AN3304

Interfacing PIC® MCUs with Single-Wire Serial EEPROMs

Author: *Erik Fasnacht*
Microchip Technology Inc.

INTRODUCTION

The AT21CS Series is a family of Serial Electrically Erasable and Programmable Read-Only Memory (EEPROM) that utilizes the Single-Wire Interface (SWI) protocol.

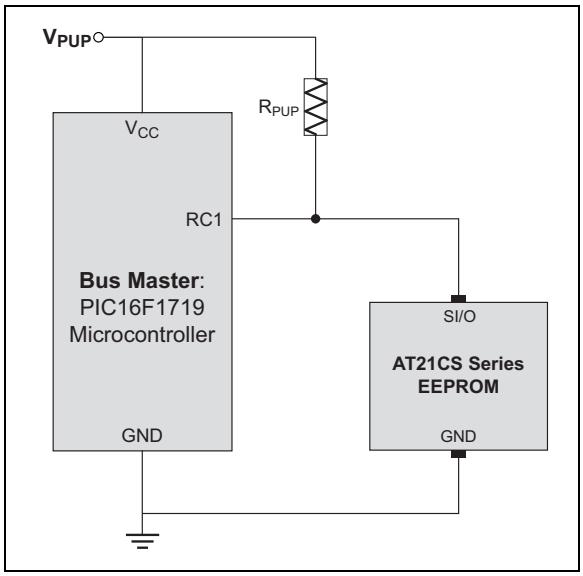
The family software addressing scheme allows up to eight devices to share a common single-wire bus. The device is optimized for use in many industrial and commercial applications where low-power and low-voltage operation are essential. Some applications examples include analog sensor calibration data storage, ink and toner printer cartridge identification, and management of after-market consumables. The family is available in space-saving package options and operates with an external pull-up voltage on the SI/O line.

HARDWARE

The hardware used in conjunction with the firmware is the DM160228 Explorer 8 Development Kit. For additional information about the hardware, refer to the DM160228 User Guide found at http://www.microchip.com.

Figure 1 is the hardware schematic that depicts the interface between the Microchip AT21CS Series of devices and the microcontroller, PIC16F1719. The schematic shows the necessary connections between the microcontroller and the serial EEPROM and the firmware was written assuming these connections. The single I/O connection between the microcontroller and the serial EEPROM includes a recommended pull-up resistor (R_{PUP}) and the hardware needs to supply the V_{PUP} to the pull-up resistor.

FIGURE 1: HARDWARE CIRCUIT



FIRMWARE

The purpose of the firmware is to show how to generate specific single-wire bus transactions using a generic I/O pin on the microcontroller. The focus is to provide the user with a strong understanding of communication with the AT21CS Series devices, thus allowing for more complex programs to be written in the future.

The firmware was written in C using MPLAB® X Integrated Development Environment (MPLAB X IDE) V5.20 and the code was generated using MPLAB® Code Configurator (MCC).

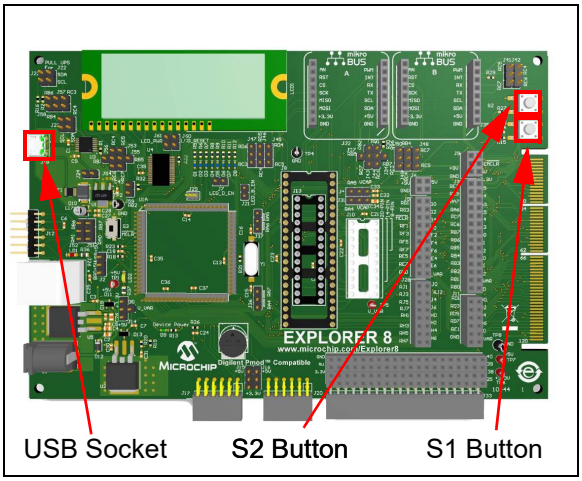
Most of the complex tasks have been done in the firmware and the user is not expected to write any low-level subroutines.

Oscilloscope screen shots of the firmware and hardware are shown in this application note to assist in better understanding single-wire bus transactions.

Overview

The firmware uses the Explorer 8 S1 and S2 buttons and the Micro-USB Socket for USB communication. The USB communication is controlled via UART and a serial port on the computer. The serial port will display messages for the transactions being performed and the data read back from the AT21CS Series device. The S2 button is used as a hardware breakpoint to start sending single-wire bus transactions and the S1 button is used to send subsequent permanent operations to the AT21CS Series device. Figure 2 highlights the location of the Explorer 8 Micro-USB socket and the S1 and S2 buttons.

FIGURE 2: EXPLORER 8 HARDWARE

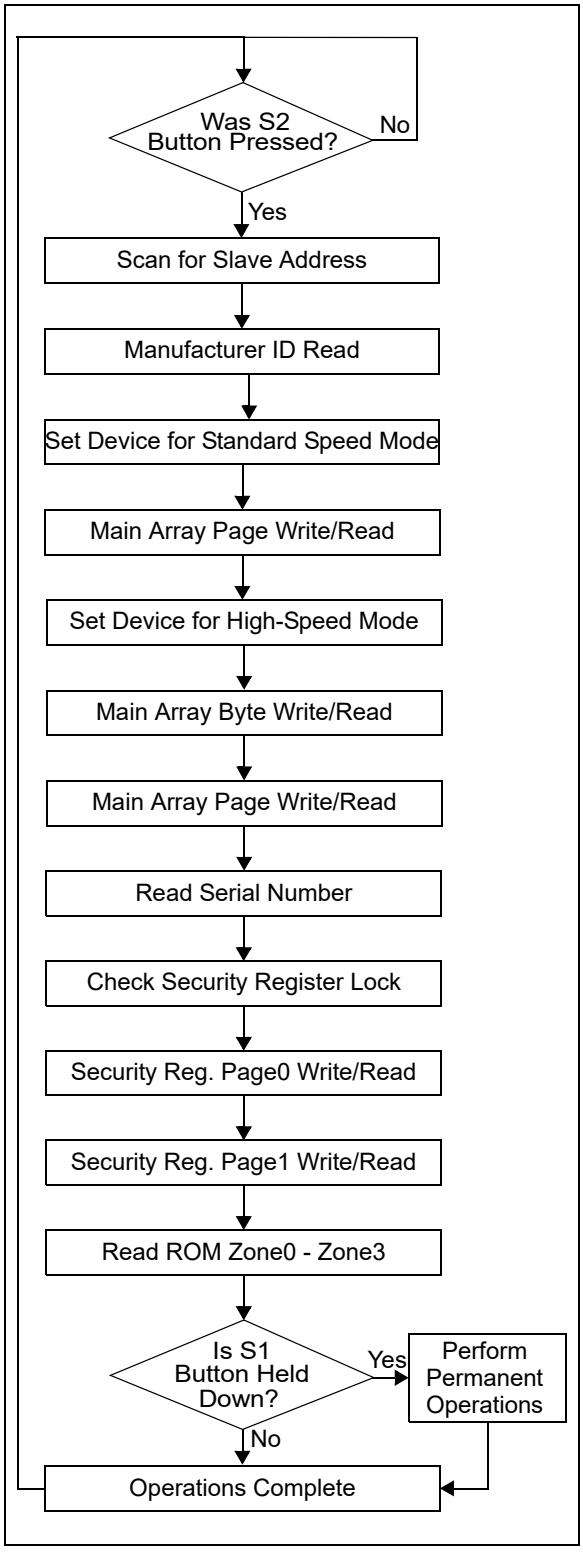


After the S2 button is pressed, the single-wire protocol will be sent to the slave device. The S1 button can be used in conjunction with the S2 button to send permanent operations to the AT21CS Series device that once completed, cannot be undone. Therefore, the firmware expects the S1 button to be held down when the S2 button is pressed in order to transmit the permanent operations. Below are the additional permanent operations.

