對外秘

Time-of-Flight Sensor

Calibration Guide

Revision 1.0

Mar. 2019

SAMSUNG ELECTRONICS

Beyond You Can Imagine

對外秘

Important Notice

Samsung Electronics Co. Ltd. ("Samsung") reserves the right to make changes to the information in this publication at any time without prior notice. All information provided is for reference purpose only. Samsung assumes no responsibility for possible errors or omissions, or for any consequences resulting from the use of the information contained herein.

This publication on its own does not convey any license, either express or implied, relating to any Samsung and/or third-party products, under the intellectual property rights of Samsung and/or any third parties.

Samsung makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does Samsung assume any liability arising out of the application or use of any product or circuit and specifically disclaims any and all liability, including without limitation any consequential or incidental damages.

Customers are responsible for their own products and applications. "Typical" parameters can and do vary in different applications. All operating parameters, including "Typicals" must be validated for each customer application by the customer's technical experts.

Samsung products are not designed, intended, or authorized for use in applications intended to support or sustain life, or for any other application in which the failure of the Samsung product could reasonably be expected to create a situation where personal injury or death may occur. Customers acknowledge and agree that they are solely responsible to meet all other legal and regulatory requirements regarding their applications using Samsung products notwithstanding any information provided in this publication. Customer shall indemnify and hold Samsung and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, expenses, and reasonable attorney fees arising out of, either directly or indirectly, any claim (including but not limited to personal injury or death) that may be associated with such unintended, unauthorized and/or illegal use.

WARNING No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electric or mechanical, by photocopying, recording, or otherwise, without the prior written consent of Samsung. This publication is intended for use by designated recipients only. This publication contains confidential information (including trade secrets) of Samsung protected by Competition Law, Trade Secrets Protection Act and other related laws, and therefore may not be, in part or in whole, directly or indirectly publicized, distributed, photocopied or used (including in a posting on the Internet where unspecified access is possible) by any unauthorized third party. Samsung reserves its right to take any and all measures both in equity and law available to it and claim full damages against any party that misappropriates Samsung's trade secrets and/or confidential information.

警 告 本文件仅向经韩国三星电子株式会社授权的人员提供，其内容含有商业秘密保护相关法规规定并受其保护的三星电子株式会社商业秘密，任何直接或间接非法向第三人披露、传播、复制或允许第三人使用该文件全部或部分内容的行为（包括在互联网等公开媒介刊登该商业秘密而可能导致不特定第三人获取相关信息的行为）皆为法律严格禁止。此等违法行为一经发现，三星电子株式会社有权根据相关法规对其采取法律措施，包括但不限于提出损害赔偿请求。

Samsung Electronics Co., Ltd.
San #24 Nongseo-Dong, Giheung-Gu
Yongin-City, Gyeonggi-Do, Korea 446-711

Contact Us: *kh74.bae@samsung.com*

Home Page: *http://www.samsungsemi.com*

SAMSUNG ELECTRONICS

Beyond You Can Imagine

Revision History

Revision No.	Date	Detailed of Modification	Drafter
1.0	Mar. 30, 2019	Initial Calibration Guide	K. Bae, H. Ko, D. Seo, M. Kang, C. Lee, M. Bae, S. Son, M. Keel

Contents

1 OVERVIEW 6

1.1 Systematic Errors 6
1.1.1 Wiggling 6
1.1.2 Fixed Phase Pattern Noise (FPPN) 6
1.1.3 Temperature Drift 6
1.1.4 Lens 7
1.2 Overall Calibration Procedure 7

2 CALIBRATION 10

2.1 Temperature Drift 10
2.1.1 Procedure 10
2.1.2 Equipments 11
2.2 Wiggling 13
2.2.1 Procdure 14
2.2.2 Equipments 15
2.2.3 Wiggling Calibration Board 17
2.3 Lens-FPPN 23
2.3.1 Procedure 23
2.3.2 Equipments 26

3 SOFTWARE 28

3.1 Overview 28
3.2 Temperature Drift Monitoring Tool 28
3.2.1 Usage 29
3.2.2 Deliverables 31
3.2.3 Temperature Drift Monitor Configuration file 32
3.2.4 Temperature Drift Monitor DLL API 32
3.2.5 Temperature Drift Monitor Output 37
3.3 Calibration Library 37
3.3.1 Deliverables 37
3.3.2 API 38
3.3.3 Calibration Configuration file 38
3.3.4 Calibration Result Output 42
3.4 DEPS Tool 42

REFERENCES 43

List of Figures

Figure Number | **Title** | **Page Number**

Figure 1.1 Systematic Error Sources 6
Figure 1.2 Overall Calibration Procedure 7
Figure 1.3 M2M Calibration Procedure 8
Figure 1.4 Overall Compensation Procedure 9
Figure 1.5 Example of Overall Compensation Procedure 9
Figure 2.1 Temperature Drift Calibration Procedure 11
Figure 2.2 Example of temperature drift compensation envitonment using a plane chart. Adjustable height stage mount for module (400~700mm) which is combined with TEC. Height will be fixed for a manufacturing environment. 12
Figure 2.3 Temperature Calibration Board 12
Figure 2.4 Comparison between a moving chart ((a) and (b)), and a virtual chart by time-delay ((c) and (d)) 14
Figure 2.5 Wiggling Calibration Procedure 15
Figure 2.6 Wiggling Calibration Chart. Height adjustable stage mount for module (400 – 700 mm). Height will be fixed for a manufacturing environment. 16
Figure 2.7 The Wiggling Calibration Board. 16
Figure 2.8 Conceptual Diagram of ToF Wiggling Calibration System using Time-delay 17
Figure 2.9 Pin Connectiion of the Module to Support External Time-delay 17
Figure 2.10 Module Connectivity: (a) on the Calibration board and (b) on the Phone PCB 18
Figure 2.11 Schematic of Wiggling Calibration Board with Control pins 18
Figure 2.12 Manufactured Wiggling Calibration Board 19
Figure 2.13 Wiggling Calibration Board connected to the Arduino Board 20
Figure 2.14 Evaluation Environment of Wiggling Calibration Board 20
Figure 2.15 Lens-FPPN Calibration Procedure 24
Figure 2.16 Lens Calibration Procedure. 24
Figure 2.17 FPPN Calibration Procedure; each rotation. 25
Figure 2.18 FPPN Calibration Procedure; merge all rotations. 26
Figure 2.19 Lens-FPPN Calibration Chart. Height adjustable stage mount for module (375 – 825 mm) and an angle adjustable chart (20º – 40º). Height and angle will be fixed for a manufacturing environment. 27
Figure 3.1 Released Softwares; 1) Temperature Drift Monitoring Tool, 2) Calibration Library, 3) Validation Tool. 28
Figure 3.2 Temperature Drift Calibration Block Diagram 28
Figure 3.3 Temperature Drift Monitor GUI 29
Figure 3.4 Temperature Drift Monitor Options 29
Figure 3.5 Set Path Dialog 30
Figure 3.6 Temperature drift offset from reference depth caused by self-induced heating 30
Figure 3.7 Display Depth Map 31
Figure 3.8 Display Information 31
Figure 3.9 Temperature Drift Monitor Configuration XML example 32
Figure 3.10 Calibration Configuration XML example 39

1 Overview

A system calibration is a process to reduce the effect of the systematic errors in a ToF system, as shown in Figure 1.1. This document aims to present calibration methods, equipments, and softwares.

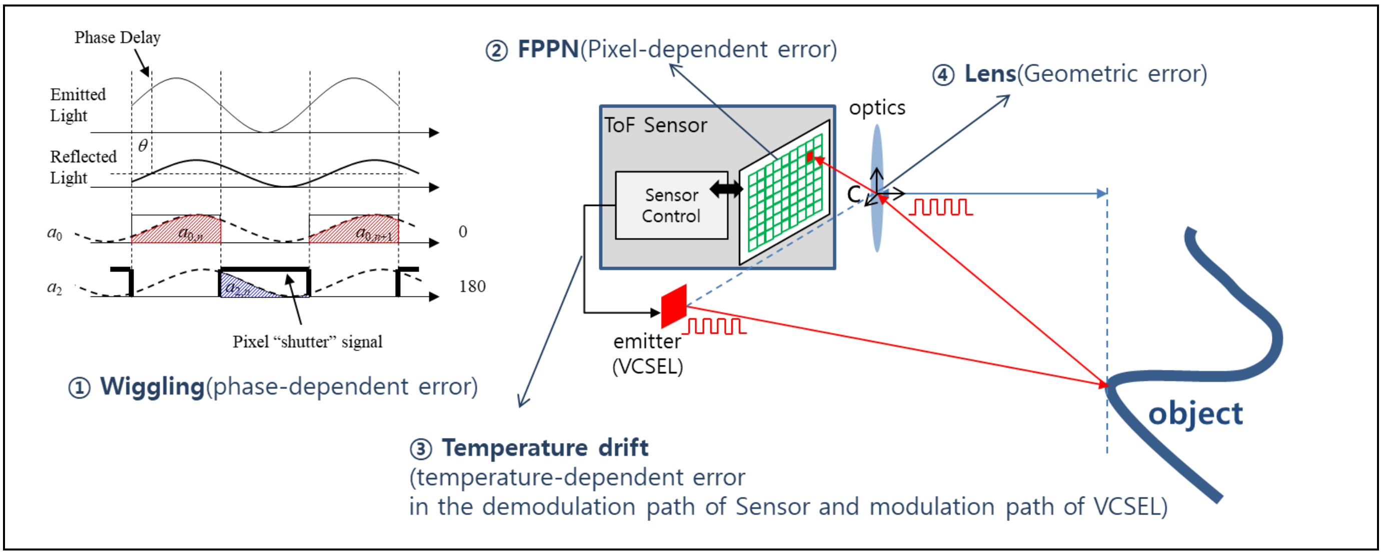

Figure 1.1 Systematic Error Sources

1.1 **Systematic Errors**

1.1.1 **Wiggling**

Phase (distance)-dependent error due to harmonic distortion.

1.1.2 **Fixed Phase Pattern Noise (FPPN)**

Pixel-dependent error due to the time-delay of demodulation signal depending on invididual pixel position and the misalignment of a VCSEL and a sensor (which is geometric error).

1.1.3 **Temperature Drift**

Temperature-dependent error because the time-delay with respect to temperature can be changed in the demodulation path of a sensor and in the modulation path of a VCSEL.

1.1.4 **Lens**

Geometric error due to lens distortion, shift, and tilt.

1.2 **Overall Calibration Procedure**

There are two types of ToF calibration

- One-time calibration: temperature drift calibration is typically performed only one time.
- Module to Module (M2M) calibraiton: wiggling, lens, and FPPN have per-module variation. Each module must be calibrated.

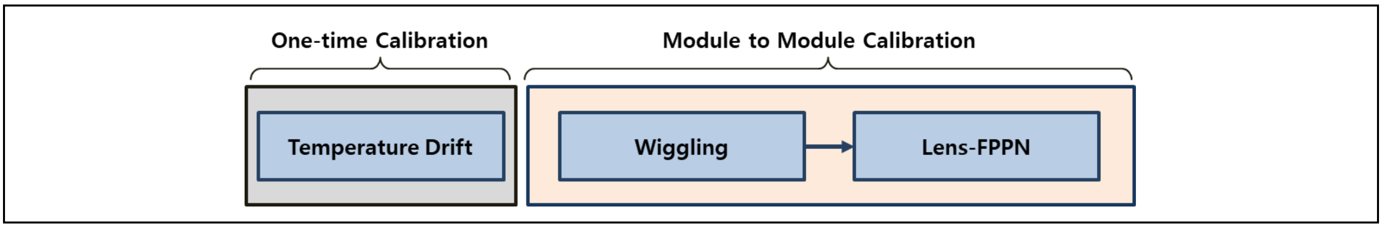

Figure 1.2 Overall Calibration Procedure

One-time calibration

1) Calibrate wiggling and lens-FPPN at constant temperature after preheating the module for 10 minutes.
2) Obtain the reference depth using the data from wiggling and FPPN calibration.
3) Obtain the statistics of temperature drift depending on the time and ambient temperature.
4) Calculate the coefficients using the statistics.

M2M Calibration

: Druing the M2M calibration, the temperature in a sensor and a laser driver will change over time. Tempeature drift compensation is performed at each calibration.

1) Calibrate wiggling using the wiggling calibration board <u>**with the external time-delay.**</u>
2) Calibrate lens-FPPN using the lens-FPPN calibraiton board <u>**without the external time-delay (normal mode).**</u>

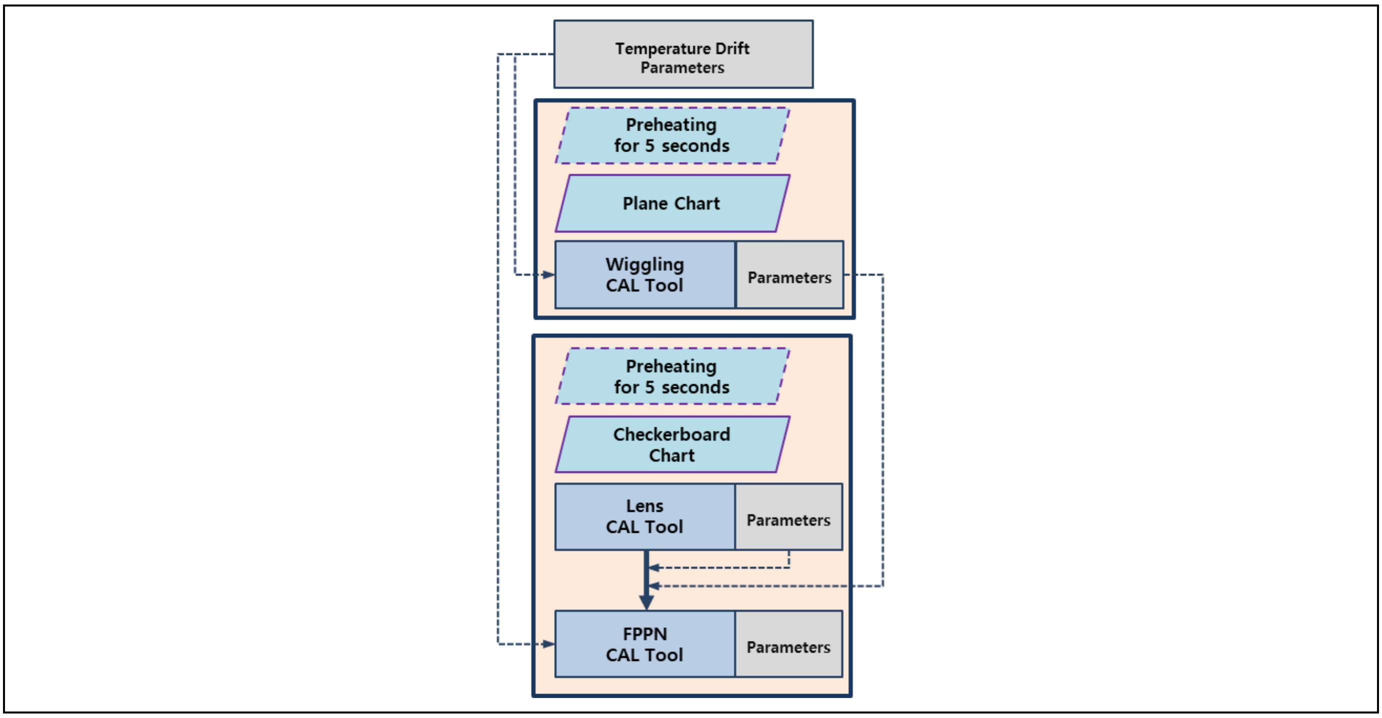

Figure 1.3 M2M Calibration Procedure

Table 1.1 M2M Calbiration Processing Time (conditions: dual modulation frequencies; 100MHz & 80MHz, PC Spec.; i7 3.6GHz, RAM 16GB, frame grabber; Simmian MV3 board).

