# PMW3330DM-TZQU: Optical Gaming Navigation Sensor

## General Description:

PMW3330DM-TZQU is PixArt Imaging's mid-range gaming sensor which comprises of navigation sensor and IR LED integrated in a 16pin molded lead-frame DIP package. It provides best in class gaming experience with the enhanced features of high speed, high resolution, high accuracy and selectable lift detection height to fulfill professional gamers' need. The sensor comes with self-adjusting variable frame rate algorithm to enable wireless gaming application. It is designed to be used with LM19-LSI lens to achieve optimum performance.

## Key Features:

- Integrated 16 pin molded lead-frame DIP package with IR LED
- Operating Voltage: 1.8V - 2.1V
- Lift detection options
  - 2mm
  - 3mm
  - Manual Lift Cut Off Calibration
- High speed motion detection 150ips (typical) and acceleration 30g (max).
- Selectable resolutions up to 7200cpi with 50cpi step size
- Resolution error of 3% (typical)
- Four wire serial port interface (SPI)
- External interrupt output for motion detection
- Internal oscillator — no clock input needed
- Self-adjusting variable frame rate for optimum power performance in wireless application
- Customizable response time and downshift time for rest modes
- Angle snapping

## Applications:

- Wired and Wireless Optical gaming mice
- Integrated input devices
- Battery-powered input devices

## Key Sensor Parameters:

| Parameter | Value |
|---|---|
| Supply Voltage | 1.8V ~ 2.1V |
| Interface | 4 wire Serial Port Interface (SPI) |
| Tracking Speed | 150ips (typical) |
| Acceleration | 30g |
| CPI Resolution | Up to 7200 |
| Frame Rate | Up to 8000 fps |
| Operating Current | VDD = 2.0V<br>Run Avg.: 7.60mA<br>Rest1: 1.20mA<br>Rest2: 32uA<br>Rest3: 18uA |
| System Clock | 68MHz |
| Package Type | 16 pin molded lead-frame DIP package with integrated IR LED |
| Lens Magnification | 1:1 |

## Ordering Information:

| Part Number | Package Type |
|---|---|
| PMW3330DM-TZQU | 16pin-DIP |
| LM19-LSI | Lens |

Lead (Pb) Free
RoHS 6 fully compliant

## Contents

# 1.0 System Level Description

This section covers PMW3330DM-TZQU's guidelines and recommendations in term of sensor, lens & PCB assemblies.

## 1.1 Pin Configuration

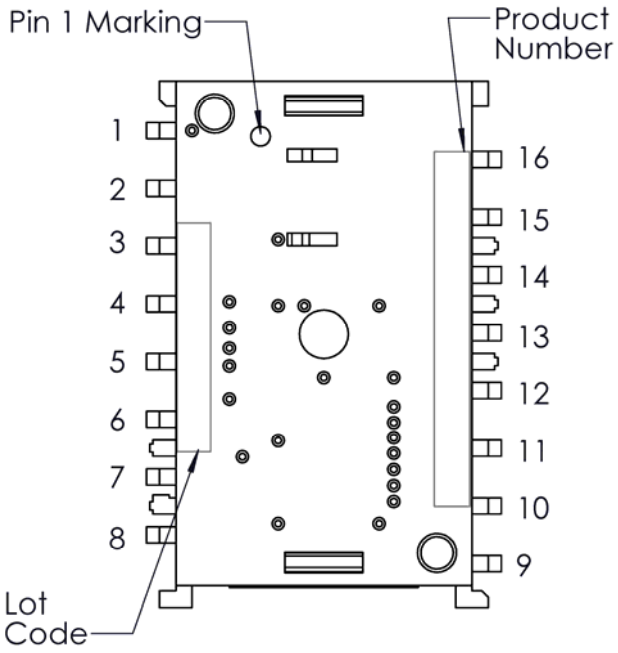



Figure 1. Device output pins

| Pin No. | Function | Symbol | Type | Description |
|---|---|---|---|---|
| 1 | NA | NC | NC | (Float) |
| 2 | NA | NC | NC | (Float) |
| 3 | Supply Voltage and I/O Voltage | VDDPIX | Power | LDO output for selective analog circuit |
| 4 | | VDD | Power | Input power supply |
| 5 | | VDDIO | Power | I/O reference voltage |
| 6 | NA | NC | NC | (Float) |
| 7 | Reset control | NRESET | Input | Chip reset(active low) |
| 8 | Ground | GND | GND | Ground |
| 9 | Motion Output | MOTION | Output | Motion detect |
| 10 | 4-wire SPI communication | SCLK | Input | Serial data clock |
| 11 | | MOSI | Input | Serial data input |
| 12 | | MISO | Output | Serial data output |
| 13 | | NCS | Input | Chip select(active low) |
| 14 | NA | NC | NC | (Float) |
| 15 | LED | LED_P | Input | LED Anode |
| 16 | NA | NC | NC | (Float) |

Table 1. PMW3330DM-TZQU Pin Description

| Items | Marking | Remark |
|---|---|---|
| Product Number | PMW3330DM-TZQU | |
| Lot Code | AYWWXXXXX | **A** : Assembly house<br>**Y**: Year<br>**WW**: Week<br>**XXXXX**: PixArt reference |

## 1.2 Package Outline Drawing

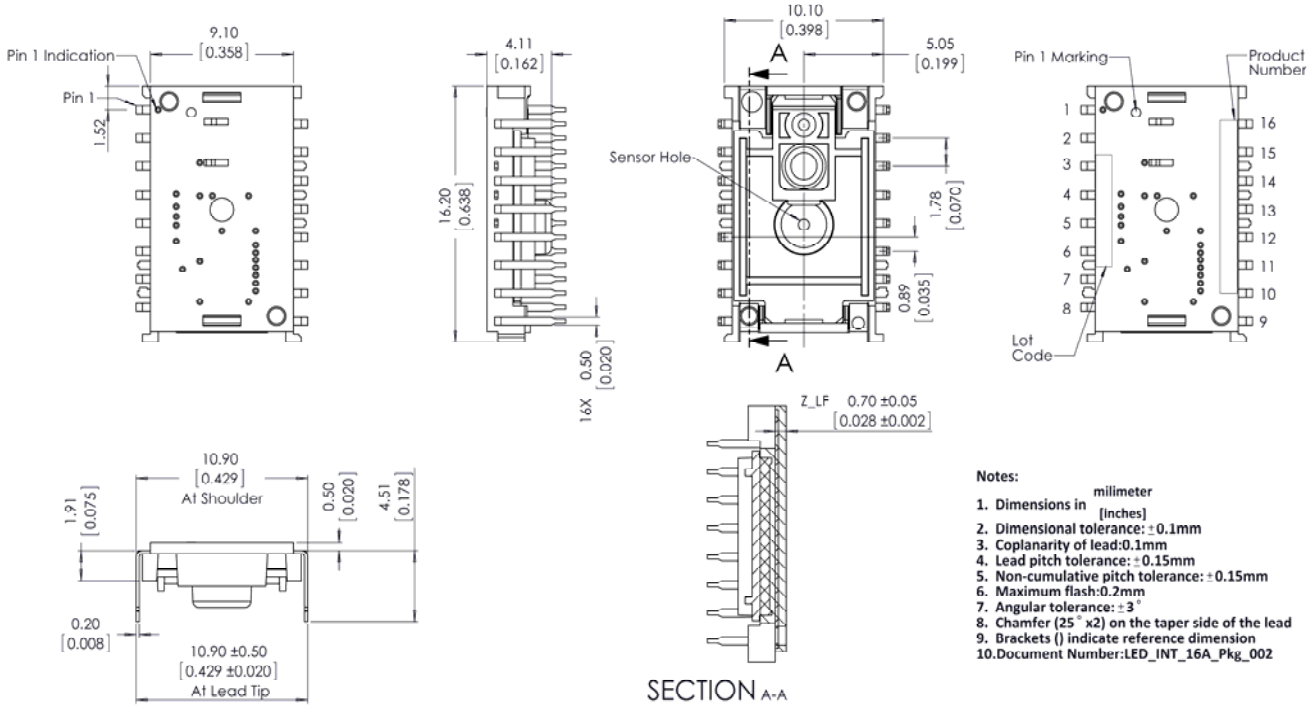



Figure 2. Packages Outline Drawing

*CAUTION: It is advised that normal static discharge precautions be taken in handling and assembling of this component to prevent damage and/or degradation which may be induced by ESD.*

## 1.3 Assembly Drawings

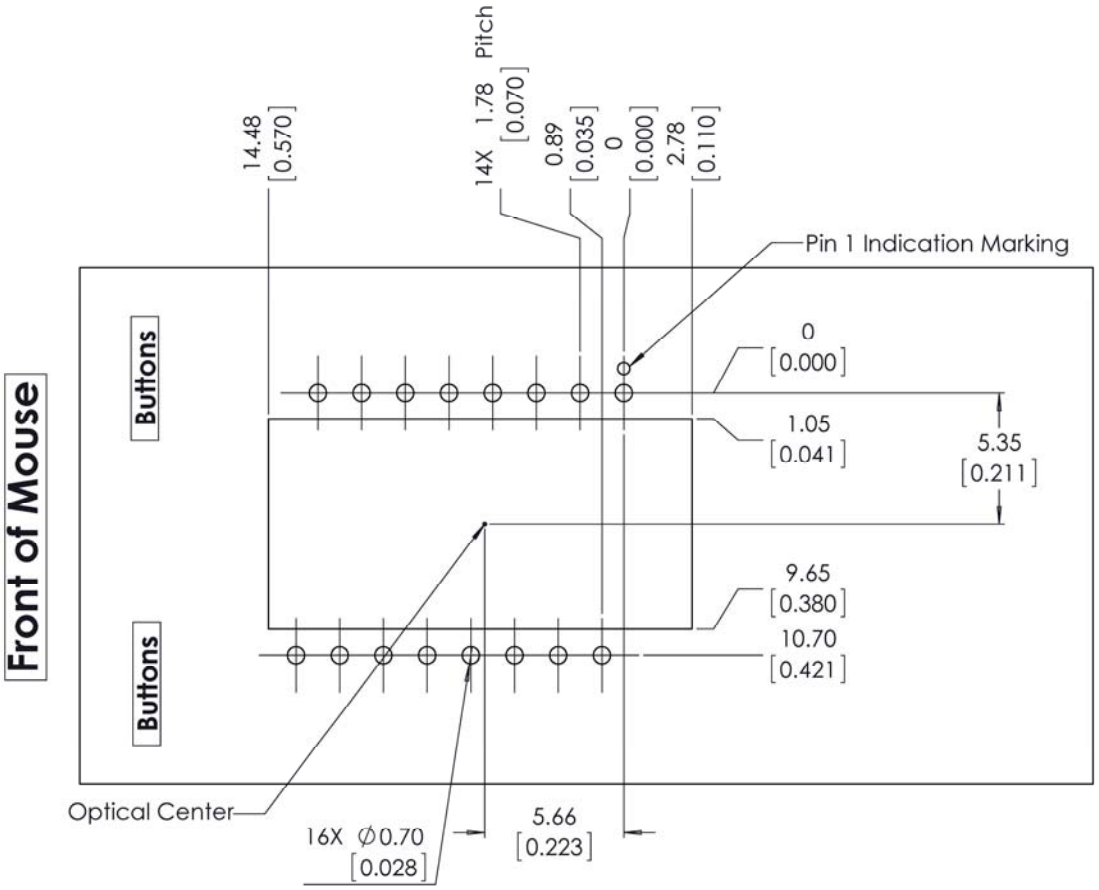



Figure 3. Recommended sensor orientation, mechanical cutouts and spacing (Top View)

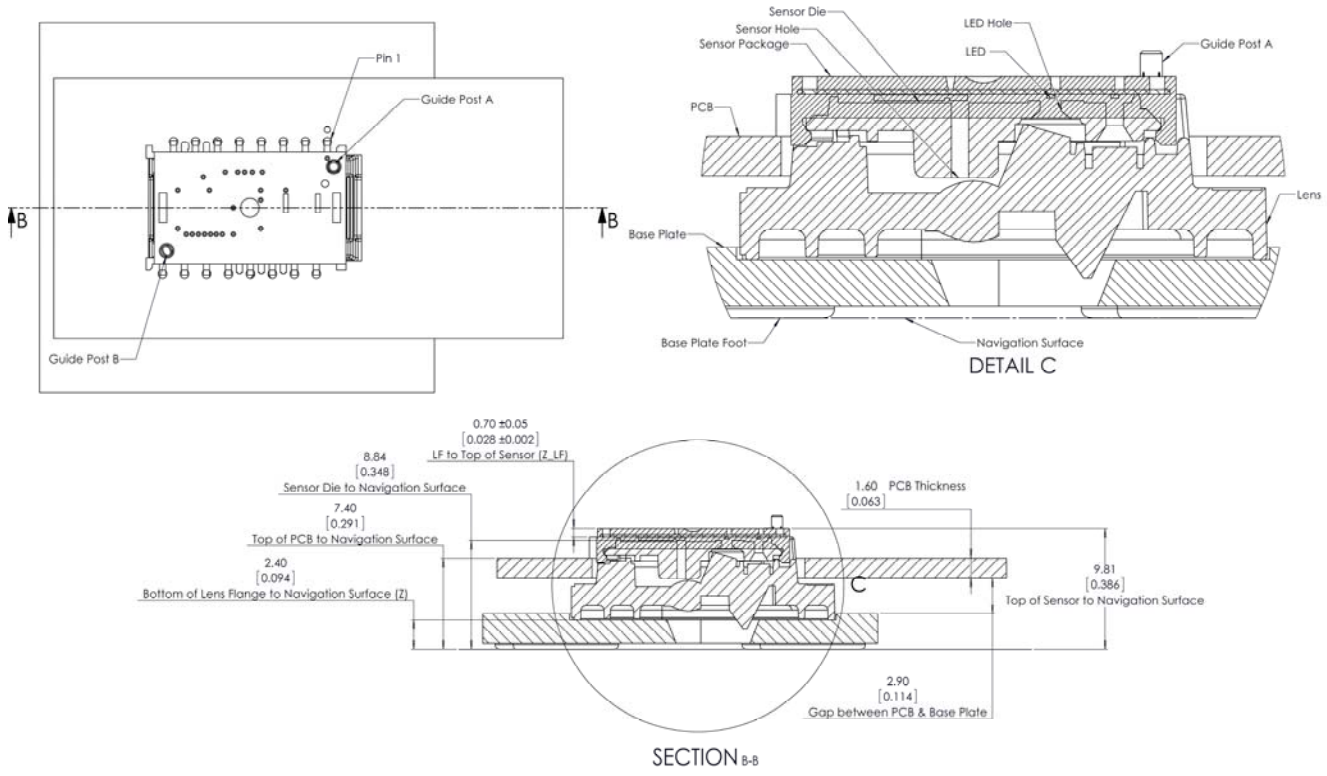



Figure 4. Assembly drawing of PMW3330DM-TZQU and distance from lens reference plane to tracking surface (Z)

