

Implementing SMBus using MSP430 Hardware I²C

*Harman Grewal**MSP430*

This application report describes the implementation of the system management bus (SMBus) using the MSP430 hardware I²C peripheral. SMBus is used as a communication link for smart batteries, power related devices and a wide variety of other system devices. This report includes the support for master and slave protocols in a SMBus communication system.

1.1 Introduction

SMBus is a two-wire serial interface based on the principles of I²C. The two lines are serial clock (SCL) and serial data (SDA) which are tied to V_{CC} using pull-up resistors. The devices communicating on this bus can drive the lines low or release them to high impedance. This connection is a wired-AND configuration. Multiple I²C/SMBus devices can be connected on the bus, but MSP430 pins should not be pulled above V_{CC}. For example, if the V_{CC} of the MSP430 is at 3V, then all devices on the bus must be pulled up to only 3V.

A device performing data transfers on the bus can be considered as a master or a slave. Each master and slave device can either be a transmitter (send data) or a receiver (receive data), and the communication on the bus is always initiated by the master by providing a valid start condition and the SCL signal.

Multiple master and slave devices may be connected on the bus, but only one device may master the bus during a data transfer. Since more than one master may simultaneously attempt to take control of the bus and start a transmission, the I²C/SMBus protocol provides an arbitration mechanism which relies on the wired-AND connection of all devices to the bus. A master device that generates a logic high on the SDA bus loses arbitration to a master that generates a logic low on the data bus. The MSP430 master transmitter that loses arbitration switches to slave receiver mode and sets the arbitration lost flag, ALIFG [1]. Each device on the bus has a unique 7-bit address, which allows a total of 128 devices to be connected on the bus. Some addresses are dedicated SMBus addresses which are reserved and must not be assigned to any SMBus device. For instance the SMBus Alert response address (0001 100b) [2].

1.2 SMBus Protocols

The different communication protocols can be found in the System Management Bus specification [2]. The communication always begins with a valid start condition from the master followed by a 7-bit slave address and the read/write bit which defines the master as a receiver /transmitter respectively, except in the quick command protocol. In quick command protocol, the read/write bit is used to turn a device on/off or enable/disable a low power mode. The read/write bit is followed by an Acknowledge from the slave. This is followed by 8-bit transfers which may be data, command or Packet Error Check (PEC). An acknowledge is sent by the receiver after each byte is received. To end the transfer, a valid stop condition is initiated by the master.

The SMBus standard introduced the Packet Error Checking (PEC) mechanism to improve communication reliability. The PEC is a CRC-8 error check byte, calculated on all message bytes except the ACK, NACK, START and STOP bits. The PEC is added to the message by the transmitter. The PEC in this application report is calculated using a cyclic redundancy check (CRC-8) polynomial, $C(x) = x^8 + x^2 + x^1 + 1$ and is calculated bit by bit in the order of bits received. Refer to SMBus specification for details on the PEC.

Another optional signal defined in the SMBus standard is the SMBALERT signal. This pin is also pulled up to V_{CC} through a resistor. A slave device can signal the master through SMBALERT requesting communication with the master. The master acknowledges such a slave device by sending the SMBus alert response address (0001 1001b) on the bus. The slave device acknowledges this Alert command by returning its 7-bit slave address on the bus and the ALERT signal becomes inactive. The eighth bit can be a '0' or '1'. If multiple devices pull the SMBALERT signal low, the lowest address device wins arbitration and its signal becomes inactive after the corresponding slave address byte is put on the bus.

The SMBus operating frequency range is 10 kHz to 100 kHz. Since a minimum speed needs to be maintained in this communication, a slave can only hold SCL low for a specified amount of time before the master times out and issues a stop condition. The slave can hold the clock low for 25 ms before timeout occurs. After this time, the slave must be able to receive a new start condition within 35 ms. Additional timing information can be found on the SMBus specification site (<http://www.smbus.org>) [2].

1.3 Software

The firmware used to test this application is attached in a zip folder along with this application report. Code coverage for this application note is supported in C and assembly using IAR and Code Composer Essentials.

The firmware was tested using the MSP430F169 device and a MSP-FET430P140 target board as shown in Figure 1-1.

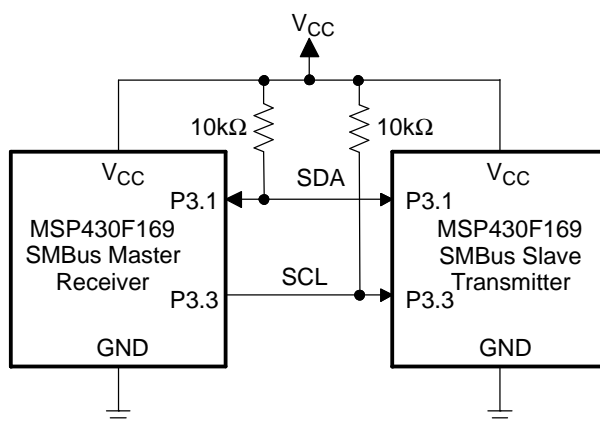


Figure 1-1. MSP430F169 Master-Slave SMBus Communication

1.4 Example 1

1.4.1 fet140_SMB_mstr.c(.s43)

The master is the receiver and detects a timeout due to the slave holding the clock line low for a period greater than “timeout”. TimerA0 is used to detect this timeout period and the master issues a stop condition at the conclusion of the byte transfer currently in progress.

