

# Tester para amplificador óptico

Ing. Lucas Constantino

Carrera de Especialización en Sistemas Embebidos

**Director:** Dr. Ing. Alejandro Ghersin (FIUBA)

**Jurados:**

Jurado 1 (pertenencia)  
Jurado 2 (pertenencia)  
Jurado 3 (pertenencia)

*Ciudad de Buenos Aires, Junio de 2025*



## *Resumen*

El resumen debe escribirse en uno o dos párrafos. Debe ser breve y conciso, sin ningún elemento de formato en el texto como itálicas o negritas. Tampoco se deben usar siglas ni acrónimos que no resulten obvios para un lector promedio de la memoria, ni referencias bibliográficas o notas al pie de página. No debe faltar qué es lo que se hizo/logró, qué importancia/valor tiene el proyecto/resultado, qué va a encontrar el lector en la memoria y qué contenidos de la especialización/maestría se aplicaron en el proyecto.



# Índice general

<b>Resumen</b>	<b>I</b>
<b>1. Introducción general</b>	<b>1</b>
1.1. Contexto de la empresa . . . . .	1
1.2. Estado del arte . . . . .	3
1.3. Motivación . . . . .	4
1.4. Objetivos y alcance . . . . .	5
<b>2. Introducción específica</b>	<b>7</b>
2.1. Requerimientos . . . . .	7
2.2. Funcionamiento de un amplificador óptico . . . . .	8
2.3. Interfaz del amplificador óptico . . . . .	9
2.4. Componentes del sistema . . . . .	11
2.4.1. Microcontrolador . . . . .	11
2.4.2. Monitor de corriente . . . . .	11
2.4.3. Pantalla táctil LCD . . . . .	11
2.5. Recursos de software . . . . .	12
2.5.1. Sistema operativo de tiempo real FreeRTOS . . . . .	12
2.5.2. Capa de abstracción de hardware (HAL) . . . . .	12
2.6. Periféricos utilizados . . . . .	13
2.6.1. Conversor analógico-digital (ADC) . . . . .	13
2.6.2. Universal Asynchronous Receiver Transmitter (UART) . . . . .	13
2.6.3. Serial Peripheral Interface (SPI) . . . . .	13
<b>3. Diseño e implementación</b>	<b>15</b>
3.1. Arquitectura de hardware . . . . .	15
3.1.1. Medición de corriente del amplificador . . . . .	16
3.1.2. Relé de alimentación . . . . .	17
3.1.3. Medición de tensión del amplificador . . . . .	17
3.1.4. Conexión a pantalla LCD . . . . .	17
3.1.5. Entradas y salidas digitales . . . . .	18
3.1.6. Indicadores de potencia óptica . . . . .	18
3.1.7. Interfaz UART . . . . .	19
3.2. Arquitectura de firmware . . . . .	19
3.2.1. Tarea del backend . . . . .	20
3.2.2. Tarea del frontend . . . . .	21
3.2.3. Máquina de estados . . . . .	23
Diseño . . . . .	23
Implementación . . . . .	25
3.2.4. Capa de abstracción del amplificador . . . . .	25
3.2.5. Driver del ADC . . . . .	26
3.2.6. Driver de la pantalla LCD . . . . .	26

3.2.7. Driver de UART . . . . .	27
3.2.8. Consola de control . . . . .	28
3.3. Dispositivo implementado . . . . .	30
<b>4. Ensayos y resultados</b>	<b>33</b>
4.1. Instrumental utilizado . . . . .	33
4.2. Pruebas de hardware . . . . .	33
4.2.1. Prueba del monitor de corriente . . . . .	34
4.3. Pruebas de firmware . . . . .	34
4.3.1. Prueba del monitor de tensión . . . . .	35
4.3.2. Prueba del monitor de corriente . . . . .	35
4.3.3. Prueba de la pantalla LCD . . . . .	36
4.3.4. Prueba de la consola de control . . . . .	37
4.4. Pruebas de integración . . . . .	38
4.4.1. Funcionamiento normal . . . . .	38
4.4.2. Detección de alarmas . . . . .	38
4.4.3. Detección de sobrecorriente . . . . .	39
4.5. Comparación del prototipo con producto existente . . . . .	40
4.6. Cumplimiento de requerimientos . . . . .	40
<b>5. Conclusiones</b>	<b>41</b>
5.1. Conclusiones generales . . . . .	41
5.2. Trabajo futuro . . . . .	41
<b>A. Circuito esquemático completo</b>	<b>43</b>
<b>B. Banco de ensayos</b>	<b>47</b>
<b>Bibliografía</b>	<b>49</b>

# Índice de figuras

1.1. Propuesta de la empresa Skyloom Global <sup>1</sup> . . . . .	2
1.2. Esquema de uso y conexionado del sistema. . . . .	3
1.3. Placa de depuración. . . . .	4
2.1. Pérdidas en la fibra <sup>2</sup> . . . . .	8
2.2. Configuración interna de un EDFA <sup>3</sup> . . . . .	9
2.3. Pantalla LCD táctil MSP2807 <sup>4</sup> . . . . .	11
3.1. Diagrama completo del hardware implementado. . . . .	15
3.2. Esquema del monitor de corriente. . . . .	16
3.3. Circuito del monitor de corriente. . . . .	17
3.4. Arquitectura de firmware. . . . .	20
3.5. Diagrama de flujo de la tarea del backend. . . . .	21
3.6. Diagrama de flujo de la tarea del frontend. . . . .	23
3.7. Diagrama simplificado de la máquina de estados. . . . .	24
3.8. Tipo de dato de la máquina de estados. . . . .	25
3.9. Capa de abstracción del EDFA. . . . .	26
3.10. Diagrama de flujo de la consola de control. . . . .	29
3.11. Placa fabricada para el trabajo. . . . .	31
4.1. Error porcentual de la constante de conversión de corriente. . . . .	34
4.2. Error porcentual de la medición de tensión. . . . .	35
4.3. Error porcentual de la medición de corriente. . . . .	36
4.4. Pantalla LCD durante el encendido de la placa. . . . .	37
4.5. Uso de la consola de control. . . . .	37
4.6. Pantalla LCD durante funcionamiento normal. . . . .	38
4.7. Detección de alarma y apagado de salida óptica. . . . .	39
4.8. Pantalla LCD luego de detección de alarma. . . . .	39
4.9. Detección de sobrecorriente y desconexión del EDFA. . . . .	40
A.1. Circuito esquemático. Conectores. . . . .	44
A.2. Circuito esquemático. Interfaz con EDFA. . . . .	45
A.3. Circuito esquemático. Alimentación del EDFA. . . . .	46
B.1. Banco de ensayos. . . . .	47



# Índice de tablas

2.1. Señales de la interfaz del EDFA. . . . .	10
3.1. Configuración del ADC. . . . .	26
3.2. Configuración de los SPI del LCD. . . . .	27
3.3. Configuración de la interfaz UART. . . . .	28
3.4. Comandos/argumentos de la consola. . . . .	30
4.1. Lista de instrumental utilizado. . . . .	33
4.2. Comparación entre el dispositivo implementado y el existente. . . . .	40



# Capítulo 1

## Introducción general

En este capítulo se explican las causas que dan origen al dispositivo desarrollado y la función que este cumple en el proyecto de la empresa Skyloom Global. Se hace una breve descripción de un amplificador óptico, del hardware ya existente y por último se presentan los objetivos y alcances determinados durante la planificación de este trabajo.

### 1.1. Contexto de la empresa

La creciente demanda de mayores velocidades de transmisión de información en el ámbito aeroespacial, combinada con las limitaciones de la radiofrecuencia, favorecen el surgimiento de nuevas empresas enfocadas en el desarrollo de nuevas tecnologías capaces de superar estas barreras. Tal es el caso de la empresa Skyloom Global, que propone la creación de una red de satélites de órbita baja (LEO, del inglés, Low Earth Orbit) [1] utilizando enlaces ópticos de alta velocidad.

Esta red se basa en un satélite geoestacionario (órbita GEO) [2] al que los satélites en LEO transmiten la información (denominado *uplink*) para luego descargar esta a una estación terrena (denominado *downlink*). Como los satélites en LEO completan unas 11 vueltas sobre su órbita por día, no tienen una visión constante de la estación terrena, sino que solo están en contacto con ella durante determinadas ventanas de tiempo. Por lo tanto, si se quisiera descargar datos, esto solo sería posible durante esas ventanas, lo que genera un retraso considerable entre el momento en que el satélite obtiene la información y el momento en que se descarga a Tierra.

El satélite en GEO, en cambio, al tener el mismo período orbital que la Tierra, mantiene una visión continua de la estación terrena. De esta forma, al realizar el traspaso de información, se pueden obtener los datos con alta disponibilidad y mínimo retraso. Un diagrama de esta propuesta se muestra en la figura 1.1.



FIGURA 1.1. Propuesta de la empresa Skyloom Global<sup>1</sup>.

El enlace óptico, que permite la transmisión y recepción de datos a una velocidad de 1 Gbps (1 Gigabit por segundo) entre satélites, se establece mediante una terminal de comunicaciones óptica, que es el principal producto actualmente en desarrollo por la empresa. Estas terminales cuentan con un láser que trabaja sobre una longitud de onda de 1550 nm (banda C del espectro). Dicho láser es el encargado de transmitir la información propiamente dicha mediante pulsos de luz (no visible).

En líneas generales, los láseres de esta longitud de onda que se encuentran en el mercado no cuentan con la potencia óptica necesaria para que sean detectados por el receptor. Esto se debe principalmente a que la distancia de espacio libre estimada entre dos satélites en LEO es de unos 4000 km [3].

Para solucionar el problema antes mencionado, se introduce a la salida del láser un amplificador dopado con erbio o EDFA (del inglés, *Erbium Doped Fiber Amplifier*) [4]. La función de este dispositivo es aumentar la potencia del láser varias veces, de forma que se alcance un nivel adecuado para la transmisión.

El modelo de amplificador utilizado por la empresa cuenta con un conector que provee una interfaz electrónica para poder controlar su funcionamiento. Esta interfaz cuenta con distintas señales y buses de comunicación que se explican con mayor detalle en la sección 2.3.

---

<sup>1</sup>Imagen tomada de <https://www.skyloom.co/>

Este amplificador formará parte de la terminal de comunicaciones y estará sometido a intensivas pruebas de funcionamiento y rendimiento durante la etapa de investigación y desarrollo. Para esto, es indispensable contar con una herramienta que permita a los ingenieros a cargo utilizar el amplificador de forma aislada, es decir, sin depender del hardware del producto final.

En la figura 1.2 se puede ver un esquema de uso del dispositivo y sus conexiones con el hardware externo.

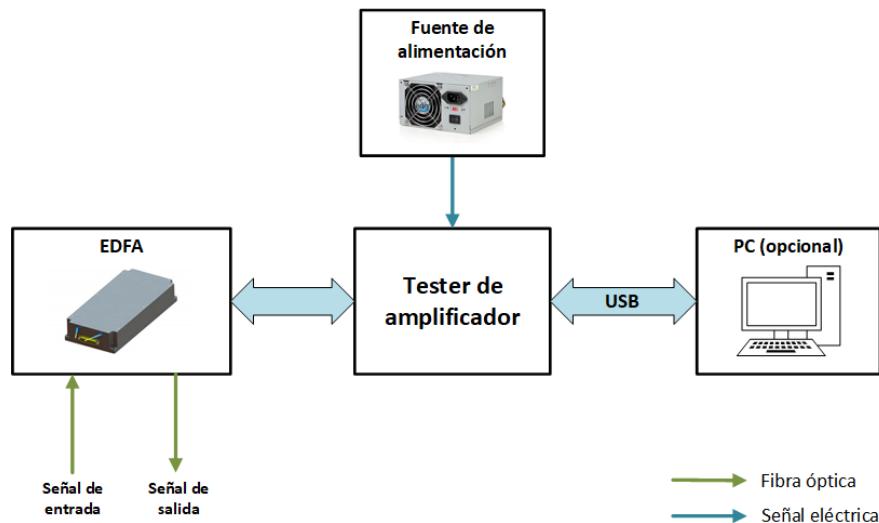


FIGURA 1.2. Esquema de uso y conexionado del sistema.

El dispositivo cuenta con tres conexiones externas: la interfaz con el amplificador, la conexión a la fuente de alimentación y un puerto USB.

La interfaz con el EDFA permite controlarlo y consultar diversos parámetros de funcionamiento a través de las señales presentes en el conector. La fuente de alimentación se encarga de energizar el tester y el EDFA. Y por último, la conexión USB permite controlar el amplificador del mismo modo que en el tester, por lo que el uso de la PC es opcional. Para poder hacer esto, sobre esta debe correr un software que permita establecer una comunicación.

## 1.2. Estado del arte

Si bien hoy en día los EDFA son ampliamente utilizados en varias aplicaciones de optoelectrónica, en la actualidad no existe un estándar para la interfaz eléctrica que poseen estos amplificadores. Esto lleva a que cada fabricante defina una propia para sus productos y que el usuario tenga que adaptar su hardware para poder usarlos.

Para poder realizar pruebas sobre el EDFA de forma individual previo a la integración al producto final, se debe contar con un tester o una placa de depuración dedicada que permita, entre otras cosas, verificar su correcto funcionamiento. Al no existir en el mercado un producto estandarizado para este tipo de amplificadores, el usuario debe adquirir una que ofrece el mismo fabricante del EDFA. Actualmente, la empresa hace uso de una de estas placas, la cual se puede ver en la figura 1.3.

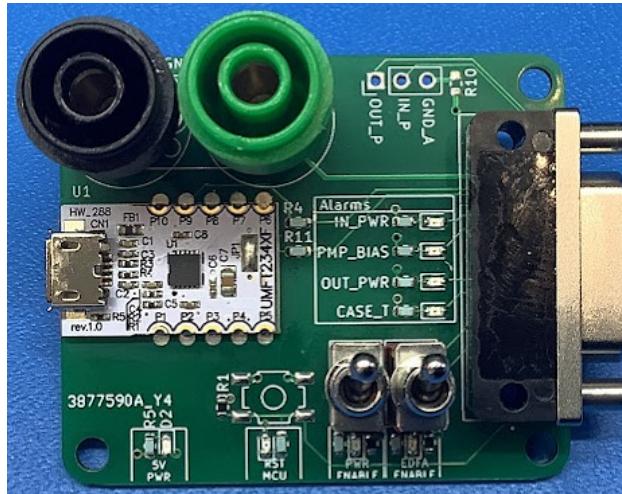


FIGURA 1.3. Placa de depuración.

