

Trabajo Fin de Grado  
Grado en Ingeniería de las Tecnologías de  
Telecomunicación

Automatización de fuente de alimentación modular  
sobre SCPI

Autor: Omar Akoudad Muñoz

Tutores: Juan Antonio Becerra González, María José Madero Ayora

Dpto. Teoría de la Señal y Comunicaciones  
Escuela Técnica Superior de Ingeniería  
Universidad de Sevilla

Sevilla, 2022





Trabajo Fin de Grado  
Grado en Ingeniería de las Tecnologías de Telecomunicación

# **Automatización de fuente de alimentación modular sobre SCPI**

Autor:

Omar Akoudad Muñoz

Tutores:

Juan Antonio Becerra González

Profesor Ayudante Doctor

María José Madero Ayora

Profesora Titular

Dpto. de Teoría de la Señal y Comunicaciones

Escuela Técnica Superior de Ingeniería

Universidad de Sevilla

Sevilla, 2022



Trabajo Fin de Grado: Automatización de fuente de alimentación modular sobre SCPI

Autor: Omar Akoudad Muñoz

Tutores: Juan Antonio Becerra González, María José Madero Ayora

El tribunal nombrado para juzgar el Proyecto arriba indicado, compuesto por los siguientes miembros:

Presidente:

Vocales:

Secretario:

Acuerdan otorgarle la calificación de:

El Secretario del Tribunal

Fecha:



# Agradecimientos

---

En primer lugar, me gustaría agradecer a mi familia, por haberme permitido llegar tan lejos y haberme alentado cada minuto de esta bonita etapa.

A mis amigos, por su apoyo incondicional y máxima confianza desde que comencé la Universidad, en especial a María, medio Grado es tuyo.

A mis compañeros, pues sin ellos no habría tenido la fuerza para levantarme de muchas de las caídas que he sufrido durante estos años, en especial agradecer a Carmelo, Ayuso, Álvaro, Carlos, Juanjo y Rubén, gracias a todos por hacerme estos años un poco más amenos.

Por último, agradecer a todos los profesores de la Escuela, ya que de todos he aprendido algo y me han permitido adquirir los conocimientos que me han conducido a alcanzar el sueño que he tenido desde niño, convertirme en ingeniero.



# Resumen

---

En este trabajo, se ha llevado a cabo la configuración y puesta en marcha de la fuente de alimentación N6700 de Keysight para ejecutar de forma automatizada diferentes tipos de pruebas y experimentos que nos permitan obtener todo tipo de datos para su posterior análisis. Para ello, se han diseñado y desarrollado una serie de códigos o programas en la plataforma de programación y cálculo numérico MATLAB, que nos ha permitido conseguir los objetivos propuestos en este Trabajo Fin de Grado.



# Abstract

---

In this work, we have carried out the configuration and start-up of the Keysight N6700 power supply to automatically run different types of tests and experiments that allow us to obtain all types of data for subsequent analysis. To this end, a series of codes or programs have been designed and developed in the MATLAB programming and numerical calculation platform, which has allowed us to achieve the objectives proposed in this Final Degree Project.



# Índice

---

<b>Agradecimientos</b>	<b>vii</b>
<b>Resumen</b>	<b>ix</b>
<b>Abstract</b>	<b>xi</b>
<b>Índice</b>	<b>xiii</b>
<b>1 Introducción</b>	<b>1</b>
1.1 <i>Objetivos y alcance</i>	1
1.2 <i>Organización y estructura de la memoria</i>	2
<b>2 Fuentes de alimentación</b>	<b>3</b>
2.1 <i>Tipos de fuentes de alimentación</i>	5
<b>3 Lenguaje SCPI</b>	<b>7</b>
3.1 <i>Jerarquía de comandos SCPI</i>	8
3.2 <i>Sintaxis SCPI</i>	9
3.3 <i>Tipos de datos</i>	10
3.4 <i>Comandos comunes</i>	11
<b>4 Fuente de alimentación N6700</b>	<b>13</b>
4.1 <i>Conexiones de la N6700</i>	14
4.1.1 <i>General-Purpose Instrumentation Bus (GPIB)</i>	14
4.1.2 <i>Universal Serial Bus (USB)</i>	16
4.1.3 <i>Ethernet</i>	16
4.2 <i>Configuración de la N6700</i>	19
4.3 <i>Módulos N6774A y N6744B</i>	21
<b>5 Desarrollo experimental</b>	<b>23</b>
5.1 <i>Conexión con el instrumento</i>	23
5.2 <i>Habilitación y deshabilitación de salidas</i>	24
5.3 <i>Reinicio de la fuente de alimentación</i>	26
5.4 <i>Prueba básica sobre magnitudes</i>	26
5.5 <i>Monitorización del consumo</i>	27
<b>6 Resultados experimentales</b>	<b>29</b>
6.1 <i>Pruebas con resistencias</i>	29
6.2 <i>Monitorización del consumo del amplificador ZX60-V62+</i>	31
6.3 <i>Monitorización del consumo en placa Arduino, módulo SIM800C</i>	35
<b>7 Conclusiones y líneas futuras de trabajo</b>	<b>39</b>
<b>Datasheet del amplificador ZX60-V62+.</b>	<b>40</b>
<b>Índice de Figuras</b>	<b>45</b>

<b>Índice de Tablas</b>	<b>47</b>
<b>Índice de Códigos</b>	<b>48</b>
<b>Bibliografía</b>	<b>49</b>

# 1 INTRODUCCIÓN

---

*Lo que no se define no se puede medir. Lo que no se mide, no se puede mejorar. Lo que no se mejora, se degrada siempre.*

*- Lord Kelvin -*

Desde las medidas más primitivas como el pie, el palmo, el codo o la pulgada, hasta el mantenimiento predictivo, pasando por la digitalización y la instrumentación virtual, el ser humano ha tenido siempre la necesidad de medir el mundo que le rodea.

Muchos han sido los avances en el campo de la instrumentación electrónica desde el desarrollo de las primeras fuentes de alimentación hasta los sofisticados aparatos de medición que nos podemos encontrar hoy en día.

Las mejoras en este campo en los próximos años serán cruciales para un correcto desarrollo de nuevas tecnologías en los diferentes sectores industriales, energéticos y de telecomunicaciones entre otros. Una de estas mejoras son las fuentes de alimentación modulares, las cuales nos permiten una mayor versatilidad que las limitadas fuentes tradicionales a la hora de alimentar, medir y caracterizar todo tipo de dispositivos bajo pruebas (DUT's, por sus siglas en inglés).

Además del desarrollo de la instrumentación electrónica, durante el siglo XX también se sentaron las bases de la programación y la automatización de procesos, dos aspectos fundamentales sin los cuales no podríamos entender la sociedad en la que vivimos hoy. La necesidad de diseñar y desarrollar programas informáticos con el objetivo de poder automatizar tareas sin ayuda del humano surgió de la necesidad de optimizar el tiempo de desarrollo y homologación de diferentes sistemas electrónicos. Uno de los lenguajes de programación más usados en el campo de la instrumentación es el lenguaje SCPI, el cual tiene un papel clave en la consecución de este trabajo.

## 1.1 Objetivos y alcance

El objetivo principal de este Trabajo de Fin de Grado es el de desarrollar y aplicar a la fuente de alimentación modular N6700 de Keysight un código programado en MATLAB para automatizar un conjunto de tareas básicas, como aplicar tensiones y corrientes a diferentes dispositivos, limitar corrientes y monitorizar los consumos de diferentes dispositivos como resistencias, amplificadores y placas electrónicas de diferentes propósitos.

Para ello, debemos entender el funcionamiento de las fuentes de alimentación modulares, detallaremos sus ventajas e inconvenientes, se realizará el montaje de sus módulos y la configuraremos correctamente.

## 1.2 Organización y estructura de la memoria

Este Trabajo Fin de Grado se puede subdividir en dos partes bien diferenciadas. Por un lado, tenemos la parte teórica, desde el Capítulo 2 hasta el Capítulo 4, que contendrá todos los recursos necesarios para que el lector pueda comprender sin dificultades la parte práctica, expuesta en los capítulos 5 y 6, que consistirá en explicar el código desarrollado y los experimentos realizados en el Laboratorio de Radiocomunicación de la Escuela Técnica Superior de Ingeniería en la Universidad de Sevilla.

Por último, en el Capítulo 7 se hablará de las conclusiones y líneas futuras de investigación para este proyecto.

- **Capítulo 2: Fuentes de alimentación.** Se hará una introducción de cómo están constituidas las fuentes de alimentación, así como se hablará de los diferentes tipos existentes y su evolución.
- **Capítulo 3: Lenguaje SCPI.** Se introduce el lenguaje SCPI, se detallarán aspectos como la sintaxis, los tipos de datos y la jerarquía de comandos, de cara a poder tener un conocimiento suficiente para afrontar el resto de los capítulos donde se aplica este lenguaje.
- **Capítulo 4: Fuente de alimentación N6700.** En este capítulo se presentará la fuente de alimentación modular N6700 de Keysight, piedra angular del proyecto. Veremos cómo configurarla, se hará una descripción de las diferentes conexiones que proporciona y hablaremos de su característica más importante, los módulos.
- **Capítulo 5: Desarrollo experimental.** Se estudiarán los códigos que se han diseñado para realizar las pruebas experimentales.
- **Capítulo 6: Resultados experimentales.** Se realizarán múltiples pruebas en las que podremos verificar el correcto funcionamiento de los códigos desarrollados en el capítulo anterior. Mediante estas pruebas, se analizará el consumo de distintos dispositivos electrónicos como amplificadores y módems de telefonía móvil.
- **Capítulo 7: Conclusiones y líneas futuras de trabajo.** Se exponen las conclusiones a las que se han llegado tras la elaboración de este trabajo y se proponen pruebas a realizar en el futuro en base al mismo.

## 2 FUENTES DE ALIMENTACIÓN

Una fuente de alimentación no es más que un dispositivo que convierte una señal de tensión alterna en una tensión continua, si es monocanal, o bien en diferentes tensiones continuas por distintas salidas, si es multicanal. El objetivo de esta conversión se debe a la necesidad de alimentar distintos aparatos electrónicos tanto de uso doméstico como televisores y ordenadores, como de uso industrial, existiendo casos de uso en prácticamente todos los sectores: investigación, telecomunicaciones, energía y otras aplicaciones generales. En concreto, el tipo de fuente de alimentación en la que nos centraremos en este Trabajo de Fin de Grado, serán las destinadas a usarse en entornos de pruebas de laboratorio. Con estas fuentes tenemos la capacidad de ver y modificar en tiempo real la tensión y corriente aplicada al circuito/carga alimentados, a través del típico display LCD.



Figura 2.1 Fuente de alimentación clásica. <https://kolbi.es>

Para conseguir esta conversión y poder proporcionar una señal de tensión/corriente estable, las fuentes de alimentación se apoyan en varias etapas bien diferenciadas: transformación, rectificación, filtrado y regulación.

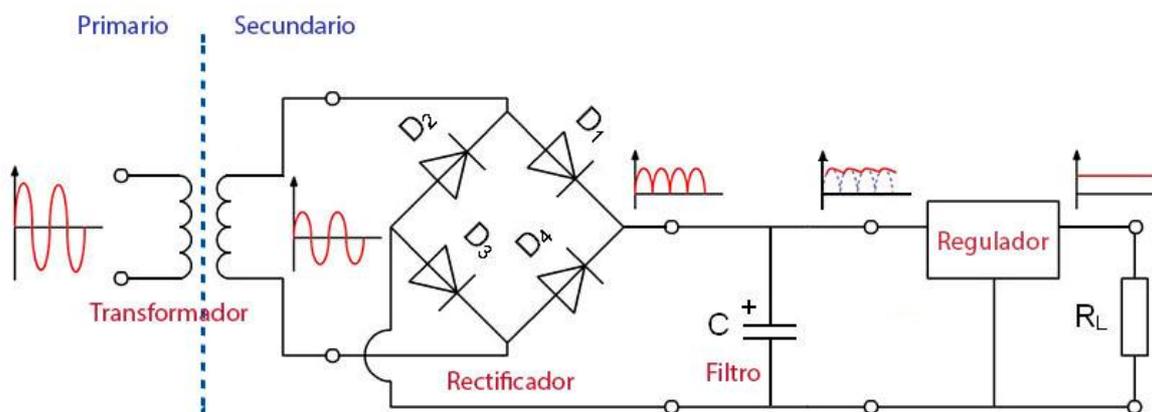


Figura 2.2 Etapas de una fuente de alimentación. <https://www.zonamaker.com>

- **Etapas de transformación:** En esta etapa contamos con el elemento fundamental, el transformador. Este dispositivo tiene el objetivo de reducir o aumentar un nivel de tensión determinado. Su estructura

está basada en un bobinado primario y otro, o varios, bobinados secundarios.

$$\frac{V_s}{V_p} = \frac{N_s}{N_p}$$

Con esta sencilla fórmula se puede determinar el número de vueltas que necesitaremos en los devanados para conseguir una determinada tensión.

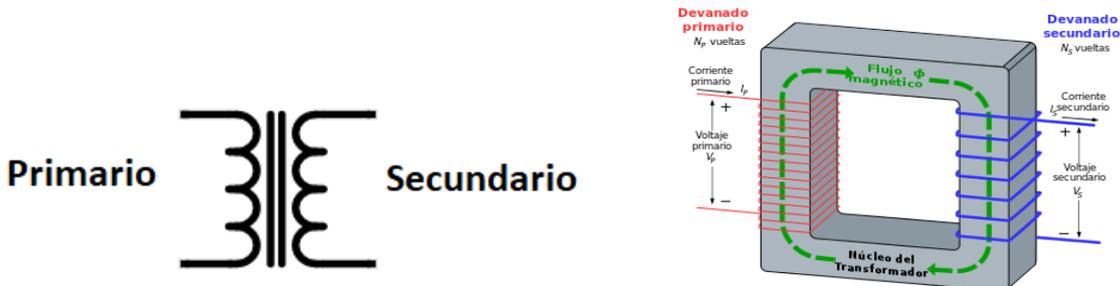


Figura 2.2 Símbolo y representación gráfica de un transformador.

<https://es.wikipedia.org/wiki/Transformador>

- **Etapa de rectificación:** Está compuesta principalmente por diodos que se encargan de rectificar la onda, es decir, esta es la etapa en la que la señal pierde su característica alterna y pasa tener solo valores positivos, quedando así una tensión continua a la salida.

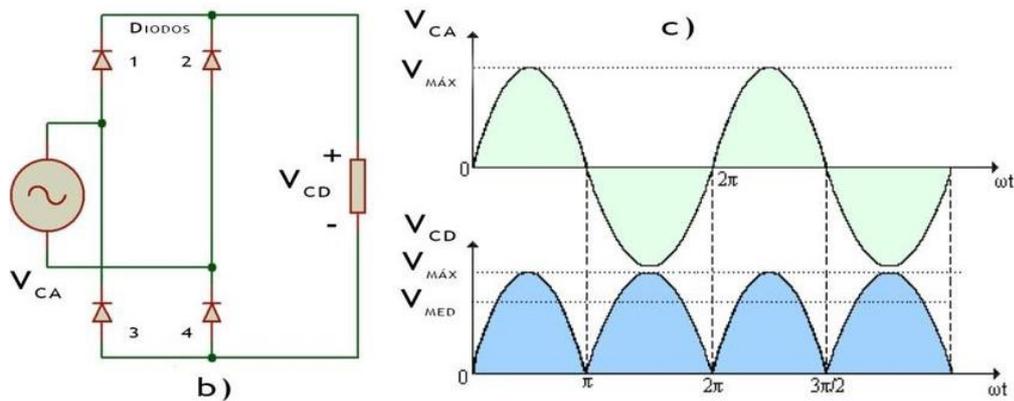


Figura 2.3 Rectificador de onda completa. <https://www.researchgate.net>

- **Etapa de filtrado:** Esta etapa queda constituida por uno o varios condensadores que se utilizan para obtener una tensión continua lo más plana posible a la salida del rectificador. Esto se logra gracias a las fases de carga y descarga de los condensadores. Por limitaciones físicas, sabemos que no se pueden obtener tensiones continuas puras debido a que, al conectar cualquier carga a la fuente de alimentación, ésta absorberá una intensidad que ha de suministrar el condensador del filtro, lo que trae como consecuencia la aparición de un rizado sobre el valor medio de la tensión continua. Cuanto mayor sea este rizado, peor será el funcionamiento de la fuente de alimentación, pues introducirá más zumbido y ruido de fondo en la tensión continua con la que se alimenta el circuito electrónico. A continuación, podemos observar la fórmula de la carga del condensador con la que obtenemos la tensión en cualquier instante de tiempo.

$$V_c(t) = V_e + (1 - e^{-t/RC})$$

Siendo C, la capacidad del condensador y R, la resistencia de la carga conectada.

