User's Guide
SLAU791–February 2019

How to set up and calibrate OPT3101 based systems for proximity sensing

This document covers the step-by-step procedure for bringing up a newly built PCB design and running calibration. These steps, especially the ones on calibration, are also relevant when updating the calibration after swapping optical components on an existing board like the OPT3101EVM.

Contents

List of Figures

Trademarks

1 Introduction

1.1 OPT3101 system

OPT3101 based systems can be used in a wide variety of applications like long range distance ranging, wide field of view proximity detection, and collision avoidance in autonomous robots just to name a few. OPT3101 is compatible with a wide variety of emitters and photodiodes that can be arranged in various configurations. OPT3101 based systems require calibration to compensate for various environmental factors. There are some calibrations required per unit and there are some calibrations which need to be done per design. The per design coefficients are common to all units. The coefficients obtained from these calibrations need to be programmed into the registers for the OPT3101. This document helps readers understand the calibration process and the significance of calibration during various stages of hardware bring-up.

1.2 Bring-up calibration process

Bring up and calibration of these systems is an important milestone in understanding the characteristics of the system under various environmental conditions. Measurements taken during calibration are used to program coefficients in the OPT3101 device, which compensate for inaccuracies caused by changing environmental conditions. To understand and estimate these coefficients, it is important to be able to control the environmental factors in a lab setting and characterize the sensitivity of OPT3101 to these factors. This document helps readers understand the various environmental conditions that affect OPT3101 based systems, the capability and the methods available as part of the OPT3101 device, and how to use the OPT3101SDK to compensate these effects. This document assumes users are familiar with the OPT3101 device and have read the OPT3101 data sheet, OPT3101 calibration document, system design guide. The OPT3101SDK and the OPT3101 configurator tool play a crucial role in bring up and calibration. It is highly recommended that users are familiar with these tools available on TI.com

2 First time bring-up

This chapter walks though the step by step process for OPT3101 system bring up. The chapter also covers common problems, tips and resolutions listed which help users quickly bring up the hardware.

2.1 Setup and Tools required

OPT3101 bring-up involves powering-up the OPT3101 system and performing initial checks to ensure the hardware and optics work as expected. It is highly recommended to have the following items readily available to aid the bring up.

- Digital multi meter for probing voltages
- Oscilloscope to debug I2C transaction waveforms and other switching nets
- Different reflectivity targets: Even every day objects like white paper and dark carpet are good
- Masking tape which is opaque to the wavelength used by the system. For example: Some masking tapes may look dark but let though infrared light (around 850 nm) and may not effectively mask IR emitters.
- A camera capable of detecting the wavelength of the emitter used in the system. For example, if the system uses IR emitters then using a near-infrared (NIR) camera will provide the best sensitivity and allow IR light leakage past the masking tape to be detected.

2.2 First checks

Once the hardware is ready for bring up and powered-up, it is recommended to probe several nodes on the PCB before starting communication with the hardware. The following list the nodes recommended for probing and their expected voltages.

- All power supply voltages are as expected
  - If an internal LDO is used, the voltage generated by OPT3101 on the AVDD and the VDD supplies needs to be checked. Both should measure 1.8V
- Pins INP and INM measure 1V
- I2C Slave lines SDA_S and SCL_S both measure the same voltage as IOVDD pulled up though pull-up resistor
- Analog and digital grounds are connected and measure 0 V with respect to each other.

These are some of the common mistakes that occur during assembly which needs to be carefully inspected.

- Emitter and photodiode have been interchanged
  - This is hard to detect especially when using through hole components which look dark.
- One or more emitter or photodiode component's polarity have been reversed
  - This can be verified using a standard handheld multi meter set in Diode mode.
- Ground isolation beads are not populated
- I2C pull-up resistors are not populated

2.3 Communicating with the hardware

Using OPT3101SDK is the best way to communicate with the hardware. The SDK provides a simple interface to the hardware with register names accessible in the code. SDK also has several methods and routines to perform various calibration tasks. SDK acts as a backbone infrastructure for communication and control, however the actual implementation of physical I2C writes and environmental control needs to be written by the user. By default the SDK has example implementations for OPT3101EVM firmware. There are some environmental factors which needs to be set before making measurements which are denoted in the SDK routines with empty methods. Users can implement their own automation in those methods.

It is important to configure the SDK based on the hardware connections. For example, the electrical connections on the SDA_M and SCL_M pins needs to be set on the SDK. Depending on what is connected on the pin the hardware may not work unless the SDK is configured to match with the hardware. OPT3101 configurator tool has an entry field named **SDA_M/SCL_M Connection** which determines the configuration generated to set a specific register during initialization sequence.

Here are some typical problems that are encountered during this step.

- I2C communication ACK not received
  - This could be due to address mismatch between the SDK implementation and hardware. OPT3101 device has configurable slave address, details for the same can be found on OPT3101 datasheet.
- I2C communication ACK received but device registers not updated from default value
  - This happens when the SDA_M/SCL_M lines are floating or pulled to ground. A specific register needs to be set to force device to behave as I2C slave. This is handled by the OPT3101 configurator tool when it generates the initialization sequence based on user input.
- Register reads and writes are unreliable and inconsistent.
  - This could be due to timing violation on the I2C bus caused by improper value of pull-up resistor. This can be mitigated with resistor value according to the estimated bus capacitance as per the I2C specification.

2.4 Crosstalk and performance

Required list of items during first time bring up are listed here

- Masking tape which is opaque to the wavelength used by the system.
- Various different targets at different reflectivity.

