MAX32664 User Guide

UG6806; Rev 1 preliminary; 08/1/19

Abstract

The MAX32664 user guide provides flow charts, timing diagrams, GPIOs/pin usage, I^2C interface protocol, and annotated I^2C 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

Introduction 4

MAX32664 Variants 5

Maxim Reference Designs with MAX32664 9

MAXREFDES220# 9

MAXREFDES101# 10

MAX32664 GPIOs and RSTN Pin 12

MAX32664 Bootup and Application Mode 14

MAX32664 Bootloader Mode 14

MAX32664 Application Mode 14

Communications to the MAX32664 over I2C 15

Bit Transfer Process 15

I2C Write 18

I2C Read 19

MAX32664 I2C Message Protocol Definition 20

MAX32664 I2C Annotated Application Mode Example 49

I2C Commands to Flash the Application Algorithm 54

In-Application Programming of the MAX32664 59

MAX32664 APIs and Methods for Reset, Sleep, Status, Heartbeat 61

Revision History 62

List of Figures

Figure 1. MAX32664 block diagram. 9
Figure 2. MAXREFDES101# block diagram. 10
Figure 3. Pin connections between the host and the MAX32664. 12
Figure 5. Entering bootloader mode using the RSTN pin and the MFIO GPIO pin. 14
Figure 6. Entering application mode using the RSTN pin and MFIO pin. 15
Figure 7. I2C Write/Read data transfer from host microcontroller. 16
Figure 8. Sequence to enter bootloader mode 56
Figure 9. Page number byte 0x44 from the .msbl file. 56
Figure 10. Initialization vector bytes 0x28 to 0x32 from the .msbl file. 56
Figure 11. Authentication bytes 0x34 to 0x43 from the .msbl file. 56
Figure 12. Send page bytes 0x4C to 0x205B from the .msbl file. 57
Figure 13. Sequence to enter application mode 58
Figure 14. MAX32664 in-application programming flowchart. 60

List of Tables

Table 1. MAX32664 Variants, Matching Algorithms, and Reference Designs 5
Table 2. RSTN Pin and GPIOs Pins 12
Table 3. Additional GPIOs Used on the MAX32664 for the MAXREFDES220# 14
Table 4. Additional GPIOs Used on the MAX32664 for the MAXREFDES101# 14
Table 5. Read Status Byte Value 16
Table 6. MAX32664 I2C Message Protocol Definitions 20
Table 7. Sensor Hub Status Byte 45
Table 8. Output FIFO Format Definitions 45
Table 9. Sequence of Commands to Write External Accelerometer Data to the Input FIFO (MAX32664GWEA/B/C) 49
Table 10. MAX32664GWEA I2C Annotated Application Mode Example 49
Table 11. MAX32664GWEB I2C Annotated Application Mode Example 50
Table 12. MAX32664GWEC (WHRM) I2C Annotated Application Mode Example 52
Table 13. MAX32664GWEC (WHRM+WSpO2) I2C Annotated Application Mode Example 53
Table 14. MAX32664GWED I2C Annotated Application Mode Example 54
Table 15. Annotated I2C Trace for Flashing the Application 56
Table 16. MAX32664 I2C Message Protocol Definitions 61

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, I^2C slave interface to a microcontroller host. A second I^2C 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, I^2C, 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 application code which consists of algorithms and the associated sensor driver. 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 ALGORITHM/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 MAX32664GWEA, 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) . 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#
MAX32664GWEB	The **Wearable Heart-rate monitoring** (**WHRM**) algorithm: The WHRM algorithm is configured to use LED1 and Photodiode (PD) 1, and it is compatible with the sensor hub combination of the MAX32664GWEB, MAX86141 AFE, and KX-122 accelerometer. It is mandatory to use accelerometer the KX-122 accelerometer or external accelerometer data with the WHRM algorithm to detect and compensate motion. If the KX-122 accelerometer is not connected to MAX32664, then external accelerometer data should be supplied at the 25Hz sampling rate. The Automatic Exposure Control (AEC) and Skin Control Detection (SCD) of WHRM are included in the WHRM algorithm. If AEC is enabled, the LED current, pulse width and sample rate are automatically determined by the algorithm. If AEC is not enabled, the LED current, LED current range, pulse width, and ADC range registers are set to defaults and may be updated by the host software.	B	MAXREFDES101#

PART NUMBER	APPLICATION ALGORITHM/FIRMWARE	BOOTLOADER KEY	MAXIM REFERENCE DESIGN
MAX32664GWEC	The **Wearable Suite Heart-Rate Monitoring and Wearable Oxygen Saturation** (**WHRM+WSpO2**, v30.xx.x where xx is 3 or greater): The Algorithm Suite can monitor heart rate and SPO2 simultaneously. It is configured to use LED1 (Green), LED2 (IR) and LED3 (Red) and Photodiode 1 and 2, and it is compatible with the sensor hub combination of the MAX32664GWEC, MAX86141 (or MAXM86161) AFE, and KX-122 accelerometer. It is mandatory to use accelerometer the KX-122 accelerometer or external accelerometer data with the WHRM algorithm to detect and compensate motion. If the KX-122 accelerometer is not connected to MAX32664 then the external accelerometer data should be supplied at the 25Hz sampling rate. The Automatic Exposure Control (AEC) and Skin Control Detection (SCD) is included in the Wearable Algorithm Suite. If AEC is enabled, the LED currents, pulse width and sample rate are automatically determined by the algorithm; AEC algorithm adjusts averaging and sample rates for an effective rate of 25 Hz. If AEC is not enabled, they are set to defaults and may be updated by the host software. Low power mode is enabled in the firmware. Normally, when MAX32664 is idle, it switches to “Deep Sleep” state to save power. An external interrupt like sensor, host MFIO or RTC alarm forces MAX32664 to wake up.	C	MAXREFDES101 with SpO2 LED Board

PART NUMBER	APPLICATION ALGORITHM/FIRMWARE	BOOTLOADER KEY	MAXIM REFERENCE DESIGN
	The **Wearable Heart-Rate Monitoring** (**WHRM**) algorithm (25Hz sampling, v30.2.x): The WHRM algorithm is configured to use LED1 (Green) and Photodiode 1 or/and 2, and it is compatible with the sensor hub combination of the MAX32664GWEB, MAX86141 AFE, and KX-122 accelerometer. It is mandatory to use accelerometer the KX-122 accelerometer or external accelerometer data with the WHRM algorithm to detect and compensate motion. If the KX-122 accelerometer is not connected to MAX32664 then the external accelerometer data should be supplied at the 25Hz sampling rate. The Automatic Exposure Control and Skin Control Detection (AEC-SCD) of WHRM (AEC-SCD is included in the WHRM algorithm). If AEC-SCD is enabled, the LED current, pulse width and sample rate are automatically determined by the algorithm; AEC algorithm adjusts averaging and sample rates for an effective rate of 25 Hz. If AEC-SCD is not enabled, the LED current, LED current range, pulse width, and ADC range registers should be configured by the host software. The **Wearable Oxygen Saturation** (**WSpO2**) algorithm (25Hz sampling, v30.2.x): The WSpO2 algorithm is packaged with the WHRM .msbl file, but HRM and SpO2 cannot be run simultaneously for this .msbl. The WSpO2 algorithm is configured to use LED2 (IR) and LED3 (Red) and Phot-diode 1 or 2, and it's compatible with the sensor hub combination of the MAX32664GWEB, MAX86141 (or MAXM86161) AFE, and KX-122 accelerometer. It is optional to use an accelerometer with the WSpO2 algorithm to detect motion. If the KX-122 accelerometer is not connected to MAX32664 external accelerometer data should be supplied, then external accelerometer should use the 25Hz sampling rate. The Automatic Gain Control (AGC) of WSpO2. If AGC is enabled, the LED currents and ADC range are automatically determined by the algorithm. If the AGC is not enabled, the LED currents, pulse width and ADC range registers should be configured by the host software.		

PART NUMBER	APPLICATION ALGORITHM/FIRMWARE	BOOTLOADER KEY	MAXIM REFERENCE DESIGN
MAX32664GWED	The finger-based **Blood Pressure Trending** (**BPT**), heart rate and SpO_2 monitoring algorithm (100Hz Sampling). The algorithm is compatible with the sensor hub combination of the MAX32664GWED and MAX30101 AFE. No accelerometer is required for this algorithm. The BPT algorithm includes Automatic gain control to adjust LED currents. Prior to running the algorithm, a calibration procedure is required to determine blood pressure and SpO_2 calibration coefficients. Automatic gain control (AGC). 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.	D	MAXREFDES220#
MAX32664GWEZ	Z-keyed algorithm/firmware of the above which may be used to in-application program the Maxim Integrated reference designs listed or for boards that use the MAX32664GWEZ.		MAXREFDES220# MAXREFDES101#

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.

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).

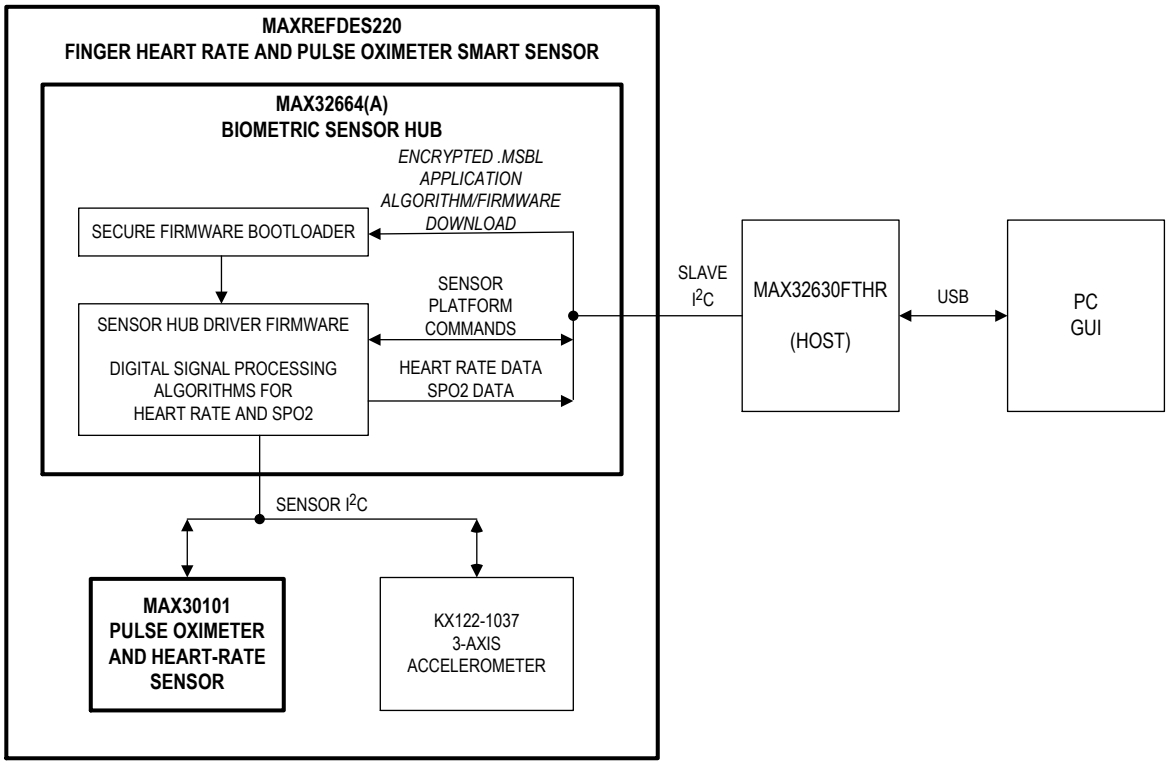


Figure 1. MAX32664 block diagram.

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.

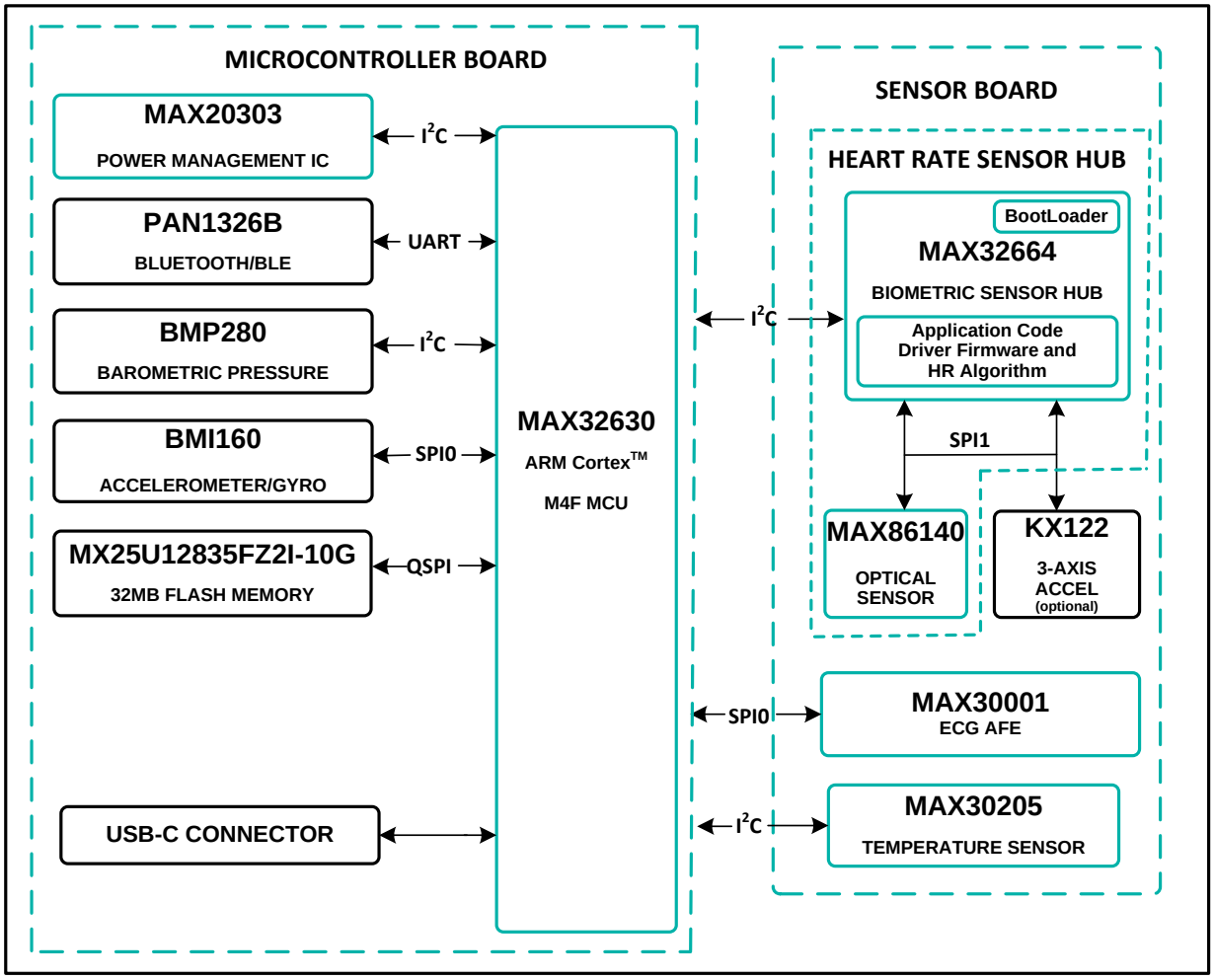


Figure 2. MAXREFDES101# block diagram.

