MAX32674C AlgoHub SensorHub – Developers Guide

Rev 0.3; 08/21

Abstract

The MAX32674C may be commanded to be an algorithm hub or a sensor 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 MAX32674C.

Table of Contents

List of Figures

List of Tables

Introduction

MAX32674C (AlgoHubSensorHub) may be used as either algorithm hub (AlgoHub) or a sensor hub (SensorHub) which provides the following innovative features:

- Biometric heart rate, SpO2 algorithm (Wearable Algorithm Suite (WAS)) was developed by a team of machine learning, data scientist, algorithm experts.
- OTA update compatibility allows the AlgoHubSensorHub to receive the latest algorithm .msbl file.
- Low-powered AlgoHubSensorHub 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.

NOTE: The instructions in this document are compatible with the MAX32674C firmware .msbl version 50.2.4+ (MAX86176). The initial stock of MAXREFDES104 comes populated with MAX32670 which has been programmed with the bootloader so that it behaves as a MAX32674C; The orderable part for MAX32670 does not include with the bootloader and customers should order the MAX32674C part which includes the bootloader. The MAX32674C part does not include the application .msbl algorithm - The latest .msbl must be OTA or flashed to the MAX32674C.

When using the MAXREFDES104, the following three software and firmware must all be updated using the latest and same software release package. Please refer to the MAXREFDES104 User Guide for detailed software, firmware installation procedure.

1. Flash the micro board **.bin** file using drag & drop to the DAPLINK folder.
2. Flash the algorithm .**msbl** file to MAX32674C using the provided batch file.
3. Install the MAXREFDE104 PC GUI using the **.msi** file.

1 AlgoHub, SensorHub Configurations

The hardware and firmware for the MAX32674C may be configured in the AlgoHub or SensorHub configurations.

1.1 AlgoHub Configuration

In the AlgoHub configuration, the MAX32666 host is connected to the MAX86176 AFE and the LIS2DS accelerometer. The host drivers are responsible for the setting of the AFE, accel registers and for retrieving the PPG, accel data. After the MAX32674C has been initialized to AlgoHub mode via I2C commands, the host pipes in the PPG, accel data to the MAX32674C algorithms and the MAX32674C provides heart rate and SpO2 data to the host. The MAX32674C is power optimized to deep sleep when it is idle. By using the combined ECG, PPG MAX86176 IC, the host processor can retrieve time synchronized ECG and PPG data.

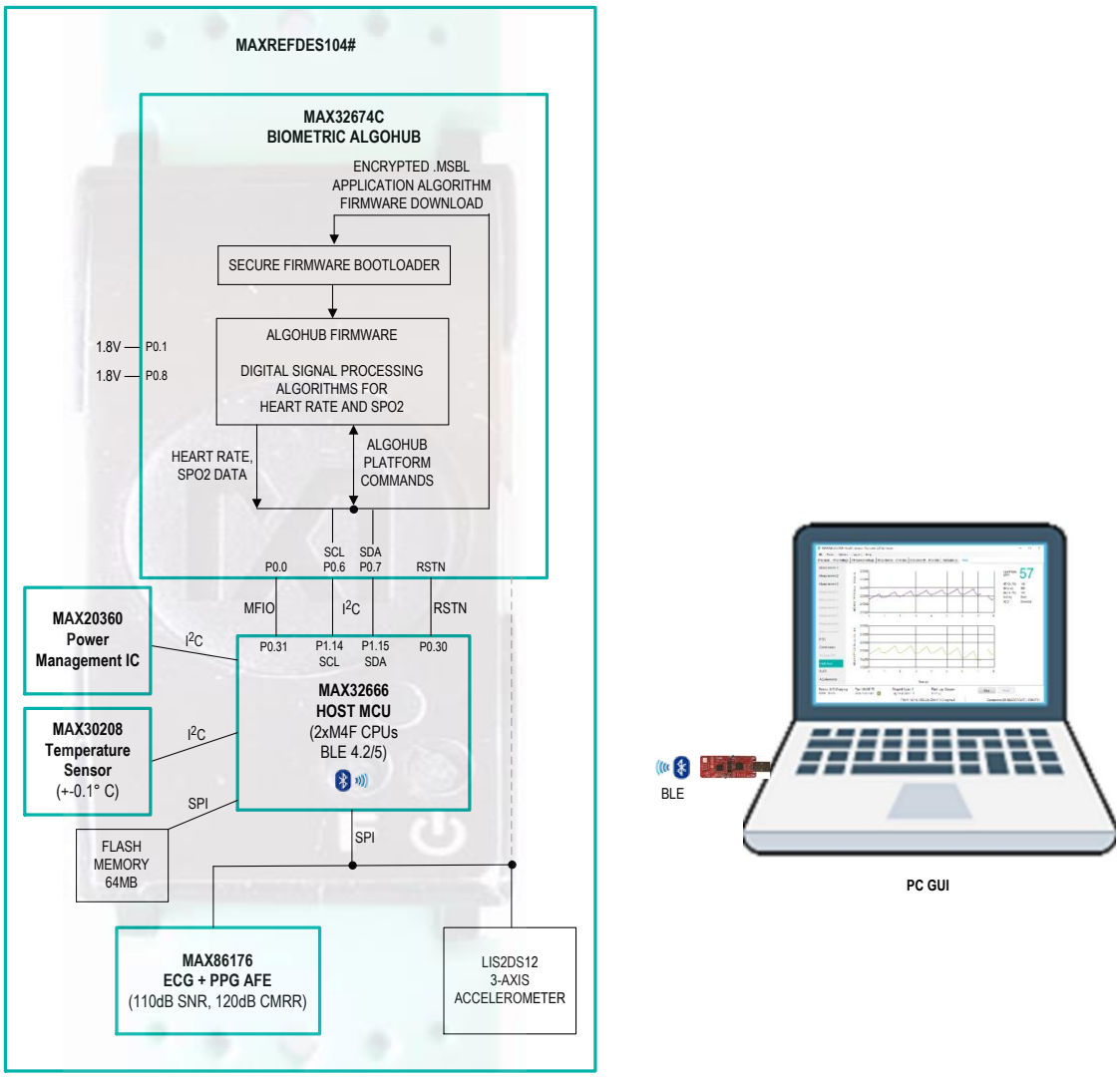


Figure 1. AlgoHub Architecture diagram for MAXREFDES104# health-sensing platform.

1.2 SensorHub Configuration

In the SensorHub configuration, the MAX32674C is connected to the MAX86176 AFE and the LIS2DS accelerometer. After the MAX32674C has been initialized to SensorHub mode via I2C commands, then the MAX32674C drivers will handle the AFE, accel register settings and retrieve the PPG, accel data. The MAX32674C algorithms processes the raw data to provide heart rate and SpO2 data to the host. The MAX32674C is power optimized to deep sleep when it is idle.

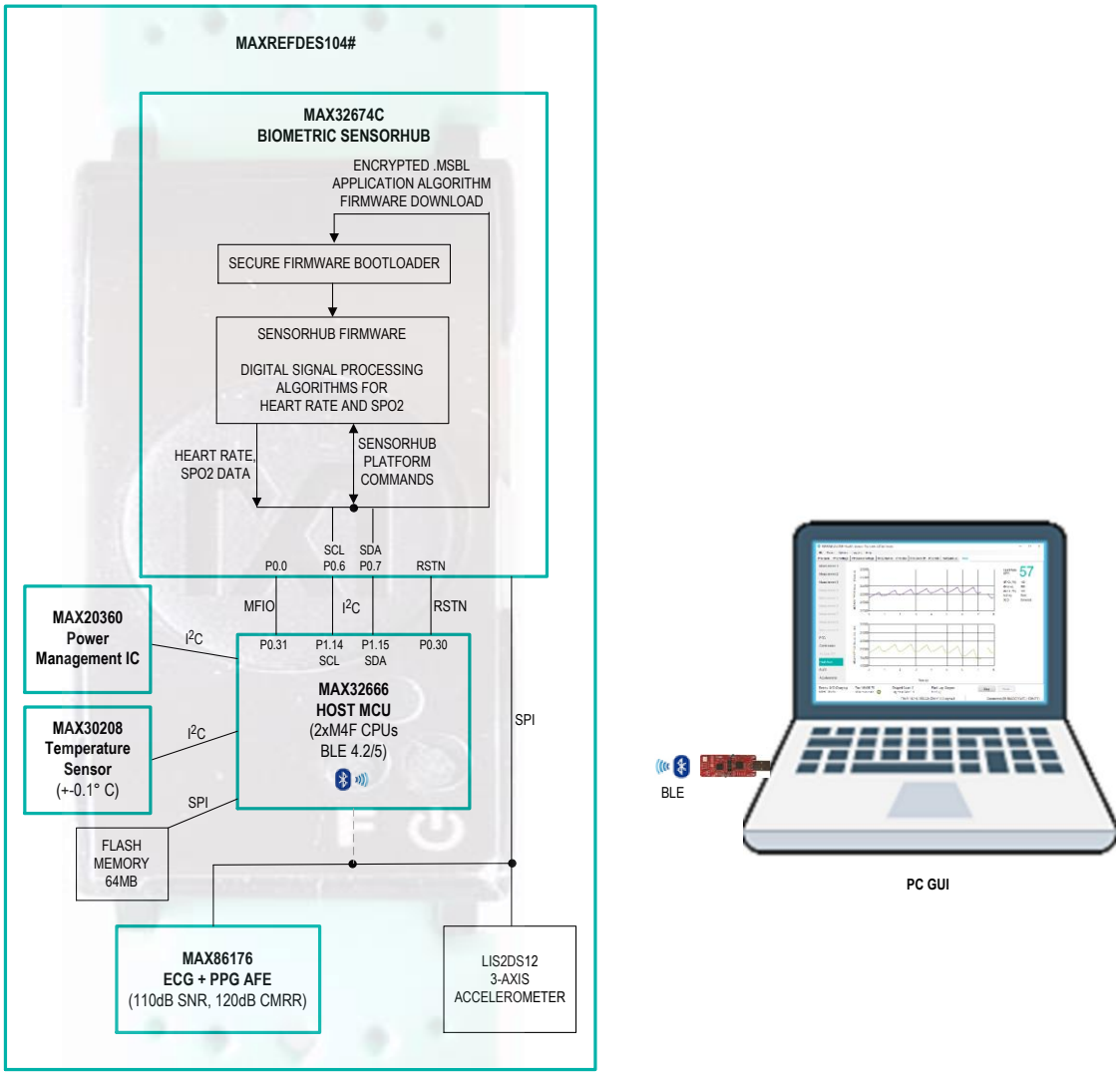


Figure 2. SensorHub Architecture diagram for MAXREFDES104# health-sensing platform.

1.3 When to use AlgoHub configuration or SensorHub configuration

The AlgoHub configuration should be used when the system requirements require that ECG and PPG run simultaneously or when it is required to have accel, ECG and PPG data time synchronized. The AlgoHub mode requires more host integration effort for AFE, accel register settings and for piping in the PPG and accel data to the MAX32674C and for creating the host code which responds to the AlgoHub algorithm requests to update the PPG register settings of the AFE. The total system power will be more for AlgoHub configuration since there will be more host to SensorHub communications and the host must handle the dynamic management of the PPG AFE and accel sensors.

The SensorHub configuration should be used when the system requirements do not require that ECG and PPG run simultaneously – ECG and PPG are run in stand-alone modes. The SensorHub mode requires less host integration effort since the MAX32674C includes the drivers to retrieve the PPG, accel data from the MAX86175 and LIS2DS12. The total system power will be less for SensorHub configuration since there will be less host to SensorHub communications and the dynamic management of the PPG AFE and the accel is off-loaded to the power optimized MAX32674C. ECG is currently not demonstrated in SensorHub configuration.

2 Reset

2.1 Reset to Bootloader or Application Mode

Two GPIO pins from the host are needed to control the RSTN and multifunction input/output (MFIO) pins.

To enter Bootloader mode:

- Set the RSTN pin low.
- While RSTN is low, set the MFIO pin to low and set either SWDCLK or UART0_RX to high. (Do this 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 AlgoHubSensorHub is in Bootloader mode.

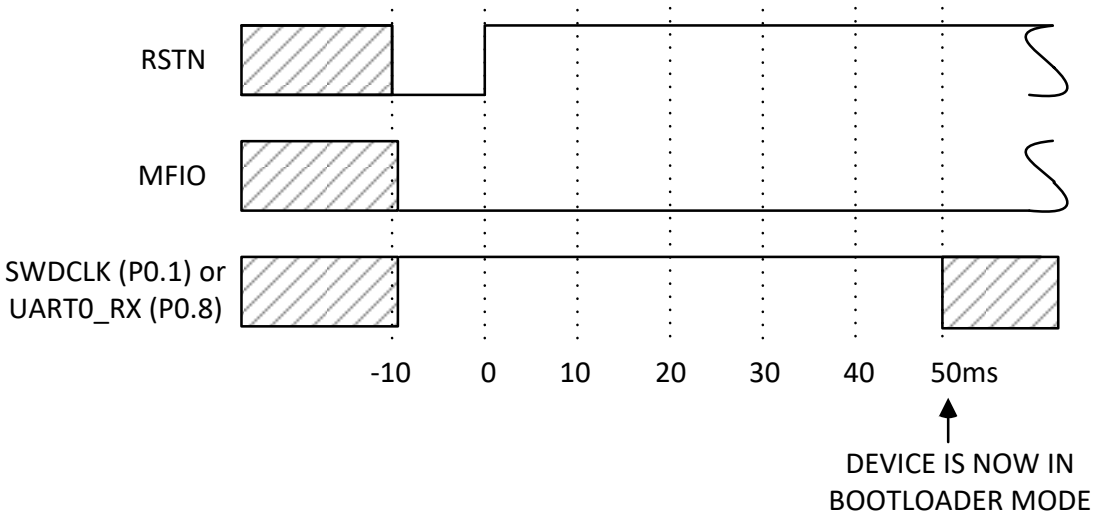


Figure 3 Entering bootloader mode using the RSTN pin and the MFIO GPIO pin.

To enter Application mode:

- Set the RSTN pin low.
- While RSTN is low, set the MFIO pin to high and set either SWDCLK or UART0_RX to high. (Do this 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 AlgoHubSensorHub 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.
- Set the ACCEL_CS_N pin to low.

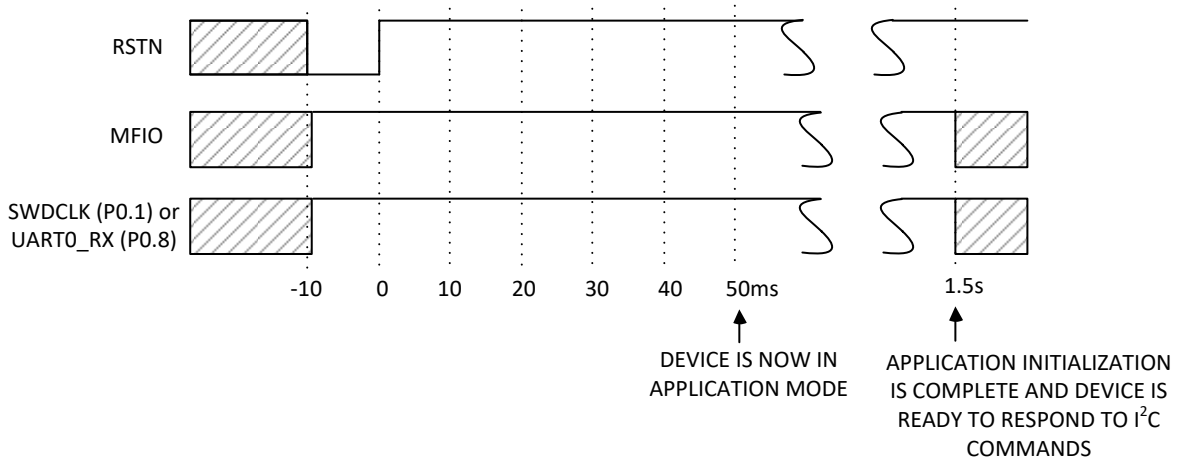


Figure 4 Entering application mode using the RSTN pin and MFIO pin.

To enter Application mode by timing out from Bootloader mode:

- Set the RSTN pin low.
- While RSTN is low, set the MFIO pin to low and set either SWDCLK or UART0_RX to high. (Do this 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 AlgoHubSensorHub is in Bootloader mode.
- If no I2C commands are sent to the AlgoHubSensorHub within the next 1s, then the AlgoHubSensorHub will automatically switch to application mode.

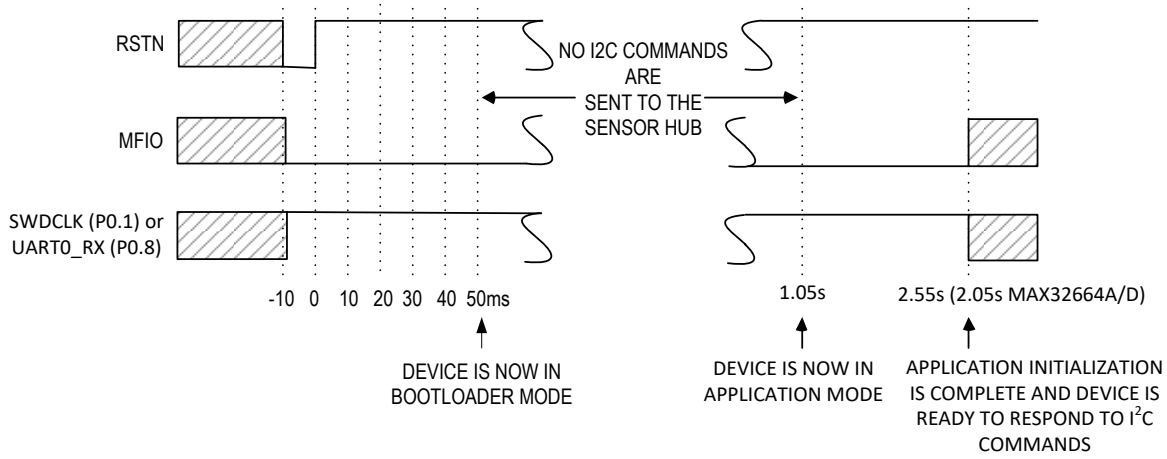


Figure 5 Entering application mode by timing out from Bootloader mode.

