# TSL2560, TSL2561

## Light-to-Digital Converters

## General Description

The TSL2560 and TSL2561 are light-to-digital converters that transform light intensity to a digital signal output capable of direct $I^2C$ (TSL2561) or SMBus (TSL2560) interface. Each device combines one broadband photodiode (visible plus infrared) and one infrared-responding photodiode on a single CMOS integrated circuit capable of providing a near-photopic response over an effective 20-bit dynamic range (16-bit resolution). Two integrating ADCs convert the photodiode currents to a digital output that represents the irradiance measured on each channel. This digital output can be input to a microprocessor where illuminance (ambient light level) in lux is derived using an empirical formula to approximate the human eye response. The TSL2560 device permits an SMB-Alert style interrupt, and the TSL2561 device supports a traditional level style interrupt that remains asserted until the firmware clears it.

While useful for general purpose light sensing applications, the TSL2560/61 devices are designed particularly for display panels (LCD, OLED, etc.) with the purpose of extending battery life and providing optimum viewing in diverse lighting conditions. Display panel backlighting, which can account for up to 30 to 40 percent of total platform power, can be automatically managed. Both devices are also ideal for controlling keyboard illumination based upon ambient lighting conditions. Illuminance information can further be used to manage exposure control in digital cameras. The TSL2560/61 devices are ideal in notebook/tablet PCs, LCD monitors, flat-panel televisions, cell phones, and digital cameras. In addition, other applications include street light control, security lighting, sunlight harvesting, machine vision, and automotive instrumentation clusters.

*Ordering Information and Content Guide appear at end of datasheet.*

## Key Benefits & Features

The benefits and features of TSL2560 and TSL2561, Light-to-Digital Converters, are listed below:

**Figure 1:**
**Added Value of Using TSL2560 and TSL2561**

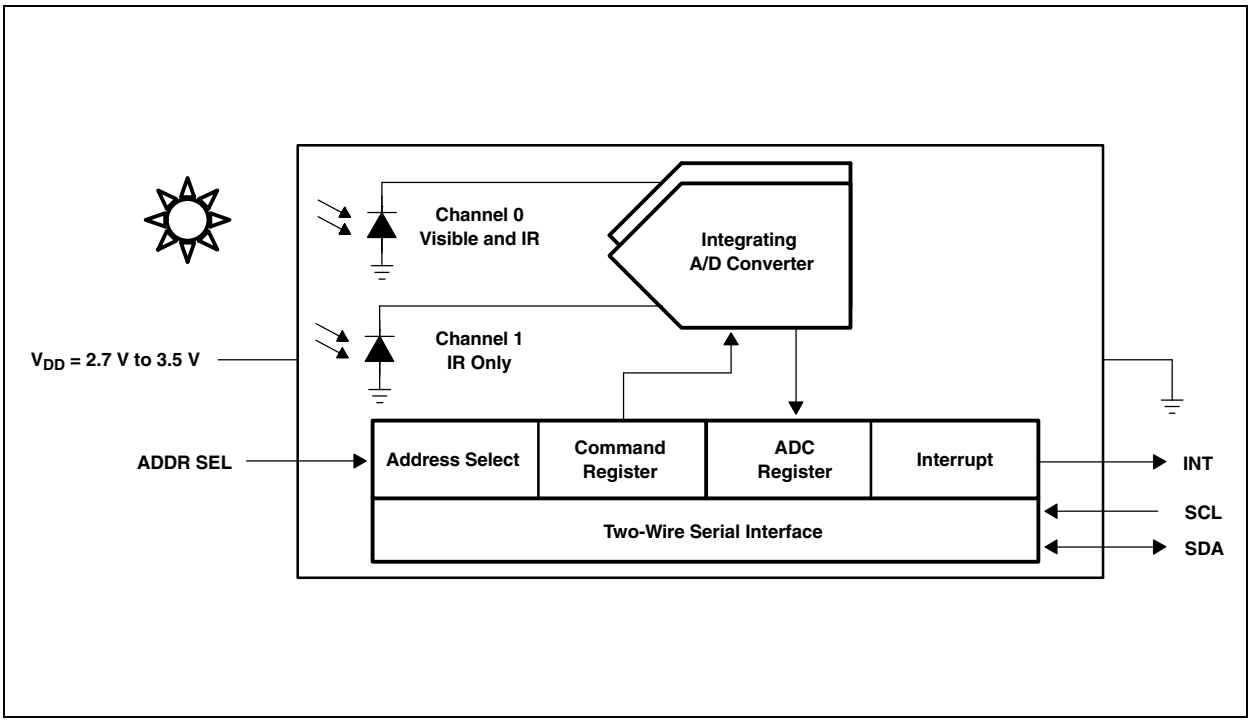
| Benefits | Features |
|---|---|
| • Enables Operation in IR Light Environments | • Patented Dual-Diode Architecture |
| • Enables Dark Room to High Lux Sunlight Operation | • 1M:1 Dynamic Range |
| • Reduces Micro-Processor Interrupt Overhead | • Programmable Interrupt Function |
| • Digital Interface is Less Susceptible to Noise | • SMBus (TSL2560) or $I^2C$ (TSL2561) Digital Interface |
| • Reduces Board Space Requirements while Simplifying Designs | • Available in 1.25mm x 1.75mm Chipscale or 2.6mm x 3.8mm TMB or 2mm x 2mm Dual Flat No-Lead (FN) Packages |

- Approximates Human Eye Response
- Programmable Interrupt Function with User-Defined Upper and Lower Threshold Settings
- 16-Bit Digital Output with SMBus (TSL2560) at 100kHz or $I^2C$ (TSL2561) Fast-Mode at 400kHz
- Programmable Analog Gain and Integration Time Supporting 1,000,000-to-1 Dynamic Range
- Automatically Rejects 50/60Hz Lighting Ripple
- Low Active Power (0.75mW Typical) with Power Down Mode

## Block Diagram

The functional blocks of this device are shown below:

**Figure 2:**
**TSL2560 and TSL2561 Block Diagram**



## Detailed Description

The TSL2560 and TSL2561 are second-generation ambient light sensor devices. Each contains two integrating analog-to-digital converters (ADC) that integrate currents from two photodiodes. Integration of both channels occurs simultaneously. Upon completion of the conversion cycle, the conversion result is transferred to the Channel 0 and Channel 1 data registers, respectively. The transfers are double-buffered to ensure that the integrity of the data is maintained. After the transfer, the device automatically begins the next integration cycle.

Communication to the device is accomplished through a standard, two-wire SMBus or $I^2C$ serial bus. Consequently, the TSL256x device can be easily connected to a microcontroller or embedded controller. No external circuitry is required for signal conditioning, thereby saving PCB real estate as well. Since the output of the TSL256x device is digital, the output is effectively immune to noise when compared to an analog signal.

The TSL256x devices also support an interrupt feature that simplifies and improves system efficiency by eliminating the need to poll a sensor for a light intensity value. The primary purpose of the interrupt function is to detect a meaningful change in light intensity. The concept of a *meaningful change* can be defined by the user both in terms of light intensity and time, or persistence, of that change in intensity. The TSL256x devices have the ability to define a threshold above and below the current light level. An interrupt is generated when the value of a conversion exceeds either of these limits.

## Pin Assignments

The TSL2560 and TSL2561 pin assignments are described below:

**Figure 3:**
**Pin Diagram of Package CS 6-Lead Chipscale (Top View)**

**Package CS 6-Lead Chipscale:**
Package drawings are not to scale

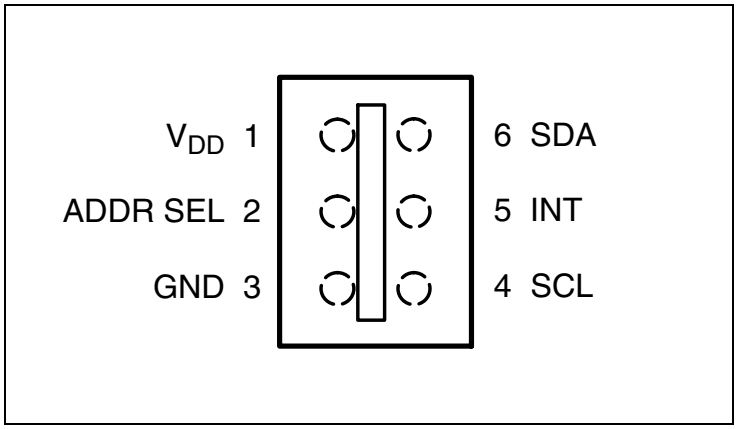



**Figure 4:**
**Pin Diagram of Package T 6-Lead TMB (Top View)**

**Package T 6-Lead TMB:**
Package drawings are not to scale

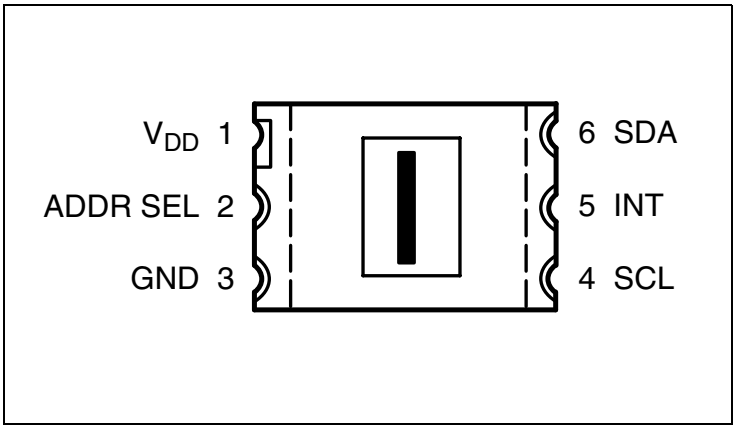



**Figure 5:**
**Pin Diagram of Package FN Dual Flat No-Lead (Top View)**

**Package FN Dual Flat No-Lead:**
Package drawings are not to scale

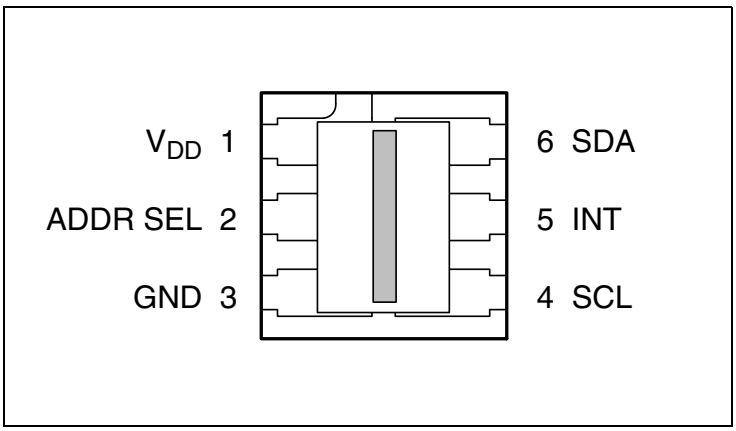


**Figure 6:**
**Pin Diagram of Package CL 6-Lead ChipLED (Top View)**

**Package CL 6-Lead ChipLED:**
Package drawings are not to scale

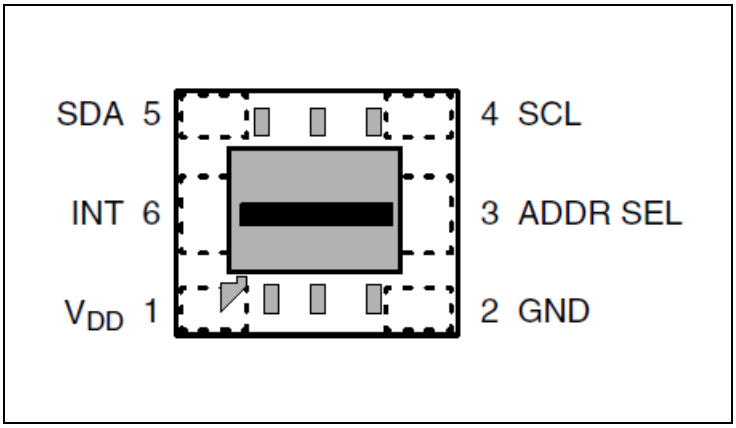



**Figure 7:**
**Terminal Functions**

| Terminal | | | Type | Description |
|---|---|---|---|---|
| Name | CS, T, FN Pkg No. | CL Pkg No. | | |
| ADDR SEL | 2 | 3 | I | SMBus device select - three-state |
| GND | 3 | 2 | | Power supply ground. All voltages are referenced to GND. |
| INT | 5 | 6 | O | Level or SMB Alert interrupt - open drain |
| SCL | 4 | 4 | I | SMBus serial clock input terminal - clock signal for SMBus serial data |
| SDA | 6 | 5 | I/O | SMBus serial data I/O terminal - serial data I/O for SMBus |
| $V_{DD}$ | 1 | 1 | | Supply voltage |

## Absolute Maximum Ratings

Stresses beyond those listed under Absolute Maximum Ratings may cause permanent damage to the device. These are stress ratings only. Functional operation of the device at these or any other conditions beyond those indicated under Recommended Operating Conditions is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

**Figure 8:**
**Absolute Maximum Ratings over Operating Free-Air Temperature Range (unless otherwise noted)**

| Symbol | Parameter | Min | Max | Unit |
|---|---|---|---|---|
| $V_{DD}$ | Supply voltage (1) | | 3.8 | V |
| $V_O$ | Digital output voltage range | -0.5 | 3.8 | V |
| $I_O$ | Digital output current | -1 | 20 | mA |
| $T_{strg}$ | Storage temperature range | -40 | 85 | °C |
| | ESD tolerance, human body model | ±2000 | | V |

**Note(s):**

1. All voltages are with respect to GND.

## Electrical Characteristics

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

**Figure 9:**
**Recommended Operating Conditions**

| Symbol | Parameter | Min | Nom | Max | Unit |
|---|---|---|---|---|---|
| $V_{DD}$ | Supply voltage | 2.7 | 3 | 3.6 | V |
| $T_A$ | Operating free-air temperature | -30 | | 70 | °C |
| $V_{IL}$ | SCL, SDA input low voltage | -0.5 | | 0.8 | V |
| $V_{IH}$ | SCL, SDA input high voltage | 2.1 | | 3.6 | V |

**Figure 10:**
**Electrical Characteristics over Recommended Operating Free-Air Temperature Range (unless otherwise noted)**

| Symbol | Parameter | Test Conditions | Min | Typ | Max | Unit |
|---|---|---|---|---|---|---|
| $I_{DD}$ | Supply current | Active | | 0.24 | 0.6 | mA |
| | | Power down | | 3.2 | 15 | µA |
| $V_{OL}$ | INT, SDA output low voltage | 3mA sink current | 0 | | 0.4 | V |
| | | 6mA sink current | 0 | | 0.6 | V |
| $I_{LEAK}$ | Leakage current | | -5 | | 5 | µA |

**Figure 11:**
**Operating Characteristics, High Gain (16×), $V_{DD}$ = 3V, $T_A$ = 25°C (unless otherwise noted)** (1), (2), (3), (4)

| Symbol | Parameter | Test Conditions | Channel | TSL2560T, FN, & CL TSL2561T, FN & CL | | | TSL2560CS, TSL2561CS | | | Unit |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | | Min | Typ | Max | Min | Typ | Max | |
| $f_{OSC}$ | Oscillator frequency | | | 690 | 735 | 780 | 690 | 735 | 780 | kHz |
| | Dark ADC count value | $E_e$ = 0, $T_{int}$ = 402ms | Ch0 | 0 | | 4 | 0 | | 4 | counts |
| | | | Ch1 | 0 | | 4 | 0 | | 4 | |
| | Full scale ADC count value (5) | $T_{int}$ > 178ms | Ch0 | | | 65535 | | | 65535 | counts |
| | | | Ch1 | | | 65535 | | | 65535 | |
| | | $T_{int}$ = 101ms | Ch0 | | | 37177 | | | 37177 | |
| | | | Ch1 | | | 37177 | | | 37177 | |
| | | $T_{int}$ = 13.7ms | Ch0 | | | 5047 | | | 5047 | |
| | | | Ch1 | | | 5047 | | | 5047 | |

| Symbol | Parameter | Test Conditions | Channel | TSL2560T, FN, & CL TSL2561T, FN & CL | | | TSL2560CS, TSL2561CS | | | Unit |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | | Min | Typ | Max | Min | Typ | Max | |
| | ADC count value | $\lambda_p$ = 640nm, $T_{int}$ = 101ms $E_e$ = 36.3µW/cm$^2$ | Ch0 | 750 | 1000 | 1250 | | | | counts |
| | | | Ch1 | | 200 | | | | | |
| | | $\lambda_p$ = 940nm, $T_{int}$ = 101ms $E_e$ = 119µW/cm$^2$ | Ch0 | 700 | 1000 | 1300 | | | | |
| | | | Ch1 | | 820 | | | | | |
| | | $\lambda_p$ = 640nm, $T_{int}$ = 101ms $E_e$ = 41µW/cm$^2$ | Ch0 | | | | 750 | 1000 | 1250 | counts |
| | | | Ch1 | | | | | 190 | | |
| | | $\lambda_p$ = 940nm, $T_{int}$ = 101ms $E_e$ = 135µW/cm$^2$ | Ch0 | | | | 750 | 1000 | 1250 | |
| | | | Ch1 | | | | | 850 | | |
| | ADC count value ratio: Ch1/Ch0 | $\lambda_p$ = 640nm, $T_{int}$ = 101ms | | 0.15 | 0.20 | 0.25 | 0.14 | 0.19 | 0.24 | |
| | | $\lambda_p$ = 940nm, $T_{int}$ = 101ms | | 0.69 | 0.82 | 0.95 | 0.70 | 0.85 | 1 | |
| $R_e$ | Irradiance responsivity | $\lambda_p$ = 640nm, $T_{int}$ = 101ms | Ch0 | | 27.5 | | | 24.4 | | counts/ (µW/cm$^2$) |
| | | | Ch1 | | 5.5 | | | 4.6 | | |
| | | $\lambda_p$ = 940nm, $T_{int}$ = 101ms | Ch0 | | 8.4 | | | 7.4 | | |
| | | | Ch1 | | 6.9 | | | 6.3 | | |
| $R_v$ | Illuminance responsivity | Fluorescent light source: $T_{int}$ = 402ms | Ch0 | | 36 | | | 35 | | counts/lux |
| | | | Ch1 | | 4 | | | 3.8 | | |
| | | Incandescent light source: $T_{int}$ = 402ms | Ch0 | | 144 | | | 129 | | |
| | | | Ch1 | | 72 | | | 67 | | |

| Symbol | Parameter | Test Conditions | Channel | TSL2560T, FN, & CL TSL2561T, FN & CL | | | TSL2560CS, TSL2561CS | | | Unit |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | | Min | Typ | Max | Min | Typ | Max | |
| | ADC count value ratio: Ch1/Ch0 | Fluorescent light source: $T_{int}$ = 402ms | | | 0.11 | | | 0.11 | | |
| | | Incandescent light source: $T_{int}$ = 402ms | | | 0.5 | | | 0.52 | | |
| $R_V$ | Illuminance responsivity, low gain mode (6) | Fluorescent light source: $T_{int}$ = 402ms | Ch0 | | 2.3 | | | 2.2 | | counts/lux |
| | | | Ch1 | | 0.25 | | | 0.24 | | |
| | | Incandescent light source: $T_{int}$ = 402ms | Ch0 | | 9 | | | 8.1 | | |
| | | | Ch1 | | 4.5 | | | 4.2 | | |
| | (Sensor Lux)/(actual Lux), high gain mode (7) | Fluorescent light source: $T_{int}$ = 402ms | | 0.65 | 1 | 1.35 | 0.65 | 1 | 1.35 | |
| | | Incandescent light source: $T_{int}$ = 402ms | | 0.60 | 1 | 1.40 | 0.60 | 1 | 1.40 | |

