



# AN1797 APPLICATION NOTE

---

## STR71x UART COMMUNICATION WITH A PC USING RS232

---

### INTRODUCTION

This document presents a standard communication interface between a STR71x microcontroller and a PC. This communication is done through the STR71x UART peripheral and a serial port of the PC using the RS 232 protocol.

## 1 SERIAL COMMUNICATION

The main features of the serial communication are summarized below.

### 1.1 MAIN FEATURES

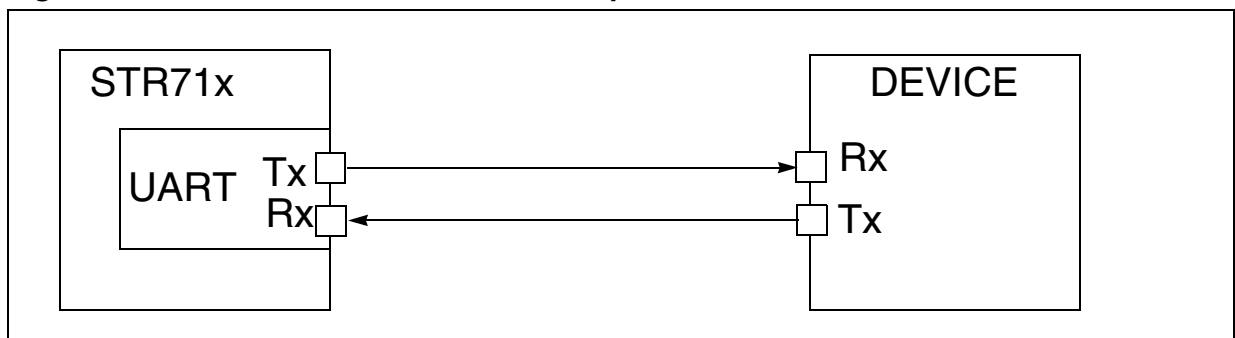
The serial interface described here, allows two digital systems to communicate using only a few wires. It is asynchronous, so it doesn't need a clock.

The serial interface requires at least two wires:

- a transmitter wire (Tx)
- a receiver wire (Rx)

The interface is full-duplex, i.e. it can transmit and receive at the same time.

**Figure 1. STR71x and UART interface Setup**



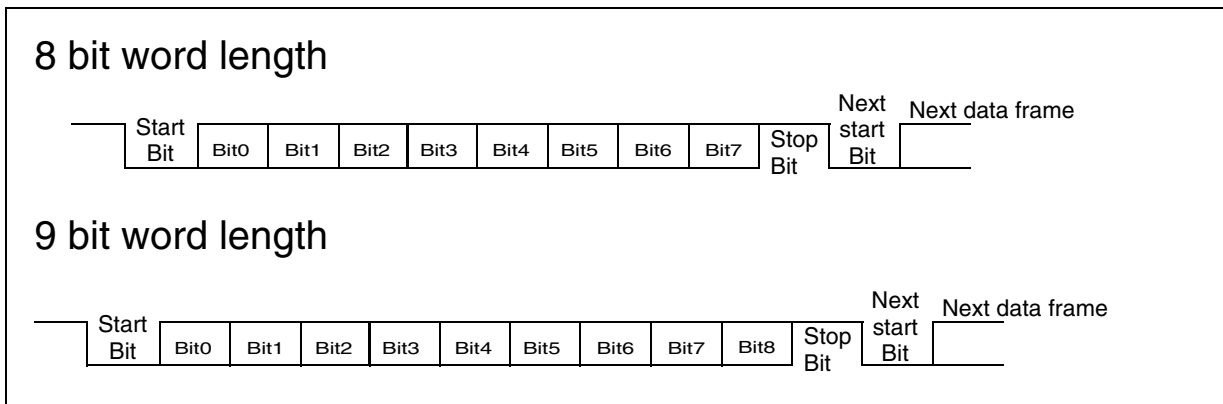
### 1.2 BAUD RATES

Transmission and reception are driven by a baud rate generator. However, be aware that to communicate correctly, the receiver must have exactly the same baud rate as the transmitter baud rate, otherwise the communication will be corrupted. If this condition is met, a wide range of baudrates is possible.

