Application Note

RS232 Design Guide

Duy Nguyen

ABSTRACT

This document discusses best practices for RS232 schematics setup and PCB layout.

Table of Contents

Trademarks

1 Introduction

RS232 design from schematic to PCB layout can be straightforward but small details related to pin biasing can present issues in some applications. Careful consideration for power up sequencing, backbias protection, and understanding if or when an undetermined digital state is present during operation can help to avoid system glitches and result in a more robust RS232 system.

2 Terminology

TTL = transistor to transistor logic, these are the pins that interface between the RS232 device and the processor such as DIN or TIN and ROUT sometimes called the logic pins. TTL defines a high logic as larger than 2V and less than 0.8V as a low logic.

RS232 side/bus = the pins on the device facing the cable that can swing from -13V to +13V.

Bias = refers to tying a voltage to a pin by using a resistor to GND or Vcc to verify a reference voltage.

Charge pump = refers to the circuit responsible for generating the RS232 logic voltages, On the RS232 device, this is the C1+, C1-, C2+, C2-, V+, and V- pins.

ISR = interrupt service routine, this is software related in which the processor or microcontroller responds to an interrupt based on the trigger.

Additional reading of RS232 terminology can be found in the RS232 glossary.

3 Best Practices

3.1 Remember to Bias TTL Pins

Generally, do not leave the input floating. Floating inputs can cause increased leakage current and can degrade the device over time. See this document for more information.

For RS232 specifically, floating inputs such as DIN can cause garbage data to be transmitted through the cable. Most system designers do not realize the TTL input is floating even when the DIN pin is tied directly to the Tx pin of the processor. Normally the MCU pin is a CMOS output which drives the TTL input but there are some scenarios where the Tx pin can become high impedance.

The first case can occur during thepowering up sequence. If the RS232 device powers up before the processor then the Tx pin is not actively driving the input. The second case can occur after the processor powers up, if the UART Tx port has not been initialized by the processor then the Tx pin connected to the TTL DIN pin can still float.

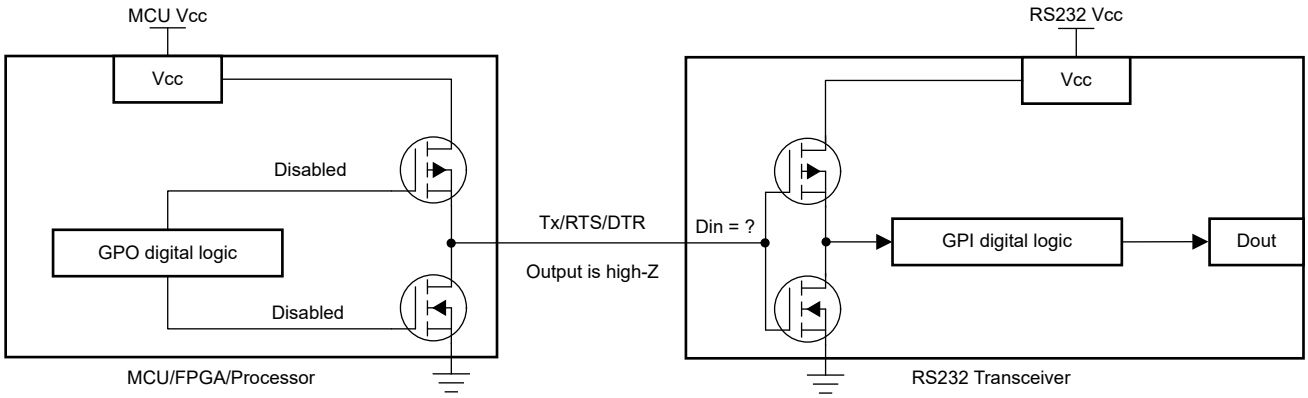

Figure 3-1. RS232 Transceiver in an Unknown Input State Condition

During the time between the processor powering up or before Tx gets initialized while the RS232 has already powered up, the TTL input can be susceptible to leakage currents. This can pull the input low or noise in the system can cause the input to fluctuate up and down. On the RS232 side, this causes unintended signals to be transmitted which can result in the receiving node to interpret incoming data as actual communication or trigger an ISR for framing errors, parity errors, or break conditions.

