

# PSOC™ 4 Liquid Level Sensing (LLS) training manual

## About this document

This training manual covers the labs for the introduction to liquid level sensing on the PSOC™ 4000T and the PSOC™ 4100T Plus training.

## Scope and purpose

The labs will cover objectives, project creation, tuning liquid level sensing sensors, and calibrating the foam rejection coefficient.

## Intended audience

The intended audiences for this document are design engineers, technicians, and developers of electronic systems.

## Table of contents

### Table of contents

<b>About this document .....</b>	<b>1</b>
<b>Table of contents .....</b>	<b>2</b>
<b>1     Introduction .....</b>	<b>3</b>
<b>2     Required development tools and prerequisite .....</b>	<b>4</b>
2.1     Tools.....	4
2.2     Prerequisites .....	4
<b>3     Liquid-level factory calibration procedure .....</b>	<b>5</b>
3.1     Objective .....	5
3.2     Description.....	5
3.3     Project creation .....	5
3.4     Disabling foam rejection feature.....	9
3.5     Liquid level calibration procedure .....	11
3.6     Output.....	18
3.7     Conclusion.....	18
<b>4     Foam rejection calibration.....</b>	<b>19</b>
4.1     Description.....	19
4.2     Enabling foam rejection .....	19
4.3     Initial calibration for foam rejection enabled sensors.....	21
4.4     Foam rejection calibration procedure.....	24
4.5     Output.....	25
4.6     Conclusion.....	25
<b>References .....</b>	<b>26</b>
<b>Glossary .....</b>	<b>27</b>
<b>Revision history.....</b>	<b>28</b>
<b>Disclaimer.....</b>	<b>29</b>

---

## Introduction

### 1 Introduction

This manual provides instructions to create, configure, and build the **PSOC™ 4000T MSCLP liquid level sensing** project to optimize the capacitive performance and tune for the liquid level sensing algorithms.

## Required development tools and prerequisite

## 2 Required development tools and prerequisite

### 2.1 Tools

- [ModusToolbox™ software v3.6](#) or later (tested with v3.6)
- ModusToolbox™ CAPSENSE™ and Multi-Sense Pack
- [CY8CPROTO-040T-MS](#), the PSOC™ 4000T Multi-Sense Prototyping Kit



**Figure 1** PSOC™ 4000T Multi-Sense Prototyping Kit

### 2.2 Prerequisites

1. [Introduction to PSOC™ 4000T Training](#)
2. Install [ModusToolbox™ software v3.6](#)
3. Install the Eclipse IDE for ModusToolbox™ from [ModusToolbox™ Setup](#).
4. Install the ModusToolbox™ CAPSENSE™ and Multi-Sense Pack

## Liquid-level factory calibration procedure

### 3 Liquid-level factory calibration procedure

#### 3.1 Objective

This manual aims to demonstrate how to implement capacitive liquid-level sensing using PSOC™ 4 microcontrollers equipped with CAPSENSE™ with Multisense. The goal is to perform non-contact, accurate measurement of liquid levels in non-conductive containers, with the added ability to reject foam interference.

The objective of this experiment is to implement and evaluate a capacitive liquid level sensing system using Infineon's PSoC™ 4 CAPSENSE™ technology. The experiment aims to demonstrate accurate non-contact measurement of liquid levels in non-conductive containers, understand the effect of capacitance variation caused by the presence of liquid, perform calibration procedures to ensure reliable detection across varying conditions.

#### 3.2 Description

The CAPSENSE™ block in PSoC™ 4 devices operate by charging and discharging the sensor electrode and then measuring the resulting signal to determine capacitance. Functionally, it behaves like a high-resolution sigma-delta converter that translates the sensor's capacitance into digital counts. The total capacitance measured is the sum of the sensor's parasitic capacitance and the additional capacitance introduced when a liquid comes in contact with the sensing region. As the liquid level rises, this effective capacitance increases, and the CAPSENSE™ hardware registers higher counts. By processing these digital values, the system can accurately map the change in capacitance to the corresponding liquid level inside the container.

#### 3.3 Project creation

1. Create a workspace in ModusToolbox™ software by opening lastest Eclipse IDE.

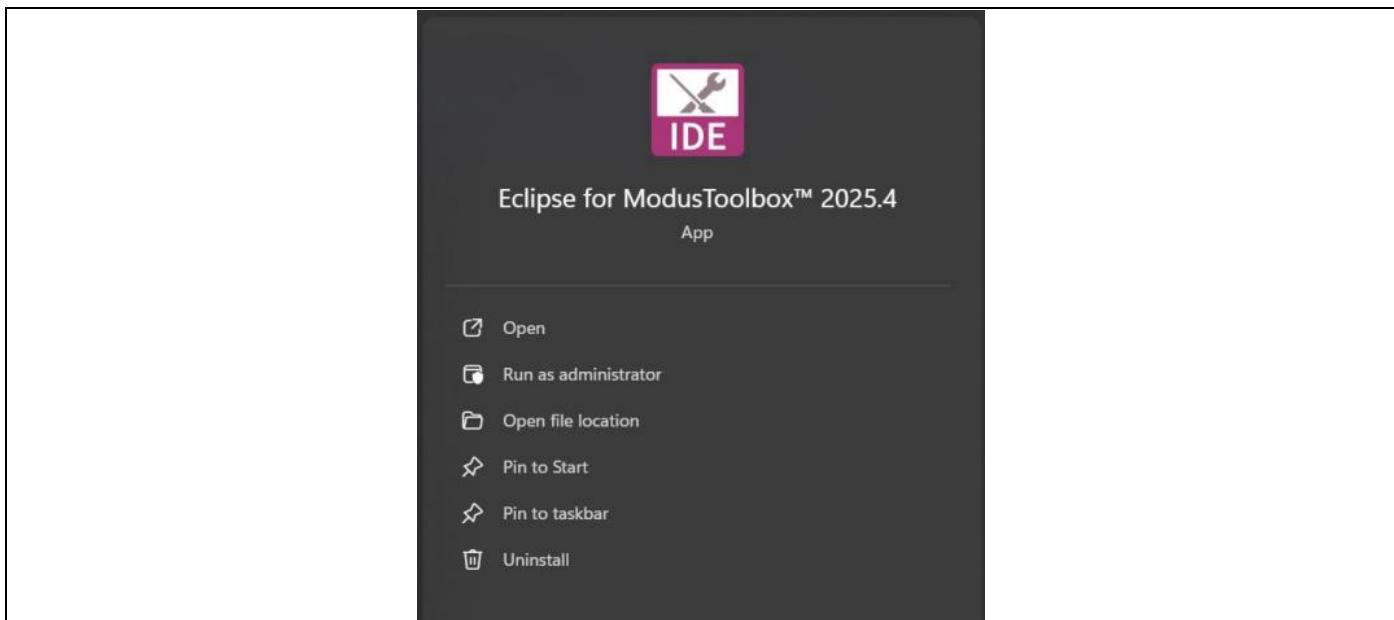
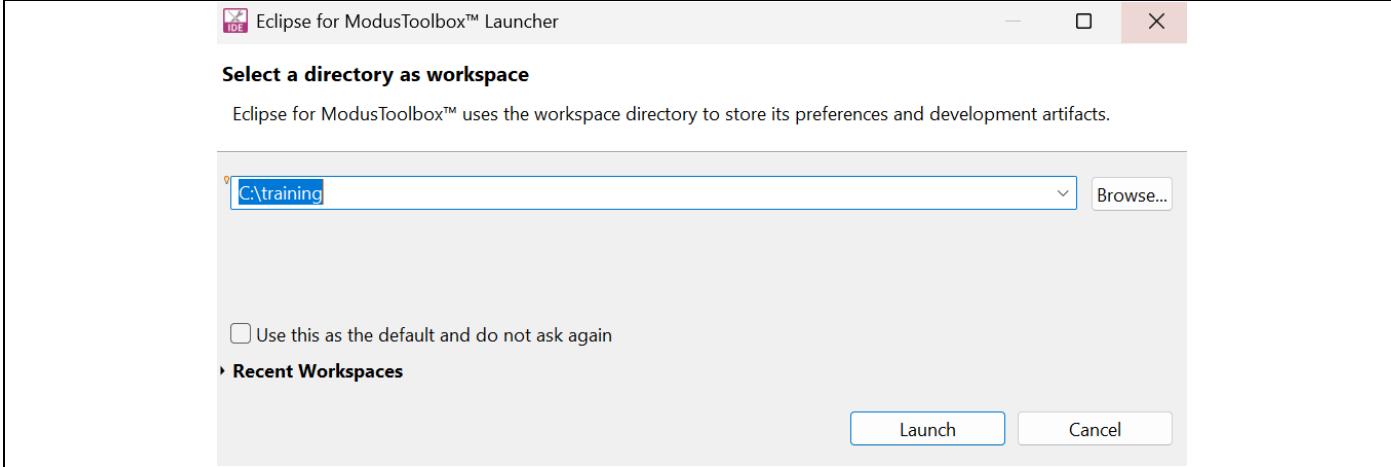


