# PAT9136E1-TXQT and PAA5160E1-Q: CPI Calibration and Adjustment (AN06)

## General Description

This application note provides Count Per Inch (CPI) calibration and adjustment methodologies for PAT9136E1 and PAA5160E1.

## Ordering Information

| Part Number | Description |
|---|---|
| PAT9136E1-TXQT | Optical Tracking Chip |
| PAA5160E1-Q | Optical Tracking Chip |

For any additional inquiries, please contact us at https://www.pixart.com

# Table of Contents



## List of Figures



## List of Tables

# 1.0 Introduction

## 1.1 Overview

The PAT9136E1-TXQT and PAA5160E1-Q are PixArt Imaging's optical tracking chips with integrated VCSEL illumination. This application note provides Count Per Inch (CPI) calibration and adjustment methodologies, aim to bridge the gap between the chip's actual reported CPI value and CPI setting.

**Note:** Throughout this application note, the PAT9136E1 and PAA5160E1 are referred to as the "chip".

## 1.2 Relevant Information

Table 1. Related Document

| No. | Item | Version |
|---|---|---|
| 1 | PAA5160E1-Q Product Datasheet | 0.83 |
| 2 | PAA5160E1-Q Sensor Resolution Normalization Application Note (AN00) | 1.0 |
| 3 | PAA5160E1-Q Chip Orientation Determining and Re-mapping Application Note (AN03) | 1.0 |
| 4 | PAT9136E1-TXQT Product Datasheet | 0.83 |
| 5 | PAT9136E1-TXQT and PAA5160E1-Q: Chip Cover Glass and Protective Cover Cleaning Procedure (AN04) | 1.0 |
| 6 | PAT9136E1-TXQT and PAA5160E1-Q: Module Test Recommendation (AN05) | 1.0 |

## 1.3 Terminology

| Term | Description |
|---|---|
| CPI | Count per Inch |
| VCSEL | Vertical Cavity Surface Emitting LASER |

## 2.0 CPI Calibration & Adjustment

Generally, user expects similar count output from the chip as what has been set in the resolution registers (register address from 0x47 to 0x4B). However, due to environment factors, including but not contained to surface types and design parameters (working range for example), the actual count output from the chip might deviate from the set or theoretical value. In this case, CPI calibration and adjustment are introduced to provide a method to correlate and / or adjust the count output from the chip to the intended or desired value.

### 2.1 CPI Calibration

The CPI calibration is a process of collecting the count that outputs from the chip through repeated tests. Then, applying a factor to the average count of these datasets. The tests are made over a fixed travel distance, and swept across the desired speed and working range as required by user's application.

1. Number of Repeated Tests/ Iterations – At least 10 iterations (more is preferred) over a particular setup.
2. Working Range/ Height – At least 3 level of heights: Minimum, nominal, and maximum height.
3. Test Speed– At least 3 level of speeds: Minimum, nominal, and maximum speed.
4. Fixed Travel Distance – Factoring in the acceleration and de-acceleration profile of the test set-up, allocate enough distance for the highest test speed to track.

#### 2.1.1 Calibration Matrix

For an effective calibration process, a matrix needs to be planned to determine the test runs based on the required parameters.

A typical example is as shown below: -

- Height = 25mm (± 5 mm)
- Speed = 2m/s (± 0.5 m/s)

With these requirements, the matrix with a total of 9 test runs to be performed on the intended tracking surface as shown in below table:

| Speed (m/s) | Height (mm) | | |
|---|---|---|---|
| | 20 | 25 | 30 |
| 1.5 | Test 1 | Test 4 | Test 7 |
| 2.0 | Test 2 | Test 5 | Test 8 |
| 2.5 | Test 3 | Test 6 | Test 9 |

In addition, each test run consists of 10 iterations:

Test 1: Speed = 1.5 m/s, Height = 20 mm

| Speed (m/s) | Iterations | | | | | | | | | | Average |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | |
| 1.5 | | | | | | | | | | | *Average* |

### 2.1.2 Test Setup and Procedure

The chip should be secured at the intended height over the tracking surface and the movement of the chip to be controlled by a precise mechanical system to ensure stability and repeatability of the chip's output.

