Measuring SpO_2 and Heart Rate Using the MAX32664C – A Quick Start Guide for Programmers

UG6924; Rev 8.3; 01/20/2022

Abstract

The MAX32664C is a variant of the MAX32664 sensor-hub family, which is specifically targeted for measurement of SpO2 and heart rate. Combined with the MAX86141/40 /MAXM86161/ MAXM86146/MAX86174 optical sensor and a 3-axis accelerometer, it provides the sensor's raw data, as well as calculated SpO2 and heart rate data, to a host device through its I2C slave interface. This document provides step-by-step instructions that enable a user to communicate with the MAX32664C and to calibrate, configure, and receive measurement and monitoring data..

Table of Contents

List of Figures

List of Tables

Introduction

MAX32664C is a Maxim sensor hub product which provides the following innovative features:

- Biometric algorithms developed by a team of machine learning, data scientist, and algorithm experts.
- AFE sensor drivers and algorithm embedded on a chip.
- OTA update compatibility allows the sensor hub to receive the latest algorithm version.
- Low-powered sensor hub to manage AFE and accelerometer data collection.
- Faster time to market; development time cut by at least six months.
- Sample host code reduces integration time.
- Reference design includes a host processor and Android app which allows for wrist-band evaluation.

The sensor hub provides heart rate and SpO_2 firmware algorithms on an IC. The embedded firmware includes the drivers that are required to interface with an optical sensor device, such as the MAX86141. The host microcontroller retrieves raw and processed data from the sensor hub via an I^2C interface.

To properly capture and calculate the data, this biometric solution requires an accelerometer. The sensor hub firmware includes the required drivers for the accelerometer. Alternatively, a host-side accelerometer can be used. In this case, the sampled accelerometer data must be periodically reported to the sensor hub by the host microcontroller using commands described in this application note.

When using the MAXREFDES103, the following three software and firmware must all be updated using the latest and same software release package.

1. Install the Maxim DeviceStudio PC GUI using the **.msi** file
2. Flash the micro board **.bin** file using drag & drop to the DAPLINK folder
3. Flash the algorithm .**msbl** file to MAX32664 using the Maxim DeviceStudio PC GUI

NOTE: The instructions in this document are compatible with: the MAX32664C firmware version 30.13.19+ (MAX86141), 32.9.23+ (MAXM86161+KX122), 32.13.12+ (MAXM86161+LIS2DS12) , or 33.13.19+ (MAXM86146) and later. If you are using older firmware, make sure to upgrade the firmware.

1 Architecture

An accelerometer is mandatory for MAX32664C heart rate monitoring. A KX122 or LIS2DS12 accelerometer may be connected directly to the sensor hub. The interrupt line of the accelerometer may be connected to the sensor hub to support power saving using motion detection. Alternatively, an external 3-axis host-side accelerometer can be used. In this case, the host needs to periodically provide accelerometer readings to the sensor hub using the commands provided in this document. MAX32664C releases for MAX86174 sensor currently supports only LIS2DS12 accelerometer.

The optical sensor utilizes green, infrared (IR), and red LEDs to transmit pulses and one or more photodiodes (PD) to collect reflected or residual light. By default, the heart rate monitoring algorithm for MAXREFDES103 uses green1 LED (LED1) and two PDs (PD1 and PD2). The SpO_2 employs one IR LED (LED2) and one red LED (LED3) with one PD (PD1).

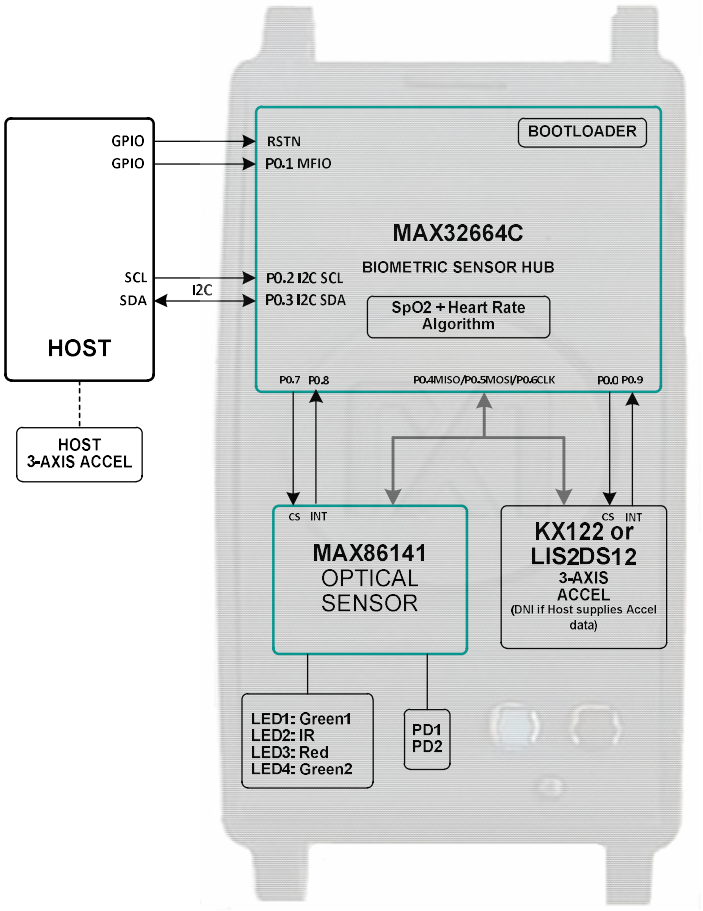


Figure 1. Architecture diagram for health-sensing applications using the MAX32664C sensor hub and the MAX86141 sensor (SPI). Kionix KX122 or STM LIS2DS12 accel are supported (30.13.12+)

Kionix is a registered trademark of Kionix, Inc.

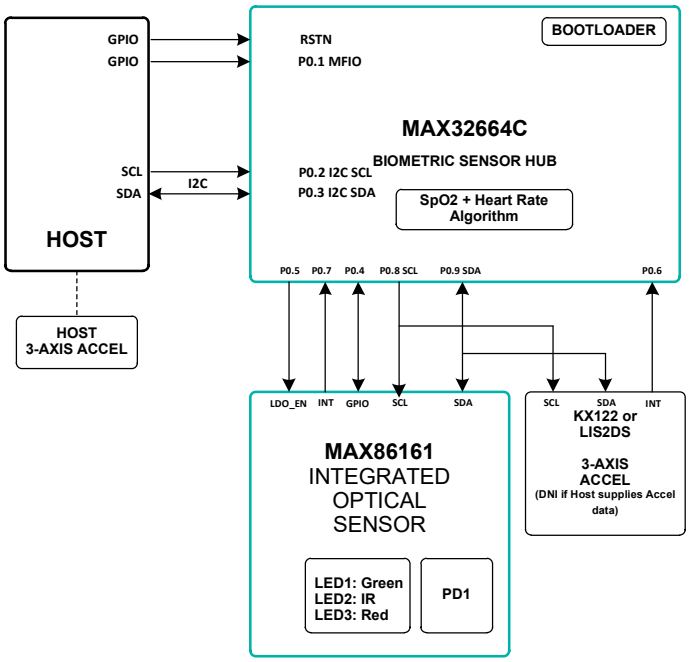


Figure 2. Architecture diagram for health-sensing applications using the MAX32664C sensor hub and the MAX86161 sensor (I^2C). Kionix KX122 or STM LIS2DS12 accel are supported (32.13.12+).

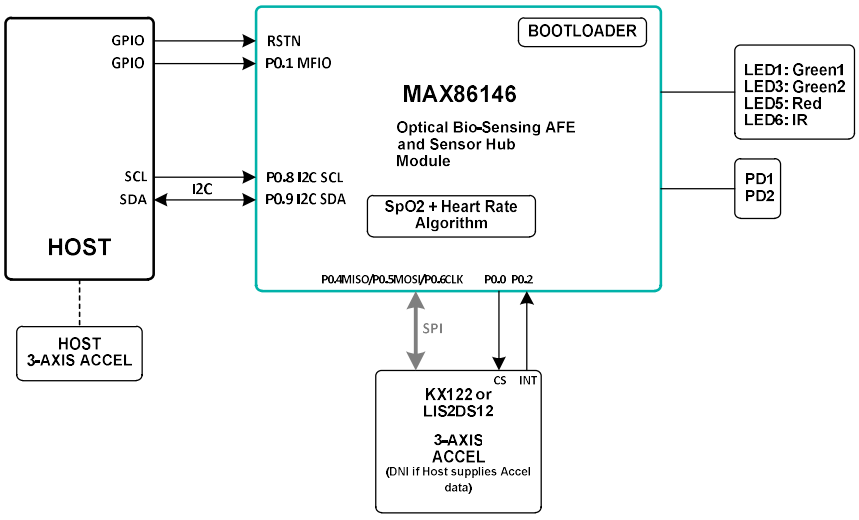


Figure 3. Architecture diagram for health-sensing applications using MAXM86146 Module. Kionix KX122 or STM LIS2DS12 accel (SPI) are supported (33.13.12+).

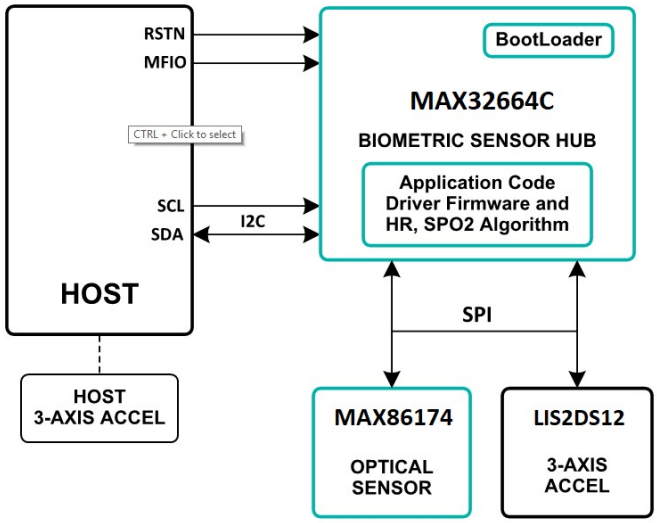


Figure 4. Architecture diagram for health-sensing applications using MAX86174 Module. STM LIS2DS12 accel (SPI) is supported (35.x.y).

1.1 Reset to Bootloader or Application Mode

A typical health-sensing design includes a host microcontroller that communicates with the sensor hub through the I²C bus. Two GPIO pins are needed to control the reset and the startup in Application or Bootloader mode through the RSTN and multifunction input/output (MFIO) pins.

To enter Bootloader mode:

- Set the RSTN pin to low for 10ms.
- While RSTN is low, set the MFIO pin to low. (The MFIO pin should be set to low at least 1ms before the RSTN pin is set to high.)
- After the 10ms has elapsed, set the RSTN pin to high.
- After an additional 50ms has elapsed, the sensor hub is in Bootloader mode.

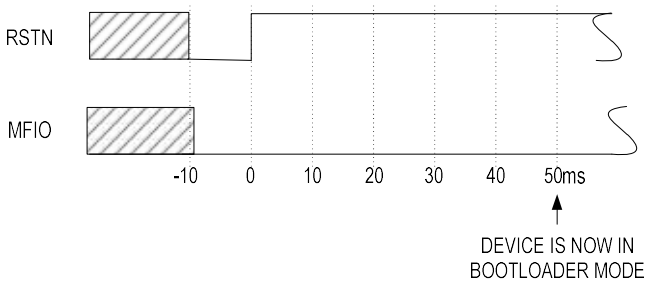


Figure 5 Entering bootloader mode using the RSTN pin and the MFIO GPIO pin.

To enter Application mode:

- Set the RSTN pin to low for 10ms.

- While RSTN is low, set the MFIO pin to high.
- After the 10ms has elapsed, set the RSTN pin to high. (The MFIO pin should be set to high at least 1ms before the RSTN pin is set to high.)
- After an additional 50ms has elapsed, the sensor hub is in Application mode and the application performs its initialization of the application software.
- After approximately 1.5 second from when the RSTN pin was set to high, the application completes the initialization, and the device is ready to accept I²C commands.

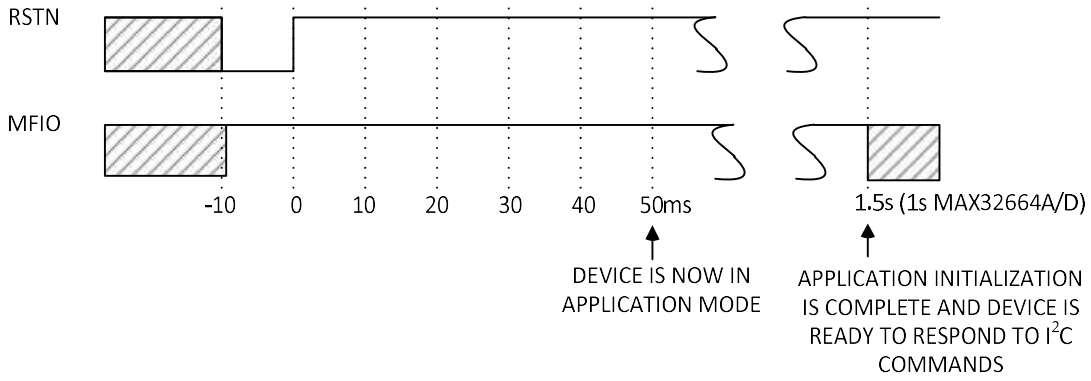


Figure 6 Entering application mode using the RSTN pin and MFIO pin.

To enter Application mode by timing out from Bootloader mode:

- Set the RSTN pin to low for 10ms.
- While RSTN is low, set the MFIO pin to low. (The MFIO pin should be set to low at least 1ms before the RSTN pin is set to high.)
- After the 10ms has elapsed, set the RSTN pin to high.
- After an additional 50ms has elapsed, the sensor hub is in Bootloader mode.
- If no I2C commands are sent to the sensor hub within the next 1s, then the sensor hub will automatically switch to application mode.

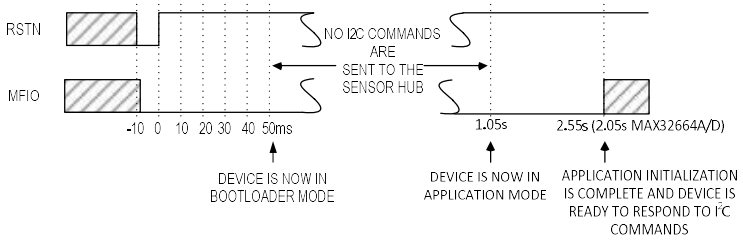


Figure 4 Entering application mode by timing out from Bootloader mode.

1.2 Sensor Hub Handshaking

1.2.1 Normal MFIO Mode (default, host periodically empties the FIFO)

Normally, when the sensor hub is idle, it switches to deep sleep mode to save power. An external interrupt-like sensor, host MFIO, or RTC alarm forces the sensor hub to wake up. Do not keep the MFIO pin at a constant low – when the MFIO pin is high, the sensor hub wakes up to service the AFE.

The host is required to wake up the sensor hub prior to any I2C communication by:

- Setting the MFIO pin to low at least 300µs before the beginning of an I2C transaction to wake the sensor hub.
- Keeping the MFIO pin low during the I2C transaction.
- Setting MFIO to high after the end of I2C communication to allow the sensor hub to switch back to deep sleep/AFE servicing mode.

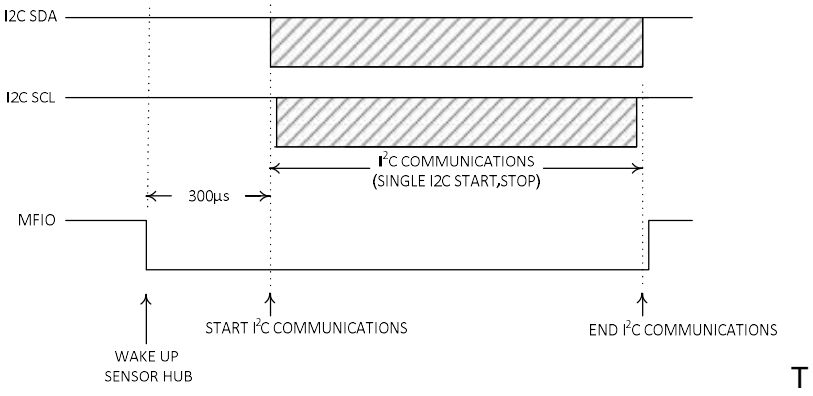


Figure 5. Normal MFIO Mode, host uses MFIO for enabling host communications.