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 application code for heart-rate algorithm and AFE drivers (MAX32664). Algorithm output and sensor data can be streamed through Bluetooth® to an Android® application or PC GUI for demonstration, evaluation, and customized development.

Android is a registered trademark of Google Inc.

The Bluetooth word mark and logos are registered trademarks owned by Bluetooth SIG, Inc. and any use of such marks by Maxim is under license.

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.

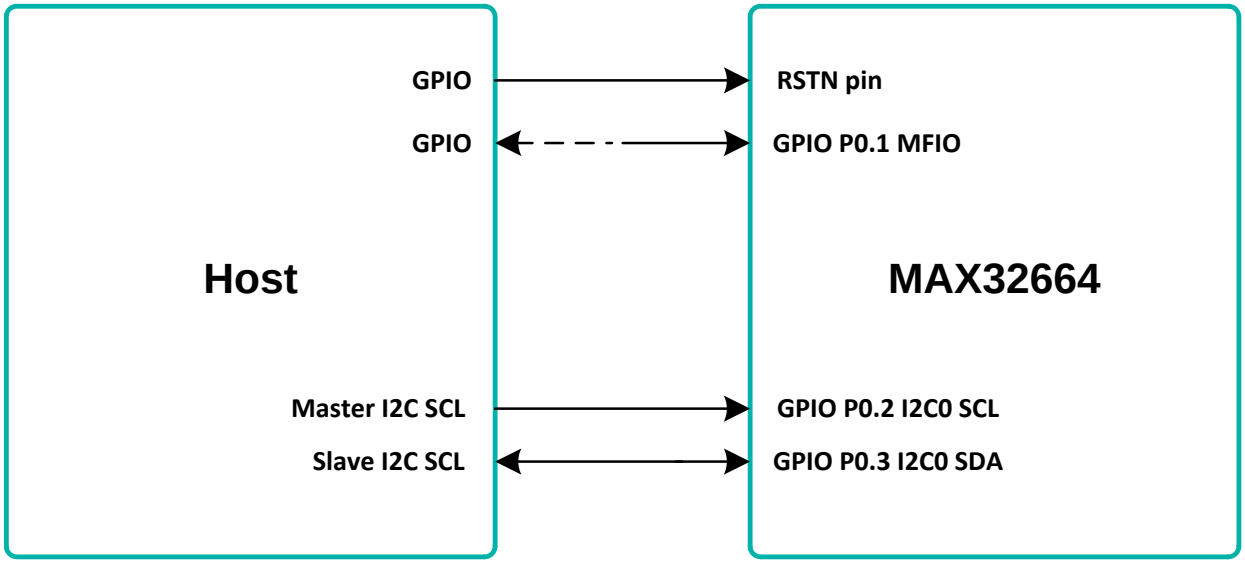


Figure 3. Pin connections between the host and the MAX32664.

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 I^2C 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. Interrupt to host is not used on low- power versions v30.2.4+ or v30.3.x+ MAX32664GWEC.	Input/Output Input only for low- power versions v30.2.4+ or v30.3.x+ MAX32664GWEC.
GPIO P0.2	I2C0_Host SCL	Input
GPIO P0.3	I2C0 Host SDA	Input/Output

The Wearable Suite Heart-Rate Monitoring and Wearable Oxygen Saturation (WHRM+WSpO2) and versions 30.2.4+ of the Wearable Heart-Rate Monitoring (WHRM) .msbl algorithm use a polling method instead of the MFIO being an interrupt to the host. For the WHRM+WSpO2 Suite, the MAX32664GWEC switches to “Deep Sleep” state to save power when in it is in the idle state. It may be woken up by the internal RTC, or the connected sensor, or if the MFIO pin. The host is required to wake up MAX32664GWEC (WHRM+WSpO2 Suite) prior to any I2C communication by:

- Setting MFIO to Low at least 250 usec before the beginning of an I2C transaction to make sure MAX32664GWEC is awake

- Keeping MFIO Low until the end of the I2C transaction to make sure MAX32664GWEC will not switch to “Deep Sleep” state
- Setting MFIO to High at end of I2C communication to allow MAX32664GWEC to switch back to “Deep Sleep” state

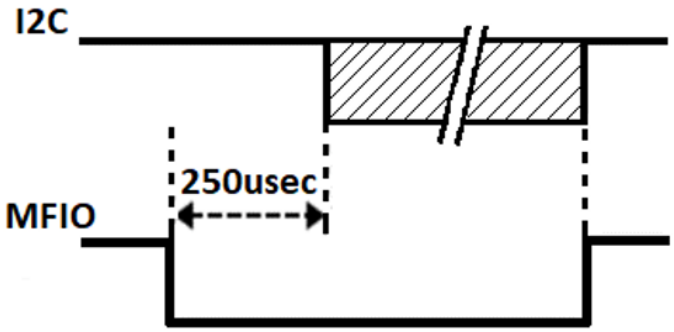


Figure 4. Host sets MFIO low to wake MAX32664GWEC (WHRM+WSpO2 Suite).

For the WHRM+WSpO2 Suite, the host is required to regularly poll MAX32664GWEC to read the measurement data. The polling period depends on the rate that MAX32664GWEC report is generated. By reducing the report period, polling is needed less often and hence the number of wake-up events will be reduced significantly.

Polling period can be set 4-5 times report period to avoid FIFO overflow. In this case, several samples will be read in each polling.

By default, the report rate (read output mode, 0x10 0x02 xx) is set to one per sample, which translates to 40 msec. In this case, a 200 msec polling period is suggested.

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

Table 4. Additional GPIOs Used on the MAX32664 for the MAXREFDES101#

MAX32664	DESCRIPTION	DIRECTION FROM THE MAX32664 SIDE
GPIO P0.0	KX122 ACCEL Select	Output
GPIO P0.4	SPI MISO: MAX86141, KX122	Input
GPIO P0.5	SPI MOSI: MAX86141, KX122	Output
GPIO P0.6	SPI CLK: MAX86141, KX122	Output
GPIO P0.7	MAX86141 Select	Output
GPIO P0.8	MAX86141 Interrupt	Input
GPIO P0.9	KX122 Interrupt (N/A for polling versions 30.2.3+ for MAX32664GWEC)	Input

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 (MFIO pin should be set low at least 1ms before RSTN pin is set high).
- After the 10ms has elapsed, set the RSTN pin high.
- After an additional 50ms has elapsed, the MAX32664 is in bootloader mode.

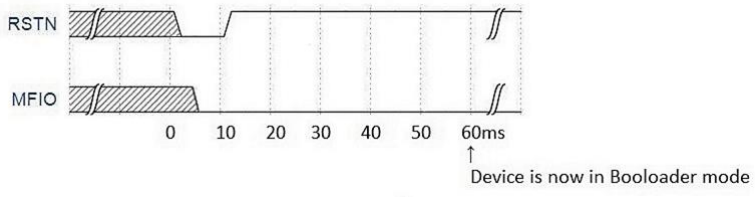


Figure 5. Entering bootloader mode using the RSTN pin and the MFIO GPIO pin.

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. (MFIO pin should be set high at least 1ms before RSTN pin is set high).
- After an additional 50ms has elapsed, the MAX32664 is in application mode and the application performs its initialization of the application software.
- After ~1 second from when the RSTN was set to high, the application completes the initialization and the device is ready to accept I2C commands

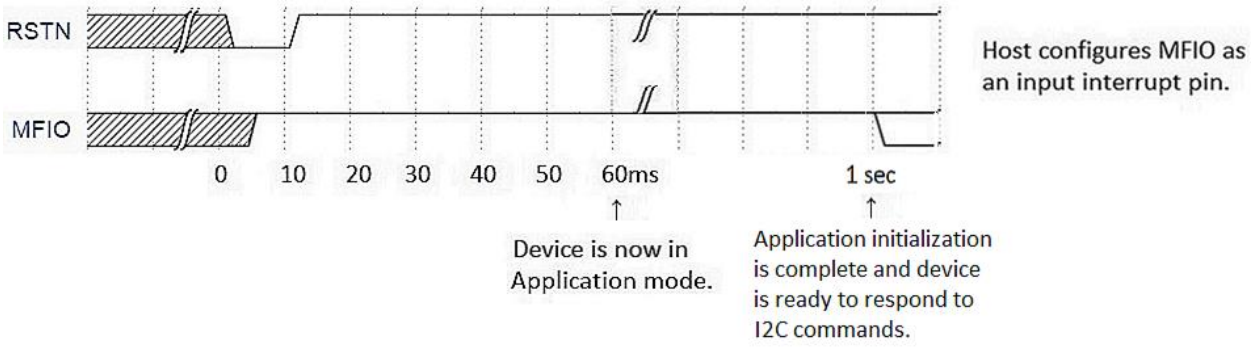


Figure 6. Entering application mode using the RSTN pin and MFIO pin.

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 7.

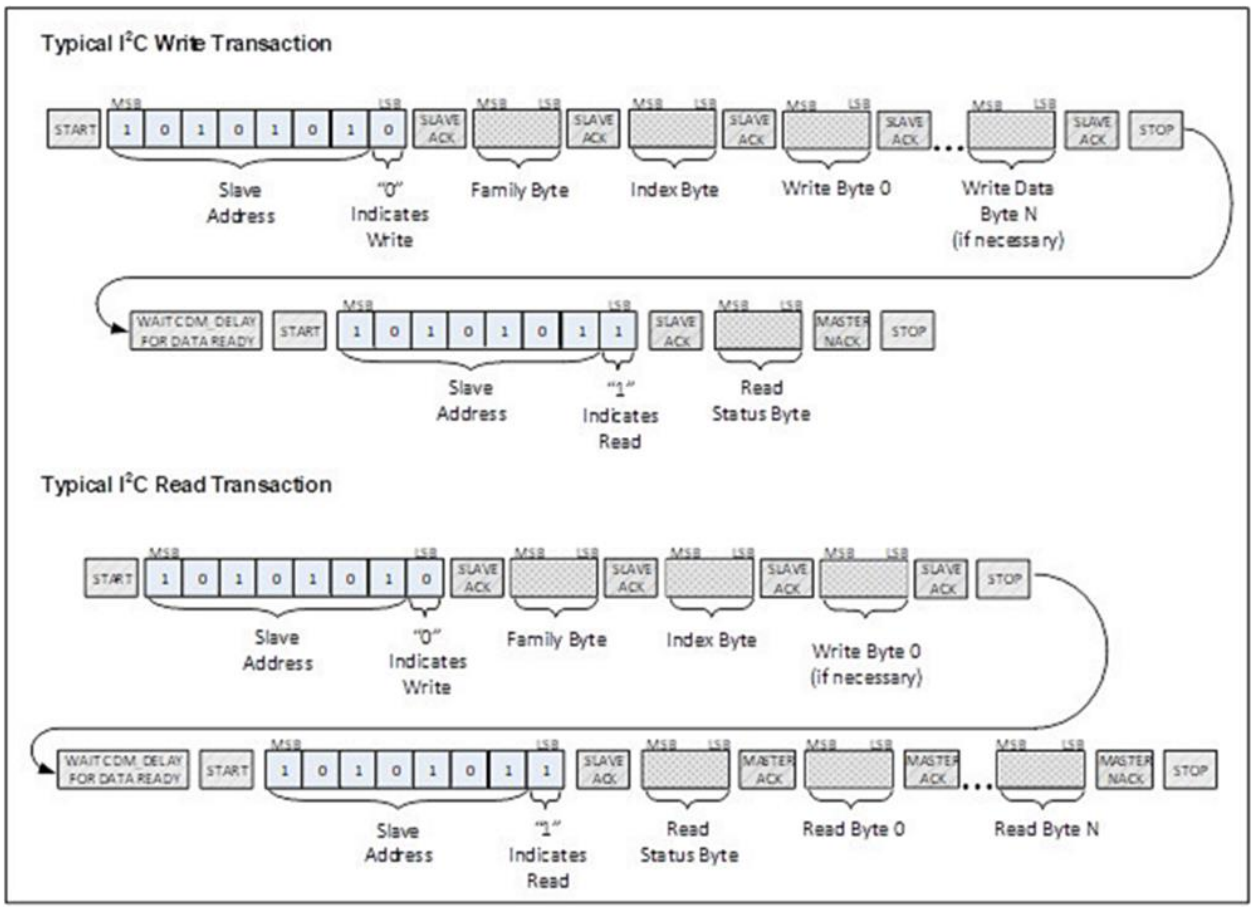


Figure 7. I²C Write/Read data transfer from host microcontroller.

