Escuela Politécnica Superior de Linares

Grado en Ingeniería de Tecnologías de Telecomunicación

UNIVERSIDAD DE JAÉN
Escuela Politécnica Superior de Linares

Trabajo Fin de Grado

AUTOMATIZACIÓN DE MEDIDAS USANDO ROBOT FRAMEWORK

Alumno: Raja Emran Arif Martinez

Tutor: Prof. D. Sebastián García Galán
Tutor: Prof. D. Francisco José Rodríguez Serrano
Depto.: Departamento de Ingeniería de Telecomunicaciones

Febrero, 2024

AGRADECIMIENTOS

Agradecimientos a mis padres por respaldar todas y cada una de mis decisiones, en especial a mi madre Isabel por la eterna confianza depositada en mí y por haberme dado la mejor educación y vida que ha podido. A mis hermanos, Saida y Amer que siempre han hecho todo lo posible para ser un punto de apoyo para mí. A mi pareja, Elena por brindarme esta oportunidad única de formarme, a pesar de no tener ya 18 años. Por su paciencia y apoyo en los momentos más cruciales. A mi tutor D. Sebastián García Galán por ayudarme a alzar el vuelo tanto en el ámbito de este proyecto, como fuera de él. A mi tutor D. Francisco José Rodríguez Serrano por guiarme de la mejor manera posible. Al profesor D. José Enrique Muñoz Expósito por dedicar su tiempo siempre que lo he requerido y por aportarme conocimientos y consejos que para mí tienen un valor muy significativo. También agradezco al resto de profesorado del departamento de teoría de la señal y departamento de telemática por sus aportaciones a lo largo de este "viaje", todos y cada uno de ellos han aportado su grano de arena no solo en el ámbito docente, sino también en el personal con conocimientos y consejos.

A mis abuelos José e Isabel, los cuales son responsables de que sea la persona que soy a día de hoy. Dotándome de la capacidad de afrontar cual reto se presente. En general a toda mi familia, que tantas alegrías me han aportado al igual que su confianza incondicional a lo largo de estos años.

Agradezco a todos y cada uno de mis compañeros, momentos, vivencias y muchas horas de estudio en conjunto con un único fin, formarnos como ingenieros de telecomunicación mientras disfrutábamos de esta etapa, la cual considero necesaria y muy enriquecedora.

Y por último a mis mascotas Blue y Barry, que cuando los momentos parecían superarme, un paseo largo y tendido con ellos siempre me llevaban de nuevo a un estado de paz y recuperación.

Raja Emran Arif Martinez

Linares, 2024

RESUMEN

El desarrollo de este proyecto se basa en la automatización de procesos de medida con instrumentos de laboratorio con el fin de facilitar el acceso a los equipos de laboratorio de manera remota. Mediante conexión Ethernet (LAN), se pretende enviar comandos de ejecución para automatizar procesos de medida de parámetros, relacionados con asignaturas del grado de ingeniería de tecnologías de telecomunicación como fundamentos radio y fundamentos de microondas.

En este caso el fabricante de los instrumentos de medida que se utilizan durante las distintas fases del proyecto es Keysight, antiguamente conocido como Agilent. Los modelos utilizados para los distintos casos de medida son los siguientes. Analizador de redes vectorial, Keysight E5063A, analizador de espectros, Keysight N9320B y generador de funciones Keysight N5171B.

El principal proceso que se desarrollará a lo largo de este documento es la automatización de distintas medidas utilizando comandos SCPI, con PYTHON como lenguaje de programación y usando ROBOT FRAMEWORK como estructura de automatizado. Se elaborará un proyecto software donde se alojarán distintos scripts de código para los distintos casos de prueba.

Finalmente, mediante la estructura de automatizado ya mencionada anteriormente se persigue desvincular la presencia física en el laboratorio del profesorado o alumnado. Facilitando el proceso de pruebas mediante una conexión remota a las instalaciones y equipo de instrumentación. El escenario de desarrollo de este proyecto es el laboratorio L-137, laboratorio de radiofrecuencia y microondas de la EPSL.

ABSTRACT

The development of this project is based on the automation of measurement processes using laboratory instruments, aiming to facilitate remote access to laboratory equipment. Through Ethernet connection (LAN), the intention is to send execution commands to automate measurement processes of parameters related to subjects within the Telecommunication Technologies Engineering degree, such as Radio Fundamentals and Microwave Fundamentals.

In this case, the manufacturer of the measurement instruments utilized throughout various project phases is Keysight, formerly known as Agilent. The models employed for different measurement scenarios are as follows: Vector Network Analyzer, Keysight E5063A; Spectrum Analyzer, Keysight N9320B; and Function Generator, Keysight N5171B.

The primary focus of this document is the automation of diverse measurements using SCPI commands, with PYTHON as the programming language and employing ROBOT FRAMEWORK as the automation structure. A software project will be developed, housing various code scripts for different test cases.

Ultimately, through the aforementioned automation structure, the objective is to decouple the physical presence of faculty or students in the laboratory. This is achieved by facilitating the testing process through remote connection to the facilities and instrumentation equipment. The development scenario for this project is Laboratory L-137, the Radiofrequency and Microwave Laboratory of the EPSL.

Universidad de Jaén

Escuela Politécnica Superior de Linares

Trabajo Fin de Grado

AUTOMATIZACIÓN DE MEDIDAS USANDO ROBOT FRAMWORK

Firma autor:

ARIF MARTINEZ RAJA EMRAN - 26241295C

Firmado digitalmente por ARIF MARTINEZ RAJA EMRAN - 26241295C
Fecha: 2024.02.12 12:52:46 +01'00'

Febrero, 2024

ÍNDICE

1 INTRODUCCIÓN 18
1.1 MOTIVACIÓN 19
1.2 OBJETIVOS 19
2 ORGANIZACIÓN DEL DOCUMENTO 20
3 ESTADO DEL ARTE 22
3.1 INTERFACES HPIB y GPIB 22
3.2 SCPI 26
3.3 AUTOMATIZACIÓN 28
4 MATERIALES: SOFTWARE-HARDWARE-COMPONENTES 29
4.1 RECURSOS SOFTWARE 29
4.1.1 NI PACKAGES 29
4.1.2 PYCHARM (IDE) 29
4.1.3 PYTHON (versión 3.7) 30
4.1.4 VISA 30
4.1.5 NI-VISA 32
4.1.6 ROBOT FRAMEWORK 33
4.1.7 PuTTY 36
4.1.8 PiFmRds 37
4.1.9 WIRESHARK 38
4.2 RECURSOS HARDWARE 38
4.3 COMPONENTES Y DUTS 43
4.3.1 ANTENA DIRECTIVA YAGI-UDA CON RADOMO 44

4.3.2 ANTENA BICONICA 44
4.3.3 ANTENA YAGI-UDA 45
4.3.4 FILTROS (Lp, Hp, Bp) 45
4.3.5 KIT DE CALIBRACIÓN MANUAL 47
4.3.6 CABLEADO TIPO N Y COAXIAL TIPO BNC 48
5 FUNDAMENTOS TEORICOS 49
5.1 COMANDOS SCPI 49
5.1.1 COMANDOS ESPECIFICOS PARA EQUIPOS KEYSIGHT 52
5.1.2 ARBOLES JERARQUICOS DE SUBSISTEMAS SCPI 53
5.2 CONCEPTUALIZACIÓN CON ROBOT FRAMEWORK 54
5.2.1 SINTAXIS GENERICA DE LOS TESTS 54
5.2.2 FORMATO DE ESPACIADO 55
5.2.3 CABECERA SETTINGS 55
5.2.4 CABECERA VARIABLES 56
5.2.5 CABECERA TEST CASES 56
5.2.6 CABEZERA KEYWORDS 56
5.2.7 DESGLOSE REPRESENTATIVO DE SINTAXIS 57
5.2.8 GENERACIÓN DE REPORTES E INFORMES 59
5.3 CONCEPTOS TEORICOS DE LOS CASOS DE PRUEBA 61
5.3.1 CASO 1. Genera una señal 61
5.3.2 CASO 2. Genera un barrido en frecuencia 63
5.3.3 CASO 3. Genera una modulación AM 64
5.3.4 CASO 4. Cálculo de parámetro S11. (Una única frecuencia) 66
5.3.5 CASO 5. Cálculo de parámetro S11. (barrido en frecuencia) 68
5.3.6 CASO 6. Respuesta en frecuencia de un filtro (LP, BP, HP) 69
5.3.7 CASO 7. Cálculo Parámetros de Scattering (4 parámetros) 71

5.3.8 CASO 8. Caracterización de Impedancia de un DUT 71
5.3.9 CASO 9. Transmisión de audio y Demodulación FM 73
6 DESARROLLO 74
6.1.1 CASO 1. Proceso de Generación de una señal 76
6.1.2 CASO 2. Proceso de Generación de Barrido en Frecuencia 81
6.1.3 CASO 3. Proceso de Modulación AM 83
6.1.4 CASO 4. Proceso de Calculo Parámetro S11 87
6.1.5 CASO 5. Proceso de Cálculo Parámetro S11 mediante barrido 91
6.1.6 CASO 6. Proceso de Respuesta en Frecuencia de un filtro 93
6.1.7 CASO 7. Proceso de Cálculo de Parámetros [S] 96
6.1.8 CASO 8. Proceso de Caracterización de Impedancia 98
6.1.9 CASO 9. Transmisión de audio y Demodulación FM 100
7 RESULTADOS 104
7.1 Resultados Caso 1. 104
7.2 Resultados Caso 2. 107
7.3 Resultados Caso 3. 109
7.4 Resultados Caso 4. 110
7.5 Resultados Caso 5. 111
7.6 Resultados Caso 6. 113
7.7 Resultados Caso 7. 115
7.8 Resultados Caso 8. 116
7.9 Resultados Caso 9. 118
7.10 Resultados Caso ejemplo de error. 120
7.11 Análisis de tráfico mediante capturas con Wireshark 122
8 CONCLUSIONES 124
9 LINEAS FUTURAS 125

10 ABREVIATURAS 127

11 BIBLIOGRAFÍA 129

ÍNDICE DE ILUSTRACIONES

1. GPIB conector físico redondo.[2] 23
2. Analizador de redes 3570A conectado a 9820A calculador con HPIB. 1972.[2] 23
3. Lay-out de pines para HPIB GPIB. [2] 24
4. Conector GPIB (a) norma europea. Conector (IEC) GPIB (b) norma americana (IEEE). [3] 24
5. Interfaz GPIB más reciente. [4] 25
6. Jerarquía SCPI. [5] 27
7. NI Package Manager Interface. 29
8. IDE Pycharm CommunityVersion. Entorno de desarrollo. 30
9. Lenguaje de programación utilizado. 30
10. Esquema de proceso y conexión virtual con VISA. [8] 31
11. Proceso y componentes necesarios para la interconexión. [9] 32
12. Aplicaciones de Robot Framework. [10] 33
13. Arquitectura modular de Robot Framework 36
14. Interfaz de terminal PuTTY 37
15. Logotipo de Wireshark. [14] 38
16. Ordenador portátil utilizado como controlador. 39
17. Cable Ethernet CAT-6 39
18. Raspberry Pi 4 modelo B. 40
19. Generador de funciones Keysight N5171B. 40
20. Salida tipo N del N5171B. 41
21. Interfaces de conexión del N5171B. 41
22. Analizador de espectros Keysight N9320B. 42
23. Entrada RF tipo N del N9320B. 42
24. Analizador de redes vectorial Keysight E5063A. 43
25. Antena Yagi-Uda con radomo protector. 44
26. Antena bicónica montada en un trípode (soporte). 45
27. Antena tipo Yagi. 45
28. Filtro paso bajas con frecuencia de corte a 900MHz. 46

29. Filtro paso banda entre 63-77 MHz. 46
30. Filtro paso altas a frecuencia 1000-3000 MHz. 47
31. Kit de calibración compatible con VNA E5063A. 47
32. Circuito abierto, corto circuito y carga de 50 Ohm. 48
33. Cable coaxial con conectores tipo BNC. 49
34. Cableado con conectores tipo N (macho y hembra). 49
35. Sintaxis para comandos SCPI subsistema. [5] 52
36. Jerarquía de subsistemas en VNA E5063A. 54
37. Ejemplo de espaciado (extraído de Robot Framework user guide) [12] 55
38. Ejemplo de sintaxis y estructura de un caso de prueba. 57
39. Ejemplo de una variable. 58
40. Importando una librería externa. 58
41. Tarea 1, donde se visualizan las distintas secciones. 58
42. Ejemplo de Log file. 59
43. Ejemplo de Report file. 60
44. Ejemplo de Output file. 60
45. Señal triangular. 62
46. Señal senoidal. 62
47. Señal cuadrada. 62
48. Ajuste de parámetros en el script de Robot Framework. 64
49. Esquema modulación AM. 65
50. Ondas de potencia en un bipuerto genérico cualquiera. [21] 68
51. Detalle de un filtro LP. Nivel de rizado. [24] 70
52. Esquema modulación FM. 74
53. Laboratorio de Radiofrecuencia y Microondas, EPSL. (L-137) 76
54. Diagrama de flujo para el Caso 1. 76
55. Osciloscopio Keysight MSO-X 3014A. 77
56. Salida en terminal con dirección IP encontrada. 78
57. Salida en el terminal con el objeto de interacción creado. 79
58. línea de comando enviada para ajustar valor de frecuencia. 79
59. Conexión entre equipos para ver señal generada y enviada. Caso 1. 81
60. Sintaxis utilizada para ajustar modo de generación de frecuencia (barrido). 82

61. Diagrame de flujo y orden de envíos comandos SCPI caso 2. 82
62. Script con ajuste de parámetros de barrido. 83
63. Parámetros ajustados automáticamente para el barrido. Caso 2. 83
64. Conexiones disponibles en panel frontal MSO-X 3014A. 84
65. Conexiones disponibles parte trasera MSO-X 3014A. 85
66. Diagrama de flujo jerarquizado para el caso 3. 85
67. Parámetros de modulación AM. 86
68. Señal con modulación AM aplicada. 86
69. Señal con modulación AM no aplicada. 87
70. Diagrama de flujo jerárquico para el caso 4. 89
71. Antena Yagi-Uda con radomo siendo testeada. Caso 4. 89
72. Antena Biconica siendo testeada. Caso 4. 90
73. Antena Yagi-Uda a 2.45 GHz siendo testeada. 90
74. Diagrama de flujo jerarquizado caso 5. 92
75. Antena Yagi con radomo siendo testeada. Caso 5. 93
76. Antena Yagi-Uda a 2.45 GHz siendo testeada. Caso 5. 93
77. Respuesta en frecuencia de un filtro LP. 94
78. Respuesta en frecuencia de un filtro BP. 94
79. Respuesta en frecuencia de un filtro HP. 95
80. Diagrama de flujo jerárquico caso 6. 96
81. Cálculo de parámetros [S]. Caso 7. 97
82. Diagrama de flujo jerárquico caso 7. Desarrollo del parámetro S11 en una única frecuencia. 98
83. Caracterizando impedancia de entrada de antena Yagi-Uda. Caso 8. 99
84. Diagrama de flujo jerárquico caso 8. 100
85. Interfaz IP Scanner. 101
86. Interfaz de control PuTTY. 102
87. Cambio de directorio dentro de la Raspberry. 102
88. Inicio de transmisión FM. 102
89. Demodulación FM. Caso 9. 103
90. Diagrama de flujo jerárquico caso 9. 103
91. Log correspondiente al caso 1. 105

92. Reporte correspondiente al caso 1. 105
93. Tarea 1 completada obteniendo dirección IP. 106
94. Grafica resultante al ejecutar el script del caso 1. 106
95. Señal generada en el caso 1. 107
96. Log correspondiente al caso 2. 107
97. Report correspondiente al caso 2. 108
98. Detalle de tarea 4 para el caso 2. 108
99. Parámetros recibidos por el generador de señal N5171B. Caso 2. 108
100. Log correspondiente al caso 3. 109
101. Report correspondiente al caso 3. 109
102. Detalle de tarea 4 para el caso 3. 110
103. Log correspondiente al caso 4. 110
104. Report correspondiente al caso 4. 111
105. Detalle de la tarea 5 del caso 4. 111
106. Log correspondiente al caso 5. 112
107. Report correspondiente al caso 5. 112
108. Grafica resultante del caso 5. 113
109. Log correspondiente al caso 6. 113
110. Report correspondiente al caso 6. 114
111. Grafica resultante de un filtro BP caso 7. 114
112. Log correspondiente al caso 7. 115
113. Report correspondiente al caso 7. 116
114. Detalle de la tarea 5 del caso 7. 116
115. Log correspondiente al caso 8. 117
116. Report correspondiente para el caso 8. 117
117. Grafica resultante de caso 8. 118
118. Log correspondiente al caso 9. 119
119. Detalle de la tarea 4 para el caso 9. 119
120. Report correspondiente al caso 9. 119
121. Detección de la señal transmitida en el analizador de espectros. 120
122. Ejemplo de Log correspondiente a un test fallido. 120
123. Ejemplo de Report correspondiente a un test fallido. 121

124. Tarea 1 fallida debido a error de conexión con instrumento E5063A (VNA). 121
125. Filtro de búsqueda en Wireshark por dirección IP y protocolo. 122
126. Parte del tráfico de paquetes capturado en el caso 1. 123
127. Información contenida en paquete o trama 373..... 123
128. Parte del tráfico capturado en el caso 9. 124
129. Información contenida en el paquete o trama 441..... 124

ÍNDICE DE TABLAS

Tabla 1. Comandos comunes SCPI. IEEE 488.2 51
Tabla 2. Tabla de caracteres especiales de la sintaxis SCPI. 52
Tabla 3. Sintaxis y apartados. 54

MEMORIA

1 INTRODUCCIÓN

El enfoque principal de este proyecto radica en la conexión remota con laboratorios, específicamente aquellos dedicados a experimentación y ejecución de medidas con instrumentación de medición, predominantemente en el área de las telecomunicaciones. Se busca implementar una solución que permita la interacción remota a través de redes LAN (Local Area Network), empleando diversas herramientas para la transmisión, procesamiento y comunicación entre el instrumento y el usuario. En este contexto, se incorpora el uso de comandos SCPI (Standard Commands for Programmable Intruments), los cuales fueron desarrollados basándose en el estándar IEEE 488.2. Estos comandos estandarizados se diseñan con el fin de ser utilizados en comunicaciones con equipos de laboratorio de diversos fabricantes, como Tektronix, Agilent, Keysight entre otros. Estos comandos SCPI son esenciales para configurar mediciones, leer datos y realizar funciones específicas del equipo de manera uniforme.

La imperante necesidad de la automatización de procesos y medidas se aborda mediante la elección de Python como el lenguaje de programación principal. La versatilidad y el amplio soporte de Python en la comunidad de desarrollo facilitan la manipulación de datos, la comunicación en red y la automatización de tareas, garantizando así un control eficiente de la instrumentación de laboratorio. La utilización de Python no solo simplifica el desarrollo, sino que también proporciona flexibilidad para adaptarse a los cambios y evoluciones del proyecto y las tecnologías involucradas.

Para estructurar y organizar las pruebas y tareas automatizadas, se adopta Robot Framework como el framework de automatización. Este framework de código abierto destaca por su sintaxis fácil de entender y la capacidad de crear scripts de prueba legibles por humanos. Su arquitectura extensible permite la incorporación de bibliotecas personalizadas, adaptando la automatización a las necesidades específicas del proyecto. La combinación de comandos SCPI, Python y Robot Framework establece una base sólida para la implementación exitosa de la automatización remota en entornos de laboratorio de telecomunicaciones.

Para la comprobación de la transmisión de información entre equipos de control e instrumentos de laboratorio se emplea el software de análisis de paquetes WireShark. El cual es ampliamente conocido en el sector de las TIC debido a su gran potencialidad a la hora de analizar paquetes de información que componen el tráfico de datos intercambiado entre equipos transmisores y receptores. Este software hace accesible información de tráfico intercambiada por equipos enlazados por una conexión, ya sea un medio guiado o no guiado.

Resumiendo, brevemente el asunto que ocupa este proyecto. Esta ampliamente basado en procesos de conexión entre equipos de manera remota, el análisis de la información intercambiada entre estos y la visualización de los resultados obtenidos con el fin de recabar conclusiones analíticas sobre las magnitudes y datos tratados.

1.1 MOTIVACIÓN

Con el constante avance de las tecnologías, el ritmo de vida experimenta una aceleración notable en el entorno profesional. La creciente necesidad de las empresas de maximizar la productividad de sus profesionales plantea desafíos tecnológicos que, a su vez, requieren soluciones efectivas. Este documento se desarrolla con el propósito de abordar tales desafíos, centrándose en la importancia crítica que ha adquirido el teletrabajo o trabajo remoto en el ámbito laboral, especialmente a raíz de la reciente pandemia mundial. El mero hecho de no requerir de la presencialidad ininterrumpida del personal de pruebas de laboratorio a la hora de realizar medidas que requieran una duración extensa en el tiempo, hace que sea de interés el proceso que se desarrolla en este proyecto. Siendo de gran utilidad la automatización de medidas con instrumentación de laboratorio. Se persigue alcanzar la transversalidad horaria y disponibilidad de los técnicos para aumentar la productividad.

1.2 OBJETIVOS

Se persigue dar accesibilidad a las operaciones con instrumentos de laboratorio de manera remota, siendo la principal meta de este trabajo. Dado que el abanico de interfaces de conexión posibles es amplio, he aquí una de las variantes aplicadas, en cuyo caso involucra la conexión mediante cableado Ethernet a instrumentación de medidas de laboratorio. La pre-programación o predesarrollo de estructuras automatizadas de código

hace que se generalice su aplicación a otros campos científicos y no solo en el ámbito de las telecomunicaciones. Es en este ámbito, donde reside el objetivo de este trabajo. Automatizar procesos o casos prácticos realizables en los laboratorios mediante Robot Framework. El cual posee una potencialidad notable ya que es capaz de generar estructuras comunes para cualquier tipo de caso. A su vez, la generación automática de un reporte de pasos cumplimentados y/o errores de proceso, haciendo más fácil la identificación de posibles problemas o incongruencias que puedan surgir a la hora de realizar medidas.

Por lo que como objetivo final existirá una reducción considerable de la presencialidad del usuario en el laboratorio y a su vez un registro más conciso y extenso de los distintos pasos realizados y resultados obtenidos mediante la automatización de los procesos de medida y transmisión de contenidos.

2 ORGANIZACIÓN DEL DOCUMENTO

A continuación, se desglosa de manera breve pero clara, la composición del documento en si con los distintos apartados que lo componen.

- Como capítulo 1, una introducción donde se persigue de manera efectiva y a la vez clara la descripción general del documento y procesos que involucran este. La cual está compuesta de unos objetivos y una motivación. En el apartado de objetivos se pretende dar luz a los puntos a los cuales se quiere llegar con el desarrollo de este proyecto. En el apartado de motivación se esclarecen las necesidades por las cuales se diseña este proyecto y algunos aspectos relativos a los procesos de laboratorio a los cuales se quiere aplicar la automatización y por qué motivos.
- En el capítulo 2, se desglosa la totalidad de capítulos que componen este proyecto con el fin de servir como guía. A su vez, aportan nuevamente un breve resumen de los contenidos expuestos en la introducción.
- En el capítulo 3, se procede a desarrollar el estado del arte de este proyecto. Con el fin de esclarecer la historia o raíces de las distintas fases involucradas en el proceso de desarrollo. Las cuales vienen a ser las primeras tecnologías implementadas para la conectividad entre equipos controladores e instrumentación de laboratorio. El estándar utilizado ampliamente en el control de equipos e instrumentación de

laboratorio. El cual cabe destacar, nace de la implementación de la tecnología GPIB-HPIB (General Purpose Interface Bus – Hewlett Packard Interface Bus) con el fin de estandarizar un único diccionario de comandos aplicables a distintos fabricantes en el ámbito de la instrumentación de laboratorio. Finalmente se expone la historia que envuelve el termino automatización y se hace referencia al framework que se utiliza para dar cobertura a la automatización de los procesos de medida de laboratorios.

- En el capítulo 4, se exponen los requisitos imprescindibles para el desarrollo, la realización y finalmente culminación de los objetivos finales de este proyecto. En cuyo caso vienen clasificados por requerimientos software, hardware y componentes.
- En el capítulo 5 se desarrollará de manera extensa y clara los conceptos teóricos aplicados a los distintos casos de prueba llevados a cabo en el proceso de este proyecto. Los cuales son el centro de interés del desarrollo ya que serán estos los que realmente demuestren que el objetivo principal del proyecto se ha alcanzado. Siendo este el de automatizar medidas de laboratorio utilizando Robot Framework y el estándar SCPI para el control de equipos instrumentales.
- En el capítulo 6, se procederá a exponer el desarrollo general y conciso de este proyecto. El cual abarcará el detalle de toda la parte experimental, de desarrollo de código y finalmente la aplicación de la plataforma de automatizado para las medidas y casos en concreto. Dando un grado de importancia al desarrollo y el entorno de trabajo con todos los componentes involucrados en el diseño, implementación y posteriores pruebas.
- En el capítulo 7, se mostrarán los resultados obtenidos en cada caso de prueba y brevemente se analizarán con intención de su posterior justificación.
- En el capítulo 8, se exponen conclusiones obtenidas tras la implementación de las distintas pruebas y casos realizados. Dando mayor visibilidad al análisis obtenido tras las 9 medidas de prueba que se han desarrollado durante la duración del proyecto en sus distintas fases.
- En el capítulo 9, se desglosará de manera objetiva líneas futuras aplicables a esta técnica de automatización y sobre que otras interfaces de conexión se puede aplicar con el fin de dar una más amplia aplicación al objetivo principal de este proyecto.

- En el capítulo 10, Se detalla un listado de abreviaturas utilizado durante el desarrollo de este documento con el fin de facilitar la comprensión al lector/es.
- En el capítulo 11, último capítulo de este proyecto. Se listan las referencias bibliográficas. Las cuales han sido utilizadas para describir los distintos componentes y técnicas utilizadas con el fin de certificar la veracidad del desarrollo de este proyecto.

3 ESTADO DEL ARTE

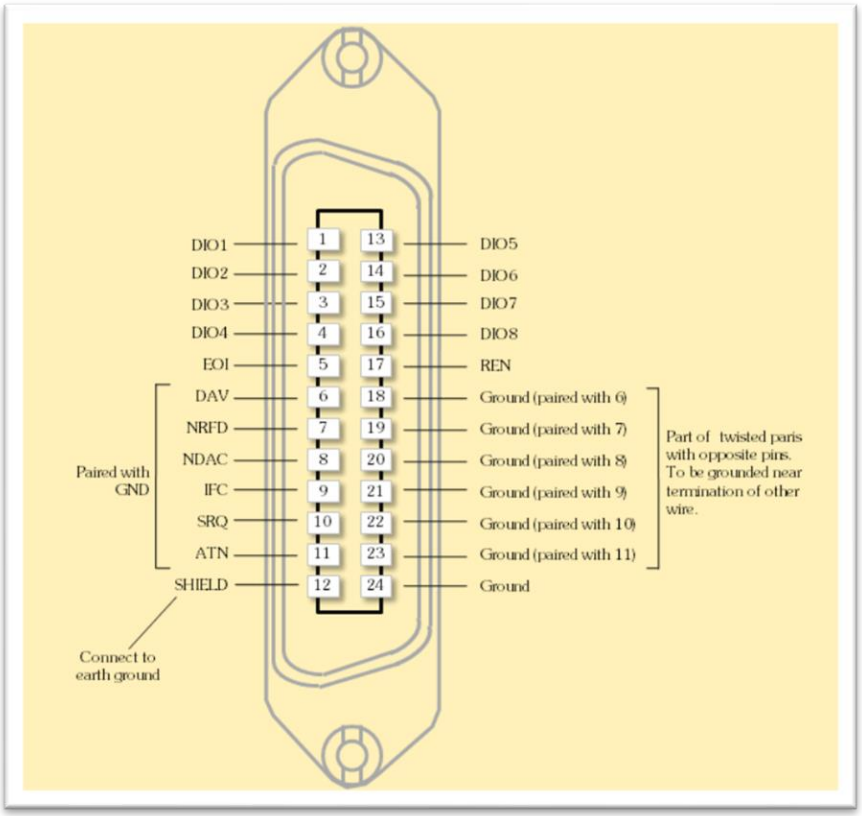
Para tener una idea o referencia más clara del punto de partida de este proyecto, a continuación, en el apartado que procede se dispone a detallar algunos aspectos históricos de las distintas fases y componentes que ocupan interés en este proyecto. Siendo de vital importancia el conocimiento de la tecnología de interfaz de conexión física entre instrumentación de medidas y equipos de control HP-IB, posteriormente evolucionada a GP-IP.

3.1 INTERFACES HPIB y GPIB

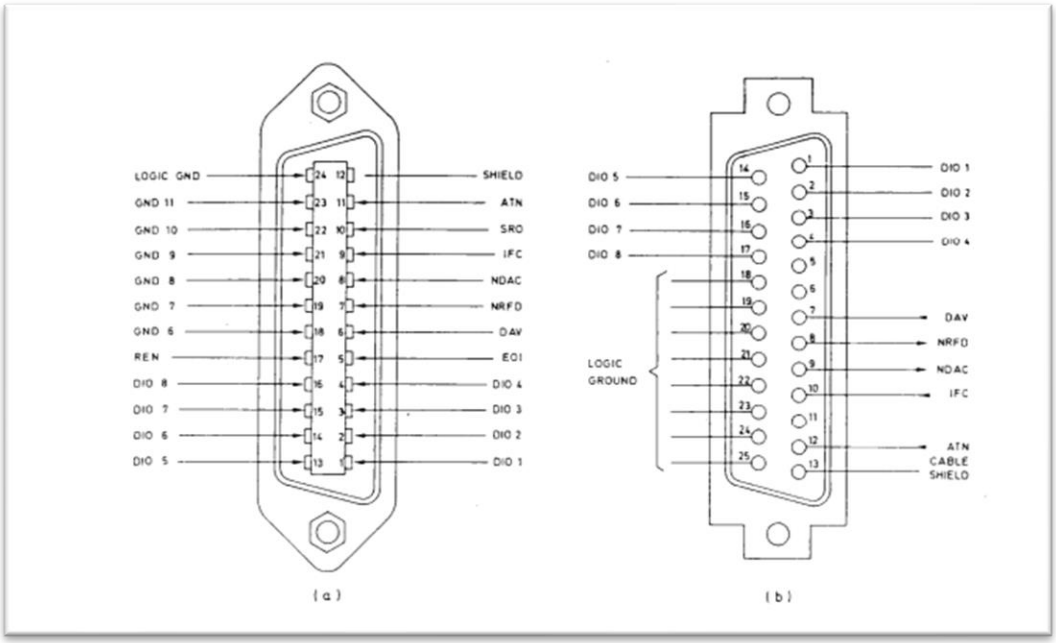
La historia del HP-IB comenzó en el año 1965, Hewlett-Packard ya era uno de los líderes en suministro de instrumentos de medición, y la adquisición de datos digitales se volvía un tema cada vez más importante. Desde el primer contador de frecuencia conectado a un grabador digital a mediados de la década de 1950, la demanda aumentó también para la programación remota de instrumentos en la década de 1960. Durante finales de la década de 1960, ya existía mucha experiencia con sistemas de bus para interconectar unidades centrales de procesamiento y componentes internos, y también diversos instrumentos y controladores estaban interconectados mediante tarjetas de interfaz de "puenteo" insertadas en un común backplane. Así que los ingenieros de HP ya tenían buena práctica en el diseño de sistemas internos de bus. El paso final desde el backplane hacia una red basada en cables estaba a la mano. [1]

1. GPIB conector físico redondo.[2]

2. Analizador de redes 3570A conectado a 9820A calculador con HPIB. 1972.[2]



3. Lay-out de pines para HPIB GPIB. [2]



4. Conector GPIB (a) norma europea. Conector (IEC) GPIB (b) norma americana (IEEE). [3]

5. Interfaz GPIB más reciente. [4]

En el transcurso del desarrollo y crecimiento de la tecnología, en la década de los 60 surgieron necesidades que debían ser cubiertas por las empresas desarrolladoras de software y hardware. Entre estas, Hewlett Packard lanzo HP-IB el cual era una interfaz de conexión capaz de intercomunicar equipos de control con instrumentación de laboratorio. Su utilidad principal fue la de interconectar y controlar instrumentación programable que el fabricante estaba desarrollando. Con el nacimiento de controladores digitales y equipos de prueba programables la necesidad de un estándar aumento.

A fines de la década de 1960, Hewlett-Packard desarrolló el GPIB (también conocido como HP-IB) para la conexión y control de instrumentos programables fabricados por la misma compañía. La evolución hacia controladores digitales y equipos de prueba programables generó la necesidad de una interfaz estándar de alta velocidad para la comunicación entre instrumentos y controladores de diversos fabricantes. En 1975, el IEEE (Institute of Electrical and Electronic Engineers) estableció el Estándar ANSI/IEEE 488-1975, definiendo las especificaciones mecánicas, eléctricas y funcionales del sistema de interfaz. Una revisión del IEEE 488-1975 se llevó a cabo en 1978, principalmente para clarificaciones editoriales y adiciones. Este protocolo, reconocido globalmente, es conocido por tres nombres:

- General Purpose Interface Bus (GPIB)
- Hewlett-Packard Interface Bus (HP-IB)
- IEEE 488 Bus

Dado que el documento original del IEEE 488 carecía de directrices para una sintaxis preferida y convenciones de formato, se continuó trabajando en la especificación para mejorar la compatibilidad y configurabilidad del sistema en entornos de prueba. Este esfuerzo resultó en el estándar suplementario IEEE 488.2, denominado Códigos, Formatos, Protocolos y Comandos Comunes, diseñado para trabajar en conjunto con el IEEE 488 (rebautizado como IEEE 488.1). Cabe destacar que IEEE 488.2 no sustituye a IEEE 488.1, ya que muchos dispositivos aún adhieren exclusivamente a este último. IEEE 488.2 amplía las capacidades de interfaz de dispositivo, establece códigos y formatos de datos comunes, define un protocolo de mensajes de dispositivo, presenta un conjunto genérico de comandos necesarios y propone un nuevo modelo de informe de estado.

