

Using C to Interface 8051 MCUs with I²C™ Serial EEPROMs

Author: Alexandru Valeanu
Microchip Technology Inc.

INTRODUCTION

The 24XXX series serial EEPROMs from Microchip Technology support a bidirectional, 2-wire bus and data transmission protocol. The bus is controlled by the microcontroller (master), which generates the Serial Clock (SCL), controls the bus access and generates the Start and Stop conditions, while the 24XXX serial EEPROM works as slave. The 24XXX serial EEPROMs are I²C™ compatible and have maximum clock frequencies ranging from 100 kHz to 1 MHz.

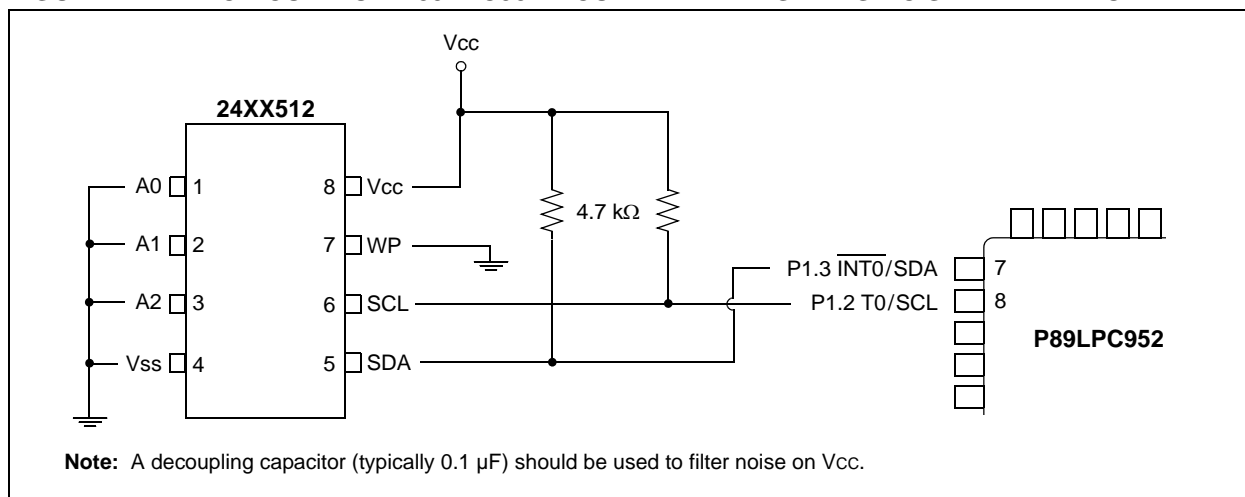
The main features of the 24XXX serial EEPROMs are:

- 2-wire serial interface bus, I²C compatible
- EEPROM densities from 128 bits to 512 Kbits
- Bus speed from 100 kHz to 1 MHz
- Voltage range from 1.7V to 5.5V
- Low power operation
- Temperature range from -40°C to +125°C
- Over 1,000,000 erase/write cycles
- Up to 8 devices may be connected to same bus

This application note is part of a series that provide source code to help the user implement the protocol with minimal effort.

Figure 1 is the hardware schematic depicting the interface between the Microchip 24XXX series of I²C serial EEPROMs and NXP's P89LPC952 8051-based MCU. The schematic shows the connections necessary between the MCU and the serial EEPROM as tested, as well as the required pull-up resistors on the clock line (SCL) and data line (SDA). Not illustrated in this application note are the write-protect feature and the cascading of multiple devices; thus, the WP pin and address pins A0, A1 and A2 are tied to V_{ss} (ground). The test software was written assuming these connections.

FIGURE 1: CIRCUIT FOR P89LPC952 MCU AND 24XXX SERIES I²C SERIAL EEPROM



FIRMWARE DESCRIPTION

Main Function

The purpose of the firmware is to show how to generate specific I²C bus transactions using the bidirectional SDA pin on the microcontroller. The focus is to provide the designer with a strong understanding of communication with the 24XXX series serial EEPROMs, thus allowing for more complex programs to be written in the future. The firmware was written in C for NXP's P89LPC952 MCU using the Keil™ μVision3® IDE and was developed on the Keil MCB950 evaluation board.

The main code demonstrates two different methods of accessing the I²C serial EEPROM: byte access and page access. The byte method accesses single bytes, where every data byte is preceded by three bytes of address: device address, MSB address, and LSB address. In the page access method, the MCU sends the address of the first byte, and the I²C serial EEPROM internally increments the address pointer for the next data byte.

The code was tested using the 24XX512 serial EEPROM. The EEPROM features 64K x 8 (512 Kbit) of memory and a page write capability of up to 128 bytes of data. Oscilloscope screen shots are shown in this application note. All timings are based on the internal RC oscillator of the MCU (7.373 MHz). If a faster clock is used, the code must be modified to generate the correct delays.

The bus speed in these examples is ~50 kHz.

I²C Functions

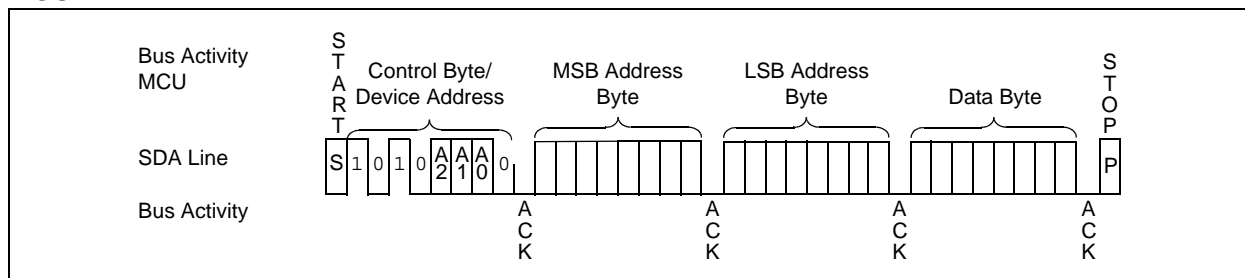
When an MCU accesses an I²C serial EEPROM, it is always the master on the I²C bus and the I²C serial EEPROM is the slave. The MCU controls all operations on the bus. Each operation is started by the MCU through a Start condition, followed by a control byte. The control byte consists of the control code (first 4 bits), the device address (next 3 bits), and the read/write (R/ \bar{W}) bit. The control code is always the same for the serial EEPROM being accessed, while the device address can range from '000' to '111', allowing up to eight different devices on the same bus. The R/ \bar{W} bit tells the serial EEPROM which operation to perform.

