AN1079

Using the C30 Compiler and the I²C™ Peripheral to Interface Serial EEPROMs with dsPIC33F

Author: *Martin Bowman*
Microchip Technology Inc.

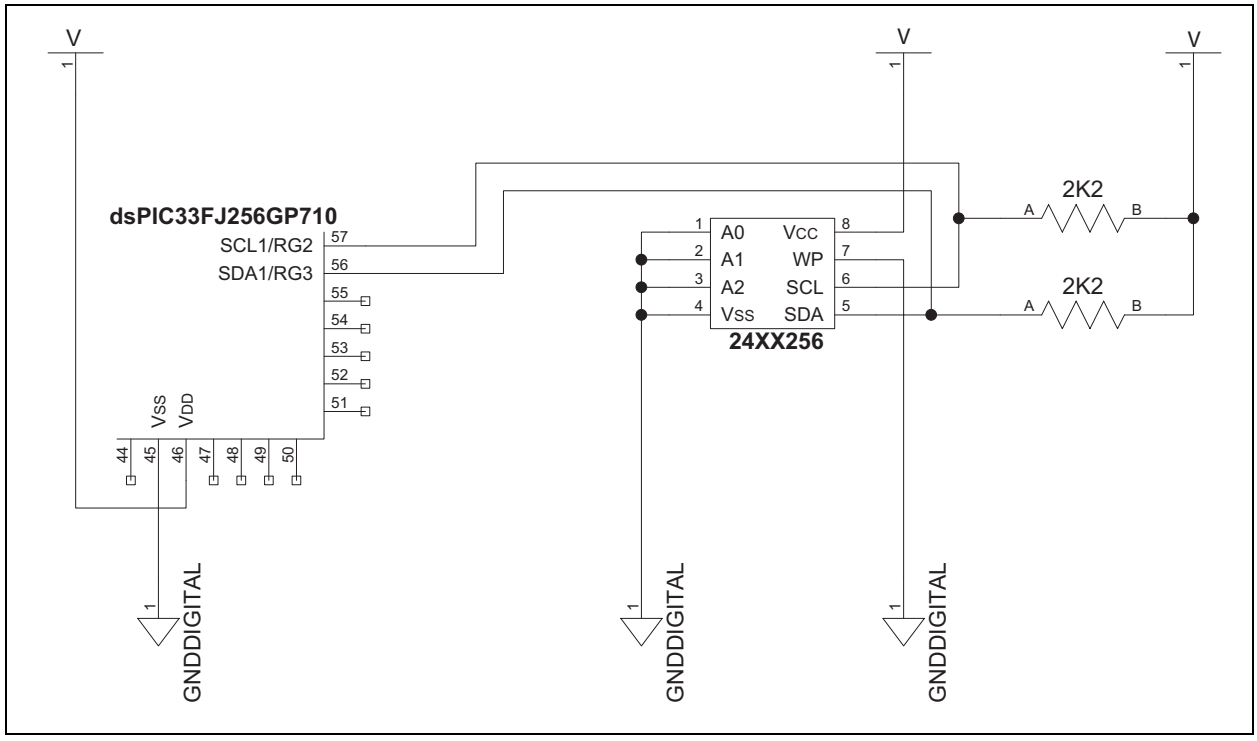
INTRODUCTION

The 24XXX series serial EEPROMs from Microchip Technology are I²C™ compatible and have maximum clock frequencies ranging from 100 kHz to 1 MHz. The I²C module available on the dsPIC33F family microcontrollers provides a very easy-to-use interface for communicating with the 24XXX series devices. The largest benefit of using the I2Cx module is that the signal timings are handled through hardware rather than software. This allows the firmware to continue executing while communication is handled in the background. This also means that an understanding of the timing specifications associated with the I²C protocol is not required in order to use the 24XXX series devices in designs.

This application note is intended to serve as a reference for communicating with Microchip's 24XXX series serial EEPROM devices with the use of the I2Cx module featured on the dsPIC33F family devices. Source code for common data transfer modes is also provided. The source code is easily transferable to the PIC24 family of devices.

Figure 1 describes the hardware schematic for the interface between Microchip's 24XXX series devices and the dsPIC33F device. The schematic shows the connections necessary between the microcontroller and the serial EEPROM as tested, and the software was written assuming these connections. The SDA and SCL pins are open-drain terminals, and therefore require pull-up resistors to VCC (typically 10 kΩ for 100 kHz, and 2 kΩ for 400 kHz and 1 MHz). Also, the WP pin is tied to ground because the write-protect feature is not used in the examples provided.

FIGURE 1: CIRCUIT FOR dsPIC33F AND 24XXX SERIES DEVICE



FIRMWARE DESCRIPTION

The purpose of the firmware is to show how to generate specific I^2C transactions using the I2C1 module on a dsPIC® Digital Signal Controller (DSC). The configuration required for I^2C Master mode is explained, as well as some of the specific details of the I^2C protocol. The focus is to provide the designer with a strong understanding of communication with the 24XXX series serial EEPROMs using the I2C1 module and I^2C, thus allowing for more complex programs to be written in the future. The firmware was written in C language using the C30 compiler. The code can easily be modified to use the Second I^2C module if available.

No additional libraries are required with the provided code. The firmware consists of two .c files (main.c, i2c_Func.c and i2c.h), organized into nine sections:

- Initialization
- Low Density Byte Write
- Low Density Byte Read
- Low Density Page Write
- Low Density Sequential Read
- High Density Byte Write
- High Density Byte Read
- High Density Page Write
- High Density Sequential Read

The program also includes the Acknowledge polling feature for detecting the completion of write cycles after the byte write and page write operations. Read operations are located directly after each write operation, thus allowing for verification that the data was properly written. No method of displaying the input data is provided, but an oscilloscope or a Microchip MPLAB® ICD 2 could be used.

The code was tested using the 24LC256 serial EEPROM. This device features 32K x 8 (256 Kbit) of memory and 64-byte pages. The 24LC256 also features a configurable, 3-bit address via the A2, A1, and A0 pins. For testing, these pins were all grounded for a chip address of '000'. Oscilloscope screen shots are labeled for ease in reading. The data sheet version of the waveforms are shown below the oscilloscope screen shots. All timings are designed to meet the 100 kHz specs, and an 8 MHz crystal oscillator is used to clock the dsPIC DSC. If a faster clock is used, the code must be modified for the I2Cx module to generate the correct clock frequency. All values represented in this application note are hex values unless otherwise noted. The firmware for this application note was developed using the Explorer 16 Development Board with the dsPIC33FJ256GP710 device.

