# ADNS-2051
## Optical Mouse Sensor

**Avago**
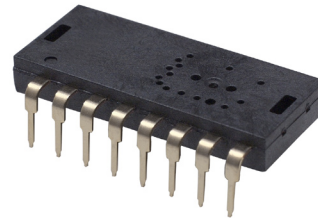TECHNOLOGIES

## Data Sheet

Lead (Pb) Free
RoHS 6 fully
compliant

## Description

The ADNS-2051 is a low cost optical sensor used to implement a non-mechanical tracking engine for computer mice.

It is based on optical navigation technology, which measures changes in position by optically acquiring sequential surface images (frames) and mathematically determining the direction and magnitude of movement.

The sensor is housed in a 16-pin staggered dual inline package (DIP) that is designed for use with the HDNS-2100 Lens and HDNS-2200 Clip and HLMP-ED80-XX000 (639 nm LED illuminator source). There are no moving parts, and precision optical alignment is not required, facilitating high volume assembly.

The output format is two channel quadrature (X and Y direction) which emulates encoder photo-transistors. The current X and Y information are also available in registers accessed via a serial port.

Default resolution is specified as 400 counts per inch (cpi), with rates of motion up to 14 inches per second (ips).

Resolution can also be programmed to 800 cpi.

The part is programmed via a two wire serial port, through registers.

## Theory of Operation

The ADNS-2051 is based on Optical Navigation Technology. It contains an Image Acquisition System (IAS), a Digital Signal Processor (DSP), a two-channel quadrature output, and a two wire serial port.

The IAS acquires microscopic surface images via the lens and illumination system provided by the HDNS-2100, 2200, and HLMP-ED80-XX000 LED. These images are processed by the DSP to determine the direction and distance of motion. The DSP generates the $\Delta x$ and $\Delta y$ relative displacement values that are converted into two channel quadrature signals.

## Features
- Precise optical navigation technology
- No mechanical moving parts
- Complete 2D motion sensor
- Serial interface and/or quadrature interface
- Smooth surface navigation
- Programmable frame speed up to 2300 frames per sec (fps)
- Accurate motion up to 14 ips
- 800 cpi resolution
- High reliability
- High speed motion detector
- No precision optical alignment
- Wave solderable
- Single 5.0 volt power supply
- Shutdown pin for USB suspend mode operation
- Power conservation mode during times of no movement
- On chip LED drive with regulated current
- Serial port registers
  – Programming
  – Data transfer
- 16-pin staggered dual inline package (DIP)

## Applications
- Mice for desktop PCs, workstations, and portable PCs
- Trackballs
- Integrated input devices

## Outline Drawing of ADNS-2051 Optical Mouse Sensor

**Pinout**

| Pin | Pin | Description |
| --- | --- | --- |
| 1 | SCLK | Serial port clock (input) |
| 2 | XA | XA quadrature output |
| 3 | XB | XB quadrature output |
| 4 | YB | YB quadrature output |
| 5 | YA | YA quadrature output |
| 6 | XY_LED | LED control |
| 7 | REFA | Internal reference |
| 8 | REFB | Internal reference |
| 9 | OSC_IN | Oscillator input |
| 10 | GND | System ground |
| 11 | OSC_OUT | Oscillator output |
| 12 | GND | System ground |
| 13 | $V_{DD}$ | 5.0 volt power supply |
| 14 | R_BIN | LED current bin resistor |
| 15 | PD | Power down pin, active high |
| 16 | SDIO | Serial data (input and output) |



**Figure 1. Top view**

2

PIN 1

A2051
XYYWWZ

22.30
(0.878)

50° ± 0.5°

*
0.01
(0.000)

*
2.98
(0.117)

LEAD WIDTH 0.50
(0.020)

0.25
(0.010)

1.26
(0.050) LEAD OFFSET

2.54
(0.100) LEAD PITCH

12.85 (AT SHOULDER)
(0.506)

9.10
(0.358)

0.99
(0.039)

3.18
(0.125)

5.16
(0.203)

0.50 CHAMFER

*
6.18
(0.243)

0.25
(0.010)

5° ± 3°

13.57 ± 0.45 (AT LEAD TIP)
(0.534 ± 0.018)

GATE LOCATION
SURFACE RECESSED
BY 0.3 mm

6.03
(0.237)

PROTECTIVE KAPTON TAPE
∅ 5.00

4.55
(0.179)

GROOVE

7.28 ± 0.03
(0.287 ± 0.001)

A

A

PIN 1

∅ 0.80 ± 0.03
CLEAR OPTICAL PATH

13.38
(0.527)

(MEASURED AT BASE) 5.60 ± 0.03
(0.220 ± 0.001)

*
0.06
(0.002)

1.85 ± 0.03
(0.073 ± 0.001)

1.43 ± 0.03
(0.056 ± 0.001)

*1.5° DRAFT

*
2.80
(0.110)

SECTION A-A

NOTES:
1. DIMENSIONS IN MILLIMETERS (INCHES).
2. DIMENSIONAL TOLERANCE: ± 0.1 mm.
3. COPLANARITY OF LEADS: 0.1 mm.
4. LEAD PITCH TOLERANCE: ± 0.15 mm.
5. CUMULATIVE PITCH TOLERANCE: ± 0.15 mm.
6. ANGULAR TOLERANCE: ± 3.0 DEGREES.
7. MAXIMUM FLASH + 0.2 mm.
8. CHAMFER (25 DEGREES x 2) ON THE TAPER SIDE OF THE LEAD.
9. * THESE DIMENSIONS ARE FOR REFERENCES ONLY AND SHOULD NOT BE USED TO MECHANICALLY REFERENCE THE SENSOR.
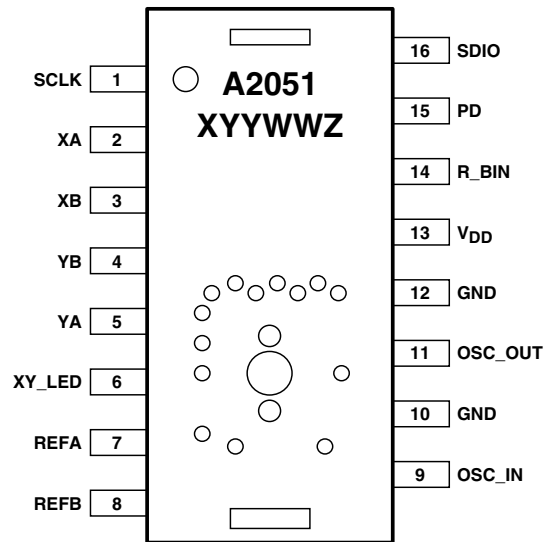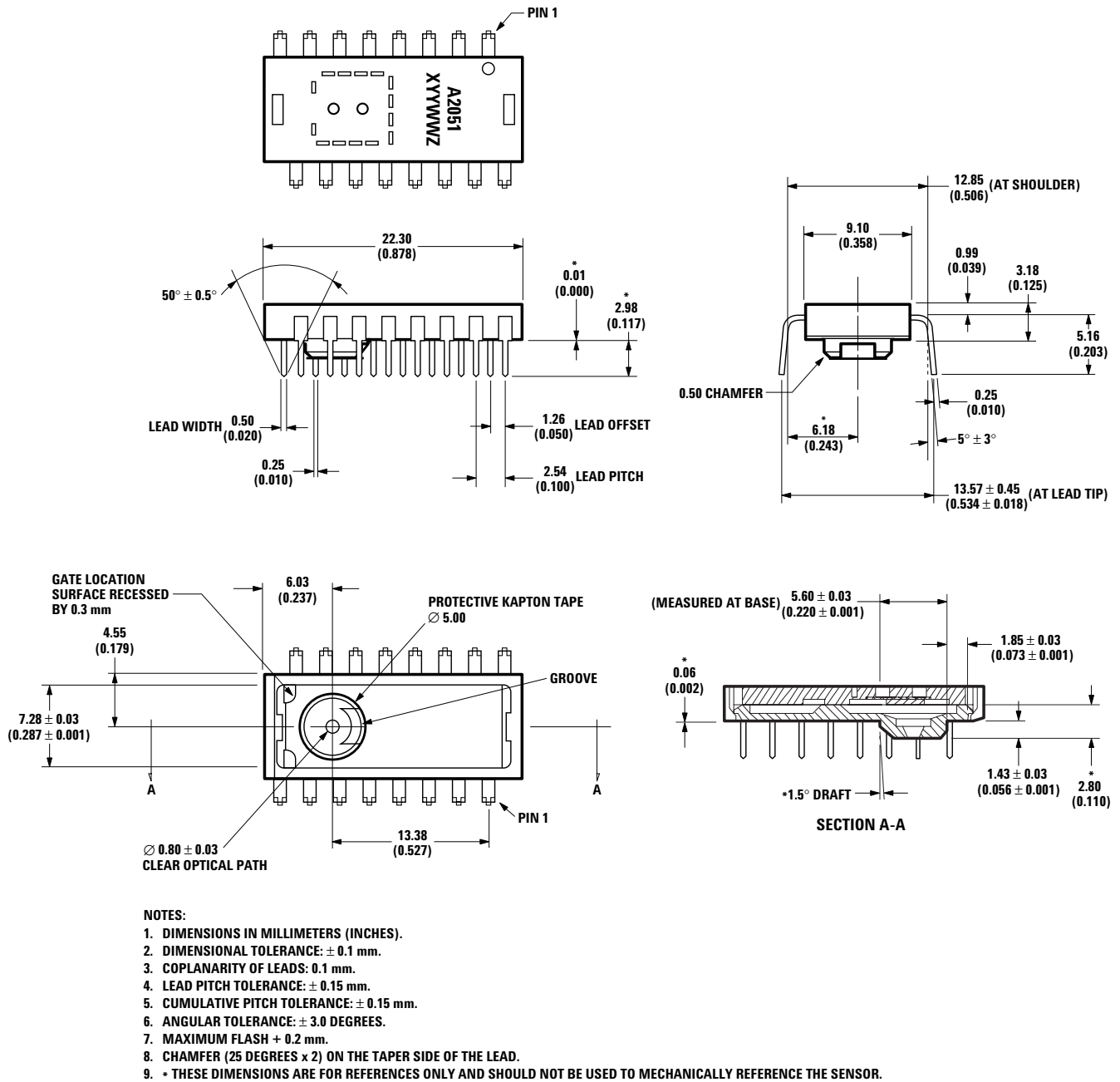
**Figure 2. Package outline drawing**

3

## Overview of Optical Mouse Sensor Assembly

2D Assembly Drawing of ADNS-2051

Figures 3 and 4, shown with HDNS-2100, HDNS-2200, and HLMP-ED80-XX000.

Avago Technologies provides an IGES file drawing describing the base plate molding features for lens and PCB alignment.

The components interlock as they are mounted onto defined features on the base plate.

The ADNS-2051 sensor is designed for mounting on a through hole PCB, looking down. There is an aperture stop and features on the package that align to the lens (see Figure 3).

The HDNS-2100 lens provides optics for the imaging of the surface as well as illumination of the surface at the optimum angle. Features on the lens align it to the sensor, base plate, and clip with the LED. The lens also has a large round flange to provide a long creepage path for any ESD events that occur at the opening of the base plate (see Figure 4).

The HDNS-2200 clip holds the LED in relation to the lens. The LED must be inserted into the clip and the LED's leads formed prior to loading on the PCB. The clip interlocks the sensor to the lens, and through the lens to the alignment features on the base plate.

The HLMP-ED80-XX000 LED is recommended for illumination. If used with the bin table, sufficient illumination can be guaranteed.
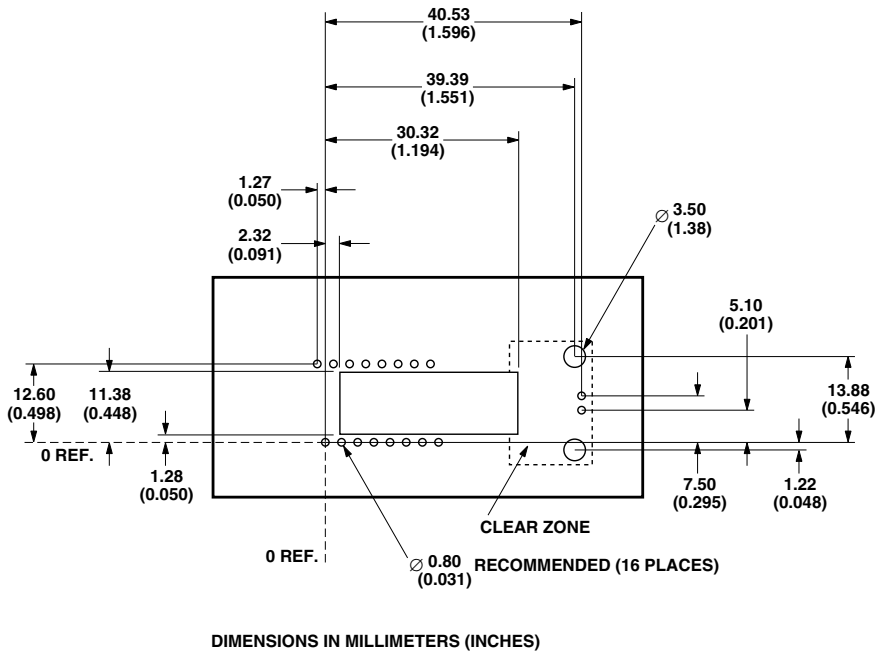
DIMENSIONS IN MILLIMETERS (INCHES)

**Figure 3. Recommended PCB mechanical cutouts and spacing (top view)**

DIMENSIONS IN MILLIMETERS (INCHES).
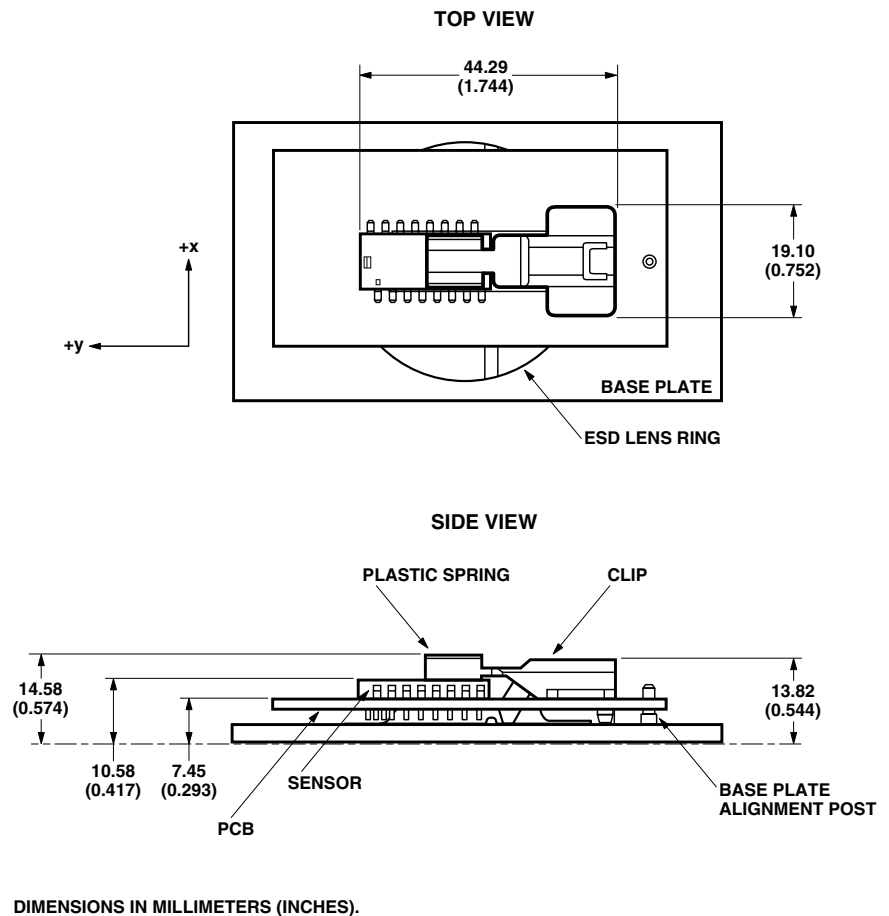
**Figure 4. 2D assembly drawing of ADNS-2051 (top and side view)**

