MICROCHIP

AN1184

Using a Timer to Interface 8051 MCUs with UNI/O® Bus-Compatible Serial EEPROMs

Author: Alexandru Valeanu
Microchip Technology Inc.

INTRODUCTION

As embedded systems become smaller, a growing need exists to minimize I/O pin usage for communication between devices. Microchip has addressed this need by developing the UNI/O® bus, a low-cost, easy-to-implement solution requiring only a single I/O pin for bidirectional communication.

UNI/O bus-compatible serial EEPROMs can be used to enhance any application facing restrictions on available I/O. Such restrictions can potentially stem from connectors, board space or from the microcontroller itself.

The 11XXX family is the newest addition to Microchip Technology's broad serial EEPROM product line, and is compatible with the newly developed UNI/O bus.

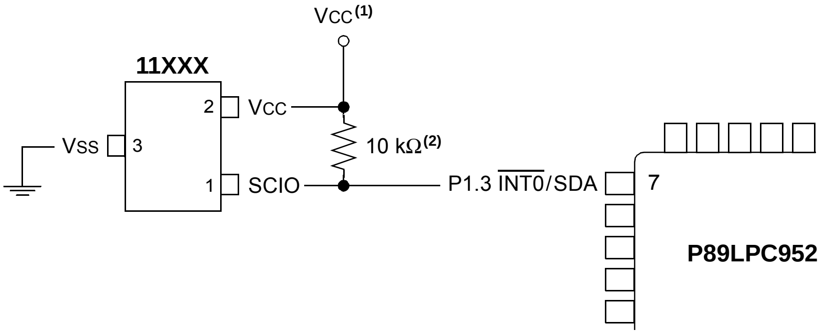
The main features of 11XXX serial EEPROMs are:

- Single I/O pin used for communication
- EEPROM densities from 1 Kbits to 16 Kbits
- Extremely small packages
- Bus speed from 10 kHz up to 100 kHz
- Voltage range from 1.8V to 5.5V
- Low-power operation
- Temperature range from -40°C to +125°C
- Over 1,000,000 erase/write cycles

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

Figure 1 describes the hardware schematic for the interface between the Microchip 11XXX series of UNI/O bus-compatible serial EEPROMs and NXP's P89LPC952 8051-based MCU. The schematic shows the connections necessary between the MCU and the serial EEPROM as tested. The software was written assuming these connections. The single I/O connection between the MCU and the serial EEPROM includes a recommended pull-up resistor. A decoupling capacitor across VCC and VSS is also recommended.

FIGURE 1: CIRCUIT FOR P89LPC952 MCU AND 11XXX SERIAL EEPROM



Note 1: A decoupling capacitor (typically 0.1 µF) should be used to filter noise on VCC.
2: A pull-up resistor (typically 10 kΩ) on SCIO is recommended to ensure bus idle during power-up.

FIRMWARE DESCRIPTION

The purpose of the firmware is to show how to generate specific UNI/O bus transactions using a generic I/O pin on the microcontroller. The focus is to provide the designer with a strong understanding of communication with the 11XXX series serial EEPROMs, thus allowing for more complex programs to be written in the future.

The firmware was written in the assembler language for the NXP P89LPC952 using the Keil™ µVision3® IDE and was developed on the Keil MCB950 evaluation board. The code can easily be modified to use any available I/O line.

The firmware consists of two files: the `main` file and the `inc` file. The `main` file is organized into the following sections:

- Initialization
- Write Enable
- Byte Write
- Write-in-Process Polling
- Byte Read
- Page Write

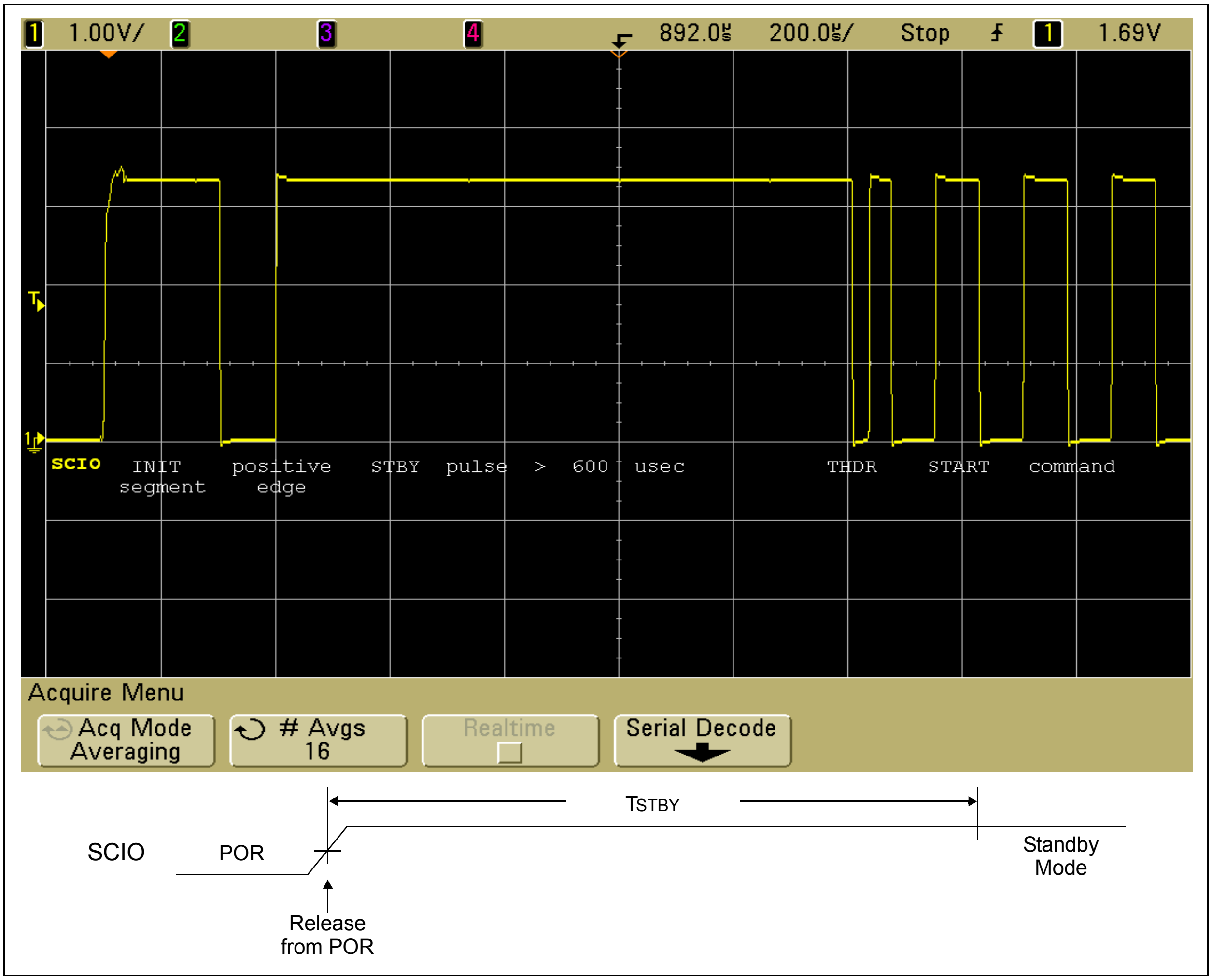
The code was tested using the 11XX160 serial EEPROM. The EEPROM features 2K x 8 (16 Kbit) of memory and 16-byte pages. Oscilloscope screen shots are shown in this application note. All timings are based on the internal RC oscillator of the MCU (7.373 MHz). If a faster clock is used, the code must be modified to generate the correct delays.

