MAX32664C AlgoHub – Developers Guide

Rev 0; 02/21

Abstract

The MAX32664C may be programmed to be an algorithm hub which can provide processed data for the measurement of SpO_2 and heart rate. This document provides step-by-step instructions that enable a host processor to communicate with the algorithm hub.

Table of Contents

List of Figures

List of Tables

Introduction

MAX32664C supports algorithm hub (AlgoHub) feature which provides the following innovative features:

- Biometric heart rate, SpO2 algorithm developed by a team of machine learning, data scientist, algorithm experts.
- OTA update compatibility allows the AlgoHub to receive the latest algorithm version.
- Low-powered AlgoHub which utilizes deep sleep when idle
- Faster time to market; development time cut by at least six months.
- Reference design includes a host processor and PC GUI app which allows for wrist-band evaluation.

The AlgoHub provides processed heart rate and SpO2 data when provided with the raw accelerometer (accel)l, PPG data.

NOTE: The instructions in this document are compatible with the MAX32664C firmware version 30.13.10 and beyond.

1 Architecture

The host processor is able to send PPG and accel data to the AlgoHub algorithms on a chip. The AlgoHub processes the inputs and provides heart rate and SpO2 metrics as an output. The AlgoHub is power optimized to deep sleep when it is idle.

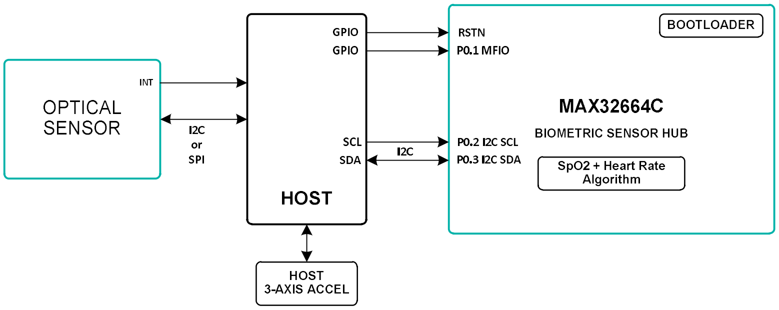


Figure 1. Architecture diagram for MAXREFDES103 health-sensing platform.

1.1 Reset to Bootloader or Application Mode

A typical health-sensing design includes a host microcontroller that communicates with the AlgoHub 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 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 AlgoHub is in Bootloader mode.

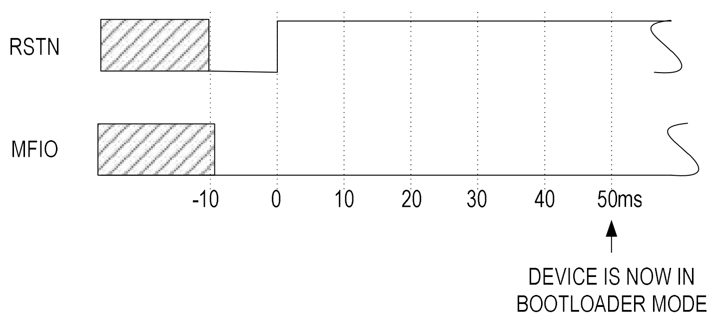


Figure 2 Entering bootloader mode using the RSTN pin and the MFIO GPIO pin.

To enter Application mode:

- Set the RSTN pin low for 10ms.
- While RSTN is low, set the MFIO pin to high.

- After the 10ms has elapsed, set the RSTN pin 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 AlgoHub 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 I^2C commands.

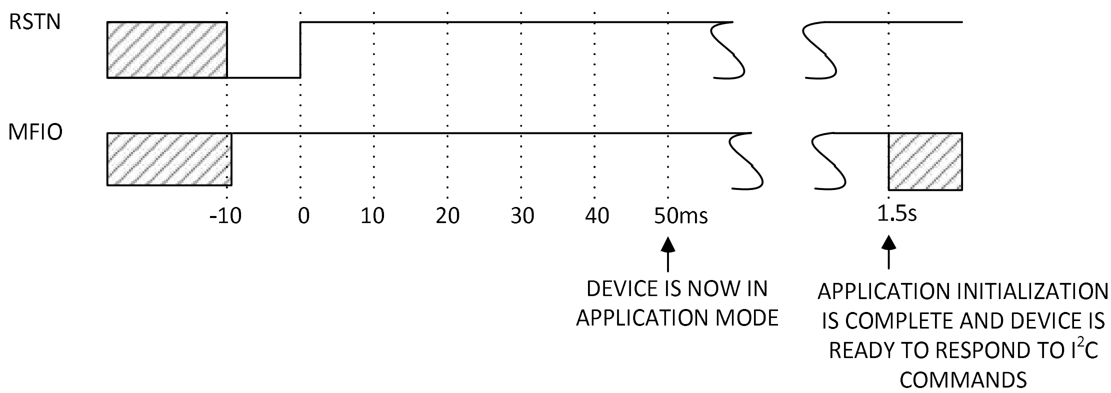


Figure 3 Entering application mode using the RSTN pin and MFIO pin.