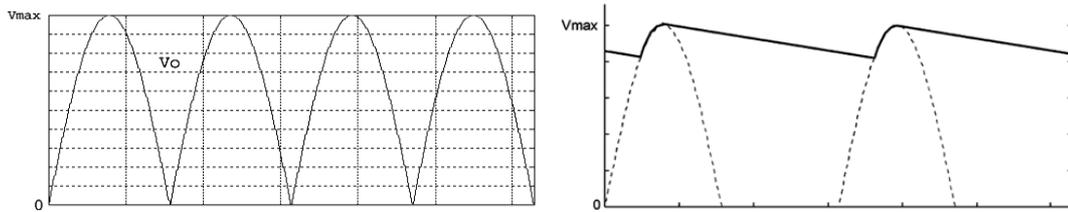


Figura 2.4 Salida del rectificador (izq.) vs Salida del rectificador filtrada (dcha.). <http://platea.pntic.mec.es>

- **Etapa de regulación:** Se compone de circuitos integrados que tienen la función de eliminar o reducir el rizado obtenido en la tensión de salida del filtro, con el objetivo de proporcionar una tensión a la entrada del circuito electrónico con la mayor precisión posible. Estos dispositivos se componen de una entrada, una salida y una conexión a tierra.

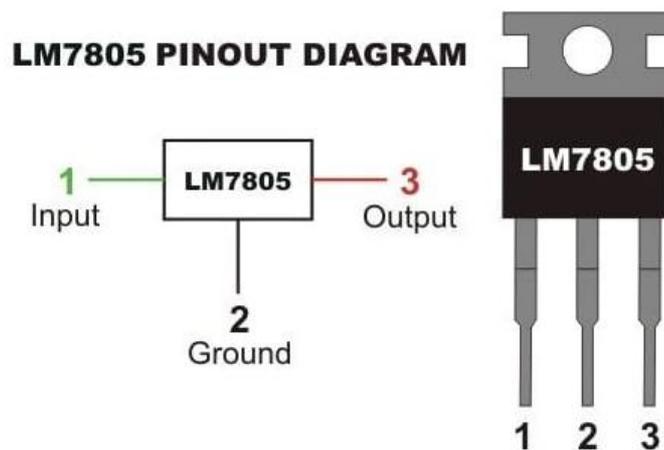


Figura 2.5 Regulador de tensión LM7805. <https://www.hwlibre.com/lm7805/>

## 2.1 Tipos de fuentes de alimentación

Las fuentes de alimentación pueden ser clasificadas atendiendo a diferentes criterios. Así, podemos tener:

- **Fuentes de alimentación de un solo canal y fuentes de alimentación multicanal:** En el caso de que tengamos un solo canal, solo podremos conectar un receptor a la fuente de alimentación, mientras que, si trabajamos con fuentes multicanal, conseguiremos conectar más de un dispositivo al mismo tiempo.
- **Fuentes de alimentación programables y no programables:** Esta clasificación se basa en la posibilidad o no de controlar e influir en los parámetros durante el funcionamiento de la fuente. Las fuentes de alimentación programables nos permiten poder realizar cambios en los parámetros de potencia o introducir una secuencia de acciones a ejecutar en tiempo real y de forma remota, mientras que las fuentes de alimentación no programables solo ofrecen la posibilidad de configurar la tensión y la corriente de forma manual.
- **Fuentes de alimentación conmutadas y fuentes de alimentación lineales:** Esta división se basa en la forma en que se convierte la corriente alterna en corriente continua. Las fuentes de alimentación lineales contienen dispositivos transformadores clásicos, bastante grandes y pesados, mientras que las fuentes de alimentación conmutadas, también conocidas como SMPS (Switch Mode Power Supply) son mucho más ligeras, ya que se consigue que el transformador sea más pequeño. Para ello, someten la tensión a dos procesos de rectificación, primero se convierte de alterna a continua, luego de continua a alterna, pero con una frecuencia diferente (entre 20 kHz y 100 kHz) y finalmente vuelve a transformarse en continua. Esto hace que se reduzcan las pérdidas y se consiga minimizar el tamaño del transformador.

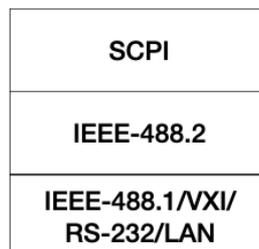


# 3 LENGUAJE SCPI

En este capítulo vamos a explicar en qué consiste el lenguaje SCPI (Standard Commands for Programmable Instruments), detallaremos su estructura, sintaxis, el porqué de su necesidad y utilidad, así como se pondrán ejemplos de algunos de los comandos más relevantes.

El lenguaje SCPI se define para conseguir una estandarización de los comandos de control y el formato de los datos de los instrumentos, con el objetivo de usarse en pruebas de control programables y medidas de dispositivos electrónicos.

El estándar SCPI fue definido en 1991 como una capa adicional a la norma IEEE 488.2 de 1987, “Standard Codes, Formats, Protocols, and Common Commands”, la cual a su vez mejoraba a la norma previa de 1975, IEEE-488.1. El principal objetivo de este estándar es poder reducir el tiempo de desarrollo en el diseño de pruebas. Esto se consigue gracias a un entorno de programación para instrumentos de control y utilización de datos basado en: mensajes del programa, respuestas del instrumento y formatos de datos en todos los instrumentos SCPI, independientemente del fabricante. Estos mensajes son enviados desde un controlador hacia un instrumento usando interfaces como IEEE 488.1, RS-232C o LAN entre otras. Las respuestas del instrumento pueden ser datos, como magnitudes físicas, o bien información sobre el estado de este.



**Figura 3.1** Torre de protocolos simplificada para el control de sistemas de instrumentación.

Antes de entrar en detalle con el lenguaje SCPI, se introduce al lector en cómo están formadas y cómo han ido evolucionando las diferentes capas de la instrumentación virtual. Los instrumentos virtuales representan un progreso desde los sistemas de instrumentación basados únicamente en el hardware a sistemas centrados en el software que aprovechan la potencia de cálculo, productividad y capacidad de conexión de los populares ordenadores de propósito general y estaciones de trabajo.

Si nos basamos en un modelo de capas para explicar la estructura de la instrumentación virtual, vemos que dicha estructura estaría formada por 4 niveles bien diferenciados:

- **Capa física y capa de enlace (IEEE 488.1/RS232/VXI/Ethernet):** Permite la comunicación entre los equipos y se basa en un bus físico, compuesto por un conjunto de líneas con niveles lógicos bien definidos y protocolos de comunicación basados en los estados lógicos de las líneas.
- **Capa de comunicación (IEEE 488.2):** Se definen códigos, formatos, protocolos y comandos comunes para todos los instrumentos, lo que permite la adopción del formato de comandos SCPI que estructura las órdenes a los dispositivos de forma coherente, permitiendo la sustitución de instrumentos de distintos fabricantes con cambios mínimos.
- **Capa de comandos (SCPI):** La norma SCPI consigue la estandarización de los comandos de control y

el formato de los datos de los instrumentos. Esta capa sería la más alta dentro de la jerarquía normativa para el control de sistemas de instrumentación.

- **Capa de aplicación (MATLAB/LabVIEW)**

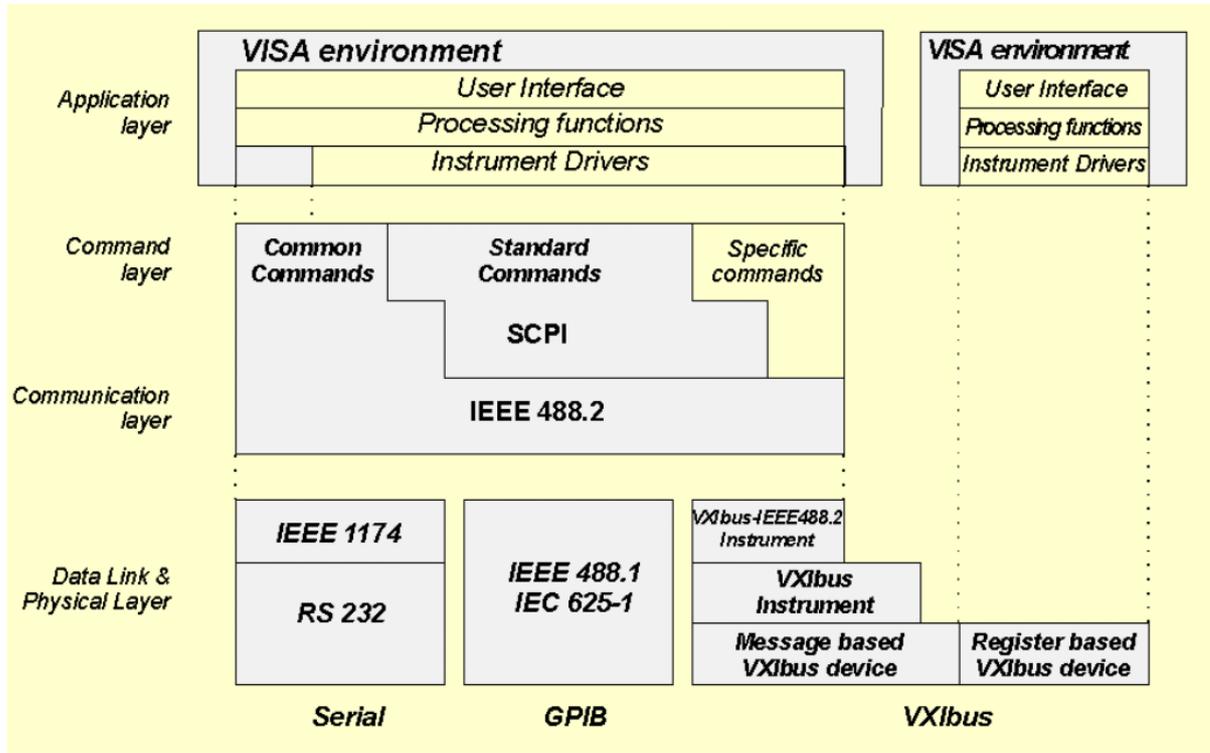


Figura 3.2 Torre de protocolos extendida para el control sistemas de instrumentación.

John M. Pieper. SCPI and VISA a valuable combination.  
<http://www.hit.bme.hu/~papay/edu/Lab/SCPIandVISA.pdf>

### 3.1. Jerarquía de comandos SCPI

Los comandos SCPI utilizan una estructura jerárquica conocida como sistema de árbol. Los comandos asociados se agrupan bajo un nodo común, o raíz, formando así subsistemas.

A continuación, se puede observar cómo quedaría la jerarquía del árbol de comandos.

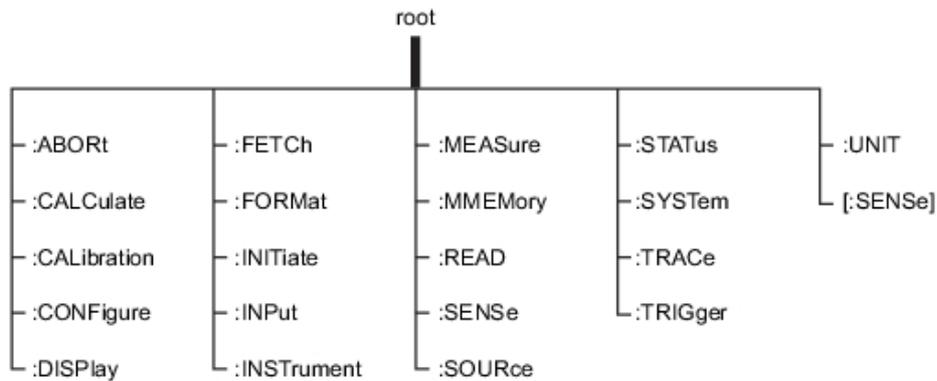


Figura 3.3 Árbol de comandos SCPI. <https://na.support.keysight.com>

Cada comando SCPI, a excepción del “:ABORT”, tiene sus propios subcomandos, que pueden ser comandos

simples o acoplados (ver ejemplos de la sección 3.2).

Cada nivel de subcomando añade una capa adicional al árbol de comandos SCPI. A modo de ejemplo, en la siguiente figura se puede ver el árbol creado a partir del comando principal “:CONFigure”.

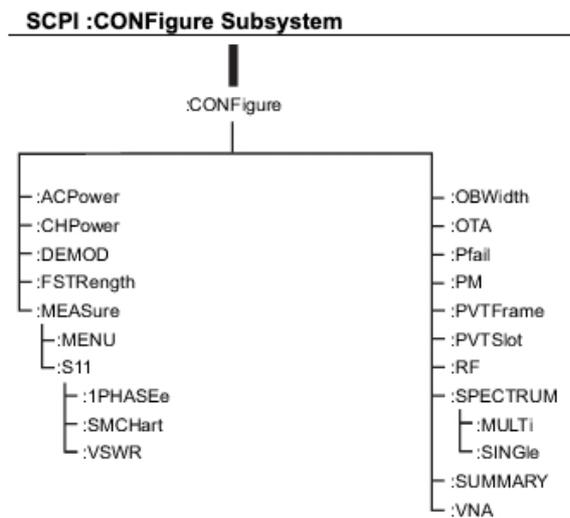


Figura 3.4 Árbol de comandos SCPI del subsistema CONFigure. <https://na.support.keysight.com>

### 3.2. Sintaxis SCPI

La mayoría de los comandos se componen de una mezcla entre caracteres en mayúsculas y minúsculas. Los caracteres en mayúsculas son obligatorios para especificar una orden, mientras que los que están en minúsculas pueden suprimirse. No obstante, para una mejor legibilidad, se usará siempre que sea posible el formato largo, con mayúsculas y minúsculas. Por ejemplo, cualquier combinación de VOLT o VOLTage es aceptada; VOLT, VolTaGe, volt o Volt, pero comandos como VOL o VOLTAG generará un error.

Un ejemplo de la sintaxis de los comandos SCPI es el siguiente:

---

*Ejemplo comando SCPI.* VOLTage:DC:RANGe {<range>|MIN|MAX|DEF}

---

- **Las llaves ( { } )** encierran los posibles parámetros. No son enviadas con el comando.
- **Una barra vertical ( | )** separa un parámetro de otro. En el ejemplo de arriba se indica que puedes elegir entre un parámetro de rango numérico, “MIN”, “MAX” o “DEF”. LA barra no es enviada con el comando.
- **Mayor/Menos que ( < > )** indica que debes especificar un valor determinado. Por ejemplo “VOLT:DC:RANG 5” o “VOLT:DC:RANG MAX”.
- **Corchetes ( [ ] )** para parámetros opcionales. Si no se especifica, el instrumento usa el valor por defecto.

Todos los comandos SCPI, excepto los comandos comunes, los cuales explicaremos más adelante, tienen una estructura jerárquica (ver Figura 3.2). Cada parte de la estructura del comando es separada por dos puntos (:), en lo que se conoce como niveles de palabra clave. Para separar comandos dentro del mismo subsistema se hará uso del punto y coma (;). Estos comandos son conocidos como comandos acoplados.

---

*Ejemplo de separador.* OUTPut:STATE ON, (@1) ;PON:STATe RST

---

Que daría el mismo resultado que enviar los comandos por separado.

---

*Ejemplo sin separador ”;”.*

---

---

*OUTPut ON, (@1)*

*OUTPut:PON:STATe RST*

---

SCPI proporciona varios niveles diferentes de control del instrumento. Los comandos simples de medición proporcionan a los usuarios un control fácil y rápido de la instrumentación. Está diseñado para permitir la incorporación de nuevos comandos sin intervenir ni causar problemas de programación.

En SCPI tenemos dos tipos de comandos bien diferenciados:

- **Comandos de eventos:** Se utilizan para realizar una acción sobre el instrumento con el objetivo de modificar alguno de los parámetros configurables disponibles.

---

*Ejemplo comando de evento SCPI. FREQuency: CW 2000*

---

- **Comandos de consultas:** Usados para solicitar al instrumento el valor de un parámetro o conjunto de parámetros. Este tipo de comandos son finalizados con el carácter "?".