INITIALIZATION

Before initiating communication with the serial EEPROM, the MCU must generate a low-to-high edge on the SCIO to release the serial EEPROM from Power-on Reset (POR). Because bus idle is high, the MCU must create a high-low-high pulse on the SCIO. Once the serial EEPROM has been released from POR, a standby pulse with a minimum timing of TSTBY is performed to place the serial EEPROM into Standby mode, as shown in Figure 2.

Note that once a command has successfully executed – indicated by the reception of a Slave Acknowledgment (SAK) following the No Master Acknowledgment (NoMAK) – the serial EEPROM enters Standby mode immediately and a standby pulse is not necessary. In this case, only the start header setup time (TSS) must be observed before the MCU may initiate another command to the same serial EEPROM.

FIGURE 2: STANDBY PULSE



WRITE ENABLE

Before a write operation to the array or the STATUS register can occur, the Write Enable Latch (WEL) bit must be set. This is done by issuing a Write Enable (WREN) command.

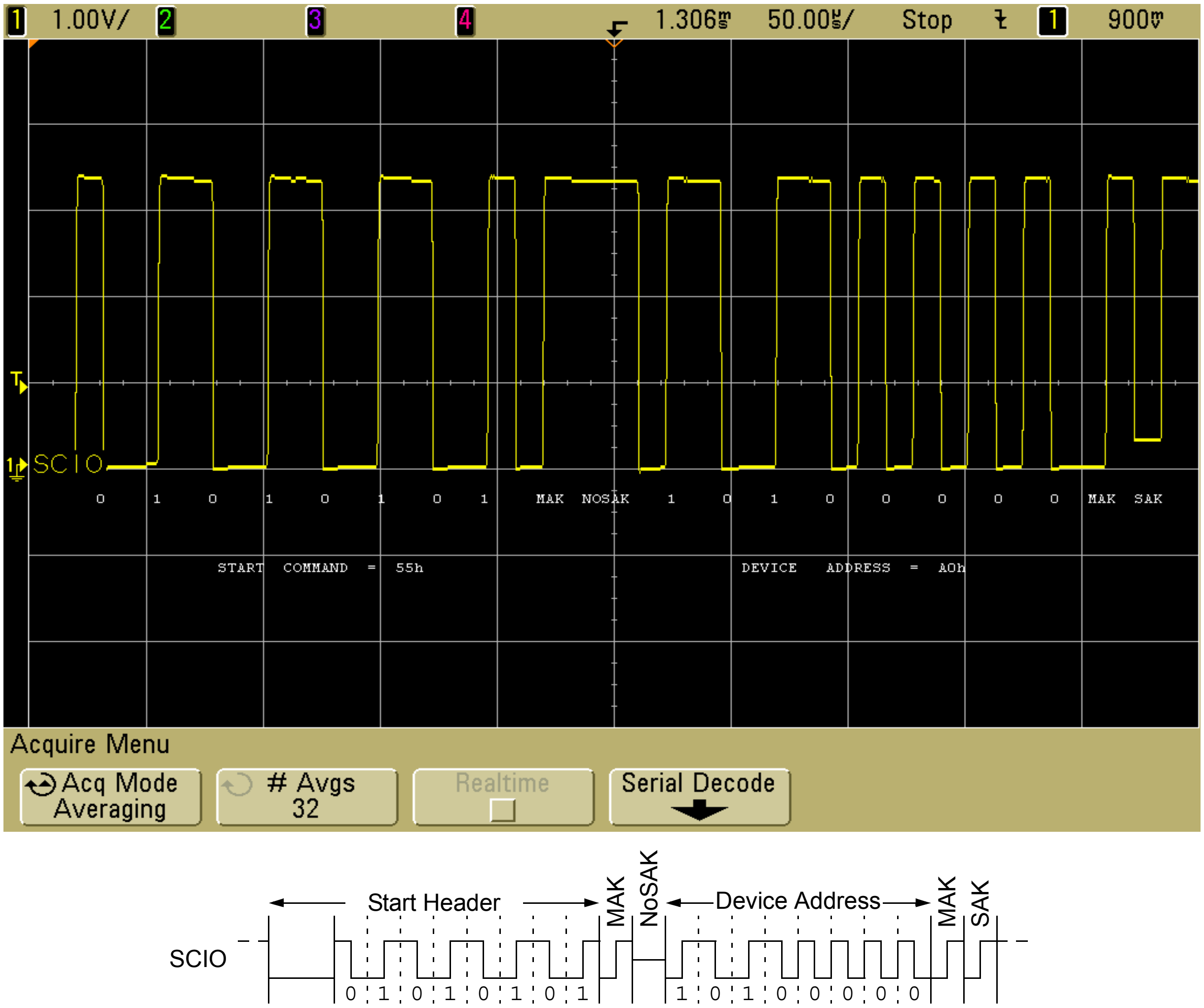
The WEL bit can be cleared by issuing a Write Disable (WRDI) command. It is also cleared upon termination of a write cycle to either the array or the STATUS register, and upon POR.

The Write Enable operation consists of the following components: the start header, which is followed by the device address and the command byte.

Start Header and Device Address

To issue a WREN command, the MCU transmits the start header. This consists of a low pulse (T_{HDR}) followed by '01010101', and a Master Acknowledge (MAK) followed by a NoSAK. Next, the MCU transmits the device address ('10100000') and another MAK. The serial EEPROM then responds with a SAK if the start header and device address were received correctly. Figure 3 shows the details of the start header and the device address.