- Lock Security Register
- Write-Protect ROMZone0 - ROMZone3
- Freeze ROM Zone State

The firmware also includes a scan for the corresponding slave address preprogrammed into the device. This is accomplished by performing a Reset and Discovery Response, followed by a device address byte. The device address byte will start with slave address 0 (000b) and will check whether the device ACKs the byte. If the device NACKs, the firmware will increment the slave address and perform the sequence again until the device ACKs. Once an ACK is detected, the firmware will recognize that slave address and that address will be used for the other operations. Figure 3 is a flowchart representing the single-wire bus transactions.

FIGURE 3: FLOW CHART



SINGLE-WIRE COMMUNICATION

Types of data transmitted over the SI/O line:

- Reset and Discovery Response
- Data Input
  - Logic '0' or Acknowledge (ACK)
  - Logic '1' or No Acknowledge (NACK)
- Data Output
  - Logic '0' or Acknowledge (ACK)
  - Logic '1' or No Acknowledge (NACK)
- Start and Stop Condition

Communication with the device is conducted in time intervals referred to as a bit frame and lasts t_{BIT} in duration. Each bit frame contains a single binary data value. Input bit frames are used to transmit data from the master to the slave device and can either be a logic '0' or a logic '1'. An output bit frame carries data from the slave device to the master. In all input and output cases, the master initiates the bit frame by driving the SI/O line low. Once the slave device detects the SI/O being driven below the V_{IL} max threshold, its internal timing circuits begin to run.

The duration of each bit frame is allowed to vary from bit to bit as long as the variation does not cause the t_{BIT} length to exceed the specified minimum and maximum values.

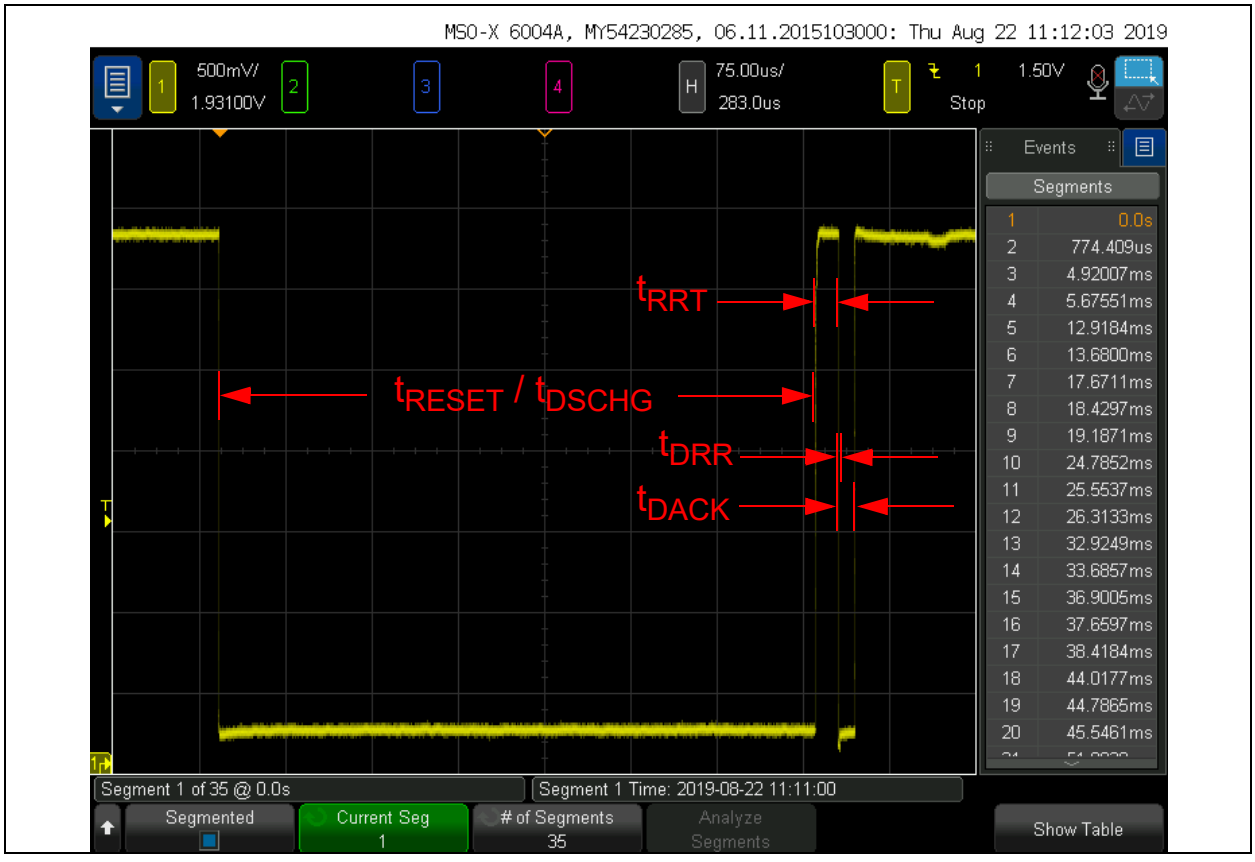
Note: The Reset and Discovery Response is not considered to be part of the data stream, whereas the remaining transactions are all required in order to send data to and receive data to and from the device. The difference between the types of data stream transactions is the duration that SI/O is driven low within the bit frame.

Reset and Discovery Response

A Reset and Discovery Response sequence is used by the master to reset the slave device as well as to perform a general bus call to determine if any devices are present on the bus.

To begin the Reset portion of the sequence, the master must drive SI/O low for a minimum time of t_{RESET} or t_{DSCHG}. This length of time differs for Standard Speed mode and for High-Speed mode and whether the device is currently busy with other operations. Figure 4, shown below, illustrates the Reset and Discovery Response.