1.2.2 MFIO Interrupt Mode

MFIO interrupt mode is mode where FIFO pin is used by the sensor hub to interrupt Host when: measurement data is ready, report FIFO overflow events and skin contact detection event (when SCD state machine is enabled – see section SCD State Machine for MFIO Interrupt Mode).

- To set up for MFIO interrupt mode, the host sets the MFIO pin to input mode and assigns a GPIO interrupt handler to this pin to receive event reporting from the sensor hub.
- The host initializes the sensor hub to AEC or AEC mode.
- sensor hub enters MFIO interrupt mode with command sequence 0xB8, 0x01.
- After the interrupt is received by the host, the host should wake the sensor hub. When in MFIO interrupt mode, the sensor hub is woken up by:
  - Host sends byte 0x00 to I2C slave address 0x00 and waits for 200us
  - Host sends bytes 0x00 to I2C slave address 0x00 and waits for 150us
- Host sends commands to read samples in the output FIFO.

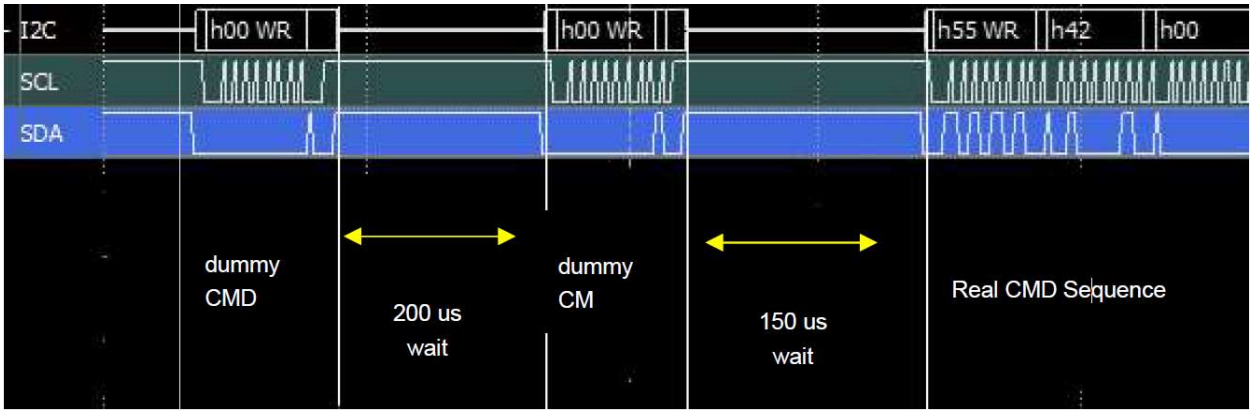


Figure 6. SDA wake-up sequence for sensor hub

MFIO interrupt mode is not defined for MAX86174 sensors. MAX32664C releases supporting MAX86174 sensor, 35.x.y firmware versions, only woke up via MFIO pin.

1.3 Sensor Hub I2C Communications

A host uses the I²C bus to communicate with the sensor hub (slave) using a series of commands. See the MAX32664 User Guide for details. The default CMD_DELAY is 2ms.

A generic write command includes the following fields:

```
Slave_WriteAddress(1 byte)|Command_Family(1 byte)|Command_Index(1 byte)|Value(multiple bytes)
```

A generic response includes the following fields:

```
Slave_ReadAddress(1 byte)|Status(1 byte)|Value (multiple bytes)
```

`Slave_WriteAddress` and `Slave_ReadAddress` are set to 0xAA and 0xAB, respectively.

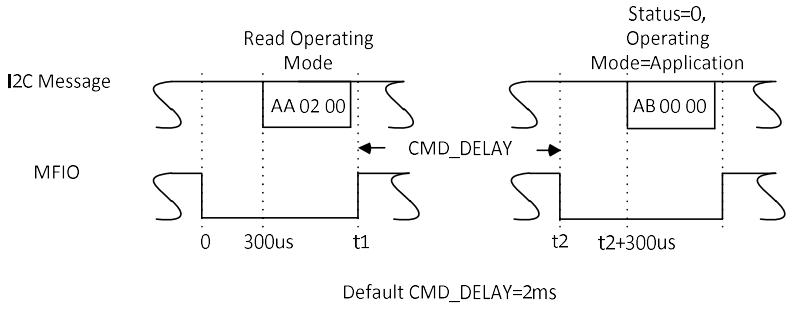


Figure 7. Example I²C, MFIO timing diagram.

The read status byte is an indicator of success (0x00) or failure, as shown in **Table 1**.

Table 1. Read Status Byte Value

STATUS BYTE VALUE	DESCRIPTION
0x00	The write transaction was successful.
0x01	Illegal Family Byte and/or Index Byte was used. Verify that the Family Byte, Index Byte are valid for the host command sent.

0x02	Illegal Index Byte and/or Write Byte was used. Verify that the Index Byte and Write Byte(s) are valid for the host command sent.
0x03	Incorrect number of bytes sent for the requested Family Byte. Verify that the correct number of bytes are sent for the host command.
0x04	Illegal configuration value was attempted to be set. Verify that the Index Byte is correct for Family Byte 0x44. Verify that the samples report period is not 0 for host command 0x10 0x02. Verify that the Write byte for host command 0x10 0x03 is in the valid range specified.
0x05	Not used in application mode. (In bootloader: Device is busy. Insert delay and resend the host command.)
0x80	Not used. General error while receiving/flashing a page during the bootloader sequence. Not used.
0x81	Bootloader checksum error while decrypting/checking page data. Verify that the keyed .msbl file is compatible with MAX32664A/B/C/D.
0x82	Bootloader authorization error. Verify that the keyed .msbl file is compatible with MAX32664A/B/C/D.
0x83	Bootloader detected that the application is not valid.
0xFE	Device is busy. Try again. Increase the delay before the command and increase the CMD_DELAY.
0xFF	Unknown error. Verify that the communications to the AFE/KX122/LIS2DS12 are correct by reading the PART_ID/WHO_AM_I register. For MAX32664B/C, the sensor hub is in deep sleep unless the host sets the MFIO pin low 300us before and during the I^2C communications. When switching to bootloader mode, allow 50ms for initialization. When switching to application mode, allow 1.5s for initialization.
NAK	NAK received. Sensor hub was busy. Resend command after 1ms with a maximum of five retries. If this issue persists, then empty the FIFO by reading all the data or reduce the report rate. Verify that the hardware I2C/MFIO rise times, voltage levels, and grounding are correct. For MAX32664B/C, verify that the MFIO line fall time is clean; increase the MFIO pin low time to wake to 300us. If this occurs when using the MFIO/RSTN lines to switch to BL or application mode, then increase the 10ms to 1.25s.

1.2 Power-Saving Considerations

1.2.1 Samples Report Rate

The sensor hub goes into deep sleep in Idle mode and wakes up on internal or external interrupts. To maximize the benefits of low power, the host may configure the samples report period of the algorithm to a longer time. In this case, the samples report is generated less frequently.

This samples report rate may be configured through an I2C command.

1.2.2 Host Reading Samples in Output FIFO Period and Samples Report Period

The host is required to regularly empty the measurement data in the sensor hub FIFO at a periodic rate. The periodic rate depends on the rate that the sensor hub samples report is generated. When the samples report period is reduced, the host will not need to empty the FIFO as often. If the FIFO becomes full, then the FIFO should be emptied.

The host may read samples in the output FIFO at a period (*host reading FIFO period*) five times the length of the samples report period (*samples report period*) to avoid FIFO overflow. In this example, an average of five samples will be in the output FIFO.

By default, the samples report period (read samples report period, 0x11 0x02) is set to 40ms. In this case, it is recommended that the host read samples from the output FIFO every 200ms (*host reading period*). At these rates, on average there will be five samples in the output FIFO for the host to read.

1.2.3 Samples Report Content

If the sensor data such as accelerometer and photoplethysmogram (PPG) signals are not required, the host may choose to request only algorithm data in the samples report. This reduces the I²C communication time and affects power consumption. This is performed by configuring the output mode to Algorithm Only.

1.3 Accelerometer

The sensor hub requires accelerometer data to function properly. An accelerometer is mandatory for a heart rate monitor to be able to compensate for the user's motion. Otherwise, the reported heart rate will not be correct during movement.

SpO_2 calculation requires a resting condition, and the algorithm uses accelerometer data to detect excessive motion. In such a condition, computation is paused, and the user is informed with a motion flag.

A sensor hub accelerometer can be integrated through the SPI port of the sensor hub. In this case, the required driver for KX122/KX112/LIS2DS12 is included in the latest application firmware. The user only needs to follow the reference schematics to connect the accelerometer and enable it before starting the algorithm, as described later in this document. Normally, the accelerometer is polled to collect samples. The interrupt line is only needed if the SCD-based power saving procedure is implemented in the host.

Alternatively, a host-side accelerometer can be used. However, this option requires strict timing synchronization between the sampled accelerometer data and PPG samples of ±40ms or less. To use the host-side accelerometer:

1. The host should start the accelerometer just before enabling the algorithm to maximize the initial synchronization between the PPG and accelerometer samples. However, accelerometer samples collected prior to receiving the confirmation of the algorithm enable I²C command should be discarded.
2. The host is required to use a 3-axis accelerometer at a 25Hz sampling rate. If a higher sampling rate is chosen, samples should be decimated to be synchronized with a 40ms PPG sampling time.
3. The host must queue five accelerometer samples and feed them at the same time to the sensor hub using the commands shown in the table below. The period of feeding samples should be 200ms. This is the longest delay that the sensor hub can tolerate to receive accelerometer samples.

Because the AFE sensor and the host accelerometer use different clock sources, exact synchronization between them is not possible. The sensor hub internally decimates or interpolates accelerometer samples to compensate for drift.

Table 2. Host-Side Accelerometer—Sending Data to the Sensor Hub

HOST COMMAND (HEX)	DESCRIPTION	SENSOR HUB RESPONSE (HEX)	DESCRIPTION

AA 44 04 01 01 (CMD_DELAY = 20ms)	Enable the host accelerometer.	AB 00	Success
AA 13 00 04	Read the sensor sample size for the accelerometer (optional).	AB 00 06	Success; 6 is the number of bytes per samples in FIFO
The following should be executed periodically at 200ms:			
AA 14 00 [Sample 1 values] … [Sample N values]	Write data to the input FIFO of the sensor hub. Each sample has three 2-byte integer values for X, Y, and Z in milli-g. N=5	AB 00	Success
AA 00 00	Read the sensor hub status.	AB 00 00	Success; sensor hub not busy

1.3.1 Converting Gs to Counts

The G's may be converted back to counts with the following psuedo code. The accelerometer resolution is set to 16 bits, 8G

```
float G_s
int plus_minus_counts
plus_minus_counts = (int)((float)(((G_s * 32768.0)+4.0)/8.0))
if (plus_minus_counts > 32767)
  plus_minus_counts =32767
else if (plus_minus_counts < -32768)
  plus_minus_counts = -32768
```

Eg., (int)((7.9997559*32768)+4)/8 =(int)(32767.5) =32767

2 Data Collection for SpO_2 Calibration Coefficients.

Due to variations in the physical design and optical cover lens of the final product, a calibration data collection procedure for SpO_2 is required to be performed once in a controlled environment. This procedure is important to ensure the quality of the SpO_2 calculation. This step is typically performed in a standard lab using the final form factor (with cover lens) with a reference SpO_2 device to determine three SpO_2 calibration coefficients: a, b, and c. The details of the SpO_2 calibration data collection and SpO2 coefficient derivation procedure are described in the **Guidelines for SpO_2 Measurement Using the Maxim MAX32664 Sensor Hub** application note.

Once the three SpO_2 calibrations coefficients are obtained, they need to be loaded to the sensor hub every time prior to starting the algorithm.

The SpO_2 calibrations coefficients need to be converted to a 32-bit integer format using the following:

- A_{int32} = round (10^5 x a)
- B_{int32} = round (10^5 x b)
- C_{int32} = round (10^5 x c)

For example, the default measured SpO_2 calibration coefficients for MAXREFDES103# are:

- a = 0.0
- b = -26.224999
- c = 112.317421

These are sent to the sensor hub using 32-bit signed integer format:

- A_{int32} = round (10^5 x a) = 0x00000000
- B_{int32} = round (10^5 x b) = 0xFFD7FBDD
- C_{int32} = round (10^5 x c) = 0x00AB61FE

The SpO_2 calibration coefficients may be stored in the host flash separately and loaded to the sensor hub after every reset.

The table below shows the sequence of commands for the enabling the SpO2 data collection mode.

Table 3. Host Commands—SpO_2 Calibration Data Collection Mode

	#	HOST COMMAND (HEX)	COMMAND DESCRIPTION	RESPONSE (HEX)
START ALGORITHM	Host initializes the sensor hub to SpO2 calibration data collection mode and starts the algorithm using following commands:			
	1.1	AA 10 00 03	Set the output FIFO mode to sensor + algorithm data (streamed data will include PPG accelerometer and algorithm data).	AB 00
	1.2	AA 10 01 01	Set the sensor hub interrupt threshold.	AB 00
	1.3	AA 10 02 01	Set the samples report period to 40ms. Samples report rate to be one report per every sensor sample.*	AB 00
	1.4	AA 44 04 01 01 (if host accelerometer is used, CMD_DELAY = 20ms)	Enable the host-side accelerometer, if used. Otherwise, the sensor hub accelerometer is used by default, and there is no need to enable it.	AB 00
	1.5	AA 50 07 0A 06	Set the mode to SpO_2 Calibration Data Collection mode	AB 00
	1.6	Optional: Any command to change the algorithm settings and configurations (**Table 12**) from the default setting should appear here BEFORE enabling the algorithm.		
	1.7	AA 52 07 01 (CMD_DELAY = 465 ms)	Enable the algorithm; the analog front-end (AFE) and sensor hub accelerometer will be enabled automatically.	AB 00
READING SAMPLES REPORT IN OUTPUT FIFO	Host reads samples periodically (do not execute at a faster rate than the samples report period):			
	2.1	AA 00 00	Read the sensor hub status byte: Bit 0: Sensor comm error Bits 1 and 2: Reserved Bit 3: FIFO filled to threshold (DataRdyInt) Bit 4: Output FIFO overflow (FifoOutOvrInt) Bit 5: Input FIFO overflow (FifoInOverInt) Bit 6: Sensor hub busy (DevBusy) Bit 7: Reserved If DataRdyInt is set, proceed to the next step.	AB 00 08
	2.2	AA 12 00	Get the number of samples (nn) in the FIFO.	AB 00 nn
	2.3	AA 12 01	Read the data stored in the FIFO; nn samples will be included. The format of the samples report is shown in **Table 4.**	AB 00 data_for_ nn_samples
STOP	Host ends the procedure:			
	3.1	AA 52 07 00 (CMD_DELAY = 120ms)	Disable the algorithm.	AB 00

*The host is required to periodically check the sensor hub for an available samples report. The default samples report period is 40ms which means sample rate is 25Hz. Depending on the output mode, the samples report may include algorithm and/or sensor data (see **section 1.2**)

The table below shows the format of received samples report when in SpO2 Calibration Data Collection Mode. R values are needed for the SpO_2 calibration process, as described in the **Guidelines for SpO_2 Measurement Using the Maxim MAX32664 Sensor Hub** application note.

