AN4051

Implementing 3.4 MHz I^2C Bus Communication with the ATSAMD21 and 24CS Serial EEPROM Family

Author: Erik Fasnacht
Microchip Technology Inc.

INTRODUCTION

The Microchip Technology Inc. 24CS Family of Serial EEPROMs is the first Serial EEPROM to introduce support for 3.4MHz High-Speed Mode I^2C Bus communication. Previous generation Serial EEPROMs have been limited to 1 MHz I^2C Bus communication. This new feature narrows the speed performance gap between I^2C bus communication and the much faster SPI bus communication while retaining the ease of use benefits of I^2C.

This application note demonstrates how to take advantage of the higher speed Bus communication of the 24CS Family of Serial EEPROMs when interfacing with the ATSAMD21 Family of 32-bit Cortex-M0+ Microcontrollers. The ATSAMD21 offers up to six Serial Communication Interfaces (SERCOM), any of which can be configurable to operate in I^2C with support for up to 3.4 MHz read and write operations, making it an ideal companion for this family of Serial EEPROMs. For additional information related to the 24CS Series of Serial EEPROM or the ATSAMD21 MCU, visit www.microchip.com.

High-Speed Mode

The 24CS Series supports the I^2C High-Speed (HS) mode, allowing it to operate at clock frequencies up to 3.4 MHz for read and write operations.

In order to place the device into HS mode, the host must first initiate a Start condition, followed by the reserved HS mode host code of ‘00001xxxb’ (Table 1). The HS mode host code must be sent to the device at Fast mode plus (1 MHz) or slower clock frequencies. Since the HS mode host code is meant to be recognized by all client devices that support the HS mode, the 24CS Series will not Acknowledge (NACK) the HS mode host code.

TABLE 1: HIGH-SPEED MODE HOST CODE

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	ACK Bit
0	0	0	0	1	x	x	x	NACK from Clients

Note: x indicates a don’t care bit, and the firmware sets these bits to ‘0’.

Once the device receives the HS mode host code and the NACK occurs, the device will relax its input filters on Serial Data (SDA) and Serial Clock (SCL) to the HS mode tolerance to accept transfers up to 3.4 MHz. The device will then enter HS mode and wait for a Repeated Start condition before the next operation can occur.

Next, the host must issue a Repeated Start condition, followed by a valid device address byte to which the device will ACK. The host can continue with read or write operations at higher clock speeds and the device will continue to operate in the HS mode until one of the following events occur:

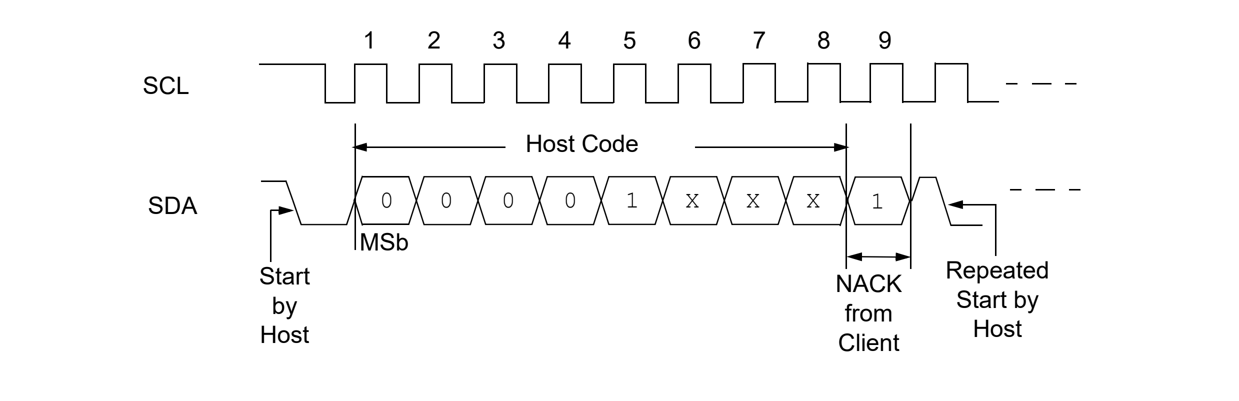
- The host sends a Stop condition. Therefore, the host should use a Repeated Start condition to begin new HS mode operations rather than a Stop-Start sequence.
- A Power-on-Reset (POR) event occurs.

> **Note:** The internal write cycle requires a Stop condition to be sent after the last data byte. This Stop condition will cause the device to exit HS mode. Therefore, if more than one page of data is to be written, HS mode must be re-entered for every write operation.

Once the device exits HS mode from one of these events, the device will switch its input and output filters back to the standard I^2C (Legacy) mode. Figure 1 illustrates the HS mode entry sequence.

> **Note:** High-Speed mode entry is ignored during a write cycle. Therefore, polling must occur while using Fast Mode Plus (1 MHz) or slower clock frequencies.

FIGURE 1: HIGH-SPEED MODE ENTRY SEQUENCE



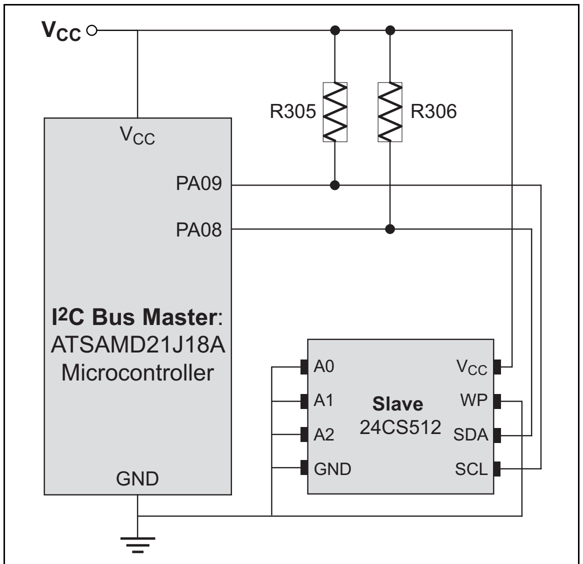
HARDWARE

The hardware used in conjunction with the firmware is the The SAM D21 Xplained Pro Evaluation Kit. For additional information about the hardware, refer to the User Guide found at www.microchip.com.

Figure 2 is the hardware schematic that depicts the interface between the Microchip 24CS Series of devices and the microcontroller, ATSAMD21J18A. The schematic shows the necessary connections between the microcontroller and the serial EEPROM. The firmware was written assuming these connections, including serial connection (PA08 is SDA and PA09 is SCL) between the microcontroller and the serial EEPROM while the evaluation kit includes pull-up resistors (R305 and R306) that are required for I^{2}C communication.

