heterodyne Systems and Technology Part I,” p. v, Williamsburg, Virginia: NASA, 1980.
- [29] U. U. Graf, C. E. Honingh, K. Jacobs y J. Stutzki, “Terahertz Heterodyne Array Receivers for Astronomy,” pp. 896–921, *Journal of Infrared, Millimeter, and Terahertz Waves*, 2015.
- [30] Micro Lambda Wireless, INC, “*MLSP Series, Low Noise Synthesizer*.” Disponible en <https://www.microlambdawireless.com/uploads/pdfs/MLSP-6-20-datasheet.pdf> (16/05/2023).
- [31] P. Dhaker, “*Introduction to SPI Interface*.” Disponible en <https://www.analog.com/en/analog-dialogue/articles/introduction-to-spi-interface.html> (16/05/2023).
- [32] C. E. Mackenzie, *Coded Character Sets, History and Development*. IBM Corporation: Addison-Wesley Publishing Company, 1 ed., 1980.
- [33] Python Package Index, “HIDAPI Library.” Disponible en <https://pypi.org/project/hidapi/> (14/12/2023).
- [34] The SciPy community, “SciPy Manual.” Disponible en <https://docs.scipy.org/doc/scipy/reference/generated/scipy.interpolate.PchipInterpolator.html> (24/01/2024).

Anexos

Anexo A. Configuración Inicial del Raspberry Pi del Sistema de Control del Oscilador Local para el *SMWT*

A continuación, se comenta el procedimiento para realizar las configuraciones iniciales del *Raspberry Pi* que se encuentra dentro de sistema de control del oscilador local del radiotelescopio Mini. Lo que se comenta a continuación fue realizado únicamente en el *Raspberry Pi 4B*, por lo que si desea realizarlo con otro modelo, los pasos pueden cambiar levemente.

A.1. Instalación Raspberry Pi OS

Para instalar el sistema operativo del *Raspberry Pi*, se utilizará *Raspberry Pi Imager* ¹, pero otros métodos pueden ser utilizados, como por ejemplo, *Etcher*.

Luego de abrir *Raspberry Pi Imager* y de que se tenga conectada la tarjeta micro SD a un computador personal, debe seleccionar el sistema operativo que se desee instalar. En este caso, se escogió *Raspberry Pi OS Lite (64-Bit)*. Luego, seleccione el sitio donde desee que se instale el sistema operativo. Para este caso, seleccione la micro SD.

En la pestaña emergente, se puede configurar inmediatamente algunos parámetros para operar de manera remota y sin monitor el *Raspberry Pi*. Dentro del mismo, seleccione la opción “Configuración” y modifique los parámetros en función del uso que será dado. En este caso, los ajustes se muestran a continuación en la figura A.1.

¹ Sitio de descarga <https://www.raspberrypi.com/software/>

Figura A.1: Configuraciones iniciales para *Raspberry Pi Imager*.

Con esto, se define el nombre de usuario “pi” con clave “qwerty”. Además, se deja habilitado inmediatamente la conexión mediante SSH y la configuración relacionada al Wi-Fi. Por último, retire la micro SD del computador e insértela al *Raspberry Pi*. Conecte esta última a la respectiva fuente de poder.

Con esto, podrá conectarse de manera inmediata con un teclado y monitor al *Raspberry Pi*. Sin embargo, aún no podrá conectarse de forma remota ya que debe asignarle una dirección IP particular a la red que esté conectada.

A.2. Configuración de Conexión Wi-Fi

En caso de no haber habilitado el Wi-Fi en conjunto con la instalación del sistema operativo del *Raspberry Pi*, o en caso de que quiera modificar sin monitor ni conexión *ethernet*, se pueden seguir los siguientes pasos para cambiar la configuración de Wi-Fi.

Inserte la tarjeta SD en el computador con el que esté trabajando. Luego, dentro de la carpeta de almacenamiento, cree un archivo vacío con extensión “.conf” denominado “wpa_supplicant.conf”. Dentro de este archivo, modifíquelo en función de los requisitos en particular. En este caso, para el Wi-Fi del Laboratorio, en la figura A.2 se muestra el resultado

```
country=CL # Your 2-digit country code
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
network={
    ssid="lab_mwl"
    psk="calannnn"
    key_mgmt=WPA-PSK
}
```

Figura A.2: Configuración del archivo “wpa_supplicant.conf” para el *Raspberry Pi*.

Recuerde guardar las modificaciones realizadas al archivo. Luego, conecte la micro SD al *Raspberry Pi*. Cuando se conecte a la fuente de poder, la Wi-Fi debería estar correctamente configurada y el archivo creado anteriormente, debería desaparecer.

A.3. Configuración SSH

En caso de no haber habilitado el SSH en conjunto con la instalación del sistema operativo del *Raspberry Pi*, puede realizarlo sin monitor.

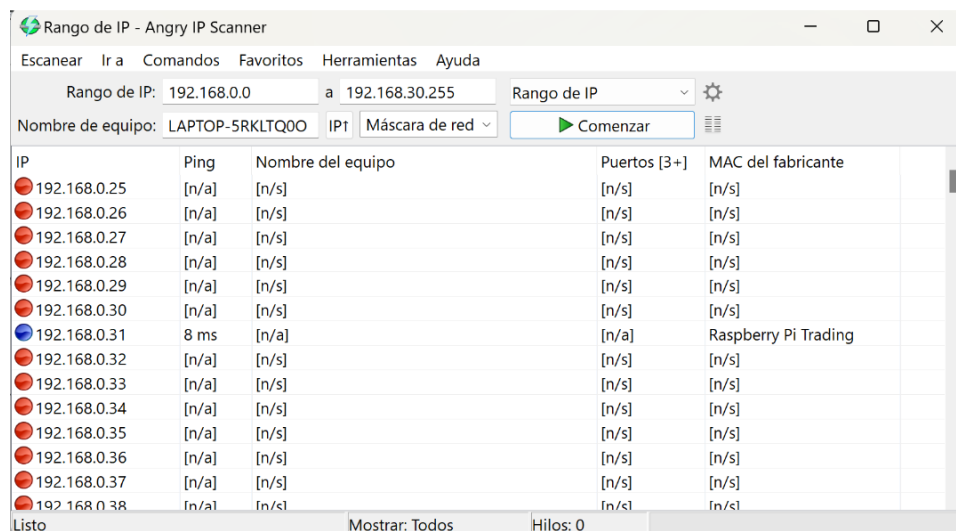
Inserte la tarjeta SD al computador en el que esté trabajando. Luego, dentro de la carpeta de almacenamiento, cree un archivo vacío y sin extensión, denominado “ssh”. En *Linux*, se pueden ejecutar los siguientes comandos

```
1 \ $ cd /Volumes/boot
2 \ $ touch ssh
```

Finalmente, inserte nuevamente la micro SD al *Raspberry Pi*.

A.4. Encontrar la IP del Raspberry Pi

Una forma posible de encontrar la dirección IP del *Raspberry Pi*, cuando no se encuentra conectada a un monitor, es utilizar algún *software* de análisis de redes. En este caso, mediante *Angry IP Scanner*², se escanea la red a la cual esté conectada tanto el *Raspberry Pi* como su computador (deben ser las mismas) y se encuentra la dirección IP del *Raspberry Pi*, como se muestra en la figura A.3.



IP	Ping	Nombre del equipo	Puertos [3+]	MAC del fabricante
192.168.0.25	[n/a]	[n/s]	[n/s]	[n/s]
192.168.0.26	[n/a]	[n/s]	[n/s]	[n/s]
192.168.0.27	[n/a]	[n/s]	[n/s]	[n/s]
192.168.0.28	[n/a]	[n/s]	[n/s]	[n/s]
192.168.0.29	[n/a]	[n/s]	[n/s]	[n/s]
192.168.0.30	[n/a]	[n/s]	[n/s]	[n/s]
192.168.0.31	8 ms	[n/a]	[n/a]	Raspberry Pi Trading
192.168.0.32	[n/a]	[n/s]	[n/s]	[n/s]
192.168.0.33	[n/a]	[n/s]	[n/s]	[n/s]
192.168.0.34	[n/a]	[n/s]	[n/s]	[n/s]
192.168.0.35	[n/a]	[n/s]	[n/s]	[n/s]
192.168.0.36	[n/a]	[n/s]	[n/s]	[n/s]
192.168.0.37	[n/a]	[n/s]	[n/s]	[n/s]
192.168.0.38	[n/a]	[n/s]	[n/s]	[n/s]

Figura A.3: *Software Angry IP Scanner*, donde se muestra la dirección IP asignada al *Raspberry Pi*.

