MAX32664 User Guide

UG6806; Rev 0; 01/19

Abstract

The MAX32664 user guide provides flow charts, timing diagrams, GPIOs/pin usage, I²C interface protocol, and annotated I²C traces between the host microcontroller and the MAX32664. Typical application uses the MAX32664 as a low-power microcontroller in a sensor hub configuration to provide processed data such as heart rate and SpO_2.

Table of Contents

List of Figures

List of Tables

Introduction

The MAX32664 is a pre-programmed microcontroller with firmware drivers and algorithms. Combined with the appropriate sensor devices, the MAX32664 acts as a sensor hub to provide processed data to a host device. This solution seamlessly enables customers to receive raw and/or calculated data from Maxim's optical sensor solutions, while keeping overall system power consumption in check. The tiny form factor (1.6mm x 1.6mm 16-bump WLP) allows for integration into extremely small applications. The MAX32664 is integrated into Maxim's complete reference design solutions, which shortens the time to market.

The MAX32664 is the same hardware as the MAX32660 but with a pre-programmed bootloader that accepts in-application programming (IAP) of Maxim supplied algorithms and sensor drivers. The MAX32664 provides a fast-mode, I2C slave interface to a microcontroller host. A second I2C interface is dedicated to communicating with sensors.

For further details on memory, register mapping, system clocks, reset, power management, GPIOs/alternate functions, DMA controller, UART, RTC, timers, WDT, I2C, and SPI, see the MAX32660 User Guide.

For ordering information, mechanical and electrical characteristics, and the pinout for the MAX32664 family of devices, refer to the MAX32664 data sheet.

For information on the Arm® Cortex®-M4 with FPU core, refer to the Cortex-M4 with FPU Technical Reference Manual.

MAX32664 Variants

The MAX32664 is pre-programmed with bootloader software that accepts in-application programming of Maxim supplied algorithms and the associated sensor drivers. The MAX32664 is used as a sensor hub controller.

The algorithm/application code provides processed and/or raw data through the I2C interface. Several variants of the MAX32664 exist based on the target application. These variants come pre-programmed with a bootloader that only accepts the matching encryption keys for the part (e.g., the MAX32664GWEA bootloader is pre-programmed with the A encryption key, reference designs are programmed with Z keying, etc.). Designers should use the table below in order to select the correctly keyed part.

Table 1. MAX32664 Variants, Matching Algorithms, and Reference Designs

PART NUMBER	APPLICATION FIRMWARE	BOOTLOADER KEY	MAXIM REFERENCE DESIGN
MAX32664GWEA	MaximFast: Maxim Integrated® finger-based heart-rate and SpO_2 monitoring algorithm (100Hz sampling). The MaximFast algorithm is compatible with the sensor hub combination of the MAX32446GWEA, MAX30101 AFE, and KX-122 accelerometer. It is recommended, but not mandatory, to use an accelerometer with the MaximFast algorithm. Do not enable the accelerometer if there is no accelerometer in your design. If the KX-122 accelerometer is not installed in the design and external accelerometer data is supplied, then the accelerometer should use the 100Hz sampling rate. Automatic gain control (AGC) for MaximFast. If the AGC is enabled, the LED currents and pulse width are automatically determined by the algorithm. If the AGC is not enabled, the LED currents and pulse width registers should be configured by the host software.	A	MAXREFDES220#

For all the MAX32664 parts, the algorithm (.msbl file) with the corresponding bootloader key must be downloaded, and these parts must be programmed using the in-application programming feature of the bootloader. The MAX32664 is only pre-programmed with the bootloader software.

Maxim Reference Designs with MAX32664

Maxim provides multiple reference designs to its customers to enable quick and effective adoption of MAX32664 and fastest time to market. For detailed schematics, refer to the user guide of each reference design.

MAXREFDES220#

The MAXREFDES220# reference design provides everything you need to quickly prototype your product to measure finger-based heart rate and blood oxygen saturation level (SpO_2).

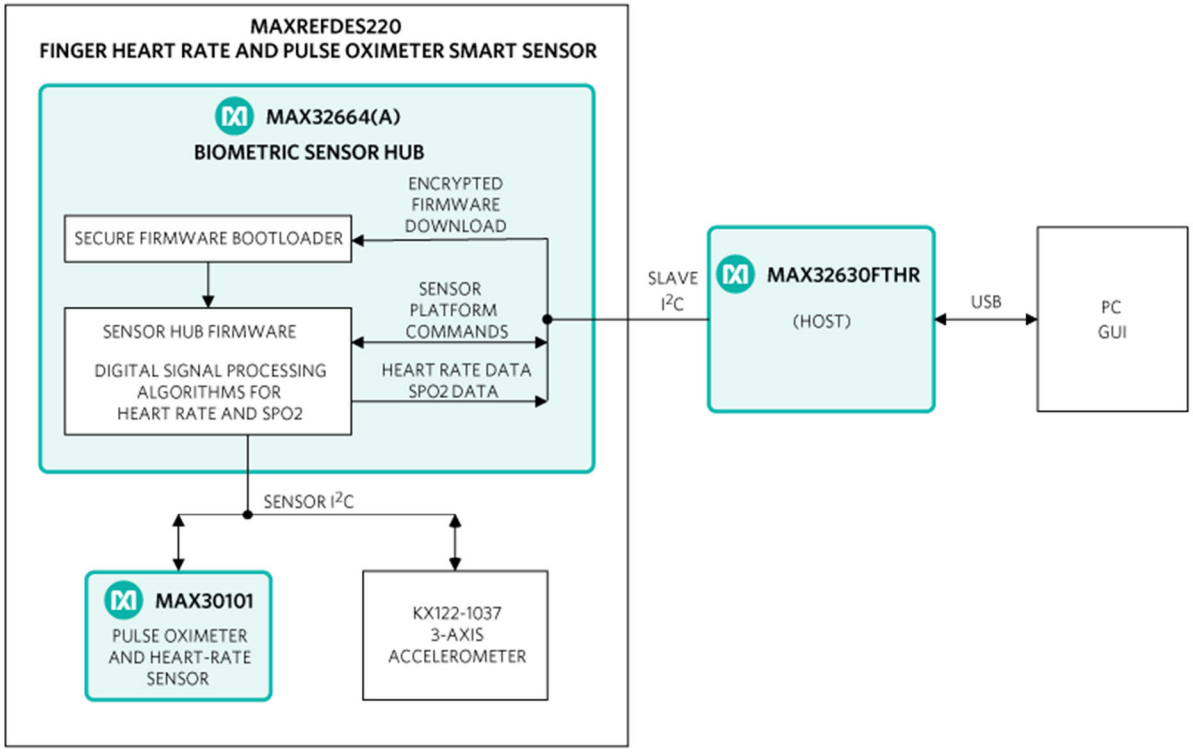


Figure 1. MAX32664 block diagram.

The MAXREFDES220# solution, which includes the MAX30101 and the MAX32664, provides an integrated hardware and software solution for finger-based applications. The MAX32664 is used as a sensor hub to collect data from the MAX30101 analog front end (AFE). The reference design also includes a tri-axis accelerometer (KX-122) to compensate for motion artifacts. (Accelerometer support in the MAXREFDES220# is optional.)

The MAX32630FTHR is used as a sample host is included in MAXREFDES220# reference design.

MAXREFDES101#

The MAXREFDES101# is a unique evaluation and development platform in a wrist-worn wearable form factor that demonstrates the functions of a wide range of Maxim's products for health-sensing applications.

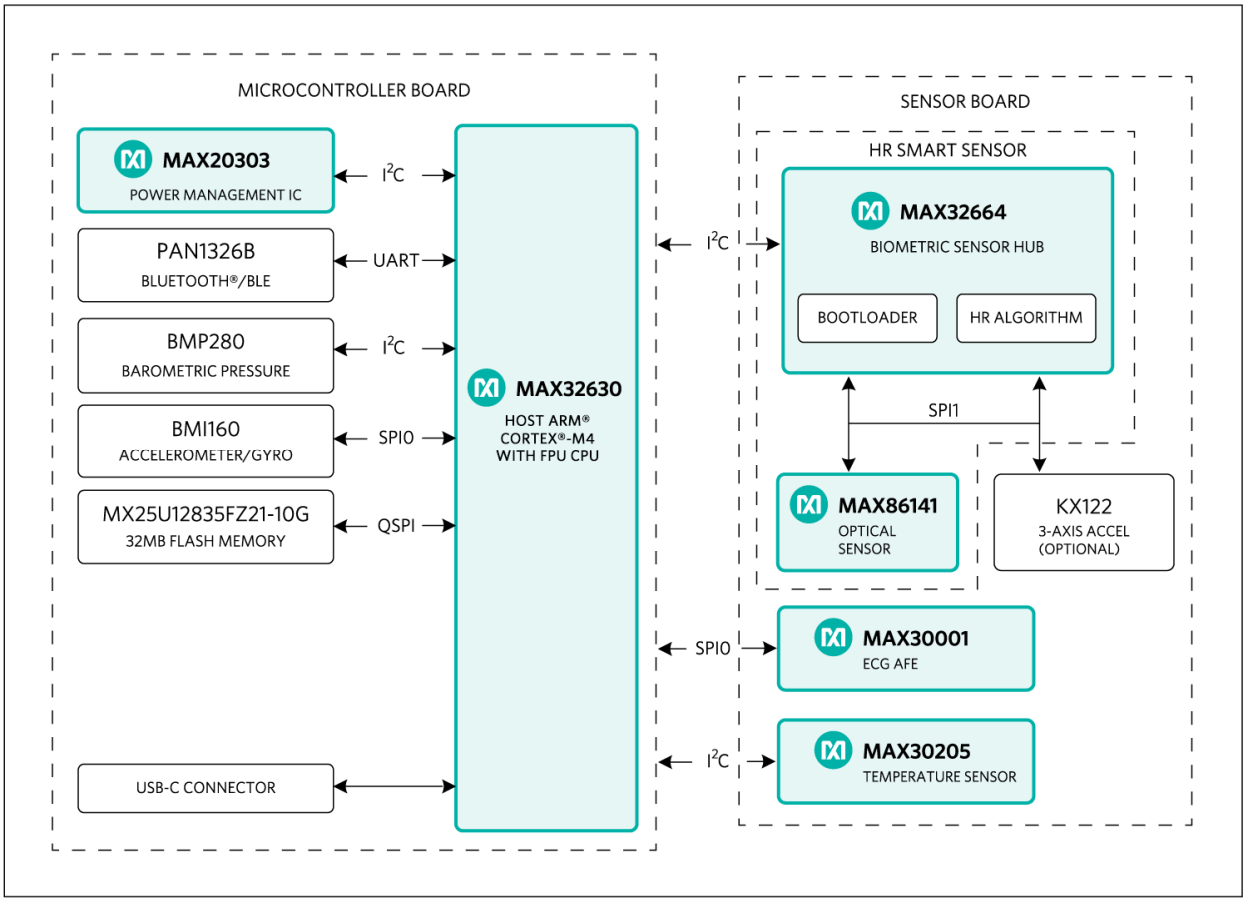


Figure 2. MAXREFDES101# block diagram.