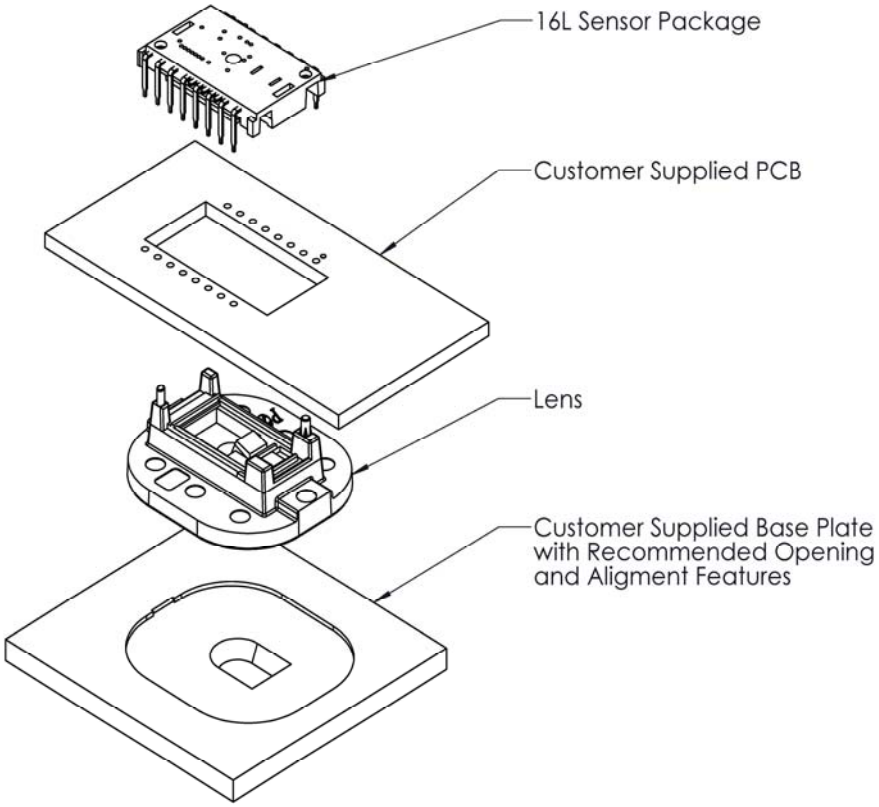



Figure 5. Exploded View of Assembly

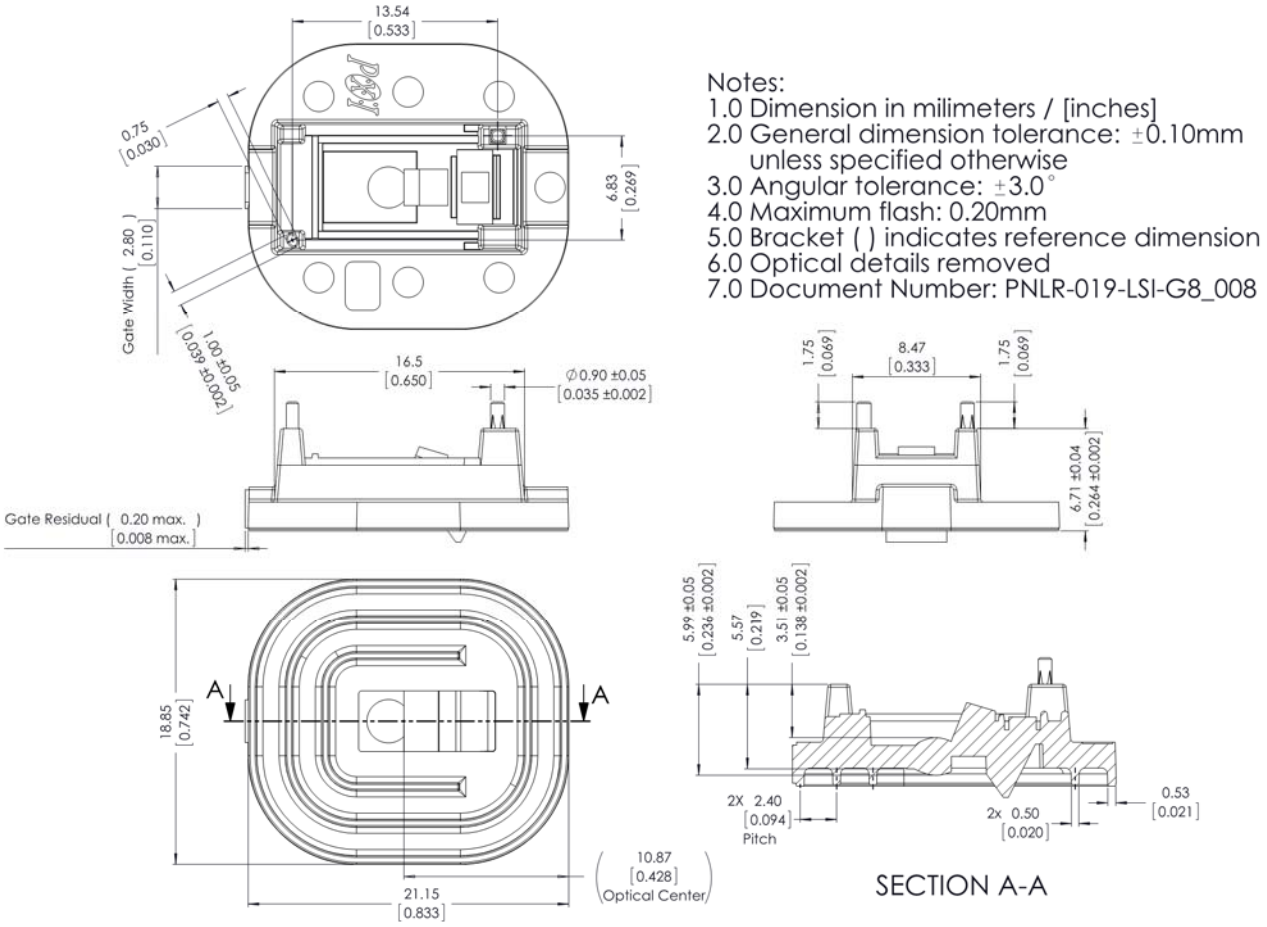



Figure 6. Lens Outline Drawing

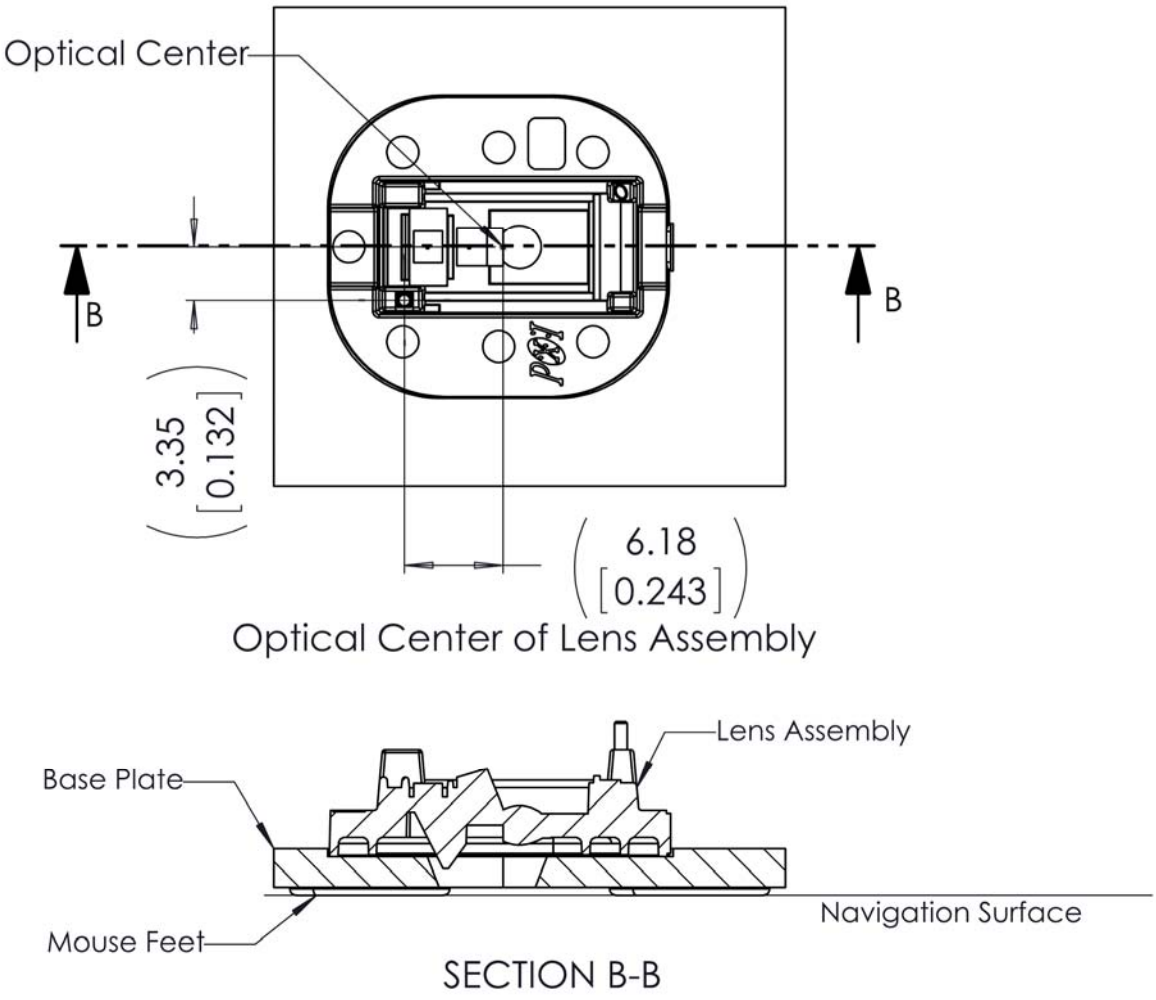



Figure 7. Cross section view of lens assembly

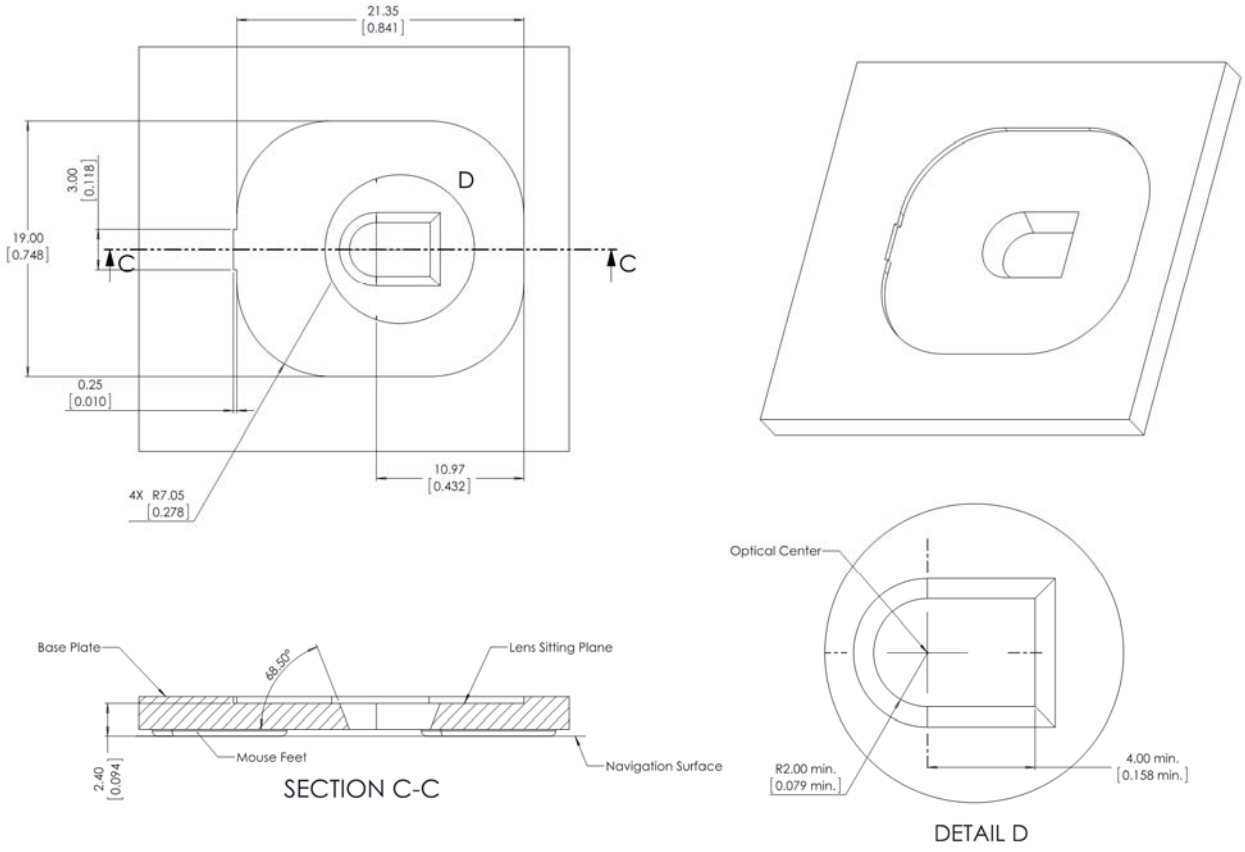



Figure 8. Recommended Base Plate Opening

*Note: Mouse feet should be placed close to the opening to stabilize the surface within the FOV of the sensor.*

## 1.4 PCB Assembly Recommendation

1) Insert the integrated sensor and all other electrical components into PCB.
2) Wave-solder the entire assembly in a no-wash solder process utilizing solder-fixture. A solder-fixture is required to protect the sensor from flux spray and wave solder.
3) Avoid getting any solder flux onto the sensor body as there is potential for flux to seep into the sensor package, the solder fixture should be designed to expose only the sensor leads to flux spray & molten solder while shielding the sensor body and optical apertures. The fixture should also set the sensor at the correct position and height on the PCB.
4) Place the lens onto the base plate. Care must be taken to avoid contamination on the optical surfaces.
5) Remove the protective kapton tapes from optical apertures of the sensor. Care must be taken to prevent Contaminants from entering the apertures. Do not place the PCB with the sensor facing up during the entire mouse assembly process. Hold the PCB vertically when removing kapton tape.
6) Insert PCB assembly over the lens onto the base plate aligning post to retain PCB assembly. The sensor package will self-align to the lens via the guide posts. 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.
7) Recommendation: The lens can be permanently secured to the sensor package by melting the lens' guide posts over the sensor with heat staking process.
8) Install mouse top case. There must be a feature in the top case to press down onto the PCB assembly to ensure all components are stacked or interlocked to the correct vertical height.

