AS7024

Biosensors

General Description

The AS7024 device provides a flexible analog front end for light sensing applications. The photodiode input circuit can be configured in different ways to guarantee best tradeoff between speed and sensitivity for a large number of different sensing applications.

AS7024 is targeted for wearables (fitness band, smart watch).

Ordering Information and Content Guide appear at end of datasheet.

Key Benefits and Features

The benefits and features of AS7024, Biosensors are listed below:

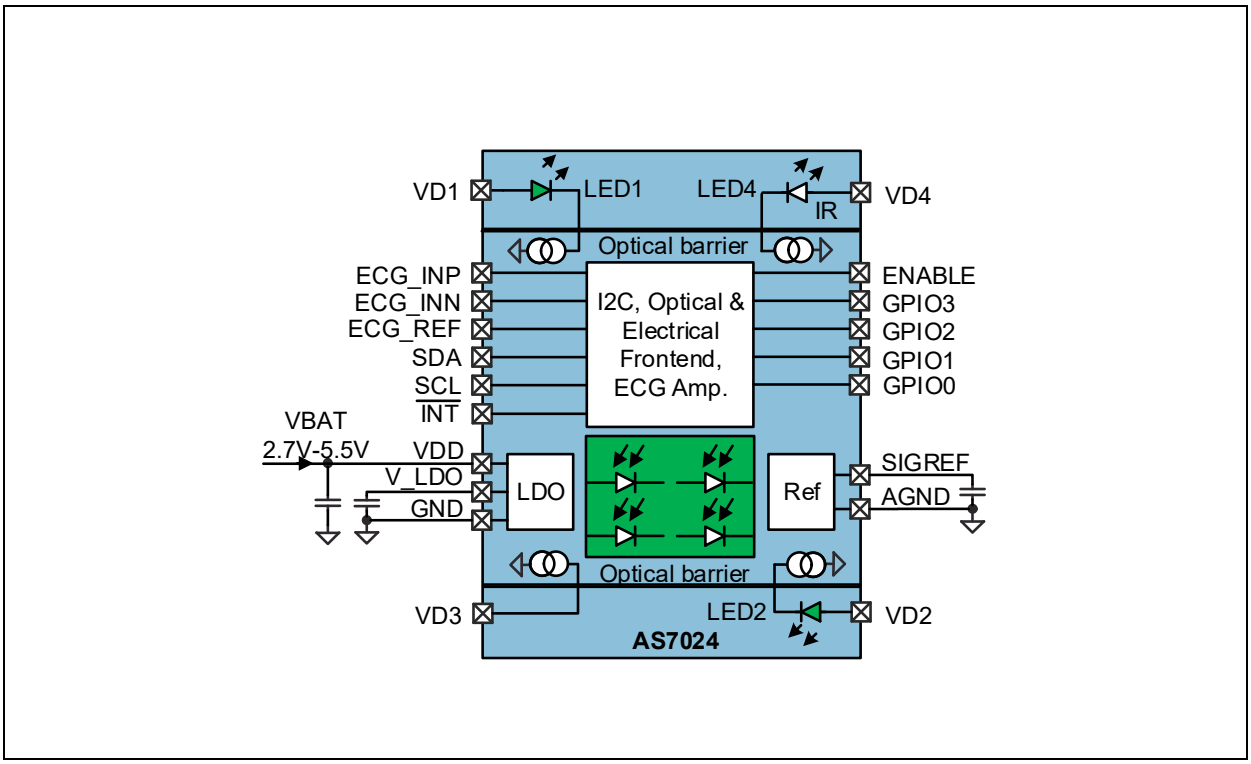
Figure 1:
Added Value of Using AS7024

Benefits	Features
• Allows smallest application size e.g. narrow HRM measurement band	• Single device integrated optical solution
• Good HRM measurement quality	• Low noise analog optical front end
• Additional information for end user	• Analog electrical front end (e.g. for temperature sensing using a NTC or galvanic skin resistivity (GSR))
• Long operating time	• Hardware sequencer to offload processor • Adjustable LED driver with current control
• Works reliably with ambient light	• Synchronous demodulator

Applications

The device is suitable for optical sensor platform.

Figure 2:
Application Schematic AS7024



Pin Assignments

Figure 3:
AS7024 Optical Module Pinout (Top View)

Optical Module Pinout:
This drawing is not to scale

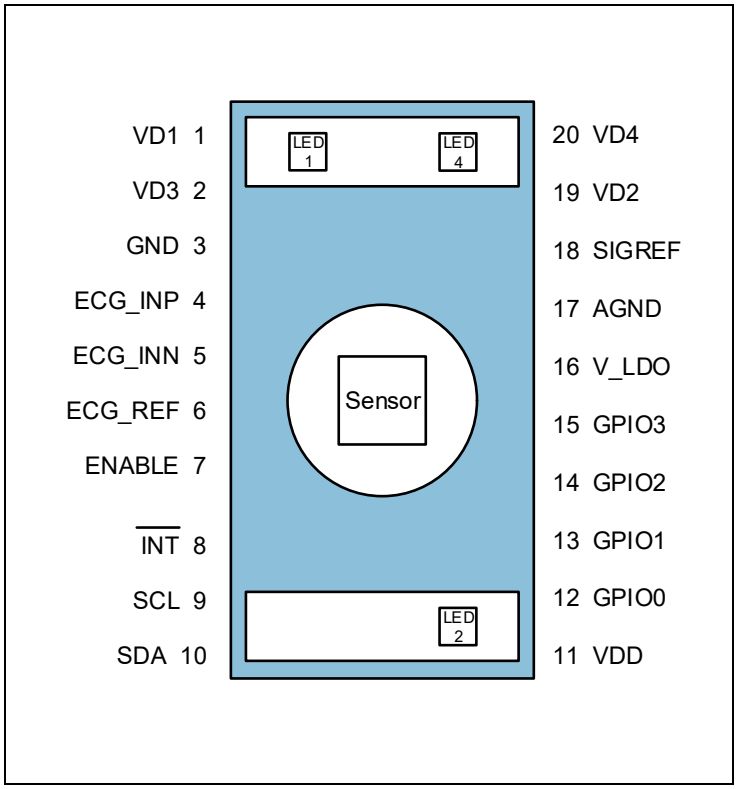


Figure 4:
Pin Description

Pin No.	Pin Name	Description
1	VD1	Supply voltage for LED D1
2	VD3	Connection to current sink 3
3	GND	Power supply ground. All voltages are referenced to GND.
4	ECG_INP	ECG amplifier positive input
5	ECG_INN	ECG amplifier negative input
6	ECG_REF	ECG amplifier reference output
7	ENABLE	Enable input for AS7024. Active high. Setting this input to low resets all internal registers and the AS7024 enters power down mode. Setting it high allows operation of the AS7024. If ENABLE is not used (AS7024 always enabled), connect to VDD.
8	$\overline{INT}$	Open drain interrupt output pin. Active low.
9	SCL	I^2C serial clock input terminal – the device does not use clock stretching therefore SCL is only an input terminal.

Pin No.	Pin Name	Description
10	SDA	I^2C serial data I/O terminal – open drain.
11	VDD	Supply voltage. Connect a 2.2 F capacitor to GND.
12	GPIO0	General purpose input/output
13	GPIO1	General purpose input/output
14	GPIO2	General purpose input/output
15	GPIO3	General purpose input/output
16	V_LDO	1.9V output voltage. Connect 2.2 F capacitor to GND (e.g. 0402 sized capacitor GRM153R60J225ME95 or 0201 sized GRM033R60J225ME47 from Murata – needs to have >1 F with 1.0V voltage bias); do not load externally
17	AGND	Analog ground. Connect to low noise GND
18	SIGREF	Analog reference output. Connect 2.2 F capacitor to GND (e.g. 0402 sized capacitor GRM153R60J225ME95 or 0201 sized GRM033R60J225ME47 from Murata – needs to have >1 F specified for 1.0V voltage bias); do not load externally The typical operating voltage on this pin is 0.6V (sigref_en=1)
19	VD2	Supply voltage for LED D2
20	VD4	Supply voltage for LED D4

Absolute Maximum Ratings

Stresses beyond those listed 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 Electrical Characteristics is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

Figure 5:
Absolute Maximum Ratings (1)

Symbol	Parameter	Min	Max	Units	Comments
Electrical Parameters					
VDD	Supply Voltage to Ground		6V	V	
V_{IN}	Input Pin Voltage to Ground pins GPIO0/1/2/3	-0.3	VDD+0.3V max. 6V	V	Diode to VDD
$V_{IN-OTHER}$	Input Pin Voltage to Ground pins SCL/SDA/INT/ENABLE and VD1/VD2/VD3/VD4	-0.3	5.5	V	No internal diode to VDD or V_LDO
$V_{VD1/2/3/4\_INTERNAL}$	Voltage between internal pin of VD1-VD4 to VDD		VDD+0.3V	V	Internal diode between current source (internal node at anode of the LED if the pin has a LED otherwise VD1/2/3/4 pin) and VDD
V_{IN-LDO}	Input Pin Voltage to Ground for pin V_LDO	-0.3	VDD+0.3V max. 6V	V	Diode to VDD
V_{IN-LDO_DIODE}	Input Pin Voltage to Ground pins for ECG_INP/ECG_INN/ECG_REF/SIGREF	-0.3	V_LDO+0.3V max. 6V	V	Diode to V_LDO
$V_{GND-AGND}$	Analog to power ground voltage difference	-0.3	+0.3	V	
I_{SCR}	Input Current (latch-up immunity)	-100	100	mA	JEDEC JESD78 Connect specified capacitor on SIGREF and V_LDO during latchup test
Electrostatic Discharge					
ESD_{HBM}	Electrostatic Discharge HBM	±2.0		kV	JS-001-2014

Symbol	Parameter	Min	Max	Units	Comments
Temperature Ranges and Storage Conditions					
T_{STRG}	Storage Temperature Range	-40	85	°C	
T_{BODY}	Package Body Temperature		260	°C	IPC/JEDEC J-STD-020 The reflow peak soldering temperature (body temperature) is specified according to IPC/JEDEC J-STD-020 "Moisture/Reflow Sensitivity Classification for Non-hermetic Solid State Surface Mount Devices."
RH_{NC}	Relative Humidity (non-condensing)	5	85	%	
MSL	Moisture Sensitivity Level	3			Maximum floor life time of 168h

Note(s):

1. All optical customer designs shall be reviewed by **ams** before production.

Electrical Characteristics

All limits are guaranteed. The parameters with min and max values are guaranteed with production tests or SQC (Statistical Quality Control) methods.

VDD=2.7 to 5.5V, typ. values are at T_{AMB}=25°C (unless otherwise specified).

Figure 6:
Operating Conditions

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
VDD	Supply voltage		2.7	3.8	5.5	V
T_{AMB}	Operating free-air temperature		–30		70	°C

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
IDD	Supply current	ENABLE=VDD, ldo_en=0; osc_en=0; internal LDO operating in low power mode – only I²C communication possible, no blocks shall be enabled (1)		22		µA
		ENABLE=VDD, ldo_en=1; osc_en=0; internal LDO operating and bandgap running – I²C communication possible, analog blocks can be enabled (1)		32		µA
		ENABLE=VDD, ldo_en=1, osc_en=1; internal LDO operating and bandgap and oscillator running – I²C communication possible, analog blocks can be enabled(1)		86		µA
		SIGREF buffer (sigref_en=1)		52		µA
		transimpedance amplifier (pd_amp_en=1)		110		µA
		Optical front end operating (one channel)		200		µA
		Gain stage (ofe1_gain_en=1 or ofe2_gain_en=1)		75		µA
		ADC sampling at 20Hz with 64µs settling time		4.5		µA
		ECG amplifier and frontend (need SIGREF enabled)		190		µA
		ECG leakage compensation (ecg_ low_leakage_en=1), low pass filter, high pass filter and gain stage		151		µA
		Power down, no I²C communication possible (2) ENABLE=GND		0.5		µA
VOL	GPIO0-3, INT, SDA output low voltage	With 3 mA load With 6 mA load	0 0		0.4 0.8	V
VOH	GPIO0-3 output high voltage	With 3 mA load	2.3		VDD	V
VIH	GPIO0-3, SCL, SDA, ENABLE input high voltage		1.25			V

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
VIL	GPIO0-3, SCL, SDA, ENABLE input low voltage				0.54	V
ILEAK1	GPIO0-3, SCL, SDA, ENABLE, INT		–1		1	µA
ILEAK2	VD1, VD23 pins		–1		1	µA
E_f2M	Tolerance of internal 2MHz oscillator	0ºC to 70ºC, VDD<5.0V	-2		+2	%
		-30ºC to 70ºC	-4		+2	%
ECG Amplifier and Filter						
ILEAK_ ECG	ECG pins leakage current	Lab evaluation shows <±20nA maximum leakage current. Not production tested.		±1		nA
V_{NOISE_ECG}	Integrated noise	ADC sampling at 400Hz; low pass filter set to 40Hz, PPG channel operating in parallel		20		µV
$CMRR_{ECG}$	Common mode rejection ratio	Measured at 50Hz and 100Hz		73		dB

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
LED						
I_{LED}	Allowed operating LED current range		0		50	mA
		1/10 duty cycle @ 1 kHz			100	
VF_{LED}	Forward voltage	Green LED, add compliance voltage of LED driver, ILED=10mA, add compliance voltage of LED driver (V_Dmin) to obtain minimum voltage on the pin to drive the current at T_{AMB}=25°C		3.1	3.3	V
		IR LED, ILED=20mA		1.4		
λp	Peak wavelength at I_{LED}=20mA	Green LED		527		nm
		IR LED		940		
LED Driver						
$I_{LED1/2/3/4}$	LED output current range	LED current is adjustable with 10 bits – registers curr1/2/3/4	0		100	mA
	Tolerance	At 35mA output current (currX[9:0]=166h, X=1…4), VDD<5.0V	-7		7	%
V_Dmin	Output voltage compliance			0.3		V
V_Dmax	Output voltage maximum				5.5	V

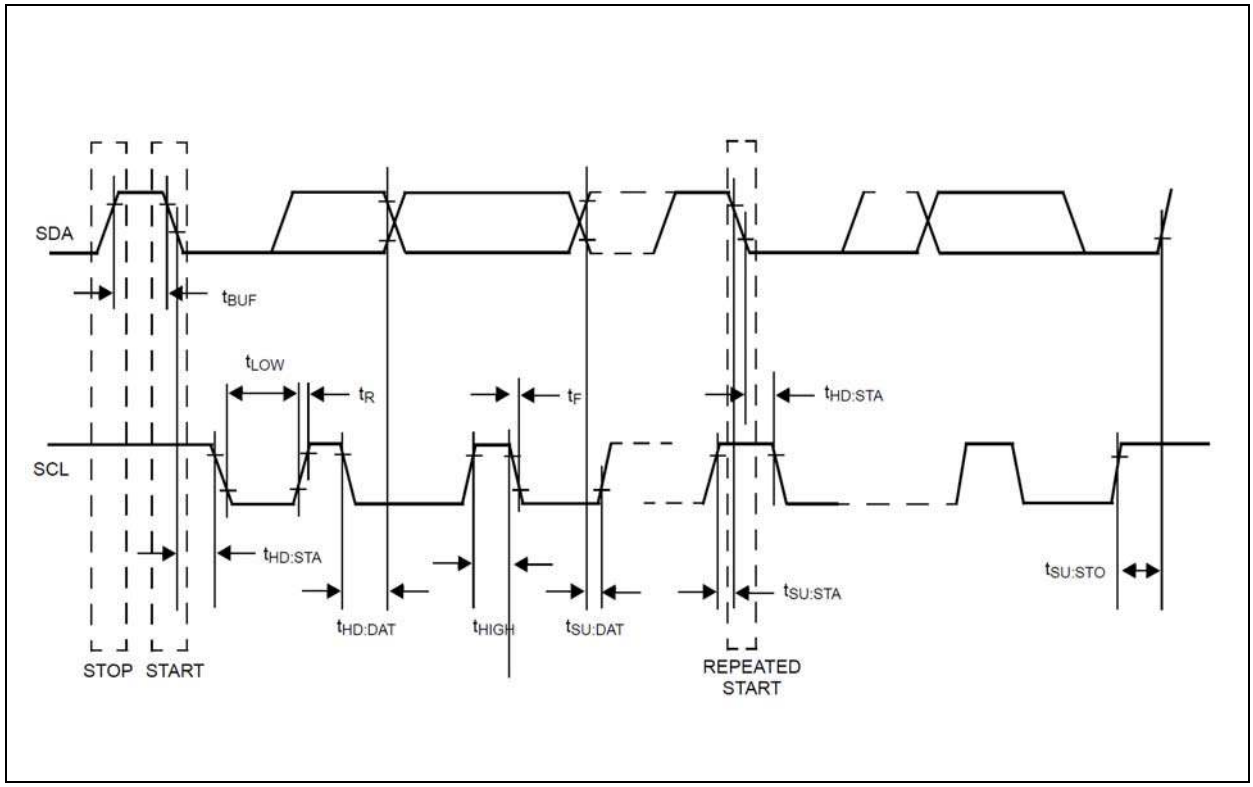
Symbol	Parameter	Conditions	Min	Typ	Max	Unit
Photodiode						
Re_{PD1-4}	Irradiance responsivity photodiode PD1…PD4	λ_P=525nm, 4 photodiodes used pd1/2/3/4=1, gain_g=4x, gain_en=1, pd_ampres = 7MΩ		45.9		mV/(µW/cm^2)
	Irradiance responsivity photodiode B	λ_P=940nm, gain_g=4x, gain_en=1, pd_ampres=7MΩ		0.3		
Id	Dark current	E_e=0, T_{AMB} =25°C	0		1	nA
Ios	Extrapolated offset current	T_{AMB} =25°C	–1		1	nA
ADC						
Vref	ADC reference voltage			1.6		V
	Resolution		14			Bit
INL	Relative accuracy	T_{AMB} =25°C	-8		8	LSB
DNL	Differential nonlinearity(3)			1.5		LSB
	Offset error		-8		8	LSB
	Gain error		-8		8	LSB
SNR	Signal-to-noise ratio	Fsample = 1kHz, Fsignal=100Hz		80		dB
THD	Total harmonic distortion	Fsample = 1kHz, Fsignal=100Hz		-70		dB
Tconv	Conversion rate	14 bit resolution			50	ksps
Vin	Input voltage range		0		Vref	V

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
I²C Mode Timings						
f_{SCLK}	SCL Clock Frequency		0		400	kHz
t_{BUF}	Bus Free Time Between a STOP and START Condition		1.3			µs
$t_{HD:STA}$	Hold Time (Repeated) START Condition (4)		0.6			µs
t_{LOW}	LOW Period of SCL Clock		1.3			µs
t_{HIGH}	HIGH Period of SCL Clock		0.6			µs
$t_{SU:STA}$	Setup Time for a Repeated START Condition		0.6			µs
$t_{HD:DAT}$	Data Hold Time (5)		0		0.9	µs
$t_{SU:DAT}$	Data Setup Time (6)		100			ns
t_R	Rise Time of Both SDA and SCL Signals		20		300	ns
t_F	Fall Time of Both SDA and SCL Signals		20		300	ns
$t_{SU:STO}$	Setup Time for STOP Condition		0.6			µs
C_B	Capacitive Load for Each Bus Line	CB — total capacitance of one bus line in pF			400	pF
$C_{I/O}$	I/O pin Capacitance (SDA, SCL)				10	pF

Note(s):

1. GPIO0-3 configured to draw minimum current (software dependent).
2. The design of AS7024 is done in a way that it does not affect other I²C communication on SCL/SDA even if AS7024 is in power down.
3. Specified only typical value for DNL to reduce production test time.
4. After this period, the first clock pulse is generated.
5. A device must internally provide a hold time of at least 300ns for the SDA signal (referred to the V_{IHMIN} of the SCL signal) to bridge the undefined region of the falling edge of SCL.
6. A fast-mode device can be used in a standard-mode system, but the requirement $t_{SU:DAT}$ = to 250ns must then be met. This is automatically the case if the device does not stretch the LOW period of the SCL signal. If such a device does stretch the LOW period of the SCL signal, it must output the next data bit to the SDA line tR max + $t_{SU:DAT}$ = 1000 + 250 = 1250ns before the SCL line is released.

Figure 7:
I²C Mode Timing Diagram

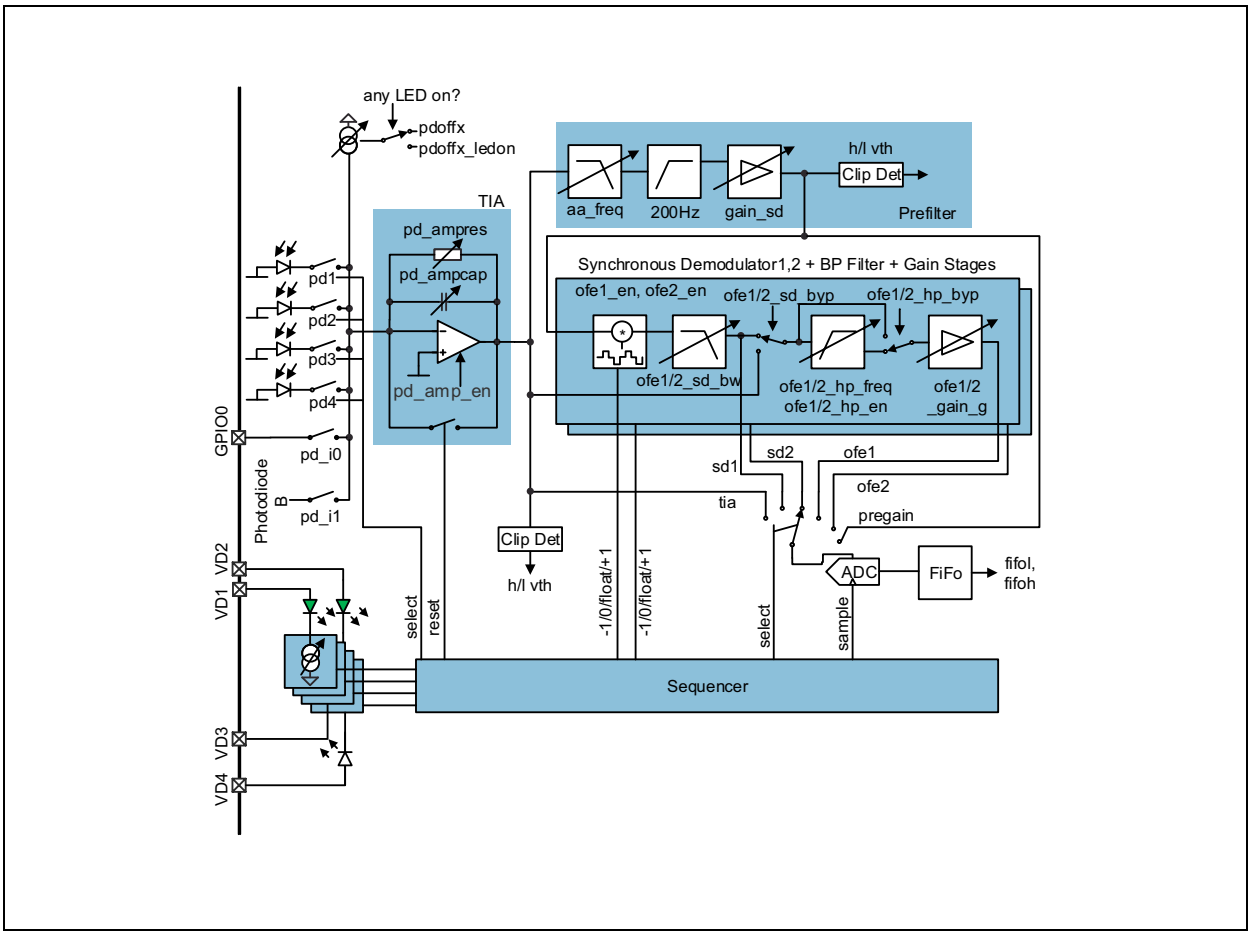


I²C Mode Timing Diagram: This figure shows the different timings required for I²C communication.

