ANALOG DEVICES

Technical Article
MS-2579

Digital Isolators Offer Easy to Use Isolated USB Option

by Eric Gaalaas, Analog Devices, Inc.

INTRODUCTION

Universal serial bus (USB) is a popular method for personal computers (PCs) to communicate with peripherals via cables. In some applications, it is desirable to isolate the USB communication to achieve safety requirements or break ground loops. Unfortunately, the isolation task is not trivial because of the bidirectional data flow on the USB cable. This article discusses this and other challenges in achieving easy to use isolated USB implementations and compares solutions. A 'transparent' ideal minimizes the impact of adding isolation to the system, and such solutions are now available. The discussion focuses on USB 2.0, which supports three data rates: 1.5 Mbps (low), 12 Mbps (full), and 480 Mbps (high). For simplicity, the 12 Mbps case is discussed most fully, but many principles in that example also apply to the other speeds.

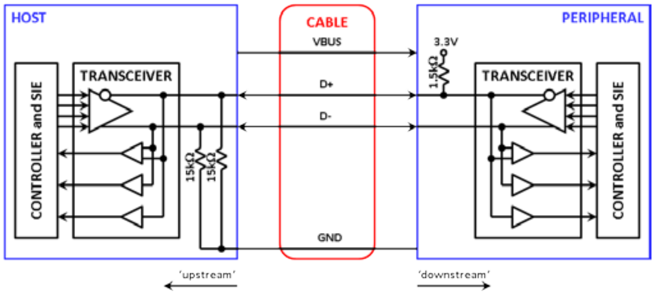

Figure 1. Full speed (12 Mbps) USB connection (nonisolated)

USB BASICS

One reason for USB's popularity is its simple 4-wire interface that provides power to a peripheral and a serial data link between the peripheral and a PC. Figure 1 shows a standard USB connection. The VBUS and GND wires provide 5 V power and ground, while D+ and D– carry data. The signaling is bidirectional half duplex, meaning that data can flow in either direction along the cable, but at any particular time, at most one transmitter actively drives the cable. During communication, the USB transmitters drive differential or single-ended states onto D+ and D–. Data is organized into packets, with special signal sequences indicating start-of-packet and end-of-packet. Sometimes the bus is idle, meaning that neither transmitter is active, and at these times, resistors attached to the ends of the cable establish 'idle' bus states at D+ and D–. The idle states help initialize the bus between one packet and the next. They also indicate to the host when the peripheral connects or disconnects, and the peripheral's desired communication speed (1.5 Mbps, 12 Mbps, or 480 Mbps).

METHODS OF ISOLATING A USB HOST AND PERIPHERAL

Now imagine electrically isolating the host and peripheral. As noted in [1], there are several options for placement of the isolation barrier. In all cases, multiple signals must be isolated, and the signals may run at fast speeds or bidirectionally, depending on where the isolation is located. This complicates implementations that are built from discrete components. The complete bill of materials can become long, and it may be difficult to find discretes that fully conform to signaling requirements.

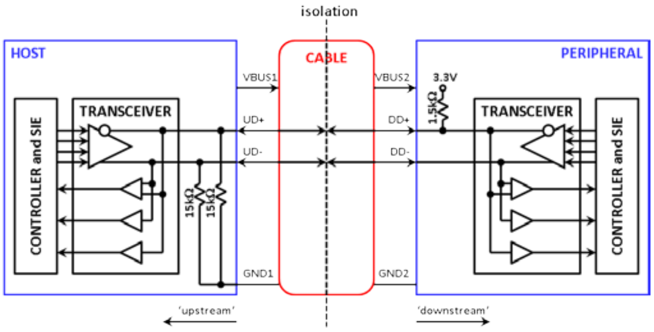

Figure 2a. Isolation splitting the cable (concept)