Esta placa, denominada EDFA IFC (Interface Card), se conecta directamente al puerto del EDFA y cuenta con la electrónica necesaria para proveer al usuario algunas funcionalidades básicas que le permiten hacer uso del amplificador. Contiene una interfaz USB-UART para enviar comandos al EDFA mediante una computadora, LEDs para las señales de alarmas, llaves para las señales de control y conectores para la alimentación del EDFA.

### 1.3. Motivación

La empresa ya lleva algún tiempo haciendo uso de la placa mencionada en la sección anterior y recientemente ha tomado la decisión de dejar de usarla y diseñar una propia debido a los motivos que se listan a continuación:

- No permite el uso de todas las señales presentes en la interfaz del amplificador.
- Tiene una tasa de fallas muy alta, en particular la interfaz UART.
- Su costo es muy elevado en relación a sus prestaciones.
- El tiempo de entrega del producto es de varias semanas.
- No posee las protecciones eléctricas necesarias para proteger el amplificador.

Las desventajas aquí expuestas son los principales motivos que dieron origen a la necesidad de contar con el sistema propuesto en este trabajo. Esto le permite a la empresa no depender del fabricante para la entrega de estos testers y por lo tanto reducir costos y tiempos.

Por otro lado, el tester desarrollado en este trabajo no solo posee las mismas funcionalidades que el ofrecido por el fabricante sino que también incorpora otras características adicionales que lo hacen más fácil de usar, menos propenso a fallas y más seguro.

## 1.4. Objetivos y alcance

El objetivo principal de este trabajo fue el desarrollo de un dispositivo capaz de controlar y realizar mediciones sobre un amplificador de fibra óptica. Las tareas contempladas fueron:

- Diseño y construcción de un prototipo funcional del dispositivo.
- Diseño e implementación del firmware del dispositivo.
- Diseño de los bancos de prueba y ensayos.
- Simulación del funcionamiento del hardware mediante software.
- Documentación de diseño y manual de uso.

Los puntos del desarrollo que no se contemplaron en el trabajo fueron:

- Diseño y construcción de la versión final del dispositivo.
- Especificación de las pruebas a ejecutar sobre el amplificador óptico utilizando el dispositivo.
- Fabricación del PCB del dispositivo.
- Procesamiento e interpretación de los valores de los parámetros del EDFA.
- Diseño y construcción de la fuente de alimentación externa.
- Diseño e implementación del software a ejecutarse en la PC.



## Capítulo 2

# Introducción específica

Este capítulo lista los requerimientos y en base a ellos presenta y describe los componentes internos del sistema desarrollado, junto con las tecnologías y recursos de software utilizados para su implementación. Asimismo, se realiza una descripción simplificada del funcionamiento de un EDFA.

### 2.1. Requerimientos

Los requerimientos fueron determinados en conjunto con la empresa en base a las funcionalidades y prestaciones con las que debe contar el sistema. Como la lista es muy extensa, se listan a continuación solamente algunos de los principales:

1. Encendido y apagado del EDFA
  - 1.1 El software debe apagar la salida óptica del dispositivo EDFA bajo prueba cuando se detecte la activación de alguna de las alarmas.
  - 1.2 Mediante la función táctil de la pantalla LCD el usuario debe poder prender y apagar la alimentación del dispositivo EDFA bajo prueba y su salida óptica.
  - 1.3 El software debe cortar la alimentación del dispositivo EDFA bajo prueba cuando se detecte que la corriente supera el valor previamente definido.
  - 1.4 El software debe medir y mostrar en pantalla los valores de tensión de alimentación y consumo de corriente del dispositivo EDFA bajo prueba mediante las señales analógicas de entrada provenientes de los respectivos monitores, con una precisión no menor al 10% (máximo desvío con respecto al valor real). Este valor debe ser de una cifra significativa para la parte entera y dos para la decimal.
2. Pantalla LCD
  - 2.1 El software debe actualizar la imagen de la pantalla cada medio segundo (2 cuadros por segundo).
  - 2.2 La pantalla deberá indicar el estado de la salida óptica del dispositivo EDFA bajo prueba y el del relé de alimentación.
  - 2.3 Mediante la función táctil de la pantalla LCD el usuario debe poder cambiar el valor para el cual se detecta una sobrecorriente. El rango válido para este valor debe ser de 0 A a 3 A, siendo la parte decimal de dos cifras significativas.

### 3. Entradas y salidas del EDFA

3.1 El software deberá mostrar en la pantalla los estados de todas las señales digitales de entrada y salida del dispositivo EDFA bajo prueba.

### 4. Requisitos de rendimiento

4.1 La apertura del relé de alimentación del dispositivo EDFA bajo prueba deberá efectuarse en un tiempo menor a 50 ms luego de detectarse una sobrecorriente o una caída de la tensión de alimentación.

4.2 El apagado de la salida óptica del dispositivo EDFA bajo prueba deberá efectuarse en un tiempo menor a 100 ms luego de detectarse la activación de una alarma.

La lista completa de requerimientos se puede ver en [5].

## 2.2. Funcionamiento de un amplificador óptico

Los amplificadores de fibra dopados con erbio son los amplificadores ópticos más importantes en el contexto de comunicaciones ópticas de larga distancia. Son utilizados en la banda L y C del espectro (aproximadamente entre 1530 nm y 1625 nm [6]), región en la que las pérdidas en la fibra óptica son menores. Esto se puede ver en la figura 2.1, que muestra las pérdidas en función de la longitud de onda de la luz utilizada.

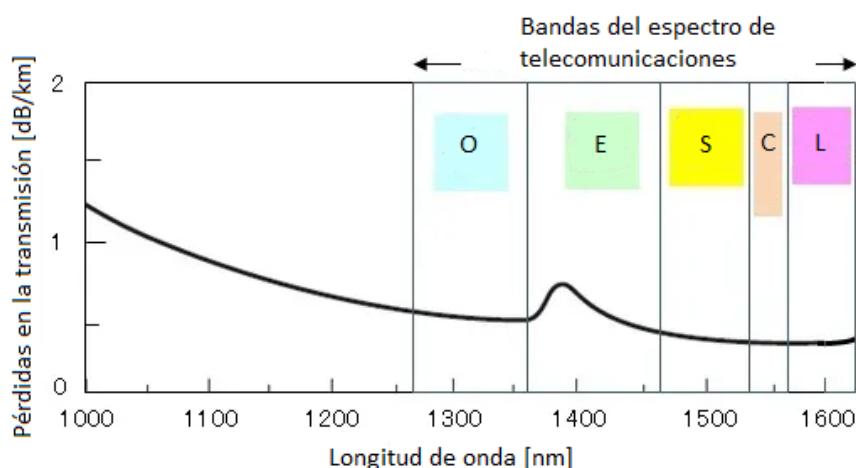


FIGURA 2.1. Pérdidas en la fibra<sup>1</sup>.

Inventado en 1987 [7], el EDFA es generalmente usado para compensar las pérdidas mencionadas en una línea de transmisión óptica. Este puede ser colocado en tres partes:

- Inmediatamente después del transmisor, para aumentar la potencia inyectada en la línea.
- En el medio del camino óptico, para compensar las pérdidas por la distancia recorrida.

<sup>1</sup>Imagen tomada de [6]

- Antes del receptor, para favorecer la detección de la señal.

La figura 2.2 muestra la configuración interna más común de un EDFA. Su componente principal es la fibra dopada con erbio (EDF por sus siglas en inglés), que generalmente es monomodo [8].

La luz ingresa al amplificador mediante el puerto de entrada y luego mediante un divisor se redirige un porcentaje de la señal (generalmente entre un 1 % y 2 % [4]) a un detector para realizar una medición de la potencia óptica. Luego, pasa por un aislador óptico que permite la transmisión de luz en un solo sentido y así evita una realimentación debida a reflexiones en etapas posteriores.

La bomba de 980 nm es un diodo láser que genera una señal lumínica en esa longitud de onda. Esta luego se mezcla con la señal de luz entrante (a la salida del aislador) y atraviesan la fibra dopada con erbio. En esta instancia es donde se genera la amplificación de la señal propiamente dicha, mediante un efecto denominado emisión estimulada [4][9].

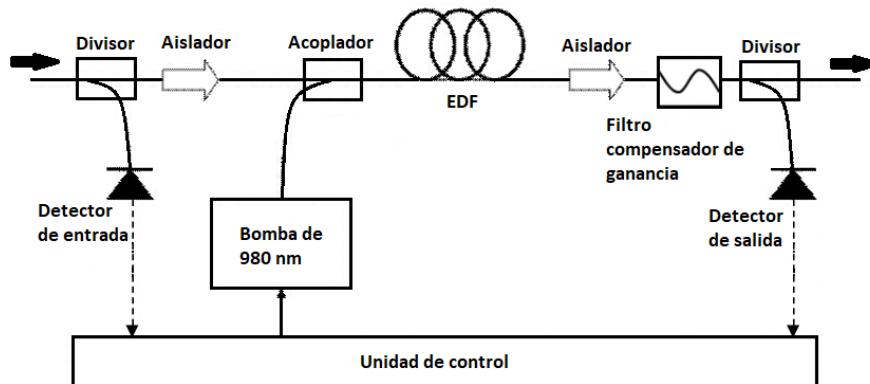


FIGURA 2.2. Configuración interna de un EDFA<sup>2</sup>.

Una vez amplificada, la señal pasa nuevamente por otro aislador que se encarga principalmente de filtrar la luz de 980 nm, evitando que se introduzca en el camino óptico. Luego, la señal pasa por un filtro aplano de ganancia (GFF por sus siglas en inglés) cuyo objetivo es hacer constante la ganancia del amplificador en todo el ancho de banda de trabajo [10] (bandas C y L del espectro). Finalmente, antes de que la luz salga del amplificador, se realiza nuevamente una medición de la potencia óptica al igual que en la entrada.

Tanto la bomba de 980 nm como ambos detectores de potencia se encuentran conectados a una unidad de control. Esta generalmente cuenta con un microcontrolador o chip dedicado que se encarga de regular la potencia que entrega la bomba en base a los valores medidos por los detectores, creando un lazo de control automático de ganancia [4][7].

## 2.3. Interfaz del amplificador óptico

Como se mencionó en la sección 1.1, para poder usar el EDFA este cuenta con un conector de 25 pines tipo Micro-D [11]. Este conector contiene varios grupos

<sup>2</sup>Imagen tomada de [4]

de señales con distintas funciones. La tabla 2.1 lista cada una de las señales de la interfaz, junto con los detalles de su dirección, tipo y función.

TABLA 2.1. Señales de la interfaz del EDFA.

Nombre de la señal	Dirección	Tipo	Función
5 V	Entrada	Potencia	Entrada de alimentación del EDFA.
PGND	Salida	Potencia	Retorno de alimentación (potencia).
GND	Salida	Tierra digital	Retorno de alimentación (digital).
IN_POW	Salida	Analógica	Indica el nivel de potencia óptica de entrada.
OUT_POW	Salida	Analógica	Indica el nivel de potencia óptica de salida.
CASE_TEMP_ALARM	Salida	Digital	Alarma de temperatura de la carcasa del EDFA.
PUMP_BIAS_ALARM	Salida	Digital	Alarma de la bomba de polarización.
OUT_POW_ALARM	Salida	Digital	Alarma de nivel de potencia de salida.
IN_POW_ALARM	Salida	Digital	Alarma de nivel de potencia de entrada.
EN/DIS	Entrada	Digital	Habilitación del amplificador.
RESET_uC	Entrada	Digital	Reset del microcontrolador del EDFA.
OUT_POW_MUTE	Entrada	Digital	Habilitación de la salida óptica.
UART_TX	Salida	Digital	Transmisor del UART interno del EDFA.
UART_RX	Entrada	Digital	Receptor del UART interno del EDFA.

A continuación, se provee una breve explicación de cada grupo de señales:

- **Alimentación:** tiene separada la tierra en digital, para la lógica y la comunicación, y la de potencia para la amplificación de la señal óptica.
- **Señales analógicas:** indican el nivel de potencia óptica de entrada y salida del EDFA.
- **Alarmas:** mediante un estado en alto indican si ocurrió alguno de los eventos que requieren la atención del usuario.
- **Señales de control:** controlan el funcionamiento de ciertos componentes del amplificador.
- **Comunicación UART:** permite el envío de comandos al EDFA y la consulta de valores de parámetros internos como temperaturas, potencias, ganancias, etc.

## 2.4. Componentes del sistema

Los componentes de hardware que conforman el sistema descrito en la sección 1.1 fueron seleccionados con el objetivo de cumplir con los requisitos, optimizando al mismo tiempo la cantidad de elementos utilizados. Así se logra mantener al sistema simple, con poca probabilidad de fallas, fácil de usar y probar. A continuación se presentan los principales componentes y sus características.

### 2.4.1. Microcontrolador

El modelo de microcontrolador utilizado es el STM32F429 [12], del fabricante ST y con arquitectura de procesador Arm Cortex-M. La principal razón por la que se decidió utilizar este modelo es porque es el que se encuentra integrado en la placa de desarrollo NUCLEO-144, utilizada durante la cursada de la especialización.

La placa de desarrollo NUCLEO-144 es ideal para ejecutar un RTOS debido a su alta velocidad de reloj, gran capacidad de memoria flash y variedad de periféricos, además de contar con la ventaja de tener integrada la interfaz de programación y depuración ST-LINK/V2 [13].

### 2.4.2. Monitor de corriente

El circuito integrado seleccionado para medir la corriente de alimentación del amplificador mientras está en funcionamiento es el INA301A3 [14].

Este chip provee en uno de sus pines una tensión analógica proporcional a la corriente que se está midiendo. Asimismo, cuenta con una salida digital que se activa cuando la corriente medida alcanza cierto nivel establecido mediante un resistor (pin ALERT) [14].

### 2.4.3. Pantalla táctil LCD

El modelo del módulo de la pantalla LCD utilizada en el trabajo es MSP2807 [15], que es una solución integrada, es decir, cuenta con toda la electrónica necesaria para poder hacer uso de la pantalla en su totalidad. Para esto cuenta con dos circuitos integrados: el controlador del display de la pantalla (lo que permite dibujar en ella) y el controlador de la función táctil (lo que permite detectar cuando se la toca). En la figura 2.3 se puede ver una imagen de la pantalla.



FIGURA 2.3. Pantalla LCD táctil MSP2807<sup>3</sup>.

El controlador del display utiliza una interfaz SPI para la recepción de datos y cuenta con una señal DC/RS para distinguir entre datos y registros internos. El controlador táctil también opera mediante SPI y dispone de una señal de interrupción T\_IRQ que indica cuando se ha producido una interacción en la pantalla [15].

