Below is a **README** for your UART project on the **STM32F401RE** running at **16 MHz** with a **9600 baud** rate. You can copy and paste it into a file (e.g. README.md) or a Word file (README.docx) as desired.

**STM32F401RE UART (USART2) at 9600 Baud, 16 MHz Clock**

This project demonstrates how to configure **USART2** on an STM32F401RE (such as a Nucleo-F401RE board configured to run at **16 MHz**) to receive characters at **9600 baud**. Each received character from the host PC causes the on-board LED (connected to **PA5**) to blink a corresponding number of times (ASCII value mod 16).

**Table of Contents**

1. [Overview](https://chatgpt.com/c/67ac9336-5850-8000-b005-1be99daeb42b#overview)
2. [Hardware Setup](https://chatgpt.com/c/67ac9336-5850-8000-b005-1be99daeb42b#hardware-setup)
3. [Software Explanation](https://chatgpt.com/c/67ac9336-5850-8000-b005-1be99daeb42b#software-explanation)
4. [Project Structure](https://chatgpt.com/c/67ac9336-5850-8000-b005-1be99daeb42b#project-structure)
5. [Building and Uploading](https://chatgpt.com/c/67ac9336-5850-8000-b005-1be99daeb42b#building-and-uploading)
6. [Usage](https://chatgpt.com/c/67ac9336-5850-8000-b005-1be99daeb42b#usage)
7. [Troubleshooting](https://chatgpt.com/c/67ac9336-5850-8000-b005-1be99daeb42b#troubleshooting)
8. [License](https://chatgpt.com/c/67ac9336-5850-8000-b005-1be99daeb42b#license)
9. [References](https://chatgpt.com/c/67ac9336-5850-8000-b005-1be99daeb42b#references)

**1. Overview**

* **MCU**: STM32F401RE (configured at **16 MHz** system clock)
* **UART**: USART2, receiving data at **9600 baud**, 8 data bits, 1 stop bit, no parity
* **Pins**:
  + **PA3** → RX (USART2\_RX, AF7)
  + **PA5** → On-board LED (LD2)
* **Terminal**: Connect via a Virtual COM port (ST-Link) at **9600,n,8,1**
* **Behavior**:
  + Wait for a character from the terminal.
  + Blink the LED **(ASCII\_value mod 16)** times.
  + Repeat.

**Important**: Many Nucleo-F401RE boards are by default set to 84 MHz using the PLL. You must ensure the system clock is actually running at **16 MHz** (via HSI or a custom PLL setting). If not, your UART baud rate will be incorrect.

**2. Hardware Setup**

1. **Nucleo-F401RE Board**
   * Ensure your board is powered via USB and recognized by your PC.
   * By default, the on-board **ST-Link** interface provides a Virtual COM port.
2. **UART/Serial Connection**
   * The **ST-Link** on-board chip connects **USART2** (pins PA2/PA3) to your PC over USB.
   * This means you simply connect the Nucleo to your PC’s USB; no extra USB/Serial dongle is needed.
3. **Terminal Settings**
   * In Tera Term, PuTTY, or another serial monitor, open the COM port labeled **“STMicroelectronics STLink Virtual COM Port”** (or similar).
   * Set **9600 baud**, 8 data bits, no parity, 1 stop bit.
4. **On-Board LED**
   * The LED labeled **LD2** (green) on the Nucleo board is connected to **PA5** in this code.

**3. Software Explanation**

* **USART2\_init()**:
  + Enables clocks for **GPIOA** and **USART2**.
  + Configures **PA3** as alternate function (AF7) for **USART2\_RX**.
  + Sets the baud rate register (**BRR**) to **0x0683** for 9600 baud at a 16 MHz clock.
  + Enables USART2 in 8N1 mode, no flow control.
* **USART2\_read()**:
  + Blocks until **RXNE** (Receive Data Register Not Empty) is set.
  + Returns the byte in **USART2->DR**.
* **LED\_blink()**:
  + Takes the character’s integer value, performs mod 16, and blinks the LED that many times.
  + Each blink has a 200 ms on/off, plus an additional 800 ms delay after the sequence.
* **delayMs()**:
  + A simple loop-based delay, calibrated for a **16 MHz** system clock.
  + Loops 2000 times for roughly 1 ms.

**4. Project Structure**

Below is an example layout (your IDE may vary):

.

├── Inc/

│ └── stm32f4xx.h // or stm32f401xe.h (CMSIS/Device headers)

├── Src/

│ └── main.c // contains the USART code & LED blinking

└── README.md

If using **Keil uVision**, **STM32CubeIDE**, or another toolchain, place main.c in the appropriate Src folder and ensure the MCU startup/clock configuration is set to 16 MHz.

**5. Building and Uploading**

1. **Open** your project in the IDE (e.g. Keil uVision or STM32CubeIDE).
2. **Check** that your clock configuration in the startup code is truly set to **16 MHz** (HSI or custom PLL with no multiplication).
3. **Compile** the project.
4. **Connect** the Nucleo board via USB and **flash** the code (debug or run).

When the board resets, USART2 is ready at 9600 baud, and LED blinking depends on received characters.

**6. Usage**

1. **Open a Serial Terminal** (TeraTerm, PuTTY, screen, etc.).
2. Choose the COM port labeled **STMicroelectronics STLink Virtual COM Port**.
3. Set the connection to **9600 baud**, 8 data bits, no parity, 1 stop bit.
4. **Type characters** on your keyboard. The on-board LED will blink (ASCII\_value mod 16) times each time you press a key.
   * You will **not** see what you typed echoed back to the terminal (not part of the code).

**Note**: You must let each blink sequence finish before sending another character, otherwise the subsequent character might be processed later (the code is blocking).

**7. Troubleshooting**

1. **No LED blinking / no response**:
   * Make sure the board is actually running at **16 MHz**. If it’s at 84 MHz, the baud rate register is incorrect for 9600.
   * Check that you selected the correct COM port in your terminal program.
   * Confirm that you’re actually connected to **PA3** for RX (the default ST-Link routing on the Nucleo connects PA2/PA3 to USB).
2. **Garbage data**:
   * A mismatch between the configured baud rate and the clock can cause bad characters.
   * Double-check your terminal is set to **9600**.
   * Verify BRR = 0x683 is correct for 16 MHz.
3. **Blinking too fast or slow**:
   * delayMs() is a rough software delay. You can adjust the loop counts or use a hardware timer for more accurate timing.
4. **Need Echo**:
   * If you want to echo received characters back to the terminal, implement a simple USART2\_write() function and call it after reading a character.

**8. License**

This project is offered under the [MIT License](https://opensource.org/licenses/MIT). You are free to use and modify it as needed.

**9. References**

* [STM32F401RE Reference Manual (RM0368)](https://www.st.com/resource/en/reference_manual/dm00096844.pdf)
* [Nucleo-F401RE Board](https://www.st.com/en/evaluation-tools/nucleo-f401re.html)
* [USART Calculation for Baud Rate @ 16 MHz](https://www.st.com/resource/en/programming_manual/dm00245755.pdf) (See details on BRR register)
* [Tera Term / PuTTY](https://www.teraterm.org/) for UART console

**Enjoy your 9600 baud UART project on the STM32F401RE at 16 MHz!**