To enter Application mode by timing out from Bootloader mode:

- Set the RSTN pin 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 AlgoHub is in Bootloader mode.
- If no I2C commands are sent to the AlgoHub within the next 1s, then the AlgoHub will automatically switch to application mode.

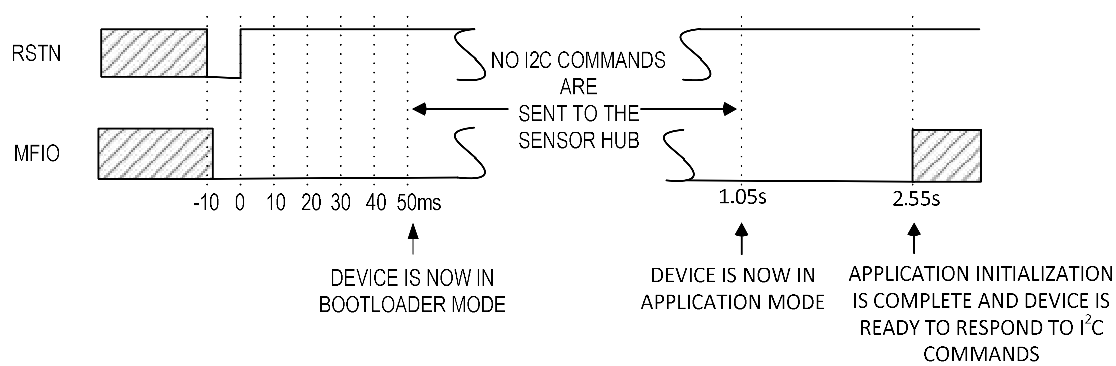


Figure 4 Entering application mode by timing out from Bootloader mode.

1.2 AlgoHub Handshaking

1.2.1 I2C and MFIO pin

Normally, when the AlgoHub is idle, it switches to deep sleep mode to save power. An external interrupt-like sensor, host MFIO, or RTC alarm forces the AlgoHub to wake up.

The host is required to wake up the AlgoHub 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 AlgoHub.
- Keeping the MFIO pin low during the I2C transaction.
- Setting MFIO to high after the end of I2C communication to allow the AlgoHub to switch back to deep sleep.

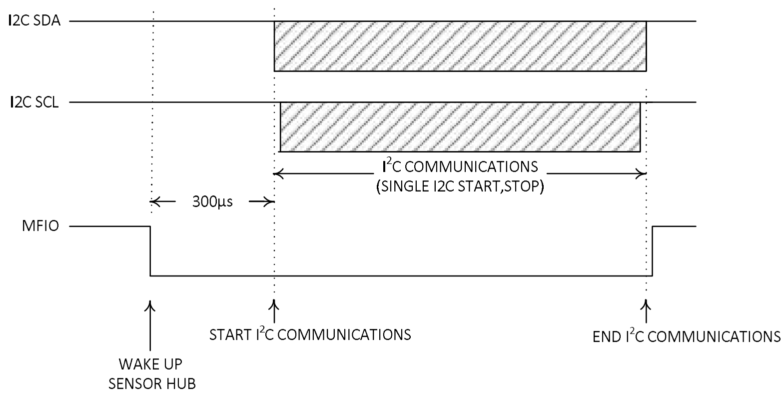


Figure 5. I2C and MFIO pin: host uses MFIO for enabling host communications.

2 AlgoHub I2C Communications

A host uses the I2C bus to communicate with the AlgoHub (slave) using a series of commands. The default CMD_DELAY is 2ms.

A generic write command includes the following fields:

```
Slave_WriteAddress(1 byte)|Command_Family(1 byte)|[Index byte]|[Write
byte]|[additional command byte(s)]
```

A generic response includes the following fields:

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

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

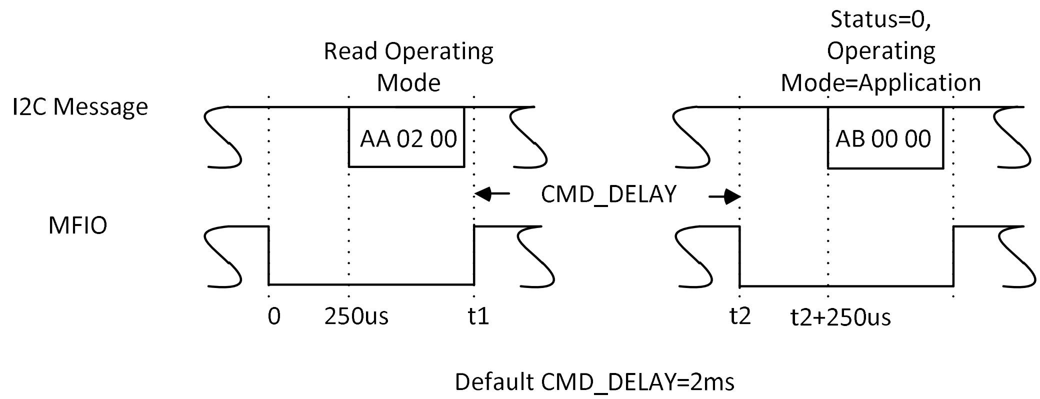


Figure 6. Example I²C, MFIO example command response.