Es importante que dentro de la aplicación, se añada el parámetro de “MAC del fabricante”, puesto que así le entregará como resultado el nombre del dispositivo que esté conectada a la red.

² Sitio de descarga <https://angryip.org/download/#windows>

A.5. Configuración de IP estática

Luego de identificar la IP del *Raspberry Pi*, puede acceder por primera vez de manera remota con la IP encontrada utilizando el comando en *Linux* “\$ ssh pi@192.168.0.31”.

Luego, para definir una IP estática y que no tenga conflictos con otra, debe verificar que el servicio “DHCPD” esté activado:

```
1 $ sudo service dhcpd status
```

En caso de que no esté activo, ingrese:

```
1 $ sudo service dhcpd start
2 $ sudo systemctl enable dhcpd
```

Luego, para verificar la dirección IP actual del *Raspberry Pi*, ejecute en la terminal:

```
1 \ $ hostname -I
```

El resultado debería reflejar la dirección IP. En este caso, la dirección indica la conexión de la interfaz *wlan0* (también se puede mostrar la IP de la conexión por ethernet). Guarde el valor con nombre “ip-address”.

Luego, identifique la interfaz de red predeterminada que esté activa. Para esto, ejecute el comando:

```
1 $ ip r | grep default
```

En el resultado, guarde el valor de la dirección IP asignada para la conexión por *wlan0* como “router-ip-address”. Si desea comprobar el nombre de la interfaz que está conectado, ejecute:

```
1 $ route | grep '^default' | grep -o '[^ ]*$'
```

Guarde el resultado como “interface-name”. Luego, debe obtener la dirección DNS, para esto ejecute:

```
1 $ sudo nano /etc/resolv.conf
```

El resultado debería abrir el archivo “.conf”. Dentro de este, busque y guarde la primera dirección IP como “dns-address”

Finalmente, ejecute el siguiente comando para abrir el archivo:

```
1 $ sudo nano /etc/dhcpd.conf
```

Dentro de este, busque y modifique las líneas.

```
1 interface [interface-name]
2 static ip\_address = [ip-address]/[cidr-suffix]
3 static routers = [router-ip-address]
```


4 static domain_name_servers = [dns-address]

En el presente caso, los parámetros se indican a continuación en la figura A.4.

```
# For wlan0 connection
interface wlan0
static ip_address=192.168.2.123/24
#static ip6_address=fd51:42f8:caae:d92e::ff/64
static routers=192.168.2.1
static domain_name_servers=192.168.2.1
```

Figura A.4: Configuración de parámetros para dirección IP fija en *Raspberry Pi*

Finalmente, reinicie el *Raspberry Pi* y puede conectarse de manera remota por la terminal de su computador mediante el comando `pi@192.168.2.123` con la clave “qwerty”

Anexo B. Documentación del Oscilador Local

B.1. Google Drive con Documentos

Se ha creado un sitio ³ en el cual se recopiló la documentación técnica de los componentes utilizados tanto de radiofrecuencias como del sistema de control. Esto se realizó con el objetivo de reunir la información de mayor relevancia y para mantenerla en un sitio en común.

Dentro del *Google Drive*, se tiene una carpeta principal denominada “LO”. Allí, se pueden encontrar las siguientes carpetas con mayor documentación y fichas técnicas:

- ADS1115
- Detector UD
- Documentos Micro Lambda (Sintetizador)
- Level Shifter
- Multiplicador (X4)
- Relé

Con esto, se facilita el acceso a toda la documentación para el trabajo futuro.

B.2. Estructura del Repositorio Instmini

A lo largo de todo el trabajo de título, la parte de programación fue realizada en el repositorio de *GitHub* que se denominó como “instmini”⁴. Dentro de este, se pueden encontrar las ramas “main”, que debe estar siempre operando, y “test”, en la que se pueden realizar múltiples pruebas sin alterar el funcionamiento de la primera rama.

Dentro de la carpeta denominada “lab_tests”, se pueden encontrar otras cuatro: “adc”, “comms”, “libs” y “synth”. En el interior de “libs”, se puede encontrar la librería “HIDA-PI” que fue instalada para ejecución de *synthesizer.c*. Dentro del directorio “adc”, se pueden encontrar respaldos de los códigos que fueron utilizados para las pruebas del sistema de medición de potencia, y en particular del módulo *ADS1115*. Además, se dejaron datos que fueron recopilados en las pruebas de la interpolación, medición de datos de voltaje y conversión a potencia en forma de *.csv* y *.png*.

³ Enlace al Google Drive https://drive.google.com/drive/folders/1r7B1EKxyTxTUVmHNT6EzjkT0JGN_-3Tm?usp=sharing

⁴ Enlace al repositorio Github: <https://github.com/Alvrinc1091/instmini.git>

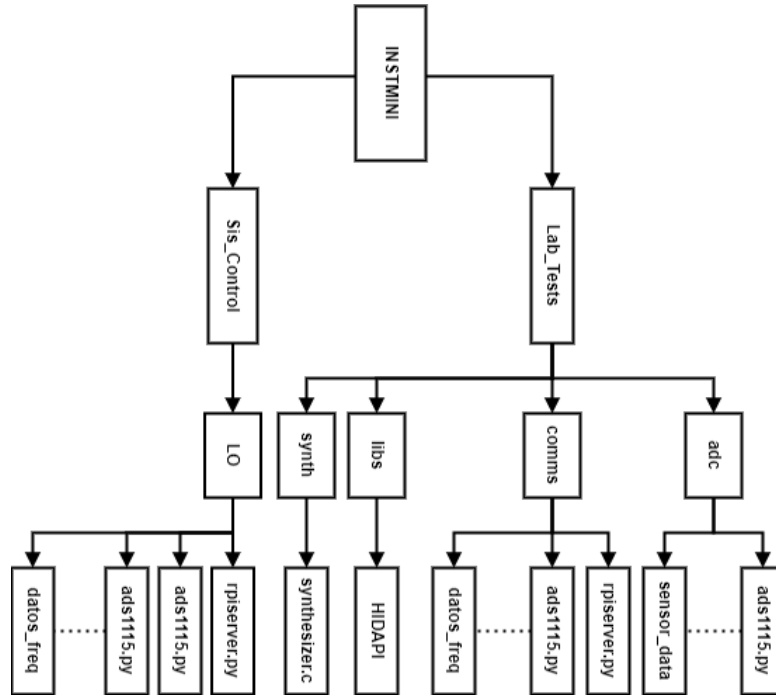


Figura B.1: Diagrama de la distribución del repositorio *Instmini*.

Dentro del directorio “synth”, se encuentran los respaldos de los códigos que fueron utilizados para las pruebas de comunicación con el sintetizador. El más relevante es el denominado *synthesizer.c*, puesto que tal código corresponde a la interfaz de línea de comandos para el control y monitoreo del sintetizador. En el directorio “comms”, se sitúan los códigos para el funcionamiento del módulo *ADS1115*, denominado como *ads1115.py*. Para la simulación del cliente que envía algún valor de frecuencia, se tiene a *client_rpi.py*. Para la comunicación básica con el sintetizador se encuentra *synth.py*. Finalmente, para el sistema de control y medición de potencia, se tiene a *rpiserver.py*. Además, se pueden encontrar otros códigos de prueba más el respaldo de datos en forma de *.png* y *.csv*.

Dentro de la carpeta “LO”, se encuentra una copia de *rpiserver.py*, *ads1115.py*, *synth.py* y *synthesizer.c*, además de los archivos *.csv* y *.png* que guardan los datos recolectados.

