Click here for production status of specific part numbers.

MAXM86161

Single-Supply Integrated Optical Module for HR and SpO_2 Measurement

General Description

The MAXM86161 is an ultra-low-power, completely integrated, optical data-acquisition system. On the transmitter side, the MAXM86161 has three programmable high-current LED drivers. On the receiver side, MAXM86161 consists of a high efficiency PIN photo-diode and an optical readout channel. The optical readout has a 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 algorithm. Due to the low power consumption, compact size, easy, flexible-to-use, and industry lead ambient light rejection capability of the MAXM86161, the device is ideal for a wide variety of optical sensing applications such as heart rate detection and pulse oximetry.

The MAXM86161 operates on a 3.0V to 5.5V V_{LED} single supply voltage. It supports a standard compatible interface and fully autonomous operation. Each device has a large 128-word built-in FIFO. The MAXM86161 is available in compact 2.9mm x 4.3mm x 1.4mm, 14-pin OLGA package.

Applications

- Optimized for In-Ear Applications
- Miniature Package for Mobile Applications
- Optimized Performance to Detect:
  - Optical Heart Rate
  - Oxygen Saturation (SpO_2)
  - Continuous Monitoring for HRV

Benefits and Features

- Complete Single-Channel Optical Data Acquisition System
- Built-In Algorithm Further Enhances Rejection of Fast Ambient Transients
- Optimized Architecture for Reflective Heart Rate and SpO_2 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)
- 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, Typical at 25sps
  - Short Exposure Integration Period of 14.8µs, 29.4µs, 58.7µs, 117.3µs
  - Low Shutdown Current = 1.6µA (typ)
- Miniature 2.9mm x 4.3mm x 1.4mm, 14-pin OLGA Package
- -40°C to +85°C Operating Temperature Range

***Ordering Information* appears at end of data sheet.**

19-100523; Rev 0; 3/19

maxim integrated™

TABLE OF CONTENTS

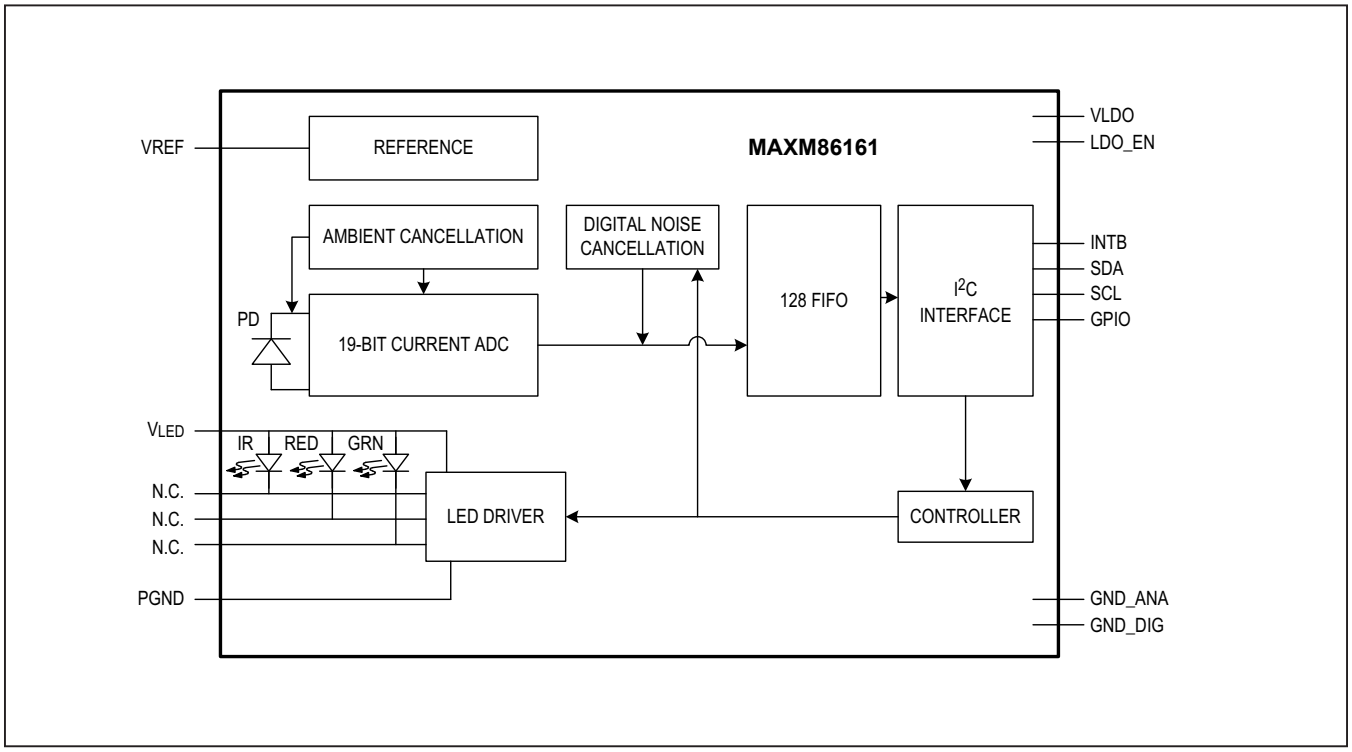
TABLE OF CONTENTS (CONTINUED)

TABLE OF CONTENTS (CONTINUED)

LIST OF FIGURES

LIST OF TABLES

Simplified Block Diagram



Absolute Maximum Ratings

VLDO to GND_ANA .....-0.3V to +2.2V
VLDO to GND_DIG .....-0.3V to +2.2V
GND_DIG to GND_ANA.....-0.3V to +0.3V
PGND to GND_ANA.....-0.3V to +0.3V
SCL, SDA, INTB, GPIO to GND_ANA .....-0.3V to +6.0V
LDO_EN to GND_DIG.....-0.3V to +6.0V
V_{LED} to PGND .....-0.3V to +6.0V

Output Short-Circuit Duration .....Continuous
Continuous Input Current Into Any Pin (except LED_DRVx Pins) .....±20mA
Continuous Power Dissipation (OLGA; Derate 5.5mW/°C above +70°C).....36mW
Operating Temperature Range..... -40°C to +85°C
Storage Temperature Range ..... -40°C to +105°C
Soldering Temperature (reflow) .....+260°C

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

PACKAGE TYPE: 14-PIN OLGA	
Package Code	F142A4+1
Outline Number	**21-100309**
Land Pattern Number	**90-100106**
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_{LED} = 5.0V, PPG1_ADC_RGE = 16µA, PPG_SR = 512sps, PPG_TINT = 117.3µs, LED_SETLNG = 6µs, LEDx_RGE = 31mA, PDBIAS1 = 0x1, T_A = 25°C, min/max are from T_A = -40°C to +85°C, unless otherwise noted.) (Note 1)

PARAMETER	SYMBOL	CONDITIONS		MIN	TYP	MAX	UNITS
POWER SUPPLY							
LED Supply Voltage (Note 2)	V_{LED}	Verified during PSRR Test		3.0		5.5	V
LDO Output Voltage	VLDO			1.68	1.8	1.92	V
Average LED Supply Current	I_{LED}	LEDx_PA = 0x00, PPG_TINT = 117.3µs, PPG_SR = 100sps	Three LED Exposures/ Sample		400	600	µA
		LEDx_PA = 0xFF, PPG_TINT = 117.3µs, PPG_SR = 100sps	One LED Exposure/Sample		600		
			Two LED Exposure/Sample		970		
			Three LED Exposure/Sample		1300		
LED Supply Current in Shutdown	$I_{LEDSHDN}$	LDO_EN = 1, SHDN = 1			1.6	11	µA
		LDO_EN = 0			0.05	0.3	
V_{REF}				1.195	1.210	1.220	V

Electrical Characteristics (continued)

(V_{LED} = 5.0V, PPG1_ADC_RGE = 16µA, PPG_SR = 512sps, PPG_TINT = 117.3µs, LED_SETLNG = 6µs, LEDx_RGE = 31mA, PDBIAS1 = 0x1, T_A = 25°C, min/max are from T_A = -40°C to +85°C, unless otherwise noted.) (Note 1)

PARAMETER	SYMBOL	CONDITIONS		MIN	TYP	MAX	UNITS
READOUT CHANNEL							
ADC Resolution					19		bits
IR ADC Count	IR_C	Propriety ATE setup, PPG_TINT = 117.3µs, PPG_SR = 512sps, T_A = +25°C	PPG1_ADC_RGE = 32µA, LED2_RGE = 124mA LED2_PA = 0xFF	-15%	155000	+15%	Counts
Green ADC Count	$GREEN_C$	Propriety ATE setup, PPG_TINT = 117.3µs, PPG_SR = 512sps, T_A = +25°C	PPG1_ADC_RGE = 16µA, LED1_RGE = 124mA LED1_PA = 0xFF	-15%	135000	+15%	Counts
Red ADC Count	RED_C	Propriety ATE setup, PPG_TINT = 117.3µs, PPG_SR = 512sps, T_A = +25°C	PPG1_ADC_RGE = 32µA LED3_RGE = 124mA LED3_PA = 0xFF	-15%	200000	+15%	Counts
IR ADC Noise	IR_{STD}	Bench test setup, PPG_TINT = 117.3µs, PPG_SR = 128sps, T_A = +25°C	PPG1_ADC_RGE = 32µA, LED2_RGE = 124mA LED2_PA = 0xFF		7		Counts
Green ADC Noise	$GREEN_{STD}$	Bench test setup, PPG_TINT = 117.3µs, PPG_SR = 128sps, T_A = +25°C	PPG1_ADC_RGE = 16µA, LED1_RGE = 124mA LED1_PA = 0xFF		7		Counts
Red ADC Noise	RED_{STD}	Bench test setup, PPG_TINT = 117.3µs, PPG_SR = 128sps, T_A = +25°C	PPG1_ADC_RGE1 = 32µA, LED3_RGE = 124mA LED3_PA = 0xFF		8.9		Counts
ADC Full Scale Input Current		PPG1_ADC_RGE = 0x0			4.0		µA
		PPG1_ADC_RGE = 0x1			8.0		
		PPG1_ADC_RGE = 0x2			16.0		
		PPG1_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

Electrical Characteristics (continued)

(V_{LED} = 5.0V, PPG1_ADC_RGE = 16µA, PPG_SR = 512sps, PPG_TINT = 117.3µs, LED_SETLNG = 6µs, LEDx_RGE = 31mA, PDBIAS1 = 0x1, T_A = 25°C, min/max are from T_A = -40°C to +85°C, unless otherwise noted.) (Note 1)

PARAMETER	SYMBOL	CONDITIONS		MIN	TYP	MAX	UNITS
Maximum PPG Sample Rate		PPG_SR = 0x13			4096		sps
Sample Rate Error		From nominal as indicated in the PPG_SR table		-2		+2	%
Dark Current Count	LED_DC_C	LEDx_PA = 0x00, PPG_TINT = 117.8µs, PPG_SR = 100sps, PPG1_ADC_RGE = 4µA			1	20	Counts
		LEDx_PA = 0x00, PPG_TINT = 117.8µs, PPG_SR = 100sps, PPG1_ADC_RGE = 4µA			0.001	0.004	% of FS
Maximum DC Ambient Light Rejection	ALR	T_A = +25°C			200		µA
Ambient Light Detect Noise		LEDx_PA = 0x00, PPG_TINT = 117.8µs, ADD_OFFSET = 0x1, PPG_SR = 100sps, PPG1_ADC_RGE = 4µA, T_A = +25°C.			5	25	Counts
ADC Count - PSRR (V_{LED}) (Note 2)	$PSRR_{VLED}$	Propriety ATE setup, LEDx_RGE = 124mA, LEDx_PA = 0xFF, T_A = +25°C to +85°C	For IR 3V < V_{LED+} < 5.5V		0.05	1	% of FS
			For Red 4V < V_{LED+} < 5.5V		0.05	1	
			For Green 4.8V < V_{LED+} < 5.5V		0.05	1	
		LEDx_RGE = 124mA, PPG_TINT = 120µs	Frequency = DC to 100kHz, 100mV_{P-P}		40		LSB
LED DRIVER							
LED Current Resolution					8		Bits
Full Scale LED Current (Note 3)	I_{LED}	LEDx_PA = 0xFF	LEDx_RGE = 0x0		31		mA
			LEDx_RGE = 0x1		62		
			LEDx_RGE = 0x2		93		
			LEDx_RGE = 0x3		124		
Minimum Output Voltage	V_{OL}	LEDx_PA = 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		
LED1 Driver Compliance Interrupt	$LED1_{COMP}$				180		mV
IR LED CHARACTERISTICS (NOTE 4)							
Centroid Wavelength	$\lambda_{centroid}$	I_F = 100mA, tp = 10ms, T_A = +25°C			880		nm
Forward Voltage	V_F	I_F = 100mA, tp = 10ms, T_A = +25°C			1.6	1.9	V
Radiant Power	Φ_e	I_F = 100mA, T_A = +25°C		38			mW
Maximum Junction Temperature	T_J					115	°C

Electrical Characteristics (continued)