To access an I²C serial EEPROM at the start, the MCU writes the device address and the byte address to the I²C serial EEPROM; thus, each access cycle starts with a Write condition. For read operations, after the above sequence, the MCU switches from Transmitter mode to Receiver mode and the serial EEPROM from Receiver to Transmitter mode through a Restart condition.

BYTE WRITE OPERATION

Figure 2 depicts the necessary components that comprise the byte write operation. Each MCU's action is acknowledged (ACK) by the I²C serial EEPROM on the 9th bit of the clock by pulling down the SDA data line; consequently, every byte transfer lasts for 9 clock transitions.

FIGURE 2: BYTE WRITE

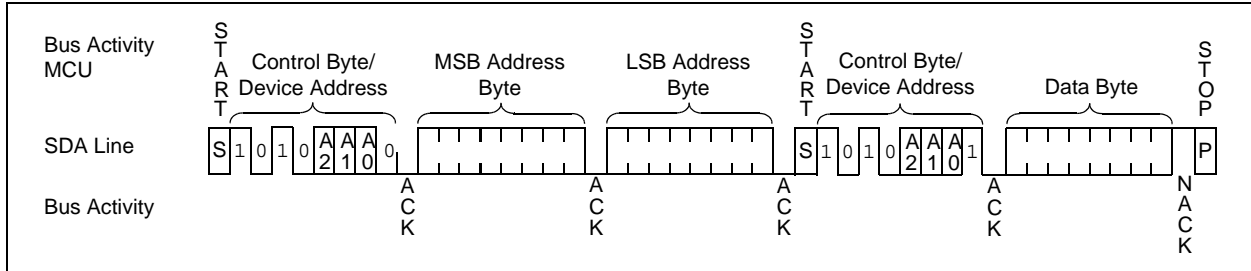


BYTE READ OPERATION

Figure 3 depicts the necessary components that comprise the byte read operation. The second Start condition instructs the I²C serial EEPROM to place data on the I²C bus.

The SDA line must remain stable while the SCL clock line is high. Any change of the SDA line while the SCL line is high is interpreted by the I²C serial EEPROM as a Start or Stop condition.

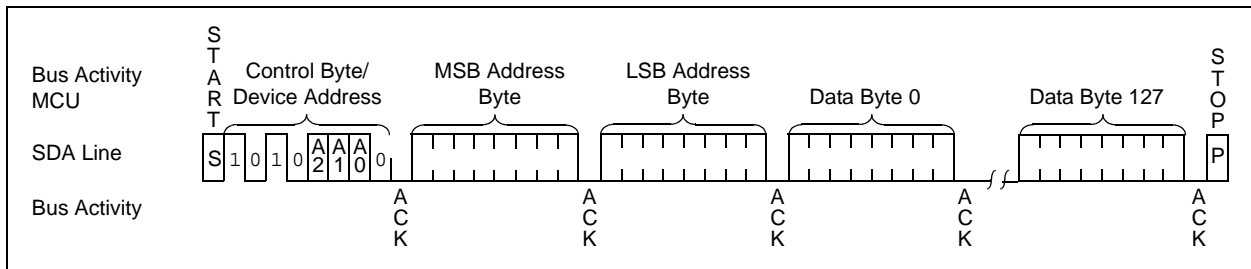
FIGURE 3: BYTE READ



PAGE WRITE OPERATION

Figure 4 depicts the necessary components that comprise the page write operation. The only difference between the page write operation and the byte write operation (Figure 2) is that the MCU, instead of sending 1 byte, sends 'n' bytes of data, up to the maximum page size of the I²C serial EEPROM.

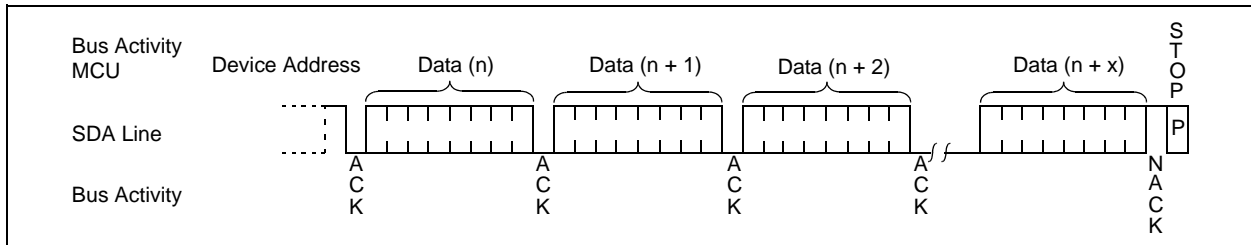
FIGURE 4: PAGE WRITE



SEQUENTIAL READ OPERATION

Figure 5 depicts the necessary components that comprise the sequential read operation. The last read byte is not acknowledged (NACK) by the MCU. This terminates the sequential read operation.

FIGURE 5: SEQUENTIAL READ



INITIALIZATION

The initialization function consists of initializing the SDA and SCL pins, setting them to the correct state, and configuring the pins.

After initialization, the MCU does the following:

