APDS-9801

Digital Proximity and Analog Ambient Light Sensor

Data Sheet

Description

APDS-9801 is a module that integrates functions of an Analog ambient light sensor (ALS) and a proximity sensor (PS). The sensor has four chips in one small package: an ambient light sensor IC, proximity sensor signal conditioning circuitry and a proximity sensor that includes both an emitter and detector. The Analog ambient light sensor has current output, with spectral response close to the CIE standard Photopic observer. The proximity sensor IC has a LED driver and receiver circuit with digital count output, featuring excellent ambient light cancellation capability. With the built-in LED, the proximity sensor is able to sense the proximity of an object, such as finger or head to a portable device.

Ambient light sensors can be used to control the brightness of display backlighting by detecting the ambient light illuminance level. Proximity sensor technology make possible applications where detection or proximity of a user's head in relationship to display will turn off/on the keypad and LCD backlight. The combination of ambient light sensors and proximity sensors in one module make it ideal for portable devices, such as mobile phone, PDA and notebooks.

Ordering Information

Part Number	Packaging Type	Quantity
APDS-9801	Tape and Reel	2500 per reel

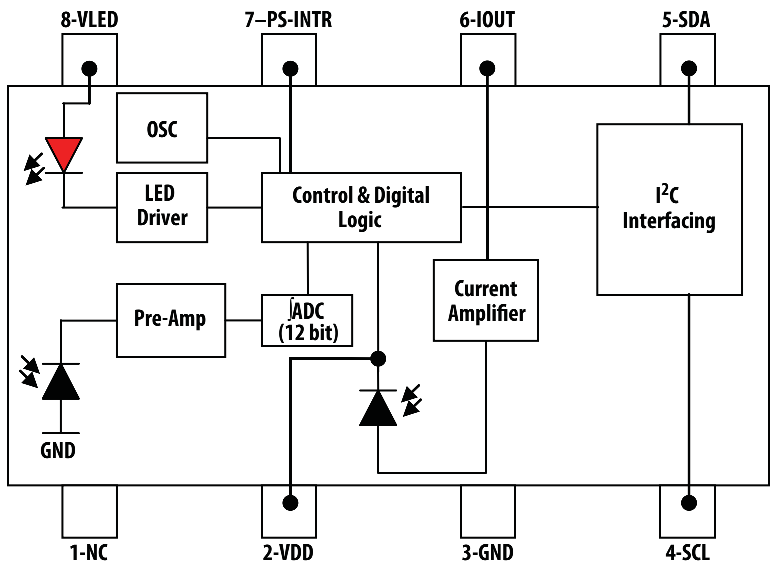
Features

- Integrated module with built-in IR LED, IR Detector, Digital Signal Conditioning ASIC and Analog ALS
- Package size: L 6.1 x W 3.9 x H 1.75 mm
- Sensor power supply voltage range: 1.7 V to 2.5 V
- I^2C Bus power supply voltage range: 1.7 V to 3.6 V
- Broad VLED range for PS: 2.5 V to 5V
- PS Shutdown Current 1µA Typical
- ALS approximate the Human Eye response
- Low sensitivity variation across various light sources
- ALS Output linearity up-to 5k Lux range
- Operational under sunlight (PS)
- Artificial light Immunity
- Low crosstalk between Emitter & Detector
- Programmable LED driving current and burst pulse control (PS)
- Interrupt logic with Programmable Threshold
- Lead-free & ROHS Compliant

Applications

- PDA and mobile phones
- Portable and Handheld devices
- Personal Computers/Notebooks
- Amusement/Games/Vending Machines
- Contactless Switches

Functional Block Diagram



I/O Pins Configuration Table:

Pin	SYMBOL	Description
1	NC	No Connect
2	VDD	Power Supply Pin
3	GND	Ground
4	SCL	I^2C Clock Input
5	SDA	I^2C Data Input/Output
6	IOUT	ALS Output Current
7	PS-INTR	Output pin for PS Level Interrupt
8	VLED	Power Supply pin for LED. Connect this pin to V-Battery or Power supply

Absolute Maximum Ratings

Parameter	Symbol	Min	Max	Unit	Conditions
Supply Voltage	V_{DD}	-0.5	4	V	T_A = 25°C
Voltage at I/O pins	V_{IO}	-0.5	5	V	T_A = 25°C
Reflow Soldering Temperature	T_S		260	°C	

Recommended Operating Conditions

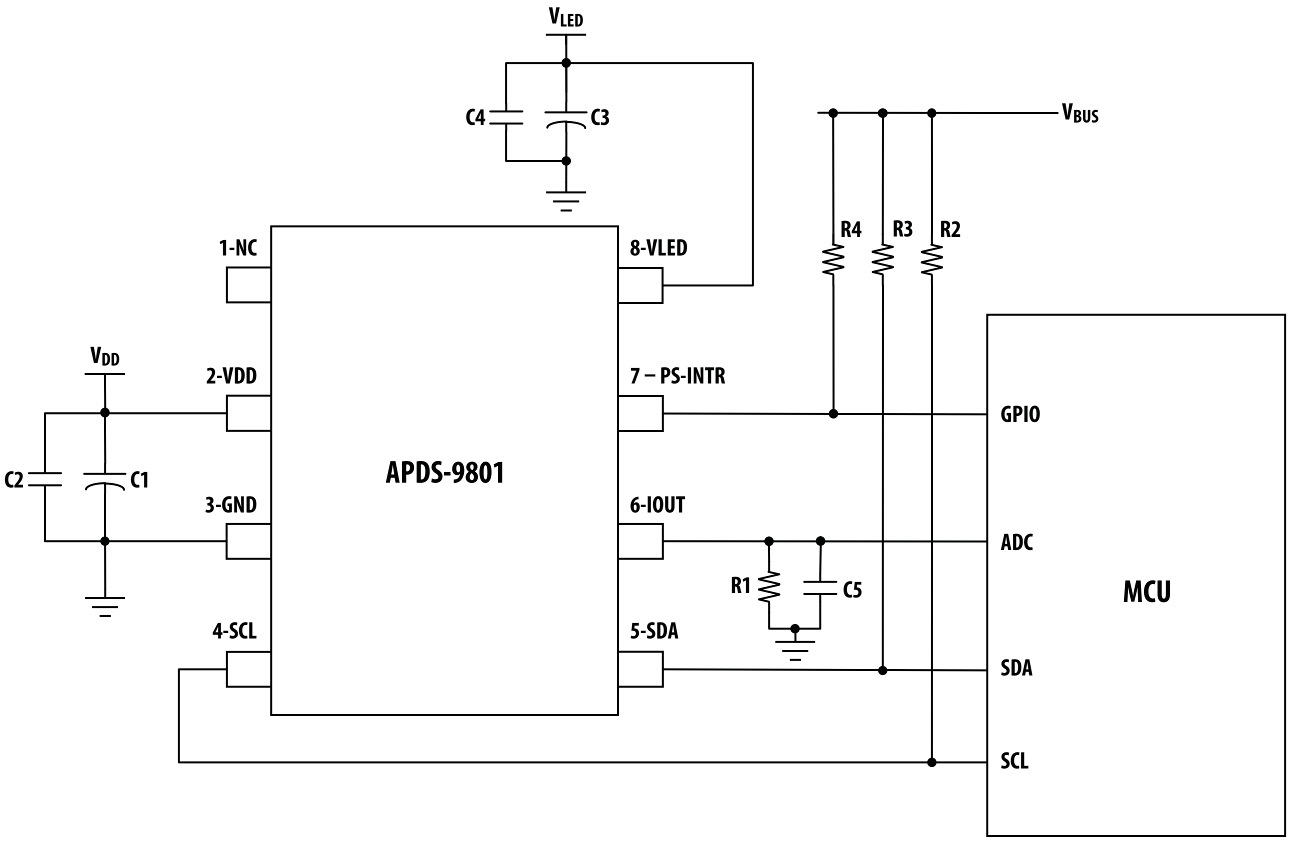
Parameter	Symbol	Min	Max	Unit	Condition
Operating Temperature	T_A	-40	70	ºC	
Storage Temperature	T_S	-40	85	ºC	
Supply Voltage	V_{DD}	1.7	2.5	V	
I^2C Bus Power Supply Voltage	V_{BUS}	1.7	3.6	V	
PS LED Power Supply Voltage	V_{LED}	2.5	5	V	

