



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

---

---

FACULTAD DE INGENIERÍA

Medición de Alta Resolución y Sensibilidad  
para Sensores Resistivos: Instrumentación y  
Aplicaciones.

TESIS

QUE PARA OBTENER EL TÍTULO DE  
Ingeniera Eléctrica Electrónica

PRESENTA

**Tonantzin Lozano Herrera**

DIRECTOR DE TESIS

**Dr. Roberto Giovanni Ramírez Chavarría**



Ciudad Universitaria, Cd. Mx., 2023



## *Dedicado a ...*

*A mis papás, Keyla y Valentín, por todo su amor, apoyo y confianza incondicionales.  
Por ser un ejemplo de cómo se alcanzan las metas y por todas las oportunidades que  
me han brindado en la vida.*

*A mis hermanos, Itza, Lola, Vale y Atzin, por su apoyo, amor y empatía, porque hubo  
noches largas de trabajo en las que sin razón también estuvieron ahí.*

*Y a todos aquellos que han dejado una enseñanza en mi para poder crecer y apreciar  
lo que la vida me ha dado.*



# Reconocimientos

---

A la Universidad Nacional Autónoma de México (UNAM), mi *alma mater* por abrirme las puertas del conocimiento.

A la Facultad de Ingeniería por brindarme las herramientas profesionales necesarias pero también por darme la oportunidad de conocer a tantas buenas personas.

A mi director de tesis, el Dr. Roberto Ramírez Chavarría, por todo su apoyo en todas mis ideas, su fe en mí y su inmensa paciencia. También, por compartir conmigo su conocimiento y por dedicar su esfuerzo para concluir este trabajo. Le agradezco infinitamente su ayuda.

A la FHA por aceptarme como invitada en tiempos tan turbulentos y que a pesar de estas circunstancias me dieran la oportunidad de colaborar con sus distinguidos investigadores.

*Ich danke der Fachhochschule Aachen (FHA), dass sie mich in solch komplizierte Zeiten als Gast mir empfangen hat. Sie hat mir die Möglichkeit mit ihren angesehenen Forschern zusammenzuarbeiten gegeben.*

Al Dr. Prof. Schöning por su hospitalidad y apoyo en durante mi breve estancia en la FHA. Al Dr. Stefan Achtsnicht y al M.C. Kevin Janus por su amistad, colaboración y por su paciencia a mis preguntas.

*Ich danke Dr. Prof. Schöning für seine Gastfreundschaft und Unterstützung während meines kurzen Aufenthalts an der FHA. An Dr. Stefan Achtsnicht und M.Sc. Kevin Janus für ihre Freundschaft, Zusammenarbeit und ihre Geduld mit meinen Fragen.*

A mis amigos Alan, Gabriel, Fernando, Efrén y demás, que sin su amistad y las horas de trabajo, nada de esto sería posible, gracias por darme la oportunidad de recorrer juntos los caminos de la facultad.

A los sinodales por su tiempo y revisión.

Esta tesis fue realizada gracias:

- Al Programa de Apoyo a Proyectos para Innovar y Mejorar la Educación (PAPI-ME) a través del proyecto UNAM-PAPIME PE100523 y PE101222.
- Al Programa de Apoyo a Proyectos de Investigación e Innovación Tecnológica (PAPIIT) a través del proyecto UNAM-PAPIIT TA101423 y TA100221.
- Al programa de becas de Iniciación a la Investigación (Verano-Otoño 2021) de la Dirección General de Cooperación e Internacionalización (DGECI).







# Declaración de autenticidad

---

Por la presente declaro que, salvo cuando se haga referencia específica al trabajo de otras personas, el contenido de esta tesis es original y no se ha presentado total o parcialmente para su consideración para cualquier otro título o grado en esta o cualquier otra Universidad. Tómese en consideración que los conceptos presentados en este trabajo han sido tomados de diversas referencias y dependiendo el caso se han traducido, adaptado y resumido para dar una mejor explicación.

Esta tesis es resultado de mi propio trabajo y no incluye nada que sea el resultado de algún trabajo realizado en colaboración, salvo que se indique específicamente en el texto.

Tonantzin Lozano Herrera. Ciudad Universitaria, Cd. Mx., 2023



# Resumen

---

La medición de resistencia de alta resolución es una tarea que requiere de instrumentos especializados y por lo común complejos. Si bien existen técnicas clásicas como el método de las cuatro puntas o instrumentos especializados como los medidores LCR o los analizadores de impedancia; también hay otras técnicas novedosas basadas en arquitecturas híbridas las cuales generalmente están compuesta por un sistema de acoplamiento analógico y una etapa de procesamiento digital. De forma que, este trabajo desarrolla una propuesta de un sistema de medición basado en un amplificador Lock-in para realizar mediciones de resistencia de alta resolución.

El diseño del sistema de medición propuesto comprende en primer lugar una etapa de acoplamiento de la señal de referencia del amplificador LIA, posteriormente una etapa de amplificación y finalmente una etapa de acoplamiento para la digitalización de la señal. Por su parte, el tratamiento digital de la señal acoplada permite hacer el cálculo de la resistencia de prueba. Es así que, se destacan las particularidades de los elementos del sistema diseñado e implementado a partir de: las restricciones observadas en las ecuaciones de diseño, las simulaciones y las mediciones experimentales.

Como parte del trabajo se desarrolla un segundo sistema de medición de resistencia basado en un ADC de alta resolución. Las particularidades de esta arquitectura y los desafíos para su implementación se presentan a lo largo de diferentes casos de configuración; esto se logra principalmente con la implementación de un software muy flexible y una arquitectura robusta.

Finalmente se presentan algunas aplicaciones de diferentes campos de investigación que hacen uso de mediciones de impedancia para caracterizar diversos fenómenos físicos y biológicos.

Así, este trabajo presenta dos sistemas de medición de resistencia de alta resolución basados en arquitecturas simples y en herramientas de software flexibles.

# *Abstract*

---

*High-resolution resistance measurement is a task that often requires specialized and complex instruments. While there are classical techniques such as the 4-wire method or specialized instruments like LCR meters or impedance analyzers; there are also other novel techniques based on hybrid architectures which include both an analogue coupling system and a digital processing stage. Thus, it is proposed a measurement system based on a Lock-in amplifier to perform high-resolution resistance measurements.*

*The design of the measurement system includes a LIA reference signal coupling stage, as well as an amplification stage and a digitization coupling stage. In the other hand, the digital processing of the signal enhances the calculation of the resistance under test. Then, the particularities of the system are pointed from: the observed constraints in the design equations, the simulations, and the experimental measurements.*

*As part of this work, a second resistance measurement system based on a high-resolution ADC is developed. The peculiarities and constrains of the architecture implementation are shown through different use cases and system settings; all this is accomplished thanks to the software flexibility and hardware robustness.*

*Finally, some applications of different fields of research are presented, they make use of impedance measurements to characterize varied physical and biological phenomena.*

*Thus, this work presents two high-resolution resistance measurement systems based on simple architectures and flexible software tools.*

# Índice general

---

<b>Índice de figuras</b>	<b>xv</b>
<b>Índice de tablas</b>	<b>xix</b>
<b>1. Introducción</b>	<b>1</b>
1.1. Antecedentes y Motivación . . . . .	1
1.2. Planteamiento del problema . . . . .	2
1.3. Contribuciones . . . . .	2
1.4. Estructura de la tesis . . . . .	2
<b>2. Marco teórico</b>	<b>5</b>
2.1. Características estáticas de un ADC . . . . .	5
2.1.1. Error de desplazamiento . . . . .	6
2.1.2. Error de ganancia . . . . .	6
2.1.3. No linealidad diferencial . . . . .	6
2.1.4. No linealidad integral . . . . .	7
2.1.5. Resolución . . . . .	8
2.2. Características dinámicas de un ADC . . . . .	9
2.2.1. Frecuencia de muestreo . . . . .	9
2.2.2. Relación señal a ruido . . . . .	9
2.2.3. Distorsión armónica total . . . . .	10
2.2.4. Relación señal a ruido y relación de distorsión . . . . .	10
2.2.5. Número efectivo de bits . . . . .	11

## ÍNDICE GENERAL

---

2.2.6. Rango dinámico . . . . .	11
2.3. Tipos de convertidores analógicos digitales . . . . .	12
2.3.1. Voltaje a Frecuencia . . . . .	12
2.3.2. Paralelo . . . . .	13
2.3.3. Rampa escalera . . . . .	14
2.3.4. Doble rampa de integración . . . . .	16
2.3.5. Aproximaciones sucesivas . . . . .	17
2.3.6. Sigma-Delta . . . . .	18
2.4. Herramientas estadísticas . . . . .	22
2.4.1. Media . . . . .	22
2.4.2. Varianza y Desviación Estándar . . . . .	22
2.4.3. Error Experimental . . . . .	23
2.5. Medición de resistencia con técnica Lock-in . . . . .	24
2.5.1. Amplificador Lock-in . . . . .	25
2.5.2. Amplificador de Instrumentación . . . . .	25
2.5.3. Convertidor Bipolar-Unipolar . . . . .	26
2.5.4. Circuito Schmitt Trigger basado en transistores . . . . .	28
2.6. Protocolos de Comunicación Serial . . . . .	29
2.6.1. Comunicación SPI . . . . .	30
2.6.2. Otros tipos de comunicación Serial . . . . .	31
<b>3. Diseño del experimento</b>	<b>37</b>
3.1. Sistema de medición con la técnica Lock-in y la tarjeta de desarrollo MSP432P401R . . . . .	37
3.1.1. Descripción de la arquitectura . . . . .	38
3.1.2. Análisis del sistema . . . . .	39
3.1.3. Desarrollo de Código en Code Composer . . . . .	51
3.1.4. Simulaciones . . . . .	54
3.2. Sistema de medición con ADC7124-8 y el microcontrolador ESP32 en el IDE de Arduino . . . . .	60
3.2.1. Desarrollo y pruebas del código . . . . .	61

3.3. Sistema de medición para ADC7124-8 utilizando el microcontrolador Raspberry PiPico y MicroPython . . . . .	62
3.3.1. Programación de ADC7124 . . . . .	63
3.4. Sistema de medición para ADC7124-8 utilizando el microcontrolador ESP32 y MicroPython . . . . .	77
3.4.1. Escritura del código . . . . .	77
<b>4. Mediciones y Análisis de Resultados</b>	<b>81</b>
4.1. Resultados del sistema de medición con técnica Lock-in y MSP432 . . . . .	81
4.1.1. Análisis Estadístico . . . . .	83
4.1.2. Arquitectura final con técnica Lock-in y MSP432 . . . . .	87
4.2. Resultados del sistema de medición con ADC7124-8 y Raspberry PiPico	88
4.2.1. Análisis Estadístico . . . . .	89
4.2.2. Arquitectura final del ADC7124-8 con Raspberry PiPico . . . . .	90
4.3. Resultados del sistema de medición con ADC7124-8 y ESP32 . . . . .	91
4.3.1. Análisis Estadístico . . . . .	92
4.3.2. Variación del tiempo de espera entre mediciones . . . . .	93
4.3.3. Arquitectura final del ADC7124-8 con ESP32 . . . . .	95
4.4. Áreas de aplicación . . . . .	96
4.4.1. Bioimpedancia . . . . .	96
4.4.2. Sensores Resistivos de Temperatura RTD . . . . .	102
4.4.3. Medición de resistencia en conductores . . . . .	103
4.5. Discusión . . . . .	104
<b>5. Conclusiones</b>	<b>105</b>
<b>A. Apéndices</b>	<b>109</b>
A.1. Instalación de MicroPython . . . . .	109
A.1.1. Descarga e instalación de librerías para ESP32-32E . . . . .	109
A.1.2. Descarga e instalación de librerías para ADC7124-8 . . . . .	111
A.1.3. Descarga e instalación de la terminal . . . . .	113
A.1.4. Descarga e instalación de MicroPython para Raspberry Pi Pico .	114

## ÍNDICE GENERAL

---

A.1.5. Descarga e instalación de MicroPython para ESP32 . . . . .	114
A.2. Código . . . . .	117
A.2.1. Medición de resistencia con LIA . . . . .	117
A.2.2. Librería para ADC7124-8 . . . . .	120
A.2.3. main.c para ADC7124-8 . . . . .	126
A.3. Manuales y hojas de Especificaciones . . . . .	127
A.4. Mediciones . . . . .	135
A.4.1. Mediciones de simulación en Multisim de la arquitectura LIA sin compuertas Schmitt Trigger . . . . .	135
A.4.2. Mediciones de simulación en Multisim de la arquitectura LIA con compuertas Schmitt Trigger . . . . .	136
A.4.3. Mediciones del convertidor bipolar-unipolar . . . . .	137
A.4.4. Tarjeta MSP432P401R: Prueba del ADC-SAR embebido . . . . .	140
A.4.5. Tarjeta MSP432P401R: medición de voltaje de terminales digi- tales con compuerta Schmitt Trigger . . . . .	141
A.4.6. Amplificador de Instrumentación AD620AN . . . . .	142
A.5. Diagramas de simulaciones . . . . .	144
A.5.1. Simulación de compuertas Schmitt Trigger en Multisim . . . . .	144
A.5.2. Simulación de la arquitectura completa del sistema con LIA en Multisim . . . . .	145
<b>Bibliografía</b>	<b>147</b>



## Índice de figuras

---

2.1. Error de Offset. . . . .	6
2.2. Error de Ganancia. . . . .	7
2.3. Error Diferencial. . . . .	7
2.4. Error Integral. . . . .	8
2.5. Relación Señal a Ruido. . . . .	9
2.6. Distorsión Armónica. . . . .	10
2.7. Rango Dinámico. . . . .	11
2.8. Arquitectura de un convertidor tipo Voltaje a Frecuencia. . . . .	13
2.9. Arquitectura de un convertidor paralelo o <i>flash</i> . . . . .	15
2.10. Arquitectura de un convertidor Rampa Escalera. . . . .	16
2.11. Arquitectura de un convertidor Doble Rampa. . . . .	17
2.12. Arquitectura de un convertidor SAR. . . . .	18
2.13. Arquitectura Sigma-Delta. . . . .	19
2.14. Arquitectura básica de un Modulador Sigma-Delta. . . . .	20
2.15. Error de cuantización ideal. . . . .	21
2.16. Arquitectura del filtro digital y de diezmado. . . . .	21
2.17. LIA arquitectura básica. . . . .	25
2.18. Arquitectura básica de un Amplificador de Instrumentación. . . . .	26
2.19. Señal Bipolar. . . . .	27
2.20. Señal Unipolar. . . . .	27
2.21. Arreglo Bipolar-Unipolar. . . . .	27
2.22. Símbolo para el disparador Schmitt. . . . .	28

## ÍNDICE DE FIGURAS

---

2.23. Arquitectura para un circuito disparador Schmitt basado en Transistores.	29
2.24. Comunicación serial y paralela. . . . .	29
2.25. Diagrama de conexión para la comunicación SPI. . . . .	30
2.26. Diagrama de conexiones para el protocolo I <sup>2</sup> C. . . . .	33
2.27. Arquitectura simplificada del protocolo USB. . . . .	33
2.28. Arquitectura básica para protocolo UART. . . . .	34
3.1. Arquitectura completa. . . . .	39
3.2. Convertidor Bipolar-Unipolar. . . . .	40
3.3. Alimentación del amplificador de instrumentación. . . . .	41
3.4. Estructura interna de bloques A-B (circuito Schmitt-Trigger). . . . .	46
3.5. Operación del disparador Schmitt cuando $I/O_2$ está encendida. . . . .	47
3.6. Simulación de la arquitectura basada en LIA sin el disparador Schmitt para los dos casos de operación. . . . .	55
3.7. Resultados de la simulación en Multisim del convertidor bipolar-unipolar.	58
3.8. Simulación de la arquitectura basada en LIA con disparadores Schmitt para los dos casos de operación. . . . .	59
3.9. Resultados de la simulación-prueba del ADC tipo SAR del microcontrolador MSP432P401R. . . . .	60
3.10. Diagrama de conexión para la comunicación SPI. . . . .	63
3.11. Mapa de bits para registros de filtros. . . . .	64
3.12. Mapa de bits de registro <i>Channel</i> . . . . .	65
3.13. Bits de configuración de registro <i>Channel</i> . . . . .	65
3.14. Conversión de voltaje. . . . .	66
3.15. Sensor RTD con conexión de dos puntas (imagen tomada de la Nota de aplicaciones CN0383). . . . .	69
3.16. ADC7124-8 Múltiples RTDs en configuración de 2 puntas (imagen tomada de la nota de aplicación CN0383). . . . .	72
3.17. Conexiones para la comunicación SPI entre ESP32 y ADC7124-8. . . . .	78
4.1. Resistencias de prueba. . . . .	84
4.2. Casos de medición con ADC7124-8 y Raspberry Pi Pico. . . . .	89

4.3. Arquitectura para medición con el ADC7124-8 y su conexión SPI con el microcontrolador a utilizar. . . . .	91
A.1. Menú <i>File</i> en Arduino. . . . .	110
A.2. Menú de preferencias de Arduino. . . . .	110
A.3. <i>Manager</i> de tarjetas adicionales. . . . .	110
A.4. Menú de herramientas de Arduino. . . . .	111
A.5. Instalación de DFRobot ESP32. . . . .	111
A.6. Menú herramientas Arduino. . . . .	112
A.7. Agregar librería con archivo ZIP a Arduino. . . . .	112
A.8. Interfaz de terminal PuTTY. . . . .	113
A.9. Placa Raspberry Pi Pico. . . . .	114
A.10. Conexión Led en Raspberry Pi Pico. . . . .	115
A.11. Diagrama de simulación de compuerta Schmitt Trigger. . . . .	144
A.12. Diagrama de simulación de arquitectura con LIA. . . . .	145



# Índice de tablas

---

2.1. Modos de operación para el protocolo SPI. . . . .	31
3.1. Valores de diseño del convertidor Bipolar-Unipolar obtenidos con <i>Solver</i> de <i>Excel</i> . . . . .	45
3.2. Valores para el circuito Schmitt Trigger. . . . .	49
3.3. Valores de simulación del circuito Schmitt Trigger en Multisim. . . . .	56
3.4. Valores de simulación del Amplificador de Instrumentación AD620AN en Multisim. . . . .	56
3.5. Prueba del rango de operación del ADC7124-8. . . . .	74
3.6. Modos de operación y Código. . . . .	75
3.7. Configuración de las opciones de filtrado. . . . .	76
4.1. Valores y tolerancias de las resistencias de prueba. . . . .	83
4.2. Promedio de mediciones del LIA con MSP432P401R. . . . .	85
4.3. Error Experimental de mediciones del LIA con MSP432P401R. . . . .	86
4.4. Desviación Estándar de mediciones del LIA con MSP432P401R. . . . .	86
4.5. Tolerancia de las mediciones hechas con el sistema basado en Lock-in. . . . .	87
4.6. Especificaciones del sistema de medición basado en Lock-in. . . . .	88
4.7. Promedio de mediciones del ADC7124-8 con Raspberry PiPico. . . . .	89
4.8. Error experimental de mediciones del ADC7124-8 con Raspberry PiPico. . . . .	90
4.9. Desviación Estándar de mediciones del ADC7124-8 con Raspberry PiPico. . . . .	90
4.10. Promedios de mediciones del ADC7124-8 con ESP32. . . . .	92
4.11. Errores experimentales de mediciones del ADC7124-8 con ESP32. . . . .	92

## ÍNDICE DE TABLAS

---

4.12. Desviación estándar de mediciones del ADC7124-8 con ESP32. . . . .	93
4.13. Tiempo de asentamiento del filtro digital para ADC7124-8. . . . .	94
4.14. RTD_0 para configuraciones de ADC fuera del bucle. . . . .	96
4.15. RTD_1 para configuraciones de ADC fuera del bucle. . . . .	96
4.16. RTD_2 para configuraciones de ADC fuera del bucle. . . . .	97
4.17. RTD_3 para configuraciones de ADC fuera del bucle. . . . .	97
4.18. RTD_4 para configuraciones de ADC fuera del bucle. . . . .	98
4.19. RTD_0 para configuraciones de ADC dentro del bucle. . . . .	98
4.20. RTD_1 para configuraciones de ADC dentro del bucle. . . . .	99
4.21. RTD_2 para configuraciones de ADC dentro del bucle. . . . .	99
4.22. RTD_3 para configuraciones de ADC dentro del bucle. . . . .	100
4.23. RTD_4 para configuraciones de ADC dentro del bucle. . . . .	100
A.1. Mediciones de simulación en Multisim de arquitectura LIA sin compuerta Schmitt Trigger. . . . .	136
A.2. Mediciones de simulación en Multisim de la arquitectura LIA con compuertas Schmitt Trigger. . . . .	137
A.3. Mediciones del convertidor bipolar-unipolar. . . . .	140
A.4. Mediciones de prueba del ADC-SAR del microcontrolador MSP432P401R. . . . .	140
A.5. Mediciones de voltajes digitales antes y después de la compuerta Schmitt Trigger, Caso I. . . . .	141
A.6. Mediciones de voltajes digitales antes y después de la compuerta Schmitt Trigger, Caso II. . . . .	142
A.7. Amplificador de Instrumentación AD620AN. . . . .	143

# Índice de Abreviaturas

---

Abreviatura	Significado
ADC	<i>Analog to Digital Converter</i>
AI	Amplificador de Instrumentación
AINM	Entrada Analógica Negativa
AINP	Entrada Analógica Positiva
CLK	<i>Clock</i>
CS	<i>Chip Select</i>
DAC	<i>Digital to Analog Converter</i>
DNL Error	<i>Differential Non-Linear Error</i>
ENIAC	<i>Electronic Numerical Integrator And Computer</i>
ENOB	<i>Effective number of bits</i>
I <sup>2</sup> C	<i>Inter-Integrated Circuit</i>
IDE	<i>Integrated Development Environment</i>
INL Error	<i>Integral Non-Linear Error</i>
LCR	<i>Inductancia, capacitancia y resistencia</i>
LIA	<i>Lock-in Amplifier</i>
LSB	<i>Least Significant Bit</i>
MISO	Maestro <i>In</i> Esclavo <i>Out</i>
MOSI	Maestro <i>Out</i> Esclavo <i>In</i>
NPN	Negativo-Positivo-Negativo
PE	Polietileno
PMBus	<i>Power Management Bus</i>
PNP	Positivo-Negativo-Positivo
PVC	Policloruro de Vinilo
RMS	<i>Root Mean Square</i>
RTD	<i>Resistance Temperature Detector</i>
SAR	<i>Successive Approximation Register</i>
SCLK	<i>Serial Clock</i>
SDA	<i>Serial Data</i>
SMBus	<i>System Management Bus</i>

## ÍNDICE DE TABLAS

---

Abreviatura	Significado
SNDR o SINAD	<i>Signal to Noise and- Distortion Ratio</i>
SNR	<i>Signal to Noise Ratio</i>
SPI	<i>Serial Peripheral Interface</i>
SPS	<i>Samples per Second</i>
TBJ	<i>Bipolar Junction Transistor</i>
THD	<i>Total Harmonic Distortion</i>
UART	<i>Universal Asynchronous Receiver-Transmitter</i>
USB	<i>Universal Serial Bus</i>
VFC	<i>Voltage to Frequency Converter</i>



# Introducción

---

## 1.1. Antecedentes y Motivación

En el contexto de la ingeniería eléctrica y electrónica el concepto de impedancia real y resistencia se suele interpretar de igual manera comúnmente, sin embargo, tanto en los laboratorios como en las clases se dedica un especial cuidado a su estudio.

Si bien, para medir una resistencia de carbón en un laboratorio de electrónica un estudiante utilizará un multímetro, en un laboratorio de bioimpedancia un científico utilizará instrumentos precisos para obtener estas mediciones. De forma que, aunque parezca trivial el concepto de resistencia, en el contexto de la instrumentación científica se han desarrollado diversas técnicas de medición como el método de los cuatro electrodos, los medidores LCR o analizadores de impedancia, y los analizadores de respuesta en frecuencia, con el objetivo de tener mediciones precisas.

Ahora bien, entendiendo que vivimos en un mundo analógico y gracias a la actual capacidad de cómputo de la mayoría de los microcontroladores, los sistemas de instrumentación científica son híbridos, es decir, están conformados por una sección de acoplamiento analógico y una sección de procesamiento digital. Esto es posible gracias a los convertidores analógicos digitales que permiten hacer la conexión entre el mundo analógico y digital.

Por lo tanto, al entender que medir resistencia va más allá de utilizar un multímetro en un laboratorio de electrónica, en ese trabajo se presenta una propuesta de un sistema de medición basado en un amplificador Lock-in y otro sistema basado en un convertidor analógico digital (ADC: *Analog to Digital Converter*) de alta resolución. Teniendo siempre en mente que la arquitectura debe de ser simple en comparación a la mayoría de los dispositivos comerciales. Así mismo, se presentan diversas aplicaciones de la medición de resistencia basada en Lock-in y cálculo de bioimpedancia.

### 1.2. Planteamiento del problema

Dado que los instrumentos de medición de resistencia de alta resolución suelen ser muy complejos, los usuarios requieren ser especialistas para utilizarlos y entender los diferentes parámetros de configuración. En ese trabajo se busca plantear el diseño de un instrumento de medición de resistencia basado en un amplificador Lock-in para medir resistencia del orden de  $m\Omega$ . Así mismo, se propondrá una arquitectura alterna para medición de resistencia basada en un convertidor de alta resolución.

### 1.3. Contribuciones

Este trabajo presenta dos sistemas de medición de resistencia, uno basado en un amplificador Lock-in y otro basado en un ADC de alta resolución, ambas propuestas se analizan matemáticamente y se implementan para verificar su funcionamiento y detectar sus deficiencias.

Se desarrolla detalladamente el diseño y se proponen las ecuaciones de análisis de los circuitos, lo que permite variar los diferentes parámetros para ajustar la arquitectura a las necesidades de medición.

Finalmente, el trabajo presenta posibles aplicaciones de bioimpedancia y en otras áreas en las que la medición de la resistencia es un parámetro que describe el comportamiento de fenómenos biológicos.

### 1.4. Estructura de la tesis

Esta tesis está dividida en cinco capítulos, el Capítulo 1 corresponde a la introducción del trabajo, el Capítulo 2 presenta todos los conceptos teóricos que serán utilizados a lo largo del proceso de diseño, de prueba y de análisis de los resultados de los dos sistemas de medición propuestos. Esto proporciona una base teórica para comprender el análisis electrónico de las arquitecturas presentados.

El Capítulo 3 presenta el método de diseño. Primero, para el sistema basado en el amplificador Lock-in, aquí se desarrollan a detalle cada una de las etapas del circuito y se obtienen las ecuaciones de diseño además las consideraciones de los diferentes componentes y casos de operación, asimismo se presenta el algoritmo para la digitalización de los datos con la tarjeta de desarrollo utilizada. Para el sistema de medición basado en el ADC de alta resolución se sigue el mismo procedimiento, la propuesta de la arquitectura y el desarrollo del algoritmo de adquisición de datos para los microcontroladores

utilizados.

El Capítulo 4 presenta el análisis estadístico de las mediciones realizadas con ambas arquitecturas para diferentes casos de prueba propuestos. Así mismo, se comenta sobre el desempeño y las deficiencias de cada arquitectura.

Finalmente, el Capítulo 5 es para las conclusiones del trabajo. Aquí se presenta una breve discusión de los resultados y del análisis estadístico del Capítulo 4, también se presentan algunas sugerencias para mejorar el desempeño del sistema de medición basado en amplificador Lock-in.



# Marco teórico

---

En este capítulo se presentarán primeramente las características estáticas y dinámicas de los convertidores analógicos-digitales (ADC: *Analog to Digital Converter*). Posteriormente, se hace una presentación de los diferentes tipos de ADCs, destacando su arquitectura y su principio de funcionamiento, así como algunas características distintivas de los mismos.

En la Sección 2.4 se definen los conceptos básicos de estadística descriptiva usados para el análisis de las mediciones experimentales.

Por su parte, en la Sección 2.5 se describen los diferentes componentes utilizados en la arquitectura propuesta en el experimento de Capítulo 3. Finalmente se presentan diferentes tipos de comunicación serial. Un Convertidor Analógico Digital es un dispositivo que permite convertir valores continuos/analógicos de voltaje a valores discretos/digitales. En principio se muestrea la señal analógica y se reconstruye con un código digital (por ejemplo en código binario).

Sin embargo, ¿cómo saber si la conversión es una representación fiel del valor real? y por lo tanto ¿cómo diferenciar entre un circuito y otro que al parecer hacen lo mismo? Pues bien, existen ciertas características electrónicas que evalúan el desempeño de un ADC, estas se clasifican en estáticas y dinámicas. Para comprender la diferencia entre estas a continuación se presentan primeramente las estáticas y posteriormente las dinámicas.

## 2.1. Características estáticas de un ADC

Las características estáticas de un ADC son aquellas que no varían en función de la señal de entrada, es decir son propias de la arquitectura o de la electrónica de los componentes del ADC, algunas de ellas son el error de desplazamiento, el error de

ganancia, la resolución, entre otras.

### 2.1.1. Error de desplazamiento

El error de offset indica cuan desplazada está el valor de la conversión del valor real, es decir, si por ejemplo para un voltaje  $V_x$  se espera un código 001b y dicho valor digital se obtiene con un voltaje mayor (o menor), entonces el ADC tiene un *offset* o un error de desplazamiento. Una característica de este error es que la diferencia entre el valor real y el resultante de la conversión es constante para todas las conversiones (1). La Fig. 2.1 lo ilustra.

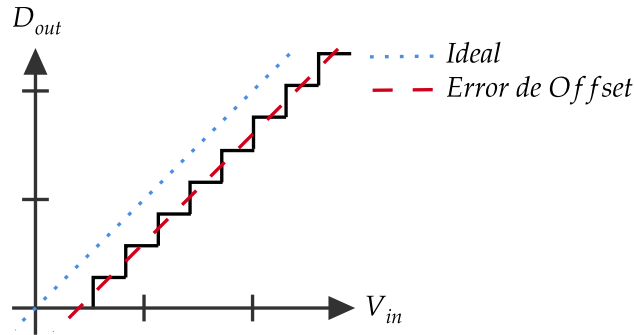


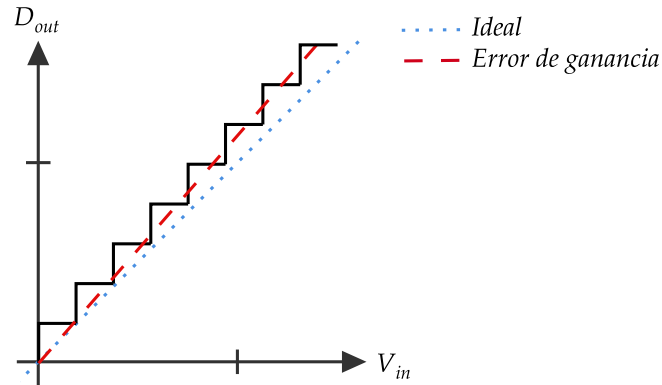
Figura 2.1: Error de Offset.

### 2.1.2. Error de ganancia

El error de ganancia indica la desviación de la pendiente de conversión contra la pendiente ideal (Fig. 2.2). Entonces por ejemplo, si en una prueba se va aumentando constantemente el voltaje a convertir y se registraran todas las conversiones sería posible trazar una recta de conversión experimental, si se compara dicha pendiente con la pendiente ideal (teórica) del ADC, y las rectas no fueran iguales entonces se habla de un error de ganancia (1).

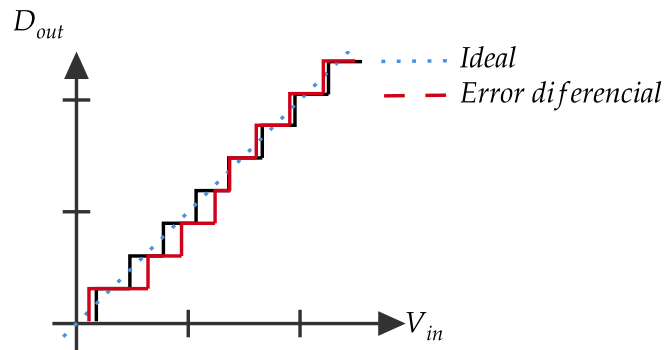
### 2.1.3. No linealidad diferencial

El error de no linealidad diferencial (DNL Error: *Differential Non-Lineal Error*) corresponde a la diferencia de cada escalón con relación a su tamaño ideal que es 1 bit menos significativo (LSB: *Least Significant Bit*) (Fig. 2.3). Es decir, idealmente, el ADC



**Figura 2.2:** Error de Ganancia.

asigna un valor digital a un nivel analógico que está en un rango específico, estos rangos analógicos tienen una magnitud de 1LSB. Existe un error diferencial cuando estos rangos no tienen esa magnitud y son mayores o menores (1).



**Figura 2.3:** Error Diferencial.

#### 2.1.4. No linealidad integral

El error de no linealidad integral (INL Error: *Integral Non-Linear Error*) es el acumulado de los errores diferenciales individuales, es decir considera todos los errores diferenciales presentados en la conversión, Fig. 2.4.

Suponiendo que se ha trazado la recta ideal de conversión del ADC en el mismo plano en el que se han graficado los resultados experimentales de la conversión del ADC. Si el ADC se comporta no linealmente el comportamiento del ADC será aproximado por una curva y no una recta, entonces, la diferencia entre el código resultante de la conversión

## 2. MARCO TEÓRICO

---

experimental para un voltaje  $V_x$  y el código dado para ese voltaje  $V_x$  por la función de transferencia del ADC será el valor del error integral.

El error DNL en conjunto con el INL pueden ser provocados por incompatibilidad entre componentes analógicos, la solución para minimizarlos puede ser la recalibración de los circuitos y la reorganización de las componentes en circuito integrado (1).

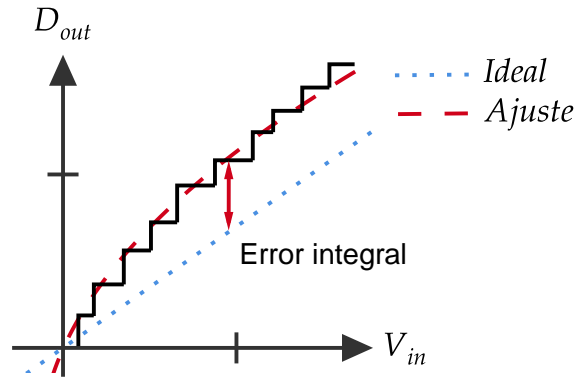


Figura 2.4: Error Integral.

### 2.1.5. Resolución

La resolución está dada en bits y se refiere al número digital de salidas que el convertidor tiene, de forma que si un ADC tiene  $n$  salidas digitales, se dice que el ADC tiene una resolución de  $n$ -bits (2).

Es importante hacer una anotación, actualmente los convertidores no suelen tener salidas paralelas, en su lugar hacen uso de la comunicación serial, entonces, la resolución del ADC se entiende como el tamaño de la palabra, en bits, que será transmitida por el bus de comunicación serial.

Por otra parte, la resolución puede ser dada términos de voltaje. Es decir, cuando se conoce la referencia de voltaje del ADC,  $V_{REF}$ , así como su resolución en bits,  $n$ , se hace el cálculo del voltaje equivalente a 1 LSB, esto representa el voltaje mínimo al que el ADC realizará una conversión.

$$1LSB = \frac{V_{REF}}{2^n} \quad (2.1)$$



## 2.2. Características dinámicas de un ADC

Las características dinámicas, a diferencia de las estáticas, dependen de la señal de entrada especialmente de la frecuencia de esta.

De igual forma, en esta sección se busca ser breve al presentar las definiciones de seis diferentes parámetros dinámicos de un ADC.

### 2.2.1. Frecuencia de muestreo

Dado que un ADC en principio convierte una señal continua de frecuencia  $f$  en una señal digital, es necesario tomar muestras de la señal original para realizar la conversión en un tiempo específico.

La frecuencia con la que las muestras son tomadas es llamada frecuencia de muestreo,  $f_s$ . La importancia de escoger correctamente  $f_s$  recae en la precisión deseada, en principio una alta tasa de muestreo significa tener una mayor cantidad de muestras para reconstruir la señal original, por el contrario una  $f_s$  baja pueden ocasionar la pérdida de información y poca precisión en la conversión (2).

### 2.2.2. Relación señal a ruido

Bajo condiciones estándar de prueba, es decir salida a escala completa y frecuencia de muestreo  $f/2$ . La relación señal a ruido (SNR: *Signal to Noise Ratio*) es el cociente entre la potencia de la señal y la potencia del ruido de la señal (1) (Fig. 2.5). La ecuación

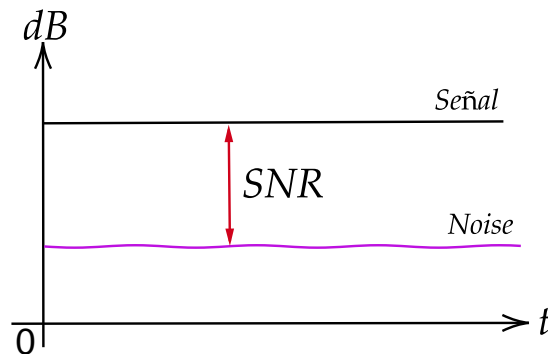


Figura 2.5: Relación Señal a Ruido.

## 2. MARCO TEÓRICO

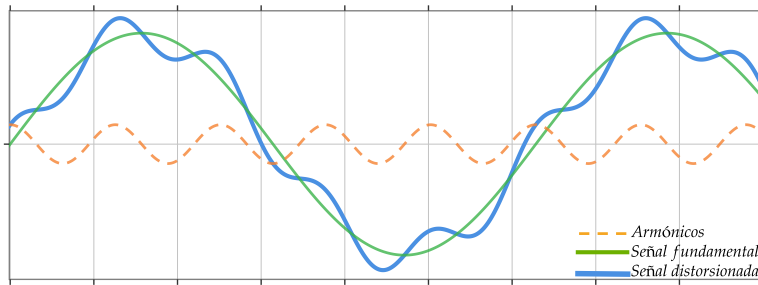
---

2.2 muestra el cálculo de la relación señal a ruido de un ADC ideal de resolución  $N$ .

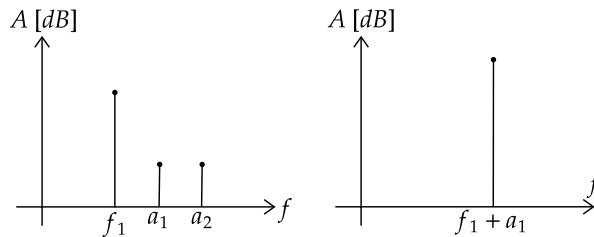
$$SNR = 6.02N + 1.76dB \quad (2.2)$$

### 2.2.3. Distorsión armónica total

La distorsión armónica total (THD: *Total Harmonic Distortion*) es el cociente entre la potencia total de los armónicos de la señal y su potencia total (1) (Fig. 2.6).



(a) Distorsión armónica en el dominio del tiempo.



(b) Distorsión armónica en el dominio de la frecuencia.

Figura 2.6: Distorsión Armónica.

### 2.2.4. Relación señal a ruido y relación de distorsión

La relación señal-ruido y distorsión (SNDR o SINAD: *Signal to Noise and- Distortion Ratio*) es el cociente entre la potencia total de la señal y el total de la potencia de ruido y la distorsión armónica (3).

La SINAD se define en la ecuación 2.3 en donde SNR es la relación señal a ruido y

THD es la distorsión armónica total.

$$SINAD = 20\log_{10}\sqrt{(10^{-SNR/20})^2 + (10^{-THD/20})^2} \quad (2.3)$$

### 2.2.5. Número efectivo de bits

El Número Efectivo de Bits (ENOB: *Effective number of bits*) indica la resolución mínima que el ADC puede alcanzar en condiciones reales de operación, es decir la resolución efectiva del ADC considerando la presencia de ruido o distorsión en el sistema.

La ecuación 2.4 muestra el cálculo del ENOB en términos de la SINAD (1).

$$ENOB = \frac{SINAD - 1.76dB}{6.02} \quad (2.4)$$

### 2.2.6. Rango dinámico

Es el cociente entre la potencia de la señal a escala completa entre la potencia de la señal para la cual la relación señal a ruido (SNR) es 0 dB, Fig. 2.7, con este parámetro se puede saber la máxima y mínima amplitud de la señal que puede ser muestreada con una exactitud dada (3).

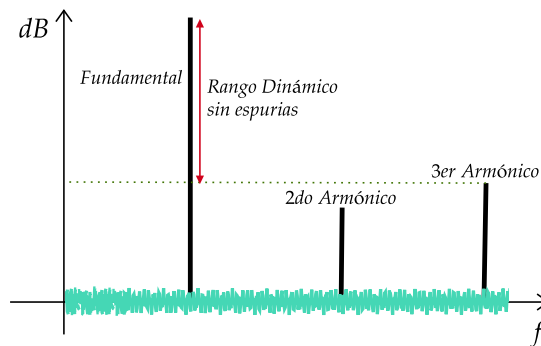


Figura 2.7: Rango Dinámico.

### 2.3. Tipos de convertidores analógicos digitales

El desarrollo de la computación digital ha involucrado una gran cantidad de inventos y propuestas para problemas electrónicos similares. Los primeros ordenadores digitales fueron creados alrededor de 1942 con la computadora ENIAC (*ENIAC: Electronic Numerical Integrator And Computer*)(1) y desde entonces, diversos métodos para digitalizar la información del mundo real han sido propuestos.