Table 4. Samples Report from Output FIFO: SpO_2 Calibration Data Collection Mode Report

DATA SOURCE	DATA ITEM	# OF BYTES (MSB FIRST)	DESCRIPTION
MAX86141/40, MAXM86161 PPG Data (18 Bytes)* MAXM86146 PPG Data is 36 Bytes as described in Table 7a MAXM86174 PPG Data is 12 Bytes as described in Table 7b	PPG1 (PD1)	3	N/A
	PPG2 (PD1)	3	IR LED counts
	PPG3 (PD1)	3	Red LED counts
	PPG4 (PD2)	3	N/A
	PPG5 (PD2)	3	N/A
	PPG6 (PD2)	3	N/A
Accelerometer (6 Bytes)*	accelX	2	Two's complement. lsb = 0.001g
	accelY	2	Two's complement. lsb = 0.001g
	accelZ	2	Two's complement. lsb = 0.001g
Wearable Suite Algorithm (24 Bytes) MAXM86174 versions report 20 bytes algo data in regular report mode. Last 4 bytes defined in this table are excluded.	Op mode	1	Current operation mode: 0: Continuous Heart rate Monitor (HRM) and Continuous SpO_2 1: Continuous HRM and One-Shot SpO_2 2: Continuous HRM 3: Sampled HRM (~15s) 4: Sampled HRM and One-Shot SpO_2 (~30s) 5: Activity Tracking 6: SpO_2 Calibration Data Collection
	HR	2	N/A
	HR confidence	1	N/A
	RR	2	N/A
	RR confidence	1	N/A
	Activity class	1	N/A
	R	2	1000x calculated R value
	SpO_2 confidence	1	SpO_2 confidence level in %, >40 is for consumer devices, >80,90 is for medical devices.
	SpO_2	2	N/A
	SpO_2 valid, % complete	1	N/A
	SpO_2 low signal quality flag	1	Shows the low quality of the PPG signal: 0: Good quality 1: Low quality
	SpO_2 motion flag	1	Shows excessive motion: 0: No motion 1: Excessive motion
	SpO_2 low PI flag	1	Shows the low perfusion index (PI) of the PPG signal: 0: Normal PI 1: Low PI
	SpO_2 unreliable R flag	1	Shows the reliability of R: 0: Reliable 1: Unreliable
	SpO_2 state	1	Reported status of the SpO_2 algorithm: 0: LED adjustment

			1: Computation 2: Success 3: Timeout
	Skin contact detector (SCD) state	1	Skin contact state (SCD updates provided in Op mode 0 to 4) 0: Undetected 1: Off skin 2: On some subject 3: On skin
	IBI offset**	1	Reported when IBI is calculated. Defines number of samples between current algo sample and previous algo sample where IBI is calculated (for Maxim Wellness library)
	Unreliable orientation flag**	1	Flag reporting not appropriate orientation of wrist for reliable SpO2 measurement (for Maxim Wellness library, sleep quality) 0: correct orientation 1: wrong orientation
	RESERVED**	2	Reserved for future use

*If the output mode includes the sensor. LED assignment example is for the default configuration.

**** IBI offset, Unreliable orientation flag abd RESERVED bytes are not define in MAX32664C-MAX86174 API**

3 Measuring SpO_2 and Heart Rate on Wrist—SpO_2 and WHRM

3.1 Raw Data Mode

For hardware testing purposes, the user may choose to start the sensor hub to collect raw PPG samples. In this case, the host configures the sensor hub to work in Raw Data mode (no algorithm) by enabling the accelerometer and the AFE. The table below lists the set of commands that are needed to work in this mode. In Raw Data mode, only raw PPG samples and accelerometer data are included in the received samples. (The PPG raw data is stripped of the tags that are in the AFE FIFO and the sensor hub outputs the data in the order specified in the samples report tables).

Table 5. Host Commands—Raw Data Mode

	#	HOST COMMAND (HEX)	COMMAND DESCRIPTION	RESPONSE (HEX)
START	Host initializes the sensor hub in Raw Data mode using following commands:			
	1.1	AA 10 00 01	Set the output FIFO mode to Sensor data only.	AB 00
	1.2	AA 10 01 01	Set the sensor hub interrupt threshold.	AB 00
	1.3	AA 44 04 01 00 (if sensor hub accelerometer is used, CMD_DELAY = 20ms) AA 44 04 01 01 (if host accelerometer is used, CMD_DELAY = 20ms)	Enable the accelerometer.	AB 00
	1.4	AA 44 00 01 00 (CMD_DELAY = 250ms)	Enable AFE (MAX86141/40, MAXM86146, MAXM86161 or MAX86714**).*	AB 00
	Any command to change the sensor registers should appear AFTER enabling the sensor or they will be overwritten. By default, the algorithm sets the following AFE registers: Sample rate: 100Hz, 1-sample averaging (samples report period: 10 msec) Integration time: 117µs ADCs 1 and 2 range: 32µA LEDs 1, 2, and 3 full range: 124mA			
	1.5	AA 40 00 12 18 or AA 40 00 12 20 For MAX86174: AA 40 00 1A 01 AA 40 00 1B 47	Set the sample rate of the MAX86141 to 100Hz with 1-sample averaging by modifying register 0x12 of the AFE or 200 Hz, 1 sample averaging (3x.13.x+) Set the sample rate of the MAX86174 to 100Hz with 1-sample averaging by modifying registers 0x1A and 0x1B of the AFE	AB 00
	1.6	AA 40 00 23 7F For MAX86174: AA 40 00 25 3F AA 40 00 26 3F	Set the MAX86141 LED1 current to half of full scale. Reduce [7F] if the signal is saturated. Set the MAX86174 LED1 current to half of full scale. Reduce [3F]* if the signal is saturated. DRVA and DRVB simultaneously so set ¼ th of current range for both drives	AB 00

	1.7	AA 40 00 24 7F For MAX86174: AA 40 00 2D 3F AA 40 00 2E 3F	Set the MAX86141 LED2 current to half of full scale. Reduce [7F] if the signal is saturated. Set the MAX86174 LED2 current to half of full scale. Reduce [3F]* if the signal is saturated. DRVA and DRVB simultaneously so set ¼ th of current range for both drives	AB 00
	1.8	AA 40 00 25 7F For MAX86174: AA 40 00 35 3F AA 40 00 36 3F	Set the MAX86141 LED3 current to half of full scale. Reduce [7F] if the signal is saturated. Set the MAX86174 LED3 current to half of full scale. Reduce [3F]* if the signal is saturated. DRVA and DRVB simultaneously so set ¼ th of current range for both drives	AB 00
READING SAMPLES REPORT IN OUTPUT FIFO	Host reads samples periodically (do not execute at a faster rate than the samples report period):			
	2.1	AA 00 00	Read the sensor hub status byte: Bit 0: Sensor comm error Bits 1 and 2: Reserved Bit 3: FIFO filled to threshold (DataRdyInt) Bit 4: Output FIFO overflow (FifoOutOvrInt) Bit 5: Input FIFO overflow (FifoInOverInt) Bit 6: Sensor hub busy (DevBusy) Bit 7: Reserved If DataRdyInt is set, proceed to the next step.	AB 00 08
	2.2	AA 12 00	Get the number of samples (nn) in the FIFO.	AB 00 nn
	2.3	AA 12 01	Read the data stored in the FIFO; nn samples. The format of samples report is shown in **Table 6**.	AB 00 data_for_ nn_samples
STOP	Host ends the procedure:			
	3.1	AA 44 00** 00 (CMD_DELAY = 250ms)	Disable the AFE (e.g., the MAX86141).**	AB 00
	3.2	AA 44 04 00, (CMD_DELAY = 20ms)	Disable the accelerometer.	AB 00

* The host is required to periodically check the sensor hub for an available samples report. The default samples report period is 40ms which means sample rate is 25Hz

** Only one PPG sensor can be connected to MAX32664C so all PPG sensors have same ID in MAX32664C API.

Table 6. Samples Report from Output FIFO: Raw Data Mode Report

DATA SOURCE	DATA ITEM	# OF BYTES (MSB FIRST)	DESCRIPTION
MAX86141 PPG Data (18 Bytes)* MAXM86146 PPG Data is 36 bytes as described in Table 7a	PPG1 (PD1)	3	Green counts
	PPG2 (PD1)	3	IR counts
	PPG3 (PD1)	3	Red counts
	PPG4 (PD2)	3	Green2 counts (N/A if configured for MAX86140, MAXM86161, N/A means the data is not meaningful)
	PPG5 (PD2)	3	N/A
	PPG6 (PD2)	3	N/A
Accelerometer (6 Bytes)	accelX	2	Two's complement. lsb = 0.001g
	accelY	2	Two's complement. lsb = 0.001g
	accelZ	2	Two's complement. lsb = 0.001g

* LED assignment example is for the default configuration.

Table 7a. PPG Samples Report from Output FIFO: MAXM86146

DATA SOURCE	DATA ITEM	# OF BYTES (MSB FIRST)	DESCRIPTION
MAXM86146 PPG Data (36 Bytes)*	PPG1 (PD1)	3	Green counts
	PPG2 (PD1)	3	N/A
	PPG3 (PD1)	3	N/A
	PPG4 (PD1)	3	N/A
	PPG5 (PD1)	3	N/A
	PPG6 (PD1)	3	N/A
	PPG7 (PD2)	3	N/A
	PPG8 (PD2)	3	Red counts
	PPG9 (PD2)	3	IR counts
	PPG10 (PD2)	3	N/A
	PPG11 (PD2)	3	N/A
	PPG12 (PD2)	3	N/A

* LED assignment example is for the default configuration.

Table 8b. PPG Samples Report from Output FIFO: MAXM86174

DATA SOURCE	DATA ITEM	# OF BYTES (MSB FIRST)	DESCRIPTION
MAXM86174 PPG Data (12 Bytes)*	PPG1 (PD1)	3	Green counts
	PPG2 (PD2)	3	Green2 counts
	PPG3 (PD1)	3	IR counts
	PPG4 (PD1)	3	Red counts

* LED assignment example is for the default configuration.

3.2 AEC Mode

Automatic Exposure Control (AEC) is Maxim's gain control algorithm that is superior to AGC. The AEC algorithm optimally maintains the best SNR range and power optimization. The targeted SNR range is maintained regardless of skin color or ambient temperature within the limits of the LED currents configurations; The AEC dynamically manages the appropriate register settings for sampling rate, LED current, pulse width and integration time.

In the example below, both the AEC and SCD are enabled. The algorithm mode of operation can be selected as described in previous section. The sequence of commands is shown in the table below.

Table 9. Host Commands—AEC Mode

	#	HOST COMMAND (HEX)	COMMAND DESCRIPTION	RESPONSE (HEX)
START ALGORITHM	Host initializes the sensor hub in AEC-SCD mode using the following commands:			
	1.0	AA 02 00 (optional)	Read the device operating mode	AB 00 00 application mode
	1.1	AA FF 03 (optional)	Read the sensor hub version. XX: major version, YY: major version, XX: patch version.	AB 00 XX YY ZZ
	1.2	AA 50 07 00 [00000000FFD7FBD D00AB61FE]	Set SpO2 coefficients. Bracketed SpO_2 coefficients are for example only. Values used in the final form factor may be different per section 2.	AB 00

	1.3	AA 10 00 03	Set the output FIFO mode to sensor + algorithm data (streamed data will include PPG, accelerometer, and algorithm data).	AB 00
	1.4	AA 10 01 01	Set the sensor hub interrupt threshold.	AB 00
	1.5	AA 10 02 01	Set the samples report period to 40ms. Samples report rate to be one report per every sensor sample.*	AB 00
	1.6	AA 44 04 01 01 (if host accelerometer is used, CMD_DELAY = 20ms)	Enable the host-side accelerometer, if used.	AB 00
	1.7	AA 50 07 0A 00	Set the algorithm operation mode to Continuous HRM and Continuous SpO_2 or as desired. See **Table 12**.	AB 00
	1.8	AA 50 07 0B 01	Enable AEC [enabled by default].	AB 00
	1.9	AA 50 07 12 01	Enable Auto PD Current Calculation [enabled by default].	AB 00
	1.10	AA 50 07 0C 01	Enable SCD [enabled by default].	AB 00
	Optional: Any command to change the algorithm settings and configurations (**Table 12**) from default should appear here before enabling the algorithm.			
	1.11	AA FF 03 (optional)	Read the sensor hub version for [MAX86141/MAX86140/MAXM86146/MAXM86161/MAX86174] (30.x.y/30.x.y/33.x.y/32.x.y/35.x.y)	AB 00 [1E/1E/21/20] XX YY
	For 33.x.y (MAXM86146), configure HR inputs to use Green1 and Green2; configure SpO2 inputs to use IR, red. See "LEDs and PDs Configuration" for other example configurations.			
	1.20	For 33.x.y (MAXM86146) AA 50 07 19 13 56 00	Map configuration slot 1 to use LED1 (green1); map slot 2 to use LED3 (green2); map slot 3 to use LED5 (red); map slot 4 to use LED6 (IR)]	AB 00
	1.21	For 33.x.y (MAXM86146) AA 50 07 17 00 11	Map HR input 1 to use slot 1, PD1; map HR input 2 to use slot 2, PD2	AB 00
	1.22	For 33.x.y (MAXM86146) AA 50 07 18 30 20	Map SpO2 IR input to use slot 4 PD1; map SpO2 red input to use slot 3 PD1	AB 00
	1.3	AA 52 07 01 (for normal algorithm report, CMD_DELAY = 465ms) AA 52 07 02 (for extended algorithm report, CMD_DELAY = 465ms)	Enable the WHRM and SpO_2 algorithm. The format of the samples report is shown in **Table 9** (normal algorithm report) or **Table 10** (extended algorithm report). Note: If the accel is not connected to the sensor hub and if host accel data is not configured and provided, the algorithm will not produce the samples report - Accel data is required.	AB 00
	1.31	AA 41 00 FF (optional)	Read register FF (PART_ID) of [MAX86141/MAX86140//MAXM86161/MAX86174]	AB 00 [25/24/36]
	1.32	AA 41 04 0F (accel data is required from either host or KX122/KX11S/LIS2DS12I)	Read register 0F (WHO_AM_I) of [KX122/KX112/LIS2DS12] if connected to sensor hub	AB 00 [1B/22/43]
READING SAMPLES REPORT IN OUTPUT FIFO	Host reads samples periodically (do not execute at a faster rate than the samples report period):			
	2.1	AA 00 00	Read sensor hub status byte: Bit 0: Sensor comm error Bits 1 and 2: Reserved Bit 3: FIFO filled to threshold (DataRdyInt) Bit 4: Output FIFO overflow (FifoOutOvrInt) Bit 5: Input FIFO overflow (FifoInOverInt) Bit 6: Sensor hub busy (DevBusy) Bit 7: Reserved	AB 00 08

			If DataRdyInt is set, proceed to next step.	
	2.2	AA 12 00	Get the number of samples (nn) in the FIFO.	AB 00 nn
	2.3	AA 12 01	Read the data stored in the FIFO; nn samples will be read. The format of the samples report is shown in **Table 9** (normal algorithm report) or **Table 10 (**extended algorithm report).	AB 00 data_for_ nn_samples
STOP	Host ends the procedure:			
	3.1	AA 52 07 00 (CMD_DELAY = 120ms)	Disable the algorithm.	AB 00

* The host is required to periodically check the sensor hub for an available samples report. The default samples report period is 40ms which means sample rate is 25Hz. Depending on the output mode, the samples report may include algorithm and/or sensor data.

Table 10. Samples Report from Output FIFO: Normal Algorithm Report

DATA SOURCE	DATA ITEM	# OF BYTES (MSB FIRST)	DESCRIPTION
MAX86141/40, MAXM86161 PPG Data (18 Bytes)* MAXM86146 PPG Data is 36 Bytes as described in Table 7a MAXM86174 PPG Data is 12 Bytes as described in Table 7b	PPG1 (PD1)	3	Green counts
	PPG2 (PD1)	3	IR LED counts
	PPG3 (PD1)	3	Red LED counts
	PPG4 (PD2)	3	Green2 counts (N/A if configured for MAX86140, MAXM86161, N/A means the data is not meaningful)
	PPG5 (PD2)	3	N/A
	PPG6 (PD2)	3	N/A
Accelerometer (6 Bytes)*	accelX	2	Two's complement. lsb = 0.001g
	accelY	2	Two's complement. lsb = 0.001g
	accelZ	2	Two's complement. lsb = 0.001g
Wearable Suite Algorithm (24 Bytes) MAXM86174 versions report 20 bytes algo data in regular report mode. Last 4 bytes defined in this table are excluded.	Op mode	1	Current operation mode: 0: Continuous HRM and Continuous SpO_2 1: Continuous HRM and One-Shot SpO_2 2: Continuous HRM 3: Sampled HRM 4: Sampled HRM and One-Shot SpO_2 5: Activity tracking 6: SpO_2 Calibration Data Collection
	HR	2	10x calculated heart rate
	HR confidence	1	Confidence level in %, >40 is for consumer devices, >80,90 is for medical devices.
	RR	2	10x RR – inter-beat interval in ms Only shows a nonzero value when a new value is calculated.
	RR confidence	1	Calculated confidence level of RR in % Only shows a nonzero value when a new value is calculated.
	Activity class	1	Activity class (Applicable to wrist form factor only, MAX86141/0) 0: Rest

			1: Other 2: Walk 3: Run 4: Bike
	R	2	1000x calculated SpO_2 R value
	SpO_2 confidence	1	SpO_2 confidence level in %, >40 is for consumer devices, >80,90 is for medical devices.
	SpO_2	2	10x calculated SpO_2 %
	SpO_2 valid, % complete	1	Calculation progress in % in one-shot mode of algorithm. In continuous mode, it is reported as zero and only jumps to 100 when the SpO_2 value is updated. Bit[7]: SpO2 valid Bit[6..0]: Percent complete
	SpO_2 low signal quality flag	1	Shows the low quality of the PPG signal: 0: Good quality 1: Low quality
	SpO_2 motion flag	1	Shows excessive motion: 0: No motion 1: Excessive motion
	SpO_2 low PI flag	1	Shows the low perfusion index (PI) of the PPG signal: 0: Normal PI 1: Low PI
	SpO_2 unreliable R flag	1	Shows the reliability of R: 0: Reliable 1: Unreliable
	SpO_2 state	1	Reported status of the SpO_2 algorithm: 0: LED adjustment 1: Computation 2: Success 3: Timeout
	SCD state	1	Skin contact state: 0: Undetected 1: Off skin 2: On some subject 3: On skin
	IBI Offset**	1	Reported when IBI is calculated. Defines number of samples between current algo sample and previous algo sample where IBI is calculated (for Maxim Wellness library)
	Unreliable orientation flag**	1	Flag reporting not appropriate orientation of wrist for reliable SpO2 measurement (for Maxim Wellness library, sleep quality) 0: correct orientation 1: wrong orientation
	RESERVED**	2	Reserved for future use

*If the output mode includes the sensor. LED assignment example is for the default configuration.

