

**Universidad Tecnológica Nacional**

Proyecto Final

---

## Analizador Lógico y de Protocolo

---

*Autor:*

- *Hacen, Karim Isaac*

*Director:*

- *Yarce, Gustavo Adalberto*

*Proyecto final presentado para cumplimentar los requisitos académicos  
para acceder al título de Ingeniero Electrónico*

*en la*

**Facultad Regional Paraná**

Agosto 2024

# Declaración de autoría

Yo declaro que el Proyecto Final “Analizador Lógico y de Protocolo” y el trabajo realizado son propios. Declaro:

- Este trabajo fue realizado en su totalidad, o principalmente, para acceder al título de grado de Ingeniero Electrónico, en la Universidad Tecnológica Nacional, Regional Paraná.
- Se establece claramente que el desarrollo realizado y el informe que lo acompaña no han sido previamente utilizados para acceder a otro título de grado o pre-grado.
- Siempre que se ha utilizado trabajo de otros autores, el mismo ha sido correctamente citado. El resto del trabajo es de autoría propia.
- Se ha indicado y agradecido correctamente a todos aquellos que han colaborado con el presente trabajo.
- Cuando el trabajo forma parte de un trabajo de mayores dimensiones donde han participado otras personas, se ha indicado claramente el alcance del trabajo realizado.

Firmas:

- 
- 

Fecha:

# Agradecimientos

A mi director, Ing. Gustavo Yarce, por su predisposición, dedicación y compromiso con el proyecto.

A mi familia, por su fundamental apoyo e indudable confianza en mí, a lo largo de esta etapa.

A mis amigos, por la asistencia prestada y sus palabras de aliento durante el desarrollo de este proyecto.

A mi pareja, Abril, por su compañía, aliento y apoyo emocional durante toda la carrera.

A la Facultad Regional Paraná y su comunidad, por la formación académica y profesional brindada.

Hacen, Karim Isaac

Universidad Tecnológica Nacional

## *Abstract*

Facultad Regional Paraná

Ingeniero en Electrónica

# **Analizador Lógico y de Protocolo**

Hacen, Karim Isaac

### **Abstract:**

With the purpose of developing an instrument capable of sampling, storing, illustrating and decoding digital signals from communication protocols such as Binary, UART, SPI, I<sup>2</sup>C, a Logic and Protocol Analyzer device and PC GUI were developed.

The acquisition system was implemented using MAX9108 analog comparators, an ADC1175 Flash ADC, and a RP2040 microcontroller, programmed in C and Assembly, using the SDK provided by the vendor. The GUI was developed in Python 3 with the PySide6, Matplotlib and NumPy modules.

Digital sampling frequencies of 1 [GSa/s] were accomplished when using 8 channels, with a fastest digital signal of 1 [MHz], besides triggered sampling and logic level selection features, as well as analog sampling frequencies of 25 [MHz] with 8 bits resolution.

### **Keywords:**

Communication protocols, Digital Sampling, Analog Comparators

**Resumen:**

Con el fin de obtener un instrumento capaz de muestrear, almacenar, visualizar y decodificar señales digitales pertenecientes a protocolos de comunicación como Binario Simple, UART, SPI e I<sup>2</sup>C se desarrolló un dispositivo Analizador Lógico y de Protocolo, junto con su GUI en PC.

El sistema de adquisición fue implementado utilizando comparadores analógicos MAX9108, un ADC Flash ADC1175 y un microcontrolador RP2040, programado en C y Assembly, utilizando el SDK provisto por el fabricante. La GUI fue desarrollada en Python 3, con los módulos PySide6, Matplotlib y NumPy.

Se obtuvieron frecuencias de muestreo digitales de 1 [GSa/s] utilizando 8 canales, con señal digital más rápida de 1 [MHz], con capacidad de realizar muestreos gatillados y ajustar el nivel de comparación a voluntad, así como frecuencias de muestreo analógicas de 25 [MHz] con 8 bits de resolución.

**Palabras Clave:**

Protocolos de comunicación, Muestreo Digital, Comparadores Analógicos.

# Índice

Declaración de autoría.....	2
Agradecimientos.....	3
Índice.....	6
Lista de Figuras.....	8
Lista de Tablas.....	10
Lista de Abreviaciones y Símbolos.....	11
Dedicado a:.....	13
Capítulo 1: Introducción.....	14
1.1 ¿Cómo se transmite la información?.....	14
1.2 ¿Qué comunicaciones estudiaremos?.....	14
1.3 ¿Por qué un Analizador Lógico?.....	15
1.4 ¿Qué hay disponible en el mercado?.....	16
1.5 ¿Qué objetivos nos proponemos?.....	16
Capítulo 2: Desarrollo.....	18
2.1 Descripción del Sistema.....	18
2.2 Hardware.....	18
2.2.1 Atenuadores.....	18
2.2.2 Protecciones contra sub y sobretensión.....	19
2.2.3 Comparadores Analógicos.....	22
2.2.4 Conversor Digital Analógico.....	23
2.2.5 Conversor Analógico Digital.....	24
2.2.6 Microcontrolador.....	27
2.2.7 Raspberry Pi Pico.....	31
2.2.8 Botón de Reset.....	33
2.2.9 Alimentación.....	33
2.2.10 PCB.....	34
2.2.11 Resistores Pull-Down.....	36
2.2.12 Gabinete.....	37
2.3 Firmware.....	38
2.3.1 Requerimientos.....	38
2.3.2 Descripción General.....	38
2.3.3 Protocolo de comunicación.....	40
2.4 Software GUI.....	42
2.4.1 Diagramas de Flujo.....	45
2.4.2 Cursor.....	47
2.4.3 Mediciones Fijas.....	47
2.4.4 Buscador de Eventos.....	48
2.4.5 Canal Analógico.....	48
2.4.6 Navegación.....	49
2.4.7 Análisis de protocolos.....	49
Capítulo 3: Resultado.....	53
3.1 Pruebas de Firmware/Hardware.....	53
3.1.1 Captura Digital.....	53

3.1.2 Captura analógica .....	54
3.1.3 Nivel de Comparación .....	55
3.1.4 Muestreo Gatillado .....	56
3.2 Protocolo .....	56
3.2.1 Binario .....	57
3.2.2 UART .....	58
3.2.3 SPI.....	58
3.2.4 I <sup>2</sup> C.....	59
Capítulo 4: Análisis de Costos .....	62
4.1 Presupuesto .....	62
Capítulo 5: Conclusiones.....	64
5.1 Objetivos Cumplidos .....	64
5.2 Futuras Mejoras .....	65
Capítulo 6: Referencias .....	66
Capítulo 7: Anexos .....	68
7.1 Esquemático PCB.....	68
7.2 Programas de prueba .....	68

# Lista de Figuras

Figura 1. Velocidad Promedio Acceso a Internet.....	14
Figura 2. Señal cuadrada.....	15
Figura 3. Diagrama de bloques general.....	18
Figura 4. Esquemático PCB Atenuadores.....	19
Figura 5. PCB Atenuadores .....	19
Figura 6. Protecciones contra tensión excesiva.....	20
Figura 7. PCB Protecciones Tensión .....	21
Figura 8. Protección contra sobretensión .....	21
Figura 9. Operacional como Comparador.....	22
Figura 10. Pruebas MAX9108.....	23
Figura 11. Acondicionamiento Señal ADC .....	25
Figura 12. Canal Analógico.....	27
Figura 13. Diagrama de Bloques RP2040. ....	28
Figura 14. Máquina de estado PIO.....	29
Figura 15. Periférico PIO. ....	29
Figura 16. Diagrama de Bloques ADC RP2040 .....	30
Figura 17. Diagrama de Conexión de Buses con Roles de Componentes.....	31
Figura 18. Footprint recomendado para el RP2040.....	32
Figura 19. Raspberry Pi Pico Rev3.....	32
Figura 20. Antirrebote Reset.....	33
Figura 21. Alimentación PCB.....	33
Figura 22. Capa superior diseño PCB y Vista superior modelo 3D PCB .....	35
Figura 23. PCB con sus componentes soldados .....	35
Figura 24. PCB Pull-down adicional .....	36
Figura 25. Parte Inferior Gabinete .....	37
Figura 26. Parte Superior Gabinete .....	37
Figura 27. Ensamblado Final .....	37
Figura 28. Diagrama de Flujo General Firmware.....	38
Figura 29. Diagrama de Flujo "Ejecutar Orden" .....	39
Figura 30. Protocolo de Comunicación.....	41
Figura 31. GUI antes del muestreo.....	43
Figura 32. Panel Lateral GUI .....	44
Figura 33. Diagrama de Flujo General GUI .....	45
Figura 34. Diagramas de Flujo ISR GUI .....	46
Figura 35. Contador de Rizado con Cursor Activo.....	47
Figura 36. Contador de Rizado con Mediciones Fijas .....	48
Figura 37. Canal Analógico con Cursor y Mediciones Fijas.....	49
Figura 38. Trama UART.....	50
Figura 39. Trama SPI Modo 0.....	51
Figura 40. Trama I <sup>2</sup> C.....	52
Figura 41. Prueba Captura Digital (correcta) .....	53
Figura 42. Prueba Captura Digital (incorrecta) .....	54
Figura 43. Señales de Prueba Captura Analógica .....	54
Figura 44. Prueba Captura Analógica.....	55



Figura 45. Prueba Comparadores Analógicos .....	55
Figura 46. Prueba Muestreo Gatillado .....	56
Figura 47. Prueba Decodificar Binario .....	57
Figura 48. Prueba Decodificador UART.....	58
Figura 49. Prueba Decodificador SPI .....	59
Figura 50. Prueba Decodificado I <sup>2</sup> C Orden Lectura .....	60
Figura 51. Prueba Decodificado I <sup>2</sup> C Respuesta Lectura.....	60
Figura 52. Prueba Decodificado I <sup>2</sup> C Transacción Completa .....	61
Figura 53. Esquemático Completo.....	68

## Lista de Tablas

Tabla 1. Comparativa Analizadores Comerciales.....	16
Tabla 2. Objetivos Propuestos vs Opción Comercial. ....	16
Tabla 3. Comparativa Amplificadores Operacionales.....	26
Tabla 4. Codificación Modos de Gatillado .....	40
Tabla 5. Codificación Frecuencias de Muestreo .....	40
Tabla 6. Identificadores de Error .....	42
Tabla 7. Modos SPI.....	51
Tabla 8. Presupuesto del Proyecto .....	62
Tabla 9. Objetivos planteados y su grado de cumplimiento .....	64
Tabla 10. Comparativa con alternativas comerciales.....	64

# Lista de Abreviaciones y Símbolos

En orden de aparición

## Abreviaciones

- OSI - Open Systems Interconnection
- ADC - Analog to Digital converter
- DAC - Digital to Analog Converter
- UART - Universal Asynchronous Receiver-Transmitter
- SPI - Serial Peripheral Interface
- I<sup>2</sup>C - Inter-Integrated Circuit
- PC - Personal Computer
- GUI - Graphical User Interface
- SDK - Software Development Kit
- USB - Universal Serial Bus
- MCU - Micro Controller Unit
- PCB - Printed Circuit Board
- VNA - Vectorial Network Analyzer
- GND - Ground
- SOT - Small Outline Transistor
- SOIC - Small Outline Integrated Circuit
- TSSOP - Thin Shrink Small Outline Package
- CLK - Clock
- VCC - Alimentación
- LC - Inductor-Capacitor
- GPIO - General Purpose Input Output
- PIO - Programmable Input Output
- SRAM - Static Random Access Memory
- RAM - Random Access Memory
- PLL - Phase Locked Loop
- QSPI - Quad Serial Peripheral Interface
- IO - Input Output
- DMA - Direct Memory Access
- FIFO - First Input, First Output
- IRQ - Interrupt Request
- DC - Direct Current
- QFN - Quad Flat No-Leads
- SMPS - Switching Mode Power Supply
- PTH - Plated Through Hole
- NPTH - Non-Plated Through Hole
- SQL - Structured Query Language
- API - Application Programming Interface
- WMI - Windows Management Instrumentation
- ISR - Interrupt Service Routine

- MOSI - Master Output Slave Input
- MISO - Master Input Slave Output
- SCK - Slave Clock
- CS - Chip Select
- SS - Slave Select
- SDA - Serial Data
- SCL - Serial Clock
- ACK - Acknowledge
- R/W - Read/Write
- R&D - Research and Development
- PDF - Portable Digital File

### **Magnitudes**


- $I_c$  - Corriente de Colector
- $I_{out}$  - Corriente de Salida
- $V_e$  - Tensión de Emisor
- $V_o$  - Tensión de Salida
- $V_s$  - Tensión de Fuente
- $V_{ss}$  - Tensión de Alimentación Negativa
- G - Ganancia
- T - Temperatura

### **Unidades**

- [Sa/s] - Muestras por Segundo
- [Hz] - Hercios
- [bps] - Bits por Segundo
- [V] - Voltios
- [ $V_{pk}$ ] - Voltios pico
- [ $\Omega$ ] - Ohmios
- USD - Dólares Estadounidenses
- [A] - Amperios
- [s] - Segundos
- [B] - Bytes
- [W] - Vatios
- [F] - Faradios
- [H] - Henrios
- [m] - Metros

## **Dedicado a:**

Mis padres, por su confianza ciega e incondicional apoyo durante toda la carrera.

 <b>UTN Regional Paraná</b>	Analizador Lógico y de Protocolo
Ingeniería en Electrónica	Capítulo 1: Introducción

## Capítulo 1: Introducción

### 1.1 ¿Cómo se transmite la información?

Uno de los campos de mayor aplicación de la electrónica como disciplina son, indefectiblemente, las comunicaciones. Esto se fundamenta en que la demanda por más y mejores métodos de mover información de un punto a otro ha demostrado un crecimiento exponencial, en general motivado por una mayor penetración de servicios como internet fijo y móvil y la velocidad individual de los mismos [1, 2]. En la Figura 1 es posible visualizar una cuantificación de este fenómeno en Argentina.

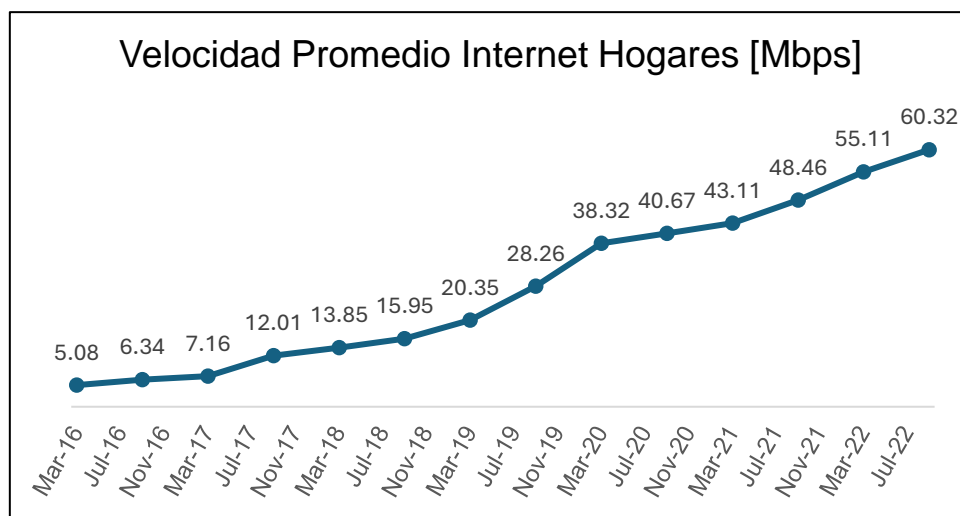


Figura 1. Velocidad Promedio Acceso a Internet.  
Generado con datos de [1, 2]

Para lograr el cometido de transportar información, se pueden aprovechar distintos fenómenos físicos dentro de un medio de enlace, como, por ejemplo:


- Una pulsación de un haz de luz infrarroja altamente coherente en una fibra óptica.
- Un cambio en la amplitud, frecuencia o fase (o una combinación de estas tres propiedades) de una onda electromagnética, en una comunicación inalámbrica.
- Un cambio de amplitud, frecuencia o fase de una señal eléctrica en un elemento conductor.

Es, en particular, de interés para el desarrollo de este proyecto el tercer punto de la lista anterior, como veremos en la siguiente sección.

### 1.2 ¿Qué comunicaciones estudiaremos?

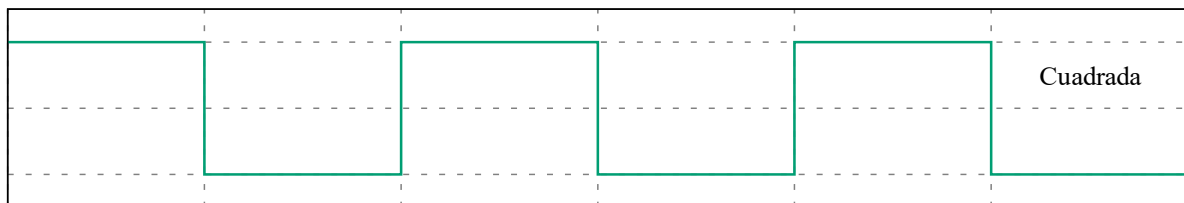
Las restricciones en variables físicas empleadas en la comunicación (léase, la capa 1 del modelo OSI) terminan por adaptarse a las necesidades y capacidades de los sistemas de la información. En el caso de este proyecto, nos enfocaremos en el intercambio de información entre diferentes componentes electrónicos, incluyendo, pero sin limitarse a:

- Microcontroladores (esquemas ordenadores completos).

 <b>UTN Regional Paraná</b>	Analizador Lógico y de Protocolo
Ingeniería en Electrónica	Capítulo 1: Introducción

- Sensores y actuadores (y los circuitos secuenciales que los acompañan).
- Memorias volátiles y no volátiles.
- Periféricos.
- ADC y DAC.

En todos los casos, las señales eléctricas utilizadas son digitales binarias (es decir, que su amplitud oscila entre dos valores fijos). Un ejemplo de ello se puede apreciar en la Figura 2.



*Figura 2. Señal cuadrada.  
Obtenido de [23].*


Las variaciones de varias de estas señales digitales en el tiempo son lo que permite establecer la comunicación entre los componentes antes mencionados.

### 1.3 ¿Por qué un Analizador Lógico?

Los desarrolladores de sistemas embebidos tienen, en la mayoría de los casos, la completa responsabilidad de programar o configurar todos los circuitos electrónicos que intervienen a la hora de comunicar los componentes mencionados en la sección anterior. Como es de esperar, este proceso es muy propenso a errores por los siguientes factores:

- Comprender en máximo detalle (para minimizar la posibilidad de una equivocación) el funcionamiento de los periféricos, protocolos y registros de configuración es poco práctico por la cantidad de tiempo que se debe invertir.
- Aún luego de hacer una investigación extensiva sobre los temas mencionados en el punto anterior y como cada fabricante incluye sus propios periféricos para su plataforma, el conocimiento de una plataforma no suele ser reutilizable al tratar con otra análoga, pero de diferente fabricante.
- Los protocolos de comunicación entre sistemas embebidos suelen ser muy sensibles a ruidos, puesto que no incluyen códigos de detección o corrección de errores (o si lo hacen, es muy escueta).
- Los circuitos lógicos de los periféricos pueden ser algo exigentes con la impedancia del bus (arquitecturas open-drain, por ejemplo).
- Una falla en alguna de las etapas de la comunicación implica un fallo en la totalidad de la comunicación.

Por lo tanto, es **fundamental** que los desarrolladores posean una herramienta que les permita visualizar cómo se comportan las señales digitales binarias que transmiten la información y así, validar el funcionamiento o depurar los errores en una comunicación. **Esa herramienta es el Analizador Lógico y de Protocolo.**

 <b>UTN Regional Paraná</b>	Analizador Lógico y de Protocolo
Ingeniería en Electrónica	Capítulo 1: Introducción

## 1.4 ¿Qué hay disponible en el mercado?

Si se desea adquirir uno de estos equipos, existen alternativas en varios rangos de precios. A continuación, haremos una comparativa de tres productos de variada complejidad.

Parámetro	Valor			Unidad
Fabricante	Kingst	Saleae		-
Modelo	LA1010	Logic 8	Logic Pro 16	-
Frecuencia de Muestreo	100@3Canales	100	500	[MSa/s]
Canales Digitales	16	8	16	-
Señal Digital más rápida	20	25	100	[MHz]
Captura Analógica	No	Si (8 Canales)	Si (16 Canales)	-
Rango de Voltaje	±50	±25	±25	[V]
Umbral de Voltaje (Mín.)	Var.: -4.	Fijo: 0.6	Fijos: 0.6, 1.2, 1.65	[V]
Umbral de Voltaje (Máx.)	Var.: 4.	Fijo: 1.2		[V]
Precio de Venta	68	499	1499	USD

Tabla 1. Comparativa Analizadores Comerciales.

Datos obtenidos de: [3, 4] "Var." significa "variable"

Predeciblemente, a mayores prestaciones, mayor precio. En general, LA1010 satisface la necesidad de un simple analizador lógico para visualizar señales digitales. Por otro lado, si se pueden prescindir de 8 canales y de un voltaje de umbral variable, a cambio de un precio mayor, Logic 8 ofrece captura analógica en todos los canales. Finalmente, a un costo de 1000 USD más, Logic 16 Pro ofrece lo mejor de LA1010 y Logic 8 con muestreos y ancho de banda digital muy superiores.

## 1.5 ¿Qué objetivos nos proponemos?


La intención de este proyecto es lograr un Analizador Lógico y de Protocolo con la mayoría de las prestaciones principales del Saleae Logic 8, al menor costo posible, a saber:

Parámetro	Valor			Unidad
Dispositivo	Mínimos	Objetivos	Logic 8	-
Voltaje de Entrada	0 a 5	0 a 9	±25	[V]
Número de Entradas Digitales	4	8	8	-
Frecuencia de Muestreo Digital	40	1000	100	[MSa/s]
Número de Entradas Analógicas	1	1	8	-
Frecuencia de Muestreo Analógica	5@8 bits	25@8 bits	10@10 bits	[MSa/s]
Umbral de Voltaje (Bajo)	Var.: 0	Var.: 0	Fijo: 0.6	[V]
Umbral de Voltaje (Alto)	Var.:3.3	Var.:3.3	Fijo: 1.2	[V]

Tabla 2. Objetivos Propuestos vs Opción Comercial.

Datos obtenidos de: [4] "Var." significa "variable"




 <b>UTN Regional Paraná</b>	Analizador Lógico y de Protocolo
Ingeniería en Electrónica	Capítulo 1: Introducción

Además de los objetivos de hardware y firmware, se suma el requerimiento de una aplicación de escritorio que funcione a modo de GUI para comandar el dispositivo, permita graficar las señales muestreadas y decodificar los siguientes protocolos de comunicación:

- Binario Simple
- UART
- SPI
- I<sup>2</sup>C

Esto último, representa el “análisis de protocolos” que el nombre del proyecto infiere. En la Figura 23 se puede apreciar el prototipo finalizado.

