



TECNOLÓGICO NACIONAL DE MÉXICO
INSTITUTO TECNOLÓGICO DE APIZACO

SIMULACIÓN DE SENSORES DIGITALES Y ANALÓGICOS

TITULACIÓN INTEGRAL INFORME DE RESIDENCIA PROFESIONAL

PARA OBTENER EL TÍTULO PROFESIONAL DE:
INGENIERÍA MECATRÓNICA

PRESENTA:
OCTAVIO MONTIEL CARDONA

NOMBRE DEL ASESOR:
M. en C. ROBERTO ACOLTZI NAVA



APIZACO, TLAX., ENERO DE 2024



Apizaco, Tlaxcala, 24/noviembre/2023.

Oficio No.: MM/651/2023.

ASUNTO: Liberación de Proyecto para Titulación Integral.

MARTÍN ROJAS RAMÍREZ
JEFE DEL DEPTO. DE DIVISIÓN DE ESTUDIOS PROFESIONALES
P R E S E N T E

Por este medio informo que ha sido liberado el siguiente proyecto para Titulación integral:

Alumno:	OCTAVIO MONTIEL CARDONA
Carrera:	INGENIERÍA MECATRÓNICA
No. De Control:	18371439
Nombre del Proyecto:	SIMULACIÓN DE SENSORES DIGITALES Y ANALÓGICOS.
Producto:	TITULACIÓN INTEGRAL INFORME DE RESIDENCIA PROFESIONAL.

Agradezco de antemano su valioso apoyo en esta importante actividad para la formación profesional de nuestros egresados.

ATENTAMENTE

Excelencia en Educación Tecnológica®
Pensar para Servir, Servir para Triunfar

ROBERTO ACOLTZI NAVA
ASESOR



ENRIQUE ACOLTZI BAUTISTA
JEFE DEL DEPTO. DE METAL-MECÁNICA

C.p. Oficina de Titulación
C.p. Departamento de Servicios Escolares
C.p. Expediente. -Jefe de proyectos de vinculación Metal-Mecánica

EAB*petry.





Apizaco, Tlax., 14/diciembre/2023
No. OFICIO: DEP-CT/875/2023

ASUNTO: AUTORIZACIÓN DE IMPRESIÓN

OCTAVIO MONTIEL CARDONA
EGRESADO DE LA CARRERA DE INGENIERÍA MECATRÓNICA
CON NÚMERO DE CONTROL: 18371439
P R E S E N T E

Por este medio, me permito informar a usted que por aprobación de la comisión revisora asignada para valorar el trabajo que mediante la opción **TITULACIÓN INTEGRAL INFORME DE RESIDENCIA PROFESIONAL** cuyo tema es: **"SIMULACIÓN DE SENSORES DIGITALES Y ANALÓGICOS"**, conforme a lo establecido en el procedimiento para la obtención del Título Profesional en el Instituto Tecnológico, la División de Estudios Profesionales a mi cargo le otorga:

AUTORIZACIÓN DE IMPRESION

Debiendo entregar **3** empastados del mismo, apropiadamente encuadernados, a la brevedad, para presentar su acto de recepción profesional.

Sin otro particular por el momento, le envió un cordial saludo.

ATENTAMENTE

Excelencia en Educación Tecnológica®
Pensar para Servir, Servir para Triunfar®


MARTÍN ROJAS RAMÍREZ

JEFE DE LA DIVISIÓN DE ESTUDIOS PROFESIONALES
DIVISIÓN DE ESTUDIOS PROFESIONALES

c.p. archivo



SECRETARÍA DE EDUCACIÓN PÚBLICA
TECNOLÓGICO NACIONAL
DE MÉXICO
INSTITUTO TECNOLÓGICO DE APIZACO
DIVISIÓN DE
ESTUDIOS PROFESIONALES



AGRADECIMIENTOS

Agradezco a mis padres, por confiar en mí, por darme la oportunidad y no abandonarme nunca, gracias por todos sus sacrificios, que día a día hicieron por verme en este lugar, agradezco cada enseñanza, cada enseñanza, cada valor que sembraron en mí y me hicieron cultivar para lograr ser lo que soy ahora, gracias por guiarme en mis momentos más complicados, pero sobre todo gracias por su apoyo incondicional.

Les agradezco a todos los profesores quienes me dieron las bases intelectuales para que pudiera llegar hasta este momento, a esta gran institución.

A mi asesor externo Mtro. Frank Krueguer, por haber confiado en mis ideas y conocimientos adquiridos en la institución para llevar a cabo este proyecto e implementarlo en el producto principal de su compañía.

A mi asesor interno M. en C. Roberto Acoltzi Nava por sus consejos, apoyo y enseñanzas a lo largo de la carrera y en el proyecto, que constituyen la base de mi vida profesional.

ÍNDICE

Capítulo 1. Generalidades del Proyecto.....	1
Introducción.....	1
Krueger Systems, Inc.....	2
Misión.....	2
Visión.....	2
Planteamiento del Problema.....	3
Objetivo General.....	4
Objetivo Específicos.....	4
Justificación.....	4
Cronograma de Actividades.....	5
Capitulo 2. Marco Teórico.....	6
Simulación.....	6
Sensores.....	7
Características de los sensores.....	8
Tipos de sensores y usos en automatización.....	8
Buses de datos.....	12
Comunicación SPI.....	14
Comunicación I2C.....	14
Galga extensiométrica.....	15
Fenómeno Físico.....	16
Uso real.....	16
Configuración de la Galga.....	17

Tecnologías.....	18
C#.....	18
dotNET.....	19
Mono.....	20
Xamarin.....	21
Herramientas.....	23
Jetbrains Rider.....	23
Capitulo 3. Procedimiento y descripción de las tareas realizadas.....	26
Documentos de apoyo para usuarios.....	26
Integración de la galga extensiométrica.....	26
Implementación del sensor SCP1000.....	30
Operación.....	31
Lectura de la presión y la temperatura.....	32
Descripción de registros.....	34
Capitulo 4. Resultados.....	36
Simulaciones.....	36
Simulación de Galga extensiométrica.....	36
Simulación sensor SCP1000.....	38
Conclusiones.....	41
Competencias Desarrolladas.....	42
Fuentes de Información.....	43

Capítulo I. Generalidades

Introducción

Los sensores digitales son hábiles en la generación de salidas codificadas en forma de pulsos o palabras digitales. En el ámbito de la informática, se emplea el código binario decimal (BCD, por sus siglas en inglés) como estándar para representar números decimales. Cada dígito decimal se codifica mediante una secuencia de 4 bits, permitiendo una representación precisa de datos numéricos.

Por otro lado, los sensores analógicos destacan por emitir señales proporcionales a la magnitud que están midiendo. Para lograr esta hazaña, estos dispositivos requieren alimentación de una fuente de corriente continua, por lo general suministrada por un equipo de control eléctrico.

El objetivo principal de este proyecto es llevar a cabo la integración de sensores digitales y analógicos en la plataforma iCircuit. A medida que avanzamos, exploraremos y evaluaremos estos sensores en función de sus respectivas hojas de datos, lo que nos permitirá comprender su funcionamiento y aplicaciones en la práctica. Acompáñenos en esta emocionante aventura tecnológica mientras desentrañamos los secretos de estos sensores y descubrimos cómo enriquecen nuestras capacidades en el ámbito de la ingeniería y la electrónica.

Krueger Systems, Inc.

Krueger Systems Incorporated es una empresa dedicada al desarrollo de software, proveyendo herramientas de diseño y simulación electrónicas, herramientas de desarrollo, entre otras, enfocadas a ser multiplataformas y de alto rendimiento.



Figura 1. Logo de iCircuit

Krueger Systems tiene sede en 3426 Wallingford Ave N #12, Seattle, Washington, Estados Unidos y se encuentra a cargo de Frank A. Krueger.

Misión

Empoderar a nuestros clientes ofreciendo software técnicamente innovador y fácil de usar, esforzándose por mejorar continuamente nuestros productos y lograr la máxima satisfacción de nuestros consumidores.

Visión

Desarrollar y distribuir software altamente capaz y eficiente, que pueda ser ejecutado en cualquier lugar, en cualquier momento y en cualquier dispositivo.

Planteamiento del problema

La simulación de circuitos electrónicos juega un rol crítico en el proceso de diseño electrónico. Esta práctica resulta invaluable para identificar y corregir posibles errores, proporcionando aproximaciones del comportamiento esperado de un circuito. Gracias a estos datos simulados, se pueden detectar y localizar fallos o cortocircuitos de manera eficaz.

La integración exitosa de sensores digitales y analógicos en aplicaciones electrónicas presenta desafíos fundamentales. Estos desafíos incluyen la selección apropiada de sensores, la garantía de precisión en las mediciones, la compatibilidad entre diferentes plataformas, la gestión de costos y la eficiencia en el consumo de energía. Resolver estos problemas es esencial para el desarrollo exitoso de sistemas electrónicos que dependen de datos de sensores para su funcionamiento.

Por lo tanto, es crucial resaltar la importancia de la simulación electrónica en las etapas de diseño y verificación. Esta práctica no solo ahorra costos significativos, sino que también aumenta la eficiencia al proporcionar información sobre el comportamiento del circuito antes de su construcción.

Objetivo General

El objetivo de este proyecto es la simulación y comprensión detallada de sensores digitales y analógicos, profundizando en su funcionamiento, estructura y las señales que generan como resultado de su operación. La meta principal es incorporar exitosamente estos sensores en el motor de simulación de iCircuit.

Este proceso incluye la implementación efectiva de buses de datos, como I2C, SPI y Neopixel, con el propósito de potenciar significativamente la capacidad y versatilidad del simulador. Al lograr este objetivo, se pretende proporcionar a iCircuit la capacidad de simular sistemas electrónicos más complejos y realistas, ofreciendo una herramienta poderosa y precisa para diseñadores e ingenieros que desean crear y validar circuitos electrónicos en diversas aplicaciones y escenarios.

Objetivos Específicos

- Elaborar documentos que describan el funcionamiento de los componentes de la aplicación.
- Integrar y diseñar sensores digitales y analógicos, considerando sus especificaciones.

Justificación

Se propone la inclusión de sensores analógicos y digitales en el proyecto debido a sus notables diferencias en la generación de señales de salida. Mientras que los sensores analógicos emiten señales de forma continua en función de sus mediciones, los sensores digitales generan señales discretas. Esta variabilidad en las respuestas de los sensores es esencial para abordar una amplia diversidad de aplicaciones y escenarios de simulación en el proyecto.