**** IBI offset, Unreliable orientation flag abd RESERVED bytes are not define in MAX32664C-MAX86174 API**

Table 11. Samples Report from Output FIFO: Extended Algorithm Report

DATA SOURCE	DATA ITEM	# OF BYTES (MSB FIRST)	DESCRIPTION
MAX86141/40, MAXM86161 PPG Data (18 Bytes)* MAXM86146 PPG Data is 36 bytes as described in Table 7	PPG1 (PD1)	3	Green counts
	PPG2 (PD1)	3	IR LED counts
	PPG3 (PD1)	3	Red LED counts
	PPG4 (PD2)	3	Green2 counts (N/A if configured for MAX86140, MAXM86161, N/A means the data is not meaningful)
	PPG5 (PD2)	3	N/A
	PPG6 (PD2)	3	N/A
Accelerometer (6 Bytes)*	accelX	2	Two's complement. lsb = 0.001g
	accelY	2	Two's complement. lsb = 0.001g
	accelZ	2	Two's complement. lsb = 0.001g
Wearable Suite Algorithm (56 Bytes) MAXM86174 versions report 52 bytes algo data in extended mode. Last 4 bytes defined in this table are excluded.	Op mode	1	Current operation mode: 0: Continuous HRM and Continuous SpO_2 1: Continuous HRM and One-Shot SpO_2 2: Continuous HRM 3: Sampled HRM 4: Sampled HRM and One-Shot SpO_2 5: Activity Tracking 6: SpO_2 Calibration Data Collection
	HR	2	10x calculated heart rate
	HR confidence	1	Confidence level in %, >40 is for consumer devices, >80,90 is for medical devices.
	RR	2	10x RR – inter-beat interval in ms Only shows a nonzero value when a new value is calculated.
	RR confidence	1	Calculated confidence level of RR in % Only shows a nonzero value when a new value is calculated.
	Activity class	1	Activity class (Applicable to wrist form factor only, MAX86141/0) 0: Rest 1: Other 2: Walk 3: Run 4: Bike
	Total walk steps	4	Total number of walking steps since the last reset
	Total run steps	4	Total number of running steps since the last reset
	Total energy exp in kcal	4	10x total energy expenditure since the last reset in kcal
	Total AMR in kcal	4	10x total active energy expenditure since the last reset in kcal
	Is LED current adjustment requested in first time slot	1	Flag to notify if the LED current adjustment is requested or not in the first time slot

	Adjusted LED current in first time slot	2	10x value of the adjusted LED current (mA) in the first time slot, valid only if “Is LED current adjustment requested in first time slot” flag is true
	Is LED current adjustment requested in second time slot	1	Flag to notify if the LED current adjustment is requested or not in the second time slot
	Adjusted LED current in second time slot	2	10x value of the adjusted LED current (mA) in the second time slot, valid only if the “Is LED current adjustment requested in second time slot” flag is true
	Is LED current adjustment requested in third time slot	1	Flag to notify if the LED current adjustment is requested or not in the third time slot
	Adjusted LED current in third time slot	2	10x value of the adjusted LED current (mA) in third time slot, valid only if the “Is LED current adjustment requested in third time slot” flag is true
	Is integration time adjustment requested	1	Flag to notify if the integration time adjustment is requested or not
	Requested integration time	1	Value of the requested integration time option, valid only if the “Is integration time adjustment requested” flag is true
	Is sampling rate adjustment requested	1	Flag to notify if the sampling rate adjustment is requested or not
	Requested sampling rate	1	Value of the requested sampling rate option, valid only if the “Is sampling rate adjustment requested” flag is true
	Requested sampling average	1	Sampling average required for the requested sampling rate, valid only if the “Is sampling rate adjustment requested” flag is true
	WHRM AFE controller state for HRM channels	1	State of the AFE manager (for WHRM channels)
	Is high motion for HRM	1	Flag to notify if the motion is considered high for heart rate measurement
	SCD state	1	Skin contact state: 0: Undetected 1: Off skin 2: On some subject 3: On skin
	R	2	1000x calculated SpO_2 R value
	SpO_2 confidence	1	SpO_2 confidence level in %, >40 is for consumer devices, >80,90 is for medical devices.
	SpO_2	2	10x calculated SpO_2 %

	SpO_2 valid, % complete	1	Calculation progress in % in one-shot mode of algorithm. In continuous mode, it is reported as zero and only jumps to 100 when the SpO_2 value is updated. Bit[7]: SpO2 valid Bit[6..0]: Percent complete
	SpO_2 low signal quality flag	1	Shows the low quality of the PPG signal: 0: Good quality 1: Low quality
	SpO_2 motion flag	1	Shows excessive motion: 0: No motion 1: Excessive motion
	SpO_2 low PI flag	1	Shows the low perfusion index (PI) of the PPG signal: 0: Normal PI 1: Low PI
	SpO_2 unreliable R flag	1	Shows the reliability of R: 0: Reliable 1: Unreliable
	SpO_2 state	1	Reported status of the SpO_2 algorithm: 0: LED adjustment 1: Computation 2: Success 3: Timeout
	IBI Offset**	1	Reported when IBI is calculated. Defines number of samples between current algo sample and previous algo sample where IBI is calculated (for Maxim Wellness library)
	Unreliable orientation flag**	1	Flag reporting not appropriate orientation of wrist for reliable SpO2 measurement (for Maxim Wellness library, sleep quality) 0: correct orientation 1: wrong orientation
	RESERVED**	2	Reserved for future use

*If the output mode includes the sensors. LED assignment example is for the default configuration.

**** IBI offset, Unreliable orientation flag abd RESERVED bytes are not define in MAX32664C-MAX86174 API**

3.3 AGC Mode

In this mode, the wearable algorithm suite (SpO_2 and WHRM) is enabled and the R value, SpO_2, SpO_2 confidence level, heart rate, heart rate confidence level, RR value, and activity class are reported. Furthermore, automatic gain control (AGC) is enabled. Because AGC is a subset of AEC functionality, to enable AGC, AEC still needs to be enabled. However, automatic calculation of target PD should be turned off, and the desired level of AGC target PD current is set by the user. The user may change the algorithm to the desired configuration mode. If signal quality is

low, a LowSNR flag will be set. Excessive motion is also reported with a flag. The sequence of commands is shown in

The following algorithm operating modes may be selected::

0. **Continuous HRM + Continuous SpO_2:** Both heart rate and SpO2 values are continuously measured and updated.
1. **Continuous HRM + One-Shot SpO_2:** Heart rate is continuously monitored; SpO_2 is measured once.
2. **Continuous HRM:** Only the heart rate algorithm in continuous mode is enabled.
3. **Sampled HRM:** It measures heart rate once using the sampled HRM algorithm and then switches to activity mode.
4. **Sampled HRM + One-Shot SpO_2:** It measures heart rate and SpO_2, and then switches to activity mode.
5. **Activity Tracking ONLY:** Only shows accelerometer data. LEDs are off.
6. **SpO_2 Calibration Data Collection:** Provides data used for SpO_2 calibration data collection. Only red and IR LEDs are activated.

Table 12. Host Commands—AGC Mode

	#	HOST COMMAND (HEX)	COMMAND DESCRIPTION	RESPONSE (HEX)
START	Host initializes the sensor hub in AGC mode using the following commands:			
	1.0	AA 02 00 (optional)	Read the device operating mode	AB 00 00 application mode
	1.1	AA FF 03 (optional)	Read the sensor hub version. XX: major version, YY: major version, XX: patch version.	AB 00 XX YY ZZ
	1.2	AA 50 07 00 [00000000FFD7FBDD00AB61FE]	Bracketed SpO_2 coefficients are for example only. Values used in the final form factor may be different per section 2.	AB 00
	1.3	AA 10 00 03	Set the output FIFO mode to sensor + algorithm data (streamed data will include PPG, accelerometer, and algorithm data).	AB 00
	1.4	AA 10 01 01	Set the sensor hub interrupt threshold.	AB 00
	1.5	AA 10 02 01	Set the samples report period to 40ms. Samples report rate to be one report per every sensor sample.*	AB 00
	1.6	AA 44 04 01 01 (if host accelerometer is used, CMD_DELAY = 20ms)	Enable the host-side accelerometer, if used.	AB 00
	1.7	AA 50 07 0A 00	Set the algorithm operation mode to Continuous HRM and Continuous SpO_2. See **Table 12**.	AB 00
	1.8	AA 50 07 0B 01	Enable AEC [enabled by default].	AB 00
	1.9	AA 50 07 12 00	Disable Auto PD Current Calculation.**	AB 00
	1.10	AA 50 07 0C 00	Disable SCD.	AB 00
	1.11	AA 50 07 11 00 64	Set AGC Target PD Current to 10µA.	AB 00
	Optional: Any command to change the algorithm settings and configurations from the default should appear here before enabling the algorithm.			
	1.12	AA FF 03 (optional)	Read the sensor hub version for [MAX86141/MAX86140/MAXM86146/MAXM86161/MAX86174] (30.x.y/30.x.y/33.x.y/32.x.y/35.x.y)	AB 00 [1E/1E/21/20] XX YY
	For 33.x.y (MAXM86146), configure HR inputs to use Green1 and Green2; configure SpO2 inputs to use IR, red. See “LEDs and PDs Configuration” for other example configurations.			
	1.20	For 33.x.y (MAXM86146) AA 50 07 19 13 56 00	Map configuration slot 1 to use LED1 (green1); map slot 2 to use LED3 (green2); map slot 3 to use LED5 (red); map slot 4 to use LED6 (IR)]	AB 00
	1.21	For 33.x.y (MAXM86146) AA 50 07 17 00 11	Map HR input 1 to use slot 1, PD1; map HR input 2 to use slot 2, PD2	AB 00
	1.22	For 33.x.y (MAXM86146) AA 50 07 18 30 20	Map SpO2 IR input to use slot 4 PD1; map SpO2 red input to use slot 3 PD1	AB 00
	1.3	AA 52 07 01 (normal algorithm report, CMD_DELAY = 465ms) AA 52 07 02 (extended algorithm report, CMD_DELAY = 465ms)	Enable WHRM and SpO_2 algorithm. The format of samples report is shown in **Table 9** (normal algorithm report) or **Table 10 (e**xtended algorithm report). Note: If the accel is not connected to the sensor hub and if host accel data is not configured and provided, the algorithm will not produce the samples report - Accel data is required.	AB 00

	1.31	AA 41 00 FF (optional)	Read register FF (PART_ID) of [MAX86141/MAX86140//MAXM86161]	AB 00 [25/24/36]
	1.32	AA 41 04 0F (accel data is required from either host or KX122/KX11S/LIS2 DS12I)	Read register 0F (WHO_AM_I) of [KX122/KX112/LIS2DS12] if connected to sensor hub	AB 00 [1B/22/43]
READING SAMPLES IN OUTPUT FIFO	Host reads samples periodically (do not execute at a faster rate than the samples report period):			
	2.1	AA 00 00	Read the sensor hub status byte: Bit 0: Sensor comm error Bits 1 and 2: Reserved Bit 3: FIFO filled to threshold (DataRdyInt) Bit 4: Output FIFO overflow (FifoOutOvrInt) Bit 5: Input FIFO overflow (FifoInOverInt) Bit 6: Sensor hub busy (DevBusy) Bit 7: Reserved If DataRdyInt is set, proceed to the next step.	AB 00 08
	2.2	AA 12 00	Get the number of samples (nn) in the FIFO.	AB 00 nn
	2.3	AA 12 01	Read the data stored in the FIFO; nn samples will be read. The format of the samples report is shown in **Table 9** (normal algorithm report) or **Table 10** (extended algorithm report).	AB 00 data_for_ nn_samples
STOP	Host ends the procedure:			
	3.1	AA 52 07 00 (CMD_DELAY = 120ms)	Disable the algorithm.	AB 00

*The host is required to periodically check the sensor hub for an available samples report. The default samples report period is 40ms which means sample rate is 25Hz. Depending on the output mode, the samples report may include algorithm and/or sensor data (see **section 1.2**)
**After disabling the Auto PD Current Calculation, the algorithm will use the value in step 1.10 to adjust AGC.

3.4 Power Saving Mode: Reduced Samples Rate, Sampled HRM

This mode is like the previously described mode where the algorithm, AEC and SCD are enabled (by default). The only differences are to change the following in **Table 8**:

- Change the output mode in step 1.3 **Table 8** to Algorithm Only (0x10 0x00 0x02).
- Change the samples report period in step 1.5 **Table 8** to 25 (0x10 0x02 0x19) or more as shown in **Table 8**. For example, set the *samples report period* to 25 which is 1s (25*40ms).
- Host reads the samples in output FIFO at a period of five times the samples report period. For example, host reads output FIFO every 5 seconds (*host reading FIFO period*)
- Choose the Sample HRM or Sample HRM + One-Shot SpO_2 algorithm operation mode in step 1.7 **Table 8,** (0x50 0x07 0x0A 0x03 or 0x04). to The Sampled HRM mode saves more power as it automatically switches to Activity Tracking mode once the heart rate is measured (~15s). In this case, the host may choose to reconfigure the operation mode (e.g., in case of motion).
- Enable the WHRM and SpO_2 algorithm in step 1.14 in normal report mode (0x52 0x07 0x01).

This configuration helps the sensor hub to wake up less often, and I²C communication time is minimized. The report detailed in **Table 9** will only include algorithm data. For the MAXREFDES103, a fully charged battery will last about 8 hours in this mode. It is believed that

the battery of the MAXREFDES103 may be updated to a 200 mAH of size 401530 and still have some clearance for expansion. Alternatively, the host may also run the algorithm in the sample one-shot mode say every 5 minutes to conserve power.

- **401530 200maH 3.7V Lithium Ion Polymer Battery, Alibaba**

Note: This mode is not appropriate for monitoring inter-beat interval (RR) value. RR and RR Confidence are reported whenever a new value is calculated by the algorithm and shown as zero for the rest of the time. Therefore, the last reported value may be missed if the samples report rate is not set to 1.

3.5 SCD Only Mode

In this mode, SCD is enabled and only SCD state is reported in the algorithm samples report. Before enabling SCD Only mode, the host should specify which LED is to be used by the SCD algorithm. The sequence of commands is shown below.