---

*Ejemplo comando de consulta SCPI. FREQuency: CW?*

---

### 3.3. Tipos de datos

Es importante hablar de los parámetros de datos, o simplemente "parámetros", pues son los valores cuantitativos utilizados como argumentos para las palabras clave del comando. El tipo de parámetro asociado con un comando SCPI en particular está determinado por el tipo de información requerida para controlar la función del instrumento en particular. Por ejemplo, los parámetros de tipo booleano (ON | OFF) se utilizan con comandos que controlan las funciones del interruptor.

Los tipos de parámetros más comunes se detallan a continuación:

- **De canal:** Nos sirve para seleccionar uno o más canales (*@<channel>*). Así, si queremos seleccionar un canal determinado como salida, enviaremos el comando:

---

*Ejemplo 3.4.1.      DISPlay[:WINDow]:CHANnel <channel>*

---

- **Númericos:** Se usan tanto en los comandos comunes (los veremos en el siguiente apartado) como en los comandos de subsistema. Comprenden números enteros o número en notación decimal/científica y pueden incluir signos.

---

*Ejemplo 3.4.2.      -7.89E-001 → Permite usar E o e para el exponente.*

---

- **Numéricos extendidos:** Se usan sobre todo para especificar cantidades y valores físicos como GHz, mV, DEG, etc. Incluyen valores como MAXimum y MINimum.

---

*Ejemplo 3.4.3.      1.2GHz → GHz se usa para representar el exponente (E009).*

---

- **Discreto:** Se utilizan para controlar la configuración del programa a un valor o condición predeterminados, por ejemplo, "INTernal" y "EXTernal". Aunque los parámetros discretos se parecen a las palabras clave de los comandos, no debemos confundirlos. En particular, debemos asegurarnos de utilizar correctamente los dos puntos y los espacios. Se utiliza los dos puntos para separar los mnemónicos de comandos entre sí y un espacio para separar los parámetros de los mnemónicos de comandos.

---

**Ejemplo 3.4.4.**      *:RADio1,2,3,4:ARB:FORMat:BORDer NORMal*

---

- **Booleano:** Representan condiciones binarias y pueden expresarse como ON, OFF, ó 0, 1.
- **Cadenas:** Se permite enviar mensajes codificados en ASCII como parámetros. Se usan comillas simples o dobles.

---

**Ejemplo 3.4.5.**      *“This is valid”, ‘THIS IS ALSO VALID’*

---

- **Datos de respuesta real:** Representan números decimales en notación decimal fija o científica. La mayoría de los lenguajes de programación de alto nivel que soportan la entrada/salida manejan la notación decimal o científica de forma transparente.

---

**Ejemplo 3.4.5.**      *+4.00000E+010, -9.990000E+002*

---

### 3.4. Comandos comunes

Existe una serie de comandos llamados “Common Commands” que son de obligado cumplimiento por la norma IEEE 488.2, se usan para controlar los registros de estado del instrumento, la sincronización, etc. Algunos de ellos se detallan a continuación:

- **\*CLS - Clear Status:** Resetea el byte de estado, vaciando la cola de errores y los registros de eventos. También cancela cualquier comando \*OPC anterior.
- **\*ESE - Event Status Enable:** Establece los bits en el registro de activación de estado de eventos estándar.
- **\*ESR - Event Status Enable:** Lee y resetea el registro de activación de estado de eventos.
- **\*OPC - Operation Complete Command:** Genera un mensaje OPC en el registro de estado de eventos estándar cuando todas las operaciones superpuestas pendientes se completan.
- **\*RST - Reset:** Ejecuta un reset en el dispositivo y cancela cualquier operación pendiente.
- **\*SRE - Service Request Enable:** Activa los bits en el registro de petición de servicio.
- **\*WAI - Wait:** Mediante este comando se consigue que el instrumento no ejecute ninguna operación hasta que todas las acciones pendientes se completen.
- **\*IDN? - Identify:** Devuelve una cadena que identifica unívocamente al instrumento.
- **\*LRN? - Learn:** Devuelve una secuencia de comandos SCPI que pueden ser usados más tarde para poner al instrumento en el mismo estado que estaba cuando se envió la propia consulta \*LRN?
- **\*ESE? - Event Status Enable Query:** Devuelve el resultado del registro de estado de eventos estándar. El registro se limpia tras leerlo.
- **\*OPC? – Operation Complete Query:** Devuelve un ASCII “+1” cuando todas las operaciones pendientes se completan.
- **\*OPT? - Identify Options Query:** Devuelve una cadena que identifica la configuración del instrumento.
- **\*RDT? - Returns Output Channel Descriptions:** Identifica los módulos instalados en la fuente de alimentación.
- **\*SRE? - Service Request Enable:** Lee el estado actual del registro SRER (Service Request Enable Register).
- **\*STB? - Status Byte Query:** Lee el valor del registro de estado.
- **\*TST? - Result of Self-test Query:** Devuelve el estado hardware del instrumento.



## 4 FUENTE DE ALIMENTACIÓN N6700

En este capítulo se presentará formalmente la fuente de alimentación que ha sido usada para este Trabajo de Fin de Grado. Veremos desde sus características técnicas hasta su configuración pasando por las ventajas e inconvenientes de usar este tipo de instrumentos.

En el desarrollo de este Trabajo Fin de Grado se ha hecho uso de la fuente de alimentación modular N6700 de Keysight. Esta fuente es capaz de proporcionar hasta cuatro salidas por bastidor. Cada módulo independiente es una fuente para cada salida, lo que nos permite personalizar la tensión, la corriente y el modo de funcionamiento según nos convenga en cada momento.



Figura 4.1 Panel frontal de la fuente de alimentación N6700C.



Figura 4.2 Parte trasera. Conexiones de los módulos 1 y 2.

El objetivo final de una fuente de alimentación de pruebas es poder llevar a cabo y evaluar los resultados de las pruebas de funcionalidad, rendimiento, calidad y estrés realizadas en dispositivos y sistemas electrónicos. Los equipos de pruebas automatizados (ATE, por sus siglas en inglés), automatizan los equipos y procesos de pruebas electrónicas tradicionalmente manuales, y requieren de una interacción humana mínima.

El dispositivo cuyos atributos son evaluados por el ATE suele denominarse “Dispositivo Bajo Prueba”, o DUT (Device Under Test) por sus siglas en inglés.

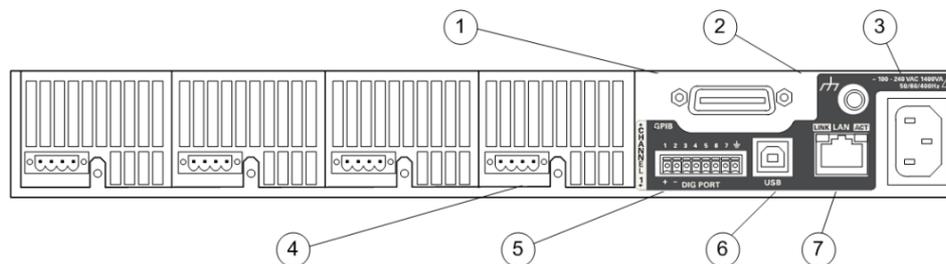
Muchos de los dispositivos electrónicos que se utilizan hoy en día se someten a pruebas de ATE para garantizar el rendimiento adecuado, la funcionalidad y la seguridad de los que lo van a utilizar, o se benefician del uso del dispositivo.

Como ejemplo de dispositivos probados en ATE podemos encontrar las placas de circuito impreso (PCB), los circuitos integrados (CI), los sistemas y módulos de automóviles y todo tipo de sistemas aviónicos.

## 4.1 Conexiones de la N6700

Como podemos apreciar en la Figura 2.5, la fuente de alimentación N6700 viene equipada de serie con interfaces GPIB, USB 2.0 y LAN Ethernet 10/100 Base-T. Mientras que GPIB es la más adecuada para su uso con sistemas existentes, Keysight ofrece USB y LAN para permitir aprovechar la disponibilidad, velocidad y la facilidad de uso de las interfaces estándar comunes de la industria informática. Esta última será la que se use para conectarnos al instrumento. Además, la N6700 es totalmente compatible con la especificación LXI clase C.

Rear Panel at a Glance



1. GPIB - GPIB interface connector.
2. Ground - Chassis ground binding post.
3. IEC 320 connector - AC input connector. Power cord requires a ground conductor.
4. Output connector - Includes +/-output and +/- sense terminals.
5. Digital connector - Pin functions are user-configurable.
6. USB - USB interface connector.
7. LAN - 10/100/1000 Base-T Left LED indicates activity. Right LED indicates link integrity.

**Figura 4.3** Panel trasero de la fuente de alimentación.

*Keysight Series N6700C Low-Profile Modular Power System User's Guide.*

### 4.1.1 General-Purpose Instrumentation Bus (GPIB)

General-Purpose Instrumentation Bus (GPIB) es un estándar que define un bus de datos digital multimaestro de 8 bits paralelos para comunicaciones de corta distancia. En 1978 el bus fue estandarizado por el Institute of Electrical and Electronics Engineers (IEEE) como el IEEE-488 (488.1).

Esta norma permite la conexión de hasta 15 dispositivos con un solo bus paralelo de 8 bits, mediante conexión en cadena, siendo el dispositivo más lento el que determina la velocidad de transferencia. Esta velocidad de transferencia tenía un máximo de 1 MB/s en el estándar original que fueron aumentados a 8 MB/s con la estandarización de la norma IEEE-488.1.

El cable de interconexión está formado por 16 conductores o líneas, organizadas en tres buses: 8 para datos (DIO1-DIO8) y 8 para señales de gestión y estado. [2]

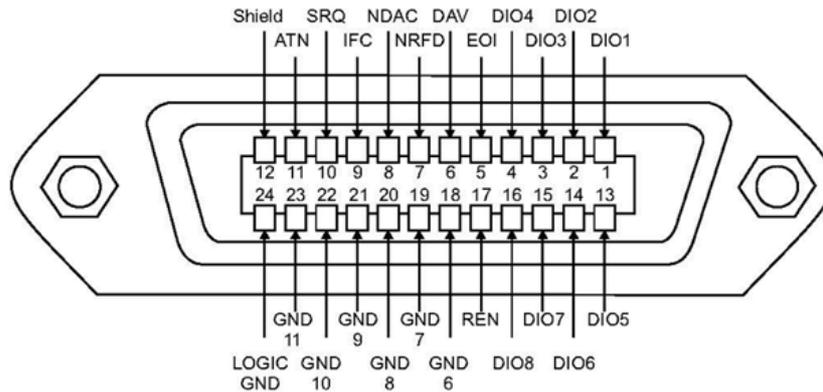


Figura 4.4 Pines de un puerto GPIB. <https://automationforum.co>

- **Bus de datos (DIO1-DIO8):** Bus bidireccional de 8 líneas orientado a la transferencia de bytes o caracteres ASCII.
- **Bus de sincronización de la transferencia de datos:** Conjunto de tres líneas (DAV: Data valid, NRD: Not Ready For Data y NDAC: Not Data Accepted) que se utilizan de forma coordinada para asegurar la transferencia de datos entre los equipos.
- **Bus de control:** Está constituido por 5 líneas (ATN: ATteNtion, IFC: InterFace Clear, SRQ: Service ReQuest, REN: Remote ENable, y EOI: End Or Identify) que se utilizan para transferir comandos entre los equipos relativos al modo de interpretar los datos que se transfieren o comandos básicos de gobierno de la interfaz del bus.

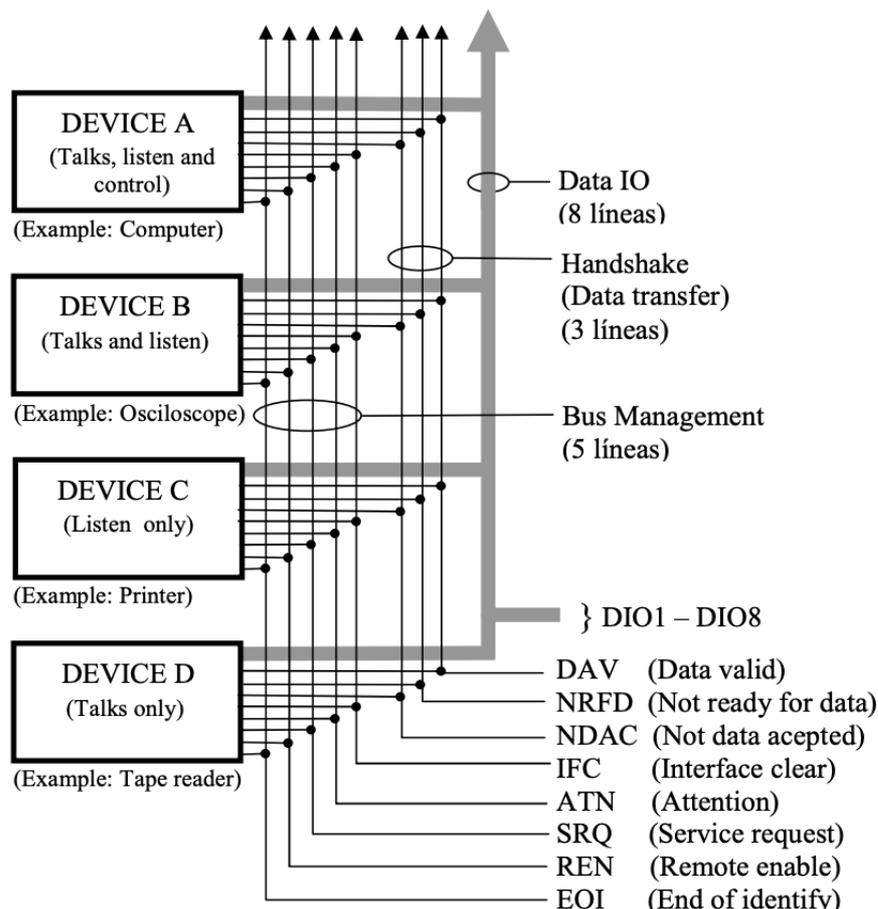


Figura 4.5 Esquema de buses GPIB con cuatro dispositivos.

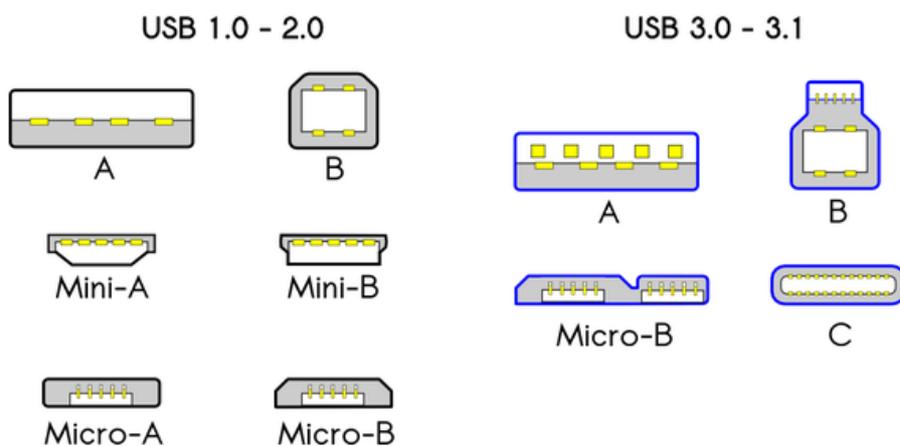
#### 4.1.2 Universal Serial Bus (USB)

Bus de comunicaciones que sigue un estándar industrial el cual define los conductores, conectores y protocolos usados en un bus para conectar, comunicar y suministrar energía entre ordenadores y periféricos. Existe una amplia variedad de hardware USB, incluyendo 14 tipos de conectores diferentes, de los cuales el USB-C es el más reciente.

Lanzado por primera vez en 1996, los estándares USB son mantenidos por el USB Implementers Forum (USB-IF). Las cuatro generaciones de USB son: USB 1.x, USB 2.0, USB 3.x y USB 4.