> **Note:** R305 and R306 as included by default with the evaluation kit are insufficient for High-Speed Mode operation. Therefore, these resistors should be replaced using appropriate resistance values.

FIGURE 2: HARDWARE CIRCUIT



To determine sufficient pull-up resistors, the NXP I^{2}C specification gives two formulas, one for maximum and one for minimum resistance.

The maximum formula will be used to determine R305 and R306 replacements (see Equation 1).

EQUATION 1: MAXIMUM RESISTANCE PULL-UP RESISTOR

$$R_{P(max)} = \frac{T_R}{0.8473 \times C_B}$$

Where:

- T_R = max rise time
- C_B = total bus capacitance

For this application note, we will use the NXP specification for max rise time on SCL (T_R = 40 ns) and assume that C_B is equal to 100 pF (see Example 1).

EXAMPLE 1:

$$R_{P(max)} = \frac{40ns}{0.8473 \times 100pf} = 472\Omega$$

> **Note:** R305 and R306 were replaced with 330Ω resistors for this application.

FIGURE 3: XPLAINED PRO HARDWARE

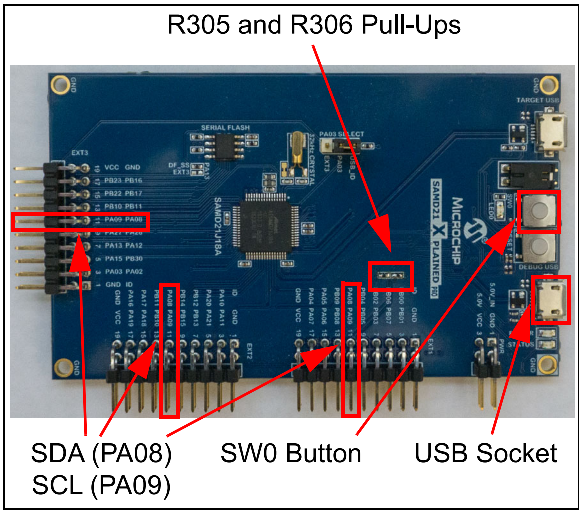
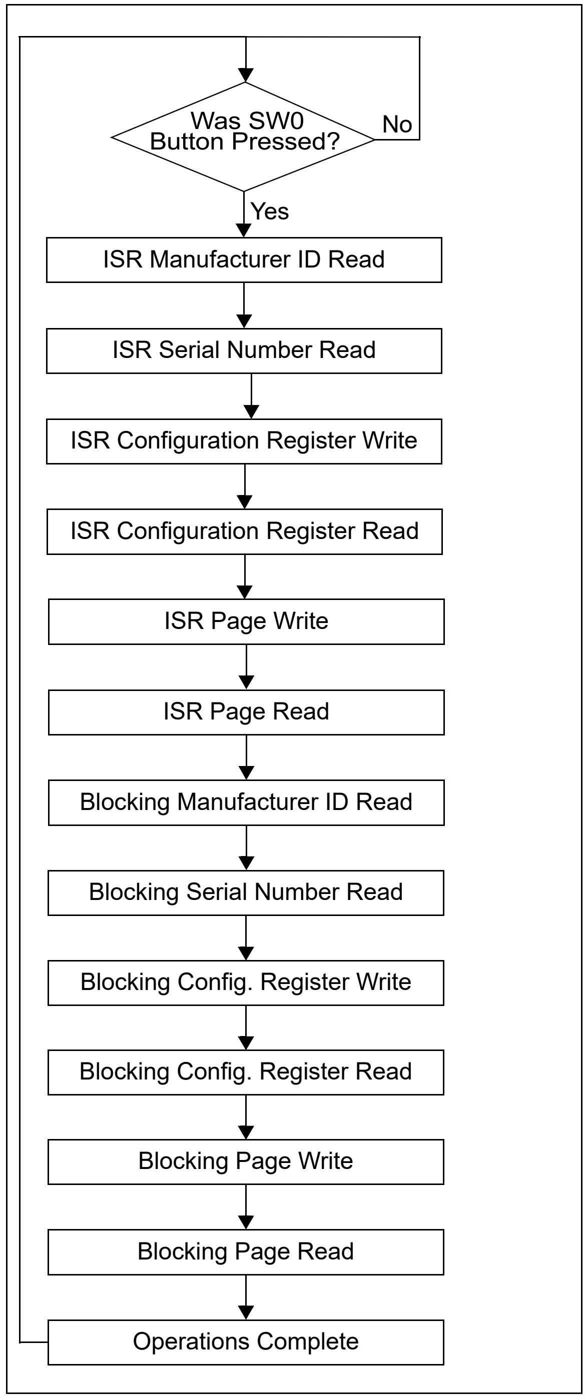


FIGURE 4: FLOW CHART



FIRMWARE

The purpose of the firmware is to generate specific I^{2}C High-Speed mode transactions using the ATSAMD21J18A microcontroller. The focus is to provide the user with a strong understanding of High-Speed mode communication with the 24CS Series Serial EEPROM devices, allowing for more complex programs to be written in the future.

The firmware was written in C using Atmel Studio 7 Integrated Development Environment V7.0.1931 and a portion of the code was generated using Atmel Advanced Software Framework (ASF).

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 of I^{2}C High-Speed mode transactions.

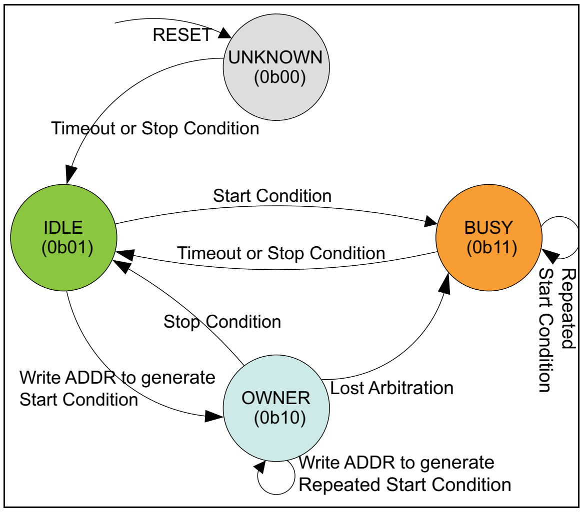
Overview

The firmware uses the SAMD21 Xplained Pro SW0 button, USB socket, and either EXT1, EXT2 or EXT 3 headers (for access to PA08, SDA and PA09 SCL). The SW0 button is used as a hardware breakpoint to start sending I^{2}C bus transactions. Figure 3 highlights the locations of interest on the SAMD21 Xplained Pro Evaluation Kit including the Micro-USB socket, the SW0 button, R305 and R306 pull-ups and SDA (PA08) and SCL (PA09) signals.