INITIALIZATION

In order to configure the I^2C module for I^2C Master mode, several key registers on the dsPIC33F need to be properly initialized. Code examples are shown for each.

I2C1 Baud Rate Register (I2C1BRG)

The I2C1BRG acts as the Baud Rate Generator (BRG) reload value. Equation 1 shows how to calculate the value for I2C1BRG, based on a desired bit rate and a known FOSC. In testing the example code provided, a 8 MHz crystal oscillator was used, and the target bit rate was 100 kHz. Therefore, I2C1BRG needed to be set to 0x004f. Example 1 shows how to do this in code. Please consult the dsPIC33F data sheet for a greater explanation of this.

EQUATION 1: I2C1BRG CALCULATION

$$I2C1BRG = \left(\frac{Fcy}{Fscl} - \frac{Fcy}{1,111,111}\right)$$

EXAMPLE 1: I2C1BRG CONFIGURATION

```
I2C1BRG = 0x004f;  //Set BAUD rate
```

I2C1 Control Register (I2C1CON)

I2C1CON is one of the Configuration registers for the I2C1 module. The I2C1CON register is a 16-bit register and the individual bits are not covered in the application note. The I2C1 module was configured for Master mode, the Slew Rate control was disabled and the peripheral was not activated until all the configuration of the module was completed. This register is configured using the code shown in Example 2.

EXAMPLE 2: I2C1CON CONFIGURATION

```
I2C1CON = 0x1200; //Enable I2C1
```

PORTG I/O Configuration

In the dsPIC33F device the I2Cx module switches the I/O pins to alternate functions when the module is enabled. No additional configuration of the TRIS register is required.

BYTE WRITE

The Byte Write operation has been broken down into the following components: the Start condition and control byte, the word address, and the data byte and Stop condition. Note that, due to the size of the 24LC256, two bytes are used for the word address. However, 16 Kb and smaller 24XXX series devices use only a single byte for the word address.

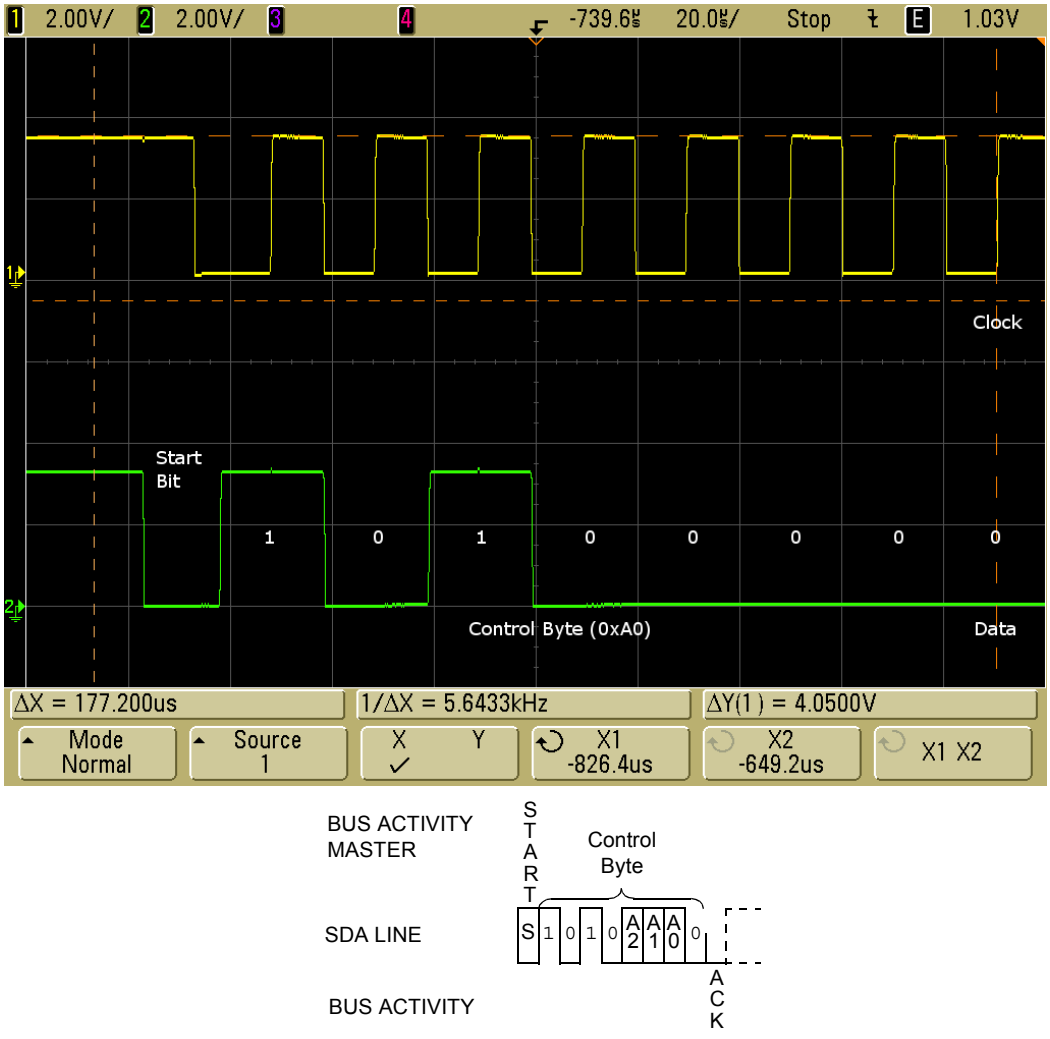
All I^2C commands must begin with a Start condition. This consists of a high-to-low transition of the SDA line while the clock (SCL) is high. After the Start condition, the 8 bits of the control byte are clocked out, with data being latched in on the rising edge of SCL. The device code (0xAh for the 24LC256), the block address (3 bits), and the R/$\overline{W}$ bit make up the control byte. Next, the EEPROM device must respond with an Acknowledge bit by pulling the SDA line low for the ninth clock cycle.

After the Start bit has been sent, the control byte can be transmitted. To do so, simply write the control byte to I2C1TRN. The I2C1 module will automatically begin transferring the data to the EEPROM device. The module will also detect whether or not the device responded with an ACK bit, and will set the ACKSTAT bit (I2C1STATbits.ACKSTAT) accordingly.

Start Bit and Control Byte Transmission

Figure 2 shows the details of the Start condition and the control byte. The left marker shows the position of the Start bit, whereas the right marker shows the ACK bit.