**Tabla 4-1** Evolución de las distintas versiones de USB.

Version	Speed	Bps	Power Delivery (W)
<b>USB 1.0</b>	Baja velocidad	1.5 Mbps	-
<b>USB 1.1</b>	Velocidad completa	12 Mbps	-
<b>USB 2.0</b>	Alta velocidad	480 Mbps	2.5 W
<b>USB 3.0</b>	Super-Alta velocidad	4.8 Gbps	7.5 W
<b>USB 3.1</b>	Super-alta velocidad+	10 Gbps	100 W
<b>USB 3.2</b>	Super-alta velocidad+	20 Gbps	100 W
<b>USB 4</b>	USB4	40 Gbps	100 W



**Figura 4.6** Diferentes tipos de puertos USB. <https://www.digikey.com>

#### 4.1.3 Ethernet

Ethernet es la tecnología de red de área local (LAN) más utilizada en el mundo actualmente. Los estudios de mercado indican que hasta la fecha se han vendido cientos de millones de tarjetas de interfaz de red (NIC), puertos repetidores y puertos de concentradores de conmutación de Ethernet, y el mercado sigue creciendo. En total, Ethernet supera en ventas a todas las demás tecnologías LAN por un margen muy amplio.

Fue desarrollado a principios de los años 1970 por la empresa de investigación y desarrollo Xerox PARC, pero no fue hasta 1983 cuando fue estandarizado por el IEEE, bajo la norma IEEE 802.3.

En los inicios, la velocidad máxima de transferencia que se podía alcanzar era de 10 Mbps. Esta velocidad ha ido incrementándose con el paso del tiempo hasta llegar a nuestros días, donde podemos alcanzar velocidades nominales de 10 Gbps.

Originariamente, en las primeras configuraciones de red, todos los equipos se conectaban al mismo cable de forma directa, de manera que cualquier mensaje enviado a través del bus llegaba a todos los terminales. Después, cada dispositivo decidía si el mensaje era relevante o no gracias a la dirección MAC, que es una ristra 48 bits que identifica unívocamente a un equipo en una red ethernet.

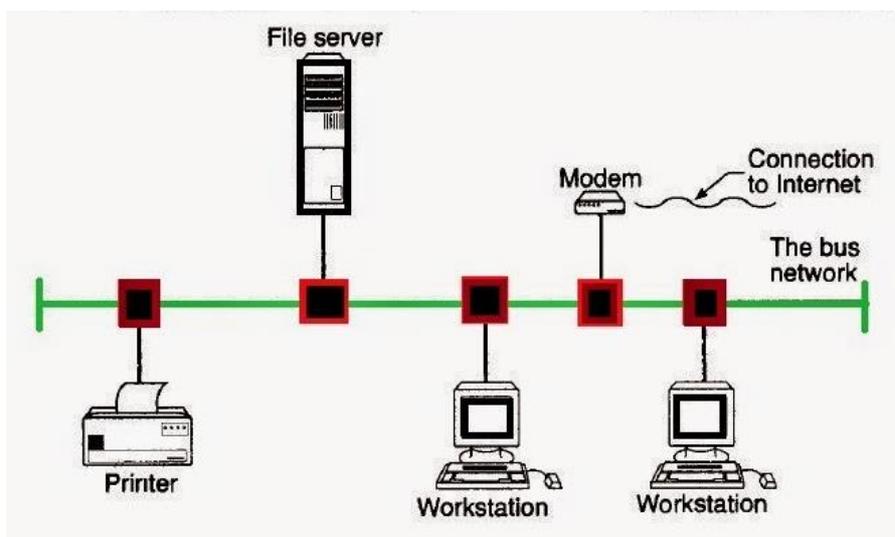


Figura 4.7 Red LAN en topología bus. <http://network-diagram.blogspot.com>

Esto ocasionaba un problema, para empezar, solo permitía mensajes de difusión, además de que todos los equipos del bus podían ver todo el tráfico de datos de la red, lo que se traducía en un gran problema de seguridad.

Con la incorporación de elementos como hubs y switches, se pudo segmentar la red, de manera que se resuelve el problema de seguridad mencionado.

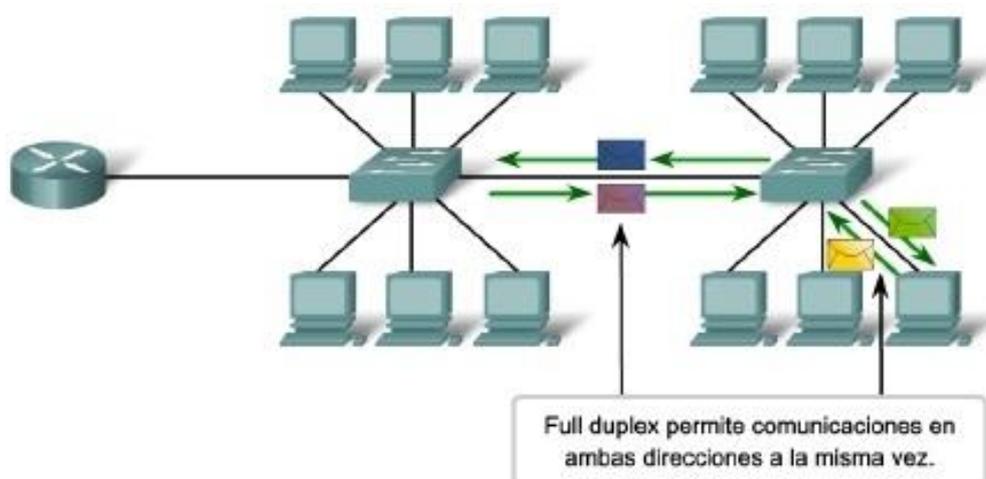


Figura 4.8 Red LAN segmentada por switches. <http://cidecame.uaeh.edu.mx>

En lo que respecta a la conexión de los diferentes elementos de la red, en la actualidad, el cable que más se suele

utilizar es el Ethernet Cat 5e o superiores. A continuación, podemos ver una tabla comparativa de las diferentes categorías:

**Tabla 4-2** Evolución de las diferentes categorías de los cables ethernet.

Categoría	Velocidad	Frecuencia	Velocidad de descarga
Ethernet Cat 5	100 Mbps	100 MHz	15.5 MB/s
Ethernet Cat 5e	1 Gbps	100 MHz	150.5 MB/s
Ethernet Cat 6	1 Gbps	250 MHz	150.5 MB/s
Ethernet Cat 6A	10 Gbps	500 MHz	1.25 GB/s
Ethernet Cat 7	10 Gbps	600 MHz	1.25 GB/s
Ethernet Cat 7A	10 Gbps	1000 MHz	1.25 GB/s
Ethernet Cat 8	40 Gbps	2000 MHz	5 GB/s

Otro de los aspectos importantes en este tipo de cables, es el apantallamiento electromagnético de los hilos de cobre que componen el propio medio.

- **UTP (Unshielded Twisted Pair):** es un cable sin apantallamiento, muy utilizado en entornos domésticos.
- **FTP (Foiled Twisted Pair):** los pares de cables van trenzados y también están sin apantallar, pero tienen una protección global que los protege a todos a la vez. También se recomiendan para uso doméstico.
- **STP (Shielded Twisted Pair):** cada par de cables trenzados están recubiertos de forma individual por una malla conductora, que actúa como apantallamiento. Son recomendables para instalaciones que van entre paredes.
- **SFTP (Shield Foiled Twisted Pair):** en instalaciones entre paredes de larga longitud se suelen emplear estos cables, más costosos, pero también más protegidos. Los pares trenzados de cable están apantallados por una malla protectora individual y a su vez tiene un recubrimiento global para todos los cables.

# UTP FTP STP SFTP



Figura 4.9 Tipos de apantallamiento electromagnético en cables ethernet. <https://computerhoy.com>

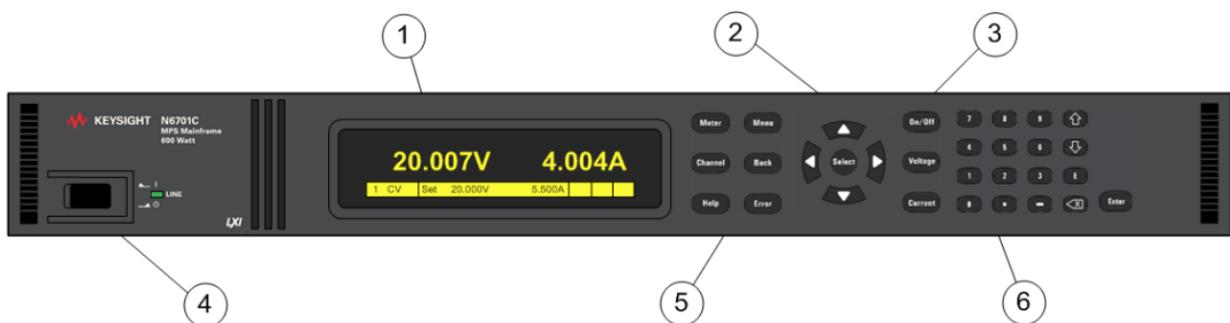
## 4.2 Configuración de la N6700

A continuación, se expondrá la configuración y puesta en marcha de la fuente de alimentación modular “**N6700 Series Modular System power supplies**”. Lo primero que se hizo fue asignarle una IP a la fuente. Debido a que no existen comandos SCPI para la configuración de parámetros de una red LAN, esta configuración se realizó a través de la botonera del panel frontal del dispositivo. Tras configurar los parámetros, se deben guardar los cambios seleccionando: **System\IO\LAN\Apply**. Los parámetros LAN son no volátiles, es decir, se mantienen ante apagados, reinicios o comandos como \*RST.

Para ver la configuración LAN activa, debemos navegar por el display, de nuevo haciendo uso del panel frontal:

**System\IO\LAN\Settings**. Se nos mostrará la dirección IP asignada, la máscara de la subred y la puerta de enlace.

### Front Panel at a Glance



1. **Display** - Turns off after 1 hour of inactivity. Press any key to restore the display.
2. **Navigation keys** - Move the cursor to a menu item. Select the highlighted menu item.
3. **Output keys** - Turn the outputs on or off. Enter voltage or current.
4. **On/Off switch and LED** - LED indicates power is on. Green indicates normal operation. Amber indicates display is in screen-saver mode.

5. **System keys** - Toggle between single-channel and multiple-channel view. Access front panel command menu. Select an output channel to control.
6. **Numeric Entry keys** - Enter values. Arrow keys increment or decrement numeric settings.

**Figura 4.10** Esquema del panel frontal de la Keysight N6700.  
*Keysight Series N6700C Low-Profile Modular Power System User's Guide.*

Para poder trabajar con la fuente, se configuró una conexión LAN a través un cable de red que conectaba el PC del laboratorio con la misma. De esta forma, y haciendo uso del programa TeamViewer, se logró tener la configuración de la fuente lista para realizar algunos tests iniciales de forma remota.

IP de la Keysight N6700: 192.168.0.10

IP del PC: 192.168.0.2

Nuestra fuente de alimentación cuenta con una interfaz web que nos permite controlarla directamente desde cualquier navegador de nuestro PC. A través de esta interfaz web podemos acceder a las funciones de control del panel frontal, incluyendo la configuración LAN. Permite hasta seis conexiones simultáneas, pero debemos tener en cuenta que, a mayor número de conexiones, menor será el rendimiento.

Para entrar en la interfaz web, debemos seguir los siguientes pasos:

1. Abrir el navegador.
2. Introducir el hostname o la dirección IP del instrumento en el buscador.
3. Pinchar en la pestaña 'Web Control' para empezar a controlar el instrumento.

Tras el paso 2, veremos que se nos muestra una interfaz como la de la Figura 5.1.

**KEYSIGHT** N6700C  
TECHNOLOGIES Serial number: MY56010060

Home Web Control Configure LAN

Connected to N6700C  
at IP address 192.168.0.10

Enable front panel identification indicator

Description

Model number	N6700C
Manufacturer	Keysight
Serial number	MY56010060
Firmware revision	E.02.01.3173
Description	Keysight N6700C MPS Mainframe - MY56010060

VISA instrument addresses

HISLIP LAN protocol	TCPIP::K-N6700C-10060.local::hislip0::INSTR
VXI-11 LAN protocol	TCPIP::K-N6700C-10060.local::inst0::INSTR
GPIB over LAN protocol	TCPIP::K-N6700C-10060.local::gpi0,5::INSTR
TCP/IP SOCKET protocol	TCPIP::K-N6700C-10060.local::5025::SOCKET
USB (USBTMC/488)	USB::10893::2::MY56010060::0::INSTR
GPIB	GPIB::5::INSTR

**Figura 4.11** Interfaz web de la N6700.

Tras el paso 3, veríamos una representación del panel frontal del instrumento en tiempo real, donde podremos

manejarlo directamente con el ratón, sin necesidad de tocar físicamente el panel.



Figura 4.12 Interfaz de control del instrumento en tiempo real.

A la hora de conectarnos a la fuente, podremos hacerlo a través de diferentes protocolos como telnet, en el puerto 5024 y/o a través de sockets, en el puerto 5025. El instrumento nos permite hasta 4 combinaciones simultáneas de sockets y conexiones telnet. En este trabajo se usará la opción socket. Un socket de datos en este puerto 5025 puede ser usado para enviar y recibir comandos ASCII/SCPI, consultas y respuestas. Todos los comandos y las respuestas a consultas deben terminar con una nueva línea para que el menjase pueda ser parseado.

También podemos realizar la apertura de un socket de control que nos permite limpiar el estado del dispositivo, el cual, al contrario del socket de datos, va variando el puerto de escucha que se obtiene enviando al socket de datos el comando SCPI: **SYSTem:COMMunicate:TCPIp:CONTRol?**. Una vez obtenido el puerto, podremos abrir la conexión con el socket de control. Al igual que con el socket de datos, todos los comandos y las respuestas a consultas enviados a este socket deben terminar con una nueva línea.

### 4.3 Módulos N6774A y N6744B

La Keysight N6700 es una fuente configurable de un rack-unit (1U) que permite añadir o quitar módulos de potencia según nos convenga, con el objetivo de crear sistemas de potencia optimizados para los diferentes tests que se quieran realizar sobre nuestros DUT's.

Los módulos que se usarán en este trabajo serán los N6774A y N6744B, los cuales nos servirán para hacer uso de dos de los cuatro posibles canales que ofrece el instrumento. Entre otras características, estos módulos son capaces de ofrecer salidas de bajo ruido y posibilitan alta capacidad de respuesta ante cambios de tensión. Además, cada uno de estos módulos está protegido frente a sobretensiones, sobreintensidades y excesos de temperatura.



Figura 4.13 Módulos N6774A y N6744B.

Las características técnicas de ambos módulos se detallan a continuación:

Tabla 4-3 Características técnicas de los módulos N6774A y N6744B.

	DC OUTPUT RATINGS (VOLTS/AMPS/WATTS)	RIPPLE & NOISE (p-p/rms)	VOLTAGE PROGRAMMING ACCURACY	CURRENT PROGRAMMING ACCURACY	VOLTAGE MEASUREMENT ACCURACY	CURRENT MEASUREMENT ACCURACY
N6744B	35 V/3 A/105 W	15 mV/5 mV	0.1% + 35 mV	0.15% + 20 mA	0.1% + 35 mV	0.15% + 4 mA
N6774A	35 V/8.5 A/ 300 W	22 mV/ 5 mV	0.1% + 35 mV	0.15% + 60 mA	0.1% + 35 mV	0.15% + 12 mA

Una de las características más interesantes de estos módulos es la del límite de potencia. Esto nos permite fijar una potencia límite a partir de la cual el instrumento debe apagar la salida tras 1 milisegundo. El bit de estado CP+ indicará que la salida ha sido apagada debido a una condición de límite de potencia. Para volver a habilitar la salida, primero debemos ajustar la carga para que consuma una potencia menor. Después, se debe inhabilitar la función de protección. Hay que tener en cuenta que en estos módulos quizá sea preferible usar el límite de intensidad o tensión para algunas aplicaciones, de manera que se pueda evitar el apagado de la salida.

# 5 DESARROLLO EXPERIMENTAL

---

En este capítulo se explicará el desarrollo y utilidad de los códigos diseñados en MATLAB, los cuales nos servirán para realizar todas las medidas experimentales necesarias. Se trata de una serie de funciones independientes que al combinarlas nos permitirán llevar a cabo diferentes pruebas, obteniendo así los datos que nos llevarán a alcanzar el objetivo de este Trabajo de Fin de Grado.

En las diferentes pruebas se probó las siguientes funcionalidades básicas:

- Construcción del socket de comunicación.
- Habilitar y deshabilitar los diferentes canales.
- Fijar una determinada tensión e intensidad.
- Preguntar por valores de magnitudes.
- Reboot del dispositivo.

Estas funcionalidades serán clave en el desarrollo de los distintos experimentos que se detallarán en capítulos posteriores.

