Application Report
SLOA132–April 2009

SMBus Compatibility With an I²C Device

Stephen Crump *Audio Products*

ABSTRACT

An I²C device usually can be controlled by an SMBus host, but there are hardware and software differences with which a designer must deal. This paper offers solutions for this issue.

Introduction

There are systems in which it is desirable to control an I²C device with an SMBus host, to simplify the systems and reduce cost. Generally it is possible to do this, but there are hardware and software differences between the 2 buses that must be considered to achieve compatibility. This paper examines these differences and recommends measures to ensure success.(1)

(1) For full details consult SMBus and I²C specifications. SMBus version 2.0 and I²C version 2.1 were used here.

General Comparison

SMBus is built on I²C and is therefore generally compatible with I²C devices, though not in all respects. SMBus Specification Version 2.0 includes an Appendix B that discusses differences in electrical specifications between I²C and SMBus. The following 2 tables summarize the differences in DC (level) and AC (timing) specifications and compatibility. Notes that follow the tables discuss a number of differences in communication.

I²C allows several modes, Standard, Fast and High-Speed. Standard mode allows clock frequencies as high as 100kHz while Fast and High-Speed modes are faster. SMbus allows clock frequencies only as high as 100kHz, so it is not necessary to compare SMBus to any but I²C Standard mode.

DC (Level) Specification Differences

PARAMETER		Std.I²C		SMBus		UNIT	COMPATIBLE?
		MIN	MAX	MIN	MAX		
V_{IL} Input LO	Fixed level	–0.5	1.5	–	0.8	V	Y
	Re. V_{DD}	–0.5	0.3 V_{DD}	N/A	N/A	V	case-by-case
V_{IH} Input HI	Fixed level	3	V_{DD}max + 0.5	2.1	–	V	**NO**
	Re. V_{DD}	0.7 V_{DD}	V_{DD}max + 0.5	N/A	N/A	V	case-by-case
V_{OL} Output LO	V_{OL} at 3mA	0	0.4	N/A	N/A	V	Y
	V_{OL} at 350 µA	N/A	N/A	0	0.4	V	
I_{PULLUP}	Pullup current	N/A	N/A	100	350	µA	–
I_{LEAK}	Input leakage	–10	10	-5	5	µA	case-by-case

Specifications for SMBus V_{IH} and a fixed-level I²C device are not compatible, but the following apply.

- If minimum V_{IH} of a target I²C device is 2.1V or less, compatibility is ensured.
- If minimum V_{IH} of a given SMBus system is 3V or higher, compatibility is ensured.

Compatibility of SMBus with an I²C device with V_{DD}–relative V_{IL} and V_{IH} is ensured for V_{DD} of 2.67 to 3.0 V, with maximum V_{IL} of 0.8V and minimum V_{IH} of 2.1V for the device. It is not ensured otherwise unless compatible V_{IL} and V_{IH} can be verified for the system and/or the device.

AC (Timing) Specification Differences

PARAMETER		Std.I²C		SMBus		UNIT	COMPATIBLE?
		MIN	MAX	MIN	MAX		
f_{scl}	Clock speed	–	100	10	100	kHz	Y
$t_{HD;DAT}$	Data hold time	0 (but an I²C device must hold internally 300nS)	3.45 (only if a device does NOT stretch clock LO time)	0.3	–	µs	Y

SMBus places a timeout period of 35ms on clock LOW state, after which a device holding the clock low must release the clock bus. SMBus also permits master and slave devices to extend the clock LOW time but limits the time extension. I²C has no similar specifications.

I²C and SMBus have the same specifications for clock and data rise and fall times, but I²C does not define how these times apply to waveforms. SMBus defines them as the time for transition between (V_{ILMAX} – 0.15V) and (V_{IHMIN} + 0.15V).

SMBus allows for bus arbitration in cases in which multiple transmitters generate Start commands at the same time. I²C does not specify bus arbitration.

Addressing Compatibility

Generally 7- and 10-bit addressing are the same for the 2 buses. However, there are addresses with assigned uses for one or the other of the 2 buses that may not be used as device addresses.

- 0000 1xx x = I²C HS-mode master code; SMBus reserved for future use.
- 0101 000 x = SMBus reserved for ACCESS.bus host.
- 0110 111 x = SMBus reserved for ACCESS.bus default address.
- 0001 100 x = SMBus Alert Response address.
- 1100 001 x = SMBus Device Default address.