 <b>UTN Regional Paraná</b>	Analizador Lógico y de Protocolo
Ingeniería en Electrónica	Capítulo 2: Desarrollo

## Capítulo 2: Desarrollo

### 2.1 Descripción del Sistema

A modo ilustrativo, se provee el siguiente diagrama de bloques del dispositivo a implementar:

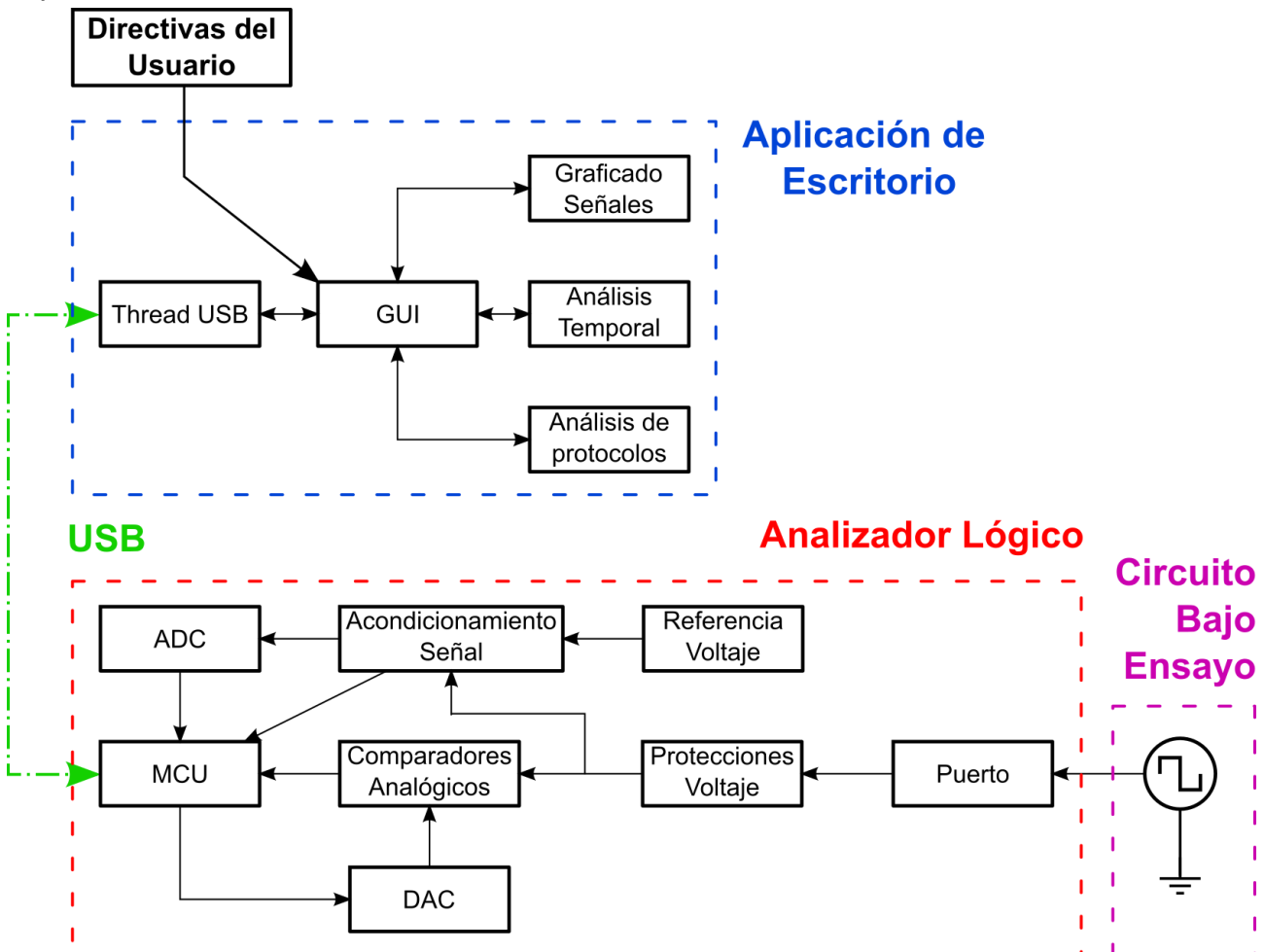


Figura 3. Diagrama de bloques general


En este capítulo detallaremos cada una de las partes de este diagrama en tres segmentos diferentes: Hardware, Firmware y Software. Se presentarán los circuitos, pruebas y líneas de razonamiento generales que fundamentaron las decisiones tomadas en el proceso de diseño y desarrollo del dispositivo.

### 2.2 Hardware

Comencemos analizando el camino que toman las señales de interés desde el circuito bajo ensayo, hasta el MCU que las adquiere, teniendo en cuenta los requerimientos de diseño que se plantearon en la Tabla 2.

#### 2.2.1 Atenuadores

En primer lugar, para lograr muestrear señales en el rango de tensión objetivo, 0 a 9 [V], y considerando la tensión máxima de entrada de la mayoría de los circuitos integrados,

 <b>UTN Regional Paraná</b>	Analizador Lógico y de Protocolo
Ingeniería en Electrónica	Capítulo 2: Desarrollo

unos 5 a 6 [V], queda claro que es necesario atenuar las señales por un factor de aproximadamente 2. Para lograr esto, previo al desarrollo de este proyecto se había ideado utilizar la alternativa más simple de todas las posibles: un divisor resistivo. A modo de pruebas preliminares, se procuró el siguiente circuito y PCB, con resistores de 100 [KΩ].

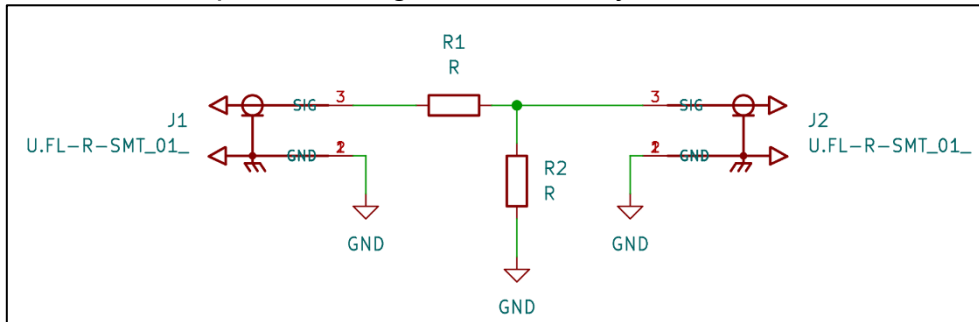


Figura 4. Esquemático PCB Atenuadores



Figura 5. PCB Atenuadores

Seguidamente, se conectó un Analizador Vectorial de Redes (NanoVNA), con el fin de estudiar el comportamiento del atenuador como un cuadripolo. De la prueba realizada, se extrae la conclusión que una impedancia tan alta de entrada imposibilita la carga de las capacidades parásitas de los dispositivos electrónicos inmediatamente posteriores a los atenuadores, que es necesario para que dichos dispositivos representen fielmente la tensión en su entrada. Asimismo, si bien este problema no existiría con valores de resistencia más bajos, se presume que 100 [KΩ] es el mínimo de impedancia de entrada tolerable para un dispositivo como el diseñado, puesto que valores menores influirían en el comportamiento de los circuitos que se le conecten para ensayar. Por este motivo, se decidió prescindir de atenuadores para el hardware del proyecto.

### 2.2.2 Protecciones contra sub y sobretensión

Con la misma necesidad expuesta en 2.2.1, se propuso un nuevo diseño para proteger al circuito de tensiones excesivas, por encima de 5 [V] y por debajo de 0 [V].

Dicho nuevo diseño consiste en:

- Un “Schottky Clamp”, un diodo Schottky en conectado a GND, que se polariza directamente cuando la tensión en un canal es menor a GND, específicamente, más

negativa que la tensión de umbral del diodo (aproximadamente 300 [mV] para el diodo Schottky).

- Un “PNP Active Clamp”, un transistor PNP que pone su juntura en saturación (y, por lo tanto, ofrece un camino de baja impedancia a GND) cuando la tensión en su emisor es mayor o igual a la tensión de su base más su tensión de polarización.

Ambas soluciones se pueden apreciar en la Figura 6. Adicionalmente, como estas protecciones generan un cortocircuito en términos prácticos, se requiere un fusible que interrumpa la conexión rápidamente. Estos fusibles, como el F0201G0R03FNTR\500 de KYOCERA/AVX, existen para las pequeñas corrientes que deberían gatillarlos (decenas de miliamperios) y están disponibles en distribuidores como Mouser Electronics [5], por unos 0.4 U\$D, pero no en Argentina, por lo que no se incorporaron.

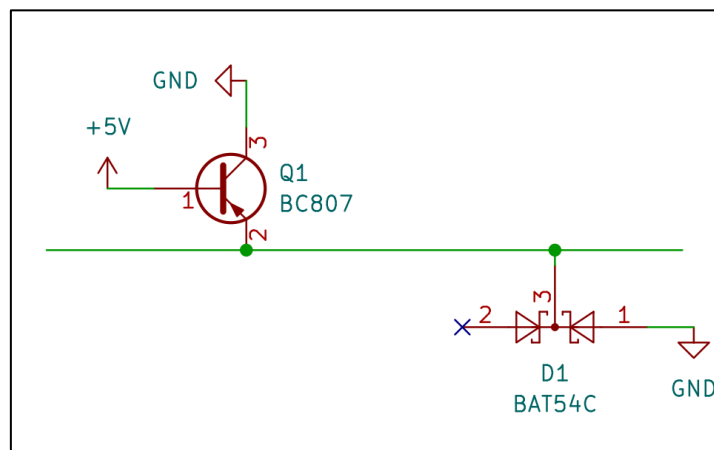



Figura 6. Protecciones contra tensión excesiva

Los dispositivos BC807 (PNP) y BAT54 (Schottky) se eligieron por cumplir los siguientes criterios [5, 6, 7]:

- Ser económicos.
- Estar disponibles en el stock de un proveedor a quién se le adquirieron otros componentes.
- Tener encapsulado SOT-23.
- Tener Corriente Máxima Continua mayor al doble del valor de fusión del fusible antes mencionado.

Ambos dispositivos fueron probados en PCB como se muestra en la Figura 8, y se obtuvieron las siguientes conclusiones:

- El diodo posee una corriente de fuga de 0.32 [ $\mu$ A] para 5 [V] de corriente continua.

 <b>UTN Regional Paraná</b>	Analizador Lógico y de Protocolo
Ingeniería en Electrónica	Capítulo 2: Desarrollo

- Al conectar una señal cuadrada de 3.3 [V<sub>pk</sub>] y 1[MHz], esta no parece ser alterada por la presencia del diodo, indicando que sus capacidades parásitas no interfieren significativamente.

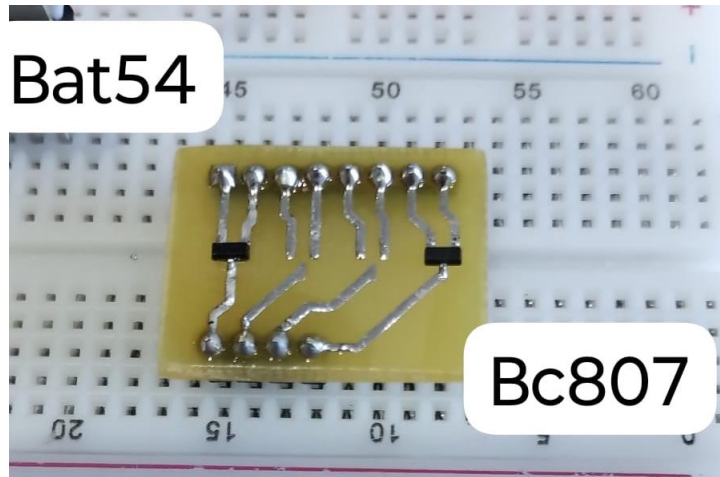


Figura 8. PCB Protecciones Tensión

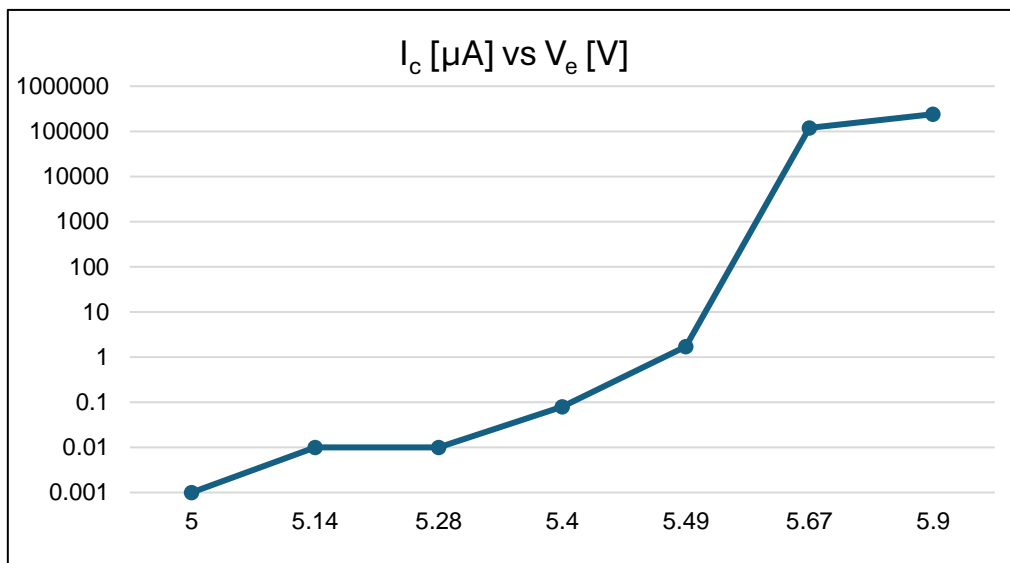



Figura 7. Protección contra sobretensión

- La protección contra sobretensión funciona a los 5.6[V] aproximadamente, como se muestra en Figura 7. Nótese la escala logarítmica del eje de ordenadas.

 <b>UTN Regional Paraná</b>	Analizador Lógico y de Protocolo
Ingeniería en Electrónica	Capítulo 2: Desarrollo

### 2.2.3 Comparadores Analógicos

Luego de garantizar que las señales que el instrumento medirá estarán dentro de sus parámetros de funcionamiento con lo expuesto en 2.2.2, estamos en condiciones de comenzar a procesarlas según sea necesario. En primer lugar, dada la naturaleza digital binaria que tienen las comunicaciones de nuestro interés, es imprescindible poder discernir entre los dos posibles estados, nivel alto y nivel bajo. Aún mejor, si se pudiera definir arbitrariamente la línea que separa dichos estados, sumaría bastante valor agregado al producto. Para ello, en un primer momento se consideró utilizar un amplificador operacional en modo de comparación, como se muestra en la Figura 9:

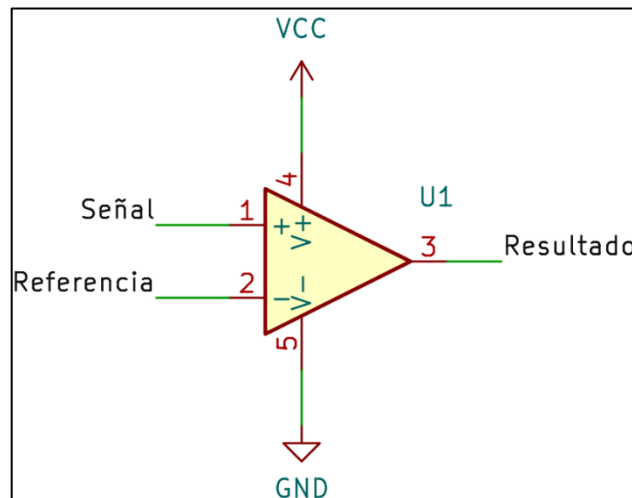



Figura 9. Operacional como Comparador.  
Obtenido de [24].

Donde la tensión en la salida del dispositivo sería  $V_{CC}$  si la tensión de la señal es mayor a la de la referencia y  $GND$  en el caso contrario. Sin embargo, si bien este circuito es simple, barato y conveniente para nuestro diseño, presenta los siguientes problemas [8]:

- Los amplificadores operacionales están diseñados para operar mayoritariamente en un bucle de realimentación (como en la configuración Amplificador No Inversor), haciendo que el rendimiento sea penalizado, particularmente en velocidad de conmutación, al trabajar en un bucle abierto.
- Adicionalmente, aún si trabajan en su zona lineal de tensión, conmutar constantemente entre las tensiones de sus rieles de alimentación implica la saturación de sus etapas internas, penalizando aún más el rendimiento en velocidad de conmutación.
- Si los valores de tensión de entrada son muy similares, se puede incurrir en inestabilidad (cambios abruptos y descontrolados de la tensión de salida) por la baja histéresis que tiene la comparación.

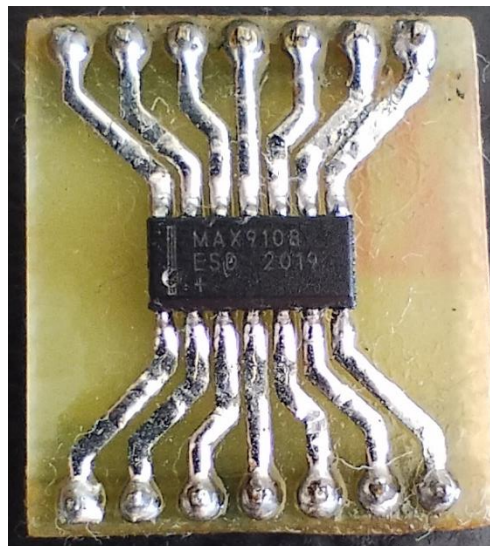
Afortunadamente, una variante de los amplificadores operacionales, llamados **comparadores analógicos**, se enfocan en resolver estos problemas, por lo que fueron seleccionados para el diseño. Al analizar las opciones disponibles en el distribuidor Mouser

 <b>UTN Regional Paraná</b>	Analizador Lógico y de Protocolo
Ingeniería en Electrónica	Capítulo 2: Desarrollo

Electronics, se terminó por seleccionar al integrado MAX9108 por los siguientes motivos [9]:

- Tensión de alimentación recomendada de 5 [V] y GND, simplificando la alimentación.
- Tensión de entrada en todo el rango de alimentación.
- Salida de 3.3 [V], obligatorio para el MCU seleccionado.
- 4 comparadores por integrado, conveniente por las restricciones del régimen de importación de pequeños envíos.
- Encapsulado SOIC-14, dentro de las tolerancias de fabricación de PCB y soldadura.
- 0.7 [mA] de consumo por comparador, ideal para la alimentación por USB.
- Impedancia de entrada superior a 1 [MΩ].
- Tiempos de conmutación de aproximadamente 40 [ns].
- Diferencia de tiempos de propagación de 1 [ns].

De la misma forma que el resto de los componentes, este también se probó en una PCB y protoboard.




*Figura 10. Pruebas MAX9108*

Se probaron, en sus 4 comparadores, señales cuadradas de 3.3 [V] con una referencia de 2 [V] y se observó (con un osciloscopio con 50 [MHz] de ancho de banda) que la máxima frecuencia en la que la integridad de la señal se mantiene es alrededor de **2 [MHz]**. Este es, por lo tanto, el cuello de botella que determina el parámetro: “Señal Digital más rápida” del dispositivo.

#### **2.2.4 Conversor Digital Analógico**

Luego de las exitosas pruebas de los comparadores analógicos, se requiere generar un voltaje de comparación variable para que el usuario delimite dónde termina el nivel alto y comienza el nivel bajo de las señales estudiadas. Para ello se decidió utilizar un Conversor Digital Analógico MCP4821, por los siguientes motivos:

- Ser barato y estar disponible proveedores nacionales.
- 12 bits de resolución, más que suficientes para el rol que cumple.

 <b>UTN Regional Paraná</b>	Analizador Lógico y de Protocolo
Ingeniería en Electrónica	Capítulo 2: Desarrollo

- Referencia interna de voltaje, simplificando su implementación.
- Alimentación de 5 [V] y GND.
- Interfaz SPI, conveniente para el MCU.

Este dispositivo también se ensayó en su PCB y funcionó tal y como se esperaba, por lo que fue incorporado al diseño.

### 2.2.5 Conversor Analógico Digital

Además de los 8 canales digitales, es deseable tener un canal con la capacidad de muestrear tensión y discernir entre más de los dos niveles lógicos provistos por los comparadores analógicos de 2.2.3. El circuito integrado que provee esta funcionalidad es, predeciblemente, un Conversor Analógico Digital y la implementación de esta etapa en el diseño toma dos formas:

#### 2.2.5.1 ADC Discreto

Los ADCs integrados en MCU rara vez son capaces lograr frecuencias de muestreo de superiores a 1 [MHz], por lo que, para lograr una performance adecuada se eligió utilizar un ADC discreto, en particular, el ADC1175 de Texas Instruments, por los siguientes motivos [10]:


- Ser relativamente barato y estar disponible proveedores nacionales.
- Proveer una excelente frecuencia de muestreo máxima garantizada de 20 [MHz], explicado por su topología “Flash”.
- Transmitir los datos como binario simple en puerto paralelo.

Sin embargo, también posee las siguientes desventajas [10]:

- Frecuencia mínima de muestreo de 1 [MHz], haciendo necesaria circuitería adicional para frecuencias menores.
- Encapsulado TSSOP, de dimensiones físicas muy cercanas a las tolerancias de fabricación de PCB y soldadura disponibles.
- Requiere señal de CLK, alimentación filtrada por varios inductores y referencias para la conversión externas.
- Rango de tensión de entrada máximo de 2.7 [V]

De todos modos, se consideró que las bondades provistas por el circuito integrado justifican el costo técnico de las desventajas.



 <b>UTN Regional Paraná</b>	Analizador Lógico y de Protocolo
Ingeniería en Electrónica	Capítulo 2: Desarrollo

### 2.2.5.2 Acondicionamiento de Señal

Como se explicó en 2.2.5.1, el rango de voltaje de entrada del ADC1175 es de 2.7 [V], por lo que si se desean medir señales en el rango de voltaje de entrada del instrumento (0 a 5 [V]), se requiere una atenuación de dichas señales. Adicionalmente, se debe tener en cuenta que el instrumento debe presentar una alta impedancia de entrada al circuito bajo ensayo (para no alterar su funcionamiento), y una baja impedancia de fuente al ADC (para cargar rápidamente la capacidad parásita de entrada y el capacitor del circuito “sample and hold”). Para lograr este propósito, se utilizó el circuito de la Figura 11, donde IC2 es un amplificador operacional en modo seguidor de tensión. Los valores del divisor resistivos fueron elegidos para atenuar la señal de entrada a la mitad y mantener la impedancia de fuente que observará el ADC relativamente baja.

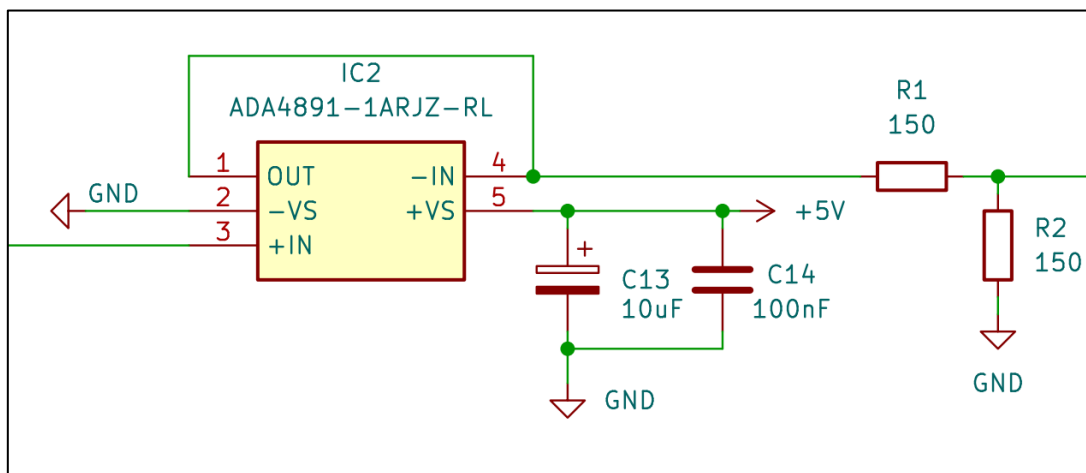


Figura 11. Acondicionamiento Señal ADC

En cuanto al amplificador operacional, se consideraron varias alternativas, comenzando por el LMH6702 de Texas Instruments, por su producto ganancia-ancho de banda máximo de 1.7 [GHz] [11] y relativo bajo costo de 3.31 U\$D. No obstante, su funcionamiento no fue satisfactorio durante la etapa de pruebas por una propiedad ignorada al momento de su selección: el **rango de voltaje de entrada/salida de modo común** (el rango de voltajes en el cual el operacional se comporta de forma lineal) del dispositivo, unos  $\pm 2.2$  [V] al ser alimentado con  $\pm 5$  [V].

Con los aprendizajes adquiridos, se decidió utilizar un Amplificador Operacional “Rail-to-Rail Input/Output”, que suelen tener un rango de voltaje de entrada/salida de modo común igual que su alimentación o muy cercana a ella. Se consideraron varios candidatos, cuyas características se muestran en la Tabla 3, seleccionándose el ADA4891-1 por los siguientes motivos:

- Muy buena relación precio-calidad.
- Excelente rango de voltaje de entrada/salida de modo común para alimentación 0 a 5 [V].
- Buena respuesta en frecuencia para la configuración seguidor de tensión.
- Óptima impedancia y capacitancia de entrada.
- Encapsulado SOT-23-5, dentro de tolerancias de fabricación.




Especificaciones Técnicas para las condiciones:  $V_{cc} = 5 [V]$ ,  $V_{ss} = 0 [V]$ ,  $G=1$ ,  $T=25 [C]$ ,  $I_{out} \geq 2.5 [mA]$ .

Modelo	SN10501	OPA357	ADA4891-1	OPA354	OPA810
Marca	Texas Instruments	Texas Instruments	Analog Devices	Texas Instruments	Texas Instruments
Precio [USD]	1	1.74	1.87	2.32	3
Rango de Voltaje de Entrada de Modo Común [V]	1 a 4 typ.	-0.1 a 5.1 minmax.	-0.3 a 5.8 typ.	-0.1 a 5.1 minmax.	-0.2 a 5.2 minmax.
Rango de Voltaje de Salida [V]	0.2 a 4.8 typ.	0.3 a 4.7 minmax.	0.08 a 4.90 typ.	0.3 a 4.7 minmax.	0.15 a 4.85 minmax.
Tiempo de Asentamiento 0.1% [ns]	27, $G=-2$ , $V_o=2[V]$ typ.	30, $G=1$ , $V_o=2[V]$ typ.	27, $G=-2$ , $V_o=2[V]$ typ.	30, $G=1$ , $V_o=2[V]$ typ.	100, $G=2$ , $V_o=2[V]$ , $V_s=\pm 2.5[V]$ typ.
Velocidad de Precesión [V/ $\mu$ s]	750, $G=2$ , $V_o=4[V]$ Step typ.	150, $G=1$ , $V_o=4[V]$ Step typ.	170, $G=2$ , $V_o=2[V]$ Step typ.	150, $G=1$ , $V_o=4[V]$ Step typ.	134, $G=2$ , $V_o=2[V]$ Step typ.
Producto Ganancia-Ancho de Banda [MHz]	170 typ.	250 typ.	240 typ.	250 typ.	133 typ.
Impedancia de Entrada [M $\Omega$ ]	33 typ.	>1000 typ.	>1000 typ.	>1000 typ.	>1000 typ.
Capacitancia de entrada [pF]	1 máx.	2 typ.	3.2 typ.	2 typ.	2.5 typ.
Corriente en Estado de Inactividad [mA]	17 typ.	6 máx.	4.4 typ.	6 máx.	4.5 máx.