After the SW0 button is pressed, various I^{2}C transactions will be sent to the client device. The firmware uses interrupt service routines (ISR) along with blocking routines for the various transactions. Figure 4 is a flowchart representing the High-Speed mode I^{2}C bus transactions.

The I^2C interface on the ATSAMD21J18A uses SERCOM interface to configure the MCU for high-speed mode operation along with standard I^2C transactions. Figure 5 depicts the I^2C bus state.

FIGURE 5: BUS STATE DIAGRAM



The firmware sends High-Speed mode entry (08h) before every transaction (see Figure 6).

Note:	High-Speed mode host code defines 'don't care' bits and the firmware sets these bits to '0'.

The MCU dictates the clock speed, which is controlled using the SERCOM2_Handler flag SERCOM_I2CM_ADDR_HS. This flag is set following the High-Speed mode entry transaction. Figure 7 illustrates the clock or SCL frequency measured when not in High-Speed mode. Figure 8 illustrates the SCL frequency measured when in High -peed mode.

FIGURE 6: HIGH-SPEED MODE ENTRY

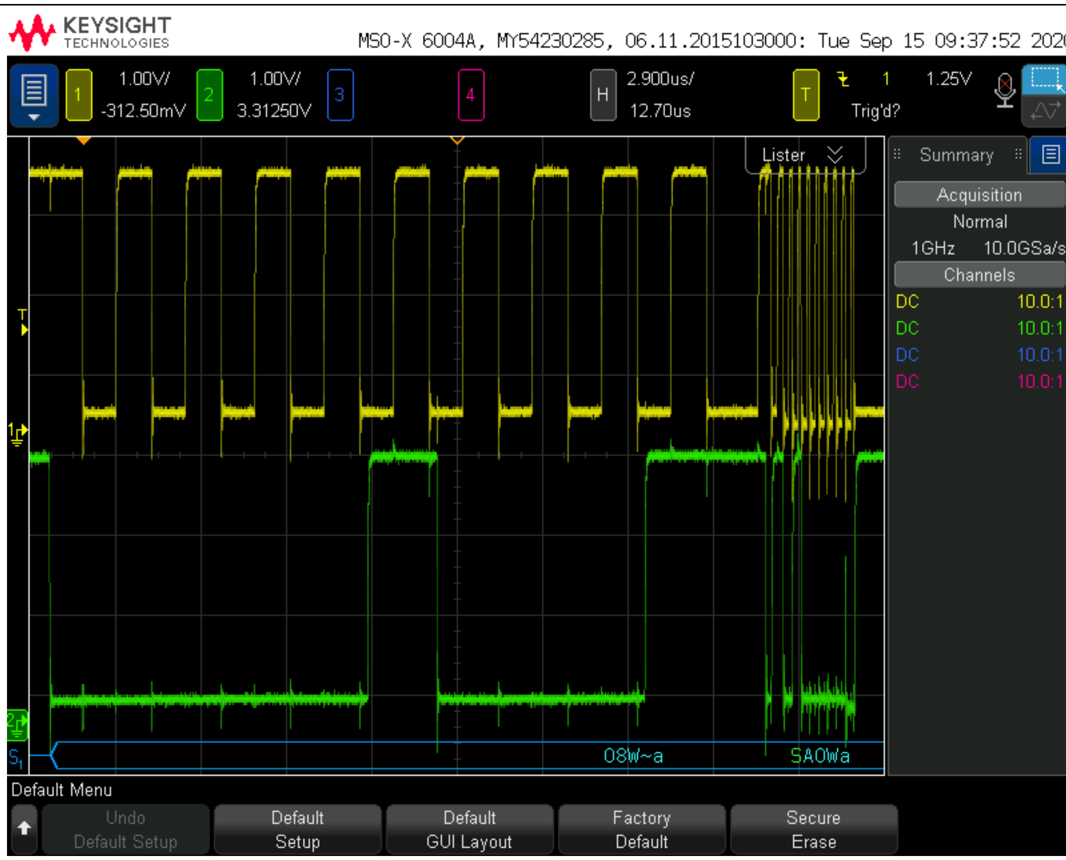


FIGURE 7: STANDARD SPEED SCL FREQUENCY

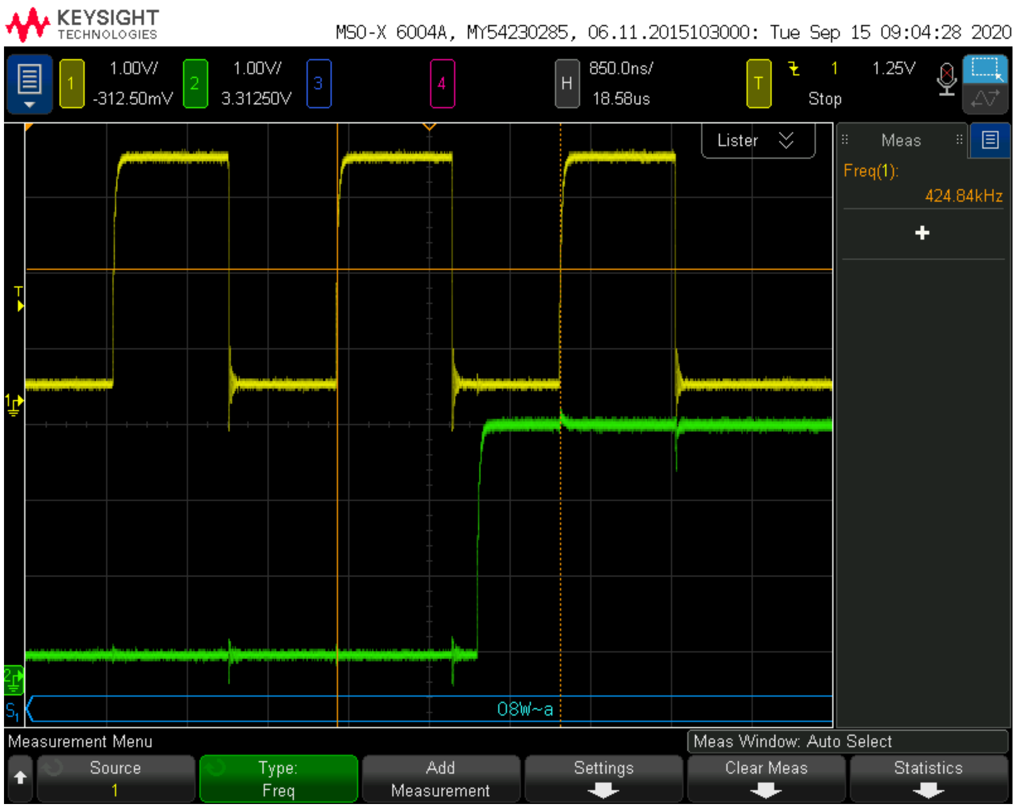
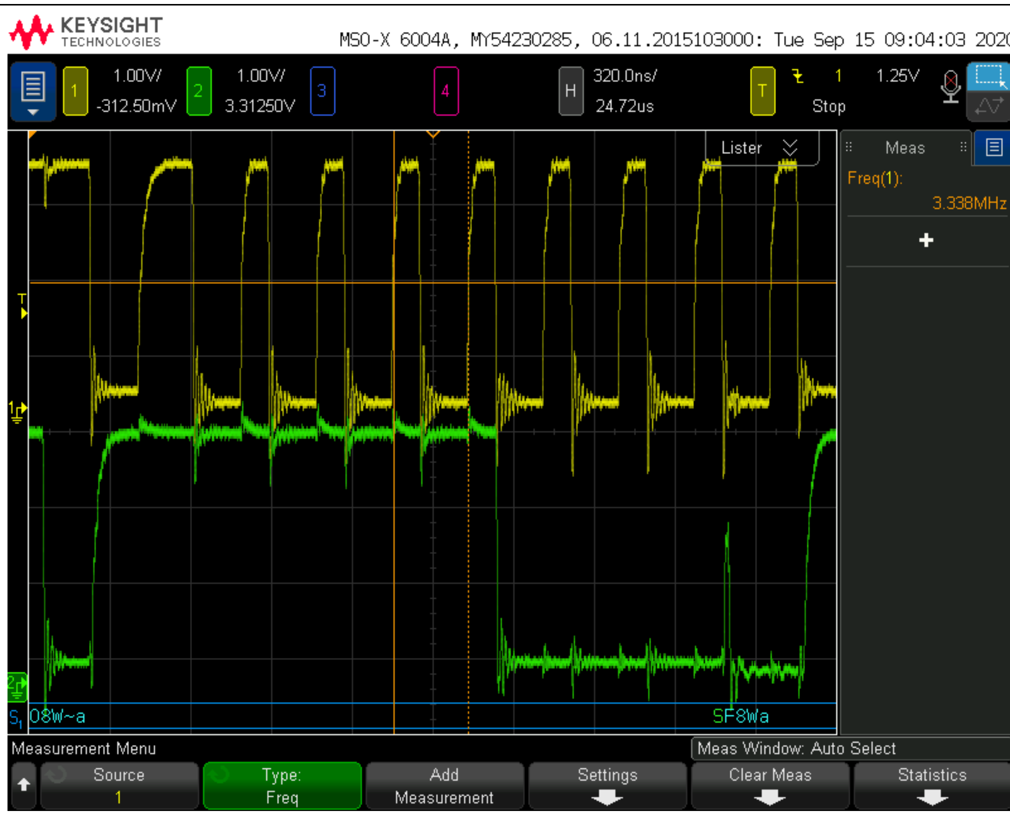


FIGURE 8: HIGH-SPEED SCL FREQUENCY

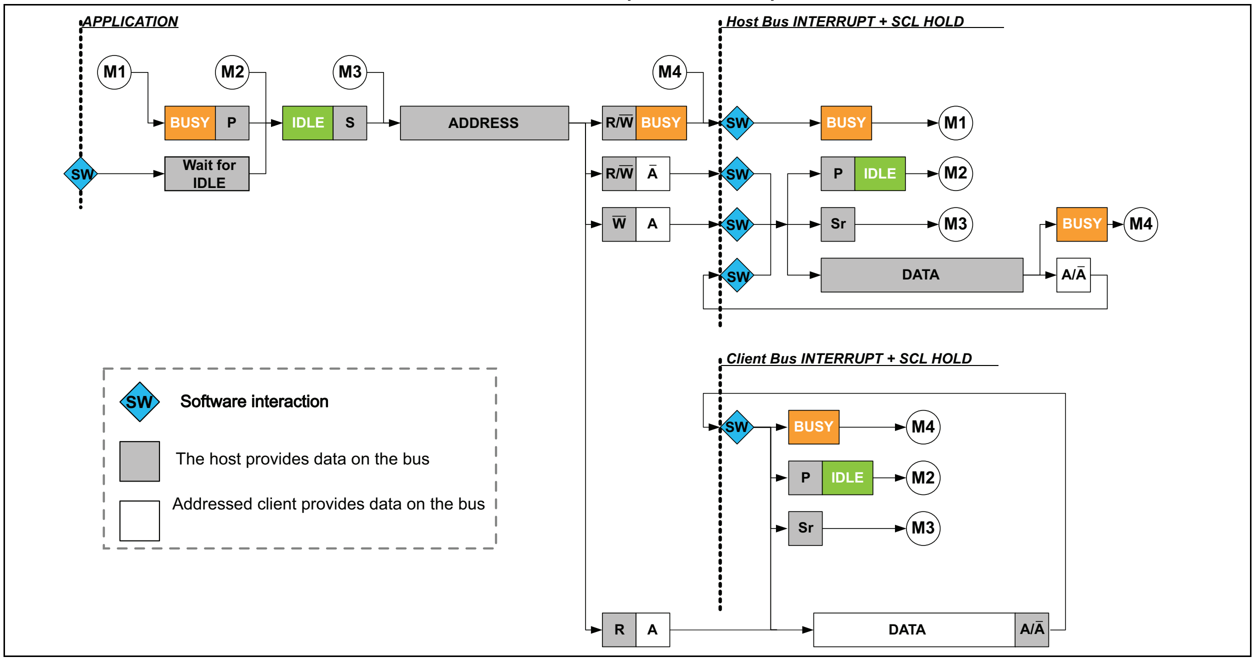


The firmware includes both interrupt service routines (ISR) and blocking functions that can be leveraged by the user, depending on the application and the MCU performance level required. ISRs are used to create asynchronous transactions, ensuring MCU performance. In contrast, blocking functions block the MCU from performing other tasks. Both ISR and blocking functions have their strengths and weaknesses, and each I²C operation has a ISR and blocking function that the user can leverage.