Table 13. Host Commands: SCD Mode

	#	HOST COMMAND (HEX)	COMMAND DESCRIPTION	RESPONSE (HEX)
START ALGORITHM	Host initializes the sensor hub in SCD mode using the following commands:			
	1.1	AA E5 LED_IDX	Sets which LED index is to be used by the SCD algorithm. (Colors are for MAXREFDES103) LED_IDX = 0 for LED1 (Green) = 1 for LED2 (IR) = 2 for LED3 (Red)	AB 00
	1.2	AA 10 00 02	Set output mode to algorithm data (SCD Only when used with 1.7).	AB 00
	1.3	AA 10 01 01	Set sensor hub interrupt threshold.	AB 00
	1.4	AA 10 02 01	Set the samples report period to 40ms. Samples report rate to be one report per every sensor sample.	AB 00
	1.5	AA 44 00 01 00	Enable AFE (e.g., MAX86141) with sensor hub samples.	AB 00
	1.6	AA 44 04 01 00 (if sensor hub accelerometer is used, CMD_DELAY = 20ms) AA 44 04 01 01 (if host accelerometer is used, CMD_DELAY = 20ms)	Enable accelerometer with sensor hub or host-side accelerometer.	AB 00
	1.7	AA 52 07 03 (CMD_DELAY = 465ms)	Enable SCD Only algorithm. The format of samples report is shown in **Table 13**	AB 00
READING SAMPLES REPORT IN OUTPUT	Host reads samples periodically (do not execute at a faster rate than the samples report period):			
	2.1	AA 00 00	Read sensor hub status byte: Bit 0: Sensor comm error Bits 1 and 2: Reserved Bit 3: FIFO filled to threshold (DataRdyInt) Bit 4: Output FIFO overflow (FifoOutOvrInt) Bit 5: Input FIFO overflow (FifoInOverInt) Bit 6: Sensor hub busy (DevBusy) Bit 7: SCDSM detected skin contact If DataRdyInt is set, proceed to next step.	AB 00 08
	2.2	AA 12 00	Get the number of samples (nn) in the FIFO.	AB 00 nn

	2.3	AA 12 01	Read the data stored in the FIFO; nn samples will be read. The format of samples report is shown in **Table 13.**	AB 00 data_for_ nn_samples
STOP	Host ends the procedure:			
	3.1	AA 44 00 00 (CMD_DELAY = 250ms)	Disable AFE (e.g., MAX86141).	AB 00
	3.2	AA 44 04 00 (CMD_DELAY = 20ms)	Disable accelerometer.	AB 00
	3.3	AA 52 07 00 (CMD_DELAY = 120ms)	Disable algorithm.	AB 00

Table 14. Samples Report from Output FIFO: SCD Only Mode

DATA SOURCE	BYTE INDEX	DATA ITEM	NUMBER OF BYTES (MSB FIRST)	DESCRIPTION
SCD Algorithm	0	SCD Classifier output	1	0 = Unidentified 1 = Off Skin 2 = On Object 3 = On Skin

4 Configurations and Settings

The settings shown in **Table 14** are available for the wearable suite (SpO_2 and WHRM) algorithm. To update the algorithm settings, make sure to send the appropriate commands BEFORE enabling the algorithm.

4.1 LEDs and PDs Configuration

The firmware is shipped with the default LEDs and PDs configuration suited for the reference design. Alternatively, the user may change the configuration if other combinations of LEDs and PDs provide superior performance according to the hardware and optomechanical design. The following 3 steps (commands elaborated in **Table 14)** are required to change LEDs and PDs configuration:

A. *Map LEDs to configuration slots and firing order (Configuration index = 0x19):* The user needs to select which LED (or LEDs) are fired in each slot. The firing starts from LEDs selected in slot 1 and continues to slot 6. For example, if LED 1 is Green, LED 2 is IR and LED 3 is red, the firing sequence 0x123000 configures the firmware to fire them in that order. PPGs are also reported in the same order.

B. *Map WHRM two input channels to slots and PDs (Configuration index = 0x17):* The user should configure which PD in which firing slot (as defined in A) is used as each of the two inputs of the algorithm. For example, 0x0001 means PD1 and PD2 of the LED fired in slot 1 (Green as configured in A) are selected for WHRM algorithm inputs.

C. *Map SpO_2 IR and Red input channels to slots and PDs (Configuration index = 0x18):* The user should configure which PD in which firing slot (as defined in A) is used as two inputs of the algorithm. For example, 0x1020 means PD1 of the LEDs fired in slot 2 (IR as configured in A) and slot 3 (Red as configured in A) are selected for IR and Red inputs of SpO_2 algorithm.

Examples:

1) Configuration settings for two PDs, slot 1 green, slot 2 IR, slot 3 red (default for **MAXREFDES103(MAX86141)**)
   0x19: 0x123000 [slot 1 use LED1 (green); slot 2 use LED2 (IR); slot 3 use LED3 (red)]
   0x17: 0x0001 [HR input 1 uses slot 1, PD1; HR input 2 uses slot 1, PD2]
   0x18: 0x1020 [SpO_2 IR input uses slot 2 PD1; SpO_2 red input uses slot 3 PD1]
2) Configuration settings for two PDs, slot 1 green, slot 2 Red, slot 3 IR (Host configures sensor hub for **Maslak D6W space optimized schematic, MAX86141**)
   0x19: 0x123000 [slot 1 use LED1 (green);slot 2 use LED2 (Red); slot 3 use LED3 (IR)]
   0x17: 0x0001 [HR input 1 uses slot 1, PD1; HR input 2 uses slot 1, PD2]
   0x18: 0x2010 [SpO_2 IR input uses slot 3 PD1; SpO_2 red input uses slot 2 PD1]
3) Configuration settings for one PD, slot 1 green, slot 2 IR, slot 3 red (default for **MAX86140, MAXM86161EVSYS**).
   0x19: 0x123000 [slot 1 use LED1 (green); slot 2 use LED2 (IR); slot 3 use LED3 (red)]
   (0x17: 0x0073) default: [HR Input 1 uses Slot 1 (green), PD1; HR Input 2 not used]
   0x18: 0x1020 [SpO_2 IR input uses slot 2 PD1; SpO_2 red input uses slot 3 PD1]
4) Configuration settings for LED1 green and LED2 green fired simultaneously using slot 1; LED4 and LED5 are IR and Red and are fired using slot 2 and 3; WHRM uses PD1 and PD2 when both LED1, LED2 are fired; $WSpO_2$ uses PD2 of LED4, LED5:
   0x19: 0x745000 [slot 1 uses LED1&2 (green) together; slot 2 uses LED4 (IR); slot 3 uses LED5 (red)]
   0x17: 0x0001 [HR input 1 uses IR slot 1, PD1; HR input 2 uses slot 1, PD2]
   0x18: 0x1121 [SpO_2 IR input uses slot 2 PD2; SpO_2 red input uses slot 3 PD2]
5) Configuration for one PD, no Green LED1; slot 2 IR, slot 3 red. (Host configures **MAXREFDES103** to the settings below)
   0x19: 0x230000 [slot 1 use LED2 (IR); slot 2 use LED3 (red)]
   0x17: 0x0073 [HR input 1 uses IR slot 1, PD1; HR Input 2 not used]
   0x18: 0x0010 [SpO_2 IR input uses slot 1 PD1; SpO_2 red input uses slot 2 PD1]
6) Configuration settings for two PDs, slot 1 green2, slot 2 IR, slot 3 red (Host configures **MAXREFDES103** to the settings below)
   0x19: 0x423000 [slot 2 use LED4 (green2); slot 2 use LED2 (IR); slot 3 use LED3 (red)]
   0x17: 0x0001 [HR input 1 uses slot 1, PD1; HR input 2 uses slot 1, PD2]
   0x18: 0x1020 [SpO_2 IR input uses slot 2 PD1; SpO_2 red input uses slot 3 PD1]
7) **MAX86146EVSYS** configuration settings for two PDs, slot 1 green1, slot 2 green2, slot 3 red, slot 4 IR.
   0x19: 0x135600 [slot 1 use LED1 (green1); slot 2 use LED3 (green2); slot 3 use LED5 (red); slot 4 LED6 (IR)]
   0x17: 0x0011 [HR input 1 uses slot 1, PD1; HR input 2 uses slot 2, PD2]
   0x18: 0x3020 [SpO_2 IR input uses slot 4 PD1; SpO_2 red input uses slot 3 PD1]
8) **MAX86146EVSYS** configuration settings for two PDs, slot 1 green1, slot 2 red, slot 3 IR.
   0x19: 0x156000 [slot 1 use LED1 (green1); slot 2 use LED5 (red); slot 3 LED6 (IR)]
   0x17: 0x0001 [HR input 1 uses slot 1, PD1; HR input 2 uses slot 1, PD2]
   0x18: 0x2010 [SpO_2 IR input uses slot 3 PD1; SpO_2 red input uses slot 2 PD1]
9) **MAXREFDES103(MAX86141)** Configuration settings for HR only two PDs, slot 1 green
   0x19: 0x100000 [slot 1 use LED1 (green)]
   0x17: 0x0001 [HR input 1 uses slot 1, PD1; HR input 2 uses slot 1, PD2]
   0x18: 0x7373 [SpO_2 IR not used; SpO_2 red input not used]

Set the algorithm operating mode to continuous HR: 0x50 0x50 0x07 0x0A 0x02

10) **MAX86146EVSYS** configuration settings for HR only: two PDs, slot 1 green1, slot 2 green2
    0x19: 0x130000 [slot 1 use LED1 (green1); slot 2 use LED3 (green2);
    0x17: 0x0011 [HR input 1 uses slot 1, PD1; HR input 2 uses slot 2, PD2]
    0x18: 0x7373 [SpO_2 IR not used; SpO_2 red input not used]
    Set the algorithm operating mode to continuous HR: 0x50 0x50 0x07 0x0A 0x02
11) Configuration settings for two PDs, slot 1 green, slot 2 IR, slot 3 red (default for **MAXREFDES105(MAX86174)**)
    0x19: 0x123000 [slot 1 use LED1 (green); slot 2 use LED2 (IR); slot 3 use LED3 (red)]
    0x17: 0x0001 [HR input 1 uses slot 1, PD1; HR input 2 uses slot 1, PD2]
    0x18: 0x1020 [SpO_2 IR input uses slot 2 PD1; SpO_2 red input uses slot 3 PD1]

Modifying the LED, PD configurations will affect which LED counts will in the description column of the PPG raw data in samples report.

Table 15. Sensor Hub Commands

FAMILY BYTE	INDEX BYTE	WRITE BYTE(S)	DESCRIPTION	RESPONSE BYTES
0x00 read	0x00		Read sensor hub status	**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 = Not busy; 1 = sensor busy, try again in 1ms, up to five times. **SCDMonSkin[7]:** 0 = Skin not detected. 1 = Skin detected.
0x01 write	0x00	**0x00**: Exit bootloader mode, enter application mode	Set the device operating mode.	

		(CMD_DELAY = 1.5s) **0x01**: Shutdown the MAX32664. Restart by power cycling or pulsing RSTN. **0x08**: Invalidate Application and Enter bootloader mode. CMD_DELAY 4.25s (this command uses two of the 10,000 write cycles). Reset the MAX32664 after the 4.25s by setting RSTN low for 10ms and waiting 50ms (33.13.19+, NA for 32.x.x)		
0x02 read	0x00		Read the device operating mode.	0x00: Application operating mode. 0x02: Reset. 0x08: Bootloader operating mode.
0x10 write	0x00	**0x00**: (no data) **0x01**: Sensor Data **0x02**: Algorithm Data **0x03**: Sensor Data and Algorithm Data **0x04**: (no data) **0x05**: Sample Counter byte, Sensor Data **0x06**: Sample Counter byte, Algorithm Data **0x07**: Sample Counter byte, Sensor Data and Algorithm Data	Set the output format of the sensor hub.	
0x11 read	0x00		Read the output format of the sensor hub.	**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
0x10 write	0x01	0x01 to 0xFF: Sensor Hub Interrupt Threshold for FIFO.	Set the threshold for the FIFO interrupt bit. The bit DataRdyInt of the sensor hub status byte is set when this threshold is reached.	
0x11 read	0x01	-	Read the threshold for the FIFO interrupt bit. The bit DataRdyInt of the sensor hub status byte is set when this threshold is reached.	**0x01 to 0xFF**: Sensor Hub Interrupt Threshold for FIFO.
0x11 read	0x02	-	Read the samples report period (e.g., a value of 25 means a samples report is generated once every 25 samples).	**0x01 (default) to 0xFE:** lsb is 40ms. N, where a samples report is generated once every N samples.
0x10 write	0x02	**0x01 to 0xFE:** lsb is 40ms. N, where a samples report is generated once every N samples.	Set the samples report period (e.g., a value of 25 means a samples report is generated once every 25 samples).	-
0x10 write	0x03	**0x02 to 0xFF**: New I^2C address (8-bit I^2C write address)	Change I^2C address of the MAX32664.	
0x10 write	0x04	**0x00 to 0xFF**: Counter	Set the sensor hub counter.	
0x11 read	0x04		Read the sensor hub counter.	0x00 to 0xFF: Counter
0x11 read	0x05		Read PPG output FIFO samples report size. (32.9.21+, 33.13.19+)	Number of bytes in the PPG samples report
0x11 read	0x06	**0x01**: Normal Algorithm Report **0x02**: Extended Algorithm Report **0x03**: SCD Only Algorithm Report (NA 32.13.2x)	Read algorithm output FIFO samples report size. (33.13.19+, 32.9.23+)	Number of bytes in the algorithm samples report
0x12 read	0x00		Read the number of samples available in the output FIFO	Number of samples available in the FIFO.
0x12 read	0x01		Read data stored in output FIFO.	Samples Report from Output FIFO. The internal FIFO read pointer increments once the sample size bytes have been read.

				See Samples Report Table for more details
0x13 read	0x01		Read the input FIFO size for the maximum number of samples that the input FIFO can hold.	Input FIFO size (16-bit unsigned)
0x14 write	0x00	N*[LSBx MSBx, LSBy MSBy, LSBy MSBz] N sets of data are sent, where N is the 40 msec samples report period.	Write data to the input FIFO. If the host accel has been enabled (0x44 0x04 0x01 0x01), then this command may be used supply host accel data to the sensor hub. Example, send five sets of host accel [LSBx MSBx, LSBy MSBy, LSBy MSBz] data to the sensor hub: 14 00 84 FE 1C 00 55 FC 85 FE 1B 00 57 FC 84 FE 1A 00 56 FC 85 FE 19 00 54 FC 87 FE 1B 00 57 FC -380=FE84, 28=001C, -939=FC55; -379=85FE, 27=001B, -938=FC56; -380=FE84, , 26=001A, -938=FC56; -379=85FE, 25=0019, -940=FC54; -377=87FE, 27=001B, -937=FC57	
0x40 write	0x00	[reg addr] [reg value]	Write a value to a writable MAX86140/MAX86141/MAXM86161 register.	
0x41 read	0x00	[reg addr]	Read the value of a MAX86140/MAX86141/MAXM86161 register.	Register value (byte)
0x40 write	0x04	[reg addr] [reg value]	Write a value to a writable accelerometer sensor register.	
0x41 read	0x04	[reg addr]	Read accelerometer sensor register.	Register value (byte)
0x42 read	0x00		Read the attributes of the MAX86140/MAX86141/MAXM86146/MAXM86161 AFE.	Number of bytes in a word for this sensor, Number of registers available for this sensor.
0x42 read	0x04		Read the attributes of the accelerometer.	Number of bytes in a word for this sensor, Number of registers available for this sensor.
0x43 read	0x00		Read all the MAX86140/ MAX86141/ MAXM86161 registers.	Register address 0, register value 0, register address 1, register value 1, …, register address n, register value n.
0x43 read	0x04		Read all the accelerometer registers.	Register address 0, register value 0, register address 1, register value 1, …, register address n, register value n.
0x44 write	0x00	**0x00**: Disable **0x01**: Enable	Write the enable/disable the MAX86140/MAX86141/MAXM86146/ MAXM86161 sensor.	

