

Best-in-Class Optical Pulse Oximeter and Heart-Rate Sensor for Wearable Health

**MAX86140/
MAX86141**

General Description

The MAX86140/MAX86141 are ultra-low-power, completely integrated, optical data acquisition systems. On the transmitter side, the MAX86140/MAX86141 have three programmable high-current LED drivers that can be configured to drive up to six LEDs using an external 3x2:1 mux. With two MAX86140/MAX86141 devices working in controller-target mode, the LED drivers can drive up to twelve LEDs using an external 3x2:1 mux. On the receiver side, MAX86140 consists of a single optical readout channel, while the MAX86141 has two optical readout channels that can operate simultaneously. The devices have low-noise signal conditioning analog front-end (AFE) including 19-bit ADC, an industry-lead ambient light cancellation (ALC) circuit, and a picket fence detect and replace function. Due to the low power consumption, compact size, ease of use, and industry-lead ambient light rejection capability of MAX86140/MAX86141, the devices are ideal for a wide variety of optical-sensing applications, such as pulse oximetry and heart rate detection.

The MAX86140/MAX86141 operate on a 1.8V main supply voltage and a 3.1V to 5.5V LED driver supply voltage. Both devices support a standard SPI compatible interface and fully autonomous operation. Each device has a large 128-word built-in FIFO. The MAX86140/MAX86141 is available in compact wafer-level package (WLP) (2.048 x 1.848mm) with 0.4mm ball pitch.

Applications

- Wearable Devices for Fitness, Wellness and Medical Applications
- Optimized for Wrist, Finger, Ear, and Other Locations
- Optimized Performance to Detect
 - Optical Heart Rate
 - Oxygen Saturation (SpO₂)
 - Muscle Oxygen Saturation (SmO₂ and StO₂)

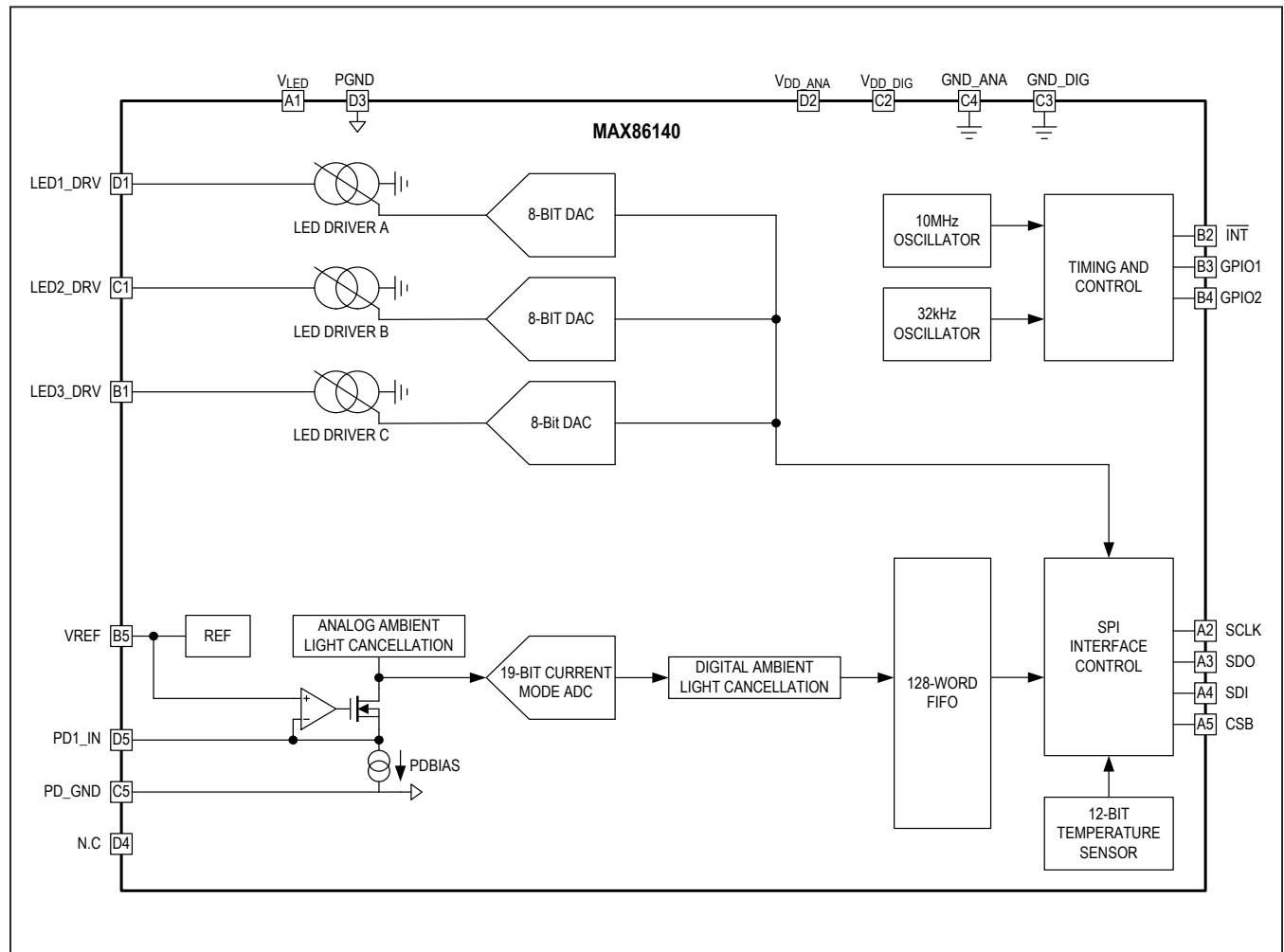
Benefits and Features

- Complete Single and Dual-Channel Optical Data Acquisition System
- Optimized Architecture for Transmissive and Reflective Heart Rate or SpO₂ Monitoring
- Low Dark Current Noise of < 50pA RMS (Sample to Sample Variance)
- Lower Effective Dark Current Noise Achievable Through Multiple Sample Modes and On-Chip Averaging
- High-Resolution, 19-Bit Charge Integrating ADC
- Three Low-Noise, 8-Bit LED Current DACs
- Excellent Dynamic Range > 89dB in White Card Loop-Back Test (Sample-to-Sample Variance)
- Dynamic Range Extendable to > 104dB for SpO₂ and > 110dB for HRM with Multiple Sample Modes and On-Chip Averaging
- Excellent Ambient Range and Rejection Capability
 - > 100µA Ambient Photodetector Current
 - > 70dB Ambient Rejection at 120Hz
- Ultra-Low-Power Operation for Wearable Devices
 - Low-Power Operation, Optical Readout Channel < 10µA (typ) at 25sps
 - Short Exposure Integration Period of 14.8µs, 29.4µs, 58.7µs, 117.3µs
 - Low Shutdown Current = 0.6µA (typ)
- Rejection of Fast Ambient Transients
- Miniature 2.048 x 1.848mm, 5 x 4 0.4mm Ball Pitch WLP
- -40°C to +85°C Operating Temperature Range

[Ordering Information](#) appears at end of data sheet.

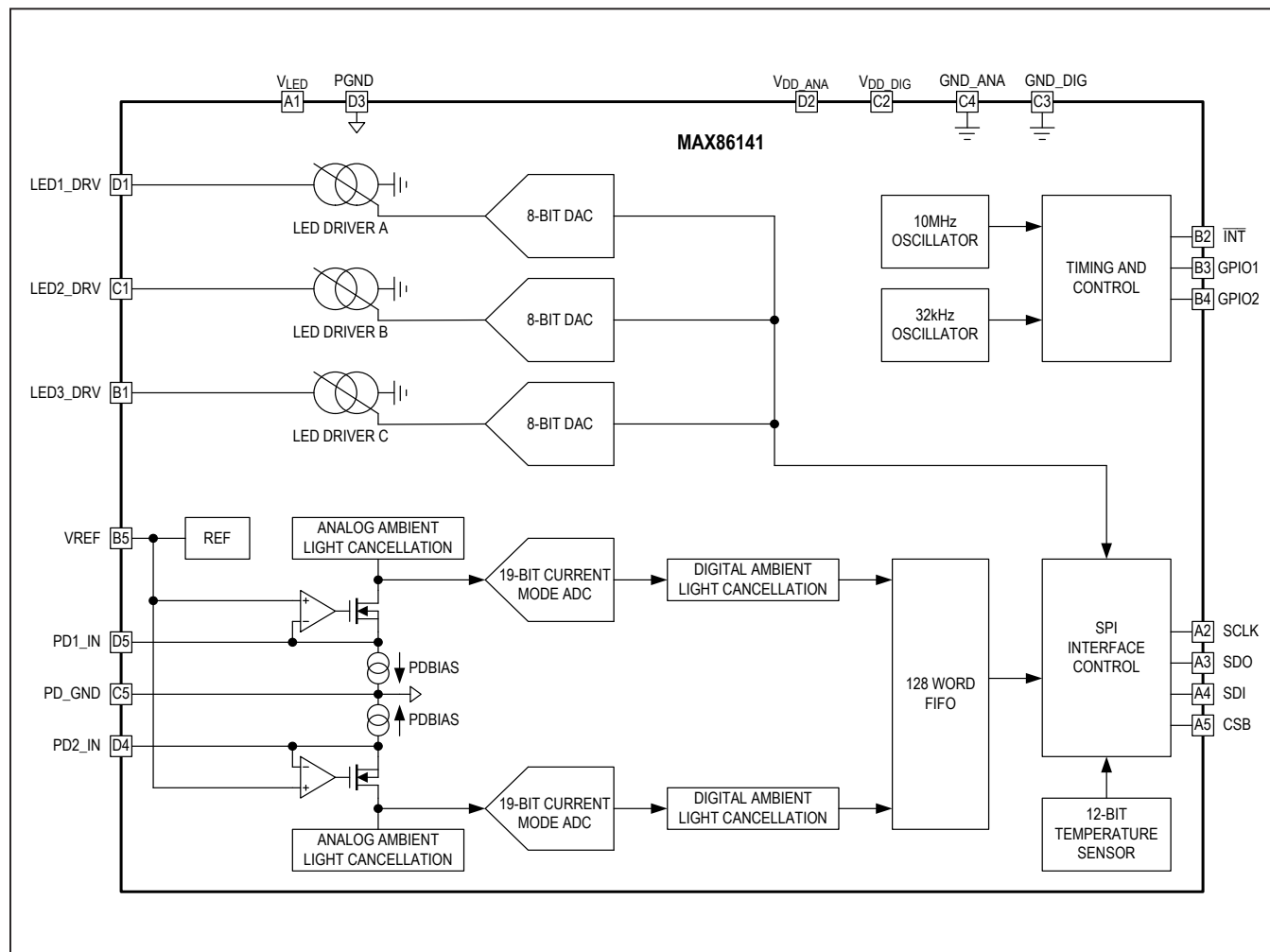
Detailed Block Diagrams

MAX86140



Detailed Block Diagrams (continued)

MAX86141



Absolute Maximum Ratings

| | | | |
|---|----------------------------------|--|-----------------|
| V _{DD_ANA} to GND_ANA..... | -0.3V to +2.2V | PD1_IN to GND_ANA..... | -0.3V to +2.2V |
| V _{DD_DIG} to GND_ANA..... | -0.3V to +2.2V | PD2_IN to GND_ANA..... | -0.3V to +2.2V |
| V _{DD_ANA} to V _{DD_DIG} | -0.3V to +0.3V | PD_GND to GND_ANA..... | -0.3V to +0.3V |
| PGND to GND_ANA..... | -0.3V to +0.3V | All other pins to GND_ANA..... | -0.3V to +2.2V |
| SCLK, SDO, SDI, CSB, INT, GPIO1, GPIO2 to GND_ANA..... | -0.3V to +6.0V | Output Short-Circuit Duration..... | Continuous |
| GND_DIG to GND_ANA..... | -0.3V to +0.3V | Continuous Input Current Into Any Pin (except LEDx_DRV Pins)..... | ±20mA |
| V _{LED} to PGND..... | -0.3V to +6.0V | Continuous Power Dissipation (WLP (derate 5.5mW/°C above +70°C))..... | 440mW |
| LED1_DRV to PGND..... | -0.3V to V _{LED} + 0.3V | Operating Temperature Range..... | -40°C to +85°C |
| LED2_DRV to PGND..... | -0.3V to V _{LED} + 0.3V | Storage Temperature Range..... | -40°C to +105°C |
| LED3_DRV to PGND..... | -0.3V to V _{LED} + 0.3V | | |

Stresses beyond those listed under “Absolute Maximum Ratings” may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of the specifications is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

Package Information

5 x 4 WLP

| PACKAGE CODE | N201A2+1 |
|--|--|
| Outline Number | 21-100134 |
| Land Pattern Number | Refer to Application Note 1891 |
| Thermal Resistance, Four-Layer Board: | |
| Junction to Ambient (θ_{JA}) | 55.49°C/W |
| Junction to Case (θ_{JC}) | N/A |

For the latest package outline information and land patterns (footprints), go to www.maximintegrated.com/packages. Note that a “+”, “#”, or “-” in the package code indicates RoHS status only. Package drawings may show a different suffix character, but the drawing pertains to the package regardless of RoHS status.

Package thermal resistances were obtained using the method described in JEDEC specification JESD51-7, using a four-layer board. For detailed information on package thermal considerations, refer to www.maximintegrated.com/thermal-tutorial.

Electrical Characteristics

(V_{DD_ANA} = 1.8V, V_{DD_DIG} = 1.8V, V_{LED} = 5.0V, PPGx_ADC_RGE = 16μA, PPG_SR = 1024sps, PPG_TINT = 14.8μs, LED_SETLNG = 6μs, LEDx_RGE = 31mA, C_{PD} = 65pF, PDBIASx = 0x1, I_{exposure} = 1μA, T_A = +25°C, min/max are from T_A = -40°C to +85°C, unless otherwise noted. (Note 1, 2))

| PARAMETER | SYMBOL | CONDITIONS | MIN | TYP | MAX | UNITS |
|--------------------------------------|------------------|--|---------------|-------|------|--------|
| Readout Channel | | | | | | |
| ADC Resolution | | | | 19 | | bits |
| ADC Full Scale Input Current | | PPGx_ADC_RGE = 0x0 | | 4.0 | | μA |
| | | PPGx_ADC_RGE = 0x1 | | 8.0 | | |
| | | PPGx_ADC_RGE = 0x2 | | 16.0 | | |
| | | PPGx_ADC_RGE = 0x3 | | 32.0 | | |
| ADC Integration Time | t _{INT} | PPG_TINT = 0x0 | | 14.8 | | μs |
| | | PPG_TINT = 0x1 | | 29.4 | | |
| | | PPG_TINT = 0x2 | | 58.7 | | |
| | | PPG_TINT = 0x3 | | 117.3 | | |
| Minimum PPG Sample Rate | | PPG_SR = 0x0A | | 8 | | sps |
| Maximum PPG Sample Rate | | PPG_SR = 0x13 | | 4096 | | sps |
| Sample Rate Error | | From nominal as indicated in the PPG_SR table | -2 | | +2 | % |
| Maximum DC Ambient Light Rejection | ALR | ALC = on, ALC_OVF = 1 | | 200 | | μA |
| AC Ambient Light Rejection | AC_ALRR | ALC = on, I _{ambient} = 1μA DC with ±0.4μA pk-pk 120Hz Sinwave | | 70 | | dB |
| DC Ambient Light Rejection | | ALC = on, I _{ambient} modulated between 0μA and 30μA, LED_SETLNG = 12μs, PPG_TINT = 117.3μs | | 0.5 | | nA |
| Dark Current Offset | DC_O | ALC = ON, PDBIASx = 0x0, ADD_OFFSET = 1 | | ±1 | | Counts |
| Dark Current Input Referred Noise | | PPG_TINT = 14.8μs | | 262 | | pArms |
| | | PPG_TINT = 29.4μs | | 128 | | |
| | | PPG_TINT = 58.7μs | | 83 | | |
| | | PPG_TINT = 117.3μs | | 56 | | pArms |
| Maximum Photodiode Input Capacitance | C _{pd} | I _{ambient} = 0μA, less than 1nA of code shift | PDBIASx = 0x1 | 65 | | pF |
| | | | PDBIASx = 0x5 | 130 | | |
| | | | PDBIASx = 0x6 | 260 | | |
| | | | PDBIASx = 0x7 | 520 | | |
| VDD DC PSR | | I _{ambient} = 0μA, V _{DD_ANA} = V _{DD_DIG} = 1.7V to 2.0V | -560 | -330 | +560 | LSB/V |
| LED Driver | | | | | | |
| LED Current Resolution | | | | 8 | | Bits |
| Driver DNL | | LEDx_RGE = 124mA | -1 | | 1 | LSB |
| Driver INL | | LEDx_RGE = 124mA | | 0.6 | | LSB |

Electrical Characteristics (continued)

($V_{DD_ANA} = 1.8V$, $V_{DD_DIG} = 1.8V$, $V_{LED} = 5.0V$, $PPG_ADC_RGE = 16\mu A$, $PPG_SR = 1024sps$, $PPG_TINT = 14.8\mu s$, $LED_SETLNG = 6\mu s$, $LEDx_RGE = 31mA$, $C_{PD} = 65pF$, $PDBIASx = 0x1$, $I_{exposure} = 1\mu A$, $T_A = +25^\circ C$, min/max are from $T_A = -40^\circ C$ to $+85^\circ C$, unless otherwise noted. (Note 1, 2))

| PARAMETER | SYMBOL | CONDITIONS | MIN | TYP | MAX | UNITS |
|--|----------------------------------|---|-------------------|-----|---------|------------|
| Full Scale LED Current | I_{LED} | $LEDx_DRV = 0xFF$ | $LEDx_RGE = 0x0$ | 31 | | mA |
| | | | $LEDx_RGE = 0x1$ | 62 | | |
| | | | $LEDx_RGE = 0x2$ | 93 | | |
| | | | $LEDx_RGE = 0x3$ | 117 | 124 129 | |
| Minimum output voltage | V_{OL} | $LEDx_DRV = 0xFF$, 95% of the desired LED current | $LEDx_RGE = 0x0$ | 160 | 253 | mV |
| | | | $LEDx_RGE = 0x1$ | 317 | | |
| | | | $LEDx_RGE = 0x2$ | 495 | | |
| | | | $LEDx_RGE = 0x3$ | 700 | | |
| LED Driver DC PSR | | $V_{LEDx_DRV} = 0.9V$, $LEDx_DRV = 0xFF$, $V_{LED} = 3.1V$ to $5.5V$, $LEDx_RGE = 124mA$ | | -1 | +400 | $\mu A/V$ |
| | | $V_{DD_ANA} = V_{DD_DIG} = 1.7V$ to $2.0V$, $T_A = +25^\circ C$, $LEDx_DRV = 0xFF$ | | 110 | 1410 | |
| LED1 Driver Compliance Interrupt | $LED1_{COMP}$ | | | 180 | | mV |
| Internal Die Temperature Sensor | | | | | | |
| Temperature Sensor Accuracy | | $T_A = +25^\circ C$ | | 1 | | $^\circ C$ |
| Temperature Sensor Minimum Range | | Temperature error $< 5^\circ C$ | | -40 | | $^\circ C$ |
| Temperature Sensor Maximum Range | | Temperature error $< 5^\circ C$ | | 85 | | $^\circ C$ |
| Temperature ADC Acquisition Time | | | | 29 | | ms |
| Power Supply | | | | | | |
| Power Supply Voltage | V_{DD_ANA} , V_{DD_DIG} | Verified during PSRR Test | 1.7 | 1.8 | 2.0 | V |
| LED Supply Voltage | V_{LED} | Verified during PSRR Test | 3.1 | | 5.5 | V |

Electrical Characteristics (continued)

(V_{DD_ANA} = 1.8V, V_{DD_DIG} = 1.8V, V_{LED} = 5.0V, PPG_x_ADC_RGE = 16μA, PPG_SR = 1024sps, PPG_TINT = 14.8μs, LED_SETLNG = 6μs, LED_x_RGE = 31mA, C_{PD} = 65pF, PDBIAS_x = 0x1, I_{exposure} = 1μA, T_A = +25°C, min/max are from T_A = -40°C to +85°C, unless otherwise noted. (Note 1, 2))

| PARAMETER | SYMBOL | CONDITIONS | | MIN | TYP | MAX | UNITS |
|--|------------------|---|-----------------|-----|-------|------|-------|
| VDD Supply Current | I _{DD} | MAX86140, Single LED Exposure/Sample. PPG_SR = 4096sps, LP_MODE = 0x0, LEDxPA = 0mA | | | 660 | 780 | μA |
| | | MAX86140, Single LED Exposure/Sample, LP_MODE = 0x1, LEDx_DRV = 0mA | PPG_SR = 256sps | | 80 | | |
| | | | PPG_SR = 100sps | | 32 | | |
| | | | PPG_SR = 50sps | | 16 | | |
| | | | PPG_SR = 25sps | | 8.5 | | |
| | | MAX86140, Dual LED Exposure/Sample, PPG_SR = 84sps, LP_MODE = 0x1, LEDx_DRV = 0mA | Single pulse | | 42 | | μA |
| | | | Dual pulse | | 89 | | μA |
| | | MAX86141, Single LED Exposure/Sample. PPG_SR = 4096sps, LP_MODE = 0x0, LEDx_DRV = 0mA | | | 978 | 1170 | μA |
| | | MAX86141, Single LED Exposure/Sample, LP_MODE = 1, LEDx_DRV = 0mA | PPG_SR = 256sps | | 115.5 | | |
| | | | PPG_SR = 100sps | | 46 | | |
| | | | PPG_SR = 50sps | | 23 | | |
| | | | PPG_SR = 25sps | | 11 | | |
| MAX86141, Two LED Exposure/Sample, PPG_SR = 84sps, LP_MODE = 1, LEDx_DRV = 0mA | Single pulse | | 60 | | μA | | |
| | Dual pulse | | 130 | | μA | | |
| Die Temperature mode, PPG_SR = 1sps, Optical channel(s) disabled | | | 8 | | μA | | |
| VLED Supply Current | I _{LED} | Single LED exposure per Sample, PPG_TINT = 117.3μs, Single-Pulse, PPG_SR = 256sps, LEDx_DRV = 0mA | | | 0.22 | | μA |
| | | Single LED exposure per Sample, PPG_TINT = 117.3μs, Single-Pulse, LEDx_DRV = 62mA | PPG_SR = 256sps | | 1880 | | |
| | | | PPG_SR = 100sps | | 735 | | |
| | | | PPG_SR = 50sps | | 370 | | |
| | | Two LED exposure per sample, PPG_TINT = 117.3μs, LEDx_DRV = 62mA, PPG_SR = 84sps | PPG_SR=25sps | | 185 | | |
| | | | Single pulse | | 1240 | | μA |
| Dual pulse | | 2480 | | | | | |
| VDD Current in Shutdown | | T _A = +25°C | | | 0.6 | 2.5 | μA |
| VLED Current in Shutdown | | T _A = +25°C | | | | 1 | μA |

Electrical Characteristics (continued)

($V_{DD_ANA} = 1.8V$, $V_{DD_DIG} = 1.8V$, $V_{LED} = 5.0V$, $PPG_ADC_RGE = 16\mu A$, $PPG_SR = 1024sps$, $PPG_TINT = 14.8\mu s$, $LED_SETLNG = 6\mu s$, $LEDx_RGE = 31mA$, $C_{PD} = 65pF$, $PDBIASx = 0x1$, $I_{exposure} = 1\mu A$, $T_A = +25^\circ C$, min/max are from $T_A = -40^\circ C$ to $+85^\circ C$, unless otherwise noted. (Note 1, 2))

| PARAMETER | SYMBOL | CONDITIONS | MIN | TYP | MAX | UNITS |
|---|---------------|---|--------------|------|-------|---------|
| Digital I/O Characteristics | | | | | | |
| SDO Output Low Voltage | V_{OL_SDO} | $I_{SINK} = 2mA$ | | | 0.4 | V |
| SDO Output High Voltage | V_{OH_SDO} | $I_{SOURCE} = 2mA$ | $V_{DD}-0.4$ | | | V |
| Open-Drain Output Low Voltage | V_{OL_OD} | $I_{SINK} = 6mA$, \overline{INT} , GPIO1, GPIO2 | | | 0.4 | V |
| Input Voltage Low | V_{IL} | SDI, SCLK, CSB, GPIO1, GPIO2 | | | 0.4 | V |
| Input Voltage High | V_{IH} | SDI, SCLK, CSB, GPIO1, GPIO2 | 1.4 | | | V |
| Input Hysteresis | V_{HYS} | SDI, SCLK, CSB | | 330 | | mV |
| | | GPIO1, GPO2 | | 240 | | |
| Input Leakage Current | I_{IN} | $V_{IN} = 0V$, $T_A = +25^\circ C$ (SDI, SCLK, CSB, GPIO1, GPIO2) | | 0.01 | 1 | μA |
| Input Capacitance | C_{IN} | SDI, SCLK, CSB, GPIO1, GPIO2 | | 10 | | pF |
| SPI Timing Charateristics (Note 3) | | | | | | |
| SCLK Frequency | f_{SCLK} | | | | 4 | MHz |
| SCLK Period | t_{CP} | | 125 | | | ns |
| SCLK Pulse Width High | t_{CH} | | 40 | | | ns |
| SCLK Pulse Width Low | t_{CL} | | 40 | | | ns |
| CSB Fall to SCLK Rise Setup Time | t_{CSS0} | To 1 st SCLK rising edge | 20 | | | ns |
| CSB Fall to SCLK Rise Hold Time | t_{CSH0} | Applies to inactive rising edge preceding 1 st rising edge | 5 | | | ns |
| CSB Rise to SCLK Rise Hold Time | t_{CSH1} | Applies to 24th rising edge | 500 | | | ns |
| SCLK Rise to CSB Fall | t_{CSF} | Applies to 24th rising edge | 500 | | | ns |
| CSB Pulse Width High | t_{CSPW} | | 250 | | | ns |
| SDI to SCLK Rise Setup Time | t_{DS} | | 10 | | | ns |
| SDI to SCLK Rise Hold Time | t_{DH} | | 10 | | | ns |
| SCLK Fall to SDO Transition | t_{DOT} | $C_{LOAD} = 50pF$ | | | 35 | ns |
| CSB Fall to SDO Enabled | t_{DOE} | $C_{LOAD} = 0pF$ | 12 | | | ns |
| CSB Rise to SDO Hi-Z | t_{DOZ} | Disable Time | | | 25 | ns |
| GPIO1 External Sync Pulse Width | $t_{PLGPIO1}$ | | 5 | | | μs |
| GPIO2 External Clock Input (Note 4) | f_{GPIO2} | External Sample Reference Clock on GPIO2 | 31900 | | 32868 | Hz |
| GPIO2 External Clock Pulse Width | $t_{PWGPIO2}$ | | 1 | | | μs |

Note 1: All devices are 100% production tested at $T_A = +25^\circ C$. Specifications over temperature limits are guaranteed by Maxim Integrated's bench or proprietary automated test equipment (ATE) characterization.

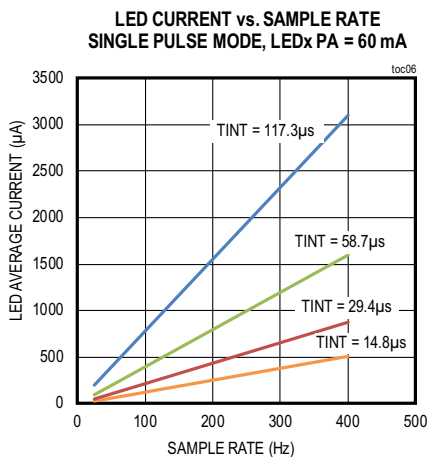
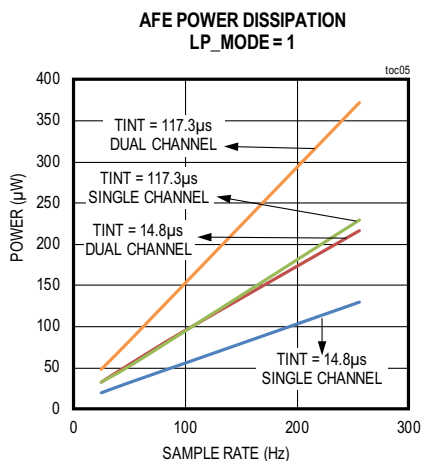
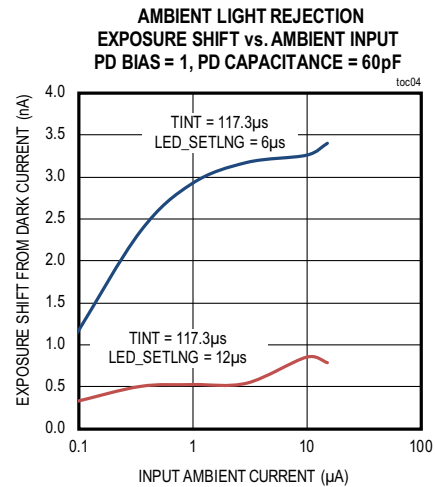
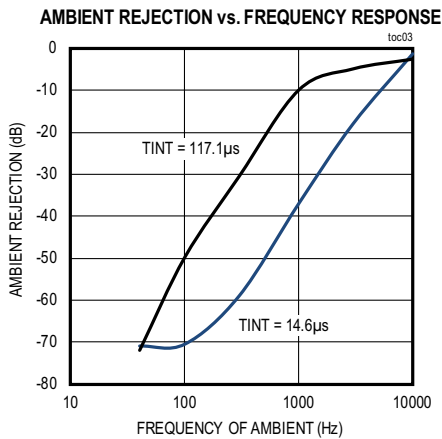
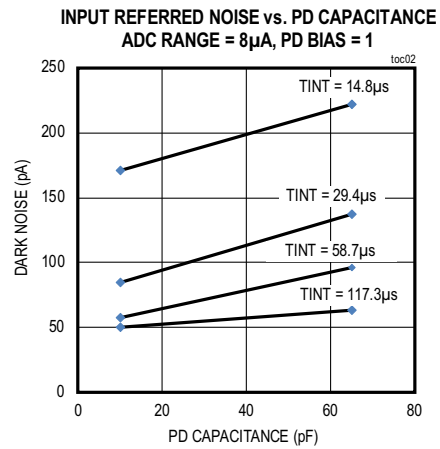
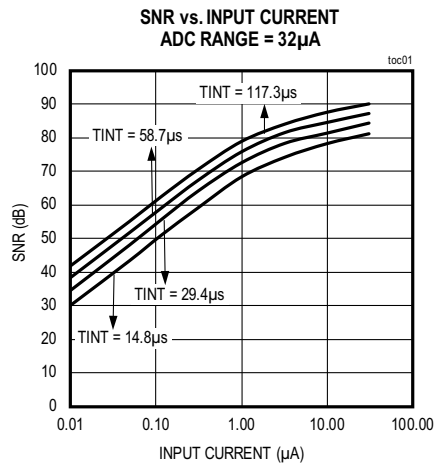
Note 2: All other register settings assumed to be POR status unless otherwise noted.

Note 3: Guaranteed by design. Not production tested.

Note 4: See *Register Map/PPG Configuration 2 (0x12)* section for the sample rate by the external clock frequency. The sample rate shifts if the external clock frequency drifts.

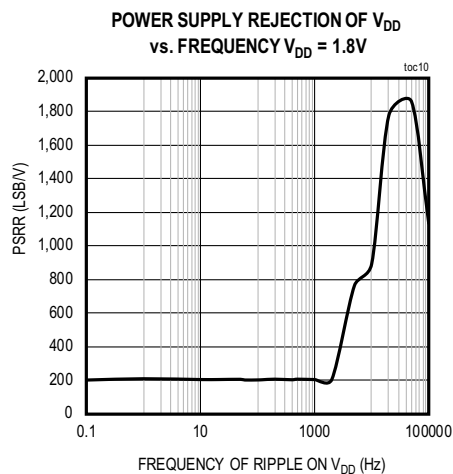
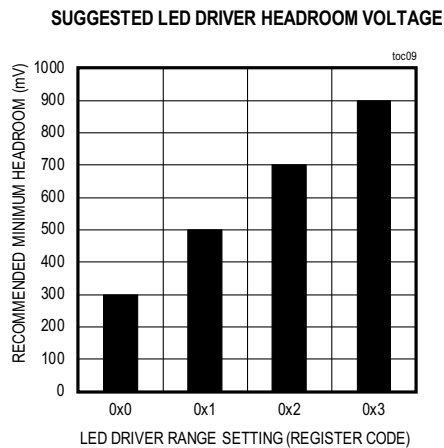
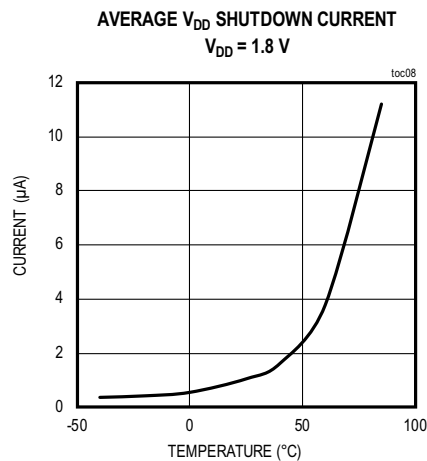
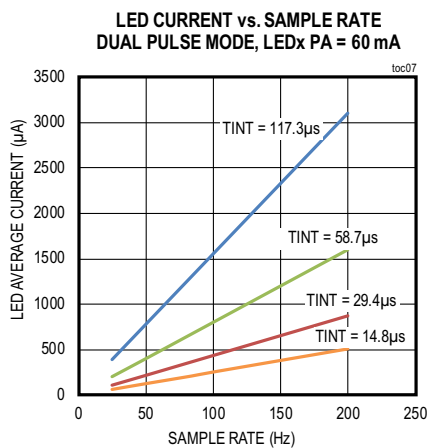
Typical Operating Characteristics

($V_{DD_ANA} = V_{DD_DIG} = 1.8V$, $V_{LED} = 5.0V$, $GND_ANA = GND_DIG = PGND = 0V$, $T_A = +25^\circ C$, unless otherwise noted.)



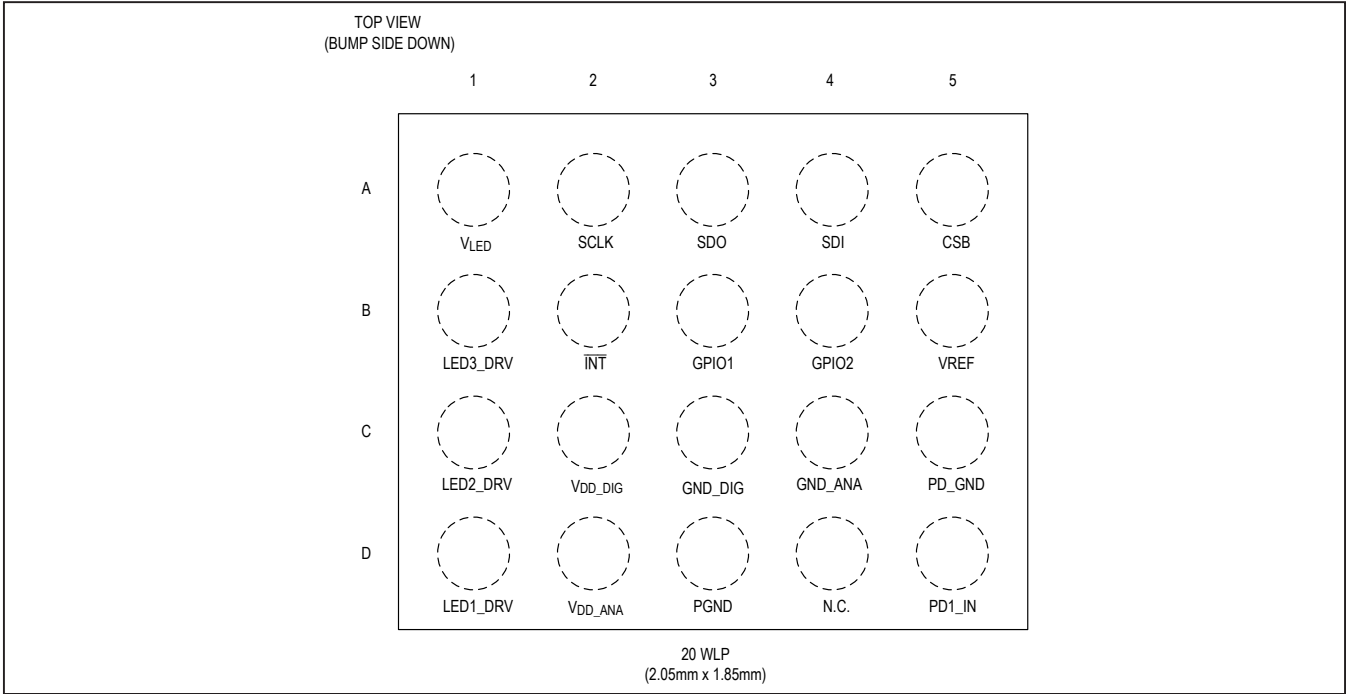
Typical Operating Characteristics (continued)

($V_{DD_ANA} = V_{DD_DIG} = 1.8V$, $V_{LED} = 5.0V$, $GND_ANA = GND_DIG = PGND = 0V$, $T_A = +25^\circ C$, unless otherwise noted.)