This second-generation health sensor platform (a follow-on to the MAXREFDES100#) integrates a PPG AFE sensor (MAX86141), a biopotential AFE (MAX30001), a human body temperature sensor (MAX30205), a microcontroller (MAX32630), a power-management IC (MAX20303), and a 6-axis accelerometer/gyroscope. The complete platform includes a watch enclosure and a biometric sensor hub with an embedded heart-rate algorithm (MAX32664). Algorithm output and raw data can be streamed through Bluetooth® to an Android® application or PC GUI for demonstration, evaluation, and customized development.

MAX32664 GPIOs and RSTN Pin

To control and communicate with the MAX32664, the RSTN pin and GPIOs P0.1, P0.2, P0.3 of the MAX32664 are connected to the host as pictured in Figure 3.

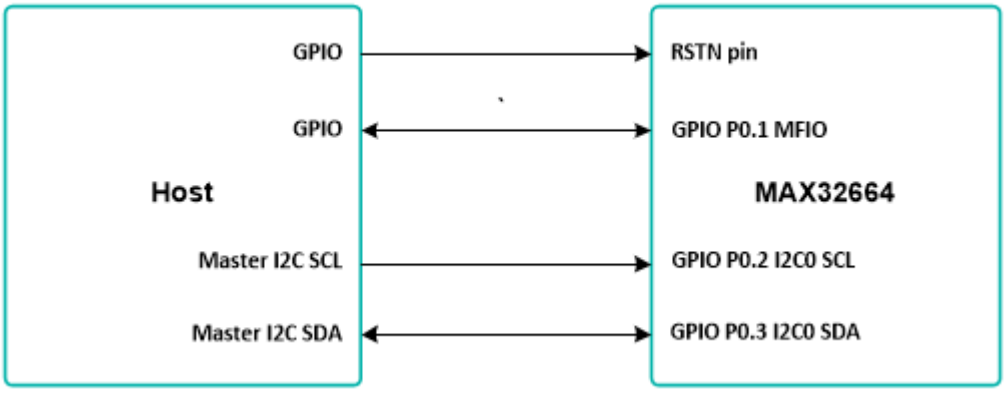


Figure 3. Pin connections between the host and the MAX32664.

The RSTN pin is used in conjunction with the GPIO P0.1 MFIO pin to control whether the MAX32664 starts up in Application mode or Bootloader mode. While in application mode, the MFIO pin can be configured to provide an interrupt signal to the host.

The host acts an I2C master to communicate with the MAX32664. GPIO P0.2 is used as the SCL line and GPIO P0.3 is used as the SDA line.

Table 2. RSTN Pin and GPIOs Pins

MAX32664	DESCRIPTION	DIRECTION FROM THE MAX32664 SIDE
Pin RSTN	Reset_N	Input
GPIO P0.1	GPIO MFIO interrupt to host, wake from host, bootloader/application on powerup,	Input/Output
GPIO P0.2	I2C0_Host SCL	Input
GPIO P0.3	I2C0 Host SDA	Input/Output

Variations of the MAX32664 use additional GPIO pins in order to communicate and control sensor devices. For example, in the MAXREFDES220#, the additional GPIOs listed in Table 3 are used to control the sensors used.

Table 3. Additional GPIOs Used on the MAX32664 for the MAXREFDES220#

MAX32664	DESCRIPTION	DIRECTION FROM THE MAX32664 SIDE
GPIO P0.6	KX122 ACCEL Interrupt	Input
GPIO P0.7	MAX30101 Interrupt	Input
GPIO P0.8	MAX30101, KX122 I2C1_SCL	Output
GPIO P0.9	MAX30101, KX122 I2C1_SDA	Input/Output

MAX32664 Bootup and Application Mode

The MAX32664 is programmed to enter either bootloader mode or application mode at the start-up based on the state of the MFIO pin.

Variations of the MAX32664 part are pre-programmed with the different algorithms and application firmware. Check with your Maxim representative.

MAX32664 Bootloader Mode

The MAX32664 enters bootloader mode based on the sequencing of the RSTN pin and the MFIO pin. The necessary sequence is as follows:

- Set the RSTN pin low for 10ms.
- While RSTN is low, set the MFIO pin to low.
- After the 10ms has elapsed, set the RSTN pin high.
- After an additional 50ms has elapsed, the MAX32664 is in bootloader mode.

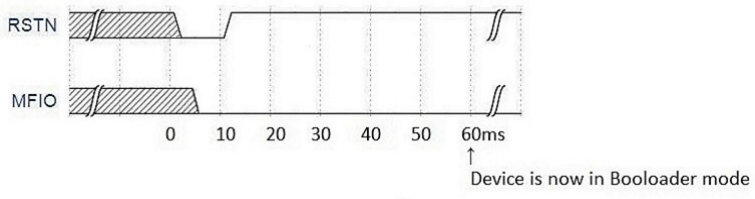


Figure 4. Entering bootloader mode using the RSTN pin and the MFIO GPIO pin.

MAX32664 Application Mode

The MAX32664 enters application mode based on the sequencing of the RSTN pin and the MFIO pin. The necessary sequence is as follows:

- Set the RSTN pin low for 10ms.
- While RSTN is low, set the MFIO pin to high.
- After the 10ms has elapsed, set the RSTN pin high.
- After an additional 50ms has elapsed, the MAX32664 is in application mode.

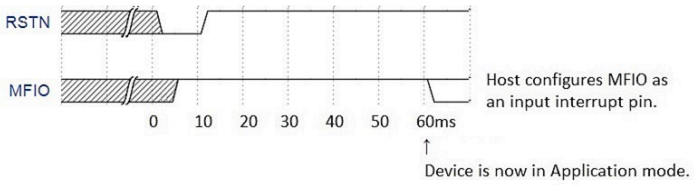


Figure 5. Entering application mode using the RSTN pin and MFIO pin.

Communications to the MAX32664 over I2C

The host communicates to the MAX32664 through the I2C bus. The MAX32664 uses 0xAA as the I2C 8-bit slave write address and 0XAB is used as the I2C 8-bit slave read address. The maximum I2C data rate supported is 3400 Kbps.

Bit Transfer Process

Both SDA and SCL signals are open-drain circuits. Each has an external pullup resistor that ensures each circuit is high when idle. The I2C specification states that during data transfer, the SDA line can change state only when SCL is low, and that SDA is stable and able to be read when SCL is high. Typical I2C write/read transactions are shown in Figure 6.

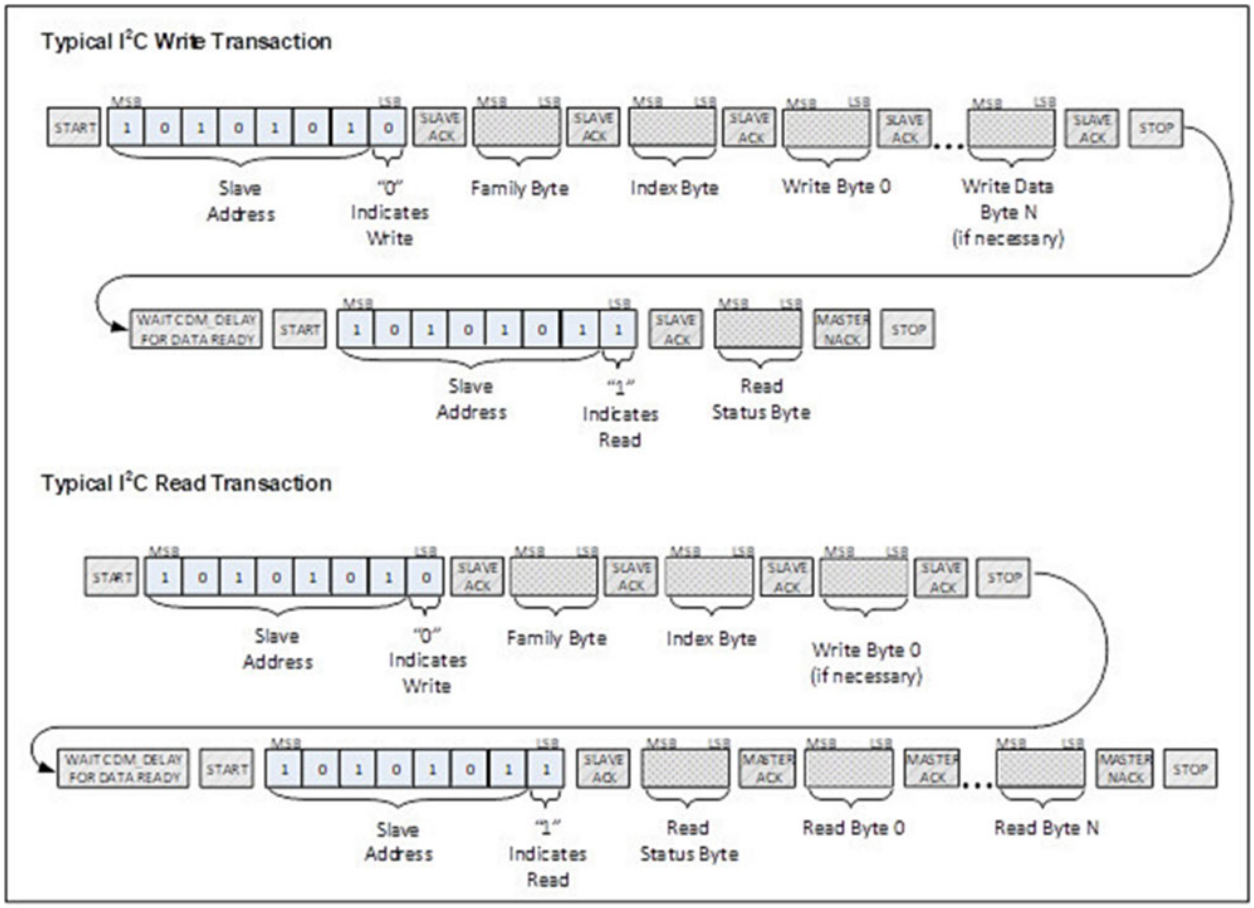


Figure 6. I2C Write/Read data transfer from host microcontroller.