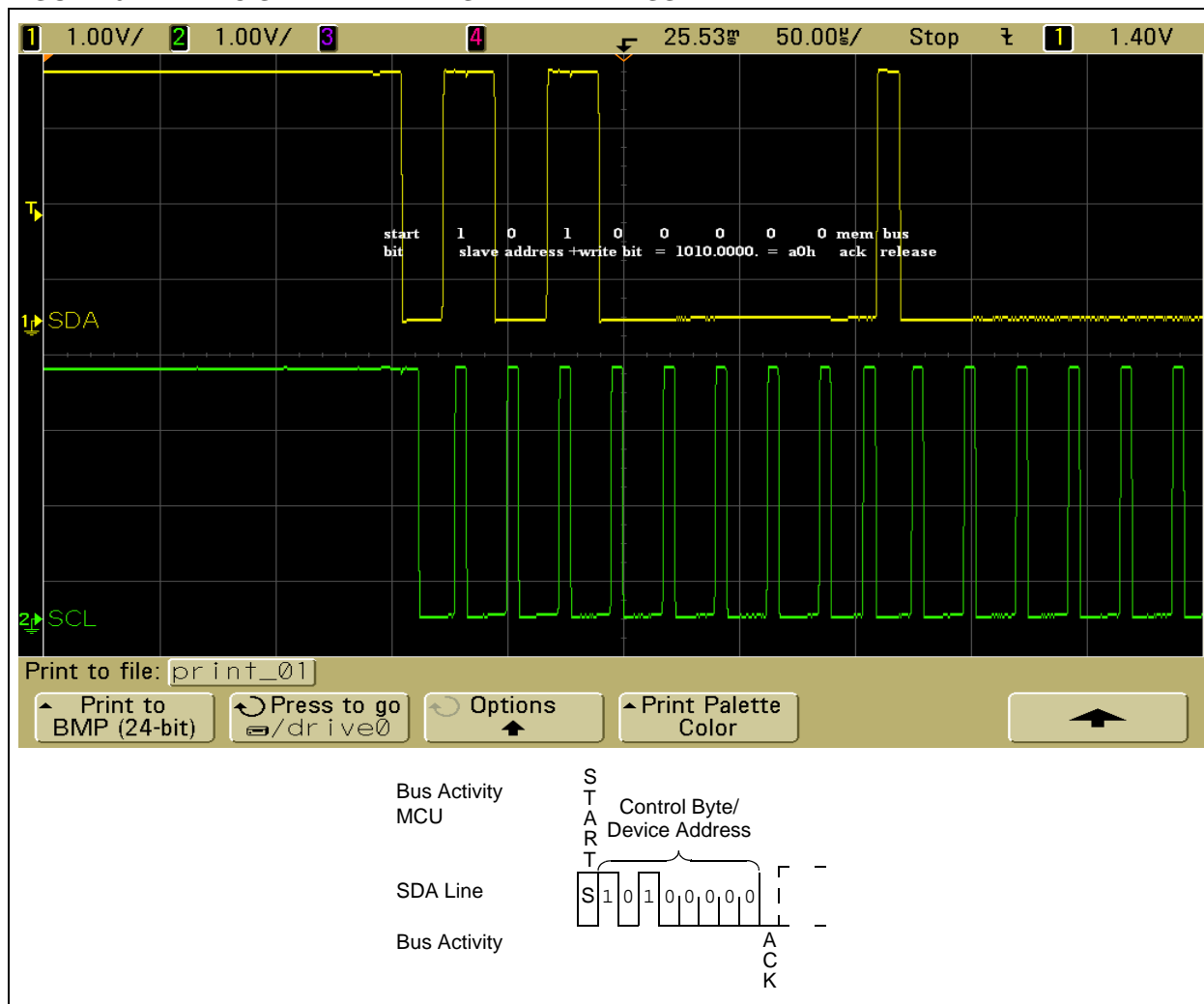
- Writes the 16-byte string `C_I2C_BB_VFLEDTX` in the I²C serial EEPROM (addresses = [0040h – 004Fh]).
- Reads back these addresses in the I²C serial EEPROM.
- Compares the read string with the written string.
- *Displays the hex values of the 16 read-back characters on the eight LEDs on the Keil evaluation board at a rate of 1 chr/sec.*
- *Sends the read characters to the UART in order to verify the operation.*

START DATA TRANSFER

The MCU generates a Start condition on the I²C bus to initiate data transfer.

Figure 6 shows a typical scope plot from the beginning of a write operation. Any memory access begins with a Start condition. This is followed by the I²C serial EEPROM (slave) address (A0h). Because the R/W bit is set to '0', the next operation of the bus is a write.

FIGURE 6: I²C START BIT AND SLAVE ADDRESS

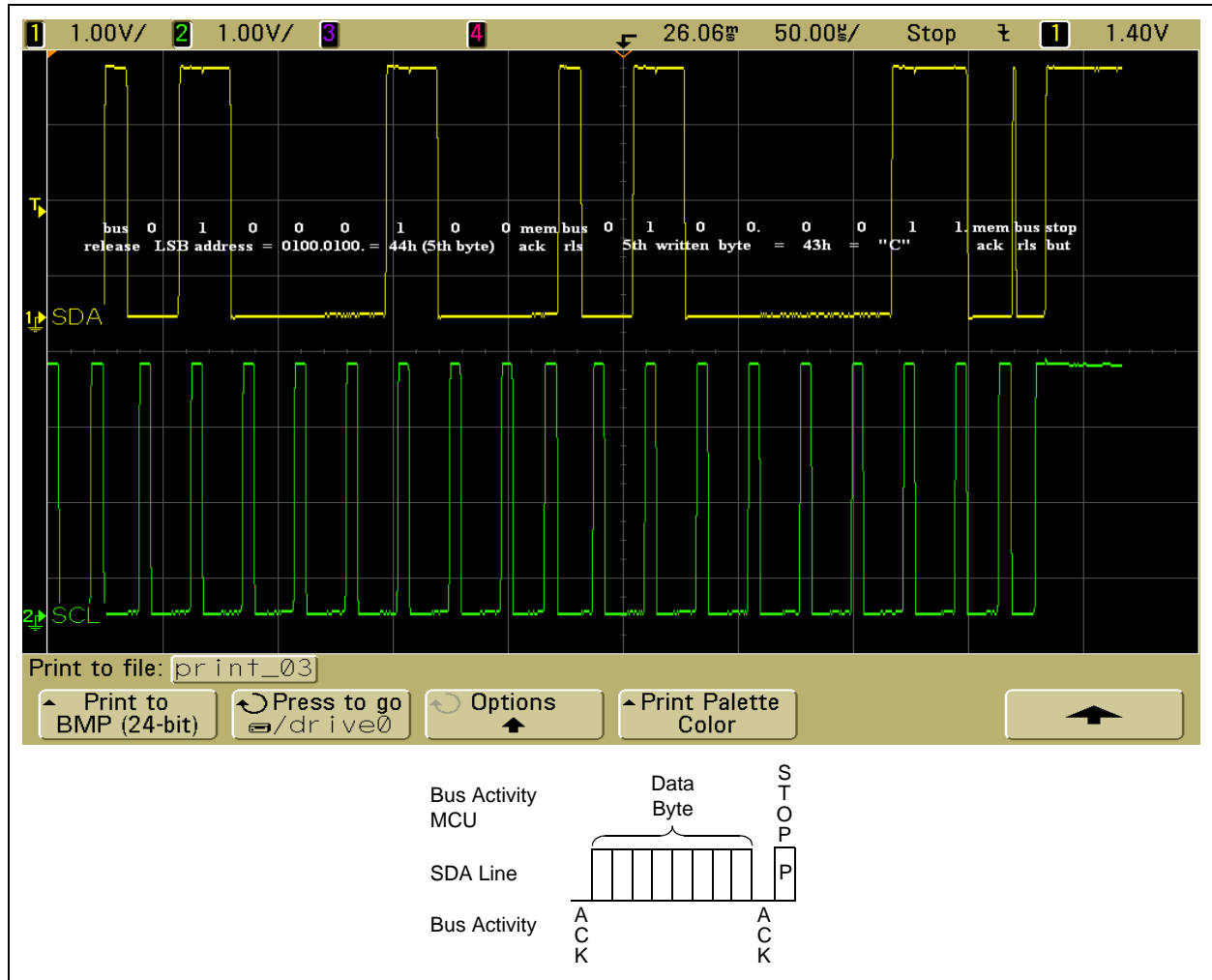


STOP DATA TRANSFER

The MCU generates a Stop condition on the I²C bus to stop data transfer.