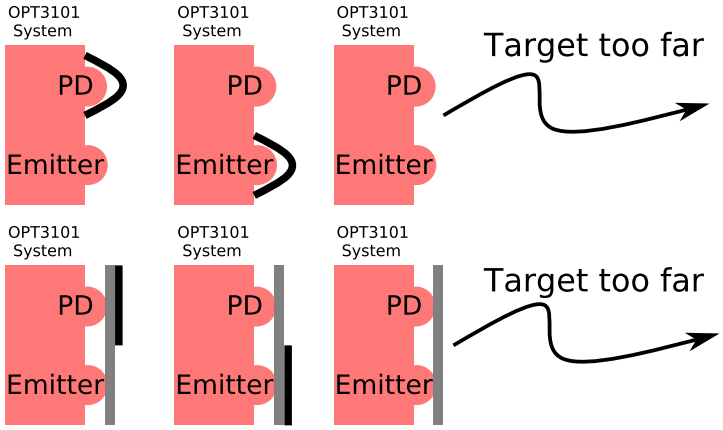
OPT3101SDK has the ***OPT3101::device::calibrationSession_firstTimeBringUp()*** method which is recommended as the first step. Details about what this method performs can be found in the detailed documentation as part of the OPT3101SDK

The method resets the device and runs initialization register writes.

- Internal crosstalk calibration and correction
  - This step helps estimate the crosstalk from the internal switching nets of OPT3101 to the input pins/circuit of OPT3101. During this estimation OPT3101 automatically turns the emitter channels off so as to estimate the pure effect of internal nets.
  - Crosstalk amplitude is estimated by the ***OPT3101::device::measureAndCorrectInternalCrosstalk()*** method part of the OPT3101SDK
  - Amplitude of this measurement is expected to be lower than **30** amplitude codes.
  - If higher amplitudes are observed, it could indicate one of the following
    - The INP and INM pins expect matching capacitances to the analog GND. Any mismatch in capacitance could show up as a higher amplitude in internal crosstalk.
    - Poor isolation between analog and digital grounds. The ferrite beads that isolate analog and digital supplies and grounds could be the problem. The ferrite beads are expected to have at least 500 ohms of impedance at 10 MHz, which needs to be verified in the ferrite bead data sheet.
    - Decoupling capacitors on the power supplies could be poorly performing. Capacitors that have low ESR at 10 MHz are recommended.

- OPT3101SDK corrects for the internal crosstalk and is ready for the next steps.
- Its important to understand that internal crosstalk is something that is recommenced to be performed every time the device is reset and initialized, thus eliminating the need to store internal crosstalk in non-volatile memory.

Figure 1. Illumination crosstalk measurement setup



- Illumination crosstalk calibration
  - This step helps estimate the crosstalk from the switching illumination nets and optical leakage between the emitter and receiver. This needs to be performed for each TX channel and each current setting. Since OPT3101 supports multiple channels and current settings based on the user configuration, the OPT3101SDK cycles though each of these settings.
  - Either the photodiode or the emitter needs to be covered with the masking tape for accurate estimation. Examples of such masking can be found in Figure 1. In systems which have a cover glass, it is recommended to apply the masking tape on top of the cover glass directly above the photodiode or the emitter. If it is impractical to identify the photodiode position behind a dark cover glass, it is recommended to point the system to a target which is way beyond the ranging distance or infinity. This step is denoted by the ***environmentalController::setTargetToInfinity_OR_coverPhotodiode()*** method in OPT3101SDK.
  - Crosstalk amplitudes for all the TX channels and current settings are expected to be lower than a few hundred amplitude codes. Values under **200** amplitude codes are considered very good. OPT3101 systems design document explains the significance of crosstalk and impact of crosstalk on system performance in section 8.1.
  - The crosstalk register values obtained in the previous step needs to be stored in non-volatile memory for future usage. In case of OPT3101SDK the default implementation is storing these values to a file.
  - Common problems encountered in this process:
    - Crosstalk amplitude reported is very high. This could be due to a number of reasons.
      - Light leakage from emitter to photodiode: This is a common cause for high crosstalk amplitudes, especially in prototype design phase. Optical isolation between the emitter and photodiode is crucial and details of the same can be found in OPT3101 systems design document section 7.2. It is important to remember that many items that look dark or opaque to naked eye may be transparent in infrared wavelengths. This is where the suggested infrared camera (NIR camera) helps. Using the NIR camera, light leakage and other potential issues can be easily identified.
      - Poor PCB layout style: OPT3101 design is sensitive to PCB layout and guidelines are provided in both OPT3101 datasheet section 10 and OPT3101 systems design document section 8. Some components such as through hole components need shielding isolation structures to achieve low crosstalk amplitude as covered in OPT3101 systems design document section 8.5. Experiments with different isolation structure would help debug the root cuase of such high crosstalk.
      - If crosstalk amplitude measured is equal to or higher than 843 amplitude codes, a register

needs to be set to apply correction. OPT3101 calibration document section 4.2.3 has more details on the same. The register can be set to the appropriate value in the ***OPT3101::device::initialize()*** method.

  - Tips for debugging high crosstalk: Conducting an experiment to obtain the crosstalk values for various different TX channel current levels can help debugging. Current coupling and optical coupling crosstalk amplitudes tend to have a linear relation with the TX channel current selection, where as the voltage coupling tends to have a highly non-linear relation. This property can be exploited by determining crosstalk amplitudes at high and low TX channel current settings. Based on the relation between TX channel current and crosstalk amplitude the determination of the dominant crosstalk phenomenon can be understood. This knowledge can be used in adding shielding or addressing problems in board design to improve the crosstalk.

- Stored crosstalk register values can be read and loaded to OPT3101 using the ***OPT3101::device::loadIllumCrosstalkSet()*** method. With this loaded to the device and the photodiode covered the amplitude read from the system should read close to 0. The uncorrected non-zero amplitude is considered as residue crosstalk and is crucial to determine the absolute accuracy of the system.
  - Typical residue crosstalk values are less than **10** amplitude codes.
  - With masking on the photodiode removed, amplitude readings from the system at various different targets at different distances can be noted down.
  - The noted amplitude value can be compared against the OPT3101 System Estimator tool for similar conditions. This is a good sanity check for the optical efficiency of the system and will help determine the performance of the system under various conditions.