Electrical & Optical Specifications (T_A = 25°C)

Parameter	Symbol	Min.	Typ.	Max.	Unit	Conditions
ALS Output Current	I_{OUT}		83		µA	V_{DD}=1.8 V, Ev=100 Lux, [1]
ALS Dark Current			300		nA	V_{DD}=1.8 V, Ev=0 Lux
ALS Peak Spectral Sensitivity			560		nm	
ALS Light Current Ratio			1.1			[2]
ALS Saturation Voltage	V_{SAT}	V_{DD}-1.0	V_{DD}-0.8		V	Load=150 kΩ, V_{DD}=1.8 V, Ev=100 Lux, [1]
SCL, SDA Input High Voltage	V_{IH}	1.25			V	
SCL, SDA Input Low Voltage	V_{IL}			0.54	V	
INTR, SDA Output Low Voltage (Open Drain) [PS]	V_{OL}			0.3	V	I_{SINK}=3 mA
				0.6	V	I_{SINK}=6 mA
I_{AVG} at 5 ms delay time			1.3		mA	V_{DD}=1.8 V, I_{DD}+I_{LEDAVG} [3]
I_{AVG} at 50 ms delay time			180		µA	V_{DD}=1.8 V, I_{DD}+I_{LEDAVG} [4]
I_{AVG} at 500 ms delay time			90		µA	V_{DD}=1.8 V, I_{DD}+I_{LEDAVG} [5]
Shutdown Current	I_{SD}		1		µA	V_{DD}=1.8 V, Ev=0 Lux
PS Output Count			1300		counts	Kodak 18% grey card, 30 mm distance, Freq=100 kHz, n=20 pulses, Duty-cycle=25%, I_{LED}=100 mA, V_{DD}=1.8 V, [6] Refer to Figure 6.
LED Peak Wavelength			940		nm	
PS LED - Output Current Peak	I_{LED}	75, 100, 125, 150			mA	Programmable via I²C bus
PS-LED Pulse Frequency		50, 100, 200			kHz	Programmable via I²C bus
PS-Pulse Duty-Cycle		12.5%, 25%, 37.5%, 50%				Programmable via I²C bus
PS-Number of Pulses		4, 8, 12, 16, 20, 24, 28, 32				Programmable via I²C bus
PS-Burst Interval Delay		5, 20, 50, 125, 250, 500, 1000, 2000			ms	Programmable via I²C bus
Full Scale ADC Count (PS)				4092	counts	Programmable via I²C bus
Crosstalk (PS)				250	counts	Freq=100kHz, n=20pulses, Duty-cycle=25%, I_{LED}=100mA, V_{DD}=1.8V, [7]

Note:
1. White LED is used as light source.
2. V_{DD}=1.8 V, Current Light Ratio = (output current at 100 Lux Incandescent) / (output current at 100 Lux Fluorescent).
3. Test conditions: V_{DD}=1.8 V, 5 ms delay, I_{LED}=100 mA, 20 pulses, 25% duty cycle, Freq=100 kHz, Ev=0 Lux.
4. Test conditions: V_{DD}=1.8 V, 50 ms delay, I_{LED}=100 mA, 20 pulses, 25% duty cycle. Freq=100 kHz. Ev=0 Lux.
5. Test conditions: V_{DD}=1.8 V, 500 ms delay, I_{LED}=100 mA, 20 pulses, 25% duty cycle. Freq=100 kHz. Ev=0 Lux.
6. Test without window between sensor and grey card object.
7. Test without window or object above sensor.

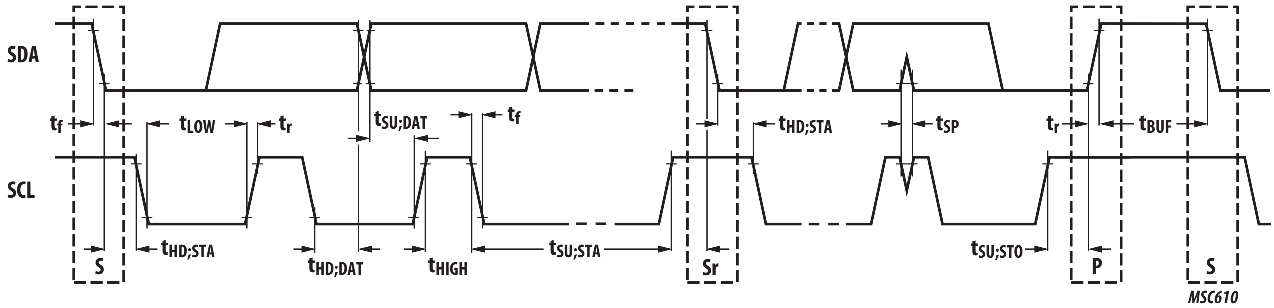
APDS–9801 Typical Application Circuit



R1	1 kΩ 1/16W 5%
R2, R3, R4	10 kΩ 5%
C1, C3	6.8 µF 10V
C2, C4	100 nF 10V
C5	10 µF 10V

Definition of timing for I^{2}C devices

This section will describe the main protocol of the I2C bus. For more details and timing diagrams, please refer to the I^{2}C bus specification.