(V_{LED} = 5.0V, PPG1_ADC_RGE = 16µA, PPG_SR = 512sps, PPG_TINT = 117.3µs, LED_SETLNG = 6µs, LEDx_RGE = 31mA, PDBIAS1 = 0x1, T_A = 25°C, min/max are from T_A = -40°C to +85°C, unless otherwise noted.) (Note 1)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
RED LED CHARACTERISTICS (NOTE 4)						
Centroid Wavelength	$\lambda_{centroid}$	I_F = 20mA, T_A = +25°C		660		nm
Forward Voltage	V_F	I_F = 20mA, tp =5ms, T_A = +25°C		2.1	2.5	V
Radiant Power	Φ_e	I_F = 20mA, T_A = +25°C	9			mW
Maximum Junction Temperature	T_J				115	°C
GREEN LED CHARACTERISTICS (NOTE 4)						
Centroid Wavelength	$\lambda_{centroid}$	I_F = 150mA, tp = 10ms, T_A = +25°C	520		535	nm
Forward Voltage	V_F	I_F = 150mA, tp = 5ms, T_A = +25°C	3.0	3.35	3.8	V
Radiant Power	Φ_e	I_F = 150mA, T_A = +25°C	25			mW
Maximum junction temperature	T_J				150	°C
PHOTODIODE (NOTE 4)						
Wavelength of Peak Sensitivity				860		nm
Spectral Bandwidth Range				420 to 1020		nm
DIGITAL I/O CHARACTERISTICS (SDA, SCL, INTB, GPIO, LDO_EN)						
Open Drain Output Low Voltage	V_{OL_OD}	SDA, INTB, GPIO, I_{SINK} = 6mA			0.4	V
Input Voltage Low	V_{IL}	SDA and SCL			0.4	V
		LDO_EN			0.3	
Input Voltage High	V_{IH}	SDA and SCL	1.4			V
		LDO_EN	1.1			
Input Hysteresis	V_{HYS}	SDA and SCL		400		mV
Input Leakage Current	I_{IN}	V_{IN} = 0V, T_A = +25°C		0.01	1	µA
		V_{IN} = 5.5V, T_A = +25°C		0.01	1	
Input Capacitance	C_{IN}			10		pF
I²C TIMING CHARACTERISTICS (NOTE 5)						
I²C Write Address				C4		Hex
I²C Read Address				C5		Hex
Serial Clock Frequency	f_{SCL}		0		400	kHz
Bus Free Time Between STOP and START Conditions	t_{BUF}		1.3			µs
Hold Time START and Repeat START Condition	t_{HD_STA}		0.6			µs
SCL Pulse-Width Low	t_{LOW}		1.3			µs

Electrical Characteristics (continued)

(V_{LED} = 5.0V, PPG1_ADC_RGE = 16µA, PPG_SR = 512sps, PPG_TINT = 117.3µs, LED_SETLNG = 6µs, LEDx_RGE = 31mA, PDBIAS1 = 0x1, T_A = 25°C, min/max are from T_A = -40°C to +85°C, unless otherwise noted.) (Note 1)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
SCL Pulse-Width High	t_{HIGH}		0.6			µs
Setup Time for a Repeated START Condition	t_{SU_STA}		0.6			µs
Data Hold Time	t_{HD_DAT}		0		900	ns
Data Setup Time	t_{SU_DAT}		100			ns
Setup Time for STOP Condition	t_{SU_STO}		0.6			µs
Pulse Width of Suppressed Spike	t_{SP}		0		50	ns
Bus Capacitance	CB				400	pF
SDA and SCL Receiving Rise Time	t_R			20 + 0.1CB	300	ns
SDA and SCL Receiving Fall Time	t_F			20 + 0.1CB	300	ns
SDA Transmitting Fall Time	t_{TF}			20 + 0.1CB	300	ns
GPIO External Sync Low Pulse Width	t_{PLGPIO}	GPIO_CTRL = 10	5			µs
		GPIO_CTRL = 2	64			µs

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

Note 2: V_{LED} should be set to accommodate the maximum LED forward voltage and the output compliance of the LED driver.

Note 3: The LED current is trim in production to meet the IR, GREEN, and RED ADC counts. Actual values can vary by up to ±50%. Values shown here are for 0% trim.

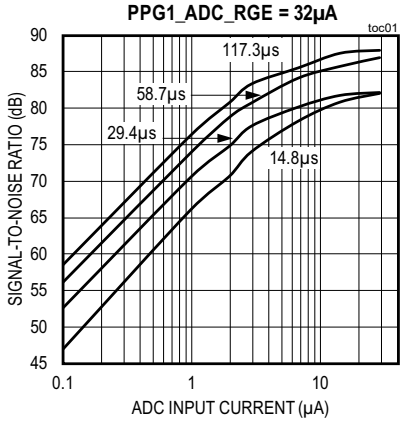
Note 4: For design guidance only. Not production tested. Tested in die form only.

Note 5: For design guidance only. Not production tested.

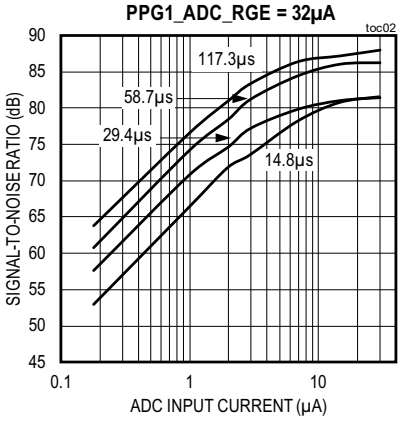
Typical Operating Characteristics

(V_{LED} = 5.0V, GND = PGND = 0V, T_A = +25°C, unless otherwise noted)

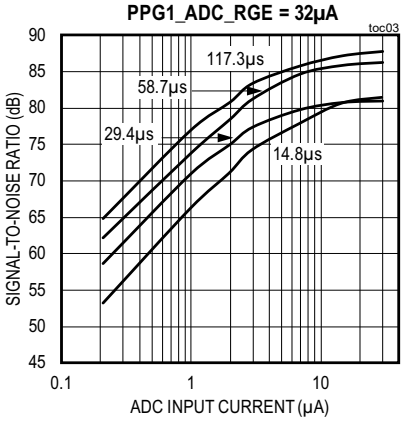
LED1 - 530nm
SIGNAL-TO-NOISE RATIO vs. ADC INPUT CURRENT
PPG1_ADC_RGE = 32µA



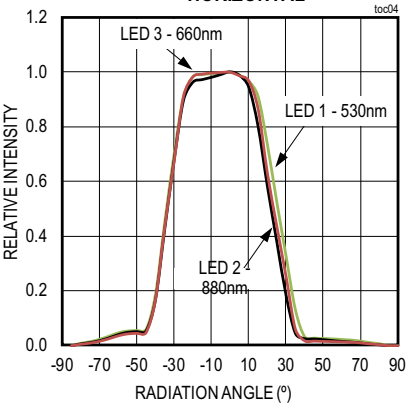
LED2 - 880nm
SIGNAL-TO-NOISE RATIO vs. ADC INPUT CURRENT
PPG1_ADC_RGE = 32µA



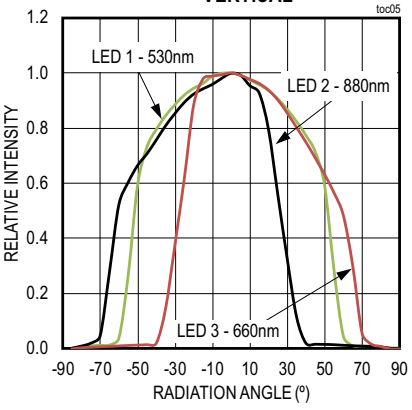
LED3 - 660nm
SIGNAL-TO-NOISE RATIO vs. ADC INPUT CURRENT
PPG1_ADC_RGE = 32µA



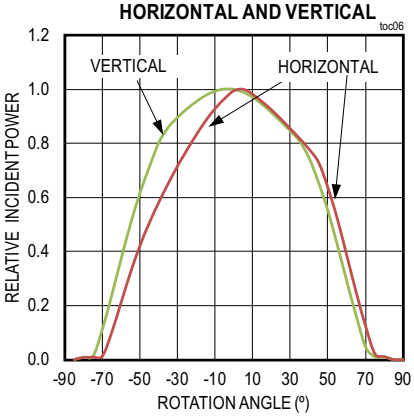
LED 2D - RADIATION PATTERN
HORIZONTAL



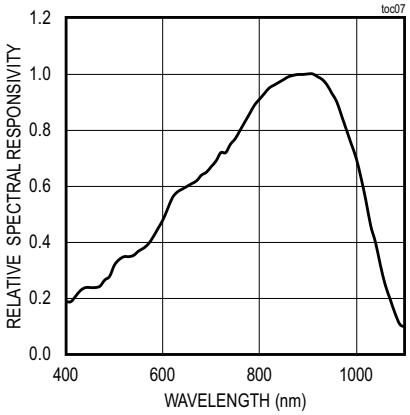
LED 2D - RADIATION PATTERN
VERTICAL



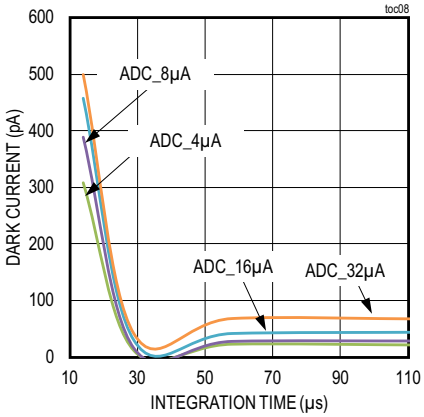
PHOTODIODE FIELD OF VIEW
HORIZONTAL AND VERTICAL



PHOTODIODE RELATIVE
SPECTRAL RESPONSIVITY



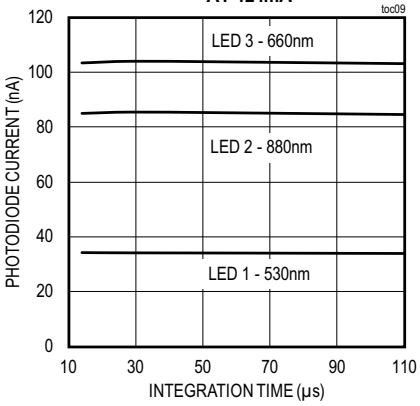
PHOTODIODE DARK CURRENT



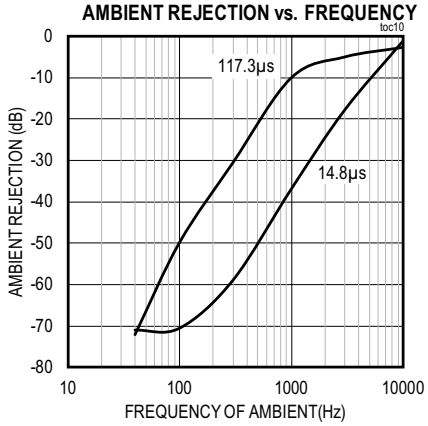
Typical Operating Characteristics (continued)

(V_{LED} = 5.0V, GND = PGND = 0V, T_A = +25°C, unless otherwise noted)

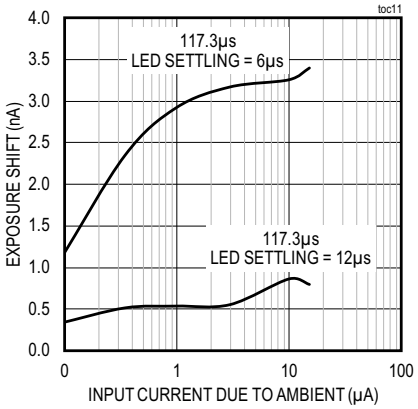
LED-PHOTODIODE CROSSTALK AT 124mA



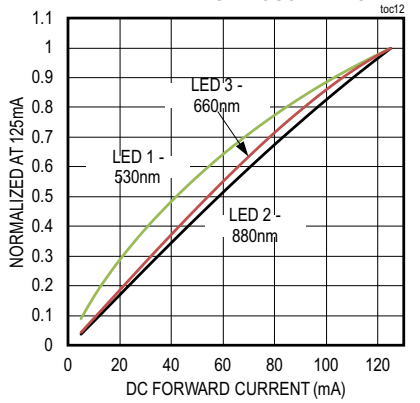
AMBIENT REJECTION vs. FREQUENCY



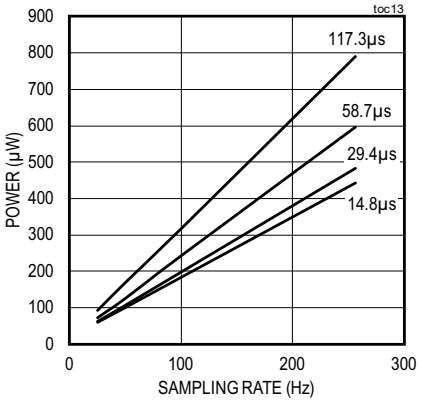
AMBIENT LIGHT REJECTION EXPOSURE SHIFT vs. AMBIENT INPUT



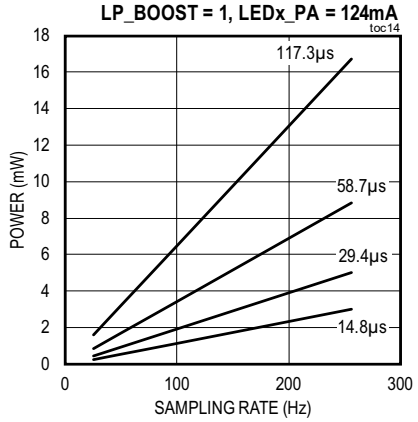
LED LINEARITY RELATIVE LUMINOUS INTENSITY



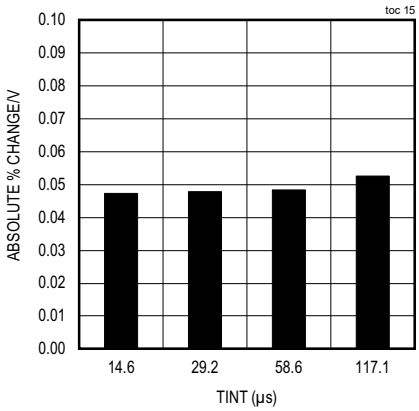
AFE POWER DISSIPATION LP_BOOST = 1, GPIO MODE = 0