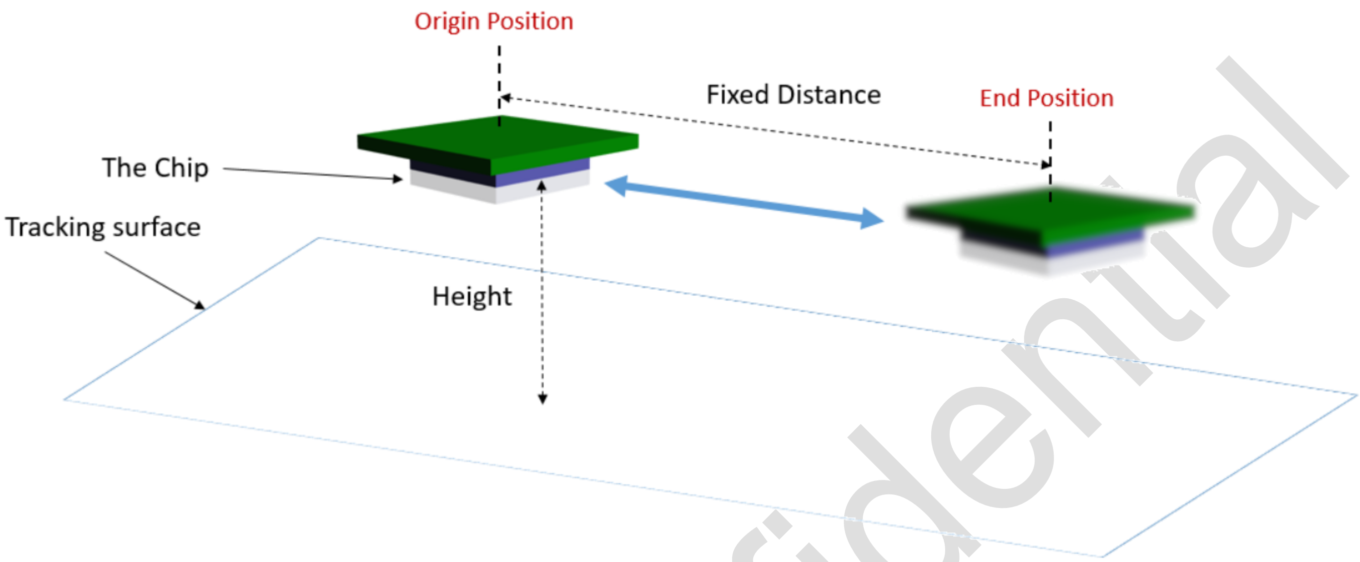


Figure 1. Illustration of Test Set-up

The test procedure is as follows:

1. Set-up the chip at the first test height and speed, according to the matrix plan.
2. With the chip placed at the "Origin Position", reset the accumulated count value.
3. Moves the chip until it reaches the "End Position" while reading and accumulate count value from the chip.
4. Calculates the CPI by dividing the accumulated count with the fixed distance in unit of inch. Record the CPI value.
5. Repeats step 1 to 4 for the remaining iterations.
6. Calculates the average CPI for all these iterations (in this example, Average1).
7. Repeats step 1 to 6, with the chip set to the second test height and speed, to acquire the next average CPI value (Average2).
8. Collects all nine average CPI values, to complete the process.

### 2.1.3 Final Average CPI

Upon completion of Section 2.1.2, the calibration matrix example is as shown below

| Speed (m/s) | Height (mm) | | |
|---|---|---|---|
| | 20 | 25 | 30 |
| 1.5 | 1956 | 1967 | 1972 |
| 2.0 | 1953 | 1962 | 1969 |
| 2.5 | 1947 | 1956 | 1968 |

**Note:** The default setting for the chip is 2000 CPI.

From these data, the average CPI for tested tracking surface is calculated to be 1961. User can use this value to calculate the actual distance travelled from the motion data reported by the chip.

Based on the above example:

- Default CPI setting = 2000
  1 mm distance travelled = 79 count [2000 count/ 25.4 mm]

- Average CPI (obtained from the calibration exercise) = 1961
  1 mm distance travelled = 77 count [1961 count/ 25.4 mm]

From this example, the actual count output from the sensor for every mm travelled is slightly less as compared to the set CPI of 2000.

If the surface range is wide, then a good guideline is to select at least a worst case surface (matte / non-reflective dark surface) and a good surface (bright glossy / reflective surface). Number of the surface selection is determined by user based on the application needs. User can perform the same calibration process on each surface and calculate the overall CPI across all surfaces to be used in conversion of counts to travelled distance. Alternatively, user can perform the same calibration process to obtain average CPI on each surface and determine the best method to apply the average CPI on their range of surfaces.

## 2.2 CPI Adjustment

To further refine the reported CPI to match the default 2000 CPI setting, user can opt for CPI adjustment, by changing the resolution registers' setting. For example, if the overall average CPI obtained from calibration is only 1961; by varying the register setting, the reported CPI from the chip can be adjusted to be closer to the CPI of 2000.

### 2.2.1 Adjustment Procedure

First, calculate the adjustment factor by dividing the CPI setting with the Overall Average CPI obtained from calibration. Based on the example in this document:

$$\begin{aligned} Adjustment\ Factor &= \frac{CPI\ setting}{Overall\ Average\ CPI} \\ &= \frac{2000}{1961} \\ &= 1.0198 \end{aligned}$$

There is an internal scaling factor for CPI calculation within the chip, and the default value is 1938. This setting needs to be adjusted by multiplying the default value with the Adjustment Factor, and then written into the chip through register setting.

$$\begin{aligned} Scaling\ Factor &= Adjustment\ Factor \times 1938 \\ &= 1.0198\ \times 1938 \\ &= 1976.37 \end{aligned}$$

The Register Setting accepts round number only, so it is either 1976 or 1977. In this example, the round up number (1977) is used.

The Scaling Factor is represented by two 16-bit registers where it needs to be separated into two bytes; namely the low byte (SCALING_LO) and high byte (SCALING_HI).

To write in the Scaling Factor, ensure that the Performance Optimization Setting in the respective chip's datasheet is written first, and then followed by below register writes:

1. Write register 0x7F with value 0x0D
2. Write register 0x71 with SCALING_LO
3. Write register 0x70 with SCALING_HI
4. Write register 0x73 with SCALING_LO
5. Write register 0x72 with SCALING_HI
6. Write register 0x7F with value 0x00
7. Write register 0x47 with value 0x01

In this example (value 1977), SCALING_LO = 0xB9 and SCALING_HI = 0x07.

The above register writes operation needs to be performed in that exact sequence without any other SPI transaction in between.

## Revision History

| Revision Number | Date | Description |
|---|---|---|
| 1.0 | 15 Mar 2024 | Initial release |