

Part Number: PMW3360DM-T2QU

Application Note AN

Related Part Ordering Information

Part Number	Type
PMW3360DM-T2QU	Optical Gaming Navigation Sensor



For any additional inquiries, please contact us at <http://www.pixart.com/contact.asp>.

Contents

1.0	Introduction.....	2
1.1	General Description.....	2
2.0	Lift Cut Off Calibration	3
2.1	Procedures/Steps	3
3.0	Lift Cut Off Calibration Process Flow Chart.....	5
4.0	Sample Code.....	6
4.1	Pseudo MCU code sample.....	6

List of Figures

Figure 1.	Lift Cut Off Calibration Flow Chart	5
-----------	-------------------------------------------	---

1.0 Introduction

1.1 General Description

The PMW3360DM sensor has the capability to improve on its lift performance by tuning internal parameters to a specific surface. The procedure to perform the tuning is referred to as “Lift Cut Off Calibration”.

This application note intends to list out the procedure and method to perform Lift Cut Off Calibration to improve on the PMW3360DM sensor lift performance on a specific surface. The calibration process involves changes to the internal sensor register settings.

2.0 Lift Cut Off Calibration

2.1 Procedures/Steps

1. Take note that the Lift cut off calibration procedure that follows references registers of seven Lift cut off calibration related registers:
 - i. Register 0x4A LiftCutoff_Tune1
 - ii. Register 0x65 LiftCutoff_Tune2
 - iii. Register 0x41 LiftCutoff_Tune3
 - iv. Register 0x58 LiftCutoff_Tune_Timeout
 - v. Register 0x5A LiftCutoff_Tune_Min_Length
 - vi. Register 0x2C Pixel_Threshold
 - vii. Register 0x2B Min_SQ_Run
 - viii. Register 0x63 Lift_Config
2. Ensure that the sensor is powered up according to the Power Up Sequence as listed in Section 7.0 of the data sheet and the Lift cut off calibration SROM (e.g. SROM0x81) is downloaded into the sensor. Delay for 30ms.
3. Prompt the user that the "Lift cut off calibration" procedure is about to begin to ensure that the mouse is placed nominally on the surface (mouse is not lifted).
4. Start the calibration procedure by setting RUN_CAL register bit to 1. The calibration procedure can be started by a prompt to the user or user-initiated through a mouse-click event.
5. Poll CAL_STAT[2:0] to check the status of the calibration procedure. There are three ways to successfully stop the calibration procedure: set RUN_CAL register bit to 0 if either:
 - i. CAL_STAT[2:0] = 0x02,
 - ii. CAL_STAT[2:0] = 0x02 and user initiates a stop through a mouse-click event, or,
 - iii. CAL_STAT[2:0] = 0x03.If CAL_STAT[2:0] = 0x04, the calibration procedure needs to be re-started.
6. Stop the calibration procedure by ensuring that the RUN_CAL register bit is 0, then wait 1msec before reading the recommended "Pixel Threshold" register value, RPTH[6:0] (lower 7 bits of LiftCutoff_Tune2 register). RPTH[6:0] recommends a pixel threshold value that replaces the default value in the tracking SROM's Pixel_Threshold register to improve lift performance. The Pixel_Threshold register requires the Tracking SROM (SROM0x03 or above) to be loaded.
7. Read the recommended "Min SQUAL Run" register value, RMSQ[7:0] (entire 8 bits of LiftCutoff_Tune3 register). RMSQ[7:0] recommends a Min SQUAL Run value that replaces the default value in the tracking SROM's Min_SQ_Run register to improve lift performance. The Min_SQ_Run register requires the Tracking SROM*2 to be downloaded.
8. Set the Register Lift_Config[1:0] with binary "10" after loaded the Tracking SROM.
9. The Lift cut off calibration procedure is complete.

