

## Shields v1.0

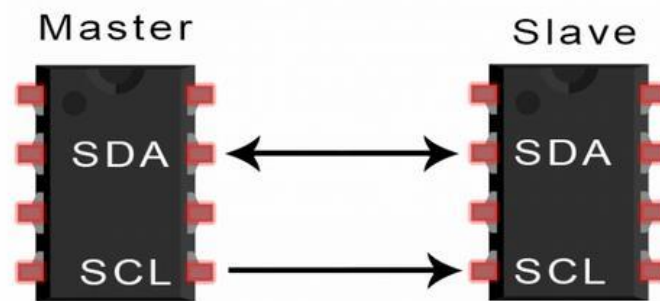
### Ejercicio #1

#### a) Explique el funcionamiento del protocolo I2C

El bus de **comunicaciones I2C** (nombrado a veces como **I cuadrado C**, I<sup>2</sup>C) es un protocolo que se efectúa por medio de DOS hilos, bidireccional y sencillo desarrollado por Philips. Solo necesita 2 cables para transferir información y a través de estos dos hilos pueden conectarse diferentes dispositivos donde algunos de ellos serán **MAESTROS** en cuanto muchos otros dispositivos serán **ESCLAVOS**.

**I2C** combina las ventajas de SPI y UART. Con **I2C**, se pueden conectar varios dispositivos esclavos a un solo dispositivo maestro (como SPI), y varios dispositivos maestros pueden controlar uno o varios dispositivos esclavos. Esto es útil cuando desea que varios microcontroladores registren datos en una sola tarjeta de memoria o muestren texto en una sola pantalla LCD.

Al igual que la comunicación UART, **I2C** solo usa dos cables para transferir datos entre dispositivos:



**SDA (DATOS EN SERIE):** LA LÍNEA UTILIZADA POR EL MAESTRO Y EL ESCLAVO PARA ENVIAR Y RECIBIR DATOS.

**SCL (SERIAL CLOCK):** LA LÍNEA QUE TRANSPORTA LA SEÑAL DEL RELOJ.

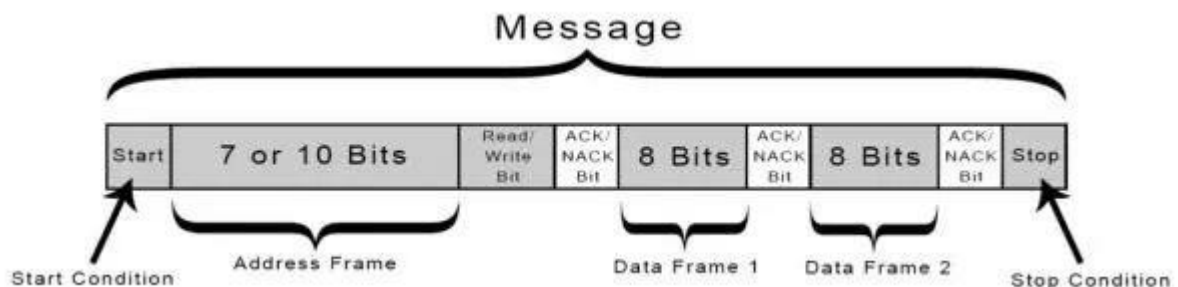
**I2C ES UN PROTOCOLO DE COMUNICACIÓN EN SERIE, POR LO QUE LOS DATOS SE TRANSMITEN BIT A BIT A LO LARGO DE UN SOLO CABLE (CABLE SDA).**

**COMO SPI, I2C ES SÍNCRONO, POR LO QUE LA SALIDA DE BITS SE SINCRONIZA CON EL MUESTREO DE BITS A TRAVÉS DE UNA SEÑAL DE RELOJ COMPARTIDA ENTRE EL MAESTRO Y EL ESCLAVO. LA SEÑAL DEL RELOJ SIEMPRE LA CONTROLA EL ANFITRIÓN.**

Wires Used	2
Maximum Speed	Standard mode= 100 kbps
	Fast mode= 400 kbps
	High speed mode= 3.4 Mbps
	Ultra fast mode= 5 Mbps
Synchronous or Asynchronous?	Synchronous
Serial or Parallel?	Serial
Max # of Masters	Unlimited
Max # of Slaves	1008