Tabla 3. Comparativa Amplificadores Operacionales

Obtenido de: [12, 13, 14, 15, 16].

El amplificador seleccionado fue ensayado con una señal cuadrada de 1 [MHz] y 3.3[V] con su propia PCB de la misma forma que el resto de los dispositivos, y dados los resultados satisfactorios, se incorporó al diseño final.

 <b>UTN Regional Paraná</b>	Analizador Lógico y de Protocolo
Ingeniería en Electrónica	Capítulo 2: Desarrollo

### 2.2.5.3 Cambiador de Nivel

A la hora de capturar los resultados de la conversión a través del puerto paralelo del ADC, es necesario bajar el nivel de tensión de salida de 5 [V] a 3.3 [V], para evitar exceder la máxima tensión de entrada del MCU. Para ello se utilizó el transceptor de bus octal SN74LVTH245A, por los siguientes motivos [17]:

- Excelente relación precio-calidad.
- Voltajes de entrada y salida compatibles con el resto de los componentes.
- Tiempo de conmutación de 4 [ns].
- Salida Tri-State, eliminando la necesidad de resistores de Pull-Down.

### 2.2.5.4 Referencia ADC

Uno de los requerimientos adicionales del ADC1175 es que el usuario provea los límites superiores e inferiores de comparación. En nuestro caso, el límite inferior siempre será GND, y el superior se definió en unos 2.5 [V].

Para lograr esta referencia de voltaje, se intentó utilizar:

- La resistencia interna que el ADC trae para este propósito.
- Divisores resistivos de varios valores, con y sin capacitores de filtro.
- Divisores resistivos con filtros LC.
- Amplificador Operacional LM358 como seguidor de tensión de un divisor resistivo.

Obteniéndose los mejores resultados con este último, aprovechando la Relación de Rechazo de Voltaje de Fuente, por lo que fue incorporado al diseño.

### 2.2.5.5 Esquema Canal Analógico

Con lo descrito en las secciones anteriores, se definió el esquema que muestra la Figura 12. El comportamiento del esquema completo se verificó satisfactoriamente y sus resultados son presentados en 3.1.2.

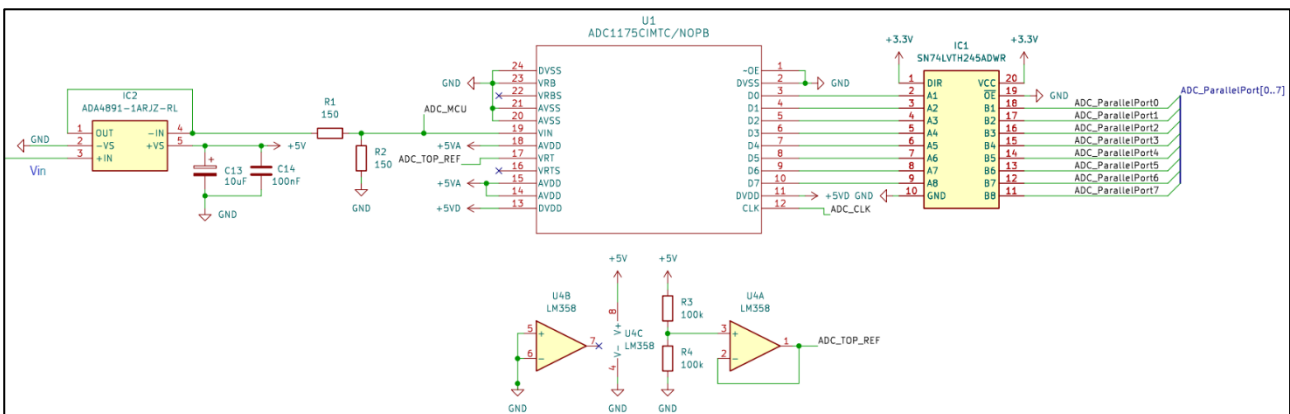



Figura 12. Canal Analógico

### 2.2.6 Microcontrolador

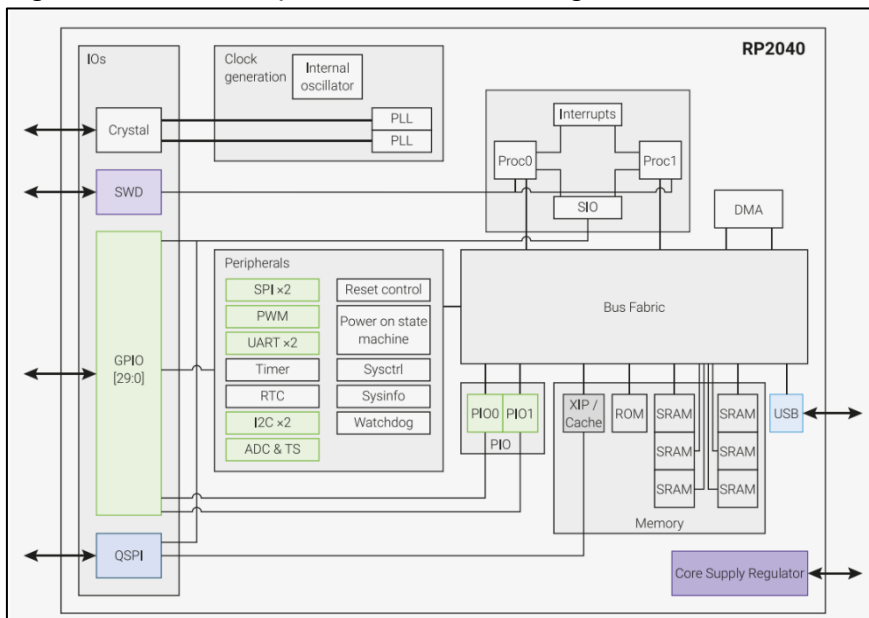
El MCU a utilizar es el RP2040 de Raspberry Pi. Este fue una elección clara (por motivos que serán aparentes a la brevedad) y, por ser fundamental para el proyecto, preferido desde

 <b>UTN Regional Paraná</b>	Analizador Lógico y de Protocolo
Ingeniería en Electrónica	Capítulo 2: Desarrollo

el punto de partida (es decir, el hardware del proyecto diseñó en torno a él). A saber, las características más importantes de este dispositivo de bajo costo son [18]:

- Dos núcleos C rtex M0+ hasta 133 [MHz].
- 264kB de SRAM en el sistema en 6 bancos.
- 30 GPIO.
- 6 IO dedicadas para Flash SPI.
- Hardware dedicado para los perif ricos m s utilizados.
- **Soporte para perif ricos extendido mediante PIO.**
- ADC de 4 canales con conversi n de 500ksps a 12 bit.
- USB 1.1 Host/Device.

Un diagrama general del MCU puede verse en la Figura 13:



*Figura 13. Diagrama de Bloques RP2040.  
Obtenido de [18].*

### 2.2.6.1 PIO

En nuestra implementaci n, es obligatorio tener la capacidad de recolectar datos en 8 canales a la velocidad m s alta posible, de forma simult nea, y con una m nima capacidad de filtrado para poder discernir eventos como flancos de subida y bajada en las se ales capturadas. Para ello, **utilizaremos los Programmable Input Output (PIO)** del microcontrolador, unas interfaces de hardware programables muy vers tiles, puesto que permiten ejecutar un lenguaje ensamblador propio, orientado la transmisi n y recepci n de datos. En la Figura 15 puede apreciarse la estructura interna de uno de ellos: Existen 2 perif ricos PIO dentro del RP2040, con 4 m quinas de estado cada uno, las cuales comparten memoria de programa. Cada una de estas m quinas de estado tiene:

- Mapeo flexible hacia los GPIO, permitiendo leer o escribir datos desde o hacia el exterior.

- Una interfaz DMA con la capacidad de mover hasta una palabra (32 bits) por ciclo de reloj desde o hacia el resto del sistema.
- Solicitud de Interrupción a los núcleos principales.
- Dos registros de desplazamiento de 32 bits bidireccionales (Out Shift e In Shift).
- Dos registros de 32 bits de propósito general (Scratch X e Y).
- 4 bus FIFO de 32 bits en ambas direcciones (TX y RX FIFO).
- Divisor de CLK fraccional (16 bits enteros, 8 fraccionales).

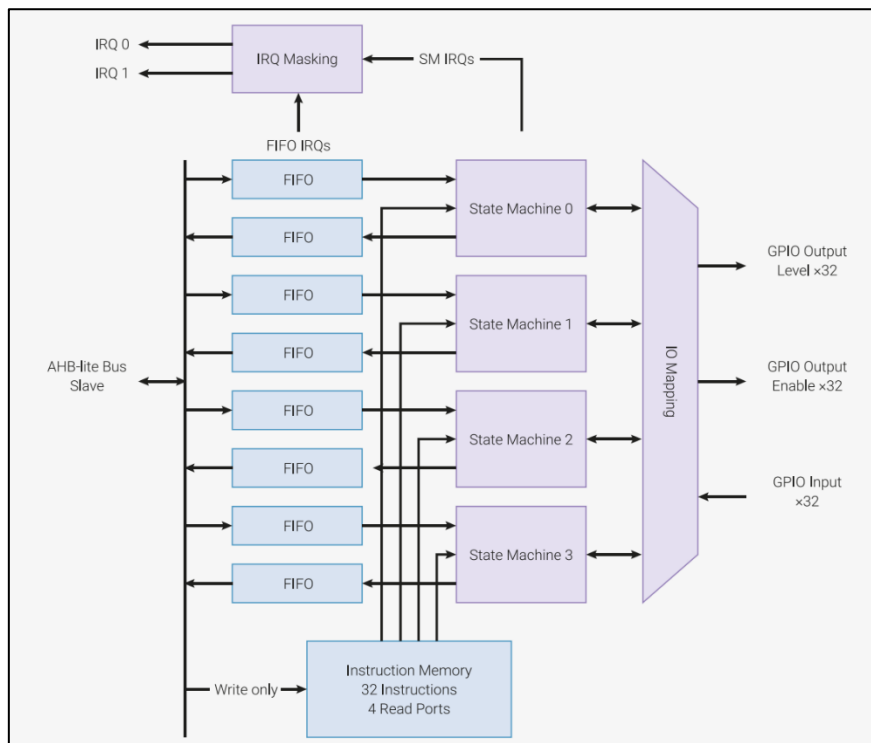


Figura 15. Periférico PIO.  
Obtenido de [18].

Cada una de las máquinas de estado tiene el modelo de programador que se muestra en la Figura 14:

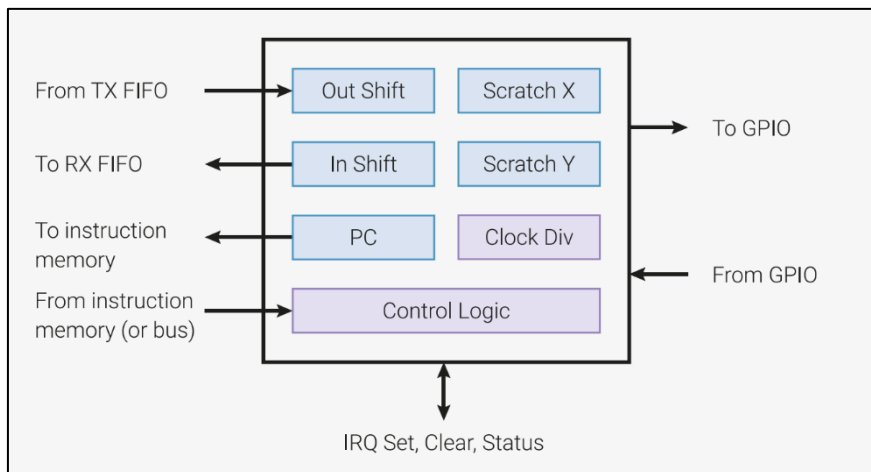



Figura 14. Máquina de estado PIO.  
Obtenido de [18].

 <b>UTN Regional Paraná</b>	Analizador Lógico y de Protocolo
Ingeniería en Electrónica	Capítulo 2: Desarrollo

Haciendo uso de estas máquinas de estado, es que se logran los distintos modos de captura de 2.3.2.1.

### 2.2.6.2 ADC On-Board

Como se comentó en 2.2.5.1, el ADC1175 tiene una frecuencia de muestreo mínima típica de 1 [MHz], por lo que se requiere hardware adicional para ofrecerle al usuario una frecuencia de muestro menor. Afortunadamente, el RP2040 ofrece un ADC On-Board de Aproximaciones Sucesivas de 500 [KHz]@12 bits (aunque truncaremos el resultado a 8 bits para mantener uniformidad). La Figura 16 ofrece un diagrama de bloques del dispositivo:

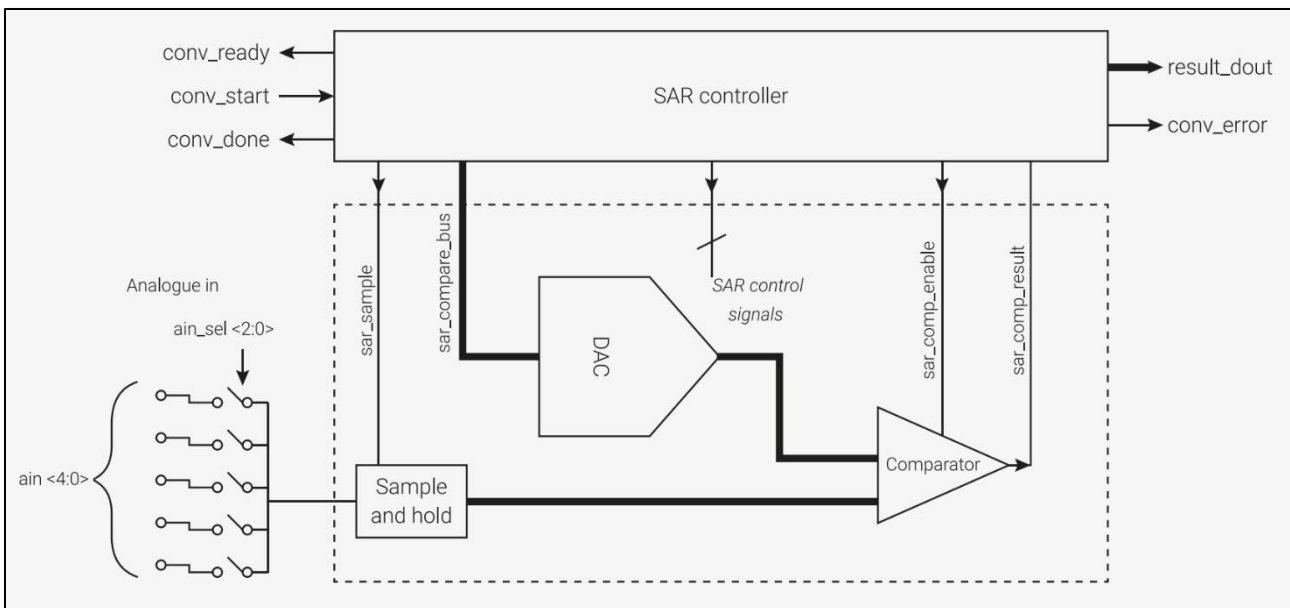



Figura 16. Diagrama de Bloques ADC RP2040  
Obtenido de [18].

### 2.2.6.3 DMA

Dados los grandes volúmenes de datos que recolectan en poco tiempo los periféricos de 2.2.6.1 y 2.2.6.2, surge la necesidad moverlos desde su origen a la memoria SRAM del sistema, para luego reportarlos a la aplicación en PC. Para lograr esto, utilizaremos el controlador DMA del sistema, que está diseñado para resolver esta misma necesidad, de forma independiente de los núcleos CórteX.

Este controlador es capaz de hacer operaciones de una lectura de 32 bits y una escritura de 32 bits por ciclo de reloj del sistema. Dichas operaciones pueden ser [18]:

- Periférico a memoria: un periférico indica al DMA que ha recibido datos, y DMA escribe los datos desde el FIFO del periférico a un arreglo en RAM. **Este es el caso de uso para el proyecto.**
- Memoria a periférico: un periférico indica que necesita información y DMA se la provee desde un arreglo de RAM o Flash, escribiendo al FIFO del periférico.
- Memoria a memoria: copia arreglos en memoria tan rápido como sea posible.

 <b>UTN Regional Paraná</b>	Analizador Lógico y de Protocolo
Ingeniería en Electrónica	Capítulo 2: Desarrollo

Finalmente, usando el diagrama de buses podemos resumir los roles de cada uno de los componentes del MCU, como se muestra en la Figura 17:

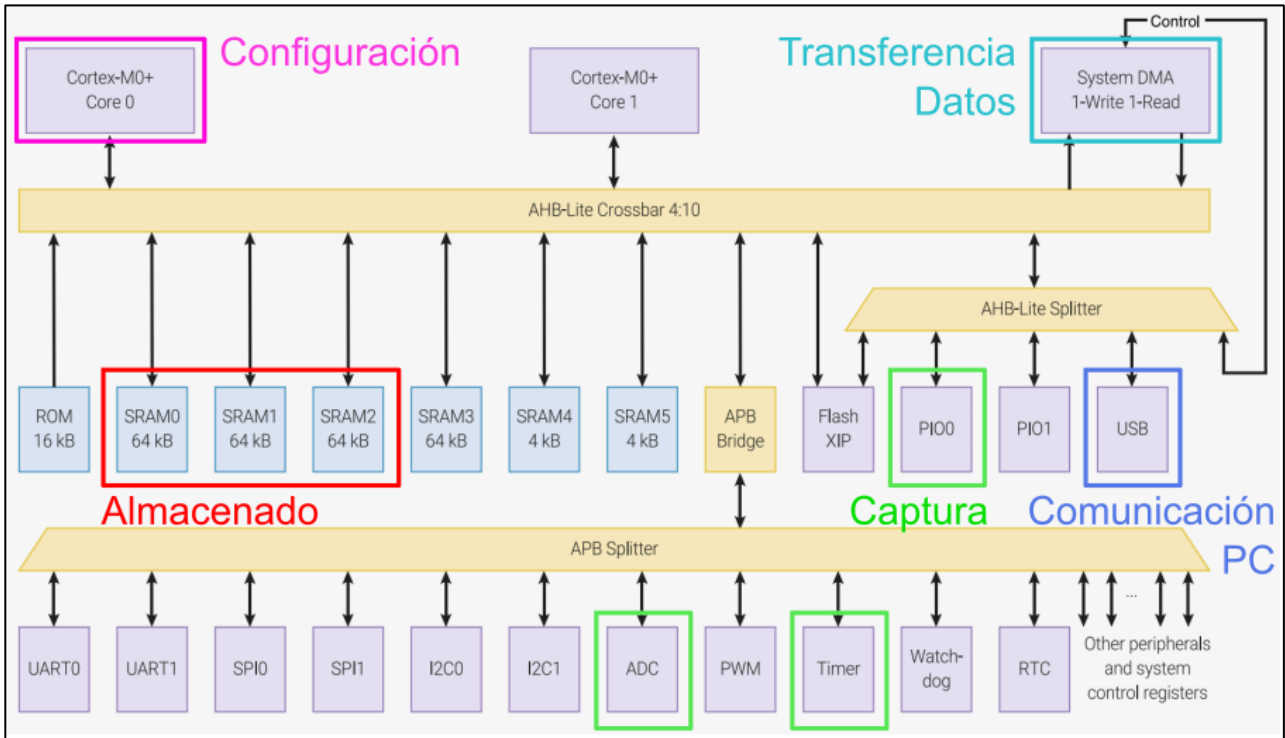


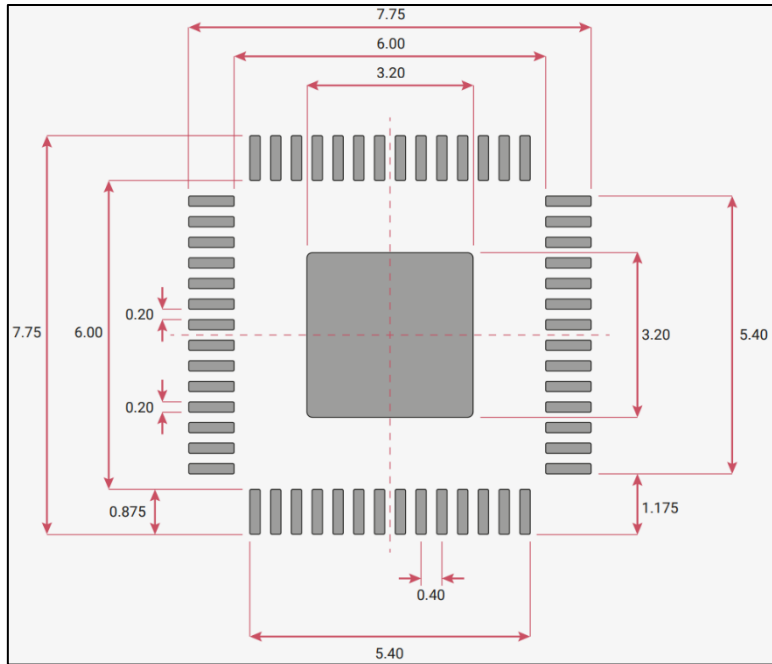
Figura 17. Diagrama de Conexión de Buses con Roles de Componentes. Obtenido de [18].

### 2.2.7 Raspberry Pi Pico

Lógicamente, el MCU seleccionado no puede funcionar por su cuenta. Para adaptarlo al proyecto, se requieren componentes adicionales como [18, 19]:

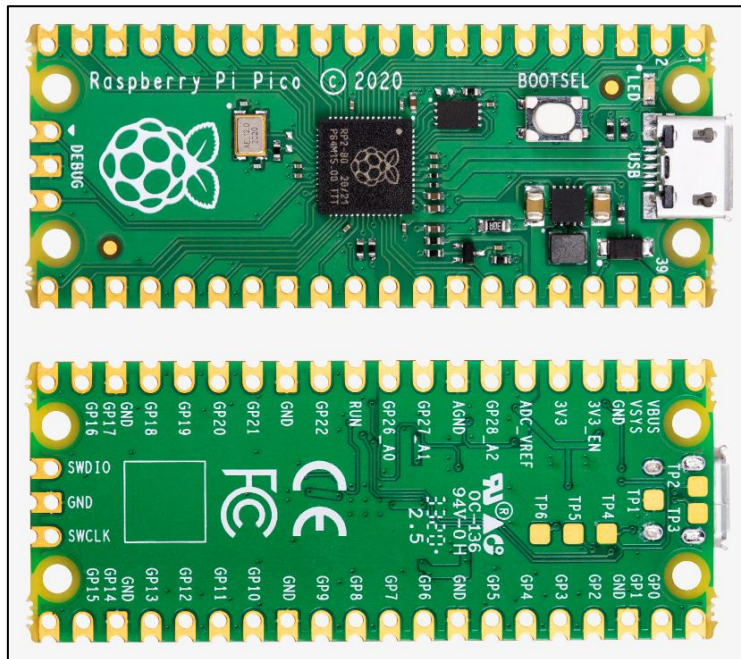
- Cristal de 12 [MHz] como referencia para el PLL del sistema.
- Flash QSPI para guardar el programa.
- Capacitores de filtro de alimentación, en encapsulados SMD 0201 y 0402.
- Un regulador DC-DC integrado con los circuitos que este requiera para su funcionamiento, para adaptar los 5 [V] del bus USB.
- Un botón para seleccionar habilitar el modo BOOTSEL, para actualizar el programa del MCU.
- Un conector USB Micro B para conectarlo a la PC.

Además, como se puede apreciar en la Figura 18 las tolerancias del footprint del encapsulado QFN56 del MCU son bastante superiores a las que los proveedores de PCB locales pueden lograr con seguridad. Aún más, no se disponen herramientas adecuadas para el proceso de soldadura (como, por ejemplo, estación de soldado u horno de soldadura) para lograr producto final de buena calidad.




*Figura 18. Footprint recomendado para el RP2040. Obtenido de [18], medidas en [mm]*

Por lo tanto, se decidió utilizar la placa de desarrollo Raspberry Pi Pico, que soluciona todas las necesidades antes mencionadas, a un precio razonable, que puede apreciarse en la Figura 19.