Sensor Block Diagram

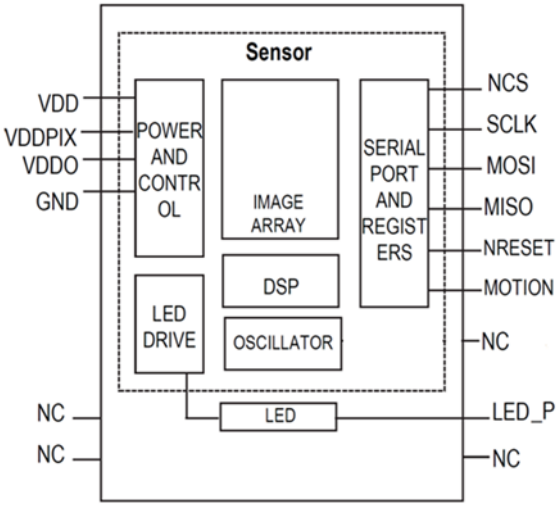



Figure 9. Block diagram of PMW3330DM-TZQU

## 1.5 Reference Schematics

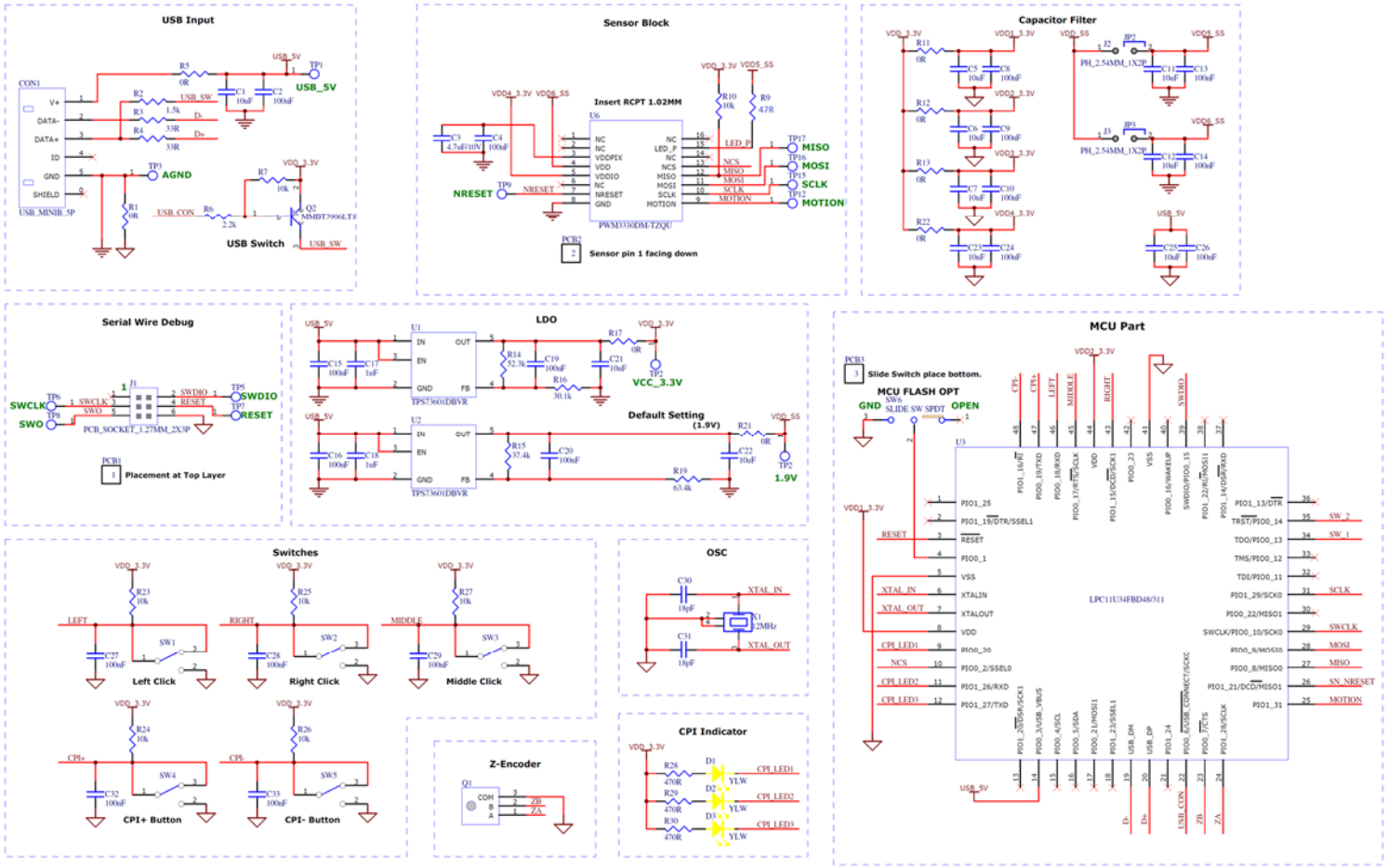



Figure 10. Schematic diagram for interface between PMW3330DM-TZQU and microcontroller on a wired solution

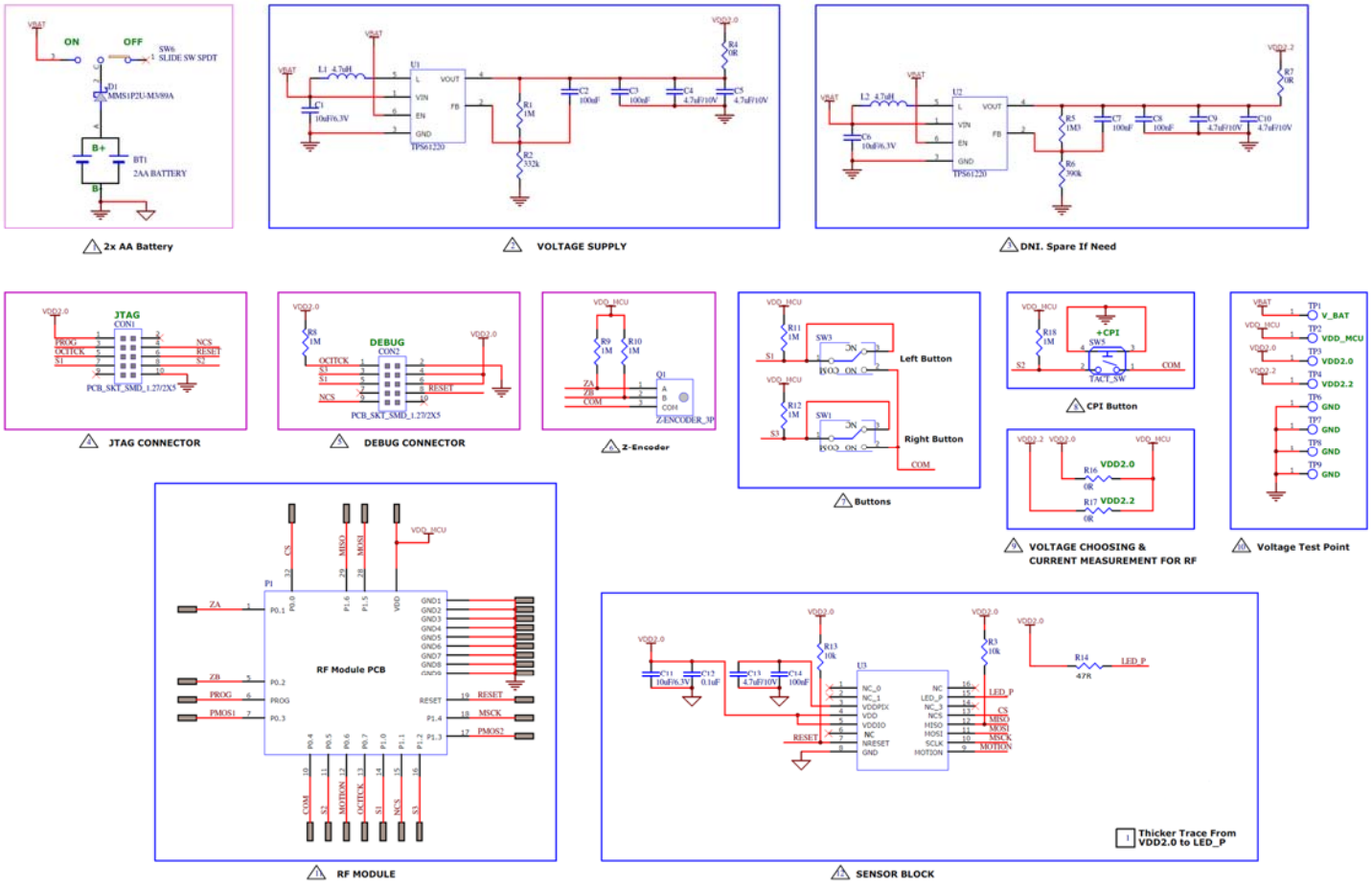



Figure 11. Schematic diagram for interface between PMW3330DM-TZQU and microcontroller in a wireless solution

# 2.0 Electrical Specifications

Regulatory Requirements

- Passes FCC "Part15, Subpart B, Class B", "CISPR 22 1997 Class B" and worldwide analogous emission limits when assembled into a mouse with shielded cable and following PixArt Imaging's recommendations.
- Passes IEC 62471: 2006 Photo biological safety of lamps and lamp systems

## 2.1 Absolute Maximum Ratings

Table 2: Absolute Maximum Ratings

| Parameter | Symbol | Minimum | Maximum | Units | Notes |
|---|---|---|---|---|---|
| Storage Temperature | $T_S$ | -40 | 85 | °C | |
| Lead Solder Temperature | $T_{SOLDER}$ | | 260 | °C | For 7 seconds, 1.6mm below seating plane. |
| Supply Voltage | $V_{DD}$ | -0.5 | 2.10 | V | |
| | $V_{DDIO}$ | -0.5 | 3.60 | V | |
| ESD (Human Body Model) | | | 2 | kV | All pins |
| Input Voltage | $V_{IN}$ | -0.5 | 3.6 | V | All I/O pins. |

## 2.2 Recommended Operating Conditions

Table 3: Recommended Operating Condition

| Parameter | Symbol | Min | Typ. | Max | Units | Notes |
|---|---|---|---|---|---|---|
| Operating Temperature | $T_A$ | 0 | | 40 | °C | |
| Power Supply Voltage | $V_{DD}$ | 1.80 | 2.0 | 2.10 | V | excluding supply noise |
| | $V_{DDIO}$ | 1.80 | 2.0 | 3.60 | V | excluding supply noise. (VDDIO must be same or greater than VDD) |
| Power Supply Rise Time | $t_{RT}$ | 0.15 | | 20 | ms | 0 to VDD min |
| Supply Noise (Sinusoidal) | $V_{NA}$ | | | 100 | mVp-p | 10 kHz —75 MHz |
| Serial Port Clock Frequency | $f_{SCLK}$ | | | 2.25 | MHz | 50% duty cycle |
| Distance from Lens Reference Plane to Tracking Surface | Z | 2.2 | 2.4 | 2.6 | mm | |
| Speed | S | | 150 | | ips | 220ips on QCK, Vespula Speed, Vespula Control and FUNC 1030 surfaces |
| Resolution error | $R_{esErr}$ | | 3 | | % | Up to 150ips on QCK with 3000 cpi |
| Acceleration | A | | | 30 | g | In run mode |

## 2.3 AC Electrical Specifications

### Table 4. AC Electrical Specifications

Electrical characteristics, over recommended operating conditions. Typical values at 25 °C, $V_{DD}$ = 2.0 V, $V_{DDIO}$ = 2.0V.