Detailed Description

Optical Analog Front End

Figure 8:
Optical Analog Front End



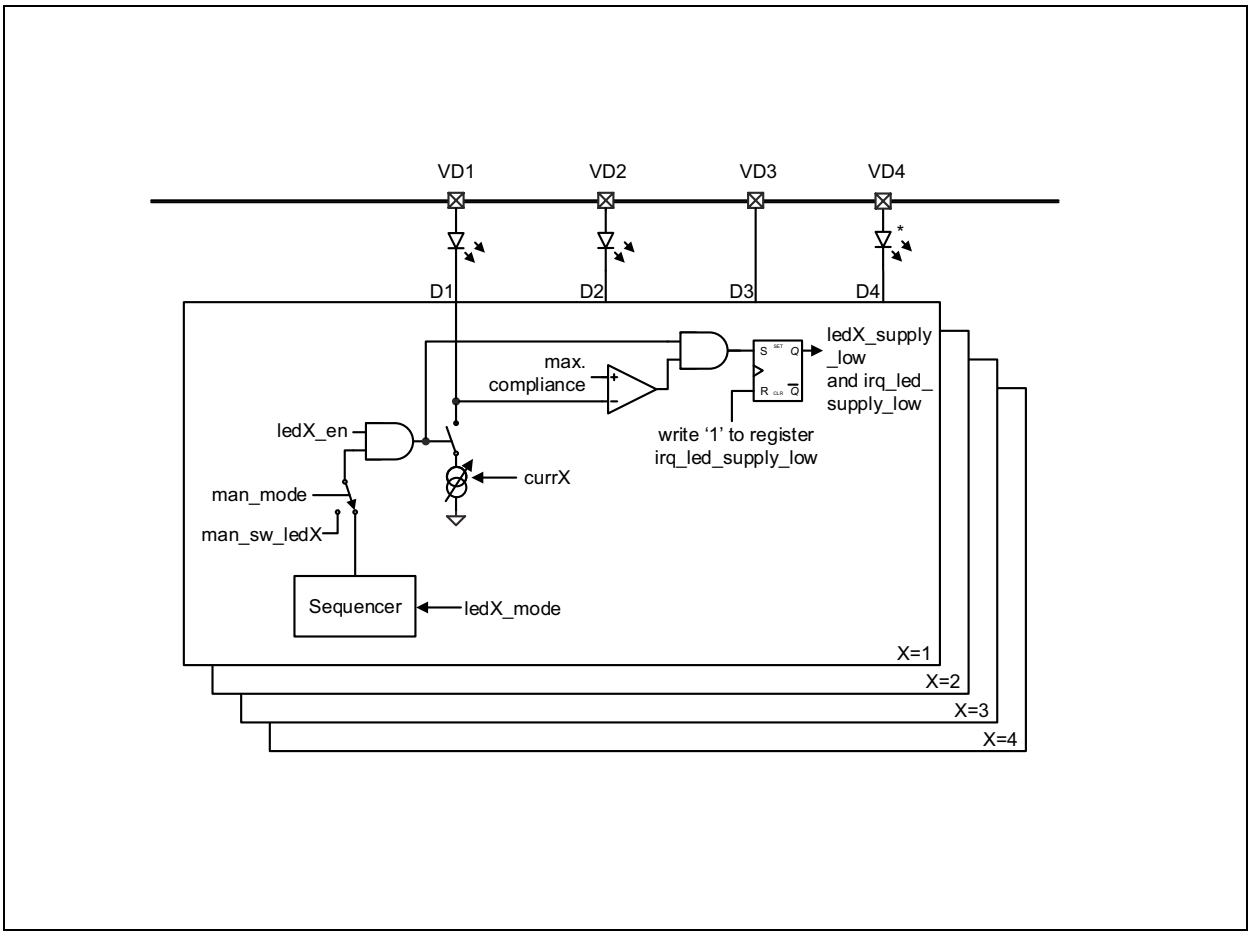
LEDs

Two green LEDs are used with anode on pin VD1 and VD2. A IR LED is connected with anode to pin VD4. VD3 allows direct access to the current sink 3.

LED-Driver

The LED-driver outputs can be controlled manually or by the built in sequencer. See Optical Front End Operating Modes

Figure 9:
LED Drivers



LED Configuration Registers

Figure 10:
LED_CFG Register

0x10: LED_CFG					
Field	**Name**	**Rst**	**Type**	**Description**	
6	sigref_en	0	RW	Signal reference: Is required for all analog blocks (except PD_Amp or light-to-frequency operation) 0 … Disable signal reference 1 … Enable signal reference	
5:4	sigref_voltage	0	RW	Voltage setting of SIGREF – datasheet parameters are guaranteed only for default value of 0.6V.	
				Setting	**IMAX**
				0	0.6V (default)
				1	0.7V
				2	0.8V
				3	0.9V
3	led4_en	0	RW	0 … Disables LED4 output source. 1 … Enables LED4 output source.	
2	led3_en	0	RW	0 … Disables LED3 output source. 1 … Enables LED3 output source.	
1	led2_en	0	RW	0 … Disables LED2 output source. 1 … Enables LED2 output source.	
0	led1_en	0	RW	0 … Disables LED1 output source. 1 … Enables LED1 output source.	

The LED_CURR defines the LED output current. Warning: it is recommended to configure the current only when the output is not active, as there is no latch implemented to keep the 10 bits consistent. New values are applied directly and immediately.

Figure 11:
LED1_CURRL Register

0x12: LED1_CURRL				
Field	**Name**	**Rst**	**Type**	**Description**
7:6	curr1[1:0]	0	RW	LED1 output current lower 2 bits

Figure 12:
LED1_CURRH Register

0x13: LED1_CURRH				
Field	**Name**	**Rst**	**Type**	**Description**
7:0	curr1[9:2]	0	RW	LED1 output current upper 8 bits Coding for curr1[9:0]: 000h … 786µA 001h … 883µA (1 LSB=97µA) 002h … 980µA 166h … 35mA 3FFh … 100mA

Figure 13:
LED2_CURRL Register

0x14: LED2_CURRL				
Field	**Name**	**Rst**	**Type**	**Description**
7:6	curr2[1:0]	0	RW	LED2 output current lower 2 bits

Figure 14:
LED2_CURRH Register

0x15: LED2_CURRH				
Field	**Name**	**Rst**	**Type**	**Description**
7:0	curr2[9:2]	0	RW	LED2 output current upper 8 bits Coding for curr2[9:0]: 000h … 786µA 001h … 883µA (1 LSB=97µA) 002h … 980µA 166h … 35mA 3FFh … 100mA

Figure 15:
LED3_CURRL Register

0x16: LED3_CURRL				
Field	**Name**	**Rst**	**Type**	**Description**
7:6	curr3[1:0]	0	RW	LED3 output current lower 2 bits

Figure 16:
LED3_CURRH Register

0x17: LED3_CURRH				
Field	**Name**	**Rst**	**Type**	**Description**
7:0	curr3[9:2]	0	RW	LED3 output current upper 8 bits Coding for curr3[9:0]: 000h … 786µA 001h … 883µA (1 LSB=97µA) 002h … 980µA 166h … 35mA 3FFh … 100mA

Figure 17:
LED4_CURRL Register

0x18: LED4_CURRL				
Field	**Name**	**Rst**	**Type**	**Description**
7:6	curr4[1:0]	0	RW	LED4 output current lower 2 bits

Figure 18:
LED4_CURRH Register

0x19: LED4_CURRH				
Field	**Name**	**Rst**	**Type**	**Description**
7:0	curr4[9:2]	0	RW	LED4 output current upper 8 bits Coding for curr4[9:0]: 000h … 786µA 001h … 883µA (1 LSB=97µA) 002h … 980µA 166h … 35mA 3FFh … 100mA

Figure 19:
LED12_MODE Register

<table>
<tr><th colspan="5">0x2c: LED12_MODE</th></tr>
<tr><th>Field</th><th>Name</th><th>Rst</th><th>Type</th><th>Description</th></tr>
<tr><td>7</td><td>man_sw_led2(1)</td><td>0</td><td>RW</td><td colspan="2">0 … LED output D2 disabled. (High impedance)
1 … LED output D2 enabled</td></tr>
<tr><td rowspan="9">6:4</td><td rowspan="9">led2_mode</td><td rowspan="9">0</td><td rowspan="9">RW</td><td colspan="2">LED2 mode</td></tr>
<tr><td>Setting</td><td>Behavior</td></tr>
<tr><td>0</td><td>Always OFF</td></tr>
<tr><td>1</td><td>Always ON when sequencer is active</td></tr>
<tr><td>2</td><td>Controlled by sequencer</td></tr>
<tr><td>3</td><td>Controlled by sequencer, only ON in even iterations: 0, 2, 4 etc.</td></tr>
<tr><td>4</td><td>Controlled by sequencer, only ON in odd iterations: 1, 3, 5 etc.</td></tr>
<tr><td>5</td><td>Controlled by sequencer, only ON in every fourth iteration, starting at 1: 1, 5, 9 etc.</td></tr>
<tr><td>6</td><td>Controlled by sequencer: secondary LED timing</td></tr>
<tr><td>7</td><td>Do not use</td></tr>
<tr><td>3</td><td>man_sw_led1(1)</td><td>0</td><td>RW</td><td colspan="2">0 … LED output D1 disabled. (High impedance)
1 … LED output D1 enable</td></tr>
</table>

0x2c: LED12_MODE					
Field	**Name**	**Rst**	**Type**	**Description**	
2:0	led1_mode	0	RW	LED1 mode	
				Setting	**Behavior**
				0	Always OFF
				1	Always ON when sequencer is active
				2	Controlled by sequencer
				3	Controlled by sequencer, only ON in even iterations: 0, 2, 4 etc.
				4	Controlled by sequencer, only ON in odd iterations: 1, 3, 5 etc.
				5	Controlled by sequencer, only ON in every fourth iteration, starting at 0: 0, 4, 8 etc.
				6	Controlled by sequencer: secondary LED timing
				7	Do not use

Note(s):

1. Function enabled only in manual mode

Figure 20:
LED34_MODE Register

0x2d: LED34_MODE					
Field	**Name**	**Rst**	**Type**	**Description**	
7	man_sw_ led4 (1)	0	RW	0 … LED output D4 disabled. (High impedance) 1 … LED output D4 enabled	
6:4	led4_mode	0	RW	LED4 mode	
				Setting	**Behavior**
				0	Always OFF
				1	Always ON when sequencer is active
				2	Controlled by sequencer
				3	Controlled by sequencer, only ON in even iterations: 0, 2, 4 etc.
				4	Controlled by sequencer, only ON in odd iterations: 1, 3, 5 etc.
				5	Controlled by sequencer, only ON in every fourth iteration, starting at 3: 3, 7, 11 etc.
				6	Controlled by sequencer: secondary LED timing
				7	Do not use
3	man_sw_ led3 (1)	0	RW	0 … LED output D3 disabled. (High impedance) 1 … LED output D3 enabled	

0x2d: LED34_MODE					
Field	Name	Rst	Type	Description	
2:0	led3_mode	0	RW	LED3 mode	
				Setting	Behavior
				0	Always OFF
				1	Always ON when sequencer is active
				2	Controlled by sequencer
				3	Controlled by sequencer, only ON in even iterations: 0, 2, 4 etc.
				4	Controlled by sequencer, only ON in odd iterations: 1, 3, 5 etc.
				5	Controlled by sequencer, only ON in every fourth iteration, starting at 2: 2, 6, 10 etc.
				6	Controlled by sequencer: secondary LED timing
				7	Do not use

Note(s):

1. Function enabled only in manual mode

The MAN_SEQ_CFG register is used to configure the operation of the optical front end.

Figure 21:
MAN_SEQ_CFG Register

0x2e: MAN_SEQ_CFG				
Field	Name	Rst	Type	Description
7	man_mode	0	RW	0 … Enables Sequencer 1 … Enables Manual control of optical front end
6	man_sw_ sdmult	0	RW	If man_mode=1 0 … Disables synchronous demodulator multiplication 1 … Enables synchronous demodulator multiplication
5	man_sw_ sdpol	0	RW	If man_mode=1 0 … Negative polarity in synchronous demodulator multiplication 1 … Positive polarity in synchronous demodulator multiplication
4	man_sw_itg	0	RW	If man_mode=1 0 … All integrator capacitors are shorted. Integrator is reset 1 … Integrator capacitors are charging up. Integrator is running

<table>
<tr><th colspan="8">0x2e: MAN_SEQ_CFG</th></tr>
<tr><th>Field</th><th>Name</th><th>Rst</th><th>Type</th><th colspan="4">Description</th></tr>
<tr><td rowspan="10">3:1</td><td rowspan="10">diode_ctrl</td><td rowspan="10">0</td><td rowspan="10">RW</td><td colspan="4">Connection of Photodiodes PD1, PD2, PD3, PD4 to the photodiode amplifier.
0 … PD1-PD4 are connected
1 … PD1 synchronous to LED1, PD2 sync/to LED2, PD3 sync/to LED3, PD4 sync/to LED4
2 … PD1 synchronous to LED1, PD2 sync/to LED1, PD3 sync/to LED2, PD4 sync/to LED2
3 … PD1 synchronous to LED1, PD2 sync/to LED1, PD3 sync/to LED4, PD4 sync/to LED4
4 … SPO2 mode *(obsolete): (negedge(sdm1) or negedge(sdp1)) - PD1=0 PD2=0 PD3=1 PD4=1; (negedge(sdm2) or negedge(sdp2)) - PD1=1 PD2=1 PD3=0 PD4=0
Note that PD_CFG.pdX takes precedence - to turn OFF one photo diode, the respective bit has to be de-asserted in the PD_CFG register.</td></tr>
<tr><td>PD_ CFG.pdX</td><td>diode_ ctrl</td><td>Photo Diode1</td><td>Photo Diode2</td><td>Photo Diode3</td><td>Photo Diode4</td></tr>
<tr><td>0</td><td>xx</td><td>OFF</td><td>OFF</td><td>OFF</td><td>OFF</td></tr>
<tr><td>1</td><td>0</td><td>ON</td><td>ON</td><td>ON</td><td>ON</td></tr>
<tr><td>1</td><td>1</td><td>LED1</td><td>LED2</td><td>LED3</td><td>LED4</td></tr>
<tr><td>1</td><td>2</td><td>LED1</td><td>LED1</td><td>LED2</td><td>LED2</td></tr>
<tr><td>1</td><td>3</td><td>LED1</td><td>LED1</td><td>LED4</td><td>LED4</td></tr>
<tr><td>1</td><td>4</td><td colspan="4">SPO2 mode (obsolete)</td></tr>
<tr><td>1</td><td>5..7</td><td colspan="4">Do not use</td></tr>
<tr><td>0</td><td>seq_en</td><td>0</td><td>RW</td><td colspan="4">0 … Disables sequencer
1 … Enables sequencer</td></tr>
</table>

Figure 22:
LEDSTATUS Register

0xa2: LEDSTATUS				
Field	**Name**	**Rst**	**Type**	**Description**
3	led4_supply_low	0	RO	If this bit is asserted, LED4 voltage has been too low.
2	led3_supply_low	0	RO	If this bit is asserted, LED3 voltage has been too low.
1	led2_supply_low	0	RO	If this bit is asserted, LED2 voltage has been too low.
0	led1_supply_low	0	RO	If this bit is asserted, LED1 voltage has been too low.

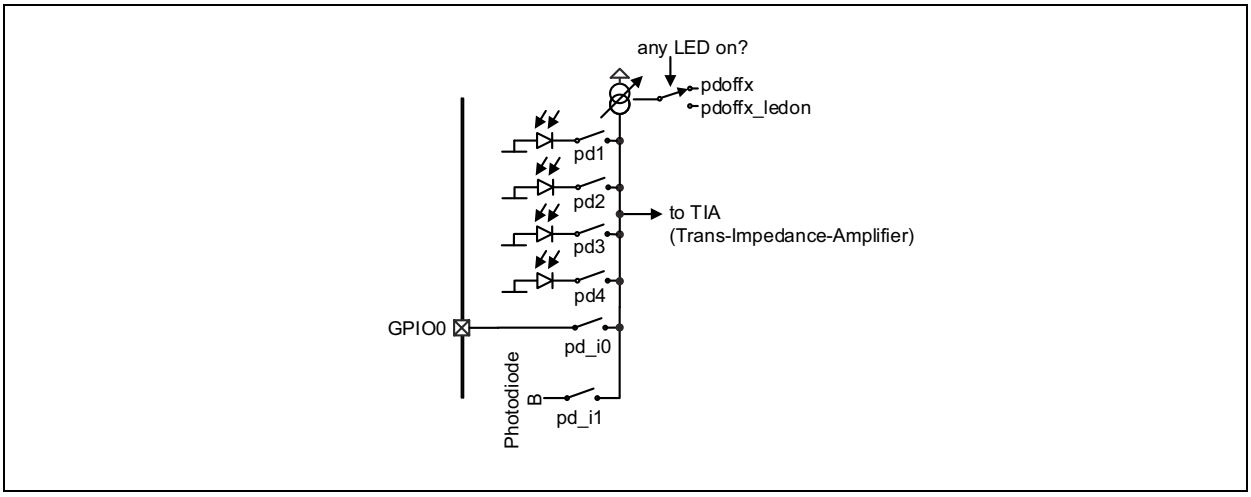
An asserted bit can be cleared by writing a '1' to the irq_led_ supply_low bit.

Photodiode Selection

In order to have flexible arrangement of the use photodiodes, PD1-PD4 can be individually connected to the photodiode amplifier input. The optional offset current allows cancellation of constant light sources like sunlight. In case of an external photodiode or any other sensor with (low) current output, the pins GPIO0 and GPIO1 can be used as input.

Additionally the sequencer can control the diodes – see diode_ctrl described in register MAN_SEQ_CFG.

Figure 23:
Photodiode Selection



Photodiode Registers

The PD_CFG register is used to configure the input to the photo amplifier.

Figure 24:
PD_CFG Register

0x1a: PD_CFG				
Field	**Name**	**Rst**	**Type**	**Description**
5	pd4	0	RW	0 … Photodiode PD4 is disconnected from photo amplifier 1 … Photodiode PD4 is connected to photo amplifier (as defined in diode_ctrl)
4	pd3	0	RW	0 … Photodiode PD3 is disconnected from photo amplifier 1 … Photodiode PD3 is connected to photo amplifier (as defined in diode_ctrl)
3	pd2	0	RW	0 … Photodiode PD2 is disconnected from photo amplifier 1 … Photodiode PD2 is connected to photo amplifier (as defined in diode_ctrl)
2	pd1	0	RW	0 … Photodiode PD1 is disconnected from photo amplifier 1 … Photodiode PD1 is connected to photo amplifier (as defined in diode_ctrl)
1	pd_i1	0	RW	0… Photodiode B (see Photodiode Characteristics) disconnected from TIA input 1… Photodiode B (see Photodiode Characteristics) connected to TIA input; set ltf1_sel=0 and ltf2_sel=0.
0	pd_i0	0	RW	0 … GPIO0-input is disconnected from photo amplifier 1 … GPIO0-input is connected to photo amplifier; set gpio_a[0]=1.