3 Per design calibration process

This section of the document walks though the various calibrations that need to be performed per design. Based on a sample size of 5 to 10 boards there are some calibration coefficients that need to be estimated. These are calibration coefficients that would take too long to determine on a per unit basis. Once estimated the coefficients can be utilized across all units, keeping the per unit factory calibration time to a minimum.

3.1 Coefficients to be estimated

The order in which the coefficient are determined is very important. After determination, each coefficient needs to be applied to the device before determining the next coefficient. This ensures that the determination process isolates only the coefficients of interest. The following is the order in which the coefficients are expected to be determined.

- Illumination crosstalk temperature coefficient
- Phase offset temperature coefficient
- Phase ambient coefficient

3.2 Setup requirements

The following are the tools and setup required to estimate these coefficients.

- A thermal chamber big enough to fit the OPT3101 system and targets to achieve good amplitude for different emitter settings. More details are available in sections below.
- Masking tape as recommended earlier to cover up the photodiode.
- Bright ambient light source such as halogen lamp.
- Software tool capable of analyzing, plotting and calculating the line of best fit from the data generated by the SDK.

3.3 Thermal chamber usage

It is critical to maintain the chamber at a thermally stable point before making measurements, especially related to air flow. It is recommended to heat or cool the chamber to desired temperature, once the temperature is reached, turn the airflow off and let the chamber stabilize and then make measurements. For the most accurate estimation, data needs to be measured at every stable temperature point for a wide range of temperature points. The greater the number of temperature points collected and the larger the range of temperatures collected, more the accurate the estimation of the coefficients will be. This process is often time consuming. There is, however, a faster but less than perfect way which in most cases may be acceptable. This way is to heat the thermal chamber with OPT3101 system in it to 70°C or 80°C with forced air flow. Once the desired temperature is reached, the airflow can be turned off followed by waiting for a few minutes to settle down. With the desired data like crosstalk or phase offset continuously being read from the system, the thermal chamber can be let to cool down naturally with the doors of the chamber slightly open. This would cool down the chamber with minimal convection from the room temperature. Based on the room temperature, data from 80°C or 70°C to about 20°C or 25°C can be obtained within a few minutes.

3.4 Ambient light source

A bright ambient light source such a halogen lamp is required to be able to estimate the ambient phase offset coefficients. The goal here is to establish different ambient brightness conditions and measure the response from OPT3101 system. The coefficients are then estimated by fitting a piece-wise linear function to the measured response to compensate for the effect.

3.5 Data Analysis and coefficient estimation

The SDK uses simple 2 point data slope estimation methods for all the coefficients. This is just a example estimation for simplicity. In real life a more elaborate estimation is recommended using tools capable of plotting and straight line fitting. In the case of temperature coefficient estimation a simple linear fit is what is required, whereas in the case of the ambient coefficient a more complex 4 segment piece-wise-linear fit needs to be used. In the later sections some example plots and best line fits are shown.

3.6 Determining Coefficients

3.6.1 Illumination crosstalk temperature

Illumination crosstalk changes with temperature. Uncorrected crosstalk (residue crosstalk) results in measurement inaccuracy. This process estimates the variation of crosstalk with temperature. Data from this process needs to be plotted, analyzed and coefficients need to be determined. The coefficients when programmed to the OPT3101 device correct for the drift minimizing the residue crosstalk.

3.6.1.1 Data generation

The process here is same as described in Section 2.4 except that the data is collected over different temperature conditions. The method described in Section 3.3 can be used to collect data over temperature. The ***OPT3101::device::calibrationSession_perDesignCalibrationCrosstalkTemp()*** method has the routine to configure the device to collect the required data. Illumination crosstalk is always performed with a masked emitter as prescribed in Figure 1.

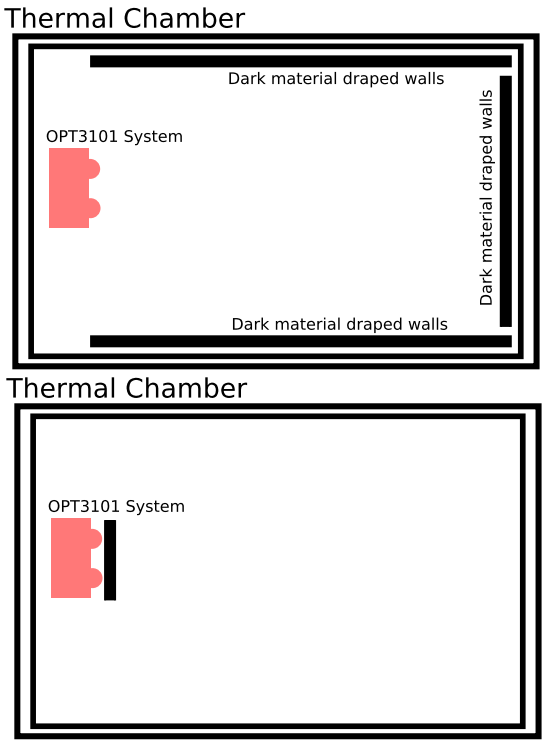


Figure 2. Illumination crosstalk temperature coefficient generation setup

Looking closely at the SDK documentation reveals the steps the ***OPT3101::device::calibrationSession_perDesignCalibrationCrosstalkTemp()*** method performs. The default implementation in the SDK is to generate data only for 2 temperature points. The temperature of the chamber is set using the ***environmentalController::setChamberTemperature()*** method which is an empty method. Users can implement their own chamber control commands inside the method. These methods are just example implementations of illumination crosstalk data generation across temperature. The real intent here is to be able to generate illumination crosstalk register values using ***OPT3101::device::measureIllumCrosstalk()*** and store them as files for processing and analysis later. Since coefficients need to be generated for all the TX channels and current settings, the SDK loops though the settings for each temperature.