FIGURE 3: START HEADER AND DEVICE ADDRESS

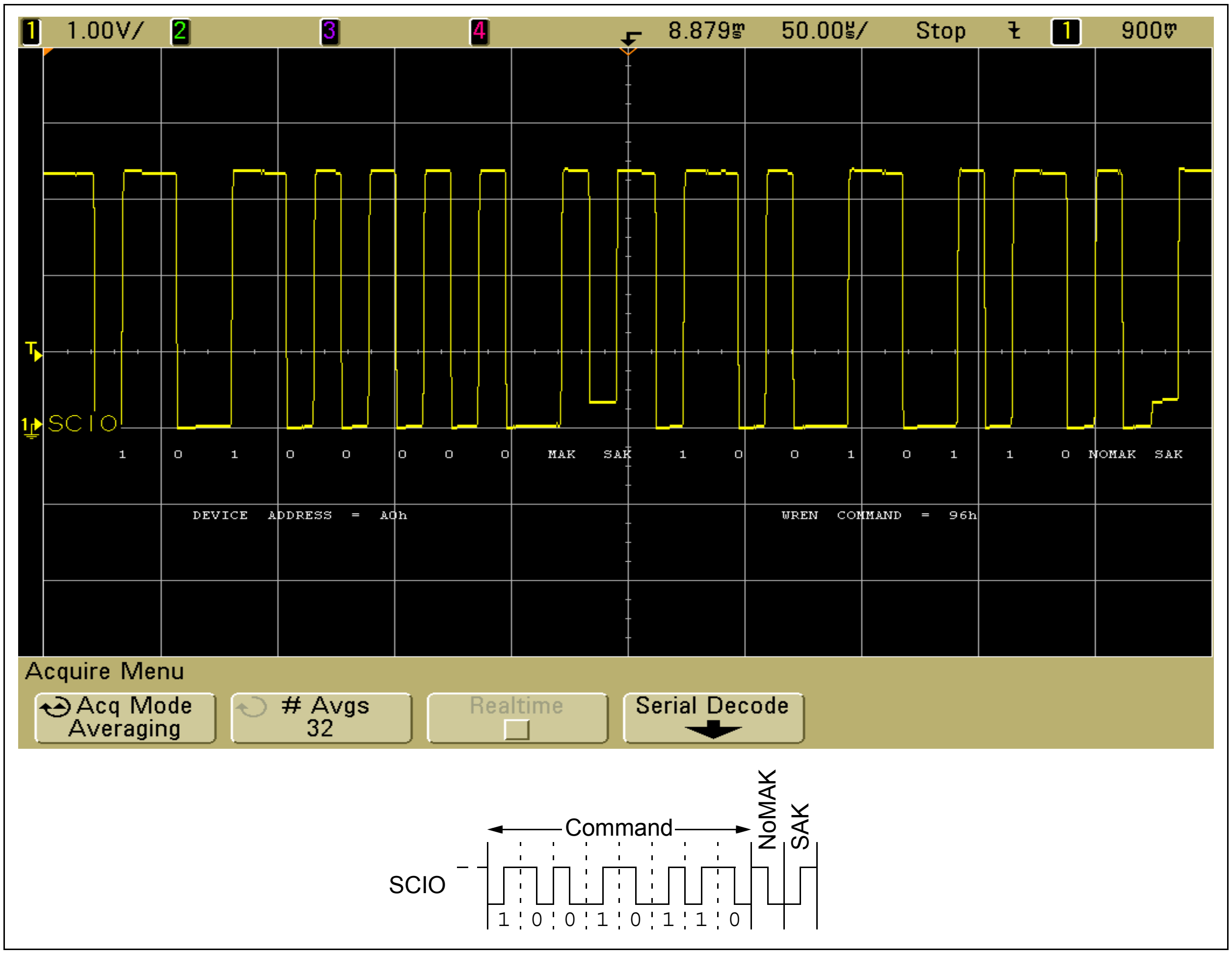


Write Enable (WREN) Command Byte

Once the SAK is received following the device address, the MCU sends the WREN command (‘`10010110`’ or 0x96) and performs a final Acknowledge sequence. During this last sequence, the MCU sends a NoMAK to signal the end of the operation. Once again, the serial EEPROM responds with a SAK, indicating it received the byte successfully.

Figure 4 shows an example of the WREN command.

FIGURE 4: WRITE ENABLE COMMAND



Acquire Menu

Acq Mode Averaging

Avgs 32

BYTE WRITE

The byte write operation consists of the following components: the Write command followed by the word address and data byte. Note that the start header and device address are not illustrated in this section but are still required to initiate the operation.