4

HDNS-2200 (CLIP)

HLMP-ED80 (LED)

ADNS-2051 (SENSOR)

CUSTOMER SUPPLIED PCB

HDNS-2100 (LENS)

CUSTOMER SUPPLIED BASE PLATE
WITH RECOMMENDED ALIGNMENT
FEATURES PER IGES DRAWING
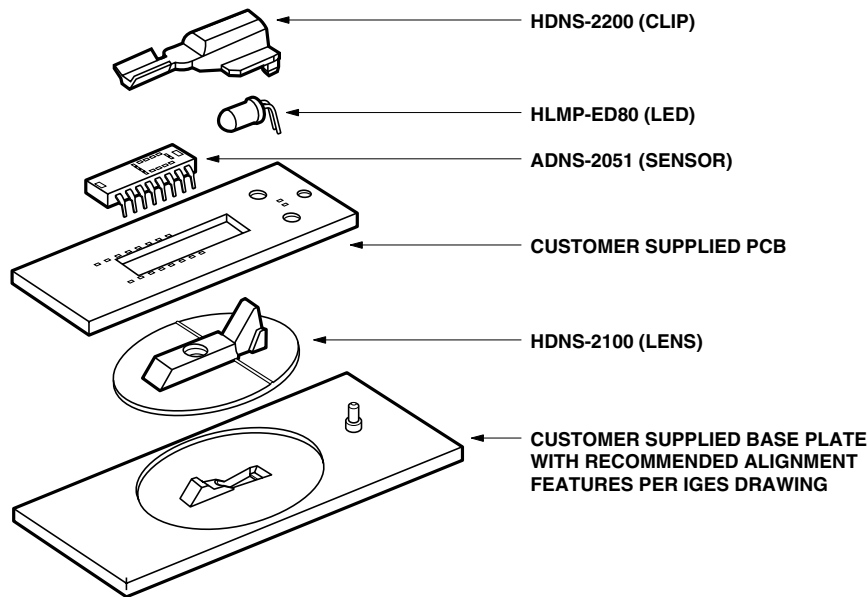
**Figure 5. Exploded view drawing**

## PCB Assembly Considerations

1. Insert the sensor and all other electrical components into PCB.

2. Bend the LED leads 90° and then insert the LED into the assembly clip until the snap feature locks the LED base.

3. Insert the LED/clip assembly into PCB.

4. Wave Solder the entire assembly in a no-wash solder process utilizing solder fixture. The solder fixture is needed to protect the sensor during the solder process. The fixture should be designed to expose the sensor leads to solder while shielding the optical aperture from direct solder contact. The solder fixture is also used to set the reference height of the sensor to the PCB top during wave soldering (Note: DO NOT remove the kapton tape during wave soldering).

5. Place the lens onto the base plate.

6. Remove the protective kapton tape from optical aperture of the sensor. Care must be taken to keep contaminants from entering the aperture. It is recommended not to place the PCB facing up during the entire mouse assembly process. The PCB should be held vertically during the kapton removal process.

7. Insert PCB assembly over the lens onto the base plate aligning post to retain PCB assembly. The sensor aperture ring should self-align to the lens.

8. The optical position reference for the PCB is set by the base plate and lens. Note that the PCB motion due to button presses must be minimized to maintain optical alignment.

9. Install mouse top case. There MUST be a feature in the top case to press down onto the clip to ensure all components are interlocked to the correct vertical height.
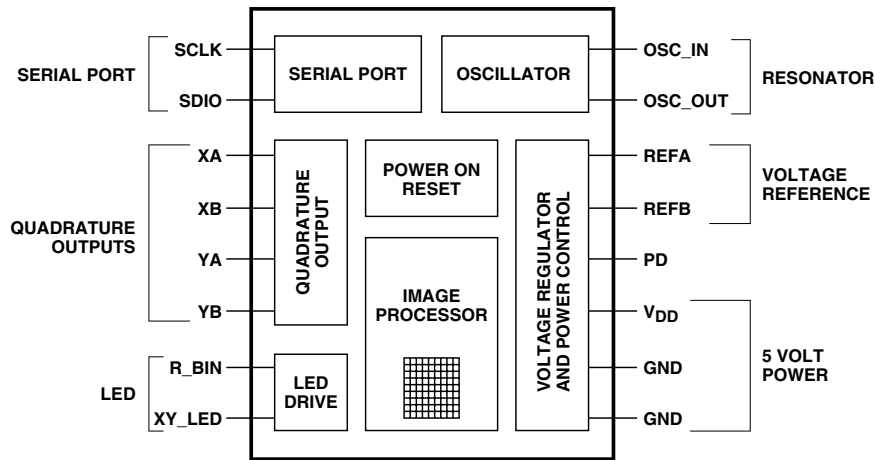
5

**Figure 6. Block diagram of ADNS-2051 optical mouse sensor**

## Design Considerations for Improving ESD Performance

The flange on the lens has been designed to increase the creepage and clearance distance for electrostatic discharge. The table on the right shows typical values assuming base plate construction per the Avago supplied IGES file and HDNS-2100 lens flange.

For improved ESD performance, the lens flange can be sealed (i.e. glued) to the base plate. Note that the lens material is polycarbonate and therefore, cyanoacrylate-based adhesives or other adhesives that may damage the lens should NOT be used.

The trimmed lens, HDNS-2100#001, is not recommended for corded applications due to the ESD spec requirement.
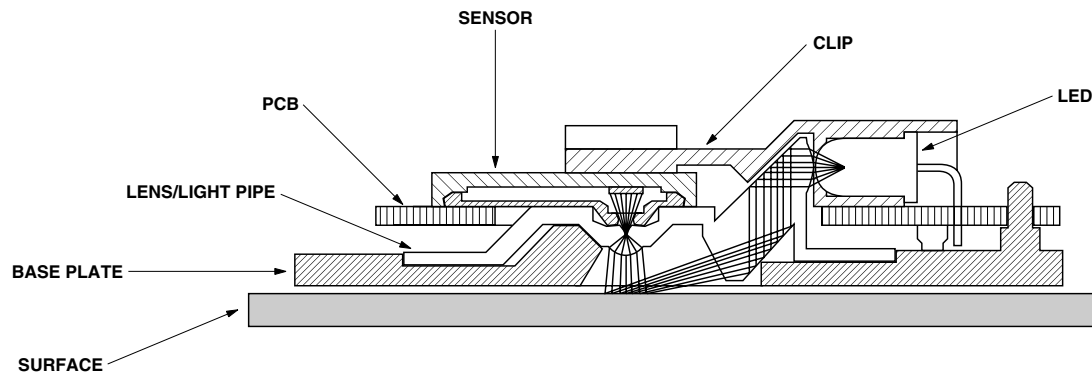
| Typical Distance | Millimeters |
|---|---|
| Creepage | 16.0 |
| Clearance | 2.1 |



**Figure 7. PCB assembly**

6

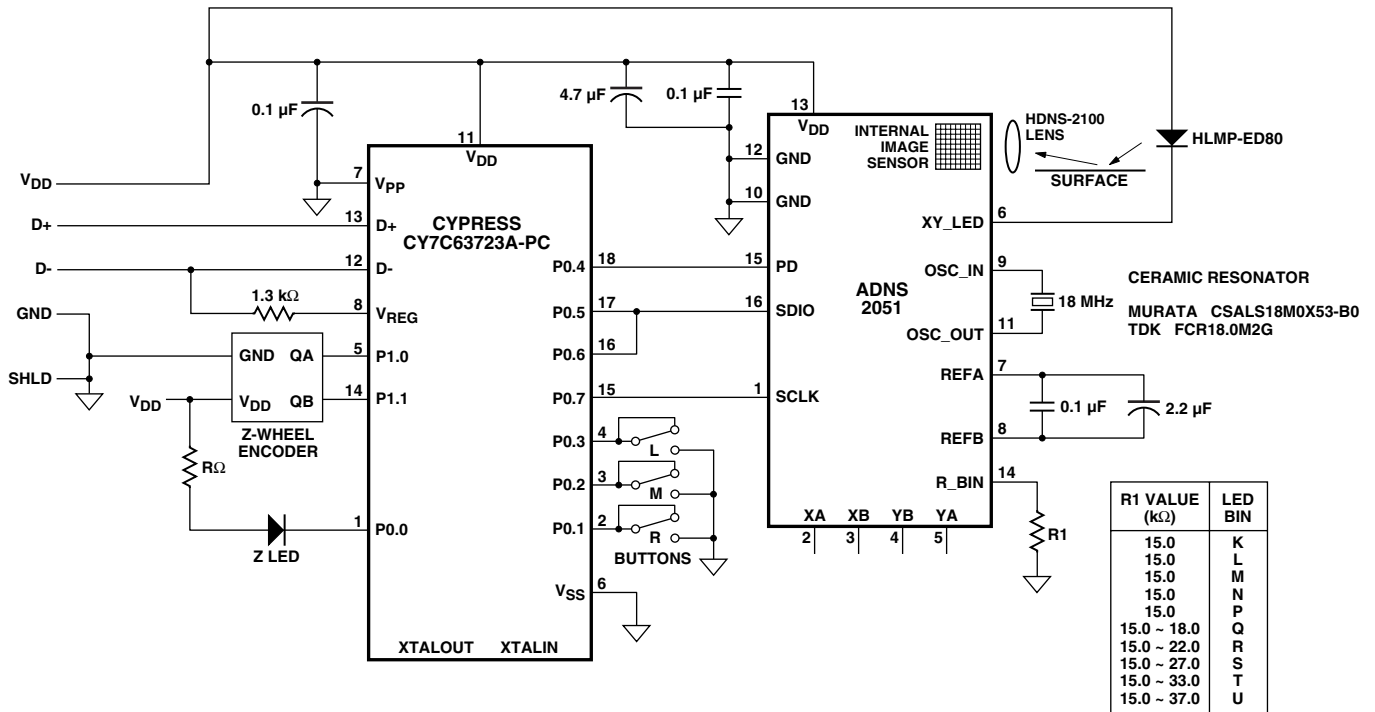## Recommended Typical Application Using SDIO Pins



**Figure 8. Application using SDIO pins**

## Notes on Bypass Capacitors:

- Caps for pins 7, 8 and 12, 13 MUST have trace lengths LESS than 5 mm.
- The 0.1 μF caps must be ceramic.
- Caps should have less than 5 nH of self inductance
- Caps should have less than 0.2 Ω ESR

Surface mount parts are recommended.

SDIO and SCLK pins should be grounded if not used.

## Regulatory Requirements

- Passes FCC B and worldwide analogous emission limits when assembled into a mouse with unshielded cable and following Avago recommendations.
- Passes EN61000-4-4/IEC801-4 EFT tests when assembled into a mouse with unshielded cable and following Avago recommendations.
- UL flammability level UL94 V-0.
- Provides sufficient ESD creepage/clearance distance to avoid discharge up to 15 kV when assembled into a mouse according to usage instructions above.
- For eye safety consideration, please refer to the technical report available on the web site,

  http://www.Avago.com

- The 15.0 kΩ resistor is determined by the absolute maximum rating of 50 mA for the HLMP-ED80-XX000. The other resistor values for brighter bins will guarantee good signals with reduced power.
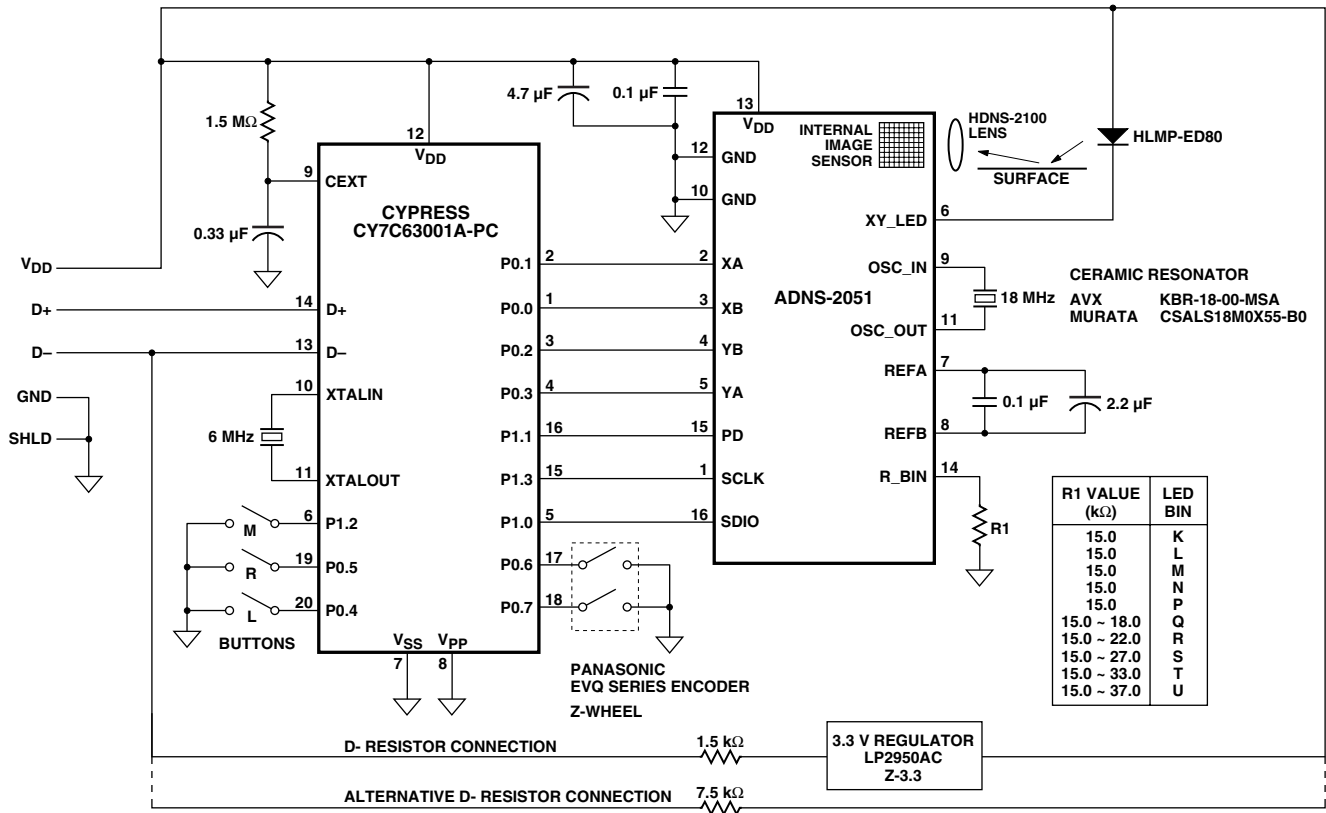
7