Figure 25:
PDOFFX_LEDOFF Register

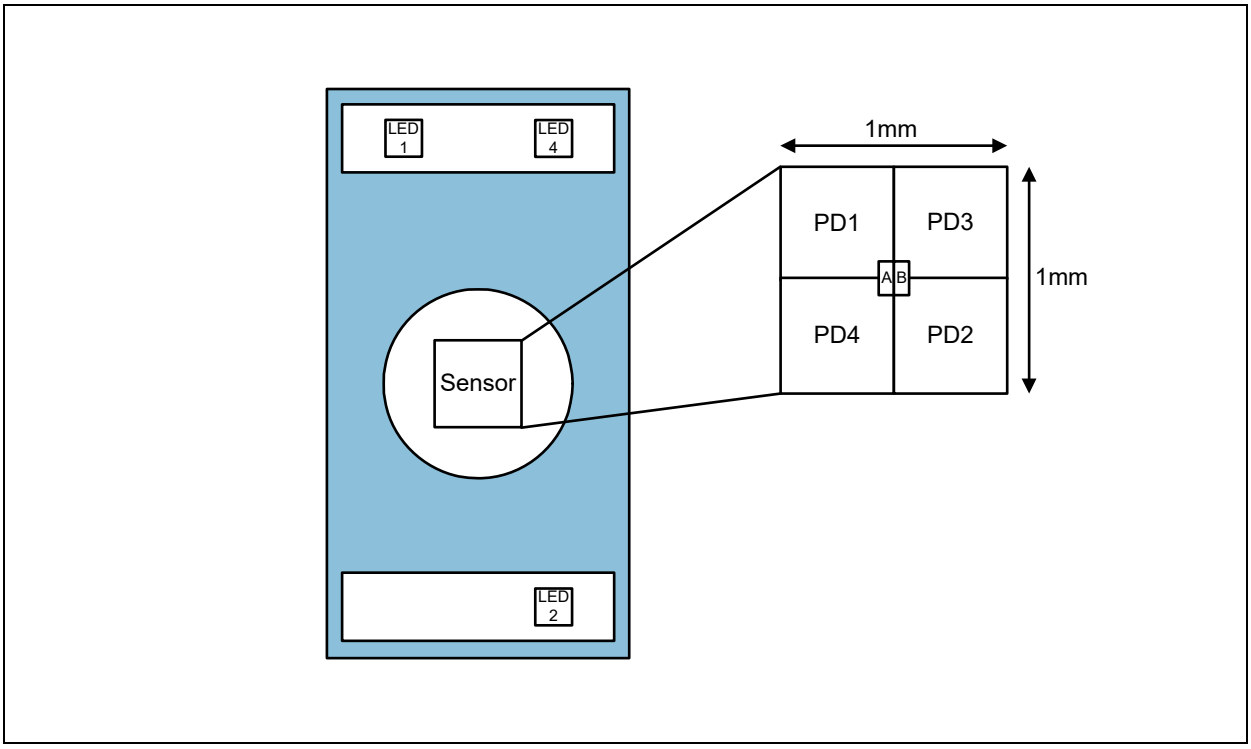
0x1b: PDOFFX_LEDOFF				
Field	**Name**	**Rst**	**Type**	**Description**
7:0	pdoffx_ ledoff	0	RW	Input offset current if all LEDs are OFF (all sw_led* sequencer outputs are zero) Ioffset = pdoffx_ledoff*10nA 0 … Offset source is turned OFF

Figure 26:
PDOFFX_LEDON Register

0x1c: PDOFFX_LEDON				
Field	**Name**	**Rst**	**Type**	**Description**
7:0	pdoffx_ ledon	0	RW	Input offset current if at least one LED is ON (one or more sw_led* sequencer outputs are non-zero) Ioffset = pdoffx_ledon*10nA 0 … Offset source is turned OFF

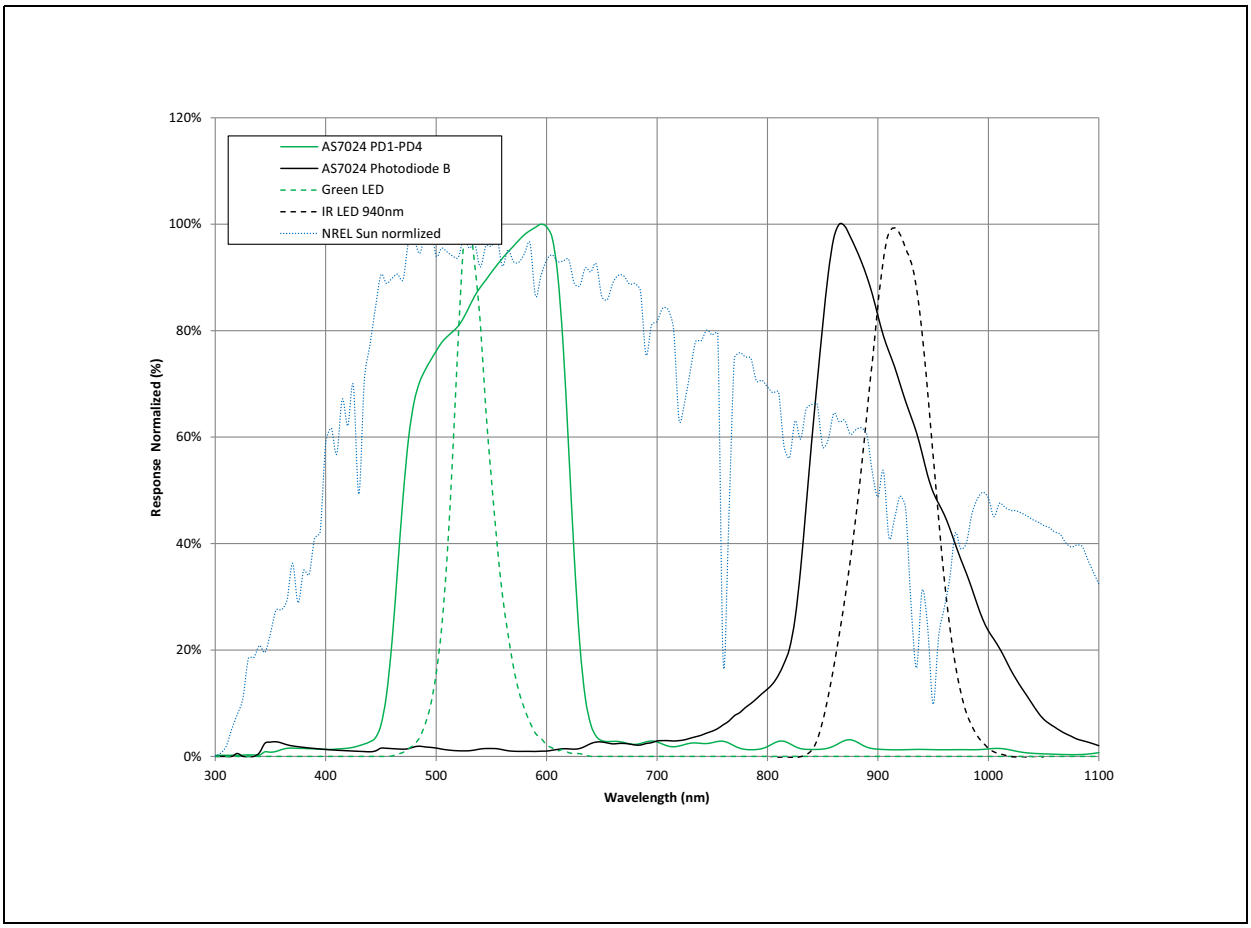
Photodiode Characteristics

Figure 27:
Photodiode Arrangement – Orientation as in Figure 2



For operation and characteristics of photodiode 'A' and photodiode 'B' see Light-to-Frequency Mode.

Figure 28:
Photodiode Sensitivity (solid green and black) and LED Emission Spectrum (dotted green and dotted black)



Note(s):

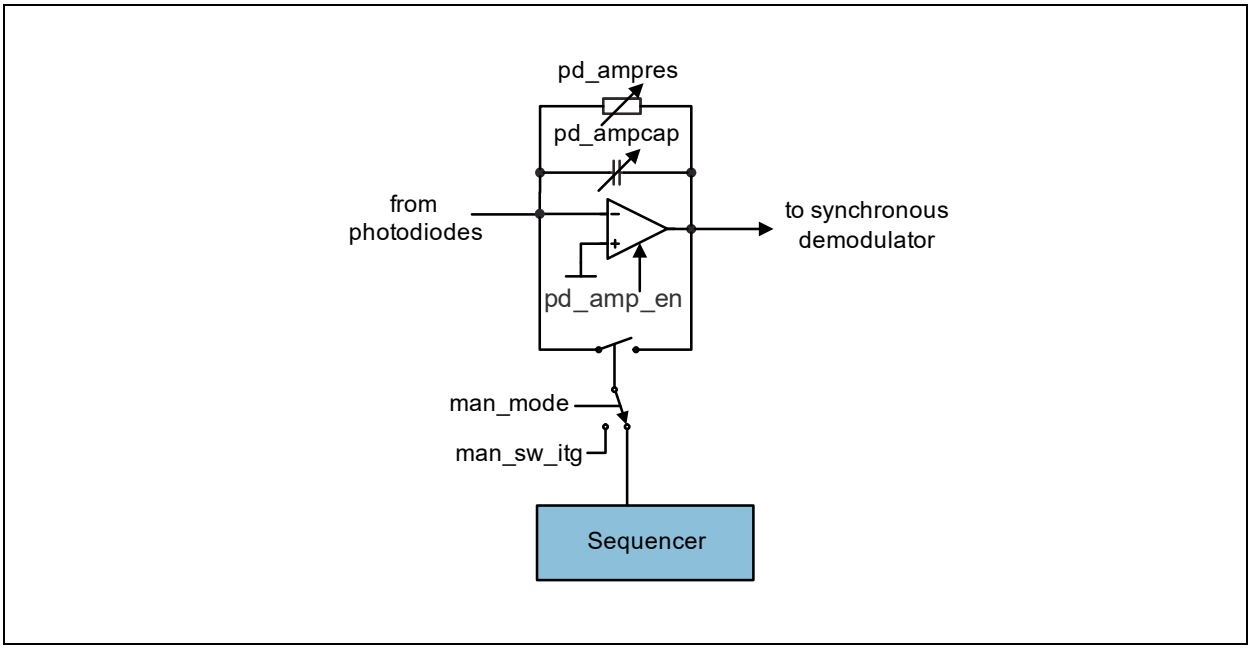
1. All 4 photodiodes used pd1/2/3/4=1;perpendicular; perpendicular light source and no diffuser used on AS7024; due to the difference in photodiode size the absolute response for Photodiode B (0.01mm^2) is much lower compared to PD1-PD4 (0.8mm^2).

Photodiode Trans-Impedance Amplifier (TIA)

The photodiode amplifier can be configured in three different modes:

- Photocurrent to frequency converter – see Light-to-Frequency Mode Registers
- Photocurrent to voltage converter
- Photocurrent integrator

Figure 29:
Trans-Impedance-Amplifier (TIA)



The integration time t_{INT} is defined either by the sequencer (man_mode=0) of manually through the bit sw_itg if man_mode=1.

Use following settings for the programming of the TIA:

Figure 30:
TIA Programming Settings

pd_ampres	pd12341	pd_ampcap	pd_ampcomp	pd_ampvo	Gain
1	1 …4	13	1	15	1V/µA
2	1 …4	7	1	15	2V/µA
3	1 …4	5	1	15	3V/µA
4	1 …2	2	0	15	5V/µA
	3 …4	3			
5	1 …2	2	0	15	7V/µA
	3 …4	3			
6	1	1	0	15	10V/µA
	2 …4	2			
7	1 …2	1	0	15	15V/µA
	3 …4	2			
Low Bandwidth Mode					
5	1 …4	31	3	15	7V/µA
Integrating Mode (pd_ampres=0)					
0	1 …4	10	3	15	1V/pQ
0	1 …4	20	3	15	1/2V/pQ
0	1 …4	30	3	15	1/3V/pQ

Note(s):

1. pd1234: number of active photodiodes (for example, pd1=1, pd2=0, pd3=1, pd4=0 -> pd1234=2).

Photodiode TIA Registers

Figure 31:
PD_AMPRCCFG Register

0x1d: PD_AMPRCCFG					
Field	**Name**	**Rst**	**Type**	**Description**	
7:5	pd_ampres	0	RW	Feedback resistor	
				Setting	**Resistance**
				0	No resistor in feedback of amplifier – photocurrent integrator
				1	1MΩ
				2	2MΩ
				3	3MΩ
				4	5MΩ
				5	7MΩ
				6	10MΩ
				7	15MΩ
4:0	pd_ampcap	0	RW	Feedback capacitor	

The PD_AMPCFG register is used to configure the operating mode of the photoamplifier.

Figure 32:
PD_AMPCFG Register

0x1e: PD_AMPCFG				
Field	**Name**	**Rst**	**Type**	**Description**
7	pd_amp_en	0	RW	0 … Activates power down mode of photo-amplifier 1 … Enables photo-amplifier (direct or automatic pd_amp_auto mode) also set en_bias_ofe=1
6	pd_amp_auto	0	RW	0 … Normal TIA mode 1 … Enable TIA only when seq_itg is set (i.e. controlled by sequencer itg setting) also set en_bias_ofe=1
5:2	pd_ampvo	15	RW	OpAmp offset. Can be used to limit signal in darkness and to shorten rise times
1:0	pd_ampcomp	3	RW	OpAmp compensation, depending on gain and number of used photo diodes Capacitor = pd_ampcap*0.1pF

Figure 33:
PD_THRESHCFG Register

0x1f: PD_THRESHCFG				
Field	**Name**	**Rst**	**Type**	**Description**
7:4	pd_ clipdetect_ h_thresh	0	RW	If the voltage on the output of the TIA exceed this threshold the irq_clipdetect interrupt is asserted. The threshold is defined as 0 … 1824mV 1 … 1748mV 2 … 1672mV 3 … 1596mV 4 … 1520mV 5 … 1444mV 6 … 1368mV 7 … 1292mV 8 … 1216mV 9 … 1140mV 10 … 1064mV 11… 988mV 12 … 912mV 13 … 836mV 14 … 760mV 15 … 684mV
3:0	pd_ clipdetect_ l_thresh	0	RW	If the voltage on the output of the TIA falls below this threshold the irq_clipdetect interrupt is asserted. The threshold is defined as 0 … 67mV 1 … 143mV 2 … 219mV 3 … 295mV 4 … 371mV 5 … 447mV 6 … 523mV 7 … 599mV 8 … 675mV 9 … 751mV 10 … 827mV 11 … 903mV 12 … 979mV 13 … 1055mV 14 … 1131mV 15 … 1207mV

Voltage Mode of the Photodiode Amplifier

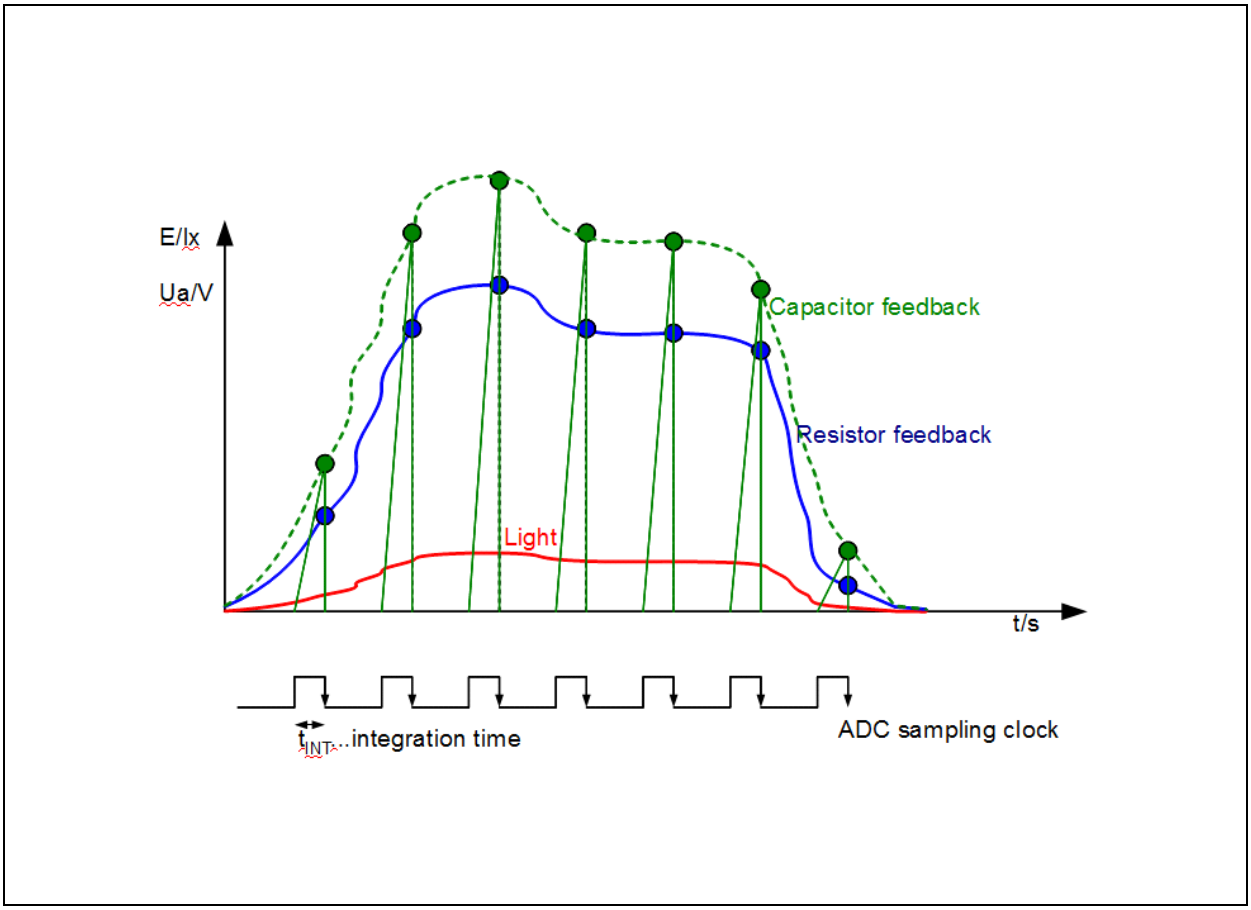
The output voltage of the photodiode amplifier is depending on the feedback component:

(EQ1) Feedback resistor: $U_{out} = I_{photo} \cdot R_{fb}$

(EQ2) Feedback capacitor: $U_{out} = I_{photo} \cdot \frac{t_{INT}}{C_{fb}}$

Note(s): The integration time t_{INT} is defined either by the sequencer (man_mode=0) of manually through the bit sw_itg if man_mode=1.
For the synchronous demodulator only use the resistive feedback.

Figure 34:
Difference Between Resistive and Capacitive Feedback



With reference to Figure 34,

- **Green:** Capacitive Integration
- **Green Dotted:** Effective Value from Capacitive Mode
- **Blue:** Resistive Feedback
- **Red:** Light Intensity

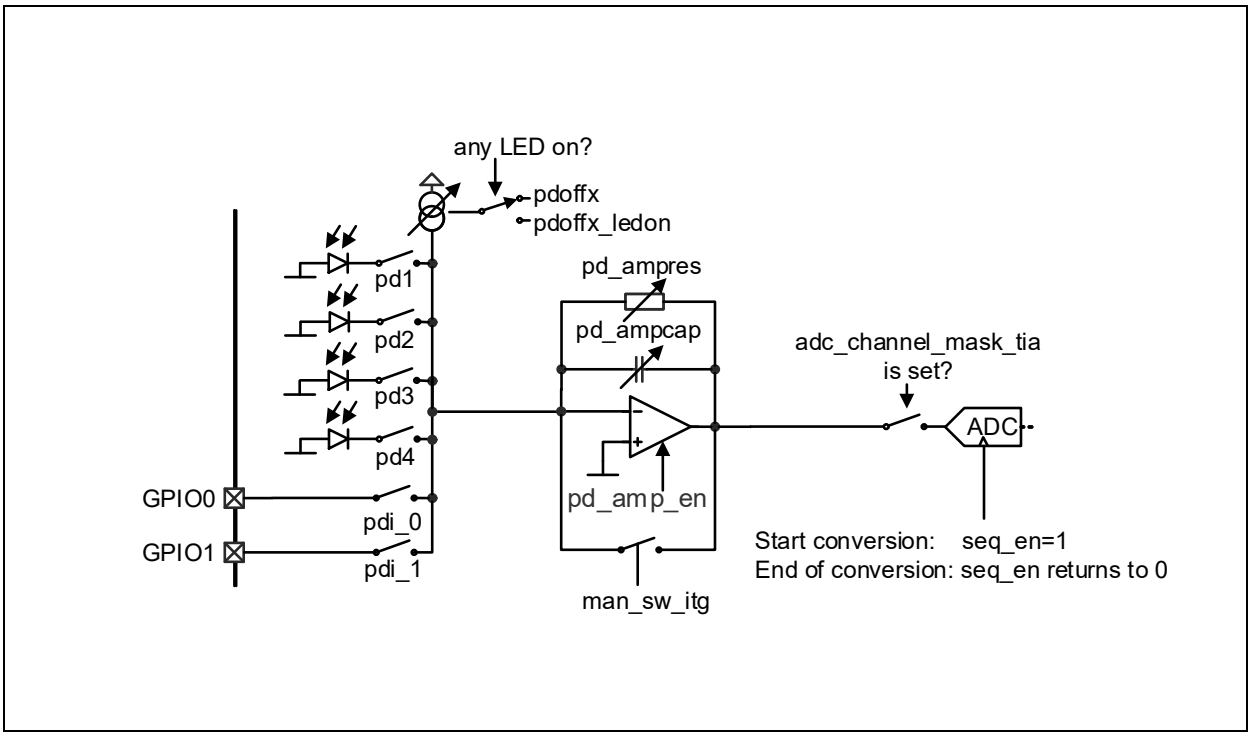
Optical Front End Operating Modes

Once the photodiode amplifier is configured the measurement can be done in two different ways. Either the LED-outputs, the photodiode amplifier and the ADC are controlled manually by means of register bits, or they are controlled by a built in sequencer.

Manual Operation of the Optical Frontend:

The optical front end can be manually controlled via the register man_mode=1

Figure 35:
Manual Operation of the Optical Frontend and LED (1)



Note(s):

1. Applies only If man_mode=1

For manual operation of the LEDs and its current sinks see LED-Driver.

Sequencer

In order to synchronize the LED-currents, the integration time and the ADC-sampling time, a built in sampling Sequencers can be used. The sequencer generates the 8 bit-timings based on a 1µs clock which can be pre-scaled with seq_div. The results of the analog to digital conversion are automatically stored in a pipeline buffer or in register adc_data and the ADC FIFO.

The timings can be programmed with following registers (apply for man_mode=0):

Figure 36:
Sequencer Control Registers Overview

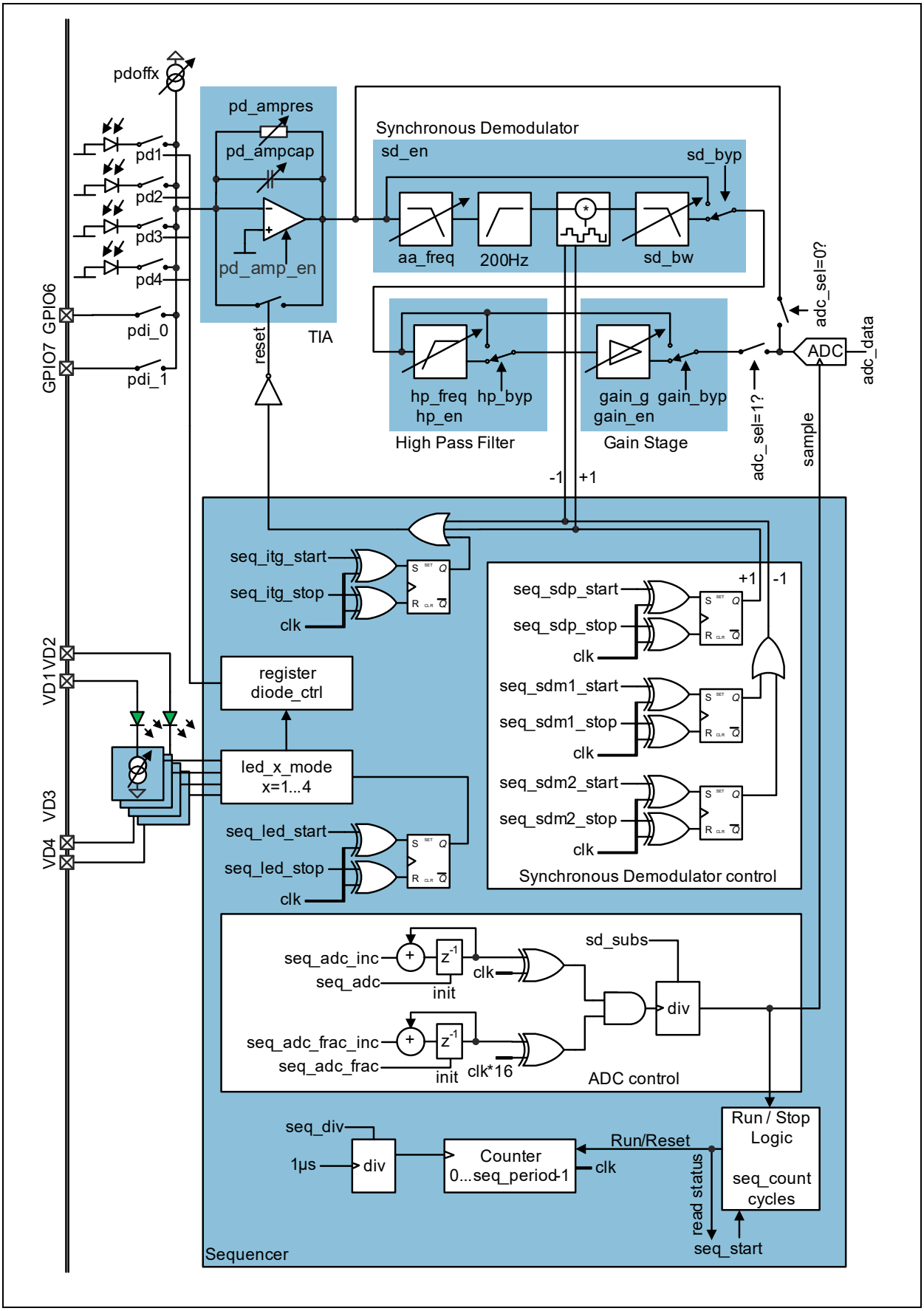
Register	Description
seq_div	Divider of the 1µs input clock for all sequencer timings
seq_count	Number of measurements in one sequence
seq_start	Writing 1 starts the sequencer, 0 stops the sequencer
seq_period	Time of one measurement cycle
seq_led_start	Start time of the LED drivers within one cycle
seq_led_stop	Stop time of the LED drivers within one cycle
seq_secled_start	Start time of the secondary LED drivers within one cycle (used for SpO2)
seq_secled_stop	Stop time of the secondary LED drivers within one cycle (used for SpO2)
seq_itg_start	Start time of the integrator
seq_itg_stop	Stop time of the integrator
seq_sdp1_start	Start time of the synchronous demodulator's 1 positive multiplication
seq_sdp1_stop	Stop time of the synchronous demodulator's 1 positive multiplication
seq_sdm1_start	Start time of the synchronous demodulator's 1 negative multiplication
seq_sdm1_stop	Stop time of the synchronous demodulator's 1 negative multiplication
seq_sdp2_start	Start time of the synchronous demodulator's 2 positive multiplication
seq_sdp2_stop	Stop time of the synchronous demodulator's 2 positive multiplication
seq_sdm2_start	Start time of the synchronous demodulator's 2 negative multiplication
seq_sdm2_stop	Stop time of the synchronous demodulator's 2 negative multiplication
seq_adc	Sampling position of the ADC

Register	Description
seq_adc2tia, seq_adc3tia	If the TIA channel is selected allow a second (and third) conversion within this cycle.
sd_subs, sd_subs_always	Synchronous demodulator subsampling ratio between sequencer frequency and ADC sampling frequency.
ulp	Ultra low power bit for the sequencer. If this bit is set and sd_subs>0, it disables the LED pulses and powers off the TIA in all sequences but the one where the TIA is sampled. This bit can be used to optimize the power consumption of the LEDs and the AS7024(2).
irq_adc_timing_error	The sequencer setup caused a timing error on ADC conversion.