2.1 Read Status Byte

The read status byte is an indicator of success (0x00) or failure, as detailed in the following table.

Table 1. Read Status Byte

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 Family Byte, Index Byte, and Write Byte are correct.
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 MAX32664C.
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. The AlgoHub is in deep sleep unless the host sets the MFIO pin low 300us before during the I2C communications. When switching to bootloader mode, allow 50ms for initialization. When switching to application mode, allow 1.5s for initialization.
NAK	NAK received. AlgoHub 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 . Verify that the hardware I2C/MFIO rise times, voltage levels, and grounding are correct. . Verify that the MFIO line fall time is clean; increase the MFIO pin low time to wake to 300us

2.2 AlgoHub I2C Command Response Definitions

The table below defines the I2C command response message protocol for the AlgoHub.

Table 2. AlgoHub I2C Command Response Definitions

FAMILY BYTE	INDEX BYTE	WRITE BYTE	DESCRIPTION	RESPONSE BYTES
0x00	0x00	-	Read sensor AlgoHub 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	0x00	0x00: Exit bootloader mode, enter application mode. 0x01: Shutdown the MAX32664C. Restart by power cycling or pulsing RSTN. 0x02: Reset. 0x08: Enter bootloader mode.	Set the device operating mode.	-
0x02	0x00	-	Read the device operating mode.	0x00: Application operating mode. 0x02: Reset. 0x08: Bootloader operating mode.
0x10	0x00	0x00: Pause (no data) 0x01: Sensor Data 0x02: Algorithm Data 0x03: Sensor Data and Algorithm Data 0x04: Pause (no data) 0x05: Sample Counter byte, Sensor Data 0x06: Sample Counter byte, Algorithm Data 0x07: Sample Counter byte,	**Set the output format to 0x01 for AlgoHub operation.**	-

		Sensor Data and Algorithm Data		
0x11	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	0x01	0x01 to 0xFF: Sensor Hub Interrupt Thresh for fifo	Set the threshold for the FIFO interrupt bit.The bit DataRdyInt of the AlgoHub status byte is set when this threshold is reached.	-
0x11	0x02			**0x01 (default) to 0xFF:** LSB is 40ms. N, where a samples report is generated once every N samples.
0x10	0x02	**0x01 to 0xFF:** 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).	-
0x11	0x01	-	Read the threshold for the FIFO interrupt bit.The bit DataRdyInt of the AlgoHub status byte is set when this threshold is reached.	0x01 to 0xFF: Sensor Hub Interrupt Threshold for FIFO.
0x10	0x02	0x01 to 0xFF: 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).	-
0x11	0x02	-	Read the samples reporting period (e.g., a value of 25 means a report is generated once every 1s. The default of 1 is one report is generated once per sample or every 40ms).	0x01 (default) to 0xFF: LSB is 40ms. N, where a samples report is generated once every N samples.
0x10	0x03	0x02 to 0xFF: New I^2C address (8-bit I^2C write address)	Change I^2C address of the MAX32664.	

0x11	0x03	-	Read the I^2C address of the AlgoHub	0x02 to 0xFF: I^2C address
0x10	0x04	0x00 to 0xFF: Counter	Set the sensor hub counter.	
0x11	0x04	-	Read the sensor hub counter.	0x00 to 0xFF: Counter
0x11	0x05	-	Read PPG output FIFO samples report size	Number of bytes in the PPG samples report
0x12	0x00	-	Get the number of samples available in the output FIFO	Number of samples available in the FIFO.
0x12	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**
0x14	0x00	AlgoHub Input Frame 1 AlgoHub Input Frame 2 . . . . AlgoHub Input Frame N-1 AlgoHub Input Frame N **N : [1 to 25]**	Write AlgoHub input frame data to the AlgoHub input FIFO. **AlgoHub Input Frame (24 Bytes) Content:** PPG1: 3 byte PPG value: [LSB .. MSB] for WHRM HR Channel 1 PPG2: 3 byte PPG value: [LSB .. MSB] for WHRM HR Channel 2 PPG3: 3 byte PPG value: [LSB .. MSB] for SpO2 IR Channel PPG4 : 3 byte PPG value: [LSB .. MSB] for SpO3 Red Channel PPG5 : 3 byte PPG value: [LSB .. MSB] for N/A PPG6 : 3 byte PPG value: [LSB .. MSB] for N/A Accelerometer set to ±8g: ACCLX: 2-byte accel X value (2's complement .001g): [LSB MSB] ACCLY: 2-byte accel Y value (2's complement .001g): [LSB MSB] ACCLZ: 2-byte accel Z value (2's complement .001g): [LSB MSB]	-
0x46 for write	0x07	0x00	**Write** SpO_2 calibration coefficients (12 bytes comprised of three 32-bit signed values**, scaled up by 100,000**)	-

