MICROCHIP

AN1234

Using C and a Hardware Module to Interface Texas Instruments' MSP430XXXX MCUs with SPI Serial EEPROMs

Author: *Alexandru Valeanu*
Microchip Technology Inc.

INTRODUCTION

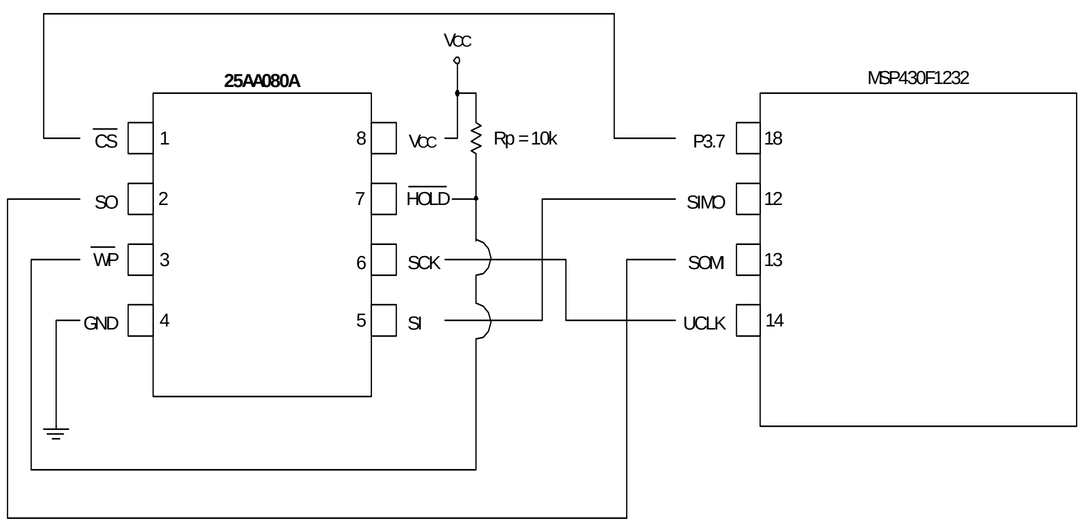
The 25XXX series serial EEPROMs from Microchip Technology support a half-duplex SPI protocol. The bus is controlled by the microcontroller (master), which accesses the 25XXX serial EEPROM (slave). The bus signals required consist of a clock input (SCK) plus separate data in (SI) and data out (SO) lines. Access to the 25XXX serial EEPROM is controlled through a Chip Select ($\overline{\text{CS}}$) input. Maximum clock frequencies range from 3 MHz to 20 MHz. Communication to the 25XXX serial EEPROM can be paused via the hold pin ($\overline{\text{HOLD}}$). While the EEPROM is paused, transitions on its inputs are ignored, except for the $\overline{\text{CS}}$, allowing the MCU to service higher priority interrupts. After releasing the $\overline{\text{HOLD}}$ pin, the operations resume from the point when the hold was asserted.

The main features of the 25XXX serial EEPROMs are:

- SPI-compatible serial interface bus
- EEPROM densities range from 1 Kbits to 1 Mbits
- Bus speed from 3 MHz to 20 MHz
- 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
- Built-in write protection

This application note is part of a series that provide source code to help users implement the protocol with minimal effort. Figure 1 is the hardware schematic depicting the interface between Microchip's 25XXX series serial EEPROMs and the MSP430F1232 based MCU from Texas Instruments. The schematic shows the necessary connections between the MCU and the serial EEPROM. The $\overline{\text{WP}}$ and $\overline{\text{HOLD}}$ pins are tied to VCC through a resistor, as they are not used in the examples provided.

FIGURE 1: CIRCUIT FOR MSP430XXXX AND 25XXX SERIAL EEPROM



Note: A 100 nF decoupling capacitor should be connected between VCC and GND.

FIRMWARE DESCRIPTION

This application note offers designers a set of examples for the read and write functions for the Microchip SPI serial EEPROM (byte read/write and page read/write) using internal hardware peripheral and C language.

The main routine writes a string in the SPI serial EEPROM, reads it back and compares the two strings, displaying a success or error message on the 4 onboard LEDs of the evaluation board. The firmware was written in C language for the MSP430F1232 MCU, using the IAR™ – IDE and the related C compiler. It was developed on the Softbaugh™ ES1232 evaluation board and debugged through the MSP430 USB debug interface, MSP-FET430UIF, from Texas Instruments. The code was tested using the 25AA080A serial EEPROM.

Oscilloscope screen shots are shown in this application note. All timings are based on the internal RC oscillator of the MCU (~ 8 MHz). If another clock is used, the code must be modified to generate the correct delays (mainly the 5 milliseconds delay, which is an alternative to the polling of the WIP flag, also presented in the application note) for the EEPROM write cycle. The bus speed in these examples is of ~ 200 kHz. If desired, the bus speed may be changed in the initialization routine (`ini_spi`) by modifying the U0BR0 and U0BR1 registers. (Please refer to the **Section "Initialization"**).

INITIALIZATION

Initialization consists of three routines: `ini_gpio`, `ini_spi` and `ini_memspi`.

The `ini_gpio` routine sets the SPI pins for their functions P3.3 = UCLK (SPI CLK), P3.2 = SOMI0 (Slave Out Master In), P3.1 = SIMO0 (Slave In Master Out). The P3.7 pin is used as GPIO output, driving the $\overline{CS}$ pin of the SPI memory.