FIGURE 2: START BIT AND CONTROL BYTE



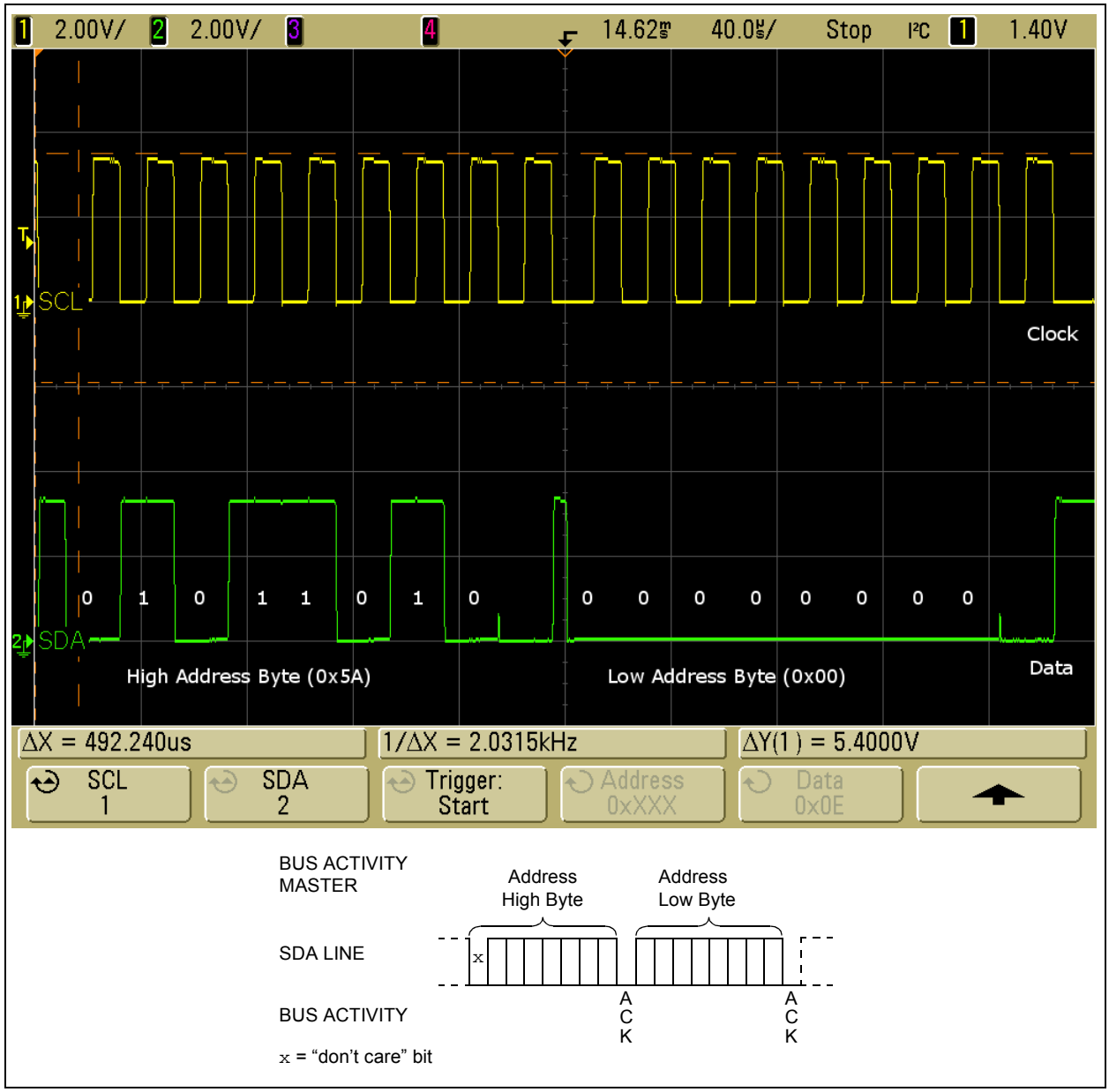
Sending the Word Address

After the EEPROM device has acknowledged receipt of the control byte, the master (dsPIC33F) begins to transmit the word address. For the 24LC256, this is a 15-bit value, so two bytes must be transmitted for the entire word address (the MSb of the high byte is a “don’t care”), with the Most Significant Byte sent first (note that 16 Kb and smaller 24XXX series devices only use a 1-byte word address). These bytes can be sent via the I2C1 module using the same method described above for the control byte.

After each byte of the word address has been transmitted, the device must respond with an Acknowledge bit.

Figure 3 shows the two address bytes and corresponding ACK bits. For reference, the previous ACK bit (in response to the control byte) is shown by the left marker. Note that the word address chosen for this application note is 0x5A00.

