



CARRERA DE ESPECIALIZACIÓN EN SISTEMAS EMBEBIDOS

MEMORIA DEL TRABAJO FINAL

Tester para amplificador óptico

Autor:
Ing. Lucas Constantino

Director:
Dr. Ing. Alejandro Ghersin (FIUBA)

Jurados:
Nombre del jurado 1 (pertenencia)
Nombre del jurado 2 (pertenencia)
Nombre del jurado 3 (pertenencia)

*Este trabajo fue realizado en la Provincia de Buenos Aires,
entre Marzo de 2022 y Junio de 2023.*

Resumen

En la presente memoria se describen el diseño, construcción y validación de un prototipo de un tester para un amplificador de fibra óptica dopado con erbio. Este desarrollo surge a partir de la necesidad de la empresa Skyloom Global de contar con una herramienta que permita controlar y medir los parámetros de este tipo de amplificador de forma aislada y se utilizará durante la etapa de desarrollo e integración de terminales de comunicación óptica para satélites de órbita baja.

El sistema fue desarrollado sobre arquitectura ARM utilizando un sistema operativo de tiempo real, donde se aplicaron los conocimientos adquiridos durante la carrera tales como programación de microcontroladores, protocolos de comunicación e ingeniería de software. Además, fueron de particular importancia la utilización de los conceptos de máquinas de estados, capas de abstracción y testing de software para la etapa de validación del sistema.

Índice general

Resumen	I
1. Introducción general	1
1.1. Contexto y motivación	1
1.2. Producto existente	3
1.3. Objetivos y alcance	4
2. Introducción específica	5
2.1. Requerimientos	5
2.2. Funcionamiento de un amplificador óptico	6
2.3. Interfaz del amplificador óptico	8
2.4. Componentes del sistema	9
2.4.1. Microcontrolador	9
2.4.2. Monitor de corriente	9
2.4.3. Pantalla táctil LCD	10
2.5. Recursos de software	11
2.5.1. Sistema operativo de tiempo real FreeRTOS	11
2.5.2. Capa de abstracción de hardware (HAL)	12
2.6. Periféricos utilizados	12
2.6.1. Conversor analógico-digital (ADC)	13
2.6.2. Universal Asynchronous Receiver Transmitter (UART) . . .	14
2.6.3. Serial Peripheral Interface (SPI)	14
3. Diseño e implementación	17
3.1. Dispositivo implementado	17
3.2. Arquitectura de hardware	18
3.2.1. Diagrama del dispositivo	19
3.2.2. Medición de corriente del amplificador	19
3.2.3. Relé de alimentación	20
3.2.4. Medición de tensión del amplificador	20
3.2.5. Conexión a pantalla LCD	21
3.2.6. Entradas y salidas digitales	21
3.2.7. Indicadores de potencia óptica	22
3.2.8. Interfaz UART	22
3.3. Arquitectura de firmware	22
3.3.1. Máquina de estados	23
3.3.2. Capa de abstracción del amplificador	24
3.3.3. Driver del ADC	25
3.3.4. Driver de la pantalla LCD	26
3.3.5. Driver de UART	26
3.3.6. Consola de control	26
4. Ensayos y resultados	29

4.1. Instrumental utilizado	29
4.2. Banco de ensayos	29
4.3. Pruebas de hardware	30
4.3.1. Pruebas del monitor de corriente	30
4.3.2. Pruebas del relé de alimentación	30
4.4. Pruebas de firmware	30
4.4.1. Pruebas del monitor de tensión	30
4.4.2. Pruebas del monitor de corriente	30
4.4.3. Pruebas de la pantalla LCD	30
4.4.4. Pruebas de la pantalla táctil	30
4.4.5. Pruebas de la comunicación UART	30
4.4.6. Pruebas de las alarmas y señales de control	30
4.5. Pruebas de integración	30
5. Conclusiones	31
5.1. Conclusiones generales	31
5.2. Trabajo futuro	31
A. Circuito esquemático completo	33
Bibliografía	37

Índice de figuras

1.1. Propuesta de la empresa Skyloom Global ¹ .	2
1.2. Esquema de uso y conexionado del sistema.	3
2.1. Pérdidas en la fibra.	6
2.2. Configuración interna de un EDFA.	7
2.3. Pantalla LCD táctil MSP2807.	10
2.4. Estructura de capas con FreeRTOS.	12
2.5. Conversión de una señal con un ADC de 3 bits.	13
2.6. Trama de transmisión UART.	14
2.7. Transacción SPI.	15
2.8. Conexionado de un maestro con múltiples esclavos.	16
3.1. Placa fabricada para el trabajo.	18
3.2. Diagrama completo del hardware implementado.	19
3.3. Esquema del monitor de corriente	20
3.4. Circuito del monitor de corriente.	21
3.5. Arquitectura de firmware.	23
3.6. Diagrama simplificado de la máquina de estados.	24
3.7. Capa de abstracción del EDFA.	25
A.1. Circuito esquemático. Conectores.	34
A.2. Circuito esquemático. Interfaz con EDFA.	35
A.3. Circuito esquemático. Alimentación del EDFA.	36

Índice de tablas

2.1. Señales de la interfaz del EDFA.	8
2.2. Características de la pantalla LCD.	11
3.1. Configuración del ADC.	25
3.2. Configuración de los SPI del LCD.	26
3.3. Configuración de la interfaz UART.	26
3.4. Comandos/argumentos de la consola.	27
4.1. Lista de instrumental utilizado.	29

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 y motivación

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) utilizando enlaces ópticos de alta velocidad.

El funcionamiento general de esta idea se basa en un satélite geoestacionario (órbita GEO) al cual los satélites en LEO transmiten la información (denominado *uplink*) para luego descargar esta a una estación en Tierra (denominado *downlink*). De esta forma se obtiene una alta disponibilidad de los datos ya que el satélite en GEO siempre tiene visión de la Tierra. Un diagrama de esta propuesta se muestra en la figura 1.1.



FIGURA 1.1. Propuesta de la empresa Skyloom Global¹.

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 en el receptor se la detecte. Esto se debe principalmente a que la distancia de espacio libre estimada entre dos satélites en LEO es de unos 4000 km [1].

Para solucionar el problema antes mencionado se introduce a la salida del láser un amplificador dopado con erbio o EDFA (de las siglas *Erbium Doped Fiber Amplifier*). 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.

Este amplificador formará parte de la terminal de comunicaciones y estará sometido a intensivas pruebas de funcionamiento y rendimiento durante la etapa de

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

investigación y desarrollo. Para esto es indispensable contar con una herramienta que brinde a los ingenieros a cargo de estas la posibilidad de utilizar el amplificador de forma aislada, es decir, sin tener que usar hardware de vuelo² adicional.

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

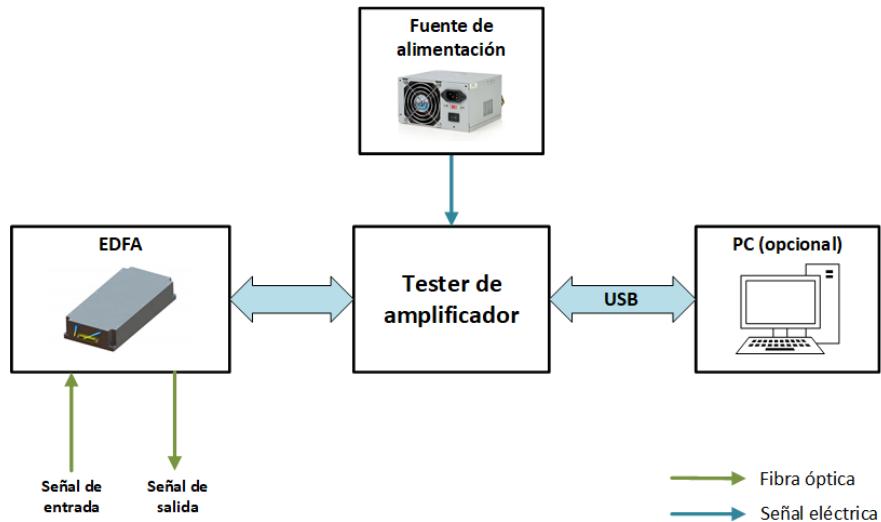


FIGURA 1.2. Esquema de uso y conexionado del sistema.