The same concept applies to the TTL Rout pins as well. These pins usually are connected to the TTL inputs from the processor like the Rx signal. If the RS232 device powers up later than the processor or the RS232 device disables its Rout drivers, like when it enters a low power mode, the TTL output from the RS232 device becomes high impedance and can float to GND or fluctuate with leakage or noise.

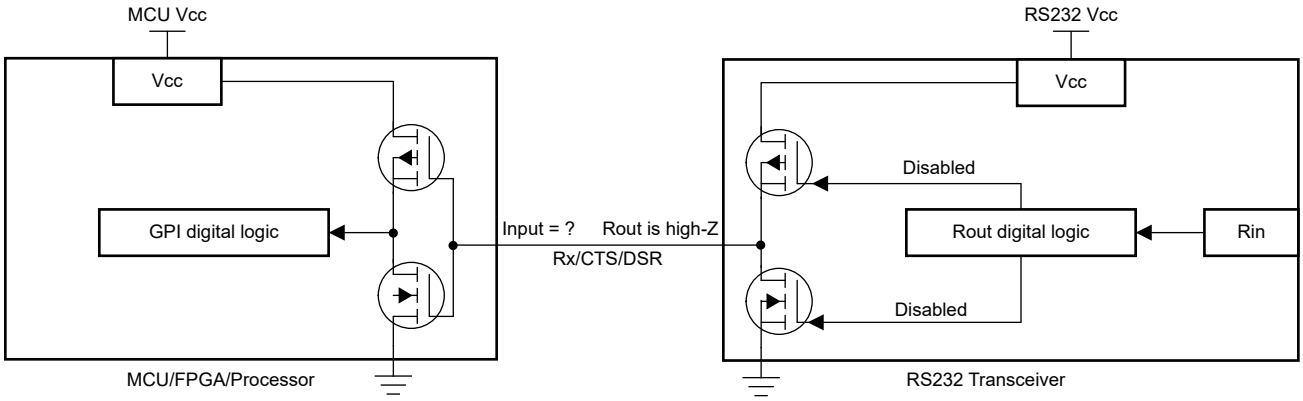

Figure 3-2. Processor Input State in an Unknown Condition

For this reason, an external pull-up or pull-down resistor on the DIN and ROUT pins are recommended. The value of this resistor can range from 4.5k ohms to 47k ohms. Using a stronger (lower value) resistor provides a better biasing voltage but results in larger leakage current from the supply rail. In most situations a 47k resistor functions normally. One key detail to keep in mind is if the resistor is a pull-up, it must be tied to the same Vcc as the device. This must prevent back biasing through the Routput if a leakage path exists.

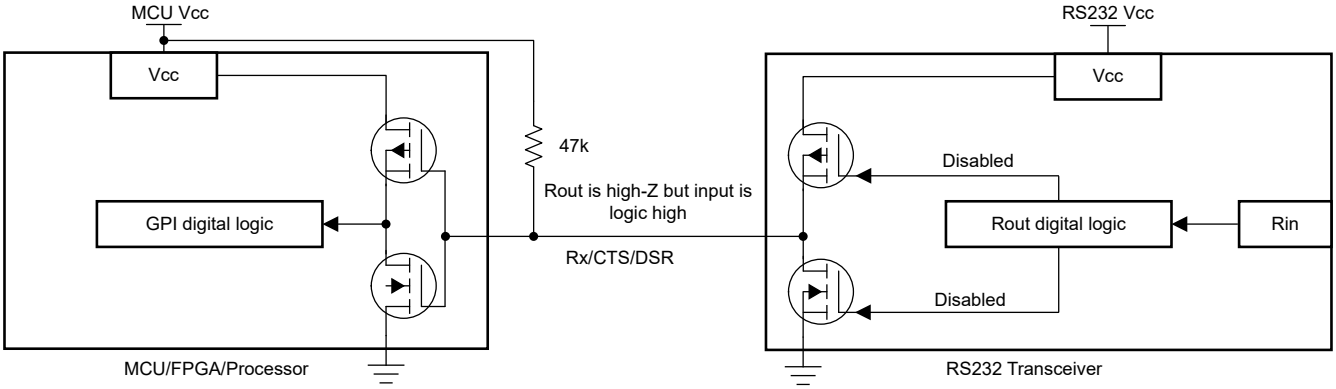

Figure 3-3. Verifying Known Input State for RS232 Transceiver

If the environment has higher electrical noise, then a stronger (lower value) resistor must be used. These resistors are recommended to be placed on the schematic even if the system designer thinks the resistors are not used as this is something that is nice to have and not need than to not have it but realize the user requires it later.