En esta sección se describirán brevemente algunas de las arquitecturas más relevantes. Algunas rayan la simplicidad pero cumplen con su propósito eficientemente y otras no tanto. Al final de la sección, se describe brevemente la arquitectura del ADC Sigma-Delta, la cual ha ganado gran popularidad en las últimas décadas gracias a las nuevas técnicas de fabricación de microchips que han permitido la producción masiva de estos componentes y en consecuencia la reducción de su costo.

#### 2.3.1. Voltaje a Frecuencia

El convertidor de voltaje a frecuencia (VFC: *Voltage to Frequency Converter*) está basado en un circuito analógico de carga y descarga de un capacitor, es decir en un integrador analógico, que está acoplado mediante un sistema de control a un contador digital. La Fig. 2.8 muestra la arquitectura básica de este tipo de ADC.

Para realizar una conversión, el circuito VFC recibirá un voltaje de entrada, el circuito analógico inicia a cargar el capacitor, y simultáneamente el sistema de control ha iniciado una ventana de tiempo ya predeterminada para la conversión.

El tiempo de conversión corre mientras el voltaje a la salida del circuito integrador va incrementando hasta ser igual al voltaje de referencia del sistema de control, aquí el comparador genera un pulso que cambia el estado de un monoestable y reinicia el circuito integrador, descargando el capacitor, la descarga del capacitor se espera que sea instantánea, de forma que el tiempo de descarga es prácticamente nulo en comparación con el tiempo de carga.

Cabe mencionar que el tiempo que tarde el circuito integrador en llegar al voltaje de referencia depende de la magnitud del voltaje de entrada, de forma que si el voltaje es grande el capacitor se cargará rápido, si el voltaje es pequeño, cargar el capacitor tomará más tiempo.

El proceso de carga y descarga del capacitor provoca un tren de pulsos en la salida del comparador, este tren de pulsos es cuantificado por un contador, de forma que la cuenta es proporcional a la frecuencia, es decir, para cuando el voltaje de entrada es alto, el capacitor se carga rápidamente, y viceversa, si el voltaje de entrada es pequeño, el capacitor de carga lentamente.

Ahora bien, dentro de una misma ventana tiempo, el capacitor se cargará y descargará más veces si el voltaje es grande o menos veces si el voltaje es pequeño ya que la ventana de tiempo es fija, de forma que en el primer caso la cuenta de pulsos será mayor que en el segundo y por lo tanto el valor de conversión también será mayor en el primer caso, cuando el voltaje es mayor.

Es importante considerar que este tipo de convertidor al trabajar con frecuencia, la conversión mínima posible es de un uno lógico, ya que es necesario por lo menos un pulso del comparador.

Además, la frecuencia máxima o el número de cargas y descargas del capacitor está limitada por el contador, si la frecuencia es mayor que la cuenta máxima del contador este se desbordará y por ende entregará un valor de conversión incorrecto.

Adicionalmente, unas de las principales características de este tipo de ADC es que integra el ruido y es de bajo consumo energético. Suele ser barato por no contar con muchos componentes. Además tiene un comportamiento muy lineal (2).

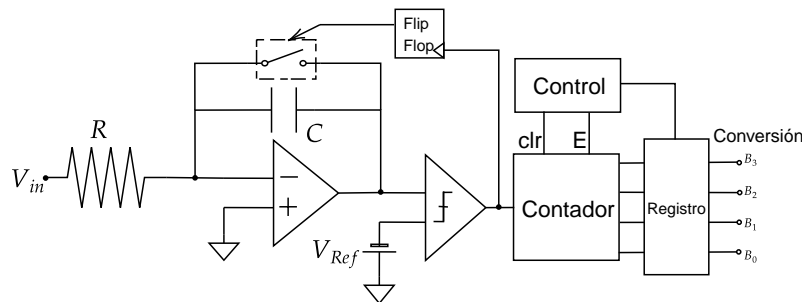


Figura 2.8: Arquitectura de un convertidor tipo Voltaje a Frecuencia.

### 2.3.2. Paralelo

El tipo de convertidor paralelo también llamado *Flash*, Fig. 2.9, es un circuito basado en una configuración de comparadores, resistencias y un codificador de prioridad. Por su arquitectura, si se quiere construir un convertidor de  $n$  bits, serán necesarios  $2^N - 1$  comparadores y múltiples resistencias para obtener los diferentes niveles de voltaje.

En principio cada comparador está configurado para un voltaje de referencia específico, en tanto que la diferencia entre los voltajes de referencia y los comparadores es de 1 LSB.

Para realizar una conversión, el voltaje de entrada llega simultáneamente a todos los comparadores, si este voltaje es mayor que el voltaje de referencia del comparador entonces a la salida habrá un 1, suponiendo que el voltaje de entrada no es el voltaje

## 2. MARCO TEÓRICO

---

a escala completa, es decir el voltaje máximo que puede convertir el ADC, entonces en algún nivel del arreglo de comparadores el voltaje de entrada será menor que la referencia, entonces a partir de este punto y en forma ascendente, los comparadores entregarán un 0 digital.

Con la combinación de unos y ceros el codificador de prioridad entrega un valor digital equivalente al voltaje de entrada.

Una buena analogía para este tipo de convertidor es un termómetro de mercurio, dependiendo de la magnitud del voltaje a convertir será el número de 1s o 0s que se obtendrán en el arreglo de comparadores.

Este tipo de convertidor es llamado *flash* por la velocidad de conversión, dado que el voltaje de entrada es recibido por todos los comparadores simultáneamente, el tiempo de conversión solo depende del tiempo de propagación del comparador y del tiempo que le tome al decodificador hacer la conversión.

Sin embargo, una de las desventajas de este tipo de circuito es la cantidad de componentes que utiliza, lo que deriva en una alta demanda de potencia y poca portabilidad. Por lo tanto, un convertidor *flash* solo es práctico si se utiliza para bajas resoluciones y cuando un bajo consumo de potencia no sea un requisito de diseño (2).

### 2.3.3. Rampa escalera

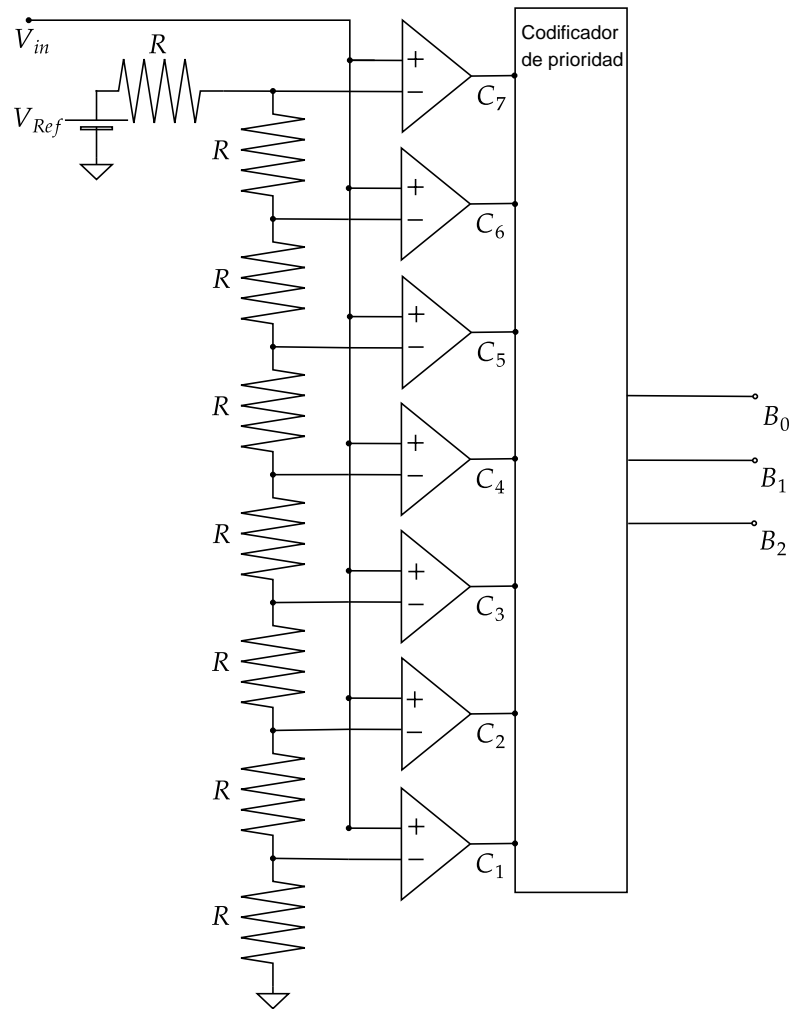
Este convertidor está compuesto por un comparador, un sistema de control, un contador y un convertidor digital analógico (DAC: *Digital to Analog Converter*), Fig. 2.10.

Al iniciar la conversión el voltaje de entrada es recibido en una terminal del comparador, esto da inicio a la cuenta digital que comienza desde cero en el contador, esta cuenta es entonces convertida por el DAC a un voltaje analógico equivalente que es comparado con el voltaje de entrada, si el voltaje del DAC es menor al de la entrada entonces la cuenta sigue incrementando, cuando el voltaje del DAC es mayor al voltaje de entrada entonces el sistema de control detiene la cuenta digital y entrega el valor de la conversión.

Este tipo de convertidor es llamado entonces rampa escalera porque en cada cuenta el voltaje del DAC va incrementando un LSB por iteración hasta alcanzar el voltaje de entrada.

El convertidor rampa escalera es lento, ya que el tiempo de conversión depende de la magnitud del voltaje de entrada, porque el ADC tiene que hacer tantas comparaciones como sea necesario para terminar una conversión.

Suponiendo el peor de los casos, cuando el ADC va a convertir el voltaje máximo posible, si entonces el ADC tiene una resolución de 8 bits, la conversión tardará  $2^8$  ciclos de reloj y el circuito tendrá que hacer  $2^8$  comparaciones. Por el contrario, si el voltaje de entrada es pequeño entonces el ADC hará rápido la conversión ya que en

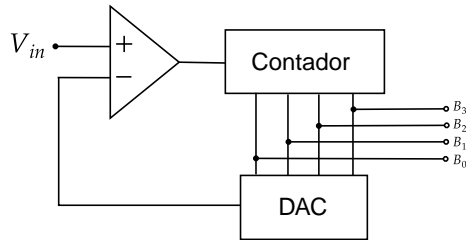


**Figura 2.9:** Arquitectura de un convertidor paralelo o *flash*.

pocas iteraciones el comparador detendrá la cuenta.

Adicionalmente, este tipo de convertidor es susceptible al ruido y a la interferencia, de señales de forma que si por alguna razón se detecta un pico de voltaje, este podría erróneamente cambiar el estado del comparador y detener la conversión antes de tiempo.

Por otra parte, algunas ventajas son la simplicidad y estabilidad del circuito, los componentes que definen el desempeño del ADC son el comparador y el contador que a su vez depende de su reloj) (4).



**Figura 2.10:** Arquitectura de un convertidor Rampa Escalera.

### 2.3.4. Doble rampa de integración

Este tipo de convertidor está formado por una parte analógica y otra digital, la primera sección comprende un integrador y un comparador los cuales se acoplan a un contador digital con un circuito de control que va a manejar el inicio y fin de conversión así como el cambio del voltaje de integración, Fig. 2.11.

Para operar, se deben considerar dos etapas, en la primera se hace la integración del voltaje de entrada y en la segunda la integración del voltaje de referencia.

Para iniciar, el circuito integrador recibe en su entrada el voltaje a convertir, entonces el capacitor del circuito integrador se inicia a cargar, simultáneamente se inicia una cuenta digital.

Dado que en esta primera etapa de integración el comparador no cambia de estado, ya que la salida del integrador es menor que tierra, el contador se desborda porque el sistema de control nunca detiene la cuenta, entonces el circuito cambia el voltaje de integración e inicia a integrar el voltaje de referencia en vez del voltaje de entrada.

La polaridad del voltaje de referencia es contraria al voltaje de entrada entonces el capacitor en vez de cargarse se inicia a descargar, además en esta segunda etapa de integración se ha reiniciado el contador digital.

De forma que mientras el capacitor se está descargando, la cuenta digital está corriendo, entonces, cuando el comparador detecta que se ha descargado el capacitor, el sistema de control detiene la cuenta digital.

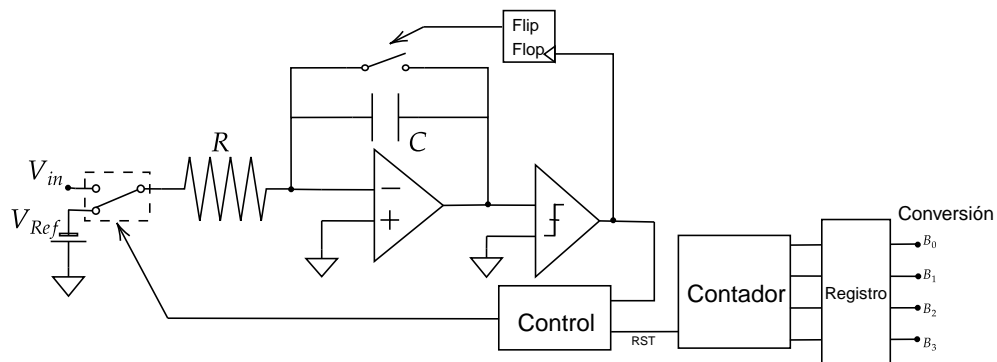
La lógica del funcionamiento de este circuito se basa en la pendiente de integración del voltaje de entrada y del voltaje de referencia, ya que para todos los voltajes de entrada el contador tardará siempre en la primera etapa de integración un tiempo  $t$  constante en llegar a la cuenta máxima y desbordarse. Pero la pendiente de carga dependerá de la magnitud del voltaje de entrada, de forma que el capacitor siempre se cargará durante un tiempo  $t$  pero no a la misma razón y por lo tanto no a la misma capacidad.

La pendiente de integración del voltaje de referencia será constante en todos los casos de



conversión, a diferencia de la pendiente de carga del voltaje de entrada que depende este mismo. De forma que el capacitor tardará más tiempo o menos tiempo en descargarse en función de la magnitud del voltaje de entrada, en consecuencia la cuenta de conversión será mayor o menor según el voltaje de entrada.

Una de las principales ventajas de este tipo de convertidores es la precisión de conversión ya que es independiente tanto de la capacitancia y la frecuencia del reloj, porque ambas variables afectan tanto la pendiente de integración del voltaje de entrada así como la pendiente de integración del voltaje de referencia (2).



**Figura 2.11:** Arquitectura de un convertidor Doble Rampa.

### 2.3.5. Aproximaciones sucesivas

El ADC de aproximaciones sucesiva compara el voltaje de entrada con la conversión digital-analógica de una aproximación digital de la cuenta de conversión. A diferencia del convertidor rampa-escalera, la cuenta no es lineal, es decir, en vez de tener un contador digital que inicia desde cero existe un registro de aproximaciones sucesivas (SAR: *Successive Approximation Register*) que cambia bit por bit la cuenta, el diagrama de bloques de la Fig. 2.12 ilustra la arquitectura.

Al comenzar la conversión, el registro de aproximaciones sucesivas colocará un 1 digital en el bit más significativo de la cuenta y los demás bits en cero, este valor digital es convertido a un voltaje equivalente con el DAC y es comparado con el voltaje de entrada, si el comparador determina que el voltaje de entrada es aún mayor que el valor digital propuesto, entonces se agregará un uno al segundo bit más significativo, posteriormente se realizará la nuevamente la conversión digital-analógica con el DAC y la comparación con el voltaje de entrada, si resultara que el voltaje de entrada es menor que el voltaje analógico de la conversión del DAC entonces se cambia el 1 por un 0 y el siguiente bit más significativo se cambia a 1, este procedimiento se repite hasta

## 2. MARCO TEÓRICO

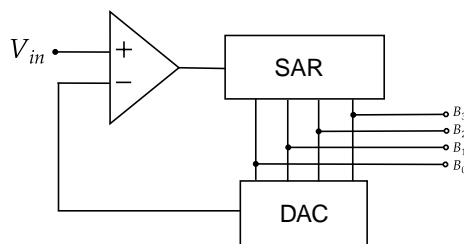
---

acabar con todos los bits de conversión.

De forma que al final de la conversión la combinación de 1 s y 0 s es el resultado. Si se considera entonces un convertidor de resolución N, la conversión de un voltaje necesitara N iteraciones.

Otra forma de entender el principio de funcionamiento del Registro de Aproximaciones Sucesivas es haciendo una analogía con una báscula de pesos o balanza, primero se coloca la pesa más grande y si es más grande que el objeto a pesar se cambia por una más pequeña, si esta resulta ser más pequeña que el objeto, se coloca otra pesa más pequeña que la anterior o la que sigue en magnitud y así sucesivamente hasta terminar de probar todas las pesas posibles.

Una ventaja importante de este tipo de convertidores es su velocidad, ya que solo le tomara un tiempo t hacer todas las comparaciones en comparación con el tiempo de conversión de convertidor rampa que depende de la magnitud del voltaje de entrada (2).



**Figura 2.12:** Arquitectura de un convertidor SAR.

### 2.3.6. Sigma-Delta

La clase de convertidores analógicos-digitales de tipo Sigma-Delta fueron desarrollados desde los años 80 (incluso antes si se habla de los conceptos básicos) sin embargo ha sido hasta las últimas décadas que las nuevas técnicas de fabricación han permitido su producción en escala.

Los elementos básicos de este tipo de convertidor son sencillos componentes y circuitos tales como:

- un comparador,
- una referencia de voltaje,
- un interruptor,

- circuitos integradores, y
- circuitos sumadores, además de
- un filtro paso bajas

Para la comprensión de su funcionamiento es necesario considerar algunos conceptos básicos de electrónica como son:

- sobremuestreo,
- desplazamiento del error de cuantización,
- filtrado digital,
- y diezmado o *decimation* en inglés

Para comenzar, la arquitectura básica del ADC Sigma-Delta, Fig. 2.13, está compuesta por un Modulador Sigma Delta y un filtro digital de diezmado. En conjunto, ambos elementos transforman la señal de entrada en una señal de pulsos digitales que es filtrada y reconstruida para entregar una señal digitalizada.



**Figura 2.13:** Arquitectura Sigma-Delta.

Entonces, el proceso de conversión inicia cuando una señal de entrada llega al modulador Sigma Delta que en principio está formado por un amplificador diferencial, un comparador, un circuito integrador y un DAC, Fig. 2.14.

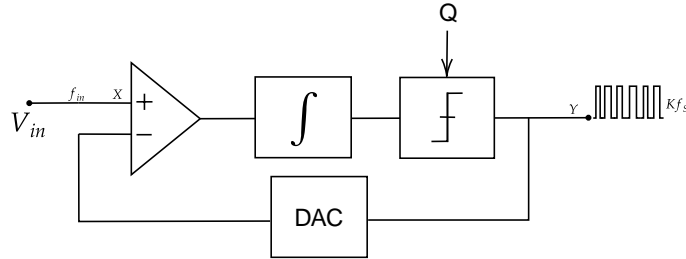
El modulador es en términos simples un amplificador diferencial, por lo tanto tiene un lazo cerrado de realimentación negativa. La señal de entrada es integrada y después comparada con un voltaje de referencia lo que genera un pulso digital, el cual a su vez es convertido a un valor analógico por el DAC de un 1 bit.

La señal de entrada ahora es comparada por el amplificador diferencial con el valor analógico del DAC, nuevamente la diferencia es integrada y posteriormente comparada con la referencia que entrega un pulso digital a la salida, el cual nuevamente es convertido por el DAC y así sucesivamente.

En este punto de la conversión, la señal analógica es ahora una serie de pulsos digitales. Considerando el teorema de muestreo de Nyquist y para evitar la pérdida de información, la frecuencia de la señal digitalizada debe de ser por lo menos 2 veces la frecuencia de la señal de entrada.

## 2. MARCO TEÓRICO

---



**Figura 2.14:** Arquitectura básica de un Modulador Sigma-Delta.

De acuerdo con la arquitectura del modulador, el componente responsable del muestreo de la señal es el comparador, de forma que, es este el que sobre muestrea la señal de entrada. Sin embargo, no es lo único que ha sucedido, también se ha distribuido el error de cuantización en un ancho de banda mayor lo que a su vez mejora la resolución del ADC. Otro fenómeno interesante en esta etapa el desplazamiento del error de cuantización gracias al circuito integrador.

A continuación se explican ambos fenómenos.

Primero, la distribución del error de cuantización,  $e(t)$ , por sobre muestreo se entiende si se considera un ADC ideal que tiene un error de cuantización de  $\pm\frac{1}{2}$  LSB y que por lo tanto puede ser representado por una función de diente de sierra que tiene por ecuación

$$e(t) = mt; -\frac{a}{2m} < t < \frac{a}{2m}, \quad (2.5)$$

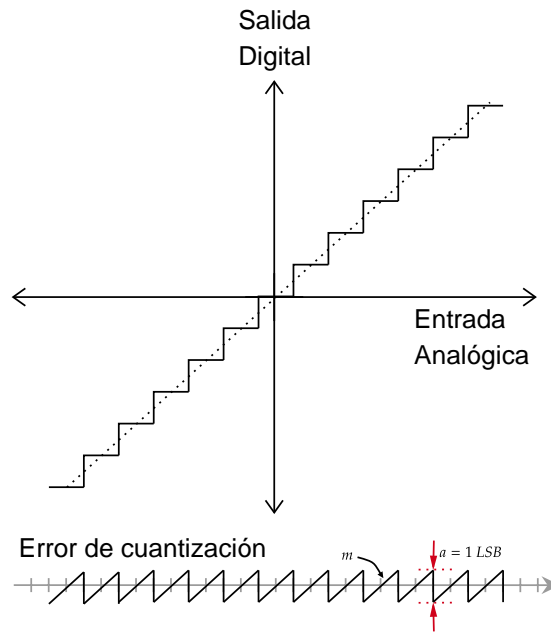
donde  $m$  es la pendiente del diente de sierra y  $a$  es la amplitud que es igual a 1 LSB como se observa en la Fig. 2.15.

Si se obtiene el valor RMS (*Root Mean Square*) del error de cuantización descrito por la siguiente ecuación

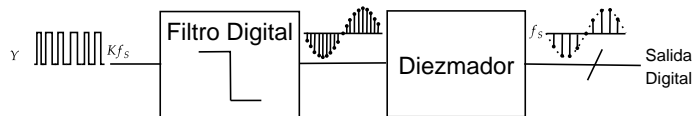
$$e(t)_{RMS} = \sqrt{\frac{1}{\frac{a}{2m} - \frac{-a}{2m}} \int_{-\frac{a}{2m}}^{\frac{a}{2m}} (mt)^2 dt}, \quad (2.6)$$

se obtiene lo siguiente de resolver la integral del valor RMS, de forma que el error de cuantización se puede expresar como

$$e(t)_{RMS} = \frac{a}{\sqrt{12}}. \quad (2.7)$$



**Figura 2.15:** Error de cuantización ideal.



**Figura 2.16:** Arquitectura del filtro digital y de diezmado.

Ahora bien, este error de cuantización está distribuido uniformemente en la banda de Nyquist, es decir en  $f_s/2$ , donde  $f_s$  es la frecuencia de muestreo, dado que el error de cuantización es independiente de la frecuencia de muestreo, si se aumenta la frecuencia de muestreo, es decir se sobre muestrea la señal en un factor  $k$ , el error de cuantización estará distribuido en un ancho de banda mayor. Entonces, el factor por el que se decida incrementar la frecuencia de muestreo  $f_s$  es entonces la relación de sobre muestreo  $k$ .

Por otra parte, el segundo fenómeno interesante es el desplazamiento del error de cuantización a altas frecuencias debido a que el modulador sigma delta integra la señal como lo muestra el diagrama a bloques de la Fig. 2.16 (2).

### 2.4. Herramientas estadísticas

Los datos obtenidos en los experimentos de este trabajo fueron interpretados mediante un breve análisis estadístico para validar las mediciones realizadas. A continuación se presentan algunos de los conceptos y métodos básicos utilizados.

Dado que los datos a analizar son obtenidos experimentalmente, se seguirá un análisis estadístico descriptivo y no uno inferencial.

Al realizar un experimento y obtener las mediciones, las tablas llenas de datos no siempre muestran todo el escenario ni los resultados, por ello se utilizan diferentes herramientas matemáticas tanto gráficas como numéricas. Para el primer tipo se tienen histogramas, diagramas de dispersión, diagramas de barras, polígonos de frecuencia acumulada, diagramas de caja, por mencionar algunos. Por otra parte, para el análisis numérico se pueden recurrir a medidas de tendencia central como con la media, mediana y moda, o a medidas de dispersión o variabilidad como son la varianza, el rango y la desviación estándar. A continuación se describen brevemente las definiciones de algunas mediciones utilizadas a lo largo del trabajo para analizar la información obtenida.

#### 2.4.1. Media

También conocida como promedio,  $\bar{x}$ , es calculada a partir de un conjunto de datos  $(x_1, x_2, \dots, x_n)$  o una muestra, la suma de todas las observaciones se divide entre la cantidad de datos considerados,  $n$ , como se expresa a continuación (5).

$$\bar{x} = \frac{x_1 + x_2 + \dots + x_n}{n} \quad (2.8)$$

#### 2.4.2. Varianza y Desviación Estándar

La varianza y la desviación estándar son dos medidas interrelacionadas entre sí. Para comenzar, se puede considerar la desviación o diferencia de los datos respecto a la media, dependiendo del dato o medición la desviación puede resultar positiva o negativa, sin embargo, esto solo representa individualmente la dispersión cada uno de los datos. Por lo tanto, para analizar la dispersión del conjunto de datos es necesario considerar un ajuste en cada una de las desviaciones de la media. Para esto, se utiliza el cuadrado de la diferencia entre cada dato  $(x_i)$  y la media  $(\bar{x})$ , la suma de estas diferencias es dividida entre el número de datos  $n$ . Esta medida es llamada varianza y dado que fue calculada con los cuadrados de las diferencias, la raíz de la varianza se define como desviación

estándar  $\sigma$  (5).

$$\sigma = \sqrt{\frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n}} \quad (2.9)$$

### 2.4.3. Error Experimental

El error en las mediciones experimentales puede tener diversas fuentes, sin embargo, al analizar o diseñar un instrumento, la precisión de este dependerá entonces de la diferencia entre el valor real o verdadero y el valor medido, por lo tanto el error de medición está asociado directamente con la precisión del instrumento (6).

De forma general se definen dos tipos de error, el error sistemático y el error aleatorio.

El error sistemático puede tener diversas causas, pero la principal es utilizar el instrumento fuera del rango de calibración o cualquier otra causa que cambie de alguna forma la calibración del sensor, para evitarlo, el mantenimiento preventivo y la revisión periódica de los instrumentos y sensores es indispensable. Otra causa puede ser que el propio sistema de medición altere a la sustancia o variable a medir, una última causa puede ser el observador que en su caso lea incorrectamente la lectura del instrumento.

Por otra parte, el error aleatorio es causado por ruido que estadísticamente está distribuido normalmente.

Dado que las mediciones en cualquier experimento están sometidas a tener alguna clase de error, es útil medir la diferencia con respecto al valor real, para valorar la calidad de las mediciones. Una forma simple de evaluar esto es el error absoluto tiene como unidad la de la variable de interés.

$$\epsilon = |x_{real} - x_i|, \quad (2.10)$$

en donde  $x_{real}$  es el valor de referencia y  $x_i$  es la medición tomada.

Sin embargo, habrá ocasiones en las que el error absoluto no proporcione suficiente información para determinar si una medición es aceptable, pensando por ejemplo que 2 cm pueden ser despreciables si se miden longitudes en términos de km ya que un centímetro representa una milésima de un kilómetro.

A partir de esto, calcular el porcentaje en el que difiere la medición con el valor real resulta ser una mejor medida para decidir si una medición tiene cierto grado de aceptabilidad.

Para esto, se realiza una regla de tres simple, en la que se quiere saber que porcentaje

## 2. MARCO TEÓRICO

---

del error de la medición representa el error absoluto. Por lo tanto, si se considera que el 100 % del valor de la medición es el valor teórico, el error absoluto representa una pequeña parte del todo, y es ese porcentaje es el error experimental, según la regla de tres simple,

$$\begin{aligned}x_{real} &\longrightarrow 100 \% \\ \epsilon &\longrightarrow \epsilon_{exp} \% \end{aligned}$$

En la ecuación 2.10 el error absoluto se define como  $\epsilon$  entonces si se quiere conocer el porcentaje del error

$$\epsilon_{exp} = \frac{\epsilon \times 100 \%}{x_{real}}$$

sustituyendo

$$\epsilon_{exp} = \frac{|x_{real} - x_i|}{x_{real}} \times 100 \% \quad (2.11)$$

por lo tanto, el error experimental es adimensional y representa un porcentaje.

### 2.5. Medición de resistencia con técnica Lock-in

La medición de resistencia puede parecer trivial si se piensa en un mutímetro digital y un resistor de carbón, comúnmente usados en los laboratorios de electrónica. Sin embargo, dependiendo de la aplicación, se puede escoger una técnica diferente de medición de resistencia. Es entonces, como la instrumentación electrónica ha propuesto diversas técnicas, entre las más comunes de medición de impedancia se encuentran: el método de los cuatro electrodos, los medidores LCR o analizadores de impedancia, y los analizadores de respuesta en frecuencia (7).

Sin embargo, en la mayoría de los casos se requieren equipos robustos y sofisticados, que no son útiles para aplicaciones en donde la portabilidad es un requerimiento, como es el caso de los dispositivos personalizados (*wearables*).

En este trabajo se desarrolla una propuesta basada en la técnica Lock-in, que en conjunto con otras etapas de procesamiento de la señal analógica y en conjunto con el procesamiento digital, permiten hacer mediciones precisas de resistencia. En conjunto, la arquitectura final incluye un ADC, un convertidor bipolar-unipolar, un par de disparadores Schmitt y un arreglo analógico basado en la técnica Lock-in.

En esta sección, se describen estos componentes con el fin de comprender su funcionalidad estos principios serán retomados en el Capítulo 3, donde se detalla el funcionamiento de la arquitectura propuesta.



### 2.5.1. Amplificador Lock-in

Un amplificador lock-in (LIA: *Lock-in Amplifier*) hace uso de varias técnicas de acoplamiento de señales para poder analizarlas en ambientes de alto ruido. Esto permite conservar la señal de interés lo mas limpia posible.

La arquitectura básica de un LIA, Fig. 2.17, contempla una señal de referencia con amplitud y frecuencia conocidas, un amplificador de bajo ruido, un multiplicador de señales y una etapa de filtrado. En conjunto, estos elementos le dan al LIA la capacidad de extraer señales inmersas en ruido.

Primeramente, el circuito multiplica la señal de entrada con la referencia interna, posteriormente, se compara la señal de entrada con una referencia interna, descartando las componentes de ruido, finalmente se amplifica la señal lo que permite manejar señales de muy baja amplitud (8).

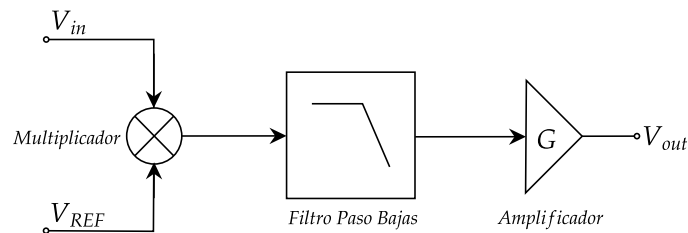


Figura 2.17: LIA arquitectura básica.

### 2.5.2. Amplificador de Instrumentación

Un amplificador de instrumentación (AI) es un circuito integrado de propósito especial ocupado principalmente para aplicaciones de medición. La salida del circuito depende de una entrada diferencial que es sensada en las terminales inversora y no inversora del circuito, dicha diferencia de potencial es multiplicada por un factor de escala que depende por lo común de una resistencia de ajuste externa (9).

Este circuito es usado principalmente para amplificar señales pequeñas. Para ello el circuito suele tener las siguientes características.

- Baja impedancia en la salida: permite que la señal de salida pueda ser tomada como una fuente de voltaje ideal.
- Alta impedancia en la entrada: permite analizar la señal de entrada sin atenuarla.
- Alta relación de rechazo al modo común : permite discriminar o anular señales de la misma amplitud, con ello se asegura que la diferencia de las señales sea exacta.

## 2. MARCO TEÓRICO

---

El amplificador de instrumentación está basado en tres amplificadores operaciones, dos de ellos configurados como no inversores y un tercero configurado como amplificador diferencial, la Fig. 2.18 ilustra las conexiones. Los dos amplificadores no inversores proporcionan ganancia al circuito y aseguran la alta impedancia de entrada, por su parte el amplificador diferencial de ganancia unitaria realiza la operación de diferencia entre las señales (10).

Por lo tanto, la salida del AI está en función de la ganancia  $G$  que depende de  $R_G$  y de la diferencia de voltaje en las terminales del amplificador, ecuación 2.12.

$$V_{out} = G(v_{in}^+ - v_{in}^-) \quad (2.12)$$

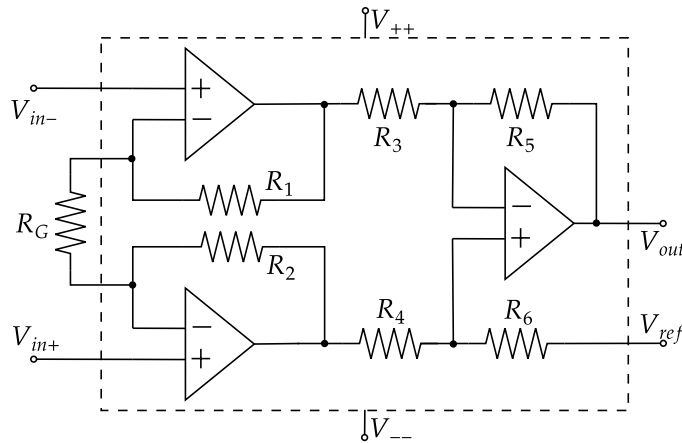
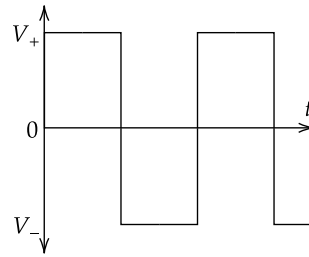


Figura 2.18: Arquitectura básica de un Amplificador de Instrumentación.

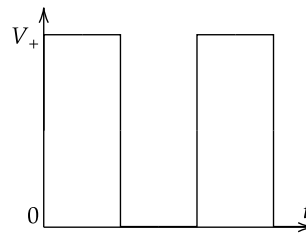
### 2.5.3. Convertidor Bipolar-Unipolar

Una señal bipolar, Fig. 2.19, es aquella que puede tener valores positivos, negativos e iguales a cero. Por su parte en una señal unipolar, Fig. 2.20, todos los valores posibles de la señal se encuentran sobre o debajo de un valor de voltaje de referencia (11).

Por lo tanto un convertidor bipolar-unipolar es aquel que transforma una señal bipolar que puede tomar valores positivos y negativos (además de cero) a una unipolar que solo puede tener valores mayores a cero (positivos) o menores a cero (negativos), en este trabajo es de interés solo tener voltajes positivos de las señales por lo que se considerará un convertidor bipolar-unipolar positivo.

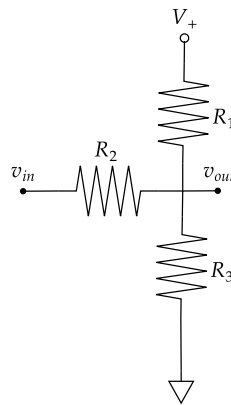


**Figura 2.19:** Señal Bipolar.



**Figura 2.20:** Señal Unipolar.

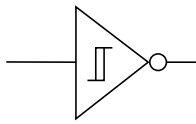
El diseño propuesto por Bengtsson (8) comprende adicionalmente un arreglo de resistencias, Fig. 2.21. El circuito es analizado en el siguiente capítulo para obtener las ecuaciones de diseño.



**Figura 2.21:** Arreglo Bipolar-Unipolar.

### 2.5.4. Circuito Schmitt Trigger basado en transistores

En principio el disparador Schmitt es un comparador de voltaje. Dicho circuito recibe un voltaje de entrada, el cual es comparado con una referencia. Es importante señalar que este tipo de circuito no es un circuito amplificador sin embargo su estructura está basada en amplificadores operacionales. Los comparadores de voltaje son comúnmente usados en generadores de funciones o interruptores. El símbolo eléctrico de este tipo de disparadores se muestra en la Fig. 2.22.



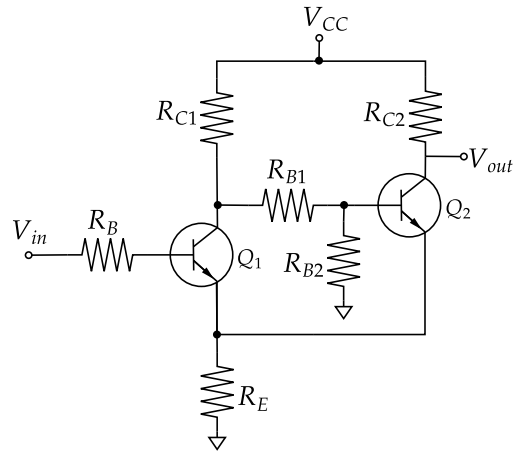
**Figura 2.22:** Símbolo para el disparador Schmitt.

El funcionamiento básico de un disparador Schmitt consiste en recibir un voltaje de entrada que supere un valor de umbral o punto de disparo que saturará al circuito lo que a su vez se refleja como el voltaje máximo de salida, por otra parte si el voltaje de entrada está por debajo del valor umbral el dispositivo cambia a un segundo estado de saturación que lo lleva al voltaje mínimo (10).

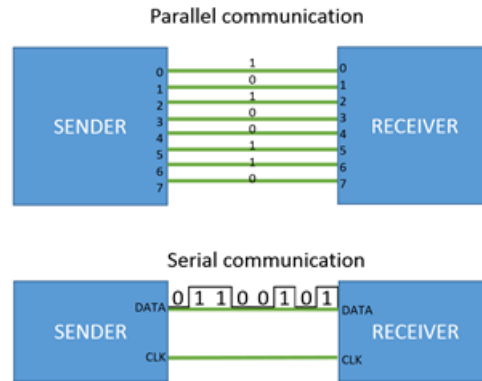
Existen diferentes propuestas para los circuitos comparadores, algunas configuraciones están basadas en amplificadores de transconductancia, otros en amplificadores operacionales, sin embargo existen diseños más simples basados en transistores.

Una circuito disparador Schmitt basado en transistores es una propuesta económica en comparación a un circuito integrado de propósito específico como un comparador, si la aplicación no requiere una rápida conmutación entre los estados o no se requieren definir umbrales muy precisos la opción basada en transistores es muy conveniente.

La configuración básica está dada por la Fig. 2.23, en donde si el voltaje de entrada es cero el primer transistor esta apagado y el transistor 2 encendido, por lo que el voltaje en la resistencia común en el emisor es mayor a cero. Cuando el voltaje de entrada aumenta,  $Q_1$  cambia de Off a conducir, si continúa aumentando la entrada llegará un punto en el que la corriente de base del  $Q_2$  disminuirá hasta que deje de estar saturado lo que a su vez ocasiona que disminuya el voltaje en la resistencia del emisor. El diseño del circuito se describe más adelante en el Capítulo 3 (12).



**Figura 2.23:** Arquitectura para un circuito disparador Schmitt basado en Transistores.



**Figura 2.24:** Comunicación serial y paralela.

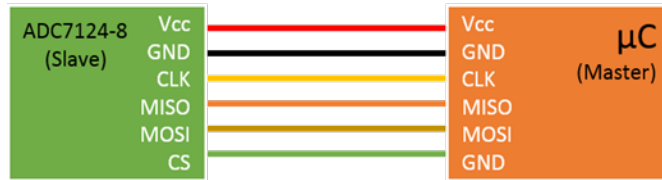
## 2.6. Protocolos de Comunicación Serial

Los dispositivos como las personas necesitan diferentes canales de comunicación estructurados para intercambiar instrucciones e información entre ellos, a lo largo del desarrollo del hardware ha surgido dos acercamientos en la transmisión de la información, la comunicación en paralelo y la comunicación serial. Por su parte, la primera se caracteriza por usar un cable por bit, mientras que la segunda envía información a otro dispositivo poco a poco por un único cable, la Fig. 2.24 lo ilustra (13).

Para poder hacer la transmisión de la información en una comunicación serial que en principio solo cuenta con una línea de comunicación, el mensaje es dividido en pequeños paquetes y después enviado por partes al destinatario bit por bit. El receptor

## 2. MARCO TEÓRICO

---



**Figura 2.25:** Diagrama de conexión para la comunicación SPI.

por su parte reconstruye el mensaje para recibir las instrucciones completas y hacerlas entendibles ya que de otra forma solo se trata de una lista de 1 s y 0 s sin sentido.

Existe una gran variedad de protocolos de comunicación serial, cada uno con sus propias características y propósitos especiales. Algunos ejemplos los mas utilizados son: I<sup>2</sup>C, USB, UART o SPI. Donde este último es usado frecuentemente en microcontroladores y dispositivos electrónicos.