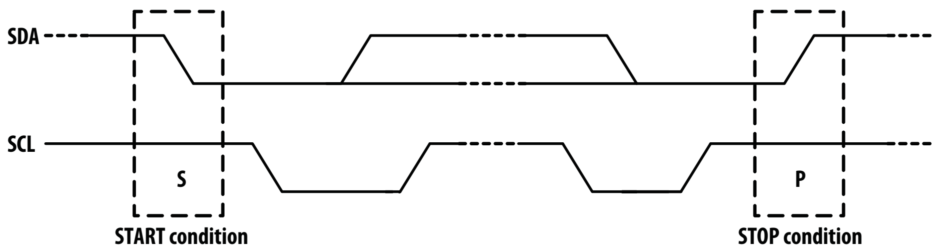


Characteristics of the SDA and SCL bus lines for I^{2}C-bus devices

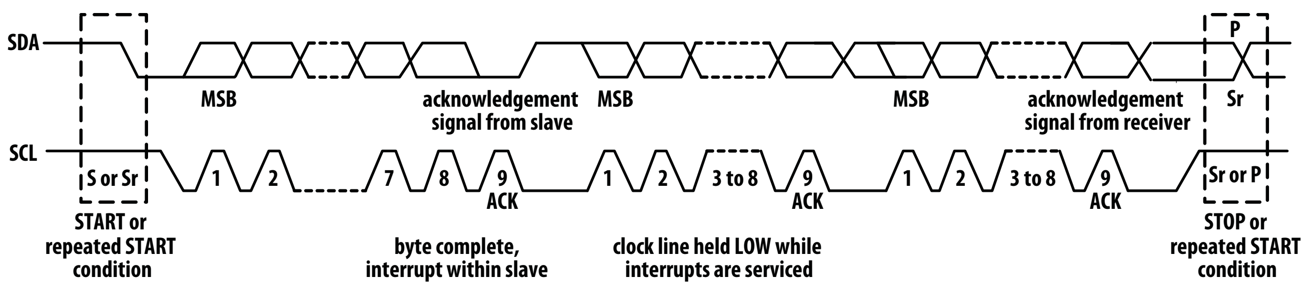
PARAMETER	SYMBOL	STANDARD-MODE MIN.	STANDARD-MODE MAX.	FAST-MODE MIN.	FAST-MODE MAX.	UNIT
SCL clock frequency	f_{SCL}	0	100	0	400	kHz
Hold time (repeated) START condition. After this period, the first clock pulse is generated	$t_{HD;STA}$	4.0	–	0.6	–	µs
LOW period of the SCL clock	t_{LOW}	4.7	–	1.3	–	µs
HIGH period of the SCL clock	t_{HIGH}	4.0	–	0.6	–	µs
Set-up time for a repeated START condition	$t_{SU;STA}$	4.7	–	0.6	–	µs
Data hold time:	$t_{HD;DAT}$	300	–	300	–	ns
Data set-up time	$t_{SU;DAT}$	250	–	100	–	ns
Rise time of both SDA and SCL signals	t_r	–	1000	–	300	ns
Fall time of both SDA and SCL signals	t_f	–	300		300	ns
Set-up time for STOP condition	$t_{SU;STO}$	4.0	–	0.6	–	µs
Bus free time between a STOP and START condition	t_{BUF}	4.7	–	1.3	–	µs
Capacitive load for each bus line	C_b	–	400	–	400	pF
Noise margin at the LOW level for each connected device (including hysteresis)	V_{nL}	$0.1V_{BUS}$	–	$0.1V_{BUS}$	–	V
Noise margin at the HIGH level for each connected device (including hysteresis)	V_{nH}	$0.2V_{BUS}$	–	$0.2V_{BUS}$	–	V

I²C Definition

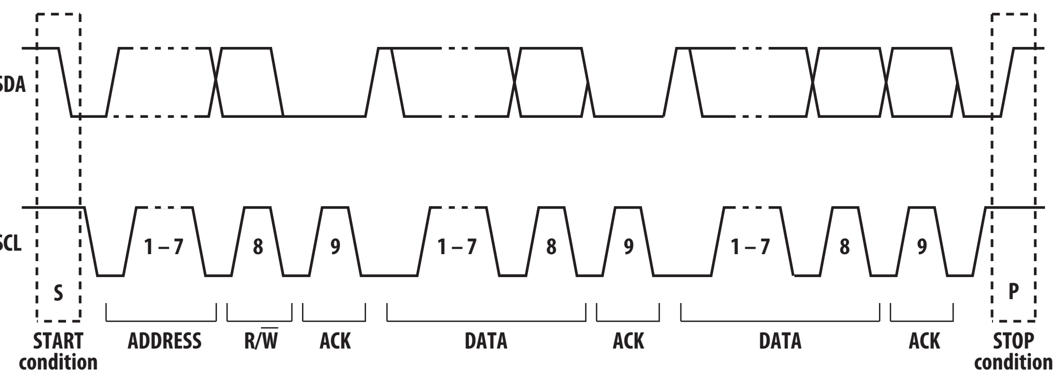
Start and Stop conditions



Data transfer on I2C-bus



A complete data transfer



1	7	1	1	8	1	1
S	Slave Address	Wr/Rd	A	Data Byte	A	P

S Start Condition

Wr Write "0"

Rd Write "1"

A Acknowledge (0 for ACK or 1 for NACK)

P Stop Condition

Sr Repeated Start Condition

[white box] from Master to Slave

[grey box] from Slave to Master

Write Byte Protocol

1	7	1	1	8	1	8	1	1
S	Slave Address	Wr	A	Common Code	A	Data Byte	A	P

Read Byte Protocol

1	7	1	1	8	1	1	7	1	1	8	1	1
S	Slave Address	Wr	A	Common Code	A	Sr	Slave Address	Rd	A	Data Byte	A	P

Slave Address

APDS-9801 PS slave address is 1010101 [0X55]

PS-I2C Interfacing

Register Address for PS:

ADDRESS	Register Name	Register Function
–	Command	Specifies register address
0h	Shutdown	Power on/off
1h	Pulse_Freq	Set the period, duty cycle and number of pulses for burst pulses
2h	Interval delay & control	Set the delay time between burst pulses & control
3h	Thres_low	Low byte of interrupt threshold
4h	Thres_high	High byte of interrupt threshold
5h	Data_low	Low byte of ADC output
6h	Data_high	High byte of ADC output
7h	interrupt	Interrupt status and enable

Command Register

The command register specifies the address of the target register for subsequence read and write operations. The write byte protocol is used to configure the COMMAND register.

7	6	5	4	3	2	1	0
CMD	Threshold Interrupt clear	EOC Interrupt clear	Software Reset	Address			

Reset Value: 0x00h

FIELD	BIT	Description
CMD	7	Select command register. Must be '1'
Threshold Interrupt clear	6	Clear the pending thresholds interrupt. Write '1' to clear. Self clearing.
EOC Interrupt clear	5	Clear the pending end of conversion of the ADC interrupt. Write '1' to clear. Self clearing.
Software Rest	4	Write 1 to this bit to reset the chip to default register value self clearing. This is for software reset only.
ADDRESS	3:0	Register address. This field selects the specific register.

Shutdown Register (0h)

7	6	5	4	3	2	1	0
							shutdown

Reset value: 0x0h

FIELD	BIT	Description
Shutdown	0	0 for shutdown, oscillator and analog block all turn off 1 for turn on, measurement triggered by "start measurement" bit of interval delay/control register

Pulse_Freq Register (1h)

7	6	5	4	3	2	1	0
Pulse count			reserved	Duty cycle		Pulse frequency	

Reset value: 0x96h

FIELD	BIT	Description
Pulse count	7:5	Set the number of pulses for each burst. 000: 4 pulses 001: 8 pulses 010: 12 pulses 011: 16 pulses 100: 20 pulses (default) 101: 24 pulses 110: 28 pulses 111: 32 pulses
Duty cycle	3:2	Set the duty cycle of the burst pulse. 00: 12.5% 01: 25.0% (default) 10: 37.5% 11: 50.0%
Pulse frequency	1:0	Set the period for the burst pulse. 00: None 01: 50 kHz 10: 100 kHz (default) 11: 200 kHz

Interval Delay & Control Register (2h)

7	6	5	4	3	2	1	0
reserved		Start Measurement	LED Current Control		Interval Delay		

Reset value: 0x00h

FIELD	BIT	Description
Start Measurement	5	Write a "1" to this bit to enable measurement. By default this bit is '0', no measurement.
LED Current Control	4:3	LED Current Control 00: 75 mA 01: 100 mA (default) 10: 125 mA 11: 150 mA
Interval Delay	2:0	Set the delay between the burst pulses. 000: 5ms 001: 20 ms 010: 50 ms 011: 125 ms 100: 250 ms 101: 500 ms (default) 110: 1s 111: 2s

Interrupt Threshold Register (Low byte) (3h)

7	6	5	4	3	2	1	0
Interrupt threshold low byte							

Reset value: 0x00h

FIELD	BIT	Description
Interrupt threshold Low byte	7:0	Lower byte of 12 bits Interrupt threshold. The 12 bit interrupt threshold values are expressed as 12 bits values spread across 2 registers (register address 3h & 4h).

Interrupt Threshold Register (High byte) (4h)

7	6	5	4	3	2	1	0
reserved				Interrupt threshold low byte			

Reset value: 0x00h

FIELD	BIT	Description
Interrupt threshold High byte	3:0	Upper 4 bits of 12 bits Interrupt threshold. The 12 bit interrupt threshold values are expressed as 12 bits values spread across 2 registers (register address 3h & 4h).

ADC Data Output Register (Low byte) (5h)

7	6	5	4	3	2	1	0
ADC output data low byte							

Reset value: 0x00h

FIELD	BIT	Description
ADC output data Low byte	7:0	Lower byte of 12bits ADC output data. The ADC data are expressed as 12 bits values spread across 2 registers (register address 5h & 6h). Read only.