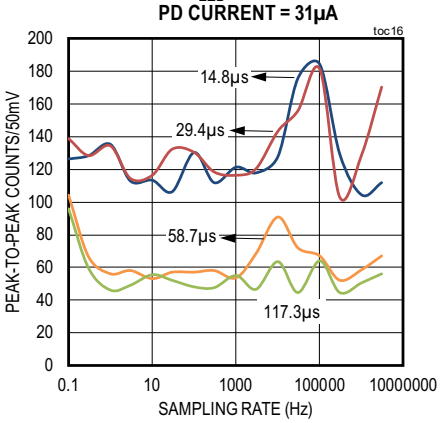
SYSTEM POWER DISSIPATION WITH 1 LED LP_BOOST = 1, LEDx_PA = 124mA



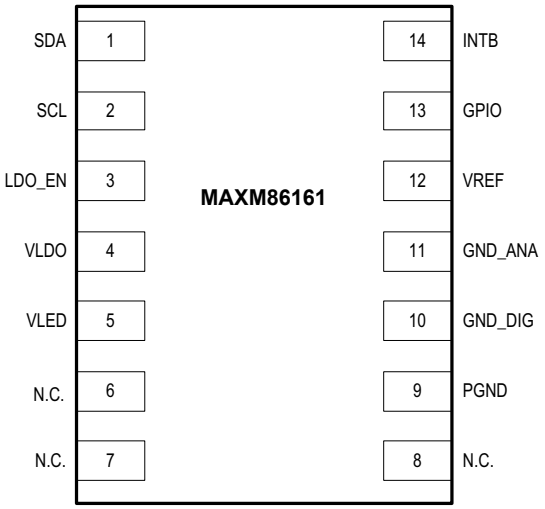
V_{LED} DC PSRR PD CURRENT = 31µA



V_{LED} AC PSRR PD CURRENT = 31µA



Pin Configuration



Pin Description

PIN	NAME	FUNCTION
POWER		
5	V_{LED}	LED Power Supply Input. Connect to external voltage supply. Bypass with a 10µF capacitor to PGND.
9	PGND	LED Power Return. Connect to GND.
10	GND_DIG	Digital Logic and Digital Pad Return. Connect to GND_ANA.
11	GND_ANA	Analog Power Return. Connect to GND.
CONTROL INTERFACE		
1	SDA	SDA Input/Output. Input/output for I^2C data.
2	SCL	SCL Input. I^2C clock.
3	LDO_EN	LDO Enable Input. Pull HIGH to turn on the internal LDO. Pull LOW to turn off the internal LDO. When pulled LOW, part in shut down.
13	GPIO	General Purpose Input/Output. Open-drain when programmed as output (active-low).
14	INTB	Open Drain Interrupt.
REFERENCE		
4	VLDO	Internal LDO output. Bypass with a 1µF capacitor to GND_ANA.
12	VREF	Internal Reference Decoupling Point. Bypass with a 1µF capacitor to GND_ANA.
LED DRIVER		
6	N.C.	No Connection. Internally connected to LEDx, Solder to PCB for mechanical stability.
7		
8		

Detailed Description

The MAXM86161 is a complete, integrated, optical data acquisition system, ideal for optical pulse oximetry and heart-rate detection applications. It has been designed for the demanding requirements of mobile and wearable devices, requiring minimal external hardware components to be integrated into a wearable device. The MAXM86161 includes 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 module is fully adjustable through software registers and the digital output data is stored in a 128-word FIFO within the IC. The FIFO allows the MAXM86161 to be connected to a microcontroller or processor on a shared bus where the data is not being read continuously from the MAXM86161 registers. It operates in fully autonomous modes for low-power battery applications.

The MAXM86161 consists of a single optical readout channel. MAXM86161 has three LED drivers and is well suited for a wide variety of optical sensing applications.

The module operates on a 3.0V to 5.5V V_{LED} single supply voltage. It has flexible timing and shutdown 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 MAXM86161 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 photo diode currents, allowing the sensor to work in high ambient light conditions. The optical ADC has programmable full-scale ranges of 4µA to 32µA. 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. The MAXM86161 includes a proprietary discrete time filter to reject 50Hz/60Hz interference and changing residual ambient light from the sensor measurements.

The MAXM86161 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 256sps and below. For more details on the power consumption at each sample rate, please refer to the *Electrical Characteristics* table.

LED Driver

The MAXM86161 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, minimizing LED power consumption. The LED pulse width can be programmed from 14.8µs to 117.3µs to allow the algorithms to optimize SpO_2 and HR accuracy at the lowest dynamic power consumption dictated by the application.

FIFO Configuration

The FIFO has 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.

Index to the data within a sample identifies the input to the PPG channels, and follows the order in the LED Sequence Control registers (Table 1).

LED Sequence Control (0x20 to 0x22)

The data format in the FIFO as well as the sequencing of exposures are controlled by the LED Sequence Registers using LEDC1 through LEDC6. There are six LED Sequence Data Types available as shown in Table 2. 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	REGISTER NAME	DEFAULT 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 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).

There are seven registers that control how the FIFO is configured and read out. These registers are illustrated in Table 4.

Table 2. LED Sequence Register Data Type

LEDCN[3:0]; (N = 1 – 6)	DATA TYPE
0000	NONE
0001	LED1
0010	LED2
0011	LED3
0100	Reserved
0101	Reserved
0110	Reserved
0111	Reserved
1000	Pilot on LED1
1001	DIRECT AMBIENT
1010	Reserved
1011	Reserved
1100	Reserved
1101	Reserved
1110	Reserved
1111	Reserved

Table 3. FIFO Data, Tag and Sample Counter Format

FIFO_DATA[23:19]; TAG]	FIFO_DATA[18:0]; DATA TYPE	COMMENTS
00001	PPG1_LEDC1_DATA	If LEDC1 is nonzero
00010	PPG1_LEDC2_DATA	If LEDC1 and LEDC2 are nonzero
00011	PPG1_LEDC3_DATA	If LEDC1, LEDC2, and LEDC3 are nonzero
00100	PPG1_LEDC4_DATA	If LEDC1, LEDC2, LEDC3, and LEDC4 are nonzero
00101	PPG1_LEDC5_DATA	If LEDC1, LEDC2, LEDC3, LEDC4, and LEDC5 are nonzero
00110	PPG1_LEDC6_DATA	If LEDC1, LEDC2, LEDC3, LEDC4, LEDC5, and LEDC6 are nonzero
00111	PPG2_LEDC1_DATA	PPG2 not available
01000	PPG2_LEDC2_DATA	PPG2 not available
01001	PPG2_LEDC3_DATA	PPG2 not available
01010	PPG2_LEDC4_DATA	PPG2 not available
01011	PPG2_LEDC5_DATA	PPG2 not available
01100	PPG2_LEDC6_DATA	PPG2 not available

Table 3. FIFO Data, Tag and Sample Counter Format (continued)

FIFO_DATA[23:19]; TAG]	FIFO_DATA[18:0]; DATA TYPE	COMMENTS
01101	PPF1_LEDC1_DATA	If LEDC1 is nonzero (Picket Fence event)
01110	PPF1_LEDC2_DATA	If LEDC1 and LEDC2 are nonzero (Picket Fence event)
01111	PPF1_LEDC3_DATA	If LEDC1, LEDC2, and LEDC3 are nonzero (Picket Fence event)
10000	Reserved	
10001	Reserved	
10010	Reserved	
10011	PPF2_LEDC1_DATA	PPG2 not available
10100	PPF2_LEDC2_DATA	PPG2 not available
10101	PPF2_LEDC3_DATA	PPG2 not available
10110	Reserved	
10111	Reserved	
11000	Reserved	
11001	PROX1_DATA	Only PILOT LED1 for LEDC1 is used
11010	PROX2_DATA	PPG2 not available
11011	Reserved	
11100	Reserved	
11101	Sub-DAC Update	This tag indicates that the sub-DAC had updated on this conversion
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

Table 4. FIFO configuration

ADDRESS	REGISTER NAME	DEFAULT VALUE	B7	B6	B5	B4	B3	B2	B1	B0
0X04	FIFO Write Pointer	00	—				FIFO_WR_PTR[6:0]			
0X05	FIFO Read Pointer	00	—				FIFO_RD_PTR[6:0]			
0X06	Overflow Counter	00	—				OVF_COUNTER[6:0]			
0X07	FIFO Data Counter	00	FIFO_DATA_COUNT[7:0]							
0X08	FIFO Data Register	00	FIFO_DATA[7:0]							
0X09	FIFO Configuration 1	00	—	FIFO_A_FULL[6:0]						
0X0A	FIFO Configuration 2	00	—	—	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 is 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 wraps 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 is read using 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 wraps 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 using 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 the data. The readout from the FIFO follows a progression defined by the LED Sequence Control registers as well. This configuration is best illustrated by a few examples.

Assume it is desired to perform an SpO_2 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 = 0x2	LED2 exposure
LEDC2 = 0x3	LED3 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
PPG_TINT[1:0]	LED Pulse Width Control
PPG_SR[3:0]	Sample Rate
LED PULSE AMPLITUDE	
LED2_PA[7:0]	LED2 Drive Current
LED3_PA[7:0]	LED3 Drive Current

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

tag 1, PPG1 LED2 data

tag 2, PPG1 LED3 data

tag 3, PPG1 Ambient data

tag 1, PPG1 LED2 data

tag 2, PPG1 LED3 data

tag 3, PPG1 Ambient data

.

.

.

tag 1, PPG1 LED2 data

tag 2, PPG1 LED3 data

tag 3, PPG1 Ambient data

where,

PPGm LED1 data = Ambient corrected exposure data from LED1 in PPGm channel,

PPGm LED2 data = Ambient corrected exposure data from LED2 in PPGm channel

PPGm Ambient data = Direct ambient sample in PPGm channel

m = 1 of PPG1 channel

To calculate the number of available items when the $\overline{INT}$ signal is seen, 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 5. Bits F23:F19 of the FIFO word contains the tag that identifies the data.

FIFO_A_FULL (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 is 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 $\overline{INT}$ 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 are read.

FIFO_RO (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 (0x0A)

The A_FIFO_TYPE bit defines the behavior of the A_FULL interrupt. If the A_FIFO_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 the A_FIFO_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 the Interrupt Status 1 register read, and does not reassert for every sample until a new A_FULL condition is detected.

FIFO_STAT_CLR (0x0A)

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

FLUSH_FIFO (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 (0x0A)

When the 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 time stamp is useful for aligning data from two devices after the host reads the FIFOs of those devices. When the TIME_STAMP_EN bit is set to 0, the sample counter is not pushed to FIFO.

Table 5. Optical FIFO Data Format

FIFO DATA FORMAT (FIFO_DATA[23:0])																							
TAG (TAG[4:0])					ADC VALUE (FIFO_DATA[18:0])																		
F23	F22	F21	F20	F19	F18	F17	F16	F15	F14	F13	F12	F11	F10	F9	F8	F7	F6	F5	F4	F3	F2	F1	F0
T4	T3	T2	T1	T0	O18	O17	O16	O15	O14	O13	O12	O11	O10	O9	O8	O7	O6	O5	O4	O3	O2	O1	O0

Table 6. Slave Address

	WRITE	READ
I2C Slave Address	0xC4	0xC5

Pseudo-Code Example of Initialize the Optical AFE

The following pseudo-code shows an example of configuring MAXM86161 for an SpO_2 applications, where LED2 and LED3 are IR and red LED, respectively.

```
DEVICE OPEN
START;
                                        // AFE Initialization
WRITE RESET[0] to 0x1;                 // Soft Reset (Register 0x0D[0])
DELAY 1ms;
WRITE SHDN[0] to 0x1;                  // Shutdown (Register 0x0D[1])
READ  Interrupt_Status_1;              // Clear Interrupt (Register 0x00)
READ  Interrupt_Status_2;              // Clear Interrupt (Register 0x01)
WRITE PPG_TINT[1:0] to 0x3;            // Pulse Width = 123.8ms (Register 0x11[1:0])
WRITE PPG1_PPG1_ADC_RGE1:0] to 0x2;       // ADC Range = 16µA (Register 0x11[3:2])

WRITE SMP_AVE[2:0] to 0x0;             // Sample Averaging = 1 (Register 0x12[2:0])
WRITE PPG_SR[4:0] to 0x00;             // Sample Rate = 25sps (Register 0x12[7:3])
WRITE LED_SETLNG[1:0] to 0x3;         // LED Settling Time = 12ms  (Register 0x13[7:6])
WRITE PDBIAS11[2:0] to 0x01;         // PD 1 Biasing for Cpd = 0~65pF (Register 0x15[2:0])

WRITE LED1_RGE[1:0] to 0x3;            // LED Driver 1 Range = 124mA (Register 0x15[2:0])
WRITE LED2_RGE[1:0] to 0x3;            // LED Driver 2 Range = 124mA (Register 0x15[2:0])
WRITE LED1_DRV[1:0] to 0x20;         // LED 1 Drive Current = 15.36mA (Register 0x23[7:0])
WRITE LED2_DRV[1:0] to 0x20;         // LED 2 Drive Current = 15.36mA (Register 0x24[7:0])
WRITE LP_Mode[0] to 0x1;               // Low Power mode enabled
                                        // FIFO Configuration
WRITE FIFO_A_FULL[6:0] to 0xF;       // FIFO INT triggered condition (Register 0x09[6:0])
WRITE FIFO_RO to 0x1;                  // FIFO Roll Over enabled (Register 0x0A[1])
WRITE A_FULL_EN to 0x1;               // FIFO_A_FULL interrupt enabled (Register 0x02[7])
WRITE LEDC1[3:0] to 0x2;               // LED2 exposure configured in time slot 1
WRITE LEDC2[3:0] to 0x3;               // LED3 exposure configured in time slot 2
WRITE LEDC3[3:0] to 0x0;
WRITE LEDC4[3:0] to 0x0;
WRITE LEDC5[3:0] to 0x0;
WRITE LEDC6[3:0] to 0x0;
WRITE SHDN[0] to 0x0;                  // 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 the A_FULL Interrupt.