FIGURE 3: WORD ADDRESS

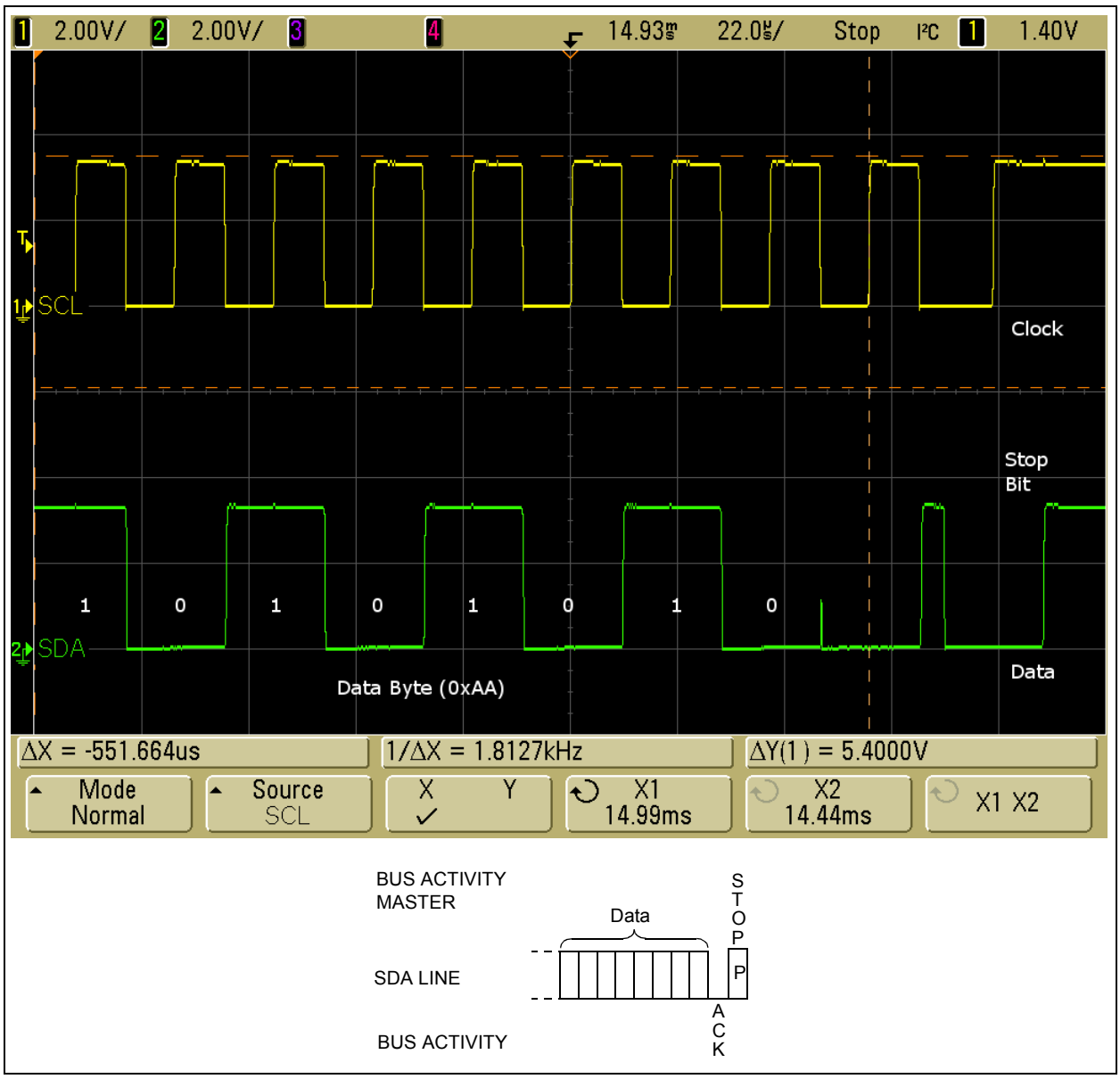


Data Byte and Stop Bit Transmission

Once the word address has been transmitted and the last ACK bit has been received, the data byte can be sent. Once again, the EEPROM device must respond with another ACK bit. After this has been received, the master generates a Stop condition. This consists of a low-to-high transition of SDA while the clock (SCL) is high. Initiating a Stop condition using the I2C1 module is similar to initiating a Start condition, except that the PEN bit (I2C1CONbits.PEN) is used for the Stop condition.

Figure 4 shows the transmission of the data byte, as well as the Stop condition indicating the end of the operation. The right marker denotes the Stop condition.

FIGURE 4: DATA BYTE AND STOP BIT



ACKNOWLEDGE POLLING

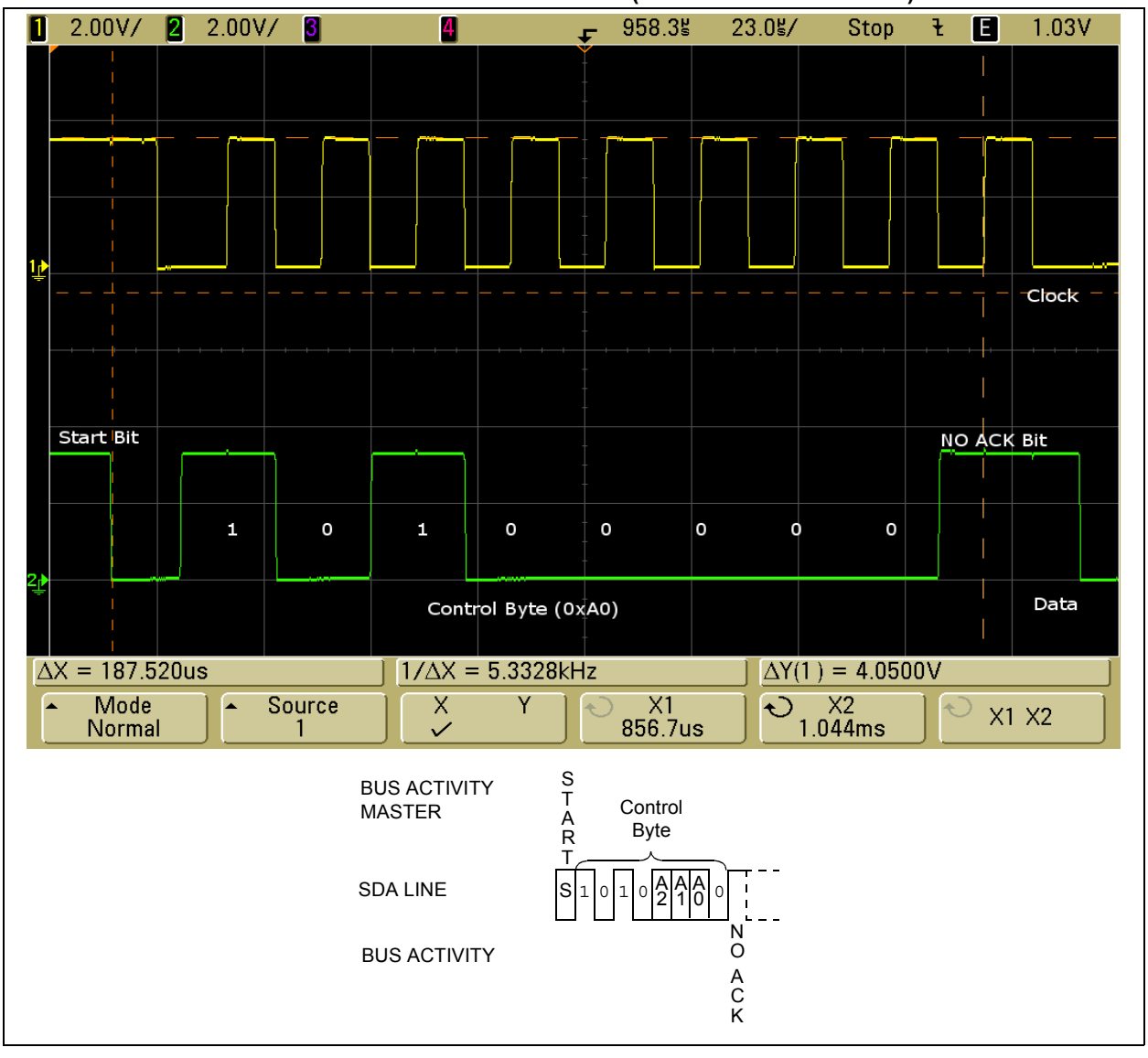
The data sheets for the 24XXX series devices specify a write cycle time (TWC), but the full time listed is not always required. Because of this, using a measured write cycle delay is not always accurate, which leads to wasted time. Therefore, in order to transfer data as efficiently as possible, it is highly recommended to use the Acknowledge Polling feature. Since the 24XXX series devices will not acknowledge during a write cycle, the device can continuously be polled until an Acknowledge is received. This is done after the Stop condition takes place to initiate the internal write cycle of the device.