The read status byte is an indicator of the success or failure of the Write Transaction. The read status byte must be accessed after each write transaction to the device. This ensures that write transaction processing is understood and any errors in the device command handling can be corrected. The value of the read status byte is summarized in Table 4.

Table 4. Read Status Byte Value

STATUS BYTE VALUE	DESCRIPTION
0x00	SUCCESS. The write transaction was successful.
0x01	ERR_UNAVAIL_CMD. Illegal Family Byte and/or Command Byte was used.
0x02	ERR_UNAVAIL_FUNC. This function is not implemented.
0x03	ERR_DATA_FORMAT. Incorrect number of bytes sent for the requested Family Byte.
0x04	ERR_INPUT_VALUE. Illegal configuration value was attempted to be set.
0x05	ERR_TRY_AGAIN. Device is busy. Try again.
0x80	ERR_BTLDR_GENERAL. General error while receiving/flashing a page during the bootloader sequence.
0x81	ERR_BTLDR_CHECKSUM. Checksum error while decrypting/checking page data.
0x82	ERR_BTLDR_AUTH. Authorization error.
0x83	ERR_BTLDR_INVALID_APP. Application not valid.
0xFF	ERR_UNKNOWN. Unknown Error.

I2C Write

The process for an I2C write data transfer is as follows:

1. The bus master indicates a data transfer to the device with a START condition.
2. The master transmits one byte with the 7-bit slave address (most significant 7 bits of the 8-bit address) and a single write bit set to zero. The eight bits to be transferred as a slave address for the MAX32664 is 0xAA for a write transaction.
3. During the next SCL clock following the write bit, the master releases SDA. During this clock period, the device responds with an ACK by pulling SDA low.
4. The master senses the ACK condition and begins to transfer the Family Byte. The master drives data on the SDA circuit for each of the eight bits of the Family byte, and then floats SDA during the ninth bit to allow the device to reply with the ACK indication.
5. The master senses the ACK condition and begins to transfer the Index Byte. The master drives data on the SDA circuit for each of the eight bits of the Index byte, and then floats SDA during the ninth bit to allow the device to reply with the ACK indication.
6. The master senses the ACK condition and begins to transfer the Write Data Byte 0. The master drives data on the SDA circuit for each of the eight bits of the Write Data Byte 0, and then floats SDA during the ninth bit to allow the device to reply with the ACK indication.
7. The master senses the ACK condition and can begin to transfer another Write Data Byte if required. The master drives data on the SDA circuit for each of the eight bits of the Write Data Byte, and then floats SDA during the ninth bit to allow the device to reply with the ACK indication. If another Write Data Byte is not required, the master indicates the transfer is complete by generating a STOP condition. A STOP condition is generated when the master pulls SDA from a low to high while SCL is high.
8. The master waits for a period of CMD_DELAY (60µs) for the device to have its data ready.
9. The master indicates a data transfer to a slave with a START condition.
10. The master transmits one byte with the7-bit slave address and a single write bit set to one. This is an indication from the master of its intent to read the device from the previously written location defined by the Family Byte and the Index Byte. The master then floats SDA and allows the device to drive SDA to send the Status Byte. The Status Byte reveals the success of the previous write sequence. After the Status Byte is read, the master drives SDA low to signal the end of data to the device.
11. The master indicates the transfer is complete by generating a STOP condition.
12. After the completion of the write data transfer, the Status Byte must be analyzed to determine if the write sequence was successful and the device has received the intended command.

I2C Read

The process for an I²C read data transfer is as follows:

1. The bus master indicates a data transfer to the device with a START condition.
2. The master transmits one byte with the 7-bit slave address and a single write bit set to zero. The eight bits to be transferred as a slave address for the MAX32664 is 0xAA for a write transaction. This write transaction precedes the actual read transaction to indicate to the device what section is to be read.
3. During the next SCL clock following the write bit, the master releases SDA. During this clock period, the device responds with an ACK by pulling SDA low.
4. The master senses the ACK condition and begins to transfer the Family Byte. The master drives data on the SDA circuit for each of the eight bits of the Family byte, and then floats SDA during the ninth bit to allow the device to reply with the ACK indication.
5. The master senses the ACK condition and begins to transfer the Index Byte. The master drives data on the SDA circuit for each of the eight bits of the Index byte, and then floats SDA during the ninth bit to allow the device to reply with the ACK indication.
6. The master senses the ACK condition and begins to transfer the Write Data Byte if necessary for the read instruction. The master drives data on the SDA circuit for each of the eight bits of the Write Data byte, and then floats SDA during the ninth bit to allow the device to reply with the ACK indication.
7. The master indicates the transfer is complete by generating a STOP condition.
8. The master waits for a period of CMD_DELAY (60µs) for the device to have its data ready.
9. The master indicates a data transfer to a slave with a START condition.
10. The master transmits one byte with the 7-bit slave address and a single write bit set to one. This is an indication from the master of its intent to read the device from the previously written location defined by the Family Byte and the Index Byte. The master then floats SDA and allows the device to drive SDA to send the Status Byte. The Status Byte reveals the success of the previous write sequence. After the Status Byte is read, the master drives SDA low to acknowledge the byte.
11. The master floats SDA and allows the device to drive SDA to send Read Data Byte 0. After Read Data Byte 0 is read, the master drives SDA low to acknowledge the byte.
12. The master floats SDA and allows the device to drive SDA to send the Read Data Byte N. After Read Data Byte N is read, the master drives SDA low to acknowledge the Read Data Byte N. This process continues until the device has provided all the data that the master expects based upon the Family Byte and Index Byte definition.
13. The master indicates the transfer is complete by generating a STOP condition.

MAX32664 I2C Message Protocol Definition

Table 5 defines the I2C message protocol for the MAX32664.

Table 5. MAX32664 I2C Message Protocol Definitions

HOST COMMAND					MAX32664
FAMILY NAME	**DESCRIPTION**	**FAMILY BYTE**	**INDEX BYTE**	**WRITE BYTES**	**RESPONSE BYTES**
Read Sensor Hub Status	Read sensor hub status	0x00	0x00	-	**Err0[0]**: 0 = No Error; 1 = Sensor Communication Problem **Err1[0]**: Not used **Err2[0]**: Not used **DataRdyInt[3]**: 0 = FIFO below threshold; 1 = FIFO filled to threshold or above. **FifoOutOvrInt[4]**: 0 = No FIFO overflow; 1 = Sensor Hub Output FIFO overflowed, data lost. **FifoInOvrInt[5]**: 0 = No FIFO overflow; 1 = Sensor Hub Input FIFO overflowed, data lost. **DevBusy[6]**: 0 = Sensor Hub ready; 1 = Sensor Hub is busy processing. See Table 6 for the for the bit field table.
Device Mode	Select the device operating mode. The application must implement this.	0x01	0x00	**0x00**: Exit bootloader mode. **0x02**: Reset. **0x08**: Enter bootloader mode.	-
Device Mode	Read the device operating mode.	0x02	0x00	-	**0x00**: Application operating mode. **0x08**: Bootloader operating mode.
Set Output Mode	Set the output format of the sensor hub.	0x10	0x00	**0x00**: Pause (no data) **0x01**: Sensor Data **0x02:** Algorithm Data **0x03:** Sensor Data and Algorithm Data **0x04**: Pause (no data) **0x05**: Sample Counter byte, Sensor Data **0x06:** Sample Counter byte, Algorithm Data **0x07:** Sample Counter byte, Sensor Data and Algorithm Data	-

HOST COMMAND					MAX32664
FAMILY NAME	DESCRIPTION	FAMILY BYTE	INDEX BYTE	WRITE BYTES	RESPONSE BYTES
Set Output Mode	Set the threshold for the FIFO interrupt bit/pin. The MFIO pin is used as the interrupt and the host should configure this pin as an input interrupt pin. The status bit DataRdyInt is set when this threshold is reached.	0x10	0x01	**0x01 to 0xFF:** Sensor Hub Interrupt Threshold for FIFO almost full.	-
Read Output FIFO	Get the number of samples available in the FIFO.	0x12	0x00	-	Number of samples available in the FIFO.
Read Output FIFO	Read data stored in output FIFO.	0x12	0x01	-	See Table 7, Output FIFO Format Definitions. The internal FIFO read pointer increments once the sample size bytes have been read.
Read Input FIFO for External Sensors (e.g., systems that have an externally supplied accelerometer)	Read the sensor sample size.	0x13	0x00	**0x04**: Accelerometer	**0x06:** Bytes per sample for the external accelerometer. Three 16-bit 2's complement with LSB = 0.001g. See Table 8 for an example.
Read Input FIFO for External Sensors	Read the input FIFO size for the maximum number of samples that the input FIFO can hold (16-bit).	0x13	0x01	-	MSB, LSB
Read Input FIFO for External Sensors	Read the sensor FIFO size for the maximum number of samples that the sensor FIFO can hold (16-bit).	0x13	0x02	**0x04**: Accelerometer	MSB, LSB
Read Input FIFO for External Sensors	Read the number of samples currently in the input FIFO (16-bit).	0x13	0x03	**0x04**: Accelerometer	MSB, LSB
Read Input FIFO for External Sensors	Read the number of samples currently in the sensor FIFO (16-bit).	0x13	0x04	-	MSB, LSB
Write Input FIFO for External Sensors	Write data to the input FIFO.	0x14	0x04	Number of Samples, Sample 1 values, ..., Sample n values See Table 8 for an example.	-