## Alternative Application using Quadrature Output Pins



**Figure 9. Application using quadrature output pins**

## Absolute Maximum Ratings

| Parameter | Symbol | Min. | Max. | Units | Notes |
|---|---|---|---|---|---|
| Storage Temperature | $T_S$ | −40 | 85 | ˚C | |
| Operating Temperature | $T_A$ | −15 | 55 | ˚C | |
| Lead Solder Temperature | | | 260 | ˚C | For 10 seconds, 1.6 mm below seating plane. |
| Supply Voltage | $V_{DD}$ | −0.5 | 5.5 | V | |
| ESD | | | 2 | kV | All pins, human body model MIL 883 Method 3015 |
| Input Voltage | $V_{IN}$ | −0.5 | $V_{DD} + 0.5$ | V | PD, SDIO, SCLK, XA, XB, YA, YB, XY_LED, R_BIN |
| Input Voltage | $V_{IN}$ | −0.5 | 3.6 | V | OSC_IN, OSC_OUT, REF_A |

8

## Recommended Operating Conditions

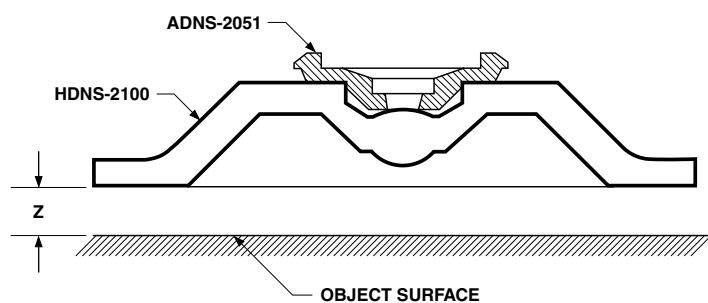| Parameter | Symbol | Min. | Typ. | Max. | Units | Notes |
|---|---|---|---|---|---|---|
| Operating Temperature | $T_A$ | 0 | | 40 | °C | |
| Power Supply Voltage | $V_{DD}$ | 4.25 | 5.0 | 5.5 | volts | Register values retained for voltage transients below 4.25 V but greater than 4 V. |
| Power Supply Rise Time | $V_{RT}$ | | | 100 | ms | |
| Supply Noise | $V_N$ | | | 100 | mV | Peak to peak within 0-100 MHz. |
| Clock Frequency | $f_{CLK}$ | 17.4 | 18.0 | 18.7 | MHz | Set by ceramic resonator. |
| Serial Port Clock Frequency | SCLK | | | $f_{CLK}/4$ | MHz | |
| Resonator Impendance | $X_{RES}$ | | | 55 | Ω | |
| Distance from Lens Reference Plane to Surface | Z | 2.3 | 2.4 | 2.5 | mm | Results in ±0.2 mm DOF. (See Figure 10.) |
| Speed | S | 0 | | 14 | in/sec | @ frame rate = 1500/second. |
| Acceleration | A | | | 0.15 | g | @ frame rate = 1500/second. |
| Light Level onto IC | $IRR_{INC}$ | 80<br>100 | | 25,000<br>30,000 | mW/m² | $\lambda = 639$ nm<br>$\lambda = 875$ nm |
| SDIO Read Hold Time | $t_{HOLD}$ | 100 | | | µs | Hold time for valid data. (Refer to Figure 28.) |
| SDIO Serial Write-Write Time | $t_{SWW}$ | 100 | | | µs | Time between two write commands. (Refer to Figure 31.) |
| SDIO Serial Write-Read Time | $t_{SWR}$ | 100 | | | µs | Time between write and read operation. (Refer to Figure 32.) |
| SDIO Serial Read-Write Time | $t_{SRW}$ | 120 | | | ns | Time between read and write operation. (Refer to Figure 33.) |
| SDIO Serial Read-Read Time | $t_{SRR}$ | 120 | | | ns | Time between two read commands. (Refer to Figure 33.) |
| Data Delay after PD ↓ | $t_{COMPUTE}$ | 3.2 | | | ms | After $t_{COMPUTE}$, all registers contain data from first image after PD↓. Note that an additional 75 frames for AGC (shutter) stabilization may be required if mouse movement occurred while PD was high. (Refer to Figure 12.) |
| SDIO Write Setup Time | $t_{SETUP}$ | 60 | | | ns | Data valid time before the rising of SCLK. (Refer to Figure 26.) |
| PD Pulse Width (to power down the chip) | $t_{PDW}$ | 700 | | | µs | Pulse width to initiate the power down cycle @ 1500 fps. (Refer to Figure 12 and Figure 14.) |
| PD Pulse Width (to reset the serial port) | $t_{PDR}$ | 100 | | | µs | Pulse width to reset the serial port @ 1500 fps (but may also initiate a power down cycle. Normal PD recovery sequence to be followed. (Refer to Figure 15.) |
| Frame Rate | FR | | 1500 | | frames/s | See Frame_Period register section. |
| Bin Resistor | R1 | 15 K | 15 K | 37 K | Ω | Refer to Figure 8. |

9

**Figure 10. Distance from lens reference plane to surface**

## AC Electrical Specifications

Electrical Characteristics over recommended operating conditions. Typical values at 25˚C, $V_{DD}$ = 5.0 V, 1500 fps, 18 MHz.

| Parameter | | Symbol | Min. | Typ. | Max. | Units | Notes |
|---|---|---|---|---|---|---|---|
| Power Down | | $t_{PD}$ | | 700 | | µs | From PD ↑ Time uncertainty due to firmware delay. (Refer to Figure 12.) |
| Power Up from PD ↓ | | $t_{PUPD}$ | | | 50 | ms | From PD↓ to valid quad signals 705 µsec + 75 frames. (Refer to Figure 12.) |
| Power Up from $V_{DD}$ ↑ | | $t_{PU}$ | | | 30 | ms | From $V_{DD}$↑ to valid quad signals 705 µsec + 40 frames |
| Rise and Fall Times: | SDIO | $t_r$ | | 30 | | ns | $C_L$ = 30 pF (the rise time is between 10% and 90%) |
| | | $t_f$ | | 16 | | ns | $C_L$ = 30 pF (the fall time is between 10% and 90%) |
| | XA, XB, YA, YB | $t_r$ | | 50 | | ns | $C_L$ = 30 pF (the rise time is between 10% and 90%) |
| | | $t_f$ | | 20 | | ns | $C_L$ = 30 pF (the fall time is between 10% and 90%) |
| | ILED | $t_r$ | | 40 | | ns | With HLMP-ED80-XX000 LED (the rise time is between 10% and 90%) |
| | | $t_f$ | | 200 | | ns | With HLMP-ED80-XX000 LED (the fall time is between 10% and 90%) |
| Serial Port Transaction Timer | | $t_{SPTT}$ | 0.7 | 0.9 | 1.0 | s | Serial port will reset if current transaction is not complete within $t_{SPTT}$. (Refer to Figure 36.) |
| Transient Supply Current | | $I_{DDT}$ | | 20 | 37 | mA | Max. supply current during a $V_{DD}$ ramp from 0 to 5.0 V with > 500 µs rise time. Does not include charging current for bypass capacitors. |

10

## DC Electrical Specifications

Electrical Characteristics over recommended operating conditions. Typical values at 25˚C, $V_{DD}$ = 5.0 V, 18 MHz.

| Parameter | Symbol | Min. | Typ. | Max. | Units | Notes |
|---|---|---|---|---|---|---|
| DC Supply Current (mouse moving) | $I_{DD\ AVG}$ | | 15 | 25 | mA | No load on XA, XB, YA, YB, SCLK, SDIO. Excluding LED current. |
| Peak Supply Current (mouse moving) | $I_{DD\ PEAK}$ | | 20 | | mA | No load on XA, XB, YA, YB, SCLK, SDIO. Excluding LED current. |
| DC Supply Current (mouse not moving) | $I_{DD}$ | | 12 | 25 | mA | No load on XA, XB, YA, YB, SCLK, SDIO. Excluding LED current. |
| DC Supply Current (power down) | $I_{DDPD}$ | | 170 | 240 | µA | PD = high; SCLK, SDIO = GND or $V_{DD}$; $V_{DD}$ = 4.25 V to 5.25 V. |
| SCLK, SDIO, PD | | | | | | |
| Input Low Voltage | $V_{IL}$ | | | 0.8 | V | |
| Input High Voltage | $V_{IH}$ | $0.5 * V_{DD}$ | | | V | |
| Output Low Voltage | $V_{OL}$ | | | 0.7 | V | @ $I_{OL}$ = 2 mA (SDIO only) |
| Output High Voltage | $V_{OH}$ | $0.6 * V_{DD}$ | | | V | @ $I_{OH}$ = 2 mA (SDIO only) |
| Output Low Voltage (XA, XB, YA, YB) | $V_{OL}$ | | | 0.4 | V | @ $I_{OL}$ = 0.5 mA. |
| Output High Voltage (XA, XB, YA, YB) | $V_{OH}$ | $0.6 * V_{DD}$ | | | V | @ $I_{OH}$ = 0.5 mA . |
| Output Low Voltage (XY_LED) | $V_{OL}$ | | | 1.1 | V | Refer to Figure 11. |
| XY LED Current | $I_{LED}$ | Typ−15% | 630/R1 | Typ + 15% | A | Refer to Figure 11, see table below. |
| XY LED Current (fault mode) | $I_{LED}$ | | | 500 | µA | R1 < 200 Ω. |
| REF_A (normal mode) | $V_{REFA}$ | | 3.3 | | V | 1.5 KΩ to 3.0 V or GND, PD = low. |
| REF_A (power down mode) | $V_{REFA}$ | | 3.3 | | V | 1.5 KΩ to 3.0 V or GND, PD = high. |
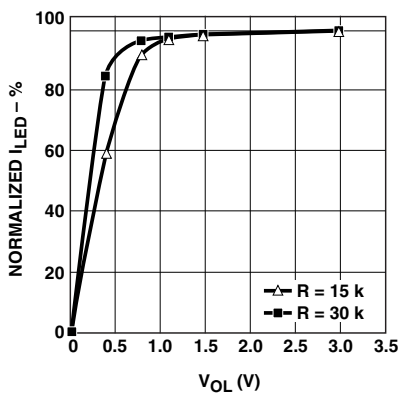


**Figure 11. Typical I-V characteristic of ADNS-2051 XY_LED pin**

## Typical LED Current Table

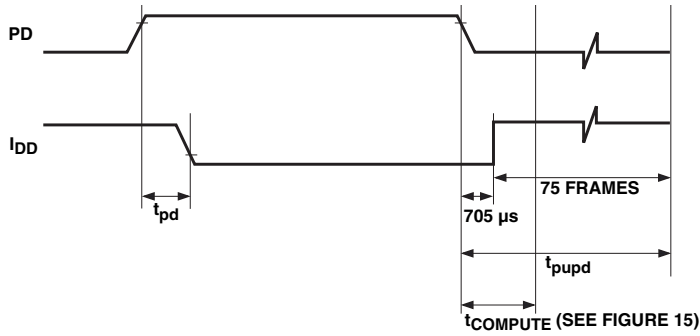| R1 Value | kΩ | 15 | 18 | 22 | 27 | 33 | 37 |
|---|---|---|---|---|---|---|---|
| LED current (typical) | mA | 42 | 35 | 29 | 23 | 19 | 17 |

11

## PD Pin Timing



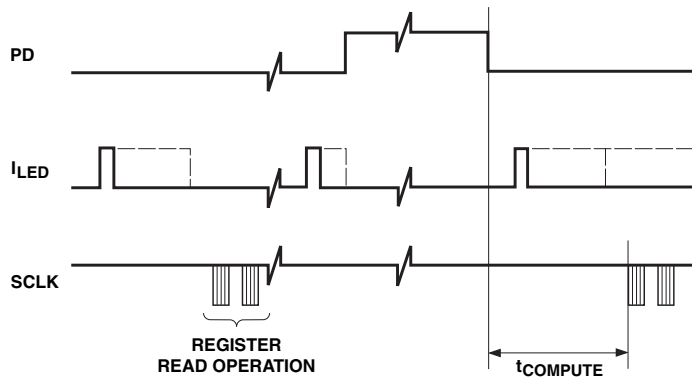**Figure 12. PD timing normal mode**
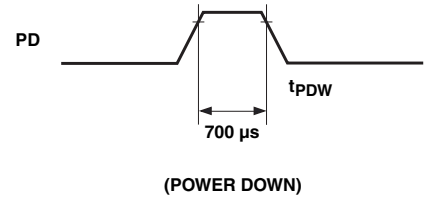