FIGURE 4: RESET AND DISCOVERY RESPONSE WAVEFORM



Data Input

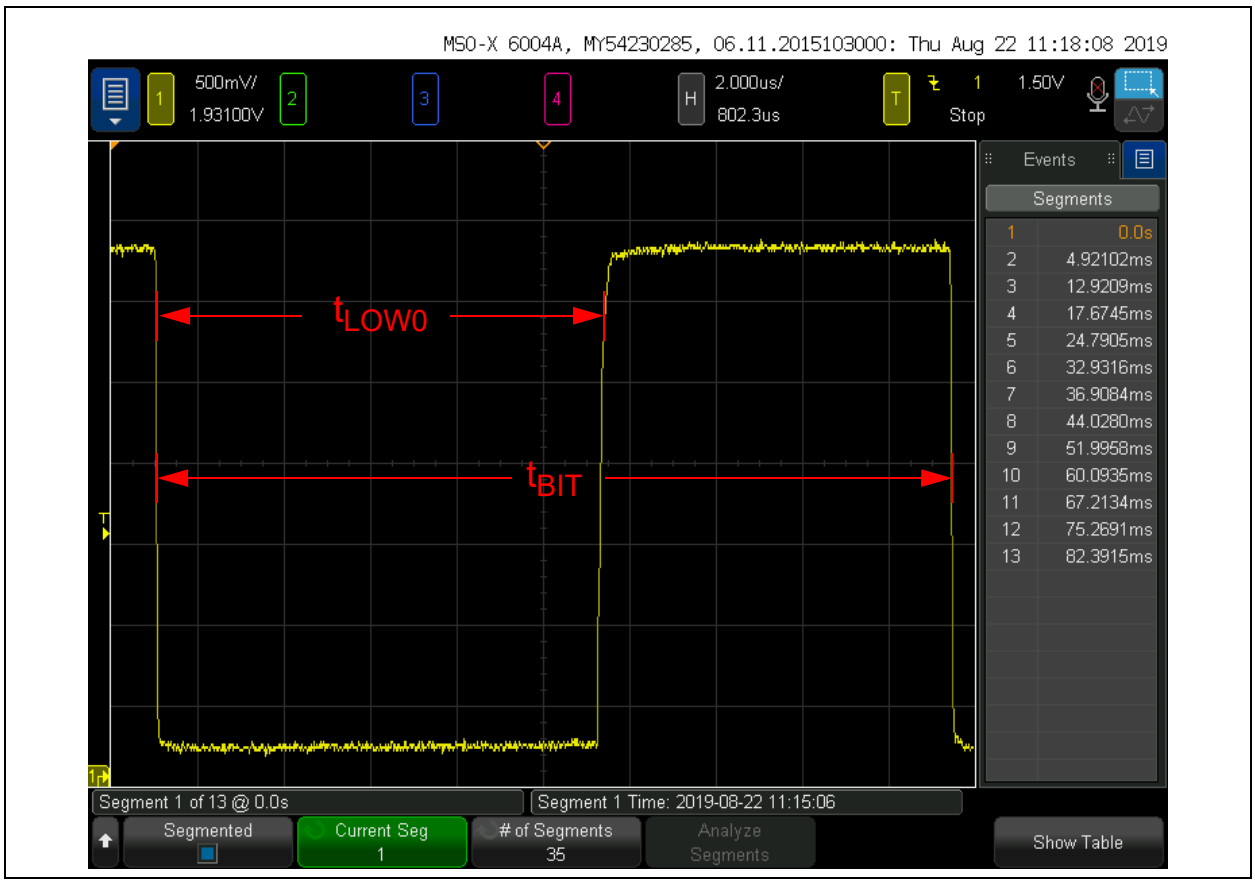
A data input bit frame can be used by the master to transmit either a logic '0' or logic '1' data bit to the slave device. The input bit frame is initiated when the master drives the SI/O line low. The length of time that the SI/O line is held low will dictate whether the master is transmitting a logic '0' or a logic '1' for that bit frame. For a logic '0' input, the length of time that the SI/O line must be held low is defined as t_{LOW0}. Similarly, for a logic '1' input, the length of time that the SI/O line must be held low is defined as t_{LOW1}.

The slave device will sample the state of the SI/O line after the maximum t_{LOW1} but prior to the minimum t_{LOW0} after SI/O was driven below the V_{IL} max threshold to determine if the data input is a logic '0' or a logic '1'. If the master is still driving the line low at the sample time, the slave device will decode that bit frame as a logic '0' as SI/O will be at a voltage less than V_{IL} max. If the master has already released the SI/O line, the slave device will see a voltage level greater than or equal to V_{IH} min due to the external pull-up resistor, and that bit frame will be decoded as a logic '1'.

LOGIC '0'

A logic '0' condition has multiple uses in the SWI protocol sequences. Just like I^2C, it is used to signify a '0' data bit, and it also is used for an Acknowledge (ACK) response. Figure 5 depicts the logic '0' input bit frame.

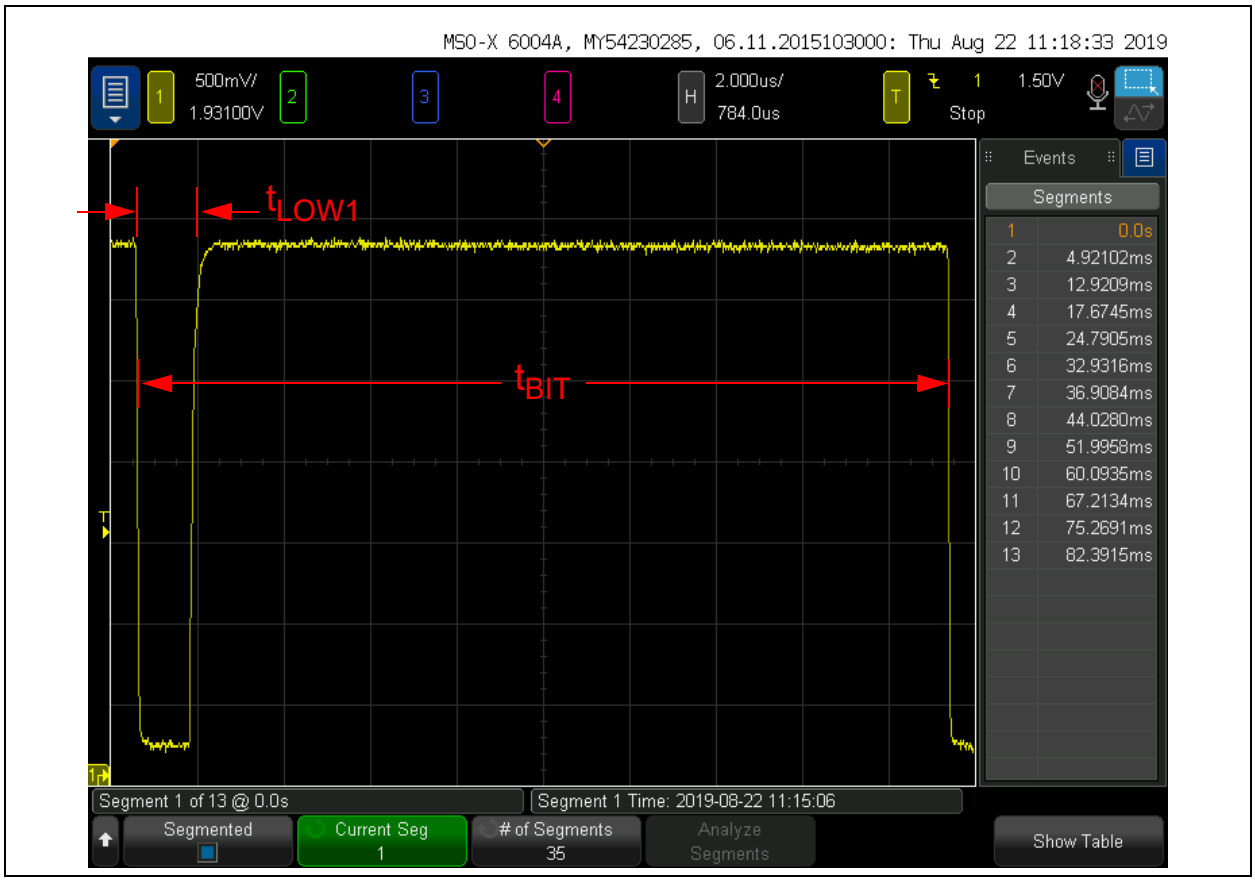
FIGURE 5: DATA INPUT LOGIC '0'