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 5.

Table 5. 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 (2 msec) 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 (2 msec) 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 6 defines the I2C message protocol for the MAX32664.

Table 6. 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 (MAX32664GWEA, MAX32664GWEB, MAX32664GWEC, MAX32664GWED)	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 7 for the for the bit field table.
Device Mode	Select the device operating mode. The application must implement this. (MAX32664GWEA, MAX32664GWEB, MAX32664GWEC, MAX32664GWED)	0x01	0x00	**0x00**: Exit bootloader mode, Enter application mode. **0x02**: Reset. **0x08**: Enter bootloader mode.	-
Device Mode	Read the device operating mode. (MAX32664GWEA, MAX32664GWEB, MAX32664GWEC, MAX32664GWED)	0x02	0x00	-	**0x00**: Application operating mode. **0x08**: Bootloader operating mode.

HOST COMMAND					MAX32664
FAMILY NAME	**DESCRIPTION**	**FAMILY BYTE**	**INDEX BYTE**	**WRITE BYTES**	**RESPONSE BYTES**
Set Output Mode	Set the output format of the sensor hub. (MAX32664GWEA, MAX32664GWEB, MAX32664GWEC, MAX32664GWED)	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	-
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. (MAX32664GWEA, MAX32664GWEB, MAX32664GWEC, MAX32664GWED)	0x10	0x01	**0x01 to 0xFF:** Sensor Hub Interrupt Threshold for FIFO.	-
Set Output Mode	Set the period of reporting, in terms of report counts. E.g., 25 means report once every 25 samples. (MAX32664GWEC)	0x10	0x02	**0x01 to 0xFF = N:** Report once every N samples	-
Read Output Mode	READ the output format of the sensor hub. (MAX32664GWEA, MAX32664GWEB, MAX32664GWEC, MAX32664GWED)	0x11	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**
Read Output Mode	Read 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. (MAX32664GWEA, MAX32664GWEB, MAX32664GWEC, MAX32664GWED)	0x11	0x01		**0x01 to 0xFF:** Sensor Hub Interrupt Threshold for FIFO.
Set Output Mode	Read the period of reporting, in terms of report counts. E.g., 25 means report once every 25 samples. The default of 1 is report once per sample or every 40 msec. (MAX32664GWEC,	0x11	0x02		**0x01 (default) to 0xFF = N:** Report once every N samples
Read Output FIFO	Get the number of samples available in the FIFO. (MAX32664GWEA, MAX32664GWEB, MAX32664GWEC, MAX32664GWED)	0x12	0x00	-	Number of samples available in the FIFO.
Read Output FIFO	Read data stored in output FIFO. (MAX32664GWEA, MAX32664GWEB, MAX32664GWEC, MAX32664GWED)	0x12	0x01	-	See Table 8, 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. (MAX32664GWEA, MAX32664GWEB, MAX32664GWEC)	0x13	0x00	**0x04**: Accelerometer	**0x06:** Bytes per sample for the external accelerometer. Three 16-bit 2's complement with LSB = 0.001g. See Table 9 for an example.

HOST COMMAND					MAX32664
FAMILY NAME	**DESCRIPTION**	**FAMILY BYTE**	**INDEX BYTE**	**WRITE BYTES**	**RESPONSE BYTES**
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). (MAX32664GWEA, MAX32664GWEB, MAX32664GWEC, MAX32664GWED)	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). (MAX32664GWEA, MAX32664GWEB, MAX32664GWEC)	0x13	0x02	**0x04**: Accelerometer	MSB, LSB
Read Input FIFO for External Sensors	Read the number of samples currently in the input FIFO (16-bit). (MAX32664GWEA, MAX32664GWEB, MAX32664GWEC)	0x13	0x03	**0x04**: Accelerometer	MSB, LSB
Read Input FIFO for External Sensors	Read the number of samples currently in the sensor FIFO (16-bit). (MAX32664GWEA, MAX32664GWEB, MAX32664GWEC, MAX32664GWED)	0x13	0x04	-	MSB, LSB
Write Input FIFO for External Sensors	Write data to the input FIFO. (MAX32664GWEA, MAX32664GWEB, MAX32664GWEC, MAX32664GWED)	0x14	0x00	Sample 1 values, …, Sample N values See Table 9 for an example.	-
Write Register	Write a value to a writable MAX86140/ MAX86141/MAX86141 register. (MAX32664GWEB, MAX32664GWEC)	0x40	0x00	Register address, Register value	-
Write Register	Write a value to a writable MAX30205 register. (MAX32664GWEB)	0x40	0x01	Register address, Register value	-
Write Register	Write a value to a writable MAX30001 register. (MAX32664GWEB)	0x40	0x02	Register address, Register value	-

HOST COMMAND					MAX32664
FAMILY NAME	**DESCRIPTION**	**FAMILY BYTE**	**INDEX BYTE**	**WRITE BYTES**	**RESPONSE BYTES**
Write Register	Write a value to a writable MAX30101 register. (MAX32664GWEA, MAX32664GWED)	0x40	0x03	Register address, Register value	-
Write Register	Write a value to a writable accelerometer sensor register. (MAX32664GWEA, MAX32664GWEB, MAX32664GWEC)	0x40	0x04	Register address, Register value	-
Read Register	Read the value of a MAX86140/ MAX86141/MAX86141 register. (MAX32664GWEB, MAX32664GWEC)	0x41	0x00	Register Address	Register value
Read Register	Read the value of a MAX30205 register. (MAX32664GWEB)	0x41	0x01	Register Address	Register value
Read Register	Read the value of a MAX30001 register. (MAX32664GWEB)	0x41	0x02	Register Address	Register value
Read Register	Read the value of a MAX30101 register. (MAX32664GWEA, MAX32664GWED)	0x41	0x03	Register Address	Register value
Read Register	Read the value of an accelerometer sensor register. (MAX32664GWEA, MAX32664GWEB, MAX32664GWEC, MAX32664GWED)	0x41	0x04	Register Address	Register value
Get Attributes of the AFE	Retrieve the attributes of the MAX86140/ MAX86141/MAX86141 AFE. (MAX32664GWEB, MAX32664GWEC)	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. (MAX32664GWEB)	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. (MAX32664GWEB)	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. (MAX32664GWEA, MAX32664GWED)	0x42	0x03	-	Number of bytes in a word for this sensor, Number of registers available for this sensor.

HOST COMMAND					MAX32664
FAMILY NAME	**DESCRIPTION**	**FAMILY BYTE**	**INDEX BYTE**	**WRITE BYTES**	**RESPONSE BYTES**
Get Attributes of the AFE	Retrieve the attributes of the accelerometer sensor AFE. (MAX32664GWEA, MAX32664GWEB, (MAX32664GWEC, MAX32664GWED)	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. (MAX32664GWEB, MAX32664GWEC)	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. (MAX32664GWEB)	0x43	0x01	-	Register address 0, register value 0, register address 1, register value 1, …, register address n, register value n
Dump Registers	Read all the MAX30001 registers. (MAX32664GWEB)	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. (MAX32664GWEA, MAX32664GWED)	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. (MAX32664GWEA, MAX32664GWEB, MAX32664GWEC, MAX32664GWED)	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/MAXM86 141 sensor. CMD_DELAY=20 msec (MAX32664GWEB, MAX32664GWEC)	0x44	0x00	**0x00**: Disable **0x01**: Enable	-
Sensor Mode Enable	Enable the MAX30205 sensor. CMD_DELAY=20 msec (MAX32664GWEB)	0x44	0x01	**0x00**: Disable **0x01**: Enable	-
Sensor Mode Enable	Enable the MAX30001 sensor. CMD_DELAY=20 msec (MAX32664GWEB)	0x44	0x02	**0x00**: Disable **0x01**: Enable	-
Sensor Mode Enable	Enable the MAX30101 sensor. CMD_DELAY=40 msec (MAX32664GWEA, (MAX32664GWED)	0x44	0x03	**0x00**: Disable **0x01**: Enable	-

HOST COMMAND					MAX32664
FAMILY NAME	**DESCRIPTION**	**FAMILY BYTE**	**INDEX BYTE**	**WRITE BYTES**	**RESPONSE BYTES**
Sensor Mode Enable	Enable the accelerometer sensor. CMD_DELAY=20 msec (MAX32664GWEA, MAX32664GWEB, MAX32664GWEC)	0x44	0x04	**0x00**: Disable, **0x00:** Sensor Hub accelerometer **0x00**: Disable, **0x01:** External Host accelerometer **0x01**: Enable, **0x00:** Sensor Hub accelerometer **0x01**: Enable, **0x01:** External Host accelerometer	-
Sensor Mode Read	Read the MAX86140/ MAX86141/MAX86141 sensor mode. (MAX32664GWEB, MAX32664GWEC)	0x45	0x00		**0x00**: Disabled **0x01**: Enabled
Sensor Mode Read	Read the MAX30205 sensor mode. (MAX32664GWEB)	0x45	0x01		**0x00**: Disabled **0x01**: Enabled
Sensor Mode Read	Read the MAX30001 sensor mode. (MAX32664GWEB)	0x45	0x02		**0x00**: Disabled **0x01**: Enabled
Sensor Mode Read	Read the MAX30101 sensor mode. (MAX32664GWEA, (MAX32664GWED)	0x45	0x03		**0x00**: Disabled **0x01**: Enabled
Sensor Mode Read	Read the external accelerometer sensor mode. (MAX32664GWEA, MAX32664GWEB, MAX32664GWEC)	0x45	0x04		**0x00**: Disabled, **0x00:** Sensor Hub accelerometer **0x00**: Disabled, **0x01:** External Host accelerometer **0x01**: Enabled, **0x00:** Sensor Hub accelerometer **0x01**: Enabled, **0x01:** External Host accelerometer

HOST COMMAND					MAX32664
FAMILY NAME	**DESCRIPTION**	**FAMILY BYTE**	**INDEX BYTE**	**WRITE BYTES**	**RESPONSE BYTES**
Sensor Configuration	Write the sensor configuration for MAX86140/ MAX86141/MAXM86161 LSb0 of Write Byte is firmware_default bit LSb1 of Write Byte is dac_calib bit. CMD_DELAY=220 msec if first Write Byte is 0x02 or 0x03. (MAX32664GWEC with MAXM86161)	0x46	0x00	**First Byte** **0x00:** Do not use firmware default and do not run Dac calib. Do not use firmware default register settings, and do not run DAC calibration, when algo/sensor is enabled. Sensor hub doesn't overwrite user settings when algo/sensor is enabled. If user doesn't disable AEC, then Sample rate, Pulse Interval and LED current will be managed by algorithm. AEC disable is the separate command. Ignores ppg_cfg1 value. **0x01:** Use firmware default but do not run Dac calib. Use firmware default register settings and disable DAC calibration. As soon as algorithm is run, it uses firmware defaults, but do not run DAC calibration. Ignores ppg_cfg1 value. **0x02:** Do not use firmware default but run Dac calib. This run DAC Calib immediately. Immediately runs DAC calibration using ppg_cfg1 value and waits for user settings to directly write user register values to MAX8614x. This mode doesn't use firmware default. Algorithm no longer run calibration again, since it was run when command is received. Only this mode immediately runs calibration. Uses ppg_cfg1 value. **0x03:** Default. Use firmware default and run Dac calib. Use firmware default register settings and run DAC calibration when algo/sensor is enabled. Ignores ppg_cfg1 value and uses firmware default ppg_cfg1. **Second Byte:** ppg_cfg value **Third Byte:** DAC CAL register value	-

HOST COMMAND					MAX32664
FAMILY NAME	**DESCRIPTION**	**FAMILY BYTE**	**INDEX BYTE**	**WRITE BYTES**	**RESPONSE BYTES**
Sensor Configuration	Read the sensor configuration for MAX86140/ MAX86141/MAXM86161 LSb0 of Write Byte is firmware default bit LSb1 of Write Byte is dac_calib bit. (MAX32664GWEC with MAXM86161)	0x47	0x00	-	**First Byte** **0x00:** Do not use firmware default and do not run Dac calib. Do not use firmware default register settings, and do not run DAC calibration, when algo/sensor is enabled. Sensor hub doesn't overwrite user settings when algo/sensor is enabled. If user doesn't disable AEC, then Sample rate, Pulse Interval and LED current will be managed by algorithm. AEC disable is the separate command. Ignores ppg_cfg1 value. **0x01:** Use firmware default but do not run Dac calib. Use firmware default register settings and disable DAC calibration. As soon as algorithm is run, it uses firmware defaults, but do not run DAC calibration. Ignores ppg_cfg1 value. **0x02:** Do not use firmware default but run Dac calib. This run DAC Calib immediately. Immediately runs DAC calibration using ppg_cfg1 value, and waits for user settings to directly write user register values to MAX8614x. This mode doesn't use firmware default. Algorithm no longer run calibration again, since it was run when command is received. Only this mode immediately runs calibration. Uses ppg_cfg1 value. **0x03:** Default. Use firmware default and run Dac calib. Use firmware default register settings and run DAC calibration when algo/sensor is enabled. Ignores ppg_cfg1 value and uses firmware default ppg_cfg1. **Second Byte:** ppg_cfg value **Third Byte:** DAC CAL register value
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. (MAX32664GWEA, MAX32664GWED)	0x50	0x00	0x00, 0 to 100 percent	-