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. Producto existente

Actualmente el fabricante del amplificador posee a la venta una placa electrónica capaz de conectarse a un EDFA y proveer ciertas funcionalidades similares. La empresa ha hecho uso de ella en el pasado pero luego pasó a descartarla debido a tres problemas mayores:

- No permite el uso de todas las señales presentes en el conector del amplificador
- Tiene una alta tasa de fallas
- Su costo es muy elevado en relación a sus prestaciones

Lo expuesto anteriormente es uno de los principales motivos que dieron origen a la necesidad de contar con el sistema propuesto en este trabajo.

²El término *de vuelo* se refiere a todo hardware que formará parte de la terminal

1.3. 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 [2].

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 1550 nm y 1650 nm), 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.

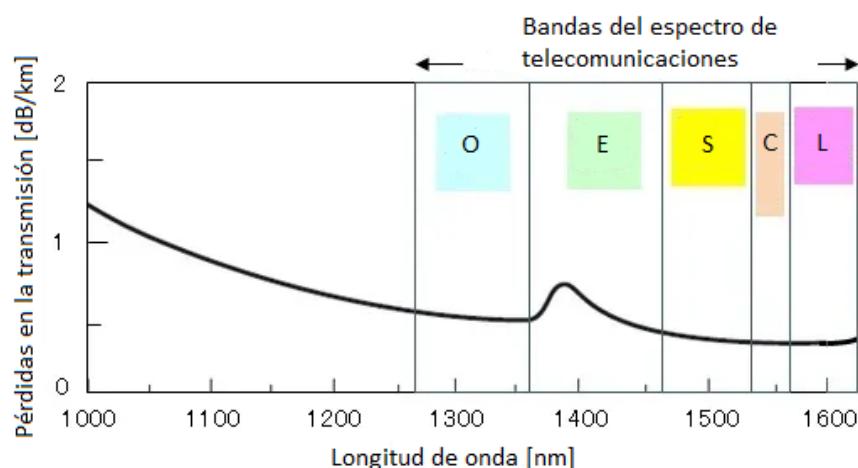


FIGURA 2.1. Pérdidas en la fibra.

Inventado en 1987, el EDFA es generalmente usado para compensar las pérdidas mencionadas en una línea de transmisión óptica. Éste 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
- 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 monomodo¹.

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 %) 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 que se combina con la señal entrante (ahora a la salida del aislador) y pasan por la fibra dopada con erbio. En esta instancia es donde se genera la amplificación de la señal propiamente dicha, mediante un efecto denominado emisión estimulada².

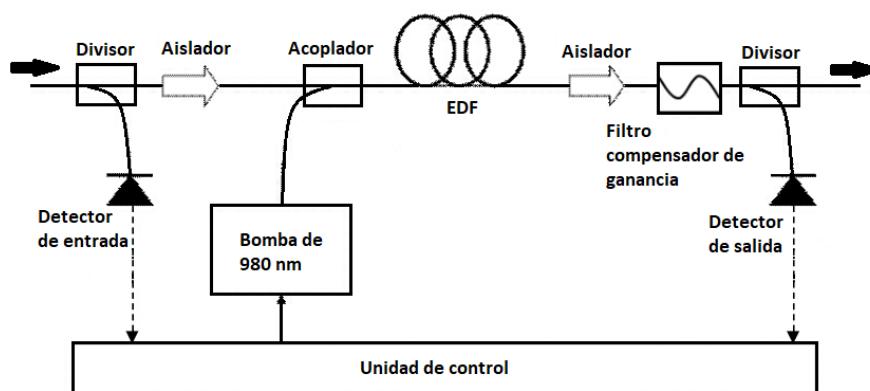


FIGURA 2.2. Configuración interna de un EDFA.

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 compensador 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 (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. Ésta 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 [5][6].

¹Tipo de fibra óptica en la que solo se propaga un modo de luz, paralela al eje de la fibra. Tienen un tamaño entre 8.3 y 10 micrones, permiten alcanzar distancias de hasta 400 km y tasas de transmisión de hasta 10 Gbps [3].

²Proceso por el cual un fotón incidente de una determinada frecuencia interactúa con un electrón excitado, causando que éste caiga a un nivel de energía menor. La energía liberada se transfiere al campo electromagnético, creando un nuevo fotón con la misma frecuencia, polarización y dirección de movimiento que el fotón incidente [4].

2.3. Interfaz del amplificador óptico

Como se mencionó en la sección 1.1, para poder controlarlo, el EDFA cuenta con un conector de 25 pines tipo microD. Este conector contiene varios grupos 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
5V	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 forman parte del sistema presentado en la sección 1.1 fueron seleccionados con el objetivo de cumplir con los requisitos y a su vez utilizar la menor cantidad posible. 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, 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. Algunas de sus características más relevantes para esta aplicación son [7]:

- Núcleo CPU Arm® Cortex®-M4 de 32 bits con unidad de punto flotante, frecuencia hasta 180 MHz, unidad de protección de memoria e instrucciones DSP
- Memoria Flash de hasta 2 MB
- Interfaces de comunicación:
 - 3 interfaces I2C
 - 4 UART/USART (hasta 11.25 Mbit/s)
 - 6 interfaces SPI (hasta 45 Mbit/s)
- Hasta 168 puertos de entrada/salida con interrupción
- 3 conversores ADC de 12 bit de resolución y 2.4 MSPS. Hasta 24 canales
- DMA de propósito general con FIFO y soporte de ráfaga
- Interfaces SWD y JTAG para depuración
- Hasta 12 *timers* de 16 bits con IC/OC y PWM

La alta velocidad de reloj, la gran capacidad de memoria Flash y la gran cantidad de periféricos disponibles hacen a este modelo ideal para la ejecución de un RTOS.

La principal ventaja de utilizar la placa de desarrollo Nucleo-144 es que esta ya trae incorporada la interfaz de programación y depuración ST-LINK/V2 [8].

2.4.2. Monitor de corriente

El circuito integrado elegido para efectuar la medición de corriente de alimentación del amplificador mientras este se encuentra en funcionamiento es el INA301A3. Sus principales características son:

- Alto rango de tensión de modo común (0 V a 36 V)
- Salida analógica y a comparador
- Máxima tensión de *offset* de salida: 35 uV
- Máximo error de ganancia: 0.1
- Ganancia del amplificador: 100 V/V
- Nivel de alerta programable mediante un resistor
- Tiempo total de respuesta de la alerta: 1 us

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) [9].

2.4.3. Pantalla táctil LCD

El modelo del módulo de la pantalla LCD utilizada en el trabajo es MSP2807, 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.

En la tabla 2.2 se listan las características de la pantalla.

TABLA 2.2. Características de la pantalla LCD.

Característica	Valor
Color	65K colores RGB
Tamaño	2.8 pulgadas (7.11 cm)
Tipo de pantalla	TFT
Resolución	320x240 pixeles
Área activa	43.2x57.6 mm
Tensión de alimentación	3.3 V - 5 V
Nivel lógico de I/O	3.3 V (TTL)

El controlador del display tiene una interfaz SPI para el envío de los datos y además una señal denominada DC/RS para indicar si lo que está enviando el maestro es un dato o un registro interno del chip. También consta de una señal de reset para borrar todo el contenido del display.

El controlador de la función táctil también tiene un bus SPI y una señal adicional de interrupción denominada T_IRQ que indica cuando el display está siendo tocado [10].