LOGIC '1'

A logic '1' condition has multiple uses in the SWI protocol sequences. Just like I^2C, it is used to signify a '1' data bit, and it also is used for a No Acknowledge (NACK) response. Figure 6 depicts the logic '1' input bit frame.

FIGURE 6: DATA INPUT LOGIC '1'



Data Output

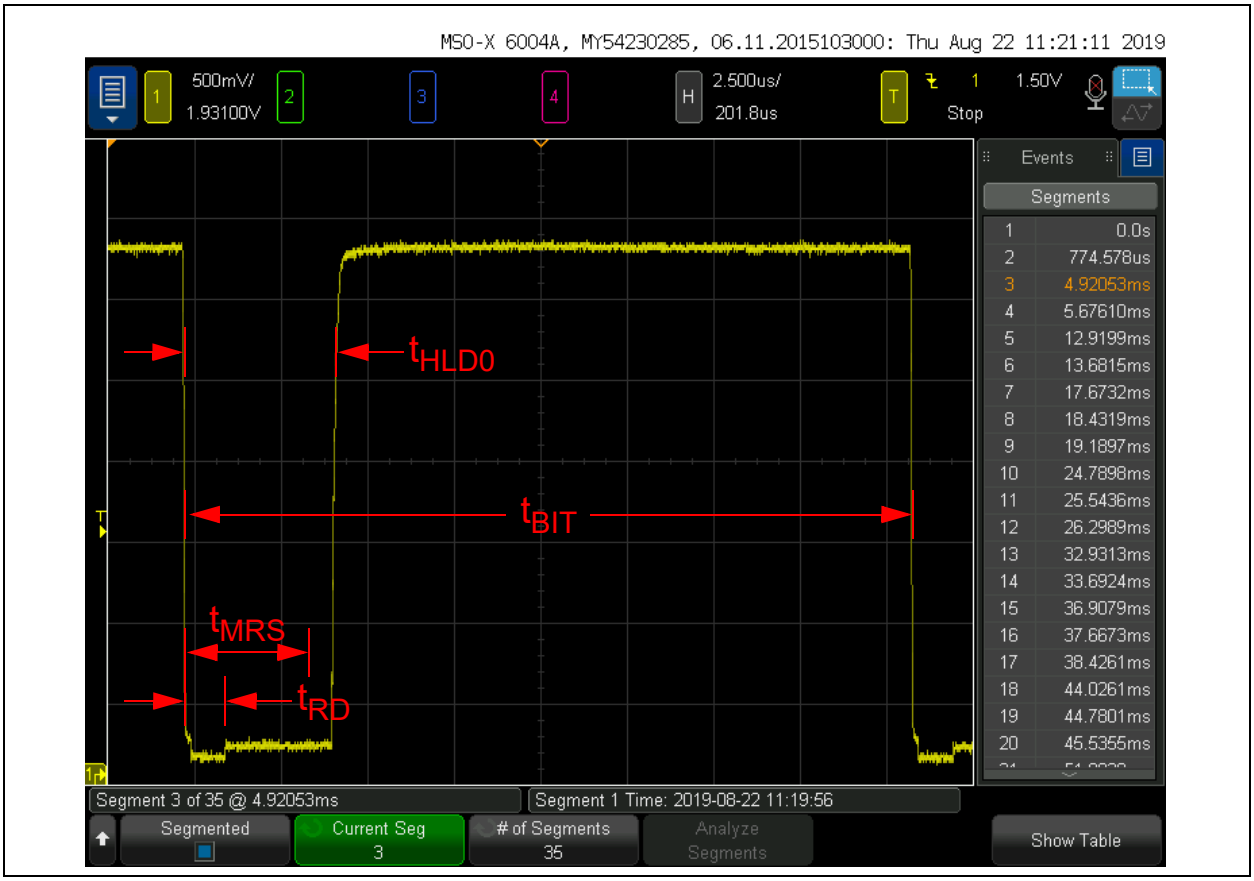
A data output bit frame is used when the master is to receive communication back from the slave device. Data output bit frames are used when reading any data out as well as any ACK or NACK responses from the slave device. Just as in the input bit frame, the master initiates the sequence by driving the SI/O line below the V_{IL} max threshold which engages the device internal timing generation circuit.

The critical timing parameter t_{RD}, which is found within the output bit frame, is defined as the amount of time the master must continue to drive the SI/O line low after crossing the below V_{IL} max threshold to request a data bit back from the device. Once the t_{RD} duration has expired, the master must release the SI/O line.

LOGIC '0'

If the slave device is responding with a logic '0' (for either a '0' data bit or an ACK response), it will begin to pull the SI/O line low concurrently during the t_{RD} window and will continue to hold it low for a duration of t_{HLD0}, after which it will release the line to be pulled back up to V_{PUP} (see Figure 7). Thus, when the master samples SI/O within the t_{MRS} window, it will see a voltage less than V_{IL} max and decode this event as a logic '0'. By definition, the t_{HLD0} time is longer than t_{MRS} time and therefore, the master is ensured to sample while the slave device is still driving the SI/O line low.

