

## APPLICATION NOTE 476

# Comparing the I<sup>2</sup>C Bus to the SMBus

*Abstract: The I<sup>2</sup>C bus and the SMBus™ are popular 2-wire buses that are essentially compatible with each other. Normally devices, both masters and slaves, are freely interchangeable between both buses. Both buses feature addressable slaves (although specific address allocations can vary between the two). The buses operate at the same speed, up to 100kHz, but the I<sup>2</sup>C bus has both 400kHz and 2MHz versions. Complete compatibility between both buses is ensured only below 100kHz. This application note focuses on the significant differences between I<sup>2</sup>C and SMB.*

The I<sup>2</sup>C bus and the SMBus are popular 2-wire buses that are essentially compatible with each other. Normally devices, both masters and slaves, are freely interchangeable between both buses. Both buses feature addressable slaves (although specific address allocations can vary between the two buses). The buses operate at the same speed, up to 100kHz, but the I<sup>2</sup>C bus has both 400kHz and 2MHz versions. Obviously, complete compatibility between both buses using all devices is ensured only below 100kHz.

This application note focuses on the significant differences between the two buses. Although it is assumed that the reader has some knowledge of the I<sup>2</sup>C bus and/or the SMBus, let's first review some protocol basics:

- **Start and Stop events.** These are especially important in that they are ways of signaling to an interface that it needs to go to an initialized or reset state.
- **Data and Clock must be high to generate Start and Stop.** A master can't generate a Start or Stop unless both the Data (SDA for I<sup>2</sup>C and SMBData for SMBus) and Clock (SCL for I<sup>2</sup>C and SMBClk for SMBus) lines are free (not pulled low). This is a consequence of being an open-collector bus.
- **Start and Stop conditions are the *only* times there will be a transition on the Data line while Clock is high.**
- **Data can change state only when Clock is low during a communication.** The data on Data must always be ready just prior to a high on Clock and be changed only after Clock has gone low (with the exception of Start and Stop).

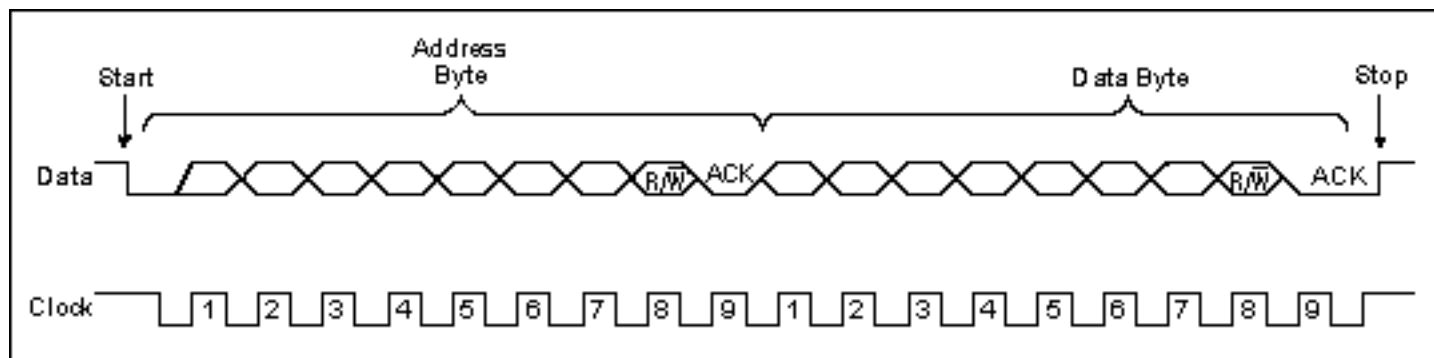


Figure 1. A typical communication, showing the Start and Stop conditions.

## Timeout and Clock Speed

Timeout and (as a consequence of timeout) minimum clock speed are the most important differences between the I<sup>2</sup>C bus and the SMBus.

I<sup>2</sup>C Bus = DC (no timeout)

SMBus = 10kHz (35mS timeout)

Timeout is where a slave device resets its interface whenever Clock goes low for longer than the timeout, typically 35mSec. Use of a timeout also dictates a minimum speed for the clock, because it can never go static. Thus, the SMBus has a minimum-clock-speed specification. By comparison, the I<sup>2</sup>C bus can go static indefinitely. In the I<sup>2</sup>C bus, either a master or a slave can hold the clock low as long as necessary to process data.