## Cómo funciona I2C

Cuando se usa **I2C**, los datos se convierten en mensajes y los mensajes se descomponen en marcos de datos. Cada mensaje tiene una trama de dirección, que contiene la dirección binaria de la estación esclava y una o más tramas de datos que contienen los datos que se transmiten. El mensaje también incluye las condiciones de inicio y parada entre cada trama de datos, bits de lectura / escritura y bits ACK / NACK:



**Condición de inicio:** antes de que la línea SCL cambie de nivel alto a nivel bajo, la línea SDA cambia de nivel alto a nivel bajo.

**Condición de parada:** después de que la línea SCL cambia de nivel bajo a nivel alto, la línea SDA cambia de nivel bajo a nivel alto.

**Trama de dirección:** una secuencia única de 7 o 10 bits para cada estación esclava, que se utiliza para identificar la estación esclava cuando la estación maestra desea comunicarse con ella.

**Bit de lectura / escritura:** un solo bit que especifica si el dispositivo maestro envía datos al dispositivo esclavo (nivel de voltaje bajo) o si el dispositivo esclavo solicita datos (nivel de voltaje alto).

**Bit ACK / NACK:** cada trama del mensaje va seguida de un bit de reconocimiento / no reconocimiento. Si la trama de dirección o la trama de datos se recibe con éxito, el bit ACK se devuelve desde el dispositivo receptor al remitente.

**I2C** no tiene una línea de selección esclava como SPI, por lo que necesita otra forma de que el dispositivo esclavo sepa que se le están enviando datos en lugar de otro dispositivo esclavo. Lo hace por dirección. La trama de dirección es siempre la primera trama después del bit de inicio en un mensaje nuevo. El dispositivo maestro envía la dirección del dispositivo esclavo que se comunica con él a cada dispositivo esclavo conectado a él. Luego, cada dispositivo esclavo compara la dirección enviada desde el dispositivo maestro con su propia dirección. Si la dirección coincide, el bit ACK de bajo voltaje se envía de vuelta al host. Si la dirección no coincide, el dispositivo esclavo no realiza ninguna operación y la línea SDA permanece alta.

## ***Bit de lectura / escritura***

La trama de dirección incluye un bit al final para notificar al dispositivo esclavo si el dispositivo maestro desea escribir datos en él o recibir datos del dispositivo maestro. Si el dispositivo maestro desea enviar datos al dispositivo esclavo, el bit de lectura / escritura es bajo. Si el dispositivo maestro solicita datos del dispositivo esclavo, este bit es alto.

## ***Marco de datos***

Una vez que el dispositivo maestro detecta el bit ACK del dispositivo esclavo, está listo para enviar la primera trama de datos.

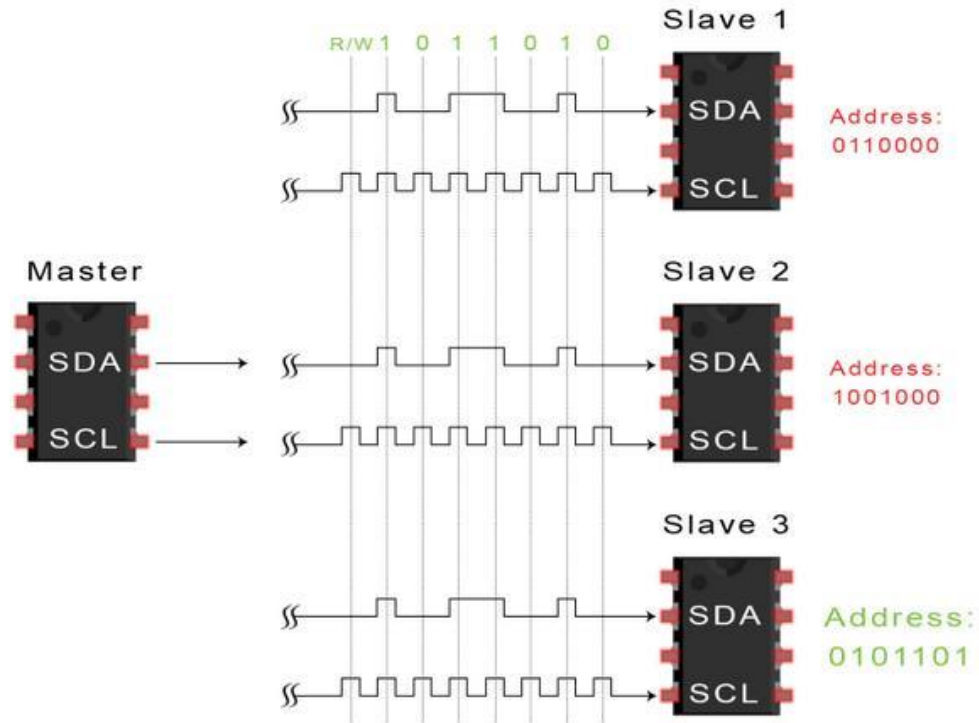
La trama de datos siempre tiene 8 bits de longitud y se envía con el bit más significativo primero. Siga el bit ACK / NACK de cada trama de datos para verificar que la trama se haya recibido correctamente. Antes de enviar la siguiente trama de datos, el maestro o esclavo debe recibir el bit ACK (dependiendo de la persona que envía los datos).