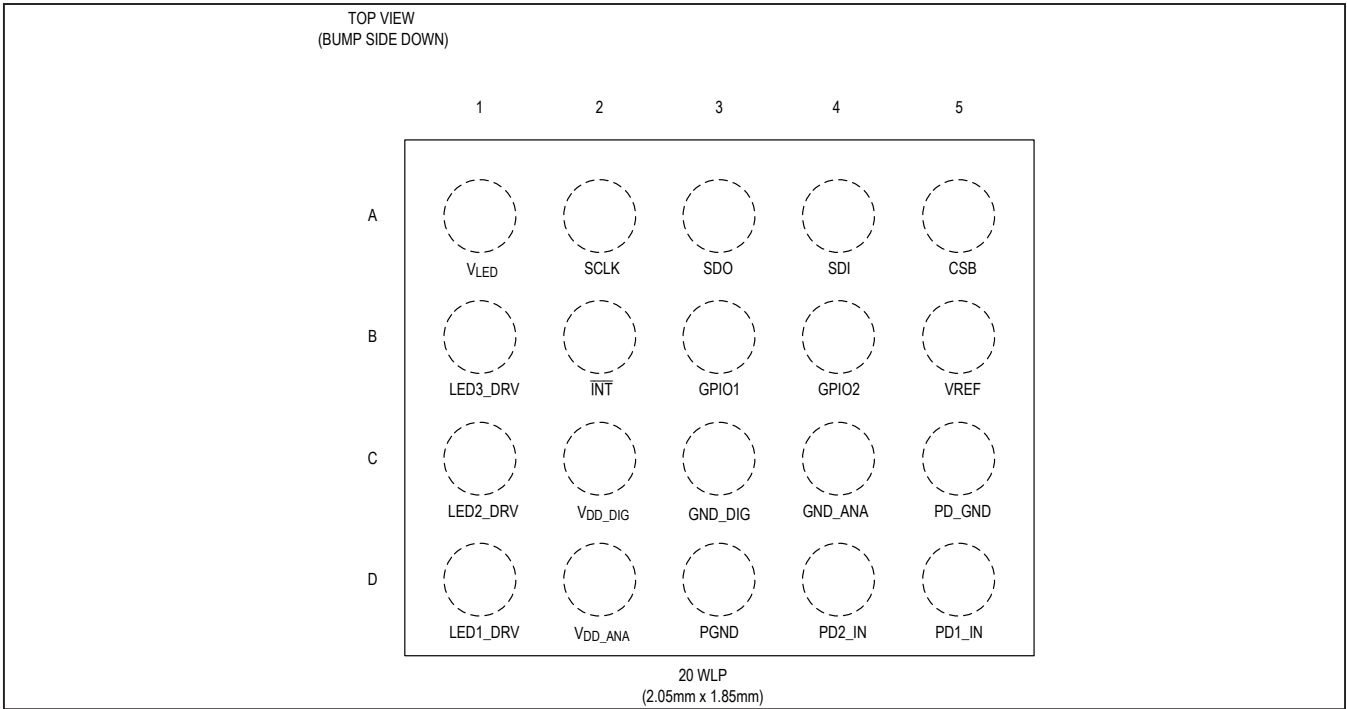


Pin Configurations

MAX86140



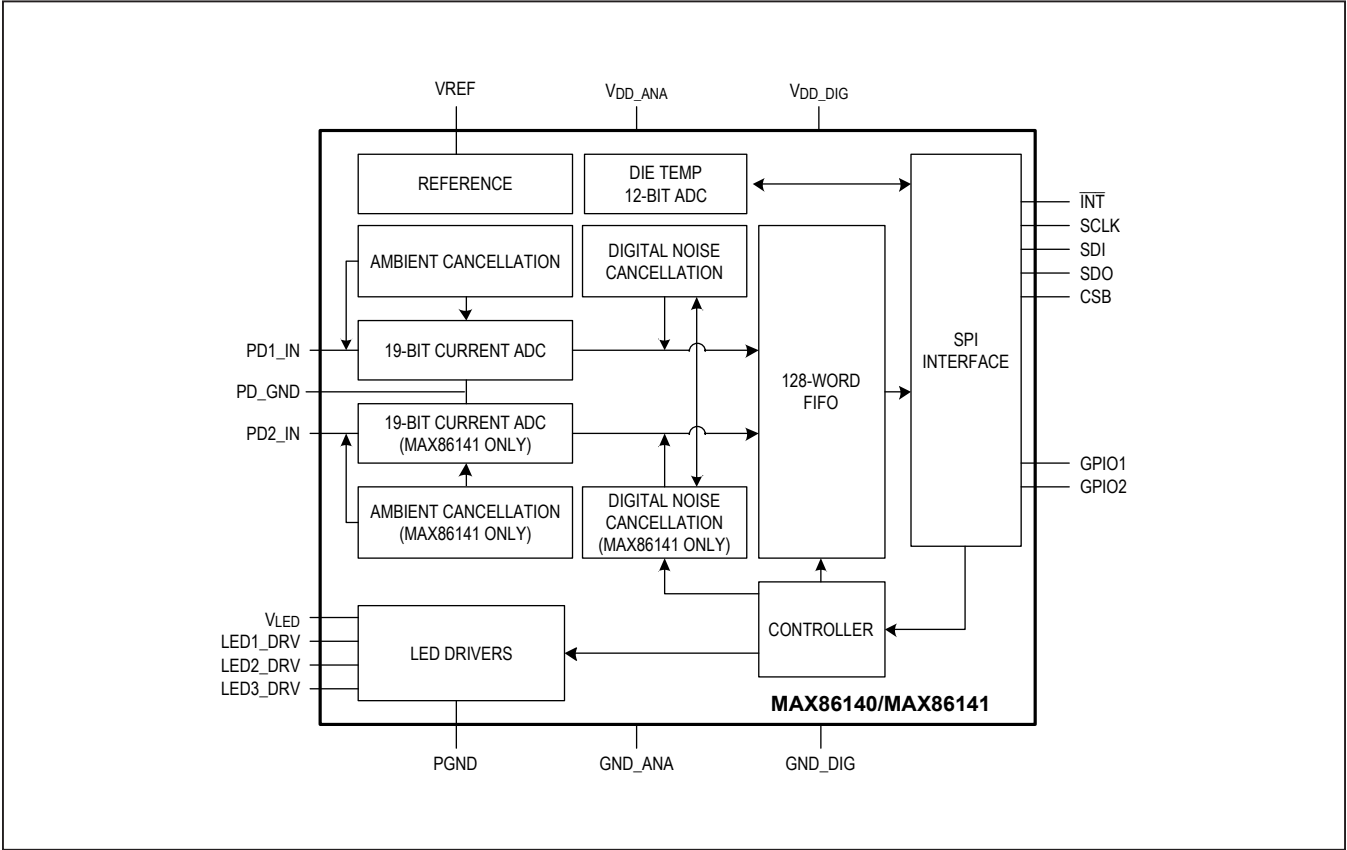
MAX86141



Pin Description

| PIN | | NAME | FUNCTION |
|-------------------|----------|-------------------------|--|
| MAX86140 | MAX86141 | | |
| POWER | | | |
| C2 | C2 | V _{DD_DIG} | Digital Logic Supply. Connect to externally-regulated supply. Bypass to GND_DIG |
| C3 | C3 | GND_DIG | Digital Logic and Digital Pad Return. Connect to PCB Ground. |
| D2 | D2 | V _{DD_ANA} | Analog Supply. Connect to externally-regulated supply. Bypass with a 0.1μF capacitor as close as possible to bump and a 10μF capacitor to GND_ANA. |
| C4 | C4 | GND_ANA | Analog Power Return. Connect to PCB Ground. |
| A1 | A1 | V _{LED} | LED Power Supply Input. Connect to external voltage supply. Bypass with a 10μF capacitor to PGND. |
| D3 | D3 | PGND | LED Power Return. Connect to PCB Ground. |
| CONTROL INTERFACE | | | |
| A2 | A2 | SCLK | SPI Clock |
| A3 | A3 | SDO | SPI Data Output |
| A4 | A4 | SDI | SPI Data Input |
| A5 | A5 | CSB | SPI Chip select |
| B2 | B2 | $\overline{\text{INT}}$ | Interrupt. Programmable open-drain Interrupt output signal pin (active-low). When not used, it can be left unconnected. |
| B3 | B3 | GPIO1 | General Purpose I/O. Open-drain when programmed as output (active-low). When not used, it can be left unconnected. |
| B4 | B4 | GPIO2 | General Purpose I/O. Open-drain when programmed as output (active-low). When not used, it can be left unconnected. |
| OPTICAL PINS | | | |
| — | D4 | PD2_IN | Photodiode Cathode Input. Tie to PD_GND when this pin is not used. |
| D5 | D5 | PD1_IN | Photodiode Cathode Input. Tie to PD_GND when this pin is not used. |
| C5 | C5 | PD_GND | Photodiode Anode |
| D1 | D1 | LED1_DRV | LED Output Driver 1. Connect the LED cathode to LED1_DRV and its anode to the V _{LED} supply. When not used, it can be left unconnected. |
| C1 | C1 | LED2_DRV | LED Output Driver 2. Connect the LED cathode to LED2_DRV and its anode to the V _{LED} supply. When not used, it can be left unconnected. |
| B1 | B1 | LED3_DRV | LED Output Driver 3. Connect the LED cathode to LED3_DRV and its anode to the V _{LED} supply. When not used, it can be left unconnected. |
| REFERENCE | | | |
| B5 | B5 | VREF | Internal Reference Decoupling Point. Bypass with a 1μF capacitor to GND_ANA. |
| N.C. | | | |
| D4 | | N.C. | No Connection. Connect to unconnected PCB pad for mechanical stability. N.C. pins should not be connected to any signal, power, or ground pins. |

Simplified Block Diagram



Detailed Description

The MAX86140/MAX86141 are complete integrated optical data acquisition systems, ideal for optical pulse oximetry and heart rate detection applications. Both parts have been designed for the demanding requirements of mobile and wearable devices and require minimal external hardware components are necessary for integration into a wearable device. They include high-resolution, optical readout signal processing channels with robust ambient light cancellation and high-current LED driver DACs to form a complete optical readout signal chain.

The MAX86140/MAX86141 are fully adjustable through software registers and the digital output data is stored in a 128-word FIFO within the IC. The FIFO allows the MAX86140/MAX86141 to be connected to a microcontroller or processor on a shared bus, where the data is not being read continuously from the MAX86140/MAX86141's registers. Both operate in fully autonomous modes for low power battery applications.

The MAX86140 consists of a single optical readout channel, while the MAX86141 incorporates dual optical readout channels that operate simultaneously. Both parts have three LED drivers and are well suited for a wide variety of optical sensing applications.

The MAX86140/MAX86141 operate on a 1.8V main supply voltage, with a separate 3.1V to 5.5V LED driver power supply. Both devices have flexible timing and shut-down configurations as well as control of individual blocks so an optimized measurement can be made at minimum power levels.

Optical Subsystem

The optical subsystem in the MAX86140/MAX86141 is composed of ambient light cancellation (ALC), a continuous-time sigma-delta ADC, and proprietary discrete time filter. ALC incorporates a proprietary scheme to cancel ambient light generated photodiode current, allowing the sensor to work in high ambient light conditions. The optical ADC has programmable full-scale ranges of 4 μ A to 32 μ A. The internal ADC is a continuous time oversampling sigma delta converter with 19-bit resolution.

The ADC output data rate can be programmed from 8sps (samples per second) to 4096sps per channel. The MAX86140/MAX86141 includes a proprietary discrete time filter to reject 50Hz/60Hz interference and changing residual ambient light from the sensor measurements.

The MAX86140/MAX86141 supports Dynamic Power Down mode (Low-Power mode) in which the power consumption is decreased between samples. This mode is only supported for sample rates 128sps and below. For more details on the power consumption at each sample rates, refer to the [Electrical Characteristics](#) table.

LED Driver

The MAX86140/MAX86141 integrates three precision LED driver-current DACs that modulate LED pulses for a variety of optical measurements. The LED current DACs have 8-bits of dynamic range with four programmable full-scale ranges of 31mA, 62mA, 94mA, and 124mA. The LED drivers are low dropout current sources, allowing for low-noise, power-supply independent LED currents to be sourced at the lowest supply voltage possible; therefore minimizing LED power consumption. The LED pulse width can be programmed from 14.8 μ s to 117.3 μ s to allow the algorithms to optimize SpO₂ and HR accuracy at the lowest dynamic power consumption dictated by the application.

FIFO Configuration

The FIFO is 128 sample depth and is designed to support various data types, as shown in [Table 2](#). Each sample width is 3 bytes, which includes a 5-bit tag width. The tag embedded in the FIFO_DATA is used to identify the source of each sample data. The description of each Tag is as shown in [Table 3](#).

LED Sequence Control (address 0x20 ~ 0x22)

The data format in the FIFO, as well as the sequencing of exposures, are controlled by the LED Sequence Registers through LEDC1 through LEDC6. There are six LED Sequence Data Items available, as shown in [Table 1](#). The exposure sequence cycles through the LED Sequence bit fields, starting from LEDC1 to LEDC6. The first LED Sequence field set to NONE (0000) ends the sequence.

Table 1. LED Sequence Control Registers

| ADDRESS | REG- ISTER NAME | DE- FAULT VALUE | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |
|---------|-------------------------------|-----------------------|------------|----|----|----|------------|----|----|----|
| 0x20 | LED Sequence Register 1 | 00 | LEDC2[3:0] | | | | LEDC1[3:0] | | | |
| 0x21 | LED Sequence Register 2 | 00 | LEDC4[3:0] | | | | LEDC3[3:0] | | | |
| 0x22 | LED Sequence Register 3 | 00 | LEDC6[3:0] | | | | LEDC5[3:0] | | | |

[Table 2](#) lists the codes for exposures selected in the LED sequence control registers.

Table 2. LED Sequence Register Data Type

| LEDCn[3:0] | DATA TYPE |
|------------|--|
| 0000 | NONE |
| 0001 | LED1 |
| 0010 | LED2 |
| 0011 | LED3 |
| 0100 | LED1 and LED2 pulsed simultaneously |
| 0101 | LED1 and LED3 pulsed simultaneously |
| 0110 | LED2 and LED3 pulsed simultaneously |
| 0111 | LED1, LED2, and LED3 pulsed simultaneously |
| 1000 | Pilot on LED1 |
| 1001 | DIRECT AMBIENT |
| 1010 | LED4 (external mux control) |
| 1011 | LED5 (external mux control) |
| 1100 | LED6 (external mux control) |
| 1101 | Reserved |
| 1110 | Reserved |
| 1111 | Reserved |

Table 3 shows the format of the FIFO data along with the associated Tag. In a sample if a picket fence event is detected, the predicted value is pushed to the FIFO along with its tag (PPFx_LEDcX_DATA).

Table 3. FIFO Data and Tag

| TAG[4:0] | DATA TYPE | COMMENTS |
|----------|-----------------|--|
| 00001 | PPG1 LEDC1 DATA | If LEDC1 is non-zero |
| 00010 | PPG1 LEDC2 DATA | If LEDC1 and LEDC2 are non-zero |
| 00011 | PPG1 LEDC3 DATA | If LEDC1, LEDC2 and LEDC3 are non-zero |
| 00100 | PPG1 LEDC4 DATA | If LEDC1, LEDC2, LEDC3, and LEDC4 are non-zero |
| 00101 | PPG1 LEDC5 DATA | If LEDC1, LEDC2, LEDC3, LEDC4, and LEDC5 are non-zero |
| 00110 | PPG1 LEDC6 DATA | If LEDC1, LEDC2, LEDC3, LEDC4, LEDC5, and LEDC6 are non-zero |
| 00111 | PPG2 LEDC1 DATA | If LEDC1 is non-zero |
| 01000 | PPG2 LEDC2 DATA | If LEDC1 and LEDC2 are non-zero |
| 01001 | PPG2 LEDC3 DATA | If LEDC1, LEDC2, and LEDC3 are non-zero |
| 01010 | PPG2 LEDC4 DATA | If LEDC1, LEDC2, LEDC3, and LEDC4 are non-zero |
| 01011 | PPG2 LEDC5 DATA | If LEDC1, LEDC2, LEDC3, LEDC4, and LEDC5 are non-zero |
| 01100 | PPG2 LEDC6 DATA | If LEDC1, LEDC2, LEDC3, LEDC4, LEDC5, and LEDC6 are non-zero |
| 01101 | PPF1 LEDC1 DATA | If LEDC1 is non-zero (Picket Fence Event) |
| 01110 | PPF1 LEDC2 DATA | If LEDC1 and LEDC2 are non-zero (Picket Fence Event) |
| 01111 | PPF1 LEDC3 DATA | If LEDC1, LEDC2, and LEDC3 are non-zero (Picket Fence Event) |
| 10000 | Reserved | |
| 10001 | Reserved | |
| 10010 | Reserved | |
| 10011 | PPF2 LEDC1 DATA | If LEDC1 is non-zero (Picket Fence Event) |
| 10100 | PPF2 LEDC2 DATA | If LEDC1 and LEDC2 are non-zero (Picket Fence Event) |
| 10101 | PPF2 LEDC3 DATA | If LEDC1, LEDC2, and LEDC3 are non-zero (Picket Fence Event) |
| 10110 | Reserved | |
| 10111 | Reserved | |
| 11000 | Reserved | |
| 11001 | PROX1 DATA | Only PILOT LED1 for LEDC1 is used |
| 11010 | PROX2 DATA | Only PILOT LED1 for LEDC1 is used |
| 11011 | Reserved | |
| 11100 | Reserved | |
| 11101 | Reserved | |
| 11110 | INVALID DATA | This tag indicates that there was an attempt to read an empty FIFO |
| 11111 | TIME STAMP | If TIME_STAMP_EN = 1, this is TIME_STAMP |

There are seven registers that control how the FIFO is configured and read out. These registers are illustrated below.

Table 4. PPG Configuration

| AD-DRESS | REGISTER NAME | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |
|----------|----------------------|----------------------|------------------|---------------|------------|------------------|-------------|---------|----|
| 0x04 | FIFO Write Pointer | – | | | | FIFO_WR_PTR[6:0] | | | |
| 0x05 | FIFO Read Pointer | – | | | | FIFO_RD_PTR[6:0] | | | |
| 0x06 | Overflow Counter | – | | | | OVF_COUNTER[6:0] | | | |
| 0x07 | FIFO Data Counter | FIFO_DATA_COUNT[7:0] | | | | | | | |
| 0x08 | FIFO Data Register | FIFO_DATA[7:0] | | | | | | | |
| 0x09 | FIFO Configuration 1 | – | FIFO_A_FULL[6:0] | | | | | | |
| 0x0A | FIFO Configuration 2 | – | – | TIME_STAMP_EN | FLUSH_FIFO | FIFO_STAT_CLR | A_FULL_TYPE | FIFO_RO | – |

Write Pointer (Register 0x04)

FIFO_WR_PTR[6:0] points to the FIFO location where the next item will be written. This pointer advances for each item pushed on to the FIFO by the internal conversion process. The write pointer is a 7-bit counter and will wrap around to count 0x00 on the next item after count 0x7F.

Read Pointer (Register 0x05)

FIFO_RD_PTR[6:0] points to the location from where the next item from the FIFO will be read via the serial interface. This advances each time an item is read from the FIFO. The read pointer can be both read and written to. This allows an item to be reread from the FIFO if it has not already been overwritten. The read pointer is updated from a 7-bit counter and will wrap around to count 0x00 from count 0x7F.

Overflow Counter (Register 0x06)

OVF_COUNTER[6:0] logs the number of items lost if the FIFO is not read in a timely fashion. This counter holds/saturates at count value 0x7F. When a complete item is popped from the FIFO (when the read pointer advances), the OVF_COUNTER is reset to zero. This counter is essentially a debug tool. It should be read immediately before reading the FIFO in order to check if an overflow condition has occurred.

FIFO Data Counter (Register 0x07)

FIFO_DATA_COUNT[7:0] is a read-only register which holds the number of items available in the FIFO for the host to read. This increments when a new item is pushed to the FIFO, and decrements when the host reads an item from the FIFO.

FIFO Data (Register 0x08)

FIFO_DATA[7:0] is a read-only register used to retrieve data from the FIFO. It is important to burst read the item from the FIFO. Each item is three bytes. So burst reading three bytes at FIFO_DATA register via the serial interface advances the FIFO_RD_PTR. The format and data type of the data stored in the FIFO is determined by the Tag associated with data. Readout from the FIFO follows a progression defined by LED Sequence Control registers as well. This configuration is best illustrated by a few examples.

Assume it is desired to perform a SpO₂ measurement and also monitor the ambient level on the photodiode to adjust the IR and red LED intensity. To perform this measurement, configure the following registers:

| | | |
|-----------------------------|--|---------------------------|
| LED Sequence Control | | |
| LEDC1 = 0x1 | | (LED1 exposure) |
| LEDC2 = 0x2 | | (LED2 exposure) |
| LEDC3 = 0x9 | | (DIRECT AMBIENT exposure) |
| LEDC4 = 0x0 | | (NONE) |
| LEDC5 = 0x0 | | (NONE) |
| LEDC6 = 0x0 | | (NONE) |
| PPG Configuration | | |
| PPG1_ADC_RGE[1:0] | | (PPG1 Gain Range Control) |
| PPG2_ADC_RGE[1:0] | | (PPG2 Gain Range Control) |
| PPG_TINT[1:0] | | (LED Pulse-Width Control) |
| PPG_SR[3:0] | | (Sample Rate) |
| LED Pulse Amplitude | | |
| LED1_PA[7:0] | | (LED1 Drive Current) |
| LED2_PA[7:0] | | (LED2 Drive Current) |

When done so the sample sequence and the data format in the FIFO follows the following time/location sequence.

tag 1, PPG1 LED1 data
tag 7, PPG2 LED1 data
tag 2, PPG1 LED2 data
tag 8, PPG2 LED2 data
tag 3, PPG1 Ambient data
tag 9, PPG2 Ambient data
tag 1, PPG1 LED1 data
tag 7, PPG2 LED1 data
tag 2, PPG1 LED2 data
tag 8, PPG2 LED2 data
tag 3, PPG1 Ambient data
tag 9, PPG2 Ambient data

.
.
.

tag 1, PPG1 LED1 data
tag 7, PPG2 LED1 data
tag 2, PPG1 LED2 data
tag 8, PPG2 LED2 data
tag 3, PPG1 Ambient data
tag 9, PPG2 Ambient data

where:

PPGm LED1 data = the ambient corrected exposure data from LED1 in PPGm channel
PPGm LED2 data = the ambient corrected exposure data from LED2 in PPGm channel
PPGm Ambient data = the direct ambient sample in PPGm channel
m = 1 of PPG1 channel, and 2 for PPG2 channel

For a second example, assume it is desired to pulse LED1 and LED2 simultaneously while also monitoring the ambient level.

| LED Sequence Control | | |
|----------------------|--|---------------------------|
| LEDC1 = 0x4 | | (LED1 and LED2 exposure) |
| LEDC2 = 0x9 | | (DIRECT AMBIENT exposure) |
| LEDC3 = 0x0 | | (NONE) |
| LEDC4 = 0x0 | | (NONE) |
| LEDC5 = 0x0 | | (NONE) |
| LEDC6 = 0x0 | | (NONE) |

In this case, the sequencing in the FIFO will then be:

```

tag 1, PPG1 LED1+LED2 data
tag 7, PPG2 LED1+LED2 data
tag 2, PPG 1 Ambient data
tag 8, PPG 2 Ambient data
tag 1, PPG1 LED1+LED2 data
tag 7, PPG2 LED1+LED2 data
tag 2, PPG1 Ambient data
tag 8, PPG2 Ambient data

```

```

.
.
.

```

```

tag 1, PPG1 LED1+LED2 data
tag 7, PPG2 LED1+LED2 data
tag 2, PPG1 Ambient data
tag 8, PPG2 Ambient data

```

where:

PPGm LED1+LED2 data = the ambient corrected exposure data from LED1 and LED2 for PPGm channel

PPGm Ambient data = the direct ambient corrected sample for PPGm channel

The number of bytes of data for the PPG channel is given by: $2 \times 3 \times K \times N$

where:

K = the number of active exposures as defined in the LED Sequence Control registers 0x20, 0x21, and 0x22.

N = the number of samples in the FIFO

To calculate the number of available items one can perform the following pseudo-code:

```

read the OVF_COUNTER register
read the FIFO_DATA_COUNT register
if OVF_COUNTER == 0 //no overflow occurred
    NUM_AVAILABLE_SAMPLES = FIFO_DATA_COUNT
else
    NUM_AVAILABLE_SAMPLES = 128 // overflow occurred and data has been lost
endif

```

[Table 6](#) shows the FIFO data format depends on the data type being stored. Optical data, whether full ambient corrected LED exposure, ambient corrected proximity or direct ambient sampled data is left-justified, as shown in [Table 6](#). Bits F23:F19 of the FIFO word contains the tag that identifies the data.

Table 6. Optical FIFO Data Format

| FIFO DATA FORMAT (FIFO_DATA[23:14]) | | | | | | | | | |
|-------------------------------------|-----|-----|-----|-----|------------------------------|-----|-----|-----|-----|
| Tag (TAG[4:0]) | | | | | ADC Value (FIFO_DATA[18:14]) | | | | |
| F23 | F22 | F21 | F20 | F19 | F18 | F17 | F16 | F15 | F14 |
| T4 | T3 | T2 | T1 | T0 | O18 | O17 | O16 | O15 | O14 |

| FIFO DATA FORMAT (FIFO_DATA[13:4]) | | | | | | | | | |
|------------------------------------|-----|-----|-----|----|----|----|----|----|----|
| ADC Value (FIFO_DATA[13:4]) | | | | | | | | | |
| F13 | F12 | F11 | F10 | F9 | F8 | F7 | F6 | F5 | F4 |
| O13 | O12 | O11 | O10 | O9 | O8 | O7 | O6 | O5 | O4 |

| FIFO DATA FORMAT (FIFO_DATA[3:0]) | | | | | | | | | |
|-----------------------------------|--|----|--|----|--|----|--|--|--|
| ADC Value (FIFO_DATA[3:0]) | | | | | | | | | |
| F3 | | F2 | | F1 | | F0 | | | |
| O3 | | O2 | | O1 | | O0 | | | |

FIFO_A_FULL (address 0x09)

The FIFO_A_FULL[6:0] field in the FIFO Configuration 1 register (0x09) sets the watermark for the FIFO and determines when the A_FULL bit in the Interrupt_Status register (0x00) gets asserted. The A_FULL bit will be set when the FIFO contains 128 minus FIFO_A_FULL[6:0] items. When the FIFO is almost full, if the A_FULL_EN mask bit in the Interrupt_Enable register (0x03) is set, then A_FULL bit gets asserted in the Interrupt Status 1 register and this bit is routed to the $\overline{\text{INT}}$ pin on the serial interface. This condition should prompt the applications processor to read samples off of the FIFO before it fills. The A_FULL bit is cleared when the status register is read.

The application processor can read both the FIFO_WR_PTR and FIFO_RD_PTR to calculate the number of items available in the FIFO, or just read the OVF_COUNTER and FIFO_DATA_COUNT registers, and read as many items as it needs to empty the FIFO. Alternatively, if the applications always responds much faster than the selected sample rate, it could just read 128 minus FIFO_A_FULL[6:0] items when it gets A_FULL interrupt and be assured that all data from the FIFO is read.

FIFO_RO (Address 0x0A)

The FIFO_RO bit in the FIFO Configuration 2 register (0x0A) determines whether samples get pushed on to the FIFO when it is full. If push is enabled when FIFO is full, old samples are lost. If FIFO_RO is not set, the new sample is dropped and the FIFO is not updated.

A_FULL_TYPE (Address 0x0A)

The A_FULL_TYPE bit defines the behavior of the A_FULL interrupt. If the A_FULL_TYPE bit is set low, the A_FULL

interrupt gets asserted when the A_FULL condition is detected and cleared by status register read, but reasserts for every sample if the A_FULL condition persists. If A_FULL_TYPE bit is set high, the A_FULL interrupt gets asserted only when a new A_FULL condition is detected. The interrupt gets cleared on Interrupt Status 1 register read, and does not reassert for every sample until a new A_FULL condition is detected.

FIFO_STAT_CLR (Address 0x0A)

The FIFO_STAT_CLR bit defines whether the A_FULL interrupt should get cleared by FIFO_DATA register read. If FIFO_STAT_CLR is set low, A_FULL and DATA_RDY interrupts do not get cleared by FIFO_DATA register read but get cleared by status register read. If FIFO_STAT_CLR is set high, A_FULL and DATA_RDY interrupts get cleared by a FIFO_DATA register read or a status register read.

FLUSH_FIFO (Address 0x0A)

The FIFO Flush bit is used for flushing the FIFO. The FIFO becomes empty and the FIFO_WR_PTR[6:0], FIFO_RD_PTR[6:0], FIFO_DATA_COUNT[7:0] and OVF_COUNTER[6:0] get reset to zero. FLUSH_FIFO is a self-clearing bit.

TIME_STAMP_EN (Address 0x0A)

When TIME_STAMP_EN bit is set to 1, the 19 bits time stamp gets pushed to the FIFO along with its Tag for every 8 samples. This timestamp is useful for aligning data from two devices after the host reads the FIFOs of those devices. When TIME_STAMP_EN bit is set to 0, the sample counter is not pushed to FIFO.

Pseudo-Code Example of Initializing the Optical AFE

The following pseudo-code shows an example of configuring MAX86140/MAX86141 for a SpO2 applications, where LED1 and LED2 are IR and red LED, respectively.

```
DEVICE OPEN
START;
// AFE Initialization
WRITE 0x1 to RESET[0];           // Soft Reset (Register 0x0D[0])
DELAY 1ms;
READ Interrupt Status 1;         // Clear Interrupt (Register 0x00)
READ Interrupt Status 2;         // Clear Interrupt (Register 0x01)
WRITE 0x1 to SHDN[0];           // Shutdown (Register 0x0D[1])
WRITE 0x3 to PPG_TINT[1:0];      // Pulse Width = 123.8ms (Register 0x11[1:0])
WRITE 0x2 to PPG1_ADC_RGE1[0];   // ADC Range = 16µA (Register 0x11[3:2])
WRITE 0x2 to PPG2_ADC_RGE1[0];   // ADC Range = 16µA (Register 0x11[3:2])
// For MAX86141 when used in Dual Channel only
WRITE 0x0 to SMP_AVE[2:0];       // Sample Averaging = 1 (Register 0x12[2:0])
WRITE 0x00 to PPG_SR[4:0];       // Sample Rate = 25sps (Register 0x12[7:3])
WRITE 0x3 to LED_SETLNG[1:0];    // LED Settling Time = 12ms (Register 0x13[7:6])
WRITE 0x01 to PDBIAS1[2:0];      // PD 1 Biasing for Cpd = 0~65pF (Register 0x15[2:0])
WRITE 0x01 to PDBIAS2[2:0];      // PD 1 Biasing for Cpd = 0~65pF (Register 0x15[2:0])
// For MAX86141 when used in Dual Channel only
WRITE 0x3 to LED1_RGE[1:0];      // LED Driver 1 Range = 124mA (Register 0x15[2:0])
WRITE 0x3 to LED2_RGE[1:0];      // LED Driver 2 Range = 124mA (Register 0x15[2:0])
WRITE 0x20 to LED1_DRV[1:0];     // LED 1 Drive Current = 15.36mA (Register 0x23[7:0])
WRITE 0x20 to LED2_DRV[1:0];     // LED 2 Drive Current = 15.36mA (Register 0x24[7:0])
WRITE 0x1 to LP_Mode[0];         // Low Power mode enabled
// FIFO Configuration
WRITE 0x10 to FIFO_A_FULL[6:0];  // FIFO INT triggered condition (Register 0x09[6:0])
WRITE 0x1 to FIFO_RO;            // FIFO Roll Over enabled (Register 0x0A[1])
WRITE 0x1 to A_FULL_EN;          // FIFO_A_FULL interrupt enabled (Register 0x02[7])
WRITE 0x1 to LEDC1[3:0];         // LED1 exposure configured in time slot 1
WRITE 0x2 to LEDC2[3:0];         // LED2 exposure configured in time slot 1
WRITE 0x0 to LEDC3[3:0];
WRITE 0x0 to LEDC4[3:0];
WRITE 0x0 to LEDC5[3:0];
WRITE 0x0 to LEDC6[3:0];
WRITE 0x0 to SHDN[0];           // Start Sampling STOP;
```

Pseudo-Code for Interrupt Handling with FIFO_A_FULL

The following pseudo-code shows an example on handling the Interrupt when using A_FULL Interrupt.

```
void irqHandler(void)
{
    uint8_t intStatus;
    //Read Status
    ReadReg(0x00, &intStatus);
    if ( intStatus & 0x80 ) { //A FULL RDY
        device_data_read(); //Data Read Routine
    }
}
```

Pseudo-Code Example of Reading Data from FIFO

Example pseudo-code for reading data from FIFO when using single photodiode channel and two LED channels.

```
#define FIFO_SAMPLES (128-0x10) // FIFO_A_FULL[6:0] = 0x10
void device_data_read(void) {
    int i;
    uint8_t sampleCnt;
    uint8_t dataBuf[FIFO_SAMPLES*3]; //(128 - FIFO_A_FULL[6:0]) samples, 3 byte/channel
    uint8_t tag1[FIFO_SAMPLES/2]; //(128 - FIFO_A_FULL[6:0])/2channels samples
    uint8_t tag2[FIFO_SAMPLES/2]; //(128 - FIFO_A_FULL[6:0])/2channels samples
    int led1[FIFO_SAMPLES/2]; //(128 - FIFO_A_FULL[6:0])/2channels samples
    int led2[FIFO_SAMPLES/2]; //(128 - FIFO_A_FULL[6:0])/2channels samples

    ReadReg(0x07, &sampleCnt); // sampleCnt should be the same value as FIFO_SAMPLES
    //Read FIFO
    ReadFifo(dataBuf, sampleCnt * 3);

    for ( i = 0; i < sampleCnt/2/*channels*/; i++ ) {
        tag1[i] = (dataBuf[i*6+0] >> 3) & 0x1f;
        led1[i] = ((dataBuf[i*6+0] << 16) | (dataBuf[i*6+1] << 8) | (dataBuf[i*6+2]))
            & 0x7ffff;
        tag2[i] = (dataBuf[i*6+3] >> 3) & 0x1f;
        led2[i] = ((dataBuf[i*6+3] << 16) | (dataBuf[i*6+4] << 8) | (dataBuf[i*6+5]))
            & 0x7ffff;
    }
}
```

Example pseudo-code for reading data from FIFO when using dual photodiode channels and two LED channels.

```
#define FIFO_SAMPLES (128-0x10) // FIFO_A_FULL[6:0] = 0x10
void device_data_read(void) {
    int i;
    uint8_t sampleCnt;
    uint8_t dataBuf[FIFO_SAMPLES*3]; //(128 - FIFO_A_FULL[6:0]) samples, 3 byte/channel
    uint8_t tag1A[FIFO_SAMPLES/4]; //(128 - FIFO_A_FULL[6:0])/4channels samples
    uint8_t tag1B[FIFO_SAMPLES/4]; //(128 - FIFO_A_FULL[6:0])/4channels samples
    uint8_t tag2A [FIFO_SAMPLES/4]; //(128 - FIFO_A_FULL[6:0])/4channels samples
    uint8_t tag2B [FIFO_SAMPLES/4]; //(128 - FIFO_A_FULL[6:0])/4channels samples
    int led1A[FIFO_SAMPLES/4];
    int led1B[FIFO_SAMPLES/4];
    int led2A[FIFO_SAMPLES/4];
    int led2B[FIFO_SAMPLES/4];

    ReadReg(0x07, &sampleCnt); // sampleCnt should be the same value as FIFO_SAMPLES
    //Read FIFO
    ReadFifo(dataBuf, sampleCnt * 3);

    for ( i = 0; i < sampleCnt/4/*channels*/; i++ ) {
        tag1A[i] = (dataBuf[i*12+0] >> 3) & 0x1f;
        led1A[i] = ((dataBuf[i*12+0] << 16) | (dataBuf[i*12+1] << 8) |
            dataBuf[i*12+2])) & 0x7ffff; // LED1, PD1
        tag1B[i] = (dataBuf[i*12+3] >> 3) & 0x1f;
        led1B[i] = ((dataBuf[i*12+3] << 16) | (dataBuf[i*12+4] << 8) |
            (dataBuf[i*12+5])) & 0x7ffff; // LED1, PD2
        tag2A[i] = (dataBuf[i*12+6] >> 3) & 0x1f;
        led2A[i] = ((dataBuf[i*12+6] << 16) | (dataBuf[i*12+7] << 8) |
            (dataBuf[i*12+8])) & 0x7ffff; // LED2, PD1
        tag2B[i] = (dataBuf[i*12+9] >> 3) & 0x1f;
        led2B[i] = ((dataBuf[i*12+9] << 16) | (dataBuf[i*12+10] << 8) |
            (dataBuf[i*12+11])) & 0x7ffff; // LED2, PD2
    }
}
```

Optical Timing

The MAX86140/MAX86141 optical controller is capable of being configured to make a variety of measurements. Each LED exposure is ambient light compensated before the ADC conversion.