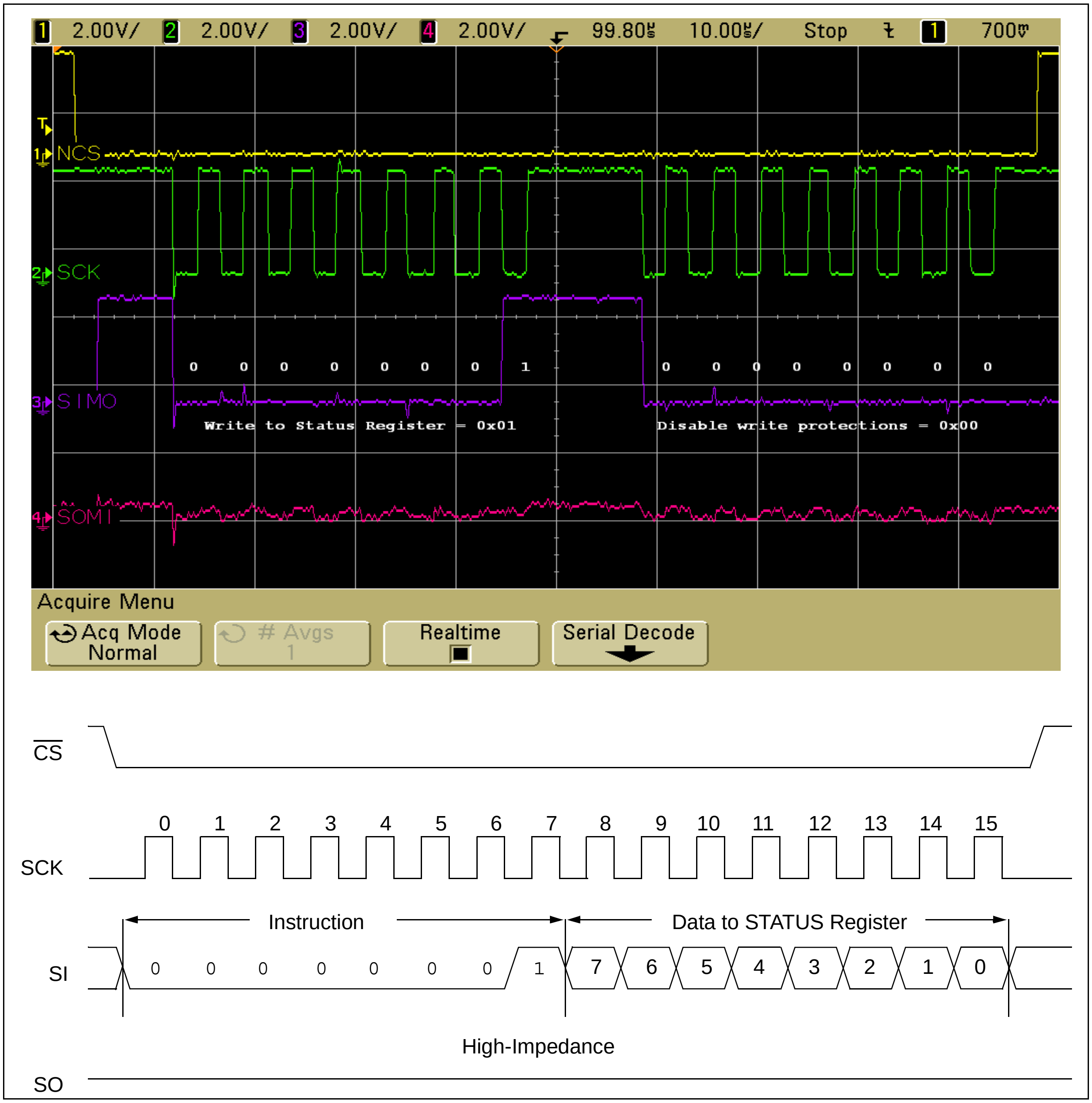
In addition, the function sets as GPIO outputs P1.3, P1.2, P1.1 and P1.0, driving the 4 onboard LEDs in order to display success or error messages.

The `ini_spi` routine prepares the MCU for communication with the serial EEPROM, using the hardware peripheral and setting the following registers: U0CTL, U0TCTL, ME, U0BR1 and U0BR0. The internal part will be configured for: 8-bits character, USART = SPI master, SCK = Idle high, SPICLK = SMCLK: 16, 3 wires scheme, enable SPI module.

If another speed is desired, the U0BR0 and U0BR1 registers must be set to other values.

The third routine, `ini_memspi`, prepares the memory for further writes. It sends to the device a 0x00 byte in order to disable all write protections. The scope plot showing this operation is depicted in Figure 2.