*Figura 19. Raspberry Pi Pico Rev3. Obtenido de [19].*



 <b>UTN Regional Paraná</b>	Analizador Lógico y de Protocolo
Ingeniería en Electrónica	Capítulo 2: Desarrollo

### 2.2.8 Botón de Reset

Como será aparente en 2.3, es necesario proveerle al usuario una forma de interrumpir un muestreo sin desconectar el analizador del PC, por lo que se hizo uso del pin de reset del MCU, activo por nivel bajo, con un circuito antirrebote, como lo muestra la Figura 20:

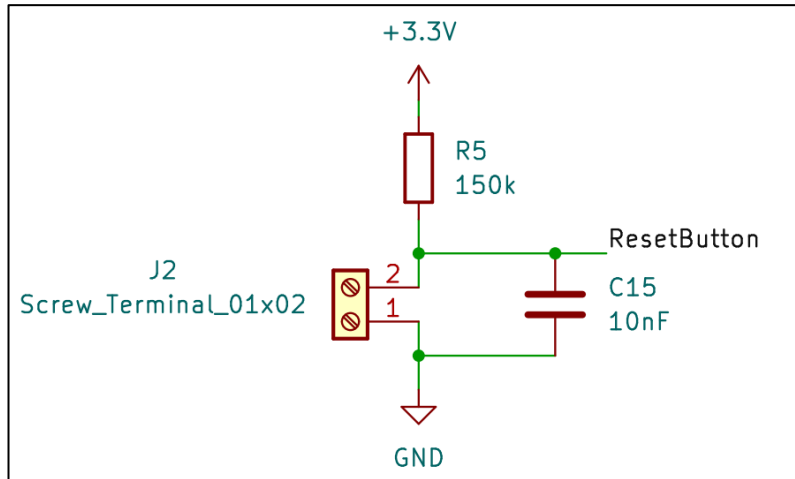


Figura 20. Antirrebote Reset.  
Obtenido de [20].

El interruptor se conecta en la bornera J2, para que su montaje no sea directamente soldado sobre la PCB, sino, más bien, sobre el gabinete. La constante de tiempo del circuito es de 1.5 [ms], un poco menor que lo generalmente recomendado [20], pero aceptable según las pruebas realizadas y conveniente por ser los valores de resistor y capacitor disponibles.

### 2.2.9 Alimentación

Para suplir las necesidades de alimentación de todos los circuitos en la PCB se confeccionó el circuito que muestra la Figura 21.

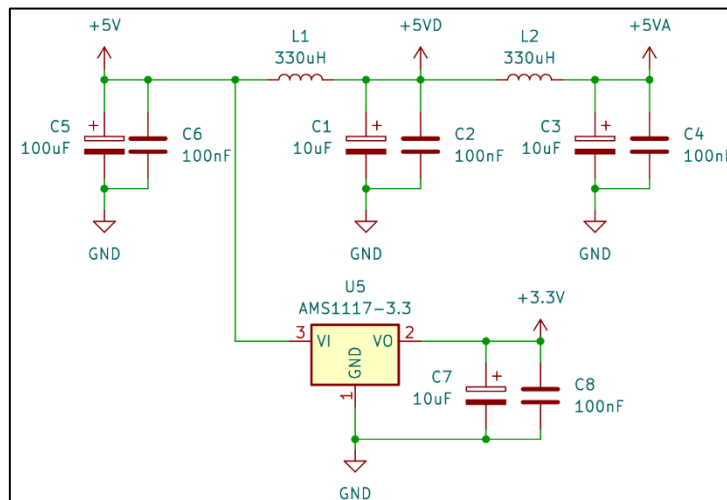



Figura 21. Alimentación PCB.

 <b>UTN Regional Paraná</b>	Analizador Lógico y de Protocolo
Ingeniería en Electrónica	Capítulo 2: Desarrollo

Algunos remarcos interesantes son:

- El riel +5[V] proviene del bus de alimentación de la interfaz USB, y pasa por un fusible de 250 [mA], puesto que el estándar USB 2.0 demanda que la Corriente Máxima Continua que un Root Port Hub (léase, un puerto USB de una PC) provea sea 500 [mA] (o “5 unidades de carga”) [21].
- +5VD y +5VA son las alimentaciones del ADC1175, filtradas para garantizar el funcionamiento libre de ruido de las partes digital y analógica del integrado, respectivamente, como lo sugiere su datasheet [10].
- Se eligió incorporar el regulador lineal AMS1117 para no exigir el SMPS incluido en la Raspberry Pi Pico (que entrega la misma tensión), al alimentar los integrados SN74LVTH245A y MCP4821.
- Se añadieron capacitores de filtrado para todos los componentes que no tuvieran uno en su inmediata vecindad, a excepción de los comparadores analógicos MAX9108 ante una aparente falta de espacio en la PCB (en retrospectiva, conociendo mejor las tolerancias del fabricante, se podrían haber ubicado de todos modos).

La alimentación funcionó satisfactoriamente con la excepción de cierta variación en la tensión provista por el bus de USB, que se registró variar entre 4.85 y 5.1 [V]. Luego de una revisión más detallada del estándar USB, se observó que tensiones entre 4.75 y 5.25 [V] están permitidas, como mínimo [21]. Esto es una **omisión grave** en la etapa de investigación y debe ser obligatoriamente corregido en próximas revisiones de hardware.

### 2.2.10 PCB

El diseño de la PCB se realizó en la suite de diseño electrónico KiCad 7.0 utilizando en su mayoría esquemáticos y footprints provistos por los fabricantes de los circuitos integrados o bien ya existentes dentro del software. En particular, se siguieron lo máximo posible los siguientes lineamientos para el PCB de dos capas:

- Ancho de pista mínimo de 0.25 [mm].
- Separación de pista mínimo de 0.25 [mm], aumentando donde fuera posible.
- Evitar transferir señales digitales de alta velocidad de una capa a otra siempre que sea posible.
- Mantener largo de pistas al mínimo posible.
- Mantener una distancia ligeramente mayor a la mínima posible para facilitar el proceso de soldadura.

A partir de esto, se obtuvo un diseño como el que se muestra en la Figura 22.

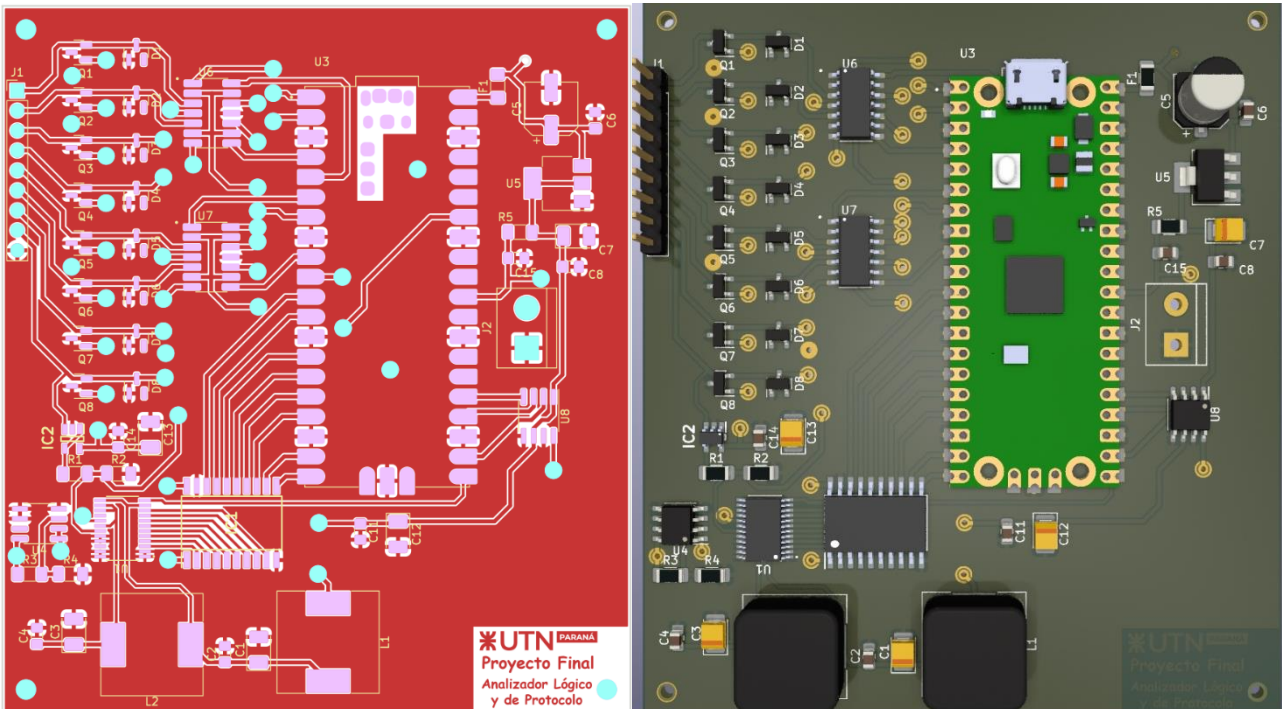


Figura 22. Capa superior diseño PCB y Vista superior modelo 3D PCB

La fabricación del PCB final, contrario al resto de PCBs del proyecto, fue tercerizado a un proveedor local por la mejor calidad y tolerancias más estrictas provistas. Lamentablemente, dicho proveedor no ofrece un proceso de PTH para las vías, por lo que tuvieron que improvisarse con segmentos de cable soldados a ambos lados de la PCB. Posteriormente, el proceso de soldadura se completó con estaño en pasta de buena calidad (fusión a 135 [C]), estaño 63/37 en hilo de 0.3 [mm] y un soldador tipo lápiz de 40 [W], por ser lo disponible en el momento. Los resultados se enseñan en la Figura 23.

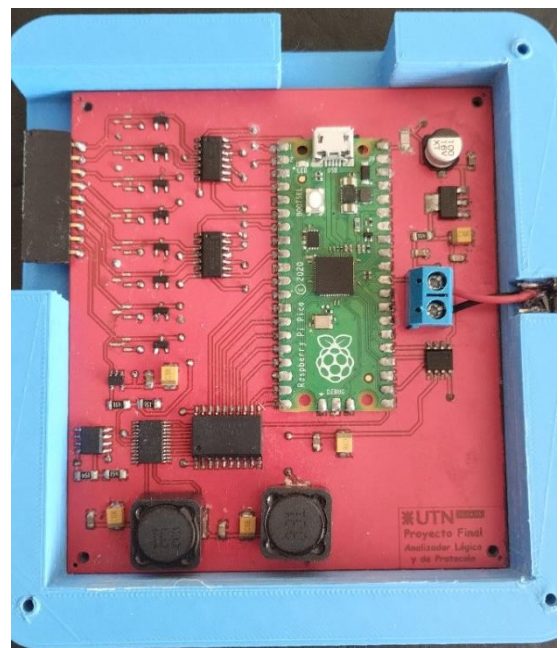

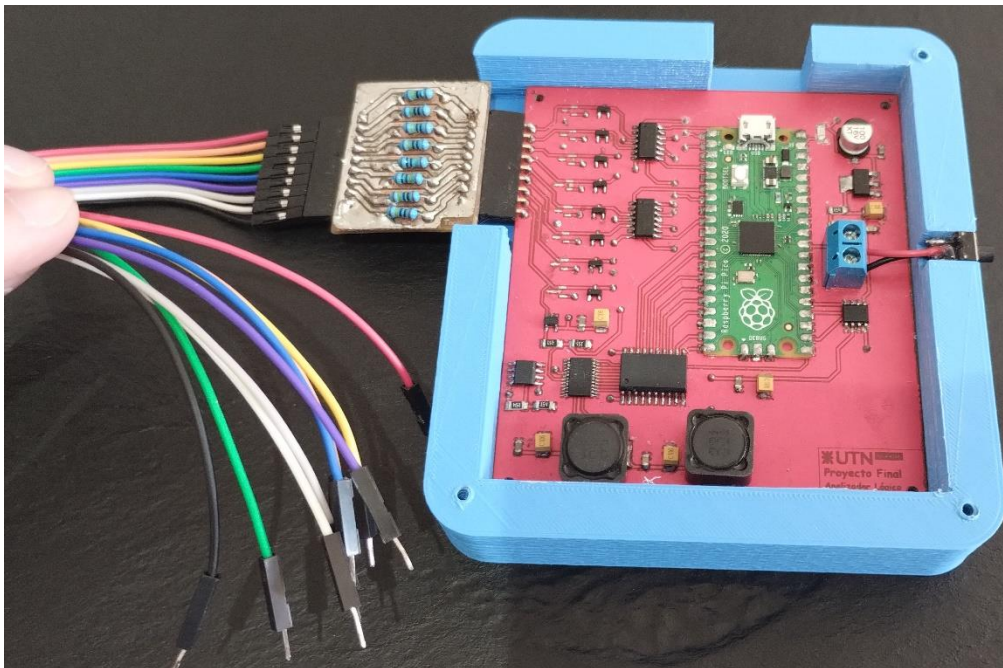


Figura 23. PCB con sus componentes soldados


 <b>UTN Regional Paraná</b>	Analizador Lógico y de Protocolo
Ingeniería en Electrónica	Capítulo 2: Desarrollo

### 2.2.11 Resistores Pull-Down

Como resultado de las altas impedancias de todos los componentes en la etapa de entrada, se obtuvo una impedancia de entrada muy elevada en el dispositivo, medida en unos 5 [MΩ] en DC con un multímetro. Por lo tanto, cualquier ruido electromagnético lindante que se induzca en el circuito tiene la capacidad de crear una tensión lo suficientemente alta para ser interpretadas por los comparadores como un nivel alto. En principio, se presumió que la corriente de pérdida de los Schottky Clamp (aproximadamente un 1[μA], variable con la temperatura [6]) sería suficiente pull-down, pero este claramente no fue el caso. Para solventar este problema, se decidió incorporar una PCB adicional con resistores de pull-down de 1 [MΩ] por canal, como se muestra en la Figura 24, con resultados satisfactorios.



*Figura 24. PCB Pull-down adicional*

 <b>UTN Regional Paraná</b>	Analizador Lógico y de Protocolo
Ingeniería en Electrónica	Capítulo 2: Desarrollo

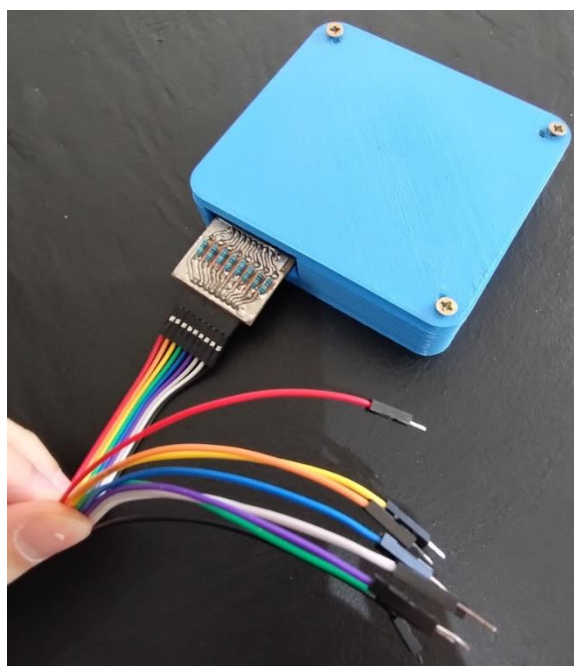
### 2.2.12 Gabinete

Finalmente, para darle una presentación más satisfactoria y profesional al dispositivo, se diseñó en FreeCAD e hizo imprimir en 3D un gabinete, como se muestra en Figura 25.




*Figura 26. Parte Superior Gabinete*

El principal defecto de este diseño es la incorrecta posición de los agujeros pasantes de la parte superior, que no coincide con los agujeros de la parte inferior y deberá ser rectificada en una futura revisión. Además, por inconvenientes técnicos del proveedor, las letras de la tapa no fueron impresas, quitándole algo de visibilidad. El ensamblado final se puede apreciar en la Figura 27.



*Figura 27. Ensamblado Final*

 <b>UTN Regional Paraná</b>	Analizador Lógico y de Protocolo
Ingeniería en Electrónica	Capítulo 2: Desarrollo

## 2.3 Firmware

### 2.3.1 Requerimientos

En función de lo expresado en 1.5, el firmware del MCU debe poder.

- Interactuar con una aplicación GUI a través del puerto USB.
- Configurar el DAC para obtener la tensión deseada.
- Muestrear los 8 canales digitales simultáneamente, pudiendo variar:
  - La cantidad de canales muestreados en 1, 2, 4 y 8.
  - La frecuencia de muestreo.
  - La profundidad de muestreo.
  - Esperar a un evento particular para comenzar a muestrear y poder definir que canal espera dicho evento.
- Manipular el ADC1175 para el muestreo analógico en el canal 8, y usar el ADC On-Board cuando la frecuencia lo demandara, además de poder variar los mismos parámetros del punto anterior.

### 2.3.2 Descripción General

Veamos un diagrama de flujo general del firmware del MCU en la Figura 28. En términos generales, el firmware consiste en un “Mega-Loop” que solo es reiniciado por una desenergización del sistema, que puede ser la desconexión del cable USB o presionar el botón de reset, descrito en 2.2.8.

Al comenzar el programa, este utiliza varias funciones del SDK del fabricante para poner en marcha todas las partes del MCU necesarias para el funcionamiento como los periféricos USB y SPI, además de inicializar variables y estructuras de control. Una vez listo, se hace polling al periférico USB hasta que se reciba el carácter de fin de mensaje o se llene el búfer de entrada (que no debería suceder). Entonces, se verifica la sintaxis del mensaje y de ser correcta, se corre la función “Ejecutar Orden”. De otro modo, se corre la función “Mensaje de error” que simplemente define un mensaje de error y un identificador que indica ese tipo de error. Finalmente, se escribe al puerto/periférico USB lo indicado por la función que se haya corrido inmediatamente antes en formato binario.

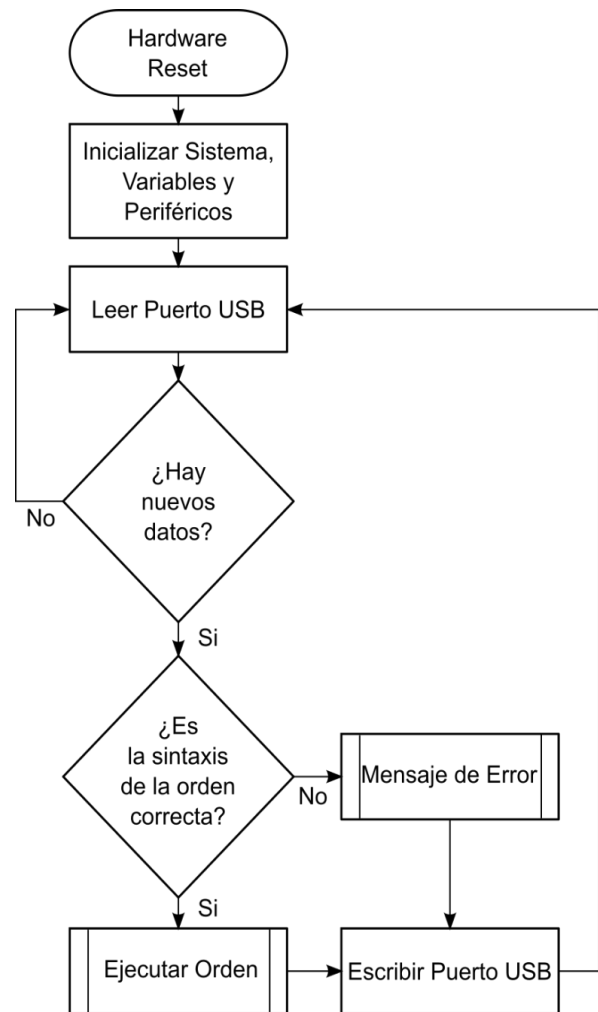


Figura 28. Diagrama de Flujo General

### 2.3.2.1 Función “Ejecutar Orden”

Con el diagrama de bloques de la Figura 29, la función “Ejecutar Orden” resume un montón de funcionalidades del dispositivo, en las que se ejecuta la directiva enviada por el puerto USB. La orden, orientada a implementar los puntos de 2.3.1, puede ser:

- Configurar el tiempo de espera de gatillado. Los tiempos de espera de gatillado son controlados por uno de los clock que genera una interrupción (o “alarma”, como los define el SDK) y en este caso se cambia el valor de la variable que guarda dicho tiempo de espera.
- Cambiar la tensión de los comparadores. La tensión del DAC que utilizan los comparadores para es variable a partir del uso de la interfaz SPI0 del MCU con una simple llamada a función.

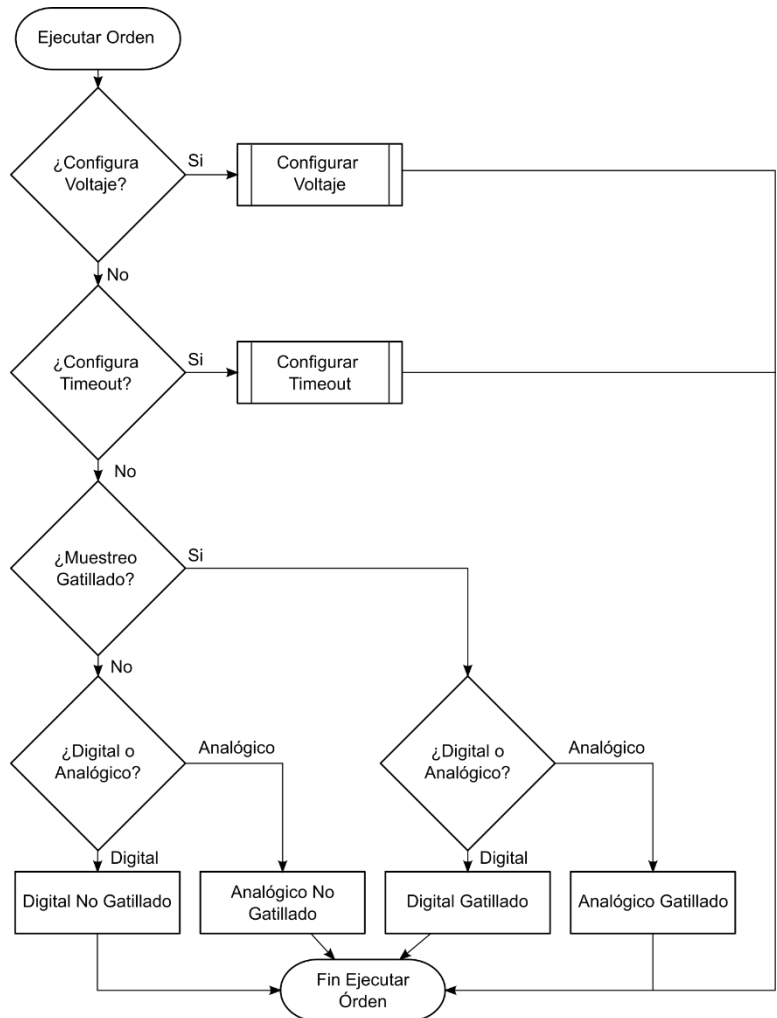



Figura 29. Diagrama de Flujo “Ejecutar Orden”

- Muestrear las señales en las entradas con una configuración particular, representados como los bloques “Digital No Gatillado”, “Digital Gatillado”, “Analógico No Gatillado” y “Analógico Gatillado” al final. Para esto el programa define si se usa un PIO:
  - El programa que cargar, definiendo su funcionamiento.
  - Escribir registros de configuración según corresponda.
  - Definir divisor de clock y realizar un underclock del sistema si hiciera falta.
  - Configurar la interrupción que implementa el tiempo de espera.
  - Configurar el controlador DMA según corresponda.

Si se utiliza el ADC On-Board, realiza algo similar, solo que no carga el programa. Luego de terminar el muestreo, también reinicia los periféricos usados para poder reconfigurarlos fácilmente.