Anexo C. Códigos en Python y C

A continuación, se muestra el código en *Python* que se utiliza en *rpiserver.py* para la comunicación con el módulo *ADS1115*, además de funciones básicas que son utilizadas en el sistema de medición de potencia del oscilador local.

Código C.1: Código de funcionamiento para módulo *ADS1115*: *ads1115.py*.

```
1
2 # Script para manejar el módulo ADC ADS1115, cuya entrada corresponde a la
3 # lectura de voltaje proveniente del UD Broadband Detector y la información
4 # de salida es manejada por la raspberry pi. Para esto se realiza un
5 # procesamiento de los datos de entrada, con los cuales se muestra el valor
6 # leído de voltaje más la conversión a potencia dBm
7
8 # Importar librerías
9 import board
10 import adafruit_ads1x15.ads1115 as ADS
11 from adafruit_ads1x15.analog_in import AnalogIn
12 import numpy as np
13 import time
14 import matplotlib.pyplot as plt
15 from scipy.interpolate import PchipInterpolator
16
17 # Configuración del ADS1115
18 i2c = board.I2C() # Configura la comunicación I2C
19 ads = ADS.ADS1115(i2c, gain = 1) # Crea una instancia del objeto ADS1115
20
21 # Configura la entrada analógica para la lectura de voltaje
22 canal = AnalogIn(ads, ADS.P0) # Cambia ADS.P0 al número de canal que estés utilizando
23
24 # Datos de frecuencia y sensibilidad.
25 # Los datos provienen de la información del fabricante
26 data_freq = np.array([36, 40, 45, 50, 52.5, 55, 57.5, 60]) #np.array([40, 45, 50, 55, 60])
27 data_sens = np.array([1117.47, 818.08, 655.59, 641.09, 574.65, 558.101, 616.68, 594.07]) #np.
    ↪ array([1993, 1902, 2256, 2076, 1876])
28
29 # Crear el interpolador PCHIP para la sensibilidad en función de la frecuencia
30 # Se utiliza para las frecuencias que se encuentren entre medio de los valores
31 # establecidos por el fabricante
32 sensibilidad_interpolator = PchipInterpolator(data_freq, data_sens)
33
34 # Diccionario para almacenar los datos en forma de diccionario
35 # {frecuencia: voltaje, potencia, tiempo}
36 datos = {}
37
38 # Función para convertir voltaje leído desde el UD Broadband Detector a potencia
39 def volt_to_power(freq):
40
41     # Realizar la lectura de voltaje desde el canal de entrada analógica
42     lectura_voltaje = canal.voltage
```

```

43
44 # Convertir lectura a voltaje
45 valor_analogico = (abs(lectura_voltaje) ) + 0.00001 # Pequeño ajuste de los datos *
    ↪ 1.001001
46 valor_analogico = (999.0 * valor_analogico + valor_analogico) / 1000.0 # Pequeño
    ↪ ajuste de los datos
47
48 # Conversión de V -> mW
49 potenciamW = (valor_analogico * 1000) / sensibilidad_interpolator(freq)
50
51 # Conversión de mW -> dBm
52 potenciadBm = 10 * np.log10(potenciamW)
53
54 # Entrega el valor de voltaje más la potencia
55 return valor_analogico, potenciadBm
56
57 # Función para imprimir voltaje, potencia y frecuencia
58 def volt_power_print(freq):
59
60     # freq_UD = freq * 4 / (10 ** 9)
61
62     # Guarda en dos variables las lecturas de voltaje y potencia
63     valor_analogico, potenciadBm = volt_to_power(freq)
64
65     print("\n")
66     print(f"Frecuencia: {freq} [GHz]")
67     print(f"Voltaje UD: {valor_analogico:.4f} [V]") # Formatear el voltaje a 4 decimales, se
        ↪ puede ajustar
68     print(f"Potencia: {potenciadBm:.4f} [dBm]") # Formatear la potencia a 4 decimales, se
        ↪ puede ajustar
69
70 # Función para realizar lecturas durante 10 segundos de voltaje y potencia
71 # El período que se toma datos se puede ajustar en función del usuario
72 def volt_power_lecture(freq):
73
74     tiempo_inicial = time.time()
75     tiempo_final = tiempo_inicial + 10 # Tomar datos durante 10 segundos
76     print("Recopilando datos...")
77
78     while time.time() < tiempo_final:
79         valor_analogico, potenciadBm = volt_to_power(freq)
80
81         # Obtener la hora actual
82         tiempo_actual = time.strftime("%Y-%m-%d %H:%M:%S")
83
84         # Verificar si ya existe una entrada para la frecuencia en el diccionario de datos
85         if freq not in datos:
86             datos[freq] = {"tiempo": [], "voltaje": [], "potencia": []}
87
88         # Guardar los datos en el diccionario
89         datos[freq]["voltaje"].append(valor_analogico)
90         datos[freq]["potencia"].append(potenciadBm)

```

```

91     datos[freq]["tiempo"].append(tiempo_actual)
92
93 # Función para guardar datos en el diccionario
94 def volt_power_save(freq, voltaje, potencia, tiempo):
95     datos[freq] = {
96         "voltaje": voltaje,
97         "potencia": potencia,
98         "tiempo": tiempo
99     }
100     print(datos)
101
102 # Función para plotear los datos en la misma figura
103 def volt_power_show():
104
105     for freq, mediciones in datos.items():
106         tiempo = mediciones["tiempo"]
107         voltaje = mediciones["voltaje"]
108         potencia = mediciones["potencia"]
109
110         # Extraer solo la hora de la marca de tiempo
111         horas = [t.split()[1] for t in tiempo]
112
113         # Generar subplots en una sola figura
114         plt.figure(figsize=(12, 10)) # Ancho x Alto
115
116         # Gráfico de tiempo vs. voltaje
117         plt.subplot(2, 1, 1)
118         plt.plot(horas, voltaje, label="Voltaje (V)")
119         plt.xlabel("Hora")
120         plt.ylabel("Voltaje (V)")
121         plt.title(f"Voltaje vs. Hora - Frecuencia {freq} GHz")
122         plt.legend()
123         plt.grid(True)
124
125         # Gráfico de voltaje vs. potencia
126         plt.subplot(2, 1, 2)
127         plt.plot(voltaje, potencia, label="Potencia (dBm)", marker = 'o')
128         plt.xlabel("Voltaje (V)")
129         plt.ylabel("Potencia (dBm)")
130         plt.title(f"Potencia vs. Voltaje - Frecuencia {freq} GHz")
131         plt.legend()
132         plt.grid(True)
133
134         # Guarda los datos en una figura. Cada vez que se ejecuta, se sobrescribe
135         # Se puede modificar para guardar distintas figuras
136         plt.savefig('sensor_data.png')
137         print('Gráfico guardado como sensor_data.png')
138
139         plt.tight_layout() # Ajustar automáticamente la disposición de los subplots
140         #plt.show()
141
142         # Guardar datos en archivo CSV. En este caso, para cada frecuencia se genera

```

```

143     # un archivo .csv, de modo tal que se tendrán distintos archivos de respaldo
144     with open(f"datos_{freq}GHz.csv", "w") as archivo_csv:
145         archivo_csv.write("Hora,Voltaje (V),Potencia (dBm)\n")
146         for i in range(len(horas)):
147             archivo_csv.write(f"{horas[i]},{voltaje[i]},{potencia[i]}\n")
148         print('Datos guardados en datos_freq.csv')
149
150 # Ejemplo de uso
151 #if __name__ == "__main__":
152     # frecuencia_ejemplo = 50
153     # volt_power_lecture(frecuencia_ejemplo)
154     # volt_power_print(frecuencia_ejemplo)
155     # volt_power_show() # Mostrar y guardar los datos después de la lectura

```

A continuación, se muestra el código en *Python* que es utilizado dentro de *rpiserver.py*, dentro del cual se encuentran funciones básicas para la comunicación y envío de comandos con el sintetizador.