Calibration items	preheating time (ms)1)	data capturing time (ms)	data loading and processing time (ms)	Num of data (depth/frame)5)
wiggling	5,000	3,500 2)	1,500	20/80
lens-FPPN	5,000	40,000 3)	7,800 4)	8/32
total	10,000	43,500	9,300	28/112
	62,800			

1) Preheating time can be reduced depending on the temperature drift performance.
2) Wiggling data capturing time measured with Samsung's Wiggling Calibratoin Board and Simmian board.
3) 8 different views of a plane chart (0, 10, 90, 100, 180, 190, 270, 280º) are assumed. Lens-FPPN data capturing time depends on the speed of rotation and stabilization of motorized stage.
4) If 4 different views of a plane chart are used, the data loading and processing time is reduced to 5,000ms.
5) One depth requires 4 raw frames.

Figure 1.4 shows the compensation procedure using calibration data. Figure 1.5 represents the example of overall compensation procedure.

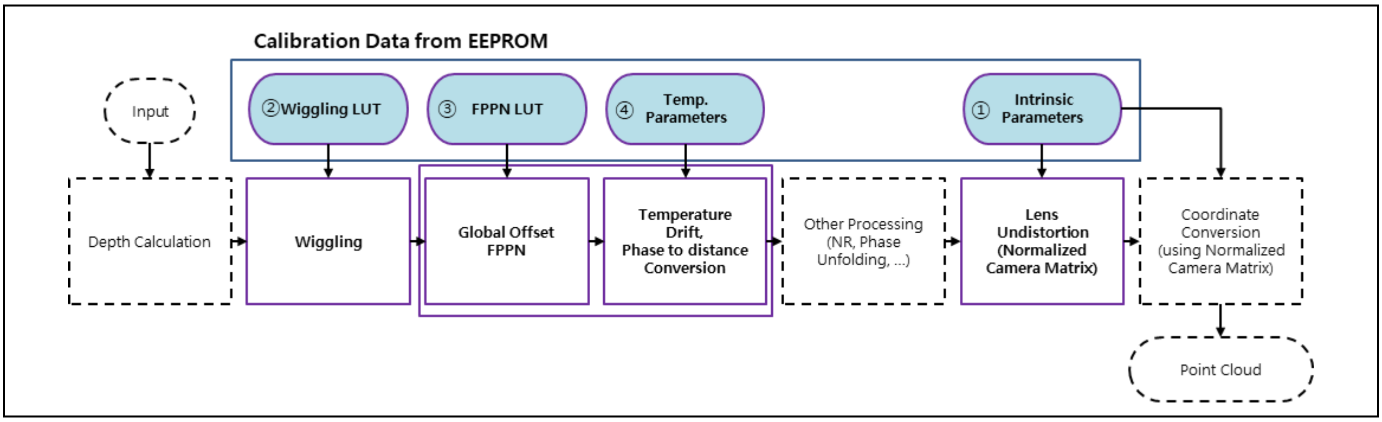

Figure 1.4 Overall Compensation Procedure

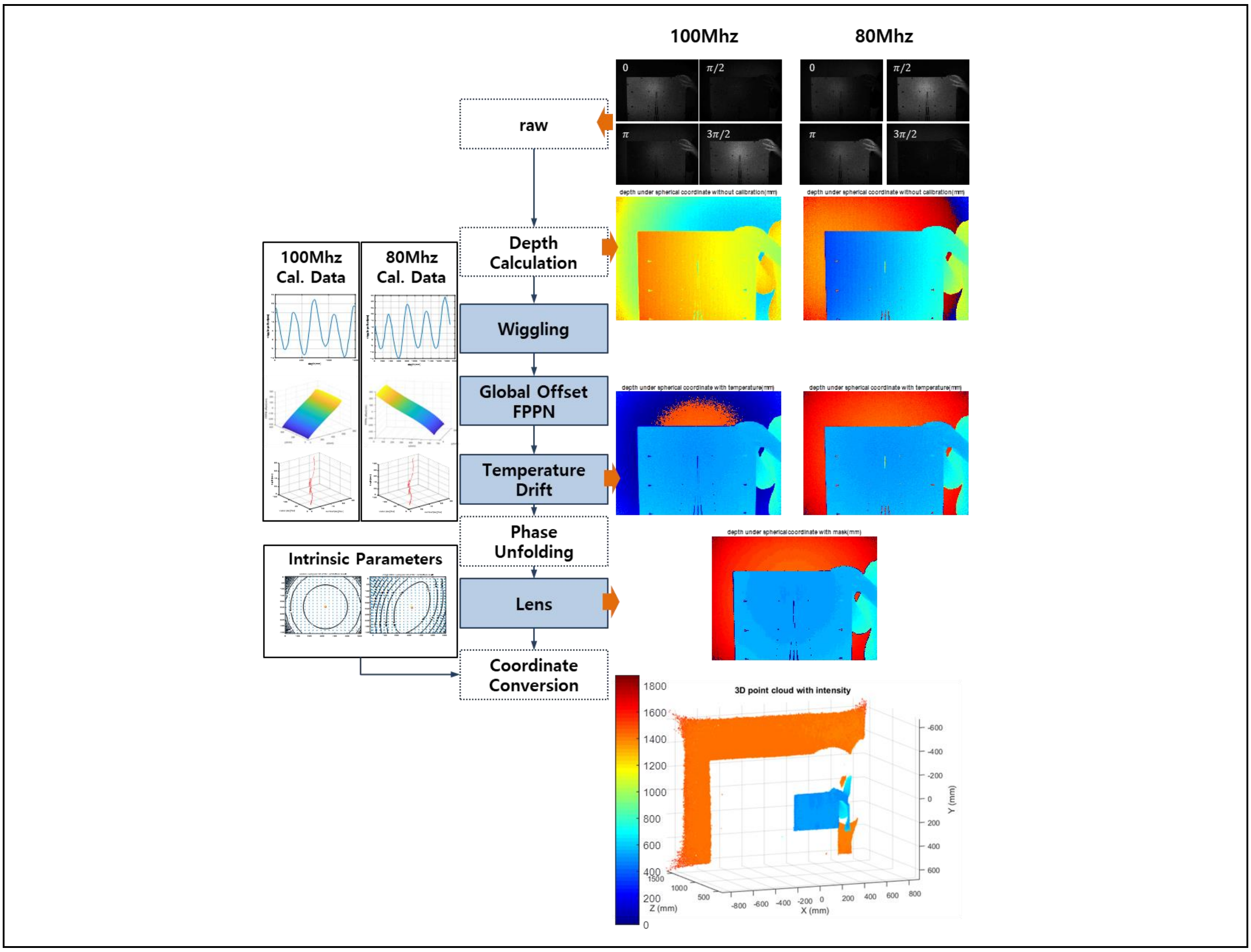

Figure 1.5 Example of Overall Compensation Procedure

2 Calibration

2.1 Temperature Drift

Temperature can change the depth offset, because the delay time in the VCSEL modulation and sensor demodulation path can be changed with temperature. Therefore, the distance offset compensation with temperature drift should be applied to the original distance, based on the temperatures in a sensor and a laser driver (here, we assume temperature of the VCSEL is represented by a laser driver's temperature). Compensation offset can be modeled as follows:

$$compnesation\ \ offset = \ a \times temp_{sensor} + b \times temp_{vcsel} + c$$

where, $temp_{sensor}$ and $temp_{vcsel}$ are the temperature data of a sensor and a laser driver, respectively, which can be obtained in the embedded line of sensor raw data (see more detailed information from the application note of the ToF sensor). The coefficients, a, b, and c are calculated using monitoring tool and calibration library with respect to the temperature data. Temperature drift calibration is typically performed only one time from mass data, but not in M2M.

Temperature drift statistics can be obtained by "Temperature Drift Monitoring Tool" depending on time and ambient temperature. Ambient temperature may be changed while obtaining statistics. "Calibration library" calculates the coefficients from this statistics.

2.1.1 Procedure

Temperature Drift Calibration Procedure (see Figure 2.1)

1) Select a camera module for temperature drift calibration.

2) Preheating for 10 minutes before calibrating the reference module for a) and b).

a) Wiggling Calibraion

b) Lens-FPPN Calibration

3) Set ambient temperature as 25, 35, 45 and 55℃

a) Obtain the temperature statistics for 30 minutes at each ambient temperature.

: "Temperature Drift Mointoring Tool" can obtain the statistics of temperature drift.

This tool uses the Wiggling and FPPN calibration data.

4) Calculate coefficients, a, b, and c using "Calibraiton Library", wihch will be written in the memory (e.g. EEPROM) on the all camera modules.

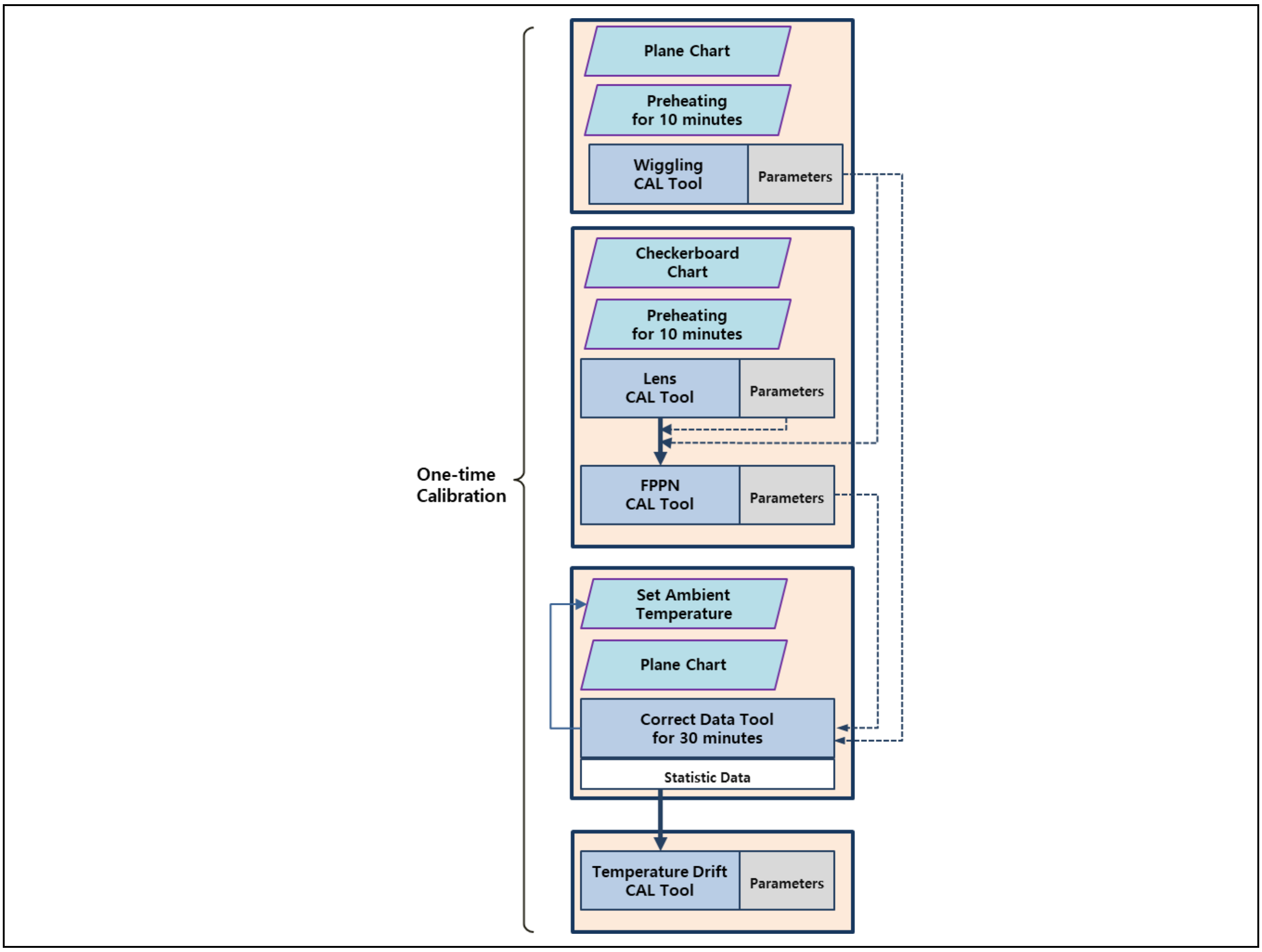

Figure 2.1 Temperature Drift Calibration Procedure

2.1.2 **Equipments**

For the temperature drift calibration, a plane chart whose reflectivity is higher than 80% is used in common. The distance from a module and a plane chart is fixed by monitoring the code to prevent saturation. This distance would be about 400 – 700 mm depending optical power of the VCSEL, integration time, and chart reflectivity.

Temperature drift calibration requires a sensor and a laser driver to be held at a specific stable temperature. For this purpose, a thermal chamber may be a candidate for adjusting temperature. However, size limitation of the thermal chamber can cause some problems. A small-size chamber will cause multiple reflection because a plane chart cannot be allowed to put into the chamber, and large-size chamber is costly and occupies huge space.

We recommend a thermoelectric cooler (TEC) to regulate the calibration temperature. A TEC is a solid state device which can control heat flux using electric current; it is composed of a precision temperature controller and a base plate, as shown in Figure 2.2. Highly thermal conductive materials such as copper or aluminum will be used as the base plate for the chuck or thermal platform. And the TEC chips is directly connected to the base plate to control temperature fast and in bi-directional way. This allows for temperature settings above and below the ambient temperature. The TEC is particularly useful in small scale temperature control, providing fast temperature response and ultra-high temperature stability. Under low temperature test, it is important to create moisture free environment [1].