HOST COMMAND					MAX32664
FAMILY NAME	DESCRIPTION	FAMILY BYTE	INDEX BYTE	WRITE BYTES	RESPONSE BYTES
Write Register	Write a value to a writable MAX86140/ MAX86141 register.	0x40	0x00	Register address, Register value	-
Write Register	Write a value to a writable MAX30205 register.	0x40	0x01	Register address, Register value	-
Write Register	Write a value to a writable MAX30001 register.	0x40	0x02	Register address, Register value	-
Write Register	Write a value to a writable MAX30101 register.	0x40	0x03	Register address, Register value	-
Write Register	Write a value to a writable accelerometer sensor register.	0x40	0x04	Register address, Register value	-
Read Register	Read the value of a MAX86140/ MAX86141 register.	0x41	0x00	Register Address	Register value
Read Register	Read the value of a MAX30205 register.	0x41	0x01	Register Address	Register value
Read Register	Read the value of a MAX30001 register.	0x41	0x02	Register Address	Register value
Read Register	Read the value of a MAX30101 register.	0x41	0x03	Register Address	Register value
Read Register	Read the value of an accelerometer sensor register.	0x41	0x04	Register Address	Register value
Get Attributes of the AFE	Retrieve the attributes of the MAX86140/ MAX86141 AFE.	0x42	0x00	-	Number of bytes in a word for this sensor, Number of registers available for this sensor.
Get Attributes of the AFE	Retrieve the attributes of the MAX30205 AFE.	0x42	0x01	-	Number of bytes in a word for this sensor, Number of registers available for this sensor.
Get Attributes of the AFE	Retrieve the attributes of the MAX30001 AFE.	0x42	0x02	-	Number of bytes in a word for this sensor, Number of registers available for this sensor.
Get Attributes of the AFE	Retrieve the attributes of the MAX30101 AFE.	0x42	0x03	-	Number of bytes in a word for this sensor, Number of registers available for this sensor.
Get Attributes of the AFE	Retrieve the attributes of the accelerometer sensor AFE.	0x42	0x04	-	Number of bytes in a word for this sensor, Number of registers available for this sensor.
Dump Registers	Read all the MAX86140/ MAX86141 registers.	0x43	0x00	-	Register address 0, register value 0, register address 1, register value 1, ..., register address n, register value n
Dump Registers	Read all the MAX30205 registers.	0x43	0x01	-	Register address 0, register value 0, register address 1, register value 1, ..., register address n, register value n

HOST COMMAND					MAX32664
FAMILY NAME	DESCRIPTION	FAMILY BYTE	INDEX BYTE	WRITE BYTES	RESPONSE BYTES
Dump Registers	Read all the MAX30001 registers.	0x43	0x02	-	Register address 0, register value 0, register address 1, register value 1, ..., register address n, register value n
Dump Registers	Read all the MAX30101 registers.	0x43	0x03	-	Register address 0, register value 0, register address 1, register value 1, ..., register address n, register value n
Dump Registers	Read all the accelerometer sensor registers.	0x43	0x04	-	Register address 0, register value 0, register address 1, register value 1, ..., register address n, register value n
Sensor Mode Enable	Enable the MAX86140/ MAX86141 sensor.	0x44	0x00	**0x00**: Disable **0x01**: Enable	-
Sensor Mode Enable	Enable the MAX30205 sensor.	0x44	0x01	**0x00**: Disable **0x01**: Enable	-
Sensor Mode Enable	Enable the MAX30001 sensor.	0x44	0x02	**0x00**: Disable **0x01**: Enable	-
Sensor Mode Enable	Enable the MAX30101 sensor.	0x44	0x03	**0x00**: Disable **0x01**: Enable	-
Sensor Mode Enable	Enable the external accelerometer sensor. This command is used when host accelerometer data is supplied to the sensor hub.	0x44	0x04	**0x00**: Disable **0x01**: Enable	-
Algorithm Configuration	Automatic Gain Control (AGC) algorithm: Set the target percentage of the full-scale ADC range that the automatic gain control (AGC) algorithm uses.	0x50	0x00	0x00, 0 to 100 percent	-
Algorithm Configuration	AGC algorithm: Set the step size toward the target for the AGC algorithm.	0x50	0x00	0x01, 0 to 100 percent	-
Algorithm Configuration	AGC algorithm: Set the sensitivity for the AGC algorithm.	0x50	0x00	0x02. 0 to 100 percent	-
Algorithm Configuration	AGC algorithm: Set the number of samples to average for the AGC algorithm.	0x50	0x00	0x03, Number of samples to average (range is 0 to 255).	-

HOST COMMAND					MAX32664
FAMILY NAME	DESCRIPTION	FAMILY BYTE	INDEX BYTE	WRITE BYTES	RESPONSE BYTES
Algorithm Configuration	Wrist Heart Rate Monitor (WHRM) algorithm: Set the sample rate for the WHRM algorithm. The WHRM algorithm is configured to use the LED1 and Photo-diode 1 (25Hz sample rate) and is compatible with the MAX86141 AFE, KX-122 accelerometer.	0x50	0x02	0x00, MSB of sample rate, LSB of sample rate (16-bit unsigned integer)	-
Algorithm Configuration	WHRM algorithm: Set the maximum allowed height (cm).	0x50	0x02	0x01, MSB of height, LSB of height (16-bit unsigned integer)	-
Algorithm Configuration	WHRM algorithm: Set the maximum allowed weight (kg).	0x50	0x02	0x02, MSB of weight, LSB of weight (16-bit unsigned integer)	-
Algorithm Configuration	WHRM algorithm: Set the maximum allowed age (years).	0x50	0x02	0x03, Age	-
Algorithm Configuration	WHRM algorithm: Set the minimum allowed height (cm).	0x50	0x02	0x04, MSB of height, LSB of height (16-bit unsigned integer)	-
Algorithm Configuration	WHRM algorithm: Set the minimum allowed weight (kg).	0x50	0x02	0x05, MSB of weight, LSB of weight (16-bit unsigned integer)	-
Algorithm Configuration	WHRM algorithm: Set the minimum allowed age (years).	0x50	0x02	0x06, Age	-
Algorithm Configuration	WHRM algorithm: Set the default height (cm).	0x50	0x02	0x07, MSB of height, LSB of height (16-bit unsigned integer)	-
Algorithm Configuration	WHRM algorithm: Set the default weight (kg).	0x50	0x02	0x08, MSB of weight, LSB of weight (16-bit unsigned integer)	-
Algorithm Configuration	WHRM algorithm: Set the default age (years).	0x50	0x02	0x09, Age	-
Algorithm Configuration	WHRM algorithm: Set the initial heart-rate value in bpm which might speed up the algorithm.	0x50	0x02	0x0A, Heart rate (bpm)	-

HOST COMMAND					MAX32664
FAMILY NAME	DESCRIPTION	FAMILY BYTE	INDEX BYTE	WRITE BYTES	RESPONSE BYTES
Algorithm Configuration	MaximFast algorithm: Set the MaximFast SpO_2 coefficients.	0x50	0x02	0x0B, four bytes signed integer A, four bytes signed integer B, four bytes signed integer C (32-bit integers which are the coefficients times 100,000) The MAXREFDES220# without the cover glass uses the following coefficients as the default values: A = 159584 B = -3465966 C = 11268987	-
Algorithm Configuration	WHRM algorithm: Enable automatic exposure control (AEC) algorithm.	0x50	0x02	**0x0B, 0x00**: Disable **0x0B, 0x01**: Enable	-
Algorithm Configuration	WHRM algorithm: Enable skin contact detection (SCD) algorithm.	0x50	0x02	**0x0C, 0x00**: Disable **0x0C, 0x01**: Enable	-
Algorithm Configuration	WHRM algorithm: Set adjusted target photo detector (PD) current period in seconds.	0x50	0x02	0x0D, MSB, LSB (unsigned 16-bit integer)	-
Algorithm Configuration	WHRM algorithm: Set SCD debounce window.	0x50	0x02	0x0E, MSB, LSB (unsigned 16-bit integer)	-
Algorithm Configuration	WHRM algorithm: Set motion magnitude threshold in 0.1g.	0x50	0x02	0x0F, MSB, LSB (unsigned 16-bit integer)	-
Algorithm Configuration	WHRM algorithm: Set the minimum PD current in 0.1mA.	0x50	0x02	0x10, MSB, LSB (unsigned 16-bit integer)	-
Algorithm Configuration	WHRM algorithm: Configure the source of the PPG signal for the PD.	0x50	0x02	**0x11, 0x01**: PD1 **0x11, 0x02**: PD2 **0x11, 0x03:** PD1 and PD2	-
Algorithm Configuration	Blood Pressure Trending (BPT) algorithm: Set if the user is on blood pressure medication.	0x50	0x04	**0x00, 0x00**: Not using blood pressure (BP) medication **0x00, 0x01**: Using BP medication	-
Algorithm Configuration	BPT algorithm: Write the three samples of the diastolic BP byte values needed by the calibration procedure.	0x50	0x04	0x01, diastolic value 1, diastolic value 2, diastolic value 3	-