Código C.2: Código de funcionamiento para comunicación básica con sintetizador: *synth.py*.

```

1
2 # Script para mandar comandos específicos al sintetizador.
3 # Se provee de funciones para abrir y cerrar la conexión con el dispositivo,
4 # Para enviar los comandos de seteo de frecuencia, temperatura y bits de estado
5
6 # Importar librerías
7 # HID (Human-Interface Device) realiza la conexión con el sintetizador
8
9 import hid
10 import time
11 import RPi.GPIO as GPIO
12
13 # Definición de las variables propias del sintetizador.
14 # Para conocerlas, se puede ejecutar "lsusb" en la terminal de la raspberry
15
16 VENDOR_ID = 0x04d8 # Reemplaza con el ID de proveedor de tu dispositivo
17 PRODUCT_ID = 0x003f # Reemplaza con el ID de producto de tu dispositivo
18
19 led_pin = 17 # pin GPIO
20
21 # Función que abre y establece la conexión con el sintetizador
22
23 def open_device():
24     try:
25         device = hid.device()
26         device.open(VENDOR_ID, PRODUCT_ID)
27         print("Conexión establecida con el dispositivo USB.")
28         return device
29     except Exception as e:
30         print(f"Error al abrir el dispositivo: {str(e)}")
31         return None

```

```

32
33 # Función que cierra la conexión con el sintetizador
34
35 def close_device(device):
36     if device:
37         device.close()
38         print("Conexión con el dispositivo USB cerrada.")
39
40 # Función para enviar el comando de frecuencia y setear el valor
41 # en el sintetizador. La frecuencia debe ser enviada de tal forma que:
42 # xxxxx.xxxxxx [MHz]
43
44 def send_freq(device, freq):
45     command = f"f{freq}" # Comando de frecuencia debe comenzar con una fxxxxx.xxxxxx
46     print(" ----- Información Sintetizador ----- ")
47     print(command)
48     print(f'Frecuencia del sintetizador fijada en: {freq} [MHz]')
49
50     if 4000 <= int(freq) <= 16000: # Frecuencia debe estar entre 4 y 16 GHz
51         device.write(list(command.encode()))
52         print(f"Frecuencia enviada al sintetizador: {command}")
53         time.sleep(1)
54
55         response_freq = device.read(64) # Variable que guarda la respuesta del sintetizador
56         if response_freq: # Este envío de comando no tiene una respuesta del sintetizador
57             response_str = "".join(map(chr, response_freq))
58             print(f"Frecuencia del sintetizador fijada en: {response_str}")
59             return response_str
60         else:
61             print("No se recibió respuesta del dispositivo para la verificación de frecuencia.")
62             return None
63     else:
64         print("Frecuencia no válida. Debe estar entre 4 GHz (4000 MHz) y 16 GHz (16000 MHz
        ↪ ).")
65         return None
66
67 # Función para enviar el comando de temperatura y conocer el valor de temperatura
68 # interna del sintetizador. Debe estar entre 0° y 60° C
69
70 def send_temp(device):
71     temp_command = "T" # Comando para conocer la temperatura del sintetizador
72     device.write(list(temp_command.encode()))
73     print(" ----- Información Sintetizador ----- ")
74     print(f"Comando de solicitud de temperatura enviado: {temp_command}")
75
76     response_temp = device.read(64) # Variable que almacena la respuesta del sintetizador
77     if response_temp:
78         response_str = "".join(map(chr, response_temp))
79         #print(f"Respuesta del dispositivo (temperatura cruda): {response_str}")
80
81         # La temperatura del dispositivo debe estar entre 0° y 60°C
82         try:

```



```

83     temperature = float(response_str.strip("+").strip("C"))
84     print(f"La temperatura del sintetizador es de: {response_str}") # Imprime la
↪ respuesta del sintetizador
85
86     # Se imprimen diferentes mensajes en función del valor leído
87     if 1.0 <= temperature <= 5.0:
88         print("Advertencia, temperatura baja")
89     elif 5.0 < temperature <= 15.0:
90         print("El dispositivo está alcanzando temperaturas bajas, ten cuidado")
91     elif 15.0 < temperature <= 50.0:
92         print("Unidad en buen estado")
93     elif 50.0 < temperature <= 55.0:
94         print("El dispositivo está alcanzando altas temperaturas, ten cuidado")
95     elif 55.0 < temperature < 59.0:
96         print("Advertencia, temperatura alta")
97     else:
98         print("Valor de temperatura fuera de rango")
99     except ValueError:
100         print("Respuesta inválida del dispositivo (temperatura):", response_str)
101 else:
102     print("No se recibió respuesta del dispositivo para la verificación de temperatura.")
103
104 # Función para enviar el comando de bits de estado del sintetizador y recibir su respuesta
105 # La típica respuesta son 8 bits de estado B0 | B1 | B2| B3| B4 | B5 | B6 | B7
106 # B0 = 1 (0) -> 10 MHz (Un)Locked (Internal Ref)
107 # B1 = 1 (0) -> YIG PLL (Un)Locked (External Ref)
108 # B2 a B5 no son utilizados
109 # B6 = 1 (0) -> Self Test Passed (Failed)
110 # B7 = 1 (0) -> NOVO (Un)Locked
111
112 # Define el pin del LED
113 led_pin = 17
114
115 # Configura el modo de pines GPIO
116 GPIO.setmode(GPIO.BCM)
117
118 # Configura el pin GPIO como salida
119 GPIO.setup(led_pin, GPIO.OUT)
120
121
122 def send_status(device):
123     status_command = "?" # Comandos para conocer los bits de estado del sintetizador
124     device.write(list(status_command.encode()))
125     print(" ----- Información Sintetizador ----- ")
126     print(f"Comando de solicitud de bits de estado enviado: {status_command}")
127
128     response_status = device.read(64) # Variable que almacena la respuesta del sintetizador
129     if response_status:
130         response_str = "".join(map(chr, response_status))
131         response_str = response_str[:8] # Tomar solo los primeros 8 caracteres de la
↪ respuesta
132         print(f"Bits de estado del sintetizador: {response_str}") # Imprime la respuesta del

```

```

↳ sintetizador
133
134 # Procesar cada bit en la respuesta
135 bits = [int(bit) for bit in response_str]
136
137 # Inicializa GPIO una vez
138 #GPIO.setmode(GPIO.BCM)
139 #GPIO.setup(led_pin, GPIO.OUT)
140
141 # Imprime mensajes en función de los bits de estado
142 if len(bits) == 8:
143     B0, B1, B2, B3, B4, B5, B6, B7 = bits
144     # Lectura del bit B0
145     print(f"B0: {'100 MHz Locked (Internal Ref)' if B0 == 1 else '100 MHz Unlocked (
↳ Internal Ref)'}")
146
147     # Lectura del bit B1
148     print(f"B1: {'YIG PLL Locked' if B1 == 1 else 'YIG PLL Unlocked'}")
149
150     # B2 a B5 no se utilizan
151
152     # Lectura del bit B6
153     print(f"B6: {'Self Test Passed' if B6 == 1 else 'Self Test Failed'}")
154     # Lectura del bit B7
155     print(f"B7: {'NOVO Locked (External Ref)' if B7 == 1 else 'NOVO Unlocked (
↳ External Ref)'}")
156
157     if response_str == '11000011':
158         #GPIO.output(led_pin, GPIO.HIGH)
159         GPIO.output(led_pin, GPIO.LOW) # Enciende Relé
160         print("RELÉ LED ON => +5V")
161     else:
162         #GPIO.cleanup()
163         GPIO.output(led_pin, GPIO.HIGH) # Apagando Relé
164         print("Revise Bits de estado de Sintetizador")
165         print("RELÉ LED OFF => 0V")
166
167     else:
168         print("Respuesta inválida del dispositivo (estado):", response_str)
169         print("RELÉ LED OFF => 0V")
170         #GPIO.cleanup()
171         GPIO.output(led_pin, GPIO.HIGH) # Apagando Relé
172 else:
173     print("No se recibió respuesta del dispositivo para la solicitud de estado.")
174     print("RELÉ LED OFF => 0V")
175     #GPIO.cleanup()
176     GPIO.output(led_pin, GPIO.HIGH) # Apagando Relé
177
178
179 # def send_status(device):
180 #     status_command = "?" # Comandos para conocer los bits de estado del sintetizador
181 #     device.write(list(status_command.encode()))