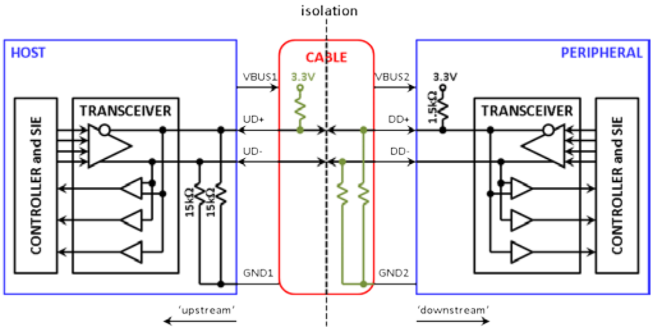

Figure 2b. Isolation splitting the cable, showing extra resistors

One isolation possibility is shown in figure 2a, where the dashed line shows isolation that conceptually splits the USB cable. Information about the state of D+ and D– can cross the barrier, but current does not. GND1 (the upstream side's ground reference) is now a separate node from GND2 (the downstream side's ground reference). Unfortunately, the isolation prevents the host from 'seeing' the downstream pull-up resistor, and the peripheral can't 'see' the upstream pull-down resistors. Therefore, some extra resistors are needed as shown in Figure 2b, to mimic the connection of their counterparts across the isolation. In this 'transparent' concept, communication between host and peripheral works very similarly to the nonisolated connection of Figure 1. The transparent USB isolator component is simply inserted between one of the transceivers and the USB cable, along with an isolated power supply. Hosts and peripherals originally designed for nonisolated applications easily connect to the USB isolator and exchange standard USB signals without needing significant modification.

This approach is very appealing provided the concept can really be implemented, but there are a number of challenges to overcome. For example, standalone optocouplers or digital isolators generally do not provide USB-compatible drive characteristics, or support bidirectional half-duplex communication. Many optocouplers cannot run at 12 Mbps or above, and have long propagation delays and timing errors that do not meet USB 2.0 timing requirements. There are other issues described later.

For now, let's shift focus and consider nontransparent alternatives like those discussed in [1]. Instead of bisecting the USB cable with isolation, these solutions place isolation inside the hardware of either the host or peripheral. It can go between the USB transceiver and serial interface engine (SIE), or between the SIE and USB controller. This enables unidirectional digital logic signals to be isolated with standalone, general-purpose isolators. However, there are several significant disadvantages. First, the USB transceiver or controller hardware must be customized to insert the isolation components. Extra microcontroller code or modification of the USB driver software may also be required. This creates extra work for the system designer, and can significantly increase required board space, as these solutions are complex and require multiple components. Another disadvantage is that overall data throughput may be reduced, since data is now being sent through a series combination of USB transceivers and the separate isolation scheme. The isolation scheme may add delays related to encoding and decoding into another serial format such as SPI, or delays related to slow speed or imprecise timing of the isolation components.

Despite these shortcomings, such solutions were the only viable options when it was too difficult to overcome the challenges of implementing the transparent USB isolator. Now, transparent solutions are available, and the remainder of this article describes how an example fully meets the requirements.

TRANSPARENT USB ISOLATOR REQUIREMENTS

A USB isolator system must satisfy several requirements to achieve fully 'transparent' operation:

1. It must drive UD+, UD–, DD+, and DD– in the same manner as a standard USB transceiver, and in fact contain two USB-compliant transceivers, one on each side of the isolation barrier (Figure 3).

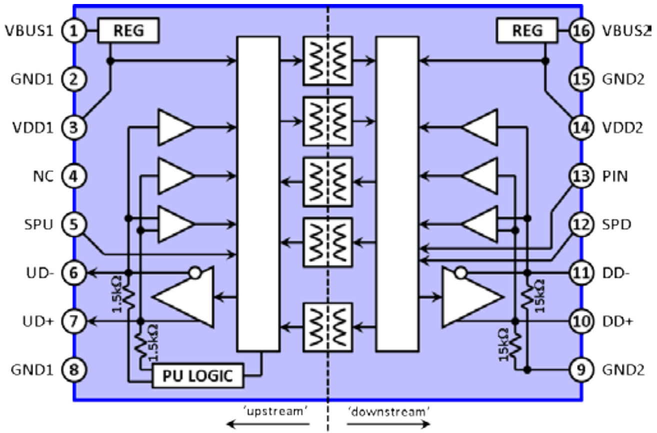

Figure 3. ADuM4160 block diagram

2. It must manage bidirectional communication on the USB cable by ensuring that its transceivers transmit and receive at appropriate times and accurately reproduce all driven and idle states. To accurately reproduce the idle states it must include a pull-up resistor on its upstream side to mimic the state of the pull-up resistor attached to the downstream peripheral. It may also include pull-down resistors on its downstream side. The bus must be monitored for signals that indicate idle bus, start-of-packet, and end-of packet to appropriately respond to those conditions.
3. Signal isolator components inside the USB isolator must communicate D+ and D– data back and forth across the isolation. If the signal isolators are unidirectional (as is generally the case), the USB isolator system needs multiple isolation channels, some transmitting in a downstream direction, and others transmitting in the opposite, upstream direction.
4. The signal isolators must run fast with accurate timing to support the desired USB signaling speeds, and conform to USB requirements for propagation delay and timing error.
5. Each side of the USB isolator should support power provided by 5 V or 3.3 V supplies. If 5 V power is provided, the isolator should derive a 3.3 V regulated supply suitable for powering that side's USB transceiver. If 3.3 V power is provided, the isolator can use it to directly power the USB transceiver and bypass its regulator.