HOST COMMAND					MAX32664
FAMILY NAME	DESCRIPTION	FAMILY BYTE	INDEX BYTE	WRITE BYTES	RESPONSE BYTES
Algorithm Configuration	BPT algorithm: Write the three samples of the systolic BP byte values needed by the calibration procedure.	0x50	0x04	0x02, systolic value 1, systolic value 2, systolic value 3	-
Algorithm Configuration	BPT algorithm: Calibrate the BPT algorithm using the result of the calibration procedure. (Use the data from the 0x51 0x04 0x03 command).	0x50	0x04	0x03, 608 bytes of calibration data	-
Algorithm Configuration	BPT algorithm: Set the estimation date, using the current date. The year is sent as a 16-bit unsigned integer, and the month and day are bytes.	0x50	0x04	0x04, year MSB, year LSB, month, day	-
Algorithm Configuration	BPT algorithm: Configure whether the user is not resting or resting.	0x50	0x04	**0x05, 0x00**: Resting **0x05, 0x01**: Not resting	-
Algorithm Configuration	BPT algorithm: Set the SpO_2 coefficients A, B, C. Defaults: C = 0x42E16137 B = 0xC20AA37F A = 3FCC448F	0x50	0x04	0x06, four bytes float C, four bytes float B, four bytes float A	-
Algorithm Configuration	SpO_2 on the wrist ($WSpO_2$) algorithm: Set the $WSpO_2$ coefficients. $WSpO_2$ is a Maxim supplied algorithm for measuring SpO_2 on the wrist. Defaults: A = 159584 B = -3465966 C = 11268987	0x50	0x05	0x00, four bytes signed integer A, four bytes signed integer B, four bytes signed integer C (32-bit integers which are the coefficients times 100,000).	-
Algorithm Configuration	$WSpO_2$ algorithm: Set the sample rate.	0x50	0x05	**0x01, 0x00**: 100Hz **0x01, 0x01**: 25Hz	-
Algorithm Configuration	$WSpO_2$ algorithm: Set the algorithm run mode.	0x50	0x05	**0x02, 0x00:** Continuous **0x02, 0x01:** One-shot	-

HOST COMMAND					MAX32664
FAMILY NAME	DESCRIPTION	FAMILY BYTE	INDEX BYTE	WRITE BYTES	RESPONSE BYTES
Algorithm Configuration	$WSpO_2$ algorithm: Set the AGC mode for the $WSpO_2$ algorithm.	0x50	0x05	**0x03, 0x00:** Disable **0x03, 0x01:** Enable	-
Algorithm Configuration	$WSpO_2$ algorithm: Set motion detection.	0x50	0x05	**0x04, 0x00:** Disable **0x04, 0x01:** Enable	-
Algorithm Configuration	$WSpO_2$ algorithm: Set the motion detection period.	0x50	0x05	0x05, MSB of Period, LSB of Period (16-bit unsigned integer)	-
Algorithm Configuration	$WSpO_2$ algorithm: Set the motion threshold for the $WSpO_2$ algorithm.	0x50	0x05	0x06, 4 bytes (a 32-bit signed integer, which is the motion threshold value times 100,000)	-
Algorithm Configuration	$WSpO_2$ algorithm: Set $WSpO_2$ AGC Timeout (sec)	0x50	0x05	0x07, 8-bit unsigned value	-
Algorithm Configuration	$WSpO_2$ algorithm: Set $WSpO_2$ Algorithm Timeout (sec)	0x50	0x05	0x08, 8-bit unsigned value	-
Algorithm Configuration	$WSpO_2$ algorithm: Set $WSpO_2$ PD configuration (source of PPG signal)	0x50	0x05	**0x09, 0x01**: PD1 **0x09, 0x02**: PD2	-
Algorithm Configuration Read	Automatic Gain Control (AGC) algorithm: Read the target percentage of the full-scale ADC range that the AGC algorithm is using.	0x51	0x00	0x00	0 to 100 Percent
Algorithm Configuration Read	AGC algorithm: Read step size toward the target.	0x51	0x00	0x01	0 to 100 Percent
Algorithm Configuration Read	AGC algorithm: Read the sensitivity for the AGC algorithm.	0x51	0x00	0x02	0 to 100 Percent
Algorithm Configuration Read	AGC algorithm: Read the number of samples to average for the AGC algorithm.	0x51	0x00	0x03	Number of samples to average (range is 0 to 255)
Algorithm Configuration Read	Wrist Heart Rate Monitor (WHRM) algorithm: Read the sample rate.	0x51	0x02	0x00	MSB of sample rate, LSB of sample rate (16-bit unsigned integer)
Algorithm Configuration Read	WHRM algorithm: Read the maximum allowed height (cm).	0x51	0x02	0x01	MSB of height, LSB of height (16-bit unsigned integer)
Algorithm Configuration Read	WHRM algorithm: Read the maximum allowed weight (kg).	0x51	0x02	0x02	MSB of weight, LSB of weight (16-bit unsigned integer)

HOST COMMAND					MAX32664
FAMILY NAME	DESCRIPTION	FAMILY BYTE	INDEX BYTE	WRITE BYTES	RESPONSE BYTES
Algorithm Configuration Read	WHRM algorithm: Read the maximum allowed age (years).	0x51	0x02	0x03	Age
Algorithm Configuration Read	WHRM algorithm: Read the minimum allowed height (cm).	0x51	0x02	0x04	MSB of height, LSB of height (16-bit unsigned integer)
Algorithm Configuration Read	WHRM algorithm: Read the minimum allowed weight (kg).	0x51	0x02	0x05	MSB of weight, LSB of weight (16-bit unsigned integer)
Algorithm Configuration Read	WHRM algorithm: Read the minimum allowed age (years).	0x51	0x02	0x06	Age
Algorithm Configuration Read	WHRM algorithm: Read the default height (cm).	0x51	0x02	0x07	MSB of height, LSB of height (16-bit unsigned integer)
Algorithm Configuration Read	WHRM algorithm: Read the default weight (kg).	0x51	0x02	0x08	MSB of weight, LSB of weight (16-bit unsigned integer)
Algorithm Configuration Read	WHRM algorithm: Read the default age (years).	0x51	0x02	0x09	Age
Algorithm Configuration Read	WHRM algorithm: Read the initial heart rate value in bpm which can speed up the algorithm.	0x51	0x02	0x0A	Heart rate (bpm)
Algorithm Configuration Read	MaximFast algorithm: Read the coefficients for the MaximFast SpO_2 algorithm.	0x51	0x02	0x0B	Four bytes signed integer A, Four bytes signed integer B, Four bytes signed integer C (32-bit integers which are the coefficients times 100,000).
Algorithm Configuration Read	WHRM algorithm: Read the AEC algorithm enable status.	0x51	0x02	0x0B	**0x00**: Disabled **0x01**: Enabled
Algorithm Configuration Read	WHRM algorithm: Read the SCD algorithm enable status.	0x51	0x02	0x0C	**0x00**: Disabled **0x01**: Enabled
Algorithm Configuration Read	WHRM algorithm: Read the adjusted target PD current period in seconds.	0x51	0x02	0x0D	MSB, LSB (16-bit unsigned integer)
Algorithm Configuration Read	WHRM algorithm: Read the SCD debounce window.	0x51	0x02	0x0E	MSB, LSB (16-bit unsigned integer)
Algorithm Configuration Read	WHRM algorithm: Read the motion magnitude threshold in 0.1g.	0x51	0x02	0x0F	MSB, LSB (16-bit unsigned integer)
Algorithm Configuration Read	WHRM algorithm: Read the minimum PD current in 0.1mA.	0x51	0x02	0x10	MSB, LSB (16-bit unsigned integer)

HOST COMMAND					MAX32664
FAMILY NAME	DESCRIPTION	FAMILY BYTE	INDEX BYTE	WRITE BYTES	RESPONSE BYTES
Algorithm Configuration Read	WHRM algorithm: Read the PD configuration of the PPG signal source.	0x51	0x02	0x11	**0x01**: PD1 **0x02**: PD2 **0x03:** PD1 and PD2
Algorithm Configuration Read	BPT algorithm: Read the calibration data results from the calibration procedure.	0x51	0x04	0x03	608 bytes of calibration data
Algorithm Configuration Read	$WSpO_2$ algorithm: Read coefficients A, B, C.	0x51	0x05	0x00	Four bytes signed integer A, Four bytes signed integer B, Four bytes signed integer C
Algorithm Configuration Read	$WSpO_2$ algorithm: Read the sample rate.	0x51	0x05	0x01	**0x00**: 100 Hz **0x01**: 25 Hz
Algorithm Configuration Read	$WSpO_2$ algorithm: Read the algorithm run mode.	0x51	0x05	0x02	**0x00**: Continuous **0x01**: One-shot
Algorithm Configuration Read	$WSpO_2$ algorithm: Read the AGC mode status.	0x51	0x05	0x03	**0x00**: Disabled **0x01**: Enabled
Algorithm Configuration Read	$WSpO_2$ algorithm: Read the motion detection status.	0x51	0x05	0x04	**0x00**: Disabled **0x01**: Enabled
Algorithm Configuration Read	$WSpO_2$ algorithm: Read the motion detection period.	0x51	0x05	0x05	MSB of Period, LSB of Period (16-bit unsigned integer)
Algorithm Configuration	$WSpO_2$ algorithm: Read the motion threshold for the $WSpO_2$ algorithm.	0x51	0x05	0x06	4 bytes (32-bit signed integers which are the motion threshold times 100,000)
Algorithm Configuration Read	$WSpO_2$ algorithm: Read the AGC timeout (sec).	0x51	0x05	0x07	AGC timeout (8-bit unsigned)
Algorithm Configuration Read	$WSpO_2$ algorithm: Read Algorithm Timeout (sec)	0x51	0x05	0x08	$WSpO_2$ algorithm timeout (8-bit unsigned)
Algorithm Configuration Read	$WSpO_2$ algorithm: Read the PD configuration (source of PPG signal).	0x51	0x05	0x09	**0x01**: PD1 **0x02**: PD2
Algorithm Mode Enable	AGC: Enable the AGC algorithm.	0x52	0x00	**0x00**: Disable **0x01**: Enable	-
Algorithm Mode Enable	AEC: Enable the AEC algorithm.	0x52	0x01	**0x00**: Disable **0x01**: Enable	-
Algorithm Mode Enable	WHRM, MaximFast: Enable the WHRM, MaximFast algorithm.	0x52	0x02	**0x00**: Disable **0x01**: Enable	-
Algorithm Mode Enable	Electrocardiogram (ECG): Enable the ECG algorithm.	0x52	0x03	**0x00**: Disable **0x01**: Enable	-