### 2.6.1. Comunicación SPI

La interfaz periférica serial (SPI: *Serial Peripheral Interface*) fue desarrollada por Motorola en la década de 1980 (14), y se define como una interfaz síncrona, dúplex completa basada en una estructura maestro-esclavo (15) lo que significa que el maestro comparte su señal de reloj con el esclavo o esclavos para ser sincronizado(s).

Hay dos tipos de configuraciones, de tres y cuatro cables, en este trabajo los dispositivos seleccionados, ESP32-32E, Raspberry Pi Pico y ADC7124-8 utilizan una interfaz de 4 cables. La Fig. 2.25 muestra los cuatro elementos de la comunicación SPI, en donde: 1) CLK es la señal de reloj, 2) MISO significa Maestro *In* Esclavo *Out*, lo que significa que el maestro está enviando datos al esclavo, 3) MOSI significa Maestro *Out* Esclavo *In* en este caso, el esclavo está enviando datos al maestro, y 4) CS que significa *Chip Select*, se utiliza para seleccionar al esclavo que enviará información, CS debe estar en 1 lógico para desconectar al esclavo del bus y en 0 lógico para activar el bus para el esclavo.

La comunicación SPI se basa en la señal del reloj y sus parámetros de polaridad y fase que definen los cuatro modos posibles de funcionamiento. Según la Documentación de Analog Devices<sup>TM</sup> (15), la polaridad está relacionada con el período de transición de CS de alto a bajo para iniciar la comunicación o activar al esclavo y viceversa de bajo a alto para finalizar la comunicación o desactivar el esclavo. Por otro lado, la fase está relacionada con el flanco de subida o bajada del reloj para desplazar o muestrear los datos. En la tabla 2.1 se muestran las diferencias de los modos.

Modo SPI	Polaridad	Fase	Descripción
0	0	0	Polaridad del reloj: 0 lógico. Datos muestreados en el flanco ascendente y desplazados hacia afuera en el flanco descendente.
0	0	1	Polaridad del reloj: 0 lógico. Datos muestreados en el flanco descendente y desplazados hacia afuera en el flanco ascendente.
2	1	1	Polaridad del reloj: 1 lógico. Datos muestreados en el flanco descendente y desplazados hacia afuera en el flanco ascendente.
3	1	0	Polaridad del reloj: 1 lógico. Datos muestreados en el flanco ascendente y desplazados hacia afuera en el flanco descendente.

**Tabla 2.1:** Modos de operación para el protocolo SPI.

### 2.6.2. Otros tipos de comunicación Serial

A continuación se describen brevemente otros protocolos de comunicación serial, con la intención de entender de forma general su funcionamiento y uso.

#### 2.6.2.1. I<sup>2</sup>C

I<sup>2</sup>C (I<sup>2</sup>C: *Inter-Integrated Circuit*) (16) es un protocolo de comunicación serial síncrono usado para enlazar más de dos dispositivos. Este protocolo permite comunicar múltiples nodos a un solo dispositivo principal, y al contrario conectar múltiples dispositivos principales a nodos múltiples, esto permite usar un solo microcontrolador que maneja una tarjeta de memoria o usar una sola pantalla.

Existen tres variantes ampliamente usadas del este protocolo, I<sup>2</sup>C Primer, SMBus (por su nombre en inglés *System Management Bus*) y PMBus (por sus siglas en inglés *Power Management Bus*) los dos últimos usados para aplicaciones de sistemas y potencia.

Por brevedad solo se describirá la variante del protocolo I<sup>2</sup>C Primer.

Las conexiones consideran un bus serial de datos (SDA: *Serial Data*), un bus de reloj (SCLK: *Serial Clock*) que es compartido por el dispositivo principal, una línea de alimentación y una tierra en común, como se ve a continuación en la Fig. 2.26. También se puede notar que el protocolo permite conectar diversos nodos y dispositivos principales en una sola red.

## 2. MARCO TEÓRICO

---

Como se observa solo existe un bus de reloj y un bus de datos por lo tanto para administrar el uso del bus las resistencias de pull-up permiten conectar o desconectar los dispositivos del bus.

El valor comercial más común para estas resistencias es de  $4.7\text{ k}\Omega$  para un voltaje  $V_{cc}$  de 5 V. Adicionalmente se recomienda usar cableado blindado 2AWG que proporciona la capacitancia necesaria para tener un bus de 1m de longitud que opera a una velocidad de 100 kBaud o en su caso de 10 m de longitud para velocidades de 10 kBaud.

### *Escritura o lectura de datos*

El proceso de escritura/lectura en los nodos consiste en iniciar la comunicación, buscar un nodo, leer o escribir la instrucción y detener la comunicación.

Para iniciar la comunicación, el dispositivo principal cambia el bus de datos a bajo para que todos los nodos conectados se enteren de que un dispositivo principal va a mandar/solicitar información. A continuación el dispositivo principal cambia a bajo el bus de reloj, a esta secuencia se les llama condición de inicio.

Una vez iniciada la comunicación, el dispositivo principal envía a todos los nodos conectados la dirección que quiere consultar y al final de la palabra indica si se trata de una escritura o una lectura configurando el último bit de la palabra.

Todos los nodos buscan la dirección indicada en sus registros y si la encuentran devuelven un bit “reconocimiento” colocando el bus de data en bajo “por un bit”. Caso contrario, si la dirección no es reconocida, el nodo deja el bus en alto.

Después de recibir el bit de reconocimiento, el dispositivo principal envía la trama de datos si se trata de una escritura o recibe los datos del nodo en caso de ser una lectura.

Al finalizar la transmisión de datos, el transmisor (nodo o dispositivo principal) envía un bit de reconocimiento para confirmar la recepción exitosa de la información.

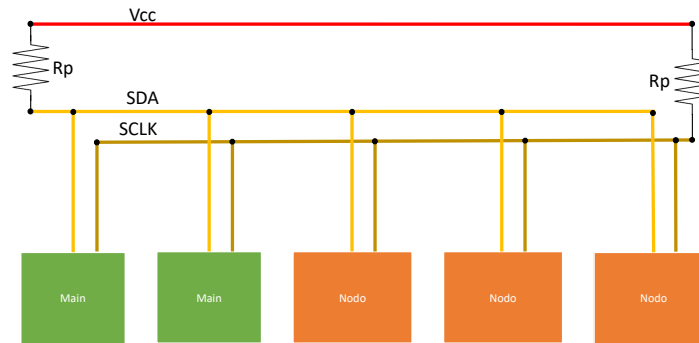
Finalmente, para detener la transmisión el dispositivo principal cambia a alto el bus de reloj y después a alto el bus de datos, esta secuencia se denomina condición de alto.

### **2.6.2.2. USB**

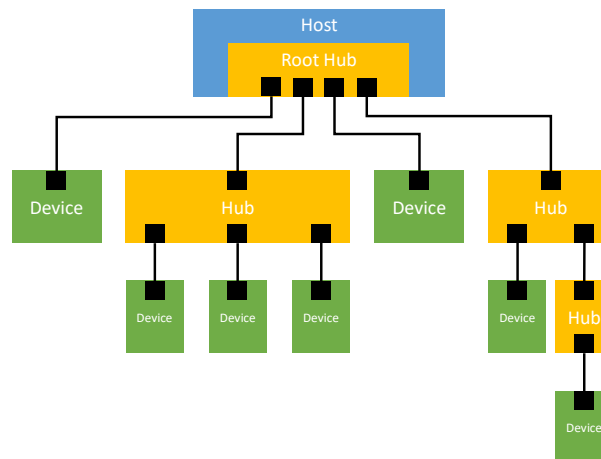
El bus serial universal (USB: *Universal Serial Bus*), es tal vez el protocolo de comunicación serial más exitoso (17). Es utilizado en una gran diversidad de dispositivos de uso cotidiano para la vida moderna, tanto por dispositivos simples como mouses y teclados de computadora así como por dispositivos altamente especializados y de propósito muy específico.

Debido a su popularidad, su uso e implementación es muy sencilla y asequible además de muy estable y segura. En principio el usuario solo debe conectar el dispositivo y la secuencia de autoconfiguración instala el dispositivo y lo prepara para ser utilizado por





**Figura 2.26:** Diagrama de conexiones para el protocolo I<sup>2</sup>C.



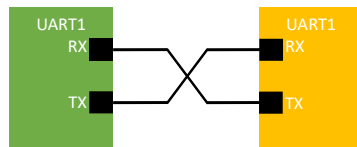
**Figura 2.27:** Arquitectura simplificada del protocolo USB.

el host.

El bus está organizado en forma de árbol y comprende a un host que a su vez contiene el Hub raíz, a este pueden estar conectados otros dispositivos y otros hubs de los que

nacen otras ramas que contienen mas dispositivos u otros hubs a menor nivel a los que se conectan mas dispositivos, Fig. 2.27.

### 2.6.2.3. UART



**Figura 2.28:** Arquitectura básica para protocolo UART.

Receptor-transmisor asíncrono Universal (UART: *Universal Asynchronous Receiver-Transmitter*) (18) es un protocolo de comunicación serial asíncrono de velocidad configurable. Al tratarse de un protocolo asíncrono el reloj del dispositivo principal es diferente al del nodo, es decir cada dispositivo tiene su propio reloj. Por lo tanto, para que se sincronice la transmisión de la información se debe configurar el mismo *boud rate* o velocidad en ambos dispositivos.

Una característica importante de este protocolo es que solo utiliza dos líneas, cada dispositivo tiene entonces un punto de conexión para el transmisor TX y otro para el receptor RX. Como se muestra en la Fig. 2.28.

La transmisión de la información se hace mediante una trama, la cual está formada por un bit de inicio, el paquete de datos a transmitir, un bit de paridad y un bit de alto de transmisión.

#### *Escritura y lectura de datos*

Normalmente la línea de transmisión es colocada en alto cuando no está transmitiendo información. El receptor detecta cuando la línea cambia de alto a bajo y se prepara para recibir la información del transmisor, este puede enviar palabras de 5 - 9 bits. Comúnmente la transmisión se inicia con el bit menos significativo.

Si se utiliza el bit de paridad la palabra podrá ser máximo de 8 bits, este bit es usado para indicarle al receptor si algún dato cambió mientras se hacia la transmisión.

Este protocolo se caracteriza por tener un funcionamiento muy simple. Para realizar una transmisión, el dispositivo:

- recibe la información en un bus paralelo,
- convierte la información en un paquete de datos,

- agrega los bits de inicio de conversión, de paridad y de fin de transmisión.
- envía toda la trama de forma serial a la velocidad o *boud rate* previamente configurado.

Por su parte el receptor inicia a muestrear la información recibida a la velocidad del *boud rate*, descarta el bit de inicio, paridad y fin de transmisión, toma el paquete de datos y lo transforma en una salida paralela.

En este capítulo se expusieron las diferentes características de los convertidores analógicos digitales para así poder determinar el comportamiento de un ADC y discriminar las diferentes opciones que ofrecen los proveedores. Asimismo el repaso a las herramientas estadísticas básicas ayudará a analizar la información experimental que se espera recabar en los diferentes casos de prueba.

Por otra parte, la breve introducción a la técnica Lock-in permitirá plantear a lo largo de este trabajo la arquitectura del sistema de medición al igual que la breve introducción a los protocolos de comunicación serial en especial el SPI para hacer pruebas con diferentes ADCs.



## Diseño del experimento

---

Este capítulo está dividido principalmente en dos partes. La primera sección hablará de la propuesta del sistema de medición de resistencia basada en un LIA y el procesamiento digital de la señal. De forma que se presenta la descripción de la arquitectura, posteriormente el análisis matemático de los diferentes componentes para describir el acoplamiento de la señal analógica además se presenta el desarrollo del análisis digital de la señal para así obtener el valor de la resistencia de prueba. Finalmente se presentan las simulaciones<sup>1</sup> de circuito analógico para verificar la validez de las ecuaciones matemáticas propuestas.

Por otra parte, en las secciones siguientes, se presentan dos propuestas de medición basadas en el ADC7124-8 de Analog Devices<sup>TM</sup>, en estas propuestas la variante es el microcontrolador utilizado para el procesamiento de la señal adquirida por el ADC. A lo largo del capítulo se presenta el procedimiento de instalación y configuración de las herramientas de software utilizadas, posteriormente se describe la técnica de adquisición de los datos, así como el análisis considerando diferentes escenarios propuestos para probar los diferentes parámetros de configuración del ADC7124-8.

### **3.1. Sistema de medición con la técnica Lock-in y la tarjeta de desarrollo MSP432P401R**

En esta sección se describirá el proceso de diseño del experimento para la medición de resistencia con la técnica Lock-in usando el microcontrolador MSP432P401R.

Básicamente la sección está dividida en la propuesta de la arquitectura, el análisis

---

<sup>1</sup>Las simulaciones fueron realizadas con el software de prueba Multisim 14.3.0 Education Edition.

### 3. DISEÑO DEL EXPERIMENTO

---

matemático y la propuesta de diseño, el desarrollo del código en el microcontrolador y la presentación de los resultados de las simulaciones realizadas.

Primeramente se hace una descripción detallada del acoplamiento de la señal analógica mediante la arquitectura propuesta, una vez explicado el tratamiento de la entrada se analiza matemáticamente el circuito para obtener las ecuaciones que describen el comportamiento del circuito, estas a su vez se utilizarán para el cálculo de la resistencia de prueba. Por otra parte se obtienen las ecuaciones de diseño por ejemplo para la ganancia del amplificador de instrumentación, el disparador Schmitt y el cambiador de nivel.

Después de analizar la electrónica analógica de la propuesta se explica la propuesta de código para la programación del microcontrolador que será utilizado como fuente digital, para digitalizar la señal con su ADC tipo SAR de 14 bits y para realizar el cálculo de la resistencia de prueba.

Finalmente, se presentan los resultados de las simulaciones de diferentes secciones del circuito analógico.

#### 3.1.1. Descripción de la arquitectura

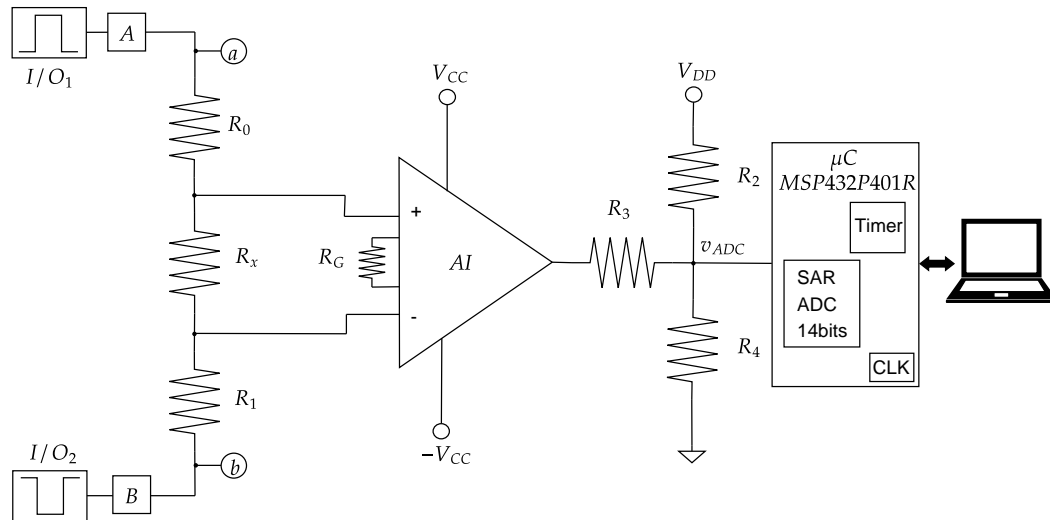
La arquitectura propuesta a continuación está basada en la técnica lock-in (8). La Fig. 3.1 muestra las etapas del acoplamiento de la señal de entrada para poder ser procesada en el microcontrolador de Texas Instrument MSP432P401R.

Primeramente se tienen dos entradas digitales ( $I/O_1$  e  $I/O_2$ ) que funcionarán como los voltajes de referencia del LIA, dado que el circuito está pensado para ser de bajo consumo energético, la alimentación de las terminales digitales la proporcionará el microcontrolador. Después de la terminal digital, se considera un circuito Schmitt Trigger para tener un nivel de voltaje idéntico en ambos lados del circuito, tanto para el nivel alto como para el nivel bajo, ya que en la práctica el nivel alto no llega a los 3.3 V ni el nivel bajo a 0 V como idealmente se considera en la teoría.

Después del disparador Schmitt se colocan dos resistores de protección para el amplificador de instrumentación,  $R_0$  y  $R_1$ . En las terminales diferenciales del amplificador de instrumentación, se coloca la resistencia de prueba  $R_x$ .

El circuito es alimentado alternadamente por las terminales digitales por lo tanto el arreglo de resistencias  $R_0$ ,  $R_1$  y  $R_x$  puede considerarse como un divisor de voltaje al tomar en cuenta la alta impedancia del AI.

La diferencia de voltaje en el AI es amplificada por una ganancia  $G$ , de tal forma que a la salida se tiene una señal bipolar proporcional a la resistencia de prueba  $R_x$ . Dado, que el ADC del microcontrolador MSP432P401R es usado para señales sobre 0 V, se agrega un convertidor bipolar-unipolar formado por las resistencias  $R_2$ ,  $R_3$  y  $R_4$ . Esto



**Figura 3.1:** Arquitectura completa.

permite tener solamente niveles positivos para la señal que será digitalizada por el ADC del microcontrolador.

Posteriormente al acoplamiento analógico de la señal de entrada, se inicia el procesamiento digital. Para comenzar, el ADC embebido en la tarjeta digitaliza la señal analógica. La cuenta obtenida es convertida al voltaje equivalente  $V_x$ , posteriormente se hace el cálculo de la resistencia de prueba  $R_x$  considerando la ganancia del AI, el *offset* agregado en el convertidor bipolar-unipolar y el divisor de voltaje en las terminales digitales.

Finalmente, en el entorno de desarrollo (IDE: *Integrated Development Environment*) de microcontrolador (*Code Composer*) es posible visualizar los resultados de las mediciones. A continuación se detallarán las condiciones de diseño y particularidades importantes de cada uno de los componentes aquí mencionados.

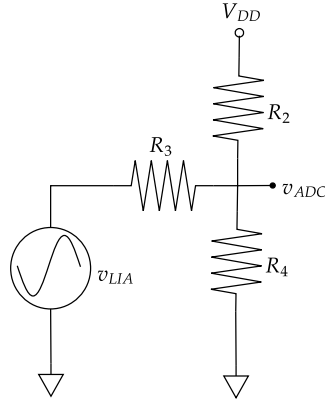
### 3.1.2. Análisis del sistema

El voltaje  $v_{ADC}$  está definido por la salida del amplificador de instrumentación y por el *offset* que agrega el convertidor bipolar-unipolar, la Fig. 3.2 ilustra lo anterior.

Analizando detenidamente el comportamiento del circuito, la salida continua del amplificador de instrumentación puede ser representada como una fuente de alterna  $v_{LIA}$  de forma que el arreglo de resistencias  $R_2$ ,  $R_3$  y  $R_4$  definirán el *offset* necesario para que la salida,  $v_{ADC}$ , no sea negativa y el ADC pueda realizar la conversión correctamente. En el circuito se observan dos fuentes de voltaje, una de directa  $V_{DD}$  y otra de

### 3. DISEÑO DEL EXPERIMENTO

---



**Figura 3.2:** Convertidor Bipolar-Unipolar.

alterna  $v_{LIA}$ , para la respuesta total  $v_{ADC}$  se deben obtener  $v_{adc}$  y  $V_{ADC}$ . Para realizar el análisis se considera el principio de superposición, teniendo dos casos, en el primero solo se toma en cuenta la fuente de alterna  $V_{LIA}$ , posteriormente se hace el análisis considerando únicamente la alimentación de directa  $V_{DD}$ .

De forma que, para el primer caso el circuito tiene como entrada  $V_{LIA}$  y  $V_{DD}$  es una conexión a tierra. La componente de alterna que entrará al ADC está definida por el divisor de voltaje formado por las resistencias en paralelo  $R_2$  y  $R_4$  que están conectadas en serie con la resistencia  $R_3$ , la ecuación 3.1 describe el divisor de voltaje resultante.

$$v_{adc} = v_{LIA} \frac{R_2 || R_4}{R_3 + R_2 || R_4} \quad (3.1)$$

Por otra parte, cuando solo se considera la componente de directa proporcionada por  $V_{DD}$ ,  $V_{LIA}$  es una conexión a tierra. El voltaje  $V_{ADC}$  está definido por el divisor de voltaje resultante del paralelo de las resistencias  $R_3$  y  $R_4$  que está conectado en serie con la resistencia  $R_2$ , la ecuación 3.2 describe el divisor de voltaje resultante.

$$V_{ADC} = V_{DD} \frac{R_3 || R_4}{R_2 + R_3 || R_4} \quad (3.2)$$

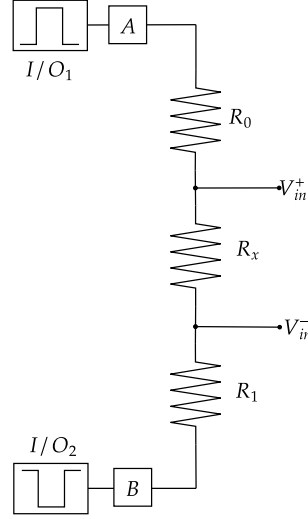
Finalmente, aplicando superposición, las ecuaciones 3.1 y 3.2 forman la respuesta total  $v_{ADC}$ .

$$v_{ADC} = V_{ADC} + v_{adc}$$

$$v_{ADC} = V_{DD} \frac{R_3 || R_4}{R_2 + R_3 || R_4} + v_{LIA} \frac{R_2 || R_4}{R_3 + R_2 || R_4} \quad (3.3)$$



en donde  $v_{LIA}$  corresponde a la salida del amplificador de instrumentación. Considerando la arquitectura completa, Fig. 3.1, se observa cómo es que las terminales digitales ( $I/O_1$  e  $I/O_2$ ) encenderán alternativamente las entradas del amplificador de instrumentación dando así la posibilidad de tener dos casos, primero: cuando  $I/O_1$  está encendida y  $I/O_2$  apagada, segundo:  $I/O_1$  está apagada y  $I/O_2$  encendida.



**Figura 3.3:** Alimentación del amplificador de instrumentación.

Para encontrar el valor de la resistencia de prueba  $R_x$  se considera el voltaje diferencial en la resistencia,  $V_x$ . Tomando en cuenta la ecuación característica del amplificador de instrumentación (2.12), la diferencia de voltaje en las terminales del amplificador es  $V_x$ , Fig. 3.3.

Por lo tanto, la salida del AI se puede reescribir como

$$v_{LIA} = G(v_{in}^+ - v_{in}^-)$$

$$v_{LIA} = GV_x, \quad (3.4)$$

en donde  $G$  es la ganancia y  $V_x$  el voltaje diferencial de entrada del amplificador de instrumentación

$$V_x = v_{in}^+ - v_{in}^-, \quad (3.5)$$

considerando lo anterior es posible despejar  $V_x$  de la ecuación 3.4

$$V_x = \frac{v_{LIA}}{G}, \quad (3.6)$$

### 3. DISEÑO DEL EXPERIMENTO

---

por su parte despejando  $v_{LIA}$  de la ecuación 3.3

$$v_{LIA} = \frac{1}{\frac{R_2||R_4}{R_3+R_2||R_4}} (v_{ADC} - V_{DD} \frac{R_3||R_4}{R_2 + R_3||R_4}) \quad (3.7)$$

y sustituyendo  $v_{LIA}$  de la ecuación 3.7 en  $V_x$  de la ecuación 3.6, se obtiene la primera expresión para calcular  $V_x$

$$V_x = \frac{1}{G} \cdot \frac{R_3 + R_2||R_4}{R_2||R_4} \cdot (v_{ADC} - V_{DD} \frac{R_3||R_4}{R_2 + R_3||R_4}) \quad (3.8)$$

A continuación se presenta el desarrollo para los dos casos que dependen del encendido de las terminales digitales del microcontrolador.

**Caso I** *Consideraciones:*  $I/O_1 = 1$  (encendida),  $I/O_2 = 0$  (apagada). Dado que los niveles lógicos del microcontrolador no son 3.3 V y 0 V exactamente, se nombrará  $V_{IO_1}$  al voltaje en el nodo  $a$  y  $V_{IO_2}$  en el nodo  $b$  que son las salidas del disparador Schmitt, Fig. 3.4. Por lo tanto el voltaje en la terminal no inversora del amplificador de instrumentación  $v_{in}^+$  será

$$v_{in}^+ = V_{IO_2} + (V_{IO_1} - V_{IO_2}) \frac{R_x + R_1}{R_0 + R_x + R_1}, \quad (3.9)$$

para simplificar se puede considerar que  $R_0 = R_1$  entonces  $R = R_1$ , al hacer esta consideración y suponiendo que el voltaje de encendido de las terminales digitales es igual en ambas, la diferencia de voltaje para ambos casos (I y II) será igual y las ecuaciones de diseño siguientes serán equivalentes, por lo tanto

$$v_{in}^+ = V_{IO_2} + (V_{IO_1} - V_{IO_2}) \frac{R_x + R}{2R + R_x}. \quad (3.10)$$

De forma similar, el voltaje en la terminal inversora del amplificador de instrumentación  $v_{in}^-$  es

$$v_{in}^- = V_{IO_2} + (V_{IO_1} - V_{IO_2}) \frac{R}{2R + R_x}. \quad (3.11)$$

Mientras que de las ecuaciones 3.10 y 3.11 se obtiene el voltaje diferencial que entra al amplificador de instrumentación  $V_x^+ = v_{in}^+ - v_{in}^-$ , haciendo la sustitución

$$v_{in}^+ - v_{in}^- = V_{IO_2} + (V_{IO_1} - V_{IO_2}) \frac{R_x + R}{2R + R_x} - (V_{IO_2} + (V_{IO_1} - V_{IO_2}) \frac{R}{2R + R_x}), \quad (3.12)$$

despejando  $R_x$ , se tiene la ecuación 3.13 para calcular la resistencia de prueba cuando  $I/O_1 = 1$  (encendida),  $I/O_2 = 0$  (apagada).

$$V_x^+ = (V_{IO_1} - V_{IO_2}) \frac{R_x}{2R + R_x}$$

$$\mathbf{R}_x^+ = \frac{2\mathbf{R}V_x^+}{V_{IO_1} - V_{IO_2} - V_x^+} \quad (3.13)$$

**Caso II** *Consideraciones:*  $I/O_1 = 0$ ,  $I/O_2 = 1$ . La ecuación que describe el voltaje en las terminales del amplificador de instrumentación se define de igual forma por el divisor de voltaje entre las resistencias  $R$  y  $R_x$ , dado que el valor de las terminales digitales no es constante, se considera al igual que en el Caso I que los voltajes en los nodos  $a$  y  $b$  son  $V_{IO_1}$  y  $V_{IO_2}$  respectivamente.

El voltaje en la terminal no inversora,  $v_{in}^+$ , del amplificador de instrumentación se describe con la ecuación 3.14

$$v_{in}^+ = V_{IO_1} + (V_{IO_2} - V_{IO_1}) \frac{R}{2R + R_x}, \quad (3.14)$$

de forma similar, el voltaje en la terminal inversora  $v_{in}^-$  es

$$v_{in}^- = V_{IO_1} + (V_{IO_2} - V_{IO_1}) \frac{R + R_x}{2R + R_x}, \quad (3.15)$$

entonces el voltaje diferencial que entra al amplificador de instrumentación  $V_x^-$  está descrito por la diferencia de voltaje  $v_{in}^+$  y  $v_{in}^-$

$$v_{in}^+ - v_{in}^- = V_{IO_1} + (V_{IO_2} - V_{IO_1}) \frac{R}{2R + R_x} - (V_{IO_1} + (V_{IO_2} - V_{IO_1}) \frac{R + R_x}{2R + R_x})$$

$$V_x^- = -(V_{IO_2} - V_{IO_1}) \frac{R_x}{2R + R_x}, \quad (3.16)$$

despejando  $R_x$ , se obtiene la ecuación 3.17 para calcular el valor de la resistencia de prueba para las condiciones de operación ya descritas

$$\mathbf{R}_x^- = \frac{-2\mathbf{R}V_x^-}{V_{IO_2} - V_{IO_1} + V_x^-} \quad (3.17)$$

### 3. DISEÑO DEL EXPERIMENTO

---

Finalmente se puede decir que la respuesta del sistema depende de la resistencia de prueba  $R_x$ , el acoplamiento de la señal da como resultado un voltaje equivalente de salida que será convertido por el ADC. Considerando las ecuaciones 3.8 y 3.13 se obtiene una expresión general que describe al voltaje que entra al ADC en términos de la resistencia de prueba  $R_x$ , ecuación 3.18.

$$V_{ADC} = G \cdot \frac{V_{IO1} - V_{IO2}}{\frac{2R}{R_x} + 1} \frac{R_2 || R_4}{R_3 + R_2 || R_4} + V_{DD} \frac{R_3 || R_4}{R_2 + R_3 || R_4} \quad (3.18)$$

**Amplificador de instrumentación** Uno de los componentes más importantes de la arquitectura es el amplificador de instrumentación algunos modelos populares son el AD620AN o el INA114, en cualquier caso al tratarse de un circuito integrado es necesario considerar la polarización con fuentes simétricas  $+V_{CC}$  y  $-V_{CC}$  por practicidad se recurre comúnmente a la polarización con  $+5$  V y  $-5$  V. Para el diseño de este experimento se selecciona el amplificador de instrumentación AD620AN, de forma que al referirse a la hoja de especificaciones (19) se puede encontrar la ecuación característica de la ganancia del amplificador.

Proponiendo una ganancia  $G$  de 100,  $R_G$  tendrá que ser  $498.98 \Omega$  y dado que se busca una ganancia precisa en el AI,  $R_G$  será ajustada mediante una resistencia variable.

$$G = 1 + \frac{49.4 \text{ k}\Omega}{R_G} \quad (3.19)$$

$$R_G = \frac{49.4 \text{ k}\Omega}{G - 1}$$

**Convertidor Bipolar-Unipolar** El propósito de este arreglo es desplazar los niveles negativos que entrega el amplificador de instrumentación a un voltaje positivo y escalar a  $3.3$  V aquellos valores que excedan este límite, esto con el objetivo de proteger el convertidor ADC de la tarjeta MSP432P401R.

Para determinar los valores de las resistencias  $R_2$ ,  $R_3$  y  $R_4$  se considera la ecuación 3.3 para  $v_{ADC}$ .

Dado que existen tres variables y solo se tiene una ecuación, es necesario el uso de herramientas matemáticas para hacer una aproximación de la solución, para esto se plantean los casos extremos de operación del amplificador de instrumentación: 1) el AI en saturado negativamente y 2) el AI saturado positivamente. El primer caso es asignado a  $0$  V, por el contrario cuando el AI está saturado con un voltaje positivo la salida máxima del convertidor bipolar-unipolar se plantea que sea  $V_{DD}$  es decir  $3.3$  V, el voltaje de escala completa que acepta el ADC del microcontrolador. Al hacer estas

Variable	Valor
$V_{SAT}^+$	3.767 V
$V_{SAT}^-$	-3.889 V
$V_{DD}$	3.3 V
$R_2$	7.637 k $\Omega$
$R_3$	9.0 k $\Omega$
$R_4$	63.598 k $\Omega$
$V_{ADC}^+$	3.3 V
$V_{ADC}^-$	$1 \times 10^{-9}$ V
$\Delta V_{ADC}$	3.299999 V
$V_{offset}$	1.676 V

**Tabla 3.1:** Valores de diseño del convertidor Bipolar-Unipolar obtenidos con *Solver* de *Excel*.

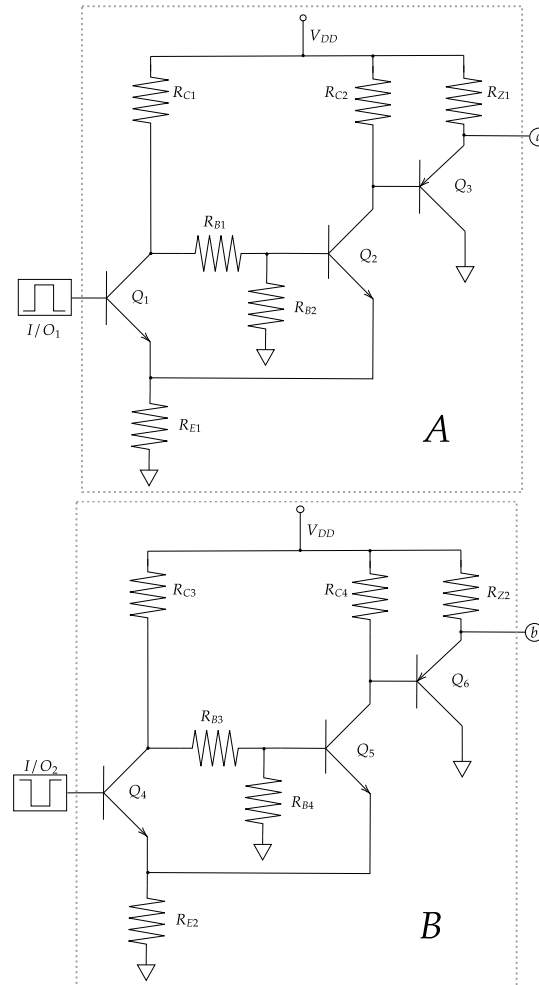
consideraciones y usando la herramienta *Solver* de *Excel* solo es necesario proponer el valor de una de las tres resistencias, además de conocer los voltajes de saturación del amplificador de instrumentación AD620AN que pueden medirse experimentalmente.

Para realizar el cálculo con la herramienta se propone  $R_3 = 9$  k $\Omega$ , al ejecutar la solución se obtiene  $R_2 = 7.637$  k $\Omega$ ,  $R_4 = 63.598$  k $\Omega$ . La tabla 3.1 muestra los resultados obtenidos con *Solver*, en donde los voltajes de saturación  $V_{SAT}^+$  y  $V_{SAT}^-$  del amplificador de instrumentación son medidos experimentalmente.  $V_{ADC}^+$  y  $V_{ADC}^-$  son los voltajes extremos que entregará el convertidor y  $V_{offset}$  es el voltaje que entregará el arreglo cuando  $V_{LIA}$  es 0 V.

**Arreglo Schmitt Trigger** La Fig. 3.4 muestra el circuito propuesto en los bloques *A* y *B* que en realidad son dos circuitos Schmitt Trigger basados en transistores (12). Idealmente los niveles lógicos son 0 V para el 0 lógico y 3.3 V para el 1 lógico, sin embargo, esta suposición no es útil en casos donde la precisión en los voltajes es indispensable. De forma que se realizaron pruebas experimentales previas que mostraron que el microcontrolador MSP432P401R no entrega voltajes constantes en los diferentes pines para los niveles lógicos. Hay que entender que el comportamiento del circuito Schmitt Trigger consiste en proporcionar dos niveles de voltaje (uno alto y otro bajo). La activación de las compuestas dependerá del estado de las terminales digitales, alto y bajo.

A continuación se presenta el análisis matemático para determinar el valor de los componentes y los niveles de voltaje del disparador, por simplicidad se presentará únicamente el análisis para el bloque *A*, sin embargo, los resultados son igualmente aplicables para el bloque *B*.

### 3. DISEÑO DEL EXPERIMENTO

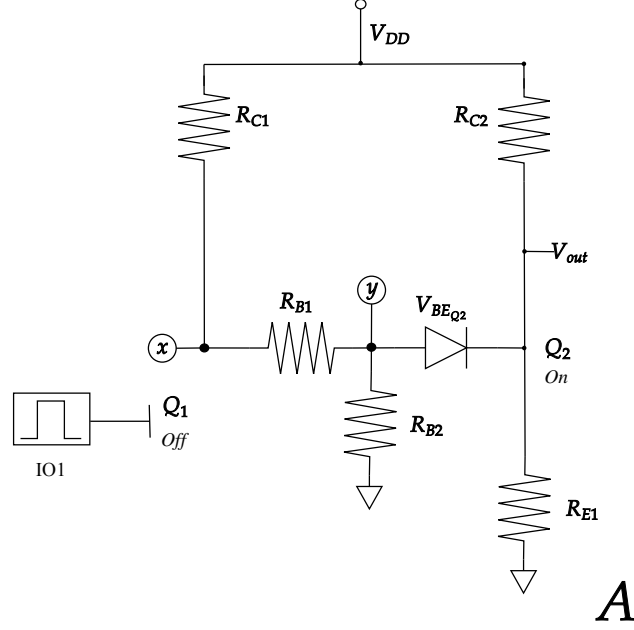


**Figura 3.4:** Estructura interna de bloques A-B (circuito Schmitt-Trigger).

El proceso de diseño se basa en el Caso II (3.1.2) analizado anteriormente, en el cual la terminal digital uno está apagada y la terminal digital dos encendida ( $I/O_1 = 0$  y  $I/O_2 = 1$ ). En este caso la salida está definida por el voltaje base-emisor  $V_{BEQ_2}$  de  $Q_2$  y por la caída de voltaje en  $R_E$ ,  $V_{RE}$ , dado que el objetivo en este caso es que el voltaje de salida sea lo más similar a 0 V, se puede proponer un voltaje en la resistencia  $R_E$  pequeño, de forma que si el nodo en la base de  $Q_2$  se nombra  $y$ ,

$$V_y = V_{BEQ_2} + V_{RE} \quad (3.20)$$

Por su parte, como el transistor  $Q_1$  está apagado, las resistencias  $R_{C1}$ ,  $R_{B1}$  y  $R_{E1}$  están



**Figura 3.5:** Operación del disparador Schmitt cuando  $I/O_2$  está encendida.

conectadas en serie, por lo que para calcular el voltaje en el nodo entre  $R_{C1}$  y  $R_{B1}$  al que se nombrará  $x$ , será

$$V_x = \frac{R_{B1} + R_{B2}}{R_{C1} + R_{B1} + R_{B2}} V_{DD}. \quad (3.21)$$

Otra forma de ver el voltaje en el nodo  $x$  es considerar que ya se ha definido el voltaje en  $y$  por lo tanto, como se desea distribuir el voltaje  $V_{DD}$  entre  $R_{C1}$  y  $R_{B1}$ , se propone que sea la mitad

$$V_x = \frac{V_{DD} - V_y}{2}, \quad (3.22)$$

de esta forma se puede calcular numéricamente el valor de  $V_x$ .

Considerando la malla externa del circuito, se observa que el voltaje de alimentación es igual a la caída de voltaje en la resistencia  $R_{C2}$  más el voltaje a la salida

$$V_{DD} = V_{R_{C2}} + V_{out}, \quad (3.23)$$

por la ley de Ohm

$$V_{DD} = I_{C2} R_{C2} + V_{out},$$

### 3. DISEÑO DEL EXPERIMENTO

---

despejando la resistencia  $R_{C_2}$

$$R_{C_2} = \frac{V_{DD} - V_{out}}{I_{C_2}}, \quad (3.24)$$

de la ecuación 3.24 solo es desconocida la corriente  $I_{C_2}$ , sin embargo, con propósitos de diseño se puede proponer un valor para la corriente de base que esté dentro del rango indicado en las hojas de especificación del transistor (20), entonces considerando la relación entre la corriente de colector y base un transistor de unión bipolar (TBJ: *Bipolar Junction Transistor*)

$$I_{C_2} = \beta_{Q_2} I_{B_2}, \quad (3.25)$$

en donde  $I_{B_2}$  es la corriente de base en el transistor  $Q_2$ . Continuando con el proceso de diseño, por simplicidad se propone que  $R_{C_1} = R_{C_2}$ , el cálculo de las resistencia se hace con la ecuación 3.24.

Hasta este punto hace falta calcular el valor de las resistencias  $R_E$ ,  $R_{B_1}$  y  $R_{B_2}$ . Para el valor de  $R_E$  se recurre a la ecuación 3.20 en donde aplicando la Ley de Ohm en  $V_E$  y despejando la resistencia  $R_E$  se tiene que

$$R_E = \frac{V_y - V_{BEQ_2}}{I_E}, \quad (3.26)$$

considerando que  $I_E \approx I_{C_2}$  y que  $I_{C_2}$  se calculó previamente en 3.25 es posible obtener el valor de  $R_E$ .

Finalmente para  $R_{B_1}$  y  $R_{B_2}$  se retoma la ecuación 3.21. En el nodo  $y$  es posible plantear un divisor de voltaje para la resistencia  $R_{B_2}$ , de forma que

$$V_y = \frac{R_{B_2}}{R_{C_1} + R_{B_1} + R_{B_2}} V_{DD}, \quad (3.27)$$

considerando las variables de las ecuaciones 3.21 y 3.27 solamente  $R_{B_1}$  y  $R_{B_2}$  son desconocidas de forma que se puede proponer un sistema de ecuaciones

$$\begin{aligned} -R_{C_1} - R_{B_1} + R_{B_2} \left( \frac{V_{DD}}{V_y} - 1 \right) &= 0 \\ R_{C_1} + R_{B_1} \left( 1 - \frac{V_{DD}}{V_x} \right) + R_{B_2} \left( 1 - \frac{V_{DD}}{V_x} \right) &= 0 \end{aligned} \quad (3.28)$$

la solución entrega el valor de  $R_{B_1}$  y  $R_{B_2}$ .