Cronograma de Actividades

#	Actividad		Enero				Febrero				Marzo				Abril				Mayo				Junio			
			1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1	Inicio	P																								
		R																								
3	Desarrollo	P																								
		R																								
7	Implementación de sensores Digitales	P																								
		R																								
8	Implementación de sensores Analógicos	P																								
		R																								
9	Prueba del Sistema	P																								
		R																								
13	Documentación	P																								
		R																								
15	Cierre de Proyecto	P																								
		R																								

Tabla 1. Cronograma de Actividades

Capítulo II. Marco Teórico

Simulación

El término simulación presenta una variedad de acepciones en diferentes áreas, a nivel científico se refiere a la utilización de programas que simulan la actividad de ciertos sistemas, permitiendo obtener predicciones sobre su funcionamiento y efectividad antes de ser utilizados en el mundo real, estos resultados se obtienen por medio de análisis estadísticos.

“Simulación es una técnica numérica para conducir experimentos en una computadora digital. Estos experimentos comprenden ciertos tipos de relaciones matemáticas y lógicas, las cuales son necesarias para describir el comportamiento y la estructura de sistemas complejos del mundo real a través de largos periodos de tiempo.”

- Thomas H. Taylor –

Importancia de la simulación

- A través de un estudio de simulación, se puede estudiar el efecto de cambios internos y externos del sistema, al hacer alteraciones en el modelo del sistema y observando los efectos de esas alteraciones en el comportamiento del sistema.
- Una observación detallada del sistema que se está simulando puede conducir a un mejor entendimiento del sistema y por consiguiente a sugerir estrategias que mejoren la operación y eficiencia del sistema.
- La simulación de sistemas complejos puede ayudar a entender mejor la operación del sistema, a detectar las variables más importantes que interactúan en el sistema y a entender mejor las interrelaciones entre estas variables.
- La técnica de simulación puede ser utilizada para experimentar con nuevas situaciones, sobre las cuales tiene poca o ninguna información. A través de esta experimentación se puede anticipar mejor a posibles resultados no previstos.
- Cuando nuevos elementos son introducidos en un sistema, la simulación puede ser usada para anticipar cuellos de botella o algún otro problema que puede surgir en el comportamiento del sistema.

- En simulación cada variable puede sostenerse constante excepto algunas cuya influencia está siendo estudiada. Como resultado el posible efecto de descontrol de las variables en el comportamiento del sistema necesita no ser tomados en cuenta.

Aplicaciones de la Simulación

Las áreas de aplicación de la simulación son muy amplias, numerosas y diversas, basta mencionar sólo algunas de ellas:

- Análisis del impacto ambiental causado por diversas fuentes.
- Análisis y diseño de sistemas de manufactura.
- Análisis y diseño de sistemas de comunicaciones
- Evaluación del diseño de organismos prestadores de servicios públicos (por ejemplo: hospitales, oficinas de correos, telégrafos, casas de cambio, etc.).
- Análisis de sistemas de transporte terrestre, marítimo o por aire.
- Análisis de grandes equipos de cómputo.

Sensores

En la automatización industrial, los sensores son de gran relevancia, ya que, ofrecen un grado de seguridad que permite garantizar el desarrollo completo del proceso que se está realizando. En este artículo te explicamos qué son, su funcionamiento, sus tipos y características de los sensores.

Los sensores son herramientas que detectan y responden a algún tipo de información del entorno físico. Existe una amplia gama de sensores utilizados en la vida diaria, que se clasifican según las cantidades y características que detectan.

Algunos ejemplos incluyen sensores de corriente eléctrica, magnéticos o de radio, sensores de humedad, sensores de velocidad o flujo de fluidos, sensores de presión, sensores térmicos o de temperatura, sensores ópticos, sensores de posición, sensores ambientales y sensores químicos.

Características de los sensores

- **Rango.** Es el valor mínimo y máximo de la variable física que el sensor puede percibir o medir.
- **Amplitud.** Es la diferencia entre los valores máximos y mínimos de entrada.
- **Exactitud.** El error en la medición se especifica en términos de precisión. Se define como la diferencia entre el valor medido y el valor real. Se define en términos de % de la escala completa o % de la lectura.
- **Precisión.** Se define como la cercanía entre un conjunto de valores y es diferente de la exactitud.
- **Sensibilidad.** Es la relación entre el valor de la salida y el valor de la entrada.
- **La alineación.** Es la máxima desviación entre los valores medidos de un sensor de la curva ideal.
- **Histéresis.** Es la diferencia en la salida cuando la entrada varía de dos maneras, aumentando y disminuyendo.
- **Resolución.** Es el cambio mínimo en la entrada que puede ser detectado por el sensor.
- **Reproducibilidad.** Se define como la capacidad del sensor de producir la misma salida cuando se aplica la misma entrada.
- **Repetibilidad.** Capacidad del sensor de producir la misma salida cada vez que se aplica la misma entrada y todas las condiciones físicas y de medición se mantienen iguales, incluyendo el operador, el instrumento, las condiciones ambientales, etc.
- **Tiempo de respuesta.** Se expresa generalmente como el tiempo en que la salida alcanza un cierto porcentaje de su valor final, en respuesta a un cambio de paso de la entrada.

Tipos de sensores y sus usos en automatización

Sensores de proximidad

Son transductores que detectan objetos o señales que se encuentran cerca del elemento sensor. Existen varios tipos de sensores de proximidad según el principio físico que utilizan, los más comunes son los que te mencionamos a continuación.



Figura 2. Sensores de Proximidad

Inductivos

Han sido diseñados para trabajar generando un campo magnético y detectando las pérdidas de corriente de dicho campo generadas al introducirse en él los objetos de detección férricos. El sensor consiste en una bobina con núcleo de ferrita, un oscilador, un sensor de nivel de disparo de la señal y un circuito de salida.

Al aproximarse un objeto metálico, se inducen corrientes de histéresis en el objeto, debido a ello hay una pérdida de energía y una menor amplitud de oscilación. El circuito sensor reconoce entonces un cambio específico de amplitud y genera una señal que conmuta la salida de estado sólido o la posición on y off.



Figura 3. Sensores inductivos

Magnéticos

Se destacan por su capacidad para detectar objetos magnéticos, generalmente imanes permanentes, a distancias considerables, a pesar de su tamaño compacto. Estos sensores se emplean para activar procesos de conmutación.



Figura 4. Sensores Magnéticos

Capacitivos

Detectan objetos metálicos, o no metálicos, midiendo el cambio en la capacitancia, la cual depende de la constante dieléctrica del material a detectar, su masa, tamaño, y distancia hasta la superficie sensible del detector. Debido a la influencia del objeto a detectar, y del cambio de capacitancia, la amplificación se incrementa haciendo entrar en oscilación el oscilador.



Figura 5. Sensores Capacitivos

Ultrasónicos

Trabajan libres de roces mecánicos y detectan objetos a distancias de hasta 8 m y emiten impulsos ultrasónicos. Estos se reflejan en un objeto, el sensor recibe el eco producido y lo convierte en señales eléctricas, las cuales son elaboradas en el aparato de valoración.



Figura 6. Sensor ultrasónico

Codificadores incrementales y absolutos

Los incrementales generan un tren de pulsos o una onda sinusoidal, donde el número de pulsos pueden ser una medida de velocidad, longitud o posición. En los absolutos, cada posición corresponde a un único código, de modo que tras un corte de energía la posición queda almacenada y puede ser leída al volver la alimentación. Esto evita tener que volver a referenciar el equipo.



Figura 7. Codificadores

Sensores fotoeléctricos

Responden al cambio en la intensidad de la luz, requieren de un componente emisor que genera la luz, y un componente receptor que percibe la luz generada por el emisor. Están diseñados especialmente para la detección, clasificación y posicionamiento de objetos; la detección de formas, colores y diferencias de superficie, incluso bajo condiciones ambientales extremas.



Figura 8. Sensores fotoeléctricos

Sensores de presión

Su objetivo es transformar una magnitud física en una eléctrica, en este caso transforman una fuerza por unidad de superficie en un voltaje equivalente a esa presión ejercida. Aunque los formatos son diferentes, destacan en general por su

robustez, ya que, en procesos industriales están sometidos a todo tipo de líquidos, existiendo así sensores de presión para agua, de presión para aceite, líquido de frenos, etc.



Figura 9. Sensor de Presión

Sensores de temperatura

Recogen información sobre la temperatura de una fuente y la cambian a una forma que pueda ser comprendida por otro dispositivo. Se trata de una categoría de sensores de uso común que detectan la temperatura o el calor y también mide la temperatura de un medio.



Figura 10. Sensor de Temperatura

Buses de Datos

Los buses de datos son una herramienta imprescindible para el almacenamiento y transporte de información entre los dispositivos electrónicos. Esta tecnología ha permitido a los equipos informáticos trabajar de forma más rápida y eficiente, mejorando la velocidad con la que se ejecutan las tareas. En este artículo vamos a profundizar en el tema, mostrando conceptos básicos del funcionamiento de los buses de datos así como sus diferentes tipologías.

Funciones del bus de datos

Las funciones del bus de datos son la base de todo sistema informático, ya que permiten el intercambio de información entre los diferentes dispositivos. Estas funciones son:

- Transporte de datos. El bus de datos permite el transporte rápido y eficiente de datos entre los dispositivos conectados. Esto significa que los usuarios pueden intercambiar información sin tener que esperar largas cantidades de tiempo para una transferencia completa.
- Arbitraje. El bus de datos es capaz de garantizar que cada dispositivo reciba exactamente la cantidad correcta de información sin sobrecargar otros dispositivos conectados al mismo. Esto se logra mediante un proceso llamado arbitraje, en el cual el bus asigna prioridades a los diferentes dispositivos para determinar cuánto tiempo y ancho de banda se les asignará a cada uno.
- Direccionamiento. Además, el bus proporciona direccionamiento para ayudar a identificar a cada dispositivo individualmente dentro del sistema informático, permitiendo así una comunicación más eficiente entre ellos.
- Control. El bus controla y regula la comunicación entre los dispositivos conectados al mismo, lo que garantiza la eficiencia del sistema y evita situaciones en las que dos o más dispositivos intenten acceder simultáneamente a la misma memoria compartida.
- Protección. El bus también proporciona protección contra errores humanos y fallas en hardware, permitiendo al sistema detectar fallas y recuperarse adecuadamente sin necesidad de reiniciarse manualmente en caso de fallas graves o incluso inexistentes.

En resumen, las funciones del bus establecen las reglas básicas para el intercambio seguro e ininterrumpido entre los diversos dispositivos conectados al mismo, haciendo posible un funcionamiento óptimo del sistema informático completo.