**Note(s):**

1. Optical measurements are made using small-angle incident radiation from light-emitting diode optical sources. Visible 640nm LEDs and infrared 940nm LEDs are used for final product testing for compatibility with high-volume production.
2. The 640nm irradiance $E_e$ is supplied by an AlInGaP light-emitting diode with the following characteristics: peak wavelength $\lambda_p$ = 640nm and spectral halfwidth $\Delta\lambda_{1/2}$ = 17nm.
3. The 940nm irradiance $E_e$ is supplied by a GaAs light-emitting diode with the following characteristics: peak wavelength $\lambda_p$ = 940nm and spectral halfwidth $\Delta\lambda_{1/2}$ = 40nm.
4. Integration time $T_{int}$, is dependent on internal oscillator frequency ($f_{osc}$) and on the integration field value in the timing register as described in the *Register Set* section. For nominal $f_{osc}$ = 735kHz, nominal $T_{int}$ = (number of clock cycles)/$f_{osc}$.
   Field value 00: $T_{int} = (11 \times 918)/f_{osc}$ = 13.7ms
   Field value 01: $T_{int} = (81 \times 918)/f_{osc}$ = 101ms
   Field value 10: $T_{int} = (322 \times 918)/f_{osc}$ = 402ms
   Scaling between integration times vary proportionally as follows: 11/322 = 0.034 (field value 00), 81/322 = 0.252 (field value 01), and 322/322 = 1 (field value 10).

5. Full scale ADC count value is limited by the fact that there is a maximum of one count per two oscillator frequency periods and also by a 2-count offset.
   Full scale ADC count value = ((number of clock cycles)/2 – 2)
   Field value 00: Full scale ADC count value = ((11 × 918)/2 – 2) = 5047
   Field value 01: Full scale ADC count value = ((81 × 918)/2 – 2) = 37177
   Field value 10: Full scale ADC count value = 65535, which is limited by 16-bit register. This full scale ADC count value is reached for 131074 clock cycles, which occurs for $T_{int}$ = 178ms for nominal $f_{osc}$ = 735kHz.
6. Low gain mode has 16× lower gain than high gain mode: (1/16 = 0.0625).
7. The sensor Lux is calculated using the empirical formula shown in Calculating Lux of this data sheet based on measured Ch0 and Ch1 ADC count values for the light source specified. Actual Lux is obtained with a commercial luxmeter. The range of the (sensor Lux) / (actual Lux) ratio is estimated based on the variation of the 640nm and 940nm optical parameters. Devices are not 100% tested with fluorescent or incandescent light sources.

**Figure 12:**
**AC Electrical Characteristics, $V_{DD}$ = 3V, $T_A$ = 25°C (unless otherwise noted)**

| Symbol | Parameter (1) | Test Conditions | Min | Typ | Max | Unit |
|---|---|---|---|---|---|---|
| $t_{(CONV)}$ | Conversion time | | 12 | 100 | 400 | ms |
| $f_{(SCL)}$ | Clock frequency ($I^2C$ only) | | 0 | | 400 | kHz |
| | Clock frequency (SMBus only) | | 10 | | 100 | kHz |
| $t_{(BUF)}$ | Bus free time between start and stop condition | | 1.3 | | | µs |
| $t_{(HDSTA)}$ | Hold time after (repeated) start condition. After this period, the first clock is generated. | | 0.6 | | | µs |
| $t_{(SUSTA)}$ | Repeated start condition setup time | | 0.6 | | | µs |
| $t_{(SUSTO)}$ | Stop condition setup time | | 0.6 | | | µs |
| $t_{(HDDAT)}$ | Data hold time | | 0 | | 0.9 | µs |
| $t_{(SUDAT)}$ | Data setup time | | 100 | | | ns |
| $t_{(LOW)}$ | SCL clock low period | | 1.3 | | | µs |
| $t_{(HIGH)}$ | SCL clock high period | | 0.6 | | | µs |
| $t_{(TIMEOUT)}$ | Detect clock/data low timeout (SMBus only) | | 25 | | 35 | ms |
| $t_F$ | Clock/data fall time | | | | 300 | ns |
| $t_R$ | Clock/data rise time | | | | 300 | ns |
| $C_i$ | Input pin capacitance | | | | 10 | pF |

**Note(s):**

1. Specified by design and characterization; not production tested.

**Figure 13:**
**Timing Diagrams**

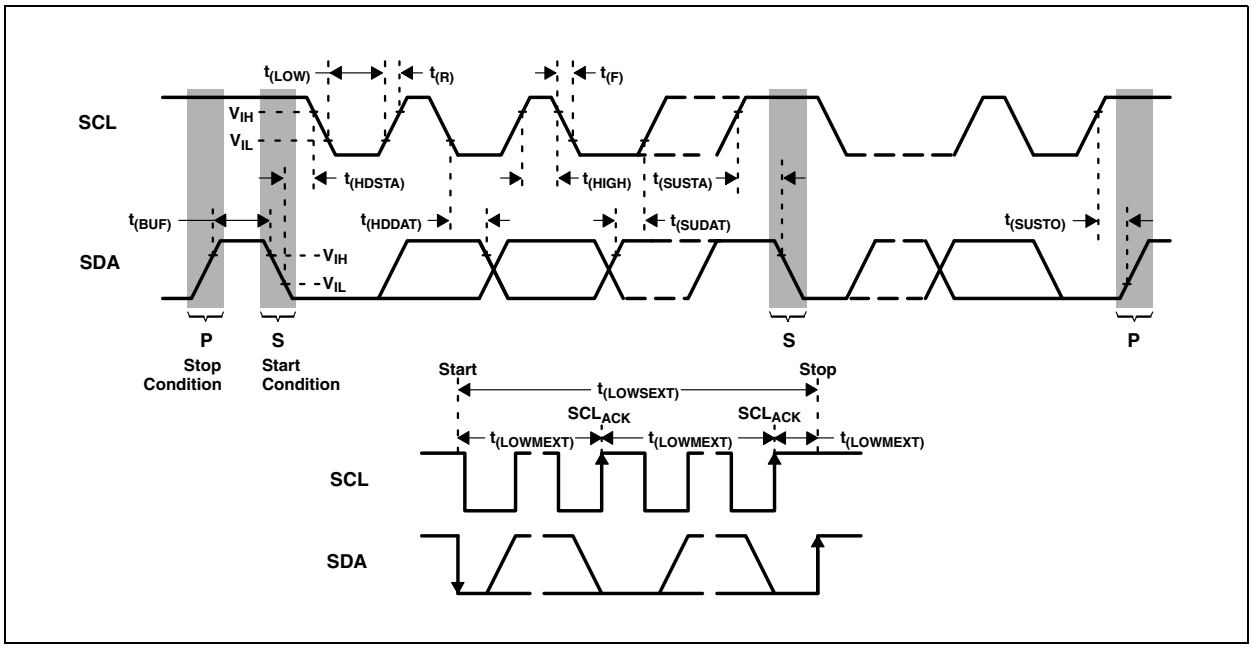



**Figure 14:**
**Example Timing Diagram for SMBus Send Byte Format**

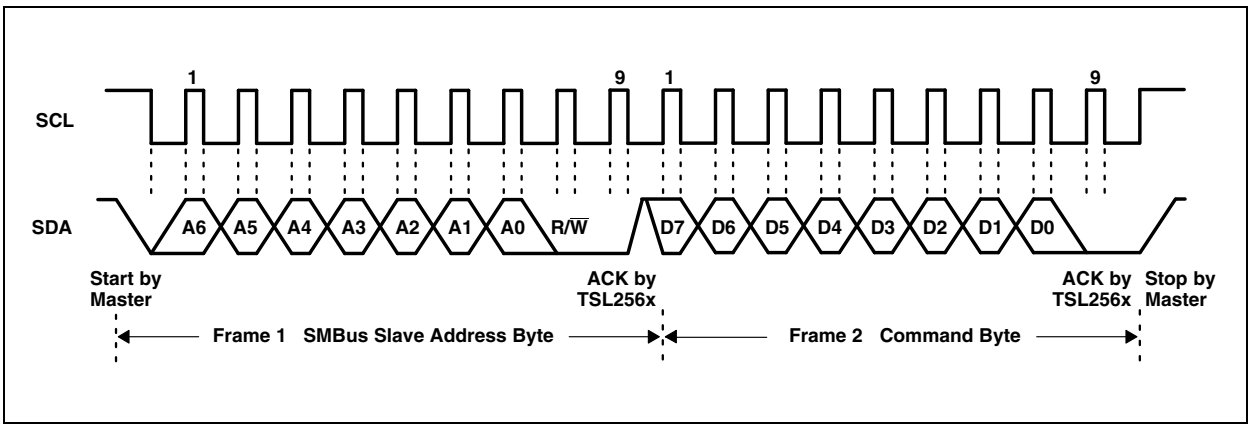
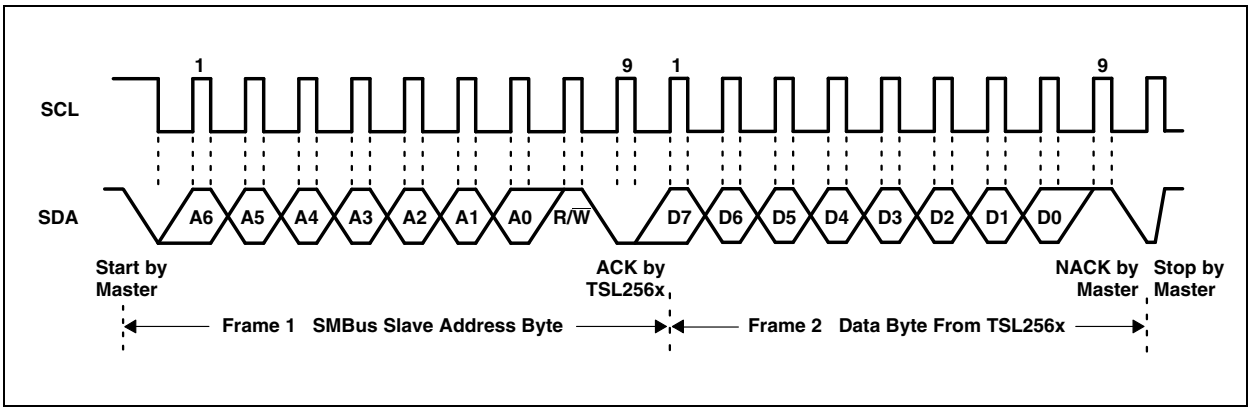