The controller can be configured to pulse one, two or three LED drivers sequentially so as to make measurements at multiple wavelengths as is done in a pulse oximetry measurements or simultaneously to drive multiple LEDs as is done with heart rate measurements on the wrist.

The controller is also configurable to measure direct ambient level for every exposure sample. The direct ambient measurement can be used to adjust the LED drive

level to compensate for increased noise levels when high interfering ambient signals are present.

The following optical timing diagrams illustrate several possible measurement configurations.

One LED Pulsing with No Direct Ambient Sampling

The optical timing diagram below represents just LED1 pulsing during the exposure time with no direct ambient sampling enabled. This timing mode would be used when heart rate is being measured with a single green LED. In this mode a single optical sampled value will appear successively in the FIFO.

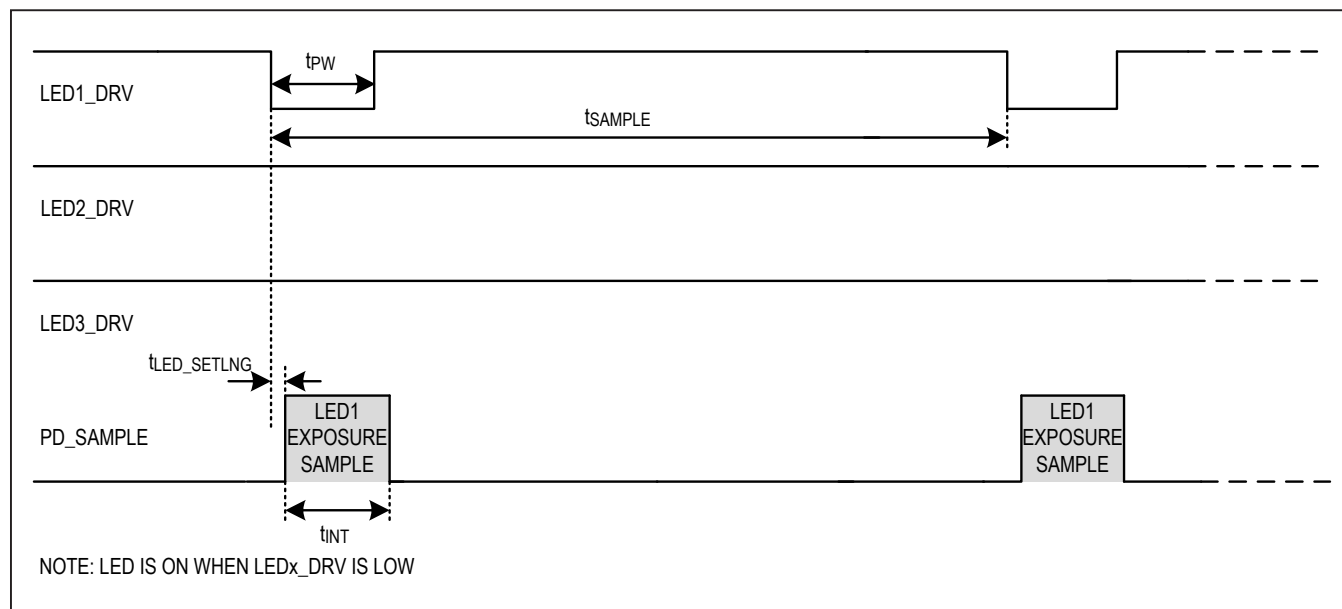


Figure 1. Timing for LED1 Pulsing with No Direct Ambient Sampling

One LED Pulsing with Direct Ambient Sampling

The optical timing diagram below represents just LED1 pulsing during the exposure time with direct ambient sampling enabled. This timing mode would be used when heart rate is being measured with a single, green LED. In this mode a single optical sampled value followed by the ambient sampled value will appear successively in the FIFO.

Two LEDs Pulse Simultaneously with Direct Ambient Sampling

The optical timing diagram below represents both LED1 and LED2 pulsing simultaneously with direct ambient sampling enabled. This timing mode would be used when heart rate is being measured with two green LEDs. In this mode a single optical sampled value followed by the ambient sampled value will appear in successive the FIFO locations. The direct ambient sampling is typically used to compensate the LED drive levels as the optical noise level can be elevated from ambient shot noise.

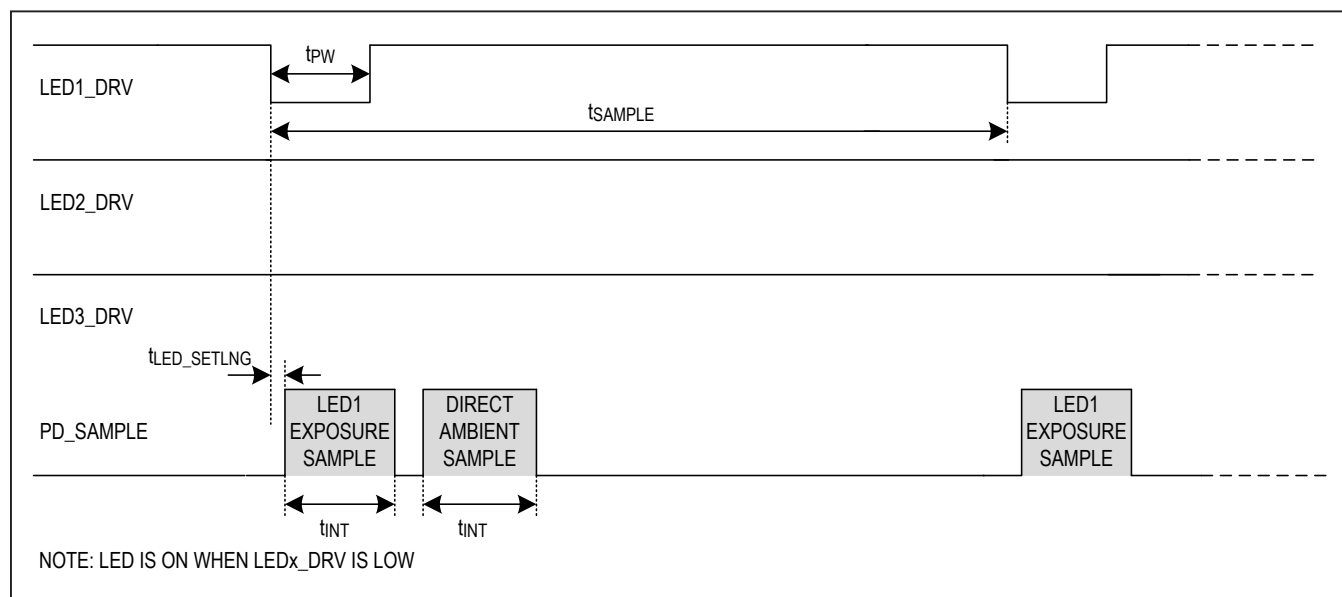


Figure 2. Timing for LED1 Pulsing with Direct Ambient Sampling

**All LED Pulsing Simultaneously with Direct
Ambient Sampling**

The optical timing diagram below represents all three LEDs pulsing simultaneously with direct ambient sampling enabled. This timing mode would be used when heart rate is being measured with three green LEDs. In this

mode, a single optical sampled value, followed by the ambient sampled value, will appear in successive the FIFO locations. The direct ambient sampling is typically used to compensate the LED drive levels as the optical noise level can be elevated from ambient shot noise.

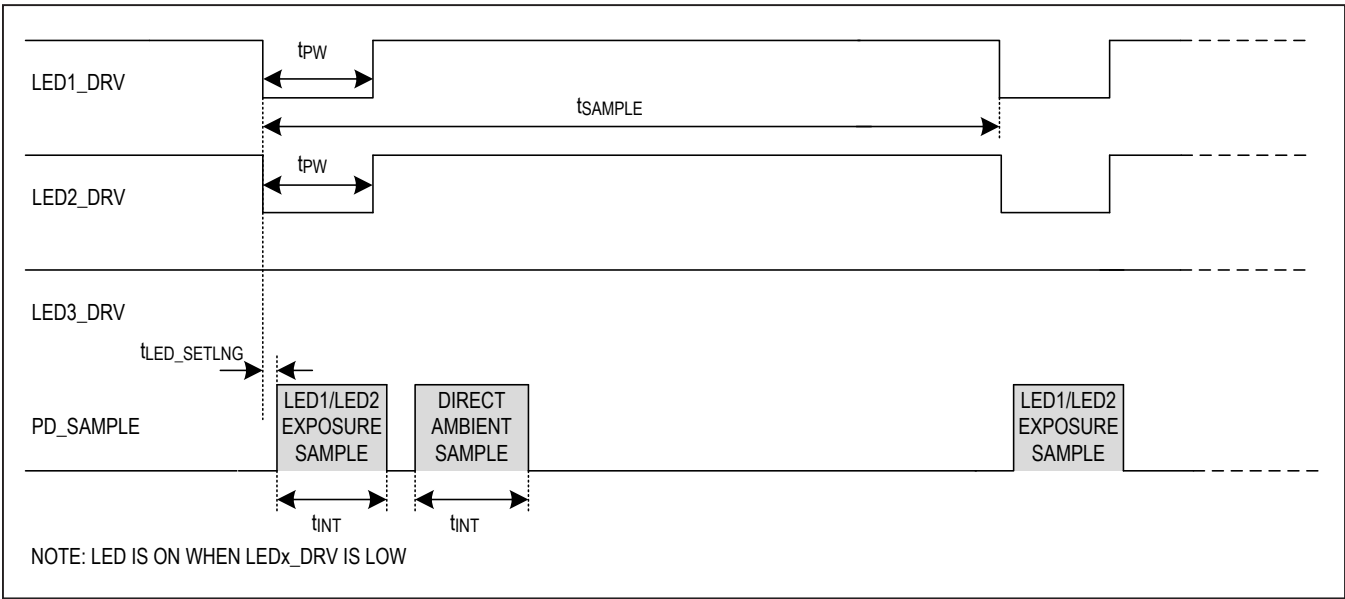


Figure 3. Timing for LED1 and LED2 Pulsing Simultaneously with Direct Ambient Sampling

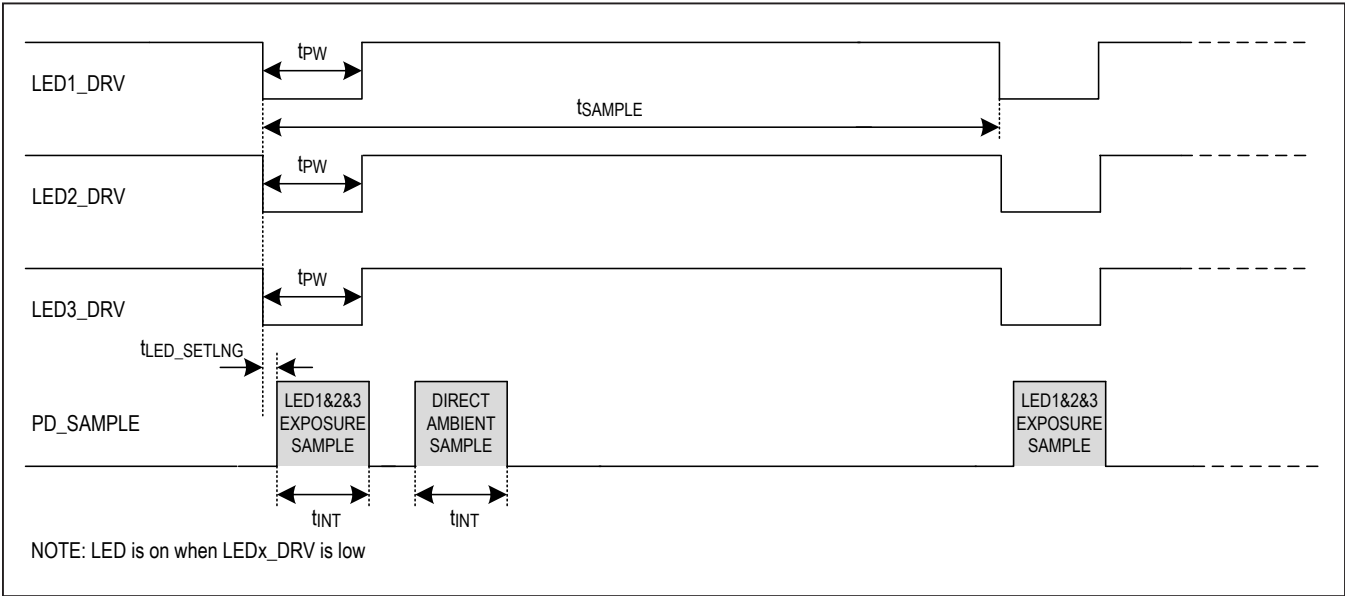


Figure 4. Timing for LED1, LED2, and LED3 Pulsing Simultaneously with Direct Ambient Sampling

Two LEDs Pulse Sequentially with Direct Ambient Sampling

The timing diagram below illustrates the optical timing when both LED1 and LED2 are enabled to pulse sequentially and direct ambient sampling is also enabled. This timing mode would be used when SpO₂ is being measured with IR and red LEDs. The optical sampled value for each LED will appear successively, followed by the direct ambient sampled value in the FIFO. when SpO₂ is being measured with IR and red LEDs. The optical sampled value

for each LED will appear successively, followed by the direct ambient sampled value in the FIFO.

All LEDs Pulse Sequential with Direct Ambient Sampling

The optical timing diagram below illustrates the three LEDs pulsing sequentially, followed by a direct ambient sample. This timing mode would be used when heart rate on a green LED is combined with and SpO₂ measurement using IR and red LEDs.

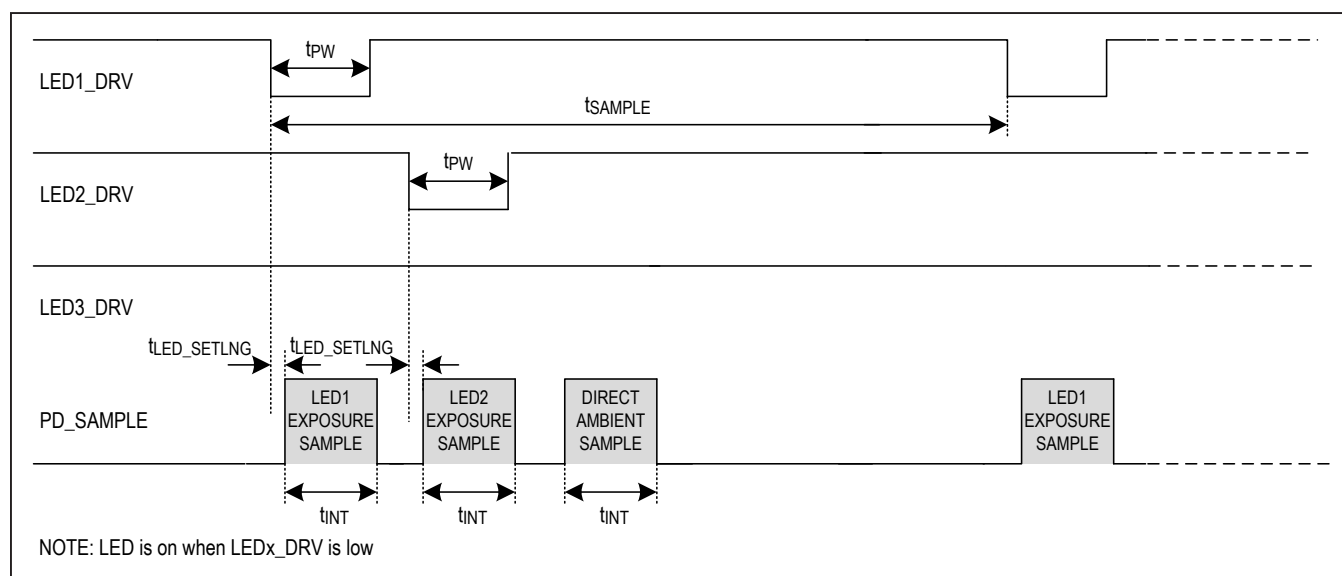


Figure 5. Timing for LED1 and LED2 Pulsing Sequentially with Direct Ambient Sampling

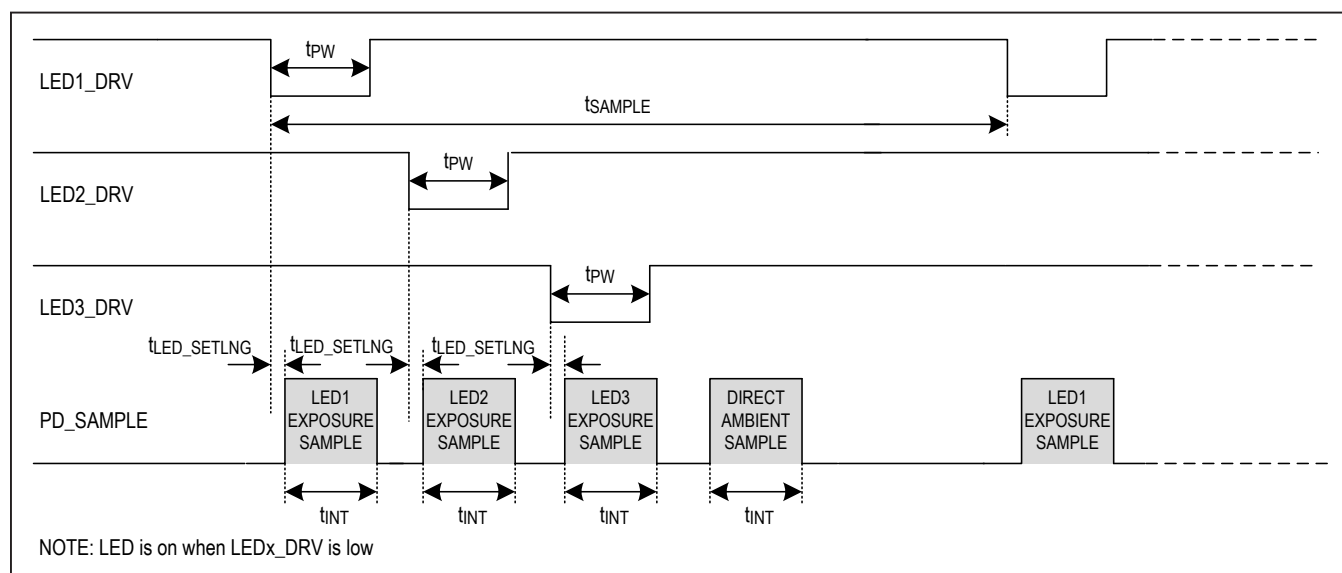


Figure 6. Timing for LED1, LED2, and LED3 Pulsing Sequentially with Direct Ambient Sampling

Note: Any I²C/SPI register write sequence will restart the finite state machine (FSM), which forces the start of a new measurement sequence (a new frame of measurements). Therefore a glitch in the frame rate can be expected between the frame measured just before and just after the I²C/SPI register write sequence.

To ensure that full frames are pushed to the FIFO, write to the registers only after a complete frame is ready (i.e., after A_FULL interrupt, where FIFO_A_FULL is in multiples of full frames) and the FIFO is read, otherwise, the FSM will restart before all measurements in the frame have been completed, thus leaving the FIFO with a partial frame. Additionally, when using the internal trigger and multiple registers are written, flush the FIFO after the last register write because if the writes are slow unwanted measurements may be pushed to the FIFO, and these will be cleared by the flush.

ADC Architecture and Transfer Function Non-Linearity (XNL) Trim

MAX86140/MAX86141 is comprised of a 16-bit current integrating incremental delta-sigma analog to digital converter (ADC), wrapped in a 5-bit sub-ranging digital to analog converter (DAC). The sub-ranging DAC is scaled to have two bits of redundancy, resulting in an overall dynamic range of 19 bits.

The native delta-sigma ADC linearity is exceptional. However, the sub-ranging DAC uses a unary architecture which has some mismatch between the unit current sources of the DAC and the ADC reference current. This mismatch results in some transfer function non-linearity (XNL) errors when the sub-DAC code transitions. For this reason, the sub-ranging DAC algorithm is designed to minimize DAC transitions by introducing large hysteresis through the overlapping sub-DAC ranges. Consequently, under normal PPG operation, the sub-DAC does not transition, and the linearity of the converter signal is driven entirely by the linear native delta-sigma ADC.

In addition to algorithmically reducing the sub-DAC transitions, the MAX86140/MAX86141 incorporates a self-calibration scheme that can be used to further reduce the

sub-DAC XNL errors. To run self-calibration, the following setup procedure should be used:

- 1) Write 0x00 to the following register addresses: 0x02, 0x03, 0x0D, 0x10, 0x12, 0x13, 0x20.
- 2) Set the PPG1_ADC_RGE, PPG2_ADC_RGE and PPG_TINT bit fields in the PPG_Configuration1 register, address 0x11, to the values required for the intended application.
- 3) Set the START_CAL bit to one on the DAC Calibration Enable register, address 0x50.
- 4) Wait for 200ms for the self-calibration procedure to complete.
- 5) Monitor the CAL_DAC_Complete bit in the DAC Calibration Enable register to go high, indicating the calibration procedure is complete.
- 6) Check the CAL_DAC1_OOR and CAL_DAC2_OOR bits in the DAC Calibration Enable register to verify that self-calibration has been completed successfully.
- 7) Configure the registers 0x02, 0x03, 0x0D, 0x10, 0x12, and 0x13 in any order for the application intended.
- 8) Finally, write register 0x20 to start the MAX86140/MAX86141 measurement sequence.

To further support dealing with the residual sub-DAC XNL error, which appears as a small offset shift when the sub-DAC transitions, an optional FIFO tag value can be enabled. This optional FIFO tag is enabled through the DAC_CODE_CHG_TAG bit in the PPG_Sync_Control register (0x10). When enabled, the FIFO will output a tag value of 0x1D on every conversion on which the sub-DAC transitions. This tag value will override the normal outputted tag for that conversion, thus allowing the conversion on which the sub-DAC update occurred to be located in the FIFO. One application of this feature would be to trigger special backend software handling for the conversions on which the sub-DAC update occurs to compensate for the residual error.

MAX86140/
MAX86141

Best-in-Class Optical Pulse Oximeter and
Heart-Rate Sensor for Wearable Health

GPIO Configuration

The MAX86140/MAX86141 support several means by which they can synchronize to external sensors, muxes, and be extended to allow for more flexibility in the measurement configuration. This functionality is extended through the GPIO1 and GPIO2 pins and is selected by the GPIO CTRL bit field in the PPG SYNC Control register (0x10). The following describes option and the functional state of GPIO1 and GPIO2 as well as the part behavior.

GPIO CTRL[3:0] 0000 and 0001: Stand Alone With and Without External Mux

Table 7. GPIO Mode 0000 and 0001

| GPIO CTRL | GPIO1 FUNCTION | GPIO2 FUNCTION | COMMENT |
|-----------|-------------------------|--------------------------------------|---|
| 0000 | Tristate or Mux Control | Disabled | GPIO1 will be active if any of the LEDCn[3:0] states A, B, or C are enabled in the exposure sequence. In this case, GPIO1 will be low during exposures on LED4, LED5 or LED6, otherwise it will be high. If LEDCn[3:0] state A, B, or C is not enabled in the exposure sequence, GPIO1 will be tristate unless externally pulled up. GPIO2 is disabled. Sample and exposure timing is controlled by the internal 32768Hz oscillator. |
| 0001 | Tristate or Mux Control | Input 32768Hz or 32000Hz Clock Input | GPIO1 will be active if any of the LEDCn[3:0] states A, B, or C are enabled in the exposure sequence. In this case, GPIO1 will be low during exposures on LED4, LED5 or LED6; otherwise, it will be high. If LEDCn[3:0] state A, B, or C is not enabled in the exposure sequence, GPIO1 will be tristate unless externally pulled up. GPIO2 is an input 32768/32000Hz. Sample and exposure timing is controlled by GPIO2 clock input. |

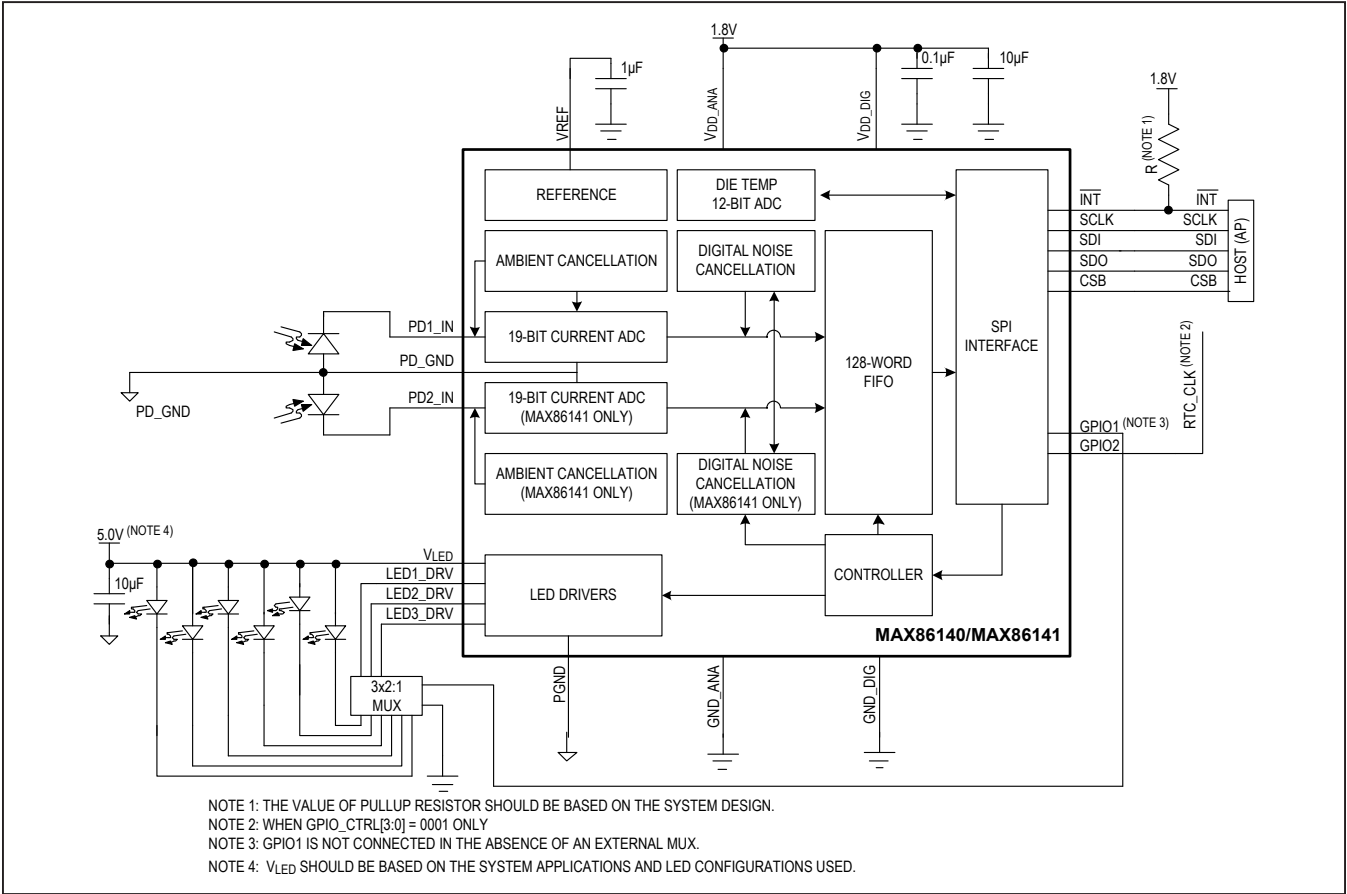


Figure 7. Block Diagram for GPIO CTRL[3:0] 0000 and 0001 with External Mux

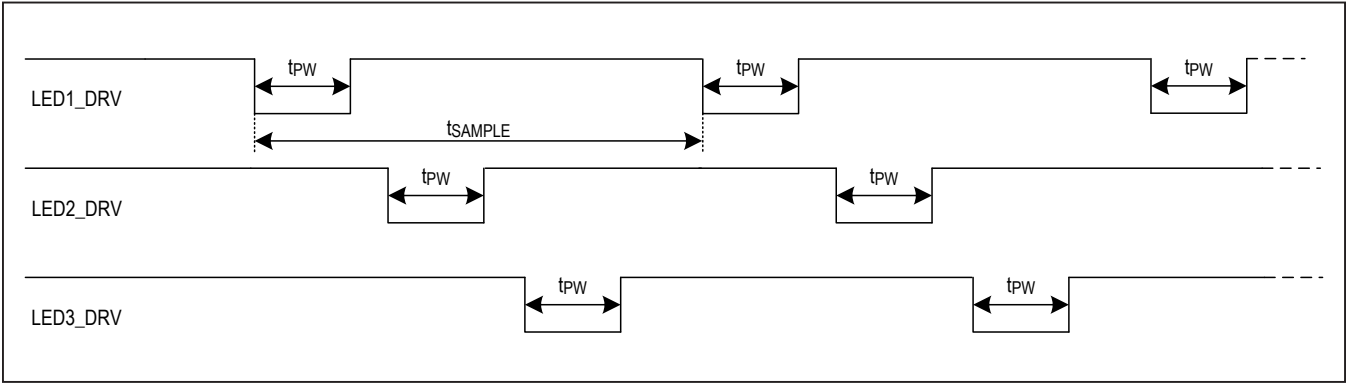


Figure 8. Timing Diagram for GPIO CTRL[3:0] 0000 and 0001 Without External Mux

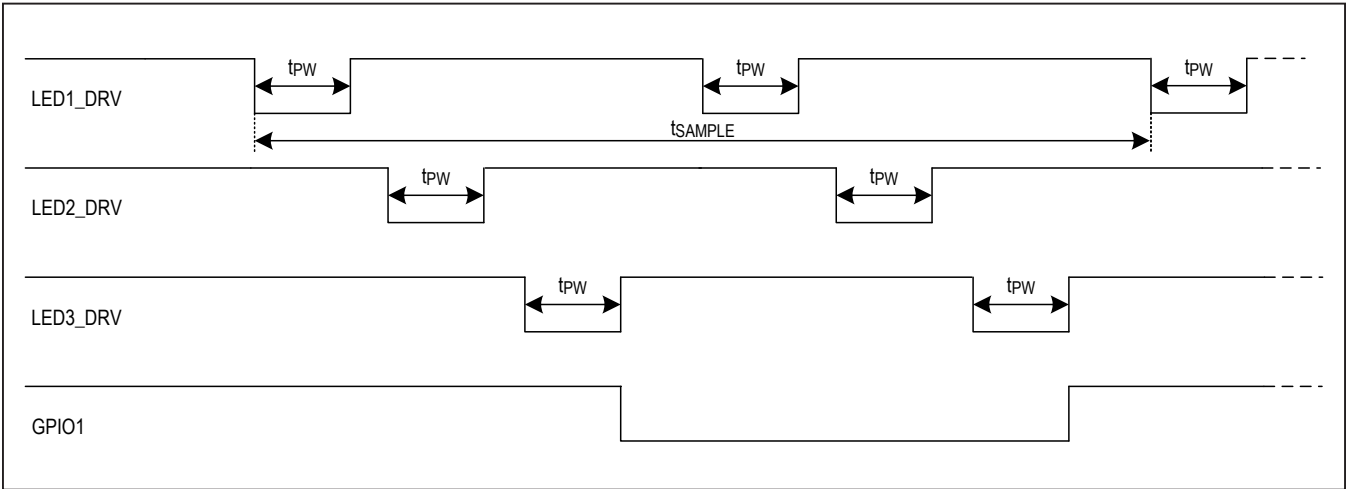


Figure 9. Timing Diagram for GPIO CTRL[3:0] 0000 and 0001 with External Mux

MAX86140/
MAX86141

Best-in-Class Optical Pulse Oximeter and
Heart-Rate Sensor for Wearable Health

GPIO CTRL[3:0] 0010: Start of Sample Input with and without External Mux

Table 8. GPIO Mode 0010

| GPIO CTRL | GPIO1 FUNCTION | GPIO2 FUNCTION | COMMENT |
|-----------|-------------------------|----------------------------|---|
| 0010 | Input Sample Trigger | Tristate or Mux Control | GPIO1 is defined as a sample trigger input (Target). This input can come from an external source or from another MAX86140/ MAX86141 in controller sample mode. GPIO2 will be active if any of the LEDCn[3:0] states A, B, or C are enabled in the exposure sequence. In this case, GPIO2 will be low during exposures on LED4, LED5 or LED6, otherwise it will be high. If LEDCn[3:0] state A, B, or C is not enabled in the exposure sequence, GPIO2 will be tristate unless externally pulled up. Exposure timing is controlled by internal oscillator. |