## 2.5. Recursos de software

Para el firmware del microcontrolador se usaron distintas herramientas que permitieron el desarrollo de una estructura de software jerárquica, simple y eficaz.

### 2.5.1. Sistema operativo de tiempo real FreeRTOS

FreeRTOS es un sistema operativo de tiempo real (RTOS) diseñado para microcontroladores y microprocesadores pequeños. Se caracteriza por ser liviano, confiable y fácil de usar. Ofrece recursos como tareas, semáforos, *mutexes*, colas y gestores de memoria dinámica [16], lo que simplifica la creación y gestión de aplicaciones sobre el sistema operativo.

FreeRTOS se sitúa en una capa intermedia del firmware, por encima de los drivers del fabricante y debajo de la aplicación de usuario, brindando una capa de abstracción que automatiza la gestión de los recursos de hardware. Su principal ventaja es la capacidad de ejecutar múltiples tareas o procesos de manera simultánea. Esto permite estructurar aplicaciones en tareas independientes y sincronizadas [17].

### 2.5.2. Capa de abstracción de hardware (HAL)

Una *hardware abstraction layer* o HAL es un conjunto de rutinas de software que brinda acceso a recursos de hardware a programas de aplicación. Se ubica inmediatamente por arriba del hardware y por debajo del sistema operativo que se ejecuta.

La principal ventaja de esta capa es que oculta la arquitectura del hardware del *kernel* del sistema operativo. Así, el código del *kernel* no tiene que ser cambiado o reescrito para que pueda correr sobre sistemas con distinto hardware y las aplicaciones de software se vuelven independientes de la plataforma (portabilidad) [18].

La HAL de la serie de microcontroladores STM32 se denomina STM32Cube y tiene como objetivo asegurar la máxima portabilidad entre dispositivos de la misma familia y proveer APIs (*Application Programming Interface*) multi-instancias para todos los periféricos (UART, SPI, temporizadores, ADC, etc.). Estas APIs están listas para usar y facilitan la implementación de la aplicación de usuario. Por ejemplo, los periféricos de comunicación cuentan con APIs para inicializarlos y configurarlos, gestionar la transferencia de datos en modo *polling*, manejar las interrupciones, el DMA (*Direct Memory Access*) y los errores de comunicación [19].

---

<sup>3</sup>Imagen tomada de [15]

## 2.6. Periféricos utilizados

A excepción del monitor de corriente, el resto de los periféricos utilizados ya se encuentran integrados en el chip del microcontrolador, por lo que no hubo necesidad de agregar ningún hardware adicional.

### 2.6.1. Conversor analógico-digital (ADC)

Un conversor analógico-digital o ADC es un dispositivo que convierte una señal eléctrica analógica proveniente, por ejemplo, de un sensor a una señal digital. De esta forma su valor puede ser almacenado en un sistema digital, por lo que estará representada por un número binario [20].

### 2.6.2. Universal Asynchronous Receiver Transmitter (UART)

Un UART es un dispositivo utilizado para establecer una comunicación serie asíncrona, con formato de datos y velocidad de transmisión configurables. Consiste solo de dos señales que conectan dos dispositivos de forma bidireccional: una para la transmisión y otra para la recepción, comúnmente llamadas TX y RX respectivamente [21].

### 2.6.3. Serial Peripheral Interface (SPI)

SPI es una especificación de interfaz de comunicación serie sincrónica para distancias cortas. Es comúnmente utilizada para enviar datos entre un sistema embedido y pequeños periféricos como sensores y memorias SD.

Los dispositivos SPI se comunican en modo *full duplex* (ambos sentidos simultáneos) y cuentan con una señal de clock, dato entrante, dato saliente y de selección de esclavo, que conecta o desconecta la operación del dispositivo con el que uno desea comunicarse. De esta forma, este estándar permite multiplexar las líneas de clock y soportar arquitecturas multi-esclavo [22].



## Capítulo 3

# Diseño e implementación

En este capítulo se realiza una explicación detallada del diseño del hardware, cómo se utilizó cada componente principal y su conexión con el microcontrolador. Asimismo, se especifica la estructura general del firmware en el marco de FreeRTOS y las principales rutinas de bajo nivel.

### 3.1. Arquitectura de hardware

La arquitectura general de la placa fue diseñada para tener como unidad de procesamiento principal al microcontrolador STM32 de la placa NUCLEO-144 y a la pantalla LCD como periférico principal de salida de datos.

La figura 3.1 muestra un diagrama completo de todo el hardware de la placa fabricada, sus entradas, salidas y el conexionado entre sus distintas partes. También incluye los periféricos utilizados en el microcontrolador y la cantidad de líneas de cada conexión.

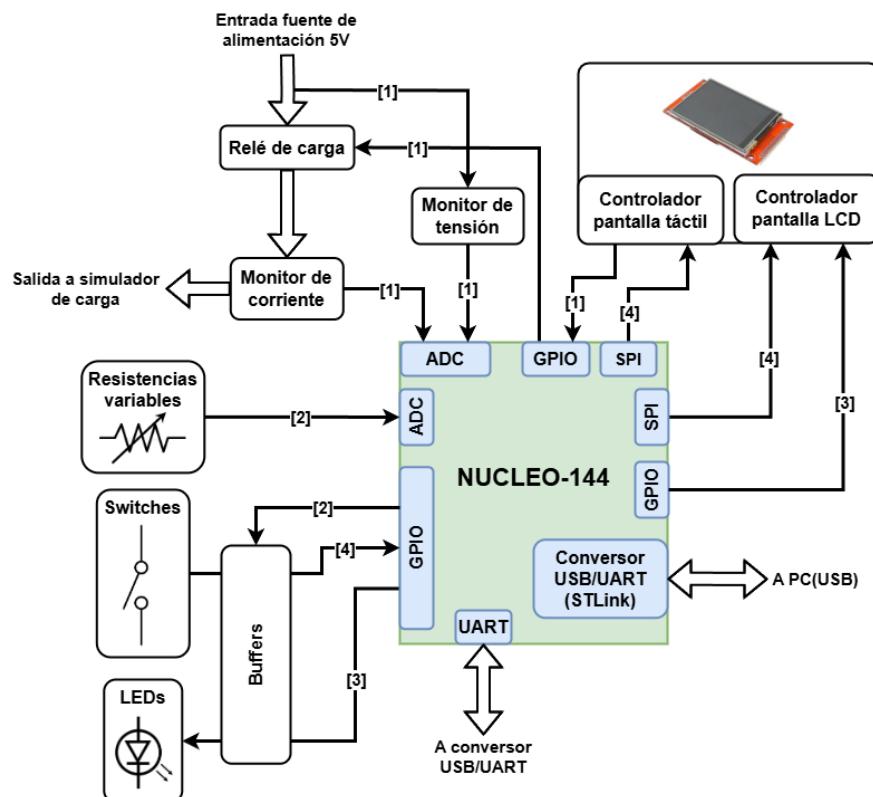


FIGURA 3.1. Diagrama completo del hardware implementado.

Todos los bloques externos a la placa NUCLEO que se ven en el diagrama, a excepción de la pantalla LCD, fueron diseñados con componentes electrónicos básicos, es decir, sin usar módulos funcionales compactos.

Los bloques internos de la placa NUCLEO que se ven en el diagrama, a excepción del conversor USB-UART, son periféricos que se encuentran embebidos en el mismo chip del microcontrolador. El conversor USB-UART, denominado STLink, se encuentra fuera de este y esta compuesto por varios componentes electrónicos.

En el apéndice A se encuentra el diseño esquemático completo de la placa.

### 3.1.1. Medición de corriente del amplificador

El principio de funcionamiento del monitor de corriente se basa en un resistor *shunt*, el cual se coloca sobre la alimentación, en serie con la línea a la que se desea conocer el consumo de corriente. Para que este resistor no disipe mucha potencia y disminuya la tensión de la línea, su valor de resistencia es de  $10\text{ m}\Omega$ . Mientras menos comparable sea su valor al de la carga más despreciables serán los efectos negativos que introducirá.

Sobre este resistor cae una tensión proporcional a la corriente consumida por el EDFA. Como el valor máximo de corriente consumida por el amplificador se encuentra cerca de los 2,5 A, sobre el resistor caen como máximo 25 mV, lo que frente a los 5 V de la alimentación no representa un valor significativo. Luego de sensarla, el chip amplifica 100 veces esta tensión diferencial y la conecta a uno de sus pines de salida.

Al ser este un valor analógico, para poder ser interpretado se lo debe convertir a digital. Para esto se usa el ADC incorporado en el microcontrolador, conectando a uno de sus pines la señal de salida del monitor de corriente. La tensión máxima que puede alcanzar la señal es de aproximadamente 2,5 V, lo que se encuentra por debajo de la tensión de referencia del ADC. Un esquema de esta conexión se puede ver en la figura 3.2.

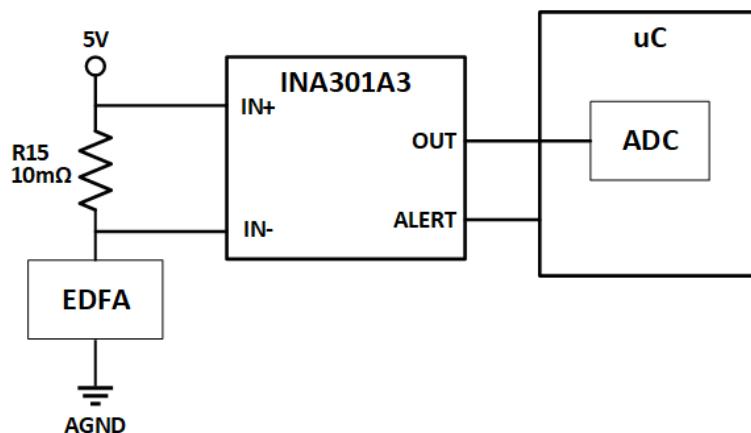


FIGURA 3.2. Esquema del monitor de corriente.

Por otro lado, el INA301A3 cuenta con un pin de salida denominado ALERT que se activa cuando la corriente medida alcanza cierto valor, lo que funciona a modo de alarma. Este nivel se determina conectando un resistor de cierto valor en el pin LIMIT.

### 3.1.2. Relé de alimentación

La línea de 5 V de alimentación del EDFA es controlada mediante la activación de un relé (código APAN3103) [23], lo que significa que la conexión y desconexión es mecánica. Esto se decidió implementarlo así debido a los lineamientos de protecciones necesarios para el hardware de vuelo.

Como el relé necesita de aproximadamente 36,7 mA para activarse, no se puede conectar directamente a un pin de salida del microcontrolador ya que este no puede suministrar la corriente necesaria. Por esta razón la activación se realiza mediante un transistor (código MMBT2222) [24], funcionando como una llave (en modo saturación o corte). De esta forma, la corriente para la activación de la bobina del relé la provee directamente la fuente de alimentación de 3,3 V.

### 3.1.3. Medición de tensión del amplificador

Hacer una medición de la tensión de la línea es mucho más sencillo que hacer una medición de la corriente, ya que esta se puede conectar directamente a un ADC si previamente se la atenúa o amplifica (dependiendo del caso).

En este caso, como la tensión de alimentación del EDFA es 5 V y la tensión de referencia del ADC del microcontrolador es 3,3 V, se la debe atenuar de forma que esta se encuentre dentro del fondo de escala del ADC.

Considerando que se desea medir una sobretensión de por lo menos 1 V en la alimentación, se opta por atenuar la tensión del EDFA a la mitad, resultando en un fondo de escala de 6,6 V.

Esta técnica permite medir tensiones mayores a la de referencia del ADC pero trae como desventaja que se pierde precisión debido al ruido y la tolerancia de los resistores.

En la figura 3.3 se puede ver el circuito implementado.

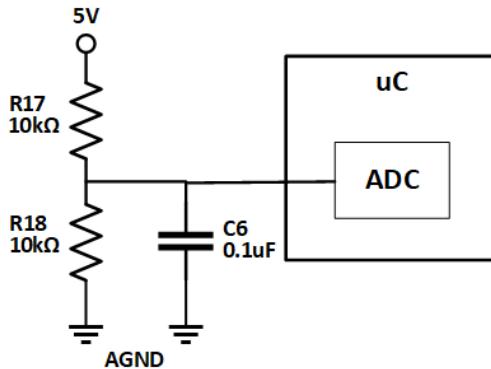


FIGURA 3.3. Circuito del monitor de corriente.

El capacitor C6 forma un filtro RC [25] para eliminar el ruido de alta frecuencia que pueda tener la línea.

### 3.1.4. Conexión a pantalla LCD

Como se puede ver en la figura 3.1, la pantalla LCD tiene dos controladores con SPI: uno para el control de la pantalla y otro para el control de la función táctil.

Si bien ambas interfaces se podrían haber conectado al mismo periférico SPI del microcontrolador en modo multi-esclavo, se decidió usar uno para cada uno de forma individual. La principal razón de esto es porque facilita la implementación del firmware ya que permite usar ambos controladores de forma independiente y a distintas velocidades sin necesidad de reconfiguración.

El controlador de la función táctil tiene, además de las señales del bus SPI, una señal denominada IRQ que se activa cuando se detecta que la pantalla está siendo tocada. Esta señal sirve para ser usada como interrupción en el programa del microcontrolador.

Por otro lado, el controlador del display también tiene señales con funciones especiales. Estas son:

- RESET: resetea el chip del controlador y, por lo tanto, toda la configuración del display y la información de cada píxel.
- DC/RS: indica al controlador si la información que se está enviando por el SPI es un dato o un registro, dependiendo del nivel de la señal.
- LED: permite controlar la intensidad de luz de la pantalla (denominada *bac-klight*).

### 3.1.5. Entradas y salidas digitales

En la tabla 2.1 se detallan las señales de entrada y salida digitales del puerto del EDFA. Las salidas son señales de alarmas y las de entrada, de control.

Estas señales se podrían conectar directamente a pines digitales del microcontrolador ya que son de tipo TTL 3,3 V, pero se decidió poner antes de este un *buffer* de tres estados (código SN74LVC2244A) [26]. Los objetivos de esta decisión son tres:

- Generar niveles de tensión mejor definidos. Esto filtra ruido y rebote en las señales y facilita la detección en ambos extremos (EDFA y microcontrolador).
- Proveer a la electrónica interna del EDFA de protección contra descargas electrostáticas (ESD).
- Suministrar la corriente necesaria a las señales de entrada del amplificador y así evitar que esta deba ser proporcionada por el microcontrolador.