**Figure 15:**
**Example Timing Diagram for SMBus Receive Byte Format**



## Typical Characteristics

Figure 16:
Spectral Responsivity

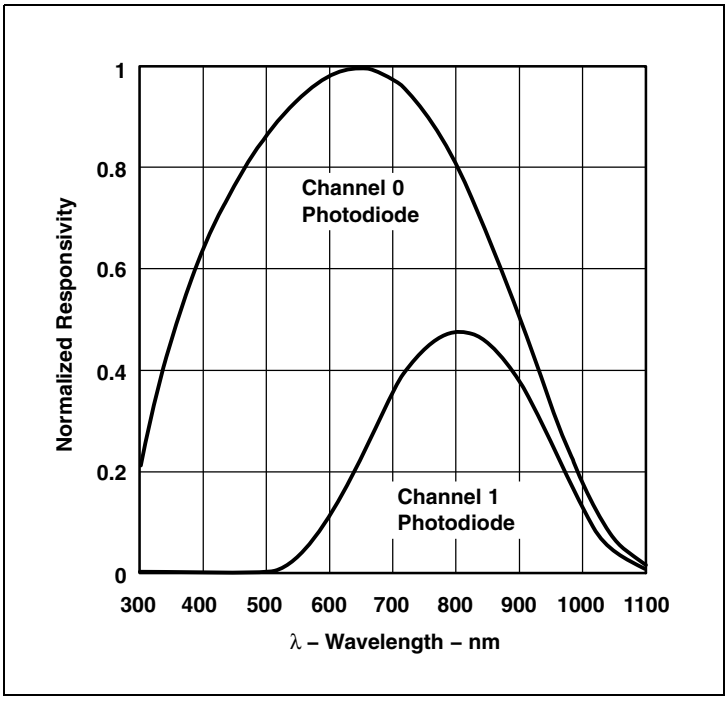



Figure 17:
Normalized Responsivity vs. Angular Displacement - CS Package

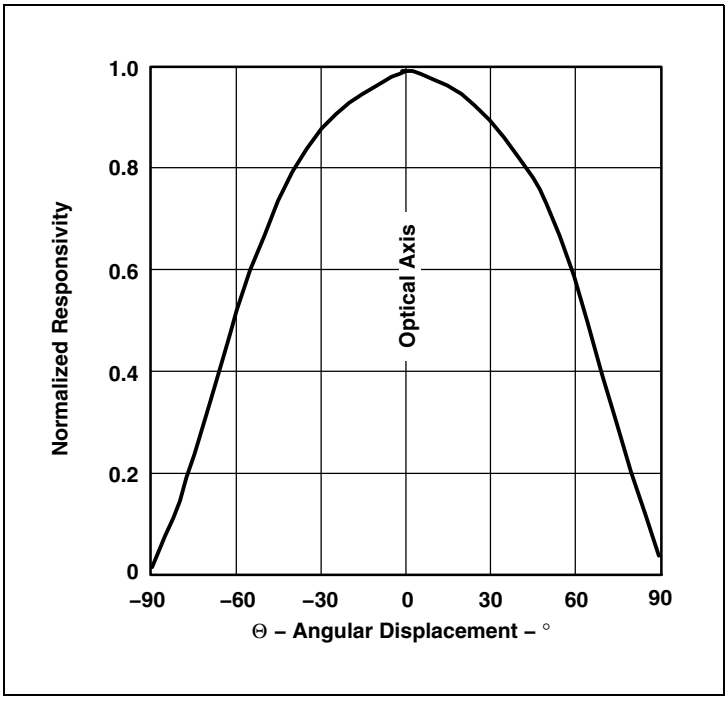


**Figure 18:**
**Normalized Responsivity vs. Angular Displacement - T Package**

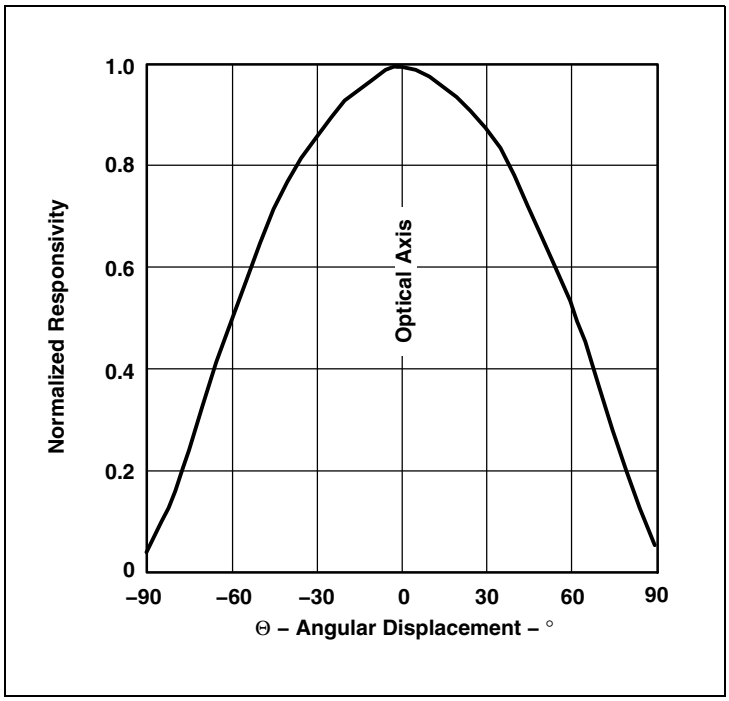



**Figure 19:**
**Normalized Responsivity vs. Angular Displacement - FN Package**

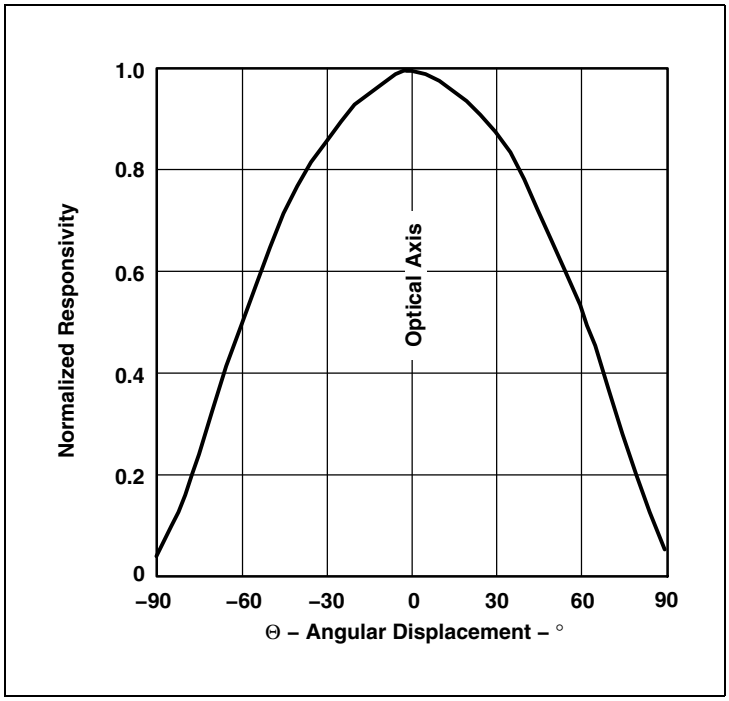
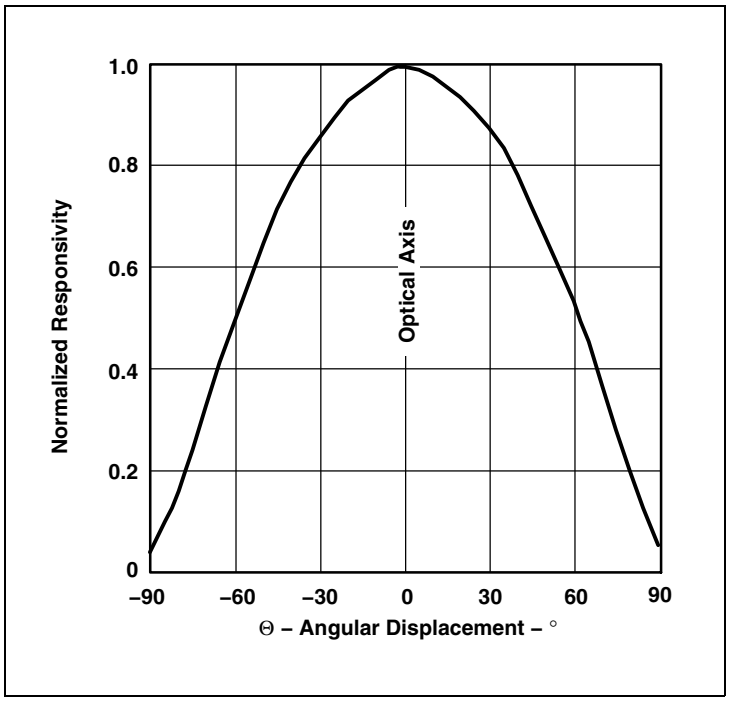


**Figure 20:**
**Normalized Responsivity vs. Angular Displacement - CL Package**



# Principles of Operation

## Analog-to-Digital Converter

The TSL256x contains two integrating analog-to-digital converters (ADC) that integrate the currents from the channel 0 and channel 1 photodiodes. Integration of both channels occurs simultaneously, and upon completion of the conversion cycle the conversion result is transferred to the channel 0 and channel 1 data registers, respectively. The transfers are double buffered to ensure that invalid data is not read during the transfer. After the transfer, the device automatically begins the next integration cycle.

## Digital Interface

Interface and control of the TSL256x is accomplished through a two-wire serial interface to a set of registers that provide access to device control functions and output data. The serial interface is compatible with System Management Bus (SMBus) versions 1.1 and 2.0, and $I^2C$ bus Fast-Mode. The TSL256x offers three slave addresses that are selectable via an external pin (ADDR SEL). The slave address options are shown in Figure 21.

**Figure 21:**
**Slave Address Selection**

| ADDR SEL Terminal Level | Slave Address | SMB Alert Address |
|---|---|---|
| GND | 0101001 | 0001100 |
| Float | 0111001 | 0001100 |
| $V_{DD}$ | 1001001 | 0001100 |

**Note(s):**

1. The Slave and SMB Alert Addresses are 7 bits. Please note the SMBus and $I^2C$ protocols (see SMBus and $I^2C$ Protocols). A read/write bit should be appended to the slave address by the master device to properly communicate with the TSL256X device.

## SMBus and I²C Protocols

Each *Send* and *Write* protocol is, essentially, a series of bytes. A byte sent to the TSL256x with the most significant bit (MSB) equal to 1 will be interpreted as a COMMAND byte. The lower four bits of the COMMAND byte form the register select address (see Figure 31), which is used to select the destination for the subsequent byte(s) received. The TSL256x responds to any Receive Byte requests with the contents of the register specified by the stored register select address.

The TSL256X implements the following protocols of the SMB 2.0 specification:

- Send Byte Protocol
- Receive Byte Protocol
- Write Byte Protocol
- Write Word Protocol
- Read Word Protocol
- Block Write Protocol
- Block Read Protocol

The TSL256X implements the following protocols of the Philips Semiconductor I²C specification:

- I²C Write Protocol
- I²C Read (Combined Format) Protocol

When an SMBus Block Write or Block Read is initiated (see description of Command Register), the byte following the COMMAND byte is ignored but is a requirement of the SMBus specification. This field contains the byte count (i.e. the number of bytes to be transferred). The TSL2560 (SMBus) device ignores this field and extracts this information by counting the actual number of bytes transferred before the Stop condition is detected.

When an I²C Write or I²C Read (Combined Format) is initiated, the byte count is also ignored but follows the SMBus protocol specification. Data bytes continue to be transferred from the TSL2561 (I²C) device to Master until a NACK is sent by the Master.

The data formats supported by the TSL2560 and TSL2561 devices are:

- Master transmitter transmits to slave receiver (SMBus and I²C):
  - The transfer direction in this case is not changed.
- Master reads slave immediately after the first byte (SMBus only):
  - At the moment of the first acknowledgment (provided by the slave receiver) the master transmitter becomes a master receiver and the slave receiver becomes a slave transmitter.

- Combined format (SMBus and I²C):
  - During a change of direction within a transfer, the master repeats both a START condition and the slave address but with the R/W bit reversed. In this case, the master receiver terminates the transfer by generating a NACK on the last byte of the transfer and a STOP condition.

For a complete description of SMBus protocols, please review the SMBus Specification at http://www.smbus.org/specs.
For a complete description of I²C protocols, please review the I²C Specification at http://www.nxp.com.

**Figure 22:**
**SMBus and I²C Packet Protocol Element Key**

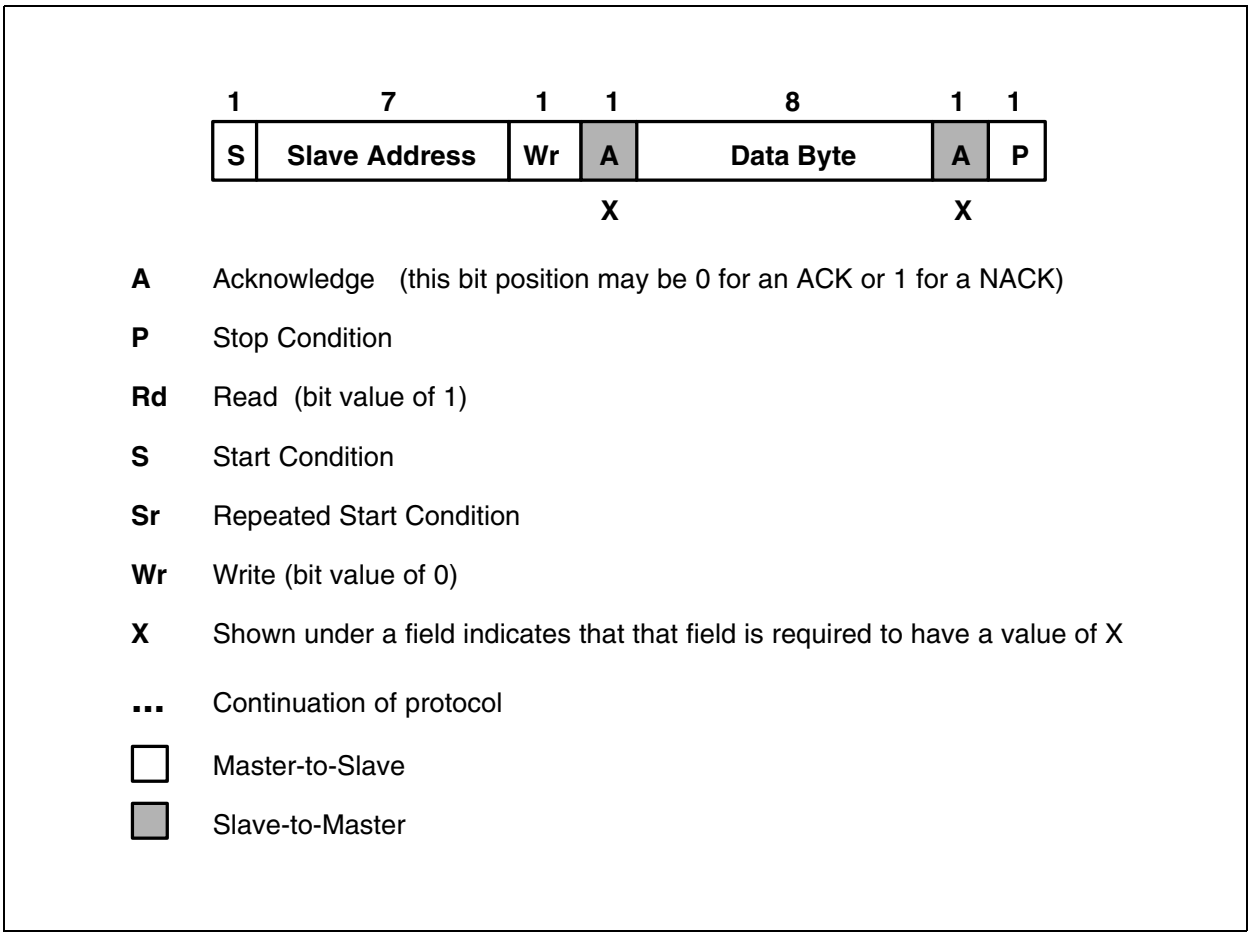


**Figure 23:**
**SMBus Send Byte Protocol**

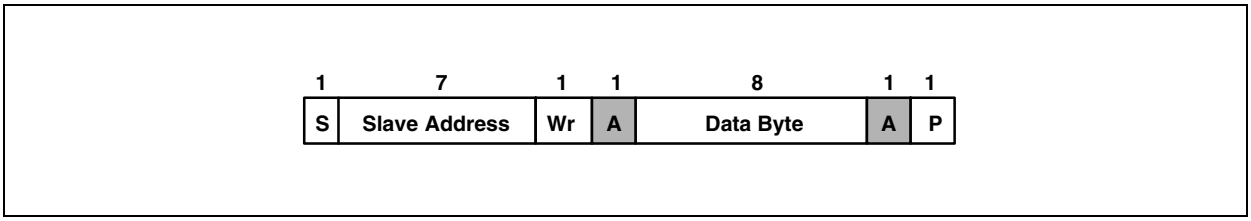



**Figure 24:**
**SMBus Receive Byte Protocol**

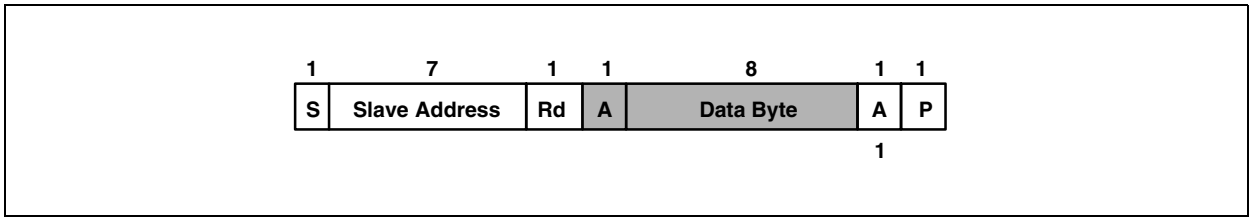



**Figure 25:**
**SMBus Write Byte Protocol**

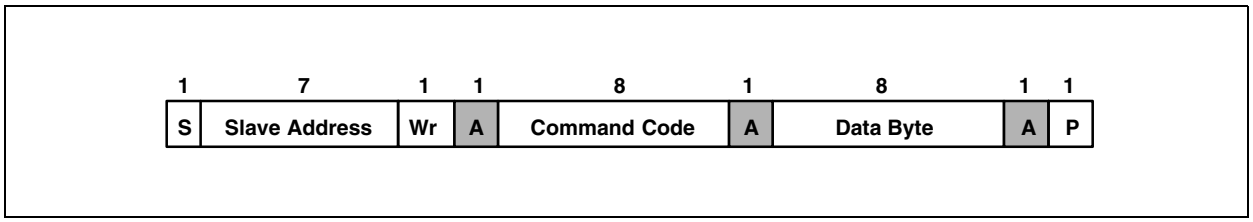



**Figure 26:**
**SMBus Read Byte Protocol**

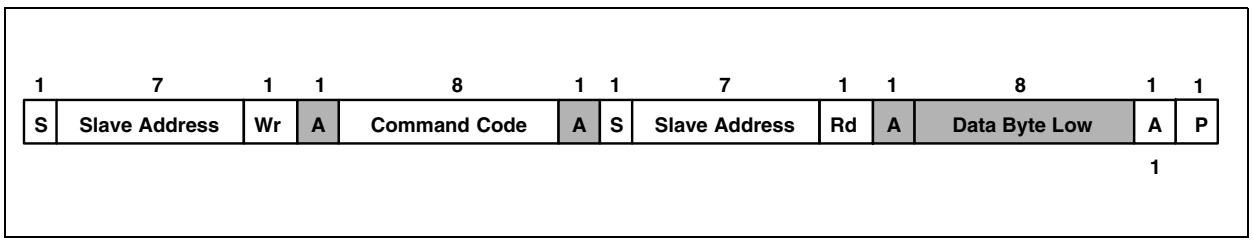



**Figure 27:**
**SMBus Write Word Protocol**

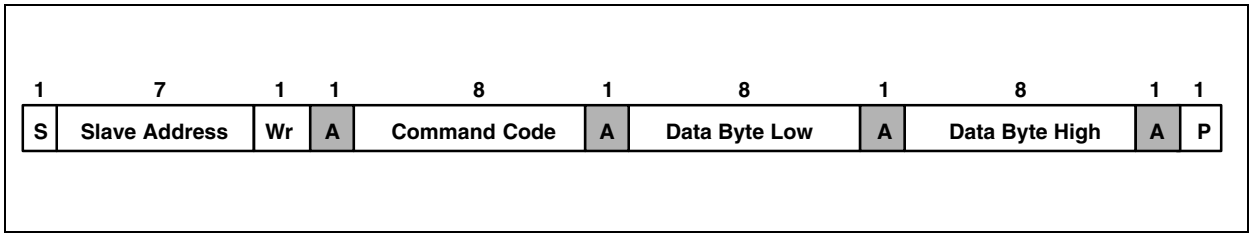


**Figure 28:**
**SMBus Read Word Protocol**

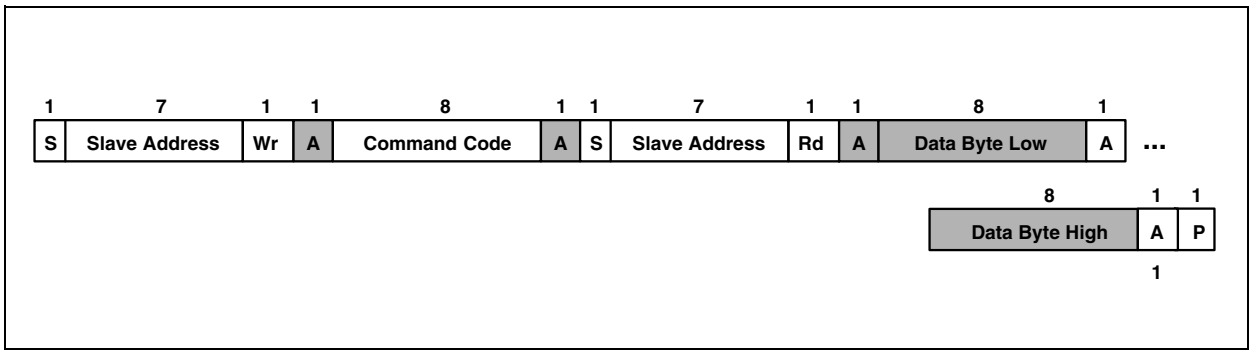
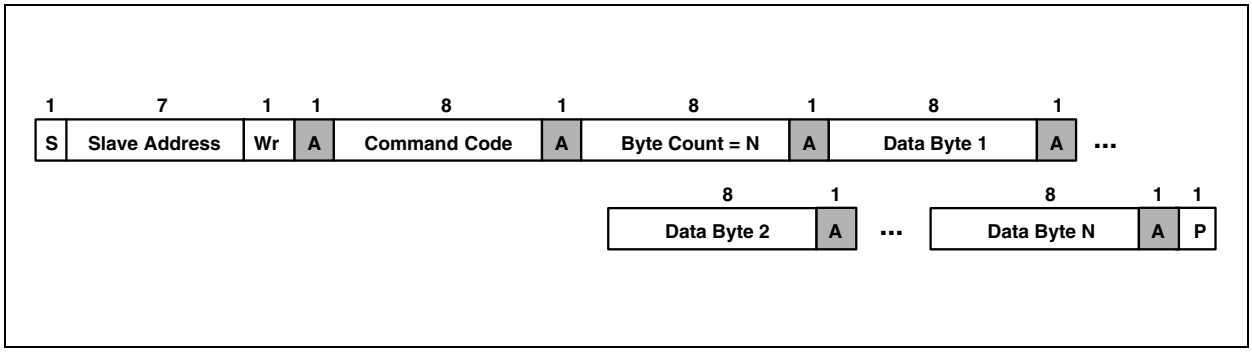


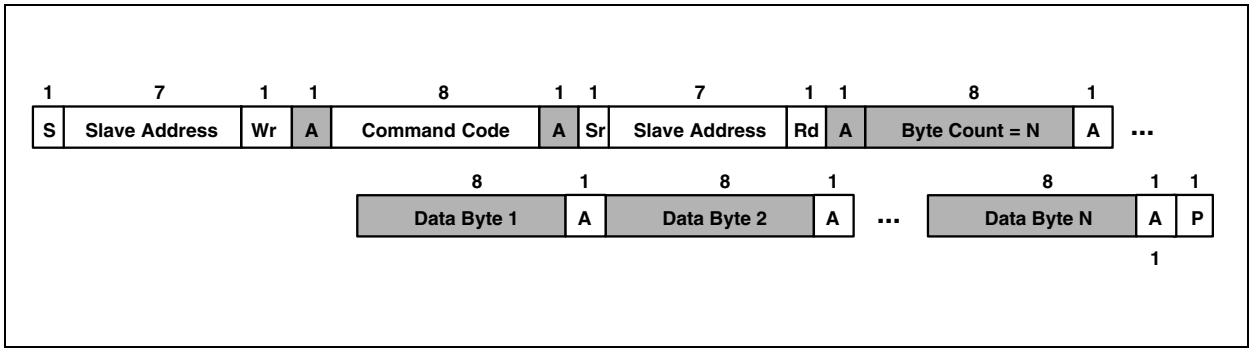

**Figure 29:**
**SMBus Block Write or I²C Write Protocols**




**Note(s):**

1. The I²C write protocol does not use the Byte Count packet, and the Master will continue sending Data Bytes until the Master initiates a Stop Condition. See the Command Register for additional information regarding the Block Read/Write protocol.

