

# AN1488

### Bit Banging I<sup>2</sup>C<sup>TM</sup> on Mid-Range MCUs with the XC8 C Compiler

Author: Eugen Ionescu Microchip Technology Inc.

### INTRODUCTION

The 24XXXX series serial EEPROMs from Microchip Technology are  $I^2C^{TM}$  compatible and feature maximum clock frequencies ranging from 100 kHz up to 1 MHz. Many times when designing an application which utilizes a serial EEPROM device, the dedicated protocol-specific serial port is in use by other parts of the design. In these instances, it is required of the designer to write software routines capable of generating the proper signals for communicating with the EEPROM device.

The  $I^2C$  bit banging is a technique for serial communications using software instead of a dedicated hardware module. This means that the code controls the state of the MCU pins, related to all parameters of the signals: timing, levels and synchronization. This application note is intended to serve as a reference for communicating with Microchip's 24XXXX series serial EEPROM devices, without relying on a hardware serial port to handle the I<sup>2</sup>C operations. Source code for common data transfer modes is also provided.

Figure 1 and Figure 2 show the hardware schematics for the interface between Microchip's 24XXXX devices and PIC16LF1947/PIC12F1822 microcontrollers. The schematics show the connections necessary between the microcontroller and the serial EEPROM, and the software was written assuming these connections. The SDA pin is an open-drain terminal, and therefore requires a pull-up resistor to Vcc (typically 10 k $\Omega$  for 100 kHz and 2 k $\Omega$  for 400 kHz and 1 MHz). Also, the A0, A1 and A2 pins are tied to ground because these set the address of the I<sup>2</sup>C EEPROM. The WP pin is grounded to allow writing to the EEPROM.



### FIGURE 1: CIRCUIT FOR PIC16LF1947 AND 24XXXX SERIES DEVICE

## AN1488





The firmware was written in C, compiled with XC8 C

Compiler and tested using a PICDEM<sup>™</sup> PIC18

Explorer Demonstration Board with a Microchip

24AA256 EEPROM device, PIC16LF1947 PIM (for

PIC16 source code) and PIC12F1822 (for PIC12

The I<sup>2</sup>C and EEPROM drivers are described in Table 1

source code) mounted on it.

and Table 2.

### FIRMWARE DESCRIPTION

The firmware shows how to generate specific  $I^2C$  transactions through the software using a PIC12/16 microcontroller. The specific details of the  $I^2C$  protocol are also shown and discussed, thus providing the building blocks for writing more complex programs later on.

The code is designed around the implementation of several steps to complete an  $I^2C$  communication session:

- Send the start bit to the device.
- Send the required command byte(s).
- Send the address byte(s).
- Send or receive the data byte.
- Send the Stop bit to the device.

Function	Parameters	Data Returned	Description
eeprom_wr_byte	EEPROM address (unsigned long), data to be written (unsigned char).	None	Write a byte to the EEPROM
eeprom_rd_byte	EEPROM address from where data will be read (unsigned long).	Data - unsigned char	Read a byte from the EEPROM
eeprom_wr_page	EEPROM starting address (unsigned long), the data array (unsigned char *) and the number of bytes which will be written (unsigned int).	None	Write multiple byte to the EEPROM, regardless page boundary
eeprom_rd_page	EEPROM starting address (unsigned long), the data array where the received values from the EEPROM will be written (unsigned char *) and the number of bytes that will be read from the slave (unsigned int).	None	Sequential read – read multiple bytes from the EEPROM

### TABLE 1: EEPROM DRIVERS

### TABLE 2:I<sup>2</sup>C BASIC DRIVERS

Function	Parameters	Data returned	Description
i2c_start	None	None	Generate I <sup>2</sup> C <sup>™</sup> Start condition
i2c_stop	None	None	Generate I <sup>2</sup> C <sup>™</sup> Stop condition
bit_out	Data to be outputed (unsigned char)	None	Output a bit to the I <sup>2</sup> C™ bus
bit_in	Pointer where data will be written (unsigned char *)	None	Input a bit from the I <sup>2</sup> C™ bus
i2c_wr	Data to be written (unsigned char)	ACK bit	Write a byte to the l <sup>2</sup> C™ bus
i2c_rd	ACK bit	Data received from the slave	Read a byte from the I <sup>2</sup> C™ bus
ack poll	ACK status (unsigned char)	None	Poll for ACK bit