Note(s):

1. The lowest data value of all registers except seq_count and seq_div is 1.
2. This bit is located in register ADC_CFGB bit 1.

Figure 37:
Block Diagram of Sequencer



Sequencer Registers

Figure 38:
SEQ_CNT Register

0x30: SEQ_CNT				
Field	**Name**	**Rst**	**Type**	**Description**
7:0	seq_count	0	RW	Number of measurements in one sequence. If seq_count = 0x0 the sequencer is running continuously if started by seq_start=1 or seq_start_sync=1. This register is reset by disabling/enabling of seq_start=0 (but not by osc_off=1)

The SEQ_DIV register sets the input divider for the main clock.

Figure 39:
SEQ_DIV Register

0x31: SEQ_DIV				
Field	**Name**	**Rst**	**Type**	**Description**
7:0	seq_div	0	RW	Divider value Sequencer time increment tclk = (seq_div + 1) * 1us

Figure 40:
SEQ_START Register

0x32: SEQ_START				
Field	**Name**	**Rst**	**Type**	**Description**
1	seq_start_ sync	0	R_PUSH	Similar to seq_start, but the sequencer will wait for overflow of the frequency divider that feeds all the switched-cap filters. This means 1) that it could take anything between 0 and 8ms before the sequencer actually starts. And 2) That the generated frequencies are in phase with the sequencer. For this to have any effect, the sequencer period should be selected with the selected frequencies (sd_bw, hp_freq) in mind.
0	seq_start	0	R_PUSH	Writing 1 starts the sequencer(s) in the according to the configuration and upon rising edge of seq_start ADC selects first channel. Writing 0 stops the sequencer(s). In manual mode, writing 1 starts one ADC conversion but does not initialize the ADC channel selection. Reading returns 1 if the sequencer is running (sequencer mode), respectively if the ADC is converting (manual mode)

With the SEQ_START register sets the configured sequencer can be started

Figure 41:
SEQ_PER Register

0x33: SEQ_PER				
Field	**Name**	**Rst**	**Type**	**Description**
7:0	seq_period	0	RW	t_period Sequencer period T = t_period * (seq_div+1) * 1us

The SEQ_PER register sets one measurement cycle of the sequencer.

Figure 42:
SEQ_LED_STA Register

0x34: SEQ_LED_STA				
Field	**Name**	**Rst**	**Type**	**Description**
7:0	seq_led_ start	0	RW	LED start time

The SEQ_LED register sets the LED drive timing. Data is stored as 16-bit value.

Figure 43:
SEQ_LED_STO Register

0x35: SEQ_LED_STO				
Field	**Name**	**Rst**	**Type**	**Description**
7:0	seq_led_ stop	0	RW	LED stop time

Figure 44:
SEQ_SECLED_STA Register

0x36: SEQ_SECLED_STA				
Field	**Name**	**Rst**	**Type**	**Description**
7:0	seq_secled_ start	0	RW	Secondary LED start time

The SEQ_LED register sets the secondary LED drive timing which is used in ledX_mode 6 only. Data is stored as 16-bit value.

Figure 45:
SEQ_SECLED_STO Register

0x37: SEQ_SECLED_STO				
Field	**Name**	**Rst**	**Type**	**Description**
7:0	seq_secled_ stop	0	RW	Secondary LED stop time

Figure 46:
SEQ_ITG_STA Register

0x38: SEQ_ITG_STA				
Field	**Name**	**Rst**	**Type**	**Description**
7:0	seq_itg_ start	1	RW	Integrator start time (start time=1 and stop time=0 means that it's - by default - always ON) Turning OFF the integrator actually means discharge the capacitor. This is only useful in capacitive integration mode, without the synchronous demodulator.

The SEQ_ITG register sets the photoamplifier integration time. Data is stored as 16-bit value.

Figure 47:
SEQ_ITG_STO Register

0x39: SEQ_ITG_STO				
Field	**Name**	**Rst**	**Type**	**Description**
7:0	seq_itg_ stop	0	RW	Integrator stop time

Figure 48:
SEQ_SDP1_STA Register

0x3a: SEQ_SDP1_STA				
Field	**Name**	**Rst**	**Type**	**Description**
7:0	seq_sdp1_ start	0	RW	Positive multiplication start time 1

The SEQ_SDP register sets the synchronous demodulator positive multiplication time. Data is stored as 16-bit value.

Figure 49:
SEQ_SDP1_STO Register

0x3b: SEQ_SDP1_STO				
Field	Name	Rst	Type	Description
7:0	seq_sdp1_ stop	0	RW	Positive multiplication stop time 1

Figure 50:
SEQ_SDP2_STA Register

0x3c: SEQ_SDP2_STA				
Field	Name	Rst	Type	Description
7:0	seq_sdp2_ start	0	RW	Positive multiplication start time 2

The SEQ_SDP register sets the synchronous demodulator positive multiplication time. Data is stored as 16-bit value

Figure 51:
SEQ_SDP2_STO Register

0x3d: SEQ_SDP2_STO				
Field	Name	Rst	Type	Description
7:0	seq_sdp2_ stop	0	RW	Positive multiplication stop time 2

Figure 52:
SEQ_SDM1_STA Register

0x3e: SEQ_SDM1_STA				
Field	Name	Rst	Type	Description
7:0	seq_sdm1_ start	0	RW	Negative multiplication start time 1

The SEQ_SDM1 register sets the synchronous demodulator negative multiplication time 1. Data is stored as 16-bit value

Figure 53:
SEQ_SDM1_STO Register

0x3f: SEQ_SDM1_STO				
Field	Name	Rst	Type	Description
7:0	seq_sdm1_ stop	0	RW	Negative multiplication stop time 1

Figure 54:
SEQ_SDM2_STA Register

0x40: SEQ_SDM2_STA				
Field	Name	Rst	Type	Description
7:0	seq_sdm2_ start	0	RW	Negative multiplication start time 2

The SEQ_SDM2 register sets the synchronous demodulator negative multiplication time 2. Data is stored as 16-bit value

Figure 55:
SEQ_SDM2_STO Register

0x41: SEQ_SDM2_STO				
Field	Name	Rst	Type	Description
7:0	seq_sdm2_ stop	0	RW	Negative multiplication stop time 2

Figure 56:
SEQ_ADC Register

0x42: SEQ_ADC				
Field	Name	Rst	Type	Description
7:0	seq_adc	0	RW	ADC sampling time The ADC conversion needs to be finished before the sequencer period ends otherwise ADC samples can be lost.

The SEQ_ADC register defines the time when the ADC starts sampling during each measurement cycle.

Figure 57:
SEQ_ADC2TIA Register

0x43: SEQ_ADC2TIA				
Field	**Name**	**Rst**	**Type**	**Description**
7:0	seq_adc2tia	0	RW	ADC second sampling time for TIA: If this time is non-zero, an ADC conversion is started at the given cycle, but only if adc_sel is currently selecting TIA. For all other channels, there is only a single ADC conversion executed in the sequencer period. Warning: If non-zero, seq_adc must be non-zero as well, and seq_adc2tia bigger than seq_adc. The difference must be high enough so that the second ADC conversion is started after the first ADC conversion has finished. Also, if the seq_adc2tia features is used, there is the additional restriction that the second ADC conversion has to be finished before the end of the sequencer period.

Figure 58:
SEQ_ADC3TIA Register

0x44: SEQ_ADC3TIA				
Field	**Name**	**Rst**	**Type**	**Description**
7:0	seq_adc3tia	0	RW	ADC third sampling time for TIA: same as seq_adc2tia. Also must make sure to not overlap ADC conversions! Also, adc3tia must be after adc2tia

Figure 59:
SD_SUBS Register

0x45: SD_SUBS				
Field	**Name**	**Rst**	**Type**	**Description**
7:0	sd_subs	0	RW	Synchronous demodulator subsampling ratio between sequencer frequency and ADC sampling frequency. ADC-Fsample = Sequencyer_Frequency/(sd_subs+1) When setting to 0, then in every sequencer iteration the ADC will run. When setting to 1, then the first sequencer iteration will not trigger the ADC, but the second one will. Setting to N will make N iterations without ADC, followed by one iteration with the ADC measurement executed. It is recommended to use the ADC interrupt in this case and not the sequencer interrupt. Also see sd_subs_always which significantly affects this mechanism.

Figure 60:
SEQ_CFG Register

0x46: SEQ_CFG				
Field	**Name**	**Rst**	**Type**	**Description**
0	sd_subs_ always	0	RW	If this bit is asserted, all sequencer periods are subject to subsampling as defined in SD_SUBS. If this bit is zero, then only the first period of an "ADC cycle" is duplicated sd_subs times, all other periods are regular. One "ADC cycle" is the time from the sequence in which adc_sel is pointing to the "smallest" adc channel up and including the sequence of the "largest" adc channel.

Figure 61:
SEQ_ERR Register

0x47: SEQ_ERR				
Field	**Name**	**Rst**	**Type**	**Description**
7	irq_adc_ timing_error	0	SS_WC	The ADC was started by the sequencer (or manually) while it was still converting. This does not flag an interrupt but when playing with the sequencer settings we suggest to check this flag to make sure that there is no problem with the sequencer programming

Figure 62:
CYC_COUNTER Register

0x60: CYC_COUNTER				
Field	**Name**	**Rst**	**Type**	**Description**
7:0	cycle_ counter	0	RO	Current cycle counter value

The SEQ_COUNTER register shows the current value of the sequence counter and period counter

Figure 63:
SEQ_COUNTER Register

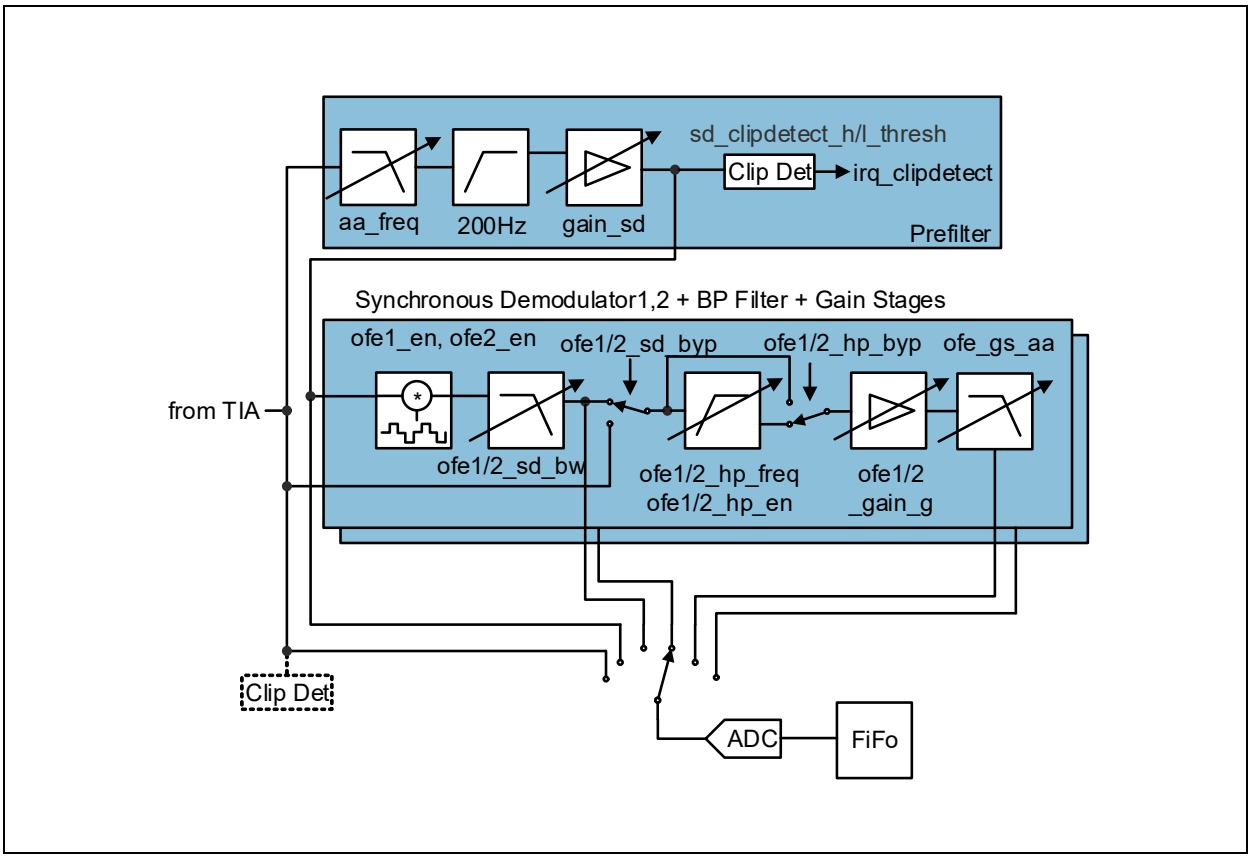
0x61: SEQ_COUNTER				
Field	**Name**	**Rst**	**Type**	**Description**
7:0	sequence_ counter	0	RO	Current sequence counter value

Figure 64:
SUBS_COUNTER Register

0x62: SUBS_COUNTER				
Field	**Name**	**Rst**	**Type**	**Description**
7:0	subs_ counter	0	RO	Current subsampling counter value

Optical Signal Conditioning

Figure 65:
Optical Signal Conditioning



Synchronous Demodulator

Two optional synchronous demodulators can be used to detect small optical signals in the presence of large unwanted noise (ambient light). Since the detector synchronizes to the LED frequency, the demodulator can only be used of the measurement sequencer is running.

It includes input filer (high pass at 200Hz, adjustable low pass) and an 2nd order adjustable output low pass. The demodulator itself multiplies the signal by +1 / 0 / -1 with a timing which is controlled by the sequencer.

Note(s): The optical signal conditioning stage need sigref_ en=1 for operation.

High Pass Filter

Two optional high pass filter can be used to remove unwanted DC-components from the signal and allows further amplification. In order to guarantee fast settling times of the filter, four cutoff frequencies can be chosen.

Gain Stage

Two optional gain stage can be used to amplify the signal after the DC-component has been removed.

Optical Signal Conditioning Registers

Figure 66:
OFE_CFGA Register

0x50: OFE_CFGA					
Field	**Name**	**Rst**	**Type**	**Description**	
7	ofe2_en	0	RW	Enable OFE2	
6	ofe1_en	0	RW	Enable OFE1	
5	en_bias_ofe	0	RW	Enable bias for OFE and TIA	
4:3	aa_freq	0	RW	Anti-aliasing filter cut-OFF frequency	
				Setting	**Signal**
				0	10kHz
				1	20kHz
				2	40kHz
				3	60kHz
2:0	gain_sd	0	RW	SD gain	
				Setting	**Nominal Gain**
				0	1
				1	2
				2	4
				3	8
				4	16
				5	32
				6	64
				7	reserved

Figure 67:
OFE_CFGB Register

0x51: OFE_CFGB				
Field	**Name**	**Rst**	**Type**	**Description**
7:4	sd_ clipdetect_ h_thresh	0	RW	If the voltage on the output of the gain_sd stage (input of synchronous demodulator) exceed this threshold the irq_ clipdetect interrupt is asserted. The threshold is defined as: 0 … 1824mV 1 … 1748mV 2 … 1672mV 3 … 1596mV 4 … 1520mV 5 … 1444mV 6 … 1368mV 7 … 1292mV 8 … 1216mV 9 … 1140mV 10 … 1064mV 11 … 988mV 12 … 912mV 13 … 836mV 14 … 760mV 15 … 684mV
3:0	sd_ clipdetect_ l_thresh	0	RW	If the voltage on the output of the gain_sd stage (input of synchronous demodulator) falls below this threshold the irq_clipdetect interrupt is asserted. The threshold is defined as: 0 … 67mV 1 … 143mV 2 … 219mV 3 … 295mV 4 … 371mV 5 … 447mV 6 … 523mV 7 … 599mV 8 … 675mV 9 … 751mV 10 … 827mV 11 … 903mV 12 … 979mV 13 … 1055mV 14 … 1131mV 15 … 1207mV

Figure 68:
OFE_CFGC Register

0x52: OFE_CFGC				
Field	**Name**	**Rst**	**Type**	**Description**
6	prefilter_aa_byp	0	RW	0 … Anti aliasing filter (aa_filter) is used 1 … Bypass anti aliasing filter
5	prefilter_hp_byp	0	RW	0 … Use 200Hz high pass filter 1 … Bypass 200Hz high pass filter
4	prefilter_gain_byp	0	RW	0 … Use gain_sd stage 1 … Bypass gain_sd stage
3	prefilter_bypass_en	0	RW	0 … Use prefilter unless any of the above register is set 1 … Bypass complete prefilter
2	prefilter_aa_en	0	RW	0 … Anti aliasing filter (aa_filter) is OFF 1 … Anti aliasing filter is ON
1	prefilter_hp_en	0	RW	0 … 200Hz high pass filter is OFF 1 … 200Hz high pass filter is ON
0	prefilter_gain_en	0	RW	0 … gain_sd stage is OFF 1 … gain_sd stage is ON

Figure 69:
OFE_CFGD Register

0x53: OFE_CFGD					
Field	**Name**	**Rst**	**Type**	**Description**	
1:0	ofe_gs_aa	0	RW	OFE anti aliasing	
				Setting	**Nominal Gain**
				0	Bypass
				1	fc=100Hz
				2	fc=300Hz
				3	fc=826Hz

Figure 70:
OFE1_CFGA Register

0x54: OFE1_CFGA				
Field	**Name**	**Rst**	**Type**	**Description**
7	ofe1_sd_pol_init	0	RW	The low level driver shall ensure that this register is 0 if one of the seq_sdm pulses is first, and is 1 if the seq_sdp is first within a sequence.
6	ofe1_sd_en	0	RW	0 … Power down of the Synchronous demodulator 1 … Enable Synchronous demodulator
5	ofe1_hp_en	0	RW	0 … Power down of the high pass filter 1 … Enable high pass filter
4	ofe1_gain_en	0	RW	0 … Power down of the Gain stage 1 … Enable Gain stage
3	ofe1_sd_byp	0	RW	0 … Synchronous demodulator is used 1 … Synchronous demodulator is bypassed
2	ofe1_hp_byp	0	RW	0 … HP filter is used 1 … HP filter is bypassed
1	ofe1_gain_byp	0	RW	0 … Gain stage is used 1 … Gain stage is bypassed
0	ofe1_sd_hld	0	RW	SD hold 0 … Output of synchronous demodulator is forced to SIGREF if not set to +1 or -1 1… Output of synchronous demodulator is tristated if not set to +1 or -1

Figure 71:
OFE1_CFGB Register

0x55: OFE1_CFGB					
Field	**Name**	**Rst**	**Type**	**Description**	
6:4	ofe1_gain_g	0	RW	Gain	
				Setting	**Gain**
				0	1
				1	2
				2	4
				3	8
				4	16
				5	32
				6	64
				7	Do not use
3:2	ofe1_sd_bw	0	RW	Synchronous demodulator low pass filter.	
				Setting	**Frequency**
				0	10Hz
				1	20Hz
				2	40Hz
				3	80Hz
1:0	ofe1_hp_freq	0	RW	High pass filter cutoff frequency	
				Setting	**Cutoff Frequency**
				0	0.33Hz
				1	1.32Hz
				2	5.28Hz
				3	10.56Hz

Figure 72:
OFE2_CFGA Register

0x58: OFE2_CFGA				
Field	**Name**	**Rst**	**Type**	**Description**
7	ofe2_sd_pol_init	0	RW	The low level driver shall ensure that this register is 0 if one of the seq_sdm pulses is first, and is 1 if the seq_sdp is first within a sequence.
6	ofe2_sd_en	0	RW	0 … Power down of the synchronous demodulator 1 … Enable synchronous demodulator
5	ofe2_hp_en	0	RW	0 … Power down of the high pass filter 1 … Enable high pass filter
4	ofe2_gain_en	0	RW	0 … Power down of the Gain stage 1 … Enable Gain stage
3	ofe2_sd_byp	0	RW	0 … Synchronous demodulator is used 1 … Synchronous demodulator is bypassed
2	ofe2_hp_byp	0	RW	0 … HP filter is used 1 … HP filter is bypassed
1	ofe2_gain_byp	0	RW	0 … Gain stage is used 1 … Gain stage is bypassed
0	ofe2_sd_hld	0	RW	SD hold 0 … Output of synchronous demodulator is forced to SIGREF if not set to +1 or -1 1 … Output of synchronous demodulator is tristated if not set to +1 or -1