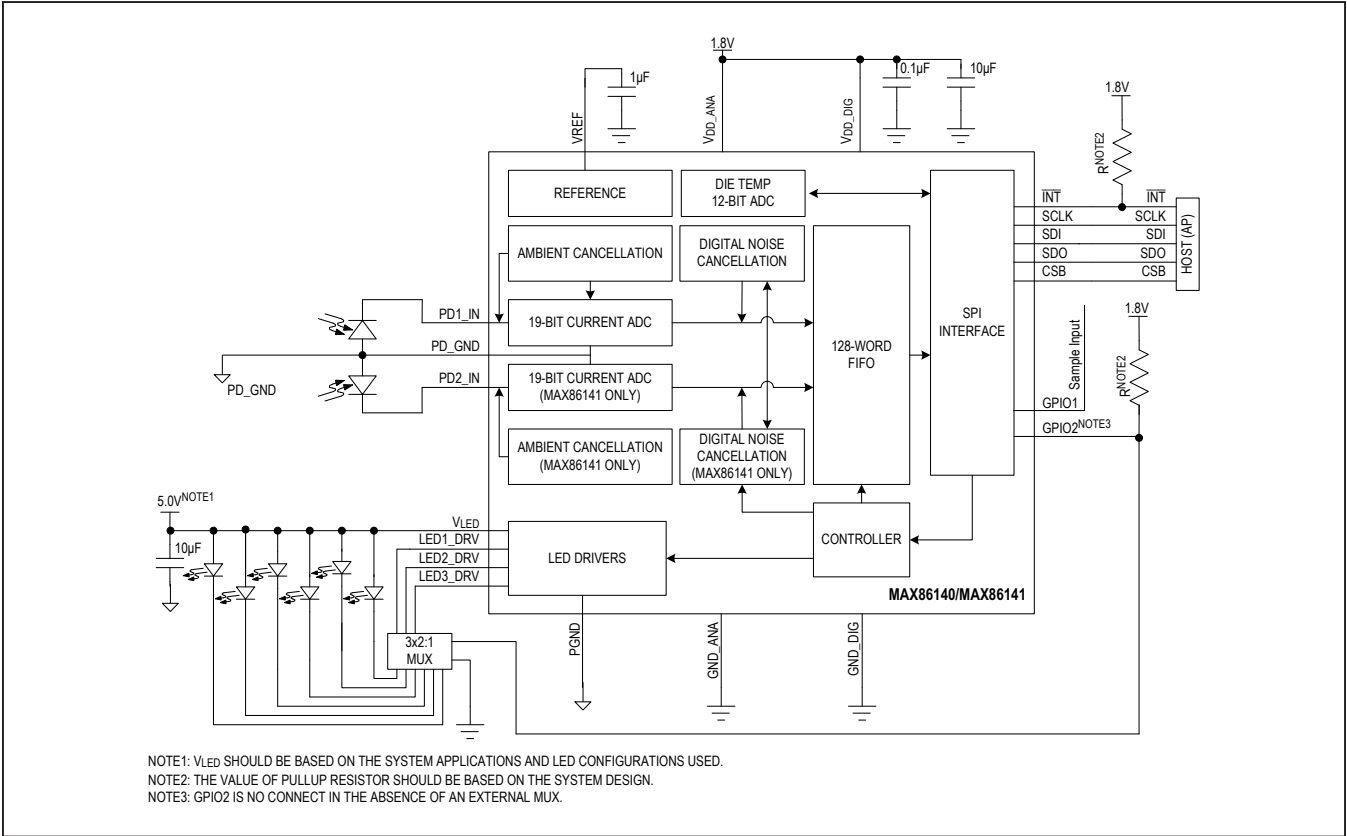


Figure 10. Block Diagram for GPIO CTRL[3:0] 0010 with External Mux

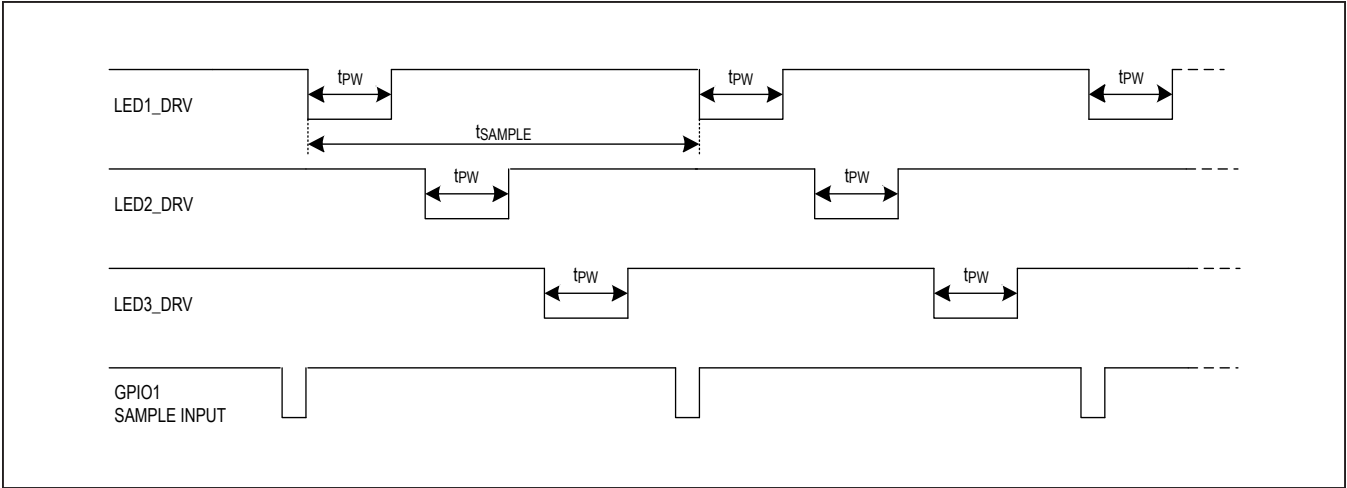


Figure 11. Timing Diagram for GPIO CTRL[3:0] 0010 Without External Mux

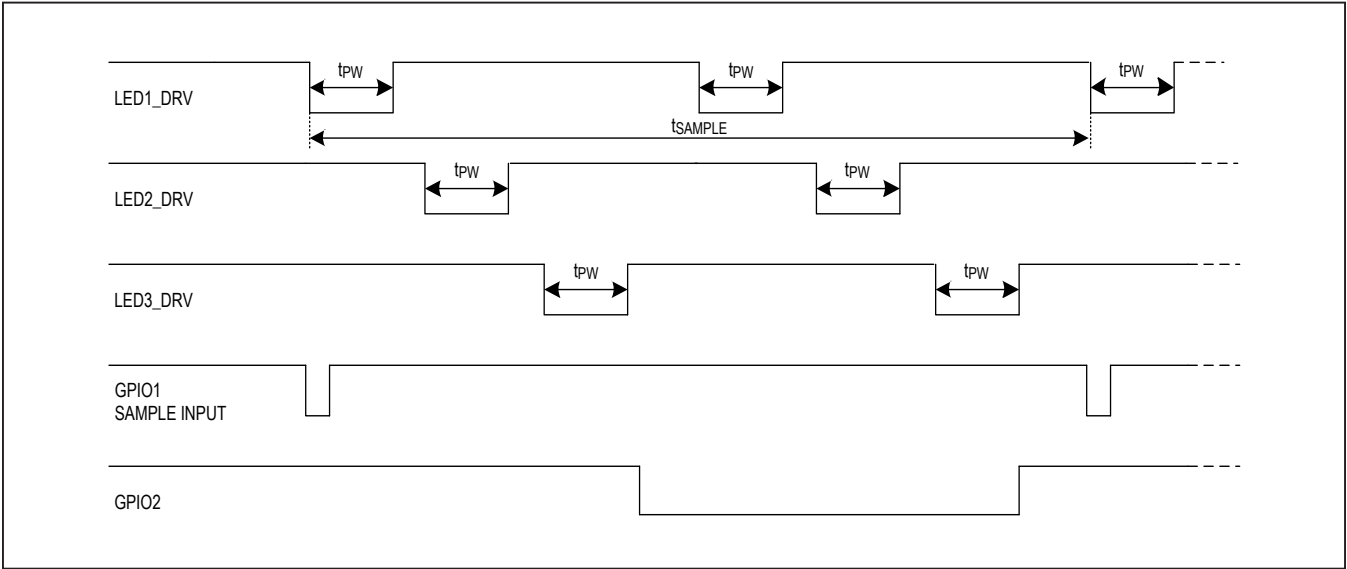


Figure 12. Timing Diagram for GPIO CTRL[3:0] 0010 with External Mux

MAX86140/
MAX86141

Best-in-Class Optical Pulse Oximeter and
Heart-Rate Sensor for Wearable Health

GPIO CTRL[3:0] 0011: Start of Sample Input with External Clock

Table 9. GPIO Mode 0011

| GPIO CTRL | GPIO1 FUNCTION | GPIO2 FUNCTION | COMMENT |
|-----------|-------------------------|--|---|
| 0011 | Input Sample Trigger | Input 32768Hz or 32000Hz Clock Input | GPIO1 is defined as a sample trigger input (Target). This input can come from an external source or from another MAX86140/ MAX86141 in controller sample mode. GPIO2 is an input 32768/32000Hz clock input. Exposure timing is controlled by GPIO2 clock input. |

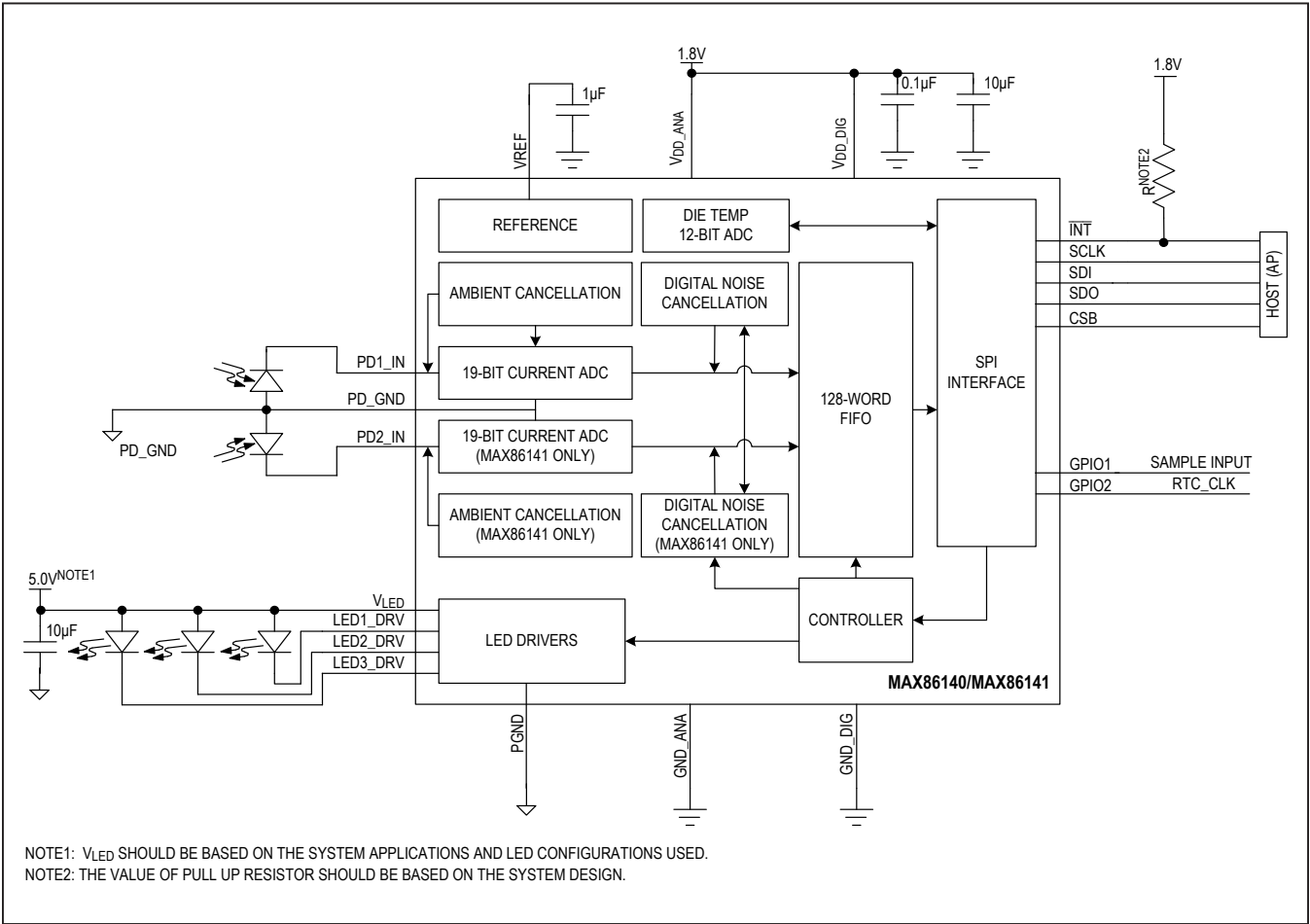


Figure 13. Block Diagram for GPIO CTRL[3:0] 0011

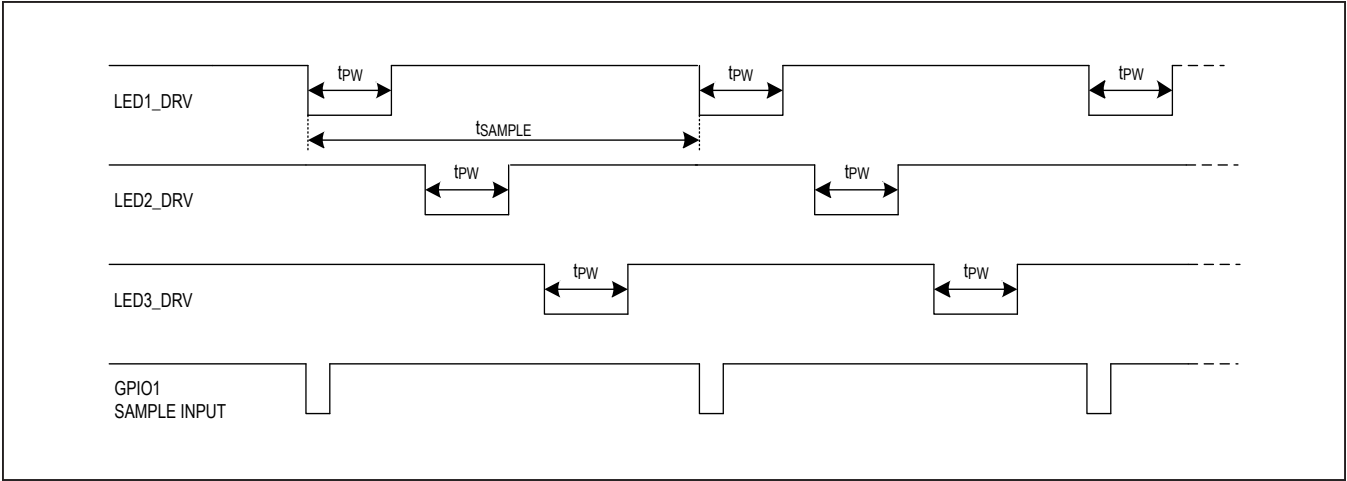


Figure 14. Timing Diagram for GPIO CTRL[3:0] 0011

GPIO CTRL[3:0] 0100: Start of Sample Output With and Without External Mux

Table 10. GPIO Mode 0100

| GPIO CTRL | GPIO1 FUNCTION | GPIO2 FUNCTION | COMMENT |
|-----------|---|----------------------------|---|
| 0100 | Active Output Controller Sample Output | Tristate or Mux Control | GPIO1 is defined as a controller sample output. The GPIO1 output can be used to trigger a second sensor. When used with a second MAX86140/MAX86141 set to target sample mode, the controller sample timing will drive target sample time. GPIO2 will be active if any of the LEDCn[3:0] states A, B, or C are enabled in the exposure sequence. In this case, GPIO2 will be low during exposures on LED4, LED5 or LED6, otherwise it will be high. If LEDCn[3:0] state A, B, or C is not enabled in the exposure sequence, GPIO2 will be tristate unless externally pulled up. Sample and exposure timing is controlled by internal oscillator. |

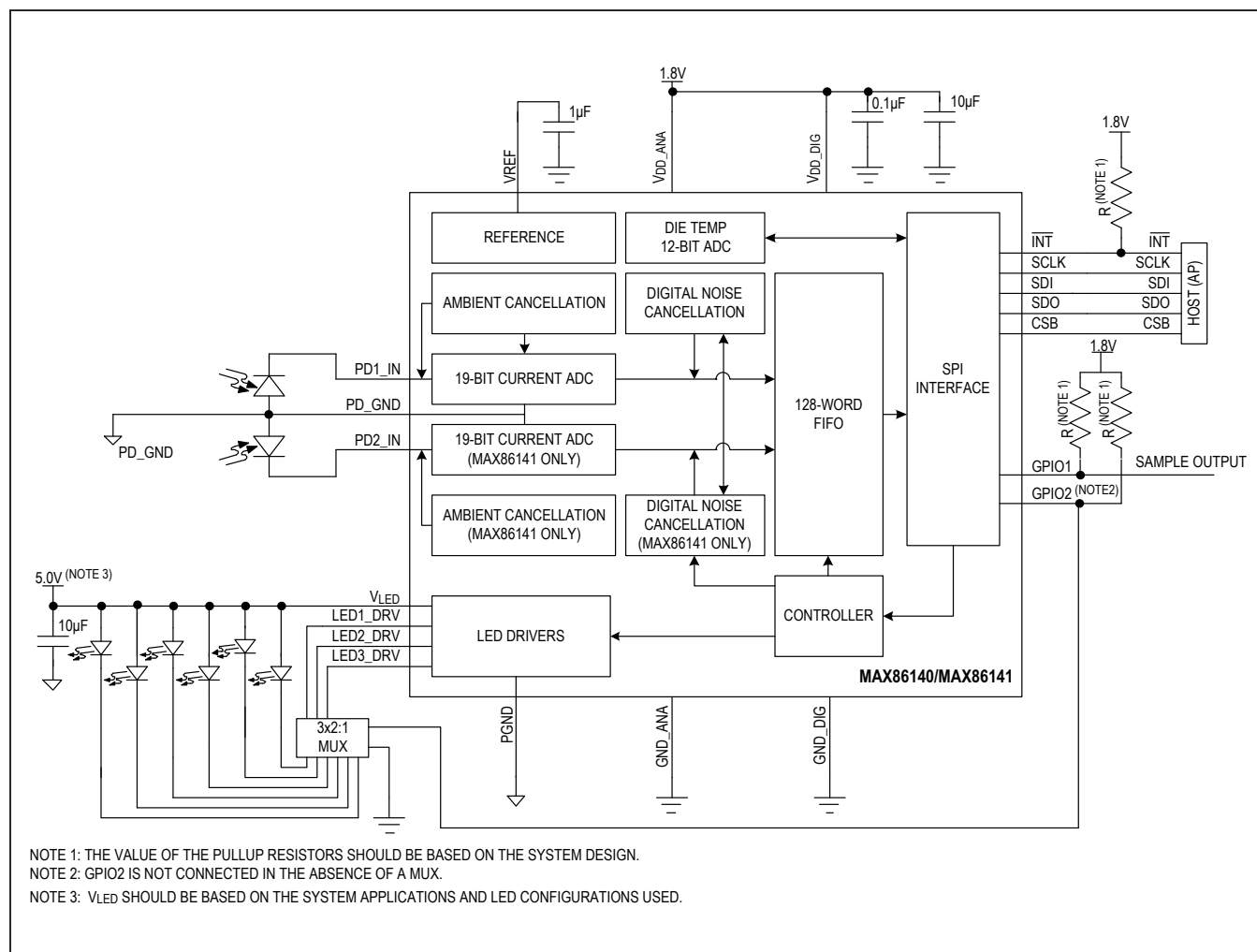


Figure 15. Block Diagram for GPIO CTRL[3:0] 0100 with External Mux

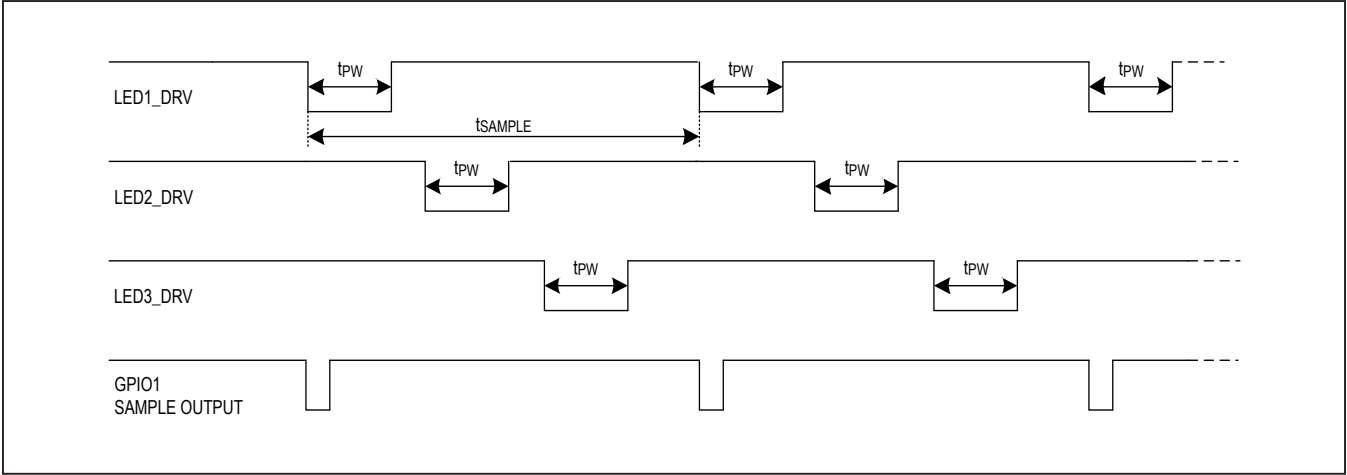


Figure 16. Timing Diagram for GPIO CTRL[3:0] 0100 Without External Mux

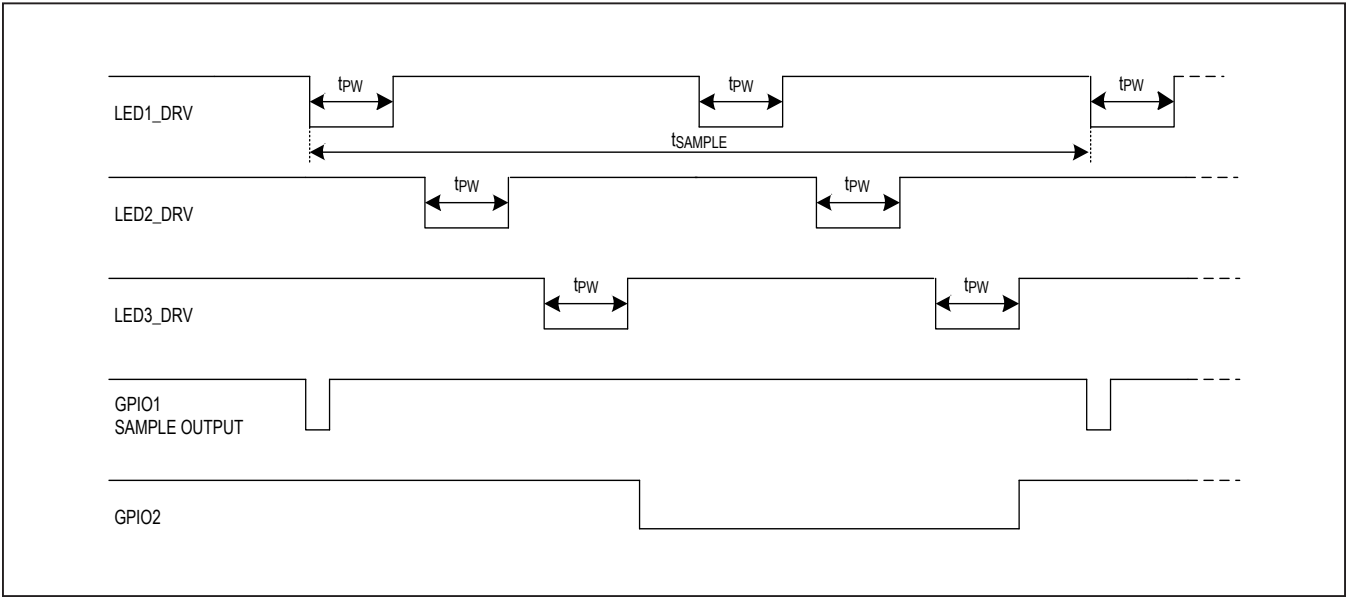


Figure 17. Timing Diagram for GPIO CTRL[3:0] 0100 with External Mux

MAX86140/
MAX86141

Best-in-Class Optical Pulse Oximeter and
Heart-Rate Sensor for Wearable Health

GPIO CTRL[3:0] 0101: Start of Sample Output with RTC Input Clock

Table 11. GPIO Mode 0101

| GPIO CTRL | GPIO1 FUNCTION | GPIO2 FUNCTION | COMMENT |
|-----------|---|------------------------------------|--|
| 0101 | Active Output Controller Sample Output | Input 32768/32000Hz Clock Input | GPIO1 is defined as a controller sample output. The GPIO1 output can be used to trigger a second sensor. When used with a second MAX86140/MAX86141 set to target sample mode, the controller sample timing will drive target sample time. GPIO2 is an input 32768/32000Hz. Exposure timing is controlled by GPIO2 clock input. |

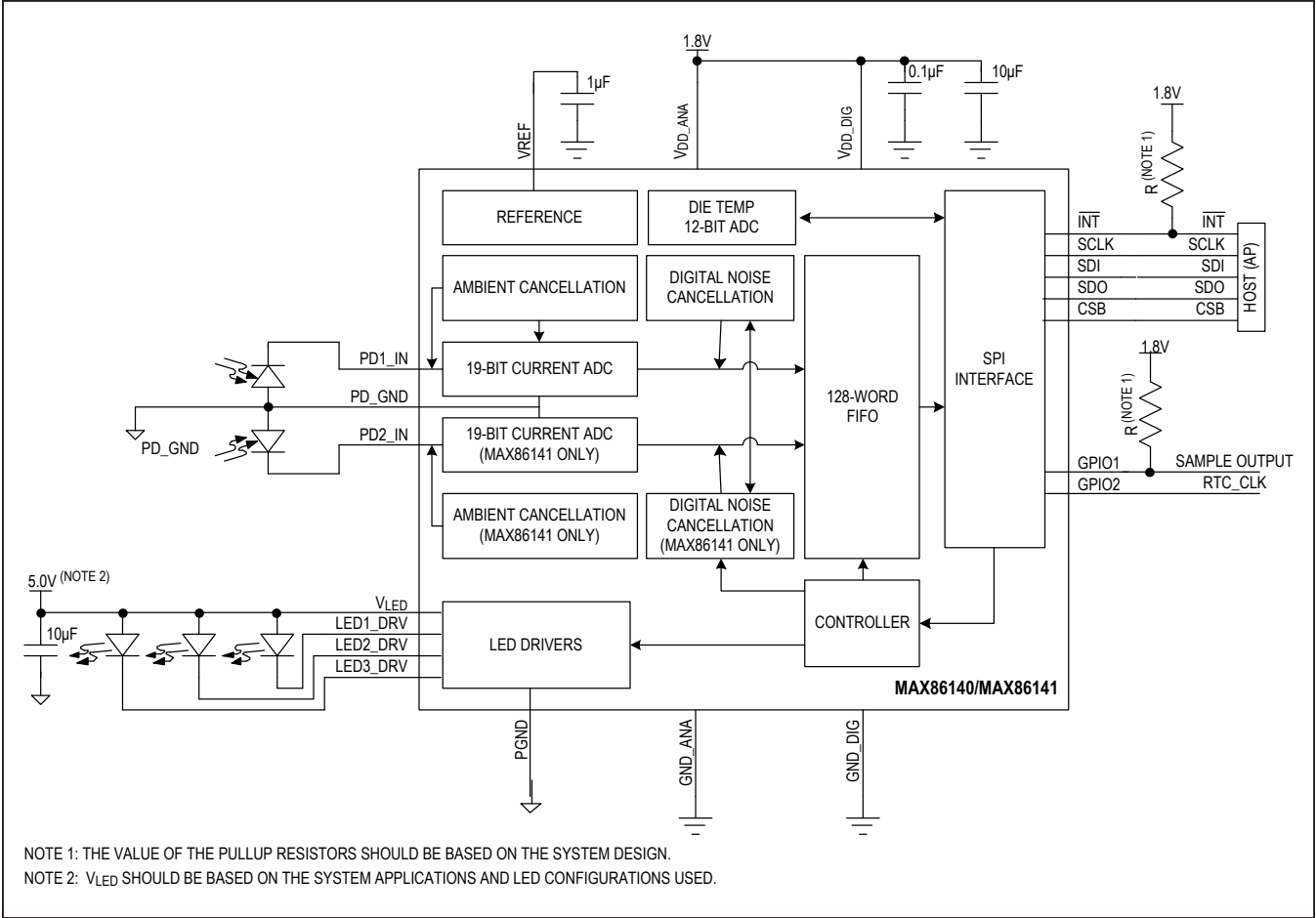


Figure 18. Block Diagram for GPIO CTRL[3:0] 0101

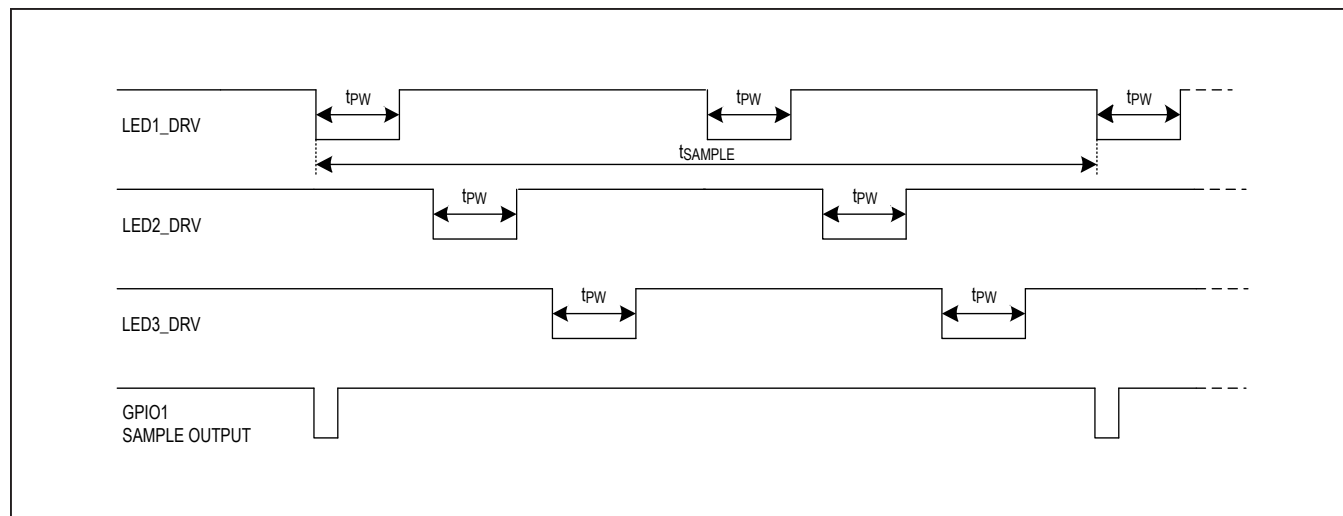


Figure 19. Timing Diagram for GPIO CTRL[3:0] 0101

GPIO CTRL[3:0] 0110 and 0111: Controller/Target with External Mux

Table 12. GPIO Mode 0110 and 0111

| GPIO CTRL | GPIO1 FUNCTION | GPIO2 FUNCTION | COMMENT |
|-----------|--|-------------------------|---|
| 0110 | Input Exposure Trigger | Tristate or Mux Control | GPIO1 is defined as an exposure trigger input (Target). This input can come from an external source or from another MAX86140 in controller sample mode. Both sample and exposure timing is controlled by the GPIO1 input. GPIO2 will be active if any of the LEDCn[3:0] states A, B, or C are enabled in the exposure sequence. In this case, GPIO2 will be high during exposures on LED4, LED5 or LED6, otherwise it will be low. If LEDCn[3:0] state A, B, or C is not enabled in the exposure sequence, GPIO2 will be tristate. |
| 0111 | Active Output Controller Exposure Output | Tristate or Mux Control | GPIO1 is defined as a controller sample output. The GPIO1 output can be used to trigger and second sensor. When used with a second MAX86140 set to target exposure mode, the controller exposure timing will drive target exposure time. GPIO2 will be active if any of the LEDCn[3:0] states A, B, or C are enabled in the exposure sequence. In this case, GPIO2 will be high during exposures on LED4, LED5 or LED6, otherwise it will be low. If LEDCn[3:0] state A, B, or C is not enabled in the exposure sequence, GPIO2 will be tristate. Sample and exposure time is controlled internally |

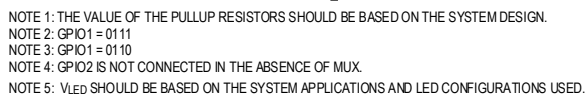


Figure 20. Block Diagram for GPIO CTRL[3:0] 0110 and 0111 With Two External Muxes

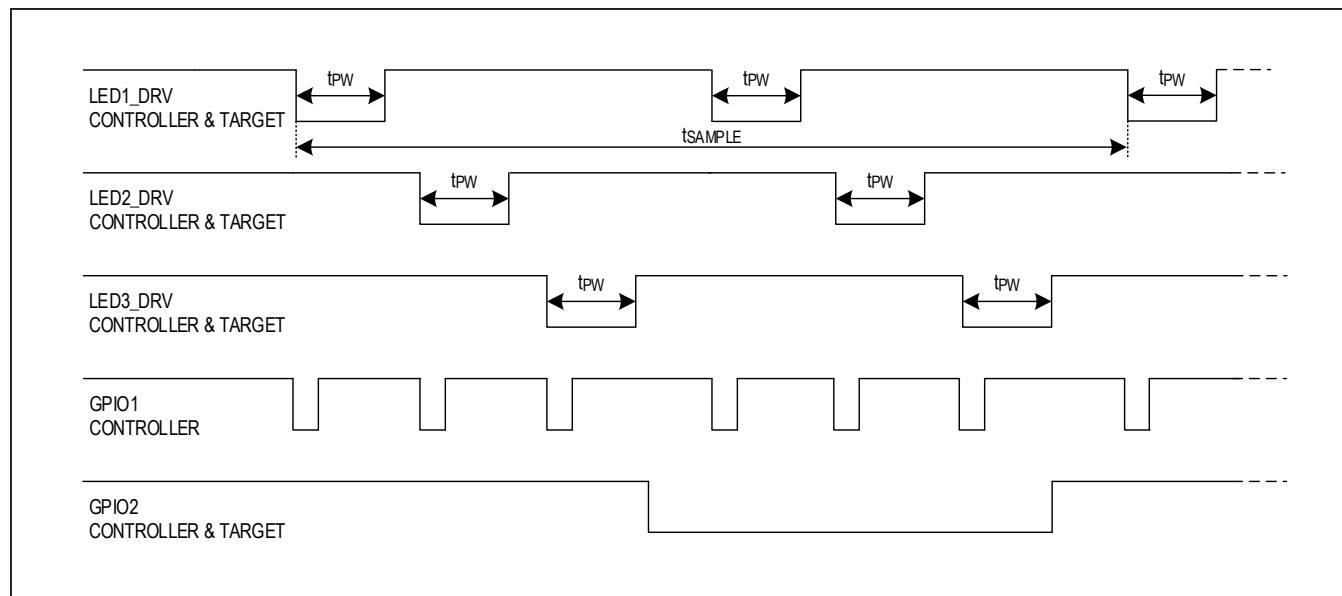
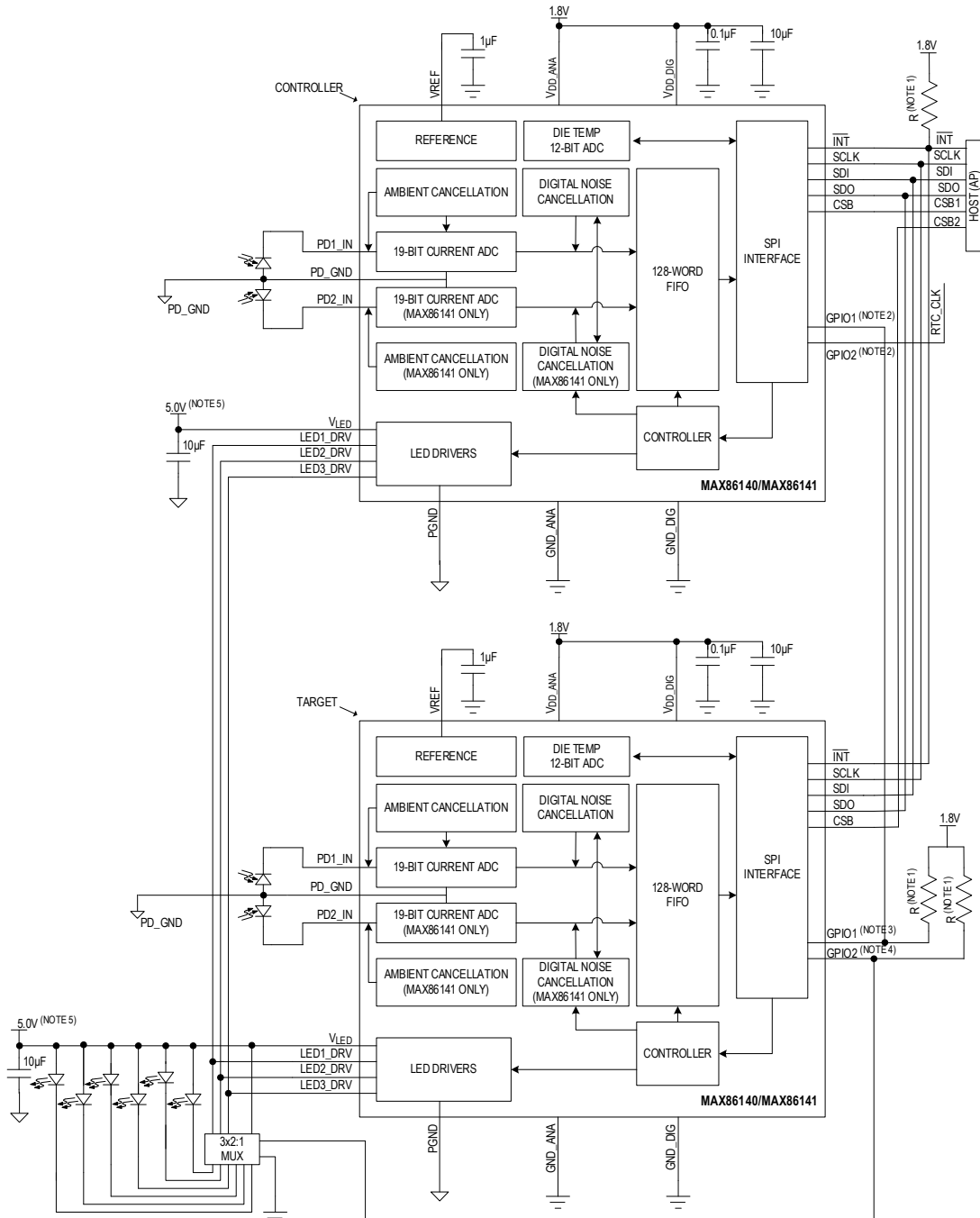


Figure 21. Timing Diagram for GPIO CTRL[3:0] 0110 and 0111 With External Mux

GPIO CTRL[3:0] 0110 and 1000: Controller/Target with and without External Mux

Table 13. GPIO Mode 0110 and 1000

| GPIO CTRL | GPIO1 FUNCTION | GPIO2 FUNCTION | COMMENT |
|-----------|--|---------------------------------|--|
| 0110 | Input Exposure Trigger | Tristate or Mux Control | GPIO1 is defined as an exposure trigger input (Target). This input can come from an external source or from another MAX86140/MAX86141 in controller sample mode. Both sample and exposure timing is controlled by the GPIO1 input. GPIO2 will be active if any of the LEDCn[3:0] states A, B, or C are enabled in the exposure sequence. In this case, GPIO2 will be low during exposures on LED4, LED5, or LED6, otherwise it will be high. If LEDCn[3:0] state A, B, or C is not enabled in the exposure sequence, GPIO2 will be tristate unless externally pulled up. |
| 1000 | Active Output Controller Exposure Output | Input 32768/32000Hz Clock Input | GPIO1 is defined as a controller sample output. The GPIO1 output can be used to trigger and second sensor. When used with a second MAX86140 /MAX86141 set to target exposure mode, the controller exposure timing will drive target exposure time. GPIO2 is an input 32768/32000Hz. Sample and exposure timing is controlled by GPIO2 clock input. |



NOTE 1: THE VALUE OF THE PULLUP RESISTORS SHOULD BE BASED ON THE SYSTEM DESIGN.
 NOTE 2: GPIO = 1000
 NOTE 3: GPIO = 0110
 NOTE 4: GPIO2 IS A NO CONNECT IN THE ABSENCE OF EXTERNAL MUX.
 NOTE 5: V_{LED} SHOULD BE BASED ON THE SYSTEM APPLICATIONS AND LED CONFIGURATIONS USED.