**Figure 30:**
**SMBus Block Read or I²C Read (Combined Format) Protocols**




**Note(s):**

1. The I²C read protocol does not use the Byte Count packet, and the Master will continue receiving Data Bytes until the Master initiates a Stop Condition. See the Command Register for additional information regarding the Block Read/Write protocol.

## Register Set

The TSL256x is controlled and monitored by sixteen registers (three are reserved) and a Command Register accessed through the serial interface. These registers provide for a variety of control functions and can be read to determine results of the ADC conversions. The Register Set is summarized in Figure 31.

**Figure 31:**
**Register Address**

| Address | Register Name | Register Function |
|---|---|---|
| -- | COMMAND | Specifies register address |
| 0h | CONTROL | Control of basic functions |
| 1h | TIMING | Integration time/gain control |
| 2h | THRESHLOWLOW | Low byte of low interrupt threshold |
| 3h | THRESHLOWHIGH | High byte of low interrupt threshold |
| 4h | THRESHHIGHLOW | Low byte of high interrupt threshold |
| 5h | THRESHHIGHHIGH | High byte of high interrupt threshold |
| 6h | INTERRUPT | Interrupt control |
| 7h | -- | Reserved |
| 8h | CRC | Factory test - not a user register |
| 9h | -- | Reserved |
| Ah | ID | Part number/Rev ID |
| Bh | -- | Reserved |
| Ch | DATA0LOW | Low byte of ADC channel 0 |
| Dh | DATA0HIGH | High byte of ADC channel 0 |
| Eh | DATA1LOW | Low byte of ADC channel 1 |
| Fh | DATA1HIGH | High byte of ADC channel 1 |

The mechanics of accessing a specific register depends on the specific SMB protocol used. Refer to the section on SMBus protocols. In general, the Command Register is written first to specify the specific control/status register for following read/write operations.

## Command Register

The Command Register specifies the address of the target register for subsequent read and write operations. The Send Byte protocol is used to configure the Command Register. The Command Register contains eight bits as described in Figure 32. The Command Register defaults to 00h at power on.

**Figure 32:**
**Command Register**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| CMD | CLEAR | WORD | BLOCK | ADDRESS | | | |

| Field | Bit | Description |
|---|---|---|
| CMD | 7 | Select Command Register. Must write as 1. |
| CLEAR | 6 | Interrupt clear. Clears any pending interrupt. This bit is a write-one-to-clear bit. It is self clearing. |
| WORD | 5 | SMB Write/Read Word Protocol. 1 indicates that this SMB transaction is using either the SMB Write Word or Read Word protocol. |
| BLOCK | 4 | Block Write/Read Protocol. 1 indicates that this transaction is using either the Block Write or the Block Read protocol. (1) |
| ADDRESS | 3:0 | Register Address. This field selects the specific control or status register for following write and read commands according to Figure 31. |

**Note(s):**

1. An $I^2C$ block transaction will continue until the Master sends a stop condition. See Figure 29 and Figure 30. Unlike the $I^2C$ protocol, the SMBus read/write protocol requires a Byte Count. All four ADC Channel Data Registers (Ch through Fh) can be read simultaneously in a single SMBus transaction. This is the only 32-bit data block supported by the TSL2560 SMBus protocol. The BLOCK bit must be set to 1, and a read condition should be initiated with a COMMAND CODE of 9Bh. By using a COMMAND CODE of 9Bh during an SMBus Block Read Protocol, the TSL2560 device will automatically insert the appropriate Byte Count (Byte Count = 4) as illustrated in Figure 30. A write condition should not be used in conjunction with the Bh register.

## Control Register (0h)

The Control Register contains two bits and is primarily used to power the TSL256x device up and down as shown in Figure 33.

**Figure 33:**
**Control Register**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Resv | Resv | Resv | Resv | Resv | Resv | POWER | |

| Field | Bit | Description |
|---|---|---|
| Resv | 7:2 | Reserved. Write as 0. |
| POWER | 1:0 | Power up/power down. By writing a 03h to this register, the device is powered up. By writing a 00h to this register, the device is powered down.<br>**Note:** If a value of 03h is written, the value returned during a read cycle will be 03h. This feature can be used to verify that the device is communicating properly. |

## Timing Register (1h)

The Timing Register controls both the integration time and the gain of the ADC channels. A common set of control bits is provided that controls both ADC channels. The Timing Register defaults to 02h at power on.

**Figure 34:**
**Timing Register**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Resv | Resv | Resv | GAIN | Manual | Resv | INTEG | |

| Field | Bit | Description |
|---|---|---|
| Resv | 7:5 | Reserved. Write as 0. |
| GAIN | 4 | Switches gain between low gain and high gain modes. Writing a 0 selects low gain (1×); writing a 1 selects high gain (16×). |
| Manual | 3 | Manual timing control. Writing a 1 begins an integration cycle. Writing a 0 stops an integration cycle.<br>**Note:** This field only has meaning when INTEG = 11. It is ignored at all other times. |
| Resv | 2 | Reserved. Write as 0. |
| INTEG | 1:0 | Integrate time. This field selects the integration time for each conversion. |

Integration time is dependent on the INTEG FIELD VALUE and the internal clock frequency. Nominal integration times and respective scaling between integration times scale proportionally as shown in Figure 35. See Note 4 and Note 5 for detailed information regarding how the scale values were obtained; see Calculating Lux and Simplified Lux Calculation for further information on how to calculate lux.

**Figure 35:**
**Integration Time**

| INTEG Field Value | Scale | Nominal Integration Time |
|---|---|---|
| 00 | 0.034 | 13.7ms |
| 01 | 0.252 | 101ms |
| 10 | 1 | 402ms |
| 11 | -- | N/A |

The manual timing control feature is used to manually start and stop the integration time period. If a particular integration time period is required that is not listed in Figure 35, then this feature can be used. For example, the manual timing control can be used to synchronize the TSL256x device with an external light source (e.g. LED). A start command to begin integration can be initiated by writing a 1 to this bit field. Correspondingly, the integration can be stopped by simply writing a 0 to the same bit field.

## Interrupt Threshold Register (2h – 5h)

The Interrupt Threshold registers store the values to be used as the high and low trigger points for the comparison function for interrupt generation. If the value generated by channel 0 crosses below or is equal to the low threshold specified, an interrupt is asserted on the interrupt pin. If the value generated by channel 0 crosses above the high threshold specified, an interrupt is asserted on the interrupt pin. Registers THRESHLOWLOW and THRESHLOWHIGH provide the low byte and high byte, respectively, of the lower interrupt threshold. Registers THRESHHIGHLOW and THRESHHIGHHIGH provide the low and high bytes, respectively, of the upper interrupt threshold. The high and low bytes from each set of registers are combined to form a 16-bit threshold value. The interrupt threshold registers default to 00h on power up.

**Figure 36:**
**Interrupt Threshold Register**

| Register | Address | Bits | Description |
|---|---|---|---|
| THRESHLOWLOW | 2h | 7:0 | ADC channel 0 lower byte of the low threshold |
| THRESHLOWHIGH | 3h | 7:0 | ADC channel 0 upper byte of the low threshold |
| THRESHHIGHLOW | 4h | 7:0 | ADC channel 0 lower byte of the high threshold |
| THRESHHIGHHIGH | 5h | 7:0 | ADC channel 0 upper byte of the high threshold |

**Note(s):**

1. Since two 8-bit values are combined for a single 16-bit value for each of the high and low interrupt thresholds, the Send Byte protocol should not be used to write to these registers. Any values transferred by the Send Byte protocol with the MSB set would be interpreted as the COMMAND field and stored as an address for subsequent read/write operations and not as the interrupt threshold information as desired. The Write Word protocol should be used to write byte-paired registers. For example, the THRESHLOWLOW and THRESHLOWHIGH registers (as well as the THRESHHIGHLOW and THRESHHIGHHIGH registers) can be written together to set the 16-bit ADC value in a single transaction.

## Interrupt Control Register (6h)

The Interrupt Register controls the extensive interrupt capabilities of the TSL256x. The TSL256x permits both SMB-Alert style interrupts as well as traditional level-style interrupts. The interrupt persist bit field (PERSIST) provides control over when interrupts occur. A value of 0 causes an interrupt to occur after every integration cycle regardless of the threshold settings. A value of 1 results in an interrupt after one integration time period outside the threshold window. A value of *N* (where *N* is 2 through15) results in an interrupt only if the value remains outside the threshold window for *N* consecutive integration cycles. For example, if *N* is equal to 10 and the integration time is 402ms, then the total time is approximately 4 seconds.

When a level Interrupt is selected, an interrupt is generated whenever the last conversion results in a value outside of the programmed threshold window. The interrupt is active-low and remains asserted until cleared by writing the Command Register with the CLEAR bit set.

In SMBAlert mode, the interrupt is similar to the traditional level style and the interrupt line is asserted low. To clear the interrupt, the host responds to the SMBAlert by performing a modified Receive Byte operation, in which the Alert Response Address (ARA) is placed in the slave address field, and the TSL256x that generated the interrupt responds by returning its own address in the seven most significant bits of the receive data byte. If more than one device connected on the bus has pulled the SMBAlert line low, the highest priority (lowest address) device will win communication rights via standard arbitration during the slave address transfer. If the device loses this arbitration, the interrupt will not be cleared. The Alert Response Address is 0Ch.

When INTR = 11, the interrupt is generated immediately following the SMBus write operation. Operation then behaves in an SMBAlert mode, and the *software set* interrupt may be cleared by an SMBAlert cycle.

**Note(s):** Interrupts are based on the value of Channel 0 only.

**Figure 37:**
**Interrupt Control Register**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Resv | Resv | INTR | | PERSIST | | | |

| Field | Bits | Description |
|---|---|---|
| Resv | 7:6 | Reserved. Write as 0. |
| INTR | 5:4 | INTR Control Select. This field determines mode of interrupt logic according to Figure 38, below. |
| PERSIST | 3:0 | Interrupt persistence. Controls rate of interrupts to the host processor as shown in Figure 39, below. |

**Figure 38:**
**Interrupt Control Select**

| INTR Field Value | Read Value |
|---|---|
| 00 | Interrupt output disabled |
| 01 | Level Interrupt |
| 10 | SMBAlert compliant |
| 11 | Test Mode: Sets interrupt and functions as mode 10 |

**Note(s):**

1. Field value of 11 may be used to test interrupt connectivity in a system or to assist in debugging interrupt service routine software.

**Figure 39:**
**Interrupt Persistence Select**

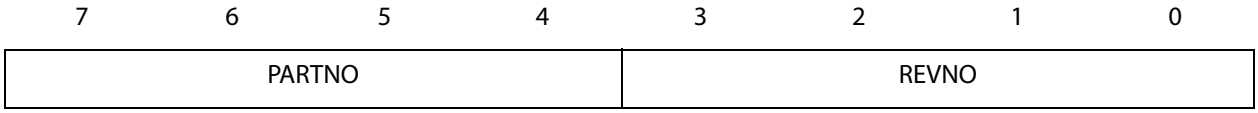
| Persist Field Value | Interrupt Persist Function |
|---|---|
| 0000 | Every ADC cycle generates interrupt |
| 0001 | Any value outside of threshold range |
| 0010 | 2 integration time periods out of range |
| 0011 | 3 integration time periods out of range |
| 0100 | 4 integration time periods out of range |

| Persist Field Value | Interrupt Persist Function |
|---|---|
| 0101 | 5 integration time periods out of range |
| 0110 | 6 integration time periods out of range |
| 0111 | 7 integration time periods out of range |
| 1000 | 8 integration time periods out of range |
| 1001 | 9 integration time periods out of range |
| 1010 | 10 integration time periods out of range |
| 1011 | 11 integration time periods out of range |
| 1100 | 12 integration time periods out of range |
| 1101 | 13 integration time periods out of range |
| 1110 | 14 integration time periods out of range |
| 1111 | 15 integration time periods out of range |

## ID Register (Ah)

The ID Register provides the value for both the part number and silicon revision number for that part number. It is a read-only register, whose value never changes.

**Figure 40:**
**ID Register**




| Field | Bits | Description | |
|---|---|---|---|
| PARTNO | 7:4 | Part Number Identification: | |
| | | **Field Value** | **Device Number** |
| | | 0000 | TSL2560CS |
| | | 0001 | TSL2561CS |
| | | 0100 | TSL2560T/FN/CL |
| | | 0101 | TSL2561T/FN/CL |
| REVNO | 3:0 | Revision number identification | |

## ADC Channel Data Registers (Ch – Fh)

The ADC channel data are expressed as 16-bit values spread across two registers. The ADC channel 0 data registers, DATA0LOW and DATA0HIGH provide the lower and upper bytes, respectively, of the ADC value of channel 0. Registers DATA1LOW and DATA1HIGH provide the lower and upper bytes, respectively, of the ADC value of channel 1. All channel data registers are read-only and default to 00h on power up.

**Figure 41:**
**ADC Channel Data Registers**

| Register | Address | Bits | Description |
|---|---|---|---|
| DATA0LOW | Ch | 7:0 | ADC channel 0 lower byte |
| DATA0HIGH | Dh | 7:0 | ADC channel 0 upper byte |
| DATA1LOW | Eh | 7:0 | ADC channel 1 lower byte |
| DATA1HIGH | Fh | 7:0 | ADC channel 1 upper byte |

The upper byte data registers can only be read following a read to the corresponding lower byte register. When the lower byte register is read, the upper eight bits are strobed into a shadow register, which is read by a subsequent read to the upper byte. The upper register will read the correct value even if additional ADC integration cycles end between the reading of the lower and upper registers.

**Note(s):** The Read Word protocol can be used to read byte-paired registers. For example, the DATA0LOW and DATA0HIGH registers (as well as the DATA1LOW and DATA1HIGH registers) may be read together to obtain the 16-bit ADC value in a single transaction.

# Application Information: Software

## Basic Operation

After applying $V_{DD}$, the device will initially be in the power-down state. To operate the device, issue a command to access the Control Register followed by the data value 03h to power up the device. At this point, both ADC channels will begin a conversion at the default integration time of 400ms. After 400ms, the conversion results will be available in the DATA0 and DATA1 registers. Use the following pseudo code to read the data registers:

```
// Read ADC Channels Using Read Word Protocol – RECOMMENDED
        Address = 0x39                                  //Slave addr – also 0x29 or 0x49

        //Address the Ch0 lower data register and configure for Read Word
        Command = 0xAC                                  //Set Command bit and Word bit

        //Reads two bytes from sequential registers 0x0C and 0x0D
        //Results are returned in DataLow and DataHigh variables
        ReadWord (Address, Command, DataLow, DataHigh)
        Channel0 = 256 * DataHigh + DataLow

        //Address the Ch1 lower data register and configure for Read Word
        Command = 0xAE                                  //Set bit fields 7 and 5

        //Reads two bytes from sequential registers 0x0E and 0x0F
        //Results are returned in DataLow and DataHigh variables
        ReadWord (Address, Command, DataLow, DataHigh)
        Channel1 = 256 * DataHigh + DataLow             //Shift DataHigh to upper byte

// Read ADC Channels Using Read Byte Protocol
        Address = 0x39                                  //Slave addr – also 0x29 or 0x49
        Command = 0x8C                                  //Address the Ch0 lower data register
        ReadByte (Address, Command, DataLow)            //Result returned in DataLow
        Command = 0x8D                                  //Address the Ch0 upper data register
        ReadByte (Address, Command, DataHigh)           //Result returned in DataHigh
        Channel0 = 256 * DataHigh + DataLow             //Shift DataHigh to upper byte

        Command = 0x8E                                  //Address the Ch1 lower data register
        ReadByte (Address, Command, DataLow             //Result returned in DataLow
        Command = 0x8F                                  //Address the Ch1 upper data register
        ReadByte (Address, Command, DataHigh)           //Result returned in DataHigh
        Channel1 = 256 * DataHigh + DataLow             //Shift DataHigh to upper byte
```

## Configuring the Timing Register

The command, timing, and control registers are initialized to default values on power up. Setting these registers to the desired values would be part of a normal initialization or setup procedure. In addition, to maximize the performance of the device under various conditions, the integration time and gain may be changed often during operation. The following pseudo code illustrates a procedure for setting up the Timing Register for various options:

```
// Set up Timing Register
        //Low Gain (1x), integration time of 402ms (default value)
        Address = 0x39
        Command = 0x81
        Data = 0x02
        WriteByte(Address, Command, Data)

        //Low Gain (1x), integration time of 101ms
        Data = 0x01
        WriteByte(Address, Command, Data)

        //Low Gain (1x), integration time of 13.7ms
        Data = 0x00
        WriteByte(Address, Command, Data)

        //High Gain (16x), integration time of 101ms
        Data = 0x11
        WriteByte(Address, Command, Data)

//Read data registers (see Basic Operation example)

//Perform Manual Integration
        //Set up for manual integration with Gain of 1x
        Data = 0x03
        //Set manual integration mode – device stops converting
        WriteByte(Address, Command, Data)

        //Begin integration period
        Data = 0x0B
        WriteByte(Address, Command, Data)

        //Integrate for 50ms
        Sleep (50)                                          //Wait for 50ms

        //Stop integrating
        Data = 0x03
        WriteByte(Address, Command, Data)

//Read data registers (see Basic Operation example)
```

## Interrupts

The interrupt feature of the TSL256x device simplifies and improves system efficiency by eliminating the need to poll the sensor for a light intensity value. Interrupt styles are determined by the INTR field in the Interrupt Register. The interrupt feature may be disabled by writing a field value of 00h to the Interrupt Control Register so that polling can be performed.