0x47 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 = -25.224999 (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**
0x46 for write	0x07	0x01	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.	-
0x47 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**)
0x46 for write	0x07	0x02	Write SpO_2 motion-detection threshold (signed 32-bit int, equal to 10^5 x milli-g threshold value)	-
0x47 for read	0x07	0x02	Read SpO_2 motion-detection threshold (signed 32-bit int, 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)
0x46 for write	0x07	0x04	Write the timeout duration for SpO2 measurement in seconds (1 byte).	-
0x47 for read	0x07	0x04	Read the timeout duration for SpO2 measurement in seconds. Default: 0x5A	SpO_2 algorithm timeout (8-bit unsigned)
0x46 for write	0x07	0x05	Write initial HR algorithm value (8-bit unsigned).	-
0x47 for read	0x07	0x05	Read initial HR algorithm value. Default: 0x3C	**Initial heart rate setting (8-bit unsigned)**
0x46 for write	0x07	0x0A	Set the algorithm operation mode (can be switched in runtime): 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 Data Collection	-
0x47 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
0x46 for write	0x07	0x0C	Write the Skin Contact Detection (SCD) algorithm enable 0x00: Disable 0x01: Enable (default)	-
0x47 for read	0x07	0x0C	Read the Skin Contact Detection (SCD) algorithm enable	0x00: Disable 0x01: Enable
0x46 for write	0x07	0x0E	Write SpO2 motion magnitude threshold in 0.001g. (16-bit unsigned)	-
0x47 for read	0x07	0x0E	Read SpO2 motion magnitude threshold.	motion magnitude threshold in 0.001g. (16-bit unsigned) Default: 0x0032 (0.05g)
0x44	0x07	0x00: Disable (CMD_DELAY = 120ms) [0x01,0x01]: Enable AlgoHub	Enable the AlgoHub, Wearable Algorithm Suite (WHRM+WSpO_2).	
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	-	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.		
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.	

3 I2C Algorithm Input and Output FIFO Format

The output format command, 0x10 0x01 is used to set AlgoHub output. AlgoHub is defined as a sensor within MAX32664C.

Table 3. Input FIFO: AlgoHub Input Sample Format

	DATA ITEM	# OF BYTES (MSB FIRST)	DESCRIPTION
	PPG1	**3**	**Green counts**
	PPG2	**3**	**Green2 counts (Can be set to PPG1 if only one Green channel is available)**
	PPG3	**3**	**IR LED counts**
	PPG4	**3**	**Red LED counts**
	PPG5	**3**	**N/A**
	PPG6	**3**	**Control Signal. Any 3 byte data. PPG1-4 can be routed to this input. An incrementing sequence serving as sample count is an option as well. See Table 4.**
	accelX	**2**	**Two's complement. LSB = 0.001g**
	accelY	**2**	**Two's complement. LSB = 0.001g**
	accelZ	**2**	**Two's complement. LSB = 0.001g**

Table 4. Output FIFO: AlgoHub Algorithm Report

DATA SOURCE	DATA ITEM	# OF BYTES (MSB FIRST)	DESCRIPTION
AlgoHub Wearable Suite Algorithm Data (24 Bytes)	Control Signal	3	Control Signal for AlgoHub Sensor. It is the control input channel data passed through AlgoHub data pipeline.
	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 last calculated heart rate
	HR confidence	1	Last calculated confidence level in %
	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: 0: Rest 1: Other 2: Walk 3: Run

			4: Bike
	R	2	1000x last calculated SpO_2 R value
	SpO_2 confidence	1	Last calculated SpO_2 confidence level in %
	SpO_2	2	10x last calculated SpO_2 %
	SpO_2 % 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.
	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)
	AlgoHub run status	1	0: Success 3: Init error 20: HR Run error 21: SCD run error 26: SpO2 run error 28: Invalid mode error 29: Invalid initial Hearth Rate Warning

4 Enabling the AlgoHub and Reading Processed Data

4.1 AlgoHub Algorithm Report

The table below show the sequence of commands for enabling the reading of the AlgoHub algorithm report.

