**Introducción a Comunicación I2C y Eco-procesamiento con Rasoberri Py, ESP32, Arduino UNO**

**I2C – Puerto, Introducción, trama y protocolo**

I2C es un puerto y protocolo de comunicación serial, define la trama de datos y las conexiones físicas para transferir bits entre 2 dispositivos digitales. El puerto incluye dos cables de comunicación, SDA y SCL. Además, el protocolo permite conectar hasta 127 dispositivos esclavos con esas dos líneas, con hasta velocidades de 100, 400 y 1000 kbits/s. También es conocido como IIC ó TWI – Two Wire Interface.

El protocolo I2C es uno de los más utilizados para comunicarse con sensores digitales, ya que a diferencia del puerto Serial, su arquitectura permite tener una confirmación de los datos recibidos, dentro de la misma trama, entre otras ventajas.

La conexión de tantos dispositivos al mismo bus, es una de las principales ventajas. Además si comparamos a I2C con otro protocolo serial, como Serial TTL, este incluye más bits en su trama de comunicación que permite enviar mensajes más completos y detallados.

**Significado de SDA y SCL e I2C**

En I2C, SDA significa "Data Line" (línea de datos) y SCL significa "Clock Line" (línea de reloj). SDA es la línea por la que se transmiten los datos, mientras que SCL proporciona la señal de sincronización para la transferencia de datos.

**Aplicación de I2C en la Raspberri py 3**

Conectar dispositivos I2C.

La Raspberry Pi tiene dos pines dedicados para I2C:

* GPIO 5 (SDA): Pin de datos
* GPIO 3 (SCL): Pin de reloj

Conecta los dispositivos I2C a estos pines, asegurándote de que todos los dispositivos compartan una conexión común a tierra (GND).

Por otro lado, el protocolo I2C requiere resistencias pull-up en las líneas SDA y SCL para garantizar un funcionamiento correcto.

**Instalar herramientas de I2C**

Para interactuar con dispositivos I2C, también es muy conveniente instalar las herramientas i2c-tools:

**sudo apt update**

**sudo apt install i2c-tools**

Estas herramientas nos permiten escanear dispositivos conectados al bus I2C y verificar su funcionamiento.

Por ejemplo, antes de comunicarnos con un dispositivo, es útil escanear el bus I2C para detectar las direcciones de los dispositivos conectados. Usa el siguiente comando en la terminal:

**i2cdetect -y 1**

Este comando mostrará una tabla con las direcciones de los dispositivos detectados. Algo así,

**0 1 2 3 4 5 6 7 8 9 a b c d e f**

**00: -- -- -- -- -- -- -- -- -- -- -- -- --**

**10: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --**

**20: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --**

**30: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --**

**40: -- -- -- -- -- -- -- -- 48 -- -- -- -- -- -- --**

**50: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --**

**60: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --**

**70: -- -- -- -- -- -- -- --**

En este ejemplo, se ha detectado un dispositivo con la dirección 0x48.

**ECO-Procesaminto Raspberry Pi 3**

El eco-procesamiento con Raspberry Pi 3 implica utilizar esta computadora de bajo costo y consumo energético para aplicaciones que promuevan la sostenibilidad y el uso eficiente de recursos, como el monitoreo ambiental, la automatización de viviendas con tecnologías eficientes, y la educación en ciencia y tecnología verde.

Ventajas de la Raspberry Pi 3 para el eco-procesamiento:

* **Bajo costo:**

Facilita la implementación de proyectos de monitoreo ambiental y automatización en áreas donde los recursos son limitados.

* **Versatilidad:**

Se adapta a una amplia variedad de proyectos, desde sensores ambientales hasta sistemas de automatización.

* **Consumo de energía reducido:**

Permite desarrollar aplicaciones con una huella de carbono menor, contribuyendo a la sostenibilidad.

* **Comunidad y soporte:**

Cuenta con una gran comunidad de desarrolladores que proporcionan recursos y soporte para proyectos de eco-procesamiento.

* **Facilidad de uso:**

El software y las herramientas para Raspberry Pi son relativamente fáciles de aprender y utilizar.

**Definición del Maestro y Esclavo por Eco-Procesamiento (Arduino y ESP32)**

En el contexto de eco-procesamiento con Arduino y ESP32, la relación maestro-esclavo se refiere a la comunicación entre dispositivos donde uno controla la transmisión de datos y el otro responde a las solicitudes. En este caso:

1. Maestro (Arduino o ESP32):
   * Es el dispositivo que inicia la comunicación y envía comandos.
   * Puede solicitar datos o enviar instrucciones al esclavo.
   * En protocolos como I2C, el maestro controla el reloj y la dirección de los dispositivos esclavos.
2. Esclavo (Arduino o ESP32):

* Responde a las solicitudes del maestro.
* Puede enviar datos cuando el maestro lo requiere.
* En I2C, el esclavo solo transmite cuando el maestro lo solicita.

Aplicación en Eco-Procesamiento:

El eco-procesamiento busca optimizar el consumo energético y la eficiencia en sistemas embebidos. En este esquema:

* El maestro puede gestionar el procesamiento de datos y enviar solo información relevante al esclavo.
* El esclavo puede entrar en modo de bajo consumo (Deep Sleep en ESP32) cuando no se necesita procesamiento activo.
* Se pueden usar protocolos eficientes como MQTT o I2C para minimizar el uso de energía en la transmisión de datos.