### 1.3 FRAMES

Communication frames begin with a start bit and end with a stop bit. Stop bits can be 0.5, 1, 1.5 or 2 bits long. Frame words can be 8 or 9 bits long. The least significant bit is sent first (see [Figure 2](#)). In 8-bit word mode, the eighth bit can be a data bit or a parity bit, but in 9-bit word mode, the ninth bit is a parity bit. Parity can be odd or even. Receiver and transmitter devices must have the same word length, the same parity and the same number of stop bits.

Figure 2. Frames and word length



## 2 RS232 COMMUNICATION WITH A PC

### 2.1 MAIN FEATURES

The electrical and protocol characteristics of the STR71x UART peripheral are different from RS232. In RS232 communication, high level is typically +12 V and low level is typically -12 V, while the UART peripheral works with TTL level (0, +5V).

The adaptation is done using a line adapter circuit (ex: ST3232), which transforms the logic levels output by the digital system into RS232-compatible levels and vice versa.

### 2.2 PC CONFIGURATION

In this example, the PC will act as a terminal. The description below refers to the Windows environment. The Windows terminal application is called HyperTerminal.

Under Windows, open the “HyperTerminal” application. To configure it, go to the communication parameters menu. The options that you set in this menu must be the same as the ones defined in your STR71x.

After selecting the right serial communication port, select the same baud rate as the one configured in the microcontroller. Data word length will be 8 bits, but you can choose to use 1 or 2 stop bits. “Flow control” can be either Xon/Xoff or none.

The PC is then correctly configured.

### 2.3 GENERAL INITIALIZATION

The external clock works at 16 MHz.

In the STR71x, four UARTs are available. In this application only UART0 is used. Two pins of the STR71x are used:

- the TX0 pin (Transmit Data Output, alternate function of P0.9).
- the RX0 pin (Receive Data Input, Input Tristate CMOS of P0.8).

The value of the Port 0 Data Register does not matter.

## 2.4 UART CONFIGURATION

The peripheral is configured using the UART Control Register (CR), and the Baud Rate Register (BRR).

As in this application a polling strategy is used, UART interrupts are disabled. Also FIFOs and loop back mode are disabled.

In the application, data length can be 8 bits or 9 bits (parity bit included). Parity must be odd or even parity if you select 7 bit mode or 8 bit mode with parity is selected. Select No Parity if 8 bit mode is selected, stop bits can be 1, 1.5 or 2 stop bits.

### 2.4.1 Baud rate selection

Then you must select the baud rates. We should always have the same baud rates in the PC and in the UART.

The baud rates are calculated as follows:

$$\text{Baudrate} = f_{\text{PCLK1}} / (16 * \text{BRR}), \text{BRR} = f_{\text{PCLK1}} / (16 * \text{Baudrate})$$

For example, in our application, we will use 9600 as baud rate, and the  $f_{\text{PCLK1}}$  is 8 MHz.

### 3 SERIAL COMMUNICATION BETWEEN STR71X AND PC VIA RS232

The software used for this application is the UART driver. It is up to you to adapt it to your specific application.

#### 3.1 GENERAL DESCRIPTION

In this application, an STR71x is connected to a PC. The communication is performed using the Windows “HyperTerminal” application.

Characters are sent by the PC in ASCII code to the STR71x. Each time 4 characters are received by the STR71x, they are sent back to the PC and can be read on its screen.

The software is a polling serial communication between an STR71x and a PC via RS232 using the UART0. The communication works at 9600bps, 8-bit word length without parity and has 1 stop bit.