Figure 22. Block Diagram for GPIO CTRL[3:0] 0110 and 1000 with External Mux

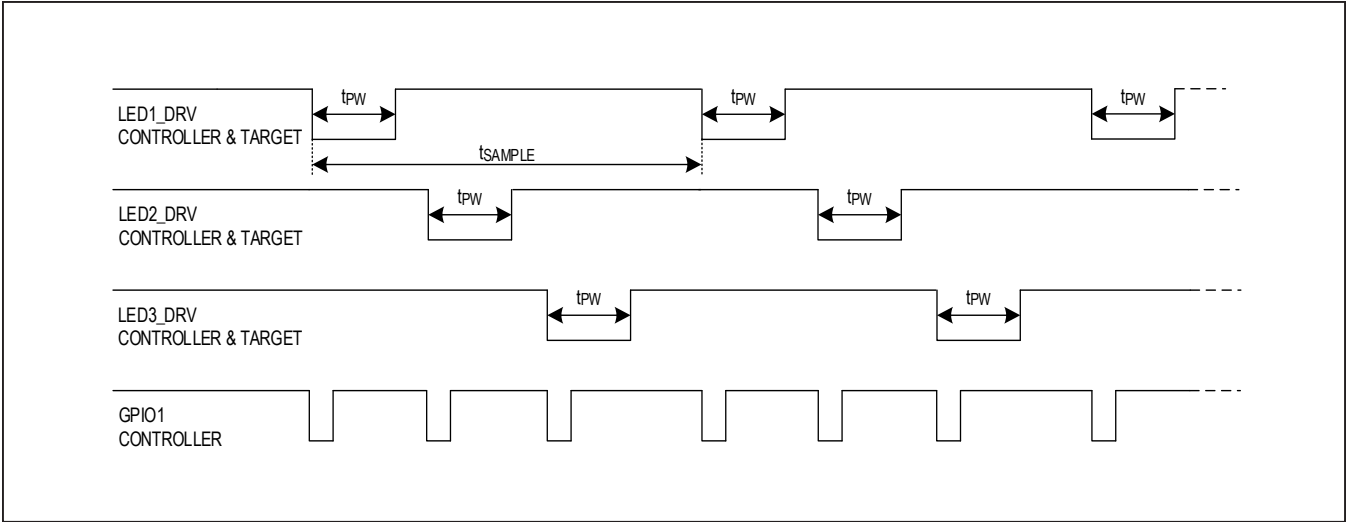


Figure 23. Timing Diagram for GPIO CTRL[3:0] 0110 and 1000 Without External Mux

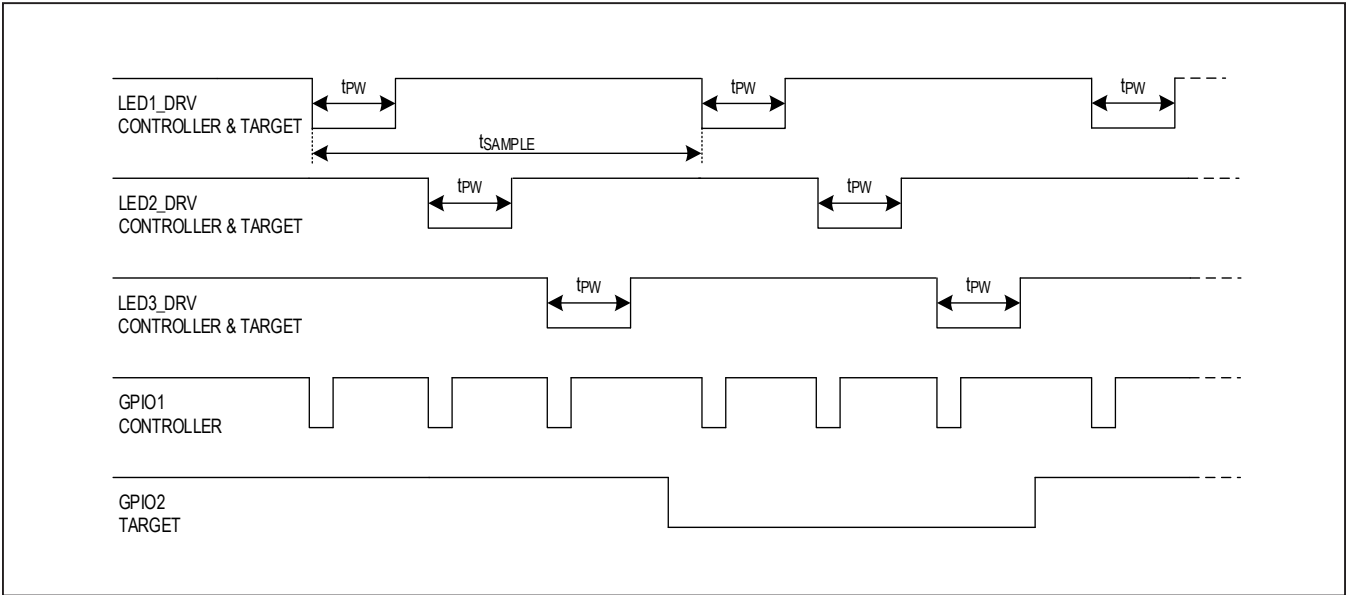


Figure 24. Timing Diagram for GPIO CTRL[3:0] 0110 and 1000 with External Mux

GPIO CTRL[3:0] 1001 Hardware Sync

Table 14. GPIO Mode 1001

| GPIO CTRL | GPIO1 FUNCTION | GPIO2 FUNCTION | COMMENT |
|-----------|------------------------|---|--|
| 1001 | Input HW_FORCE_SYNC | Input 32768Hz or 32000Hz Clock Input | GPIO1 is defined as a start of sample sync input. The rising edge of GPIO1 causes the present sample sequence to be terminated and reinitiated on the next rising edge of GPIO2 input. GPIO2 is an input 32768/32000Hz. Sample and exposure timing is controlled by GPIO2 clock input. |

GPIO CTRL[3:0] 1010 Hardware Sync

Table 15. GPIO Mode 1010

| GPIO CTRL | GPIO1 FUNCTION | GPIO2 FUNCTION | COMMENT |
|-----------|-------------------------------|----------------------------|---|
| 1010 | Input Sample Sync ONE_SHOT | Tristate or Mux Control | GPIO1 is defined as a start of the power-up sequence for one sample. The falling edge of GPIO1 starts the power-up sequence followed by the exposure sequence as programmed in the LEDCn[3:0] registers. After the sample data is pushed to the FIFO, the device waits for the next Sample Sync pulse on GPIO1. GPIO2 is active if any of the exposure sequence registers LEDCn[3:0] are set to value 0xA, 0xB, or 0xC. In this case, GPIO2 is low during exposures on LED4, LED5, or LED6, respectively. Otherwise, GPIO2 is high. If the exposure sequence registers LEDCn[3:0] are not state 0xA, 0xB, or 0xC, GPIO2 is tristate. Sample period and exposure time are controlled internally. |

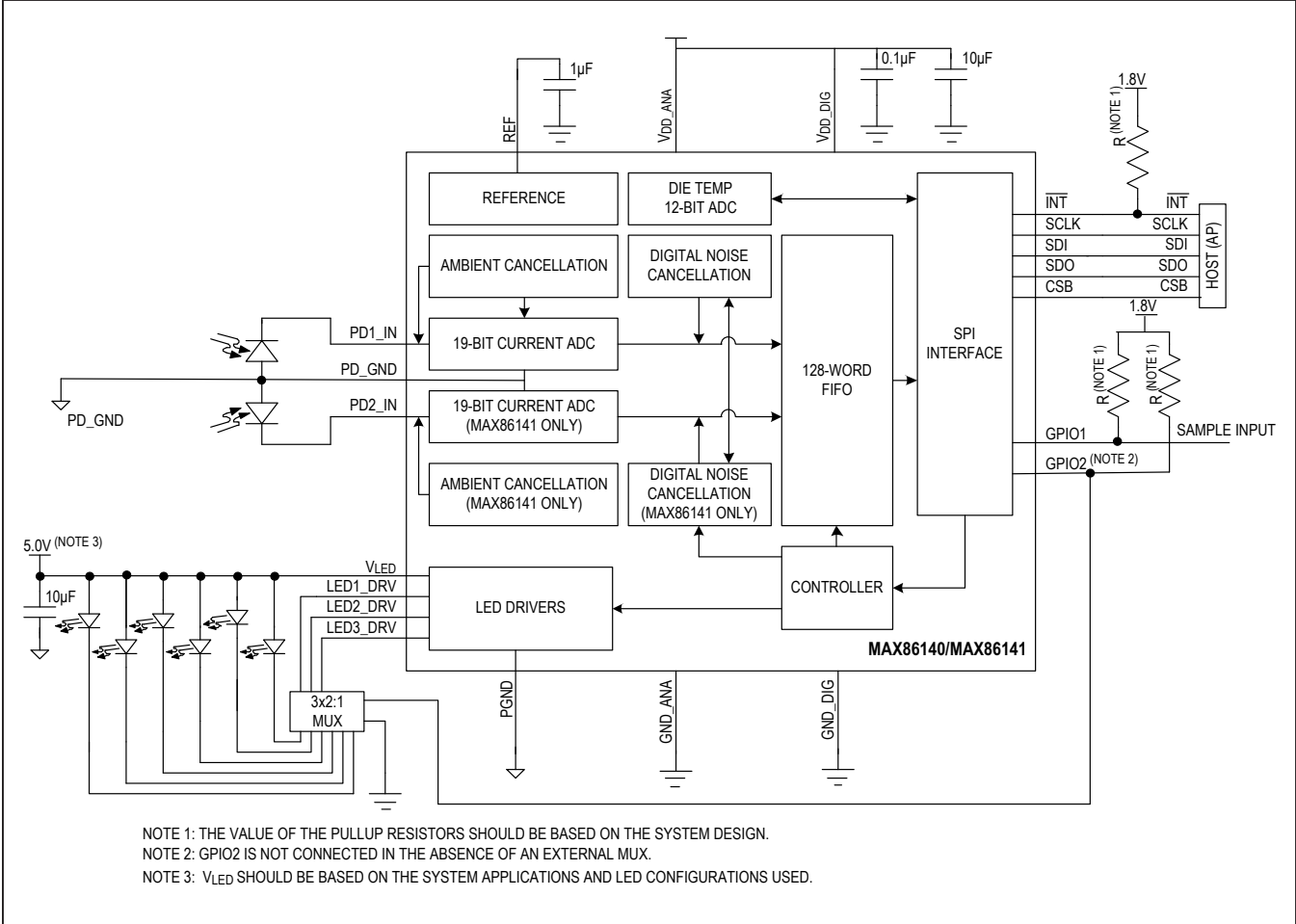


Figure 25. Block Diagram for GPIO CTRL[3:0] 1010

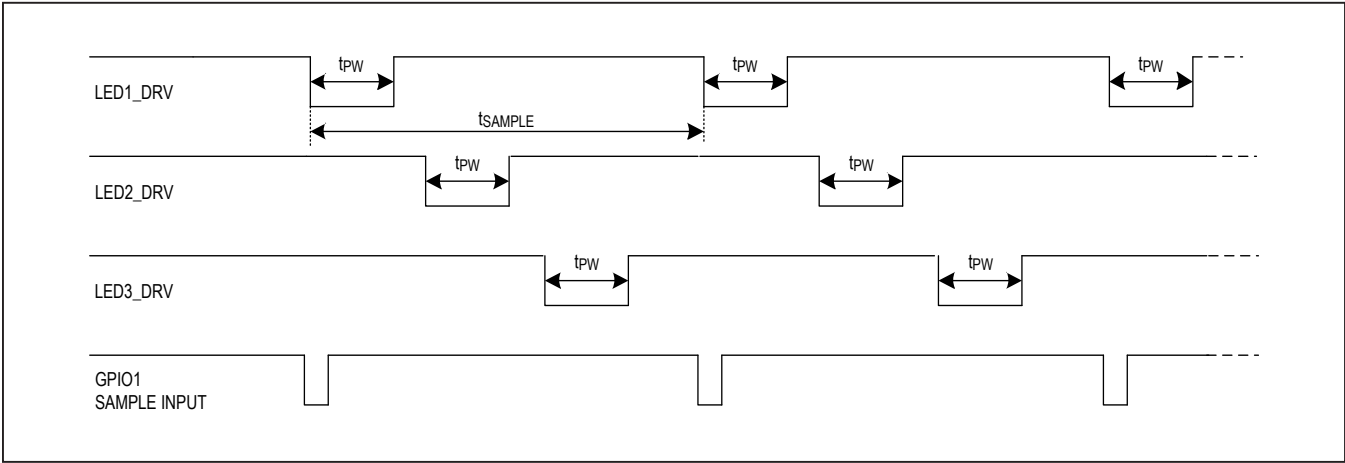


Figure 26. Timing Diagram for GPIO CTRL[3:0] 1010 Without External Mux

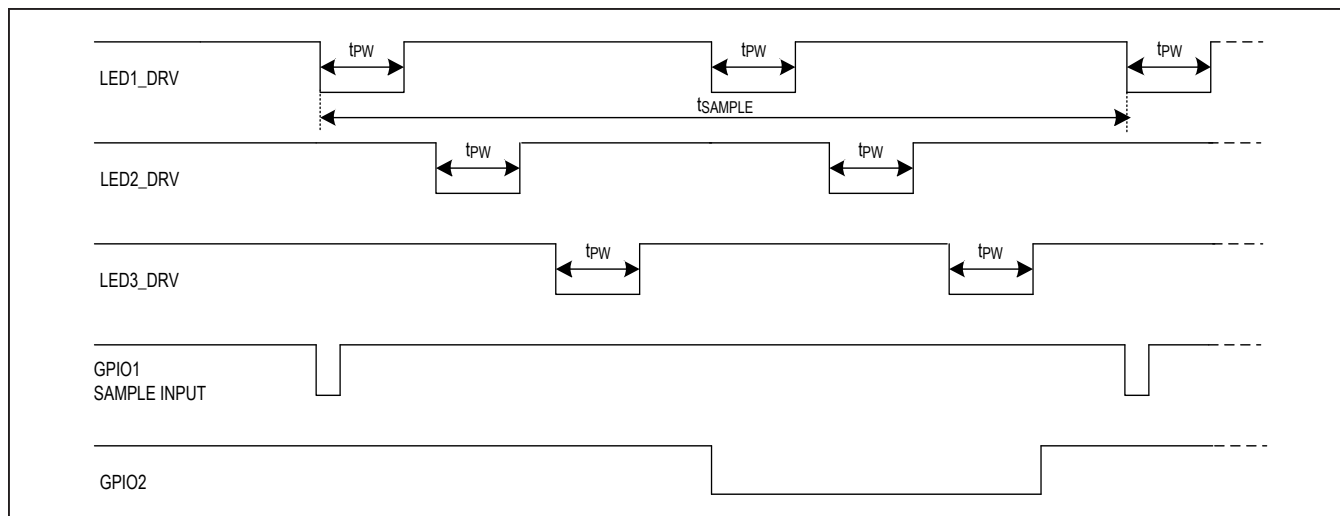


Figure 27. Timing Diagram for GPIO CTRL[3:0] 1010 With External Mux

Proximity Mode Function

The MAX86140/MAX86141 includes an optical proximity function that could significantly reduce energy consumption and extend battery life when the sensor is not in contact with the skin. Proximity mode is enabled by setting PROX_INT_EN bit field to 1 in the Interrupt Enable 2 register (address 0x02[4]), setting a threshold in the PROX_INT_THRESH register (address 0x14) and assigning an LED current in the PILOT_PA (address 0x29). Proximity mode also requires that LED Sequence Register 1, field LEDC1 (address [3:0]) to be assigned to a specific measurement and that measurement is correctly connected to a light source. The LEDC1 measurement is used to detect the optical presents of a reflecting object in proximity mode and thus must be valid for proximity mode to work.

When enabled, the Proximity Detect Interrupt (register 0x01[4]) will be asserted and proximity mode will be entered when the value of the measurement assigned to LEDC1 drops below the PROX_INT_THRESH. When entering proximity mode, the MAX86140/MAX86141 will drop the current to the LED(s) assigned to LEDC1 to PILOT_PA value, reduce the sample rate to 8sps and operates in Low Power mode. The intent here is to both reduce the consumed LED current and MAX86140/MAX86141 power to a minimum during situations where there is no reflective returned signal. It is also intended to reduce the emitted light to a minimum or even below that perceivable by the human eye.

When the proximity mode is enabled and the measurement assigned to LEDC1 with the LED current in PILOT_PA

exceeds the PROX_INT_THRESH, the MAX86140/MAX86141 also generates a Proximity Detect Interrupt (register 0x01[4]). In such an event MAX86140/MAX86141 will switch to normal mode, changing the sample rate to that assigned in PPG Configuration 2 register (address 0x12) bit field PPG_SR and the LED current assigned to the measurement of LEDC1. Therefore, the MAX86140 is able to switch to proximity mode and back to normal mode without microprocessor interaction.

The threshold applied to PROX_INT_THRESH should be well below that of a usable signal at the maximum LED current applied to LEDC1 but high enough to not be triggered by noise from distant objects. Further the current assigned to PILOT_PA should be much lower than that assigned to LEDx_DRV in normal mode. This will ensure that the signal obtained from LEDC1 drops significantly when entering proximity mode, thus providing enough hysteresis to eliminate multiple interrupts being generated at the proximity/normal mode transition.

To guarantee that MAX86140/MAX86141 will successfully transition from proximity mode to normal mode, the PROX_INT_THRESH should be low enough and the PILOT_PA high enough to ensure that the device mounted on the darkest of skins returns a signal above the PROX_INT_THRESH at the PILOT_PA current.

Note that proximity mode is only available to LEDC1 measurements that are made with PD1_IN optical channel without an external mux. When proximity mode is active, LEDC2 to LEDC6 is ignored. The threshold applied to PROX_INT_THRESH register are in units of 2048LSBs.

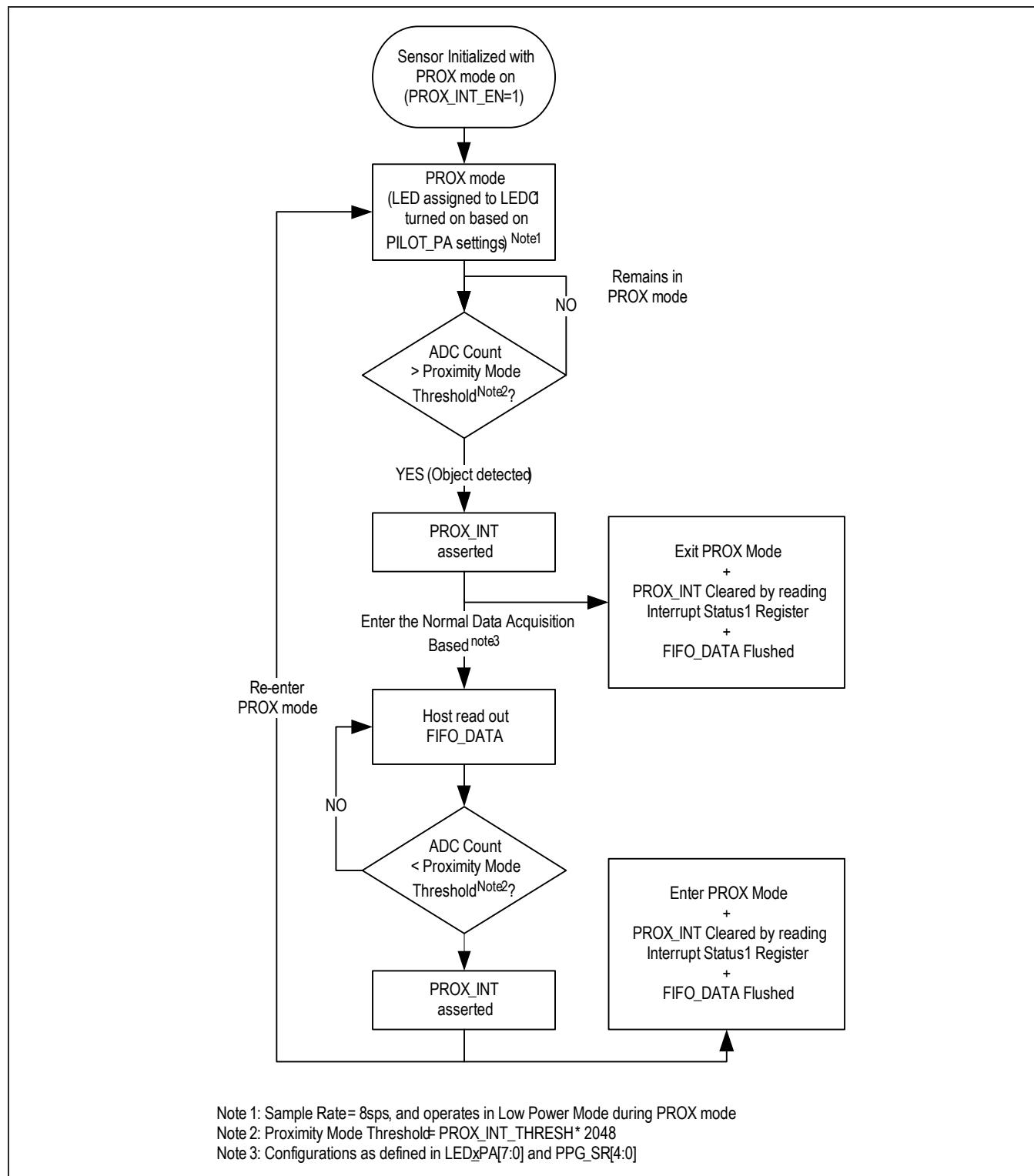


Figure 28. Proximity Function Flow Diagram

Picket Fence Detect-and-Replace Function

Under typical situations, the rate of change of ambient light is such that the ambient signal level during exposure can be accurately predicted and high levels of ambient rejection are obtained. However, it is possible to have situations where the ambient light level changes extremely rapidly, for example when in a car with direct sunlight exposure passes under a bridge and into a dark shadow. In these situations, it is possible for the MAX86140/MAX86141 ambient light correction (ALC) circuit to fail and produce an erroneous estimation of the ambient light during the exposure interval. The built-in picket fence detect-and-replace function corrects the final PPG results in case of an ALC circuit failure due to these extreme conditions.

The picket fence detect-and-replace function works on the basis that the extreme conditions causing a failure of the ALC are rare events. These events resulting in a large deviation from the past sample history of a normal PPG riding on a motion effect signal, which normally would change relatively slowly with respect to the sampling interval. Under these conditions, it is possible to detect sample values that are well outside the normal sample to sample deviation and replace those samples with an extrapolated value based on the relatively recent history of samples.

The picket fence detect-and-replace function is disabled by default and is enabled by setting PF_ENABLE (register 0x16[7]) bit to 1. The function begins with detecting a picket fence event. Detection is done by taking the absolute value of the difference between the present ADC converted value and a predicted point, called an estimation error, and comparing this estimation error to a threshold. If the estimation error exceeds the threshold, then the present ADC converted point is considered a picket fence event.

The predicted point referred to above is computed in one of two ways, set by the value in the PF_ORDER (address 0x16[6]) bit. If PF_ORDER = 0 the predicted point is simply the previous ADC converted point. If PF_ORDER = 1, the predicted point is a least square fit extrapolation based on the previous four picket fence outputs, which, under normal circumstances, is identical to the ADC converted inputs.

The threshold used in detecting a picket fence event is a low pass version of the running estimation error computed above times a multiplier. The multiplier used is set by the THRESHOLD_SIGMA_MULT (address 0x16[1:0]) bits and can be 4, 8, 16, or 32 times the running low-pass filter output of the estimation error.

The low-pass filter function is controlled by two parameters, the IIR_TC (address 0x16[5:4]) bits and IIR_INIT_VALUE (address 0x16[3:2]) bits. The IIR_TC bits control the filter's time constant and are adjustable from 8 to 64 samples. The IIR_INIT_VALUE bits control the initial values for the IIR low pass filter when the algorithm is initialized.

[Figure 29](#) illustrates the function in block diagram form. If the picket fence detect-and-replace function is enabled the input from the ADC, $s(n)$ generates $p(n)$ in a way that is dependent on the value of the PF_ORDER bit. Value $s(n)$ is subtracted from $p(n)$ and turned into a positive number $d(n)$ and fed into the IIR low-pass filter producing value $lpf(n)$. The output of the low pass filter $lpf(n)$ is then multiplied by a user constant, THRESHOLD_SIGMA_MULT to produce the picket fence threshold, PFT(n). The value $d(n)$ is then compared to this threshold and if greater than the PFT(n), the point $s(n)$ is replaced with the point $p(n)$.

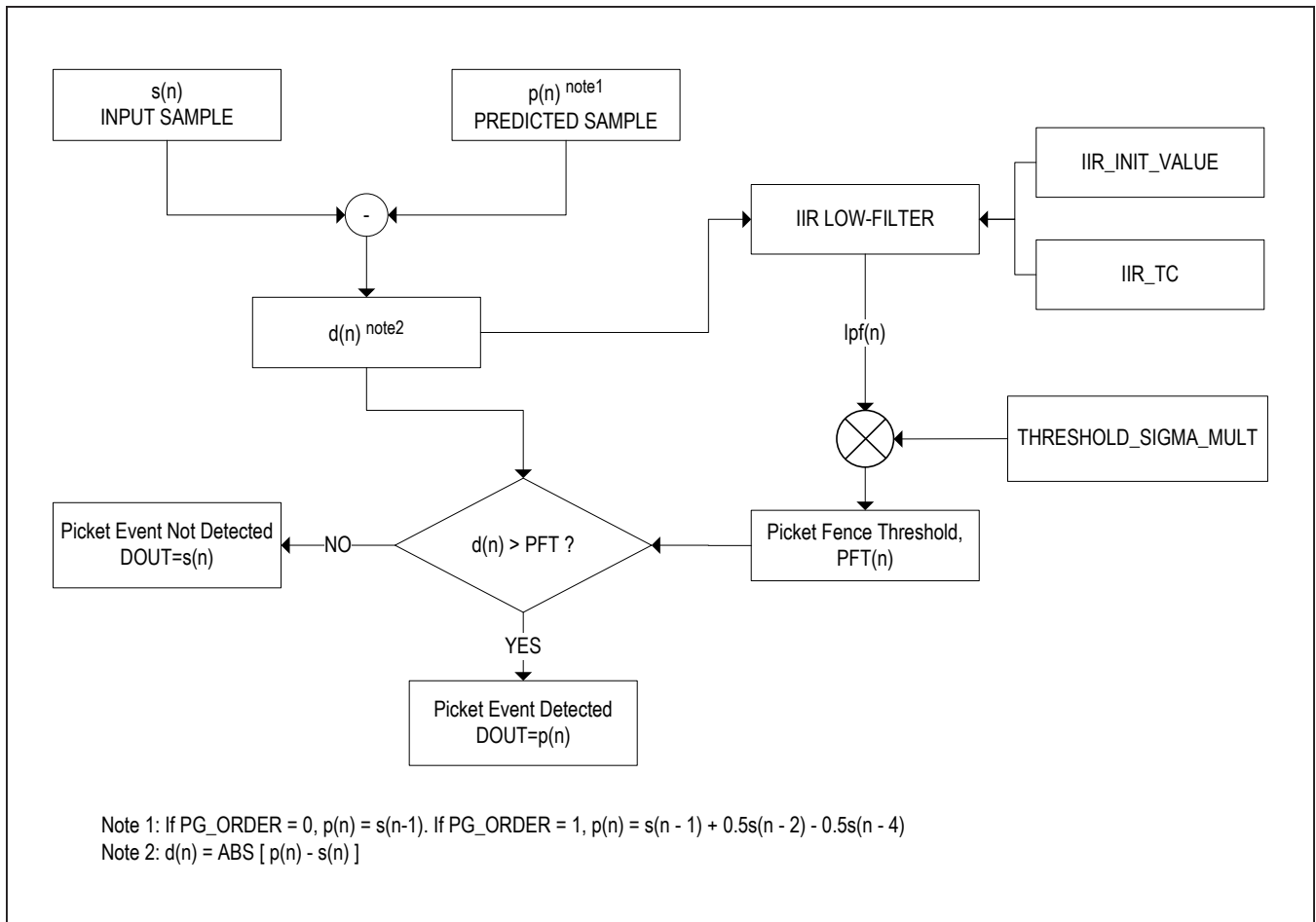


Figure 29. Picket Fence Detect-and-Replace Function Flow

This scheme essentially produces a threshold that tracks the past returned optical signal with a band width based on the past historical change sample to sample. [Figure 30](#) illustrates graphically how the threshold detection scheme works on a real PPG signal. Note that the black trace is

the real ADC sample points, the red trace is the output of the low-pass filter of the error estimation mirrored around the ADC points and the blue traces are the threshold values.

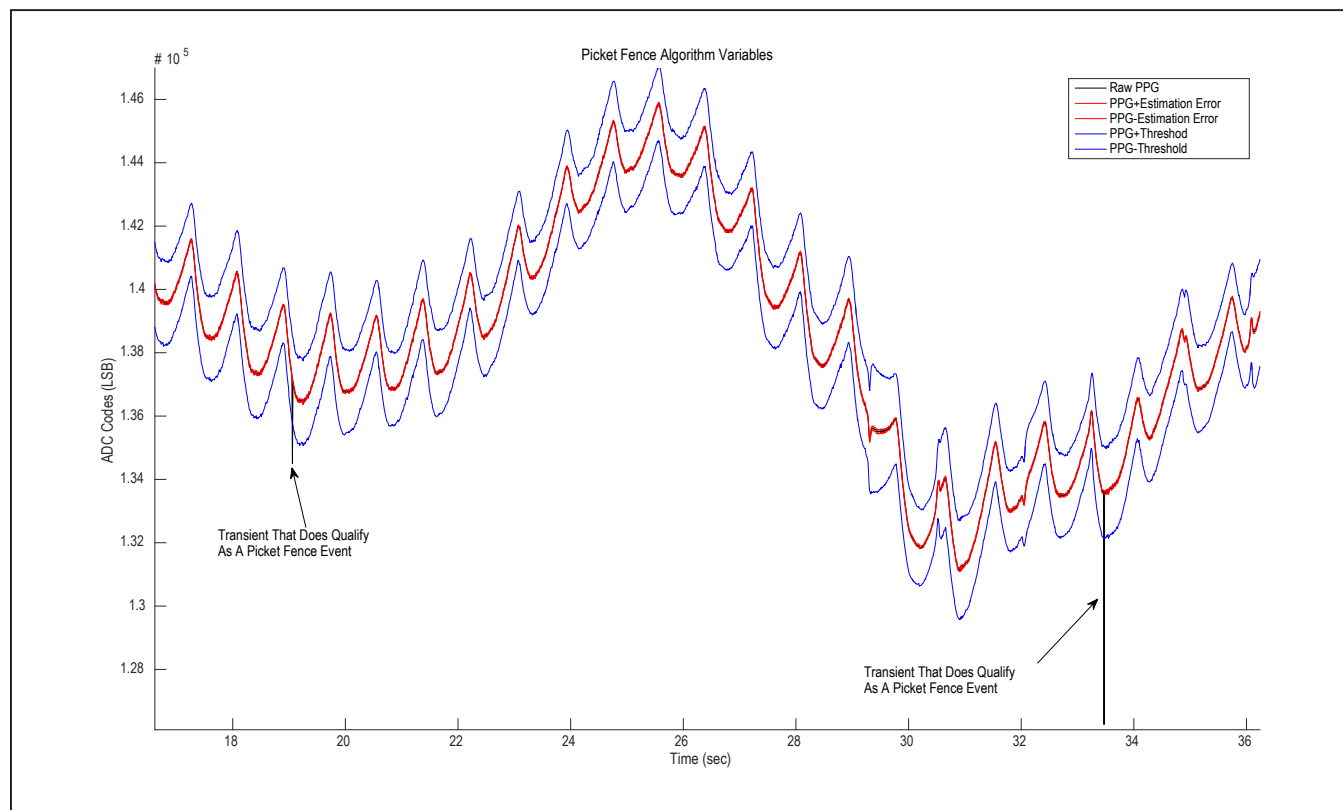


Figure 30. Picket Fence Variables In A PPG Waveform

The recommended settings for the picket fence detect-and-replace function are the default power on reset values for all registers except THRESHOLD_SIGMA_MULT bits. Here it is recommended that the 32x value 0x3 be used so only large excursions are classified as picket fence events. Lower values of THRESHOLD_SIGMA_MULT can cause the algorithm to go off track with extremely noisy waveform.