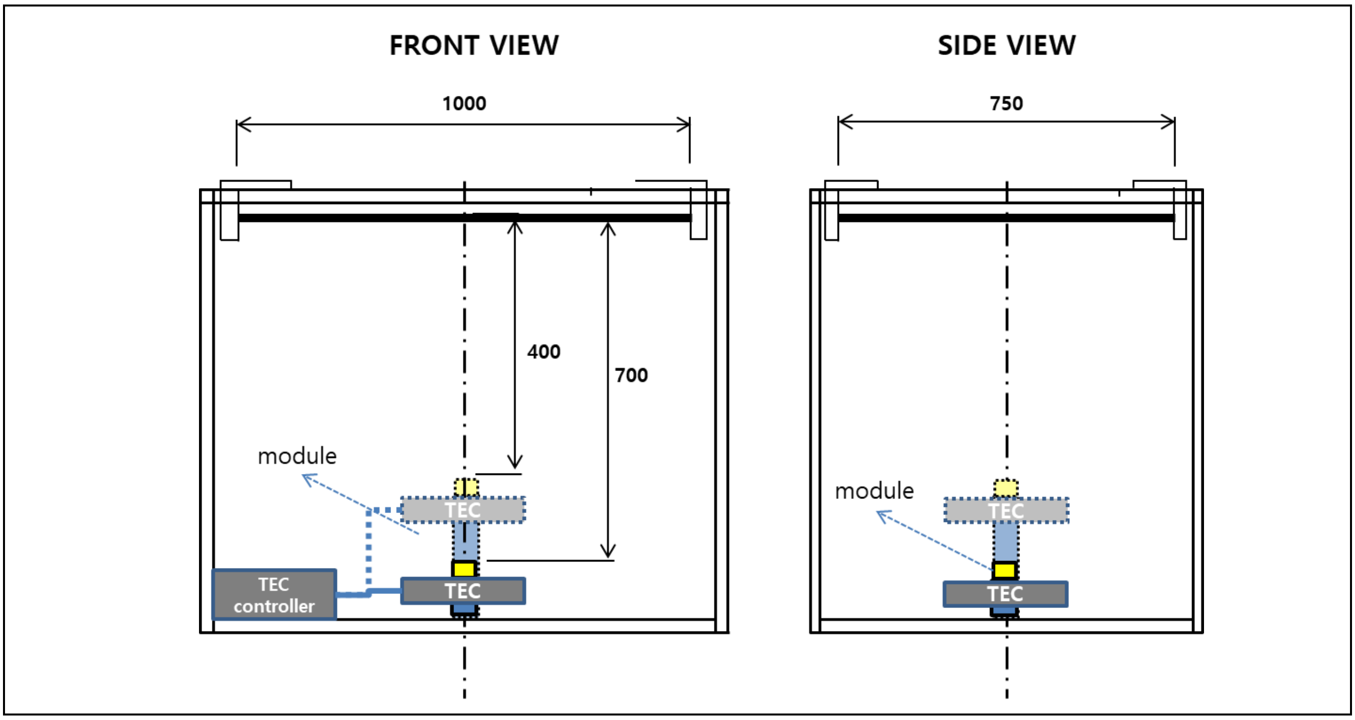

Figure 2.2 Example of temperature drift compensation envitonment using a plane chart. Adjustable height stage mount for module (400~700mm) which is combined with TEC. Height will be fixed for a manufacturing environment.

When calibrating temperature drift, “Lens-FPPN & Temperature Drift Calibration Board” <u>without the external time-delay</u> as shown in Figure 2.3 has to be used.

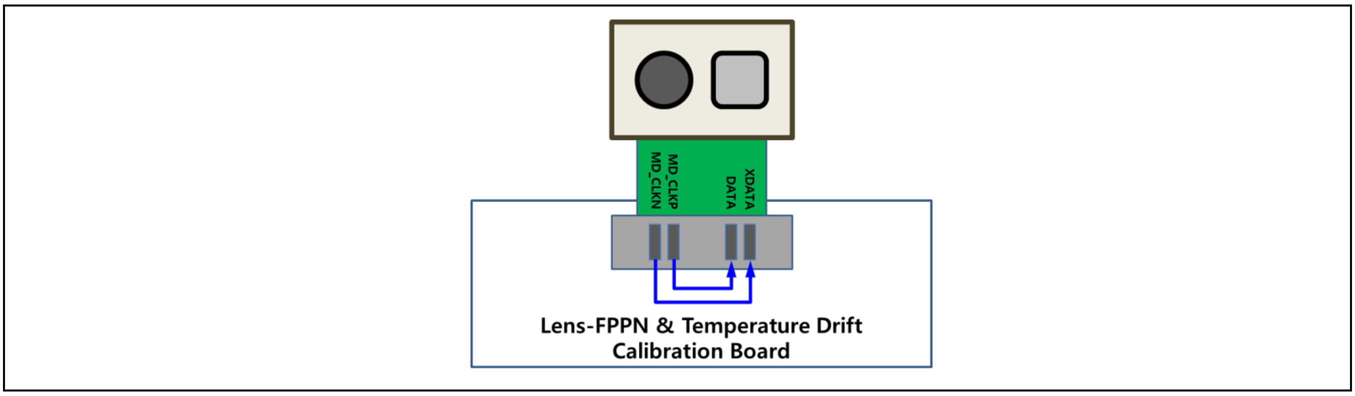

Figure 2.3 Temperature Calibration Board

2.2 Wiggling

The wiggling error, also called as circular error, is a known systematic sinusoidal depth distortion. The error varies with the distance. Note that since the wiggling error is a nonlinear function of distance, in general, a look up table (LUT) is used; the residual error between the measured depth and the groud truth (GT) will be pre-calculated in the calibration procedure. In the conventional approach, the wiggling calibration is performed by a moving chart on a measurement track line. But in the mass production, this method can not be used because of the space and capture time limitation, even though only the center region of interest (ROI) is used.

We will use an external time-delay board and a simple plane chart to calibrate wiggling error. The procedure does not require moving chart on a measurement track line. A white plane chart in a fixed distance is sufficient. This method reduces calibration time and space for the mass production. It is possible to use only small data set. The time delay in the modulation signal for the laser driver should be adjusted by an external time-delay board in order to add a phase shift between the modulation and the demodulation of the light signal. Delay in the modulation path is equivalent to a distance shift of the object at an optical center.

Note that the raw data of the charts obtained by an external time-delay board will be different from those obtained by a moving chart as shown in Figure 2.4. In case of the moving chart, the distance from the sensor and the chart increases gradually in z-direction on the Cartesian coordinates system of the sensor (see Figure 2.4 (a) and (b)). However, when using the time-delay circuit, the virtual distance from the sensor and the chart is consistently adjusted in r-direction on the Spherical coordinates system, but not in z-direction on the Cartesian coordinates system. Therefore, the resultant raw data looks like capturing curved surfaces as shown as in Figure 2.4 (b) and (d).

For the test of this method, the precision time-delay ICs such as NB6L295 from On-semiconductor® are used. The NB6L295 is a dual channel programmable delay chip [2]; the main specifications are 0 – 11.2 ns total time delay (extended mode), 11ps time delay per step, 511 step (9-bit control), linearity of +/-20ps (max.). The delay time can be digitally controlled by an MCU board such as Arduino. A single time step of the NB6L295 is equivalent to about 1.54 mm distance shift. Since the time-delay IC contains 2048 steps when used by two chips in series, the maximum delay represents a distance shift of about 3.1478m.

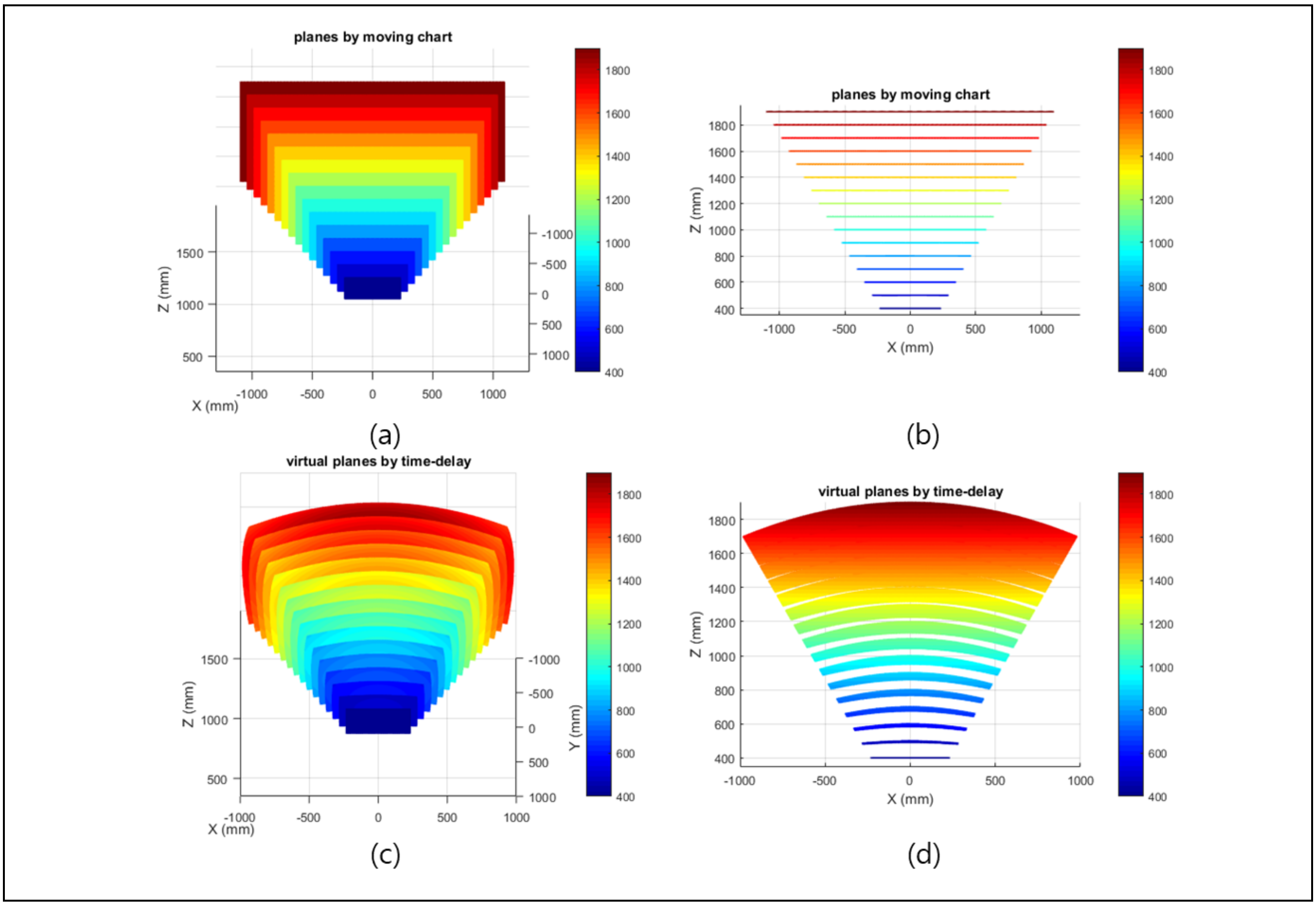

Figure 2.4 Comparison between a moving chart ((a) and (b)), and a virtual chart by time-delay ((c) and (d))

2.2.1 **Procdure**

Wiggling Calibration Procedure (see Figure 2.5)

1) Set a module to be able to capture a plane chart

2) Preheat the module for 5 seconds.

3) Set the number of the steps to specify the number of images used in calibration (default, 20)

5) Raw data is captured automatically.

6) Calculate wiggling LUT using Calibraiton Library.

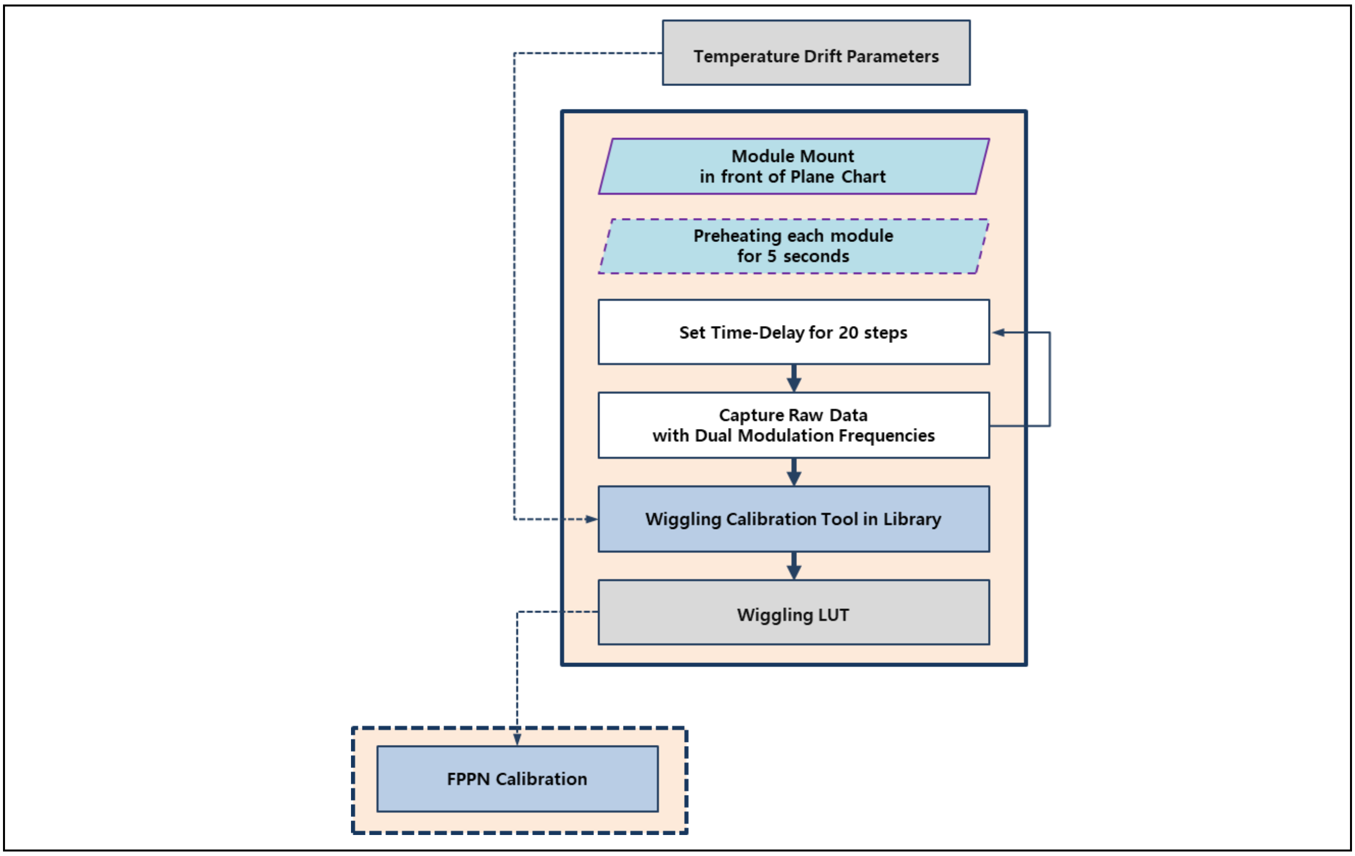

Figure 2.5 Wiggling Calibration Procedure

2.2.2 **Equipments**

For the wiggling calibration, a white plane chart can be used. The reflectivity of plane chart is more than 80%. The distance from a module and a plane chart is fixed by monitoring the code to prevent saturation. This distance is about 400 – 700 mm depending optical power of VCSEL, integration time, and chart reflectivity.

Table 2.1 The Specification of Wiggling Calibration Chart.