ADC Data Output Register (High byte) (6h)

7	6	5	4	3	2	1	0
reserved				ADC output data high byte			

Reset value: 0x00h

FIELD	BIT	Description
ADC output data high byte	3:0	Upper 4 bits of 12 bits ADC output data. The ADC data are expressed as 12 bits values spread across 2 registers (register address 5h & 6h). Read only.

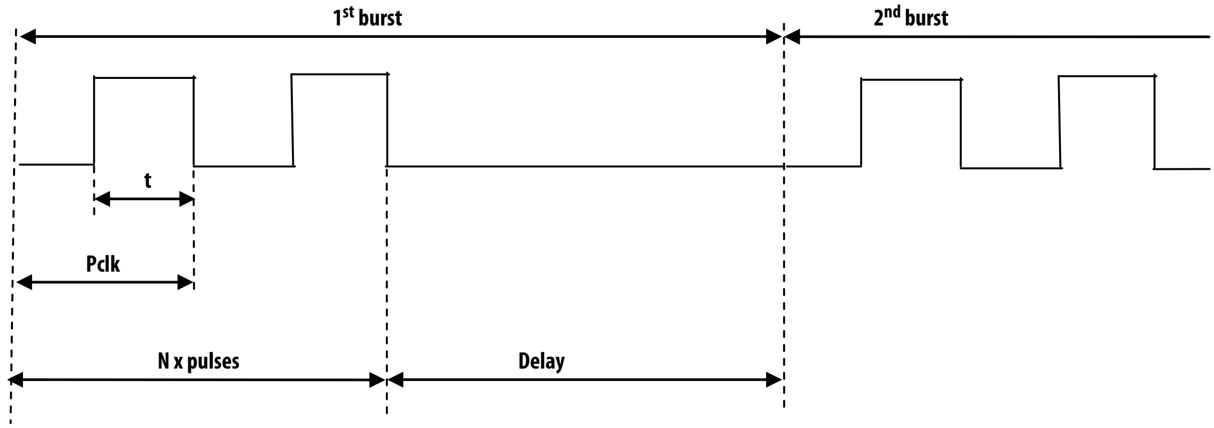
Interrupt Register (7h)

7	6	5	4	3	2	1	0
resv	Negative Threshold interrupt status	Positive Threshold interrupt status	EOC interrupt status	resv		Threshold Interrupt enable	EOC Interrupt enable

Reset value: 0x00h

FIELD	BIT	Description
Negative Threshold interrupt status	6	Read only. Interrupt happens when ADC output data value fall below the interrupt threshold set by threshold registers. Write a '1' to bit 6 of the command register to clear the interrupt. (note 2)
Positive Threshold interrupt status	5	Read only. Interrupt happens when ADC output data value rise above the interrupt threshold set by threshold registers. Write a '1' to bit 6 of the command register to clear the interrupt. (note 2)
EOC Interrupt status	4	Read only. Interrupt happens when it is the end of conversion for the ADC. Write a '1' to bit 5 of the command register to clear the interrupt.
Threshold Interrupt Enable	1	'1': threshold interrupt (when ADC rise above or fall below threshold set) enable to external interrupt pin. '0': threshold interrupt (when ADC rise above or fall below threshold set) disable to external interrupt pin.
EOC Interrupt Enable	0	'1': EOC interrupt enable to external interrupt pin. '0': EOC interrupt disable to external interrupt pin.

Note 1: Figure 3 Definition of transmit burst pulses

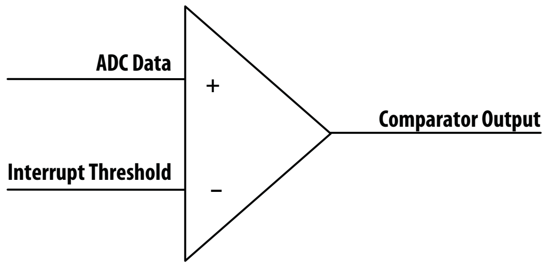


Duty cycle = t/Pclk

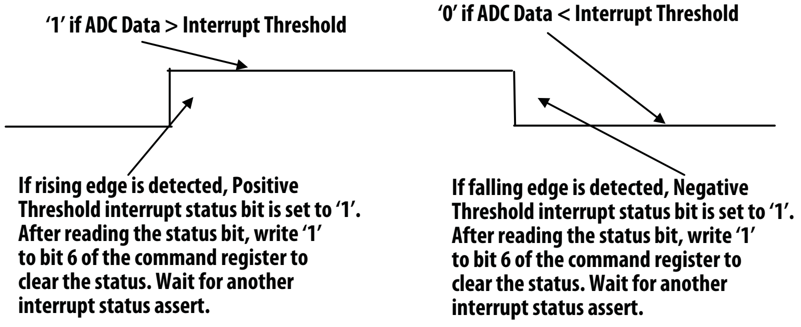
Delay = the time between the last burst pulse to the first burst pulse of the next burst

Note 2: Interrupt Status Implementation

The following diagram explained how the positive threshold interrupt status and negative threshold interrupt status is implemented. ADC Data and Interrupt threshold is compared. The output is high or low depends on the comparison result. The detection of rising edge of the comparator set the positive threshold interrupt status bit to '1'. The detection of falling edge of the comparator set the negative threshold interrupt status bit to '1'.



Comparator output waveform:



PS – APPLICATION SOFTWARE

Configuration the registers

The Pulse_Freq register and Interval Delay and Control register are initialized to default values when power up. Setting these registers to desired values would be part of setup procedure. The value can be change to optimize the performance need. Below are samples code illustrates the setting of registers for various option.

Set up Pulse_Freq

```
//20 pulses, 25% Duty Cycle and 100kHz Pulse Frequency
DeviceAddr = 0x55                          //Slave address also be –or 0x55
Command = 0x81                             //Set Command bit and address of Pulse_Freq
Value = 0x86
WriteByte_i2c(DeviceAddr, Command, Value)

//24 pulses, 50% Duty Cycle and 100kHz Pulse Frequency
DeviceAddr = 0x55
Command = 0x81
Value = 0x8E
WriteByte_i2c(DeviceAddr, Command, Value)
```

Set up Interval Delay and LED Current Control

```
//100mA of LED current and 5ms interval delay between the burst pulses
DeviceAddr = 0x55                          //Slave address also be – 0x55
Command = 0x82                             //Set Command bit and address of register
Value = 0x08
WriteByte_i2c(DeviceAddr, Command, Value)

//150mA of LED current and 250ms interval delay between the burst pulses
DeviceAddr = 0x55
Command = 0x82
Value = 0x1C
WriteByte_i2c(DeviceAddr, Command, Value)
```

Enable Measurement

```
DeviceAddr = 0x55
Command = 0x82
ReadByte_i2c(DeviceAddr, Command, & Value)  //Read back register value
Value |= 0x20                               //Set Enable Measurement bit
WriteByte_i2c(DeviceAddr, Command, Value)
```

PS – Interrupts

The interrupt feature of the device is simplifies and improves system efficiency by eliminating the need to poll the sensor for proximity distance value. The feature may enable at Interrupt Register.

An interrupt will be happen when the values of ADC conversion value change from lower to upper or upper to lower over the interrupt threshold value. Negative Threshold interrupt status show when ADC output values fall below interrupt threshold from upper, and Positive Threshold interrupt status are vice versa.

End of ADC Conversion interrupt also can be use. An interrupt will be generated when completion of each conversion of ADC.

Write '1' to bit 6 of the command register to clear the threshold interrupt or write '1' to bit 5 of the command register to clear the End_of_Conversion interrupt.