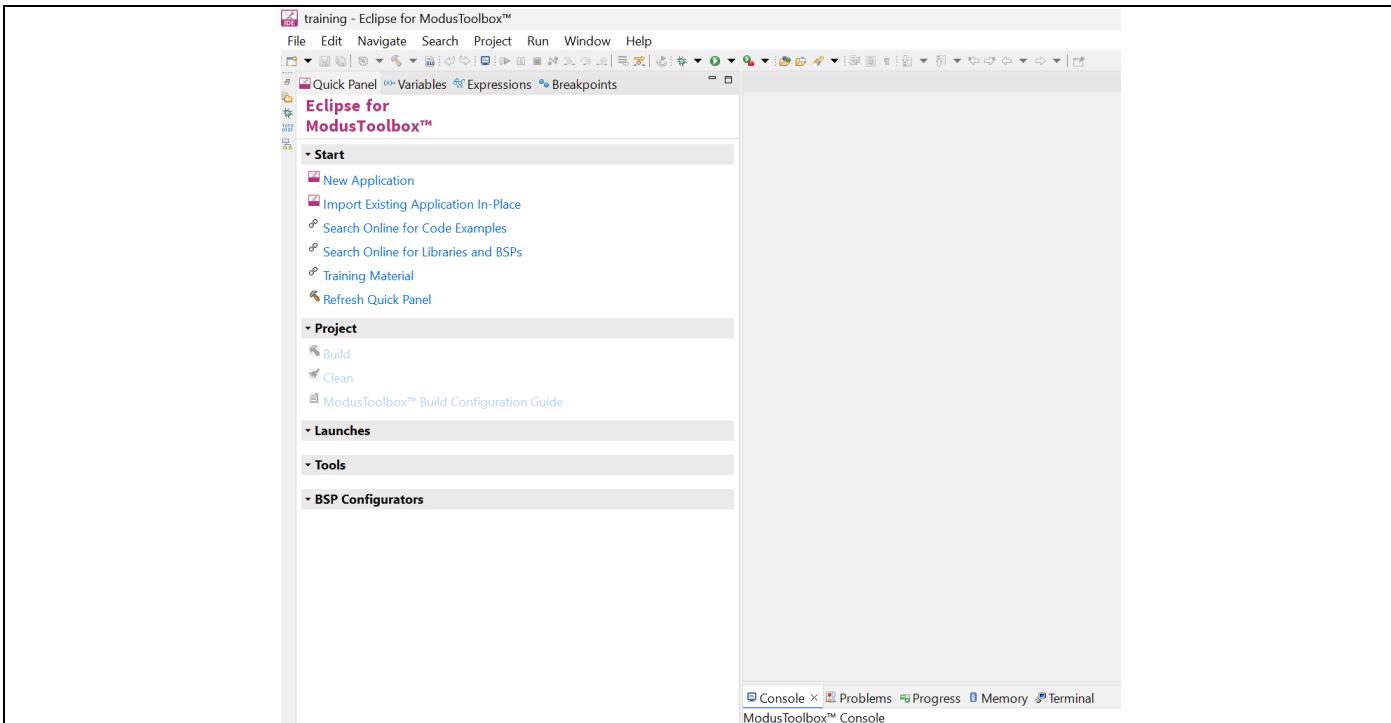
Figure 2 Opening the Eclipse IDE

## Liquid-level factory calibration procedure

2. Choose a workspace directory for your project and click on **Launch** button.



**Figure 3 Selecting a directory as workspace**

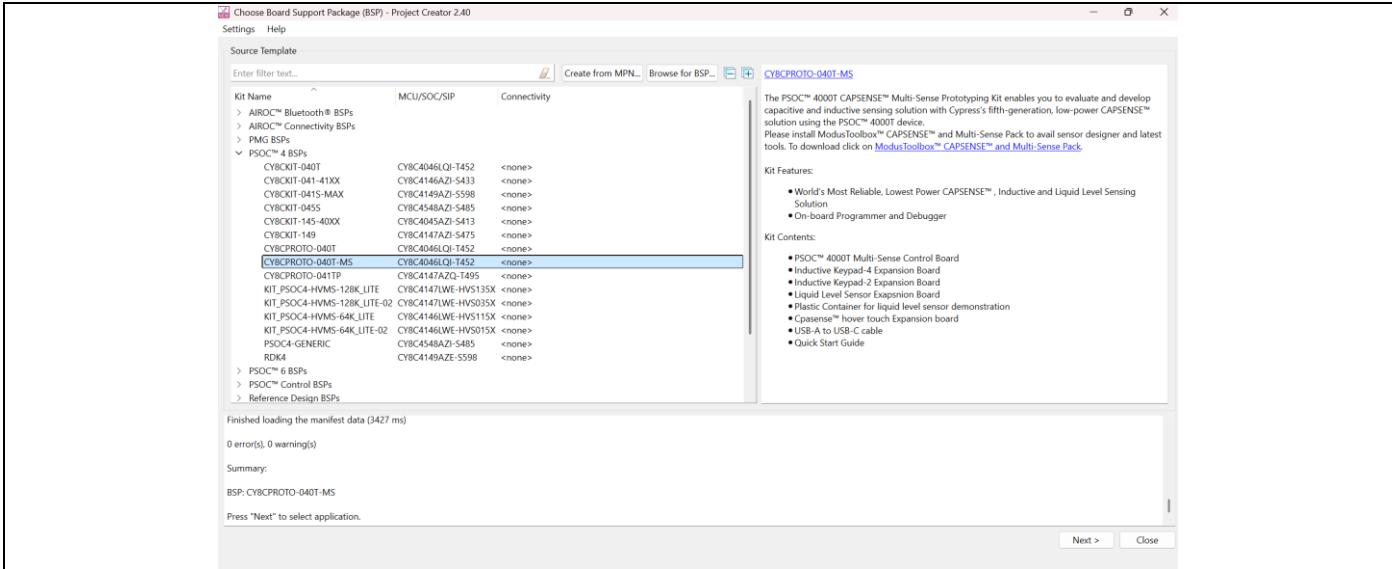


**Figure 4 New workspace**

3. Select **New Application** from the quick panel for creating new project.

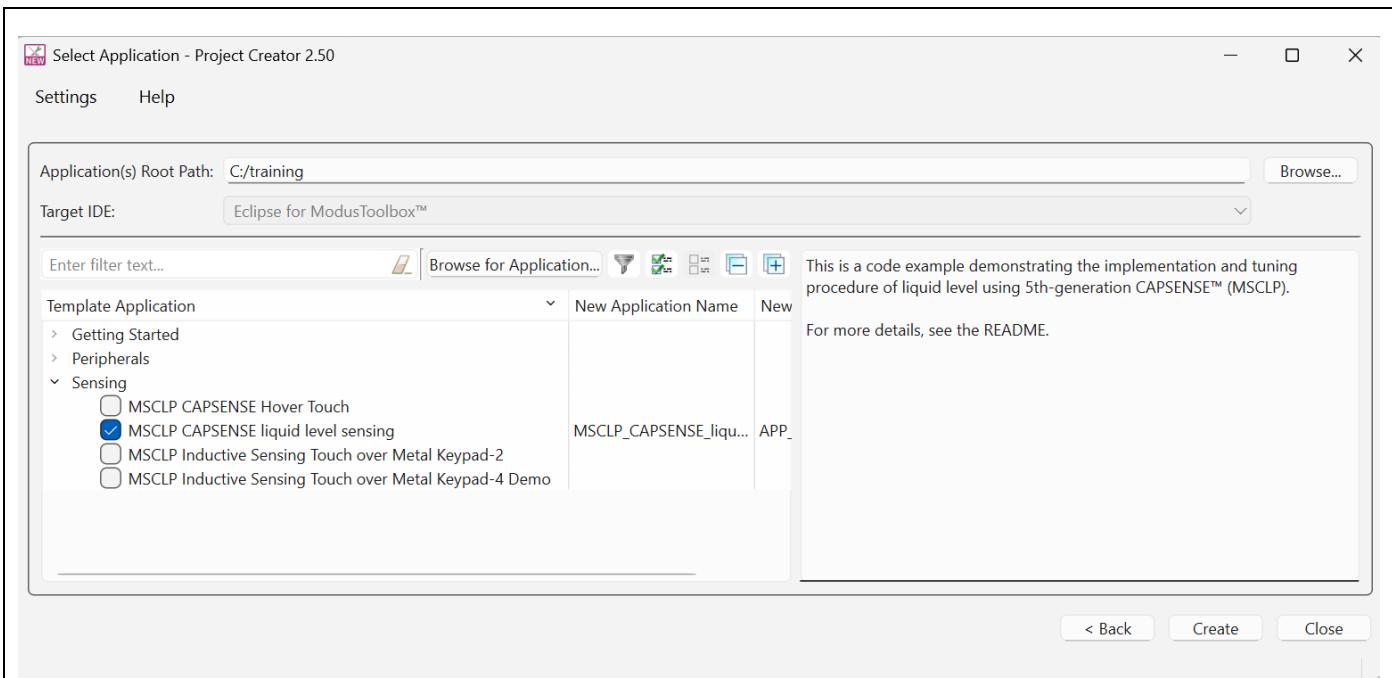
## Liquid-level factory calibration procedure

### 4. Select the CY8CPROTO-040T-MS BSP, then press Next.



**Figure 5 Select the CY8CPROTO-040T-MS BSP**

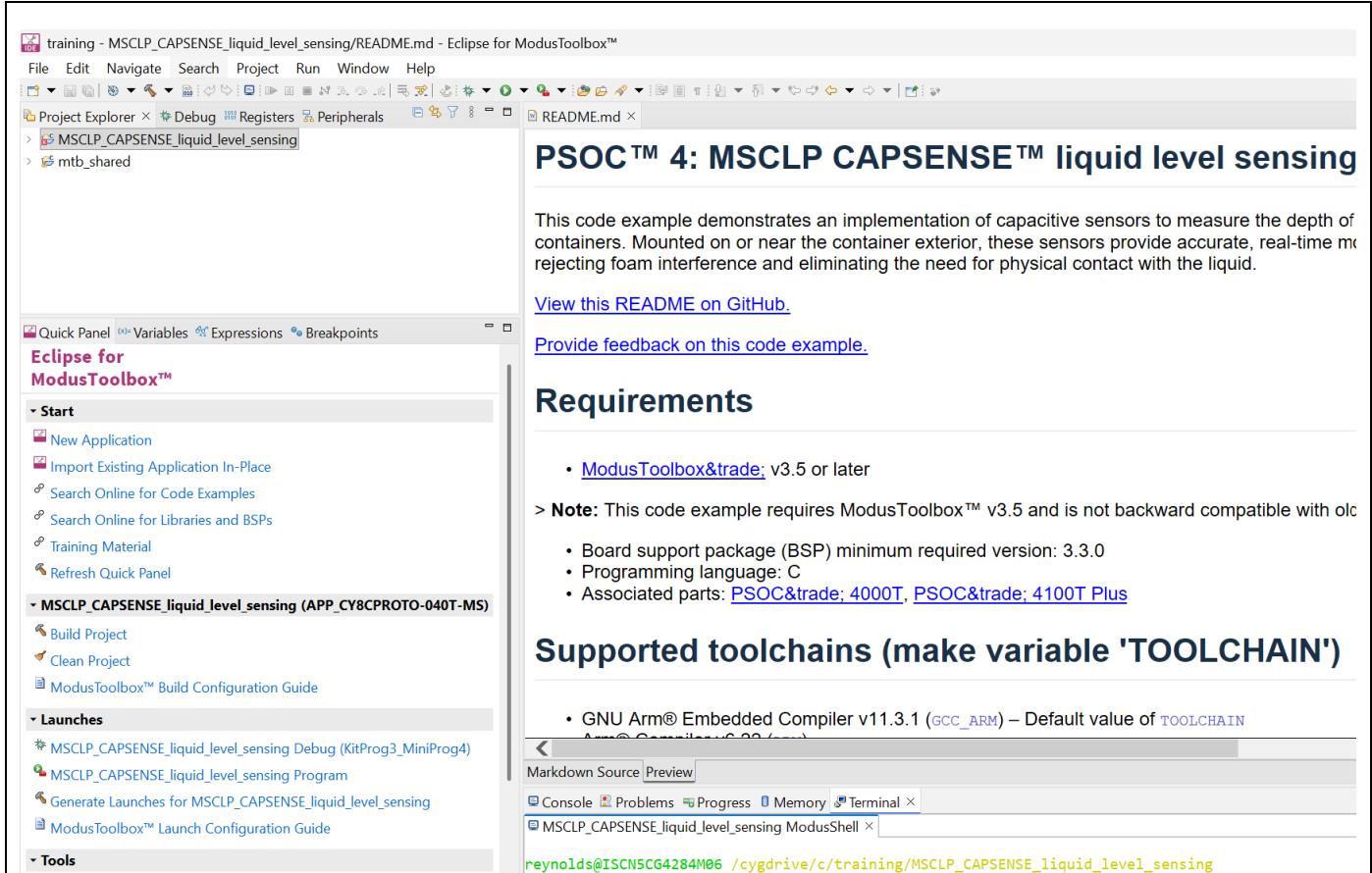
### 5. Select the MSCLP CAPSENSE™ liquid level sensing application and click on Create.



**Figure 6 Select the MSCLP CAPSENSE™ liquid level sensing application**

## Liquid-level factory calibration procedure

6. When you click create, a new project is created in the project explorer.



**Figure 7 Example application created**

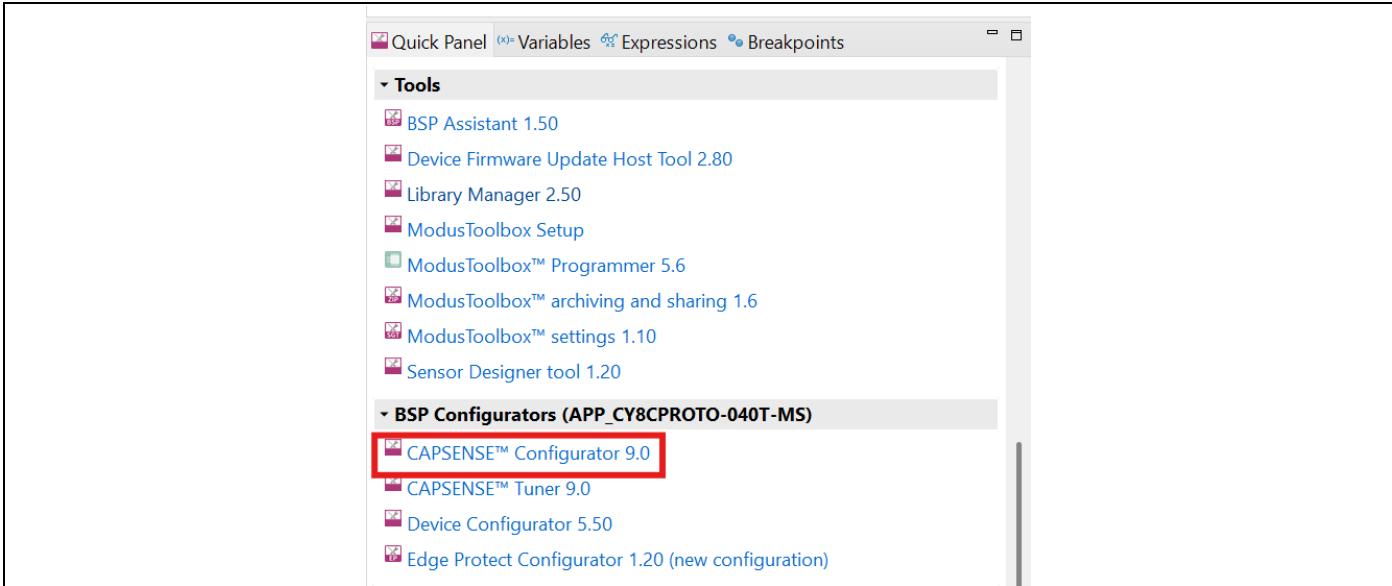
## Liquid-level factory calibration procedure

### 3.4 Disabling foam rejection feature

This code example enables the foam rejection feature by default. We will disable that here to show how to calibrate a liquid level sensor without the foam rejection feature.