 <b>UTN Regional Paraná</b>	Analizador Lógico y de Protocolo
Ingeniería en Electrónica	Capítulo 2: Desarrollo

### 2.3.3 Protocolo de comunicación

El protocolo de comunicación diseñado es muy similar a una arquitectura cliente-servidor, donde la GUI, en su papel de cliente, demanda información al RP2040, que tiene el papel de servidor. Para solicitar información al MCU, se definieron una serie de directivas que siempre comienzan con el símbolo “#” y terminan con “;”, que se enseñan a continuación:

#### 2.3.3.1 Muestreo Digital

#ABCDEFGH I;

- A: “D”, Muestreo Digital
- B: 0-4, Modo de Gatillado, Tabla 4
- C: Canal de gatillado 1-8
- D: Numero de canales: 8, 4, 2, 1
- EF: 00-16, Frecuencia de Muestreo, Tabla 5
- GHI: 000-192, Profundidad de Muestreo, en [KiB]

Modo de Gatillado	
0	No Gatillado
1	Flanco de Subida
2	Flanco de Bajada
3	Nivel Bajo
4	Nivel Alto

Tabla 4. Codificación Modos de Gatillado

#### 2.3.3.2 Muestreo Analógico

#ABCDEFGH I;

- A: “A”, Muestreo Analógico
- B: 0-4, Modo de Gatillado, Tabla 4
- C: Canal de gatillado 8
- D: Numero de canales: 1
- EF: 00-13, Frecuencia de Muestreo, Tabla 5
- GHI: 000-192, Profundidad de Muestreo, en [KiB]

Frecuencia de Muestro	
00	1 [KHz]
01	2.5 [KHz]
02	5 [KHz]
03	10 [KHz]
04	25 [KHz]
05	50 [KHz]
06	100 [KHz]
07	250 [KHz]
08	500 [KHz]
09	1 [MHz]
10	2 [MHz]
11	5 [MHz]
12	10 [MHz]
13	25 [MHz]
14	50 [MHz]
15	100 [MHz]
16	125 [MHz]

Tabla 5. Codificación Frecuencias de Muestreo

#### 2.3.3.3 Voltaje Referencia

#ABCDE;

- A: “C”, Configurar
- B: “V”, Voltaje DAC
- CDE: Voltaje, Formato "C.DE" [V]

#### 2.3.3.4 Tiempo de Espera

#ABCDEF;


- A: “C”, Configurar
- B: “T”, Tiempo
- CDEF: Tiempo, Formato "CDEF"x10 [ms]

#### 2.3.3.5 Respuestas y Códigos de Error

Existen varias posibles respuestas a los comandos descriptos anteriormente, que se detallan a continuación:

- “#ACK;” indica que el comando se ha aceptado y ha ejecutado fehacientemente.
- “#NACK;” indica un comando de sintaxis incorrecta
- “#END;” indica el fin de un bloque de datos muestreados.



 <b>UTN Regional Paraná</b>	Analizador Lógico y de Protocolo
Ingeniería en Electrónica	Capítulo 2: Desarrollo

- “#BADCOMMAND;” indica un comando de sintaxis correcta, pero con capacidades no soportadas. Ver Tabla 6.
- “#TRIGTIMEOUT;” indica que el tiempo de espera se ha excedido.

En la Figura 30 se pueden ver ejemplos de distintos casos del protocolo de comunicación. El texto en verde representa la parte variable de comandos de configuración y el texto en rojo, la parte incorrecta de un comando.

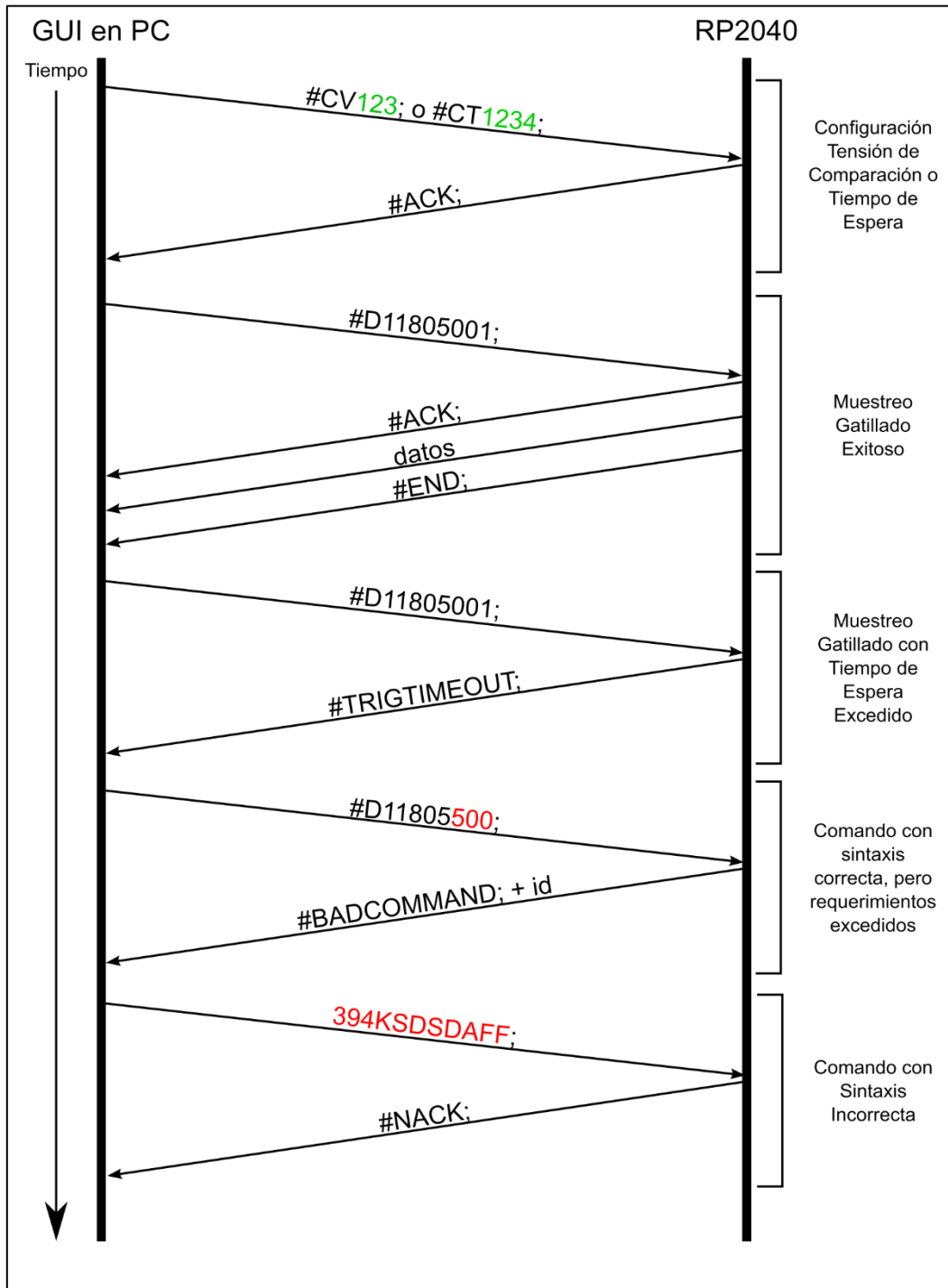



Figura 30. Protocolo de Comunicación

 <b>UTN Regional Paraná</b>	Analizador Lógico y de Protocolo
Ingeniería en Electrónica	Capítulo 2: Desarrollo


### 2.3.3.6 Tabla de Identificadores de Error

Identificadores de Error		
Id.	Relativo a	Significado
1	Funcionamiento General	En un comando de muestreo, caracteres que deberían ser numéricos no lo son.
2		En un comando de configuración de tensión de referencia, caracteres que deberían ser numéricos no lo son.
3		En un comando de configuración de tiempo de espera, caracteres que deberían ser numéricos no lo son.
4		La segunda letra al inicio del comando no corresponde a ninguna directiva válida.
5		La primera letra al inicio del comando no corresponde a ninguna directiva válida.
6	Muestreo Digital	Modo de gatillado inválido.
7		Canal de gatillado inválido.
8		Número de canales muestreados inválido.
9		Canal de gatillado no es un canal muestreado.
10		Frecuencia de muestreo inválida.
11		Profundidad de muestreo inválida.
12	Muestreo Analógico	Modo de gatillado inválido.
13		Canal de gatillado inválido.
14		Número de canales muestreados inválido.
15		Frecuencia de muestreo inválida.
16		Profundidad de muestreo inválida.
17	Configurar Voltaje	Voltaje de comparación inválido.
18	Configurar Timeout	Tiempo de espera inválido.

Tabla 6. Identificadores de Error

## 2.4 Software GUI

Para poder presentar una interfaz de usuario al proyecto se implementó una aplicación de escritorio en PySide6 (Qt for Python), utilizando Matplotlib como framework gráfico y Numpy para el tratado de los datos. Una captura de pantalla de la GUI antes de comenzar a muestrear se enseña en la Figura 31, con sus componentes numerados.

 <b>UTN Regional Paraná</b>	Analizador Lógico y de Protocolo
Ingeniería en Electrónica	Capítulo 2: Desarrollo

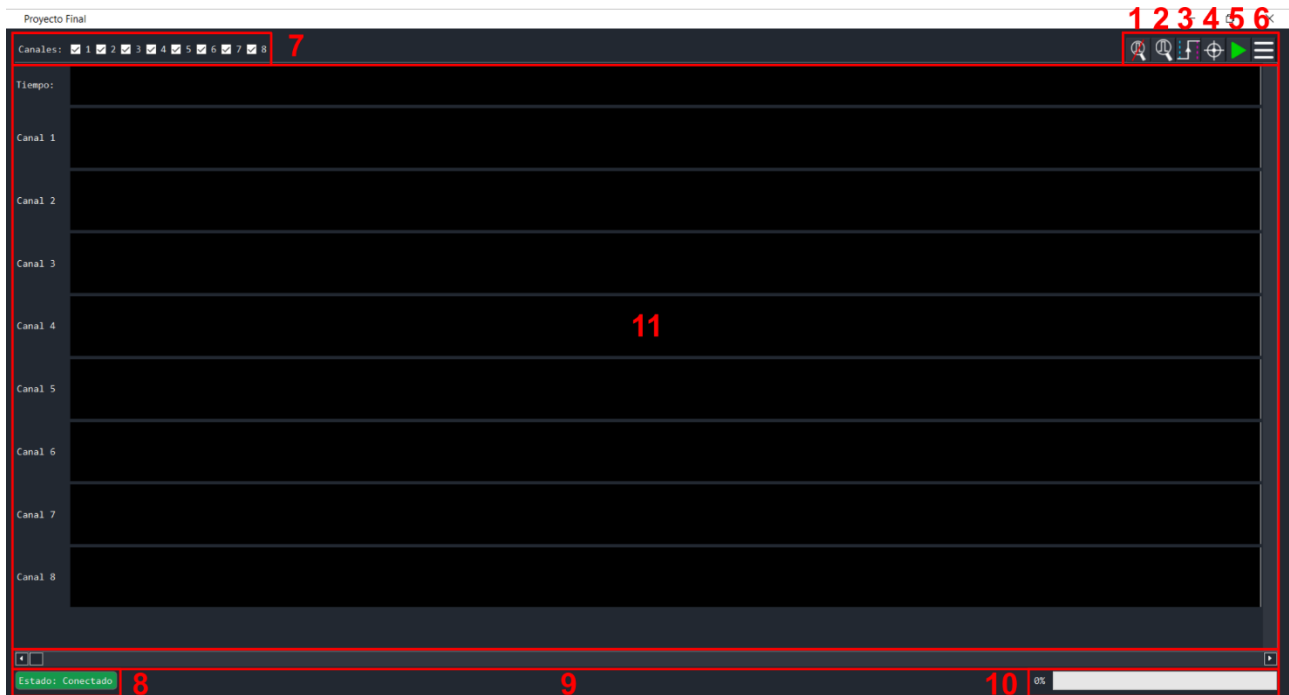



Figura 31. GUI antes del muestreo

Veamos la funcionalidad de cada una de sus partes:

1. “Borrar Análisis de Protocolo”: elimina cualquier Análisis de Protocolos.
2. “Análisis de Protocolo”: Inicia la GUI de Análisis de Protocolos, descrita en 2.4.7.
3. “Mediciones Fijas”: Inicia o detiene la funcionalidad de cursores fijos, descrita en 2.4.3.
4. “Cursor”: Apaga o enciende el cursor, descrito en 2.4.2.
5. “Muestrear”: Emite una directiva de muestreo al Analizador Lógico.
6. “Panel Lateral”: Muestra u oculta el panel lateral, enseñado en la Figura 32.
7. “Visibilidad de canales”: Muestra u oculta canales individuales.
8. “Etiqueta de estado”: Informa del estado de conexión/muestreo del Analizador.
9. “Barra de navegación”: Permite desplazarse en el buffer capturado.
10. “Barra de progreso de muestreo”: Reporta en qué instancia del proceso de muestreo se encuentra el sistema.
11. “Visualización”: 8 canales digitales o uno analógico disponen su información aquí.

Adicionalmente, podemos ver las tres páginas del panel lateral en la Figura 32.

- En la página 1 “Configuración”, los componentes numerados 1 a 4 y 6 a 8 implementan los dos últimos puntos de 2.3.1, permitiendo controlar todos los parámetros del Analizador Lógico al usuario y el componente 5 simplemente calcula el tiempo de muestreo a modo de guía.
- En la página 2 “Análisis”, se encuentra el componente 9, que permite buscar eventos como flancos o niveles lógicos dentro de un canal particular. En esta página también aparecen las mediciones fijas, descritos en 2.4.3.

 <b>UTN Regional Paraná</b>	Analizador Lógico y de Protocolo
Ingeniería en Electrónica	Capítulo 2: Desarrollo

- En la página 3 “Protocolo”, se permite al usuario cambiar la visualización de los mensajes decodificados, en el componente 10 y en el 11, se provee una tabla a modo de navegación de todos los mensajes decodificados.

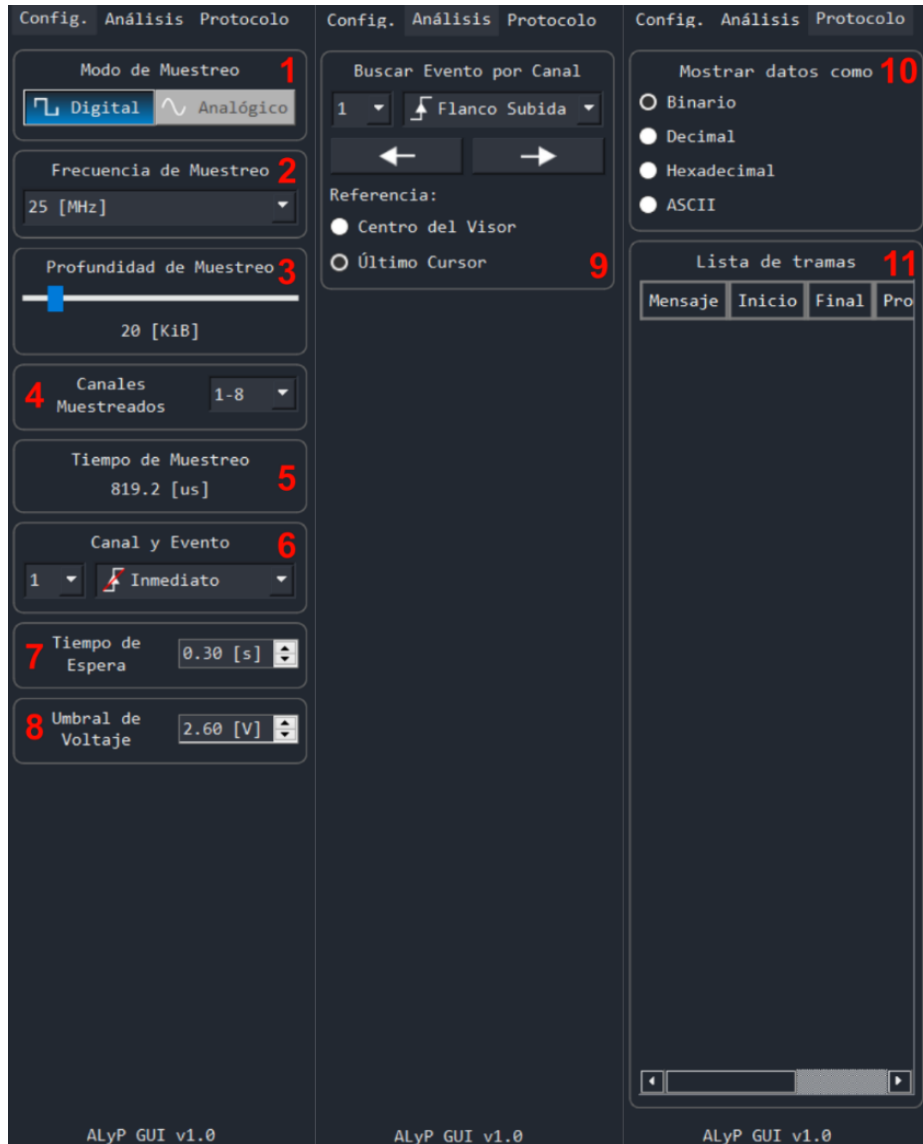



Figura 32. Panel Lateral GUI

 <b>UTN Regional Paraná</b>	Analizador Lógico y de Protocolo
Ingeniería en Electrónica	Capítulo 2: Desarrollo

### 2.4.1 Diagramas de Flujo

Para explicar el funcionamiento del software en una forma más amena, este se resumió en la Figura 33. En términos generales la GUI es un “Mega-Loop”, constantemente esperando a las señales emitidas por las interacciones del usuario con la aplicación. Cada una de estas interacciones o señales llaman a una función, similar a una Rutina de Servicio de Interrupción que terminan implementando las funcionalidades de la aplicación. En cuanto a estas rutinas, hay aproximadamente de 30 de ellas, y explicarlas todas es tedioso e innecesario, por lo que se proveen solo dos de ellas a modo de ejemplo, en la Figura 34.

A la izquierda de ella, se aprecia la función detrás de la detección del analizador en mediante la API WMI de Windows, que realiza la siguiente solicitud SQL: “SELECT DeviceID FROM Win32\_USBHub”, para obtener los dispositivos conectados por USB, y en función de si se encuentra el analizador conectado o no, refleja esta información en la etiqueta de estado (ver punto de 8 de Figura 31). Esta función es llamada cada 500 [ms] por un temporizador, logrando un balance entre velocidad de respuesta y carga de procesamiento.

Por otro lado, a la derecha, se detalla la función de muestreo, que utiliza multithreading para no interrumpir la ejecución del Mega-Loop principal. En simples términos, envía un comando de configuración de voltaje, de tiempo de espera y finalmente, de muestreo, esperando las respuestas correspondientes y gestionando cualquier error en la comunicación que pudiera ocurrir. Si todo resultó bien, se grafican las muestras adquiridas por el Analizador Lógico.

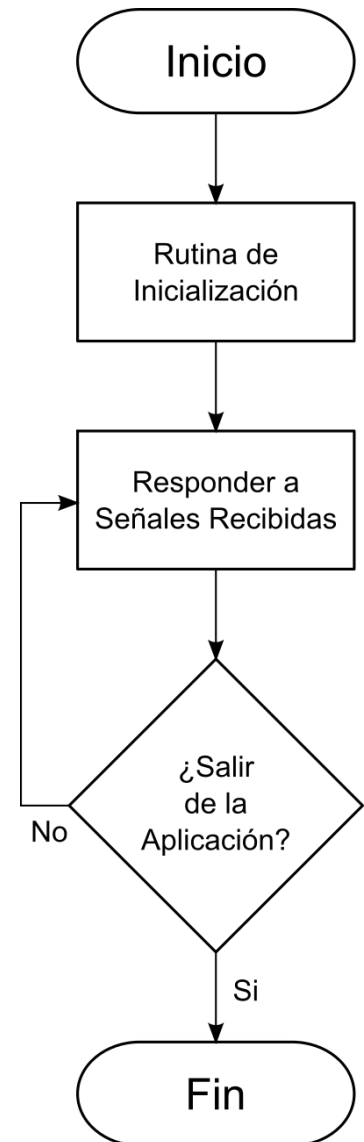
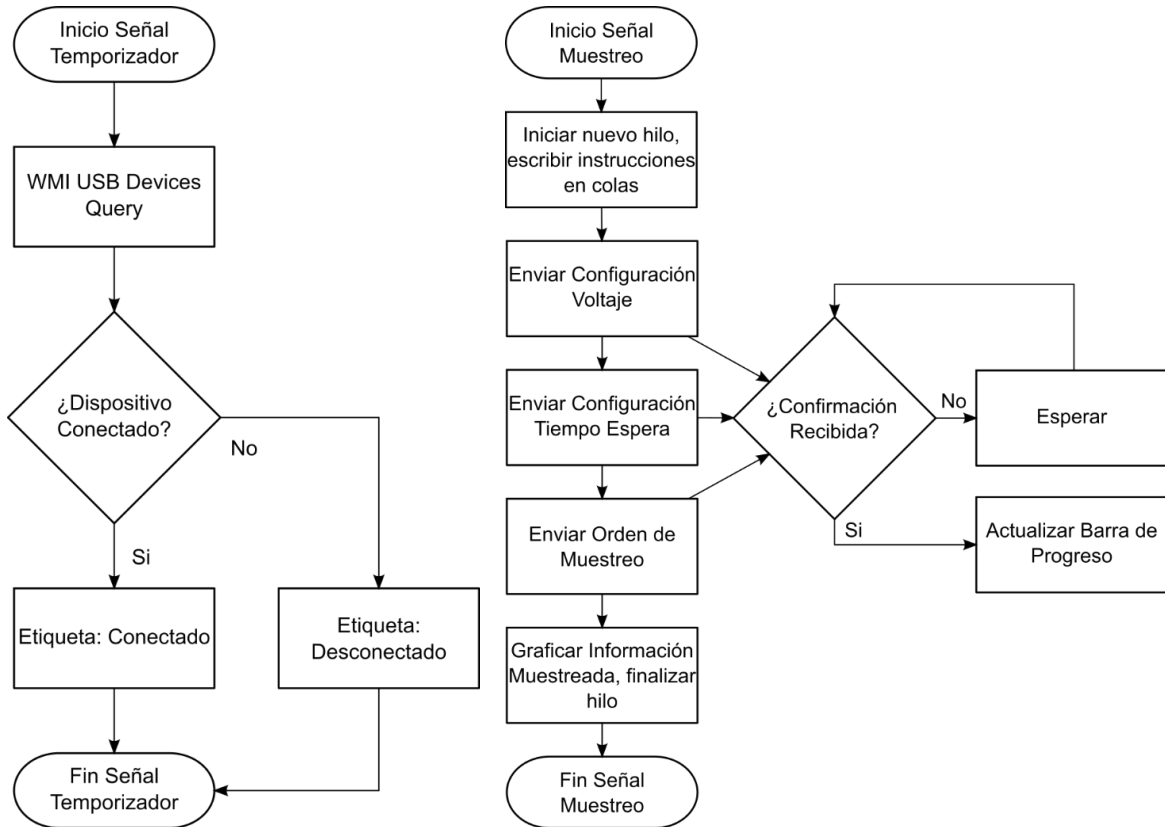



Figura 33. Diagrama de Flujo General GUI



*Figura 34. Diagramas de Flujo ISR GUI*

 <b>UTN Regional Paraná</b>	Analizador Lógico y de Protocolo
Ingeniería en Electrónica	Capítulo 2: Desarrollo

### 2.4.2 Cursor

Para proveer al usuario con una forma rápida de comparar las señales de distintos canales, se añadió la funcionalidad de cursor, visible en la Figura 35.



*Figura 35. Contador de Rizado con Cursor Activo*

Si se analiza en detalle la parte inferior de la Figura 35 se puede ver una etiqueta mostrando el tiempo transcurrido desde el inicio de la captura hasta la posición actual del cursor, y mediciones que informan el tiempo transcurrido entre los flancos inmediatamente a la izquierda e inmediatamente a la derecha, y la inversa de ese tiempo en formato como “frecuencia”. En el caso de ejemplo, podemos ver que el contador de rizado implementado tiene una frecuencia de 250 [KHz] en su señal más veloz (código disponible en 7.2)

### 2.4.3 Mediciones Fijas

Luego de algunas pruebas por parte de otros estudiantes de la carrera, se decidió que era necesario implementar alguna forma de recabar información entre dos instantes distintos de la captura. Para lograrlo, se añadió la posibilidad de realizar mediciones fijas con dos cursores adicionales, que el usuario puede añadir haciendo clic derecho en la muestra de interés. Se repiten las mismas mediciones que en el caso del cursor, además de la cantidad de muestras entre los dos puntos. Lo expuesto se puede apreciar en la Figura 36. Esta funcionalidad también está disponible en el canal analógico, donde se muestra también la diferencia de tensión en ambos puntos.