<u>Set up Threshold Interrupt</u>

```
//Example threshold value = 0x1CA

//Write the interrupt threshold low byte
DeviceAddr = 0x55                        //Slave address also be – 0x55
Command = 0x83                           //Set Command bit and addr of thresh low byte register
Value = 0xCA
WriteByte_i2c(DeviceAddr, Command, Value)

//Write the interrupt threshold high byte
DeviceAddr = 0x55                        //Slave address also be – 0x55
Command = 0x84                           //Set Command bit and addr of thresh high byte register
Value = 0x01
WriteByte_i2c(DeviceAddr, Command, Value)
```

<u>Enable Interrupt to external interrupt pin</u>

```
//Enable Threshold interrupt
DeviceAddr = 0x55
Command = 0x87                           //Set Command bit and addr of interrupt register
Value = 0x02                             //Enable Threshold Interrupt
WriteByte_i2c(DeviceAddr, Command, Value)

//Enable End_of_Conversion interrupt
DeviceAddr = 0x55
Command = 0x87                           //Set Command bit and addr of interrupt register
Value = 0x01                             //Enable EOC Interrupt
WriteByte_i2c(DeviceAddr, Command, Value)
```

<u>Clear the pending Interrupt</u>

```
//Clear pending Threshold interrupt
DeviceAddr = 0x55                        //Slave address also be – 0x55
Command = 0x40                           //Clear Threshold Interrupt
Write_i2c(DeviceAddr, Command)

//Clear pending End of Conversion of the ADC Interrupt
DeviceAddr = 0x55
Command = 0x20                           //Clear EOC Interrupt
Write_i2c(DeviceAddr, Command)
```

Read ADC Data Output Values

```
//Read the ADC low byte channel and store at ADC_DataLow
DeviceAddr = 0x55                              //Slave address also be – 0x55

//Set Command bit and addr of low byte of ADC Data output register
Command = 0x85
ReadByte_i2c(DeviceAddr, Command, &ADC_DataLow)

//Read the ADC high byte channel and store at ADC_DataHigh
//Set Command bit and addr of high byte of ADC Data output register
command = 0x86
ReadByte_i2c(DeviceAddr, Command, &ADC_DataHigh)
```

Software Examples

```
/**************************************************************************
Definitions
**************************************************************************/
#define    DeviceAddr         0x55    //Slave address for device

#define    CMD                0x80    //Command
#define    ADDR_SD            0x00    //Shutdown Register address
#define    ADDR_PFR           0x01    //Pulse_Freq Register address
#define    ADDR_ICR           0x02    //Interval delay & Control Register address
#define    ADDR_THRESLOW      0x03    //Threshold Low Register address
#define    ADDR_THRESHIGH     0x04    //Threshold High Register address
#define    ADDR_ADCDATALOW    0x05    //ADC Data Output Low Register address
#define    ADDR_ADCDATAHIGH   0x06    //ADC Data Output Low Register address
#define    ADDR_INTP          0x07    //Interrupt Register address

/**************************************************************************
Global Variable
**************************************************************************/
unsigned char PFR_Data;                                //hold data of Pulse_Freq Register
unsigned char ICR_Data;                                //hold data of Interval delay & Control Register
unsigned char ReadThresholdLow, ReadThresholdHigh      //read back threshold register value
unsigned char ADC_DataLow, ADC_DataHIGH                //read back ADC data
unsigned char IntSta;                                  //read back interrupt status
```

```
/**************************************************************************
Function Protoypes
**************************************************************************/
void PXS_ShutDown(unsigned char value);
void PXS_PulseFreq(unsigned char value);
void PXS_PulseFreqRead(void);
void PXS_IntvDelay(char value);
void PXS_MeasurementEna(void);
void PXS_IntvDelayRead(void);
void PXS_ThresHold(unsigned char ThresholdLow, unsigned char ThresholdHigh);
void PXS_ThresHoldRead(void);
void PXS_IntpEna(unsigned char value);
void PXS_IntpStatusRead(void);
void PXS_IntpClr(void);
void PXS_SoftwareReset(void);
void PXS_ADCRead(void);

void Main(void)
{
	PXS_ShutDown(1);				//Power ON.
	PXS_PulseFreq(0x86);				//20 pulses,
							//25% Duty cycle, 100kHz pulse frequency
	PXS_IntvDelay(0x08);				//100mA LED Current Control and
							//5ms delay between burst pulses
	PXS_ThresHold(0xCA, 0x00);			//Set Threshold values
	PXS_IntpEna(0x20);				//Enable Interrupt Threshold
	PXS_MeasurementEna();				//Enable Measurement

	While(1)
	{
		If(IntEna)				//if Interrupt occur
		{
			PXS_IntpStatusRead();		//Read Interrupt Status
			PXS_ADCRead();			//Read ADC Output value
			PXS_IntpClr();			//Clear Interrupt
		}
	}
}
```

```
/*************************************************************************
DESC: 0 for shutdown, oscillator and analog block turn off
        1 for Power ON, measurement triggered by "start measurement" bit of interval delay/control register
RETURNS: Nothing
*************************************************************************/
void PXS_ShutDown (unsigned char value)        //1 = ON, 0 = OFF
{
    unsigned char command;

    command = CMD| ADDR_SD;                        //Address of Shutdown Register
    WriteByte_i2c(DeviceAddr, command, value);
}


/*************************************************************************
DESC: Set the period, duty cycle and number of pulses for burst pulses
RETURNS: Nothing
*************************************************************************/
void PXS_PulseFreq(unsigned char value)
{
    unsigned char command, value;

    command = CMD|ADDR_PFR;                        //Address of Pulse_Freq   Register
    WriteByte_i2c(DeviceAddr, command, value);
}


/*************************************************************************
DESC: Read back Pulse_Freq Register Value
RETURNS: Pulse_Freq Register Value
*************************************************************************/
void PXS_PulseFreqRead(void)
{
    unsigned char command;

    command = CMD|ADDR_PFR;                        //Address of Pulse_Freq Register
    ReadByte_i2c(DeviceAddr, command, &PFR_Data);
}


/*************************************************************************
DESC: Set the delay time between burst pulses & LED Current Control
RETURNS: Nothing
*************************************************************************/
void PXS_IntvDelay(char value)
{
    unsigned char command;

    command = CMD|ADDR_ICR;                        //Address of Interval Delay & Control Register
    ICR_Data = value;
    WriteByte_i2c(DeviceAddr, command, ICR_Data);
}
```

```
/**********************************************************************
DESC: Enable measurement
RETURNS: Nothing
**********************************************************************/
void PXS_MeasurementEna(void)
{
	unsigned char command;

	command = CMD|ADDR_ICR;				//Address of Interval Delay & Control Register
	ICR_Data = 0x20 | ICR_Data;			//Enable measurement
	WriteByte_i2c(DeviceAddr, command, ICR_Data);
}


/**********************************************************************
DESC: Read back Interval delay & Control Register Value
RETURNS: Interval delay & Control Register Value
**********************************************************************/
void PXS_IntvDelayRead(void)
{
	unsigned char command;

	command = CMD|ADDR_ICR;
	ReadByte_i2c(DeviceAddr, command, &ICR_Data);
}


/**********************************************************************
DESC: Set the interrupt threshold
RETURNS: Nothing
**********************************************************************/
void PXS_ThresHold(unsigned char ThresholdLow, unsigned char ThresholdHigh)
{
	unsigned char command;

	command = CMD|ADDR_THRESLOW;
	WriteByte_i2c(DeviceAddr, command, ThresholdLow);		//Set Interrupt Threshold Low Byte data

	command = CMD|(ADDR_THRESHIGH);
	WriteByte_i2c(DeviceAddr, command, ThresholdHigh);		//Set Interrupt Threshold High Byte data
}
```

```
/************************************************************************
DESC: Read back interrupt threshold value
RETURNS: Threshold Register value
************************************************************************/
void PXS_ThresHoldRead(void)
{
        unsigned char command;

        //Read back Interrupt Threshold Low Byte data
        command = CMD|ADDR_THRESLOW;
        ReadByte_i2c(DeviceAddr, command, &ReadThresholdLow);

        //Read back Interrupt Threshold High Byte data
        command = CMD|(ADDR_THRESHIGH);
        ReadByte_i2c(DeviceAddr, command, &ReadThresholdHigh);
}


/************************************************************************
DESC: Set the interrupt enable
RETURNS: Nothing
************************************************************************/
void PXS_IntpEna(unsigned char value)
{
        unsigned char command;

        command = CMD|ADDR_INTP;
        WriteByte_i2c(DeviceAddr, command, value);
}


/************************************************************************
DESC: Read back Interrupt Register Status
RETURNS: Interrupt status
************************************************************************/
void PXS_IntpStatusRead(void)
{
        unsigned char command;

        command = CMD|ADDR_INTP;
        ReadByte_i2c(DeviceAddr, command, &IntSta);
}


/************************************************************************
DESC: Clear the pending thresholds & EOC interrupt
RETURNS: Nothing
************************************************************************/
void PXS_IntpClr(void)
{
        unsigned char command;

        //Set the Clear bits of the pending threshold interrupt and EOC interrupt
        command = 0x60;
        Write_i2c(DeviceAddr, command);
}
```

```
/*********************************************************************
DESC: Software Reset
RETURNS: Nothing
*********************************************************************/
void PXS_SoftwareReset(void)
{
	unsigned char command;

	command = 0x10;					//Set the Software Reset bits
	Write_i2c(DeviceAddr, command);
}


/*********************************************************************
DESC: Read ADC Data Output Value
RETURNS: ADC Low Byte and ADC HIGH Byte value
*********************************************************************/
void PXS_ADCRead(void)
{
	unsigned char command;

	//Read ADC Register Low Byte data
	command = CMD|ADDR_ADCDATALOW;
	ReadByte_i2c(DeviceAddr, command, &ADC_DataLow);

	//Read ADC Register High Byte data
	command = CMD|(ADDR_ADCDATAHIGH);
	ReadByte_i2c(DeviceAddr, command, &ADC_DataHIGH);
}
```