```

```

182 # print(" ----- Información Sintetizador ----- ")
183 # print(f"Comando de solicitud de bits de estado enviado: {status_command}")
184
185 # response_status = device.read(64) # Variable que almacena la respuesta del
↳ sintetizador
186 # if response_status:
187 #     response_str = "".join(map(chr, response_status))
188 #     response_str = response_str[:8] # Tomar solo los primeros 8 caracteres de la
↳ respuesta
189 #     print(f"Bits de estado del sintetizador: {response_str}") # Imprime la respuesta del
↳ sintetizador
190
191 #     # Procesar cada bit en la respuesta
192 #     bits = [int(bit) for bit in response_str]
193
194 #     # Imprime mensajes en función de los bits de estado
195 #     if len(bits) == 8:
196 #         B0, B1, B2, B3, B4, B5, B6, B7 = bits
197 #         # Lectura del bit B0
198 #         print(f"B0: {'100 MHz Locked (Internal Ref)' if B0 == 1 else '100 MHz Unlocked
↳ (Internal Ref)'}")
199
200 #         # Lectura del bit B1
201 #         print(f"B1: {'YIG PLL Locked (External Ref)' if B1 == 1 else 'YIG PLL
↳ Unlocked (External Ref)'}")
202
203 #         # B2 a B5 no se utilizan
204
205 #         # Lectura del bit B6
206 #         print(f"B6: {'Self Test Passed' if B6 == 1 else 'Self Test Failed'}")
207 #         # Lectura del bit B7
208 #         print(f"B7: {'NOVO Locked' if B7 == 1 else 'NOVO Unlocked'}")
209
210 #         if response_str == '11000011':
211 #             GPIO.setmode(GPIO.BCM)
212 #             GPIO.setup(led_pin, GPIO.OUT)
213 #             GPIO.output(led_pin, GPIO.HIGH)
214 #             print("RELÉ LED ON => +5V")
215 #         else:
216 #             GPIO.cleanup()
217 #             print("RELÉ LED OFF => 0V")
218
219 #         else:
220 #             print("Respuesta inválida del dispositivo (estado):", response_str)
221 #     else:
222 #         print("No se recibió respuesta del dispositivo para la solicitud de estado.")
223
224 # if __name__ == "__main__":
225 #     device = open_device()
226 #     if device:
227 #         send_freq(device, 9000)
228 #         send_temp(device)

```

```

229 # send_status(device)
230 # close_device(device)

```

Código C.3: Código de simulación de cliente: *client_rpi.py*.

```

1
2 import socket
3
4 def enviar_frecuencia(host, port, freq):
5     client = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
6     client.connect((host, port))
7     client.send(str(freq).encode())
8     response = client.recv(1024)
9     print(f"Respuesta del servidor: {response.decode()}")
10    client.close()
11
12 if __name__ == "__main__":
13     host = '192.168.2.123' #'169.254.81.30'
14     port = 8000
15     freq = 12500000000 # Cambiar para enviar distintas frecuencias # 8375000000 #
16     ↪ 9000000000 # 9250000000 # 10125000000 # 11000000000 # 11875000000
17     print(f'Frecuencia enviada {freq} [Hz]') # 12500000000 # 12750000000 #
18     ↪ 13625000000 # 14500000000 # 15000000000 # 15500000000
19     enviar_frecuencia(host, port, freq)

```

A continuación, se muestra el código en *Python* con el que se trabajó para la utilización de la interfaz de línea de comandos del sistema de control del oscilador local.

Código C.4: Código de funcionamiento para sistema de control: *rpiserver.py*.

```

1
2 import socket
3 import threading
4 import RPi.GPIO as GPIO
5 import time
6 from ads1115 import volt_power_print, volt_power_show, volt_power_lecture
7 from synth import open_device, close_device, send_freq, send_temp, send_status
8
9 # -----> Definición Variables
10 # Definición de variables
11 frecuencia = 0 # Variable global para almacenar el valor de la frecuencia que es recibida
12     ↪ desde el PIC
13 freq_UD = 0 # Variable global para almacenar el valor de la frecuencia que se utiliza en
14     ↪ medición de potencia
15 freq_synth = 0 # Variable global para almacenar el valor de la frecuencia que se envía al
16     ↪ sintetizador
17
18 # Configuración del pin GPIO
19
20 # Define el pin del LED
21 led_pin = 17 # Pin 17 GPIO utilizado para el control del Relé

```

```

19
20 # Configura el modo de pines GPIO
21 GPIO.setmode(GPIO.BCM)
22
23 # Configura el pin GPIO como salida
24 GPIO.setup(led_pin, GPIO.OUT)
25
26 # -----
27
28 # -----> Definición Funciones
29 # -----> manejar_cliente
30 # Función manejar_cliente utilizada para manejar las conexiones entrantes de clientes
31 # En la variable frecuencia, recibirá el valor de la frecuencia de interés que provenga desde el
    ↪ PIC
32 # Luego define freq_UD para guardar la variable que se utiliza en la medición de potencia
33 # Mientras que freq_synth guarda la variable que se envía al sintetizador
34 # Finalmente, cada vez que se recibe una nueva frecuencia, envía al sintetizador los
    ↪ comandos
35 # Temperatura T y Estado ? para imprimir el estado inicial del dispositivo
36
37 def manejar_cliente(client_socket):
38     global frecuencia, freq_UD, freq_synth
39
40     data = client_socket.recv(1024)
41     if data:
42
43         frecuencia = float(data.decode())
44
45         freq_UD = frecuencia * 4 / (10 ** 9) # Multiplicador x4 de la frecuencia y conversión
    ↪ a GHz
46         freq_synth = frecuencia / (10 ** 6) # Pasa freq recibida a MHz (xx 000 MHz = xx
    ↪ GHz)
47         freq_synth = f"{freq_synth:.6f}".rstrip('0').rstrip('.')
48
49         print(f"Frecuencia recibida: {frecuencia} [Hz]")
50         print(f"Frecuencia Sintetizador: {freq_synth} [MHz]")
51         print(f"Frecuencia en Medidor de Potencia: {freq_UD} [GHz]")
52
53         resultado = f"\n Frecuencia seteada en Sintetizador: {freq_synth} [MHz] \
54             \n Frecuencia en Medidor de Potencia {freq_UD} [GHz]"
55
56         client_socket.send(resultado.encode())
57
58     if device:
59         # Llamar a la función send_freq para configurar la frecuencia en el sintetizador
60         response = send_freq(device, freq_synth)
61         if response:
62             # Llamar a la función send_temp para obtener la temperatura del dispositivo
63             send_temp(device)
64             # Llamar a la función send_status para obtener el estado del dispositivo
65             send_status(device)
66         # Llamar a la función close_device para cerrar el sintetizador

```

```

67     #close_device(device)
68
69     client_socket.close()
70
71     # -----> manejar_comandos
72     def manejar_comandos():
73
74         while True:
75
76             command = input("Ingrese un comando: ")
77
78             # Imprime el valor de Frecuencia (X4), Voltaje leído por el UD y la Potencia
79             if command == 'V' or command == 'v':
80                 print(" ----- Información Sintetizador ----- ")
81                 print(f"Frecuencia Sintetizador: {freq_synth} [MHz]")
82                 print(" ----- ")
83                 print("--- Información UD Broadband Detector ---")
84                 volt_power_print(freq_UD)
85                 print(" ----- ")
86
87             # Muestra la lista de comandos disponibles
88             elif command == 'C' or command == 'c':
89                 print(" 'F' -> Envía el valor de Frecuencia al Sintetizador (Valor debe ser Fxxxxx
↪ .xxxxxx) (aún no implementada )\
90                 \n'V' -> Muestra valores de Voltaje [V] y Potencia [dBm]\
91                 \n'P' -> Guarda los datos de V y P en un CSV además de graficarlos\
92                 \n'T' -> Muestra la Temperatura actual del Sintetizador\
93                 \n'S' -> Muestra los Bits de estado del Sintetizador\
94                 \n'Q' -> Cerrar el Servidor")
95
96             # elif command.startswith(('f', 'F')) and command[1:].isdigit():
97             #     freq_synth = int(command[1:])
98             #     freq_UD = freq_synth * 4 / (103)
99             #     volt_power_print(freq_UD)
100
101             #     if device:
102             #         try:
103             #             send_freq(device, freq_synth)
104             #             print(f"Frecuencia seteada en {freq_synth} [MHz]')
105
106             #         except Exception as e:
107             #             print(f'Error al configurar la frecuencia: {e}')
108             #     else:
109             #         print('El dispositivo no está conectado. Verifique la conexión')
110
111             # Guarda los datos (recopila los últimos 10 seg.) de Voltaje y Potencia en un CSV
112             # Genera dos gráficos: Tiempo vs Volt y Volt vs Pot
113
114             elif command == "P" or command == 'p':
115                 volt_power_lecture(freq_UD)
116                 volt_power_show()
117