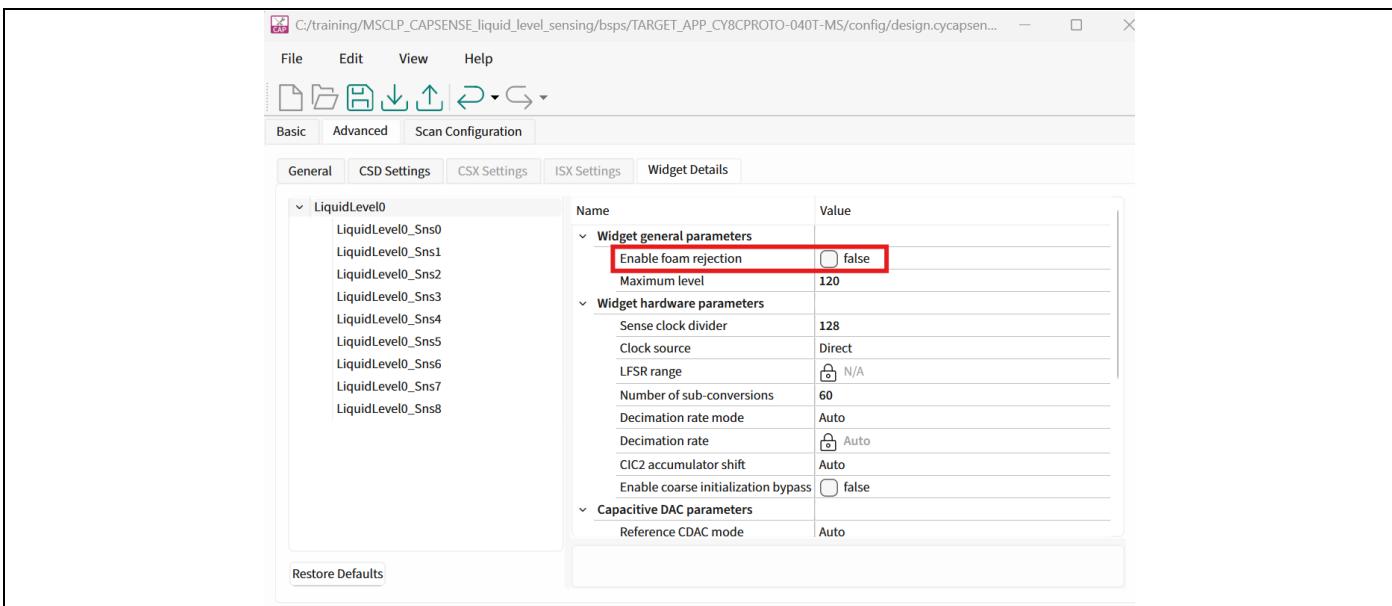
Note: You can complete the liquid level calibration procedure with the foam rejection feature enabled. You must first run the [initial calibration for foam rejection enabled sensors](#).

1. Open the CAPSENSE™ configurator.



**Figure 8** Opening the CAPSENSE™ Configurator

2. Navigate to the **Advanced** tab then go to the **Widget Details**. For the **LiquidLevel0** widget, disable the **foam rejection** feature



**Figure 9** Disable the foam rejection feature

3. Save the CAPSENSE™ configurator.

## Liquid-level factory calibration procedure

- In the Eclipse IDE, open the main.c file and comment out the lines associated with the foam rejection widget.

```

for (;;)
{
    // uint32_t level_w_FR, level_wo_FR;
    /* Scan the normal Liquid Level Widget */
    Cy_CapSense_ScanWidget(CY_CAPSENSE_LIQUIDLEVEL0_WDGT_ID, &cy_capsense_context);
    /* Wait until the scan is finished */
    while(Cy_CapSense_IsBusy(&cy_capsense_context)) {}

    /* Scan the Foam Rejection Widget */
    // Cy_CapSense_ScanWidget(CY_CAPSENSE_LIQUIDLEVEL0_FR_WDGT_ID, &cy_capsense_context);
    /* Wait until the scan is finished */
    // while(Cy_CapSense_IsBusy(&cy_capsense_context)) {}

    /* Process all th widgets */
    Cy_CapSense_ProcessAllWidgets(&cy_capsense_context);

    /* Send CAPSENSE™ data to the Tuner */
    Cy_CapSense_RunTuner(&cy_capsense_context);

    /* store the liquid level before and after foam rejection */
    // level_wo_FR = CY_CAPSENSE_LIQUIDLEVEL0_PTRPOSITION_VALUE->x;
    // level_w_FR = CY_CAPSENSE_LIQUIDLEVEL0_FR_PTRPOSITION_VALUE->x;
    //

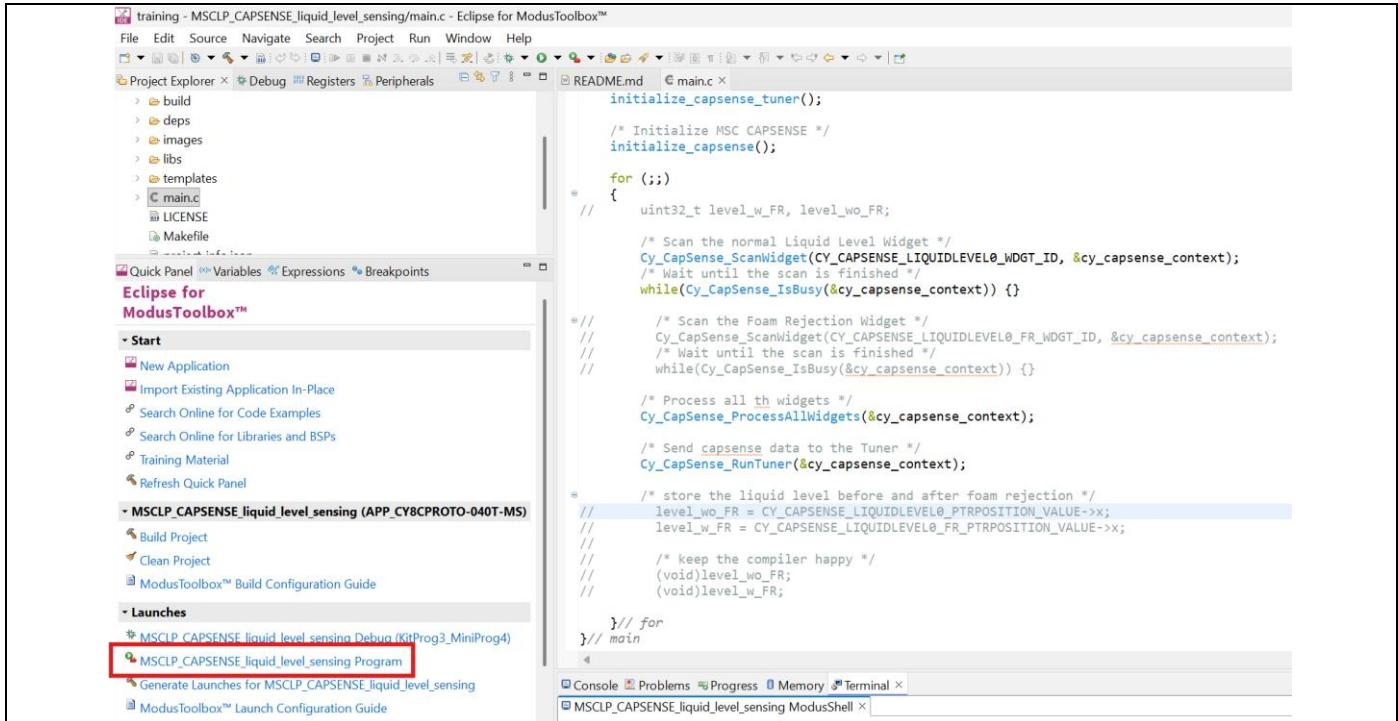
    /* keep the compiler happy */
    // (void)level_wo_FR;
    // (void)level_w_FR;

}

```

**Figure 10 Comment out foam rejection widget**

- Program the device



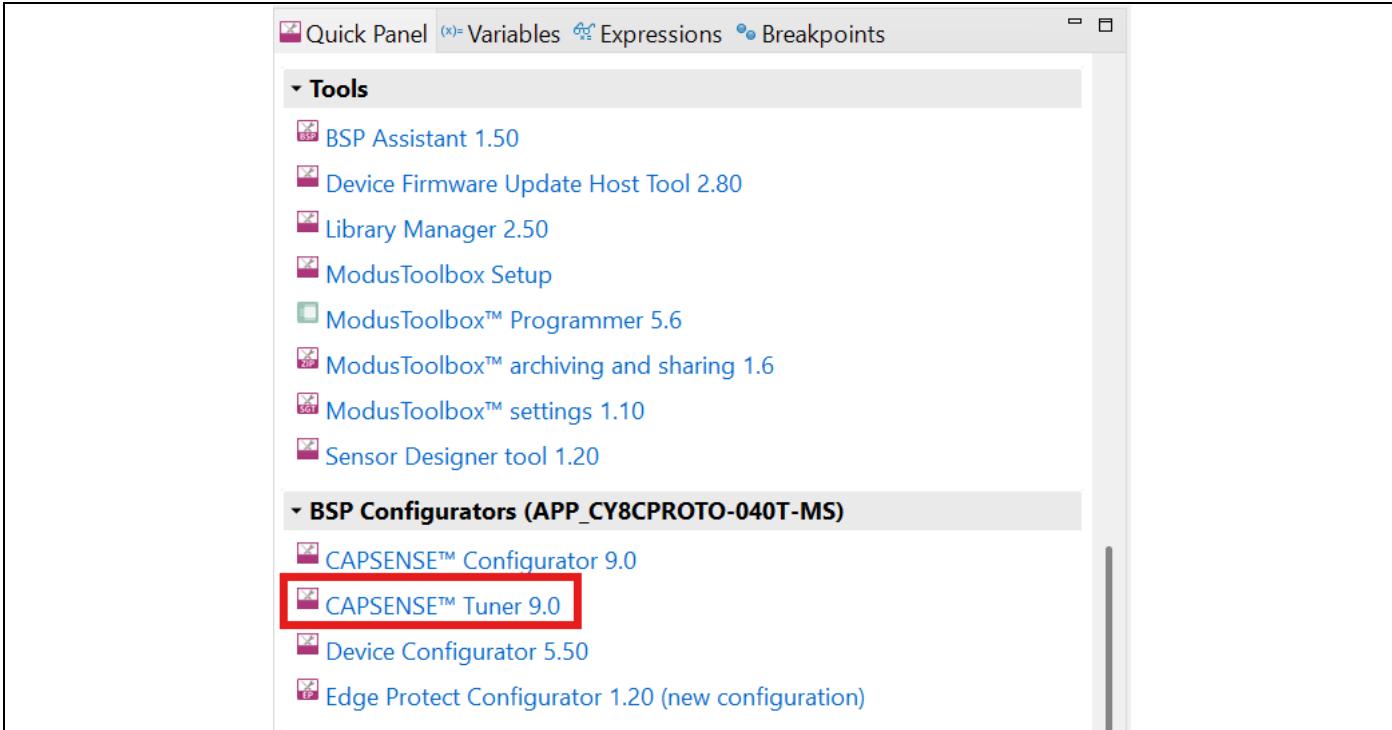
**Figure 11 Program the device**

## Liquid-level factory calibration procedure

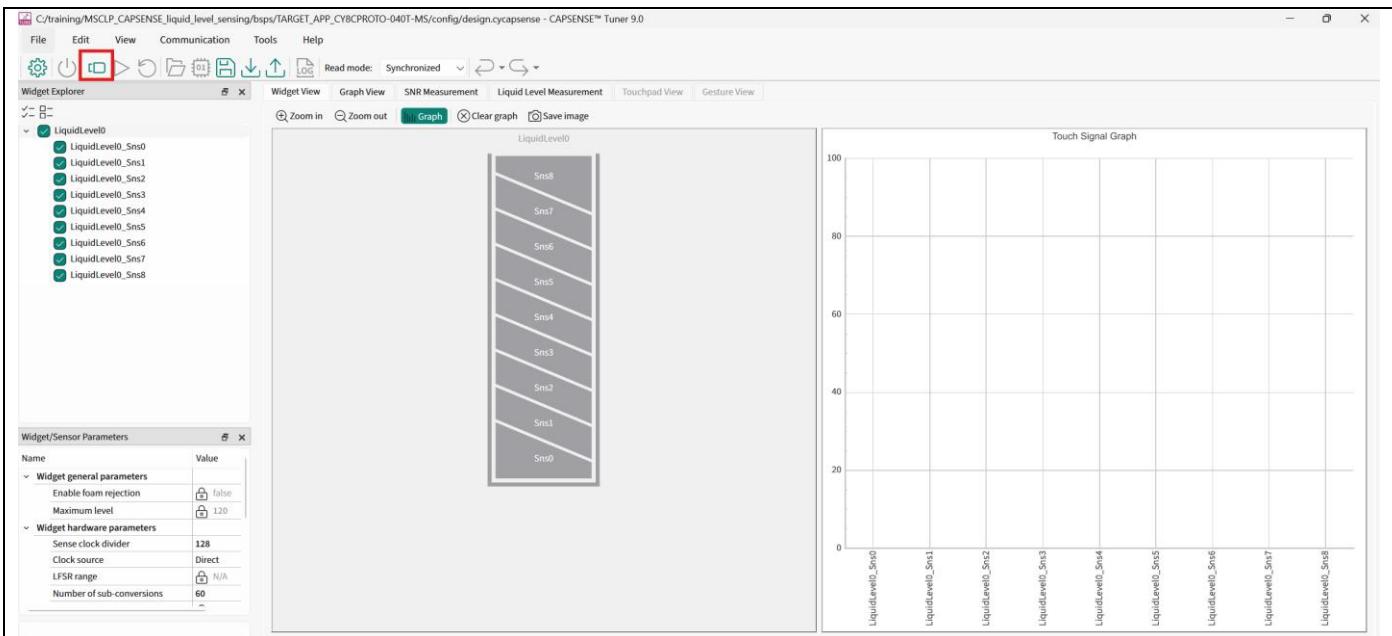
### 3.5 Liquid level calibration procedure

