

## Using C and a Hardware Module to Interface 8051 MCUs with I<sup>2</sup>C™ 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 I<sup>2</sup>C™ compatible and have maximum clock frequencies ranging from 100 kHz to 1 MHz.

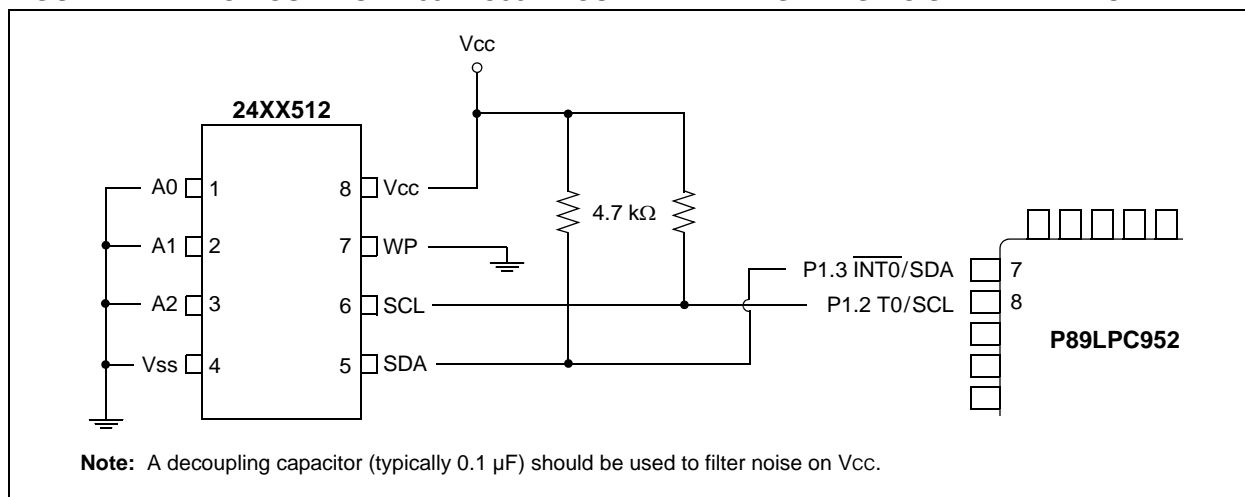
The main features of the 24XXX serial EEPROMs are:

- 2-wire serial interface bus, I<sup>2</sup>C 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 I<sup>2</sup>C 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<sub>ss</sub> (ground). The test software was written assuming these connections.

**FIGURE 1: CIRCUIT FOR P89LPC952 MCU AND 24XXX SERIES I<sup>2</sup>C SERIAL EEPROM**



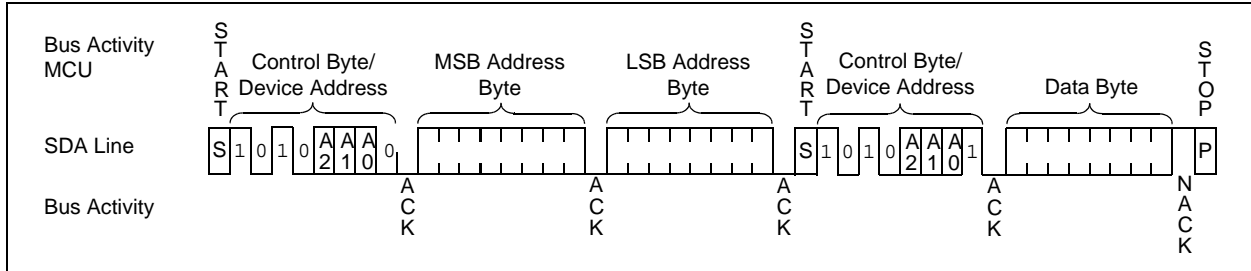


## BYTE READ OPERATION

Figure 3 depicts the necessary components that comprise the byte read operation. The second Start condition instructs the I<sup>2</sup>C serial EEPROM to place data on the I<sup>2</sup>C 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 I<sup>2</sup>C serial EEPROM as a Start or Stop condition.