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 o RTOS (de *Real Time Operating System*) para microcontroladores o pequeños microprocesadores, diseñado para ocupar poco espacio, ser confiable y fácil de usar. Está escrito en C para que sea fácil de mantener y trasladar a nuevos dispositivos.

FreeRTOS provee recursos que facilitan la ejecución de aplicaciones sobre el sistema operativo como tareas, semáforos, *mutexes*, temporizadores por software, colas y gestores de memoria dinámica [11].

Si se separa el firmware en capas desde la más baja (hardware) hasta la más alta (aplicación de usuario) el lugar que ocupa FreeRTOS se ubica por arriba de los drivers provistos por el fabricante, brindando una capa de abstracción de mayor nivel. De esta forma, el sistema operativo actúa como puente entre las capas más bajas y las más altas, gestionando los recursos de hardware de forma automática y brindando servicios a la capa de aplicación.

En la figura 2.4 se puede ver la estructura de capas del firmware con FreeRTOS incluida.

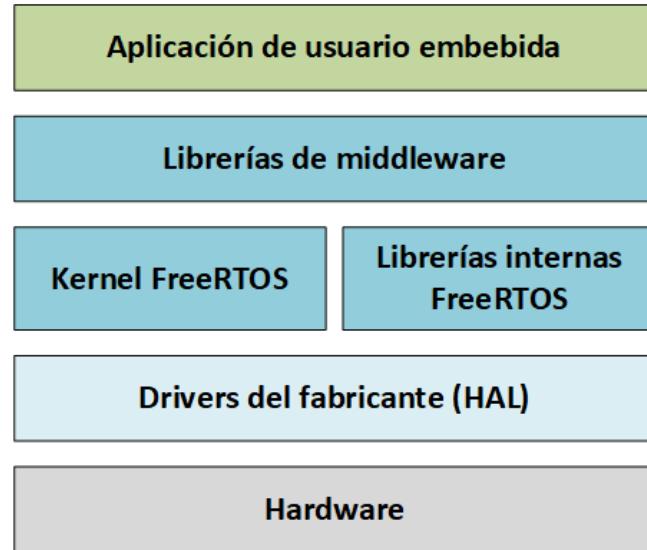


FIGURA 2.4. Estructura de capas con FreeRTOS.

Uno de las ventajas principales de usar FreeRTOS es la posibilidad de ejecutar varias tareas o *tasks* en forma simultánea tal como lo hace un sistema operativo cualquiera. Un *scheduler* gestiona la ejecución de cada una de forma individual y con su propio contexto (*stack* y variables). Esto permite estructurar la aplicación separándola en tareas independientes pero sincronizadas [12].

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, como se puede ver en la figura 2.4.

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) [13].

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-instanciamiento para todos los periféricos (UART, SPI, temporizadores, ADC, etc.). Éstas 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 [14].

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 que

agregar ningún hardware adicional y para poder utilizarlos solo hizo falta leer las hojas de datos. De todas formas, en esta sección se provee una breve descripción de cada uno.

2.6.1. Conversor analógico-digital (ADC)

Un conversor analógico-digital o ADC (de *Analog-to-Digital Converter*) es un dispositivo que convierte una señal eléctrica analógica proveniente, por ejemplo, de un sensor a una señal digital. La principal ventaja de esta conversión es que su valor puede ser almacenado en un sistema digital, por lo que estará representada por un número binario.

La señal pasa de ser continua en el tiempo y continua en amplitud a discreta en el tiempo y discreta en amplitud. Esto quiere decir que la señal analógica podría tomar cualquier valor en cualquier instante. En cambio, cuando es convertida la señal se encuentra cuantizada, es decir, podrá tomar solamente determinados valores que dependen de la cantidad de bits del ADC y del fondo de escala (resolución).

Esto hace que no se puedan representar todos los valores que se miden y que cada muestra tomada estará truncada al valor más cercano representable. Este error introducido en la medición se denomina error de cuantización. A modo de ejemplo, en la figura 2.5 se muestra una conversión analógica-digital de una señal senoidal con un ADC de 3 bits. La señal de entrada (en rojo) es muestreada a intervalos constantes y la señal resultante (en azul) está cuantizada.

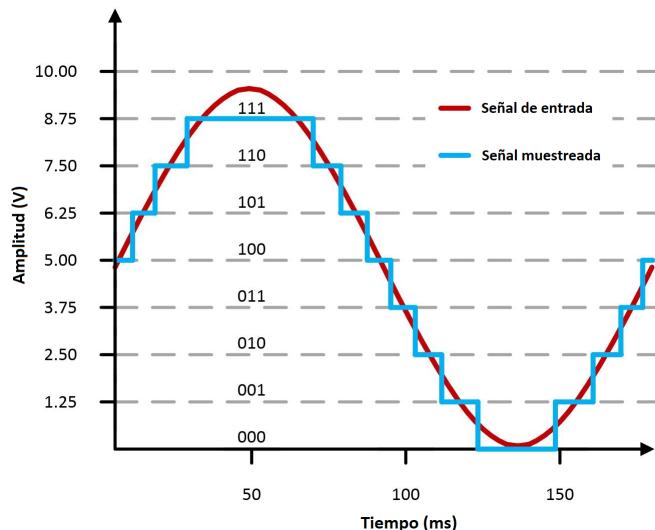


FIGURA 2.5. Conversión de una señal con un ADC de 3 bits.

De la figura 2.5 se ve que mientras más bits de resolución tenga el ADC, se pueden tomar valores más cercanos al medido y por lo tanto tener una mejor representación de la señal.

Esto impacta directamente en el parámetro de relación señal-ruido que, junto con el ancho de banda (velocidad de muestreo), es uno de los parámetros con los que se caracteriza el rendimiento de un ADC [15].

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.

La forma de transmisión de datos del protocolo tiene una estructura fija. Por cada dato que se transmite (cuyo ancho puede variar entre 5 y 9 bits), se transmiten además bits adicionales de comienzo y fin de trama (*Start* y *Stop*) y, opcionalmente, los de paridad para la detección de errores. En la figura 2.6 se puede ver la composición de una trama completa.

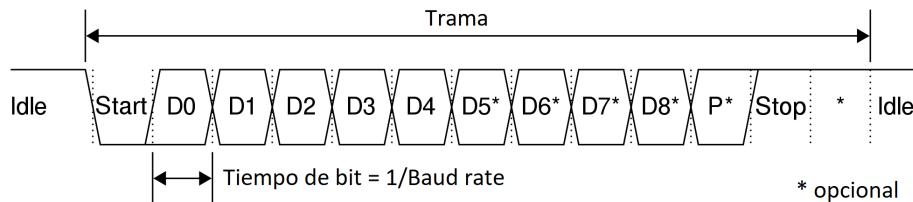


FIGURA 2.6. Trama de transmisión UART.

Los bits de *Start* y *Stop* son para indicarle al receptor cuando debe comenzar y cuando parar de tomar los datos transmitidos. También sirve para realizar una sincronización de los relojes del transmisor y del receptor. Como el protocolo es asíncrono, es decir, no transmite el reloj en ninguna de sus líneas, los relojes internos de ambos deben estar en fase.

El estado *Idle* de la línea es el estado inactivo, es decir, el estado de reposo cuando no se encuentra transmitiendo [16].

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 embobido y pequeños periféricos como sensores y memorias SD.

Los dispositivos SPI se comunican en modo *full duplex* (ambos sentidos simultáneos) mediante una arquitectura maestro-esclavo con líneas separadas para los datos y el clock. Generalmente constan de 4 líneas:

- SCLK: reloj o clock (generado por el maestro)
- MOSI: *Master Out Slave In*. Salida de datos de maestro a esclavo
- MISO: *Master In Slave Out*. Salida de datos de esclavo a maestro
- CS/SS: *Chip Select*. Salida de maestro para indicar al esclavo que se están enviando o recibiendo datos

Para comenzar la comunicación el maestro primero configura el clock, utilizando una frecuencia soportada por el dispositivo esclavo, generalmente entre 100 kHz y 10 MHz. Luego selecciona el esclavo poniendo un nivel bajo en la línea CS. En

este momento, a veces, se requiere un tiempo mínimo de espera antes de enviar ciclos de clock, por ejemplo para una conversión analógico-digital.