Before calibrating, ensure that the liquidlevel sensors are properly tuned following the guidance in the “Tuning Procedure” section of PSOC™ 4: [MSCLP CAPSENSE™ liquid-level sensing code example](#)

1. Open the CAPSENSE™ Tuner and connect it to the device



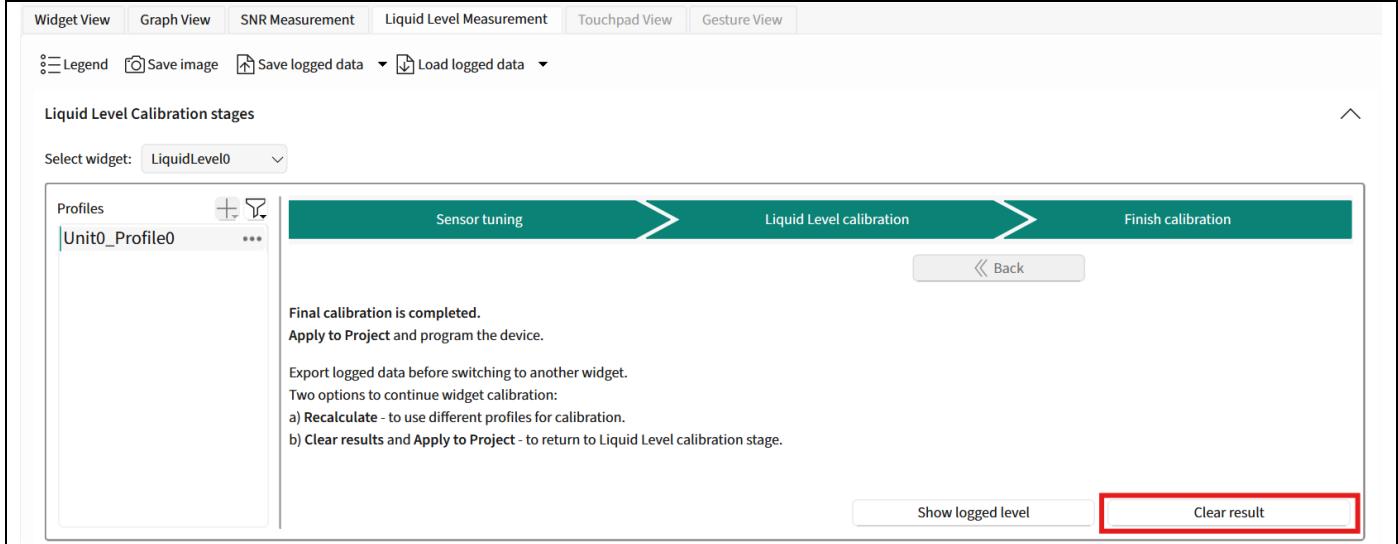
**Figure 12** Opening the CAPSENSE™ Tuner



**Figure 13** CAPSENSE™ Tuner Widget view

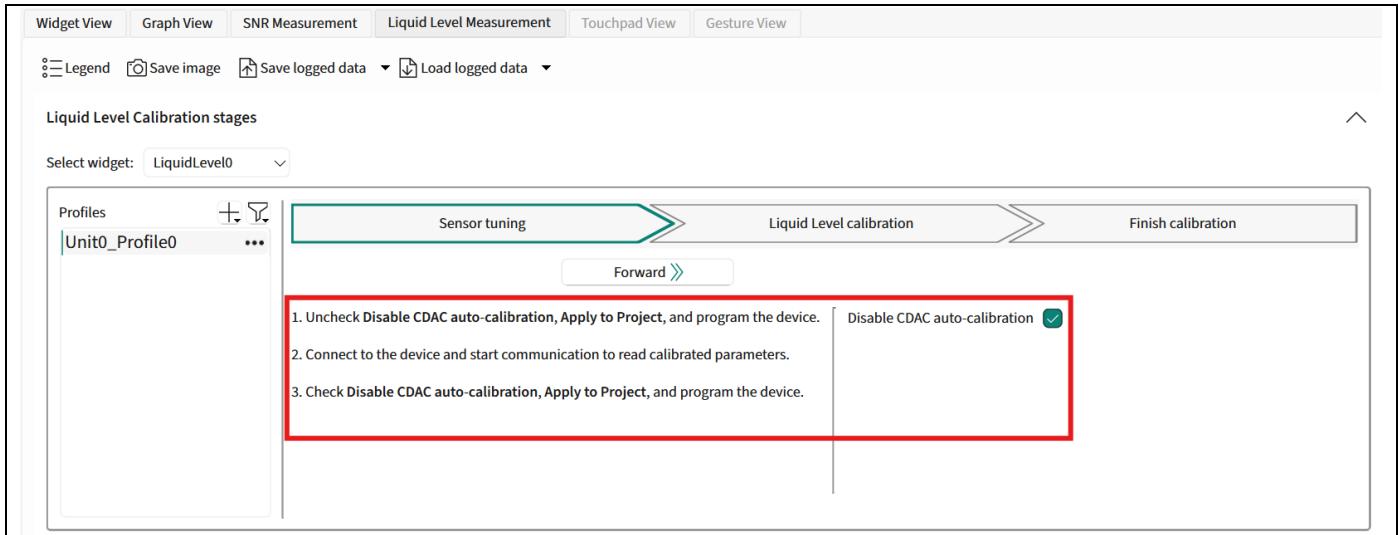
## Liquid-level factory calibration procedure

2. Navigate to the Liquid level measurement tab. Since the code example was calibrated with the foam rejection enabled, we will need to clear the results before beginning the calibration process.



**Figure 14** Clearing the results from previous calibration

3. With the tank empty, complete the process shown in the tool.



**Figure 15** Locking CDAC calibration parameters

Note: This process allows the device to calibrate the sensors to the empty tank, then locks those calibration parameters.

## Liquid-level factory calibration procedure

4. In the CAPSENSE™ Tuner, configure the system parameters:

- Tank Height:** The total height of the liquid to be measured in mm.
- Minimum resolution (height step):** The minimum change in liquid depth you would like to measure. Note that the amount of calibration data you need to collect is inversely proportional to this value. Lower minimum resolution values yield more data collection.
- Number of averaged samples per step:** The number of raw counts samples that are averaged to yield one data point while logging data. Higher values increase data logging time and accuracy. Note that if foam rejection is enabled, the minum resolution and number of averaged samples per step will only be configurable after completing the [initial calibration for foam rejection enabled sensors](#).

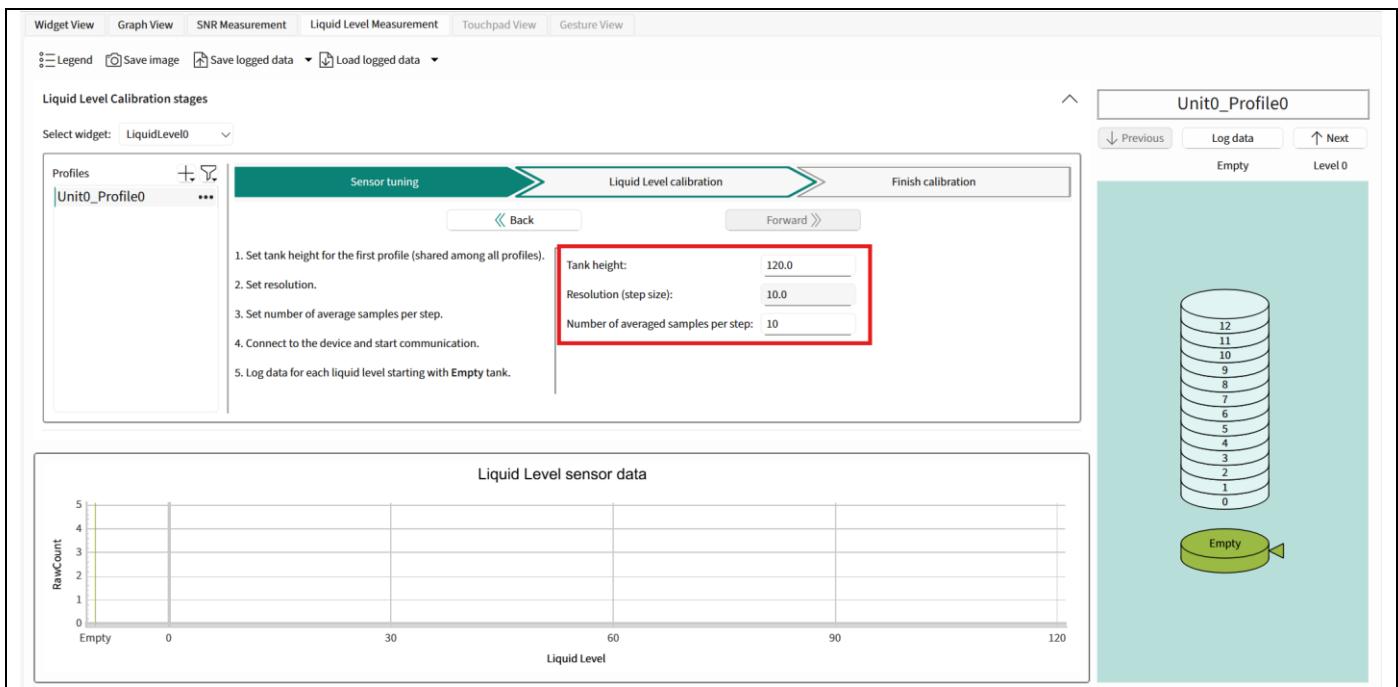
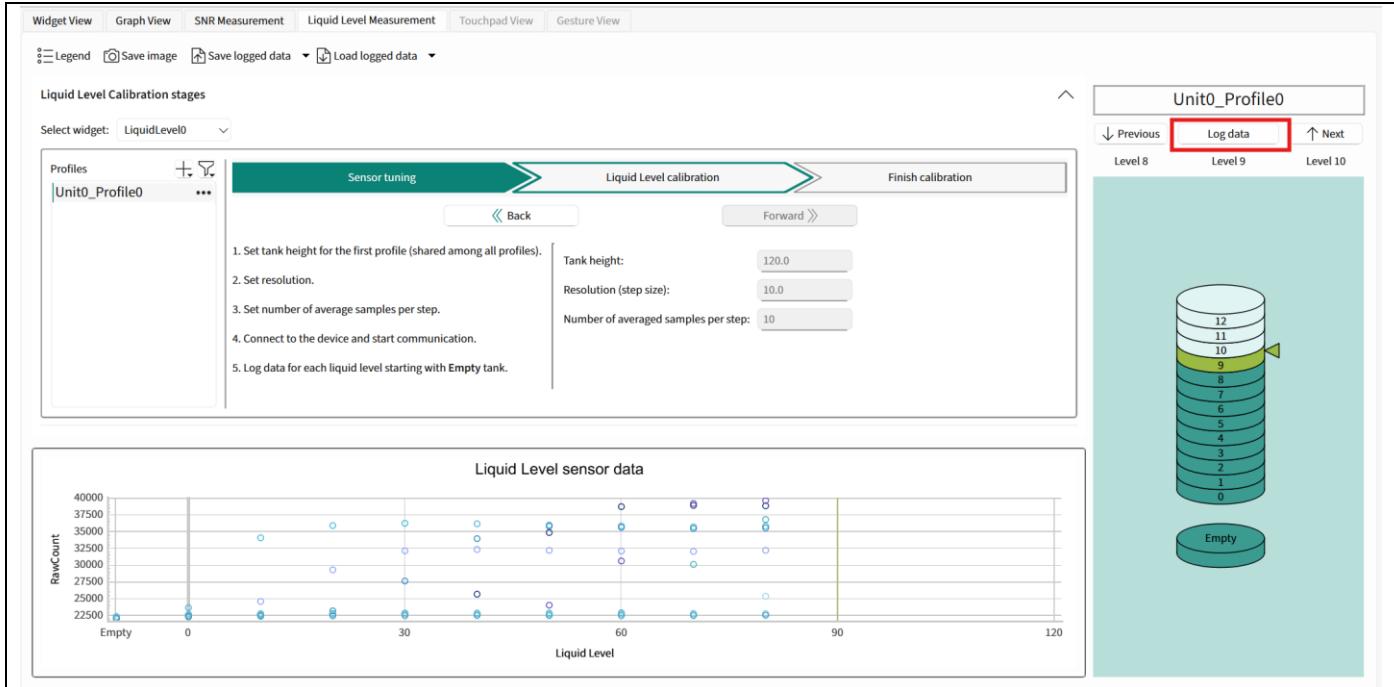


