



## ABSTRACT

RS-232 has been existing for several decades. Although this single-ended interface is common, many questions and problems do arise during the design and applications. This article collects some most frequently asked questions to help the user understand this popular interface standard. The basics about how to set up the interface are given in this article. The application tips help provide a successful implementation in the system.

---

## Table of Contents

<b>1 Introduction</b> .....	2
1.1 What is the valid signal level of RS-232?.....	2
1.2 What is the charge pump for in RS-232 transceivers? How does it work? .....	2
1.3 How to select right capacitor for the charge pumps?.....	2
1.4 How to estimate the power dissipation of RS-232?.....	3
1.5 How to set up the handshaking of the RS-232 communication?.....	3
1.6 How to configure the unused pins of RS-232 transceivers?.....	4
1.7 What is the slew rate and how is it related to the data rate?.....	4
<b>2 Summary</b> .....	4

## Trademarks

All trademarks are the property of their respective owners.

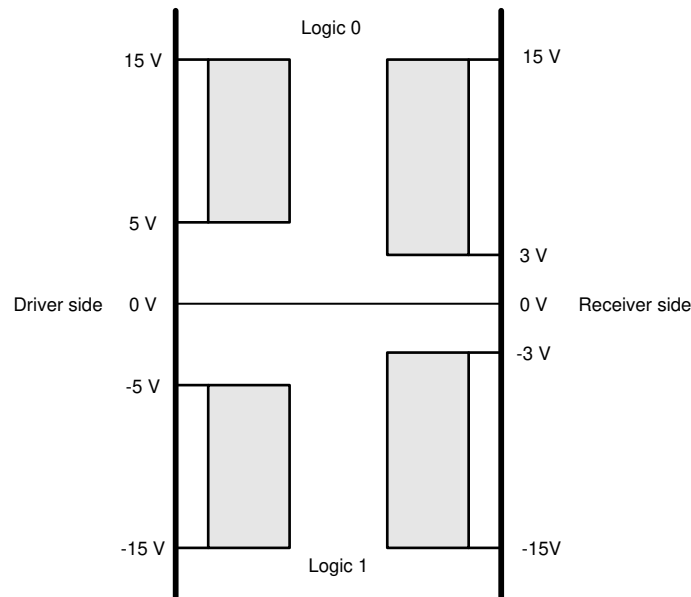
## 1 Introduction

### 1.1 What is the valid signal level of RS-232?

Before deploying RS-232, you need to know what the valid signal is in the interface. Knowing the legit signal level usually is that first step to implement or debug the communication.

Valid RS-232 signals are either in the range of +3 V to +15 V or the range -3 V to -15 V with respect to the common ground (Figure 1-1). To be more specific, the driver output is Logic 0 when the voltage is between +5 V and +15 V and is Logic 1 when the voltage is between -5 V to -15 V.

Received signal voltage level is that Logic 0 works on the received signal voltages up to +3 V to +15 V and Logic 1 works with voltages up to -3 V to -15 V.



**Figure 1-1. RS-232 signal level**

### 1.2 What is the charge pump for in RS-232 transceivers? How does it work?

In modern RS-232 transceivers, only one supply is needed to generate the valid RS-232 signaling. To generate the voltage level higher than the supply, it is common that charge pumps are employed in RS-232 ICs. The detail of the charge pump operation can be found in this [article](#).

The basic idea is to use a clock signal to control switches alternatively. In the first clock cycle, the voltage up to the supply is stored on a capacitor. This charge is transferred to the second capacitor with one end being connected to the supply in the next cycle. By keeping transferring the energy from the supply to the capacitors, 2x voltage of the supply is generated on the capacitors.

### 1.3 How to select right capacitor for the charge pumps?

The charge pump needs to work with external flying capacitor and storage capacitor to generate the required voltage from the supply. Please refer to the [blog](#) mentioned previously for the impact of different capacitor values. Since the charge pump can double the supply voltage theoretically, it is recommended that the external capacitors have at least 10V working voltage. Higher voltage rating could provide some margin for the voltage variation in the applications.

### 1.4 How to estimate the power dissipation of RS-232?

To have enough budgets in the power design of the system, it is normally required to estimate the power dissipation of the transceiver during the normal operation. However it is rare to find the active power consumption on the RS-232 transceiver's data sheet. The reason behind and estimate can be found in this [article](#).

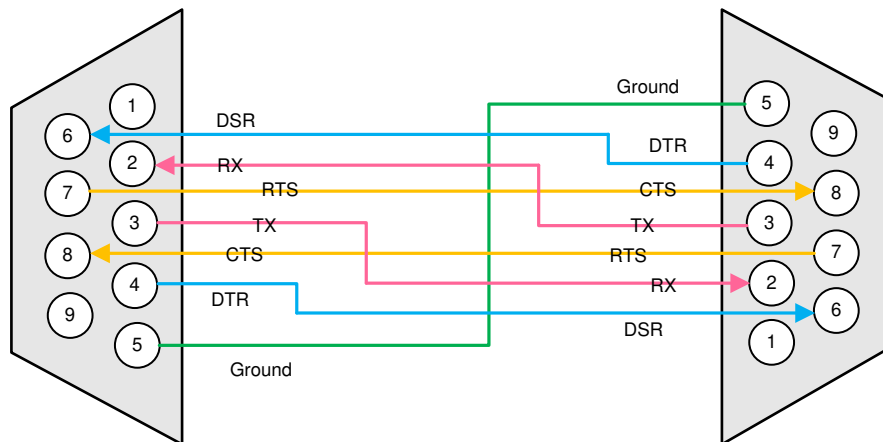
To put it simply, since the power consumption is not linear, it is hard to quantize the value in various applications. The trick for a quick estimate is to convert the capacitive load into the equivalent resistance. Before the driver is running out of juice, the power vs. load relation is pretty linear excluding the static power.