Functions are provided for both low density (<= 2 Kb, 1 address byte) and high density (>2 Kb, 2 address byte) serial EEPROM devices (see the comments inside the "I2C\_EEPROM.c" file). The high-density versions are illustrated in the examples provided.

The program also exhibits the Acknowledge polling feature for detecting the completion of write cycles after the byte write and page write operations. Data is read after each write to verify if it was properly written. For read/write byte operations, the input data is displayed on the PICDEM PIC18 Explorer onboard LEDs (only for PIC16 code). For read/write byte and read/write page operations, the data can be seen using the MPLAB<sup>®</sup> X debugging feature (the output and input data can be displayed into the "Variables" window). By default, the firmware writes a byte to the EEPROM device and reads it back (display the received data on LEDs). The page write/read operations are commented in the main ("PIC16 I2C BITBANG EEPROM.c" files and "PIC12 I2C BITBANG EEPROM.c"). Figure 3 and Figure 4 show a flowchart for the byte/page read/write operations. Since PIC12F1822 has a reduced number of pins, the LEDs are replaced with a software flag ("test" variable).

The code was tested using a 24AA256 serial EEPROM. This device features 32k x 8 (256 Kbit) of memory in 64-byte pages. The oscilloscope screenshots are labeled for ease in reading. The data sheet versions of the waveforms are shown below the oscilloscope screenshots. A 10 MHz crystal oscillator is used to clock the PIC16LF1947. For the PIC12F1822 MCU, the internal 8 MHz oscillator is used. All values represented in this application note are decimal values unless otherwise noted.

### FIGURE 3: BYTE WRITE/READ OPERATION FLOWCHART





### **BYTE WRITE**

The byte write operation has been broken down into the following components: the Start condition and control byte, the address bytes (only one address byte for low-density devices), and the data byte and Stop condition. All I<sup>2</sup>C 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 eight bits of the control byte are clocked out to the EEPROM, with data being latched in on the rising edge of SCL. The device code (0xA0 for the 24AA256), the block address (three bits) and the R/W bit make up the control byte. Next, the EEPROM device must respond with an ACK bit by pulling the SDA line low for the ninth clock cycle. A byte write operation for a high-density device (>2 kb, 2 address bytes) is described in Figure 5.

### FIGURE 5: BYTE WRITE OPERATION FLOWCHART



The low-density devices (<= 2 kb) have only one address byte.

Figure 6 shows the control byte structure for the 24AA256 EEPROM device.

### FIGURE 6: CONTROL BYTE STRUCTURE



### Start Bit and Control Byte Transmission

Figure 7 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. Note that the SDA line defaults to a high value due to the attached pull-up resistor. Therefore, SDA only goes low when transmitting a '0'.



FIGURE 7: START BIT AND CONTROL BYTE

### Sending the Word Address

After the EEPROM device has acknowledged receipt of the control byte, the master begins to transmit the word address. After this operation, the device must respond with another ACK bit. Figure 8 shows the address bytes and corresponding ACK bits.





Figure 9 shows the transmission of the data byte, as well as the Stop condition indicating the end of the

operation. Again, the left marker shows the previous

ACK bit (that of the word address). The right marker

denotes the Stop condition.

### Data Byte and Stop Bit Transmission

Once the word address has been transmitted and the ACK bits have 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 is achieved by creating a low-to-high transition of SDA while the clock (SCL) is high.