Figure 16 Disable the CDAC auto-calibration

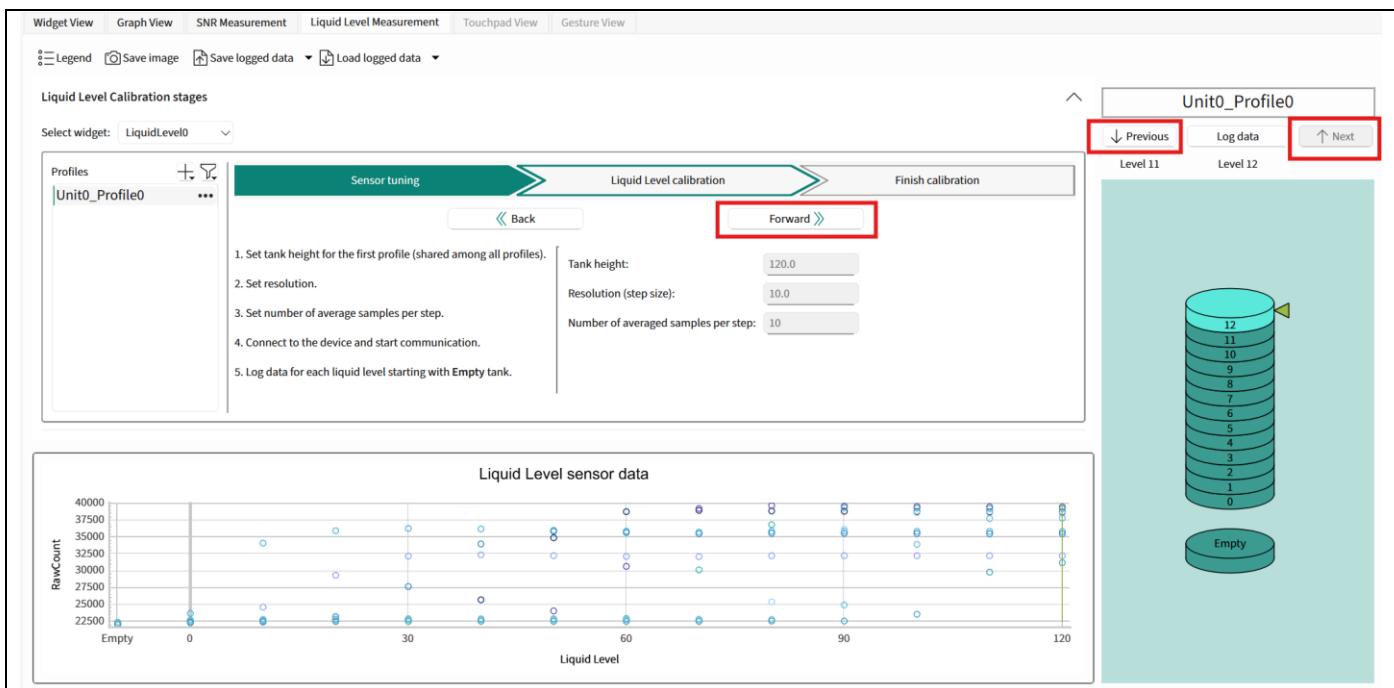
## Liquid-level factory calibration procedure

5. Incrementally fill the tank in steps defined by the resolution and log the data at each level. For each subsequent step, fill the tank by the level indicated by minimum resolution and log data.



**Figure 17** Tuner showing liquid level during calibration procedure

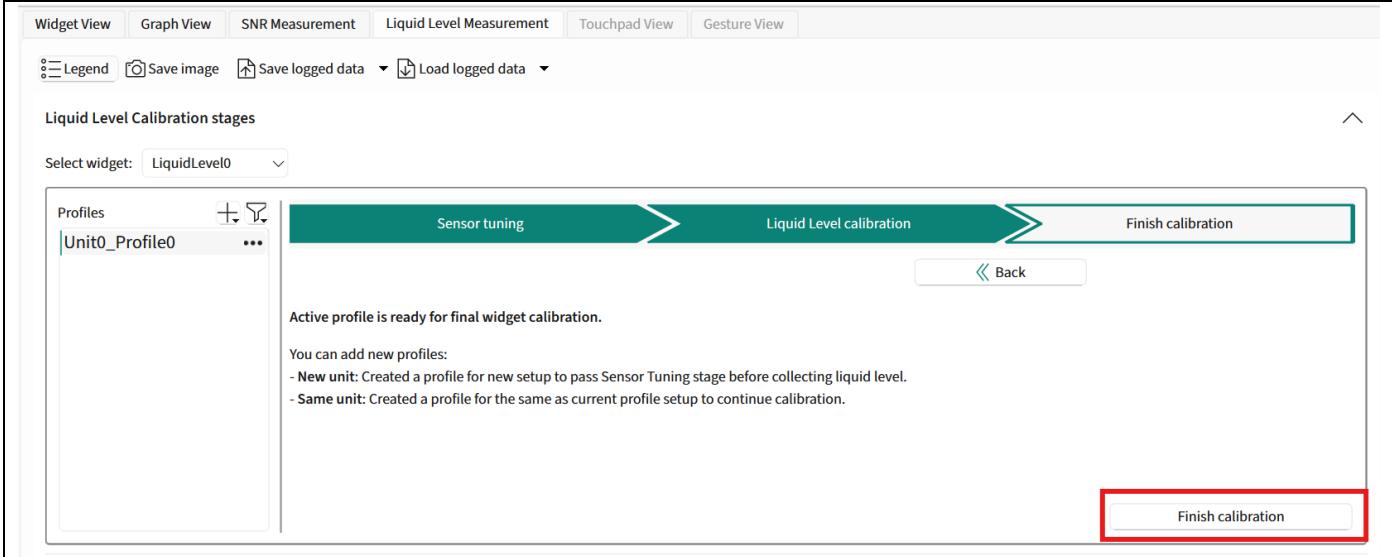
Note: You can navigate to any level by clicking previous/next level. Click ‘log data’ after filling to the next level every time.



**Figure 18** Navigating to previous or next levels

## Liquid-level factory calibration procedure

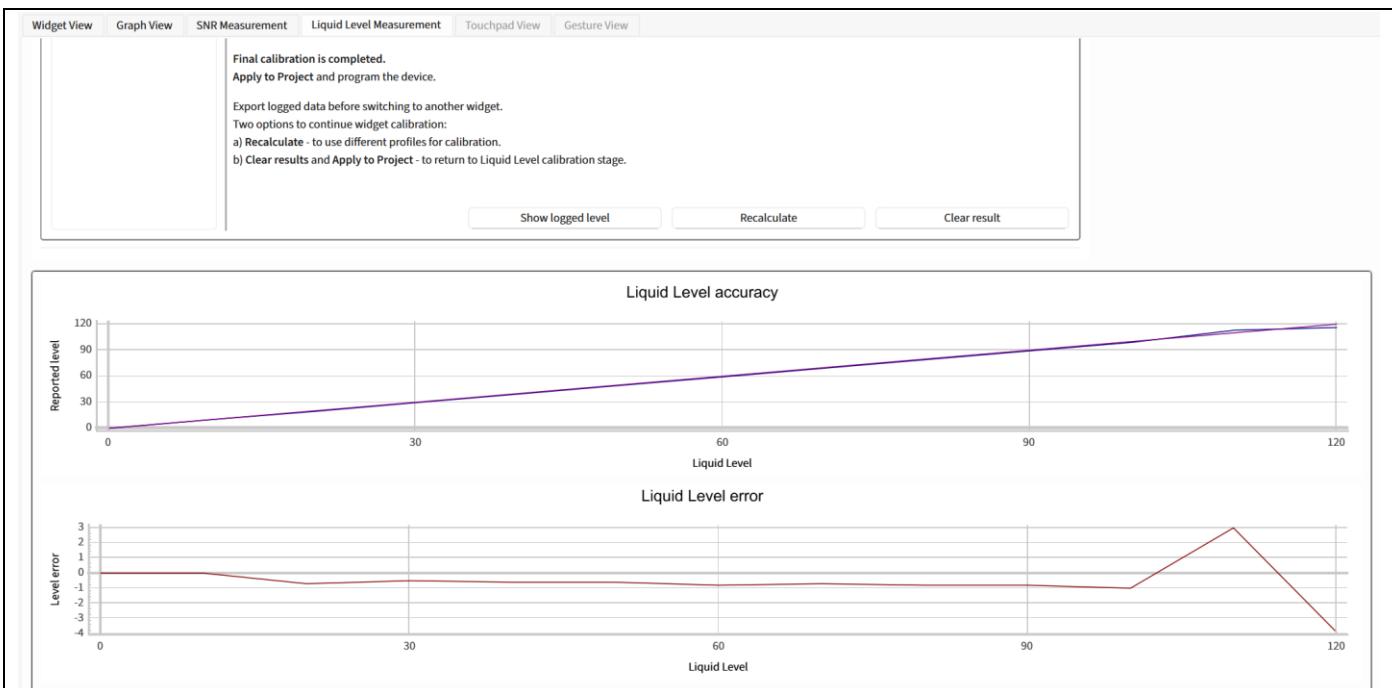
- Click Forward, then finish Calibration after logging the data.



**Figure 19** Finishing calibration

Note that at this step, multiple units and multiple profiles can be created. This allows for the calibration of the same sensor hardware to different scenarios or for using multiple units to calibrate for the same scenario, making the data set more robust.

- Once the calibration is complete you can see the accuracy and error plots.



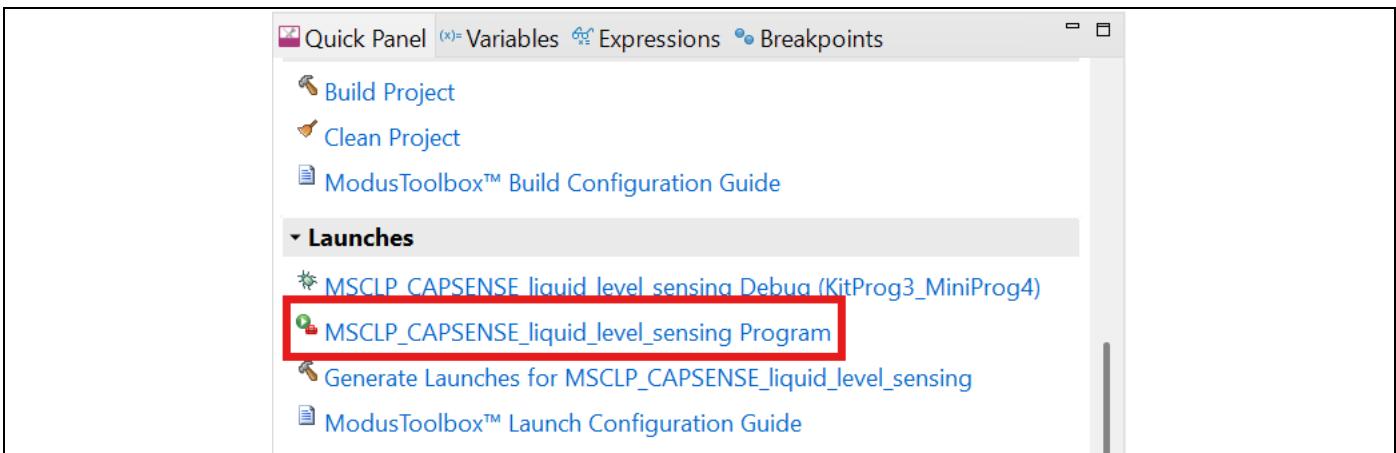
**Figure 20** Linearity plots

- Select Apply to Project, then disconnect from the tuner, and program the part to apply the calibration to the device.

