Technical Article

Intro to I2C: What the Internet Doesn't Tell You

Kaitlyn Mazzarella

I trust Internet search engines as my first source to provide me with reliable and quick answers, so when I first encountered the I^{2}C protocol, I asked my trusted-source to help out. My search responded with: the Inter-Integrated Circuit (I^{2}C) protocol is a bi-directional two-wire serial bus that provides a communication link between integrated circuits. This stated the obvious, but I still had so many more questions: What is it really? What type of devices use I^{2}C? How will I^{2}C help me solve real problems in my system?

I^{2}C is a popular communication protocol that enables a master device such as a processor, microcontroller (MCU) or application-specific integrated circuit (ASIC) to communicate with other peripheral devices on the same two-wire bus. One line is dedicated for data transmission and the other for a clock signal. Think of it like a two-lane highway: each lane has cars flowing from one end to the other, just as a data packet would be transmitted from the master (processor, MCU, ASIC) to the peripheral device (temperature sensors, humidity sensors and more).

So that's what I^{2}C actually is … now, why would you want to use it? I^{2}C makes it easy to implement multiple peripherals on the same bus – for example, using various sensors to monitor the temperature of a server. The I^{2}C protocol was actually designed to support multiple devices on a single bus, while other protocols like serial peripheral interface (SPI) were designed for point-to-point single-device support. The two-wire I^{2}C interface can also help simplify routing versus the four-wire SPI interface and reduce general-purpose inputs/outputs (GPIOs).

TI's portfolio for building your I^{2}C tree can be separated into three functions, as shown in Figure 1: buffered translators, I/O expanders and I^{2}C switches.