FIGURE 2: WRITE TO STATUS REGISTER



WRITE ENABLE

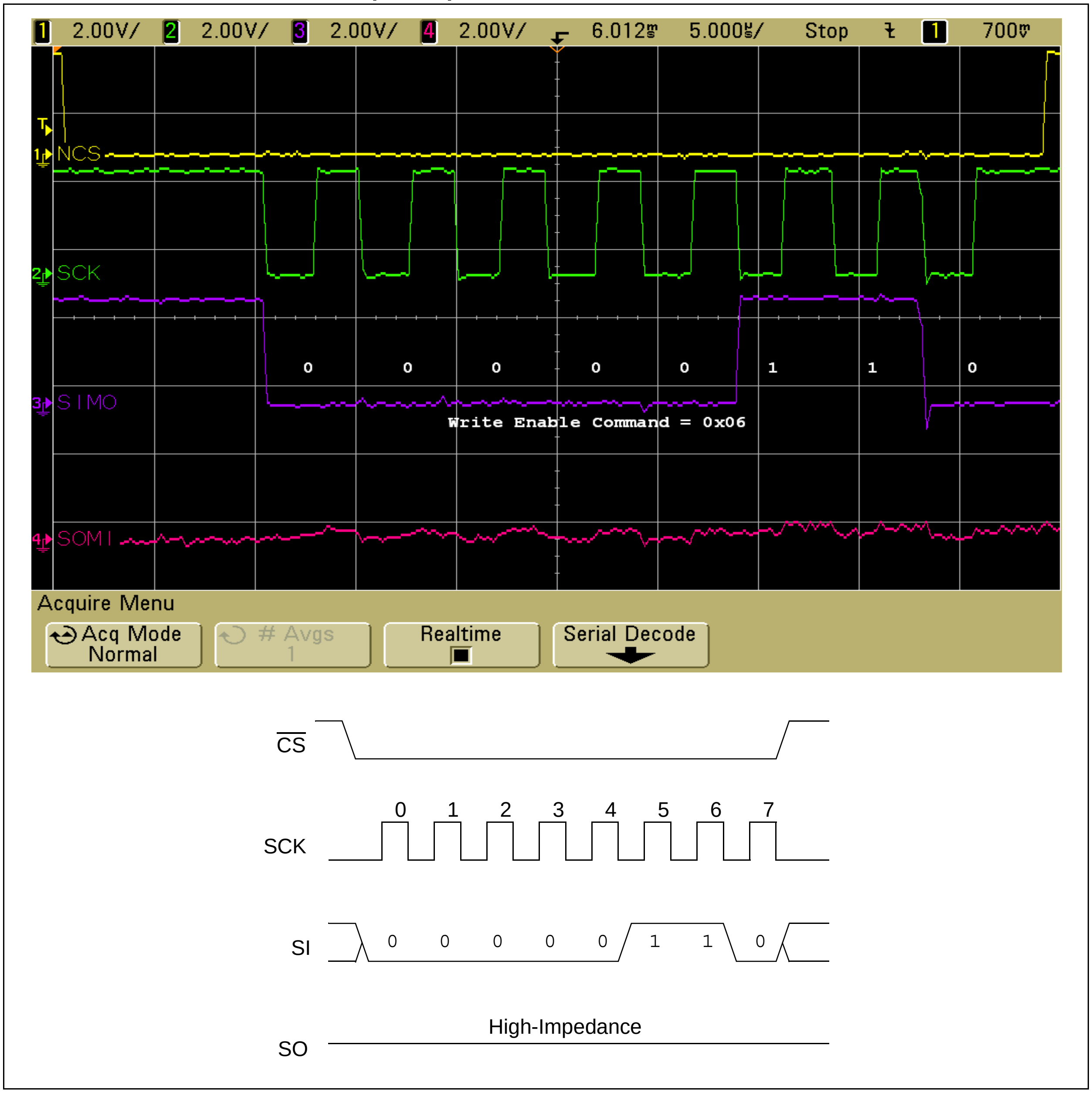
Before any write operation can occur, the MCU must set the Write Enable Latch (WEL).

This is done by issuing a WREN command.

The MCU clears the WEL bit by issuing a Write Disable (WRDI) command. The WEL bit is also automatically reset if the serial EEPROM is powered down and when a write cycle is completed.

Figure 3 shows the WREN command.

FIGURE 3: WRITE ENABLE (WREN)