Typical Characteristics

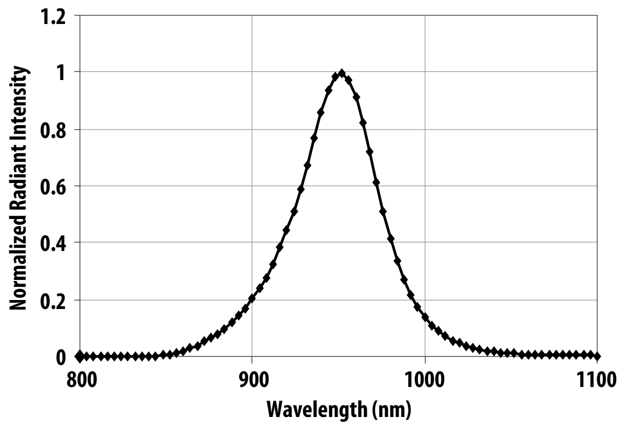


Figure 1. PS LED Spectral response

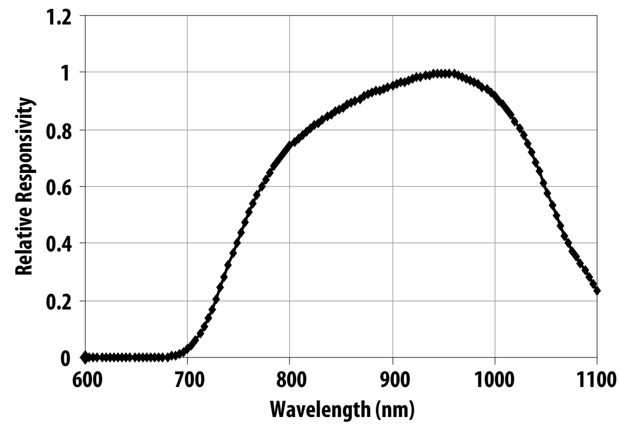


Figure 2. PS Photo Detector Spectral Response

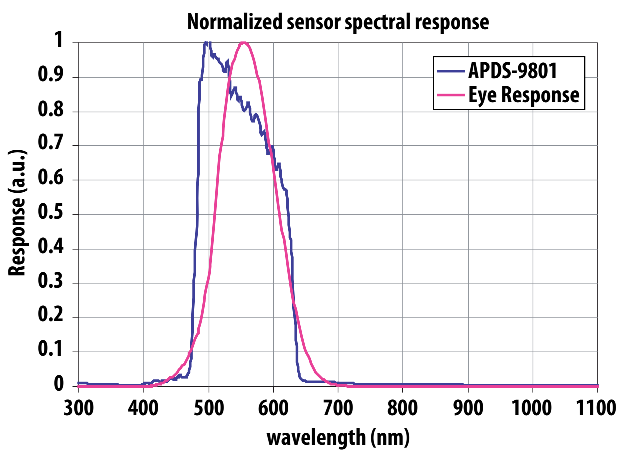


Figure 3. ALS Photo Detector Spectral Response

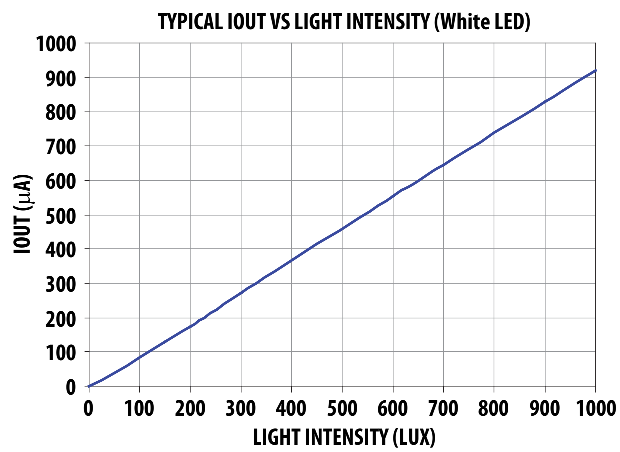


Figure 4. ALS Typical Output Current vs. Light Intensity

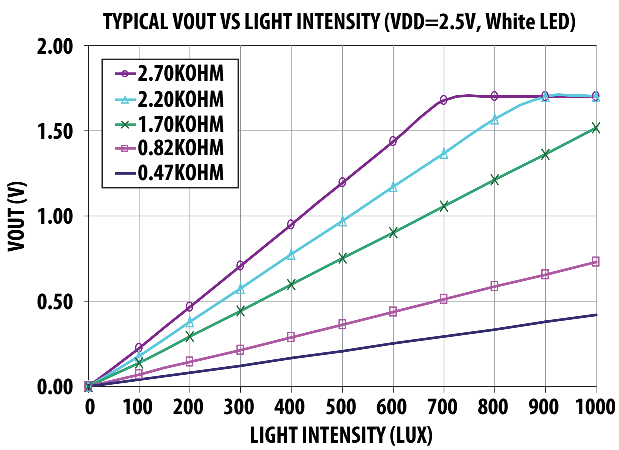


Figure 5. ALS Typical Output Voltage vs. Light Intensity

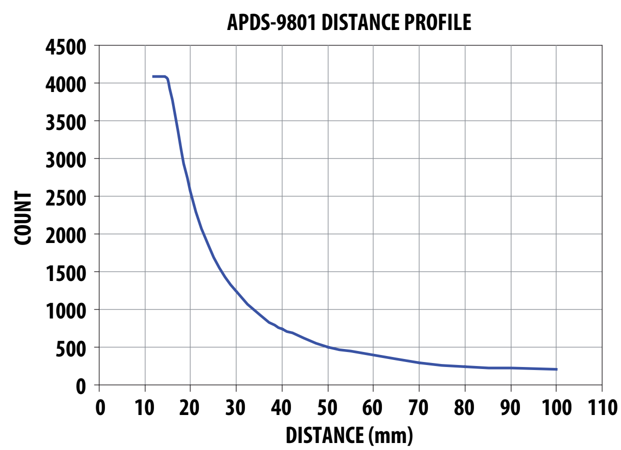
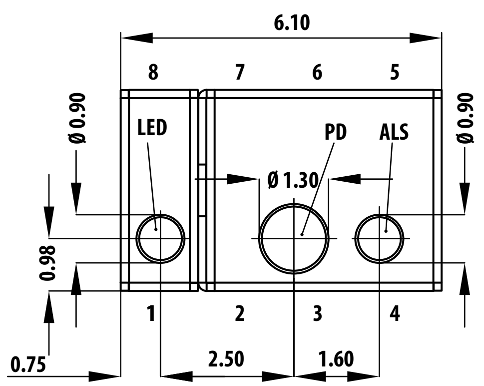