HOST COMMAND					MAX32664
FAMILY NAME	DESCRIPTION	FAMILY BYTE	INDEX BYTE	WRITE BYTES	RESPONSE BYTES
Algorithm Mode Enable	Blood Pressure Trending (BPT): Enable the BPT algorithm.	0x52	0x04	**0x00**: Disable **0x01**: Enable	-
Algorithm Mode Enable	$WSpO_2$: Enable the algorithm.	0x52	0x05	**0x00**: Disable **0x01**: Enable	-
Bootloader Flash	Set the initialization vector (IV) bytes. This is not required for a non-secure bootloader.	0x80	0x00	Use bytes 0x28 to 0x32 from the .msbl file as the IV bytes	-
Bootloader Flash	Set the authentication bytes. This is not required for a non-secure bootloader.	0x80	0x01	Use bytes 0x34 to 0x43 from the .msbl file.	-
Bootloader Flash	Set the number of pages.	0x80	0x02	0x00, Number of pages located at byte 0x44 from the .msbl file.	-
Bootloader Flash	Erase the application flash memory.	0x80	0x03		-
Bootloader Flash	Send the page values.	0x80	0x04	The first page is specified by byte 0x4C from the .msbl file. The total bytes for each message protocol are the page size + 16 bytes of CRC.	-
Bootloader Information	Get bootloader version.	0x81	0x00	-	Major version byte, Minor version byte, Revision byte
Bootloader Information	Get the page size in bytes.	0x81	0x01	-	Upper byte of page size, Lower byte of page size
Identity	Read the MCU type.	0xFF	0x00	-	**0x00:** MAX32625 **0x01:** MAX32660/MAX32664
Identity	Read the sensor hub version.	0xFF	0x03	-	Major version byte, Minor version byte, Revision byte
Identity	Read the algorithm: version.	0xFF	0x07	-	Major version byte, Minor version byte, Revision byte

Table 6 defines the bit fields of the sensor hub status byte.

Table 6. Sensor Hub Status Byte

BIT	7	6	5	4	3	2	1	0
Field	Reserved	DevBusy	FifoInOverInt	FifoOutOvrInt	DataRdyInt	Err2	Err1	Err0

Table 7 defines the output FIFO format for the Read Output FIFO I2C message.

Table 7. Output FIFO Format Definitions

SENSOR OR ALGORITHM	SIZE (BYTES)	DESCRIPTION	
MAX86140/MAX86141	3	LED1 value: 24-bit.	
	3	LED2 value: 24-bit.	
	3	LED3 value: 24-bit.	
	3	LED4 value: 24-bit.	
	3	LED5 value: 24-bit.	
	3	LED5 value: 24-bit.	
MAX30205	2	Temperature value: 16 bits, LSB = 0.00390625°C.	
MAX30001	3	ECG: 24-bit.	
	3	R-to-R: 24-bit.	
	3	BIOZ: 24-bit.	
	3	PACE: 24-bit.	
MAX30101	3	LED1 value: 24-bit.	
	3	LED2 value: 24-bit.	
	3	LED3 value: 24-bit.	
	3	LED4 value: 24-bit.	
Accelerometer	2	X: 16-bit two's compliment. LSB = 0.001g	
	2	Y: 16-bit two's compliment. LSB = 0.001g	
	2	Z: 16-bit two's compliment. LSB = 0.001g	
AGC Algorithm	0	No output from algorithm.	
AEC Algorithm	0	No output from algorithm.	
WHRM/MaximFast	2	Heart Rate (bpm): 16-bit, LSB = 0.1 bpm	
	1	Confidence level (0 - 100%): 8-bit, LSB = 1%	
	2	SpO_2 value (0 - 100%): 16-bit, LSB = 0.1%	
	1	WHRM State Machine Status Codes (signed 8-bit): 0: Success +1: Not ready -1: Object detected -2: Excessive sensor device motion -3: No object detected -4: Pressing too hard -5: Object other than finger detected -6: Excessive finger motion	MaximFast State Machine Status Codes: 0: No object detected. 1: Object detected. 2: Object other than finger detected 3: Finger detected.
BPT	1	Calibration/estimation mode status.	
	1	Calibration mode progress. Value ranges from 0 to 100. 100 meaning calibration is complete.	
	2	Heart rate (bpm). LSB = 0.1 bpm.	
	1	Systolic BP value. Only valid once progress = 100.	
	1	Diastolic BP value. Only valid once progress = 100.	

SENSOR OR ALGORITHM	SIZE (BYTES)	DESCRIPTION
WSpO$_2$	2	r: 10x SpO$_2$ r value.
	1	SpO$_2$ confidence in %.
	2	10x the measured value of SpO$_2$.
	1	SpO$_2$ calculation progress (%) - only in one-shot mode.
	1	lowSNRflag: A flag showing low SNR .
	1	Motion flag: A flag showing high motion.
	1	IsWSpo2Calculated: Algorithm reported status.

Table 8 provides the sequence of commands for writing external accelerometer data to the input FIFO.

Table 8. Sequence of Commands to Write External Accelerometer Data to the Input FIFO

HOST COMMAND	COMMAND DESCRIPTION	MAX32664 RESPONSE	RESPONSE DESCRIPTION
0xAA 0x44 0x04 0x1	Enable the input FIFO for host supplied accelerometer data.	0xAB 0x00	No error.
0xAA 0x13 0x00 0x04	Read the sensor sample size for the accelerometer. (optional)	0xAB 0x00 0xN	No error. N is the number of samples in the FIFO.
0xAA 0x14 0x04 Number of Samples, Sample 1 values, ..., Sample N values	Write data to the input FIFO.	0xAB 0x00	No error.
0xAA 0x00 0x00	Read the sensor hub status to verify that the FifoOutOvrInt bit 4 is not set in the Sensor Hub Status byte. (optional)	0xAB 0x00 0x88	No error. Sensor hub is not busy.

MAX32664 I2C Annotated Application Mode Example

Table 9 shows a capture of the I2C traffic between the example host microcontroller (MAX32630FTHR) and the MAX32664GWEA for commanding the MAX32664GWEA to stream raw and algorithm data. The MAXREFDES220# is used for this example.

Table 9. MAX32664GWEA I2C Annotated Application Mode Example

HOST COMMAND	COMMAND DESCRIPTION	MAX32664 RESPONSE	RESPONSE DESCRIPTION
0xAA 0x02 0x00†	Read device mode.	0xAB 0x00 0x00	No error. Mode is application operating mode.
0xAA 0xFF 0x03	Read the sensor hub version.	0xAB 0x00 0x01 0x09 0x00	No error. Version is 1.9.0
0xAA 0x42 0x03†	Get the MAX30101 register attributes.	0xAB 0x00 0x01 0x24	No error. Attributes are 1 byte, 0x24 registers available.
0xAA 0x43 0x03	Read all the MAX30101 registers.	0xAB 0x00 0x00 0x00 0x01 0x00 0x02 0x00 ... 0xFF 0x15	No error. Reg 0x00 = 0, Reg 0x01 = 0, Reg 0x02 = 0, ..., Reg 0xFF = 0x15
0xAA 0x41 0x03 0x07†	Read the MAX30101 register 7.	0xAB 0x00 0x60	No error. Register 7 is 0x60.
0xAA 0x10 0x00 0x03*	Set output mode to raw and algorithm data.	0xAB 0x00	No error.
0xAA 0x10 0x01 0x0F*	Set FIFO threshold as almost full at 0x0F. Increase or decrease this value if you want more or fewer samples per interrupt.	0xAB 0x00	No error.
0xAA 0x44 0x03 0x01*	Enable the MAX30101 sensor.	0xAB 0x00	No error.
0xAA 0x44 0x04 0x01†	Enable the accelerometer. (Only enable if the board has an accelerometer.)	0xAB 0x00	No error.
0xAA 0x52 0x02 0x01*	Enable WHRM/MaximFast 10.x algorithm.	0xAB 0x00	No error.
0xAA 0x00 0x00*	Read the sensor hub status.	0xAB 0x00 0x08	No error. DataRdyInt bit is set
0xAA 0x12 0x00*	Get the number of samples in the FIFO.	0xAB 0x00 0x0F	No error. 0x0F samples are in the FIFO.

HOST COMMAND	COMMAND DESCRIPTION	MAX32664 RESPONSE	RESPONSE DESCRIPTION
0xAA 0x12 0x01*	Read the data stored in the FIFO.	0xAB 0x00 0x03 0x6A 0x43 0x03 0x04 0x92 0x00 0x00 0x00 0x00 0x2E 0x15x 0x02 0x76 0x63 0x03 0xE4 0x03, data for fourteen other samples	No error. IR counts = 223811, Red counts = 19778, LED3 = 0, LED4 = 11797, Heart Rate = 63.0, Confidence = 99, SpO_2 = 99.6, MaximFast State Machine Status = 3, data for fourteen other samples.
0xAA 0x40 0x03 0x0D 0x32	Set the MAX30101 register 0xD (LED2, pulse amplitude) to 0x32.	0xAB 0x00	No error.
0xAA 0x40 0x03 0x0C 0x32	Set the MAX30101 register 0x0C (LED1, pulse amplitude) to 0x32.	0xAB 0x00	No error.
0xAA 0x40 0x03 0x0A 0x23	Set the MAX30101 register 0x0A (ADC Range, Sample Rate, Pulse Width) to 0x23.	0xAB 0x00	No error.
0xAA 0x40 0x03 0x08 0x2F	Set the MAX30101 register 0x08 (Sample Average, FIFO Full) to 0x2F.	0xAB 0x00	No error.
0xAA 0x00 0x00*	Read the Sensor Hub Status.	0xAB 0x00 0x08	No error. DataRdyInt bit is set.
0xAA 0x12 0x00*	Get the number of samples in FIFO.	0xAB 0x00 0x0F	No error. Fifteen samples available.
0xAA 0x12 0x01*	Read the data stored in FIFO.	0xAB 0x00 0x03 0x6A 0x43 0x03 0x04 0x92 0x00 0x00 0x00 0x00 0x2E 0x15 0x02 0x76 0x63 0x03 0xE4 0x03, data for fourteen other samples	No error. IR counts = 223811, Red counts = 19778, LED3 = 0, LED4 = 11797, Heart Rate = 63.0, Confidence = 99, SpO_2 = 99.6, MaximFast State Machine Status = 3, data for fourteen other samples.
0xAA 0x00 0x00*	Read the Sensor Hub Status.	0xAB 0x00 0x08	No error. DataRdyInt bit is set.
0xAA 0x12 0x00*	Get the number of samples in FIFO.	0xAB 0x00 0x0F	No error. Fifteen samples available.
0xAA 0x12 0x01*	Read the data stored in FIFO.	0xAB 0x00 …	No error. Data read.