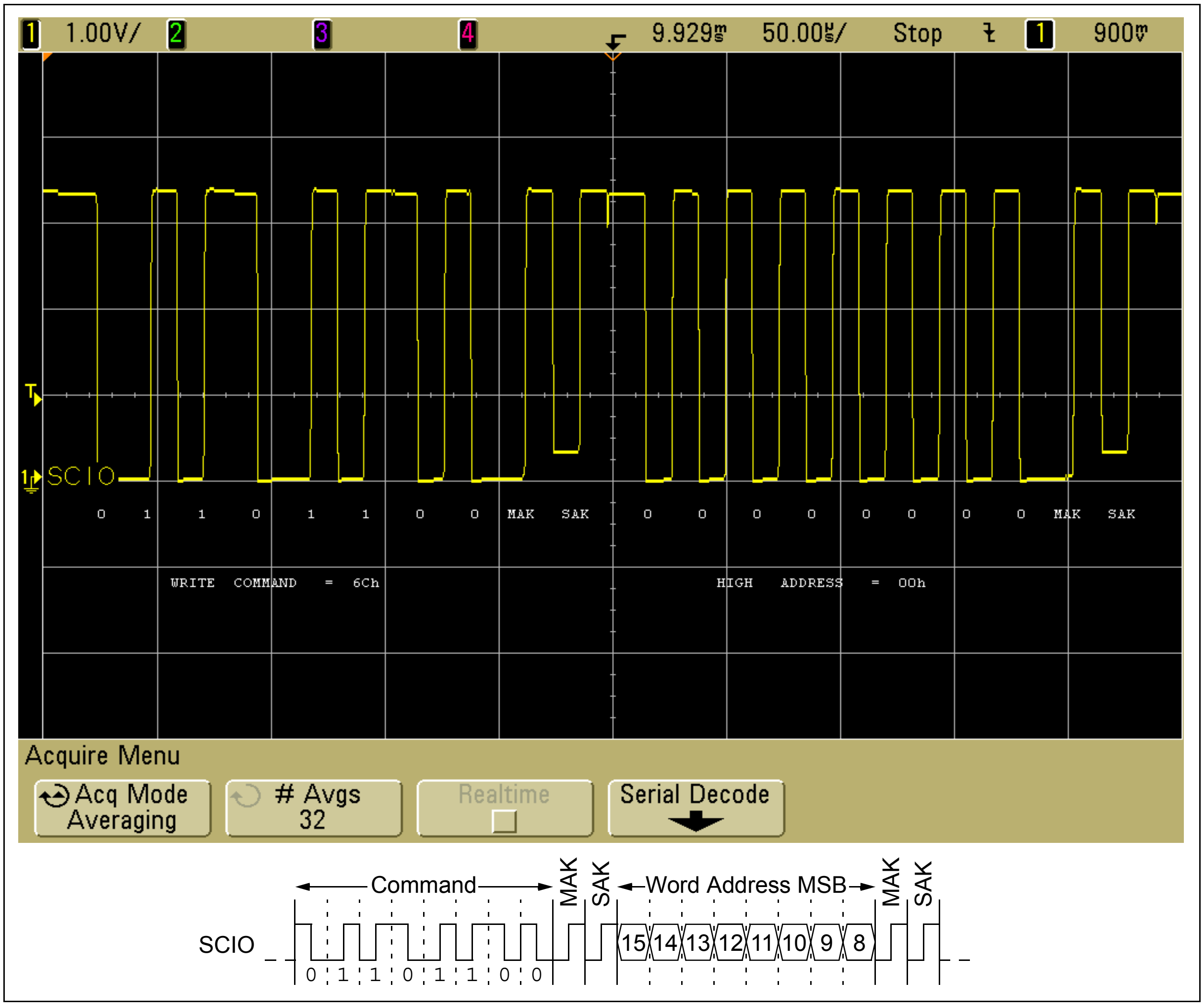
The acknowledge scheme is included as part of the provided functions but will be shown as part of the commands. Please consult the device data sheet for more information.

Sending the Write Command and Word Address

After the EEPROM device has acknowledged the start header and device address, the MCU sends the Write command, followed by the word address. The Write command is ‘`01101100`’ or 0x6C. The word address for the 11XX160 is a 16-bit value, so two bytes must be transmitted for the entire word address, with the Most Significant Byte sent first. After the command byte and the word address bytes have been sent, the MCU generates a MAK; the serial EEPROM responds with a SAK if there are no errors.

Figure 5 shows the command byte, the high address byte and the corresponding MAK/SAK. The low address byte is shown in Figure 6.

FIGURE 5: WRITE COMMAND AND WORD ADDRESS



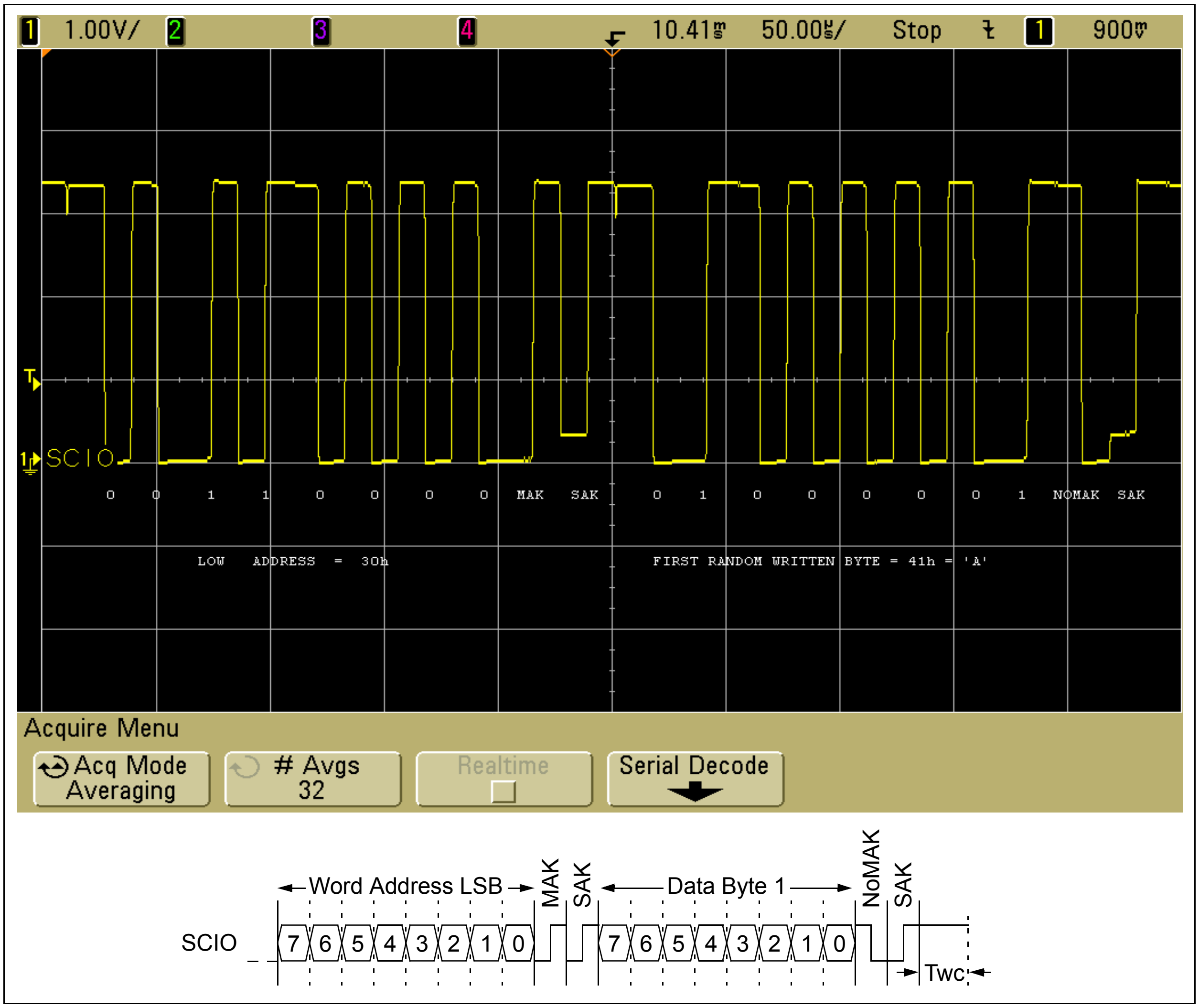
Data Byte and Command Termination

Once the word address has been transmitted and the last SAK received, the MCU sends the data byte.