A TRANSPARENT USB ISOLATOR REALIZATION

The Analog Devices ADuM4160 USB digital isolator[2] meets all requirements, and is integrated into a 16-lead SOIC package. A block diagram is shown in Figure 3. It contains a pair of USB transceivers, five channels of *i*Coupler®-based digital isolation, control logic, and two 'smart regulators.' It also includes a 1.5 kΩ upstream pull-up resistor, and 15 kΩ downstream pull-down resistors.

Its USB transceivers are controlled by a simplified controller, which doesn't need to fully decode and analyze the packets to support the isolation function. Instead, it can monitor UD+, UD–, DD+, and DD– for signals that indicate idle bus, start-of-packet, and end-of-packet, and use them to correctly enable or disable USB transmitters while ignoring packet content. When transmitting a packet downstream from host to peripheral, the upper two isolation channels in Figure 3 are active, as are the upstream USB receiver and downstream USB transmitter. Data is copied from UD+/UD– to DD+/DD–. When the packet ends, the end-of-packet sequence is detected and all USB transmitters disabled, allowing the bus to reach the idle state. If the peripheral subsequently starts transmitting a packet upstream, the USB isolator detects the start-of-packet sequence, enables the third and fourth isolation channels and upstream USB transmitter, and copies data from DD+/DD– to UD+/UD– until the packet ends. Then the bus is again returned to idle with all transmitters off, awaiting new data.

The ADuM4160 uses a fifth isolation channel to communicate the status of a control line on the downstream side[3], which activates a pull-up resistor integrated into the upstream side. This allows the downstream port to control when the upstream port attaches to the USB bus. The pin can be tied to the peripheral pull-up, a control line, or the VDD2 pin, depending on when the initial bus connect is to be performed. Attaching the pin to the peripheral pull-up enables its state to be mimicked by the upstream pull-up, and the ADuM4160's pull-downs mimic the state of those attached to the host. All active and idle states are copied from one side of the isolation to the other.

The isolation channels are digital isolators using chip scale transformers to achieve isolated communication. The individual channels can each operate beyond 100 Mbps, easily supporting 12 Mbps USB 'full speed' data. Integrating all channels together in a single chip enables tight control of timing, giving low timing error that meets USB timing requirements. Overall propagation delay through the ADuM4160 is equivalent to the delay through a standard USB hub. Quiescent power consumption is below USB limits for idle buses.

The smart regulators support the power supply options mentioned above in requirement 5, without requiring explicit user control[4]. To power one side of the USB isolator from 5 V (for example the upstream side), the 5 V supply is connected to the appropriate VBUS pin (e.g. VBUS1), while VDD1 is not connected. When sensors detect that voltage is applied to VBUS1 but not VDD1, they activate the 3.3 V regulator to power VDD1.

To instead power one side of the USB isolator from 3.3 V (for example the downstream side), the 3.3 V supply is connected to both VBUS2 and VDD2. When sensors detect applied voltage at both pins simultaneously, the on-chip regulator is disabled in order to directly use the externally provided 3.3 V.

CONCLUSION

The 'transparent' USB isolator in which isolation conceptually bisects the USB cable is very easy to use with USB hardware that was originally designed for nonisolated applications. This contrasts with alternatives, which place the isolation inside the host or peripheral hardware, requiring substantial hardware modification and sometimes degrading USB performance. The transparent concept is very challenging to implement using discrete components like off-the-shelf general-purpose isolators. However, recent integrated solutions like the ADuM4160 overcome the challenges in a single convenient package, greatly simplifying the addition of isolation in USB applications.

REFERENCES

[1] Hauck, Lane. "Isolating USB." *EDN Magazine*, July 2006.

[2] Information on ADuM4160, *i*Coupler digital isolators, and other Analog Devices products can be found at www.analog.com/*i*Coupler.

[3] U.S. Patent #8432182.

[4] U.S. Patent #8564327.

ABOUT THE AUTHOR

Eric Gaalaas [eric.gaalaas@analog.com] is a mixed-signal integrated circuit design engineer with the *i*Coupler Digital Isolator Group at Analog Devices Inc. He has a BS degree and MEng degree in electrical engineering from Cornell University. Eric is an author of many related publications, and a holder of nine patents.

REFERENCES

Share this article on facebook twitter

NOTES

One Technology Way • P.O. Box 9106 • Norwood, MA 02062-9106, U.S.A.
Tel: 781.329.4700 • Fax: 781.461.3113 • www.analog.com