```

```

118     # Comunicación con el sintetizador. Muestra la temperatura actual del sintetizador
119     elif command == 'T' or command == 't':
120         #device = open_device()
121         if device:
122             try:
123                 # Ejecutar la función para enviar el comando de temperatura
124                 send_temp(device)
125                 #     close_device(device)
126             except Exception as e:
127                 print(f"Error al obtener la temperatura: {e}")
128         else:
129             print("El dispositivo no está conectado. Comando no disponible.")
130
131
132     # Comunicación con el sintetizador. Muestra los bits de estado del sintetizador
133     # Bits de estado: B0 | B1 | B2 | B3 | B4 | B5 | B6 | B7
134     # B0 = 1 (0) -> 10 MHz (Un)Locked (Internal Ref)
135     # B1 = 1 (0) -> YIG PLL (Un)Locked
136     # B2 a B5 no son utilizados
137     # B6 = 1 (0) -> Self Test Passed (Failed)
138     # B7 = 1 (0) -> NOVO (Un)Locked (External Ref)
139     elif command == 'S' or command == 's': # Equivalente a enviar ?
140         # Ejecutar la función para enviar el comando de estado
141         #device = open_device()
142         if device:
143             try:
144                 send_status(device)
145                 #     close_device(device)
146             except Exception as e:
147                 print(f"Error al obtener los bits de estado: {e}")
148         else:
149             print("El dispositivo no está conectado. Comando no disponible.")
150
151     # Cierra el servidor y la recepción de comandos
152     elif command == 'Q' or command == 'q':
153         print("Finalizando Programa...")
154         time.sleep(2)
155         GPIO.output(led_pin, GPIO.HIGH) # Apagando Relé
156         print("Programa Finalizado")
157
158         break # Salir del bucle principal y cerrar el servidor
159
160 # Función para inicializar el servidor y escuchar datos de frecuencia
161 def iniciar_servidor(host, port):
162     server = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
163     server.bind((host, port))
164     server.listen(5)
165     print(f"\nServidor escuchando en {host}:{port}")
166
167     while True:
168         try:
169             client_sock, addr = server.accept()

```

```

170     print(" \n ----- Información del Servidor ----- ")
171     print(f"Conexión entrante de {addr[0]}:{addr[1]}")
172     client_thread = threading.Thread(target=manejar_cliente, args=(client_sock,))
173     client_thread.start()
174     print(" ----- ")
175     except KeyboardInterrupt:
176         break # Salir del bucle si se presiona Ctrl+C
177
178 # Función para enviar constantemente los comandos de V, para mostrar freq, volt y pot
179 # También muestra la temperatura y bits de estado del sintetizador
180 def data_periodica(device):
181     #device = open_device()
182     while True:
183         # Envía los comandos V, T y S
184         volt_power_print(freq_UD)
185         if device:
186             try:
187                 send_temp(device)
188                 send_status(device)
189             except Exception as e:
190                 print(f"Error al comunicarse con el dispositivo: {e}. Por favor verifique la conexión
↪ n.")
191         else:
192             print(f"Error al comunicarse con el dispositivo. Por favor verifique la conexión.")
193         # Enviar la siguiente información en X seg
194         time.sleep(5)
195
196 if __name__ == "__main__":
197     host = '192.168.2.123' #'169.254.81.30' # Definir IP
198     port = 8000          # Definir el puerto de comunicación
199
200     #GPIO.output(led_pin, GPIO.LOW) # Enciende Relé
201
202     # Iniciar el hilo para iniciar el servidor
203     # Siempre va a estar corriendo de fondo
204     server_thread = threading.Thread(target=iniciar_servidor, args=(host, port))
205     server_thread.daemon = True
206     server_thread.start()
207
208     # Abre el dispositivo una vez antes de iniciar los hilos
209     #device = open_device()
210
211     # Inicia el hilo para acciones periódicas
212     # Siempre va a estar mandando los comandos cada cierto período
213     device = open_device()
214     acciones_thread = threading.Thread(target=data_periodica, args=(device,))
215     acciones_thread.daemon = True
216     acciones_thread.start()
217
218     # Llama a la función del manejo de comandos, de manera tal que siempre
219     # esté escuchando la recepción de comandos
220     manejar_comandos()

```


A continuación, se muestra el código en C para la interfaz de línea de comandos que se utilizó para la comunicación, control y monitoreo del sintetizador.

Código C.5: Código de funcionamiento para control y monitoreo con sintetizador: *synthesizer.c*.

```
1
2 // RUN:
3 // gcc -o synthesizer synthesizer.c -lhidapi-hidraw -lpthread
4 // sudo ./synthesizer
5 // sends the temperature with comments every 29secs
6 // if the temperature gets out of the range, the program stops
7 // every 31 seconds, it also checks the status bits and stops if some of them are different from
   ↪ 0
8 // when exit command, the program show the final information
9
10 #include <stdio.h>
11 #include <stdlib.h>
12 #include <hidapi/hidapi.h>
13 #include <string.h>
14 #include <unistd.h>
15 #include <pthread.h> // Include the pthread library
16
17 #define VENDOR_ID 0x04d8 // Replace with your device's vendor ID
18 #define PRODUCT_ID 0x003f // Replace with your device's product ID
19
20 void decodeBits(const char *bits) {
21     if (strlen(bits) != 8) {
22         printf("Input must be an 8-bit binary sequence\n");
23         return;
24     }
25
26     int d0 = bits[0] - '0';
27     int d1 = bits[1] - '0';
28     int d6 = bits[6] - '0';
29     int d7 = bits[7] - '0';
30
31     if (d0 == 0) {
32         printf("D0: 100 MHz Unlocked (Internal Ref)\n");
33     } else {
34         printf("D0: 100 MHz Locked (Internal Ref)\n");
35     }
36
37     if (d1 == 0) {
38         printf("D1: YIG PLL Unlocked\n");
39     } else {
40         printf("D1: YIG PLL Locked\n");
41     }
42
43     // D2 to D5 not used
44
```

```

45  if (d6 == 0) {
46      printf("D6: Self Test Failed\n");
47  } else {
48      printf("D6: Self Test Passed\n");
49  }
50
51  if (d7 == 0) {
52      printf("D7: NOVO Unlocked (External Ref)\n");
53  } else {
54      printf("D7: NOVO Locked (External Ref)\n");
55  }
56 }
57
58 void *temperature_monitor(void *arg) {
59     hid_device *device = (hid_device *)arg;
60
61     while (1) {
62         unsigned char temp_command[] = "T";
63         int result = hid_write(device, temp_command, sizeof(temp_command));
64         if (result == -1) {
65             printf("Error al enviar el comando de temperatura.\n");
66             break;
67         }
68
69         usleep(100000); // Introduce a delay of 100 milliseconds
70
71         unsigned char response[64];
72         int bytes_read = hid_read(device, response, sizeof(response));
73         if (bytes_read > 0) {
74             char response_data[64];
75             memset(response_data, 0, sizeof(response_data));
76             strncpy(response_data, (char *)response, bytes_read);
77
78             float temperature = atof(response_data);
79             printf("\n Temperature: %.2f°C\n", temperature);
80
81             if (temperature >= 1.0 && temperature <= 5.0) {
82                 printf("Warning, low temperature\n");
83             } else if (temperature > 5.0 && temperature <= 15.0) {
84                 printf("Device reaching low temperatures, please be careful\n");
85             } else if (temperature > 15.0 && temperature <= 50.0) {
86                 printf("Unit in good health\n");
87             } else if (temperature > 50.0 && temperature <= 55.0) {
88                 printf("Device reaching high temperatures, please be careful\n");
89             } else if (temperature > 55.0 && temperature < 59.0) {
90                 printf("Warning, high temperature\n");
91             } else if (temperature >= 59.0 || temperature <= 0.9) {
92                 printf("Temperature outside safe range. Stopping program.\n");
93                 pthread_exit(NULL); // Exit the thread
94             }
95         } else {
96             printf("No se recibió ninguna respuesta del dispositivo para verificación de