2.2 Reset to an Unsupported State

Do not set P0.1 (ACC_INT) and P0.8 (ACC_CS) both low when powering up or exiting reset – doing so will cause the MAX32674C to enter an unsupported state. Set either or both P0.1, P0.8 to high when powering up or exiting reset.

3 AlgoHubSensorHub Handshaking: I2C and MFIO pin

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

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

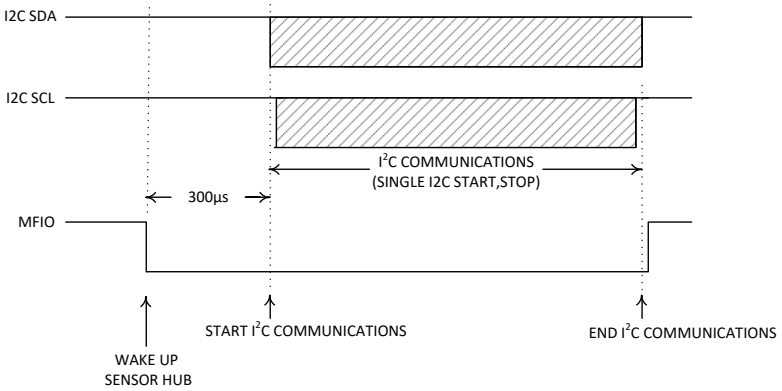


Figure 6. I2C and MFIO pin: host uses MFIO for enabling host communications.

4 AlgoHubSensorHub I2C Communications

A host uses the I2C bus to communicate with the AlgoHubSensorHub (slave) using a series of commands. The default CMD_DELAY is 2ms. The I2C data rate 400Kbps has been tested; the 1000Kbps data rate has not been tested.

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

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)
```

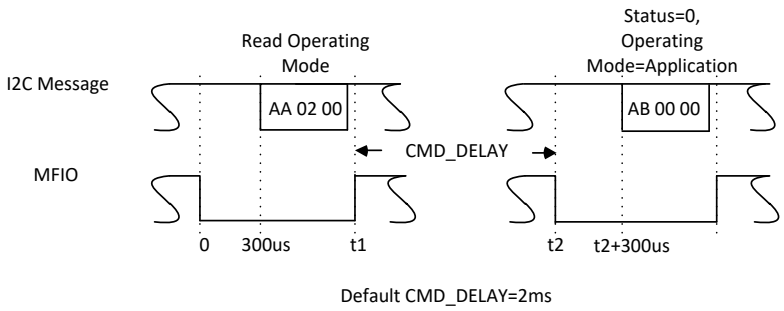


Figure 7. Example I²C, MFIO example command response.

4.1 MAX32674C I2C Bit Transfer Process.

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

The defined bit transfer process is described below. It is recommended that I2C GPIO 'bit-bang' software be implemented on the host if the host MCU I2C hardware/HAL is not compatible with AlgoHubSensorHub protocol.

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

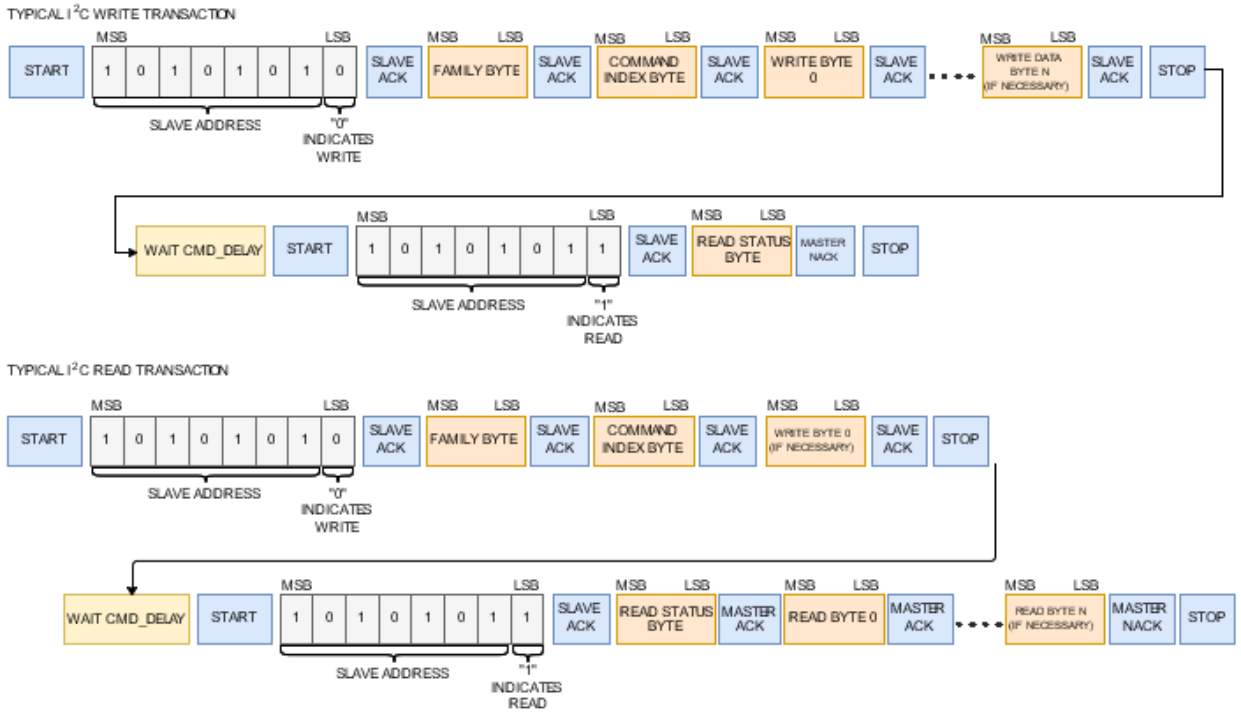


Figure 8. I2C Write/Read data transfer from host microcontroller.

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

4.2 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	Application mode: 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 MAX32674C.
0x82	Bootloader authorization error. Verify that the keyed .msbl file is compatible with MAX32674C.
0x83	Bootloader detected that the application is not valid.
0x84	Bootloader: application program was not erased before trying to flash a new one.
0xAA	Bootloader mode: The write transaction was successful.
0xAB	Bootloader: partial page data received; there is still more page data.
0xFE	Device is busy. Try again. Increase the delay before the command and increase the CMD_DELAY.
0Xff	Unknown error. The AlgoHubSensorHub 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. AlgoHubSensorHub 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

4.3 AlgoHub Mode: PPG, Accel Sampling Rates Settings

The AlgoHub algorithm accepts 25 Hz data.

The host should configure the PPG sampling rates according to the AFE request when using AEC mode.

If AEC mode is not used, then the host should configure the PPG sampling rates to the following:

- PPG 25 Hz sampling rate, sample averaging of 1.
- PPG 100 Hz sampling rate, sample averaging of 4. Host should switch to this rate when running, jogging activity has been detected. Host should use one way hysteresis to filter switching back to 25 Hz, sample average of 1.

The host should configure the Accel sampling rates to the following:

- 25 Hz sampling rate (do not use LIS2DS12 low precision setting - it is too noisy). If 25 Hz is not available, the host should down-sample the higher frequency to 25 Hz with linear interpolation.

4.4 Down-Sampling Host Accel Data

Example code for down-sampling may be found on the web:
https://stackoverflow.com/questions/31836598/subsampling-an-array-of-numbers
To test, down-sample 5Hz to 4 Hz:

```
o[0] = i[0]
o[1] = .75*i[1] + .25*i[2]
o[2] = .5*i[2] + .5*i[3]
o[3] = .25*i[3] + .75*i[4] ….
```

4.5 AlgoHubSensorHub I2C Command Response Definitions

The table below defines the I2C command response message protocol for the AlgoHubSensorHub. (not listed in this table is the first byte of the response is always the Read Status byte)

Table 2. AlgoHubSensorHub I2C Command Response Definitions

FAMILY BYTE	INDEX BYTE	WRITE BYTE(S)	DESCRIPTION	RESPONSE BYTES
0x00	0x00	-	Read sensor AlgoHubSensorHub 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 = AlgoHubSensorHub output FIFO overflowed, data lost. **FifoInOvrInt[5]**: 0 = No FIFO overflow; 1 = AlgoHubSensorHub Input FIFO overflowed, data lost.
0x01	0x00	**0x00**: Exit bootloader mode, enter application mode. (CMD_DELAY = 1.5s) **0x01**: Shutdown the MAX32674C. Restart by power cycling or pulsing RSTN (application mode)	Set the device operating mode.	-

		0x08: Enter bootloader mode (if commanded within the first 20ms afer reset)		
0x02	0x00	-	Read the device operating mode.	0x00: Application operating mode. 0x02: Reset. 0x08: Bootloader operating mode.
0x10	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 AlgoHubSensorHub.	
0x10	0x01	**0x01** to **0xFF**: Sensor Hub Interrupt Threshold for FIFO.	Set the threshold for the FIFO interrupt bit.The bit DataRdyInt of the AlgoHubSensorHub status byte is set when this threshold is reached.	-
0x11	0x01	-	Read the threshold for the FIFO interrupt bit.The bit DataRdyInt of the AlgoHubSensorHub status byte is set when this threshold is reached.	**0x01** to **0xFF**: AlgoHubSensorHub Interrupt Threshold for FIFO.
able0x10	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 report period (e.g., a value of 25 means a samples report is generated once every 25 samples).	**0x01** (default) to **0xFF:** lsb is 40ms. N, where a samples report is

				generated once every N samples.
0x10	0x03	**0x02** to **0xFF**: New I2C address (8-bit I2C write address)	Change I2C address of the MAX32674C	-
0x10	0x04	**0x00** to **0xFF**: Counter	Set the AlgoHubSensorHub counter.	-
0x11	0x04	-	Read the AlgoHubSensorHub 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 (CMD_ DELAY = 5ms)	-	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 AlgoHub	0x00	AlgoHub Input Frame 1 AlgoHub Input Frame 2 . . . . AlgoHub Input Frame N-1 AlgoHub Input Frame N **N : [1 to 25]** CMD_DELAY = 16ms	Write AlgoHub input frame data to the AlgoHub input FIFO. **AlgoHub Input Frame (24 Bytes) Content:** PPG1: 3 byte PPG value: [MSB .. LSB] for WHRM HR Channel 1 PPG2: 3 byte PPG value: [MSB .. LSB] for WHRM HR Channel 2 PPG3: 3 byte PPG value: [MSB .. LSB]for SpO2 IR Channel PPG4 : 3 byte PPG value: [MSB .. LSB] for SpO2 Red Channel PPG5 : 3 byte PPG value:[MSB .. LSB] for N/A PPG6 : 3 byte PPG value: [MSB .. LSB] for N/A Accelerometer set to ±8g: ACCLX: 2-byte accel X value (2's complement .001g): [MSB LSB] ACCLY: 2-byte accel Y value (2's complement .001g): [MSB LSB]	mm nn mmnn is the ppg, accel number of bytes received (16-bit unsigned)s

			ACCLZ: 2-byte accel Z value (2's complement .001g): [MSB LSB]	
0x42 Sensor Hub	0x04		Read the attributes of the accelerometer.	Number of bytes in a word for this sensor, Number of registers available for this sensor.
0x42 Sensor Hub	0x06		Read the attributes of the MAX86176.	Number of bytes in a word for this sensor, Number of registers available for this sensor.
0x40 Sensor Hub	0x04	[reg_addr] [reg value]	Write a value to a writable accelerometer sensor register.	
0x41 Sensor Hub	0x04	[reg_addr]	Read accelerometer sensor register.	Register value (byte)
0x40 Sensor Hub	0x06	[reg_addr] [reg value]	Write a value to a writable MAX86176 register.	
0x41 Sensor Hub	0x06	[reg_addr]	Read the value of a MAX86176 register.	Register value (byte)
0x44 Sensor Hub	0x04	**0x00, 0x00:** Disable sensor hub accelerometer , CMD_DELAY = 50ms **0x01, 0x00:** Enable sensor hub accelerometer CMD_DELAY = 50ms	Write the enable/disable the accelerometer.	-
0x44 Sensor Hub	0x06	**0x00**: Disable, CMD_DELAY = 200ms **0x01, 0x00**: Enable Raw Data, CMD_DELAY = 500ms	Write the enable/disable the MAX86176 AFE.	-
0x44 AlgoHub	0x07	**0x00, 0x01:** Disable WAS AlgoHub, external sensor input**,** CMD_DELAY = 200ms. **0x01, 0x01:** Enable WAS AlgoHub, enable external sensor input .	Enable/Disable Wearable Algorithm Suite AlgoHub.	-

		CMD_DELAY = 500ms		
0x46 for write. AlgoHub	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, scaled up by 100,000)	-
0x47 for read. AlgoHub	0x07	0x00	Read SpO_2 calibration coefficients (12 bytes comprised of three 32-bit signed values, scaled up by 100,000) Default: 0xFFF9482C FFE3792B 00A8E9A4 A = -4.4027606 (0xFFF9482C) B = -18.695252 (0x FFE3792) C = 110.6986082 (0x00A8E9A4)	32-bit signed integer A, 32-bit signed integer B, 32-bit signed integer C Values scaled up by 100,000
0x46 for write. AlgoHub	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.	-
0x47 for read. AlgoHub	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. AlgoHub	0x07	0x02 [MSB .. .. LSB]	Write SpO_2 motion-detection threshold (signed 32-bit int, equal to 10^5 x milli-g threshold value)	-
0x47 for read. AlgoHub	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. AlgoHub	0x07	0x03 [byte]	Write SpO2 AGC Timeout (sec) .	
0x47 for read. AlgoHub	0x07	0x03	Read SpO2 AGC Timeout (sec) . Default: 0x1E	SpO2 AGC Timeout (8-bit unsigned, seconds)
0x46 for write. AlgoHub	0x07	0x04	Write the timeout duration for SpO2 measurement in seconds (1 byte).	-
0x47 for read. AlgoHub	0x07	0x04	Read the timeout duration for SpO2 measurement in seconds. Default: 0x78	SpO_2 algorithm timeout (8-bit unsigned, seconds)
0x46 for write. AlgoHub	0x07	0x05 [byte]	Write initial HR algorithm value (8-bit unsigned).	-

0x47 for read. AlgoHub	0x07	0x05	Read initial HR algorithm value. Default: 0x3C	Initial heart rate (8-bit unsigned)
0x46 for write. AlgoHub	0x07	0x06 [MSB] [LSB]	Write height (16-bit unsigned, cm).	-
0x47 for read. AlgoHub	0x07	0x06	Read height. Default: 0x00AF	Height (16-bit unsigned, cm)
0x46 for write. AlgoHub	0x07	0x07 [MSB] [LSB]	Write weight (16-bit unsigned, kg).	-
0x47 for read. AlgoHub	0x07	0x07	Read weight. Default: 0x004E	Weight (16-bit unsigned, kg)
0x46 for write. AlgoHub	0x07	0x08 [byte]	Write age (8-bit unsigned, years).	
0x47 for read. AlgoHub	0x07	0x08	Read age. Default: 0x1E	Age (8-bit unsigned, years)
0x46 for write. AlgoHub	0x07	0x09 [byte]	Write gender. Byte: **0x00**: Male **0x01**: Female	
0x47 for read. AlgoHub	0x07	0x09	Read gender	Gender **0x00**: Male (default) **0x01**: Female
0x46 for write. AlgoHub	0x07	0x0A [algo operation mode byte]	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. AlgoHub	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. AlgoHub	0x07	0x0B [byte]	Write the enable/disable AEC byte: **0x00**: Disable **0x01**: Enable	-
0x47 for read. AlgoHub	0x07	0x0B	Read the enable/disable AEC.	**0x00**: Disabled **0x01**: Enabled (default)
0x46 for write. AlgoHub	0x07	0x0C [byte]	Write the Skin Contact Detection (SCD) algorithm enable **0x00**: Disable **0x01**: Enable	-
0x47 for read. AlgoHub	0x07	0x0C	Read the Skin Contact Detection (SCD) algorithm enable	**0x00**: Disable **0x01**: Enable (default)
0x46 for write. AlgoHub	0x07	0x0D [MSB] [LSB]	Write adjusted target PD current period (16-bit unsigned, seconds)	-
0x47 for read. AlgoHub	0x07	0x0D	Read adjusted target PD current period in seconds. Default: 0x0708	Adjusted target PD current period (16-bit unsigned, seconds)
0x46 for write. AlgoHub	0x07	0x0E [byte]	Write HR motion magnitude threshold (16-bit unsigned, 0.001g)	-
0x47 for read. AlgoHub	0x07	0x0E [MSB] [LSB]	Read HR motion magnitude threshold. Default: 0x0032 (0.05g)	[MSB] [LSB] motion magnitude threshold (16-bit unsigned, 0.001g)
0x46 for write. AlgoHub	0x07	0x0F [MSB] [LSB]	Write minimum PD current (16-bit unsigned, 0.1uA).	-
0x47 for read. AlgoHub	0x07	0x0F	Read minimum PD current. Default: 0x0032	Minimum PD current (16-bit unsigned, 0.1uA)
0x46 for write. AlgoHub	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.	-

0x47 for read. AlgoHub	0x07	0x10	Read initial PD current. Default: 0x0064	Initial PD current (16-bit unsigned, 0.1uA)
0x46 for write. AlgoHub	0x07	0x11 [MSB] [LSB]	Write target PD current (16-bit unsigned, 0.1uA). Applicable only if Auto Target PD Current Calculation is enabled.	
0x47 for read. AlgoHub	0x07	0x11	Read target PD current. Default: 0x0064	Target PD current (16-bit unsigned, 0.1uA)
0x46 for write. AlgoHub	0x07	0x12 [byte]	Write enable/disable automatic calculation of target PD current. byte: **0x00**: Disable **0x01**: Enable	-
0x47 for read. AlgoHub	0x07	0x12	Read enable/disable automatic calculation of target PD current	**0x00**: Disable **0x01**: Enable (default)
0x46 for write. AlgoHub	0x07	0x13 [byte1 byte 2]	Write minimum AFE integration time. byte1: **0x00** to **0x08**: PPG measurement 1 to 9. (1: Green, 2: IR, 3: red, N/A, ... N/A) byte2: **0x00**: 14.8µs **0x01**: 29.4µs **0x02**: 58.7µs **0x03**: 117.3µs	-
0x47 for read. AlgoHub	0x07	0x13	Read minimum AFE integration time.	Minimum AFE integration time for each PPG measurement 1 to 9. [byte1 to byte9] byteN: **0x00**: 14.8µs (default) **0x01**: 29.4µs **0x02**: 58.7µs **0x03**: 117.3µs
0x46 for write. AlgoHub	0x07	0x14 [byte1 byte2]	Write minimum sampling rate and averaging. byte1: **0x00** to **0x08**: PPG measurement 1 to 9. (1: Green, 2: IR, 3: red, N/A, ... N/A) byte2: **0x00**: 25sps, avg = 1 **0x01**: 50sps, avg = 2 **0x02**: 100sps, avg = 4 **0x03**: 200sps, avg = 8 **0x04**: 400sps, avg = 16	-
0x47 for read. AlgoHub	0x07	0x14	Read minimum sampling rate and averaging.	Minimum sampling rate and averaging for each PPG measurement 1 to 9. [byte1 to byte9]

				byteN: **0x00**: 25sps, avg = 1 **0x01**: 50sps, avg = 2 **0x02**: 100sps, avg = 4 **0x03**: 200sps, avg = 8 **0x04**: 400sps, avg = 16
0x46 for write. AlgoHub	0x07	0x15 [byte1 byte2]	Write maximum integration time: **0x00** to **0x08**: PPG measurement 1 to 9. (1: Green, 2: IR, 3: red, N/A, ... N/A) byte2: **0x00**: 14.8µs **0x01**: 29.4µs **0x02**: 58.7µs **0x03**: 117.3µs	-
0x47 for read. AlgoHub	0x07	0x15	Read maximum integration time:	Maximum integration time for each PPG measurement 1 to 9. [byte1 to byte9] byteN: **0x00**: 14.8µs **0x01**: 29.4µs **0x02**: 58.7µs **0x03**: 117.3µs (default)
0x46 for write. AlgoHub	0x07	0x16 [byte1 byte 2]	Write maximum sampling rate and averaging: byte1: **0x00** to **0x08**: PPG measurement 1 to 9. (1: Green, 2: IR, 3: red, N/A, ... N/A) byte2: **0x00**: 25sps, avg = 1 **0x01**: 50sps, avg = 2 **0x02**: 100sps, avg = 4 **0x03**: 200sps, avg = 8 **0x04**: 400sps, avg = 16	-
0x47 for read. AlgoHub	0x07	0x16	Read maximum sampling rate and averaging.	Maximum sampling rate and averaging for each PPG measurement 1 to 9. [byte1 to byte9] byteN: **0x00**: 25sps, avg = 1 **0x01**: 50sps, avg = 2 **0x02**: 100sps, avg = 4 **0x03**: 200sps, avg = 8 **0x04**: 400sps, avg = 16 (default)
0x46 for write. AlgoHub	0x07	0x17 0xWX, 0xYZ	Write slot and PD configuration for the two HR inputs to the WHRM algorithm. 0xWX, 0xYZ: WX is input 1 of the WHRM algorithm. W (MS nibble of WX):	-

			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 (LS nibble of WX): X = 0 for PD1 X = 1 for PD2 X = 3 for PD not used. YZ is input 2 of the WHRM algorithm. Y (MS nibble of YZ): 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 (LS nibble of YZ): Z = 0 for PD1 Z = 1 for PD2 Z = 3 for PD not used.	
0x47 for read. AlgoHub	0x07	0x17	Read Slot and PD configuration for the two HR inputs to the WHRM algorithm.	0xWX, 0xYZ Default: 0x0001 WX is input 1 of the WHRM algorithm. W (MS nibble of WX): 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 (LS nibble of WX): X = 0 for PD1 X = 1 for PD2 X = 3 for PD not used. YZ is input 2 of the WHRM algorithm. Y (MS nibble of YZ): 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 (LS nibble of YZ): Z = 0 for PD1 Z = 1 for PD2 Z = 3 for PD not used.

0x46 for write. AlgoHub	0x07	0x18 0xWX 0xYZ	Write Slot and PD configuration for the IR, red inputs to the WSpO2 algorithm. 0xWX, 0xYZ: WX is the LED/PD used for IR for the $WSpO_2$ algorithm. W (MS nibble of WX): 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 (LS nibble of WX): 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 (MS nibble of YZ): 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 (LS nibble of YZ): Z = 0 for PD1 Z = 1 for PD2 Z = 3 for PD not used	-
0x47 for read. AlgoHub	0x07	0x18	Read Slot and PD configuration for the IR, red inputs to the WSpO2 algorithm.	0xWX, 0xYZ Default: 0x1020 WX is the LED/PD used for IR for the $WSpO_2$ algorithm. W (MS nibble of WX): 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 (LS nibble of WX): 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 (MS nibble of YZ): 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 (LS nibble of YZ): Z = 0 for PD1 Z = 1 for PD2 Z = 3 for PD not used
0x46 for write. AlgoHub	0x07	0x1A [byte1 byte2]	Write initial AFE integration time. byte1: **0x00** to **0x08**: PPG measurement 1 to 9. (1: Green, 2: IR, 3: red, N/A, ... N/A) byte2: **0x00**: 14.8µs **0x01**: 29.4µs **0x02**: 58.7µs **0x03**: 117.3µs	-
0x47 for read. AlgoHub	0x07	0x1A	Read initial AFE integration time.	Initial AFE integration time for each PPG measurement 1 to 9. [byte1 to byte9] byteN: **0x00**: 14.8µs **0x01**: 29.4µs **0x02**: 58.7µs **0x03**: 117.3µs (default)
0x46 for write. AlgoHub	0x07	0x1B [byte1 byte2]	Write initial sampling rate and averaging. **0x00** to **0x08**: PPG measurement 1 to 9. (1: Green, 2: IR, 3: red, N/A, ... N/A) byte2: **0x00**: 25sps, avg = 1 **0x01**: 50sps, avg = 2 **0x02**: 100sps, avg = 4 **0x03**: 200sps, avg = 8 **0x04**: 400sps, avg = 16	
0x47 for read. AlgoHub	0x07	0x1B	Read minimum sampling rate and averaging.	Minimum sampling rate and averaging for each PPG measurement 1 to 9. [byte1 to byte9] byteN: **0x00**: 25sps, avg = 1 **0x01**: 50sps, avg = 2 **0x02**: 100sps, avg = 4 (default) **0x03**: 200sps, avg = 8 **0x04**: 400sps, avg = 16
0x46 for write. AlgoHub	0x07	0x1C [byte1 byte2]	Write maximum DAC offset. byte1: **0x00** to **0x08**: PPG measurement 1 to 9. (1: Green, 2: IR, 3: red, N/A, ... N/A)	-

			byte: **0x00**: 0 uA **0x01**: 8 uA **0x02**: 16 uA **0x03**: 24 uA	
0x47 for read. AlgoHub	0x07	0x1C	Read maximum DAC offset .	Maximum DAC offset for each PPG measurement 1 to 9. [byte1 to byte9] ByteN: **0x00**: 0 uA **0x01**: 8 uA **0x02**: 16 uA **0x03**: 24 uA (default)
0x46 for write. AlgoHub	0x07	0x1D [byte1 MSB LSB]	Write minimum LED current. byte1: **0x00** to **0x08**: PPG measurement 1 to 9. (1: Green, 2: IR, 3: red, N/A, ... N/A) MSB LSB: 16-bit unsigned, lsb=0.1mA	-
0x47 for read. AlgoHub	0x07	0x1D	Read minimum LED current .	Minimum LED current for each PPG measurement 1 to 9. [MSBLSB_1 to MSBLSB_9] MSBLSB_N: 16-bit unsigned, lsb 0.1mA