Chart type	White Plane
Chart size	1000 mm × 750 mm
Material (lambertian reflectivity)	> 80%
Distance from module	400 – 700 mm

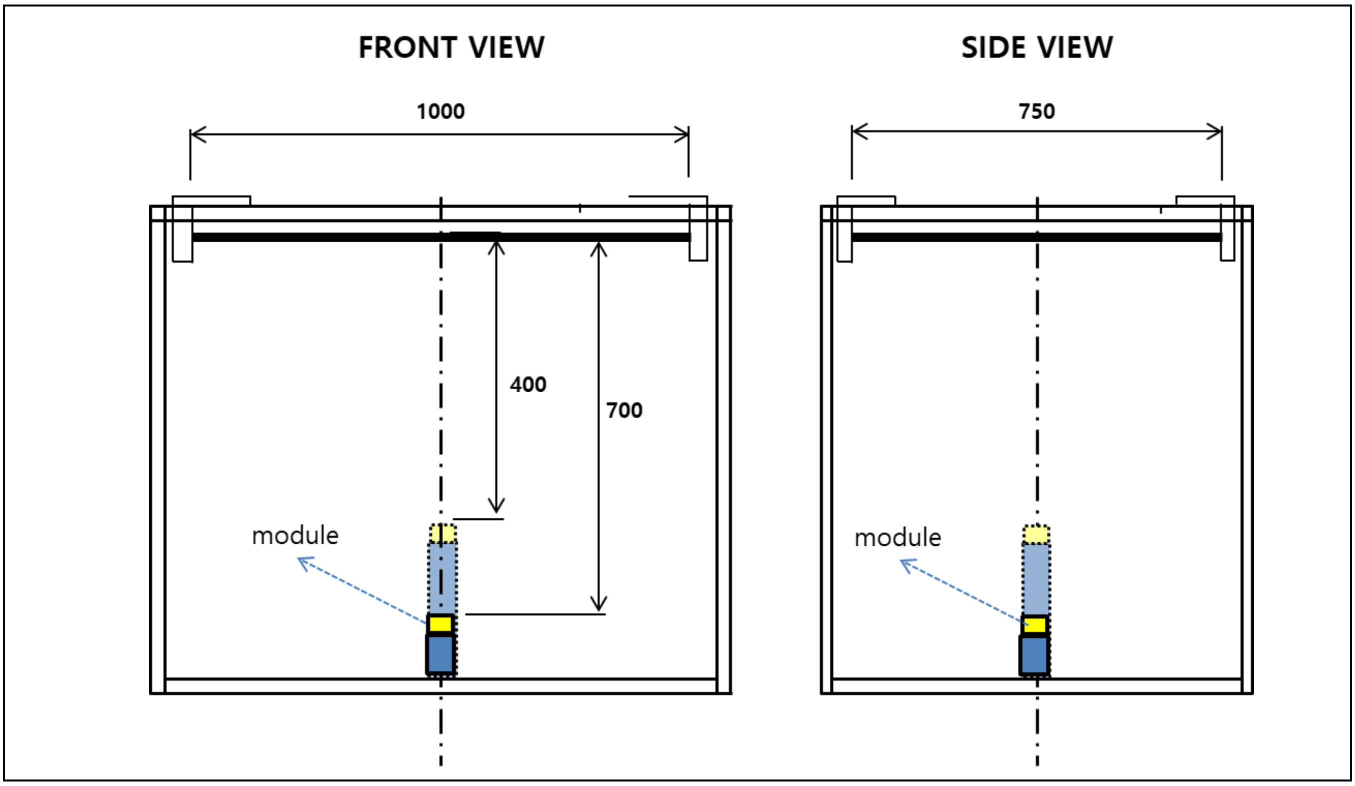

Figure 2.6 Wiggling Calibration Chart. Height adjustable stage mount for module (400 – 700 mm). Height will be fixed for a manufacturing environment.

When calibrating wiggling, “Wiggling Calibration Board” <u>with the external time-delay</u> as shown in Figure 2.7 has to be used.

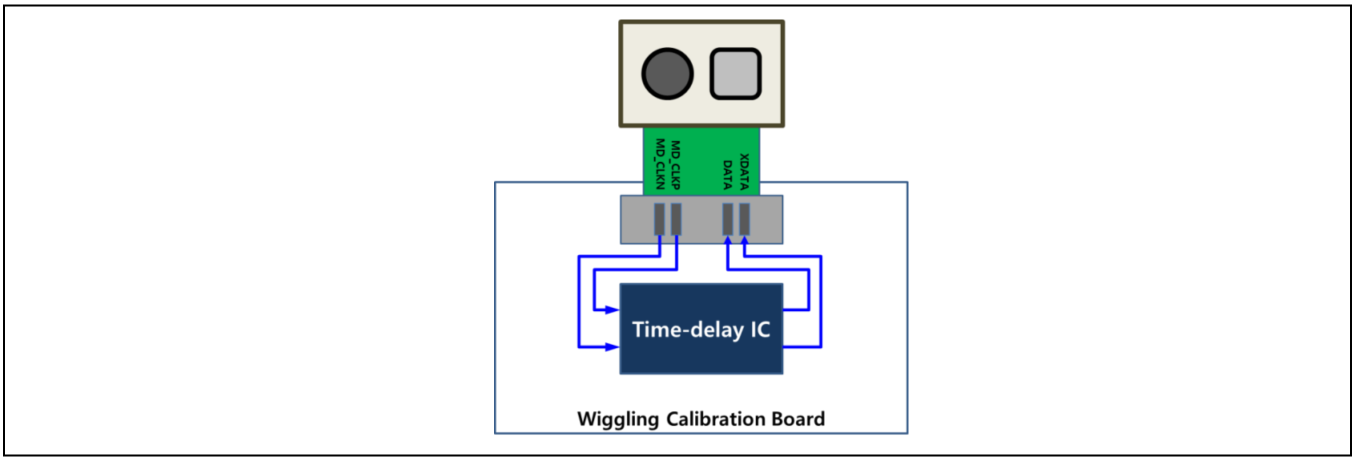

Figure 2.7 The Wiggling Calibration Board.

2.2.3 **Wiggling Calibration Board**

The Wigging Calibration Board is a board to generate the reference distance required for Wiggling calibration of a ToF system. Wiggling Calibration board inserts a time delay corresponding to the reference distance between RX (sensor) and TX (laser driver) using a high-precision time delay IC. Figure 2.8 shows conceptual diagram of ToF Wiggling Calibration System using time-delay.

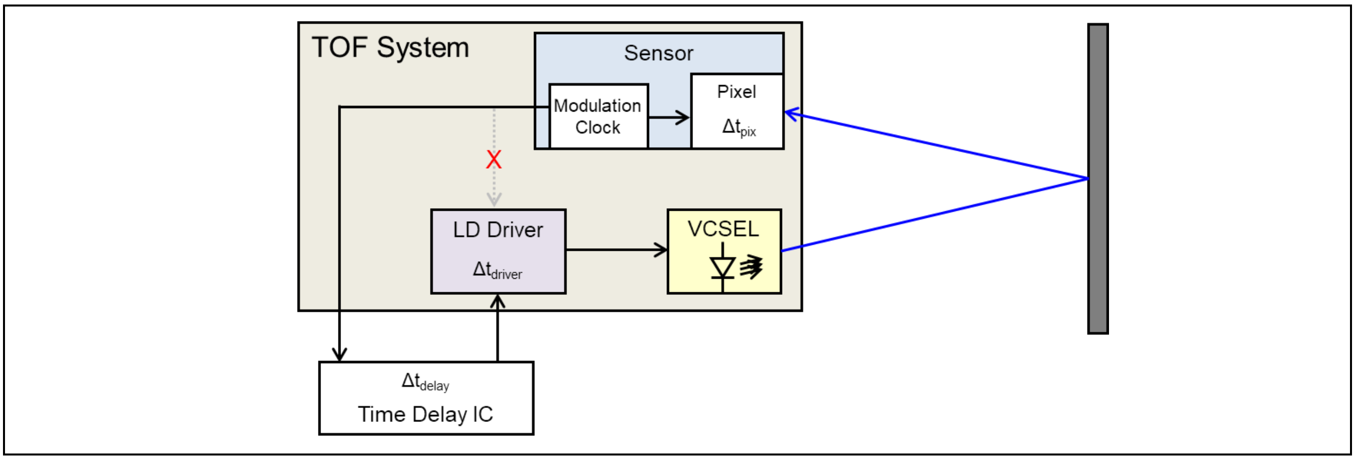

Figure 2.8 Conceptual Diagram of ToF Wiggling Calibration System using Time-delay

<u>Module build for Wiggling Calibration Board</u>

The ToF camera module must be modified to insert the Wiggling Calibration Board between TX and RX sides. Both the modulation signal outputs of the ToF sensor and the inputs of the laser driver IC must be out through the connector pins, as shown Figure 2.9.

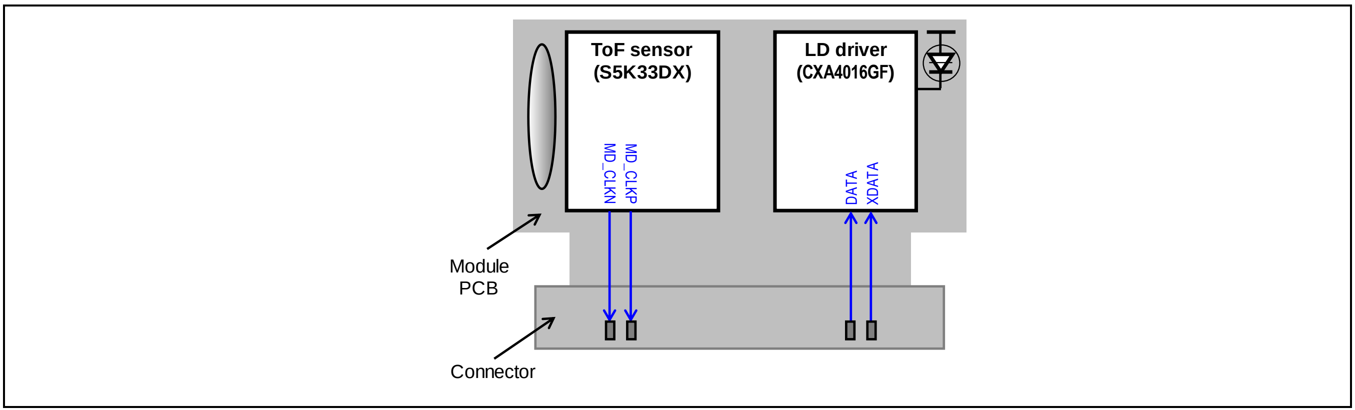

Figure 2.9 Pin Connectiion of the Module to Support External Time-delay

Figure 2.10 shows the signal connection with the Wiggling Calibration Board and on the phone PCB. Delay de-embedding will be required for calibration of the Wiggling Calibration Board's inherent delay (red lines in Figure 2.10(a)). Additional delay calibration by connections on the phone PCB (blue lines in Figure 2.10 (b)) may not be required (the additional delay will be sufficiently small because it is much shorter than other traces).

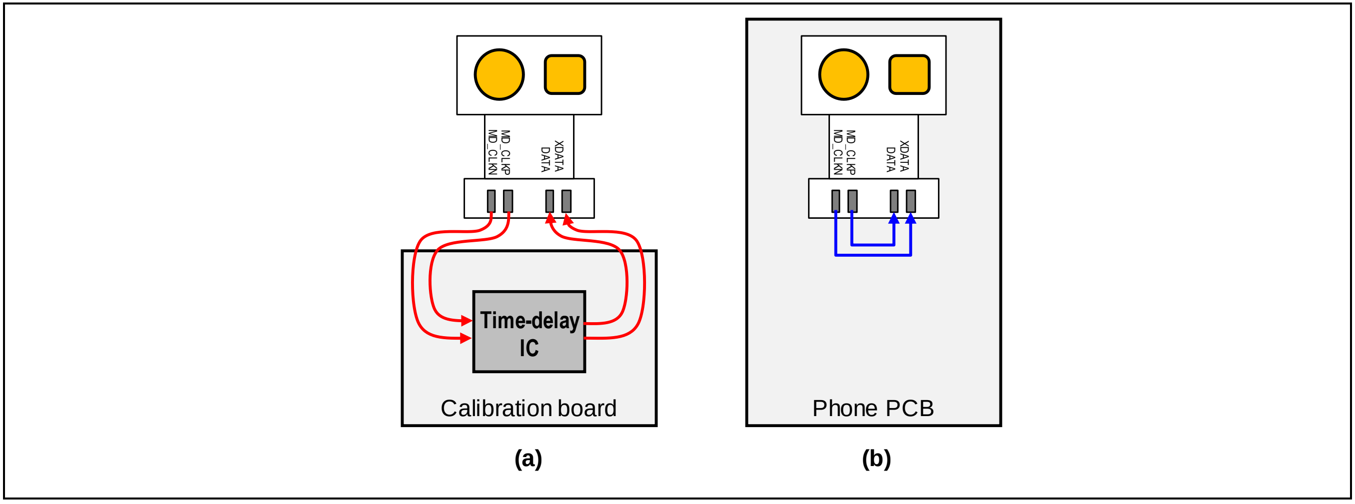

Figure 2.10 Module Connectivity: (a) on the Calibration board and (b) on the Phone PCB

Board Configuration

Figure 2.11 shows a schematic of the Wiggling Calibration Board. Because the NB6L295 outputs signals in LVPECL signaling and the laser driver has an LVDS input, LVPECL-to-LVDS interface are required, as shown in the right hand side of Figure 2.11. The schematic diagram does not include a power domain that generates 3.3 V using 5 V External power. The power domain circuit configuration may be designed by a voltage regulator IC. And two NB6L295 ICs were connected in series to cover long time range.

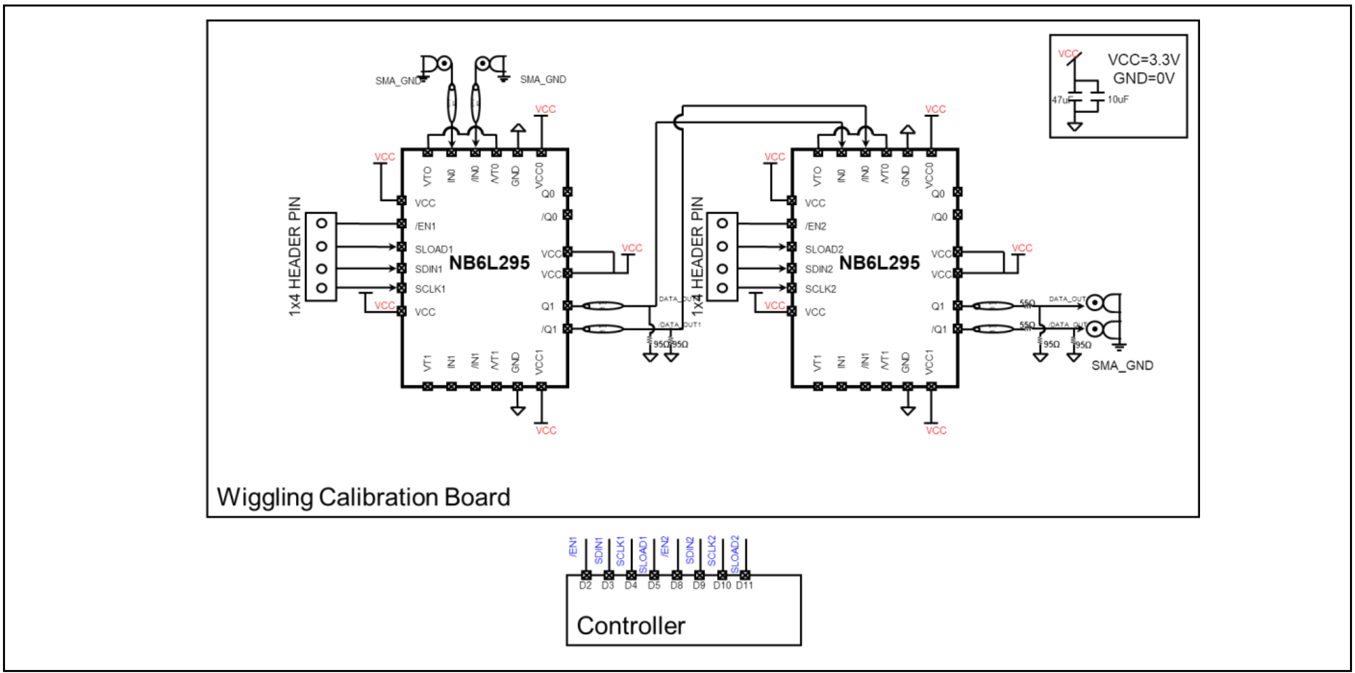

Figure 2.11 Schematic of Wiggling Calibration Board with Control pins

Board Manufacture Guide

Figure 2.12 shows a manufactured Wiggling Calibration Board.