Figure 7 shows a typical scope plot of a byte write operation followed by a Stop condition. Every operation on the I²C bus ends with a Stop condition.

FIGURE 7: BYTE WRITE AND STOP CONDITION



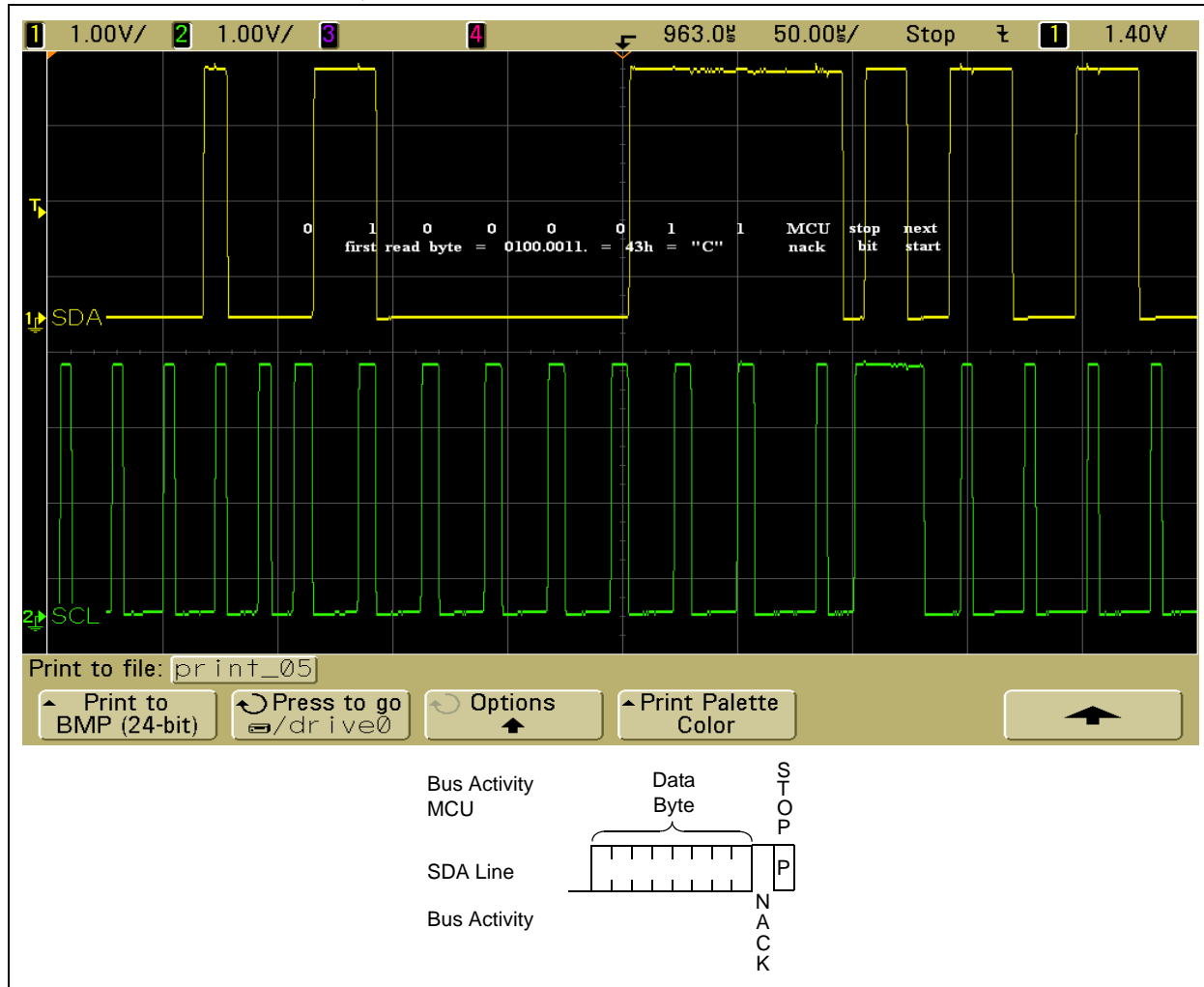
AN1195

NACK_MCU: NO ACKNOWLEDGE FROM MCU

The `nack_mcu` function is used at the end of a byte read sequence, but before the Stop condition, to indicate the last read byte. An Acknowledge or a No Acknowledge from the receiver to the transmitter is performed on the 9th bit of the clock.

Figure 8 shows a typical scope plot depicting the No Acknowledge condition from the MCU at the end of a byte read sequence.