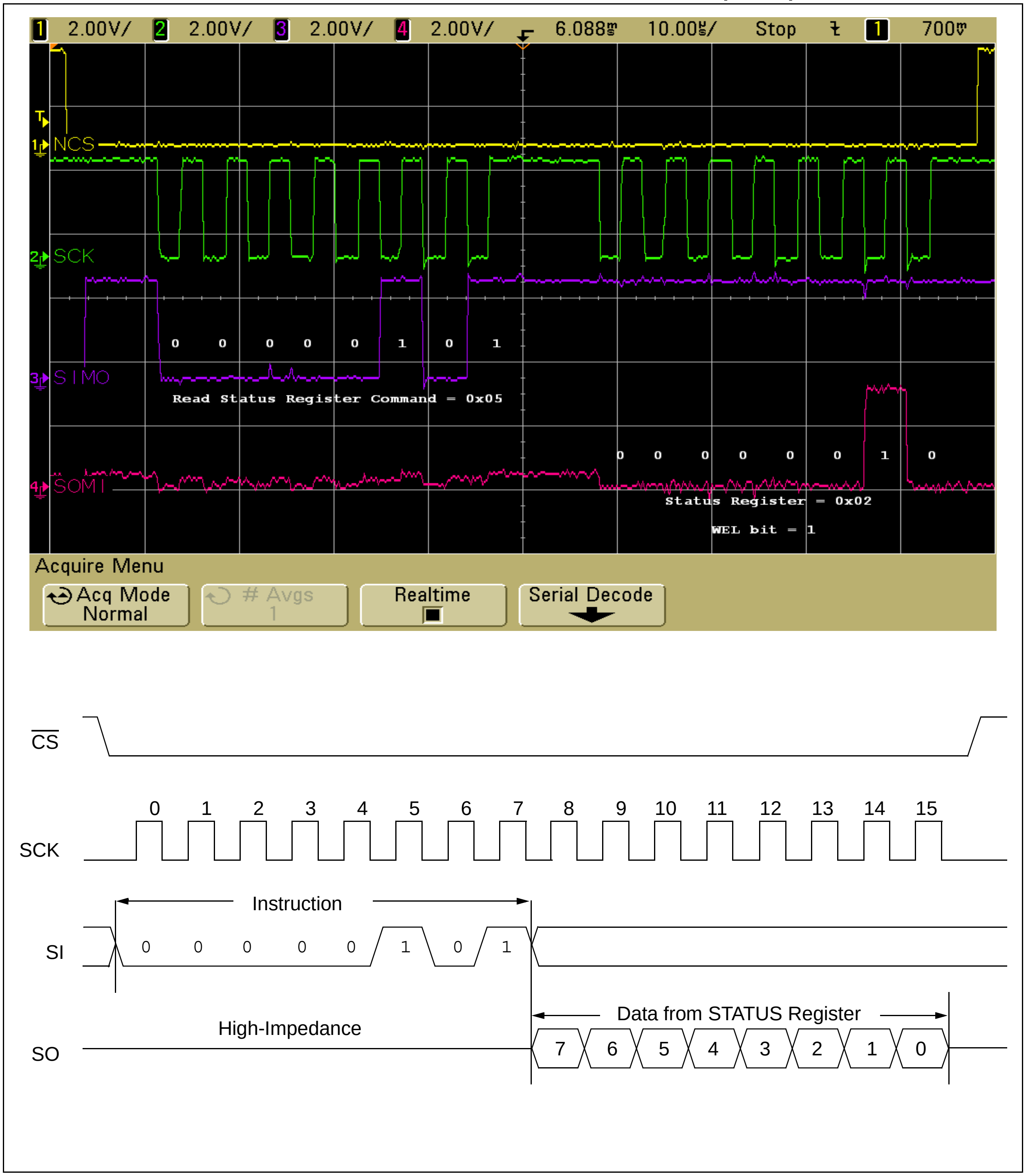


READ STATUS REGISTER TO CHECK FOR WEL BIT

Figure 4 shows an example of the Read Status Register command to check for the WEL bit. This bit must be set before a write is attempted either to the STATUS register or the array. Before attempting to write, a good programming practice is to check the WEL bit. Once again, the device is selected and the opcode for a Read Status Register is sent.

The STATUS register is shifted out on the Serial Out pin. A value of 0x02 shows that the WEL bit in the STATUS register has been set. The device is now ready to do a write.

FIGURE 4: READ STATUS REGISTER TO CHECK FOR WEL BIT (RDSR)

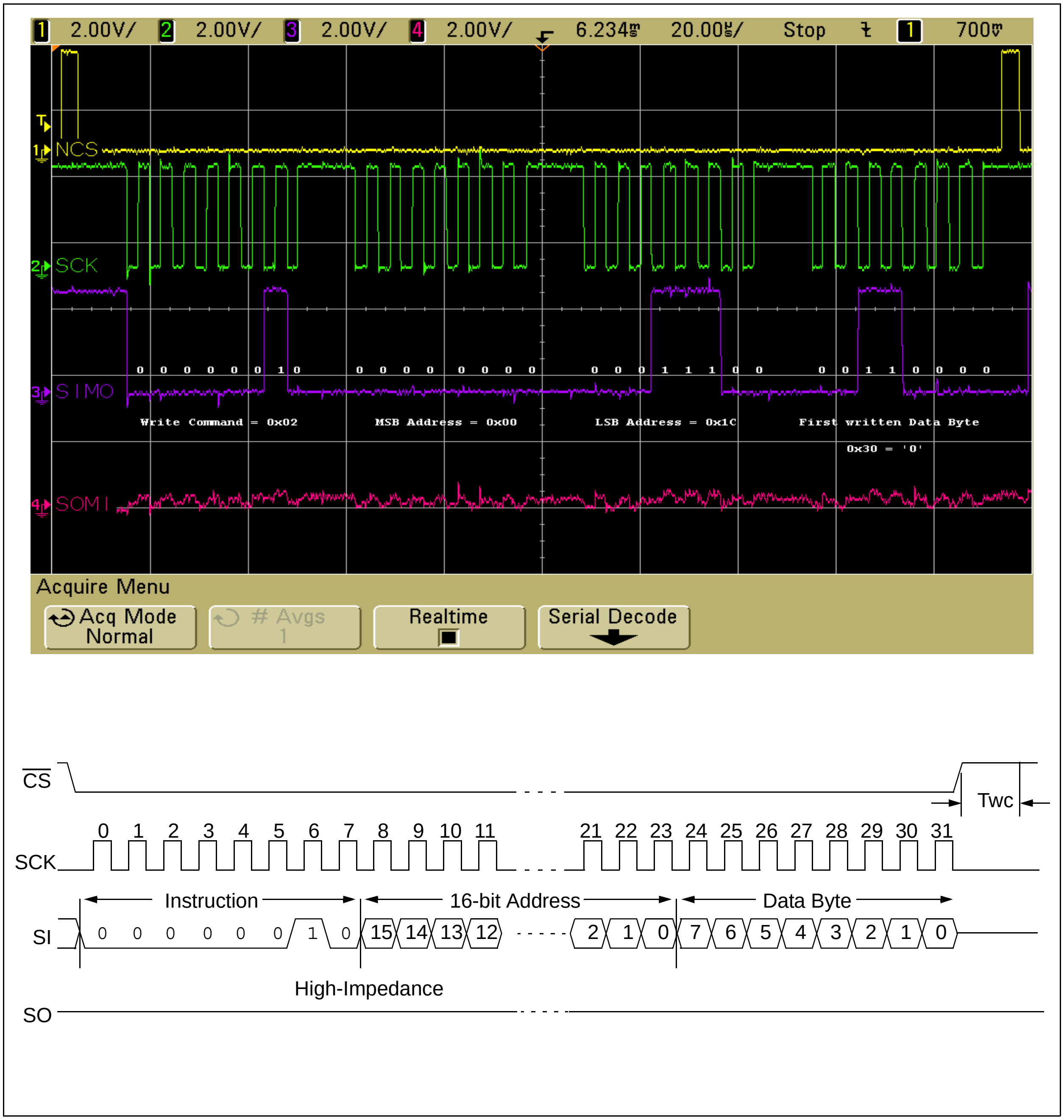


BYTE WRITE SEQUENCE

The byte write operation consists of the MCU sending the Write command, followed by the word address and the data byte. The word address for the 25XX080A is a 16-bit value, so, two bytes must be transmitted for the entire word address, with the Most Significant Byte (MSB) sent first. Note that the WREN command is not illustrated in this section but is still required to initiate the operation.

Figure 5 shows the following sequence: Write command (0x02), MSB address (0x00), LSB address (0x1C) and the first written byte (0x30 = 0).