After sending the data byte, the MCU terminates the command by generating a NoMAK in place of the MAK, and the serial EEPROM again responds with a SAK. This also initiates the internal write cycle (TWC).

Figure 6 shows the transmission of the low address byte and the data byte, as well as the NoMAK and SAK.

FIGURE 6: DATA BYTE AND STOP BIT



WRITE-IN-PROCESS POLLING

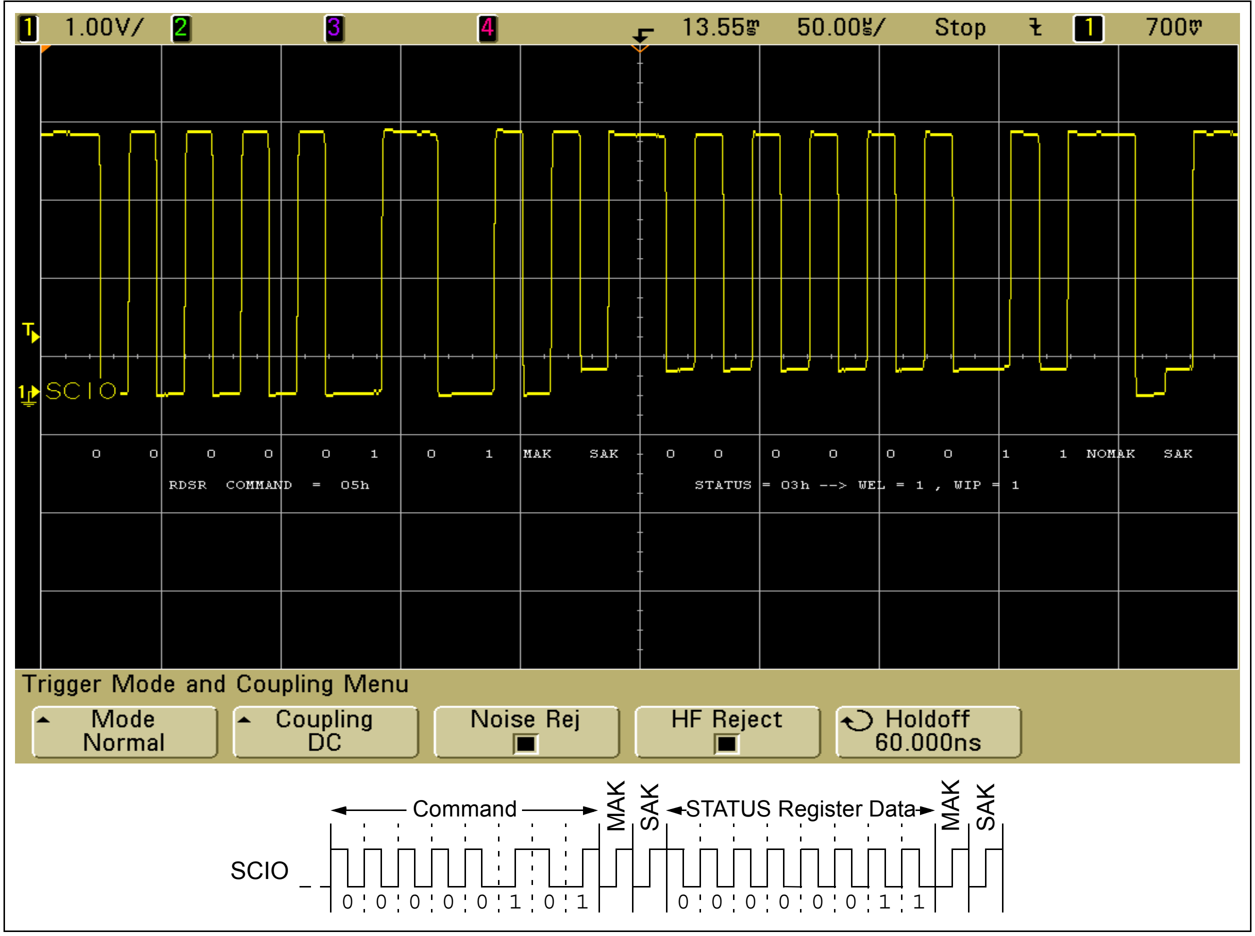
After an array or STATUS register write instruction is executed, the MCU must observe a write cycle time (TWC). Write cycle time is a maximum, so the actual time required is typically less. Therefore, to transfer data as efficiently as possible, using the Write-In-Process (WIP) polling feature is highly recommended. Because the STATUS register can be read during a write cycle, the WIP bit can be continuously monitored to determine the completion of the write cycle.

Write-In-Process Polling Routine

The process of WIP polling consists of the MCU sending a start header and device address after observing the TSS period. The MCU follows this by sending the Read Status Register (RDSR) command (‘`00000101`’ or 0x05) and MAK. After sending the subsequent SAK, the serial EEPROM transmits the STATUS register. At this point, the STATUS register can be requested again by sending a MAK. The WEL and WIP values sent are updated dynamically, so the MCU can continuous check the STATUS register. Sending a NoMAK terminates the command.

Figure 7 shows an example of WIP polling to check if a Write operation has finished. In this example, the WIP bit is set (‘`1`’), indicating that the write cycle has not yet completed.