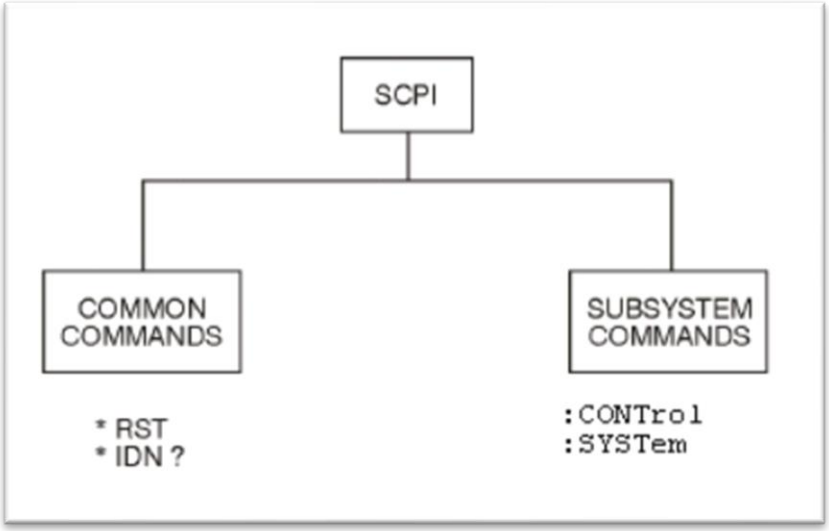
3.2 SCPI

En 1990, la especificación IEEE 488.2 incorporó el documento **Standard Commands for Programmable Instrumentation** (SCPI). SCPI define comandos específicos que cada clase de instrumento, generalmente abarcando dispositivos de diversos fabricantes, debe seguir. De esta manera, SCPI garantiza completa compatibilidad y configurabilidad del sistema entre estos instrumentos, eliminando la necesidad de aprender conjuntos de comandos diferentes para cada instrumento en un sistema compatible con SCPI. Además, facilita la sustitución de un instrumento de un fabricante por otro instrumento de otro fabricante.

El estándar SCPI IEEE-488.2 es una especificación técnica que, desde su introducción en 1990, ha desempeñado un papel fundamental en la normalización de la comunicación entre instrumentos programables. Este estándar, una extensión del protocolo IEEE-488.2, establece un conjunto uniforme de comandos para la interfaz entre instrumentos de medición y prueba, independientemente del fabricante. Al proporcionar una capa de abstracción, SCPI simplifica la programación y el control de instrumentos dentro de un sistema, garantizando la compatibilidad y configurabilidad. Centrándose en aspectos clave como la configuración, calibración, adquisición de datos y control remoto, SCPI se ha

convertido en un marco fundamental para la estandarización de la instrumentación programable. Los fabricantes que adoptan SCPI se comprometen a seguir un conjunto coherente de comandos y estructuras de datos. Esta uniformidad facilita la interoperabilidad entre instrumentos de diferentes proveedores, simplificando el desarrollo de software y mejorando la eficiencia en entornos que utilizan una variedad de instrumentos. [5]

La estructura de comandos SCPI ha sido diseñada con instrucciones de formato legible tanto por humanos como por máquinas. Esta característica permite una fácil comprensión y programación, facilitando la implementación de SCPI en diversas aplicaciones. Además, las revisiones periódicas del estándar han permitido abordar las cambiantes necesidades y tecnologías emergentes, asegurando la relevancia continua de SCPI en el ámbito de la instrumentación y su capacidad para adaptarse a entornos de prueba modernos y complejos.



6. Jerarquía SCPI. [5]

En la ilustración anterior podemos visualizar en que clasificación se basan los dos niveles de comandos SCPI. Existe el grupo de comandos denominados “Comandos comunes” los cuales son comandos raíz que ejecutan de manera directa acciones predeterminadas como por ejemplo pueden ser (RST) o (IDN). El formato correcto de implementación es un asterisco (*) seguido del comando raíz. Este tipo de comandos no exigen de comandos consecutivos y suelen ejercer una única acción sobre el instrumento. Seguidamente se puede ver la categoría de “comandos de subsistema”. Estos comandos

están diseñados para controlar funciones específicas de instrumentos y equipos de medición. Se emplea un enfoque que permite una interacción detallada con el equipo, destacando su utilidad en la personalización y control preciso de sus capacidades. Este análisis busca proporcionar una comprensión profunda de la implementación de los comandos de subsistema en los ámbitos de la investigación científica y tecnológica, contribuyendo así al conocimiento en este campo. En el apartado de desarrollo se procederá a extender dicha jerárquica entre comandos al igual que listar en su totalidad los comandos estándar aceptados por los fabricantes de equipos.

3.3 AUTOMATIZACIÓN

Previo a la amplia disponibilidad de componentes electrónicos, después de la segunda guerra mundial la automatización fue construida por usuarios finales y diseñada para tareas específicas. La mención más temprana de la palabra automatización en la literatura estadounidense data de 1875. Esta describe la automatización del lavado y esterilización de equipo de laboratorio dedicado a la investigación química. En los años siguientes, el desarrollo y posterior comercialización de equipos autómatas fue creciendo de manera lenta pero notable. Grandes amoladoras para la preparación de muestras de carbón son un ejemplo. Durante la primera guerra mundial la necesidad creciente y el crecimiento industrial de los años 1920´s llevaron a la necesidad de equipos de testeo autómatas. En los años 50, la automatización de la titulación englobaba dispositivos coulomb métricos, potenciométricos y fotométricos. La combinación de registradores gráficos, células fotoeléctricas y temporizadores dio lugar a la creación de otros equipos automatizados, como destiladores y recolectores de fracciones. La primera forma genuina de automatización independiente para laboratorios incluyó analizadores de química clínica, los cuales fueron introducidos en la misma década. [6]

Una vez puestos en contexto y habiendo expuesto la historia más reciente de los tres factores principales que ocupan este proyecto se pasará a detallar los recursos utilizados de manera detallada.

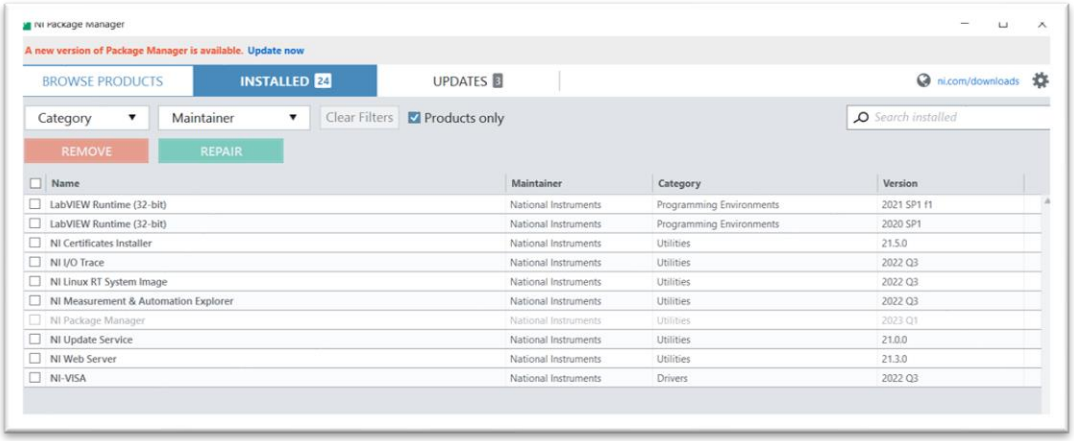
4 MATERIALES: SOFTWARE-HARDWARE-COMPONENTES

4.1 RECURSOS SOFTWARE

En este apartado se procede a desglosar los distintos requerimientos software necesarios para la correcta implantación del desarrollo del proyecto y los distintos procesos que involucra este mismo.

4.1.1 NI PACKAGES

Es una herramienta muy versátil para la gestión, distribución y actualización de software del fabricante National Instruments (NI). Centraliza de manera muy eficiente la gestión de software ya sea para la instalación, actualización y en el caso de que sea necesario la eliminación de este. Otra función la cual cabe destacar es la posibilidad de crear paquetes propios donde alberguen distintos softwares del fabricante con el fin de distribuir estos de manera conjunta a distintas fases de un proyecto, incluyendo la posibilidad de compartir estos con otros sistemas externos.



7. NI Package Manager Interface.

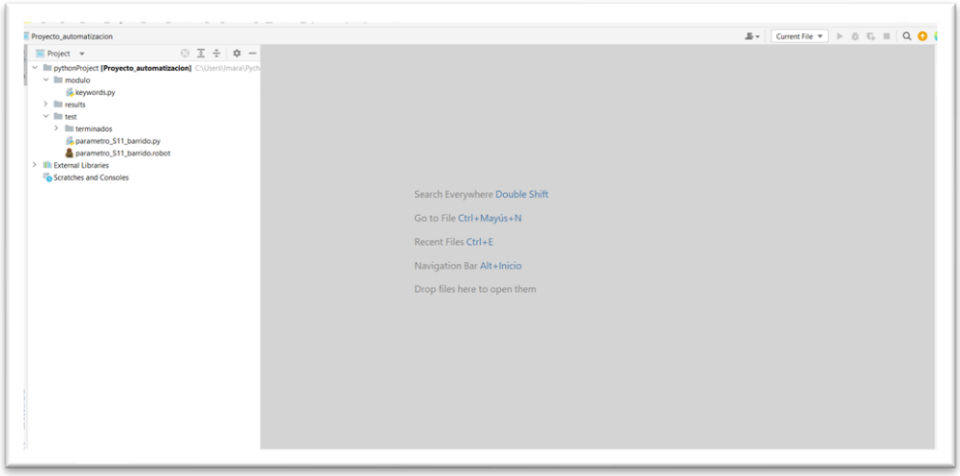
4.1.2 PYCHARM (IDE)

Este entorno de desarrollo integrado (IDE), es el seleccionado para el desarrollo de las distintas funciones de automatizado y posterior aplicación de Robot Framework. En él se procede a estructurar los distintos casos de prueba que componen la parte de experimentación y toma de medidas en el proyecto. La elección de este interprete de código se basa en la previa experiencia del autor con dicho entorno, la fácil y a su vez

intuitiva integración de distintos lenguajes tanto de programación como de automatizado en una única plataforma.

La versión utilizada para el caso que ocupa es la siguiente:

PyCharm Community Edition 2022.3.3	JetBrains s.r.o.	21/07/2023		223.8836.43



8. IDE Pycharm CommunityVersion. Entorno de desarrollo.

4.1.3 PYTHON (versión 3.7)

Es Python, la versión 3.7 la elegida para el completo desarrollo del código de automatizado. La razón por la cual se elige esta versión es por dar prueba de que ofrece los mínimos problemas de compatibilidad frente a otras plataformas requeridas en el desarrollo de este proyecto. Siendo algunas de ellas Robot Framework, por ejemplo.

9. Lenguaje de programación utilizado.

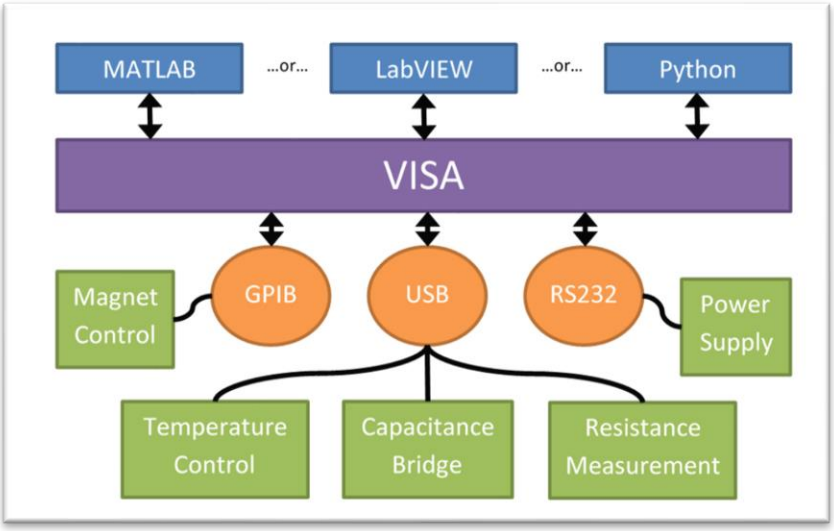
4.1.4 VISA

VISA, cuyas siglas corresponden a Arquitectura Software de Instrumentos Virtuales, representa un estándar de interfaz de programación diseñado para la interacción

entre software y hardware en el ámbito de instrumentos de medición y prueba. Destinado principalmente a proporcionar una interfaz unificada que permita comunicarse de manera consistente con distintos instrumentos, sin importar el fabricante o el tipo de interfaz física utilizada como GPIB, USB o Ethernet, entre otras.

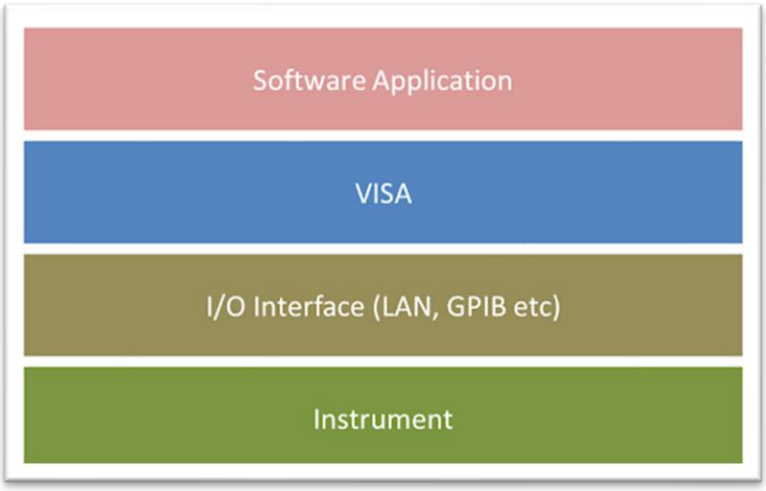
Los controladores de instrumentos virtuales se encargan de implementar la especificación de VISA y actúan como una capa de abstracción que se interpone entre la aplicación de software y el hardware del instrumento. Este enfoque simplifica el proceso de desarrollo de software, ya que se puede utilizar un conjunto estandarizado de comandos para controlar una variedad de instrumentos, mejorando así la interoperabilidad y la portabilidad de las aplicaciones. [7]

VISA establece un marco estándar que facilita la comunicación entre software y hardware de instrumentos, permitiendo la creación de controladores de instrumentos virtuales que simplifican la programación y el control de diversos dispositivos de medición y prueba.



10. Esquema de proceso y conexión virtual con VISA. [8]

4.1.5 NI-VISA



11. Proceso y componentes necesarios para la interconexión. [9]

NI-VISA, la implementación de National Instruments del estándar Virtual Instrument Software Architecture (VISA), surge como una herramienta necesaria en el ámbito de la instrumentación y el control. Diseñada para simplificar la comunicación entre software y hardware en instrumentos de medición y prueba, esta plataforma de software se presenta como un conjunto completo de bibliotecas y controladores que facilitan la interacción uniforme y eficiente con una variedad de instrumentos.

El núcleo de NI-VISA radica en proporcionar a los usuarios y técnicos de pruebas de laboratorio una interfaz de programación única y estandarizada. Esta interfaz simplifica el desarrollo de software al ofrecer un conjunto coherente de comandos VISA, independientemente de la marca o el tipo de interfaz física utilizada por los instrumentos, que puede incluir GPIB, USB, Ethernet o VXI. Al adoptar esta solución, los usuarios y técnicos pueden crear aplicaciones de control y adquisición de datos que son interoperables con una amplia gama de dispositivos, eliminando las complejidades asociadas con las particularidades de cada instrumento. [9]

En términos prácticos, NI-VISA facilita la integración de múltiples instrumentos en un sistema de prueba o medición, mejorando la eficiencia y la portabilidad del software. Su utilización en entornos de laboratorio y automatización industrial permite a usuarios y técnicos de pruebas de laboratorio desarrollar aplicaciones robustas y flexibles, optimizando las mediciones y pruebas en diversas configuraciones. La interoperabilidad

que proporciona esta implementación de VISA contribuye a una mayor agilidad en el desarrollo y mantenimiento de aplicaciones, fomentando así avances más rápidos y eficaces en el desarrollo tecnológico.

4.1.6 ROBOT FRAMEWORK

12. Aplicaciones de Robot Framework. [10]

Robot Framework es un framework de automatización de código abierto que se utiliza para facilitar la creación y ejecución de pruebas automáticas y tareas de automatización, incluida la RPA (Robotic Process Automation). Su flexibilidad y extensibilidad lo hacen adecuado tanto para pruebas de software como para la automatización de procesos de medidas de laboratorio, tal y como ocupa el asunto principal de este proyecto. Su estructura general, siendo omisibles algunas de estas es la siguiente. Keywords o palabras clave. Son comandos utilizados en las tareas de automatización, ya sea predefinidos, personalizados o desarrollados por el propio usuario. Test Cases o casos de prueba. Representan las acciones específicas y agrupan casos de prueba relacionados. Estos son la parte fundamental de la prueba o medida y se pueden desglosar en pasos. Automáticamente Robot Framework genera unos reportes donde de manera rápida e intuitiva se visualiza el resultado y los valores perseguidos en la medida. Siendo esta una potencialidad destacable de este marco de trabajo. Por último, Librerías. Estas proporcionan funcionalidades adicionales para extender las capacidades del marco de trabajo. El abanico es amplio en cuanto a librerías disponibles, aplicables a distintos ámbitos. También existe la posibilidad de desarrollar librerías propias e importar estas al test o task en desarrollo. [11]

Automatización de Tareas con Robot Framework:

En un proyecto centrado en tareas y automatización, Robot Framework se utiliza para orquestar y ejecutar estas tareas eficientemente. La sintaxis clara y estructurada facilita la definición de tareas, haciendo que la automatización sea accesible incluso para aquellos que carecen de conocimientos en programación.

Robot Framework ofrece una estructura común la cual hace relativamente fácil el desarrollo de la prueba. Aunque esta estructura es fija y la mayor parte de esta es inflexible ya que debe de existir como parte del test. Si que hay partes las cuales se pueden modificar o incluso excluir dependiendo del enfoque que se le desee dar al caso de prueba o tarea.

Las partes que componen la estructura estándar de Robot Framework son las siguientes:

- Settings
- Variables
- Tests cases
- Keywords

SETTINGS Es la parte encargada de enlazar y principalmente donde se desarrolla el proceso de importación de librerías externas. Ya que aporta flexibilidad, existe la posibilidad de desarrollar librerías propias las cuales ofrezcan funciones o definiciones especificas propias de pruebas que se deseen realizar. Algunos de los requisitos más importantes se importan desde este apartado como pueden ser las librerías, ficheros de recursos y ficheros de variables. Siendo esto último necesario si se desarrolla un fichero que contenga todas las variables necesarias para la ejecución o verificación del test o tarea. Por último, es el apartado donde se pueden definir los metadatos requeridos para los test suites y test cases. Siendo un test suite o conjunto de prueba, una colección de archivos y directorios que pueden ser versionados dependiendo del código de producción. [12]

VARIABLES Es el apartado de la estructura general del test donde se definen las variables utilizadas a lo largo de la ejecución de la prueba. Existe la posibilidad de crear variables globales las cuales serán recordadas o respetadas a lo largo de la totalidad del código o caso de prueba. Las variables pueden tener valores predeterminados los cuales son adjudicados en ficheros externos o en la misma línea de código respetando la

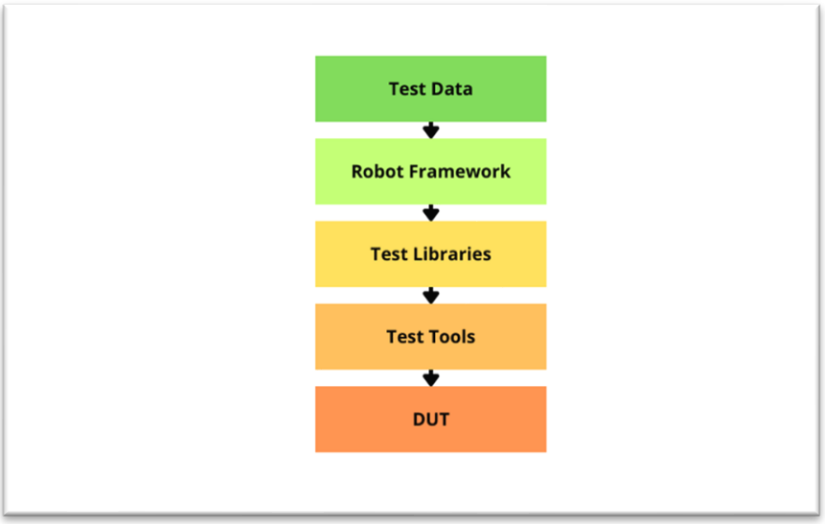
indentación requerida. En cuyo caso es cuatro espacios o equivalentemente dos tabulaciones.

TEST CASES En este apartado es donde mediante el uso de keywords se implementará de manera específica los casos de prueba concretos. Es aquí donde se desglosan los pasos distintos que componen el caso de prueba en su totalidad. Cabe destacar que el uso de keywords este sujeto a elección del usuario y aunque forma parte de la estructura general de un caso de prueba de Robot Framework, no es necesaria su implementación. En el caso que concierne este proyecto, se ha decidido eliminar su uso. Ya que la implementación de acciones específicas se lleva a cabo en un script Python externo al test case. Lo cual requiere de la importación de este en el apartado de "Settings". Para resumir de manera clara el proceso seguido, se ha implementado una serie de funciones que interactuarán con la instrumentación de laboratorio mediante lenguaje Python. Estas funciones están recogidas en un script .py que recibe el nombre de keywords. Simbólicamente haciendo referencia al apartado genérico que compone la estructura especifica de un caso de prueba con Robot.

KEYWORDS Es en este apartado donde se definen mediante las llamadas keywords las funciones o acciones a realizar. Existen librerías pre definidas en Robot Framework que contienen un amplio repertorio de funciones ya implementadas de manera que dependiendo del caso de prueba, existe la posibilidad de importar estas librerías. Dichas librerías pre definidas tienen ya funciones asociadas a "palabras clave" que desempeñan distintas tareas. En este caso, y como se ha mencionado con anterioridad. Se ha optado por partir desde cero, siendo parte del desarrollo de este proyecto la implementación total de funciones específicas para cada caso de prueba.

Resumiendo, estos son los 4 apartados genéricos que componen un fichero Robot Framework para un test case en concreto. A continuación, se detalla de manera visual la estructura general de Robot, recordando que admite cierta flexibilidad a la hora de su implementación. [12]

Robot Framework es un marco genérico, independiente de aplicaciones y tecnologías. Tiene una arquitectura altamente modular ilustrada en el siguiente diagrama.



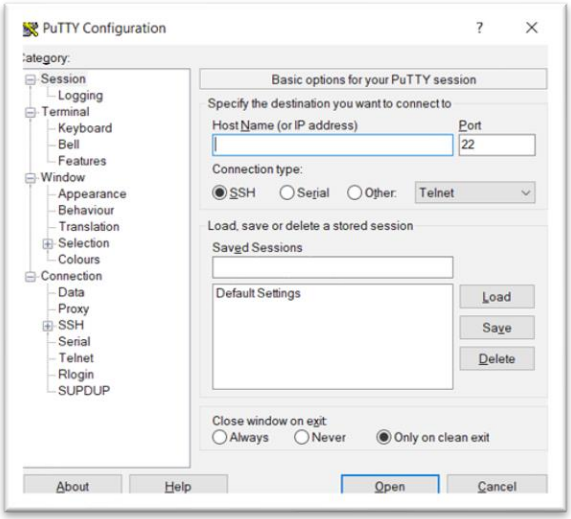
13. Arquitectura modular de Robot Framework

Para finalizar, cabe destacar que robot Framework se integra eficazmente en iniciativas de RPA, automatizando procesos de medidas de laboratorio mediante la simulación de la interacción humana en interfaces de usuario y sistemas de software. En este caso procesos de medidas en laboratorios relacionados con el ámbito de las telecomunicaciones.

4.1.7 PuTTY

PuTTY, desarrollado por Simon Tatham, es una aplicación de emulación de terminal de código abierto que facilita conexiones remotas con dispositivos como la Raspberry Pi a través de protocolos como SSH o Telnet. Esta herramienta, fácil de usar es adecuada para usuarios de todos los niveles. En el contexto de la Raspberry Pi, PuTTY se utiliza principalmente para la administración y control de la placa. Su terminal emulado permite a los usuarios interactuar con la Raspberry Pi como si estuvieran físicamente presentes, facilitando la ejecución de comandos, configuración y desarrollo de proyectos. Una de las principales ventajas de PuTTY es su capacidad para establecer conexiones seguras mediante SSH, proporcionando técnicas de cifrado de un extremo a otro extremo para proteger la transferencia de datos entre el dispositivo local y la Raspberry Pi. Esto es crucial para prevenir posibles brechas de datos sensibles, como contraseñas o claves de acceso, por terceros. Además de SSH, PuTTY también es útil para transferir archivos de manera segura mediante protocolos como SCP o SFTP, facilitando la copia de archivos

entre la Raspberry Pi y dispositivos locales, así como la realización de copias de seguridad de los datos de la placa.



14. Interfaz de terminal PuTTY.

4.1.8 PiFmRds

El proyecto PiFmRds fue creado por Oliver Mattos y Oskar Weigl como una extensión del proyecto PiFm, que permitía a la Raspberry Pi actuar como un transmisor de radio FM. La versión "Rds" incorpora la capacidad de transmitir información de datos, como datos de estación RDS, junto con la señal de audio. Este programa genera una modulación de frecuencia (FM) con datos RDS (Sistema de Datos de Radio) generados en tiempo real. Puede incluir audio monofónico o estereofónico. Está basado en el transmisor de FM creado por Oliver Mattos y Oskar Weigl, y posteriormente adaptado para utilizar DMA por Richard Hirst. Christophe Jacquet lo adaptó y añadió el generador y modulador de datos RDS. El transmisor utiliza el generador PWM (Modulación por Ancho de Pulso) de la Raspberry Pi para producir señales de VHF. [13]

4.1.9 WIRESHARK

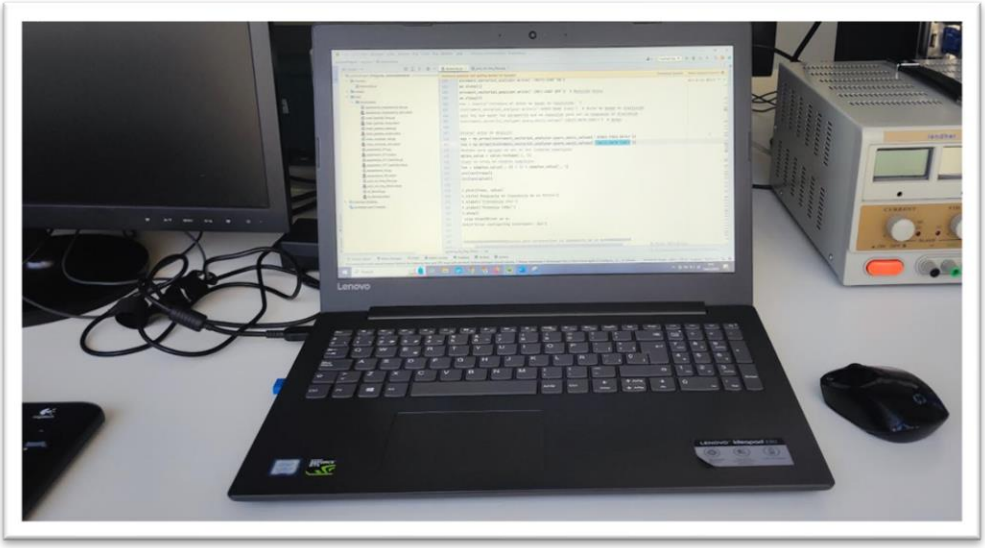
15. Logotipo de Wireshark. [14]

Wireshark es una herramienta de análisis de redes de código abierto que proporciona una interfaz gráfica intuitiva para capturar, visualizar y analizar el tráfico de red en tiempo real. Fue diseñado originalmente como un sniffer de paquetes, el cual ha evolucionado para convertirse en una herramienta multifuncional que ofrece una amplia gama de características y funcionalidades para el trabajo con entornos de redes y seguridad informática. Una de las características destacadas de Wireshark es su capacidad para capturar y analizar paquetes de datos en diversas interfaces de red, permitiendo a los usuarios examinar el tráfico en segmentos específicos de la red. Además, la herramienta es compatible con una amplia variedad de protocolos de red, desde los estándares comunes como TCP, UDP e IP, hasta protocolos más especializados y aplicaciones específicas. Permite al usuario examinar los paquetes capturados de manera detallada, con información desglosada sobre cada uno de ellos, incluyendo direcciones IP, puertos, tipos de protocolo y datos contenidos. La capacidad de filtrar y buscar dentro de grandes conjuntos de datos facilita el análisis y la identificación de problemas de red. [14]

4.2 RECURSOS HARDWARE

Es necesario cumplir con los siguientes requisitos hardware para la culminación del objetivo de este proyecto. Para ello, será necesario satisfacer las fases de componentes hardware.

- **Controlador** – Un pc que servirá de base de control. Donde se implementa la gran mayoría de los procesos de desarrollo de códigos, comandos y envió y recepción de información y resultados.



16. Ordenador portátil utilizado como controlador.

- **Interfaces de conexión** – Para el caso que ocupa este trabajo fin de grado, la conexión utilizada para la comunicación bidireccional entre instrumentación y controlador es cableado Ethernet.

17. Cable Ethernet CAT-6

- **Equipos externos** - El uso de un procesador Raspberry Pi para la transmisión de un fichero de audio con técnicas de modulación FM es requisito. En concreto el modelo Raspberri Pi 4 model B.

18. Raspberry Pi 4 modelo B.

- **Instrumentación** – los equipos, principales objetivos de este proceso son los modelos citados a continuación, provenientes del fabricante Keysight.

Modelo N5171B – Generador de funciones de alta frecuencia instalado en el laboratorio de radiofrecuencia y microondas de la Escuela Politécnica Superior de Linares.

19. Generador de funciones Keysight N5171B.

Este equipo consta de un rango de frecuencias comprendido entre los 9 KHz hasta los 3 GHz. Con capacidad de generar señales de distintos tipos entre los que se encuentran señales senoidales, triangulares, cuadradas y alguna otra forma. Brinda la posibilidad de modulación tanto en modalidad AM, FM y PM. El modo barrido también es una opción disponible para este instrumento, en el cual la versatilidad de los parámetros ajustables es amplia. Pudiéndose realizar barridos de frecuencia y de valores de amplitud de manera simultánea. Dispone de una salida RF en su frontal de tipo N hembra.

20. Salida tipo N del N5171B.

A continuación, se muestran las interfaces de conexión que dispone el equipo en su parte posterior. De todas estas, el conexionado principal y único que abarca este desarrollo está centrado exclusivamente en la entrada RJ-45 conexión Ethernet.

21. Interfaces de conexión del N5171B.

Para el correcto desarrollo del conjunto de scripts encargados de implementar la automatización de toma de medidas con este equipo citado, se recurre al manual de programador específico para este modelo. El cual recoge de manera detallada todos los comandos SCPI que admite el generador de funciones. Existe un manual para cada instrumento, los cuales vienen recogidos en la bibliografía que compone este documento. Estando todos disponibles en la página corporativa del fabricante Keysight. [15]

<u>Modelo N9320B –</u> Analizador de espectros de alta frecuencia instalado en el laboratorio de radiofrecuencia y microondas de la Escuela Politécnica Superior de Linares.

22. Analizador de espectros Keysight N9320B.

Este equipo tiene un rango de frecuencia de trabajo comprendido entre los 9 KHz y los 3 GHz. Con opciones de demodulación tanto para técnicas de modulación AM como FM. Existe la opción de barrido para de alguna manera poder visualizar aquellas características más concretas que cada caso de prueba requiera. Al igual que el generador de señales N5171B, posee una entrada RF tipo N hembra la cual tiene una impedancia de entrada de 50 Ω. Mediante las prestaciones que aporta este equipo se procederá a la demodulación de una transmisión FM desde un equipo externo e independiente a los instrumentos de laboratorio.

23. Entrada RF tipo N del N9320B.

El manual de programador del fabricante recoge de manera detallada todos los comandos SCPI admitidos por el instrumento. [16]

Modelo E5063A – Analizador de redes vectorial de alta frecuencia instalado en el laboratorio de radiofrecuencia y microondas de la Escuela Politécnica Superior de Linares.

24. Analizador de redes vectorial Keysight E5063A.

Este dispositivo está diseñado para trabajar en un rango de frecuencias comprendido entre los 100 KHz y los 4.5 GHz. Suficiente para analizar parámetros de microondas a ciertas longitudes de onda, milimétricas. Es con este equipo con el que se pretende visualizar y posteriormente analizar parámetros de impedancia de entrada, parámetros de Scattering y respuestas en frecuencia de dispositivos bajo pruebas (DUT).

Al igual que los equipos anteriormente citados, el manual de usuario especifico de este analizador vectorial recoge de manera concisa las secuencias de comandos admitidos por este equipo. [17]

4.3 COMPONENTES Y DUTS

Dentro del alcance global del proyecto, se encuentra un conjunto de componentes que abarcan los diversos dispositivos que se someterán a distintas pruebas a lo largo de su desarrollo. Estos dispositivos, conocidos formalmente como DUT (Device Under Test) o dispositivos bajo prueba, serán objeto de las evaluaciones planificadas durante la duración del proyecto. A continuación, se presenta de manera concisa la lista de DUTS.

4.3.1 ANTENA DIRECTIVA YAGI-UDA CON RADOMO

25. Antena Yagi-Uda con radomo protector.

Este tipo de antena es ampliamente utilizada tanto en transmisión como recepción debido a sus amplias prestaciones. Se considera una antena directiva la cual posee una composición similar si no, igual a la antena Yagi-Uda descrita en el apartado 4.3.3 de este capítulo.

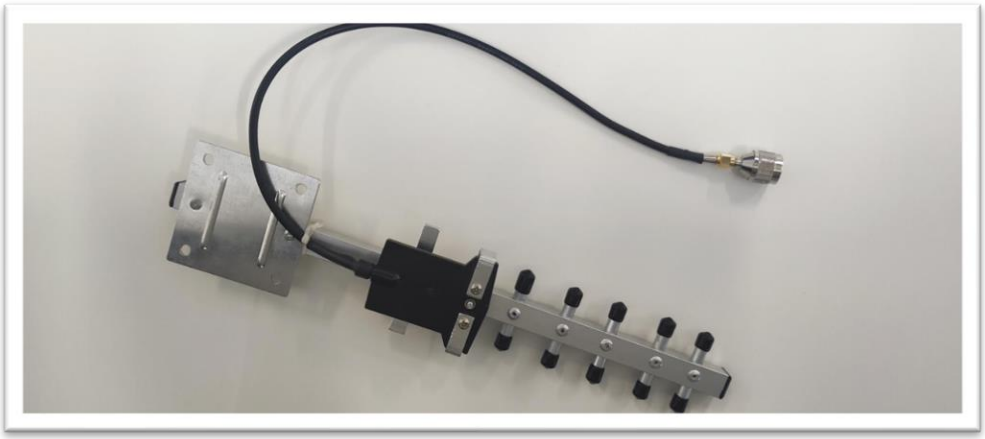
4.3.2 ANTENA BICONICA

Una antena bicónica es un tipo de antena con forma de cono invertido, que consta de dos brazos conductores desplegados en direcciones opuestas. Se utiliza en aplicaciones de radiofrecuencia y comunicaciones debido a su capacidad para captar y emitir eficientemente señales en un amplio rango de frecuencias. Su diseño simétrico permite una radiación omnidireccional, lo que facilita la recepción o transmisión uniforme de señales en todas las direcciones.