```
Interrupt handlervoid 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 a single photodiode channel and two LED channels.

```
void device_data_read(void) {
     uint8_t sampleCnt;
     uint8_t regVal;
     uint8_t dataBuf[128*2*3];       //128 FIFO samples, 2 channel, 3 byte/channel
     int led1[32];
     int led2[32];
     ReadReg(0x07, &sampleCnt);
     //Read FIFO
     ReadFifo(dataBuf, sampleCnt * 3);
     int i = 0;
     for ( i = 0; i < sampleCnt; i++ ) {
          led1[i] = ((dataBuf[i*6+0] << 16 ) | (dataBuf[i*6+1] << 8) | (dataBuf[i*6+2]))
& 0x7ffff;
          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.

```
void device_data_read(void) {
     uint8_t sampleCnt;
     uint8_t regVal;
     uint8_t dataBuf[128*2*2*3];      //128 FIFO samples, 2 channel, 2 PD, 3 byte/channel
     int led1A[32];
     int led1B[32];
     int led2A[32];
     int led2B[32];
     ReadReg(0x07, &sampleCnt);
```

```
    //Read FIFO
    ReadFifo(dataBuf, sampleCnt * 3);
    int i = 0;
    for ( i = 0; i < sampleCnt; i++ ) {
       led1A[i] = ((dataBuf[i*12+0] << 16 ) | (dataBuf[i*12+1] << 8)  | (dataBuf[i*12+2]))
& 0x7ffff;     // LED1, PD1
       led1B[i] = ((dataBuf[i*12+3] << 16 ) | (dataBuf[i*12+4] << 8)  | (dataBuf[i*12+5]))
& 0x7ffff;     // LED1, PD2
       led2A[i] = ((dataBuf[i*12+6] << 16 ) | (dataBuf[i*12+7] << 8)  | (dataBuf[i*12+8]))
& 0x7ffff;     // LED2, PD1
      led2B[i] = ((dataBuf[i*12+9] << 16 ) | (dataBuf[i*12+10] << 8) | (dataBuf[i*12+11]))
& 0x7ffff;    // LED2, PD2
  }
 }
```

Optical Timing

The MAXM86161 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 pulse oximetry measurements or simultaneously to drive multiple LEDs such 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 appears successively in the FIFO.

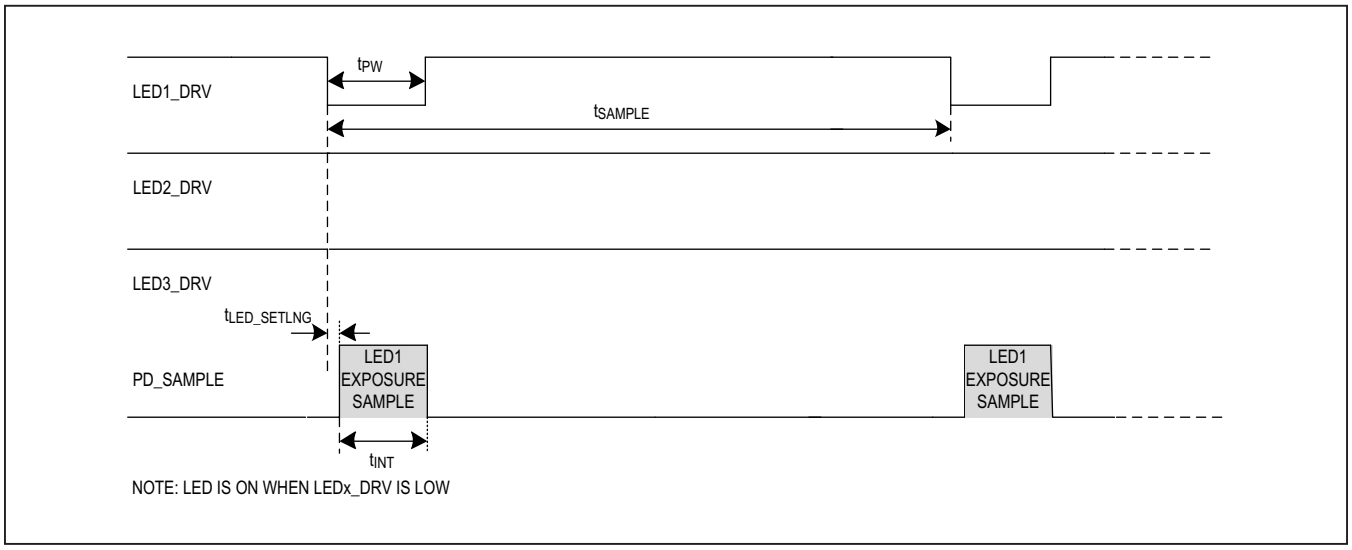


Figure 1. Timing for LED1 Pulsing with No Direct Ambient Sampling

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 appears successively in the FIFO.

Two LEDs Pulsing Sequentially with Direct Ambient Sampling

The timing diagram below illustrates the optical timing when both LED1 and LED2 are enabled to pulse sequentially. Direct ambient sampling is also enabled. This timing mode would be used when SpO_2 is being measured with IR and red LEDs. When SpO_2 is being measured with IR and red LEDs, the optical sampled value for each LED appears successively, followed by the direct ambient sampled value in the FIFO.

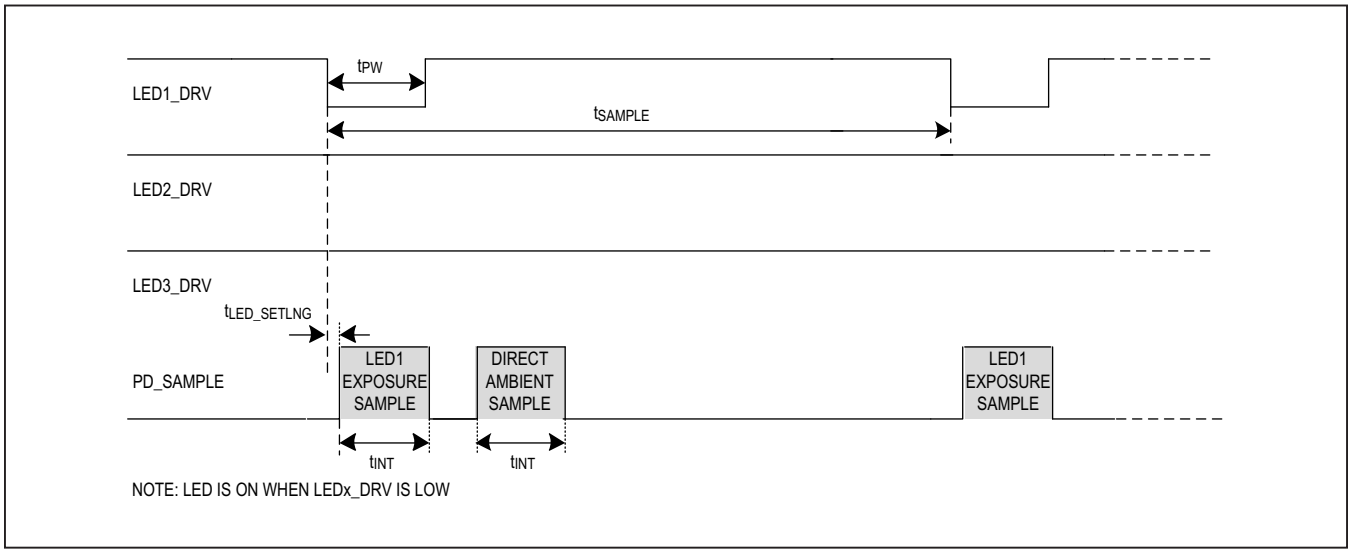


Figure 2. Timing for LED1 Pulsing with Direct Ambient Sampling

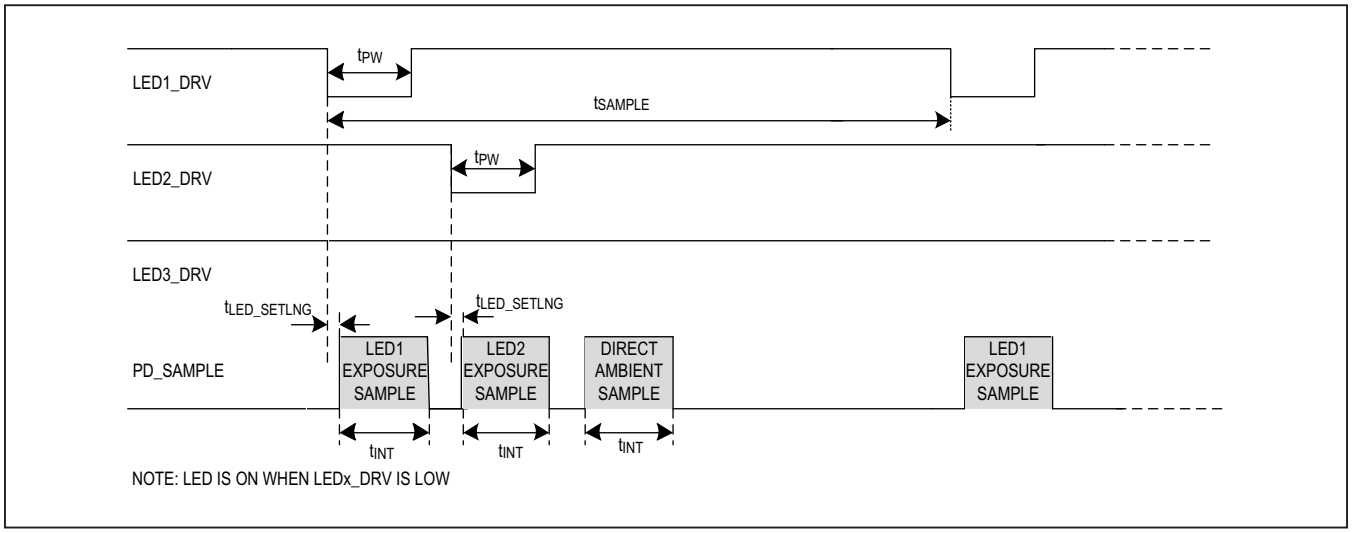


Figure 3. Timing for LED1 and LED2 Pulsing Sequentially with Direct Ambient Sampling

All LEDs Pulsing Sequentially 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 the heart rate on a green LED is combined with an SpO_2 measurement using IR and RED LEDs.

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

MAXM86161 is comprised of a 16-bit current integrating incremental delta-sigma analog-to-digital converter (ADC), wrapped in a 5-bit subranging digital-to-analog converter (DAC). The subranging 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 subranging DAC uses a unary architecture which has some mismatch between unit current sources of the DAC and the ADC reference current. This mismatch results in some transfer function nonlinearity (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 MAXM86161 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 and PPG_TINT bit fields in the PPG_Configuration1 register 0x11 to the values required for the intended application.
3) Set the START_CAL bit to one on the DAC Calibration Enable register 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 bits in the DAC Calibration Enable register to verify that self-calibration has 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 MAXM86161 measurement sequence.

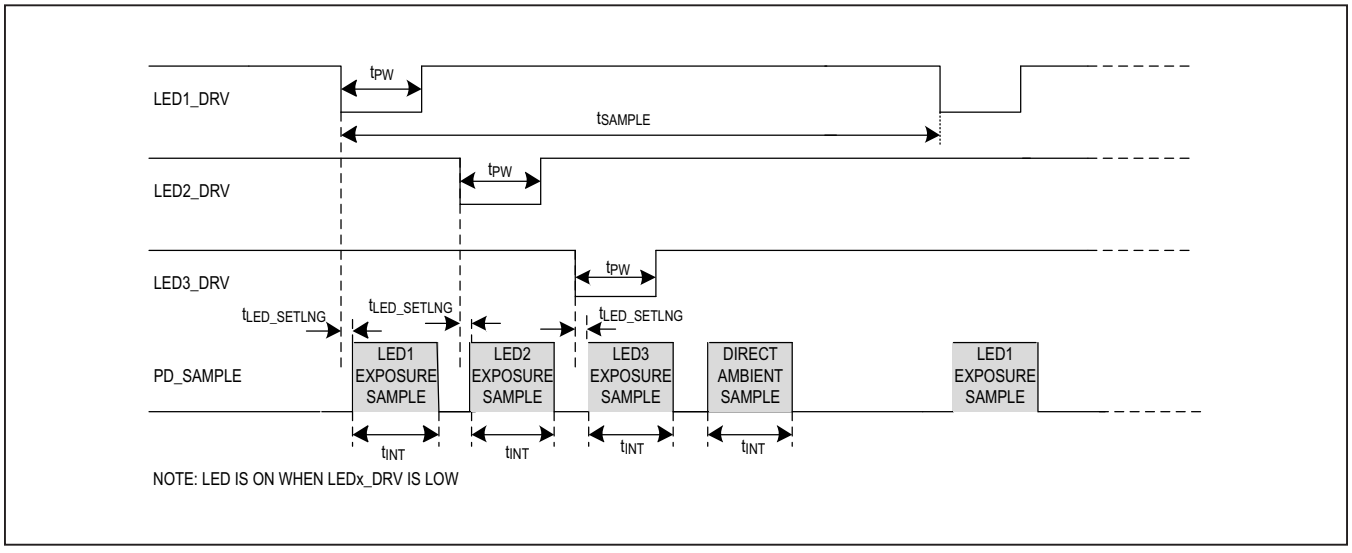


Figure 4. Timing for LED1, LED2, and LED3 Pulsing Sequentially with Direct Ambient Sampling

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 outputs a tag value of 0x1D on every conversion on which the sub-DAC transitions. This tag value overrides 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.

Proximity Mode Function

The MAXM86161 includes an optical proximity function which 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 the PROX_INT_EN bit field to 1 in the Interrupt Enable 2 register (0x02[4]), setting a threshold in the PROX_INT_THRESH register (0x14) and assigning an LED current in the PILOT_PA (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]) is asserted and proximity mode is entered when the value of the measurement assigned to LEDC1 drops below the PROX_INT_THRESH. When entering proximity mode, the MAXM86161 drops 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 is to both reduce the consumed LED current and MAXM86161 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 MAXM86161 also generates a Proximity Detect Interrupt (register 0x01[4]). In such an event, MAXM86161 switches to normal mode, changing the sample rate to that assigned in PPG Configuration 2 register (0x12) bit field PPG_SR and the LED current assigned to the measurement of LEDC1. Thus, the MAXM86161 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_PA in normal mode. This ensures 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 MAXM86161 successfully transitions 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 the PROX_INT_THRESH register are in units of 2048LSBs.

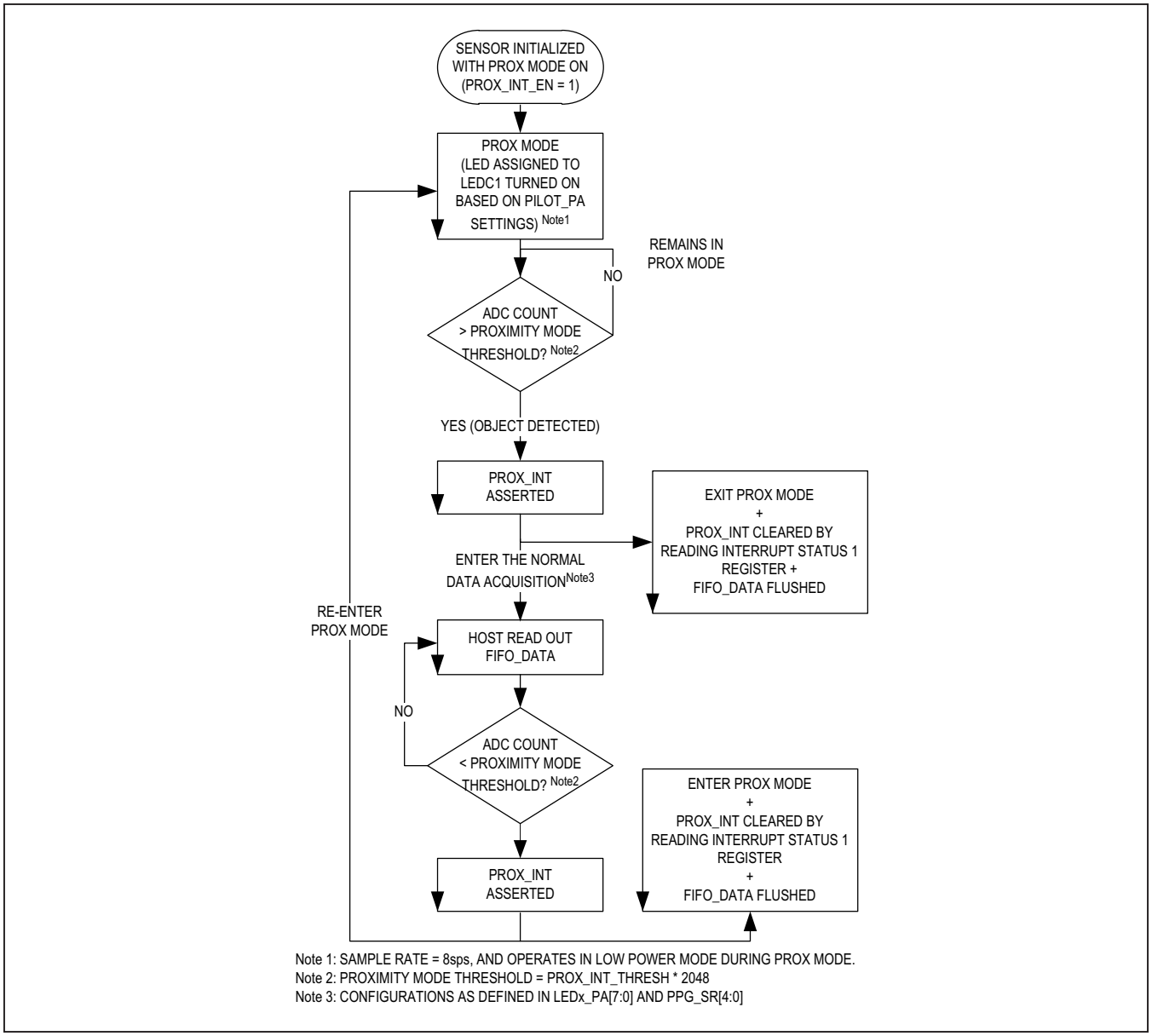


Figure 5. Proximity Function Flow Diagram

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 MAXM86161 ambient light correction (ALC) circuit to fail and produce an erroneous estimation of the ambient light during the exposure interval. The MAXM86161 has a built-in algorithm called the picket fence function that corrects the final PPG results in case of ALC circuit failure due to these extreme conditions.

The picket fence 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 function is enabled by setting PF_ENABLE (0x16[7]) bit to 1. The power-on reset default of MAXM86161 has the picket fence function disabled. 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 at 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 (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 passed version of the running estimation error computed above times a multiplier. The multiplier used is set by the THRESHOLD_SIGMA_MULT (0x16[1:0]) bits and can be 4, 8, 16, or 32 times the running low-passed filter output of the estimation error.