Acknowledge Polling Routine

The process of acknowledge polling consists of sending a Start condition and then a Write command to the EEPROM device, then simply checking to see if the ACK bit was received via the ACKSTAT bit. If it was not received (i.e., if ACKSTAT is high), then the device is still performing its write cycle.

Figure 5 shows an example of acknowledge polling to check if a write operation has finished. In this example, the device did not acknowledge the poll (the ACK bit is high), which indicates that the write cycle has not yet completed.

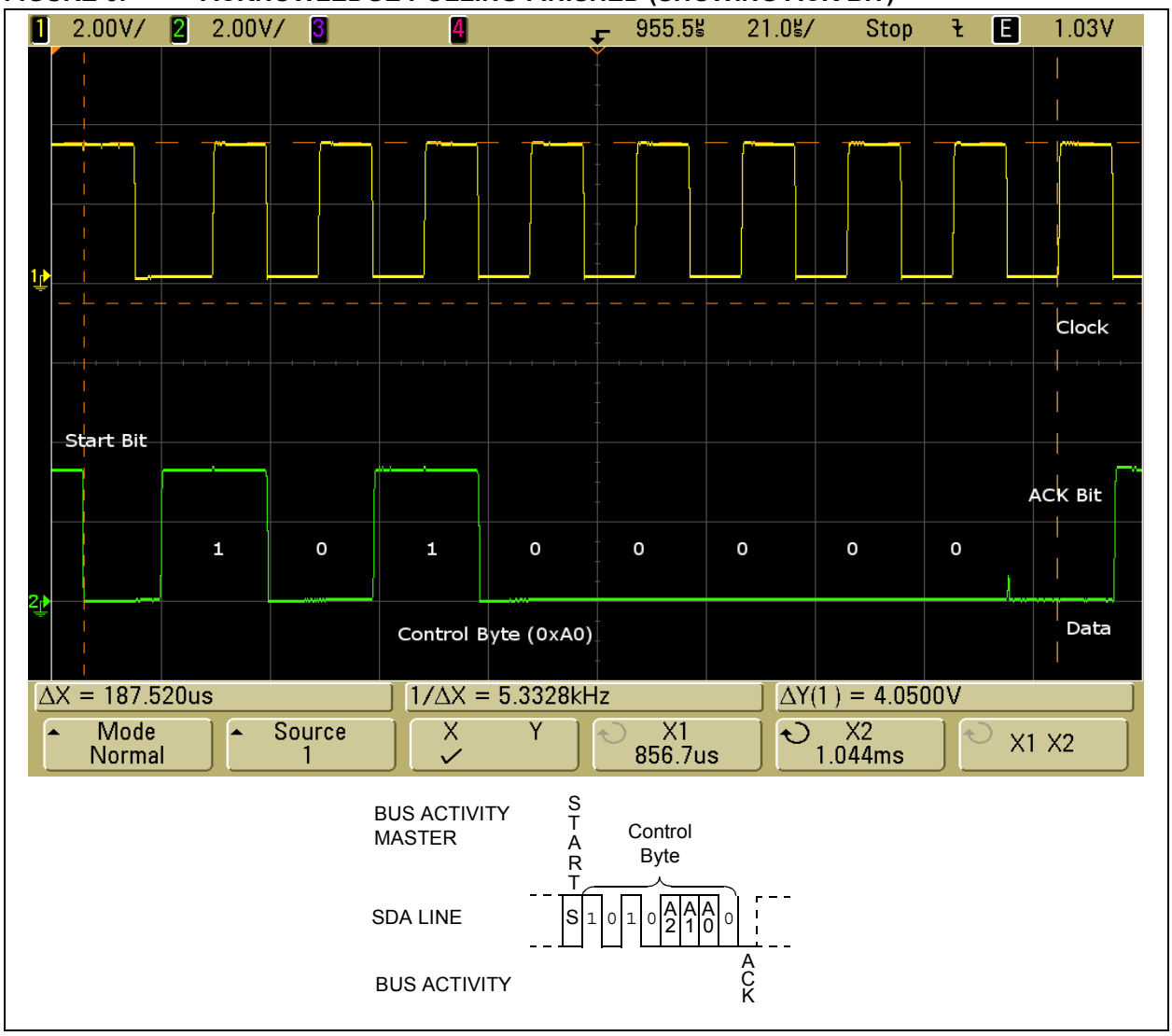
FIGURE 5: ACKNOWLEDGE POLLING ROUTINE (SHOWING NO ACK BIT)



Response to Acknowledge Polling

Figure 6 shows the final acknowledge poll after a write operation, in which the device responds with an ACK bit, indicating that the write cycle has completed and the device is ready to continue.

FIGURE 6: ACKNOWLEDGE POLLING FINISHED (SHOWING ACK BIT)