FIGURE 8: BYTE READ, NACK_MCU AND STOP CONDITION

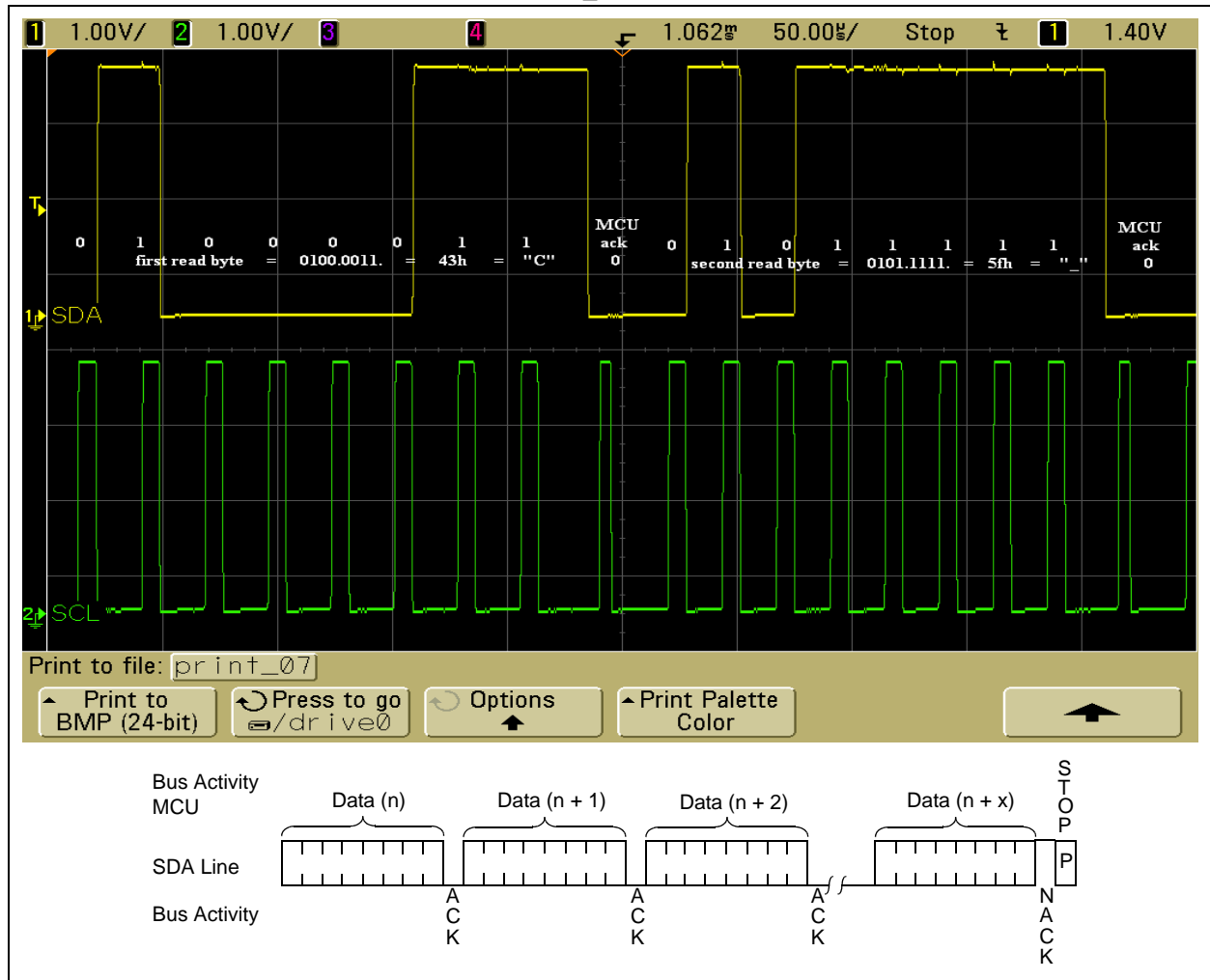


ACK_MCU: ACKNOWLEDGE FROM MCU

The `ack_mcu` function is used to acknowledge a byte or continue an operation. Only the last read byte will be not acknowledged (`nack_mcu`) by the MCU.

An MCU's acknowledgement is defined as a '0' on the 9th bit of the clock, as shown in Figure 9.

FIGURE 9: SEQUENTIAL READ AND ACK_MCU



WRITE 8 BITS

The 8-bits write function does all of the following:

- Sends data bytes or address bytes from the MCU to the I²C serial EEPROM.
- Shifts from parallel format to the serial I²C format.
- Receives an acknowledge from the I²C serial EEPROM on the 9th bit of the clock.

The MCU sets the data line on the falling edge of the clock, and the I²C serial EEPROM latches this in on the rising edge of the clock.

In Figure 6 a spike labeled "bus release" can be seen between the 9th clock pulse and the next clock pulse. The spike is the sign that both devices – the MCU and the I²C serial EEPROM – released the open-drain SDA line in order to be able to continue the communication.

READ 8 BITS OF DATA

This read function is used in both byte read and sequential read operations. The structure of the byte read operation is shown in Figure 3. The structure of the sequential read operation is shown in Figure 5.

During the read operation, the SDA pin must be programmed as input in order to receive the serial data from the I²C serial EEPROM. At the end of the function, the SDA must again be programmed as open-drain in order to generate the NACK and Stop.

For sequential read operations, the MCU acknowledges all but the last byte.

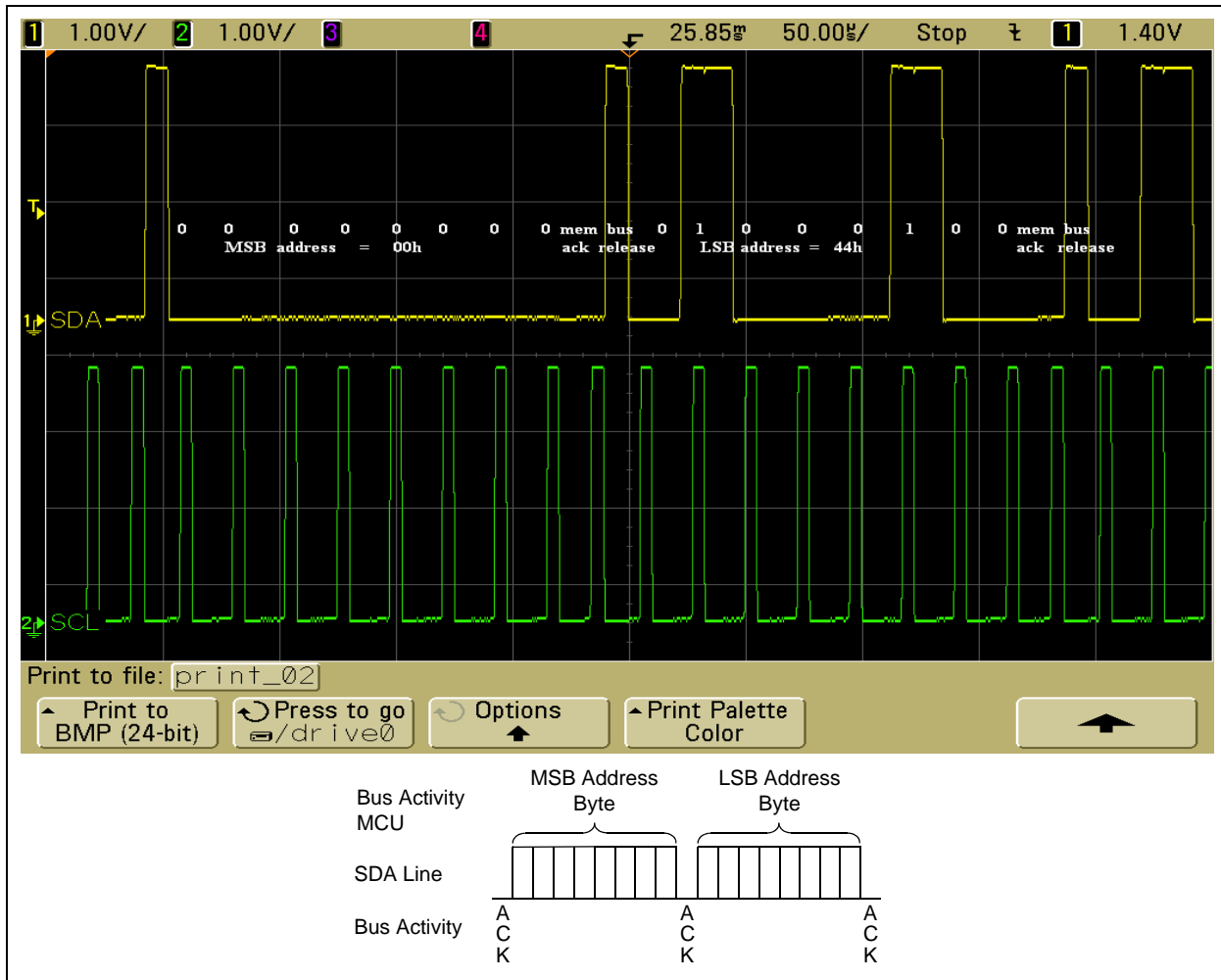
WRITE DATA BYTES

The structure of this byte write operation is shown in Figure 2.