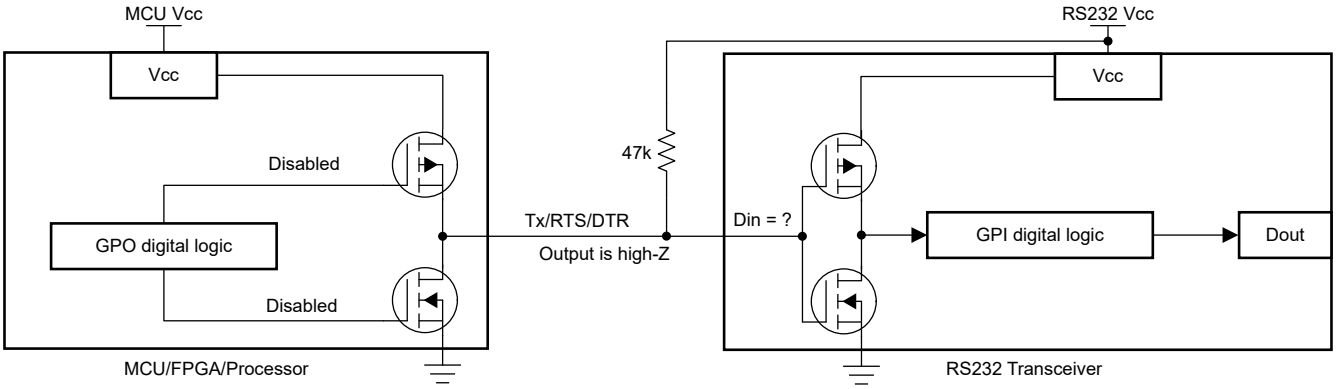

Figure 3-4. Verifying Known Input State for Processor

One important detail to keep in mind is some internal designs can include an ESD or flywheel diode that provides a leakage path on its output pins. When this is paired with an external pull up resistor, this can result in a backbiasing path that can potentially power up the Vcc rail. The system designer needs to evaluate if the device has this internal leakage current potential.

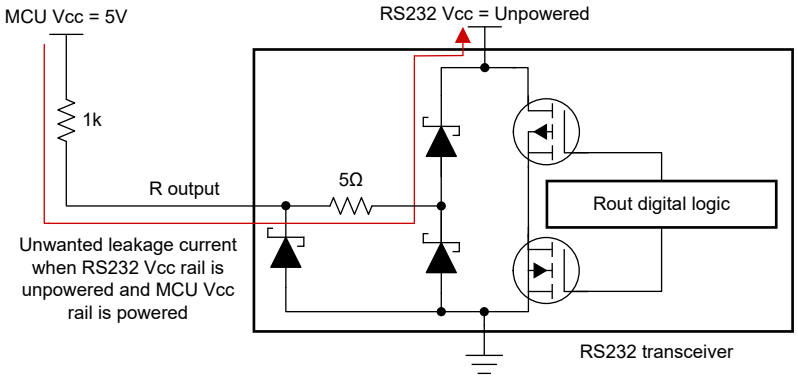

Figure 3-5. Potential Leakage Current Through External Pull Up Resistor Through ESD Cell