BYTE READ

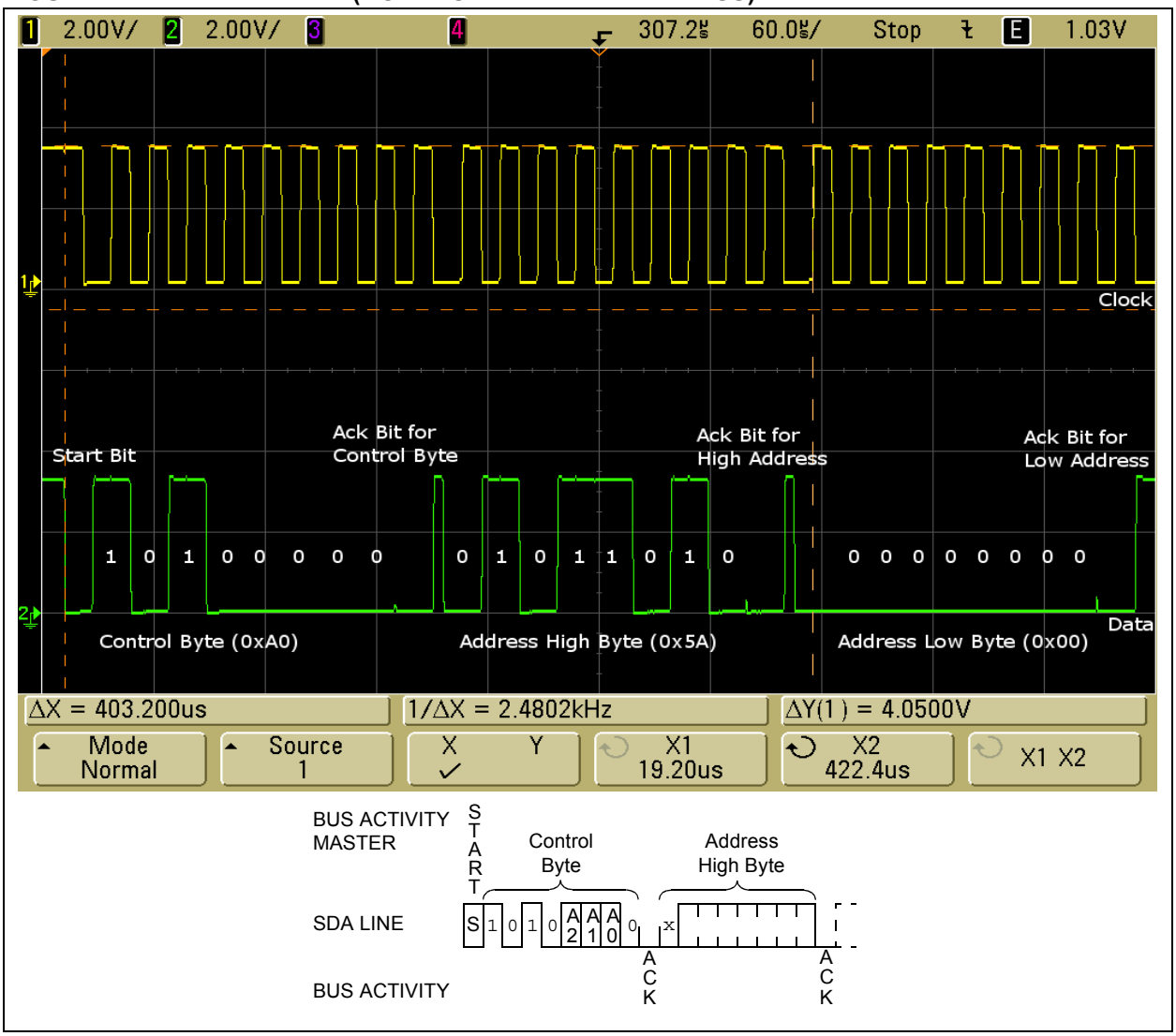
The byte read operation can be used in order to read data from the 24XXX series devices in a random access manner. It is similar to the byte write operation, but slightly more complex. The word address must still be transmitted, and to do this, a control byte with the R/$\overline{W}$ bit set low must be sent first. However, this conflicts with the desired operation, that is, to read data. Therefore, after the word address has been sent, a new Start condition and a control byte with R/$\overline{W}$ set high must be transmitted. Note that a Stop condition is not generated after sending the word address.

Using the I2C1 module, transmitting the first control byte and the word address is done in the same fashion as for a byte write.

Writing Word Address for Read

Figure 7 shows an example of the first control byte and the word address of a byte read operation. The left marker indicates the Start bit and the right marker indicates the ACK bit after receipt of the word address (0x5A00 in this example). Once again, the R/$\overline{W}$ bit must be low in order to transmit the word address.

FIGURE 7: BYTE READ (CONTROL BYTE AND ADDRESS)



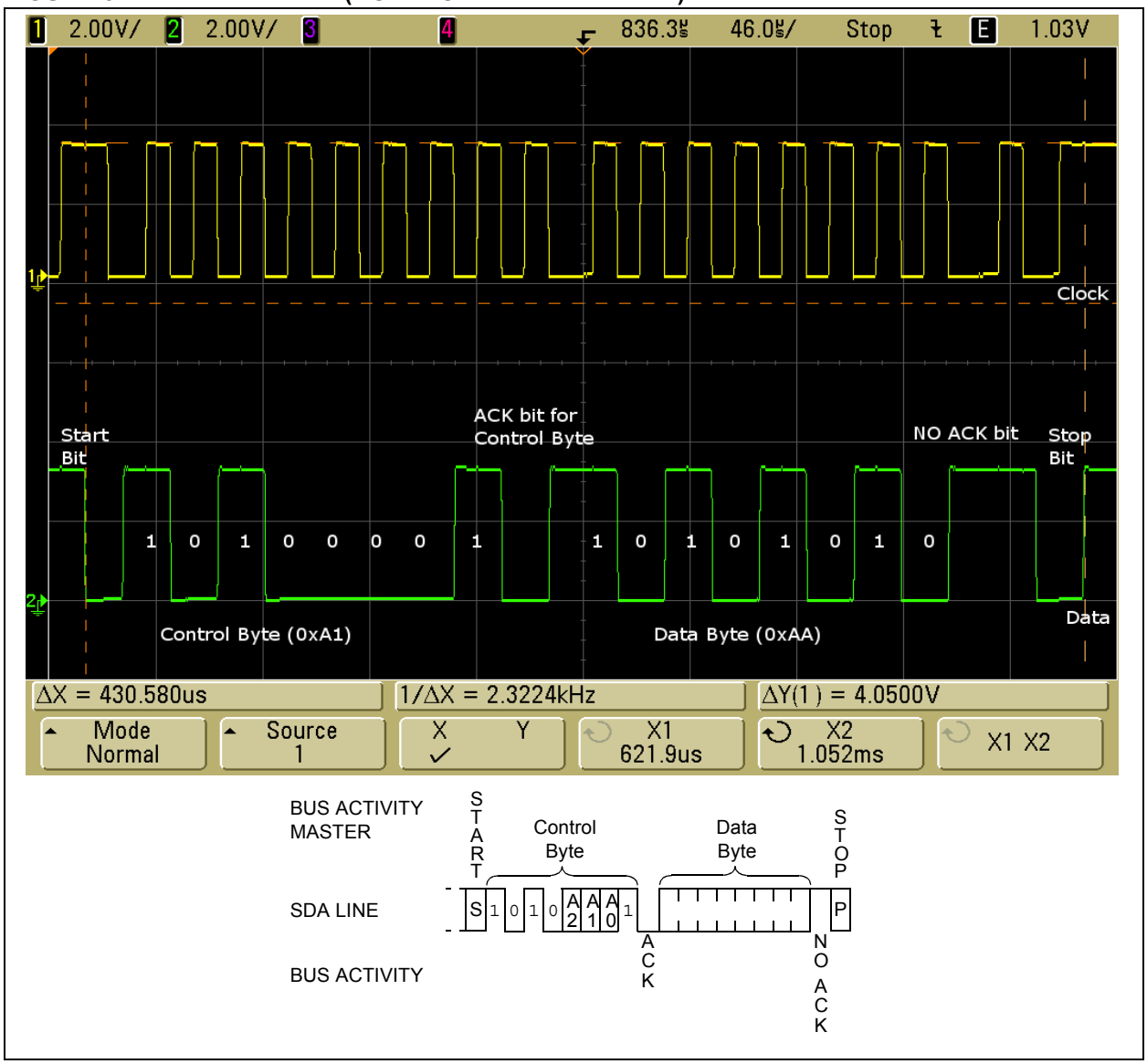
Reading Data Byte Back

After the word address has been transmitted, the Restart Enable bit (I2C1CONbits.RSEN) is used to initiate a Restart condition. Note that a Restart is very similar to a Start, except that a Restart does not first check for a valid bus condition (this is important since either SCL or SDA may be low at this point, which would cause an error during an attempted Start condition). The second control byte (with the R/$\overline{W}$ bit set) is then transmitted as normal.