Comunicación SPI

La interfaz SPI (Serial Peripheral Interface) fue desarrollada por Motorola (actualmente parte de NXP Semiconductors) alrededor de 1985. Este protocolo síncrono está diseñado para la comunicación entre dispositivos en distancias cortas. Desde su creación, se ha convertido en un estándar de facto ampliamente adoptado por numerosos fabricantes de semiconductores, especialmente en microprocesadores y microcontroladores.

La popularidad de SPI se basa en sus múltiples ventajas. En primer lugar, ofrece una interfaz de hardware simple y directa que proporciona una gran flexibilidad en cuanto a la cantidad de bits que se pueden transferir. Emplea un modelo maestro-esclavo, con un maestro único que puede gestionar múltiples dispositivos secundarios mediante comunicación dúplex y operar a velocidades de reloj de hasta 50 MHz. SPI no utiliza un protocolo estándar y se enfoca en la transferencia de paquetes de datos, lo que la convierte en una elección ideal para transmitir flujos largos de información.

El protocolo SPI requiere un máximo de cuatro líneas de señal. El dispositivo maestro, generalmente un procesador o controlador, controla y suministra el reloj (SCK) y las líneas de selección de chip (CS). La comunicación bidireccional se realiza a través de las líneas de datos Master Out Slave In (MOSI) y Master In Slave Out (MISO). Esta configuración simplificada y su alta eficiencia hacen que SPI sea una opción destacada para muchas aplicaciones de comunicación de datos en la industria.

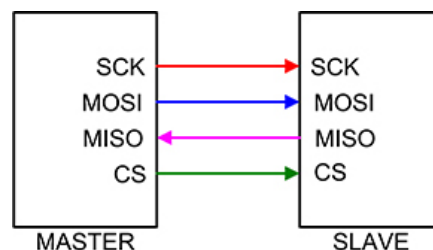


Figura 11. Conexión Dúplex

Comunicación I2C

El protocolo I2C (Inter-Integrated Circuit) es un estándar diseñado para facilitar la comunicación entre múltiples circuitos integrados digitales, comúnmente conocidos como "chips" periféricos, y uno o más chips controladores. Similar a la Interfaz Periférica Serial

(SPI), el protocolo I2C se utiliza principalmente para comunicaciones de corta distancia dentro de un mismo dispositivo o placa. Además, a diferencia de las Interfaces Seriales Asíncronas como RS-232 o UARTs, el protocolo I2C solo requiere dos cables de señal para el intercambio de información, lo que lo hace altamente eficiente y adecuado para una variedad de aplicaciones.

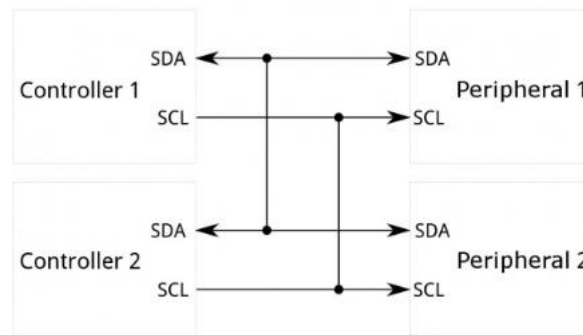


Figura 12. Conexión I2C

La velocidad de transmisión de datos está entre la serie asíncrona y la SPI; la mayoría de los dispositivos I2C pueden comunicarse a 100kHz o 400kHz. El I2C tiene una cierta sobrecarga; por cada 8 bits de datos que se envían, hay que transmitir un bit extra de metadatos.

Galga extensiométrica

Una galga extensométrica o extensómetro es un sensor que mide la deformación, presión, carga, par y posición de determinado material. Se basa en el efecto piezorresistivo, que es la propiedad que tienen ciertos materiales de cambiar el valor nominal de su resistencia eléctrica cuando se les somete a ciertos esfuerzos mecánicos que causan deformación. Esta variación es dependiente de la dirección de las fuerzas ejercidas. Un esfuerzo que deforma la galga producirá una variación en su resistencia eléctrica. Esta variación se produce por el cambio de longitud, el cambio originado en la sección o el cambio generado en la resistividad. Inventada por los ingenieros Edward E. Simmons y Arthur C. Ruge en 1938, la galga extensiométrica hace una lectura directa de las deformaciones longitudinales en cierto punto del material que se está analizando. La magnitud que lo representa es la ϵ (épsilon), que es adimensional y expresa el cambio de la longitud sobre la longitud inicial.

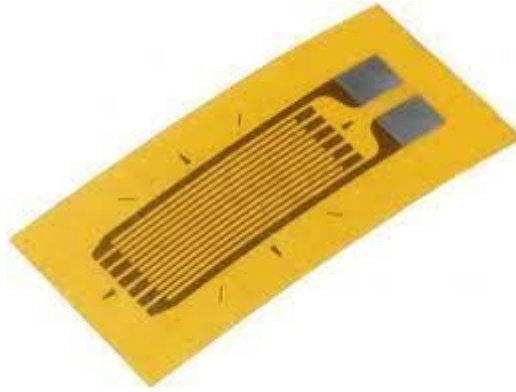


Figura 13. Galga Extensiométrica

Fenómeno Físico

Las galgas extensiométricas aprovechan la propiedad física de la resistencia eléctrica y su dependencia no sólo de la resistividad del conductor, que es una propiedad del propio material, sino también de la geometría del conductor. Cuando un conductor eléctrico es deformado dentro de su límite de elasticidad, de tal forma que no se produzca rotura o deformación permanente en el mismo, éste se volverá más estrecho y alargado.

Uso real

Las galgas se utilizan para la medición electrónica de diferentes magnitudes mecánicas tales como la presión, la carga, la deformación, el torque, entre otras. Estas mediciones pueden catalogarse en mixtas, dinámicas y estáticas. Las mixtas se usan para elementos sometidos a cargas que están variando, las dinámicas a elementos que vibran o son impactados y las estáticas, como su mismo nombre indica, son elementos sometidos a cargas que no están en movimiento.

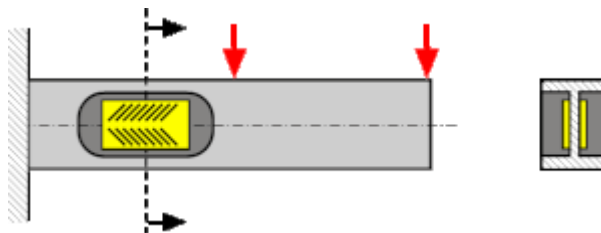


Figura 14. Esquema de Medición de esfuerzo cortante en viga

Configuración de la Galga

Las galgas extensiométricas dependen del propósito final del estudio. Se pueden usar configuraciones que implican una o más galgas que estén en diferentes direcciones. Esta configuración depende de si los esfuerzos que han de medirse están en direcciones uniaxiales, biaxiales o si están en diferentes direcciones.

En el caso de esfuerzos para una sola dirección, se usan con frecuencia sensores largos y angostos, que permiten maximizar la deformación sobre la galga del material en la dirección de interés.

Cuando se trabaja en varias direcciones, podemos lograr mediciones simultáneas utilizando varias galgas configuradas en las direcciones de interés. Sin embargo, es posible hacer esta tarea más sencilla y precisa por medio del uso de galgas de múltiples elementos.

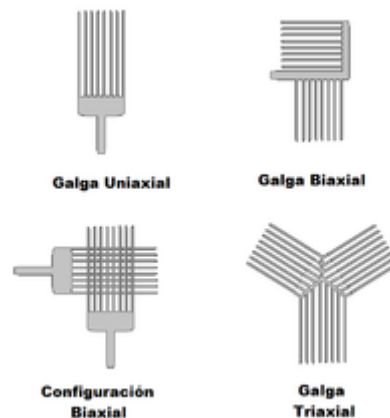


Figura 15. Configuración de las Galgas Extensiométricas

Tecnologías

C#



Figura 16. C# Logo no oficial

C# (pronunciado "Ci Sharp" en inglés) es un lenguaje de programación moderno, orientado a objetos y de tipo seguro, el cual permite a los desarrolladores crear muchos tipos de aplicaciones seguras y robustas que se ejecutan en .NET. Este lenguaje tiene sus raíces en la familia de lenguajes C y resultará inmediatamente familiar a los programadores de C, C++, Java y JavaScript.

C# es un lenguaje de programación orientado a objetos y a componentes, proporciona construcciones de lenguaje que soportan directamente estos conceptos, lo que convierte a C# en un lenguaje natural para crear y utilizar componentes de software. Desde su origen, ha añadido características para soportar nuevas cargas de trabajo y prácticas de diseño de software emergentes.

Varias características de C# ayudan a crear aplicaciones robustas y duraderas. El colector de basura recupera automáticamente la memoria ocupada por objetos no utilizados que no se pueden alcanzar. Los tipos nulos protegen de las variables que no hacen referencia a los objetos asignados. El manejo de excepciones proporciona un enfoque estructurado y extensible para la detección y recuperación de errores. El soporte del lenguaje para las operaciones asíncronas proporciona una sintaxis para construir sistemas distribuidos. Este lenguaje tiene un sistema de tipos unificado. Todos los tipos de C#, incluidos los tipos primitivos como `int` y `double`, heredan de un único tipo de objeto raíz. Todos los tipos comparten un conjunto de operaciones comunes. Los valores de cualquier tipo pueden ser

almacenados, transportados y operados de manera consistente. Además, se admiten tanto tipos de referencia definidos por el usuario como tipos de valor. C# permite la asignación dinámica de objetos y el almacenamiento en línea de estructuras ligeras, también admite métodos y tipos genéricos, que proporcionan mayor seguridad de tipos y rendimiento. C# proporciona iteradores, que permiten a los implementadores de clases de colección definir comportamientos personalizados para el código del cliente.

dotNET



Figura 17. dotNET Logo

Los programas en C# se ejecutan en .NET, un sistema de ejecución virtual llamado Common Language Runtime (CLR) y un conjunto de bibliotecas de clases. El CLR es la implementación por parte de Microsoft de la infraestructura de lenguaje común (CLI), un estándar internacional. La CLI es la base para crear entornos de ejecución y desarrollo en los que los lenguajes y las bibliotecas funcionan a la perfección.

El código fuente escrito en C# se compila en un lenguaje intermedio (IL) que se ajusta a la especificación de la CLI. El código IL y los recursos, como los mapas de bits y las cadenas, se almacenan en un ensamblaje, normalmente con la extensión .dll. Un ensamblaje contiene un manifiesto que proporciona información sobre los tipos, la versión y la cultura del ensamblaje.