HOST COMMAND					MAX32664
FAMILY NAME	**DESCRIPTION**	**FAMILY BYTE**	**INDEX BYTE**	**WRITE BYTES**	**RESPONSE BYTES**
Algorithm Configuration	AGC algorithm: Set the step size toward the target for the AGC algorithm. (MAX32664GWEA, MAX32664GWED)	0x50	0x00	0x01, 0 to 100 percent	-
Algorithm Configuration	AGC algorithm: Set the sensitivity for the AGC algorithm. (MAX32664GWEA, MAX32664GWED)	0x50	0x00	0x02, 0 to 100 percent	-
Algorithm Configuration	AGC algorithm: Set the number of samples to average for the AGC algorithm. (MAX32664GWEA)	0x50	0x00	0x03, Number of samples to average (range is 0 to 255).	-
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 Photodiode 1 (25Hz sample rate) and is compatible with the MAX86141 AFE, KX-122 accelerometer. (MAX32664GWEB)	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). (MAX32664GWEB)	0x50	0x02	0x01, MSB of height, LSB of height (16-bit unsigned integer)	-
Algorithm Configuration	WHRM algorithm: Set the maximum allowed weight (kg). (MAX32664GWEB)	0x50	0x02	0x02, MSB of weight, LSB of weight (16-bit unsigned integer)	-
Algorithm Configuration	WHRM algorithm: Set the maximum allowed age (years). (MAX32664GWEB)	0x50	0x02	0x03, Age	-
Algorithm Configuration	WHRM algorithm: Set the minimum allowed height (cm). (MAX32664GWEB)	0x50	0x02	0x04, MSB of height, LSB of height (16-bit unsigned integer)	-
Algorithm Configuration	WHRM algorithm: Set the minimum allowed weight (kg). (MAX32664GWEB)	0x50	0x02	0x05, MSB of weight, LSB of weight (16-bit unsigned integer)	-
Algorithm Configuration	WHRM algorithm: Set the minimum allowed age (years). (MAX32664GWEB)	0x50	0x02	0x06, Age	-

HOST COMMAND					MAX32664
FAMILY NAME	**DESCRIPTION**	**FAMILY BYTE**	**INDEX BYTE**	**WRITE BYTES**	**RESPONSE BYTES**
Algorithm Configuration	WHRM algorithm: Set the default height (cm). (MAX32664GWEB)	0x50	0x02	0x07, MSB of height, LSB of height (16-bit unsigned integer)	-
Algorithm Configuration	WHRM algorithm: Set the default weight (kg). (MAX32664GWEB)	0x50	0x02	0x08, MSB of weight, LSB of weight (16-bit unsigned integer)	-
Algorithm Configuration	WHRM algorithm: Set the default age (years). (MAX32664GWEB)	0x50	0x02	0x09, Age	-
Algorithm Configuration	WHRM algorithm: Set the initial heart-rate value in bpm which might speed up the algorithm. (MAX32664GWEB)	0x50	0x02	0x0A, Heart rate (bpm)	-
Algorithm Configuration	MaximFast algorithm: Set the MaximFast SpO_2 coefficients. (MAX32664GWEA)	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. (MAX32664GWEB)	0x50	0x02	**0x0B, 0x00**: Disable **0x0B, 0x01**: Enable	-
Algorithm Configuration	WHRM algorithm: Enable skin contact detection (SCD) algorithm. (MAX32664GWEB)	0x50	0x02	**0x0C, 0x00**: Disable **0x0C, 0x01**: Enable	-
Algorithm Configuration	WHRM algorithm: Set adjusted target photo detector (PD) current period in seconds. (MAX32664GWEB)	0x50	0x02	0x0D, MSB, LSB (unsigned 16-bit integer)	-
Algorithm Configuration	WHRM algorithm: Set SCD debounce window. (MAX32664GWEB)	0x50	0x02	0x0E, MSB, LSB (unsigned 16-bit integer)	-
Algorithm Configuration	WHRM algorithm: Set motion magnitude threshold. (MAX32664GWEB)	0x50	0x02	0x0F, MSB, LSB (unsigned 16-bit integer, 0.1g)	-

HOST COMMAND					MAX32664
FAMILY NAME	**DESCRIPTION**	**FAMILY BYTE**	**INDEX BYTE**	**WRITE BYTES**	**RESPONSE BYTES**
Algorithm Configuration	WHRM algorithm: Set the minimum PD current in 0.1mA. (MAX32664GWEB)	0x50	0x02	0x10, MSB, LSB (unsigned 16-bit integer)	-
Algorithm Configuration	WHRM algorithm: Configure the source of the PPG signal for the PD. (MAX32664GWEB)	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. (MAX32664GWED)	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. (MAX32664GWED)	0x50	0x04	0x01, diastolic value 1, diastolic value 2, diastolic value 3	-
Algorithm Configuration	BPT algorithm: Write the three samples of the systolic BP byte values needed by the calibration procedure. (MAX32664GWED)	0x50	0x04	0x02, systolic value 1, systolic value 2, systolic value 3	-
Algorithm Configuration	BPT algorithm: Write the calibration data for this user. (Use the data from the 0x51 0x04 0x03 command). CMD_DELAY=30 msec (MAX32664GWED)	0x50	0x04	0x03, 824 bytes of calibration data	-
Algorithm Configuration	BPT algorithm: Configure whether the user is not resting or resting. (MAX32664GWED)	0x50	0x04	**0x05, 0x00**: Resting **0x05, 0x01**: Not resting	-

HOST COMMAND					MAX32664
FAMILY NAME	**DESCRIPTION**	**FAMILY BYTE**	**INDEX BYTE**	**WRITE BYTES**	**RESPONSE BYTES**
Algorithm Configuration	BPT algorithm: Set the SpO_2 coefficients A, B, C. (MAX32664GWED)	0x50	0x04	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	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 (MAX32664GWEC)	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. (MAX32664GWEC)	0x50	0x05	**0x01, 0x00**: 100Hz **0x01, 0x01**: 25Hz	-
Algorithm Configuration	$WSpO_2$ algorithm: Set the algorithm run mode. (MAX32664GWEC)	0x50	0x05	**0x02, 0x00:** Continuous **0x02, 0x01:** One-shot **0x02, 0x02:** Data for SpO2 Calibration	-
Algorithm Configuration	$WSpO_2$ algorithm: Set the AGC mode for the $WSpO_2$ algorithm. (MAX32664GWEC)	0x50	0x05	**0x03, 0x00:** Disable **0x03, 0x01:** Enable	-
Algorithm Configuration	$WSpO_2$ algorithm: Set motion detection. (MAX32664GWEC)	0x50	0x05	**0x04, 0x00:** Disable **0x04, 0x01:** Enable	-
Algorithm Configuration	$WSpO_2$ algorithm: Set the motion detection period, seconds. (MAX32664GWEC)	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 in 0.00001 g. (MAX32664GWEC)	0x50	0x05	0x06, 4 bytes (a 32-bit signed integer, which is the motion threshold value times 100,000)	-

HOST COMMAND					MAX32664
FAMILY NAME	**DESCRIPTION**	**FAMILY BYTE**	**INDEX BYTE**	**WRITE BYTES**	**RESPONSE BYTES**
Algorithm Configuration	$WSpO_2$ algorithm: Set $WSpO_2$ AGC timeout (sec). (MAX32664GWEC)	0x50	0x05	0x07, 8-bit unsigned value	-
Algorithm Configuration	$WSpO_2$ algorithm: Set $WSpO_2$ algorithm timeout (sec). (MAX32664GWEC)	0x50	0x05	0x08, 8-bit unsigned value	-
Algorithm Configuration	$WSpO_2$ algorithm: Set $WSpO_2$ PD configuration (source of PPG signal). (MAX32664GWEC)	0x50	0x05	**0x09, 0x01**: PD1 **0x09, 0x02**: PD2 **0x09, 0x02**: PD1, PD2	-
Algorithm Configuration	Wearable Algo Suite (WHRM+WSPO2) algorithm: Set the SP02 coefficients A, B, C. (MAX32664GWEC)	0x50	0x07	0x00, four bytes signed integer A, four bytes signed integer B, four bytes signed integer C The MAXREFDES101 with SpO2 LED/PD board without the cover glass uses the following coefficients as the default values: A= -1666666 B= 833333 C= 10000000	-
Algorithm Configuration	Wearable Algo Suite (WHRM+WSPO2) algorithm: Set the motion detection period, seconds. (MAX32664GWEC)	0x50	0x07	0x01, MSB of period, LSB of period (16-bit unsigned integer)	-
Algorithm Configuration	Wearable Algo Suite (WHRM+WSPO2) algorithm: Set the motion threshold for the $WSpO_2$ algorithm. (MAX32664GWEC)	0x50	0x07	0x02, four bytes (32-bit signed integers which are the motion threshold times 100,000)	-
Algorithm Configuration	Wearable Algo Suite (WHRM+WSPO2) algorithm: Set the WSPO2 AGC timeout (sec) (MAX32664GWEC)	0x50	0x07	0x03, WSPO2 AGC timeout (8-bit unsigned)	-
Algorithm Configuration	Wearable Algo Suite (WHRM+WSPO2) algorithm: Set the SPO2 algorithm timeout (sec) (MAX32664GWEC)	0x50	0x07	0x04, SPO2 algorithm timeout (8-bit unsigned)	-

HOST COMMAND					MAX32664
FAMILY NAME	**DESCRIPTION**	**FAMILY BYTE**	**INDEX BYTE**	**WRITE BYTES**	**RESPONSE BYTES**
Algorithm Configuration	Wearable Algo Suite (WHRM+WSPO2) algorithm: Set the initial heart rate setting (MAX32664GWEC)	0x50	0x07	0x05, Initial heart rate setting (8-bit unsigned)	-
Algorithm Configuration	Wearable Algo Suite (WHRM+WSPO2) algorithm: Set the user's height (MAX32664GWEC)	0x50	0x07	0x06, Height (16-bit unsigned integer which is the height in cm times 256)	-
Algorithm Configuration	Wearable Algo Suite (WHRM+WSPO2) algorithm: Set the user's weight (MAX32664GWEC)	0x50	0x07	0x07, Weight (16-bit unsigned integer which is the weight in kg times 256)	-
Algorithm Configuration	Wearable Algo Suite (WHRM+WSPO2) algorithm: Set the user's age (MAX32664GWEC)	0x50	0x07	0x08, Age in years (8-bit unsigned)	-
Algorithm Configuration	Wearable Algo Suite (WHRM+WSPO2) algorithm: Set the user's gender (MAX32664GWEC)	0x50	0x07	0x09, **0x00:** male **0x01:** female	-
Algorithm Configuration	Wearable Algo Suite (WHRM+WSPO2) algorithm: Set the algorithm run mode (MAX32664GWEC)	0x50	0x07	0x0A, **0x00:** Continuous HRM + continuous SPO2 **0x01:** Continuous HRM + One-shot SPO2 **0x02:** Continuous HRM **0x03:** Sampled HRM **0x04:** Sampled HRM + One-shot SPO2 **0x05**: Activity Tracking only **0x06:** SPO2 calibration	-
Algorithm Configuration	Wearable Algo Suite (WHRM+WSPO2) algorithm: Set the AEC algorithm enable (MAX32664GWEC)	0x50	0x07	0x0B, **0x00:** Disabled **0x01:** Enabled	-
Algorithm Configuration	Wearable Algo Suite (WHRM+WSPO2) algorithm: Set the SCD algorithm enable (MAX32664GWEC)	0x50	0x07	0x0C, **0x00:** Disabled **0x01:** Enabled	-

HOST COMMAND					MAX32664
FAMILY NAME	**DESCRIPTION**	**FAMILY BYTE**	**INDEX BYTE**	**WRITE BYTES**	**RESPONSE BYTES**
Algorithm Configuration	Wearable Algo Suite (WHRM+WSPO2) algorithm: Set the target PD current period (period to update the target PD current with the AEC formula) (MAX32664GWEC)	0x50	0x07	0x0D, Target PD current period (16-bit unsigned integer)	-
Algorithm Configuration	Wearable Algo Suite (WHRM+WSPO2) algorithm: Set the motion magnitude threshold (MAX32664GWEC)	0x50	0x07	0x0E, Motion magnitude threshold (16-bit unsigned integer, 0.001g)	-
Algorithm Configuration	Wearable Algo Suite (WHRM+WSPO2) algorithm: Set the minimum PD current setting (MAX32664GWEC)	0x50	0x07	0x0F, Minimum PD current setting (16-bit unsigned integer, 0.1 uA)	-
Algorithm Configuration	Wearable Algo Suite (WHRM+WSPO2) algorithm: Set the initial PD current setting (MAX32664GWEC)	0x50	0x07	0x10, Initial PD current setting (16-bit unsigned integer, , 0.1 uA)	-
Algorithm Configuration	Wearable Algo Suite (WHRM+WSPO2) algorithm: Set the target PD current setting (MAX32664GWEC)	0x50	0x07	0x11, Target PD current setting (16-bit unsigned integer, 0.1 uA)	-
Algorithm Configuration	Wearable Algo Suite (WHRM+WSPO2) algorithm: Set the auto target PD current calculation enable (MAX32664GWEC)	0x50	0x07	0x12, **0x00:** Value of target PD current is used (AGC functionality) **0x01:** Target PD current is calculated automatically	-
Algorithm Configuration	Wearable Algo Suite (WHRM+WSPO2) algorithm: Set the minimum integration time setting (MAX32664GWEC)	0x50	0x07	0x13, **0x00:** 14.8 usec (default) **0x01:** 29.4 usec **0x02:** 58.7 usec **0x03:** 117.3 usec	-
Algorithm Configuration	Wearable Algo Suite (WHRM+WSPO2) algorithm: Set the minimum frequency sampling setting (MAX32664GWEC)	0x50	0x07	0x14, **0x00:** 25 SPS, averaging=1 (default) **0x01:** 50 SPS, averaging=2 **0x02:** 100 SPS, averaging=4 **0x03:** 200 SPS, averaging=8 **0x04:** 400 SPS, averaging=16	-