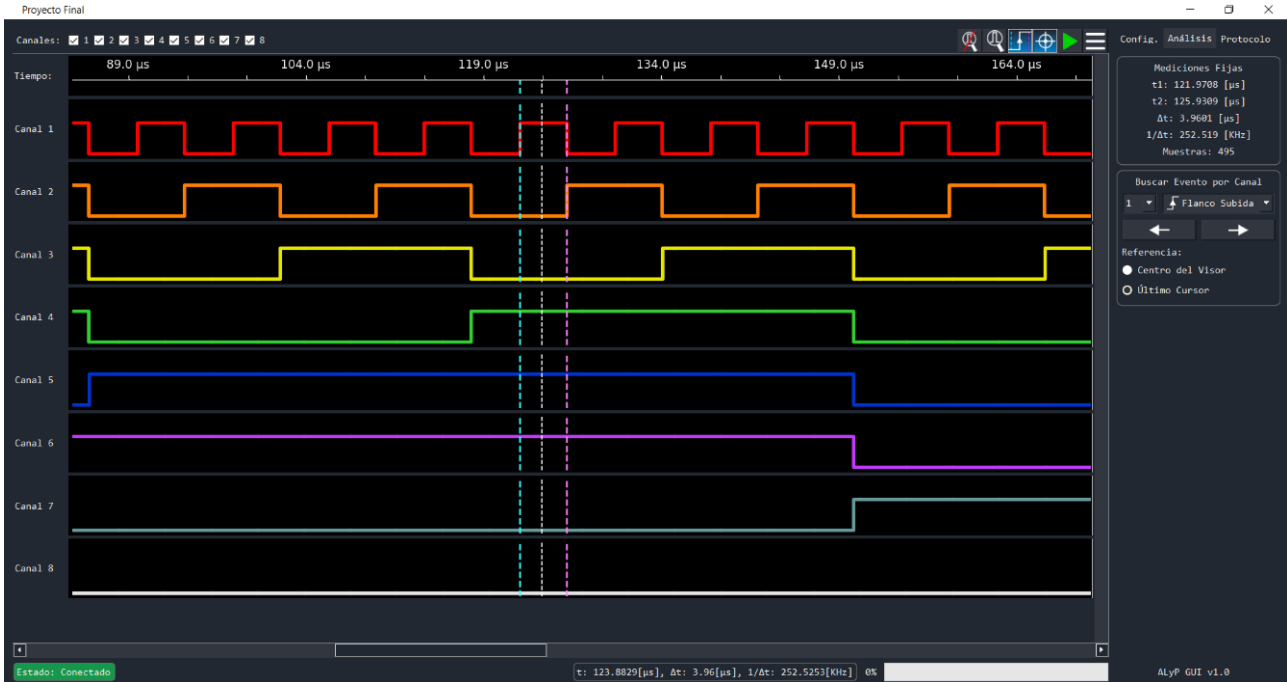


Figura 36. Contador de Rizado con Mediciones Fijas

### 2.4.4 Buscador de Eventos

Otra funcionalidad de la GUI es el buscador de eventos, que permite buscar flancos de subida y bajada y niveles altos o bajos dentro de un canal. Para utilizarlo, el usuario indica el evento y canal de interés en la página 2 del panel lateral y puede utilizar como referencia (desde donde se buscará) la última posición del cursor o el centro de la pantalla.

### 2.4.5 Canal Analógico

Las muestras obtenidas desde el ADC1175 o del ADC del MCU se presentan de una forma similar a la digital, como se muestra en la Figura 37. Todas las funcionalidades de cursor y mediciones fijas se mantienen en este modo de muestreo, también apreciables en la figura antes mencionada



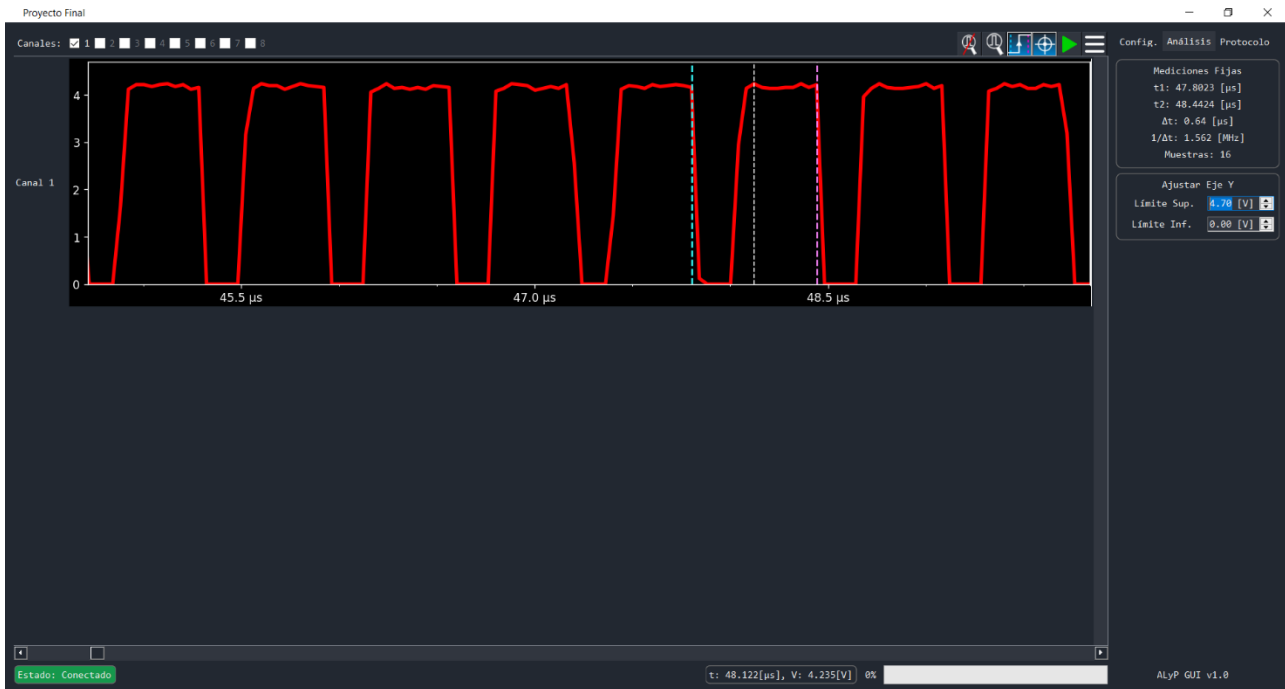


Figura 37. Canal Analógico con Cursor y Mediciones Fijas.

Si se aprecia en detalle el valor de tensión medido, se notará que el nivel alto de la señal cuadrada es de 4.25 [V], aproximadamente, cuando esta debería ser de unos 5 [V]. La atenuación introducida se debe a la siguiente combinación de factores:

- Atenuación del amplificador operacional en altas frecuencias y tensiones próximas a la de alimentación.
- Inestable tensión de alimentación y, por lo tanto, de referencia para el ADC.
- Alta impedancia de fuente (señal generada con un ATmega2560, a frecuencias cercanas a la de CLK).

Se presume, en una futura revisión de hardware, que, solventando los dos primeros puntos mediante una etapa intermedia de alimentación, se podrá recibir una representación de mejor calidad de la señal ensayada.


#### 2.4.6 Navegación

Para que el usuario pueda navegar en las muestras obtenidas, además de utilizar la barra de navegación (ver 9 en Figura 31), se agregaron las alternativas mencionadas a continuación:

- Click sobre un punto del gráfico, centra el gráfico en dicho punto.
- Scroll-Up magnifica la zona donde está el cursor.
- Scroll-Down remueve la magnificación, respecto de dónde está el cursor.

#### 2.4.7 Análisis de protocolos

Luego de implementar todas las funcionalidades del programa que le dan al usuario herramientas para tratar las señales capturadas, se procedió a desarrollar los algoritmos de decodificación de los protocolos propuestos en 1.5.

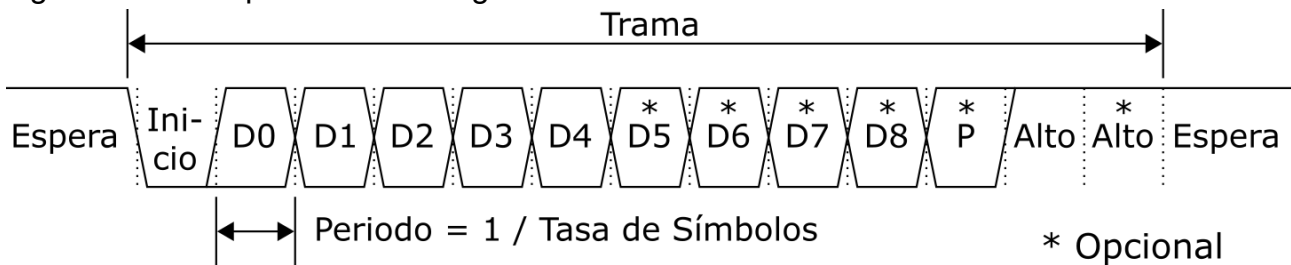
 <b>UTN Regional Paraná</b>	Analizador Lógico y de Protocolo
Ingeniería en Electrónica	Capítulo 2: Desarrollo

### 2.4.7.1 Binario

El primer decodificador no corresponde a un protocolo particular, sino que actúa como una simple herramienta de lectura de buses para no recaer en una decodificación manual de las señales digitales muestreadas. El funcionamiento de este algoritmo es extremadamente simple: recorrer el arreglo 2D de muestras y definir un mensaje cada vez que el valor binario representado en las señales cambia. Un ejemplo de este algoritmo funcionando se enseña en la Figura 47.

### 2.4.7.2 UART

UART es un protocolo de comunicación entre dispositivos electrónicos (en general, empleado en entre dos MCU) que no requiere una señal de reloj compartida. Las tramas siguen la forma que enseña la Figura 38:



*Figura 38. Trama UART  
Original en inglés obtenido de [25].*

Y siguen el formato:

- Bit de Inicio: un bit de inicio indica el comienzo de la transmisión de datos. El nivel de este bit es bajo, generando un flanco de bajada.
- Bits de Datos: pueden ser, entre 5 y 9 bits, y contienen la información transmitida.
- Bit de Paridad: un bit opcional adicional para la detección de errores, puede ser paridad par o impar.
- Bits de Parada: uno o dos bits de parada marcan el final de la transmisión de datos. Estos bits son altos, al igual que el estado de espera del bus.

La decodificación de mensajes de una UART es un poco más compleja, por lo que se detalla en los siguientes pasos:


- Definir el ancho de un símbolo a partir de la tasa de baudios.
- Buscar flancos de bajada que indiquen el comienzo de una trama.
- Leer los datos según la configuración de datos de bits, paridad y de parada.
- Si corresponde, invertir bits de datos y verificar si la paridad es correcta.

Un ejemplo de este algoritmo funcionando se enseña en la Figura 48.

### 2.4.7.3 SPI

SPI es un protocolo de comunicación serie síncrono para intercambiar datos entre dispositivos, compartiendo todos los participantes las mismas señales de reloj y datos, en una configuración de Maestro-Esclavo.

Las líneas de comunicación son las siguientes:

 <b>UTN Regional Paraná</b>	Analizador Lógico y de Protocolo
Ingeniería en Electrónica	Capítulo 2: Desarrollo

- MOSI: Línea de datos por la cual el maestro envía datos al esclavo.
- MISO: Línea de datos por la cual el esclavo envía datos al maestro.
- SCK: Señal de reloj generada por el maestro para sincronizar la transmisión de datos.
- CS: Línea utilizada por el maestro para seleccionar el esclavo con el que desea comunicarse. Cada esclavo tiene su propia línea CS.

Para lograr la comunicación, el maestro genera el reloj en la línea SCK y selecciona el esclavo adecuado bajando su línea SS/CS. Entonces, los datos se transmiten simultáneamente en ambas direcciones usando las líneas MISO y MOSI.

En particular, para definir en qué nivel el bus está inactivo y en qué momento corresponde escribir o leer los buses se detalla en la Tabla 7:

Modos SPI			
Modo	Bus inactivo si CLK en nivel	Leer en flanco de	Emitir en flanco de
0	Bajo	Subida	Bajada
1	Bajo	Bajada	Subida
2	Alto	Bajada	Subida
3	Alto	Subida	Bajada

Tabla 7. Modos SPI

Un ejemplo de una transmisión bidireccional en modo SPI 0 se enseña en la Figura 39.

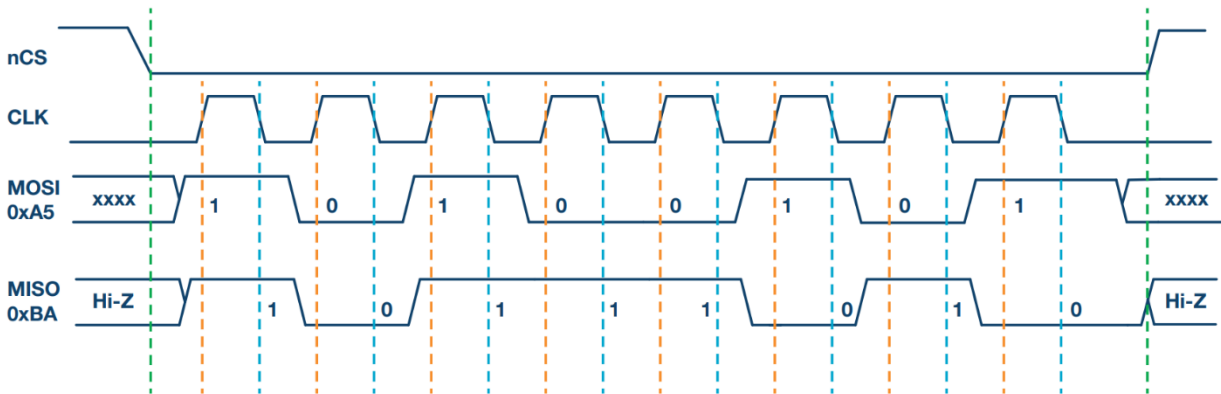



Figura 39. Trama SPI Modo 0.  
Obtenido de [27].

Con esta información, se decodifica el protocolo de la siguiente manera:

- Determinar frecuencia de CLK, puesto que el estándar de facto no la define.
- Buscar flancos (de subida o bajada, según corresponda) en la señal SCK que no tengan flancos muy próximos por detrás, a modo de discernir el inicio de las tramas.
- Verificar que, luego de los inicios de trama, existan 15 flancos (correspondientes a 8 bits), indicando que se ha capturado correctamente SCK.
- Finalmente, evaluar MISO y MOSI en los flancos correspondientes, según el modo.

Un ejemplo de este algoritmo funcionando se enseña en la Figura 49.

 <b>UTN Regional Paraná</b>	Analizador Lógico y de Protocolo
Ingeniería en Electrónica	Capítulo 2: Desarrollo

#### 2.4.7.4 I<sup>2</sup>C

I<sup>2</sup>C es un protocolo de comunicación serie síncrono, desarrollado por Phillips Semiconductor en 1980, ampliamente utilizado para conectar Microcontroladores con periféricos como sensores, memorias y otros dispositivos. Del mismo modo que SPI, se sigue una arquitectura maestro esclavo y se comparten las líneas de reloj y datos, a saber:

- SDA: Línea de datos bidireccional.
- SCL: Línea de reloj bidireccional.

Por otro lado, a diferencia de SPI, no existe una línea de selección de esclavo como CS, sino que se determina a través de direcciones de 7 o 10 bits, como se puede apreciar en la Figura 40.

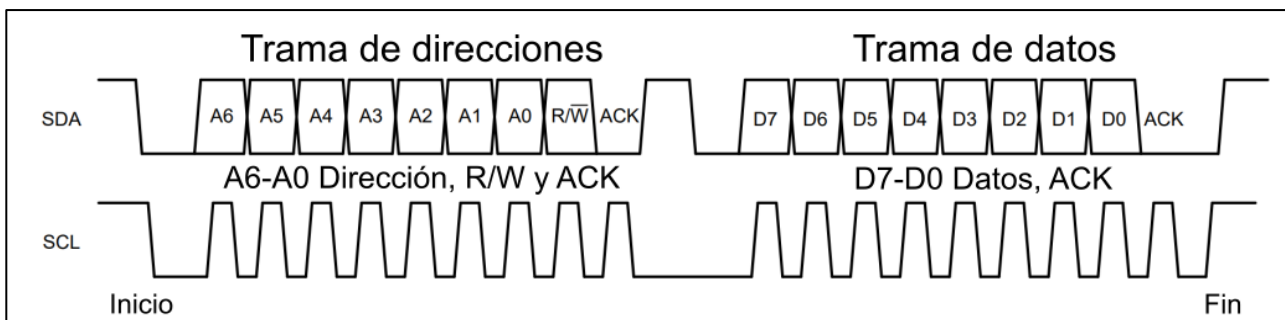



Figura 40. Trama I<sup>2</sup>C.  
Obtenido de [26].

El protocolo, bastante más complejo que los anteriores, tiene el siguiente señalizado:

- Inicio y Fin: la comunicación comienza con una condición de inicio (START) y termina con una condición de parada (STOP), ambas generadas por el maestro. START se genera cuando SDA pasa de alto a bajo mientras SCL está alto y STOP, cuando SDA pasa de bajo a alto mientras SCL está alto.
- Direccionamiento: se transmite una trama de dirección, con la identificación del destinatario, un bit indicador de si el mensaje es de lectura o escritura (R/W) y un bit de reconocimiento o acuse de recibo (ACK) que confirma la recepción del mensaje por parte del esclavo.
- Transferencia de Datos: estos se transfieren en bytes y cada byte es seguido por un ACK. Para una operación de lectura, el maestro debe generar este bit, y, para una operación de escritura, la responsabilidad es del esclavo. Se pueden transmitir todos los bytes que se deseen luego de la trama de dirección.

Con esta información, se decodifica el protocolo de la siguiente manera:

- Determinar los comienzos y fines de tramas buscando los START y STOP explicados anteriormente.
- Buscar los inicios de trama de dirección y datos de una forma similar a SPI, dentro de las condiciones de START y STOP.
- Leer los datos de las tramas encontradas, e interpretar el tipo de orden según R/W y verificar los bits de ACK. Un ejemplo de este algoritmo funcionando se enseña en la Figura 50.

 <b>UTN Regional Paraná</b>	Analizador Lógico y de Protocolo
Ingeniería en Electrónica	Capítulo 3: Resultados

## Capítulo 3: Resultados

### 3.1 Pruebas de Firmware/Hardware

Las pruebas de esta sección tienen por objetivo comprobar las funcionalidades del sistema y determinar en qué grado se completaron los objetivos descritos en 1.5.

#### 3.1.1 Captura Digital

Para poder comprobar este parámetro y la calidad de la captura, se generaron 4 señales idénticas con otro MCU RP2040 y se muestrearon con 2 canales a cada una de las señales. Las mismas son cuadradas y van desde 0 a 3.3 [V] con frecuencias de 1.25 [MHz] (visible en Figura 41) y 2 [MHz] (visible en Figura 42). La elección de estos valores no es una casualidad, puesto que la primera, es la señal digital más rápida que se ha logrado capturar con buena calidad y la segunda, la más lenta en la que la captura es regularmente deficiente. Ambas fueron muestreadas a 125 [MHz] con 8 canales, logrando una velocidad de muestreo de 1 [GSa/s]. Con estos resultados, corroboramos que se lograron las características de las filas 3 a 5 de la Tabla 2, y, por la misma métrica podemos establecer que la **señal digital más rápida** soportada es de **1 [MHz]**, con un margen de tolerancia. El programa del MCU está disponible en 7.2.

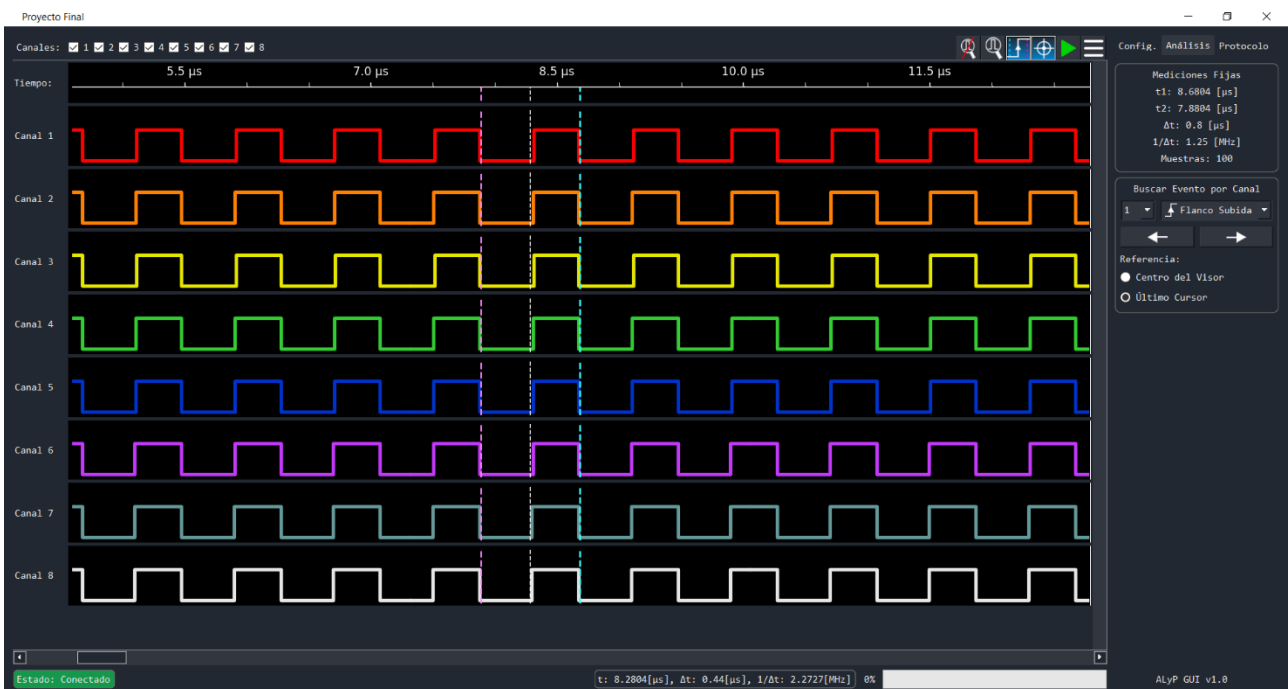


Figura 41. Prueba Captura Digital (correcta)

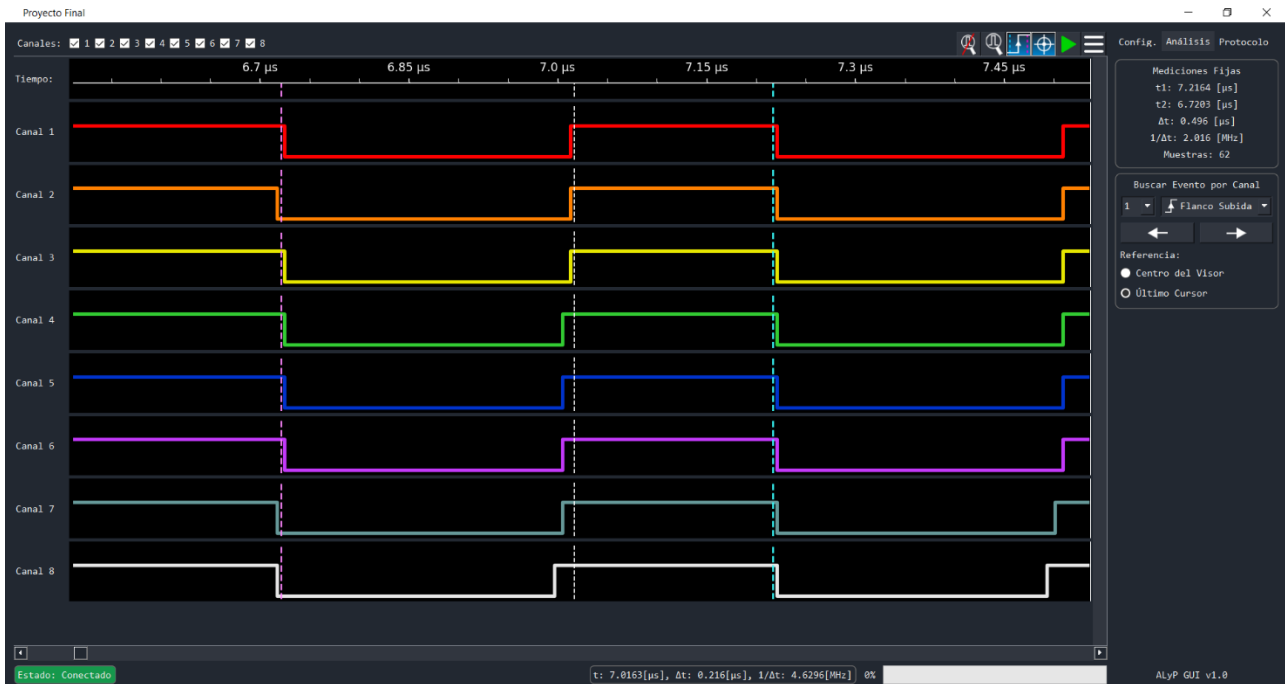


Figura 42. Prueba Captura Digital (incorrecta)

### 3.1.2 Captura analógica

Para comprobar la correcta captura del canal analógico, se generó una forma de onda arbitraria, similar a una señal triangular de 1 [MHz], con algo de distorsión como se puede ver en el trazo rojo de la Figura 43. Esta señal fue generada con un amplificador seguidor de tensión con una tasa de precesión (slew rate) baja, logrando un efecto similar al de un amplificador integrador, al recibir en su entrada la señal cuadrada (trazo amarillo).