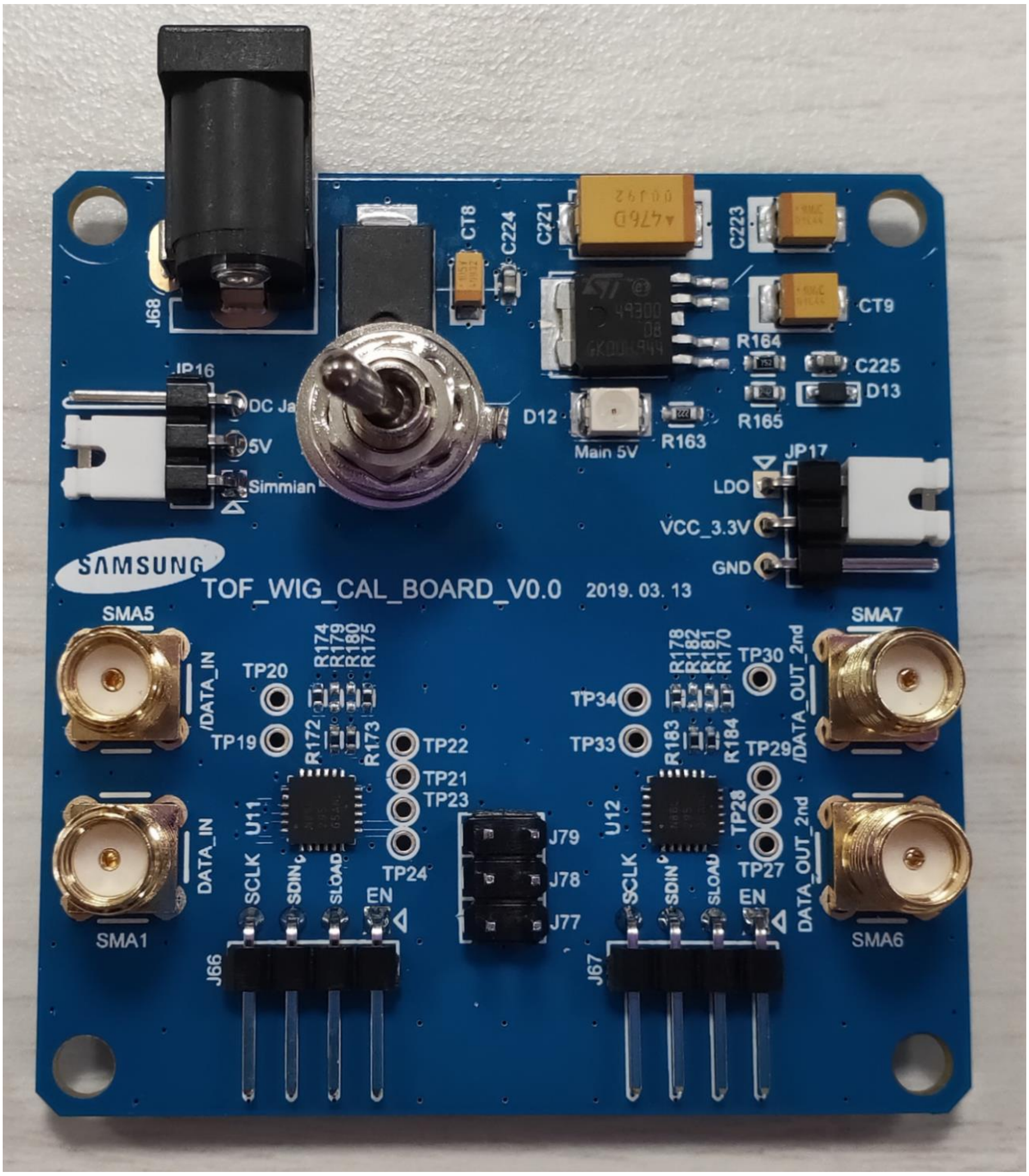

Figure 2.12 Manufactured Wiggling Calibration Board

Figure 2.13 shows a photo of the Wiggling Calibration Board connected with the Arduino board. The Arduino board was used for time delay control as one example, and you can control it with any types of MCU boards. The datasheet of “**NB6L295**” provides a detailed description of 3-wire interface.

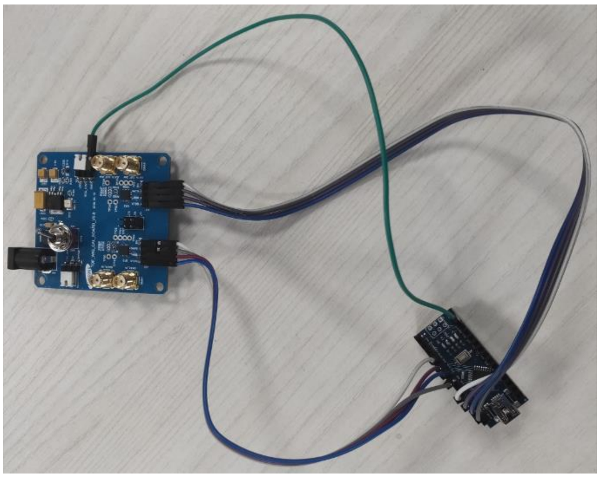

Figure 2.13 Wiggling Calibration Board connected to the Arduino Board

<u>Board Evaluation Environment</u>

The datasheet of the time-delayc IC describes a table for the delay time for each control bit. However, specified time delay may vary with environment change such as temperature, supply voltage and so on. Hence, we recommend to evaluate the Wiggling Calibration Board and make a table for delay time with each control bit.

Figure 2.14 shows the evaluation environment of the manufactured Wiggling Calibration Board. For high-accuracy evaluation, we used a function generator from **Agilent Technology 8113DA** for Input signal generation and an oscilloscope from **Keysight DSA-Z 504A** for measuring output signal delay. The function generator generates a pair of differential clock signal in 1-V DC offset and 600-mV amplitude at 100 MHz. A power splitter has been used to split the differential clock signals into the Wiggling Calibraton Board and the oscilloscope.

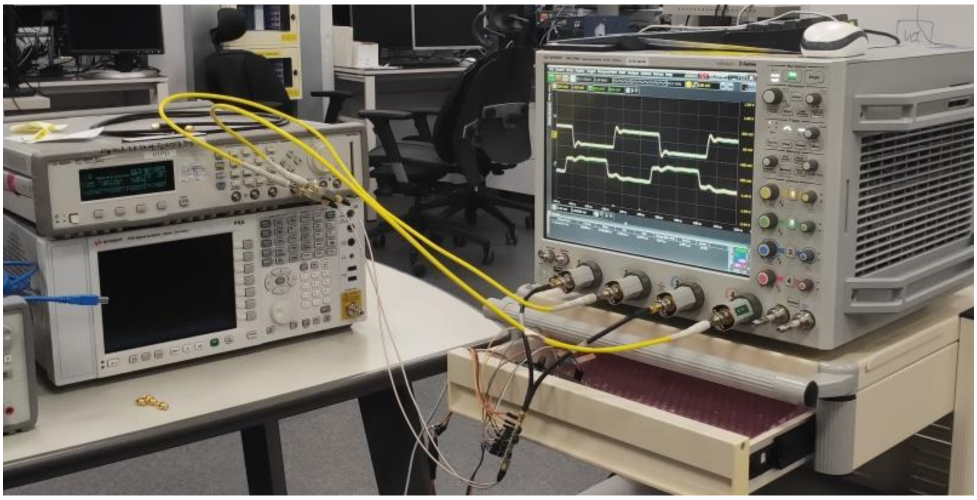

Figure 2.14 Evaluation Environment of Wiggling Calibration Board

One of the example of the time delay with the control code is shown in Table 2.2. The embedded delay means offset delay from the board environment such as cable, connector, PCB trace and “zero-code” delay in the time-delay IC; this delay will be de-embedded during the wiggling calibration process. Control code can control total 2048 steps (1024 steps per IC). The control code of each IC is 9 bit, but the IC has two rails of time delay which can be independently controlled by a flag signal for 1024-step extension (see more details from the IC’s datasheet [2]). This table is an example of evaluation of a Wiggling Calibration Board. Since there may be a deviation of time delay on the Wiggling Calibration Board, each board will need to be evaluated. For precise wiggling calibration, we recommend creating tables on each board.

Table 2.2 Measured delay value of the board according to input-code (Example)

	Control code																																				Delay (Meas.)
	2nd Time Delay IC																		1st Time Delay IC																		ps
	PD1									PD0									PD1									PD0									
	D8	D7	D6	D5	D4	D3	D2	D1	D0	D8	D7	D6	D5	D4	D3	D2	D1	D0	D8	D7	D6	D5	D4	D3	D2	D1	D0	D8	D7	D6	D5	D4	D3	D2	D1	D0	
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Embedded delay
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	+13
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	+19
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	+41.33
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	+85.67
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	+170.67
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	+340.33
8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	+672.67
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	+1349
10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	+2697.33
11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	+5373
12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	+5386
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	1	1	1	1	1	1	1	+5390.67
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1	1	1	1	1	1	1	1	+5411.67
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	1	1	1	1	1	1	1	1	+5453.67
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	1	1	1	1	1	1	1	1	+5537.33
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	1	1	1	1	1	1	1	1	+5709.33
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	+6046.67
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	+6725.33

20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	+8081.67
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	+10772
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	+10781.33
23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	+10791.67
24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	+10811
25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	+10852.67
26	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	+10937.67
27	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	+11106.33
28	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	+11443.67
29	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	+12115
30	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	+13450.67
31	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	+16110
32	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	+16120
33	0	0	0	0	0	0	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	+16125.33
34	0	0	0	0	0	0	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	+16148.67
35	0	0	0	0	0	1	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	+16188.67
36	0	0	0	0	1	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	+16277.67
37	0	0	0	1	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	+16446
38	0	0	1	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	+16783
39	0	1	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	+17456.67
40	1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	+18801
41	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	+21470

Precautions when using the Wiggling Calibration Board

The time-delay IC on the Wiggling Calibration Board has a delay-variation depending on the temperature. Time delay drift may occur over time, when the internal temperature of the ICs is stabilized. Therefore, sufficient warm-up time may be required when using the Wiggling Calibration Board. We recommend having a warm-up time of at least 15 minutes for stable results. This warm-up time varies with different board design, and is required only at the beginning of the equipment initialization; after the warm-up time, no additional warm-up is required for wiggling calibration process of each individual module.

2.3 **Lens-FPPN**

A geometric calibration uses the standard technique from Zhang [3]. The parameters are usually estimated through homographies between 3D and 2D points. By using at least three different views of a planar checkerboard, the lateral intrinsic camera parameters (principal point, $\mathbf{cc}$, focal length, $\boldsymbol{f}_{\mathbf{c}}$ and lens distortions, $\mathbf{kc}$) are estimated. Lens-FPPN calibration uses at least four different views of a plane due to FPPN.

$\mathbf{K}$ is known as the camera matrix, and defined as follows:

$$\mathrm{K} = \begin{bmatrix} \boldsymbol{f}_{\mathbf{c}}(1) & 0 & \mathbf{cc}(1) \\ 0 & \boldsymbol{f}_{\mathbf{c}}(2) & \mathbf{cc}(2) \\ 0 & 0 & 1 \end{bmatrix}$$

Lens distortion uses the Brown's model as follows:

$$\mathbf{x}_{\mathbf{d}} = \begin{bmatrix} \mathbf{x}_{\mathbf{d}}(1) \\ \mathbf{x}_{\mathbf{d}}(2) \end{bmatrix} = (1 + \mathbf{kc}(1)r^2 + \mathbf{kc}(2)r^4 + \mathbf{kc}(5)r^6)\mathbf{x}_{\mathrm{n}} + \mathbf{dx}$$

where $\mathbf{dx}$ is the tangential distortion vector:

$$\mathbf{x}_{\mathbf{d}} = \begin{bmatrix} 2\mathbf{kc}(1)xy + \mathbf{kc}(4)(r^2 + 2x^2) \\ \mathbf{kc}(3)(r^2 + 2y^2) + 2\mathbf{kc}(4)xy \end{bmatrix}$$

Therefore, the 5-vector $\mathbf{kc}$ contains both radial and tangential distortion coefficients.

Fixed pattern phase noise (FPPN) accounts for pixel-location-dependent phase deviations. The conventional approach is to estimate the FPPN with the aid of a plane and approximate the error with a polynomial fitting. Lens and FPPN calibration can be performed with the same equipment and environment simultaneously because extrinsic parameters of different views of a plane can be obtained in the lens calibration and ground truth (GT) of a plane can be calculated using extrinsic and intrinsic parameters.

2.3.1 **Procedure**

Firstly lens calibration is performed. Therefore intrinsic and extrinsic parameters can be obtained. Using extrinsic paramters, FPPN calibration can generate spatial depth offset LUT.

Lens-FPPN Calibration Procedure (as shown in Figure 2.15)

1) Select the modules for Lens-FPPN calibration.

2) Preheating for 5 seconds before calibrating the reference module.

3) Set angles of checker chart as 0°, 90°, 180° and 270°. Additional 10°, 100°, 190° and 280° can be set.

4) Obtain raw data images in each angle for dual modulation frequency.

5) Lens calibration is performed usinig Calibration Library.

a) Calculate intrinsic paramters for each module.

b) Calculate extrinsic parameters for each raw data images.