HOST COMMAND					MAX32664
FAMILY NAME	**DESCRIPTION**	**FAMILY BYTE**	**INDEX BYTE**	**WRITE BYTES**	**RESPONSE BYTES**
Algorithm Configuration	Wearable Algo Suite (WHRM+WSPO2) algorithm: Set the maximum integration time setting (MAX32664GWEC)	0x50	0x07	0x15, **0x00:** 14.8 usec (default) **0x01:** 29.4 usec **0x02:** 58.7 usec **0x03:** 117.3 usec	-
Algorithm Configuration	Wearable Algo Suite (WHRM+WSPO2) algorithm: Set the maximum frequency sampling setting (MAX32664GWEC)	0x50	0x07	0x16, **0x00:** 25 SPS, averaging=1 (default) **0x01:** 50 SPS, averaging=2 **0x02:** 100 SPS, averaging=4 **0x03:** 200 SPS, averaging=8 **0x04:** 400 SPS, averaging=16	-
Algorithm Configuration	Wearable Algo Suite (WHRM+WSPO2) algorithm: Set the PD configuration for WHRM (MAX32664GWEC)	0x50	0x07	0x17, **0x01:** Photodiode 1 (default) **0x02:** Photodiode 2 **0x03:** Photodiode 1 & 2	-
Algorithm Configuration	Wearable Algo Suite (WHRM+WSPO2) algorithm: Set the PD configuration for WSPO2 (MAX32664GWEC)	0x50	0x07	0x18, **0x01:** Photodiode 1 (default) **0x02:** Photodiode 2 **0x03:** Photodiode 1 & 2	-
Algorithm Configuration Read	Automatic Gain Control (AGC) algorithm: Read the target percentage of the full-scale ADC range that the AGC algorithm is using. (MAX32664GWEA, MAX32664GWED))	0x51	0x00	0x00	0 to 100 Percent
Algorithm Configuration Read	AGC algorithm: Read step size toward the target. (MAX32664GWEA, MAX32664GWED)	0x51	0x00	0x01	0 to 100 Percent
Algorithm Configuration Read	AGC algorithm: Read the sensitivity for the AGC algorithm. (MAX32664GWEA, MAX32664GWED)	0x51	0x00	0x02	0 to 100 Percent
Algorithm Configuration Read	AGC algorithm: Read the number of samples to average for the AGC algorithm. (MAX32664GWEA, MAX32664GWED)	0x51	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 Read	Wrist Heart Rate Monitor (WHRM) algorithm: Read the sample rate. (MAX32664GWEB)	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). (MAX32664GWEB)	0x51	0x02	0x01	MSB of height, LSB of height (16-bit unsigned integer)
Algorithm Configuration Read	WHRM algorithm: Read the maximum allowed weight (kg). (MAX32664GWEB)	0x51	0x02	0x02	MSB of weight, LSB of weight (16-bit unsigned integer)
Algorithm Configuration Read	WHRM algorithm: Read the maximum allowed age (years). (MAX32664GWEB)	0x51	0x02	0x03	Age
Algorithm Configuration Read	WHRM algorithm: Read the minimum allowed height (cm). (MAX32664GWEB)	0x51	0x02	0x04	MSB of height, LSB of height (16-bit unsigned integer)
Algorithm Configuration Read	WHRM algorithm: Read the minimum allowed weight (kg). (MAX32664GWEB)	0x51	0x02	0x05	MSB of weight, LSB of weight (16-bit unsigned integer)
Algorithm Configuration Read	WHRM algorithm: Read the minimum allowed age (years). (MAX32664GWEB)	0x51	0x02	0x06	Age
Algorithm Configuration Read	WHRM algorithm: Read the default height (cm). (MAX32664GWEB)	0x51	0x02	0x07	MSB of height, LSB of height (16-bit unsigned integer)
Algorithm Configuration Read	WHRM algorithm: Read the default weight (kg). (MAX32664GWEB)	0x51	0x02	0x08	MSB of weight, LSB of weight (16-bit unsigned integer)
Algorithm Configuration Read	WHRM algorithm: Read the default age (years). (MAX32664GWEB)	0x51	0x02	0x09	Age
Algorithm Configuration Read	WHRM algorithm: Read the initial heart rate value in bpm which can speed up the algorithm. (MAX32664GWEB)	0x51	0x02	0x0A	Heart rate (bpm)
Algorithm Configuration Read	MaximFast algorithm: Read the coefficients for the MaximFast SpO_2 algorithm. (MAX32664GWEA)	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).

HOST COMMAND					MAX32664
FAMILY NAME	**DESCRIPTION**	**FAMILY BYTE**	**INDEX BYTE**	**WRITE BYTES**	**RESPONSE BYTES**
Algorithm Configuration Read	WHRM algorithm: Read the AEC algorithm enable status. (MAX32664GWEB)	0x51	0x02	0x0B	**0x00**: Disabled **0x01**: Enabled
Algorithm Configuration Read	WHRM algorithm: Read the SCD algorithm enable status. (MAX32664GWEB)	0x51	0x02	0x0C	**0x00**: Disabled **0x01**: Enabled
Algorithm Configuration Read	WHRM algorithm: Read the adjusted target PD current period in seconds. (MAX32664GWEB)	0x51	0x02	0x0D	MSB, LSB (16-bit unsigned integer)
Algorithm Configuration Read	WHRM algorithm: Read the SCD debounce window. (MAX32664GWEB)	0x51	0x02	0x0E	MSB, LSB (16-bit unsigned integer)
Algorithm Configuration Read	WHRM algorithm: Read the motion magnitude threshold. (MAX32664GWEB)	0x51	0x02	0x0F	MSB, LSB (16-bit unsigned integer, 0.1g)
Algorithm Configuration Read	WHRM algorithm: Read the minimum PD current in 0.1mA. (MAX32664GWEB)	0x51	0x02	0x10	MSB, LSB (16-bit unsigned integer)
Algorithm Configuration Read	WHRM algorithm: Read the PD configuration of the PPG signal source. (MAX32664GWEB)	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. Host may use this for saving the user calibration data when switching users or for writing user calibration data after a reset. (MAX32664GWED)	0x51	0x04	0x03	824 bytes of calibration data
Algorithm Configuration Read	$WSpO_2$ algorithm: Read the SpO2 coefficients A, B, C. (MAX32664GWEC)	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. (MAX32664GWEC)	0x51	0x05	0x01	**0x00**: 100 Hz **0x01**: 25 Hz
Algorithm Configuration Read	$WSpO_2$ algorithm: Read the algorithm run mode. (MAX32664GWEC)	0x51	0x05	0x02	**0x00**: Continuous **0x01**: One-shot **0x02**: Data for SpO2 Calibration

HOST COMMAND					MAX32664
FAMILY NAME	**DESCRIPTION**	**FAMILY BYTE**	**INDEX BYTE**	**WRITE BYTES**	**RESPONSE BYTES**
Algorithm Configuration Read	$WSpO_2$ algorithm: Read the AGC mode status. (MAX32664GWEC)	0x51	0x05	0x03	**0x00**: Disabled **0x01**: Enabled
Algorithm Configuration Read	$WSpO_2$ algorithm: Read the motion detection status. (MAX32664GWEC)	0x51	0x05	0x04	**0x00**: Disabled **0x01**: Enabled
Algorithm Configuration Read	$WSpO_2$ algorithm: Read the motion detection period, seconds. (MAX32664GWEC)	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. (MAX32664GWEC)	0x51	0x05	0x06	four 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 the algorithm timeout (sec) (MAX32664GWEC)	0x51	0x05	0x08	$WSpO_2$ algorithm timeout (8-bit unsigned)
Algorithm Configuration Read	$WSpO_2$ algorithm: Read the PD configuration (source of PPG signal). (MAX32664GWEC)	0x51	0x05	0x09	**0x01**: PD1 **0x02**: PD2 **0x03**: PD1, PD2
Algorithm Configuration Read	Wearable Aglo Suite (WHRM+WSPO2) algorithm: Read the SPO2 coefficients A, B, C. (MAX32664GWEC)	0x51	0x07	0x00	Four bytes signed integer A, four bytes signed integer B, four bytes signed integer C
Algorithm Configuration Read	Wearable Aglo Suite (WHRM+WSPO2) algorithm: Read the motion detection period, seconds. (MAX32664GWEC)	0x51	0x07	0x01	MSB of period, LSB of period (16-bit unsigned integer)
Algorithm Configuration	Wearable Aglo Suite (WHRM+WSPO2) algorithm: Read the motion threshold for the $WSpO_2$ algorithm. (MAX32664GWEC)	0x51	0x07	0x02	Four bytes (32-bit signed integers which are the motion threshold times 100,000)
Algorithm Configuration Read	Wearable Algo Suite (WHRM+WSPO2) algorithm: Read the WSPO2 AGC timeout (sec) (MAX32664GWEC)	0x51	0x07	0x03	WSPO2 AGC timeout (8-bit unsigned)

HOST COMMAND					MAX32664
FAMILY NAME	**DESCRIPTION**	**FAMILY BYTE**	**INDEX BYTE**	**WRITE BYTES**	**RESPONSE BYTES**
Algorithm Configuration Read	Wearable Algo Suite (WHRM+WSPO2) algorithm: Read the SPO2 algorithm timeout (sec) (MAX32664GWEC)	0x51	0x07	0x04	SPO2 algorithm timeout (8-bit unsigned)
Algorithm Configuration Read	Wearable Algo Suite (WHRM+WSPO2) algorithm: Read the initial heart rate setting (MAX32664GWEC)	0x51	0x07	0x05	Initial heart rate setting (8-bit unsigned)
Algorithm Configuration Read	Wearable Algo Suite (WHRM+WSPO2) algorithm: Read the user's height (MAX32664GWEC)	0x51	0x07	0x06	Height (16-bit unsigned integer which is the height in cm times 256)
Algorithm Configuration Read	Wearable Algo Suite (WHRM+WSPO2) algorithm: Read the user's weight (MAX32664GWEC)	0x51	0x07	0x07	Weight (16-bit unsigned integer which is the weight in kg times 256)
Algorithm Configuration Read	Wearable Algo Suite (WHRM+WSPO2) algorithm: Read the user's age (MAX32664GWEC)	0x51	0x07	0x08	Age in years (8-bit unsigned)
Algorithm Configuration Read	Wearable Algo Suite (WHRM+WSPO2) algorithm: Read the user's gender (MAX32664GWEC)	0x51	0x07	0x09	**0x00:** male **0x01:** female
Algorithm Configuration Read	Wearable Algo Suite (WHRM+WSPO2) algorithm: Read the algorithm run mode (MAX32664GWEC)	0x51	0x07	0x0A	**0x00:** Continuous HRM + continuous SPO2 **0x01:** Continuous HRM + One-shot SPO2 **0x02:** Continuous HRM **0x03:** Sampled HRM **0x04:** Sampled HRM + One-shot SPO2 **0x05**: Activity Tracking only **0x06:** SPO2 calibration
Algorithm Configuration Read	Wearable Algo Suite (WHRM+WSPO2) algorithm: Read the AEC algorithm enable (MAX32664GWEC)	0x51	0x07	0x0B	**0x00:** Disabled **0x01:** Enabled
Algorithm Configuration Read	Wearable Algo Suite (WHRM+WSPO2) algorithm: Read the SCD algorithm enable (MAX32664GWEC)	0x51	0x07	0x0C	**0x00:** Disabled **0x01:** Enabled

HOST COMMAND					MAX32664
FAMILY NAME	**DESCRIPTION**	**FAMILY BYTE**	**INDEX BYTE**	**WRITE BYTES**	**RESPONSE BYTES**
Algorithm Configuration Read	Wearable Algo Suite (WHRM+WSPO2) algorithm: Read the target PD current period (period to update the target PD current with the AEC formula) (MAX32664GWEC)	0x51	0x07	0x0D	Target PD current period (16-bit unsigned integer)
Algorithm Configuration Read	Wearable Algo Suite (WHRM+WSPO2) algorithm: Read the motion magnitude threshold (MAX32664GWEC)	0x51	0x07	0x0E	Motion magnitude threshold (16-bit unsigned integer, 0.001g)
Algorithm Configuration Read	Wearable Algo Suite (WHRM+WSPO2) algorithm: Read the minimum PD current setting (MAX32664GWEC)	0x51	0x07	0x0F	Minimum PD current setting (16-bit unsigned integer, 0.1 uA)
Algorithm Configuration Read	Wearable Algo Suite (WHRM+WSPO2) algorithm: Read the initial PD current setting (MAX32664GWEC)	0x51	0x07	0x10	Initial PD current setting (16-bit unsigned integer, 0.1 uA)
Algorithm Configuration Read	Wearable Algo Suite (WHRM+WSPO2) algorithm: Read the target PD current setting (MAX32664GWEC)	0x51	0x07	0x11	Target PD current setting (16-bit unsigned integer, 0.1 uA)
Algorithm Configuration Read	Wearable Algo Suite (WHRM+WSPO2) algorithm: Read the auto target PD current calculation enable (MAX32664GWEC)	0x51	0x07	0x12	**0x00:** Value of target PD current is used (AGC functionality) **0x01:** Target PD current is calculated automatically
Algorithm Configuration Read	Wearable Algo Suite (WHRM+WSPO2) algorithm: Read the minimum integration time setting (MAX32664GWEC)	0x51	0x07	0x13	**0x00:** 14.8 usec (default) **0x01:** 29.4 usec **0x02:** 58.7 usec **0x03:** 117.3 usec
Algorithm Configuration Read	Wearable Algo Suite (WHRM+WSPO2) algorithm: Read minimum frequency sampling setting (MAX32664GWEC)	0x51	0x07	0x14	**0x00:** 25 SPS, averaging=1 (default) **0x01:** 50 SPS, averaging=2 **0x02:** 100 SPS, averaging=4 **0x03:** 200 SPS, averaging=8 **0x04:** 400 SPS, averaging=16