**Figure 13. PD timing sleep mode**



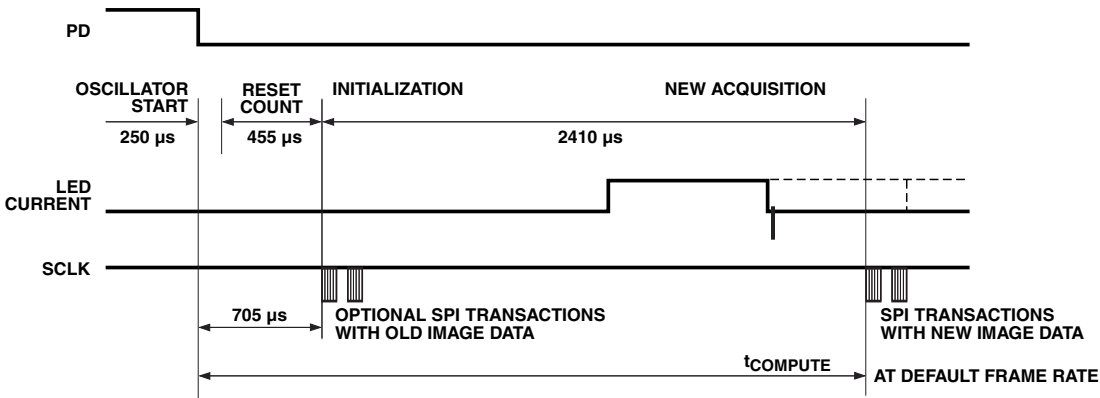**Figure 14. PD minimum pulse width**



**Figure 15. Detail of PD falling edge timing**

## Quadrature Mode Timing

The output waveforms emulate the output from encoders. With the resolution set to 400 cpi, from one to five quadrature states can exist within one frame time. The minimum state time is 133 µs. If the resolution is 800 cpi, then up to ten quadrature states can exist within a frame time. If the motion within a frame is greater than these values, the extra motion will be reported in the next frame. The following diagrams (see Figures 16, 17, and 18) show the timing for positive X motion, to the right or positive Y motion, up. If a power down via the PD pin occurs during a transfer, the transfer will resume after PD is de-asserted. The timing for that quadrature state will be increased by the length of the PD time.
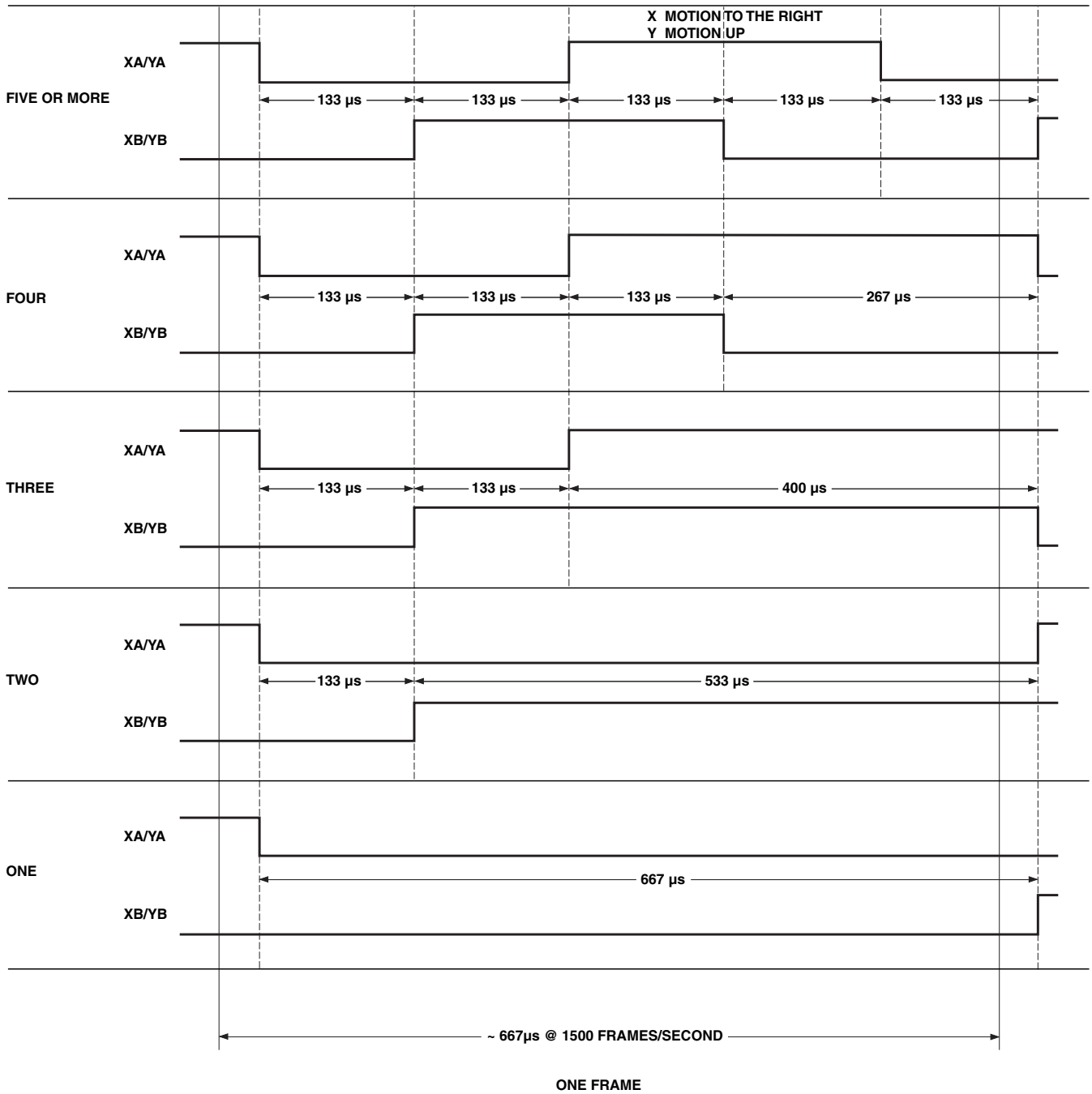
12

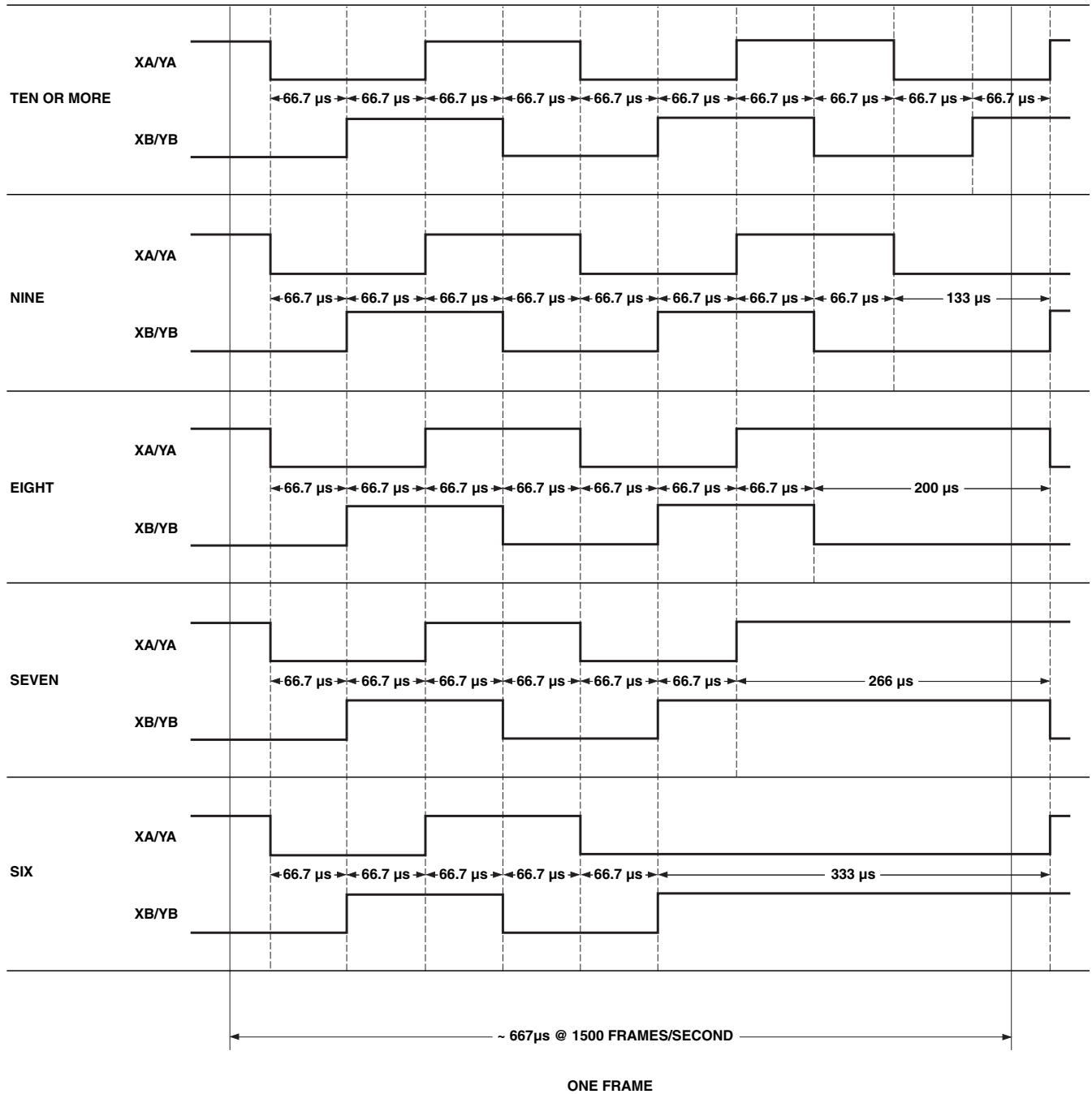**Figure 16. Quadrature states per frame (400 cpi mode)**

13

Figure 17. Quadrature states per frame (800 cp imode)

14
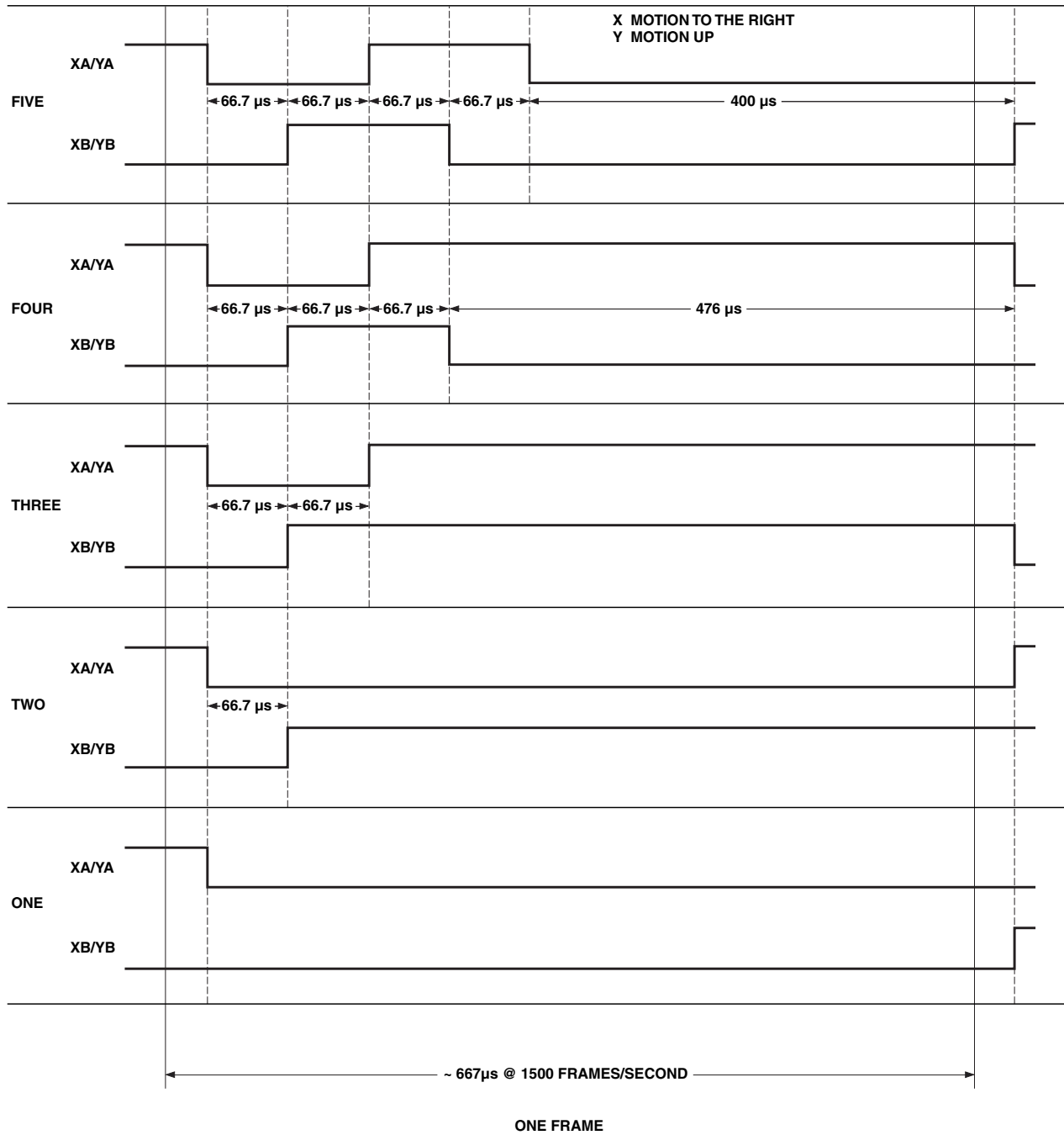
**Figure 18. Quadrature states per frame (800 cpi mode)**

15

## Quadrature State Machine

The following state machine shows the states of the quadrature pins. The two things to note are that while the PD pin is asserted, the state machine is halted. Once PD is de-asserted, the state machine picks up from where it left off. State 0 is entered after a power up reset.



| STATE | X AND Y OUTPUT | |
|---|---|---|
| | A | B |
| 0 | 0 | 0 |
| 1 | 0 | 1 |
| 2 | 1 | 0 |
| 3 | 1 | 1 |
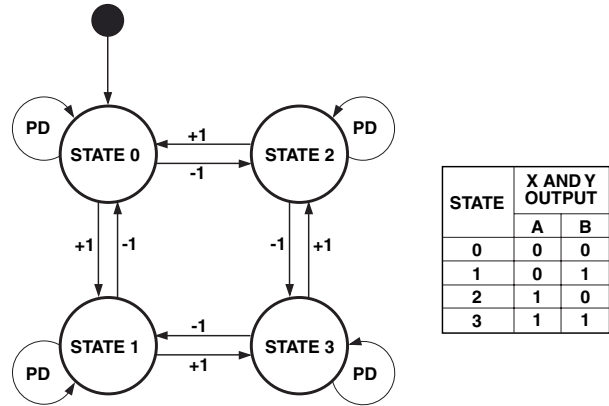
**Figure 19. Quadrature state machine**

## Quadrature Output Waveform

The two channel quadrature outputs are 5.0 volt CMOS outputs. The Δx count is used to generate the XA and XB signals, and Δy count is used for the YA and YB signals.



**Figure 20. Quadrature output waveform**

## Typical Performance Characteristics

Performance characteristics over recommended operating conditions. Typical values at 25°C, $V_{DD}$ = 5.0 V, 18 MHz.

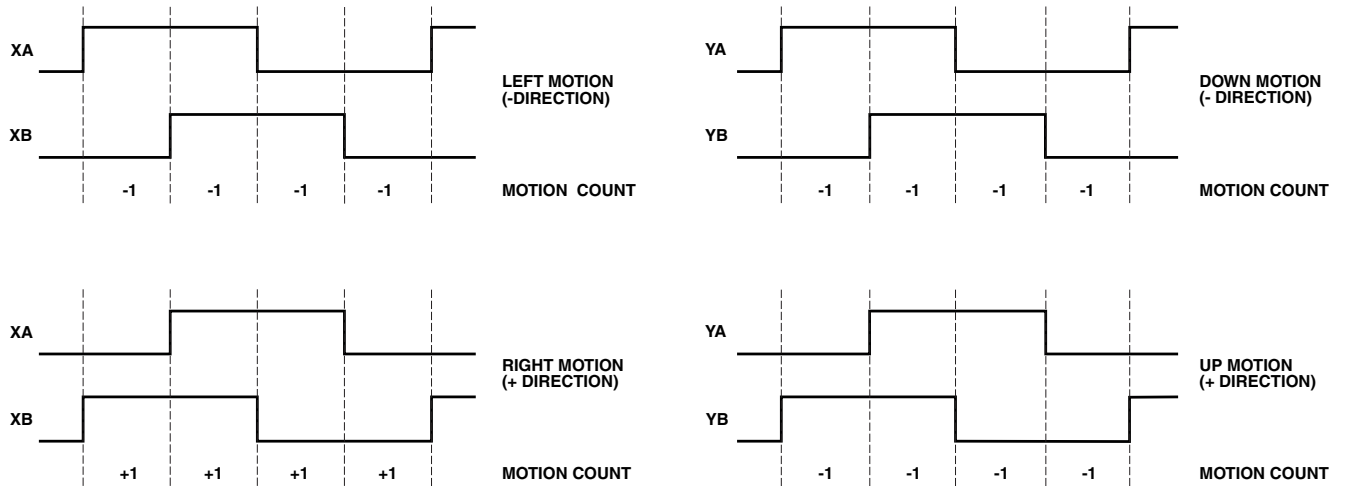| Parameter | Symbol | Min. | Typ. | Max. | Units | Notes |
|---|---|---|---|---|---|---|
| Path Error (Deviation) | $P_{ERROR}$ | | 0.5 | | % | Path Error (Deviation) is the error from the ideal cursor path. It is expressed as a percentage of total travel and is measured over standard surfaces. |

The following graphs (Figures 21, 22, 23, and 24) are the typical performance of the ADNS-2051 sensor, assembled as shown in the 2D assembly drawing with the HDNS-2100 Lens/Prism, the HDNS-2200 clip, and the HLMP-ED80-XX000 LED (page 3, Figure 4).
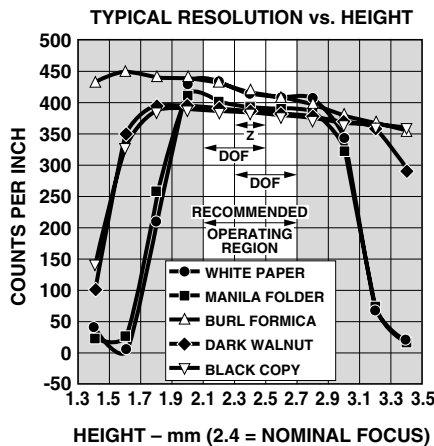