The hardware I²C has the built-in I2CBB (Bus Busy) bit which is set after a start condition. I2CBB and I2CSCLOW may be used to determine how long SCL has been held low after a start condition. The corresponding slave code source file is fet140_SMB_slav.c.

1.4.2 fet140_SMB_slav.c(.s43)

This is the slave code source for the master code, fet140_SMB_mstr.c. TimerA0 is used to generate the delay between consecutive data bytes. This simulates a delay in data handling causing SCL to be held low during that period exercising the timeout features of the SMBus protocol. Data is transmitted inside the I2C_ISR.

1.5 Example 2

1.5.1 *fet140_SMB_mstr_slvrst.c(.s43)*

In this example the master device issues a start condition and waits in LPM0 for data reception.

1.5.2 *fet140_SMB_slav_slvrst.c(.s43)*

This is the slave code source used with *fet140_SMB_mstr_slvrst.c* (Master). The slave resets the communication port upon detecting a timeout. TimerA1 is used to detect the 25 ms timeout. TimerA2 is used to cause delay in data transfer to hold SCL low. TimerB0 is used as the capture input to detect SCL transitions to start/stop TimerA0 to detect timeout. SCL (Pin 31) is tied to TB0 (Pin 36) to detect transitions.

1.6 Example 3

1.6.1 *fet140_SMB_mstr_PEC.c*

This is the master code source that reads one data byte and the corresponding PEC byte from the slave. It performs a PEC using a CRC-8 algorithm on the received bytes and slave address.

1.6.2 *fet140_SMB_slave_PEC.c*

This is the slave code source used with *fet140_SMB_mstr_PEC.c*. The example writes one byte of data and one byte of PEC from the slave. The PEC byte is calculated using a CRC-8 algorithm on the transmitted byte and slave address.

1.7 Example 4

A test is also performed on the MSP430F169 device and SMBus-compatible TMP175 device. Pull-up resistors (10k) are used on the SCL, SDA and ALERT lines, as shown in [Figure 1-2](#).

1.7.1 *fet140_SMB_tmp175.c*

This is the master code source used to communicate with the TMP175, digital temperature sensor [3]. The TMP175 is a SMBus-compatible slave device. This example sets up the TMP175 in 9-bit temperature mode with interrupt (TM=1) to test the SMBALERT function. The master reads the temperature from the TMP175. Since the TMP175 is in the interrupt mode, the ALERT pin becomes active when the temperature equals or exceeds T(high) or equals or falls below T(low). The ALERT pin remains active until the device successfully responds to the SMBus alert response address. The alert pin is pulled-up to V_{CC} through a resistor. This pin is tied to P2.0 (Pin 20) of the MSP430F169. P2.0 is setup to detect a falling edge transition and respond with the SMBus Alert response command.

TimerA0 is setup to periodically have the master send out the start condition to request a new temperature reading from the TMP175.

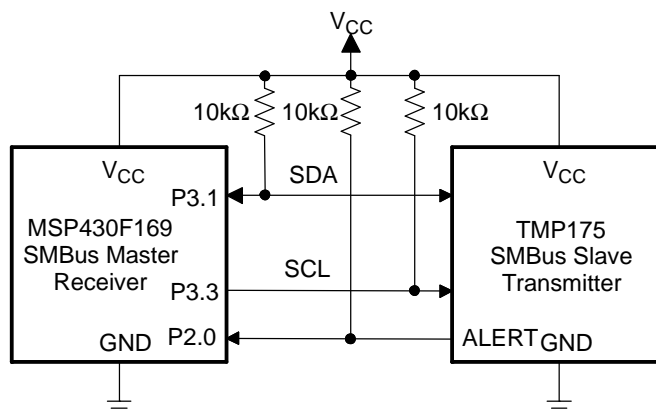


Figure 1-2. MSP430F169 Master-TMP175 Slave SMBus Communication

1.8 Conclusion

The hardware I²C peripheral and its built-in features help the MSP430 to easily adopt SMBus protocols. The ultra low power operation of the MSP430 is useful when interfacing with power management devices, such as the smart battery systems used in notebook computers, cameras, cellular phones or other portable electronic devices on a SMBus network.

1.9 References

1. MSP430x1xx Family User's Guide, Texas Instruments Inc, Literature Number SLAU049E, 2005
2. System Management Bus (SMBus) Specification, Version 2, Aug 2000
3. TMP175 device datasheet, Texas Instruments Inc, Literature Number SBOS288E, Dec 2004

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 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.

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.

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
DSP	dsp.ti.com	Broadband	www.ti.com/broadband
Interface	interface.ti.com	Digital Control	www.ti.com/digitalcontrol
Logic	logic.ti.com	Military	www.ti.com/military
Power Mgmt	power.ti.com	Optical Networking	www.ti.com/opticalnetwork
Microcontrollers	microcontroller.ti.com	Security	www.ti.com/security
		Telephony	www.ti.com/telephony
		Video & Imaging	www.ti.com/video
		Wireless	www.ti.com/wireless

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