26. Antena bicónica montada en un trípode (soporte).

4.3.3 ANTENA YAGI-UDA

Una antena Yagi-Uda es un tipo de antena direccional utilizada en comunicaciones. Consiste en un conjunto de elementos conductores, incluyendo un elemento activo (dipolo) y varios elementos parasitarios (director y reflector) dispuestos linealmente. Esta configuración direccional permite un mayor alcance y eficiencia en la transmisión y recepción de señales en una dirección específica.



27. Antena tipo Yagi.

4.3.4 FILTROS (Lp, Hp, Bp)

El algoritmo que se desarrolla y posteriormente se implementa en el caso de prueba número 6, consta principalmente de la visualización grafica de la respuesta en frecuencia de un filtro. Para ello se dispone de los tres tipos de filtros mecánicos mencionados en el

título de este subapartado. Siendo posible la ejecución del caso de prueba para las tres variantes de filtros con tan solo ajustar las bandas de frecuencia de visualización.

28. Filtro paso bajas con frecuencia de corte a 900MHz.

29. Filtro paso banda entre 63-77 MHz.

30. Filtro paso altas a frecuencia 1000-3000 MHz.

4.3.5 KIT DE CALIBRACIÓN MANUAL

El kit de calibración es esencial para aumentar el grado de exactitud de las medidas tomadas con los equipos de laboratorio. En concreto, el analizador de redes vectorial E5063A. Este instrumento ha de ser calibrado antes de proceder con la toma de medidas y en concreto con antelación a la ejecución de los casos de prueba que componen el desarrollo del proyecto en su totalidad. Se presenta el kit de calibración disponible en el laboratorio el cual es en formato manual. Esto queriendo decir que la calibración del VNA ha de realizarse de manera presencial. Existe un kit de calibración denominado E-Calibration kit, el cual no es más que un dispositivo electrónico que emula las condiciones necesarias para la calibración del equipo.

31. Kit de calibración compatible con VNA E5063A.

32. Circuito abierto, corto circuito y carga de 50 Ohm.

El proceso de calibración es sencillo y dependiendo del grado de exactitud que se demande, existen varias alternativas para ello. El proceso de calibración básico consta de 3 pasos. Existe la necesidad de calibración en modo (Short) o corto circuito, modo (Open) o circuito abierto y finalmente modo de impedancia de valor 50Ω. Para un mayor detalle referirse al manual de usuario del equipo en contexto.

Una vez realizada la calibración, se puede proceder a la ejecución de los distintos casos de prueba que involucran el uso de VNA E5063A.

4.3.6 CABLEADO TIPO N Y COAXIAL TIPO BNC

El cableado es parte del material disponible en el laboratorio para las actividades docentes. Sirviendo este como conexionado entre instrumentación de laboratorio al igual que conexión entre instrumentos de medidas y dispositivos bajo prueba.

33. Cable coaxial con conectores tipo BNC.

34. Cableado con conectores tipo N (macho y hembra).

5 FUNDAMENTOS TEORICOS

Con el fin de obtener una comprensión más precisa del progreso, se procederá a simplificar y presentar de manera clara y concisa los diversos casos de prueba abordados en este documento. Para lograrlo, se requerirá hacer referencia a algunos conceptos teóricos vinculados a las materias enseñadas en el grado de Ingeniería de Tecnologías de Telecomunicación.

5.1 COMANDOS SCPI

En este apartado, se persigue detallar la sintaxis especifica de comandos SCPI de los distintos subsistemas que componen la jerarquía de comandos aceptados por los equipos bajo estudio. Para detallar de manera clara la arquitectura que sigue este estándar y su implementación en el desarrollo del proyecto. Previamente se ha señalado que el

estándar 488.2 ha servido como punto inicial para el desarrollo e implementación del concepto de comandos SCPI. Con el fin de clarificar y simplificar la comprensión, se proporciona una detallada descripción de las características clave de dicho estándar, así como la estructura y forma de los comandos comprensibles por los instrumentos pertinentes en el contexto de este proyecto.

Los comandos comunes (Common Commands) en el contexto de SCPI (Standard Commands for Programmable Instruments) son un conjunto estandarizado de instrucciones que facilitan la comunicación entre equipos de medición y control a través de cualquier medio de conexión compatible. Estos comandos están diseñados para ser universalmente reconocidos y utilizados en diversos instrumentos de medición, independientemente del fabricante. Técnicamente, los Common Commands en SCPI siguen la estructura definida por el estándar IEEE 488.2. Estos comandos suelen incluir instrucciones para configurar parámetros, realizar mediciones específicas, y controlar diversas funciones del instrumento. Además, están diseñados de manera que la sintaxis y el formato sean coherentes entre diferentes dispositivos, lo que simplifica la programación y la automatización de procesos en entornos de laboratorio y medición. A continuación, se detalla una tabla la cual contiene en su gran mayoría la totalidad de los comandos raíz o comandos comunes que derivan del estándar IEEE 488.2. [18]

Common Commands (SCPI)

Comando	Tipo	Descripción
***CLS**	**Command**	Comando de Borrado de Estado. Borra los registros de eventos en todos los grupos de registros. También borra la cola de errores.
***ESE**	**Command**	Habilita bits en el registro de habilitación para el grupo de registros de eventos estándar. Los bits seleccionados son luego informados en el bit 5 del Registro de Bytes de Estado. Un registro de habilitación define qué bits en el registro de eventos se informarán al grupo de registros de bytes de estado. Puede escribir o leer desde un registro de habilitación.
***ESR?**	**Query**	Consulta del Registro de Estado de Eventos Estándar. Realiza una consulta al registro de eventos para el grupo de registros de eventos estándar.
***IDN?**	**Query**	Consulta de Identificación. Devuelve la cadena de identificación del instrumento.
***OPC/*OPC?**	**Command/Query**	La diferencia entre *OPC y *OPC? es que *OPC? emite "1" cuando la operación actual se completa. *OPC espera a que la operación termine.
***OPT?**	**Query**	Devuelve un string que identifica las posibles opciones instaladas,
***RST**	**Command**	Restablece el instrumento al estado predeterminado de fábrica, esto es similar a realizar un PRESET. La diferencia radica en que *RST restablece el instrumento para la operación SCPI, mientras que PRESET lo hace para la operación desde el panel frontal.
***SRE**	**Command**	Habilita bits en el registro de habilitación para el grupo de registros de bytes de estado.
***STB?**	**Query**	Realiza una consulta al registro de condiciones para el grupo de registros de bytes de estado y devuelve un valor decimal igual a la suma ponderada por binario de todos los bits establecidos en el registro.
***TRG**	**Command**	Activa el proceso de medida del instrumento si se selecciona TRIGger:SOURce BUS.
***WAI**	**Command**	Configura el búfer de salida del instrumento para esperar a que se completen todas las operaciones pendientes antes de ejecutar comandos adicionales a través de la interfaz.
***TST?**	**Query**	Realiza una autoevaluación básica del instrumento y devuelve una indicación de aprobado o fallado.
***CAL?**	**Query**	Este comando de consulta realiza una calibración completa del instrumento y luego devuelve un <valor_numérico> para indicar el resultado de la calibración. Siendo 0 aprobado y 1 fallo.

Tabla 1. Comandos comunes SCPI. IEEE 488.2

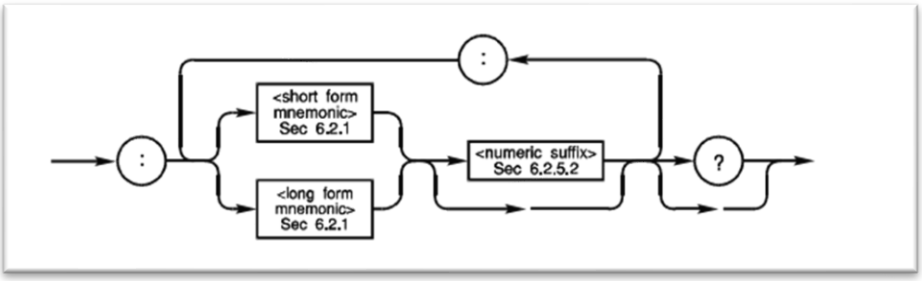
5.1.1 COMANDOS ESPECIFICOS PARA EQUIPOS KEYSIGHT

SINTÁXIS COMANDOS SCPI, CARACTERES ESPECIALES.

Carácter (ejemplos)	Descripción
Mayúsculas **CALC**ulate	Las letras mayúsculas son las letras mínimamente requeridas del encabezado del programa de comando o consulta. Las letras minúsculas representan la forma completa (ortografía completa), que puedes omitir si lo deseas.
< value >	Los corchetes angulares se utilizan para señalar que el contenido dentro de ellos tiene un significado específico o simboliza algo distinto a las propias palabras. Siendo "value" el valor del parámetro.
[: SOURce]	Los corchetes cuadrados indican que la información contenida en ellos es opcional y no es indispensable para llevar a cabo la secuencia de comandos.
|	La barra vertical se puede leer como "o" y se utiliza para separar opciones alternativas de parámetros. Siendo solo una necesaria.
{ }	Las llaves se emplean para delimitar elementos que pueden repetirse, incluso sin aparecer, dentro de una estructura.
: **:SENSe :FREQuency :STARt**	Los dos puntos son necesarios para acceder a los distintos subsistemas que componen la estructuración del equipo.

Tabla 2. Tabla de caracteres especiales de la sintaxis SCPI.

De manera visual se procede a graficar como es la sintaxis para los comandos SCPI. Teniendo en cuenta que existe una jerarquía que afecta a los subsistemas que puedan existir y que en efecto podrán ser de distinto orden dependiendo del equipo que se esté utilizando.



35. Sintaxis para comandos SCPI subsistema. [5]

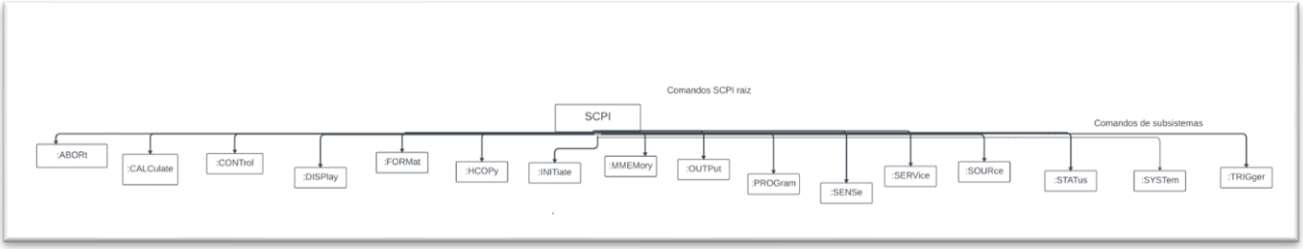
Los comandos subsistema, los cuales a continuación se esclarecerán, son precedidos por dos puntos siempre. Es la directriz establecida en el estándar IEEE 488.1 de 1999. Seguidos del comando mnemotécnico asociado al subsistema al que se accede. Un ejemplo es:

[:SENSe]:FREQuency:CENTer <value>

- [:SENSe] – Consultando la tabla 2, se puede simplificar el comando anterior omitiendo esta parte.
- :FREQuency – Como se indica en la tabla 2, esta sección del conjunto de comandos se puede sustituir por :FREQ , omitiendo las minúsculas que la componen. Es importante tener en cuenta que se puede optar por la sintaxis que combina mayúsculas y minúsculas en un mismo comando, pero no es compatible y por tanto dará un error el combinar los dos formatos de sintaxis dentro de una misma línea de comandos.
- :CENTer también puede ser utilizado como :CENT, siendo consecuente con la omisión de las letras minúsculas.
- <value> - En esta posición y precedido siempre de un espacio se espera un valor, en este caso numérico del parámetro al que se hace referencia con el comando en su totalidad. Siendo ":SENSe" el subsistema dedicado al ajuste de medidas y adquisición de datos. En el ejemplo se hace referencia al valor de frecuencia central. La posibilidad numérica esta acotada al valor total de la frecuencia en Hz o un valor numérico en formato e. Ejemplo → 10e9 (10 GHz).

5.1.2 ARBOLES JERARQUICOS DE SUBSISTEMAS SCPI

Para facilitar la comprensión de la arquitectura jerárquica de los distintos subsistemas que componen el sistema en su totalidad de un instrumento de medidas, se procede a detallar de manera grafica el árbol de comandos SCPI específico para el equipo de instrumentación de medidas Keysight E5063A (Analizador de redes Vectorial).



36. Jerarquía de subsistemas en VNA E5063A.

Para consultar la jerarquía detallada de los subsistemas SCPI y comandos aceptados por el VNA E5063A, será necesario referirse al manual de programador del equipo. Se omite el desglose jerárquico de los dos instrumentos restantes ya que su estructura y contenido es similar al del instrumento citado anteriormente.

5.2 CONCEPTUALIZACIÓN CON ROBOT FRAMEWORK

Para culminar con la conceptualización de las herramientas utilizadas durante el proceso de desarrollo, se pasa a exponer detalladamente Robot Framework. Su sintaxis, estructuras genéricas y otros aspectos. Se ha de tener en cuenta que en el contexto que ocupa este proyecto, el enfoque central es (RPA) el cual se define como la automatización de procesos robóticos. Este hace referencia a tareas específicas automatizadas como son la toma de medidas con instrumentación de laboratorio.

5.2.1 SINTAXIS GENERICA DE LOS TESTS

Las secciones que componen la estructura genérica de un test o task es la que ya se ha mencionado en el apartado de requerimientos software. Como ya se ha citado, está compuesta por “Settings”, “Variables”, “Test Cases” o “Tasks” en el contexto del proyecto. Y finalmente “Keywords”. Existe un apartado adicional el cual es denominado “Comments”, cabe destacar que este apartado es ignorado por Robot Framework ya que su contenido es meramente orientativo. Todos estos apartados tienen una sintaxis concreta la cual está compuesta por tres asteriscos iniciales y tres finales que envuelven la palabra clave. Esta técnica tiene un grado de flexibilidad ya que admite variantes del formato de apartado, como se ha comentado anteriormente.

*** Settings ***	*** Variables ***
*** Test Cases ***	*** keywords ***
*** Comments ***	

Tabla 3. Sintaxis y apartados.

5.2.2 FORMATO DE ESPACIADO

Robot Framework analiza datos de la siguiente manera. Primero divide los datos en líneas y luego las líneas en tokens como palabras clave y argumentos. Al utilizar el formato separado por espacios, el separador entre tokens es de dos o más espacios, o alternativamente, una o más tabulaciones.

```
*** Settings ***
Documentation     Example using the space separated format.
Library           OperatingSystem

*** Variables ***
${MESSAGE}        Hello, world!

*** Test Cases ***
My Test
    [Documentation]    Example test.
    Log    ${MESSAGE}
    My Keyword    ${CURDIR}

Another Test
    Should Be Equal    ${MESSAGE}    Hello, world!

*** Keywords ***
My Keyword
    [Arguments]    ${path}
    Directory Should Exist    ${path}
```

37. Ejemplo de espaciado (extraído de Robot Framework user guide) [12]

5.2.3 CABECERA SETTINGS

En la sección *** Settings *** de un caso de prueba en Robot Framework, se pueden establecer configuraciones clave para adaptar el comportamiento del caso de prueba. En primer lugar, se puede asignar un nombre distintivo al caso de prueba, facilitando su identificación única. La inclusión de etiquetas (tags) posibilita la categorización y agrupación eficiente de casos de prueba relacionados. La sección de documentación permite proporcionar detalles explicativos sobre el propósito y la ejecución del caso de prueba. La opción de plantillas (templates) permite la reutilización de configuraciones y pasos comunes para mantener consistencia en la estructura de los casos de prueba. Finalmente, la configuración de bibliotecas (Library) posibilita la integración de funciones y keywords externos, expandiendo las capacidades del caso de prueba. Estas opciones ofrecen un nivel significativo de flexibilidad y personalización en la configuración y ejecución de casos de prueba en el entorno del Robot Framework. [19]

5.2.4 CABECERA VARIABLES

En la sección *** Variables *** de un caso de prueba en Robot Framework, se pueden definir y asignar variables que contienen datos o valores específicos. Este apartado proporciona un espacio estructurado para la declaración de variables que serán utilizadas a lo largo del caso de prueba. Las variables pueden representar información dinámica, variables globales que conservarán su valor a lo largo de todo el caso de prueba o parámetros configurables. Esta sección permite una gestión eficiente de datos al proporcionar un lugar centralizado para su definición, facilitando así la comprensión del caso de prueba. Al utilizar variables, se mejora la flexibilidad y la reutilización del código, ya que los valores pueden ser modificados de manera centralizada sin necesidad de editar múltiples instancias a lo largo del caso de prueba. [19]

5.2.5 CABECERA TEST CASES

En la sección *** Test Cases *** de un archivo en Robot Framework se encuentra el núcleo de la automatización, donde se definen y detallan los casos de prueba específicos. Cada caso de prueba se estructura mediante una serie de pasos que involucran keywords y acciones que se deben llevar a cabo durante la ejecución. En esta sección, se describen los flujos de trabajo esperados, las validaciones y cualquier interacción con el sistema bajo prueba. Los casos de prueba se presentan de manera clara y legible, facilitando la comprensión del propósito y la lógica subyacente de cada escenario. Además, se pueden incorporar variables y llamadas a bibliotecas específicas para ampliar la funcionalidad y reutilización de código. No hay que olvidar que en el enfoque de este proyecto la terminología test case se sustituye o presupone una tarea o proceso robótico de automatización, task. [19]

5.2.6 CABEZERA KEYWORDS

En la sección *** Keywords *** de un archivo en Robot Framework, se definen funciones y bloques de código reutilizables que encapsulan lógica específica. Estas keywords representan acciones o pasos individuales que pueden ser invocados y utilizados en varios casos de prueba. La sección proporciona un espacio organizado para la creación y mantenimiento de funciones personalizadas, permitiendo una modularidad efectiva en la

automatización. Cada keyword encapsula una tarea específica, desde operaciones simples hasta flujos de trabajo más complejos. [19]

En este caso se omite el uso del apartado *** Keywords *** ya que se ha definido un fichero externo que recoge de manera ordenada las funciones específicas desarrolladas para la interacción con los equipos de instrumentación. El uso de estas funciones se implementa mediante la importación del fichero, siendo esta acción equivalente a la de importar una librería predeterminada.

```
1   *** Settings ***
2   Library  ../../modulo/keywords.py
3
4   *** Variables ***
5   # se pueden meter los valores directamente sustituyendo en el valor de la variable, para realizar la medida.
6   ${spectrum_address}
7   ${instrument_spectrum}
8   ${cnt_freq}     107.9e6     #valor de frecuencia central para la demodulación fm
9   ${span}     0               #valor de span.  a span distinto a 0, no se ejecuta la demodulación
10
11  *** Test Cases ***
12
13  Task 1 Find
14      [Documentation]    Finding the instrument on the network
15      [Tags]  Step 00     finding the correct instrument to iniciate procedure
16      Log     ${spectrum_address}
17      ${spectrum_address}=   find_n9320b
18      Set Global Variable    ${spectrum_address}
19      Log     ${spectrum_address}
20
21  Task 2 Connect
22      [Documentation]    Connecting to spectrum analyzer via IP address
23      [Tags]  Step 01     Opening connection with the instrument
24      Log  ${spectrum_address}
25      ${instrument_spectrum}=  connect_to_spectrum_analyzer       ${spectrum_address}
26      Set Global Variable    ${instrument_spectrum}
27      Log  ${spectrum_address}
28      Log  ${instrument_spectrum}
29
30  Task 3 Reset instrument
```

38. Ejemplo de sintaxis y estructura de un caso de prueba.

5.2.7 DESGLOSE REPRESENTATIVO DE SINTAXIS

Una variable creada para ajustar el valor de frecuencia central en una operación con el instrumento N9320B, analizador de espectros. Para la creación de una nueva variable se compone de la siguiente manera. El valor de la variable se define tras la separación de espacios, en este caso dos tabulaciones.

```
${cnt_freq}      107.9e6
```

39. Ejemplo de una variable.

A continuación, un ejemplo de importación de la librería keywords, la cual se localiza dentro del directorio modulo como se puede comprobar en la ilustración 38.

```
Library   ../../modulo/keywords.py
```

40. Importando una librería externa.

Finalmente se procede al desarrollo del test case con sus distintas secciones internas. En él se define el nombre de la prueba o tarea, información adicional y variables, valores de ajuste y keywords que desempeñarán distintas tareas.

```
*** Test Cases ***
Task 1 Find
    [Documentation]    Finding the instrument on the network
    [Tags]  Step 00     finding the correct instrument to iniciate procedure
    Log     ${spectrum_address}
    ${spectrum_address}=   find_n9320b
    Set Global Variable    ${spectrum_address}
    Log     ${spectrum_address}
```

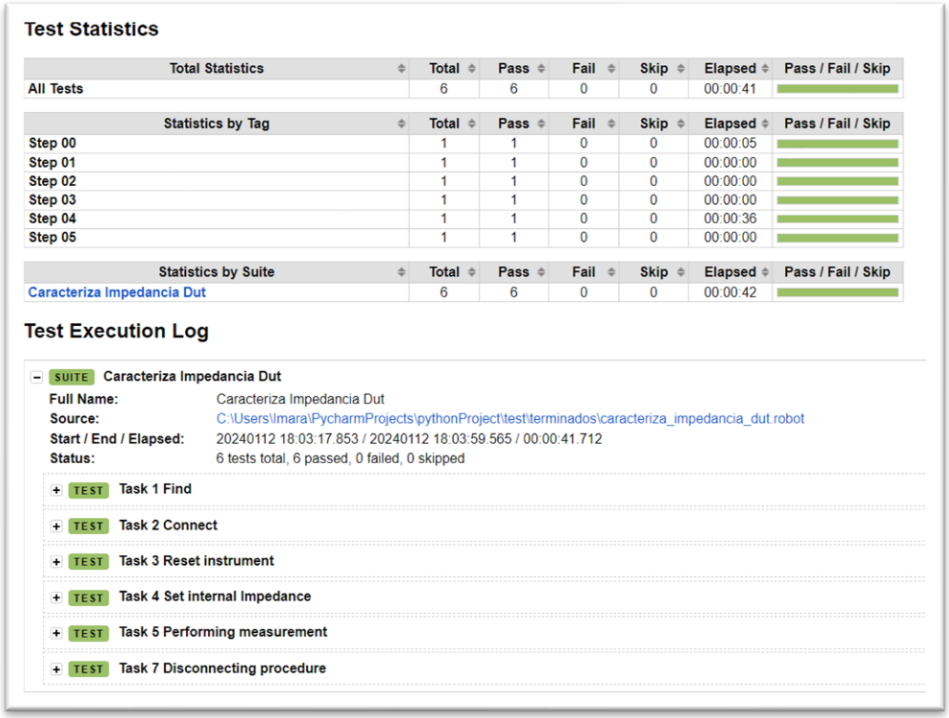
41. Tarea 1, donde se visualizan las distintas secciones.

- Se puede ver el nombre de la tarea – “Task 1 Find”
- [Documentatión] – Descripción de la acción que acomete la tarea.
- [Tags] - En Robot Framework, los [Tags] o etiquetas son una característica que se utiliza para categorizar y organizar los casos de prueba. Aquí simplemente se le asigna un numero de paso por tener una organización.
- Log – es una manera de obtener una comprobación visual de un dato cualquiera en el reporte que genera Robot Framework. En este caso se utiliza como un control de errores para determinar en qué parte de la ejecución puede surgir una incongruencia.

- Set Global Variable – es un keyword contenido por Robot Framework el cual ajusta el estado de una variable a global. Siendo esta asignada un valor estático el cual será invariante a lo largo de todo el caso de prueba. En este caso se puede observar que a la variable ${spectrum_address} le asigna la denominación de global. Por esclarecer, esta variable guarda la dirección IP de un equipo en concreto.

5.2.8 GENERACIÓN DE REPORTES E INFORMES

Tras ejecutar un test en Robot Framework, se generan varios archivos que proporcionan información detallada sobre la ejecución y los resultados de las pruebas. Estos son un Log File, Report File, Output File. Cada uno de estos ficheros contiene información muy útil y detallada sobre los procesos, tiempos y sucesos de ejecución. Log File o archivo de registro detalla información sobre la ejecución, incluyendo mensajes de inicio y finalización, así como eventos y resultados de cada paso del caso de prueba. Este fichero se genera con extensión .log.



Test Statistics

Total Statistics	Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip
All Tests	6	6	0	0	00:00:41	

Statistics by Tag	Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip
Step 00	1	1	0	0	00:00:05	
Step 01	1	1	0	0	00:00:00	
Step 02	1	1	0	0	00:00:00	
Step 03	1	1	0	0	00:00:00	
Step 04	1	1	0	0	00:00:36	
Step 05	1	1	0	0	00:00:00	

Statistics by Suite	Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip
Caracteriza Impedancia Dut	6	6	0	0	00:00:42	

Test Execution Log

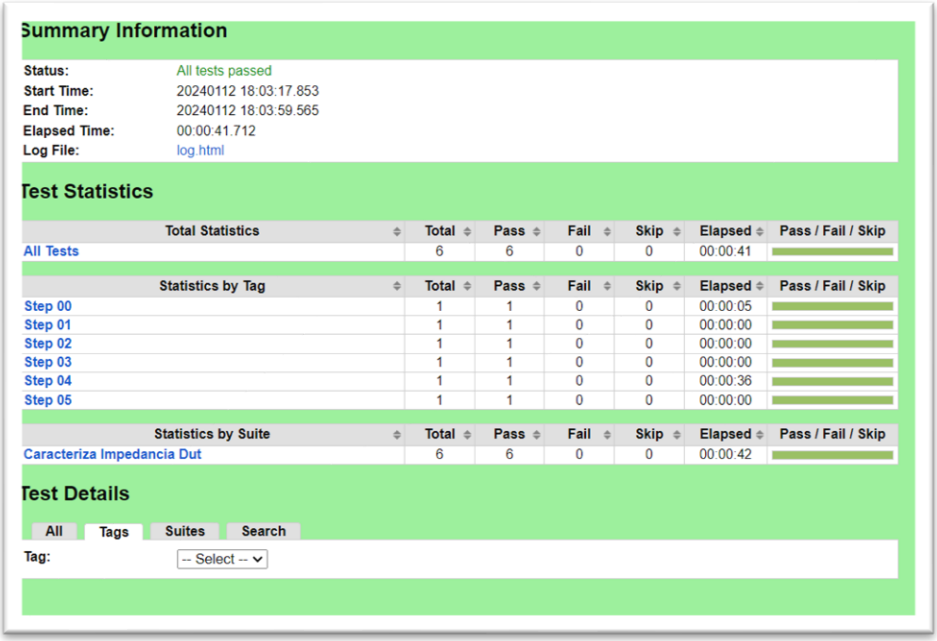
SUITE Caracteriza Impedancia Dut

Full Name: Caracteriza Impedancia Dut
Source: C:\Users\lmara\PycharmProjects\pythonProject\test\terminados\caracteriza_impedancia_dut.robot
Start / End / Elapsed: 20240112 18:03:17.853 / 20240112 18:03:59.565 / 00:00:41.712
Status: 6 tests total, 6 passed, 0 failed, 0 skipped

TEST Task 1 Find
TEST Task 2 Connect
TEST Task 3 Reset instrument
TEST Task 4 Set internal Impedance
TEST Task 5 Performing measurement
TEST Task 7 Disconnecting procedure

42. Ejemplo de Log file.

Report File o archivo de informe proporciona una visión general de los resultados de la ejecución. Incluye estadísticas, resúmenes de casos de prueba pasados y fallidos, y enlaces a detalles sobre cada caso de prueba. Este fichero se genera con extensión .html.



Summary Information

Status:	All tests passed
Start Time:	20240112 18:03:17.853
End Time:	20240112 18:03:59.565
Elapsed Time:	00:00:41.712
Log File:	log.html

Test Statistics

Total Statistics	Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip
All Tests	6	6	0	0	00:00:41	

Statistics by Tag	Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip
Step 00	1	1	0	0	00:00:05	
Step 01	1	1	0	0	00:00:00	
Step 02	1	1	0	0	00:00:00	
Step 03	1	1	0	0	00:00:00	
Step 04	1	1	0	0	00:00:36	
Step 05	1	1	0	0	00:00:00	

Statistics by Suite	Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip
Caracteriza Impedancia Dut	6	6	0	0	00:00:42	

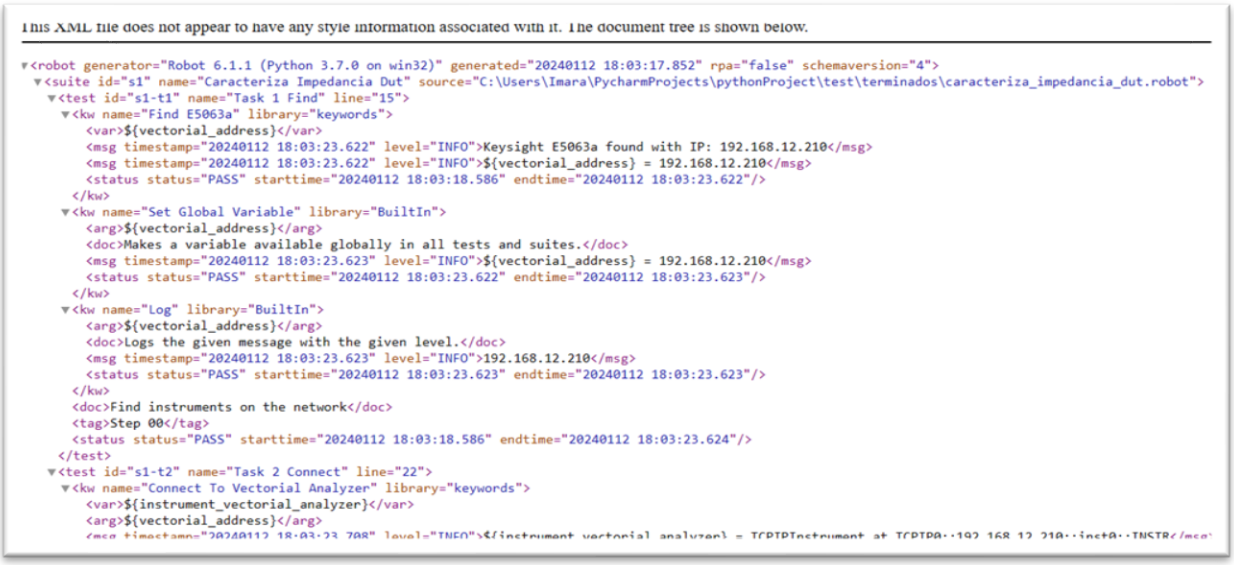
Test Details

All | Tags | Suites | Search

Tag: -- Select --

43. Ejemplo de Report file.

El Output File o archivo de salida, en formato XML, contiene información estructurada sobre la ejecución de los casos de prueba, incluyendo detalles de resultados, tiempos de ejecución y otros datos relevantes. Este fichero se genera con extensión .xml si se predefine.



This XML file does not appear to have any style information associated with it. The document tree is shown below.


```
▼<robot generator="Robot 6.1.1 (Python 3.7.0 on win32)" generated="20240112 18:03:17.852" rpa="false" schemaversion="4">
 ▼<suite id="s1" name="Caracteriza Impedancia Dut" source="C:\Users\Imara\PycharmProjects\pythonProject\test\terminados\caracteriza_impedancia_dut.robot">
  ▼<test id="s1-t1" name="Task 1 Find" line="15">
   ▼<kw name="Find E5063a" library="keywords">
     <var>${vectorial_address}</var>
     <msg timestamp="20240112 18:03:23.622" level="INFO">Keysight E5063a found with IP: 192.168.12.210</msg>
     <msg timestamp="20240112 18:03:23.622" level="INFO">${vectorial_address} = 192.168.12.210</msg>
     <status status="PASS" starttime="20240112 18:03:18.586" endtime="20240112 18:03:23.622"/>
    </kw>
   ▼<kw name="Set Global Variable" library="BuiltIn">
     <arg>${vectorial_address}</arg>
     <doc>Makes a variable available globally in all tests and suites.</doc>
     <msg timestamp="20240112 18:03:23.623" level="INFO">${vectorial_address} = 192.168.12.210</msg>
     <status status="PASS" starttime="20240112 18:03:23.622" endtime="20240112 18:03:23.623"/>
    </kw>
   ▼<kw name="Log" library="BuiltIn">
     <arg>${vectorial_address}</arg>
     <doc>Logs the given message with the given level.</doc>
     <msg timestamp="20240112 18:03:23.623" level="INFO">192.168.12.210</msg>
     <status status="PASS" starttime="20240112 18:03:23.623" endtime="20240112 18:03:23.623"/>
    </kw>
    <doc>Find instruments on the network</doc>
    <tag>Step 00</tag>
    <status status="PASS" starttime="20240112 18:03:18.586" endtime="20240112 18:03:23.624"/>
   </test>
  ▼<test id="s1-t2" name="Task 2 Connect" line="22">
   ▼<kw name="Connect To Vectorial Analyzer" library="keywords">
     <var>${instrument_vectorial_analyzer}</var>
     <arg>${vectorial_address}</arg>
```


44. Ejemplo de Output file.

Existen muchas más prestaciones que Robot Framework aporta a la automatización. Para ello es aconsejable consultar la web de guía de usuario de Robot donde se da acceso a mayor detalle de todo lo referido a este marco de trabajo de automatización.

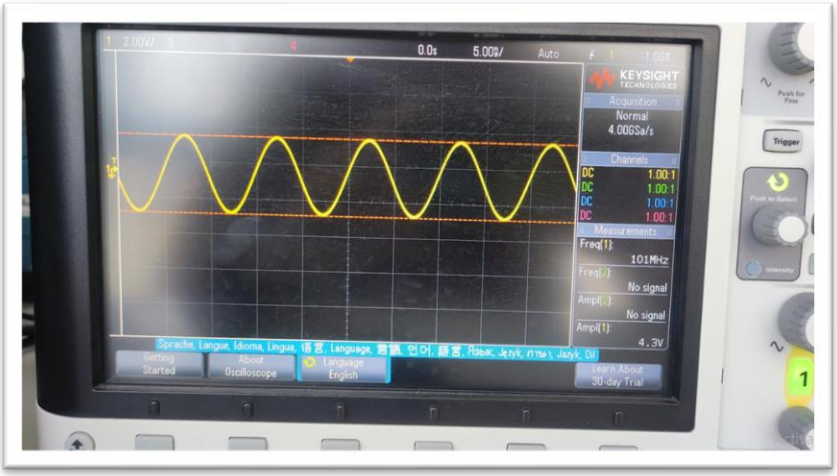
5.3 CONCEPTOS TEORICOS DE LOS CASOS DE PRUEBA

Es en este apartado del capítulo 5 donde se pretende clarificar y detallar brevemente algunos conceptos teóricos clave para el correcto entendimiento de los distintos casos de prueba que componen la totalidad de este documento.