HOST COMMAND					MAX32664
FAMILY NAME	**DESCRIPTION**	**FAMILY BYTE**	**INDEX BYTE**	**WRITE BYTES**	**RESPONSE BYTES**
Algorithm Configuration Read	Wearable Algo Suite (WHRM+WSPO2) algorithm: Read the maximum integration time setting (MAX32664GWEC)	0x51	0x07	0x15	**0x00:** 14.8 usec (default) **0x01:** 29.4 usec **0x02:** 58.7 usec **0x03:** 117.3 usec
Algorithm Configuration Read	Wearable Algo Suite (WHRM+WSPO2) algorithm: Read the maximum frequency sampling setting (MAX32664GWEC)	0x51	0x07	0x16	**0x00:** 25 SPS, averaging=1 (default) **0x01:** 50 SPS, averaging=2 **0x02:** 100 SPS, averaging=4 **0x03:** 200 SPS, averaging=8 **0x04:** 400 SPS, averaging=16
Algorithm Configuration Read	Wearable Algo Suite (WHRM+WSPO2) algorithm: Set the PD configuration for WHRM (MAX32664GWEC)	0x50	0x07	0x17, **0x01:** Photodiode 1 (default) **0x02:** Photodiode 2 **0x03:** Photodiode 1 & 2	
Algorithm Configuration Read	Wearable Algo Suite (WHRM+WSPO2) algorithm: Set the PD configuration for WSPO2 (MAX32664GWEC)	0x50	0x07	0x18, **0x01:** Photodiode 1 (default) **0x02:** Photodiode 2 **0x03:** Photodiode 1 & 2	
Algorithm Configuration Read	Wearable Algo Suite (WHRM+WSPO2) algorithm: Read WSPO2 algorithm timeout (sec) (MAX32664GWEC)	0x51	0x07	0x03	WSPO2 algorithm timeout (8-bit unsigned)
Algorithm Mode Enable	AGC: Enable the AGC algorithm. CMD_DELAY=20 msec (MAX32664GWEA)	0x52	0x00	**0x00**: Disable **0x01**: Enable	-
Algorithm Mode Enable	WHRM, MaximFast: Enable the WHRM, MaximFast algorithm. CMD_DELAY=40 msec (MAX32664GWEA, MAX32664GWEB)	0x52	0x02	**0x00**: Disable **0x01**: Enable Mode 1 **0x02**: Enable Mode 2 (See Table 8 for Algorithm Mode definitions)	-
Algorithm Mode Enable	Blood Pressure Trending (BPT): Enable the BPT algorithm. CMD_DELAY=20 msec (MAX32664GWED)	0x52	0x04	**0x00**: Disable **0x01**: Enable Calibration Mode **0x02**: Enable Estimation Mode	-

HOST COMMAND					MAX32664
FAMILY NAME	**DESCRIPTION**	**FAMILY BYTE**	**INDEX BYTE**	**WRITE BYTES**	**RESPONSE BYTES**
Algorithm Mode Enable	$WSpO_2$: Enable the algorithm. CMD_DELAY=20 msec (MAX32664GWEC)	0x52	0x05	**0x00**: Disable **0x01**: Enable Mode 1	-
Algorithm Mode Enable	Wearable Algo Suite (WHRM+WSPO2) algorithm: Enable the algorithm. CMD_DELAY=120 msec CMD_DELAY=320 msec for enable with DAC (MAX32664GWEC)	0x52	0x07	**0x00**: Disable **0x01**: Enable Mode 1 **0x02**: Enable Mode 1 with DAC	
Bootloader Flash	Set the initialization vector (IV) bytes. (MAX32664GWEA, MAX32664GWEB, MAX32664GWEC, MAX32664GWED)	0x80	0x00	Use bytes 0x28 to 0x32 from the .msbl file as the IV bytes	-
Bootloader Flash	Set the authentication bytes. (MAX32664GWEA, MAX32664GWEB, MAX32664GWEC, MAX32664GWED)	0x80	0x01	Use bytes 0x34 to 0x43 from the .msbl file.	-
Bootloader Flash	Set the number of pages. (MAX32664GWEA, MAX32664GWEB, MAX32664GWEC, MAX32664GWED)	0x80	0x02	0x00, Number of pages located at byte 0x44 from the .msbl file.	-
Bootloader Flash	Erase the application flash memory. (MAX32664GWEA, MAX32664GWEB, MAX32664GWEC, MAX32664GWED)	0x80	0x03		-
Bootloader Flash	Send the page values. (MAX32664GWEA, MAX32664GWEB, MAX32664GWEC, MAX32664GWED)	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. (MAX32664GWEA, MAX32664GWEB, MAX32664GWEC, MAX32664GWED)	0x81	0x00	-	Major version byte, Minor version byte, Revision byte

HOST COMMAND					MAX32664
FAMILY NAME	**DESCRIPTION**	**FAMILY BYTE**	**INDEX BYTE**	**WRITE BYTES**	**RESPONSE BYTES**
Bootloader Information	Get the page size in bytes. (MAX32664GWEA, MAX32664GWEB, MAX32664GWEC, MAX32664GWED)	0x81	0x01	-	Upper byte of page size, Lower byte of page size
Identity	Read the MCU type. (MAX32664GWEA, MAX32664GWEB, MAX32664GWEC, MAX32664GWED)	0xFF	0x00	-	**0x00:** MAX32625 **0x01:** MAX32660/MAX32664
Identity	Read the sensor hub version. (MAX32664GWEA, MAX32664GWEB, MAX32664GWEC, MAX32664GWED)	0xFF	0x03	-	Major version byte, Minor version byte, Revision byte
Identity	Read the algorithm: version. (MAX32664GWEA, MAX32664GWEB, MAX32664GWEC, MAX32664GWED)	0xFF	0x07	-	Major version byte, Minor version byte, Revision byte

Table 7 defines the bit fields of the sensor hub status byte.

Table 7. Sensor Hub Status Byte

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

Table 8, defines output FIFO format for the Read Output FIFO I²C message for MAX32664GWEA/B/C/D. Slot1Data, Slot2Data, … defines the data fields that are placed in the output FIFO; N/A means that the value is not applicable.

Table 8. Output FIFO Format Definitions

SENSOR OR ALGORITHM	SIZE (BYTES)	DESCRIPTION
MAX30101 (MAX32664GWEA, MAX32664GWED)	3	LED1 value: 24-bit. IR LED count.
	3	LED2 value: 24-bit. Red LED count.
	3	LED3 value: 24-bit (N/A).
	3	LED4 value: 24-bit (N/A).
MAX86141 WHRM (MAX32664GWEB, MAX32664GWEC)	3	Slot1Data value: 24-bit. Green LED, PD1 count.
	3	Slot2Data value: 24-bit (N/A).
	3	Slot3Data value: 24-bit (N/A).
	3	Slot4Data value: 24-bit. Green LED, PD2 count.
	3	Slot5Data value: 24-bit (N/A).
	3	Slot6Data value: 24-bit (N/A).
MAXM86161 WHRM (MAX32664GWEB, MAX32664GWEC)	3	Slot1Data value: 24-bit. Green LED, PD1 count.
	3	Slot2Data value: 24-bit (N/A).
	3	Slot3Data value: 24-bit (N/A).
	3	Slot4Data value: 24-bit (N/A).
	3	Slot5Data value: 24-bit (N/A).
	3	Slot6Data value: 24-bit (N/A).
MAX86141 $WSpO_2$ (MAX32664GWEC)	3	Slot1Data value: 24-bit (N/A).
	3	Slot2Data value: 24-bit (N/A).
	3	Slot3Data value: 24-bit (N/A).
	3	Slot4Data value: 24-bit. Red LED count.
	3	Slot5Data value: 24-bit. IR LED count.
	3	Slot6Data value: 24-bit (N/A).
MAXM86161 $WSpO_2$ (MAX32664GWEC)	3	Slot1Data value: 24-bit (N/A).
	3	Slot2Data value: 24-bit Red LED count.
	3	Slot3Data value: 24-bit IR LED count.
	3	Slot4Data value: 24-bit (N/A).
	3	Slot5Data value: 24-bit (N/A).
	3	Slot6Data value: 24-bit (N/A).
MAX86141 Heart-Rate Monitoring and Wearable Oxygen Saturation (WHRM+WSPO2) (MAX32664GWEC)	3	Slot1Data value: 24-bit Green LED, PD1 count.
	3	Slot2Data value: 24-bit (N/A).
	3	Slot3Data value: 24-bit (N/A).
	3	Slot4Data value: 24-bit. Green LED, PD2 count.
	3	Slot5Data value: 24-bit. IR LED count.
	3	Slot6Data value: 24-bit. Red LED count.
MAX30205 (MAX32664GWEB)	2	Temperature value: 16 bits, LSB = 0.00390625°C.
MAX30001 (MAX32664GWEB)	3	ECG: 24-bit.
	3	R-to-R: 24-bit.
	3	BIOZ: 24-bit.

SENSOR OR ALGORITHM	SIZE (BYTES)	DESCRIPTION
	3	PACE: 24-bit.
KX-122 Accelerometer (MAX32664GWEA, MAX32664GWEB, MAX32664GWEC)	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
MaximFast Algorithm, Mode 1 (MAX32664GWEA)	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	MaximFast State Machine Status Codes: 0: No object detected. 1: Object detected. 2: Object other than finger detected 3: Finger detected.
MaximFast Algorithm Mode 2 (Sensor Hub v10.1.0 or above) (MAX32664GWEA)	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	MaximFast State Machine Status Codes: 0: No object detected. 1: Object detected. 2: Object other than finger detected 3: Finger detected.
	2	r: 16-bit LSB = 0.1 SpO2 r value.
	1	MaximFast State Machine Extended Status Codes: 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
	1	0 (reserved for future use)
	1	0 (reserved for future use)
AGC Algorithm (MAX32664GWEA, MAX32664GWED)	0	No output from algorithm.
WHRM Algorithm Mode 1 (MAX32664GWEB, MAX32664GWEC)	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

SENSOR OR ALGORITHM	SIZE (BYTES)	DESCRIPTION
WSpO$_2$ Algorithm Mode 1 (MAX32664GWEC)	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 excessve motion.
	1	WSpo2 State: Algorithm reported state: 0: Adjustment state 1: Computation state 2: Successful state 3: Timeout state
Wearable Suite Algorithm (WHRM+WSPO2) Mode 1 (MAX32664GWEC)	1	Operating Mode: 0: Continuous HRM and Continuous Spo2 1: Continuous HRM and One-shot Spo2 2: Continuous HRM 3: Sampled HRM 4: Sampled HRM and One-shot Spo2 5: Activity tracking 6: SpO2 calibration
	2	Heart Rate (bpm): 16-bit, LSB = 0.1 bpm
	1	HR Confidence level (0 - 100%): 8-bit, LSB = 1%
	2	Resperation Rate (period): 16-bit, LSB = 0.1 msec
	1	RR Confidence level (0 - 100%): 8-bit, LSB = 1%
	1	Activity class: 0: Rest 1: Other 2: Walk 3: Run 4: Bike 5: Rhythmic
	2	r: 16-bit LSB = 0.1 SpO2 r value.
	1	SpO$_2$ Confidence level (0 - 100%): 8-bit, LSB = 1%
	2	SpO$_2$ value (0 - 100%): 16-bit, LSB = 0.1%
	1	SpO$_2$ percent complete (oly applicable for one-shot mode), (0 - 100%): 8-bit, LSB = 1%
	1	SpO$_2$ low signal quality: 0: good quality 1: low quality
	1	SpO$_2$ motion: 0: little or no motion 1: excessive motion
	1	SpO$_2$ low perfusion: 0: normal perfusion 1: low perfusion
	1	SpO$_2$ unreliable measurement of fr: 0: measurement of r is reliable 1: measurement of r is not reliable
	1	SpO$_2$ State 0: LED adjustment 1: Computation 2: Success 3: Timeout

SENSOR OR ALGORITHM	SIZE (BYTES)	DESCRIPTION
BPT (MAX32664GWD)	1	Calibration/estimation mode status. 0: No Signal 1: In Progress 2: Successful 3: Bad Signal 4: Implied Motion Detected 5: Failure 6: Calibration Done
	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.
	2	SpO_2 value (70% - 100%): 16-bit, LSB = 0.1%
	2	r: 16-bit LSB = 0.1 SpO2 r value.
	1	HRaboveResting flag: a flag showing non-resting heart rate (>95 bpm)

Table 9 provides the sequence of commands for writing external accelerometer data to the input FIFO.

Table 9. Sequence of Commands to Write External Accelerometer Data to the Input FIFO (MAX32664GWEA/B/C)

HOST COMMAND	COMMAND DESCRIPTION	READ MAX32664 RESPONSE	RESPONSE DESCRIPTION
0xAA 0x10 0x00 0x03†	Set output mode to sensor and algorithm data.	0xAB 0x00	No error.
0xAA 0x10 0x01 0x0F*	Enable the input FIFO for host supplied accelerometer data.	0xAB 0x00	No error.
0xAA 0x44 0x03 0x01*	Enable the MAX30101 sensor. (MAX32664GWEA)	0xAB 0x00	No error.
0xAA 0x44 0x04 0x01 0x01*	Enable the input FIFO for host supplied accelerometer data.	0xAB 0x00	No error.
0xAA 0x52 0x02 0x01*	Enable MaximFast algorithm mode 1. (MAX32664GWEA)	0xAB 0x00	No error.
0xAA 0x13 0x00 0x04†	Read the sensor sample size for the accelerometer. (optional)	0xAB 0x00 0x06	No error. 6 bytes is the sample size.
0xAA 0x14 0x00 Sample 1 values, …, Sample N values*	Write data to the input FIFO. Six bytes per accelerometer sample.	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.
0xAA 0x12 0x01*	Read the data stored in the FIFO.	0xAB 0x00 0x03 0x6A 0x43 0x03 0x04 0x92 0x00 0x00 0x00 0x00 0x2E 0x15 0xFC 0xD8 0x00 0x04 0x02 0x3e 0x02 0x76 0x63 0x03 0xE4 0x03, data for fourteen other samples	No error. IR counts = 223811, Red counts = 19778, LED3 = 0, LED4 = 11797, X accelerometer = -0.808, Y accelerometer = 0.004, Z accelerometer = 0.574, Heart Rate = 63.0, Confidence = 99, SpO_2 = 99.6, MaximFast State Machine Status = 3, data for fourteen other samples.

**Mandatory*
†Recommended

MAX32664 I²C Annotated Application Mode Example

Table 10 shows a capture of the I²C traffic between the example host microcontroller (MAX32630FTHR) and the MAX32664GWEA for commanding the MAX32664GWEA to stream sensor and algorithm data (mode 1). The MAXREFDES220# is used for this example.