		CMD_DELAY = 250ms		
0x45 read	0x00		Read the enable/disable of the MAX86140/MAX86141/ MAXM86146/MAXM86161	**0x00**: Disable **0x01**: Enable
0x44 write	0x04	**0x00, 0x00:** Disable sensor hub accelerometer **0x00, 0x01:** Disable external host accelerometer **0x01, 0x00:** Enable sensor hub accelerometer **0x01, 0x01:** Enable external host accelerometer CMD_DELAY = 20ms	Write the enable/disable, sensor hub accel/external accel sensor.	
0x45 read	0x04		Read the enable/disable, sensor hub accel/external accel sensor.	**0x00**, **0x00:** Sensor hub accelerometer disabled **0x00**, **0x01:** External host accelerometer disabled **0x01**, **0x00:** Sensor hub accelerometer enabled **0x01**, **0x01:** External host accelerometer enabled
0x44 write	0xFF	0x44	Single command to enable multiple sensors. CMD_DELAY = 465ms for AFE+accel sensor. Example: Enable [MAX86140/MAX86141/MAXM86146/MAXM86161] and accel. 44 FF 02 04 00 00 00 00 00 Exceptions: 1. If any sensor in the list is already enabled, it turns off and enables again. 2. If enabling one of the sensors in the list fails, the sensor hub disables all the sensors in the command list. 3. All sensors in this command list must be valid available hardware, otherwise, the sensor hub disables all the sensors listed in this command.	**N, SI, SM, SE, … SI, SM, SE:** Enable multiple sensors, where: • N is the number of sensors • SI is the sensor index • SM is the sensor mode • SE is 1 if the sensor is an external host or 0 if the sensor is connected to the sensor hub Sensor indices are defined as: **0x00:** MAX86140/ MAX86141/ MAXM86146/ MAXM86161

				0x01: MAX30205 **0x02:** MAX30001 **0x03:** MAX30101/MAX30102 **0x04:** Accelerometer Sensor modes are defined in the first byte of Write Bytes field of the Sensor Mode Enable commands, 0x44 0x00 to 0x44 0x04
0x46 write	0x04	00 [byte1] [byte2] [byte3]	Write enable/disable wake up on motion detection (3-byte value): byte1: **0x00:** Disable **0x01:** Enable byte2: **0x01 to 0xFE:** Wake up filter period (seconds). Motion must be present during this period time before a wake-up is generated. **0xFF:** Disable byte3: **0x01 to 0x80** lsb = 0.0625g (1/16g. For example, 0x08 is 0.5g. *As defined in the KX122/LIS2DS12 data sheet. **To disable wake up on motion, use 0x00FFFF.**	
0x50 for write	0x07	0x00 [A_MSB .. .. A_LSB] [B_MSB .. .. B_LSB] [C_MSB .. .. C_LSB]	Write SpO_2 calibration coefficients (12 bytes comprised of three 32-bit signed values (A, B, C), scaled up by 100,000).	
0x51 for read	0x07	0x00	Read SpO_2 calibration coefficients (12 bytes comprised of three 32-bit signed values, scaled up by 100,000). Default: 0x00000000 FFD7FBDD 00AB61FE A = 0 (0x00000000) B = -26.224999 (0xFFD7FBDD) C = 112.317421 (0x00AB61FE	32-bit signed integer A, 32-bit signed integer B, 32-bit signed integer C Values scaled up by 100,000
0x50 for write	0x07	0x01 [MSB LSB]	Write SpO_2 motion-detection period (unsigned 16-bit int, seconds). The algorithm will consider the state to be motionless if the motion is below the threshold for this duration of time.	

0x51 for read	0x07	0x01	Read SpO_2 motion-detection period. The algorithm will consider the state to be motionless if the motion is below the threshold for this duration of time. Default: 0x0002	MSB of period, LSB of period (16-bit unsigned integer**,** seconds)
0x50 for write	0x07	0x02 [MSB .. .. LSB]	Write SpO_2 motion-detection threshold (signed 32-bit int, equal to 10^5 x milli-g threshold value).	
0x51 for read	0x07	0x02	Read SpO_2 motion-detection threshold (signed 32-bit int [MSB .. .. LSB], equal to 10^5 x milli-g threshold value). Default: 0x01C9C380 (0.3g)	4 bytes (32-bit signed integers which are the milli-g motion threshold times 100,000)
0x50 for write	0x07	0x03 [byte]	Write SpO2 AGC Timeout (sec) .	
0x51 for read	0x07	0x03	Read SpO2 AGC Timeout (sec) . Default: 0x3C	SpO2 AGC Timeout (sec, 8-bit unsigned)
0x50 for write	0x07	0x04 [byte]	Write the timeout duration for SpO2 measurement in seconds (1 byte).	
0x51 for read	0x07	0x04	Read the timeout duration for SpO2 measurement in seconds. Default: 0x5A	SpO_2 algorithm timeout (sec, 8-bit unsigned)
0x50 for write	0x07	0x05 [byte]	Write initial HR algorithm value (8-bit unsigned).	
0x51 for read	0x07	0x05	Read initial HR algorithm value. Default: 0x3C	Initial heart rate setting (8-bit unsigned)
0x50 for write	0x07	0x06 [MSB] [LSB]	Write height (16-bit unsigned, cm).	
0x51 for read	0x07	0x06	Read height. Default: 0x00AF	Height (16-bit unsigned, cm)
0x50 for write	0x07	0x07 [MSB] [LSB]	Write weight (16-bit unsigned, kg).	
0x51 for read	0x07	0x07	Read weight. Default: 0x004E	Weight (16-bit unsigned, kg)
0x50 for write	0x07	0x08 [byte]	Write age (8-bit unsigned, years).	
0x51 for read	0x07	0x08	Read age. Default: 0x1E	Age (8-bit unsigned, years)
0x50 for write	0x07	0x09 [byte]	Write gender. byte: 0x00: Male 0x01: Female	
0x51 for read	0x07	0x09	Read gender	Gender **0x00**: Male (default) **0x01**: Female
0x50 for write	0x07	0x0A [algo operation mode byte]	Set the algorithm operation mode (can be switched in runtime): **0x00**: Continuous HRM + Continuous SpO_2 **0x01**: Continuous HRM + One-Shot SpO_2 **0x02**: Continuous HRM	-

			0x03: Sampled HRM **0x04**: Sampled HRM + One-Shot SpO_2 **0x05**: Activity Tracking ONLY **0x06**: SpO_2 Calibration Data Collection **0x07**: Continuous HRM ,Fast SpO_2	
0x51 for read	0x07	0x0A	Read the algorithm operation mode.	**0x00**: Continuous HRM, continuous SpO_2 (default) **0x01**: Continuous HRM, one-shot SpO_2 **0x02**: Continuous HRM **0x03**: Sampled HRM **0x04**: Sampled HRM, one-shot SpO_2 **0x05**: Activity tracking only **0x06**: SpO_2 calibration **0x07**: Continuous HRM ,Fast SpO_2
0x50 for write	0x07	0x0B [byte]	Write the enable/disable AEC byte: **0x00**: Disable **0x01**: Enable	
0x51 for read	0x07	0x0B	Read the enable/disable AEC.	**0x00**: Disabled **0x01**: Enabled (default)
0x50 for write	0x07	0x0C [byte]	Write the enable/disable Skin Contact Detection (SCD) algorithm. byte: **0x00**: Disable **0x01**: Enable	
0x51 for read	0x07	0x0C	Read the enable/disable Skin Contact Detection (SCD) algorithm.	**0x00**: Disabled **0x01**: Enabled (default)
0x50 for write	0x07	0x0D [MSB] [LSB]	Write adjusted target PD current period (16-bit unsigned, seconds)	
0x51 for read	0x07	0x0D	Read adjusted target PD current period in seconds. Default: 0x0708	Adjusted target PD current period (16-bit unsigned, seconds)
0x50 for write	0x07	0x0E [MSB] [LSB]	Write HR motion magnitude threshold (16-bit unsigned, 0.001g)	
0x51 for read	0x07	0x0E	Read HR motion magnitude threshold. Default: 0x0032 (0.05g)	[MSB] [LSB] motion magnitude threshold (16-bit unsigned, 0.001g)
0x50 for write	0x07	0x0F [MSB] [LSB]	Write minimum PD current (16-bit unsigned, 0.1uA).	
0x51 for read	0x07	0x0F	Read minimum PD current. Default: 0x0032	minimum PD current (16-bit unsigned, 0.1uA)
0x50 for write	0x07	0x10 [MSB] [LSB]	Write initial PD current (16-bit unsigned, 0.1uA). This sets the target PD current you would like AEC algorithm to maintain initially. It does not correspond to any register. Once	

			you set what PD current you need, algorithm will calculate the appropriate LED current.	
0x51 for read	0x07	0x10	Read initial PD current. Default: 0x0064	initial PD current (16-bit unsigned, 0.1uA)
0x50 for write	0x07	0x11 [MSB] [LSB]	Write target PD current (16-bit unsigned, 0.1uA). Applicable only if Auto Target PD Current Calculation is enabled.	
0x51 for read	0x07	0x11	Read target PD current. Default: 0x0064	Target PD current (16-bit unsigned, 0.1uA).
0x50 for write	0x07	0x12 [byte]	Write enable/disable automatic calculation of target PD current. byte: **0x00**: Disable **0x01**: Enable	
0x51 for read	0x07	0x12	Read enable/disable automatic calculation of target PD current	**0x00**: Disable **0x01**: Enable (default)
0x50 for write	0x07	0x13 [byte]	Write minimum integration time. byte: **0x00**: 14.8µs **0x01**: 29.4µs **0x02**: 58.7µs **0x03**: 117.3µs	
0x51 for read	0x07	0x13	Read minimum integration time.	Minimum integration time. **0x00**: 14.8µs (default) **0x01**: 29.4µs **0x02**: 58.7µs **0x03**: 117.3µs
0x50 for write	0x07	0x14 [byte]	Write minimum sampling rate and averaging. byte: 0x00: 25sps, avg = 1 0x01: 50sps, avg = 2 0x02: 100sps, avg = 4 0x03: 200sps, avg = 8 0x04: 400sps, avg = 16	
0x51 for read	0x07	0x14	Read minimum sampling rate and averaging.	Minimum sampling rate and averaging. **0x00**: 25sps, avg = 1 **0x01**: 50sps, avg = 2 (default) **0x02**: 100sps, avg = 4 **0x03**: 200sps, avg = 8 **0x04**: 400sps, avg = 16
0x50 for write	0x07	0x15 [byte]	Write maximum integration time: byte: **0x00**: 14.8µs **0x01**: 29.4µs **0x02**: 58.7µs **0x03**: 117.3µs	
0x51 for read	0x07	0x15	Read maximum integration time:	Maximum integration time: **0x00**: 14.8µs **0x01**: 29.4µs **0x02**: 58.7µs **0x03**: 117.3µs (default)
0x50 for write	0x07	0x16 [byte]	Write maximum sampling rate and averaging:	

			byte: **0x00**: 25sps, avg = 1 **0x01**: 50sps, avg = 2 **0x02**: 100sps, avg = 4 **0x03**: 200sps, avg = 8 **0x04**: 400sps, avg = 16	
0x51 for read	0x07	0x16	Read maximum sampling rate and averaging:	Maximum sampling rate and averaging: **0x00**: 25sps, avg = 1 **0x01**: 50sps, avg = 2 **0x02**: 100sps, avg = 4 (default) **0x03**: 200sps, avg = 8 **0x04**: 400sps, avg = 16
0x50 for write	0x07	0x17, 0xWX, 0xYZ	Write slot and PD configuration for the two HR inputs to the WHRM algorithm WX is input 1 of the WHRM algorithm. W = 0 for Slot 1 W = 1 for Slot 2 W = 2 for Slot 3 W = 3 for Slot 4 W = 4 for Slot 5 W = 5 for Slot 6 W = 7 for Slot not used X = 0 for PD1 X = 1 for PD2 X = 3 for PD not used. YZ is input 2 of the WHRM algorithm. Y = 0 for Slot 1 Y = 1 for Slot 2 Y = 2 for Slot 3 Y = 3 for Slot 4 Y = 4 for Slot 5 Y = 5 for Slot 6 Y = 7 for Slot not used Z = 0 for PD1 Z = 1 for PD2 Z = 3 for PD not used. The LED # that is fired in each slot is defined in 0x19 command. This command is not available for MAXM86161 and the MAXM86161 settings is set to 0x0073 by default.	-
0x51 for read	0x07	0x17	Read Slot and PD configuration for the two HR inputs to the WHRM algorithm. ,	**0xWX 0xYZ** MAX86141 default: 0x0001 MAX86140, MAXM86146, MAXM86161 default: 0x0073 WX is the LED/PD used for IR for the WSpO$_2$ algorithm.

				W = 0 for Slot 1 W = 1 for Slot 2 W = 2 for Slot 3 W = 3 for Slot 4 W = 4 for Slot 5 W = 5 for Slot 6 W = 7 for Slot not used X = 0 for PD1 X = 1 for PD2 X = 3 for PD not used. YZ is the LED/PD used for red for the $WSpO_2$ algorithm. WHRM algorithm. Y = 0 for Slot 1 Y = 1 for Slot 2 Y = 2 for Slot 3 Y = 3 for Slot 4 Y = 4 for Slot 5 Y = 5 for Slot 6 Y = 7 for Slot not used Z = 0 for PD1 Z = 1 for PD2 Z = 3 for PD not used
0x50 for write	0x07	0x18 0xWX 0xYZ	Write Slot and PD configuration for the IR, red inputs to the WSpO2 algorithm, WX is the LED/PD used for IR for the $WSpO_2$ algorithm. W = 0 for Slot 1 W = 1 for Slot 2 W = 2 for Slot 3 W = 3 for Slot 4 W = 4 for Slot 5 W = 5 for Slot 6 W = 7 for Slot not used X = 0 for PD1 X = 1 for PD2 X = 3 for PD not used. YZ is the LED/PD used for red for the $WSpO_2$ algorithm. Y = 0 for Slot 1 Y = 1 for Slot 2 Y = 2 for Slot 3 Y = 3 for Slot 4 Y = 4 for Slot 5 Y = 5 for Slot 6 Y = 7 for Slot not used Z = 0 for PD1 Z = 1 for PD2 Z = 3 for PD not used	-
0x50 for read	0x07	0x18	Read Slot and PD configuration for the IR, red inputs to the WSpO2 algorithm.	**0xWX 0xYZ**

				MAXM86141/40, MAXM86161 Default: 0x1020 MAXM86146 default: 0x2111 WX is the LED/PD used for IR for the WSpO$_2$ algorithm. W = 0 for Slot 1 W = 1 for Slot 2 W = 2 for Slot 3 W = 3 for Slot 4 W = 4 for Slot 5 W = 5 for Slot 6 W = 7 for Slot not used X = 0 for PD1 X = 1 for PD2 X = 3 for PD not used. YZ is the LED/PD used for red for the WSpO$_2$ algorithm. WHRM algorithm. Y = 0 for Slot 1 Y = 1 for Slot 2 Y = 2 for Slot 3 Y = 3 for Slot 4 Y = 4 for Slot 5 Y = 5 for Slot 6 Y = 7 for Slot not used Z = 0 for PD1 Z = 1 for PD2 Z = 3 for PD not used
0x50 for write	0x07	0x19 0xUV 0xWX 0xYZ (3x.12.x+)	Write slots used for LED firing sequence U is Slot 1 V is Slot 2 W is Slot 3 X is Slot 4 Y is Slot 5 Z is Slot 6 U, V, W, X, Y, Z are defined as: 0: No LED firing 1: LED1 firing 2: LED2 firing 3: LED3 firing 4: LED4 firing 5: LED5 firing 6: LED6 firing 7: LED1 and LED2 firing 8: LED1 and LED3 firing 9: LED2 and LED3 firing LED firing sequence in firing slots 1-3 (slot 1-6 in case of MAXM86146) PPGs are reported in the same order.	-

			The non-firing slots should appear to the end of sequence. Each LED can only be fired once (e.g. if one slot is set to LED1_AND_LED2, LED1 or LED 2, no other firing slot can be set to LED1	
0x51 for read	0x07	0x19	Read slots used for LED firing sequence.	**0xUV 0xWX 0xYZ** MAX86141/40, MAXM86161 default: 0x123000 MAXM86146 default: 0x123456 U is Slot 1 V is Slot 2 W is Slot 3 X is Slot 4 Y is Slot 5 Z is Slot 6 U, V, W, X, Y, Z are defined as: 0: No LED firing 1: LED1 firing 2: LED2 firing 3: LED3 firing 4: LED4 firing 5: LED5 firing 6: LED6 firing 7: LED1 and LED2 firing 8: LED1 and LED3 firing 9: LED2 and LED3 firing
0x50 for write	0x07	0x1A [byte]	Write initial integration time.(30.13.19+, 33.13.19+) byte: **0x00**: 14.8µs **0x01**: 29.4µs **0x02**: 58.7µs **0x03**: 117.3µs	
0x51 for read	0x07	0x1A	Read initial integration time. (30.13.19+, 33.13.19+)	**0x00**: 14.8µs **0x01**: 29.4µs **0x02**: 58.7µs **0x03**: 117.3µs (default)
0x50 for write	0x07	0x1B [byte]	Write initial sampling rate and averaging. (30.13.19+, 33.13.19+) byte: 0x00: 25sps, avg = 1 0x01: 50sps, avg = 2 0x02: 100sps, avg = 4 0x03: 200sps, avg = 8 0x04: 400sps, avg = 16	

0x51 for read	0x07	0x1B	Read initial sampling rate and averaging. (30.13.19+, 33.13.19+)	**0x00**: 25sps, avg = 1 **0x01**: 50sps, avg = 2 (default) **0x02**: 100sps, avg = 4 (default) **0x03**: 200sps, avg = 8 **0x04**: 400sps, avg = 16
0x50 for write	0x07	0x1D [byte]	Write initial SpO2 perfusion index threshold (red, IR). (30.13.19+, 33.13.19+) byte: 0x01 – 0xFF: 0.001 to 0.255	
0x51 for read	0x07	0x1D	Read initial SpO2 perfusion index threshold (red, IR). (30.13.19+, 33.13.19+)	byte: 0x01 – 0xFF: 0.001 to 0.255 (default is 0.05, 0x32)
0x52	0x07	**0x00**: Disable (CMD_DELAY = 120ms) **0x01**: Enable Normal Algorithm Report (CMD_DELAY = 465ms) **0x02**: Enable Extended Algorithm Report (CMD_DELAY = 465ms)	Write enable/disable the Wearable Algorithm Suite (WHRM+WSpO$_2$) algorithm.	
0x80	0x00	Use bytes 0x28 to 0x32 from the .msbl file as the IV bytes.	Bootloader mode flash the application .msbl: Set the initialization vector (IV) bytes.	
0x80	0x01	Use bytes 0x34 to 0x43 from the .msbl file.	Bootloader mode flash the application .msbl: Set the authentication bytes.	
0x80	0x02	0x00, Number of pages located at byte 0x44 from the .msbl file.	Bootloader mode flash the application .msbl: Set the number of pages to flash.	
0x80	0x03	- (CMD_DELAY = 1400ms)	Bootloader mode flash the application .msbl: Erase the application flash memory.	
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 plus 16 bytes of CRC.	Bootloader mode flash the application .msbl: Send the page values. Each page sent includes 16 CRC bytes for that page, so there are 8208 bytes per page sent in the payload of the message.	