The versatility of the interrupt feature provides many options for interrupt configuration and usage. The primary purpose of the interrupt function is to provide a *meaningful change* in light intensity. However, it also be used as an end-of-conversion signal. The concept of a *meaningful* change can be defined by the user both in terms of light intensity and time, or persistence, of that change in intensity. The TSL256x device implements two 16-bit-wide interrupt threshold registers that allow the user to define a threshold above and below the current light level. An interrupt will then be generated when the value of a conversion exceeds either of these limits. For simplicity of programming, the threshold comparison is accomplished only with Channel 0. This simplifies calculation of thresholds that are based, for example, on a percent of the current light level. It is adequate to use only one channel when calculating light intensity differences since, for a given light source, the channel 0 and channel 1 values are linearly proportional to each other and thus both values scale linearly with light intensity.

To further control when an interrupt occurs, the TSL256x device provides an interrupt persistence feature. This feature allows the user to specify a number of conversion cycles for which a light intensity exceeding either interrupt threshold must persist before actually generating an interrupt. This can be used to prevent transient changes in light intensity from generating an unwanted interrupt. With a value of 1, an interrupt occurs immediately whenever either threshold is exceeded. With values of *N*, where *N* can range from 2 to 15, *N* consecutive conversions must result in values outside the interrupt window for an interrupt to be generated. For example, if *N* is equal to 10 and the integration time is 402ms, then an interrupt will not be generated unless the light level persists for more than 4 seconds outside the threshold.

Two different interrupt styles are available: Level and SMBus Alert. The difference between these two interrupt styles is how they are cleared. Both result in the interrupt line going active low and remaining low until the interrupt is cleared. A level style interrupt is cleared by setting the CLEAR bit (bit 6) in the Command Register. The SMBus Alert style interrupt is cleared by an Alert Response as described in the Interrupt Control Register section and SMBus specification.

To configure the interrupt as an end-of-conversion signal, the interrupt PERSIST field is set to 0. Either Level or SMBus Alert style can be used. An interrupt will be generated upon

completion of each conversion. The interrupt threshold registers are ignored. The following example illustrates the configuration of a level interrupt:

```
// Set up end–of–conversion interrupt, Level style
        Address = 0x39                          //Slave addr also 0x29 or 0x49
        Command = 0x86                          //Address Interrupt Register
        Data = 0x10                             //Level style, every ADC cycle
        WriteByte(Address, Command, Data)
```

The following example pseudo code illustrates the configuration of an SMB Alert style interrupt when the light intensity changes 20% from the current value, and persists for 3 conversion cycles:

```
// Read current light level
        Address = 0x39                          //Slave addr also 0x29 or 0x49
        Command = 0xAC                          //Set Command bit and Word bit
        ReadWord (Address, Command, DataLow, DataHigh)
        Channel0 = (256 * DataHigh) + DataLow

        //Calculate upper and lower thresholds
        T_Upper = Channel0 + (0.2 * Channel0)
        T_Lower = Channel0 – (0.2 * Channel0)

        //Write the lower threshold register
        Command = 0xA2                          //Addr lower threshold reg, set Word Bit
        WriteWord (Address, Command, T_Lower.LoByte, T_Lower.HiByte)

        //Write the upper threshold register
        Command = 0xA4                          //Addr upper threshold reg, set Word bit
        WriteWord (Address, Command, T_Upper.LoByte, T_Upper.HiByte)

        //Enable interrupt
        Command = 0x86                          //Address interrupt register
        Data = 0x23                             //SMBAlert style, PERSIST = 3
        WriteByte(Address, Command, Data)
```

In order to generate an interrupt on demand during system test or debug, a test mode (INTR = 11) can be used. The following example illustrates how to generate an interrupt on demand:

```
// Generate an interrupt
        Address = 0x39                          //Slave addr also 0x29 or 0x49
        Command = 0x86                          //Address Interrupt register
        Data = 0x30                             //Test interrupt
        WriteByte(Address, Command, Data)

        //Interrupt line should now be low
```

## Calculating Lux

The TSL256x is intended for use in ambient light detection applications such as display backlight control, where adjustments are made to display brightness or contrast based on the brightness of the ambient light, as perceived by the human eye. Conventional silicon detectors respond strongly to infrared light, which the human eye does not see. This can lead to significant error when the infrared content of the ambient light is high, such as with incandescent lighting, due to the difference between the silicon detector response and the brightness perceived by the human eye.

This problem is overcome in the TSL256x through the use of two photodiodes. One of the photodiodes (channel 0) is sensitive to both visible and infrared light, while the second photodiode (channel 1) is sensitive primarily to infrared light. An integrating ADC converts the photodiode currents to digital outputs. Channel 1 digital output is used to compensate for the effect of the infrared component of light on the channel 0 digital output. The ADC digital outputs from the two channels are used in a formula to obtain a value that approximates the human eye response in the commonly used Illuminance unit of Lux:

### *CS Package*

For $0 < CH1/CH0 \leq 0.52$

$Lux = 0.0315 \times CH0 - 0.0593 \times CH0 \times ((CH1/CH0)^{1.4})$

For $0.52 < CH1/CH0 \leq 0.65$

$Lux = 0.0229 \times CH0 - 0.0291 \times CH1$

For $0.65 < CH1/CH0 \leq 0.80$

$Lux = 0.0157 \times CH0 - 0.0180 \times CH1$

For $0.80 < CH1/CH0 \leq 1.30$

$Lux = 0.00338 \times CH0 - 0.00260 \times CH1$

For $CH1/CH0 > 1.30$

$Lux = 0$

### *T, FN, and CL Package*

For $0 < CH1/CH0 \leq 0.50$

$Lux = 0.0304 \times CH0 - 0.062 \times CH0 \times ((CH1/CH0)^{1.4})$

For $0.50 < CH1/CH0 \leq 0.61$

$Lux = 0.0224 \times CH0 - 0.031 \times CH1$

For $0.61 < CH1/CH0 \leq 0.80$

$Lux = 0.0128 \times CH0 - 0.0153 \times CH1$

For $0.80 < CH1/CH0 \leq 1.30$

$Lux = 0.00146 \times CH0 - 0.00112 \times CH1$

For $CH1/CH0 > 1.30$

$Lux = 0$

The formulas shown above were obtained by optical testing with fluorescent and incandescent light sources, and apply only to open-air applications. Optical apertures (e.g. light pipes) will affect the incident light on the device.

## Simplified Lux Calculation

Below is the argument and return value including source code (shown on following page) for calculating lux. The source code is intended for embedded and/or microcontroller applications. Two individual code sets are provided, one for the T, FN, and CL packages, and one for the CS package. All floating point arithmetic operations have been eliminated since embedded controllers and microcontrollers generally do not support these types of operations. Since floating point has been removed, scaling must be performed prior to calculating illuminance if the integration time is not 402ms and/or if the gain is not 16× as denoted in the source code on the following pages. This sequence scales first to mitigate rounding errors induced by decimal math.

```
extern unsigned int CalculateLux(unsigned int iGain, unsigned int tInt, unsigned int ch0, unsigned int ch1, int iType)

//****************************************************************************** //
// Copyright © ams AG, Inc.
//
// THIS CODE AND INFORMATION IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY AND/OR FITNESS FOR A
PARTICULAR PURPOSE.
//
// Module Name:
//          lux.cpp
//
//******************************************************************************

#define LUX_SCALE          14        // scale by 2^14
#define RATIO_SCALE        9         // scale ratio by 2^9

//--------------------------------------------------------------
// Integration time scaling factors
//--------------------------------------------------------------

#define CH_SCALE           10 // scale channel values by 2^10
#define CHSCALE_TINT0 0x7517 // 322/11 * 2^CH_SCALE
#define CHSCALE_TINT1 0x0fe7 // 322/81 * 2^CH_SCALE

//--------------------------------------------------------------
// T, FN, and CL Package coefficients
//--------------------------------------------------------------
// For Ch1/Ch0=0.00 to 0.50
//          Lux/Ch0=0.0304–0.062*((Ch1/Ch0)^1.4)
```

```
//          piecewise approximation
//                    For Ch1/Ch0=0.00 to 0.125:
//                              Lux/Ch0=0.0304–0.0272*(Ch1/Ch0)
//
//                    For Ch1/Ch0=0.125 to 0.250:
//                              Lux/Ch0=0.0325–0.0440*(Ch1/Ch0)
//
//                    For Ch1/Ch0=0.250 to 0.375:
//                              Lux/Ch0=0.0351–0.0544*(Ch1/Ch0)
//
//                    For Ch1/Ch0=0.375 to 0.50:
//                              Lux/Ch0=0.0381–0.0624*(Ch1/Ch0)
//
// For Ch1/Ch0=0.50 to 0.61:
//          Lux/Ch0=0.0224–0.031*(Ch1/Ch0)
//
// For Ch1/Ch0=0.61 to 0.80:
//          Lux/Ch0=0.0128–0.0153*(Ch1/Ch0)
//
// For Ch1/Ch0=0.80 to 1.30:
//          Lux/Ch0=0.00146–0.00112*(Ch1/Ch0)
//
// For Ch1/Ch0>1.3:
//          Lux/Ch0=0
//–––––––––––––––––––––––––––––––––––––––––––––––––
#define K1T 0x0040          // 0.125 * 2^RATIO_SCALE
#define B1T 0x01f2          // 0.0304 * 2^LUX_SCALE
#define M1T 0x01be          // 0.0272 * 2^LUX_SCALE

#define K2T 0x0080          // 0.250 * 2^RATIO_SCALE
#define B2T 0x0214          // 0.0325 * 2^LUX_SCALE
#define M2T 0x02d1          // 0.0440 * 2^LUX_SCALE

#define K3T 0x00c0          // 0.375 * 2^RATIO_SCALE
#define B3T 0x023f          // 0.0351 * 2^LUX_SCALE
#define M3T 0x037b          // 0.0544 * 2^LUX_SCALE

#define K4T 0x0100          // 0.50 * 2^RATIO_SCALE
#define B4T 0x0270          // 0.0381 * 2^LUX_SCALE
#define M4T 0x03fe          // 0.0624 * 2^LUX_SCALE
#define K5T 0x0138          // 0.61 * 2^RATIO_SCALE
#define B5T 0x016f          // 0.0224 * 2^LUX_SCALE
#define M5T 0x01fc          // 0.0310 * 2^LUX_SCALE

#define K6T 0x019a          // 0.80 * 2^RATIO_SCALE
#define B6T 0x00d2          // 0.0128 * 2^LUX_SCALE
#define M6T 0x00fb          // 0.0153 * 2^LUX_SCALE

#define K7T 0x029a          // 1.3 * 2^RATIO_SCALE
#define B7T 0x0018          // 0.00146 * 2^LUX_SCALE
```

```
#define M7T 0x0012          // 0.00112 * 2^LUX_SCALE

#define K8T 0x029a          // 1.3 * 2^RATIO_SCALE
#define B8T 0x0000          // 0.000 * 2^LUX_SCALE
#define M8T 0x0000          // 0.000 * 2^LUX_SCALE

//--------------------------------------------------
// CS package coefficients
//--------------------------------------------------
// For 0 <= Ch1/Ch0 <= 0.52
//          Lux/Ch0 = 0.0315-0.0593*((Ch1/Ch0)^1.4)
//          piecewise approximation
//                    For 0 <= Ch1/Ch0 <= 0.13
//                              Lux/Ch0 = 0.0315-0.0262*(Ch1/Ch0)
//                    For 0.13 <= Ch1/Ch0 <= 0.26
//                              Lux/Ch0 = 0.0337-0.0430*(Ch1/Ch0)
//                    For 0.26 <= Ch1/Ch0 <= 0.39
//                              Lux/Ch0 = 0.0363-0.0529*(Ch1/Ch0)
//                    For 0.39 <= Ch1/Ch0 <= 0.52
//                              Lux/Ch0 = 0.0392-0.0605*(Ch1/Ch0)
// For 0.52 < Ch1/Ch0 <= 0.65
//          Lux/Ch0 = 0.0229-0.0291*(Ch1/Ch0)
// For 0.65 < Ch1/Ch0 <= 0.80
//          Lux/Ch0 = 0.00157-0.00180*(Ch1/Ch0)
// For 0.80 < Ch1/Ch0 <= 1.30
//          Lux/Ch0 = 0.00338-0.00260*(Ch1/Ch0)
// For Ch1/Ch0 > 1.30
//          Lux = 0
//--------------------------------------------------
#define K1C 0x0043          // 0.130 * 2^RATIO_SCALE
#define B1C 0x0204          // 0.0315 * 2^LUX_SCALE
#define M1C 0x01ad          // 0.0262 * 2^LUX_SCALE

#define K2C 0x0085          // 0.260 * 2^RATIO_SCALE
#define B2C 0x0228          // 0.0337 * 2^LUX_SCALE
#define M2C 0x02c1          // 0.0430 * 2^LUX_SCALE

#define K3C 0x00c8          // 0.390 * 2^RATIO_SCALE
#define B3C 0x0253          // 0.0363 * 2^LUX_SCALE
#define M3C 0x0363          // 0.0529 * 2^LUX_SCALE

#define K4C 0x010a          // 0.520 * 2^RATIO_SCALE
#define B4C 0x0282          // 0.0392 * 2^LUX_SCALE
#define M4C 0x03df          // 0.0605 * 2^LUX_SCALE

#define K5C 0x014d          // 0.65 * 2^RATIO_SCALE
#define B5C 0x0177          // 0.0229 * 2^LUX_SCALE
#define M5C 0x01dd          // 0.0291 * 2^LUX_SCALE

#define K6C 0x019a          // 0.80 * 2^RATIO_SCALE
```

```
#define B6C 0x0101          // 0.0157 * 2^LUX_SCALE
#define M6C 0x0127          // 0.0180 * 2^LUX_SCALE

#define K7C 0x029a          // 1.3 * 2^RATIO_SCALE
#define B7C 0x0037          // 0.00338 * 2^LUX_SCALE
#define M7C 0x002b          // 0.00260 * 2^LUX_SCALE

#define K8C 0x029a          // 1.3 * 2^RATIO_SCALE
#define B8C 0x0000          // 0.000 * 2^LUX_SCALE
#define M8C 0x0000          // 0.000 * 2^LUX_SCALE

// lux equation approximation without floating point calculations
//////////////////////////////////////////////////////////////////////////////
//      Routine: unsigned int CalculateLux(unsigned int ch0, unsigned int ch0, int iType)
//
//      Description: Calculate the approximate illuminance (lux) given the raw
//              channel values of the TSL2560. The equation if implemented
//              as a piece–wise linear approximation.
//
//      Arguments: unsigned int iGain – gain, where 0:1X, 1:16X
//              unsigned int tInt – integration time, where 0:13.7mS, 1:100mS, 2:402mS,
//              3:Manual
//              unsigned int ch0 – raw channel value from channel 0 of TSL2560
//              unsigned int ch1 – raw channel value from channel 1 of TSL2560
//              unsigned int iType – package type (T or CS)
//
//      Return:  unsigned int – the approximate illuminance (lux)
//
//////////////////////////////////////////////////////////////////////////////
unsigned int CalculateLux(unsigned int iGain, unsigned int tInt, unsigned int ch0,
unsigned int ch1, int iType)
{

        //–––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––
        // first, scale the channel values depending on the gain and integration time
        // 16X, 402mS is nominal.
        // scale if integration time is NOT 402 msec
        unsigned long chScale;
        unsigned long channel1;
        unsigned long channel0;
        switch (tInt)
        {
            case 0:                                         // 13.7 msec
                    chScale = CHSCALE_TINT0;
                    break;
            case 1                                          // 101 msec
                    chScale = CHSCALE_TINT1;
                    break;
            default:                                        // assume no scaling
                    chScale = (1 << CH_SCALE);
```

```
                break;
}

// scale if gain is NOT 16X
if (!iGain) chScale = chScale << 4;                    // scale 1X to 16X

// scale the channel values
channel0 = (ch0 * chScale) >> CH_SCALE;
channel1 = (ch1 * chScale) >> CH_SCALE;
//--------------------------------------------------------------------------------

// find the ratio of the channel values (Channel1/Channel0)
// protect against divide by zero
unsigned long ratio1 = 0;
if (channel0 != 0) ratio1 = (channel1 << (RATIO_SCALE+1)) / channel0;

// round the ratio value
unsigned long ratio = (ratio1 + 1) >> 1;

// is ratio <= eachBreak ?
unsigned int b, m;
switch (iType)
{

    case 0:                                            // T, FN and CL package
                if ((ratio >= 0) && (ratio <= K1T))
                {b=B1T; m=M1T;}
                else if (ratio <= K2T)
                {b=B2T; m=M2T;}
                else if (ratio <= K3T)
                {b=B3T; m=M3T;}
                else if (ratio <= K4T)
                {b=B4T; m=M4T;}
                else if (ratio <= K5T)
                {b=B5T; m=M5T;}
                else if (ratio <= K6T)
                {b=B6T; m=M6T;}
                else if (ratio <= K7T)
                {b=B7T; m=M7T;}
                else if (ratio > K8T)
                {b=B8T; m=M8T;}
                break;

case 1:// CS package
    if ((ratio >= 0) && (ratio <= K1C))
                {b=B1C; m=M1C;}
                else if (ratio <= K2C)
                {b=B2C; m=M2C;}
                else if (ratio <= K3C)
                {b=B3C; m=M3C;}
```