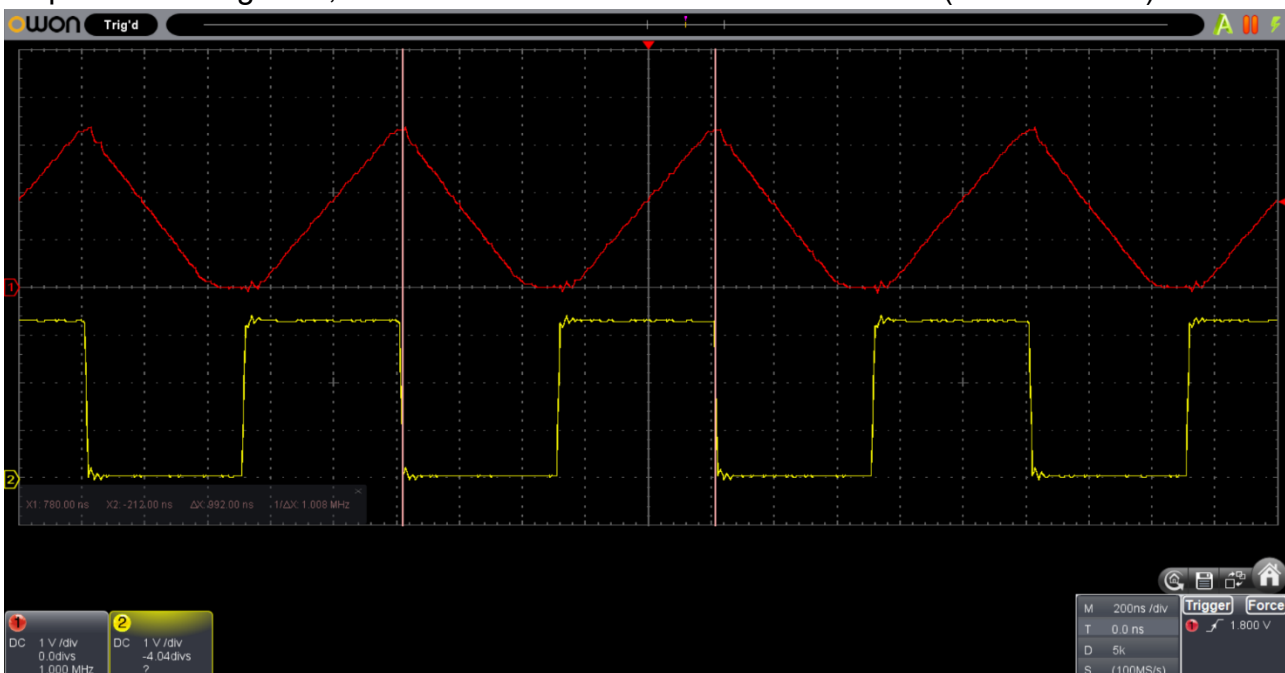


Figura 43. Señales de Prueba Captura Analógica

La captura de la señal generada con el canal analógico se puede apreciar en la Figura 44, muestreada a 25 [MHz], y se puede verificar que la captura es correcta.

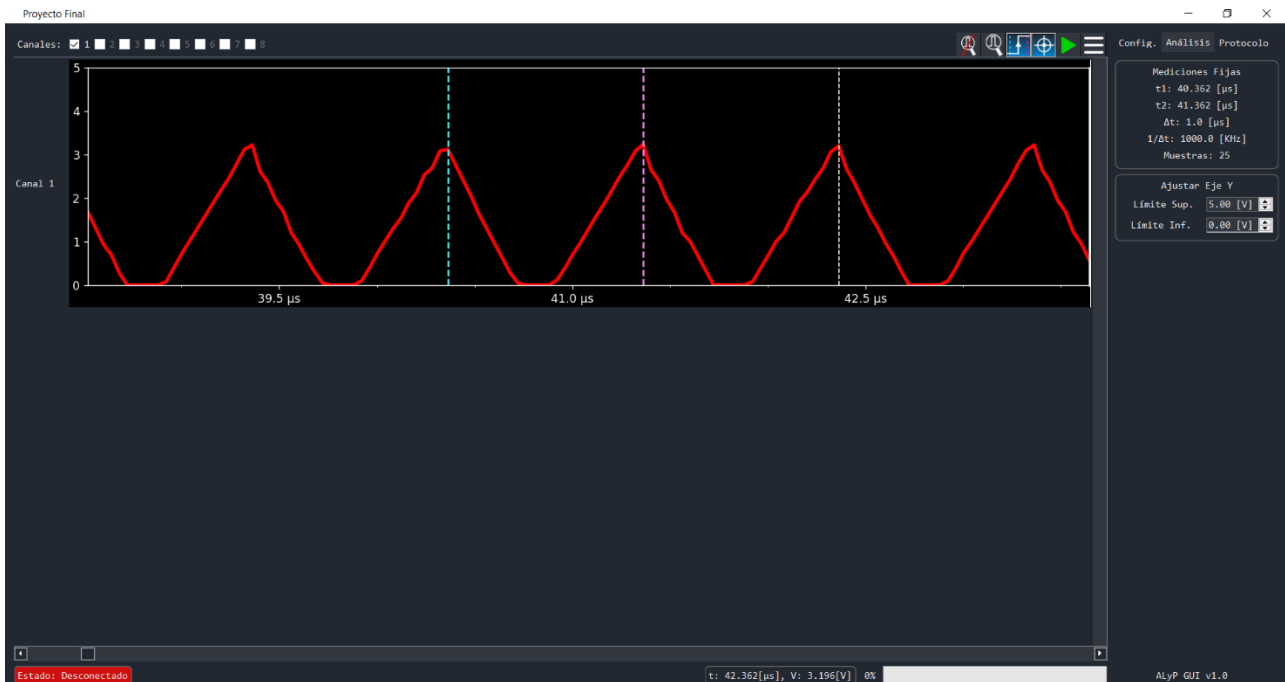


Figura 44. Prueba Captura Analógica

### 3.1.3 Nivel de Comparación

Para comprobar el buen funcionamiento de los comparadores, se introdujo la señal amarilla de la Figura 43, a 1.25 [MHz] en los canales 1, 2, 4, 5, 6 y 8 del analizador y la señal roja de la misma figura, en los canales 3 y 7, con un nivel de comparación de 1.65 [V], muestreados a 125 [MHz]. Los resultados son visibles en la Figura 45, que demuestra el comportamiento esperado de un comparador analógico, al coincidir los cambios de las señales de los canales 3 y 7 en el momento que la señal (cuasi)triangular supera el umbral de tensión seleccionado.

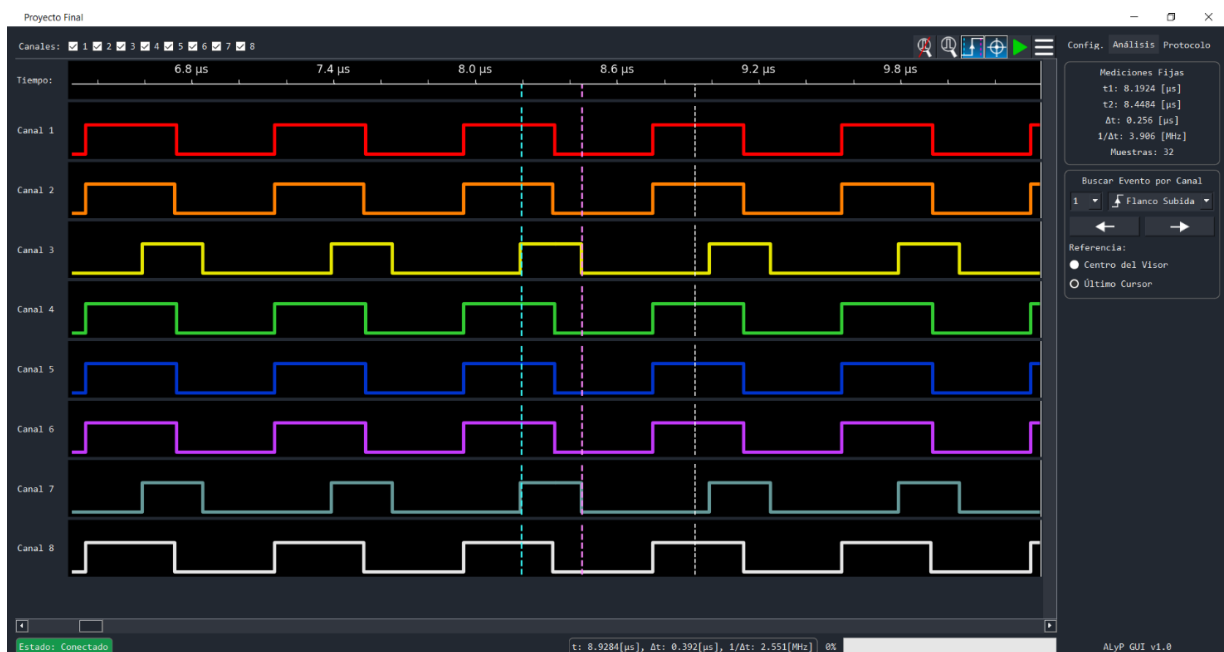



Figura 45. Prueba Comparadores Analógicos

 <b>UTN Regional Paraná</b>	Analizador Lógico y de Protocolo
Ingeniería en Electrónica	Capítulo 3: Resultados

### 3.1.4 Muestreo Gatillado

Hasta este punto, todas las pruebas de captura se han realizado con un muestreo no gatillado, es decir, inmediatamente después de llegada la orden al MCU. A continuación, se detalla en la Figura 46 una captura gatillada de un contador de rizado, generada con el programa de 7.2 en un MCU Atmega2560, esperando un flanco de subida en el canal 7.

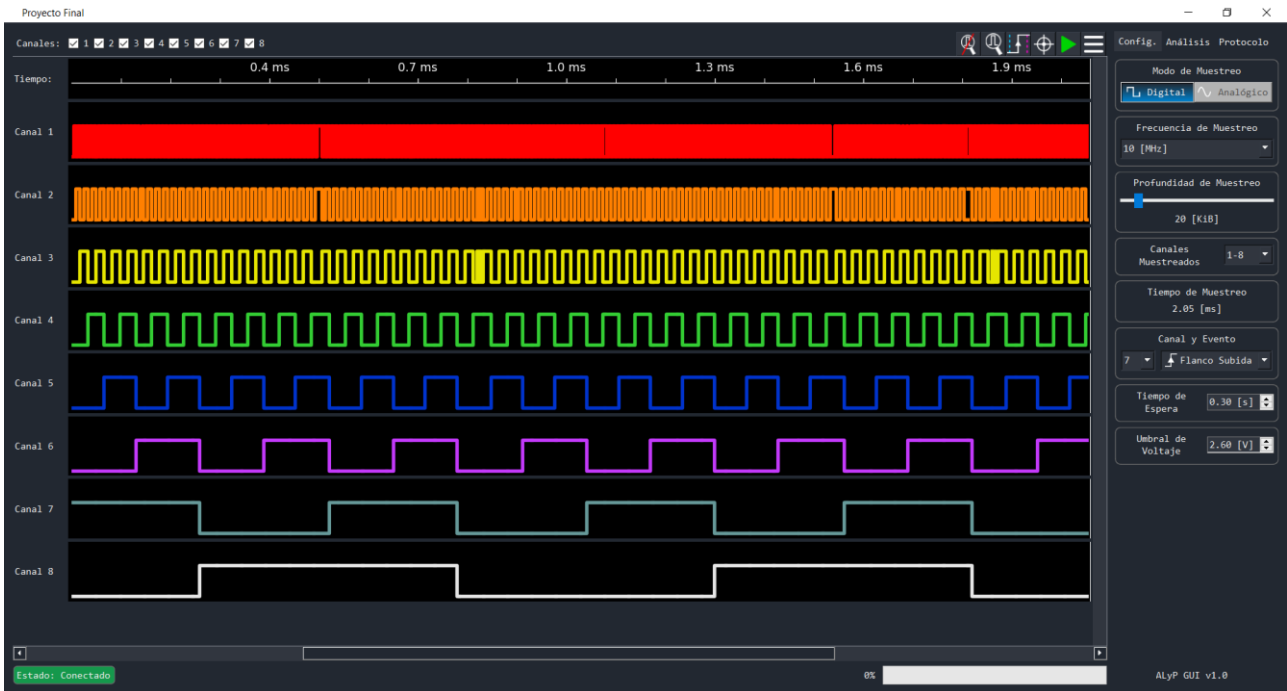


Figura 46. Prueba Muestreo Gatillado

Si bien esta captura refleja un comportamiento correcto, comenzando el muestreo justo después de el flanco de subida (verificable analizando que el largo del resto de periodos es igual al primero), requirió varios intentos para lograrse, por ruidos que alteran los primeros ciclos de captura de los periféricos PIO. Esto hace que el gatillado, aunque correctamente programado en firmware, funcione **defectuosamente en algunos casos** por defectos de hardware, como la limitación de PCB (agujeros NPTH) descrita en 2.2.10. Estos efectos se ven mitigados si:

- Baja la frecuencia de muestreo.
- No hay un muestreo previo, o se han muestreado todos niveles bajos.
- Sube la tensión de referencia.

## 3.2 Protocolo

Las pruebas de esta sección tienen por objetivo comprobar el correcto funcionamiento de los algoritmos de decodificación de protocolos. Todos los programas “Maestro” fueron ejecutados en una placa de desarrollo Arduino Mega (AtMega2560), mientras que, los programas “Esclavo”, corrieron en un Arduino Nano (AtMega328).




### 3.2.1 Binario

Para esta sección se utilizó nuevamente el programa de 7.2, puesto que permite visualizar claramente, en la Figura 47, la cuenta ascendente del contador de rizado al ser decodificado como binario. Como se puede corroborar, la decodificación de los valores y los comienzos y fin de “tramas” ocurren correctamente.



Figura 47. Prueba Decodificar Binario

 <b>UTN Regional Paraná</b>	Analizador Lógico y de Protocolo
Ingeniería en Electrónica	Capítulo 3: Resultados

### 3.2.2 UART

Para esta prueba, visible en la Figura 48, se envía la cadena de caracteres “Hello From UART\n” en el maestro (visible en el canal 1), y como el programa es un simple loopback en el esclavo, este responde con la misma cadena, con un retardo de una trama (visible en el canal 2). Las UART están trabajando a 250000 baudios, con 8 bits de datos, 1 bit de paridad par, cuya corroboración se puede apreciar en los “+POK” de los mensajes, y un bit de alto. Por lo exhibido, se concluye que el funcionamiento es correcto.

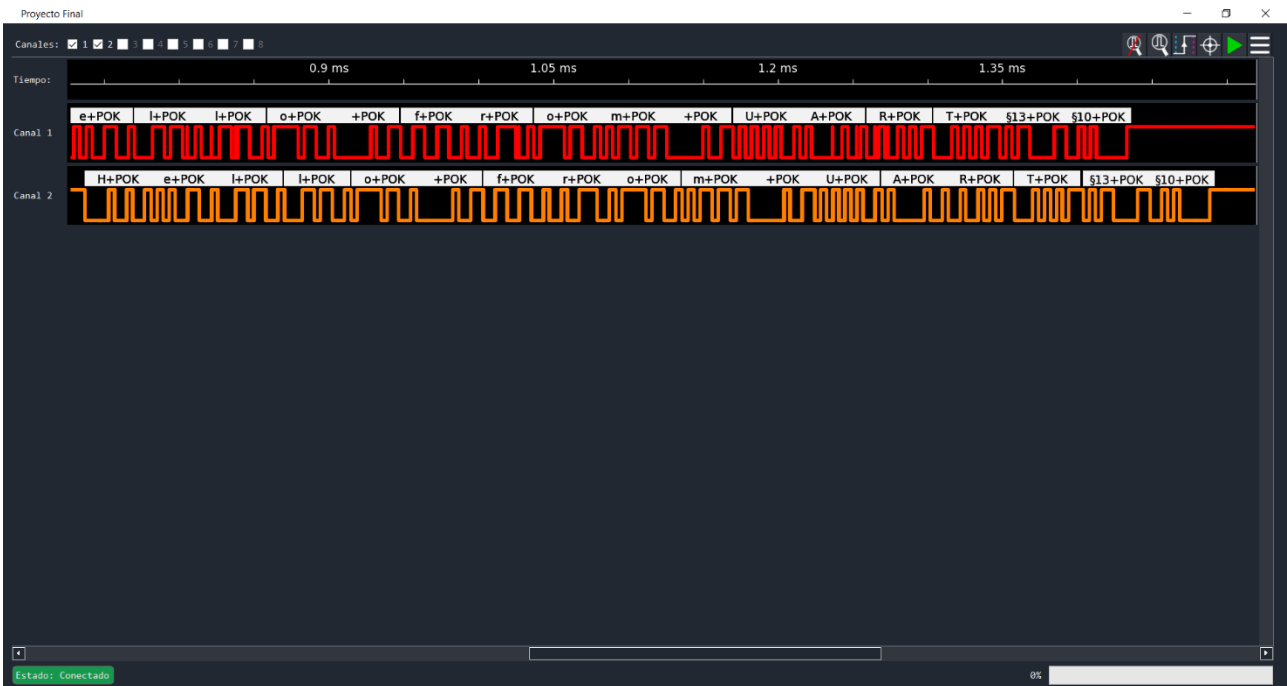


Figura 48. Prueba Decodificador UART

### 3.2.3 SPI

Para esta prueba, visible en la Figura 49, se envía la cadena de caracteres “Hello From SPI!” en el maestro y como el programa es un simple loopback en el esclavo, este responde con la misma cadena, con un retardo de una trama. Los periféricos SPI están trabajando en SPI Mode 0, a una frecuencia de reloj de 1 [MHz]. Las señales son:

- Canal 1, SCK.
- Canal 2, CS.
- Canal 3, MOSI.
- Canal 4, MISO.

Por lo exhibido, se concluye que el funcionamiento es correcto.

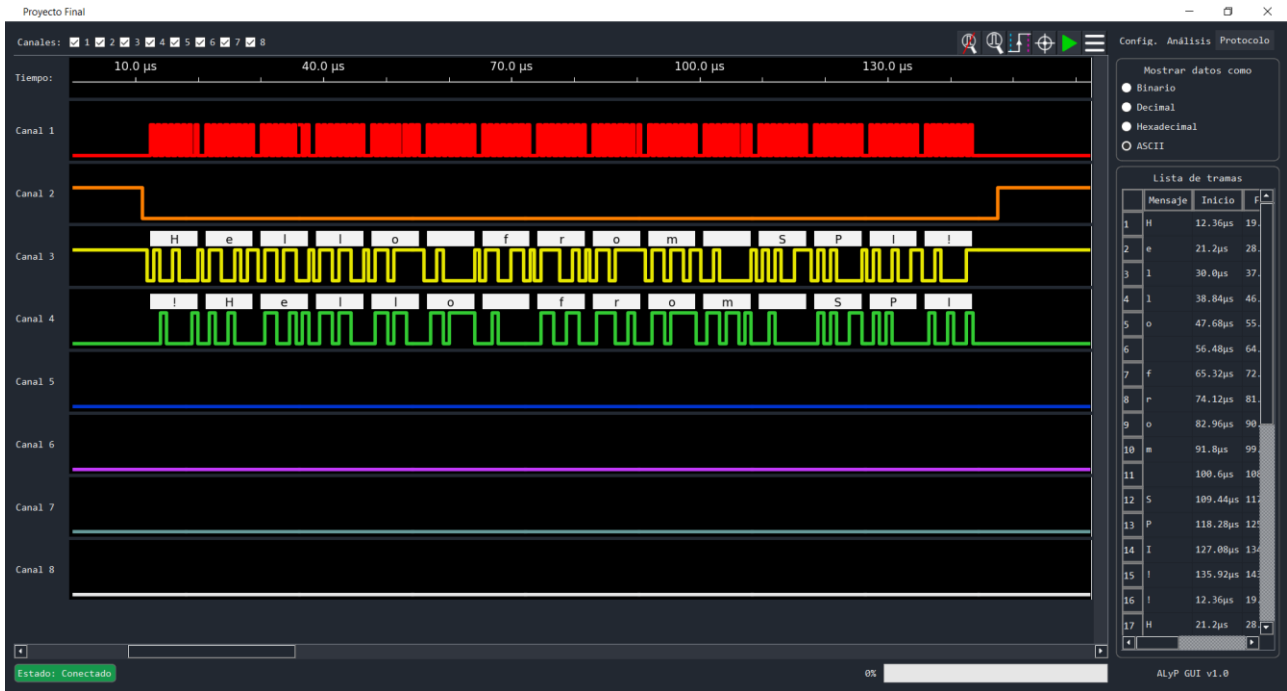


Figura 49. Prueba Decodificador SPI

### 3.2.4 I<sup>2</sup>C

La prueba de I<sup>2</sup>C es ligeramente más compleja que las anteriores. Para comprobar el funcionamiento el maestro solicita 4 bytes del esclavo número 4 cada 500 [ms] a través de una orden de lectura, visible en la Figura 50. El esclavo, por su parte, está programado para contestar con 4 caracteres (debido a las limitaciones de la librería de Arduino) de la cadena: "Hola desde I2C!" en cada orden de lectura, visibles en la Figura 51, en ese caso desde el segundo al quinto carácter. Durante el tiempo que el esclavo procesa qué caracteres contestará, mantiene el bus ocupado (en nivel bajo), visible en la Figura 52. Por lo exhibido, se concluye que el funcionamiento es correcto.