Hasta este punto se ha analizado únicamente la sección correspondiente al disparador

---



Schmitt, sin embargo, también es importante considerar una etapa de acoplamiento de impedancias. El transistor  $Q_3$  tipo PNP (positivo-negativo-positivo) en una configuración colector común en conjunto con la resistencia  $R_{Z_1}$  en la Fig. 3.4 hace esta función. Por lo tanto, se puede proponer un valor para  $R_{Z_1}$  bajo para tener una impedancia baja en la salida de los nodos  $a$  y  $b$ . De forma que el voltaje en el nodo será

$$V_a = V_{out} + V_{EB} \quad (3.29)$$

Para el diseño del disparador Schmitt se consideran entonces como componentes el transistor TBJ tipo NPN (negativo-positivo-negativo) modelo 2N2222A y el transistor tipo PNP modelo PN2907A además de los siguientes parámetros:  $V_{DD} = 3.3$  V y una corriente de base de  $10 \mu\text{A}$  cuando el transistor está apagado (20) con una  $\beta_{Q_2} = 100$  para el transistor 2N2222A que se evaluará en la ecuación 3.25.

Se propone un voltaje de salida de 100 mV para cuando la terminal ( $IO_1$  o  $IO_2$ ) está en 0 lógico y se considera un pequeña caída de voltaje en la resistencia  $R_E$  de 0.05 V.

Evaluado la ecuación 3.24  $R_{C_1} = 3.200$  k $\Omega$  y por simplicidad  $R_{C_2} = R_{C_1}$ .

Para el voltaje en el nodo  $y$  (Fig. 3.5) se utiliza la ecuación 3.20 de forma que  $V_y = 0.75$  V. Posteriormente se calcula el voltaje en el nodo  $x$  con la ecuación 3.22 el cual es  $V_x = 1.275$  V. Para continuar se resuelve el sistema de ecuaciones 3.28 del que se obtiene  $R_{B_1} = 829.630 \Omega$  y  $R_{B_2} = 1.185$  k $\Omega$ .

Variable	Valor
$V_{DD}$	3.3 V
$V_{BE}$	0.7 V
$I_{B_2}$	$10 \mu\text{A}$
$\beta_{Q_2}$	100
$V_{out}$	100 mV
$R_{C_1}$	$3.200 \text{ k}\Omega \rightarrow 3.3 \text{ k}\Omega$
$R_{C_2}$	$3.200 \text{ k}\Omega \rightarrow 3.3 \text{ k}\Omega$
$V_y$	0.75 V
$V_x$	1.275 V
$R_{B_1}$	$829.630 \Omega \rightarrow 820 \Omega$
$R_{B_2}$	$1.185 \text{ k}\Omega \rightarrow 1.5 \text{ k}\Omega$
$R_E$	$50 \Omega \rightarrow 47 \Omega$

**Tabla 3.2:** Valores para el circuito Schmitt Trigger.

Finalmente con la ecuación 3.26 se calcula el valor la resistencia del emisor,  $R_E = 50$   $\Omega$ . La tabla 3.2 simplifica los valores de los componentes del diseño del disparador. Se tiene que considerar que la simulación se realizará con los valores comerciales de los

### 3. DISEÑO DEL EXPERIMENTO

---

resistores indicados.

**ADC del microcontrolador MSP432P401R** Después de hacer el acoplamiento de las señales para proteger el ADC embebido en la tarjeta, se procede a configurarlo.

El ADC del microcontrolador MSP432P401R es de arquitectura SAR de 14 bits. La hoja de especificaciones proporciona suficiente información sobre los registros y la secuencia de activación para configurar el ADC a través del microcontrolador.

Entonces, para iniciar se seleccionó indirectamente una frecuencia de muestreo de 12 MHz ya que el ADC tomará una muestra en 4 ciclos de reloj y dado que el reloj del microcontrolador opera a 48 MHz

$$f_s = 12 \text{ MHz}$$

entonces la frecuencia de muestreo del ADC,  $f_s$ , será el inverso de  $4T$ .

La conversión analógica-digital entrega un número digital (puede ser en notación decimal, hexadecimal o binaria, dependiendo de la configuración de visualización de las variables) equivalente al voltaje  $v_{ADC}$ . Por lo tanto para saber a que voltaje corresponde la conversión se necesita calcular primero el valor de 1 LSB, esta depende del voltaje de referencia  $V_{ref}$  y de la resolución  $n$  del ADC,

$$1LSB = \frac{V_{ref}}{2^n}, \quad (3.30)$$

considerando la referencia interna de 3.3 V y una resolución de 14 bits (3.30)

$$1LSB = 201.416 \mu\text{V},$$

de forma que la relación con el voltaje de entrada definido por la resistencia de prueba  $R_x$  es

$$\begin{aligned} conversion &= \frac{v_{ADC}}{1LSB} \\ conversion &= \frac{v_{ADC}}{201.416 \mu\text{V}}, \end{aligned}$$

despejando el voltaje analógico  $v_{ADC}$  se obtiene

$$\begin{aligned} v_{ADC} &= conversion \times 1LSB \\ v_{ADC} &= \frac{conversion}{4964.8484} [\text{V}] \end{aligned} \quad (3.31)$$

Con la ecuación 3.31 es posible iniciar a desarrollar el código en *Code Composer* para calcular el valor de la resistencia de prueba al conocer el valor de  $v_{ADC}$  según el IDE.

### 3.1.3. Desarrollo de Código en Code Composer

Para la propuesta de medición con la técnica Lock-in se utilizó el microcontrolador MSP432P401R, por lo tanto se desarrolló una secuencia de operación para procesar la señal analógica resultante del arreglo de acoplamiento analógico anteriormente descrito (3.1.2).

El código desarrollado está disponible en el repositorio público de *GitHub* de este proyecto. A continuación se hace una breve descripción de su estructura lógica.

El código está dividido en tres secciones: 1) definición de variables y librerías, 2) inicialización de puertos y periféricos, 3) secuencia principal (*main*).

Primeramente, se hace la declaración de las variables globales a utilizar como el valor de la resistencia de prueba y el valor de la conversión del ADC. Además se incluye la librería “msp.h” para poder configurar fácilmente el ADC y las interrupciones del microcontrolador.

Posteriormente se inicializan los puertos a utilizar, dos como salidas digitales (P1.0 y P4.0) que van a ser utilizados para alimentar el circuito conmutando su activación y un pin más como entrada analógica (P6.1 o A14) para el ADC. Nota: es importante revisar el mapa de pines de la tarjeta para seleccionar correctamente el pin de entrada analógica, ya que no todos los pines se pueden configurar como entradas analógicas.

La inicialización de los periféricos se realiza a través de la definición de diferentes funciones que son ejecutadas en el *main*, esto da la ventaja de tener un código más limpio y ordenado. Las funciones de inicialización propuestas fueron para el temporizador (*System Tick Timer*), el ADC de 14 bits y las interrupciones de este. Dado que la configuración del temporizador es realmente sencilla y el fabricante ha dispuesto la información en su sitio web, se omite el desarrollo, la documentación principal puede consultar [aquí](#) (21).

Por otra parte, la configuración del ADC se hace mediante los siguientes registros.

1. Registro de control 0: *ADC14CTL0 Control 0 Register*
2. Registro de control 1: *004h ADC14CTL1 Control 1 Register*
3. Registro de memoria: *ADC14MCTL0 Memory Control 0 Register*
4. Registro de interrupciones: *ADC14IER0 Interrupt Enable 0 Register*

En donde el registro de control 0 es para configurar lo siguiente:

### 3. DISEÑO DEL EXPERIMENTO

---

1. el modo Sample-Hold,
2. el tiempo de muestreo, y
3. la activación del ADC.

En la hoja de especificaciones (22) se ofrecen ocho opciones de muestreo: 4, 8, 16, 32, 64, 96, 128 o 192 ciclos de reloj necesarios para tomar una muestra. Para este trabajo se seleccionaron 4 ciclos de reloj para el tiempo de muestreo, por lo tanto la frecuencia de muestreo es  $\frac{1}{4T}$  como se explicó en 3.1.2. La siguiente sección de código muestra los parámetros seleccionados

```
//ADC14_CTL0_SHT0_0      numero de ciclos de reloj (4)
//ADC14_CTL0_SHP        ADC14 sample-and-hold pulse-mode select
// ADC14_CTL0_ON        activar el ADC
ADC14->CTL0 = ADC14_CTL0_SHT0_0 | ADC14_CTL0_SHP | ADC14_CTL0_ON;
```

El registro de control 1 es para configurar lo siguiente: a) el tiempo del ciclo de conversión y b) la resolución del ADC, si entonces se seleccionan 16 ciclos de reloj y el ADC está configurado para una resolución de 14 bits, el siguiente código muestra lo anterior en el registro de control 1

```
ADC14->CTL1 = ADC14_CTL1_RES_14BIT;    //(0x00 00 00 30)
```

El registro de memoria por su parte es para seleccionar:

1. el modo de operación (diferencial o con referencia a tierra),
2. el voltaje de referencia del ADC, en este caso se selecciona la referencia interna (3.3V), y
3. el canal de conversión es decir el pin de la entrada analógica (P6.1 o A14).

```
ADC14->MCTL[0] |= ADC14_MCTLN_INCH_14
```

El registro de interrupciones es configurado para ejecutar su secuencia al final de la conversión, esto permite aprovechar el cuerpo de la interrupción para hacer el cálculo de la resistencia de prueba  $R_x$ .

```
ADC14->IER0 |= ADC14_IER0_IE0;
```

Por su parte el registro  $ADC14->MEM[0]$  es utilizado para almacenar el valor de la conversión digital cuando se concluye la conversión.

La última funcionalidad definida en el cuerpo del código antes del *main* es la interrupción del ADC, aquí se desarrolla el cálculo de la resistencia de prueba basada en las ecuaciones definidas en la Sección 3.1.2.

La lógica de la conversión se basa en la definición de los parámetros del experimento como la ganancia del amplificador de instrumentación  $G$ , las resistencias  $R$  de protección del AI, los valores de convertidor bipolar-unipolar, el voltaje de referencia y variables auxiliares para realizar los cálculos.

Posteriormente, se define el *offset* de la ecuación 3.8 (término del lado derecho). Como variable auxiliar  $a$  representa el valor que multiplica al inverso de la ganancia del AI.

A continuación se lee el registro de memoria  $ADC14 \rightarrow MEM[0]$  que contiene el valor de la conversión digital. Entonces se hace la conversión del valor a un voltaje usando la ecuación 3.31.

Ahora es posible calcular el voltaje  $V_{LIA}$  con la ecuación 3.7. y con ello se obtiene el voltaje  $V_x$  de la ecuación 3.6. Entonces se evalúa el caso de operación, si el voltaje  $V_{ADC}$  es mayor o igual al del *offset* se considera la ecuación del Caso I, 3.13, sino se usa la ecuación del Caso II, 3.17, para calcular el valor de la resistencia de prueba  $R_x$  a su vez se consideran los voltajes de las terminales digitales acoplados con el circuito Schmitt Trigger basado en transistores.

```
// ADC14 interrupt service routine
void ADC14_IRQHandler(void) {
    float RG=498.89;
    float G=1+49400/RG;
    float R=100;
    float R2=7600;
    float R3=9000;
    float R4=63500;
    float VDD=3.3;
    float a, offset, VLIA, v_io1, v_io2;
    offset = VDD*((R3*R4/(R3+R4))/(R2+(R3*R4/(R3+R4))));
    a=((R2*R4/(R2+R4))/(R3+(R2*R4/(R2+R4))));
    a=1/a;
    val_adc=ADC14->MEM[0];
    SysTick_Wait(900); //T=n/3e+06
    v_adc=val_adc/4964.85;
    VLIA = a*(v_adc-offset);
    Vx= VLIA/G;
}
```

### 3. DISEÑO DEL EXPERIMENTO

---

```
    if (v_adc > offset){ //caso 1
        v_io1=3.036;
        v_io2=0.779;
        Rx=2*R*Vx/(v_io1-v_io2-Vx);
    }
    else{ //caso 2
        v_io1=0.801;
        v_io2=3.051;
        Rx=-2*R*Vx/(v_io2-v_io1+Vx);
    }
    SysTick_Wait(900);
} //end ADC14_IRQHandler
```

La secuencia del *main* ejecuta en orden las funciones de inicialización, primero los puertos, luego el temporizador y finalmente el ADC. Después se crea un ciclo infinito que conmuta la activación de las terminales digitales luego se activa el ADC y se inicia una conversión analógica-digital, como se han activado las interrupciones el código ejecuta la secuencia de la interrupción al completar la conversión y entrega el valor de la resistencia de prueba en la sección de variables al colocar un punto de prueba. La secuencia se repite después del conmutando las terminales digitales.

#### 3.1.4. Simulaciones

Una parte de la propuesta del experimento está basada en componentes analógicos, principalmente la sección correspondiente al acoplamiento de la señal analógica para su conversión digital por el ADC del MSP432P401R.

En primer lugar se realizó la simulación de la arquitectura de la Fig. 3.1 sin contemplar los disparadores Schmitt, sin embargo, en la práctica el comportamiento del sistema no fue el esperado por las variaciones de los valores analógicos en las terminales digitales del microcontrolador, entonces se planteó el diseño para los disparadores y consecuentemente se simuló su funcionamiento.

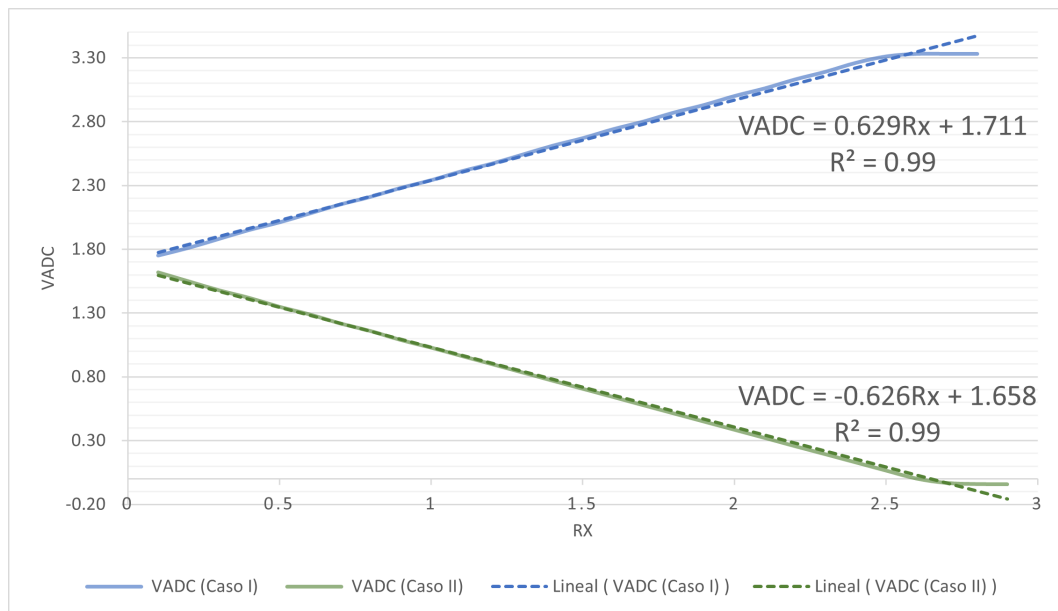
Posteriormente se analiza el comportamiento del AD620AN, principalmente para verificar los voltajes de saturación.

A continuación se simula el comportamiento del cambiador de nivel bipolar-unipolar, así como sus límites de operación.

Finalmente se incluye la simulación de la arquitectura completa considerando el reajuste por los disparadores Schmitt.

**Simulación para la arquitectura completa sin considerar los disparadores Schmitt en las terminales digitales** En este apartado se muestran los resultados de la simulación de la arquitectura completa para la medición de una resistencia variable con la propuesta del LIA que contempla las terminales digitales del microcontrolador ( $I/O_1$  e  $I/O_2$ ), las resistencias de protección  $R$ , el amplificador de instrumentación AD620AN configurado para una ganancia de 100 y finalmente el convertidor bipolar-unipolar para escalar los valores en un rango de 0 V a 3.3 V.

La Fig. 3.6 muestra los resultados de la simulación del circuito con Multisim. Como se observa, existen dos pendientes de conversión una para el Caso I (positiva) y otra para el Caso II (negativa) en donde se considera la conmutación de las terminales digitales, de esta prueba se puede decir que la pendiente de conversión es ligeramente diferente ya que depende del valor experimental de las terminales digitales.



**Figura 3.6:** Simulación de la arquitectura basada en LIA sin el disparador Schmitt para los dos casos de operación.

**Simulación del circuito Schmitt Trigger basado en transistores** Por otra parte, derivado de las pruebas experimentales con el microcontrolador, se agregan los disparadores Schmitt Trigger descritos como cajas negras en la Fig. 3.1 como los bloques

### 3. DISEÑO DEL EXPERIMENTO

---

A y B porque los niveles lógicos medidos en los pines del MSP432P401R no son constantes ni simétricos, es decir el voltaje de  $I/O_1$  cuando está encendida no es el mismo para  $I/O_2$  y de igual forma para cuando están apagadas las terminales. Esta condición complica el cálculo de la resistencia  $R_x$  y la calibración del convertidor bipolar-unipolar derivando a su vez en asimetría en las conversiones.

En el apartado 3.1.2 se describió el proceso de diseño y la tabla 3.2 resume los valores de los componente necesarios a utilizar. Para verificar la propuesta antes de realizar las pruebas experimentales se realizó la simulación del circuito en Multisim A.5.1. Por su parte, la tabla 3.3 muestra la comparación de los resultados de la simulación y el cálculo de los voltajes teóricos.

Como se observa el comportamiento del circuito es el esperado, esto verifica que el método de diseño es correcto.

	Teórico	Simulación
$V_{in}$	0 V	0.60 V
$V_y$	750 mV	701 mV
$V_x$	1.275 V	1.19 V
$V_E$	50 mV	58.5 mV
$V_{out}$	100 mV	113 mV

**Tabla 3.3:** Valores de simulación del circuito Schmitt Trigger en Multisim.

$V_{in}$ [mV]	$V_{out}$ [V]	G
-50.50	-4.01	79.41
-45.10	-4.01	88.91
-40.60	-3.98	98.03
-30.60	-3.05	99.67
-20.50	-2.04	99.51
-10.30	-1.02	99.03
0	$1.69 \times 10^{-3}$	NA
10.30	1.03	100.00
20.50	2.05	100.00
30.60	3.05	99.67
40.60	3.84	94.58
45.10	3.84	85.14
50.50	3.84	76.04

**Tabla 3.4:** Valores de simulación del Amplificador de Instrumentación AD620AN en Multisim.



**Simulación del Amplificador de Instrumentación AD620AN** Esta simulación tiene como propósito comprobar los niveles de saturación del amplificador así como observar el comportamiento de la ganancia del amplificador para una ganancia fija de 100.

En el apéndice A.5 se encuentran los diagramas de simulación. La tabla 3.4 muestra los resultados obtenidos.

Como se puede observar el rango de operación experimental será de +/- 40.60 mV ya que con voltajes mayores el amplificador de instrumentación se satura entregando -4.01 V y 3.84 V.

**Simulación del convertidor bipolar-unipolar** Como se ha explicado en el apartado 3.1.2, el diseño del convertidor está basado en los voltajes experimentales de saturación del amplificador de instrumentación. La tabla 3.1 muestra los valores de los componentes.

A continuación se presentan los valores obtenidos en la simulación para el rango de operación del AI, es decir dentro de los +/-3.8 V. La Fig. 3.7 muestra el comportamiento lineal ( $R^2 = 1$ ) del convertidor bipolar-unipolar propuesto.

**Simulación de la arquitectura completa** Finalmente se realiza la simulación de la arquitectura propuesta considerando a diferencia de la Sección 3.1.4 los disparadores Schmitt para mantener simetría en las mediciones.

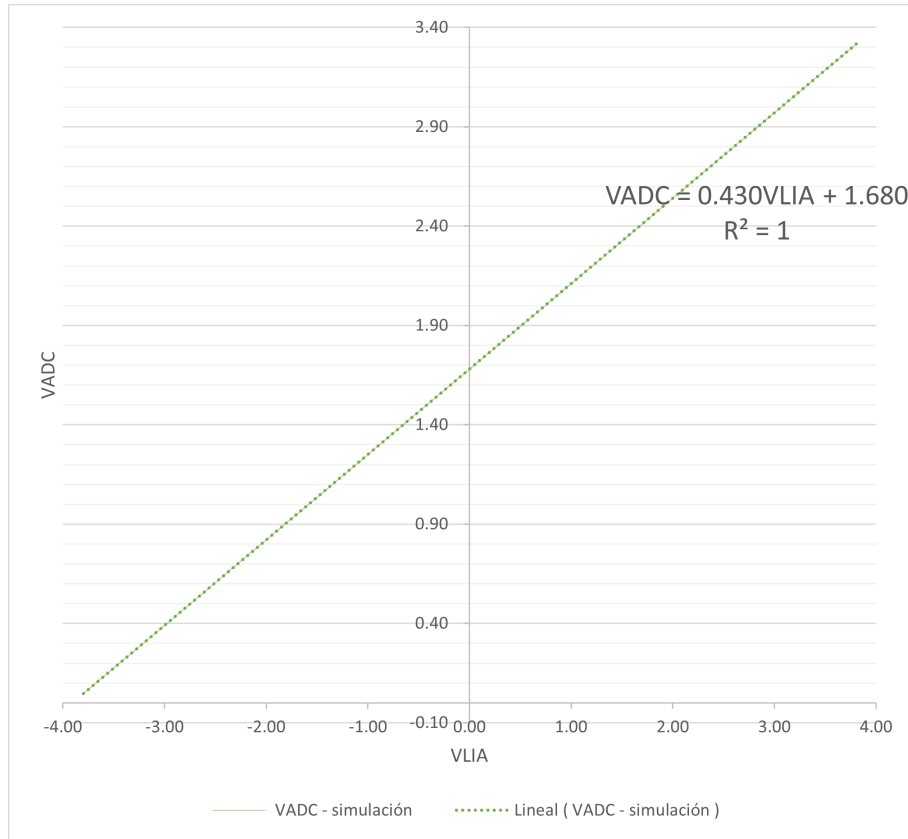
Como se ha presentado anteriormente, la simulación de las diferentes etapas de acoplamiento funciona adecuadamente, por lo tanto para esta prueba se considera una resistencia variable que va de 0.10  $\Omega$  a 4.30  $\Omega$ , esto primeramente para monitorear el voltaje de saturación del AI y posteriormente para comprobar el funcionamiento del convertidor bipolar-unipolar.

La Fig. 3.8 muestra los dos casos de operación del LIA y como se observa el voltaje VADC se encuentra dentro del rango de operación del ADC del microcontrolador. El diagrama de la simulación puede revisarse en el apéndice A.5.2. Por su parte las mediciones realizadas se pueden consultar en el apéndice A.4.2.

**Simulación (prueba) del ADC del microcontrolador MSP432P401R** La Sección 3.1.3 explica detalladamente el proceso de programación del ADC y la lógica detrás de las ecuaciones de diseño propuestas, sin embargo, se hace una prueba previa para verificar el correcto funcionamiento del ADC tipo SAR del microcontrolador.

A continuación se observan en la Fig. 3.9 los resultados de conversión para un voltaje de prueba que varía de 0.10 V a 3.3 V.

### 3. DISEÑO DEL EXPERIMENTO



**Figura 3.7:** Resultados de la simulación en Multisim del convertidor bipolar-unipolar.

De la simulación experimental se puede observar que el ADC tiene un **error de ganancia**, es decir a medida que aumenta el voltaje de entrada el valor de la conversión es mayor al valor esperado.

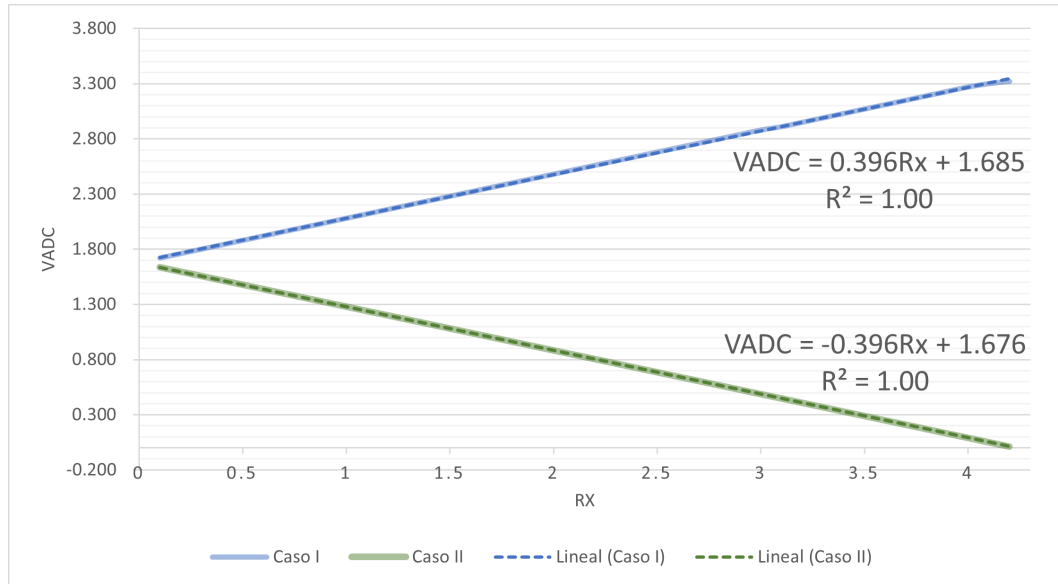
Para analizar el error experimental, se toman en cuenta las ecuaciones de las rectas de conversión experimental y teórica, 3.32 y 3.32<sup>2</sup>, respectivamente.

$$\#Conv_{Exp} = 5060.3v_{ADC} - 1.2577$$

$$v_{ADC_{Exp}} = \frac{\#Conv_{Exp} + 1.2577}{5060.3} \quad (3.32)$$

$$\#Conv_{Teo} = 4964.8v_{ADC} + 2 \times 10^{-11}$$

<sup>2</sup>Esta es la ecuación teórica 3.31



**Figura 3.8:** Simulación de la arquitectura basada en LIA con disparadores Schmitt para los dos casos de operación.

$$v_{ADC_{Teo}} = \frac{\#Conv_{Teo}}{4964.8} \quad (3.33)$$

Observando las pendientes, es notable que la recta experimental alcanzará la cuenta completa, 16,384, a voltaje menor en comparación con la recta teórica, ese voltaje de entrada,  $v_{ADC}$ , será el valor de conversión con el máximo error experimental.

Cuando se evalúa con la recta experimental  $v_{ADC} = 3.2380$  V la conversión es 16,384. Por su parte, para el mismo valor de voltaje, teóricamente se espera una conversión de 16,076.

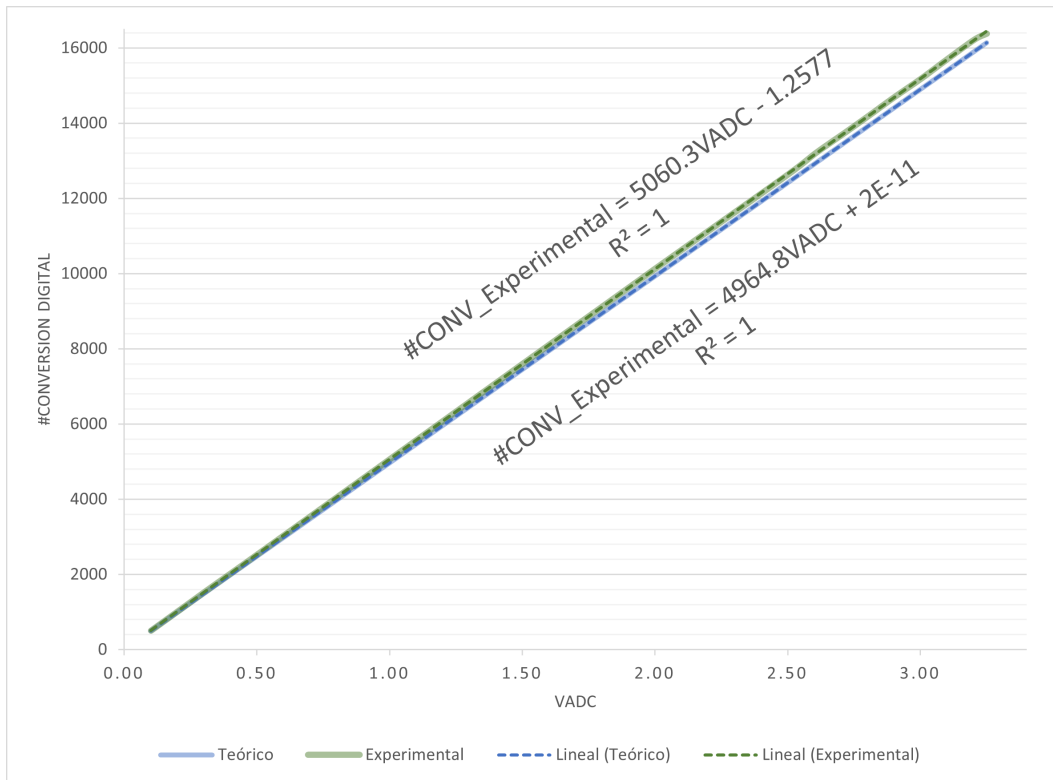
Entonces, el máximo error de prueba experimental de acuerdo con la ecuación 2.11 es

$$\%E = 1.916 \%,$$

esto se tomará en consideración al analizar las mediciones experimentales en el siguiente capítulo. En el apéndice A.4.4 se pueden revisar las mediciones realizadas.

### 3. DISEÑO DEL EXPERIMENTO

---



**Figura 3.9:** Resultados de la simulación-prueba del ADC tipo SAR del microcontrolador MSP432P401R.

### 3.2. Sistema de medición con ADC7124-8 y el microcontrolador ESP32 en el IDE de Arduino

El IDE de Arduino es una herramienta útil para la programación de microcontroladores. Actualmente, muchas librerías de código abierto están disponibles para descargarse, de hecho existen librerías para el microcontrolador ESP32, con las que en consecuencia se puede programar el ADC7124-8 de Analog Devices<sup>TM</sup>, para lo cual solo es necesario agregar las correspondientes librerías para el ADC7124-8. A continuación, el ADC es configurado a través del microcontrolador ESP32 utilizando el IDE de Arduino.

### 3.2.1. Desarrollo y pruebas del código

Una vez concluida la instalación de la librería para el microcontrolador ESP432 y el ADC7124-8, el primer paso es leer y escribir en los registros del ADC utilizando el protocolo de comunicación SPI para la conexión entre el microcontrolador y el ADC.

#### 3.2.1.1. Comunicación SPI

La hoja de datos del ADC7124-8 (23) recomienda, en su sección de *Digital Communication*, utilizar el modo 3 del protocolo de comunicación serial SPI, que muestrea los datos en el flanco ascendente del reloj y desplaza el registro de datos en el flanco descendente. A pesar de estas recomendaciones, al programar el ADC en Arduino el modo SPI que permitía la conexión era el modo 0.

#### 3.2.1.2. Lectura y escritura de registros del ADC

Las funciones definidas en las librerías instaladas se prueban para escribir y leer los diversos registros del ADC. Sin embargo, después de probar los diferentes parámetros posibles las lecturas de los registros no fueron exitosas; por lo tanto, se decidió programar el microcontrolador manualmente para manipular directamente los registros del ADC.

La hoja de datos del ADC (23) incluye un mapa de registros con las direcciones y los valores predeterminados de estos. Como el objetivo es programar el ADC, el primer ejercicio es leer algunos registros de la lista proporcionada. El primero que se probó fue el ID del ADC. El siguiente código en Arduino lo lee correctamente y devuelve el ID esperado.

```
#include <SPI.h>

void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
  Serial.println('Test');

  SPI.begin();
  delay(10);
  SPI.beginTransaction(SPISettings(1000000, MSBFIRST, SPI_MODE0));
```

### 3. DISEÑO DEL EXPERIMENTO

---

```
delay(10);

//Get the device ID
//Ask for ID
byte CommRegWrite=0x40;           //instruction for reading
byte CommOut=CommRegWrite|0x05;   //adress of ID register
Serial.println(CommOut,BIN);
SPI.transfer(CommOut);           //transmit the data to device –
    write in device/slave
delayMicroseconds(1);
byte answer=SPI.transfer(0xFF);   //reading instruction
Serial.println(answer,HEX);
Serial.println(‘‘Finished’’);
}

void loop() {
    // put your main code here, to run repeatedly
}
```

Parece ser un buen comienzo, pero los resultados de las lecturas de los registros de los canales asignados en las direcciones continuas desde 0x09 a 0x18 no coinciden con sus valores predeterminados. Los resultados siempre muestran un desplazamiento en los registros y una lectura errónea para el bit menos significativo (bit 0). Este problema conduce a una nueva opción para la programación de ADC descrita en la siguiente sección.

### 3.3. Sistema de medición para ADC7124-8 utilizando el microcontrolador Raspberry PiPico y MicroPython

Como se explicó, usando las librerías de Arduino no se pudo leer y escribir correctamente los registros con el microcontrolador ESP32, debido a esto, se optó por un nuevo sistema. En el segundo intento, se usó el microcontrolador Raspberry Pi Pico para escribir y leer los registros del ADC a través de la herramienta MicroPython. Para ello, se necesita instalar Python y una terminal para visualizar los resultados. Actualmente, hay muchos IDEs de libre acceso. Para este trabajo se seleccionó Anaconda, en su [sitio web](#) se pueden

encontrar todas las instrucciones de instalación.

Además, es necesario descargar MicroPython, para obtener la última versión disponible visitar su [sitio web](#), la versión ocupada en este trabajo está disponible en el siguiente [repositorio de GitHub](#).

Por último, es necesaria una terminal extra para conectar el microcontrolador con un puerto serial para poder visualizar los resultados, en este caso se seleccionó la Terminal Putty, el ZIP de instalación está disponible en este [repositorio de GitHub](#).

### 3.3.1. Programación de ADC7124

Una vez instalada la herramienta MycroPython en el microcontrolador, el siguiente paso para configurar el ADC7124-8 es establecer una comunicación estable entre los componentes (en este caso usando el protocolo SPI) para leer y escribir los registros del ADC. Posteriormente se propondrá una rutina de conversión y adquisición de datos.

#### 3.3.1.1. Comunicación SPI y escritura/lectura de registros

Por lo desarrollado en este trabajo, no sería práctico utilizar la comunicación en paralelo ya que el ADC7124-8 necesitarían 24 cables lo que es impráctico y además propiciaría problemas técnicos y errores humanos, por lo tanto la comunicación serial resulta ser más simple y fácil de implementar. La documentación disponible en las referencias del [sitio web](#) de la herramienta *MicroPython* propone un método de configuración para la comunicación serial que puede operar en los cuatro modos de la comunicación SPI cambiando los parámetros de fase y polaridad.

Lo primero que hay que verificar son las conexiones entre los componentes, en la Fig. 2.25 se muestran los pines asignados en la instrucción *SoftSPI* que se presenta en la documentación y más adelante en el código de ejemplo. Los cuatro modos de operación



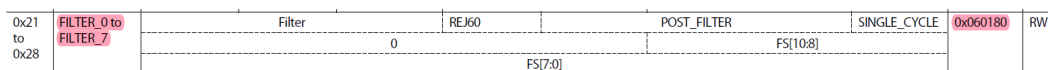
Figura 3.10: Diagrama de conexión para la comunicación SPI.

fueron configurados fácilmente, sin embargo, el que tuvo el mejor desempeño fue el modo 0, es decir cuando se selecciona fase=0 y polaridad=0.

En esta prueba las combinaciones de polaridad y fase fueron configuradas a través de

### 3. DISEÑO DEL EXPERIMENTO

---



**Figura 3.11:** Mapa de bits para registros de filtros.

la terminal para la lectura y escritura de los registros del ADC. Con excepción de la configuración de fase=0 y polaridad=0, los modos 1, 2 y 3 daban siempre resultados que no coincidían con los valores predeterminados que indica la hoja de datos del ADC7128-8.

Como variable de las pruebas se modificó el *baudrate* probando los valores de: 100,000 y 1,000,000. En el primer caso, cuando se seleccionaba la polaridad = 1 y la fase = 0 el *buffer* debe ser limpiado antes de la primera lectura para sincronizar las instrucciones y los resultados, por su parte cuando se seleccionaba el modo 1, polaridad = 0 y fase = 1, era imposible leer los valores predeterminados de los registros.

A continuación, al cambiar el baudrate a 1,000,000 también se detectaron errores de lectura, cuando se configuró el modo 2, polaridad=1 y fase=1, fue necesario leer repetidamente los registros antes de obtener el resultado correcto, en comparación con el modo 0 con el cual todos los resultados de lectura y escritura fueron siempre precisos, debido a esto, se propone utilizar este modo de configuración para establecer la comunicación serial.

La siguiente salida muestra la lectura del registro Filter\_0 del ADC cuando se establecen polaridad = 0, fase = 0 . El código está disponible en el siguiente repositorio de [GitHub](#). Los valores esperados son comparados con el mapa de registros de la hoja de datos del ADC, como se muestra en la Fig. 3.11 el valor de *reset* coincide con la lectura hecha.

```
>>> from machine import Pin, SoftSPI
>>> spi = SoftSPI(baudrate=1_000_000, polarity=0, phase=0, sck=Pin(0),
                 mosi=Pin(2), miso=Pin(1))
>>> x=spi.read(4, 0x61)
>>> x
b'\x00\x06\x01\x80'
```

#### 3.3.1.2. Configuración de un solo canal de ADC para conversiones de voltaje

Una vez que la comunicación SPI es estable, la siguiente tarea es habilitar un solo canal para la conversión analógica-digital de un voltaje. Haciendo referencia a la hoja de datos del ADC (23), la siguiente configuración permite habilitar el *Channel\_0* y convertir la

---



lectura en el valor de voltaje medido, el código se puede encontrar en este [repositorio](#). Antes de explicar el código desarrollado es importante comprender la estructura de cada canal, como se observa en la Fig. 3.12, el mapa de bits muestra que los primeros cinco bits, del [4:0], corresponden a la Entrada Analógica Negativa (AINM), los siguientes cinco [9:5] a la Entrada Analógica Positiva (AINP), los bits 11 y 12 como el 15 deben programarse con 0 para un correcto funcionamiento y activación del canal, finalmente los bits [14:12] permiten al usuario seleccionar la Configuración que es igual a seleccionar el Registro de Configuración que habilitará características como el modo de alimentación, la ganancia, el desplazamiento y las opciones de filtrado, todas esas características son la Configuración del Canal.

Por lo tanto, cuando se escribe en los bits de configuración, por ejemplo, b'001 (formato binario) en los bits *Setup*([14:12]) en el Registro de cualquier *Channel\_XX*, el usuario ha seleccionado los registros CONFIG\_1, FILTER\_1, OFFSET\_1 y GAIN\_1. La Fig. 3.13 ilustra como se elige en los bits de *Setup* el conjunto de registros que caracterizan la operación del canal. Entendiendo lo anterior, para la configuración del *Channel\_0*

Reg.	Name	Bits	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset	RW
0x09 to 0x18	CHANNEL_0 to CHANNEL_15	[15:8] [7:0]	Enable [15]	Setup [14:12]				0 [11:10]	AINP [9:8]		0x8001	RW
				AINP [7:5]				AINM[4:0]				

Figura 3.12: Mapa de bits de registro *Channel*.

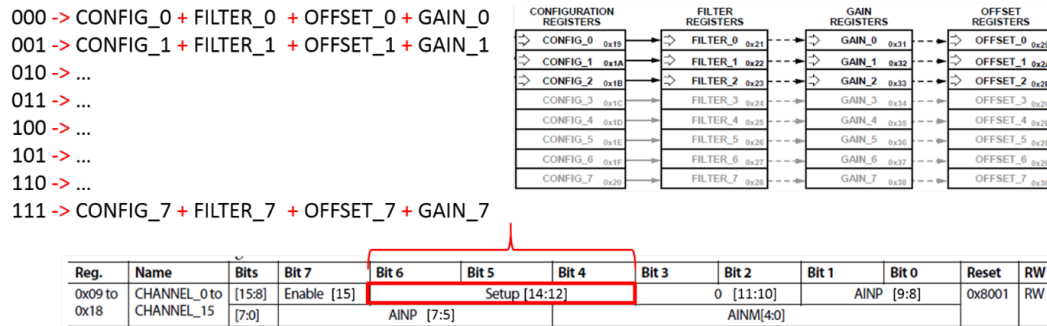


Figura 3.13: Bits de configuración de registro *Channel*.

se seleccionó el registro Config\_0, por lo tanto, en los bits de configuración para la *Channel\_0* se escribió b'000.