Figure 21. Typical resolution vs. Z (comparative surfaces)[2,3]
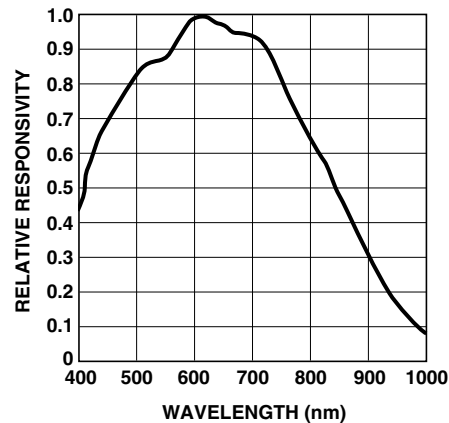


Figure 22. Wavelength responsivity[1]
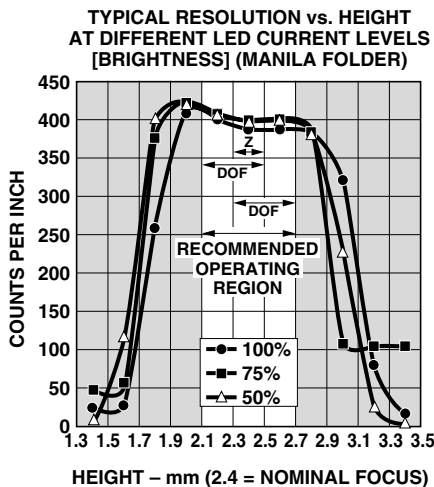


Figure 23. Typical resolution vs. z (manila folder and LED variation)[2,3]
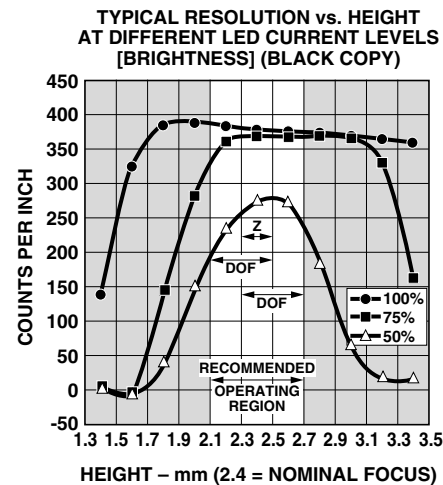


Figure 24. Typical resolution vs. z (black copy and LED variation)[2,3]

Note:
1. The ADNS-2051 is designed for optimal performance when used with the HLMP-ED80-XX000 (red LED 639 nm). For use with other LED colors (i.e., blue, green), please consult factory. When using alternate LEDs, there may also be performance degradation and additional eye safety considerations.
2. Z = Distance from Lens Reference plane to Surface.
3. DOF = Depth of Field.

17

## Synchronous Serial Port

The synchronous serial port is used to set and read parameters in the ADNS-2051, and can be used to read out the motion information instead of the quadrature data pins.

The port is a two wire, half duplex port. The host micro-controller always initiates communication; the ADNS-2051 never initiates data transfers.

SCLK: The serial port clock. It is always generated by the master (the micro-controller).

SDIO: The data line.

PD: A third line is sometimes involved. PD (Power Down) is usually used to place the ADNS-2051 in a low power mode to meet USB suspend specification. PD can also be used to force re-synchronization between the micro-controller and the ADNS-2051 in case of an error.

## Write Operation

Write operations, where data is going from the micro-controller to the ADNS-2051, is always initiated by the micro-controller and consists of two bytes. The first byte contains the address (seven bits) and has a "1" as its MSB to indicate data direction. The second byte contains the data. The transfer is synchronized by SCLK. The micro-controller changes SDIO on falling edges of SCLK. The ADNS-2051 reads SDIO on rising edges of SCLK.
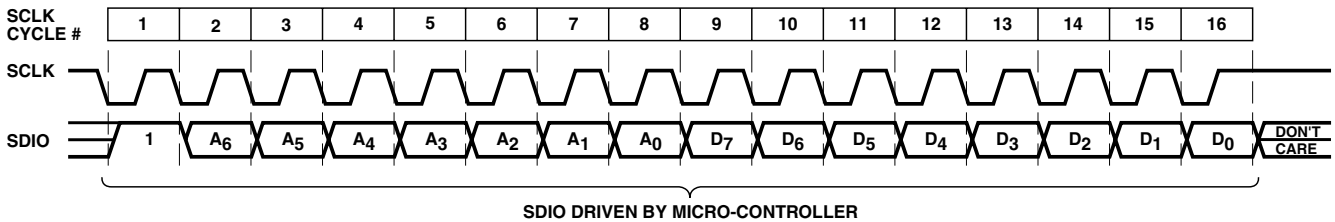


**Figure 25. Write operation**
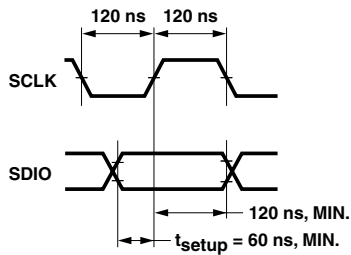


**Figure 26. SDIO setup and hold times SCLK pulse width**

18

## Read Operation

A read operation, which means that data is going from the ADNS-2051 to the micro-controller, is always initiated by the micro-controller and consists of two bytes. The first byte contains the address, is written by the micro-controller, and has a "0" as its MSB to indicate data direction. The second byte contains the data and is driven by the ADNS-2051. The transfer is synchronized by SCLK. SDIO is changed on falling edges of SCLK and read on every rising edge of SCLK. The micro-controller must go to a high Z state after the last address data bit. The ADNS-2051 will go to the high Z state after the last data bit (see detail "B" in Figure 28). One other thing to note during a read operation is that SCLK will need to be delayed after the last address data bit to ensure that the ADNS-2051 has at least 100 μs to prepare the requested data. This is shown in the timing diagrams below.
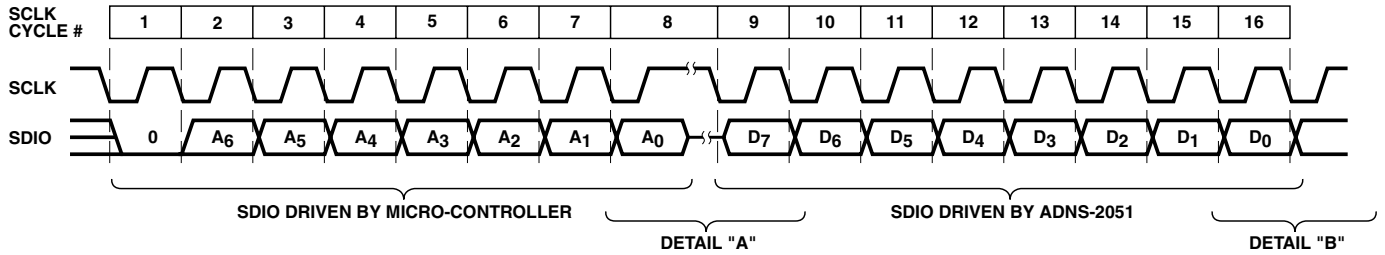
**Figure 27. Read operation**
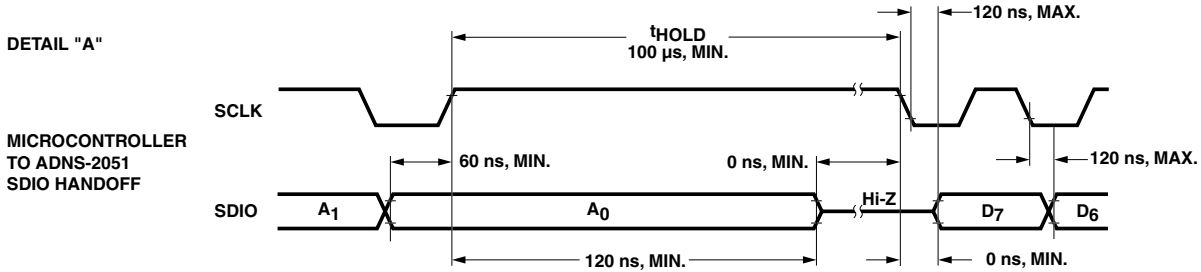
**Figure 28. Microcontroller to ADNS-2051 SDIO handoff**
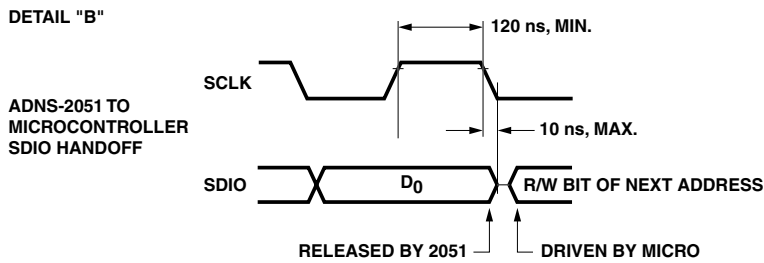
**Figure 29. ADNS-2051 to microcontroller SDIO handoff**

**Note:**

The 120 ns high state of SCLK is the minimum data hold time of the ADNS-2051. Since the falling edge of SCLK is actually the start of the next read or write command, the ADNS-2051 will hold the state of $D_0$ on the SDIO line until the falling edge of SCLK. In both write and read operations, SCLK is driven by the micro-controller.

Serial port communications is not allowed while PD (power down) is high. See "**Error Detection and Recovery**" regarding re-synchronizing via PD.

19

## Forcing the SDIO Line to the Hi-Z State

There are times when the SDIO line from the ADNS-2051 should be in the Hi-Z state. If the microprocessor has completed a write to the ADNS-2051, the SDIO line is Hi-Z, since the SDIO pin is still configured as an input. However, if the last operation from the microprocessor was a read, the ADNS-2051 will hold the D0 state on SDIO until a falling edge of SCLK.

To place the SDIO pin into the Hi-Z state, raise the PD pin for 100 μs (min). The PD pin can stay high, with the ADNS-2051 in the shutdown state, or the PD pin can be lowered, returning the ADNS-2051 to normal operation. The SDIO line will now be in the Hi-Z state.
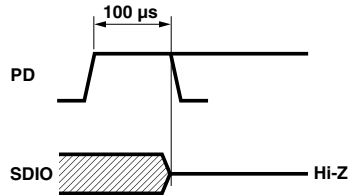


Figure 30. SDIO Hi-Z state and timing

## Required Timing between Read and Write Commands (tsxx)

There are minimum timing requirements between read and write commands on the serial port.

If the rising edge of the SCLK for the last data bit of the second write command occurs before the 100 microsecond required delay, then the first write command may not complete correctly.
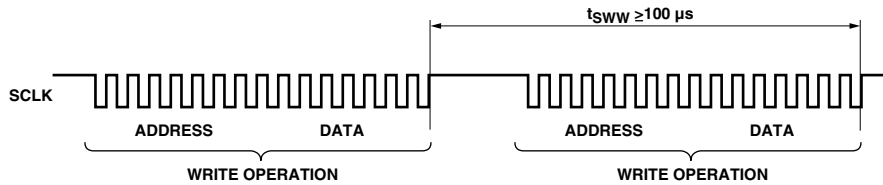


Figure 31. Timing between two write commands

If the rising edge of SCLK for the last address bit of the read command occurs before the 100 microsecond required delay, then the write command may not complete correctly.
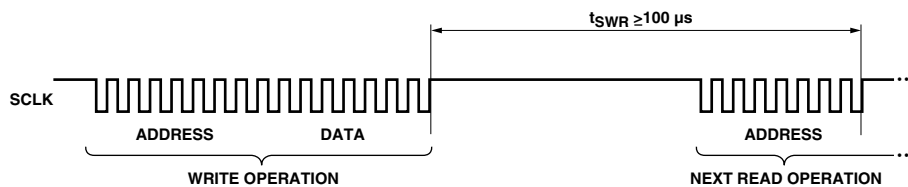


Figure 32. Timing between write and read commands

20

The falling edge of SCLK for the first address bit of either the read or write command must be at least 120 ns after the last SCLK rising edge of the last data bit of the previous read operation.
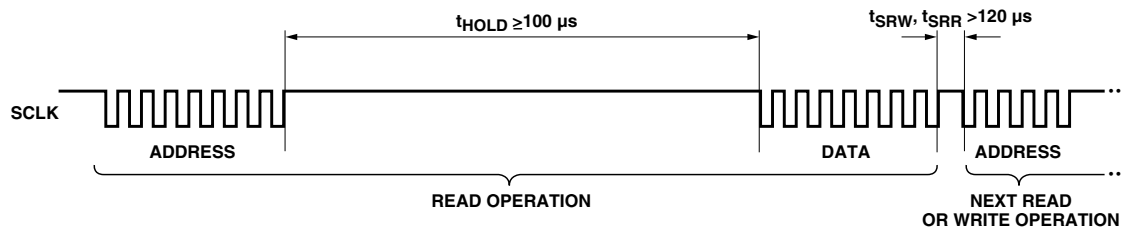


**Figure 33. Timing between read and either write or subsequent read commands**
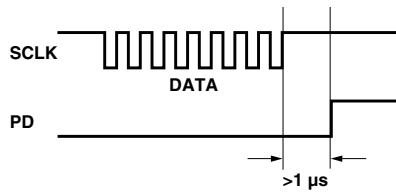


**Figure 34. Timing between SCLK and PD rising edge**

21

## Error Detection and Recovery

1. The ADNS-2051 and the micro-controller might get out of synchronization due to ESD events, power supply droops or micro-controller firmware flaws. In such a case, the micro-controller should raise PD for 100 µs. The ADNS-2051 will reset the serial port but will not reset the registers and be prepared for the beginning of a new transmission.

2. The ADNS-2051 has a transaction timer for the serial port. If the sixteenth SCLK rising edge is spaced more than approximately 0.9 seconds from the first SCLK edge of the current transaction, the serial port will reset.

3. Invalid addresses:

   – Writing to an invalid address will have no effect. Reading from an invalid address will return all zeros.

4. Collision detection on SDIO

   – The only time that the ADNS-2051 drives the SDIO line is during a READ operation. To avoid data collisions, the micro-controller should relinquish SDIO before the falling edge of SCLK after the last address bit. The ADNS-2051 begins to drive SDIO after the next rising edge of SCLK. The ADNS-2051 relinquishes SDIO within 120 ns of the falling SCLK edge after the last data bit. The micro-controller can begin driving SDIO any time after that. In order to maintain low power consumption in normal operation or when the PD pin is pulled high, the micro-controller should not leave SDIO floating until the next transmission (although that will not cause any communication difficulties).

5. In case of synchronization failure, both the ADNS-2051 and the micro-controller may drive SDIO. The ADNS-2051 can withstand 30 mA of short circuit current and will withstand infinite duration short circuit conditions.

6. Termination of a transmission by the micro-controller may sometimes be required (for example, due to a USB suspend interrupt during a read operation). To accomplish this the micro-controller should raise PD. The ADNS-2051 will not write to any register and will reset the serial port (but nothing else) and be prepared for the beginning of future transmissions after PD goes low.

7. The micro-controller can verify success of write operations by issuing a read command to the same address and comparing written data to read data.

8. The micro-controller can verify the synchronization of the serial port by periodically reading the product ID register.
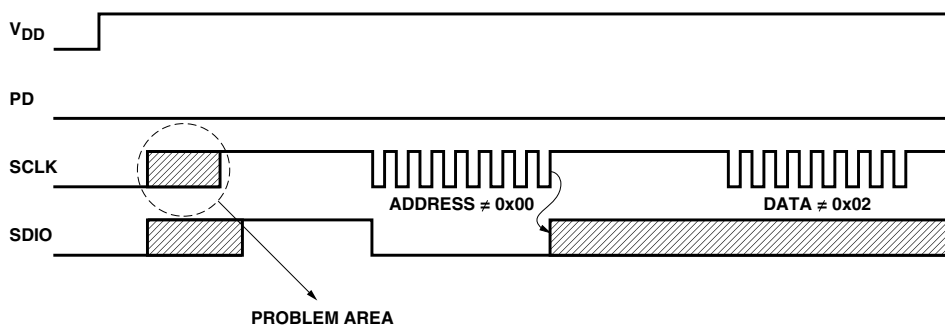


**Figure 35. Power up serial port watchdog timer sequence**

22

## Notes on Power up and the Serial Port

The sequence in which $V_{DD}$, PD, SCLK, and SDIO are set during powerup can affect the operation of the serial port. The diagram below shows what can happen shortly after powerup when the microprocessor tries to read data from the serial port.