FIGURE 9: DATA BYTE AND STOP BIT



### ACKNOWLEDGE POLLING

The data sheets for the 24XXXX series devices specify a write cycle time (Twc), but the write may complete in less time. Therefore, in order to transfer data as quickly as possible, it is highly recommended to use the Acknowledge Polling feature. Since the 24XXXX 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 followed by a Write command to the EEPROM device, then simply checking to see if an ACK bit is received. If the EEPROM does not generate an ACK, then the device is still performing its write cycle or isn't able to respond for another reason.

Figure 10 and Figure 11 show 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.

Note: NO ACK means that the device does NOT generate an ACK bit. This can happen because the device has received an illegal operation (wrong address or instruction), it is not ready or not present. When the device does not generate an ACK bit, the SDA line is released high. In case of a NO ACK, the firmware stays inside a loop ("ack\_poll()" function), polling for an ACK bit. This function is used only at the end of the communication (after master sends Stop bit).

### FIGURE 10: ACKNOWLEDGE POLLING FLOWCHART





FIGURE 11: ACKNOWLEDGE POLLING ROUTINE (SHOWING NO ACK BIT)

### **Response to Acknowledge Polling**

Figure 12 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 12: ACKNOWLEDGE POLLING FINISHED (SHOWING ACK BIT)

### PAGE WRITE

The memory is organized in fixed-length contiguous blocks, called pages (see Figure 13).

Another method for increasing throughput when writing large blocks of data is to use page write operations. Using the page write feature, up to a 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 (Figure 13). Any attempts to write across a page boundary will result in the address wrapping back to the beginning of the current page, thus overwriting any previously stored data 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 a whole page. The 24XXXX 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.

This firmware allows writing from any address ( $\leq$  0x7FFF, the maximum memory array address for 24AA256 (256 Kbits) device) as many bytes as the user wants. The write cycle time and the address updating is automatically done by drivers ( $I^2C\_EEPROM.c$  file), avoiding the overwriting operation. The related routine can be used for sequential write operations. Figure 14 shows a flowchart for the "eeprom\_wr\_page()" function.





# AN1488



### Sending Multiple Bytes Successively

Figure 15 shows two consecutive data bytes during a page write operation. The entire transfer cannot be shown legibly due to length, but this screenshot shows the main difference between a page write and a byte write. Notice that after the device acknowledges the first data byte (0x00 in this example), the master immediately begins transmitting the second data byte (0x01 in this example). The <code>eeprom\_wr\_page()</code>

function completes all of the necessary math to check if the entire data array can be written on the page with the desired address, to split it into sub-arrays, to wait for the write cycle time and to increment the address for not causing a page boundary. The MPLAB X "Variables" window can be used to check "send" and "receive" vectors. In addition, if the input values do not match with the output ones, the firmware turns the LEDs on.

FIGURE 15: PAGE WRITE (TWO CONSECUTIVE DATA BYTES)



### BYTE READ

In order to read data from the 24XXXX series device in a random access manner, the byte read operation can be used. It is similar to the byte write operation, but with additional steps. 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. This conflicts with the current operation, that is, to read data because the desired address to be read must be written to the EEPROM before data can be read. 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.

After the data byte has been read from the 24XXXX device, the master must NOT generate an ACK bit, that is, leaving the SDA line high in place of an 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 16 shows a flowchart for the byte read operation.

### Writing Word Address for Read

Figure 17 shows an example of the first control byte and 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 (0x3E in this example). Once again, the R/W bit must be low in order to transmit the word address.

### FIGURE 16: BYTE READ OPERATION FLOWCHART





### **Reading A Data Byte**

Figure 18 shows the control byte and data byte during the actual read part of the operation. A new Start 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 NOT generating an ACK bit (releasing the SDA line); 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 18: BYTE READ (CONTROL BYTE AND DATA)

### 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 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 ACK bit back to the 24XXXX 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/address bytes. A flowchart for the sequential read operations is shown in Figure 19.