Table 5. Host Commands—Read AlgoHub Algorithm Report

		#	HOST COMMAND (HEX)	COMMAND DESCRIPTION	RESPONSE (HEX)
START ALGOHUB		Host initializes the AlgoHub and starts the algorithm using following commands:			
		1.1	AA 02 00 (optional)	Read the operating mode	AB 00 00 application mode
		1.2	AA FF 03 (optional)	Read the AlgoHub version for MAX32664C (Major Minor Prefix)	AB 00 ZZ XX YY
		1.3	AA 10 00 01	Set the output FIFO mode to sensor data only	AB 00
		1.4	AA 44 07 01 01 (CMD_DELAY = 120 ms)	Enable AlgoHub sensor.	AB 00
READING SAMPLES REPORT IN OUTPUT FIFO		Host provides PPG, accel data and reads samples			
		2.1	AA 14 00 [PPG1 PPG2 PPG3 PPG4 PPG5 PPG6 ACCLX ACCLY ACCLZ] …………………… …………………… …….[PPG1 PPG2 PPG3 PPG4 PPG5 PPG6 ACCLX ACCLY ACCLZ] **Up to 25 frames**	Write PPG, accel data to the algorithm input FIFO. PPG1: 3 byte PPG value: [LSB .. MSB] PPG2: 3 byte PPG value: [LSB .. MSB] PPG3: 3 byte PPG value: [LSB .. MSB] PPG4: 3 byte PPG value: [LSB .. MSB] PPG5: 3 byte PPG value: [LSB .. MSB] PPG6: 3 byte PPG value: [LSB .. MSB] Accelerometer set to ±8g: ACCLX: 2 byte accel X value (2’s complement .001g): [LSB MSB] ACCLY: 2 byte accel Y value (2’s complement .001g): [LSB MSB] ACCLZ: 2 byte accel Z value (2’s complement .001g): [LSB MSB]	
		Delay for AlgoHub Processing – See “AlgoHub Response Duration”			
		2.2	AA 00 00	Read the AlgoHub 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: AlgoHub busy (DevBusy) Bit 7: Reserved If DataRdyInt is set, proceed to the next step.	AB 00 08
		2.3	AA 12 00	Get the number of samples (nn) in the FIFO.	AB 00 nn
			AA 12 01	Read the data stored in the FIFO; nn samples will be included. The format of the samples report is shown in the normal algorithm report table.	AB 00 data_for_nn_samples
STOP		Host ends the procedure:			
		3.1	AA 44 07 00 (CMD_DELAY = 120ms)	Disable AlgoHub sensor.	AB 00

4.2 SpO_2 Data Collection Report for 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 SpO2 Measurement] 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)

The table below show the sequence of commands for enabling the reading of the normal algorithm report.

Table 6. 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 02 00 (optional)	Read the operating mode	AB 00 00 application mode
	1.2	AA FF 03 (optional)	Read the AlgoHub version for MAX32664C (ZZ.x.y)	AB 00 ZZ XX YY
	1.3	AA 10 00 01	Set the output FIFO mode to sensor data only	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 46 07 0A 06	Set the mode to SpO_2 Calibration Data Collection mode	AB 00
	1.10	AA 46 07 0E 00 32	Set Spo2 motion magnitude threshold to 0x0032, 0.05G	AB 00
	1.11	AA 44 07 01 01 (CMD_DELAY = 120 ms)	Enable the algoHub operation.	AB 00
READING SAMPLES REPORT IN OUTPUT	Host provides PPG, accel data and reads			
	2.1	AA 14 00 [PPG1 PPG2 PPG3 PPG4 PPG5 PPG6 ACCLX ACCLY ACCLZ] ……………………… ……………………… …….[PPG1 PPG2 PPG3 PPG4 PPG5	Write PPG, accel data to the algorithm input FIFO. PPG1: 3 byte PPG value: [LSB .. MSB] PPG2: 3 byte PPG value: [LSB .. MSB] PPG3: 3 byte PPG value: [LSB .. MSB] PPG4: 3 byte PPG value: [LSB .. MSB] PPG5: 3 byte PPG value: [LSB .. MSB] PPG6: 3 byte PPG value: [LSB .. MSB] Accelerometer set to ±8g:	

		PPG6 ACCLX ACCLY ACCLZ] **Up to 25 frames**	ACCLX: 2 byte accel X value (2's complement .001g): [LSB MSB] ACCLY: 2 byte accel Y value (2's complement .001g): [LSB MSB] ACCLZ: 2 byte accel Z value (2's complement .001g): [LSB MSB]	
		Delay for AlgoHub processing – see "AlgoHub Response Duration"		
	2.2	AA 00 00	Read the AlgoHub 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: AlgoHub busy (DevBusy) Bit 7: Reserved If DataRdyInt is set, proceed to the next step.	AB 00 08
	2.3	AA 12 00	Get the number of samples (nn) in the FIFO.	AB 00 nn
		AA 12 01	Read the data stored in the FIFO; nn samples will be included. The format of the samples report is shown in the normal algorithm report table.	AB 00 data_for_ nn_samples
STOP	Host ends the procedure:			
	3.1	AA 44 07 00 (CMD_DELAY = 120ms)	Disable the algorithm.	AB 00

The tables below show the sequence of commands for enabling the reading of the extended algorithm report.