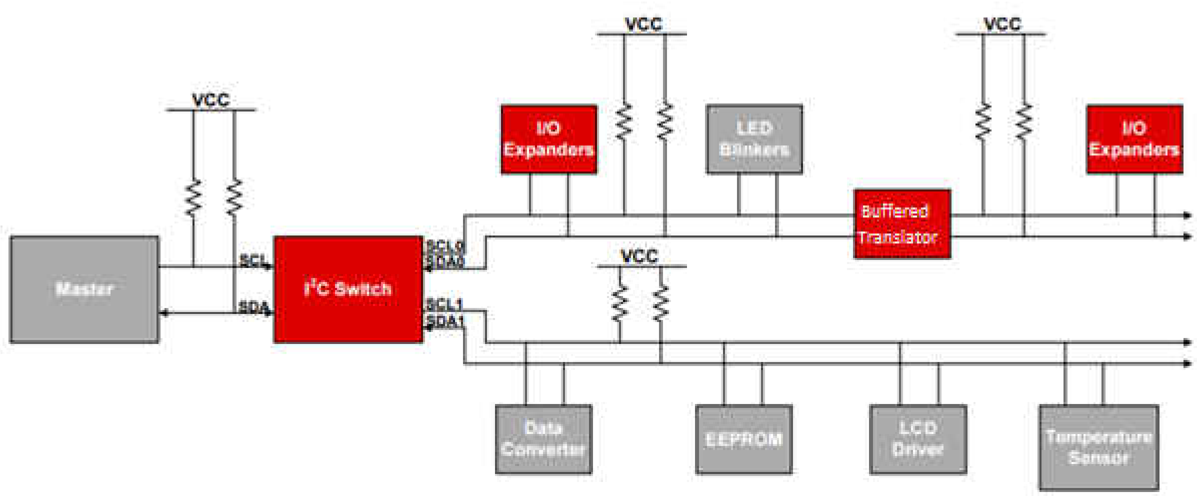
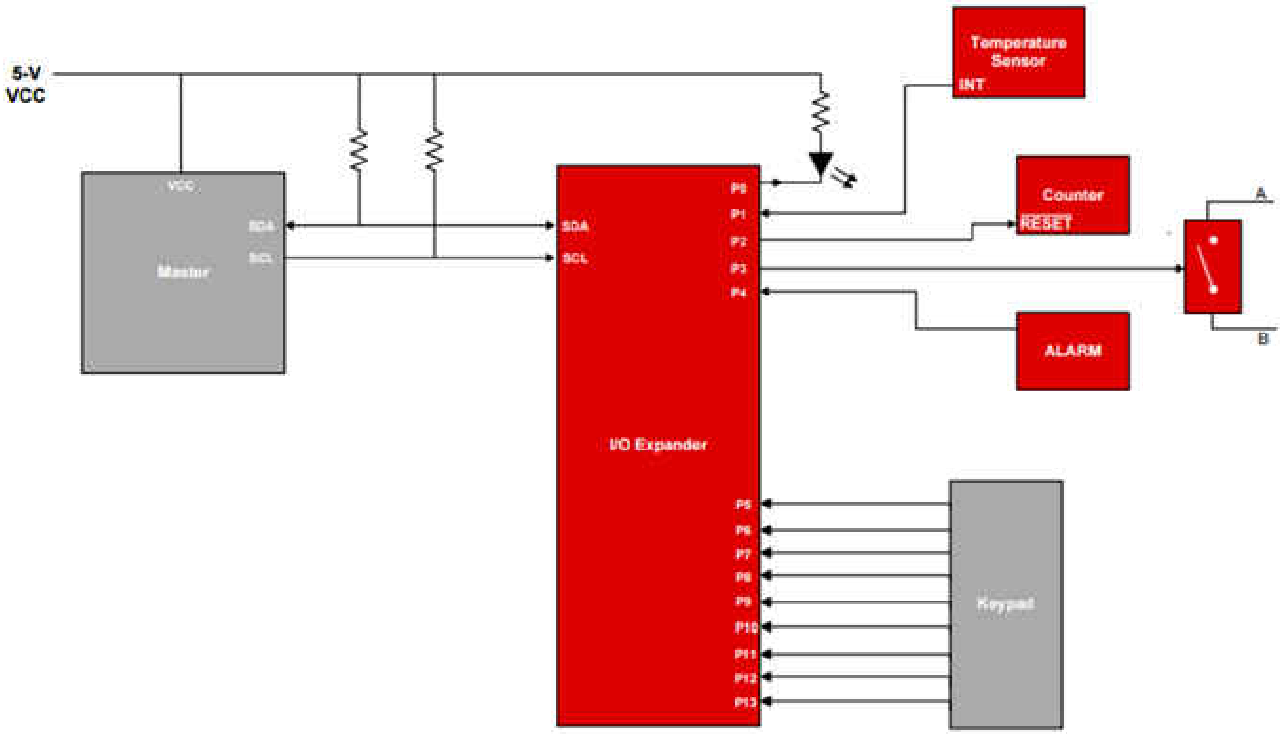


Figure 1. An Electronic System with Various I^{2}C Functional Devices

I^{2}C buffered translators can help when your design encounters degraded signals over a long trace and/or voltage mismatch between integrated circuits (ICs). For example, if you find that you have a voltage mismatch between a 3.3V MCU I^{2}C I/O and a 5V temperature sensor, you can use an I^{2}C voltage translator such as the PCA9306. Additionally, with that same voltage mismatch over a long trace, you can use a buffered translator such as the TCA9617B to address the voltage difference while outputting a refreshed signal.

I^2C I/O expanders do exactly as the name suggests – they expand the number of I/Os when your selected processor runs out of available ports, as shown in Figure 2. These devices differ from standard GPIO expanders because they communicate directly with devices on the I^2C bus. This portfolio also offers a variety of features depending on your specific system needs, such as channel count, reset pins and level-shifting.



NOTE: Arrows distinguish if I/O is an input or output.

Figure 2. A Microcontroller or Processor Using an I/O Expander to Control Peripherals

I^2C switches can solve two different problems. The first problem is related to address-conflicts that commonly occur when two devices on the same bus have the same address. If a system has two temperature sensors and two humidity sensors on the same I^2C bus – all with the address 0xFF – an I^2C switch can switch to a single sensor at a time; see Figure 3. The second problem is related to power consumption. I^2C switches power only one portion of the I^2C bus at a time, thus keeping unused portions of the bus powered down when not in use.