All of this comes about as a result of how the two buses deal with slave errors and recovering from those errors. To understand error recovery, consider that there are two states, Start and Stop, which on a properly designed I<sup>2</sup>C or SMBus slave dictate that the slave interface go to a specific state. In the case of a Start, the interface should initialize itself and be ready to receive a communication. This should occur regardless of what preceded the Start condition (for example, if the slave was in the middle of a communication in which the master got confused and had to start over). In the case of a Stop, an interface should again initialize itself, but should be expecting a Start prior to any new communication. Both Starts and Stops are the only Data transitions that take place when Clock is high. In order for the transitions on Data to take place, the Data line (and the Clock line) must be "free" to allow the master to place highs and lows on the line as it needs.

In the I<sup>2</sup>C bus, if the slave locks up and holds either Clock or Data low, error recovery is impossible. Very few slave devices actually have the ability to hold Clock. As a result, the most common bus error is slave devices that have ended up in a state where Data (the data line) is low. In the I<sup>2</sup>C bus, a master accomplishes error recovery by clocking Clock until Data is high and then issuing a Start followed by a Stop.

In contrast to the I<sup>2</sup>C bus, SMBus slaves are expected to reset their interface whenever Clock is low for longer than the timeout specified in the SMBus specification of 35mS. As such, SMBus masters such as the Intel PIIX4 don't have any error recovery routine built in. In reality, slave devices need to examine both Clock and Data, even though the SMBus specification doesn't dictate that. This should be done, as experience has shown that slaves in an error state can hold Data low, even though Clock can be high. Implementing timeout exactly according to the SMBus specification doesn't address this error mode.

**Table 1. Considerations for Mixing Devices and Masters with Regard to Timeout**

	I <sup>2</sup> C Slave	SMB Slave
I <sup>2</sup> C Master		Clock must be > 10kHz
SMB Master	Potential for bus lockup	

The SMBus is limited to a clock speed of 100kHz, whereas I<sup>2</sup>C permits speeds up to 400kHz.

## Logic Levels

Despite differences in logic-level specifications between the two buses, generally devices can be mixed and matched with abandon. It has proven rare that either an SMBus or an I<sup>2</sup>C bus has had a fault due to level differences. This may come as a surprise when we compare the respective level specifications as shown in Figure 1.

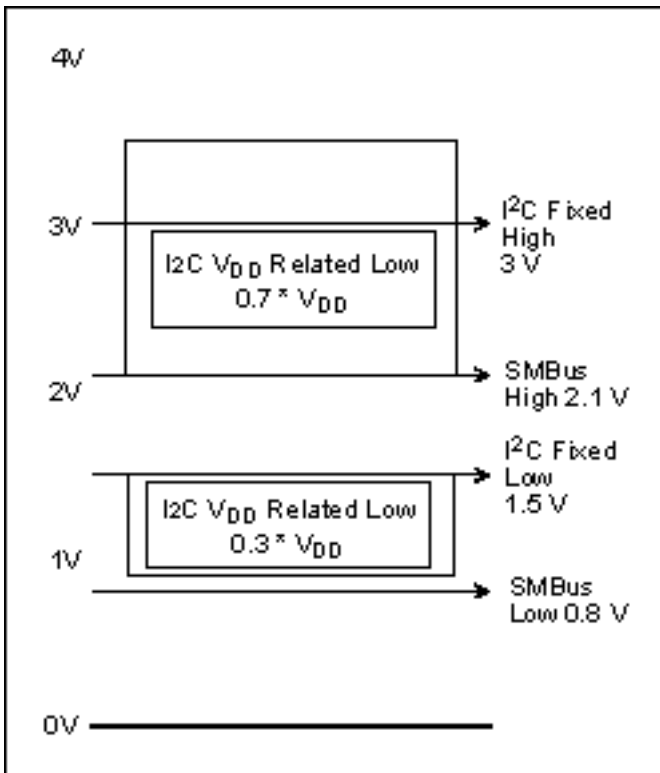


Figure 2. This is a comparison of level specifications between the I²C bus and the SMBus. I²C- $V_{DD}$ -related high and low value ranges are shown for power supplies from 3 volts to 5 volts.