Durante cada ciclo de clock ocurre una transmisión *full duplex*. El maestro envía un bit por la línea MOSI y el esclavo la lee, mientras que el esclavo envía otro bit por la línea MISO y el maestro la lee. Ésta secuencia se mantiene incluso cuando se está enviando un dato en una sola dirección.

En la figura 2.7 se pueden ver las señales durante una transacción de maestro a esclavo y de esclavo a maestro. Se observa que la señal CS se mantiene en nivel bajo durante todo el tiempo en que la transmisión está activa. Además, las transiciones de las señales MOSI y MISO para cada bit ocurren solamente en cada flanco negativo del clock. En este caso el ancho de la palabra transmitida es de 8 bits, pero se puede implementar de igual forma para cantidades mayores o menores. Esto es posible debido a que no existe un protocolo estándar para la transmisión SPI.

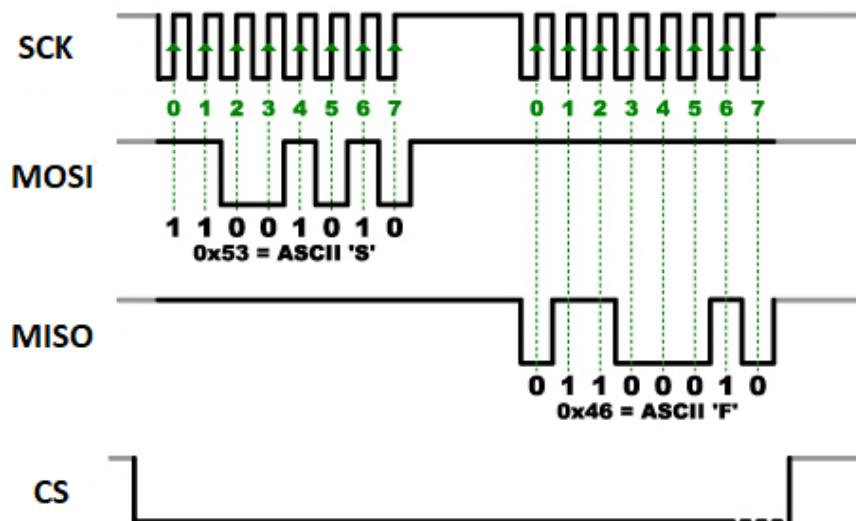


FIGURA 2.7. Transacción SPI.

En el caso de la figura 2.7 el nivel del clock cuando la interfaz se encuentra inactiva (*Idle*) es alto y el flanco en que cambia el bit transmitido es descendente pero, al igual que la velocidad de transmisión, estos parámetros pueden ser configurados con distintos valores por el maestro antes de realizar cada transacción.

El hecho de contar con una línea de selección de esclavo permite realizar una conexión de un maestro con múltiples esclavos simplemente agregando una línea CS por cada esclavo y reutilizando las demás, como se observa en la figura 2.8. De esta forma, si un esclavo recibe ciclos de clock y datos pero no está habilitado mediante la señal CS, los descartará [17].

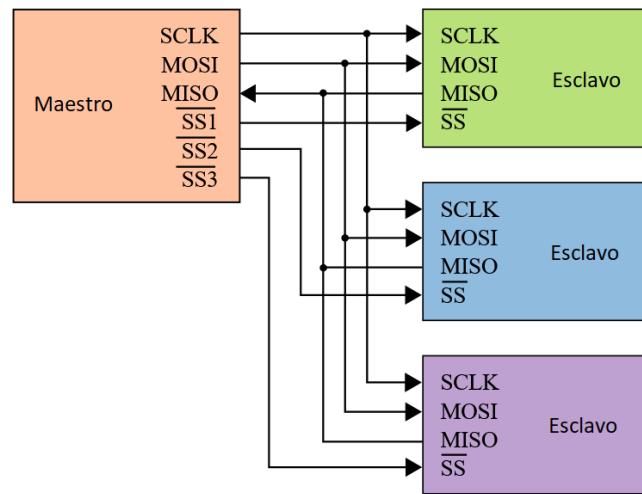


FIGURA 2.8. Conexión de un maestro con múltiples esclavos.

A diferencia de otros protocolos como UART, SPI define solamente una estructura de hardware para la transmisión de información. No define un protocolo para los datos transmitidos. Esto quiere decir que los bits transmitidos no tienen una función determinada predefinida si no que ésta es específica de la aplicación y corre por parte del hardware (o software) realizar la interpretación correcta.

Capítulo 3

Diseño e implementación

En este capítulo se realiza una explicación detallada del diseño del hardware, como 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.

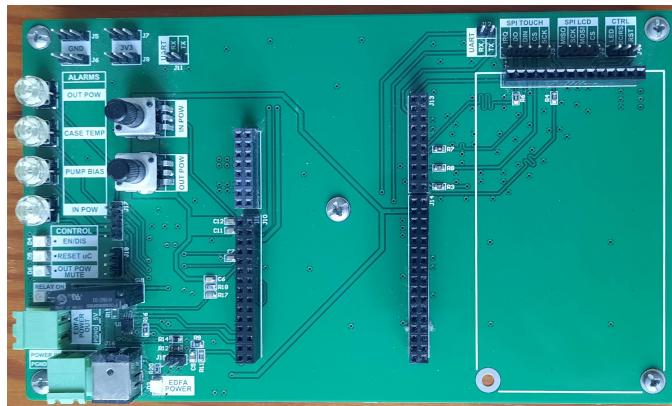
3.1. Dispositivo implementado

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 usarse. La placa construida se puede interpretar como un prototipo de desarrollo o una versión de ingeniería ya que tiene dos objetivos: verificar el funcionamiento del hardware diseñado y desarrollar el firmware que se ejecutará en el microcontrolador.

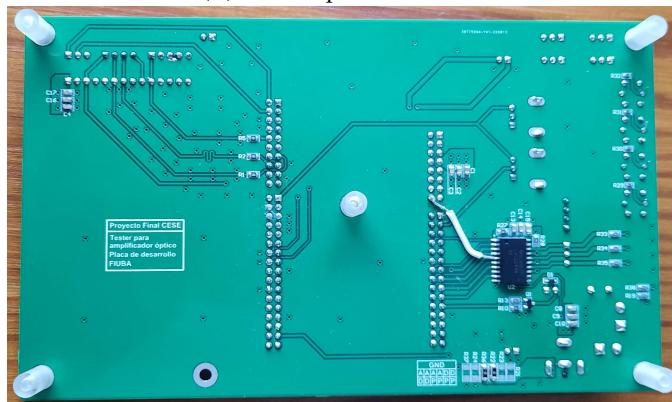
La versión armada para este trabajo, además de incluir los mismos componentes que se pretenden usar en el producto final, cuenta con todo el hardware necesario para simular la interfaz electrónica del modelo del amplificador óptico. Esto permite generar todas las señales que se encuentran en el conector, tanto las analógicas como las 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.