The body of the function is a sequence of LCALL instructions preceded by loads of the EEPROM data buffer.

The start data transfer sequence is described in detail in the section entitled “Start Data Transfer” and in Figure 6. The stop data transfer sequence is described in detail in the section entitled “Stop Data Transfer” and in Figure 7. Figure 10 depicts the MSB address byte (00h) and the LSB address byte (44h).

FIGURE 10: WRITE MSB AND LSB ADDRESS BYTES



READ DATA BYTE

The read data byte function reads a data byte from the I²C serial EEPROM (slave). The structure of the byte read operation is shown in Figure 3.

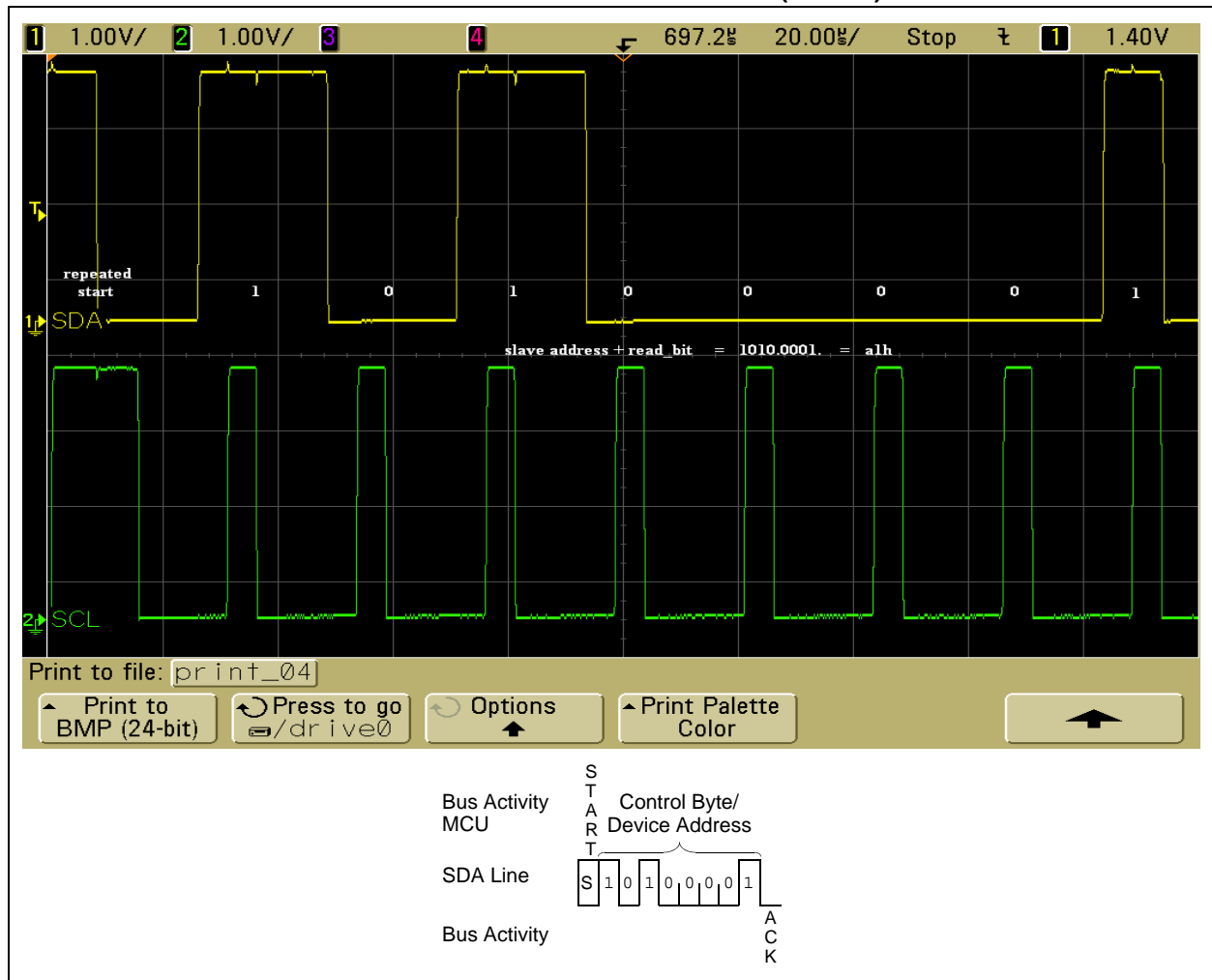
After the first Start condition, the MCU sends the device address, the MSB address byte, then the LSB address byte to the I²C serial EEPROM. Each of these bytes is acknowledged by the EEPROM.

Once the MCU has sent the address to the I²C serial EEPROM, it generates a Start condition (Repeated Start), which switches the I²C serial EEPROM from Receiver to Transmitter mode and the MCU from Transmitter to Receiver mode. Before the read, the MCU must send a new device address for a read.

The MCU must generate the necessary NACK or ACK conditions to terminate or continue the bus operation.

All the necessary scope plots have been presented in the previous paragraphs except the Repeated Start and the I²C serial EEPROM address read sequence, which is shown in Figure 11.