Teniendo en cuenta la siguiente arquitectura, Fig. 3.14, el voltaje es medido en el divisor de voltaje formado por las resistencias de 1 kΩ y el voltaje de referencia de 3.3 V.

$$V = 1.65 \text{ V}$$



```
>>> def conv(data):
...     vx=data*2.5/16777216
...     print(vx)
...
>>> x=spi.read(4, 0x42)           #Data register
>>> x
b'\x00\xa7\xad\x9b'
>>> a=0xa7ad9b
>>> conv(a)
1.637482                          #conversion result in volts [V]
```

### 3.3.1.3. Sensor de temperatura interno

Como ejercicio, se activa el sensor de temperatura interno del ADC. La hoja de datos recomienda en su Sección “*Temperature Sensor*” las configuraciones del ADC7124-8 y la ecuación 3.34 para calcular el valor de temperatura en °C.

$$T(^{\circ}C) = \frac{\text{Output value} - 0 \times 800000}{13548} - 272.5 \quad (3.34)$$

El fabricante indica que la desviación del valor real puede ser de 0.50 °C también indica que el error de lectura puede ser inferior al 1.2% cuando se miden temperaturas alrededor de -40 °C (+/-1.5 °C) y menor al 1% cuando se miden temperaturas alrededor de 125 °C (+/- 1.25 °C).

Como resultado, se midió la temperatura ambiente, lo que verificó el correcto funcionamiento del sensor. El siguiente conjunto de configuraciones habilita el sensor de temperatura, el código fuente se puede encontrar en el siguiente repositorio de [GitHub](#).

```
>>>from machine import Pin, SoftSPI
>>>spi = SoftSPI(baudrate=1_000_000, polarity=0, phase=0, sck=Pin(0), mosi
    =Pin(2), miso=Pin(1))
>>>x=spi.read(1, 0x09)           #Channel_0
>>>x=spi.read(1, 0xb2)
```

### 3. DISEÑO DEL EXPERIMENTO

---

```
>>>x=spi.read(1, 0x13)

>>>x=spi.read(1, 0x19)          #Config_0
>>>x=spi.read(1, 0x08)
>>>x=spi.read(1, 0x10)

>>>x=spi.read(1, 0x01)          #ADC_control
>>>x=spi.read(1, 0x01)
>>>x=spi.read(1, 0x00)

>>>x=spi.read(4, 0x42)          #ADC_Data
>>>x
b'\x00\xbc\xfa\xe5'

>>>a=0xbcf5e5-0x800000          #conversion to temperature
>>>a                            #following the previous equation
3996389
>>>a/13584
294.1982
>>>294.1982-272.5
21.69821                        #result in C
```

El código anterior se escribió directamente en el terminal PuTTY, donde el registro de Channel\_0 está configurado para habilitar el sensor de temperatura interno, el registro de Config\_0 establece una ganancia unitaria, se establece una conversión bipolar y la referencia interna para el ADC.

#### 3.3.1.4. Configuración de un solo canal o varios canales de ADC

**Activación de un solo canal.** Siguiendo la medición de resistencia propuesta en la nota de aplicación CN0383 de Analog Devices<sup>TM</sup> (24) del ADC7124-8. Se proponen dos ejercicios para implementar un sensor de temperatura resistivo (RTD: *Resistance Temperature Detector*). Primero una sola lectura utilizando un canal y las conexiones de 2 puntas y segundo, se desarrollo una aplicación para la activación y lectura de múltiples canales, presentando al usuario la posibilidad de seleccionar el canal de activación y lectura.

Además, debido a que el código comenzó a aumentar su extensión, se decidió usar la herramienta Rshell para cargar los archivos Python en el microcontrolador; esto también proporciona la ventaja de estandarizar el código para su uso en otros microcontroladores que pueden trabajar con la herramienta de MicroPython. Otra de las ventajas es que al tener el código en un archivo *.py* es fácil actualizarlo cuando se realiza un cambio, sin embargo, a pesar de esto, la depuración es compleja y difícil, ya que no es ocupado ningún compilador que facilite esta tarea de verificación previa de la sintaxis.

**Activación de un canal.** En primer lugar, se siguen las conexiones sugeridas en la Fig. 3.15 de la nota de aplicación CN0383 (24) para medir una sola resistencia con la técnica de las 2 puntas.

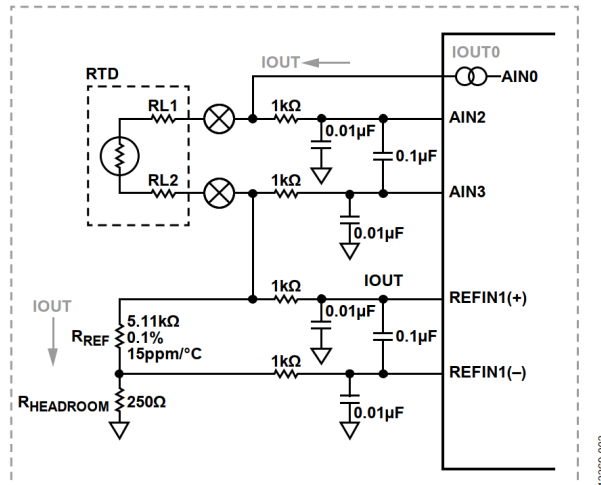


Figure 3. 2-Wire RTD Analog Input Configuration Measurement

**Figura 3.15:** Sensor RTD con conexión de dos puntas (imagen tomada de la Nota de aplicaciones CN0383).

Para una lectura de un canal y teniendo en cuenta que el código se cargará en el microcontrolador, se escribe una librería llamada RTDSettings donde se activa el Channel.0 y se configura para medir desde las terminales AIN2 y AIN3 del microcontrolador, también se asocia a la Configuración 0, eso significa que se tiene que configurar el Registro config.0.

El registro Config.0 es configurado para una lectura bipolar, una ganancia = 16 y para trabajar con las referencias internas en las terminales REFIN1(+) y REFIN1(-), la corriente de excitación se activa en el registro *IO\_Control* seleccionado la terminal AIN0 y una corriente de 500  $\mu\text{A}$ .

Finalmente, el registro *ADC\_Control* está configurado para una conversión continua,

### 3. DISEÑO DEL EXPERIMENTO

---

en el modo de potencia baja (*Low Power*) y con el reloj interno. El código completo se puede encontrar en esta carpeta del repositorio de [GitHub](#).

Si se observa el código, el archivo `main.py` llama al método de inicialización del ADC que activa los registros de configuración. Por su parte, el método de lectura del ADC ignora el primer byte de información al realizar la lectura del registro `DATA`, recordando que se seleccionó un modo bipolar de operación, la nota de aplicación (24) propone la ecuación 3.35 para la conversión de valor de resistencia. La función de conversión del ADC devuelve el valor de resistencia  $R_{RTD}$  considerando como parámetros que  $R_{ref}$  es de 5.11 k $\Omega$  y que la ganancia  $G$  está configurada en 16, finalmente *Conversion* es el valor que devuelve el ADC.

$$R_{RTD} = \frac{\text{Conversion} - 2^{23}}{G \cdot 2^{23}} R_{ref} \quad (3.35)$$

**Resultados.** La siguiente salida presenta los resultados de una ejecución del código en donde se esperaba medir una resistencia de 50  $\Omega$ , experimentalmente se midieron 50.726  $\Omega$ , lo que representa un error de un 1.45 %.

```
>>> a=0x94548b
>>> r=(a-2**23)*5110/(16*2**23)
>>> r
50.72635
```

**Cargar archivos a la placa con Rshell.** Como se indicó, Rshell es una herramienta que permite cargar archivos de python (con extensión `.py`) en la placa para ejecutar y programar el microcontrolador. Para cargar la librería `RTDSettings.py` y el archivo `main.py` en la Raspberry, Pi Pico se siguió el siguiente procedimiento:

1. Abrir la terminal de Anaconda (instalada en el Capítulo 3.3).
2. Para abrir la herramienta, escribir: `rshell`
3. Para enumerar los archivos cargados, escribir: `ls/pyboard`
4. Para eliminar todos los archivos de la placa, escribir: `rm/pyboard/*`
5. Para copiar un archivo en la placa, obtener la dirección del archivo, por ejemplo:

`Documentos/ADCexercise/RTDSettings.py`

6. A continuación, para copiar el archivo se utiliza el siguiente comando, es importante ocupar el mismo nombre del archivo en la copia que se hace en la placa.

```
cp Documentos/ADCexercise/RTDSettings.py
/pyboard/RTDSettings.py
```

7. Se repite lo anterior para el archivo *main.py*

```
cp Documentos/ADCexercise/main.py /pyboard/main.py
```

8. Para verificar la copia, listar los archivos en la placa con la instrucción del paso 3: *ls/pyboard*
9. Después de cargar los archivos, es necesario desconectar la placa para reiniciar el microcontrolador. Al conectarse de nuevo, la salida se puede ver automáticamente en el terminal serie.

**Activación de múltiples canales.** El siguiente ejercicio consiste en proporcionar una aplicación al usuario que permita seleccionar y activar un solo canal/RTD, usando la técnica de dos puntas propuesta en la nota de aplicación CN0383. La cual indica que es posible medir cinco resistencias o sensores RTD diferentes simultáneamente. El diagrama de la Fig. 3.16 muestra la arquitectura.

El código se dividió en dos archivos, una librería, *RTDSettings*, que asigna los pines para cada canal de conversión, también la activación de las corrientes de excitación configuradas en el registro *IO\_Control\_1* que conmuta el pin de configuración y mantiene la magnitud de  $500\mu\text{A}$ , debido a que el usuario puede seleccionar solo un canal a la vez, se implementa un método que activa un canal para realizar una conversión única, la función recibe como parámetro el número de la resistencia o RTD que se desea medir. Por otro lado, el código principal en el archivo *main.py* lee la entrada del usuario para activar el canal con los métodos descritos anteriormente, realiza la conversión y muestra el valor de la resistencia, al finalizar pregunta nuevamente qué canal debe activarse. Ambos archivos se cargaron en la placa con la herramienta rshell descrita en la Sección 3.3.1.4.

Los resultados medidos se visualizaron en la terminal PuTTY, el código completo se puede encontrar en esta carpeta de [GitHub](#).

Para analizar las limitaciones del sistema se calcula el valor máximo de resistencia que se puede medir considerando la arquitectura. Sí entonces el ADC es de 24 bits, el valor de la conversión máxima del mismo es  $2^{24}$ , sustituyendo en la ecuación 3.35,  $R_{RTD_{\text{MAX}}} = 319.375 \Omega$ .

Por otra parte, el valor mínimo que se esperaría medir considerando la resolución del ADC es igual al escalón de cambio entre mediciones, es decir 1 LSB, en este caso la sí la conversión del ADC es 1, al sustituir en la ecuación 3.35 la resistencia mínima medible es  $R_{RTD_{\text{Step}}} = 38.07 \mu\Omega$ , lo que también es la resolución del sistema.

### 3. DISEÑO DEL EXPERIMENTO

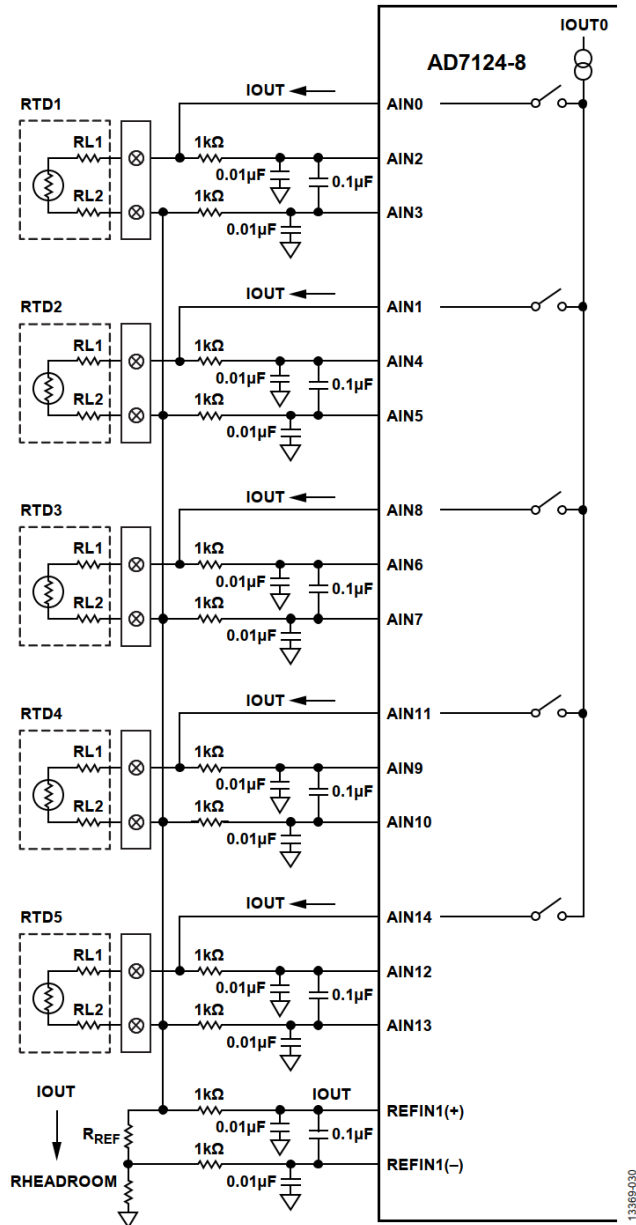


Figure 30. AD7124-8 Multiple 2-Wire Configurations

**Figura 3.16:** ADC7124-8 Múltiples RTDs en configuración de 2 puntas (imagen tomada de la nota de aplicación CN0383).



La siguiente salida en el terminal PuTTY muestra los resultados medidos experimentalmente, considerando que se conectan las siguientes resistencias de prueba:

1. canal 0 - 100  $\Omega$ ,
2. canal 1 - 250  $\Omega$ ,
3. canal 2 - 50  $\Omega$ ,
4. canal 3 - 1 k $\Omega$  y
5. canal 4 - 150  $\Omega$ .

```
Select RTD:
RTD: 0
Channel 0 selected
101.6087
Select RTD:
RTD: 1
Channel 1 selected
250.4924
Select RTD:
RTD: 2
Channel 2 selected
50.84692
Select RTD:
RTD: 3
Channel 3 selected
319.375
Select RTD:
RTD: 4
Channel 4 selected
150.8125
Select RTD:
RTD: 0
Channel 0 selected
101.6062
Select RTD:
```

### 3. DISEÑO DEL EXPERIMENTO

---

RTD	Valor teórico [ $\Omega$ ]	Valor experimental 1 [ $\Omega$ ]	Valor experimental 2 [ $\Omega$ ]
0	100	101.6087	101.6062
1	250	250.4924	250.4493
2	50	50.84692	50.83881
3	1000	<i>Fuera de rango</i>	<i>Fuera de rango</i>
4	150	150.8125	150.7206

**Tabla 3.5:** Prueba del rango de operación del ADC7124-8.

```
RTD: 3
Channel 3 selected
319.375
RTD: 4
Channel 4 selected
150.7206
Select RTD:
RTD: 1
Channel 1 selected
250.4493
Select RTD:
RTD: 2
Channel 2 selected
50.83881
```

Como se muestra en la tabla 3.5, las mediciones están dentro del rango esperado. También se comprueba el rango de operación del sistema con el canal 3 en donde al colocar una resistencia de 1 k $\Omega$  el sistema mide 319.375  $\Omega$  por ello se coloca *Fuera de rango* en la tabla.

#### 3.3.1.5. Configuraciones adicional de ADC.

El ADC7124-8 presenta una amplia variedad de opciones de configuración, en este sentido, dos muy importantes son el funcionamiento del modo de alimentación del ADC y las opciones de filtrado. A continuación se hacen pruebas configurando independientemente y en conjunto dichas características.

F(Full)/L(Low)	Código
F	0x00C0
F	0x0080
L	0x0000

**Tabla 3.6:** Modos de operación y Código.

**Modo de alimentación.** El primer enfoque de la configuración fue la configuración del registro `ADC_Control` que controla el modo de conversión y el modo de potencia. Para fines prácticos, la lectura continua es más apropiada porque realiza las conversiones automáticamente conmutando entre las entradas, utilizando el secuenciador interno del ADC, es importante mencionar que el secuenciador opera solamente en la activación del canal, debido a esto, las corrientes de excitación deben conmutarse manualmente.

Considerando lo anterior, un primer enfoque del código fue activar todos los canales simultáneamente en la etapa de configuración considerando una conversión continua. La tabla 3.6 muestra los valores escritos en el registro `ADC_Control`. Sin embargo, a pesar de las indicaciones de la hoja de datos, no se logró la sincronización entre corrientes y la secuenciación de los canales. Por los resultados obtenidos, se decidió utilizar el modo continuo pero secuenciando tanto la activación de la corriente de excitación así como la activación del canal manualmente. El código para la lectura continua se puede encontrar en esta [carpeta de GitHub](#).

Además, en este conjunto de mediciones se compararon dos de los modos de potencia, el modo *Full Power* y el modo *Low Power*, el modo de potencia permite aumentar la velocidad de datos de salida hasta 19.200 SPS cuando el modo *Full Power* está habilitado, y se establece la opción de filtrado adicional.

En el siguiente extracto de código, la función `def ADC_Init():` muestra el registro `ADC_Control`. Siguiendo el mapa de bits de la hoja de especificaciones del ADC, en el registro de configuración, el primer byte siempre estará `0x00` y el segundo byte definirá el modo de potencia, la tabla 3.6 muestra las opciones para los modos de operación.

```
def ADC_Init():
    #ADC_control Register
    #Internal CLK
    #continuous conversion mode
    x=spi.read(1, 0x01)
    x=spi.read(1, 0x00)           #this byte is always 0x00
    x=spi.read(1, 0xc0)         #this byte defines the power mode
    x=spi.read(3, 0x41)
```

### 3. DISEÑO DEL EXPERIMENTO

---

```
print(“ADC.Init():”, x)
```

La siguiente función configura el ADC en modo *Full Power* y conversión continua.

**Filtrado.** En segundo lugar, una vez que las lecturas fueron exitosas y un conjunto de ellas fueron almacenadas y analizadas (los resultados se muestran en la siguiente Sección 4.3), se agregó una característica adicional al código. Las opciones de filtrado son ampliamente flexibles para programar. En el entendido de que todos los canales utilizan la configuración del registro 0, solo es necesario configurar el registro `Filter_0`, recordado la estructura del ADC (Fig. 3.13).

En esta configuración se seleccionó el filtro SINC4 y un filtrado a la salida de 25 SPS. Ocupando las configuraciones anteriores para los modos de potencia *Full* y *Low* es posible tener tasas de filtrado de 40 y 2,400 SPS para el modo de baja potencia y de 50 y 19.200 SPS para el modo de potencia completa. El siguiente [repositorio de GitHub](#) contiene el código con las configuraciones de filtrado.

Para aclarar, según el mapa de bits del registro de configuración del registro `Filter_0`, en la hoja de datos (23) se indica que al escribir `0xF6`, se activa el filtrado de entradas de 25 SPS, los dos bytes siguientes definirán la velocidad del filtrado en la salida. Es importante diferenciar entre la configuración de filtrado a la salida y la velocidad de datos de salida. El filtrado en la salida está configurado para rechazar señales a 50 y 60 Hz, pero también para disminuir el tiempo de estabilización del filtro en este caso la hoja de datos (23) indica que la velocidad máxima es de 27.27 SPS. Por otro lado, la velocidad de datos de salida es el número de muestras que se toman en un segundo; a veces esto también se conoce como frecuencia de muestreo y se expresa en Hertz [Hz].

La tabla 3.7 muestra un par de configuraciones posibles tanto para el filtrado en la salida como para la velocidad de datos de muestreo. También es importante tener en cuenta que el registro de `ADC.Control` debe configurarse por separado para cambiar el modo de alimentación de potencia baja a completa.

F(Full)/L(Low)	Salida Velocidad de datos [SPS]	Código
F	50	0xF60180
F	19,200	0xF60001
L	40	0xF6003C
L	2,400	0xF60001

**Tabla 3.7:** Configuración de las opciones de filtrado.

El siguiente código muestra el filtrado configurado para un filtrado en la salida de 25 SPS y una velocidad de muestreo de 50 SPS en el modo *Full Power*.

```
#Filter_0 Register  
#post filtering 25 SPS  
#Filter output data rate  
x=spi.read(1, 0x21)  
x=spi.read(1, 0xF6) #this value always activate the postfiltering for  
25 SPS  
x=spi.read(1, 0x01) #1st byte for Output Data Rate  
x=spi.read(1, 0x80) #2nd byte for Output Data Rate
```

### 3.4. Sistema de medición para ADC7124-8 utilizando el microcontrolador ESP32 y MicroPython

Finalmente, la última propuesta de experimento se hace con el microcontrolador ESP32. Como se propuso anteriormente la programación del ADC7124-8 es realmente sencilla al usar la herramienta de Micropython, por lo tanto el código desarrollado se aprovechará en esta sección para programar el ADC con la tarjeta ESP32.

#### 3.4.1. Escritura del código

Con el microcontrolador Raspberry Pi Pico la primera tarea fue activar algunos pines, luego leer y escribir algunos registros del ADC probando la comunicación SPI y sus diferentes modos de operación. De forma similar, se repitió esa secuencia con el ESP32 con la finalidad de reutilizar el código implementado para el microcontrolador Raspberry Pi Pico haciendo únicamente ajustes en los pines y los parámetros de la comunicación SPI, en este caso, se reutilizó la arquitectura propuesta para medir 5 resistencias.

##### 3.4.1.1. Implementación de comunicación SPI

En primer lugar, la comunicación entre el microcontrolador y el ADC tiene que ser estable y fiable para configurar los registros ADC. Como se probó con la Raspberry Pi Pico, la comunicación SPI tiene cuatro posibles modos de operación que dependen de la de fase y la polaridad, para la placa ESP32 ambos parámetros deben establecerse en 1, como indica la siguiente línea de código.

### 3. DISEÑO DEL EXPERIMENTO

```
spi = SoftSPI(baudrate=1000000, polaridad=1, fase=1, sck=Pin(18), mosi=Pin(23), miso=Pin(19))
```

La Fig. 3.17 ilustra las conexiones necesarias para la comunicación SPI.

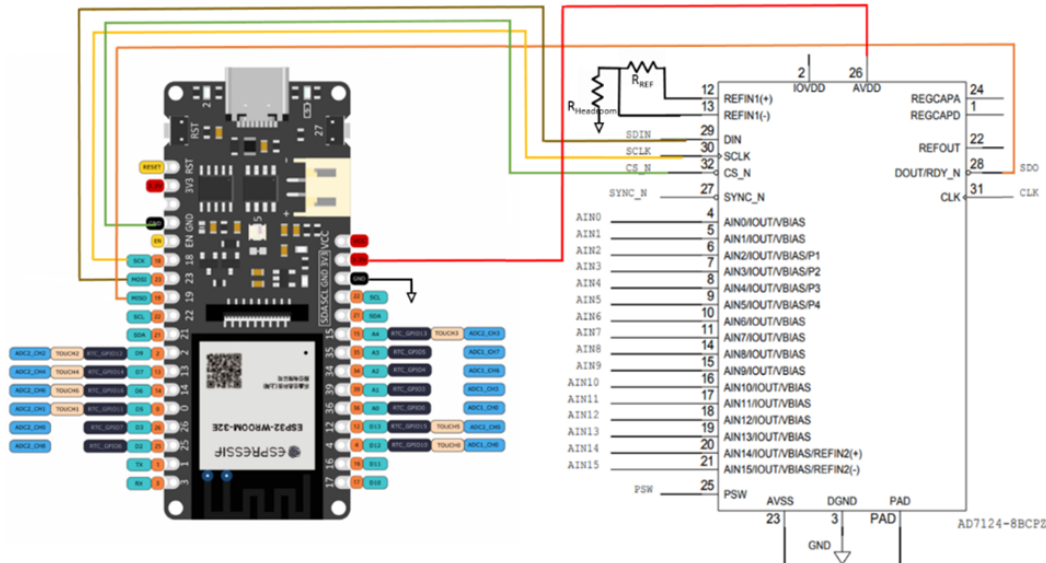


Figura 3.17: Conexiones para la comunicación SPI entre ESP32 y ADC7124-8.

Para probar la herramienta Rshell, se carga el siguiente código en la tarjeta ESP32 para leer continuamente el registro del ID del ADC y escribir en un registro de configuración y en otro de filtrado.

```
from machine import Pin, SoftSPI
from time import sleep

spi = SoftSPI(baudrate=1000000, polarity=1, phase=1, sck=Pin(18), mosi=Pin(23), miso=Pin(19))

# print('status')
# x=spi.read(1, 0x40)

while True:
    #id 0x14
    print('id')
    x=spi.read(2, 0x45)
    print(x)
```

```
#config register 0x0860
x=spi.read(1, 0x19)
x=spi.read(1, 0x08)
x=spi.read(1, 0x04)
print('config')
x=spi.read(3, 0x59)
print(x)
x=spi.read(3, 0x5a)
print(x)

#filter 0x060180
x=spi.read(1, 0x21)
x=spi.read(1, 0xf6)
x=spi.read(1, 0x03)
x=spi.read(1, 0xc0)
print('filter')
x=spi.read(4, 0x61)
print(x)
x=spi.read(4, 0x62)
print(x)

#Channel 0x8001
x=spi.read(1, 0x09)
x=spi.read(1, 0x80)
x=spi.read(1, 0x43)
print('Channel')
x=spi.read(3, 0x49)
print(x)
x=spi.read(3, 0x4a)
print(x)
sleep(5)
```

Para cargar el archivo (disponible en este repositorio [de GitHub](#)), se utilizan los comandos utilizados para la Raspberry Pi Pico. Recordando que es importante considerar la

---

### 3. DISEÑO DEL EXPERIMENTO

---

dirección local en la que este guardado el archivo *.py*.

```
rshell -p COM9
ls /pyboard
rm /pyboard/*
cp Documentos/ESP32example/main.py /pyboard/main.py
```

Es importante recordar que siempre es necesario un archivo *main.py* porque es el archivo que será ejecutado automáticamente al cargarse en el microcontrolador. Además, al cambiar de archivo en el microcontrolador es importante asegurarse que la dirección local de donde se realiza la copia es la deseada, para evitar usar versiones desactualizadas.

#### 3.4.1.2. Implementación de código

Una vez que la comunicación SPI es estable y confiable, es posible cambiar una sola línea en el código usado en la Sección 3.3.1.5 para medir las cinco resistencias. Esta es la mayor ventaja de usar Micro Python para programar el ADC7124-8, porque el mismo código se puede reutilizar en otros microcontroladores ajustando solo parámetros mínimos. En este caso, solo se necesitan establecer los nuevos pines de la placa y establecer la fase y la polaridad en 1, como muestra el siguiente extracto de código.

Con estos ajustes, se realizan las mediciones descritas en el siguiente capítulo.

```
spi = SoftSPI(baudrate=1000000, polarity=1, phase=1, sck=Pin(18), mosi=Pin(23), miso=Pin(19))
```

En este capítulo se presentó el proceso de diseño de los dos sistemas de medición.

Primeramente se describió la arquitectura y se obtuvieron las ecuaciones de diseño del sistema basado en la técnica Lock-in y el microcontrolador de Texas Instruments, así mismo las simulaciones verificaron el comportamiento teórico del sistema. Los resultados de esta sección serán utilizados en el capítulo siguiente para hacer el cálculo de la resistencia de prueba.

Para la segunda propuesta se describió el sistema de medición basado en el convertidor ADC 7124 - 8 en donde se presentaron varios métodos de programación para realizar mediciones en diferentes escenarios. Así mismo se explicó la implementación de diversas herramientas de software que ayudan a tener un sistema de medición robusto. En el siguiente capítulo se hace uso de este código para realizar las mediciones de las resistencias de prueba.



# Mediciones y Análisis de Resultados

---

En este capítulo se presentarán los resultados obtenidos de la implementación de las arquitecturas propuestas en el capítulo anterior. Primero se muestran las condiciones de prueba para la medición de diferentes resistencias con el sistema basado en el amplificador Lock-in, posteriormente se analizan los datos estadísticamente para evaluar el desempeño del dispositivo.

En una segunda parte, se presentan los resultados obtenidos con el ADC7124-8 programado con la tarjeta de desarrollo Raspberry PiPico y con esas mismas configuraciones pero con en la tarjeta de desarrollo ESP32. En ambos casos se hace el análisis de estadístico y se determinan las características principales del sistema de medición.

Finalmente muestra los resultados obtenidos en la prueba de variación del tiempo entra la configuración del ADC y la medición de la resistencia de prueba, en este caso también se hace el análisis estadístico y se presentan los resultados.

Es importante señalar que para la caracterización del instrumento no fue posible utilizar instrumentos patrón que tienen tolerancias menores a 0.01 %, esto debido a las condiciones de confinamiento en las que este trabajo fue realizado. De forma que se recurrió a los componentes disponibles en ese momento.

## 4.1. Resultados del sistema de medición con técnica Lock-in y MSP432

En esta sección se presentan los resultados de las mediciones realizadas con la arquitectura propuesta en la Sección 3.1. Para esto, se proponen 10 elementos<sup>1</sup> diferentes de

---

<sup>1</sup>La lista muestra los el valor medido de los elementos de prueba.

#### 4. MEDICIONES Y ANÁLISIS DE RESULTADOS

---

prueba, 8 resistores de carbón y 2 conductores de diferente longitud.

1. Un paralelo de dos resistores de carbón de  $1.3 \Omega$  ( $0.65 \Omega$ )
2. Un resistor de carbón de  $1.1 \Omega$
3. Un resistor de carbón de  $1.5 \Omega$
4. Un resistor de carbón de  $2.2 \Omega$
5. Un paralelo de dos resistores de carbón de  $3.9 \Omega$  y  $7.6 \Omega$  ( $2.6 \Omega$ )
6. Un resistor de carbón de  $3.4 \Omega$
7. Un resistor de carbón de  $3.9 \Omega$
8. Un paralelo de dos resistores de carbón de  $5.7 \Omega$  y  $14.8 \Omega$  ( $4.1 \Omega$ )
9. Un conductor amarillo de 133 cm de largo calibre 21 AWG
10. Un conductor rojo de 162.3 cm de largo calibre 21 AWG

Para los dos conductores propuestos se hace el cálculo teórico de la resistencia considerando en ambos casos un conductor homogéneo de sección transversal uniforme sometido a un campo eléctrico uniforme, entonces la resistencia de dicho conductor se puede calcular con la ecuación 4.1 (25)

$$R = \rho \frac{L}{A} \quad (4.1)$$

en donde  $R$  es la resistencia del conductor,  $\rho$  es la resistividad del material expresada comúnmente por los fabricantes en  $\Omega \text{ km}$ ,  $L$  es la longitud del conductor y  $A$  es el área transversal del conductor. Los datos anteriores son encontrados en las tablas de especificaciones de los diferentes fabricantes, [SAB Bröckskes](#) indica que para el calibre 21 AWG el área transversal del conductor es de  $0.412 \text{ mm}^2$  y la resistividad del material es de  $44 \Omega \text{ km}$ .

Haciendo el cálculo teórico de la resistencia de los conductores con la ecuación 4.1, el conductor amarillo (133 cm de longitud) tiene una resistencia teórica de  $0.142 \Omega$  y el conductor rojo (162.3 cm de longitud) una de  $0.173 \Omega$ .

Adicionalmente, la tabla 4.1 muestra la tolerancia de cada componente utilizado y la resistencia y tolerancia equivalente para cada caso.

Resistencia de prueba	Elementos	Resistencia Equivalente
1.	$1.3 \Omega \pm 1 \%    1.3 \Omega \pm 1 \%$	$0.65 \Omega \pm 1 \%$
2.	$1.1 \Omega \pm 5 \%$	$1.1 \Omega \pm 5 \%$
3.	$1.5 \Omega \pm 5 \%$	$1.5 \Omega \pm 5 \%$
4.	$2.2 \Omega \pm 5 \%$	$2.2 \Omega \pm 5 \%$
5.	$3.9 \Omega \pm 5 \%    7.5 \Omega \pm 1 \%$	$2.6 \Omega \pm 2.11 \%$
6.	$3.3 \Omega \pm 5 \%$	$3.3 \Omega \pm 5 \%$
7.	$3.9 \Omega \pm 5 \%$	$3.9 \Omega \pm 5 \%$
8.	$5.6 \Omega \pm 5 \%    15 \Omega \pm 5 \%$	$4.1 \Omega \pm 5 \%$
9.	Conductor amarillo 21 AWG - 133 cm	0.142 $\Omega$
10.	Conductor rojo 21 AWG - 162.3 cm	0.173 $\Omega$

**Tabla 4.1:** Valores y tolerancias de las resistencias de prueba.

#### 4.1.1. Análisis Estadístico

A continuación se presenta el análisis estadístico de las mediciones tomadas para el sistema de medición de resistencia basado en la técnica Lock-in y el microcontrolador MSP432P401.

Cada una de las 10 resistencias de prueba (8 resistencias de carbón y 2 conductores, Fig. 4.1) fue medida 10 veces, durante el experimento se midió a su vez el voltaje  $V_{ADC}$ , el valor de la conversión digital leída directamente de Code Composer así como el valor de la resistencia de prueba.

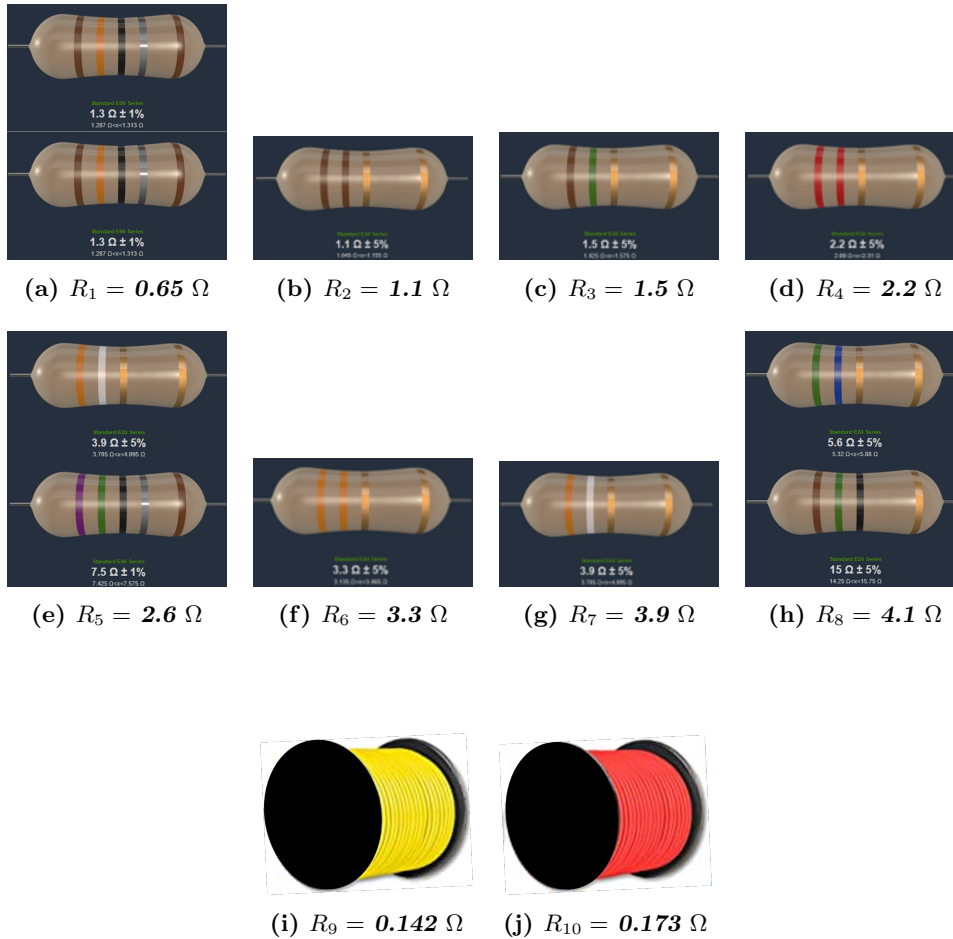
De forma que con dichas mediciones se hace el cálculo del promedio, el error experimental y la desviación estándar para evaluar el desempeño de la arquitectura propuesta.

**Promedio** La tabla 4.2 muestra los promedios de las mediciones experimentales del voltaje de entrada al ADC del microcontrolador ( $V_{ADC}$ ), también el valor promedio de las conversiones realizadas por el ADC usando el código desarrollado en la Sección 3.1.3 y finalmente el valor promedio de la resistencia medida considerando las ecuaciones de diseño descritas en la Sección 3.1.

Es importante mencionar que para las dos últimas muestras, los conductores amarillo (0.142  $\Omega$ ) y rojo (0.173  $\Omega$ ), el error de ganancia del ADC que se describió en la sección de simulaciones del capítulo anterior 3.1.4, influye en la conversión del voltaje  $V_{ADC}$  y por lo tanto en el cálculo de  $R_x$ <sup>2</sup>.

<sup>2</sup>Los renglones con las notas *conductor amarillo* y *conductor rojo* se refieren a los datos experimentales con el error de ganancia del ADC, aplica para las tablas 4.2, 4.3, 4.4 y 4.5.

#### 4. MEDICIONES Y ANÁLISIS DE RESULTADOS



**Figura 4.1:** Resistencias de prueba.

Para verificar el desempeño del circuito analógico, es considerando el caso ideal en el que el ADC realiza una conversión sin el error de ganancia<sup>3</sup>, con el valor teórico de conversión se hace el cálculo de la resistencia  $R_x$ , los resultados muestran que el promedio coincide con el cálculo teórico de la resistencia del conductor a diferencia del promedio experimental que refleja el error de ganancia del ADC del microcontrolador.

**Error Experimental** La tabla 4.3 muestra el error experimental calculado con la ecuación 2.11. Como se puede observar en general el error es menor al 10% para el

<sup>3</sup>Renglones con nota *conductor amarillo teórico* y *conductor rojo teórico*, aplica para las tablas 4.2, 4.3, 4.4 y 4.5.

Valor Teórico [ $\Omega$ ]	Caso I			Caso II		
	$V_{ADC}$ [V]	#Conv ADC	$R_x$ [ $\Omega$ ]	$V_{ADC}$ [V]	#Conv ADC	$R_x$ [ $\Omega$ ]
0.65	1.890	9,562	0.715	1.454	7,295	0.607
1.1	2.032	10,295	1.126	1.314	6,673	0.957
1.5	2.201	11,135	1.607	1.162	5,421	1.697
2.2	2.435	12,361	2.338	0.909	4,763	2.083
2.6	2.584	13,082	2.735	0.779	3,699	2.699
3.4	2.787	14,100	3.346	0.524	2,525	3.431
3.9	3.061	15,488	4.159	0.279	1,412	4.030
4.1	3.118	15,739	4.301	0.249	1,265	4.143
0.142 (conductor amarillo)	1.729	8,748	0.233	1.629	8,249	0.054
0.142 (conductor amarillo teórico)	1.729	8,584	0.139	1.629	8,089	0.146
0.173 (conductor rojo)	1.739	8,801	0.262	1.618	8,186	0.090
0.173 (conductor rojo teórico)	1.739	8,635	0.168	1.618	8,031	0.179

**Tabla 4.2:** Promedio de mediciones del LIA con MSP432P401R.

Caso I y menor al 13.10% en el Caso II, esto sin considerar las mediciones de los conductores de prueba amarillo y rojo.

Para las mediciones de los conductores se puede observar nuevamente que el error de ganancia del ADC provoca que el error experimental en la medición sea de casi 64%, sin embargo, considerando solo el acoplamiento de la señal analógica y suponiendo una conversión ideal del ADC, el error de dichas mediciones se reduce teóricamente de 2.47% a 9.17%. Esto se debe principalmente a la magnitud de las mediciones tomadas ya que una diferencia de décimas de ohm son significativas en las mediciones de los conductores.

**Desviación Estándar** Por otra parte, para evaluar la dispersión de las mediciones se hace el cálculo de la desviación estándar, tabla 4.4. De los datos obtenidos se puede decir que en el Caso I y el Caso II, la desviación de la medición del voltaje  $V_{ADC}$  es del orden de mV lo que es bueno al considerar que la resolución del multímetro utilizado es de 1 mV.

Por su parte, la desviación del valor de la conversión analógica-digital es menor en el Caso II ya que para las mediciones realizadas en el Caso I, en específico para las resistencias de 1.1  $\Omega$ , 2.2  $\Omega$  y 3.4  $\Omega$ , la desviación es mayor a 100 LSBs es decir la conversión del voltaje  $v_{ADC}$  resulta tener una variación significativa de 20 mV.

Por su parte, la desviación que tiene el valor experimental de la resistencia  $R_x$  es de 10.5 m $\Omega$  a 319 m $\Omega$ . Donde las mediciones del Caso I para los resistores de 1.1  $\Omega$ , 2.2  $\Omega$  y 3.4  $\Omega$  presentan mayor variación.