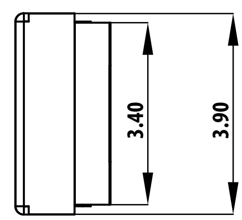
Figure 6. PS Output vs. Distance Typical Profile

APDS-9801 Package Outline

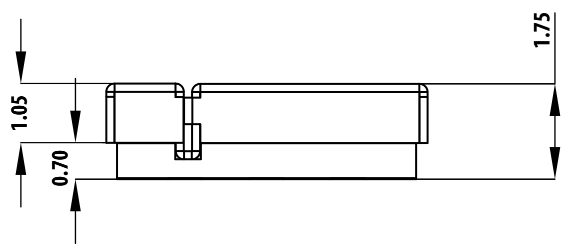
TOP VIEW



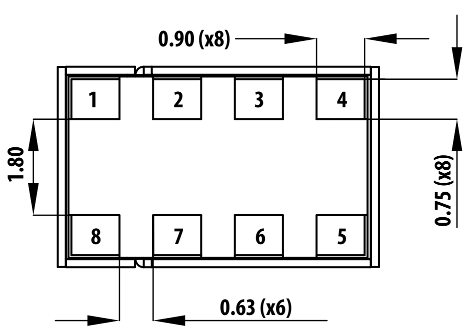
RIGHT VIEW



FRONT VIEW



BOTTOM VIEW

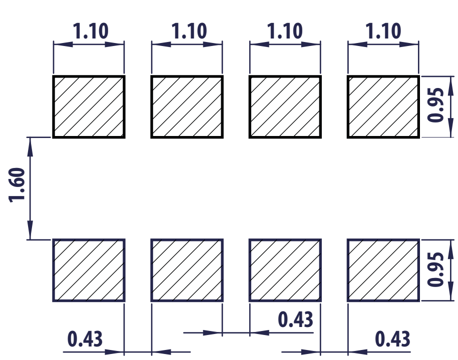


PINOUT

1. NC
2. VDD
3. GRD
4. SCLK
5. SDA
6. IOUT
7. PS-INTR
8. VLED

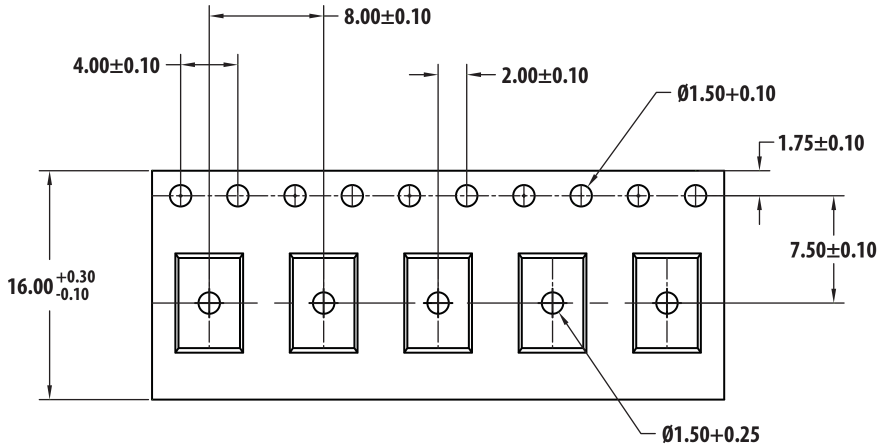
Notes:
1. All dimensions are in millimeters. Dimension tolerance is ±0.1 mm unless otherwise stated.

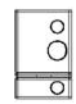
Recommended Land Pattern



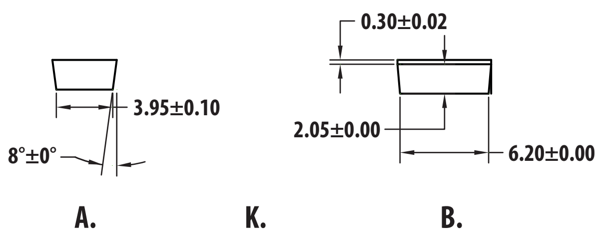
Notes:
1. All dimensions are in millimeters.
2. Do NOT connect NC pins.