Photodiode Biasing

The MAX86140/MAX86141 provides multiple photodiode biasing options (see [Table 16](#)). These options allow the MAX86140/MAX86141 to operate with a large range of photodiode capacitance. The PDBIASx settings adjust the PD_IN bias point impedance to ensure that the photodiode settles rapidly enough to support the sample timing.

With a higher PD bias setting, the input-referred noise of the MAX86140/MAX86141 is increased. The relationship between Photodiode bias and noise with increasing photodiode capacitance is illustrated in the "Input Referred Noise vs. PD Capacitance" graph of the [Typical Operating Characteristics](#) section. Because of the increased noise with a higher PDBIASx setting, the lowest recommended PDBIASx settings should be used for a given photodiode capacitance.

Layout Guidelines

The MAX86140/MAX86141 is a high dynamic range analog front-end (AFE) and its performance can be adversely impacted by the physical printed circuit board (PCB) layout. It is important that all bypass recommendations in the pin table are followed. Specifically, it is recommended that the V_{DD_ANA} and V_{DD_DIG} pins be shorted at the PCB. It is also recommended that GND_ANA, GND_DIG, and PGND be shorted to a single PCB ground plane. These three pins should be shorted on the edge of the WLP grid array. The pins should be connected using a single via to the PCB ground plane as close to the AFE as possible. Use multiple vias if grounds are not coplanar.

The combined V_{DD_ANA} and V_{DD_DIG} pins should then be decoupled with a 22μF 0402 and a 0.1μF ceramic chip capacitor to the PCB ground plane. Note that effective capacitance is approximately 10μF with voltage derating. In addition, the VREF pin should be decoupled to the PCB GND plane with a 1.0μF ceramic capacitor. The voltage on the VREF pin is nominally 1.21V, so a 6.3V rated ceramic capacitor should be adequate for this purpose. It is recommended that all decoupling caps use individual vias to the PCB GND plane to avoid mutual impedance coupling between decoupled supplies when sharing vias.

The most critical aspect of the PCB layout of MAX86140/MAX86141 is the handling of the PD_IN and PD_GND nodes. Parasitic capacitive coupling to the PD_IN can result in additional noise being injected into the MAX86140/MAX86141 front-end. To minimize external interference coupling to PD_IN, it is recommended that the PD_IN node be fully shielded by the PD_GND node. An example of this recommendation is shown below. In the three layers shown, the PD_IN node is shielded with a coplanar PD_GND trace on the top layer, the layer on which the MAX86140/MAX86141 is mounted. Ensure that no traces route adjacent to the PD_IN vias. PD_GND fill should be isolating PD_IN vias from other traces. There should be no traces or vias other than PDx_IN going through PD_GND. Do not route traces over PD_GND (i.e., the PCB GND plane should be adjacent to PD_GND). On the bottom layer, the photodiode cathode is entirely shielded with the PD_GND shield, which is also the photodiode anode. Note, also, that the PD_GND shield also is extended below the photodiode. This is done because, in most photodiodes, the cathode is the bulk of the silicon. Therefore, shielding beneath the photodiode will terminate the capacitance to the bulk or cathode side to the reference node (PD_GND). Finally, the PD_GND pin should be attached to the PCB GND in only one point. This is shown on the top layer.

Table 16. Recommended PDBIASx Values Based on the Photodiode Capacitance

| PDBIASx[2:0] | PHOTODIODE CAPACITANCE |
|------------------|------------------------|
| 0x001 | 0pF to 65pF |
| 0x101 | 65pF to 130pF |
| 0x110 | 130pF to 260pF |
| 0x111 | 260pF to 520pF |
| All other values | Not recommended |

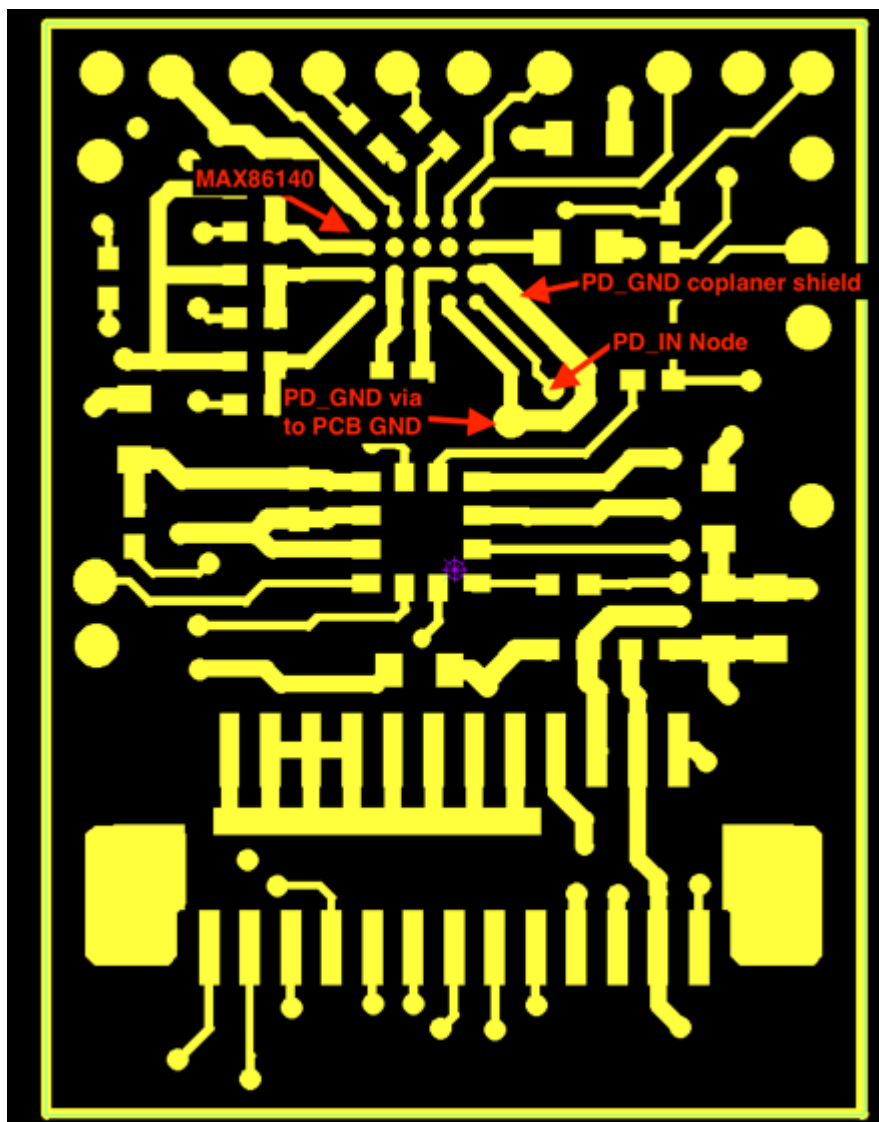


Figure 31. Example PCB Layout, Layer 1 (Top, MAX86140)

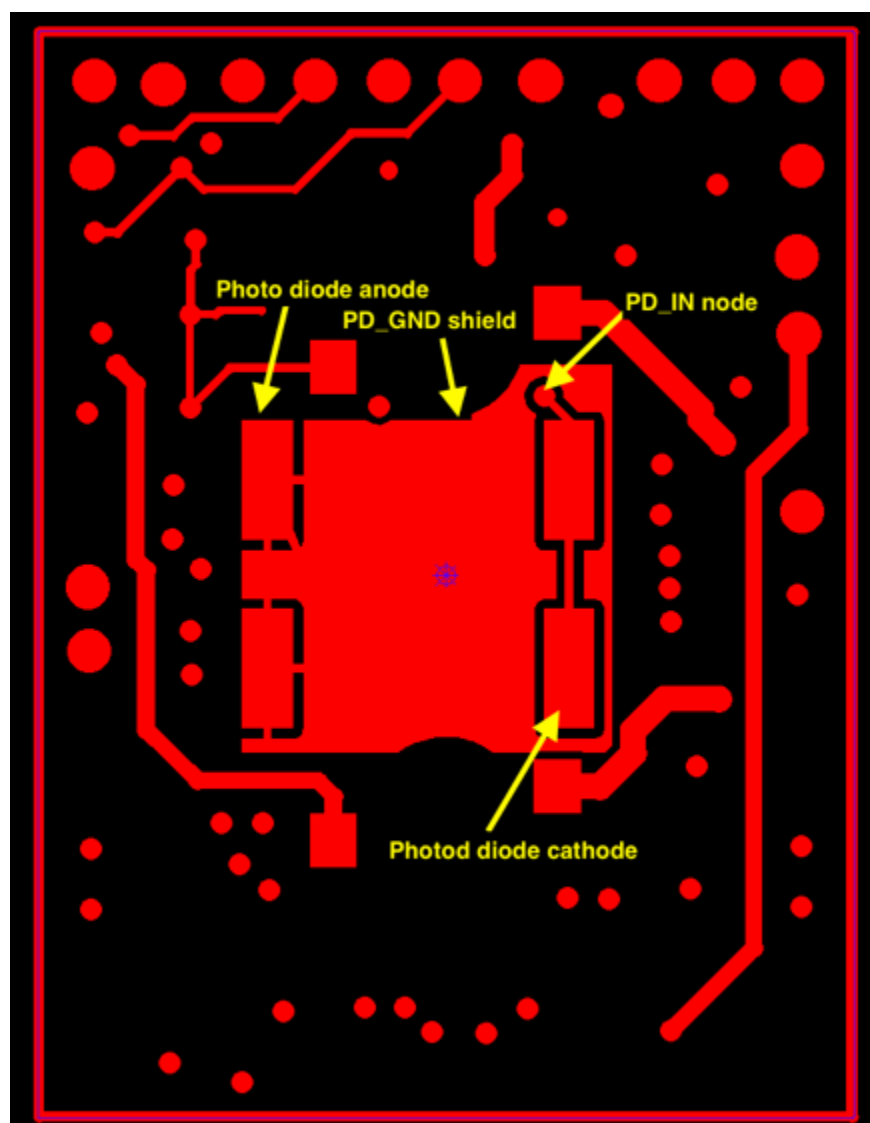


Figure 32. Example PCB Layout, Layer 6 (Bottom, Optical Layer)

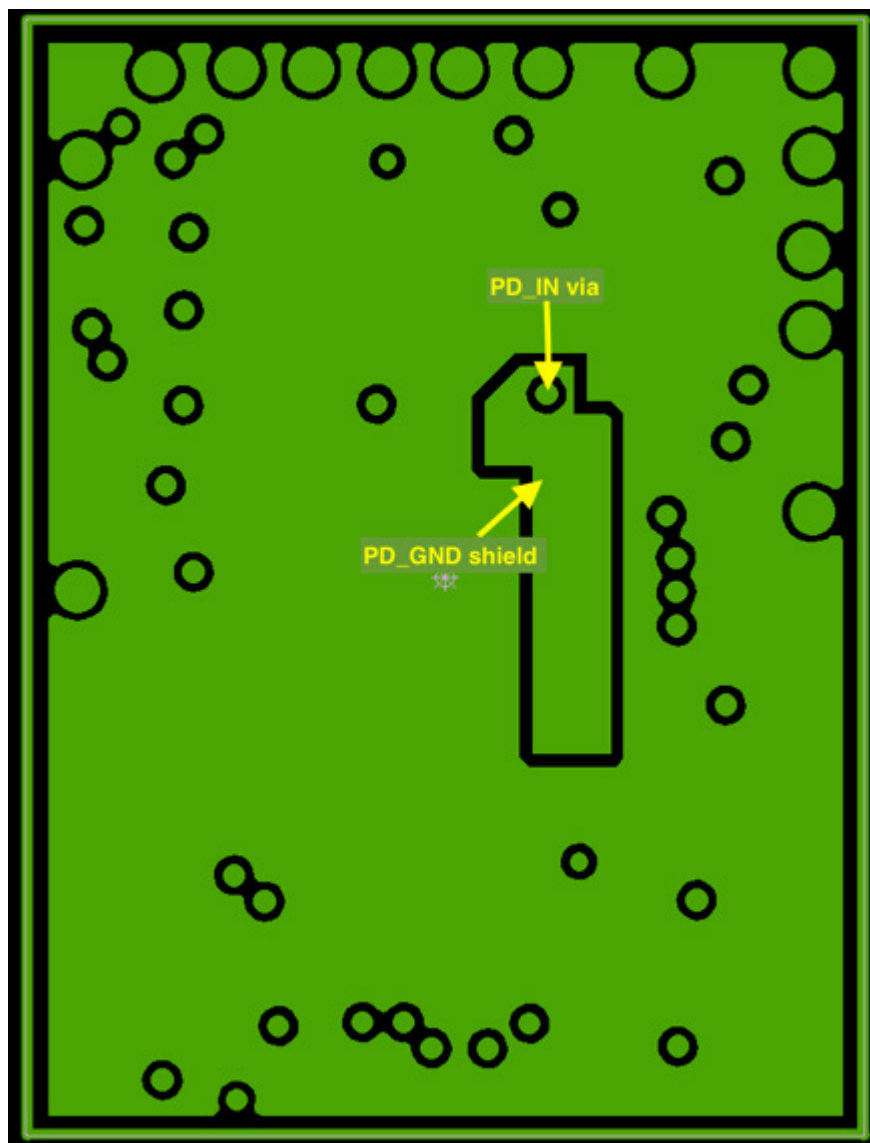


Figure 33. Example PCB Layout, Layer5 (Ground Layer)

SPI Timing

Detailed SPI Timing

The detailed SPI timing is illustrated below. The timings indicated are all specified in the [Electrical Characteristics](#) table.

Single-Word SPI Register Read/Write Transaction

The MAX86140/MAX86141 is SPI/QSPI/Micro-wire/DSP compatible. The operation of the SPI interface is shown below. Data is strobed into the MAX86140/MAX86141 on the SCLK rising edge while clocked out on the SCLK falling edge. All single-word SPI read and write operations are done in a 3-byte, 24 clock cycle SPI instruction framed by a CSB low interval. The content of the SPI operation consists of a one-byte register address (A[7:0]), followed by

a one-byte command word that defines the transaction as write or read, followed by a single-byte data word either written to, or read from, the register location provided in the first byte.

Write mode operations will be executed on the 24th SCLK rising edge using the first three bytes of data available. In write mode, any data supplied after the 24th SCLK rising edge will be ignored. Subsequent writes require CSB to deassert high and then assert low for the next write command. A rising CSB edge preceding the 24th rising edge of SCLK by t_{CSA} (detailed SPI timing diagram), will result in the transaction being aborted.

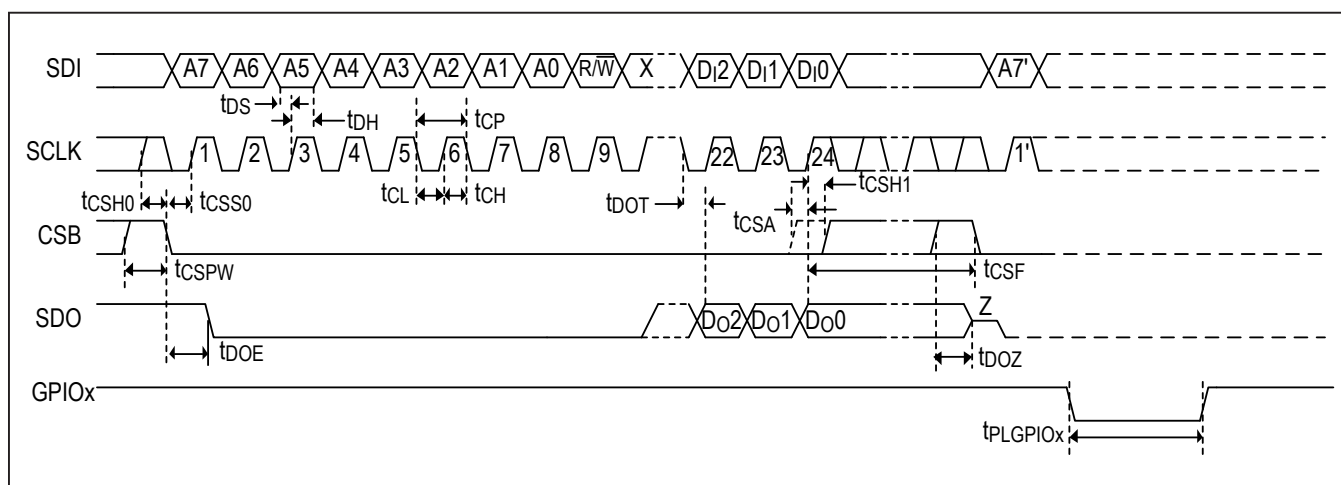


Figure 34. Detailed SPI Timing Diagram

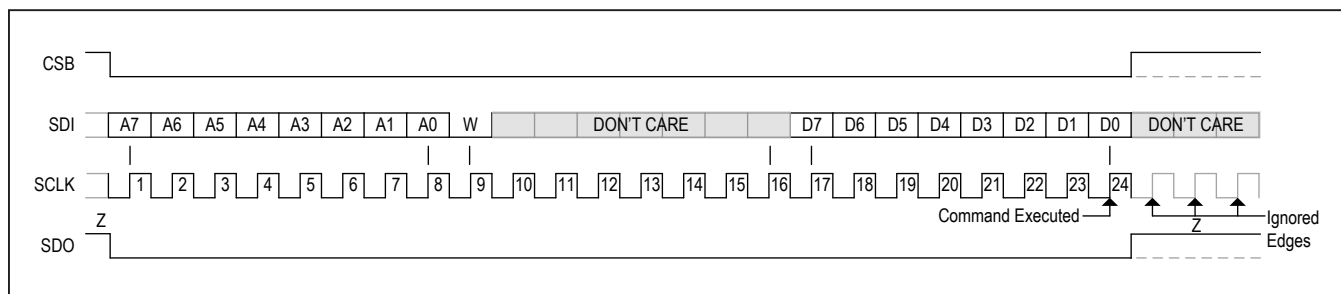


Figure 35. SPI Write Transaction

Read mode operations will access the requested data on the 16th SCLK rising edge, and present the MSB of the requested data on the following SCLK falling edge, allowing the μ C to latch the data MSB on the 17th SCLK rising edge. Configuration and status registers are available through normal mode readback sequences. FIFO reads must be performed with a burst mode FIFO read (see [SPI FIFO Burst Mode Read Transaction](#)). If more than 24 SCLK rising edges are provided in a normal read sequence, the excess edges will be ignored and the device will read back zeros.

SPI FIFO Burst Mode Read Transaction

The MAX86140/MAX86141 provides a FIFO burst read mode to increase data transfer efficiency. The first 16 SCLK cycles operate exactly as described for the normal

read mode, the first byte being the register address, the second being a read command. The subsequent SCLKs consist of FIFO data, 24 SCLKs per word. All words in the FIFO should be read with a single FIFO burst read command.

Each FIFO sample consists of 3 bytes per sample and thus requires 24 SCLKs per sample to readout. The first byte (SCLK 17 to 24) consists of a tag indicating the data type of the subsequent bits. Following the tag is the MSBs of the subsequent data (MSB, MSB-1, and MSB-2). The next byte (SCLK 25 to 32) consists of data bits MSB-3 to MSB-11. The final byte of each sample (SCLK 33 to 40) consists of the data LSB bits. The number of words in the FIFO depends on the FIFO configuration. See [FIFO Configuration](#) for more details the FIFO configuration and readout.

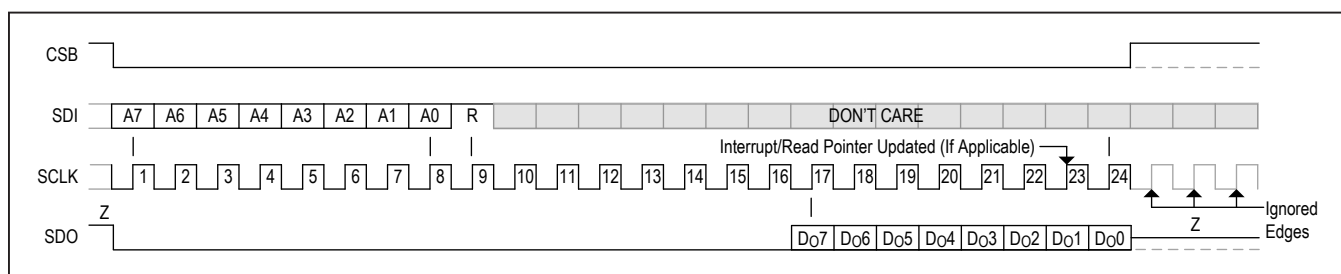


Figure 36. SPI Read Transaction

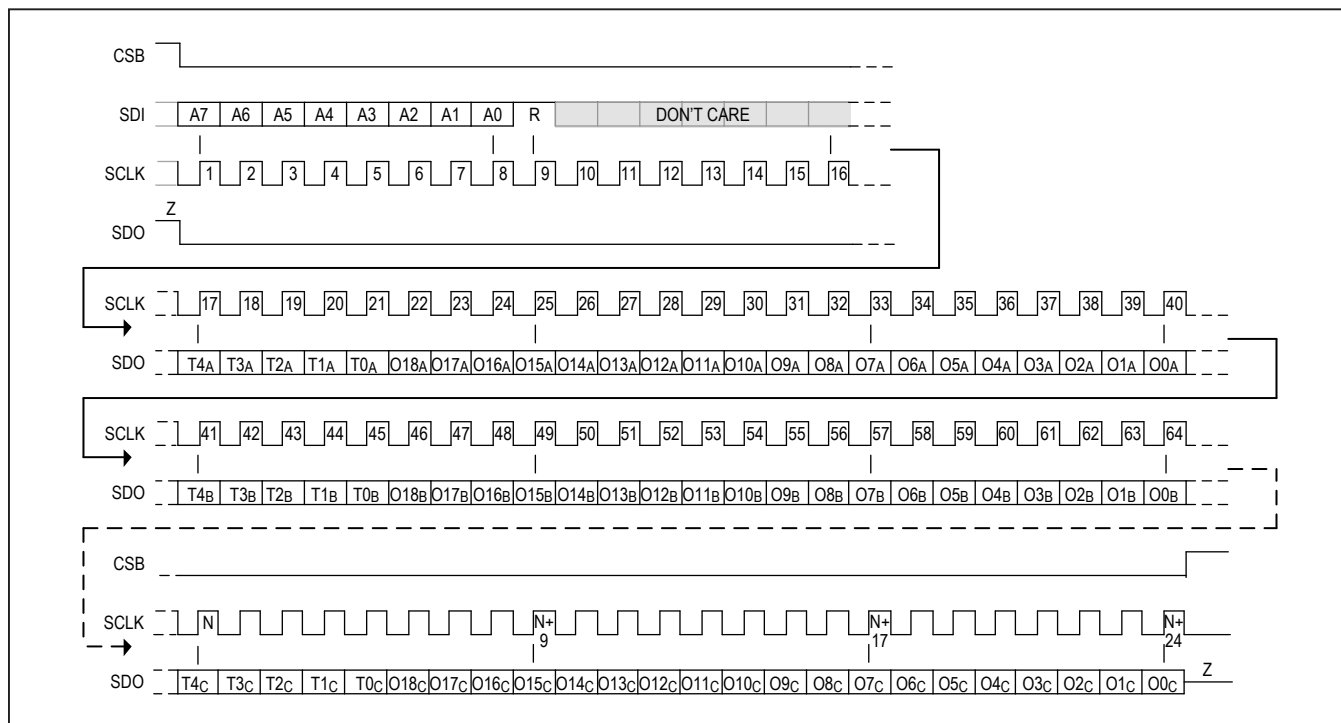


Figure 37. SPI FIFO Burst Mode Read Transaction

Register Map

| ADDRESS | NAME | MSB | | | | | | | LSB |
|-------------------|-------------------------------|----------------------|------------------|-------------------|---------------|-------------------|-----------------|---------------|-------------|
| Status | | | | | | | | | |
| 0x00 | Interrupt Status 1[7:0] | A_FULL | DATA_RDY | ALC_OVF | PROX_INT | LED_COMPB | DIE_TEMP_RDY | VDD_OOR | PWR_RDY |
| 0x01 | Interrupt Status 2[7:0] | – | – | – | – | – | – | – | SHA_DONE |
| 0x02 | Interrupt Enable 1[7:0] | A_FULL_EN | DATA_RDY_EN | ALC_OVF_EN | PROX_INT_EN | LED_COMPB_EN | DIE_TEMP_RDY_EN | VDD_OOR_EN | – |
| 0x03 | Interrupt Enable 2[7:0] | – | – | – | – | – | – | – | SHA_DONE_EN |
| FIFO | | | | | | | | | |
| 0x04 | FIFO Write Pointer[7:0] | – | FIFO_WR_PTR[6:0] | | | | | | |
| 0x05 | FIFO Read Pointer[7:0] | – | FIFO_RD_PTR[6:0] | | | | | | |
| 0x06 | Over Flow Counter[7:0] | – | OVF_COUNTER[6:0] | | | | | | |
| 0x07 | FIFO Data Counter[7:0] | FIFO_DATA_COUNT[7:0] | | | | | | | |
| 0x08 | FIFO Data Register[7:0] | FIFO_DATA[7:0] | | | | | | | |
| 0x09 | FIFO Configuration 1[7:0] | – | FIFO_A_FULL[6:0] | | | | | | |
| 0x0A | FIFO Configuration 2[7:0] | – | – | – | FLUSH_FIFO | FIFO_STAT_CLR | A_FULL_TYPE | FIFO_RO | – |
| System Control | | | | | | | | | |
| 0x0D | System Control[7:0] | – | – | – | – | SINGLE_PPG | LP_MODE | SHDN | RESET |
| PPG Configuration | | | | | | | | | |
| 0x10 | PPG Sync Control[7:0] | TIME_STAMP_EN | – | – | SW_FORCE_SYNC | GPIO_CTRL[3:0] | | | |
| 0x11 | PPG Configuration 1[7:0] | ALC_DISABLE | ADD_OFFSET | PPG2_ADC_RGE[1:0] | | PPG1_ADC_RGE[1:0] | | PPG_TINT[1:0] | |
| 0x12 | PPG Configuration 2[7:0] | PPG_SR[4:0] | | | | | SMP_AVE[2:0] | | |
| 0x13 | PPG Configuration 3[7:0] | LED_SETLNG[1:0] | | DIG_FILT_SEL | – | – | BURST_RATE[1:0] | | BURST_EN |
| 0x14 | Prox Interrupt Threshold[7:0] | PROX_INT_THRESH[7:0] | | | | | | | |
| 0x15 | Photo Diode Bias[7:0] | – | PDBIAS2[2:0] | | | – | PDBIAS1[2:0] | | |

Register Map (continued)

| ADDRESS | NAME | MSB | | | | | | LSB |
|-------------------------------------|------------------------------|------------------|----------|-------------------|------------|---------------------|--|---------------------------|
| PPG Picket Fence Detect and Replace | | | | | | | | |
| 0x16 | Picket Fence[7:0] | PF_ENABLE | PF_ORDER | IIR_TC[1:0] | | IIR_INIT_VALUE[1:0] | | THRESHOLD_SIGMA_MULT[1:0] |
| LED Sequence Control | | | | | | | | |
| 0x20 | LED Sequence Register 1[7:0] | LEDC2[3:0] | | | LEDC1[3:0] | | | |
| 0x21 | LED Sequence Register 2[7:0] | LEDC4[3:0] | | | LEDC3[3:0] | | | |
| 0x22 | LED Sequence Register 3[7:0] | LEDC6[3:0] | | | LEDC5[3:0] | | | |
| LED Pulse Amplitude | | | | | | | | |
| 0x23 | LED1 PA[7:0] | LED1_DRV[7:0] | | | | | | |
| 0x24 | LED2 PA[7:0] | LED2_DRV[7:0] | | | | | | |
| 0x25 | LED3 PA[7:0] | LED3_DRV[7:0] | | | | | | |
| 0x26 | LED4 PA[7:0] | LED4_DRV[7:0] | | | | | | |
| 0x27 | LED5 PA[7:0] | LED5_DRV[7:0] | | | | | | |
| 0x28 | LED6 PA[7:0] | LED6_DRV[7:0] | | | | | | |
| 0x29 | LED PILOT PA[7:0] | PILOT_PA[7:0] | | | | | | |
| 0x2A | LED Range 1[7:0] | – | – | LED3_RGE[1:0] | | LED2_RGE[1:0] | | LED1_RGE[1:0] |
| 0x2B | LED Range 2[7:0] | – | – | LED6_RGE[1:0] | | LED5_RGE[1:0] | | LED4_RGE[1:0] |
| PPG1_HI_RES_DAC | | | | | | | | |
| 0x2C | S1 HI RES DAC1[7:0] | S1_HRES_DAC1_OVR | – | S1_HRES_DAC1[5:0] | | | | |
| 0x2D | S2 HI RES DAC1[7:0] | S2_HRES_DAC1_OVR | – | S2_HRES_DAC1[5:0] | | | | |
| 0x2E | S3 HI RES DAC1[7:0] | S3_HRES_DAC1_OVR | – | S3_HRES_DAC1[5:0] | | | | |
| 0x2F | S4 HI RES DAC1[7:0] | S4_HRES_DAC1_OVR | – | S4_HRES_DAC1[5:0] | | | | |
| 0x30 | S5 HI RES DAC1[7:0] | S5_HRES_DAC1_OVR | – | S5_HRES_DAC1[5:0] | | | | |
| 0x31 | S6 HI RES DAC1[7:0] | S6_HRES_DAC1_OVR | – | S6_HRES_DAC1[5:0] | | | | |

Register Map (continued)

| ADDRESS | NAME | MSB | | | | | | | LSB |
|-----------------|------------------------------------|------------------|---|-------------------|---|----------------|---|-----------|-----------|
| PPG2_HI_RES_DAC | | | | | | | | | |
| 0x32 | S1 HI RES DAC2[7:0] | S1_HRES_DAC2_OVR | — | S1_HRES_DAC2[5:0] | | | | | |
| 0x33 | S2 HI RES DAC2[7:0] | S2_HRES_DAC2_OVR | — | S2_HRES_DAC2[5:0] | | | | | |
| 0x34 | S3 HI RES DAC2[7:0] | S3_HRES_DAC2_OVR | — | S3_HRES_DAC2[5:0] | | | | | |
| 0x35 | S4 HI RES DAC2[7:0] | S1_HRES_DAC2_OVR | — | S4_HRES_DAC2[5:0] | | | | | |
| 0x36 | S5 HI RES DAC2[7:0] | S2_HRES_DAC2_OVR | — | S5_HRES_DAC2[5:0] | | | | | |
| 0x37 | S6 HI RES DAC2[7:0] | S3_HRES_DAC2_OVR | — | S6_HRES_DAC2[5:0] | | | | | |
| Die Temperature | | | | | | | | | |
| 0x40 | Die Temperature Configuration[7:0] | — | — | — | — | — | — | — | TEMP_EN |
| 0x41 | Die Temperature Integer[7:0] | TEMP_INT[7:0] | | | | | | | |
| 0x42 | Die Temperature Fraction[7:0] | — | — | — | — | TEMP_FRAC[3:0] | | | |
| SHA256 | | | | | | | | | |
| 0xF0 | SHA Command[7:0] | SHA_CMD[7:0] | | | | | | | |
| 0xF1 | SHA Configuration[7:0] | — | — | — | — | — | — | SHA_EN | SHA_START |
| Memory | | | | | | | | | |
| 0xF2 | Memory Control[7:0] | — | — | — | — | — | — | MEM_WR_EN | BANK_SEL |
| 0xF3 | Memory Index[7:0] | MEM_IDX[7:0] | | | | | | | |
| 0xF4 | Memory Data[7:0] | MEM_DATA[7:0] | | | | | | | |
| Part ID | | | | | | | | | |
| 0xFF | Part ID[7:0] | PART_ID[7:0] | | | | | | | |

Interrupt Status 1 (0x00)

| BIT | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------|-----------|-----------|-----------|-----------|-----------|--------------|-----------|-----------|
| Field | A_FULL | DATA_RDY | ALC_OVF | PROX_INT | LED_COMPB | DIE_TEMP_RDY | VDD_OOR | PWR_RDY |
| Reset | 0x0 | 0x0 | 0x0 | 0x0 | 0x0 | 0x0 | 0x0 | 0x0 |
| Access Type | Read Only | Read Only | Read Only | Read Only | Read Only | Read Only | Read Only | Read Only |

A_FULL

This is a read-only bit. This bit is cleared when the Interrupt Status 1 Register is read. It is also cleared when FIFO_DATA register is read, if FIFO_STAT_CLR = 1.

| VALUE | ENUMERATION | DECODE |
|-------|-------------|--|
| 0 | OFF | Normal Operation |
| 1 | ON | Indicates that the FIFO buffer will overflow the threshold set by FIFO_A_FULL[6:0] on the next sample. |

DATA_RDY

This is a read-only bit and it is cleared by reading the Interrupt Status 1 register (0x00). It is also cleared by reading the FIFO_DATA register if FIFO_STAT_CLR = 1.

| VALUE | ENUMERATION | DECODE |
|-------|-------------|---|
| 0 | OFF | Normal Operation |
| 1 | ON | This interrupt triggers when there is a new data in the FIFO. |

ALC_OVF

This is a read-only bit. The interrupt is cleared by reading the Interrupt Status 1 register (0x00).

| VALUE | ENUMERATION | DECODE |
|-------|-------------|--|
| 0 | OFF | Normal Operation |
| 1 | ON | This interrupt triggers when the ambient light cancellation function of the photodiode has reached its maximum limit due to overflow, and therefore, ambient light is affecting the output of the ADC. |

PROX_INT

| VALUE | ENUMERATION | DECODE |
|-------|-------------|---|
| 0 | OFF | Normal Operation |
| 1 | ON | Indicates that the ADC reading of the LED configured in LEDC1 has crossed the proximity threshold. If PROX_INT_EN is 0, then the prox mode is disabled and the exposure sequence configured in LED Sequence Control Registers begins immediately. This bit is cleared when the Interrupt Status 1 Register is read. |

LED_COMPB

LED is not compliant. At the end of each sample, if the LED driver is not compliant, LED_COMPB interrupt is asserted if LED_COMPB_EN is set to 1. It is a read-only bit and is cleared when the status register is read.

| VALUE | ENUMERATION | DECODE |
|-------|---------------|-----------------------------|
| 0 | COMPLIANT | LED driver is compliant |
| 1 | NOT_COMPLIANT | LED driver is not compliant |

DIE_TEMP_RDY

This is a read-only bit and is automatically cleared when the temperature data is read, or when the Interrupt Status 1 Register is read.

| VALUE | ENUMERATION | DECODE |
|-------|-------------|---|
| 0 | OFF | Normal Operation |
| 1 | ON | Indicates that the TEMP ADC has finished it's current conversion. |

VDD_OOR

This is a read-only bit. It is automatically cleared when the Interrupt Status 1 register is read.