Para el caso de los conductores, las mediciones indican que no hay una gran variación

#### 4. MEDICIONES Y ANÁLISIS DE RESULTADOS

---

Valor Teórico [ $\Omega$ ]	Caso I [%]	Caso II [%]
0.65	10.06	7.16
1.1	4.60	12.96
1.5	7.14	13.10
2.2	6.64	5.30
2.6	5.21	3.82
3.4	4.18	1.29
3.9	6.64	3.33
4.1	4.91	1.53
0.142 (conductor amarillo)	63.88	62.24
0.142 (conductor amarillo teórico)	2.48	2.56
0.173 (conductor rojo)	51.38	48.13
0.173 (conductor rojo teórico)	9.17	7.74

**Tabla 4.3:** Error Experimental de mediciones del LIA con MSP432P401R.

Valor Teórico [ $\Omega$ ]	Caso I			Caso II		
	$V_{ADC}[V]$	#Conv ADC	$R_x[m\Omega]$	$V_{ADC}[V]$	#Conv ADC	$R_x[m\Omega]$
0.65	$516.4 \times 10^{-6}$	18	10.5	$737.9 \times 10^{-6}$	52	30.2
1.1	$2.4 \times 10^{-3}$	118	68.7	$2.0 \times 10^{-3}$	26	14.8
1.5	$1.1 \times 10^{-3}$	32	18.4	$1.9 \times 10^{-3}$	48	28.0
2.2	$4.2 \times 10^{-3}$	149	87.6	$1.6 \times 10^{-3}$	80	47.0
2.6	$1.5 \times 10^{-3}$	18	10.8	$2.8 \times 10^{-3}$	36	21.0
3.4	$61.9 \times 10^{-3}$	319	188.5	$4.4 \times 10^{-3}$	68	40.9
3.9	$9.2 \times 10^{-3}$	47	28.1	$9.2 \times 10^{-3}$	36	21.4
4.1	$2.8 \times 10^{-3}$	23	13.4	$3.6 \times 10^{-3}$	92	55.2
0.142 (conductor amarillo)	$994.4 \times 10^{-6}$	14	8.0	$788.8 \times 10^{-6}$	11	6.5
0.142 (conductor amarillo teórico)	$994.4 \times 10^{-6}$	5	2.8	$788.8 \times 10^{-6}$	4	2.3
0.173 (conductor rojo)	$6.1 \times 10^{-3}$	31	17.9	$6.0 \times 10^{-3}$	31	18.0
0.173 (conductor rojo teórico)	$6.1 \times 10^{-3}$	30	17.3	$6.0 \times 10^{-3}$	30	17.3

**Tabla 4.4:** Desviación Estándar de mediciones del LIA con MSP432P401R.

en las mediciones e incluso considerando el error del ADC la desviación es de 8 m $\Omega$  para el conductor amarillo y de 18 m $\Omega$  para el conductor rojo.

**Incertidumbre** Como parte del análisis de los resultados y por la naturaleza experimental de este trabajo, el análisis de la incertidumbre es una tarea imprescindible. La incertidumbre permite determinar la veracidad de las mediciones que el instrumento puede realizar. Su cálculo requiere conocer el modelo matemático de la variable de

interés e idealmente todas las fuentes de error. En experimentos simples es sencillo analizar todas las variables y proporcionar la precisión de una medición.

Sin embargo, en este trabajo hay que considerar que el modelo matemático obtenido es complejo y depende de muchas variables, tal como se observa en la ecuación 3.18. En consecuencia, analizar adecuadamente la contribución de cada componente resultaría impráctico por la complejidad del modelo matemático. Además un correcto cálculo de incertidumbre demanda analizar factores externos que también afecten la medición, y puesto que hacer una lista exhaustiva requiere gran experiencia, se concluye que no es viable hacer un análisis de incertidumbre como lo propone la literatura (26).

No obstante, una opción viable para analizar la incertidumbre del instrumento es la desviación estándar de las mediciones ya calculada. De esta forma se conoce indirectamente la precisión de las mediciones, por lo tanto se puede considerar que la desviación estándar representa la suma de todas fuentes de error.

Por lo tanto, la desviación estándar puede ser considerada como la incertidumbre absoluta de las mediciones tomadas. Por otra parte, la tolerancia (incertidumbre relativa) es posible calcularla a partir del porcentaje que representa dicha desviación, la tabla 4.5 muestra la tolerancia para cada elemento medido.

Valor teórico [ $\Omega$ ]	CASO I		CASO II	
	$\sigma$ [m $\Omega$ ]	Tolerancia [%]	$\sigma$ [m $\Omega$ ]	Tolerancia [%]
0.65	10.5	1.62	30.2	4.65
1.1	68.7	6.25	14.8	1.35
1.5	18.4	1.23	28.0	1.87
2.2	87.6	3.98	47.0	2.14
2.6	10.8	0.42	21.0	0.81
3.4	188.5	5.54	40.9	1.20
3.9	28.1	0.72	21.4	0.55
4.1	13.4	0.33	55.2	1.35
0.142	8.0	5.63	6.5	4.58
0.14 (Conductor amarillo teórico)	2.8	1.97	2.3	1.62
0.173	17.9	10.35	18	10.40
0.173 (Conductor rojo teórico)	17.3	10.00	17.3	10.00

**Tabla 4.5:** Tolerancia de las mediciones hechas con el sistema basado en Lock-in.

#### 4.1.2. Arquitectura final con técnica Lock-in y MSP432

Como se ha presentado, la medición de resistencia basada en la técnica Lock-in (A.5.2) es eficiente para resistencias de pequeñas. Durante el desarrollo de estas pruebas se encontró que el acoplamiento analógico de la señal es importante, porque si la señal no es

## 4. MEDICIONES Y ANÁLISIS DE RESULTADOS

---

estable y constante el análisis digital entrega resultados incoherentes. Esta consideración provocó principalmente el diseño del circuito Schmitt Trigger (A.5.1).

Por otra parte, de los resultados obtenidos, principalmente para los casos de los resistores de  $1.1 \Omega$ ,  $2.2 \Omega$  y  $3.4 \Omega$ , se pudo observar que presentan desviaciones importantes de los valores teóricos, esto se puede atribuir al error humano involucrado en la medición ya que en los otros casos (incluso con valores de resistencia menores) la desviación de los resultados es más aceptable. La tabla 4.6 resume las especificaciones del sistema.

Parámetro	Valor	Unidades
Fuente de alimentación	+/- 5 (AI) +3.3 (Convertidor Bi-Unipolar)	V
Frecuencia de operación ADC ( $f_s$ )	12	MHz
Voltaje ADC (in)	0 - 3.3	V
Frecuencia de conmutación de terminales digitales	0.3	Hz
Rango	0 - 3.395	$\Omega$
Resolución (1 LSB)	0.414	m $\Omega$
Tolerancia	+/- 4.65	%

**Tabla 4.6:** Especificaciones del sistema de medición basado en Lock-in.

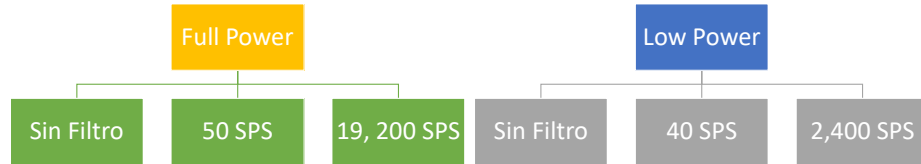
### 4.2. Resultados del sistema de medición con ADC7124-8 y Raspberry PiPico

A continuación se presentarán las mediciones obtenidas con el sistema de medición basado en el ADC7124-8, la configuración y desarrollo del código así como las conexiones y la arquitectura es descrita detalladamente en la Sección 3.3.

Para la prueba se debe recordar que el ADC7124-8 tiene diferentes modos de operación, *Full*, *Mid* y *Low* y diferentes opciones de filtrado digital, aquí se plantearon varios escenarios, la Fig. 4.2 los muestra. Como se observa se realizaron seis pruebas diferentes, para configurar las características del ADC se cambiaron los parámetros mencionados en el registro de filtrado como lo indica la hoja de especificaciones así como el modo en el control de registro.

Adicionalmente, se propone una rutina de lectura que realiza 20-mediciones por canal automáticamente mostrando los datos en la terminal serial, posteriormente dichos datos son procesados en Excel.





**Figura 4.2:** Casos de medición con ADC7124-8 y Raspberry Pi Pico.

Para calificar el desempeño del sistema de medición, se calcularon los promedios, el error experimental y la desviación estándar de las mediciones para cada canal. Para el experimento se consideraron los siguientes valores de prueba:  $Channel_0 - 100 \Omega$ ,  $Channel_1 - 248.12 \Omega$ ,  $Channel_2 - 50 \Omega$ ,  $Channel_2 - 248.12 \Omega$  y  $Channel_4 - 150 \Omega$ .

#### 4.2.1. Análisis Estadístico

**Promedio.** Como se observa en la tabla 4.7 cuando el ADC es configurado para el modo *Low Power* las mediciones se acercan mas al valor teórico, de forma similar los canales para el RTD 2 y 4 parecen ser los que se acercan más al valor real.

RTD	Valor Teórico [ $\Omega$ ]	Conversión Continua Full Power			Conversión Continua Low Power		
		S/Filtro [ $\Omega$ ]	50 SPS [ $\Omega$ ]	19,200 SPS [ $\Omega$ ]	S/Filtro [ $\Omega$ ]	40 SPS [ $\Omega$ ]	2,400 SPS [ $\Omega$ ]
RTD.0	100	103.895	103.899	103.899	101.603	101.607	101.605
RTD.1	248.12	256.109	256.154	256.154	250.505	250.555	250.551
RTD.2	50	52.115	52.139	52.140	50.963	50.986	50.987
RTD.3	248.12	255.978	256.023	256.016	250.372	250.413	250.421
RTD.4	150	154.231	154.272	154.266	150.832	150.866	150.872

**Tabla 4.7:** Promedio de mediciones del ADC7124-8 con Raspberry PiPico.

**Error experimental.** El error experimental de las mediciones se calcula con la ecuación 2.11. Por su parte la tabla 4.8 muestra el error experimental y como se puede apre-

## 4. MEDICIONES Y ANÁLISIS DE RESULTADOS

---

ciar en todos los casos de medición es menor al 5% lo que indica que las mediciones realizadas son aceptables. Mas aún, el canal 4 muestra el error más bajo en comparación con los demás con uno de 0.55457%. Cabe mencionar que la configuración del modo de conversión continua en *Low Power* sin filtrado es la que muestra en general un error de medición menor a 2% en todos los canales.

Error	Conversión Continua Full Power			Conversión Continua Low Power		
	S/Filtro [%]	50 SPS [%]	19,200 SPS [%]	S/Filtro[%]	40 SPS [%]	2,400 SPS [%]
RTD_0	3.895	3.900	3.899	1.603	1.607	1.605
RTD_1	3.200	3.238	3.238	0.961	0.981	0.980
RTD_2	4.230	4.278	4.279	1.926	1.972	1.973
RTD_3	3.167	3.185	3.182	0.908	0.924	0.927
RTD_4	2.821	2.848	2.844	0.555	0.578	0.581

**Tabla 4.8:** Error experimental de mediciones del ADC7124-8 con Raspberry PiPico.

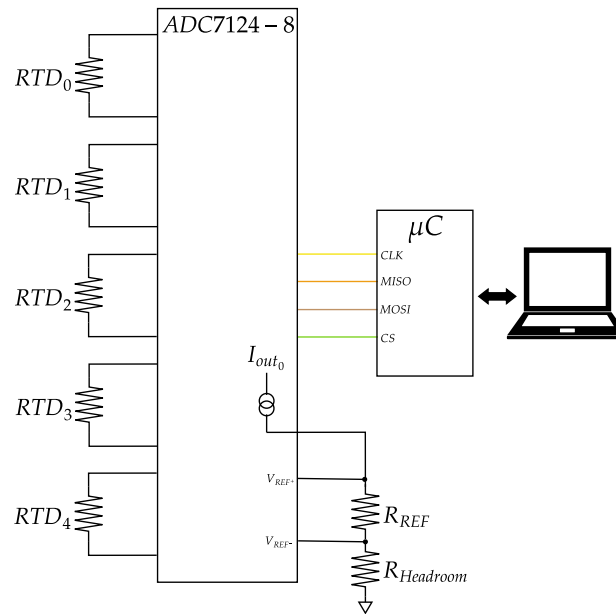
**Desviación estándar.** Finalmente la tabla 4.9 muestra la dispersión de los resultados, aquí se puede apreciar que en general las mediciones varían en el orden de  $m\Omega$ , sin embargo la dispersión es menor cuando se ocupa el modo de baja potencia en especial en los canales 0 y 2. El canal 2 tiene la desviación estándar mas baja que es de 0.7  $m\Omega$ . De forma general se puede decir que al usar el modo *Full Power* la desviación estándar disminuye al aumentar la tasa de filtrado, sin embargo ocurre lo contrario en el modo *Low Power*, mientras se aumenta la tasa de filtrado la dispersión de los datos aumenta.

Desviación Estándar	Conversión Continua Full Power			Conversión Continua Low Power		
	S/Filtro [ $m\Omega$ ]	50 SPS [ $m\Omega$ ]	19,200 SPS [ $m\Omega$ ]	S/Filtro[ $m\Omega$ ]	40 SPS [ $m\Omega$ ]	2,400 SPS [ $m\Omega$ ]
RTD_0	8.797	1.314	2.705	0.772	4.776	4.542
RTD_1	5.808	4.613	9.326	2.224	4.300	11.703
RTD_2	1.039	0.789	1.493	0.768	1.334	2.257
RTD_3	7.335	4.594	3.597	3.207	9.831	15.538
RTD_4	7.001	4.130	3.572	2.375	4.974	7.896

**Tabla 4.9:** Desviación Estándar de mediciones del ADC7124-8 con Raspberry PiPico.

### 4.2.2. Arquitectura final del ADC7124-8 con Raspberry PiPico

La Fig. 4.3 muestra la arquitectura completa de este experimento, la cual incluye las diferentes resistencias de prueba  $RTD_0, \dots, RTD_4$ , el ADC7124-8 y el microcontrolador Raspberry Pi Pico.



**Figura 4.3:** Arquitectura para medición con el ADC7124-8 y su conexión SPI con el microcontrolador a utilizar.

### 4.3. Resultados del sistema de medición con ADC7124-8 y ESP32

Para las pruebas en el microcontrolador ESP32, se propusieron los siguientes escenarios de medición.

1. ADC *Low Power* sin opciones de filtrado (20 muestras).
2. ADC *Full Power* sin opciones de filtrado (código para los puntos 1 y 2 disponibles en este [GitHub](#) - 20 muestras).
3. ADC *Low Power* con filtro para 2,400 SPS (50 muestras).
4. ADC *Low Power* con filtro para 40 SPS (50 muestras).
5. ADC *Full Power* con filtro para 19,200 SPS (50 muestras).
6. ADC *Full Power* con filtro para 50 SPS (código para los puntos 3 a 6 disponibles en este [GitHub](#)- 50 muestras).

## 4. MEDICIONES Y ANÁLISIS DE RESULTADOS

---

7. Configuración de ADC fuera de bucle infinito para el modo *Full Power* con filtro para 19,200 SPS que cambia el tiempo entre la activación del canal y la lectura (20 muestras).
8. Configuración de ADC en el bucle infinito para el modo *Full Power* con filtro para 19,200 SPS que cambia el tiempo entre la activación del canal y la lectura (código para los puntos 7 y 8 esta disponible en este [GitHub](#)- 20 muestras).

### 4.3.1. Análisis Estadístico

**Promedio** La tabla 4.10 muestra los promedios de las mediciones de los puntos 1 a 6 de los escenarios planteados. Como se puede apreciar, el modo *Low Power* es el que se aproxima más al valor teórico ya que la diferencia entre el valor medido y el teórico es en algunos casos menor a 1  $\Omega$  y como máximo 2.204  $\Omega$  a diferencia del caso de conversión en el modo *Full Power* que llega a ser de hasta 7.809  $\Omega$ .

Promedio	Valor Teórico [ $\Omega$ ]	Conversión Continua Full Power			Conversión Continua Low Power		
		S/Filtro [ $\Omega$ ]	50 SPS [ $\Omega$ ]	19,200 SPS [ $\Omega$ ]	S/Filtro [ $\Omega$ ]	40 SPS [ $\Omega$ ]	2,400 SPS [ $\Omega$ ]
RTD_0	100	101.579	103.875	103.875	101.578	101.578	101.578
RTD_1	10	10.174	10.405	10.405	10.174	10.174	10.174
RTD_2	50	50.799	51.949	51.950	50.799	50.799	50.799
RTD_3	248.12	250.341	255.929	255.927	250.334	250.332	250.324
RTD_4	150	150.920	154.295	154.297	150.909	150.905	150.900

**Tabla 4.10:** Promedios de mediciones del ADC7124-8 con ESP32.

**Error Experimental** El error en las mediciones es calculado con la ecuación 2.11. En los resultados (para los escenarios 1 a 6 planteados) presentados en la tabla 4.11 se puede observar que también el modo de baja potencia, a pesar de la activación de las opciones de filtrado, tiene en general un menor error experimental, especialmente los canales 3 y 4. Cuando en el canal se activó el filtrado de 2,400 SPS se obtuvo un error experimental de aproximadamente 0.1296 %.

Error	Conversión Continua Full Power			Conversión Continua Low Power		
	S/Filtro [%]	50 SPS [%]	19,200 SPS [%]	S/Filtro [%]	40 SPS [%]	2,400 SPS [%]
RTD_0	1.579	3.875	3.875	1.578	1.578	1.578
RTD_1	1.739	4.045	4.046	1.740	1.739	1.741
RTD_2	1.599	3.898	3.900	1.598	1.597	1.599
RTD_3	0.136	2.376	2.371	0.134	0.133	0.130
RTD_4	0.613	2.864	2.864	0.606	0.603	0.600

**Tabla 4.11:** Errores experimentales de mediciones del ADC7124-8 con ESP32.

**Desviación Estándar** Finalmente, analizando la desviación de los resultados mostrados en la tabla 4.12, se puede decir que, el modo *Full Power* mejora la dispersión de los resultados. Con excepción del caso del RTD\_1 en el cual la menor dispersión se midió al usar el modo *Low Power*. Sin embargo, por su parte, los canales 0 y 2 fueron los que tuvieron una menor dispersión de los resultados.

Desviación Estándar	Conversión Continua Full Power			Conversión Continua Low Power		
	S/Filtro [mΩ]	50 SPS [mΩ]	19,200 SPS [mΩ]	S/Filtro [mΩ]	40 SPS [mΩ]	2,400 SPS [mΩ]
RTD_0	0.522	0.986	0.729	0.613	1.054	1.107
RTD_1	0.266	0.291	0.327	0.261	0.902	0.815
RTD_2	0.740	0.780	1.094	0.826	0.998	1.296
RTD_3	1.970	3.384	3.151	4.721	2.919	3.835
RTD_4	2.161	3.615	5.490	6.362	3.880	4.969

**Tabla 4.12:** Desviación estándar de mediciones del ADC7124-8 con ESP32.

### 4.3.2. Variación del tiempo de espera entre mediciones

Una característica de interés para evaluar el desempeño del ADC es el tiempo de respuesta se asentamiento del filtro digital.

Según la hoja de datos (23) el tiempo de asentamiento depende de la frecuencia de conversión e indirectamente del modo de operación (*Full* o *Low*). De la hoja de especificaciones (23) se toma la ecuación 4.2

$$t_{settle} = \frac{(4 \times 32 \times FS) + DeadTime}{f_{CLK}} \quad (4.2)$$

en donde  $FS$  es el valor del registro de filtrado.

El tiempo  $DeadTime$  depende del valor de  $FS$ , si  $FS > 1$  entonces  $DeadTime = 95$ , si  $FS = 1$  entonces  $DeadTime = 61$ .

El valor de la frecuencia del reloj depende del modo de operación, en donde para *Full Power* es de 614.4 kHz y para *Low Power* es de 76.8 kHz.

Considerando entonces la ecuación 4.2 y los parámetros descritos, se calcula el tiempo de asentamiento del filtro digital para 50 SPS, 19,200 SPS, 40 SPS, y 2,400 SPS, Tab. 4.13. Este es el tiempo que se debe esperar entre la configuración y una conversión. Para simplificar la prueba, se configuró el filtro de 19,200 SPS del ADC y se propone un rango de espera entre configuración de 0 s a 1 s en pasos de 0.1 s. Los siguientes gráficos muestran cómo varían los resultados por cada intervalo y para cada canal.

Es importante mencionar que durante las mediciones, cuando el tiempo es 0 s las lecturas eran incoherentes mostrando incluso valores negativos (para la medición de

## 4. MEDICIONES Y ANÁLISIS DE RESULTADOS

---

Modo		FS	$t_{settle}$
Full	50sps	384	307.6 $\mu$ s
Full	19,200sps	1	80.2 ms
Low	40sps	60	2.5 ms
Low	2,400sps	1	101.2 ms

**Tabla 4.13:** Tiempo de asentamiento del filtro digital para ADC7124-8.

resistencias) por lo que se decidió omitir los datos.

Además, se planteó una variante a esta prueba, la primera parte muestra los resultados del ADC configurado solamente antes del comienzo del bucle infinito, la segunda parte muestra los resultados de configurar el ADC al inicio del bucle principal, lo que significa reconfigurar el ADC cada vez que se realiza una nueva medición.

Como se observa en el siguiente extracto de código, la configuración del ADC se coloca fuera del bucle principal con la función `RTD.ADC_Init()`, adicionalmente, la variable “*wait*” aumenta su valor de 0 s a 1 s en pasos de 0.1 s y con ello el tiempo de espera entre configuración y medición.

```
...
RTD.ADC_Init()
...
while True:
...
for x in range(5):
print('RTD no. ', x)
RTD.Sel_Channel(x)      #current activation
sleep(wait)
print(RTD.Resistance_conversion(RTD.Read_ADC()))
```

Para la segunda prueba, la configuración del ADC, con la función `RTD.ADC_Init()`, se coloca dentro del bucle principal; en este caso se utiliza la misma variable “*wait*” para hacer el cambio de tiempo entre mediciones.

```
...
while True:
...
RTD.ADC_Init()
...
```

```
for x in range(5):
    print('RTD no. ', x)
    RTD.Sel_Channel(x)      #current activation
    sleep(wait)
    print(RTD.Resistance_conversion(RTD.Read_ADC()))
```

### 4.3.2.1. Configuración del ADC fuera del bucle.

Como se ha explicado, se hicieron dos pruebas. A continuación las tablas [4.14](#), [4.15](#), [4.16](#), [4.17](#) y [4.18](#) muestran los resultados obtenidos para cuando el ADC es configurado una sola vez antes de iniciar el bucle infinito.

Como se puede observar en las mediciones, no hay un patrón específico que indique que aumentando el tiempo entre la medición y la configuración mejore el desempeño de los filtros y el ADC en general, sin embargo, la desviación estándar sí es relativamente menor cuando aumenta el tiempo de espera entre la configuración y la medición. Lo que queda claro es que el tiempo entre la configuración del ADC y la primer medición si debe ser mayor a 0 s. En los gráficos se puede observar la desviación estándar ( $(+/-)$  *Diferencia* [mΩ]) en las mediciones con respecto al promedio para cada tiempo.

### 4.3.2.2. Configuración del ADC dentro del bucle.

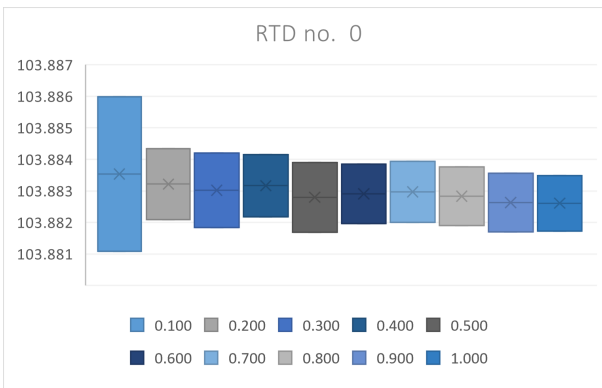
Por otra parte, en esta prueba se repitió la secuencia de mediciones cambiando el lugar de activación y configuración del ADC, en este caso las tablas [4.19](#), [4.20](#), [4.21](#), [4.22](#) y [4.23](#) muestran los resultados.

En esta prueba, de forma similar, no queda claro si aumentar el tiempo de espera entre la configuración del ADC y toma de la muestras es clave para mejorar el rendimiento del ADC incluso si se reinicia cada vez que termina el ciclo de toma de muestras. Lo que si queda claro es que el tiempo de espera tiene que ser mayor a 0 s.

### 4.3.3. Arquitectura final del ADC7124-8 con ESP32

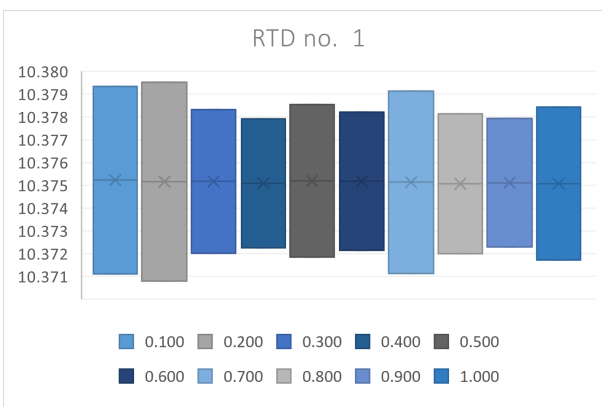
A continuación se presenta la arquitectura (Fig. [4.3](#)) completa que incluye la conexión de las resistencias de prueba y el microcontrolador, cabe mencionar que a pesar de proponer diferentes escenarios de prueba todos ocuparon la misma arquitectura ya que las configuraciones de filtrado, retardo en activación de los canales y la conversión se realizaron directamente en el código desarrollado en Micropython.

## 4. MEDICIONES Y ANÁLISIS DE RESULTADOS



Tiempo de espera [s]	Diferencia [ $m\Omega$ ]
0.10	4.899
0.20	2.259
0.30	2.358
0.40	1.973
0.50	2.213
0.60	1.895
0.70	1.938
0.80	1.861
0.90	1.858
1.00	1.756

**Tabla 4.14:** RTD\_0 para configuraciones de ADC fuera del bucle.



Tiempo de espera [s]	Diferencia [ $m\Omega$ ]
0.10	8.217
0.20	8.715
0.30	6.302
0.40	5.653
0.50	6.679
0.60	6.063
0.70	7.991
0.80	6.127
0.90	5.626
1.00	6.716

**Tabla 4.15:** RTD\_1 para configuraciones de ADC fuera del bucle.

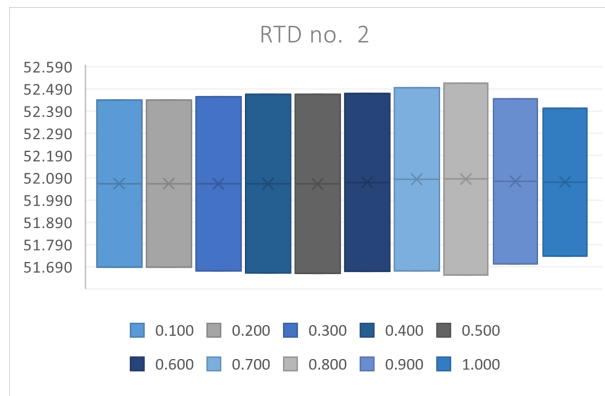
Esto presentó una gran ventaja ya que se pudieron probar diferentes características del ADC sin necesidad de modificar las conexiones lo que da robustez al hardware y libertad al software.

### 4.4. Áreas de aplicación

#### 4.4.1. Bioimpedancia

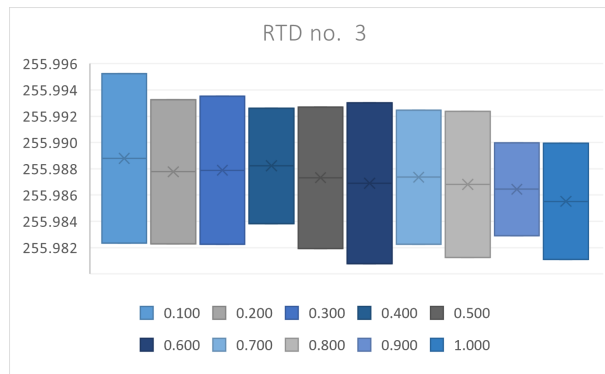
La bioimpedancia es la impedancia en el cuerpo humano u otros materiales biológicos, su estudio consiste en desarrollar técnicas de medición además de analizar su compor-





Tiempo de espera [s]	Diferencia [mΩ]
0.10	751.805
0.20	752.352
0.30	784.424
0.40	803.340
0.50	805.869
0.60	798.803
0.70	825.572
0.80	861.401
0.90	741.233
1.00	664.761

**Tabla 4.16:** RTD\_2 para configuraciones de ADC fuera del bucle.



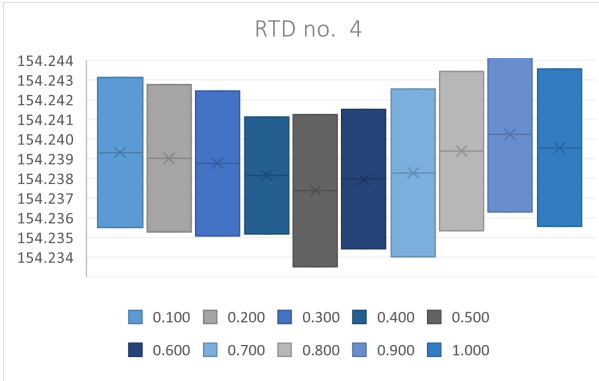
Tiempo de espera [s]	Diferencia [mΩ]
0.10	12.899
0.20	10.973
0.30	11.291
0.40	8.782
0.50	10.759
0.60	12.248
0.70	10.203
0.80	11.133
0.90	7.075
1.00	8.839

**Tabla 4.17:** RTD\_3 para configuraciones de ADC fuera del bucle.

tamiento bajo diferentes circunstancias (27).

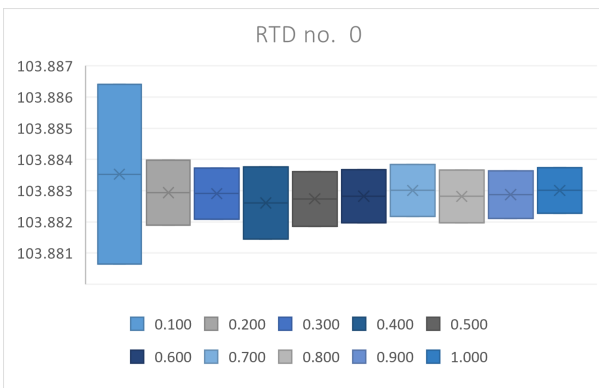
El campo de investigación de la bioimpedancia es muy amplio y abarca una gran diversidad de aplicaciones que son actualmente investigadas por diversos grupos científicos. Algunas aplicaciones consisten en caracterizaciones del cuerpo humano como la medición del dolor en pacientes clínicos (28), o por ejemplo en el diseño sistemas de medición para obtener modelos eléctricos de células (29), (30), o para medir la calidad del agua (31), etc. (27). En cualquiera de los casos la bioimpedancia es el parámetro por caracterizar para dichas aplicaciones.

## 4. MEDICIONES Y ANÁLISIS DE RESULTADOS



Tiempo de espera [s]	Diferencia [mΩ]
0.10	7.634
0.20	7.499
0.30	7.351
0.40	5.931
0.50	7.729
0.60	7.081
0.70	8.524
0.80	8.105
0.90	7.864
1.00	7.992

**Tabla 4.18:** RTD\_4 para configuraciones de ADC fuera del bucle.



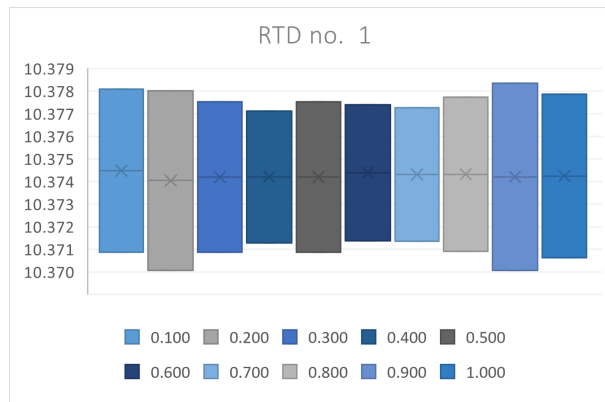
Tiempo de espera [s]	Diferencia [mΩ]
0.10	5.755
0.20	2.091
0.30	1.632
0.40	2.307
0.50	1.758
0.60	1.705
0.70	1.665
0.80	1.697
0.90	1.530
1.00	1.468

**Tabla 4.19:** RTD\_0 para configuraciones de ADC dentro del bucle.

### 4.4.1.1. Medición del dolor

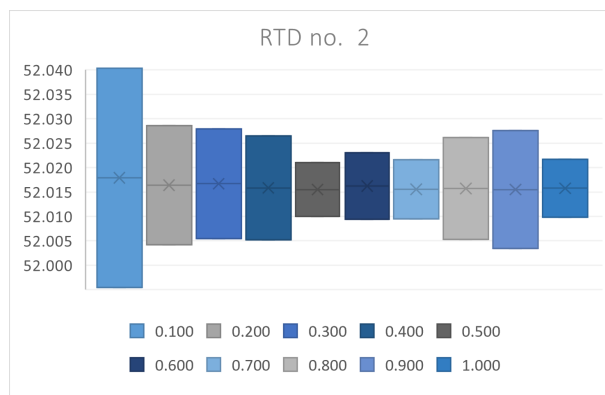
La propuesta de Ghita et al. (28) presenta un sistema de monitoreo del dolor en pacientes clínicos. La idea está basada en la medición de la impedancia de la piel, la cual es usada como un indicador indirecto de diferentes características del cuerpo humano.

La relación dolor-impedancia se entiende por la conexión que tienen los nervios y la piel y por la actividad eléctrica en la piel que tiene relación con diferentes síntomas como el estrés mental, las glándulas sudorosas o el dolor, todos estos provocan de alguna manera un cambio en la impedancia que puede ser medida y analizada a través de diversas señales eléctricas que se caracterizan para obtener modelos matemáticos que describan la respuesta del cuerpo humano, en este caso su respuesta a estímulos



Tiempo de espera [s]	Diferencia [mΩ]
0.10	7.225
0.20	7.936
0.30	6.651
0.40	5.838
0.50	6.666
0.60	6.022
0.70	5.904
0.80	6.834
0.90	8.292
1.00	7.241

**Tabla 4.20:** RTD\_1 para configuraciones de ADC dentro del bucle.



Tiempo de espera [s]	Diferencia [mΩ]
0.10	44.858
0.20	24.388
0.30	22.517
0.40	21.246
0.50	11.087
0.60	13.682
0.70	12.172
0.80	20.837
0.90	24.126
1.00	11.850

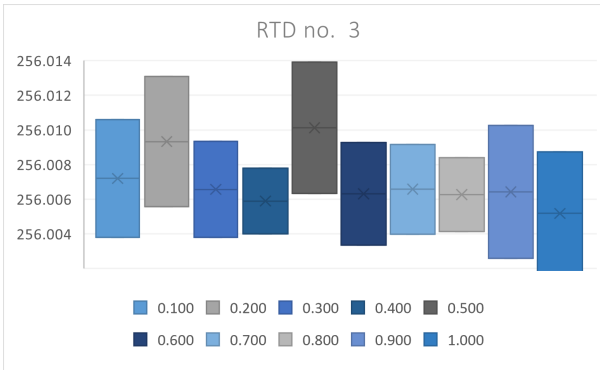
**Tabla 4.21:** RTD\_2 para configuraciones de ADC dentro del bucle.

dolorosos.

En el diseño del sistema de medición se considera el rango de frecuencias en las que responden las diferentes capas de la piel. Son consideradas así, la epidermis (capa superficial) dominada por respuestas a bajas frecuencias (por debajo de 1 kHz) y la dermis en la que dominan frecuencias altas (por encima de 100 kHz).

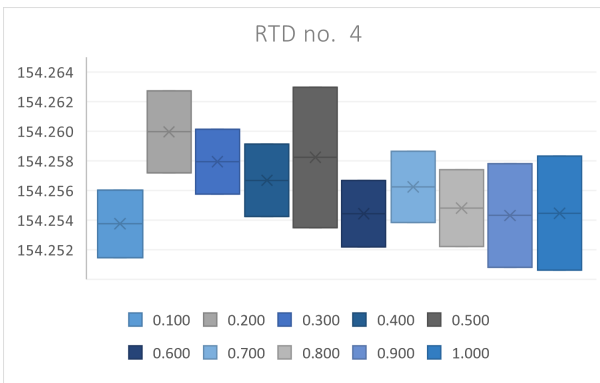
Los resultados mostrados en el estudio (28) demuestran que existe una relación entre la impedancia de la piel y el dolor. También se indica que al aplicar un estímulo doloroso aumenta la impedancia de la piel y que existe un estado de memoria, ya que al dejar de aplicar el estímulo doloroso la impedancia no vuelve a los valores iniciales (al estado de no dolor). Este fenómeno lo encontraron los autores en las dos pruebas de concepto, cuando se aplicó un estímulo doloroso mecánico y cuando se aplicó uno térmico.

#### 4. MEDICIONES Y ANÁLISIS DE RESULTADOS



Tiempo de espera [s]	Diferencia [mΩ]
0.10	6.793
0.20	7.502
0.30	5.553
0.40	3.786
0.50	7.580
0.60	5.906
0.70	5.178
0.80	4.258
0.90	7.672
1.00	7.093

**Tabla 4.22:** RTD\_3 para configuraciones de ADC dentro del bucle.



Tiempo de espera [s]	Diferencia [mΩ]
0.10	4.566
0.20	5.533
0.30	4.366
0.40	4.865
0.50	9.476
0.60	4.461
0.70	4.809
0.80	5.176
0.90	7.001
1.00	7.699

**Tabla 4.23:** RTD\_4 para configuraciones de ADC dentro del bucle.

Finalmente con los resultados que obtuvieron los autores, proponen el modelo matemático de la bioimpedancia de la piel que depende de diferentes parámetros complejos que varían en el dominio de la frecuencia. Ahora bien una de las conclusiones del estudio es que debido a las diferentes características físicas de los individuos como el género, la estatura, el índice de masa corporal, el peso, el porcentaje de grasa, etcétera, el dispositivo propuesto requiere una calibración por persona para poder identificar los umbrales de dolor, de memoria de dolor y los estados de no dolor.

#### 4.4.1.2. Modelos matemáticos de las células

Una aplicación más para la bioimpedancia es el análisis del comportamiento eléctrico de células mediante el desarrollo de los modelos matemáticos (29).

Las células en sí pueden considerarse como partículas eléctricas que pueden ser polarizadas al someterse a campos eléctricos. La polarización no es instantánea de forma que hay que esperar un tiempo para ver el efecto de la polarización en las partículas (células) al aplicar o quitar el campo eléctrico, a este fenómeno se le llama relajación dieléctrica y es caracterizado por una constante de tiempo  $\tau$ .

De forma que para describir este comportamiento matemáticamente, comúnmente se utiliza la técnica de espectroscopia de impedancia eléctrica. Los modelos de las partículas que bien pueden ser células biológicas, como bacterias o proteínas, permiten esbozar diferentes escenarios como la concentración molar, el crecimiento de células o la identificación y conteo de estas.

Otra técnica es analizar la distribución de los tiempos de relajación usada comúnmente en aplicaciones electroquímicas. En la aplicación de células suspendidas en microfluidos, esta técnica proporciona las constantes de tiempo  $\tau$  de los tiempos de relajación correspondientes a los tiempos de polarización de las partículas.

El estudio reportado en (29) lleva a cabo varias pruebas con diferentes muestras y las conclusiones llevan a diferentes modelos matemáticos los cuales en general demuestran que a mayor concentración de células se espera un mayor tiempo de relajación. También se muestra que la impedancia es directamente proporcional a la de la concentración de la solución.

Por otra parte en este mismo campo de la medición de impedancia eléctrica en células otro estudio (30) propone analizar partículas de 15 a 48 micrómetros, ya que este es el rango en el que se encuentran en general las células cancerígenas.

El trabajo se hace con el análisis de espectros de impedancia eléctrica que en combinación con la obtención del modelo del circuito eléctrico equivalente de la impedancia de la célula permiten caracterizar las partículas suspendidas en una solución electrolítica.

Los autores concluyen que al aumentar la concentración partículas de la solución, la resistencia disminuye. Por el contrario, la capacitancia aumenta al aumentar la concentración de la solución, este resultado permite plantear aplicaciones que utilicen el sistema de medición presentado para monitorear cambios en la concentración de partículas de forma no invasiva y exacta.

### 4.4.1.3. Calidad del agua

La contaminación de los diferentes cuerpos de agua de los que la sociedad depende es un tema de interés mundial (31). Mantener un constante monitoreo de la calidad del agua requiere de sensores que proporcionen información en tiempo real y que sean de bajo costo. Para esto se ha propuesto recientemente un sensor basado en la medición de la impedancia de soluciones electrolíticas.

El sistema de medición está basado en un amplificador Lock-in LIA que mide la parte real y compleja de la impedancia. Esto permite monitorear las propiedades eléctricas de la solución, una compuesta por agua y clorina y otra por agua y sal. Los resultados muestran que al aumentar la concentración de la clorina disminuye la impedancia de la solución esta relación se puede observar también para las mediciones de la solución de agua con sal en la que la impedancia igualmente disminuye.

Otra investigación llevada a cabo en la universidad de Wroclaw (32) muestra la relación que existe entre la impedancia eléctrica y la concentración de bio-organismos en las tuberías de agua y su relación con el material de las mismas. La presencia de estos microorganismos disminuye el nivel de la calidad del agua por lo tanto es importante monitorear el crecimiento y los efectos (como la corrosión) que éstas provocan en la degradación de los materiales de las tuberías.

La identificación de los organismos y bacterias presentes en el agua pueden ser hecha mediante cultivos, sin embargo, esta técnica requiere tiempo y no ofrece el espectro completo de los organismos presentes en las muestras. El estudio realizado presenta una técnica de monitoreo en tiempo real basada en la medición de la impedancia de la muestra.