5.3.1 CASO 1. Genera una señal

Para la correcta generación de una señal senoidal, es requisito básico que los datos a introducir en el generador de funciones N5171B sean mínimamente el valor de amplitud y la frecuencia de la señal. Como valor de amplitud es comprensible que será parte de la elección del ingeniero que valor sea el adecuado, dependiendo de la aplicación deseada. Variantes de valores de amplitud existen como opciones a elegir, entre las cuales figuran, valores de pico (V_p), valores de pico-pico (V_{pp}) y valor eficaz (V_{eff}). El concepto de frecuencia se define como el número de vueltas o periodos de la señal por espacio temporal. Lo cual se mide en Hercios y registra o denota el número de vueltas completas que da la señal por segundo de tiempo. Con estos dos valores se puede generar una onda o señal de tipo senoidal desde cualquier generador de funciones. Además de estos dos parámetros clave, existen algunos más que pueden ser ajustados a elección del usuario como pueden ser el tipo de señal. Con esto se refiere a la forma concreta con la que se graficara el progreso de la onda en el tiempo. Existen opciones varias como pueden ser de tipo señal cuadrada, donde el valor de pico de la señal es un valor constante durante un determinado instante de tiempo. De tipo triangular, el cual registra una variación brusca y a su vez periódica del nivel de amplitud. A continuación, se muestra de manera visual algunos tipos de señales, téngase en cuenta que no se grafican todos los tipos en su totalidad.

45. Señal triangular.



46. Señal senoidal.

47. Señal cuadrada.

5.3.2 CASO 2. Genera un barrido en frecuencia

En el contexto del presente desarrollo, se aborda la parametrización del instrumento N5171B, un generador de funciones de la firma Keysight. Desde un enfoque teórico, se ofrece una exposición sobre la técnica de barrido en frecuencia con el propósito de mejorar la comprensión de las aplicaciones y análisis derivados de este procedimiento.

El barrido en frecuencia se configura como una metodología que permite delimitar y analizar el comportamiento en términos de frecuencia de señales, dispositivos bajo prueba (DUT) y otros elementos de instrumentación. En el caso de una señal senoidal diseñada con parámetros específicos, al someterla a un barrido en frecuencia, se logra caracterizar su comportamiento en el dominio de la frecuencia al establecer límites iniciales y finales. Además, se ajusta el número de puntos de visualización, representativo del muestreo que el instrumento llevará a cabo sobre la información generada durante el barrido.

Dentro de los aspectos teóricos asociados a la técnica de barrido en frecuencia, se destaca la posibilidad de delimitar el barrido mediante puntos de muestreo y la opción de acotar temporalmente el proceso. Es viable establecer una duración temporal máxima para el barrido, generando barridos en frecuencia en intervalos de tiempo definidos entre los límites de frecuencia preestablecidos. Asimismo, se presenta la opción de restringir el barrido a una única iteración o ejecutarlo en modo continuo, ampliando las posibilidades de aplicación y análisis.

En el contexto de aplicaciones prácticas, se exploran escenarios como el análisis del comportamiento en frecuencia de un DUT, utilizando, por ejemplo, la respuesta en frecuencia de un filtro paso altas como caso de estudio. Para visualizar y comprender de manera más detallada el comportamiento, se requiere acotar el barrido entre frecuencias inicial y final, junto con la determinación del número de puntos para generar la respuesta en frecuencia. Este enfoque permitirá graficar y analizar con mayor profundidad la respuesta en frecuencia del filtro paso banda, ilustrando la utilidad y versatilidad de la técnica de barrido en frecuencia en aplicaciones prácticas y analíticas.

48. Ajuste de parámetros en el script de Robot Framework.

5.3.3 CASO 3. Genera una modulación AM

La modulación AM, o también referida como modulación de amplitud, es una técnica de transmisión de información en la que la amplitud de una señal portadora se modifica de acuerdo con la información que se desea transmitir. Esta información puede ser una señal analógica, como una señal de audio o de video, o una señal digital, como un mensaje de texto o un archivo de datos.

En la modulación AM, la señal portadora es una onda sinusoidal de una frecuencia determinada. La información que se desea transmitir se representa como una señal de baja frecuencia, llamada señal moduladora. La señal moduladora se multiplica por la señal portadora, lo que produce una señal modulada AM. Existe el concepto de índice de modulación el cual se refiere a una medida de cómo varía la onda modulada en relación con la señal portadora. En términos sencillos, el índice de modulación indica cuánto cambia la amplitud de la onda modulada en comparación con la amplitud constante de la onda portadora. Un índice de modulación más alto implica una mayor variación en la amplitud, lo que se traduce en una modulación más intensa.

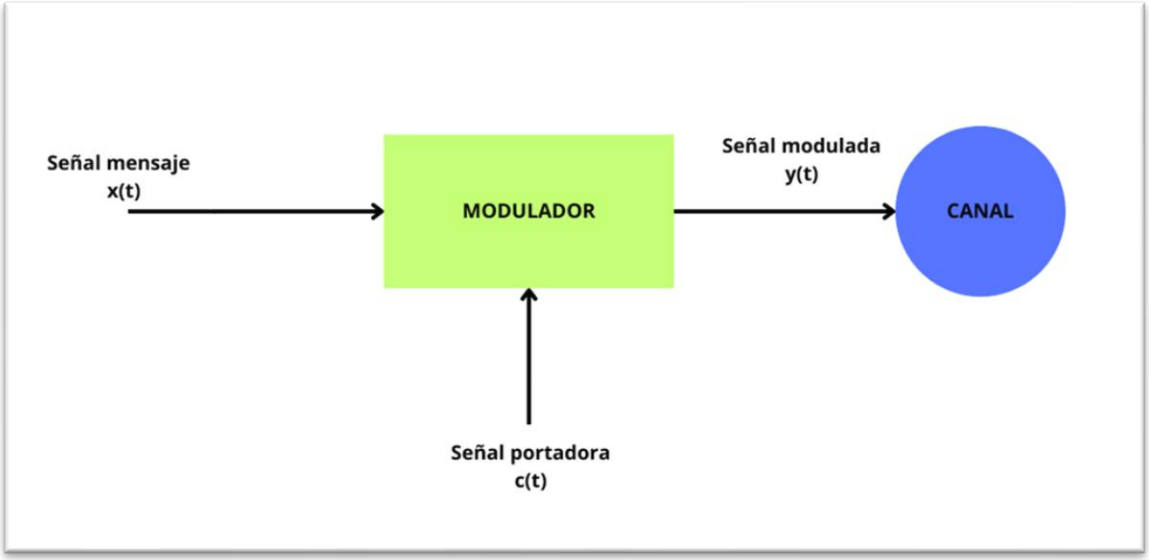
La señal modulada AM tiene dos componentes principales:

- La componente continua, que es la señal portadora sin modulación.
- La componente moduladora, que es la variación de la amplitud de la señal portadora.

La componente moduladora contiene la información que se desea transmitir. La amplitud de la componente moduladora varía de acuerdo con la amplitud de la señal moduladora. La señal modulada AM se transmite a través de un medio de transmisión, como el aire o un cable. En el extremo receptor, la señal modulada AM se demodula para recuperar la información que se desea transmitir. La modulación AM es una técnica sencilla y eficiente que se utiliza en una amplia variedad de aplicaciones, incluidas las siguientes:

- Radiodifusión: la radiodifusión AM es la forma más común de transmisión de audio.
- Radioafición: la radioafición AM se utiliza para la comunicación entre radioaficionados.
- Transmisión de datos: la modulación AM se utiliza para transmitir datos a través de ondas de radio.

Ventajas destacables de la modulación AM entre otras son las siguientes. Es una técnica sencilla y eficiente. Es relativamente económica, es compatible con una amplia variedad de dispositivos. Algunas desventajas de la modulación AM. Puede ser susceptible a la interferencia y no es tan eficiente como otras técnicas de modulación, como la modulación de frecuencia (FM). Actualmente su uso es cada vez menos frecuente.



49. Esquema modulación AM.

5.3.4 CASO 4. Cálculo de parámetro S11. (Una única frecuencia)

Los parámetros de Scattering, o parámetros [S] son parámetros muy importantes tanto para la fase de diseño como la fase de estudio y análisis de circuitos de microondas y en general para las radiocomunicaciones si se tiene en cuenta que las impedancias que presentan los distintos tipos de dispositivos, ya sean de radiación, de recepción o equipos intermedios y de procesado, requerirán de un buen diseño y en consecuencia adaptación para maximizar el aprovechamiento de recursos (potencia) entregados a estos. Es por este motivo entre otros por lo que los conceptos teóricos sobre estos parámetros tienen tanto interés y forman gran parte de los casos de prueba que conlleva este proyecto.

Antes de entrar en detalle sobre los parámetros [S], se considera importante repasar de manera breve el concepto de coeficiente de reflexión. Para un circuito de un puerto, el coeficiente de reflexión Γ es un parámetro clave que describe cómo una onda electromagnética se refleja en el punto de carga. Este coeficiente es particularmente relevante en el análisis de líneas de transmisión y bipuertos. El coeficiente de reflexión se define como la relación entre la onda incidente y la onda reflejada. Esto es aplicable, sea la onda de la índole que sea. Por lo que para una mejor comprensión se pasa a referir las ondas como ondas de potencia. En el caso que ocupa la entrega de potencia a dispositivos de radiación, no es posible terminar el puerto de salida con los siguientes formatos: circuito abierto, corto circuito. Cuando se trabaja a frecuencias tan altas, imponer la salida de un dispositivo en circuito abierto o corto circuito supondría inestabilidad y oscilación a la salida de este. [20]

La admitancia de un cortocircuito, la impedancia de un circuito abierto, y los parámetros híbridos [h] y [g] son comúnmente utilizados para formular modelos de circuitos de dos puertos que relacionan la impedancia de entrada y salida, la transferencia directa e inversa de redes lineales relativamente complejas. Estos modelos son simples en el sentido de que incorporan pocos parámetros eléctricos cuya medición o cálculo se basa en las características eléctricas de cortocircuitos o circuitos abiertos en los puertos de salida o entrada. Aunque la parametrización de estos modelos convencionales de dos puertos se puede hacer a frecuencias de señal relativamente bajas, las altas frecuencias a las que se trabaja en el procesamiento de señal presentan dos desafíos en la electrónica de banda ancha.

El primer desafío radica en que la presencia inevitable de inductancia parasitaria en las conexiones de circuitos hace que sea imposible alcanzar cortocircuitos perfectos a frecuencias altas. Por otro lado, si los caminos son de muy baja impedancia en los puertos de entrada o salida de los circuitos, estos pueden hacer que los dispositivos activos incorporados funcionen de manera no lineal o incluso fallen debido a excesivas corrientes. El segundo desafío, la inestabilidad que aparece en las redes electrónicas de alta frecuencia o banda ancha, se ve aumentada cuando estas redes se terminan con puertos de entrada o salida en circuito abierto.

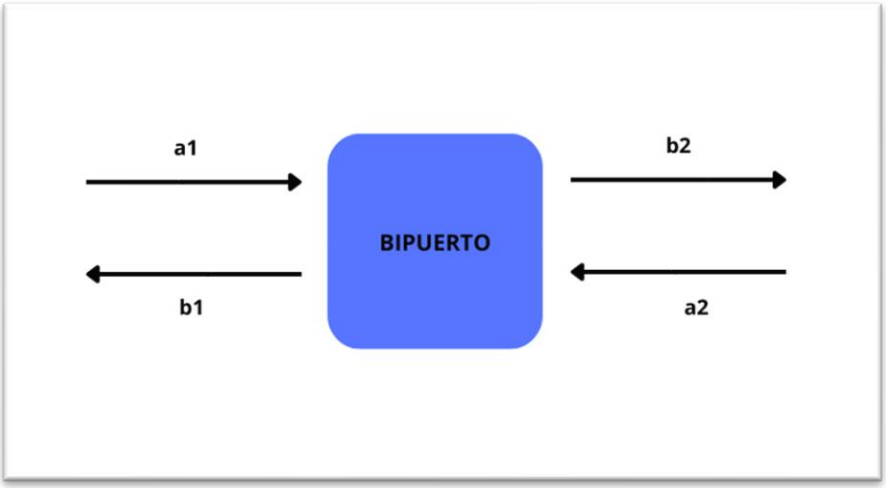
Por ejemplo, intentos de medir parámetros de impedancia en circuito abierto de un circuito de banda ancha son afectados por oscilaciones parasitarias en la red. El desafiante, si no imposible, problema planteado por la medición de parámetros convencionales de dos puertos motiva la caracterización de parámetros de dispersión o S de sistemas lineales de dos puertos.

A diferencia de los parámetros de impedancia, admitancia e híbridos, los parámetros de dispersión [S] se miden sin necesidad de cortocircuitar o abrir circuitos en los puertos de entrada y/o salida. En su lugar, estos puertos se terminan en impedancias características fijas y conocidas que a menudo son similares o incluso idénticas a las impedancias terminales del diseño. Los parámetros de dispersión medidos siempre se pueden convertir en parámetros de admitancia, impedancia y híbridos correspondientes, brindando un grado de confianza a los diseñadores ya que pueden estar más familiarizados con estos parámetros. Los parámetros de dispersión de una red lineal de dos puertos relacionan ondas de energía incidente y reflejada en los puertos de entrada y salida. Siendo más fácil utilizar los parámetros [S], que utilizar voltajes y corrientes en ambos puertos. De ahí que se utilicen para el diseño de amplificadores de microondas. Los parámetros [S] se extraen de manera conveniente en el laboratorio, pero también son explícitamente útiles en el diseño. La razón subyace de que los voltajes y corrientes son más difíciles de cuantificar a frecuencias de microondas mientras que la energía y los niveles de potencia promedio no.

Cuando ambos puertos no tienen perdidas, los parámetros [S] tienen una propiedad clave en el diseño de redes de adaptación sin pérdidas. Esto es que sirven para lograr la transferencia máxima de potencia de señal desde una fuente hasta su carga.

$$\begin{pmatrix} b1 \\ . \\ . \\ . \\ bn \end{pmatrix} = \begin{pmatrix} S_{11} & S_{12} & . & . & S_{1n} \\ S_{21} & . & . & . & . & S_{1n} \\ S_{31} & . & . & . & . & . & S_{1n} \\ S_{n1} & . & . & . & . & . & S_{nn} \end{pmatrix} * \begin{pmatrix} a_1 \\ . \\ . \\ . \\ a_n \end{pmatrix} \qquad \text{(Ecua. 1)}$$

La Ecuación (1) muestra la matriz de parámetros Scattering generalizada de orden n relacionada con ondas de potencia incidentes y regresivas. Siendo cualquier a_n la potencia que incide por cualquiera de sus puertos y los b_n valores de potencia que salen por sus puertos.



50. Ondas de potencia en un bipuerto genérico cualquiera. [21]

5.3.5 CASO 5. Cálculo de parámetro S11. (barrido en frecuencia)

Para definir el concepto teórico que abarca este caso de prueba, será de gran interés referirse al apartado referente al caso 4, ya que casi la totalidad de los conceptos teóricos son compartidos por estos dos casos. Siendo el criterio central, los parámetros de dispersión [S]. Con la diferencia resaltable de que en esta tarea de prueba el valor del parámetro de Scattering será entregado al usuario en forma de grafica. Siendo así para poder analizar de manera visual el comportamiento y a su vez el valor cambiante de dicho parámetro con la evolución de la frecuencia.

La frecuencia de resonancia en antenas es un concepto clave que se refiere a la frecuencia específica en la cual la antena opera de manera más eficiente, logrando la máxima transferencia de energía entre la línea de transmisión y el espacio libre. En este punto, la antena muestra una mayor capacidad para irradiar o recibir señales electromagnéticas. Este fenómeno de resonancia se caracteriza por una impedancia adaptada, minimizando las pérdidas de energía y optimizando la eficacia del sistema. La frecuencia de resonancia está vinculada a la cancelación entre las reactancias inductivas y capacitivas en la estructura de la antena. En el punto de resonancia, estas reactancias son iguales en magnitud, pero opuestas en fase, resultando en una impedancia puramente resistiva. Esto reduce las pérdidas y mejora la eficiencia de la antena al permitirle irradiar o recibir señales de manera más efectiva en esa frecuencia específica, haciendo la elección de la frecuencia de resonancia esencial para el diseño óptimo en aplicaciones particulares. [22]

En este contexto, se explorará entre las diferentes bandas donde la frecuencia de resonancia es un hecho, con la intención de seleccionar aquella que ofrezca las mejores prestaciones. La decisión se basará en el criterio de lograr un ahorro significativo de potencia inyectada o, en su defecto, un mayor y utilizable ancho de banda, de acuerdo con las necesidades específicas del diseñador o fabricante.

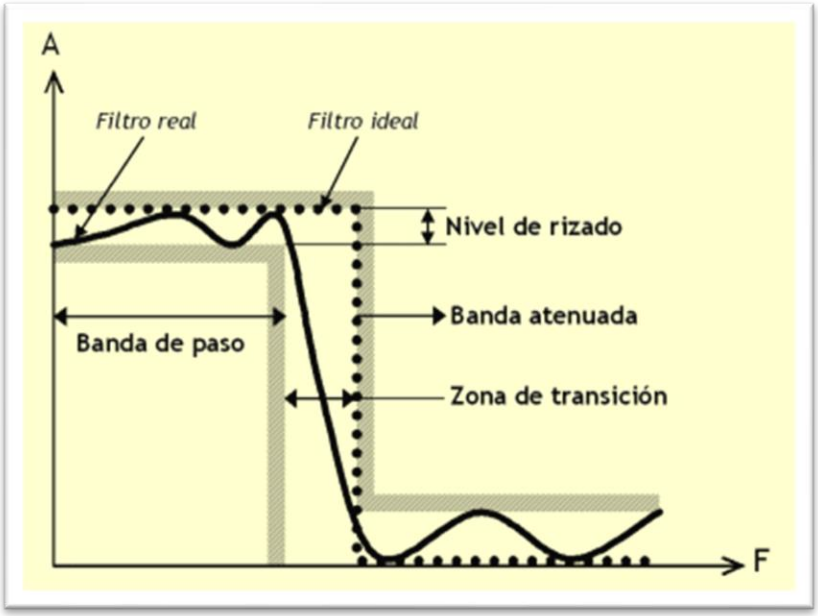
5.3.6 CASO 6. Respuesta en frecuencia de un filtro (LP, BP, HP)

Con este caso de prueba se persigue graficar la respuesta en frecuencia del comportamiento de un filtro como dispositivo bajo prueba. Para comenzar se clarifican algunos conceptos teóricos referentes a conceptos de filtros. Un filtro se puede describir como un dispositivo que altera de manera controlada una señal que atraviesa por él. [23] A veces se utiliza la palabra “filtro" para referirse a dispositivos diseñados para seleccionar frecuencias determinadas, permitiendo el paso de señales en bandas concretas de frecuencia mientras otras son bloqueadas. Existen definiciones más exactas, pero en general la mayoría de los filtros que ocupan interés en este proyecto son selectores de frecuencias.

Los filtros o selectores de frecuencia, conocidos como filtros, hacen posible el paso de señales en ciertas bandas referidas como bandas de paso, a su vez bloquean frecuencias no deseadas. Estas bandas son referidas como bandas de corte. Teóricamente, las bandas

de paso deberán permitir transmisiones sin distorsión, mientras que en la banda de corte la ganancia debería ser nula con el fin de eliminar las frecuencias no deseadas.

Existen cuatro tipos de filtros selectores de frecuencia: paso bajas, paso altas, paso banda y banda de rechazo. También existen filtros multibanda, como los filtros peine, que pueden rechazar o permitir el paso de los armónicos de una frecuencia específica. Para el desarrollo de las pruebas contenidas en este proyecto, los filtros selectores que ocupan interés serán los tres primeros mencionados. A continuación, se pasa a describir las distintas bandas que componen un filtro, y de esta manera tener una mayor comprensión de la gráfica que obtendremos al ejecutar este caso de prueba. Las distintas bandas se resumen en banda de paso, banda de corte, banda de transición o banda de atenuación. La banda de paso facilita el paso de las frecuencias que se encuentran dentro de un intervalo deseado, a su vez reduce o bloquea aquellas que caen fuera de este rango. En condiciones ideales, se espera que la transmisión en la banda de paso sea libre de distorsiones. Caso que no se consigue en filtros reales debido a su compleja composición y cara implementación. Es por esto que en filtros reales existe un rizado en la banda de paso. [24]



51. Detalle de un filtro LP. Nivel de rizado. [24]

La banda de corte, también conocida como banda de rechazo es la banda que bloquea o atenúa las frecuencias que caen dentro de un rango no deseado, permitiendo el paso de aquellas fuera de dicho rango. Idealmente la ganancia en la banda de corte debería

ser nula. Siendo esto difícil de conseguir debido a un nivel de rizado que al igual que a la banda de paso, afecta también a la banda de corte. Y nuevamente siendo el motivo las limitaciones debidas a la fase de diseño.

La banda de transición es la región específica en la respuesta en frecuencia que se encuentra entre la banda de paso y la banda de corte. Su función principal es permitir una transición suave y gradual en la atenuación de las frecuencias a medida que se pasa de la región donde el filtro permite el paso de señales (banda de paso) a la región donde bloquea o atenúa esas señales (banda de corte).

Estas bandas definen la respuesta en frecuencia de un filtro y son fundamentales para entender cómo el filtro afecta diferentes componentes de la señal en función de su frecuencia. Dependiendo del tipo de filtro (paso bajo, paso alto, paso banda, rechazo de banda), la configuración específica de estas bandas variará dependiendo del comportamiento del filtro que se requiera. [25]

5.3.7 CASO 7. Cálculo Parámetros de Scattering (4 parámetros)

Los conceptos teóricos necesarios para la comprensión de este caso de prueba son detallados en el apartado referente al caso 4. Siendo los parámetros de dispersión o Scattering los que ocupan la mayor importancia en este desarrollo al igual que en los casos mencionados anteriormente.

5.3.8 CASO 8. Caracterización de Impedancia de un DUT

Siguiendo con la comprensión de los parámetros de dispersión [S], es crucial destacar conceptos estrechamente vinculados. Resulta necesario familiarizarse con ideas como el coeficiente de reflexión y la impedancia de entrada de un circuito eléctrico, ya que existe una relación fundamentada en transformaciones que conecta estos conceptos.

El coeficiente de reflexión se considera un concepto fundamental en el ámbito de los circuitos eléctricos y las ondas electromagnéticas. Se refiere a la porción o cantidad de una onda incidente en un circuito o línea de transmisión que se refleja al incidir en una interfaz entre dos medios con diferentes impedancias. Comúnmente representado como Γ. Este coeficiente cuantifica la magnitud de la onda reflejada en comparación con la onda incidente. Un valor del coeficiente de reflexión cercano a cero indica una baja reflexión,

sugiriendo así una buena adaptación de impedancias, mientras que un valor cercano a uno señala una alta reflexión, indicando posiblemente una mala adaptación. La comprensión del coeficiente de reflexión es crucial en el diseño y análisis de sistemas de transmisión de señales, ya que ofrece información fundamental sobre la interacción de las ondas electromagnéticas con dispositivos y componentes eléctricos. [26]

La impedancia de un circuito eléctrico se define como la oposición o medida de resistencia que el circuito presenta al flujo de corriente alterna (CA). El concepto de impedancia abarca tanto la resistencia como la reactancia, siendo la resistencia la parte real y la reactancia la parte imaginaria de la impedancia. Esta última vinculada a los elementos capacitivos e inductivos que presenta el circuito. En comparación con la resistencia, la impedancia es un enfoque más completo que considera las complejidades asociadas con la variación de la frecuencia en los circuitos eléctricos. La relación que une el concepto de impedancia y coeficiente de reflexión está basado en una transformación bilineal. [27]

Una transformación bilineal es una operación matemática que combina dos funciones lineales para producir una nueva función. En el caso de matrices, una transformación bilineal toma dos matrices y produce un tercer resultado mediante la combinación lineal de los elementos de ambas matrices. Matemáticamente, si A y B son matrices, la transformación bilineal se puede expresar como:

$$f(A,B) = A * M * B^T$$

La relación existente entre el coeficiente de reflexión e impedancia tras la transformación bilineal da como resultado una conversión directa bilateral entre ambos conceptos mencionados.

$$\Gamma = \frac{Z - Z0}{Z + Z0}$$

Siendo Z la impedancia en la interfaz donde se quiere calcular el coeficiente de reflexión. Por tanto, Z puede tomar valores como

$$\{zl(l = carga), \qquad z(l = en\ cualquier\ punto), \qquad zin(l = 0)\}$$

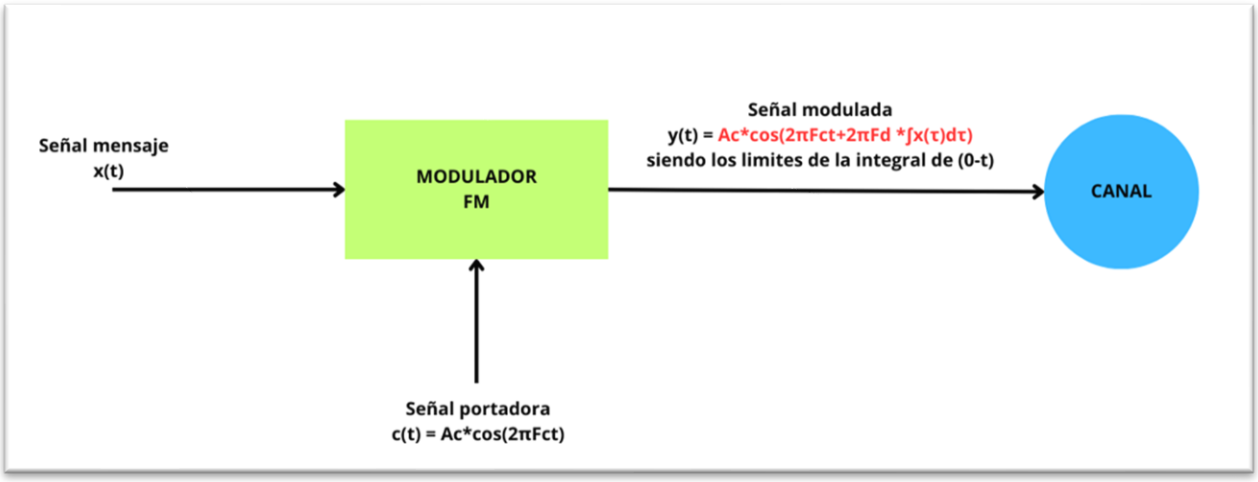
En el contexto de circuitos de alta frecuencia y microondas, hemos explorado el concepto del coeficiente de reflexión (Γ) y su relación con el parámetro S11. El coeficiente

de reflexión refleja la proporción de onda que refleja en un puerto de un dispositivo al encontrarse con una transición entre dos medios con diferentes impedancias. Al analizar un sistema de dos puertos, S11 se revela como un parámetro fundamental, describiendo la cantidad de onda reflejada en el puerto 1 en comparación con la onda incidente. Aunque las magnitudes de Γ y S11 son iguales, la fase (θ) puede variar, destacando la importancia de comprender no solo la cantidad de reflexión, sino también la información de fase asociada para interpretar con precisión el comportamiento de las ondas en sistemas de dos puertos.

5.3.9 CASO 9. Transmisión de audio y Demodulación FM

Como caso de prueba final, se lleva a cabo la transmisión de un archivo de audio mediante la modulación en frecuencia (FM), seguida por la recepción de la señal a través de una transmisión no guiada. A continuación, se aplicará el proceso de demodulación en el equipo receptor correspondiente, finalizando con la escucha del fichero de audio a través de dicho equipo encargado de demodular la señal transmitida.

La transmisión del fichero de audio se realiza mediante técnicas de modulación de frecuencia. La modulación de frecuencia es una técnica de modulación utilizada en sistemas de comunicación para transmitir información a través de variaciones en la frecuencia de una onda portadora. En FM, la amplitud de la señal portadora permanece constante, pero su frecuencia se modifica proporcionalmente a la amplitud de la señal de entrada o moduladora. Este proceso proporciona ventajas en términos de calidad de la señal y resistencia frente al ruido, ya que las variaciones en la amplitud no afectan significativamente la calidad de la transmisión. La frecuencia instantánea de la señal modulada está directamente relacionada con la amplitud de la señal de entrada, lo que facilita la recuperación precisa de la información en el receptor mediante el proceso de demodulación.



52. Esquema modulación FM.

6 DESARROLLO

En el marco del proyecto, el presente apartado de desarrollo constituye el núcleo central donde se abordarán detalladamente los aspectos clave y las fases implementadas que configuran la ejecución y materialización del proyecto. Aquí, se explorarán las metodologías, herramientas y enfoques empleados para la consecución de los objetivos planteados. A lo largo de este desarrollo, se examinará la implementación de las distintas herramientas, la superación de desafíos técnicos y la integración de componentes esenciales, proporcionando una visión integral del proceso de desarrollo y aplicación de este a los distintos casos de prueba que se persiguen, con el fin de alcanzar la culminación exitosa del proyecto.

En consonancia con la estructura previamente detallada, se procede a desglosar los diversos casos de prueba que han sido ejecutados, abarcando integralmente el conjunto de este proceso de desarrollo en sus distintas fases. Este análisis detallado permitirá examinar de manera sistemática la ejecución de los casos de prueba, destacando las variables clave evaluadas y proporcionando una visión completa de la implementación y validación de los componentes esenciales en el contexto del proyecto.

Como parte común a todos los casos de prueba, se sigue un orden uniforme en la ejecución de las diversas etapas de diseño, prueba e implementación. Inicialmente, se procede al desarrollo de funciones en Python diseñadas para interactuar con los

instrumentos relevantes. Posteriormente, se implementan funciones específicas que automatizan la ejecución de comandos sobre dichos instrumentos. Se lleva a cabo una revisión exhaustiva para garantizar la correcta ejecución de todos los procesos. Finalmente, se aplica Robot Framework para automatizar la interacción con los equipos, permitiendo la obtención de información detallada y generación de archivos detallados sobre los procesos realizados. Este enfoque sistemático proporciona una base sólida para la realización eficiente y efectiva de los casos de prueba en el marco del proyecto. Para una mejor evaluación del conjunto de scripts desarrollados a lo largo de la fase inicial, se crea un repositorio en GitHub para dar accesibilidad al conjunto de funciones y scripts desarrollados. Este contenido se encuentra localizado en el siguiente enlace citado.

- https://github.com/imaranuss/automation.git

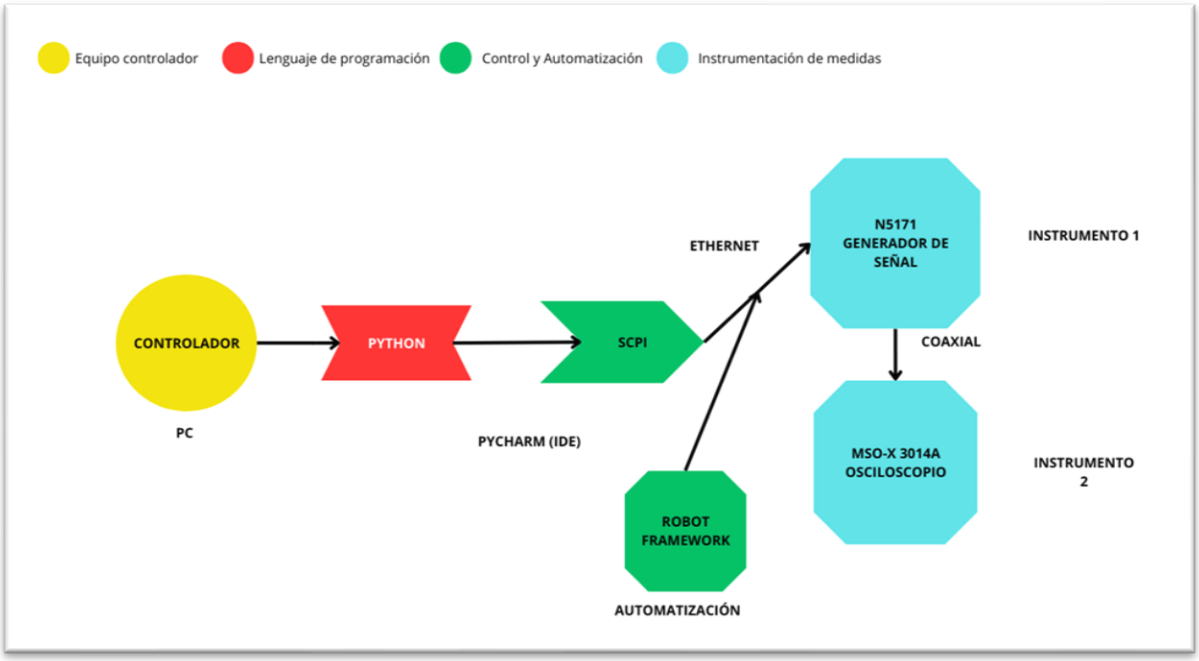
Una vez el diseño de los distintos módulos que componen la estructura concreta de cada caso de prueba han sido desarrollados, se pasa a la ejecución para llevar a cabo las diversas tareas e interacciones. Las cuales se basan en la comunicación mediante el envío de comandos SCPI a los equipos de laboratorio, Como último paso se procede a la fase de pruebas. En esta etapa, se realiza un análisis detenido para asegurar con precisión que las funciones y comunicaciones con los instrumentos se llevan a cabo de manera efectiva. Simultáneamente, se examina el tráfico basado en TCP/IP entre los equipos para verificar el intercambio de información entre el equipo controlador y los instrumentos al ejecutar las estructuras de código desarrolladas. Para este propósito, se realiza una captura continua de información mediante Wireshark durante la implementación de los casos de prueba. Este enfoque garantiza una evaluación exhaustiva de la funcionalidad y la interoperabilidad de los componentes implementados.

Para entrar en contexto, se menciona que la totalidad de los procesos, ya sean de diseño, desarrollo, implementación y pruebas se han realizado en el entorno de trabajo que comprende el laboratorio “José Curpían” de Radiofrecuencia y Microondas de la EPSL.

53. Laboratorio de Radiofrecuencia y Microondas, EPSL. (L-137)

Es en este laboratorio donde se realiza de manera practica la docencia de algunas de las asignaturas. Entre las que figuran; fundamentos de radiocomunicaciones, fundamentos de ingeniería de microondas, comunicaciones ópticas y otras más que componen el Grado en Ingeniería de Tecnologías de Telecomunicación en la Escuela Politécnica Superior de Linares. Ubicada en la Avenida de la Universidad, s/n, 23700 Linares, Jaén

6.1.1 CASO 1. Proceso de Generación de una señal



54. Diagrama de flujo para el Caso 1.

Como se ha visto anteriormente en el capítulo 5 de conceptos teóricos, este caso de prueba se basa en la generación de una señal u onda la cual será parametrizada mediante un valor de frecuencia y un valor de amplitud, el cual podrá ser ajustado en unidades lineales de tensión (voltios) o unidades logarítmicas (dBm´s).