The detection circuitry has a 10ms delay time, and will continue to trigger as long as the VDD_DIG is out of range.

| VALUE | ENUMERATION | DECODE |
|-------|-------------|--|
| 0 | OFF | Normal operation |
| 1 | ON | Indicates that VDD_DIG is greater than 2.05V or less than 1.65V. |

PWR_RDY

This is a read-only bit and indicates that VDD_ANA had gone below the UVLO threshold. This bit is not triggered by a soft reset. This bit is cleared when either Interrupt Status 1 Register is read, or by setting SHDN bit to 1.

| VALUE | ENUMERATION | DECODE |
|-------|-------------|---|
| 0 | OFF | Normal Operation |
| 1 | ON | Indicates that VDD_ANA went below the UVLO threshold. |

Interrupt Status 2 (0x01)

| BIT | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------|---|---|---|---|---|---|---|-----------|
| Field | – | – | – | – | – | – | – | SHA_DONE |
| Reset | – | – | – | – | – | – | – | 0x0 |
| Access Type | – | – | – | – | – | – | – | Read Only |

SHA_DONE

SHA256 Authentication Done status bit is set to 1 when the authentication algorithm completes. This is a read-only bit and gets cleared when the Status Register is read.

| VALUE | ENUMERATION | DECODE |
|-------|-------------|-----------------------------|
| 0x0 | | SHA Authentication not done |
| 0x1 | | SHA Authentication done |

Interrupt Enable 1 (0x02)

| BIT | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------|-------------|-------------|-------------|-------------|--------------|-----------------|-------------|---|
| Field | A_FULL_EN | DATA_RDY_EN | ALC_OVF_EN | PROX_INT_EN | LED_COMPB_EN | DIE_TEMP_RDY_EN | VDD_OOR_EN | – |
| Reset | 0x0 | 0x0 | 0x0 | 0x0 | 0x0 | 0x0 | 0x0 | – |
| Access Type | Write, Read | Write, Read | Write, Read | Write, Read | Write, Read | Write, Read | Write, Read | – |

A_FULL_EN

| VALUE | ENUMERATION | DECODE |
|-------|-------------|------------------------------|
| 0 | OFF | A_FULL interrupt is disabled |
| 1 | ON | A_FULL interrupt is enabled |

DATA_RDY_EN

| VALUE | ENUMERATION | DECODE |
|-------|-------------|--------------------------------|
| 0 | OFF | DATA_RDY interrupt is disabled |
| 1 | ON | DATA_RDY interrupt is enabled. |

ALC_OVF_EN

| VALUE | ENUMERATION | DECODE |
|-------|-------------|-------------------------------|
| 0 | OFF | ALC_OVF interrupt is disabled |
| 1 | ON | ALC_OVF interrupt is enabled |

PROX_INT_EN

When PROX_INT_EN is enabled, the exposure programmed in the LEDC1 Sequence Register is used for proximity detection. If the ADC reading for this exposure is below 2048 times the threshold programmed in PROX_INT_THRESH register, the device is in proximity mode. Otherwise, it is in normal mode.

When the device is in proximity mode, the sample rate used is 8Hz, and the device starts data acquisition in pilot mode, using only one exposure of the LED programmed in LEDC1 register, and the LED current programmed in PILOT_PA register.

When the device is in normal mode, the sample rate used is as defined under PPG_SR register, and the device starts data acquisition in normal mode, using all the exposures programmed in the LED Sequence registers and appropriate LED currents.

PROX_INT interrupt is asserted when the device enters proximity mode or normal mode if PROX_INT_EN is programmed to 1.

| VALUE | ENUMERATION | DECODE |
|-------|-------------|--|
| 0 | OFF | Proximity mode and PROX_INT interrupt are disabled |
| 1 | ON | Proximity mode and PROX_INT interrupt are enabled |

LED_COMPB_EN

| VALUE | ENUMERATION | DECODE |
|-------|-------------|---------------------------------|
| 0 | DISABLE | LED_COMPB interrupt is disabled |
| 1 | ENABLE | LED_COMPB interrupt is enabled |

DIE_TEMP_RDY_EN

| VALUE | ENUMERATION | DECODE |
|-------|-------------|------------------------------------|
| 0 | OFF | DIE_TEMP_RDY interrupt is disabled |
| 1 | ON | DIE_TEMP_RDY interrupt is enabled |

VDD_OOR_EN

| VALUE | ENUMERATION | DECODE |
|-------|-------------|--------------------------------|
| 0 | OFF | Disables the VDD_OVR interrupt |
| 1 | ON | Enables the VDD_OVR interrupt |

Interrupt Enable 2 (0x03)

| BIT | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------|---|---|---|---|---|---|---|-------------|
| Field | – | – | – | – | – | – | – | SHA_DONE_EN |
| Reset | – | – | – | – | – | – | – | 0x0 |
| Access Type | – | – | – | – | – | – | – | Write, Read |

SHA_DONE_EN

Enable SHA_DONE Interrupt

| VALUE | ENUMERATION | DECODE |
|-------|-------------|-----------------------------|
| 0x0 | | SHA_DONE interrupt disabled |
| 0x1 | | SHA_DONE interrupt enabled |

FIFO Write Pointer (0x04)

| BIT | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------|---|------------------|---|---|---|---|---|---|
| Field | – | FIFO_WR_PTR[6:0] | | | | | | |
| Reset | – | 0x0 | | | | | | |
| Access Type | – | Read Only | | | | | | |

FIFO_WR_PTR

This points to the location where the next sample will be written. This pointer advances for each sample pushed on to the circular FIFO.

See [FIFO Configuration](#) for details.

FIFO Read Pointer (0x05)

| BIT | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------|---|------------------|---|---|---|---|---|---|
| Field | – | FIFO_RD_PTR[6:0] | | | | | | |
| Reset | – | 0x0 | | | | | | |
| Access Type | – | Write, Read | | | | | | |

FIFO_RD_PTR

The FIFO Read Pointer points to the location from where the processor gets the next sample from the FIFO via the serial interface. This advances each time a sample is popped from the circular FIFO.

The processor may also write to this pointer after reading the samples. This allows rereading (or retrying) samples from the FIFO. However, writing to FIFO_RD_PTR may have adverse effects if it results in the FIFO being almost full.

Refer to [FIFO Configuration](#) for details.

Overflow Counter (0x06)

| BIT | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------|---|------------------|---|---|---|---|---|---|
| Field | – | OVF_COUNTER[6:0] | | | | | | |
| Reset | – | 0x0 | | | | | | |
| Access Type | – | Read Only | | | | | | |

OVF_COUNTER

When FIFO is full, any new samples will result in new or old samples getting lost, depending on FIFO_RO. OVF_COUNTER counts the number of samples lost. It saturates at 0x7F.

Refer to [FIFO Configuration](#) for details.

FIFO Data Counter (0x07)

| BIT | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------|----------------------|---|---|---|---|---|---|---|
| Field | FIFO_DATA_COUNT[7:0] | | | | | | | |
| Reset | 0x0 | | | | | | | |
| Access Type | Read Only | | | | | | | |

FIFO_DATA_COUNT

This is a read-only register that holds the number of items available in the FIFO for the host to read. This increments when a new item is pushed to the FIFO and decrements when the host reads an item from the FIFO.

Refer to [FIFO Configuration](#) for details.

FIFO Data Register (0x08)

| BIT | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------|----------------|---|---|---|---|---|---|---|
| Field | FIFO_DATA[7:0] | | | | | | | |
| Reset | 0x0 | | | | | | | |
| Access Type | Read Only | | | | | | | |

FIFO_DATA

This is a read-only register and is used to get data from the FIFO. Refer to [FIFO Configuration](#) for details.

FIFO Configuration 1 (0x09)

| BIT | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------|---|------------------|---|---|---|---|---|---|
| Field | – | FIFO_A_FULL[6:0] | | | | | | |
| Reset | – | 0x3F | | | | | | |
| Access Type | – | Write, Read | | | | | | |

FIFO_A_FULL

These bits indicate how many new samples can be written to the FIFO before the interrupt is asserted. For example, if set to 0xF, the interrupt triggers when there are 15 empty spaces left (113 entries), and so on.

Refer to [FIFO Configuration](#) for details.

| FIFO_A_FULL | FREE SPACE BEFORE INTERRUPT | # OF SAMPLES IN FIFO |
|-------------|-----------------------------|----------------------|
| 0 | 0 | 128 |
| 1 | 1 | 127 |
| 2 | 2 | 126 |
| 3 | 3 | 125 |
| ---- | ---- | ---- |
| 126 | 126 | 2 |
| 127 | 127 | 1 |

FIFO Configuration 2 (0x0A)

| BIT | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------|---|---|---|-------------|---------------|-------------|-------------|---|
| Field | – | – | – | FLUSH_FIF0 | FIFO_STAT_CLR | A_FULL_TYPE | FIFO_RO | – |
| Reset | – | – | – | 0x0 | 0x0 | 0x0 | 0x0 | – |
| Access Type | – | – | – | Write, Read | Write, Read | Write, Read | Write, Read | – |

FLUSH_FIFO

When this bit is set to '1', the FIFO gets flushed, FIFO_WR_PTR and FIFO_RD_PTR are reset to zero and FIFO_DATA_COUNT becomes 0. The contents of the FIFO are lost.

FIFO_FLUSH is a self-clearing bit.

Refer to [FIFO Configuration](#) for details.

FIFO_STAT_CLR

This defines whether the A_FULL interrupt should get cleared by FIFO_DATA register read.

Refer to [FIFO Configuration](#) for details.

| VALUE | ENUMERATION | DECODE |
|-------|---------------|---|
| 0 | RD_DATA_NOCLR | A_FULL and DATA_RDY interrupts do not get cleared by FIFO_DATA register read. They get cleared by status register read. |
| 1 | RD_DATA_CLR | A_FULL and DATA_RDY interrupts get cleared by FIFO_DATA register read or status register read. |

A_FULL_TYPE

This defines the behavior of the A_FULL interrupt.

| VALUE | ENUMERATION | DECODE |
|-------|-------------|--|
| 0 | AFULL_RPT | A_FULL interrupt gets asserted when the a_full condition is detected. It is cleared by status register read, but re-asserts for every sample if the a_full condition persists. |
| 1 | AFULL_ONCE | A_FULL interrupt gets asserted only when the a_full condition is detected. The interrupt gets cleared on status register read, and does not re-assert for every sample until a new A_FULL condition is detected. |

FIFO_RO

Push enable when FIFO is full:

This bit controls the behavior of the FIFO when the FIFO becomes completely filled with data.

Push to FIFO is enabled when FIFO is full if FIFO_RO = 1 and old samples are lost. Both FIFO_WR_PTR increments for each sample after the FIFO is full. FIFO_RD_PTR also increments for each sample pushed to the FIFO.

Push to FIFO is disabled when FIFO is full if FIFO_RO = 0 and new samples are lost. FIFO_WR_PTR does not increment for each sample after the FIFO is full.

When the device is in PROX mode, push to FIFO is enabled independent of FIFO_RO setting.

Refer to [FIFO Configuration](#) for details.

| VALUE | ENUMERATION | DECODE |
|-------|-------------|--|
| 0 | OFF | The FIFO stops on full. |
| 1 | ON | The FIFO automatically rolls over on full. |

System Control (0x0D)

| BIT | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------|---|---|---|---|-------------|-------------|-------------|-------------|
| Field | – | – | – | – | SINGLE_PPG | LP_MODE | SHDN | RESET |
| Reset | – | – | – | – | 0x0 | 0x0 | 0x0 | 0x0 |
| Access Type | – | – | – | – | Write, Read | Write, Read | Write, Read | Write, Read |

SINGLE_PPG

In signal PP devices, this bit is ignored. In dual PPG devices, if this bit is 0, use two PPG channels; otherwise, use only PPG1 channel.

| VALUE | ENUMERATION | DECODE |
|-------|-------------|-------------------------------|
| 0x0 | DUAL_PPG | Both PPG channels are enabled |
| 0x1 | SINGLE_PPG | Only PPG1 channel is enabled |

LP_MODE

In low power mode, the sensor can be dynamically powered down between samples to conserve power. This dynamic power down mode option only supports samples rates of 256sps and below.

| VALUE | ENUMERATION | DECODE |
|-------|-------------|---|
| 0 | OFF | Dynamic power down is disabled. |
| 1 | ON | Dynamic power down is enabled. The device automatically enters low power mode between samples for samples rates 256sps and below. This mode is not available for higher sample rates. |

SHDN

The part can be put into a power-save mode by setting this bit to one. While in power-save mode, all configuration registers retain their values, and write/read operations function as normal. All interrupts are cleared to zero in this mode.

| VALUE | ENUMERATION | DECODE |
|-------|-------------|--|
| 0 | OFF | The part is in normal operation. No action taken. |
| 1 | ON | The part can be put into a power-save mode by writing a '1' to this bit. While in this mode all configuration registers remain accessible and retain their data. ADC conversion data contained in the registers are previous values. Writeable registers also remain accessible in shutdown. All interrupts are cleared. In this mode the oscillator is shutdown and the part draws minimum current. If this bit is asserted during an active conversion then the conversion is aborted. |

RESET

When this bit is set, the part undergoes a forced power-on-reset sequence. All configuration, threshold, and data registers including distributed registers are reset to their power-on-state. This bit then automatically becomes '0' after the reset sequence is completed.

| VALUE | ENUMERATION | DECODE |
|-------|-------------|--|
| 0 | OFF | The part is in normal operation. No action taken. |
| 1 | ON | The part undergoes a forced power-on-reset sequence. All configuration, threshold and data registers including distributed registers are reset to their power-on-state. This bit then automatically becomes '0' after the reset sequence is completed. |

PPG Sync Control (0x10)

| BIT | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------|---------------|---|---|---------------|----------------|---|---|---|
| Field | TIME_STAMP_EN | – | – | SW_FORCE_SYNC | GPIO_CTRL[3:0] | | | |
| Reset | 0x0 | – | – | 0x0 | 0x0 | | | |
| Access Type | Write, Read | – | – | Write, Read | Write, Read | | | |

TIME_STAMP_EN

Enable pushing TIME_STAMP to FIFO. Refer to FIFO Configuration for details.

| VALUE | ENUMERATION | DECODE |
|-------|-------------|--|
| 0x0 | DISABLE | TIME_STAMP is not pushed to FIFO |
| 0x1 | ENABLE | TIME_STAMP is pushed to FIFO for a block of eight samples. |

SW_FORCE_SYNC

Writing a 1 to this bit, aborts current sample and starts a new sample. This is a self clearing bit.

GPIO_CTRL

The table below shows how the two GPIO ports are control for different modes of operation.

When two devices are configured to work as controller-target device pairs, they have to be configured identical for the following configuration register fields:

- PPG_SR
- PPG_TINT
- SMP_AVE
- TIME_STAMP_EN
- FIFO_A_FULL
- FIFO_ROLLS_ON_FULL

Number of LED Sequence Registers (LEDC1 to LEDC6) programmed should be same in both the devices. In Exposure Trigger mode, if Ambient is programmed in one of the registers, it needs to be in the same LEDCx register in both the devices.

GPIO_CTRL register for both the devices should be programmed to be either Sample Trigger or Exposure Trigger.

It is also important to configure the Target first and then the Controller.

DATA_RDY or A_FULL interrupt should be enabled only on the Controller. When interrupt is asserted read the Controller first and then the Target. Read same number of items from both devices.

Refer to GPIO Configuration for details.

| GPIO_CTRL [3:0] | GPIO1 FUNCTION | GPIO2 FUNCTION | COMMENT |
|-----------------|-------------------------|--------------------------------------|---|
| 0000 | Tristate or Mux Control | Disabled | GPIO1 will be active if any of the LEDCn[3:0] states A, B, or C are enabled in the exposure sequence. In this case, GPIO1 will be low during exposures on LED4, LED5, or LED6, otherwise it will be high. If LEDCn[3:0] state A, B, or C is not enabled in the exposure sequence, GPIO1 will be tristate unless externally pulled up. GPIO2 is disabled. Sample and exposure timing is controlled by the internal 32768Hz oscillator. |
| 0001 | Tristate or Mux Control | Input 32768Hz or 32000Hz Clock Input | GPIO1 will be active if any of the LEDCn[3:0] states A, B, or C are enabled in the exposure sequence. In this case, GPIO1 will be low during exposures on LED4, LED5 or LED6, otherwise it will be high. If LEDCn[3:0] state A, B, or C is not enabled in the exposure sequence, GPIO1 will be tristate unless externally pulled up. GPIO2 is an input 32768/32000Hz. Sample and exposure timing is controlled by GPIO2 clock input. |

| GPIO_CTRL [3:0] | GPIO1 FUNCTION | GPIO2 FUNCTION | COMMENT |
|----------------------------|---|---|---|
| 0010 | Input Sample Trigger | Tristate or Mux Control | GPIO1 is defined as a sample trigger input (Target). This input can come from an external source or from another MAX86140/MAX86141 in controller sample mode. GPIO2 will be active if any of the LEDCn[3:0] states A, B, or C are enabled in the exposure sequence. In this case, GPIO2 will be low during exposures on LED4, LED5, or LED6, otherwise it will be high. If LEDCn[3:0] state A, B, or C is not enabled in the exposure sequence, GPIO2 will be tristate unless externally pulled up. Exposure timing is controlled by internal oscillator. |
| 0011 | Input Sample Trigger | Input 32768Hz or 32000Hz Clock Input | GPIO1 is defined as a sample trigger input (Target). This input can come from an external source or from another MAX86140/MAX86141 in controller sample mode. GPIO2 is an input 32768/32000Hz clock input. Exposure timing is controlled by GPIO2 clock input. |
| 0100 | Active Output Controller Sample Output | Tristate or Mux Control | GPIO1 is defined as a controller sample output. The GPIO1 output can be used to trigger a second sensor. When used with a second MAX86140/MAX86141 set to target sample mode, the controller sample timing will drive target sample time. GPIO2 will be active if any of the LEDCn[3:0] states A, B, or C are enabled in the exposure sequence. In this case, GPIO2 will be low during exposures on LED4, LED5 or LED6, otherwise it will be high. If LEDCn[3:0] state A, B, or C is not enabled in the exposure sequence, GPIO2 will be tristate unless externally pulled up. Sample and exposure timing is controlled by internal oscillator. |
| 0101 | Active Output Controller Sample Output | Input 32768/ 32000Hz Clock Input | GPIO1 is defined as a controller sample output. The GPIO1 output can be used to trigger and second sensor. When used with a second MAX86140/MAX86141 set to target sample mode, the controller sample timing will drive target sample time. GPIO2 is an input 32768/32000Hz. Exposure timing is controlled by GPIO2 clock input. |
| 0110 | Input Exposure Trigger | Tristate or Mux Control | GPIO1 is defined as an exposure trigger input (Target). This input can come from an external source or from another MAX86140/MAX86141 in controller sample mode. Both sample and exposure timing is controlled by the GPIO1 input. GPIO2 will be active if any of the LEDCn[3:0] states A, B, or C are enabled in the exposure sequence. In this case, GPIO2 will be low during exposures on LED4, LED5, or LED6, otherwise it will be high. If LEDCn[3:0] state A, B, or C is not enabled in the exposure sequence, GPIO2 will be tristate unless externally pulled up. |
| 0111 | Active Output Controller Exposure Output | Tristate or Mux Control | GPIO1 is defined as a controller sample output. The GPIO1 output can be used to trigger and second sensor. When used with a second MAX86140/MAX86141 set to target exposure mode, the controller exposure timing will drive target exposure time. GPIO2 will be active if any of the LEDCn[3:0] states A, B, or C are enabled in the exposure sequence. In this case, GPIO2 will be low during exposures on LED4, LED5 or LED6, otherwise it will be high. If LEDCn[3:0] state A, B, or C is not enabled in the exposure sequence, GPIO2 will be tristate unless externally pulled up. Sample and exposure time is controlled internally |
| 1000 | Active Output Controller Exposure Output | Input 32768/ 32000Hz Clock Input | GPIO1 is defined as a controller sample output. The GPIO1 output can be used to trigger second sensor. When used with a second MAX86140/MAX86141 set to target exposure mode, the controller exposure timing will drive target exposure time. GPIO2 is an input 32768/32000Hz. Sample and exposure timing is controlled by GPIO2 clock input. |
| 1001 | Input HW_ FORCE_ SYNC | Input 32768Hz or 32000Hz Clock Input | GPIO1 is defined as a start of sample sync input. The falling edge of GPIO1 causes the present sample sequence to be terminated and reinitiated on the next rising edge of GPIO2 input. GPIO2 is an input 32768/32000Hz. Sample and exposure timing is controlled by GPIO2 clock input. |

PPG Configuration 1 (0x11)

| BIT | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------|-------------|-------------|-------------------|---|-------------------|---|---------------|---|
| Field | ALC_DISABLE | ADD_OFFSET | PPG2_ADC_RGE[1:0] | | PPG1_ADC_RGE[1:0] | | PPG_TINT[1:0] | |
| Reset | 0x0 | 0 | 0x0 | | 0x0 | | 0x3 | |
| Access Type | Write, Read | Write, Read | Write, Read | | Write, Read | | Write, Read | |

ALC_DISABLE

| VALUE | ENUMERATION | DECODE |
|-------|-------------|-----------------|
| 0 | OFF | ALC is enabled |
| 1 | ON | ALC is disabled |

ADD_OFFSET

ADD_OFFSET is an option designed for dark current measurement. By adding offset to the PPG Data would allow dark current measurement without clipping the signal below 0.

When ADD_OFFSET is set to 1, an offset is added to the PPG Data to be able to measure the dark current. The offset is 8192 counts if PPG_SR is programmed for single pulse mode. The offset is 4096 counts if PPG_SR is programmed for dual pulse mode.

PPG2_ADC_RGE

These bits set the ADC range of the SPO₂ sensor, as shown in the table below.

| PPG_ADC_RGE[1:0] | LSB (pA) | FULL SCALE (nA) |
|------------------|----------|-----------------|
| 00 | 78125 | 4096 |
| 01 | 15.625 | 8192 |
| 10 | 31.25 | 16384 |
| 11 | 62.5 | 32768 |

PPG1_ADC_RGE

These bits set the ADC range of the SPO₂ sensor, as shown in the table below.

| PPG_ADC_RGE[1:0] | LSB (pA) | FULL SCALE (nA) |
|------------------|----------|-----------------|
| 00 | 7,8125 | 4096 |
| 01 | 15.625 | 8192 |
| 10 | 31.25 | 16384 |
| 11 | 62.5 | 32768 |

PPG_TINT

These bits set the pulse width of the LED drivers and the integration time of PPG ADC as shown in the table below.

$$t_{PW} = t_{TINT} + t_{LED_SETLNG} + 0.5\mu s$$

| PPG_TINT[1:0] | t _{INT} , INTEGRATION TIME (μs) | RESOLUTION BITS |
|---------------|--|-----------------|
| 00 | 14.8 | 19 |
| 01 | 29.4 | 19 |
| 10 | 58.7 | 19 |
| 11 | 117.3 | 19 |

PPG Configuration 2 (0x12)

| BIT | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------|-------------|---|---|---|---|--------------|---|---|
| Field | PPG_SR[4:0] | | | | | SMP_AVE[2:0] | | |
| Reset | 0x11 | | | | | 0x0 | | |
| Access Type | Write, Read | | | | | Write, Read | | |

PPG_SR

These bits set the effective sampling rate of the PPG sensor as shown in the table below. The default on-chip sampling clock frequency is 32768Hz.

Note: If a sample rate is set that can not be supported by the selected pulse width and number of exposures per sample, then the highest available sample rate will be automatically set. The user can read back this register to confirm the sample rate.

| SAMPLING CLOCK FREQUENCY | 32768Hz | 32000Hz | |
|-----------------------------|--------------------|--------------------|----------------------|
| PPG_SR[4:0] | Samples per Second | Samples per Second | Pulses Per Sample, N |
| 0x00 | 24.995 | 24.409 | 1 |
| 0x01 | 50.027 | 48.855 | 1 |
| 0x02 | 84.021 | 82.051 | 1 |
| 0x03 | 99.902 | 97.561 | 1 |
| 0x04 | 199.805 | 195.122 | 1 |
| 0x05 | 399.610 | 390.244 | 1 |
| 0x06 | 24.995 | 24.409 | 2 |
| 0x07 | 50.027 | 48.855 | 2 |
| 0x08 | 84.021 | 82.051 | 2 |
| 0x09 | 99.902 | 97.561 | 2 |
| 0x0A | 8.000 | 7.8125 | 1 |
| 0x0B | 16.000 | 15.625 | 1 |
| 0x0C | 32.000 | 31.250 | 1 |
| 0x0D | 64.000 | 62.500 | 1 |
| 0x0E | 128.000 | 125.000 | 1 |
| 0x0F | 256.000 | 250.000 | 1 |
| 0x10 | 512.000 | 500.000 | 1 |
| 0x11 | 1024.000 | 1000.000 | 1 |
| 0x12 | 2048.000 | 2000.000 | 1 |
| 0x13 | 4096.000 | 4000.000 | 1 |
| 0x14-1F | Reserved | Reserved | Reserved |

Maximum Sample rates (sps) supported for all the Integration Time (PPG_TINT) and Number of Exposures:

| NUMBER OF EXPOSURE PER SAMPLE | PPG_TINT = 0 (14.8μs) | PPG_TINT = 1 (29.4μs) | PPG_TINT = 2 (58.7μs) | PPG_TINT = 3 (117.3μs) |
|-------------------------------|-----------------------|-----------------------|-----------------------|------------------------|
| 1 Exposure, N = 1 | 4096 | 2048 | 2048 | 1024 |
| 2 Exposures, N = 1 | 2048 | 1024 | 1024 | 512 |
| 3 Exposures, N = 1 | 1024 | 1024 | 512 | 512 |
| 4 Exposures, N = 1 | 1024 | 512 | 512 | 400 |
| 5 Exposures, N = 1 | 512 | 512 | 512 | 256 |
| 6 Exposures, N = 1 | 512 | 512 | 400 | 256 |
| 1 Exposure, N = 2 | 100 | 100 | 100 | 100 |
| 2 Exposures, N = 2 | 100 | 84 | 84 | 84 |
| 3 Exposures, N = 2 | 50 | 50 | 50 | 50 |
| 4 Exposures, N = 2 | 25 | 25 | 25 | 25 |
| 5 Exposures, N = 2 | 25 | 25 | 25 | 25 |
| 6 Exposures, N = 2 | 25 | 25 | 25 | 25 |

SMP_AVE

To reduce the amount of data throughput, adjacent samples (in each individual channel) can be averaged and decimated on the chip by setting this register.

These bits set the number of samples that are averaged on chip before being written to the FIFO.

| SMP_AVE[2:0] | SAMPLE AVERAGE |
|--------------|------------------|
| 000 | 1 (no averaging) |
| 001 | 2 |
| 010 | 4 |
| 011 | 8 |
| 100 | 16 |
| 101 | 32 |
| 110 | 64 |
| 111 | 128 |

When BURST_EN is 1, SMP_AVE defines the number of conversions per burst. Depending on the BURST_RATE programmed and the PPG_SR used, it may not be possible to accommodate some of SMP_AVE values. In that case, SMP_AVE will take the highest value that can be accommodated. If SMP_AVE = 0 cannot be accommodated, burst mode is disabled.

Note: PPG_SR itself depends on Number of conversions per sample (LEDC1 to LEDC6) and the LED Integration time (PPG_TINT).

The following table shows the maximum SMP_AVE allowed for various configurations of BURST_RATE and PPG_SR:

| PPG_SR USED | BURST_RATE = 0 (8Hz) | BURST_RATE = 1 (32Hz) | BURST_RATE = 2 (84Hz) | BURST_RATE = 3 (256Hz) |
|--------------------|----------------------|-----------------------|-----------------------|------------------------|
| 0 (25Hz, N = 1) | 1 | DIS | DIS | DIS |
| 1 (50Hz, N = 1) | 2 | 0 | DIS | DIS |
| 2 (84Hz, N = 1) | 3 | 1 | DIS | DIS |
| 3 (100Hz, N = 1) | 3 | 1 | DIS | DIS |
| 4 (200Hz, N = 1) | 4 | 2 | 0 | DIS |
| 5 (400Hz, N = 1) | 5 | 3 | 1 | DIS |
| 6 (25Hz, N = 2) | 1 | DIS | DIS | DIS |
| 7 (50Hz, N = 2) | 2 | 0 | DIS | DIS |
| 8 (84Hz, N = 2) | 3 | 1 | DIS | DIS |
| 9 (100Hz, N = 2) | 3 | 1 | DIS | DIS |
| A (8Hz, N = 1) | DIS | DIS | DIS | DIS |
| B (16Hz, N = 1) | 0 | DIS | DIS | DIS |
| C (32Hz, N = 1) | 1 | DIS | DIS | DIS |
| D (64Hz, N = 1) | 2 | 0 | DIS | DIS |
| E (128Hz, N = 1) | 3 | 1 | 0 | DIS |
| F (256Hz, N = 1) | 4 | 2 | 1 | DIS |
| 10 (512Hz, N = 1) | 5 | 3 | 2 | DIS |
| 11 (1024Hz, N = 1) | 6 | 4 | 3 | 0 |
| 12 (2048Hz, N = 1) | 7 | 5 | 4 | 1 |
| 13 (4096Hz, N = 1) | 7 | 6 | 5 | 2 |

PPG Configuration 3 (0x13)

| BIT | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------|-----------------|---|--------------|---|---|-----------------|---|-------------|
| Field | LED_SETLNG[1:0] | | DIG_FILT_SEL | – | – | BURST_RATE[1:0] | | BURST_EN |
| Reset | 0x1 | | 0x0 | – | – | 0x0 | | 0x0 |
| Access Type | Write, Read | | Write, Read | – | – | Write, Read | | Write, Read |

LED_SETLNG

Delay from rising-edge of LED to start of ADC integration. This allows for the LED current to settle before the start of ADC integration.

| LED_SETLNG[1:0] | DELAY (μs) |
|-----------------|---------------|
| 00 | 4.0 |
| 01 | 6.0 (default) |
| 10 | 8.0 |
| 11 | 12.0 |

DIG_FILT_SEL

Select digital filter type.

| VALUE | DECODE |
|-------|---------|
| 0x0 | Use CDM |
| 0x1 | Use FDM |

BURST_RATE

| BURST_RATE[1:0] | FREQUENCY OF BURST (Hz) |
|-----------------|-------------------------|
| 00 | 8 |
| 01 | 32 |
| 10 | 84 |
| 11 | 256 |

BURST_EN

When Burst Mode is disabled, PPG data conversions are continuous at the sample rate defined by PPG_SR register. When Burst mode is enabled, a burst of PPG data conversions occur at the sample rate defined by PPG_SR register. Number of conversion in the burst is defined by the SMP_AVE register. Average data from the burst of data conversions is pushed to the FIFO. The burst repeats at the rate defined in BURST_RATE[2:0] register. If the number of conversions cannot be accommodated, the device will use the next highest number of conversions.

If the effective PPG_SR is too slow to accommodate the burst rate programmed, BURST_EN is automatically set to 0, and the device runs in continuous mode.

Note: Each data conversion cycle is a sequence of conversions defined in the LEDC1 to LEDC6 registers.

| VALUE | ENUMERATION | DECODE |
|-------|-------------|-------------------------------|
| 0x0 | | Disable Burst Conversion mode |
| 0x1 | | Enable Burst Conversion Mode |

Prox Interrupt Threshold (0x14)

| BIT | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------|----------------------|---|---|---|---|---|---|---|
| Field | PROX_INT_THRESH[7:0] | | | | | | | |
| Reset | 0x00 | | | | | | | |
| Access Type | Write, Read | | | | | | | |

PROX_INT_THRESH

This register sets the LED1 ADC count that will trigger the transition between proximity mode and normal mode. The threshold is defined as the 8 MSB bits of the ADC count. For example, if PROX_INT_THRESH[7:0] = 0x01, then an ADC value of 2048 (decimal) or higher triggers the PROX interrupt. If PROX_INT_THRESH[7:0] = 0xFF, then only a saturated ADC triggers the interrupt.

See the [Proximity Mode Function](#) section in the detailed description for more details on the operation of proximity mode.

Photo Diode Bias (0x15)

| BIT | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------|---|--------------|---|---|---|--------------|---|---|
| Field | – | PDBIAS2[2:0] | | | – | PDBIAS1[2:0] | | |
| Reset | – | 0x0 | | | – | 0x0 | | |
| Access Type | – | Write, Read | | | – | Write, Read | | |

PDBIAS2

See [Photodiode Biasing](#) for more information.

| PDBIAS2[2:0] | PHOTODIODE CAPACITANCE |
|------------------|------------------------|
| 001 | 0pF to 65pF |
| 101 | 65pF to 130pF |
| 110 | 130pF to 260pF |
| 111 | 260pF to 520pF |
| All other values | Not recommended |

PDBIAS1

See [Photodiode Biasing](#) for more information.

| PDBIAS1[2:0] | PHOTODIODE CAPACITANCE |
|------------------|------------------------|
| 001 | 0pF to 65pF |
| 101 | 65pF to 130pF |
| 110 | 130pF to 260pF |
| 111 | 260pF to 520pF |
| All other values | Not recommended |

Picket Fence (0x16)

| BIT | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------|-------------|-------------|-------------|---|---------------------|---|---------------------------|---|
| Field | PF_ENABLE | PF_ORDER | IIR_TC[1:0] | | IIR_INIT_VALUE[1:0] | | THRESHOLD_SIGMA_MULT[1:0] | |
| Reset | 0x0 | 0x1 | 0x00 | | 0x00 | | 0x00 | |
| Access Type | Write, Read | Write, Read | Write, Read | | Write, Read | | Write, Read | |

PF_ENABLE

Refer to [Picket Fence Detect-and-Replace Function](#) for details.

PF_ENABLE set to 1 enabled the picket fence detect-and-replace function.

| VALUE | ENUMERATION | DECODE |
|-------|-------------|---------------------------|
| 0 | OFF | Disable (default) |
| 1 | ON | Enable Detect and Replace |

PF_ORDER

PF_ORDER determines which prediction method is used: the last sample or a linear fit to the previous four samples.

Refer to [Picket Fence Detect-and-Replace Function](#) for details.

| VALUE | ENUMERATION | DECODE |
|-------|-------------|---|
| 0 | OFF | Last Sample (1 point) |
| 1 | ON | Fit 4 points to a line for prediction (default) |

IIR_TC

IIR_TC determines the IIR filter bandwidth where the lowest setting has the narrowest bandwidth of a first-order filter.

Refer to [Picket Fence Detect-and-Replace Function](#) for details.

| IIR_TC[1:0] | COEFFICIENT | SAMPLES TO 90% |
|-------------|-------------|----------------|
| 00 | 1/64 | 146 |
| 01 | 1/32 | 72 |
| 10 | 1/16 | 35 |
| 11 | 1/8 | 17 |

IIR_INIT_VALUE

This IIR filter estimates the true standard deviation between the actual and predicted sample and tracks the ADC Range setting.

Refer to [Picket Fence Detect-and-Replace Function](#) for details.

| IIR_INIT_VALUE[1:0] | CODE |
|---------------------|------|
| 00 | 64 |
| 01 | 48 |
| 10 | 32 |
| 11 | 24 |

THRESHOLD_SIGMA_MULT

GAIN resulting from the THRESHOLD_SIGMA_MULT setting determines the number of standard deviations of the delta between the actual and predicted sample beyond which a picket-fence event is triggered.

Refer to [Picket Fence Detect-and-Replace Function](#) for details.

| THRESHOLD_SIGMA_MULT[1:0] | GAIN |
|---------------------------|------|
| 00 | 4 |
| 01 | 8 |
| 10 | 16 |
| 11 | 32 |

LED Sequence Register 1 (0x20)

| BIT | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------|-------------|---|---|---|-------------|---|---|---|
| Field | LEDC2[3:0] | | | | LEDC1[3:0] | | | |
| Reset | 0x0 | | | | 0x0 | | | |
| Access Type | Write, Read | | | | Write, Read | | | |

LEDC2

These bits set the data type for LED Sequence 2 of the FIFO.

See [FIFO Configuration](#) for more information.

LEDC1

These bits set the data type for LED Sequence 1 of the FIFO.

See [FIFO Configuration](#) for more information.

LED Sequence Register 2 (0x21)

| BIT | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------|-------------|---|---|---|-------------|---|---|---|
| Field | LEDC4[3:0] | | | | LEDC3[3:0] | | | |
| Reset | 0x0 | | | | 0x0 | | | |
| Access Type | Write, Read | | | | Write, Read | | | |

LEDC4

These bits set the data type for LED Sequence 4 of the FIFO.

See [FIFO Configuration](#) for more information.

LEDC3

These bits set the data type for LED Sequence 3 of the FIFO.

See [FIFO Configuration](#) for more information.

LED Sequence Register 3 (0x22)

| BIT | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------|-------------|---|---|---|-------------|---|---|---|
| Field | LEDC6[3:0] | | | | LEDC5[3:0] | | | |
| Reset | 0x0 | | | | 0x0 | | | |
| Access Type | Write, Read | | | | Write, Read | | | |

LEDC6

These bits set the data type for LED Sequence 6 of the FIFO.

See [FIFO Configuration](#) for more information.

LEDC5

These bits set the data type for LED Sequence 5 of the FIFO.

See [FIFO Configuration](#) for more information.