Para poder simular la activación de las alarmas en la placa se colocaron pulsadores con testigo lumínico (código TL1265YQSCLR) [27] en el extremo donde estaría el EDFA, con el propósito de activarlos manualmente durante el testeo del firmware.

Para las señales de control se utilizaron LEDs que indican el estado de activación de estas (código AA3528LVBS/D) [28].

### 3.1.6. Indicadores de potencia óptica

Las señales IN\_POW y OUT\_POW son analógicas y pueden tomar cualquier valor en el rango de 0 V a 3,3 V. Este valor es proporcional a la potencia óptica

medida por los detectores de entrada y salida del amplificador, el cual corresponde al rango de -10 dBm a 10 dBm para el de entrada y 5 dBm a 37 dBm para el de salida.

Para poder simular estas señales se utilizaron dos resistores variables (código PTV09A-4020U-B103) [29] que permiten variar de forma manual el valor de tensión entre 0 V y 3,3 V.

### 3.1.7. Interfaz UART

Como se puede ver en la figura 3.1, el dispositivo debe contar con dos interfaces UART: una para la comunicación entre el tester y el EDFA y otra para la comunicación entre el tester y la PC. Para esto se requiere hacer uso de dos de los controladores UART internos que posee el microcontrolador.

## 3.2. Arquitectura de firmware

Como se mencionó anteriormente, la arquitectura del firmware del trabajo tiene como base la utilización de FreeRTOS y los recursos que ofrece. Esto permite la implementación de un diseño eficaz y sencillo mediante la división en dos partes bien definidas: la parte de procesamiento de datos y control (o backend) y la parte gráfica que interactúa con el usuario (o frontend). Para implementar cada una de ellas se utilizó una tarea o proceso, recurso provisto por el sistema operativo. Como la velocidad de reloj del microcontrolador es alta, se puede asumir que estas se ejecutan de forma independiente y simultánea. Por otro lado, estas comparten ciertas variables cuyo acceso debe ser controlado para que no existan colisiones de lectura o escritura. Este acceso es gestionado por un semáforo.

El proceso del backend tiene como tarea principal ejecutar una máquina de estados y se encarga de controlar el estado general del dispositivo en todo momento. Esto significa que lee los datos de entrada, los procesa y en base a ellos toma decisiones que terminan por afectar las señales de salida y en última instancia a la tarea del display. A demás, ejecuta una consola de control que le permite al usuario interactuar con el tester e indirectamente con el EDFA.

Por otro lado, el proceso del frontend esta asociado al display LCD y tiene como objetivo principal la interacción directa con el usuario. Para ello, debe leer la información de salida del backend, presentarla en la pantalla y, a su vez, detectar cuándo y dónde el usuario presiona sobre ella y enviar estos datos al backend mediante recursos compartidos.

En la figura 3.4 se puede ver un diagrama de cómo se relacionan ambas tareas entre sí y con el hardware.

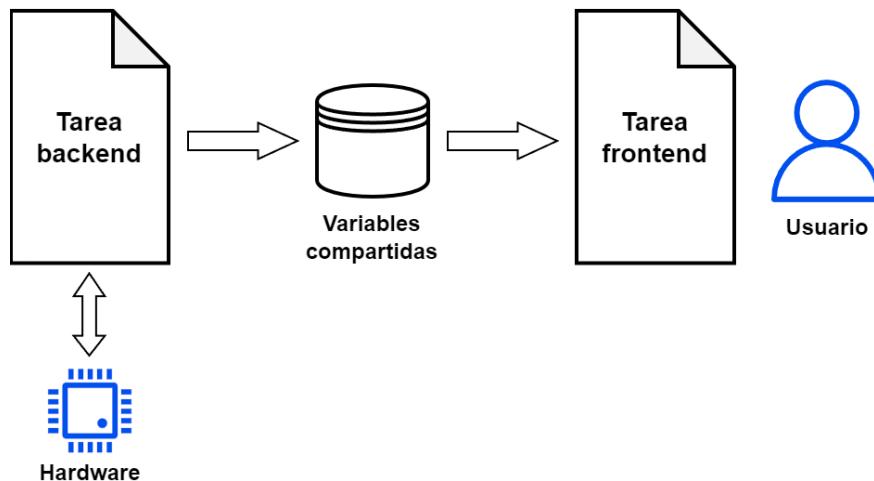


FIGURA 3.4. Arquitectura de firmware.

### 3.2.1. Tarea del backend

El proceso del backend ejecuta las funciones más importantes del programa de forma continua. Estas son la ejecución de la máquina de estados, la toma de las mediciones de las variables, la administración de la consola de control y el envío de datos hacia el frontend.

Luego del inicio del programa por primera vez y una vez que todo el hardware se ha configurado, la tarea del backend se crea y comienza a ejecutarse sobre el sistema operativo. Lo primero que esta hace es inicializar todas las variables que usará e iniciar la máquina de estados. En este momento se asigna el estado *Idle* como el inicial.

Luego comienza la ejecución de la parte cíclica de la tarea. Primero se llama al generador de eventos de la máquina de estados, que puede devolver cualquiera de los eventos válidos para el programa. Este puede tener como origen la consola de control, la activación de alguna alarma, algún valor fuera de rango, etc.

Una vez que se tiene el evento se lo ingresa a la máquina de estados, se ejecuta la función correspondiente y se actualiza el estado. Luego se toman las mediciones de todas las variables de interés y se lee el estado de las alarmas del EDFA. Teniendo los datos recién adquiridos se toma el semáforo para escribir estos en la variable compartida con el frontend.

Por último, se espera a que hayan transcurrido 20 milisegundos desde la medición anterior antes de volver a ejecutar el lazo. En la figura 3.5 se puede ver un diagrama de flujo de la tarea del backend.

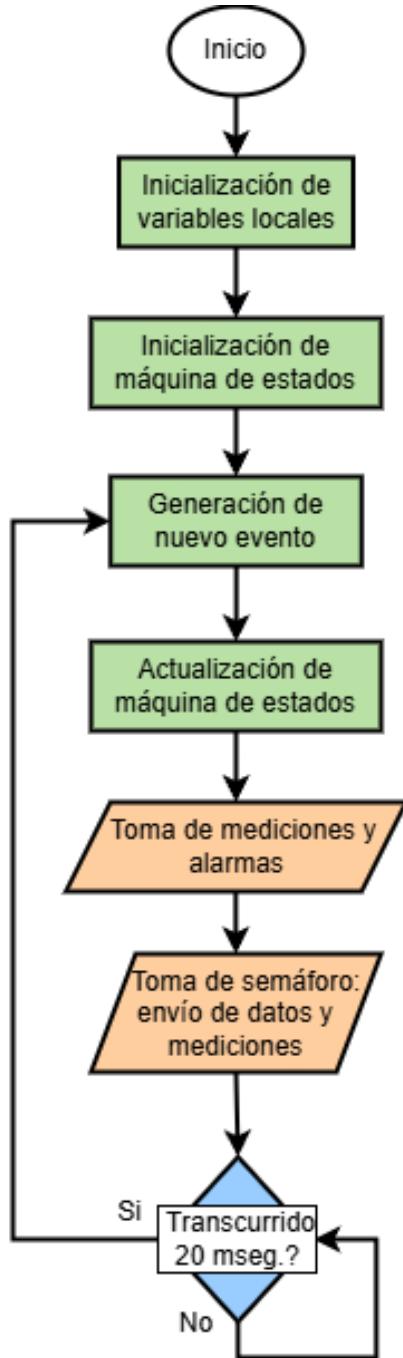


FIGURA 3.5. Diagrama de flujo de la tarea del backend.

### 3.2.2. Tarea del frontend

El proceso del frontend ejecuta de forma cíclica el refresco de la pantalla LCD, es decir, dibuja la pantalla en su totalidad coloreando los píxeles. A bajo nivel, esto significa escribir el registro de la memoria asociado a cada píxel con un valor que corresponde a un color determinado.

Debido a la cantidad de píxeles de la pantalla LCD y el tiempo que le toma al programa actualizarla en su totalidad, se decidió actualizarla a una tasa de diez veces por segundo (cada 0,1 seg.). Teniendo en cuenta que en esta aplicación los

eventos visibles ocurren a baja velocidad, aumentar la tasa de muestreo no aportaría una mejora significativa. En cuanto a las variables medidas, para poder ser legibles, sus valores se actualizan solamente dos veces por segundo.

El proceso de refresco comienza cuando la tarea toma el semáforo que le permite acceder a la variable compartida con el proceso de backend. El frontend copia los valores localmente y libera el semáforo. Luego realiza la actualización de toda la pantalla con los valores obtenidos, a excepción de las mediciones analógicas.

A continuación el programa verifica si ya transcurrió medio segundo desde la última actualización de los valores analógicos. En caso afirmativo, realiza el mismo proceso que antes pero con los valores de las mediciones. Si no pasó dicho tiempo, continúa y espera que trascurra 0,1 segundo antes de realizar la próxima actualización de la pantalla.

En la figura 3.6 se puede ver un diagrama de flujo general de la tarea de frontend.

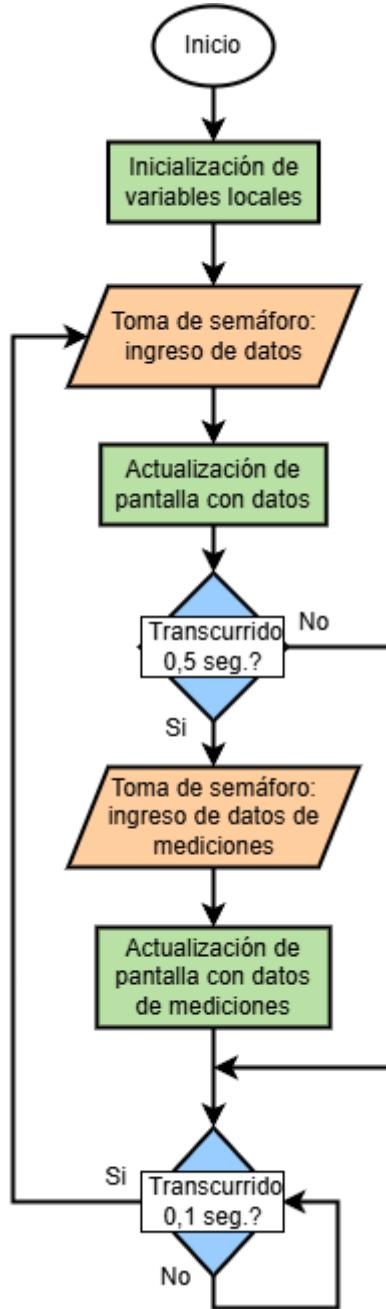


FIGURA 3.6. Diagrama de flujo de la tarea del frontend.

### 3.2.3. Máquina de estados

#### Diseño

La máquina de estados implementada administra el funcionamiento del amplificador en todo momento mediante su interfaz eléctrica y el control de la alimentación. Cada uno de los estados de la máquina describen de forma análoga el estado real del EDFA. En base a los requerimientos se han definido los siguientes:

- Desconectado: alimentación del amplificador desconectada (relé desactivado y valor de tensión válido).
- Conectado: amplificador energizado (relé activado).

- Activo: salida óptica del amplificador encendida.
- Alimentación fuera de rango: valor de la alimentación del EDFA fuera del rango permitido.
- Alarma encendida: alarma del EDFA encendida (cualquiera de ellas).

Todas las posibles transiciones entre estos estados son disparadas por eventos que pueden ocurrir en cualquier momento, y ejecutan una acción según el estado en que se encuentre la máquina. La cantidad de eventos que pueden llegar a la máquina es elevada, por lo que, por simplicidad solo se mencionan algunos a modo de ejemplo:

- Eventos de la alimentación: baja tensión, sobretensión y sobrecorriente.
- Eventos de la pantalla táctil: botón de encendido presionado, botón de apagado presionado, etc.
- Eventos del amplificador: alarma de potencia de entrada encendida, potencia óptica de salida excedida, etc.
- Eventos de la consola de control: comando de conexión de alimentación recibido, comando de encendido de salida del amplificador recibido, comando inválido, etc.

En la figura 3.7 se puede ver un diagrama simplificado de transiciones de la máquina de estados implementada, que incluye sus estados y eventos asociados.

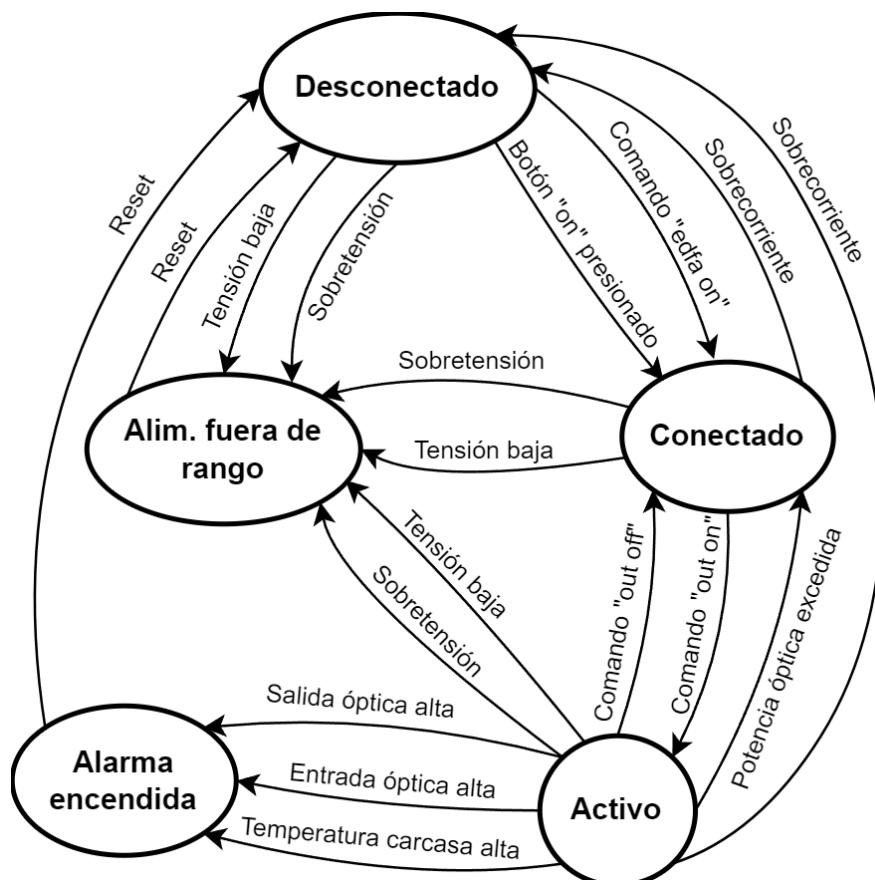


FIGURA 3.7. Diagrama simplificado de la máquina de estados.