Cuando se ejecuta el programa C#, el ensamblaje se carga en el CLR. El CLR realiza una compilación Just-In-Time (JIT) para convertir el código IL en instrucciones nativas de la máquina. El CLR proporciona otros servicios relacionados con la recolección automática de basura, el manejo de excepciones y la gestión de recursos. El código ejecutado por el CLR

se denomina a veces "código gestionado". El "código no gestionado" se compila en lenguaje de máquina nativo dirigido a una plataforma específica.

La interoperabilidad del lenguaje es una característica clave de .NET. El código IL producido por el compilador de C# se ajusta a la especificación de tipos comunes (CTS).

Además de los servicios de tiempo de ejecución, .NET también incluye amplias bibliotecas. Estas bibliotecas soportan muchas cargas de trabajo diferentes. Están organizadas en espacios de nombres que proporcionan una amplia variedad de funcionalidades útiles. Las bibliotecas incluyen todo tipo de funciones, desde la entrada y salida de archivos hasta la manipulación de cadenas y el análisis sintáctico de XML, pasando por los marcos de aplicaciones web y los controles de Windows Forms. La aplicación típica de C# utiliza ampliamente la biblioteca de clases .NET para manejar las tareas comunes de "fontanería".

Mono



Figura 18. Mono Logo

Mono, la plataforma de desarrollo de código abierto basada en .NET Framework, permite a los desarrolladores crear aplicaciones multiplataforma con una mayor productividad. La implementación de .NET de Mono se basa en los estándares ECMA para C# y la Common Language Infrastructure.

Apoyado anteriormente por Novell, Xamarin y ahora por Microsoft y la Fundación .NET, el proyecto Mono cuenta con una comunidad colaboradora activa y entusiasta. Mono incluye tanto herramientas para desarrolladores como la infraestructura necesaria para ejecutar aplicaciones de cliente y servidor .NET.

Hay varios componentes que conforman Mono:

- **Compilador de C#:** El compilador de C# de Mono tiene características completas para C# 1.0, 2.0, 3.0, 4.0, 5.0 y 6.0 (ECMA). Una buena descripción de las características de las distintas versiones está disponible en Wikipedia.
- **Mono Runtime:** El tiempo de ejecución implementa la Infraestructura de Lenguaje Común (CLI) de ECMA. El tiempo de ejecución proporciona un compilador Just-in-Time (JIT), un compilador Ahead-of-Time (AOT), un cargador de bibliotecas, el recolector de basura, un sistema de hilos y funcionalidad de interoperabilidad.
- **Biblioteca de clases de .NET Framework:** La plataforma Mono ofrece un amplio conjunto de clases que proporcionan una base sólida para crear aplicaciones. Estas clases son compatibles con las clases de .NET Framework de Microsoft.
- **Biblioteca de clases Mono:** Mono también proporciona muchas clases que van más allá de la biblioteca de clases base proporcionada por Microsoft. Éstas proporcionan funcionalidades adicionales que son útiles, especialmente en la construcción de aplicaciones Linux. Algunos ejemplos son clases para Gtk+, archivos Zip, LDAP, OpenGL, Cairo, POSIX, etc.

Xamarin



Figura 19. Xamarin Logo

Xamarin es una plataforma de código abierto para crear aplicaciones modernas y de alto rendimiento para iOS, Android y Windows con .NET. Xamarin es una capa de abstracción que gestiona la comunicación del código compartido con el código de la plataforma subyacente. Xamarin se ejecuta en un entorno gestionado que proporciona comodidades como la asignación de memoria y la recolección de basura.

Xamarin permite a los desarrolladores compartir una media del 90% de su aplicación entre plataformas. Este patrón permite a los desarrolladores escribir toda su lógica de negocio en

un único lenguaje (o reutilizar el código de la aplicación existente) pero conseguir un rendimiento, aspecto y sensación nativos en cada plataforma.

Las aplicaciones de Xamarin pueden escribirse en PC o Mac y compilarse en paquetes de aplicaciones nativas, como un archivo .apk en Android, o un archivo .ipa en iOS.

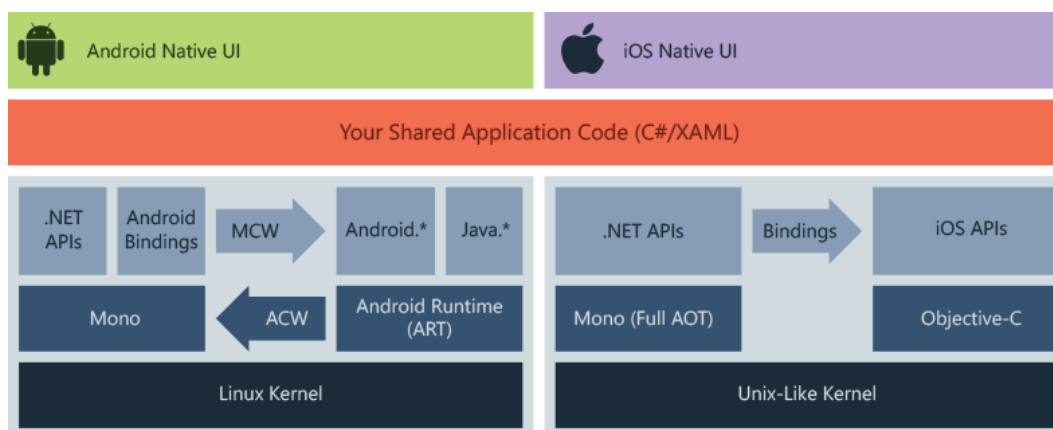


Figura 20. Arquitectura General de una Aplicación Xamarin

La arquitectura general de una aplicación Xamarin multiplataforma es una solución poderosa y versátil para desarrolladores que desean crear aplicaciones móviles nativas para múltiples plataformas, como Android, iOS y, en algunos casos, Windows. Xamarin se destaca por su capacidad para aprovechar al máximo la reutilización de código, lo que facilita el desarrollo y el mantenimiento de aplicaciones para diferentes sistemas operativos.

En la esencia de esta arquitectura se encuentra la idea de que puedes crear una interfaz de usuario nativa específica para cada plataforma utilizando las herramientas y controles proporcionados por Xamarin. Esto significa que puedes diseñar la interfaz de usuario de tu aplicación de manera que se ajuste perfectamente a las pautas de diseño y las expectativas de los usuarios en Android e iOS, sin comprometer la calidad ni la experiencia del usuario.

El núcleo de la lógica de negocio de la aplicación se escribe en C#, un lenguaje de programación sólido y ampliamente adoptado. Lo que hace que Xamarin sea aún más poderoso es su capacidad para compartir gran parte de este código de lógica de negocio entre las diferentes plataformas. Esto se logra mediante el uso de bibliotecas y técnicas de programación compartida, lo que reduce significativamente la duplicación de esfuerzos y el

tiempo de desarrollo. Como mencionaste, en muchos casos, hasta un 80% del código puede ser compartido, lo que resulta en un desarrollo más eficiente y menos propenso a errores.

Herramientas

Jetbrains Rider



Figura 21, Rider Logo

JetBrains Rider es un IDE (Entorno de Desarrollo Integrado) altamente versátil y completo que se ha ganado una sólida reputación en la comunidad de desarrolladores .NET. Está basado en la plataforma IntelliJ y se beneficia de la experiencia acumulada de JetBrains en la creación de herramientas de desarrollo de alta calidad, como ReSharper. Rider ha sido diseñado para ofrecer una experiencia de desarrollo eficiente y fluida en una amplia gama de proyectos .NET, incluyendo los basados en .NET Framework, Mono y .NET Core. A continuación, profundizaremos en algunas de las características y capacidades más destacadas de JetBrains Rider:

1. **Compatibilidad Multiplataforma:** Rider es multiplataforma, lo que significa que se puede ejecutar en sistemas operativos Windows, macOS y Linux. Esto permite a los desarrolladores trabajar en su entorno preferido sin comprometer la calidad de la herramienta.
2. **Soporte para Múltiples Lenguajes:** Rider es conocido por su sólido soporte para una amplia variedad de lenguajes utilizados en el desarrollo .NET. Esto incluye C#, VB.NET, F#, así como tecnologías relacionadas como ASP.NET, XAML, XML, JavaScript, TypeScript, JSON, HTML, CSS y SQL. Esta diversidad de lenguajes

permite a los desarrolladores trabajar en proyectos complejos que involucran componentes en diversos idiomas.

3. **Funciones Potentes y Productividad Mejorada:** Rider viene con un conjunto de características avanzadas que mejoran la productividad del desarrollador. Esto incluye refactorización de código, navegación inteligente, inspección de código en tiempo real, finalización automática de código y más. Además, muchas de estas características funcionan no solo en C#, sino en varios lenguajes, lo que aumenta la eficiencia en proyectos multidisciplinarios.
4. **Compatibilidad con una Amplia Gama de Aplicaciones .NET:** Rider le permite trabajar con una variedad de aplicaciones .NET, incluyendo aplicaciones de escritorio, aplicaciones web, bibliotecas y servicios. Esto abarca desde aplicaciones de Windows Forms y ASP.NET hasta proyectos basados en .NET Core y Mono. La compatibilidad con Unity y Xamarin también lo convierte en una opción valiosa para los desarrolladores de juegos y aplicaciones móviles.
5. **Depuración y Pruebas Integradas:** Rider ofrece un conjunto robusto de herramientas de depuración que permiten a los desarrolladores rastrear y solucionar problemas en sus aplicaciones de manera eficiente. Además, facilita la ejecución de pruebas unitarias y de integración, lo que contribuye a mejorar la calidad del código.
6. **Soporte para las Últimas Tecnologías .NET:** Rider se mantiene actualizado con las últimas tecnologías y marcos de trabajo de .NET, lo que significa que los desarrolladores pueden aprovechar las características y ventajas más recientes en sus proyectos.

En resumen, JetBrains Rider es una herramienta esencial para cualquier desarrollador .NET que busca una solución completa, poderosa y multiplataforma para el desarrollo de aplicaciones. Su amplio conjunto de características, compatibilidad con varios lenguajes y tipos de proyectos, así como su enfoque en la productividad, hacen que Rider sea una opción excepcional para la creación de aplicaciones .NET de alta calidad de manera eficiente.