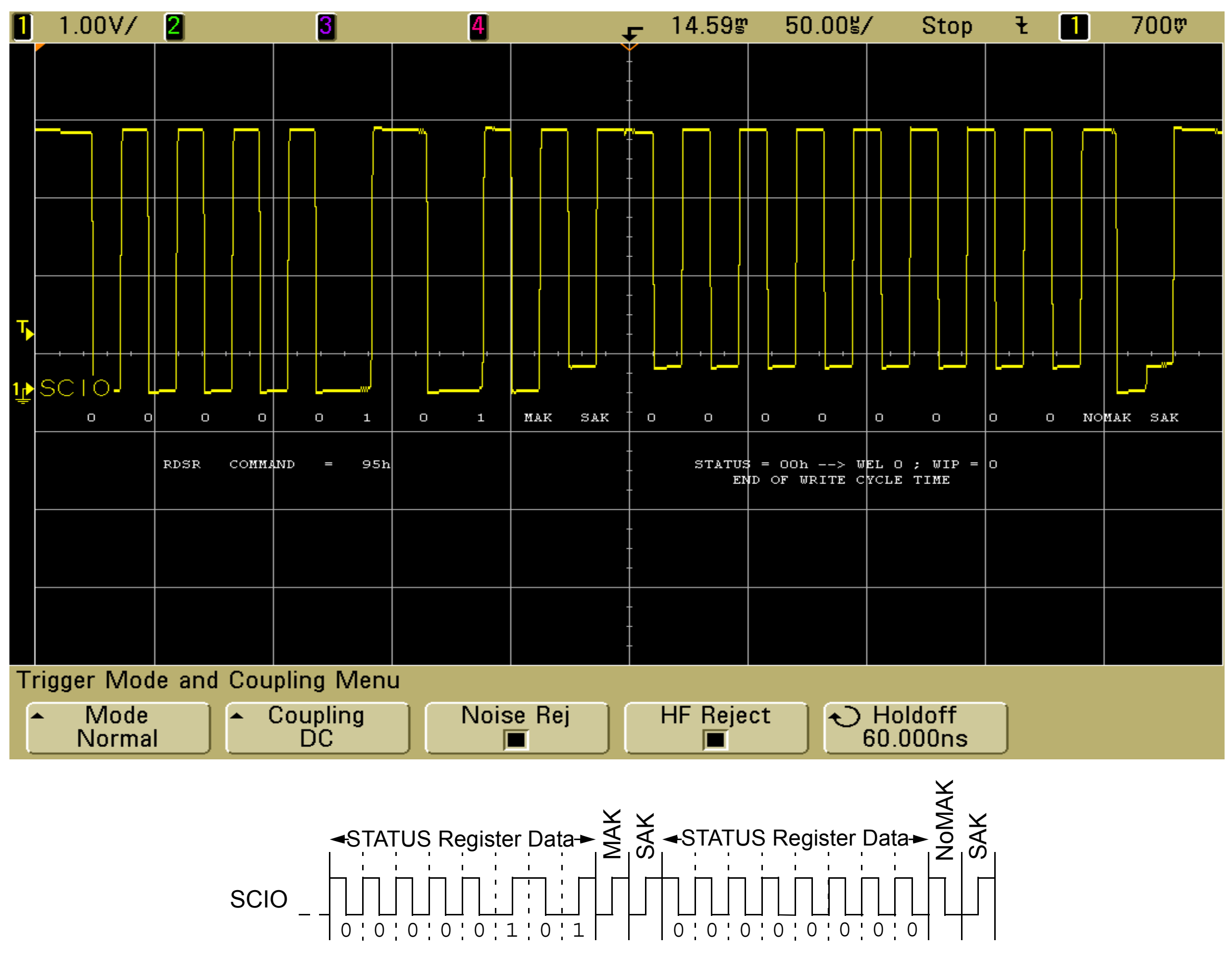
FIGURE 7: WIP POLLING ROUTINE (SHOWING WRITE-IN-PROCESS)


1 1.00V/ 2 3 4 13.55㎳ 50.00㎲/ Stop 1 700㎷
SCIO
0 0 0 0 0 1 0 1 MAK SAK 0 0 0 0 0 0 1 1 NOMAK SAK
RDSR COMMAND = 05h
STATUS = 03h --> WEL = 1 , WIP = 1
Trigger Mode and Coupling Menu
Mode
Normal
Coupling
DC
Noise Rej
HF Reject
Holdoff
60.000ns
Command
MAK
SAK
STATUS Register Data
MAK
SAK
SCIO
0 0 0 0 0 1 0 1
0 0 0 0 0 0 1 1

WIP Polling Complete

Figure 8 shows the final read of the STATUS register after the Page Write operation, in which the WIP bit is clear (‘0’). This indicates that the write cycle is complete and the serial EEPROM is ready to continue.

FIGURE 8: WIP POLLING FINISHED (SHOWING WRITE CYCLE COMPLETE)