### Implementación

Para implementar de forma conveniente la máquina de estados lo que se hizo fue usar el tipo de dato que se ve en la figura 3.8.

```
typedef struct tablecell_t
{
    struct tablecell_t * nextState;
    FSMEvent_t event;
    void (*action)(appData_t *);
} FSMTableCell_t;
```

FIGURA 3.8. Tipo de dato de la máquina de estados.

Si se interpreta a la máquina de estados como una tabla de doble entrada, donde los eventos se colocan en las columnas y los estados en las filas, cada intersección representa la función a ejecutar cuando se da una transición. Este tipo de dato contiene tres variables y representaría una celda de dicha tabla:

- **nextState**: puntero a otra variable del mismo tipo. Indica el próximo estado de la máquina luego de ejecutar la función.
- **event**: evento definido en una lista con todos posibles. Indica bajo qué evento se ejecutará la función apuntada por esta celda.
- **action**: puntero a función. Apunta a la función a ejecutar propiamente dicha.

Por lo tanto, si se define un arreglo de este tipo de dato para cada uno de los estados de la máquina y se lo llena con la cantidad de elementos correspondientes a cada posible transición, se cuenta con una representación completa de toda la máquina.

#### 3.2.4. Capa de abstracción del amplificador

Con el objetivo de generar una estructura jerárquica, se decidió crear una capa adicional que se ubica por encima de los drivers de la pantalla LCD, de la pantalla táctil, de la UART, de las señales analógicas y de los GPIO provistos por la HAL del fabricante (ST). A su vez, los drivers de la pantalla LCD y la pantalla táctil se implementan con base en el driver de SPI, ya que debe hacer uso de las funciones de bajo nivel para el envío y recepción de datos y comandos. Para el ADC también se creó una pequeña capa superior que permite obtener todos los valores de las señales analógicas en formato de punto flotante.

De esta forma se genera un nivel más de abstracción que provee un conjunto de funciones o servicios para interactuar con el EDFA, funcionando a modo de puente entre este y la aplicación de usuario (backend). En la figura 3.9 se puede ver la estructura de esta capa junto con sus distintos niveles.

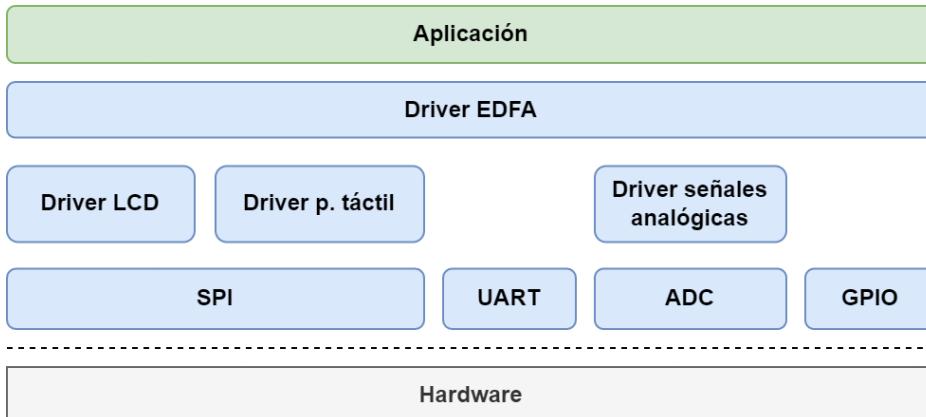


FIGURA 3.9. Capa de abstracción del EDFA.

### 3.2.5. Driver del ADC

Para la lectura de los valores analógicos se decidió utilizar el driver del ADC mediante DMA. Esto trae la principal ventaja de que solo se necesita realizar la configuración del ADC una sola vez al comienzo del programa y luego el muestreo y la conversión de los valores se realizará automáticamente de forma periódica.

En este caso esta implementación es particularmente útil debido a que se evita ocupar al procesador realizando la conversión manualmente y se puede atender a otras tareas. De esta forma, como el movimiento del dato convertido desde el periférico a la memoria se hace de forma automática, cuando el programa requiere leer el dato del valor convertido este ya se encuentra disponible en memoria.

Para filtrar ruido y posibles valores espurios que tome la señal analógica se aplica un pequeño filtro promediando 10 muestras continuas en cada canal usado. La configuración completa del ADC se muestra en la tabla 3.1.

TABLA 3.1. Configuración del ADC.

Parámetro	Configuración
Cantidad de canales usados	4
Cantidad de bits	12 bits
Modo de conversión	Continua
Tasa de muestreo	600 kspS
Cantidad de muestras promediadas	10
Disparo de conversión	Por software

Dado que el ADC opera con 12 bits y la tensión de referencia del microcontrolador es de 3,3 V, se obtienen 4096 niveles discretos (de 0 a 4095), lo que resulta en una resolución de aproximadamente 0,8 mV por cada bit menos significativo.

### 3.2.6. Driver de la pantalla LCD

Como se mencionó en la sección 3.1.4, cada bus SPI (LCD y pantalla táctil) tiene una interfaz dedicada en el microcontrolador funcionando este en modo maestro. La tabla 3.2 muestra un resumen de la configuración de ambos controladores.

TABLA 3.2. Configuración de los SPI del LCD.

Parámetro	LCD	Pantalla táctil
Velocidad de transmisión	8 Mbps	4 Mbps
Ancho de palabra	8 bits	8 bits
Polaridad	Baja	Baja
Dirección	Bidireccional	Bidireccional

Como se mencionó en la sección 3.2.4, este driver está implementado sobre el de SPI. A su vez, las funciones que contiene también tienen cierta jerarquía entre si, permitiendo hacer funciones complejas usando otras más básicas. Tomando como ejemplo la función para dibujar un cuadrado sin relleno de color se tienen, de mayor a menor abstracción, las siguientes funciones:

- **LCD\_Draw\_Hollow\_Rectangle\_Coord**: recibe 2 pares de coordenadas para los vértices y el color. Llama 2 veces a la función que dibuja una línea horizontal y dos veces a la que dibuja una línea vertical, con la coordenada de origen y el largo correspondiente.
- **LCD\_Draw\_Horizontal\_Line**: recibe la coordenada de origen, el largo y el color de la línea. Llama a la función que define la sección de la memoria del chip asociada a los píxeles a dibujar. Luego llama a la función que transmite el color a esta sección de memoria.
- **LCD\_Draw\_Colour\_Burst**: transmite el color de los píxeles llamando a la función de transmisión de datos del SPI.
- **LCD\_Set\_Address**: envía los comandos y direcciones de registros para definir la zona de memoria a escribir.
- **LCD\_Write\_Data**: envía el dato recibido llamando a la función de transmisión de datos del SPI. Pone la señal DC del chip del LCD en alto, indicando que el valor transmitido es un dato.
- **LCD\_Write\_Command**: envía el comando recibido llamando a la función de transmisión de datos del SPI. Pone la señal DC del chip del LCD en bajo, indicando que el valor transmitido es un comando.

### 3.2.7. Driver de UART

El controlador de UART para ambas interfaces se configura de modo que genere una interrupción en el programa luego de recibir una cadena de caracteres. Esto funciona de forma que los caracteres que se reciben se guardan continuamente en un buffer de entrada hasta que la línea de datos pasa a estar inactiva. Luego se atiende la interrupción y en ella se copia al programa principal la cadena de caracteres del buffer para ser analizada.

En la tabla 3.3 se puede ver la configuración de ambas interfaces UART.

TABLA 3.3. Configuración de la interfaz UART.

Parámetro	Configuración
Velocidad de transmisión	115200 baudios
Ancho de palabra	8 bits
Bits de stop	1 bit
Paridad	Ninguna
Control de hardware	Ninguno

### 3.2.8. Consola de control

La consola de control brinda una pequeña interfaz que funciona como intermediario entre el usuario y el amplificador, permitiéndole consultar y controlar su estado en todo momento. También se utiliza para ingresar valores al dispositivo, tales como los niveles de tensión y corriente máximos admitidos.

Las funciones de la consola se ejecutan en la tarea del backend, puede inyectar eventos a la máquina de estados y utiliza el bus USB-UART para el flujo de caracteres. Su funcionamiento es similar al de cualquier consola o línea de comando y se basa en el ingreso de comandos en formato instrucción/argumento.

Su funcionamiento básico es el siguiente: luego de recibir una cadena de caracteres por UART y recuperarla en el backend, como se explicó en la subsección anterior, esta se pasa por lo que se denomina un *parser*. Este lo que hace es primero identificar los caracteres de espacio que contiene la cadena para delimitar y extraer los comandos y argumentos (subcadenas). A continuación, se verifica si el comando ingresado coincide con alguno de los comandos válidos mediante la comparación de cadenas. En caso que este sea válido, se pasa a evaluar el argumento en base al tipo de dato que espera el comando. Por ejemplo, si el comando espera un número positivo se verifica que la cadena no contenga letras, que el valor sea positivo y que se encuentre dentro de los valores máximos y mínimos.

Si ambos comando y argumento son válidos entonces el programa generará el evento correspondiente a ese comando para luego ingresar a la máquina de estados. En caso que alguno no lo sea, ya sea por error de sintaxis o de argumento fuera de rango, no se generará ningún evento y la consola imprimirá "Comando inválido".

En la figura 3.10 se puede ver el diagrama de flujo desde que se recibe la cadena de caracteres en el buffer de la UART hasta la generación del evento correspondiente.

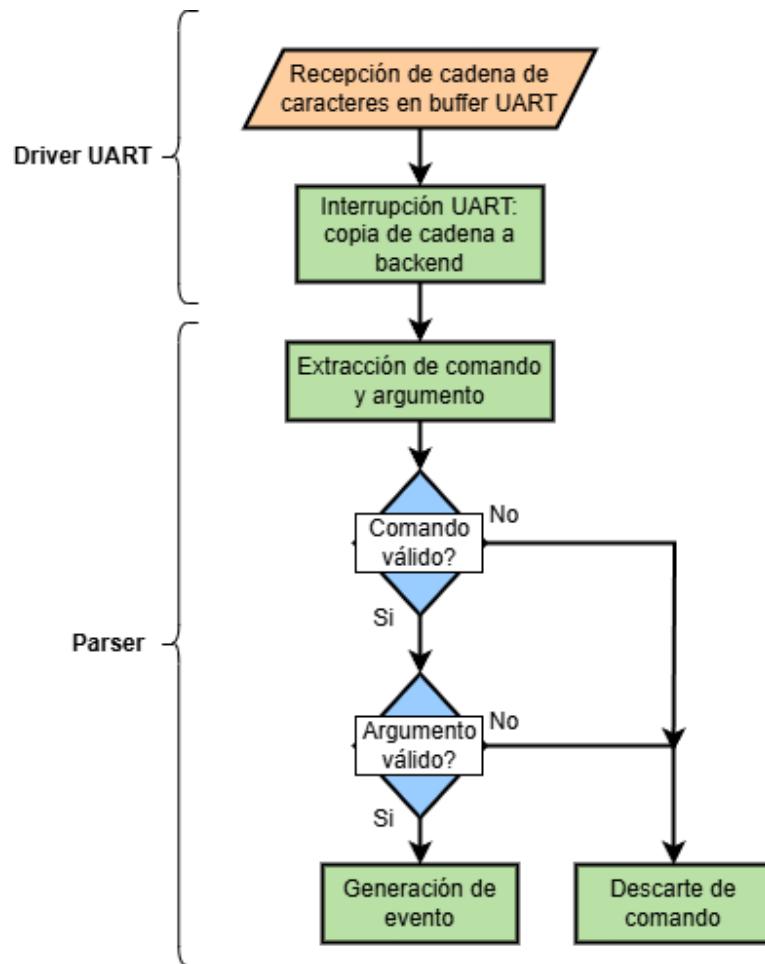


FIGURA 3.10. Diagrama de flujo de la consola de control.

La tabla 3.4 lista todos los pares comando/argumento que puede recibir la consola.

TABLA 3.4. Comandos/argumentos de la consola.

Comando	Argumento	Descripción
edfa	on	Conecta la alimentación del EDFA.
edfa	off	Desconecta la alimentación del EDFA.
out	on	Prende la salida óptica del EDFA.
out	off	Apaga la salida óptica del EDFA.
status	-	Indica el estado general del EDFA y los valores de las variables.
reset	-	Envía la señal de reset al microcontrolador del EDFA.
ilim	(valor)	Establece el límite de corriente al valor del argumento.
vul	(valor)	Establece el límite superior de tensión al valor del argumento.
vll	(valor)	Establece el límite inferior de tensión al valor del argumento.
oipl	(valor)	Establece el límite de potencia de entrada al valor del argumento.
oopl	(valor)	Establece el límite de potencia de salida al valor del argumento.

### 3.3. Dispositivo implementado

Es importante aclarar que el dispositivo que se desarrolló e integró como solución a lo planteado en el capítulo 1 no es el producto final destinado a ser utilizado. La placa construida se puede interpretar como un prototipo de desarrollo o una versión de ingeniería, ya que cumple con dos objetivos principales: verificar el funcionamiento del hardware diseñado y desarrollar el firmware que se ejecutará en el microcontrolador.

La versión armada para este trabajo, incluye los mismos componentes previstos para el producto final y, además, incorpora el hardware necesario para emular la interfaz electrónica del modelo del amplificador óptico, lo que permite generar todas las señales del conector, tanto analógicas como digitales.

Una vez alcanzados estos objetivos, la siguiente etapa consiste en desarrollar la placa del producto final, con el factor de forma adecuado para integrar y montar todos los componentes en una carcasa transportable.

El PCB fue enviado a fabricar a la empresa china JLCPCB y consta de 4 capas: dos planos internos de referencia conectados a 3,3 V y GND y dos capas de señal (*Top* y *Bottom*). En la figura 3.11a y 3.11b se pueden ver dos fotos del PCB ensamblado sin el display ni la placa NUCLEO conectados (lado superior y lado inferior respectivamente). Por último, en la figura 3.11c se puede ver la placa con ambos módulos conectados.