Después de enviar todas las tramas de datos, el dispositivo maestro puede enviar una condición de parada al dispositivo esclavo para detener la transmisión. La condición de parada es que después de que la línea SCL cambia de baja a alta, la línea SDA cambia de baja a alta y la línea SCL permanece alta.

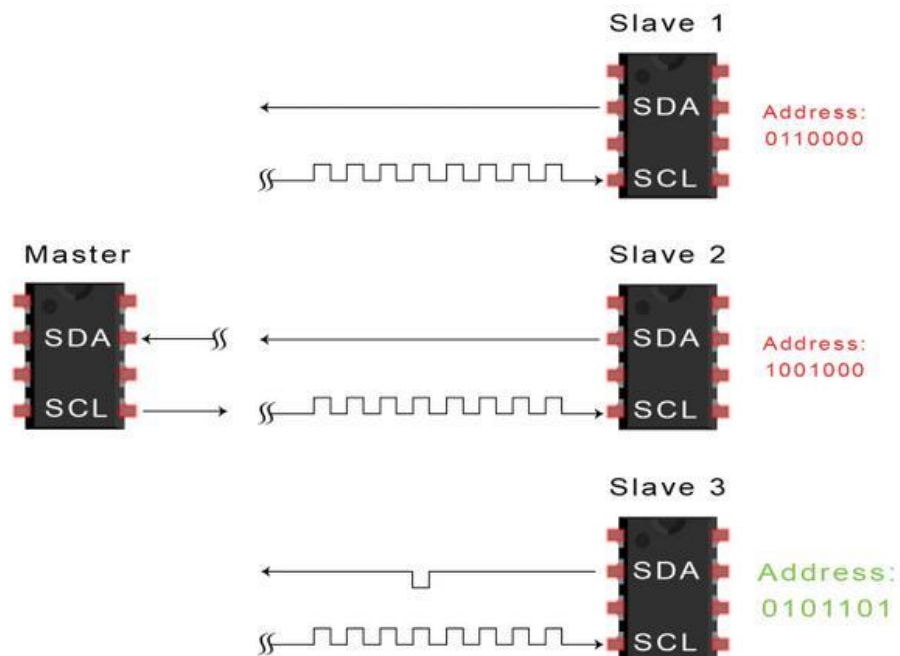
## ***Pasos de transmisión de datos I2C***

1. El host envía datos a cada dispositivo esclavo conectado y luego cambia la señal SDA de alta a baja, y luego cambia SCL de alta a baja.