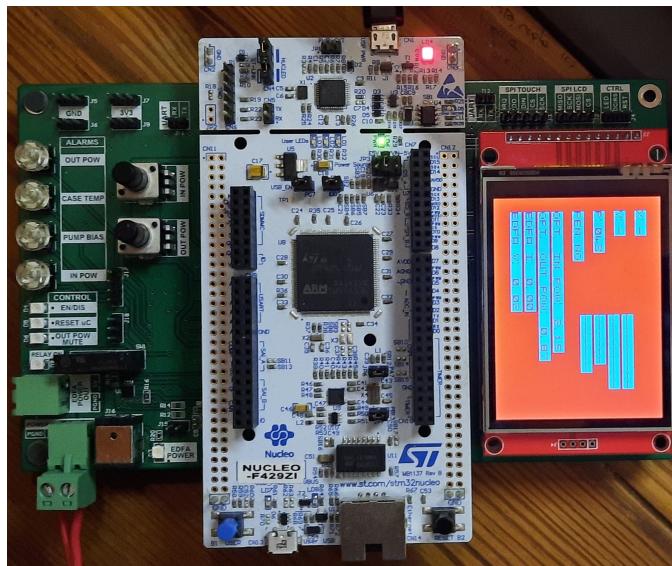
El PCB fue enviado a fabricar a la empresa china JLC y consta de 4 capas: dos planos internos de referencia conectados a 3.3 V y GND y dos capas de señal (*Top* y *Bottom*). En la figura 3.1a y 3.1b se pueden ver dos fotos del PCB ensamblado sin el display ni la placa Nucleo conectados (lado superior y lado inferior respectivamente). Por último, en la figura 3.1c se puede ver la placa con ambos módulos conectados.



(A) Lado superior del PCB.



(B) Lado inferior del PCB.



(C) Placa ensamblada.

FIGURA 3.1. Placa fabricada para el trabajo.

3.2. 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 y a la pantalla LCD como periférico de entrada y salida.

3.2.1. Diagrama del dispositivo

La figura 3.2 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.

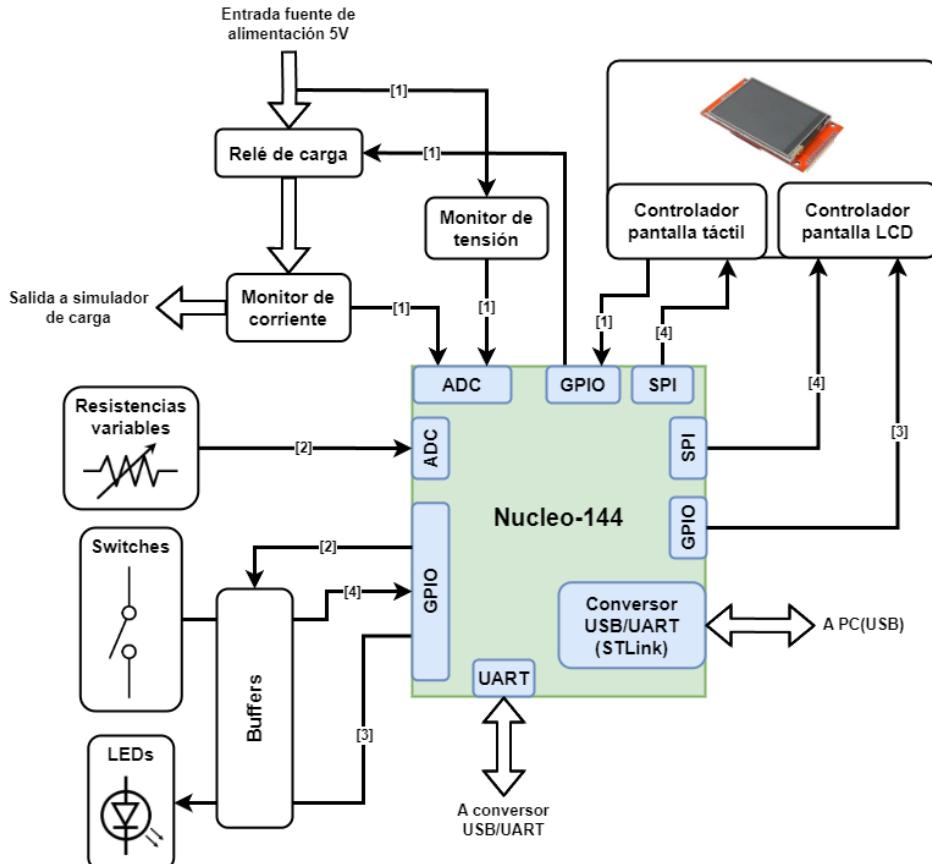


FIGURA 3.2. Diagrama completo del hardware implementado.

3.2.2. 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 más chico sea su valor más despreciables son 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 ésta tensión diferencial y la conecta a uno de sus pines de salida.

Al ser éste 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 3.3.

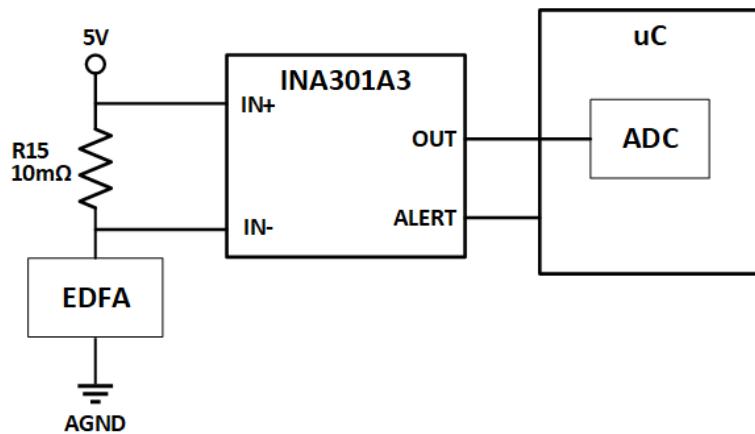


FIGURA 3.3. Esquema del monitor de corriente

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.

3.2.3. Relé de alimentación

La línea de 5 V de alimentación del EDFA es controlada mediante la activación de un relé (código APAN3103), 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 36.7 mA para activarse, no se puede conectar directamente a un pin de salida del microcontrolador ya que no puede suministrar la corriente necesaria. Por ésta razón la activación se realiza mediante un transistor (código MMBT2222), funcionando como una llave (en modo saturación o corte). De ésta forma, la corriente para la activación de la bobina del relé la provee directamente la fuente de alimentación de 3.3 V.

3.2.4. Medición de tensión del amplificador

Hacer una medición de la tensión de la línea es mucho mas 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 3.4 se puede ver el circuito implementado.

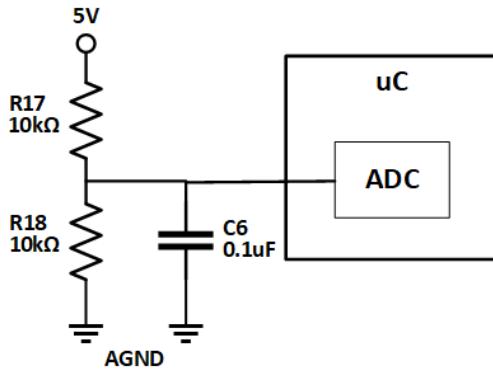


FIGURA 3.4. Circuito del monitor de corriente.

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

3.2.5. Conexión a pantalla LCD

Como se puede ver en la figura 3.2, 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 de la forma en que se muestra en la figura 2.8, se decidió usar uno para cada uno de forma individual. La principal razón de ésto 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. Ésta 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. Éstas 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*)

3.2.6. 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.

Éstas 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). Los objetivos de ésta 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 ésta 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) 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 éstas (código AA3528LVBS/D).

3.2.7. 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. Éste 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 éstas señales se utilizaron dos resistores variables (código PTV09A-4020U-B103) que permiten variar de forma manual el valor de tensión entre 0 V y 3.3 V.

3.2.8. Interfaz UART

Como se puede ver en la figura 3.2, 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.

3.3. 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. Estas tareas se pueden asumir que se ejecutan de forma independiente y simultánea pero, pero compartiendo recursos entre ambas (variables).

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.

Por otro lado, el proceso del frontend esta asociado al display LCD y tiene como objetivo principal interactuar directamente con el usuario. Para ello debe leer la información de salida del backend, presentarla en la pantalla y también detectar cuando (y donde) el usuario presiona sobre ella y enviar estos datos al backend (recursos compartidos).

En la figura 3.5 se puede ver un diagrama de como se relacionan ambas tareas entre si y con el hardware.