Figure 2 shows examples of how the system can be set to measure this inside a thermal chamber. These are just examples of the physical setup and not absolute requirements. The most important aspect is being able to isolate pure illumination crosstalk as described in Section 2.4.

3.6.1.2 Data analysis and coefficient determination

OPT3101::crosstalkC::storeToFile() method is used to store crosstalk values for each temperature and TX channel and current settings based on the configuration. The SDK shows a simplified version of the coefficient estimation in the method ***OPT3101::calibrationC::illumCrosstalkTempCoff()***. In best practice a best line fit needs to be performed on the crosstalk register data for I and Q values independently. It is recommended to use the OPT3101 device internal temperature sensor as the X axis for the line fit. Each TX channel and current have two slope values to be stored. One for I and the other for Q. The class ***OPT3101::crosstalkTempCoffC*** provides the data structure to store these coefficients. There are member functions inside this class to store these in a file for future retrieval. The ***OPT3101::device::loadIllumCrosstalkTempCoff()*** and ***OPT3101::device::loadIllumCrosstalkTempCoffSet()*** methods provide a way to load the calibration coefficients to the hardware. The documentation included in the OPT3101SDK has more details regarding this. When the temperature crosstalk values are very low it might be hard to fit a line to the data and get a coefficient. In such cases the coefficient can be set to zero and the impact due to residue crosstalk can be calculated using the formula (1) provided in this application note.

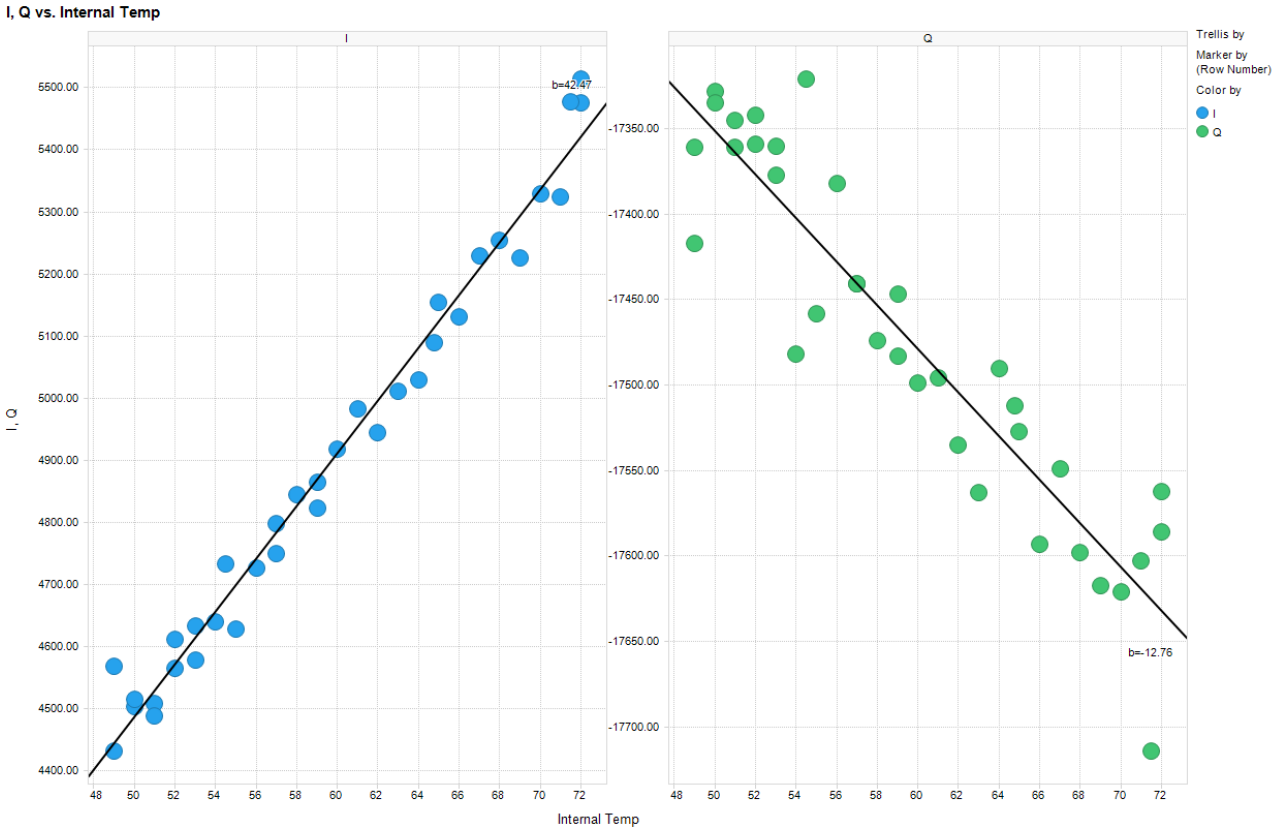


Figure 3. Illumination crosstalk coefficient determination

3.6.2 Phase offset temperature coefficient

OPT3101 systems are based on continuous wave time of flight which requires the phase offset to be programmed as part of factory calibration. The phase offset does vary with temperature. OPT3101 can be programmed with coefficients to compensate for this phenomenon. Since this is primarily caused by emitter characteristics, one coefficient is required per TX channel per current setting.

3.6.2.1 Data generation

Before measuring phase offset it is critical to program the illumination crosstalk temperature coefficients and perform illumination crosstalk correction. The recommended steps can also be found in the SDK method ***OPT3101::device::calibrationSession_perDesignCalibrationPhaseTemp()***. The ***OPT3101::device::loadIllumCrosstalkSet()*** method loads the crosstalk values from the file stored during the initial bring-up routine part of ***OPT3101::device::calibrationSession_firstTimeBringUp()***.