0x46 for write. AlgoHub	0x07	0x1E [byte1 MSB LSB]	Write maximum LED current. byte1: **0x00** to **0x08**: PPG measurement 1 to 9. (1: Green, 2: IR, 3: red, N/A, ... N/A) MSB LSB: 16-bit unsigned, lsb=0.1mA	-
0x47 for read. AlgoHub	0x07	0x1E	Read maximum LED current .	Maximum LED current for each PPG measurement 1 to 9. [MSBLSB_1 to MSBLSB_9] MSBLSB_N: 16-bit unsigned, lsb 0.1mA
0x46 for write. AlgoHub	0x07	0x1F [byte1 MSB LSB]	Write minimum LED current step that the algorithm can increase/decrease with each measurement. byte1: **0x00** to **0x08**: PPG measurement 1 to 9. (1: Green, 2: IR, 3: red, N/A, ... N/A) MSB LSB: 16-bit unsigned, lsb=0.1mA	-
0x47 for read. AlgoHub	0x07	0x1F	Read minimum LED current step that the algorithm can increase/decrease with each measurement.	Minimum LED current step for each PPG measurement 1 to 9.

				[MSBLSB_1 to MSBLSB_9] MSBLSB_N: 16-bit unsigned, lsb 0.1mA
0x46 for write. AlgoHub	0x07	0x20 [byte1 byte 2]	Write HR master channel selection option for each measurement. byte1: **0x00** to **0x08**: PPG measurement 1 to 9. (1: Green, 2: IR, 3: red, N/A, ... N/A) byte2: **0x00:** PD1 selected as master channel. **0x01:** PD2 selected as master channel. **0x02:** master channel selected automatically.	-
0x47 for read. AlgoHub	0x07	0x20	Read HR master channel selection option for each measurement .	HR master channel selection for each PPG measurement 1 to 9. [byte1 to byte9] byteN: **0x00:** PD1 selected as master channel. **0x01:** PD2 selected as master channel. **0x02:** master channel selected automatically.
0x46 for write. AlgoHub	0x07	0x21 [byte1 byte2]	Write full scale PD current. byte1: **0x00** to **0x08**: PPG measurement 1 to 9. (1: Green, 2: IR, 3: red, N/A, ... N/A) byte2: **0x00**: 4 uA **0x01**: 8 uA **0x02**: 16 uA **0x03**: 32 uA	-
0x47 for read. AlgoHub	0x07	0x21	Read full scale PD current.	Full scale PD current for each PPG measurement 1 to 9. [byte1 to byte9] byteN: **0x00**: 4 uA **0x01**: 8 uA **0x02**: 16 uA **0x03**: 32 uA
0x46 for write. AlgoHub	0x07	0x22 [byte]	Write AFE type. byte: **0x00**: AFE that allows independent integration time and sampling average settings for different measurements, e.g., MAX86171. **0x01**: AFE that uses shared integration time and sampling average settings for different measurements, e.g., MAX86141	-
0x47 for read. AlgoHub	0x07	0x22	Read AFE type.	AFE type. **0x00**: AFE that allows independent integration

				time and sampling average settings for different measurements, e.g., MAX86171. (default) **0x01**: AFE that uses shared integration time and sampling average settings for different measurements, e.g., MAX86141
0x46 for write. AlgoHub	0x07	0x23 [byte1 byte2]	Write DAC offset for PD channel 1. byte1: **0x00** to **0x08**: PPG measurement 1 to 9. (1: Green, 2: IR, 3: red, N/A, ... N/A) byte2: **0x00**: 0 uA **0x01**: 8 uA **0x02**: 16 uA **0x03**: 24 uA	-
0x47 for read. AlgoHub	0x07	0x23	Read DAC offset for PD channel 1.	DAC offset for PD channel 1 for each PPG measurement 1 to 9. [byte1 to byte9] ByteN: **0x00**: 0 uA (default) **0x01**: 8 uA **0x02**: 16 uA **0x03**: 24 uA
0x46 for write. AlgoHub	0x07	0x24 [byte1 byte2]	Write DAC offset for PD channel 2. byte1: **0x00** to **0x08**: PPG measurement 1 to 9. (1: Green, 2: IR, 3: red, N/A, ... N/A) byte: **0x00**: 0 uA **0x01**: 8 uA **0x02**: 16 uA **0x03**: 24 uA	-
0x47 for read. AlgoHub	0x07	0x24	Read DAC offset for PD channel 2.	DAC offset for PD channel 2 for each PPG measurement 1 to 9. [byte1 to byte9] ByteN: **0x00**: 0 uA (default) **0x01**: 8 uA **0x02**: 16 uA **0x03**: 24 uA
0x46 for write. AlgoHub	0x07	0x25 [byte1 [MSB LSB]	Write initial LED current. (16-bit unsigned, lsb = 0.1mA) byte1: **0x00** to **0x08**: PPG measurement 1 to 9. (1: Green, 2: IR, 3: red, N/A, ... N/A)	-

0x47 for read. AlgoHub	0x07	0x25	Read initial LED current.	Initial LED current for each PPG measurement 1 to 9. [MSB1 LSB1 to MSB9 LSB9] MSBn LSBn: Initial LED current. (16-bit unsigned, lsb = 0.1mA)
0x46 for write. AlgoHub	0x07	0x26 CMD_DELAY = 25ms	Reset AFE and algorithm to the default settings below: 0x50 0x07 0x00: A=-4.4027606, B=18.695252, C= 110.6986082 0x50 0x07 0x0A: 0x00 0x50 0x07 0x06: 0xAF 0x50 0x07 0x07: 0xE4 0x50 0x07 0x13: 0x00 0x50 0x07 0x15: 0x03 0x50 0x07 0x1C: 0x03	-
0x47 for read. AlgoHub	0x07	0x27 (CMD_DELAY = 5 ms)	Read algorithm AFE settings requests.	[byte1 byte2 byte3 byte4 byte5] byte1, byte2: bit 15: **0**: No LED current update request **1**: LED current update requested bits 14-0: LED current (15-bit unsigned, 0.1mA) byte3: bit 7: **0**: No integration time update request **1**: Integration time update requested bits 6-0: AFE integration time: **0x00**: 14.8µs **0x01**: 29.4µs **0x02**: 58.7µs **0x03**: 117.3µs byte4: bit 7: **0**: No sampling rate update request **1**: Sampling rate update requested bits 6-0: Sampling rate and averaging. **0x00**: 25sps, avg = 1 **0x01**: 50sps, avg = 2 **0x02**: 100sps, avg = 4 **0x03**: 200sps, avg = 8

				0x04: 400sps, avg = 16 byte5: DAC offset bit 7: **0**: No DAC offset update request **1**: DAC offset update requested **0x00**: 0 uA **0x01**: 8 uA **0x02**: 16 uA **0x03**: 24 uA
0x46 for write. AlgoHub	0x07	0x28 (CMD_DELAY = 5 ms)	Clear algorithm AFE settings request flag. This command is sent after the host has honored the algorithm AFE requests.	-
0x50 for write. Sensor Hub	0x08	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, scaled up by 100,000)	-
0x51 for read. Sensor Hub	0x08	0x00	Read SpO_2 calibration coefficients (12 bytes comprised of three 32-bit signed values, scaled up by 100,000) Default: 0xFFF9482C FFE3792B 00A8E9A4 A = -4.4027606 (0xFFF9482C) B = -18.695252 (0x FFE3792) C = 110.6986082 (0x00A8E9A4)	32-bit signed integer A, 32-bit signed integer B, 32-bit signed integer C Values scaled up by 100,000
0x50 for write. Sensor Hub	0x08	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. Sensor Hub	0x08	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. Sensor Hub	0x08	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. Sensor Hub	0x08	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)
0x50 for write. Sensor Hub	0x08	0x03 [byte]	Write SpO2 AGC Timeout (sec) .	

0x51 for read. Sensor Hub	0x08	0x03	Read SpO2 AGC Timeout (sec) . Default: 0x1E	SpO2 AGC Timeout (8-bit unsigned, seconds)
0x50 for write. Sensor Hub	0x08	0x04	Write the timeout duration for SpO2 measurement in seconds (1 byte).	-
0x51 for read. Sensor Hub	0x08	0x04	Read the timeout duration for SpO2 measurement in seconds. Default: 0x78	SpO_2 algorithm timeout (8-bit unsigned, seconds)
0x50 for write. Sensor Hub	0x08	0x05 [byte]	Write initial HR algorithm value (8-bit unsigned).	-
0x51 for read. Sensor Hub	0x08	0x05	Read initial HR algorithm value. Default: 0x3C	Initial heart rate (8-bit unsigned)
0x50 for write. Sensor Hub	0x08	0x06 [MSB] [LSB]	Write height (16-bit unsigned, cm).	-
0x51 for read. Sensor Hub	0x08	0x06	Read height. Default: 0x00AF	Height (16-bit unsigned, cm)
0x50 for write. Sensor Hub	0x08	0x07 [MSB] [LSB]	Write weight (16-bit unsigned, kg).	-
0x51 for read. Sensor Hub	0x08	0x07	Read weight. Default: 0x004E	Weight (16-bit unsigned, kg)
0x50 for write. Sensor Hub	0x08	0x08 [byte]	Write age (8-bit unsigned, years).	
0x51 for read. Sensor Hub	0x08	0x08	Read age. Default: 0x1E	Age (8-bit unsigned, years)
0x50 for write Sensor Hub	0x08	0x09 [byte]	Write gender. Byte: **0x00**: Male **0x01**: Female	
0x51 for read Sensor Hub	0x08	0x09	Read gender	Gender **0x00**: Male (default) **0x01**: Female

0x50 for write. Sensor Hub	0x08	0x0A [algo operation mode byte]	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	-
0x51 for read. Sensor Hub	0x08	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
0x50 for write. Sensor Hub	0x08	0x0B [byte]	Write the enable/disable AEC byte: **0x00**: Disable **0x01**: Enable	-
0x51 for read. Sensor Hub	0x08	0x0B	Read the enable/disable AEC.	**0x00**: Disabled **0x01**: Enabled (default)
0x50 for write. Sensor Hub	0x08	0x0C [byte]	Write the Skin Contact Detection (SCD) algorithm enable **0x00**: Disable **0x01**: Enable	-
0x51 for read. Sensor Hub	0x08	0x0C	Read the Skin Contact Detection (SCD) algorithm enable	**0x00**: Disable **0x01**: Enable (default)
0x50 for write. Sensor Hub	0x08	0x0D [MSB] [LSB]	Write adjusted target PD current period (16-bit unsigned, seconds)	-
0x51 for read. Sensor Hub	0x08	0x0D	Read adjusted target PD current period in seconds. Default: 0x0708	Adjusted target PD current period (16-bit unsigned, seconds)
0x50 for write. Sensor Hub	0x08	0x0E [byte]	Write HR motion magnitude threshold (16-bit unsigned, 0.001g)	-
0x51 for read. Sensor Hub	0x08	0x0E [MSB] [LSB]	Read HR motion magnitude threshold. Default: 0x0032 (0.05g)	[MSB] [LSB] motion magnitude threshold (16-bit unsigned, 0.001g)
0x50 for write.	0x08	0x0F [MSB] [LSB]	Write minimum PD current (16-bit unsigned, 0.1uA).	-

Sensor Hub				
0x51 for read. Sensor Hub	0x08	0x0F	Read minimum PD current. Default: 0x0032	Minimum PD current (16-bit unsigned, 0.1uA)
0x50 for write. Sensor Hub	0x08	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. Sensor Hub	0x08	0x10	Read initial PD current. Default: 0x0064	Initial PD current (16-bit unsigned, 0.1uA)
0x50 for write. Sensor Hub	0x08	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. Sensor Hub	0x08	0x11	Read target PD current. Default: 0x0064	Target PD current (16-bit unsigned, 0.1uA)
0x50 for write. Sensor Hub	0x08	0x12 [byte]	Write enable/disable automatic calculation of target PD current. byte: **0x00**: Disable **0x01**: Enable	-
0x51 for read. Sensor Hub	0x08	0x12	Read enable/disable automatic calculation of target PD current	**0x00**: Disable **0x01**: Enable (default)
0x50 for write. Sensor Hub	0x08	0x13 [byte1 byte2]	Write minimum AFE integration time. byte1: **0x00** to **0x08**: PPG measurement 1 to 9. (1: Green, 2: IR, 3: red, N/A, ... N/A) byte2: **0x00**: 14.8µs **0x01**: 29.4µs **0x02**: 58.7µs **0x03**: 117.3µs	-
0x51 for read. Sensor Hub	0x08	0x13	Read minimum AFE integration time.	Minimum AFE integration time for each PPG measurement 1 to 9. [byte1 to byte9] byteN: **0x00**: 14.8µs (default) **0x01**: 29.4µs **0x02**: 58.7µs **0x03**: 117.3µs
0x50 for write.	0x08	0x14 [byte1 byte2]	Write minimum sampling rate and averaging.	-

Sensor Hub			byte1: **0x00** to **0x08**: PPG measurement 1 to 9. (1: Green, 2: IR, 3: red, N/A, ... N/A) byte2: **0x00**: 25sps, avg = 1 **0x01**: 50sps, avg = 2 **0x02**: 100sps, avg = 4 **0x03**: 200sps, avg = 8 **0x04**: 400sps, avg = 16	
0x51 for read. Sensor Hub	0x08	0x14	Read minimum sampling rate and averaging.	Minimum sampling rate and averaging for each PPG measurement 1 to 9. [byte1 to byte9] byteN: **0x00**: 25sps, avg = 1 **0x01**: 50sps, avg = 2 **0x02**: 100sps, avg = 4 **0x03**: 200sps, avg = 8 **0x04**: 400sps, avg = 16
0x50 for write. Sensor Hub	0x08	0x15 [byte1 byte2]	Write maximum integration time: byte1: **0x00** to **0x08**: PPG measurement 1 to 9. (1: Green, 2: IR, 3: red, N/A, ... N/A) byte2: **0x00**: 14.8µs **0x01**: 29.4µs **0x02**: 58.7µs **0x03**: 117.3µs	-
0x51 for read. Sensor Hub	0x08	0x15	Read maximum integration time:	Maximum integration time for each PPG measurement 1 to 9. [byte1 to byte9] byteN: **0x00**: 14.8µs **0x01**: 29.4µs **0x02**: 58.7µs **0x03**: 117.3µs (default)
0x50 for write. Sensor Hub	0x08	0x16 [byte1 byte2]	Write maximum sampling rate and averaging: byte1: **0x00** to **0x08**: PPG measurement 1 to 9. (1: Green, 2: IR, 3: red, N/A, ... N/A) byte2: **0x00**: 25sps, avg = 1 **0x01**: 50sps, avg = 2 **0x02**: 100sps, avg = 4 **0x03**: 200sps, avg = 8 **0x04**: 400sps, avg = 16	-
0x51 for read. Sensor Hub	0x08	0x16	Read maximum sampling rate and averaging:	Maximum sampling rate and averaging for each PPG measurement 1 to 9. [byte1 to byte9]

				byteN: **0x00**: 25sps, avg = 1 **0x01**: 50sps, avg = 2 **0x02**: 100sps, avg = 4 **0x03**: 200sps, avg = 8 **0x04**: 400sps, avg = 16 (default)
0x50 for write. Sensor Hub	0x08	0x17 0xWX, 0xYZ	Write slot and PD configuration for the two HR inputs to the WHRM algorithm. 0xWX, 0xYZ: WX is input 1 of the WHRM algorithm. W (MS nibble of WX): 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 (LS nibble of WX): X = 0 for PD1 X = 1 for PD2 X = 3 for PD not used. YZ is input 2 of the WHRM algorithm. Y (MS nibble of YZ): 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 (LS nibble of YZ): Z = 0 for PD1 Z = 1 for PD2 Z = 3 for PD not used.	-
0x51 for read. Sensor Hub	0x08	0x17	Read Slot and PD configuration for the two HR inputs to the WHRM algorithm.	0xWX, 0xYZ Default: 0x0001 WX is input 1 of the WHRM algorithm. W (MS nibble of WX): 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 (LS nibble of WX): X = 0 for PD1 X = 1 for PD2 X = 3 for PD not used.

				YZ is input 2 of the WHRM algorithm. Y (MS nibble of YZ): 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 (LS nibble of YZ): Z = 0 for PD1 Z = 1 for PD2 Z = 3 for PD not used.
0x50 for write. Sensor Hub	0x08	0x18 0xWX 0xYZ	Write Slot and PD configuration for the IR, red inputs to the WSpO2 algorithm. 0xWX, 0xYZ: WX is the LED/PD used for IR for the $WSpO_2$ algorithm. W (MS nibble of WX): 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 (LS nibble of WX): 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 (MS nibble of YZ): 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 (LS nibble of YZ): Z = 0 for PD1 Z = 1 for PD2 Z = 3 for PD not used	-
0x51 for read. Sensor Hub	0x08	0x18	Read Slot and PD configuration for the IR, red inputs to the WSpO2 algorithm.	0xWX, 0xYZ Default: 0x1020 WX is the LED/PD used for IR for the $WSpO_2$ algorithm. W (MS nibble of WX): 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 (LS nibble of WX): 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 (MS nibble of YZ): 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 (LS nibble of YZ): Z = 0 for PD1 Z = 1 for PD2 Z = 3 for PD not used
0x50 for write. Sensor Hub	0x08	0x19 [byte1 WX]	Write LED driver and MUX configuration. byte1: **0x00** to **0x08**: PPG measurement 1 to 9. (1: Green, 2: IR, 3: red, N/A, ... N/A) W (MS nibble of WX): W = 0 for LED Driver B will fire LED1 W = 1 for LED Driver B will fire LED2 W = 2 for LED Driver B will fire LED3 W = 3 for LED Driver B will fire LED4 W = 4 for LED Driver B will fire LED5 W = 5 for LED Driver B will fire LED6 X (LS nibble of WX): X = 0 for LED Driver A will fire LED1 X = 1 for LED Driver A will fire LED2 X = 2 for LED Driver A will fire LED3 X = 3 for LED Driver A will fire LED4 X = 4 for LED Driver A will fire LED5 X = 5 for LED Driver A will fire LED6	
0x51 for read. Sensor Hub	0x08	0x19	Read LED driver and MUX .	LED driver and MUX configuration for each PPG measurement 1 to 9. [WX1 to WX9] W (MS nibble of WX_N): W = 0 for LED Driver B will fire LED1 W = 1 for LED Driver B will fire LED2 W = 2 for LED Driver B will fire LED3