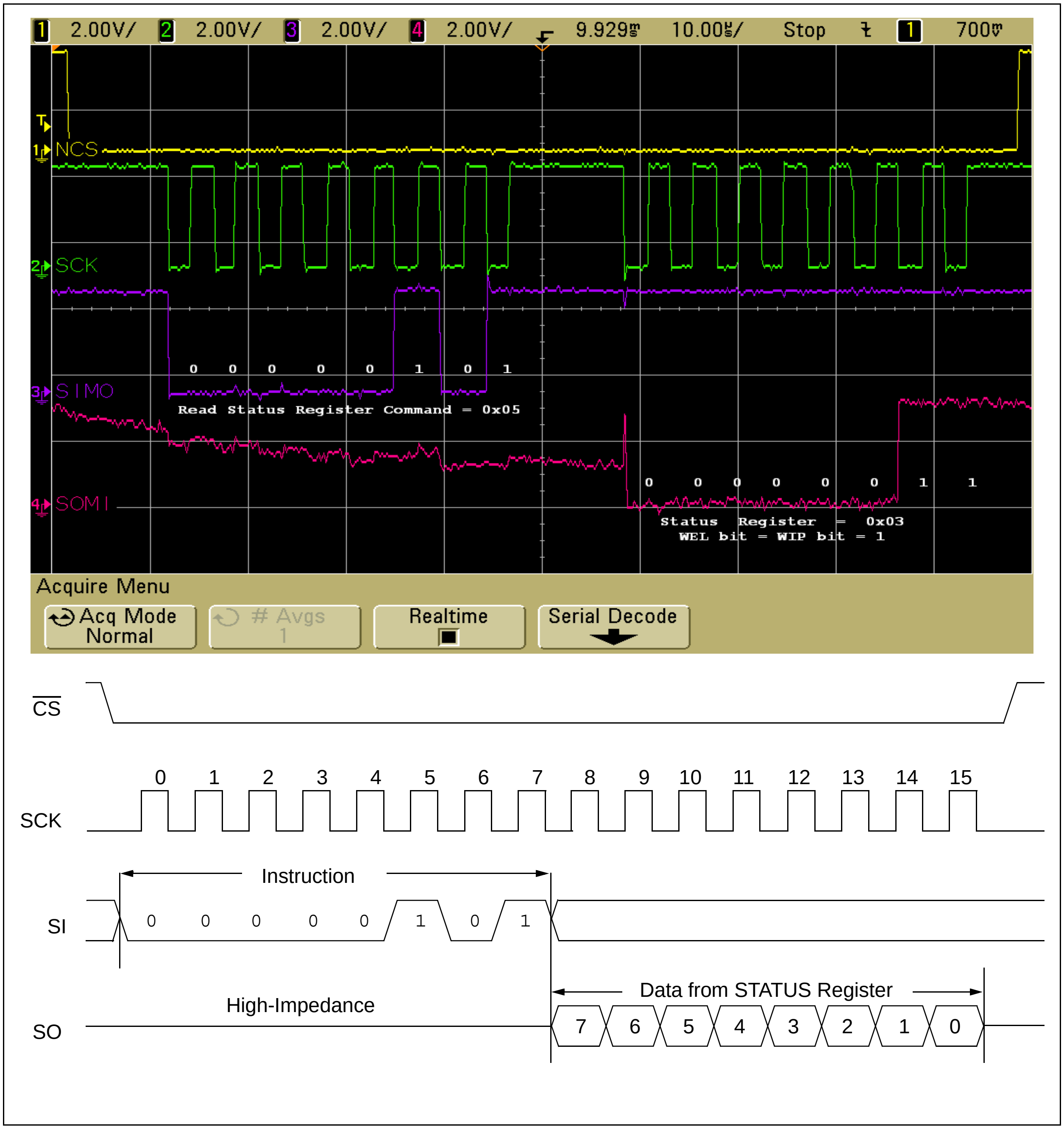
FIGURE 5: BYTE WRITE COMMAND, ADDRESS AND DATA



RDSR – CHECK FOR WIP SET

After the MCU issues a Write command, the STATUS register may be read to check if the internal write cycle has been initiated, and it can be continuously monitored to look for the end of the write cycle. The MCU selects the serial EEPROM and sends the Read Status Register command (RDSR) (`00000101` or 0x05), as shown in Figure 6. The STATUS register is then shifted out on the Serial Out (SO) pin, resulting in a value of `00000011` or 0x03, also shown in Figure 6. Both the WEL bit (bit 1) and the WIP bit (bit 0) are set (`1`), indicating that the write cycle is in progress.