El experimento demostró pequeñas variaciones en la impedancia durante los primeros días de medición, después se registró un aumento sistemático. Estos resultados reflejan el estado del sistema de distribución de agua en el que el número de bacterias aumenta según las condiciones ambientales ya que las mediciones de impedancia varían al comparar tuberías de diferentes materiales tales como polietileno (PE), acero, policloruro de vinilo (PVC), cobre y hierro.

Los resultados concluyen que el material que presenta un menor crecimiento bacteriano es el PVC en contraste al PE. La recomendación final es considerar el material de construcción de las tuberías para evitar así la contaminación del agua.

### 4.4.2. Sensores Resistivos de Temperatura RTD

Los sensores resistivos de temperatura (RTD) son dispositivos basados en diferentes materiales metálicos. Dado que existe una relación entre la resistencia de un conductor

y la temperatura, es posible medir indirectamente con gran exactitud y alta resolución la temperatura, esto se hace mediante la medición de alta resolución de la resistencia.

La propuesta (33) presenta un dispositivo *wearable* para el monitoreo de la temperatura corporal. Las mediciones son hechas con un sensor RTD que detecta pequeños cambios (incrementos o decrementos) en la temperatura (resistencia del sensor) de los pacientes.

Uno de los beneficios de usar este método es la alta linealidad de la relación entre la resistencia y la temperatura lo que permite fijar rangos de operación muy específicos y asegurar la precisión en las mediciones de resistencia.

La medición de la resistencia se hace mediante un puente de Wheatstone y un amplificador de instrumentación, ese sencillo sistema permite medir la resistencia de forma precisa lo que a su vez permite desarrollar un dispositivo *wearable* de bajo consumo energético y no invasivo.

#### 4.4.3. Medición de resistencia en conductores

Las características eléctricas de los conductores ayudan a entender su comportamiento y en su caso para el desarrollo de nuevas tecnologías. El estudio (34) muestra una técnica de medición de impedancia resistiva y capacitiva para obtener los parámetros eléctricos de un nano-conductor basado en gas en el cual la conductividad del conductor está relacionada con la concentración del gas.

Los autores proponen una arquitectura basada en un amplificador Lock-in, una señal de prueba pasa por el nano conductor (en forma de corriente), la señal es convertida a voltaje para ser amplificada y posteriormente filtrada. Teóricamente se espera que la impedancia del nano-conductor de paladio esté en el rango de las unidades y decenas de  $k\Omega$ .

Posteriormente se hace la multiplicación de la señal con la señal de referencia, el producto es filtrado para eliminar el ruido que experimentalmente se encontró era de 3 mVpp.

Finalmente, de los experimentos realizados los autores muestran una relación directamente proporcional entre la disminución del porcentaje de hidrógeno en el nano-conductor y la disminución de la impedancia del nano-conductor, dando como resultado un instrumento de medición con una resolución del 2% dentro de un rango de 10  $k\Omega$  a 40  $k\Omega$  para la impedancia resistiva y con una resolución de 3% para la impedancia capacitiva en un rango de 0.5 nF a 1.8 nF.

### 4.5. Discusión

En este capítulo se presentaron los resultados experimentales y el análisis estadístico de las diferentes pruebas hechas para ambos sistemas de medición, el basado en la técnica Lock-in y el basado en el convertido analógico digital ADC7124-8.

En ambos casos el cálculo del promedio, el error experimental y la desviación estándar proporcionaron información para evaluar el desempeño del sistema de medición, así como características importantes de considerar en el diseño.

Por ejemplo para la técnica lo Lock-in se descubrió que el acoplamiento analógico es muy importante y la propuesta sugerida es robusta, sin embargo, por su parte el procesamiento digital es igualmente importante ya que los errores encontrados en las mediciones se dieron en el tratamiento digital de la información y no en el procesamiento analógico de la señal.

Por su parte el sistema de medición basado en el ADC7124-8 tiene la particularidad de variar su desempeño de acuerdo con los filtros utilizados y las conexiones en los diferentes canales de medición.

Finalmente se presentó una sección de aplicaciones que mostró diferentes áreas de estudio utilizan la impedancia para caracterizar diferentes problemas y encontrar soluciones a ellos.



## Conclusiones

---

En este trabajo se expusieron dos arquitecturas para medir la parte real de la impedancia.

El primer diseño consistió en una arquitectura basada en un amplificador Lock-in y diferentes componentes analógicos para acoplar la señal además de un procesamiento digital para analizar las mediciones hechas por el ADC embebido en la tarjeta de desarrollo MSP432P401R.

De la arquitectura propuesta se debe destacar la robustez del acoplamiento analógico de la señal ya que se hace con componentes simples como resistencias y transistores, lo que aseguró la integridad de los componentes electrónicos como el amplificador de instrumentación y el microcontrolador de Texas Instruments en especial el ADC embebido en esta.

Del análisis matemático realizado para la arquitectura basada en el LIA, las ecuaciones de diseño (Schmitt Trigger, cambiador de nivel, ganancia AI y conversión del ADC) así como para el cálculo de la resistencia de prueba ( $R_x$ ) son el producto final de la etapa de diseño.

Es importante destacar que el diseño del LIA tuvo que considerar un elemento extra, las compuertas Schmitt Trigger que no estaban en el diseño original. La propuesta de las compuertas surgió al establecer el valor de los voltajes digitales en las terminales de la tarjeta de desarrollo, la asimetría y variación, afectaría directamente el cálculo de  $R_x$ . Las compuertas eliminaron este problema e hicieron más robusto el acoplamiento analógico de la señal y el circuito basado en transistores evitó el uso de otros circuitos integrados.

Una herramienta importante en el desarrollo de este trabajo fue la simulación en Multisim<sup>1</sup>, este software permitió evaluar los modelos matemáticos propuestos así como el

---

<sup>1</sup>Todas las simulaciones fueron realizadas con la versión educativa 14.3.0

## 5. CONCLUSIONES

---

comportamiento ideal del sistema antes de su construcción.

Primero se hicieron diferentes esquemas para verificar el comportamiento de la compuerta Schmitt Trigger y así validar las ecuaciones de diseño propuestas. En segundo lugar se simuló el comportamiento del convertidor bipolar o unipolar con el fin de asegurar la integridad del ADC de la tarjeta. Y en tercer lugar se hace la simulación completa del sistema para verificar el modelo matemático propuesto para el cálculo de la resistencia de prueba. Todas estas simulaciones proveyeron información valiosa sobre los parámetros de diseño y del comportamiento del sistema.

De forma que las simulaciones son indispensables para: 1) verificar las ecuaciones de diseño, 2) observar el comportamiento de los componentes y 3) hacer el planteamiento de diversos casos de prueba sin necesidad de construir los circuitos físicamente.

Después de obtener las ecuaciones de diseño, simular la arquitectura, y realizar el código para activar el ADC y calcular el valor de  $R_x$ . Se procedió a construir el sistema de medición físicamente. Para esto se construyeron por separado las diferentes etapas del sistema. Esto con el fin de proteger los componentes analógicos y para verificar individualmente el comportamiento de cada circuito.

Una vez hecho el acoplamiento de todas las etapas del circuito se procedió a hacer las mediciones de diferentes resistencias de prueba y con los datos obtenidos se hizo el análisis estadístico para evaluar el sistema de medición.

De forma que se puede decir que el sistema basado en LIA entrega buenos resultados para mediciones que no requirieran una precisión de menos del 2 % con respecto al valor de la conversión del ADC. Esto debido al error de ganancia del ADC embebido en la tarjeta de desarrollo.

El error experimental de las resistencias tuvo una variación de 1.3 % a 13 % para los resistores de carbón, sin embargo, en la medición de conductores con una resistencia teórica debajo de los 100 m $\Omega$  los errores experimentales fueron de más del 60 % lo que hace al sistema de instrumentación inapropiado para estas mediciones.

Al hacer el análisis teórico de la conversión en el que solamente se consideró el acoplamiento analógico de la señal y un comportamiento ideal del ADC los errores experimentales de las mediciones de conductores disminuirán aproximadamente a un 3 %, esto significa que el instrumento de medición podría medir resistencia del orden de cientos de mili-ohms. Por lo tanto, se puede decir que el sistema de acoplamiento analógico de la señal es robusto pero el ADC utilizado no es lo suficientemente preciso para esta aplicación.

Sin embargo, de las mediciones realizadas para caracterizar el ADC de la tarjeta de desarrollo de Texas Instruments, se obtuvo la recta experimental de conversión 3.32, que como se vio, tiene un error de ganancia. Si entonces en vez de considerar la ecuación teórica de la recta de conversión del ADC 3.31, se utiliza la ecuación experimental 3.32, el rango de operación del sistema de medición que es  $R_x = (1.001, \dots, 4.513) \Omega$  podría

---

ampliarse al rango de mili-ohms, ya que al considerar las ecuaciones teóricas al medir resistencias menores a  $1.001 \Omega$  el error experimental es mayor al 10 %.

Por otra parte, el diseño con el ADC7124-8 de Analog Devices<sup>TM</sup> se basó en la configuración de la comunicación serial SPI, la programación del ADC y la toma de mediciones.

Para esta propuesta fue indispensable el uso de herramientas flexibles de software para poder reutilizar el sistema de medición y el código, en este caso se ocupó Micropython para trabajar con dos tarjetas de desarrollo diferentes, Raspberry PiPico y ESP32.

En la primera prueba, la principal variable fue la tasa de filtrado que depende directamente del modo de operación del ADC, *Low* o *High Power*. En el experimento se midieron 5 resistencias diferentes usando los 5 canales de conversión disponibles que ofrece el ADC7124-8.

Los datos mostraron que las mediciones tenían un menor error experimental al usar el modo de *Low Power*, pero en general las mediciones tuvieron un error menor al 2 %. Por su parte, la desviación estándar también fue menor en las mediciones de *Low Power*, pero se observó un aumento de la desviación estándar al aumentar la tasa de filtrado para ambos casos, *Low* y *High Power*.

Los resultados anteriores también se observaron cuando el ADC se configuró con la misma rutina pero con el microcontrolador ESP32, el error experimental fue menor al 2 % al igual que la desviación estándar para el caso de *Low Power*, por lo tanto el rango de operación del sistema de medición es  $R_{RTD} = (38.072\mu\Omega, \dots, 319.375\Omega)$

Finalmente para la última prueba hecha con el ADC7124-8, configurado en el modo *FuFull Power* con una tasa de filtrado digital de 19200 SPS, se comprobó que el tiempo de asentamiento del filtrado digital es importante, sin embargo, las mediciones no indican de alguna manera que aumentar el tiempo entre la configuración y la medición de la resistencia de prueba sea un factor que mejore el desempeño del ADC. Es decir, es suficiente con que el tiempo de espera entre la configuración y la medición sea mayor al tiempo de asentamiento indicado en la hoja de especificaciones.




## A.1. Instalación de Microphyton

### A.1.1. Descarga e instalación de librerías para ESP32-32E

El proceso de instalación y descarga está disponible en el sitio oficial del fabricante DFRobot, para obtener más detalles, consulte su sitio [web](#) <sup>1</sup>. En la siguiente sección, se da una breve descripción para ilustrar los pasos.

#### A.1.1.1. Instrucciones de instalación de librerías

1. Abrir el entorno de desarrollo de Arduino y seleccionar en el menú *File* la opción de *Preferences* (Fig. A.1).
2. En la ventana emergente, seleccionar el icono  para agregar el microcontrolador (Fig. A.2).
3. Incluir en el campo la siguiente dirección  
[http://download.dfrobot.com/FireBeetle/package\\_DFRobot\\_index.json](http://download.dfrobot.com/FireBeetle/package_DFRobot_index.json) para agregar, hacer clic en Aceptar (Fig. A.3).
4. Hacer clic en Ok en la ventana de *Preferences* antes de cerrar.
5. En el menú de *Tools*, seleccionar *Boards* y a continuación *Boards Manager...* (Fig. A.4).

---

<sup>1</sup>[https://wiki.dfrobot.com/FireBeetle\\_Board\\_ESP32\\_E\\_SKU\\_DFR0654](https://wiki.dfrobot.com/FireBeetle_Board_ESP32_E_SKU_DFR0654)

---

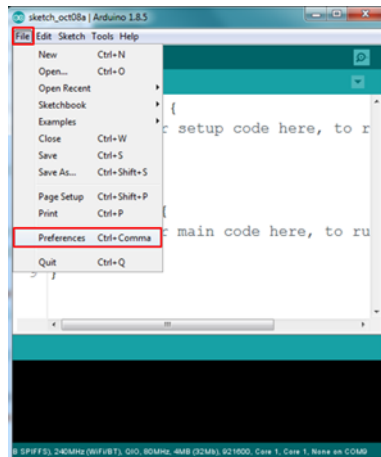


Figura A.1: Menú *File* en Arduino.

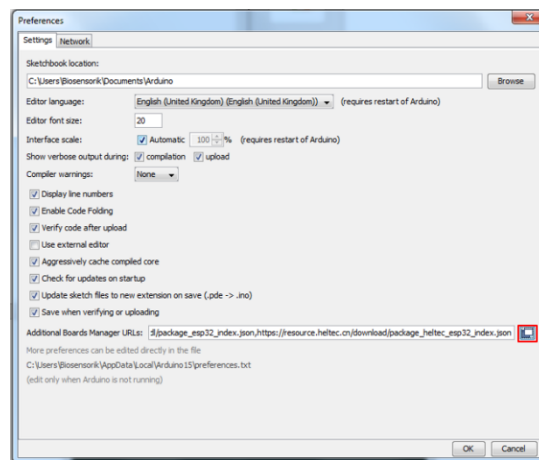


Figura A.2: Menú de preferencias de Arduino.

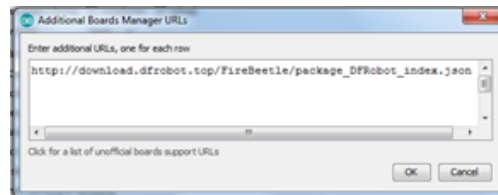


Figura A.3: *Manager* de tarjetas adicionales.

6. Dejar que se actualice el *Boards Manager*.
7. Buscar la tarjeta ESP32 e instalar las tarjetas DFRobot ESP32 de DFRobot (Fig. A.5).

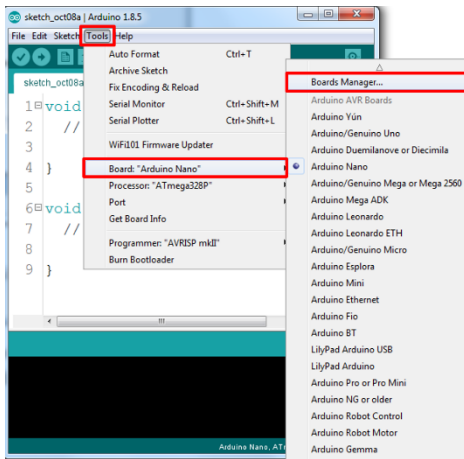


Figura A.4: Menú de herramientas de Arduino.

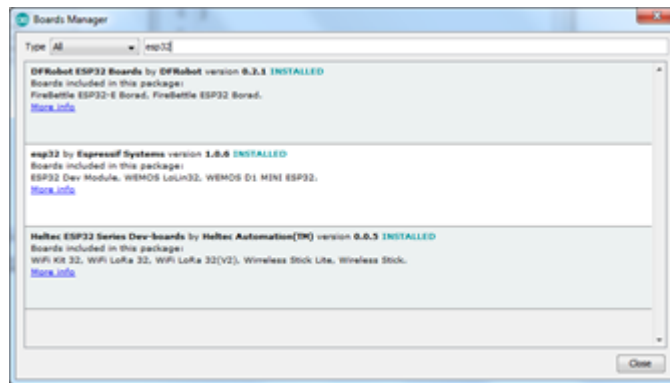


Figura A.5: Instalación de DFRobot ESP32.

8. A continuación seleccionar la tarjeta en el menú de *Tools* (Fig. A.6).

### A.1.2. Descarga e instalación de librerías para ADC7124-8

Las librerías del ADC7124-8 están disponibles en el repositorio de [GitHub](#) de este proyecto.

#### A.1.2.1. Instrucciones de instalación

De vuelta en el IDE de Arduino.

## A. APÉNDICES

---

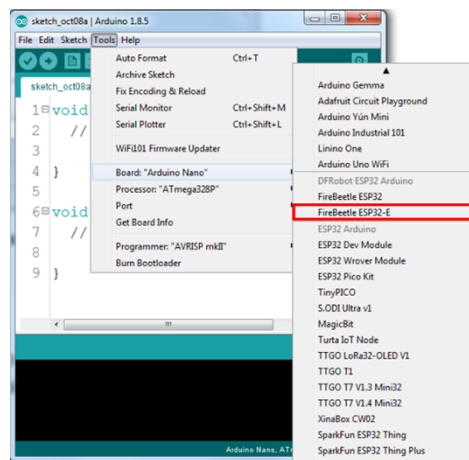


Figura A.6: Menú herramientas Arduino.

1. Descargar el archivo ZIP de la librería del ADC7124-8 del repositorio y guardarla localmente.
2. En el IDE de Arduino seleccionar *Sketch -> Include Library -> Add.ZIP Library...* (Fig. A.7).

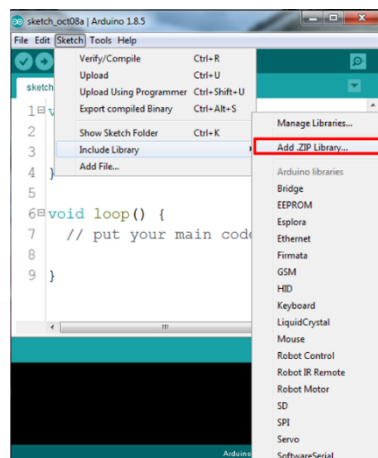


Figura A.7: Agregar librería con archivo ZIP a Arduino.

3. Seleccionar el archivo ZIP descargado anteriormente y agregar.

Para obtener más información, consultar la [documentación de Arduino](#).



### A.1.3. Descarga e instalación de la terminal

#### A.1.3.1. Putty

Hay que seguir las instrucciones de descarga e instalación del sitio web [de Putty Terminal](#). Tras su sencilla instalación al abrir la terminal aparecerá la siguiente ventana (Fig. A.8).

Se debe seleccionar en *Category* la opción de “*Session*” (1). En la siguiente ventana seleccionar el tipo de conexión “*Serial*” (2), luego en el campo *Serial line* (3) escribir el número del puerto COM que está utilizando la tarjeta (en caso de no saber que puerto está siendo ocupado, se puede revisar en el Administrador de dispositivos el número del puerto COM), finalmente, la velocidad (*Speed*) (4) debe ser 9600.

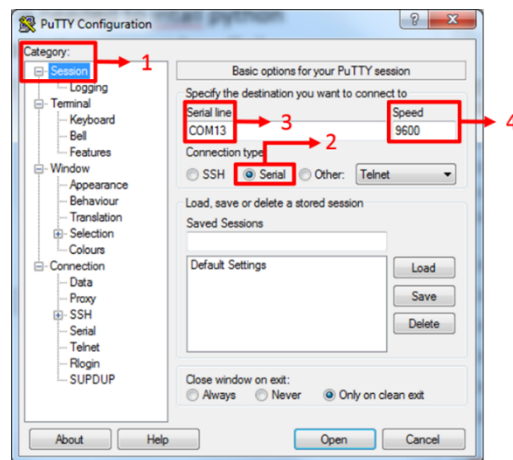


Figura A.8: Interfaz de terminal PuTTY.

#### A.1.3.2. Rshell

Para instalar la herramienta Rshell hay que escribir el siguiente comando en la terminal de Anaconda para descargar la librería de python.

```
python -m pip install rshell
```

Después de esto, es necesario descargar la herramienta MycroPython para el microcontrolador Raspberry Pi Pico. La instalación se explica en la siguiente sección.

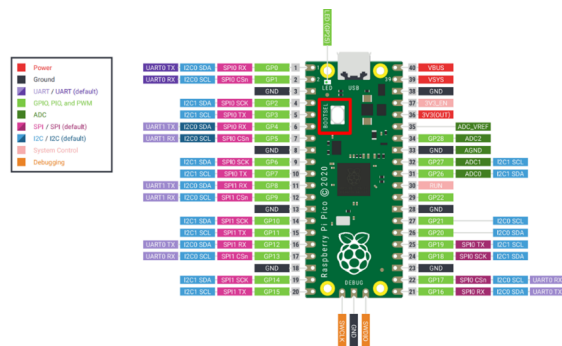


Figura A.9: Placa Raspberry Pi Pico.

#### A.1.4. Descarga e instalación de MicroPython para Raspberry Pi Pico

Después de descargar la herramienta MycroPython para la placa Raspberry, disponible en este repositorio de [GitHub](#) o en el [sitio web](#) oficial, se debe descargar el firmware (rp2-pico-20210902-v1.17.uf2) para la Raspberry Pi Pico. El archivo se puede encontrar en este [GitHub](#) o en el [sitio web](#) oficial.

Como siguiente paso, la placa se conecta a la PC y se reinicia presionando el botón BOOTSEL por 5 sec, que se muestra en la Fig. A.9. La placa aparecerá como un dispositivo de almacenamiento. Copiar el archivo del *firmware* previamente descargado. Desconectar el microcontrolador y reconectar para probar la instalación. Al abrir la terminal PuTTY con las configuraciones mencionadas, es posible escribir código en Python para ser ejecutado por el microcontrolador.

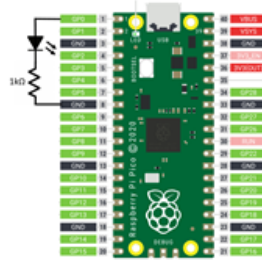
El siguiente código de ejemplo permite encender y apagar un led que está conectado al pin GPIO0, se conecta una resistencia serie para protección del led. Las conexiones se pueden observar en la Fig. A.10.

```

from machine import Pin
p0 = Pin(0, Pin.OUT)           # create output pin on GPIO0
p0.on()                        # set pin to on (high) level
p0.off()                       # set pin to off (low) level
    
```

#### A.1.5. Descarga e instalación de MicroPython para ESP32

En esta sección se describe detalladamente como instalar Micropython para el microcontrolador ESP32, posteriormente se presentan algunas pruebas de conexión para



**Figura A.10:** Conexión Led en Raspberry Pi Pico.

configurar la comunicación serial. También se muestran las conexiones necesarias entre los dispositivos señalizando cada terminal. Y dado que el código anteriormente desarrollado será reutilizado, solo se describirán los parámetros que tienen que ser ajustados.

Anteriormente se trabajó con el microcontrolador Raspberry Pi Pico, con el cual se configuraron diferentes modos de potencia y filtrado en el ADC7124-8, en este caso, se espera reutilizar el código desarrollado aprovechando que es posible programar el microcontrolador ESP32 con la herramienta MicroPython. Para esto, es necesario contar con la herramienta Rshell. El procedimiento para descargar la herramienta de MicroPython se describe a continuación y posteriormente el proceso de instalación.

1. Descargar la carpeta esptool-master del repositorio [de GitHub](#) y guardar localmente (también disponible en este [Repositorio](#)).
2. Descargar el *firmware* para el microcontrolador ESP32 que está disponible en este repositorio de [GitHub](#) (el nombre del archivo es esp32-20210902-v1.17.bin), la versión más reciente se puede encontrar en el [sitio web](#) oficial de MicroPython.
3. Descomprimir la carpeta esptool-master y agregar el archivo del *firmware* a la carpeta esptool-master.
4. Conectar la placa ESP32 a la PC y en el Administrador de dispositivos buscar el nombre del puerto COM, por ejemplo *COM9*.
5. Borrar la memoria flash del ESP32, para ello hay que acceder a través del terminal CMD a la dirección local en donde se guardó la carpeta esptool-master del primer paso. Escribir el siguiente comando, al estar en la dirección de la carpeta, no hay que olvidar cambiar el nombre del puerto:

```
python esptool.py -p <port name (COM9)> erase_flash
```

6. Para instalar MicroPython en el microcontrolador, escribir el siguiente comando cambiando el nombre del puerto y asegurándose de que el nombre del firmware es el mismo que está en su carpeta:

## A. APÉNDICES

---

```
python esptool.py --chip esp32 --port <port name (COM9)> write_flash -z 0x1000 esp32-20210902-v1.17.bin
```

7. Para comenzar a usar la herramienta Rshell, abrir la terminal Anaconda y escribir el siguiente comando, hay que considerar el puerto que está utilizando la placa.

```
rshell -p <port name (COM9)>
```

## A.2. Código

Todos los codigos utilizados en el desarrollo de este proyecto son de libre acceso y se pueden consultar en el repositorio de [Github https://github.com/lozaton/HighResolutionResistanceMeasurement](https://github.com/lozaton/HighResolutionResistanceMeasurement).

### A.2.1. Medición de resistencia con LIA

Código escrito en el IDE Code Composer para el microcontrolador MSP432P401R.

```

//*****
#include "msp.h"
float v_adc=0;
float Rx;
float val_adc ,Vx;
//*****GPIO Setup*****
void GPIO_Init(void){
    P1->SEL0 = 0X00;
    P1->SEL1 = 0X00;
    P1->DIR = 0X01;    //set P1.0 output
    P1->OUT = 0X01;

    P3->SEL0 = 0X00;
    P3->SEL1 = 0X00;
    P3->DIR = 0X01;    //set P3.0 output
    P3->OUT = 0X01;

    P4->SEL0 = 0X00;
    P4->SEL1 = 0X00;
    P4->DIR = 0X01;    //SET P4.0 output
    P4->OUT = 0X00;

    P6->SEL1 |= BIT4;    // Configure P6.1 A14 for ADC
    P6->SEL0 |= BIT4;
} //end GPIO_Init

```

## A. APÉNDICES

---

```
//*****SysTick Initialization*****
void SysTick_Init(void){ //24MHz
    SysTick->LOAD = 0x00FFFFFF; // maximum reload value
    SysTick->CTRL = 0x00000005; // enable, no interrupts
} //end SysTick_Init
//***** SysTick_Wait *****
void SysTick_Wait(uint32_t n){
    SysTick->LOAD = n-1;
    SysTick->VAL = 0;
    while((SysTick->CTRL&0x00010000)== 0){};
} //end SysTick_Wait
//*****ADC Setup *****
void ADC_Init(void){
    // Enable global interrupt
    __enable_irq();
    // Enable ADC interrupt in NVIC module
    NVIC->ISER[0] = 1 << ((ADC14_IRQn) & 31);
    // Sampling time, S&H=16, ADC14 on
    //ADC14_CTL0_SHT0_0      numero de ciclos de reloj (4) necesarios
    // para el tiempo de de muestreo de una muestra MEM 0-7/ MEM 24-31
    //ADC14_CTL0_SHT1_0      numero de ciclos de reloj (4) necesarios
    // para el tiempo de de muestreo de una muestra MEM 8-23
    //ADC14_CTL0_SHP      ADC14 sample-and-hold pulse-mode select
    ADC14->CTL0 = ADC14_CTL0_SHT0_0 | ADC14_CTL0_SHT1_0 | ADC14_CTL0_SHP |
        ADC14_CTL0_ON;
    //ADC14_CTL1_RES_14BIT  14 bit (16 clock cycle conversion time)
    ADC14->CTL1 = ADC14_CTL1_RES_14BIT; // Use sampling timer,
    // 14-bit conversion results
    ADC14->MCTL[0] |= ADC14_MCTLN_INCH14; // A14 6.1 ADC input select;
    Vref=AVCC
    ADC14->IER0 |= ADC14_IER0_IE0; // Enable ADC conv complete
    // interrupt
    SCB->SCR &= ~SCB_SCR_SLEEPONEXIT_Msk; // Wake up on exit from ISR
} //end Int_ADC
```

```

//*****INTERRUPCIONES*****
// ADC14 interrupt service routine
void ADC14_IRQHandler(void) {
    // float RG=498.89;
    //float G=1+49400/RG;
    float G=100;
    float R=101;
    float R2=7600;
    float R3=9030;
    float R4=63600;
    float VDD=3.295;
    float a, offset, VLIA, v_io1, v_io2;

    SysTick_Wait(900000); //T=n/3e+06
    val_adc=ADC14->MEM[0];
    offset = VDD*((R3*R4/(R3+R4))/(R2+(R3*R4/(R3+R4))));
    a=((R2*R4/(R2+R4))/(R3+(R2*R4/(R2+R4))));
    a=1/a;
    v_adc=val_adc*3.2/16384; //Vref 3.3V
    VLIA = a*(v_adc-offset);
    Vx= VLIA/G;

    if (v_adc > offset){ //caso 1
        v_io1=2.472;
        v_io2=0.817;
        Rx=2*R*Vx/(v_io1-v_io2-Vx);
    }
    else{ //caso 2
        v_io1=0.802;
        v_io2=2.464;
        Rx=-2*R*Vx/(v_io2-v_io1+Vx);
    }
    SysTick_Wait(900);
} //end ADC14_IRQHandler

```

## A. APÉNDICES

---

```
//*****MAIN*****
int main(void) {
    volatile unsigned int i;
    WDTA->CTL = WDTA.CTLPW |           // Stop WDT
                WDTA.CTLHOLD;

    GPIO_Init();
    SysTick_Init();
    ADC_Init();
    // Ensures SLEEPONEXIT takes effect immediately
    __DSB();

    while (1)
    {
        P3->OUT = (P3->OUT&(0x01)) ^ 1;
        P4->OUT = (P4->OUT&(0x01)) ^ 1;
        P1->OUT = (P1->OUT&(0x01)) ^ 1;

        //for (i = 20000; i > 0; i--);      // Delay
        // Start sampling/conversion
        ADC14->CTL0 |= ADC14.CTL0_ENC | ADC14.CTL0_SC;
        // SysTick_Wait(30000000); //T=n/3e+06   3s
        __sleep();
        __no_operation();                // For debugger
    }
} //end MAIN
```

### A.2.2. Librería para ADC7124-8

```
from machine import Pin, SoftSPI
spi = SoftSPI(baudrate=1_000_000, polarity=0, phase=0, sck=Pin(0), mosi=
    Pin(2), miso=Pin(1))

def GPIO_Init():
    #RESET
```



```
#disable all channels
#Channel_0 Register
x=spi.read(1, 0x09)
x=spi.read(1, 0x00)
x=spi.read(1, 0x00)
#Channel_1 Register
x=spi.read(1, 0x0A)
x=spi.read(1, 0x00)
x=spi.read(1, 0x00)
#Channel_2 Register
x=spi.read(1, 0x0B)
x=spi.read(1, 0x00)
x=spi.read(1, 0x00)
#Channel_3 Register
x=spi.read(1, 0x0C)
x=spi.read(1, 0x00)
x=spi.read(1, 0x00)
#Channel_4 Register
x=spi.read(1, 0x0D)
x=spi.read(1, 0x00)
x=spi.read(1, 0x00)

def Set_Init():
    #IO_Control_1 Register
    #disable current
    x=spi.read(1, 0x03)
    x=spi.read(1, 0x00)
    x=spi.read(1, 0x00)
    x=spi.read(1, 0x80)

    #Config_0 Register
    #bipolar
    #gain=16
    #ref(+), ref(-)
```

## A. APÉNDICES

---

```
x=spi.read(1, 0x19)
x=spi.read(1, 0x08)
x=spi.read(1, 0x04)

#Filter_0 Register
#post filtering 25 SPS
#Filter output data rate
#-----
#   Power mode   | SPS Output |   code
# F(full)/L(low) | Data Rate  |
#-----
#   F           |     50    | 0xF60180
#   F           | 19,200   | 0xF60001
#   L           |     40    | 0xF6003C
#   L           | 2,400    | 0xF60001
x=spi.read(1, 0x21)
x=spi.read(1, 0xF6) #this value always activate the postfiltering for
                    25 SPS
x=spi.read(1, 0x00)
x=spi.read(1, 0x01)

def CH0():
    #GPIO_Init()
    #Channel_0 Register
    #enable
    #AINP=AIN2
    #AINM=AIN3
    #setup = Config_0
    x=spi.read(1, 0x09)
    x=spi.read(1, 0x80)
    x=spi.read(1, 0x43)
    #IO_Control_1 Register
    #Iout0 = 500 uA
    #Iout_CH = AIN0
```

```
x=spi.read(1, 0x03)
x=spi.read(1, 0x00)
x=spi.read(1, 0x04)
x=spi.read(1, 0x00)

def CH1():
    #GPIO_Init()
    #Channel_1 Register
    #enable
    #AINP=AIN4
    #AINM=AIN5
    #setup = Config_0
    x=spi.read(1, 0x0A)
    x=spi.read(1, 0x80)
    x=spi.read(1, 0x85)
    #IO_Control_1 Register
    #Iout0 = 500 uA
    #Iout_CH = AIN1
    x=spi.read(1, 0x03)
    x=spi.read(1, 0x00)
    x=spi.read(1, 0x04)
    x=spi.read(1, 0x01)

def CH2():
    #GPIO_Init()
    #Channel_2 Register
    #enable
    #AINP=AIN6
    #AINM=AIN7
    #setup = Config_0
    x=spi.read(1, 0x0B)
    x=spi.read(1, 0x80)
    x=spi.read(1, 0xCE)
    #IO_Control_1 Register
```

## A. APÉNDICES

---

```
#Iout0 = 500 uA
#Iout_CH = AIN8
x=spi.read(1, 0x03)
x=spi.read(1, 0x00)
x=spi.read(1, 0x04)
x=spi.read(1, 0x08)
```

**def** CH.3():

```
#GPIO_Init()
#Channel_3 Register
#enable
#AINP=AIN9
#AINM=AIN10
#setup = Config_0
x=spi.read(1, 0x0C)
x=spi.read(1, 0x81)
x=spi.read(1, 0x2A)
#IO_Control_1 Register
#Iout0 = 500 uA
#Iout_CH = AIN11
x=spi.read(1, 0x03)
x=spi.read(1, 0x00)
x=spi.read(1, 0x04)
x=spi.read(1, 0x0B)
```

**def** CH.4():

```
#GPIO_Init()
#Channel_4 Register
#enable
#AINP=AIN12
#AINM=AIN13
#setup = Config_0
x=spi.read(1, 0x0D)
x=spi.read(1, 0x81)
```

```

x=spi.read(1, 0x8D)
#IO_Control_1 Register
#Iout0 = 500 uA
#Iout_CH = AIN14
x=spi.read(1, 0x03)
x=spi.read(1, 0x00)
x=spi.read(1, 0x04)
x=spi.read(1, 0x0E)

def ADC_Init():
    #ADC_control Register
    #Internal CLK
    #continuous conversion mode
    #-----
    #   Power mode   |   code
    # F(full)/L(low) |
    #-----
    #     F           |   0x00C0
    #     F           |   0x0080
    #     L           |   0x0000
    x=spi.read(1, 0x01)
    x=spi.read(1, 0x00)
    x=spi.read(1, 0xC0)
    x=spi.read(3, 0x41)
    print("ADC_Init():_", x)

def Sel_Channel(CH_id):#CH_id is an integer and the number of the RTD to
activate {0,...,4}
    channels = [CH.0, CH.1, CH.2, CH.3, CH.4]
    GPIO_Init()
    channel_sel=channels[CH_id]()
    print("Channel_", CH_id, "selected")

def Read_ADC():

```

## A. APÉNDICES

---

```
x=spi.read(1, 0x42) #ignore first byte "adress instruction"
x=spi.read(3, 0x42)
x=int.from_bytes(x,"big")
return x

def Resistance_Convertion(dataadc):
    Rref = 5110 #reference resistance
    G = 16.0 # Gain
    r = (dataadc - 2**23) * Rref / (G * 2**23)
    return r
```

### A.2.3. main.c para ADC7124-8

```
from machine import Pin, SoftSPI
from time import sleep
spi = SoftSPI(baudrate=1000000, polarity=1, phase=1, sck=Pin(18), mosi=Pin
    (23), miso=Pin(19))
# print("status")
# x=spi.read(1, 0x40)
while True:
    #id 0x14
    print("id")
    x=spi.read(2, 0x45)
    print(x)
    #config register 0x0860 offset
    x=spi.read(1, 0x19)
    x=spi.read(1, 0x08)
    x=spi.read(1, 0x04)
    print("config")
    x=spi.read(3, 0x59)
    print(x)
    x=spi.read(3, 0x5a)
    print(x)
    #filter 0x060180 offset
```

```
x=spi.read(1, 0x21)
x=spi.read(1, 0xf6)
x=spi.read(1, 0x03)
x=spi.read(1, 0xc0)
print(" filter")
x=spi.read(4, 0x61)
print(x)
x=spi.read(4, 0x62)
print(x)
    #Channel 0x8001 offset
x=spi.read(1, 0x09)
x=spi.read(1, 0x80)
x=spi.read(1, 0x43)
print(" Channel")
x=spi.read(3, 0x49)
print(x)
x=spi.read(3, 0x4a)
print(x)
sleep(5)
```

### A.3. Manuales y hojas de Especificaciones

## 2N2222A

### Small Signal Switching Transistor

#### NPN Silicon

##### Features

- MIL-PRF-19500/255 Qualified
- Available as JAN, JANTX, and JANTXV

##### MAXIMUM RATINGS (T<sub>A</sub> = 25°C unless otherwise noted)

Characteristic	Symbol	Value	Unit
Collector-Emitter Voltage	V <sub>CEO</sub>	50	Vdc
Collector-Base Voltage	V <sub>CBO</sub>	75	Vdc
Emitter-Base Voltage	V <sub>EBO</sub>	6.0	Vdc
Collector Current - Continuous	I <sub>C</sub>	800	mAdc
Total Device Dissipation @ T <sub>A</sub> = 25°C	P <sub>T</sub>	500	mW
Total Device Dissipation @ T <sub>C</sub> = 25°C	P <sub>T</sub>	1.0	W
Operating and Storage Junction Temperature Range	T <sub>J</sub> , T <sub>stg</sub>	-65 to +200	°C

##### THERMAL CHARACTERISTICS

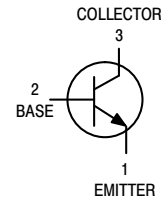
Characteristic	Symbol	Max	Unit
Thermal Resistance, Junction to Ambient	R <sub>θJA</sub>	325	°C/W
Thermal Resistance, Junction to Case	R <sub>θJC</sub>	150	°C/W

Stresses exceeding those listed in the Maximum Ratings table may damage the device. If any of these limits are exceeded, device functionality should not be assumed, damage may occur and reliability may be affected.



ON Semiconductor®

<http://onsemi.com>



TO-18  
CASE 206AA  
STYLE 1

##### ORDERING INFORMATION

Device	Package	Shipping
JAN2N2222A	TO-18	Bulk
JANTX2N2222A		
JANTXV2N2222A		



# PN2907A

Preferred Device

## General Purpose Transistor

PNP Silicon

### Features

- These are Pb-Free Devices\*

### MAXIMUM RATINGS

Rating	Symbol	Value	Unit
Collector-Emitter Voltage	$V_{CE0}$	-60	Vdc
Collector-Base Voltage	$V_{CB0}$	-60	Vdc
Emitter-Base Voltage	$V_{EB0}$	-5.0	Vdc
Collector Current - Continuous	$I_C$	-600	mAdc
Total Device Dissipation @ $T_A = 25^\circ\text{C}$ Derate above $25^\circ\text{C}$	$P_D$	625 5.0	mW mW/ $^\circ\text{C}$
Total Device Dissipation @ $T_C = 25^\circ\text{C}$ Derate above $25^\circ\text{C}$	$P_D$	1.5 12	W mW/ $^\circ\text{C}$
Operating and Storage Junction Temperature Range	$T_J, T_{stg}$	-55 to +150	$^\circ\text{C}$

### THERMAL CHARACTERISTICS

Characteristic	Symbol	Max	Unit
Thermal Resistance, Junction-to-Ambient	$R_{\theta JA}$	200	$^\circ\text{C}/\text{W}$
Thermal Resistance, Junction-to-Case	$R_{\theta JC}$	83.3	$^\circ\text{C}/\text{W}$

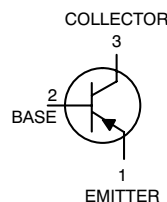
Stresses exceeding Maximum Ratings may damage the device. Maximum Ratings are stress ratings only. Functional operation above the Recommended Operating Conditions is not implied. Extended exposure to stresses above the Recommended Operating Conditions may affect device reliability.

\*For additional information on our Pb-Free strategy and soldering details, please download the ON Semiconductor Soldering and Mounting Techniques Reference Manual, SOLDERRM/D.

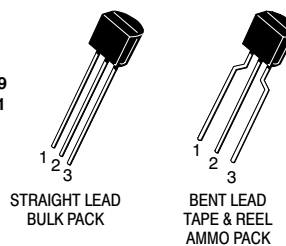


ON Semiconductor®

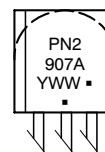
<http://onsemi.com>



TO-92  
CASE 29  
STYLE 1



### MARKING DIAGRAM



PN2907A = Device Code  
Y = Year  
WW = Work Week  
▪ = Pb-Free Package

(Note: Microdot may be in either location)

### ORDERING INFORMATION

See detailed ordering and shipping information in the package dimensions section on page 3 of this data sheet.

Preferred devices are recommended choices for future use and best overall value.