Para ello, el primer paso es el desarrollo de un script el cual contendrá no solo las funciones requeridas para este caso de prueba, si no para todo el resto de casos de prueba. De alguna manera se considerará que este script es una librería la cual tiene todas las herramientas o funciones necesarias para los nueve casos de prueba que se han desarrollado en el contexto de este proyecto. Será requerida la importación del script Keywords en los distintos ficheros Robot Framework. El fichero recibe el nombre de "Keywords" en referencia al apartado de keywords que forma parte de la estructura general y simplificada de Robot Framework.

Para este caso de prueba, se requiere la interacción entre el generador de señales N5171B y para visualizar correctamente la implementación, se hace uso del osciloscopio MSO-X 3014A del mismo fabricante, keysight. El cual se encuentra en el laboratorio de radiofrecuencia y microondas de la EPSL, dependencia L-137.

55. Osciloscopio Keysight MSO-X 3014A.

El primer paso es la búsqueda del instrumento necesario para la realización del caso de prueba. Siendo este el generador de señales N5171B. Para ello los equipos necesarios deben de estar conectados a la red LAN mediante cableado Ethernet (conectores RJ-45). Mediante el desarrollo de una función en Python, se hace un escaneo de los equipos conectados y se determina bajo qué dirección IP se encuentra el generador de funciones.

El uso de la librería socket que tiene Python es necesaria debido a la decisión tomada en este caso de implementar esta. Para ello se procede con el comando “import socket”. En este caso tras la ejecución de la función (find_N5171B) se obtiene una salida en el terminal del IDE que devuelve la dirección IP del equipo.

```
C:\Users\Imara\AppData\Local\Programs\Python\Python37
Keysight N5171B found with IP: 192.168.12.123

Process finished with exit code 0
```

56. Salida en terminal con dirección IP encontrada.

Tras haber cumplimentado este paso, se procede a abrir conexión con dicho equipo. Aquí es donde entra en juego el concepto que aporta VISA.

Primeramente, se importa la librería PyVisa para seguidamente crear un objeto de control sobre el instrumento. Tanto este proceso, como el anterior son comunes en todos los casos de prueba para la detección, descubrimiento de la dirección IP y posterior conexión con el equipo.

Una vez se haya obtenido la dirección IP, esta se convierte en un componente esencial para los procedimientos subsiguientes, ya que se convierte en un parámetro imperativo que se debe proporcionar a cada función para que pueda llevar a cabo las tareas específicas asignadas. La siguiente operación que se ejecuta aborda la conexión y generación de una instancia u objeto que controla, envía y solicita datos de control, comandos SCPI y procesos que el instrumento debe realizar. Al invocar esta función, la cual recibe el nombre de (connect_to_signal_generator()), se obtiene en el terminal del entorno de desarrollo integrado (IDE) un string que representa el objeto recién creado, a partir del cual se llevarán a cabo las acciones mencionadas anteriormente.

```
C:\Users\Imara\AppData\Local\Programs\Python\Python37\py
TCPIPInstrument at TCPIP0::192.168.12.123::inst0::INSTR

Process finished with exit code 0
```

57. Salida en el terminal con el objeto de interacción creado.

Se puede decir que se ha abierto una conexión directa con el instrumento N5171B. A partir de este punto es esencial determinar que parámetros se desean ajustar en el generador de funciones para cumplir con el propósito de este caso de prueba. Como el mismo nombre indica, este primer caso de prueba se basa en la generación de una forma de onda. La cual requiere del ajuste de parámetros de frecuencia, amplitud. Para ello se desarrolla una función que ajuste dichos parámetros en el instrumento mediante comandos SCPI.

El formato estructural del comando que se genera para el ajuste en remoto del generador de funciones conlleva la siguiente estructura.

```
instrument_generator.write(f":FREQ {frequency}")
```

58. línea de comando enviada para ajustar valor de frecuencia.

Se puede observar que la sección que ocupa “instrument_generator” es el objeto creado anteriormente mediante VISA. El cual sirve de control y a su vez de conexión con el instrumento. Seguidamente se utiliza “.write”. Este método se utiliza para enviar comandos al instrumento. Cuando se invoca “.write”, el argumento proporcionado se transmite al dispositivo conectado, permitiendo enviar instrucciones o configuraciones al mismo. En el caso que ocupa el ejemplo en la ilustración 58, este comando solicita un ajuste de frecuencia cuyo valor estará determinado por la variable “{frequency}”.

Existe un método más y es “.query”. Este método se utiliza para enviar comandos que solicitan información o mediciones específicas al instrumento y recibir la respuesta. A diferencia de “.write”, “.query” espera una respuesta del instrumento y devuelve la respuesta al programa. La sintaxis es igual a la referida en la ilustración 58, sustituyendo el método write por el método query.

Una vez aclarado el formato o sintaxis de solicitud y control con el instrumento, se detalla la estructura general de la prueba. Esta consta de una búsqueda del instrumento mediante la librería socket con el fin de obtener la dirección IP bajo la que se aloja. Seguidamente se crea un objeto de control con el cual se comienza a mandar comandos SCPI. En este caso los comandos necesarios para lograr el propósito principal de este caso de prueba son: (*RST, :FREQ <valor de frecuencia>, :POW:LEV:IMM:AMPL <valor de amplitud (en unidades de tensión o decibelios referidos al milivoltio)>). Se puede apreciar que el comando que procede con la acción de reseteo del equipo es un comando común del estándar IEEE 488.2, tal y como se menciona en el capítulo 5 bajo el apartado de SCPI. Este comando volverá el instrumento a sus ajustes de fabrica eliminando posibles trazas pertenecientes a mediciones anteriores con el fin de evitar posibles errores.

Una vez el proceso de reseteo ha finalizado, una secuencia de comandos con la sintaxis general de comunicación a la cual hace referencia la ilustración 58, se envían al equipo para que ejecute las acciones asociadas a los comandos.

En concreto, este primer caso de prueba se basa en la generación de una forma de onda con el ajuste de los parámetros necesarios, luego todo este proceso se compacta dentro de una función implementada en el fichero denominado “keywords”, pasando al siguiente paso el cual es la decisión de aplicar modulación a la señal generada o directamente sacar esta por la salida RF de la que el equipo dispone. Como paso intermedio se activa la salida RF y se llama una función desarrollada para visualizar el resultado de la ejecución en su totalidad. Esta función es llamada (visualiza()). Como pasos finales, se desactiva la salida RF, y finalmente se termina o extingue la conexión con el equipo, principalmente para liberar recursos.

Por lo tanto, después de seguir el procedimiento descrito, se logra que el generador de funciones ejecute la secuencia de comandos enviados, resultando en la visualización de la señal. Este proceso se automatiza completamente a través del framework de Robot, el cual realiza automáticamente las diversas etapas mencionadas anteriormente y posteriormente genera varios archivos. Estos archivos contienen información clasificada, conforme se detalla en el capítulo 5, apartado de generación de reportes en informes.

59. Conexión entre equipos para ver señal generada y enviada. Caso 1.

6.1.2 CASO 2. Proceso de Generación de Barrido en Frecuencia

El objetivo principal es realizar un barrido utilizando el generador de funciones N5171B. Para lograrlo, se establecerá la comunicación entre este equipo y el controlador mediante la transmisión de comandos SCPI. Este enfoque permitirá automatizar los procesos de medición utilizando el framework de Robot.

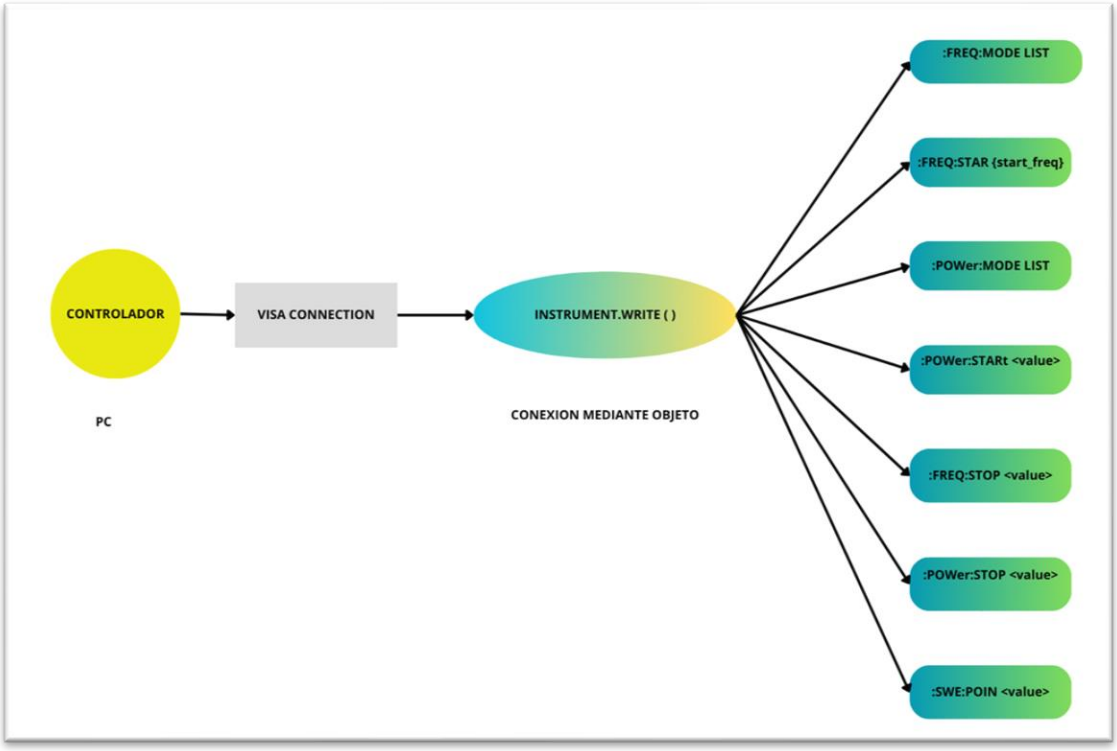
En este escenario de prueba, se sigue un enfoque similar al diagrama de procesos ilustrado en la Figura 54. Los instrumentos necesarios son el generador de funciones y el osciloscopio, siendo este último utilizado para visualizar el proceso generado por el generador de funciones. La función Python asociada a este caso de prueba lleva el nombre de (generador_barrido()) y ejecuta una serie de comandos SCPI para permitir al usuario parametrizar el tipo de barrido.

Los parámetros comunes con el caso anterior incluyen el ajuste del rango de frecuencias y el rango de amplitudes. Además, se incorporan ajustes adicionales, como el modo de medida, que, para este caso específico, se configura en "barrido". Existe la opción de que el proceso de medida sea exclusivamente en un punto con un valor fijo o, en este caso, un barrido que abarque un rango de valores. Estos son aplicables al ajuste de barrido en frecuencia y al barrido de variación de amplitud. Consecuentemente, se elige el modo de ejecución "barrido", cuyo comando SCPI equivalente es "LIST". La sintaxis enviada para este comando es la siguiente:

```
instrument_generator.write(":FREQ:MODE LIST")
```

60. Sintaxis utilizada para ajustar modo de generación de frecuencia (barrido).

Será requerimiento esencial el ajustar valores de frecuencia y amplitud que delimiten puntos de inicio y puntos de parada para parametrizar el barrido. Añadiendo un nuevo ajuste el cual determina la calidad de la función generada mediante el número de puntos del barrido. El usuario puede definir y modificar el número de puntos en un barrido según sus requisitos particulares. Este ajuste influye en la resolución y la cantidad de detalles presentes en la señal generada. Un mayor número de puntos en el barrido puede ofrecer una representación más minuciosa de la señal, aunque también puede demandar mayores recursos computacionales y de procesamiento. Para una mejor comprensión del concepto de puntos, se relación con el termino muestreo, el cual se tiene una familiarización mayor debido a los conocimientos adquiridos en el Grado de Ingeniería de Tecnologías de Telecomunicación, impartido en el campus de la EPSL.



61. Diagrame de flujo y orden de envíos comandos SCPI caso 2.

Tras detallar minuciosamente el procedimiento de envío de comandos en la Ilustración 61, el caso de prueba avanza de manera análoga al primer caso, y en consecuencia, los restantes casos que componen este proyecto seguirán procesos u órdenes similares tras ejecutar las tareas específicas de cada uno. En particular, acciones repetitivas como la decisión de aplicar modulación, la activación de salidas RF y la clausura definitiva de la conexión con el instrumento serán acciones recurrentes en los demás casos de prueba.

```
1   *** Settings ***
4
5   *** Variables ***
6   ${gen_address}
7   ${instrument_generator}
8   ${TEST_NAME}      main_barrido_freq
9   ${start_freq}             3e6
10  ${stop_freq}              10e6
11  ${start_amplitude}        -3
12  ${stop_amplitude}         -15
13  ${N_puntos}               2000
```

62. Script con ajuste de parámetros de barrido.

63. Parámetros ajustados automáticamente para el barrido. Caso 2.

6.1.3 CASO 3. Proceso de Modulación AM

Como objetivo principal de este caso de prueba, se busca lograr la automatización de un proceso de modulación de una señal mediante AM. Este procedimiento consta de varias fases enumeradas a continuación: la generación de una señal, donde los valores

paramétricos carecen de protagonismo en la perspectiva del caso de prueba. Posteriormente, se realizan los ajustes necesarios de una modulación AM con el fin de aplicarla a la señal generada. A continuación, se aplicarán otras funciones diseñadas en Python para la culminación del caso de prueba en su entorno específico como son el cierre de conexión con el equipo entre otras.

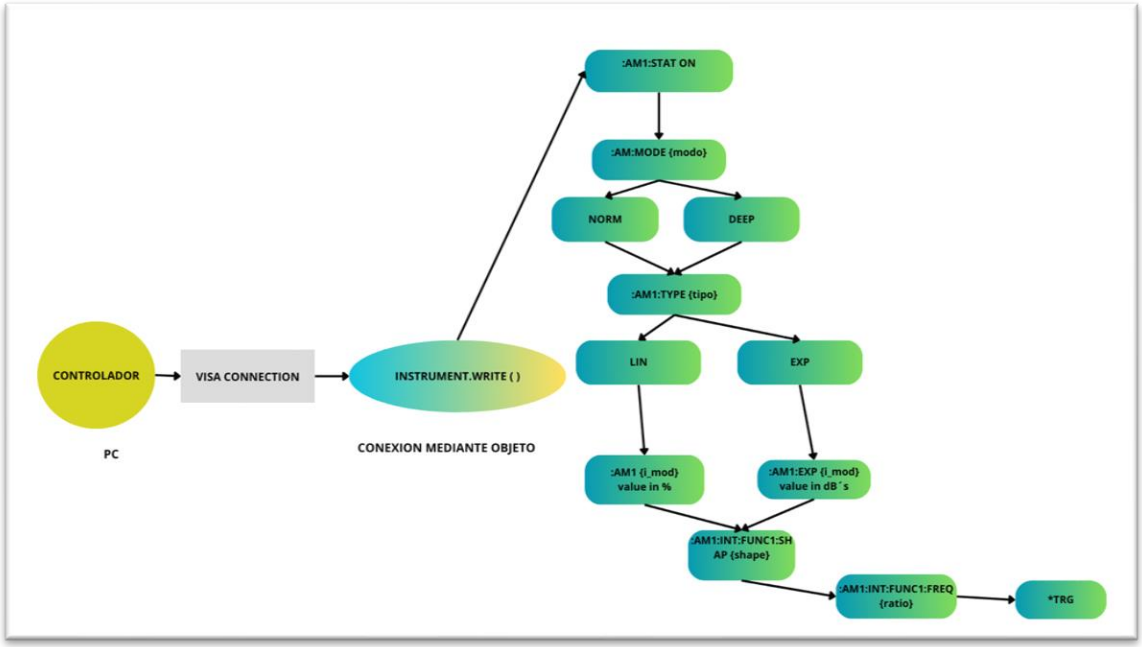
En el contexto de las modulaciones, los conceptos teóricos abordados en el capítulo 5 pueden ser tomadas como herramientas fundamentales para la correcta planificación y la posterior transmisión de comandos SCPI al instrumento. Estos principios brindan una base sólida que facilita el diseño preciso de las configuraciones de modulación, así como la generación de comandos específicos que optimizan el rendimiento del instrumento en la aplicación de dichas modulaciones. Para el caso que ocupa este apartado de prueba, es de interés el proceso de modulación AM. El cual brevemente se recuerda que consiste en modular mediante la variación de amplitud de la señal. En este punto, toman importancia conceptos como el índice de modulación, el cual se aborda con algo más de detalle en el apartado 5 de conceptos teóricos.

El conjunto de instrumentos necesario para llevar a cabo este caso de prueba incluye el generador de funciones y el osciloscopio. Es relevante destacar que este último instrumento no forma parte de la instrumentación principal en este desarrollo, principalmente porque carece de conexión tipo RJ-45, lo que imposibilita la conexión a través de Ethernet con el equipo.

64. Conexiones disponibles en panel frontal MSO-X 3014A.

65. Conexiones disponibles parte trasera MSO-X 3014A.

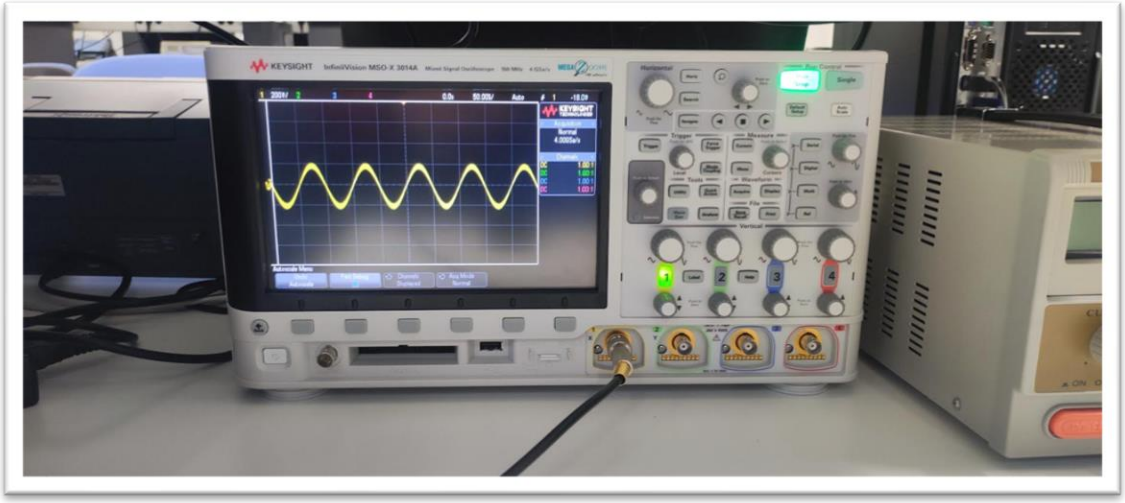
A pesar de esta limitación, su uso es necesario y se logra mediante alternativas de conexión. En este caso, se ha optado por la conexión mediante coaxial con conectores tipo BNC. Es necesario definir ciertos parámetros que influirán en otros parámetros consecuentes, dependiendo de las opciones ajustadas, como el modo de modulación y el tipo, entre otras. A continuación, se clarificarán estos conceptos mediante un diagrama jerárquico de los comandos SCPI.



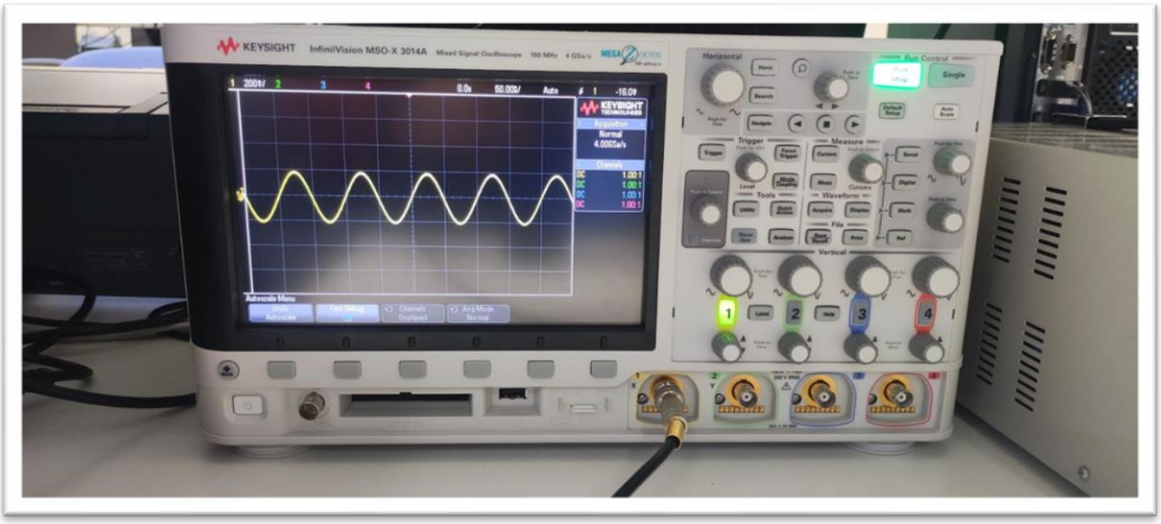
66. Diagrama de flujo jerarquizado para el caso 3.

Atendiendo a la estructura jerárquica de la ilustración 66, se puede alcanzar la comprensión del orden de comandos SCPI, los cuales han sido integrados en una estructura de código Python para dar forma al desarrollo de este caso de prueba. Posteriormente tanto este como el resto de casos que componen el desarrollo total de este proyecto, se les aplica la capa de automatización basada en Robot Framework con el fin de alcanzar la meta común que comparten todos los casos de prueba y a su vez la única finalidad de este proyecto, la automatización de toma de medidas.

67. Parámetros de modulación AM.



68. Señal con modulación AM aplicada.



69. Señal con modulación AM no aplicada.

6.1.4 CASO 4. Proceso de Calculo Parámetro S11

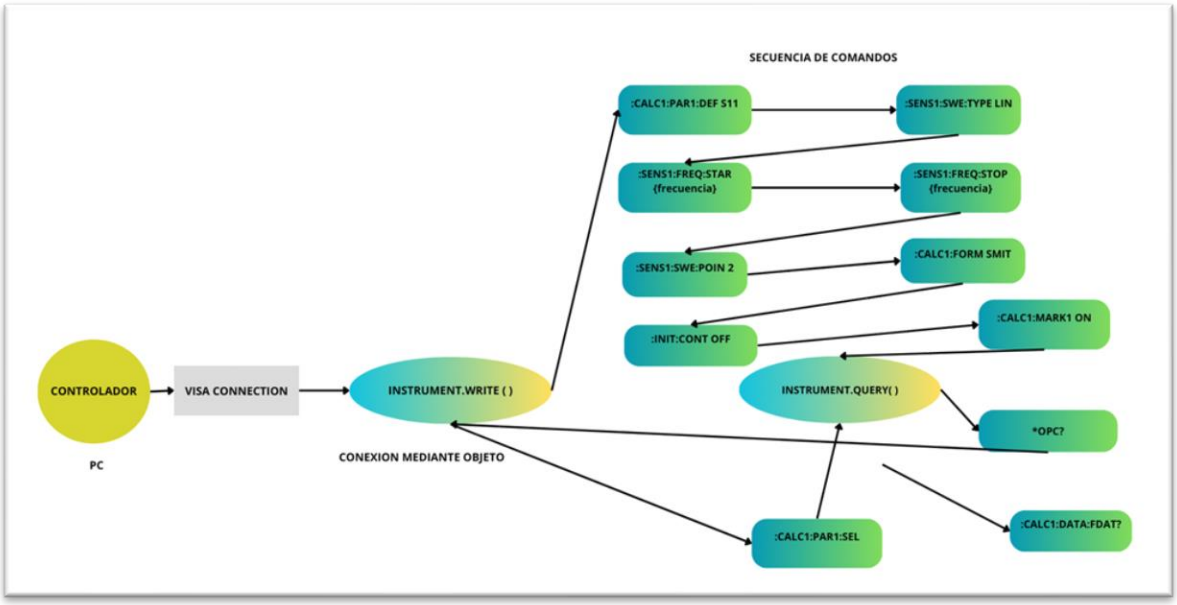
Es a partir de este caso de prueba, caso 4 desde donde se comienza a implementar medidas con dispositivos y componentes físicos, disponibles en el laboratorio L-137. A partir de este punto se persigue caracterizar distintos parámetros y medidas de relevancia en el ámbito de las radiocomunicaciones y microondas. De este modo se verán involucrados dispositivos bajo pruebas tales como los mencionados en el capítulo 4 bajo la denominación de componentes y DUTS.

Para este caso en particular, se pretende realizar medias con el instrumento analizador de redes vectoriales del fabricante Keysights modelo E5063A. Caso 4, que ocupa la importante tarea de obtener la medida correcta del parámetro S11 del DUT en cuestión. Cabe destacar que el desarrollo llevado a cabo en este apartado es aplicable a varios dispositivos bajo prueba. Con esto se pretende clarificar que, adecuando los parámetros o variables, componentes del script tanto en Python como Robot Framework, es posible aplicar dicha automatización a distintos dispositivos. Para la implementación y fase de pruebas de este caso en concreto se han utilizado varios tipos de antenas disponibles en el laboratorio. Entre las que figuran la antena Yagi-Uda, la antena biconica y la antena Yagi-Uda protegida por un radomo, todas estas descritas brevemente en el apartado de componentes en el capítulo 4.

Por tanto, el parámetro S11 en este apartado se obtendrá para una única frecuencia, siendo este concepto de interés para el análisis de comportamiento del dispositivo en determinados valores discretos e invariantes de frecuencia. A su vez de gran utilidad para el ámbito de diseño ya que, en este se persigue conseguir óptimos diseños de antenas con la finalidad de que estas radien a una determinada frecuencia. La grafica que se desea obtener será de tipo Carta de Smith, la cual es ampliamente conocida en el ámbito de las microondas debido a su gran utilidad para la obtención de resultados, a pesar de que fue creada hace ya bastantes años. Como primera función tras las ya conocidas: find_e5063a, connect_to_vectorial_analyzer(), reset_vectorial(), en este caso de prueba se comienza a implementar una función desarrollada previamente la cual ajusta la impedancia característica del instrumento de medida (VNA). Esta función recibe el nombre de set_impedancia() y simplemente brinda al usuario la opción de ajustar la impedancia característica del equipo a valores deseados. Es común la utilización de valores concretos como 50 Ω y 75 Ω.

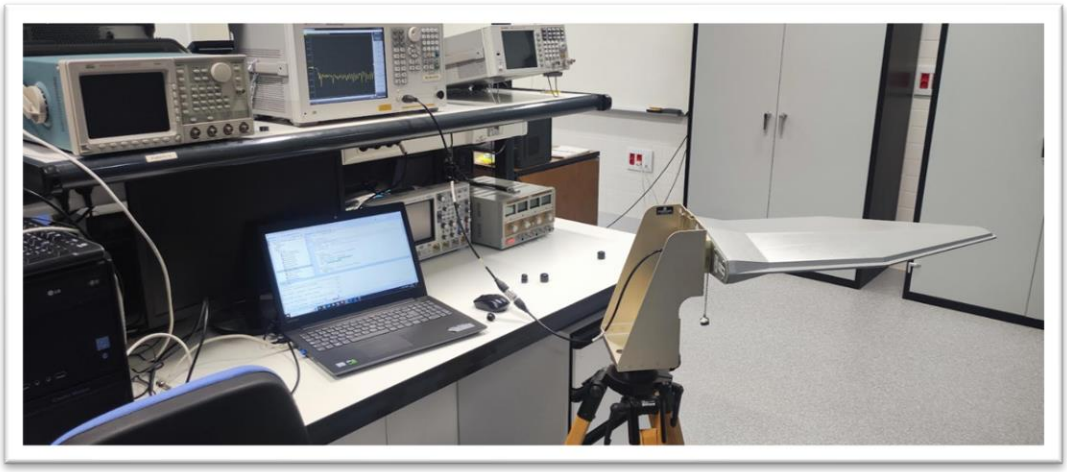
El comando SCPI correspondiente a este ajuste es (:SENS1:CORR:IMP {z0}) perteneciente a la categoría de comandos de subsistemas que componen la totalidad de comandos admitidos por el (VNA) modelo E5063A.

Una vez ajustada la impedancia característica deseada, se pasa a la función principal que conlleva este caso de prueba, nombrada (parametro_s11_sfreq()). La cual implementa el envío de una serie de comandos SCPI detallados en la siguiente ilustración.

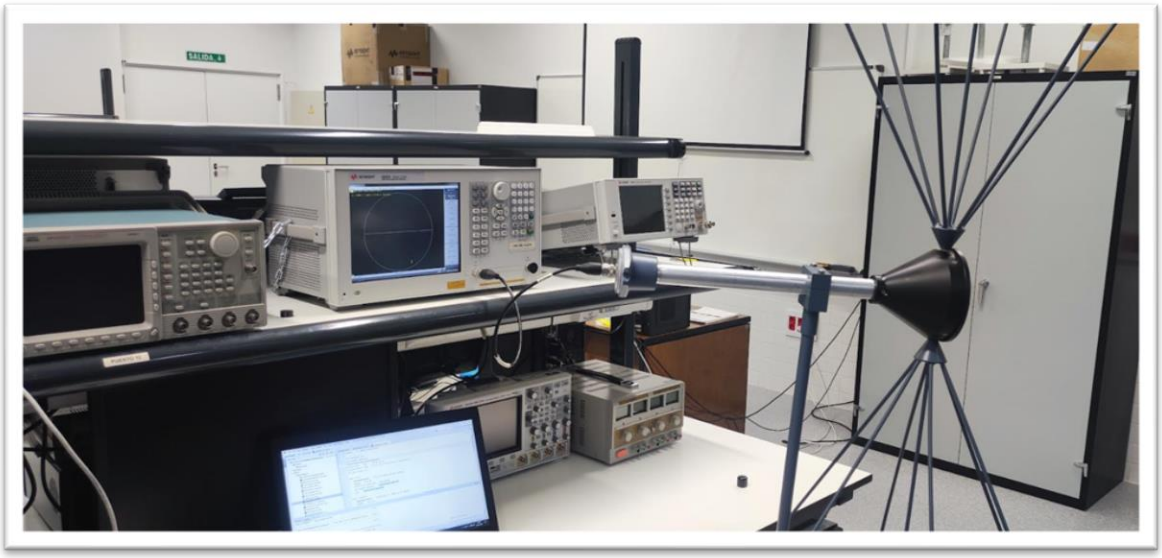


70. Diagrama de flujo jerárquico para el caso 4.

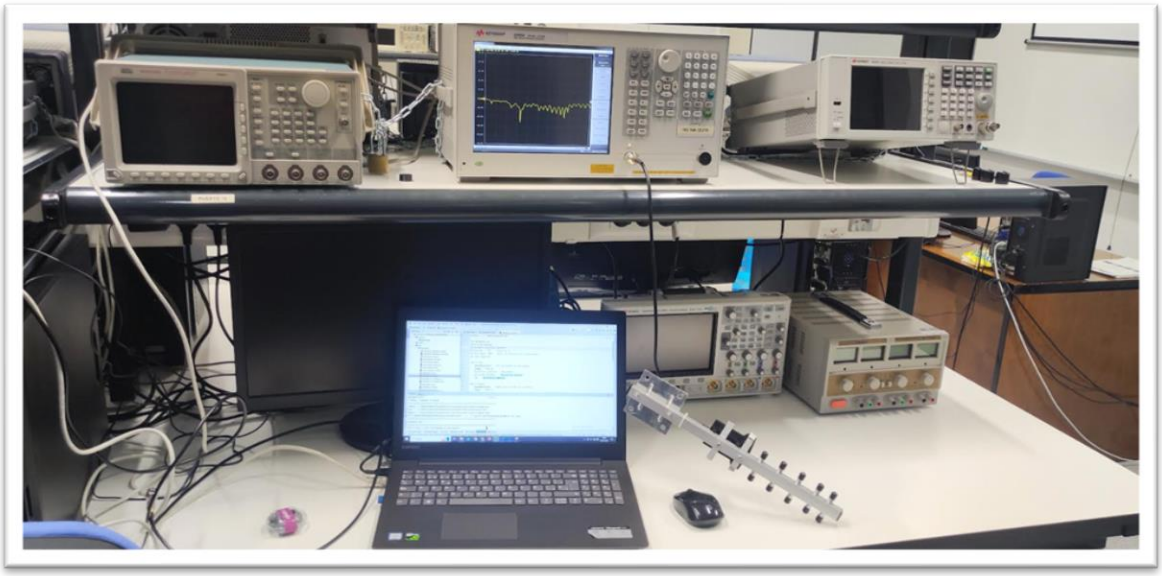
La disposición del VNA conectado al dispositivo bajo prueba se observa en la ilustración 71. Se detallan varias disposiciones con distintos dispositivos con el fin de corroborar la efectividad del script desarrollado y su aplicación a distintos DUTS con un alto grado de efectividad.



71. Antena Yagi-Uda con radomo siendo testeada. Caso 4.



72. Antena Biconica siendo testeada. Caso 4.



73. Antena Yagi-Uda a 2.45 GHz siendo testeada.

Se persigue exclusivamente obtener el parámetro S11 para el dispositivo bajo prueba a una única frecuencia. Esta información se obtendrá visualmente a través de la gráfica generada por pantalla del instrumento. Posteriormente, mediante la aplicación de Robot Framework y como resultado del proceso automatizado, el archivo .log generado por Robot desglosará de manera precisa las diversas tareas asignadas a la automatización, proporcionando los resultados del parámetro S11 de forma numérica.