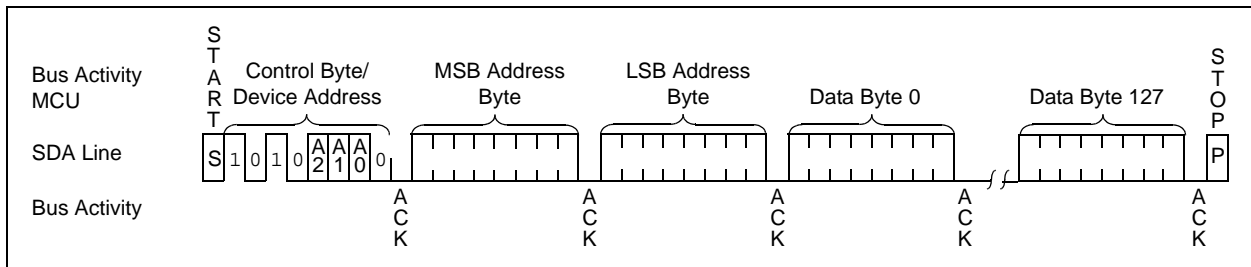
**FIGURE 3: BYTE READ**



## 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 I<sup>2</sup>C serial EEPROM.

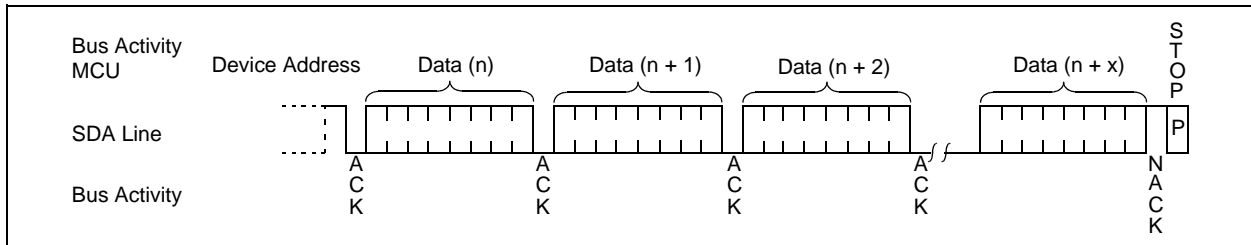
**FIGURE 4: PAGE WRITE**



## 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

Initialization consists of initializing the hardware peripheral within the MCU. It performs the following actions:

- I2EN = 1; //enable I<sup>2</sup>C part
- CRSEL = 0; //select internal baud generator
- STA = STO = SI = AA = 0; //clear Start, Stop, serial flag, ACK bit
- Sets the bus speed at ~75 kHz: I2SCLL = I2SCLH = 25

The bit frequency ( $f_{bit}$ ) will be:

### EQUATION 1: BIT RATE

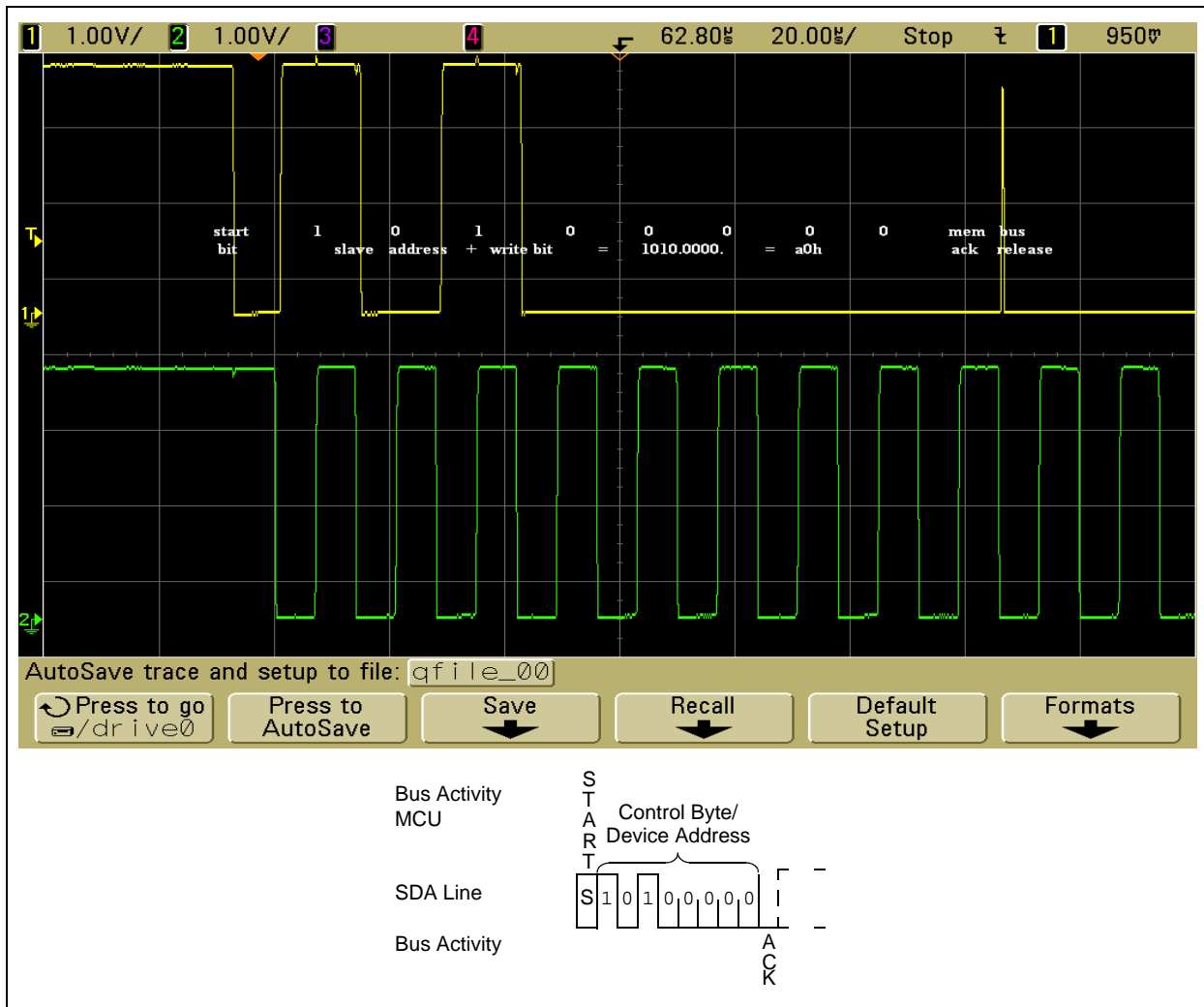
$$f_{bit} = \frac{fPCLK}{2 \times [I2SCLH + I2SCLL]}$$