## 5.1 Conexión con el instrumento

En este primer paso del flujo que iremos analizando hasta llegar a la toma de datos, detallaremos cómo establecer la conexión con la fuente de alimentación N6700. Este paso será el que nos permitirá poder enviarle los diferentes comandos al instrumento para controlarlo remotamente.

En el código 5.1 podemos ver cómo se crea la conexión TCP/IP. Con la apertura del socket en el puerto 5025 conseguimos establecer la comunicación de cara a enviar las consultas y recibir las respuestas del instrumento.

Este fichero será llamado desde los distintos ficheros de prueba, con el objetivo de realizar la conexión en cada ejecución. Es importante destacar que esta conexión deberá ser cerrada al final de cada test, realizando una llamada a la función *fclose(objeto\_de\_comunicación)*, ya que si se dejase el socket abierto podría perjudicar futuras conexiones, dejando a la fuente inoperable.

Tras la instanciación de nuestro objeto de comunicación (el cual identificará al instrumento que se quiere controlar) creado a partir de la llamada a la función *tcpip(ip, port)*, deberemos hacer uso de la función *fopen(objeto\_de\_comunicación)* para estar en posición de enviar y recibir los comandos deseados.

## Código 5.1 Conexión con el instrumento.

```
%{
*****
Fuente Keysight Config
*****
-----
Author: Omar Akoudad Muñoz
-----
-----
Description
-----
-Script para la construcción y conexión del objeto TCP/IP.
%}
t = tcpip('192.168.0.10',5025);
fopen(t);
```

## 5.2 Habilitación y deshabilitación de salidas

La posibilidad de poder (des)habilitar los diferentes canales de la fuente de alimentación, es una acción que nos puede ser útil en un gran número de experimentos. Con esta característica, seremos capaces de cortar la tensión e intensidad de un determinado canal si el experimento así lo requiere.



Figura 5.1 Ejemplo con el canal 1 habilitado en modo "Constant Voltage" y canal 2 deshabilitado

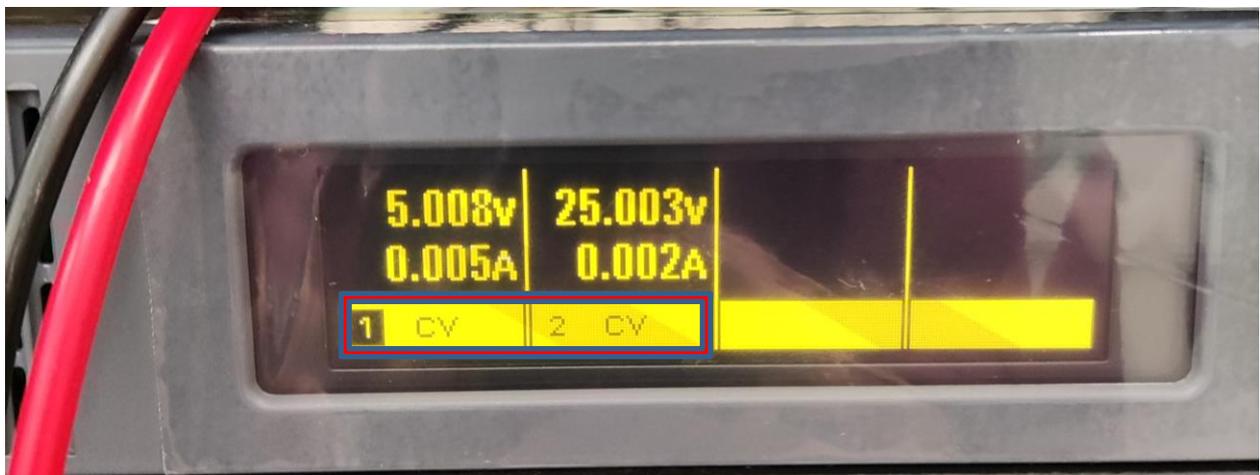


Figura 5.2 Ejemplo con ambos canales habilitados en modo "Constant Voltage"

Para ello, se ha desarrollado el código 5.2 de cara a poder incluirlo en los experimentos en función de la acción que queramos llevar a cabo. La función *fuentesK\_OutputEnabled(output)* recibirá por parámetro la variable *output*, que determinará el número del canal (1 a 4) que queramos habilitar/deshabilitar. Una vez obtenido el canal requerido, se enviará el comando a la fuente de alimentación a través de la función *fprintf()*, que recibe por parámetro el objeto de comunicación “*t*” creado para abrir la conexión TCP/IP y la función *sprintf()*, donde enviaremos el comando SCPI deseado y el propio canal sobre el que aplicarlo, en este caso *OUTPut:STATE ON* para habilitar el canal y *OUTPut:STATE OFF* para deshabilitarlo.

### Código 5.2 Habilitar y deshabilitar salidas.

```
%{
*****
Fuente Keysight Output Enabled
*****
-----
Author: Omar Akoudad Muñoz
-----
-----
Description
-----
-Script para habilitar la salida requerida.
%}
function out = fuentesK_OutputEnabled(output)
fuentesK_Config;
out = output;
chanList = ['@' num2str(out)];
fprintf(t, sprintf('OUTPut:STATE ON, (%s)',chanList));
fclose(t);
end
```

```
%{
*****
Fuente Keysight Output Disabled
*****
-----
Author: Omar Akoudad Muñoz
-----
-----
Description
-----
-Script para deshabilitar la salida requerida.
%}
function out = fuentesK_OutputDisabled(output)
fuentesK_Config;
out = output;
chanList = ['@' num2str(out)];
fprintf(t, sprintf('OUTPut:STATE OFF, (%s)',chanList));
fclose(t);
end
```

## 5.3 Reinicio de la fuente de alimentación

Otra de las funcionalidades útiles que podemos aprovechar de este instrumento es la acción reinicio. Como comentamos antes, es posible que, debido a ciertas acciones ejecutadas en la fuente de alimentación, como dejar abierto un socket o enviar determinados comandos no soportados por según qué versiones, la fuente deje de responder y quede inoperable. Si esto ocurre, la solución más simple y rápida es la de reiniciar el instrumento a través del comando SCPI *SYSTem:REBoot*. Para llevarlo a cabo, se diseñó un script básico con el objetivo de poder realizar esta acción sobre el instrumento.

**Código 5.3** Reinicio de la fuente.

```
%{
*****
Fuente Keysight Output Disabled
*****
-----
Author: Omar Akoudad Muñoz
-----
-----
Description
-----
-Script para el reinicio de la fuente.
%}
fuenteK_Config;
fprintf(t, 'SYSTem:REBoot');
fclose(t);
```

## 5.4 Prueba básica sobre magnitudes

En este apartado vamos a profundizar en una de las partes clave del trabajo. Veremos cómo establecer una tensión y cómo limitar la corriente máxima que queremos proporcionar al dispositivo bajo prueba en cuestión.

En el siguiente código podemos ver lo que se consideraría nuestra primera prueba funcional. Prueba que, apoyándose en los códigos 4.1 y 4.2, nos servirá para establecer una tensión, así como fijar una intensidad máxima (es muy importante limitar la intensidad en los diferentes tipos de experimentos) de cara a poder llevar a cabo los experimentos del capítulo 6.

En este caso, se ha visto conveniente realizar una comprobación para confirmar que los comandos enviados a la fuente de alimentación se ejecutan correctamente. Esto se consigue con el comando *\*OPC?*, ya explicado en capítulos anteriores.

Además, tras comprobar la correcta ejecución de los comandos, solicitaremos al instrumento que nos indique tanto la tensión como la intensidad establecida a través de las consultas *VOLTage?* y *CURRent?* para asegurarnos de que las magnitudes son realmente las que queríamos introducir en el experimento.

**Código 5.4** Test básico de magnitudes.

```

%{
*****
Fuente Keysight Basic Test
*****
-----
Author: Omar Akoudad Muñoz
-----
-----
Description
-----
-Script para establecer una tensión determinada y una corriente máxima.
-----
Steps
-----
1. Habilitamos la(s) salida(s) deseada(s) realizando una llamada a la
función fuenteK_OutputEnabled(output).
2. Llamamos al script fuenteK_Config.m para crear y conectar el objeto
tcpip
3. Se establecen los valores de tensión e intensidad máxima
4. Consultamos si los valores establecidos son los correctos
5. Cerramos la conexión
%}
% Habilitamos la salida(s) que deseemos
fuenteK_OutputEnabled(1);
fuenteK_Config;
chanIndex = 1;
voltage = 3;
current = 0.1;
chanList = ['@' num2str(chanIndex)];
fprintf(t, sprintf('VOLTage %f, (%s)', voltage, chanList));
fprintf(t, sprintf('CURRent %f, (%s)', current, chanList));

% Comprobamos si los comandos se han ejecutado correctamente

opcComp = sscanf(query(t, '*OPC?'), '%d');
while opcComp ~= 1
    opcComp = sscanf(query(t, '*OPC?'), '%d');
end
voltt = query(t, sprintf('VOLTage? (%s)', chanList));
fprintf(voltt);

curre = query(t, sprintf('CURRent? (%s)', chanList));
fprintf(curre);

fclose(t);

```

## 5.5 Monitorización del consumo

Por último, se programó un script con el objetivo de monitorizar el consumo de los diferentes dispositivos y poder determinar la precisión real de la fuente. En este código, se reutilizarán varias de las funcionalidades explicadas anteriormente, como la habilitación de canal o la creación del objeto de comunicación.

El script consiste en realizar un número determinado (según la ventana de tiempo que queramos considerar) de

consultas a la tensión, intensidad y potencia consumida de cara a poder obtener dichas magnitudes y representarlas en diferentes gráficos. Como podemos ver en el Código 5.5, se hará uso de un bucle for preguntando por las magnitudes requeridas para obtener los resultados del experimento. Esto se consigue mediante los comandos SCPI *MEASure:CURRent?* y *MEASure:VOLT?*. Una vez obtenidos los vectores con todos los datos, se pasará a pintar las diferentes gráficas. En los capítulos siguientes pueden observarse algunos ejemplos de gráficas obtenidas con el código en cuestión.

### Código 5.5 Monitorización del consumo del dispositivo.

```
%{
*****
Fuente Keysight Consumption
*****
-----
Author: Omar Akoudad Muñoz
-----
Description
-----
% Script para monitorizar el consumo en el tiempo
fuenteK_OutputEnabled(1);
fuenteK_Config;
chanIndex = 1;
chanList = ['@' num2str(chanIndex)];
tic,
for i=1:100
curre = query(t, sprintf('MEASure:CURRent? (%s)', chanList));
volt = query(t, sprintf('MEASure:VOLT? (%s)', chanList));
I(i) = str2num(curre);
V(i) = str2num(volt);
W(i) = I(i)*V(i);
tiempo(i)=toc;
fprintf('%s I: %4.8f mA V: %4.3f V P: %4.2f
W\n',datestr(datetime('now'),'HH:MM:SS.FFF'),I(i)*1000,V(i),V(i)*I(i))
;
end
fclose(t);
figure, plot(tiempo,V,'b'), xlabel('Time (s)'), ylabel('Voltage
(V)'), grid on,
figure, plot(tiempo,I,'r'), xlabel('Time (s)'), ylabel('Current
(A)'), grid on,
figure, plot(tiempo,I.*V,'g'), xlabel('Time (s)'), ylabel('Power
(W)'), grid on,
```

# 6 RESULTADOS EXPERIMENTALES

---

Después de estudiar los códigos desarrollados del proyecto en el capítulo 5, pasaremos a explicar cómo se combinarán entre sí con el objetivo de realizar un conjunto de pruebas experimentales con distintos dispositivos en el laboratorio.

La primera prueba consistirá en una serie de mediciones básicas. Se comprobará el funcionamiento básico de la fuente haciendo uso de uno de los elementos más básicos de la instrumentación: la resistencia.

Para el segundo experimento, nos serviremos del amplificador de ganancia ultraplana y bajo ruido ZX60-V62+ de Mini-Circuits. Para esta prueba, un poco más sofisticada que la anterior, se hará uso del código *fuentesK\_Consumption.m* ya explicado en el capítulo anterior.

Por último, realizaremos la misma prueba que en el caso anterior, pero con un dispositivo diferente. Utilizaremos una placa Arduino que incorpora un módulo SIM800C diseñado para transmitir y recibir por el método de transferencia de datos de las redes de telefonía móvil 2G, GSM (Global System for Mobile) /GPRS (General Packet Radio Service).

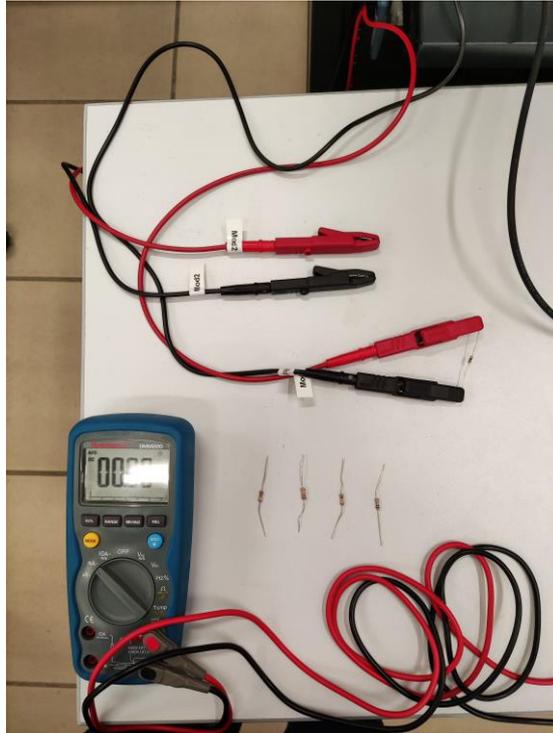
## 6.1 Pruebas con resistencias

Para comprobar el correcto funcionamiento de la fuente de alimentación, se realizaron varias pruebas con diferentes resistencias.

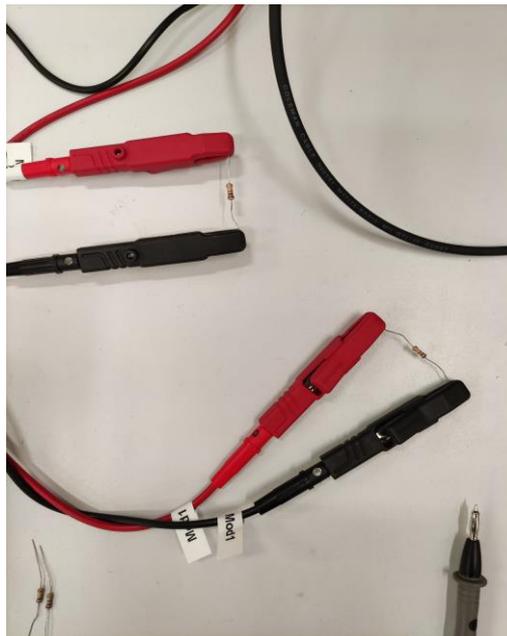
Para estas pruebas, el entorno de trabajo estaba compuesto por:

- PC conectado a la fuente de alimentación.
- Fuente de alimentación modular N6700C.
- Módulos N6774A y N6744B.
- Multímetro DMM220.
- Cables de alimentación a pinzas de cocodrilo.
- Resistencias de 10  $\Omega$ , 1 k $\Omega$ , 10 k $\Omega$ , 1 M $\Omega$ .

En la siguiente figura se puede ver el espacio de trabajo completo:



**Figura 6.1** Espacio de trabajo.



**Figura 6.2** Resistencias alimentadas por la fuente.

Para estas primeras pruebas, se propuso el objetivo de comprobar el comportamiento de la fuente haciendo uso de los códigos explicados en el capítulo anterior. Para ello, se alimentaron las diferentes resistencias aprovechando los dos canales disponibles de la fuente. En la parte inferior de la Figura 6.2 se puede apreciar el correspondiente etiquetado “*Mod1*” haciendo referencia a los cables pertenecientes al canal 1 de la fuente de alimentación.

Se fue probando con diferentes valores de resistencias para comprobar la respuesta de la fuente. En la siguiente tabla podemos ver algunas de las magnitudes obtenidas tras realizar las pruebas.