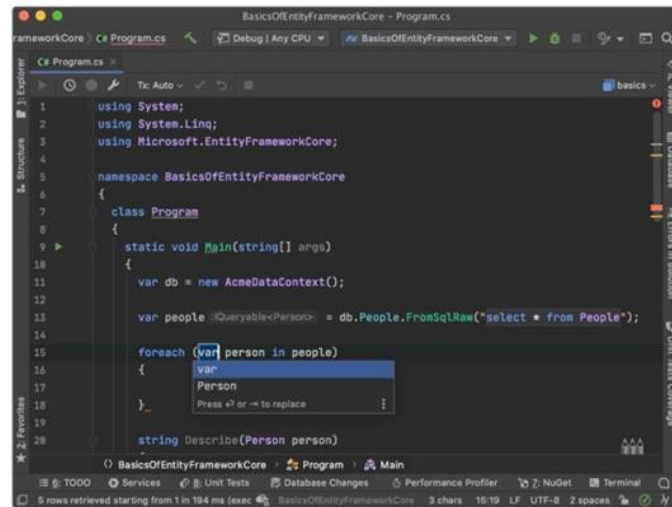


Figura 22. Interfaz de JetBrains Rider

Capítulo III. Procedimiento y descripción de las tareas realizadas

Documentos de apoyo para usuarios

Se realizó la redacción de documentos para que se pudiera entender mejor el funcionamiento de la aplicación y los componentes que la integran.

El primero de nuestros documentos es una introducción a la simulación de circuitos a través de la aplicación iCircuit. En esta guía, se ofrece un detallado enfoque sobre cómo crear, diseñar y compartir documentos de manera sencilla y eficaz. Además, hemos preparado otros documentos que profundizan en la explicación exhaustiva del funcionamiento de componentes clave, como el capacitor, la resistencia, los diodos, el switch, el multímetro y el push button. Estos documentos no solo explican el funcionamiento de dichos componentes, sino que también indican su ubicación en la aplicación y describen cómo se pueden simular con facilidad. Estos documentos ilustran de manera concluyente la capacidad y eficiencia de iCircuit para realizar simulaciones precisas y efectivas de circuitos.

Integración de la Galga Extensiométrica

Las galgas extensiométricas son dispositivos que modifican su resistencia en respuesta a fuerzas aplicadas, como cargas, momentos, presión, aceleración y vibración. Debido a que este cambio en la resistencia es muy pequeño, es común conectarlas en un circuito conocido como "Puente de Wheatstone". Este puente puede configurarse de varias maneras, como un puente completo con cuatro galgas activas, un medio puente con dos galgas activas y dos resistencias, o un cuarto de puente con una galga activa y tres resistencias.

La cantidad de galgas activas en el puente y su disposición en relación con la fuerza aplicada determinarán el tipo de puente utilizado, su aplicación específica y cómo se calcula la deformación resultante. La elección precisa de la configuración del puente es esencial para garantizar mediciones precisas en diversas aplicaciones donde se utilizan galgas extensiométricas.

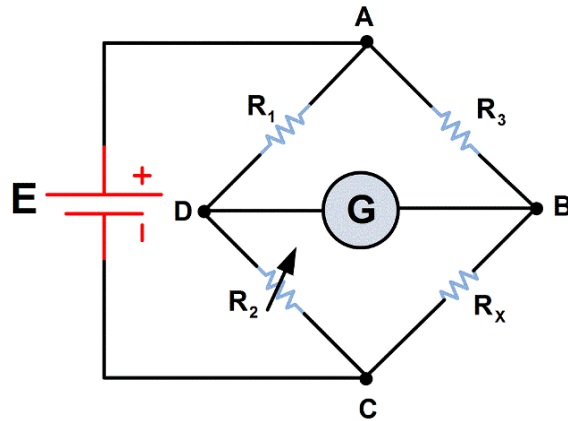


Figura 24. Puente de Wheatstone

El voltaje de salida de un puente de Wheatstone es directamente proporcional a la excitación del puente y muy sensible a las pequeñas deformaciones mecánicas que provocan desequilibrios de resistencia en una o más patas de un puente. Debido a que la deformación mecánica resultante de una fuerza aplicada es un porcentaje muy pequeño, la deformación correspondiente que representa a menudo se expresa como 10^{-6} o micro deformación.

$$\text{micro-strain}(\mu\varepsilon) = \Delta L/L * 10^{-6}$$

Formula de micro deformación

La deformación (ε), en el contexto de la mecánica de materiales, es una medida fundamental que nos permite comprender cómo un material responde a la aplicación de fuerzas externas. Esta magnitud se calcula como un cambio fraccional en las dimensiones del material, ya sea en longitud (largo), ancho o alto, como resultado de la aplicación de una fuerza a lo largo de esa dimensión específica. Es importante destacar que la deformación puede tomar valores tanto positivos ($+\varepsilon$) como negativos ($-\varepsilon$), lo que significa que un material puede estirarse o comprimirse en respuesta a una carga aplicada. La magnitud de la deformación es, en general, extremadamente pequeña y se expresa típicamente como un porcentaje del cambio con respecto a la dimensión original. Este concepto es esencial en ingeniería y ciencia de materiales, ya que nos ayuda a comprender cómo los materiales se comportan bajo carga, lo que a su vez influye en el diseño de estructuras, la seguridad de productos y una variedad de aplicaciones industriales y tecnológicas. La capacidad de medir y controlar la deformación

es crucial para garantizar la integridad y el rendimiento adecuado de numerosos componentes y materiales en una amplia gama de aplicaciones.

“En la mayoría de las aplicaciones del mundo real, las mediciones de deformación rara vez se encuentran por encima de unos pocos mili-deformación ($\varepsilon = 0.003$ o alrededor de $3000\mu\varepsilon$).”

Normalmente se utilizan dos mediciones del sensor para resolver la tensión o la carga: la señal de salida del puente del sensor V_o y la excitación del puente del sensor V_{ex} .

La sensibilidad, un concepto crítico en la ingeniería y la medición de instrumentos, se manifiesta de diferentes maneras según la aplicación. Al igual que la sensibilidad de un puente, que refleja cuánto cambia la magnitud de su salida en relación con un voltio de excitación a escala completa, la sensibilidad de un medidor de tensión, también conocido como Factor de Calibre (GF, por sus siglas en inglés), se relaciona con cómo cambia su resistencia en proporción a la tensión aplicada. Es decir, el GF nos proporciona información sobre cuánto se modifica la resistencia del medidor de tensión en respuesta a las variaciones de la tensión aplicada.

En el caso de las galgas extensiométricas metálicas, que son comunes en numerosas aplicaciones de medición, el factor de calibre suele tener un valor cercano a 2.0. Esto significa que el cambio en la resistencia de la galga extensiométrica es aproximadamente el doble del cambio en sus dimensiones cuando se le aplica una fuerza. Este valor de GF es una característica importante para considerar, al seleccionar un medidor de tensión para una aplicación específica, ya que afecta directamente la capacidad del medidor para medir con precisión las tensiones o deformaciones en un material o estructura.

Es relevante destacar que, si bien un valor de GF de alrededor de 2.0 es común, este puede variar ligeramente según la composición y el diseño de la galga extensiométrica, así como según las condiciones de uso. Esta variación en el GF puede influir en la precisión de las mediciones y debe tenerse en cuenta en aplicaciones donde se requiere una alta exactitud en la medición de tensiones relativas. Por lo tanto, comprender y calibrar adecuadamente el Factor de Calibre es esencial para garantizar mediciones precisas en una amplia gama de aplicaciones de ingeniería y ciencia.

Diferencial de Resistencia
$\Delta R = R * GF * \varepsilon$
R = Resistencia
GF = Factor de Calibre
ε = Deformación

Tabla 2. Formula “Diferencial de Resistencia”

La integración exitosa del sensor en la aplicación comenzó con una investigación exhaustiva sobre el sensor en cuestión, incluyendo sus funciones y su diagrama de conexión. Esto permitió obtener un conocimiento detallado sobre sus características, como su tamaño físico y la cantidad de pines que posee. Esta información es fundamental para diseñar la interfaz de conexión y asegurar una integración efectiva del sensor en la aplicación.

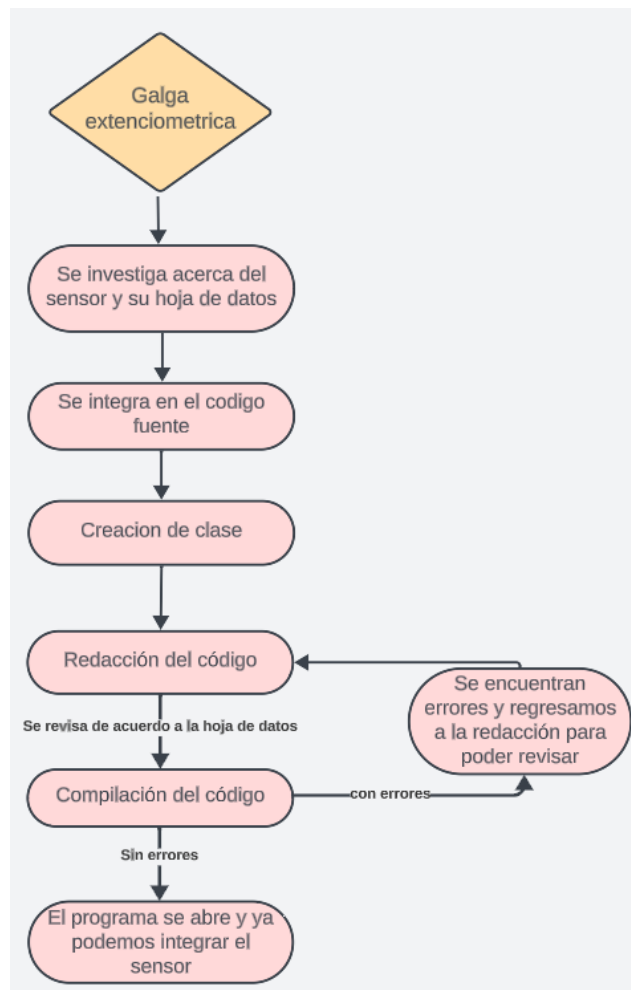


Figura 25. Diagrama de Flujo de la Galga Extensiométrica

Implementación del Sensor SCP1000

El SCP1000 realiza un procedimiento de datos casi completo en el chip. Los datos de salida de presión y temperatura se calibran y compensan internamente. Las únicas operaciones sobre los datos de salida necesarias para obtener la presión en (Pa) y la temperatura en (°C) es la multiplicación simple con constantes.



Figura 26. Sensor SCP1000

La comunicación entre el SCP1000 y su microcontrolador anfitrión (μC) se basa en la interfaz en serie, una línea de interrupción y pines especializados utilizados para activar funciones especiales. La interfaz serial permite operaciones de lectura y escritura de registros y la línea de interrupción señala eventos que requieren la intervención del host.

Funciones Compatibles

En la siguiente tabla se presenta las diferentes versiones y características compatibles de SCP1000.