Software Differences

SMBus protocols for message transactions are generally different from I²C data transfer commands. It is still possible to program an SMBus master to deliver I²C data transfer commands unless the SMBus master's software is dedicated only to SMBus protocols. However, the SMBus data transfer command form that includes what is called Packet Error Checking (PEC) must not be used with an I²C device.

The SMBus protocols "Send Byte" and "Receive Byte" are directly compatible with the I²C data transfer commands "Single-Byte Write" and "Single-Byte Read".

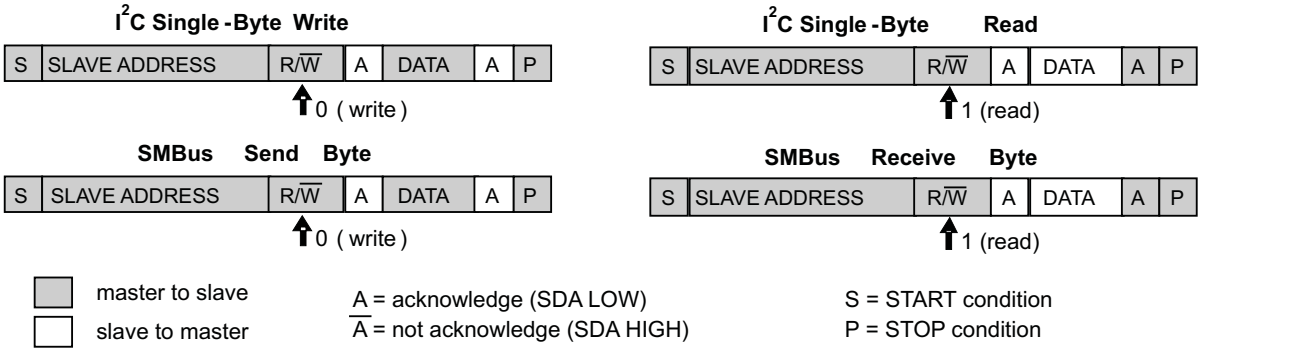

Figure 1. I²C Single-Byte Write and Read Compared to SMBus Send and Receive Byte

The SMBus specification does not define protocols identical to the I²C data transfer commands Multi-Byte Write and Read. The closest SMBus protocols are Block Write and Block Read, which include Command

and Byte Count bytes not found in the I²C sequences. The Command and Byte Count segments are required to communicate required action and the number of data bytes to be transferred to SMBus devices. However, if they can be programmed as a subaddress or data byte and an additional data byte, Block Write can be used to write data to an I²C device. The I²C device will continue to increment its subaddress and accept data bytes until it receives a Stop command.

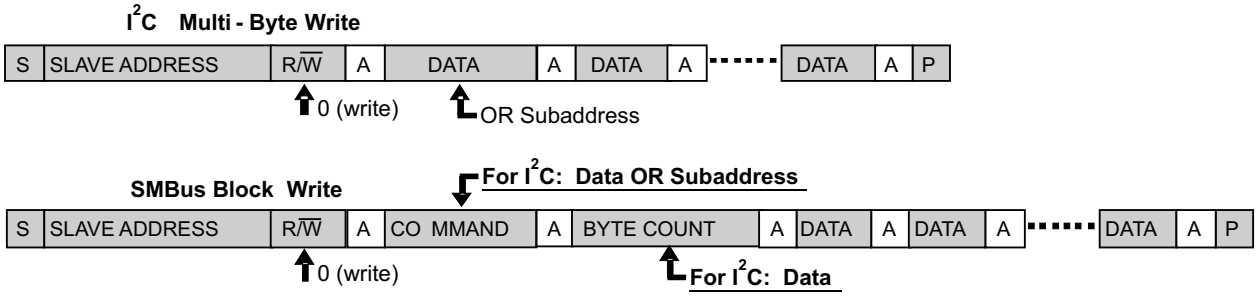

Figure 2. I²C Multi-Byte Write Compared to SMBus Block Write

The SMBus protocol Block Read includes a re-Start after the Command byte to reverse the direction of the transfer, so it is not directly compatible with the I²C data transfer command Multi-Byte Read. But if an SMBus controller can be programmed to issue a protocol like a Block Write with a Read instruction rather than a Write, it can communicate with an I²C device in the same way as Block Write.