				W = 3 for LED Driver B will fire LED4 W = 4 for LED Driver B will fire LED5 W = 5 for LED Driver B will fire LED6 X (LS nibble of WX_N): X = 0 for LED Driver A will fire LED1 X = 1 for LED Driver A will fire LED2 X = 2 for LED Driver A will fire LED3 X = 3 for LED Driver A will fire LED4 X = 4 for LED Driver A will fire LED5 X = 5 for LED Driver A will fire LED6
0x50 for write. Sensor Hub	0x08	0x1A [byte1 byte2]	Write initial AFE integration time. byte1: **0x00** to **0x08**: PPG measurement 1 to 9. (1: Green, 2: IR, 3: red, N/A, ... N/A) byte2: **0x00**: 14.8µs **0x01**: 29.4µs **0x02**: 58.7µs **0x03**: 117.3µs	-
0x51 for read. Sensor Hub	0x08	0x1A	Read initial AFE integration time.	Initial AFE integration time for each PPG measurement 1 to 9. [byte1 to byte9] ByteN: **0x00**: 14.8µs **0x01**: 29.4µs **0x02**: 58.7µs **0x03**: 117.3µs (default)
0x50 for write. Sensor Hub	0x08	0x1B [byte1 byte2]	Write initial sampling rate and averaging. **0x00** to **0x08**: PPG measurement 1 to 9. (1: Green, 2: IR, 3: red, N/A, ... N/A) byte2: **0x00**: 25sps, avg = 1 **0x01**: 50sps, avg = 2 **0x02**: 100sps, avg = 4 **0x03**: 200sps, avg = 8 **0x04**: 400sps, avg = 16	
0x51 for read. Sensor Hub	0x08	0x1B	Read minimum sampling rate and averaging.	Minimum sampling rate and averaging for each PPG measurement 1 to 9. [byte1 to byte9] ByteN: **0x00**: 25sps, avg = 1 **0x01**: 50sps, avg = 2

				0x02: 100sps, avg = 4 (default) **0x03**: 200sps, avg = 8 **0x04**: 400sps, avg = 16
0x50 for write. Sensor Hub	0x08	0x1C [byte1 byte2]	Write maximum DAC offset. byte1: **0x00** to **0x08**: PPG measurement 1 to 9. (1: Green, 2: IR, 3: red, N/A, ... N/A) byte: **0x00**: 0 uA **0x01**: 8 uA **0x02**: 16 uA **0x03**: 24 uA	-
0x51 for read. Sensor Hub	0x08	0x1C	Read maximum DAC offset .	Maximum DAC offset for each PPG measurement 1 to 9. [byte1 to byte9] ByteN: **0x00**: 0 uA **0x01**: 8 uA **0x02**: 16 uA **0x03**: 24 uA (default)
0x50 for write. Sensor Hub	0x08	0x1D [byte1 MSB LSB]	Write minimum LED current. byte1: **0x00** to **0x08**: PPG measurement 1 to 9. (1: Green, 2: IR, 3: red, N/A, ... N/A) MSB LSB: 16-bit unsigned, lsb=0.1mA	-
0x51 for read. Sensor Hub	0x08	0x1D	Read minimum LED current .	Minimum LED current for each PPG measurement 1 to 9. [MSBLSB_1 to MSBLSB_9] MSBLSB_N: 16-bit unsigned, lsb 0.1mA
0x50 for write. Sensor Hub	0x08	0x1E [byte1 MSB LSB]	Write maximum LED current. byte1: **0x00** to **0x08**: PPG measurement 1 to 9. (1: Green, 2: IR, 3: red, N/A, ... N/A) MSB LSB: 16-bit unsigned, lsb=0.1mA	-
0x51 for read. Sensor Hub	0x08	0x1E	Read maximum LED current .	Maximum LED current for each PPG measurement 1 to 9. [MSBLSB_1 to MSBLSB_9] MSBLSB_N: 16-bit unsigned, lsb 0.1mA
0x50 for write.	0x08	0x1F [byte1 MSB LSB]	Write minimum LED current step that the algorithm can increase/decrease with each measurement.	-

Sensor Hub			byte1: **0x00** to **0x08**: PPG measurement 1 to 9. (1: Green, 2: IR, 3: red, N/A, ... N/A) MSB LSB: 16-bit unsigned, lsb=0.1mA	
0x51 for read. Sensor Hub	0x08	0x1F	Read minimum LED current step that the algorithm can increase/decrease with each measurement.	Minimum LED current step for each PPG measurement 1 to 9. [MSBLSB_1 to MSBLSB_9] MSBLSB_N: 16-bit unsigned, lsb 0.1mA
0x50 for write. Sensor Hub	0x08	0x20 [byte1 byte 2]	Write HR master channel selection option for each measurement. byte1: **0x00** to **0x08**: PPG measurement 1 to 9. (1: Green, 2: IR, 3: red, N/A, ... N/A) byte2: **0x00:** PD1 selected as master channel. **0x01:** PD2 selected as master channel. **0x02:** master channel selected automatically.	-
0x51 for read. Sensor Hub	0x08	0x20	Read HR master channel selection option for each measurement .	HR master channel selection for each PPG measurement 1 to 9. [byte1 to byte9] byteN: **0x00:** PD1 selected as master channel. **0x01:** PD2 selected as master channel. **0x02:** master channel selected automatically.
0x50 for write. Sensor Hub	0x08	0x21 [byte1 byte2]	Write full scale PD current. byte1: **0x00** to **0x08**: PPG measurement 1 to 9. (1: Green, 2: IR, 3: red, N/A, ... N/A) byte2: **0x00**: 4 uA **0x01**: 8 uA **0x02**: 16 uA **0x03**: 32 uA	-
0x51 for read. Sensor Hub	0x08	0x21	Read full scale PD current.	Full scale PD current for each PPG measurement 1 to 9. [byte1 to byte9] byteN: **0x00**: 4 uA **0x01**: 8 uA **0x02**: 16 uA **0x03**: 32 uA
0x50 for write.	0x08	0x23 [byte1 byte2]	Write DAC offset for PD channel 1. byte1:	-

Sensor Hub			**0x00** to **0x08**: PPG measurement 1 to 9. (1: Green, 2: IR, 3: red, N/A, ... N/A) byte2: **0x00**: 0 uA **0x01**: 8 uA **0x02**: 16 uA **0x03**: 24 uA	
0x51 for read. Sensor Hub	0x08	0x23	Read DAC offset for PD channel 1.	DAC offset for PD channel 1 for each PPG measurement 1 to 9. [byte1 to byte9] ByteN: **0x00**: 0 uA (default) **0x01**: 8 uA **0x02**: 16 uA **0x03**: 24 uA
0x50 for write. Sensor Hub	0x08	0x24 [byte1 byte2]	Write DAC offset for PD channel 2. byte1: **0x00** to **0x08**: PPG measurement 1 to 9. (1: Green, 2: IR, 3: red, N/A, ... N/A) byte: **0x00**: 0 uA **0x01**: 8 uA **0x02**: 16 uA **0x03**: 24 uA	-
0x51 for read. Sensor Hub	0x08	0x24	Read DAC offset for PD channel 2.	DAC offset for PD channel 2 for each PPG measurement 1 to 9. [byte1 to byte9] ByteN: **0x00**: 0 uA (default) **0x01**: 8 uA **0x02**: 16 uA **0x03**: 24 uA
0x50 for write. Sensor Hub	0x08	0x25 [byte1 [MSB LSB]	Write initial LED current. (16-bit unsigned, lsb = 0.1mA) byte1: **0x00** to **0x08**: PPG measurement 1 to 9. (1: Green, 2: IR, 3: red, N/A, ... N/A)	-
0x51 for read. Sensor Hub	0x08	0x25	Read initial LED current.	Initial LED current for each PPG measurement 1 to 9. [MSB1 LSB1 to MSB9 LSB9] MSBn LSBn: Initial LED current. (16-bit unsigned, lsb = 0.1mA)
0x52 Sensor Hub	0x08	**0x00**: Disable (CMD_DELAY = 200ms) **0x01**: Enable Normal	Write enable/disable the Wearable Algorithm Suite (WHRM+WSpO$_2$) algorithm, SensorHub.	

		Algorithm Report (CMD_DELAY = 500ms)		
0x54 for write	[byte]	-	byte: **0x00**: MAX32674C releases sensor SPI bus to Host. AlgoHub mode. **0x01**: sensor SPI bus controlled by MAX32674C. SensorHub mode.	-
0x80 Bootloader	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 Bootloader	0x01	Use bytes 0x34 to 0x43 from the .msbl file.	Bootloader mode flash the application .msbl: Set the authentication bytes.	-
0x80 Bootloader	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 Bootloader	0x03	- (CMD_DELAY = 1400ms)	Bootloader mode flash the application .msbl: Erase the application flash memory.	-
0x80 Bootloader	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. (CMD_DELAY = 680ms)	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.	-
0x80 Bootloader	0x06	[MSB] [LSB]	Bootloader mode flash the application .msbl: Size of the partial page if using partial page writing (16-bit unsigned, range 1 to 8208)	-
0x81 Bootloader	0x00	-	Bootloader mode: Get bootloader version.	Major version byte, Minor version byte, Revision byte
0x81 Bootloader	0x01	-	Bootloader mode flash the application .msbl: Get the page size in bytes.	Upper byte of page size, Lower byte of page size
0x81 Bootloader	0x02	-	Get unique serial number (USN).	USN: 24 bytes
0x82 Bootloader	0x00 (CMD_ DELAY		Save bootloader configurations. Write this command after changes are made to any of the Bootloader	

Configuration	= 300ms)		Configuration settings. The device should be restarted for the new configuration to be active.	
0x82 Bootloader Configuration	0x01	0x00	0x01 Configure the device to check the state of the MFIO GPIO pin to decide if entering bootloader mode or application mode on after a reset.	
0x82 Bootloader Configuration	0x01	0x01	x00 Configure the MFIO pin to be P0.0	
0x82 Bootloader Configuration	0x01	0x20	0x00 Configure the MFIO pin the be use active-low for bootloader mode. The device enters bootloader mode if the MFIO GPIO pin is held low during power-on or during a RSTN device pin cycle.	
0xFF	0x03		Read the AlgoHubSensorHub version.	Major version byte, Minor version byte, Revision byte

5 Configuring the AlgoHub and Reading Processed Data

5.1 AlgoHub Output FIFO Format

The algorithm output FIFO format for AlgoHub is listed below (The initial response byte, “Read Status Byte” is not shown) . The algorithm outputs 24 bytes to the output FIFO.

Table 3. AlgoHub Output FIFO Format

DATA ITEM	# OF BYTES (MSB FIRST)	DESCRIPTION
PPG	PPG1	0,0,0
AFE Settings Request Flag, Algorithm operation mode	1	bit 8: 0: no AFE settings request flag 1: AFE settings request flag set; Host to issue the command 0x47 0x07 0x27 to read the AFE requested settings and then issue 0x47 0x07 0x28 when the request has been honored. bits 6 to 0: Algorithm 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	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 Calculated SpO_2 R value
SpO_2 confidence	1	Calculated SpO_2 confidence level in %
SpO_2	2	10x Calculated SpO_2 %
SpO_2 percent complete	1	Calculation progress in percenate 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 motion60012 00 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
Algorithm status	1	Algorithm return status: 0: Success 1: Null pointer error 2: Instance already initialized 3: Uninitialized instance 4: AFE controller error 5: Error, not categorized 6: Incompatible internal library error

5.2 AlgoHub AEC Normal Algorithm Report Configuration and Batched Processing

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.

For AlgoHub mode, the host is responsible for initializing the AFE, accel registers and informing the AlgoHub of the AFE initialization settings and for updating the AFE registers when the AFE request flag is set.

In the example below, both the AEC and SCD are enabled. The sequence of commands is shown in the table below.

Table 4. Host Commands—AlgoHub AEC and Read Normal Algorithm Report

	#	HOST COMMAND (HEX)	COMMAND DESCRIPTION	RESPONSE (HEX)
START ALGORITHM	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 AlgoHubSensorHub version for MAX32674C (x.y.z)	AB 00 32 YY ZZ
	1.3	AA 54 00	MAX32674C releases sensor SPI bus to Host. AlgoHub mode.	AB 00
	1.4	AA 10 01 01	Set the AlgoHub interrupt threshold.	AB 00
	1.5	AA 46 07 0B 01	Enable AEC	AB 00
	1.6	AA 46 07 12 01	Enable automatic PD calculation	AB 00
	1.7	AA 46 07 1A 00 03	Set PPG measurement 1 AFE integration time to 117.3us	AB 00
	1.8	AA 46 07 1A 01 03	Set PPG measurement 2 AFE integration time to 117.3us	AB 00
	1.9	AA 46 07 1A 02 03	Set PPG measurement 3 AFE integration time to 117.3us	AB 00
	1.10	AA 46 07 1B 00 01	Set PPG measurement 1 AFE sampling rate to 50sps, averaging to 2	AB 00
	1.11	AA 46 07 1B 01 04	Set PPG measurement 2 AFE sampling rate to 400sps, averaging to 16	AB 00
	1.12	AA 46 07 1B 02 04	Set PPG measurement 3 AFE sampling rate to 400sps, averaging to 16	AB 00
	1.13	AA 46 07 0F 00 7D	Set minimum PD current to 12.5uA	AB 00
	1.14	AA 46 07 10 01 38	Set initial PD current to 31.2uA	AB 00
	1.15	AA 46 07 0D 07 08	Set the adjusted target PD current period to 1800s	AB 00
	1.16	AA 46 07 0E 00 32	Set the HR motion magnitude threshold to 0.05g.	AB 00
	1.17	AA 46 07 23 00 00	Set the DAC offset for PD channel 1 for PPG measurement 1	AB 00
	1.18	AA 46 07 24 00 00	Set the DAC offset for PD channel 2 for PPG measurement 1	AB 00
	1.19	AA 46 07 1C 00 00	Set the maximum DAC offset for PPG measurement 1 to 0	AB 00
	1.20	AA 46 07 23 01 00	Set the DAC offset for PD channel 1 for PPG measurement 2	AB 00
	1.21	AA 46 07 24 01 00	Set the DAC offset for PD channel 2 for PPG measurement 2	AB 00
	1.22	AA 46 07 23 02 00	Set the DAC offset for PD channel 3 for PPG measurement 1	AB 00
	1.23	AA 46 07 24 02 00	Set the DAC offset for PD channel 2 for PPG measurement 3	AB 00
	1.24	AA 46 07 11 01 38	Set target PD current to 31.2uA	AB 00
	1.25	AA 46 07 0C 01	Enable SCD	AB 00
	1.26	AA 46 07 25 00 00 64	Set initial LED current to 10mA for PPG measurement 1	AB 00

	1.27	AA 46 07 25 01 00 C8	Set initial LED current to 20mA for PPG measurement 2	AB 00
	1.28	AA 46 07 25 02 00 C8	Set initial LED current to 20mA for PPG measurement 3	AB 00
	1.29	AA 10 00 03	Set the output of the AlgoHub to sensor and algorithm.	AB 00
	1.30	AA 44 07 01 01 (CMD_DELAY = 500ms)	Enable Wearable Algorithm Suite AlgoHub	AB 00
READING SAMPLES REPORT IN OUTPUT FIFO	Host repeats the following commands every 40ms			
	2.1	AA 14 00 [PPG1 PPG2 PPG3 PPG4 PPG5 PPG6 ACCLX ACCLY ACCLZ] (CMD_DELAY = 16 ms)	Write LEDPPG, accel data to the algorithm input FIFO. PPG1: 3 bytes [MSB .. LSB] WHRM HR Channel 1 (green1) PPG2: 3 bytes [MSB .. LSB] WHRM HR Channel 2 (green2) PPG3: 3 bytes [MSB .. LSB] SpO2 IR Channel (IR) PPG4: 3 bytes [MSB .. LSB] SpO2 IR Channel (red) PPG5: 3 bytes [MSB .. LSB] PPG6: 3 bytes [MSB .. LSB] Accelerometer set to ±8g: ACCLX: 2 byte accel X value (2's complement .001g): [MSB LSB] ACCLY: 2 byte accel Y value (2's complement .001g): [MSB LSB] ACCLZ: 2 byte accel Z value (2's complement .001g): [MSB LSB]	AB 00 mm nn mmnn is the ppg, accl number of bytes received (16-bit unsigned)
		Delay for 20 msec		
	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 01 (CMD_DELAY = 5 ms)	Read the data stored in the FIFO. The format of the samples report is shown in the normal algorithm report table.	AB 00 [Normal Algorithm Report]
		2.4, 2.5 are commanded if the AFE settings request flag is set		
	2.4	AA 47 07 27 (CMD_DELAY = 5 ms)	Read algorithm AFE settings requests.	[byte1 byte2 byte3 byte4 byte5] byte1, byte2: bit 15: **0**: No LED current update request **1**: LED current update requested bits 14-0: LED current (15-bit unsigned, 0.1mA) byte3: bit 7: **0**: No integration time update request

				1: Integration time update requested bits 6-0: AFE integration time: **0x00**: 14.8µs **0x01**: 29.4µs **0x02**: 58.7µs **0x03**: 117.3µs byte4: bit 7: **0**: No sampling rate update request **1**: Sampling rate update requested bits 6-0: Sampling rate and averaging. **0x00**: 25sps, avg = 1 **0x01**: 50sps, avg = 2 **0x02**: 100sps, avg = 4 **0x03**: 200sps, avg = 8 **0x04**: 400sps, avg = 16 byte5: DAC offset bit 7: **0**: No DAC offset update request **1**: DAC offset update requested **0x00**: 0 uA **0x01**: 8 uA **0x02**: 16 uA **0x03**: 24 uA
	2.5	AA 47 07 28 (CMD_DELAY = 5 ms)	Clear algorithm AFE settings request flag. This command is sent after the host has honored the algorithm AFE requests.	AB 00 08
STOP	Host disables sensors and algorithm:			
	3.1	AA 44 07 00 01 (CMD_DELAY = 200ms)	Disable the WAS algorithm.	AB 00
	3.2	AA 46 26 (CMD_DELAY = 25ms)	Reset the AFE settings of the WAS algorithm.	AB 00

The example below illustrates how the host may utilize batch processing to communicate with the AlgoHub at a 1s rate. The sequence of commands is shown in the table below.

Table 5. Host Commands—Batched Processing AlgoHub AEC and Read Normal Algorithm Report

	#	HOST COMMAND (HEX)	COMMAND DESCRIPTION	RESPONSE (HEX)
START ALGO	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 AlgoHubSensorHub version for MAX32674C (x.y.z)	AB 00 32 YY ZZ