Table 10. MAX32664GWEA I²C Annotated Application Mode Example

HOST COMMAND	COMMAND DESCRIPTION	READ 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 0x0 0x0A 0x01 0x00	No error. Version is 10.1.0
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

HOST COMMAND	COMMAND DESCRIPTION	READ MAX32664 RESPONSE	RESPONSE DESCRIPTION
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 sensor and algorithm data.	0xAB 0x00	No error.
0xAA 0x10 0x01 0x0F*	Set FIFO threshold to 0x0F. Increase or decrease this value if you want more or fewer samples per interrupt.	0xAB 0x00	No error.
0xAA 0x52 0x00 0x01†	Enable AGC algorithm.	0xAB 0x00	No error.
0xAA 0x44 0x03 0x01*	Enable the MAX30101 sensor.	0xAB 0x00	No error.
0xAA 0x44 0x04 0x01 0x00†	Enable the sensor hub accelerometer. (Only enable if the board has an accelerometer.)	0xAB 0x00	No error.
0xAA 0x52 0x02 0x01*	Enable MaximFast algorithm mode 1.	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.
0xAA 0x12 0x01*	Read the data stored in the FIFO.	0xAB 0x00 0x03 0x6A 0x43 0x03 0x04 0x92 0x00 0x00 0x00 0x00 0x2E 0x15 0xFC 0xD8 0x00 0x04 0x02 0x3e 0x02 0x76 0x63 0x03 0xE4 0x03, data for fourteen other samples	No error. IR counts = 223811, Red counts = 19778, LED3 = 0, LED4 = 11797, X accelerometer = -0.808, Y accelerometer = 0.004, Z accelerometer = 0.574, 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 0x03 0x6A 0x43 0x03 0x04 0x92 0x00 0x00 0x00 0x00 0x2E 0x15 0xFC 0xE2 0x00 0x07 0x02 0x36 0x02 0x76 0x63 0x03 0xE4 0x03, data for fourteen other samples	No error. IR counts = 223811, Red counts = 19778, LED3 = 0, LED4 = 11797, X accelerometer = -0.798, Y accelerometer = 0.007, Z accelerometer = 0.566, 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

Table 11 shows a capture of the I²C traffic between the example host microcontroller (MAX32630FTHR) and the MAX32664GWEB for commanding the MAX32664GWEB to stream sensor and algorithm data. The MAXREFDES101# is used for this example.

Table 11. MAX32664GWEB I²C Annotated Application Mode Example

HOST COMMAND	COMMAND DESCRIPTION	READ MAX32664 RESPONSE	RESPONSE DESCRIPTION

HOST COMMAND	COMMAND DESCRIPTION	READ MAX32664 RESPONSE	RESPONSE DESCRIPTION
0xAA 0x10 0x00 0x03*	Set output mode to sensor and algorithm data.	0xAB 0x00	No error.
0xAA 0x10 0x01 0x05*	Set FIFO threshold to 0x05. Increase or decrease this value if you want more or less frequent samples.	0xAB 0x00	No error.
0xAA 0x44 0x00 0x01*	Enable the MAX86140/MAX86141 sensor.	0xAB 0x00	No error.
0xAA 0x44 0x04 0x01*	Enable the sensor hub accelerometer.	0xAB 0x00	No error.
0xAA 0x52 0x02 0x01*	Enable WHRM algorithm.	0xAB 0x00	No error.
0xAA 0x42 0x00†	Get the MAX86140/MAX86141 register attributes.	0xAB 0x00 0x01 0x2B	No error. Attributes are 1 byte, 0x2B registers available.
0xAA 0x43 0x00	Read all the MAX86140/MAX86141 registers.	0xAB 0x00 0x00 0x00 0x01 0x00 0x02 0x86 … 0x42 0x00	No error. Reg 0x00 = 0, Reg 0x01 = 0, Reg 0x02 = 0x86, …, Reg 0x42 = 0
0xAA 0x40 0x00 0x011 0x3F†	Set the MAX86140/MAX86141 register 0x11 to 0x3F (PPG2_ADC_RGE = 32768, PPG1_ADC_RGE = 32768, PPG_TINT = 117.3).	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 0x08	No error. Eight samples are in the FIFO.
0xAA 0x12 0x01*	Read the data stored in the FIFO.	0xAB 0x00 0x03 0X92 0x03 0x00 0x00 0x00 0x00 0x00 0x00 0x05 0x1C 0x29 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x96 0x0B 0x83 0xFC 0x37 0x02 0x92 0x63 0x00 0x00 0x00 data for seven other samples	No error. LED1 PhotoDiode1 counts = 235891, LED2 PD1= 0, LED3 PD1= 0, LED1 PD2 counts = 338247, LED2 PD2 = 0, LED 3 PD2 = 0, X accelerometer = 0.15, Y accelerometer = 2.947, Z accelerometer = -0.969, Heart Rate = 65.8, Confidence = 99, SpO_2 = 0, WHRM State Machine Status = 0, data for seven 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 0x08	No error. Eight samples available.
0xAA 0x12 0x01*	Read the data stored in FIFO.	0xAB 0x00 0x03 0X99 0x73 0x00 0x00 0x00 0x00 0x00 0x00 0x05 0X29 0x47 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x97 0x0B 0xB3 0xFC 0x36 0x02 0x8F 0x63 0x00 0x00 0x00 data for seven other samples	No error. LED1 PhotoDiode1 counts = 235891, LED2 PD1 = 0, LED3 PD1 = 0, LED1 PD2 counts = 338247, LED2 PD2 = 0, LED3 PD2 = 0, X accelerometer = 0.151, Y accelerometer = 2.947, Z accelerometer = -0.97, Heart Rate = 65.5, Confidence = 99, SpO_2 = 0, WHRM State Machine Status = 0, data for seven 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 0x08	No error. Eight samples available.
0xAA 0x12 0x01*	Read the data stored in FIFO.	0xAB 0x00 …	No error. Data read.

**Mandatory*

†Recommended

Table 12. MAX32664GWEC (WHRM) I²C Annotated Application Mode Example

Table 12 shows a capture of the I²C traffic between the example host microcontroller (MAX32630FTHR) and the MAX32664GWEC for commanding the MAX32664GWEC to stream sensor and algorithm data. This example assumes that the MAX32664GWEC has been flashed with the for the WHRM algorithm (v30.2.x).

HOST COMMAND	COMMAND DESCRIPTION	READ MAX32664 RESPONSE	RESPONSE DESCRIPTION
0xAA 0x10 0x00 0x03†	Set output mode to sensor and algorithm data.	0xAB 0x00	No error.
0xAA 0x10 0x01 0x05*	Set FIFO threshold to 0x05. Increase or decrease this value if you want more or less frequent samples.	0xAB 0x00	No error.
0xAA 0x46 0x00 0x03† (Only available for MAXM86161)	Perform MAXM86161 calibration.	0xAB 0x00	No error.
0xAA 0x44 0x00 0x01*	Enable the MAX86140/MAX86141/MAXM86161 sensor.	0xAB 0x00	No error.
0xAA 0x44 0x04 0x01*	Enable the sensor hub accelerometer.	0xAB 0x00	No error.
0xAA 0x52 0x05 0x02*	Configure WSpO2 algorithm to continuous.	0xAB 0x00	No error.
0xAA 0x52 0x05 0x01*	Enable WSpO2 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 0x08	No error. Eight samples are in the FIFO.
0xAA 0x12 0x01*	Read the data stored in the FIFO.	0xAB 0x00 0x00 0X00 0x00 0x02 0x50 0xA6 0x02 0xCA 0x71 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0xFC 0xD8 0x00 0x04 0x02 0X3E 0x00 0x29 0x02 0XB0 0x63 0x03 0xDE 0x00 0x00 0x00, 0x00 data for seven other samples	No error. Seq1Data= 0, Seq2Data Red LED = 151718, Seq3Data IR LED = 182897, Seq4Data = 0, Seq5Data = 0, Seq6Data = 0, X accelerometer = -0.808, Y accelerometer = 0.004, Z accelerometer = 0.574, r = 4.1, Heart Rate = 68.8, Confidence = 99, SpO_2 = 99.0, Progress = 0, Low SNR = 0, Motion = 0, SpO2 State = 2, data for seven 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 0x08	No error. Eight samples available.
0xAA 0x12 0x01*	Read the data stored in FIFO.	0xAB 0x00 0x00 0X00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x02 0x52 0XF8 0x02 0xCB 0x83 0x00 0x00 0x00 0xFC 0xE5 0x00 0x07 0x02 0X26 0x00 0x29 0x02 0XB0 0x63 0x03 0xDE 0x00 0x00 0x00, 0x00 data for seven other samples	No error. Seq1Data= 0, Seq2Data = 0, Seq3Data = 0, Seq4Data Red LED = 152312, Seq5Data IR LED = 183171, Seq6Data = 0, X accelerometer = -0.795, Y accelerometer = 0.007, Z accelerometer = 0.55, r = 4.1, Heart Rate = 68.8, Confidence = 99, SpO_2 = 99.0, Progress = 0, Low SNR = 0, Motion = 0, SpO2 State = 2, data for seven 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 0x08	No error. Eight samples available.
0xAA 0x12 0x01*	Read the data stored in FIFO.	0xAB 0x00 …	No error. Data read.

**Mandatory*

†Recommended

Table 13. MAX32664GWEC (WHRM+WSpO2) I²C Annotated Application Mode Example

Table 13 shows a capture of the I²C traffic between the example host microcontroller (MAX32630FTHR) and the MAX32664GWEC for commanding the MAX32664GWEC to stream sensor and algorithm data with SCD disabled and AGC on. This example assumes that the MAX32664GWEC has been flashed with the for the WHRM+SpO2 algorithm. This example sets the report rate to once every sensor sample, and that will cause the report to be every 40 msec since the sample rate is 25 Hz. For WHRM+SpO2, the host is required to poll the MAX32664GWEC for available reports.

HOST COMMAND	COMMAND DESCRIPTION	READ MAX32664 RESPONSE	RESPONSE DESCRIPTION
0xAA 0x10 0x00 0x03†	Set output mode to sensor and algorithm data.	0xAB 0x00	No error.
0xAA 0x10 0x01 0x05*	Set FIFO threshold to 0x05. Increase or decrease this value if you want more or less frequent samples.	0xAB 0x00	No error.
0xAA 0x44 0x10 0x01 0x01*	Set report rate to be one report per every sensor sample.	0xAB 0x00	No error.
0xAA 0x44 0x00 0x01*	Enable the MAX86141 sensor.	0xAB 0x00	No error.
0xAA 0x44 0x04 0x01*	Enable the sensor hub accelerometer.	0xAB 0x00	No error.
0xAA 0x50 0x07 0x0A 0x00*	Configure WHRM+SpO2 algorithm to continuous.	0xAB 0x00	No error.
0xAA 0x50 0x07 0x0B 0x01*	Enable AEC	0xAB 0x00	No error.
0xAA 0x50 0x07 0x12 0x00*	Disable Auto PD Current Calculation	0xAB 0x00	No error.
0xAA 0x50 0x07 0x12 0x00*	Disable SCD	0xAB 0x00	No error.
0xAA 0x52 0x07 0x01*	Enable WHRM+SpO2 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 0x08	No error. Eight samples are in the FIFO.
0xAA 0x12 0x01*	Read the data stored in the FIFO.	0xAB 0x00 data for seven other samples	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 0x08	No error. Eight samples available.
0xAA 0x12 0x01*	Read the data stored in FIFO.	0xAB 0x00 data formatted per Table 8 for MAX86141 WHRM+SpO2, KX-122 Accelerometer Wearable Suite Algorithm (WHRM+WSPO2) and data for seven other sample sets	No error. Data for eight sample sets.
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 0x08	No error. Eight samples available.
0xAA 0x12 0x01*	Read the data stored in FIFO.	0xAB 0x00 …	No error. Data read.

**Mandatory*

†Recommended

Table 14 shows a capture of the I2C traffic between the example host microcontroller (MAX32630FTHR) and the MAX32664GWED for commanding the MAX32664GWED to stream sensor and algorithm data. The MAXREFDES220# is used for this example.

Table 14. MAX32664GWED I2C Annotated Application Mode Example

HOST COMMAND	COMMAND DESCRIPTION	READ MAX32664 RESPONSE	RESPONSE DESCRIPTION
Host sends commands to configure BPT user profile and then sends commands to start the user calibration while user finger is on the sensor.			
0xAA 0x50 0x04 0x01 0x79 0x77 0x7A*	Configure BPT algorithm user profile for cuff readings of diastolic 121, 119, 122	0xAB 0x00	No error.
0xAA 0x50 0x04 0x02 0x51 0x4F 0x52*	Configure BPT algorithm cuff readings to diastolic 81, 79, 82	0xAB 0x00	No error.
0xAA 0x50 0x04 0x05 0x00†	Configure BPT algorithm to user is resting.	0xAB 0x00	No error.
0xAA 0x50 0x04 0x00 0x00†	Configure BPT algorithm to user is not using BP medication.	0xAB 0x00	No error.
0xAA 0x10 0x00 0x03†	Set output mode to sensor and algorithm data.	0xAB 0x00	No error.
0xAA 0x10 0x01 0x0F*	Set FIFO threshold to 0x0F. Increase or decrease this value if you want more or less frequent samples.	0xAB 0x00	No error.
0xAA 0x44 0x03 0x01*	Enable the MAX30101 sensor.	0xAB 0x00	No error.
0xAA 0x52 0x04 0x01*	Enable BPT algorithm calibration mode.	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 0x11	No error. Seventeen samples are in the FIFO.
0xAA 0x12 0x01*	Read the data stored in the FIFO.	0xAB 0x00 0x03 0x25 0xF4 0x02 0x6E 0x22 0x00 0x00 0x00 0x00 0x00 0x2C 0x04 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 data for sixteen other samples	No error. LED1 IR = 206324, LED2 = 159266, LED3 = N/A, LED4 = N/A, Status = 4, Progress = 0, Heart Rate = 0.0, Systolic = 0, Diastolic = 0, SpO2 = 0, r = 0, HRaboveResting = 0, data for sixteen 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 the FIFO.	0xAB 0x00 0x11	No error. Seventeen samples are in the FIFO.
0xAA 0x12 0x01*	Read the data stored in the FIFO.	0xAB 0x00 0x03 0xC4 0x39 0x02 0xD5 0xF1 0x00 0x00 0x00 0x00 0x0F 0x39 0x02 0x64 0x02 0x76 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 data for sixteen other samples	No error. LED1 IR = 206324, LED2 = 159266, LED3 = N/A, LED4 = N/A, Status = 2, Progress = 100%, Heart Rate = 63.0, Systolic = 0, Diastolic = 0, SpO2 = 0, r = 0, HRaboveResting = 0, data for sixteen other samples.
Host sends commands to BPT algorithm to make an read the estimated BP while the user has finger on the sensor.			
0xAA 0x10 0x00 0x03†	Set output mode to sensor and algorithm data.	0xAB 0x00	No error.
0xAA 0x10 0x01 0x0F*	Set FIFO threshold to 0x0F. Increase or decrease this value if you want more or less frequent samples.	0xAB 0x00	No error.
0xAA 0x52 0x00 0x01†	Enable AGC.	0xAB 0x00 0x00	No error.