## START DATA TRANSFER

The MCU generates a Start condition on the I<sup>2</sup>C 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 I<sup>2</sup>C 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: I<sup>2</sup>C START BIT AND SLAVE ADDRESS

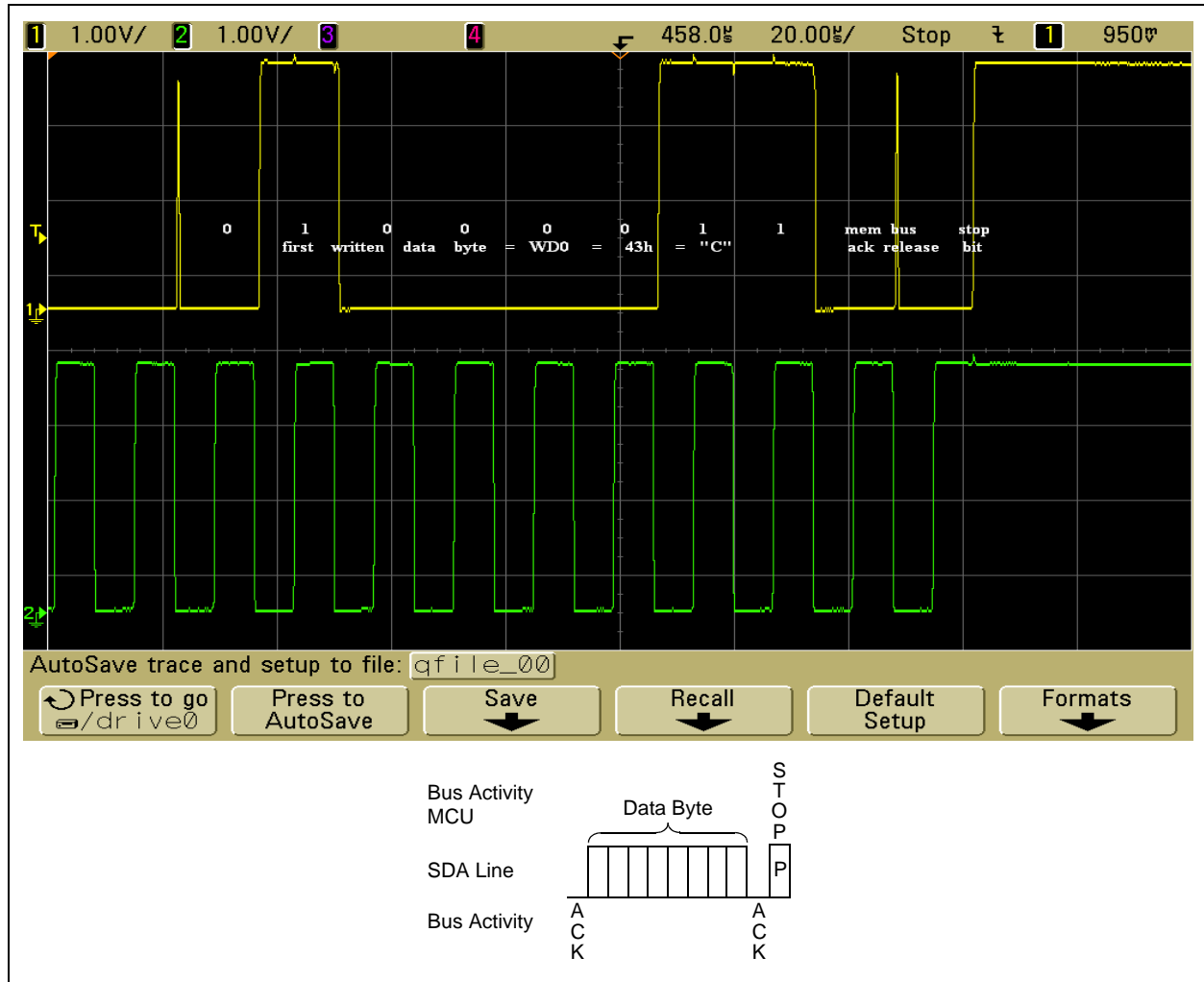


## STOP DATA TRANSFER

The MCU generates a Stop condition on the I<sup>2</sup>C 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 I<sup>2</sup>C bus ends with a Stop condition.