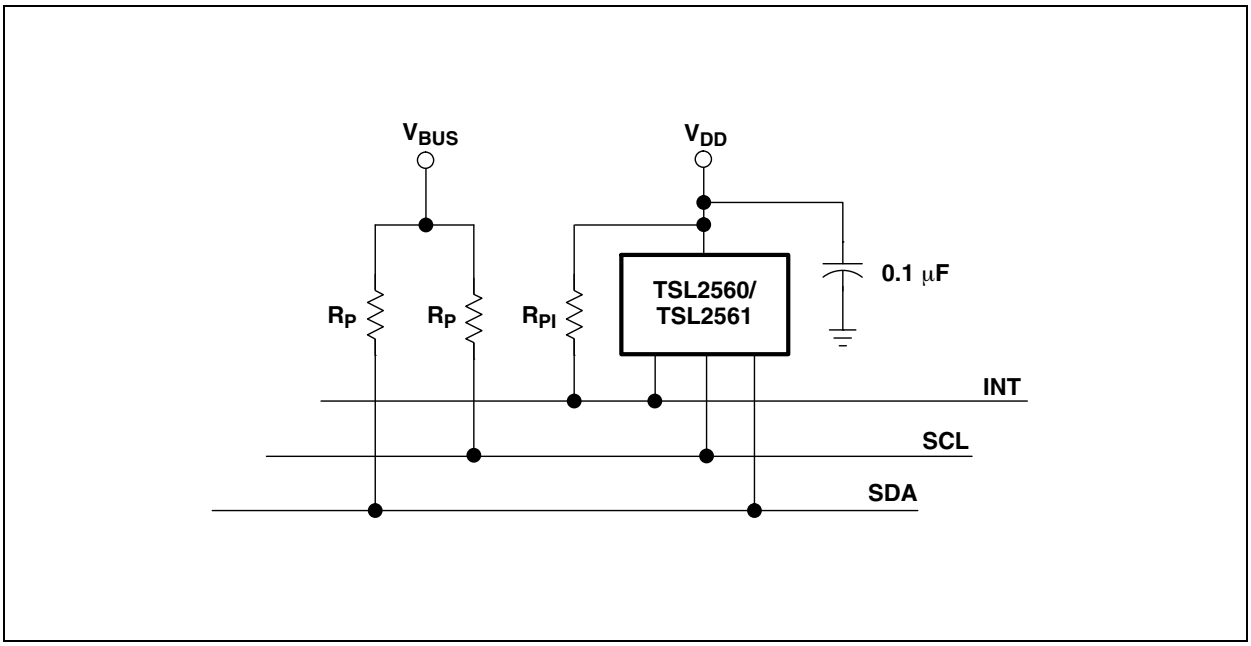
```
            else if (ratio <= K4C)
            {b=B4C; m=M4C;}
            else if (ratio <= K5C)
            {b=B5C; m=M5C;}
            else if (ratio <= K6C)
            {b=B6C; m=M6C;}
            else if (ratio <= K7C)
            {b=B7C; m=M7C;}
            else if (ratio > K8C)
            {b=B8C; m=M8C;}
            break;
    }
    unsigned long temp;
    temp = ((channel0 * b) – (channel1 * m));

    // do not allow negative lux value
    if (temp < 0) temp = 0;

    // round lsb (2^(LUX_SCALE–1))
    temp += (1 << (LUX_SCALE–1));

    // strip off fractional portion
    unsigned long lux = temp >> LUX_SCALE;

    return(lux);

}
```

# Application Information: Hardware

## Power Supply Decoupling and Application Hardware Circuit

The power supply lines must be decoupled with a 0.1µF capacitor placed as close to the device package as possible (Figure 42). The bypass capacitor should have low effective series resistance (ESR) and low effective series inductance (ESI), such as the common ceramic types, which provide a low impedance path to ground at high frequencies to handle transient currents caused by internal logic switching.

**Figure 42:**
**Bus Pull-Up Resistors**



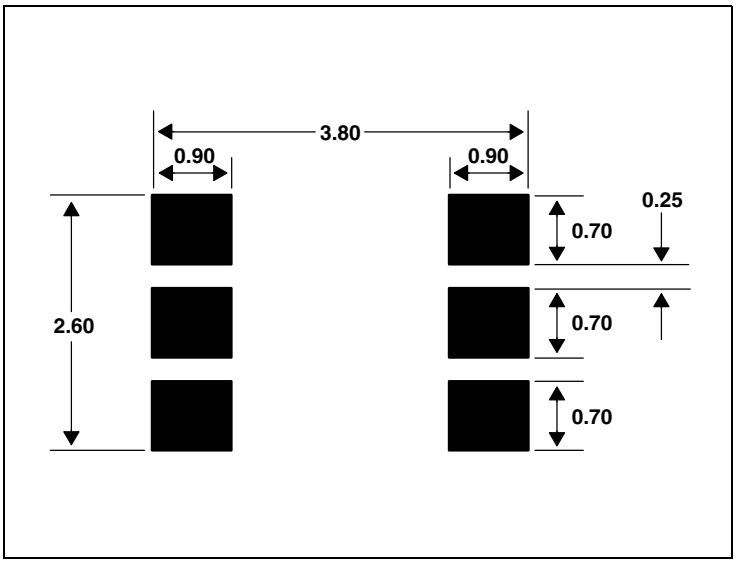

Pull-up resistors ($R_p$) maintain the SDAH and SCLH lines at a *high* level when the bus is free and ensure the signals are pulled up from a low to a high level within the required rise time. For a complete description of the SMBus maximum and minimum $R_p$ values, please review the SMBus Specification at http://www.smbus.org/specs. For a complete description of $I^2C$ maximum and minimum $R_p$ values, please review the $I^2C$ Specification at http://www.nxp.com.

A pull-up resistor (RPI) is also required for the interrupt (INT), which functions as a wired-AND signal in a similar fashion to the SCL and SDA lines. A typical impedance value between 10kΩ and 100kΩ can be used. Please note that while Figure 42 shows INT being pulled up to $V_{DD}$, the interrupt can optionally be pulled up to $V_{BUS}$.

## PCB Pad Layout

Suggested PCB pad layout guidelines for the TMB-6 (T) surface mount package, chipscale (CS) package, Dual Flat No-Lead (FN) surface mount package, and ChipLED–6 (CL) surface mount package are shown in Figure 43, Figure 44, Figure 45, and Figure 46.

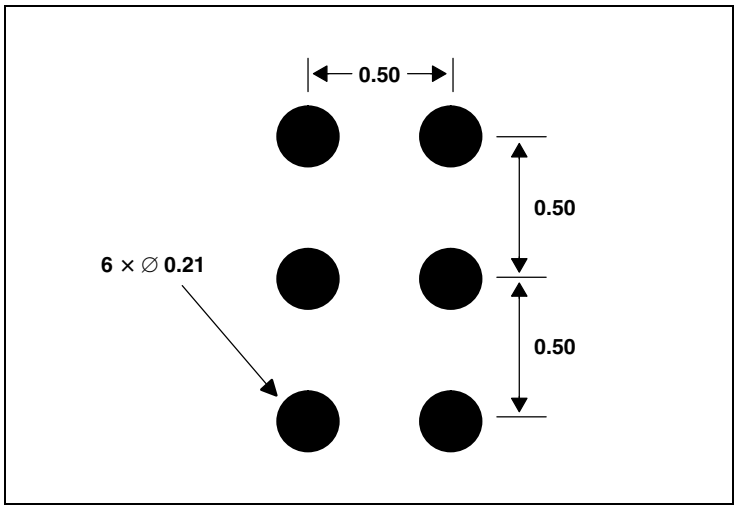
**Figure 43:**
**Suggested T Package PCB Layout**




**Note(s):**

1. All linear dimensions are in millimeters.
2. This drawing is subject to change without notice.

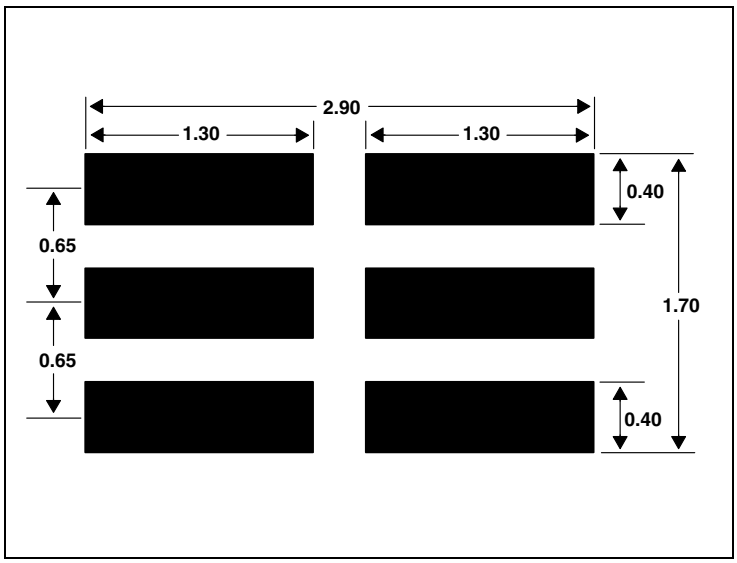
**Figure 44:**
**Suggested CS Package PCB Layout**




**Note(s):**

1. All linear dimensions are in millimeters.
2. This drawing is subject to change without notice.

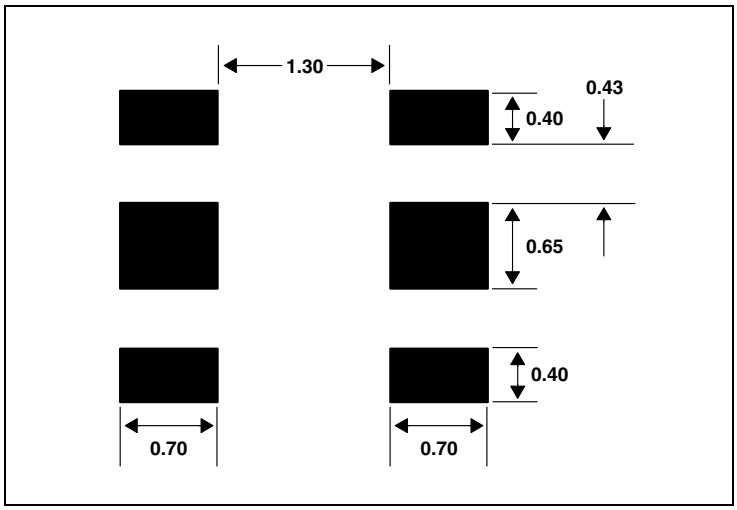
**Figure 45:**
**Suggested FN Package PCB Layout**




**Note(s):**

1. All linear dimensions are in millimeters.
2. This drawing is subject to change without notice.

**Figure 46:**
**Suggested CL Package PCB Layout**

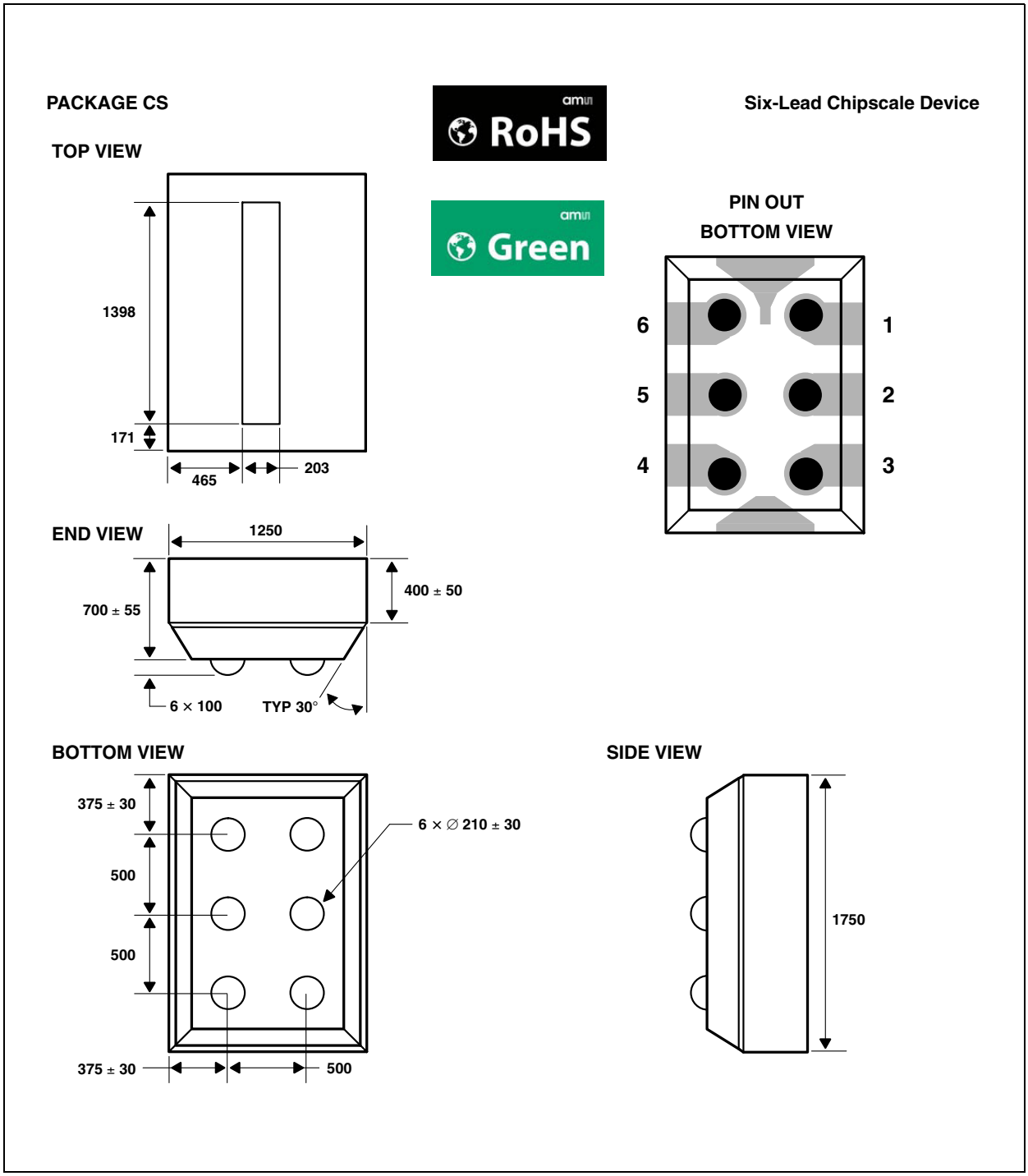



**Note(s):**

1. All linear dimensions are in millimeters.
2. This drawing is subject to change without notice.

# Packaging Mechanical Data

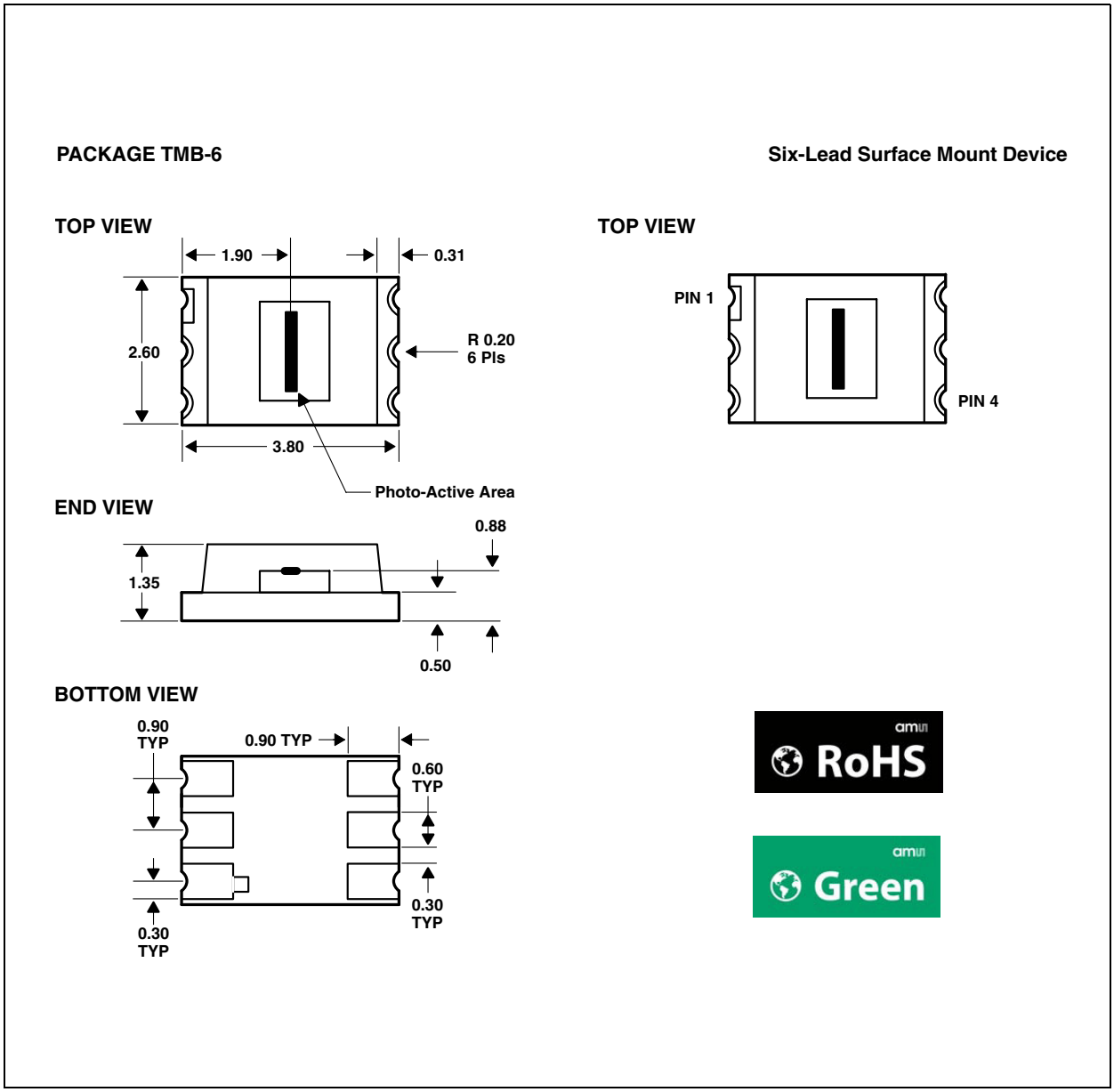
**Figure 47:**
**Package CS -Six-Lead Chipscale Packaging Configuration**




**Note(s):**

1. All linear dimensions are in micrometers. Dimension tolerance is ±25µm unless otherwise noted.
2. Solder bumps are formed of Sn (96.5%), Ag (3%), and Cu (0.5%).
3. The top of the photodiode active area is 410µm below the top surface of the package.
4. The layer above the photodiode is glass and epoxy with an index of refraction of 1.53.
5. This drawing is subject to change without notice.