6) FPPN calibration is performed usinig Calibration Library.

a) Calculate spatical depth offset.

b) Applies polynomial surface fitting and extracts coefficients

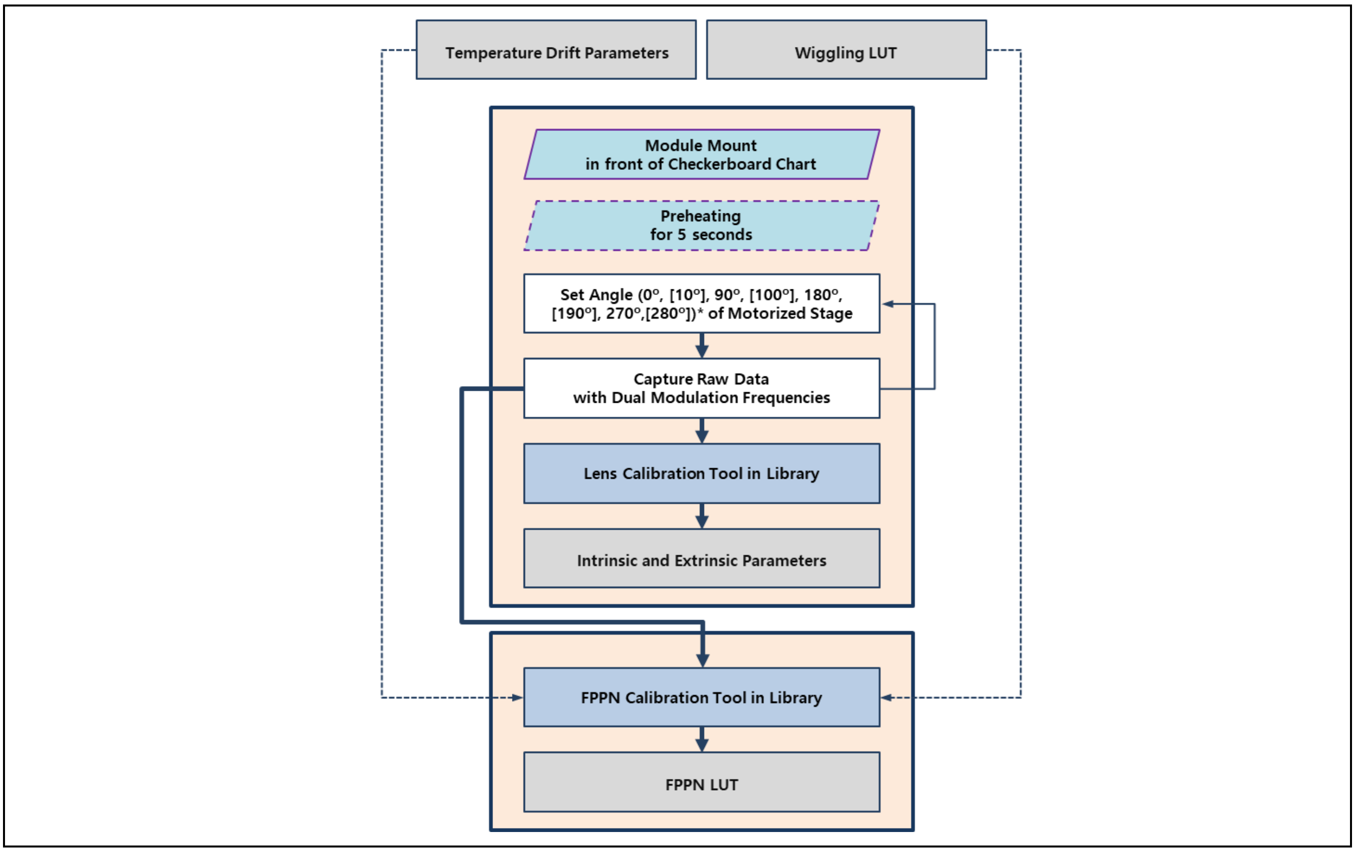

Figure 2.15 Lens-FPPN Calibration Procedure

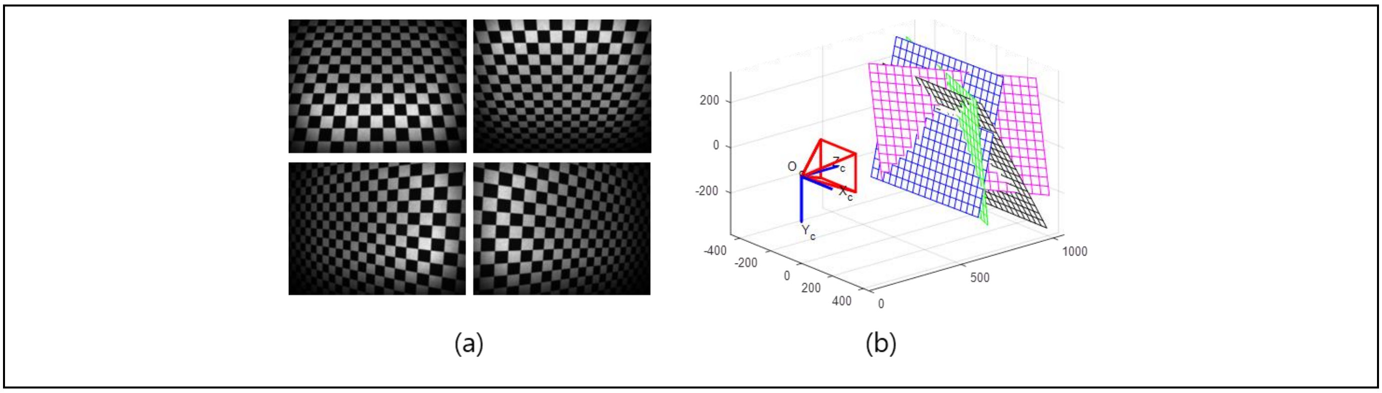

Figure 2.16 Lens Calibration Procedure.

Lens-FPPN Calibration Procedure in detail

FPPN calibration is to compensate pixel-location-dependent error due to the time-delay of the demodulation signal depending on pixel location and the misalignment of VCSEL and sensor (which is a geometric error). Each pixel of a ToF sensor can have different depth offset due to time-delay with respect to the injection of the reference control signal. Also, VCSEL and sensor can be misaligned due to limited space to be located in for small form factor. This misalignement can cause per-pixel depth offset.

As FPPN input data, same checker board images are used with different rotation angles which consists of basically 0°, 90°, 180° and 270°. In order to improve the performance, additional different rotataion angles can be added such as 10°, 100°, 190° and 280°. Figure 2.17 show FPPN calibration procedure in detail while using 4 different rotation angle inputs. Ground truth depth image of each input checker board image can be generated with extrinsic paramters obtained from lens calibration. Input raw images are converted as measured depth. Low confidence depth data can be removed by using confidence map. Depth offsets are calculated in higher confidence area. Next, remained depth offset images are merged into one offset image.

Since areas of the low-confidence depth pixels are excluded, there are no depth offset data in some spatial positions as shown in Figure 2.18. In order to fill lack of depth offsets, special filtering is applied and then whole depth offsets can be obtained. FPPN depth offset can be estimated by polynomial surface fitting technique. Finally, calibration library calculates the FPPN depth offset LUT.

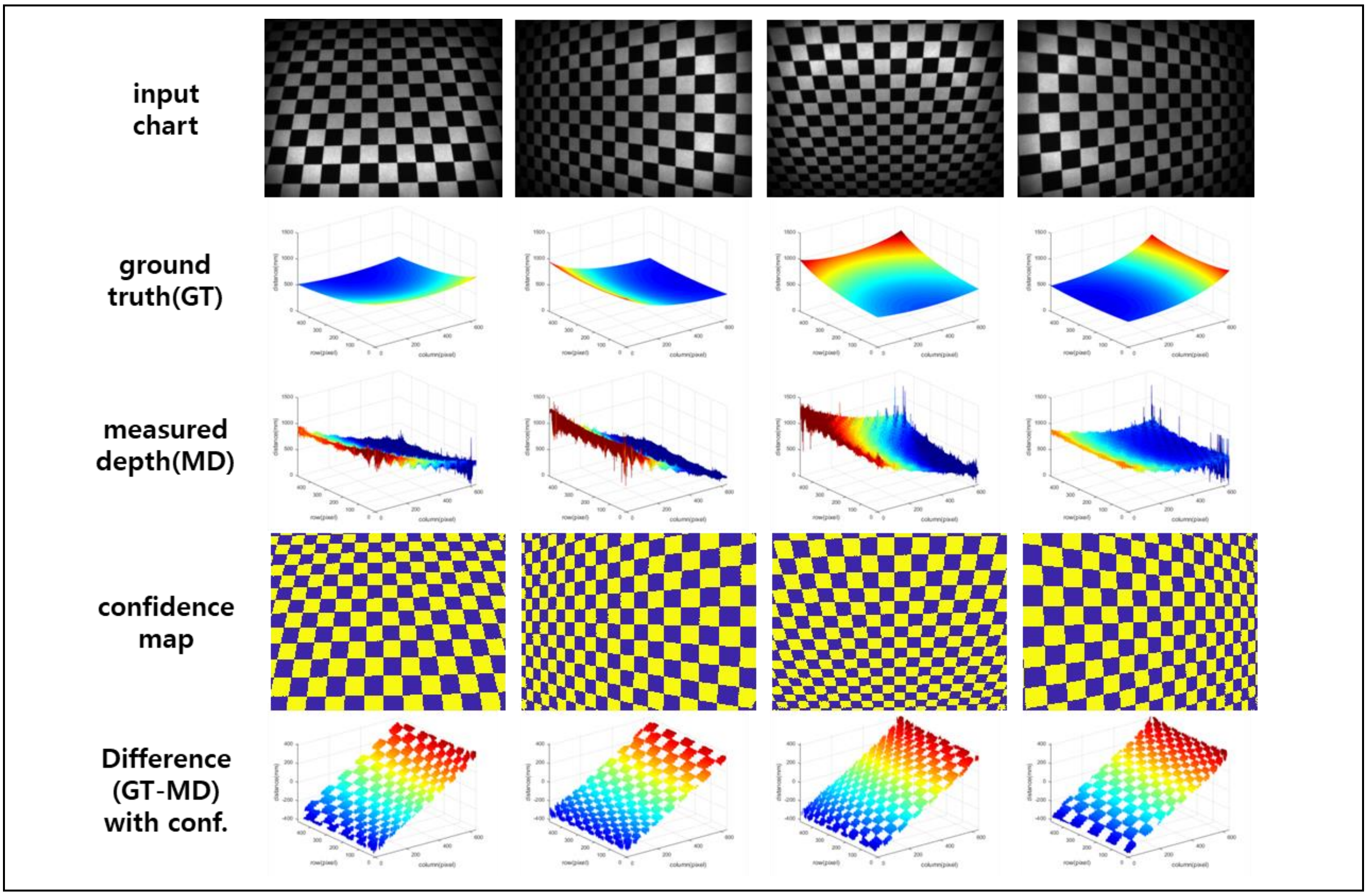

Figure 2.17 FPPN Calibration Procedure; each rotation.

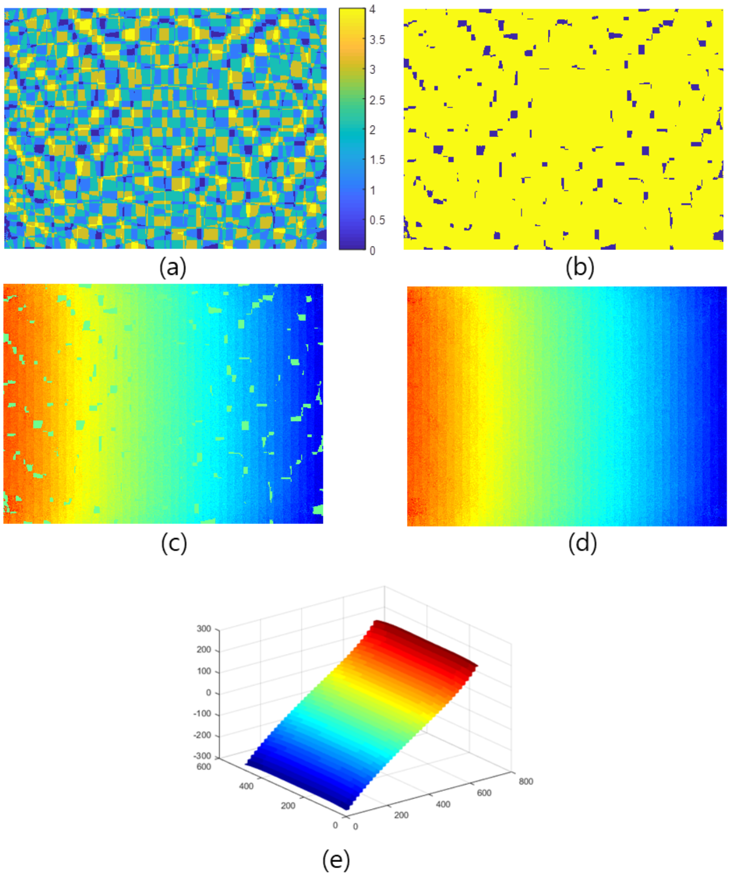

Figure 2.18 FPPN Calibration Procedure; merge all rotations.

2.3.2 **Equipments**

Figure 2.19 and Table 2.3 shows the example of lens-FPPN chart. The specification of lens-FPPN chart can be changed depending on the various conditions. The distance from a module and a plane chart is fixed by monitoring the code to prevent saturation. This distance is about 375 – 825 mm, depending on the optical power of VCSEL, integration time, and chart reflectivity.

When calibrating lens-FPPN, “Lens-FPPN & Temperature Drift Calibration Board” without the external time-delay as shown in Figure 2.3 has to be used.

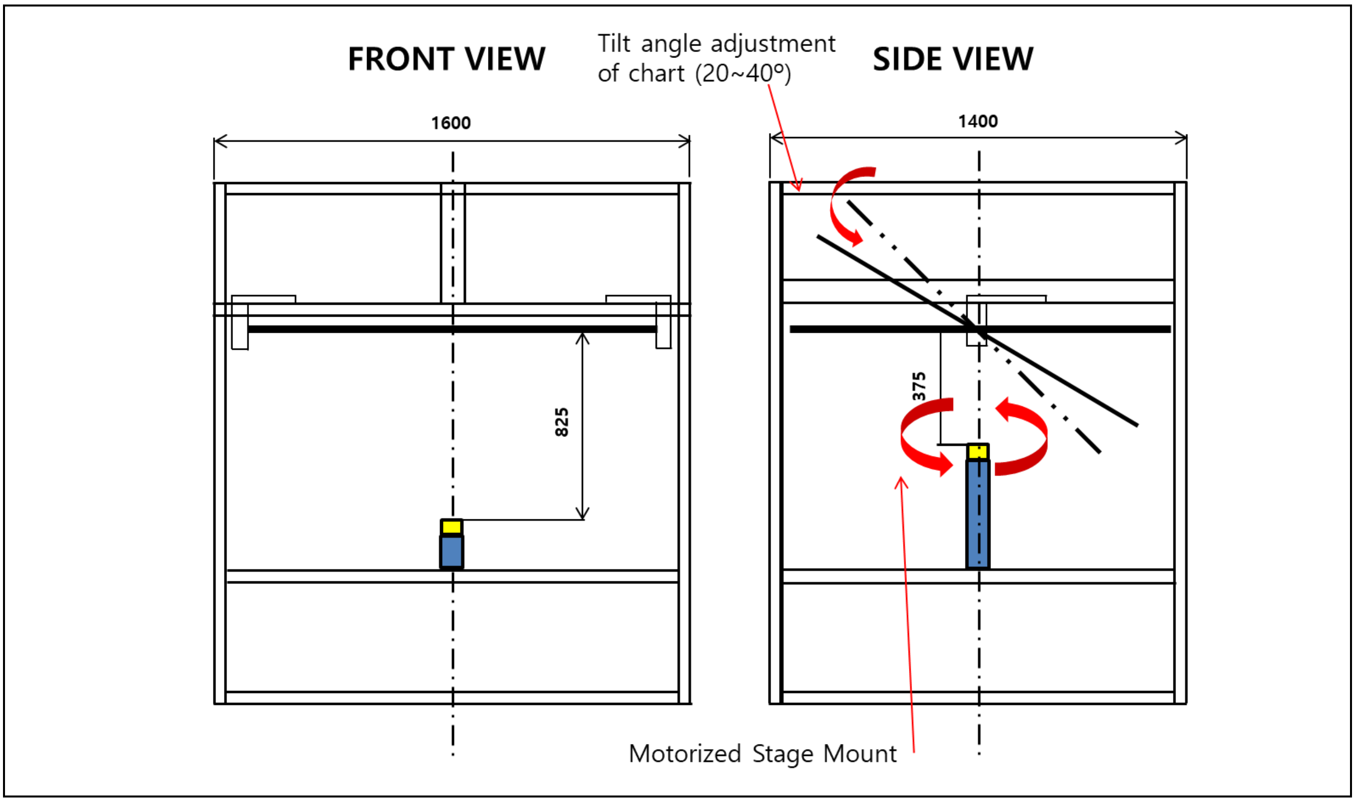

Figure 2.19 Lens-FPPN Calibration Chart. Height adjustable stage mount for module (375 – 825 mm) and an angle adjustable chart (20º – 40º). Height and angle will be fixed for a manufacturing environment.