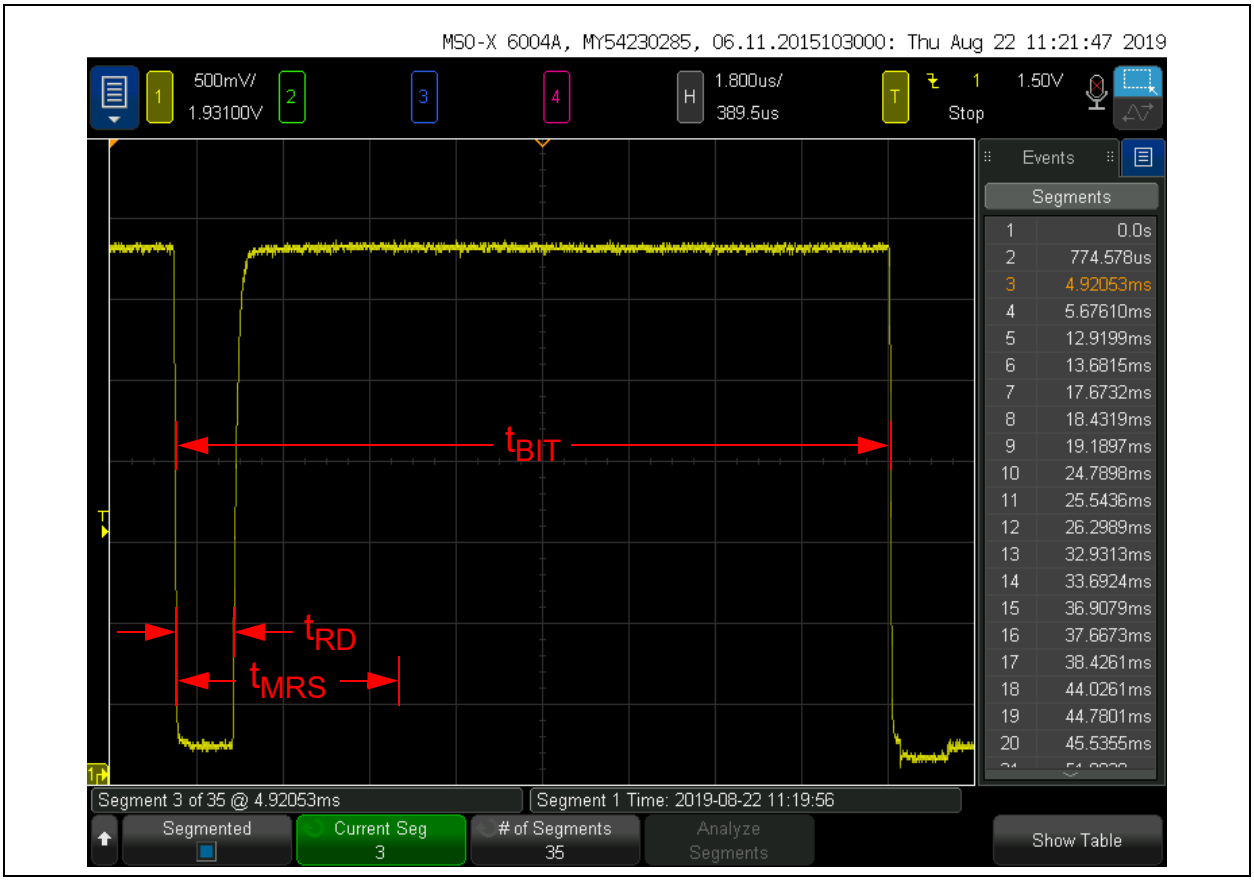
FIGURE 7: DATA OUTPUT LOGIC '0'



LOGIC '1'

If the slave device intends to respond with a logic '1' (for either a '1' data bit or a NACK response), it will not drive the SI/O line low. Once the master releases the SI/O line after the maximum t_{RD} has elapsed, the line will be pulled up to V_{PUP}. Thus, when the master samples the SI/O line within the t_{MRS} window, it will detect a voltage greater than V_{IH} min and decode this event as a logic '1'. Figure 8 depicts the logic '1' output bit frame.

FIGURE 8: DATA OUTPUT LOGIC '1'

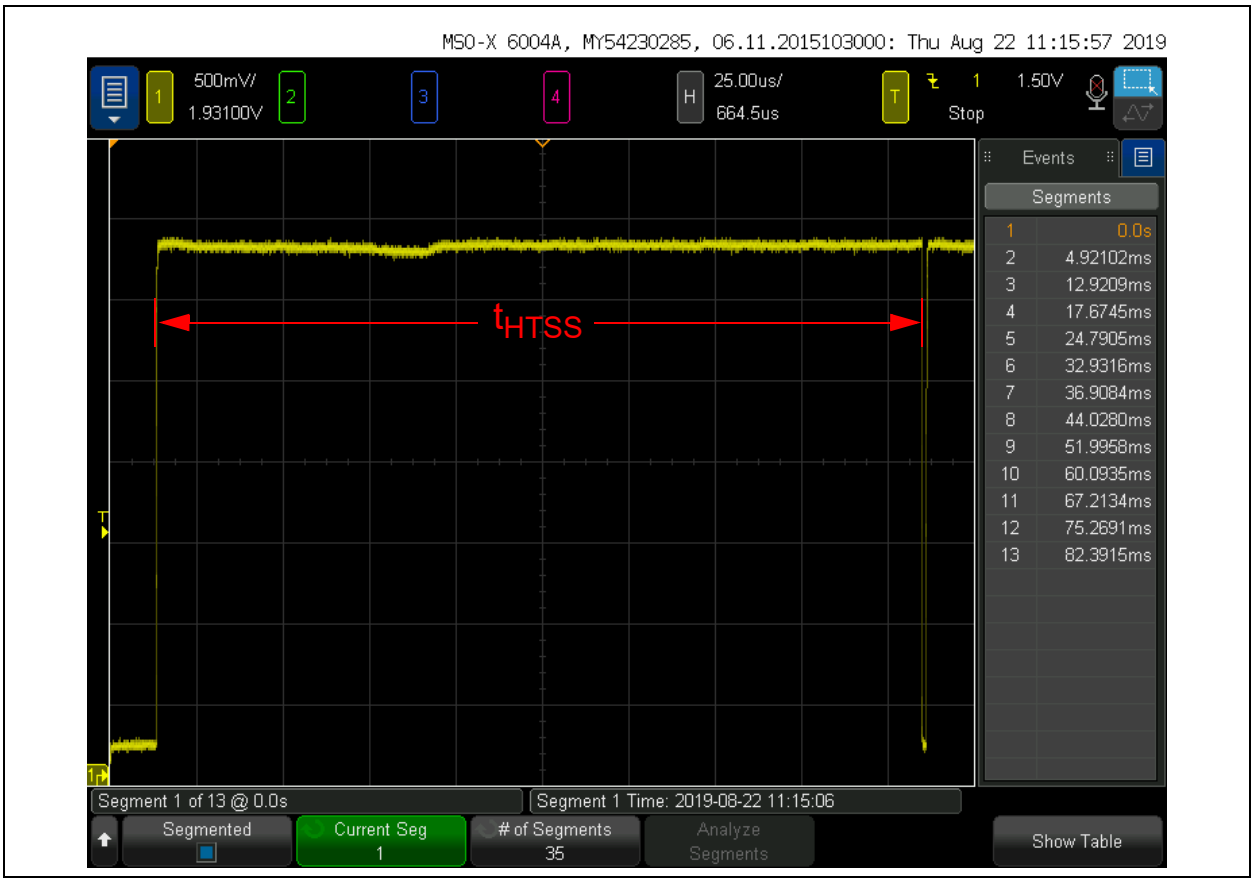


Start and Stop Conditions

All transactions to the slave device begin with a Start condition; therefore, a Start condition can only be transmitted by the master to the slave. Likewise, all transactions are terminated with a Stop condition and thus a Stop condition can only be transmitted by the master to the slave.