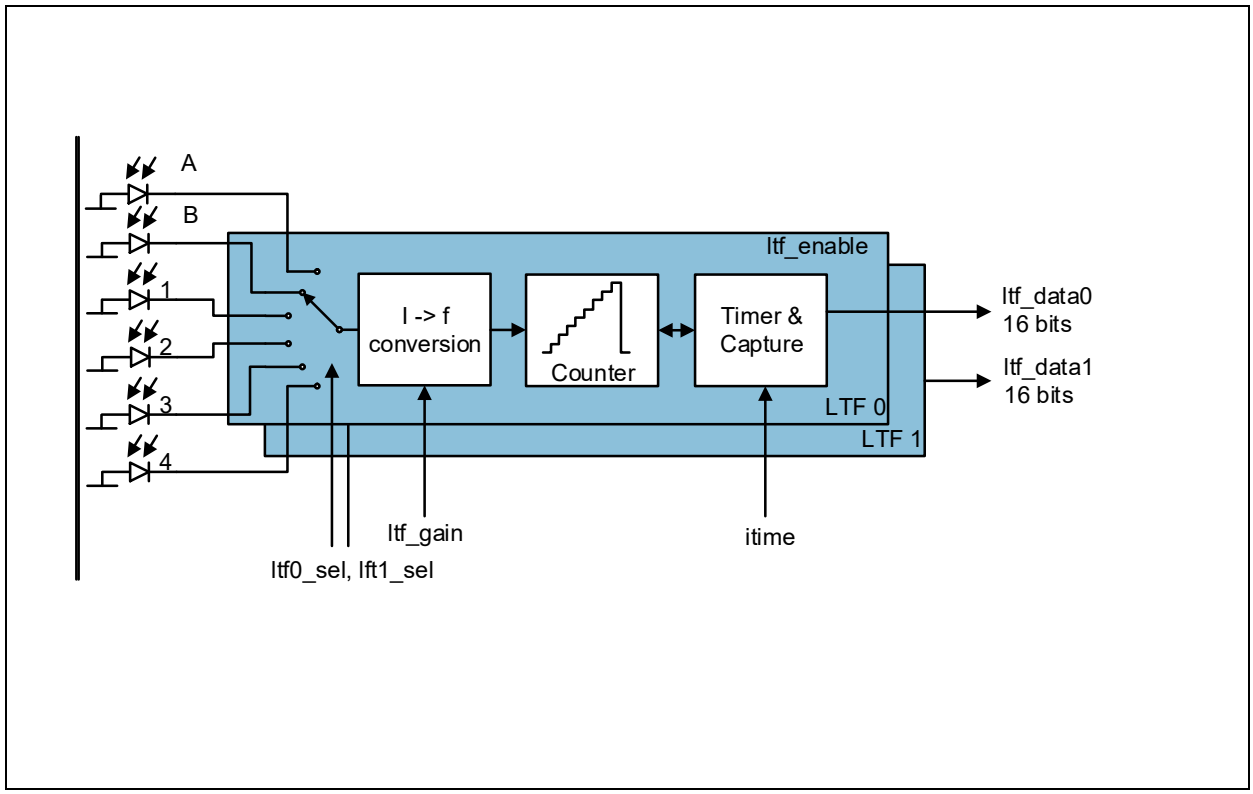
Figure 73:
OFE2_CFGB Register

0x59: OFE2_CFGB					
Field	**Name**	**Rst**	**Type**	**Description**	
6:4	ofe2_gain_g	0	RW	Gain	
				Setting	**Gain**
				0	1
				1	2
				2	4
				3	8
				4	16
				5	32
				6	64
				7	Do not use
3:2	ofe2_sd_bw	0	RW	Synchronous demodulator low pass filter.	
				Setting	**Frequency**
				0	10Hz
				1	20Hz
				2	40Hz
				3	80Hz
1:0	ofe2_hp_freq	0	RW	High pass filter cutoff frequency	
				Setting	**Cutoff Frequency**
				0	0.33Hz
				1	1.32Hz
				2	5.28Hz
				3	10.56Hz

Light-to-Frequency Mode

The LTF (light-to-frequency, or FM, frequency mode) mode.

Figure 74:
Light-to-Frequency Mode Internal Circuit



Note(s):

1. Do not use diodes which are connected to the TIA (register pd_a, pd_b, pd1...4) at the same time when ltf_en is enabled on the same diode.

Light-to-Frequency Mode Registers

Figure 75:
LTFDATA0_L Register

0x20: LTFDATA0_L				
Field	**Name**	**Rst**	**Type**	**Description**
7:0	ltfdata0[7:0]	0	RO	LTF result channel 0 low byte. Software must make sure that the LTF integration is not running when accessing the LTFDATA registers. These are the direct counter registers, they are not latched. If buffering is required, consider using FIFO mode.

Figure 76:
LTFDATA0_H Register

0x21: LTFDATA0_H				
Field	**Name**	**Rst**	**Type**	**Description**
7:0	ltfdata0[15:8]	0	RO	LTF result channel 0 high byte

Figure 77:
LTFDATA1_L Register

0x22: LTFDATA1_L				
Field	**Name**	**Rst**	**Type**	**Description**
7:0	ltfdata1[7:0]	0	RO	LTF result channel 1 low byte. Software must make sure that the LTF integration is not running when accessing the LTFDATA registers. If buffering is required, consider using FIFO mode.

Figure 78:
LTFDATA1_H Register

0x23: LTFDATA1_H				
Field	**Name**	**Rst**	**Type**	**Description**
7:0	ltfdata1[15:8]	0	RO	LTF result channel 1 high byte

Figure 79:
ITIME Register

0x24: ITIME				
Field	**Name**	**Rst**	**Type**	**Description**
7:0	itime	0	RW	LTF integration time. MODCLK is 2/3MHz (666.67kHz). One LSB of itime is 3.072ms (2048 MODCLK cycles). 0=3.072ms ... 255=786.432ms Using the itime_unit register (see below), the unit of itime can be reduced by 2, 4, or 8. This shorter integration times can be selected (required for flicker detection), but it can also be used to increase the resolution of itime. For example, if 50ms integration time are desired, the best value for regular itime would be 15 (=16 periods=49.152ms). However, but setting itime_unit=2 (LSB=768µs), one can select 64 (=65 periods=49.9ms) Warning: selecting an integration time smaller than 3.072ms will reduce the resolution of the conversion, as the maximum ltfdata value is not 1024 (10 bits) anymore, but 512 (9 bits) in case of 1.536ms integration time, 256 (8 bits) for 768µs and 128 (7 bits) for 384µs

Figure 80:
LTF_CONFIG Register

0x25: LTF_CONFIG				
Field	**Name**	**Rst**	**Type**	**Description**
7	infinite_ itime	0	RW	If this is asserted, then integration does not stop. The ITIME setting is ignored. Use with watch the ltfdata counters. (Warning: must be filtered in software to prevent inconsistent upper/lower byte). It's implemented as a count disable on the integration counter, so when resetting bit to 0 again, the itime counter will continue and results can be read afterwards through the regular mechanisms (ltfdata or FIFO) This is intended for very long integration times - as the timing is controlled by software/I^2C, accuracy fully depends on the system and I^2C master.
6	az_disable_ auto	0	RW	0: Run autozero on both channels every time FM mode is activated for the first time after ENAB is being asserted. 1: Do not run autozero automatically. Autozero can only be activated manually (AZ_CONTROL)
5:4	reserved	0	RW	Reserved – leave at 0.
1	ltf_fifo_ mode	0	RW	Run LTF integrations back to back, the LTF modulator is running continuously (the modulators are not reset between integrations cycles). After each integration, the result gets written to the FIFO. The FIFO is being filled automatically, FIFO threshold interrupt is flagged as configured. The first item read from the FIFO is from channel 0, the next one from channel 1, etc. Note that there is no ltf_done interrupt triggered after each integration. A FIFO threshold of 1 can be used to generate an interrupt for each result. irq_ltf_enab should be kept asserted to avoid missing an ltf_sat interrupt. Do not enable ADC/sequencer FIFO mode and ltf_fifo_ mode at the same time, corrupted data would be the result. Make sure to empty the FIFO in time, if the FIFO is full, new data is not being stored in the FIFO. Source of data read from the FIFO after an overflow condition is undefined (can be from channel 0 or channel 1) Stop the procedure by clearing this bit.
0	ltf_enable	0	RW	This bit must be asserted for any LTF function (powers up the LTF clock tree)

Figure 81:
LTF_SEL Register

0x26: LTF_SEL					
Field	**Name**	**Rst**	**Type**	**Description**	
6:4	ltf1_sel	2	RW	Select the sensor diode for LTF1	
				Setting	**Source**
				0	A
				1	A/16
				2	B
				3	B/16
				4	PD1
				5	PD2
				6	PD3
				7	PD4
2:0	ltf0_sel	0	RW	Select the sensor diode for LTF0	
				Setting	**Source**
				0	A
				1	A/16
				2	B
				3	B/16
				4	PD1
				5	PD2
				6	PD3
				7	PD4

Figure 82:
LTF_GAIN Register

0x27: LTF_GAIN					
Field	**Name**	**Rst**	**Type**	**Description**	
5:4	itime_unit	0	RW	Select the itime unit. See itime register description.	
				Setting	**Behavior**
				0	Normal, time LSB=3.072ms
				1	/2, LSB=1.536ms
				2	/4, time LSB=768µs
				3	/8, time LSB=384µs
3:0	ltf_gain	0	RW	Select the gain	
				Setting	**Gain**
				0	0.25
				1	0.5
				2	1
				3	2
				4	4
				5	8
				6	16
				7	32
				8	64
				9-15	Reserved – do not use

Figure 83:
LTF_CONTROL Register

0x28: LTF_CONTROL				
Field	**Name**	**Rst**	**Type**	**Description**
0	ltf_start	0	R_PUSH	Writing 1 starts the counter, and it will run for the specified time (itime). Afterwards it stops automatically and interrupt is flagged. writing 0 to the counter stops it as well. reading the value returns whether the counter is running. If ltf_fifo_mode is non-zero, then FM conversions are done continuously until a 0 is written to this bit again.

Figure 84:
AZ_CONTROL Register

0x29: AZ_CONTROL				
Field	**Name**	**Rst**	**Type**	**Description**
1	az_enable_1	0	RW_SM	Writing a '1' to this register starts the AZ engine for channel 1. This is usually not necessary, as AZ is executed automatically before the first LTF integration (unless az_disable_auto is set) The bit is cleared to '0' automatically when the AZ has finished. You cannot write a '0' to this register.
0	az_enable_0	0	RW_SM	The same as az_enable_1, but for channel 0.

Figure 85:
OFFSET0 Register

0x2a: OFFSET0				
Field	**Name**	**Rst**	**Type**	**Description**
7:0	offset0[7:0]	0	RW_SM	This register holds the value of the offset on the channel 0 OpAmp. It can be overwritten, and it gets overwritten by the auto-zero mechanism. The value is in sign/magnitude encoding. The value is ±127, sign/magnitude

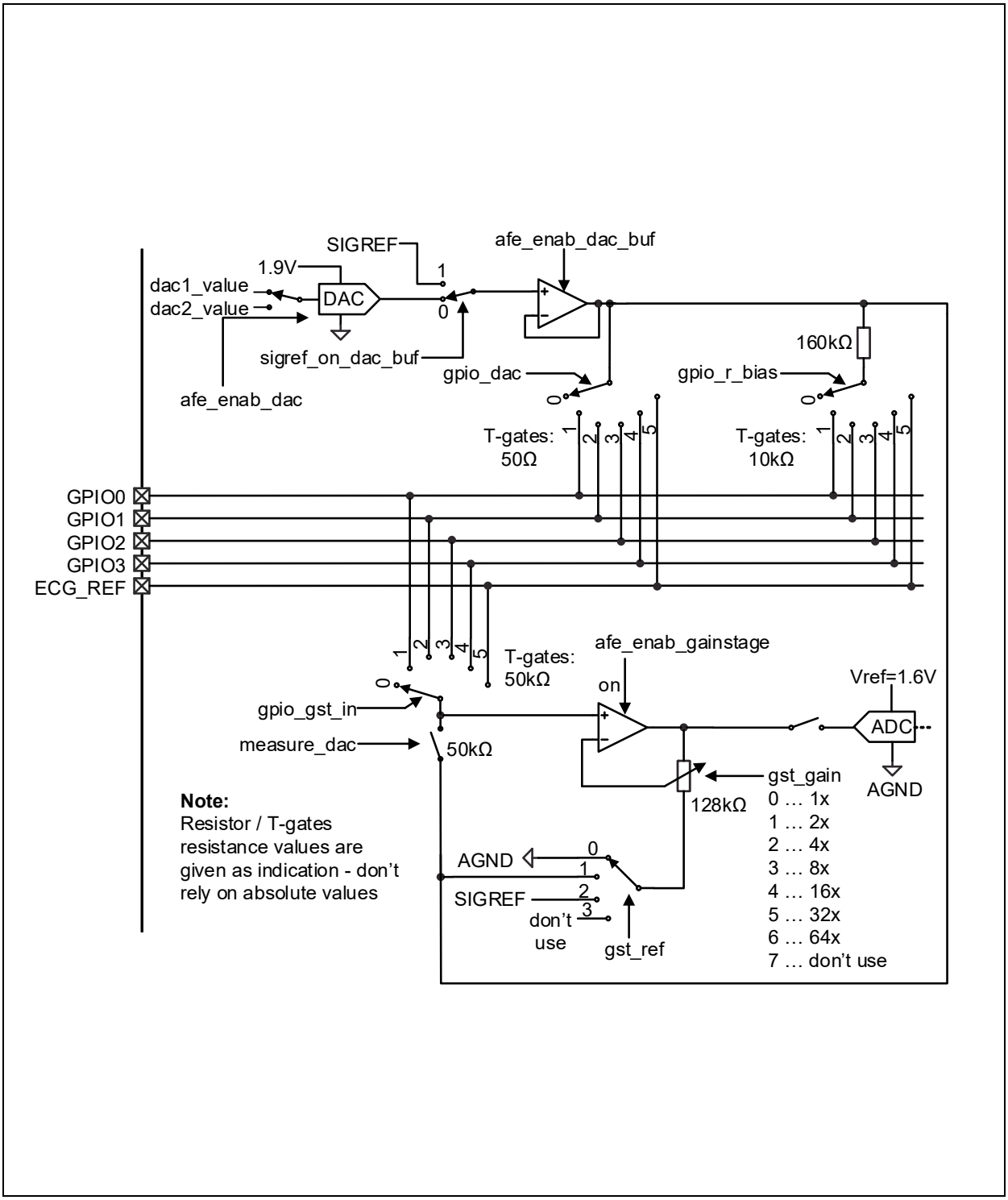
Figure 86:
OFFSET1 Register

0x2b: OFFSET1				
Field	**Name**	**Rst**	**Type**	**Description**
7:0	offset1[7:0]	0	RW_SM	This register holds the value of the offset on the channel 1 OpAmp. It can be overwritten, and it gets overwritten by the auto-zero mechanism. The value is in sign/magnitude encoding. The value is ± 127, sign/magnitude

Electrical Analog Front End

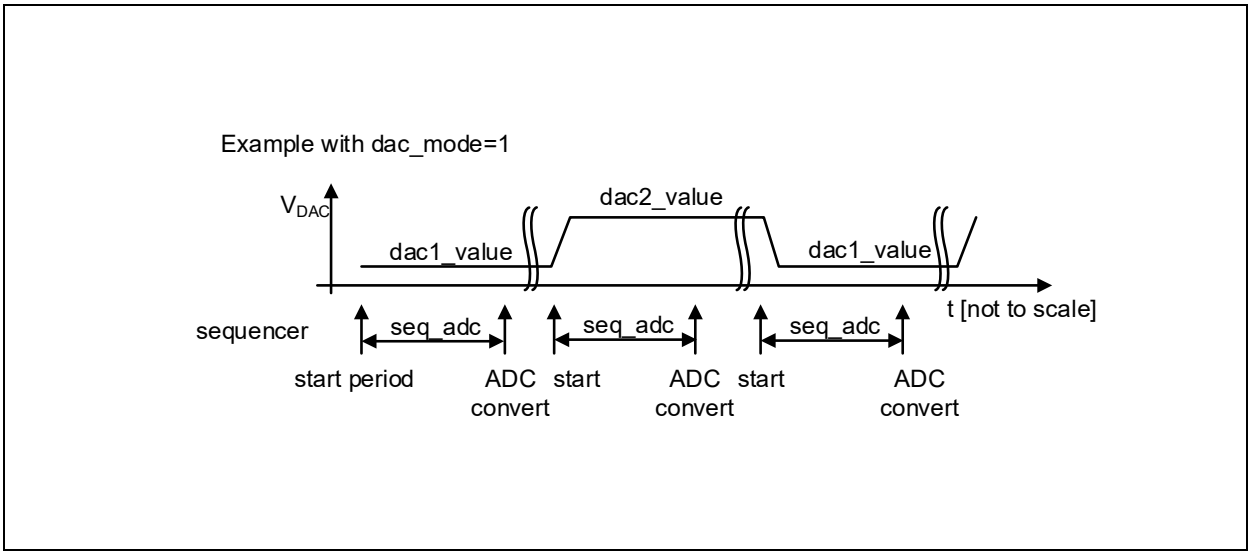
The electrical analog front end consists of three identical signal paths with independent settings of bias condition, gain and offset.

Figure 87:
Electrical Analog Front End Internal Circuit



DAC Switching

Figure 88:
Electrical Analog Front End DAC Level Switching



If register dac_mode is not zero, the DAC switches its codes between dac1_value and dac2_value on the beginning of every/every 2nd/every 4th sequencer cycle where the ADC is converting the electrical frontend channel. ADC conversions of any other channel do not switch the DAC.

Input Pins

Four general purpose pins and ECG_REF can be used either as configurable GPIO pin or as analog input pins for the electrical analog front end. The analog inputs can be configured to setup different amplifier topologies.

EAF (Electrical Analog Frontend) Registers

Figure 89:
AFE_CFG Register

0x70: AFE_CFG				
Field	**Name**	**Rst**	**Type**	**Description**
3	afe_enab	0	RW	0 … EAF bias deactivated 1 … EAF bias activated (need to be set for any functions of the EAF are used).
2	afe_enab_dac	0	RW	0 … DAC inside the EAF OFF 1 … DAC inside the EAF ON
1	afe_enab_dac_buf	0	RW	0 … DAC buffer ON 1 … DAC buffer OFF
0	afe_enab_gainstage	0	RW	0 … Gain stage in EAF OFF 1 … Gain stage in EAF ON

The AFE_CFG register is used to configure the analog frontend.

Figure 90:
EAF_GST Register

0x80: EAF_GST					
Field	**Name**	**Rst**	**Type**	**Description**	
7:5	gpio_gst_in	0	RW	Gain stage input selection	
				Setting	**Meaning**
				0	Not connected
				1	GPIO0
				2	GPIO1
				3	GPIO2
				4	GPIO3
				5	ECG_REF
4:3	gst_ref	0	RW	Gain stage reference voltage	
				Setting	**Meaning**
				0	AGND
				1	DAC buffer
				2	SIGREF
				3	Reserved
2:0	gst_gain	0	RW	Gain stage gain	
				Setting	**Meaning**
				0	1
				1	2
				2	4
				3	8
				4	16
				5	32
				6	64
				7	Reserved

The EAF register is used to configure the electrical frontend

Figure 91:
EAF_BIAS Register

0x81: EAF_BIAS					
Field	**Name**	**Rst**	**Type**	**Description**	
7:5	gpio_r_bias	0	RW	Resistive biasing	
				Setting	**Meaning**
				0	No resistive biasing
				1	Resistive biasing on GPIO0
				2	Resistive biasing on GPIO1
				3	Resistive biasing on GPIO2
				4	Resistive biasing on GPIO3
				5	Resistive biasing on ECG_REF

Figure 92:
EAF_DAC Register

0x82: EAF_DAC					
Field	**Name**	**Rst**	**Type**	**Description**	
4	sigref_on_dac_buf	0	RW	If asserted, connect SIGREF to DAC buffer.	
3	measure_dac	0	RW	If this bit is asserted, the DAC output is connected to the gain stage input (independent of gpio_gst_in selection, therefore the DAC output is measurable on the GPIO pin)	
2:0	gpio_dac	0	RW	DAC on GPIO	
				Setting	**Meaning**
				0	No DAC biasing
				1	DAC on GPIO0
				2	DAC on GPIO1
				3	DAC on GPIO2
				4	DAC on GPIO3
				5	DAC on ECG_REF

Figure 93:
EAF_DAC1_L Register

0x83: EAF_DAC1_L				
Field	Name	Rst	Type	Description
7:6	dac1_ value[1:0]	0	RW	DAC value 1 (2LSB)

The EAF_DAC1/2_L/H registers is used to configure the dac value. See register dac_mode for selection of dac register 1 or 2.