	1.3	AA 54 00	MAX32674C releases sensor SPI bus to Host. AlgoHub mode.	AB 00
	1.4	AA 10 01 01	Set the AlgoHub interrupt threshold.	AB 00
	1.5	AA 46 07 0B 01	Enable AEC	AB 00
	1.6	AA 46 07 12 01	Enable automatic PD calculation	AB 00
	1.7	AA 46 07 1A 00 03	Set PPG measurement 1 AFE integration time to 117.3us	AB 00
	1.8	AA 46 07 1A 01 03	Set PPG measurement 2 AFE integration time to 117.3us	AB 00
	1.9	AA 46 07 1A 02 03	Set PPG measurement 3 AFE integration time to 117.3us	AB 00
	1.10	AA 46 07 1B 00 01	Set PPG measurement 1 AFE sampling rate to 50sps, averaging to 2	AB 00
	1.11	AA 46 07 1B 01 04	Set PPG measurement 2 AFE sampling rate to 400sps, averaging to 16	AB 00
	1.12	AA 46 07 1B 02 04	Set PPG measurement 3 AFE sampling rate to 400sps, averaging to 16	AB 00
	1.13	AA 46 07 0F 00 7D	Set minimum PD current to 12.5uA	AB 00
	1.14	AA 46 07 10 01 38	Set initial PD current to 31.2uA	AB 00
	1.15	AA 46 07 0D 07 08	Set the adjusted target PD current period to 1800s	AB 00
	1.16	AA 46 07 0E 00 32	Set the HR motion magnitude threshold to 0.05g.	AB 00
	1.17	AA 46 07 23 00 00	Set the DAC offset for PD channel 1 for PPG measurement 1	AB 00
	1.18	AA 46 07 24 00 00	Set the DAC offset for PD channel 2 for PPG measurement 1	AB 00
	1.19	AA 46 07 1C 00 00	Set the maximum DAC offset for PPG measurement 1 to 0	AB 00
	1.20	AA 46 07 23 01 00	Set the DAC offset for PD channel 1 for PPG measurement 2	AB 00
	1.21	AA 46 07 24 01 00	Set the DAC offset for PD channel 2 for PPG measurement 2	AB 00
	1.22	AA 46 07 23 02 00	Set the DAC offset for PD channel 3 for PPG measurement 1	AB 00
	1.23	AA 46 07 24 02 00	Set the DAC offset for PD channel 2 for PPG measurement 3	AB 00
	1.24	AA 46 07 11 01 38	Set target PD current to 31.2uA	AB 00
	1.25	AA 46 07 0C 01	Enable SCD	AB 00
	1.26	AA 46 07 25 00 00 64	Set initial LED current to 10mA for PPG measurement 1	AB 00
	1.27	AA 46 07 25 01 00 C8	Set initial LED current to 20mA for PPG measurement 2	AB 00
	1.28	AA 46 07 25 02 00 C8	Set initial LED current to 20mA for PPG measurement 3	AB 00
	1.29	AA 10 00 03	Set the output of the AlgoHub to sensor and algorithm.	AB 00
	1.30	AA 44 07 01 01 (CMD_DELAY = 500ms)	Enable Wearable Algorithm Suite AlgoHub	AB 00
READING SAMPLES	Host repeats the following commands every 1s			
	2.1	AA 14 00 [PPG1 PPG2 PPG3 PPG4 PPG5 PPG6 ACCLX ACCLY ACCLZ]	Write LEDPPG, accel data to the algorithm input FIFO. PPG1: 3 bytes [LSB .. MSB] PPG2: 3 bytes [LSB .. MSB] PPG3: 3 bytes [LSB .. MSB] PPG4: 3 bytes [LSB .. MSB]	AB 00 mm nn mmnn is the ppg, accl number of bytes received (16-bit unsigned)

		25 [ppg, accel input frames] (max 25) (CMD_DELAY = 5 ms)	PPG5: 3 bytes [LSB .. MSB] PPG6: 3 bytes [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
	2.4	AA 12 01 (CMD_DELAY = 5 ms)	Read the FIFO data for nn samples report. The format of the samples report is shown in the normal algorithm report table.	AB 00 nn*[Normal Algorithm Report]
		2.5, 2.6 are commanded if the AFE settings request flag is set		
	2.5	AA 47 07 27 (CMD_DELAY = 5 ms)	Read algorithm AFE settings requests.	[byte1 byte2 byte3 byte4 byte5] byte1, byte2: bit 15: **0**: No LED current update request **1**: LED current update requested bits 14-0: LED current (15-bit unsigned, 0.1mA) byte3: bit 7: **0**: No integration time update request **1**: Integration time update requested bits 6-0: AFE integration time: **0x00**: 14.8µs **0x01**: 29.4µs **0x02**: 58.7µs **0x03**: 117.3µs byte4: bit 7: **0**: No sampling rate update request **1**: Sampling rate update requested bits 6-0:

				Sampling rate and averaging. **0x00**: 25sps, avg = 1 **0x01**: 50sps, avg = 2 **0x02**: 100sps, avg = 4 **0x03**: 200sps, avg = 8 **0x04**: 400sps, avg = 16 byte5: DAC offset bit 7: **0**: No DAC offset update request **1**: DAC offset update requested **0x00**: 0 uA **0x01**: 8 uA **0x02**: 16 uA **0x03**: 24 uA
	2.6	AA 47 07 28 (CMD_DELAY = 5 ms)	Clear algorithm AFE settings request flag. This command is sent after the host has honored the algorithm AFE requests.	AB 00 08
STOP	Host disables sensors and algorithm:			
	3.1	AA 44 07 00 01 (CMD_DELAY = 200ms)	Disable the WAS algorithm.	AB 00
	3.2	AA 46 26 (CMD_DELAY = 25ms)	Reset the AFE settings of the WAS algorithm.	AB 00

5.3 AlgoHub, AGC and Normal Algorithm Report Configuration

The Automatic Gain Control (AGC) may be enabled by enabling AEC and disabling automatic PD calculation. The sequence of commands for enabling AGC is shown in the table below.

Table 6. Host Commands—AlgoHub AGC and Read Normal Algorithm Report

	#	HOST COMMAND (HEX)	COMMAND DESCRIPTION	RESPONSE (HEX)
START ALGORITHM	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 AlgoHubSensorHub version for MAX32674C (x.y.z)	AB 00 32 YY ZZ
	1.3	AA 54 00	MAX32674C releases sensor SPI bus to Host. AlgoHub mode.	AB 00
	1.4	AA 10 01 01	Set the AlgoHub interrupt threshold.	AB 00
	1.5	AA 46 07 0B 01	Enable AEC	AB 00
	1.6	AA 46 07 12 00	Disable automatic PD calculation	AB 00
	1.7	AA 46 07 1A 00 03	Set PPG measurement 1 AFE integration time to 117.3us	AB 00
	1.8	AA 46 07 1A 01 03	Set PPG measurement 2 AFE integration time to 117.3us	AB 00

	1.9	AA 46 07 1A 02 03	Set PPG measurement 3 AFE integration time to 117.3us	AB 00
	1.10	AA 46 07 1B 00 01	Set PPG measurement 1 AFE sampling rate to 50sps, averaging to 2	AB 00
	1.11	AA 46 07 1B 01 04	Set PPG measurement 2 AFE sampling rate to 400sps, averaging to 16	AB 00
	1.12	AA 46 07 1B 02 04	Set PPG measurement 3 AFE sampling rate to 400sps, averaging to 16	AB 00
	1.13	AA 46 07 0F 00 7D	Set minimum PD current to 12.5uA	AB 00
	1.14	AA 46 07 10 01 38	Set initial PD current to 31.2uA	AB 00
	1.15	AA 46 07 0D 07 08	Set the adjusted target PD current period to 1800s	AB 00
	1.16	AA 46 07 0E 00 32	Set the HR motion magnitude threshold to 0.05g.	AB 00
	1.17	AA 46 07 23 00 00	Set the DAC offset for PD channel 1 for PPG measurement 1	AB 00
	1.18	AA 46 07 24 00 00	Set the DAC offset for PD channel 2 for PPG measurement 1	AB 00
	1.19	AA 46 07 1C 00 00	Set the maximum DAC offset for PPG measurement 1 to 0	AB 00
	1.20	AA 46 07 23 01 00	Set the DAC offset for PD channel 1 for PPG measurement 2	AB 00
	1.21	AA 46 07 24 01 00	Set the DAC offset for PD channel 2 for PPG measurement 2	AB 00
	1.22	AA 46 07 23 02 00	Set the DAC offset for PD channel 3 for PPG measurement 1	AB 00
	1.23	AA 46 07 24 02 00	Set the DAC offset for PD channel 2 for PPG measurement 3	AB 00
	1.24	AA 46 07 11 01 38	Set target PD current to 31.2uA	AB 00
	1.25	AA 46 07 0C 01	Enable SCD	AB 00
	1.26	AA 46 07 25 00 00 64	Set initial LED current to 10mA for PPG measurement 1	AB 00
	1.27	AA 46 07 25 01 00 C8	Set initial LED current to 20mA for PPG measurement 2	AB 00
	1.28	AA 46 07 25 02 00 C8	Set initial LED current to 20mA for PPG measurement 3	AB 00
	1.29	AA 10 00 03	Set the output of the AlgoHub to sensor and algorithm.	AB 00
	1.30	AA 44 07 01 01 (CMD_DELAY = 500ms)	Enable Wearable Algorithm Suite AlgoHub	AB 00
READING SAMPLES REPORT IN OUTPUT FIFO	Host repeats the following commands every 40ms			
	2.1	AA 14 00 [PPG1 PPG2 PPG3 PPG4 PPG5 PPG6 ACCLX ACCLY ACCLZ] (CMD_DELAY = 16 ms)	Write LEDPPG, accel data to the algorithm input FIFO. PPG1: 3 bytes [MSB .. LSB] WHRM HR Channel 1 (green1) PPG2: 3 bytes [MSB .. LSB] WHRM HR Channel 2 (green2) PPG3: 3 bytes [MSB .. LSB] SpO2 IR Channel (IR) PPG4: 3 bytes [MSB .. LSB] SpO2 IR Channel (red) PPG5: 3 bytes [MSB .. LSB] PPG6: 3 bytes [MSB .. LSB] Accelerometer set to ±8g: ACCLX: 2 byte accel X value (2's complement .001g): [MSB LSB] ACCLY: 2 byte accel Y value (2's complement .001g): [MSB LSB] ACCLZ: 2 byte accel Z value (2's complement .001g): [MSB LSB]	AB 00 mm nn mmnn is the ppg, accl number of bytes received (16-bit unsigned)

		Delay for 20 msec		
	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 01 (CMD_DELAY = 5 ms)	Read the data stored in the FIFO. The format of the samples report is shown in the normal algorithm report table.	AB 00 [Normal Algorithm Report]
		2.4, 2.5 are commanded if the AFE settings request flag is set		
	2.4	AA 47 07 27 (CMD_DELAY = 5 ms)	Read algorithm AFE settings requests.	[byte1 byte2 byte3 byte4 byte5] byte1, byte2: bit 15: **0**: No LED current update request **1**: LED current update requested bits 14-0: LED current (15-bit unsigned, 0.1mA) byte3: bit 7: **0**: No integration time update request **1**: Integration time update requested bits 6-0: AFE integration time: **0x00**: 14.8µs **0x01**: 29.4µs **0x02**: 58.7µs **0x03**: 117.3µs byte4: bit 7: **0**: No sampling rate update request **1**: Sampling rate update requested bits 6-0: Sampling rate and averaging. **0x00**: 25sps, avg = 1 **0x01**: 50sps, avg = 2 **0x02**: 100sps, avg = 4 **0x03**: 200sps, avg = 8 **0x04**: 400sps, avg = 16 byte5: DAC offset bit 7:

				0: No DAC offset update request **1**: DAC offset update requested **0x00**: 0 uA **0x01**: 8 uA **0x02**: 16 uA **0x03**: 24 uA
	2.5	AA 47 07 28 (CMD_DELAY = 5 ms)	Clear algorithm AFE settings request flag. This command is sent after the host has honored the algorithm AFE requests.	AB 00 08
STOP	Host disables sensors and algorithm:			
	3.1	AA 44 07 00 01 (CMD_DELAY = 200ms)	Disable the WAS algorithm.	AB 00
	3.2	AA 46 26 (CMD_DELAY = 25ms)	Reset the AFE settings of the WAS algorithm.	AB 00

5.4 Configuration for AlgoHub Normal Algorithm Report when Host Implements AEC/AGC

If the system requirements does not need the algohub to handle AEC/AGC and if the host firmware has implemented the AEC/AGC code, then the algohub may disable the algohub AEC (which also disables algohub AGC) and configure the algohub per the following table.

Table 7. Host Commands—AlgoHub AlgoHub Normal Algorithm Report when Host Implements AEC/AGC

	#	HOST COMMAND (HEX)	COMMAND DESCRIPTION	RESPONSE (HEX)
START ALGORITHM	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 AlgoHubSensorHub version for MAX32674C (x.y.z)	AB 00 32 YY ZZ
	1.3	AA 54 00	MAX32674C releases sensor SPI bus to Host. AlgoHub mode.	AB 00
	1.4	AA 10 01 01	Set the AlgoHub interrupt threshold.	AB 00
	1.5	AA 46 07 0B 00	Enable AEC	AB 00
	1.6	AA 46 07 0C 01	Enable SCD	AB 00
	1.7	AA 46 07 25 00 00 64	Set initial LED current to 10mA for PPG measurement 1	AB 00
	1.8	AA 46 07 25 01 00 C8	Set initial LED current to 20mA for PPG measurement 2	AB 00
	1.9	AA 46 07 25 02 00 C8	Set initial LED current to 20mA for PPG measurement 3	AB 00
	1.10	AA 10 00 03	Set the output of the AlgoHub to sensor and algorithm.	AB 00
	1.11	AA 44 07 01 01 (CMD_DELAY = 500ms)	Enable Wearable Algorithm Suite AlgoHub	AB 00
READING	Host repeats the following commands every 40ms			
	2.1	AA 14 00 [PPG1 PPG2 PPG3 PPG4	Write LEDPPG, accel data to the algorithm input FIFO.	AB 00 mm nn

		PPG5 PPG6 ACCLX ACCLY ACCLZ] (CMD_DELAY = 16 ms)	PPG1: 3 bytes [MSB .. LSB] WHRM HR Channel 1 (green1) (or if green is not used, then use (IR)) PPG2: 3 bytes [MSB .. LSB] WHRM HR Channel 2 (green2) (or if green is not used, then use (IR)) PPG3: 3 bytes [MSB .. LSB] SpO2 IR Channel (IR) PPG4: 3 bytes [MSB .. LSB] SpO2 IR Channel (red) PPG5: 3 bytes [MSB .. LSB] PPG6: 3 bytes [MSB .. LSB] Accelerometer set to ±8g: ACCLX: 2 byte accel X value (2's complement .001g): [MSB LSB] ACCLY: 2 byte accel Y value (2's complement .001g): [MSB LSB] ACCLZ: 2 byte accel Z value (2's complement .001g): [MSB LSB]	mmnn is the ppg, accl number of bytes received (16-bit unsigned)
		Delay for 20 msec		
	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 01 (CMD_DELAY = 5 ms)	Read the data stored in the FIFO. The format of the samples report is shown in the normal algorithm report table.	AB 00 [Normal Algorithm Report]
		2.4, 2.5 are commanded if the AFE settings request flag is set		
	2.4	AA 47 07 27 (CMD_DELAY = 5 ms)	Read algorithm AFE settings requests.	[byte1 byte2 byte3 byte4 byte5] byte1, byte2: bit 15: **0**: No LED current update request **1**: LED current update requested bits 14-0: LED current (15-bit unsigned, 0.1mA) byte3: bit 7: **0**: No integration time update request **1**: Integration time update requested bits 6-0: AFE integration time: **0x00**: 14.8µs **0x01**: 29.4µs **0x02**: 58.7µs **0x03**: 117.3µs byte4: bit 7:

				0: No sampling rate update request **1**: Sampling rate update requested bits 6-0: Sampling rate and averaging. **0x00**: 25sps, avg = 1 **0x01**: 50sps, avg = 2 **0x02**: 100sps, avg = 4 **0x03**: 200sps, avg = 8 **0x04**: 400sps, avg = 16 byte5: DAC offset bit 7: **0**: No DAC offset update request **1**: DAC offset update requested **0x00**: 0 uA **0x01**: 8 uA **0x02**: 16 uA **0x03**: 24 uA
	2.5	AA 47 07 28 (CMD_DELAY = 5 ms)	Clear algorithm AFE settings request flag. This command is sent after the host has honored the algorithm AFE requests.	AB 00 08
STOP	Host disables sensors and algorithm:			
	3.1	AA 44 07 00 01 (CMD_DELAY = 200ms)	Disable the WAS algorithm.	AB 00
	3.2	AA 46 26 (CMD_DELAY = 25ms)	Reset the AFE settings of the WAS algorithm.	AB 00

5.5 AlgoHub Processing Response Duration

AlgoHub response duration depends on number of sample frames pushed to AlgoHub input FIFO in a single transaction. The CMD_DELAY between the data feed command and the status byte is 5ms. The maximum number of sample frames that may be sent in one command is 25 per period. The AlgoHub processing time response time can be calculated as:

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

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

6 Configuring the SensorHub and Reading Processed Data

6.1 I2C SensorHub Output FIFO Format

The SensorHub output FIFO may contain Sample Counter byte and/or Sensor Data, and/or Algorithm Data. The inclusion of these three categories of output data is controlled by the “Set the output format of the sensor hub”, 0x10 0x00 command and the formats are listed below (The initial response byte, “Read Status Byte” is not shown).

Table 8. SensorHub Sample Counter Byte Output FIFO Format

DATA ITEM	# OF BYTES (MSB FIRST)	DESCRIPTION
Sample Counter byte	1	Counter which cycles from 0-0xFF.

Table 9. SensorHub SensorData Output FIFO Format

DATA SOURCE (24 bytes)	DATA ITEM	# OF BYTES (MSB FIRST)	DESCRIPTION
MAX86176 PPG Data (18 Bytes)	PPG1 (PD1)	3	Green1
	PPG2 (PD1)	3	IR LED counts
	PPG3 (PD1)	3	Red LED counts
	PPG4 (PD2)	3	Green2
	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

Table 10. SensorHub Algorithm Data Output FIFO Format

DATA ITEM (20 bytes)	# OF BYTES (MSB FIRST)	DESCRIPTION
Algorithm operation mode	1	Algorithm 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	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 Calculated SpO_2 R value
SpO_2 confidence	1	Calculated SpO_2 confidence level in %
SpO_2	2	10x Calculated SpO_2 %
SpO_2 percent complete	1	Calculation progress in percentage 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

6.2 SensorHub, AEC aqnd Normal Algorithm Report Configuration

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 MAX32674C is configured to SensorHub mode with AEC, SCD enabled. The sequence of commands is shown in the table below.