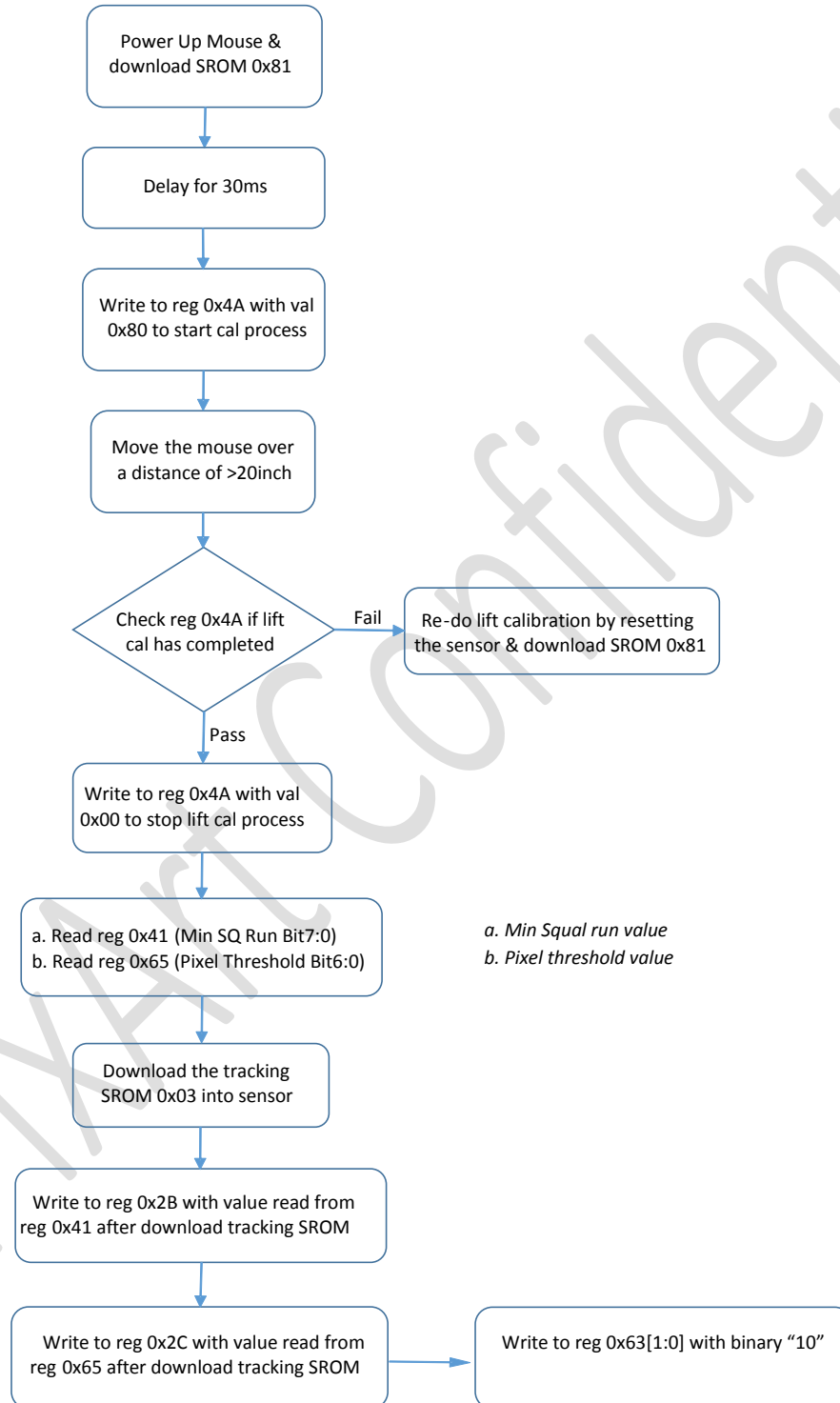
10. A special note on the available SROM versions;
Lift cut off calibration SROM: SROM 0x81 or above (size 4KB).
Tracking SROM: SROM 0x03 or above (size 4KB).

Note:

In order to convert back to the default lift performance without using lift cut off calibration, change the value in Register “Min_SQ_Run” and Register “Pixel_Threshold” to its default value, and set the Register “Lift_Config” [1:0] to original setting, either 2mm setting or 3mm setting.

3.0 Lift Cut Off Calibration Process Flow Chart

Figure 1. Lift Cut Off Calibration Flow Chart



4.0 Sample Code

4.1 Pseudo MCU code sample

```
// Return result
#define PASS 1
#define FAIL 0

// Calibration return bits in HEX
#define CAL_INPROGRESS 0x81
#define CAL_SUCCESS_1 0x82
#define CAL_SUCCESS_2 0x83
#define CAL_FAIL 0x84

// Cal bits
#define RUN_CAL_EN 0x80
#define RUN_DISABLE 0x00

// Register definition
#define SROM_ID_ADDR 0x2A
#define LCO_TUNE1_ADDR 0x4A
#define PIXEL_THRESHOLD_ADDR 0x2C
#define LCO_TUNE2_ADDR 0x65
#define LCO_TUNE3_ADDR 0x41
#define MIN_SQ_RUN_ADDR 0x2B

// SROM bytes array
extern char SROM_81[];
extern char SROM_03[];

// Download SROM, return 1 if successful, or 0 if failed
int Download_SROM(char* srom_bytes, int size);
// Read/Write register
uchar Read_Register(uchar Addr);
void Write_Register(uchar Addr, uchar Value);
// Delay function (ms)
void delay(int ms);

uchar LiftCutOff_Cal()
{
    char cal_status = 0;
    uchar pix_thresh;
    uchar min_sq;

    // Download SROM 0x81, return fail if download was not successful
    if (Download_SROM(SROM_81, sizeof(SROM_81)) == 0)
        return FAIL;

    // Delay for 30ms
    delay_ms(30);

    // Enable Calibration
    Write_Register(LCO_TUNE1_ADDRESS, RUN_CAL_EN);

    // Loop to wait for calibration to end
    while (1)
    {
        delay(10);
        cal_status = Read_Register(SURF_TUNE1_ADDRESS);

        if (cal_status != CAL_INPROGRESS)
        {
            // Done calibration
            break;
        }
    }
}
```

```
}  
// Stop Calibration  
Write_Register(SURF_TUNE1_ADDRESS, RUN_DISABLE);  
  
// If calibration failed, return and let process to decide for further action  
if (cal_status == CAL_FAIL)  
    return FAIL;  
  
// If success, delay 1 ms first before reading results  
delay(1);  
  
// read RPTH[6:0] (0x65) and write to Pixel Threshold (0x2c)  
pix_thresh = Read_Register(LCO_TUNE2_ADDR) & 0x7F;  
  
// read RMSQ[7:0] (0x41) and write to Min_SQ_Run (0x2B)  
min_sq = Read_Register(LCO_TUNE3_ADDR);  
  
// Download SROM 0x03, return fail if download was not successful  
if (Download_SROM(SROM_03, sizeof(SROM_03)) == 0)  
    return FAIL;  
  
Write_Register(PIXEL_THRESHOLD_ADDR, pix_thresh);  
Write_Register(MIN_SQ_RUN_ADDR, min_sq);  
  
// Write to reg 0x63[1:0] with binary "10"  
Write_Register(0x63, 0x02);  
  
return PASS;  
}
```