**FIGURE 7: BYTE WRITE AND STOP CONDITION**



## REPEATED START

The Repeated Start condition switches the MCU from Transmitter to Receiver mode and the serial EEPROM from Receiver to Transmitter. The bit is followed by the write of the I<sup>2</sup>C device address and R/W bit sequence.

The sequence is presented in Figure 8.

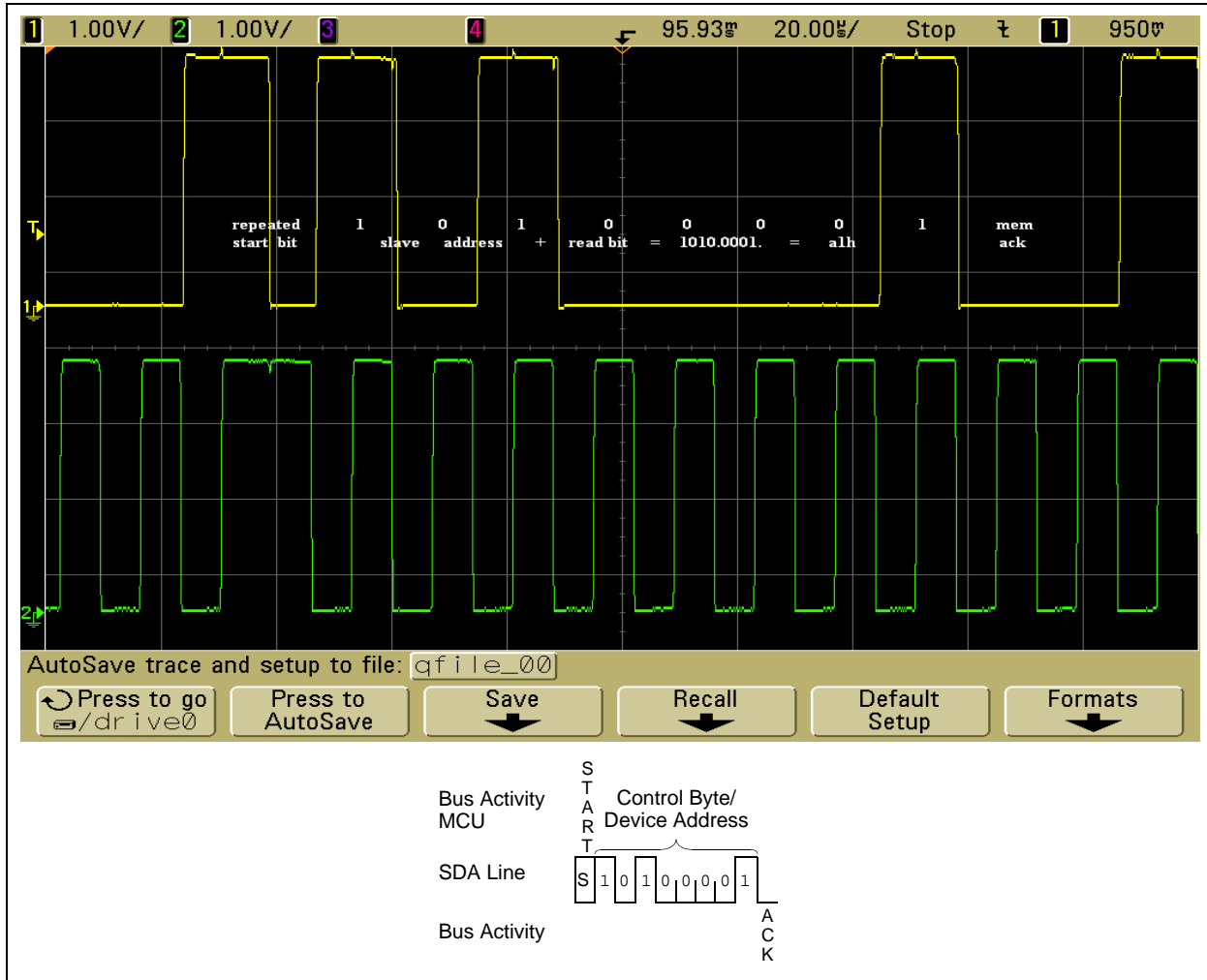
## WRITE I<sup>2</sup>C SERIAL EEPROM ADDRESS AND READ CONDITION

Following the Repeated Start condition, the MCU writes the I<sup>2</sup>C serial EEPROM address (slave address) with the R/W bit set correctly.