FIGURE 11: REPEATED START AND I²C SERIAL EEPROM (SLAVE) ADDRESS READ



WRITE A STRING (PAGE WRITE)

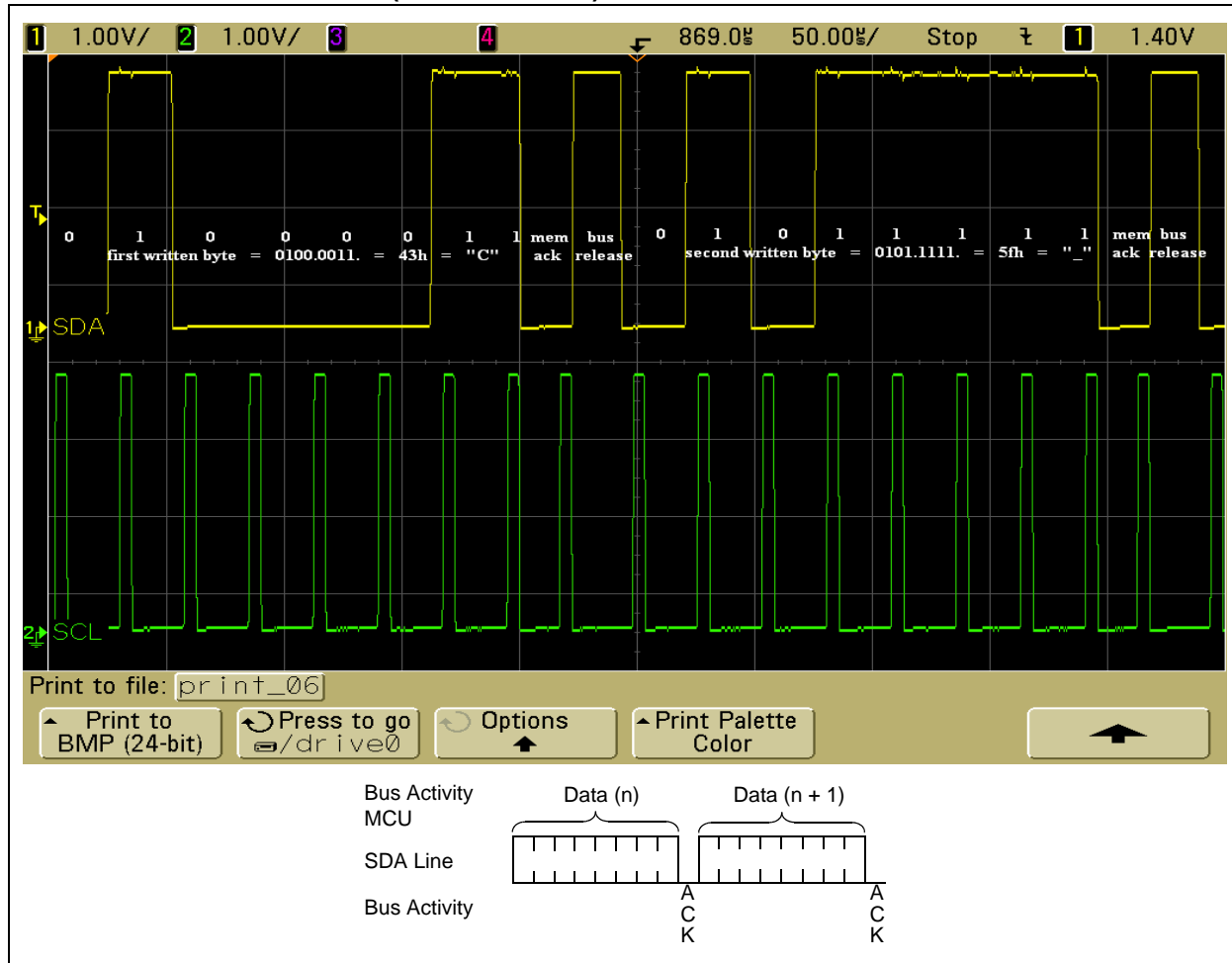
In this application note, the length of the string is 16 bytes and the physical page size for the 24XX512 is 128 bytes. The length of the written string must be shorter than the physical page size. If the page write operation overwrites the physical page boundary, the internal address counter rolls over and overwrites the first bytes of the current page.

The structure of the page write operation is shown in Figure 4.

The sequence must send the address of the first byte to be written. The address is automatically incremented at every byte write by the internal logic of the I²C serial EEPROM. Each received byte is acknowledged by the EEPROM.

All sequences have been described in the preceding paragraphs and related figures except for the structure of the page write operation, which is shown in Figure 12. The scope plot depicts the write of the first two consecutive bytes.

FIGURE 12: PAGE WRITE (FIRST 2 BYTES)



READ A STRING (SEQUENTIAL READ)

In contrast to the page write operation described in the previous paragraph, there is no maximum length for sequential read. After 64 Kbytes have been read, the internal address counter rolls over to the beginning of the array.

To indicate the end of the read, the MCU sends a NACK before the Stop condition. All other previously read bytes are acknowledged by the MCU.

The structure of the sequential read operation is shown in Figure 5. Figure 9 shows a typical scope plot depicting this operation.

BYTE WRITE VERSUS PAGE WRITE

At first glance, the page write method appears superior to the byte write method: it's simpler and faster. However, a careful analysis shows that the byte write method has a major advantage over page write owing to the roll-over phenomenon (see Note).

Note: Page write operations are limited to writing bytes within a single physical page, regardless of the number of bytes actually being written. Physical page boundaries start at addresses that are integer multiples of the page buffer size (or page size), and they end at addresses that are integer multiples of [page size-1]. If a `Page Write` command attempts to write across a physical page boundary, the result is that the data wraps around to the beginning of the current page (overwriting data previously stored there) instead of being written to the next page as might be expected. It is therefore necessary for the application software to prevent page write operations that would attempt to cross a page boundary.

As a consequence of the roll-over phenomenon, applications that write long strings to the I²C serial EEPROM risk overlapping the page boundary in the middle of a string. In such instances, the firmware should use byte write to avoid this condition. The disadvantage of doing this is the slower speed involved in writing the entire string: every byte write cycle time is approximately 5 ms.

The following summarizes the differences between the byte write and page write methods.

Byte Write

- Is slower – It needs a 5 ms write cycle time for each byte.
- Is more general – It may write a string of any length.

Page Write

- Is faster – It needs only one write cycle time for the whole page.
- Care must be taken to observe page boundaries during page writes.

CONCLUSION

This application note offers designers a set of I²C functions for reading and writing to an I²C serial EEPROM. All routines were written in C for an 8051-based MCU.

The code was developed on the Keil MCB950 evaluation board using the schematic shown in Figure 1. It was tested using the NXP P89LPC952 MCU and debugged using the Keil μ Vision3 IDE.

AN1195

NOTES:

Note the following details of the code protection feature on Microchip devices:

- Microchip products meet the specification contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is one of the most secure families of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the Microchip products in a manner outside the operating specifications contained in Microchip's Data Sheets. Most likely, the person doing so is engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as “unbreakable.”

Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip's code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

Information contained in this publication regarding device applications and the like is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. MICROCHIP MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION, INCLUDING BUT NOT LIMITED TO ITS CONDITION, QUALITY, PERFORMANCE, MERCHANTABILITY OR FITNESS FOR PURPOSE. Microchip disclaims all liability arising from this information and its use. Use of Microchip devices in life support and/or safety applications is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless Microchip from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights.