Features	SCP1000-D01	SCP1000-D11
Supply voltage	2.4 V – 3.3 V	2.4 V – 3.3 V
Measuring range	30 kPa – 120kPa	30 kPa – 120kPa
Resolution ¹	1.5 Pa	1.5 Pa
Interface	SPI max 500 kHz	TWI max 400 kHz
Temperature output	Yes	Yes
Clock	Internal	Internal
Marking on top of component	D01	D11

¹ typical value in high resolution measurement mode

Figura 27. Funciones Compatibles del SCP1000

Operación

El sensor de presión SCP1000 ofrece una versatilidad excepcional gracias a sus 4 modos de medición diferentes, además de las opciones de modo de espera y apagado. Esta amplia gama de modos garantiza que el SCP1000 sea una herramienta adaptable y adecuada para diversas aplicaciones.

En todos estos modos de medición, el sensor proporciona una precisión excepcional con una longitud de palabra de salida de presión de 19 bits. Esto significa que puede capturar y transmitir datos de presión con un alto nivel de detalle, lo que es esencial para aplicaciones que requieren mediciones precisas de la presión atmosférica, como la meteorología o la navegación.

Además de la precisión en la medición de presión, el SCP1000 también ofrece una lectura de temperatura precisa con una longitud de palabra de salida de temperatura de 14 bits. Esto permite la monitorización confiable de las variaciones de temperatura ambiental en tiempo real, lo que es crucial en numerosas aplicaciones, desde el control de climatización hasta la gestión de procesos industriales.

En resumen, el sensor de presión SCP1000 es una solución completa y versátil para la medición de presión y temperatura, con múltiples modos de operación que se adaptan a diversas necesidades. Su alta resolución y precisión hacen que sea una elección ideal para una amplia gama de aplicaciones científicas y comerciales.

Puesta en marcha

La secuencia de inicialización del SCP1000 se realiza cada vez que el sensor:

- Este encendido
- Despertar del modo de pagado
- Despertar después del reinicio

Durante el arranque, las fuentes de alimentación deben estabilizarse en el rango especificando antes de que los datos de configuración y calibración puedan cargarse desde la memoria no volátil a los registros volátiles. La secuencia de inicio se divide en 3 fases que se describen en la siguiente figura.

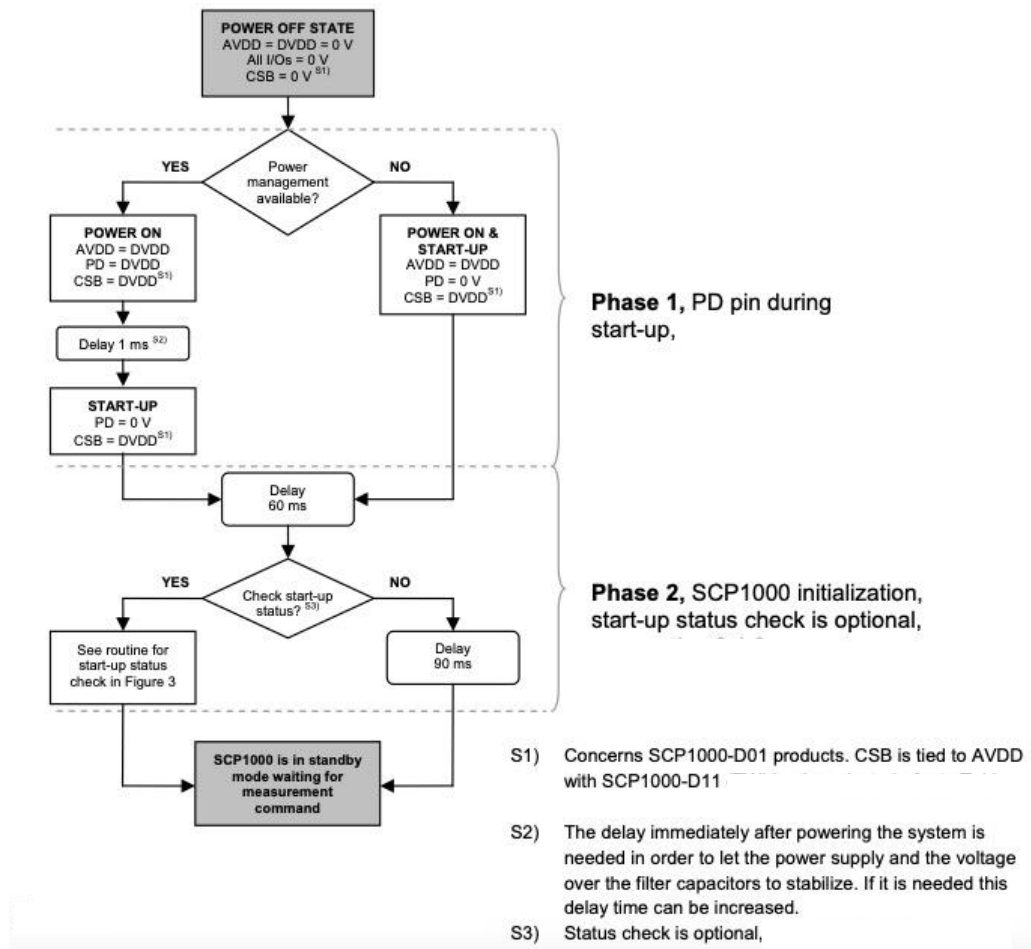


Figura 28. Secuencia de iniciación del SCP1000

Lectura de la presión y la temperatura

Después de que el pin DRDY haya señalado la disponibilidad de nuevos datos de medición, se recomienda leer los datos de salida inmediatamente en el siguiente orden:

- Leer el registro TEMPOUT (datos de temperatura en bits, en caso de que los datos de temperatura no sean necesarios, este paso se puede omitir)
- Leer el registro DATARD8 (los bits [2:0] contienen el MSB de los datos de presión)
- Leer el registro DATARD16 (contiene los 16 LSB de los datos de presión)

Los datos de presión se entregan en formato de números enteros. Cuando se utiliza el SCP1000 dentro de su rango de operación nominal (30 a 120 kPa), su salida varía entre 120,000 y 480,000 unidades. La conversión de esta salida del formato decimal a la unidad estándar de presión en pascales (Pa) se realiza de la siguiente manera:

Ecuación 1

$$Pres[Pa] = \frac{Pres[dec]}{4} = 0.25 * Pres[dec]$$

Donde Pres[dec] es la presión leída desde SCP1000 en formato decimal.

Los datos de temperatura se presentan en formato complemento a 2 y se convierte de formato decimal a [°C] de la siguiente manera:

Ecuación 2

$$Temp[°C] = \frac{Temp[dec]}{20} = 0.05 * Temp[dec]$$

Donde Temp[dec] es la temperatura leída de SCP1000 en formato decimal.

Ejemplos de conversión de temperatura a [°C]

En caso de que el bit de signo sea cero, “0” (Consulte la siguiente figura), el valor binario TEMPOUT se convierte a un valor decimal y luego a [°C] de la siguiente manera:

An example TEMPOUT bit pattern for positive temperature																
Bit#	B15	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
~[°C]			s	204.8	102.4	51.2	25.6	12.8	6.4	3.2	1.6	0.8	0.4	0.2	0.1	0.05
Data bit #	x	x	s	T12	T11	T10	T9	T8	T7	T6	T5	T4	T3	T2	T1	T0
TEMPOUT, raw data			0	0	0	0	1	0	0	0	1	0	1	1	1	0
s = sign bit																
x = not used bit																

Figura 29. Ejemplo de patrón de bits TEMPOUT de datos de temperatura positivos.

El valor binario de la figura anterior se puede convertir normalmente a un valor decimal porque el bit de signo es cero:

Bin “00 0010 0010 1110” → dec “558”

→ conversión a [°C]: $558/20 = 27.9$ °C

En caso de que el bit de signo sea uno “1” (Consulte la siguiente figura), el valor binario TEMPOUT se convierte a un valor decimal y luego a [°C] de la siguiente manera:

An example TEMPOUT bit pattern for negative temperature																
Bit#	B15	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
~[°C]			s	204.8	102.4	51.2	25.6	12.8	6.4	3.2	1.6	0.8	0.4	0.2	0.1	0.05
Data bit #	x	x	s	T12	T11	T10	T9	T8	T7	T6	T5	T4	T3	T2	T1	T0
TEMPOUT, raw data			1	1	1	1	1	1	1	0	0	1	0	1	0	0
TEMPOUT, inverted data			0	0	0	0	0	0	0	1	1	0	1	0	1	1
Add '1' (one LSB) to inverted TEMPOUT data																+1
Absolute value for TEMPOUT data			0	0	0	0	0	0	0	1	1	0	1	1	0	0
s = sign bit																
x = not used bit																

Figura 30. Ejemplo de patrón de bits TEMPOUT de datos de temperatura negativos.

Los datos TEMPOUT sin procesar (bin “11 1111 1001 0100”) no se convierten a valores decimales porque el bit de signo es “1”. Primero se invierte el patrón de bit (bin “00 0000 0110 1011”) y luego se agrega 1 LSB al patrón de bits invertido. El patrón de bits resultante se presenta en la última fila de la figura anterior (valor absoluto para datos TEMPOUT), que normalmente se convierte a valor decimal:

bin “00 0000 0110 1100” → dec “108” → dec “-108” (porque el bit de signo es “1”) → conversión a [°C]: $-108/20 = -5,4$ °C

Descripción del registro

Tipos de registros SCP1000

- Registros de acceso directo
- Registros de acceso indirecto
- Registros EEPROM

Lista de registros de acceso directo e indirecto de SCP1000 y registros EEPROM.

Address		Name	Description	Mode (R, W, RW)	Register access	Width[bits]
SPI	TWI					
0x00		REVID	ASIC revision number	R	Direct	8
0x01		DATAWR	Indirect register access data	RW	Direct	8
0x02		ADDPTR	Indirect register access pointer	RW	Direct	8
0x03		OPERATION	Operation register	RW	Direct	8
0x04		OPSTATUS	Operation status	R	Direct	8
0x06		RSTR	ASIC software reset	W	Direct	8
0x07		STATUS	ASIC top-level status	R	Direct	8
0x1F	0x7F	DATARD8	Pressure output data (MSB) or 8 bit data read from EEPROM	R	Direct	8
0x20	0x80	DATARD16	Pressure output data (LSB) or 8-bit data read from indirect register	R	Direct	16
0x21	0x81	TEMPOUT	14-bit temperature output data	R	Direct	16
	0x00	CFG	Configuration register	RW	Indirect	8
	0x05	TWIADD	TWI address	W	Indirect	8
	0x29	USERDATA1	User data	RW	EEPROM	8
	0x2A	USERDATA2	User data	RW	EEPROM	8
	0x2B	USERDATA3	User data	RW	EEPROM	8
	0x2C	USERDATA4	User data	RW	EEPROM	8

Register address in hex format.