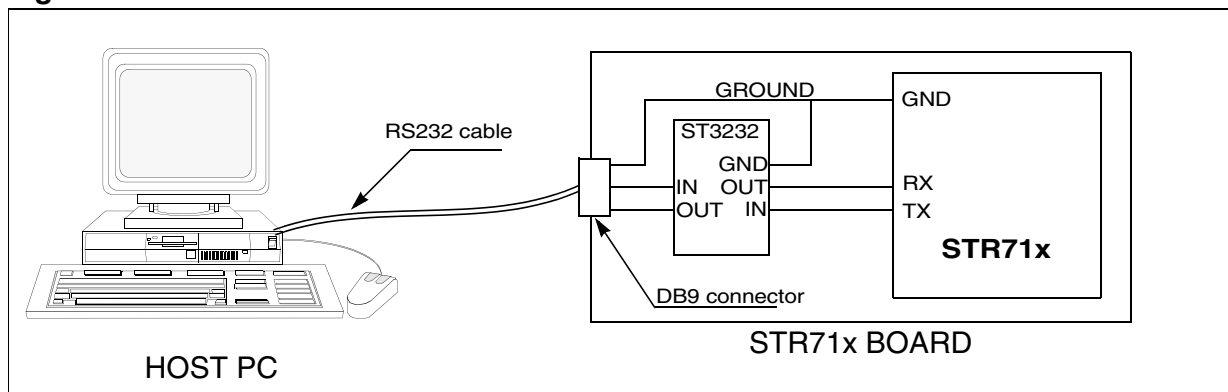
The first part of the program performs the initialization of the STR71x core and UART0 peripheral. The program sends the content of a 4-byte buffer to the PC. The initialized value of the buffer is “test” in ASCII code. When the 4 characters are sent, the program enters reception mode. In reception mode, the software waits for the user to send 4 ASCII characters from the PC. Each character is put in the buffer until the fourth is received. Then it goes back to transmission mode which will send the new contents of the buffer and so on.

As the STR71x sends the same characters as received, it is very easy to check if the communication is correct. Note that this program does not manage communication errors.

#### 3.2 HARDWARE

The STR71x UART peripheral cannot be directly connected to a PC, as it uses the RS232 protocol. The conversion between UART and RS232 can be done using a ST3232. An overview schematic is presented below ([Figure 3](#)).

Figure 3. Hardware Overview



Be sure that the three main devices (PC, STR71x, ST3232) have the same electrical reference (GND).

The Receive Data pin (RD) of the serial port of the PC must correspond to the TX pin of the STR71x, and the Transmit Data pin (TD) to the RX pin.