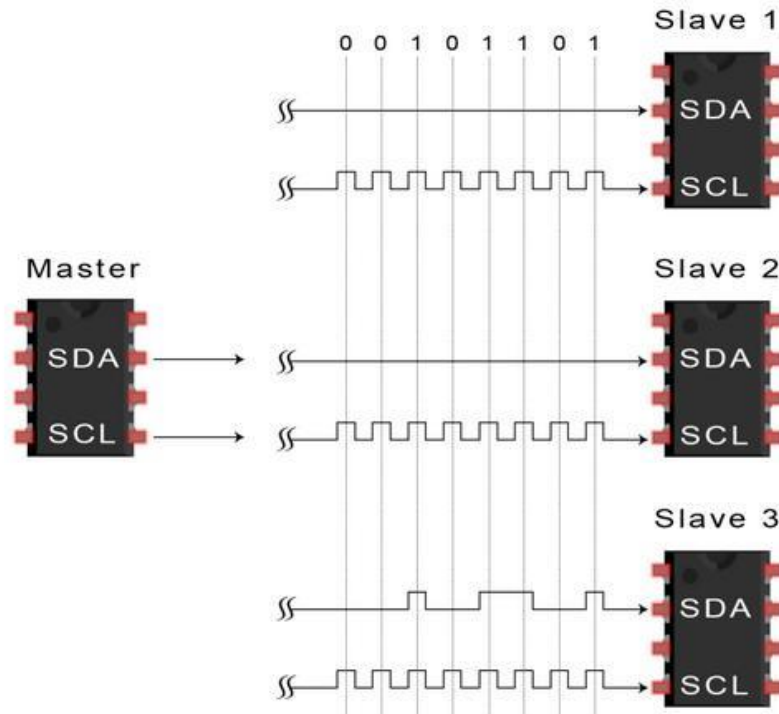
2. El dispositivo maestro envía a cada dispositivo esclavo la dirección de 7 o 10 bits del dispositivo esclavo con el que desea comunicarse y el bit de lectura / escritura:



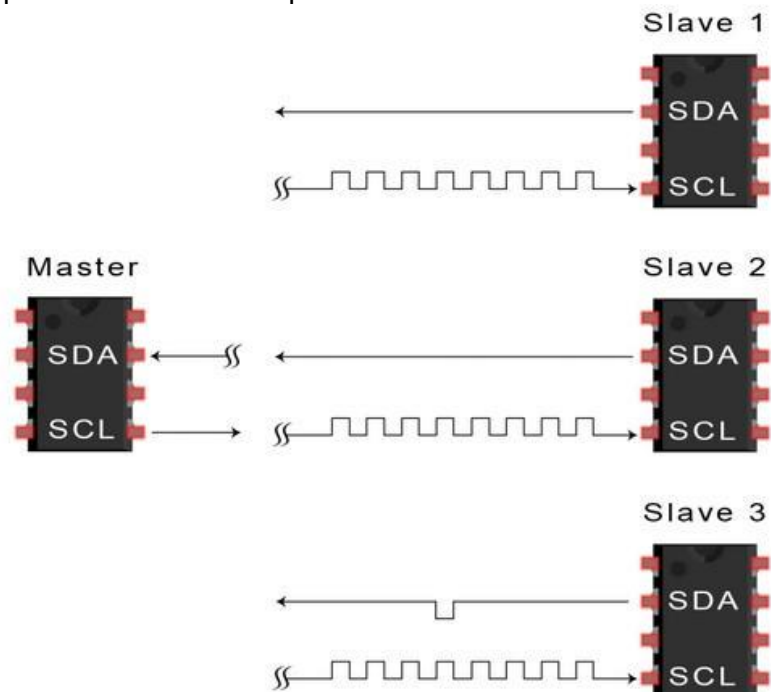
3. Cada dispositivo esclavo compara la dirección enviada por el dispositivo maestro con su propia dirección. Si la dirección coincide, el dispositivo esclavo devuelve el bit ACK bajando la línea SDA. Si la dirección del dispositivo maestro no coincide con la dirección del dispositivo esclavo, el dispositivo esclavo mantiene la línea SDA alta.