## Liquid-level factory calibration procedure

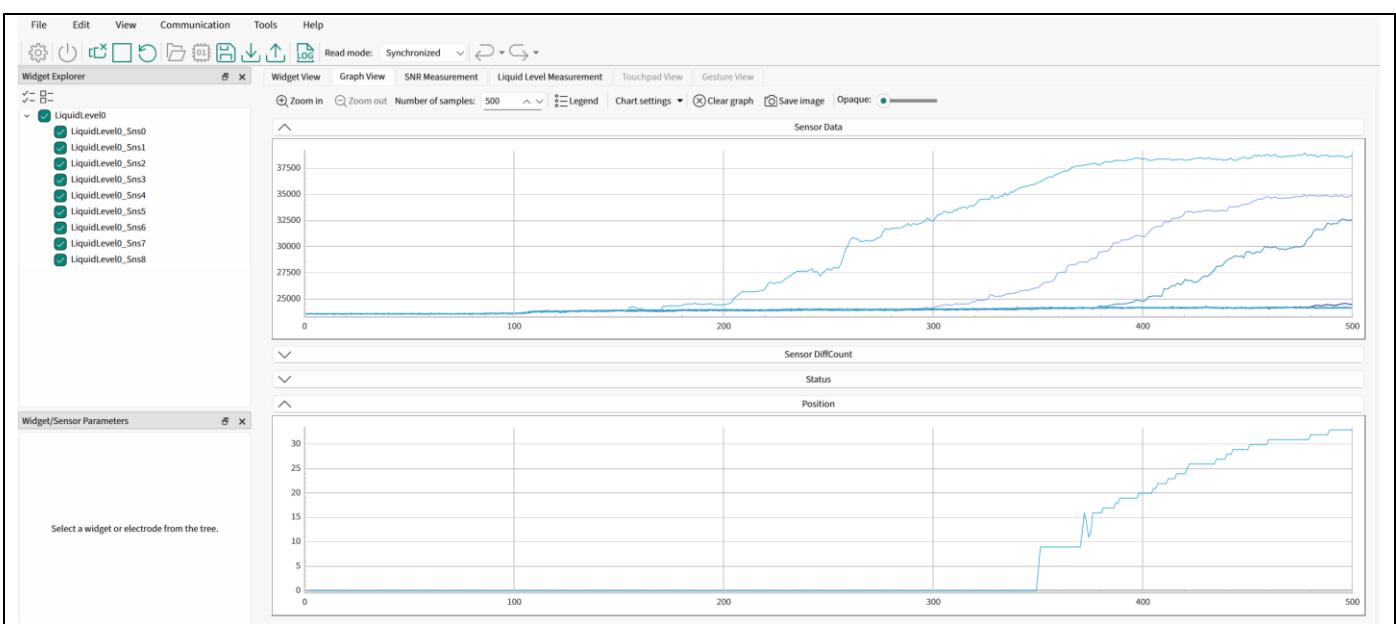


**Figure 21 Applying the settings to the project and disconnecting the tuner**



**Figure 22 Programming the device**

9. Empty the tank.
10. Reconnect to the tuner and navigate to the Graph View tab.

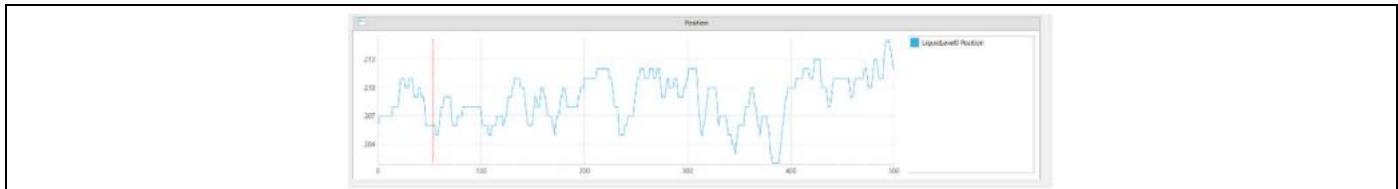


**Figure 23 Showing the Gragh view after connecting**

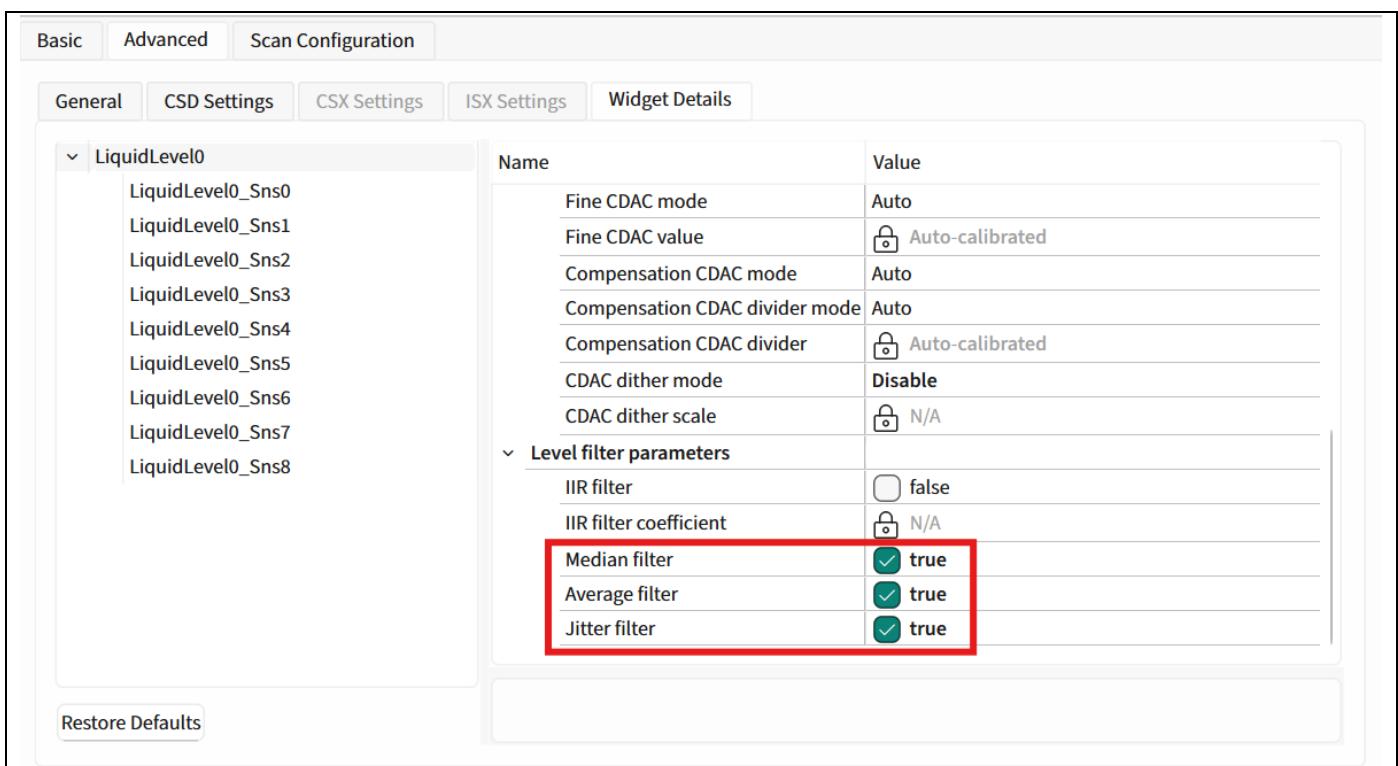
11. Observe the liquid level displayed in the Position graph.

## Liquid-level factory calibration procedure

12. If readings fluctuate, apply digital filtering methods such as IIR, median, averaging, or jitter filters in CAPSENSE™ Configurator for improved stability.



**Figure 24 Fluctuating liquid level position**



**Figure 25 Enabling filters**

**IIR Filter:** Programmable filter that behaves as a lowpass filter. Lower coefficient values have a lower cutoff-frequency and reduce response time.

**Median filter:** Averages three samples to remove noise spikes.

**Average filter:** Averages two samples to remove periodic noise.

**Jitter filter:** Eliminates toggling noise.

## Liquid-level factory calibration procedure

### 3.6 Output



**Figure 26 LLS tank connected to the device**

Fill the tank with a known amount and check if the position closely matches the expected value. The expected position value is defined by the Equation below.

$$\text{Position} = \text{PositionMax} \times \frac{\text{Minimum resolution}}{\text{Tank height}} \times \text{LiquidLevelCurrent}$$

- PositionMax: The liquid-level sensor widget's maximum Level parameter.
- Minimum resolution: Minimum resolution set in the tuner.
- Tank height: Tank height value set in the tuner.
- LiquidLevelCurrent: The current level of liquid present in the container.

### 3.7 Conclusion

The implementation of liquid level sensing using Infineon's PSoC™ 4 CAPSENSE™ technology provides a reliable, accurate, and non-invasive liquid level detection methods. By utilizing capacitive sensing principles, the system effectively measures liquid levels through the walls of non-conductive containers, eliminating direct contact with the liquid. The manual insights on how the factors affect sensor performance, the importance of precise and reliable liquid detection.

Through proper calibration the sensing solution delivers high-resolution measurements with robustness under real-world operating conditions for water. The integration of ModusToolbox™ software tools further simplifies sensor configuration, tuning, and optimization, making the approach scalable and adaptable to a wide range of applications.

## Foam rejection calibration

# 4 Foam rejection calibration

## 4.1 Description

Foam on the liquid surface can cause false readings in capacitive liquid level. The foam rejection process compensates for this by detecting and filtering out the capacitance contribution of foam. During calibration, the system measures both the actual liquid level and the foam-affected readings, calculates a correction factor, and applies it in firmware. This ensures that the sensor accurately reports the true liquid level, even in the presence of bubbles or foam.

## 4.2 Enabling foam rejection

1. To re-enable foam rejection for the code example, open the CAPSENSE™ Configurator and navigate to the widget details.

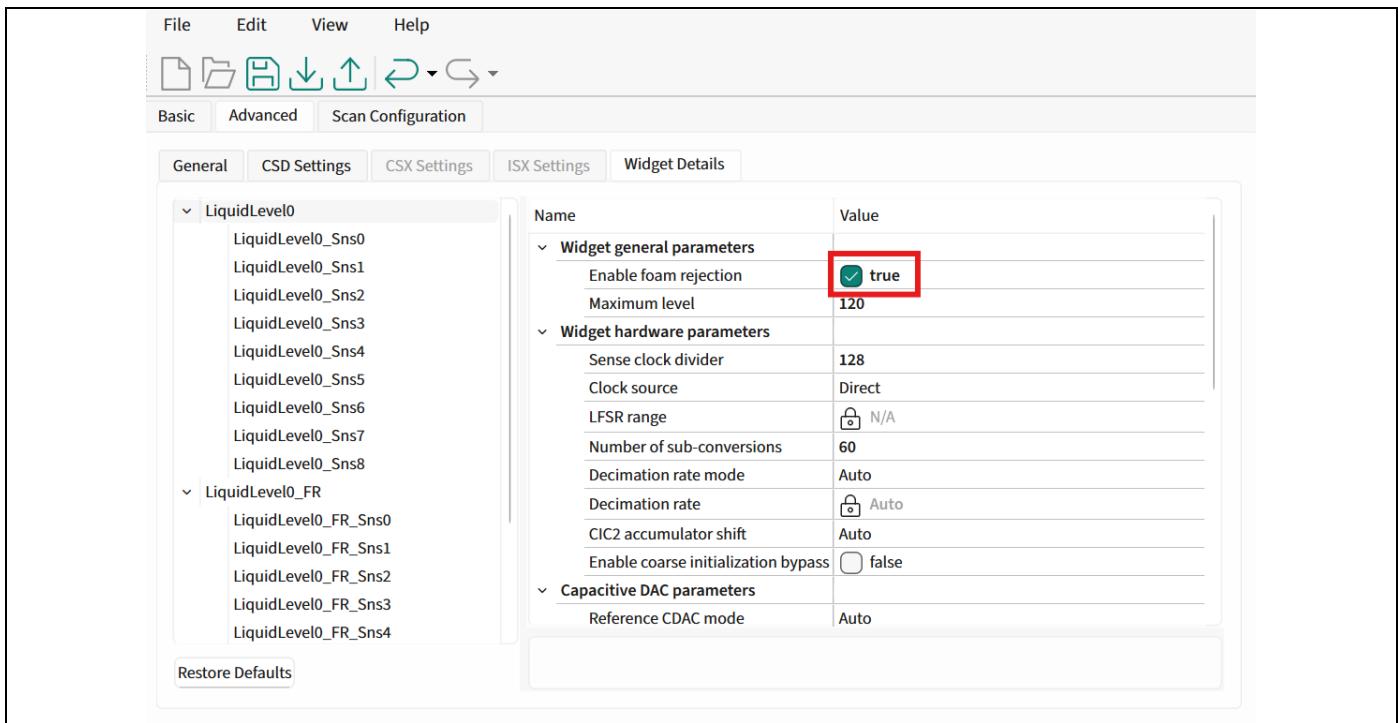


Figure 27 Enabling foam rejection

## Foam rejection calibration

2. In the Eclipse IDE, open the main.c file and add the lines associated with the foam rejection widget.

```
for (;;)
{
    uint32_t level_wo_FR, level_w_FR;

    /* Scan the normal Liquid Level Widget */
    Cy_CapSense_ScanWidget(CY_CAPSENSE_LIQUIDLEVEL0_WDGT_ID,
                           &cy_capsense_context);
    /* Wait until the scan is finished */
    while (Cy_CapSense_IsBusy(&cy_capsense_context))
    {
    }

    /* Scan the Foam Rejection Widget */
    Cy_CapSense_ScanWidget(CY_CAPSENSE_LIQUIDLEVEL0_FR_WDGT_ID,
                           &cy_capsense_context);
    /* Wait until the scan is finished */
    while (Cy_CapSense_IsBusy(&cy_capsense_context))
    {
    }

    /* Process all th widgets */
    Cy_CapSense_ProcessAllWidgets(&cy_capsense_context);

    /* Send capsense data to the Tuner */
    Cy_CapSense_RunTuner(&cy_capsense_context);

    /* store the liquid level before and after foam rejection */
    level_wo_FR = CY_CAPSENSE_LIQUIDLEVEL0_PTRPOSITION_VALUE->x;
    level_w_FR = CY_CAPSENSE_LIQUIDLEVEL0_FR_PTRPOSITION_VALUE->x;

    /* keep the compiler happy */
    (void) level_wo_FR;
    (void) level_w_FR;
}
```