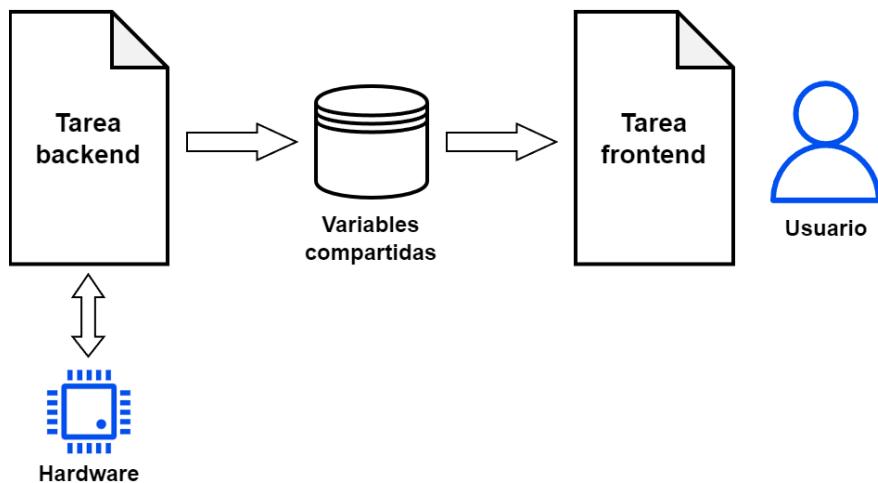


FIGURA 3.5. Arquitectura de firmware.

3.3.1. Máquina de estados

La máquina de estados implementada en el firmware representa de forma análoga el estado del amplificador en todo momento. Los estados definidos para este dentro del programa son:

- Desconectado: alimentación del amplificador desconectada (relé desactivado)
- 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 dependiendo del estado en que se encuentre la máquina. La cantidad de eventos que pueden llegar a la máquina son muchos y por simplicidad solo se mencionan algunos a modo de ejemplo:

- Eventos de la alimentación: sobretensión, sobrecorriente, etc.

- 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 3.6 se puede ver un diagrama de transiciones simplificado de la máquina de estados implementada (estados y eventos).

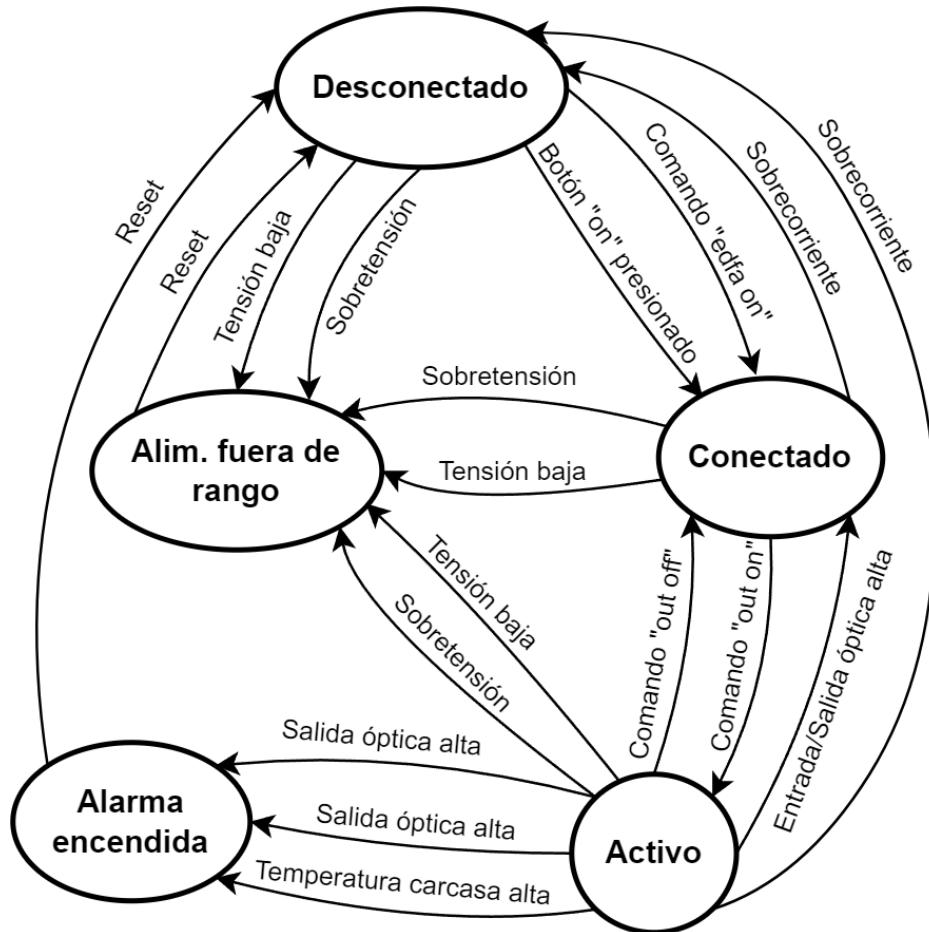


FIGURA 3.6. Diagrama simplificado de la máquina de estados.

3.3.2. Capa de abstracción del amplificador

Con el objetivo de generar una estructura jerárquica por capas, se decidió crear un *wrapper* que se ubique 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 mas de abstracción que provee un conjunto de funciones o servicios para interactuar con el EDFA, funcionando a modo de puente entre éste y la aplicación de usuario (backend). En la figura 3.7 se puede ver la estructura de esta capa junto con sus distintos niveles.

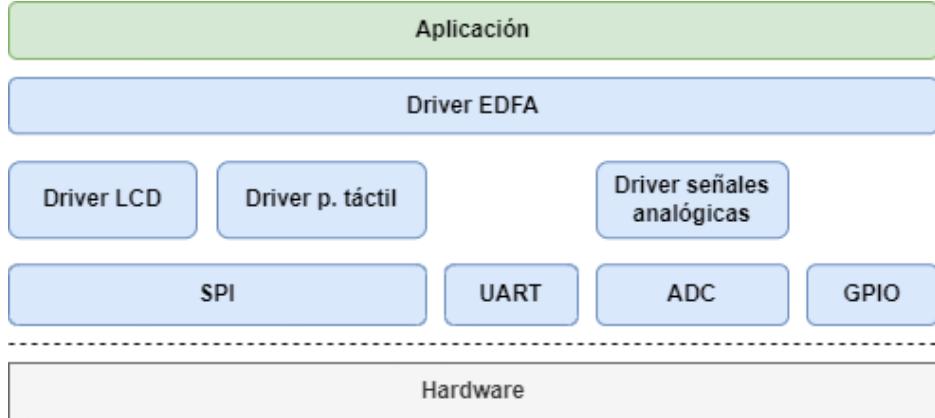


FIGURA 3.7. Capa de abstracción del EDFA.

3.3.3. 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

Siendo 12 la cantidad de bits con la que trabaja el ADC y la tensión de referencia del microcontrolador de 3.3 V se tienen 4095 valores de tensión representables, lo que da una resolución de 0.8 mV.

3.3.4. Driver de la pantalla LCD

Como se mencionó en la sección 3.2.5, 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

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. 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, para poder ser legibles, sus valores se actualizan solamente dos veces por segundo.

3.3.5. 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 se guardan en el buffer de entrada hasta que la linea de datos pasa a estar inactiva.

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

3.3.6. 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.