In order to read the data byte, the ACKDT bit is first set to indicate that a NO ACK should be sent. Then the RCEN bit is set to initiate the read and the data byte can be copied from I2C1RCV. Once the data byte has been read back from the 24XXX series device, the master must respond back with a NO ACK bit. To do this, the ACKEN bit is set, sending out the NO ACK bit. This indicates to the device that no more data will be read. Finally, the master generates a Stop condition to end the operation.

Figure 8 shows the control byte and data byte during the actual read part of the operation. A Restart condition is generated immediately after receipt of the previous ACK bit and is marked with the left marker. At the end of the transfer, the master indicates that no more data will be read by the use of the NO ACK bit (holding SDA high in place of an ACK bit); this is shown by the right marker. After the NO ACK bit has been sent, the master generates a Stop condition to end the operation.

FIGURE 8: BYTE READ (CONTROL BYTE AND DATA)



PAGE WRITE

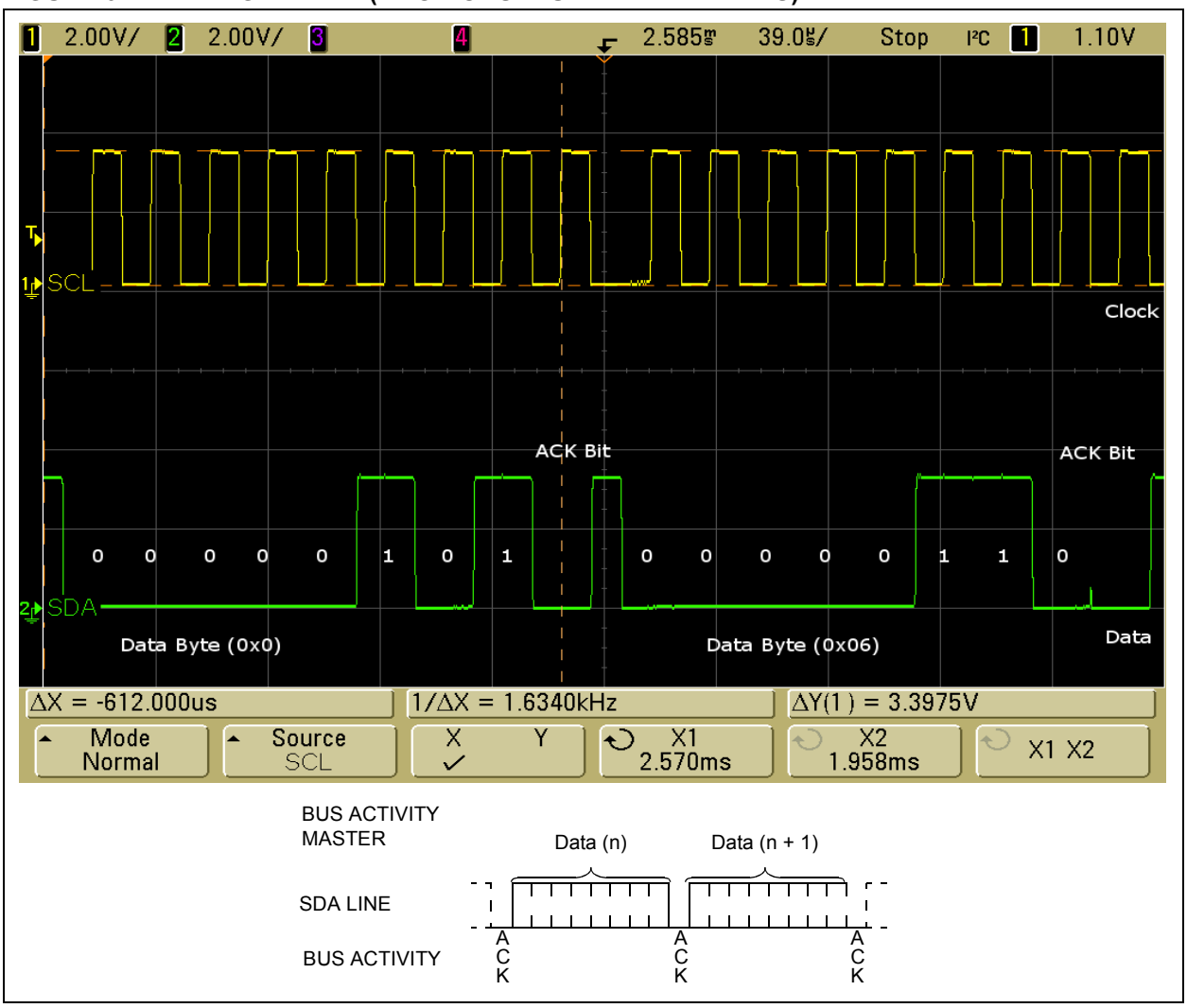
A very useful method for increasing throughput when writing large blocks of data is to use page write operations. All of the 24XXX series devices, with the exception of the 24XX00, support page writes, and the page size varies from 8 bytes to 128 bytes. Using the page write feature, up to 1 full page of data can be written consecutively with the control and word address bytes being transmitted only once. It is very 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 which are integer multiples of the page size, and end at addresses which are [integer multiples of the page size] minus 1. Any attempts to write across a page boundary will 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 Stop condition after the first data byte has been transmitted, the master continues to send more data bytes, up to 1 page total. The 24XXX will automatically increment the internal Address Pointer with receipt of each byte. As with the byte write operation, the internal write cycle is initiated by the Stop condition.

Sending Multiple Bytes Successively

Figure 9 shows two consecutive data bytes during a page write operation. The entire transfer cannot be shown legibly due to length, but this screen shot shows the main difference between a page write and a byte write. Notice that after the device acknowledges the first data byte (0x05 in this example), the master immediately begins transmitting the second data byte (0x06 in this example).