Phase offset is recommended to be measured with very high signal amplitude (between 16000 and 24000) to minimize the error from residue crosstalk. The target color or distance does not matter because the goal is to measure the variation of phase offset with temperature and not the phase offset itself. Since different TX channels and different current settings may produce different amplitude levels, it is possible that different targets at different distances may need to be used for every TX channel and current setting.

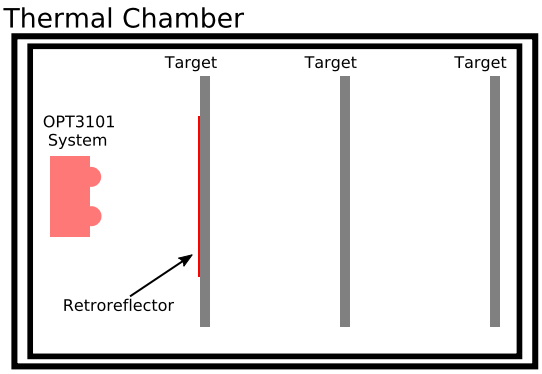


Figure 4. Thermal chamber setup for phase offset temperature coefficient estimation

Figure 4 shows an example of a typical setup where targets may have to be placed at different positions based on the TX channel and current setting to get the best amplitude for the measurement (16000 to 24000). These are some of the challenges that one may face during this stage.

- In case of very low current settings, if a high enough amplitude cannot be obtained, a retro reflective tape can be stuck on the target to provide amplitude gain.
- In cases where the thermal chamber is not big enough to accommodate a target far enough to not saturate the OPT3101 system, a dark low reflective target could be used to reduce the working range.
- A partially white/back or white/retro-reflective target could be used to balance the working distance and amplitude.

Looking closely at SDK documentation reveals the steps the ***OPT3101::device::calibrationSession_perDesignCalibrationPhaseTemp()*** method performs. The default implementation in the SDK is to generate data only for 2 temperature points. The temperature of the chamber is set using the ***environmentalController::setChamberTemperature()*** method which is an empty method. Users can implement their own chamber control commands inside the method. These methods are just example implementations of phase offset generation across temperature. The real intent here is to be able to generate phase offset values using **OPT3101::device::calibrationSession_perDesignCalibrationPhaseTemp()** and store them as files for processing and analysis later. Since target distance may need to be changed for each TX and current setting, the SDK method may need to modified according to the setup.

3.6.2.2 Data analysis and coefficient determination

OPT3101::phaseOffsetC::storeToFile() method is used to store phase offset values for each temperature and TX channel and current settings based on the configuration. The SDK shows a simplified version of the coefficient estimation in the method ***OPT3101::calibrationC::illumCrosstalkTempCoff()***. In practice a best line fit needs to be performed on the crosstalk register data for I and Q values independently. It is recommended to use the OPT3101 device internal temperature sensor as the X axis for the line fit. Each TX channel and current have two slope values to be stored. One for I and the other for Q. The class ***OPT3101::crosstalkTempCoffC*** provides the data structure to store these coefficients. There are member functions inside this class to store these in a file for future retrieval. The ***OPT3101::device::loadIllumCrosstalkTempCoff()*** and ***OPT3101::device::loadIllumCrosstalkTempCoffSet()*** methods provide ways to load the coefficients to the hardware. Documentation included in the OPT3101SDK has more details regarding this. When crosstalk values are very low the temperature it might be hard to fit a line to the data and get a coefficient. In such cases the coefficient can be set to zero and the impact due to residue crosstalk can be calculated using the formula (1) provided in this application note. Phase offset vs temperature is a linear fit. Depending on whether an external temperature is used or not, the slope can be obtained with internal temperature or the external temperature sensor. Figure 5 shows an example plot with line of best fit. The slope from the curve is used to determine the coefficient. ***OPT3101::calibrationC*** class in the SDK has methods to store and retrieve the coefficients for each TX channel and current setting to files.