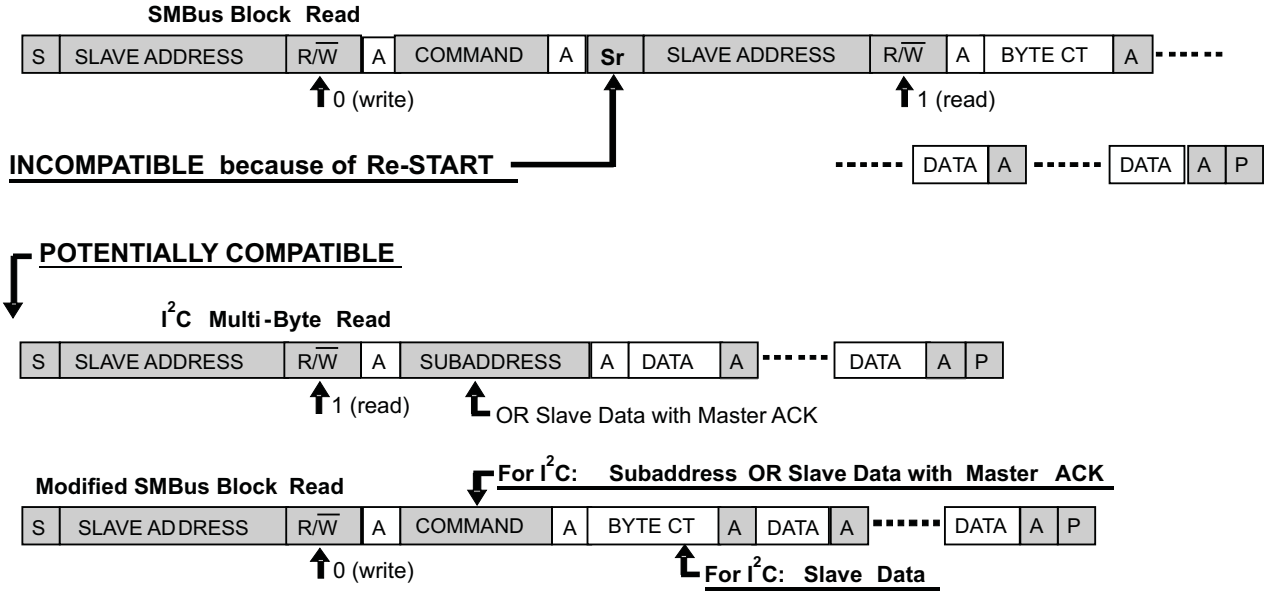

Figure 3. I²C Multi-Byte Read Compared to SMBus Block Read

I²C includes a set of data transfer commands called combined format. These include both reads and writes in a single command. Reads and writes are separated by a re-Start command followed by the slave address, repeated, with the Read-/Write bit reversed to change the direction of transfer. In this respect these commands are similar to SMBus Block Read. However, the SMBus protocols must begin with the slave address, a WRITE command and a command byte, and so the re-Start is required to change direction in SMBus Block Read. The SMBus protocols include not only the command bytes but byte count bytes as well, as shown in the SMBus Block Write and Read sequences above. The use of these must be modified as shown above for compatibility with I²C.

Clearly a controller dedicated only to SMBus protocols could not easily communicate a Multi-Byte Write or

Read to an I²C device. However, it is unlikely that controllers will be so restricted. The basic communication pattern, a series of bytes each followed by an ACK, is the same for both buses. So a controller with even minimal flexibility should be able to structure sequences that communicate either SMBus protocols or I²C data transfer commands. Such a device could work compatibly with both types of devices. It is necessary to review the details of the 2 bus specifications to ensure this.

PEC (Packet Error Checking or Packet Error Code)

A Packet Error Code may be attached to most SMBus commands for error checking. There is no corresponding transaction defined for I²C. The following diagram illustrates one case of an SMBus command with PEC, a Block Write with the PEC as its last transmitted byte.

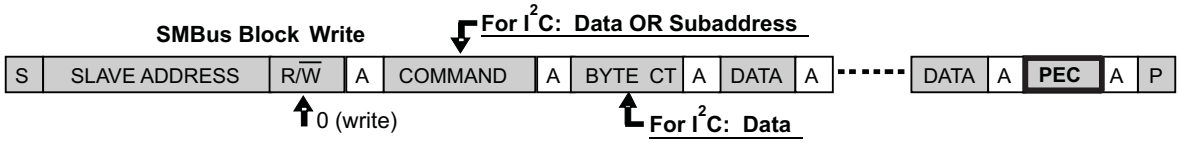

Figure 4. SMBus Packet Error Checking (PEC)

There is a risk of register corruption if an SMBus send or write command that includes a PEC is directed to an I²C device. Typically an I²C device will increment its register address with each succeeding write. Since it will not be able to distinguish a PEC from a data byte, it will write the PEC to the next address after the address for the last legitimate write, corrupting the next register. Of course, adding a PEC to a transaction with an I²C device is an error, since the device cannot know what it is.