**Figura 6.3** Tensión aplicada e intensidad consumida por resistencia de 10  $\Omega$



**Figura 6.4** Tensión aplicada e intensidad consumida por resistencias de 1 k $\Omega$  y 10 k $\Omega$

**Tabla 6-1** Valores de consumo de las resistencias.

Resistencia (Ohmios)	Voltaje aplicado (V)	Intensidad consumida fuente (mA)
10 $\Omega$	0.157 V	16 mA
1 k $\Omega$	5.008 V	5 mA
10 k $\Omega$	25.000 V	2 mA

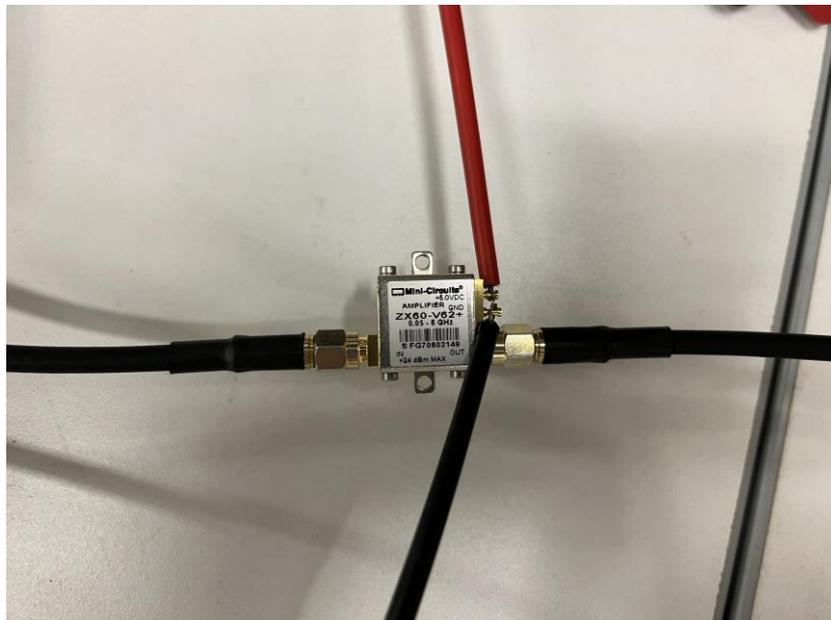
Cabe destacar que la fuente no nos permite limitar la intensidad por debajo de los 15 mA (valor mínimo asignable permitido). Esto, lógicamente, no se traduce en que el instrumento no sea capaz de proporcionar valores menores de 15 mA, pues la intensidad consumida dependerá de la carga conectada, como se puede comprobar en la Tabla 6.1 donde se obtienen valores de intensidad de un orden de magnitud por debajo del mínimo limitante.

## 6.2 Monitorización del consumo del amplificador ZX60-V62+

En este experimento, se usará el amplificador ZX60-V62+. Este amplificador de la familia de amplificadores ZX60 se caracteriza por su alta ganancia, bajo ruido y excelente directividad. Los amplificadores ZX60 ofrecen una construcción monolítica robusta, que integra el conector de RF en el propio cuerpo de la carcasa, proporcionando una alta fiabilidad y capacidad de resistencia en aplicaciones críticas.



**Figura 6.5** Amplificador ZX60-V62+ de Mini-Circuits. <https://www.minicircuits.com>



**Figura 6.6** Montaje del amplificador ZX60-V62+.

En la Figura 6.6, podemos ver lo que sería el montaje para realizar las correspondientes medidas de este experimento. De izquierda a derecha, vemos la entrada del amplificador, a la que aplicamos una señal de 3.6 GHz, la salida, y la alimentación del dispositivo a 6 V.

Del propio datasheet del amplificador se puede sacar el consumo del dispositivo, este se establece en 0.725 W. Para verificar este consumo, se monitorizó el consumo durante aproximadamente un minuto.

En las siguientes figuras se puede ver el voltaje aplicado al dispositivo y su consumo, así como la señal a la salida del amplificador:



Figura 6.7 Tensión y consumo del amplificador ZX60-V62+.

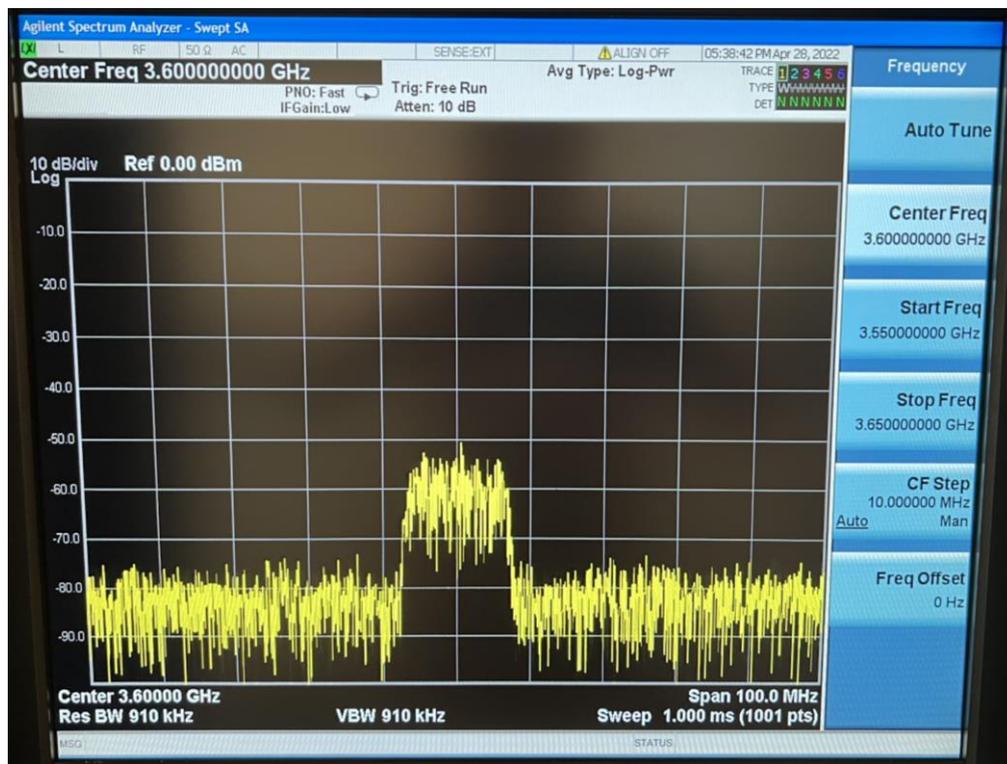
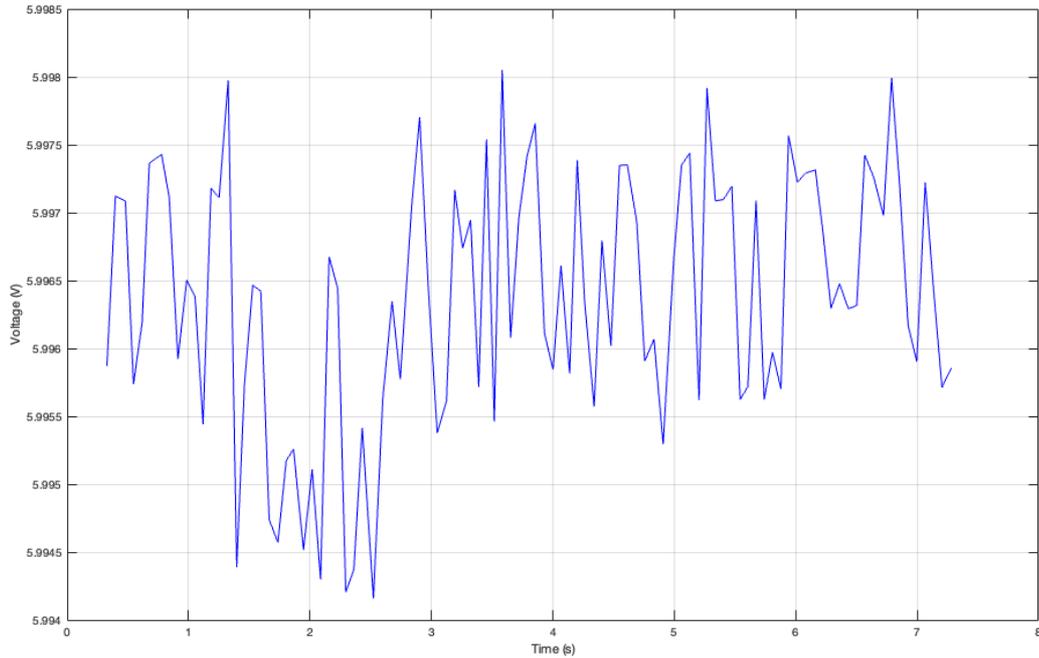


Figura 6.8 Señal de 3.6 GHz a la salida del amplificador

De acuerdo con el datasheet del fabricante, la ganancia típica a 3.6 GHz es de unos 15.6 dB. A pesar del ruido apreciable, el cual nos limitará la precisión del cálculo, en la Figura 6.8 vemos como efectivamente la ganancia ronda este valor. Por ejemplo, si medimos en las zonas pico de la señal obtenemos:

$$G = -55 \text{ dBm} - (-73 \text{ dBm}) = 18 \text{ dB}$$

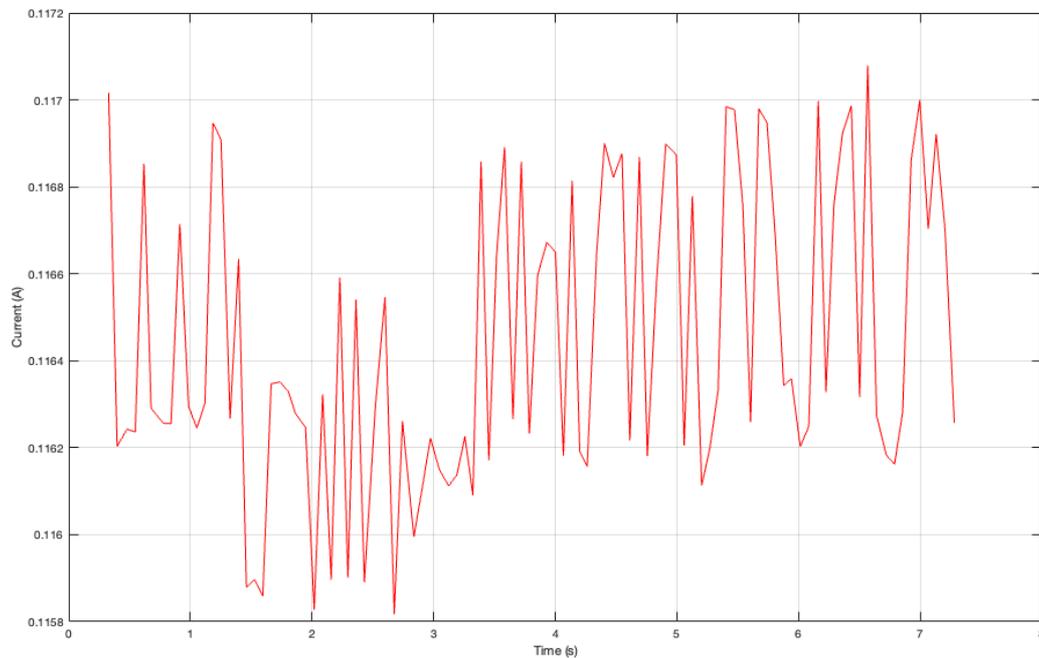
Lo siguiente que vamos a analizar será el propio consumo del dispositivo. Para este caso, hemos representado tanto la tensión aplicada como el propio consumo de intensidad. Como podemos ver en la Figura 6.11, el consumo del amplificador es de 0.70 W, lo cual está próximo a los 0.725 W proporcionados por el fabricante.



**Figura 6.9** Gráfica del voltaje aplicado al amplificador ZX60-V62+.

Vemos que, para el caso de la tensión, si queremos medir la estabilidad que nos proporciona la fuente a la hora de alimentar este dispositivo, tendremos que hacerlo una vez se estabilice, entre los segundos 3 y 7:

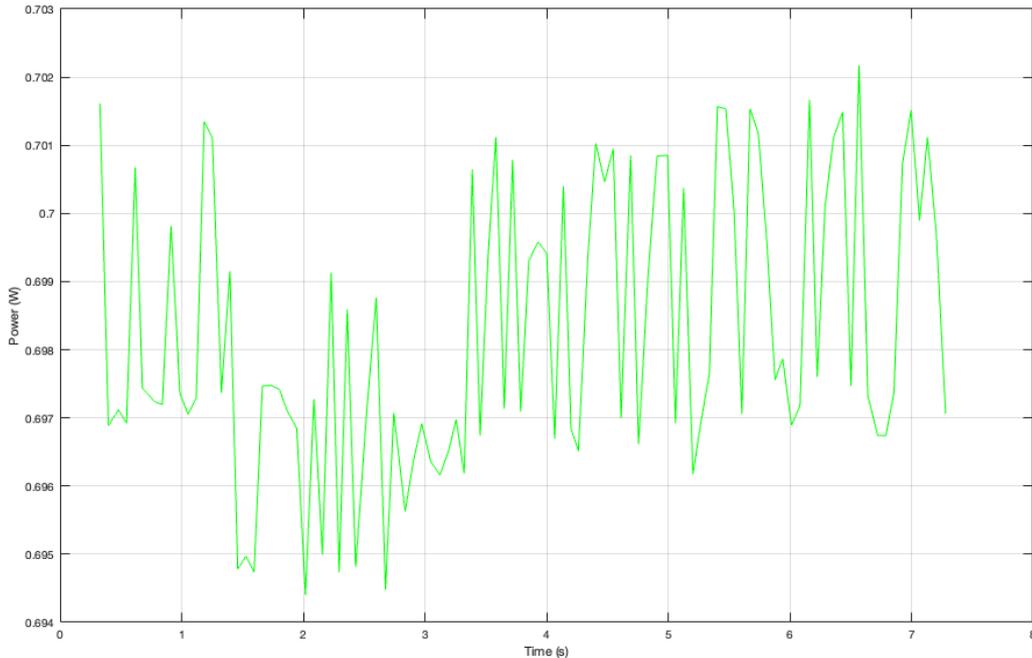
$$\Delta V = 5.998 - 5.9954 = 2.6 \text{ mV}$$



**Figura 6.10** Gráfica de la intensidad consumida por el amplificador ZX60-V62+.

Para medir la estabilidad obtenida en el consumo, haremos lo propio.

$$\Delta I = 0.1171 - 0.1162 = 0.9 \text{ mA}$$



**Figura 6.11** Gráfica del consumo del amplificador ZX60-V62+.

En términos de precisión vemos que el consumo tiene un delta de  $\Delta P = 0.7021 W - 0.6961 W = 6 mW$ , de manera que podemos determinar que el consumo es tan estable como se esperaba.

### 6.3 Monitorización del consumo en placa Arduino, módulo SIM800C

En este último experimento, se propuso el mismo objetivo que en el anterior. Para el caso que nos ocupa, mediremos el consumo del módulo SIM800C de SimCom, un transmisor/receptor de datos por GSM/GPRS, y se comprobará que los datos obtenidos son similares a los proporcionados por el fabricante.

El SIM800C es un módulo GSM/GPRS de cuádruple banda que opera en las frecuencias GSM850MHz, EGSM900MHz, DCS1800MHz y PCS1900MHz.

El SIM800C está diseñado con una técnica de ahorro de energía con la que se consigue que el consumo de corriente de este módulo ronde los 0.6 mA en modo ahorro. En nuestro experimento no se usará este modo ahorro de energía, de manera que tendremos consumos más elevados.