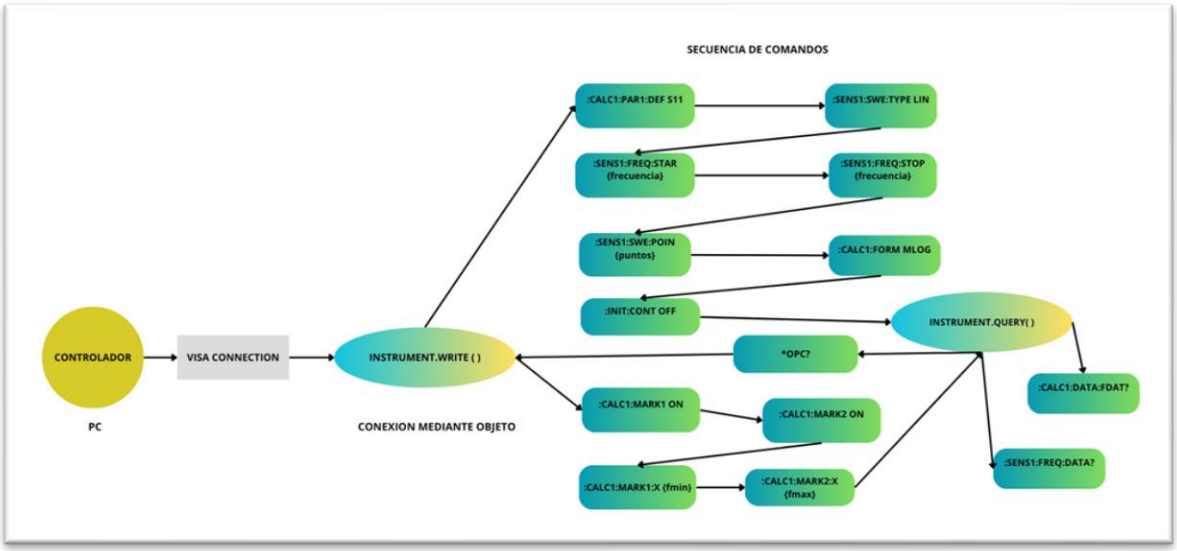
6.1.5 CASO 5. Proceso de Cálculo Parámetro S11 mediante barrido

El siguiente apartado que se procede a detallar guarda amplia similitud con el caso 4, con la salvedad de que, en esta tarea de prueba el parámetro S11 se caracteriza en un rango de frecuencias determinado. Siendo así, se persigue graficar el comportamiento del DUT a distintas frecuencias con el fin de localizar la frecuencia de resonancia, en el caso de antenas. Todo esto en conjunto con el valor dinámico a lo largo de la variación de frecuencias del parámetro S11. Siendo una opción a contemplar, la conversión mediante transformación bilineal de dicho parámetro a: coeficiente de reflexión o incluso valores de impedancia de entrada de un determinado dispositivo, si fuese necesario.

En cuanto a la frecuencia de resonancia, resulta fundamental comprender el concepto aplicado a las antenas. Una vez que se ha generado una representación gráfica del comportamiento del DUT, es posible analizar y determinar la frecuencia específica a la cual se debe diseñar la antena para lograr una utilización óptima de los recursos disponibles. Este análisis se realiza teniendo en cuenta varias condiciones esenciales entre las que se encuentran aquellas determinantes en comunicaciones: la potencia inyectada, máxima condición de radiación y el ancho de banda.

La elección de la frecuencia de resonancia puede estar influenciada por la necesidad de un ancho de banda más amplio, pero también pueden ser consideraciones como la eficiencia de radiación, la interferencia electromagnética y otros factores específicos de la aplicación.

El contenido anterior destaca que una antena, en realidad, presenta múltiples frecuencias de resonancia. Esto sugiere la posibilidad hipotética de utilizarla para resonar a varias frecuencias, aunque, en general, suelen diseñarse para la frecuencia en la cual se obtiene un mayor ancho de banda o según las necesidades mencionadas en el anterior párrafo. Siguiendo con el desarrollo de la tarea 5, la cual ocupa el caso de automatizar el proceso de medida del parámetro de Scattering S11, la consecución de funciones implementadas en el proceso de automatización viene expuestas en la siguiente ilustración.

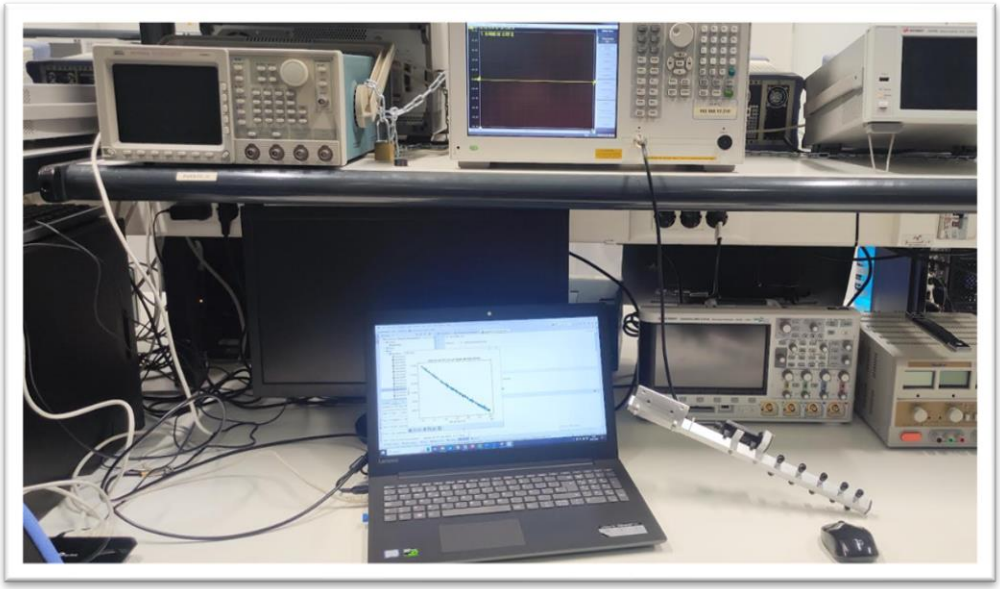


74. Diagrama de flujo jerarquizado caso 5.

Como se puede contemplar, existe una cantidad de funciones las cuales ya han sido implementadas en otros casos de prueba. Se ha tomado la decisión de unificar acciones concretas en distintas funciones con la intención de hacer más escalable el conjunto de scripts que componen la finalidad de este proyecto. Varias de estas son la función de encontrar la dirección IP del equipo, conectarse al equipo, reseteo del equipo, ajustar la impedancia característica del equipo, activar/desactivar la salida RF del equipo y finalmente la desconexión con el equipo que ocupa el caso. A su vez, cada caso de prueba contiene funciones específicas que componen el núcleo de cada tarea por individual. Haciendo que cada proceso preserve su independencia. Para el caso de prueba 5, esa función toma el nombre de (parametro_s11_barrido()). Cuya estructura es contenida bajo la ilustración 74. El proceso que se sigue está compuesto del desarrollo especifico de una función con Python, que envié comandos de acciones a ejecutar y su vez solicite datos e información procesada por el instrumento. Una vez se ha logrado el objetivo previo, el script genera una gráfica que describe visualmente el comportamiento o variación del parámetro S11 dentro del rango dinámico asignado. Permitiendo determinar las distintas bandas de frecuencia con el propósito de especificar la banda para la cual se diseñó la antena en relación al concepto de resonancia.

Como último paso, se procede a desarrollar la estructura que compone la prueba o tarea en el Robot Framework, mediante la cual se aplica el proceso de automatización de las medidas a tomar.

75. Antena Yagi con radomo siendo testeada. Caso 5.

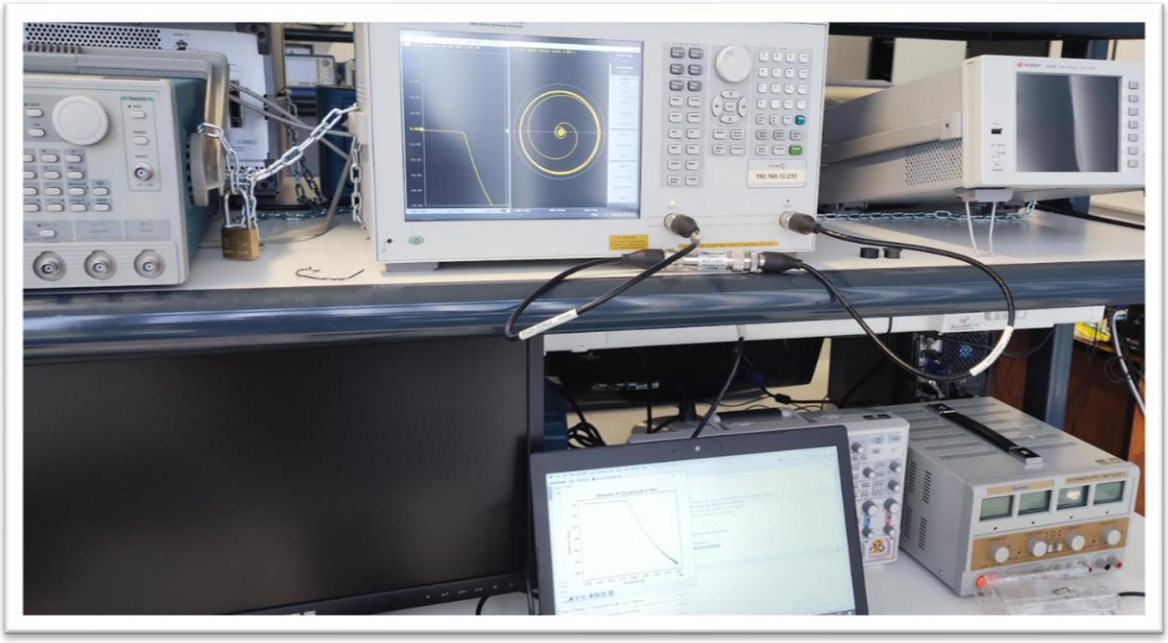


76. Antena Yagi-Uda a 2.45 GHz siendo testeada. Caso 5.

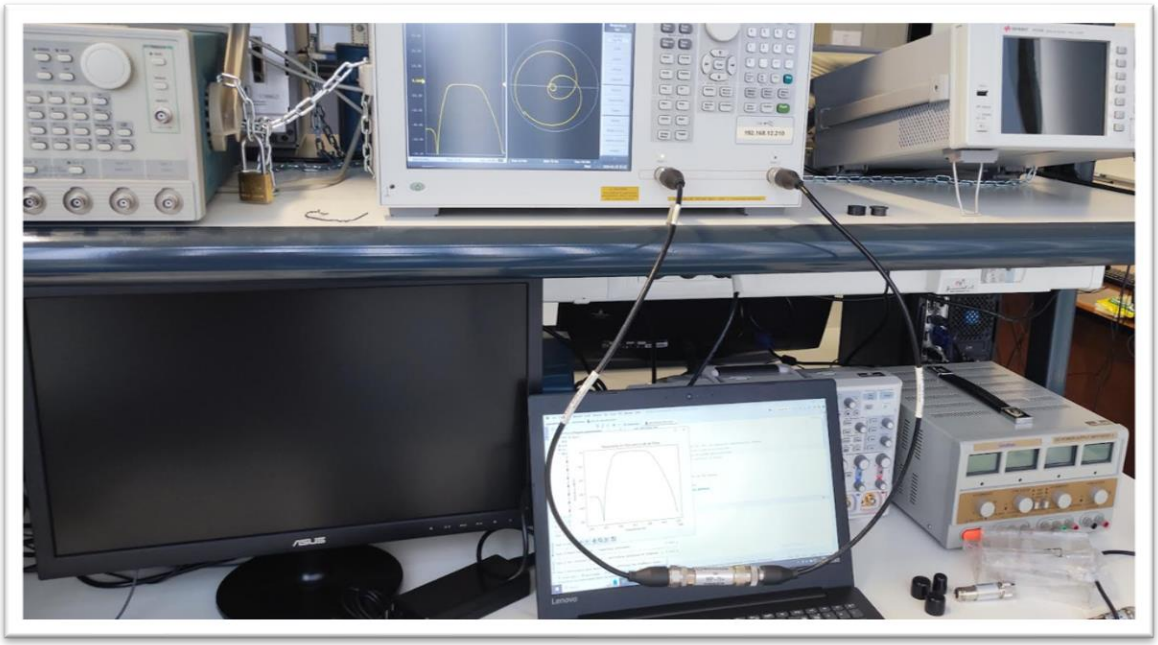
6.1.6 CASO 6. Proceso de Respuesta en Frecuencia de un filtro

Caso 6, el cual se basa en desarrollar una serie de funciones específicas que componen junto con el resto de funciones ya desarrolladas la librería especifica propia, con la cual se desarrolla este proyecto. El enfoque principal de este apartado o caso de prueba se basa en graficar el comportamiento en frecuencia de un DUT para posteriormente analizar de manera concisa las características que presenta en el dominio frecuencial el

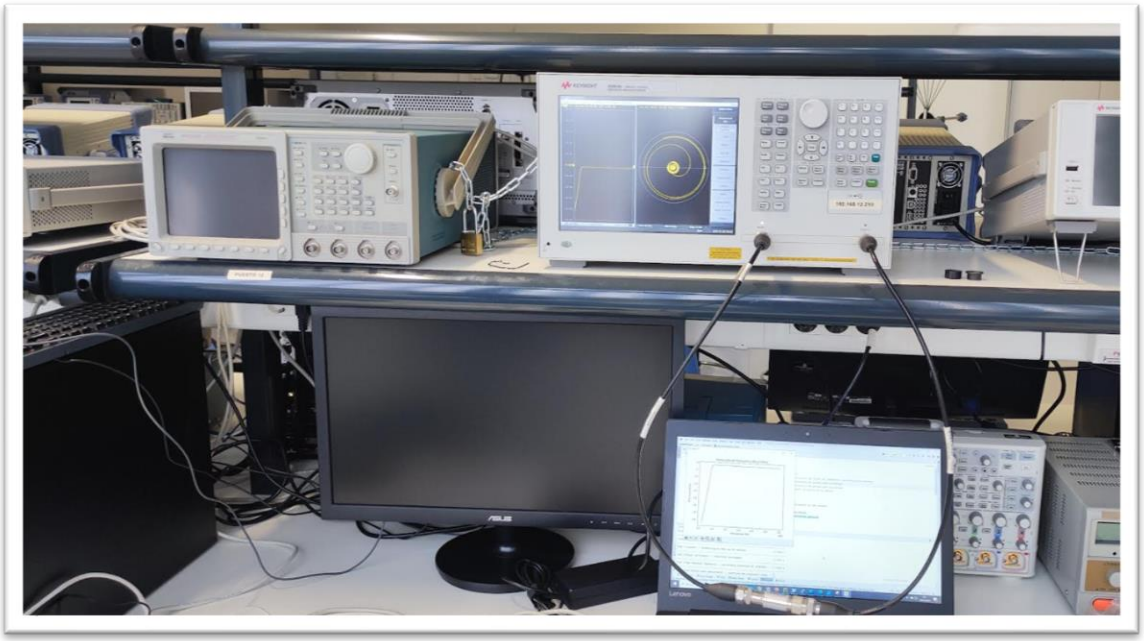
dispositivo bajo prueba. Existe la posibilidad de que el dispositivo bajo prueba sea de distinta índole, incluyendo filtros, como es el caso concreto de esta tarea de prueba.



77. Respuesta en frecuencia de un filtro LP.

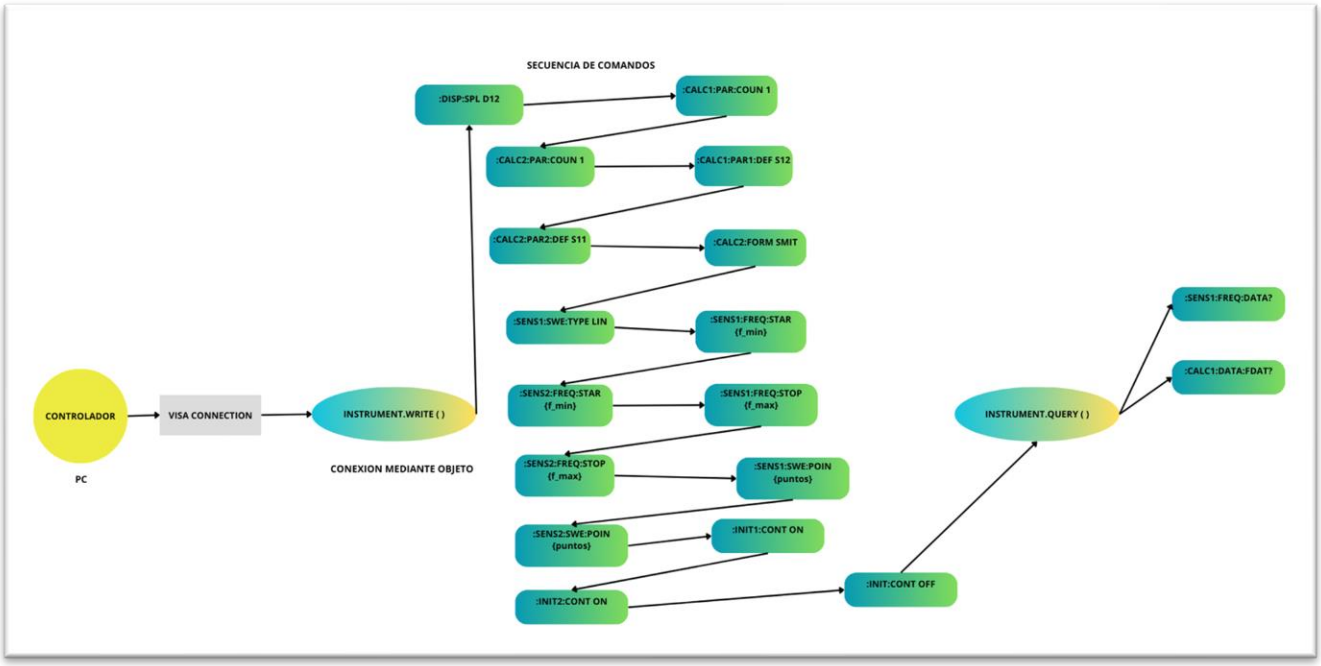


78. Respuesta en frecuencia de un filtro BP.



79. Respuesta en frecuencia de un filtro HP.

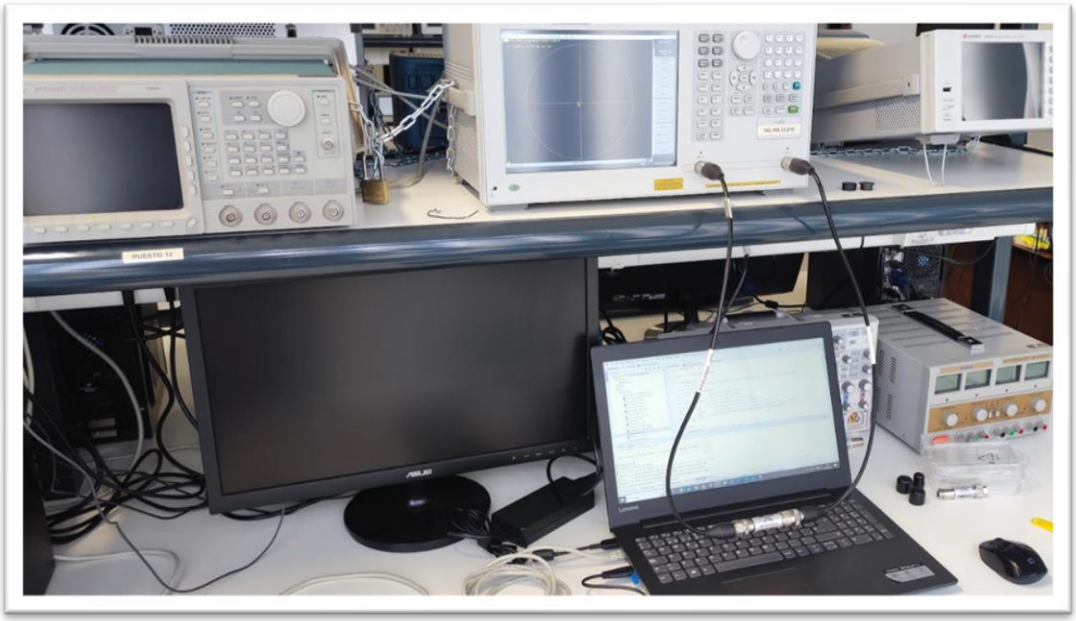
La posibilidad de medir la respuesta en frecuencia del tipo de filtro vendrá delimitada por el ajuste de los distintos parámetros que componen un barrido en frecuencia. Existe disponibilidad en el laboratorio de radiofrecuencia de filtros del tipo LP, HP, BP, los cuales son objeto principal de estudio en este caso. Consecuentemente se procede al desarrollo de la estructura principal de control directa sobre el equipo, la cual está compuesta por comandos SCPI los cuales son enviados mediante código Python. Esto es común para todos los casos de prueba, habiendo visto con anterioridad la estructura o sintaxis que compone el envío de un ajuste en el apartado de desglose representativo de sintaxis, en el capítulo 5. Tras el desarrollo de las funciones necesarias, se procede a implementar el script o test de Robot Framework con el fin de automatizar la toma de medida que ocupa este proceso en concreto. Dando por finalizada la prueba tras la generación de los ficheros de reporte e informes favorables que facilita Robot Framework tras su ejecución.



80. Diagrama de flujo jerárquico caso 6.

6.1.7 CASO 7. Proceso de Cálculo de Parámetros [S]

En este caso, se ha optado por crear un caso de prueba donde se compacte la toma de medidas de los cuatro parámetros de Scattering posibles. Concretamente esta tarea de prueba posibilita la automatización y a su vez la elección de qué parámetro se desea caracterizar para el DUT seleccionado. Los conceptos teóricos son compartidos a su vez con los casos de prueba 4, 5 y 8, este último se desarrolla en el siguiente subapartado. Enumerando brevemente, los conceptos o contenidos teóricos de notable importancia para la correcta comprensión tanto de procesos como de resultados obtenidos en este apartado son; los parámetros de dispersión o parámetros [S] y la existente relación mediante transformación bilineal de los parámetros [S] a impedancias y coeficientes de reflexión. Todos revisados en el capítulo 5, fundamentos teóricos.

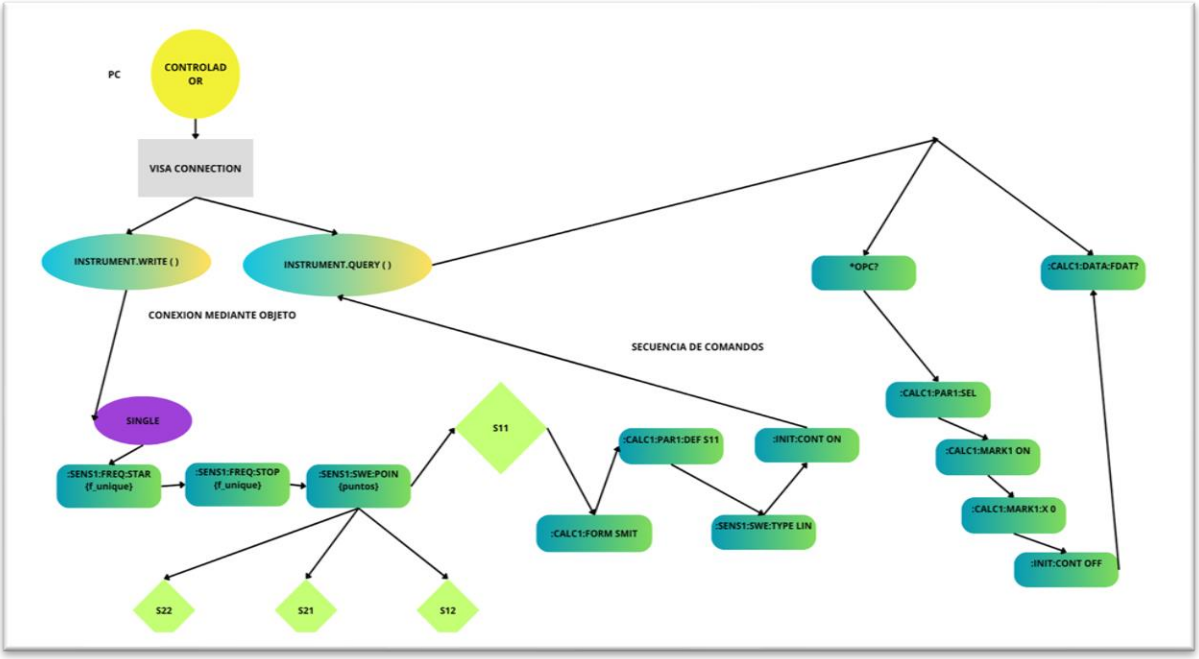


81. Cálculo de parámetros [S]. Caso 7.

Descriptivamente se procede a listar las fases y su contenido para el desarrollo de este apartado. Será necesario crear distintas opciones para poder abarcar el total de posibilidades que conlleva calcular los parámetros de Scattering para una única frecuencia o en un rango de frecuencias. Por lo que se crea una función donde dependiendo de la opción Single o Sweep, esta proceda por un camino u otro. Una vez se ha tomado la decisión del tipo de barrido, el siguiente paso es elegir que parámetro se desea medir. Para ello habrá que tener en cuenta que el parámetro S12 y S21 requerirán de una distinta configuración de conexionado. Donde habrá que físicamente conectar el bipuerto bajo prueba a los puertos 1 y 2.

El proceso seguido en este caso es análogo al resto de casos de prueba, con la excepción que en este caso se desarrolla mediante Python una función que otorga al usuario la capacidad de elección sobre que parámetro desea medir. Siendo esto un contratiempo debido a que, según que parámetro se deseé medir, existirá una configuración física y que requerirá presencialidad. De la misma manera, tras la fase de desarrollo del algoritmo de comunicación con el instrumento se aplica la capa de automatización mediante el diseño y desarrollo del test basado en Robot Framework. Cabe destacar, que la estructura seguida en los diferentes scripts de Robot hace uso de las distintas etapas genéricas que proporciona este framework. Las cuales son ***Settings***, ***Variables***, ***Test cases***, omitiendo el uso del apartado de ***Keywords***.

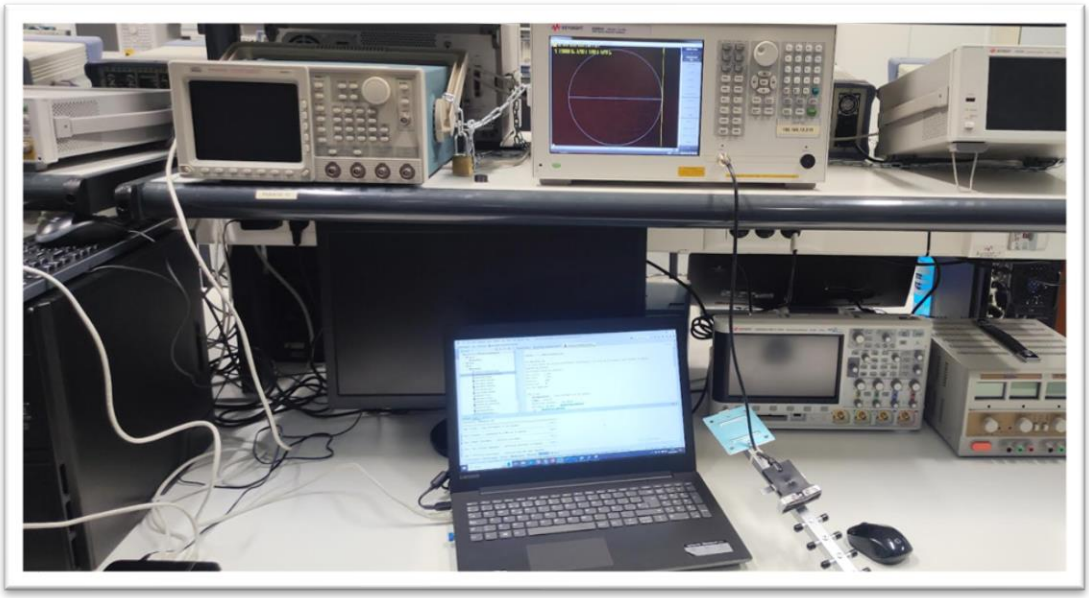
Ya que como se menciona con anterioridad en el contexto de este proyecto, la implementación equivalente de Keywords se realiza en un script externo el cual se ejecuta e importa como librería propia.



82. Diagrama de flujo jerárquico caso 7. Desarrollo del parámetro S11 en una única frecuencia.

6.1.8 CASO 8. Proceso de Caracterización de Impedancia

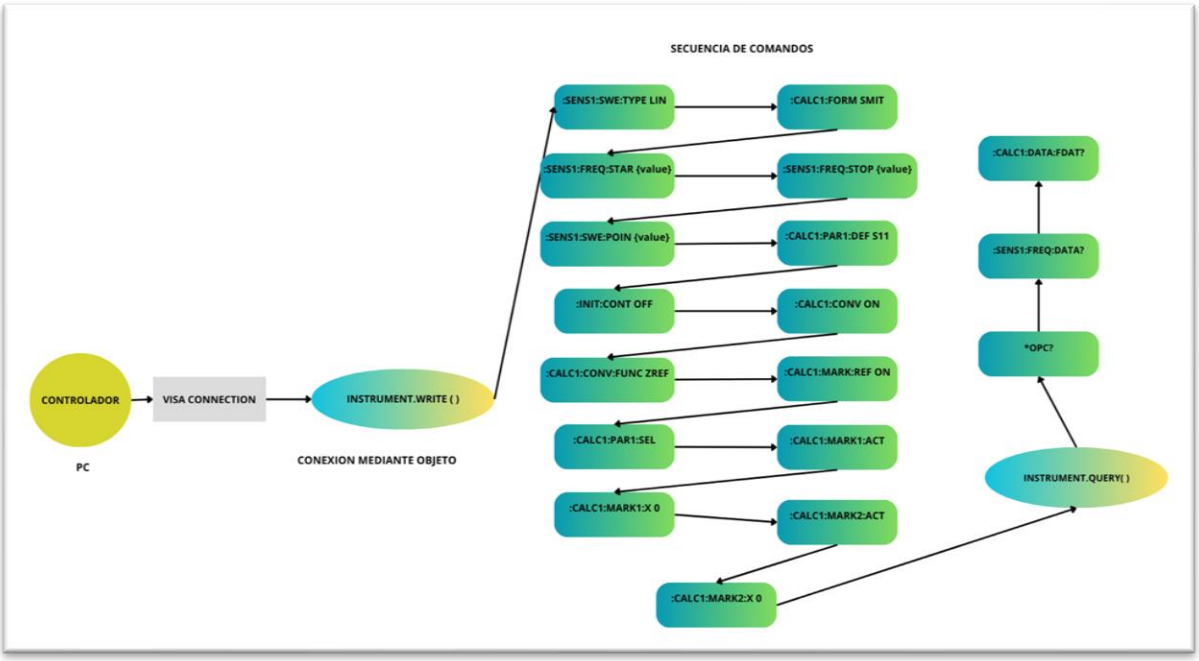
Apartado en el cual toma mayor importancia la relación vinculante entre el parámetro S11 e impedancia, la cual en este caso será impedancia de entrada. Para ello es relevante la comprensión de la transformación bilineal que existe entre el parámetro de Scattering y la impedancia como concepto genérico. Referido en el capítulo 5, apartado caracterización de impedancia, caso 8. Se considera oportuno mencionar que en el caso que comprende este procedimiento, al estar basado en la automatización, se opta por aplicar una conversión directa entre parámetros [S] e impedancia [Z], que el instrumento facilita. Tras buscar su equivalente en comando SCPI se procede a la aplicación directa de este con el fin de obtener el valor convertido.



83. Caracterizando impedancia de entrada de antena Yagi-Uda. Caso 8.

Para ello se crea una función la cual es nombrada (caracteriza_dut()), en la cual se desarrollará una serie de envíos de comandos SCPI para el correcto ajuste del instrumento de medida. Siendo algunos de estos comandos aquellos que ejecutan acciones tales como el tipo de barrido, formato de visualización, frecuencia inicial, frecuencia final, número de puntos, parámetro a medir (S11), un único barrido, encendido de modo conversión. Es en este último mencionado es donde se aplica la conversión entre parámetros de Scattering e impedancia de entrada del DUT. Otros comandos gestionarán opciones como encendido de marcadores, posición de marcadores y la solicitud de la medida realizada para poder posteriormente procesar dicha información de manera analítica.

El comando preciso para aplicar la conversión entre parámetro [S] e impedancia [Z] es :CALC1:MARK:REF ON. Tras el diseño de esta función el próximo paso es aplicar la capa de automatización, tal y como se ha realizado en todos los casos anteriores. Una vez esta capa es aplicada se da por concluido el desarrollo total de este caso de prueba. Siendo ya objeto de la fase de análisis y pruebas del proceso.



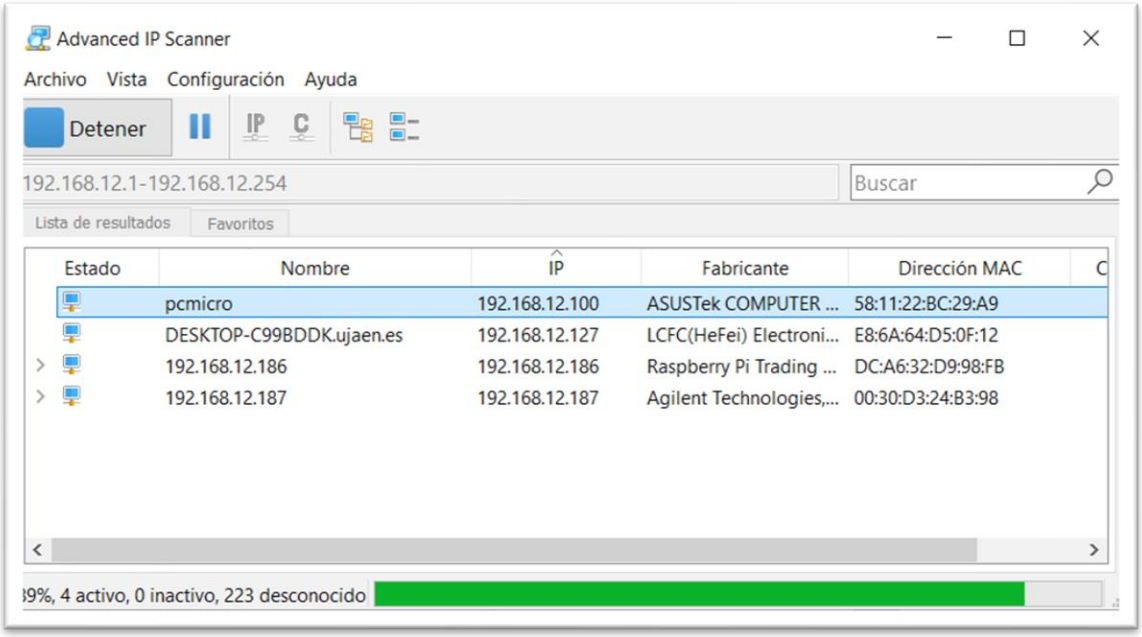
84. Diagrama de flujo jerárquico caso 8.