**Figure 28 Add code for the foam rejection widget**

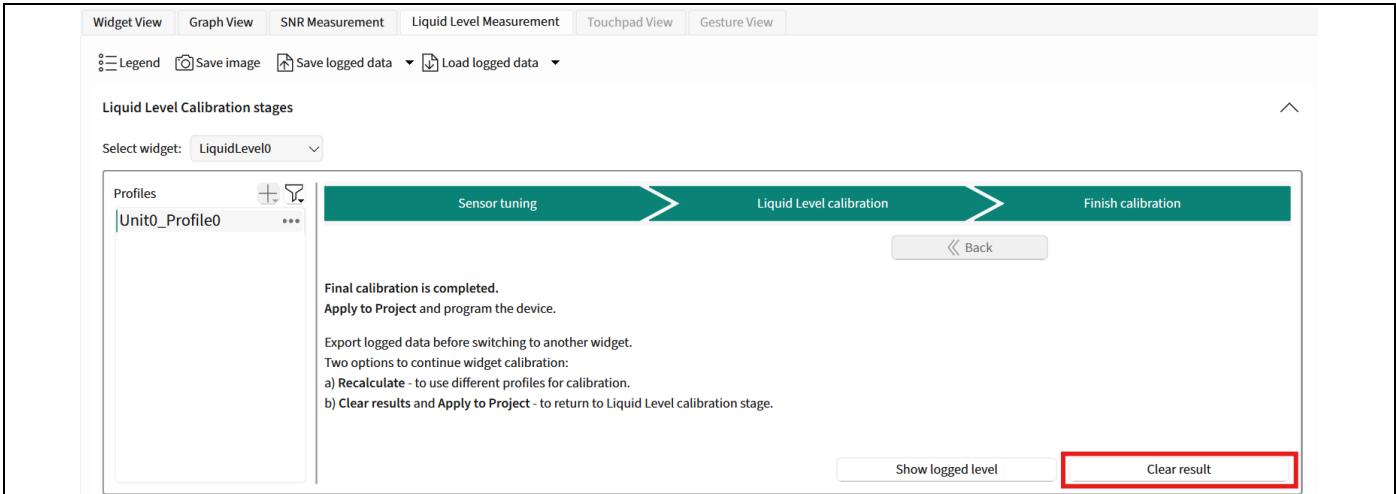
3. Program the device.

## Foam rejection calibration

### 4.3 Initial calibration for foam rejection enabled sensors

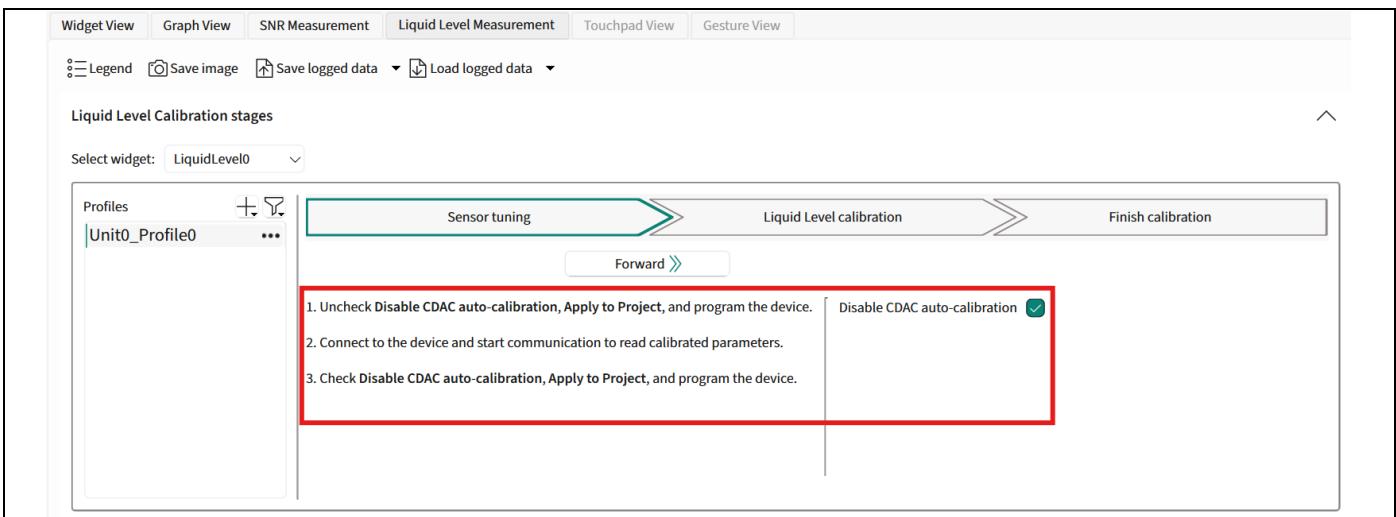
Before calibrating, ensure that the liquidlevel sensors are properly tuned following the guidance in the “Tuning Procedure” section of PSOC™ 4: [MSCLP CAPSENSE™ liquid-level sensing code example](#)

1. If the liquid level calibration was completed with foam rejection disabled, the results will need to be cleared from the **Liquid Level Measurement** tab.



**Figure 29** Clearing results

2. With the tank empty, complete the process shown in the tool.

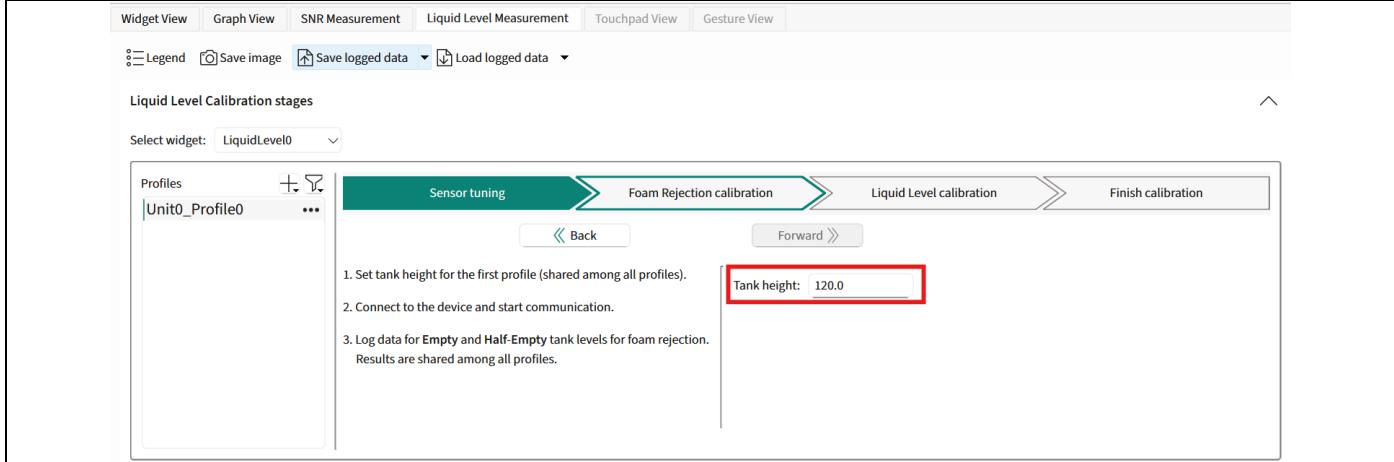


**Figure 30** Locking CDAC calibration parameters

Note: This process allows the device to calibrate the sensors to the empty tank, then locks those calibration parameters.

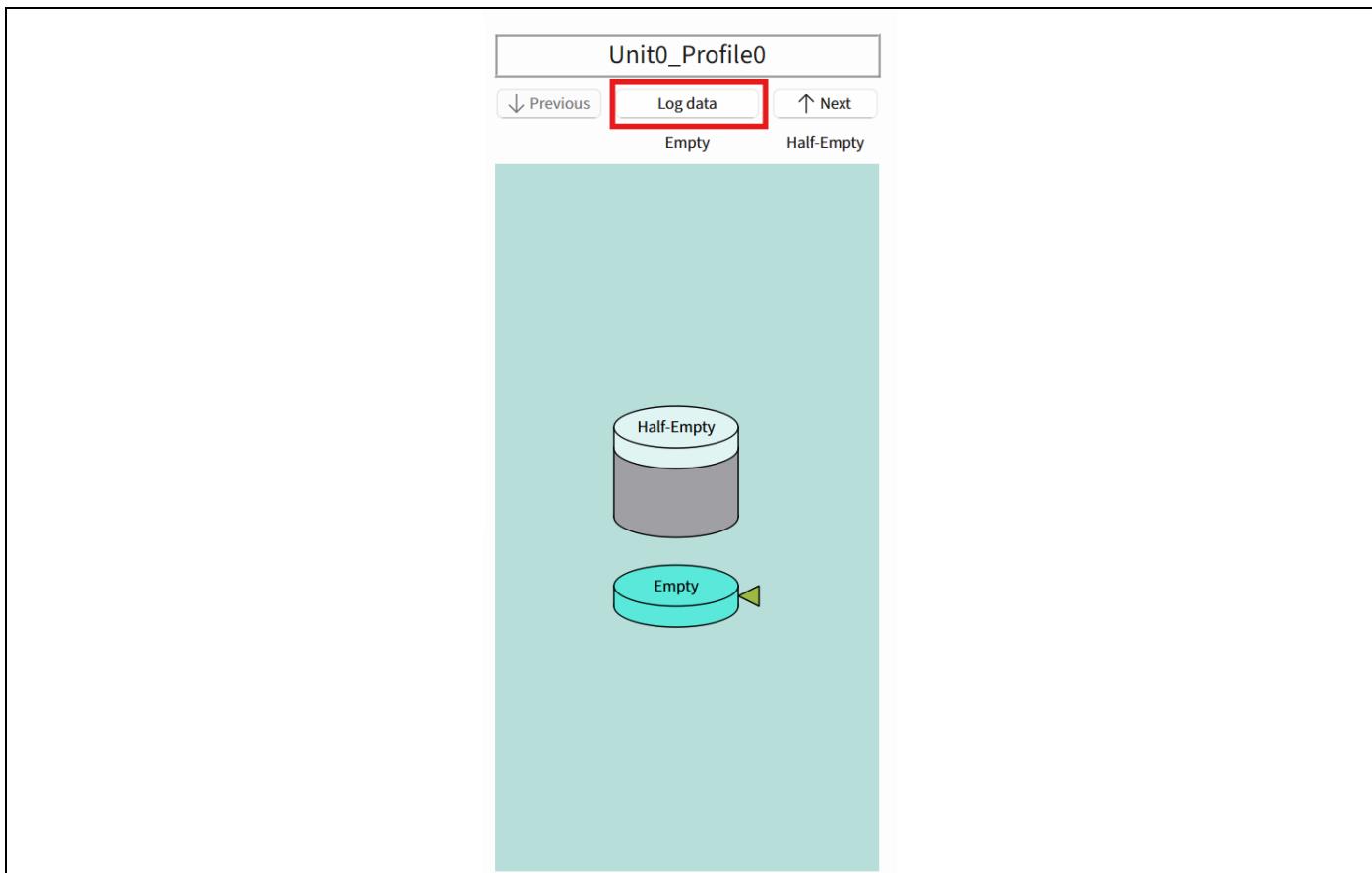
## Foam rejection calibration

- Set the tank height.



**Figure 31** Setting the tank height

- With the tank empty, log the empty data.



**Figure 32** Empty tank calibration for foam rejection

- Fill the tank halfway, ensure that the step in the tool is halfway, then log the data.
- Press the **Forward** button.
- At this point, the [liquid level calibration procedure](#) must be completed.

## Foam rejection calibration

- When the foam rejection feature is enabled, the charts will show accuracy and error plots for the liquid level and foam rejection widgets.