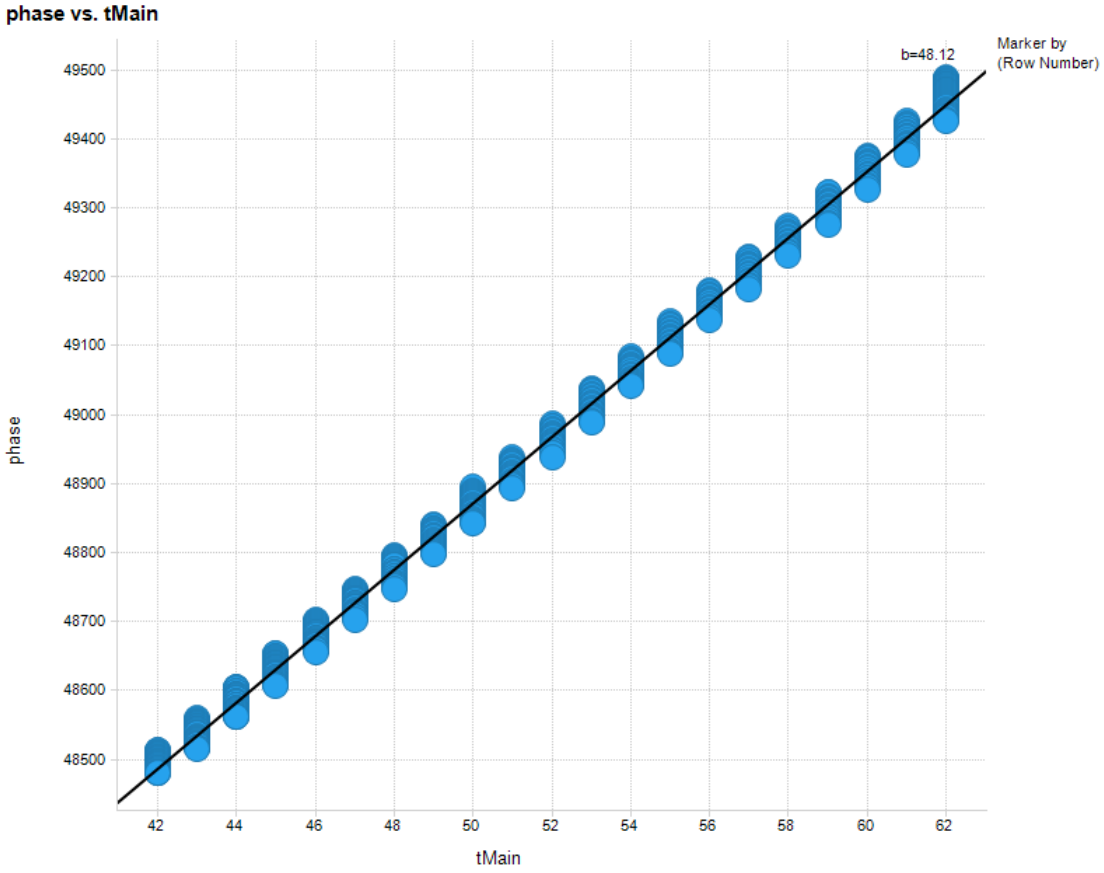


Figure 5. Phase offset vs internal temperature sensor

3.6.3 Phase offset Ambient coefficient

OPT3101 systems are based on continuos wave time of flight which require phase offset to be programed part of factory calibration. The phase offset does vary with ambient light depending on the properties of the photodiode, which needs to be compensated. OPT3101 can be programmed with coefficient to compensate for this phenomenon. Since this is primary caused by photodiode characteristics, one set of coefficient are applicable to all TX channel and current setting.

3.6.3.1 Data generation

Before measuring phase offset ambient coefficient it is critical to program the illumination crosstalk temperature coefficients, program the phase temperature coefficients and perform illumination crosstalk correction. The recommended steps can also be found in the SDK method ***OPT3101::device::calibrationSession_perDesignCalibrationPhaseAmbient()***. ***OPT3101::device::loadIllumCrosstalkSet()*** method loads the crosstalk values from the file stored during the initial bring-up routine part of ***OPT3101::device::calibrationSession_firstTimeBringUp()*** and phase temperature coefficients from the file stored in Section 3.6.2.2.

Similar to the phase temperature coefficient, phase offset is recommended to be measured with very high signal amplitude (between 16000 and 24000) to minimize error from residue crosstalk. The target color or distance does not matter because the goal is to measure the variation of phase offset with ambient and not the phase offset itself. To get the highest ambient level for a given ambient lamp, it is recommended to run this with the lowest current setting. Running at lowest current setting would help move the target closer to the OPT3101 system, making it easy to illuminate with target with ambient light.

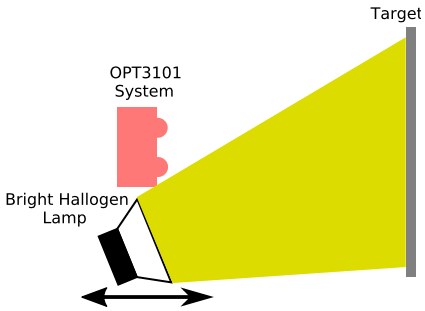


Figure 6. Phase ambient coefficient measurement setup

Figure 6 shows an example setup where a halogen lamp is used to generate ambient light on the target. The lamp can be moved closer or farther away from target to generate the desired ambient levels. This is handled by the SDK method ***environmentalController::setAmbientLight()*** which is an empty method. Users can implement methods to automate this process or pause the program for manual intervention. Phase measured vs ambient measured data is required for data plotting and analysis. A simple and quick way to generate the require data is to setup the OPT3101 system to stream phase and ambient data to a file continuously while moving the ambient light source closer to the target.

3.6.3.2 Data analysis and coefficient determination

A piece wise linear (PWL) curve fit is required to fit the Phase offset vs Ambient data generated. Figure 7 shows an example data plot between the phase offset and ambient levels. There are a few things to note in this plot. The blue line is the data generated from the sensor system. Reference data captured as part of data generation step with the ambient light source turned off is subtracted from the ambient data to generate clean X axis starting with 0. The same process is applied to the phase offset which generates a clean plot with the origin as one of the data points.