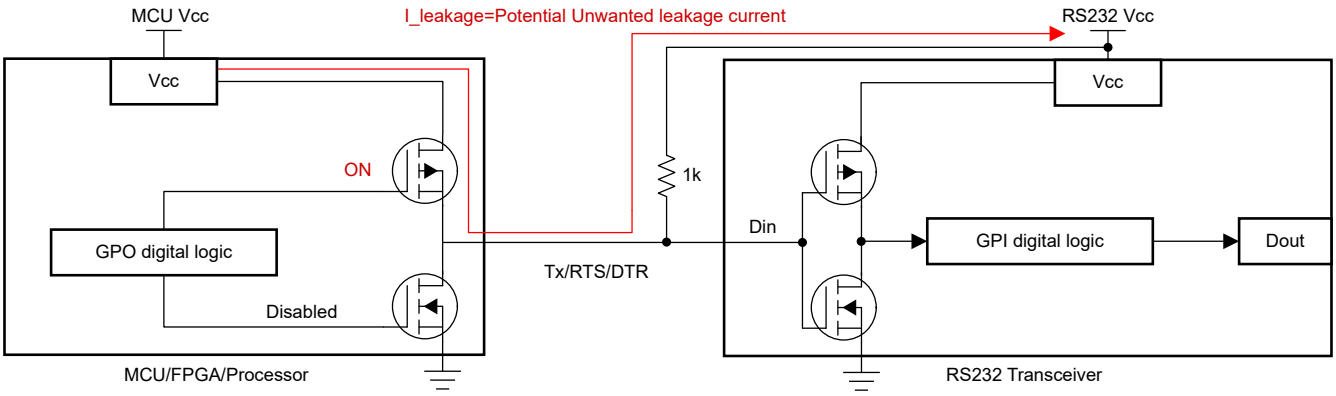

Figure 3-6. Potential Leakage Current Through External Pull Up Resistor if Processor PFET is Active

In the worst-case scenario, this can raise Vcc high enough to power the device up or other devices on the Vcc rail. Selecting a larger value pull up resistor limits the current and voltage build up and reduces the risk. The other workaround is to place a pull down resistor from Vcc of the unpowered rail to ground though this results in more power consumption when the rail is powered.

3.2 Limit Series Resistors and External Capacitance on the RS232 Pins

Adding series resistors or external capacitance on the RS232 line can add some level or protection against noise or shorts but will also limit the max data rate and max cable distance the RS232 system can support. It is recommend to keep series resistors low to less than a few hundred ohms. Any external capacitance must be kept in the lower hundreds of picofarads.

3.3 If External ESD is Required, Use Bidirectional TVS Diodes Not Unidirectional

Some systems require additional protection circuitry on the RS232 pins to clamp transient voltages and protect the device pins. This can be a good idea since the RS232 pins are more likely to come in contact with people, charged cables, or temporary shorts during hot insertion. TVS diodes can be added to combat these transients but RS232 pins, a typical TVS diode must not be used. This is because the RS232 pins swing from -5.5V or lower to +5.5V or higher which typically causes unidirectional TVS diodes to conduct during the negative swings. To prevent the negative signal from clamping, a bidirectional TVS diode must be used to support both voltage rail swings.

How to Choose a TVS Diode for RS-232, RS-485 and CAN

[FAQ] RS232 and RS485 Protection - Interface forum - Interface - TI E2E support forums

4 Schematic

Check	Comments
Schematic pinout matches datasheet pinout	Check this to rule out carelessness
Expected net names from RS232 side match TTL side	Caps placed on Vcc usually in the 0.1uF range. Optional: set up bulk capacitors using 1uF, 0.1uF, and 10pF to smooth out multiple frequencies
All input pins are biased (external pull up or down resistor) unless internally biased	This is to avoid additional leakage through shoot through currents and verifies signal integrity in case of noise or leakage if input is floating
Cap values match datasheet recommendations	Using values outside of datasheet suggest can cause electrical parameters to be violated
Limit series resistance and external cap to GND on RS232 pins	Can use a 0 ohm resistor in series, but exceeding 50 ohms or 200pF is not recommended.
Voltage rating on charge pump caps at least 2x of typical VoH/VoL	Verifies that caps do not break during transients and can extend the lifespan of the capacitor
Optional: If external ESD is used on RS232 pins, must be bi-directional TVS	This is to prevent unintended forward biasing if a uni-directional TVS diode is used on an RS232 pin
Verify Vcc power rail meets recommended operating conditions for device	Devices are designed and specified for specific voltage rails, this check verifies device operates as intended
RS232 nets are connected correctly to the physical connector pin assignments	RS232 has a standard for how DB9 connectors must be connected to UART nets to verify compatibility across systems Figure 7 shows DB9 (the most common connector for RS232)