For read sequences, the I<sup>2</sup>C serial EEPROM address is A1h because the R/W bit is '1', indicating that the MCU waits to read bytes from the I<sup>2</sup>C serial EEPROM.

The sequence is presented in Figure 8.

**FIGURE 8: REPEATED START AND I<sup>2</sup>C SERIAL EEPROM (SLAVE) ADDRESS**



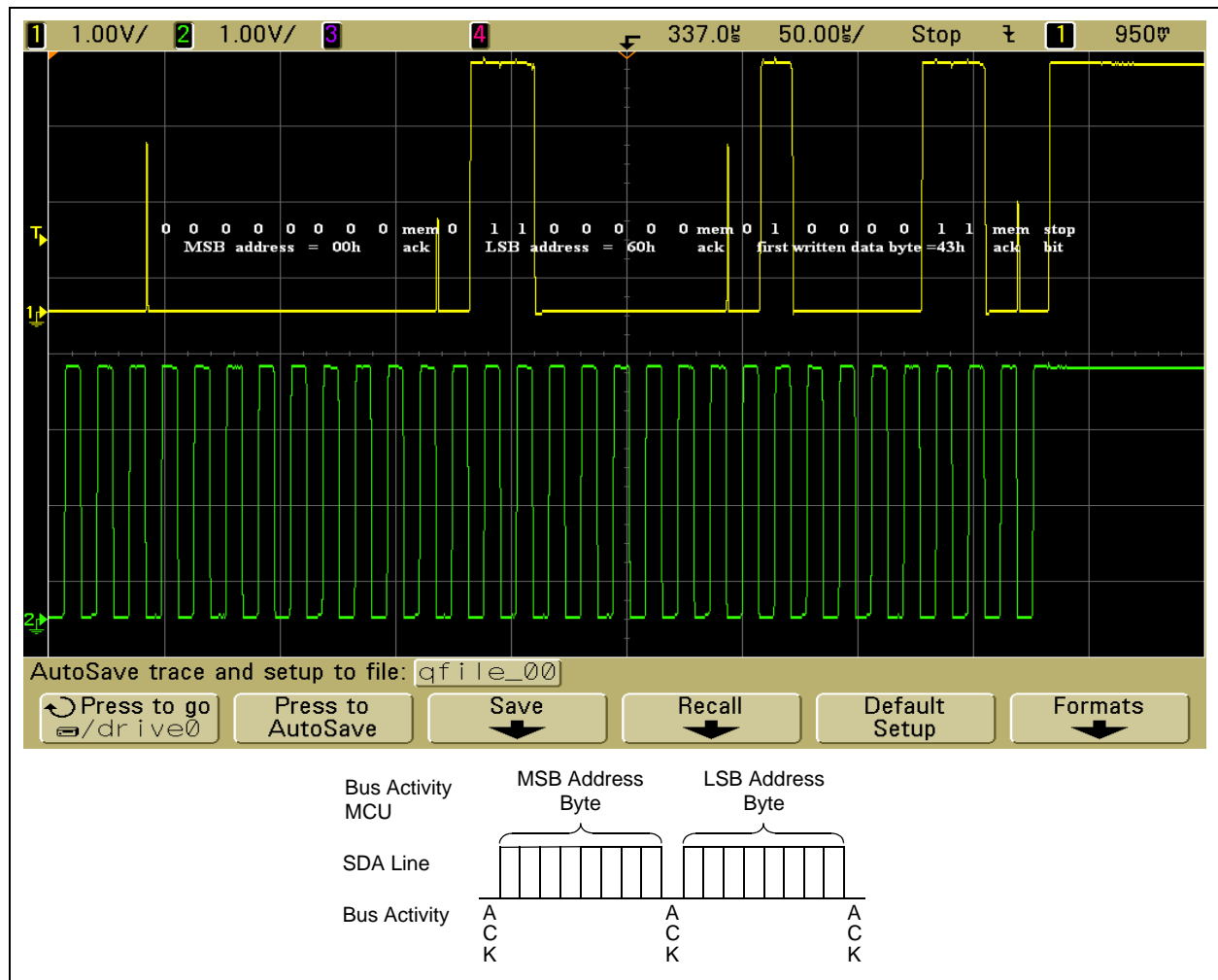
## WRITE DEVICE ADDRESS AND ADDRESS BYTES

After the Start condition, the MCU writes three bytes, consisting of the device address and the MSB and LSB addresses. Because the device address is followed by the write of the two address bytes, the device address has the  $R/\overline{W}$  bit set to '0'.

The scope plot in Figure 9 depicts the MSB address byte (00h), the LSB address byte (60h), followed by the first written byte (through the byte access method) of the string (43h). Because the byte access method accesses single bytes, the data byte is followed by a Stop bit.

The short spikes observed in the scope plot represent a short period of time during which both the MCU and the I<sup>2</sup>C serial EEPROM release the data bus.