RW – Read / Write register, R – Read only register.

Figura 31. Descripción de Registro

Interfaces serie

La comunicación entre el sensor SCP1000 y el controlador maestro se lleva a cabo mediante la transferencia de datos en serie, respaldada por una línea de interrupción dedicada (pin DRDY). Dependiendo del modo de operación, también es posible utilizar un pin de disparo externo (TRIG) en la interfaz en serie. El SCP1000 ofrece dos opciones de interfaces en serie: SPI y TWI (similar a I2C). Es importante resaltar que, para cada producto específico, solo se habilita una interfaz particular mediante una programación previa en la fábrica.

En ambos buses, SPI y TWI, el SCP1000 opera como un dispositivo esclavo. En este contexto, el sensor se configura en el modo esclavo SPI.

CAPITULO IV. RESULTADOS

Simulaciones

Con el propósito de asegurarnos de que los componentes desarrollados en el proyecto funcionan de manera adecuada, llevamos a cabo un conjunto de simulaciones. Estas simulaciones no solo se enfocaron en la teoría, sino que también involucraron ejercicios prácticos que nos permitieron comprobar y verificar los resultados en un entorno controlado. Esto nos proporcionó una evaluación completa del rendimiento de los elementos desarrollados, garantizando su eficacia y confiabilidad en situaciones reales.

Simulación de Galga Extensiométrica

En el circuito que se presenta a continuación, se ofrece una representación visual del funcionamiento de la galga extensiométrica, y se lleva a cabo una exhaustiva verificación de su correcta operación.

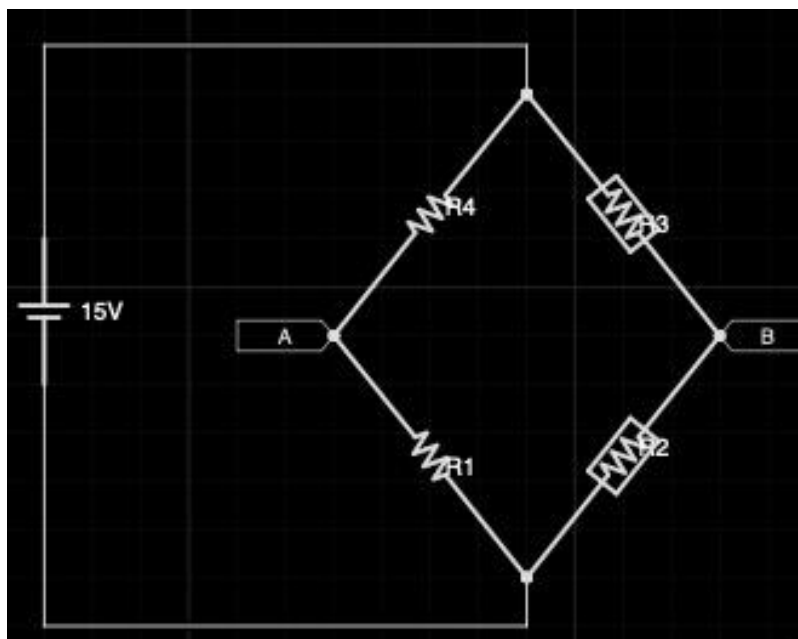


Figura 32. Circuito (Punto de Wheatstone)

En este ejemplo, realizaremos el cálculo del valor de voltaje de salida del puente de Wheatstone, considerando los valores indicados en la figura que se presenta a continuación.

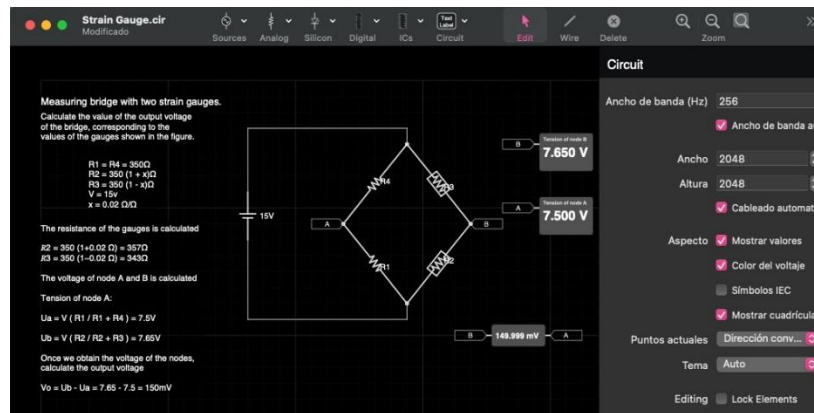


Figura 33. Ejercicio de Comprobación

En la figura mostrada, se aprecian dos resistencias con un valor nominal de 350 ohmios identificadas como R1 y R4. Acompañando a estas resistencias, encontramos dos galgas extensiométricas, una en estado de contracción y la otra en estado de expansión.

La resistencia de la galga R2 se calcula multiplicando su resistencia base por un factor que se obtiene al sumar 1 al factor de calibración. Este proceso nos proporcionará el valor de su resistencia específica.

Para la galga R3, seguimos un procedimiento similar al de R2, pero en este caso utilizamos un factor de calibración negativo para calcular su resistencia resultante.

Después calculamos el voltaje de los nodos A y B

$$Ua = V (R1 / (R1 + R4)) = 7.5V$$

$$Ub = V (R2 / (R2 + R3)) = 7.65V$$

Figura 34. Voltaje de nodos

Una vez que hemos obtenido los valores de voltaje en los nodos, estaremos en condiciones de calcular el voltaje de salida.

$$Vo = Ub - Ua = 7.65 - 7.5 = 150mV$$

Figura 35. Voltaje de salida

Ahora podemos confirmar que la galga está funcionando adecuadamente, ya que durante la simulación hemos obtenido los datos esperados sin ningún inconveniente.

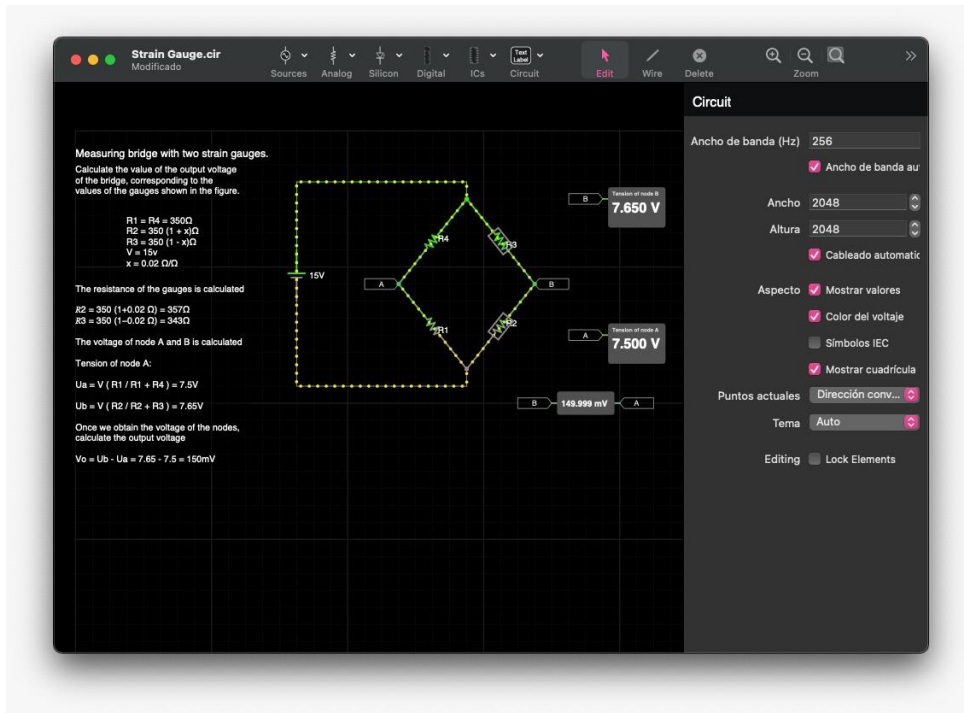


Figura 36. Ejercicio ejecutándose

Simulación Sensor SCP1000

El sensor de Presión Absoluta se basa en el protocolo de comunicación SPI para su funcionamiento. Para asegurar un funcionamiento óptimo y sin contratiempos, se llevará a cabo el desarrollo de un código específico en Arduino.

```
Código
1 #include <SPI.h>
2
3 const int PRESSURE = 0x1F;
4 const int PRESSURE_LSB = 0x20;
5 const int TEMPERATURE = 0x21;
6 const byte READ = 0xFC;
7 const byte WRITE = 0xF2;
8
9 const int dataReadyPin = 6;
10 const int chipSelectPin = 7;
11
12 void setup()
13 {
14   Serial.begin(9600);
15
16   SPI.begin();
17
18   pinMode(dataReadyPin, INPUT);
19   pinMode(chipSelectPin, OUTPUT);
20
21   writeRegister(0x02, 0x2D);
22   writeRegister(0x01, 0x03);
23   writeRegister(0x03, 0x02);
24   delay(1000);
25 }
26
27 void loop()
28 {
29   writeRegister(0x03, 0x00);
```

Figura 37. Código Arduino de esta practica

El código se incorpora directamente en la misma aplicación, eliminando la necesidad de transferir datos manualmente y, además, proporciona información detallada sobre posibles errores que puedan surgir. Para observar el código le damos clic en el Arduino.