```

```

    ↪ temperatura.\n");
97     }
98
99     sleep(29); // Sleep for 29 seconds before the next temperature check
100 }
101
102 return NULL;
103 }
104
105
106 // Function to constantly verify the status bits
107 void *status_verification(void *arg) {
108     hid_device *device = (hid_device *)arg;
109
110     while (1) {
111         unsigned char status_command[] = "?";
112         int result = hid_write(device, status_command, sizeof(status_command));
113         if (result == -1) {
114             printf("Error al enviar el comando de estado.\n");
115             break;
116         }
117
118         usleep(100000); // Introduce a delay of 100 milliseconds
119
120         unsigned char response[64];
121         int bytes_read = hid_read(device, response, sizeof(response));
122         if (bytes_read > 0) {
123             char response_data[64];
124             memset(response_data, 0, sizeof(response_data));
125             strncpy(response_data, (char *)response, bytes_read);
126
127             // Check if the first, second, seventh, and eighth bits are 1
128             if (response_data[0] != '1' || response_data[1] != '1' ||
129                 response_data[6] != '1' || response_data[7] != '1') {
130                 printf("Status bits verification failed. Stopping program.\n");
131                 pthread_exit(NULL); // Exit the thread
132             }
133         } else {
134             printf("No se recibió ninguna respuesta del dispositivo para verificación de estado.\n
135 ↪ ");
136         }
137
138         sleep(31); // Sleep for 31 seconds before the next verification
139     }
140
141     return NULL;
142 }
143
144 // Function to send final status commands
145 void send_final_status(hid_device *device) {
146     struct FinalCommand {

```

```

147     char *command;
148     char *text;
149 };
150
151 struct FinalCommand final_commands[] = {
152     {"?", "Status request"},
153     {"T", "Temperature request"},
154     {"V1", "Voltage 1 request"},
155     {"V2", "Voltage 2 request"},
156     {"V3", "Voltage 3 request"},
157     {"V4", "Voltage 4 request"},
158     {"V5", "Voltage 5 request"},
159     {"V6", "Voltage 6 request"},
160     {"V7", "Voltage 7 request"},
161     {"V8", "Voltage 8 request"},
162     {"V9", "Voltage 9 request"},
163 //     {"SF", "Set frequency"},
164     {"R0010", "Request R0010"},
165     {"R0013", "Request R0013"},
166     {"R0014", "Request R0014"},
167     {"R0015", "Request R0015"},
168     {NULL, NULL}
169 };
170
171 for (int i = 0; final_commands[i].command != NULL; i++) {
172     printf("Sent command: %s (%s)\n", final_commands[i].command, final_commands[i].
↪ text);
173
174     unsigned char command_bytes[64];
175     memset(command_bytes, 0, sizeof(command_bytes));
176     strncpy((char *)command_bytes, final_commands[i].command, sizeof(command_bytes)
↪ ) - 1);
177
178     int result = hid_write(device, command_bytes, sizeof(command_bytes));
179
180
181     if (result != -1) {
182         usleep(2000000); // Introduce a delay of 2 seconds
183
184         unsigned char response[64];
185         int bytes_read = hid_read(device, response, sizeof(response));
186         if (bytes_read > 0) {
187             char response_data[64];
188             memset(response_data, 0, sizeof(response_data));
189             strncpy(response_data, (char *)response, bytes_read);
190             printf("Response from device: %s\n", response_data);
191
192             if (final_commands[i].command[0] == '?') {
193                 decodeBits(response_data);
194             } else if (final_commands[i].command[0] == 'f' || final_commands[i].command
↪ [0] == 'F') {
195                 char freq_number[64];

```

```

196         sscanf(response_data, "Frecuencia seteada en %s GHz", freq_number);
197         printf("Response from device: Frecuencia seteada en %s GHz\n",
↪ freq_number);
198     }
199     } else {
200         printf("No response received from the device.\n");
201     }
202     } else {
203         printf("Error sending command.\n");
204     }
205 }
206 }
207
208
209
210 int main() {
211     if (hid_init()) {
212         printf("Error initializing HIDAPI library.\n");
213         return 1;
214     }
215
216     hid_device *device = hid_open(VENDOR_ID, PRODUCT_ID, NULL);
217     if (!device) {
218         printf("Error: Device not found.\n");
219         return 1;
220     }
221
222     printf("Conexión establecida. Puedes enviar comandos.\n");
223
224     pthread_t temp_thread, status_thread;
225     pthread_create(&temp_thread, NULL, temperature_monitor, device);
226     pthread_create(&status_thread, NULL, status_verification, device);
227
228     char command[64];
229     while (1) {
230         printf("Ingrese el comando (o \"exit\" para salir): ");
231         fgets(command, sizeof(command), stdin);
232         command[strcspn(command, "\n")] = 0;
233
234         if (strcmp(command, "exit") == 0) {
235             send_final_status(device);
236             break;
237         } else {
238             unsigned char command_bytes[64];
239             memset(command_bytes, 0, sizeof(command_bytes));
240             strncpy((char *)command_bytes, command, sizeof(command_bytes) - 1);
241
242             int result = hid_write(device, command_bytes, sizeof(command_bytes));
243             if (result == -1) {
244                 printf("Error al enviar el comando.\n");
245             } else {
246                 printf("Comando enviado correctamente: %s\n", command);

```

```

247
248     usleep(100000); // Introduce a delay of 100 milliseconds
249
250     unsigned char response[64];
251     int bytes_read = hid_read(device, response, sizeof(response));
252     if (bytes_read > 0) {
253         char response_data[64];
254         memset(response_data, 0, sizeof(response_data));
255         strncpy(response_data, (char *)response, bytes_read);
256         printf("Respuesta del dispositivo: %s\n", response_data);
257
258         if (strcmp(command, "?") == 0) {
259             decodeBits(response_data);
260         } else if (command[0] == 'f' || command[0] == 'F') {
261             char freq_number[64];
262             sscanf(response_data, "Frecuencia seteada en %s GHz", freq_number);
263             printf("Respuesta del dispositivo: Frecuencia seteada en %s GHz\n",
↪ freq_number);
264         }
265         } else {
266             printf("No se recibió ninguna respuesta del dispositivo.\n");
267         }
268     }
269 }
270 }
271
272 hid_close(device);
273 hid_exit();
274
275 printf("Programa finalizado.\n");
276
277 return 0;
278 }

```

Anexo D. Control de Frecuencia en Oscilador Local Actual

El procedimiento que se realiza en código para controlar la frecuencia que se envía al dispositivo *Agilent* se detalla a continuación.