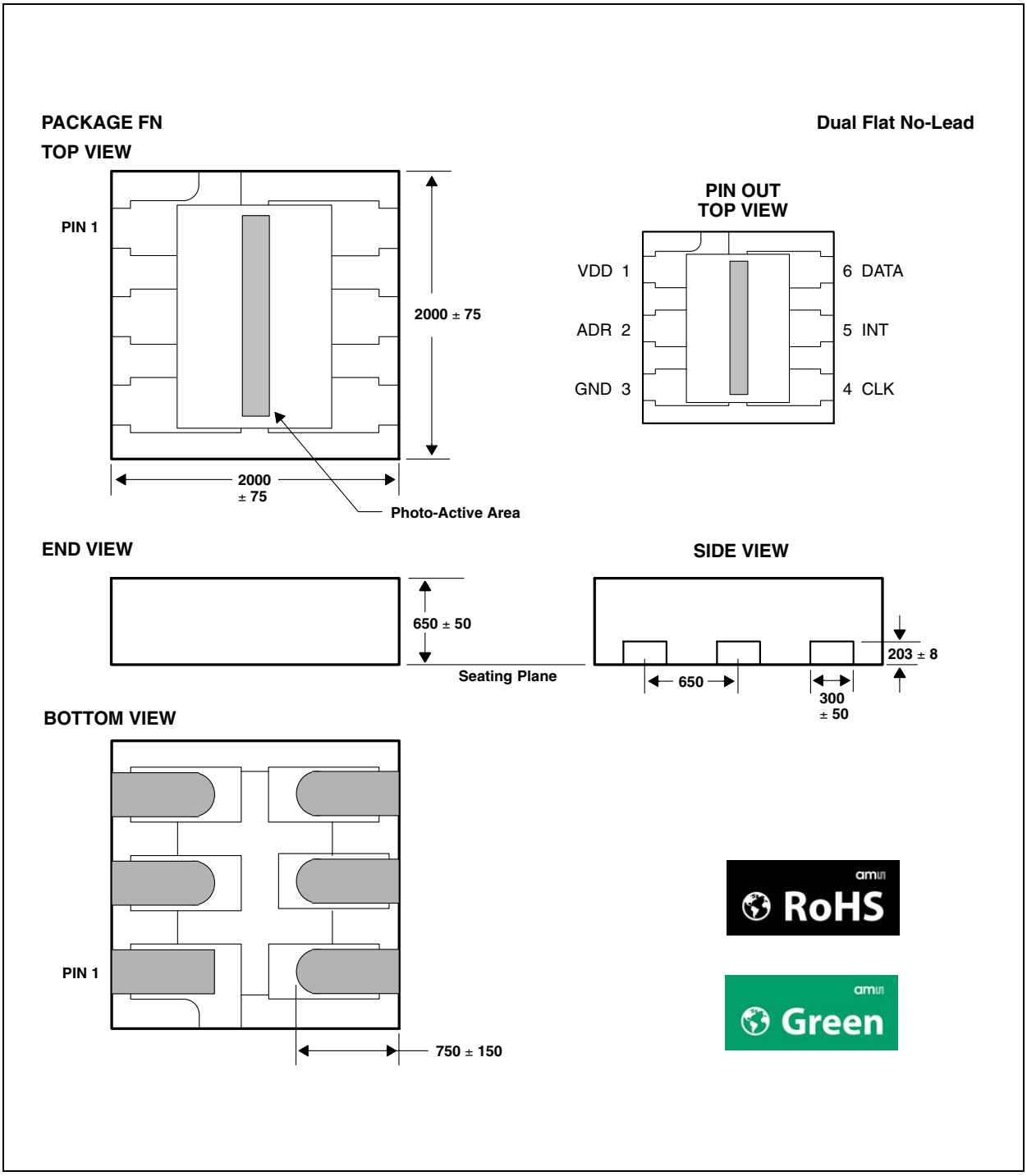
**Figure 48:**
**Package T - Six-Lead TMB Plastic Surface Mount Packaging Configuration**




**Note(s):**

1. All linear dimensions are in millimeters. Dimension tolerance is ±0.20mm unless otherwise noted.
2. The photo-active area is 1398µm by 203µm.
3. Package top surface is molded with an electrically nonconductive clear plastic compound having an index of refraction of 1.55.
4. Contact finish is 0.5µm minimum of soft gold plated over a 18µm thick copper foil pattern with a 5µm to 9µm nickel barrier.
5. The underside of the package includes copper traces used to connect the pads during package substrate fabrication. Accordingly, exposed traces and vias should not be placed under the footprint of the TMB package in a PCB layout.
6. This package contains no lead (Pb).
7. This drawing is subject to change without notice.

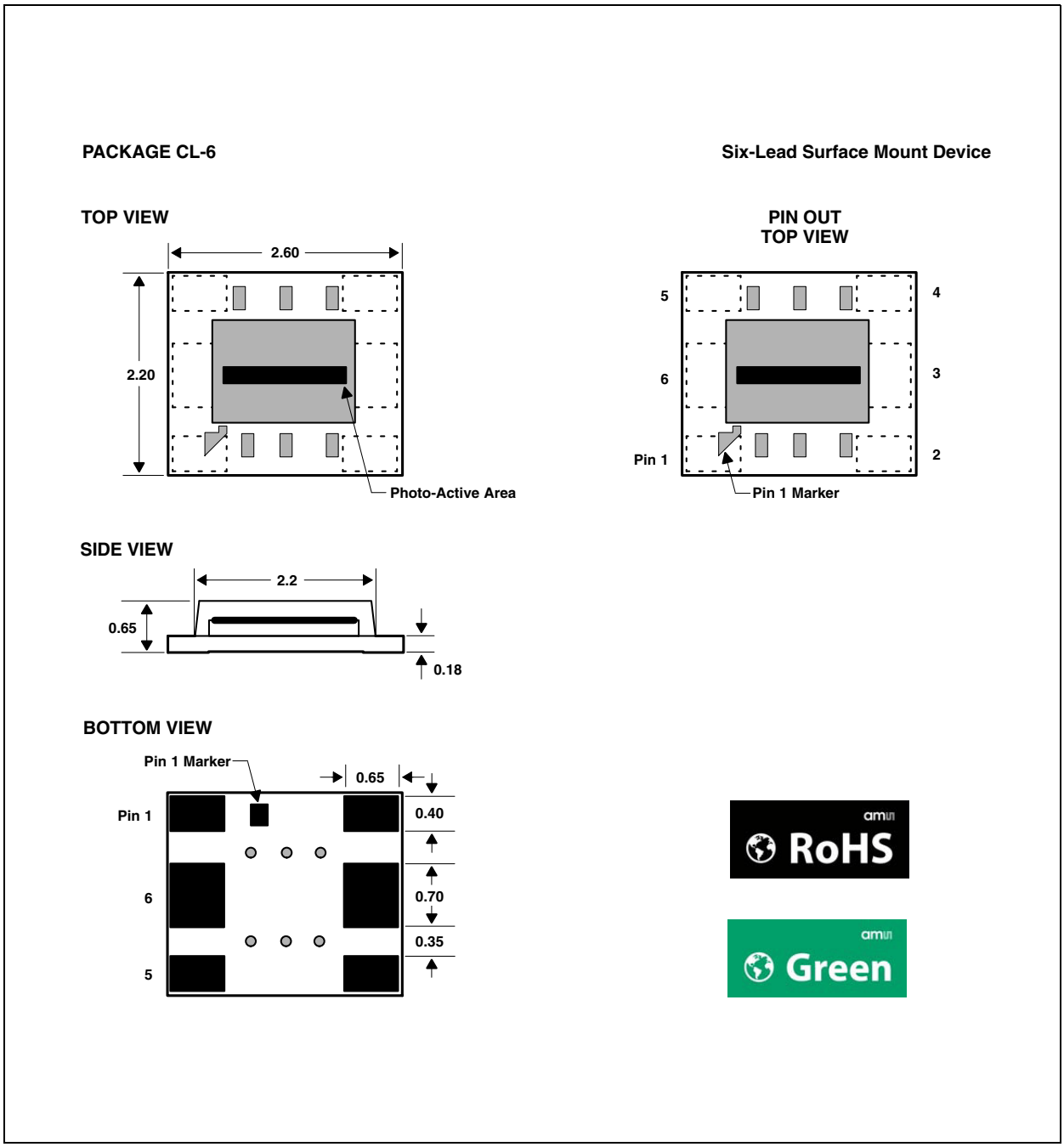
**Figure 49:**
**Package FN - Dual Flat No-Lead Packaging Configuration**




**Note(s):**

1. All linear dimensions are in micrometers. Dimension tolerance is ±20µm unless otherwise noted.
2. The photo-active area is 1398µm by 203µm.
3. Package top surface is molded with an electrically nonconductive clear plastic compound having an index of refraction of 1.55.
4. Contact finish is copper alloy A194 with pre-plated NiPdAu lead finish.
5. This package contains no lead (Pb).
6. This drawing is subject to change without notice.

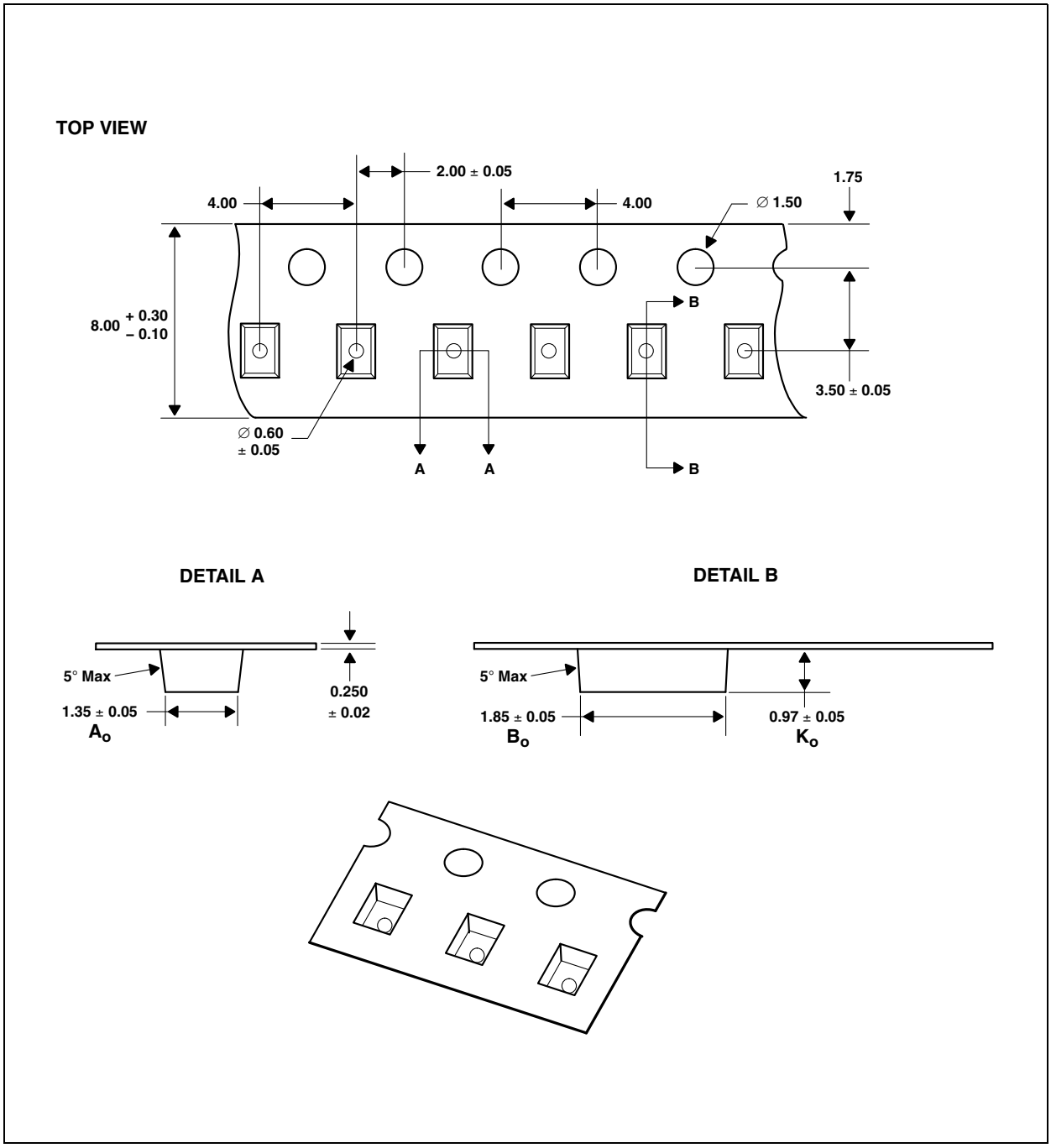
**Figure 50:**
**Package CL - Six-Lead ChipLED Plastic Surface Mount Packaging Configuration**




**Note(s):**

1. All linear dimensions are in millimeters. Dimension tolerance is ±0.10mm unless otherwise noted.
2. The photo-active area is 1398µm by 203µm.
3. Package top surface is molded with an electrically nonconductive clear plastic compound having an index of refraction of 1.55.
4. Contact finish is 0.1µm (minimum) to 1.0µm (maximum) of soft gold plated over a 15µm (minimum) to 30µm (maximum) thick copper foil pattern with a 3µm (minimum) to 15µm (maximum) nickel barrier.
5. This package contains no lead (Pb).
6. This drawing is subject to change without notice.

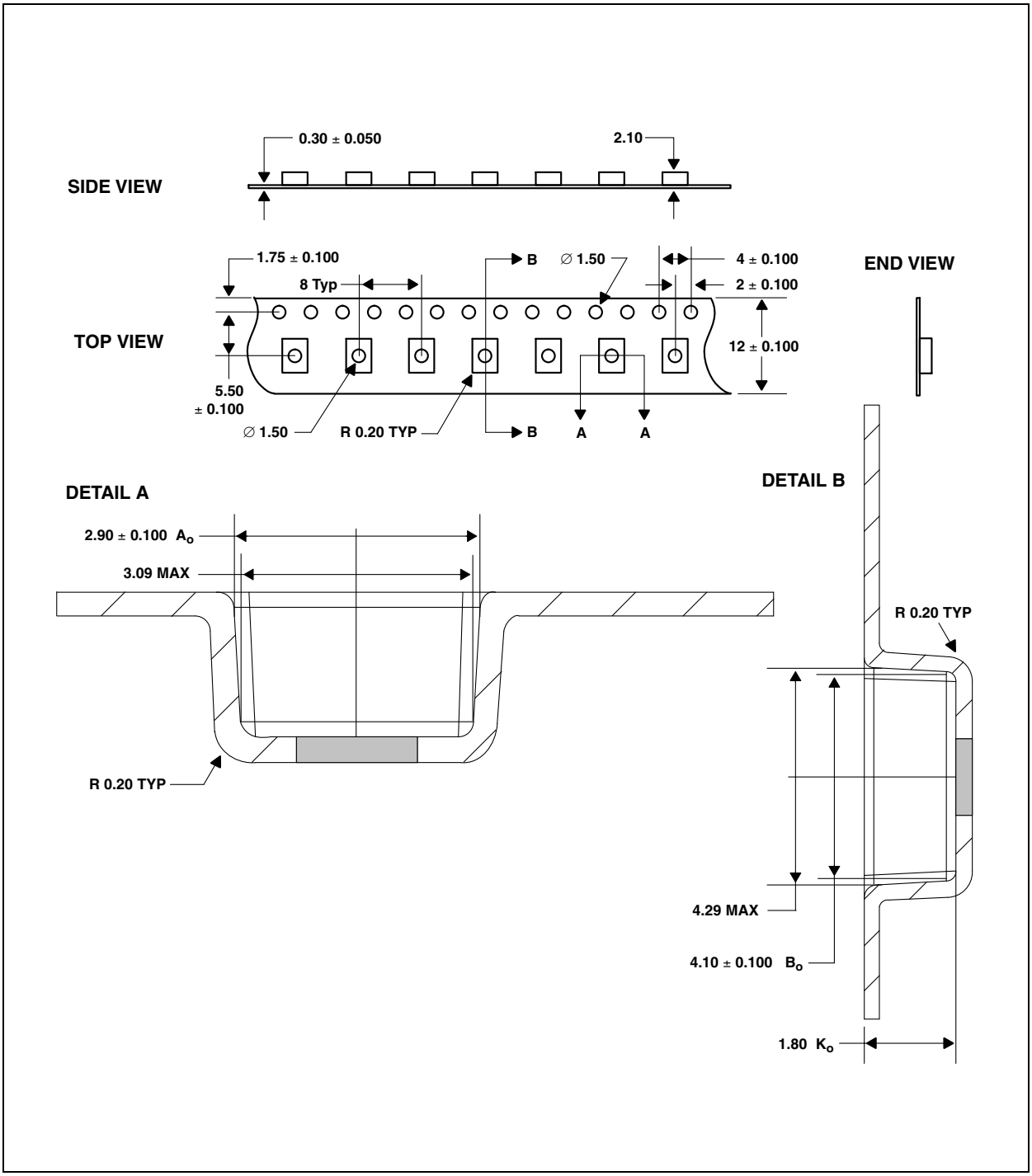
**Figure 51:**
**TSL2560/TSL2561 Chipscale Carrier Tape**




**Note(s):**

1. All linear dimensions are in millimeters. Dimension tolerance is ±0.10mm unless otherwise noted.
2. The dimensions on this drawing are for illustrative purposes only. Dimensions of an actual carrier may vary slightly.
3. Symbols on drawing $A_o$, $B_o$, and $K_o$ are defined in ANSI EIA Standard 481-B 2001.
4. Each reel is 178 millimeters in diameter and contains 3500 parts.
5. **ams** AG packaging tape and reel conform to the requirements of EIA Standard 481-B.
6. In accordance with EIA standard, device pin 1 is located next to the sprocket holes in the tape.
7. This drawing is subject to change without notice.