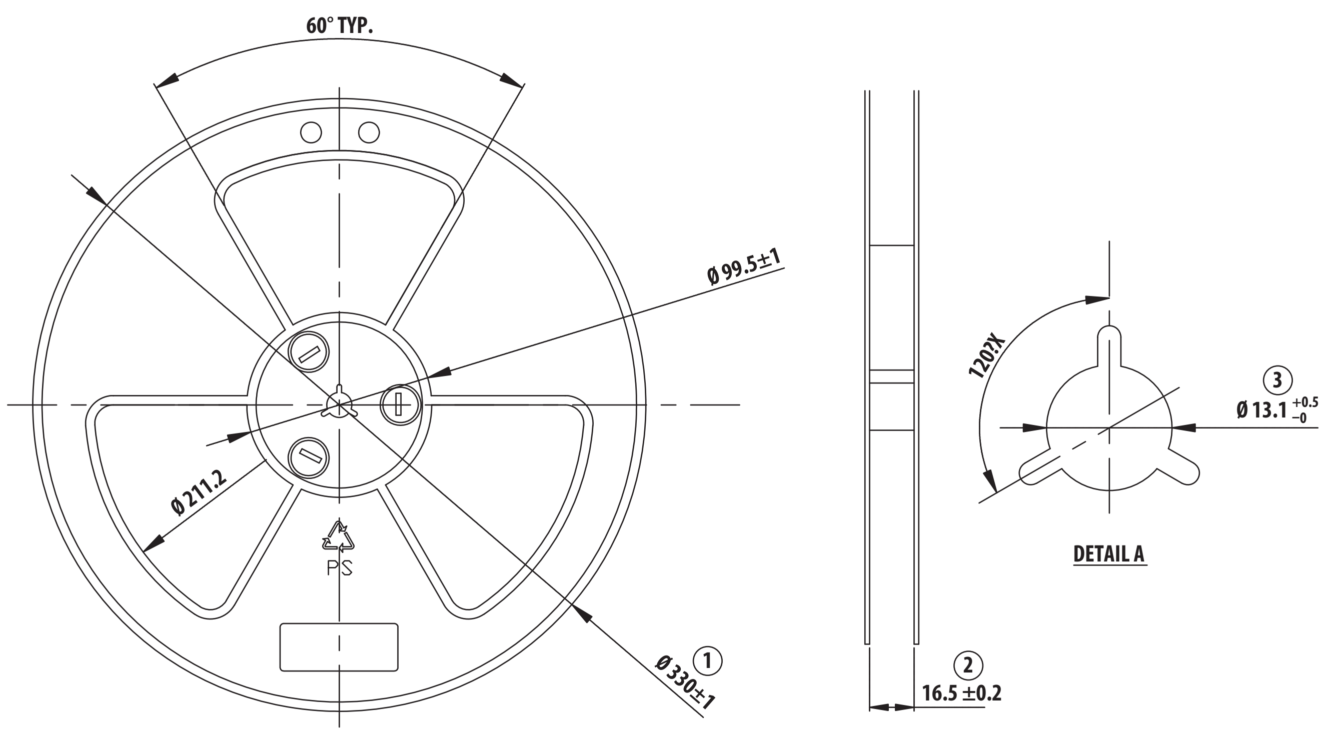
APDS-9801 Tape & Reel Dimensions





Unit Orientation in Carrier Tape

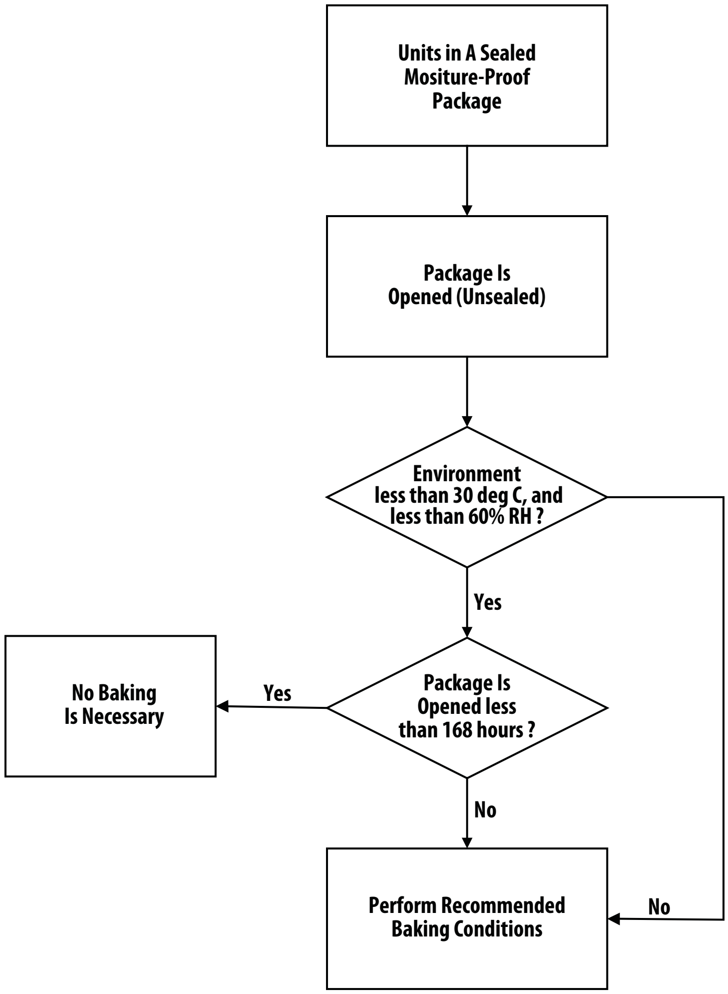




All dimensions in millimeter.

Moisture Proof Packaging

All APDS-9801 options are shipped in moisture proof package. Once opened, moisture absorption begins. This part is compliant to JEDEC MSL 3.



Baking Conditions

Package	Temp.	Time
In Reels	60°C	48 hours
In Bulk	100°C	4 hours

If the parts are not stored in dry conditions, they must be baked before reflow to prevent damage to the parts.

Baking should only be done once.

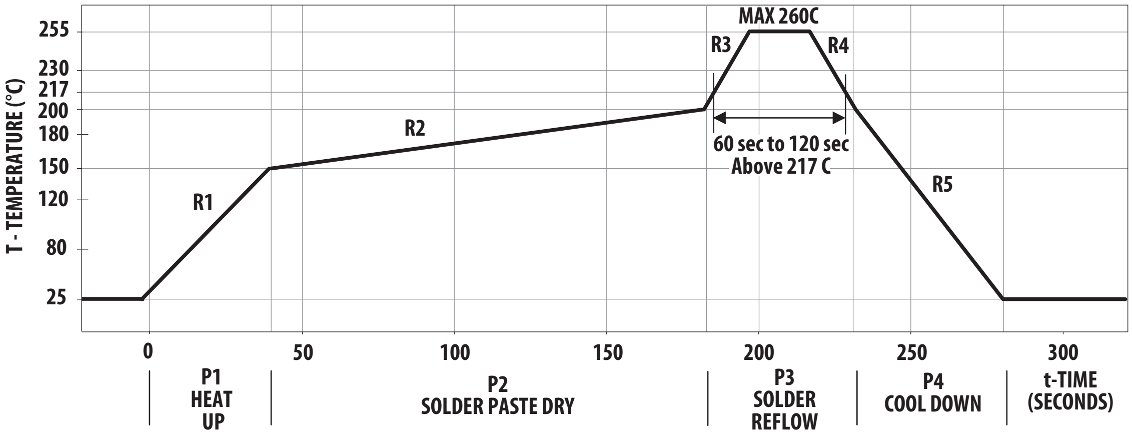
Recommended Storage Conditions

Storage Temperature	10°C to 30°C
Relative Humidity	Below 60% RH

Time from unsealing to soldering

After removal from the bag, the parts should be soldered within 168 hours if stored at the recommended storage conditions. If times longer than 168 hours are needed, the parts must be stored in a dry box.

Recommended Reflow Profile



Process Zone	Symbol	ΔT	Maximum ΔT/ Δtime or Duration
Heat Up	P1, R1	25°C to 150°C	3°C/s
Solder Paste Dry	P2, R2	150°C to 200°C	100s to 180s
Solder Reflow	P3, R3 P3, R4	200°C to 260°C 260°C to 200°C	3°C/s -6°C/s
Cool Down	P4, R5	200°C to 25°C	-6°C/s
Time maintained above liquidus point, 217°C		> 217°C	60s to 120s
Peak Temperature		260°C	–
Time within 5°C of actual Peak Temperature		> 255°C	20s to 40s
Time 25°C to Peak Temperature		25°C to 260°C	8mins

The reflow profile is a straight-line representation of a nominal temperature profile for a convective reflow solder process. The temperature profile is divided into four process zones, each with different ΔT/Δtime temperature change rates or duration. The ΔT/Δtime rates or duration are detailed in the above table. The temperatures are measured at the component to printed circuit board connections.

In **process zone P1**, the PC board and component pins are heated to a temperature of 150°C to activate the flux in the solder paste. The temperature ramp up rate, R1, is limited to 3°C per second to allow for even heating of both the PC board and component pins.

Process zone P2 should be of sufficient time duration (100 to 180 seconds) to dry the solder paste. The temperature is raised to a level just below the liquidus point of the solder.

Process zone P3 is the solder reflow zone. In zone P3, the temperature is quickly raised above the liquidus point of solder to 260°C (500°F) for optimum results. The dwell time above the liquidus point of solder should be between 60 and 120 seconds. This is to assure proper coalescing of the solder paste into liquid solder and the formation of good solder connections. Beyond the recommended dwell time the intermetallic growth within the solder connections becomes excessive, resulting in the formation of weak and unreliable connections. The temperature is then rapidly reduced to a point below the solidus temperature of the solder to allow the solder within the connections to freeze solid.

Process zone P4 is the cool down after solder freeze. The cool down rate, R5, from the liquidus point of the solder to 25°C (77°F) should not exceed 6°C per second maximum. This limitation is necessary to allow the PC board and component pins to change dimensions evenly, putting minimal stresses on the component.

It is recommended to perform reflow soldering no more than twice.

For product information and a complete list of distributors, please go to our web site: **www.avagotech.com**

AV02-2573EN - September 6, 2010

AVAGO TECHNOLOGIES