This diagram shows the $V_{DD}$ rising to valid levels, at some point the microcontroller starts its program, sets the SCLK and SDIO lines to be outputs, and sets them high. It then waits to ensure that the ADNS-2051 has powered up and is ready to communicate. The microprocessor then tries to read from location 0x00, Product_ID, and is expecting a value of 0x02. If it receives this value, it then knows that the communication to the ADNS-2051 is operational.

The problem occurs if the ADNS-2051 powers up before the microprocessor sets the SCLK and SDIO lines to be outputs and high. The ADNS-2051 sees the raising of the SCLK as a valid rising edge, and clocks in the state of the SDIO as the first bit of the address (sets either a read or a write depending upon the state).

In the case of SDIO low, then a read operation has started. When the microprocessor begins to actually send the address, the ADNS-2051 already has the first bit of an address. When the seventh bit is sent by the micro, the ADNS-2051 has a valid address, and drives the SDIO line high within 120 ns (see detail "A" in Figure 27 and Figure 28). This results in a bus fight for SDIO. Since the address is wrong, the data sent back will be incorrect.

In the case of SDIO high, a write operation is started. The address and data are out of synchronization, and the wrong data will be written to the wrong address.
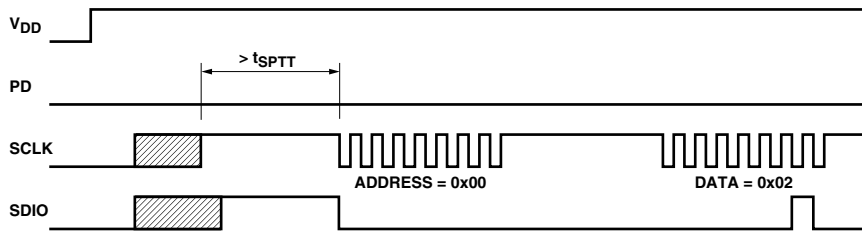


**Figure 36. Power up serial port watchdog timer sequence**
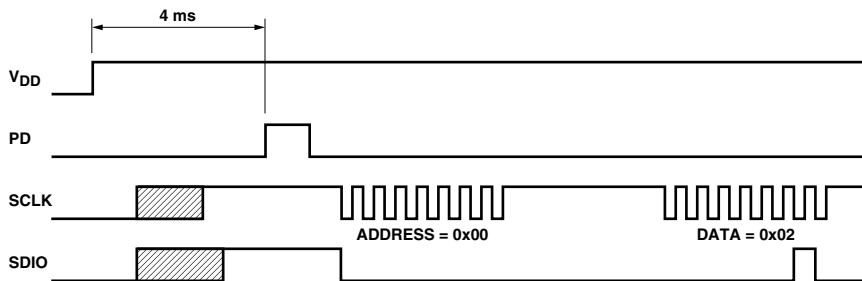


**Figure 37. Power up serial port PD sync sequence**

## Two Solutions

There are two different ways to solve the problem, waiting for the serial port watchdog timer to time out, or using the PD line to reset the serial port.

### 1. Serial Port Watchdog Timer Timeout

If the microprocessor waits at least $t_{SPTT}$ from $V_{DD}$ valid, it will ensure that the ADNS-2051 has powered up and the watchdog timer has timed out. This assumes that the microprocessor and the ADNS-2051 share the same power supply. If not, then the microprocessor must wait $t_{SPTT}$ from ADNS-2051 $V_{DD}$ valid. Then when the SCLK toggles for the address, the ADNS-2051 will be in sync with the microprocessor.

### 2. PD Sync

The PD line can be used to resync the serial port. If the microprocessor waits for 4 ms from $V_{DD}$ valid, and then outputs a valid PD pulse (see Figure 15), then the serial port will be ready for data.

### Resync Note

If the microprocessor and the ADNS-2051 get out of sync, then the data either written or read from the registers will be incorrect. An easy way to solve this is to output a PD pulse to resync the parts after an incorrect read.

23

## SPI communication code for the Cypress CY7C63000 or CY7C63001
(Please consult factory for the CY7C63722 or CY7C63723 codes.)

**Note:** This programming sequence is not covered in Avago's product warranty. It is only a recommended example when using the  mentioned Cypress microcontrollers. For the latest updates on Cypress microcontrollers, please contact Cypress at email:  usbapps@cypress.com or call (858) 613-7929 (US).

The following code can be used to implement the SPI data communications.  See the schematic in Figure 9.

```
;       Notes:
;       CY7C63001 20pin package
;       ADNS-2051
;       SDIO line connected to pin5 (P1.0)
;       PD connected to pin 16 (P1.1)
;       SCLK line connected to pin15 (P1.3)
;       I/O port
Port1_Data:             equ    01h                   ;    GPIO data port 1
Port1_Interrupt:        equ    05h                   ;    Interrupt enable for port 1
Port1_Pullup:           equ    09h                   ;    Pullup resistor control for port 1
;
;       Port bit definitions
SDIO:                   equ    01h                   ;    bit 0
PD:                     equ    02h                   ;    bit 1
SCLK:                   equ    08h                   ;    bit 3
Pt1_Current:            equ    00h                   ;    port1 current setting
;
;       GPIO Isink registers
Port1_Isink:            equ    38h
Port1_Isink0:           equ    38h
Port1_Isink1:           equ    39h
Port1_Isink3:           equ    3Bh
;
;
;       data memory variables
spi_addr:               equ    40h                   ;    address of spi writes
spi_data:               equ    41h                   ;    data of spi writes
bit_counter:            equ    44h                   ;    SPI bit counter
port1_wrote:            equ    45h                   ;    what we wrote last
;
;
;       initialize Port 1
;
                            mov A, Pt1_Current    ;    select DAC setting
                        iowr Port1_Isink0         ;    isink current Port 1 bit[0]
                        iowr Port1_Isink1         ;    isink current Port 1 bit[1]
                        iowr Port1_Isink3         ;    isink current Port 1 bit[3]
                        mov A, 0h                 ;    enable Port 1 bit [7:0] pullups
                        iowr Port1_Pullup

                        mov A, ~(PD|SDIO)         ;    turn on the ADNS-2051
                        mov [port1_wrote], A
                        mov A, [port1_wrote]
                        iowr Port1_Data           ;    PD low, SCLK, SDIO

                        mov A, 0
                        iowr Port1_Interrupt      ;    disable port 1 interrupts
;       There are possible problems with the SPI port if the microcontroller starts executing
;       instructionsbefore the ADNS-2051 sensor has powered up.  See page 18 for details.
;       It is assumed that power to the microcontroller is OK if the next instructions can be executed.
;       These instructions will reset the SPI port of the sensor.
```

24

```
Resync_sensor:          mov A,~(SCLK|SDIO|PD)   ;   set the SCLK, SDIO and PD lines low
                        and [port1_wrote], A
                        mov A, [port1_wrote]
                        iowr Port1_Data
                                                ;   If the power to the sensor needs more time
                                                ;   to stabilize, insert a delay here
                        call delay700us         ;   wait about 4 milliseconds for the sensor
                        call delay700us         ;   oscillator to stabilize
                        call delay700us
                        call delay700us
                        call delay700us
                        call delay700us
                        mov A, (SCLK|SDIO|PD     ;   set the SCLK, SDIO and PD lines high
                        or [port1_wrote], A      ;   this shuts down the oscillator and
                        mov A, [port1_wrote]     ;   resets the SPI port
                        iowr Port1_Data
                        call delay700us          ;   wait for the PD to reset the part
                        mov A, ~PD               ;   set the PD line low to put the sensor
                        and [port1_wrote], A     ;   back into normal operation
                        mov A, [port1_wrote]
                        iowr Port1_Data
                        call delay700us          ;   wait about 4 milliseconds for the sensor
                        call delay700us          ;   oscillator to stabilize
                        call delay700us
                        call delay700us
                        call delay700us
                        call delay700us          ;   sensor SPI port now in sync
;
;   ReadSPI routine
;
;   Includes delays for long traces or cables between the uP and ADNS-2051
;   Has correct timing of SCLK and SDIO
;
;   On entry:           spi_addr = Address of SPI register in the ADNS-2051
;   spi_data = undefined
;
;   On exit             spi_addr = undefined
;   spi_data = register contents from ADNS-2051
;

ReadSPI:                mov A, 64               ;   wait 200us (optional)
                        mov [bit_counter], A    ;   (about 3us per loop)
Waitrspi:               nop
                        nop
                        nop
                        nop
                        nop
                        nop
                        dec [bit_counter]
                        jnz Waitrspi

                                                ;   read address
                        mov A,~80h
                        and [spi_addr], A       ;   lower MSB of address (read)
                        call writeaddr
                                                ;   wait 200us (about 3us per loop)(100us minimum required)
                        mov A,64                ;   wait for data to be ready
                        mov [bit_counter], A
Waitrspi2:              nop
                        nop
                        nop
                        nop
                        nop
                        nop
                        dec [bit_counter]
                        jnz Waitrspi2
                        mov A, 0h               ;   clear the data
```

25

```
                    mov [spi_data], A
                    mov A, 08h
                    mov [bit_counter], A
                    mov A, SDIO
                    or [port1_wrote], A
                    mov A, [port1_wrote]        ;   write a 1 to SDIO
                    iowr Port1_Data
nextr:              mov A, ~SCLK                ;   lower SCLK
                    and [port1_wrote], A
                    mov A, [port1_wrote]
                    iowr Port1_Data
                    nop                         ;   wait for cable to settle
                    nop                         ;   if ADNS-2051 is connected to
                    nop                         ;   IC via short PCB traces,
                    nop                         ;   then the number of NOPs can
                    nop                         ;   reduced or eliminated
                    nop
                    nop
                    mov A,[spi_data]            ;   shift next bit
                    asl
                    mov [spi_data], A           ;   shift next bit
                    iord Port1_Data             ;   read SDIO
                    and A, SDIO
                    jz rdx
rd1:                mov A, 01h
                    or [spi_data], A
rdx:                mov A, SCLK                 ;   raise SCLK
                    or [port1_wrote], A
                    mov A, [port1_wrote]
                    iowr Port1_Data
                    nop                         ;   wait for cable to settle
                    nop
                    nop
                    nop
                    nop
                    nop
                    nop
                    dec [bit_counter]
                    jnz nextr
                    ret
;
;   WriteSPI routine
;
;
;   Includes delays for long traces or cables between the uP and ADNS-2051.
;   Has correct timing of SCLK and SDIO
;
;   On entry:          spi_addr = Address of SPI register in the ADNS-2051
;   spi_data = Data to be written to the SPI register
;
;   On exit            spi_addr = undefined
;   spi_data = undefined
;
WriteSPI:           mov A, 64                   ;   wait 200us (optional)
                    mov [bit_counter], A        ;   about 3us per loop
Waitspi:            nop
                    nop
                    nop
                    nop
                    nop
                    nop
                    dec [bit_counter]
                    jnz Waitspi
                                                ;   write address
```

```
                        mov A, 80h
                        or [spi_addr], A          ;   set MSB of address (write)
                        call writeaddr
                        jmp wrdata
writeaddr:              mov A, 08h                ;   8 bits to shift out
                        mov [bit_counter], A
nexta:                  mov A, ~SCLK              ;   lower SCLK
                        and [port1_wrote], A
                        mov A, [port1_wrote]
                        iowr Port1_Data
                        mov A, [spi_addr]         ;   shift next bit
                        asl
                        mov [spi_addr], A
                        jnc addr0
addr1:                  mov A, SDIO
                        or [port1_wrote], A       ;   raise SDIO
                        jmp addrx
addr0:                  mov A, ~SDIO
                        and [port1_wrote], A      ;   lower SDIO
addrx:                  mov A, [port1_wrote]
                        iowr Port1_Data
                        nop                       ;   wait for cable to settle
                        nop
                        nop
                        nop
                        nop
                        nop
                        nop
                        mov A, SCLK               ;   raise SCLK
                        or [port1_wrote], A
                        mov A, [port1_wrote]
                        iowr Port1_Data           ;   ADNS-2051 reads the address bit
                        nop                       ;   wait for cable to settle
                        nop
                        nop
                        nop
                        nop
                        nop
                        nop
                        dec [bit_counter]
                        jnz nexta
                        ret
wrdata:                 mov A, 08h                ;   8 bits of data
                        mov [bit_counter], A
nextw:                  mov A, ~SCLK              ;   lower SCLK
                        and [port1_wrote], A
                        mov A, [port1_wrote]
                        iowr Port1_Data
                        mov A, [spi_data]         ;   shift next bit
                        asl
                        mov [spi_data], A
                        jnc wr0
wr1:                    mov A, SDIO
                        or [port1_wrote], A       ;   raise SDIO
                        jmp wrx
wr0:                    mov A, ~SDIO
                        and [port1_wrote], A      ;   lower SDIO
wrx:                    mov A, [port1_wrote]
                        iowr Port1_Data
                        nop                       ;   wait for cable to settle
                        nop
                        nop
                        nop
                        nop
                        nop
                        nop
```

27

```
                    mov A, SCLK                 ; raise SCLK
                    or [port1_wrote], A
                    mov A, [port1_wrote]
                    iowr Port1_Data             ;   ADNS-2051 reads the data bit
                    nop                         ;   wait for cable to settle
                    nop
                    nop
                    nop
                    nop
                    nop
                    nop
                    dec [bit_counter]
                    jnz nextw
                    ret
delay700us:
                    mov A, ffh                  ;   wait for 710us
                    mov [bit_counter], A        ;   reuse bit_counter
waitd0:
                    nop
                    nop
                    nop
                    nop
                    nop
                    nop                         ;   2us
                    dec [bit_counter]
                    jnz waitd0
                    ret
```

## Example calling syntax

```
;
;   WriteSPI
;
;   Set register 0a to 40h, LED blink mode
                    mov A, 0ah                  ;   move address into A
                    mov [spi_addr], A           ; move address into spi_addr
                    mov A, 40h                  ;   move data into A
                    mov [spi_data], A           ;   move data into spi_data
                    call WriteSPI               ;   call WriteSPI routine, on return,
                                                ;   spi_addr and spi_data will be undefined
;
;   ReadSPI
;
;   Read register 02h, the motion register
                    mov A, 02h                  ;   move address into A
                    mov [spi_addr], A           ;   move address into spi_addr
                    call ReadSPI                ;   call ReadSPI, on return, data is in spi_data,
                                                ;   spi_addr is undefined
```

28

## Registers

The ADNS-2051 can be programmed through registers, via the serial port, and configuration and motion data can be read from these registers.

| Address | Register | Address | Register | Address | Register |
|---------|----------|---------|----------|---------|----------|
| 0x00 | Product_ID | 0x06 | Average_Pixel | 0x0c | Data_Out_Lower |
| 0x01 | Revision_ID | 0x07 | Maximum_Pixel | 0x0d | Data_Out_Upper |
| 0x02 | Motion | 0x08 | Reserved | 0x0e | Shutter_Lower |
| 0x03 | Delta_X | 0x09 | Reserved | 0x0f | Shutter_Upper |
| 0x04 | Delta_Y | 0x0a | Configuration_bits | 0x10 | Frame_Period_Lower |
| 0x05 | SQUAL | 0x0b | Reserved | 0x11 | Frame_Period_Upper |

**Product_ID**                                   Address: 0x00
Access: Read                                     Reset Value: 0x02

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-----|---|---|---|---|---|---|---|---|
| Field | $PID_7$ | $PID_6$ | $PID_5$ | $PID_4$ | $PID_3$ | $PID_2$ | $PID_1$ | $PID_0$ |

Data Type: Eight bit number with the product identifier.