The Start and Stop conditions require identical biasing of the SI/O line. The Start/Stop condition is created by holding the SI/O line at a voltage of V_{PUP} for a duration of t_{HTSS}. Figure 9 depicts the Start and Stop conditions.

FIGURE 9: START AND STOP CONDITIONS



AT21CS SERIES OPERATIONS

Device Addressing

Accessing the device requires a Start condition followed by an 8-bit device address byte.

The device protocol sequence emulates what would be required for an I^2C Serial EEPROM, with the exception that the beginning four bits of the device address are used as an opcode for the different commands and actions that the device can perform.

Since multiple slave devices can reside on the bus, each slave device must have its own unique address so that the master can access each device independently. After the 4-bit opcode, the following three bits of the device address byte are comprised of the slave address bits. The three slave address bits are preprogrammed prior to shipment and are set to read-only. Obtaining devices with different slave address bit values is done by purchasing a specific ordering code.

Following the three slave address bits is a Read/$\overline{\text{Write}}$ select bit where a logic '1' indicates a read and a logic '0' indicates a write. Upon the successful comparison of the device address byte, the device will respond with an ACK (logic '0'). If the 4-bit opcode is invalid or the three bits of slave address do not match what is preprogrammed in the device, the device will not respond or NACK on the SI/O line and will return to a Standby state. Refer to Figure 10 for an example waveform of the device address byte.

TABLE 1: DEVICE ADDRESS BYTE

4-bit Opcode				Preprogrammed Slave Address Bits			Read/$\overline{\text{Write}}$
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Refer to Table 2				A2	A1	A0	R/$\overline{\text{W}}$

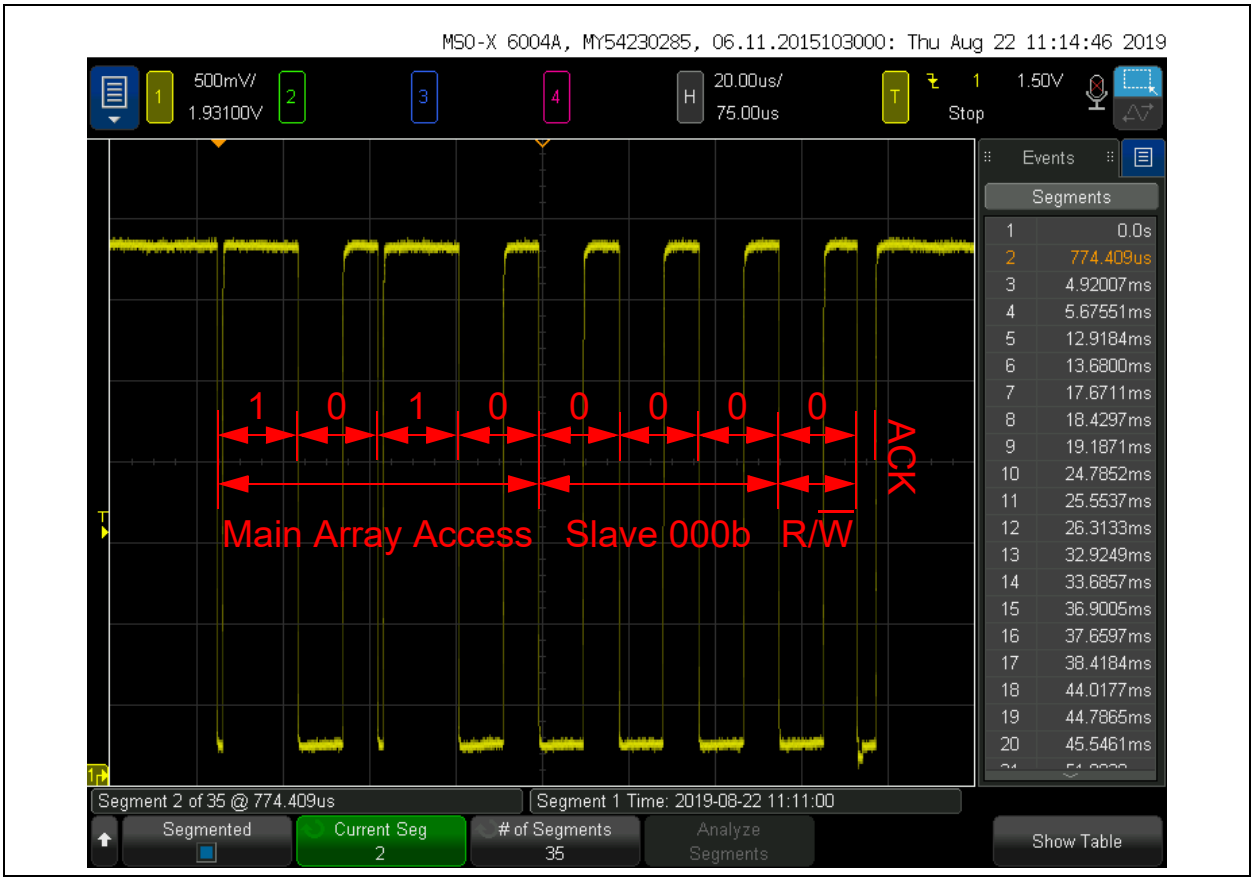
AVAILABLE OPCODES

Table 2 outlines available opcodes for the device.

TABLE 2: OPCODES USED BY THE DEVICE

Command	4-bit Opcode	Brief Description of Functionality
Main Memory Array Access	1010 (Ah)	Read and write the contents of the main memory array.
Security Register Access	1011 (Bh)	Read and write the contents of the Security register.
Lock Security Register	0010 (2h)	Permanently lock the contents of the Security register.
ROM Zone Register Access	0111 (7h)	Inhibit further modification to a zone of the main memory array.
Freeze ROM Zone State	0001 (1h)	Permanently lock the current state of the ROM Zone registers.
Manufacturer ID Read	1100 (Ch)	Query manufacturer and density of device.
Standard Speed Mode	1101 (Dh)	Switch to Standard Speed mode operation (AT21CS01 only command, the AT21CS11 will NACK this command).
High-Speed Mode	1110 (Eh)	Switch to High-Speed mode operation (device power-on default. The AT21CS11 will ACK this command).

FIGURE 10: DEVICE ADDRESS BYTE



Immediately following the ACK response to the device address byte, a memory address byte must be transmitted to the device. The memory address byte contains a 7-bit memory array address to specify which location within the device to start reading or writing. Refer to Table 3 to review these bit positions.