HOST COMMAND	COMMAND DESCRIPTION	READ MAX32664 RESPONSE	RESPONSE DESCRIPTION
0xAA 0x44 0x03 0x01*	Enable the MAX30101 sensor.	0xAB 0x00	No error.
0xAA 0x52 0x04 0x02*	Enable BPT algorithm estimation mode.	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 0x11	No error. Seventeen samples are in the FIFO.
0xAA 0x12 0x01*	Read the data stored in the FIFO.	0xAB 0x00 0x03 0x25 0xF4 0x02 0xD5 0xF1 0x00 0x00 0x00 0x00 0x00 0x2C 0x04 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 data for sixteen other samples	No error. LED1 IR = 246841, LED2 = 185841, LED3 = N/A, LED4 = N/A, Status = 4, Progress = 0, Heart Rate = 0.0, Systolic = 0, Diastolic = 0, SpO2 = 0, r = 0, HRaboveResting = 0, data for sixteen 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 the FIFO.	0xAB 0x00 0x0F	No error. Fifteen samples are in the FIFO.
0xAA 0x12 0x01*	Read the data stored in the FIFO.	0xAB 0x00 0x03 0x9E 0x32 0x02 0xF5 0x6c 0x00 0x00 0x00 0x00 0x1B 0XD2 0x02 0x64 0x02 0XA8 0x77 0x4A 0x03 0xE8 0x00 0x00 0x00 data for fourteen other samples	No error. LED1 IR = 206324, LED2 = 159266, LED3 = N/A, LED4 = N/A, Status = 2, Progress = 100%, Heart Rate = 68.0, Systolic = 119, Diastolic = 74, SpO2 = 100.0, r = 0, HRaboveResting = 0, data for fourteen other samples.
Host reads user profile and calibration data			
0xAA 0x51 0x04 0x03†	Read the 824 bytes of calibration data for this user.	0xAB 0x00 0x2E 0xE8 0x02 0x00 0x88 0x39 0x02 0x00 0x79 0x00 eight hundred fourteen more bytes of calibration data	No error. 824 bytes of user profile and calibration data.
Host writes user profile and calibration data			
0xAA 0x50 0x04 0x03 0x2E 0xE8 0x02 0x00 0x88 0x39 0x02 0x00 0x79 0x00 eight hundred fourteen more bytes of calibration data †		0xAB 0x00	No error.

**Mandatory*
†Recommended

I²C Commands to Flash the Application Algorithm/Firmware

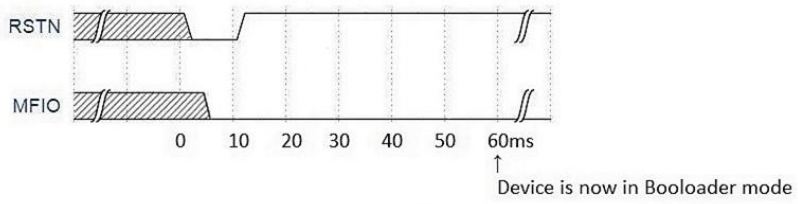
The MAX32664 is pre-programmed with bootloader firmware which accepts in-application programming of the Maxim supplied application algorithm/firmware file (.msbl). Table 15 is a capture of the I²C commands that are necessary to flash the application algorithm/firmware to the MAX32664.

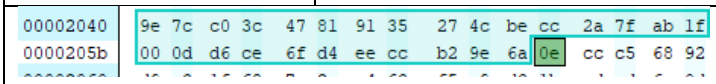
IMPORTANT: *Do not enable the accelerometer if your board does not have the accelerometer.*

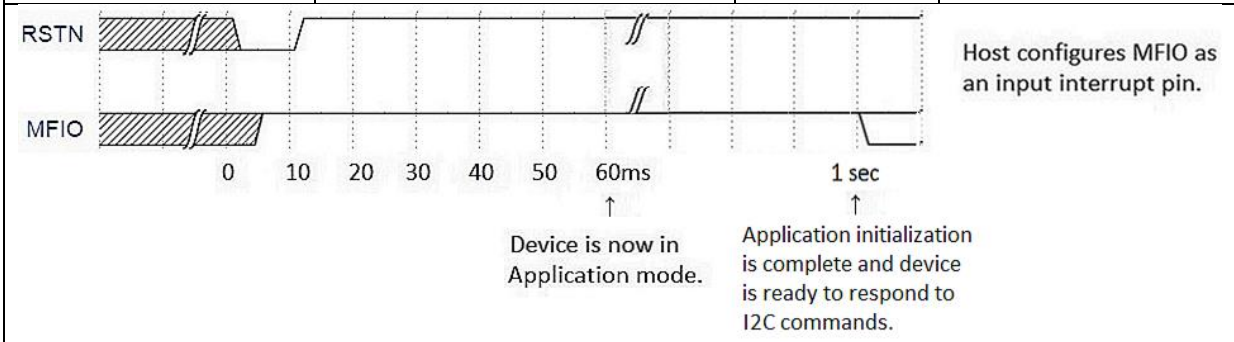
This example was captured with the MAX32630FTHR acting as the host microcontroller. The MAX32664 uses the 8-bit slave address of 0xAA. The example encrypted algorithm file used was the MAX32660_SmartSensor_OS24_MaximFast_1.8.2a.msbl (26 pages, 8196 bytes for the page size). Each

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 15. Annotated I²C Trace for Flashing the Application

HOST COMMAND	COMMAND DESCRIPTION	READ MAX32664 RESPONSE	RESPONSE DESCRIPTION
Sequence the MAX32664 to enter bootloader mode. *  *Figure 8. 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 the value at 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 9. 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 10. 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.
00000030 13 87 b9 00 2b f5 ad cd 2e 47 d2 83 23 88 37 63 00000043 02 ed 27 af 1a 00 00 20 04 00 00 00 c2 31 90 2c *Figure 11. 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.
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 0000006f e2 23 4f 71 d4 6b 98 e3 a7 f9 85 80 7a 4e 17 e7			

HOST COMMAND	COMMAND DESCRIPTION	READ MAX32664 RESPONSE	RESPONSE DESCRIPTION
 *Figure 12. 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	No error.
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.
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. *			

HOST COMMAND	COMMAND DESCRIPTION	READ MAX32664 RESPONSE	RESPONSE DESCRIPTION
 *Figure 13. 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

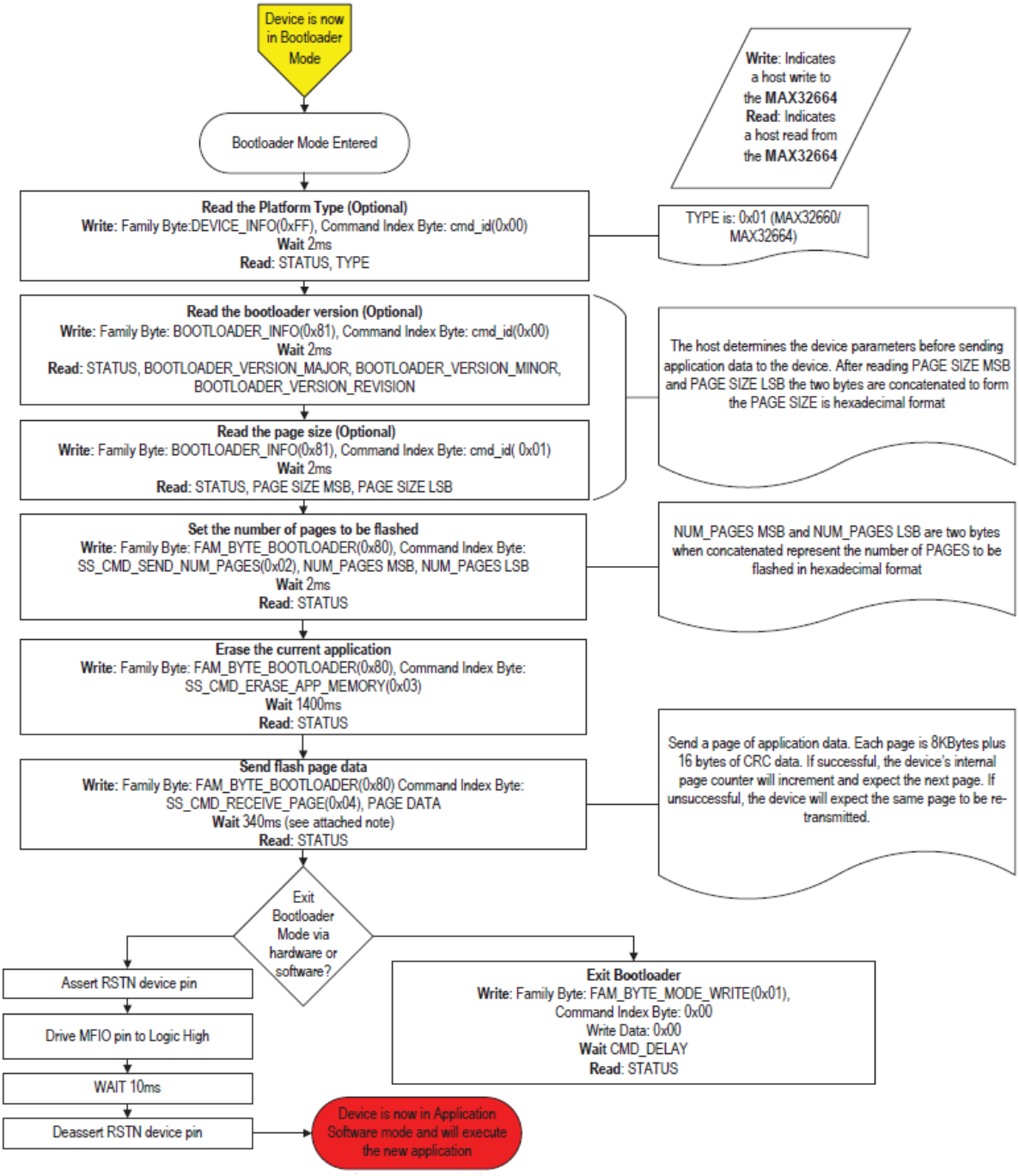
It is recommended to program the latest version of the MAX32664 sensor hub application algorithm/firmware .msbl file into the MAX32664 chip. Check the version that is programmed into the chip by using the command “Identity, Read sensor hub version.” The latest sensor hub algorithm/firmware is available 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 application algorithm/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 14 is a flowchart of the in-application programming.



Note: 340ms may need to be adjusted to 680ms or more when the reference design host is communicating with slower PCs/systems.

Figure 14. MAX32664 in-application programming flowchart.

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

Table 16 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 16. MAX32664 APIs and Methods for Reset, Sleep, Status, Heartbeat

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		WHRM+WSpO2 v30.3.x and WHRM v30.2.4+ use sleep/deep-sleep for low-powered mode between polling periods.
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 5 and Figure 6.	
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.

Default Application .msbl Versions Pre-Programmed on the MAX32664GWEA/B/C/D

The MAX32664GWEA/B/C/D are pre-programmed with the bootloader and the application .msbl application/sensor hub version listed in Table 17. The pre-programmed application .msbl versions are not updated by Maxim. The pre-programmed parts may not be programmed with the latest version of the .msbl application; It is recommended that application be updated to the latest .msbl which are available on the Maxim Integrated website.

Table 17. MAX32664GWEA/B/C/D Pre-Programmed .msbl Version

	MAX32664GWEA	MAX32664GWEB	MAX32664GWEC	MAX32664GWED
Pre-Programmed . msbl application/sensor hub version	Pre-programmed .msbl version 1.9.1	Pre-programmed .msbl version 20.1.2	Pre-programmed .msbl version 30.2.2	Pre-programmed .msbl version 40.2.2

Revision History

REVISION NUMBER	REVISION DATE	DESCRIPTION	PAGES CHANGED
0	01/19	Initial release	—
1	06/20	Add MAX32664B/C/D application firmware description Add additional GPIOs for MAXREFDES101. Add 0x11 to the commands. Change 0x14 0x04 to 0x14 0x00. Updated 0x44 0x04 command and Sequence of Commands for external host. Add PD1, PD2 to 0x51/0 0x05 0x09. Add additional modes to 0x52 0x02 and 0x52 0x04. Label commands in message protocol table and output format table as MAX32664A/B/C/D. Add mode 2 for MaximFast in output format table. Sequence of Commands table updated. Add annotated I2C traces for MAX32664GWEC, MAX32664GWED. Corrected timing for CMD_DELAY, and time for application to respond to I2C commands. Added Wearable Algo Suite (WHRM+WSPO2) and updated MFIO for WHRM+WSPO2. Added sensor configuration and MAXM86161 calibration.	