La consola corre sobre la tarea del backend y puede injectar eventos a la máquina de estados. Su funcionamiento es similar al de cualquier consola, usando un esquema de comando y argumento. 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

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 400W
Resistencia de potencia variable	AVT05006E25R00KE	Resistencia de potencia de alambre bobinado de 25Ohm/50W
Potenciómetro	3590S-1-201L	Potenciómetro de 200Ohm/2W 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 2.5mm2
PC	-	Computadora de escritorio con software PuTTY (cliente de terminal)

4.2. Banco de ensayos

En la figura 4.1 se puede ver una foto de

4.3. Pruebas de hardware

en este caso, el diseño del dispositivo no es complejo y el funcionamiento de las distintas partes del hardware es trivial. En general, solo basta con medir continuidad o valores de tensión en ciertos puntos para concluir que funcionan correctamente. Los únicos casos para los cuales es necesario realizar una prueba

4.3.1. Pruebas del monitor de corriente

4.3.2. Pruebas del relé de alimentación

4.4. Pruebas de firmware

4.4.1. Pruebas del monitor de tensión

4.4.2. Pruebas del monitor de corriente

4.4.3. Pruebas de la pantalla LCD

4.4.4. Pruebas de la pantalla táctil

4.4.5. Pruebas de la comunicación UART

4.4.6. Pruebas de las alarmas y señales de control

4.5. Pruebas de integración

Las pruebas de integración tienen como objetivo probar la interacción del hardware y el firmware entre sí, es decir, el funcionamiento del sistema como una unidad.

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 proyecto 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 3.1, 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 el desarrollo del mismo

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 éstos se encuentran:

- Rediseñar y fabricar la versión final del PCB corrigiendo los errores encontrados y aplicando las modificaciones necesarias para poder conectarlo a un EDFA y montarse sobre la carcasa. También se deberá incluir el microcontrolador de la placa Nucleo o un equivalente junto con su interfaz de programación
- 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

Círculo esquemático completo

Write your Appendix content here.

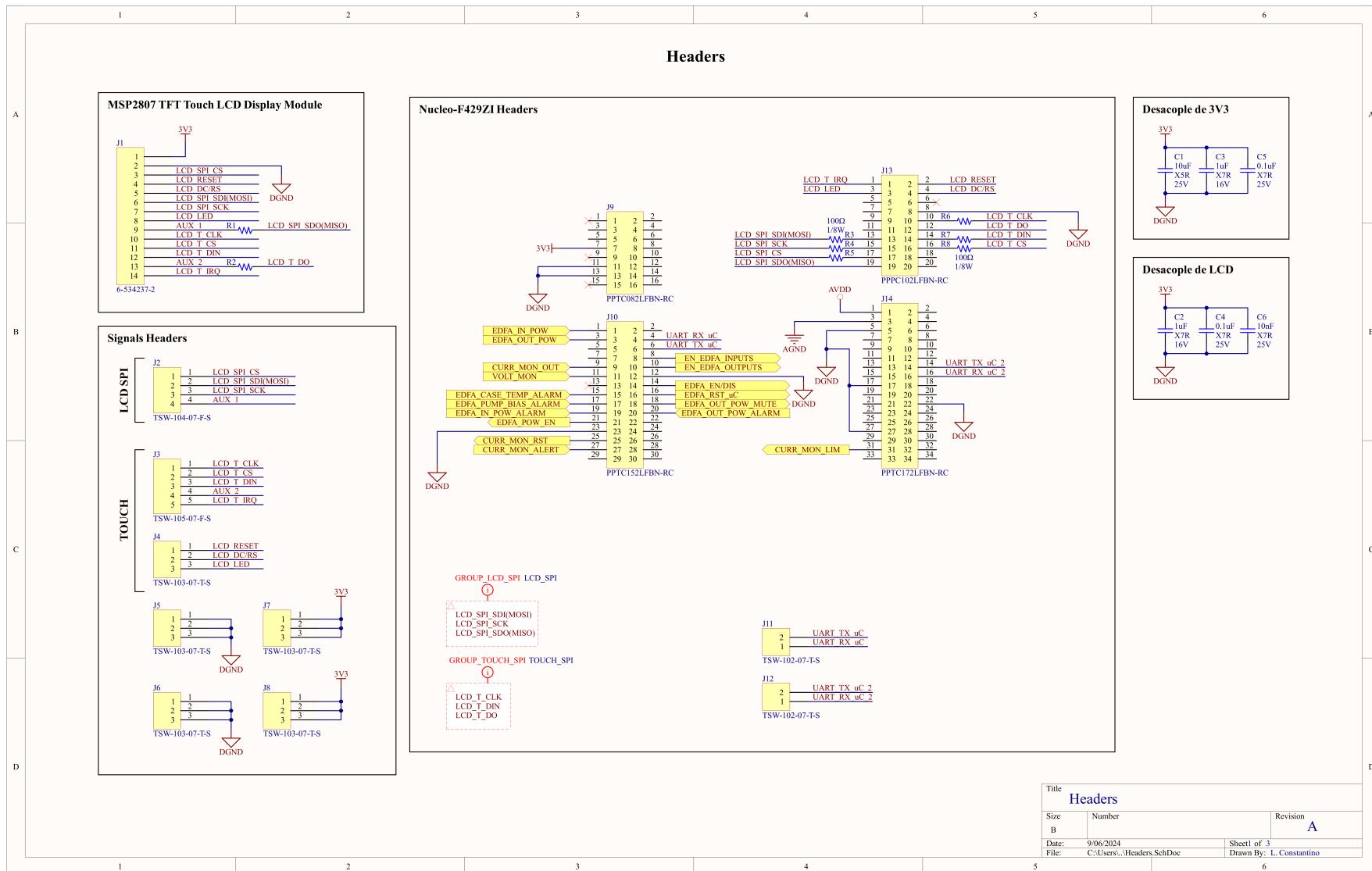


FIGURA A.1. Circuito esquemático. Conectores.