ISR FUNCTIONS

The first six I²C operations (as shown in Figure 4) that are sent to the device are the ISR functions. The firmware modifies the ISR variables before enabling the interrupt and the ISR is invoked by the ACK bit in the I²C protocol as shown in Figure 9.

FIGURE 9: I²C HOST BEHAVIOR DIAGRAM (SCLSM = 1)



The ISR functions are denoted by _isr at the end of the function name. These ISR functions use the built-in SERCOM2_Handler function and use a switch case loop to determine which I²C operation is to be performed.

In order to make sure the ISR isn't currently running and changing the ISR variables, polling routine isr_isBusy () was created and must be checked before every ISR operation is performed. After every ISR read operation, it is required to wait until the ISR is complete to ensure the read buffer is defined.

BLOCKING FUNCTIONS

The last 6 I²C operations (as shown in Figure 4) in the firmware utilize blocking functions. These functions can be leveraged by the user if MCU performance isn't as critical. The blocking functions wait for the Host on Bus (MB) and Client on Bus (SB) flags to be cleared before proceeding with the next operation.

I²C HIGH-SPEED OPERATIONS

The firmware includes many I²C transactions. This section includes a waveform of each transaction. Note that the ISR and blocking function for each transaction are identical as seen by the device so only one waveform is needed.

Note:	High-Speed mode entry is included in each transaction.

These transactions are as follows:

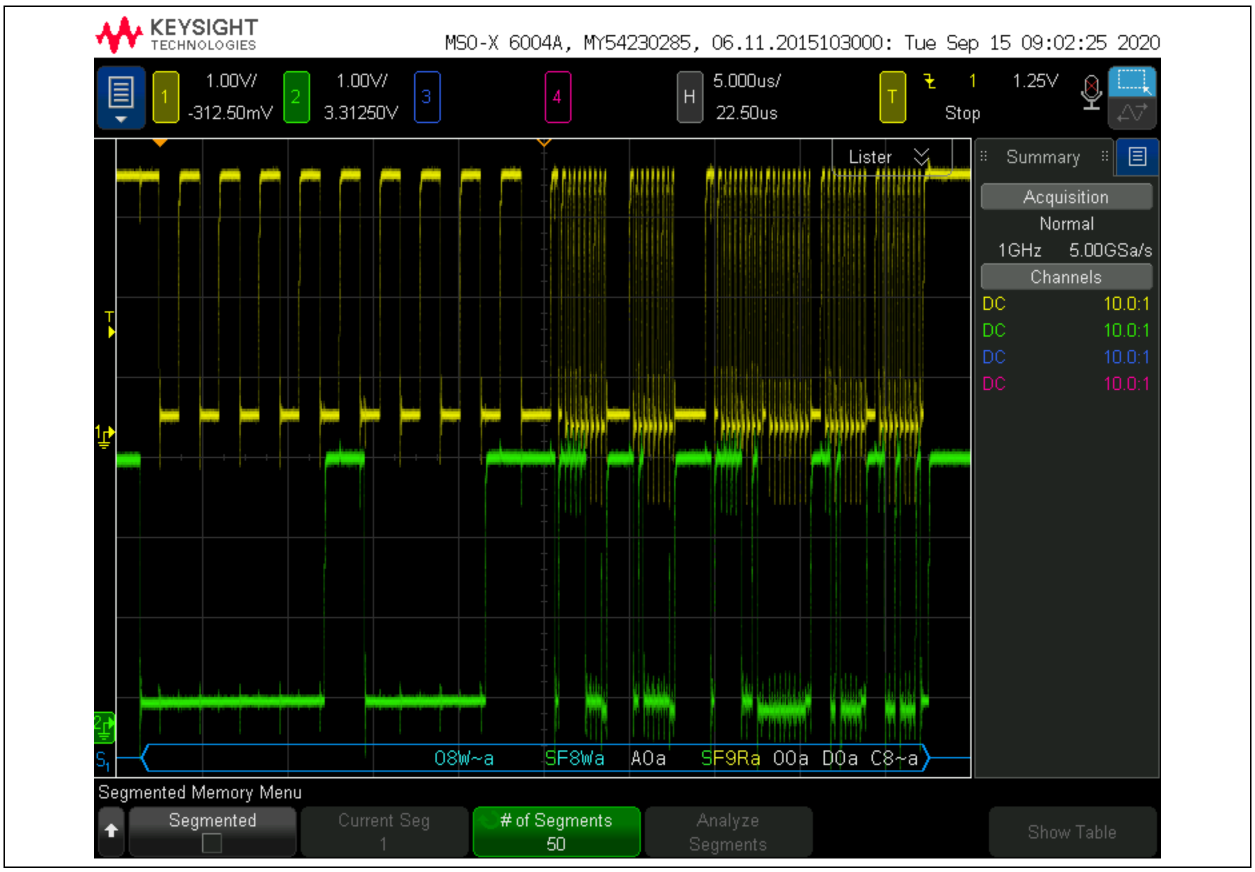
- Manufacturer ID Read
- Serial Number Read
- Configuration Register Write
- Configuration Register Read
- Page Write
- Page Read

Manufacturer ID Read

The first byte of Manufacturer ID data contains the eight Most Significant bits (D23-D16) of the 24-bit data value. The MCU can then return an ACK to indicate it successfully received the data, upon which the device will send the second byte (D15-D8) of Manufacturer ID data. The process repeats until all three bytes have been read out and the MCU sends a NACK (logic '1') to complete the sequence. Figure 10 depicts a High-Speed Manufacturer ID read operation.

Note:	When using the i2c_mfgidRead_isr function, it is necessary to wait after the read operation using the isr_isBusy () polling function to ensure the data buffer is valid.