| Parameter | Symbol | Minimum | Typical | Maximum | Units | Notes |
|---|---|---|---|---|---|---|
| Motion Delay After Reset | $t_{MOT-RST}$ | 50 | | | ms | From reset to valid motion, assuming motion is present |
| Shutdown | $t_{STDWN}$ | | | 500 | µs | From Shutdown mode active to low current |
| Wake From Shutdown | $t_{WAKEUP}$ | 50 | | | ms | From Shutdown mode inactive to valid motion. Notes: A RESET must be asserted after a shutdown. Refer to section "Notes on Shutdown", also note $t_{MOT-RST}$ |
| MISO Rise Time | $t_{r-MISO}$ | | 50 | | ns | $C_L$ = 100pF |
| MISO Fall Time | $t_{f-MISO}$ | | 50 | | ns | $C_L$ = 100pF |
| MISO Delay After SCLK | $t_{DLY-MISO}$ | | | 90 | ns | From SCLK falling edge to MISO data valid, no load conditions |
| MISO Hold Time | $t_{hold-MISO}$ | 200 | | | ns | Data held until next falling SCLK edge |
| MOSI Hold Time | $t_{hold-MOSI}$ | 200 | | | ns | Amount of time data is valid after SCLK rising edge |
| MOSI Setup Time | $t_{setup-MOSI}$ | 120 | | | ns | From data valid to SCLK rising edge |
| SPI Time Between Write Commands | $t_{SWW}$ | 45 | | | µs | From rising SCLK for last bit of the first data byte, to rising SCLK for last bit of the second data byte. |
| SPI Time Between Write And Read Commands | $t_{SWR}$ | 45 | | | µs | From rising SCLK for last bit of the first data byte, to rising SCLK for last bit of the second address byte. |
| SPI Time Between Read And Subsequent Commands | $t_{SRW}$<br>$t_{SRR}$ | 20 | | | µs | From rising SCLK for last bit of the first data byte, to falling SCLK for the first bit of the address byte of the next command. |
| SPI Read Address-Data Delay | $t_{SRAD}$ | 35 | | | µs | From rising SCLK for last bit of the address byte, to falling SCLK for first bit of data being read. |
| NCS Inactive After Motion Burst | $t_{BEXIT}$ | 500 | | | ns | Minimum NCS inactive time after motion burst before next SPI usage |
| NCS To SCLK Active | $t_{NCS-SCLK}$ | 120 | | | ns | From last NCS falling edge to first SCLK rising edge |
| SCLK To NCS Inactive (For Read Operation) | $t_{SCLK-NCS}$ | 120 | | | ns | From last SCLK rising edge to NCS rising edge, for valid MISO data transfer |
| SCLK To NCS Inactive (For Write Operation) | $t_{SCLK-NCS}$ | 35 | | | µs | From last SCLK rising edge to NCS rising edge, for valid MOSI data transfer |

| Parameter | Symbol | Minimum | Typical | Maximum | Units | Notes |
|---|---|---|---|---|---|---|
| NCS To MISO High-Z | $t_{NCS-MISO}$ | | | 500 | ns | From NCS rising edge to MISO high-Z state |
| MOTION Rise Time | $t_{r-MOTION}$ | | 50 | | ns | $C_L$ = 100pF |
| MOTION Fall Time | $t_{f-MOTION}$ | | 50 | | ns | $C_L$ = 100pF |
| Input Capacitance | $C_{in}$ | | 50 | | pF | SCLK, MOSI, NCS |
| Load Capacitance | $C_L$ | | | 100 | pF | MISO, MOTION |
| Transient Supply Current | $I_{DDT}$ | | | 70 | mA | Max supply current during the supply ramp from 0V to $V_{DD}$ with min 150 us and max 20ms rise time. (Does not include charging currents for bypass capacitors) |
| | $I_{DDTIO}$ | | | 60 | mA | Max supply current during the supply ramp from 0V to $V_{DDIO}$ with min 150 us and max 20ms rise time. (Does not include charging currents for bypass capacitors) |

## 2.4 DC Electrical Specifications

### Table 5. DC Electrical Specifications

Electrical characteristics, over recommended operating conditions. Typical values at 25 °C, $V_{DD}$ = 2.0 V, $V_{DDIO}$ = 2.0V

| Parameter | Symbol | Min | Typical | Max | Units | Notes |
|---|---|---|---|---|---|---|
| DC Supply Current | $I_{DD\_RUN\_VFR1}$ [4k FPS]<br>$I_{DD\_RUN\_VFR2}$ [8k FPS]<br><br>$I_{DD\_REST1}$<br>$I_{DD\_REST2}$<br>$I_{DD\_REST3}$ | | 7.0<br>12.8<br><br>1.2<br>32<br>18 | | mA<br>mA<br><br>mA<br>uA<br>uA | Average current, including LED current at 1ms motion polling on Qck surface |
| Power Down Current | $I_{PD}$ | | 3 | | µA | |
| Input Low Voltage | $V_{IL}$ | | | 0.3 x $V_{DDIO}$ | V | SCLK, MOSI, NCS |
| Input High Voltage | $V_{IH}$ | 0.7 x $V_{DDIO}$ | | | V | SCLK, MOSI, NCS |
| Input Hysteresis | $V_{I\_HYS}$ | | 100 | | mV | SCLK, MOSI, NCS |
| Input Leakage Current | $I_{leak}$ | | ±1 | ±10 | µA | Vin=$V_{DDIO}$ or 0V, SCLK, MOSI, NCS |
| Output Low Voltage | $V_{OL}$ | | | 0.45 | V | Iout=1mA, MISO, MOTION |
| Output High Voltage | $V_{OH}$ | $V_{DDIO}$ - 0.45 | | | V | Iout=-1mA, MISO, MOTION |

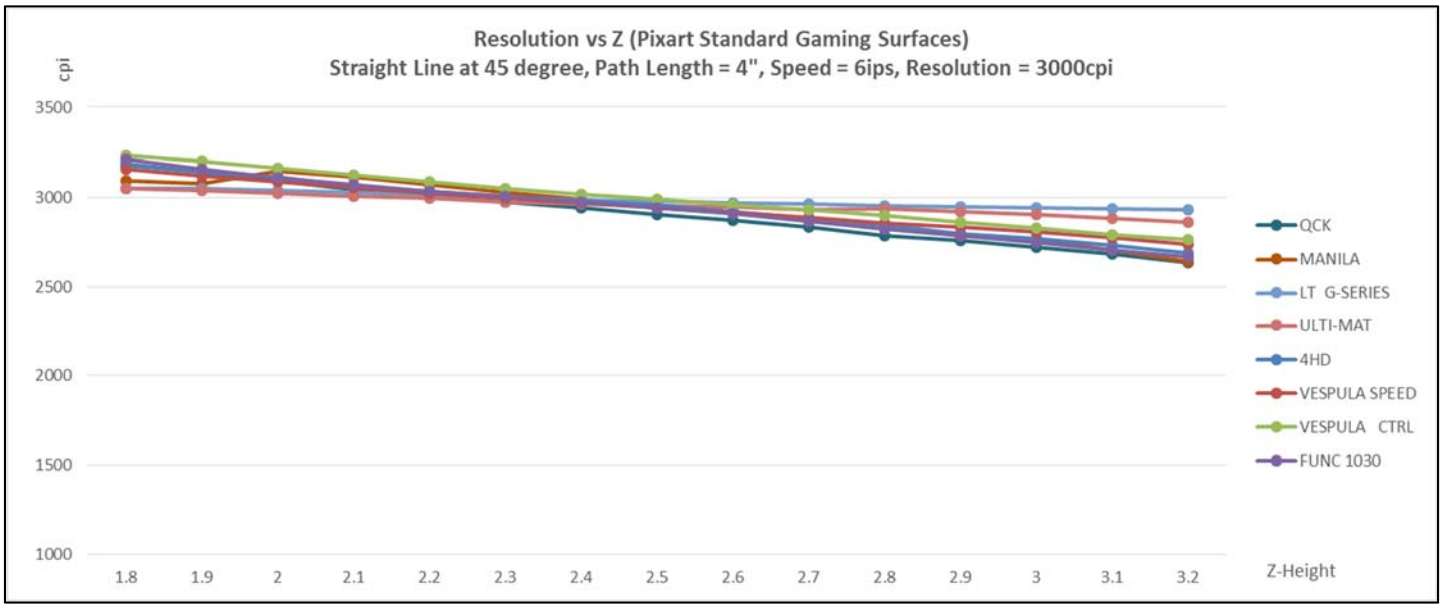



Figure 12. Mean Resolution vs. Z at default resolution at 3000cpi

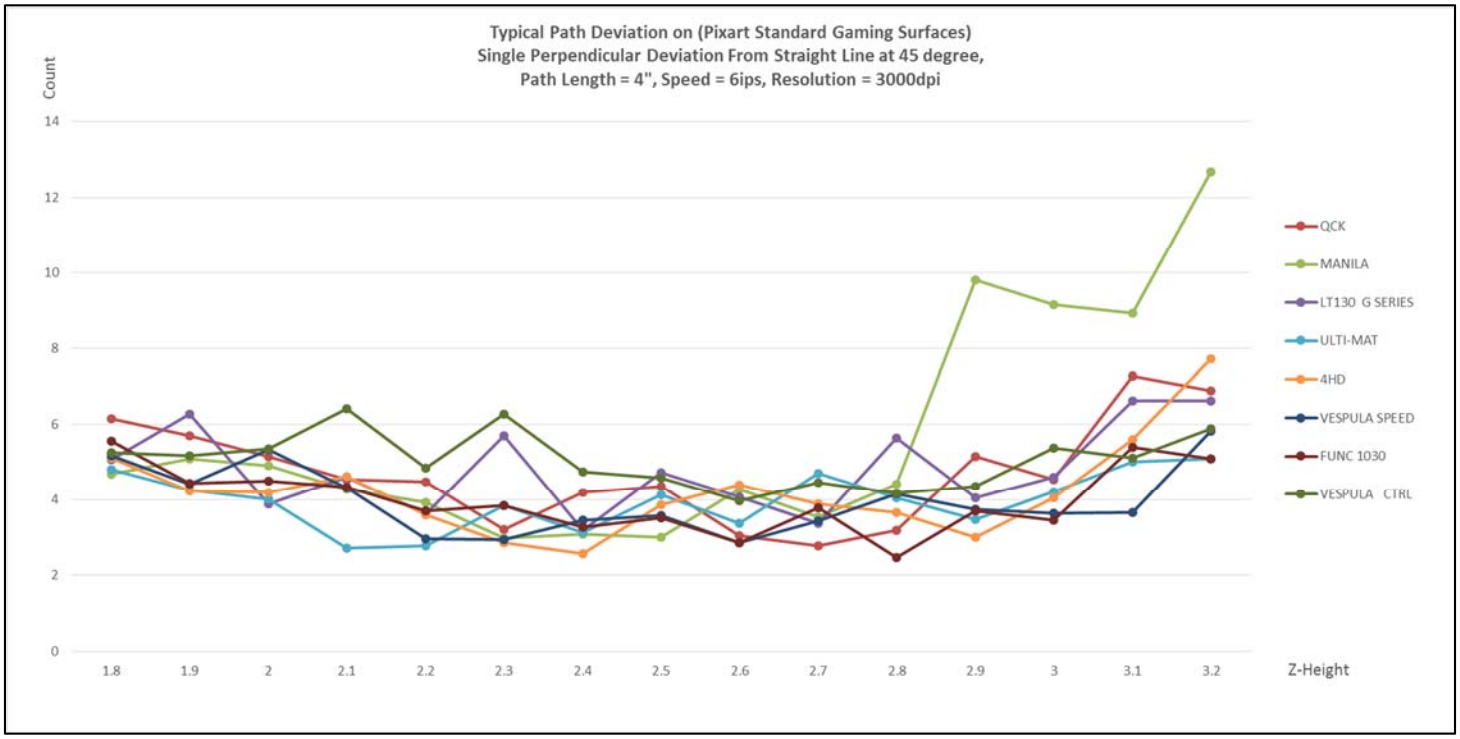



Figure 13. Path error vs. Z-height at default resolution at 3000cpi (mm)

## 3.0 Serial Peripheral Interface (SPI)

The synchronous serial port is used to set and read parameters in PMW3330DM-TZQU sensor, and to read out the motion information.

The port is a four wire port. The host microcontroller always initiates communication; PMW3330DM-TZQU sensor never initiates data transfers. SCLK, MOSI, and NCS may be driven directly by a microcontroller. The port pins may be shared with other SPI slave devices. When the NCS pin is high, the inputs are ignored and the output is tri-stated.

The lines that comprise the SPI port are:

| | |
|---|---|
| SCLK | Clock input, generated by the master (microcontroller). |
| MOSI | Input data. (Master Out/Slave In) |
| MISO | Output data. (Master In/Slave Out) |
| NCS | Chip select input (active low). NCS needs to be low to activate the serial port; otherwise, MISO will be high Z, and MOSI & SCLK will be ignored. NCS can also be used to reset the serial port in case of an error. |

### Motion Pin Timing

The motion pin is an active low output that signals the micro-controller when motion has occurred. The motion pin is lowered whenever the motion bit is set; in other words, whenever there is non-zero data in the Delta_X_L, Delta_X_H, Delta_Y_L or Delta_Y_H registers. Clearing the motion bit (by reading MOTION registers) will put the motion pin high.

### Chip Select Operation

The serial port is activated after NCS goes low. If NCS is raised during a transaction, the entire transaction is aborted and the serial port will be reset. After a transaction is aborted, the normal address-to-data or transaction-to-transaction delay is still required before beginning the next transaction. To improve communication reliability, all serial transactions should be framed by NCS. In other words, the port should not remain enabled during periods of non-use because ESD and EFT/B events could be interpreted as serial communication and put the chip into an unknown state. In addition, NCS must be raised after each burst-mode transaction is complete to terminate burst-mode. The port is not available for further use until burst-mode is terminated.

### Write Operation

Write operation, defined as data going from the micro-controller to PMW3330DM-TZQU sensor, 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. PMW3330DM-TZQU sensor reads MOSI on rising edges of SCLK.

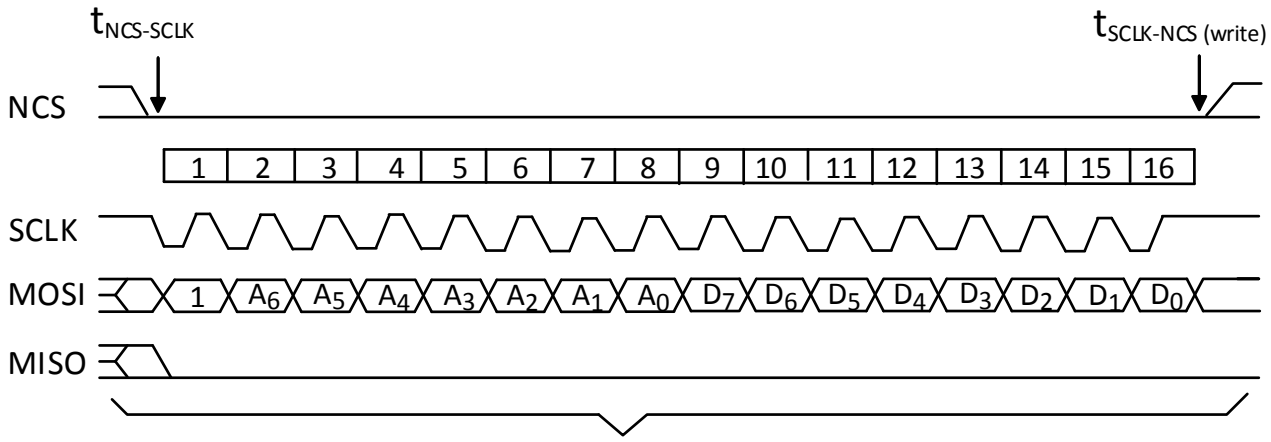



Figure 14. Write operation

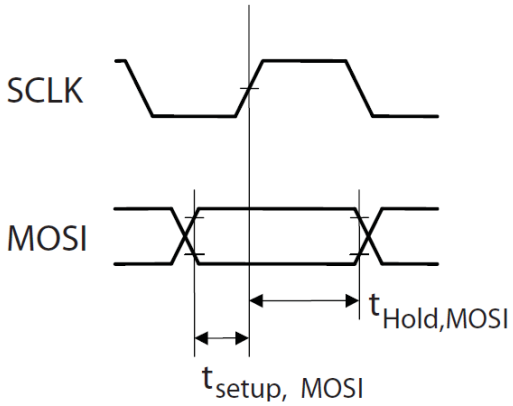



Figure 15. MOSI setup and hold time

## Read Operation

A read operation, defined as data going from PMW3330DM-TZQU sensor to the micro-controller, is always initiated by the micro-controller and consists of two bytes. The first byte contains the address, is sent by the micro-controller over MOSI, and has a "0" as its MSB to indicate data direction. The second byte contains the data and is driven by PMW3330DM-TZQU sensor over MISO. The sensor outputs MISO bits on falling edges of SCLK and samples MOSI bits on every rising edge of SCLK.