For further comparison, Table 2 lists the specifications together.

**Table 2. Level Specifications for the I²C Bus and the SMBus**

$V_{DD}$  is typically expected to be between 3 volts and 5 volts.

High	I²C $V_{DD}$ Dependent	$0.7 \times V_{DD}$
	I²C Fixed	3.0V
	SMBus	2.1V
Low	I²C $V_{DD}$ Dependent	$0.3 \times V_{DD}$
	I²C Fixed	1.5V
	SMBus	0.8V

Although there may seem to be quite a bit of difference between level specifications, device interchangeability has not been shown to be a problem in this regard. This is because masters and slaves virtually always have output-voltage swings close to the full supply voltage.

## Pullup Resistors and Current Levels

What wasn't readily evident in the discussion of voltage levels is that these levels are also specified with different currents between the buses. The SMBus specifies a minimum sink current of 100µA, and a maximum of 350µA, compared to 3mA for the I²C bus. This in turn would determine the lowest acceptable value of the pullup resistor, examples of which are shown in the table below.

**Table 3. Minimum Pullup Resistor Values per Bus Specifications**

	3V $V_{DD}$	5V $V_{DD}$
I <sup>2</sup> C Bus	> 1k	> 1.6k
SMBus	> 8.5k	> 14k

In reality, it is not unusual to encounter pullup resistor values in SMBus systems, which violate this specification by being lower than recommended. A very popular range for pullup resistor values, even in some SMBus systems, seems to be 2.4k to 3.9k.

## General Call and Alert Response

The I<sup>2</sup>C bus has an address called General Call, which is 0000 000, and all slaves designed to respond to the General Call will act accordingly. This is a means by which several devices on the system can be communicated with simultaneously.

The SMBus also has a provision for a line called ALERT#, not to be confused with the I<sup>2</sup>C General Call address. This line acts as an interrupt to the SMBus master. Upon receipt of an interrupt, the SMBus master can issue an Alert Response. This Alert Response is sent to address 0001 100, and any slave device that generated an interrupt attempts to identify itself by putting its own address on the bus. The device with the lowest address will dominate (due to the open-collector bus). If it is serviced and cleared, the master can repeat the process and work up through any other interrupts that may have occurred.

## Other Little Details

There are subtler issues having to do with rise and fall times and bus capacitance that are generally not a concern. In addition, there are some protocol differences with regard to use of Acknowledge and No Acknowledge conditions that are rarely encountered. With most mainstream choices of masters and slaves, none of these factors is typically a problem.

## Summary of Significant Differences

1. Timeout and minimum clock speed
2. Voltage levels
3. Pullup resistor values and current levels

**Table 4. Comparisons**

	I <sup>2</sup> C	SMBus
Timeout	No	Yes
Minimum Clock Speed	DC	10kHz
Maximum Clock Speed	100kHz (400kHz and 2MHz also available)	100kHz
$V_{HIGH}$	$0.7 \times V_{DD}$ , 3.0V Fixed	2.1V
$V_{LOW}$	$0.3 \times V_{DD}$ , 1.5V Fixed	0.8V
Max I	3mA	350 $\mu$ A
Clock Nomenclature	SCL	SMBCLK
Data Nomenclature	SDA	SMBDAT
General Call	Yes	Yes
Alert#	No	Yes

SMBus is a trademark of Intel Corp.

---

Application Note 476: [www.maxim-ic.com/an476](http://www.maxim-ic.com/an476)

#### **More Information**

For technical support: [www.maxim-ic.com/support](http://www.maxim-ic.com/support)

For samples: [www.maxim-ic.com/samples](http://www.maxim-ic.com/samples)

Other questions and comments: [www.maxim-ic.com/contact](http://www.maxim-ic.com/contact)

---

#### **Automatic Updates**

Would you like to be automatically notified when new application notes are published in your areas of interest?

[Sign up for EE-Mail™.](#)

AN476, AN 476, APP476, Appnote476, Appnote 476

Copyright © by Maxim Integrated Products

Additional legal notices: [www.maxim-ic.com/legal](http://www.maxim-ic.com/legal)