FIGURE 10: MANUFACTURER ID READ OPERATION

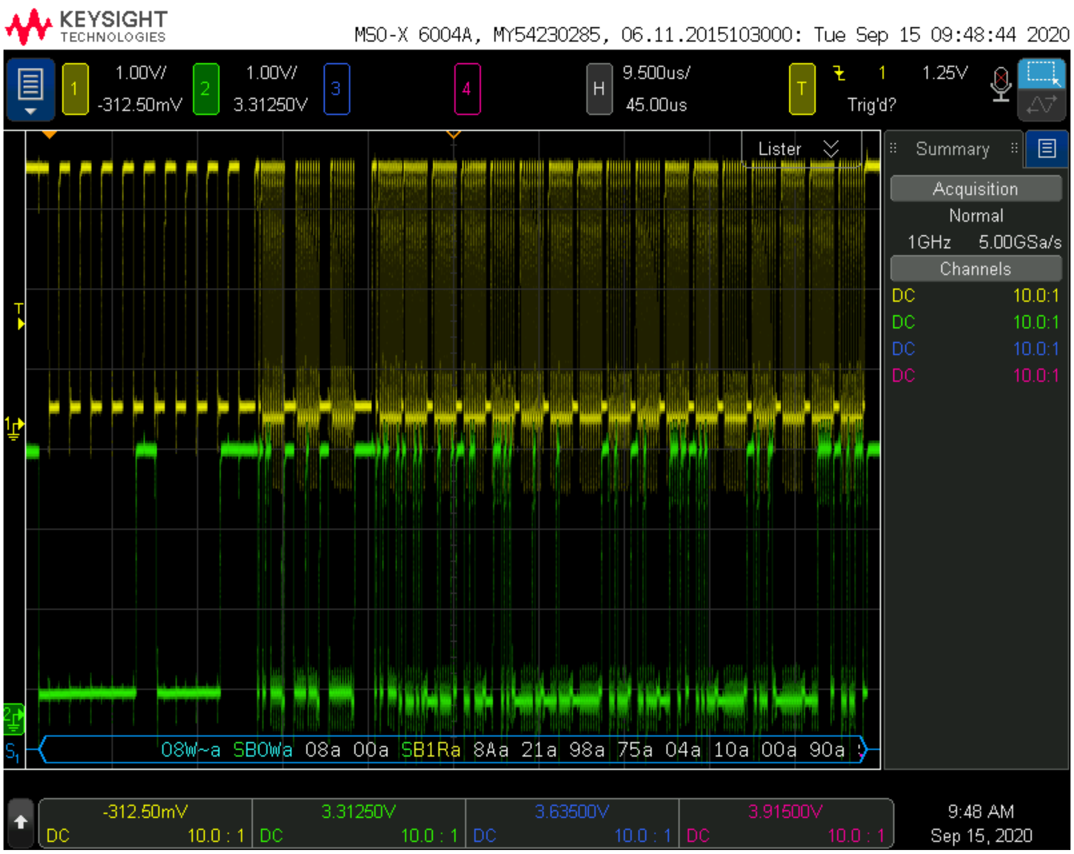


Serial Number Read

The first 16 bytes of the Security register are, by definition, read-only and contain a preprogrammed, ensured unique 128-bit serial number. There isn't a dedicated function to read the serial number; rather a random read operation is used with the correct device and word address bytes to read the serial number. Figure 11 depicts a High-Speed Serial Number read operation.

> **Note:** When using the i2c_randomRead_isr function, it is necessary to wait after the read operation using the isr_isBusy () polling function to ensure the data buffer is valid.

FIGURE 11: SERIAL NUMBER READ OPERATION

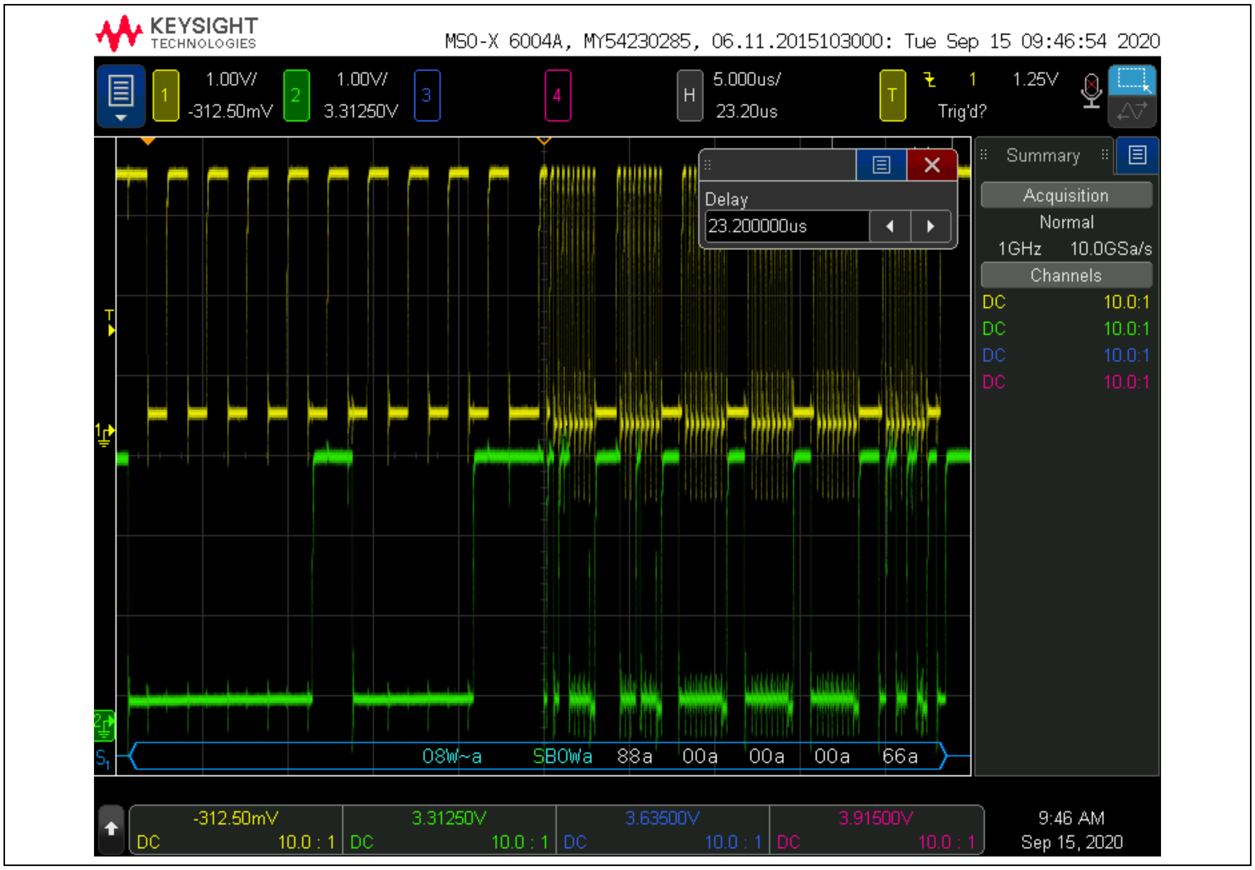


Configuration Register Write

In order to write the Configuration register, a confirmation byte must be sent to the device. This is done automatically in the firmware and the firmware determines the confirmation byte value sent by determining the new LOCK bit state. Figure 12 depicts a High-Speed Configuration Register write operation.