4.3 AlgoHub Response Duration

AlgoHub response duration depends on number of frames pushed to AlgoHub input fifo in a single transaction. Wait time to ask for status after data feed command is 5ms. Duration to read AlgoHub report is based on number of input frames in transaction which is limited to 25 frames. Duration can be calculated as:

Wait duration for AlgoHub Results = 4ms + 2ms * (# input frames in transaction)

For example for 25 input samples, wait time to ask for associated 25 AlgoHub results is 4 + 25*2 = 54ms.

5 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, and authorization bytes, might 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 7. Annotated I^{2}C Trace for Flashing the Application

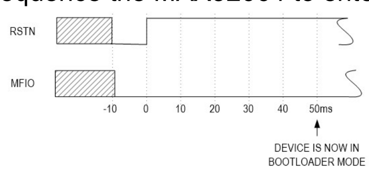
<table>
<tr><th>HOST COMMAND</th><th>COMMAND DESCRIPTION</th><th>READ MAX32664C RESPONSE</th><th>RESPONSE DESCRIPTION</th></tr>
<tr><td colspan="4">Sequence the MAX32664 to enter bootloader mode. *


Figure 7. Sequence to enter bootloader mode.</td></tr>
<tr><td>0xAA 0x01 0x00 0x08*</td><td>Set mode to 0x08 for bootloader mode.</td><td>0xAB 0x00</td><td>No error.</td></tr>
<tr><td>0xAA 0x02 0x00</td><td>Read mode.</td><td>0xAB 0x00 0x08</td><td>No error. Mode is bootloader.</td></tr>
<tr><td>0xAA 0x81 0x00</td><td>Read bootloader firmware version.</td><td>0xAB 0x00 0x03 0xYY 0xZZ</td><td>No error. Version is 3.YY.ZZ.</td></tr>
<tr><td>0xAA 0x81 0x01</td><td>Read bootloader page size.</td><td>0xAB 0x00 0x20 0x00</td><td>No error. Page size is 8192.</td></tr>
<tr><td>0xAA 0x80 0x02 0x00 0x1A</td><td>Bootloader flash. Set the “number of pages” to 31 based on the value at byte 0x44 from the application .msbl file.</td><td>0xAB 0x00</td><td>No error.</td></tr>
<tr><td colspan="4">00000044 | 02 ed 27 af 1a 00 00 20 04 00 00 00 c2 31 90 2c
Figure 8. Page number byte 0x44 from the .msbl file.</td></tr>
<tr><td>0xAA 0x80 0x00 0x1A 0xDB 0xE5 0x0D 0x90 0x79 0xE6 0xC6 0x13 0x87 0xB9*</td><td>Bootloader flash. Set the initialization vector bytes to the 0x28 to 0x32 values from the .msbl file.</td><td>0xAB 0x00</td><td>No error.</td></tr>
</table>

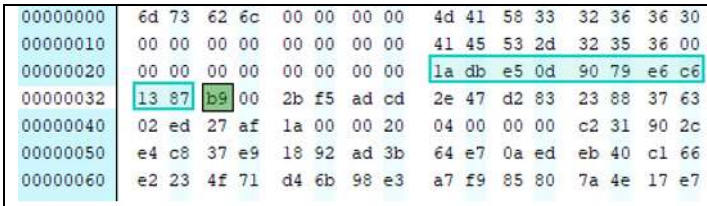


Figure 9. Initialization vector bytes 0x28 to 0x32 from the .msbl file.

0xAA 0x80 0x01 0x2B 0xF5 0xAD 0xCD 0x2E 0x47 0xD2 0x83 0x23 0x88 0x37 0x62 0x02 0xED 0x27 0xAF*	Bootloader flash. Set the authentication bytes to the 0x34 to 0x43 values from the .msbl file.	0xAB 0x00	No error.

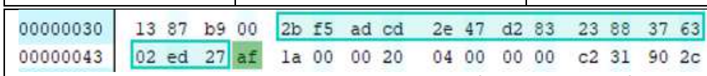


Figure 10. Authentication bytes 0x34 to 0x43 from the .msbl file.

0xAA 0x80 0x03*	Bootloader flash. Erase application.	0xAB 0x00	No error.
0xAA 0x80 0x04 0xC2 0x31 0x90 ... 0x9E 0x6A 0x0E*	Bootloader flash. Send page bytes 0x4C to 0x205B from the .msbl file.	0xAB 0x00	No error.

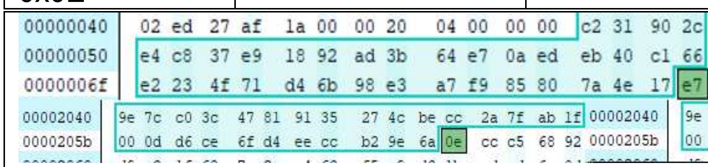


Figure 11. Send page bytes 0x4C to 0x205B from the .msbl file.