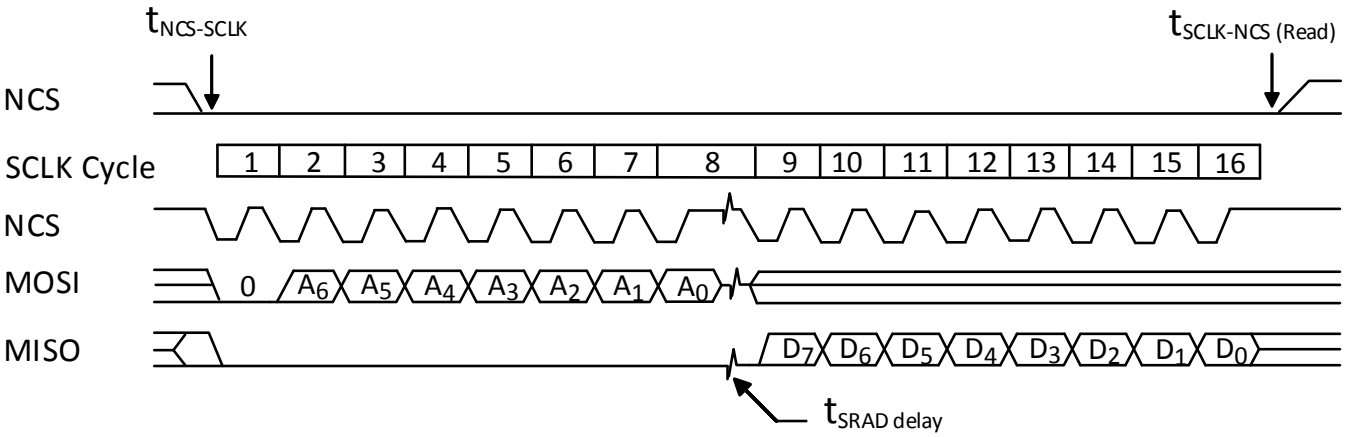



Figure 16. Read operation

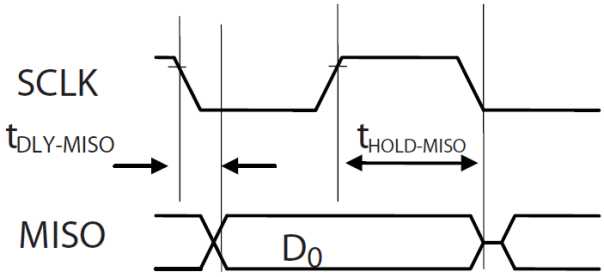



Figure 17. MISO Delay and hold time

**Note:** The minimum high state of SCLK is also the minimum MISO data hold time of PMW3330DM-TZQU sensor. Since the falling edge of SCLK is actually the start of the next read or write command, PMW3330DM-TZQU sensor will hold the state of data on MISO until the falling edge of SCLK.

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

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

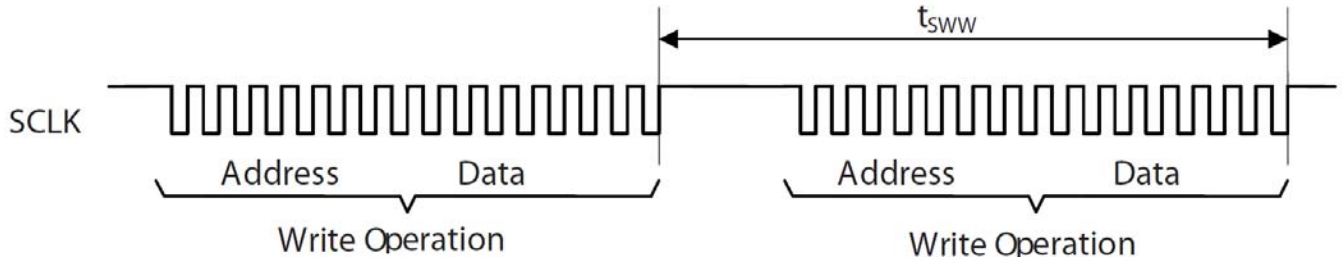



Figure 18. Timing between two write commands

If the rising edge of the SCLK for the last data bit of the second write command occurs before the $t_{SWW}$ delay, then the first write command may not complete correctly.

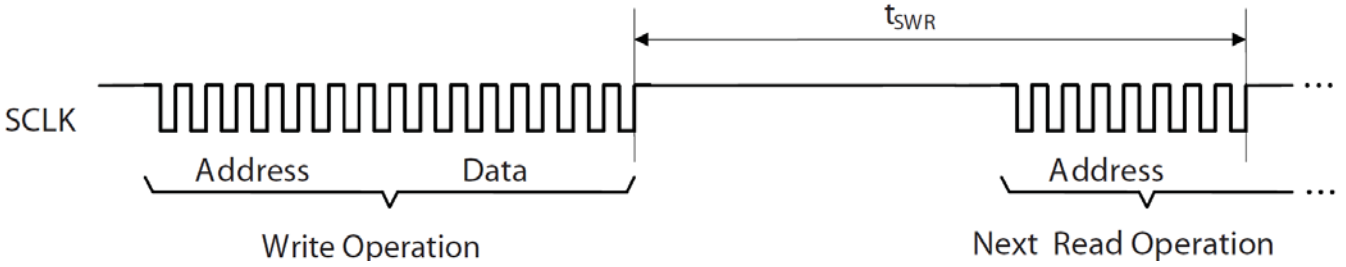



Figure 19. Timing between write and either write or subsequent read commands

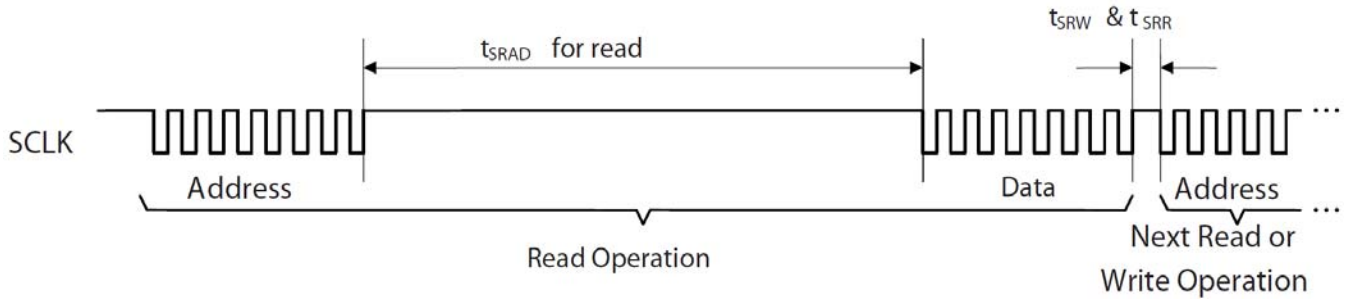



Figure 20. Timing between read and either write or subsequent read commands

If the rising edge of SCLK for the last address bit of the read command occurs before the $t_{SWR}$ required delay, the write command may not complete correctly. During a read operation SCLK should be delayed at least $t_{SRAD}$ after the last address data bit to ensure that the Sensor has time to prepare the requested data.

The falling edge of SCLK for the first address bit of either the read or write command must be at least $t_{SRR}$ or $t_{SRW}$ after the last SCLK rising edge of the last data bit of the previous read operation. In addition, during a read operation SCLK should be delayed after the last address data bit to ensure that PMW3330DM-TZQU sensor has time to prepare the requested data.

# 4.0 Burst mode operation

### Burst Mode Operation

Burst mode is a special serial port operation mode which may be used to reduce the serial transaction time for motion read. The speed improvement is achieved by continuous data clocking to or from multiple registers without the need to specify the register address, and by not requiring the normal delay period between data bytes.

### Motion Read

Reading the Burst_Motion_Read register activates this mode. PMW3330DM-TZQU sensor will respond with the following motion burst report in order. Motion burst report:

BYTE[00] = Motion
BYTE[01] = Observation
BYTE[02] = Delta_X_L
BYTE[03] = Delta_X_H
BYTE[04] = Delta_Y_L
BYTE[05] = Delta_Y_H
BYTE[06] = SQUAL
BYTE[07] = Pixel_Sum
BYTE[08] = Maximum_Pixel
BYTE[09] = Minimum_Pixel
BYTE[10] = Shutter_Upper
BYTE[11] = Shutter_Lower

After sending the register address, the microcontroller must wait for $t_{SRAD}$, and then begin reading data. All data bits can be read with no delay between bytes by driving SCLK at the normal rate. The data are latched into the output buffer after the last address bit is received. After the burst transmission is complete, the microcontroller must raise the NCS line for at least $t_{BEXIT}$ to terminate burst mode. The serial port is not available for use until it is reset with NCS, even for a second burst transmission.

### Procedure to start motion burst:

1. Lower NCS signal, wait for tNCS-SCLK delay.
2. Send Burst_Motion_Read address (0x16).
3. Wait for $t_{SRAD}$
4. Start reading SPI Data continuously up to 12 bytes. Motion burst may be terminated by pulling NCS high for at least $t_{BEXIT}$.
5. To read new motion burst data, repeat from step 1.

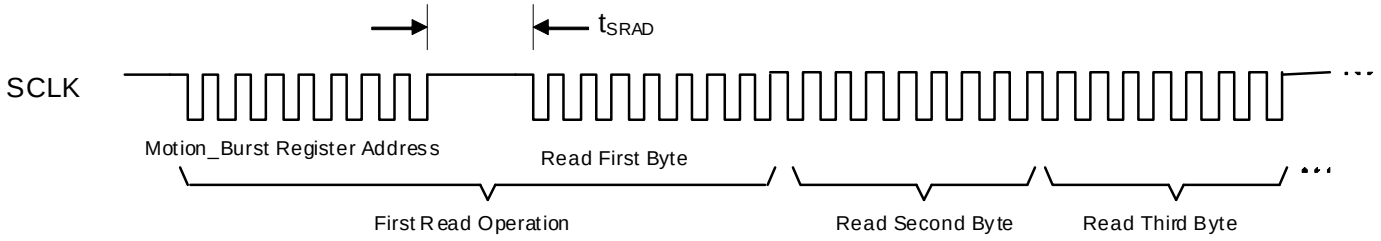



Figure 21. Motion Read sequence

**Note:** Motion burst data can be read from the Burst_Motion_Read registers even in run or rest mode.

## 5.0 Power Up

Although the sensor performs an internal power up self reset, it is still recommend that the Power_Up_Reset register is written every time power is applied. The appropriate sequence is as follows:

1. Apply power to VDD and VDDIO in any order, with a delay of no more than 100ms in between each supply. Ensure all supplies are stable.
2. Drive NCS high, and then low to reset the SPI port.
3. Write 0x5A to Power_Up_Reset register (or, alternatively toggle the NRESET pin).
4. Wait for at least 50ms.
5. Read from registers 0x02, 0x03, 0x04, 0x05 and 0x06 one time regardless of the motion pin state.
6. Load power up initialization register setting.
   - Write 0x80 to register 0x78
   - Write 0x80 to register 0x79
   - Write 0x80 to register 0x14
   - Write 0x40 to register 0x20
   - Write 0x40 to register 0x1A
   - Write 0x90 to register 0x1D
   - Write 0x12 to register 0x62
   - Write 0x12 to register 0x63
   - Write 0x02 to register 0x7f
   - Write 0x01 to register 0x40
   - Write 0x00 to register 0x7f

During power-up there will be a period of time after the power supply is high but before normal operation. The table below shows the state of the various pins during power-up and reset.

| State of Signal Pins After VDD is Valid | | |
|---|---|---|
| Pin | During Reset | After Reset |
| NRESET | Functional | Functional |
| NCS | Ignored | Functional |
| MISO | Undefined | Depends on NCS |
| SCLK | Ignored | Depends on NCS |
| MOSI | Ignored | Depends on NCS |
| MOTION | Undefined | Functional |

NRESET

The NRESET pin can be used to perform a full chip reset. When asserted, it performs the same reset function as the Power_Up_Reset_Register. The NRESET pin needs to be asserted (held to logic 0) for at least 100 ns.

*Note:- NRESET pin has a built in weak pull up circuit. During active low reset phase, it can draw a static current of up to 600uA.*

## 6.0 Shutdown

PMW3330DM-TZQU can be set in Shutdown mode by writing to Shutdown register. The SPI port should not be accessed when Shutdown mode is asserted, except the power-up command (writing 0x5a to register 0x3a). Other ICs on the same SPI bus can be accessed, as long as the sensor's NCS pin is not asserted.

To de-assert Shutdown mode:

1. Drive NCS high, and then low to reset the SPI port.
2. Write 0x5A to Power_Up_Reset register (or, alternatively toggle the NRESET pin).
3. Wait for at least 50ms.
4. Read from registers 0x02, 0x03, 0x04, 0x05 and 0x06 one time regardless of the motion pin state.
5. Load power up initialization register setting.
   - Write 0x80 to register 0x78
   - Write 0x80 to register 0x79
   - Write 0x80 to register 0x14
   - Write 0x40 to register 0x20
   - Write 0x40 to register 0x1A
   - Write 0x90 to register 0x1D
   - Write 0x12 to register 0x62
   - Write 0x12 to register 0x63
   - Write 0x02 to register 0x7f
   - Write 0x01 to register 0x40
   - Write 0x00 to register 0x7f

| Pin | Status when Shutdown Mode |
|---|---|
| NRESET | High |
| NCS | High*1 |
| MISO | Hi-Z*2 |
| SCLK | Ignore if NCS = 1*3 |
| MOSI | Ignore if NCS = 1*4 |
| MOTION | Output High |

*1. NCS pin must be held to 1 (high) if SPI bus is shared with other devices. It is recommended to hold to 1 (high) during Shutdown unless powering up the Sensor. It must be held to 0 (low) if the sensor is to be re-powered up from shutdown (writing 0x5a to register 0x3a).
*2. MISO should be either pull up or down during shutdown in order to meet the low power consumption specification in the Data sheet.
*3. SCLK is ignored if NCS is 1 (high). It is functional if NCS is 0 (low).
*4. MOSI is ignored if NCS is 1 (high). If NCS is 0 (low), any command present on the MOSI pin will be ignored except power-up command (writing 0x5a to register 0x3a).