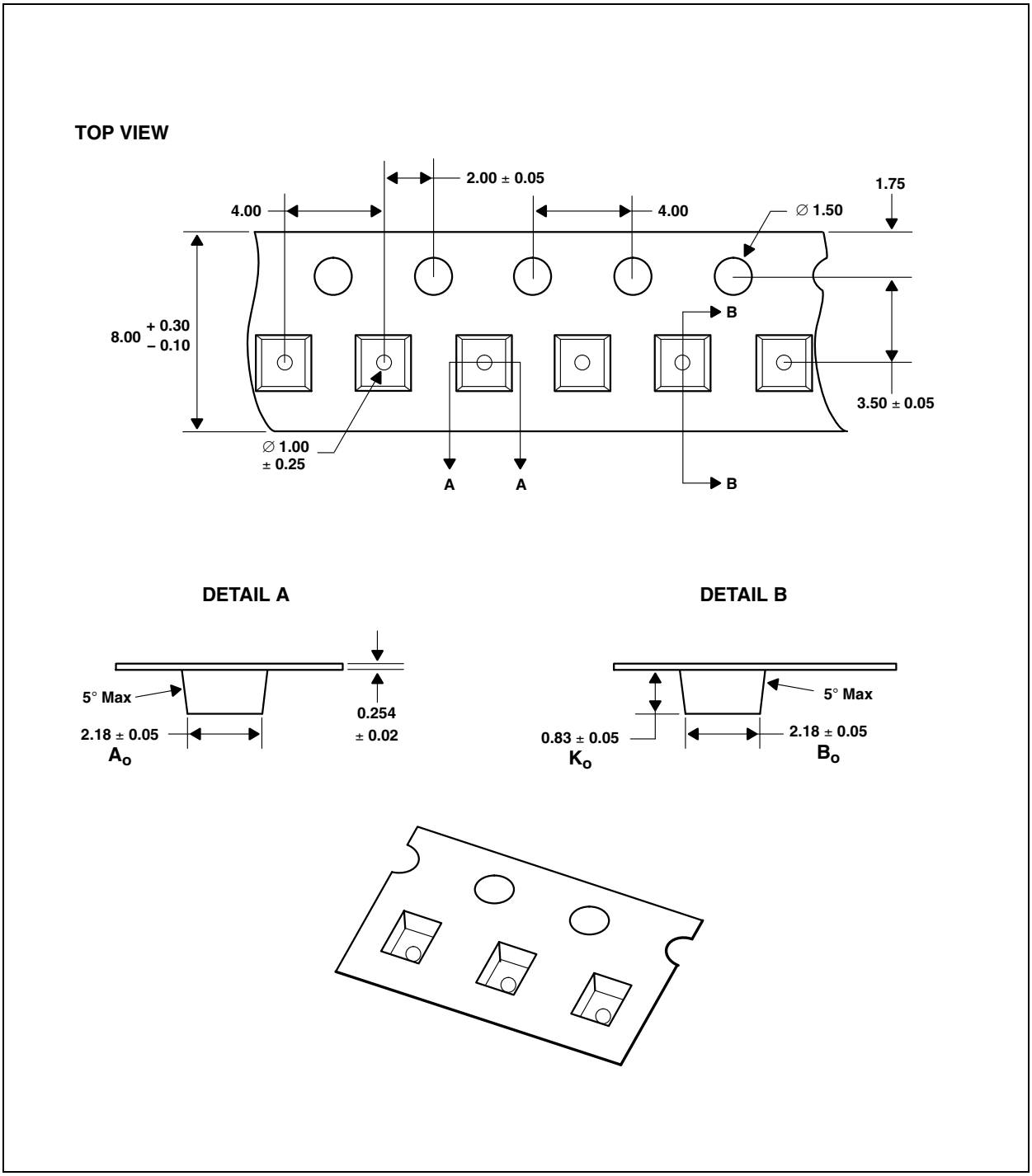
**Figure 52:**
**TSL2560/TSL2561 TMB Carrier Tape**




**Note(s):**

1. All linear dimensions are in millimeters.
2. The dimensions on this drawing are for illustrative purposes only. Dimensions of an actual carrier may vary slightly.
3. Symbols on drawing $A_o$, $B_o$, and $K_o$ are defined in ANSI EIA Standard 481-B 2001.
4. Each reel is 178 millimeters in diameter and contains 1000 parts.
5. **ams** AG packaging tape and reel conform to the requirements of EIA Standard 481-B.
6. In accordance with EIA standard, device pin 1 is located next to the sprocket holes in the tape.
7. This drawing is subject to change without notice.

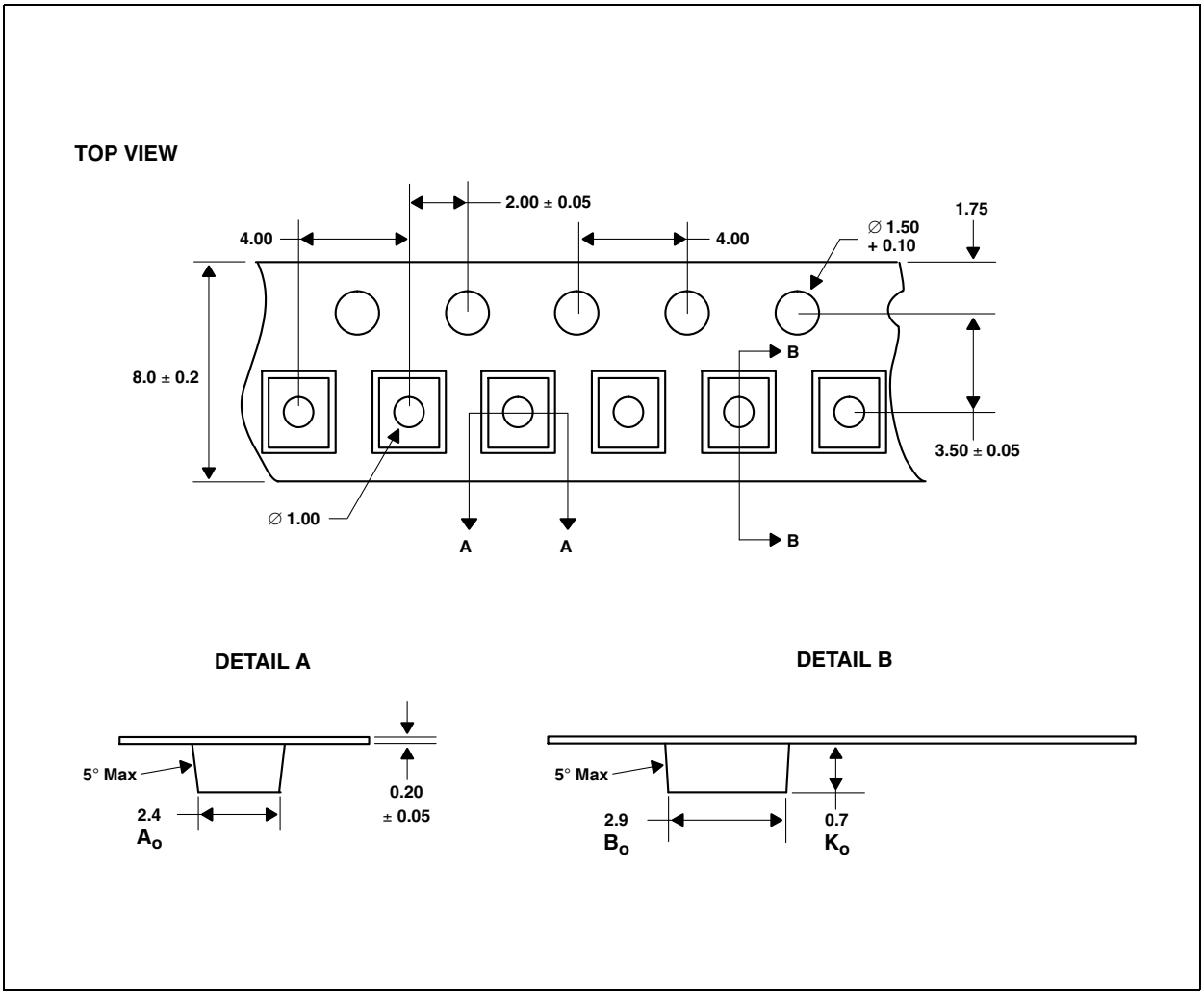
**Figure 53:**
**TSL2560/TSL2561 FN Carrier Tape**




**Note(s):**

1. All linear dimensions are in millimeters. Dimension tolerance is ±0.10mm unless otherwise noted.
2. The dimensions on this drawing are for illustrative purposes only. Dimensions of an actual carrier may vary slightly.
3. Symbols on drawing $A_o$, $B_o$, and $K_o$ are defined in ANSI EIA Standard 481-B 2001.
4. Each reel is 178 millimeters in diameter and contains 3500 parts.
5. **ams** AG packaging tape and reel conform to the requirements of EIA Standard 481-B.
6. In accordance with EIA standard, device pin 1 is located next to the sprocket holes in the tape.
7. This drawing is subject to change without notice.

**Figure 54:**
**TSL2560/TSL2561 CL Carrier Tape**




**Note(s):**

1. All linear dimensions are in millimeters. Dimension tolerance is ±0.10mm unless otherwise noted.
2. The dimensions on this drawing are for illustrative purposes only. Dimensions of an actual carrier may vary slightly.
3. Symbols on drawing $A_o$, $B_o$, and $K_o$ are defined in ANSI EIA Standard 481-B 2001.
4. Each reel is 178 millimeters in diameter and contains 2500 parts.
5. **ams** AG packaging tape and reel conform to the requirements of EIA Standard 481-B.
6. In accordance with EIA standard, device pin 1 is located next to the sprocket holes in the tape.
7. This drawing is subject to change without notice.

## Manufacturing Information

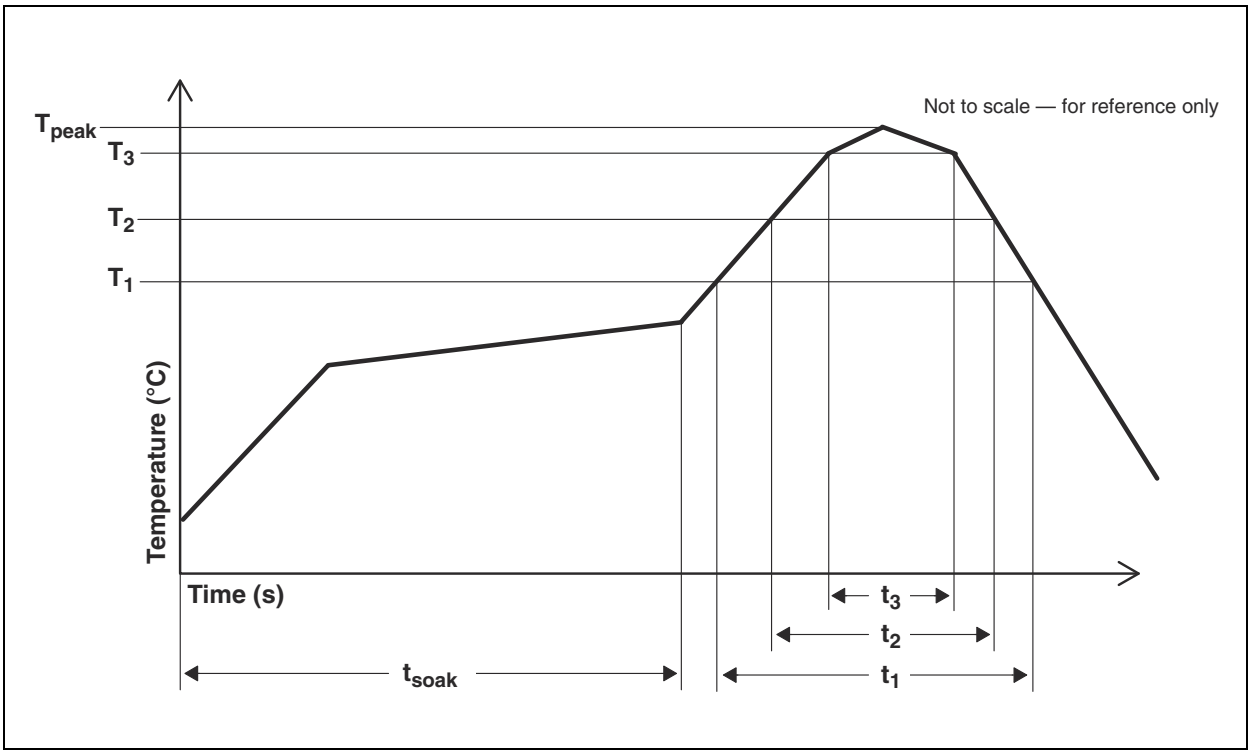
The CS, T, FN, and CL packages have been tested and have demonstrated an ability to be reflow soldered to a PCB substrate. The process, equipment, and materials used in these test are detailed below.

The solder reflow profile describes the expected maximum heat exposure of components during the solder reflow process of product on a PCB. Temperature is measured on top of component. The components should be limited to a maximum of three passes through this solder reflow profile.

**Figure 55:**
**TSL2560/61 Solder Reflow Profile**

| Parameter | Reference | TSL2560/61 |
|---|---|---|
| Average temperature gradient in preheating | | 2.5°C/s |
| Soak time | $t_{soak}$ | 2 to 3 minutes |
| Time above 217°C | $t_1$ | Max 60 s |
| Time above 230°C | $t_2$ | Max 50 s |
| Time above $T_{peak}$ -10°C | $t_3$ | Max 10 s |
| Peak temperature in reflow | $T_{peak}$ | 260° C (-0°C/+5°C) |
| Temperature gradient in cooling | | Max -5°C/s |

**Figure 56:**
**TSL2560/TSL2561 Solder Reflow Profile Graph**



## Moisture Sensitivity

Optical characteristics of the device can be adversely affected during the soldering process by the release and vaporization of moisture that has been previously absorbed into the package molding compound. To ensure the package molding compound contains the smallest amount of absorbed moisture possible, each device is dry-baked prior to being packed for shipping. Devices are packed in a sealed aluminized envelope with silica gel to protect them from ambient moisture during shipping, handling, and storage before use.

The CS package has been assigned a moisture sensitivity level of MSL 2 and the devices should be stored under the following conditions:

- Temperature Range: 5°C to 50°C
- Relative Humidity: 60% maximum
- Floor Life: 1 year out of bag at ambient < 30°C / 60% RH

Rebaking will be required if the aluminized envelope has been open for more than 1 year. If rebaking is required, it should be done at 90°C for 3 hours.

The T, FN, and CL packages have been assigned a moisture sensitivity level of MSL 3 and the devices should be stored under the following conditions:

- Temperature Range: 5°C to 50°C
- Relative Humidity: 60% maximum
- Total Time: 6 months from the date code on the aluminized envelope - if unopened
- Opened Time: 168 hours or fewer

Rebaking will be required if the devices have been stored unopened for more than 6 months or if the aluminized envelope has been open for more than 168 hours. If rebaking is required, it should be done at 90°C for 4 hours.

# Ordering & Contact Information

**Figure 57:**
**Ordering Information**

| Ordering Code | Device | Interface | Package - Leads | Package Designator |
|---|---|---|---|---|
| TSL2560CS | TSL2560 | SMBus | Chipscale | CS |
| TSL2560T | TSL2560 | SMBus | TMB-6 | T |
| TSL2560FN | TSL2560 | SMBus | Dual Flat No-Lead-6 | FN |
| TSL2560CL | TSL2560 | SMBus | ChipLED-6 | CL |
| TSL2561CS | TSL2561 | $I^2C$ | Chipscale | CS |
| TSL2561T | TSL2561 | $I^2C$ | TMB-6 | T |
| TSL2561FN | TSL2561 | $I^2C$ | Dual Flat No-Lead-6 | FN |
| TSL2561CL | TSL2561 | $I^2C$ | ChipLED-6 | CL |

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

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
Tobelbaderstrasse 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 059Q (2009-Nov) to current revision 1-00 (2016-May-05) | Page |
|---|---|
| Content of TAOS datasheet was converted to the latest **ams** design | |
| Updated Key Benefits & Features | 2 |
| Updated link above Figure 22 | 19 |
| Updated link below Figure 42 | 41 |

**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

# Mouser Electronics

Authorized Distributor

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

ams:
TSL2561T