0xAA 0x80 0x04 0xCC 0xC5 0x68 ... 0xF7 0xD6 0x4C*	Bootloader flash. Send page bytes 0x205C to 0x406B from the .msbl file.	0xAB 0x00	No error.
0xAA 0x80 0x04 0x2E 0xA6 0x13 ... 0x84 0xF7 0xCF*	Bootloader flash. Send page bytes 0x406C to 0x607B from the .msbl file.	0xAB 0x00	No error.
0xAA 0x80 0x04 0xD7 0x1F 0x7F ... 0x55 0xAB 0xB8*	Bootloader flash. Send page bytes 0x607C to 0x808B from the .msbl file.	0xAB 0x00	No error.
0xAA 0x80 0x04 0xC4 0x63 0x2B ...	Bootloader flash. Send page bytes	0xAB 0x00	No error.

0x48 0xCD 0x52*	0x808C to 0xA09B from the .msbl file.		
0xAA 0x80 0x04 0x89 0x33 0x22 ... 0x31 0xAD 0x19*	Bootloader flash. Send page bytes 0xA09C to 0xC0AB from the .msbl file.	0xAB 0x00	No error.
0xAA 0x80 0x04 0x8B 0x97 0x18 ... 0xF3 0xCF 0x90*	Bootloader flash. Send page bytes 0xC0AC to 0xE0BB from the .msbl file.	0xAB 0x00	No error.
0xAA 0x80 0x04 0xD0 0x78 0x38 ... 0x1F 0x7F 0x92*	Bootloader flash. Send page bytes 0xE0BC to 0x100CB from the .msbl file.	0xAB 0x00	No error.
0xAA 0x80 0x04 0xB1 0xE9 0x8F ... 0xF4 0x23 0xD8*	Bootloader flash. Send page bytes 0x100CC to 0x120DB from the .msbl file.	0xAB 0x00	No error.
0xAA 0x80 0x04 0xF8 0xC6 0x83 ... 0xF4 0x24 0xE2*	Bootloader flash. Send page bytes 0x120DC to 0x140EB from the .msbl file.	0xAB 0x00	No error.
0xAA 0x80 0x04 0x1F 0x4F 0x5C ... 0xCC 0x2E 0xCD*	Bootloader flash. Send page bytes 0x140EC to 0x160FB from the .msbl file.	0xAB 0x00	No error.
0xAA 0x80 0x04 0x40 0x1F 0x03 ... 0x26 0xEB 0xB9*	Bootloader flash. Send page bytes 0x160FC to 0x1810B from the .msbl file.	0xAB 0x00	No error.
0xAA 0x80 0x04 0x2F 0xD9 0xB2 ... 0xEE 0x2A 0x8F*	Bootloader flash. Send page bytes 0x1810C to 0x1A11B from the .msbl file.	0xAB 0x00	No error.
0xAA 0x80 0x04 0x51 0x32 0x47 ... 0x41 0xE6 0x47*	Bootloader flash. Send page bytes 0x1A11C to 0x1C12B from the .msbl file.	0xAB 0x00	No error.
0xAA 0x80 0x04 0x22 0xA6 0x06 ...	Bootloader flash. Send page bytes 0x1C12C to	0xAB 0x00	No error.

0x2A 0xCB 0x44*	0x1E13B from the .msbl file.		
0xAA 0x80 0x04 0x68 0x9E 0x1E ... 0x53 0x89 0xE8*	Bootloader flash. Send page bytes 0x1E13C to 0x2014B from the .msbl file.	0xAB 0x00	No error.
0xAA 0x80 0x04 0x5F 0x1A 0x6A ... 0x14 0xA1 0x85*	Bootloader flash. Send page bytes 0x2014C to 0x2215B from the .msbl file.	0xAB 0x00	No error.
0xAA 0x80 0x04 0xE8 0xDE 0xC9 ... 0x81 0xD8 0x00*	Bootloader flash. Send page bytes 0x2215C to 0x2416B from the .msbl file.	0xAB 0x00	No error.
0xAA 0x80 0x04 0x0E 0xD2 0x16 ... 0x8D 0x69 0xEE*	Bootloader flash. Send page bytes 0x2416C to 0x2617B from the .msbl file.	0xAB 0x00	No error.
0xAA 0x80 0x04 0x2F 0x4B 0x38 ... 0x02 0xA7 0xDC*	Bootloader flash. Send page bytes 0x2617C to 0x2818B from the .msbl file.	0xAB 0x00	No error.
0xAA 0x80 0x04 0xA5 0xFE 0xFD ... 0xE3 0x38 0x89*	Bootloader flash. Send page bytes 0x2818C to 0x2A19B from the .msbl file.	0xAB 0x00	No error.
0xAA 0x80 0x04 0x52 0x88 0x9A ... 0xF0 0xC5 0x9D*	Bootloader flash. Send page bytes 0x2A19C to 0x2C1AB from the .msbl file.	0xAB 0x00	No error.
0xAA 0x80 0x04 0xA3 0xA6 0x92 ... 0xA0 0x4D 0xBE*	Bootloader flash. Send page bytes 0x2C1AC to 0x2E1BB from the .msbl file.	0xAB 0x00	No error.
0xAA 0x80 0x04 0x47 0x09 0x75 ... 0x24 0xBD 0x3D*	Bootloader flash. Send page bytes 0x2E1BC to 0x301CB from the .msbl file.	0xAB 0x00	No error.
0xAA 0x80 0x04 0x44 0xEC 0xE6 ...	Bootloader flash. Send page bytes 0x301CC to	0xAB 0x00	No error.