FIGURE 19: SEQUENTIAL READ OPERATION FLOWCHART



### **Reading Data Bytes Successively**

Figure 20 shows the last two bytes of a 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. As with all other operations, a Stop condition is generated to end the operation.





### CONCLUSION

This application note illustrates the main characteristics of I<sup>2</sup>C communications with Microchip's 24XXXX series serial EEPROM devices, focusing primarily on the 24AA256. The C code provided is highly portable and can be used on many PIC12/16 family microcontrollers with only minor modifications. The code was tested on Microchip's PICDEM PIC18 Explorer Demo Board with the connections shown in Figure 1 and Figure 2.

# AN1488

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.

### QUALITY MANAGEMENT SYSTEM CERTIFIED BY DNV = ISO/TS 16949=

#### Trademarks

The Microchip name and logo, the Microchip logo, dsPIC, FlashFlex, KEELOQ, KEELOQ logo, MPLAB, PIC, PICmicro, PICSTART, PIC<sup>32</sup> logo, rfPIC, SST, SST Logo, SuperFlash and UNI/O are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

FilterLab, Hampshire, HI-TECH C, Linear Active Thermistor, MTP, SEEVAL and The Embedded Control Solutions Company are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Silicon Storage Technology is a registered trademark of Microchip Technology Inc. in other countries.

Analog-for-the-Digital Age, Application Maestro, BodyCom, chipKIT, chipKIT logo, CodeGuard, dsPICDEM, dsPICDEM.net, dsPICworks, dsSPEAK, ECAN, ECONOMONITOR, FanSense, HI-TIDE, In-Circuit Serial Programming, ICSP, Mindi, MiWi, MPASM, MPF, MPLAB Certified logo, MPLIB, MPLINK, mTouch, Omniscient Code Generation, PICC, PICC-18, PICDEM, PICDEM.net, PICkit, PICtail, REAL ICE, rfLAB, Select Mode, SQI, Serial Quad I/O, Total Endurance, TSHARC, UniWinDriver, WiperLock, ZENA and Z-Scale 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.

GestIC and ULPP are registered trademarks of Microchip Technology Germany II GmbH & Co. KG, a subsidiary of Microchip Technology Inc., in other countries.

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

© 2013, Microchip Technology Incorporated, Printed in the U.S.A., All Rights Reserved.

Printed on recycled paper.

ISBN: 9781620770559

Microchip received ISO/TS-16949:2009 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 mulfacture 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://www.microchip.com/ support

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

**Cleveland** Independence, OH Tel: 216-447-0464 Fax: 216-447-0643

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

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

Indianapolis Noblesville, IN Tel: 317-773-8323 Fax: 317-773-5453

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: 86-10-8569-7000 Fax: 86-10-8528-2104

**China - Chengdu** Tel: 86-28-8665-5511 Fax: 86-28-8665-7889

**China - Chongqing** Tel: 86-23-8980-9588 Fax: 86-23-8980-9500

**China - Hangzhou** Tel: 86-571-2819-3187 Fax: 86-571-2819-3189

**China - Hong Kong SAR** Tel: 852-2943-5100 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: 86-755-8864-2200 Fax: 86-755-8203-1760

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

**China - Xian** Tel: 86-29-8833-7252 Fax: 86-29-8833-7256

**China - Xiamen** Tel: 86-592-2388138 Fax: 86-592-2388130

**China - Zhuhai** Tel: 86-756-3210040 Fax: 86-756-3210049

### ASIA/PACIFIC

India - Bangalore Tel: 91-80-3090-4444 Fax: 91-80-3090-4123

**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 - Osaka** Tel: 81-6-6152-7160 Fax: 81-6-6152-9310

**Japan - Tokyo** Tel: 81-3-6880- 3770 Fax: 81-3-6880-3771

**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: 886-3-5778-366 Fax: 886-3-5770-955

Taiwan - Kaohsiung Tel: 886-7-213-7828 Fax: 886-7-330-9305

Taiwan - Taipei Tel: 886-2-2508-8600 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

11/29/12