USAGE: The value in this register does not change, it can be used to verify that the serial communications link is OK.

**Revision_ID**                                  Address: 0x01
Access: Read                                     Reset Value: 0xNN

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-----|---|---|---|---|---|---|---|---|
| Field | $RID_7$ | $RID_6$ | $RID_5$ | $RID_4$ | $RID_3$ | $RID_2$ | $RID_1$ | $RID_0$ |

Data Type: Eight bit number with current revision of the IC.

USAGE: NN is a value between 00 and FF which represent the current design revision of the device.

**Motion**                                       Address: 0x02
Access: Read                                     Reset Value: 0x00

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-----|---|---|---|---|---|---|---|---|
| Field | MOT | Reserved | FAULT | OVFY | OVFX | Reserved | Reserved | RES |

Data Type: Bit field

USAGE: Register 0x02 allows the user to determine if motion has occurred since the last time it was read. If so, then the user should read registers 0x03 and 0x04 to get the accumulated motion. It also tells if the motion buffers have overflowed and whether or not an LED fault occurred since the last reading. The current resolution is also shown.

29

| Field Name | Description |
|---|---|
| MOT | Motion since last report or PD<br>**0 = No motion**<br>1 = Motion occurred, data ready for reading in Delta_X and Delta_Y registers |
| Reserved | Reserved for future |
| FAULT | LED Fault detected – set when R_BIN is too low or too high, shorts to $V_{DD}$ or Ground<br>**0 = No fault**<br>1 = Fault detected |
| OVFY | Motion overflow Y, $\Delta Y$ buffer has overflowed since last report<br>**0 = No overflow**<br>1 = Overflow has occurred |
| OVFX | Motion overflow X, $\Delta X$ buffer has overflowed since last report<br>**0 = No overflow**<br>1 = Overflow has occurred |
| Reserved | Reserved for future |
| Reserved | Reserved for future |
| RES | Resolution in counts per inch<br>**0 = 400**<br>1 = 800 |

**Notes for Motion:**

1. Reading this register freezes the Delta_X and Delta_Y register values. Read this register before reading the Delta_X and Delta_Y registers. If Delta_X and Delta_Y are not read before the motion register is read a second time, the data in Delta_X and Delta_Y will be lost.

2. Avago RECOMMENDS that registers 0x02, 0x03 and 0x04 be read sequentially.

3. Internal buffers can accumulate more than eight bits of motion for X or Y. If either one of the internal buffers overflows, then absolute path data is lost, and the OVFX or OVFY bit is set. These bits (OVFX and OVFY) are cleared once some motion has been read from the Delta_X and Delta_Y registers, and if the buffers are not at full scale. Since more data is present in the buffers, the cycle of reading the Motion, Delta_X and Delta_Y registers should be repeated until the motion bit (MOT) is cleared. Until MOT is cleared, either the Delta_X or Delta_Y registers will read either positive or negative full scale. If the motion register has not been read for a long time, at 400 cpi it may take up to 16 read cycles to clear the buffers, at 800 cpi, up to 32 cycles.

4. FAULT is a sticky bit that is cleared by reading the Motion register. It signifies that an LED fault has occurred since the last time the motion register was read. Once an LED fault has cleared, the hardware will drive the LED normally.

**Delta_X**                                          Address: 0x03
Access: Read                                       Reset Value: 0x00

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Field | $X_7$ | $X_6$ | $X_5$ | $X_4$ | $X_3$ | $X_2$ | $X_1$ | $X_0$ |

Data Type: Eight bit 2's complement number.

USAGE: X movement is counts since last report. Absolute value is determined by resolution. Reading clears the register.

| MOTION | -128 | -127 | | -2 | -1 | 0 | +1 | +2 | | +126 | +127 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| DELTA_X | 80 | 81 | | FE | FF | 00 | 01 | 02 | | 7E | 7F |
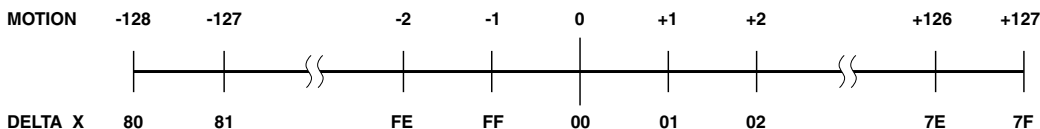
**Delta_Y**　　　　　　　　　　　　　　Address: 0x04
Access: Read　　　　　　　　　　　　Reset Value: 0x00

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-----|---|---|---|---|---|---|---|---|
| Field | $Y_7$ | $Y_6$ | $Y_5$ | $Y_4$ | $Y_3$ | $Y_2$ | $Y_1$ | $Y_0$ |

Data Type: Eight bit 2's complement number.

USAGE: Y movement is counts since last report. Absolute value is determined by resolution. Reading clears the register.
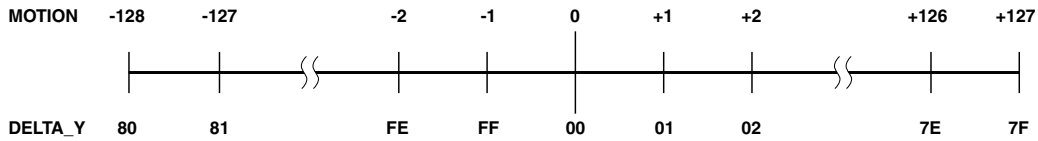
| MOTION | -128 | -127 | | -2 | -1 | 0 | +1 | +2 | | +126 | +127 |
|--------|------|------|--|----|----|---|----|----|--|------|------|
| DELTA_Y | 80 | 81 | | FE | FF | 00 | 01 | 02 | | 7E | 7F |

**Surface_Quality**　　　　　　　　　Address: 0x05
Access: Read　　　　　　　　　　　　Reset Value: 0x00

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-----|---|---|---|---|---|---|---|---|
| Field | $SQ_7$ | $SQ_6$ | $SQ_5$ | $SQ_4$ | $SQ_3$ | $SQ_2$ | $SQ_1$ | $SQ_0$ |

Data Type: Eight bit number.

USAGE: SQUAL is a measure of the number of features visible by the sensor in the current frame. The maximum value is 255. Since small changes in the current frame can result in changes in SQUAL, variations in SQUAL when looking at a surface are expected. The graph below shows 250 sequentially acquired SQUAL values, while a sensor was moved slowly over white paper. SQUAL is nearly equal to zero, if there is no surface below the sensor.

**SQUAL VALUES (WHITE PAPER)**
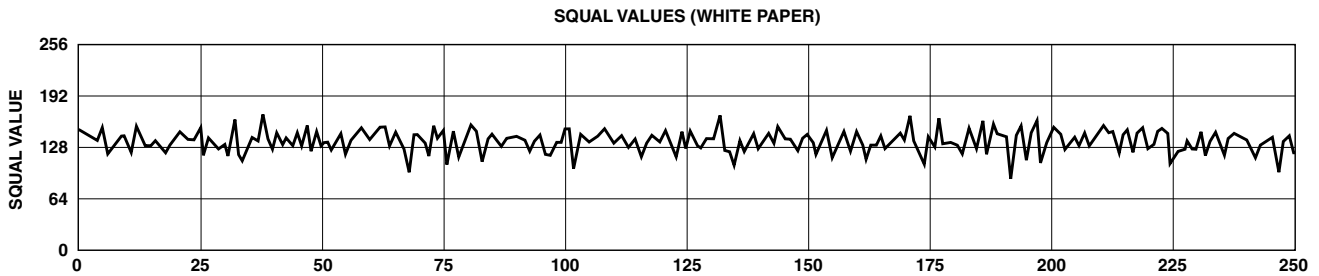
The focus point is important and could affect the squal value, the graph below showing another setup with various z-height. The graph clearly shows that the squal count is dependent on focus distance.

**Note:**
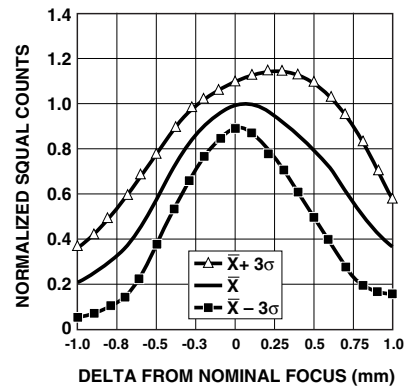This graph is obtained by getting multiple readings over different heights.

Figure 38. Typical mean squal vs. Z (white paper)

31

**Average_Pixel**                     Address: 0x06
Access: Read                          Reset Value: 0x00

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-----|---|---|---|---|---|---|---|---|
| Field | 0 | 0 | $AP_5$ | $AP_4$ | $AP_3$ | $AP_2$ | $AP_1$ | $AP_0$ |

Data Type: Six bit number.

USAGE: Average Pixel value in current frame. Minimum value = 0, maximum = 63. The average pixel value can be adjusted every frame. Shown below is a graph of 250 sequentially acquired average pixel values, while the sensor was moved slowly over white paper.
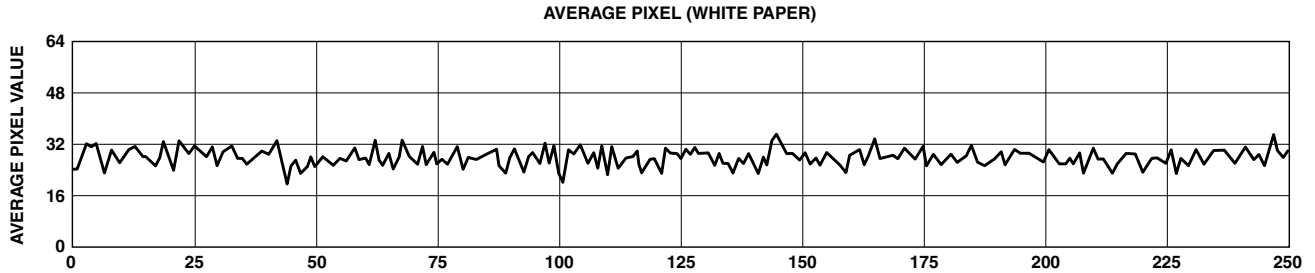


**Maximum_Pixel**                     Address: 0x07
Access: Read                          Reset Value: 0x00

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-----|---|---|---|---|---|---|---|---|
| Field | 0 | 0 | $MP_5$ | $MP_4$ | $MP_3$ | $MP_2$ | $MP_1$ | $MP_0$ |

Data Type: Six bit number.

USAGE: Maximum Pixel value in current frame. Minimum value = 0, maximum value = 63. The maximum pixel value can be adjusted every frame. Shown below is a graph of 250 sequentially acquired maximum pixel values, while the sensor was moved slowly over white paper.
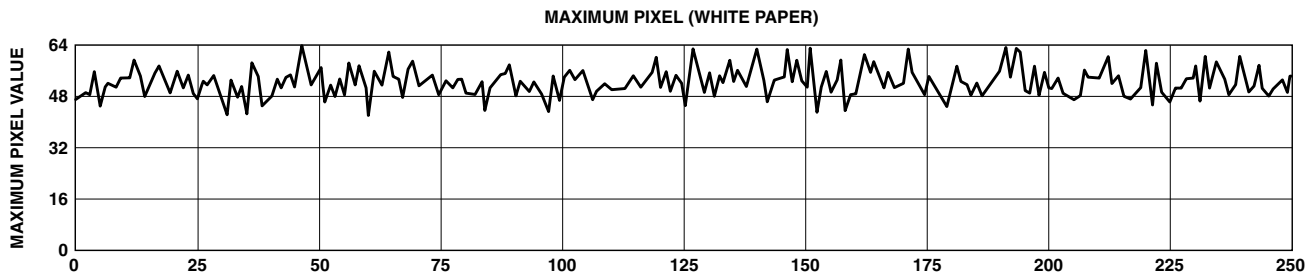


**Reserved**                          Address: 0x08
**Reserved**                          Address: 0x09

32

**Configuration_bits**  Address: 0x0a
Access: Read/Write  Reset Value: 0x00

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Field | RESET | LED_MODE | Sys Test | RES | PixDump | Reserved | Reserved | Sleep |

Data Type: Bit field

USAGE: Register 0x0a allows the user to change the configuration of the sensor. Shown below are the bits, their default values, and optional values.

| Field Name | Description |
|---|---|
| RESET | Power up defaults (bit always reads 0)<br>**0 = No effect**<br>1 = Reset registers and bits to power up default settings (bold entries) |
| LED_MODE | LED Shutter Mode<br>**0 = Shutter mode off (LED always on) (even if no motion up to 1 sec.)**<br>1 = Shutter mode on (LED only on when the electronic shutter is open) |
| Sys Test | System Tests (bit always reads 0)<br>**0 = No tests**<br>1 = perform all system tests, output 16 bit CRC via Data_Out_Upper and Data_Out_Lower registers.<br><br>*Note: Since part of the system test is a RAM test, the RAM will be overwritten with the default values when the test is done. If any configuration changes from the default are needed for operation, make the changes AFTER the system test is run. This operation requires substantially more time to complete than other register transactions.* |
| RES | Resolution in counts per inch<br>**0 = 400**<br>1 = 800 |
| Pix Dump | Dump the pixel array through Data_Out_Upper and Data_Out_Lower, 256 bytes<br>**0 = disabled**<br>1 = dump pixel array |
| Reserved | Reserved |
| Reserved | Reserved |
| Sleep | Sleep Mode<br>**0 = Normal, fall asleep after one second of no movement (1500 frames/s)**<br>1 = Always awake |

**Reserved**  Address: 0x0b

33

**Data_Out_Lower**                                   Address: 0x0c
Access: Read                                         Reset Value: undefined

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| Field | $DO_7$ | $DO_6$ | $DO_5$ | $DO_4$ | $DO_3$ | $DO_2$ | $DO_1$ | $DO_0$ |

**Data_Out_Upper**                                   Address: 0x0d
Access: Read                                         Reset Value: undefined

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| Field | $DO_{15}$ | $DO_{14}$ | $DO_{13}$ | $DO_{12}$ | $DO_{11}$ | $DO_{10}$ | $DO_9$ | $DO_8$ |

Data Type: Sixteen bit word.

USAGE: Data can be written to these registers from the system self test, or the pixel dump command. The data can be read out 0x0d, or 0x0d first, then 0x0c.

|  | Data_Out_Upper | Data_Out_Lower | Note |
|---|---|---|---|
| **System test result 1:** | FE | D4 | One of two results returned. These |
| **System test result 2:** | 4D | 10 | values are subject to change with |
|  |  |  | each device design revision. |
| **Pixel Dump command** | Pixel Address | Pixel Data (Lower 6 bits) |  |

Once the pixel dump command is given, the sensor writes the address and the value for the first pixel into the Data_Out_Upper and Data_Out_Lower registers. The MSB of Data_Out_Lower is the status bit for the data. If the bit is high, the data are NOT valid. Once the MSB is low, the data for that particular read are valid and should be saved. The pixel address and data will then be incremented on the next frame. Once the pixel dump is complete, the PixDump bit in register 0x0a should be set to zero. To obtain an accurate image, the LED needs to be turned on by changing the sleep mode of the configuration register 0x0a to always awake.