*Note:- There is long wakeup time from shutdown. These features should not be used for power management during normal mouse motion.*

# 7.0 Lift Detection Options

There are three lift detection options for this sensor:-

- 2mm (default setting)
- 3mm
- Manual Lift Cut Off calibration

For application which required to have a lift detection that is lower than 2mm, manual lift cut off calibration procedure need to be implemented in the micro controller side. Please refer to section 7.3 for the manual lift cut off calibration for the complete implementation details.

## 7.1 General 2mm Lift Cut Off Setting

Setting for 2 mm lift detection: -

1) Write register 0x60 with value 0x04.
2) Write Register 0x69 with value 0x06.

## 7.2 General 3mm Lift Cut Off Setting

Setting for 3 mm lift detection: -

1) Write register 0x60 with value 0x0C.
2) Write register 0x69 with value 0x10.

## 7.3 Manual Lift Cut Off Calibration

This sensor has the ability to optimize its lift performance by calibrate internal parameters to the tracking surface. This lift cut off calibration feature involves user interaction.

1. To ensure that the lift cut off calibration procedure starts from a known configuration, reset the sensor and apply the recommended “Power Up” sequence.

2. Set up the following registers:

a) Write 0x80 to register Config (register 0x40).
b) Write 0x03 to register Resolution (register 0x1b).
c) Write 0x47 to register 0x5C.

3. Prompt the user that the lift cut off calibration procedure is about to begin to ensure that the mouse is placed on the surface (mouse is not lifted).

4. The lift cut off calibration procedure can be triggered by a GUI prompt to the user or user-initiated through a mouse-click event. Once the procedure is triggered, write 0x91 to register LiftCutOff_Calibration_Control (register 0x22). Read register bit LiftCutOff_Calibration_Status (register 0x23) bit [2] = 0 to confirm that the sensor is in lift cut off calibration mode.

5. The external MCU is responsible for tracking accumulated distance, duration and speed during lift cut off calibration motion.

a) The recommended minimum accumulated distance is 20 inches.
b) The recommended maximum duration is 20 seconds.
c) The recommended minimum and maximum mouse speed is 1 and 10 ips respectively.

6. If either 5a or 5b is satisfied, stop the lift cut off calibration procedure by writing 0x10 to register LiftCutOff_Calibration_Control (register 0x22). Read register bit LiftCutOff_Calibration_Status (register 0x23) bit [2] = 1 to confirm that the sensor is out of lift cut off calibration mode. The user should then be prompted that the lift cut off calibration procedure has stopped. The procedure is valid if all of the following are true:

a) Accumulated distance is more than the recommended value in step 5a.
b) Duration is less than the recommended value in step 5b.
c) Mouse speed during the procedure is within the recommended values in step 5c.

If the procedure is invalid, either re-start the procedure from step 3 or exit lift cut off calibration by proceeding to step 8.

If the procedure is valid, proceed to read register bit LiftCutOff_Calibration_Status (register 0x23) bit [0]:

a) If LiftCutOff_Calibration_Status (register 0x23) bit [0] = 1, the sensor is able to calibrate to the surface. Proceed to step 7.
b) If LiftCutOff_Calibration_Status (register 0x23) bit [0] = 0, the sensor is unable to calibrate to the surface. Proceed to step 8.

7. Proceeding from this step means that the sensor is able to recommend a new set of “Min SQUAL Run” and SQUAL threshold values. Update the sensor settings with the recommended values through these steps:

a) Read register LCC_Msq (register 0x70) and write the read back value to register 0x4e.
b) Read register LCC_Sqth (register 0x71) and write the read back value to register 0x57.

Complete lift cut off calibration by writing to the following registers:

a) Write 0x04 to register address 0x41.
b) Write 0x0C to register address 0x42.
c) Write 0x44 to register address 0x5C.
d) Write 0x64 to register address 0x5E.
e) Write 0x81 to register address 0x35.
f) Write 0x00 to register Config (register 0x40).
g) Write desired resolution setting to the register Resolution (register 0x1b).

Proceed to step 9.

8. Proceeding from this step means the sensor is unable to calibrate the surface. Write to the following registers to exit from lift cut off calibration:

a) Write 0x40 to register 0x5c.
b) Write 0x00 to Config (register 0x40).
c) Write desired resolution setting to Resolution (0x1b).

Proceed to step 9.

9. The lift cut off calibration procedure is completed.

## 8.0 Registers Table

PMW3330DM-TZQU registers are accessible via the serial port. The registers are used to read motion data and status as well as to set the device configuration.

| Address | Register | Access: R = Read, W = Write, Read / Write= RW | Default Value |
|---|---|---|---|
| 0x00 | Product_ID | R | 0x45 |
| 0x02 | Motion | RW | 0x00 |
| 0x03 | Delta_X_L | R | 0x00 |
| 0x04 | Delta_X_H | R | 0x00 |
| 0x05 | Delta_Y_L | R | 0x00 |
| 0x06 | Delta_Y_H | R | 0x00 |
| 0x07 | SQUAL | R | 0x00 |
| 0x08 | Pixel_Sum | R | 0x00 |
| 0x09 | Maximum_Pixel | R | 0x00 |
| 0x0A | Minimum_Pixel | R | 0x00 |
| 0x0B | Shutter_Lower | R | 0x12 |
| 0x0C | Shutter_Upper | R | 0x00 |
| 0x15 | Chip_Observation | RW | 0x80 |
| 0x16 | Burst_Motion_Read | R | 0x00 |
| 0x19 | Pix_Grab_Status | R | 0x00 |
| 0x1B | Resolution | RW | 0x3B |
| 0x1E | Angle_Snap | RW | 0x04 |
| 0x20 | Axis_Control | RW | 0x00 |
| 0x22 | LiftCutoff_Calibration_Control | RW | 0x10 |
| 0x23 | LiftCutoff_Calibration_Status | RO | 0x00 |
| 0x24 | Run_DownShift | RW | 0x08 |
| 0x25 | Rest1_Period | RW | 0x01 |
| 0x26 | Rest1_Downshift | RW | 0x4F |
| 0x27 | Rest2_Period | RW | 0x19 |
| 0x28 | Rest2_Downshift | RW | 0x5E |
| 0x29 | Rest3_Period | RW | 0x3F |
| 0x32 | Pix_Grab | RW | 0x00 |
| 0x3A | Power_Up_Reset | W | N/A |
| 0x3B | Shutdown | W | N/A |
| 0x3F | Inv_Product_ID | R | 0xBA |
| 0x40 | Config | RW | 0x00 |
| 0x4E | Min_Squal_Run_LCC_Paf | RW | 0x0F |
| 0x70 | LCC_Msq | RO | 0x00 |
| 0x71 | LCC_Sqth | RO | 0x00 |

## 9.0 Registers Description

| Register: 0x00 | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Name: Product_ID | | | | | | | | |
| Bit # | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Field | $PID_7$ | $PID_6$ | $PID_5$ | $PID_4$ | $PID_3$ | $PID_2$ | $PID_1$ | $PID_0$ |
| Reset Value | 0x45 | | | | | | | |
| Access | Read Only | | | | | | | |
| Data Type | 8-bit unsigned integer | | | | | | | |
| Usage | This value is a unique identification assigned to this model only. The value in this register does not change; it can be used to verify that the serial communications link is functional. | | | | | | | |

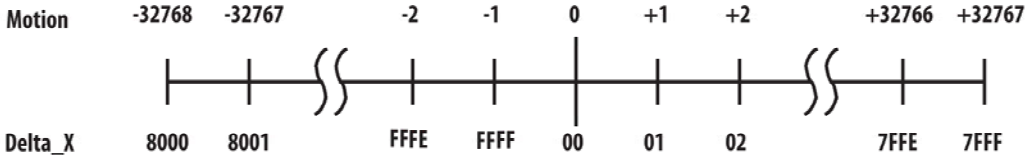
| Register: 0x02 | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Name: Motion | | | | | | | | |
| Bit # | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Field | MOT | Reserved | Reserved | Reserved | Reserved | Reserved | $OP\_MODE_1$ | $OP\_MODE_0$ |
| Reset Value | 0x00 | | | | | | | |
| Access | Read / Write | | | | | | | |
| Data Type | 8-bit Field | | | | | | | |

**Usage**

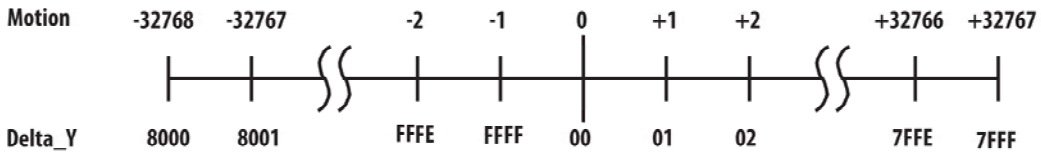
This register allows the user to determine if motion has occurred since the last time it was read.
The procedure to read the motion registers (Delta_X_L, Delta_X_H, Delta_Y_L and Delta_Y_H) is as follows:
1. Read the Motion register. This will freeze the Delta_X_L, Delta_X_H, Delta_Y_L and Delta_Y_H register values.
2. If the MOT bit is set, Delta_X_L, Delta_X_H, Delta_Y_L and Delta_Y_H registers should be read in the given sequence to get the accumulated motion. Note: if Delta_X_L, Delta_X_H, Delta_Y_L and Delta_Y_H registers are not read before the motion register is read for the second time, the data in Delta_X_L, Delta_X_H, Delta_Y_L and Delta_Y_H will be lost.
3. To read a new set of motion data (Delta_X_L, Delta_X_H, Delta_Y_L and Delta_Y_H), repeat from Step 1.

| 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_L, Delta_X_H, Delta_Y_L and Delta_Y_H registers |
| OP_Mode [1:0] | **00 – Run mode**<br>01 – Rest 1<br>10 – Rest 2<br>11 – Rest 3 |

Write any value to this register will clear all motion data.

| Register: 0x03 | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Name: Delta_X_L | | | | | | | | |
| Bit # | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Field | $X_7$ | $X_6$ | $X_5$ | $X_4$ | $X_3$ | $X_2$ | $X_1$ | $X_0$ |
| Reset Value | 0x00 | | | | | | | |
| Access | Read Only | | | | | | | |
| Data Type | 16 bits 2's complement number. Lower 8 bits of Delta_X. | | | | | | | |
| Usage | X movement is counts since last report. Absolute value is determined by resolution.<br>  | | | | | | | |

| Register: 0x04 | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Name: Delta_X_H | | | | | | | | |
| Bit # | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Field | $X_{15}$ | $X_{14}$ | $X_{13}$ | $X_{12}$ | $X_{11}$ | $X_{10}$ | $X_9$ | $X_8$ |
| Reset Value | 0x00 | | | | | | | |
| Access | Read Only | | | | | | | |
| Data Type | 16 bits 2's complement number. Upper 8 bits of Delta_X. | | | | | | | |
| Usage | Delta_X_H must be read after Delta_X_L to have the full motion data. | | | | | | | |

| Register: 0x05 | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Name: Delta_Y_L | | | | | | | | |
| Bit # | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Field | $Y_7$ | $Y_6$ | $Y_5$ | $Y_4$ | $Y_3$ | $Y_2$ | $Y_1$ | $Y_0$ |
| Reset Value | 0x00 | | | | | | | |
| Access | Read Only | | | | | | | |
| Data Type | 16 bits 2's complement number. Lower 8 bits of Delta_Y. | | | | | | | |
| Usage | Y movement is counts since last report. Absolute value is determined by resolution.<br>  | | | | | | | |

| Register: 0x06 | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Name: Delta_Y_H | | | | | | | | |
| Bit # | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Field | $Y_{15}$ | $Y_{14}$ | $Y_{13}$ | $Y_{12}$ | $Y_{11}$ | $Y_{10}$ | $Y_9$ | $Y_8$ |
| Reset Value | 0x00 | | | | | | | |
| Access | Read Only | | | | | | | |
| Data Type | 16 bits 2's complement number. Upper 8 bits of Delta_Y. | | | | | | | |
| Usage | Delta_Y_H must be read after Delta_Y_L to have the full motion data. | | | | | | | |
| ***Note:*** *It is recommended that register 0x02, 0x03, 0x04, 0x05 and 0x06 to be read sequentially.* | | | | | | | | |

| Register: 0x07 | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Name: SQUAL | | | | | | | | |
| Bit # | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Field | $SQ_7$ | $SQ_6$ | $SQ_5$ | $SQ_4$ | $SQ_3$ | $SQ_2$ | $SQ_1$ | $SQ_0$ |
| Reset Value | 0x00 | | | | | | | |
| Access | Read Only | | | | | | | |
| Data Type | 8-bit unsigned integer | | | | | | | |
| Usage | The SQUAL (Surface quality) register is a measure of the number of valid features visible by the sensor in the current frame. Use the following formula to find the total number of valid features.<br><br>Number of Features = SQUAL Register Value * 4<br><br>The maximum SQUAL register value is 0x9c. 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 500 sequentially acquired SQUAL values, while a sensor was moved slowly over white paper.<br><br>SQUAL values are only valid in run mode. Disable Rest mode before measuring SQUAL. | | | | | | | | |