Table 11. Host Commands—SensorHub, AEC and Read Normal Algorithm Report

	#	HOST COMMAND (HEX)	COMMAND DESCRIPTION	RESPONSE (HEX)
START ALGORITHM	Host initializes SensorHub and starts the algorithm using following commands:			
	1.1		Host sets SPI0_CSACC_N (P0.8) to high	
	1.2		Host initiates “Reset to Application Mode” sequence	
	Host initializes the SensorHub and starts the algorithm using following commands:			
	1.3	AA 02 00 (optional)	Read the operating mode	AB 00 00 application mode
	1.2	AA FF 03 (optional)	Read the AlgoHubSensorHub version for MAX32674C (x.y.z)	AB 00 32 YY ZZ
	1.3	AA 10 01 01	Set the SensorHub interrupt threshold.	AB 00
	1.4	AA 54 01	MAX32674C releases sensor SPI bus to MAX32674C. SensorHub mode.	AB 00
	1.5	AA 40 06 1E 1F	Write 0x1F to MAX86176 register 0x1E	AB 00
	1.6	AA 40 06 1D 05	Write 0x05 to MAX86176 register 0x1D	AB 00
	1.7	AA 40 06 11 07	Write 0x07 to MAX86176 register 0x11	AB 00
	1.8	AA 40 06 12 04	Write 0x04 to MAX86176 register 0x12	AB 00
	1.9	AA 40 06 90 02	Write 0x90 to MAX86176 register 0x90	AB 00
	1.10	AA 50 08 0B 01	Enable AEC	AB 00
	1.11	AA 50 08 12 01	Enable automatic PD calculation	AB 00
	1.12	AA 50 08 1A 00 03	Set PPG measurement 1 AFE integration time to 117.3us	AB 00
	1.13	AA 50 08 1A 01 03	Set PPG measurement 2 AFE integration time to 117.3us	AB 00
	1.14	AA 50 08 1A 02 03	Set PPG measurement 3 AFE integration time to 117.3us	AB 00
	1.15	AA 50 08 1B 00 01	Set PPG measurement 1 AFE sampling rate to 50sps, averaging to 2	AB 00
	1.16	AA 50 08 1B 01 04	Set PPG measurement 2 AFE sampling rate to 400sps, averaging to 16	AB 00
	1.17	AA 50 08 1B 02 04	Set PPG measurement 3 AFE sampling rate to 400sps, averaging to 16	AB 00
	1.18	AA 50 08 0F 00 7D	Set minimum PD current to 12.5uA	AB 00
	1.19	AA 50 08 10 01 38	Set initial PD current to 31.2uA	AB 00
	1.20	AA 50 08 0D 07 08	Set the adjusted target PD current period to 1800s	AB 00
	1.21	AA 50 08 0E 00 32	Set the HR motion magnitude threshold to 0.05g.	AB 00
	1.22	AA 50 08 23 00 00	Set the DAC offset for PD channel 1 for PPG measurement 1	AB 00
	1.23	AA 50 08 24 00 00	Set the DAC offset for PD channel 2 for PPG measurement 1	AB 00
	1.24	AA 50 08 1C 00 00	Set the maximum DAC offset for PPG measurement 1 to 0	AB 00
	1.25	AA 50 08 23 01 00	Set the DAC offset for PD channel 1 for PPG measurement 2	AB 00
	1.26	AA 50 08 24 01 00	Set the DAC offset for PD channel 2 for PPG measurement 2	AB 00
	1.27	AA 50 08 23 02 00	Set the DAC offset for PD channel 3 for PPG measurement 1	AB 00
	1.28	AA 50 08 24 02 00	Set the DAC offset for PD channel 2 for PPG measurement 3	AB 00
	1.29	AA 50 08 25 00 00 64	Set initial LED current to 10mA for PPG measurement 1	AB 00

	1.30	AA 50 08 25 01 00 C8	Set initial LED current to 20mA for PPG measurement 2	AB 00
	1.31	AA 50 08 25 02 00 C8	Set initial LED current to 20mA for PPG measurement 3	AB 00
	1.32	AA 50 08 11 01 38	Set target PD current to 31.2uA	AB 00
	1.33	AA 50 08 0C 01	Enable SCD	AB 00
	1.34	AA 10 00 07	Set the output format to Sample Counter byte, Sensor Data and Algorithm	AB 00
	1.35	AA 40 06 10 01	Write 0x01 to MAX86176 register 0x10	AB 00
	1.36	AA 40 06 10 02	Write 0x02 to MAX86176 register 0x10	AB 00
	1.37	AA 40 06 15 1E	Write 0x15 to MAX86176 register 0x1E	AB 00
	1.38	AA 40 06 18 80	Write 0x18 to MAX86176 register 0x80	AB 00
	1.39	AA 40 06 19 9F	Write 0x19 to MAX86176 register 0x9F	AB 00
	1.40	AA 40 06 1A 3F	Write 0x1A to MAX86176 register 0x3F	AB 00
	1.41	AA 40 06 1C 20	Write 0x1C to MAX86176 register 0x20	AB 00
	1.42	AA 40 06 1D 05	Write 0x1D to MAX86176 register 0x05	AB 00
	1.43	AA 40 06 1E 1F	Write 0x1E to MAX86176 register 0x1F	AB 00
	1.44	AA 40 06 86 40	Write 0x86 to MAX86176 register 0x40	AB 00
	1.45	AA 44 04 01 00 (CMD_DELAY = 50ms)	Enable the accelerometer.	AB 00
	1.46	AA 44 06 01 00 (CMD_DELAY = 500ms)	Enable MAX86176 AFE.	AB 00
	1.47	AA 52 08 01 (CMD_DELAY = 500ms)	Write enable/disable the Wearable Algorithm Suite (WHRM+WSpO$_2$) algorithm, SensorHub	AB 00
	1.48	AA 41 00 FF (optional)	Read register FF (PART_ID) of MAX86176	AB 00 [39]
	1.49	AA 41 04 0F (accel data is required LIS2DS12l)	Read register 0F (WHO_AM_I) of LIS2DS12	AB 00 [43]
READING SAMPLES REPORT IN OUTPUT	Host repeats the following commands every 40ms			
	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) If DataRdyInt is set, proceed to the next step.	AB 00 08
	2.2	AA 12 01	Read the data stored in the FIFO	AB 00 [sample counter byte] [sensor data] [algorithm data]
STOP	Host disables sensors and algorithm:			
	3.1	AA 44 04 00 (CMD_DELAY = 50ms)	Disable the accelerometer.	AB 00
	3.2	AA 44 06 00 (CMD_DELAY = 200ms)	Disable the MAX86176 AFE.	AB 00
	3.2	AA 52 08 00 (CMD_DELAY = 200ms)	Disable the Wearable Algorithm Suite (WHRM+WSpO2) algorithm	AB 00

6.3 SensorHub, AGC and Normal Algorithm Report

The Automatic Gain Control (AGC) may be enabled by enabling AEC and disabling automatic PD calculation. The sequence of commands for enabling AGC is shown in the table below.

Table 12. Host Commands—SensorHub AGC and Read Normal Algorithm Report

	#	HOST COMMAND (HEX)	COMMAND DESCRIPTION	RESPONSE (HEX)
START ALGORITHM	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 AlgoHubSensorHub version for MAX32674C (x.y.z)	AB 00 32 YY ZZ
	1.3	AA 10 01 01	Set the AlgoHub interrupt threshold.	AB 00
	1.4	AA 54 01	MAX32674C releases sensor SPI bus to MAX32674C. SensorHub mode.	AB 00
	1.5	AA 40 06 1E 1F	Write 0x1F to MAX86176 register 0x1E	AB 00
	1.6	AA 40 06 1D 05	Write 0x05 to MAX86176 register 0x1D	AB 00
	1.7	AA 40 06 11 07	Write 0x07 to MAX86176 register 0x11	AB 00
	1.8	AA 40 06 12 04	Write 0x04 to MAX86176 register 0x12	AB 00
	1.9	AA 40 06 90 02	Write 0x90 to MAX86176 register 0x90	AB 00
	1.10	AA 50 08 0B 01	Enable AEC	AB 00
	1.11	AA 50 08 12 00	Disable automatic PD calculation	AB 00
	1.12	AA 50 08 1A 00 03	Set PPG measurement 1 AFE integration time to 117.3us	AB 00
	1.13	AA 50 08 1A 01 03	Set PPG measurement 2 AFE integration time to 117.3us	AB 00
	1.14	AA 50 08 1A 02 03	Set PPG measurement 3 AFE integration time to 117.3us	AB 00
	1.15	AA 50 08 1B 00 01	Set PPG measurement 1 AFE sampling rate to 50sps, averaging to 2	AB 00
	1.16	AA 50 08 1B 01 04	Set PPG measurement 2 AFE sampling rate to 400sps, averaging to 16	AB 00
	1.17	AA 50 08 1B 02 04	Set PPG measurement 3 AFE sampling rate to 400sps, averaging to 16	AB 00
	1.18	AA 50 08 0F 00 7D	Set minimum PD current to 12.5uA	AB 00
	1.19	AA 50 08 10 01 38	Set initial PD current to 31.2uA	AB 00
	1.20	AA 50 08 0D 07 08	Set the adjusted target PD current period to 1800s	AB 00
	1.21	AA 50 08 0E 00 32	Set the HR motion magnitude threshold to 0.05g.	AB 00
	1.22	AA 50 08 23 00 00	Set the DAC offset for PD channel 1 for PPG measurement 1	AB 00
	1.23	AA 50 08 24 00 00	Set the DAC offset for PD channel 2 for PPG measurement 1	AB 00
	1.24	AA 50 08 1C 00 00	Set the maximum DAC offset for PPG measurement 1 to 0	AB 00
	1.25	AA 50 08 23 01 00	Set the DAC offset for PD channel 1 for PPG measurement 2	AB 00
	1.26	AA 50 08 24 01 00	Set the DAC offset for PD channel 2 for PPG measurement 2	AB 00
	1.27	AA 50 08 23 02 00	Set the DAC offset for PD channel 3 for PPG measurement 1	AB 00
	1.28	AA 50 08 24 02 00	Set the DAC offset for PD channel 2 for PPG measurement 3	AB 00
	1.29	AA 50 08 25 00 00 64	Set initial LED current to 10mA for PPG measurement 1	AB 00

	1.30	AA 50 08 25 01 00 C8	Set initial LED current to 20mA for PPG measurement 2	AB 00
	1.31	AA 50 08 25 02 00 C8	Set initial LED current to 20mA for PPG measurement 3	AB 00
	1.32	AA 50 08 11 01 38	Set target PD current to 31.2uA	AB 00
	1.33	AA 50 08 0C 01	Enable SCD	AB 00
	1.34	AA 10 00 07	Set the output format to Sample Counter byte, Sensor Data and Algorithm	AB 00
	1.35	AA 40 06 10 01	Write 0x01 to MAX86176 register 0x10	AB 00
	1.36	AA 40 06 10 02	Write 0x02 to MAX86176 register 0x10	AB 00
	1.37	AA 40 06 15 1E	Write 0x15 to MAX86176 register 0x1E	AB 00
	1.38	AA 40 06 18 80	Write 0x18 to MAX86176 register 0x80	AB 00
	1.39	AA 40 06 19 9F	Write 0x19 to MAX86176 register 0x9F	AB 00
	1.40	AA 40 06 1A 3F	Write 0x1A to MAX86176 register 0x3F	AB 00
	1.41	AA 40 06 1C 20	Write 0x1C to MAX86176 register 0x20	AB 00
	1.42	AA 40 06 1D 05	Write 0x1D to MAX86176 register 0x05	AB 00
	1.43	AA 40 06 1E 1F	Write 0x1E to MAX86176 register 0x1F	AB 00
	1.44	AA 40 06 86 40	Write 0x86 to MAX86176 register 0x40	AB 00
	1.45	AA 44 04 01 00 (CMD_DELAY = 50ms)	Enable the accelerometer.	AB 00
	1.46	AA 44 06 01 00 (CMD_DELAY = 500ms)	Enable MAX86176 AFE.	AB 00
	1.47	AA 52 08 01 (CMD_DELAY = 500ms)	Write enable/disable the Wearable Algorithm Suite (WHRM+WSpO$_2$) algorithm, SensorHub	AB 00
	1.48	AA 41 00 FF (optional)	Read register FF (PART_ID) of MAX86176	AB 00 [39]
	1.49	AA 41 04 0F (accel data is required LIS2DS12l)	Read register 0F (WHO_AM_I) of LIS2DS12	AB 00 [43]
READING SAMPLES REPORT IN OUTPUT	Host repeats the following commands every 40ms			
	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) If DataRdyInt is set, proceed to the next step.	AB 00 08
	2.2	AA 12 01	Read the data stored in the FIFO	AB 00 [sample counter byte] [sensor data] [algorithm data]
STOP	Host disables sensors and algorithm:			
	3.1	AA 44 04 00 (CMD_DELAY = 50ms)	Disable the accelerometer.	AB 00
	3.2	AA 44 06 00 (CMD_DELAY = 200ms)	Disable the MAX86176 AFE.	AB 00
	3.2	AA 52 08 00 (CMD_DELAY = 200ms)	Disable the Wearable Algorithm Suite (WHRM+WSpO2) algorithm	AB 00

6.4 SensorHub, PPG Raw Data Report

The host may configure the SensorHub to ouput raw rata (no algorithm) by enabling the AFE. The table below lists the set of commands that are needed to obtain the PPG raw data.

Table 13. Host Commands—PPG Raw Data Report

	#	HOST COMMAND (HEX)	COMMAND DESCRIPTION	RESPONSE (HEX)
START ALGORITHM	Host initializes the SensorHub and AFE for raw data report:			
	1.1	AA 02 00 (optional)	Read the operating mode	AB 00 00 application mode
	1.2	AA FF 03 (optional)	Read the AlgoHubSensorHub version for MAX32674C (x.y.z)	AB 00 32 YY ZZ
	1.3	AA 54 01	MAX32674C releases sensor SPI bus to MAX32674C. SensorHub mode.	AB 00
	1.4	AA 40 06 10 01	Write 0x01 to MAX86176 register 0x10	AB 00
	1.5	AA 40 06 10 01	Write 0x02 to MAX86176 register 0x10	AB 00
	1.6	AA 40 06 11 07	Write 0x07 to MAX86176 register 0x11	AB 00
	1.7	AA 40 06 12 04	Write 0x04 to MAX86176 register 0x12	AB 00
	1.8	AA 40 06 14 55	Write 0x55 to MAX86176 register 0x14	AB 00
	1.9	AA 40 06 15 19	Write 0x19 to MAX86176 register 0x15	AB 00
	1.10	AA 40 06 18 80	Write 0x80 to MAX86176 register 0x18	AB 00
	1.11	AA 40 06 19 9F	Write 0x9F to MAX86176 register 0x19	AB 00
	1.12	AA 40 06 1A 3F	Write 0x3F to MAX86176 register 0x1A	AB 00
	1.13	AA 40 06 1C 20	Write 0x20 to MAX86176 register 0x1C, FR_CLK_SEL = 1, 32.786kHz	AB 00
	1.14	AA 40 06 1D 05	Write 0x05 to MAX86176 register 0x1D	AB 00
	1.15	AA 40 06 1E 1F	Write 0x1F to MAX86176 register 0x1E. FR_CLK_DIV = 0x51F. PPG_FR = PPG_FR_CLK/FR_CLK_DIV = 32768/1311 = 25 Hz	AB 00
	1.16	AA 40 06 20 08	Write 0x08 to MAX86176 register 0x20	AB 00
	1.17	AA 40 06 21 18	Write 0x18 to MAX86176 register 0x21	AB 00
	1.18	AA 40 06 22 3F	Write 0x3F to MAX86176 register 0x22	AB 00
	1.19	AA 40 06 23 50	Write 0x50 to MAX86176 register 0x23	AB 00
	1.20	AA 40 06 24 08	Write 0x08 to MAX86176 register 0x24	AB 00
	1.21	AA 40 06 25 14	Write 0x14 to MAX86176 register 0x25	AB 00
	1.22	AA 40 06 26 00	Write 0x00 to MAX86176 register 0x26	AB 00
	1.23	AA 40 06 28 01	Write 0x01 to MAX86176 register 0x28	AB 00
	1.24	AA 40 06 29 1A	Write 0x1A to MAX86176 register 0x29	AB 00
	1.25	AA 40 06 2A 3F	Write 0x3F to MAX86176 register 0x2A	AB 00
	1.26	AA 40 06 2B 50	Write 0x50 to MAX86176 register 0x2B	AB 00
	1.27	AA 40 06 2C 01	Write 0x01 to MAX86176 register 0x2C	AB 00
	1.28	AA 40 06 2D 28	Write 0x28 to MAX86176 register 0x2D	AB 00
	1.29	AA 40 06 2E 00	Write 0x09 to MAX86176 register 0x2E	AB 00
	1.30	AA 40 06 30 02	Write 0x02 to MAX86176 register 0x30	AB 00
	1.31	AA 40 06 31 1A	Write 0x1A to MAX86176 register 0x31	AB 00
	1.32	AA 40 06 32 3F	Write 0x3F to MAX86176 register 0x32	AB 00
	1.33	AA 40 06 33 50	Write 0x50 to MAX86176 register 0x33	AB 00
	1.34	AA 40 06 34 01	Write 0x01 to MAX86176 register 0x34	AB 00
	1.35	AA 40 06 35 28	Write 0x28 to MAX86176 register 0x35	AB 00
	1.36	AA 40 06 36 00	Write 0x00 to MAX86176 register 0x36	AB 00
	1.37	AA 40 06 86 40	Write 0x40 to MAX86176 register 0x86	AB 00
		AA 44 06 01 00	Enable MAX86176 AFE.	AB 00

		(CMD_DELAY = 500ms)		
READING SAMPLES	Host repeats the following commands every 40ms			
	2.1	AA 12 00	Read the number of samples available.	AB 00 01
	2.2	AA 12 01	Read the data stored in the FIFO	AB 00 ([sample counter byte] [raw data])x(number of samples)
STOP	Host disables sensors and algorithm:			
	3.1	AA 44 06 00 (CMD_DELAY = 200ms)	Disable the MAX86176 AFE.	AB 00

The SensorHub PPG raw data output FIFO format is listed below. The Type is defined in Table 12 of the MAX86176 Datasheet.

Table 14. SensorHub PPG Raw Data Output FIFO Format

DATA SOURCE (24 bytes)	DATA ITEM	# OF BYTES (MSB FIRST)	DESCRIPTION
MAX86176 PPG Data (18 Bytes)	Green (PD1)	3	[23:20]Type [19:0]: Green1
	Green (PD2)	3	[23:20]Type [19:0]: Green2
	IR (PD1)	3	[23:20]Type [19:0]: IR1
	IR (PD2)	3	[23:20]Type [19:0]: IR2
	Red (PD1)	3	[23:20]Type [19:0]: Red1
	Red (PD2)	3	[23:20]Type [19:0]: Red2

7 Configuring a New MAX32674C

The MAX32674C is not pre-programmed with the application algorithm. See the section “Application .msbl Programming Sequence” to flash the latest .msbl application algorithm to the MAX32674.

The default settings for the MFIO pin must be configured once before the .msbl is flashed. The table below configures and saves the MFIO pin to P0.0.

Table 15. Configure the MFIO pin and Save the Configuration

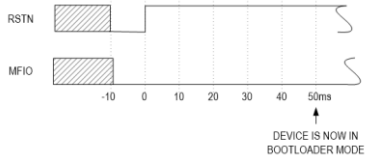
	#	HOST COMMAND (HEX)	COMMAND DESCRIPTION	RESPONSE (HEX)
CONFIGURE THE MFIO PIN	Host must initialize the factory fresh MAX32674C once.			
	1.0		Reset to bootloader mode	
	1.1	AA 01 08	Stay in bootloader mode. This command must be sent within 20ms after reset.	AB AA
	1.2	AA 02 00 (optional)	Read the operating mode	AB AA 08 bootloader mode
	1.3	AA 82 01 00 01	Configure bootloader to use the MFIO pin to determine if the device enters bootloader or application mode after a reset.	AB AA
	1.4	AA 82 01 01 00	Configure the MFIO pin to P0.0	AB AA
	1.5	AA 82 01 02 00	Configure the MFIO pin to active low for bootloader mode.	AB AA
	1.6	AA 82 00 (CMD_DELAY = 300ms)	Save the configuration	AB AA

	1.7	Reset (see the next section on how to reset)	The MAX32674C must be reset to store the configure	

8 Application .msbl Programming Sequence

To program the MAX32674C application .msbl, the host microprocessor places the MAX32674C into bootloader mode and the host may implement the software to flash the .msbl file (I2C, MFIO, RSTN lines are connected between the host and the MAX32674C). The MAX32674C 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 contents, might be different for the latest .msbl, but the locations of these values in the .msbl file remain the same; The example below used the data from the v50.1.0 .msbl file. 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 16. Annotated I2C Trace for Flashing the Application