Figure 94:
EAF_DAC1_H Register

0x84: EAF_DAC1_H				
Field	Name	Rst	Type	Description
7:0	dac1_ value[9:2]	0	RW	DAC value 1 (upper 8 bits) 10 bit value: 0x000 … 0V 0x3FF … 1.9V

Figure 95:
EAF_DAC2_L Register

0x85: EAF_DAC2_L				
Field	Name	Rst	Type	Description
7:6	dac2_ value[1:0]	0	RW	DAC value 2 (2LSB)

Figure 96:
EAF_DAC2_H Register

0x86: EAF_DAC2_H				
Field	Name	Rst	Type	Description
7:0	dac2_ value[9:2]	0	RW	DAC value 2 (upper 8 bits) 10 bit value: 0x000 … 0V 0x3FF … 1.9V

Figure 97:
EAF_DAC_CFG Register

<table>
<tr><th colspan="5">0x87: EAF_DAC_CFG</th></tr>
<tr><th>Field</th><th>Name</th><th>Rst</th><th>Type</th><th>Description</th></tr>
<tr><td rowspan="6">1:0</td><td rowspan="6">dac_mode</td><td rowspan="6">0</td><td rowspan="6">RW</td><td>DAC mode The EAF has a DAC that can be switched out on GPIOs.
dac_mode 0 uses statically dac1_value, the other modes switch dynamically between the two values.
The system switches from one value to the next always at the beginning of a sequence in which the ADC will sample the AFE channel.</td></tr>
<tr><td>
<table>
<tr><th>Setting</th><th>Meaning</th></tr>
<tr><td>0</td><td>1-1-1-1-1-1-1-1-1-1-</td></tr>
<tr><td>1</td><td>1-2-1-2-1-2-1-2-1-2-</td></tr>
<tr><td>2</td><td>1-1-2-2-1-1-2-2-1-1-2-</td></tr>
<tr><td>3</td><td>1-1-1-1-2-2-2-2-1-1-1-</td></tr>
</table>
</td></tr>
</table>

Possible Configurations of Every Amplifier Stage

Figure 98:
Non Inverting Amplifier with Offset and Input Voltage Divider (Temperature Sensor)

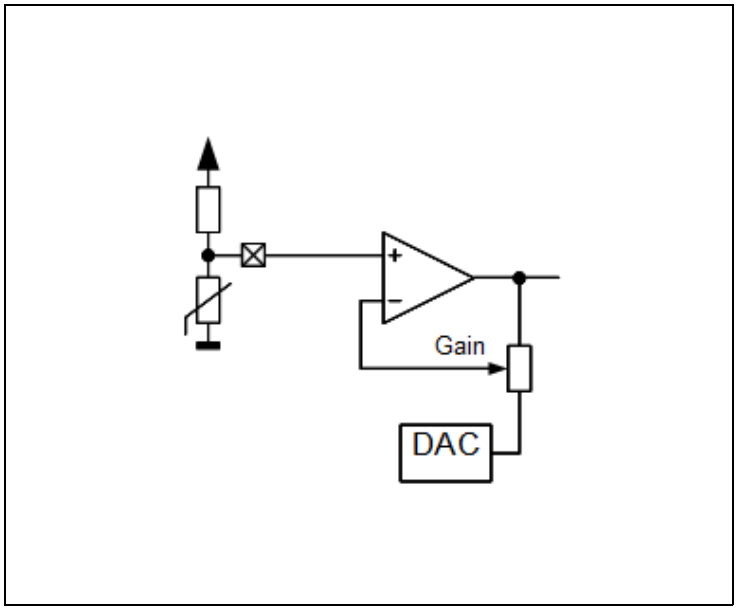


Figure 99:
Non Inverting Amplifier with Current Source and Offset (Temperature Sensor)

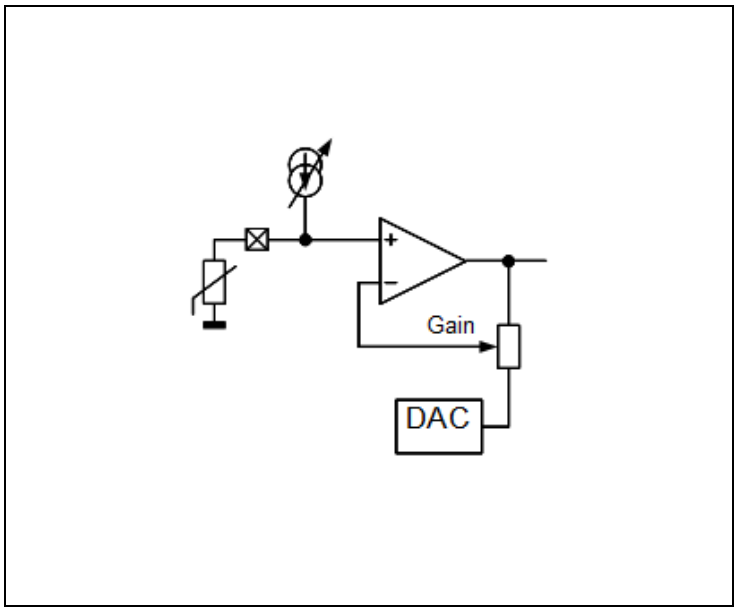


Figure 100:
Non Inverting Amplifier with Current Source and Reference Path (Temperature Sensor)

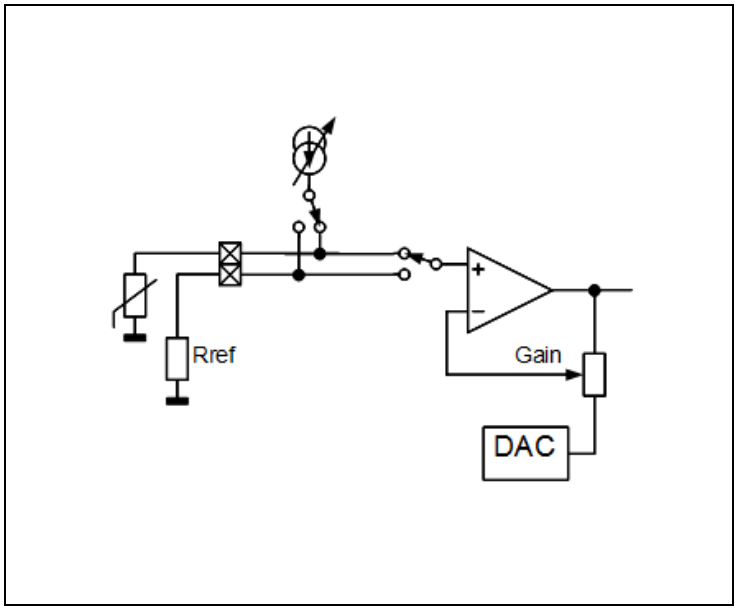


Figure 101:
Non Inverting Amplifier High Impedance, GND Referenced

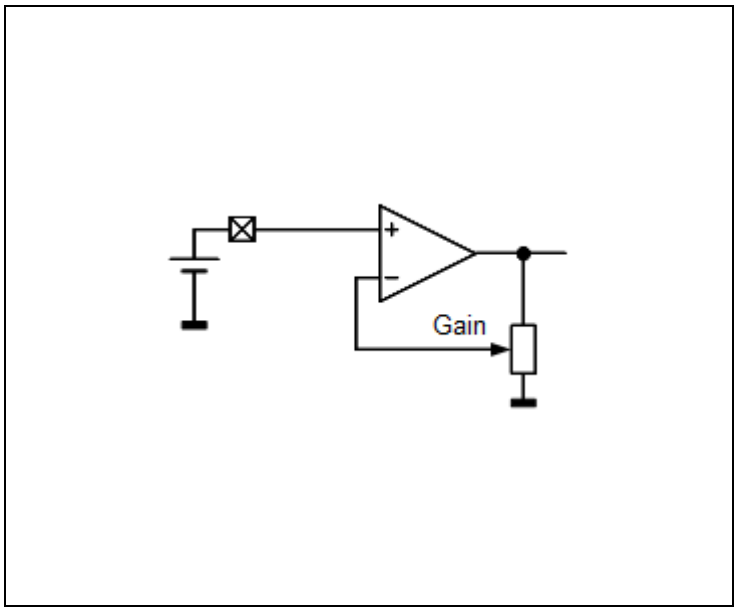


Figure 102:
Non Inverting Amplifier with DC-Blocking, Referenced to V_ADCRef/2

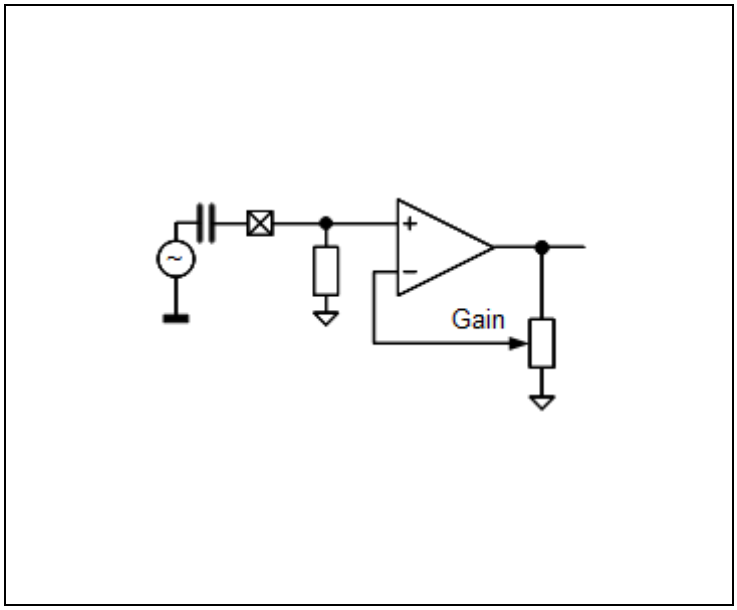
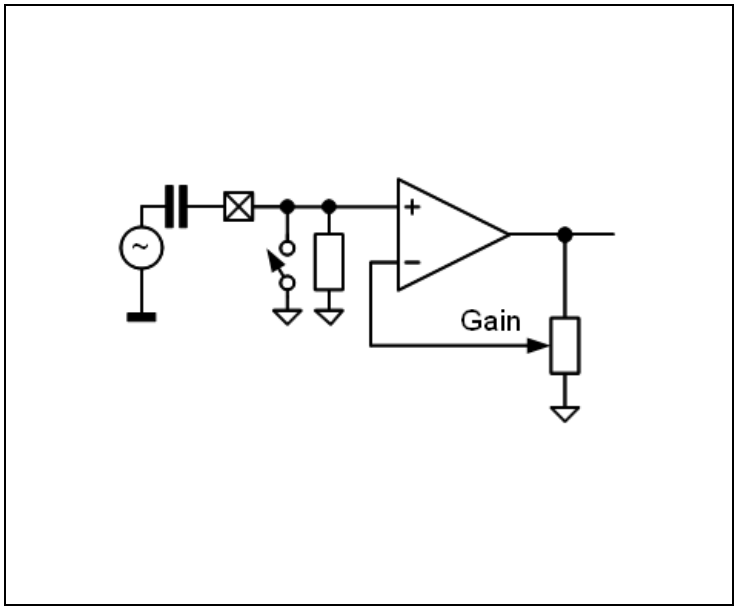
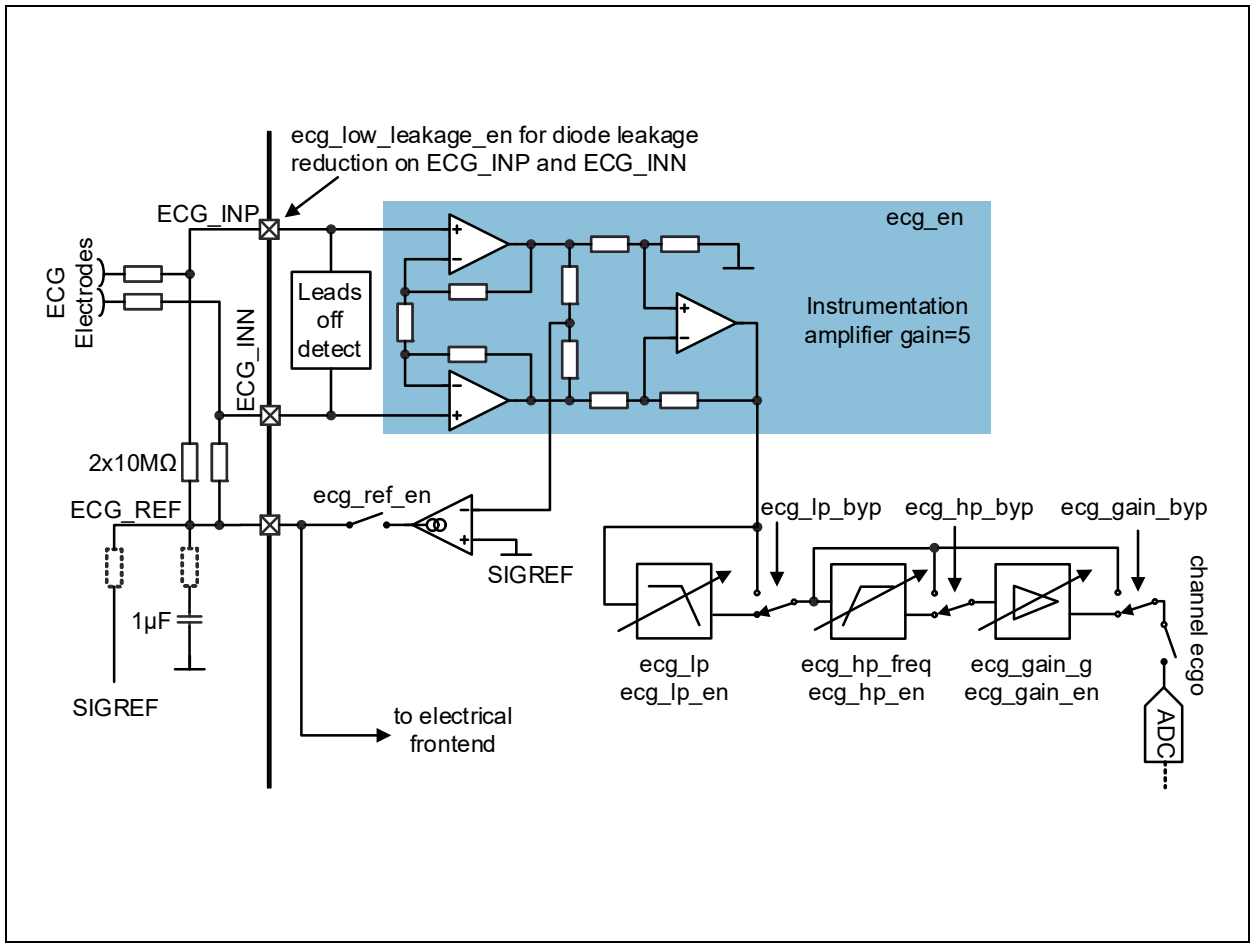


Figure 103:
Non Inverting Amplifier with DC-Blocking and Fast Settling Time, Referenced to ADCRef /2



ECG Amplifier

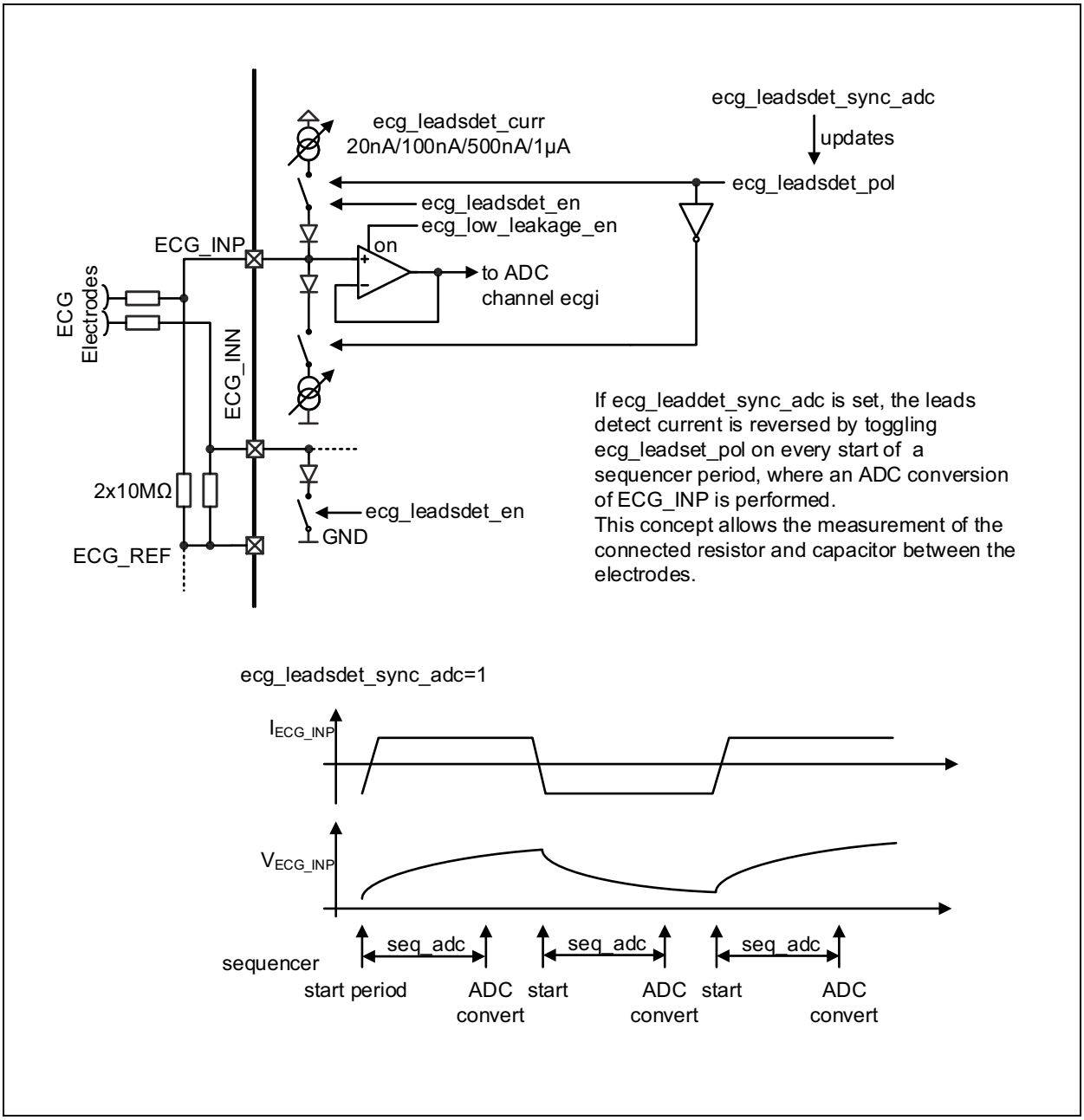
Figure 104:
ECG Amplifier Internal Circuit



The ECG (electro cardiogram) amplifier is a high impedance, low noise instrumentation amplifier with analog circuitry to bandpass filter the signal and amplify it before converting it with the ADC.

ECG Lead OFF Detection

Figure 105:
ECG Lead OFF Detection Internal Circuit



The ECG lead OFF detection can be used for detection if the user actually touches the leads. It is a circuitry to measure the capacitor and/or resistance between the two lead inputs ECG_ INP and ECG_INN.

ECG Registers

Figure 106:
ECG_CFGA Register

0x5c: ECG_CFGA				
Field	Name	Rst	Type	Description
7	ecg_en	0	RW	Enable ECG instrumentation amplifier
6	not used	NA	NA	Do not rely on the content of this register
5	ecg_lp_en	0	RW	0 … LP filter disabled 1 … LP filter enabled
4	ecg_hp_en	0	RW	0 … Power down of the high pass filter 1 … Enable high pass filter
3	ecg_gain_en	0	RW	0 … Power down of the Gain stage 1 … Enable Gain stage
2	ecg_lp_byp	0	RW	0 … LP stage is used 1 … LP stage is bypassed
1	ecg_hp_byp	0	RW	0 … HP filter is used 1 … HP filter is bypassed
0	ecg_gain_byp	0	RW	0 … Gain stage is used 1 … Gain stage is bypassed

Figure 107:
ECG_CFGB Register

0x5d: ECG_CFGB						
Field	**Name**	**Rst**	**Type**	**Description**		
6:5	ecg_lp_freq	0	RW	ECG low pass filter.		
				Setting	**Frequency**	
				0	40Hz	
				1	80Hz	
				2	160Hz	
				3	320Hz	
4:3	ecg_hp_freq	0	RW	High pass filter cutoff frequency		
				Setting	**Filter Frequency**	**Cutoff Frequency**
				0	122Hz	0.33Hz
				1	488Hz	1.32Hz
				2	1953Hz	5.28Hz
				3	3906Hz	10.56Hz
2:0	ecg_gain_g	0	RW	Gain		
				Setting	**Gain**	
				0	1	
				1	2	
				2	4	
				3	8	
				4	16	
				5	32	
				6	64	
				7	128	

Figure 108:
ECG_CFGC Register

0x5e: ECG_CFGC				
Field	**Name**	**Rst**	**Type**	**Description**
1	ecg_low_ leakage_en	0	RW	Enable ECG leakage compensation
0	ecg_ref_en	0	RW	ECG Reference Feedback Amplifier Enable

Figure 109:
ECG_CFGD Register

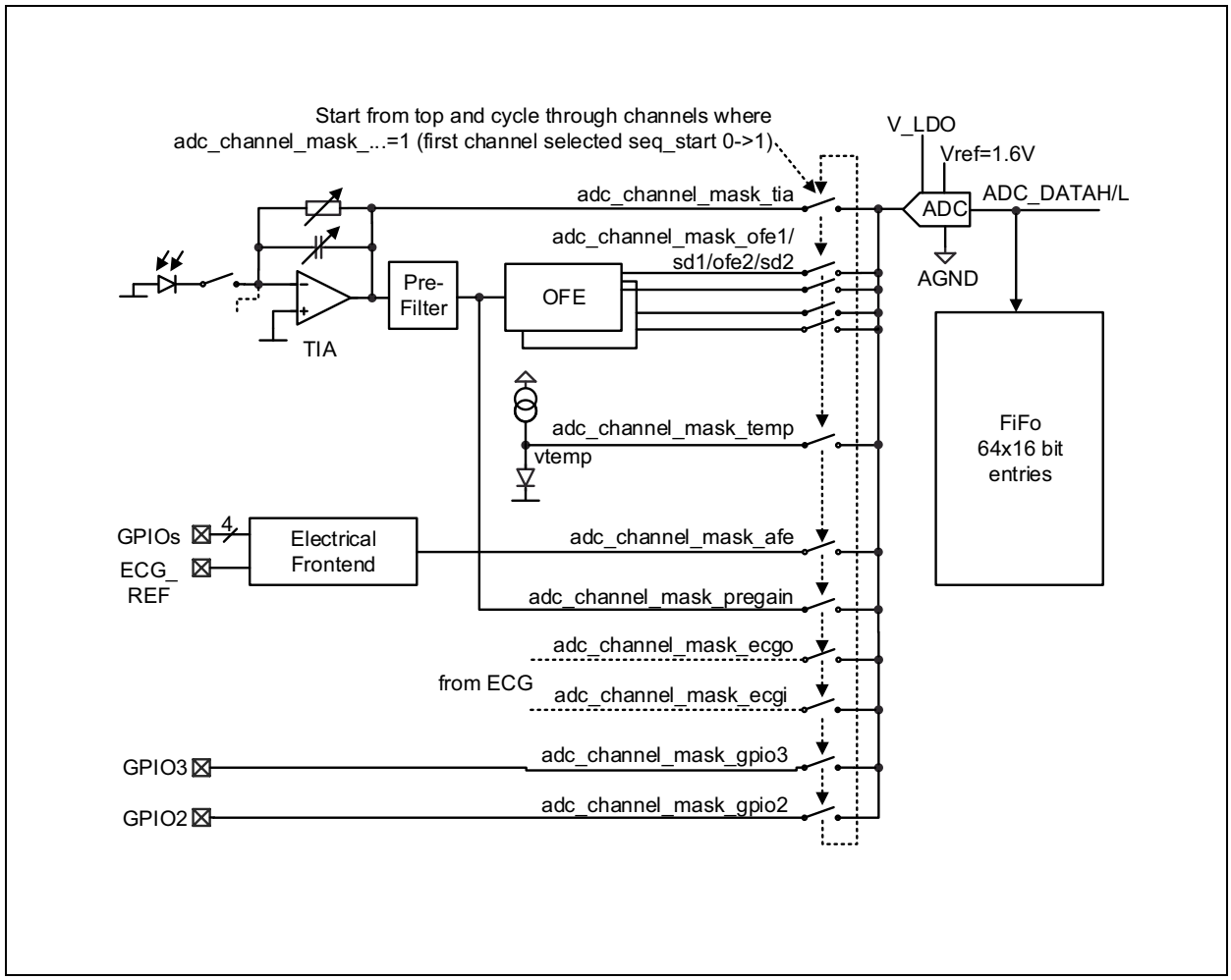
0x5f: ECG_CFGD					
Field	**Name**	**Rst**	**Type**	**Description**	
4	ecg_leadsdet_ sync_adc	0	RW	ECG Leads Detection Automatic Update. If this is asserted, then ecg_leadsdet_pol is inverted automatically at the start of a sequence (at count=2) if in this sequence the ADC will convert the ECGi channel.	
3	ecg_leadsdet_pol	0	RW_SM	ECG Leads Detection Polarity. Can be written to manually if ecg_leadsdet_sync_adc is clear, otherwise it is automatically toggled.	
2:1	ecg_leadsdet_curr	0	RW	ECG Leads Detection Current	
				Setting	**Current**
				0	20nA
				1	100nA
				2	500nA
				3	1µA
0	ecg_leadsdet_en	0	RW	ECG Leads Detection Enable	

ADC and FIFO

The ADC is a 14bit successive-approximation register (SAR) type. It supports 14bit with conversion time up to 50ksps.