**FIGURE 9: WRITE MSB AND LSB ADDRESS BYTES**



## WRITE 8 BITS OF DATA

The write data function is used in both byte write and page write operations. The structure of the byte write operation is shown in Figure 2. The structure of the page write operation is shown in Figure 4. The only difference between the two operations is that in the page write operation the MCU sends 'n' bytes of data instead of only one.

As a rule, the function shifts from a parallel format to the serial I<sup>2</sup>C format. The bus speed is in accordance with the initialization routine and is ~75 kHz.

The MCU sets the data line on the falling edge of the clock, and the I<sup>2</sup>C 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 I<sup>2</sup>C serial EEPROM – released the open-drain SDA line in order to be able to continue the communication.

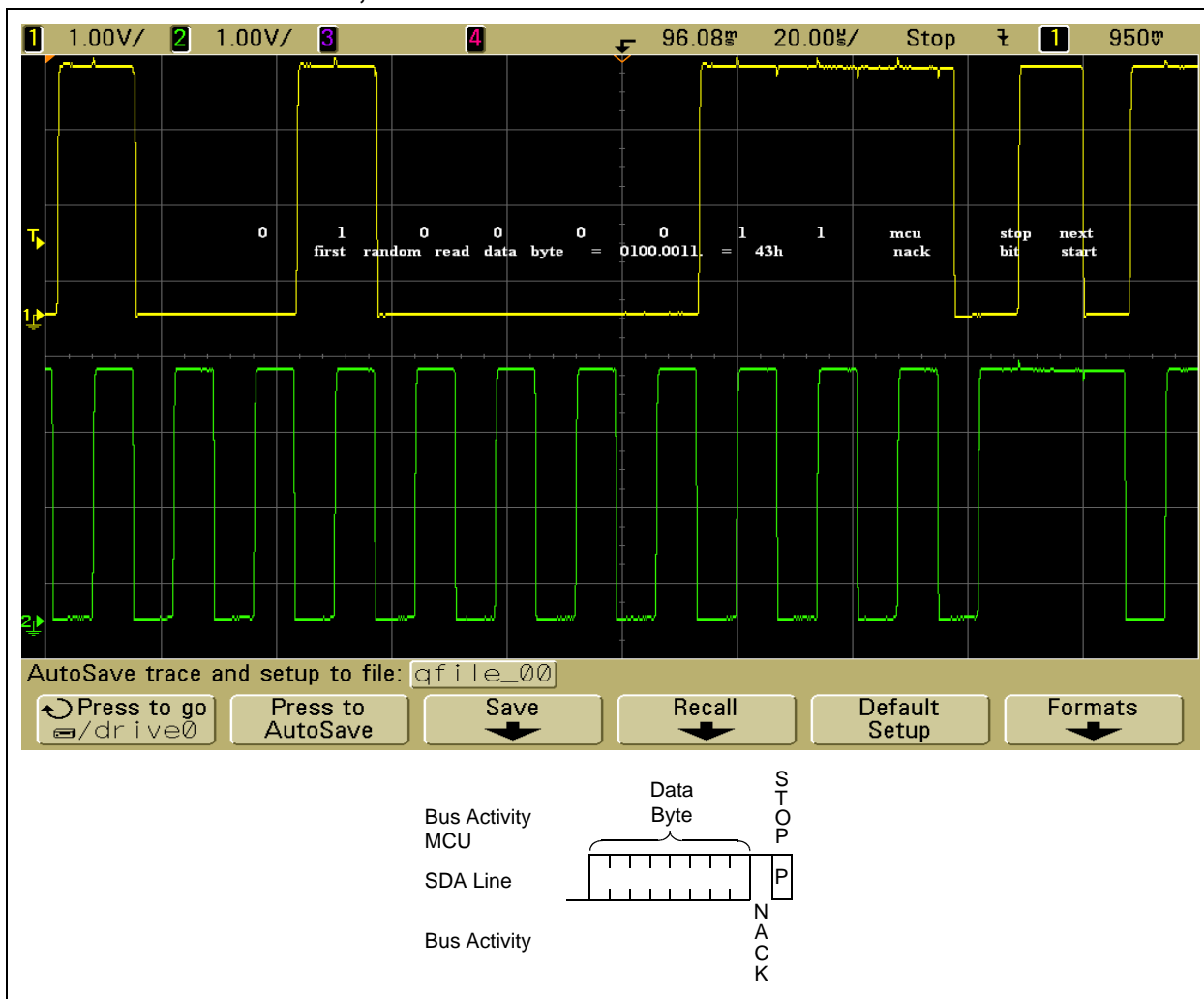
## READ 8 BITS OF DATA

The read data 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.

In the byte read operation, the 8-bit read data is situated at the end of the command and is not acknowledged by the MCU through a NACK on the 9th clock pulse of the read data byte, which is shown in Figure 10.