HOST COMMAND	COMMAND DESCRIPTION	READ MAX32674C RESPONSE	RESPONSE DESCRIPTION
Sequence the MAX32674C to enter bootloader mode. *  *Figure 9. Sequence to enter bootloader mode.*			
0xAA 0x02 0x00†	Read mode.	0xAB 0xAA 0x08	No error. Mode is bootloader.
0xAA 0x81 0x01†	Read bootloader page size.	0xAB 0xAA 0x20 0x00	No error. Page size is 8192.
0xAA 0x80 0x02 0x00 0x*21**	Bootloader flash. Set the “number of pages” to 33 based on the value at byte *0x44* from the application .msbl file.	0xAB 0xAA	No error.
00000044 ba 46 34 2b 21 00 00 20 04 00 00 00 c7 f4 20 8b *Figure 10. Page number byte 0x44 from the .msbl file.*			
0xAA 0x80 0x00 *0x8E 0XA2 0x9D 0x1A 0xE2 0x8F 0x7F 0x25 0x5E 0x0B 0x91**	Bootloader flash. Set the initialization vector bytes to the *0x28 to 0x32* values from the .msbl file.	0xAB 0xAA	No error.
00000020 00 00 00 00 00 00 00 00 8e a2 9d 1a e2 8f 7f 25 00000032 5e 0b 91 00 0d e8 f8 12 7e 2e 8e d8 a9 a3 f1 60 *Figure 11. Initialization vector bytes 0x28 to 0x32 from the .msbl file.*			
0xAA 0x80 0x01 *0x0D 0xE8 0xF8 0x12 0x7E 0x2E 0x8E 0xD8 0xA9 0xA3 0xF1 0x60 0xBA 0x46 0x34 0x2B**	Bootloader flash. Set the authentication bytes to the *0x34 to 0x43* values from the .msbl file.	0xAB 0xAA	No error.

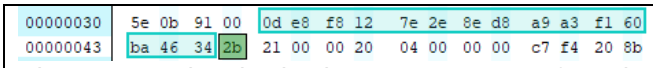


Figure 12. Authentication bytes 0x34 to 0x43 from the .msbl file.

0xAA 0x80 0x03* (CMD_DELAY = 1400ms)	Bootloader flash. Erase application.	0xAB 0xAA	No error.
0xAA 0x80 0x04 *0xC7 0xF4 0x20 ... 0xD4 0x75 0xBF** (CMD_DELAY = 680ms)	Bootloader flash. Send page bytes *0x4C to 0x205B* from the .msbl file (8028 bytes).	0xAB 0xAA	No error.

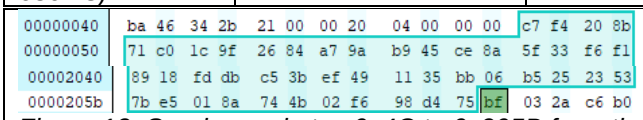


Figure 13. Send page bytes 0x4C to 0x205B from the .msbl file.

0xAA 0x80 0x04 *0x03 0x2A 0xC6 ... 0xA8 0xEE 0xFC** (CMD_DELAY = 680ms)	Bootloader flash. Send page bytes *0x205C to 0x406B* from the .msbl file (8028 bytes).	0xAB 0xAA	No error.
0xAA 0x80 0x04 *0x39 0x43 0x5B ... 0x12 0x3E 0x1B** (CMD_DELAY = 680ms)	Bootloader flash. Send page bytes *0x406C to 0x607B* from the .msbl file (8028 bytes).	0xAB 0xAA	No error.
0xAA 0x80 0x04 *0x39 0xAC 0xB8 ... 0x14 0x6F 0x13** (CMD_DELAY = 680ms)	Bootloader flash. Send page bytes *0x607C to 0x808B* from the .msbl file (8028 bytes).	0xAB 0xAA	No error.
0xAA 0x80 0x04 *0xAA 0xF3 0x95 ... 0x53 0x08 0xEA** (CMD_DELAY = 680ms)	Bootloader flash. Send page bytes *0x808C to 0xA09B* from the .msbl file (8028 bytes).	0xAB 0xAA	No error.
0xAA 0x80 0x04 *0xB1 0xBF 0x28 ... 0x89 0x86 0x35** (CMD_DELAY = 680ms)	Bootloader flash. Send page bytes *0xA09C to 0xC0AB* from the .msbl file (8028 bytes).	0xAB 0xAA	No error.
0xAA 0x80 0x04 *0x56 0x26 0x44 ... 0x72 0x1D 0x74** (CMD_DELAY = 680ms)	Bootloader flash. Send page bytes *0xC0AC to 0xE0BB* from the .msbl file (8028 bytes).	0xAB 0xAA	No error.
0xAA 0x80 0x04 *0x95 0x25 0x5E*	Bootloader flash. Send page bytes	0xAB 0xAA	No error.

*... 0x43 0xF0 0x7F** (CMD_DELAY = 680ms)	*0xE0BC to 0x100CB* from the .msbl file.		
0xAA 0x80 0x04 *0x94 0xEC 0xEC ... 0x0B 0x37 0x95** (CMD_DELAY = 680ms)	Bootloader flash. Send page bytes *0x100CC to 0x120DB* from the .msbl file (8028 bytes).	0xAB 0xAA	No error.
0xAA 0x80 0x04 *0xA6 0xEA 0x53 ... 0xC2 0x26 0x87** (CMD_DELAY = 680ms)	Bootloader flash. Send page bytes *0x120DC to 0x140EB* from the .msbl file (8028 bytes).	0xAB 0xAA	No error.
0xAA 0x80 0x04 *0x01 0x87 0x1D ... 0xBE 0xFF 0xEC** (CMD_DELAY = 680ms)	Bootloader flash. Send page bytes *0x140EC to 0x160FB* from the .msbl file (8028 bytes).	0xAB 0xAA	No error.
0xAA 0x80 0x04 *0xEA 0x65 0xD6 ... 0x63 0xCF 0x5F** (CMD_DELAY = 680ms)	Bootloader flash. Send page bytes *0x160FC to 0x1810B* from the .msbl file (8028 bytes).	0xAB 0xAA	No error.
0xAA 0x80 0x04 *0x4F 0x7E 0xEF ... 0x26 0x3D 0x47** (CMD_DELAY = 680ms)	Bootloader flash. Send page bytes *0x1810C to 0x1A11B* from the .msbl file (8028 bytes).	0xAB 0xAA	No error.
0xAA 0x80 0x04 *0xF0 0x64 0x3D ... 0xAA 0x65 0x47** (CMD_DELAY = 680ms)	Bootloader flash. Send page bytes *0x1A11C to 0x1C12B* from the .msbl file (8028 bytes).	0xAB 0xAA	No error.
0xAA 0x80 0x04 *0x54 0xDF 0xAB ... 0xA4 0xA9 0xE5** (CMD_DELAY = 680ms)	Bootloader flash. Send page bytes *0x1C12C to 0x1E13B* from the .msbl file (8028 bytes).	0xAB 0xAA	No error.

0xAA 0x80 0x04 *0xB0 0x0A 0xA5 ... 0xA6 0x24 0xA8**

(CMD_DELAY = 680ms)Bootloader flash. Send page bytes *0x1E13C to 0x2014B* from the .msbl file (8028 bytes).	0xAB 0xAA	No error.

0xAA 0x80 0x04 *0xE6 0x8E 0x35 ... 0x25 0x11 0xD4** (CMD_DELAY = 680ms)*	Bootloader flash. Send page bytes *0x2014C to 0x2215B* from the .msbl file (8028 bytes).	0xAB 0xAA	No error.
0xAA 0x80 0x04 *0xA0 0x88 0x34 ... 0x40 0x82 0x2E** (CMD_DELAY = 680ms)	Bootloader flash. Send page bytes *0x2215C to 0x2416B* from the .msbl file (8028 bytes).	0xAB 0xAA	No error.
0xAA 0x80 0x04 *0xD5 0x2F 0xAC ... 0xAC 0xCE 0x69** (CMD_DELAY = 680ms)	Bootloader flash. Send page bytes *0x2416C to 0x2617B* from the .msbl file (8028 bytes).	0xAB 0xAA	No error.
0xAA 0x80 0x04 *0x06 0x6C 0xF9 ... 0x9B 0xF2 0x91** (CMD_DELAY = 680ms)	Bootloader flash. Send page bytes *0x2617C to 0x2818B* from the .msbl file (8028 bytes).	0xAB 0xAA	No error.
0xAA 0x80 0x04 *01Bx 0xAD 0x09 ... 0xD3 0x7B 0xE6** (CMD_DELAY = 680ms)	Bootloader flash. Send page bytes *0x2818C to 0x2A19B* from the .msbl file (8028 bytes).	0xAB 0xAA	No error.
0xAA 0x80 0x04 *0xF3 0x08 0x6A ... 0x9C 0x46 0x31** (CMD_DELAY = 680ms)	Bootloader flash. Send page bytes *0x2A19C to 0x2C1AB* from the .msbl file (8028 bytes).	0xAB 0xAA	No error.
0xAA 0x80 0x04 *0x1E 0xCE 0x62 ... 0x99 0x60 0xC0** (CMD_DELAY = 680ms)	Bootloader flash. Send page bytes *0x2C1AC to 0x2E1BB* from the .msbl file (8028 bytes).	0xAB 0xAA	No error.
0xAA 0x80 0x04 *0xEF 0x61 0x8D ... 0xD9 0x40 0x92** (CMD_DELAY = 680ms)	Bootloader flash. Send page bytes *0x2E1BC to 0x301CB* from the .msbl file (8028 bytes).	0xAB 0xAA	No error.
0xAA 0x80 0x04 *0xA0 0x25 0x37 ... 0xA7 0x15 0xAA** (CMD_DELAY = 680ms)	Bootloader flash. Send page bytes *0x301CC to 0x321DB* from the .msbl file (8028 bytes).	0xAB 0xAA	No error.

0xAA 0x80 0x04 *0xF4 0xB0 0x84 ... 0x41 0x8B 0x6A** (CMD_DELAY = 680ms)	Bootloader flash. Send page bytes *0x321DC to 0x341EB* from the .msbl file (8028 bytes).	0xAB 0xAA	No error.
0xAA 0x80 0x04 *0x7D 0x02 0x6E ... 0x87 0x83 0x70** (CMD_DELAY = 680ms)	Bootloader flash. Send page bytes *0x341EC to 0x361FB* from the .msbl file (8028 bytes).		No error.
0xAA 0x80 0x04 *0x1E 0x8D 0x35 ... 0x18 0x19 0xD8** (CMD_DELAY = 680ms)	Bootloader flash. Send page bytes *0x361FC to 0x3820B* from the .msbl file (8028 bytes).		No error.
0xAA 0x80 0x04 *0xEE 0x60 0xBA ... 0x5C 0x27 0x13** (CMD_DELAY = 680ms)	Bootloader flash. Send page bytes *0x3820C to 0x3A21B* from the .msbl file (8028 bytes).		No error.
0xAA 0x80 0x04 *0x58 0xF0 0xE9 ... 0xAF 0x7E 0x79** (CMD_DELAY = 680ms)	Bootloader flash. Send page bytes *0x3A21C to 0x3C22B* from the .msbl file (8028 bytes).		No error.
0xAA 0x80 0x04 *0xD3 0x6F 0x26 ... 0x30 0xCB 0x5C** (CMD_DELAY = 680ms)	Bootloader flash. Send page bytes *0x3C22C to 0x3E23B* from the .msbl file (8028 bytes).		No error.
0xAA 0x80 0x04 *0x8F 0xAA 0xB6 ... 0xF6 0x66 0xEF** (CMD_DELAY = 680ms)	Bootloader flash. Send page bytes *0x3E23C to 0x4024B* from the .msbl file (8028 bytes).		No error.
0xAA 0x80 0x04 *0x8A 0x0B 0xD3 ... 0xE2 0x24 0x74** (CMD_DELAY = 680ms)	Bootloader flash. Send page bytes *0x4024C to 0x4225B* from the .msbl file (8028 bytes).		No error.

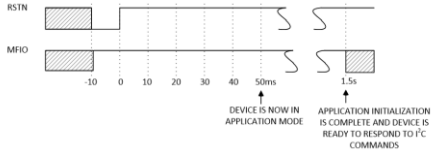


Figure 14. Sequence to enter application mode.

Alternately, the MAX32674C can be commanded to application mode.†

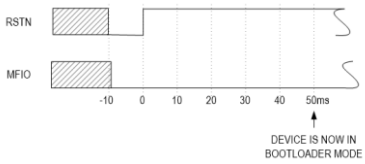
0xAA 0x01 0x00 0x00† (CMD_DELAY = 1.5s)	Set mode to 0x00 for application mode.	0xAB 0xAA	No error.
0xAA 0x02 0x00+	Read mode.	0xAB 0xAA 0x00	No errors. Mode is application.

**Mandatory*

†Recommended

For host micros that have limited RAM or limited I2C message sizes, the AlgoHub allows for the pages to be loaded with among several messages. The table below illustrates how to flash an .msbl file which has 5 pages using the partial page writing. In this example, the partial page size is set to 4000 bytes.

Table 17. Annotated I²C Trace for Flashing the Application using the Partial Page Writing Command

HOST COMMAND	COMMAND DESCRIPTION	READ MAX32674C RESPONSE	RESPONSE DESCRIPTION
Sequence the MAX32674C to enter bootloader mode. *  *Figure 15. Sequence to enter bootloader mode.*			
0xAA 0x02 0x00†	Read mode.	0xAB 0xAA 0x08	No error. Mode is bootloader.
0xAA 0x81 0x01†	Read bootloader page size.	0xAB 0xAA 0x20 0x00	No error. Page size is 8192.
0xAA 0x80 0x02 0x00 *0x05**	Bootloader flash. Set the "number of pages" to *5* based on the value at byte 0x44 from the application .msbl file.	0xAB 0xAA	No error.
0xAA 0x80 0x00 [*11 bytes*]*	Bootloader flash. Set the initialization vector bytes to the *0x28 to 0x32* values from the .msbl file.	0xAB 0xAA	No error.
0xAA 0x80 0x01 [*16 bytes*]*	Bootloader flash. Set the authentication bytes to the *0x34 to 0x43* values from the .msbl file.	0xAB 0xAA	No error.
0x80 0x06 0x0F 0xA0*	Set partial page load size as 4000 (0x0FA0)	0xAB 0xAA	No error.
0xAA 0x80 0x03* (CMD_DELAY = 1400ms)	Bootloader flash. Erase application.	0xAB 0xAA	No error.

0xAA 0x80 0x04 [*4000 bytes*]* (CMD_DELAY = 680ms)	Bootloader flash. Send first partial bytes of the first page, *0x4C to 0xFEB* from the .msbl file.	0xAB 0xAA	o error.
0xAA 0x80 0x04 [*4000 bytes*]* (CMD_DELAY = 680ms)	Bootloader flash. Send second partial bytes of the first page, *0xFEC to 0x1F8B* from the .msbl file.	0xAB 0xAA	No error.
0xAA 0x80 0x04 [*208 bytes*]* (CMD_DELAY = 680ms)	Bootloader flash. Send the remaining partial page bytes of the first page, *0x1F8C to 0x205B* from the .msbl file	0xAB 0xAA	No error.
0xAA 0x80 0x04 [*4000 bytes*]* (CMD_DELAY = 680ms)	Bootloader flash. Send first partial bytes of the second page, *0x205C to 0x2FFB* from the .msbl file.	0xAB 0xAA	No error.
0xAA 0x80 0x04 [*4000 bytes*]* (CMD_DELAY = 680ms)	Bootloader flash. Send second partial bytes of the second page, *0x2FFC to 0x3F9B* from the .msbl file.	0xAB 0xAA	No error.
0xAA 0x80 0x04 [*208 bytes*]* (CMD_DELAY = 680ms)	Bootloader flash. Send the remaining partial page bytes of the second page, *0x3F9C to 0x406B* from the .msbl file	0xAB 0xAA	No error.
0xAA 0x80 0x04 [*4000 bytes*]* (CMD_DELAY = 680ms)	Bootloader flash. Send first partial bytes of the third page, *0x406C to 0x500B* from the .msbl file.	0xAB 0xAA	No error.
0xAA 0x80 0x04 [*4000 bytes*]* (CMD_DELAY = 680ms)	Bootloader flash. Send second partial bytes of the third page, *0x500C to 0x5FAB* from the .msbl file.	0xAB 0xAA	No error.
0xAA 0x80 0x04 [*208 bytes*]* (CMD_DELAY = 680ms)	Bootloader flash. Send the remaining partial page bytes of the third page, *0x5FAC to 0x607B* from the .msbl file	0xAB 0xAA	No error.
0xAA 0x80 0x04 [*4000 bytes*]*	Bootloader flash. Send first partial	0xAB 0xAA	No error.

(CMD_DELAY = 680ms)	bytes of the fourth page, *0x607C to 0x701B* from the .msbl file.		
0xAA 0x80 0x04 [*4000 bytes*]* (CMD_DELAY = 680ms)	Bootloader flash. Send second partial bytes of the fourth page, *0x701C to 0x7FBB* from the .msbl file.	0xAB 0xAA	No error.
0xAA 0x80 0x04 [*208 bytes*]* (CMD_DELAY = 680ms)	Bootloader flash. Send the remaining partial page bytes of the fourth page, *0x7FBC to 0x808B* from the .msbl file	0xAB 0xAA	No error.
0xAA 0x80 0x04 [*4000 bytes*]* (CMD_DELAY = 680ms)	Bootloader flash. Send first partial bytes of the last page, *0x808C to 0x902B* from the .msbl file.	0xAB 0xAA	No error.
0xAA 0x80 0x04 [*4000 bytes*]* (CMD_DELAY = 680ms)	Bootloader flash. Send second partial bytes of the last page, *0x902C to 0x9FCB* from the .msbl file.	0xAB 0xAA	No error.
0xAA 0x80 0x04 [*208 bytes*]* (CMD_DELAY = 680ms)	Bootloader flash. Send the remaining partial page bytes of the last page, *0x9FCC to 0xA09B* from the .msbl file	0xAB 0xAA	No error.

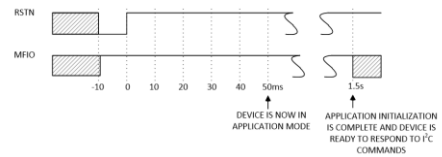


Figure 16. Sequence to enter application mode.

Alternately, the MAX32674C can be commanded to application mode.†			
0xAA 0x01 0x00 0x00† (CMD_DELAY = 1.5s)	Set mode to 0x00 for application mode.	0xAB 0xAA	No error.
0xAA 0x02 0x00+	Read mode.	0xAB 0xAA 0x00	No errors. Mode is application.

**Mandatory*

†*Recommended*

9 SpO_2 Coefficients for Final Form Factor

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)

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

10 APIs for Sleep, Shutdown

Summarized below are the commands and methods to place the MAX32674C, AFE, accelerometer into sleep or shutdown.

Table 18. Sleep, Shutdown I²C Message Protocol Definitions

COMMAND NAME	HOST COMMAND TO MAX32674C	DESCRIPTION
MAX32674C sleeps when idle and MFIO is low		MAX32674C uses deep-sleep for low-powered mode when idle and MFIO low.
MAX32674C shutdown	0xAA 0x01 0x00 0x01	Place the MAX32674C into shutdown. Restart by power cycling or pulsing RSTN.
MAX32674C hard reset	Use MFIO and RSTN pins according to section 2.	
WDT in MAX32674C		Not implemented.
MAX86176 AFE reset. Use AFE write register command. SensorHub.	0xAA 0x40 0x06 0x10 0x01	Write 0x01 to MAX86176 register 0x10 (System Configuration) to put the reset the MAX86176.
MAX86176 AFE shutdown. Use AFE write register command. SensorHub.	0xAA 0x40 0x06 0x10 0x02	Write 0x02 to MAX86176 register 0x10 (System Configuration) to put the MAX86176 into shutdown (SHDN) mode.
LIS2DS12 wake on activity or wake on double/single tap. Use accel write register command. SensorHub.	0xAA 0x40 0x04 [reg_addr] [value]	See AN4748 LIS2DS12 for register settings. Accel interrupt pin can be connected to the host GPIO.

11 MAX32674C Processing Capabilities

The MAX32674C IC hardware is the same as the MAX32670.

- MIPS: Arm Cortex-M4 with FPU: 1.27 Dhrystone MIPS/MHz

- RAM: 160KB SRAM
- Flash: 384KB Flash Memory with Error Correction
- CPU Frequency: 100MHz

12 Heart Rate Algorithm Perfomance

Table 19. 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.

13 SpO2 Algorithm Perfomance

Table 20. SpO2 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”

14 References

MAXREFDES104# website files (Design Files, Firmware, Software, User Guide, Sample Host Code):

MAXREFDES104#: Wrist-Based SpO2, HR, ECG, temperature Health Sensor Platform

Frequently Asked Questions: Maxim Support Center

Validation and Performance of a Wearable Heart rate Monitoring Algorithm

Guidelines For Spo2 Measurement

Revision History

REVISION NUMBER	REVISION DATE	DESCRIPTION	PAGES CHANGED
0.01	01/21	Initial release	—
0.02	5/21	Added section 1.1-1.3, 2-4, 6-10, figure 2, 7, 9-16, table 5-10. Updated section 1-5, figure 1, table 2, 7.	all
0.03	8/21	Added section 2.2, 5.4, 6.4, 7 table 7, 13, 14, 15. Updated section 1.3, 2.1, figures 1-5,8, table 1, 2, 11, 16, 17	all