The low-pass filter function is controlled by two parameters, the IIR_TC (0x16[5:4]) bits and IIR_INIT_VALUE (0x16[3:2]) bits. The IIR_TC bits control the filters 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.

When a picket fence event is detected, the option of how to extrapolate the correct point is again controlled by the PF_ORDER bit. This point can be identical as the previous point (PF_ORDER = 0) or a least square fit extrapolation based on the previous four ADC converted points (PF_ORDER = 1).

Figure 6 illustrates the function in block diagram form. If the picket fence algorithm is enabled (bit PF_ENABLE = 1), 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).

This scheme essentially produces a threshold that tracks the past returned optical signal with a bandwidth based on the past historical change sample-to-sample. Figure 7 below illustrates graphically how the threshold detection scheme works on a real PPG signal. Note that the black trace is the real ADC sample point, the red traces are the output of the low-pass filter of the error estimation mirrored around the ADC points, and the blue traces are the threshold values.

The recommended settings for the picket fence algorithm are the default power on reset values for all registers but THRESHOLD_SIGMA_MULT bits. Here it is recommended that the 32x value 0x3 is 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 waveforms.

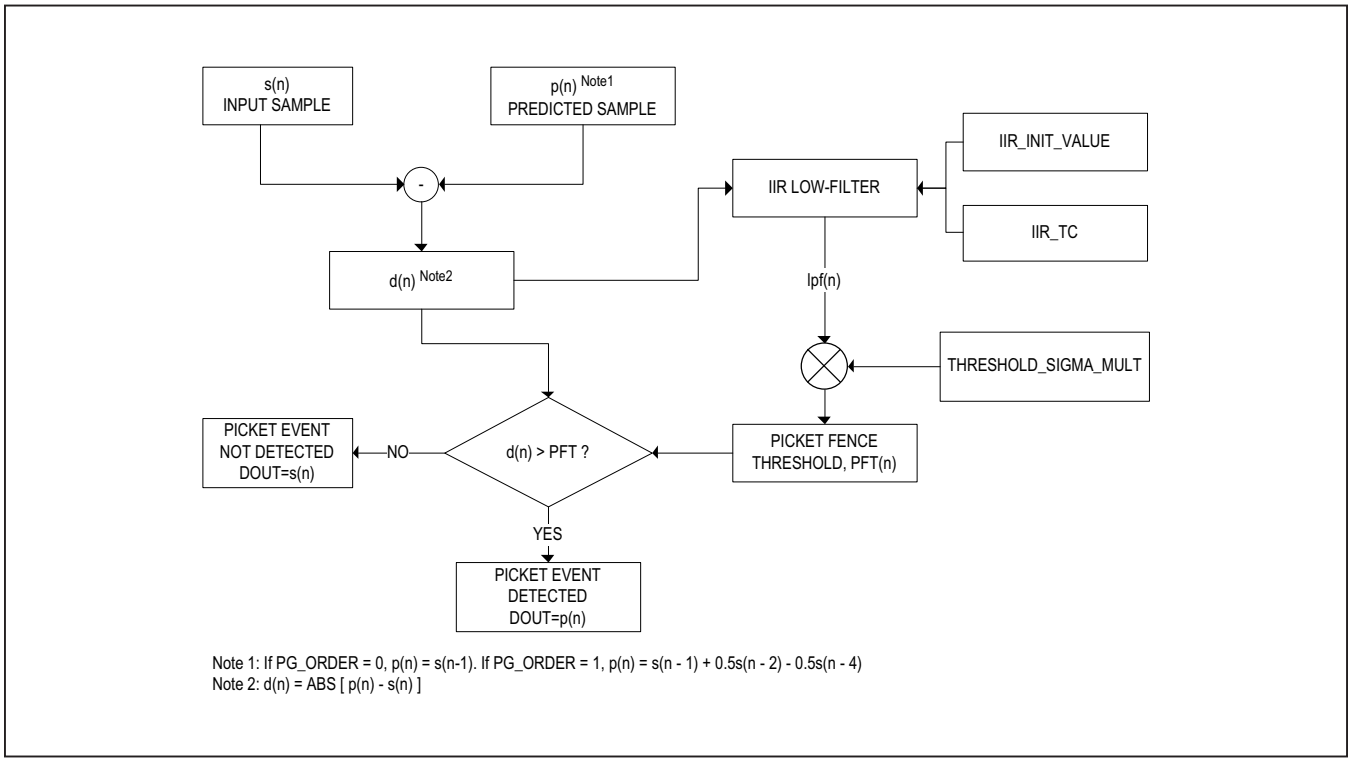


Figure 6. Picket Fence Function Flow

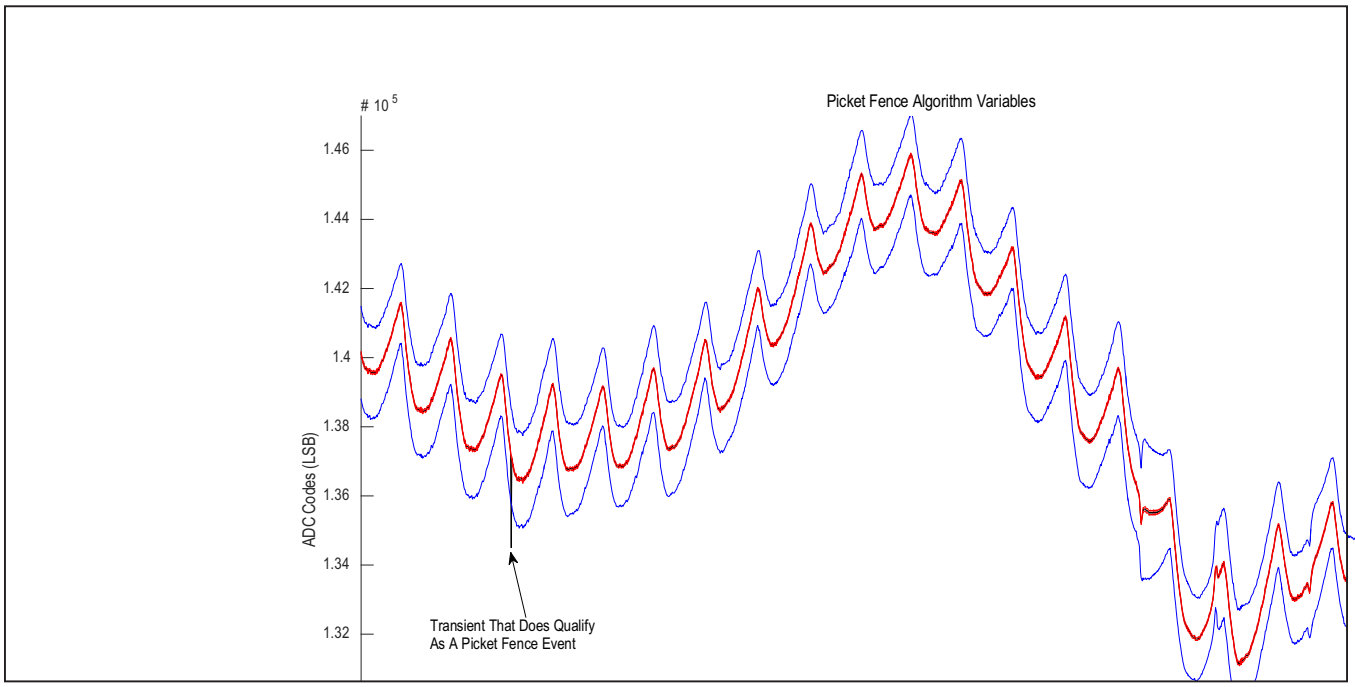


Figure 7. Picket Fences Variables In A PPG Waveform

Layout Guidelines

MAXM86161 is a high dynamic range analog front-end (AFE) and its performance can be adversely impacted by the physical printed circuit board (PCB) layout. See the below layout recommendations for detail.

1) All bypass capacitors should be placed as close to the part as possible.
2) All LEDx_DRV pins should be soldered down for mechanical stability (the Maxim layout example has three TPs for debug purposes only).
3) GND_ANA, GND_DIG, and PGND should be shorted to a single PCB GND plane. These three pins have been assigned along a single column so they can be easily shorted
4) VREF and VLDO pins should be decoupled to the PCB GND plane with a 1.0µF ceramic capacitor. The voltage on the VREF pin is nominally 1.21V and that for VLDO is nominally 1.8V, so a 6.3V rated ceramic capacitor should be adequate for this purpose.
5) Shield the vias with board ground plane.
6) Use short and low resistance trace for V_{LED} and all GND.
7) All decoupling capacitors use individual vias to the PCB GND plane to avoid coupling between decoupled supplies when sharing vias.

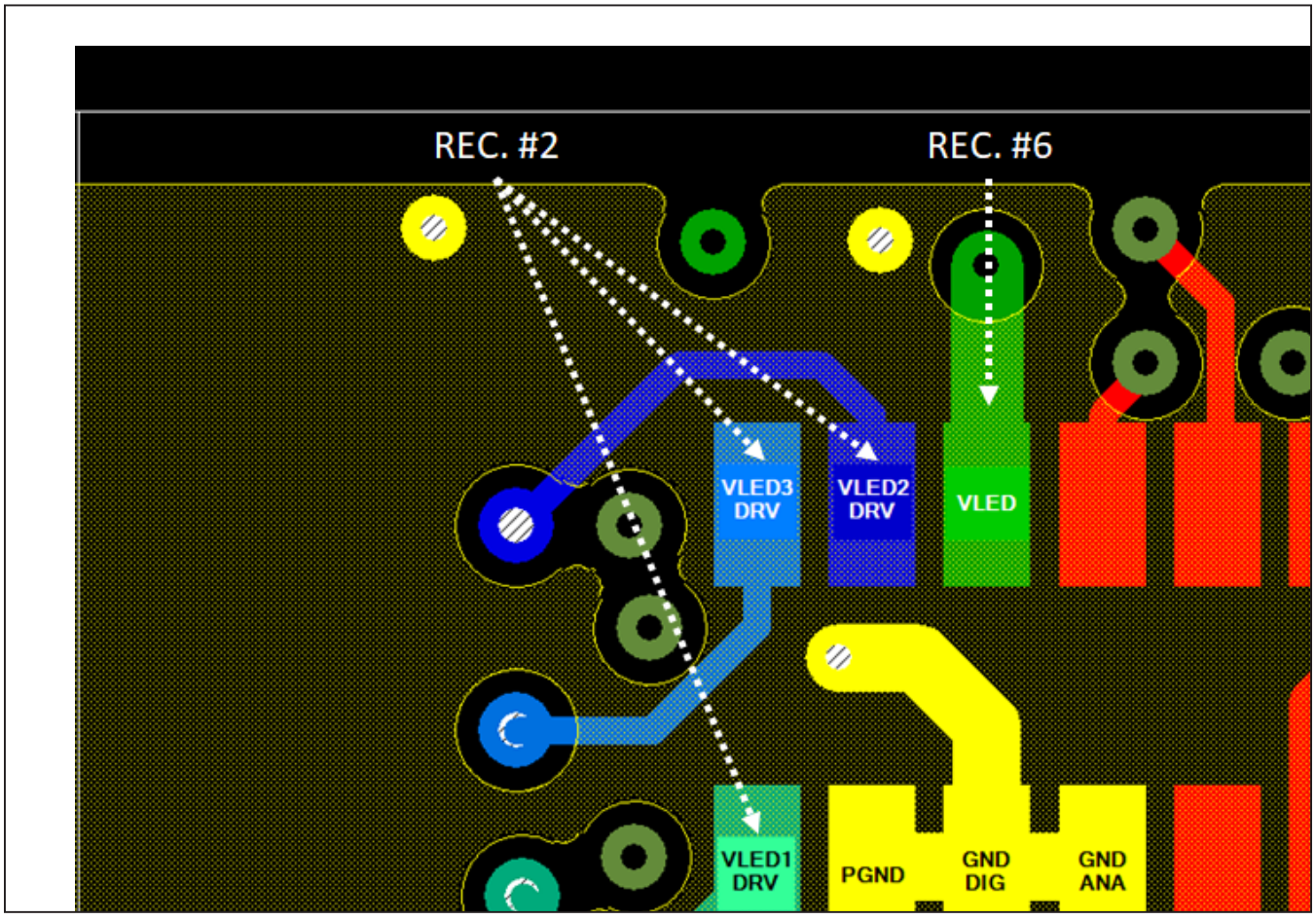


Figure 8. Layout Guideline

I²C/SMBus Compatible Serial Interface

The MAXM86161 features an I²C/SMBus compatible, 2-wire serial interface consisting of a serial data line (SDA) and a serial clock line (SCL). SDA and SCL facilitate communication between the MAXM86161 and the master at clock rates up to 400kHz.

Figure 9 shows the 2-wire interface timing diagram. The master generates SCL and initiates data transfer on the bus. The master device writes data to the MAXM86161 by transmitting the proper slave address followed by the register address and then the data word. Each transmit sequence is framed by a START (S) or REPEATED START (Sr) condition and a STOP (P) condition. Each word transmitted to the MAXM86161 is 8 bits long and is followed by an acknowledge clock pulse. A master reading data from the MAXM86161 transmits the proper slave address followed by a series of nine SCL pulses. The MAXM86161 transmits data on SDA in sync with the master-generated SCL pulses. The master acknowledges receipt of each byte of data. Each read sequence is framed by a START (S) or REPEATED START (Sr) condition, a not acknowledge, and a STOP (P) condition. SDA operates as both an input and an open-drain output. A pullup resistor is required on SDA. SCL operates only as an input. A pullup resistor is required on SCL if there are multiple masters on the bus, or if the single master has an open-drain SCL output. Series resistors in line with SDA and SCL are optional. Series resistors protect the digital inputs of the MAXM86161 from high voltage spikes on the bus lines, and minimize crosstalk and undershoot of the bus signals.

Detailed I²C Timing Diagram

The detailed timing diagram of various electrical characteristics is shown in Figure 9.

Bit Transfer

One data bit is transferred during each SCL cycle. The data on SDA must remain stable during the high period of the SCL pulse. Changes in SDA while SCL is high are control signals (see the START and STOP Conditions section).

START and STOP Conditions

SDA and SCL idle high when the bus is not in use. A master initiates communication by issuing a START condition. A START condition is a high-to-low transition on SDA with SCL high. A STOP condition is a low-to-high transition on SDA while SCL is high (Figure 10). A START condition from the master signals the beginning of a transmission to the MAXM86161. The master terminates transmission, and frees the bus, by issuing a STOP condition. The bus remains active if a REPEATED START condition is generated instead of a STOP condition.

Early STOP Conditions

The MAXM86161 recognizes a STOP condition at any point during data transmission except if the STOP condition occurs in the same high pulse as a START condition. For proper operation, do not send a STOP condition during the same SCL high pulse as the START condition.

Slave Address

The slave address is defined as the seven most significant bits (MSBs) followed by the read/write bit. For the MAXM86161 the seven most significant bits are 0b1100010X. Where X is determined by the read/write bit. For read mode, set the read/write bit to 1. For write mode, set the read/write bit to 0. The address is the first byte of information sent to the IC after the START condition.

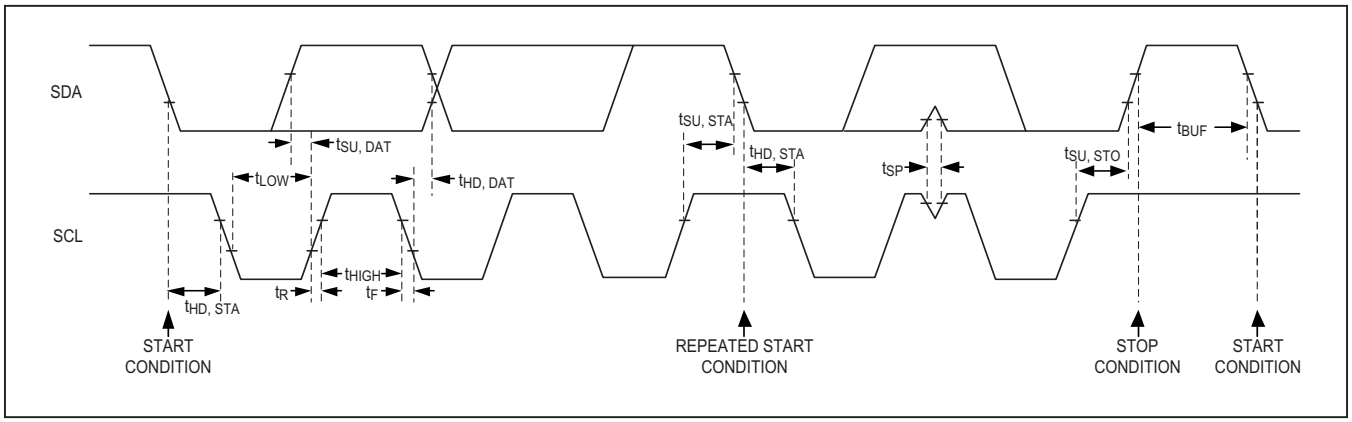


Figure 9. Detailed I²C Timing Diagram

Acknowledge Bit

The acknowledge bit (ACK) is a clocked 9th bit that the MAXM86161 uses to handshake receipt of each byte of data when in write mode (Figure 11). The MAXM86161 pulls down SDA during the entire master-generated 9th clock pulse if the previous byte is successfully received. Monitoring ACK allows for detection of unsuccessful data transfers. An unsuccessful data transfer occurs if a receiving device is busy or if a system fault has occurred. In the event of an unsuccessful data transfer, the bus master retries communication. The master pulls down SDA during the 9th clock cycle to acknowledge receipt of data when the MAXM86161 is in read mode. An acknowledge is sent by the master after each read byte to allow data transfer to continue. A not-acknowledge is sent when the master reads the final byte of data from the MAXM86161 followed by a STOP condition.