Table 2.3 The Specification of Lens-FPPN Calibration Chart.

Chart type	Checkerboard
Chart size	1610 mm × 1410 mm
Material (lambertian reflectivity)	White > 80%, Dark > 10%
Square size	40 mm × 40 mm
Square count	40 x 35
Tilt angle adjustment	30º
Angle of Motorized Stage	0º, 90º, 180º, 270º + 10º, 100º, 190º, 280º (added angles depending on performance)
Distance from module	375 – 825 mm

3 Software

3.1 Overview

Software section is to explain usage of software calibration tool and library provided by Samsung Electronics. ToF calibration tools consist of monitoring tool, calibration library and validatation tool, as shown in Figure 3.1. In this section only temperature drift monitoring GUI (graphic user interface) tool for one-time calibration and calibration library for module to module calibration are covered except validataion tool.

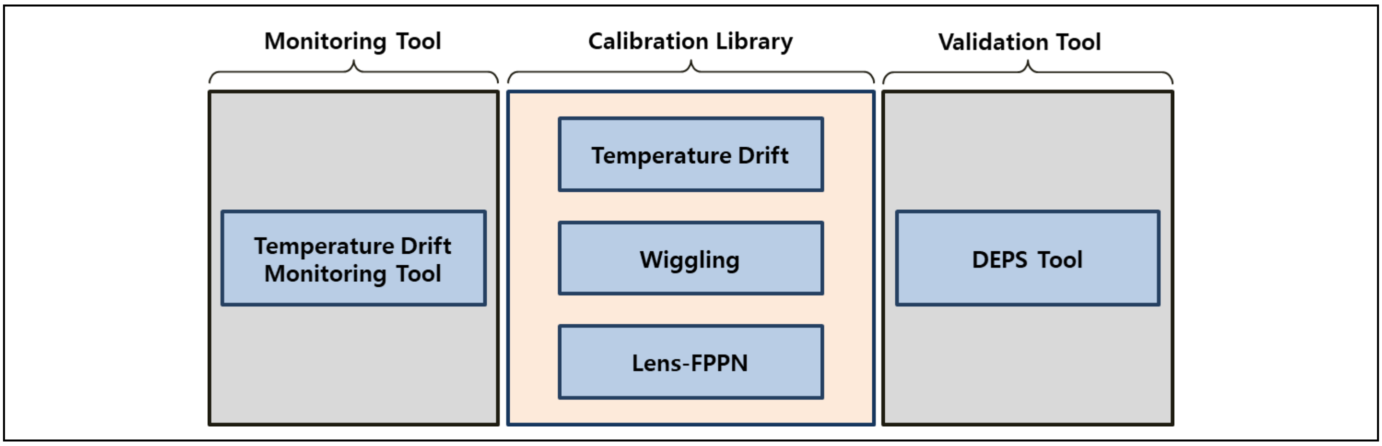

Figure 3.1 Released Softwares; 1) Temperature Drift Monitoring Tool, 2) Calibration Library, 3) Validation Tool.

3.2 Temperature Drift Monitoring Tool

As mentioned in Section 2.1, temperature drift leads to a change in measured distance. In the calibration procedure, the systematic errors could be calibrated with the temperature drift parameters. For these parameters, Temperature Drift Monitor tool measures the calibration data at measurement of fixed distance.

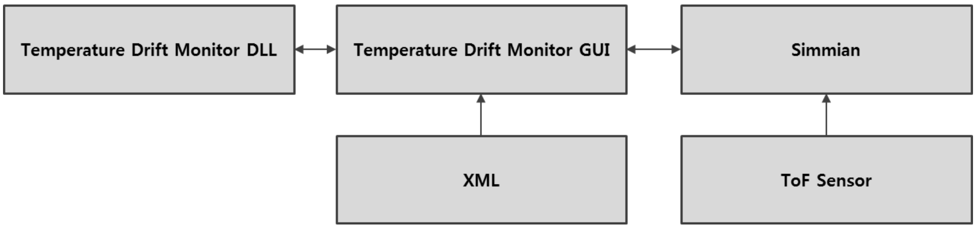

Figure 3.2 Temperature Drift Calibration Block Diagram

The Temperature Drift Monitor tool is composed of data-processing, graphic-user-interface (GUI) and data-capture blocks. The data-processing block is a dynamic link library (DLL) type. Therefore, users can design their own user interface. The usage guide of Temperature Drift Monitor DLL will be explained in Section 3.2.1. The Temperature Drift Monitor GUI is shown in Fig. 3.3 and controlled by configuration parameters in xml file. The Simmian software (NxSimmian and Simmian) and device drivers must be installed for operation of Temperature Drift Monitor GUI.

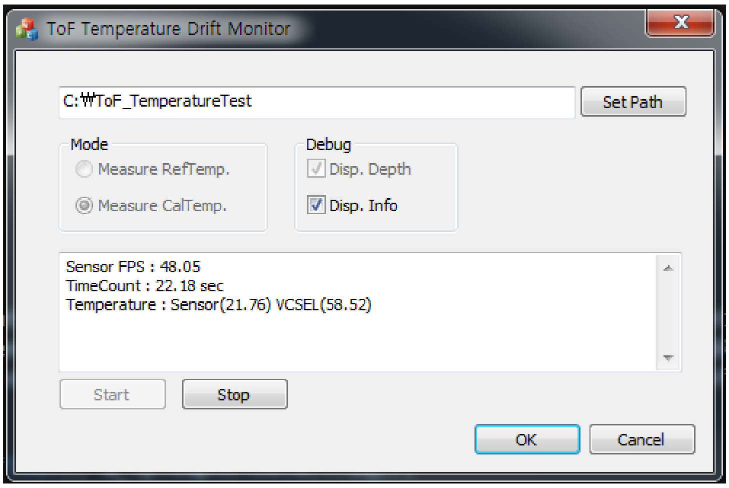

Figure 3.3 Temperature Drift Monitor GUI

3.2.1 **Usage**

The Temperature Drift Monitor options are shown in Fig. 3.4.

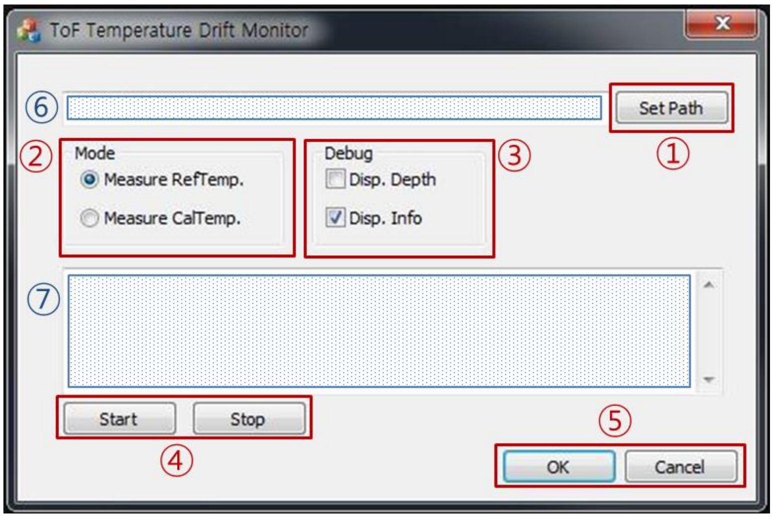

Figure 3.4 Temperature Drift Monitor Options

1. Set Path: Selecting a working folder for measuring temperature calibration data. The selected path is shown in the path window (Edit Control Num.6). The folder should include a sub-folder of folder named ‘Data’. The ‘Data’ folder is distributions from Samsung Electronics.

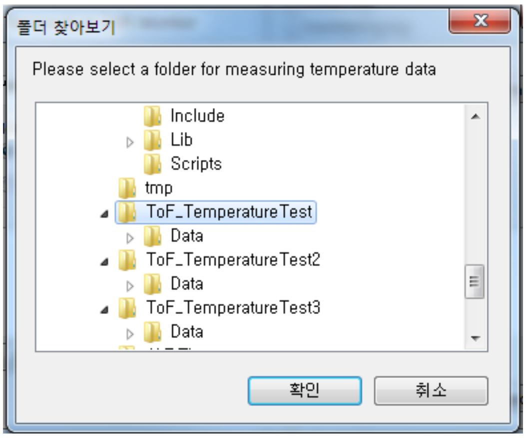

Figure 3.5 Set Path Dialog

2. Mode: Selecting temperature drift monitoring mode. 'Measure RefTemp.' is the mode to measure a reference temperature drift statistics after some warm-up time, t_w. During the warm-up time, distance measurement may not be correct (see Figure 3.6). 'Measure CalTemp.' is the mode to measure a calibrated temperature drift statistics based on reference temperature drift statistics for varying temperature.

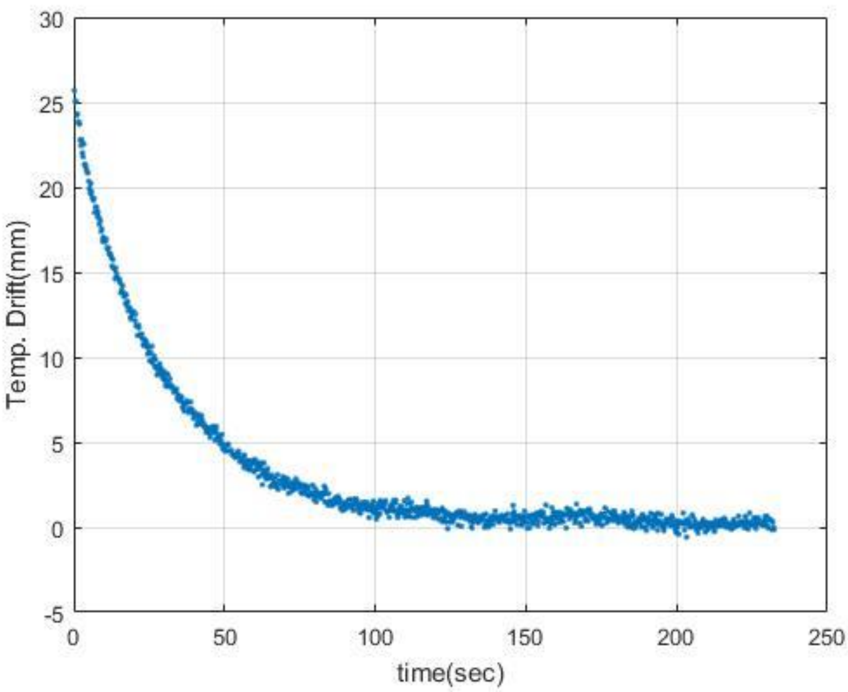

Figure 3.6 Temperature drift offset from reference depth caused by self-induced heating

3. Debug: The checkbox 'Disp. Depth' controls a display window of depth map as shown in Figure 3.7. This checkbox is disable during mode operation. 'Disp. Info' turn on and off about information window (Edit Control Num.7), as shown in Figure 3.8.

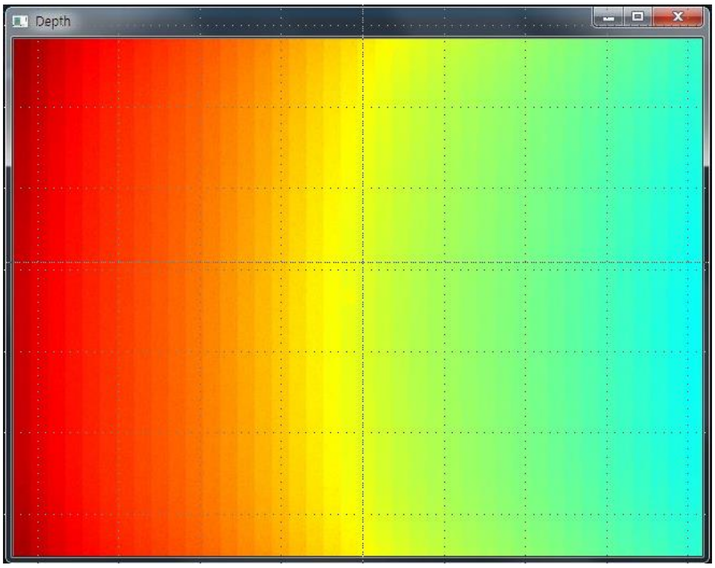

Figure 3.7 Display Depth Map

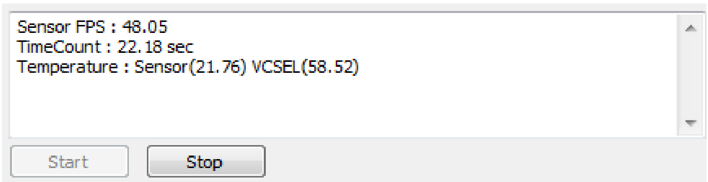

Figure 3.8 Display Information

4. Start and Stop: Start and stop the selected mode operation. After starting a mode, the start button is disable. The disabled start button is enabled after finishing the mode operation or click the stop button.

5. OK and Cancel: Close Temperature Drift Monitor tool

3.2.2 **Deliverables**

Below items are deliverables to measure temperature calibration data.