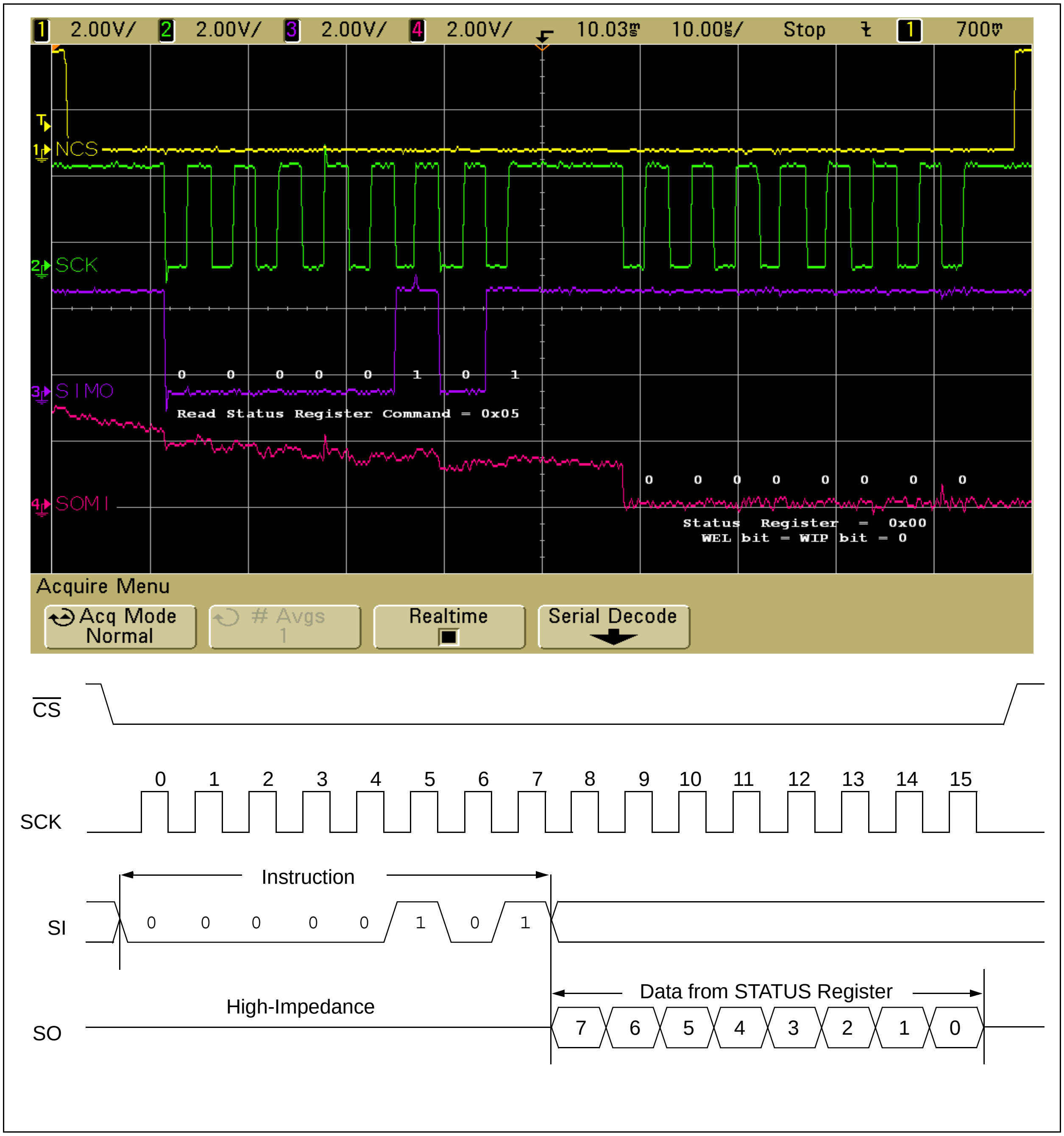
FIGURE 6: READ STATUS REGISTER TO CHECK FOR WIP BIT SET



RDSR – WIP BIT CLEARED

The firmware remains in a continuous loop and the WIP status is evaluated until the WIP bit is cleared ('0'). Figure 7 shows the RDSR command. This is followed by a value of 0x00 being shifted out on the SO pin, indicating that the write cycle has finished and the serial EEPROM is ready to receive additional commands. The WEL bit is also cleared at the end of a write cycle, which serves as additional protection against unwanted writes.