The ADC is started by the sequencer and its timing or in manual mode (man_mode=1) by setting seq_start=1 (seq_start stays '1' as long as the conversion runs). The AS7024 can be configured to trigger an interrupt upon end of conversion.

Figure 110:
ADC Internal Circuit and Multiplexer



For best accuracy, the ADC can be optionally calibrated.

Note(s): If GPIO2 or GPIO3 is used as ADC input, there is no anti-aliasing filter in front of the ADC (needs to be added externally).

ADC Threshold

At the output of the ADC converter a digital threshold can be enabled. If the output of the ADC exceeds the threshold adc_threshold, it triggers an interrupt. This mechanism can be used to identify if an object is in proximity of the sensor and then to interrupt the host. In cases where no object is detected, the host can be sleeping therefore reducing power consumption of the system.

For detailed description of the threshold calculation see the register ADC_THRESHOLD and ADC_THRESHOLD_CFG description.

ADC Registers

Figure 111:
ADC_THRESHOLD Register

0x68: ADC_THRESHOLD				
Field	**Name**	**Rst**	**Type**	**Description**
7:0	adc_threshold	0xff	RW	If the ADC returns a value above adc_threshold (not equal), then the adc_threshold interrupt can be triggered. Note that when comparing, only the upper 8 bits are compared, the lower 6 bits are ignored. A value of 0xff can therefore never trigger the interrupt.

Figure 112:
ADC_THRESHOLD_CFG Register

0x69: ADC_THRESHOLD_CFG				
Field	**Name**	**Rst**	**Type**	**Description**
1	adc_thresh_ differential	0	RW	If adc_thresh_tiaonly is asserted and any of seq_adc[23]tia is non-zero, meaning that there are two or three ADC TIA measurements in one sequencer period, then the second is subtracted from the first, and the *difference* is being compared to the adc_threshold.
0	adc_thresh_ tiaonly	0	RW	Normally, the adc_threshold works regardless of the adc channel. If this bit is set, then the threshold is only checked if the adc channel is TIA.

Figure 113:
ADC_CFGA Register

<table>
<tr><th colspan="5">0x88: ADC_CFGA</th></tr>
<tr><th>Field</th><th>Name</th><th>Rst</th><th>Type</th><th>Description</th></tr>
<tr><td rowspan="9">3:1</td><td rowspan="9">adc_multi_n</td><td rowspan="9">0</td><td rowspan="9">RW</td><td colspan="2">Defines number of samples that are taken in multimode (adc_multimode =1) (1)</td></tr>
<tr><td>Setting</td><td>Number of Samples per ADC Conversion Command</td></tr>
<tr><td>0</td><td>2</td></tr>
<tr><td>1</td><td>4</td></tr>
<tr><td>2</td><td>8</td></tr>
<tr><td>3</td><td>16</td></tr>
<tr><td>4</td><td>32</td></tr>
<tr><td>5</td><td>48</td></tr>
<tr><td>6</td><td>64</td></tr>
<tr><td>7</td><td>96</td></tr>
<tr><td>0</td><td>adc_multimode</td><td>0</td><td>RW</td><td colspan="2">0 … If ADC is started one sample is measured
1 … If ADC is started multiple samples are stored in sequence in the FIFO. The number of samples is defined with "adc_multi_n".</td></tr>
</table>

Note(s):

1. If the ADC is triggered with the sequencer, the very first ADC conversion after seq_en=1 stores the number of samples according to above table. All subsequent samples use one sample less (e.g. 7 instead of 8).

The ADC_CFGA,B,C register is used to configure the ADC operation.

Figure 114:
ADC_CFGB Register

<table>
<tr><th colspan="8">0x89: ADC_CFGB</th></tr>
<tr><th>Field</th><th>Name</th><th>Rst</th><th>Type</th><th colspan="4">Description</th></tr>
<tr><td rowspan="10">5:3</td><td rowspan="10">adc_clock</td><td rowspan="10">0</td><td rowspan="10">RW</td><td colspan="4">ADC clock divider: The ADC clock is freely configurable.</td></tr>
<tr><td>Setting</td><td>Periods</td><td>µs</td><td>kHz</td></tr>
<tr><td>0</td><td>2</td><td>1</td><td>1000</td></tr>
<tr><td>1</td><td>4</td><td>2</td><td>500</td></tr>
<tr><td>2</td><td>6</td><td>3</td><td>333</td></tr>
<tr><td>3</td><td>8</td><td>4</td><td>250</td></tr>
<tr><td>4</td><td>10</td><td>5</td><td>200</td></tr>
<tr><td>5</td><td>12</td><td>6</td><td>167</td></tr>
<tr><td>6</td><td>14</td><td>7</td><td>143</td></tr>
<tr><td>7</td><td>16</td><td>8</td><td>125</td></tr>
<tr><td>2</td><td>adc_calibration</td><td>0</td><td>RW</td><td colspan="4">To activate the optional self calibration, this bit must be asserted, and an ADC “conversion” has to be started in manual mode (man_mode=1) by asserting seq_start.</td></tr>
<tr><td>1</td><td>ulp</td><td>0</td><td>RW</td><td colspan="4">Ultra low power bit for the sequencer. If this bit is set and sd_subs>0, it disables the LED pulses and powers off the TIA in all sequences but the one where the TIA is sampled.</td></tr>
<tr><td>0</td><td>adc_en</td><td>0</td><td>RW</td><td colspan="4">0 … Reset ADC
1 … Enable ADC
Warning: In reset state the ADC clears its calibration data. Re-calibration is necessary next time it is enabled again.</td></tr>
</table>

Figure 115:
ADC_CFGC Register

<table>
<tr><th colspan="8">0x8a: ADC_CFGC</th></tr>
<tr><th>Field</th><th>Name</th><th>Rst</th><th>Type</th><th colspan="4">Description</th></tr>
<tr><td>4</td><td>adc_selfpd</td><td>1</td><td>RW</td><td colspan="4">1 … Power down the ADC when not converting; use this to conserve power, but set adc_settling_time to minimum 64us to permit settling of the ADC reference buffer.
0 … Always enable ADC</td></tr>
<tr><td>3</td><td>adc_discharge</td><td>1</td><td>RW</td><td colspan="4">0: Suppress ADC capacitor discharging – use with caution
1: Discharge ADC capacitor before tracking
If asserted, the capacitor is discharged before the tracking phase. If zero, the discharge phase is suppressed and the tracking phase is started one cycle earlier.</td></tr>
<tr><td rowspan="10">2:0</td><td rowspan="10">adc_settling_
time</td><td rowspan="10">0</td><td rowspan="10">RW</td><td colspan="4">ADC settling time: Use with synchronous demodulator. It defines the number of ADC clock cycles the sampling window is kept open additionally.
If the gain stage in the optical frontend is used (gain_byp=0), set this to minimum 8µs. If adc_selfpd=1, set this to minimum 64µs.</td></tr>
<tr><td>Setting</td><td>Periods</td><td>µs
(@500kHz)</td><td>µs
(@250kHz)</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>1</td><td>4</td><td>8</td><td>16</td></tr>
<tr><td>2</td><td>8</td><td>16</td><td>32</td></tr>
<tr><td>3</td><td>16</td><td>32</td><td>64</td></tr>
<tr><td>4</td><td>32</td><td>64</td><td>128</td></tr>
<tr><td>5</td><td>64</td><td>128</td><td>256</td></tr>
<tr><td>6</td><td>128</td><td>256</td><td>512</td></tr>
<tr><td>7</td><td>256</td><td>512</td><td>1ms</td></tr>
</table>

Figure 116:
ADC_CHANNEL_MASK_L Register

0x8b: ADC_CHANNEL_MASK_L				
Field	**Name**	**Rst**	**Type**	**Description**
7	adc_channel_ mask_pregain	0	RW	Pregain channel selection
6	adc_channel_ mask_afe	0	RW	Electrical front end
5	adc_channel_ mask_temp	0	RW	Temperature measurement
4	adc_channel_ mask_sd2	0	RW	Synchronous modulator 2 output just before the gain stage
3	adc_channel_ mask_ofe2	0	RW	Synchronous modulator 2 output after the gain stage
2	adc_channel_ mask_sd1	0	RW	Synchronous modulator 1 output just before the gain stage
1	adc_channel_ mask_ofe1	0	RW	Synchronous modulator 1 output after the gain stage
0	adc_channel_ mask_tia	0	RW	Transimpedance amplifier output

The adc channel is chosen automatically from the bits within the adc_channel_mask_* set. It starts from right and finishes left (LSB->MSB) and wraps back from the most significant asserted bit to the least significant of the asserted bits. After every ADC conversion it switches to the next enabled channel, (except around the adc2tia/adc3tia cases). See register description FIFOH and FIFOL for encoding of the first channel in the data stream.

This applies to both, manual mode and sequencer mode. In sequencer mode, it starts with the smallest channel when the sequencer is being started. In manual mode, the adc_sel is reset with every write to either ADC_CHANNEL_MASK_L or ADC_CHANNEL_MASK_H

Figure 117:
ADC_CHANNEL_MASK_H Register

0x8c: ADC_CHANNEL_MASK_H				
Field	**Name**	**Rst**	**Type**	**Description**
3	adc_channel_ mask_gpio2	0	RW	GPIO2 input – set gpio2_a=1 and Write 0x47 to register 0xC6 Write 0x0C to register 0xC2 Write 0x0C to register 0xC3
2	adc_channel_ mask_gpio3	0	RW	GPIO3 input – set gpio3_a=1 and Write 0x47 to register 0xC6 Write 0x0C to register 0xC2 Write 0x0C to register 0xC3
1	adc_channel_ mask_ecgi	0	RW	ECG amplifier input – use for leads off detection
0	adc_channel_ mask_ecgo	0	RW	ECG amplifier output – amplified ECG signal

Figure 118:
ADC_DATA_L Register

0x8e: ADC_DATA_L				
Field	**Name**	**Rst**	**Type**	**Description**
7:0	adc_data[7:0]	0	RO	Current ADC output: low byte.

The ADC_DATA register shows the current raw output of the ADC

Figure 119:
ADC_DATA_H Register

0x8f: ADC_DATA_H				
Field	**Name**	**Rst**	**Type**	**Description**
5:0	adc_data[13:8]	0	RO	Current ADC output: high byte warning: there is no latch mechanism implemented to guarantee consistency if the ADC is possibly running when reading this register, then the data can be corrupted - use the FIFO to guarantee data consistency.

FIFO Registers

Figure 120:
FIFO_CFG Register

0x78: FIFO_CFG				
Field	**Name**	**Rst**	**Type**	**Description**
5:0	fifo_threshold	19	RW	FIFO threshold. The fifo_threshold interrupt is flagged if there are more than this many entries in the FIFO. 0 … Interrupt with 1 (16bit) entry in FIFO 63 … Interrupt when FIFO is full but one; note that the FIFO is 64 entries deep

Figure 121:
FIFO_CNTRL Register

0x79: FIFO_CNTRL				
Field	**Name**	**Rst**	**Type**	**Description**
0	fifo_clear	0	PUSH1	Write a 1 here to clear the FIFO can be useful when switching from one sequencer mode to another to make sure that there are no old FIFO entries left

Figure 122:
FIFOSTATUS Register

0xa3: FIFOSTATUS				
Field	**Name**	**Rst**	**Type**	**Description**
7	fifooverflow	0	RO	FIFO overflow indicator
6:0	fifolevel	0	RO	FIFO fill level (0 …64)

Figure 123:
FIFOL Register

0xfe: FIFOL				
Field	Name	Rst	Type	Description
7:0	fifol	0	PUSHPOP	Low byte of FIFO

FIFOL can be read out with single reads (2 consecutive I²C addresses have to be read to get one FIFO entry) or with block-read (up to 2 x fifo_depth values can be read in a single block-read)
Upon reading of FIFOH, it automatically advances the internal read pointer and decreases FIFO level.
If reading beyond end of FIFO, data will return 00h. There is no underrun flag, this is not an error condition.

Use ams SDK functions to read from the FIFO register to keep the reading in synchronization with the ADC channel selection. If synchronization is no concern use [fifoh[7:0] : fifol[7:2]] as ADC result as the ADC data is multiplied by x4 before it is pushed in to the FIFO. FIFOl[0] is used as an ADC first channel indication. The first channel indication bit toggles upon every new entry unless the first ADC channel is transmitted. Then toggling can be stopped for up to 5 FIFO entries and the very first stopping indicates the first ADC channel. To allow encoding of any number of ADC channels, the first ADC channel encoding is dropped from time to time.

Figure 124:
FIFOH Register

0xff: FIFOH				
Field	Name	Rst	Type	Description
7:0	fifoh	0	PUSHPOP	High byte of FIFO

See Interrupts for the actual FIFO interrupt.

Digital Interface

Power Management

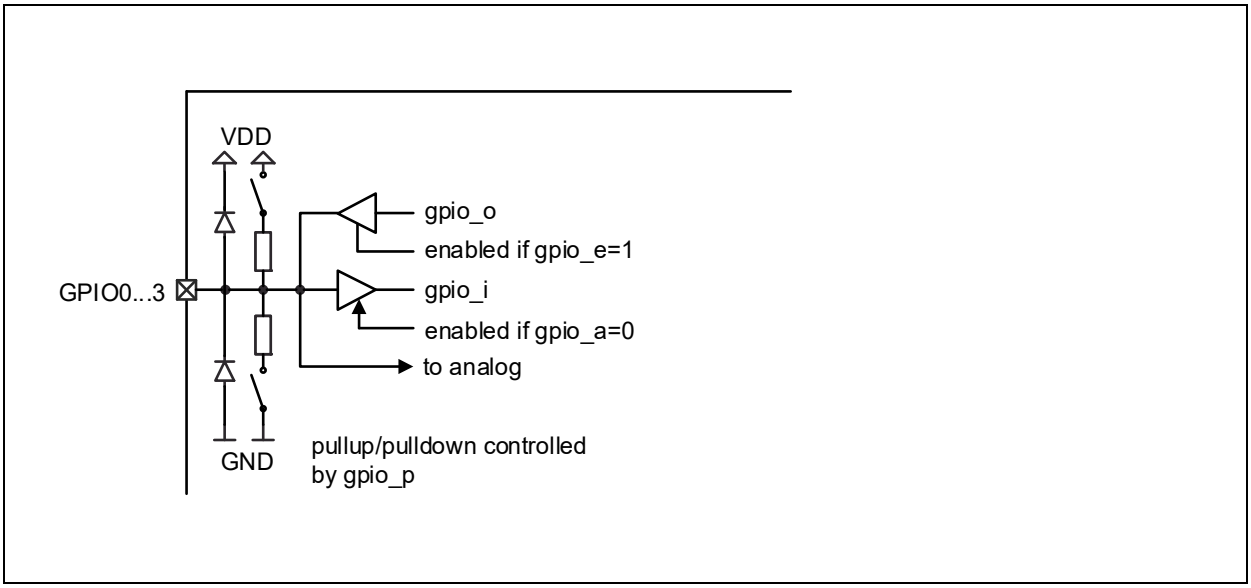
After setting the pin ENABLE=1 the AS7024 registers can be accessed by the I²C interface. Before enabling any additional function (current source, TIA, ADC…) set the bit ldo_en=1 to set the internal LDO to normal mode.

For operating the ADC or the sequencer enable the oscillator by setting osc_en=1.

GPIO Pins

Each of the GPIO pins can be digitally controlled and is capable of adding a pullup and/or pulldown:

Figure 125:
GPIO Internal Circuit



Interrupts

An interrupt output pin INT can be used to interrupt the host. Following interrupt sources are possible

irq_adc: End of ADC conversion

irq_sequencer: End of sequencer sequence reached.

irq_ltf: A light-to-frequency conversion is finished.

irq_adc_threshold: ADC threshold triggered – see ADC Threshold.

irq_fifothreshold: FIFO almost full (as defined in register fifo_threshold)

irq_fifooverflow: FIFO overflow (error condition, data is lost)

irq_clipdetect: TIA output and/or SD output exceeded threshold– see details in CLIPSTATUS

irq_led_supply_low: led supply low comparator triggered – see details in LEDSTATUS

Depending on the setting in register INTENAB each of the above interrupt source can assert INT output pin (active low).

I²C

The AS7024 includes an I²C slave using an I²C address of 0x30 (7-bit format; R/W bit has to be added) respectively 60h (8-bit format for writing) and 61h (8-bit format for reading). It expects external pullup resistors.

I²C Serial Control Interface

I²C Feature List

Fast mode (400kHz) and standard mode (100kHz) support

7+1-bit addressing mode

Write formats: Single-Byte-Write, Page-Write

Read formats: Current-Address-Read, Random-Read, Sequential-Read

SDA input delay and SCL spike filtering by integrated RC-components

I²C Protocol

Figure 126:
I²C Symbol Definition

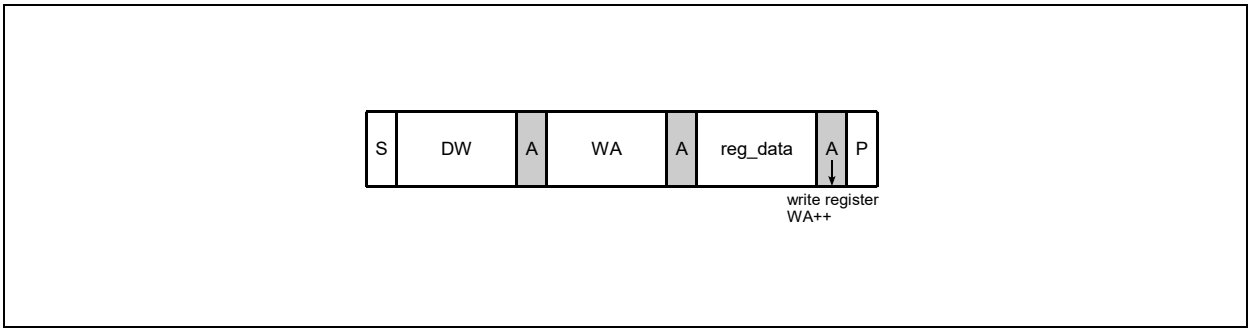
Symbol	Definition	RW	Note
S	Start condition after stop	R	1 bit
Sr	Repeated start	R	1 bit
DW	Device address for write	R	0110 0000b (60h)
DR	Device address for read	R	0110 0001b (61h)
WA	Word address	R	8 bit
A	Acknowledge	W	1 bit
N	No Acknowledge	R	1 bit
reg_data	Register data/write	R	8 bit
data (n)	Register data/read	W	8 bit
P	Stop condition	R	1 bit
WA++	Increment word address internally	R	During acknowledge

I²C Symbol Definition: Shows the symbols used in the following mode descriptions.

I²C Write Access

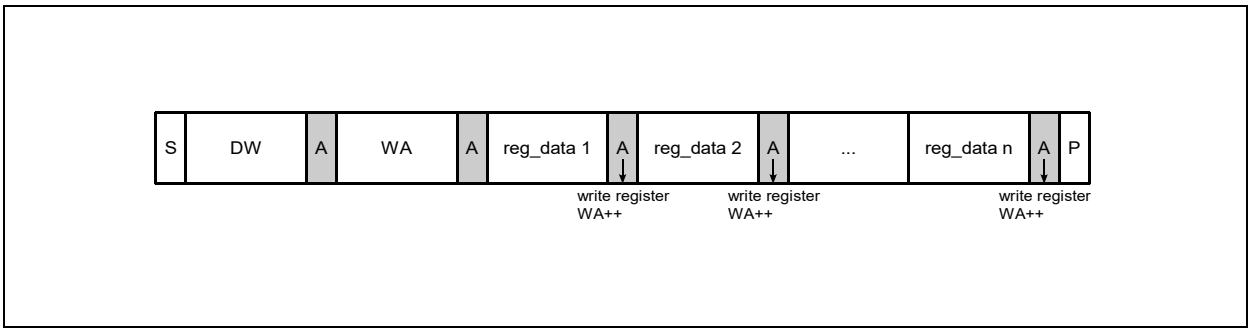
Byte Write and Page Write formats are used to write data to the slave.

Figure 127:
I²C Byte Write



I²C Byte Write: Shows the format of an I²C byte write access.

Figure 128:
I²C Page Write



I²C Page Write: Shows the format of an I²C page write access.

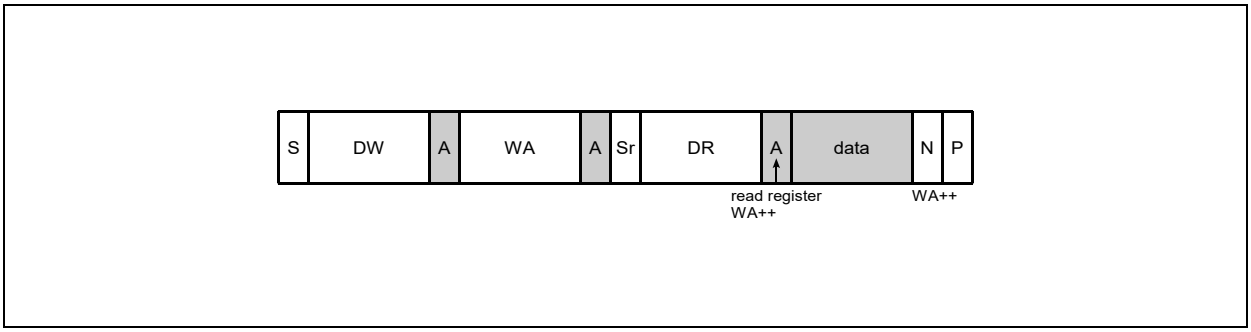
The transmission begins with the START condition, which is generated by the master when the bus is in IDLE state (the bus is free). The device-write address is followed by the word address. After the word address any number of data bytes can be sent to the slave. The word address is incremented internally, in order to write subsequent data bytes on subsequent address locations.

For reading data from the slave device, the master has to change the transfer direction. This can be done either with a repeated START condition followed by the device-read address, or simply with a new transmission START followed by the device-read address, when the bus is in IDLE state. The device-read address is always followed by the 1st register byte transmitted from the slave. In Read Mode any number of subsequent register bytes can be read from the slave. The word address is incremented internally.

I²C Read Access

Random, Sequential and Current Address Read are used to read data from the slave.

Figure 129:
I²C Random Read



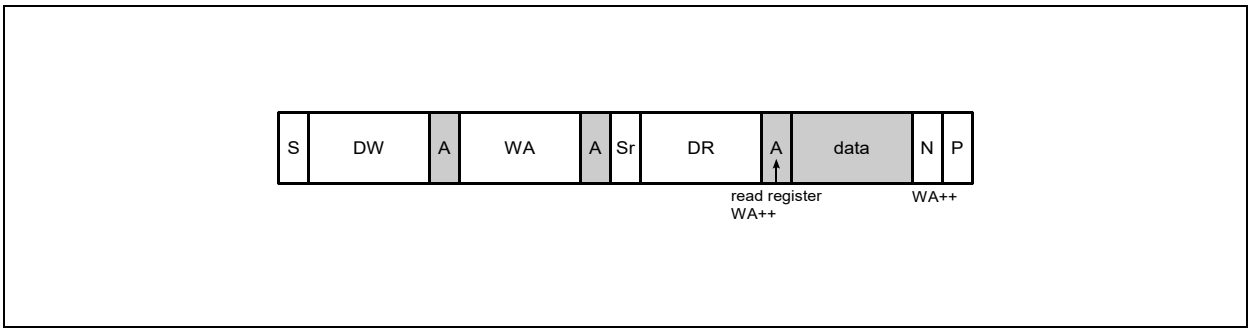
I²C Random Read: Shows the format of an I²C random read access.