TABLE 3: MEMORY ADDRESS BYTE

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Don't Care	A6	A5	A4	A3	A2	A1	A0

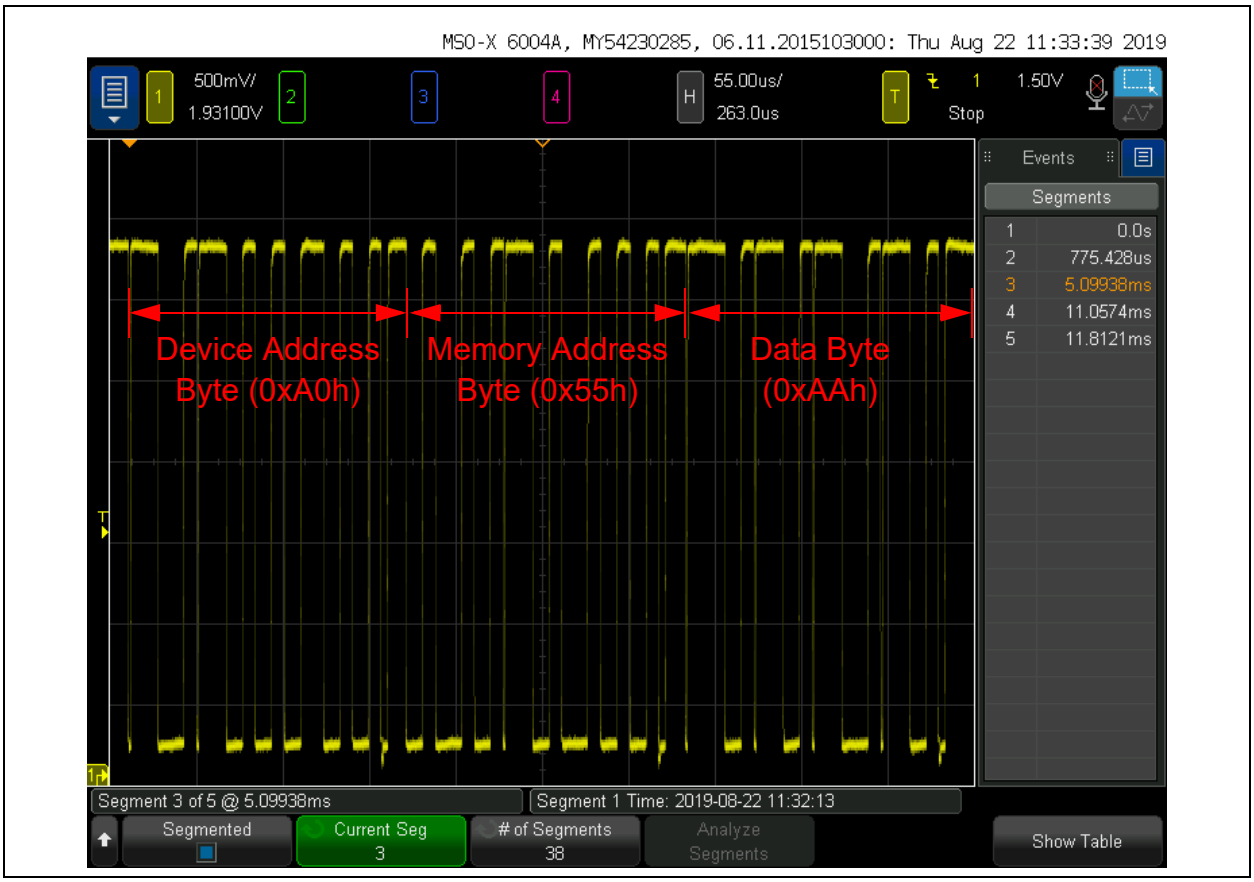
Write Operations

All write operations to the device begin with the master sending a Start condition, followed by a device address byte (opcode Ah for the Main Array and opcode Bh for the Security register) with the R/$\overline{W}$ bit set to '0' followed by the memory address byte. Next, the data value(s) to be written to the device are sent. Data values must be sent in 8-bit (byte) increments to the device and the write operation is followed by a Stop condition. If a Stop condition is sent somewhere other than at a byte boundary, the current write operation will be aborted. The device allows single byte writes, partial page writes, and full page writes.

To ensure that the address and data sent to the device are not corrupted while any type of internal write operation is in progress, commands sent to the device are blocked from being recognized until the internal operation is completed. If a write interruption occurs (SI/O pulsed low) and is small enough to not deplete the internal power storage, the device will NACK, signaling that the operation is in progress. If an interruption is longer than t_{DSCHG} then internal write operation will be terminated and may result in data corruption. Figure 11 depicts a byte write operation to the main memory array.

Note: Any attempt to interrupt the internal write cycle by driving the SI/O line low may cause the data being programmed to become corrupted. Other memory locations within the memory array will not be affected. If the master must interrupt a write operation, the SI/O line must be driven low for a minimum time of t_{DSCHG}.

FIGURE 11: BYTE WRITE TO MAIN MEMORY ARRAY



Read Operations

Read operations are initiated in a similar way as write operations with the exception that the Read/Write select bit in the device address byte must be set to a logic '1'. There are multiple read operations supported by the device:

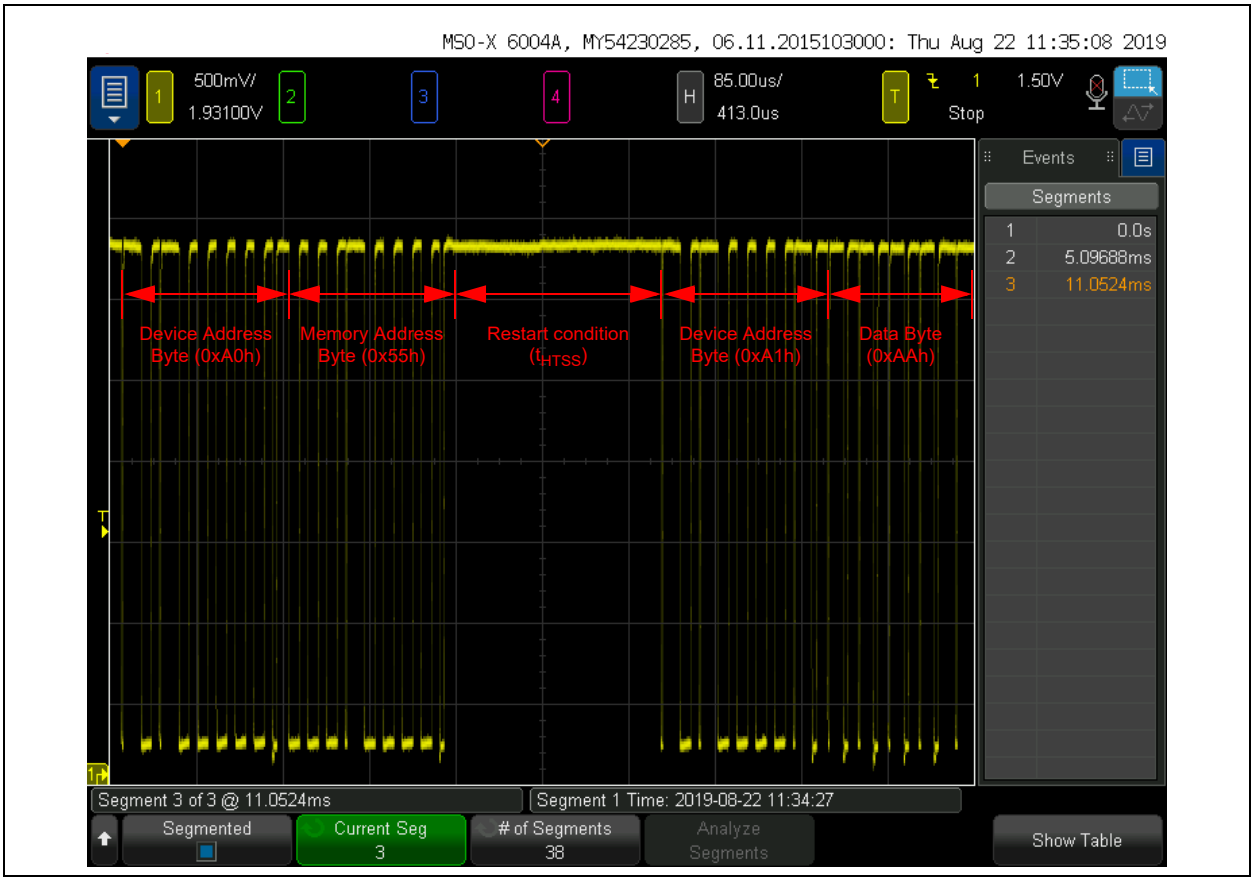
- Current Address Read
- Random Read
- Sequential Read
- Read from the Security Register
- Manufacturer ID Read