**Mandatory*
†Recommended

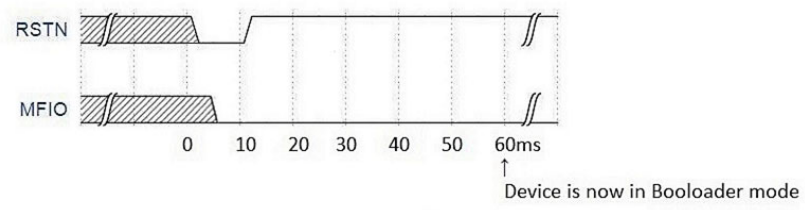
I2C Commands to Flash the Application Algorithm

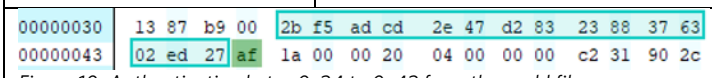
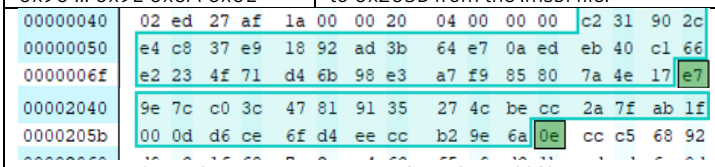
The MAX32664 is pre-programmed with bootloader firmware which accepts in-application programming of the Maxim supplied algorithm file (.msbl). Table 10 is a capture of the I2C commands that are necessary to flash the application algorithm to the MAX32664.

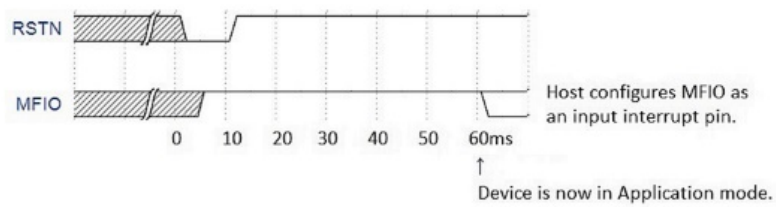
IMPORTANT: *Do not enable the accelerometer if your board does not have the accelerometer.*

This example was captured with the MAX32630FTHR acting as the host microcontroller. The MAX32664 uses the 8-bit slave address of 0xAA. The example encrypted algorithm file used was the MAX32660_SmartSensor_OS24_MaximFast_1.8.2a.msbl (26 pages, 8196 bytes for the page size). Each page sent includes 16 CRC bytes for that page, so there are 8208 bytes per page sent in the payload of the message. The number of pages is located at address 0x44 in the .msbl file.

Table 10. Annotated I2C Trace for Flashing the Application

HOST COMMAND	COMMAND DESCRIPTION	MAX32664 RESPONSE	RESPONSE DESCRIPTION
Sequence the MAX32664 to enter bootloader mode. *  *Figure 7. Sequence to enter bootloader mode.*			
0xAA 0x01 0x00 0x08*	Set mode to 0x08 for bootloader mode.	0xAB 0x00	No error.
0xAA 0x02 0x00	Read mode.	0xAB 0x00 0x08	No error. Mode is bootloader.
0xAA 0xFF 0x00+	Get ID and MCU type.	0xAB 0x00 0x01	No error. MCU is MAX32660/MAX32664.
0xAA 0x81 0x00	Read bootloader firmware version.	0xAB 0x00 0x03 0x00 0x00	No error. Version is 3.0.0.
0xAA 0x81 0x01	Read bootloader page size.	0xAB 0x00 0x20 0x00	No error. Page size is 8192.
0xAA 0x80 0x02 0x00 0x1A*	Bootloader flash. Set the number of pages to 31 based on byte 0x44 from the application .msbl file, which is created from the user application .bin file.	0xAB 0x00	No error.
00000044 | 02 ed 27 af 1a 00 00 20 04 00 00 00 c2 31 90 2c *Figure 8. Page number byte 0x44 from the .msbl file.*			
0xAA 0x80 0x00 0x1A 0xDB 0xE5 0x0D 0x90 0x79 0xE6 0xC6 0x13 0x87 0xB9*	Bootloader flash. Set the initialization vector bytes 0x28 to 0x32 from the .msbl file.	0xAB 0x00	No error.
00000000 6d 73 62 6c 00 00 00 00 4d 41 58 33 32 36 36 30 00000010 00 00 00 00 00 00 00 00 41 45 53 2d 32 35 36 00 00000020 00 00 00 00 00 00 00 00 1a db e5 0d 90 79 e6 c6 00000032 13 87 b9 00 2b f5 ad cd 2e 47 d2 83 23 88 37 63 00000040 02 ed 27 af 1a 00 00 20 04 00 00 00 c2 31 90 2c 00000050 e4 c8 37 e9 18 92 ad 3b 64 e7 0a ed eb 40 c1 66 00000060 e2 23 4f 71 d4 6b 98 e3 a7 f9 85 80 7a 4e 17 e7 *Figure 9. Initialization vector bytes 0x28 to 0x32 from the .msbl file.*			
0xAA 0x80 0x01 0x2B 0xF5 0xAD 0xCD 0x2E 0x47 0xD2 0x83 0x23 0x88 0x37 0x62 0x02 0xED 0x27 0xAF*	Bootloader flash. Set the authentication bytes 0x34 to 0x43 of the .msbl file.	0xAB 0x00	No error.

HOST COMMAND	COMMAND DESCRIPTION	MAX32664 RESPONSE	RESPONSE DESCRIPTION
 *Figure 10. Authentication bytes 0x34 to 0x43 from the .msbl file.*			
0xAA 0x80 0x03*	Bootloader flash. Erase application.	0xAB 0x00	No error.
0xAA 0x80 0x04 0xC2 0x31 0x90 … 0x9E 0x6A 0x0E*	Bootloader flash. Send page bytes 0x4C to 0x205B from the .msbl file.	0xAB 0x00	No error.
 *Figure 11. Send page bytes 0x4C to 0x205B from the .msbl file.*			
0xAA 0x80 0x04 0xCC 0xC5 0x68 … 0xF7 0xD6 0x4C*	Bootloader flash. Send page bytes 0x205C to 0x406B from the .msbl file.	0xAB 0x00	No error.
0xAA 0x80 0x04 0x2E 0xA6 0x13 … 0x84 0xF7 0xCF*	Bootloader flash. Send page bytes 0x406C to 0x607B from the .msbl file.	0xAB 0x00	No error.
0xAA 0x80 0x04 0xD7 0x1F 0x7F … 0x55 0xAB 0xB8*	Bootloader flash. Send page bytes 0x607C to 0x808B from the .msbl file.	0xAB 0x00	No error.
0xAA 0x80 0x04 0xC4 0x63 0x2B … 0x48 0xCD 0x52*	Bootloader flash. Send page bytes 0x808C to 0xA09B from the .msbl file.	0xAB 0x00	No error.
0xAA 0x80 0x04 0x89 0x33 0x22 … 0x31 0xAD 0x19*	Bootloader flash. Send page bytes 0xA09C to 0xC0AB from the .msbl file.	0xAB 0x00	No error.
0xAA 0x80 0x04 0x8B 0x97 0x18 … 0xF3 0xCF 0x90*	Bootloader flash. Send page bytes 0xC0AC to 0xE0BB from the .msbl file.	0xAB 0x00	No error.
0xAA 0x80 0x04 0xD0 0x78 0x38 … 0x1F 0x7F 0x92*	Bootloader flash. Send page bytes 0xE0BC to 0x100CB from the .msbl file.	0xAB 0x00	No error.
0xAA 0x80 0x04 0xB1 0xE9 0x8F … 0xF4 0x23 0xD8*	Bootloader flash. Send page bytes 0x100CC to 0x120DB from the .msbl file.	0xAB 0x00	No error.
0xAA 0x80 0x04 0xF8 0xC6 0x83 … 0xF4 0x24 0xE2*	Bootloader flash. Send page bytes 0x120DC to 0x140EB from the .msbl file.	0xAB 0x00	No error.
0xAA 0x80 0x04 0x1F 0x4F 0x5C … 0xCC 0x2E 0xCD*	Bootloader flash. Send page bytes 0x140EC to 0x160FB from the .msbl file.	0xAB 0x00	No error.
0xAA 0x80 0x04 0x40 0x1F 0x03 … 0x26 0xEB 0xB9*	Bootloader flash. Send page bytes 0x160FC to 0x1810B from the .msbl file.	0xAB 0x00	No error.
0xAA 0x80 0x04 0x2F 0xD9 0xB2 … 0xEE 0x2A 0x8F*	Bootloader flash. Send page bytes 0x1810C to 0x1A11B from the .msbl file.	0xAB 0x00	No error.
0xAA 0x80 0x04 0x51 0x32 0x47 … 0x41 0xE6 0x47*	Bootloader flash. Send page bytes 0x1A11C to 0x1C12B from the .msbl file.	0xAB 0x00	No error.
0xAA 0x80 0x04 0x22 0xA6 0x06 … 0x2A 0xCB 0x44*	Bootloader flash. Send page bytes 0x1C12C to 0x1E13B from the .msbl file.	0xAB 0x00	
0xAA 0x80 0x04 0x68 0x9E 0x1E … 0x53 0x89 0xE8*	Bootloader flash. Send page bytes 0x1E13C to 0x2014B from the .msbl file.	0xAB 0x00	No error.
0xAA 0x80 0x04 0x5F 0x1A 0x6A … 0x14 0xA1 0x85*	Bootloader flash. Send page bytes 0x2014C to 0x2215B from the .msbl file.	0xAB 0x00	No error.
0xAA 0x80 0x04 0xE8 0xDE 0xC9 … 0x81 0xD8 0x00*	Bootloader flash. Send page bytes 0x2215C to 0x2416B from the .msbl file.	0xAB 0x00	No error.
0xAA 0x80 0x04 0x0E 0xD2 0x16 … 0x8D 0x69 0xEE*	Bootloader flash. Send page bytes 0x2416C to 0x2617B from the .msbl file.	0xAB 0x00	No error.
0xAA 0x80 0x04 0x2F 0x4B 0x38 … 0x02 0xA7 0xDC*	Bootloader flash. Send page bytes 0x2617C to 0x2818B from the .msbl file.	0xAB 0x00	No error.
0xAA 0x80 0x04 0xA5 0xFE 0xFD … 0xE3 0x38 0x89*	Bootloader flash. Send page bytes 0x2818C to 0x2A19B from the .msbl file.	0xAB 0x00	No error.
0xAA 0x80 0x04 0x52 0x88 0x9A … 0xF0 0xC5 0x9D*	Bootloader flash. Send page bytes 0x2A19C to 0x2C1AB from the .msbl file.	0xAB 0x00	No error.

