PRÁCTICA 8: COMUNICACIÓN ENTRE PUERTOS SERIALES

Esta octava y última práctica trata de ver como se puede generar una comunicación bidireccional de bits entre dos puertos seriales. Para hacer está práctica, únicamente hemos necesitado utilizar dos pines de la placa ESP32-S3. Se tratan de los pines GPIO43 y GPIO44, que también se llaman **TXD** (de transmisión) y **RXD** (de recepción), respectivamente. El objetivo principal es que cuando hay datos disponibles en un puerto serie, se envían al puerto serie principal y viceversa. Este mecanismo se suele utilizar en aplicaciones donde necesitas retransmitir datos entre diferentes interfaces seriales, como un periférico y un monitor serial o incluso entre dos microcontroladores.

Código 1

```
#include <Arduino.h>
# define RXD1 17
# define TXD1 18

void setup() {
Serial.begin(115200, SERIAL_8O1);
Serial1.begin(115200, SERIAL_8O1, RXD1, TXD1);
}

void loop() {
  if(Serial1.available()>0){
  Serial.write(Serial1.read());
  }
  if(Serial.available()>0){
  Serial1.write(Serial.read());
  }
}
```

Inclusión de librerías y definición de pines

```
#include <Arduino.h>
#define RXD1 17
#define TXD1 18
```

- ***#include <Arduino.h>***: Incluye las definiciones y funciones básicas de Arduino.
- ***#define RXD1 17***: Define el pin 17 como el pin RX (recepción) para Serial1.
- ***#define TXD1 18***: Define el pin 18 como el pin TX (transmisión) para Serial1.

Función *setup()*

```
void setup() {
  Serial.begin(115200, SERIAL_8O1);
  Serial1.begin(115200, SERIAL_8O1, RXD1, TXD1);
}
```

- ***Serial.begin(115200, SERIAL_8O1)***: Inicializa el puerto serial principal (Serial) a una velocidad de 115200 baudios, con un formato de 8 bits de datos, sin paridad, y 1 bit de parada (SERIAL_8O1).

* ***Serial1.begin(115200, SERIAL_8O1, RXD1, TXD1)***: Inicializa Serial1 a una velocidad de 115200 baudios, con el mismo formato de datos, y utilizando los pines definidos para RX (**RXD1**) y TX (**TXD1**).

Función *loop()*

```
void loop() {
    // De Serial1 a Serial
  if (Serial1.available() > 0) {
    Serial.write(Serial1.read());
  }
    //De Serial a Serial1
  if (Serial.available() > 0) {
    Serial1.write(Serial.read());
  }
}
```

* **Comunicación de *Serial1* a *Serial***
    - ***if (Serial1.available() > 0)***: Comprueba si hay datos disponibles para leer en Serial1.
    - ***Serial.write(Serial1.read())***: Lee un byte de datos de Serial1 y lo escribe en el puerto serial principal (Serial). Esto envía los datos recibidos en Serial1 al monitor serial conectado al puerto principal.
* **Comunicación de *Serial* a *Serial1***
    - ***if (Serial.available() > 0)***: Comprueba si hay datos disponibles para leer en el puerto serial principal (Serial).
    - ***Serial1.write(Serial.read())***: Lee un byte de datos del puerto serial principal (Serial) y lo escribe en Serial1. Esto envía los datos recibidos en el puerto serial principal a Serial1.

Gracias al funcionamiento explicado anteriormente, este programa nos permite escribir en tiempo real en el mismo terminal una vez se ha compilador el código con los carácteres **ASCII**.

Imagen del Terminal