6.1.9 CASO 9. Transmisión de audio y Demodulación FM

Como último caso de prueba se procede a utilizar varios dispositivos con el fin de demonstrar que es posible la automatización simultanea de varios equipos de toma de medidas. Los instrumentos de laboratorio son en un principio el analizador de espectros N9320B. El cual se describe brevemente en el capítulo 4, bajo el apartado de recursos hardware. Aparte, se utilizará una Raspberry Pi para que esta sirva de transmisor FM. La principal idea es que el analizador de espectros haga de receptor de una señal radio transmitida a una frecuencia determinada y seguidamente la demodule, con el fin de reproducir la información que transporta. Reproducir, porque se toma la decisión que la señal modulada transporte un fichero de audio. De este modo ya que el analizador de espectros esta provisto por un módulo de salida de audio, el cual es accesible cuando se ajusta el modo de demodulación FM. Se podrá escuchar el fichero .wav que se transmite a la frecuencia deseada. Es condición obligatoria que el ajuste del SPAN tome valor 0 o equivalentemente “zero span”. De este único modo se puede acceder al menú de demodulación FM y optar por la reproducción de la información transportada por la señal modulada.

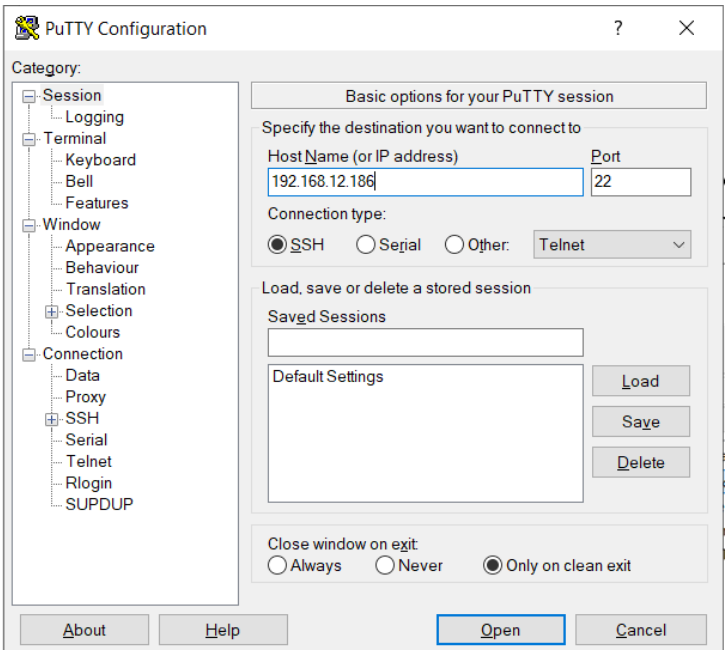
Para convertir el dispositivo Raspberry en transmisor FM solo hay que introducir un software de emisión de señales radio, habilitar una antena para transmitir mediante la conexión de un cable al pin correspondiente (GPIO 04) y ejecutar desde PuTTY los comandos necesarios para arrancar la transmisión.

Esta es la interfaz del software utilizado para el descubrimiento de direcciones IP, de especial interés es la dirección IP del dispositivo transmisor, Raspberry PI. Existe la posibilidad de interconectar mediante red Ethernet o conexión Wifi, en este caso se opta por la conexión LAN.



85. Interfaz IP Scanner.

Como se ha mencionado con anterioridad el software de control PuTTY es similar en cuanto a comportamiento, al de una terminal de usuario Windows como el CMD o PowerShell. Mediante el protocolo SSH se ejecutan comandos con el fin de interactuar con dispositivos periféricos. SSH consiste en un protocolo de red utilizado para conectar y gestionar dispositivos remotos con cierto nivel de seguridad mediante una conexión cifrada. En resumen, PuTTY mediante SSH interviene como pasarela de control para el dispositivo transmisor, el cual ya implementa el software de transmisión FM.



86. Interfaz de control PuTTY.

El procedimiento de configuración y control del dispositivo transmisor se cumple al entrar en el directorio del software de radio FM, el cual se consigue con la siguiente línea de comando.

```
pi@raspberrypi:~ $ cd /home/pi/PiFmRds/src
```

87. Cambio de directorio dentro de la Raspberry.

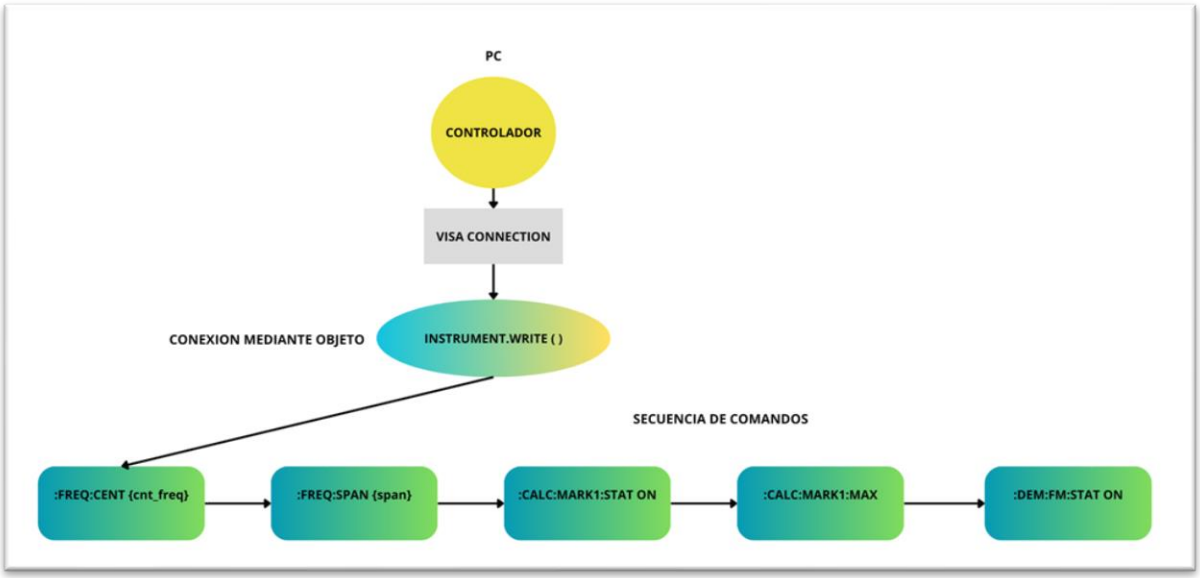
Es imprescindible ejecutar el siguiente comando para iniciar la transmisión. Previamente definiendo una frecuencia específica para la transmisión. En este caso de prueba en concreto se selecciona la frecuencia de 107.9 MHz la cual pertenece al rango de frecuencias de libre acceso.

```
i@raspberrypi:~/PiFmRds/src $ sudo ./pi_fm_rds -audio noct20.wav&
```

88. Inicio de transmisión FM.

89. Demodulación FM. Caso 9.

Por lo que el desarrollo de una función que permita demodular la señal transmitida es el siguiente paso. Para ello se crea dicha función la cual recibe el nombre de (demod_fm()) basada en lenguaje de programación Python. Esta función ha de contener y a su vez enviar los comandos SCPI que ajusten los parámetros requeridos en el analizador de espectros, para que el proceso de demodulación se cumpla correctamente.



90. Diagrama de flujo jerárquico caso 9.

Tras el diseño o desarrollo de esta función se puede dar por terminado el total de funciones que han sido parte de la fase de desarrollo de código Python. Es aquí donde se

consigue terminar la librería propia con los “keywords” que se han creado específicamente para la correcta implementación de este proyecto. A partir de aquí, solo queda implementar la capa de automatizado de Robot Framework con el fin de obtener información detallada de este último caso de prueba. Breve descripción del proceso, una vez comienza a emitir el dispositivo transmisor se ejecuta el script de Robot y automáticamente el analizador de espectros ajustará su configuración para así poder llevar a cabo el proceso de demodulación de la señal transmitida.

7 RESULTADOS

Es en este momento donde se exponen de manera gráfica y detallada los distintos resultados obtenidos tras la ejecución de todos los casos de prueba diseñados a lo largo de la duración de este proyecto. Con el fin de poder concretar que efectivamente el desarrollo y posterior implementación del proceso de automatización de medias es efectivo y en general útil en el entorno de procesos de laboratorio. Para exponer que los resultados obtenidos son satisfactorios se propone la visualización de los reportes e informes generados con Robot Framework al igual que imágenes de los ajustes realizados en los distintos equipos para probar que efectivamente el control remoto mediante comandos SCPI se lleva a cabo correctamente, concepto central del propósito de este proyecto.

7.1 Resultados Caso 1.

A continuación, se detallan los informes generados automáticamente mediante la aplicación de Robot Framework para la ejecución del caso de prueba 1. En primer lugar, se verifica la correcta ejecución de todos los pasos, seguido de un desglose concreto que muestra los valores numéricos necesarios. El propósito de esta sección de resultados es proporcionar la máxima información disponible.

Como parte del resultado se obtiene una gráfica de la señal generada a través del display del analizador de espectros, como se puede visualizar en la ilustración 94.

Main Genera Onda Log

Generated
20240126 18:22:39 UTC+01:00
2 days 17 hours ago

Test Statistics

Total Statistics	Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip
All Tests	9	9	0	0	00:00:14	

Statistics by Tag	Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip
step 00 Finding the IP address of the instrument	1	1	0	0	00:00:06	
step 01 Creating a connection with the instrument	1	1	0	0	00:00:00	
step 02 Apply a reset to the instrument	1	1	0	0	00:00:00	
step 03 Generating a signal	1	1	0	0	00:00:00	
step 04 selecting modulation state	1	1	0	0	00:00:00	
step 05 adjusting output state	1	1	0	0	00:00:00	
step 06 graph of signal	1	1	0	0	00:00:08	
step 07 Disconnecting output	1	1	0	0	00:00:00	
step 08 Ending connection with instrument	1	1	0	0	00:00:00	

Statistics by Suite	Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip
Main Genera Onda	9	9	0	0	00:00:15	

91. Log correspondiente al caso 1.

Main Genera Onda Report

Generated
20240126 18:22:39 UTC+01:00
2 days 17 hours ago

Summary Information

Status:	All tests passed
Start Time:	20240126 18:22:25.388
End Time:	20240126 18:22:39.899
Elapsed Time:	00:00:14.511
Log File:	log.html

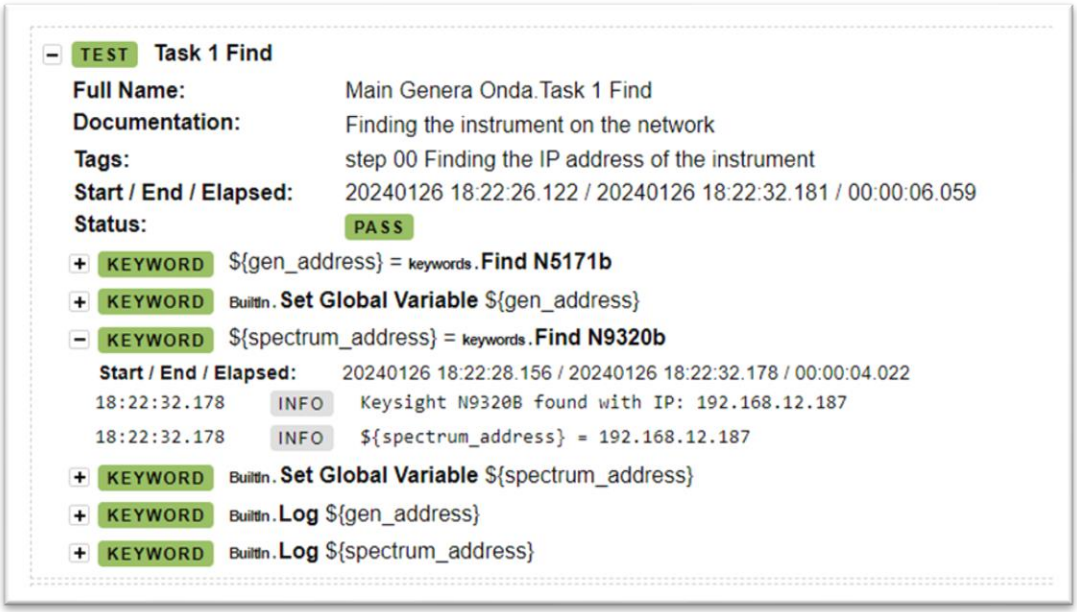
Test Statistics

Total Statistics	Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip
All Tests	9	9	0	0	00:00:14	

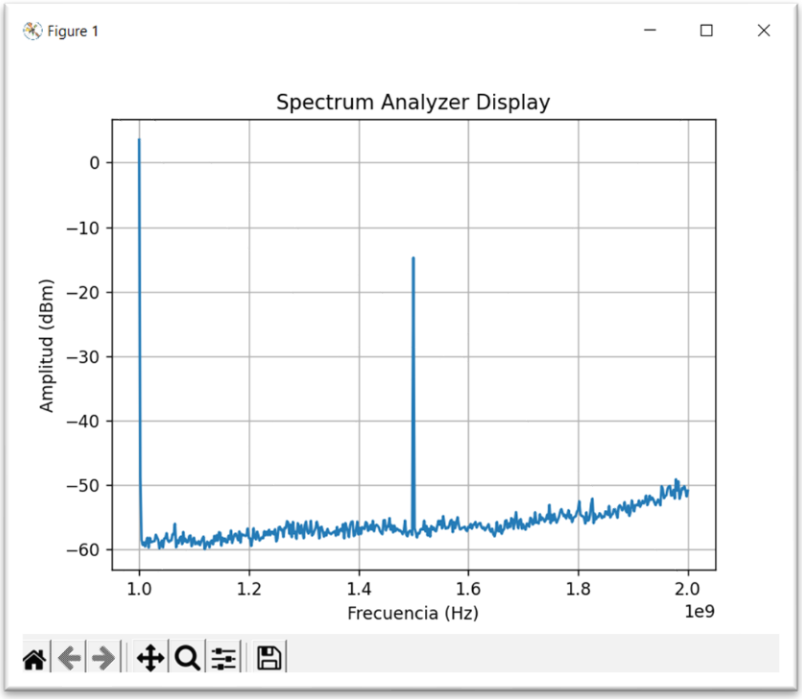
Statistics by Tag	Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip
step 00 Finding the IP address of the instrument	1	1	0	0	00:00:06	
step 01 Creating a connection with the instrument	1	1	0	0	00:00:00	
step 02 Apply a reset to the instrument	1	1	0	0	00:00:00	
step 03 Generating a signal	1	1	0	0	00:00:00	
step 04 selecting modulation state	1	1	0	0	00:00:00	
step 05 adjusting output state	1	1	0	0	00:00:00	
step 06 graph of signal	1	1	0	0	00:00:08	
step 07 Disconnecting output	1	1	0	0	00:00:00	
step 08 Ending connection with instrument	1	1	0	0	00:00:00	

Statistics by Suite	Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip
Main Genera Onda	9	9	0	0	00:00:15	

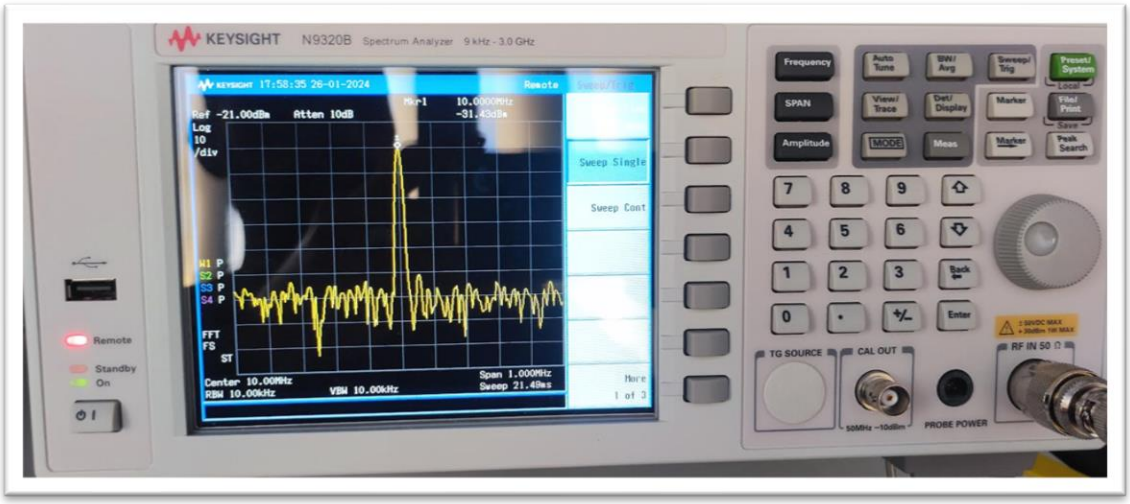
92. Reporte correspondiente al caso 1.



93. Tarea 1 completada obteniendo dirección IP.



94. Grafica resultante al ejecutar el script del caso 1.



95. Señal generada en el caso 1.

7.2 Resultados Caso 2.

Para el caso que ocupa el caso de prueba 2, el cual genera un barrido en frecuencia utilizando el generador de funciones se presentan los reportes generados y una imagen donde se comprueban los ajustes realizados de manera automática mediante Robot Framework y el script de funciones en Python.

Main Barrido Freq Log

Generated
20240207 15:36:10 UTC+01:00
38 seconds ago

Test Statistics

Total Statistics	Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip
All Tests	8	8	0	0	00:00:05	

Statistics by Tag	Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip
Step 00 this step finds intruments IP address	1	1	0	0	00:00:01	
Step 01 this step opens connection with instrument	1	1	0	0	00:00:00	
Step 02 this step applies a reset	1	1	0	0	00:00:00	
Step 03 this step sets Sweep parameters	1	1	0	0	00:00:04	
Step 04 this step selects modulation state	1	1	0	0	00:00:00	
Step 05 this step activates output	1	1	0	0	00:00:00	
Step 06 this step turns off output	1	1	0	0	00:00:00	
Step 07 this step ends connection with instrument	1	1	0	0	00:00:00	

Statistics by Suite	Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip
Main Barrido Freq	8	8	0	0	00:00:06	

96. Log correspondiente al caso 2.

Main Barrido Freq Report

Generated
20240207 15:36:10 UTC+01:00
2 minutes 18 seconds ago

Summary Information

Status:	All tests passed
Start Time:	20240207 15:36:03.916
End Time:	20240207 15:36:10.268
Elapsed Time:	00:00:06.352
Log File:	log.html

Test Statistics

Total Statistics	Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip
All Tests	8	8	0	0	00:00:05	

Statistics by Tag	Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip
Step 00 this step finds intruments IP address	1	1	0	0	00:00:01	
Step 01 this step opens connection with instrument	1	1	0	0	00:00:00	
Step 02 this step applies a reset	1	1	0	0	00:00:00	
Step 03 this step sets Sweep parameters	1	1	0	0	00:00:04	
Step 04 this step selects modulation state	1	1	0	0	00:00:00	
Step 05 this step activates output	1	1	0	0	00:00:00	
Step 06 this step turns off output	1	1	0	0	00:00:00	
Step 07 this step ends connection with instrument	1	1	0	0	00:00:00	

Statistics by Suite	Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip
Main Barrido Freq	8	8	0	0	00:00:06	

97. Report correspondiente al caso 2.

TEST **Task 4 Performing sweep in frequency**

Full Name:	Main Barrido Freq.Task 4 Performing sweep in frequency
Documentation:	performing frequency sweep
Tags:	Step 03
Start / End / Elapsed:	20240126 18:25:54.900 / 20240126 18:25:58.912 / 00:00:04.012
Status:	PASS

KEYWORD keywords . **Generator Barrido** ${instrument_generator}, ${start_freq}, ${stop_freq}, ${start_amplitude}, ${stop_amplitude}, ${N_puntos}

Start / End / Elapsed: 20240126 18:25:54.901 / 20240126 18:25:58.911 / 00:00:04.010

18:25:58.911 INFO Los valores ajustados son frecuencia inicio: 3e6,frecuencia final: 10e6, amplitud inicial: -3, amplitud final: -15, numero de puntos del barrido: 2000

98. Detalle de tarea 4 para el caso 2.

99. Parámetros recibidos por el generador de señal N5171B. Caso 2.

7.3 Resultados Caso 3.

En este apartado se procede a detallar los resultados de la modulación AM realizada en el caso de prueba 3. Se expone el detalle visual de los reportes generados con Robot Framework donde se puede comprobar que efectivamente se consigue la implementación de la modulación AM.

Main Modular Am Log

Generated
20240207 15:42:33 UTC+01:00
1 minute 6 seconds ago

Test Statistics

Total Statistics	Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip
All Tests	9	9	0	0	00:00:14	

Statistics by Tag	Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip
Step 00 this step finds the instrument´s IP address	1	1	0	0	00:00:01	
Step 01 this step opens a connection	1	1	0	0	00:00:00	
Step 02 this step applies a reset	1	1	0	0	00:00:00	
Step 03 this step performs main task	1	1	0	0	00:00:00	
Step 04 this step selects AM modulation	1	1	0	0	00:00:03	
Step 05 this step turns state of modulation	1	1	0	0	00:00:00	
Step 06 this step changes state of output	1	1	0	0	00:00:10	
Step 07 this step turns off output	1	1	0	0	00:00:00	
Step 08 this step ends connection with instrument	1	1	0	0	00:00:00	

Statistics by Suite	Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip
Main Modular Am	9	9	0	0	00:00:15	

100. Log correspondiente al caso 3.

Main Modular Am Report

Generated
20240207 15:42:33 UTC+01:00
2 minutes 30 seconds ago

Summary Information

Status:	All tests passed
Start Time:	20240207 15:42:18.538
End Time:	20240207 15:42:33.599
Elapsed Time:	00:00:15.061
Log File:	log.html

Test Statistics

Total Statistics	Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip
All Tests	9	9	0	0	00:00:14	

Statistics by Tag	Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip
Step 00 this step finds the instrument´s IP address	1	1	0	0	00:00:01	
Step 01 this step opens a connection	1	1	0	0	00:00:00	
Step 02 this step applies a reset	1	1	0	0	00:00:00	
Step 03 this step performs main task	1	1	0	0	00:00:00	
Step 04 this step selects AM modulation	1	1	0	0	00:00:03	
Step 05 this step turns state of modulation	1	1	0	0	00:00:00	
Step 06 this step changes state of output	1	1	0	0	00:00:10	
Step 07 this step turns off output	1	1	0	0	00:00:00	
Step 08 this step ends connection with instrument	1	1	0	0	00:00:00	

Statistics by Suite	Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip
Main Modular Am	9	9	0	0	00:00:15	

101. Report correspondiente al caso 3.

TEST **Task 4 Performing main task**

Full Name:	Main Modular Am.Task 4 Performing main task
Documentation:	generating a signal
Tags:	Step 03
Start / End / Elapsed:	20240126 18:30:31.237 / 20240126 18:30:31.238 / 00:00:00.001
Status:	PASS

KEYWORD keywords.**Genera Onda** ${instrument_generator}, ${frequency}, ${amplitude}

Start / End / Elapsed: 20240126 18:30:31.237 / 20240126 18:30:31.238 / 00:00:00.001

```
18:30:31.238   INFO   El valor de frecuencia es: 10e6
                      El valor de amplitud es: -10dBm´s
```

102. Detalle de tarea 4 para el caso 3.

7.4 Resultados Caso 4.

En esta tarea de prueba se consigue automatizar el proceso de cálculo del parámetro de Scattering S11 para una única frecuencia. Tras la exitosa ejecución, los ficheros generados son los siguientes.

parametro S11 Log

Generated
20240207 16:19:46 UTC+01:00
14 seconds ago

Test Statistics

Total Statistics	Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip
All Tests	7	7	0	0	00:00:15	

Statistics by Tag	Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip
Step 00 this step finds the instrument´s IP address	1	1	0	0	00:00:05	
Step 01 This step is connecting to the VNA via IP address	1	1	0	0	00:00:00	
Step 02 this step resets the instrument	1	1	0	0	00:00:00	
Step 03 this step sets impedance value	1	1	0	0	00:00:00	
Step 04 this step performs main task	1	1	0	0	00:00:10	
Step 05 this step turns on output	1	1	0	0	00:00:00	
Step 06 this step ends connection with instrument	1	1	0	0	00:00:00	

Statistics by Suite	Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip
parametro S11	7	7	0	0	00:00:16	

103. Log correspondiente al caso 4.

parametro S11 Report

Generated
20240207 16:19:46 UTC+01:00
3 minutes 23 seconds ago

Summary Information

Status:	All tests passed
Start Time:	20240207 16:19:30.022
End Time:	20240207 16:19:46.163
Elapsed Time:	00:00:16.141
Log File:	log.html

Test Statistics

Total Statistics	Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip
All Tests	7	7	0	0	00:00:15	

Statistics by Tag	Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip
Step 00 this step finds the instrument´s IP address	1	1	0	0	00:00:05	
Step 01 This step is connecting to the VNA via IP address	1	1	0	0	00:00:00	
Step 02 this step resets the instrument	1	1	0	0	00:00:00	
Step 03 this step sets impedance value	1	1	0	0	00:00:00	
Step 04 this step performs main task	1	1	0	0	00:00:10	
Step 05 this step turns on output	1	1	0	0	00:00:00	
Step 06 this step ends connection with instrument	1	1	0	0	00:00:00	

Statistics by Suite	Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip
parametro S11	7	7	0	0	00:00:16	

104. Report correspondiente al caso 4.

TEST **Task 5 Performing measurement**

Full Name:	parametro S11.Task 5 Performing measurement
Documentation:	measuring S11 parameter
Tags:	Step 04 this step performs main task
Start / End / Elapsed:	20240207 16:19:36.018 / 20240207 16:19:46.110 / 00:00:10.092
Status:	PASS

KEYWORD keywords.**Parametro S11 Sfreq** ${instrument_vectorial_analyzer}, ${freq_unique}

Start / End / Elapsed: 20240207 16:19:36.018 / 20240207 16:19:46.109 / 00:00:10.091

```
16:19:46.109   INFO   Parameter s11 is:  (63.2457480467-13.7312501221j)
                      Module: 64.7191770335681 phase: -12.24935298466569 degrees
```

105. Detalle de la tarea 5 del caso 4.

En la ilustración 105, se puede visualizar el valor numérico obtenido del parámetro S11, el cual es mostrado en formato modulo/fase y formato complejo con parte Re y Im.

7.5 Resultados Caso 5.

El caso 5 se evalúa del mismo modo que el caso 4, con la diferencia de que en este caso se pretende graficar el comportamiento del parámetro de Scattering S11 en un rango de frecuencias. Por lo que el script se diseña con el fin de que facilite una gráfica donde se pueda analizar la evolución del parámetro.

parametro S11 barrido Log

Generated
20240207 16:27:53 UTC+01:00
2 minutes 28 seconds ago

Test Statistics

Total Statistics	Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip
All Tests	7	7	0	0	00:01:16	

Statistics by Tag	Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip
Step 00 This step finds the IP address of the instrument	1	1	0	0	00:00:05	
Step 01 This step opens connection with the instrument	1	1	0	0	00:00:00	
Step 02 This step applies a reset to the instrument	1	1	0	0	00:00:00	
Step 03 This step sets the impedance value	1	1	0	0	00:00:00	
Step 04 This step performs the main task	1	1	0	0	00:01:11	
Step 05 This step turns on output	1	1	0	0	00:00:00	
Step 06 This step ends connection with the instrument	1	1	0	0	00:00:00	

Statistics by Suite	Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip
parametro S11 barrido	7	7	0	0	00:01:17	

106. Log correspondiente al caso 5.

parametro S11 barrido Report

Generated
20240207 16:27:53 UTC+01:00
3 minutes 58 seconds ago

Summary Information

Status:	All tests passed
Start Time:	20240207 16:26:36.543
End Time:	20240207 16:27:53.307
Elapsed Time:	00:01:16.764
Log File:	log.html

Test Statistics

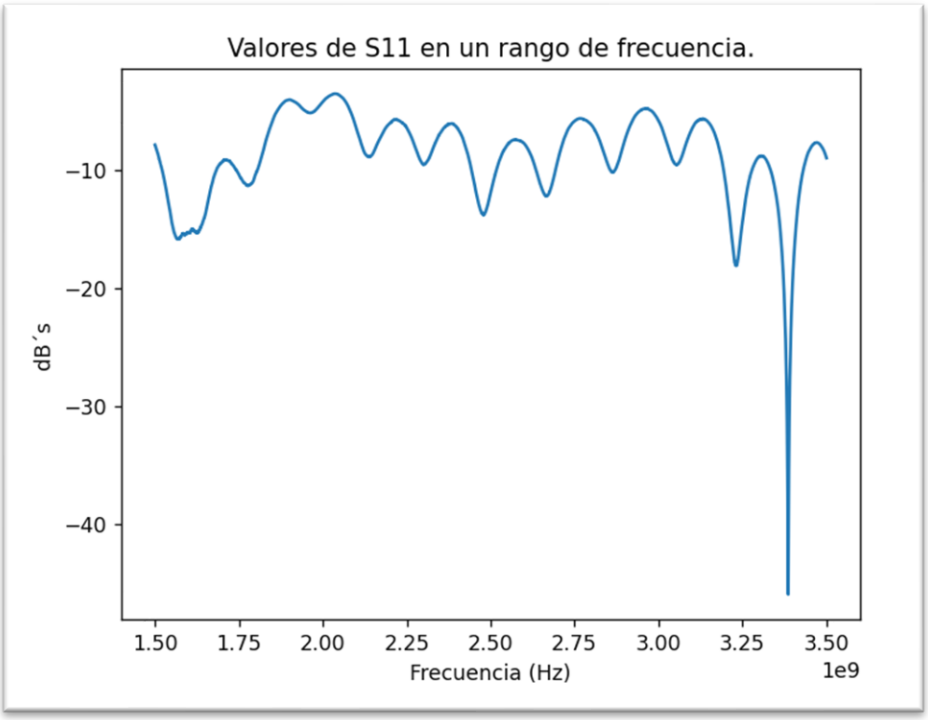
Total Statistics	Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip
All Tests	7	7	0	0	00:01:16	

Statistics by Tag	Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip
Step 00 This step finds the IP address of the instrument	1	1	0	0	00:00:05	
Step 01 This step opens connection with the instrument	1	1	0	0	00:00:00	
Step 02 This step applies a reset to the instrument	1	1	0	0	00:00:00	
Step 03 This step sets the impedance value	1	1	0	0	00:00:00	
Step 04 This step performs the main task	1	1	0	0	00:01:11	
Step 05 This step turns on output	1	1	0	0	00:00:00	
Step 06 This step ends connection with the instrument	1	1	0	0	00:00:00	

Statistics by Suite	Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip
parametro S11 barrido	7	7	0	0	00:01:17	

107. Report correspondiente al caso 5.

Es a partir de la ilustración 108 donde de manera analítica y aproximada, podemos visualizar la frecuencia de resonancia la cual se ajusta a la necesidad de un ancho de banda mayor. Por lo que se da por completada la meta perseguida de dar utilidad a la automatización de toma de medidas.



108. Grafica resultante del caso 5.

7.6 Resultados Caso 6.

Para este caso de prueba, se persigue graficar de manera concisa la respuesta en frecuencia de un DUT. En este caso se procede con la toma de medida para un filtro banda de paso, con una banda de paso desde los 63 MHz hasta los 77 MHz. Quedará demostrado gráficamente el comportamiento en frecuencia mediante las gráficas generadas.

Plot Res Freq Filtro Log

Generated
20240207 16:35:25 UTC+01:00
6 seconds ago

Test Statistics

Total Statistics	Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip
All Tests	6	6	0	0	00:00:32	

Statistics by Tag	Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip
Step 00 This step finds the instrument	1	1	0	0	00:00:05	
Step 01 This step opens a connection with the instrument	1	1	0	0	00:00:00	
Step 02 This step applies a reset to the instrument	1	1	0	0	00:00:00	
Step 03 This step sets the impedance value	1	1	0	0	00:00:00	
Step 04 This step performs the main task	1	1	0	0	00:00:27	
Step 05 This step ends connection with the instrument	1	1	0	0	00:00:00	

Statistics by Suite	Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip
Plot Res Freq Filtro	6	6	0	0	00:00:33	

109. Log correspondiente al caso 6.

Plot Res Freq Filtro Report

Generated
20240207 16:35:25 UTC+01:00
1 minute 33 seconds ago

Summary Information

Status:	All tests passed
Start Time:	20240207 16:34:52.305
End Time:	20240207 16:35:25.269
Elapsed Time:	00:00:32.964
Log File:	log.html

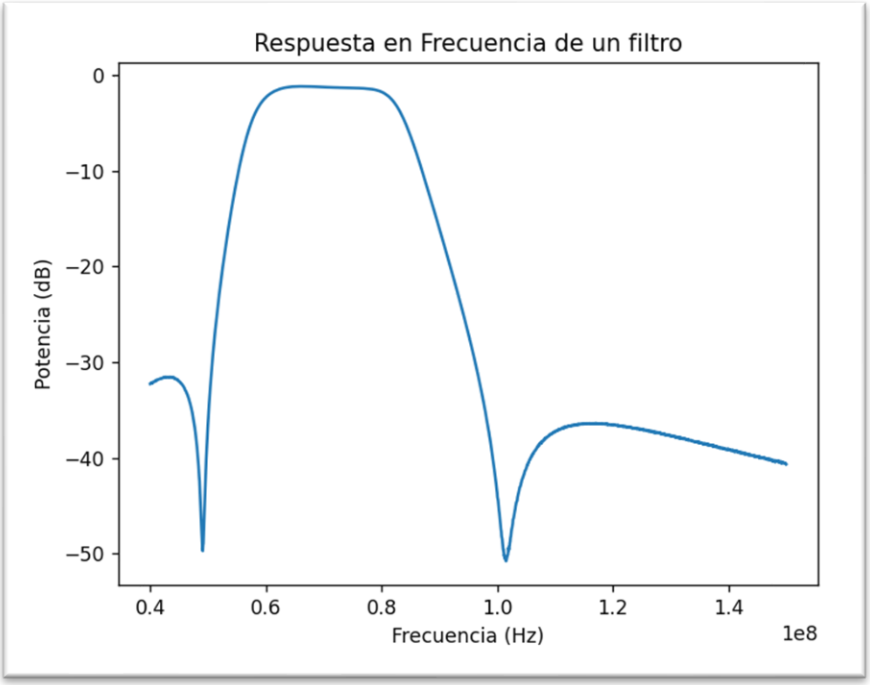
Test Statistics

Total Statistics	Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip
All Tests	6	6	0	0	00:00:32	

Statistics by Tag	Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip
Step 00 This step finds the instrument	1	1	0	0	00:00:05	
Step 01 This step opens a connection with the instrument	1	1	0	0	00:00:00	
Step 02 This step applies a reset to the instrument	1	1	0	0	00:00:00	
Step 03 This step sets the impedance value	1	1	0	0	00:00:00	
Step 04 This step performs the main task	1	1	0	0	00:00:27	
Step 05 This step ends connection with the instrument	1	1	0	0	00:00:00	

Statistics by Suite	Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip
Plot Res Freq Filtro	6	6	0	0	00:00:33	

110. Report correspondiente al caso 6.



111. Grafica resultante de un filtro BP caso 7.

Se consigue graficar el comportamiento frecuencial del DUT, dando cobertura al concepto de toma de medidas en remoto mediante la automatización del proceso. Obteniendo graficas las cuales pueden ser analizadas a distancia.