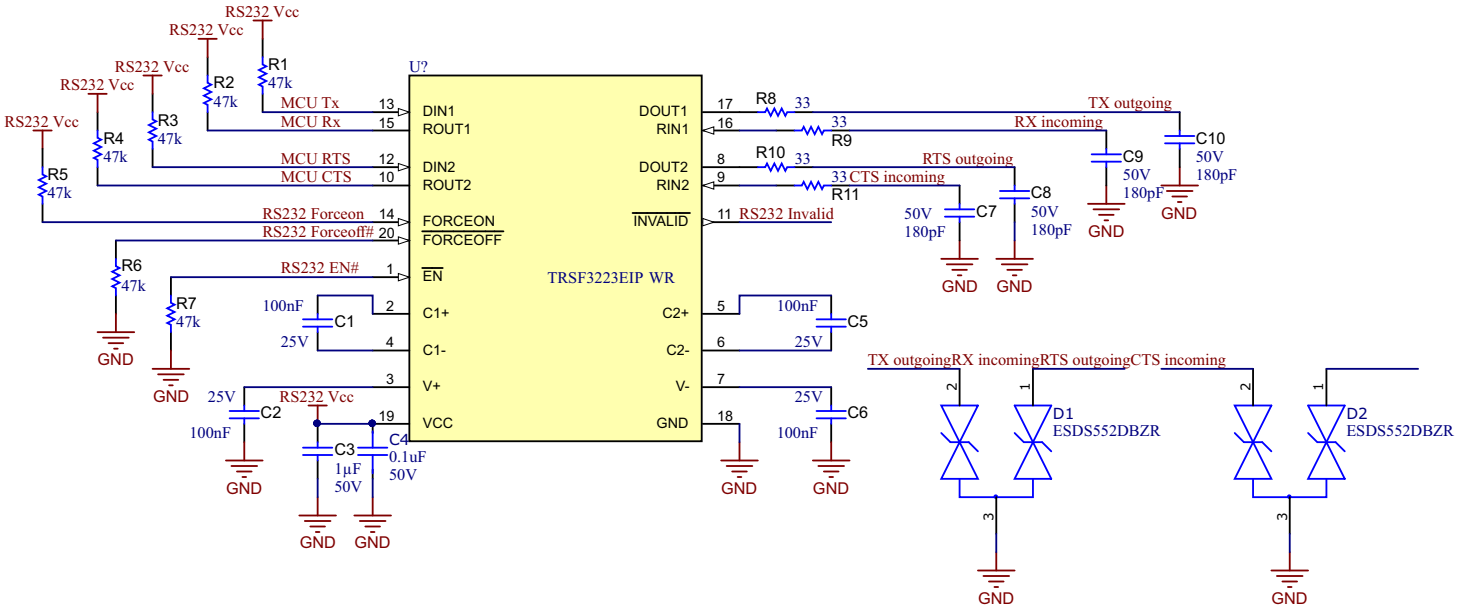

Figure 4-1. Example Schematic

5 PCB

- Decoupling caps and charge pump caps placed close as possible on PCB
- Due to switching frequency on charge pumps circuit and RS232 signals, place the RS232 device away from noise sensitive circuits to avoid AC coupling
- The RS232 device is usually closer to edge of PCB to interface with DB9 connector but does not need to be as close as possible since slew rate and data rate of device are slow and are not impacted by transmission theory of reflections
- Impedance matching and trace length matching are not required

6 Summary

Schematic and PCB design for RS232 circuits seems straightforward however small details such as biasing TTL inputs and outputs are not obvious for most designers. Missing these small details can cause unintended system issues such as glitches on the UART lines and can hang up the processor as a result. For this reason it is imperative that the designer keep TTL pins biased during scenarios of uncertainty such as the processor being powered up before the RS232 transceiver or vice versa.

7 FAQ

See [FAQ] RS-232 Transceiver Resources - Interface forum - Interface - TI E2E support forums for more information.

8 References

- Texas Instruments, *[FAQ] RS-232 Transceiver Resources*, FAQ.
- Texas Instruments, *How to Choose a TVS Diode for RS-232, RS-485 and CAN Based on Voltage Ratings*, application note.
- Texas Instruments, *Implications of Slow or Floating CMOS Inputs*, application note.
- Texas Instruments, *RS232 Glossary and Selection Guide*, application note.

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATASHEETS), 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 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, TI’s General Quality Guidelines, or other applicable terms available either on 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. Unless TI explicitly designates a product as custom or customer-specified, TI products are standard, catalog, general purpose devices.

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

Last updated 10/2025