The device contains a single, shared-memory Address Pointer that maintains the address of the next byte in the main memory array or Security register to be accessed. For example, if the last byte read or written was memory location 0Dh of the main memory array, then the Address Pointer will be pointing to memory location 0Eh. As such, when changing from a read in one region to another region, the first read operation in the new region should begin with a random read instead of a current address read to ensure the Address Pointer is set to a known value within the desired region.

If the end of the main memory array or the Security register is reached, then the Address Pointer will "rollover" to the beginning (address 00h) of that region. The Address Pointer retains its value between operations as long as the pull-up voltage on the SI/O pin is maintained or the device has not been reset. Figure 12 depicts a random read operation within the main memory array.

Note:	If the last operation to the device accessed the Security register, then a random read should be performed to ensure that the Address Pointer is set to a known memory location within the device.

FIGURE 12: RANDOM READ WITHIN THE EEPROM



CONCLUSION

This application note offers designers a set of firmware routines to access the AT21CS Series Serial EEPROMs using a generic I/O pin on the PIC16F1719 microcontroller. All routines were written in C using MPLAB X IDE V5.20 and the code was generated using MPLAB Code Configurator. The hardware used in this application note is the Explorer 8 Development Kit (DM160228). Details related to single-wire protocol and device operation can be found in the appropriate device data sheet found at www.microchip.com.

APPENDIX A: REVISION HISTORY

Revision A (12/2019)

Initial release of this document.

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

ISBN: 978-1-5224-5458-8

For information regarding Microchip's Quality Management Systems, please visit www.microchip.com/quality.

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

Austin, TX
Tel: 512-257-3370

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
Novi, MI
Tel: 248-848-4000

Houston, TX
Tel: 281-894-5983

Indianapolis
Noblesville, IN
Tel: 317-773-8323
Fax: 317-773-5453
Tel: 317-536-2380

Los Angeles
Mission Viejo, CA
Tel: 949-462-9523
Fax: 949-462-9608
Tel: 951-273-7800

Raleigh, NC
Tel: 919-844-7510

New York, NY
Tel: 631-435-6000

San Jose, CA
Tel: 408-735-9110
Tel: 408-436-4270

Canada - Toronto
Tel: 905-695-1980
Fax: 905-695-2078

ASIA/PACIFIC

Australia - Sydney
Tel: 61-2-9868-6733

China - Beijing
Tel: 86-10-8569-7000

China - Chengdu
Tel: 86-28-8665-5511

China - Chongqing
Tel: 86-23-8980-9588

China - Dongguan
Tel: 86-769-8702-9880

China - Guangzhou
Tel: 86-20-8755-8029

China - Hangzhou
Tel: 86-571-8792-8115

China - Hong Kong SAR
Tel: 852-2943-5100

China - Nanjing
Tel: 86-25-8473-2460

China - Qingdao
Tel: 86-532-8502-7355

China - Shanghai
Tel: 86-21-3326-8000

China - Shenyang
Tel: 86-24-2334-2829

China - Shenzhen
Tel: 86-755-8864-2200

China - Suzhou
Tel: 86-186-6233-1526

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

China - Xian
Tel: 86-29-8833-7252

China - Xiamen
Tel: 86-592-2388138

China - Zhuhai
Tel: 86-756-3210040

ASIA/PACIFIC

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

India - New Delhi
Tel: 91-11-4160-8631

India - Pune
Tel: 91-20-4121-0141

Japan - Osaka
Tel: 81-6-6152-7160

Japan - Tokyo
Tel: 81-3-6880- 3770

Korea - Daegu
Tel: 82-53-744-4301

Korea - Seoul
Tel: 82-2-554-7200

Malaysia - Kuala Lumpur
Tel: 60-3-7651-7906

Malaysia - Penang
Tel: 60-4-227-8870

Philippines - Manila
Tel: 63-2-634-9065

Singapore
Tel: 65-6334-8870

Taiwan - Hsin Chu
Tel: 886-3-577-8366

Taiwan - Kaohsiung
Tel: 886-7-213-7830

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

Thailand - Bangkok
Tel: 66-2-694-1351

Vietnam - Ho Chi Minh
Tel: 84-28-5448-2100

EUROPE

Austria - Wels
Tel: 43-7242-2244-39
Fax: 43-7242-2244-393

Denmark - Copenhagen
Tel: 45-4450-2828
Fax: 45-4485-2829

Finland - Espoo
Tel: 358-9-4520-820

France - Paris
Tel: 33-1-69-53-63-20
Fax: 33-1-69-30-90-79

Germany - Garching
Tel: 49-8931-9700

Germany - Haan
Tel: 49-2129-3766400

Germany - Heilbronn
Tel: 49-7131-72400

Germany - Karlsruhe
Tel: 49-721-625370

Germany - Munich
Tel: 49-89-627-144-0
Fax: 49-89-627-144-44

Germany - Rosenheim
Tel: 49-8031-354-560

Israel - Ra'anana
Tel: 972-9-744-7705

Italy - Milan
Tel: 39-0331-742611
Fax: 39-0331-466781

Italy - Padova
Tel: 39-049-7625286

Netherlands - Drunen
Tel: 31-416-690399
Fax: 31-416-690340

Norway - Trondheim
Tel: 47-7288-4388

Poland - Warsaw
Tel: 48-22-3325737

Romania - Bucharest
Tel: 40-21-407-87-50

Spain - Madrid
Tel: 34-91-708-08-90
Fax: 34-91-708-08-91

Sweden - Gothenberg
Tel: 46-31-704-60-40

Sweden - Stockholm
Tel: 46-8-5090-4654

UK - Wokingham
Tel: 44-118-921-5800
Fax: 44-118-921-5820