1 1.00V/ 2 3 4 14.59s 50.00µs/ Stop 1 700mV

RDSR COMMAND = 95h

STATUS = 00h --> WEL 0 ; WIP = 0
END OF WRITE CYCLE TIME

Trigger Mode and Coupling Menu

Mode Normal | Coupling DC | Noise Rej ■ | HF Reject ■ | Holdoff 60.000ns

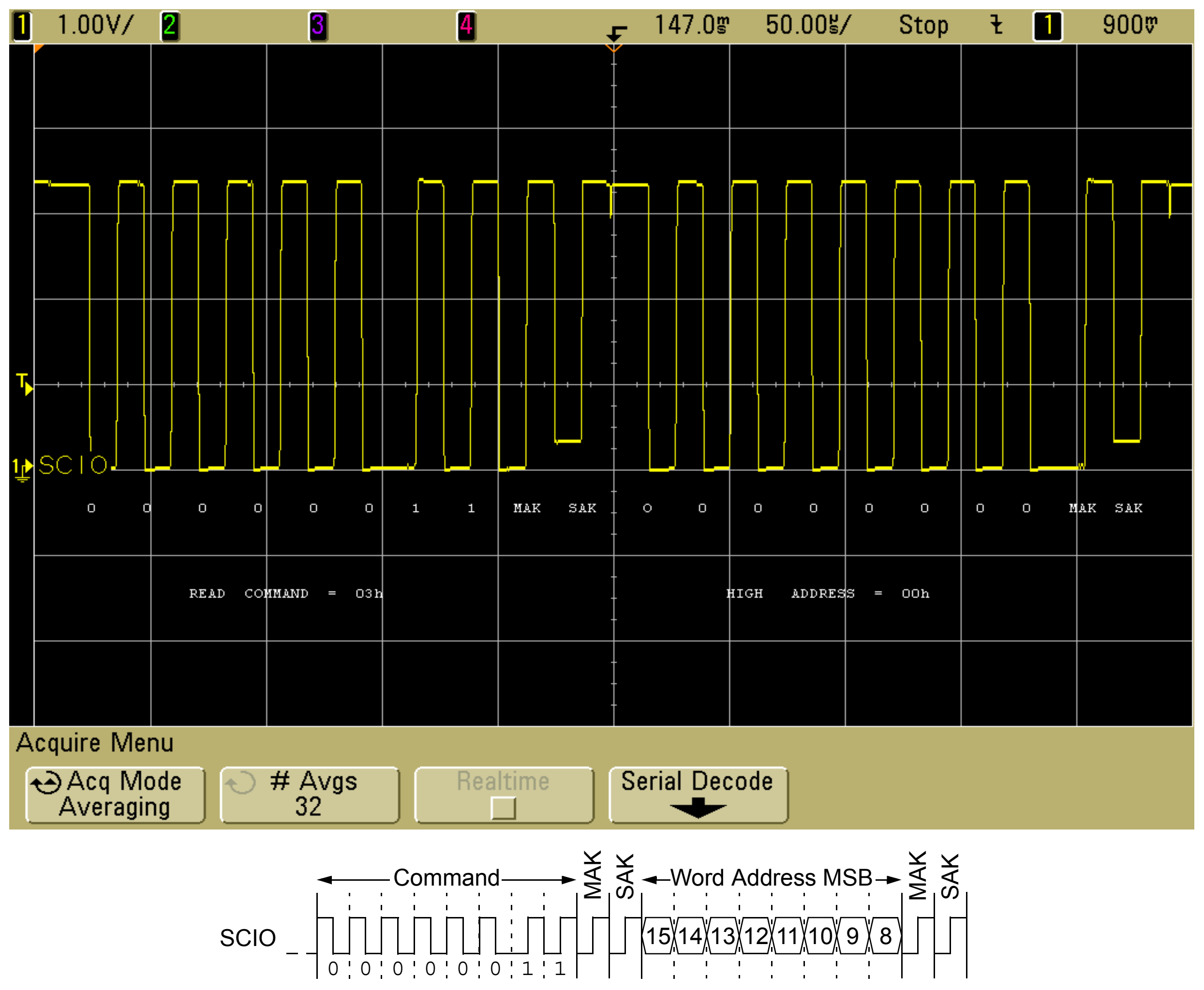
BYTE READ

The byte read operation can be used to read data from the serial EEPROM. The start header and device address must first be sent as in a byte write operation; they have been omitted from this section. The MCU transmits the command byte followed by the word address bytes to the serial EEPROM. The MCU generates a MAK after each byte, and this is followed by a SAK if there are no errors.

Command and Word Address for Read

Figure 9 shows an example of the Read command '00000011' or 0x03, followed by the high address byte. The low address byte has been omitted from this example.

FIGURE 9: BYTE READ (COMMAND BYTE AND WORD ADDRESS)



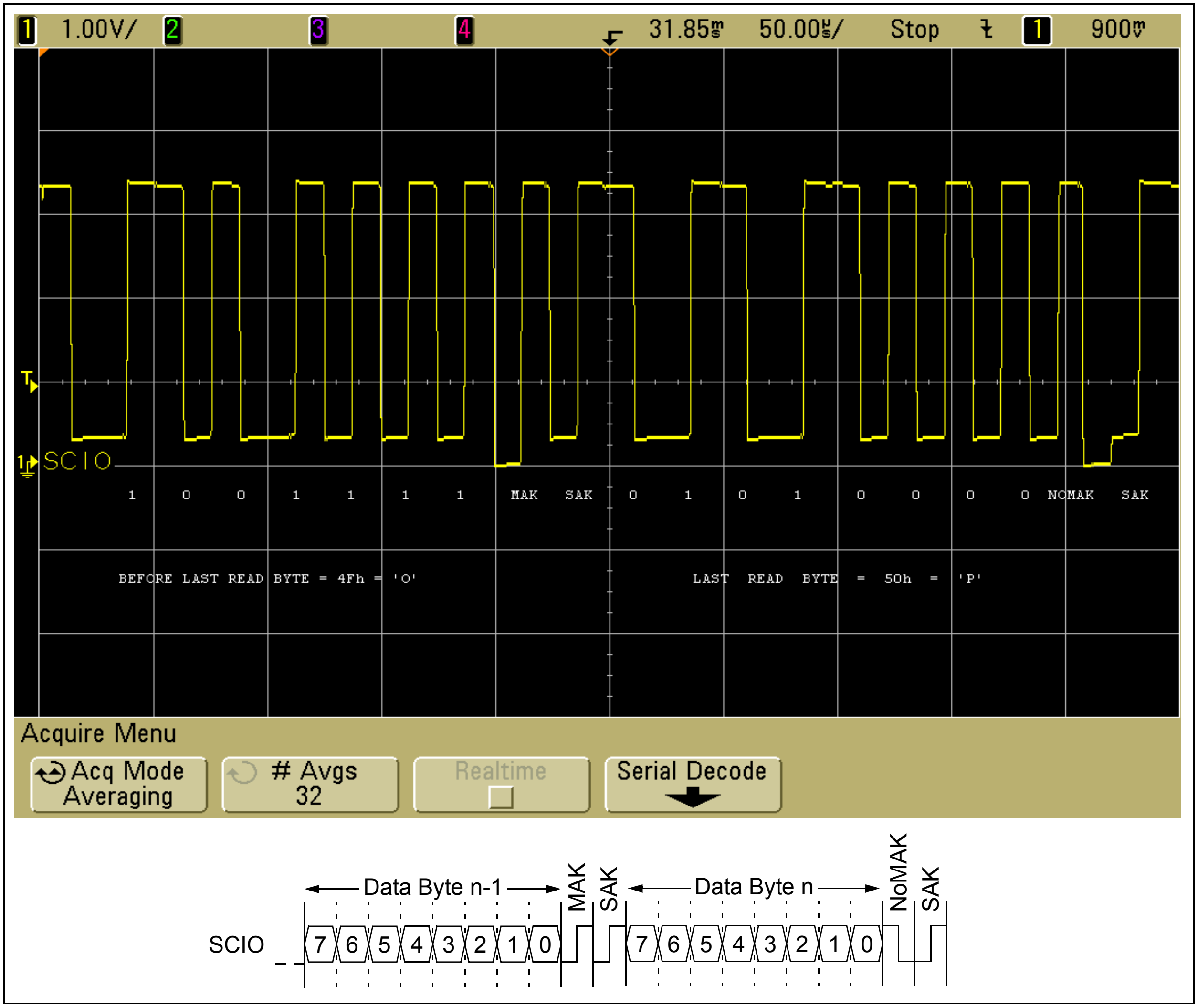
Reading Data Bytes Back

After the Read command and word address have been sent and acknowledged, the serial EEPROM starts to send the data from the array starting at the address specified.