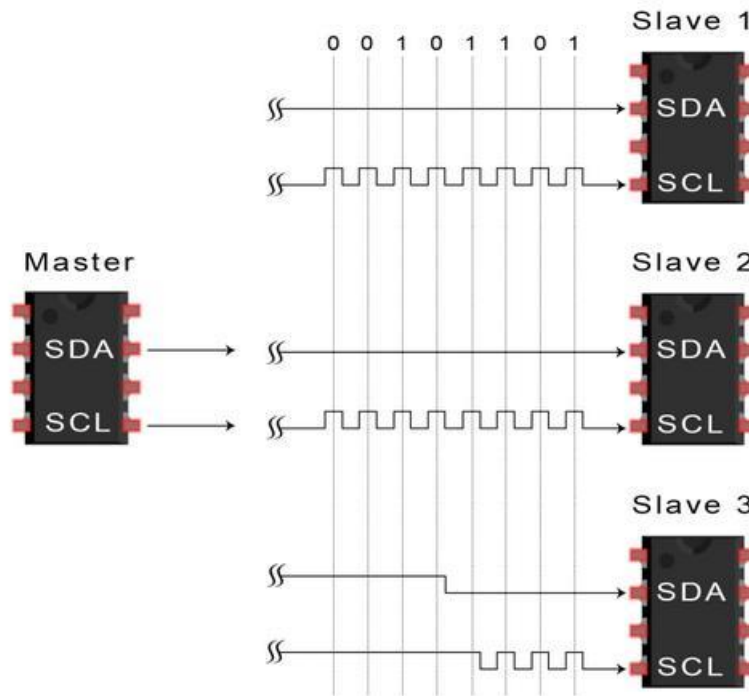
4. El dispositivo maestro envía o recibe tramas de datos:



5. Después de transmitir cada trama de datos, el dispositivo receptor devuelve otro bit ACK a remitente para confirmar la recepción exitosa de la trama:

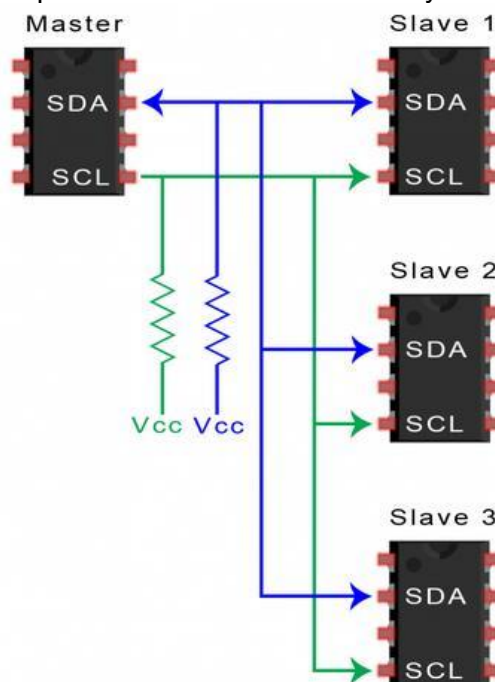


6. Para detener la transmisión de datos, el maestro envía una condición de detención al esclavo cambiando SCL a alto antes de cambiar SDA a alto:



## Maestro único con varios esclavos

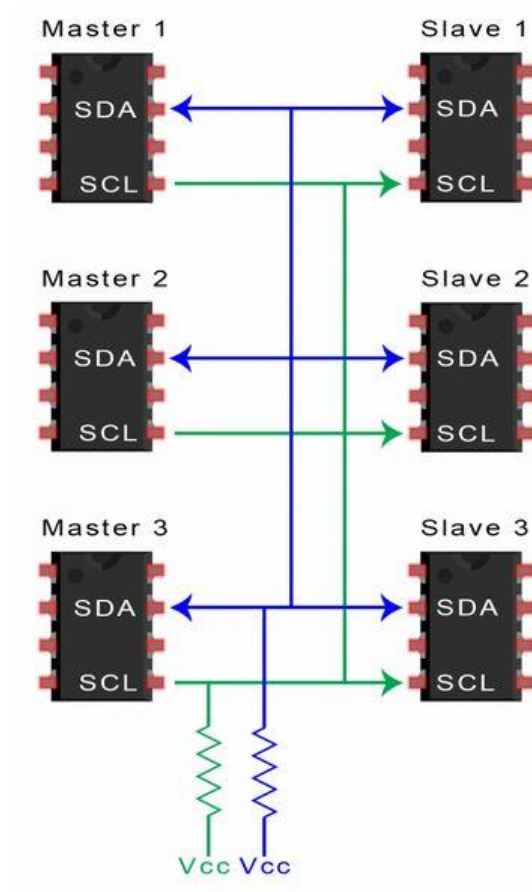
Debido a que **I2C** utiliza direccionamiento, se pueden controlar varios dispositivos esclavos desde un solo dispositivo maestro. Con direcciones de 7 bits, se pueden utilizar 128 (27) direcciones únicas. No es común utilizar direcciones de 10 bits, pero se proporcionan 1.024 (210) direcciones únicas. Para conectar varios dispositivos esclavos a un solo dispositivo maestro, conéctelos así, usando resistencias pull-up de 4.7K ohmios para conectar las líneas SDA y SCL a Vcc:





## ***Hay varios dispositivos esclavos y varios dispositivos maestros***

Se pueden conectar varios dispositivos maestros a un solo dispositivo esclavo o múltiples dispositivos esclavos. Cuando dos dispositivos maestros intentan enviar o recibir datos a través de la línea SDA al mismo tiempo, existe un problema con varios dispositivos maestros en el mismo sistema. Para resolver este problema, cada dispositivo maestro necesita detectar si la línea SDA es baja o alta antes de enviar un mensaje. Si la línea SDA es baja, significa que otro controlador principal ha controlado el bus y el controlador principal debe esperar para enviar un mensaje. Si la línea SDA es alta, la transmisión de información es segura. Para conectar múltiples dispositivos maestros a múltiples dispositivos esclavos, use el siguiente diagrama para conectar las líneas SDA y SCL a Vcc con resistencias pull-up de 4.7K ohmios:



## ***Ventajas y desventajas de I2C***

Comparado con otros protocolos, I2C suena muy complicado, no es fácil de implementar en el programa y causa problemas como pérdida de datos, no respuesta, "muertos, etc.".

### ***Ventajas:***

**Utiliza solo dos cables**

**Admite múltiples servidores maestros y múltiples servidores esclavos**

**El bit ACK / NACK confirma que todas las tramas se han transmitido correctamente**

**El HW no es tan complicado como UART**

**Protocolo conocido y ampliamente utilizado**

### ***Desventajas:***

**La tasa de Transferencia de datos es más lenta que SPI**

**El tamaño del marco de datos esta imitado a 8 bits**

**Implementar HW más complejo que SPI**

## **b) Qué son los sensores resistivos?**

Los Sensores Resistivos son los sensores cuyo principio físico de funcionamiento es la variación de la resistencia eléctrica de un componente eléctrico o electrónico.

Básicamente, un sensor resistivo es un dispositivo que convierte cantidades físicas medidas tales como desplazamiento, deformación, fuerza, aceleración, humedad, temperatura, etc. en valores de resistencia.

Según la variable física que hace variar la resistencia se clasifican en:

- Potenciómetros (Potentiometers)
- Galgas extensométricas (Strain gages)
- Sensores de temperatura de resistencia metálica o RTD (Resistance Temperature Detectors) Termistores [Thermistors (Thermal Resistors)]
- Magnetorresistencias (Magnetoresistors)
- Fotorresistencias o LDR (Light Dependent Resistors)
- Higrómetros (Hygrometers)
- Sensores y detectores de gases
- Sensores de conductividad de líquidos

## **Cómo se conectan a través de un divisor resistivo?**

Un divisor Resistivo ó Divisor de Tensión, es una configuración de circuito eléctrico que reparte la tensión eléctrica de una fuente entre una o más impedancias conectadas en serie.



En electrónica y electricidad se usa para alimentar (proporcionar tensión de alimentación) a un aparato, con una tensión más pequeña que la que proporcionan las pilas o baterías disponibles.

Se trata de un circuito práctico de amplia utilización en electrónica que permite obtener a partir de un generador de tensión de un valor dado, otro generador con una fracción de tensión cualquiera. Mediante este circuito es posible alimentar (proporcionar tensión de alimentación o polarización) a un equipo de bajo consumo o a un componente electrónico como por ejemplo un transistor.