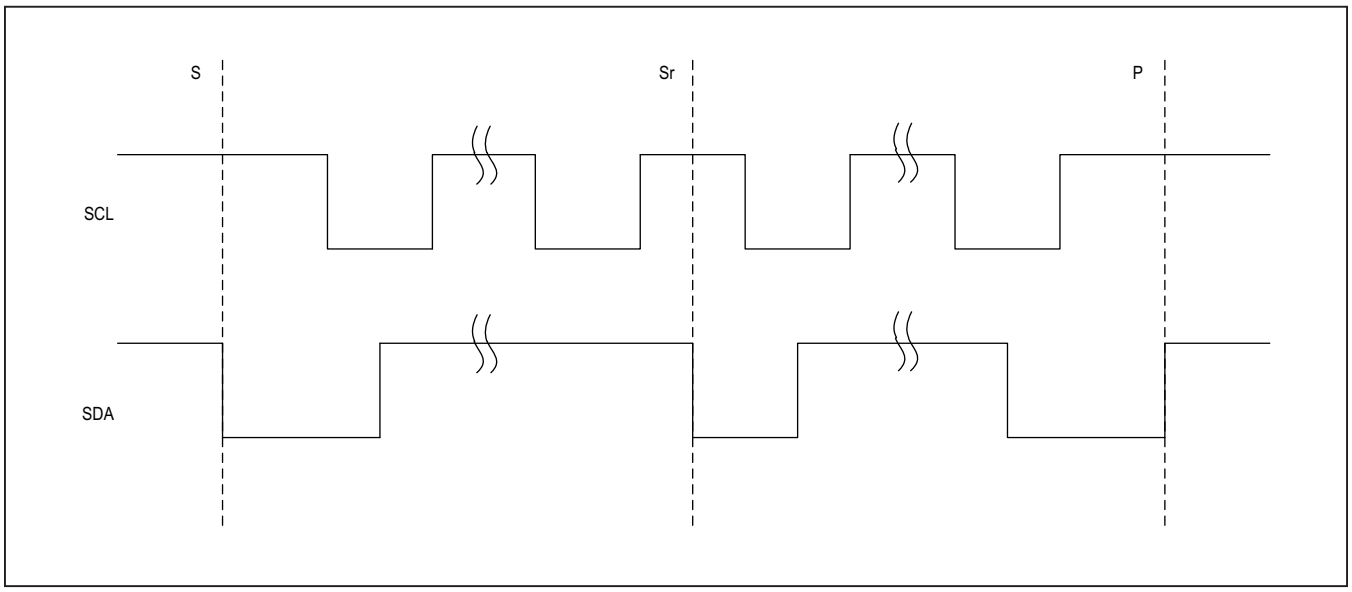


Figure 10. I²C START, STOP, and REPEATED START Conditions

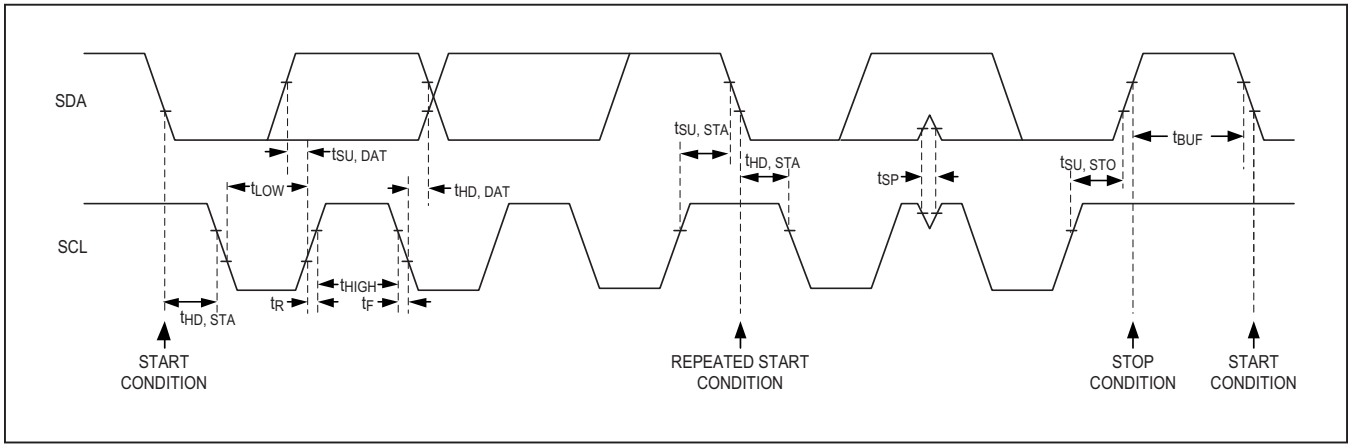


Figure 11. I²C Acknowledge Bit

I²C Write Data Format

A write to the MAXM86161 includes transmission of a START condition, the slave address with the R/$\overline{W}$ bit set to 0, one byte of data to configure the internal register address pointer, one or more bytes of data, and a STOP condition. Figure 12 illustrates the proper frame format for writing one byte of data to the MAXM86161. Figure 13 illustrates the frame format for writing n-bytes of data to the MAXM86161.

The slave address with the R/$\overline{W}$ bit set to 0 indicates that the master intends to write data to the MAXM86161. The MAXM86161 acknowledges receipt of the address byte during the master-generated 9th SCL pulse.

The second byte transmitted from the master configures the MAXM86161 internal register's address pointer. The pointer tells the MAXM86161 where to write the next byte of data. An acknowledge pulse is sent by the MAXM86161 upon receipt of the address pointer data.

The third byte sent to the MAXM86161 contains the data that is written to the chosen register. An acknowledge pulse from the MAXM86161 signals receipt of the data byte. The address pointer auto increments to the next register address after each received data byte. This auto-increment feature allows a master to write to sequential registers within one continuous frame. The master signals the end of transmission by issuing a STOP condition. The auto-increment feature is disabled when there is an attempt to write to the FIFO_DATA register.

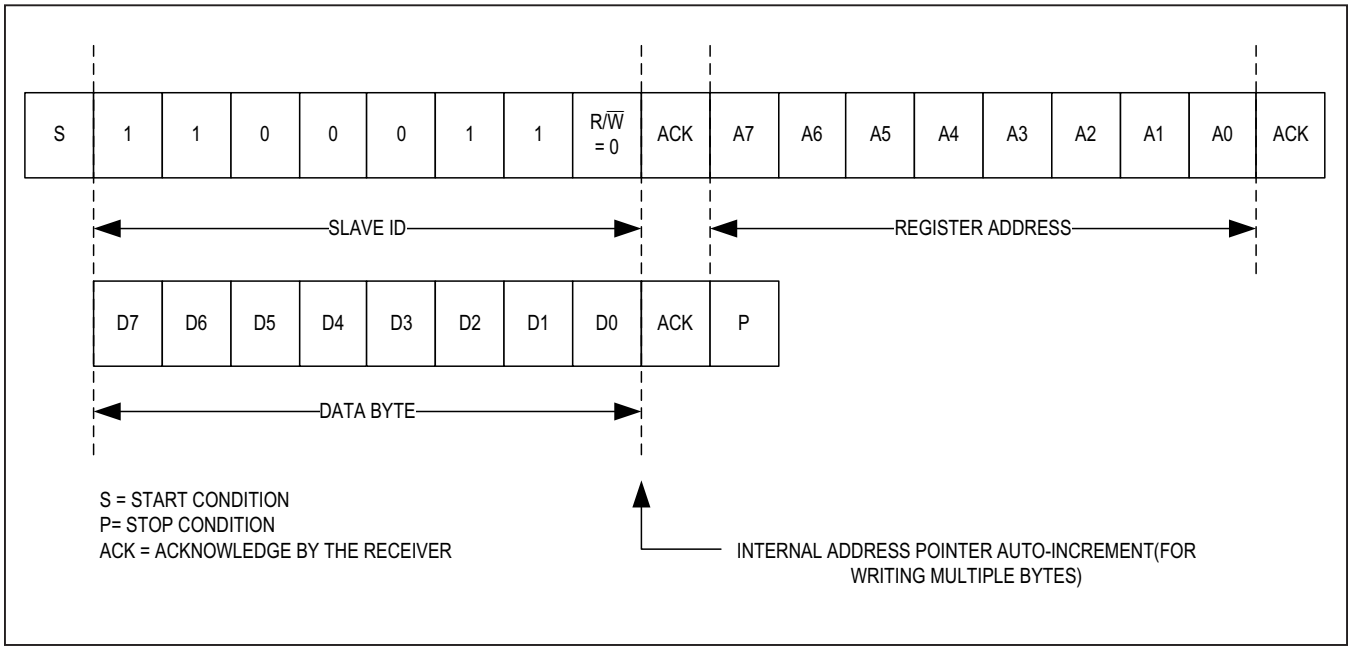


Figure 12. I²C Single Byte Write Transaction

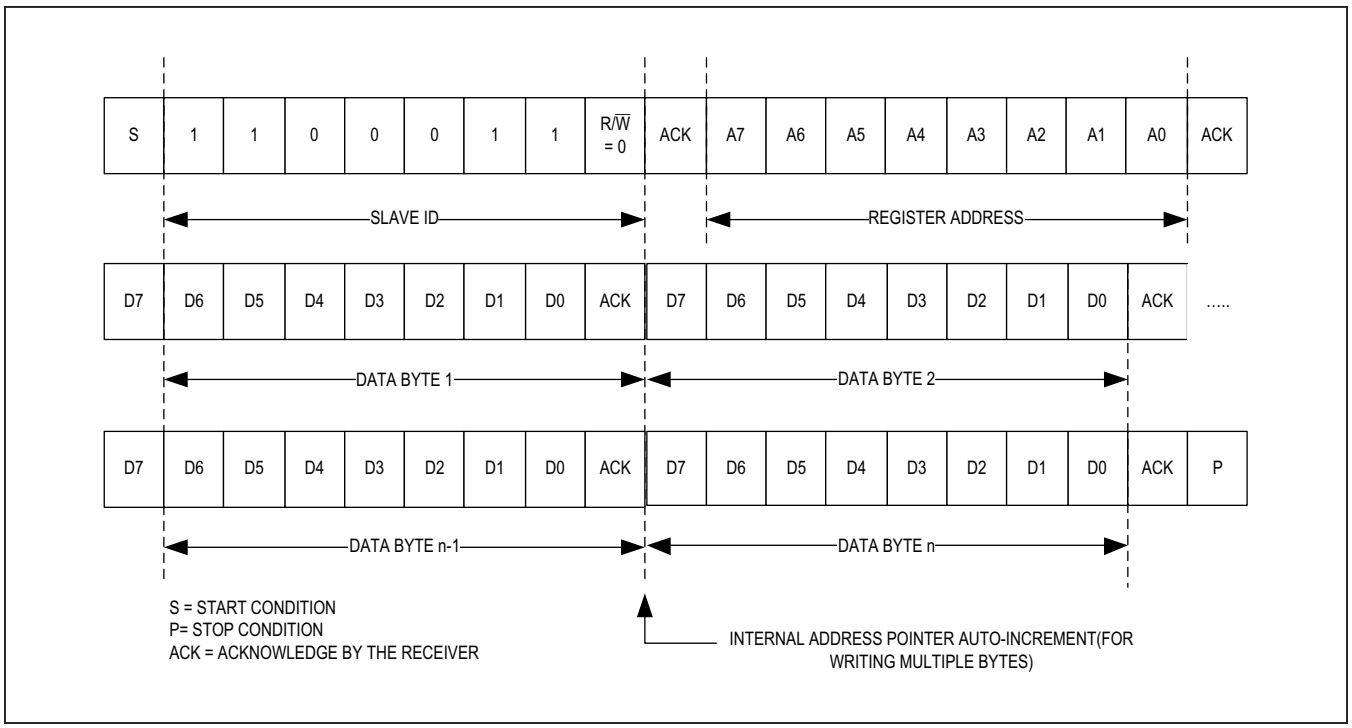


Figure 13. I²C Multi-Byte Write Transaction

I²C Read Data Format

Send the slave address with the R/$\overline{W}$ bit set to 1 to initiate a read operation. The MAXM86161 acknowledges receipt of its slave address by pulling SDA low during the 9th SCL clock pulse. A START command followed by a read command resets the address pointer to register 0x00.

The first byte transmitted from the MAXM86161 is the contents of register 0x00. Transmitted data is valid on the rising edge of SCL. The address pointer auto-increments after each read data byte. This auto-increment feature allows all registers to be read sequentially within one continuous frame. The auto-increment feature is disabled when there is an attempt to read from the FIFO_DATA register. A STOP condition can be issued after any number of read data bytes. If a STOP condition is issued followed by another read operation, the first data byte to be read is from register 0x00.

The address pointer can be preset to a specific register before a read command is issued. The master presets the address pointer by first sending the MAXM86161 slave address with the R/$\overline{W}$ bit set to 0 followed by the register address. A REPEATED START condition is then sent followed by the slave address with the R/$\overline{W}$ bit set to 1. The MAXM86161 then transmits the contents of the specified register. The address pointer auto-increments after transmitting the first byte.

The master acknowledges receipt of each read byte during the acknowledge clock pulse. The master must acknowledge all correctly received bytes except the last byte. The final byte must be followed by a not acknowledge from the master and then a STOP condition. Figure 14 illustrates the frame format for reading one byte from the MAXM86161. Figure 15 illustrates the frame format for reading multiple bytes from the MAXM86161.

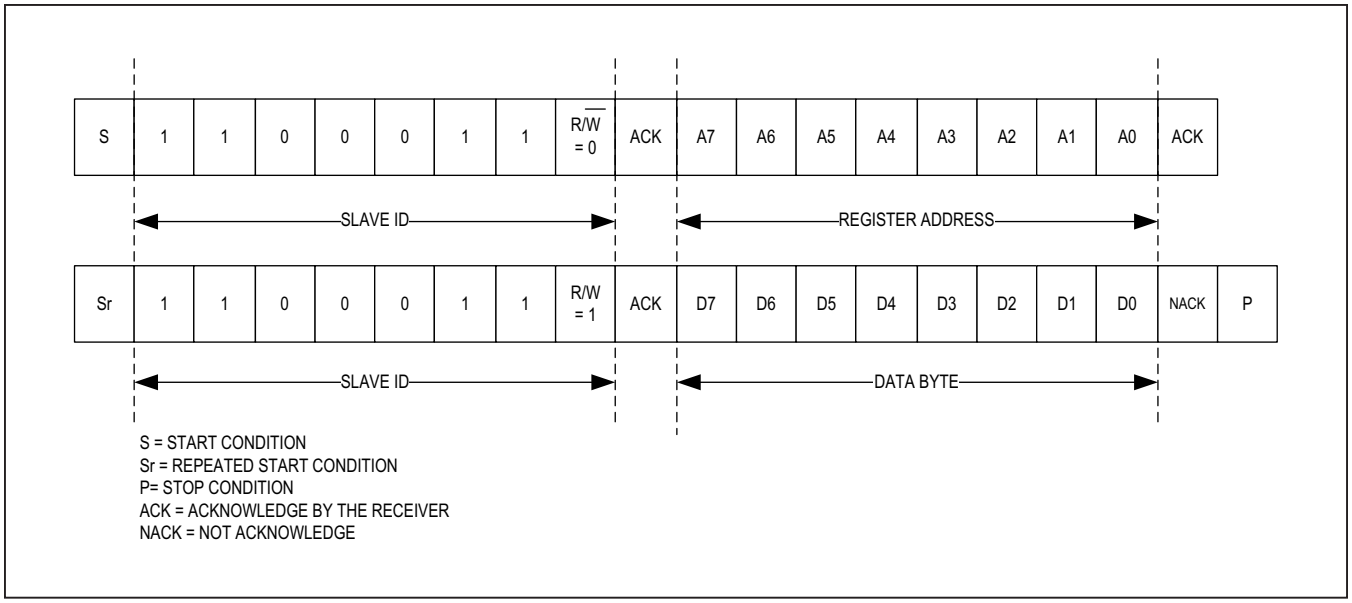


Figure 14. I^2C Single Byte Read Transaction

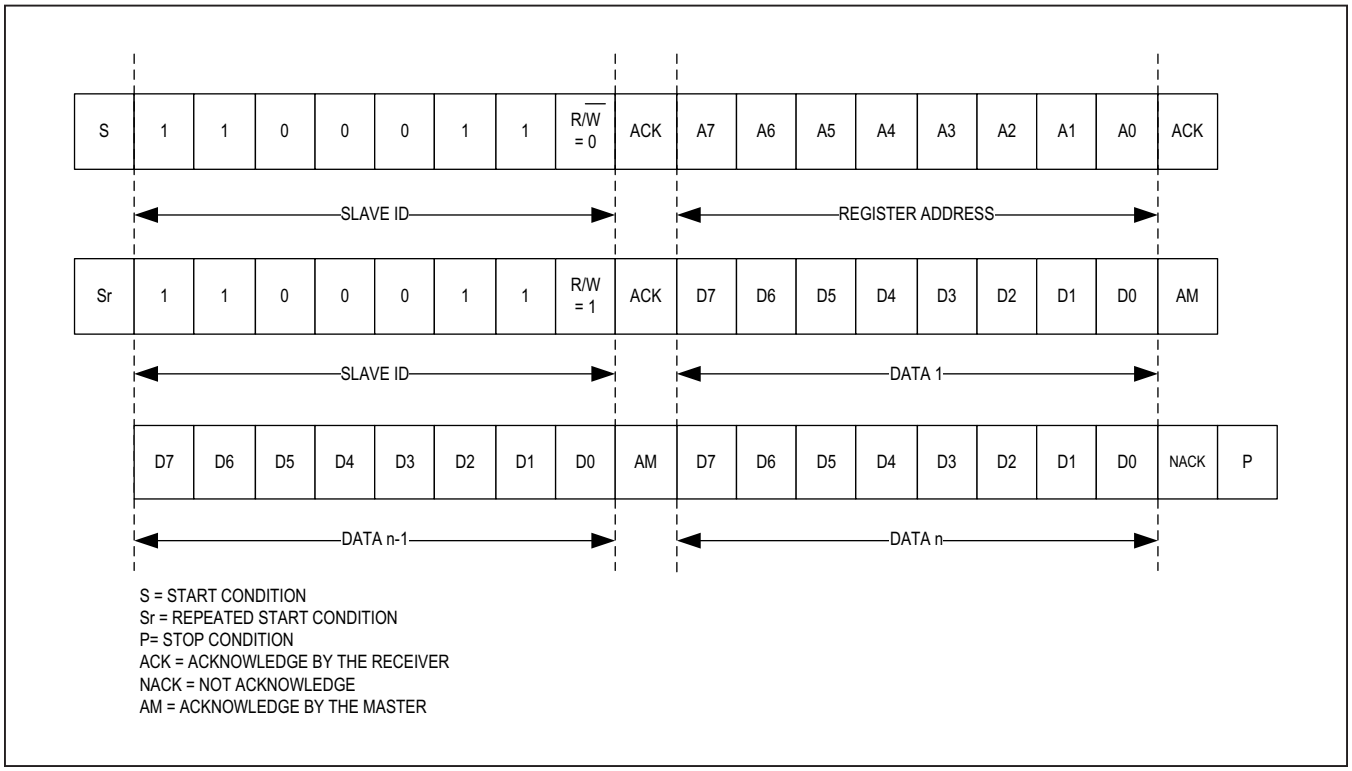


Figure 15. I^2C Multibyte Read Transaction

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	–	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_OMPB_EN	DIE_TEMP_RDY_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]	–	–	TIME_STAMP_EN	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	DAC_CODE_CHG_TAG	–	SW_FORCE_SYNC	GPIO_CTRL[3:0]			
0x11	PPG Configuration 1[7:0]	ALC_DISABLE	ADD_OFFSET	–		PPG1_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]	–	–			–	PDBIAS1[2:0]		
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]	

Register Map (continued)

ADDRESS	NAME	MSB							LSB
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]							
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]	
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]					
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]			
DAC CALIBRATION									
0x50	DAC Calibration Enable[7:0]	–	CAL_DAC_Complete	–	CAL_DAC1_OOR	–	START_CAL	–	–

Register Map (continued)

ADDRESS	NAME	MSB							LSB
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									
0xFE	Revision ID[7:0]	–	–	–	–	–	–	–	–
0xFF	Part ID[7:0]	PART_ID[7:0]							

Register Details

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	–	PWR_RDY
Reset	0x0	0x0	0x0	0x0	0x0	0x0	–	0x0
Access Type	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 by reading the Interrupt Status 1 Register. 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

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.

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.

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 it 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 its current conversion.

PWR_RDY

This is a read-only bit and it indicates that V_{DD} has gone below UVLO Threshold. This bit is not triggered by a soft reset. This bit is cleared when Interrupt Status 1 Register is read, or by setting SHDN bit to 1.

VALUE	ENUMERATION	DECODE
0	OFF	Normal Operation
1	ON	Indicates that VBATT 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 it 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	–	–
Reset	0x0	0x0	0x0	0x0	0x0	0x0	–	–
Access Type	Write, Read	Write, Read	Write, Read	Write, Read	Write, Read	Write, Read	–	–

A_FULL_EN

VALUE	ENUMERATION	DECODE
0	DISABLE	A_FULL interrupt is disabled
1	ENABLE	A_FULL interrupt in enabled

DATA_RDY_EN

VALUE	ENUMERATION	DECODE
