**USART1 TX/RX Example with LED Blink on TX Only (STM32F103C8)**

**Overview**

This project demonstrates USART1 communication on the STM32F103C8 microcontroller. It continuously transmits the character 'B' at a baud rate of 9600 and receives incoming data via USART1. Additionally, an onboard LED blinks whenever a character is transmitted, but not on reception.

**Features**

* Transmits the character 'B' continuously at **9600 baud**.
* Receives data via **USART1\_RX** and stores it in a buffer.
* **LED Blinks** on **transmission only**.
* Uses **interrupt-based RX handling**.
* System clock configured to **72 MHz** using **8 MHz HSE + x9 PLL**.

**Hardware Requirements**

* **STM32F103C8** (Blue Pill or similar)
* **USB to TTL Converter** (for serial communication, e.g., FTDI, CP2102)
* **LED (On-board or External)**
* **Connecting Wires**

**Pin Configuration**

| **Pin** | **Function** | **Description** |
| --- | --- | --- |
| PA9 | USART1\_TX | Transmit data (TX) |
| PA10 | USART1\_RX | Receive data (RX) |
| PC13 | On-board LED | Blinks on TX only |

**Software Setup**

**Development Environment**

* **Compiler:** Keil uVision, STM32CubeIDE, or ARM GCC Toolchain
* **Debugger:** ST-Link, J-Link, or USB-TTL converter
* **Peripheral Library:** STM32F10x Standard Peripheral Library

**Code Breakdown**

**1. Clock and GPIO Configuration**

* Enables the necessary peripheral clocks.
* Configures **PA9 (TX)** as **Alternate Function Push-Pull**.
* Configures **PA10 (RX)** as **Floating Input**.
* Configures **PC13 (LED)** as **Output Push-Pull**.

**2. USART1 Initialization**

* Sets **9600 baud rate** at **72 MHz system clock**.
* Enables **TX and RX** functionalities.
* Enables **USART1 RX Interrupt** for incoming data.
* Enables **USART1 in NVIC** for interrupt handling.

**3. Main Loop**

* Waits for **TXE (Transmit Empty) flag** to send data.
* Sends **'B'** character via USART1.
* Toggles the **PC13 LED** when transmitting.
* Includes a **500ms delay** for LED visibility.

**4. USART1 IRQ Handler**

* Checks **RXNE (Receive Not Empty) flag**.
* Stores received data into **rxBuffer**.
* Resets buffer index if it overflows.

**5. Delay Function**

* Implements a **blocking delay** using **NOP (No Operation)** instructions.

**How to Run the Code**

1. **Flash the Firmware**:
   * Use **Keil, STM32CubeIDE, or OpenOCD** to compile and upload the code.
   * Ensure the correct **USART1 TX/RX pins** are connected to a serial terminal.
2. **Monitor Serial Output**:
   * Open a **serial terminal** (Putty, RealTerm, Tera Term, etc.).
   * Set **Baud Rate: 9600, Data: 8N1 (No Parity, 1 Stop Bit)**.
   * Observe **'B'** being printed continuously.
3. **Test Reception**:
   * Send characters from the terminal to the STM32.
   * Verify received data is stored in the buffer.

**Notes**

* Ensure proper **clock settings** in system configuration.
* If using a **USB-TTL converter**, check the TX/RX connections.
* For debugging, use **ST-Link** or **USART debug prints**.

**License**

This project is open-source and free to use for educational and personal projects.

**Author**

Developed for STM32 embedded systems experimentation.