HOST COMMAND	COMMAND DESCRIPTION	MAX32664 RESPONSE	RESPONSE DESCRIPTION
0xAA 0x80 0x04 0xA3 0xA6 0x92 … 0xA0 0x4D 0xBE*	Bootloader flash. Send page bytes 0x2C1AC to 0x2E1BB from the .msbl file.	0xAB 0x00	No error.
0xAA 0x80 0x04 0x47 0x09 0x75 … 0x24 0xBD 0x3D*	Bootloader flash. Send page bytes 0x2E1BC to 0x301CB from the .msbl file.	0xAB 0x00	No error.
0xAA 0x80 0x04 0x44 0xEC 0xE6 … 0xBC 0xC9 0x5E*	Bootloader flash. Send page bytes 0x301CC to 0x321DB from the .msbl file.	0xAB 0x00	No error.
0xAA 0x80 0x04 0xD3 0x58 0x34 … 0x62 0x00 0x37*	Bootloader flash. Send page bytes 0x321DC to 0x341EB from the .msbl file.	0xAB 0x00	No error.
Sequence the MAX32664 to enter application mode. *  *Figure 12. Sequence to enter application mode.*			
0xAA 0x02 0x00+	Read mode.	0xAB 0x00 0x00	No errors. Mode is application.
0xAA 0xFF 0x00	Get ID and MCU type.	0xAB 0x00 0x01	No error. MCU is MAX32660/MAX32664
0xAA 0xFF 0x03	Get Sensor Hub version.	0xAB 0x00 0x01 0x08 0x02	No error. Version is 1.8.2.
0xAA 0x42 0x03+	Get the MAX30101 AFE register attributes.	0xAB 0x00 0x01 0x24	No error. Attributes are 1 byte, 0x24 registers available.
0xAA 0x43 0x03	Read all the MAX30101 registers.	0xAB 0x00 0x00 0x00 0x01 0x00 0x02 0x40…	No error. Reg 0x00=0, reg 0x01=0, reg0x02=0x40, … Returns the Read Status Byte and 36 pairs of numbers.
0xAA 0x41 0x03 0x07	Read the MAX30101 register 7.	0xAB 0x00 0x00	No error. Register 0x07 is 0.

**Mandatory*

+Recommended

It is recommended to program the latest version of the MAX32664 sensor hub algorithm .msbl file into the MAX32664 chip. Check the version that is programmed into the chip by using the command "Identity, Read sensor hub version." The latest sensor hub algorithm is available for download for the MAX32664, MAXREFDES220#, and MAXREFDES101# from the Maxim website.

In-Application Programming of the MAX32664

The MAX32664 allows for in-application programming of the firmware.

In-application programming allows for the programming of the sensor hub application firmware during manufacturing and for allowing over-the-air (OTA) updates of the application firmware in the product.

Figure 13 is a flowchart of the in-application programming.

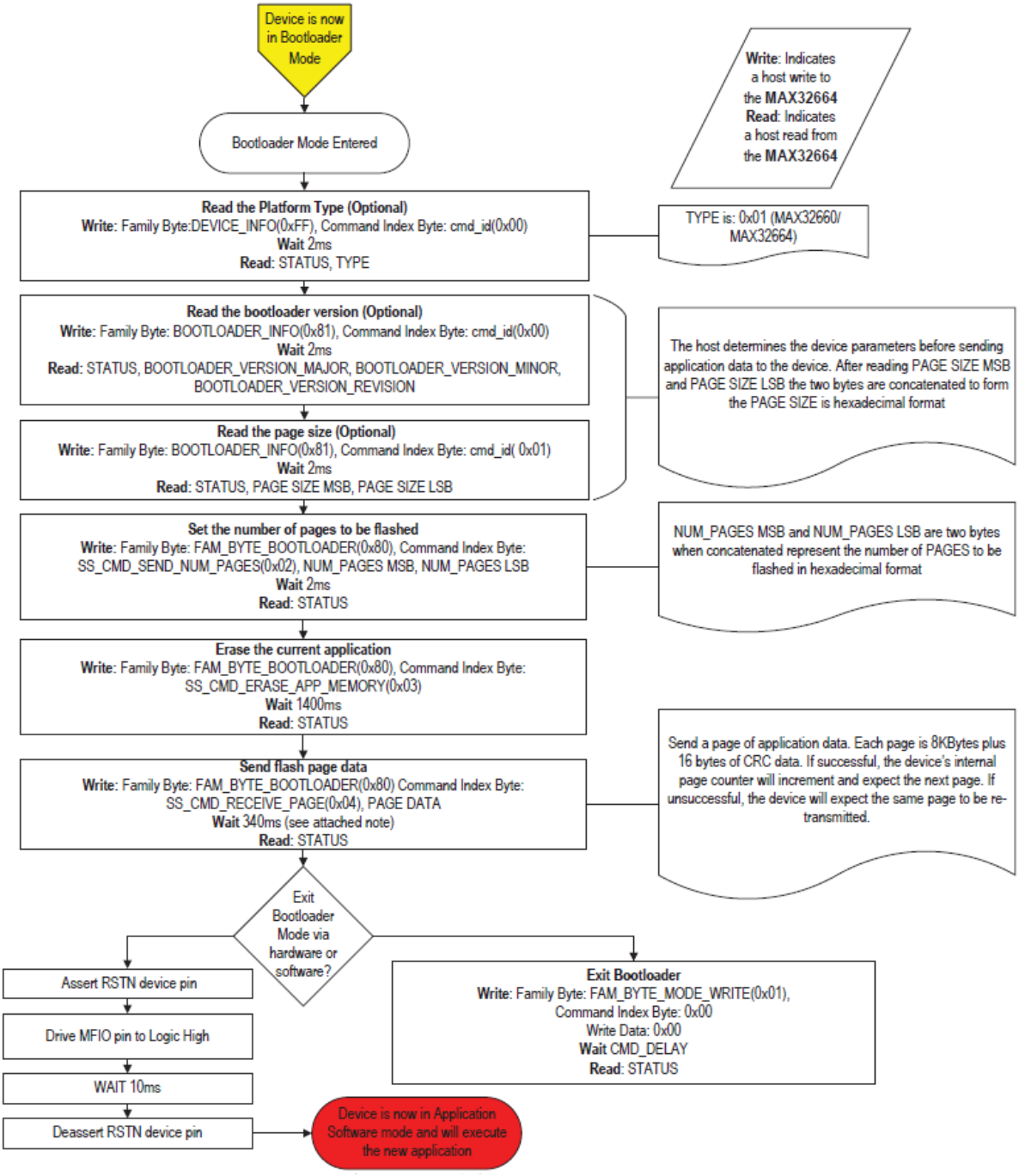


Figure 13. MAX32664 in-application programming flowchart.

MAX32664 APIs and Methods for Reset, Sleep, Status, Heartbeat

Table 11 summarizes the commands and methods to place the MAX32664 into reset or sleep, to interrogate its status, or to generate the "heartbeat" (a periodic signal generated by the software to indicate normal operation).

Table 11. MAX32664 I2C Message Protocol Definitions

COMMAND NAME	HOST COMMAND TO MAX32664	DESCRIPTION
MAX32664 Soft Reset	0xAA 0x01 0x00 0x02	Puts MAX32664 into reset.
MAX30101 AFE Soft Reset by Write Register to AFE	0xAA 0x40 0x03 0x09 0x40	Write 0x40 to MAX30101 register 0x09 to issue a soft reset to the MAX30101. The AFE must be enabled using the enable command.
MAX32664 Sleep	0xAA 0x01 0x00 0x01	To be implemented in the future.
MAX32664 Sleep between Interrupts		To be implemented in the future.
MAX30101 AFE Sleep, Use Write Reg to AFE	0xAA 0x40 0x03 0x09 0x80	Write 0x80 to MAX30101 register 0x09 to put the MAX30101 into shutdown mode. The AFE must be enabled using the enable command.
MAX32664 Hard Reset	Use MFIO and RSTN pins according to Figure 4 and Figure 5.	
WDT in MAX32664 Bootloader Mode		Not implemented.
WDT in MAX32664 .msbl Application mode		Not implemented.
Bootloader or Application Status	0XAA 0x02 0x00	Send the read mode command. Response is 0xAB 0x00 0x08 if in bootloader mode or 0xAB 0x00 0x00 if in application mode.
Heartbeat for Application Mode		Sample source and .msbl file to toggle P0.9 is provided.

Revision History

REVISION NUMBER	REVISION DATE	DESCRIPTION	PAGES CHANGED
0	01/19	Initial release	—