### 1.5 How to set up the handshaking of the RS-232 communication?

Handshaking is a way of flow control of the communication in many RS-232 applications, which can be implemented in software or hardware. The main purpose of handshaking is to guarantee successful communication by preventing receiver from overloading. Receivers will let the transmitter device to hold data transmission if they are overloaded.

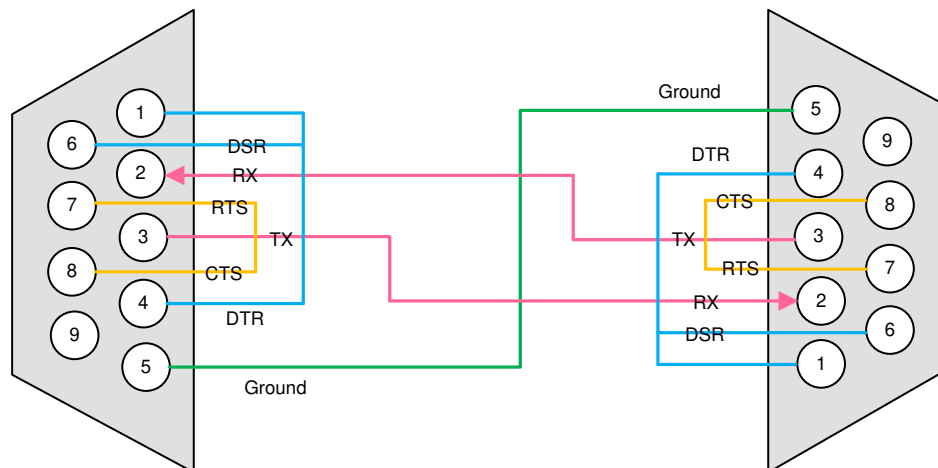
There are three types of handshaking: Software handshaking, Hardware handshaking and Both. Using RS-232 software handshaking can save the additional lines. One of the applications is sending data over telephone lines. One RS-232 software protocol is called Xon/Xoff. The Xon/Xoff operates by sending the control characters along the data line. Command Xon starts the transmission and Xoff stops it. For example, when an instrument cannot accept more data from a computer, it could send a single Xoff character to the computer to let it stop sending data. When the instrument is ready, it sends a Xon character to restart transmission. Xon is ASCII character 17 and Xoff is ASCII character 19.

Hardware flow control (also called as RTS / CTS flow control) is superior compared to software flow control with the cost of extra lines. In the diagram of 9 pin null modem communication ([Figure 1-2](#)), the RTS (Request To Send) of the DTE (Data Terminal Equipment) is connected to the CTS (Clear To Send) of the DCE (Data Circuit-terminating Equipment), same for the DTR (Data Terminal Ready) and the DSR (Data Set Ready). The DTE initiates the transmission by setting the RTS in the ON state. When the DCE is ready, it puts the CTS on. Then the DTE responds by placing the DTR line into the ON state, which stays on when the data is being transmitted.



**Figure 1-2. 9 Pin RS-232 handshaking connection**

In some applications (for example, diagnostic purpose), the loopback function is requested to bypass the hardware handshaking. The connection below ([Figure 1-3](#)) is an example of how to implement it. By connecting CTS to RTS and DTR to DSR on the same side, the correct response is ensured.



**Figure 1-3. 9 pin RS-232 loopback connection**

### 1.6 How to configure the unused pins of RS-232 transceivers?

In many applications, all of the channels of the transceivers are not used. Let us use TRS3232 as an example, which comes with two drivers and two receivers. If you only need to have one active channel in the system, you need to know what to do with the unused channel. In TR3232, it is recommended to bias the transmitter input (DIN) pins with pull up or pull down resistors. The receiver input (RIN) pins are internally integrated with 5-k $\Omega$  resistors to ground per RS-232 standard. Therefore RIN, ROUT, DOUT can all left open.

### 1.7 What is the slew rate and how is it related to the data rate?

In RS-232 standard, the data rate is specified with the slew rate. Slew rate is defined as the change of an edge represented by  $\Delta V/\Delta t$ , while rise time only counts the time of a transition (10% to 90%, or 20% to 80%). RS-232 standard (TIA/EIA-232-F) limits the maximum slew to help reduce the likelihood of cross-talk between adjacent signals. The history and technical discussion can be found in this [application note](#).

For long cable or heavy load, the slew rate is limited by the device output current charging the capacitive load. With short cable or light load, the on-chip slew rate control still makes sure the edge not too fast. For an example, 200 kbps data rate signal has 5  $\mu$ s unit bit time. The standard limits the transition time to be 4% of the bit time, which is 0.2  $\mu$ s in this case. With the maximum slew rate 30 V/ $\mu$ s, the output voltage is  $30 \times 0.2 = 6$  V - the minimum required voltage on the bus. We can tell this is the fastest data rate defined by the standard. If a higher data rate is needed, the slew rate should be increased.

## 2 Summary

After knowing the signal voltage level, power dissipation estimate, and handshaking protocol, the user can set the interface up for a successful communication scheme. Having answers to the application questions like capacitor type, slew rate, and unused pins should help the user start on that path.

## IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATA SHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES "AS IS" AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, regulatory or other requirements.

These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you will fully indemnify TI and its representatives against, any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI's products are provided subject to [TI's Terms of Sale](#) or other applicable terms available either on [ti.com](http://ti.com) or provided in conjunction with such TI products. TI's provision of these resources does not expand or otherwise alter TI's applicable warranties or warranty disclaimers for TI products.

TI objects to and rejects any additional or different terms you may have proposed.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265  
Copyright © 2022, Texas Instruments Incorporated