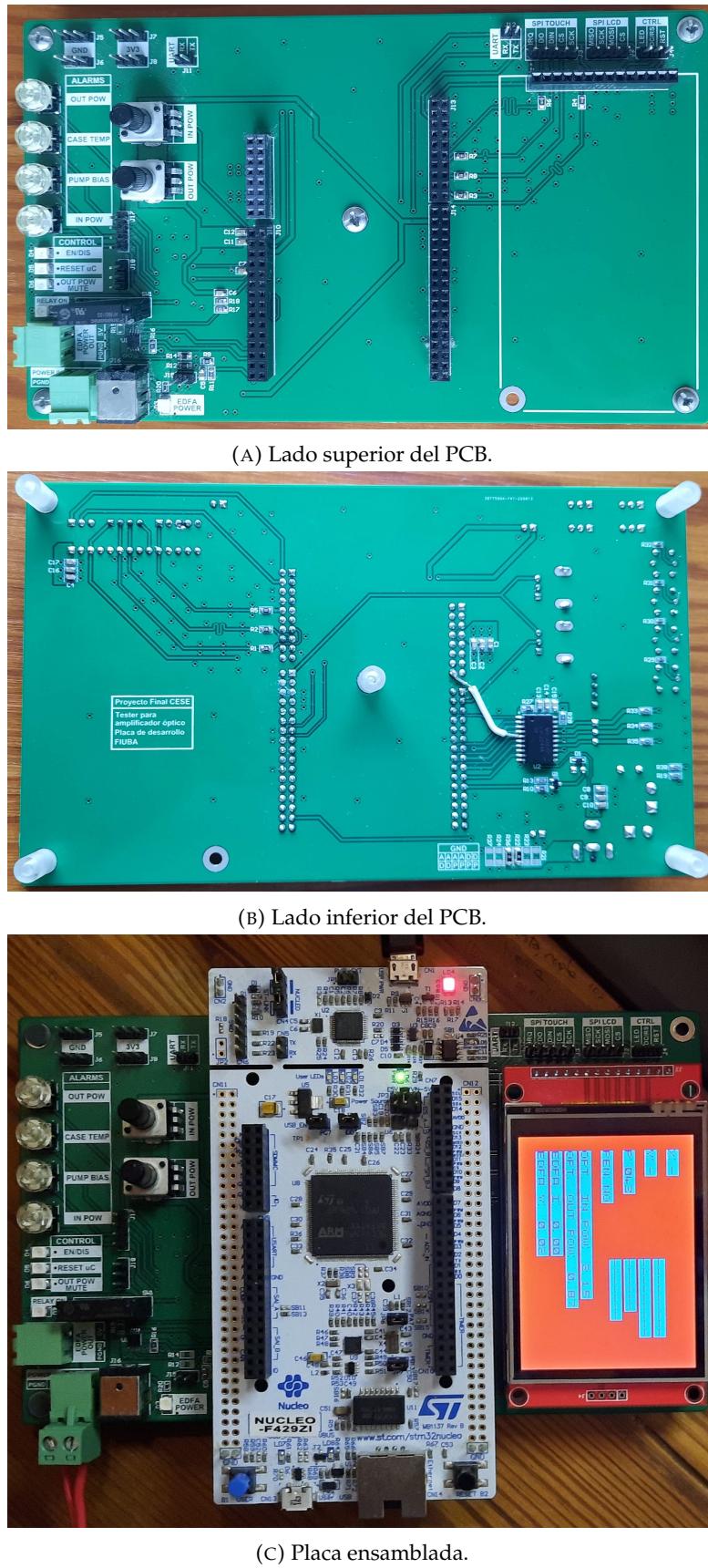


FIGURA 3.11. Placa fabricada para el trabajo.



## Capítulo 4

# Ensayos y resultados

En este capítulo se explica cómo se llevaron a cabo las pruebas de validación de hardware y software a las que fue sometido el dispositivo, los bancos de ensayos, instrumentos utilizados y los resultados obtenidos.

### 4.1. Instrumental utilizado

Para poder ejecutar las pruebas de integración de hardware y software se requirió del uso de una variedad de instrumentos y componentes en los bancos de ensayos. La tabla 4.1 lista los detalles de cada uno de ellos.

TABLA 4.1. Lista de instrumental utilizado.

Item	Modelo	Descripción
Multímetro	UNI-T UT61A	Multímetro digital autorango de alta precisión.
Fuente de tensión	KM GP-300ATX	Fuente de computadora de 400 W.
Resistencia de potencia variable	AVT05006E25R00KE	Resistencia de potencia de alambre bobinado de 25 Ohm / 50 W.
Potenciómetro	3590S-1-201L	Potenciómetro de 200 Ohm / 2 W y 10 vueltas para panel.
Conversor USB a UART	EM7-6043	Conversor USB a serie TTL CH-340.
Cables varios	-	Cables dupont y unipolar de 2,5 mm <sup>2</sup> .
PC	-	Computadora de escritorio con software PuTTY (cliente de terminal).

En el apéndice B se puede ver una foto del banco de ensayos completo con todos los instrumentos y hardware utilizado.

### 4.2. Pruebas de hardware

Para validar el funcionamiento del hardware, basta con medir continuidad o niveles de tensión en ciertos puntos de la placa. Por esta razón, se omite especificar las pruebas a excepción de las del monitor de corriente, ya que es el único caso en el que es necesario realizar mediciones.

#### 4.2.1. Prueba del monitor de corriente

En el circuito del monitor de corriente lo que se desea verificar es que a la salida del amplificador se obtenga una tensión proporcional al valor de corriente que circula. La constante de conversión surge de multiplicar el valor de la resistencia de sensado (10 mOhm) por la constante de amplificación del circuito integrado (100 V/V), lo que resulta en un valor de 1 V/A. Por lo tanto, la conversión entre corriente y tensión es unitaria.

Para realizar las mediciones se conecta la entrada de tensión a la fuente de 5 V, una resistencia de potencia en la salida de tensión y se varía su valor de forma de hacer circular desde 0 A hasta aproximadamente 3,3 A.

Con los datos obtenidos se calcula para cada medición el error porcentual con respecto al valor teórico de la constante de conversión. En la figura 4.1 se pueden ver los valores computados para distintos valores de corriente en la carga.

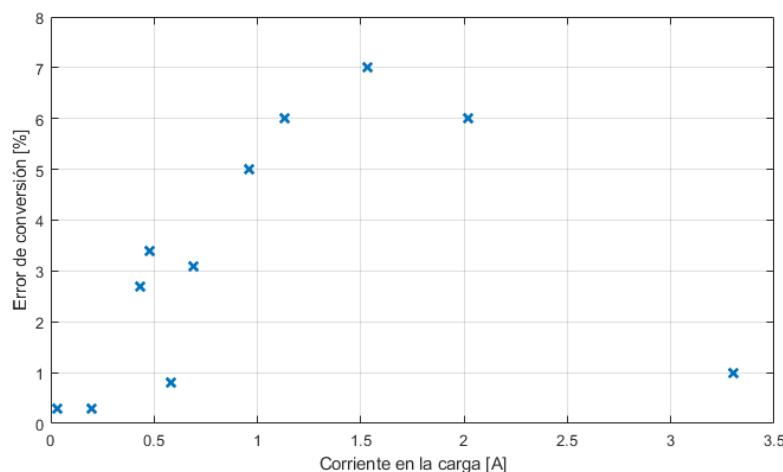


FIGURA 4.1. Error porcentual de la constante de conversión de corriente.

Del gráfico se observa que en este rango se tiene un error de medición máximo del 7 % y un valor promedio aproximado del 3,23 % respecto del valor real.

Esta diferencia entre el valor de la constante medida y el teórico se la puede atribuir principalmente a dos efectos: la dispersión del valor de la resistencia de sensado y a la presencia de ruido en el circuito. El error de la ganancia del circuito integrado está especificado en un máximo de 0,2 % por lo que no aporta variaciones significativas.

#### 4.3. Pruebas de firmware

Las pruebas de firmware tienen como finalidad validar el funcionamiento individual de una biblioteca o una parte en particular del código desarrollado. Para ello, en algunos casos lo que se hace es directamente mostrar funcionalidades específicas del programa funcionando, como por ejemplo en el driver de la pantalla LCD y de la consola de control.

### 4.3.1. Prueba del monitor de tensión

Para el monitor de tensión y el monitor de corriente (siguiente subsección) se desea verificar que las mediciones se realizan con un error menor al especificado en el requerimiento 1.4, en la sección 2.1.

Para realizar las mediciones se conecta la entrada de tensión a una fuente variable y se lee directamente el valor medido que muestra la pantalla LCD. Se toman valores en el rango entre 0 V y 5 V en pasos aproximados de 0,5 V.

Con el valor de tensión de entrada y el medido por la placa se computa el error para cada medición (con respecto al de tensión de entrada). En la figura 4.2 se pueden ver los valores computados para los distintos valores de tensión de entrada.

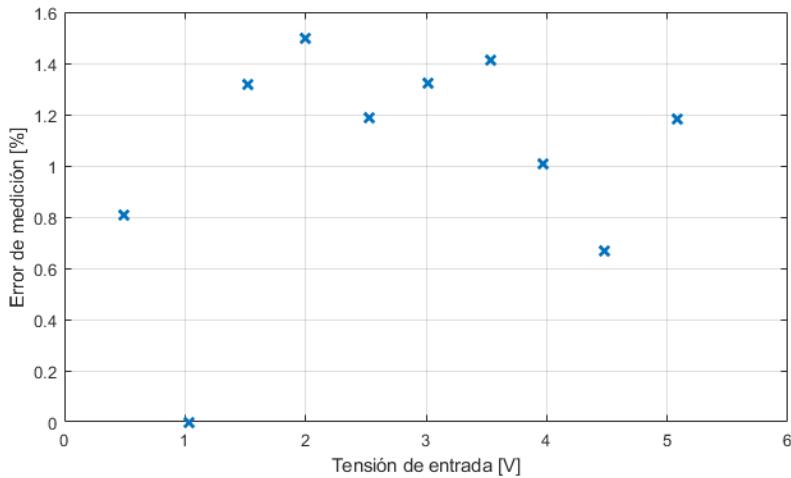


FIGURA 4.2. Error porcentual de la medición de tensión.

Del gráfico se observa que en este rango se tiene un error de medición máximo del 1,5 % y un valor promedio aproximado del 1,04 % respecto del valor real.

### 4.3.2. Prueba del monitor de corriente

Para esta prueba se procede de igual forma que en la subsección 4.2.1 pero en lugar de medir la tensión de salida del monitor se toma directamente la lectura de medición de corriente de la pantalla LCD.

Con los datos obtenidos se calcula para cada medición el error porcentual con respecto al valor real de corriente de carga. En la figura 4.3 se pueden ver los valores computados para distintos valores de corriente en la carga.

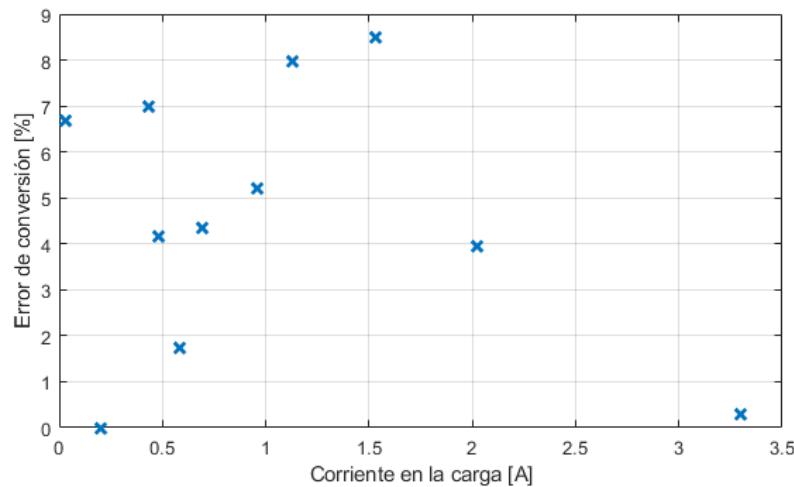


FIGURA 4.3. Error porcentual de la medición de corriente.

Observando el gráfico se desprende que en este rango se tiene un error de medición máximo del 8,5 % y un valor promedio aproximado del 4,52 % respecto del valor real. Estos valores resultan ligeramente mayores a los obtenidos en la sección 4.2.1, lo cual es de esperarse ya que durante el procesamiento digital de los datos se introduce más incertezza (debido a la resolución del ADC) y error de cálculo.

#### 4.3.3. Prueba de la pantalla LCD

En este caso se muestra directamente el LCD durante el funcionamiento normal del programa, lo que implica no solo que la biblioteca desarrollada para la pantalla funciona, si no que los drivers del SPI sobre los que esta implementada también.

En la figura 4.4 se puede ver la pantalla LCD durante el inicio del programa luego de encender la placa. Como se puede ver, contiene texto y rectángulos de distintos colores, lo que implica el uso de la mayoría de las funciones implementadas en la biblioteca.



FIGURA 4.4. Pantalla LCD durante el encendido de la placa.

#### 4.3.4. Prueba de la consola de control

La consola de control se encuentra implementada sobre el driver de la UART por lo que al validar el funcionamiento de esta también se estaría validando el del funcionamiento del driver.

Para ello lo que se hace es, mediante el emulador de puertos PuTTY, enviar mediante UART distintos comandos al microcontrolador y verificar que tienen el efecto deseado en la placa. En la figura 4.5 se puede ver la terminal luego de la ejecución de varios comandos.

```
COM7 - PuTTY

=====
EDFA Tester =====
Software release: v1.0a

TRABAJO FINAL CESE - Lucas Constantino - 2024

>ilim 2.2
>vll 4.5
>status

--- EDFA -----
Supply: OFF          Optical Output: OFF
EDFA V: 0.06 V       EDFA I: 0.00 A       EDFA P: 0.00 W
--- LIMITS -----
UV: 4.50 V           OV: 5.50 V           OC: 2.20 A
--- OPTICAL POWER --
OPT. IN POW.: 0.97 dBm   OPT. OUT POW.: 2.46 dBm
--- ALARMS -----
CASE TEMP: NO        PUMP BIAS: NO        IN POW.: NO        OUT POW.: NO

>
```

FIGURA 4.5. Uso de la consola de control.

## 4.4. Pruebas de integración

Las pruebas de integración tienen como objetivo verificar el cumplimiento de los requerimientos establecidos en el documento [5] y mencionados en la sección 2.1. Al hacer esto se valida la correcta interacción entre el hardware y el firmware, es decir, el funcionamiento del sistema como una unidad.

### 4.4.1. Funcionamiento normal

En la figura 4.6 se puede ver el estado de la pantalla LCD durante el funcionamiento normal, con el amplificador alimentado, su salida óptica prendida y ninguna alarma activa.