Código D.1: Código para control de frecuencia del *Agilent*.

```
1 void setsynth(void)
2
3 /* Calculate and set synthesizer sig & ref freqs (synth & rsynth)
4 and sent to synthesizer (sout)*/
5
6 double fsky, synth0, synth0_sub, dsynth, dsynth_sub;
7
8 fsky = frest*(1. - (vlsr + onvant)/CLT);
9
10 synth0 = (fsky + sband*f1_if) - LOCIF;
11 dsynth = offrq*sband/2.;
12
13 synth = (synth0 + dsynth)/(harm*HARM2);
14 rsynth = (synth0 - dsynth)/(harm*HARM2);
15
16 /* Chile uses subharmonic mixer, so we added __sub variables
17 __sub variables set the frequency synthesizer, normal variables
18 are used in velocity scale calculations */
19
20 synth0_sub = (fsky + sband*f1_if)/2 - LOCIF;
21
22 dsynth_sub = dsynth/2;
23
24 synth_sub = (synth0_sub + dsynth_sub)/(harm*HARM2); // Changed for Chile sub
↪ harmonix mixers
25 rsynth_sub = (synth0_sub - dsynth_sub)/(harm*HARM2); // Changed for Chile sub
↪ harmonix mixers
26
27 SOUT();
28
29 tvflag = 1; /* monitor needs updating */
```

Así, se puede trabajar con sistemas de ecuaciones que reflejan el control de la frecuencia en el *Agilent*. Primero, se determina la frecuencia del objeto en el cielo con la ecuación D.1.

$$f_{sky} = f_{rest} \cdot \left(1 - \frac{vlsr - onvant}{CLT}\right) \quad (D.1)$$

Luego, se realiza una corrección debido al desplazamiento en 60 MHz del oscilador YIG, lo cual se muestra en la ecuación D.2.

$$synth_0 = (f_{sky} + sband \cdot f_{1-IF}) - LOCIF \quad (D.2)$$

En caso de que durante la observación se utilice el método de *frequency switching*, se

define una variable auxiliar $dsynth = (of\text{frq} \cdot sband)/2$. Finalmente, la frecuencia que se envía al sintetizador *Agilent* se define por medio de la variable $synth$, mientras que la variable $rsynth$ define la frecuencia de referencia que se envía al dispositivo, lo cual se muestra en las expresiones de la ecuación D.3.

$$synth = \frac{synth_0 + dsynth}{harm \cdot HARM2}, \quad rsynth = \frac{synth_0 - dsynth}{harm \cdot HARM2} \quad (D.3)$$

Como en el *SMWT* se utilizan mezcladores sub armónicos, las expresiones anteriores deben ser levemente ajustadas para configurar la frecuencia del *Agilent*. En este caso, la corrección que se realiza queda definida por la expresión D.4.

$$synth_{0-sub} = \left(\frac{f_{sky} + sband \cdot f_{1-IF}}{2} \right) - LOCIF \quad (D.4)$$

Utilizando que $dsynth_{sub} = dsynth/2$, se pueden re definir las expresiones que configuran la frecuencia del *Agilent*.

$$synth_{sub} = \frac{synth_{0-sub} + dsynth_{sub}}{harm \cdot HARM2}, \quad rsynth_{sub} = \frac{synth_{0-sub} - dsynth_{sub}}{harm \cdot HARM2} \quad (D.5)$$

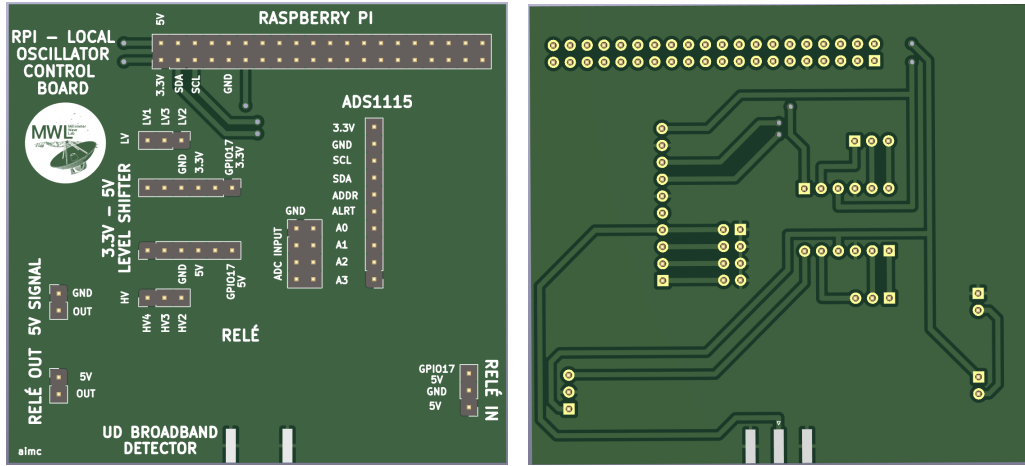
En conclusión, la expresión D.5 define la frecuencia que se envía desde el sistema de control hacia el sintetizador *Agilent*. A continuación se muestra una tabla D.1 con las definiciones de variables mostradas anteriormente en el código.

Tabla D.1: Definiciones de variables de interés para el control de frecuencia.

Variable	Descripción
<i>vlsr</i>	Velocity center of band [km/s]
<i>onvant</i>	Velocity of the antenna [km/s]
<i>restfreq</i>	Frecuencia en reposo
<i>harm</i>	Armónico con el que se observa = 10
<i>CLT</i>	Velocidad de la luz $3 \cdot 10^5$ [km/s]
<i>sband</i>	Lado de la banda a la que se observa. Puede ser -1 o 1
<i>LOCIF</i>	Corrección que se utiliza para el $LO = 60$ [MHz]
<i>of\text{frq}</i>	Frequency switching offset = 0
<i>HARM2</i>	Armónico = 2
f_{1-IF}	Frecuencia a la que opera $IF = 250$ [MHz]

Anexo E. Placa de Control RPI-LO, Versión 2

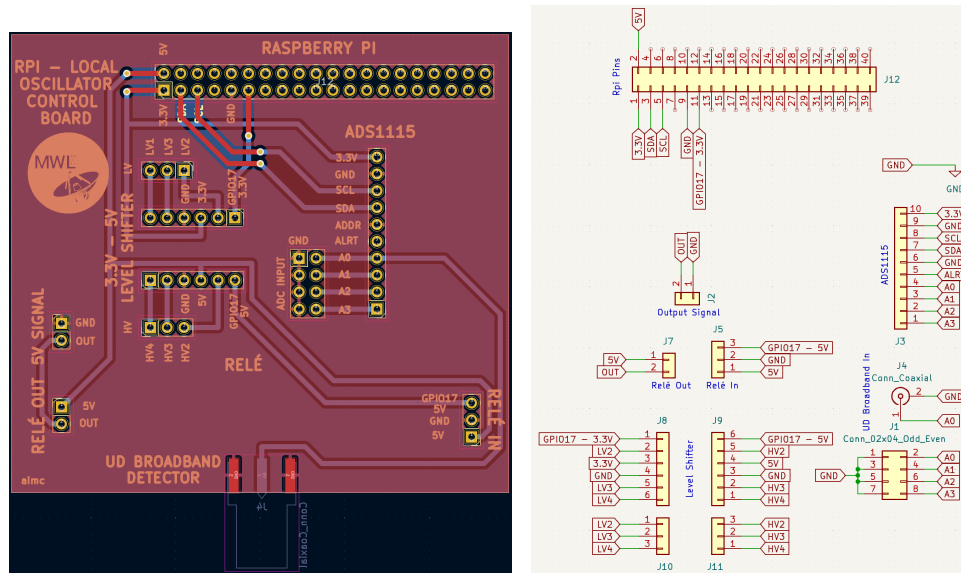
La segunda versión de la placa del sistema de control RPI-LO incorpora modificaciones en el tamaño, con el objetivo de minimizar el espacio que utilizan los componentes y la propia placa. En las siguientes figuras E.1.b y E.1.b se muestra el diseño en 3D de la placa.



(a) Placa RPI-LO, versión 2. Vista Frontal. (b) Placa RPI-LO, versión 2. Vista Trasera.

Figura E.1: Vista 3D frontal y trasera de la Placa RPI-LO, versión 2.

En las siguientes figuras E.2.a y E.2.b se muestran los diagramas PCB y esquemáticos, respectivamente, de la placa RPI-LO.



(a) PCB de la Placa RPI-LO, versión 2. (b) Esquemático de la Placa RPI-LO, versión 2.

Figura E.2: PCB y esquemático de la Placa RPI-LO, versión 2.