LED1 PA (0x23)

| BIT | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------|---------------|---|---|---|---|---|---|---|
| Field | LED1_DRV[7:0] | | | | | | | |
| Reset | 0x00 | | | | | | | |
| Access Type | Write, Read | | | | | | | |

LED1_DRV

These bits set the nominal drive current of LED 1, as shown in the table below.

| LEDx_RGE[1:0] | 00 | 01 | 10 | 11 |
|---------------|------------------|------------------|------------------|------------------|
| LEDx_DRV[7:0] | LED Current (mA) | LED Current (mA) | LED Current (mA) | LED Current (mA) |
| 00000000 | 0.00 | 0.00 | 0.00 | 0.00 |
| 00000001 | 0.12 | 0.24 | 0.36 | 0.48 |
| 00000010 | 0.24 | 0.48 | 0.73 | 0.97 |
| 00000011 | 0.36 | 0.73 | 1.09 | 1.45 |
| | | | | |
| 11111100 | 30.6 | 61.3 | 91.9 | 122.5 |
| 11111101 | 30.8 | 61.5 | 92.3 | 123.0 |
| 11111110 | 30.9 | 61.8 | 92.6 | 123.5 |
| 11111111 | 31.0 | 62.0 | 93.0 | 124.0 |
| LSB | 0.12 | 0.24 | 0.36 | 0.48 |

LED2 PA (0x24)

| BIT | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------|---------------|---|---|---|---|---|---|---|
| Field | LED2_DRV[7:0] | | | | | | | |
| Reset | 0x00 | | | | | | | |
| Access Type | Write, Read | | | | | | | |

LED2_DRV

These bits set the nominal drive current of LED 2. See *LED1_DRV* for description.

LED3 PA (0x25)

| BIT | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------|---------------|---|---|---|---|---|---|---|
| Field | LED3_DRV[7:0] | | | | | | | |
| Reset | 0x00 | | | | | | | |
| Access Type | Write, Read | | | | | | | |

LED3_DRV

These bits set the nominal drive current of LED 2. See *LED1_DRV* for description.

LED4 PA (0x26)

| BIT | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------|---------------|---|---|---|---|---|---|---|
| Field | LED4_DRV[7:0] | | | | | | | |
| Reset | 0x00 | | | | | | | |
| Access Type | Write, Read | | | | | | | |

LED4_DRV

These bits set the nominal drive current of LED 4. See *LED1_DRV* for description.

LED5 PA (0x27)

| BIT | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------|---------------|---|---|---|---|---|---|---|
| Field | LED5_DRV[7:0] | | | | | | | |
| Reset | 0x00 | | | | | | | |
| Access Type | Write, Read | | | | | | | |

LED5_DRV

These bits set the nominal drive current of LED 5. See *LED1_DRV* for description.

LED6 PA (0x28)

| BIT | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------|---------------|---|---|---|---|---|---|---|
| Field | LED6_DRV[7:0] | | | | | | | |
| Reset | 0x00 | | | | | | | |
| Access Type | Write, Read | | | | | | | |

LED6_DRV

These bits set the nominal drive current of LED 6. See *LED1_DRV* for description.

LED PILOT PA (0x29)

| BIT | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------|---------------|---|---|---|---|---|---|---|
| Field | PILOT_PA[7:0] | | | | | | | |
| Reset | 0x00 | | | | | | | |
| Access Type | Write, Read | | | | | | | |

PILOT_PA

The purpose of PILOT_PA is to set the LED power during the PROX mode, as well as in Multi-LED mode. These bits set the nominal drive current for the pilot mode as shown in the table below.

When LED x is used, the respective LEDx_RGE is used to control the range of the LED driver in conjunction with PILOT_PA. For instance, if LED1 is used in the PILOT mode, then, LED1_RGE together with PILOT_PA will be used to set the LED1 current.

| LEDX_RGE[1:0] | 00 | 01 | 10 | 11 |
|---------------|------------------|------------------|------------------|------------------|
| PILOT_PA[7:0] | LED Current (mA) | LED Current (mA) | LED Current (mA) | LED Current (mA) |
| 00000000 | 0.00 | 0.00 | 0.00 | 0.00 |
| 00000001 | 0.12 | 0.24 | 0.36 | 0.48 |
| 00000010 | 0.24 | 0.48 | 0.73 | 0.97 |
| 00000011 | 0.36 | 0.73 | 1.09 | 1.45 |
| | | | | |
| 11111100 | 30.6 | 61.3 | 91.9 | 122.5 |
| 11111101 | 30.8 | 61.5 | 92.3 | 123.0 |
| 11111110 | 30.9 | 61.8 | 92.6 | 123.5 |
| 11111111 | 31.0 | 62.0 | 93.0 | 124.0 |
| LSB | 0.12 | 0.24 | 0.36 | 0.48 |

LED Range 1 (0x2A)

| BIT | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------|---|---|---------------|---|---------------|---|---------------|---|
| Field | – | – | LED3_RGE[1:0] | | LED2_RGE[1:0] | | LED1_RGE[1:0] | |
| Reset | – | – | 0x00 | | 0x00 | | 0x00 | |
| Access Type | – | – | Write, Read | | Write, Read | | Write, Read | |

LED3_RGE

Range selection of the LED current. Refer to *LED1_PA[7:0]* for more details.

| LEDX_RGE[1:0] (X = 1 TO 6) | LED CURRENT(mA) |
|-------------------------------|-----------------|
| 00 | 31 |
| 01 | 62 |
| 10 | 93 |
| 11 | 124 |

LED2_RGE

Range selection of the LED current. Refer to *LED3_RGE[1:0]* for more details.

LED1_RGE

Range selection of the LED current. Refer to *LED3_RGE[1:0]* for more details.

LED Range 2 (0x2B)

| BIT | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------|---|---|---------------|---|---------------|---|---------------|---|
| Field | – | – | LED6_RGE[1:0] | | LED5_RGE[1:0] | | LED4_RGE[1:0] | |
| Reset | – | – | 0x00 | | 0x00 | | 0x00 | |
| Access Type | – | – | Write, Read | | Write, Read | | Write, Read | |

LED6_RGE

Range selection of the LED current. Refer to *LED3_RGE[1:0]* for more details.

LED5_RGE

Range selection of the LED current. Refer to *LED3_RGE[1:0]* for more details.

LED4_RGE

Range selection of the LED current. Refer to *LED3_RGE[1:0]* for more details.

S1 HI RES DAC1 (0x2C)

| BIT | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------|------------------|---|-------------------|---|---|---|---|---|
| Field | S1_HRES_DAC1_OVR | – | S1_HRES_DAC1[5:0] | | | | | |
| Reset | 0x0 | – | 0x00 | | | | | |
| Access Type | Write, Read | – | Write, Read | | | | | |

S1_HRES_DAC1_OVR

| VALUE | ENUMERATION | DECODE |
|-------|-------------|---|
| 0 | OFF | The high resolution DAC for PPG1 is controlled by the chip. |
| 1 | ON | This allows the high-resolution DAC for PPG1 used in exposure 1 to be controlled by the software. |

S1_HRES_DAC1

If S1_HI_RES_DAC1_OVR = 1, then S1_HRES_DAC1 sets the high-resolution DAC code used in PPG1 ADC. This allows the algorithm to control ADC subranging.

If S1_HI_RES_DAC1_OVR = 0, then S1_HRES_DAC1 has no effect on the PPG1 ADC.

S2 HI RES DAC1 (0x2D)

| BIT | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------|------------------|---|-------------------|---|---|---|---|---|
| Field | S2_HRES_DAC1_OVR | – | S2_HRES_DAC1[5:0] | | | | | |
| Reset | 0x0 | – | 0x00 | | | | | |
| Access Type | Write, Read | – | Write, Read | | | | | |

S2_HRES_DAC1_OVR

| VALUE | ENUMERATION | DECODE |
|-------|-------------|---|
| 0 | OFF | The high resolution DAC for PPG1 is controlled by the chip. |
| 1 | ON | This allows the high-resolution DAC for PPG1 used in exposure 2 to be controlled by the software. |

S2_HRES_DAC1

If S2_HI_RES_DAC1_OVR = 1, then S2_HRES_DAC1 sets the high-resolution DAC code used in PPG1 ADC. This allows the algorithm to control ADC subranging.

If S2_HI_RES_DAC1_OVR = 0, then S2_HRES_DAC1 has no effect on the PPG1 ADC.

S3 HI RES DAC1 (0x2E)

| BIT | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------|------------------|---|-------------------|---|---|---|---|---|
| Field | S3_HRES_DAC1_OVR | – | S3_HRES_DAC1[5:0] | | | | | |
| Reset | 0x0 | – | 0x0 | | | | | |
| Access Type | Write, Read | – | Write, Read | | | | | |

S3_HRES_DAC1_OVR

| VALUE | ENUMERATION | DECODE |
|-------|-------------|---|
| 0 | OFF | The high resolution DAC for PPG1 is controlled by the chip |
| 1 | ON | This allows the high-resolution DAC for PPG1 used in exposure 3 to be controlled by the software. |

S3_HRES_DAC1

If S3_HI_RES_DAC1_OVR = 1 then S3_HRES_DAC1 sets the high-resolution DAC code used in PPG1 ADC. This allows the algorithm to control ADC subranging.

If S3_HI_RES_DAC1_OVR = 0 then S3_HRES_DAC1 has no effect on the PPG1 ADC.

S4 HI RES DAC1 (0x2F)

| BIT | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------|------------------|---|-------------------|---|---|---|---|---|
| Field | S4_HRES_DAC1_OVR | – | S4_HRES_DAC1[5:0] | | | | | |
| Reset | 0b0 | – | 0x0 | | | | | |
| Access Type | Write, Read | – | Write, Read | | | | | |

S4_HRES_DAC1_OVR

| VALUE | ENUMERATION | DECODE |
|-------|-------------|---|
| 0 | OFF | The high resolution DAC for PPG1 is controlled by the chip. |
| 1 | ON | This allows the high-resolution DAC for PPG1 used in exposure 4 to be controlled by the software. |

S4_HRES_DAC1

If S4_HI_RES_DAC1_OVR = 1 then S4_HRES_DAC1 sets the high-resolution DAC code used in PPG1 ADC. This allows the algorithm to control ADC subranging.

If S4_HI_RES_DAC1_OVR = 0 then S4_HRES_DAC1 has no effect on the PPG1 ADC.

S5 HI RES DAC1 (0x30)

| BIT | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------|------------------|---|-------------------|---|---|---|---|---|
| Field | S5_HRES_DAC1_OVR | – | S5_HRES_DAC1[5:0] | | | | | |
| Reset | 0b0 | – | 0x0 | | | | | |
| Access Type | Write, Read | – | Write, Read | | | | | |

S5_HRES_DAC1_OVR

| VALUE | ENUMERATION | DECODE |
|-------|-------------|---|
| 0 | OFF | The high-resolution DAC for PPG1 is controlled by the chip. |
| 1 | ON | This allows the high-resolution DAC for PPG1 used in exposure 5 to be controlled by the software. |

S5_HRES_DAC1

If S5_HI_RES_DAC1_OVR = 1, then S5_HRES_DAC1 sets the high-resolution DAC code used in PPG1 ADC. This allows the algorithm to control ADC subranging.

If S5_HI_RES_DAC1_OVR = 0, then S5_HRES_DAC1 has no effect on the PPG1 ADC.

S6 HI RES DAC1 (0x31)

| BIT | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------|------------------|---|-------------------|---|---|---|---|---|
| Field | S6_HRES_DAC1_OVR | – | S6_HRES_DAC1[5:0] | | | | | |
| Reset | 0b0 | – | 0x0 | | | | | |
| Access Type | Write, Read | – | Write, Read | | | | | |

S6_HRES_DAC1_OVR

| VALUE | ENUMERATION | DECODE |
|-------|-------------|---|
| 0 | OFF | The high-resolution DAC for PPG1 is controlled by the chip. |
| 1 | ON | This allows the high-resolution DAC for PPG1 used in exposure 6 to be controlled by the software. |

S6_HRES_DAC1

If S6_HI_RES_DAC1_OVR = 1, then S6_HRES_DAC1 sets the high-resolution DAC code used in PPG1 ADC. This allows the algorithm to control ADC subranging.

If S6_HI_RES_DAC1_OVR = 0, then S6_HRES_DAC1 has no effect on the PPG1 ADC.

S1_HI RES DAC2 (0x32)

| BIT | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------|------------------|---|-------------------|---|---|---|---|---|
| Field | S1_HRES_DAC2_OVR | – | S1_HRES_DAC2[5:0] | | | | | |
| Reset | 0x0 | – | 0x00 | | | | | |
| Access Type | Write, Read | – | Write, Read | | | | | |

S1_HRES_DAC2_OVR

| VALUE | ENUMERATION | DECODE |
|-------|-------------|---|
| 0 | OFF | The high-resolution DAC for PPG2 is controlled by the chip. |
| 1 | ON | This allows the high-resolution DAC for PPG2 used in exposure 1 to be controlled by the software. |

S1_HRES_DAC2

If S1_HI_RES_DAC2_OVR = 1, then S1_HRES_DAC2 sets the high-resolution DAC code used in PPG2 ADC. This allows the algorithm to control ADC subranging.

If S1_HI_RES_DAC2_OVR = 0, then S1_HRES_DAC2 has no effect on the PPG2 ADC.

S2_HI RES DAC2 (0x33)

| BIT | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------|------------------|---|-------------------|---|---|---|---|---|
| Field | S2_HRES_DAC2_OVR | – | S2_HRES_DAC2[5:0] | | | | | |
| Reset | 0x0 | – | 0x0 | | | | | |
| Access Type | Write, Read | – | Write, Read | | | | | |

S2_HRES_DAC2_OVR

| VALUE | ENUMERATION | DECODE |
|-------|-------------|---|
| 0 | OFF | The high-resolution DAC for PPG2 is controlled by the chip. |
| 1 | ON | This allows the high-resolution DAC for PPG2 used in exposure 2 to be controlled by the software. |

S2_HRES_DAC2

If S2_HI_RES_DAC2_OVR = 1, then S2_HRES_DAC2 sets the high-resolution DAC code used in PPG2 ADC. This allows the algorithm to control ADC subranging.

If S2_HI_RES_DAC2_OVR = 0, then S2_HRES_DAC2 has no effect on the PPG2 ADC.

S3_HI RES DAC2 (0x34)

| BIT | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------|------------------|---|-------------------|---|---|---|---|---|
| Field | S3_HRES_DAC2_OVR | – | S3_HRES_DAC2[5:0] | | | | | |
| Reset | 0b0 | – | 0x0 | | | | | |
| Access Type | Write, Read | – | Write, Read | | | | | |

S3_HRES_DAC2_OVR

| VALUE | ENUMERATION | DECODE |
|-------|-------------|---|
| 0 | OFF | The high-resolution DAC for PPG2 is controlled by the chip. |
| 1 | ON | This allows the high-resolution DAC for PPG2 used in exposure 3 to be controlled by the software. |

S3_HRES_DAC2

If S3_HI_RES_DAC2_OVR = 1, then S3_HRES_DAC2 sets the high-resolution DAC code used in PPG2 ADC. This allows the algorithm to control ADC subranging.

If S3_HI_RES_DAC2_OVR = 0 then S3_HRES_DAC2 has no effect on the PPG2 ADC.

S4 HI RES DAC2 (0x35)

| BIT | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------|------------------|---|-------------------|---|---|---|---|---|
| Field | S4_HRES_DAC2_OVR | – | S4_HRES_DAC2[5:0] | | | | | |
| Reset | 0b0 | – | 0x0 | | | | | |
| Access Type | Write, Read | – | Write, Read | | | | | |

S4_HRES_DAC2_OVR

| VALUE | ENUMERATION | DECODE |
|-------|-------------|---|
| 0 | OFF | The high-resolution DAC for PPG2 is controlled by the chip. |
| 1 | ON | This allows the high-resolution DAC for PPG2 used in exposure 4 to be controlled by the software. |

S4_HRES_DAC2

If S4_HI_RES_DAC2_OVR = 1, then S4_HRES_DAC2 sets the high resolution DAC code used in PPG2 ADC. This allows the algorithm to control ADC subranging.

If S4_HI_RES_DAC2_OVR = 0, then S4_HRES_DAC2 has no effect on the PPG2 ADC.

S5 HI RES DAC2 (0x36)

| BIT | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------|------------------|---|-------------------|---|---|---|---|---|
| Field | S5_HRES_DAC2_OVR | – | S5_HRES_DAC2[5:0] | | | | | |
| Reset | 0b0 | – | 0x0 | | | | | |
| Access Type | Write, Read | – | Write, Read | | | | | |

S5_HRES_DAC2_OVR

| VALUE | ENUMERATION | DECODE |
|-------|-------------|---|
| 0 | OFF | The high-resolution DAC for PPG2 is controlled by the chip. |
| 1 | ON | This allows the high-resolution DAC for PPG2 used in exposure 5 to be controlled by the software. |

S5_HRES_DAC2

If S5_HI_RES_DAC2_OVR = 1, then S5_HRES_DAC2 sets the high-resolution DAC code used in PPG2 ADC. This allows the algorithm to control ADC subranging.

If S5_HI_RES_DAC2_OVR = 0, then S5_HRES_DAC2 has no effect on the PPG2 ADC.

S6 HI RES DAC2 (0x37)

| BIT | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------|------------------|---|-------------------|---|---|---|---|---|
| Field | S6_HRES_DAC2_OVR | – | S6_HRES_DAC2[5:0] | | | | | |
| Reset | 0b0 | – | 0x0 | | | | | |
| Access Type | Write, Read | – | Write, Read | | | | | |

S6_HRES_DAC2_OVR

| VALUE | ENUMERATION | DECODE |
|-------|-------------|---|
| 0 | OFF | The high-resolution DAC for PPG2 is controlled by the chip. |
| 1 | ON | This allows the high-resolution DAC for PPG2 used in exposure 6 to be controlled by the software. |

S6_HRES_DAC2

If S6_HI_RES_DAC2_OVR = 1, then S6_HRES_DAC2 sets the high-resolution DAC code used in PPG2 ADC. This allows the algorithm to control ADC subranging.

If S6_HI_RES_DAC2_OVR = 0, then S6_HRES_DAC2 has no effect on the PPG2 ADC.

Die Temperature Configuration (0x40)

| BIT | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------|---|---|---|---|---|---|---|-------------|
| Field | – | – | – | – | – | – | – | TEMP_EN |
| Reset | – | – | – | – | – | – | – | 0x0 |
| Access Type | – | – | – | – | – | – | – | Write, Read |

TEMP_EN

The bit gets cleared after temperature measurement completes.

| VALUE | ENUMERATION | DECODE |
|-------|-------------|-----------------------------------|
| 0 | NORMAL | Idle |
| 1 | ASSERTED | Start one temperature measurement |

Die Temperature Integer (0x41)

| BIT | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------|---------------|---|---|---|---|---|---|---|
| Field | TEMP_INT[7:0] | | | | | | | |
| Reset | 0x0 | | | | | | | |
| Access Type | Read Only | | | | | | | |

TEMP_INT

This register stores the integer temperature data in 2s compliment form. 0x00 = 0°C, 0x7F = 127°C and 0x80 = -128°C

Note: TINT and TFRAC registers should be read through the Serial Interface in burst mode, to ensure that they belong to the same sample.

Die Temperature Fraction (0x42)

| BIT | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------|---|---|---|---|----------------|---|---|---|
| Field | – | – | – | – | TEMP_FRAC[3:0] | | | |
| Reset | – | – | – | – | 0x0 | | | |
| Access Type | – | – | – | – | Read Only | | | |

TEMP_FRAC

This register store the fractional temperature data in increments of 0.0625°C. 0x1 = 0.0625°C and 0xF = 0.9375°C.

Note: TINT and TFRAC registers should be read through the Serial Interface in burst mode, to ensure that they belong to the same sample.

SHA Command (0xF0)

| BIT | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------|--------------|---|---|---|---|---|---|---|
| Field | SHA_CMD[7:0] | | | | | | | |
| Reset | 0x0 | | | | | | | |
| Access Type | Write, Read | | | | | | | |

SHA_CMD

| VALUE | DECODE |
|--------|--------------------|
| 0x35 | MAC WITH ROM ID |
| 0x36 | MAC WITHOUT ROM ID |
| OTHERS | RESERVED |

SHA Configuration (0xF1)

| BIT | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------|---|---|---|---|---|---|-------------|-------------|
| Field | — | — | — | — | — | — | SHA_EN | SHA_START |
| Reset | — | — | — | — | — | — | 0x0 | 0x0 |
| Access Type | — | — | — | — | — | — | Write, Read | Write, Read |

SHA_EN

Authentication is performed using a FIPS 180-3 compliant SHA-256 one-way hash algorithm on a 512-bit message block. The message block consists of a 160-bit secret, a 160-bit challenge and 192 bits of constant data. Optionally, the 64-bit ROM ID replaces 64 of the 192 bits of constant data used in the hash operation. 16 bits out of the 160-bit secret and 16 bits of ROM ID are programmable—8 bits each in metal and 8 bits each in OTP bits.

The host and the MAX86140 both calculate the result based on a mutually known secret. The result of the hash operation is known as the message authentication code (MAC) or message digest. The MAC is returned by the MAX86140 for comparison with the host's MAC. Note that the secret is never transmitted on the bus and thus cannot be captured by observing bus traffic. Each authentication attempt is initiated by the host system by writing a 160-bit random challenge into the SHA memory address space 0x00h to 0x09h. The host then issues the compute MAC or compute MAC with ROM ID command. The MAC is computed per FIPS 180-3, and stored in address space 0x00h to 0x0Fh overwriting the challenge value.

Note that the results of the authentication attempt are determined by host verification. Operation of the MAX86140 is not affected by authentication success or failure.

Sequence of operation is as follows:

- Enable SHA_DONE Interrupt.
- Enable SHA_EN bit.
- Write 160-bit random challenge value to RAM using registers MEM_IDX and MEM_DATA.
- Write command, with ROM ID (0x35) or without ROM ID (0x36), to SHA_CMD register.
- Write 1 to SHA_START and 1 to SHA_EN bit.
- Wait for SHA_DONE interrupt.
- Read 256 MAC value from RAM using registers MEM_IDX and MEM_DATA.
- Compare MAC from MAX86140 with Host's precalculated MAC.
- Check PASS or FAIL.
- Disable SHA_EN bit (Write 0 to SHA_EN bit).

| VALUE | ENUMERATION | DECODE |
|-------|-------------|----------------------------|
| 0 | NORMAL | Authentication is disabled |
| 1 | ASSERTED | Authentication is enabled |

SHA_START

The bit gets cleared after authentication completes. The valid command (0x35 or 0x36) should be written to the SHA_CMD register and challenge value should be written to the RAM by Host before writing 1 to this bit.

Memory Control (0xF2)

| BIT | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------|---|---|---|---|---|---|-------------|-------------|
| Field | – | – | – | – | – | – | MEM_WR_EN | BANK_SEL |
| Reset | – | – | – | – | – | – | 0x0 | 0x0 |
| Access Type | – | – | – | – | – | – | Write, Read | Write, Read |

MEM_WR_EN

Enable write access to Memory through SPI.

| VALUE | ENUMERATION | DECODE |
|-------|-------------|--|
| 0 | DISABLED | Writing to memory through SPI is disabled. |
| 1 | ENABLED | Writing to memory through SPI is enabled |

BANK_SEL

Selects the memory bank for reading and writing.

Burst reading or writing the memory past 0xFF automatically increments BANK_SEL to 1.

| VALUE | DECODE |
|-------|---------------------------------------|
| 0 | Select Bank 0, address 0x00 to 0xFF |
| 1 | Select Bank 1, address 0x100 to 0x17f |

Memory Index (0xF3)

| BIT | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------|--------------|---|---|---|---|---|---|---|
| Field | MEM_IDX[7:0] | | | | | | | |
| Reset | 0x0 | | | | | | | |
| Access Type | Write, Read | | | | | | | |

MEM_IDX

Index to Memory for reading and writing. The memory is 384 bytes, and is divided into two banks - Bank 0 from 0x00 to 0xFF and Bank 1 is from 0x100 to 0x17F. The bank is selected by the BANK_SEL register bit. MEM_IDX is the starting address for burst writing to or reading from memory. Burst accessing the memory past 0xFF accesses Bank 1. The memory address saturates at 0x17F.

Memory Data (0xF4)

| BIT | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------|-------------------|---|---|---|---|---|---|---|
| Field | MEM_DATA[7:0] | | | | | | | |
| Reset | 0x0 | | | | | | | |
| Access Type | Write, Read, Dual | | | | | | | |

MEM_DATA

Data to be written or data read from Memory

Reading this register does not automatically increment the register address. So burst reading this register read the same register over and over, but the address to the Memory autoincrements until BANK_SEL becomes 1 and MEM_IDX becomes 0x7F.

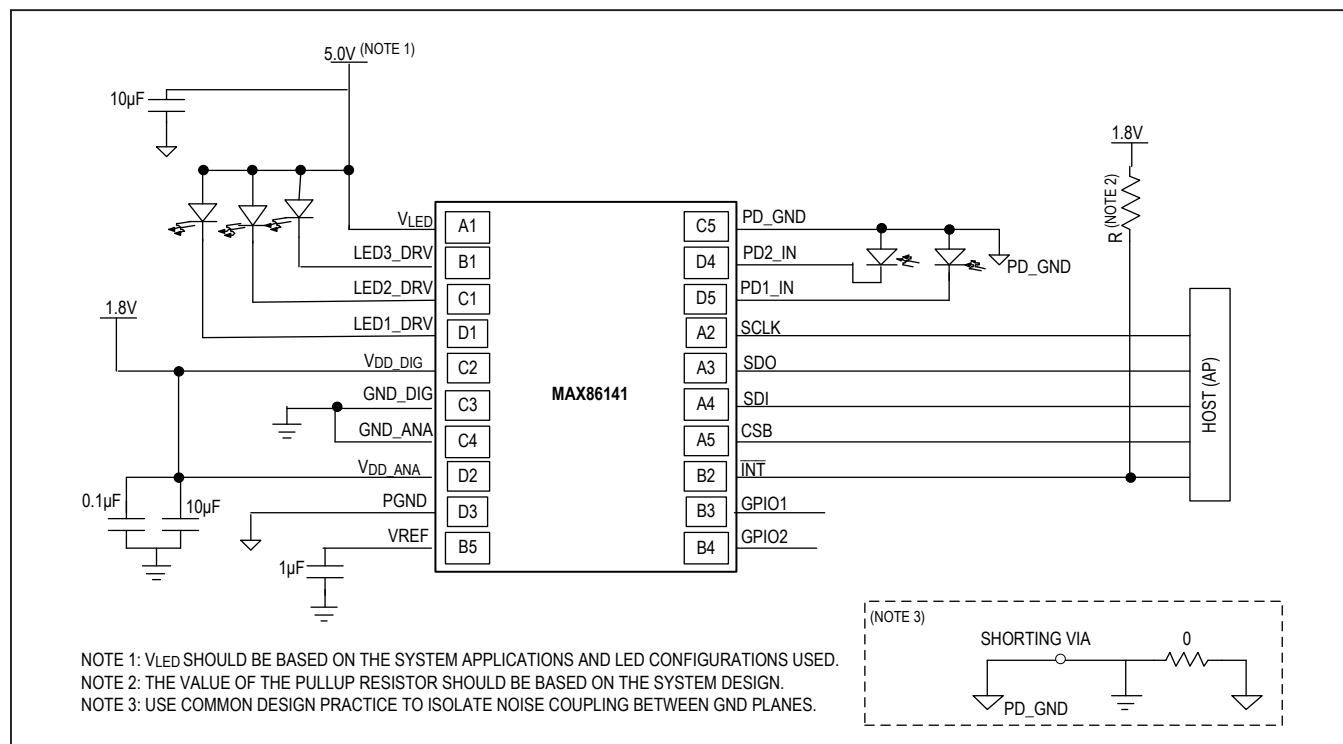
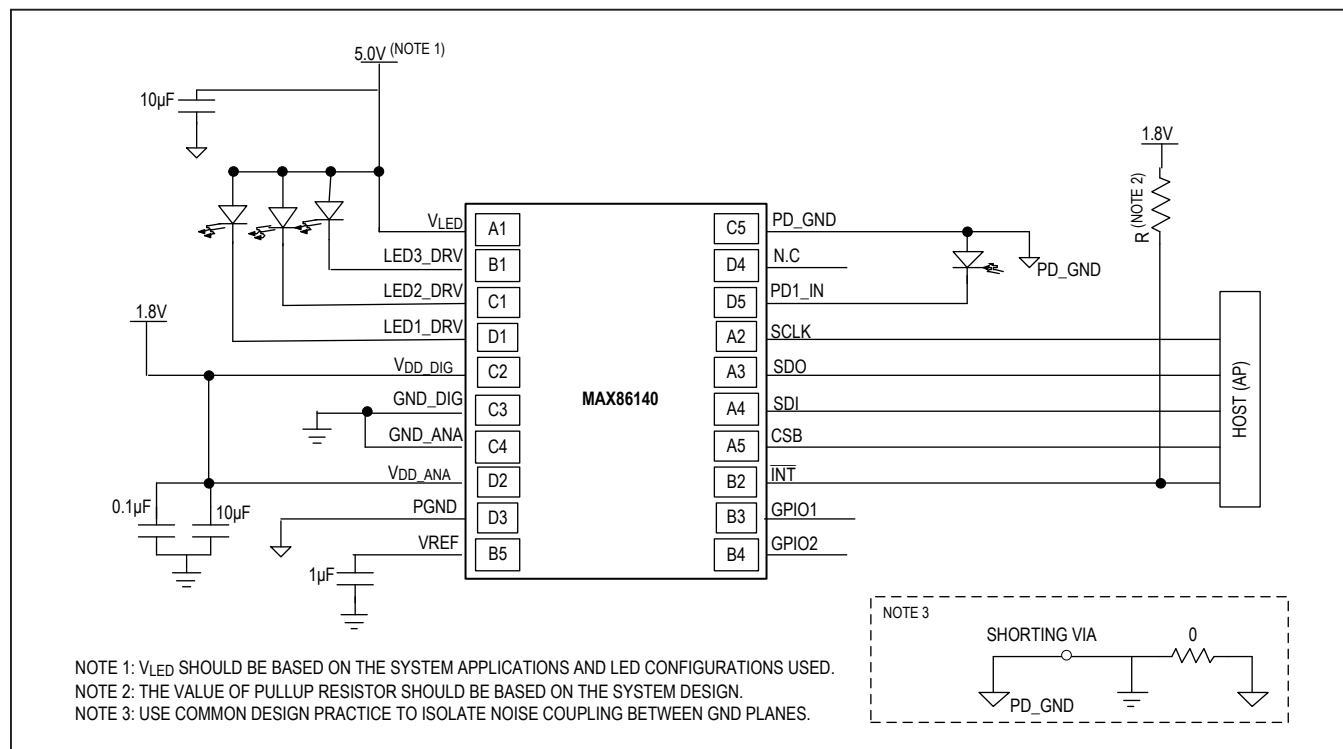
Part ID (0xFF)

| BIT | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------|--------------|---|---|---|---|---|---|---|
| Field | PART_ID[7:0] | | | | | | | |
| Reset | 0xFF | | | | | | | |
| Access Type | Read Only | | | | | | | |

PART_ID

This register stores the part identifier for the chip.

| PART_ID | MAX # | # OF PPG CHANNELS |
|---------|----------|-------------------|
| 0x24 | MAX86140 | 1 |
| 0x25 | MAX86141 | 2 |



MAX86140/ MAX86141

Best-in-Class Optical Pulse Oximeter and Heart-Rate Sensor for Wearable Health

Ordering Information

| PART NUMBER | TEMP RANGE | PIN-PACKAGE | CONFIGURATION |
|--------------|----------------|---|----------------------------|
| MAX86140ENP+ | -40°C to +85°C | 20-pin WLP, 2.048mm x 1.848mm, 5 x 4, 0.4mm ball pitch | Single-Channel Optical AFE |
| MAX86141ENP+ | -40°C to +85°C | 20-pin WLP, 2.048mm x 1.848mm, 5 x 4, 0.4mm ball pitch | Dual-Channel Optical AFE |

+Denotes a lead(Pb)-free/RoHS-compliant package.

T = Tape and reel.

Revision History

| REVISION NUMBER | REVISION DATE | DESCRIPTION | PAGES CHANGED |
|--------------------|------------------|--|--|
| 0 | 5/17 | Initial release | — |
| 1 | 8/17 | Added MAX86141 part number to data sheet | 1–88 |
| 2 | 8/19 | Updated the <i>General Description</i> , <i>Benefits and Features</i> , <i>Simplified Block Diagram</i> , <i>Absolute Maximum Ratings</i> , <i>Pin Description</i> , <i>Detailed Description</i> , <i>Optical Subsystem</i> , Table 3, <i>FIFO Data (Register 0x08)</i> , <i>FIFO_A_FULL</i> (address 0x09), <i>A_FULL_TYPE</i> (Address 0x0A), <i>GPIO Configuration</i> , <i>Picket Fence Detect-and-Replace Function</i> , Figure 31, <i>Photodiode Biasing</i> , <i>Layout Guidelines</i> , Table 16, <i>SPI FIFO Burst Mode Read Transaction</i> , VDD_OOR Bit, PPG2_ADC_RGE table, PPG1_ADC_RGE table, PPG_TINT table, both PPG_SR tables, SMP_AVE table, LED_SETLNG table, DIG_FILT_SEL table, BURST_RATE table, PDBIAS2 table, PDBIAS1 table, PF_ENABLE table, IIR_TC table, IIR_INIT_VALUE table, THRESHOLD_SIGMA_MULT table, LED1_DRV table, PILOT_PA table, LED3_RGE table, S1_HRES_DAC1 description, S2_HRES_DAC1 description, S3_HRES_DAC1 description, S4_HRES_DAC1 description, S5_HRES_DAC1 description, S6_HRES_DAC1 description, TEMP_EN table, SHA_CMD table, SHA_EN table, MEM_WR_EN table, and BANK_SEL table; replaced the <i>Typical Operating Characteristics</i> , <i>Pin Configurations</i> , Table 6, <i>Pseudo-Code Example of Initializing the Optical AFE</i> , <i>Pseudo-Code for Interrupt Handling with FIFO_A_FULL</i> , <i>Pseudo-Code Example of Reading Data from FIFO</i> , Figure 7, Figure 11, Figure 12, Figure 14, Figure 15, Figure 16, Figure 17, Figure 18, Figure 20, Figure 21, Figure 22, Figure 24, Figure 25, Figure 26, Figure 27, Figure 29, and <i>Typical Application Circuit</i> , adding additional <i>Typical Application Circuits</i> ; added a <i>Detailed Block Diagram</i> , new Figures 25–27 (after renumbering), <i>GPIO CTRL[3:0]</i> <i>1010 Hardware Sync</i> and Table 15 (and renumbered remaining tables); deleted Figure 9, Figure 13, Figure 19, Figure 23, Figure 28, and renumbered existing figures | 1–15, 17–23, 28–44, 46–49, 53–55, 59, 68–74, 76–86, 88 |
| 3 | 1/21 | Updated the <i>Electrical Characteristics</i> , <i>Pin Configurations</i> , and <i>Proximity Mode Function</i> sections; updated Table 2, Figure 1, Figure 2, Figure 10, and Figure 13; updated Register 0x25, and the VDD_OOR, PWR_RDY, FIFO_A_FULL, PPG_TINT, THRESHOLD_SIGMA_MULT, LED1_DRV and TEMP_INT bits | 6–8, 11–12, 16 24–25, 30, 32, 44, 56, 59, 63, 68, 74, 76, 84 |
| 4 | 10/21 | Updated <i>Detailed Block Diagrams</i> MAX86140, MAX86141, <i>Pin Description</i> , caption for Figures 7, 10, 15, and 22 | 2, 3, 12, 27, 29, 33, 39 |
| 5 | 10/21 | Replaced the term Master with Controller and Slave with Target in the entire document. Added Note to Figure 6 and <i>ADC Architecture and Transfer Function Non-Linearity (XNL) Trim</i> Section. Updated Figure 20, 21, 22, 23, and 24 | 1–90 |

Mouser Electronics

Authorized Distributor

Click to View Pricing, Inventory, Delivery & Lifecycle Information:

[Analog Devices Inc.:](#)

[MAX86140ENP+](#) [MAX86141ENP+](#) [MAX86140ENP+T](#) [MAX86141ENP+T](#)