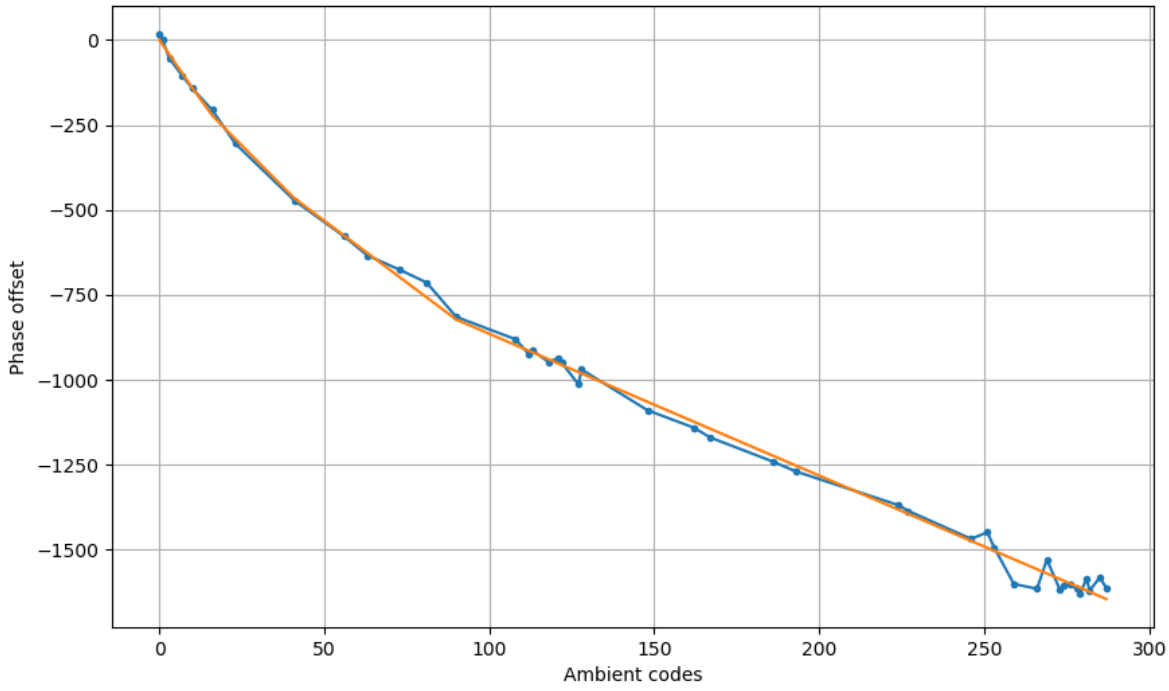


Figure 7. Phase offset vs Ambient data

Split points for the PWL can be chosen based on the performance curve and desired accuracy. Once the slopes are calculated, these need to be programmed to the SDK using the method ***OPT3101::device::calibration_perDesignCalibrationPhaseAmbientSetCoffAfterCurveFit()***. The SDK translates these coefficients to appropriate register writes. It is critical to isolate and measure the phase offset change only with respect to ambient by maintaining the target at a constant distance. A common challenge is to be able to achieve sufficient ambient range with this setup. A retro reflective tape could also be used to boost the ambient levels. For narrow field of view systems, ambient light could be directly pointed at the system from the side (outside of the field of view). This would also help collect higher ambient levels.

4 Per unit Factory calibration

This section of the document walks though the various calibrations that need to be performed per unit in a factory production environment. Each and every unit of the OPT3101 system needs some calibration steps before it is capable of generating accurate data. Depending on the accuracy required, the manufacturing tolerance, and the variation in the calibration data over a period of time, it is possible to eliminate the need for per unit calibration and program all the units with a common value obtained over a large sample of units. This is subjected to the factors mentioned above and is left to the discretion of the user.

4.1 Setup and Tools required

Performing per unit factory calibration involves two steps. First determination of illumination crosstalk, followed by the phase offset measurement.

- Setup to measure illumination crosstalk:
  - This typically involves masking the optical component or pointing to a target way beyond the ranging distance of the OPT3101 system. Detailed requirements are covered as part of Section 2.4.
  - In the case of an operator handled calibration setup, applying a masking tape on the photodiode or pointing towards a target that is far enough (like a distant wall or ceiling) as shown in Figure 1 is practical.
  - In the case of a robot automated calibration setup, pointing the system to a far enough target or a specially designed dark box with walls covered with a very low reflectivity material is practical.
- Setup to measure phase offset:
  - Targets are required at known distances to calibrate the phase offset.
  - Target selection is based on the OPT3101 system configuration. Each TX channel may needs own target if they are pointing in different directions.
  - Phase offset needs to be determined for each TX channel and current settings. It is recommended to place the target in a position that represents the real use case scenario in a system. For example, if one or more channels are pointing at an angle and expected to detect a flat target at an angle and report a distance, it is then recommenced to place the calibration for target at a similar position.
  - Distance to the target and the reflectivity needs to be identified based on the amplitude the OPT3101 system generates. Phase offset determination is recommended at signal amplitudes of 16000 to 24000 to get best results.
    - This is not a strict requirement but a recommendation. The guideline shows the amplitude range to minimize error from residue crosstalk. If for the TX channel and the current setting the residue crosstalk is < 10 amplitude codes, the phase offset can be performed at the lower signal amplitude. The distance error would proportionally be higher depending on the formula (1) specified in this application note.
  - Distance at which this amplitude is achieved may be dependent on the TX channel and current setting. It is often impractical to have a single target that satisfies all the current settings and TX channels. For example if with a 170 mA current setting on TX0, the target at 0.5 meter is able to produce a signal amplitude of 24000 codes, then for a current of 42.5mA , the same target at a distance of 0.25 meters would produce amplitude of 24000 codes.

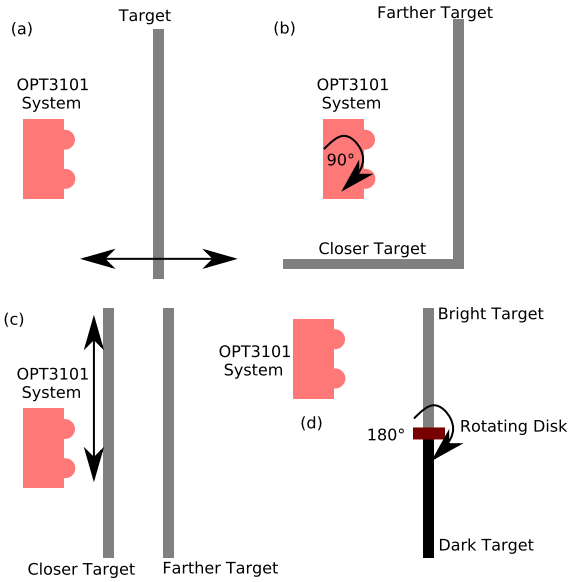


Figure 8. Phase offset calibration setup

Figure 8 shows a few example setup for phase offset factory calibration with multiple targets.

a. shows an example setup where the target moves to different distances to achieve the best amplitude based on the TX Channel and current setting.
b. shows an example setup where two targets are placed at different distances orthogonal to the OPT3101 system. OPT3101 system is mounted on a rotating housing which can point to either of the target based on current setting calibrated.
c. shows an example setup where a closer target falls in between the farther target and OPT3101 systems like a curtain. This setup enables two targets at different distances.
d. shows an example setup where two different reflectivity targets are placed on a rotating wheel. The wheel is rotated based on target desired

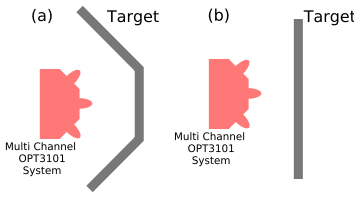


Figure 9. Phase offset calibration setup

Figure 9 shows an example setup for a multiple TX channel configuration. Based on the application requirement, the setup which closely represents the real work use case needs to be chosen.