FIGURE 7: READ STATUS REGISTER – WIP BIT CLEARED



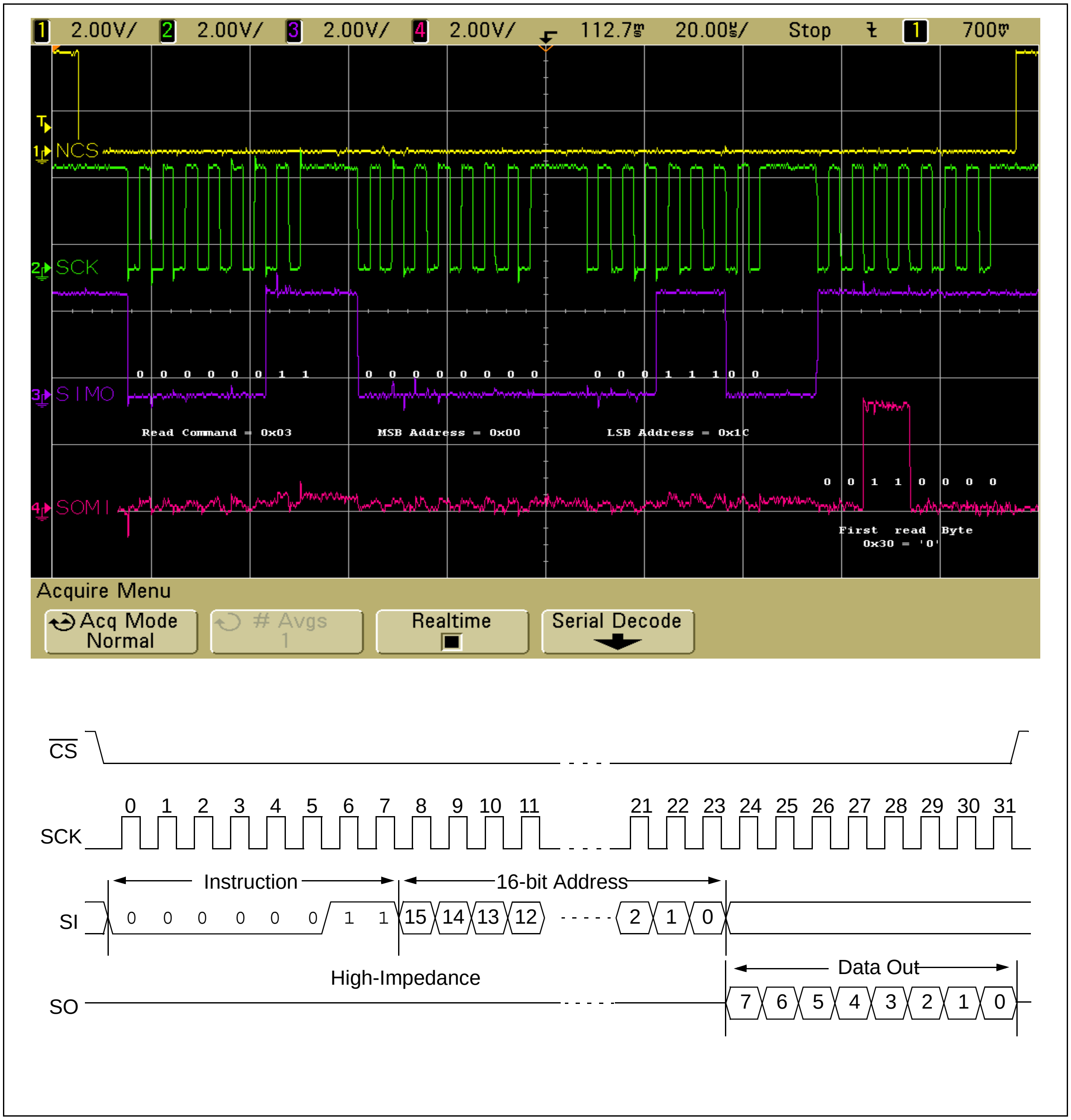
READ BYTE SEQUENCE

The byte read operation can be used to read data from the serial EEPROM.

The MCU transmits the Read command byte (0x03) followed by the word address bytes (MSB = 0x00, LSB = 0x1C) to the serial EEPROM.

Figure 8 shows an example of the Read command, followed by the MSB and LSB address bytes, and the first read byte (0x30 = 0). After the MCU reads the data byte, it will raise up the $\overline{CS}$ signal in order to end the command.

FIGURE 8: READ COMMAND, ADDRESS AND DATA



PAGE WRITE SEQUENCE

Page write operations provide a technique for increasing throughput when writing large blocks of data. The 25XX080A serial EEPROM features a 16-bytes page. Up to 1 full page of data can be written consecutively by using the page write feature.

It is important to note that page write operations are limited to writing bytes within a single physical page, regardless of the number of bytes actually 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 at 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. The serial EEPROM automatically increments the internal Address Pointer to the next higher address with receipt of each byte. It starts with the same 3 bytes: Write command, MSB address, LSB address. Comparing to the byte write function, the only difference is that the page write routine stops the communication after several data bytes (not after the first one) by raising up the $\overline{CS}$ signal.

Figure 9 shows the last 2 written characters during a page write operation (0x4F = O and 0x50 = P).

The firmware of this application note presents a useful feature: the string write function (`spi_wrstr`).

The routine has the following tasks:

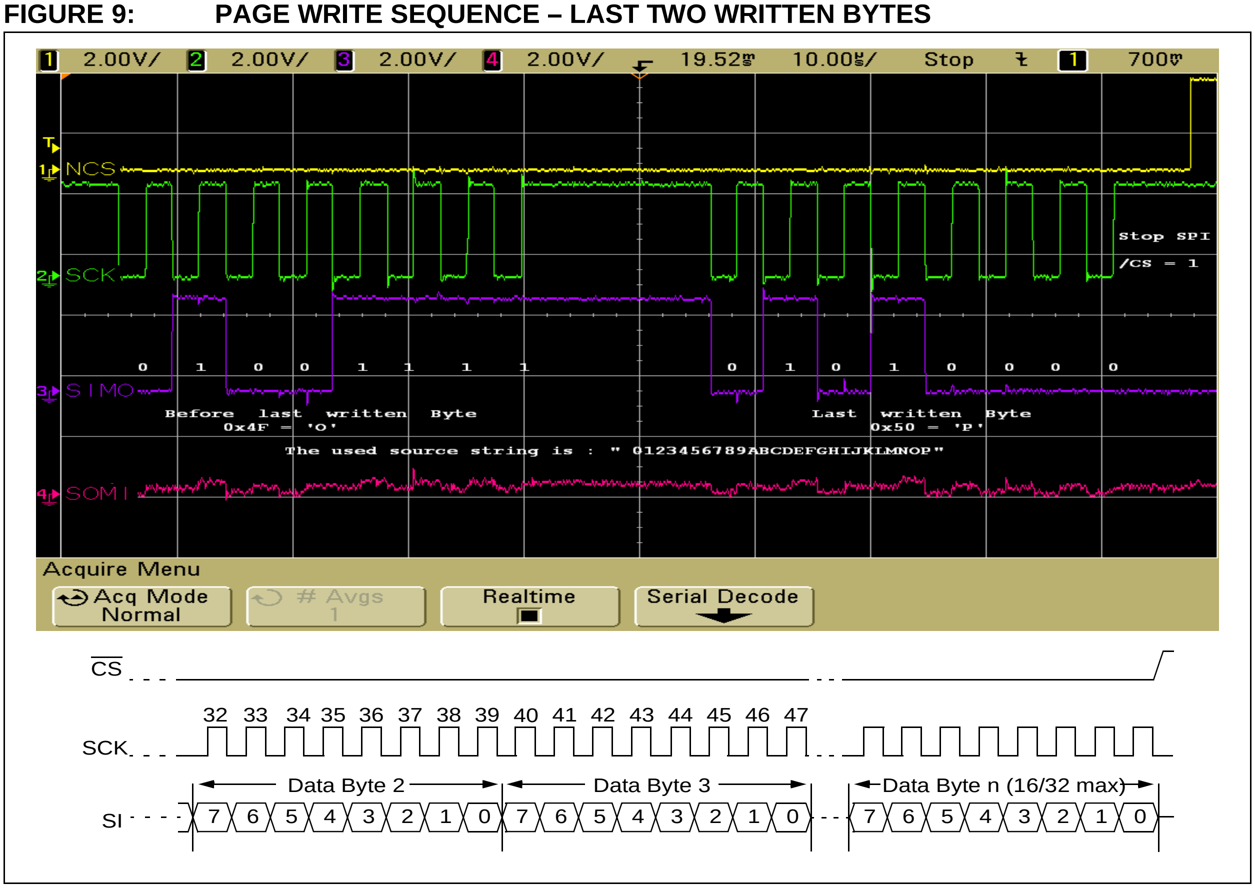
- calculates the length of the string to be written in the memory
- calculates the size of the substrings to be written inside an individual page
- splits accordingly the initial string in the related substrings
- inserts after each substring the related write cycle time (delay or polling of the WIP flag)
- re-initializes the start address for each substring (page)
- for all of these, it calls several times the page write function = `spi_wrpg`

Also, it features the following advantages:

- the most general method to write strings
- the fastest method (minimum of Twc periods)
- the most economical: saves memory space, by overriding page boundaries
- (no breaks between strings)
- increases the lifetime of the NV memory

Accordingly, by using this routine, the programmer must pass to the function only the name of the string to be written and the memory start address.