Note:	When performing a write operation, it is necessary to wait the max write cycle time (T_{WC}) before performing additional operations.

FIGURE 12: CONFIGURATION REGISTER WRITE OPERATION

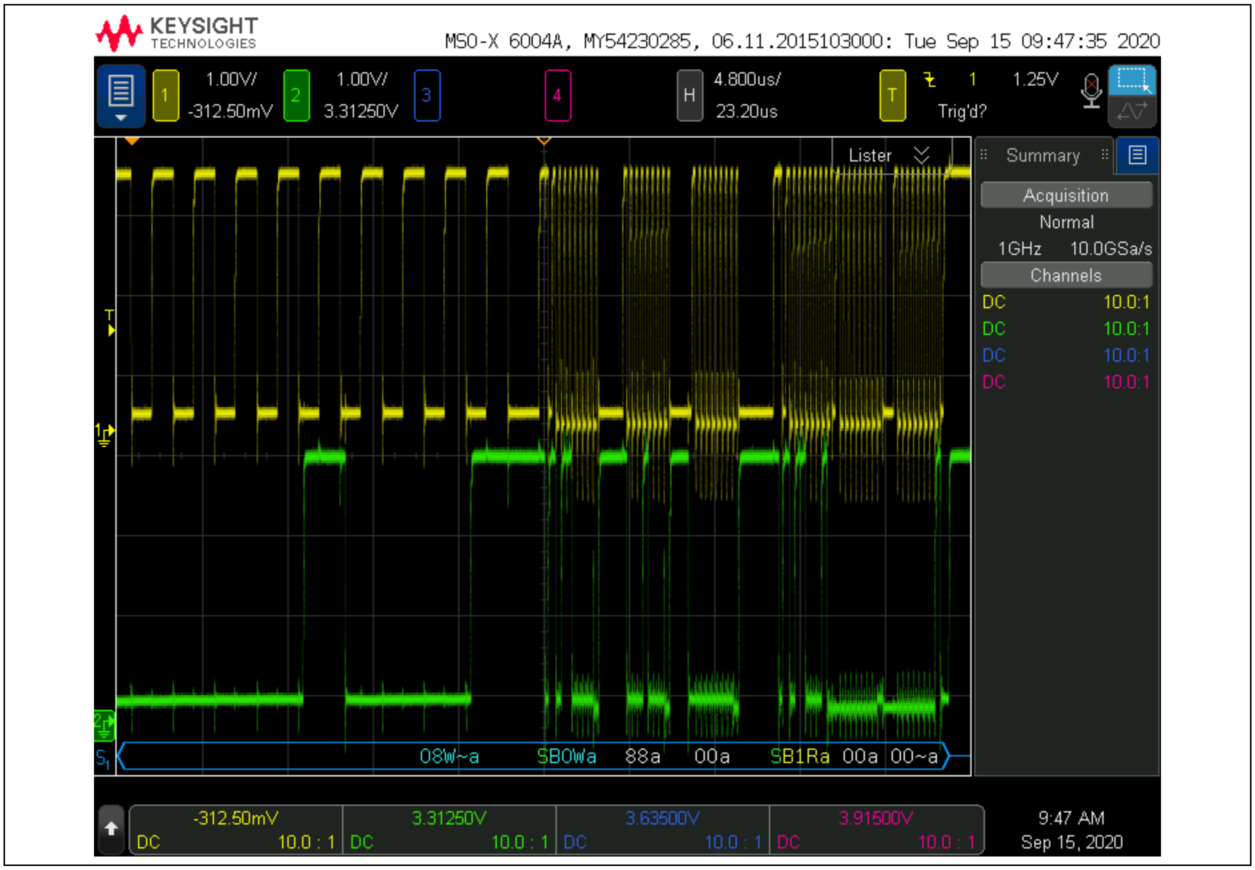


Configuration Register Read Operation

To access the Configuration register, a specific device and word address bytes must be sent to the device. Refer to the appropriate device data sheet for the device and word address byte requirements. There isn't a dedicated function to read the Configuration register; rather a random read operation is used with the correct device address and word address bytes to read the Configuration register. Figure 13 depicts High-Speed Configuration Register read operation.

Note:	When using the i2c_randomRead_isr function, it is necessary to wait after the read operation using the isr_isBusy () polling function to ensure the data buffer is valid.