0	DISABLE	DATA_RDY interrupt is disabled
1	ENABLE	DATA_RDY interrupt is enabled.

ALC_OVF_EN

VALUE	ENUMERATION	DECODE
0	DISABLE	ALC_OVF interrupt is disabled
1	ENABLE	ALC_OVF interrupt in enabled

PROX_INT_EN

When this 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 devices enters proximity mode or normal mode if the PROX_INT_EN is programmed to 1.

VALUE	ENUMERATION	DECODE
0	DISABLE	Proximity mode and PROX_INT interrupt are disabled
1	ENABLE	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	DISABLE	DIE_TEMP_RDY interrupt is disabled
1	ENABLE	DIE_TEMP_RDY interrupt is enabled

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	DISABLE	SHA_DONE interrupt disabled
0x1	ENABLE	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 is to be written. This pointer advances for each sample pushed on to the circular FIFO.

Refer to the *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 using the serial interface. This advances each time a sample is popped from the circular FIFO.

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

Refer to the *FIFO Configuration* for details.

OVER FLOW 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 the *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 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.

Refer to the *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 the *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 is 15 empty space left (113 entries), and so on.

Refer to the *FIFO Configuration* for details.

FIFO_A_FULL[6:0]	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_FIFO	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 the *FIFO Configuration* for details.

FIFO_STAT_CLR

This defines whether the A-FULL interrupt should get cleared by FIFO_DATA register read.

Refer to the *FIFO Configuration* for details.

VALUE	ENUMERATION	DECODE
0	RD_DATA_NOCLR	A_FULL and DATA_RDY interrupts do not get cleared by the 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 reasserts 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 reassert 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 the *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

Use one PPG Channel.

In Single PPG 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. This mode is not available for higher sample rates.

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 sample rates 256sps and below.

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	DAC_ CODE_ CHG_TAG	–	SW_ FORCE_ SYNC	GPIO_CTRL[3:0]			
Reset	0x0	0b0	–	0x0	0x0			
Access Type	Write, Read	Write, Read	–	Write, Read	Write, Read			

TIME_STAMP_EN

Enable pushing TIME_STAMP to FIFO. Refer to the *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.

DAC_CODE_CHG_TAG

Override Tag with 0x1D in the FIFO when the subranging DAC code changes for the expsoure data.

VALUE	ENUMERATION	DECODE
0x0		FIFO has the original tags for each exposure
0x1		If DAC code for any exposure has changed in the current sample, the tag in the FIFO DATA for that exposure is 0x1D (decimal 29).

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 controlled for different modes of operation.

When two devices are configured to work as master-slave device pairs, they have to be configured identically 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 the 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 slave first and then the master.

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

Refer to the GPIO Configuration for details.

GPIO_ CTRL [3:0]	GPIO FUNCTION	COMMENT
0000	Tristate or Mux Control	GPIO1 is active if any of the LEDCn[3:0] states A, B, or C are enabled in the exposure sequence. If LEDCn[3:0] state A, B, or C is not enabled in the exposure sequence, GPIO1 is in three strate unless externally pulled up.
0010	Input Sample Trigger	GPIO1 is defined as a sample trigger input (slave). This input can come from an external source or from another MAXM86161 in master sample mode. Exposure timing is controlled by an internal oscillator.
0110	Input Exposure Trigger	GPIO1 is defined as an exposure trigger input (slave). This input can come from an external source or from another MAXM86161 in master sample mode. Both sample and exposure timing is controlled by the GPIO1 input.
1001	Input HW_FORCE_ SYNC	GPIO1 is defined as a start of sample sync input. The falling edge of GPIO1 causes the present sample sequence to be terminated.
1010	Input Sample Sync ONE_SHOT	GPIO1 is defined as start of powerup sequence for one sample. The falling edge of GPIO1 starts the powerup 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 fully shuts down and waits for the next Sample Sync pulse on GPIO1.

PPG CONFIGURATION 1 (0X11)

BIT	7	6	5	4	3	2	1	0
Field	ALC_DIS-ABLE	ADD_OFF-SET	–		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. Adding offset to the PPG Data allows the 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_PPG1_ADC_RGE

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

PPG_PPG1_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

PPG1_PPG1_ADC_RGE

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

PPG_PPG1_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]	TPW, PULSE WIDTH (µs)	TTINT, INTEGRATION TIME (µs)	RESOLUTION BITS
00	21.3	14.8	19
01	35.9	29.4	19
10	65.2	58.7	19
11	123.8	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 cannot be supported by the selected pulse width and number of exposures per sample, then the highest available sample rate is 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

SAMPLING CLOCK FREQUENCY	32768Hz	32000Hz	
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 might not be possible to accommodate some of SMP_AVE values. In that case, SMP_AVE takes 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.

TLED_SETLNG, 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	ENUMERATION	DECODE
0x0		Use CDM
0x1		Use FDM

BURST_RATE

VALUE	ENUMERATION	DECODE
0x0		8 Hz
0x1		32 Hz
0x2		84 Hz
0x3		256 Hz

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 uses 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.

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 triggers 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.

Please 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 the Photo Diode Biasing for more information.

PDBIAS2[2:0]	PHOTO DIODE CAPACITANCE
0x001	0pF to 65pF
0x101	65pF to 130pF
0x110	130pF to 260pF
0x111	260pF to 520pF
All other values	Not recommended

PDBIAS1

See the Photo Diode Biasing for more information.

PDBIAS1[2:0]	PHOTO DIODE CAPACITANCE
0x001	0pF to 65pF
0x101	65pF to 130pF