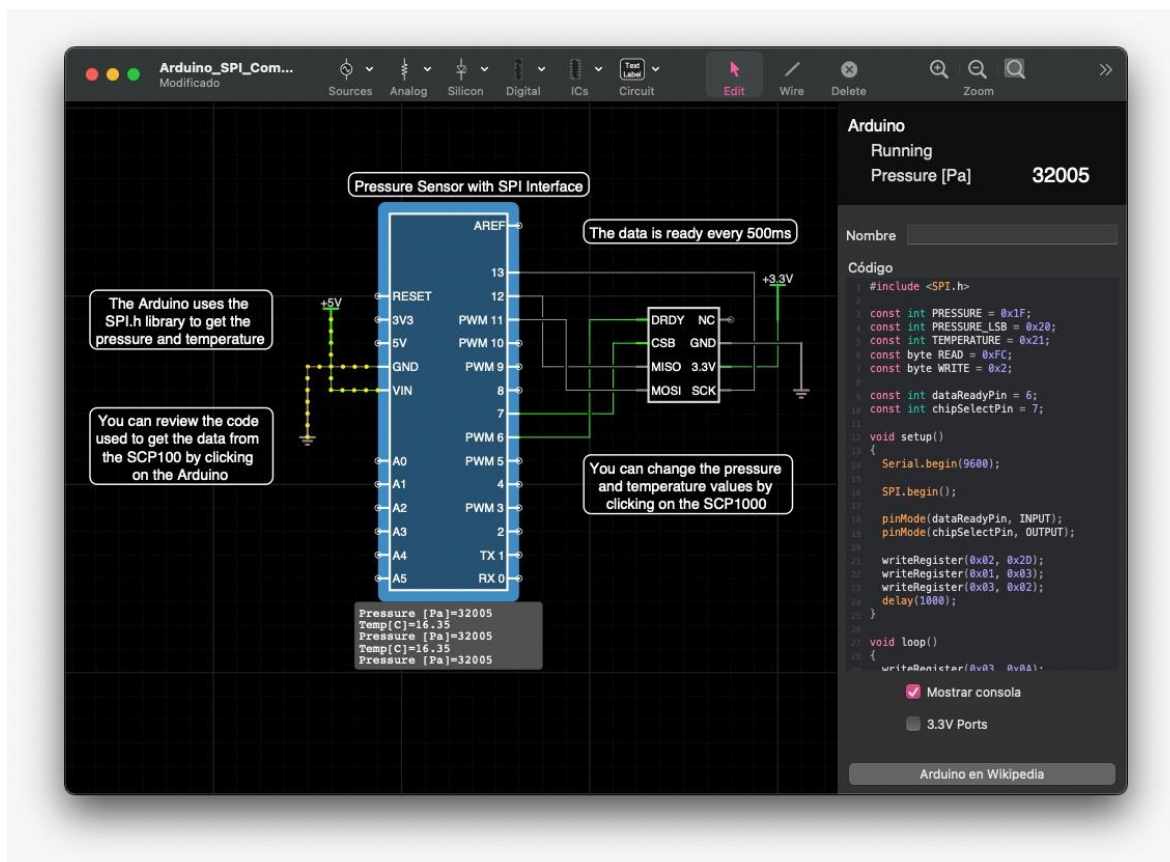


Figura 38. Definición de código Arduino en el editor

Durante la simulación del circuito, se evidencia un funcionamiento fluido y sin complicaciones. La característica más destacada es la posibilidad de ajustar los valores de presión y temperatura mediante un simple clic en el sensor. Esta funcionalidad no solo es práctica, sino que también permite experimentar y observar los efectos de variar la temperatura y presión en tiempo real. Esta interactividad facilita la comprensión del comportamiento del sistema y la identificación de su respuesta ante diversas condiciones.

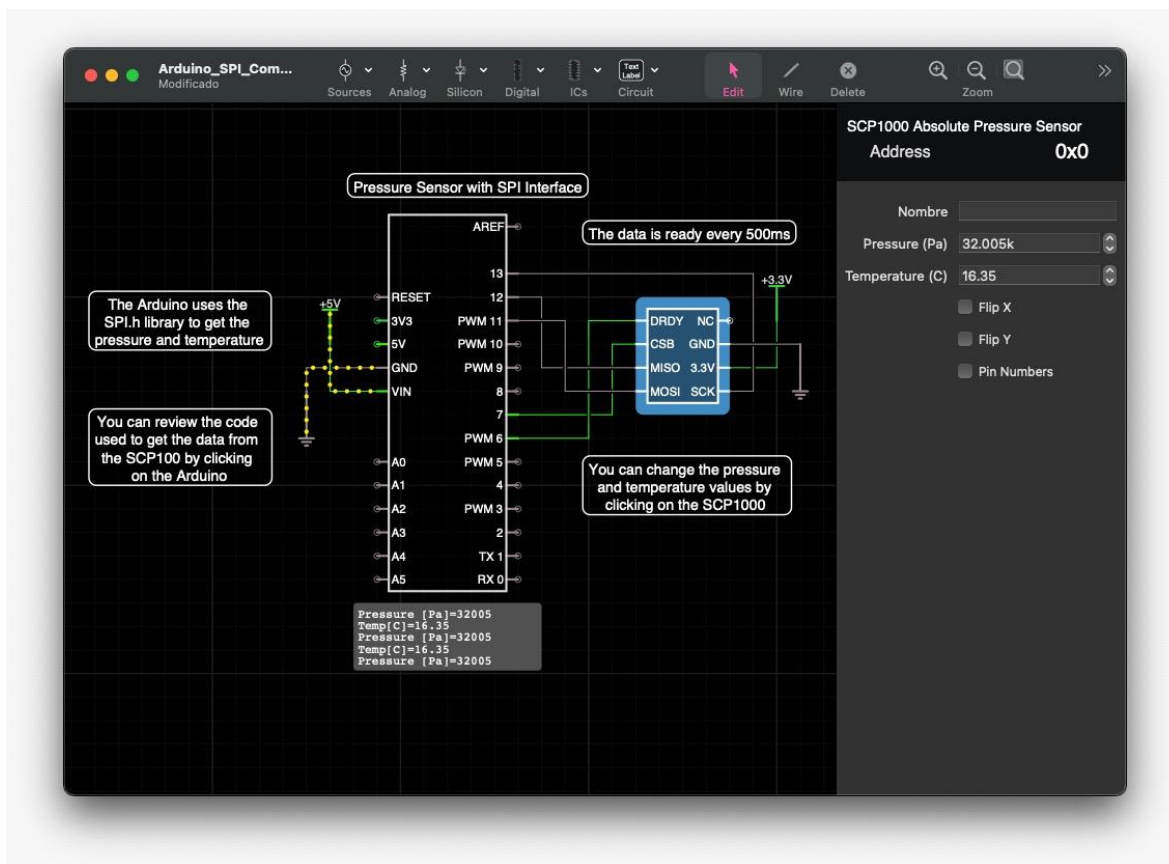


Figura 39. Ejemplo de simulación ejecutándose

Conclusiones

En conclusión, la integración y simulación de sensores analógicos y digitales desempeñan un papel crucial en el mundo de la electrónica y la ingeniería. Los sensores analógicos proporcionan mediciones precisas y continuas, ideales para aplicaciones donde la precisión es esencial, como sistemas de control y mediciones de alta fidelidad. Por otro lado, los sensores digitales ofrecen una salida discreta que es adecuada para sistemas donde la información se procesa digitalmente, lo que facilita su integración con microcontroladores y sistemas embebidos.

La simulación se revela como un enfoque esencial para comprender y validar el funcionamiento de estos sensores en diferentes aplicaciones y escenarios. Facilita la evaluación de su comportamiento, la detección de posibles problemas y la optimización de su rendimiento sin la necesidad de construir prototipos físicos costosos.

En este contexto, la combinación de sensores analógicos y digitales con herramientas de simulación como iCircuit se traduce en un enfoque poderoso y eficiente para el desarrollo de sistemas electrónicos. Esta integración brinda a los diseñadores y desarrolladores la capacidad de explorar y analizar sistemas en un entorno virtual.

Competencias Desarrolladas

Durante la estancia en Krueger Systems, se desarrollaron y aplicaron competencias que abarcaron dos ámbitos fundamentales: la electrónica y la informática.

En el campo de la electrónica, se llevó a cabo un exhaustivo estudio de una amplia variedad de sensores que hacen uso de protocolos de comunicación I2C o SPI. Estos sensores tienen aplicaciones diversificadas, desde la medición de temperatura y presión hasta la detección de movimiento y la captura de datos ambientales. Esta inmersión en el mundo de los sensores no solo proporcionó un entendimiento profundo de su funcionamiento y su capacidad para comunicarse con otros dispositivos, sino que también permitió la selección precisa del sensor más adecuado para aplicaciones específicas, enriqueciendo nuestra capacidad de toma de decisiones en proyectos futuros.

Por otro lado, en el ámbito informático, se aplicaron habilidades clave relacionadas con la programación orientada a objetos. Estas competencias resultaron esenciales para la organización y manipulación eficiente de datos, lo que habilitó la incorporación de funcionalidades avanzadas en diversas aplicaciones y sistemas informáticos. La programación orientada a objetos se destacó como una herramienta esencial para abordar la intrincada complejidad intrínseca al desarrollo de software.

En resumen, la experiencia en Krueger Systems nos permitió adquirir un profundo conocimiento tanto en el ámbito de la electrónica como en la informática, equipándonos con habilidades valiosas que serán de gran utilidad en futuros proyectos y desafíos en ambas disciplinas.

Fuentes de Información

Marujita. (2023). Bus de datos. *Muy Tecnológicos*.

<https://muytecnologicos.com/diccionario-tecnologico/bus-de-datos>

Paguayo. (2022, 25 agosto). *Protocolo I2C (Inter Integrated Circuit) - MCI capacitación*.

MCI Capacitación. <https://cursos.mcielectronics.cl/2022/08/23/i2c/>

¿Qué son las galgas extensiométricas? ¿Cómo se usan? (s. f.).

<https://es.omega.com/prodinfo/galgas->

[extensiometricas.html#:~:text=Las%20galgas%20extensiom%C3%A9tricas%20son%20una,para%20la%20medici%C3%B3n%20de%20tensiones.](https://es.omega.com/prodinfo/galgas-extensiometricas.html#:~:text=Las%20galgas%20extensiom%C3%A9tricas%20son%20una,para%20la%20medici%C3%B3n%20de%20tensiones.)

Sensor SCP 1000 y su conexión con arduino por SPI. (s. f.).

<http://mindstormsyarduino.blogspot.com/2011/06/sensor-scp-1000-y-su-conexion-con.html>

Engineering, O. (2023). Strain gage technical data. <https://www.omega.com/en-us/>.

<https://www.omega.com/en-us/resources/strain-gauge-technical-data>

Sensors | TTI, Inc. (s. f.). <https://www.tti.com/content/ttiinc/en/resources/product->

[types/sensors.html?utm=1353&gad=1&gclid=Cj0KCQjwj5mpBhDJARIsAOVjBdr1V3zwKaCMTlSN_7KfljrHjwEwvSyRykrZ3mIcSOB4gDmZnCosbZwaArg7EALw_wcB](https://www.tti.com/content/ttiinc/en/resources/product-types/sensors.html?utm=1353&gad=1&gclid=Cj0KCQjwj5mpBhDJARIsAOVjBdr1V3zwKaCMTlSN_7KfljrHjwEwvSyRykrZ3mIcSOB4gDmZnCosbZwaArg7EALw_wcB)