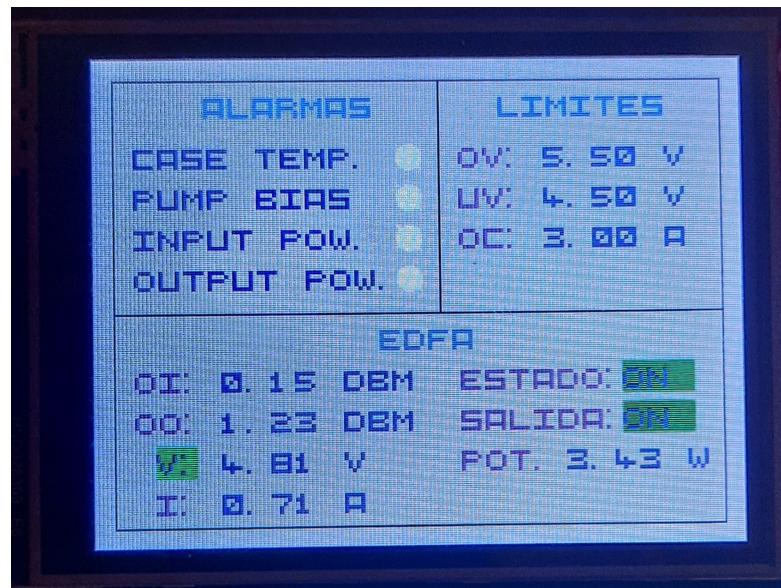


FIGURA 4.6. Pantalla LCD durante funcionamiento normal.

De aquí se observa que los valores límites, las alarmas y las variables medidas se actualizan con la frecuencia pretendida en el programa.

### 4.4.2. Detección de alarmas

Cuando alguna de las alarmas se activa mientras la salida óptica del amplificador permanece encendida, esta se debe apagar inmediatamente mediante la señal de control OUT\_POW\_MUTE, ya que de no hacerlo, esto podría dañar el EDFA.

En la figura 4.7 se puede ver la conmutación de la señal de control OUT\_POW\_MUTE (canal D1 en rojo) cuando la señal de alarma CASE\_TEMP\_ALARM se activa (canal D0 en blanco). Ambas señales son activas en nivel alto.



FIGURA 4.7. Detección de alarma y apagado de salida óptica.

En la figura anterior se puede observar que el tiempo de retardo que existe entre la detección de la alarma del EDFA y el apagado de la salida óptica es de aproximadamente 33 ms, valor que cumple con el requerimiento 4.2 especificado en la sección 2.1.

En la figura 4.8 se puede ver el estado de la pantalla LCD luego de la detección de la alarma. El amplificador permanece alimentado pero su salida apagada.

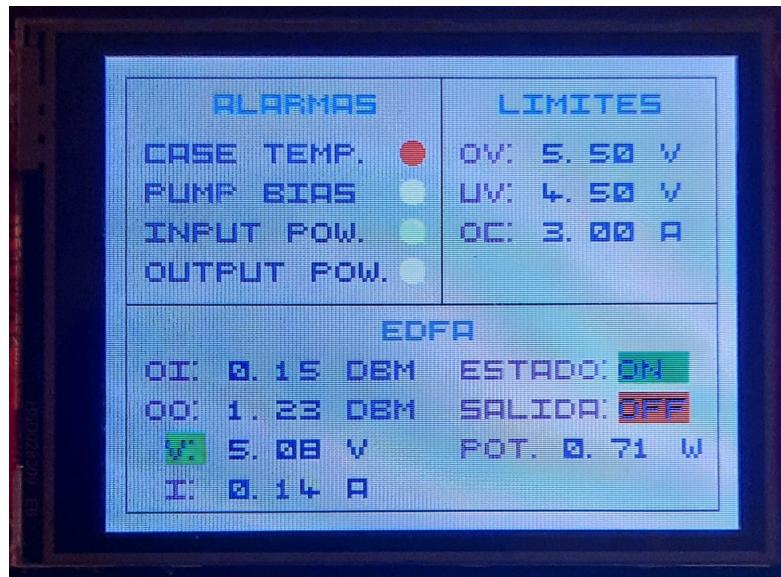


FIGURA 4.8. Pantalla LCD luego de detección de alarma.

#### 4.4.3. Detección de sobrecorriente

Al igual que para el caso de la detección de alarmas, la detección de una sobrecorriente también debe procesarse rápidamente. En este caso, se asoció la señal de alarma del monitor de corriente a una interrupción en el programa, y se midió el retardo entre el momento en que se activa dicha señal y la desactivación de la señal que controla el relé de alimentación.

En la figura 4.9 se puede ver la conmutación de la señal que controla la activación del relé (canal D1 en rojo) luego de que la señal de alarma del monitor de corriente se activa (canal D0 en blanco). En este caso la alarma es activa en nivel bajo.

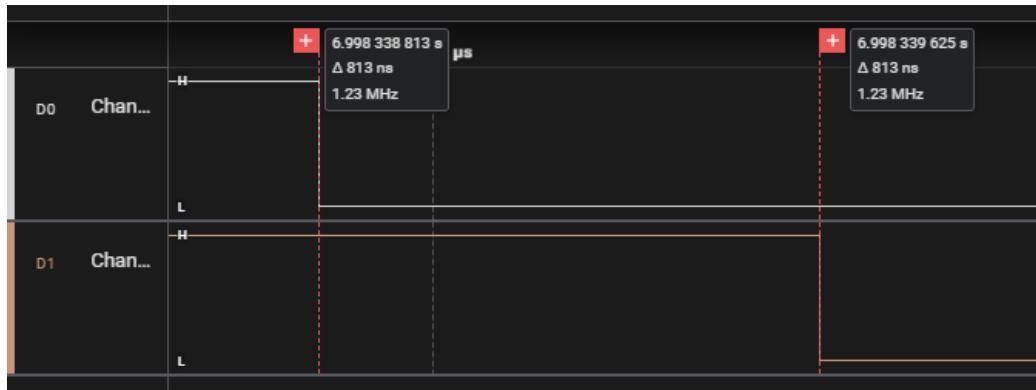


FIGURA 4.9. Detección de sobrecorriente y desconexión del EDFA.

Se puede observar que el tiempo de retardo que existe entre la detección de la sobrecorriente y la desconexión del relé es menor a 1  $\mu$ s, valor que cumple holgadamente con el requerimiento 4.1 especificado en la sección 2.1.

## 4.5. Comparación del prototipo con el producto existente

En la tabla 4.2 se puede ver una comparación entre las funcionalidades del producto existente y las del prototipo desarrollado en este trabajo.

TABLA 4.2. Comparación entre el dispositivo implementado y el existente.

Característica	Producto existente	Dispositivo desarrollado
Interfaz UART	Sí	Sí
Testigo de alarmas	Sí	Sí
Llaves para señales de control	Sí	Sí
Alimentación de EDFA	Sí	Sí
Detección de sobrecorriente	No	Sí
Detección de sobretensión	No	Sí
Detección de baja tensión	No	Sí
Protección de potencia óptica de entrada	No	Sí
Protección de potencia óptica de salida	No	Sí
Desconexión automática de EDFA en caso de alarma	No	Sí
Protección contra descarga electrostática en pines	No	Sí

## 4.6. Cumplimiento de requerimientos

## Capítulo 5

# Conclusiones

El último capítulo resume los resultados alcanzados en este trabajo, el grado de cumplimiento de los requerimientos y plantea las mejoras necesarias en etapas futuras.

### 5.1. Conclusiones generales

En líneas generales, a pesar de los contratiempos encontrados durante el desarrollo, el trabajo se llevó a cabo con éxito ya que:

- Se obtuvo un prototipo validado del hardware, lo que permite volver a usar el mismo diseño y componentes en el producto final.
- Se desarrolló una primera versión validada del firmware.
- Se logró interiorizarse en la tecnología de los EDFA y su funcionamiento.
- Como se explica en la sección 3.3, el dispositivo desarrollado no es el producto final si no una primera versión que funciona como prueba de concepto y simulador del comportamiento del amplificador. Aún así, logra cumplir con los requerimientos principales establecidos por la empresa, lo que permite darlo por finalizado y pasar a la etapa de desarrollo del producto final.
- Se hizo uso de muchas herramientas, conceptos y metodologías incorporadas durante la cursada de la especialización, dando como resultado un trabajo profesional y facilitando su desarrollo.

### 5.2. Trabajo futuro

Con el objetivo de contar con un producto final apto para uso en tareas de integración e investigación y desarrollo, se planea avanzar en varios aspectos. Entre estos se encuentran:

- El rediseño y la fabricación de la versión final del PCB del tester, corrigiendo los errores de diseño encontrados en el prototipo. También se deberá incluir el microcontrolador de la placa NUCLEO-144 o un equivalente junto con su interfaz de programación. Todo esto se debe hacer teniendo en cuenta su factor de forma ya que se deberá conectar a un EDFA y montarse sobre una carcasa.
- Mejorar el firmware añadiendo más funcionalidades y robustecerlo corrigiendo errores.

- Una vez rediseñado el PCB, volver a validar el hardware y firmware utilizando un EDFA provisto por la empresa.
- Realizar las modificaciones necesarias en el firmware para lograr compatibilidad con los *scripts* de Python que la empresa utiliza para ejecutar pruebas de aceptación y funcionamiento automatizadas.

## **Apéndice A**

### **Círcito esquemático completo**

A continuación se puede ver el circuito esquemático completo del dispositivo fabricado y ensamblado.

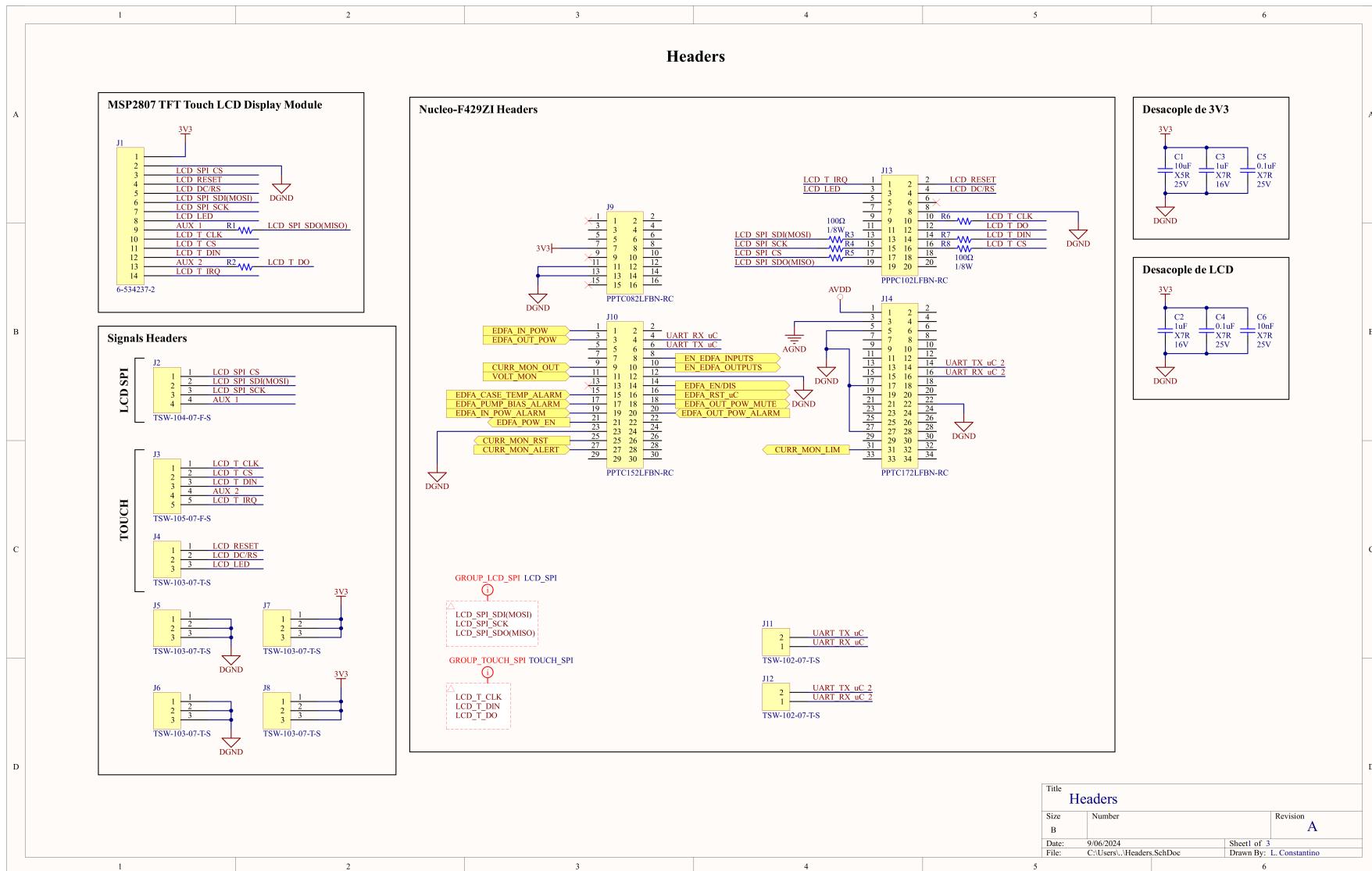


FIGURA A.1. Circuito esquemático. Conectores.