0x110	130pF to 260pF
0x111	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 the *Picket Fence Detect-and-Replace Function* for details.

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

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 the *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[1:0] 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 the *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 SIGMA_MULT[1:0] 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 the *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 the *FIFO Configuration* for more information.

LEDC1

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

See the *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 the *FIFO Configuration* for more information.

LEDC3

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

See the *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 the *FIFO Configuration* for more information.

LEDC5

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

See the *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_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

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 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 of LED 2. 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[7:0] 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 LEDx is used, the respective LEDx_RGE[1:0] is used to control the range of the LED driver in conjunction with PILOT_PA[7:0]. For instance, if LED1 is used in the PILOT mode, then, LED1_RGE[1:0] together with PILOT_PA[7:0] 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. Please 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. Please refer to LED3_RGE[1:0] for more details.

LED1_RGE

Range selection of the LED current. Please 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 soft-ware.

S1_HRES_DAC1

If S1_ HI_RES_DAC1_OVR = 1, then bits S1_HRES_DAC1[5:0] set 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 bits S1_HRES_DAC1[5:0] have 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 bits S2_HRES_DAC1[5:0] set 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 bits S2_HRES_DAC1[5:0] have 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					

S2_HRES_DAC1_OVR

VALUE	ENUMERATION	DECODE
0x0	OFF	The high resolution DAC for PPG1 is controlled by the chip.
0x1	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 bits S3_HRES_DAC1[5:0] set 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 bits S3_HRES_DAC1[5:0] have 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
0x0	OFF	The high resolution DAC for PPG1 is controlled by the chip.
0x1	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 bits S4_HRES_DAC1[5:0] set 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 bits S4_HRES_DAC1[5:0] have 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
0x0	OFF	The high resolution DAC for PPG1 is controlled by the chip.
0x1	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 bits S5_HRES_DAC1[5:0] set 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 bits S5_HRES_DAC1[5:0] have 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
0x0	OFF	The high resolution DAC for PPG1 is controlled by the chip.
0x1	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 bits S6_HRES_DAC1[5:0] set 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 bits S6_HRES_DAC1[5:0] have no effect on the PPG1 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
0x0		Idle
0x1		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. For example, 0x00 = 0°C (typ), 0x7F = 127°C (typ) and 0x80 = -128°C (typ)

Note: TEMP_INT and TEMP_FRAC registers should be read using 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 using the Serial Interface in burst mode to ensure that they belong to the same sample.

DAC CALIBRATION ENABLE (0X50)

BIT	7	6	5	4	3	2	1	0
Field	–	CAL_DAC_ Complete	CAL_ DAC2_OOR	CAL_ DAC1_OOR	–	START_ CAL	–	–
Reset	–	0b0	0b0	0b0	–	00	–	–
Access Type	–	Read Only	Read Only	Read Only	–	Write, Read	–	–

CAL_DAC_Complete

High after DAC Calibration completes. It gets reset when calibration is restarted using the START_CAL bit

VALUE	ENUMERATION	DECODE
0x0		Calibration not complete or reset.
0x1		Calibration complete.

CAL_DAC2_OOR

High if any DAC2 Calibration Coefficient is out of range (OOR)

VALUE	ENUMERATION	DECODE
0x0		In Range
0x1		Any DAC2 Coefficient is out of range (OOR)

CAL_DAC1_OOR

High if any DAC1 Calibration Coefficient is out of range (OOR)

VALUE	ENUMERATION	DECODE
0x0		In Range
0x1		Any DAC1 Coefficient is out of range (OOR)

START_CAL

Start DAC1 and DAC2 calibration. This bit clears after calibration.

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	ENUMERATION	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 MAXM86161 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 MAXM86161 for comparison with the host's MAC. Note that the secret is never transmitted on the bus; thus, it 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 MAXM86161 is not affected by authentication success or failure.

Sequence of operation is as follows:

1) Enable SHA_DONE Interrupt
2) Enable SHA_EN bit
3) Write 160 bit random challenge value to RAM using registers MEM_IDX and MEM_DATA.
4) Write command, with ROM ID (0x35) or without ROM ID (0x36) to SHA_CMD register.
5) Write 1 to SHA_START and 1 to SHA_EN bit.
6) Wait for SHA_DONE interrupt.
7) Read 256 MAC value from RAM using registers MEM_IDX and MEM_DATA.
8) Compare MAC from MAXM86161 wth Host's precalculated MAC.
9) Check PASS or FAIL
10) Disable SHA_EN bit (Write 0 to SHA_EN bit).

VALUE	ENUMERATION	DECODE
0x0		Authentication is disabled
0x1		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 via.

VALUE	ENUMERATION	DECODE
0x0		Writing to Memory via is disabled.
0x1		Writing to Memory via 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	ENUMERATION	DECODE
0x0		Select Bank 0, address 0x00 to 0xFF
0x1		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 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 reads 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	0x36							
Access Type	Read Only							

PART_ID

This register stores the Part identifier for the chip.

0x36

Ordering Information

PART NUMBER	TEMP RANGE	PIN-PACKAGE
MAXM86161EFD+	-40°C to +85°C	4.3mm x 2.9mm x 1.4mm 14-Pin OLGA
MAXM86161EFD+T	-40°C to +85°C	4.3mm x 2.9mm x 1.4mm 14-Pin OLGA

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

T = Tape and reel.

Revision History

REVISION NUMBER	REVISION DATE	DESCRIPTION	PAGES CHANGED
0	3/19	Initial release	—