4.2 Factory calibration

Factory calibration of assembled units is handled by the ***OPT3101::device::calibrationSession_perUnitFactoryCalibration()*** method. Detailed steps of what is actually performed in this method can be reviewed in the SDK documentation. Once the system is powered-up, reset applied, and initialized, the internal crosstalk correction is performed. The system is now ready for factory calibration.

Coefficients determined from Section 3 are programmed to the device. There are several methods like ***OPT3101::device::loadIllumCrosstalkTempCoffSet()*** ***OPT3101::device::loadPhaseOffsetTempCoffSet()*** ***OPT3101::device::loadPhaseAmbientCoffSet()*** that load the coefficients to the device from files stored as part of per design calibration. This is followed by illumination crosstalk determination for all the TX channels and current settings. With all the coefficients and crosstalk registers programmed the system is now ready for phase offset calibration.

4.2.1 Per unit Illumination crosstalk

As described in the setup section, the illumination crosstalk is performed for all the TX channel and current settings with the optical components masked. The SDK loops though all of the TX channels and current settings running the ***OPT3101::device::measureIllumCrosstalk()*** method which measures and stores the illumination crosstalk values in registers. It is recommended to have a quality control check on the crosstalk amplitude measured. The limits could be relaxed during initial production lots and could be tightened based on manufacturing tolerance. It is also recommended to log the crosstalk files generated by the SDK for lot tracing and statistical analysis for limit changes in future.
OPT3101::device::loadIllumCrosstalkSet() and ***OPT3101::device::loadIllumCrosstalkTempCoffSet()*** load the device with the crosstalk correction registers. With these methods run the amplitude read from the OPT3101 system is expected to be < 10 amplitude codes for all the TX channels and current settings (with the masks still in place). The amplitude measured during this step could also be used for quality control.

4.2.2 Per unit Phase offset

The system now fully corrected and compensated for crosstalk is ready to measure phase offset. ***OPT3101::device::measurePhaseOffset()*** of the SDK generates the phase offset for each TX channel and current setting. Since each TX channel and current setting may require a change in the target setup the ***environmentalController::setTargetDistance()*** which is an empty method is run. Users can implement their own hardware control to achieve the movement required from the target. Phase offset and other phase offset related coefficients measured by the SDK are stored as files which are loaded on to the device by ***OPT3101::device::loadPhaseOffsetSet()***, ***OPT3101::device::loadPhaseOffsetTempCoffSet()*** and ***OPT3101::device::loadPhaseAmbientCoffSet()*** methods. The system now is fully calibrated with all the required coefficients and compensations programmed.

4.2.3 Per unit storing calibrating data in non-volatile memory

OPT3101 does not have non-volatile memory and needs calibration data to be stored in non-volatile memory. One option is to store these settings in an external EEPROM, which can be directly connected to OPT3101 I2C master pins. OPT3101 datasheet section 7.4.2 has more details regarding use of external EEPROM. The other alternative is to store the calibration data in host non-volatile memory. If an external EEPROM connected to OPT3101, OPT3101 is capable of autoloading calibration data on power-up. In the case of systems where the calibration data is stored in host non-volatile memory, the host needs to load the calibration data to OPT3101 explicitly on power-up after initialization.

4.2.3.1 Data and format of non-volatile memory

OPT3101 has a register word size of 3 bytes (24 bits) and has a 8 bit register address. A list of addresses to be stored in non-volatile memory is generated by the OPT3101 configurator tool. ***OPT3101::calibrationC::registerAddressList*** is an array of register addresses which are required to be stored. The ***OPT3101::device::calibrationSession_perUnitFactoryCalibrationWriteRegisterDataToNonVolatileMemory()*** method handles storing the registers to an external EEPROM. In the case of systems where there is no external EEPROM the method needs to be implemented by the user to read the register data from the device and store it in the host non-volatile memory. The data format for the storage is shown in Table 27 of OPT3101 datasheet.

5 System mode operation in field

5.1 System operation

In the field the following are the recommended steps for power-up and effective operation of the system.

- System power-up
- OPT3101 device reset
- OPT3101 device initialization using ***OPT3101::device::initialize()***
- Correct for internal crosstalk using ***OPT3101::device::measureAndCorrectInternalCrosstalk()***
- If an external EEPROM is connected to the OPT3101 I2C master pins and programmed with calibration data as prescribed in Section 4, the system is ready and fully calibrated and operational.
- If no external EEPROM is connected then the host needs to load the calibration data stored in non-volatile memory to OPT3101 device. Once calibration registers listed in ***OPT3101::calibrationC::registerAddressList*** are loaded to the device, the system is ready and fully calibrated and operational.

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATASHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, or other requirements. These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you will fully indemnify TI and its representatives against, any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI’s products are provided subject to TI’s Terms of Sale (www.ti.com/legal/termsofsale.html) or other applicable terms available either on ti.com or provided in conjunction with such TI products. TI’s provision of these resources does not expand or otherwise alter TI’s applicable warranties or warranty disclaimers for TI products.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265