El módulo SIM800C, montado sobre una placa Arduino, se compone de:

- Un puerto serie de tres buses y un puerto serie full-modem
- Un puerto USB
- Un canal de audio que incluye un micrófono y un altavoz
- Entrada y salida programables de propósito general
- Slot para tarjeta SIM
- Soporta Bluetooth

Una vez encendido, tarda unos segundos en empezar a transmitir los datos, ya que primero debe obtener la red. Las fases por las que pasa nuestro dispositivo serían las siguientes:

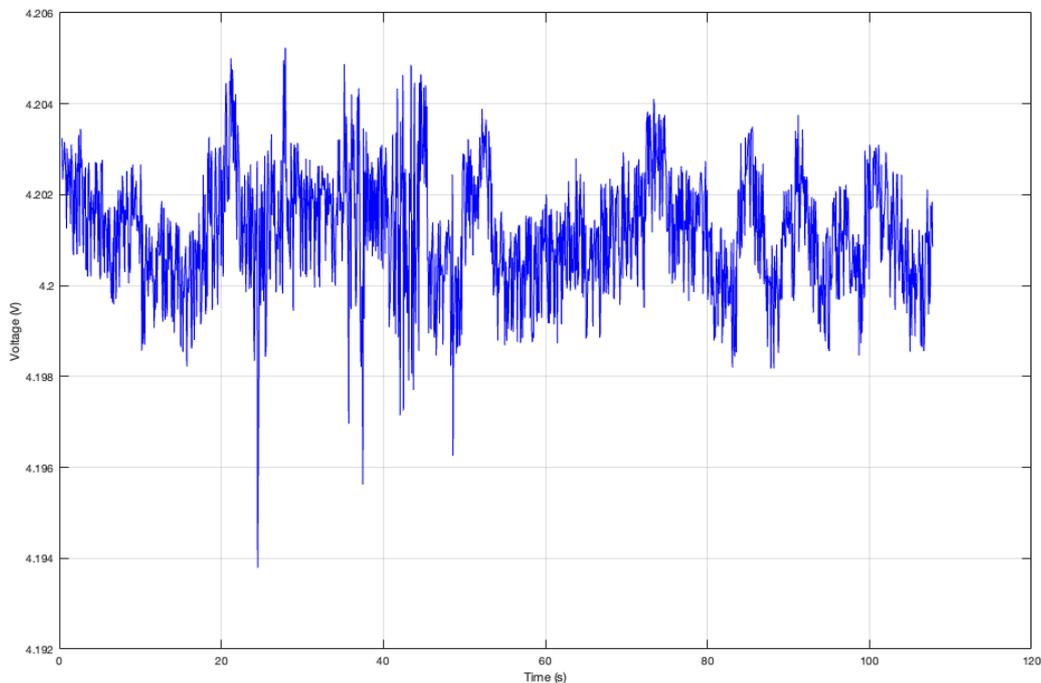
1. Encendido del dispositivo
2. Asignación de red
3. Transmisión de datos

4. Recepción de datos
5. Apagado del dispositivo

Para la monitorización de esta placa, la ventana de tiempo tuvo que ampliarse con respecto a la prueba del transistor ZX60-V62+ debido a que, en este caso, como ya hemos mencionado, tendremos que esperar a que el dispositivo realice todos los pasos indicados anteriormente.

La alimentación del dispositivo según el fabricante parte desde los 3.4 V hasta los 4.4 V, siendo 4.0 V el valor recomendado. En nuestro caso hemos alimentado el dispositivo con 4.2 V. Podemos observar en la Figura 6.12 cómo se comporta la tensión aplicada durante todo el experimento. Vemos como entre los segundos 20 y 50 tenemos una ligera inestabilidad de la tensión debido a que es el intervalo de tiempo en el que el dispositivo realiza tanto el encendido, la transmisión y recepción como el apagado, fases que se aprecian con claridad en las gráficas de consumo (Figura 6.13 y Figura 6.14). Aún así, el incremento de tensión en este caso no sobrepasa los 6 mV.

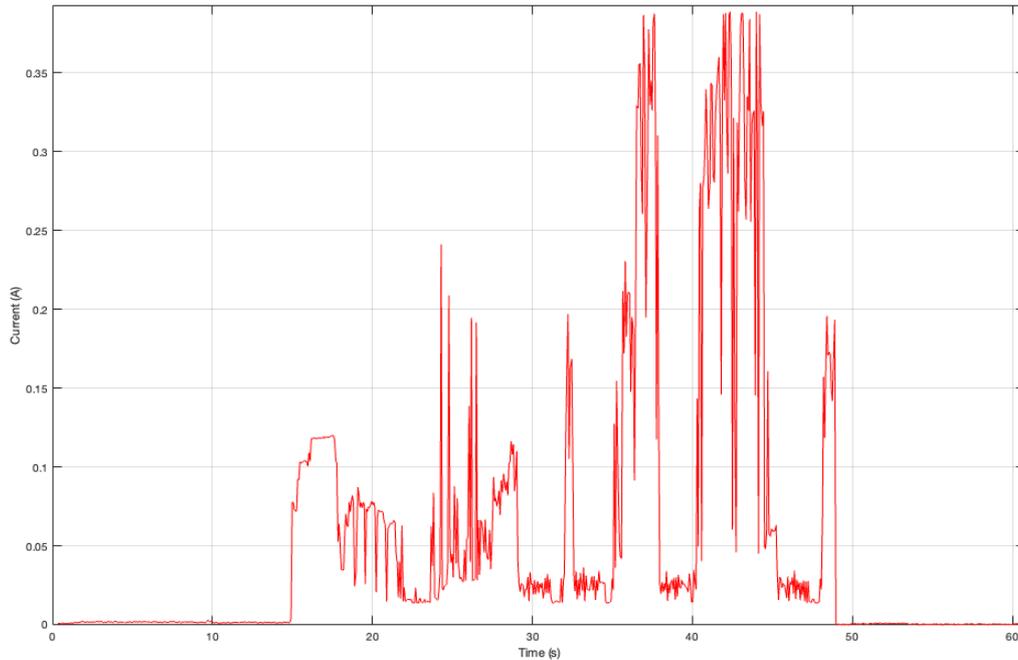
$$\Delta V = 4.204 - 4.198 = 6 \text{ mV}$$



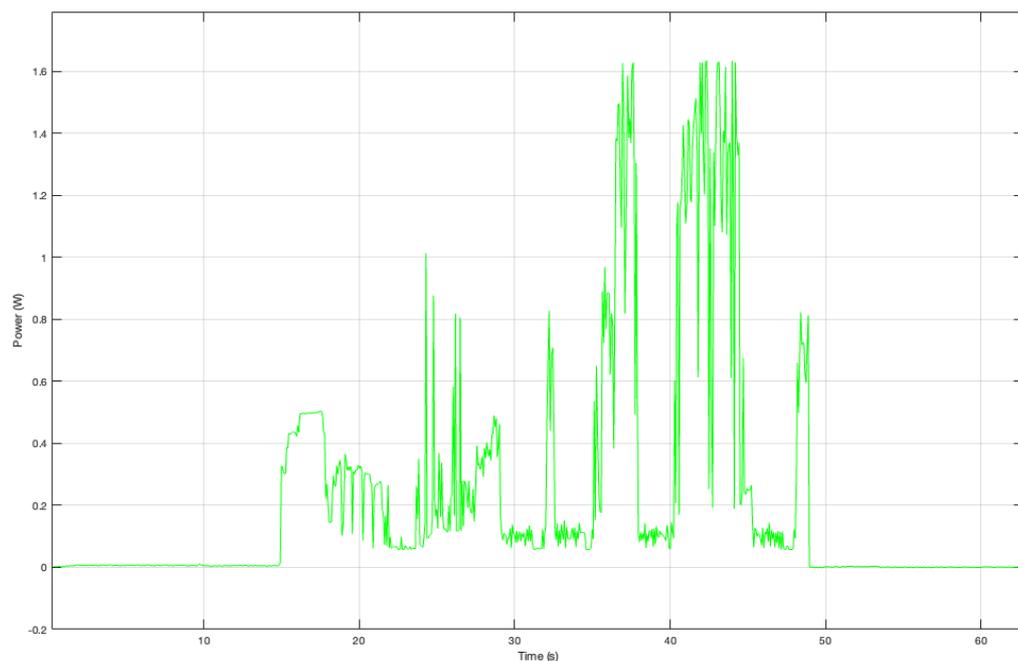
**Figura 6.12** Tensión aplicada a la placa Arduino.

Como se puede apreciar en la Figura 6.13, el pico observado entre los 35 y 46 segundos se corresponde con la fase de transmisión y recepción del módulo. Vemos que el valor alcanzado en esta etapa, en torno a los 395 mA, es cercano a lo indicado por el fabricante en la Hardware Design Guide del dispositivo.

Data mode GPRS (1Rx,4Tx):			
GSM850	394		mA
EGSM900	416		mA
DCS1800	271		mA
PCS1900	285		mA



**Figura 6.13** Intensidad consumida por la placa Arduino.



**Figura 6.14** Gráfica de la monitorización del consumo del módulo SIM800C.

En la Figura 6.14 podemos ver la curva de consumo de potencia del dispositivo. En ella se puede observar cada una de las fases explicadas al comienzo de este capítulo.

En primer lugar, aproximadamente a los 15 segundos, podemos ver cómo el dispositivo se enciende, consumiendo para ello unos 0.5 W. Después, tenemos la fase de asignación de red por parte de la tarjeta SIM, que comprende desde los 25 segundos hasta los 33 segundos, cuando consigue obtener red con éxito. En el tramo que abarca desde los 35 segundos hasta los 46 segundos, observamos la transmisión y recepción del módulo, que es la parte de mayor consumo de todo el proceso superando los 1.6 W, y por último vemos la fase de desconexión y el propio apagado del dispositivo.



## 7 CONCLUSIONES Y LÍNEAS FUTURAS DE TRABAJO

---

En este Trabajo Fin de Grado, se ha presentado un instrumento novedoso y versátil que nos proporciona un sinfín de posibilidades a la hora de ejecutar diferentes estudios que involucran todo tipo de DUT's. Gracias a la fuente programable N6700 de Keysight y su modularidad (cuatro posibles canales), conseguimos una alta flexibilidad que nos permite la configuración de múltiples ATE's que a menudo requieren más de una fuente de alimentación. Además, no hay que olvidar todas las ventajas que presenta el propio concepto de instrumentación virtual, que nos permite generar un gran número de experimentos optimizando los tiempos de desarrollo gracias a la posibilidad de reutilizar códigos y configuraciones.

Además, se ha hecho uso del lenguaje SCPI, estándar que nos permite una altísima compatibilidad, obteniendo la posibilidad de reutilizar la mayor parte del trabajo realizado con otros instrumentos que soporten dicho estándar.

Hemos visto en el desarrollo de este proyecto, cómo con pocas líneas de código y gracias también a la multidisciplinariedad del lenguaje de programación MATLAB, se ha conseguido obtener curvas de consumo de diferentes dispositivos, customizar tensiones e intensidades de una forma muy sencilla.

De cara a futuras líneas de trabajo, como ya hemos mencionado, la posibilidad de tener múltiples canales nos da un margen muy amplio para realizar todo tipo de experimentos con dispositivos electrónicos de diversa índole. Por ejemplo, podríamos caracterizar todo tipo de curvas, tanto de amplificadores como de cualquier dispositivo semiconductor (diodos, transistores, circuitos integrados, modulares, reguladores, etc.).

# **Datasheet del amplificador ZX60-V62+.**

---

Ultra Flat Gain  
**Wideband Amplifier**

**ZX60-V62+**

50Ω      0.05 to 6 GHz



CASE STYLE: GC957

**The Big Deal**

- Ultra Flat Gain
- Broadband High Dynamic Range
- Wideband, 0.05 to 6 GHz

**Product Overview**

The ZX60-V62+ (RoHS compliant) uses Mini-Circuits' HBT technology to offer ultra flat gain over a broad frequency range and high IP3. Housed in a rugged, cost effective unibody chassis, this amplifier supports a wide variety of applications requiring moderate power output, low distortion and 50 ohm matched input/output ports.

**Key Features**

Feature	Advantages
Ultra Flat Gain	±1.1 dB over 50 to 6000 MHz; ±0.1 dB over 700 to 2700 MHz; ±0.2 dB over 500 to 4500 MHz supports a variety of multi band applications
Broadband: 0.05 to 6 GHz	Broadband covering primary wireless communications bands: Cellular, PCS, LTE, WiMAX, UHF, VHF, L band, Satcom, radar, etc.
High IP3 vs. DC power consumption 39 dBm typical at 0.05 GHz 36 dBm typical at 0.8 GHz	This model matches good IP3 performance relative to power consumption. The HBT structure provides good linearity over a broad frequency range as shown in the IP3 being typically 20 dB above the P1dB point to 0.8 GHz. This feature makes this amplifier ideal for use in: <ul style="list-style-type: none"> <li>• driver amplifiers for complex waveform upconverter paths</li> <li>• drivers in linearized transmit systems</li> </ul>
Unconditionally Stable	No risk of damage to other components from impedance mismatch or internal oscillation
Very Small Size, 0.75" x 0.75"	The unique unibody construction enables the ZX60-V62+ to be used in compact designs.

**Notes**

- A. Performance and quality attributes and conditions not expressly stated in this specification document are intended to be excluded and do not form a part of this specification document.  
B. Electrical specifications and performance data contained in this specification document are based on Mini-Circuit's applicable established test performance criteria and measurement instructions.  
C. The parts covered by this specification document are subject to Mini-Circuits standard limited warranty and terms and conditions (collectively, "Standard Terms"); Purchasers of this part are entitled to the rights and benefits contained therein. For a full statement of the Standard Terms and the exclusive rights and remedies thereunder, please visit Mini-Circuits' website at [www.minicircuits.com/MCLStore/terms.jsp](http://www.minicircuits.com/MCLStore/terms.jsp)



[www.minicircuits.com](http://www.minicircuits.com) P.O. Box 350166, Brooklyn, NY 11235-0003 (718) 934-4500 sales@minicircuits.com

# Ultra Flat Gain Wideband Amplifier

## ZX60-V62+

50Ω      0.05 to 6 GHz

### Features

- Ultra Flat Gain,  $\pm 0.7$  dB over 50-4000 MHz
- Gain, 15.4 dB typ. at 2 GHz
- High Pout, P1dB, +19 dBm typ. at 2 GHz
- High IP3, 39 dBm typ. at 50 MHz, 33.4 dBm at 2 GHz
- Excellent ESD protection, class 1C for HBM

### Applications

- Base station infrastructure
- Portable wireless
- CATV & DBS
- MMDS & Wireless LAN
- LTE
- SATCOM
- Radar



Generic photo used for illustration purposes only

CASE STYLE: GC957

Connectors	Model
SMA	ZX60-V62+

**+RoHS Compliant**

The +Suffix identifies RoHS Compliance. See our web site for RoHS Compliance methodologies and qualifications

### Electrical Specifications at 25°C and 5.0V unless noted

Parameter	Condition (GHz)	Min.	Typ.	Max.	Units
Frequency Range		0.05		6	GHz
Gain	0.05	14.8	16.6	18.2	dB
	0.8	13.9	15.5	17.2	
	2.0		15.4		
	3.0		15.5		
	4.0	13.5	15.6	17.0	
Gain Flatness	0.05 - 4		$\pm 0.7$		dB
	0.7 - 2.6		$\pm 0.2$		
Input Return Loss	0.05	11.0	15.4	20.1	dB
	0.8		14.7		
	2.0		20.1		
	3.0		26.6		
	4.0		20.7		
Output Return Loss	0.05	12.0	13.8	11.0	dB
	0.8		15.3		
	2.0		11.0		
	3.0		10.5		
	4.0		12.0		
Output IP3	0.05	31.5	39.1	30.4	dBm
	0.8		36.2		
	2.0		33.4		
	3.0		30.4		
	4.0		27.6		
Output Power @ 1 dB compression	0.05	17.5	19.7	19.0	dBm
	0.8		17.5		
	2.0		17.2		
	3.0		17.9		
	4.0		15.8		
Noise Figure	0.05		5.0	6.2	dB
	0.8		5.0		
	2.0		5.1		
	3.0		5.1		
	4.0		5.1		
Directivity (Isolation-Gain)	0.05 - 6		6.0		dB
DC Voltage		4.8	5.0	5.2	V
DC Current		72	82	92	mA

### Notes

- A. Performance and quality attributes and conditions not expressly stated in this specification document are intended to be excluded and do not form a part of this specification document.  
 B. Electrical specifications and performance data contained in this specification document are based on Mini-Circuit's applicable established test performance criteria and measurement instructions.  
 C. The parts covered by this specification document are subject to Mini-Circuits standard limited warranty and terms and conditions (collectively, "Standard Terms"); Purchasers of this part are entitled to the rights and benefits contained therein. For a full statement of the Standard Terms and the exclusive rights and remedies thereunder, please visit Mini-Circuits' website at [www.minicircuits.com/MCLStore/terms.jsp](http://www.minicircuits.com/MCLStore/terms.jsp)



[www.minicircuits.com](http://www.minicircuits.com) P.O. Box 350166, Brooklyn, NY 11235-0003 (718) 934-4500 sales@minicircuits.com

REV. B  
 ECO-000670  
 ED-14683/1  
 ZX60-V62+  
 CW/TH/CP  
 191118  
 Page 2 of 4

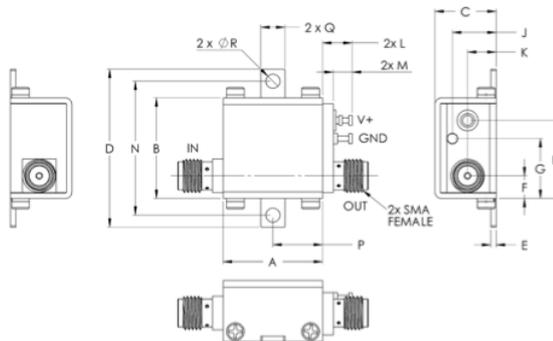
## ZX60-V62+

### Maximum Ratings

Parameter	Ratings
Operating Temperature	-40°C to 85°C Case
Storage Temperature	-55°C to 100°C
DC Voltage	6 V
Input RF Power (no damage)	24 dBm
Power Consumption	0.725 W

Permanent damage may occur if any of these limits are exceeded.