### 3.3 SOFTWARE IMPLEMENTATION

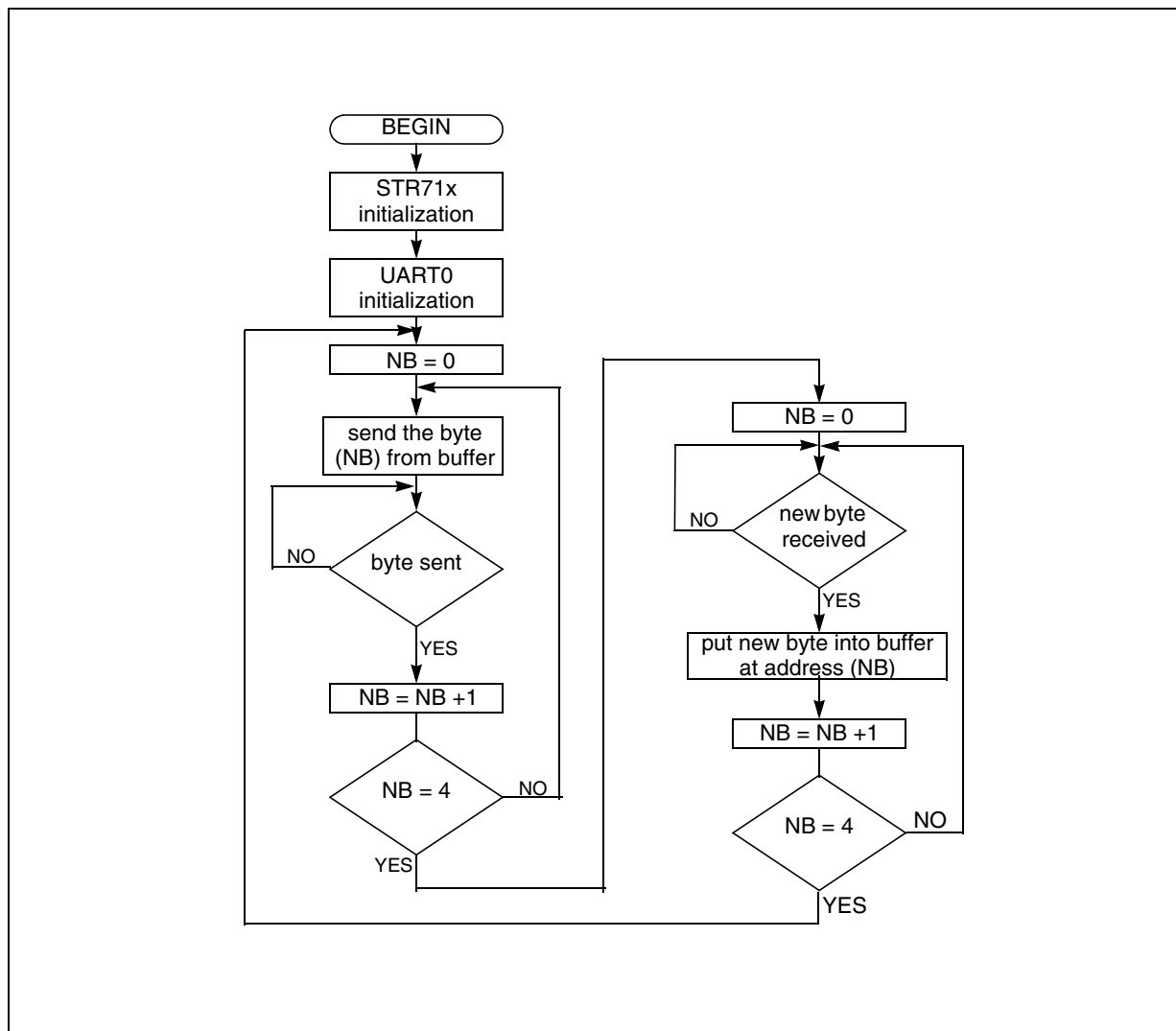
#### 3.3.1 Code description

The C code described below is for guidance only. The file cannot be used alone, the complete software can be found at <http://www.st.com/mcu>

#### 3.3.2 Flowchart

The flowchart of the application is presented below.

**Figure 4. Transmission mode, Reception mode**





“THE PRESENT NOTE WHICH IS FOR GUIDANCE ONLY AIMS AT PROVIDING CUSTOMERS WITH INFORMATION REGARDING THEIR PRODUCTS IN ORDER FOR THEM TO SAVE TIME. AS A RESULT, STMICROELECTRONICS SHALL NOT BE HELD LIABLE FOR ANY DIRECT, INDIRECT OR CONSEQUENTIAL DAMAGES WITH RESPECT TO ANY CLAIMS ARISING FROM THE CONTENT OF SUCH A NOTE AND/OR THE USE MADE BY CUSTOMERS OF THE INFORMATION CONTAINED HEREIN IN CONNECTION WITH THEIR PRODUCTS.”

Information furnished is believed to be accurate and reliable. However, STMicroelectronics assumes no responsibility for the consequences of use of such information nor for any infringement of patents or other rights of third parties which may result from its use. No license is granted by implication or otherwise under any patent or patent rights of STMicroelectronics. Specifications mentioned in this publication are subject to change without notice. This publication supersedes and replaces all information previously supplied. STMicroelectronics products are not authorized for use as critical components in life support devices or systems without express written approval of STMicroelectronics.

The ST logo is a registered trademark of STMicroelectronics.

All other names are the property of their respective owners

© 2005 STMicroelectronics - All rights reserved

STMicroelectronics group of companies

Australia – Belgium - Brazil - Canada - China – Czech Republic - Finland - France - Germany - Hong Kong - India - Israel - Italy - Japan - Malaysia - Malta - Morocco - Singapore - Spain - Sweden - Switzerland - United Kingdom - United States of America

[www.st.com](http://www.st.com)