In the case of a sequential read operation, the last read byte will be not acknowledged and the other <n-1> read bytes will be acknowledged by the MCU.

**FIGURE 10: BYTE READ, NACK AND STOP CONDITION**





## WRITE DATA BYTE

The structure of this byte write operation is shown in Figure 2. Basically the function consists of a sequence of function calls. 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. The write MSB and LSB address byte sequence is described in the section entitled “**Write Device Address and Address Bytes**” and in Figure 9.

## READ DATA BYTE

The read data byte function reads a data byte from the I<sup>2</sup>C serial EEPROM. The structure of the 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 I<sup>2</sup>C serial EEPROM.

Once the entire memory address has been sent to the serial EEPROM, a new Start condition is generated to switch the MCU from Transmitter to Receiver mode and the serial EEPROM from Receiver to Transmitter 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.

## WRITE A STRING (PAGE WRITE)

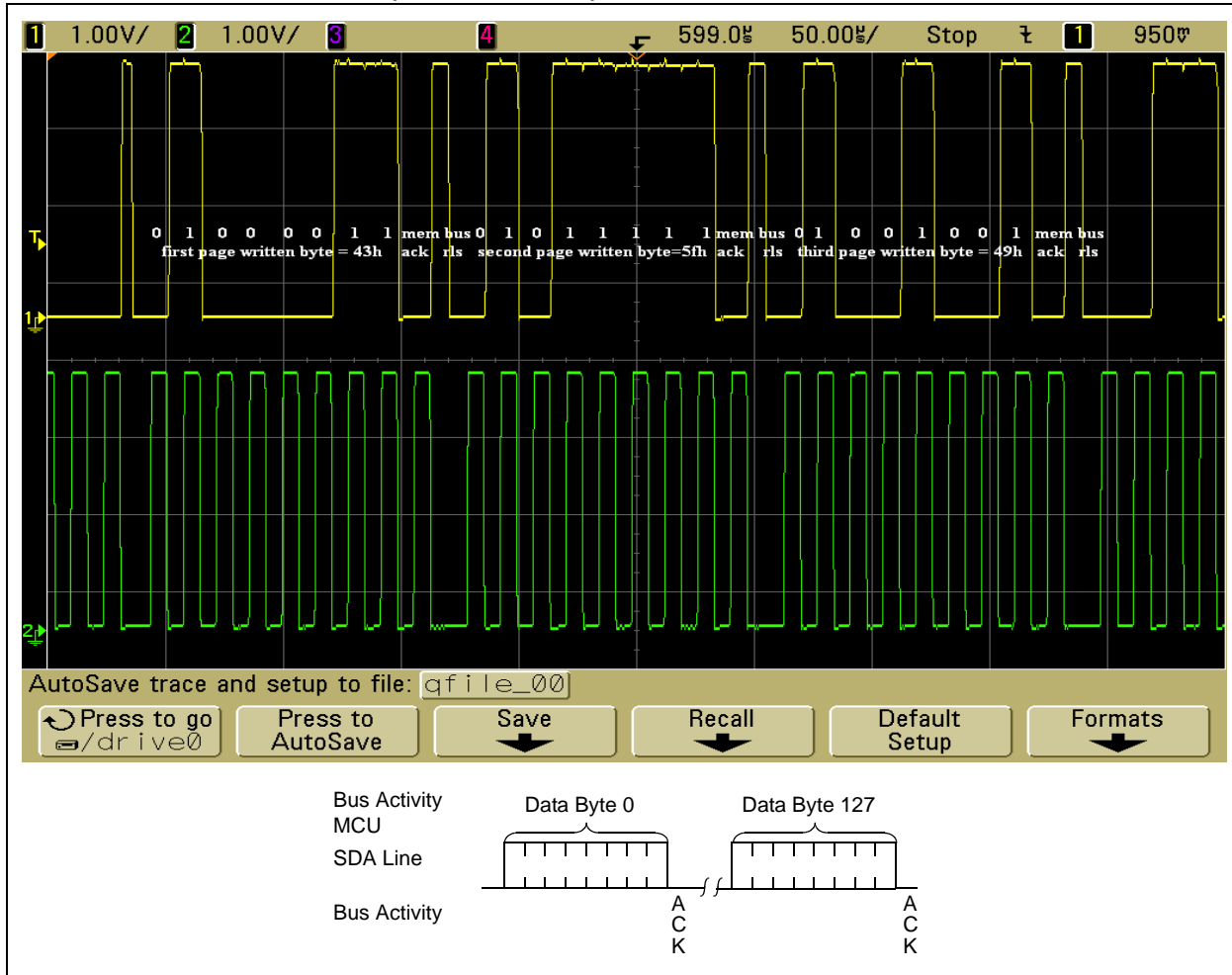
In this application note, the length of the string is 16 bytes and the physical page size for the 24XX512 is 128 bytes. The length of the written string must be shorter than the physical page size. If the page write operation overwrites the physical page boundary, the internal address counter rolls over 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 I<sup>2</sup>C serial EEPROM. Each received byte is acknowledged by the EEPROM.