FIGURE 9: PAGE WRITE (TWO CONSECUTIVE DATA BYTES)



SEQUENTIAL READ

Just as the page write operation exists to allow for more efficient write operations, the sequential read operation exists to allow for more efficient read operations. While the page write is limited to writing within a single physical page, the sequential read operation can read out the entire contents of memory in a single operation.

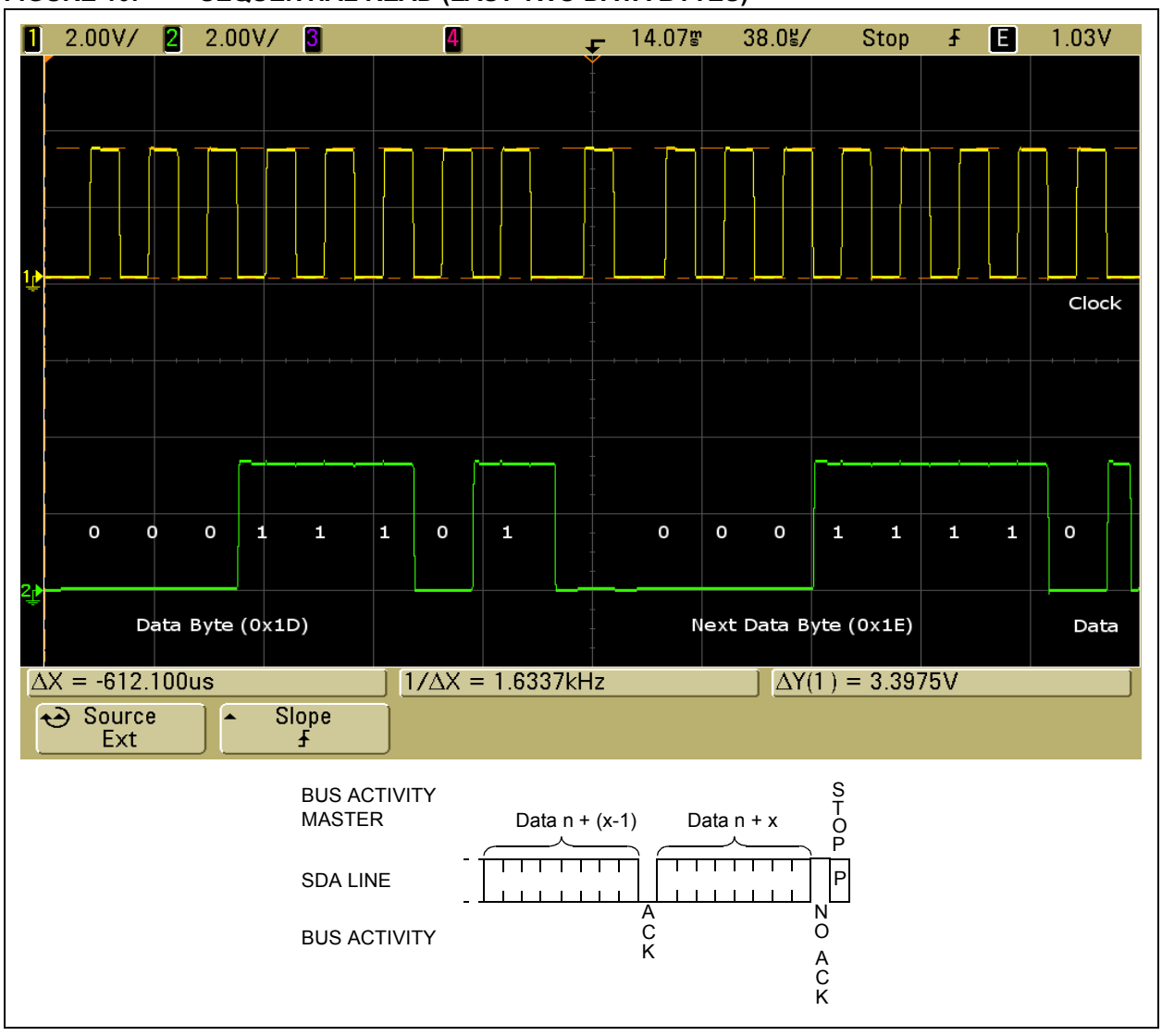
The sequential read operation is very similar to the byte read operation, except that the master must pull SDA low after receipt of each data byte to send an Acknowledge bit back to the 24XXX series device. This ACK bit indicates that more data is to be read. As long as this ACK bit is transmitted, the master can continue to read back data without the need for generating Start/Stop conditions or for sending more control/word address bytes.

In order to do this with the MSSP module, the ACKDT bit must be properly set up before initiating the Acknowledge via the ACKEN bit. Clearing the ACKDT bit produces an ACK bit, whereas setting the ACKDT bit produces a NO ACK bit.

Reading Data Bytes Successively

Figure 10 shows the last two bytes of a 16-byte sequential read operation. Note that the master pulls SDA low to transmit an ACK bit after the first data byte, but leaves SDA high to transmit a NO ACK bit after the final data byte. And as with all other operations, a Stop condition is generated to end the operation.

FIGURE 10: SEQUENTIAL READ (LAST TWO DATA BYTES)



CONCLUSION

When communicating with the 24XXX series EEPROM devices, there are many benefits of using the dsPIC33F I2Cx module over bit-banging through software. The designer does not have to be familiar with the I^2C timing specifications, nor is the designer required to write full software routines to provide I^2C functionality. This results in much shorter development time.

This application note illustrated the main characteristics of I^2C communications with Microchip's 24XXX series serial EEPROM devices with the use of the dsPIC33F I2Cx module. The C30 Code provided is highly portable and can be used with only minor modifications on the PIC24 family of microcontrollers and dsPIC30F DSCs equipped with the I^2C module. The code was tested on Microchip's Explorer 16 Demonstration Board with the connections shown in Figure 1.

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.

Trademarks

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 Mountain View, California. 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
Habour 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 - Fuzhou
Tel: 86-591-8750-3506
Fax: 86-591-8750-3521

China - Hong Kong SAR
Tel: 852-2401-1200
Fax: 852-2401-3431

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 - Shunde
Tel: 86-757-2839-5507
Fax: 86-757-2839-5571

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

China - Xian
Tel: 86-29-8833-7250
Fax: 86-29-8833-7256

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 - Gumi
Tel: 82-54-473-4301
Fax: 82-54-473-4302

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

Malaysia - Penang
Tel: 60-4-646-8870
Fax: 60-4-646-5086

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

12/08/06