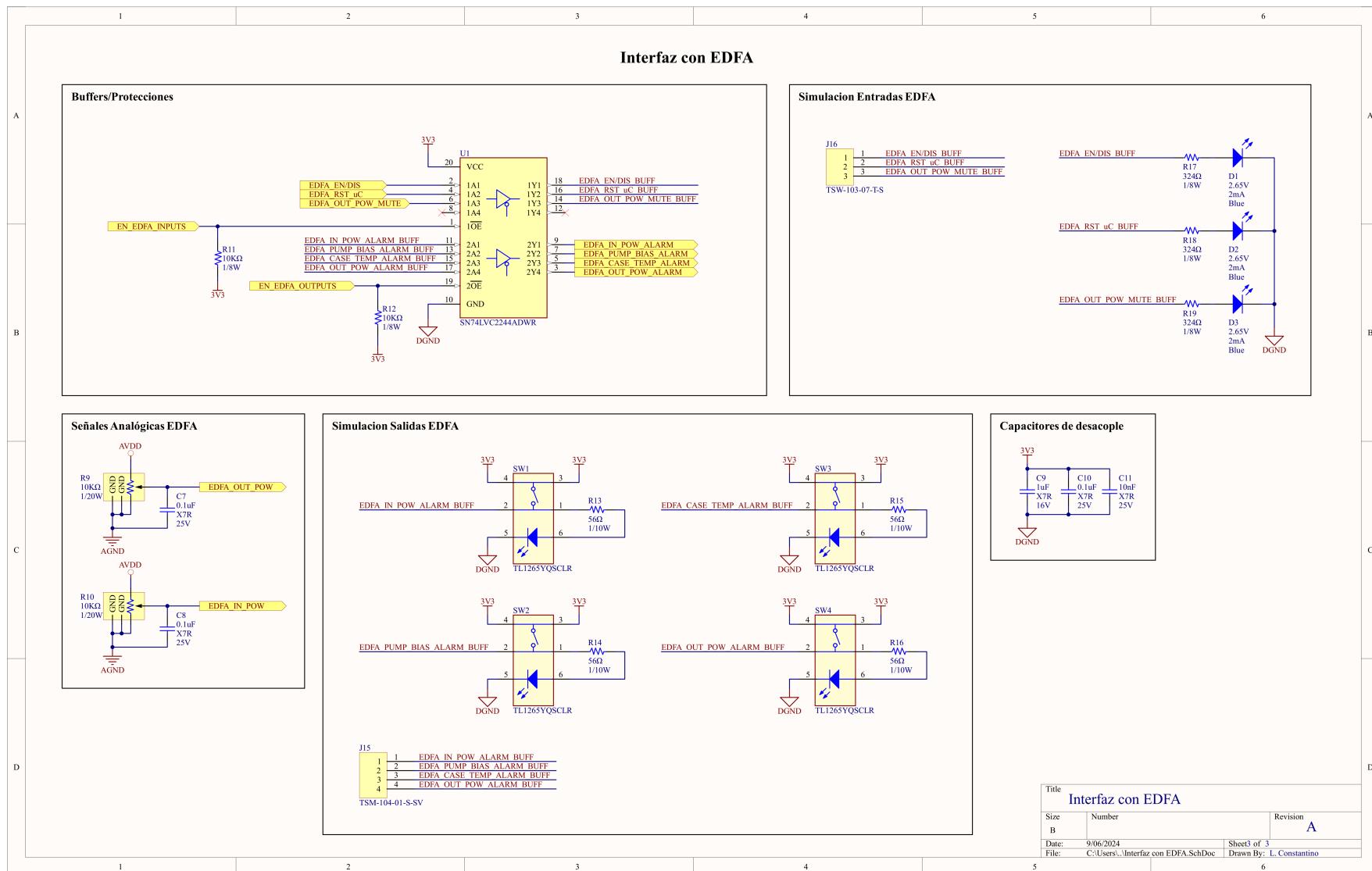


FIGURA A.2. Circuito esquemático. Interfaz con EDFA.

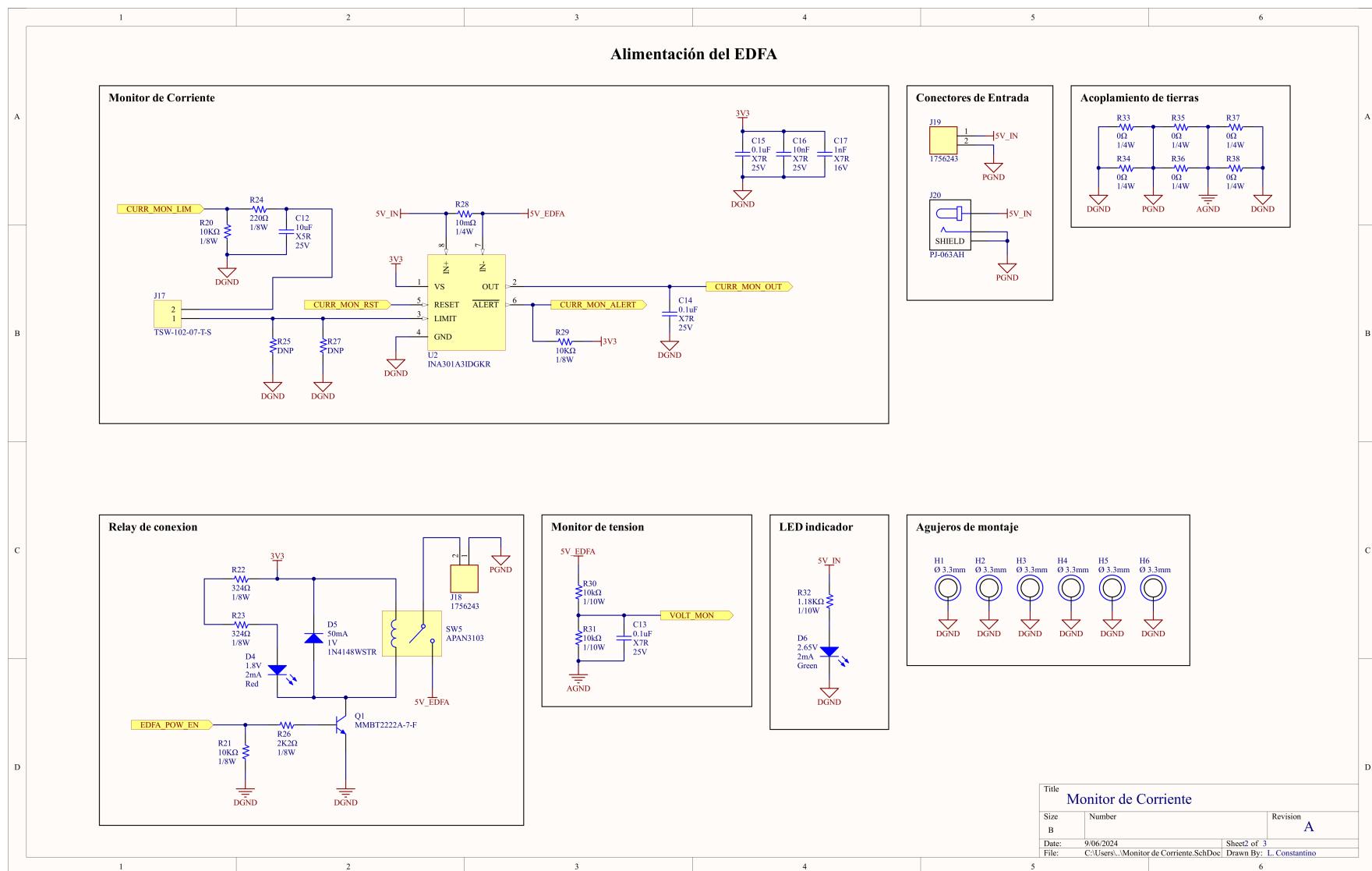


FIGURA A.3. Circuito esquemático. Alimentación del EDFA.

Bibliografía

- [1] *Skyloom Global*. URL: <https://www.skyloom.co/> (visitado 22-03-2023).
- [2] 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>.
- [3] Wikipedia. *Fibra óptica*. URL: https://es.wikipedia.org/wiki/Fibra_%C3%B3ptica (visitado 14-04-2023).
- [4] Wikipedia. *Stimulated emission*. URL: https://en.wikipedia.org/wiki/Stimulated_emission (visitado 14-04-2023).
- [5] RP Photonics Encyclopedia. *Erbium-doped Fiber Amplifiers*. URL: https://www.rp-photonics.com/erbium_doped_fiber_amplifiers.html (visitado 14-04-2023).
- [6] Fiber Optics Share. *Basic Knowledge About EDFA*. URL: <https://www.fiberopticsshare.com/basic-knowledge-edfa.html> (visitado 14-04-2023).
- [7] ST Microelectronics. *STM32F429ZI*. URL: <https://www.st.com/resource/en/datasheet/stm32f429zi.pdf>.
- [8] ST Microelectronics. *Nucleo-144*. URL: <https://www.farnell.com/datasheets/2014265.pdf>.
- [9] Texas Instruments. *INA301*. URL: <https://www.ti.com/lit/ds/symlink/ina301.pdf?src=supplier=Digi-Key>.
- [10] LCDWiki. *MSP2807*. URL: http://www.lcdwiki.com/2.8inch_SPI_Module_ILI9341_SKU:MSP2807 (visitado 07-04-2023).
- [11] Wikipedia. *FreeRTOS*. URL: <https://en.wikipedia.org/wiki/FreeRTOS> (visitado 30-03-2023).
- [12] AWS. *FreeRTOS*. URL: <https://docs.aws.amazon.com/freertos/latest/userguide/dev-guide-freertos-kernel.html> (visitado 07-04-2023).
- [13] Wikipedia. *Hardware abstraction*. URL: https://en.wikipedia.org/wiki/Hardware_abstraction (visitado 11-04-2023).
- [14] 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 (visitado 13-04-2023).
- [15] Wikipedia. *Analog-to-digital converter*. URL: https://en.wikipedia.org/wiki/Analog-to-digital_converter (visitado 31-03-2023).
- [16] Embedded. *UART*. URL: <https://www.embedded.com/understanding-the-uart/> (visitado 02-04-2023).

- [17] Wikipedia. *SPI*. URL:
https://en.wikipedia.org/wiki/Serial_Peripheral_Interface (visitado 13-04-2023).