Random Read and Sequential Read are combined formats. The repeated START condition is used to change the direction after the data transfer from the master.

The word address transfer is initiated with a START condition issued by the master while the bus is idle. The START condition is followed by the device-write address and the word address.

In order to change the data direction a repeated START condition is issued on the 1st SCL pulse after the acknowledge bit of the word address transfer. After the reception of the device-read address, the slave becomes the transmitter. In this state the slave transmits register data located by the previous received word address vector. The master responds to the data byte with a not-acknowledge, and issues a STOP condition on the bus.

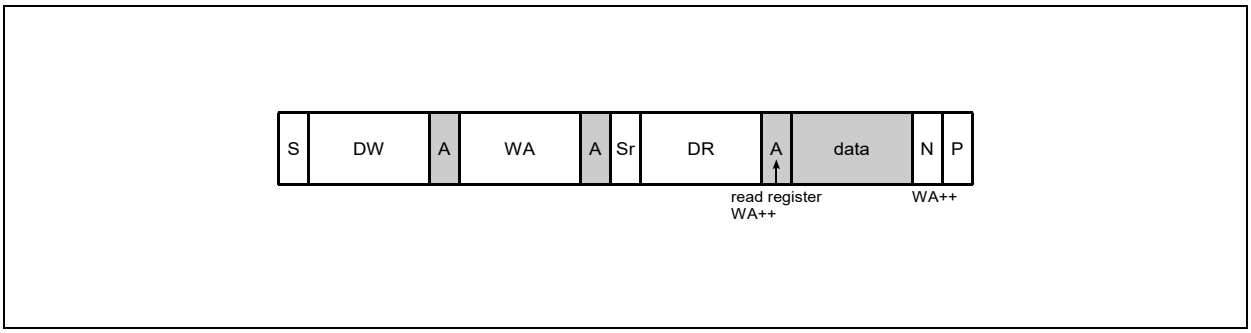
Figure 130:
I²C Sequential Read



I²C Sequential Read: Shows the format of an I²C sequential read access.

Sequential Read is the extended form of Random Read, as more than one register-data bytes are transferred subsequently. In difference to the Random Read, for a sequential read the transferred register-data bytes are responded by an acknowledge from the master. The number of data bytes transferred in one sequence is unlimited (consider the behavior of the word-address counter). To terminate the transmission the master has to send a not-acknowledge following the last data byte and generate the STOP condition subsequently.

Figure 131:
I²C Current Address Read



I²C Current Address Read: Shows the format of an I²C current address read access.

To keep the access time as small as possible, this format allows a read access without the word address transfer in advance to the data transfer. The bus is idle and the master issues a START condition followed by the Device-Read address. Analogous to Random Read, a single byte transfer is terminated with a not-acknowledge after the 1st register byte. Analogous to Sequential Read an unlimited number of data bytes can be transferred, where the data bytes has to be responded with an acknowledge from the master. For termination of the transmission the master sends a not-acknowledge following the last data byte and a subsequent STOP condition.

Figure 132:
CONTROL Register

0x00: CONTROL				
Field	**Name**	**Rst**	**Type**	**Description**
1	osc_en	0	RW	Enable the oscillator. The oscillator must be enabled for any analog block (ADC, sequencer, optical frontend, sequencer); not mandatory for current sinks or ECG amplfier
0	ldo_en	0	RW	If the EN input is not asserted, the chip is in reset If asserted, I^2C transactions are possible. Upon assertion of ldo_en, the reference and the LDO are enabled The LDO must be enabled for anything but plain I^2C register read/write

Figure 133:
GPIO_A Register

0x08: GPIO_A				
Field	**Name**	**Rst**	**Type**	**Description**
3	gpio3_a	0	RW	1=Put GPIO3 in analog mode; set this bit when used for an analog function e.g. the electrical frontend. If set execute following I^2C commands (otherwise an internal pulldown will be enabled) in this sequence: Write 0x47 to register 0xC6 Write 0x0C to register 0xC2 Write 0x0C to register 0xC3
2	gpio2_a	0	RW	1=Put GPIO2 in analog mode If set execute following I^2C commands (otherwise an internal pulldown will be enabled) in this sequence: Write 0x47 to register 0xC6 Write 0x0C to register 0xC2 Write 0x0C to register 0xC3
1	gpio1_a	0	RW	1=Put GPIO1 in analog mode(1)
0	gpio0_a	0	RW	1=Put GPIO0 in analog mode(1)

Note(s):

1. No further I^2C commands are required (different to GPIO2/3).

Figure 134:
GPIO_E Register

0x09: GPIO_E				
Field	Name	Rst	Type	Description
3	gpio3_e	0	RW	GPIO3 output enabled if set
2	gpio2_e	0	RW	GPIO2 output enabled if set
1	gpio1_e	0	RW	GPIO1 output enabled if set
0	gpio0_e	0	RW	GPIO0 output enabled if set

Figure 135:
GPIO_O Register

0x0a: GPIO_O				
Field	Name	Rst	Type	Description
3	gpio3_o	0	RW	If gpio3_e=1, gpio3_o defines the output state of GPIO3
2	gpio2_o	0	RW	If gpio2_e=1, gpio2_o defines the output state of GPIO2
1	gpio1_o	0	RW	If gpio1_e=1, gpio1_o defines the output state of GPIO1
0	gpio0_o	0	RW	If gpio0_e=1, gpio0_o defines the output state of GPIO0

Figure 136:
GPIO_I Register

0x0b: GPIO_I				
Field	Name	Rst	Type	Description
3	gpio3_i	0	RO	The digital value sensed on GPIO3
2	gpio2_i	0	RO	The digital value sensed on GPIO2
1	gpio1_i	0	RO	The digital value sensed on GPIO1
0	gpio0_i	0	RO	The digital value sensed on GPIO0

Figure 137:
GPIO_P Register

0x0c: GPIO_P				
Field	Name	Rst	Type	Description
7	gpio3_pd	0	RW	GPIO3 pulldown configuration 0: No pulldown on GPIO3 1: Pulldown to GND on GPIO3
6	gpio3_pu	0	RW	GPIO3 pullup configuration 0: No pullup on GPIO3 1: Pullup to VDD on GPIO3
5	gpio2_pd	0	RW	GPIO2 pulldown configuration
4	gpio2_pu	0	RW	GPIO2 pullup configuration
3	gpio1_pd	0	RW	GPIO1 pulldown configuration
2	gpio1_pu	0	RW	GPIO1 pullup configuration
1	gpio0_pd	0	RW	GPIO0 pulldown configuration
0	gpio0_pu	0	RW	GPIO0 pullup configuration

Figure 138:
GPIO_SR Register

0x0d: GPIO_SR				
Field	Name	Rst	Type	Description
3	gpio3_sr	0	RW	GPIO3 slew rate configuration 0: Default slew rate 1: Increased slew rate
2	gpio2_sr	0	RW	GPIO2 slew rate configuration
1	gpio1_sr	0	RW	GPIO1 slew rate configuration
0	gpio0_sr	0	RW	GPIO0 slew rate configuration

Figure 139:
SUBID Register

0x91: SUBID				
Field	**Name**	**Rst**	**Type**	**Description**
7:3	subid	NA	RO	Defines product version. Do not rely on bits defined as 'X'. 1XXXXb
2:0	Revision	NA	RO	Reserved. Do no use and do not rely that the content stays the same for each device.

Internal information: At least one bit is set on subid.

Figure 140:
ID Register

0x92: ID					
Field	**Name**	**Rst**	**Type**	**Description**	
7:2	id	0	RO	Part Number Identification.	
				Value	**Meaning**
				110011	AS7024
1:0	id_reserved	0	RO	Reserved. Do no use and do not rely that the content stays the same for each device.	

Figure 141:
STATUS Register

0xa0: STATUS				
Field	**Name**	**Rst**	**Type**	**Description**
7	irq_led_supply_low	0	R_PUSH1	Check LEDSTATUS
6	irq_clipdetect	0	R_PUSH1	Check CLIPSTATUS
5	irq_fifooverflow	0	R_PUSH1	FIFO overflow (error condition, new data is lost)
4	irq_fifothreshold	0	R_PUSH1	FIFO is almost full (as defined in fifo_threshold, usually 3/4)
3	irq_adc_threshold	0	R_PUSH1	The ADC value was above the programmed adc_threshold register setting
2	irq_ltf	0	R_PUSH1	LTF measurement is done. check LTFSTATUS (or ignore it)
1	irq_sequencer	0	R_PUSH1	All configured sequencer iterations have finished
0	irq_adc	0	R_PUSH1	ADC has finished

The STATUS register shows the current state of the interface. Some bits in here can trigger an interrupt.

An asserted bit can be cleared by writing a '1' to it - in case of irq_led_supply_low and irq_clipdetect, this also clears the underlying condition in the CLIPSTATUS and LEDSTATUS registers.

The FIFO threshold interrupt cannot be cleared directly, but only by lowering the FIFO level. The FIFO overflow interrupt is sticky and must be cleared explicitly.

Figure 142:
CLIPSTATUS Register

0xa1: CLIPSTATUS				
Field	**Name**	**Rst**	**Type**	**Description**
3	pd_clipdetect_l	0	RO	If this bit is asserted, photo diode amplifer has been below the lower threshold
2	pd_clipdetect_h	0	RO	If this bit is asserted, photo diode amplifer has been above the upper threshold
1	sd_clipdetect_l	0	RO	If this bit is asserted, photo diode amplifer has been below the lower threshold
0	sd_clipdetect_h	0	RO	If this bit is asserted, photo diode amplifer has been above the upper threshold

An asserted bit can be cleared by writing a '1' to the irq_clipdetect.

Figure 143:
LEDSTATUS Register

0xa2: LEDSTATUS				
Field	**Name**	**Rst**	**Type**	**Description**
3	led4_supply_low	0	RO	If this bit is asserted, LED4 voltage has been too low.
2	led3_supply_low	0	RO	If this bit is asserted, LED3 voltage has been too low.
1	led2_supply_low	0	RO	If this bit is asserted, LED2 voltage has been too low.
0	led1_supply_low	0	RO	If this bit is asserted, LED1 voltage has been too low.

An asserted bit can be cleared by writing a '1' to the irq_led_supply_low bit.

Figure 144:
INTENAB Register

0xa8: INTENAB				
Field	**Name**	**Rst**	**Type**	**Description**
7	irq_led_supply_low_enab	0	RW	1 … Enable led supply low interrupt
6	irq_clipdetect_enab	0	RW	1 … Enable clipdetect interrupt
5	irq_fifooverflow_enab	0	RW	1 … Enable fifooverflow interrupt
4	irq_fifothreshold_enab	0	RW	1 … Enable fifothreshold interrupt
3	irq_adc_threshold_enab	0	RW	1 … Enable irq_adc_threshold as an interrupt source
2	irq_ltf_enab	0	RW	1 … Enable LTF as an interrupt source
1	irq_sequencer_enab	0	RW	1 … Enable irq_sequencer as an interrupt source
0	irq_adc_enab	0	RW	1 … Enable irq_adc as an interrupt source

Each of the STATUS register bits can cause an interrupt (register INTR) if the respective bit is asserted in the INTENAB register.

Figure 145:
INTR Register

0xa9: INTR				
Field	**Name**	**Rst**	**Type**	**Description**
7	irq_led_supply_low_intr	0	RO	
6	irq_clipdetect_intr	0	RO	
5	irq_fifooverflow_intr	0	RO	
4	irq_fifothreshold_intr	0	RO	
3	irq_adc_threshold_intr	0	RO	
2	irq_ltf_intr	0	RO	
1	irq_sequencer_intr	0	RO	
0	irq_adc_intr	0	RO	

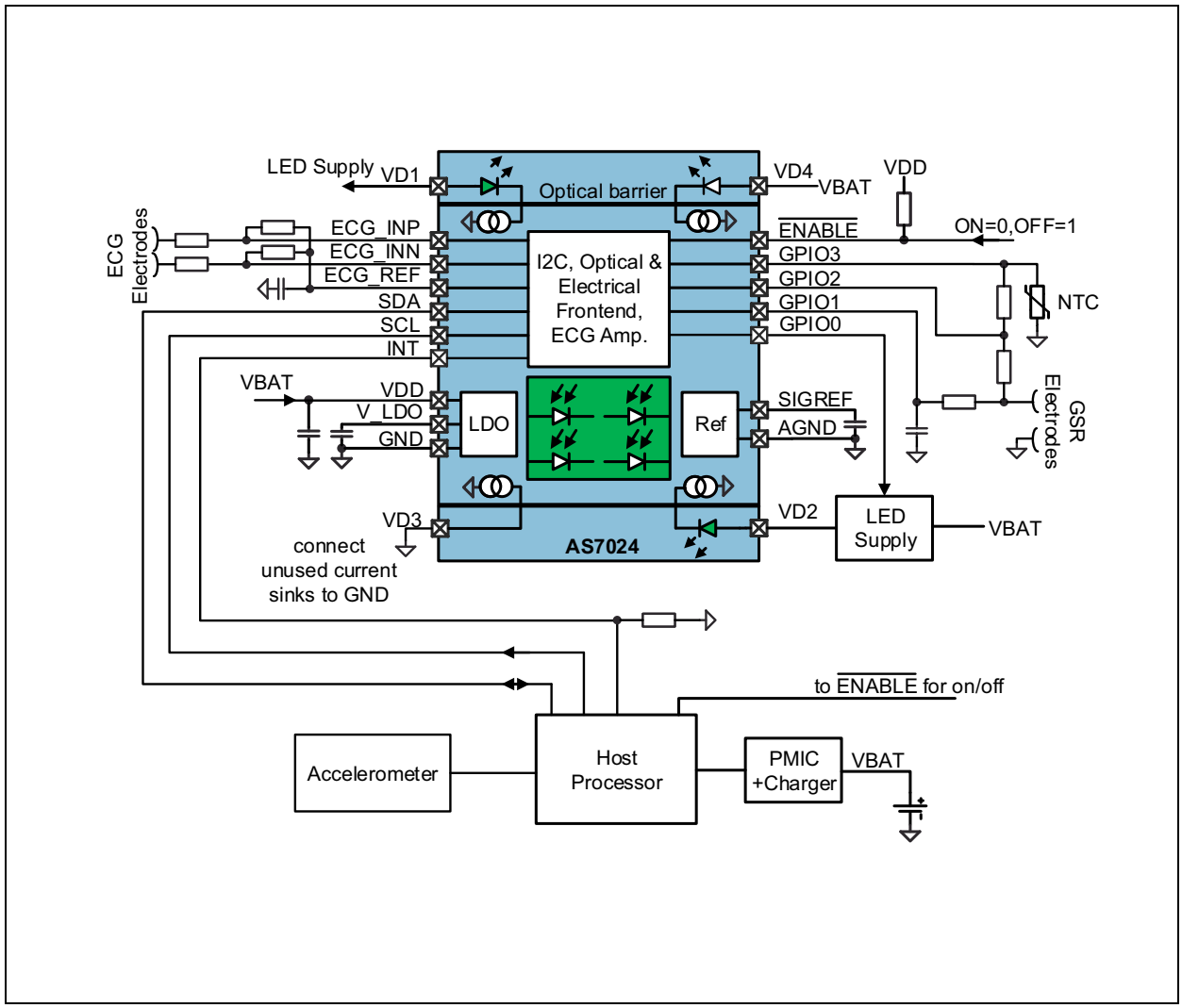
The INTR registers shows the bit or bits that are responsible for an asserted interrupt. Effectively, these bits are OR-ed together to drive the interrupt pin INT low (open drain output).

Application Information

The following figure shows the complete integration of the AS7024 in a mobile optical measurement system for HRM, SpO2, GSR (galvanic skin resistivity) and skin temperature using an NTC.

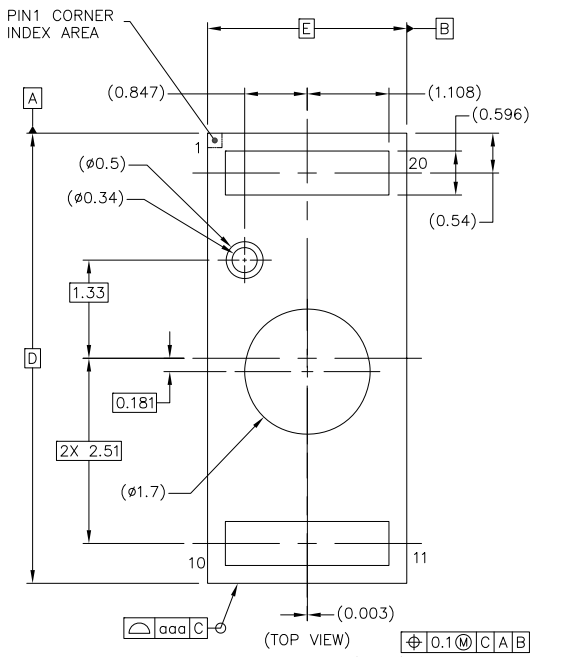
The device can be powered directly by a LiIon battery as it has its own power management. Nevertheless the I^2C interface can be powered by 1.8V circuitry.

Figure 146:
Optical HRM Measurement System for Wrist-Based Application

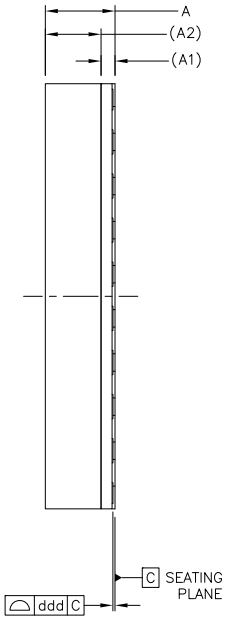


Package Drawings & Markings

Figure 147:
Package Drawing



	SYMBOL	COMMON DIMENSIONS		
		MIN.	NOR.	MAX.
TOTAL THICKNESS	A	0.9	1	1.1
SUBSTRATE THICKNESS	A1		0.2	REF
LID THICKNESS	A2		0.8	REF
BODY SIZE	D		6.1	BSC
	E		2.7	BSC
LEAD WIDTH	W	0.2	0.25	0.3
LEAD LENGTH	L	0.55	0.6	0.65
LEAD PITCH	e		0.633	BSC
LEAD COUNT	n		20	
EDGE BALL CENTER TO CENTER	D1		5.697	BSC
	E1		1.95	BSC
BODY CENTER TO CONTACT BALL	SD		0.3165	BSC
	SE		---	BSC
BALL WIDTH	b	---	---	---
BALL DIAMETER			---	
BALL OPENING			---	
BALL PICTH	e1		---	
BALL COUNT	n1		---	
PRE-SOLDER		---	---	---
PACKAGE EDGE TOLERANCE	aaa		0.1	
MOLD FLATNESS	bbb		---	
COPLANARITY	ddd		0.08	
BALL OFFSET (PACKAGE)	eee		---	
BALL OFFSET (BALL)	fff		---	



Ordering & Contact Information

Figure 148:
Ordering Information

Ordering Code	Type	Marking	Delivery Form	Delivery Quantity
AS7024-AB	AS7024	NA	Tape and Reel	5000 pcs/reel

Buy our products or get free samples online at:
www.ams.com/Products

Technical Support is available at:
www.ams.com/Technical-Support

Provide feedback about this document at:
www.ams.com/Document-Feedback

For further information and requests, e-mail us at:
ams_sales@ams.com

For sales offices, distributors and representatives, please visit:
www.ams.com/Contact

Headquarters
ams AG
Tobelbader Strasse 30
8141 Premstaetten
Austria, Europe

Tel: +43 (0) 3136 500 0

Website: www.ams.com

RoHS Compliant & ams Green Statement

RoHS: The term RoHS compliant means that ams AG products fully comply with current RoHS directives. Our semiconductor products do not contain any chemicals for all 6 substance categories, including the requirement that lead not exceed 0.1% by weight in homogeneous materials. Where designed to be soldered at high temperatures, RoHS compliant products are suitable for use in specified lead-free processes.

ams Green (RoHS compliant and no Sb/Br): ams Green defines that in addition to RoHS compliance, our products are free of Bromine (Br) and Antimony (Sb) based flame retardants (Br or Sb do not exceed 0.1% by weight in homogeneous material).

Important Information: The information provided in this statement represents ams AG knowledge and belief as of the date that it is provided. ams AG bases its knowledge and belief on information provided by third parties, and makes no representation or warranty as to the accuracy of such information. Efforts are underway to better integrate information from third parties. ams AG has taken and continues to take reasonable steps to provide representative and accurate information but may not have conducted destructive testing or chemical analysis on incoming materials and chemicals. ams AG and ams AG suppliers consider certain information to be proprietary, and thus CAS numbers and other limited information may not be available for release.

Copyrights & Disclaimer

Devices sold by ams AG are covered by the warranty and patent indemnification provisions appearing in its General Terms of Trade. ams AG makes no warranty, express, statutory, implied, or by description regarding the information set forth herein. ams AG reserves the right to change specifications and prices at any time and without notice. Therefore, prior to designing this product into a system, it is necessary to check with ams AG for current information. This product is intended for use in commercial applications. Applications requiring extended temperature range, unusual environmental requirements, or high reliability applications, such as military, medical life-support or life-sustaining equipment are specifically not recommended without additional processing by ams AG for each application. This product is provided by ams AG “AS IS” and any express or implied warranties, including, but not limited to the implied warranties of merchantability and fitness for a particular purpose are disclaimed.

ams AG shall not be liable to recipient or any third party for any damages, including but not limited to personal injury, property damage, loss of profits, loss of use, interruption of business or indirect, special, incidental or consequential damages, of any kind, in connection with or arising out of the furnishing, performance or use of the technical data herein. No obligation or liability to recipient or any third party shall arise or flow out of ams AG rendering of technical or other services.

Document Status

Document Status	Product Status	Definition
Product Preview	Pre-Development	Information in this datasheet is based on product ideas in the planning phase of development. All specifications are design goals without any warranty and are subject to change without notice
Preliminary Datasheet	Pre-Production	Information in this datasheet is based on products in the design, validation or qualification phase of development. The performance and parameters shown in this document are preliminary without any warranty and are subject to change without notice
Datasheet	Production	Information in this datasheet is based on products in ramp-up to full production or full production which conform to specifications in accordance with the terms of ams AG standard warranty as given in the General Terms of Trade
Datasheet (discontinued)	Discontinued	Information in this datasheet is based on products which conform to specifications in accordance with the terms of ams AG standard warranty as given in the General Terms of Trade, but these products have been superseded and should not be used for new designs

Revision Information

Changes from 1-03 (2017-Sep-27) to current revision 1-04 (2018-Sep-11)	Page
Removed “Confidential” from footer	

Note(s):

1. Page and figure numbers for the previous version may differ from page and figure numbers in the current revision.
2. Correction of typographical errors is not explicitly mentioned.

Content Guide