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

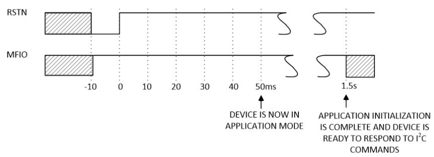


Figure 12. Sequence to enter application mode.

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

6 Heart Rate Algorithm Perfomance

Wearable Heart Rate Monitoring – WHRM		
Category	**Features**	**Specifications**
Algorithm	Measurement principle:	Optical PPG signal from wrist, 3D Axis Accelerometer
	Measurement range:	HR: [30 240] BPM Cadence (steps per minute): [90 360]
	Measurement accuracy:	HR: Accuracy Definition -> within +/-10% error band vs. reference (chest strap) Resting: 94 – 100% Walking: 93 - 99% Biking: 91 - 97% Running: 91 - 97% Daily Life: 90 - 100% Step counting: Accuracy Definition: 100% – (Absolute Percent Error) Treadmill walking: 89 – 95% Treadmill running: 86 – 92% Outdoor walking: 80 – 90% Activity Classification: Accuracy Definition: 100% – (Absolute Percent Error) Rest: 87 - 93% Treadmill walking: 93 - 99% Treadmill running: 90 - 95% Outdoor walking: 91 - 97% Outdoor biking: 80 - 90% Energy consumption Kcal: calculated according to ACSM & ADA
	Reference measurement device:	HR: ECG based chest strap IBI: ECG based chest strap (sampling rate min 1kHz)
	Average response time	25Hz, first response time 15sec
	Inputs:	Single/Multiple Channel PPG signal 3-axis accelerometer signals

	Built-in features:	Activity Classifier Built-in Step Counter Motion compensation of PPG for accurate HR estimation Inter-beat interval estimator Energy expenditure estimation
Measurement Positions	Wrist, Ear, Finger, Chest, Abdomen	Sports and daily life activities
Sensor & Signal Requirements	LED requirements:	Please refer to "Reference Design Document" for details
	Perfusion index range:	Minimum AC to calculate HR is 20nA with average 0.8% PI
	Sampling rate:	25 Hz
Calibration	Calibration:	Algorithm activity classifier is tuned for the sensor placement on wrist, a calibration might be required to train algorithm to improve its performance for another body location.

7 SpO_2 Algorithm Perfomance

SpO_2 on Wrist		
Category	**Features**	**Specifications**
Algorithm	Measurement principle:	Optical PPG signal from wrist
	Measurement range:	70 – 100% SpO2
	Measurement accuracy:	RMSE ≤ 3.5% as required by FDA for reflective mode pulse oximeters
	Clinical test:	Certified calibration lab
	Measurement time:	30 – 60 sec, on-demand operations, one shot
	Inputs:	Red and Infrared PPG signals 3-axis accelerometer signals
	Built-in features:	Precise motion detector Automatic AFE setting adjustment for optimum PPG quality Signal conditioning Signal selection according to signal quality for discarding noisy signal portions Adjustable time-out duration Adjustable confidence threshold
Measurement Positions	Standing:	Arm is kept horizontal at the level of heart; palm is facing the floor
	Sitting	Arms are placed on a table Arms are crossed, arm with wristwatch is above the other arm
	Lying down:	Arms are horizontal
Sensor & Signal Requirements	LED requirements:	Center wavelength shift ≤ ±5nm LED full width at half maximum (FWHM) ≤ 20nm
	Perfusion index range:	PI ≥ 0.05%
	Sampling rate:	25 Hz
	Optical layout design:	Please refer to "Design Guide for SpO2 Measurement"
Calibration	Calibration lab:	Lab calibration is required for wearable's finished industrial design. Please refer to lab calibration procedure guide "Design Guide for SpO2 Measurement"

8 MAX32664C Power Consumption Estimate

The AlgoHub 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 AlgoHub to save power when the processor is idle or there is no need for any processing.

Table 8. Comparison of Active and Deep Sleep Power

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

9 References

MAXREFDES103# website files:

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

Frequently Asked Questions: Maxim Support Center

Validation and Performance of a Wearable Heart rate Monitoring Algorithm

MAX32664 website: C-keyed .msbl for MAX32664C:

MAX32664 Design Resources Website

Revision History

REVISION NUMBER	REVISION DATE	DESCRIPTION	PAGES CHANGED
0	02/21	Initial release	—