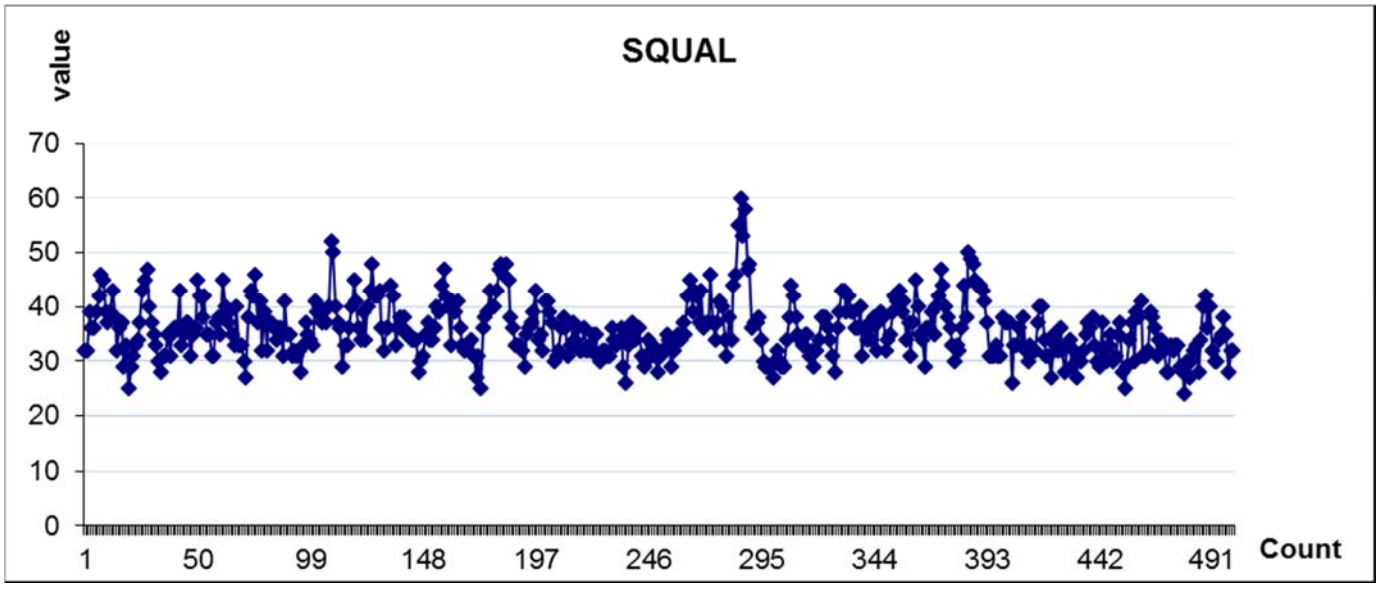



Figure 22. Average SQUAL on white paper

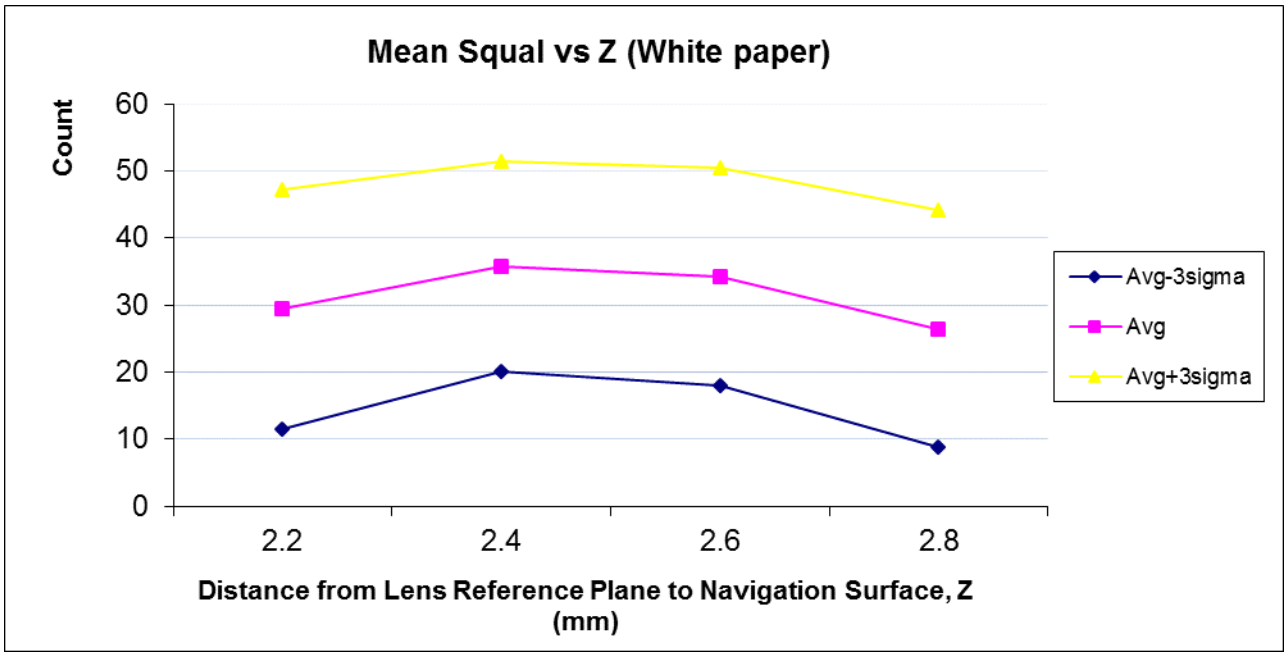



Figure 23. Mean Squal vs Z

| Register: 0x08 | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Name: Pixel_Sum | | | | | | | | |
| Bit # | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Field | $AP_7$ | $AP_6$ | $AP_5$ | $AP_4$ | $AP_3$ | $AP_2$ | $AP_1$ | $AP_0$ |
| Reset Value | 0x00 | | | | | | | |
| Access | Read Only | | | | | | | |
| Data Type | 8-bit unsigned integer | | | | | | | |
| Usage | This register is used to find the average pixel value. It reports the upper byte of an 17-bit counter which sums all 784 pixels in the current frame.<br><br>The maximum register value is 0xC2. The minimum register value is 0. The pixel sum value can change every frame. Disable rest mode before reading Pixel sum value. | | | | | | | |

| Register: 0x09 | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Name: Maximum_Pixel | | | | | | | | |
| Bit # | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Field | $MP_7$ | $MP_6$ | $MP_5$ | $MP_4$ | $MP_3$ | $MP_2$ | $MP_1$ | $MP_0$ |
| Reset Value | 0x00 | | | | | | | |
| Access | Read Only | | | | | | | |
| Data Type | 8-bit unsigned integer | | | | | | | |
| Usage | Maximum Pixel value in current frame. Minimum value = 0, maximum value = 127. The maximum pixel value can change every frame. | | | | | | | |

| Register: 0x0A | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Name: Minimum_Pixel | | | | | | | | |
| Bit # | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Field | $MinP_7$ | $MinP_6$ | $MinP_5$ | $MinP_4$ | $MinP_3$ | $MinP_2$ | $MinP_1$ | $MinP_0$ |
| Reset Value | 0x00 | | | | | | | |
| Access | Read Only | | | | | | | |
| Data Type | 8-bit unsigned integer | | | | | | | |
| Usage | Minimum Pixel value in current frame. Minimum value = 0, maximum value = 127. The minimum pixel value can change every frame. | | | | | | | |

| Register: 0x0B | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Name: Shutter_Lower | | | | | | | | |
| Bit # | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Field | $S_7$ | $S_6$ | $S_5$ | $S_4$ | $S_3$ | $S_2$ | $S_1$ | $S_0$ |
| Reset Value | 0x12 | | | | | | | |
| Access | Read Only | | | | | | | |
| Data Type | 12-bit unsigned number | | | | | | | |
| Usage | Lower byte of the 12-bit Shutter register. | | | | | | | |

| Register: 0x0C | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Name: Shutter_Upper | | | | | | | | |
| Bit # | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Field | Reserved | Reserved | Reserved | $S_{12}$ | $S_{11}$ | $S_{10}$ | $S_9$ | $S_8$ |
| Reset Value | 0x00 | | | | | | | |
| Access | Read Only | | | | | | | |
| Data Type | 12-bit unsigned number | | | | | | | |
| Usage | Upper 4-bit of the 12-bit Shutter register. Units are clock cycles of the internal oscillator. Read Shutter_Upper first, then Shutter_Lower. They should be read consecutively. The shutter is adjusted to keep the average pixel values within normal operating ranges. The shutter value is checked and automatically adjusted to a new value if needed on every frame when operating in default mode. | | | | | | | |

| Register: 0x15 | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Name: Chip_Observation | | | | | | | | |
| Bit # | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Field | Reserved | Reserved | $CO_5$ | $CO_4$ | $CO_3$ | $CO_2$ | $CO_1$ | $CO_0$ |
| Reset Value | 0x80 | | | | | | | |
| Access | Read / Write | | | | | | | |
| Data Type | Bit Field | | | | | | | |
| Usage | The user must clear the register by writing 0x00, wait for a minimum $T_{dly\_obs}$ ms & read the register. The active process $CO_{5-0}$ will have set the corresponding bit. The register may be used as part of recovery scheme to detect a problem caused by EFT/B or ESD event.<br><br>$T_{dly\_obs}$ is defined as the longest frame period +0.5ms. The longest frame period is when sensor is in Rest3 mode. Clock frequency tolerance value need to be taken into account. For example if the default Rest3 period of 500ms is used, then $T_{dly\_obs}$ = (500 x 1.4) + 0.5 = 700.5ms | | | | | | | |

| Register: 0x16 | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Name: Burst_Motion_Read | | | | | | | | |
| Bit # | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Field | $MB_7$ | $MB_6$ | $MB_5$ | $MB_4$ | $MB_3$ | $MB_2$ | $MB_1$ | $MB_0$ |
| Reset Value | 0x00 | | | | | | | |
| Access | Read | | | | | | | |
| Data Type | Bit Field | | | | | | | |
| Usage | The Burst_Motion_Read register is used for high-speed access to the Motion, Observation, Delta_X_L, Delta_X_H, Delta_Y_L, Delta_Y_H, SQUAL, Pixel_Sum, Maximum_Pixel, Minimum_Pixel, Shutter_Upper and Shutter_Lower registers. See Burst Mode-Motion Read section for use details. | | | | | | | |

| Register: 0x19 | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Name: Pix_Grab_Status | | | | | | | | |
| Bit # | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Field | Reserved | PG_VALID | PG_FIRST | Reserved | Reserved | Reserved | Reserved | Reserved |
| Reset Value | 0x00 | | | | | | | |
| Access | Read | | | | | | | |
| Data Type | Bit Field | | | | | | | |
| Usage | This register provide additional information for user to monitor the sensor navigation status.<br>**Field Name** — **Description**<br>PG_VALID — 1 = Pixel Grab valid<br>PG_FIRST — 1 = Pixel Grab first | | | | | | | |

| Register: 0x1B | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Name: Resolution | | | | | | | | |
| Bit # | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Field | RES 7 | RES 6 | RES 5 | RES 4 | RES 3 | RES 2 | RES 1 | RES 0 |
| Reset Value | 0x3B | | | | | | | |
| Access | Read / Write | | | | | | | |
| Data Type | 8-bit unsigned integer | | | | | | | |
| Usage | This register set the resolution XY of the sensor, The approximate resolution value for each register setting can be calculated using the following formula. Each bit change is ~50cpi. The recommended maximum write value is 0x8f, which the resolution setting is approximately 7200cpi. | | | | | | | |

| Field Name | Description |
|---|---|
| RES 7-0 | Set resolution with CPI step of approximately 50 cpi<br>0x00: 50 cpi (minimum)<br>0x01: 100 cpi<br>:<br>**0x3b: 3000 cpi**<br>:<br>0x8f: 7200 cpi |

| Register: 0x1e | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Name: Angle_Snap | | | | | | | | |
| Bit # | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Field | EN | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| Reset Value | 0x04 | | | | | | | |
| Access | Read /Write | | | | | | | |
| Data Type | 8-bit unsigned integer | | | | | | | |
| Usage | Write to this register to enable the angle snap feature.<br>**Field Name** — **Description**<br>EN — **0: angle snap disable**<br>1: angle snap enable | | | | | | | |

| Register: 0x20 | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Name: Axis_Control | | | | | | | | |
| Bit # | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Field | Swap_XY | INV_Y | INV_X | Reserved | Reserved | Reserved | Reserved | Reserved |
| Reset Value | 0x00 | | | | | | | |
| Access | Read /Write | | | | | | | |
| Data Type | Bit Field | | | | | | | |
| Usage | This register set the axis direction of the sensor reporting.<br>**Field Name** — **Description**<br>Swap_XY — 1: Swap XY directions<br>INV_Y — 1: Invert Y direction<br>INV_X — 1: Invert X direction | | | | | | | |

| Register: 0x22 | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Name: LiftCutoff_Calibration_Control | | | | | | | | |
| Bit # | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Field | LCCCLK_EN | Reserved | 0 | 1 | Reserved | Reserved | Reserved | LCC_START |
| Reset Value | 0x10 | | | | | | | |
| Access | Read /Write | | | | | | | |
| Data Type | Bit Field | | | | | | | |
| Usage | This register is to configure the lift cutoff calibration settings<br>**Field Name** — **Description**<br>LCCCLK_EN — Lift cut off calibration procedure enable<br>**0: Disable**<br>1: Enable<br>LCC_START — Lift cut off calibration Start/Stop<br>**0: Stop**<br>1: Start | | | | | | | |

| Register: 0x23 | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Name: LiftCutoff_Calibration_Status | | | | | | | | |
| Bit # | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Field | Reserved | Reserved | Reserved | Reserved | Reserved | LCC_DONE | Reserved | LCC_VALID |
| Reset Value | 0x00 | | | | | | | |
| Access | Read Only | | | | | | | |
| Data Type | Bit Field | | | | | | | |
| Usage | This register provide the lift cutoff calibration status<br>**Field Name** — **Description**<br>LCC_DONE — Lift cut off calibration done<br>LCC_VALID — lift cut off calibration Result<br>**0x0: Invalid**<br>0x1: Valid | | | | | | | |

| Register: 0x24 | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Name: Run_Downshift | | | | | | | | |
| Bit # | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Field | 0 | 0 | $RD_5$ | $RD_4$ | $RD_3$ | $RD_2$ | $RD_1$ | $RD_0$ |
| Reset Value | 0x08 | | | | | | | |
| Access | Read / Write | | | | | | | |
| Data Type | 8-bit unsigned integer | | | | | | | |
| Usage | This register set the Run to Rest1 downshift time. Use the formula below for calculation.<br><br>Run Downshift time (ms) = RD[7:0] x 256 x VFR1<br>Example for 4k FPS = 8 x 256 x 0.25 = 512ms<br><br>Max Downshift time is 63 x 256 x 0.25 = 4032ms<br>Min value is 0x01. A value of 0x00 will be internally clipped to 0x01.<br><br>All the above values are expected to have a +40% & -20% tolerance. | | | | | | | |

| Register: 0x25 | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Name: Rest1_Period | | | | | | | | |
| Bit # | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Field | $R1P_7$ | $R1\ P_6$ | $R1\ P_5$ | $R1\ P_4$ | $R1\ P_3$ | $R1\ P_2$ | $R1\ P_1$ | $R1\ P_0$ |
| Reset Value | 0x01 | | | | | | | |
| Access | Read / Write | | | | | | | |
| Data Type | 8-bit unsigned integer | | | | | | | |
| Usage | This register set the rest1 period<br><br>Rest1 period = R1P[7:0] x 1ms<br>Default rest1 period = 1 x 1ms = 1 ms<br><br>Min value is 0x01. A value of 0x00 is invalid.<br><br>All the above values are expected to have a +40% & -20% tolerance. | | | | | | | |