Other Software Differences

There are differences in the use of the NACK signal. It is necessary to allow for these to ensure compatibility.

- An I²C slave receiver is allowed to NACK the slave address, if it is unable to receive (for example, because it is performing some real time task). The master can then generate a STOP signal or a repeated START signal to repeat the transfer. SMBus requires devices always to acknowledge their own addresses, to ensure detection of a removable device on the bus.
- An I²C slave may acknowledge its own address but later in the transfer decide that it cannot receive any more data. The device may indicate this by generating a NACK on the first byte to follow.
- SMBus can use the NACK also to indicate the reception of an invalid command or data.

Summary

This is not a complete catalog of differences between the 2 buses. A comprehensive review of these is beyond the scope of this paper, though it attempts to cover the majority of the ground between them. Generally SMBus is compatible with I²C devices, but there are a number of differences that can make them incompatible. These differences must be considered and corrections provided where they produce incompatibility.

Many I²C devices comply with SMBus hardware requirements in spite of being I²C compatible or compliant. These devices can operate directly from the hardware signals from an SMBus master. It is necessary to compare specifications for the target device to SMbus requirements and even the specification for the SMBus host to determine this.

Similarly it will typically be possible to program an SMBus master to communicate correctly with an I²C device. In some cases the SMBus master must be programmed with commands that are not found in the SMBus specification. Also, it will always be necessary to avoid using a PEC in a transaction between an SMBus master and an I²C device. Generally, though, SMBus masters will be flexible enough in their programming to accommodate these needs.

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

TI assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using TI components. To minimize the risks associated with customer products and applications, customers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information published by TI regarding third-party products or services does not constitute a license from TI to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of TI information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

Resale of TI products or services with statements different from or beyond the parameters stated by TI for that product or service voids all express and any implied warranties for the associated TI product or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

TI products are not authorized for use in safety-critical applications (such as life support) where a failure of the TI product would reasonably be expected to cause severe personal injury or death, unless officers of the parties have executed an agreement specifically governing such use. Buyers represent that they have all necessary expertise in the safety and regulatory ramifications of their applications, and acknowledge and agree that they are solely responsible for all legal, regulatory and safety-related requirements concerning their products and any use of TI products in such safety-critical applications, notwithstanding any applications-related information or support that may be provided by TI. Further, Buyers must fully indemnify TI and its representatives against any damages arising out of the use of TI products in such safety-critical applications.

TI products are neither designed nor intended for use in military/aerospace applications or environments unless the TI products are specifically designated by TI as military-grade or "enhanced plastic." Only products designated by TI as military-grade meet military specifications. Buyers acknowledge and agree that any such use of TI products which TI has not designated as military-grade is solely at the Buyer's risk, and that they are solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI products are neither designed nor intended for use in automotive applications or environments unless the specific TI products are designated by TI as compliant with ISO/TS 16949 requirements. Buyers acknowledge and agree that, if they use any non-designated products in automotive applications, TI will not be responsible for any failure to meet such requirements.

Following are URLs where you can obtain information on other Texas Instruments products and application solutions:

Products		**Applications**	
Amplifiers	amplifier.ti.com	Audio	www.ti.com/audio
Data Converters	dataconverter.ti.com	Automotive	www.ti.com/automotive
DLP® Products	www.dlp.com	Broadband	www.ti.com/broadband
DSP	dsp.ti.com	Digital Control	www.ti.com/digitalcontrol
Clocks and Timers	www.ti.com/clocks	Medical	www.ti.com/medical
Interface	interface.ti.com	Military	www.ti.com/military
Logic	logic.ti.com	Optical Networking	www.ti.com/opticalnetwork
Power Mgmt	power.ti.com	Security	www.ti.com/security
Microcontrollers	microcontroller.ti.com	Telephony	www.ti.com/telephony
RFID	www.ti-rfid.com	Video & Imaging	www.ti.com/video
RF/IF and ZigBee® Solutions	www.ti.com/lprf	Wireless	www.ti.com/wireless

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265