The scope plot in Figure 11 depicts the write of the first three consecutive bytes.

**FIGURE 11: PAGE WRITE (FIRST 3 BYTES)**



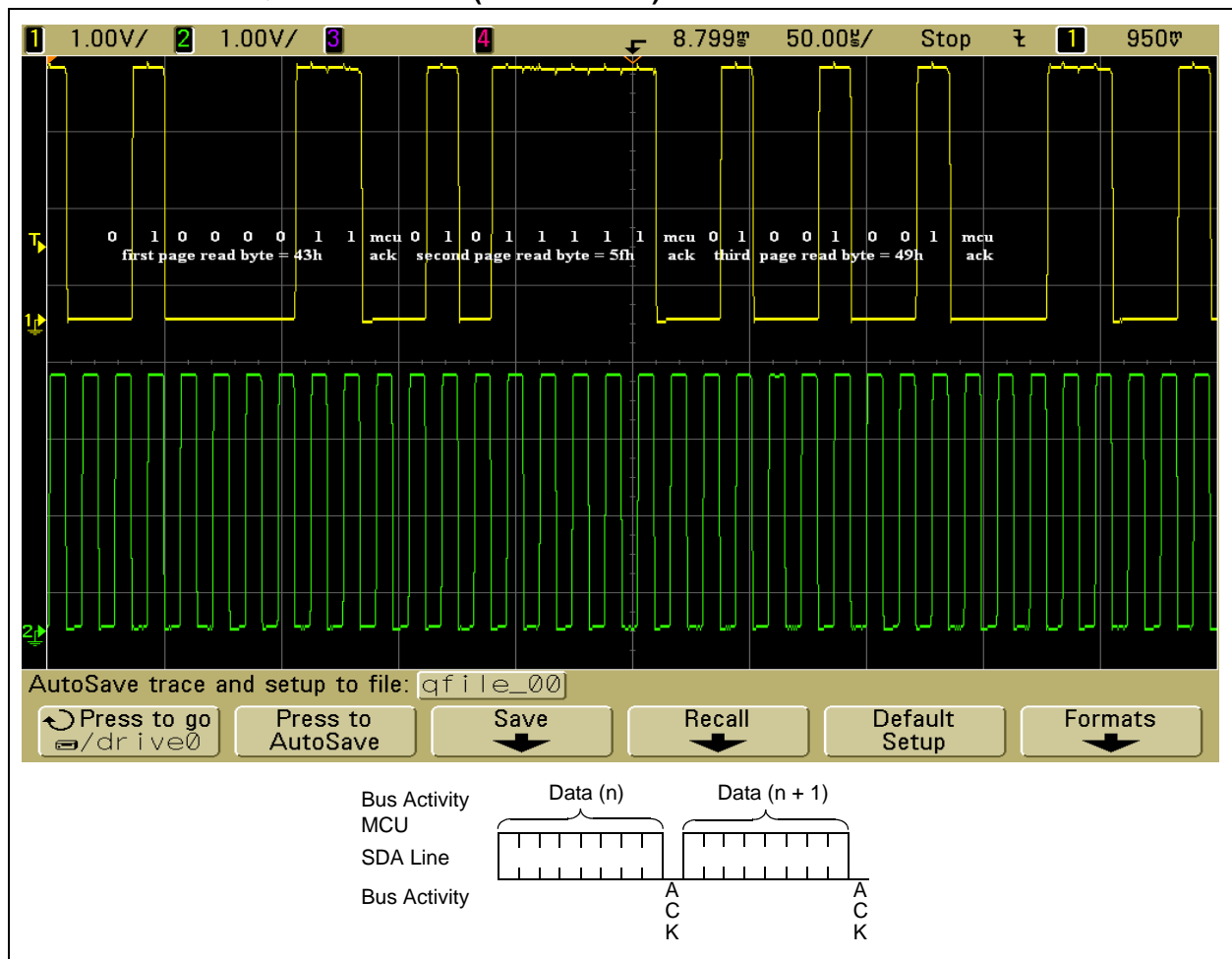
## READ A STRING (SEQUENTIAL READ)

In contrast to the page write operation described in the previous paragraph, there is no maximum length for a 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 12 shows a typical scope plot depicting this operation.

**FIGURE 12: SEQUENTIAL READ (FIRST BYTES)**



## 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 I<sup>2</sup>C 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 firmware routines to access I<sup>2</sup>C serial EEPROMs using a hardware peripheral. The code demonstrates byte and page operations. 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  $\mu$ Vision3 IDE.

---

**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, PIC<sup>32</sup> 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<sup>®</sup> MCUs and dsPIC<sup>®</sup> DSCs, KEELOQ<sup>®</sup> 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](http://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