7.7 Resultados Caso 7.

En esta tarea de prueba, se incorpora la capacidad de automatizar la toma de medidas de los cuatro posibles parámetros de Scattering. La compactación de la medición de estos cuatro parámetros en una única función se presenta como un aspecto de gran interés. Es esencial considerar la presencialidad en el laboratorio durante este procedimiento, ya que la configuración de conexionado puede variar según el parámetro que se requiera medir. Esto se debe a que, mientras que los parámetros S11 y S22 son atributos de un único puerto, ya sea el puerto de salida o el de entrada. Existe la posibilidad de que la medida se base en los parámetros S12 o S21. En este último caso, se requerirá la conexión del DUT a ambos puertos disponibles en el Analizador de Red Vectorial (VNA).

parametros 4S Log

Generated
20240207 18:04:44 UTC+01:00
1 minute 0 seconds ago

Test Statistics

Total Statistics	Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip
All Tests	7	7	0	0	00:00:21	

Statistics by Tag	Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip
Step 00 finding the correct instrument to iniciate procedure	1	1	0	0	00:00:05	
Step 01 Opening connection with the instrument	1	1	0	0	00:00:00	
Step 02 Performing a reset to the instrument to start with no errors	1	1	0	0	00:00:00	
Step 03 This step sets the impedance value	1	1	0	0	00:00:00	
Step 04 In this step the parameter of scattering is obtained	1	1	0	0	00:00:16	
Step 05 In this step, the output is turned on	1	1	0	0	00:00:00	
Step 06 Ends the connection with the instrument	1	1	0	0	00:00:00	

Statistics by Suite	Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip
parametros 4S	7	7	0	0	00:00:22	

112. Log correspondiente al caso 7.

parametros 4S Report

Generated
20240207 18:04:44 UTC+01:00
2 minutes 15 seconds ago

Summary Information

Status:	All tests passed
Start Time:	20240207 18:04:22.120
End Time:	20240207 18:04:44.219
Elapsed Time:	00:00:22.099
Log File:	log.html

Test Statistics

Total Statistics	Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip
All Tests	7	7	0	0	00:00:21	

Statistics by Tag	Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip
Step 00 finding the correct instrument to iniciate procedure	1	1	0	0	00:00:05	
Step 01 Opening connection with the instrument	1	1	0	0	00:00:00	
Step 02 Performing a reset to the instrument to start with no errors	1	1	0	0	00:00:00	
Step 03 This step sets the impedance value	1	1	0	0	00:00:00	
Step 04 In this step the parameter of scattering is obtained	1	1	0	0	00:00:16	
Step 05 In this step, the output is turned on	1	1	0	0	00:00:00	
Step 06 Ends the connection with the instrument	1	1	0	0	00:00:00	

Statistics by Suite	Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip
parametros 4S	7	7	0	0	00:00:22	

113. Report correspondiente al caso 7.

TEST **Task 5 Performing measurement**

Full Name:	parametros 4S.Task 5 Performing measurement
Documentation:	measuring Scattering parameters
Tags:	Step 04 In this step the parameter of scattering is obtained
Start / End / Elapsed:	20240207 18:08:47.257 / 20240207 18:08:50.434 / 00:00:03.177
Status:	PASS

KEYWORD keywords.**Parametro Scattering Sfreq** ${instrument_vectorial_analyzer}, ${barrido}, ${parametro}, ${F_unica}, ${F_inicial}, ${F_final}, ${points}

Start / End / Elapsed: 20240207 18:08:47.257 / 20240207 18:08:50.433 / 00:00:03.176

```
18:08:50.433    INFO    Parameter S11 is (14.8610793806+12.0706267864j)
```

114. Detalle de la tarea 5 del caso 7.

7.8 Resultados Caso 8.

En este caso de prueba se pretende automatizar de manera eficiente el cálculo o medida de impedancia de entrada de un DUT mediante la conversión interna del instrumento de parámetro S11 a Impedancia de entrada. Con la ejecución de un único comando SCPI se consigue un valor de impedancia variable a lo largo de un rango de frecuencias predeterminado por el usuario.

Caracteriza Impedancia Dut Log

Generated
20240207 18:42:17 UTC+01:00
5 seconds ago

Test Statistics

Total Statistics	Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip
All Tests	6	6	0	0	00:01:01	

Statistics by Tag	Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip
Step 00 this step finds the instrument´s IP address	1	1	0	0	00:00:05	
Step 01 this step opens a connection with the instrument	1	1	0	0	00:00:00	
Step 02 this step applies a reset	1	1	0	0	00:00:00	
Step 03 this step sets impedance value	1	1	0	0	00:00:00	
Step 04 this step performs the main task	1	1	0	0	00:00:56	
Step 05 this step ends a connection with the instrument	1	1	0	0	00:00:00	

Statistics by Suite	Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip
Caracteriza Impedancia Dut	6	6	0	0	00:01:02	

115. Log correspondiente al caso 8.

Caracteriza Impedancia Dut Report

Generated
20240207 18:42:17 UTC+01:00
1 minute 11 seconds ago

Summary Information

Status:	All tests passed
Start Time:	20240207 18:41:14.992
End Time:	20240207 18:42:17.054
Elapsed Time:	00:01:02.062
Log File:	log.html

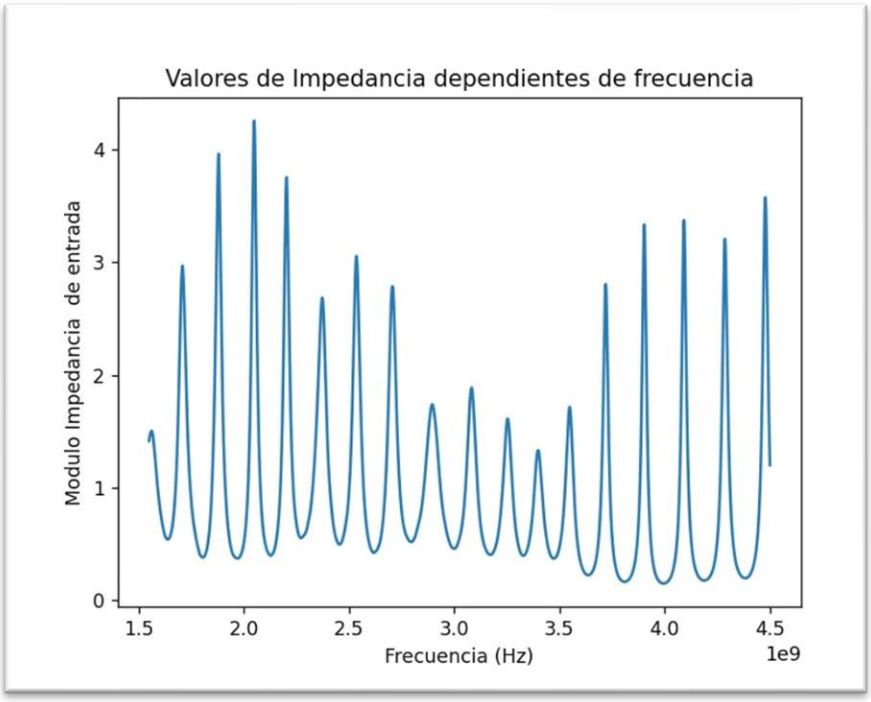
Test Statistics

Total Statistics	Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip
All Tests	6	6	0	0	00:01:01	

Statistics by Tag	Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip
Step 00 this step finds the instrument´s IP address	1	1	0	0	00:00:05	
Step 01 this step opens a connection with the instrument	1	1	0	0	00:00:00	
Step 02 this step applies a reset	1	1	0	0	00:00:00	
Step 03 this step sets impedance value	1	1	0	0	00:00:00	
Step 04 this step performs the main task	1	1	0	0	00:00:56	
Step 05 this step ends a connection with the instrument	1	1	0	0	00:00:00	

Statistics by Suite	Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip
Caracteriza Impedancia Dut	6	6	0	0	00:01:02	

116. Report correspondiente para el caso 8.



117. Grafica resultante de caso 8.

7.9 Resultados Caso 9.

Para este último caso, como se ha explicado en el capítulo 6, apartado de desarrollo para el caso de prueba 9 se procede a la transmisión de un fichero de audio mediante técnicas de modulación FM. Desde una Raspberry Pi y mediante el software PiFmRds se transmite a la frecuencia 107.9 MHz el audio con la idea de captarlo mediante el analizador de espectros N9320B y aplicar técnicas de demodulación FM. El proceso es exitoso y los resultados del proceso de automatización se muestran a continuación.

Tx Demod Log

Generated
20240207 18:51:51 UTC+01:00
28 seconds ago

Test Statistics

Total Statistics	Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip
All Tests	5	5	0	0	00:00:09	

Statistics by Tag	Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip
Step 00 finding the correct instrument to iniciate procedure	1	1	0	0	00:00:04	
Step 01 Opening connection with the instrument	1	1	0	0	00:00:00	
Step 02 Performing a reset to the instrument to start with no errors	1	1	0	0	00:00:00	
Step 03 In this step the demodulation is applied	1	1	0	0	00:00:05	
Step 04 Ends the connection with the instrument	1	1	0	0	00:00:00	

Statistics by Suite	Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip
Tx Demod	5	5	0	0	00:00:10	

118. Log correspondiente al caso 9.

TEST **Task 4 Performing main task**

Full Name: Tx Demod.Task 4 Performing main task
Documentation: Demodulating audio transmision FM
Tags: Step 03 In this step the demodulation is applied
Start / End / Elapsed: 20240207 18:51:46.023 / 20240207 18:51:51.068 / 00:00:05.045
Status: PASS

KEYWORD keywords.**Demod Fm** ${instrument_spectrum}, ${cnt_freq}, ${span}
Start / End / Elapsed: 20240207 18:51:46.024 / 20240207 18:51:51.068 / 00:00:05.044
18:51:51.068 INFO Demodulation centered at: 107.9e6 Hz

119. Detalle de la tarea 4 para el caso 9.

Tx Demod Report

Generated
20240207 18:51:51 UTC+01:00
2 minutes 59 seconds ago

Summary Information

Status:	All tests passed
Start Time:	20240207 18:51:41.125
End Time:	20240207 18:51:51.094
Elapsed Time:	00:00:09.969
Log File:	log.html

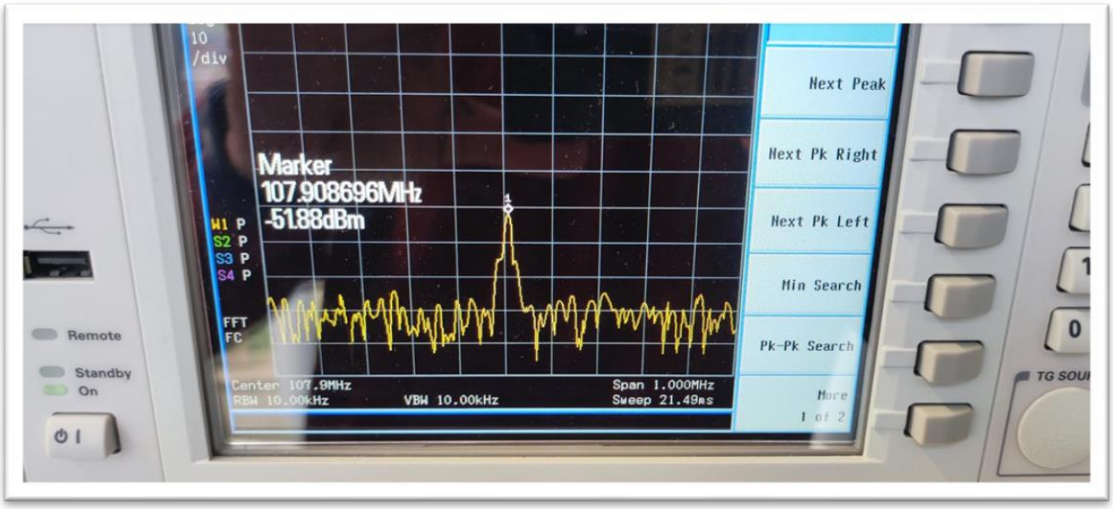
Test Statistics

Total Statistics	Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip
All Tests	5	5	0	0	00:00:09	

Statistics by Tag	Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip
Step 00 finding the correct instrument to iniciate procedure	1	1	0	0	00:00:04	
Step 01 Opening connection with the instrument	1	1	0	0	00:00:00	
Step 02 Performing a reset to the instrument to start with no errors	1	1	0	0	00:00:00	
Step 03 In this step the demodulation is applied	1	1	0	0	00:00:05	
Step 04 Ends the connection with the instrument	1	1	0	0	00:00:00	

Statistics by Suite	Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip
Tx Demod	5	5	0	0	00:00:10	

120. Report correspondiente al caso 9.



121. Detección de la señal transmitida en el analizador de espectros.

Ajustando el span dentro del script de Robot Framework, se puede visualizar el espectro de la señal modulada.

7.10 Resultados Caso ejemplo de error.

Para identificar informes no satisfactorios en caso de que alguna tarea de las que componen los tests no se ejecute correctamente, se detalla a continuación los posibles informes generados con el propósito de analizar situaciones de fallos en la ejecución.

parametros 4S Log

Generated
20240129 16:36:51 UTC+01:00
53 seconds ago

Test Statistics

Total Statistics	Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip
All Tests	7	0	7	0	00:00:06	

Statistics by Tag	Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip
Ends the connection with the instrument	1	0	1	0	00:00:00	
finding the correct instrument to iniciate procedure	1	0	1	0	00:00:06	
In this step, the output is turned on	1	0	1	0	00:00:00	
In this step the parameter of scattering is obtained	1	0	1	0	00:00:00	
Opening connection with the instrument	1	0	1	0	00:00:00	
Performing a reset to the instrument to start with no errors	1	0	1	0	00:00:00	
Step 00	1	0	1	0	00:00:06	
Step 01	1	0	1	0	00:00:00	
Step 02	1	0	1	0	00:00:00	
Step 03	2	0	2	0	00:00:00	
Step 04	1	0	1	0	00:00:00	
Step 05	1	0	1	0	00:00:00	

Statistics by Suite	Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip
parametros 4S	7	0	7	0	00:00:07	

122. Ejemplo de Log correspondiente a un test fallido.

parametros 4S Report

Generated
20240129 16:36:51 UTC+01:00
2 minutes 8 seconds ago

Summary Information

Status:	7 tests failed
Start Time:	20240129 16:36:44.595
End Time:	20240129 16:36:51.620
Elapsed Time:	00:00:07.025
Log File:	log.html

Test Statistics

Total Statistics	Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip
All Tests	7	0	7	0	00:00:06	

Statistics by Tag	Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip
Ends the connection with the instrument	1	0	1	0	00:00:00	
finding the correct instrument to iniciate procedure	1	0	1	0	00:00:06	
In this step, the output is turned on	1	0	1	0	00:00:00	
In this step the parameter of scattering is obtained	1	0	1	0	00:00:00	
Opening connection with the instrument	1	0	1	0	00:00:00	
Performing a reset to the instrument to start with no errors	1	0	1	0	00:00:00	
Step 00	1	0	1	0	00:00:06	
Step 01	1	0	1	0	00:00:00	
Step 02	1	0	1	0	00:00:00	
Step 03	2	0	2	0	00:00:00	
Step 04	1	0	1	0	00:00:00	
Step 05	1	0	1	0	00:00:00	

Statistics by Suite	Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip
parametros 4S	7	0	7	0	00:00:07	

123. Ejemplo de Report correspondiente a un test fallido.

TEST **Task 1 Find**

Full Name:	parametros 4S.Task 1 Find
Documentation:	Find instruments on the network
Tags:	finding the correct instrument to iniciate procedure, Step 00
Start / End / Elapsed:	20240129 16:36:45.377 / 20240129 16:36:51.420 / 00:00:06.043
Status:	FAIL
Message:	Keysight E5063a not found.

KEYWORD ${vectorial_address} = keywords.**Find E5063a**

Start / End / Elapsed: 20240129 16:36:45.377 / 20240129 16:36:51.420 / 00:00:06.043

16:36:51.420 **FAIL** Keysight E5063a not found.

KEYWORD BuiltIn.**Set Global Variable** ${vectorial_address}

KEYWORD BuiltIn.**Log** ${vectorial_address}

124. Tarea 1 fallida debido a error de conexión con instrumento E5063A (VNA).

En la última imagen, se observa un error de conexión con el instrumento debido a que no está conectado a la red LAN. Como consecuencia, aparece un mensaje indicando que el equipo no se encuentra dentro de la red. Debido a este fallo inicial, las demás tareas no se ejecutan, provocando un error completo en la realización del caso de pruebas. Esta demonstración cumple con el objetivo de demonstrar que a través de la automatización de casos de pruebe se consigue determinar en qué paso o tarea ocurren eventos de interés a la hora de realizar una tarea completa.

7.11 Análisis de tráfico mediante capturas con Wireshark

Con el fin de comprobar que existe una comunicación entre equipo controlador e instrumentos de medida, se procede a detallar y analizar el tráfico capturado con el software Wireshark. Estas capturas se han realizado durante el proceso de ejecución de los distintos casos de prueba que a continuación se exponen. En el breve análisis realizado sobre las capturas se pretende localizar los distintos envíos y recepciones de información entre el equipo controlador y los instrumentos que ocupan el caso de prueba. El equipo controlador el cual estará alojado en la dirección IP: 192.168.12.127 es el encargado de iniciar la sesión de intercambio de información y datos. Mientras que el instrumento de medidas devolverá datos a su vez desde una dirección IP la cual será diferente, dependiendo del caso de prueba y del instrumento involucrado en esta.

Para la primera captura se utiliza el caso de prueba número 1. Este caso de prueba se basa en la generación de una onda o señal. La captura de tráfico es filtrada con las direcciones IP de los instrumentos involucrados con el fin de obtener información de manera más rápida y clara. Si se aumenta el grado de exactitud del filtrado podemos discriminar con mayor acierto información irrelevante.

125. Filtro de búsqueda en Wireshark por dirección IP y protocolo.

El contenido de la captura ya filtrada muestra como la comunicación se lleva a cabo entre el equipo controlador y el instrumento de medida. Si se analiza en detalle cualquier paquete capturado se puede visualizar el contenido de información intercambiada entre ambos extremos.

365	20.153410	192.168.12.127	192.168.12.123	TCP	54	57128 → 111 [FIN, ACK] Seq=61 Ack
366	20.153578	192.168.12.123	192.168.12.127	TCP	60	111 → 57128 [ACK] Seq=33 Ack=62 W
367	20.153894	192.168.12.123	192.168.12.127	TCP	60	111 → 57128 [FIN, ACK] Seq=33 Ack
368	20.153931	192.168.12.127	192.168.12.123	TCP	54	57128 → 111 [ACK] Seq=62 Ack=34 W
369	20.155144	192.168.12.127	192.168.12.123	TCP	66	57129 → 49152 [SYN] Seq=0 Win=642
370	20.155267	150.214.170.15	192.168.12.127	DNS	118	Standard query response 0x96b1 A
371	20.155415	192.168.12.123	192.168.12.127	TCP	66	49152 → 57129 [SYN, ACK] Seq=0 Ac
372	20.155484	192.168.12.127	192.168.12.123	TCP	54	57129 → 49152 [ACK] Seq=1 Ack=1 W
373	20.156417	192.168.12.127	192.168.12.123	VXI-11…	122	V1 CREATE_LINK Call (Reply In 374
374	20.157122	192.168.12.123	192.168.12.127	VXI-11…	98	V1 CREATE_LINK Reply (Call In 373
375	20.173021	192.168.12.127	192.168.12.187	TCP	66	57130 → 111 [SYN] Seq=0 Win=64240
376	20.173711	192.168.12.187	192.168.12.127	TCP	62	111 → 57130 [SYN, ACK] Seq=0 Ack=

126. Parte del tráfico de paquetes capturado en el caso 1.

Si se selecciona un paquete se puede ver el contenido de este a un nivel más específico. En este caso se visualiza que el paquete contiene los comandos SCPI específicos del caso de prueba. En este caso un ajuste de la frecuencia, nivel de señal en dBm´s y ajustes de salida RF.

Wireshark · Seguir secuencia TCP (tcp.stream eq 6) · captura_main_genera_onda.pcapng

```
...@.......................
...............................inst0......
(......................................D..K.....................................................*RST
..... ..K...............................L..<.....................................................
:FREQ 1.5e9
...... ..<..............................
...h..4.....................................................*:POWer:LEVel:IMMediate:AMPLitude -10 dbm
&.... ..4...........................*...L..K.....................................................:OUTP:MOD ON
MM... ..K...............................H..~.....................................................:OUTPut ON
... ..~.............................
```

127. Información contenida en paquete o trama 373.

Como segundo ejemplo se utiliza el caso de prueba 9, con el cual se consigue la transmisión de una señal aplicando técnicas de modulación FM. Con el fin de demodular la señal y escuchar el fichero de audio transmitido en el instrumento N9320B analizador de espectros. El tráfico capturado muestra como el equipo controlador envía los ajustes necesarios al instrumento N9320B, analizador de espectros. Siendo algunos de estos ajustes, la frecuencia central, valor del span y el ajuste a modo demodulador FM.

441	8.653099	192.168.12.127	192.168.12.187	VXI-11…	122	V1 CREATE_LINK Call (Reply In 442) inst0
442	8.657703	192.168.12.187	192.168.12.127	VXI-11…	98	V1 CREATE_LINK Reply (Call In 441) No Error LID=-1041247812
443	8.663603	192.168.12.127	192.168.12.187	VXI-11…	126	V1 DEVICE_WRITE Call (Reply In 444) LID=-1041247812 *RST\r\n
444	8.665424	192.168.12.187	192.168.12.127	VXI-11…	90	V1 DEVICE_WRITE Reply (Call In 443) No Error
445	8.668023	192.168.12.127	192.168.12.187	VXI-11…	138	V1 DEVICE_WRITE Call (Reply In 446) LID=-1041247812 :FREQ:CENT 107.9e6\r\n
446	8.672173	192.168.12.187	192.168.12.127	VXI-11…	90	V1 DEVICE_WRITE Reply (Call In 445) No Error
447	8.723555	192.168.12.127	192.168.12.187	TCP	54	58109 → 1024 [ACK] Seq=225 Ack=117 Win=2102272 Len=0
465	9.674823	192.168.12.127	192.168.12.187	VXI-11…	134	V1 DEVICE_WRITE Call (Reply In 466) LID=-1041247812 :FREQ:SPAN 0\r\n
466	9.676306	192.168.12.187	192.168.12.127	VXI-11…	90	V1 DEVICE_WRITE Reply (Call In 465) No Error
467	9.721714	192.168.12.127	192.168.12.187	TCP	54	58109 → 1024 [ACK] Seq=305 Ack=153 Win=2102016 Len=0
483	10.678080	192.168.12.127	192.168.12.187	VXI-11…	142	V1 DEVICE_WRITE Call (Reply In 484) LID=-1041247812 :CALC:MARK1:STAT ON\r\n
484	10.679842	192.168.12.187	192.168.12.127	VXI-11…	90	V1 DEVICE_WRITE Reply (Call In 483) No Error
485	10.724546	192.168.12.127	192.168.12.187	TCP	54	58109 → 1024 [ACK] Seq=393 Ack=189 Win=2102016 Len=0
601	11.691193	192.168.12.127	192.168.12.187	VXI-11…	138	V1 DEVICE_WRITE Call (Reply In 602) LID=-1041247812 :CALC:MARK1:MAX\r\n

128. Parte del tráfico capturado en el caso 9.

```
...@..$3..................
.................................inst0......(..
$3...................................n...D..d.......................................................*RST
..... ..d...............................P..HP.......................................................:FREQ:CENT 107.9e6
... ..HP...........................L..h.............................................................:FREQ:SPAN 0
.9... ..h...............................T..7.........................................................:CALC:MARK1:STAT ON
...... ..7..............................P..K.........................................................:CALC:MARK1:MAX
ON
... ..K..............................P.............................................................:DEM:FM:STAT ON
ON
... ..............................,.............................................................
```

129. Información contenida en el paquete o trama 441.

8 CONCLUSIONES

Se ha alcanzado la justificación de que este proyecto cumple con los objetivos perseguidos desde un inicio y demuestra de manera clara la viabilidad del concepto aplicado a la automatización de medidas. Siendo altamente gratificante poder demonstrar que mediante la implementación de scripts basados en Python en conjunto con el uso de comandos SCPI, se consigue el control completo de equipos e instrumentación de laboratorio. Corroborando a su vez la posibilidad de futuras líneas de trabajo al incorporar otras tecnologías de conexionado. A su vez siendo todo esto indispensable previo a la automatización de dichos procesos. Haciendo que aumente la escalabilidad y el grado de aplicación del enfoque principal que ocupa este proyecto.

A su vez, se menciona que, debido al trabajo de campo realizado durante este proyecto, se ha adquirido una visión más clara e intuitiva de los distintos componentes que forman el conjunto de este proyecto. Siendo algunos de los principales el uso de comandos SCPI, el cual guarda estrecha relación con el estándar IEEE 488.2, la plataforma de trabajo Robot Framework, cuya principal potencialidad es la de compactar procesos de manera ordenada y controlada con el fin de aplicar la automatización de procesos. El lenguaje de

programación Python, el cual brinda versatilidad, adquiriendo por ello una amplia aceptación en diversos campos. Todo esto adicionando al conocimiento ya adquirido durante el grado, nuevas herramientas de gran importancia para futuras aplicaciones, tras la inminente incorporación al mundo laboral.

Un análisis de los procesos y resultados adquiridos durante las distintas fases que componen el proyecto, prueban que tanto los procesos de desarrollo, implementación y pruebas han sido exitosos. Obteniendo resultados numéricos y gráficos que aplican conceptos teóricos de materias de estudio tales como radiocomunicaciones y microondas. Pero es necesario entender que este conjunto es aplicable a otros ámbitos, siempre y cuando sea posible aplicar el concepto de automatización a los procesos.

A pesar de que, durante el desarrollo de este documento se ha nombrado escasamente el concepto de conexión en remoto, este puede dar pie a futuras líneas de desarrollo de las que se expondrá de manera más extensa en el siguiente apartado.

9 LINEAS FUTURAS

Una de las principales inspiraciones que impulsan las líneas futuras de desarrollo, presentadas a continuación, vienen marcadas por la pandemia global de COVID-19 en 2020. Durante este período, se paralizaron las actividades que requerían salir de casa y relacionarse con otras personas, incluyendo la docencia presencial. Esta situación resalta la importancia de implementar la automatización y la conectividad remota. Es entonces cuando surgen ideas a partir de necesidades que podrían cambiar la forma en que realizamos actividades prácticas en asignaturas, posiblemente permitiendo que algunas de ellas se lleven a cabo sin la necesidad de que docentes o estudiantes estén presentes físicamente. Las posibles soluciones se presentan a continuación como parte de las futuras direcciones que se podrían aplicar al desarrollo central del proyecto.

Como propuesta para futuras líneas de desarrollo, se contemplan diversas opciones que pueden ser exploradas siguiendo la línea de este proyecto. Una de estas alternativas consiste en la expansión del concepto de conexión remota. Esta ampliación tiene como objetivo llevar el control de la instrumentación de equipos de medición a un nivel más avanzado. Se busca emplear una tecnología de conexión diferente entre el sistema de

control y los instrumentos de medición con el propósito de elevar el grado de alcance y beneficiar la conexión remota.

Durante el desarrollo de este proyecto, se ha implementado la tecnología Ethernet. Sin embargo, como propuesta para futuras mejoras, se plantea la posibilidad de extender la operatividad de los scripts de control y automatización más allá de una red LAN específica. Esto implica la capacidad de acceder y controlar los instrumentos desde una red de área local distinta. A modo de contextualización, en el laboratorio de radiofrecuencia se cuenta con un switch que podría actuar como puerta de enlace desde otra red, facilitando así una conexión remota en un sentido más amplio.

Una alternativa adicional para el desarrollo futuro podría considerar la habilitación de una conexión remota con uno de los ordenadores previamente instalados en el laboratorio. Esto permitiría el acceso desde cualquier ubicación geográfica, ya que el equipo controlador formaría parte de la misma red local LAN. Después de instalar el paquete de scripts desarrollados durante este proyecto, sería factible llevar a cabo tareas o medidas mediante una conexión remota. Incluso, al implementar modificaciones en el código, sería posible desarrollar nuevos procesos de medición. Nótese que a pesar de ser muy similares ambas propuestas citadas, a su vez difieren en el modo de entrada o acceso al laboratorio.

Como propuesta final, considerando que la tecnología Wifi implementa el protocolo TCP/IP, el cual a su vez es empleado por la tecnología Ethernet, la última propuesta de este apartado se centra en la adaptación de los instrumentos de laboratorio a la tecnología Wifi. El objetivo es permitir su utilización desde cualquier punto dentro de las instalaciones del Campus Científico Tecnológico de Linares. Persiguiendo como ultima meta hacer más accesible el uso y control de equipos de medidas.

10 ABREVIATURAS

A continuación, se procede a detallar una lista de abreviaturas incluidas en el contexto de este documento con el fin de facilitar la comprensión al lector.

- LAN– Local Area Network.
- SCPI – Standard Commands for Programmable Instruments.
- HPIB – Hewlett Packard Interface Bus.
- GPIB – General Purpose Interface Bus.
- TIC – Tecnologías de la Información y las Comunicaciones.
- DUT – Device Under Test.
- IEC – International Electronic Commission.
- IEEE – Institute of Electrical and Electronics Engineers.
- VISA – Virtual Instrument Software Architecture.
- USB – Universal Serial Bus.
- VXI – Virtualization Experience Infrastructure (CISCO).
- RPA – Robotic Process Automation.
- IDE - Integrated development environment.
- AM – Amplitude modulation.
- FM – Frequency modulation.
- PM – Phase modulation.
- SSH – Secure Shell.
- DMA – Direct Memory Access.
- PWM - Pulse Width Modulation.
- VHF – Very High Frequency.
- CA – Corriente Alterna.
- VNA – Vectorial Network Analyzer.
- BNC – Bayonet Neill–Concelman (tipo de conectores nombrado con los apellidos de sus inventores).
- TCP – Transmission Control Protocol.
- IP – Internet Protocol.
- LP – Low pass filter.

- HP – High pass filter.
- BP – Band pass filter.
- CMD – Command (abreviatura de comando en inglés).
- SSH – Secure Shell.

11 BIBLIOGRAFÍA

[1] «National Instruments web archive,» 20 agosto 2006. [En línea]. Available: https://web.archive.org/web/20060820090937/http://zone.ni.com/devzone/conceptd.nsf/webmain/1DF3ADFEBDFB9F77862567C300782C11.

[2] «The hp 9845 project,» 2010. [En línea]. Available: https://www.hp9845.net/9845/tutorials/hpib/.

[3] «IEEE488,» 27 mayo 2009. [En línea]. Available: http://www3.fi.mdp.edu.ar/mediciones/apuntes/IEEE488.pdf.

[4] «Chroma Systems Solutions,» 2024. [En línea]. Available: https://shop.chromausa.com/products/gpib-interface-19055-19056-19057-19032p.

[5] «Keysight,» 2000. [En línea]. Available: https://www.keysight.com/us/en/assets/9018-07931/reference-guides/9018-07931.pdf.

[6] «Sage Journals,» 14 agosto 2012. [En línea]. Available: https://journals.sagepub.com/doi/full/10.1177/2211068212455631.

[7] J. M. Pieper, «hit.bme.hu,» 20 diciembre 2012. [En línea]. Available: https://www.hit.bme.hu/~papay/edu/Lab/SCPIandVISA.pdf.

[8] «RAZORBILL INSTRUMENTS,» 2024. [En línea]. Available: https://razorbillinstruments.com/part-2-ivi-visa-and-scripting-to-communicate-with-lab-instruments/.

[9] «navjodh,» 24 marzo 2015. [En línea]. Available: http://navjodh.com/test-and-measurement/virtual-instrument-software-architecture-visa-basics-and-interoperability/.

[10] «JazzTeam,» 2024. [En línea]. Available: https://jazzteam.org/technical-articles/robot-framework-overview/.

[11] S. Bisht, Robot Framework Test Automation, Packt Publishing Ltd, 2013.

[12] «Robot Framework User Guide,» 2017. [En línea]. Available: https://robotframework.org/robotframework/latest/RobotFrameworkUserGuide.html.

[13] «Github,» 2019. [En línea]. Available: https://github.com/ChristopheJacquet/PiFmRds.

[14] «Redeszone,» 2019. [En línea]. Available: https://www.redeszone.net/2019/03/01/wireshark-3-disponible/.

[15] «Keysight,» 1 enero 2021. [En línea]. Available: https://www.keysight.com/us/en/assets/9018-03690/programming-guides/9018-03690.pdf.

[16] «Keysight,» 5 diciembre 2018. [En línea]. Available: https://www.keysight.com/us/en/assets/9018-02757/programming-guides/9018-02757.pdf.

[17] «Keysight,» 21 enero 2004. [En línea]. Available: https://www.keysight.com/us/en/assets/9018-04907/programming-guides/9018-04907.pdf.

[18] «keysight,» 1992. [En línea]. Available: https://rfmw.em.keysight.com/spdhelpfiles/truevolt/webhelp/US/Content/__I_SCPI/IEEE-488_Common_Commands.htm.

[19] «Robot Framework User Guide versión 7.0,» 2016. [En línea]. Available: https://robotframework.org/robotframework/latest/RobotFrameworkUserGuide.html#high-level-architecture.

[20] «ieee.li/pdf,» agosto 2006. [En línea]. Available: https://www.ieee.li/pdf/essay/scattering_parameters_concept_theory_applications.pdf.

[21] «Wikipedia,» 14 noviembre 2019. [En línea]. Available: https://es.wikipedia.org/wiki/Par%C3%A1metros_de_dispersi%C3%B3n.

[22] D. M. Pozar, Microwave Engineering, Third Edition,, John Wiley & Sons Inc., 1998.

[23] «Avcontractor,» 2004. [En línea]. Available: https://avcontractor.com.mx/pdf/FiltrosActivos_Ing.FedericoMiraya.pdf.

[24] F. Miraya, «avcontractor,» 2004. [En línea]. Available:

https://avcontractor.com.mx/pdf/FiltrosActivos_Ing.FedericoMiraya.pdf.

[25] M. A. P, Principios de Electrónica, McGraw-Hill, 1996.

[26] «Agilent,» 1999. [En línea]. Available: http://www.sss-mag.com/pdf/AN154.pdf.

[27] «Inet educación,» 2020. [En línea]. Available: https://www.inet.edu.ar/wp-content/uploads/2020/07/ELECTRONICA_Gu--a06-Impedancia.pdf.

[29] «Keysight,» 2008. [En línea]. Available: https://www.keysight.com/us/en/assets/9018-04907/programming-guides/9018-04907.pdf.