FIGURE 13: CONFIGURATION REGISTER READ OPERATION



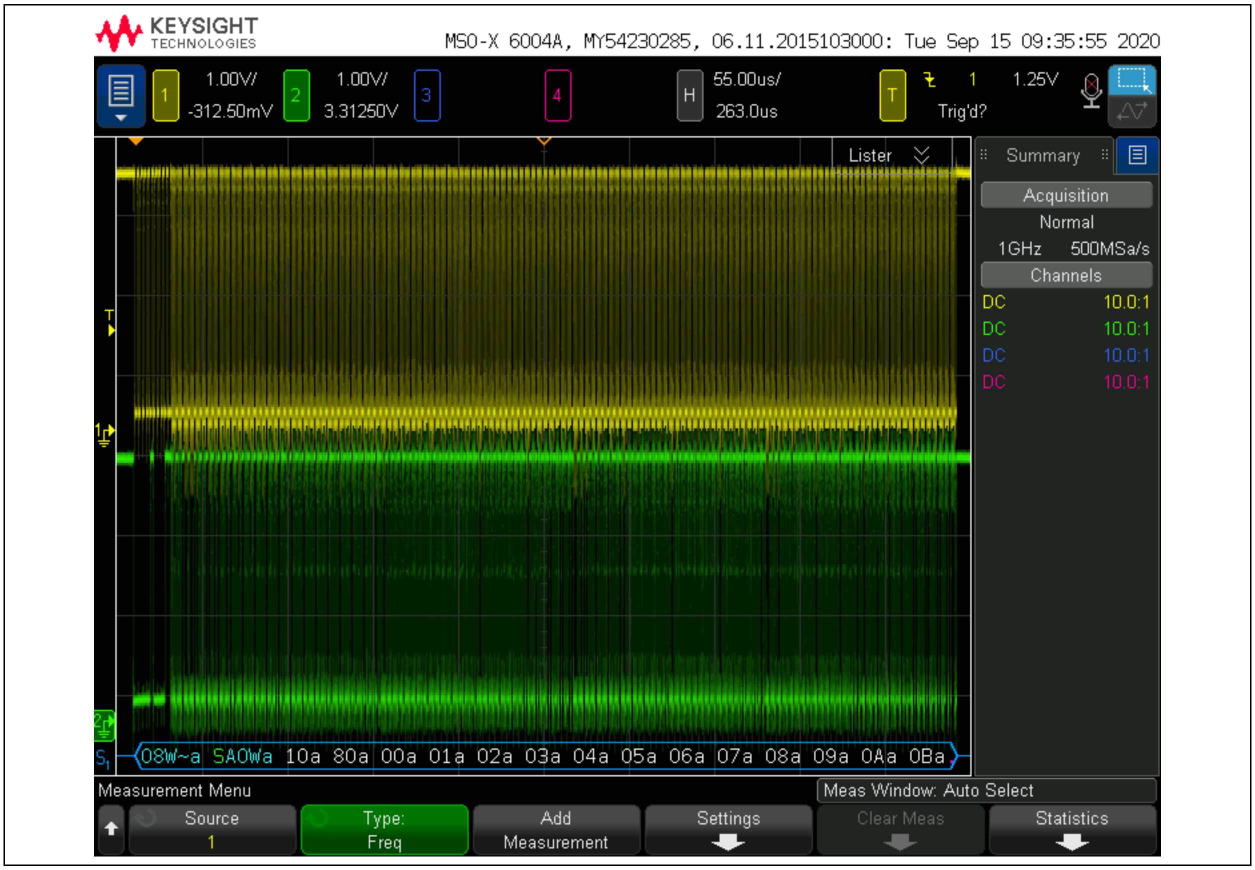
Page Write Operation

The firmware includes a page write operation, where 128 bytes are written starting at address 1080h. The number of bytes to be written and word address can be modified by the user as he or she chooses. High-Speed mode is exited after every write operation and therefore must be entered prior to write operation.

Figure 14 depicts a High-Speed page write operation.

Note:	When performing a write operation, it is necessary to wait the max write cycle time (T_{WC}) before performing additional operations.

FIGURE 14: PAGE WRITE OPERATION

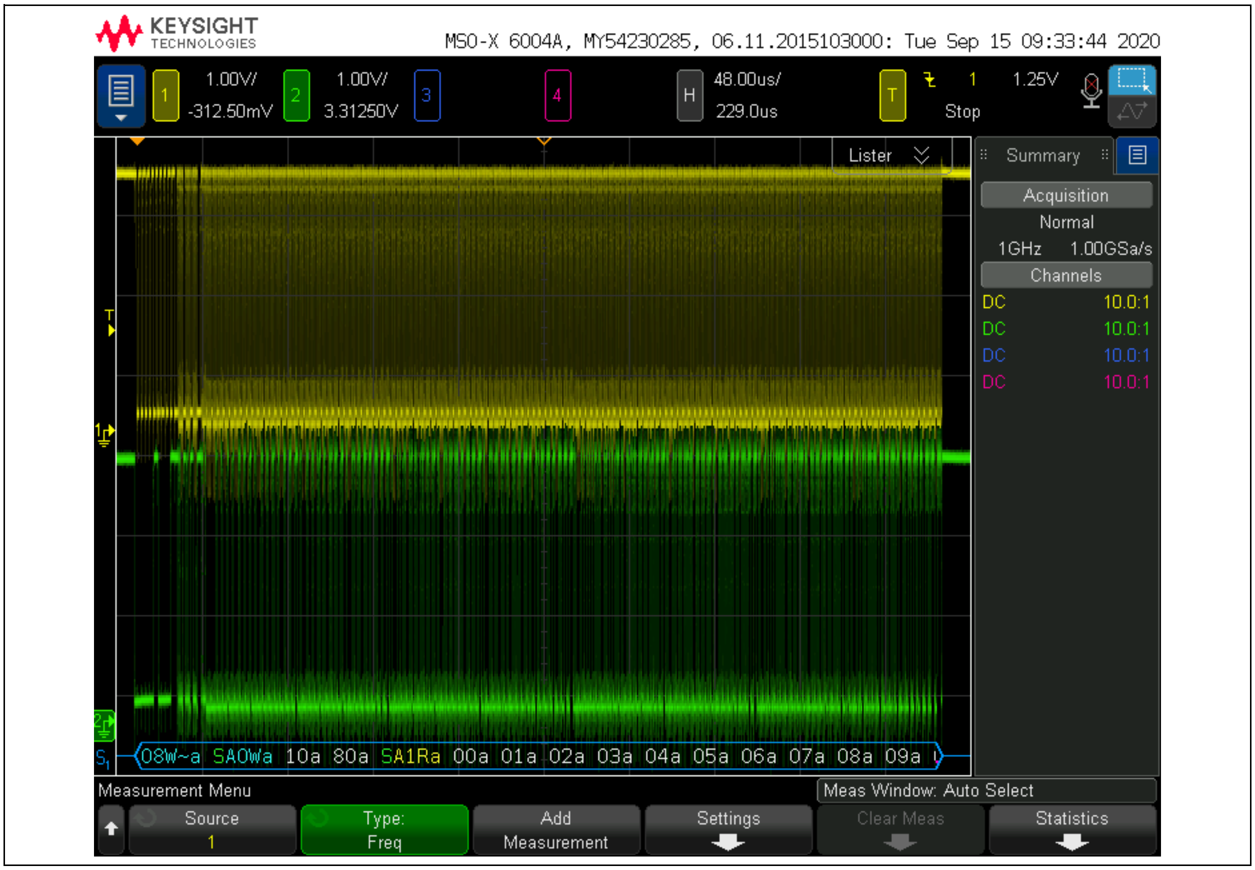


Page Read Operation

The firmware includes a page read operation, where 128 bytes are read starting at address 1080h. The number of bytes to be written and word address can be modified by the user as he or she chooses. Figure 15 depicts a High-Speed page read operation.

Note:	When using the i2c_randomRead_isr function, it is necessary to wait after the read operation using the isr_isBusy () polling function to ensure the data buffer is valid.

FIGURE 15: PAGE READ OPERATION



CONCLUSION

This application note offers designers a set of firmware routines to access I^{2}C Serial EEPROMs using High-Speed mode and the ATSAMD21J18A microcontroller. All routines were written in C using Atmel Studio 7 Integrated Development Environment V7.0.1931 and a portion of the code was generated using Atmel Advanced Software Framework (ASF). The hardware used in this application note is the SAM D21 Xplained Pro Evaluation Kit. Details related to I^{2}C High-Speed mode and device operation can be found in the appropriate device data sheet found at www.microchip.com.

APPENDIX A: REVISION HISTORY

Revision A (06/2021)

Initial release of this document.

ISBN: 978-1-5224-830-4

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: 45-4485-5910
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

02/28/20