To read a single byte, the MCU generates a NoMAK after the byte is read. To continuously read the array, the MCU generates a MAK after each data byte. The serial EEPROM responds with a SAK if there are no errors.

Figure 10 shows the MCU reading two bytes of data. The MCU sends a NoMAK after the second byte to indicate that no more data is requested and to terminate the command.

FIGURE 10: BYTE READ (DATA BYTES AND COMMAND TERMINATION)


1 1.00V/ 2 3 4 31.85ms 50.00us/ Stop 1 900mV
SCIO
1 0 0 1 1 1 1 MAK SAK 0 1 0 1 0 0 0 0 NOMAK SAK
BEFORE LAST READ BYTE = 4Fh = 'O'
LAST READ BYTE = 50h = 'P'
Acquire Menu
Acq Mode Averaging
# Avgs 32
Realtime
Serial Decode
Data Byte n-1
MAK
SAK
Data Byte n
NoMAK
SAK
SCIO
7 6 5 4 3 2 1 0
7 6 5 4 3 2 1 0

PAGE WRITE

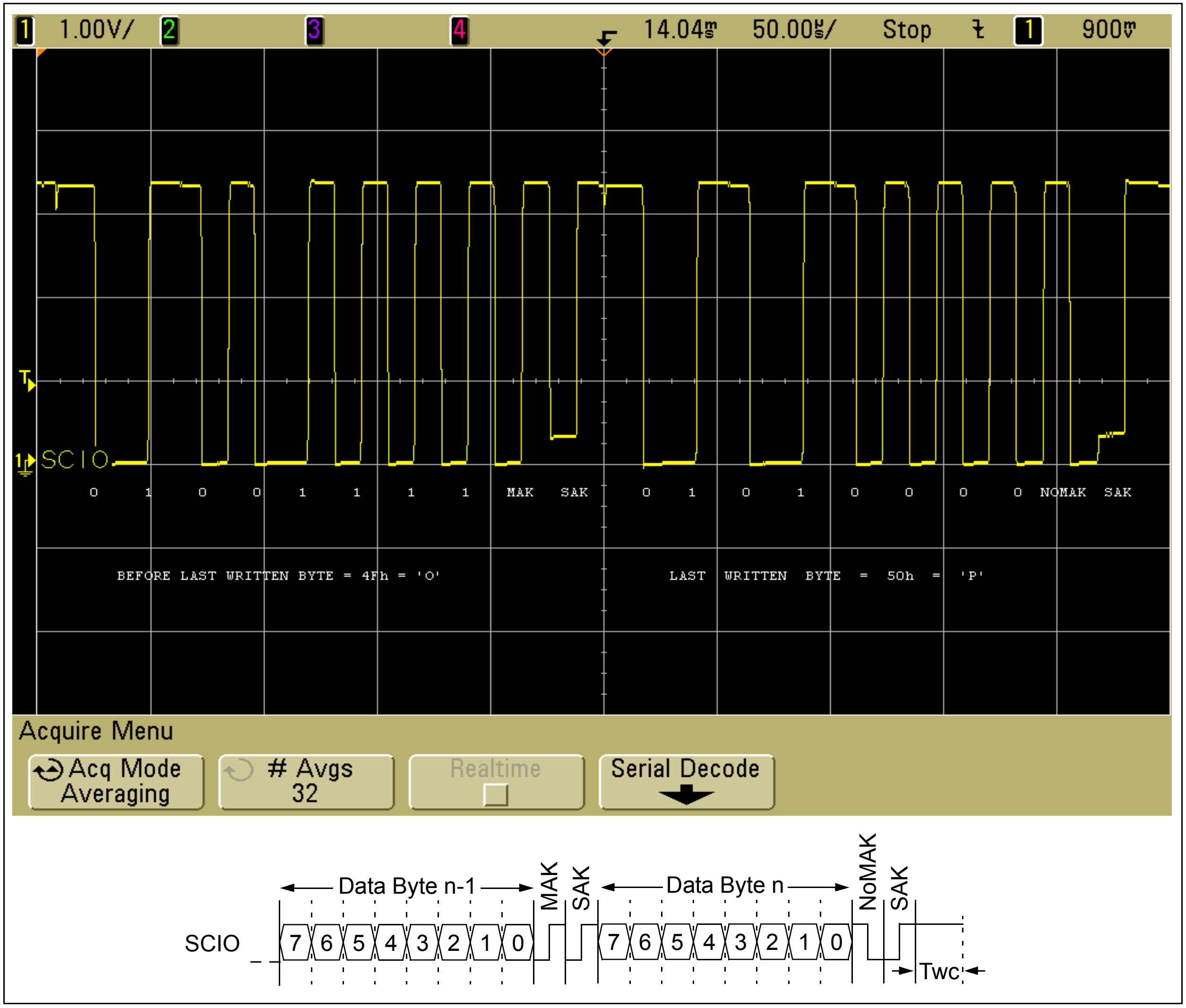
Page write operations provide a technique for increasing throughput when writing large blocks of data. The serial EEPROM features a 16-byte page. By using the page write feature, up to 1 full page of data can be written consecutively, with the start header, device address, command and word address bytes being transmitted only once. It is important to point out, however, that 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 size, and end at addresses that are [integer multiples of the page size] minus 1. Attempts to write across a page boundary result in the data being wrapped back to the beginning of the current page, thus overwriting any data previously stored there.

The page write operation is very similar to the byte write operation. However, instead of generating a NoMAK after the first data byte has been transmitted, the MCU continues to send more data bytes, up to 1 page total. The serial EEPROM automatically increments the internal Address Pointer with receipt of each byte. As with the byte write operation, the internal write cycle (TWC) is initiated by the NoMAK generated by the MCU.

Sending Multiple Bytes Successively

Figure 11 shows two consecutive data bytes during a page write operation. Notice that a MAK is sent after the first byte of data and a NoMAK is sent after the last byte of data.

FIGURE 11: PAGE WRITE (TWO CONSECUTIVE DATA BYTES)



BEFORE LAST WRITTEN BYTE = 4Fh = 'O'

LAST WRITTEN BYTE = 50h = 'P'

CONCLUSION

This application note offers designers a set of firmware routines to access UNI/O serial EEPROMs using a generic I/O pin on the MCU. The code demonstrates byte and page operations. All routines were written in assembler 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 µVision3 IDE.

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, rfPIC, SmartShunt and UNI/O 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^{32} logo, PowerCal, PowerInfo, PowerMate, PowerTool, REAL ICE, rfLAB, Select Mode, Total Endurance, 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.

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

01/02/08