		(CMD_DELAY = 680ms)		
0x81	0x00	-	Bootloader mode flash the application .msbl: Get bootloader version.	
0x81	0x01	-	Bootloader mode flash the application .msbl: Get the page size in bytes.	
0xB3 read			Read the sensor hub initial authentication vector for Maxim Wellness Library Suite	[ARRAY0] ARRAY0: six bytes of sensor hub initialization vector data which are inputs to mxm_algosuite_manager_getauthinitials()
0XB5 write	[ARRAY1] twelve bytes of library public key which are output from mxm_algosuite_manager_getauthinitials()		Send the public key from the Maxim Wellness Library Suite to the sensor hub	
0xB4 read			Read the 12-byte sensor hub authentication public key for the Maxim Wellness Library Suite which will form the first 12 bytes of ARRAY2.	Twelve bytes of the authentication data
0xB2 read			Read the 32-byte sensor hub authentication public key for Maxim Wellness Library Suite which will form the last 32 bytes of ARRAY2.	
0xB8 write	0x01		Set the sensor hub to use MFIO to wake the host, MFIO Interrupt Mode.	
0xFF	0x03		Read the sensor hub version.	Major version byte, Minor version byte, Revision byte
0xFF	0x07		Read the algorithm: version. Deprecated - no longer supported.	Major version byte, Minor version byte, Revision byte. Deprecated

5 Using SCD State and Motion Detection for Power Saving

For the case of when the device is not on the skin, a motion-detection-enabled state machine can be implemented in the host to reduce power consumption. In this case, the sensor hub stays in sleep mode until a motion event is reported by the accelerometer, or an I^2C command is received from the host. Below is an example of such a state machine.

- **Active State:** Normally, the sensor hub runs in Active state in AEC mode (**section 3.2**) or Power Saving Mode (**section 3.4**). If the SCD state in the report shows off-skin for certain time, the state machine switches to Probing state.
- **Probing State:** In this state, the host periodically turns the algorithm on and off. If an On-Skin state is reported while the algorithm is running, it will switch back to Active state and continue running the algorithm. Otherwise, after several attempts of turning the algorithm on and off (the off period can be increased after each attempt), it will switch to Off-Skin state. In Active and Probing states, the procedure to start, read report, or stop are similar to the regular sequence described in **Table 8** for AEC mode**,** or as highlighted for Power Saving mode.
- **Off-Skin State**: In Off-Skin state, the goal is to save more power by allowing the sensor hub to stay in sleep mode, so long as there is no motion. Depending on the use of a host or sensor hub accelerometer (**section 1.3**), the host is required to configure the sensor hub differently, as shown in **Table 15.**
- If the KX122/LIS2DS12 is connected to the sensor hub, the sensor hub must be configured to wake up on motion. In this case, the accelerometer is enabled in the interrupt mode and the motion threshold and the duration of motion is configured using the wake up on motion configuration command, as shown in **Table 15.**

  To support this feature, the interrupt line of the accelerometer is required to be connected to the sensor hub. Once the sensor hub is configured, the host should start only the accelerometer. As soon as a motion interrupt occurs, the sensor hub will wake up and read accelerometer samples and store them in the sensor hub FIFO. The host should periodically read the sensor hub FIFO to check if any accelerometer sample has been captured since the last polling period. If there is a sample, the host should switch to Active state by first disabling the wake up on motion configuration and then restarting the algorithm.

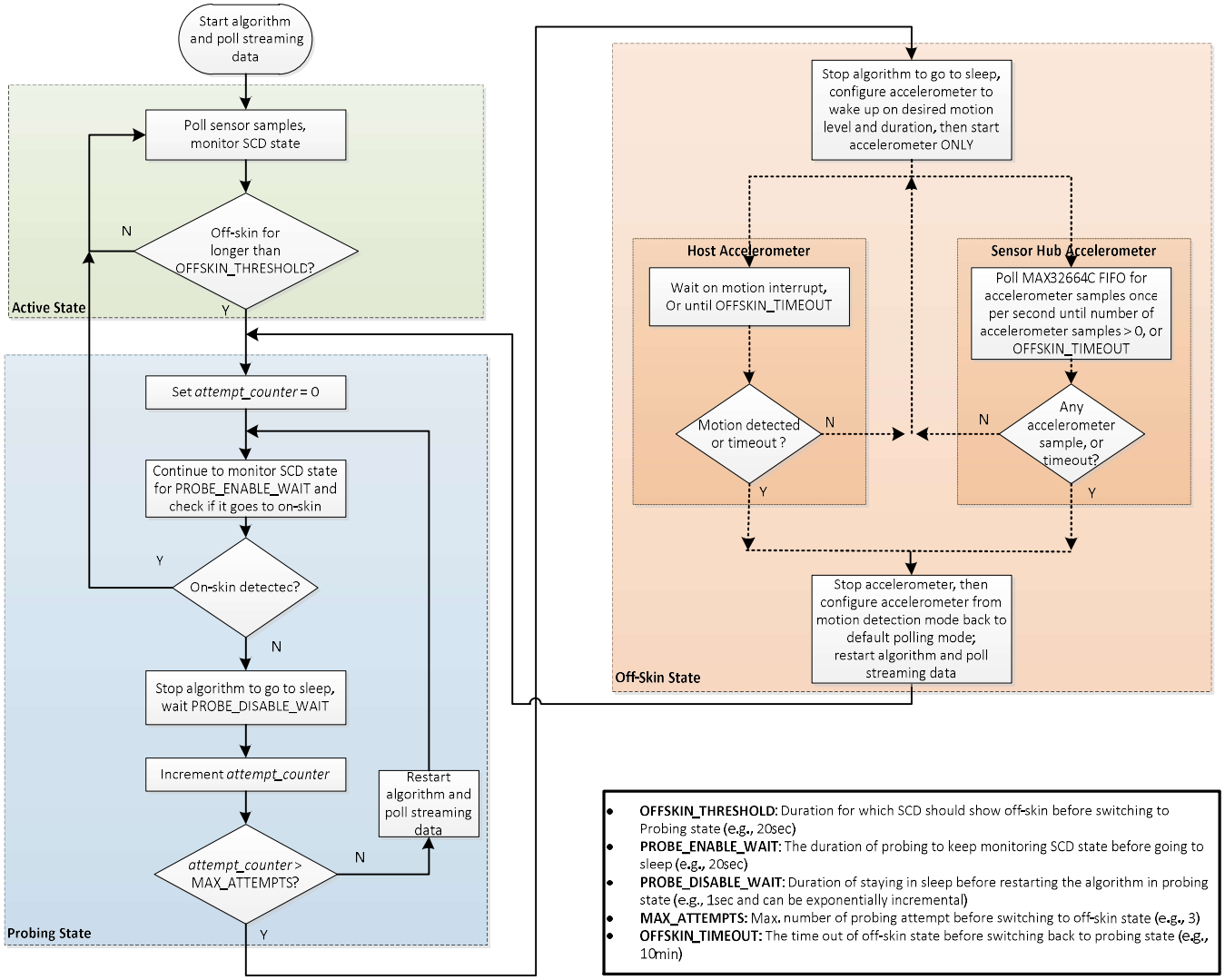


Figure 8. Example of an SCD-enabled, power saving state machine.

Table 16. Host Commands to Enable/Disable Wake Up on Motion Configuration of Sensor Hub Accelerometer for Off-Skin State

	#	HOST COMMAND (HEX)	COMMAND DESCRIPTION	RESPONSE (HEX)
START OFF-SKIN STATE	Host initializes the sensor hub to wake up on accelerometer motion detection to go into off-skin state.			
	1.1	Stop the sensor, accelerometer, and algorithm if already enabled, as seen in step 3 in **Table 8**		
	1.2	AA 46 04 00 01 [05] [08]	Set the sensor hub accelerator in wake up on motion mode if motion is greater than a threshold for a certain duration, for example: [05]: 0.2s motion duration [08]: 0.5g motion	AB 00
	1.3	AA 10 00 01	Set the output FIFO mode to accelerometer data only.	AB 00
	1.4	AA 10 02 01	Set the samples report period to 40ms. Samples report rate to be one report per every sensor sample.*	AB 00
	1.5	AA 44 04 01 00 (CMD_DELAY = 20ms)	Enable the sensor-hub accelerometer. It will generate a report only if there is motion, according to step 1.4.	AB 00
READING SAMPLES REPORT IN OUTPUT FIFO	Host reads samples periodically repeatedly during off-skin state:			
	2.1	AA 00 00	Read sensor hub status byte: Bit 0: Sensor comm error Bits 1 and 2: Reserved Bit 3: FIFO filled to threshold (DataRdyInt) Bit 4: Output FIFO overflow (FifoOutOvrInt) Bit 5: Input FIFO overflow (FifoInOverInt) Bit 6: Sensor hub busy (DevBusy) Bit 7: Reserved If DataRdyInt is set, proceed to next step.	AB 00 08
	2.2	AA 12 00	Get the number of samples (nn) in the FIFO.	AB 00 nn
	2.3	AA 12 01	Read the data stored in the FIFO; nn samples will be read. The samples report will only include accelerometer data (6 bytes).	AB 00 data_for_ nn_samples
END OFF-SKIN	Host ends the wake up on motion configuration.			
	3.1	AA 44 04 00 (CMD_DELAY = 20ms)	Disable the accelerometer.	AB 00
	3.2	AA 46 04 00 00 FF FF	Disable wake up on motion. See **Table 13.**	AB 00
	3.3	Proceed to start algorithm in AEC or Power Saving mode as in **Table 8.**		

6 Host Implemented Power Savings Using Accelerometer Wake

If the host has spare GPIOs, the accel interrupt can also be connected to the host. In this case, when SCD detects "off skin", the sensor hub/AFE may be put to deep sleep/shutdown and the accel can be configured to wake on motion via register writes from the host. Alternatively MAX86141 interrupt and proximity detection may be used in a similar manner.

7 SCD State Machine for MFIO Interrupt Mode.

SCD state machine for MFIO interrupt mode (SCDSM) is sensor hub application mode firmware algorithm which is only enabled in the MFIO interrupt mode. SCDSM is enabled via the command sequence AA BC 01 and it is disabled via the command sequence AA BB 00. SCDSM is also disabled when the algorithm is disabled or after a reset. Figure 11 is the flow chart for SCDSM. (SCDSM is not available for LIS2DS12)

- SCDSM is valid only when sensor hub has direct connection to accelerometer.
- SCDSM is only available in MFIO interrupt mode.
- SCDSM is only available for continuous HRM modes.
- Enable SCDSM after the algorithm is enabled.

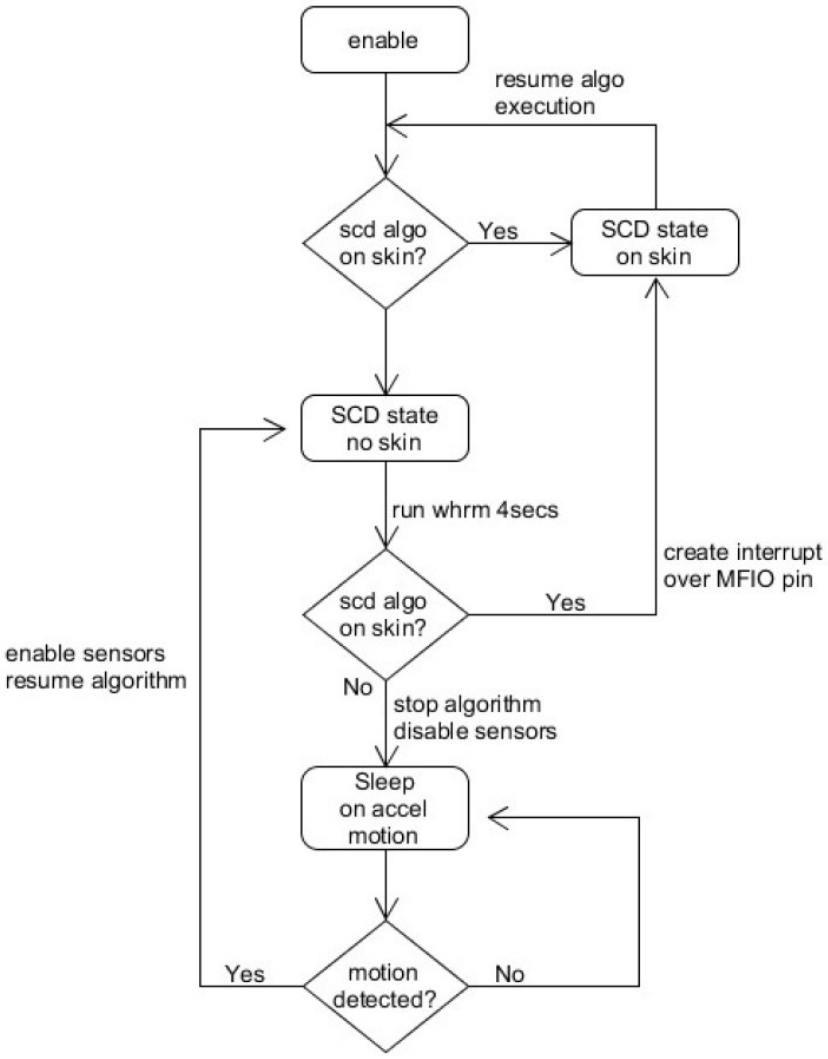


Figure 9. SCDSM flowchart.

Table 17. SCDSM Sensor Hub Commands

HOST I2C COMMAND (HEX)	DESCRIPTION	RESPONSE (HEX)
0xAA 0xBB 0x01	Disable skin contact state machine (SCDSM) on sensor hub.	AB 00
0xAA 0xBC 0x01	Enable skin contact state machine (SCDSM) on sensor hub. - SCDSM is valid only when sensor hub has direct connection to accelerometer. - SCDSM is only available in MFIO interrupt mode. - SCDSM is only available for continuous HRM modes. - Enable SCDSM after the algorithm is enabled.	AB 00
0xAA 0xBD 0x00	Get the status of skin contact state machine (SCDSM)	AB 00 <scdm_status> scdm_status: **0x00:** Disabled **0x01:** Enabled

Note: SCDSM functionality is not defined for ST LISDS12 configuration and accordingly not defined for MAX32664C – MAX86174 combination , ie 35.x.y releases.

8 Authentication Process for Maxim Wellness Suite Library

The wrist-based Maxim Wellness Suite Library (HRV, respiration rate, sleep quality, stress, sports coaching) is available as a separate library. Algorithms within in the Wellness Suite Library will not operate without proper authentication from the sensor hub. The authentication sequence with sensor hub is listed below.