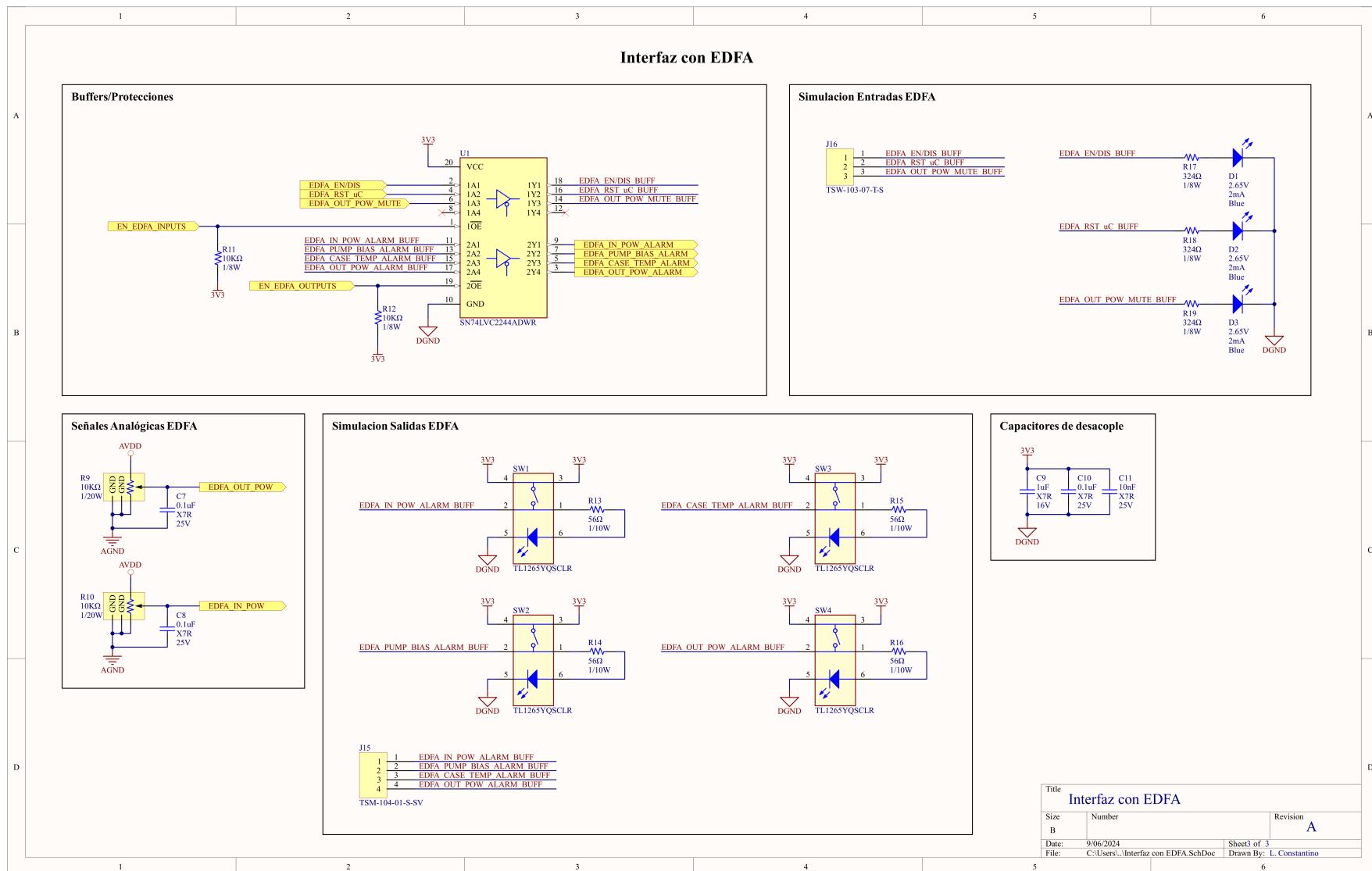


FIGURA A.2. Circuito esquemático. Interfaz con EDFA.

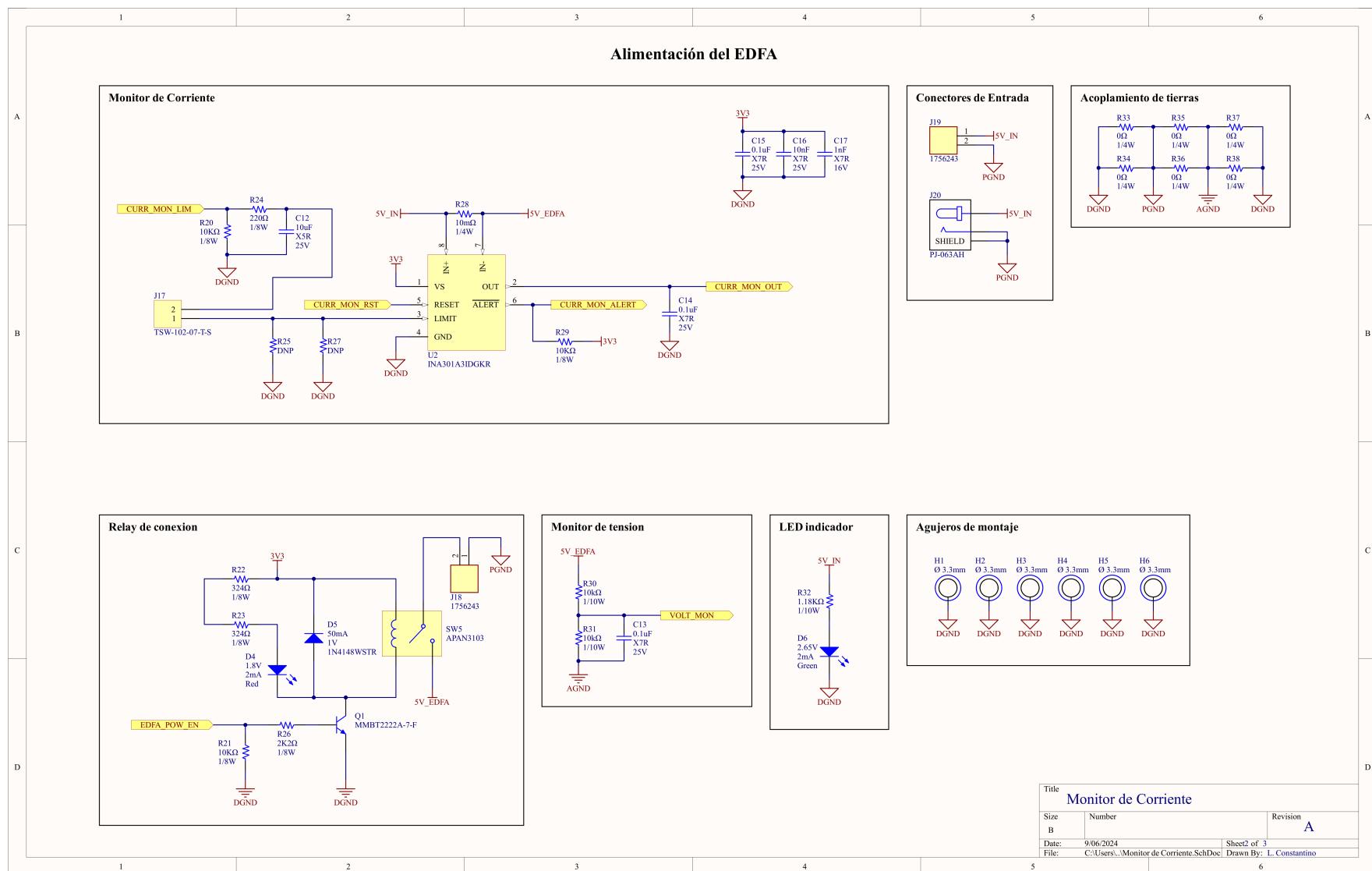


FIGURA A.3. Circuito esquemático. Alimentación del EDFA.

## Apéndice B

# Banco de ensayos

A continuación se puede ver el banco de ensayos con todos los instrumentos y componentes necesarios para ejecutar las pruebas de hardware y firmware.

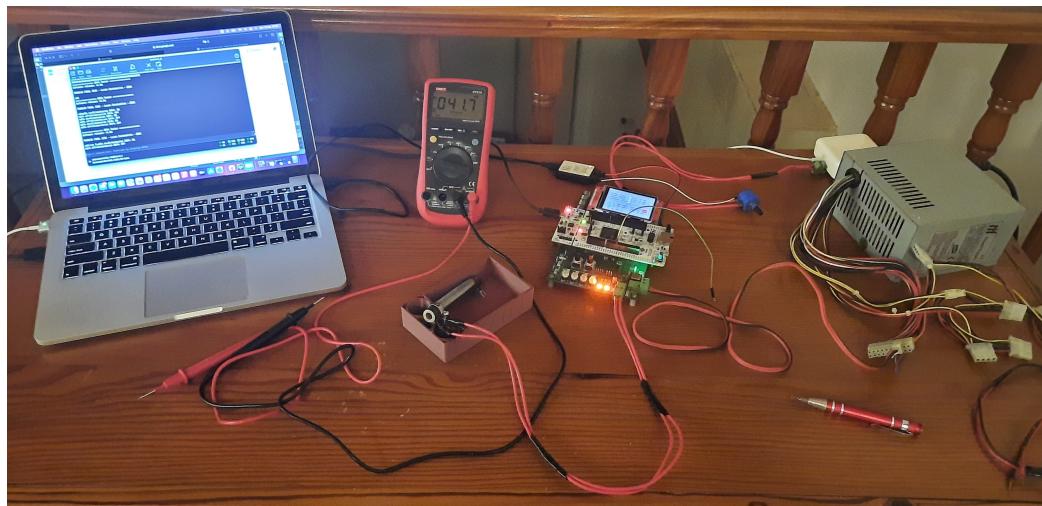


FIGURA B.1. Banco de ensayos.



# Bibliografía

- [1] Wikipedia. *Low Earth Orbit*. URL: [https://en.wikipedia.org/wiki/Low\\_Earth\\_orbit](https://en.wikipedia.org/wiki/Low_Earth_orbit) (visitado 09-09-2024).
- [2] Wikipedia. *Geostationary Orbit*. URL: [https://en.wikipedia.org/wiki/Geostationary\\_orbit](https://en.wikipedia.org/wiki/Geostationary_orbit) (visitado 09-09-2024).
- [3] Skyloom Global. URL: <https://www.skyloom.co/> (visitado 22-03-2023).
- [4] Fiber Optics Share. *Basic Knowledge About EDFA*. URL: <https://www.fiberopticshare.com/basic-knowledge-edfa.html> (visitado 14-04-2023).
- [5] Ing. Lucas Constantino. *Tester para amplificador óptico. Especificación de requerimientos de software*. 2022. URL: <https://drive.google.com/file/d/12-mV9YiTrhrPz9u6ybiGlePk2mSL5Vg6/view?usp=sharing>.
- [6] FiberLabs Inc. *Optical Communication Band*. URL: <https://www.fiberlabs.com/glossary/optical-communication-band/> (visitado 10-09-2024).
- [7] RP Photonics Encyclopedia. *Erbium-doped Fiber Amplifiers*. URL: [https://www.rp-photonics.com/erbium\\_doped\\_fiber\\_amplifiers.html](https://www.rp-photonics.com/erbium_doped_fiber_amplifiers.html) (visitado 14-04-2023).
- [8] Wikipedia. *Fibra óptica*. URL: [https://es.wikipedia.org/wiki/Fibra\\_%C3%B3ptica](https://es.wikipedia.org/wiki/Fibra_%C3%B3ptica) (visitado 14-04-2023).
- [9] Wikipedia. *Stimulated emission*. URL: [https://en.wikipedia.org/wiki/Stimulated\\_emission](https://en.wikipedia.org/wiki/Stimulated_emission) (visitado 14-04-2023).
- [10] ThorLabs. *Gain Flattening Filters*. URL: [https://www.thorlabs.com/newgroupage9.cfm?objectgroup\\_id=15022](https://www.thorlabs.com/newgroupage9.cfm?objectgroup_id=15022) (visitado 16-09-2024).
- [11] Amphenol. *Conectores Micro-D*. URL: <https://www.amphenolcanada.com/pdf/Micro-D-Catalog.pdf> (visitado 08-09-2024).
- [12] ST Microelectronics. *STM32F429ZI*. URL: <https://www.st.com/resource/en/datasheet/stm32f429zi.pdf>.
- [13] ST Microelectronics. *Nucleo-144*. URL: <https://www.farnell.com/datasheets/2014265.pdf>.
- [14] Texas Instruments. *INA301*. URL: <https://www.ti.com/lit/ds/symlink/ina301.pdf?src=supplier=Digi-Key>.
- [15] LCDWiki. *MSP2807*. URL: [http://www.lcdwiki.com/2.8inch\\_SPI\\_Module\\_ILI9341\\_SKU:MSP2807](http://www.lcdwiki.com/2.8inch_SPI_Module_ILI9341_SKU:MSP2807) (visitado 07-04-2023).
- [16] Wikipedia. *FreeRTOS*. URL: <https://en.wikipedia.org/wiki/FreeRTOS> (visitado 30-03-2023).
- [17] AWS. *FreeRTOS*. URL: <https://docs.aws.amazon.com/freertos/latest/userguide/dev-guide-freertos-kernel.html> (visitado 07-04-2023).
- [18] Wikipedia. *Hardware abstraction*. URL: [https://en.wikipedia.org/wiki/Hardware\\_abstraction](https://en.wikipedia.org/wiki/Hardware_abstraction) (visitado 11-04-2023).
- [19] ST. *Description of STM32F1 HAL and low-layer drivers*. URL: [https://www.st.com/resource/en/user\\_manual/dm00154093-description-of-stm32f1-hal-and-low-layer-drivers-stmicroelectronics.pdf](https://www.st.com/resource/en/user_manual/dm00154093-description-of-stm32f1-hal-and-low-layer-drivers-stmicroelectronics.pdf) (visitado 13-04-2023).
- [20] Wikipedia. *Analog-to-digital converter*. URL: [https://en.wikipedia.org/wiki/Analog-to-digital\\_converter](https://en.wikipedia.org/wiki/Analog-to-digital_converter) (visitado 31-03-2023).

- [21] Embedded. *UART*. URL: <https://www.embedded.com/understanding-the-uart/> (visitado 02-04-2023).
- [22] Wikipedia. *SPI*. URL: [https://en.wikipedia.org/wiki/Serial\\_Peripheral\\_Interface](https://en.wikipedia.org/wiki/Serial_Peripheral_Interface) (visitado 13-04-2023).
- [23] Panasonic. *Relé de propósito general serie PA-N*. URL: [https://mm.digikey.com/Volume0/opasdata/d220001/medias/docus/891/PA-N\\_Series.pdf](https://mm.digikey.com/Volume0/opasdata/d220001/medias/docus/891/PA-N_Series.pdf) (visitado 09-09-2024).
- [24] Fairchild Semiconductor. *Transistor bipolar NPN*. URL: <https://rocelec.widen.net/view/pdf/ylkha27spa/FAIRS44268-1.pdf?t.download=true&u=5oefqw> (visitado 09-09-2024).
- [25] Wikipedia. *RC Circuit*. URL: [https://en.wikipedia.org/wiki/RC\\_circuit](https://en.wikipedia.org/wiki/RC_circuit) (visitado 09-09-2024).
- [26] Texas Instruments. *Buffers de ocho canales serie SN74LVC2244A*. URL: <https://www.ti.com/lit/ds/symlink/sn74lvc2244a.pdf> (visitado 08-09-2024).
- [27] E-Switch. *Pulsadores serie TL1265*. URL: <https://eswitch.wpengine.com/wp-content/uploads/2022/06/TL1265.pdf> (visitado 08-09-2024).
- [28] Kingbright. *LEDs de montaje superficial serie AA3528LVBS/D*. URL: <https://www.kingbrightusa.com/images/catalog/SPEC/AA3528LVBS-D.pdf> (visitado 08-09-2024).
- [29] Bourns. *Resistencias variables serie PTV09*. URL: <https://www.bourns.com/docs/Product-Datasheets/PTV09.pdf> (visitado 08-09-2024).