FIGURE 9: PAGE WRITE SEQUENCE – LAST TWO WRITTEN BYTES

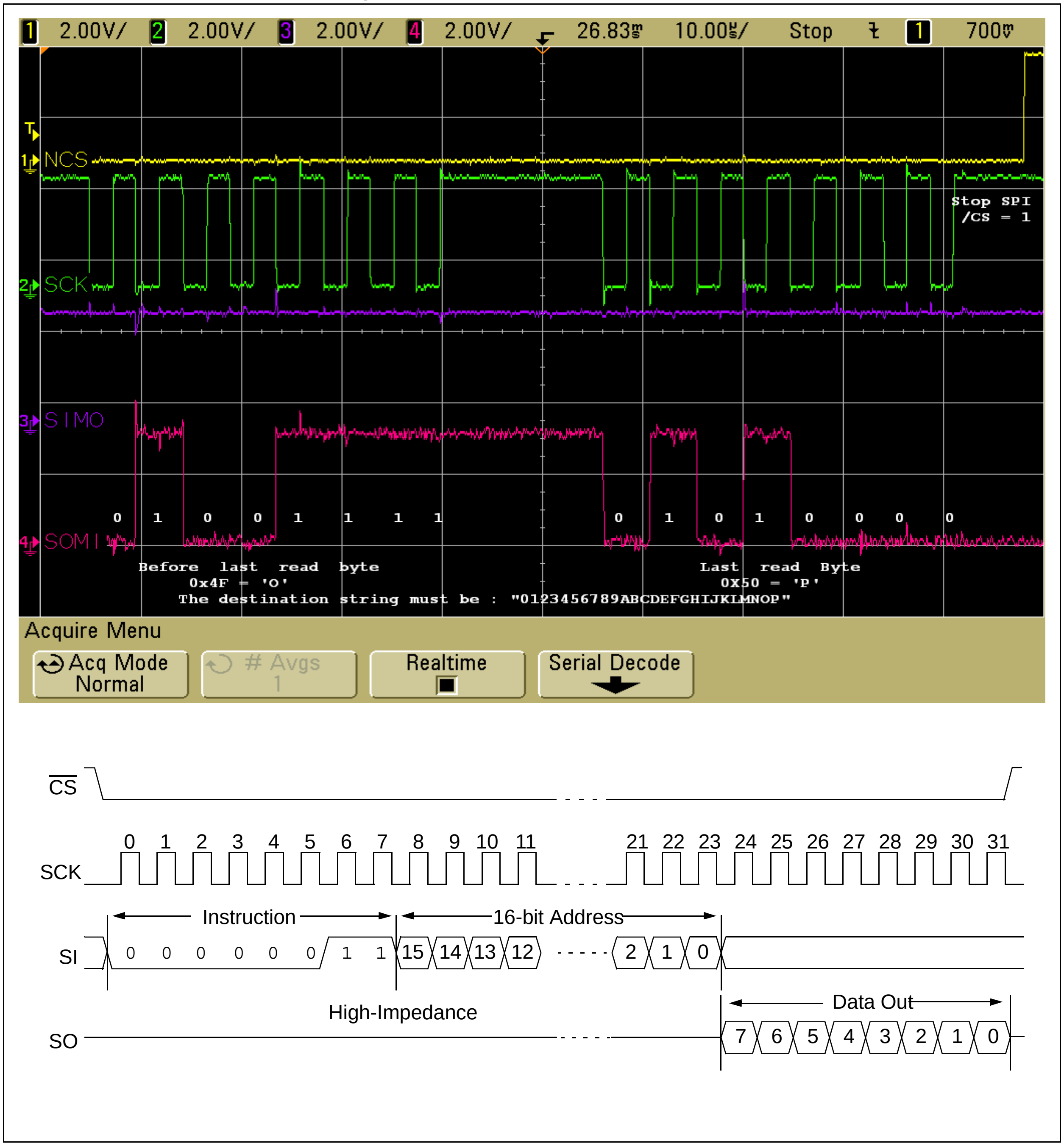


PAGE READ SEQUENCE

Page read operations read a complete string, starting with the specified address. In contrast to the page write operations described on the previous page, there is no maximum length for the page read.

After 1 Kbyte has been read, the internal address counter rolls over to the beginning of the array. Figure 10 depicts the last two read bytes, as the start of the command (Read command, word address, first read byte) is the same as in the case of the read byte sequence.

FIGURE 10: PAGE READ SEQUENCE – LAST TWO BYTES



CONCLUSION

This application note offers designers a set of firmware routines to access Microchip's SPI serial EEPROMs using a hardware peripheral. The code demonstrates byte and page operations. All routines were written using the C compiler from IAR, included in the IDE of the same company. All experiments were performed on the ES-1232 evaluation board from Softbaugh, equipped with an MSP430F1232 MCU from Texas Instruments.

The code was debugged using the USB debug interface, MSP-FET430UIF, available from Texas Instruments

The firmware was tested using the schematic shown in Figure 1.

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.

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