En definitiva, sirve para **obtener una tensión más pequeña partiendo de una tensión mayor**.

## Qué es el acondicionamiento de señales?

El acondicionamiento de señal es un proceso de adquisición de datos que se lleva a cabo mediante un instrumento llamado acondicionador de señal. Ese instrumento convierte un tipo de señal eléctrica o mecánica (señal de entrada) en otro (señal de salida). El objetivo consiste en amplificar la señal y convertirla a otro formato fácil de leer y compatible con fines de adquisición de datos o de control de una máquina.

Un acondicionador de señal ayuda a obtener medidas precisas, como condición esencial para la exactitud de la adquisición de datos o del control de máquinas.

Cuando deseamos realizar una medida, lo primordial es tener un [transductor](#) que pueda transformar la medida física en una medida eléctrica, Esa medida eléctrica es necesario acondicionarla.

Para entender esto, acondicionar una señal significa realizar el siguiente proceso: Convertir la señal, modificar el nivel de la señal, linealización para una respuesta y por ultimo filtrar la señal. Un acondicionador de señal son pasos de adquisición de datos que es llevado acabo con un dispositivo que puede convertir un tipo de señal electrónica a otra. Su principal objetivo es convertir una señal que puede ser difícil de leer y convertirlas a otro formato para que tengan de una manera fácil de leer los datos adquiridos.

La mayoría de estas señales requieren de alguna manera de preparación antes de que sean digitalizadas. Por ejemplo las señales de un [termopar](#), estas señales proporcionan unos niveles muy pequeños de tensión que deben ser amplificados antes que sean digitalizados. Podemos mencionar de otros sensores, como detectores de temperatura mediante resistencia (RTD), los termistores, las galgas extensiométricas y los acelerómetros, toda esta tecnología que se requiere un trabajo de preparación con lleva un acondicionador de señales.

## Etapas del Acondicionamiento de señales

Amplificación.

Filtrado.

Linealización.

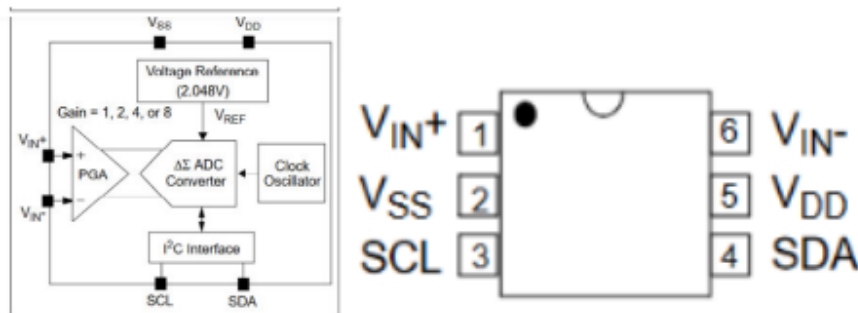
Conversión de **señal**.

Multiplexores.

**c) Cómo funciona el integrado mcp3421 y como lo utilizaría para construir una shield de transducción resistiva?**

El MPC3421 se trata de un conversor analógico-digital de 18 bits con un único canal para señales diferenciales. Las características de este dispositivo son las siguientes:

- 18 bits de resolución.
- Auto calibración interna con offset y ganancia interna en cada conversión.
- Voltaje de referencia interno de 2.048V.
- Amplificador programable integrado.
- Oscilador integrado.
- INL de 10 ppm del fondo de escala.
- Cuatro velocidades de transmisión de datos configurable; 3.75 SPS (18 bits), 15 SPS (16 bits), 60 SPS (14 bits), 240 SPS (12 bits).
- Bajo consumo de funcionamiento.
- Interfaz I2C de 100kHz a 400 kHz.
- Tensión de operación de 2.7V a 5.5V.
- Encapsulado SOT23.



Esta board permite al usuario medir el voltaje de entrada desconocido con una cantidad mínima de preparación.

Muchas almohadillas de prueba, incluidas las SCL y SDA, permiten controlar las comunicaciones de datos de señales I2C.

El usuario puede conectar los pines SDA y SCL a su propia placa PCB y evaluar el rendimiento del dispositivo mcp3421.