# Low Cost Low Power Instrumentation Amplifier

## AD620

### FEATURES

#### Easy to use

- Gain set with one external resistor  
(Gain range 1 to 10,000)
- Wide power supply range ( $\pm 2.3$  V to  $\pm 18$  V)
- Higher performance than 3 op amp IA designs
- Available in 8-lead DIP and SOIC packaging
- Low power, 1.3 mA max supply current

#### Excellent dc performance (B grade)

- 50  $\mu$ V max, input offset voltage
- 0.6  $\mu$ V/ $^{\circ}$ C max, input offset drift
- 1.0 nA max, input bias current
- 100 dB min common-mode rejection ratio (G = 10)

#### Low noise

- 9 nV/ $\sqrt{\text{Hz}}$  @ 1 kHz, input voltage noise
- 0.28  $\mu$ V p-p noise (0.1 Hz to 10 Hz)

#### Excellent ac specifications

- 120 kHz bandwidth (G = 100)
- 15  $\mu$ s settling time to 0.01%

### APPLICATIONS

- Weigh scales
- ECG and medical instrumentation
- Transducer interface
- Data acquisition systems
- Industrial process controls
- Battery-powered and portable equipment

### CONNECTION DIAGRAM

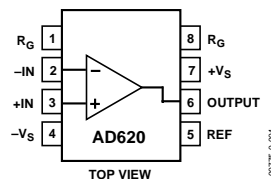


Figure 1. 8-Lead PDIP (N), CERDIP (Q), and SOIC (R) Packages

### PRODUCT DESCRIPTION

The AD620 is a low cost, high accuracy instrumentation amplifier that requires only one external resistor to set gains of 1 to 10,000. Furthermore, the AD620 features 8-lead SOIC and DIP packaging that is smaller than discrete designs and offers lower power (only 1.3 mA max supply current), making it a good fit for battery-powered, portable (or remote) applications.

The AD620, with its high accuracy of 40 ppm maximum nonlinearity, low offset voltage of 50  $\mu$ V max, and offset drift of 0.6  $\mu$ V/ $^{\circ}$ C max, is ideal for use in precision data acquisition systems, such as weigh scales and transducer interfaces. Furthermore, the low noise, low input bias current, and low power of the AD620 make it well suited for medical applications, such as ECG and noninvasive blood pressure monitors.

The low input bias current of 1.0 nA max is made possible with the use of Super $\beta$  processing in the input stage. The AD620 works well as a preamplifier due to its low input voltage noise of 9 nV/ $\sqrt{\text{Hz}}$  at 1 kHz, 0.28  $\mu$ V p-p in the 0.1 Hz to 10 Hz band, and 0.1 pA/ $\sqrt{\text{Hz}}$  input current noise. Also, the AD620 is well suited for multiplexed applications with its settling time of 15  $\mu$ s to 0.01%, and its cost is low enough to enable designs with one in-amp per channel.

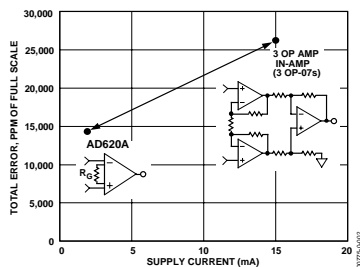


Figure 2. Three Op Amp IA Designs vs. AD620

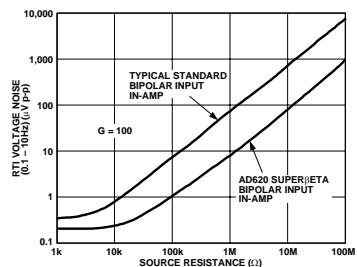


Figure 3. Total Voltage Noise vs. Source Resistance

#### Rev. G

Information furnished by Analog Devices is believed to be accurate and reliable. However, no responsibility is assumed by Analog Devices for its use, nor for any infringements of patents or other rights of third parties that may result from its use. Specifications subject to change without notice. No license is granted by implication or otherwise under any patent or patent rights of Analog Devices. Trademarks and registered trademarks are the property of their respective owners.

One Technology Way, P.O. Box 9106, Norwood, MA 02062-9106, U.S.A.  
 Tel: 781.329.4700 [www.analog.com](http://www.analog.com)  
 Fax: 781.326.8703 © 2004 Analog Devices, Inc. All rights reserved.



# 8-Channel, Low Noise, Low Power, 24-Bit, Sigma-Delta ADC with PGA and Reference

Data Sheet

AD7124-8

## FEATURES

### 3 power modes

#### RMS noise

Low power: 24 nV rms at 1.17 SPS, gain = 128 (255  $\mu$ A typical)

Mid power: 20 nV rms at 2.34 SPS, gain = 128 (355  $\mu$ A typical)

Full power: 23 nV rms at 9.4 SPS, gain = 128 (930  $\mu$ A typical)

Up to 22 noise free bits in all power modes (gain = 1)

#### Output data rate

Full power: 9.38 SPS to 19,200 SPS

Mid power: 2.34 SPS to 4800 SPS

Low power: 1.17 SPS to 2400 SPS

Rail-to-rail analog inputs for gains > 1

Simultaneous 50 Hz/60 Hz rejection at 25 SPS (single cycle settling)

Diagnostic functions (which aid safe integrity level (SIL) certification)

Crosspoint multiplexed analog inputs

8 differential/15 pseudo differential inputs

Programmable gain (1 to 128)

Band gap reference with 10 ppm/ $^{\circ}$ C drift maximum (70  $\mu$ A) (B grade)

Matched programmable excitation currents

Internal clock oscillator

On-chip bias voltage generator

Low-side power switch

General-purpose outputs

Multiple filter options

Internal temperature sensor

Self and system calibration

Sensor burnout detection

Automatic channel sequencer

Per channel configuration

Power supply: 2.7 V to 3.6 V and  $\pm$ 1.8 V

Independent interface power supply

Power-down current: 5  $\mu$ A maximum

Temperature range:  $-40^{\circ}$ C to  $+125^{\circ}$ C

32-lead LFCSP

3-wire or 4-wire serial interface

SPI, QSPI™, MICROWIRE™, and DSP compatible

Schmitt trigger on SCLK

ESD: 4 kV

## APPLICATIONS

Temperature measurement

Pressure measurement

Industrial process control

Instrumentation

Smart transmitters

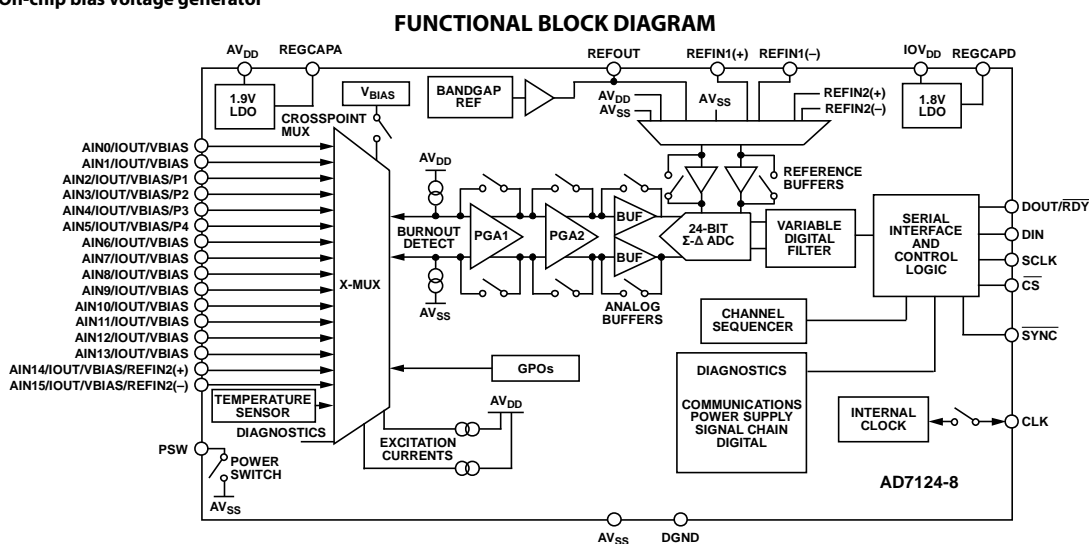


Figure 1.

Rev. E

### Document Feedback

Information furnished by Analog Devices is believed to be accurate and reliable. However, no responsibility is assumed by Analog Devices for its use, nor for any infringements of patents or other rights of third parties that may result from its use. Specifications subject to change without notice. No license is granted by implication or otherwise under any patent or patent rights of Analog Devices. Trademarks and registered trademarks are the property of their respective owners.

One Technology Way, P.O. Box 9106, Norwood, MA 02062-9106, U.S.A.  
Tel: 781.329.4700 ©2015–2018 Analog Devices, Inc. All rights reserved.  
Technical Support [www.analog.com](http://www.analog.com)

## **MSP432P4xx SimpleLink™ Microcontrollers**

# **Technical Reference Manual**



Literature Number: SLAU356I  
March 2015–Revised June 2019



# ESP32 Series Datasheet

## Including:

ESP32-D0WD-V3

ESP32-D0WDR2-V3

ESP32-U4WDH

ESP32-S0WD – [Not Recommended for New Designs \(NRND\)](#)


ESP32-D0WD – [Not Recommended for New Designs \(NRND\)](#)

ESP32-D0WDQ6 – [Not Recommended for New Designs \(NRND\)](#)

ESP32-D0WDQ6-V3 – [Not Recommended for New Designs \(NRND\)](#)



Version 4.0  
Espressif Systems  
Copyright © 2022



## A.4. Mediciones

### A.4.1. Mediciones de simulación en Multisim de la arquitectura LIA sin compuertas Schmitt Trigger

$R_X$	Caso I			Caso II		
	$I/O_1 =$ 3.13 V	$I/O_2 =$ 33 mV	$R =$ 100 $\Omega$	$I/O_1 =$ 20 mV	$I/O_2 =$ 3.07 V	$R =$ 100 $\Omega$
	$V_X$ [mV]	$V_{LIA}$ [V]	$V_{ADC}$ [V]	$V_X$ [mV]	$V_{LIA}$ [V]	$V_{ADC}$ [mV]
0.1	1.550	0.156	1.75	-1.5	-150.0	1.6
0.2	3.100	0.311	1.81	-3.1	-302.0	1.6
0.3	4.650	0.465	1.88	-4.6	-454.0	1.5
0.4	6.190	0.620	1.95	-6.1	-606.0	1.4
0.5	7.740	0.774	2.01	-7.6	-757.0	1.4
0.6	9.280	0.928	2.08	-9.1	-908.0	1.3
0.7	10.800	1.080	2.15	-10.6	-1.1	1.2
0.8	12.400	1.240	2.21	-12.1	-1.2	1.2
0.9	13.900	1.390	2.28	-13.7	-1.4	1.1
1	15.400	1.510	2.34	-15.2	-1.5	1.0
1.1	17.000	1.690	2.41	-16.7	-1.7	965.0
1.2	18.500	1.850	2.47	-18.2	-1.8	901.0
1.3	20.000	2.000	2.54	-19.7	-2.0	836.0
1.4	21.600	2.150	2.61	-21.2	-2.1	771.0
1.5	23.100	2.310	2.67	-22.7	-2.3	707.0
1.6	24.600	2.460	2.74	-24.2	-2.4	643.0
1.7	26.100	2.610	2.80	-25.7	-2.6	578.0
1.8	27.700	2.760	2.87	-27.2	-2.7	514.0
1.9	29.200	2.910	2.93	-28.7	-2.9	450.0
2	30.700	3.070	3.00	-30.2	-3.0	386.0
2.1	32.200	3.220	3.06	-31.7	-3.2	322.0
2.2	33.800	3.370	3.13	-33.2	-3.3	257.0
2.3	35.300	3.520	3.19	-34.7	-3.5	194.0
2.4	36.800	3.670	3.26	-36.2	-3.6	130.0
2.5	38.300	3.790	3.31	-37.6	-3.8	65.8
2.6	39.800	3.830	3.33	-39.1	-3.9	3.3
2.7	41.300	3.840	3.33	-40.6	-4.0	-31.3
2.8	42.800	3.840	3.33	-42.1	-4.0	-40.3

## A. APÉNDICES

---

$R_X$	Caso I			Caso II		
	$I/O_1 =$ 3.13 V	$I/O_2 =$ 33 mV	$R =$ 100 $\Omega$	$I/O_1 =$ 20 mV	$I/O_2 =$ 3.07 V	$R =$ 100 $\Omega$
	$V_X$ [mV]	$V_{LIA}$ [V]	$V_{ADC}$ [V]	$V_X$ [mV]	$V_{LIA}$ [V]	$V_{ADC}$ [mV]

**Tabla A.1:** Mediciones de simulación en Multisim de arquitectura LIA sin compuerta Schmitt Trigger.

### A.4.2. Mediciones de simulación en Multisim de la arquitectura LIA con compuertas Schmitt Trigger

$R_X$	Caso I			Caso II		
	$I/O_1 =$ 3.04 V	$I/O_2 =$ 948 mV	$R =$ 100 $\Omega$	$I/O_1 =$ 952 mV	$I/O_2 =$ 2.83 V	$R =$ 100 $\Omega$
	$V_X$ [mV]	$V_{LIA}$ [V]	$V_{ADC}$ [V]	$V_X$ [mV]	$V_{LIA}$ [V]	$V_{ADC}$ [V]
0.1	939.000 $\mu$ V	0.0954	1.720	-939.000 $\mu$ V	- 0.0920	1.640
0.2	1.880	0.1890	1.760	-1.880	- 0.1860	1.600
0.3	2.810	0.2830	1.800	-2.810	- 0.2790	1.560
0.4	3.750	0.3760	1.840	-3.750	- 0.3730	1.520
0.5	4.690	0.4690	1.880	-4.690	- 0.4660	1.480
0.6	5.620	0.5630	1.920	-5.620	- 0.5590	1.440
0.7	6.560	0.6560	1.960	-6.560	0.6530	1.400
0.8	7.490	0.7490	2.000	-7.490	- 0.7460	1.360
0.9	8.420	0.8420	2.040	-8.420	- 0.8390	1.320
1	9.350	0.9350	2.080	-9.350	- 0.9320	1.280
1.1	10.300	1.0300	2.120	-10.300	- 1.0200	1.240
1.2	11.200	1.1200	2.160	-11.200	- 1.1200	1.200
1.3	12.100	1.2100	2.200	-12.100	- 1.2100	1.160
1.4	13.100	1.3100	2.240	-13.100	- 1.3000	1.120
1.5	14.000	1.4000	2.280	-14.000	- 1.4000	1.080
1.6	14.900	1.4900	2.320	-14.900	- 1.4900	1.040
1.7	15.900	1.5800	2.360	-15.900	- 1.5900	1.000
1.8	16.800	1.6800	2.400	-16.800	- 1.6700	0.961
1.9	17.700	1.7700	2.440	-17.700	1.7700	0.921
2	18.600	1.8600	2.480	-18.600	- 1.8600	0.881
2.1	19.600	1.9500	2.520	-19.600	- 1.9500	0.842



$R_X$	Caso I			Caso II		
	$I/O_1 =$ 3.04 V	$I/O_2 =$ 948 mV	$R =$ 100 $\Omega$	$I/O_1 =$ 952 mV	$I/O_2 =$ 2.83 V	$R =$ 100 $\Omega$
	$V_X$ [mV]	$V_{LIA}$ [V]	$V_{ADC}$ [V]	$V_X$ [mV]	$V_{LIA}$ [V]	$V_{ADC}$ [V]
2.2	20.500	2.0500	2.560	-20.500	- 2.0400	0.802
2.3	21.400	2.1400	2.600	-21.400	- 2.1300	0.762
2.4	22.300	2.2300	2.640	-22.600	- 2.2300	0.723
2.5	23.500	2.3200	2.680	-23.200	- 2.3200	0.683
2.6	24.200	2.4100	2.720	-24.200	- 2.4100	0.644
2.7	25.100	2.5100	2.760	-25.100	- 2.5000	0.604
2.8	26.000	2.6000	2.800	-26.000	- 2.5900	0.565
2.9	26.900	2.6900	2.840	-26.900	- 2.6900	0.526
3	27.800	2.7800	2.880	-27.800	- 2.7800	0.486
3.1	28.800	2.8700	2.910	-28.800	- 2.8700	0.447
3.2	29.700	2.9600	2.950	-29.700	- 2.9600	0.408
3.3	30.600	3.0500	2.990	-30.600	- 3.0500	0.368
3.4	31.500	3.1500	3.030	-31.500	- 3.1400	0.329
3.5	32.400	3.2400	3.070	-32.400	- 3.2300	0.290
3.6	33.300	3.3300	3.110	-33.300	- 3.3200	0.251
3.7	34.200	3.4200	3.150	-34.200	- 3.4200	0.212
3.8	35.200	3.5100	3.190	-35.200	- 3.5100	0.172
3.9	36.100	3.6000	3.230	36.100	- 3.6000	0.133
4	37.000	3.6900	3.270	-37.000	- 3.6900	0.094
4.1	37.900	3.7700	3.300	-37.900	- 3.7800	0.055
4.2	38.800	3.8100	3.320	-38.800	- 3.8700	0.017
4.3				-39.700	- 3.9500	- 0.016

**Tabla A.2:** Mediciones de simulación en Multisim de la arquitectura LIA con compuertas Schmitt Trigger.

#### A.4.3. Mediciones del convertidor bipolar-unipolar

$V_{in}$ [V] multímetro	$V_{ADC}$ [V] Teórico	$V_{ADC}$ [V] Experimen- tal	Error Abso- luto [V]	Error Ex- perimental [%]
3.9040	3.3586	3.3550	0.0036	0.107
3.8060	3.3165	3.3130	0.0035	0.104

## A. APÉNDICES

---

$V_{in}$ [V] multímetro	$V_{ADC}$ [V] Teórico	$V_{ADC}$ [V] Experimen- tal	Error Abso- luto [V]	Error Ex- perimental [%]
3.7790	3.3049	3.3010	0.0039	0.117
3.7000	3.2709	3.2680	0.0029	0.088
3.6000	3.2279	3.2250	0.0029	0.090
3.5010	3.1853	3.1830	0.0023	0.073
3.4040	3.1436	3.1410	0.0026	0.084
3.3010	3.0993	3.0970	0.0023	0.076
3.2050	3.0581	3.0550	0.0031	0.100
3.1010	3.0134	3.0110	0.0024	0.078
3.0040	2.9717	2.9690	0.0027	0.089
2.9020	2.9278	2.9260	0.0018	0.061
2.8040	2.8857	2.8830	0.0027	0.092
2.7010	2.8414	2.8390	0.0024	0.084
2.6010	2.7984	2.7960	0.0024	0.085
2.5030	2.7563	2.7540	0.0023	0.082
2.4040	2.7137	2.7120	0.0017	0.062
2.3020	2.6698	2.6680	0.0018	0.069
2.2000	2.6260	2.6240	0.0020	0.076
2.1070	2.5860	2.5840	0.0020	0.077
2.0040	2.5417	2.5400	0.0017	0.068
1.9000	2.4970	2.4960	0.0010	0.040
1.8010	2.4544	2.4530	0.0014	0.059
1.7020	2.4119	2.4110	0.0009	0.036
1.6060	2.3706	2.3700	0.0006	0.026
1.5010	2.3255	2.3250	0.0005	0.020
1.4020	2.2829	2.2820	0.0009	0.039
1.3030	2.2403	2.2400	0.0003	0.015
1.2050	2.1982	2.1980	0.0002	0.009
1.1040	2.1548	2.1540	0.0008	0.036
1.0010	2.1105	2.1100	0.0005	0.024
0.9020	2.0679	2.0680	- 0.0001	0.003
0.8030	2.0254	2.0250	0.0004	0.018
0.7050	1.9832	1.9830	0.0002	0.012
0.6060	1.9407	1.9410	- 0.0003	0.017
0.5010	1.8955	1.8960	- 0.0005	0.025
0.4038	1.8537	1.8540	- 0.0003	0.014
0.3031	1.8104	1.8110	- 0.0006	0.031
0.2000	1.7661	1.7650	0.0011	0.063

$V_{in}$ [V] multímetro	$V_{ADC}$ [V] Teórico	$V_{ADC}$ [V] Experimen- tal	Error Abso- luto [V]	Error Ex- perimental [%]
0.1016	1.7238	1.7230	0.0008	0.047
-	1.6801	1.6800	0.0001	0.008
- 0.1075	1.6339	1.6340	- 0.0001	0.005
- 0.2048	1.5921	1.5930	- 0.0009	0.058
- 0.3061	1.5485	1.5490	- 0.0005	0.030
- 0.4002	1.5081	1.5080	0.0001	0.005
- 0.5020	1.4643	1.4650	- 0.0007	0.047
- 0.6020	1.4213	1.4220	- 0.0007	0.048
- 0.7000	1.3792	1.3800	- 0.0008	0.059
- 0.8040	1.3345	1.3360	- 0.0015	0.115
- 0.9020	1.2923	1.2920	0.0003	0.026
- 1.0030	1.2489	1.2500	- 0.0011	0.087
- 1.1040	1.2055	1.2070	- 0.0015	0.125
- 1.2080	1.1608	1.1620	- 0.0012	0.106
- 1.3010	1.1208	1.1230	- 0.0022	0.197
- 1.4020	1.0774	1.0790	- 0.0016	0.152
- 1.5040	1.0335	1.0350	- 0.0015	0.144
- 1.6050	0.9901	0.9920	- 0.0019	0.193
- 1.7010	0.9488	0.9510	- 0.0022	0.230
- 1.8000	0.9063	0.9080	- 0.0017	0.193
- 1.9040	0.8615	0.8640	- 0.0025	0.285
- 2.0020	0.8194	0.8220	- 0.0026	0.316
- 2.1060	0.7747	0.7770	- 0.0023	0.298
- 2.2060	0.7317	0.7340	- 0.0023	0.314
- 2.3010	0.6909	0.6940	- 0.0031	0.455
- 2.4000	0.6483	0.6510	- 0.0027	0.417
- 2.5050	0.6032	0.6060	- 0.0028	0.472
- 2.6010	0.5619	0.5650	- 0.0031	0.556
- 2.7010	0.5189	0.5220	- 0.0031	0.601
- 2.8050	0.4742	0.4780	- 0.0038	0.808
- 2.9000	0.4333	0.4370	- 0.0037	0.848
- 3.0030	0.3890	0.3930	- 0.0040	1.017
- 3.1020	0.3465	0.3504	- 0.0039	1.131
- 3.2050	0.3022	0.3075	- 0.0053	1.745
- 3.3000	0.2614	0.2660	- 0.0046	1.778
- 3.4070	0.2154	0.2205	- 0.0051	2.391
- 3.5040	0.1736	0.1789	- 0.0053	3.025

$V_{in}$ [V] multímetro	$V_{ADC}$ [V] Teórico	$V_{ADC}$ [V] Experimen- tal	Error Abso- luto [V]	Error Ex- perimental [%]
- 3.6000	0.1324	0.1381	- 0.0057	4.325
- 3.7030	0.0881	0.0945	- 0.0064	7.275
- 3.8090	0.0425	0.0486	- 0.0061	14.304

**Tabla A.3:** Mediciones del convertidor bipolar-unipolar.

#### A.4.4. Tarjeta MSP432P401R: Prueba del ADC-SAR embebido

$V_{in}$ [V]	Conversión Teórica	Conversión Experimen- tal	$V_{conv}$ [V]	Error Absoluto [V]	Error Experimen- tal [%]
0.1007	500	502	0.1011	0.408	0.408
0.2026	1006	1021	0.2056	1.503	1.503
0.3098	1538	1568	0.3158	1.943	1.943
0.5030	2497	2538	0.5112	1.629	1.629
1.0010	4970	5065	1.0202	1.915	1.915
1.5080	7487	7636	1.5380	1.990	1.990
2.0070	9964	10164	2.0472	2.003	2.003
2.5070	12447	12668	2.5515	1.777	1.777
2.6040	12928	13200	2.6587	2.100	2.100
2.7200	13504	13772	2.7739	1.982	1.982
2.8060	13931	14204	2.8609	1.957	1.957
2.9010	14403	14692	2.9592	2.006	2.006
3.0060	14924	15204	3.0623	1.874	1.874
3.1010	15396	15704	3.1630	2.001	2.001
3.2040	15907	16228	3.2686	2.016	2.016
3.2500	16136	16383	3.2998	1.532	1.532

**Tabla A.4:** Mediciones de prueba del ADC-SAR del microcontrolador MSP432P401R.

### A.4.5. Tarjeta MSP432P401R: medición de voltaje de terminales digitales con compuerta Schmitt Trigger

#### A.4.5.1. Mediciones para el Caso I

Medición del voltaje en la terminal digital que entra a la compuerta Schmitt Trigger así como la salida.

Medición	$I/O_1$ IN [V]	$I/O_1$ OUT [V]	$I/O_2$ IN [mV]	$I/O_2$ OUT [V]
1	1.941	2.481	-3.9	0.823
2	1.734	2.480	-111.0	0.819
3	1.896	2.481	-24.2	0.821
4	1.901	2.479	12.8	0.819
5	1.890	2.481	800.0	0.822
6	1.934	2.481	5.8	0.822
7	1.937	2.484	17.7	0.823
8	1.892	2.481	23.4	0.821
<b>Promedio [V]</b>	<b>1.891</b>	<b>2.481</b>	<b>- 0.010</b>	<b>0.821</b>
<b>Desv Std [mV]</b>	<b>66.80</b>	<b>1.41</b>	<b>43.44</b>	<b>1.58</b>

**Tabla A.5:** Mediciones de voltajes digitales antes y después de la compuerta Schmitt Trigger, Caso I.

#### A.4.5.2. Mediciones para el Caso II

Medición del voltaje en la terminal digital que entra a la compuerta Schmitt Trigger así como la salida.

Medición	$I/O_1$ IN [mV]	$I/O_1$ OUT [V]	$I/O_2$ IN [V]	$I/O_2$ OUT [V]
1	-90.6	0.811	1.919	2.473
2	-17.8	0.815	1.780	2.475
3	-27.8	0.815	1.879	2.475
4	-36.6	0.806	1.862	2.470
5	-1.5	0.809	1.932	2.472
6	-17.2	0.807	1.805	2.472
7	-11.9	0.807	1.878	2.472

## A. APÉNDICES

---

Medición	$I/O_1$ IN [mV]	$I/O_1$ OUT [V]	$I/O_2$ IN [V]	$I/O_2$ OUT [V]
8	-5.5	0.810	1.750	2.472
<b>Promedio [V]</b>	<b>- 0.026</b>	<b>0.810</b>	<b>1.851</b>	<b>2.473</b>
<b>Desv Std [mV]</b>	<b>28.42</b>	<b>3.51</b>	<b>65.66</b>	<b>1.69</b>

**Tabla A.6:** Mediciones de voltajes digitales antes y después de la compuerta Schmitt Trigger, Caso II.

### A.4.6. Amplificador de Instrumentación AD620AN

Prueba al Amplificador de Instrumentación considerando una ganancia de 100.

$V_{in}$ [mV]	$V_{out}$ [V]	$G_{exp}$	<b>Error Experimental [%]</b>
-78.9	- 4.290	54.37	45.63
-52.3	- 4.290	82.03	17.97
-46.3	- 4.290	92.66	7.34
-42.2	- 4.210	99.76	0.24
-38.5	- 3.850	100.00	-
-36.8	- 3.611	98.13	1.87
-35.4	- 3.511	99.18	0.82
-32.5	- 3.290	101.23	1.23
-28.0	- 2.877	102.75	2.75
-27.6	- 2.746	99.49	0.51
-25.8	- 2.566	99.46	0.54
-24.7	- 2.471	100.04	0.04
-22.7	- 2.264	99.74	0.26
-21.5	- 2.144	99.72	0.28
-18.4	- 1.835	99.73	0.27
-18.0	- 1.826	101.44	1.44
-15.7	- 1.561	99.43	0.57
-14.3	- 1.423	99.51	0.49
-12.5	- 1.239	99.12	0.88
-11.4	- 1.135	99.56	0.44
-9.1	- 0.901	99.01	0.99
-8.8	- 0.883	100.34	0.34
-6.7	- 0.661	98.66	1.34

$V_{in}$ [mV]	$V_{out}$ [V]	$G_{exp}$	Error Experimental [%]
-6.4	- 0.648	101.25	1.25
-5.9	- 0.590	100.00	-
-3.4	- 0.345	101.47	1.47
-3.0	- 0.306	101.87	1.87
-1.6	- 0.155	96.75	3.25
2.1	0.221	105.43	5.43
3.0	0.318	106.10	6.10
4.0	0.367	91.75	8.25
8.5	0.852	100.24	0.24
10.4	1.045	100.48	0.48
11.9	1.190	100.00	0.00
12.3	1.226	99.67	0.33
13.9	1.402	100.86	0.86
15.3	1.543	100.85	0.85
15.7	1.569	99.94	0.06
16.8	1.687	100.42	0.42
16.9	1.696	100.36	0.36
17.5	1.767	100.97	0.97
19.7	2.034	103.25	3.25
20.9	2.095	100.24	0.24
22.3	2.230	100.00	-
23.3	2.364	101.46	1.46
23.7	2.374	100.17	0.17
24.1	2.409	99.96	0.04
24.5	2.394	97.71	2.29
26.4	2.644	100.15	0.15
26.6	2.550	95.86	4.14
28.7	2.870	100.00	-
29.9	2.966	99.20	0.80
30.3	3.011	99.37	0.63
32.3	3.227	99.91	0.09
36.7	3.624	98.75	1.25
37.4	3.962	105.94	5.94
42.0	4.190	99.76	0.24
45.0	4.190	93.11	6.89
52.4	4.190	79.96	20.04

**Tabla A.7:** Amplificador de Instrumentación AD620AN.

## A.5. Diagramas de simulaciones

### A.5.1. Simulación de compuertas Schmitt Trigger en Multisim

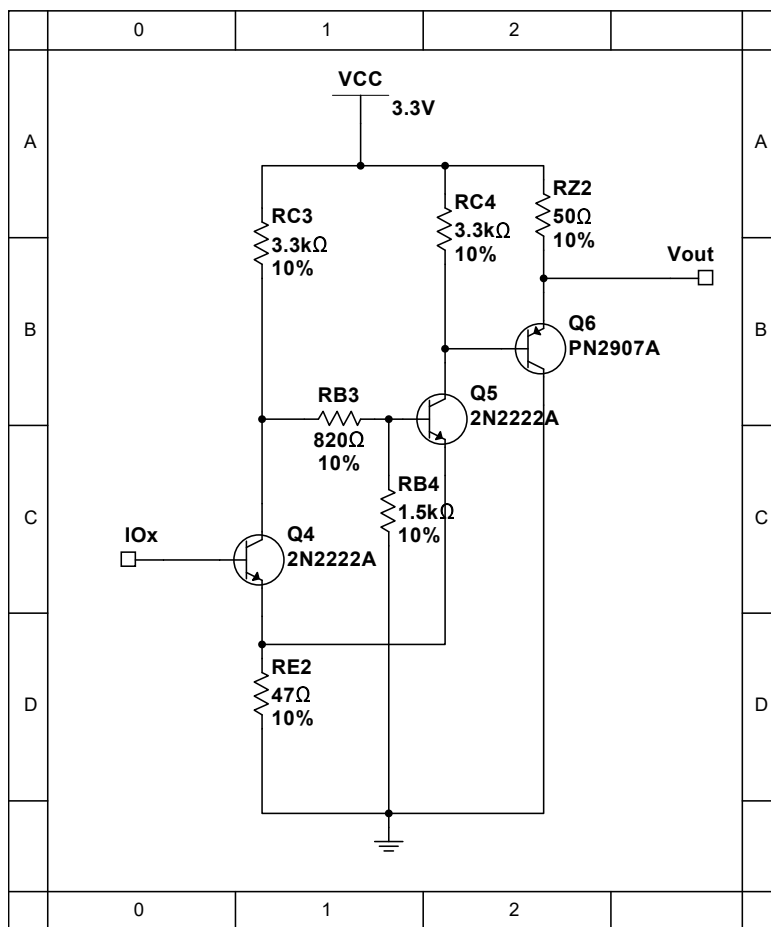


Figura A.11: Diagrama de simulación de compuerta Schmitt Trigger.



A.5.2. Simulación de la arquitectura completa del sistema con LIA en Multisim

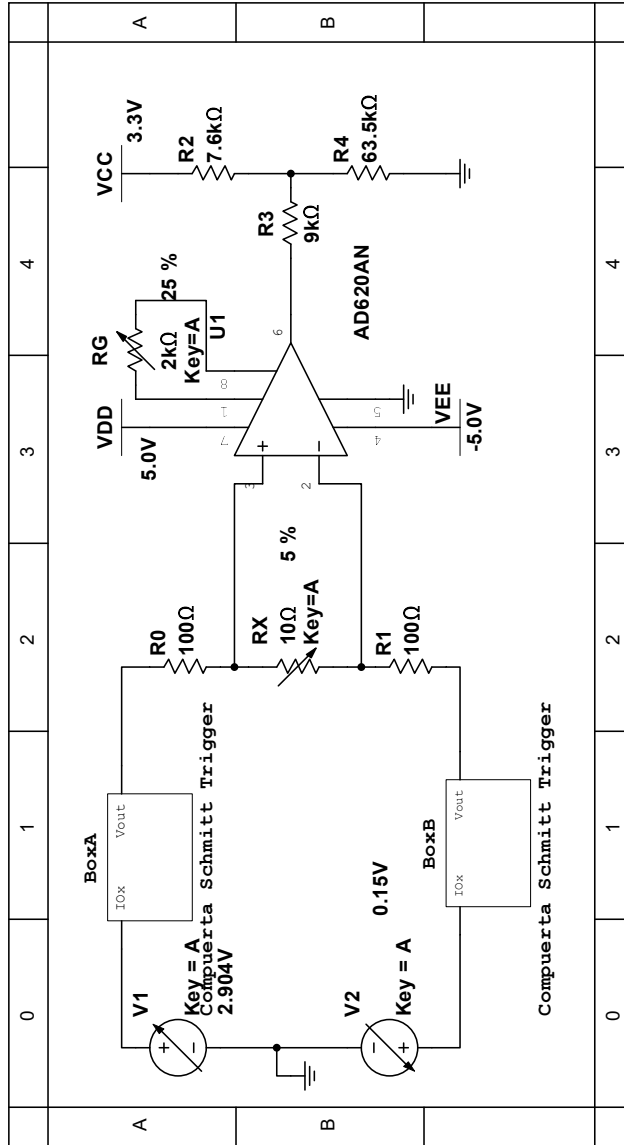


Figura A.12: Diagrama de simulación de arquitectura con LIA.



# Bibliografía

---

- [1] Gabriele Manganaro. *Advanced Data Converters*. University Press, Cambridge, Cambridge, Reino Unido, 2012. [6](#), [7](#), [8](#), [9](#), [10](#), [11](#), [12](#)
- [2] Walt Kester. *Data Conversion Handbook*. Analog Devices Inc. - Elsevier, Massachusetts, EUA, 2005. [8](#), [9](#), [13](#), [14](#), [17](#), [18](#), [21](#)
- [3] Razavi Behzad. *Principles of Data Conversion System Design*. IEEE Press, New York, EUA, 1995. [10](#), [11](#)
- [4] D. I. Crecraft and S. Gergely. *Analog Electronics. Circuits, Systems and Signal Processing*. Butterworth-Heinemann, Massachusetts, EUA, 2002. [15](#)
- [5] Jay L. Devore. *Probabilidad y Estadística para Ingeniería y Ciencias*. Cengage Learning Editores, S.A. de C.V., DF, México, 7 edition, 2008. [22](#), [23](#)
- [6] John G. MaWebster and Halit Eren. *Measurement, Instrumentation, and Sensors Handbook. Spatial, Mechanical, Thermal, and Radiation Measurement*. CRC Press, Florida, EUA, 2 edition, 2014. [23](#)
- [7] R. Rodríguez Gómez and J.C. Cruz Hurtado. Sistema de medición y análisis de impedancia. *Revista de Ingeniería Electrónica, Automática y Comunicaciones*, 36(1):56–66, Abril 2015. [24](#)
- [8] Lars E. Bengtsson. A microcontroller-based lock-in amplifier for sub-milliohm. *Review of Scientific Instruments*, 83(7):075103–0 – 075103–8, Julio 2012. [25](#), [27](#), [38](#)
- [9] Robert Boylestad and Louis Nashelsky. *Electrónica: Teoría de circuitos y dispositivos electrónicos*. Pearson, México, 10 edition, 2009. [25](#)
- [10] Thomas Floyd. *Dispositivos Electrónicos*. Pearson, México, 8 edition, 2008. [26](#), [28](#)

## BIBLIOGRAFÍA

---

- [11] Behrouz Forouzan and Sophia Chung Fegan. *Data communications and networking*. McGraw-Hill, New York, EUA, 4 edition, 2007. 26
- [12] Joseph A. Walston and John R. Miller, editors. *Transistor Circuit Design*. McGraw-Hill, New York, EUA, 1963. 28, 45
- [13] Jon Stenerson. *Programming Controllogix® Programmable Automation Controllers*. Delmar Cengage Learning, New York, EUA, 2009. 29
- [14] Serial protocols. spi background, 2021. Disponible en <https://www.totalphase.com/support/articles/200349236-SPI-Background#history>. 30
- [15] Piyu Dhaker. Analog dialogue.introduction to spi interface, 2021. Disponible en <https://www.analog.com/en/analog-dialogue/articles/introduction-to-spi-interface.html>. 30
- [16] Mary Grace Legaspi and Eric Peña. I<sup>2</sup>c communication protocol: Understanding i<sup>2</sup>c primer, pmbus, and smbus. *Analog Dialogue*, 55(4), Noviembre 2021. 31
- [17] Janet L. Alexson. *USB Complete: The Developer's Guide*. Lakeview Research, Wisconsin, EUA, 5 edition, 2015. 32
- [18] Mary Grace Legaspi and Eric Peña. Uart: A hardware communication protocol understanding universal asynchronous receiver/transmitter. *Analog Dialogue*, 54(4), Diciembre 2020. 34
- [19] Analog Devices Inc. *Datasheet of the Instrumentation Amplifier AD620AN*, 2011. 44
- [20] Semiconductor Components Industries. *2N2222A. Small Signal Switching Transistor*, Noviembre 2013. 48, 49
- [21] Texas Instruments. *Module 9. Lab 9 : SysTick Timer*, 2018. 51
- [22] Texas Instruments. *MSP432P4xx SimpleLink™ Microcontrollers. Technical Reference Manual*, Junio 2018. 52
- [23] Analog Devices Inc. *Datasheet of the ADC7124-8*, 2018. 61, 64, 76, 93
- [24] Analog Devices Inc. *Circuit Note CN-0383*, 2021. Disponible en <https://www.analog.com/en/design-center/reference-designs/circuits-from-the-lab/cn0383.html#rd-overview>. 68, 69, 70
- [25] Robert Resnick, David Halliday, and Kenneth Krane. *Física*, volume 2. Grupo Editorial Patria S.A. de C.V., 5 edition, 2017. 82
- [26] D. C Baird. *Experimentación. Una introducción a la teoría de mediciones y al diseño de experimentos*. Prentice-Hall, 2 edition, 1991. 87

- [27] What is bioimpedance?, 2011. Disponible en <https://www.mn.uio.no/fysikk/english/research/projects/bioimpedance/whatis/>. 97
- [28] Mihaela Ghita, Martine Neckebroek, Jasper Juchem, Dana Copot, Cristina I. Muresan, and Clara M. Ionescu. Bioimpedance sensor and methodology for acute pain monitoring. *Sensors*, 20(23), Noviembre 2020. 97, 98, 99
- [29] A de León-Hernández, C Sánchez-Pérez, and R G Ramírez-Chavarría. An equivalent electric circuit as a tool for monitoring micron sized-particles in suspension from electrical impedance measurements. *Journal of Physics: Conference Series*, 1723(1):012045, Enero 2021. 97, 101
- [30] Roberto G. Ramírez-Chavarría, Celia Sánchez-Pérez, Luisa Romero-Ornelas, and Eva Ramón-Gallegos. Time-constant-domain spectroscopy: An impedance-based method for sensing biological cells in suspension. *IEEE Sensors Journal*, 21(1):185–192, Enero 2021. 97, 101
- [31] Raiff Sales da Fonseca, Marlio J. C. Bonfim, Rafael Domingues Gonçalves, Rafael da Silva Ferraz, and Eduardo Parente Ribeiro. Water pollution assessment by microcontroller-based lock-in amplifier. In *2019 4th International Symposium on Instrumentation Systems, Circuits and Transducers (INSCIT)*, pages 1–5, 2019. 97, 102
- [32] Wolf, Mirela, Traczewska, Teodora, and Grzebyk, Tomasz. The application of impedance measurement to assess biofilm development on technical materials used for water supply system construction. *E3S Web Conf.*, 22:00187, 2017. 102
- [33] Manuel A. Herrera-Juárez and Roberto G. Ramírez-Chavarría. Low-energy and modular wearable device for wireless measurement of physiological signals. *Engineering Proceedings*, 2(1), 2020. 103
- [34] Jiawei Xu, Guy Meynants, and Patrick Merken. Low-power lock-in amplifier for complex impedance measurement. In *2009 3rd International Workshop on Advances in sensors and Interfaces*, pages 110–114, 2009. 103