What initial internet searching will try to tell you is that I^2C makes it easy for system designers to implement robust system controls. Not only is this a useful protocol, but there are many device functions that can benefit your system. You can use buffered repeaters, I/O expanders and I^2C switches to enhance, extend and modify your I^2C bus implementation in order to meet your unique design objectives. For more information on I^2C design, read "Understanding the I^2C Bus."

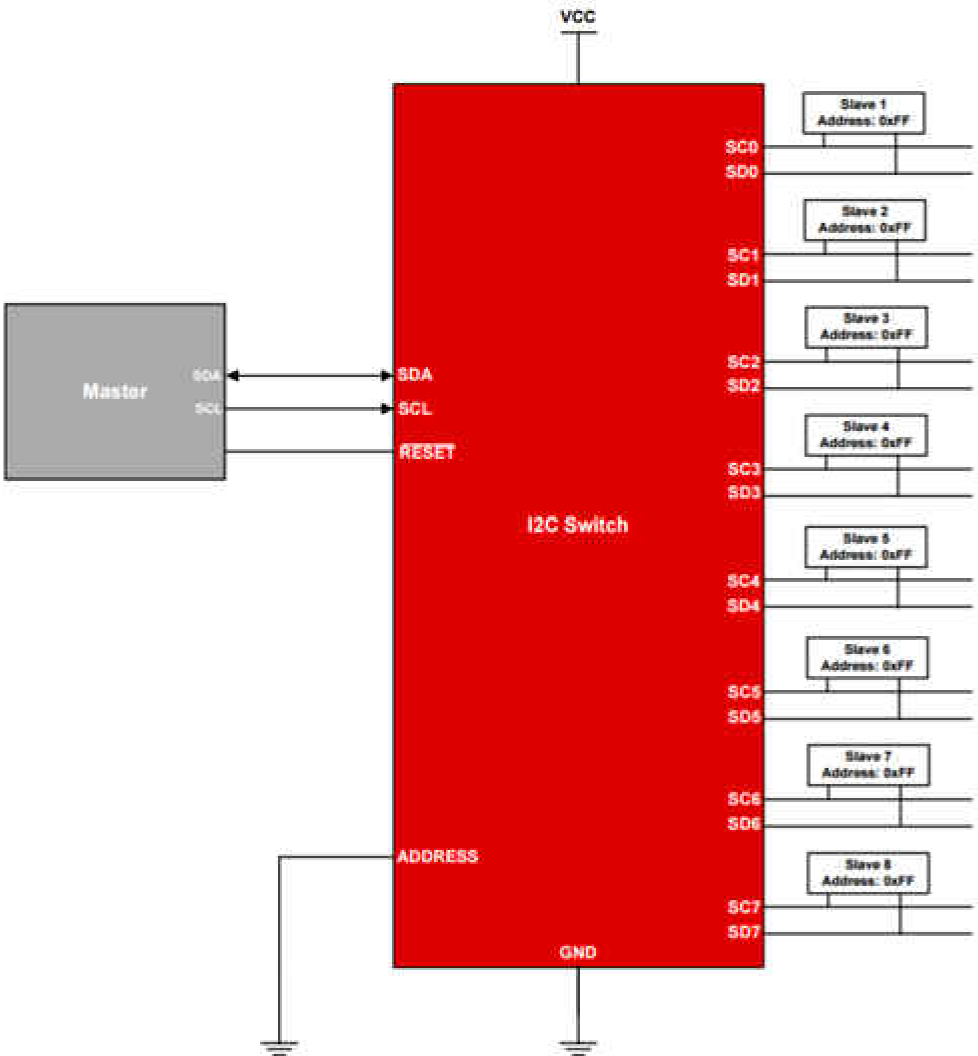


Figure 3. I^{2}C Switch Example with Eight Channels

Additional Resources

- Learn more about TI's portfolio of I^{2}C buffered translators, I/O expanders and switches.