34

## Pixel Address Map

(Looking through the HDNS-2100 Lens)

**LAST PIXEL**

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| FF | EF | DF | CF | BF | AF | 9F | 8F | 7F | 6F | 5F | 4F | 3F | 2F | 1F | 0F |
| FE | EE | DE | CE | BE | AE | 9E | 8E | 7E | 6E | 5E | 4E | 3E | 2E | 1E | 0E |
| FD | ED | DD | CD | BD | AD | 9D | 8D | 7D | 6D | 5D | 4D | 3D | 2D | 1D | 0D |
| FC | EC | DC | CC | BC | AC | 9C | 8C | 7C | 6C | 5C | 4C | 3C | 2C | 1C | 0C |
| FB | EB | DB | CB | BB | AB | 9B | 8B | 7B | 6B | 5B | 4B | 3B | 2B | 1B | 0B |
| FA | EA | DA | CA | BA | AA | 9A | 8A | 7A | 6A | 5A | 4A | 3A | 2A | 1A | 0A |
| F9 | E9 | D9 | C9 | B9 | A9 | 99 | 89 | 79 | 69 | 59 | 49 | 39 | 29 | 19 | 09 |
| F8 | E8 | D8 | C8 | B8 | A8 | 98 | 88 | 78 | 68 | 58 | 48 | 38 | 28 | 18 | 08 |
| F7 | E7 | D7 | C7 | B7 | A7 | 97 | 87 | 77 | 67 | 57 | 47 | 37 | 27 | 17 | 07 |
| F6 | E6 | D6 | C6 | B6 | A6 | 96 | 86 | 76 | 66 | 56 | 46 | 36 | 26 | 16 | 06 |
| F5 | E5 | D5 | C5 | B5 | A5 | 95 | 85 | 75 | 65 | 55 | 45 | 35 | 25 | 15 | 05 |
| F4 | E4 | D4 | C4 | B4 | A4 | 94 | 84 | 74 | 64 | 54 | 44 | 34 | 24 | 14 | 04 |
| F3 | E3 | D3 | C3 | B3 | A3 | 93 | 83 | 73 | 63 | 53 | 43 | 33 | 23 | 13 | 03 |
| F2 | E2 | D2 | C2 | B2 | A2 | 92 | 82 | 72 | 62 | 52 | 42 | 32 | 22 | 12 | 02 |
| F1 | E1 | D1 | C1 | B1 | A1 | 91 | 81 | 71 | 61 | 51 | 41 | 31 | 21 | 11 | 01 |
| F0 | E0 | D0 | C0 | B0 | A0 | 90 | 80 | 70 | 60 | 50 | 40 | 30 | 20 | 10 | 00 |

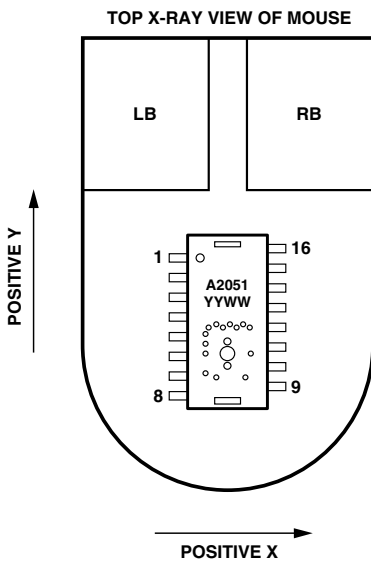**FIRST PIXEL**

**TOP X-RAY VIEW OF MOUSE**



**Figure 39. Directions are for a complete mouse, with the HDNS-2100 lens**
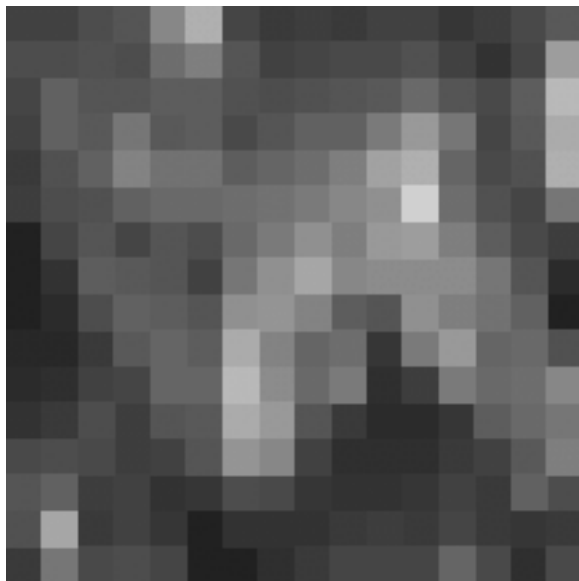
## Pixel Dump Pictures

The following images (Figure 40) are the output of the pixel dump command. The data ranges from zero for complete black, to 63 for complete white. An internal AGC circuit adjusts the shutter value to keep the brightest feature (max. pixel) in the mid 50s.
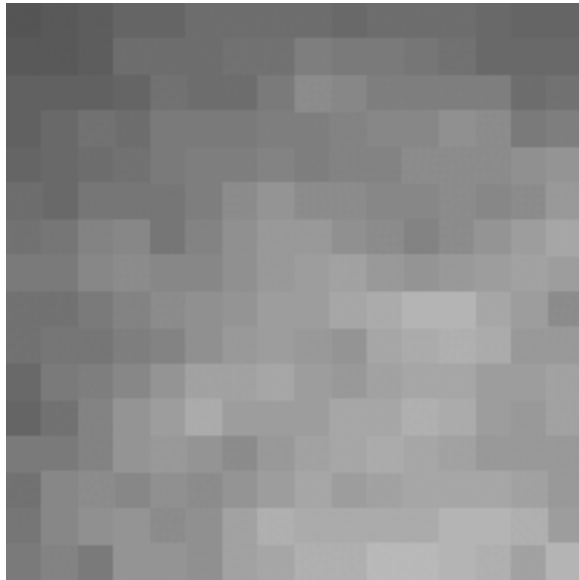


**(a) White Paper**
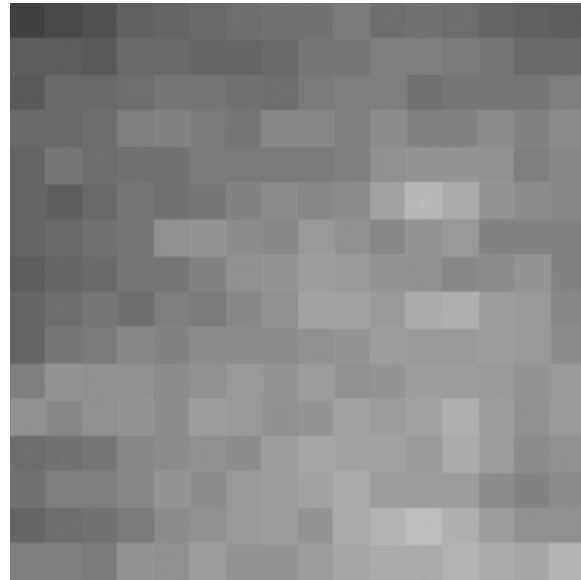


**(b) Manila Folder**



**(c) Neoprene Mouse Pad (Gray)**



**(d) USAF Test Chart Group 3, Element 1**
**8 line pairs per mm**

**Figure 40. Pixel dump pictures**

36

**Shutter_Lower**　　　　　　　　　　　　Address: 0x0e
Access: Read　　　　　　　　　　　　　　Reset Value: 0x64

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-----|---|---|---|---|---|---|---|---|
| Field | $S_7$ | $S_6$ | $S_5$ | $S_4$ | $S_3$ | $S_2$ | $S_1$ | $S_0$ |

**Shutter_Upper**　　　　　　　　　　　　Address: 0x0f
Access: Read　　　　　　　　　　　　　　Reset Value: 0x00

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-----|---|---|---|---|---|---|---|---|
| Field | $S_{15}$ | $S_{14}$ | $S_{13}$ | $S_{12}$ | $S_{11}$ | $S_{10}$ | $S_9$ | $S_8$ |

Data Type: Sixteen bit word.

USAGE: Units are clock cycles; default value is 64. Read Shutter_Upper first, then Shutter_Lower. They should be read consecutively. The shutter is adjusted to keep the average and maximum pixel values within normal operating ranges. The shutter value can be adjusted to a new value on every frame. When the shutter adjusts, it changes by ±1/16 of the current value. Shown below is a graph of 250 sequentially acquired shutter values, while the sensor was moved slowly over white paper.

**SHUTTER VALUES (WHITE PAPER)**
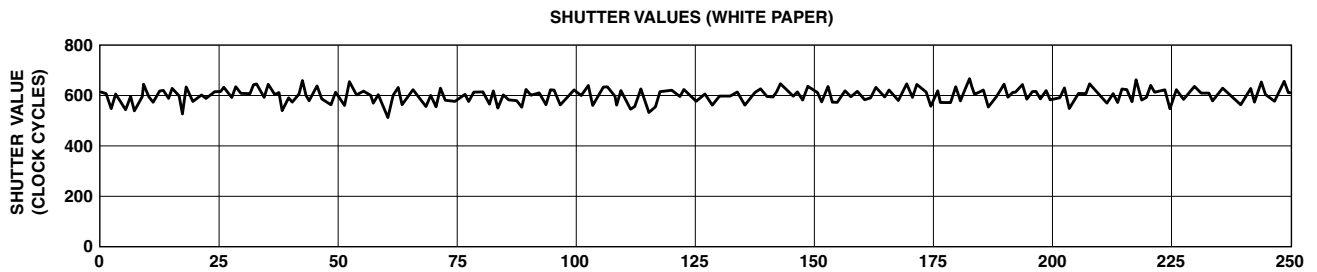


The focus point is important and could affect the shutter value. The graph below shows another setup with various z-height. This graph clearly shows that the shutter value is dependent on focus distance.
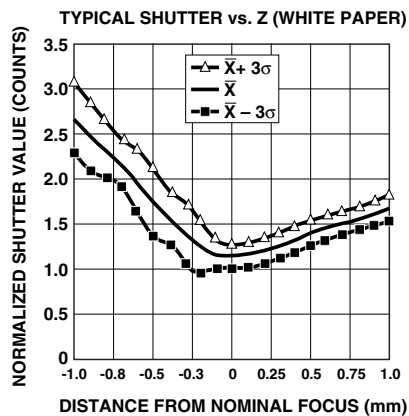
**TYPICAL SHUTTER vs. Z (WHITE PAPER)**



DISTANCE FROM NOMINAL FOCUS (mm)

**Figure 41. Typical shutter vs. Z (white paper)**

**Note:** This graph shows average readings over different heights.

37

The maximum value of the shutter is dependent upon the frame rate and clock frequency. The formula for the maximum shutter value is:

$$\text{Max. Shutter Value} = \frac{\text{Clock Frequency}}{\text{Frame Rate}} - 2816$$

For a clock frequency of 18 MHz, the following table shows the maximum shutter value. 1 clock cycle is 55.56 nsec.

| Frames/second | Max Shutter | | Shutter | |
| | Decimal | Hex | Upper | Lower |
| --- | --- | --- | --- | --- |
| 2300 | 5010 | 0x1392 | 13 | 92 |
| 2000 | 6184 | 0x1828 | 18 | 28 |
| 1500 | 9184 | 0x23E0 | 23 | E0 |
| 1000 | 15184 | 0x3B50 | 3B | 50 |
| 500 | 33184 | 0x81A0 | 81 | A0 |

← Default Max. Shutter (aligned to 1500 row)

**Frame_Period_Lower**          Address: 0x10
Access: Read/Write          Reset Value: 0x20

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Field | $FP_7$ | $FP_6$ | $FP_5$ | $FP_4$ | $FP_3$ | $FP_2$ | $FP_1$ | $FP_0$ |

**Frame_Period_Upper**          Address: 0x11
Access: Read/Write          Reset Value: 0xd1

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Field | $FP_{15}$ | $FP_{14}$ | $FP_{13}$ | $FP_{12}$ | $FP_{11}$ | $FP_{10}$ | $FP_9$ | $FP_8$ |

Data Type: Sixteen bit 2's complement word.

USAGE: The frame period counter counts up until it overflows. Units are clock cycles. The formula is:

$$\frac{\text{Clock Rate}}{\text{Frame Rate}} = \text{Counts (decimal)} \rightarrow \text{Counts (hex)} \rightarrow \text{Counts (2's complement hex)}$$

For an 18 MHz clock, here are the Frame_Period values for popular frame rates.

| Frames/second | Counts | | | Frame_Period | |
| | Decimal | Hex | 2's Comp | Upper | Lower |
| --- | --- | --- | --- | --- | --- |
| 2300* | 7826 | 0x1E92 | 0xE16E | E1 | 6E |
| 2000* | 9000 | 0x2328 | 0xDCD8 | DC | D8 |
| 1500 | 12000 | 0x2EE0 | 0xD120 | D1 | 20 |
| 1000 | 18000 | 0x4650 | 0xB9B0 | B9 | B0 |
| 500 | 36000 | 0x8CA0 | 0x7360 | 73 | 60 |

← Default Frame Time (aligned to 2000*/1500 rows)
← Minimum Frame Time (aligned to 1000 row)

**\*Note:**
To optimize tracking performance on dark surfaces, it is recommended that an adaptive frame rate based on shutter value be implemented, for frame rates greater than 1500. Changing the frame rate results in changes in the maximum speed, acceleration limits, and dark surface performance.
To read from the registers, read Frame_Period_Upper first followed by Frame_Period_Lower.
To write to the registers, write Frame_Period_Lower first followed by Frame_Period_Upper.

## IC Register State after Reset (power up or setting bit 7, register 0x0a)

| Address | Register | Value | Meaning |
| --- | --- | --- | --- |
| 0x0 | Product_ID | 0x02 | Product ID = 2 (Fixed value) |
| 0x01 | Revision_ID | 0xNN | Revision of IC (Fixed value). (For each device design revision.) |
| 0x02 | Motion | 0x00 | No Motion<br>LED = No Fault<br>No X data overflow<br>No Y data overflow<br>Resolution is 400 counts per inch |
| 0x03 | Delta_X | 0x00 | No X motion |
| 0x04 | Delta_Y | 0x00 | No Y motion |
| 0x05 | SQUAL | 0x00 | No image yet to measure |
| 0x06 | Average_Pixel | 0x00 | No image yet to measure |
| 0x07 | Maximum_Pixel | 0x00 | No image yet to measure |
| 0x08 | Reserved | — | |
| 0x09 | Reserved | — | |
| 0x0a | Configuration_bits | 0x00 | Part is not Reset<br>LED Shutter Mode is off<br>No System tests<br>Resolution = 400 counts per inch<br>Pixel Dump is disabled<br>Sleep mode is enabled |
| 0x0b | Reserved | — | |
| 0x0c | Data_Out_Lower | undefined | No data to read |
| 0x0d | Data_Out_Upper | undefined | No data to read |
| 0x0e | Shutter_Lower | 0x64 | Initial shutter value |
| 0x0f | Shutter_Upper | 0x00 | Initial shutter value |
| 0x10 | Frame_Period_Lower | 0x20 | Initial frame period value (corresponds to 1500 fps) |
| 0x11 | Frame_Period_Upper | 0xd1 | Initial frame period value (corresponds to 1500 fps) |

## Optical Mouse Design References

Application Note AN1179

Eye Safety calculation AN1228

## Ordering Information

Specify part number as follows:

ADNS-2051 = Sensor IC in a 16-pin staggered DIP, 20 per tube.

HDNS-2100 = Round Optical Mouse Lens

HDNS-2100#001 = Trimmed Optical Mouse Lens

HDNS-2200 = LED Assembly Clip (Black)

HDNS-2200#001 = LED Assembly Clip (Clear)

HLMP-ED80-XX000 = LED

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

**AVAGO**
T E C H N O L O G I E S