### Outline Drawing



**!** NOTE: When soldering the DC connections, caution must be used to avoid overheating the DC terminal. See Application Note, [AN-40-010](#).

### Outline Dimensions (inch/mm)

A	B	C	D	E	F	G	H	J	K	L	M	N	P	Q	R	wt
.74	.75	.46	1.18	.04	.17	.45	.59	.33	.21	.22	.14	1.00	.37	.18	.106	grams
18.80	19.1	11.68	30.0	1.02	4.32	11.4	14.99	8.38	5.33	5.59	3.56	25.40	9.40	4.57	2.69	23.0

#### Notes

- A. Performance and quality attributes and conditions not expressly stated in this specification document are intended to be excluded and do not form a part of this specification document.  
 B. Electrical specifications and performance data contained in this specification document are based on Mini-Circuits' applicable established test performance criteria and measurement instructions.  
 C. The parts covered by this specification document are subject to Mini-Circuits' standard limited warranty and terms and conditions (collectively, "Standard Terms"); Purchasers of this part are entitled to the rights and benefits contained therein. For a full statement of the Standard Terms and the exclusive rights and remedies thereunder, please visit Mini-Circuits' website at [www.minicircuits.com/MCLStore/terms.jsp](http://www.minicircuits.com/MCLStore/terms.jsp)



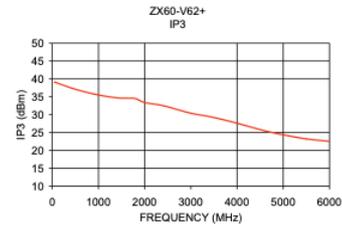
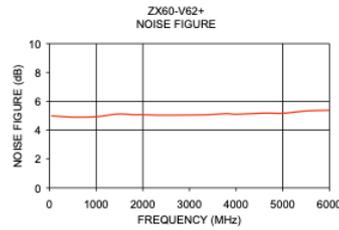
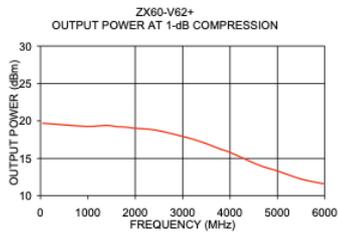
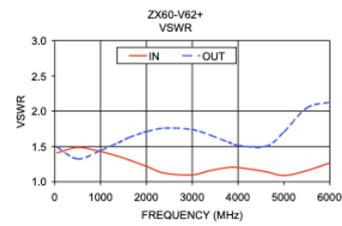
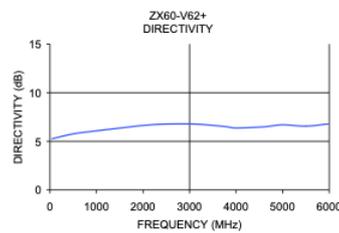
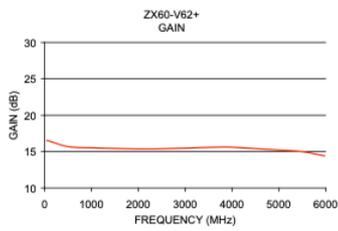
[www.minicircuits.com](http://www.minicircuits.com) P.O. Box 350166, Brooklyn, NY 11235-0003 (718) 934-4500 sales@minicircuits.com

Page 3 of 4

## Typical Performance Data/Curves

## ZX60-V62+

FREQUENCY (MHz)	GAIN (dB)	DIRECTIVITY (dB)	VSWR (:1)		P <sub>OUT</sub> at 1dB COMPR. (dBm)	NOISE FIGURE (dB)	OUTPUT IP <sub>3</sub> (dBm)
			IN	OUT			
50.00	16.55	5.27	1.41	1.49	19.7	5.0	39.1
500.00	15.69	5.77	1.49	1.32	19.5	4.9	37.1
1000.00	15.53	6.08	1.43	1.44	19.3	4.9	35.5
1400.00	15.44	6.31	1.36	1.56	19.4	5.1	34.7
1600.00	15.42	6.42	1.31	1.62	19.2	5.1	34.6
1800.00	15.39	6.53	1.27	1.67	19.2	5.1	34.5
2000.00	15.37	6.63	1.22	1.71	19.0	5.1	33.4
2400.00	15.38	6.76	1.12	1.76	18.8	5.0	32.5
3000.00	15.47	6.79	1.10	1.74	17.9	5.1	30.4
3400.00	15.57	6.68	1.16	1.66	17.2	5.1	29.5
3800.00	15.63	6.50	1.20	1.56	16.2	5.1	28.3
4000.00	15.61	6.37	1.20	1.52	15.8	5.1	27.6
4600.00	15.37	6.49	1.14	1.50	14.1	6.2	25.5
5000.00	15.22	6.70	1.09	1.70	13.3	5.2	24.4
5500.00	14.99	6.56	1.16	2.04	12.2	5.3	23.2
6000.00	14.38	6.80	1.27	2.13	11.6	5.4	22.5



### Notes

- A. Performance and quality attributes and conditions not expressly stated in this specification document are intended to be excluded and do not form a part of this specification document.
- B. Electrical specifications and performance data contained in this specification document are based on Mini-Circuit's applicable established test performance criteria and measurement instructions.
- C. The parts covered by this specification document are subject to Mini-Circuit's standard limited warranty and terms and conditions (collectively, "Standard Terms"); Purchasers of this part are entitled to the rights and benefits contained therein. For a full statement of the Standard Terms and the exclusive rights and remedies thereunder, please visit Mini-Circuit's website at [www.minicircuits.com/MCStore/terms.jsp](http://www.minicircuits.com/MCStore/terms.jsp)



[www.minicircuits.com](http://www.minicircuits.com) P.O. Box 350166, Brooklyn, NY 11235-0003 (718) 934-4500 sales@minicircuits.com

Page 4 of 4

# Índice de Figuras

---

<b>Figura 2.1</b> Fuente de alimentación clásica. <a href="https://kolbi.es">https://kolbi.es</a>	3
<b>Figura 2.2</b> Etapas de una fuente de alimentación. <a href="https://www.zonamaker.com">https://www.zonamaker.com</a>	3
<b>Figura 2.3</b> Rectificador de onda completa. <a href="https://www.researchgate.net">https://www.researchgate.net</a>	4
<b>Figura 2.4</b> Salida del rectificador (izq.) vs Salida del rectificador filtrada (dcha.). <a href="http://platea.pntic.mec.es">http://platea.pntic.mec.es</a>	5
<b>Figura 2.5</b> Regulador de tensión LM7805. <a href="https://www.hwlibre.com/lm7805/">https://www.hwlibre.com/lm7805/</a>	5
<b>Figura 3.1</b> Torre de protocolos simplificada para el control sistemas de instrumentación.	7
<b>Figura 3.2</b> Torre de protocolos extendida para el control sistemas de instrumentación.	8
<b>Figura 3.3</b> Árbol de comandos SCPI. <a href="https://na.support.keysight.com">https://na.support.keysight.com</a>	8
<b>Figura 3.4</b> Árbol de comandos SCPI del subsistema CONFigure. <a href="https://na.support.keysight.com">https://na.support.keysight.com</a>	9
<b>Figura 4.1</b> Panel frontal de la fuente de alimentación N6700C.	13
<b>Figura 4.2</b> Parte trasera. Conexiones de los módulos 1 y 2.	13
<b>Figura 4.3</b> Panel trasero de la fuente de alimentación.	14
<b>Figura 4.4</b> Pines de un puerto GPIB. <a href="https://automationforum.co">https://automationforum.co</a>	15
<b>Figura 4.5</b> Esquema de buses GPIB con cuatro dispositivos.	15
<b>Figura 4.6</b> Diferentes tipos de puertos USB. <a href="https://www.digikey.com">https://www.digikey.com</a>	16
<b>Figura 4.7</b> Red LAN en topología bus. <a href="http://network-diagram.blogspot.com">http://network-diagram.blogspot.com</a>	17
<b>Figura 4.8</b> Red LAN segmentada por switches. <a href="http://cidecame.uaeh.edu.mx">http://cidecame.uaeh.edu.mx</a>	17
<b>Figura 4.9</b> Tipos de apantallamiento electromagnético en cables ethernet. <a href="https://computerhoy.com">https://computerhoy.com</a>	19
<b>Figura 4.10</b> Esquema del panel frontal de la Keysight N6700.	20
<b>Figura 4.11</b> Interfaz web de la N6700.	20
<b>Figura 4.12</b> Interfaz de control del instrumento en tiempo real.	21
<b>Figura 4.13</b> Módulos N6774A y N6744B.	22
<b>Figura 5.1</b> Ejemplo con el canal 1 habilitado en modo "Constant Voltage" y canal 2 deshabilitado	24
<b>Figura 5.2</b> Ejemplo con ambos canales habilitados en modo "Constant Voltage"	24
<b>Figura 6.1</b> Espacio de trabajo.	30

<b>Figura 6.2</b> Resistencias alimentadas por la fuente.	30
<b>Figura 6.3</b> Tensión aplicada e intensidad consumida por resistencia de 10 $\Omega$	31
<b>Figura 6.4</b> Tensión aplicada e intensidad consumida por resistencias de 1 k $\Omega$ y 10 k $\Omega$	31
<b>Figura 6.5</b> Amplificador ZX60-V62+ de Mini-Circuits. <a href="https://www.minicircuits.com">https://www.minicircuits.com</a>	32
<b>Figura 6.6</b> Montaje del amplificador ZX60-V62+.	32
<b>Figura 6.7</b> Tensión y consumo del amplificador ZX60-V62+.	33
<b>Figura 6.8</b> Señal de 3.6 GHz a la salida del amplificador	33
<b>Figura 6.9</b> Gráfica del voltaje aplicado al amplificador ZX60-V62+.	34
<b>Figura 6.10</b> Gráfica de la intensidad consumida por el amplificador ZX60-V62+.	34
<b>Figura 6.11</b> Gráfica del consumo del amplificador ZX60-V62+.	35
<b>Figura 6.12</b> Tensión aplicada a la placa Arduino.	36
<b>Figura 6.13</b> Intensidad consumida por la placa Arduino.	37
<b>Figura 6.14</b> Gráfica de la monitorización del consumo del módulo SIM800C.	37

# Índice de Tablas

---

<b>Tabla 4-1</b> Evolución de las distintas versiones de USB.	16
<b>Tabla 4-2</b> Evolución de las diferentes categorías de los cables ethernet.	18
<b>Tabla 4-3</b> Características técnicas de los módulos N6774A y N6744B.	22
<b>Tabla 6-1</b> Valores de consumo de las resistencias.	31

# Índice de Códigos

---

Código 5.1 Conexión con el instrumento.	24
Código 5.2 Habilitar y deshabilitar salidas.	25
Código 5.3 Reinicio de la fuente.	26
Código 5.4 Test básico de magnitudes.	26
Código 5.5 Monitorización del consumo del dispositivo.	28

# BIBLIOGRAFÍA

---

- [1] A. Z. J. y. J. V. L. Zorzano, «Sistema de Aprendizaje de Programación de Instrumentos con Comandos SCPI,» de *Departamento de Ingeniería Eléctrica. Escuela Técnica Superior de Ingeniería Industrial. Universidad de la Rioja. España.*, 2006.
- [2] «ISTR - Ingeniería Software y Tiempo Real,» [En línea]. Available: [https://www.ctr.unican.es/asignaturas/instrumentacion\\_5\\_IT/GPIB1\\_04.pdf](https://www.ctr.unican.es/asignaturas/instrumentacion_5_IT/GPIB1_04.pdf). [Último acceso: 2022].
- [3] R. R. Piñero, Desarrollo de una herramienta de drive testing para caracterizar la propagación radio, Trabajo Final de Carrera, 2017.
- [4] D. L. Morilla, Desarrollo de herramientas para el control remoto de una fuente de alimentación mediante Python, Trabajo Fin de Grado, Universidad de Sevilla, 2018.
- [5] A. V. Gregorio, Implementación de Interfaz de Control de Laboratorio de Radiofrecuencia en Python, Trabajo Fin de Grado, Universidad de Sevilla, 2020.
- [6] M. E. Borrero, Herramienta Software para el Control Remoto de una Fuente de Alimentación mediante una Interfaz Gráfica, Sevilla: Universidad de Sevilla, Julio de 2011.
- [7] E. A. Rubio, Contro de Instrumentos mediante el Bus GPIB programado con MATLAB, Proyecto Final de Carrera, Universitat Autònoma de Barcelona, 2010.
- [8] I. P. Muñoz, Control remoto de instrumentación para radiocomunicación mediante Python, Trabajo Fin de Grado, Universidad de Sevilla, 2017.
- [9] MathWorks, «MATLAB,» [En línea]. Available: <https://es.mathworks.com/products/matlab.html>. [Último acceso: 2022].
- [10] K. Technologies, «IEEE 488.2 Common Commands,» [En línea]. Available: [https://na.support.keysight.com/pxi/help/latest/Programming/GPIB\\_Command\\_Finder/Common\\_Commands.htm](https://na.support.keysight.com/pxi/help/latest/Programming/GPIB_Command_Finder/Common_Commands.htm). [Último acceso: Diciembre 2021].
- [11] K. Technologies, «Keysight Series N6700C Low-Profile Modular Power System Operating and Service Guide,» [En línea]. Available: <https://www.keysight.com/es/en/assets/9018-04957/service-manuals/9018-04957.pdf>. [Último acceso: Diciembre 2021].
- [12] B. Daniel, «What Is Automatic Test Equipment (ATE)?,» 2011. [En línea]. Available: <https://www.trentonsystems.com/blog/automatic-test-equipment-overview>.
- [13] S. Consortium, Standar Commands for Programmable Instruments, Volume 1: Syntax and Style, 1999.
- [14] «Keysigth Technologies N6700 Modular Power System Family,» 2017. [En línea]. Available: [https://www.mouser.es/datasheet/2/878/5989\\_6319EN-1143005.pdf](https://www.mouser.es/datasheet/2/878/5989_6319EN-1143005.pdf). [Último acceso: 2022].

- [15] R. & Schwarz, «Herramienta básica de secuenciación de comandos SCPI,» 2010. [En línea]. Available: [https://www.rohde-schwarz.com/es/aplicaciones/herramienta-b-sica-de-secuenciacion-para-comandos-scp-nota-de-aplicacion\\_56280-15512.html](https://www.rohde-schwarz.com/es/aplicaciones/herramienta-b-sica-de-secuenciacion-para-comandos-scp-nota-de-aplicacion_56280-15512.html). [Último acceso: 2022].
- [16] SIMCOM, «SIM800C Hardware Design V1.02,» [En línea]. Available: [https://www.elecrow.com/download/SIM800C\\_Hardware\\_Design\\_V1.02.pdf](https://www.elecrow.com/download/SIM800C_Hardware_Design_V1.02.pdf). [Último acceso: 2022].
- [17] Engineer Ambitiously, «What Is LXI?,» Diciembre 2021. [En línea]. Available: <https://www.ni.com/es-es/innovations/white-papers/08/what-is-lxi-.html>. [Último acceso: 2022].
- [18] Alissa Irei, «Understanding the evolution of Ethernet,» November 2019. [En línea]. Available: <https://www.techtarget.com/searchnetworking/feature/Understanding-the-evolution-of-Ethernet>. [Último acceso: 2022].
- [19] IONOS, «¿Qué es Ethernet (IEEE 802.3)?,» [En línea]. Available: <https://www.ionos.es/digitalguide/servidores/know-how/ethernet-ieee-8023/>. [Último acceso: 2022].