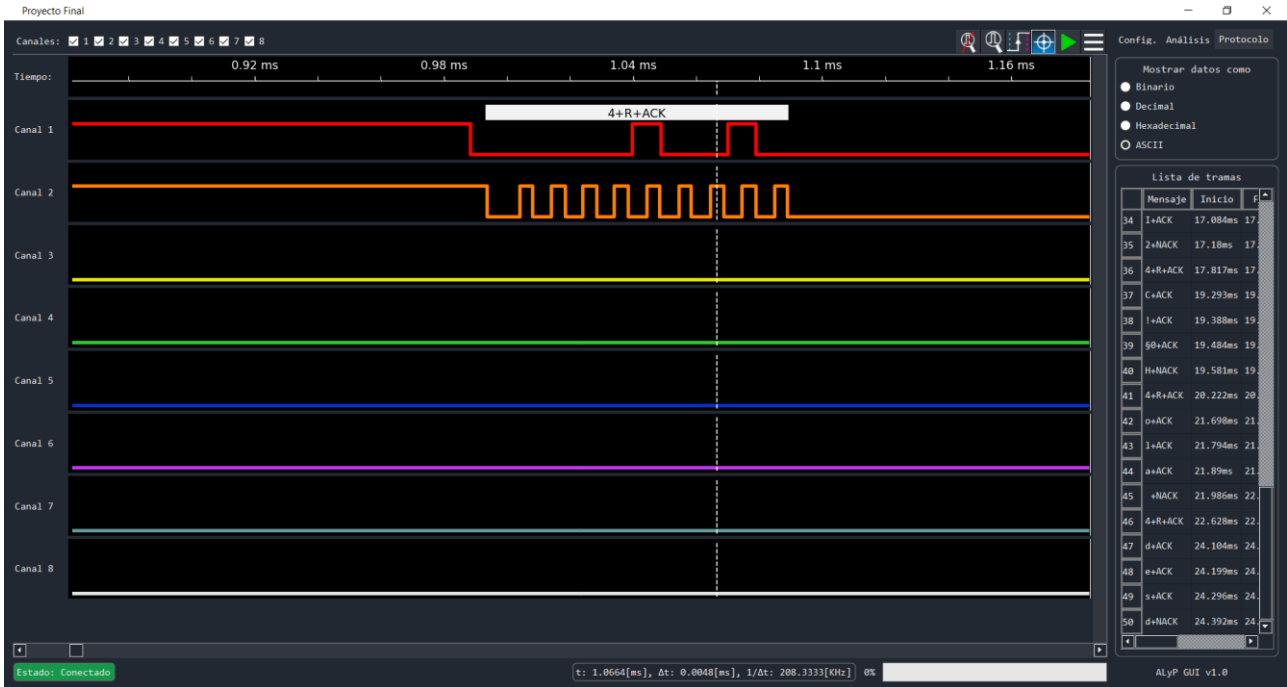


Figura 50. Prueba Decodificado I<sup>2</sup>C Orden Lectura

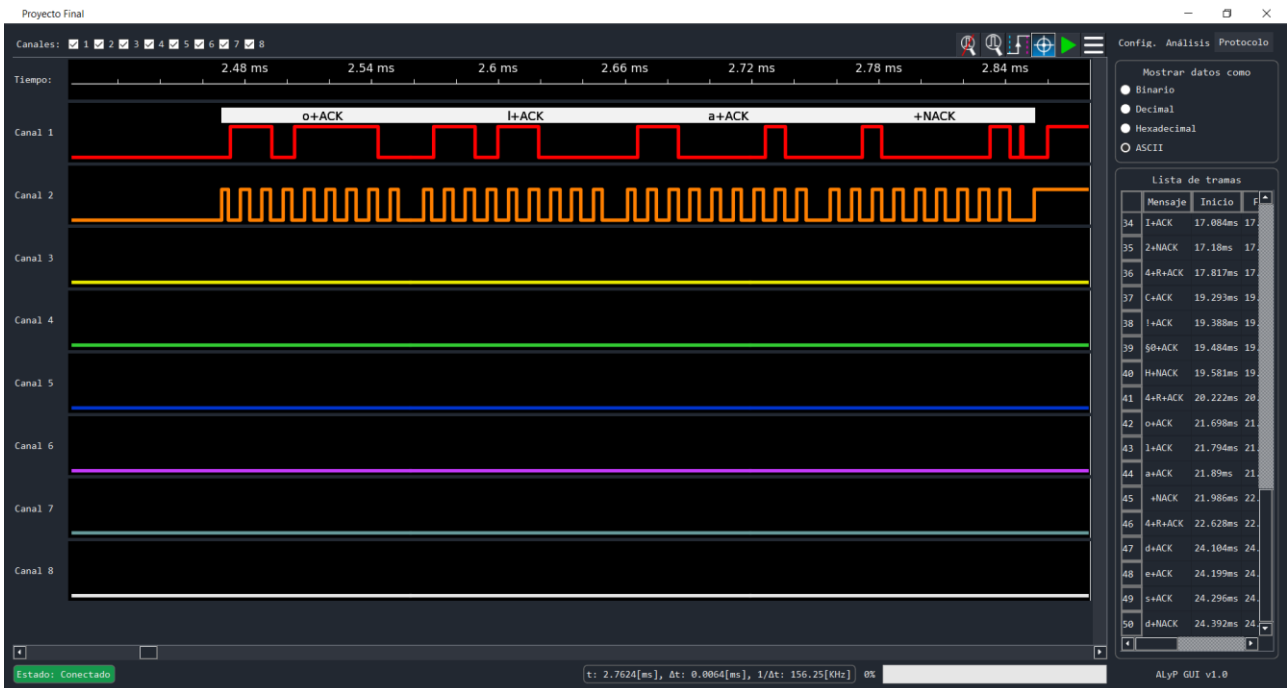


Figura 51. Prueba Decodificado I<sup>2</sup>C Respuesta Lectura

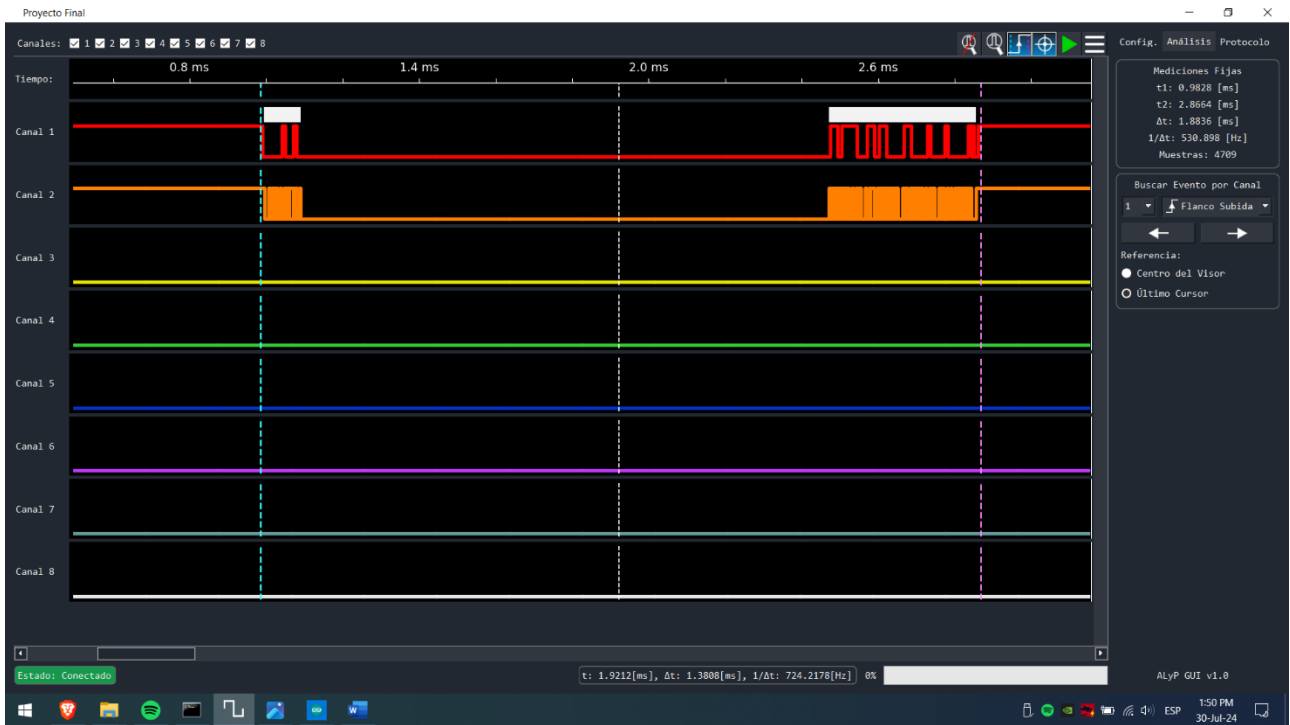



Figura 52. Prueba Decodificado I<sup>2</sup>C Transacción Completa

 <b>UTN Regional Paraná</b>	Analizador Lógico y de Protocolo
Ingeniería en Electrónica	Capítulo 4: Análisis de Costos

## Capítulo 4: Análisis de Costos

### 4.1 Presupuesto


A continuación, en la Tabla 8, se presentan los costos individuales de cada uno de los componentes electrónicos, insumos, horas de trabajo y cualquier otro producto o servicio en los que se haya incurrido en gastos para la realización del proyecto.

Presupuesto Proyecto Final				
Artículo	Modelo/Valor	Cantidad	Precio Unitario (U\$D)	Precio Total (U\$D)
Resistor	150 [Ω] 1% 1/10 [W]	2	\$0.06	\$0.11
Resistor	150 [kΩ] 1% 1/10 [W]	3	\$0.06	\$0.17
Resistor	1 [MΩ] 1% 1/2 [W]	8	\$0.05	\$0.39
Capacitor Tantalio	10 [μF]	5	\$1.24	\$6.21
Capacitor Cerámico	100 [nF]	6	\$0.13	\$0.77
Capacitor Electrolítico	100 [μF]	1	\$0.41	\$0.41
Capacitor Cerámico	10 [nF]	1	\$0.13	\$0.13
Inductor Polvo de Hierro	330 [μH]	2	\$6.06	\$12.13
Diodo	BAT54C	8	\$0.36	\$2.88
Transistor PNP	BC807	8	\$0.11	\$0.91
Amplificador Operacional	LM358	1	\$0.37	\$0.37
Amplificador Operacional	ADA4891-1ARJZ-RL	1	\$1.88	\$1.88
Comparador Analógico	MAX9108ESD_	2	\$5.27	\$10.54
Convertor de Nivel	SN74LVTH245ADWR	1	\$0.86	\$0.86
Convertor Analógico Digital	ADC1175CIMTC/NOPB	1	\$5.00	\$5.00
Convertor Digital Analógico	MCP4821	1	\$14.30	\$14.30
Regulador de Voltaje Lineal	AMS1117-3.3	1	\$3.84	\$3.84
Placa de Desarrollo	Raspberry Pi Pico	1	\$19.64	\$19.64
Fusible	250 [mA]	1	\$1.07	\$1.07
Pulsador	Tact Switch	1	\$0.25	\$0.25
Tira de pines hembra	40 x 2.54 [mm]	0.33	\$1.88	\$0.62
Bornera atornillable	2 x 5.08 [mm]	1	\$2.86	\$2.86
Tira de cables	8 x 20 [cm]	1	\$2.86	\$2.86
PCB	Proveedor Local	1	\$11.00	\$11.00
Mano de obra e insumos	Ensamblado	1	\$50.00	\$50.00
Fletes Varios	Desde EE. UU.	3	\$30.00	\$90.00
Impresión 3D	Gabinete	1	\$10.00	\$10.00
Cable USB	USB A a USB Micro B	1	\$3.57	\$3.57
Subtotal Costos Variables				\$252.76
Artículo	Descripción	Cantidad	Precio Unitario (U\$D)	Precio Total (U\$D)
Hora Ingenio Desarrollo	R&D	300	\$7.00	\$2,100.00
Subtotal Ingeniería No Recurrente				\$2,100.00
Precio de venta, Amortización en 20 ventas, Sin impuestos				\$357.76

Tabla 8. Presupuesto del Proyecto


Para obtener los precios, se utilizaron los siguientes criterios:

- Cuando los artículos fueran adquiridos en plataformas nacionales como MercadoLibre, se usaron los valores en AR\$ convertidos a U\$D con la cotización del dólar blue del día de consulta.

 <b>UTN Regional Paraná</b>	Analizador Lógico y de Protocolo
Ingeniería en Electrónica	Capítulo 4: Análisis de Costos

- Cuando fueran adquiridos en distribuidores en el extranjero, se utilizaron los valores minoristas en U\$D para cada artículo y se tuvieron en cuenta los costos de envío por ser muy significativos.
- Para los costos de desarrollo, se tuvo como referencia la cotización de unidad arancelaria “ingenio” del Colegio de Ingenieros Especialistas de Entre Ríos [22], vigentes para el mes de agosto de 2024.
- Para amortizar la inversión de los costos de desarrollo, se definió repartirlo en el costo de las primeras 20 unidades vendidas.

Es necesario mencionar que este presupuesto refleja la eficiencia de una escala de producción de una unidad. Al adquirir los insumos a granel, como importador o ambos, el costo se vería **notablemente reducido**, haciéndolo aún más competitivo respecto de las otras alternativas del mercado.

 <b>UTN Regional Paraná</b>	Analizador Lógico y de Protocolo
Ingeniería en Electrónica	Capítulo 5: Conclusiones

## Capítulo 5: Conclusiones

### 5.1 Objetivos Cumplidos

Para verificar el grado de cumplimiento de los objetivos de 1.5, es conveniente recordar la mayor parte de la información de la Tabla 2, como se muestra en la Tabla 9:

Parámetro	Valor				Unidad
	Mínimos	Objetivos	Logrado	Cumplido	
Dispositivo					-
Voltaje de Entrada	0 a 5	0 a 9	0 a 5	✓	[V]
Número de Entradas Digitales	4	8	8	✓	-
Frecuencia de Muestreo Digital	40	1000	1000	✓	[MSa/s]
Número de Entradas Analógicas	1	1	1	✓	-
Frecuencia de Muestreo Analógica	5@8 bits	25@8 bits	25@8 bits	✓	[MSa/s]
Umbral de Voltaje (Mín.)	Var.: 0	Var.: 0	Var.: 0	✓	[V]
Umbral de Voltaje (Máx.)	Var.:3.3	Var.:3.3	Var.:3.3	✓	[V]


Tabla 9. Objetivos planteados y su grado de cumplimiento  
"Var." significa "variable"

Se cumplieron todos los parámetros mínimos propuestos para el proyecto, y todos los parámetros objetivo excepto el rango de voltaje de entrada. Además, se incorporaron correctamente todas las funcionalidades de muestreo gatillado, y los 4 decodificadores de protocolo propuestos. Para comparar las capacidades técnicas con alternativas comerciales, se puede revisar la Tabla 10.

Parámetro	Valor				Unidad
	Kingst	Proyecto Final	Saleae		
Fabricante	Kingst	Proyecto Final	Saleae		-
Modelo	LA1010	Proyecto Final	Logic 8	Logic Pro 16	-
Frecuencia de Muestreo	100@3Canales	1000	100	500	[MSa/s]
Canales Digitales	16	8	8	16	-
Señal Digital más rápida	20	1	25	100	[MHz]
Captura Analógica	No	Sí (1 canal)	Si (8 Canales)	Si (16 Canales)	-
Rango de Voltaje	±50	0 a 5	±25	±25	[V]
Umbral de Voltaje (Mín.)	Var.: -4 mín.	Var.: 0	Fijo: 0.6	Fijos: 0.6, 1.2, 1.65	[V]
Umbral de Voltaje (Máx.)	Var.: 4 máx.	Var.: 3.3	Fijo: 1.2		[V]
Precio de Venta	68	357.76	499	1499	USD

Tabla 10. Comparativa con alternativas comerciales.  
"Var." significa "variable". Datos obtenidos de [3, 4]




 <b>UTN Regional Paraná</b>	Analizador Lógico y de Protocolo
Ingeniería en Electrónica	Capítulo 5: Conclusiones

El instrumento diseñado, aún con el elevado precio que definimos en la sección anterior, ofrece un buen compromiso entre las capacidades del Kingst LA1010 y el Saleae Logic 8, en un punto intermedio del rango de precios. Esto resulta ideal para las personas que no deseen desembolsar 500 U\$D para hacer un simple depurado de un circuito digital o protocolo de comunicación, pero tampoco resignarse a frecuencias de muestreo bajas o la ausencia de un canal analógico. La principal debilidad del producto es su inmadurez, propia del punto del proceso de diseño (que, a esta altura, es poco más que un prototipo, ni siquiera un MVP) que tiene, dado que compite con alternativas que llevan al menos 10 años de existencia en el mercado y múltiples revisiones de software y hardware. Si se deseara llegar al punto de MVP, se deberían implementar las dos primeras mejoras de 5.2.

## 5.2 Futuras Mejoras


Existen, al momento de la redacción, decenas de mejoras para incorporar al proyecto, consistiendo la gran mayoría de ellas en más funcionalidades de la GUI o mejoras en las capacidades de hardware. A continuación, se detallan en orden de prioridad los principales posibles avances a concretar:

1. Revisión de Hardware y PCB. Principalmente, para reducir ruido en la captura y estabilizar alimentación.
2. Cambio de Framework Gráfico. Aunque extremadamente completo y maduro, Matplotlib está diseñado para priorizar calidad antes que tiempo de procesamiento, que hace la experiencia del usuario algo fastidiosa.
3. Incorporar las funcionalidades de guardar a (y cargar desde) archivos el muestreo realizado y generar informes en PDF desde la GUI.
4. Utilizar comparadores analógicos de mayor velocidad para mejor aprovechar la frecuencia de muestreo del MCU.
5. Incorporar muestreo analógico a todos los canales.
6. Incorporar hardware para permitir un mayor rango de voltaje de entrada.

 <b>UTN Regional Paraná</b>	Analizador Lógico y de Protocolo
Ingeniería en Electrónica	Capítulo 6: Referencias

## Capítulo 6: Referencias

- [1] ENACOM, «Informe de Gestión 2021,» 2021.
- [2] ENACOM, «Informe de Gestión 2022,» 2022.
- [3] QDKingst, «Product Selection of Logic Analyzer,» 2024. [En línea]. Disponible: <https://www.qdkingst.com/en/products>. [Último acceso: 3 Julio 2024].
- [4] Saleae, «Saleae Logic 8,» 2024. [En línea]. Disponible: <https://www.saleae.com/products/saleae-logic-8>. [Último acceso: 3 Julio 2024].
- [5] Mouser Electronics, «F0201G0R03FNTR\500,» 4 Julio 2024. [En línea]. Disponible: <https://ar.mouser.com/ProductDetail/KYOCERA-AVX/F0201G0R03FNTR%5C500?qs=sGAEpiMZZMslz3CjQ1xegZx2pdJYlyprFm4IV%2FkDHIC64g%2FdjgANcg%3D%3D>. [Último acceso: 4 Julio 2024].
- [6] Diodes Incorporated, «BAT54 /A /C /S,» Noviembre 2023. [En línea]. Disponible: [https://www.diodes.com/assets/Datasheets/BAT54\\_A\\_C\\_S.pdf](https://www.diodes.com/assets/Datasheets/BAT54_A_C_S.pdf). [Último acceso: 4 Julio 2024].
- [7] Diodes Incorporated, «BC807-16/-25/-40,» Septiembre 2021. [En línea]. Disponible: [https://www.diodes.com/assets/Datasheets/BC807-16\\_-25\\_-40.pdf](https://www.diodes.com/assets/Datasheets/BC807-16_-25_-40.pdf). [Último acceso: 4 Julio 2024].
- [8] Analog Devices, «AN-849 Using Op Amps as Comparators,» 2011. [En línea]. Disponible: <https://www.analog.com/media/en/technical-documentation/application-notes/an-849.pdf>. [Último acceso: 5 Julio 2024].
- [9] Maxim Integrated, «MAX9107/MAX9108/MAX9109 Datasheet,» Marzo 2018. [En línea]. Disponible: <https://www.analog.com/media/en/technical-documentation/data-sheets/MAX9107-MAX9109.pdf>. [Último acceso: 5 Julio 2024].
- [10] Texas Instruments, «ADC1175 Datasheet,» Abril 2013. [En línea]. Disponible: [https://www.ti.com/lit/ds/symlink/adc1175.pdf?ts=1720369812403&ref\\_url=https%253A%252F%252Fwww.google.com%252F](https://www.ti.com/lit/ds/symlink/adc1175.pdf?ts=1720369812403&ref_url=https%253A%252F%252Fwww.google.com%252F). [Último acceso: 8 Julio 2024].
- [11] Texas Instruments, «LMH6702 Datasheet,» Mayo 2016. [En línea]. Disponible: <https://www.ti.com/lit/ds/symlink/lmh6702.pdf>. [Último acceso: 8 Julio 2024].
- [12] Texas Instruments, «SN10501 Datasheet,» Enero 2009. [En línea]. Disponible: <https://www.ti.com/lit/ds/symlink/sn10501.pdf?ts=1720440546523>. [Último acceso: 8 Julio 2024].
- [13] Texas Instruments, «OPAx357 Datasheet,» Abril 2018. [En línea]. Disponible: <https://www.ti.com/lit/ds/symlink/opa357.pdf>. [Último acceso: 8 Julio 2024].
- [14] Analog Devices, «ADA4891-1 Datasheet,» 2015. [En línea]. Disponible: [https://www.analog.com/media/en/technical-documentation/data-sheets/ADA4891-1\\_4891-2\\_4891-3\\_4891-4.PDF](https://www.analog.com/media/en/technical-documentation/data-sheets/ADA4891-1_4891-2_4891-3_4891-4.PDF). [Último acceso: 08 Julio 2024].

 <b>UTN Regional Paraná</b>	Analizador Lógico y de Protocolo
Ingeniería en Electrónica	Capítulo 6: Referencias

- [15] Texas Instruments, «OPAx354 Datasheet,» Abril 2018. [En línea]. Disponible: <https://www.ti.com/lit/ds/symlink/opa354.pdf>. [Último acceso: 8 Julio 2024].
- [16] Texas Instruments, «OPA810 Datasheet,» Julio 2020. [En línea]. Disponible: <https://www.ti.com/lit/ds/symlink/opa810.pdf>. [Último acceso: 8 Julio 2024].
- [17] Texas Instruments, «SN74LVTH245A Datasheet,» Septiembre 2003. [En línea]. Disponible: <https://www.ti.com/lit/ds/symlink/sn74lvth245a.pdf>. [Último acceso: 8 Julio 2024].
- [18] Raspberry Pi Foundation, «RP2040 Datasheet,» 02 Mayo 2024. [En línea]. Disponible: <https://datasheets.raspberrypi.com/rp2040/rp2040-datasheet.pdf>. [Último acceso: 8 Julio 2024].
- [19] Raspberry Pi Foundation, «Raspberry Pi Pico Datasheet,» 2 Mayo 2024. [En línea]. Disponible: <https://datasheets.raspberrypi.com/pico/pico-datasheet.pdf>. [Último acceso: 9 Julio 2024].
- [20] Texas Instruments, «Debounce a Switch,» Octubre 2020. [En línea]. Disponible: <https://www.ti.com/lit/ab/scea094/scea094.pdf>. [Último acceso: 9 Julio 2024].
- [21] USB Implementers Forum, «USB 2.0 Specification,» 27 Abril 2000. [En línea]. Disponible: <https://www.usb.org/document-library/usb-20-specification>. [Último acceso: 9 Julio 2024].
- [22] Colegio de Ingenieros Especialistas de Entre Ríos, «Resoluciones,» 07 Julio 2024. [En línea]. Disponible: <https://www.cieer.org.ar/release/images/documentos/resoluciones/2049%202024NUEVO%20VALOR%20INGENIO.pdf>. [Último acceso: 31 Julio 2024].
- [23] Wikipedia, «File:Waveforms es.svg,» 27 Septiembre 2016. [En línea]. Disponible: [https://commons.wikimedia.org/wiki/File:Waveforms\\_es.svg](https://commons.wikimedia.org/wiki/File:Waveforms_es.svg). [Último acceso: 3 Julio 2024].
- [24] R. L. Boylestad y L. Nashelsky, Electrónica: Teoría de circuitos y dispositivos electrónicos, Ciudad de México: Pearson, 2018.
- [25] AmenophisIII, «UART Frame.svg,» 5 Mayo 2017. [En línea]. Disponible: [https://commons.wikimedia.org/wiki/File:UART\\_Frame.svg](https://commons.wikimedia.org/wiki/File:UART_Frame.svg). [Último acceso: 23 Julio 2024].
- [26] Texas Instruments, «A Basic Guide to I2C,» Noviembre 2022. [En línea]. Disponible: <https://www.ti.com/lit/an/sbaa565/sbaa565.pdf>. [Último acceso: 23 Julio 2024].
- [27] Analog Devices, «Introduction to SPI Interface,» Septiembre 2018. [En línea]. Disponible: <https://www.analog.com/media/en/analog-dialogue/volume-52/number-3/introduction-to-spi-interface.pdf>. [Último acceso: 23 Julio 2024].

## Capítulo 7: Anexos

### 7.1 Esquemático PCB

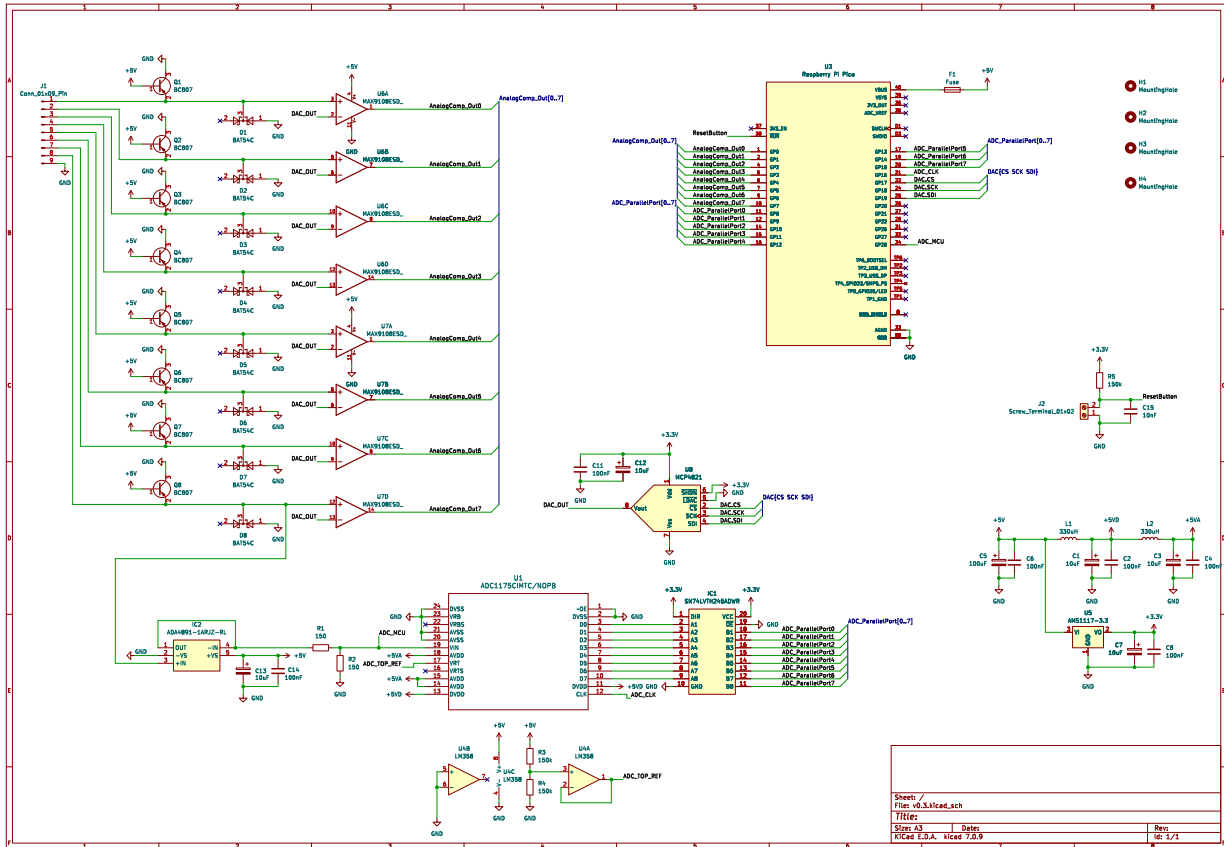


Figura 53. Esquemático Completo

### 7.2 Programas de prueba

Todos los programas de prueba, así como el resto de los archivos del proyecto, está disponible en <https://github.com/KarimHacen7/ProyectoFinal>