Table 18. Authentication Commands

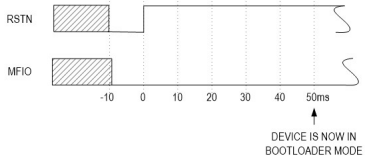
	#	HOST COMMAND (HEX)	COMMAND DESCRIPTION	RESPONSE (HEX)
AUTHENTICATE	Host executes the authentication steps below.			
	1.1	AA B3 00	Read authentication ARRAY0, 6 bytes, from sensor hub.	AB 00 <ARRAY0, six bytes>
	1.2		ARRAY0 is an input to `mxm_algosuite_manager_getauthinitials()*`, wellness library suite. `mxm_algosuite_manager_getauthinitials()*` outputs ARRAY1, 12 bytes	
	1.3	AA B5 00 <ARRAY1 twelve bytes>	Send ARRAY1, 12 bytes, to sensor hub.	AB 00
	1.4	AA B4 00	Read authentication ARRAY2, 12 bytes, from sensor hub.	AB 00 <ARRAY2, twelve bytes >
	1.5	AA B2 00	Read authentication ARRAY3, 32 bytes, from sensor hub.	AB 00 <ARRAY2, thirty-two bytes >
	1.6		To finalize authentication in wellness library suite call `mxm_algosuite_manager_authenticate()*` with the inputs ARRAY2 (12 bytes) and ARRAY3. (32 bytes)	

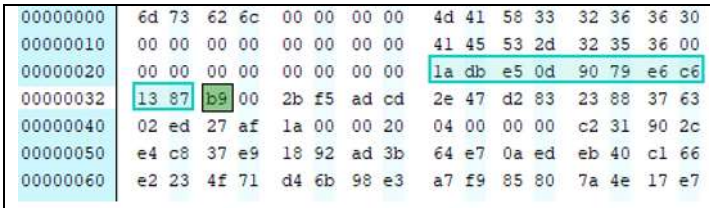
* Refer to Wellness Library Integration Guide distributed with MRD103 Documentation.

9 Application .msbl Programming Sequence

To program the MAX32664C application .msbl, the host microprocessor may implement the software to flash the .msbl file. The MAX32664C uses the 8-bit slave address of 0xAA. 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. Values for the number of pages, initialization vector, authorization bytes, and page bytes may be different for the latest .msbl, but the locations of these values in the .msbl file remain the same. There are additional bytes in the .msbl past the last page; these are the file checksum bytes. Since the bootloader uses the commands listed below and it does not accept files, the file checksum bytes are not used by the bootloader.

Table 19. Annotated I2C Trace for Flashing the Application

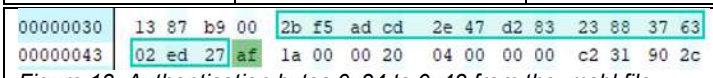
HOST COMMAND	COMMAND DESCRIPTION	READ MAX32664C RESPONSE	RESPONSE DESCRIPTION
Sequence the MAX32664C to enter bootloader mode. *  *Figure 10. Sequence to enter bootloader mode.*			
0xAA 0x01 0x00 0x08*	Set mode to 0x08 for bootloader mode.	0xAB 0x00	No error.
0xAA 0x02 0x00	Read mode.	0xAB 0x00 0x08	No error. Mode is bootloader.
0xAA 0x81 0x00	Read bootloader firmware version.	0xAB 0x00 0x03 0xYY 0xZZ	No error. Version is 3.YY.ZZ.
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.	0xAB 0x00	No error.
00000044 | 02 ed 27 af 1a 00 00 20 04 00 00 00 c2 31 90 2c *Figure 11. 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 to the 0x28 to 0x32 values 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 12. Initialization vector bytes 0x28 to 0x32 from the .msbl file.

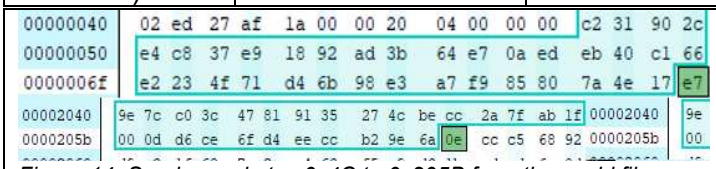
0xAA 0x80 0x01 0x2B 0xF5 0xAD 0xCD 0x2E 0x47 0xD2 0x83 0x23 0x88 0x37 0x63 0x02 0xED 0x27 0xAF*	Bootloader flash. Set the authentication bytes to the 0x34 to 0x43 values from the .msbl file.	0xAB 0x00	No error.



```
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 13. Authentication bytes 0x34 to 0x43 from the .msbl file.

0xAA 0x80 0x03* (CMD_DELAY = 1400ms)	Bootloader flash. Erase application.	0xAB 0x00	No error.
0xAA 0x80 0x04 0xC2 0x31 0x90 ... 0x9E 0x6A 0x0E* (CMD_DELAY = 680ms)	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
00002040  9e 7c c0 3c  47 81 91 35  27 4c be cc  2a 7f ab 1f  00002040  9e
0000205b  00 0d d6 ce  6f d4 ee cc  b2 9e 6a 0e  cc c5 68 92  0000205b  00
```

Figure 14. Send page bytes 0x4C to 0x205B from the .msbl file.

0xAA 0x80 0x04 0xCC 0xC5 0x68 ... 0xF7 0xD6 0x4C* (CMD_DELAY = 680ms)	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* (CMD_DELAY = 680ms)	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* (CMD_DELAY = 680ms)	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* (CMD_DELAY = 680ms)	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* (CMD_DELAY = 680ms)	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* (CMD_DELAY = 680ms)	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* (CMD_DELAY = 680ms)	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* (CMD_DELAY = 680ms)	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* (CMD_DELAY = 680ms)	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* (CMD_DELAY = 680ms)	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* (CMD_DELAY = 680ms)	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* (CMD_DELAY = 680ms)	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* (CMD_DELAY = 680ms)	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* (CMD_DELAY = 680ms)	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* (CMD_DELAY = 680ms)	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* (CMD_DELAY = 680ms)	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* (CMD_DELAY = 680ms)	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* (CMD_DELAY = 680ms)	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* (CMD_DELAY = 680ms)	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* (CMD_DELAY = 680ms)	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* (CMD_DELAY = 680ms)	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* (CMD_DELAY = 680ms)	Bootloader flash. Send page bytes 0x2E1BC to 0x301CB from the .msbl file.	0xAB 0x00	No error.
0xAA 0x80 0x04 0x44 0xEC 0xE6 ...	Bootloader flash. Send page bytes 0x301CC to	0xAB 0x00	No error.

0xBC 0xC9 0x5E* (CMD_DELAY = 680ms)	0x321DB from the .msbl file.		
0xAA 0x80 0x04 0xD3 0x58 0x34 ... 0x62 0x00 0x37* (CMD_DELAY = 680ms)	Bootloader flash. Send page bytes 0x321DC to 0x341EB from the .msbl file.	0xAB 0x00	No error.

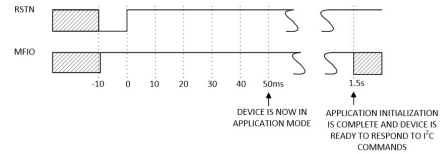


Figure 15. Sequence to enter application mode.

Alternately, the MAX32664C can be commanded to application mode.+			
0xAA 0x01 0x00 0x00+ (CMD_DELAY = 1.5s)	Set mode to 0x00 for application mode.	0xAB 0x00	No error.
0xAA 0x02 0x00+	Read mode.	0xAB 0x00 0x00	No errors. Mode is application.

**Mandatory*

+Recommended

10 Piping in Host Accelerometer Data to the Sensor Hub

The table below provides the sequence of commands for writing external (host connected) accelerometer data to the input FIFO of the sensor hub; The KX122/LIS2DS12 is not connected to the connected to the MAX32664 sensor hub.

Table 20. Sequence of Commands to Write External Accelerometer Data to the Input FIFO

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 the threshold for the FIFO to 0x05.	0xAB 0x00	No error.
0xAA 0x44 0x04 0x01 0x01* (CMD_DELAY = 20ms)	Enable the input FIFO for host supplied accelerometer data.	0xAB 0x00	No error.
0xAA 0x52 0x07 0x01* (CMD_DELAY = 465ms)	Enable normal algorithm report.	0xAB 0x00	No error.
0xAA 0x14 0x00 Sample 1 value to Sample N value*	Write data to the input FIFO. 6 bytes per accelerometer sample. Example, send five sets of host accel [LSBx MSBx, LSBy MSBy, LSBy MSBz] data to the sensor hub: 14 00 84 FE 1C 00 55 FC 85 FE 1B 00 57 FC 84 FE 1A 00 56 FC 85 FE 19 00 54 FC 87 FE 1B 00 57 FC -380=FE84, 28=001C, -939=FC55; -379=85FE, 27=001B, -938=FC56; -380=FE84, , 26=001A, -938=FC56; -379=85FE, 25=0019, -940=FC54; -377=87FE, 27=001B, -937=FC57	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 [0x06]	No error. [six] samples are in the FIFO.
0xAA 0x12 0x01*	Read the [six] sensor data and normal algorithm report stored in the FIFO.	0xAB 0x00 6*[sensor data + normal algorithm report]	No error. 6*[sensor data + normal algorithm report]

**Mandatory*
†Recommended

11 Sensor Hub Methods for Reset, Sleep, Shutdown

Table 21. Sensor Hub Methods for Reset, Sleep, Shutdown

COMMAND NAME	HOST COMMAND TO MAX32664	DESCRIPTION

MAX32664A/B/C/D shutdown	0xAA 0x01 0x00 0x01	Place the MAX32664 into shutdown (MAX32660 "Backup" mode with RAM disabled). Restart by power cycling or pulsing RSTN.
MAX32664 sleeps when idle and MFIO is low.		V20.2.0+, v30.2.4+, v32.1.2+, v33.6.0 uses deep-sleep for low-powered mode when idle and MFIO low.
Use write register command for MAX86140/MAX86141/ MAXM86146/MAXM86161 to turn off LED1, LED2, and LED3.	0x40 0x00 0x23 0x00 0x40 0x00 0x24 0x00 0x40 0x00 0x25 0x00 0x40 0x00 0x20 0x99 0x40 0x00 0x21 0x09	Set the current of LED1, LED2, and LED3 to 0. Set the sequence control registers LEDC1, LEDC2, and LEDC3 to ambient.
MAX86140/MAX86141/ MAXM86146/MAXM86161 AFE shutdown. Use AFE write register command.	0xAA 0x40 0x00 0x0D 0x02	Write 0x02 to MAX86140/1, MAXM86146/61 register 0x0D (System Control) to put the MAX86140/1, MAXM86146/61 into shutdown (SHDN) mode. The AFE must be enabled using the enable command when using the read, write AFE register command)
KX122, KX112 standby. Use accel write register command.	0xAA 0x40 0x04 0x18 0x00	Write 0x00 to KX122 register 0x018 (CNTL1) to put the KX122 into "Standby" mode. The KX122 must be enabled using the enable command when using the read, write KX122 register command)
LIS2DS12 wake on activity or wake on double/single tap. Use accel write register command.	0xAA 0x40 0x04 [reg_addr] [value]	See AN4748 LIS2DS12 for register settings. Accel interrupt pin can be connected to the host GPIO.
MAX32664 hard reset	Use MFIO and RSTN pins according to section 1.1	
WDT in MAX32664		Not implemented.

12 MAX32664 Power Consumption Estimate

The MAX32664 sensor hub family runs in two distinct operating modes. The Active mode is the mode in which the execution of the firmware occurs. The Deep Sleep mode is enabled by the sensor hub to save power when the processor is idle or there is no need for any processing. It makes all internal clocks of the MAX32664 gated off. In this mode, only RTC is enabled as a source of backup for wakeup. As soon as a sensor interrupt is received, the MAX32664 wakes up, completes the processing, and goes back to sleep. It also must wake up prior to I^2C communication by setting MFIO low.

Table 22. Comparison of Active and Deep Sleep Power—Single Supply (V_{DD} Only)

MAX32664 OPERATIONAL MODE	POWER CONSUMPTION
Active	15.5664mW
Deep Sleep	0.00756mW

The tables below show the power consumption in each mode. To estimate the power consumption while running the algorithm, the percentage of time that the MAX32664 is in Active mode is measured. For this measurement, the report interval is set to 1 second and only algorithm data is reported.

Table 23. Estimated Power Consumption for the MAX32664C

WEARABLE SUITE ALGORITHM	MEASURED CPU ACTIVE TIME (AVERAGE) %	CALCULATED POWER CONSUMPTION (AVERAGE)*
		SINGLE-SUPPLY V_{DD} + INTERNAL LDO
Continuous HRM + Continuous SPO_2 mode	4.7%	0.74mW
Continuous HRM	4.3%	0.68mW
Sampled HRM	4.3%	0.68mW
Activity Tracking Only	4.2%	0.66mW

*V_{DD}: 1.8V and CPU clock: 96MHz.

13 Sensor Hub .msbl Version Numbering Convention

Table 24. Sensor Hub .msbl Version Numbering Convention

MAX32664+ Optical Sensor	.msbl versions	WLP IC	Form Factor
MAX32664C+ MAX86141/0	MAX32664C_HSP2_WHRM_AEC_SCD_WSPO2_C_30.x.y .msbl	MAX32664GWEC	Wrist
	MAX32664C_HSP2_WHRM_AEC_SCD_WSPO2_Z_30.x.y .msbl	MAX32664GWEZ	
MAXM86146	MAX32664C_OB07_WHRM_AEC_SCD_WSPO2_C_33.x.y.msbl	MAX32664GWEC	Chest
MAX32664C+ MAXM86161	MAX32664C_MAXM86161_WHRM_AEC_SCD_WSPO2_C_32.x.y.msbl	MAX32664GWEC	Ear
	MAX32664C_MAXM86161_WHRM_AEC_SCD_WSPO2_Z_32.x.y.msbl	MAX32664GWEZ	

14 Default Application .msbl Versions Pre-Programmed on the MAX32664C

The MAX32664C is pre-programmed with the bootloader and the application .msbl application/sensor hub version listed in the table below. It is recommended that the sensor hub be updated with the latest application .msbl available on the Maxim Integrated website in order to be compatible with the latest sensor hub documentation.

Table 25. MAX32664C/MAXM86146 Pre-Programmed .msbl Version

MAXIM PART	PRE-PROGRAMMED .msbl APPLICATION/SENSOR HUB VERSION
MAX32664C	Version 30.2.2 (MAX86141) (deprecated)
MAXM86146	Application not pre-programmed.

15 MAX32664 Processing Capabilities

The MAX32664 IC hardware is the same as the MAX32660.

1. MIPS: Arm Cortex-M4 with FPU: 1.27 Dhrystone MIPS/MHz
2. RAM: 96kB SRAM
3. Flash: 256kB Flash Memory
4. CPU Frequency: 96MHz

16 References

MAX32664 website: MAX32664 user guides; C-keyed .msbl for MAX32664C; sample host code: MAX32664 Design Resources Website

Application Note 7148, protocol definition between sample host (MAX32630) and PC UART/BLE: Interface Guide for MAX32664 Sensor Hub-Based Reference Design Platforms

Frequently Asked Questions: Maxim Support Center

MAXREFDES103# hardware, software files: MAXREFDES103#: Wrist-Based SpO_2, HR, and HRV Health Sensor Platform

MAXM86146EVSYS# hardware, software files: MAXM86146: Evaluation System for the MAXM86146

MAXM86161EVSYS# hardware, software files: MAXM86161: Evaluation System for the MAXM86161

Validation And Performance Of A Wearable Heart rate Monitoring Algorithm

Revision History

REVISION NUMBER	REVISION DATE	DESCRIPTION	PAGES CHANGED
0	06/19	Initial release	—
1	08/19	Updated for low power and host accelerometer.	All
2	08/19	Updated 1.2.2 Polling Period. Updated Table 12 for configuration index 0x15, 0x16, 0x17, and 0x18.	8, 23
3	10/19	Updated tables 4, 8, and 9 for definition of reported R value. Updated Table 11 for family bytes 0x46, 0x01, 0x03. New Table 12 to include additional commands in support of sensors like the MAXM86161 with an I^2C connection. New section 5 on SCD-enabled power saving.	All
4	1/20	Updated section 1, section 3.2, Table 7, Table 8, Table 9, Table 10, section 3.4, Table 11; added section 4.1	5, 14-18, 20-23
5	2/21	Figures 3, 4, 6, 7, 9-12 added. Figures 1, 2, and 8 updated. Updated all sections and Tables. Sections 3.2 and 3.3 swapped. Added section 3.5, 6, 7, 8, and 10. Table 8 and 11 swapped; Deleted Table 16. Added Tables 12, 13, 17, and 18. Four bytes added to the algorithm FIFO output. Added section 10.	All
6	3/21	Added in section 9, 10, 11. Updated section 4.1, 8. Updated Table 1,14 and Figure 7. Table 15 merged with Table 14.	All
7.1	6/21	Table 4, 9, 10, 14 updated. Updated figure 1, table 8,11, 20, section 4. Added section 14, 15, table 24.	All
7.2	9/13	Updated figure 4, table 4,5,7,8,8,10,11	All

8.3	01/20/2022	Added new algo mode to table 14.	40.41