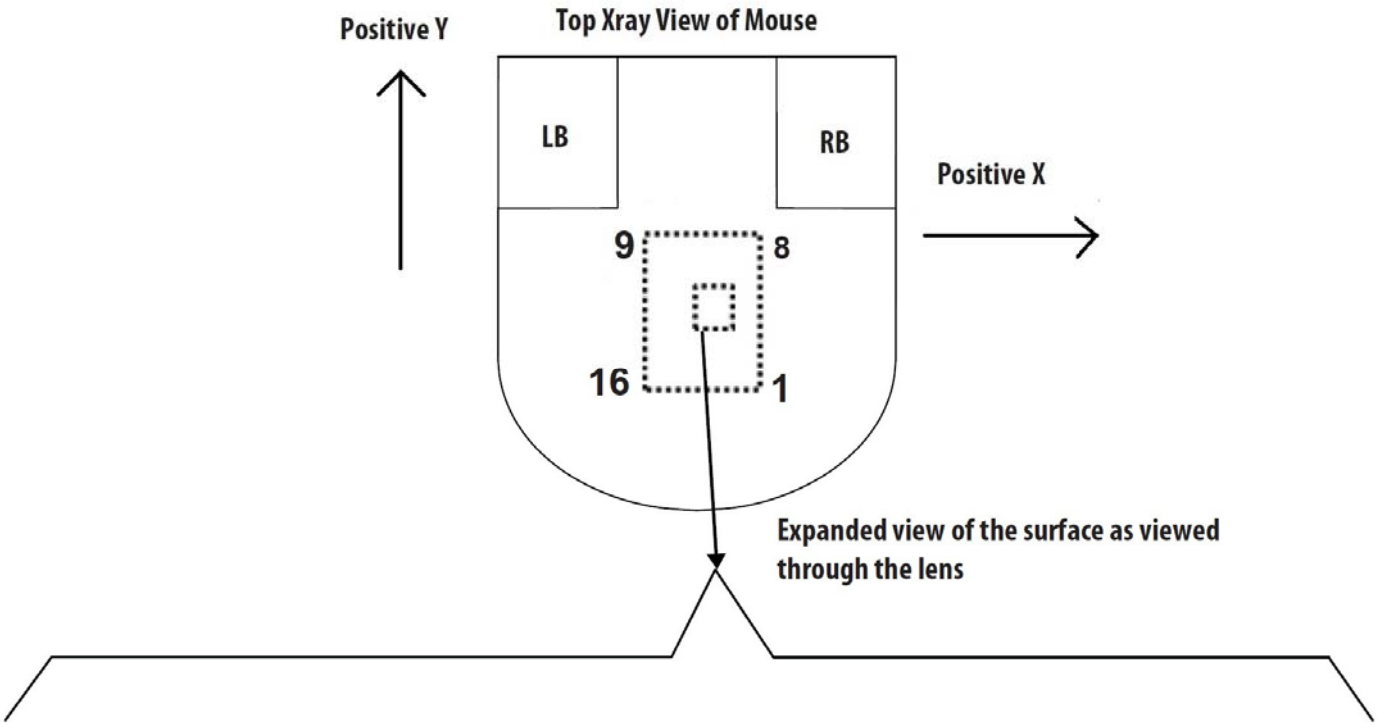
| Register: 0x26 | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Name: Rest1_Downshift | | | | | | | | |
| Bit # | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Field | $R1D_7$ | $R1D_6$ | $R1D_5$ | $R1D_4$ | $R1D_3$ | $R1D_2$ | $R1D_1$ | $R1D_0$ |
| Reset Value | 0x4F | | | | | | | |
| Access | Read / Write | | | | | | | |
| Data Type | 8-bit unsigned integer | | | | | | | |
| Usage | This register set the Rest1 to Rest2 downshift time. Use the formula below for calculation.<br><br>Rest1 Downshift time (ms) = R1D[7:0] x 256 x rest1_period(default =1ms)<br>Default = 79 x 256 x 1ms = 20224ms = 20s<br><br>Min value is 0x01. A value of 0x00 will be internally clipped to 0x01.<br><br>All the above values are expected to have a +40% & -20% tolerance. | | | | | | | |

| Register: 0x27 | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Name: Rest2_Period | | | | | | | | |
| Bit # | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Field | $R2P_7$ | $R2P_6$ | $R2P_5$ | $R2P_4$ | $R2P_3$ | $R2P_2$ | $R2P_1$ | $R2P_0$ |
| Reset Value | 0x19 | | | | | | | |
| Access | Read / Write | | | | | | | |
| Data Type | 8-bit unsigned integer | | | | | | | |
| Usage | This register set the rest2 period<br><br>Rest2 period = R2P[7:0] x slow clock (1ms) x4<br>Default rest2 period = 25 x 1ms x4= 100ms<br><br>Min value is 0x01. A value of 0x00 is invalid.<br><br>All the above values are expected to have a +40% & -20% tolerance. | | | | | | | |

| Register: 0x28 | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Name: Rest2_Downshift | | | | | | | | |
| Bit # | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Field | $R2D_7$ | $R2D_6$ | $R2D_5$ | $R2D_4$ | $R2D_3$ | $R2D_2$ | $R2D_1$ | $R2D_0$ |
| Reset Value | 0x5E | | | | | | | |
| Access | Read / Write | | | | | | | |
| Data Type | 8-bit unsigned integer | | | | | | | |
| Usage | This register set the Rest2 to Rest3 downshift time. Use the formula below for calculation.<br><br>Rest2 Downshift time (ms) = R2D[7:0] x 64 x rest2_period(default =100ms)<br>Default = 94 x 64 x 100ms = 601.6s = 10 min<br><br>Min value is 0x01. A value of 0x00 will be internally clipped to 0x01.<br><br>All the above values are expected to have a +40% & -20% tolerance. | | | | | | | |

| Register: 0x29 | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Name: Rest3_Period | | | | | | | | |
| Bit # | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Field | $R3P_7$ | $R3P_6$ | $R3P_5$ | $R3P_4$ | $R3P_3$ | $R3P_2$ | $R3P_1$ | $R3P_0$ |
| Reset Value | 0x3F | | | | | | | |
| Access | Read / Write | | | | | | | |
| Data Type | 8-bit unsigned integer | | | | | | | |
| Usage | This register set the rest3 period<br><br>Rest3 period = R3P[7:0] x slow clock (1ms) x8<br>Default rest3 period = 63 x 1ms x8 = 504ms<br><br>Min value is 0x01. A value of 0x00 is invalid.<br><br>All the above values are expected to have a +40% & -20% tolerance. | | | | | | | |

| Register: 0x32 | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Name: Pix_Grab | | | | | | | | |
| Bit # | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Field | VALID | $PIX_6$ | $PIX_5$ | $PIX_4$ | $PIX_3$ | $PIX_2$ | $PIX_1$ | $PIX_0$ |
| Reset Value | 0x00 | | | | | | | |
| Access | Read / Write | | | | | | | |
| Data Type | 8-bit unsigned integer | | | | | | | |
| Usage | 1. Write register 0x40 with value 0x80<br>2. Write register 0x6b with value 0x80<br>3. Write any value to register 0x32<br>4. Continuously read register 0x19 until getting both PG_VALID and PG_FIRST as “1”<br>5. Read register 0x32 for the 1 bit VALID flag and 7 bit ADC data (PIX6–0) for 784 times to form a complete pixel array information with at least 1 frame period (recommended 500us) delay between each readout. Any readout data with VALID = “0” indicate an error, and pix grab process needs to repeat from step (3)<br>6. Write register 0x40 with value 0x00<br>7. Write register 0x6b with value 0x00 | | | | | | | |




Last output

| | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 27 | 55 | 83 | 111 | 139 | 167 | 195 | 223 | 251 | 279 | 307 | 335 | 363 | 391 | 419 | 447 | 475 | 503 | 531 | 559 | 587 | 615 | 643 | 671 | 699 | 727 | 755 | 783 |
| 26 | 54 | 82 | 110 | 138 | 166 | 194 | 222 | 250 | 278 | 306 | 334 | 362 | 390 | 418 | 446 | 474 | 502 | 530 | 558 | 586 | 614 | 642 | 671 | 698 | 726 | 754 | 782 |
| 25 | 53 | 81 | 109 | 137 | 165 | 193 | 221 | 249 | 277 | 305 | 333 | 361 | 389 | 417 | 445 | 473 | 501 | 529 | 557 | 585 | 613 | 641 | 669 | 697 | 725 | 753 | 781 |
| . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . |
| . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . |
| 2 | 30 | 58 | 86 | 114 | 142 | 170 | 198 | 226 | 254 | 282 | 310 | 338 | 366 | 394 | 422 | 450 | 478 | 506 | 534 | 562 | 590 | 618 | 646 | 674 | 702 | 730 | 758 |
| 1 | 29 | 57 | 85 | 113 | 141 | 169 | 197 | 225 | 253 | 281 | 309 | 337 | 365 | 393 | 421 | 449 | 477 | 505 | 533 | 561 | 589 | 617 | 645 | 673 | 701 | 729 | 757 |
| 0 | 28 | 56 | 84 | 112 | 140 | 168 | 196 | 224 | 252 | 280 | 308 | 336 | 364 | 392 | 420 | 448 | 476 | 504 | 532 | 560 | 588 | 616 | 644 | 672 | 700 | 728 | 756 |

First output

Figure 24. Pixel Map (Surface referenced)

***Note**: The X/Y reporting direction is showed above which is follow all the power up initialization in Power Up section.*

| Register: 0x3A | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Name: Power_Up_Reset | | | | | | | | |
| Bit # | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Field | $PUR_7$ | $PUR_6$ | $PUR_5$ | $PUR_4$ | $PUR_3$ | $PUR_2$ | $PUR_1$ | $PUR_0$ |
| Reset Value | N/A | | | | | | | |
| Access | Write Only | | | | | | | |
| Data Type | 8-bit unsigned integer | | | | | | | |
| Usage | Write 0x5a to this register to reset the chip. All settings will revert to default values. Reset is required after recovering from shutdown mode. | | | | | | | |

| Register: 0x3B | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Name: Shutdown | | | | | | | | |
| Bit # | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Field | $SD_7$ | $SD_6$ | $SD_5$ | $SD_4$ | $SD_3$ | $SD_2$ | $SD_1$ | $SD_0$ |
| Reset Value | N/A | | | | | | | |
| Access | Write Only | | | | | | | |
| Data Type | 8-bit unsigned integer | | | | | | | |
| Usage | Write 0xB6 to set the chip to shutdown mode. Refer to the Shutdown section for more details on recovery procedure | | | | | | | |

| Register: 0x3F | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Name: Inverse_Product_ID | | | | | | | | |
| Bit # | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Field | $IPID_7$ | $IPID_6$ | $IPID_5$ | $IPID_4$ | $IPID_3$ | $IPID_2$ | $IPID_1$ | $IPID_0$ |
| Reset Value | 0xBA | | | | | | | |
| Access | Read Only | | | | | | | |
| Data Type | Bit Field | | | | | | | |
| Usage | This value is the inverse of the Product_ID. It is used to test the SPI port hardware. | | | | | | | |

| Register: 0x40 | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Name: Config | | | | | | | | |
| Bit # | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Field | AWAKE | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Reset Value | 0x00 | | | | | | | |
| Access | Read / Write | | | | | | | |
| Data Type | Bit Field | | | | | | | |
| Usage | This register configures the operating mode of the sensor<br>**Field Name** — **Description**<br>AWAKE — **0: Enable rest mode**<br>1: Disable rest mode, sensor always in active mode | | | | | | | |

| Register: 0x4E | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Name: Min_Squal_Run_LiftCutoff_Calibration | | | | | | | | |
| Bit # | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Field | $MSRST_7$ | $MSRST_6$ | $MSRST_5$ | $MSRST_4$ | $MSRST_3$ | $MSRST_2$ | $MSRST_1$ | $MSRST_0$ |
| Reset Value | 0x0f | | | | | | | |
| Access | Read / Write | | | | | | | |
| Data Type | 8-bit unsigned integer | | | | | | | |
| Usage | This register set the min number of feature to navigate | | | | | | | |

| Register: 0x70 | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Name: LiftCutoff_Calibration_Min_Squal_Threshold | | | | | | | | |
| Bit # | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Field | $STMST_7$ | $STMST_6$ | $STMST_5$ | $STMST_4$ | $STMST_3$ | $STMST_2$ | $STMST_1$ | $STMST_0$ |
| Reset Value | 0x00 | | | | | | | |
| Access | Read /Write | | | | | | | |
| Data Type | 8-bit unsigned integer | | | | | | | |
| Usage | After a successful lift cut off calibration procedure, this register provides lift cut off calibration Minimum SQUAL Run threshold | | | | | | | |

| Register: 0x71 | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Name: LiftCutoff_Calibration_Squal_Threshold | | | | | | | | |
| Bit # | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Field | $STST_7$ | $STST_6$ | $STST_5$ | $STST_4$ | $STST_3$ | $STST_2$ | $STST_1$ | $STST_0$ |
| Reset Value | 0x00 | | | | | | | |
| Access | Read /Write | | | | | | | |
| Data Type | 8-bit unsigned integer | | | | | | | |
| Usage | After a successful lift cut off calibration procedure, this register provides lift cut off calibration Minimum SQUAL threshold | | | | | | | |