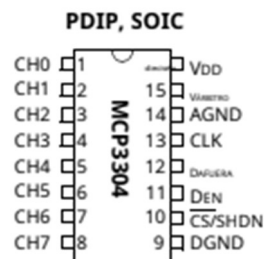


## ¿QUÉ ES EL mcp3304?

El mcp3304 es un microcontrolador de entrada diferencial de 13 bits, convertidor A/D de baja potencia con interfaz serie SPI(Serial Peripheral Interface). El convertidor MCP3304 presenta entradas diferenciales completas y bajo consumo de energía en un paquete pequeño que es ideal para sistemas alimentados por batería y aplicaciones de adquisición de datos remota. También es programable por el usuario para configurarlo en cuatro pares de entradas diferenciales u ocho entradas de un solo extremo.



<b>MCP3304</b>	<b>Símbolo</b>	<b>Descripción</b>
<b>PDIP, SOIC</b>		
1	CH0	Analog Input
2	CH1	Analog Input
3	CH2	Analog Input
4	CH3	Analog Input
5	CH4	Analog Input
6	CH5	Analog Input
7	CH6	Analog Input
8	CH7	Analog Input
9	DGND	Digital Ground
10	$\overline{\text{CS}}/\text{SHDN}$	Chip Select / Shutdown Input
11	D <sub>IN</sub>	Serial Data In
12	D <sub>OUT</sub>	Serial Data Out
13	CLK	Serial Clock
14	AGND	Analog Ground
15	V <sub>REF</sub>	Reference Voltage Input
16	V <sub>DD</sub>	+4.5V to 5.5V Power Supply
—	NC	No Connection

### **A- Entradas analógicas (CH0-CH7)**

Estos pines tienen un rango de voltaje absoluto de VSS- 0.3V a VDD+ 0,3 V. El rango de entrada diferencial de escala completa se define como el valor absoluto de (IN+) - (IN-). Esta diferencia no puede exceder el valor de Vref- Se producirá saturación de 1 LSB o código digital.

### **B-Tierra digital (DGND)**

Conexión a tierra al circuito digital interno. Para garantizar la precisión, este pin debe estar conectado a la misma tierra que AGND. Si hay disponible un plano de tierra analógico, se recomienda conectar este dispositivo al plano de tierra analógico del circuito.

### **C - Select/Shutdown de chip (CS/SHDN)**

El pin CS/SHDN se usa para iniciar la comunicación con el dispositivo cuando se baja. Este pin finalizará una conversión y pondrá el dispositivo en modo de espera de bajo consumo cuando se coloque en alto. El pin CS/SHDN debe colocarse alto entre conversiones y no puede vincularse bajo para conversiones múltiples.

### **D - Entrada de datos en serie (D<sub>IN</sub>)**

El pin de entrada de datos en serie del puerto SPI se utiliza para sincronizar los datos de configuración del canal de entrada. Los datos se bloquean en el flanco ascendente del reloj.

### **E - Salida de datos en serie (D<sub>OUT</sub>)**

El pin de salida de datos en serie SPI se utiliza para cambiar los resultados de la conversión A/D. Los datos siempre cambiarán en el flanco descendente de cada reloj a medida que se realiza la conversión.

### **F - Reloj serie (CLK)**

El pin de reloj SPI se utiliza para iniciar una conversión, así como para registrar cada bit de la conversión a medida que se lleva a cabo.

### **G - Tierra analógica (AGND)**

Conexión a tierra al circuito analógico interno. Para garantizar la precisión, este pin debe estar conectado a la misma tierra que DGND. Si hay disponible un plano de tierra analógico, se recomienda conectar este dispositivo al plano de tierra analógico del circuito.

### **H - Referencia de voltaje ( $V_{REF}$ )**

Este pin de entrada proporciona el voltaje de referencia para el dispositivo, que determina el rango máximo de la señal de entrada analógica y el tamaño de LSB.

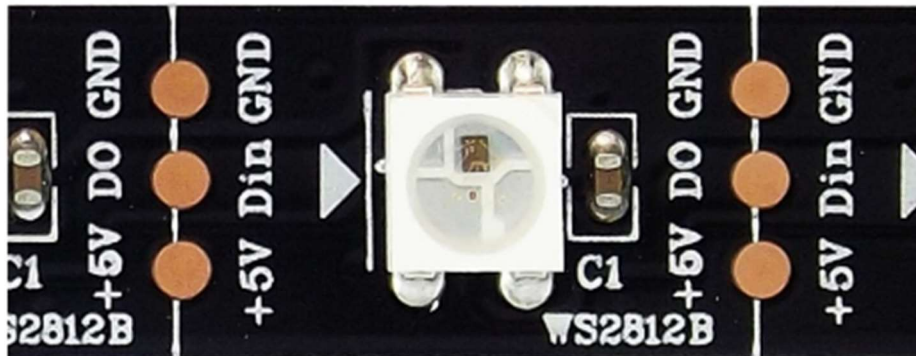
### **I - Fuente de alimentación ( $V_{DD}$ )**

El dispositivo puede funcionar de 2,7 V a 5,5 V, pero el rendimiento de conversión de datos es de 4,5 V a 5,5 V rango de suministro. Para garantizar la precisión, se debe colocar un condensador de derivación de cerámica de 0,1  $\mu$ F lo más cerca posible del pin.

## **EL Led ws2812**

El led WS2812 es un chip led encapsulado de tipo smd 5050 que internamente tiene un driver capaz de controlar 1 led RGB, además, el chip WS2812 posee una estructura interna simple para poder controlar de manera práctica un led RGB. El WS2812 dispone de 4 pines:  $V_{dd}$  (3,5 a 5.3VDC), GND,  $D_{in}$  y  $D_{out}$ . A través de los pines  $D_{in}$  y  $D_{OUT}$  es donde sucede la magia. Los bits de datos que representan el brillo RGB son alimentados en serie en el pasador  $D_{in}$ , y el chip de tiras a cabo los primeros 24 bits (8 bits cada uno de R, G, B) y envía los bits restantes a través del pin  $D_{out}$ . Mediante la conexión de los LED en una cadena, con el  $D_{out}$  de un LED debe ir a la  $D_{in}$  de el siguiente LED, cada led solo necesita 24 bits, y envía el resto de la secuencia de datos afuera al siguiente LED. En

teoría no hay límite para el número de LEDs que puede manejar con una sola línea de datos, la única limitación es que el tiempo que tarda en actualizar todos los LEDs.



Los led RGB se conectan con una fuente de alimentación externa de +5V DC para lo cual se puede seguir un esquema como el de la siguiente imagen: conectar la entrada de + 5V de la tira led RGB al terminal + (positivo) en la fuente de alimentación (no se conecta a la de Arduino), DIN al pin digital D6 del Arduino, y – de nuestra fuente a GND de nuestra cinta led RGB, además de conectar nuestro terminal negativo a un pin GND de nuestro Arduino. Para el uso de la cinta LED, se utiliza la librería NeoPixel de Adafruit (`#include "Adafruit_NeoPixel.h"`)

## Esquema de conexiones