- ToFTemperatureDriftMonitor.exe : execution program
- ToFTemperatureMonitor_DLL.dll : temperature drift monitor library
- ToFTempMonitor.xml : configuration file in XML
- Data folder included RegisterMap, Setfile, Wiggling and FPPN calibration data : data files
- opencv_world341.dll : OpenCV library

3.2.3 **Temperature Drift Monitor Configuration file**

Temperature Drift Monitor is controlled by configuration parameters in XML files (see Figure 3.9).

```
<TempMonitorXML>
  <!-- Path Config -->
  <RawFilePath>\Data\</RawFilePath>
  <SetFilePath>\Data\setfile\</SetFilePath>
  <BinFilePath>\Data\bin\</BinFilePath>
  <WiggCalFilePath>\Data\wigg\</WiggCalFilePath>
  <FppnCalFilePath>\Data\fppn\</FppnCalFilePath>
  <!-- FileName Config -->
  <RegisterMapFileName>RegsMap.bin</RegisterMapFileName>
  <SetFileName>setfile.tset</SetFileName>
  <RefBinFileName>refTempData.bin</RefBinFileName>
  <RefFinalBinFileName>refTempFinalData.bin</RefFinalBinFileName>
  <WiggCalFileName>wigg_f1_tbl.bin</WiggCalFileName>
  <FppnCalFileName>fppn_f1_tbl.bin</FppnCalFileName>
  <!-- Control Config -->
  <ReferenceCaptureTimeSec>720</ReferenceCaptureTimeSec>
  <ReferenceCaptureFrameNum>10</ReferenceCaptureFrameNum>
 </TempMonitorXML>
```

Figure 3.9 Temperature Drift Monitor Configuration XML example

- RawFilePath : folder where captured data will be saved
- SetFilePath : folder of setfiles
- BinFilePath : folder where temperature drift data will be saved
- WiggCalFilePath : folder of wiggling calibration data
- FppnCalFilePath : folder of fppn calibration data
- RegisterMapFileName : register map filename
- SetFileName : setfile filename
- CalBinFileName : calibrated temperature drift statistics data filename
- RefBinFileName : temperature drift statistics data during warm-up time
- RefFinalBinFileName : temperature drift statistics data after warm-up time
- WiggCalFileName : wiggling calibration filename
- FppnCalFileName : fppn calibration filename
- ReferenceCaptureTimeSec : warm-up time (sec)
- ReferenceCaptureFrameNum : frame numbers to average

3.2.4 **Temperature Drift Monitor DLL API**

Temperature Drift Monitor DLL API is class structure supported by Samsung Electronics for processing temperature data. This class includes 10 functions and 7 variables.

```
class TOFTEMPERATUREMONITOR_DLL_API ToF_TemperatureMonitor

{

public:

ToF_TemperatureMonitor();

~ToF_TemperatureMonitor();

// Data
float m_elapsedTime;
float m_sensorTemp;
float m_vcselTemp;

float m_refCalData[2];
float m_refSensorTemp[2];
float m_refVcselTemp[2];

// Function
void SetRawSize(int row, int col);
void ResetTime();
float ElapsedTime();

bool SetSaveData(CString path, int calmode);
bool LoadRefData(CString path);
void SetRefCount(int refCount);


bool ReadRaw(CString path, int calmode, bool refmode);

bool SetWiggCalData(CString path);
bool SetFppnCalData(CString path);

// Debug
bool m_bDispDepth;
void CtrlDispDepth(bool dispDepth);



}
```

Variables

- `m_elapsedTime` : elapsed time after time initialization
- `m_sensorTemp` : sensor temperature value
- `m_vcselTemp` : VCSEL temperature value
- `m_refCalData` : distance mean value after warm-up time (Freq1 and Freq2)

- `m_refSensorTemp` : sensor temperature mean value after warm-up time (Freq1 and Freq2)
- `m_refVcselTemp` : VCSEL temperature mean value after warm-up time (Freq1 and Freq2)
- `m_bDispDepth` : control variable for display depth map window

Function

- `void SetRawSize(int row, int col)`

  set raw image size

- `void ResetTime()`

  time initialization

- `float ElapsedTime()`

  return elapsed time after time initialization

- `bool SetSaveData(CString path, int calmode)`

  set binary file path for each mode (‘Measure RefTemp.’ and ‘Measure CalTemp.’)

  calmode – ‘Measure RefTemp.’ mode should set 0 and 2. ‘Measure CalTemp.’ mode need to set 1

- `bool LoadRefData(CString path)`

  set binary file path of temperature drift statistics (‘Measure RefTemp.’ outputs) for ‘Measure CalTemp.’ mode

- `void SetRefCount(int refCount)`

  set frame numbers for averaging reference temperature data

- `bool ReadRaw(CString path, int calmode, bool refmode)`

  read raw data file based on raw data file path, mode parameter

  path – raw file path to read (raw and embedded file)
  calmode – 0 is ‘Measure RefTemp.’ and 1 is ‘Measure CalTemp.’
  refmode – true after warm-up time and false before warm-up time

- `bool SetWiggCalData(CString path)`

  `set wiggling calibration data path`

- `bool SetFppnCalData(CString path)`

  `set fppn calibration data path`

- `void CtrlDispDepth(bool dispDepth)`

  control display depth map window

This is example code for Temperature Drift Monitor API.

```
#include <afxstr.h>
#include "ToFTemperatureMonitor_API.h"

int main()
{
 ToF_TemperatureMonitor TempMonitor;
 int TemperatureMonitorMode = 0; // mode
 bool bRunnig = true;

 int row = 480; // raw width
 int col = 640; // raw height
 int nRefCount = 10; // averaging 10 frames

 CString strRawFilePath = _T("c:\\tof_temperature\\Data\\");
 CString strRefBinFilePath = _T("c:\\tof_temperature\\Data\\bin\\refTempData.bin");
 CString strRefFinalBinFilePath =
 _T("c:\\tof_temperature\\Data\\bin\\refTempFinalData.bin");
 CString strCalBinFilePath = _T("c:\\tof_temperature\\Data\\bin\\calTempData.bin");

 CString strWiggFilePath = _T("c:\\tof_temperature\\Data\\wigg\\wigg_f1_tbl.bin");
 CString strFppnFilePath = _T("c:\\tof_temperature\\Data\\fppn\\fppn_f1_tbl.bin");


 if (TemperatureMonitorMode)
 {
     /* measure reference temperature drift statistics */
     if (!TempMonitor.SetSaveData(strRefBinFilePath, 0)
        || !TempMonitor.SetSaveData(strRefFinalBinFilePath, 2))
       return -1;

     TempMonitor.SetRefCount(nRefCount);
  }
  else
  {
     /* measure calibrated temperature drift statistics */
     CString strFreq1BinFilePath, strFreq2BinFilePath;

     int r_idx = strRefBinFilePath.ReverseFind(_T('.'));
     strFreq1BinFilePath = strRefBinFilePath.Left(r_idx) + _T("_f1.bin");
     strFreq2BinFilePath = strRefBinFilePath.Left(r_idx) + _T("_f2.bin");



     if ((GetFileAttributes(strFreq1BinFilePath) == 0xFFFFFFFF)

     && (GetFileAttributes(strFreq2BinFilePath) == 0xFFFFFFFF))
```

```
    {

       /* error */
     printf("No RefTemp. Data");
     return -1;

    }

    else
     TempMonitor.LoadRefData(strRefFinalBinFilePath);

    if (!TempMonitor.SetSaveData(strCalBinFilePath, 1))
      return -1;
}

/* file setting */
TempMonitor.SetRawSize(row, col);

/* init cal_data*/
if (!TempMonitor.SetWiggCalData(strWiggFilePath))
{

    printf("Failed to load the WiggCal Data");
    return -1;
}


if (!TempMonitor.SetFppnCalData(strFppnFilePath))
{
    printf("Failed to load the FPPNCal Data");
    return -1;
}

/* time reset */
TempMonitor.ResetTime();

bool bRef = false;

while (bRunnig)
{

      float elapsed_time = TempMonitor.ElapsedTime();

     /* user capture code */
     // TODO : place code here to capture raw file


     if (elapsed_time > TemperatureMonitorMode == 0)
         bRef = true;

     /* read raw file */
     TempMonitor.ReadRaw(strRawFilePath, TemperatureMonitorMode, bRef);

     if (!bRunnig) break;
```

```
    }

    return 0;
}
```

3.2.5 Temperature Drift Monitor Output

Temperature Drift Monitor tool generates outputs as binary format of temperature drift statistics. The file name can be changed by user.

- `calTempData.bin`

  temperature drift data
  (float elapsed_time, float sensor_temperature, float vcsel_temperature, float depth)

- `refTempData.bin`

  temperature drift data druing warm-up time
  (float elapsed_time, float sensor_temperature, float vcsel_temperature, float depth)

- `refTempFinalData.bin`

  temperature drift data after warm-up time (mean value of data the number of `SetRefCount set value`)
  (float sensor_temperature_mean, float vcsel_temperature_mean, float depth_mean)

3.3 Calibration Library

Calibration library is common dynamic library (*.dll) developed in C++. Since calibration library internally uses OpenCV library's functions and data structures, OpenCV dynamic library (currently 3.4.1 version) is included in deliverables. Also, two sets of calibrarion library will be provided according to operating systems x64 and x86.

3.3.1 Deliverables

Below items are deliverables to calibrate ToF sensor.

(1) Opencv_world341.dll : OpenCV library

(2) ToFCalParam.xml : calibration opearation configuration file in XML

(3) ToF_Calibration_DLL_x64.dll(.lib)/ ToF_Calibration_DLL_x64.lib(.lib) : calibration dynamic libraries for x64

(4) ToF_Calibration_DLL_Win32.dll(.lib) / ToF_Calibration_DLL_Win32.lib(.lib): calibration dynamic libraries for x86

(5) ToF_Calibration_EXE_x64/ToF_Calibration_EXE_Win32 : execution example program

(6) ToF_Calibration_API.h : API definition C++ header file

(7) ToF_Calibration_EXE.cpp : Execution program example inC++ source file

3.3.2 **API**

Only one main API is needed to execute calibration for module to module calibration by calling in execution program.

```
TOF_CALBRATION_API int Calibrate_ToF()
```

Execute ToF calibrations. Returns success (0) or fail(-1)

Parameters :

- `TOF_CALBRATION_API :` defines as `__declspec(dllimport)` for dynamic libraray usage

Calibration execution example is given below. Firstly function `Calibrate_ToF()` is called. And then calibration execution result can be monitored by checking return value.

```
#include <iostream>
#include "ToF_Calibration_API.h"

using namespace std;

int main()
{
	int ret = 0;
	ret = Calibrate_ToF();
	if(ret == 0)
		cout<<endl<< ">>>>>>> Tof Calibration is done <<<<<<<" <<endl;
	else
		cout<<endl<< ">>>>>>> Tof Calibration is fail <<<<<<<" <<endl;

	return ret;
}
```

3.3.3 **Calibration Configuration file**

In order to control and set the proper calibration environmnent, ToFCalParam.xml is provided. ToFCalParam.xml includes input/output interfaces and specific parmaters in each calibration module.

```xml
<!-- Common Calibration Parameters -->
<COMM_RawFilePath>"..\..\DB\"</COMM_RawFilePath>
<COMM_CalFilePath>Calibration\</COMM_CalFilePath>
<COMM_Operation_Mode>5</COMM_Operation_Mode>
<COMM_ImageWidth>640</COMM_ImageWidth>
<COMM_ImageHeight>480</COMM_ImageHeight>
<COMM_InfoWidth>3840</COMM_InfoWidth>
<COMM_InfoHeight>1</COMM_InfoHeight>
<COMM_ModulationFrequency1>100</COMM_ModulationFrequency1>
<COMM_ModulationFrequency2>80</COMM_ModulationFrequency2>
<COMM_ModeDualFreqOn>1</COMM_ModeDualFreqOn>
<COMM_EnableTemp>1</COMM_EnableTemp>
<COMM_EnableWigg>1</COMM_EnableWigg>
<COMM_EnableFPPN>1</COMM_EnableFPPN>
<COMM_EnableLens>1</COMM_EnableLens>
<COMM_EnableBinG>1</COMM_EnableBinG>
<!-- Temperature Drift Calibration Parameters -->
<TEMP_RawFilePath>temperature\</TEMP_RawFilePath>
<!-- Wiggling Calibration Parameters -->
<WIGG_RawFilePath>wiggling\</WIGG_RawFilePath>
<WIGG_DisStr>0</WIGG_DisStr>
<WIGG_DisEnd>1392</WIGG_DisEnd>
<WIGG_DisStep>64</WIGG_DisStep>
<WIGG_DisCap>460</WIGG_DisCap>
<!-- FPPN Calibration Parameters -->
<FPPN_RawFilePath>lens_fppn\</FPPN_RawFilePath>
<!-- Lens Calibration Parameters -->
<LENS_RawFilePath>lens_fppn\</LENS_RawFilePath>
<LENS_Num>8</LENS_Num>
<LENS_NumInc>1</LENS_NumInc></CalParameter>
```

Figure 3.10 Calibration Configuration XML example

3.3.3.1

3.3.3.2 Common Calibration Parameters

Common paramters for all calibration modes can be set.

COMM_Tab_Mode: tab mode selection

- 2 : 2 tab, 3: 2 tab shuffle, 4 : 4 tab, 5 : 4 tab shuffle
- Default : 5 tab shuffle

COMM_ImageWidth: final depth output image width

- Default : 640

COMM_ImageHeight: final depth output image height

- Default : 480

COMM_InfoWidth: information (embedded) file/data width

- Default : 3840

COMM_InfoHeight: information (embedded) file/data height

- Default : 1

COMM_ModulationFrequency1: first modulation frequency value in MHz

- Default : 100

COMM_ModulationFrequency2: second modulation frequency value in MHz

- Default : 80

COMM_DualFreqModeOn: dual modulation frequency mode on/off

- 0 : off, 1 : on
- Default : 1-on

COMM_RawFilePath: common calibration input file location

- Default: "..\..\DB\"

COMM_CalFilePath: calibration result output file location. Internally this location is added to COMM_RawFilePath

- Default: "Calibration\" (actual path is made as "..\..\DB\Calibration\")

COMM_EnableTemp: temperature calibration on/off

- 0 : off, 1 : on
- Default : 1-on

COMM_EnableWigg: wiggling calibration on/off

- 0 : off, 1 : on
- Default : 1-on

COMM_EnableFPPN: FPPN calibration on/off

- 0 : off, 1 : on
- Default : 1-on

COMM_EnableLens: lens calibration on/off

- 0 : off, 1 : on
- Default : 1-on

COMM_EnableBinG: Calibration result generation in binary format on/off

- 0 : off, 1 : on
- Default : 1-on

3.3.3.3 Temperature Drift Calibration Paramters

Temperature calibration paramters can be set.

TEMP_RawFilePath: temperature calibration input file location. Internally this location is added to COMM_RawFilePath.

- Default: “temperature\” (actual path is set as “..\..\DB\temperature\ “)

3.3.3.4 Wiggling Calibration Paramters

Wiggling calibration parameters can be set.

WIGG_RawFilePath: wiggling calibration input file location. Internally this location is added to COMM_RawFilePath.

- Default: “wiggling\” (actual path is set as “..\..\DB\wiggling\”)

WIGG_DisStr: starting control code setting of the time delay circuit

- Default : 0

WIGG_DisEnd: ending control code setting of the time delay circuit

- Default : 1280

WIGG_DisStep: increment of control code

- Default : 64

WIGG_DistCap: initial estimate of the distance between the sensor and the captured chart in mm

- Default : 600

3.3.3.5 FPPN Calibration Parameters

FPPN calibration parameters can be set.

FPPN_RawFilePath: FPPN calibration input file location. Internally this location is added to COMM_RawFilePath.

- Default: “lens_fppn\” (actual path is set as”. .\..\DB\lens_fppn\”)

3.3.3.6 Lens Calibration Paramters

Lens calibration parameters can be set.

LENS_RawFilePath: lens calibration input file location. Internally this location is added to COMM_RawFilePath.

- Default: “lens_fppn\” (actual path is set as “..\..\DB\ lens_fppn\”)

LENS_Num: total calibration input raw image number

- Default : 8

LENS_NumInc: increment of calibration input raw image number

- Default : 1

3.3.4 **Calibration Result Output**

Calibration library generates each calibration block result and merge full calibration results in a binary format. Each calibration block results can be used internally. Also full calibration results can be used to write in a memory such as EEPROM on the camera module.

3.3.4.1 Calibration output in each calibration block

temp_f1_tbl.bin/ temp_f2_tbl.bin: temperature calibration results in binary format for each modulation frequency

wigg_f1_tbl.bin/ wigg_f2_tbl.bin: wiggling calibration results in binary format for each modulation frequency

fppn_f1_tbl.bin/ fppn_f2_tbl.bin: FPPN calibration results in binary format for each modulation frequency

- lens.bin: lens calibration result in binary format

3.3.4.2 Calibration full output

_ToF_cal.bin : merged full calibration result according to memory map.

3.4 **DEPS Tool**

DEPS tool and document will be updated.

References

[1] https://en.wikipedia.org/wiki/Thermoelectric_cooling

[2] https://www.onsemi.com/pub/Collateral/NB6L295-D.PDF

[3] Z. Zhang, "Flexible Camera Calibration by Viewing a Plane from Unknown Orientations," ICCV99, vol. 1, pp. 666-673, 1999.5.

[4] D. Brown, "Decentering distortion of lenses". Photogrammetric Engineering. 32 (3), pp.444-462, 1966.

[5] https://opencv.org/