Trademarks

The Microchip name and logo, the Microchip logo, Accuron, dsPIC, KEELOQ, KEELOQ logo, MPLAB, PIC, PICmicro, PICSTART, PRO MATE, rPIC and SmartShunt are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

FilterLab, Linear Active Thermistor, MXDEV, MXLAB, SEEVAL, SmartSensor and The Embedded Control Solutions Company are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Analog-for-the-Digital Age, Application Maestro, CodeGuard, dsPICDEM, dsPICDEM.net, dsPICworks, dsSPEAK, ECAN, ECONOMONITOR, FanSense, In-Circuit Serial Programming, ICSP, ICEPIC, Mindi, MiWi, MPASM, MPLAB Certified logo, MPLIB, MPLINK, mTouch, PICkit, PICDEM, PICDEM.net, PICtail, PIC³² logo, PowerCal, PowerInfo, PowerMate, PowerTool, REAL ICE, rLAB, Select Mode, Total Endurance, UNI/O, WiperLock and ZENA are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

SQTP is a service mark of Microchip Technology Incorporated in the U.S.A.

All other trademarks mentioned herein are property of their respective companies.

© 2008, Microchip Technology Incorporated, Printed in the U.S.A., All Rights Reserved.

 Printed on recycled paper.

**QUALITY MANAGEMENT SYSTEM
CERTIFIED BY DNV
== ISO/TS 16949:2002 ==**

Microchip received ISO/TS-16949:2002 certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona; Gresham, Oregon and design centers in California and India. The Company's quality system processes and procedures are for its PIC[®] MCUs and dsPIC[®] DSCs, KEELOQ[®] code hopping devices, Serial EEPROMs, microperipherals, nonvolatile memory and analog products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001:2000 certified.



WORLDWIDE SALES AND SERVICE

AMERICAS

Corporate Office

2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 480-792-7200
Fax: 480-792-7277
Technical Support:
<http://support.microchip.com>
Web Address:
www.microchip.com

Atlanta

Duluth, GA
Tel: 678-957-9614
Fax: 678-957-1455

Boston

Westborough, MA
Tel: 774-760-0087
Fax: 774-760-0088

Chicago

Itasca, IL
Tel: 630-285-0071
Fax: 630-285-0075

Dallas

Addison, TX
Tel: 972-818-7423
Fax: 972-818-2924

Detroit

Farmington Hills, MI
Tel: 248-538-2250
Fax: 248-538-2260

Kokomo

Kokomo, IN
Tel: 765-864-8360
Fax: 765-864-8387

Los Angeles

Mission Viejo, CA
Tel: 949-462-9523
Fax: 949-462-9608

Santa Clara

Santa Clara, CA
Tel: 408-961-6444
Fax: 408-961-6445

Toronto

Mississauga, Ontario,
Canada
Tel: 905-673-0699
Fax: 905-673-6509

ASIA/PACIFIC

Asia Pacific Office

Suites 3707-14, 37th Floor
Tower 6, The Gateway
Harbour City, Kowloon
Hong Kong
Tel: 852-2401-1200
Fax: 852-2401-3431

Australia - Sydney

Tel: 61-2-9868-6733
Fax: 61-2-9868-6755

China - Beijing

Tel: 86-10-8528-2100
Fax: 86-10-8528-2104

China - Chengdu

Tel: 86-28-8665-5511
Fax: 86-28-8665-7889

China - Hong Kong SAR

Tel: 852-2401-1200
Fax: 852-2401-3431

China - Nanjing

Tel: 86-25-8473-2460
Fax: 86-25-8473-2470

China - Qingdao

Tel: 86-532-8502-7355
Fax: 86-532-8502-7205

China - Shanghai

Tel: 86-21-5407-5533
Fax: 86-21-5407-5066

China - Shenyang

Tel: 86-24-2334-2829
Fax: 86-24-2334-2393

China - Shenzhen

Tel: 86-755-8203-2660
Fax: 86-755-8203-1760

China - Wuhan

Tel: 86-27-5980-5300
Fax: 86-27-5980-5118

China - Xiamen

Tel: 86-592-2388138
Fax: 86-592-2388130

China - Xian

Tel: 86-29-8833-7252
Fax: 86-29-8833-7256

China - Zhuhai

Tel: 86-756-3210040
Fax: 86-756-3210049

ASIA/PACIFIC

India - Bangalore

Tel: 91-80-4182-8400
Fax: 91-80-4182-8422

India - New Delhi

Tel: 91-11-4160-8631
Fax: 91-11-4160-8632

India - Pune

Tel: 91-20-2566-1512
Fax: 91-20-2566-1513

Japan - Yokohama

Tel: 81-45-471- 6166
Fax: 81-45-471-6122

Korea - Daegu

Tel: 82-53-744-4301
Fax: 82-53-744-4302

Korea - Seoul

Tel: 82-2-554-7200
Fax: 82-2-558-5932 or
82-2-558-5934

Malaysia - Kuala Lumpur

Tel: 60-3-6201-9857
Fax: 60-3-6201-9859

Malaysia - Penang

Tel: 60-4-227-8870
Fax: 60-4-227-4068

Philippines - Manila

Tel: 63-2-634-9065
Fax: 63-2-634-9069

Singapore

Tel: 65-6334-8870
Fax: 65-6334-8850

Taiwan - Hsin Chu

Tel: 886-3-572-9526
Fax: 886-3-572-6459

Taiwan - Kaohsiung

Tel: 886-7-536-4818
Fax: 886-7-536-4803

Taiwan - Taipei

Tel: 886-2-2500-6610
Fax: 886-2-2508-0102

Thailand - Bangkok

Tel: 66-2-694-1351
Fax: 66-2-694-1350

EUROPE

Austria - Wels

Tel: 43-7242-2244-39
Fax: 43-7242-2244-393

Denmark - Copenhagen

Tel: 45-4450-2828
Fax: 45-4485-2829

France - Paris

Tel: 33-1-69-53-63-20
Fax: 33-1-69-30-90-79

Germany - Munich

Tel: 49-89-627-144-0
Fax: 49-89-627-144-44

Italy - Milan

Tel: 39-0331-742611
Fax: 39-0331-466781

Netherlands - Drunen

Tel: 31-416-690399
Fax: 31-416-690340

Spain - Madrid

Tel: 34-91-708-08-90
Fax: 34-91-708-08-91

UK - Wokingham

Tel: 44-118-921-5869
Fax: 44-118-921-5820