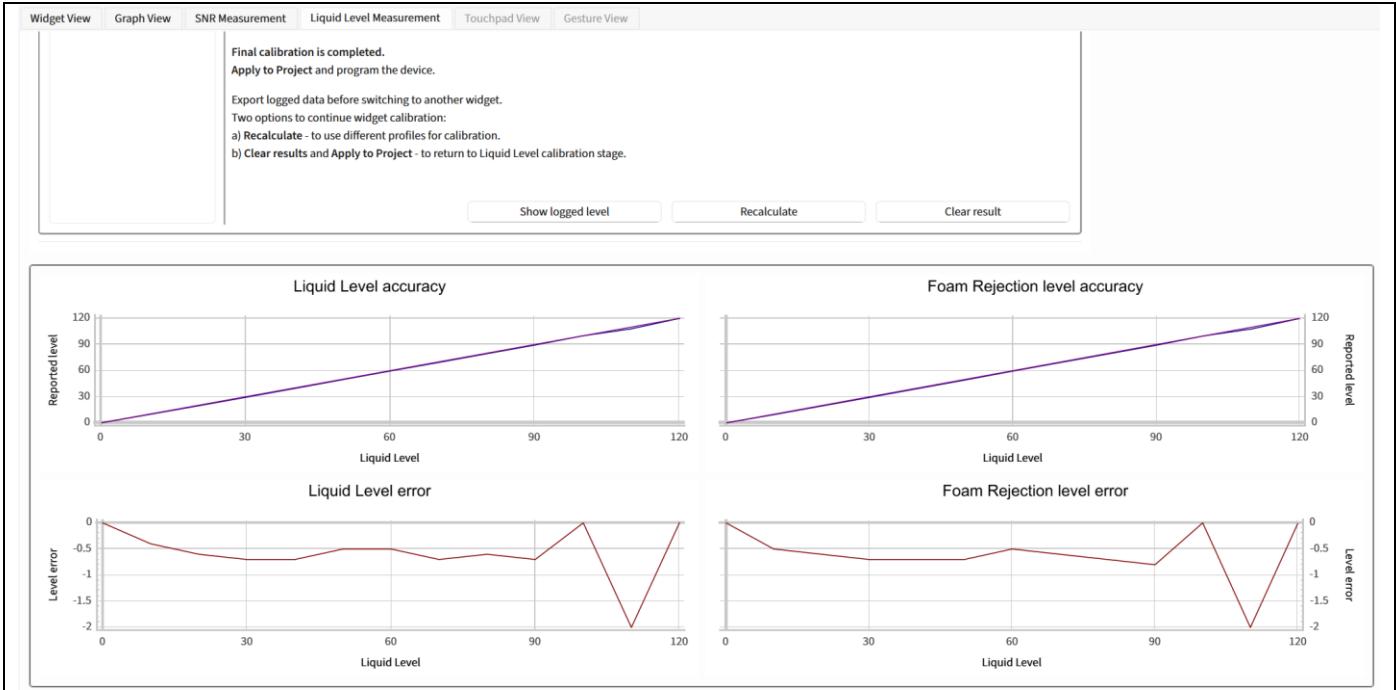


Figure 33 Linearity plots for liquid level and foam rejection widgets

## Foam rejection calibration

### 4.4 Foam rejection calibration procedure

The foam rejection widget contains an additional parameter – the foam correction coefficient that determines how much compensation is applied to the level detection to remove signal from the foam. The foam correction coefficient is given by the following equation:

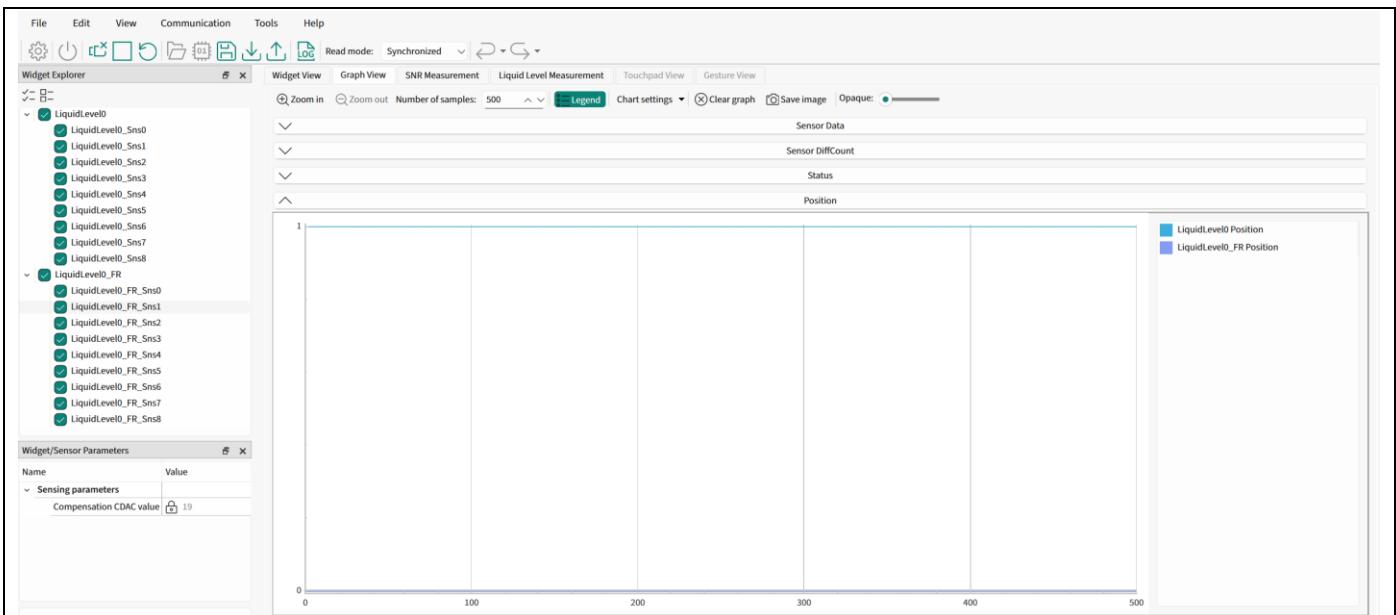
$$\text{Coeff}_{\text{foam}} = 256 \times \frac{\text{Level}_{\text{Foam}} - \text{Level}_{\text{known}}}{\text{Level}_{\text{liquid}} - \text{Level}_{\text{foam}}}$$

**Figure 34** Foam correction coefficient equation

- $\text{Level}_{\text{Foam}}$  = The level displayed by the CAPSENSE™ Tuner for the foam rejection widget
- $\text{Level}_{\text{Known}}$  = The level of the target liquid known by measurement with an external tool
- $\text{Level}_{\text{Liquid}}$  = The level displayed by the CAPSENSE™ Tuner for the base liquid-level widget

To calculate the foam correction coefficient, ensure that you have correctly followed the process detailed in [liquid level calibration procedure](#). Then, perform the following steps:

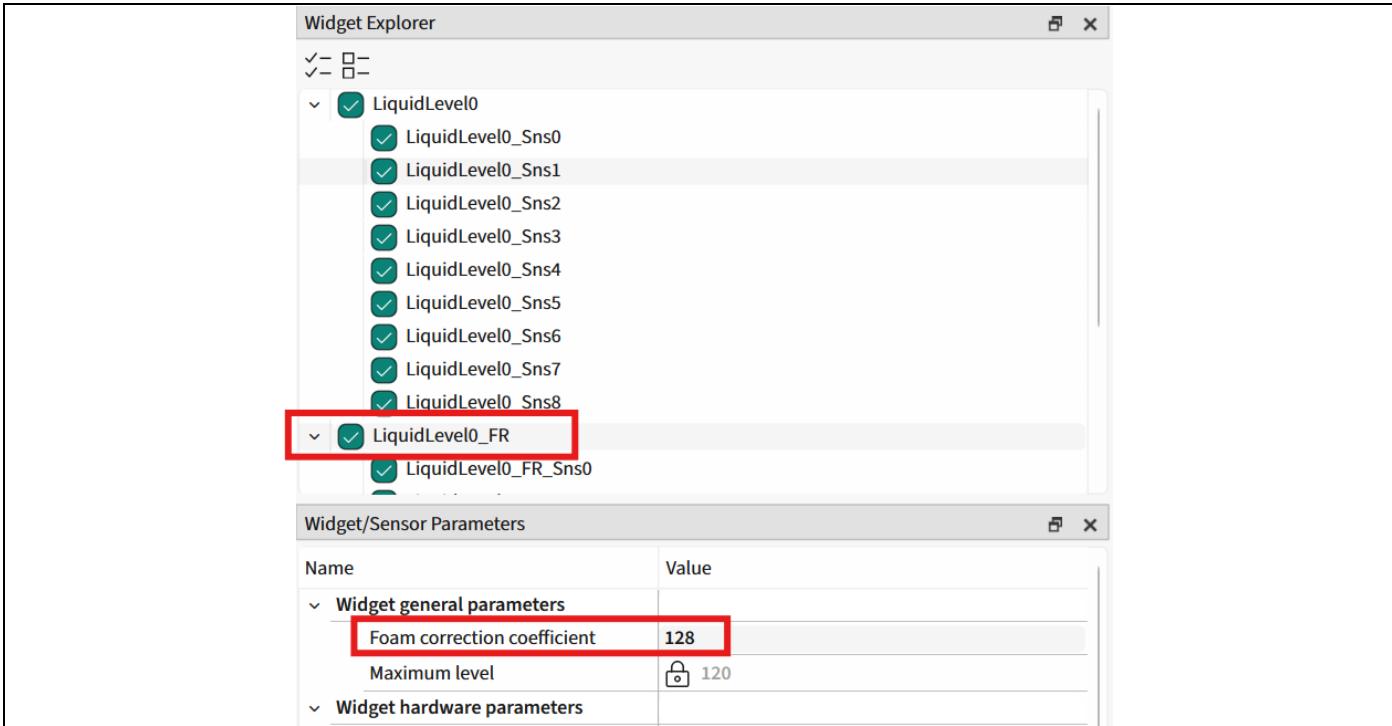
1. Fill the tank to a known liquid level and generate foam on the surface using a surfactant.
2. Enable both liquid level and foam rejection widgets in the CAPSENSE™ Tuner.



**Figure 35** Tuner data showing the position of the liquid level base and foam rejection widgets

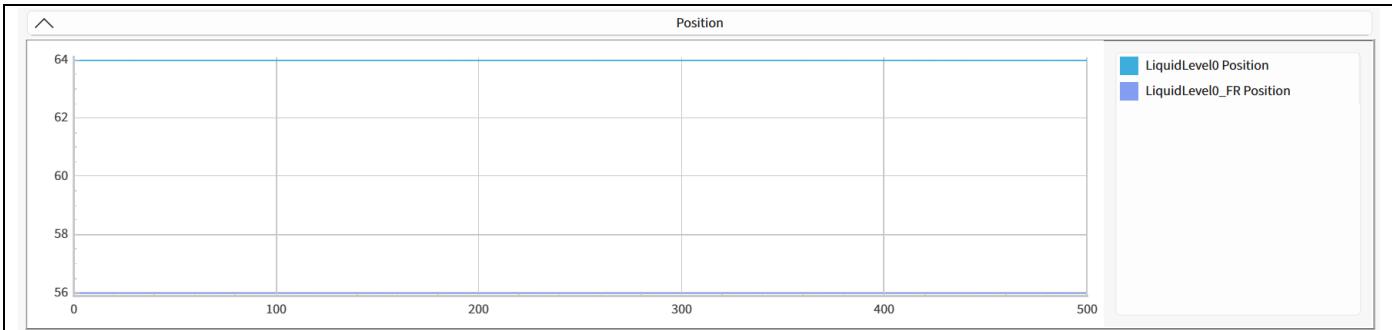
3. Record sensor outputs ( $\text{Level}_{\text{Foam}}$ ,  $\text{Level}_{\text{Known}}$ , and  $\text{Lev}_{\text{Liquid}}$ ).
4. Calculate the foam correction coefficient to adjust for measurement deviation caused by foam.
  - a. For example: If the  $\text{Level}_{\text{Liquid}}$  is 76, the  $\text{Level}_{\text{Foam}}$  is 72, and the  $\text{Level}_{\text{Known}}$  is 76. According to the equation, the foam correction coefficient will be 128.
5. Apply the correction factor in firmware and validate readings to ensure that the output matches the actual liquid level.

## Foam rejection calibration



**Figure 36 Applying the foam correction coefficient**

## 4.5 Output



**Figure 37 Liquid level and foam rejection level**

## 4.6 Conclusion

The foam rejection process ensures accurate liquid level measurements by compensating for the capacitance effect of foam. By applying a correction factor during calibration, the system reliably distinguishes between actual liquid and foam, maintaining precise readings under real-world conditions.

This user manual guides on how to set up, calibrate, and operate a Liquid Level Sensing (LLS) system, gaining hands-on experience in real-time monitoring, accurate measurement of liquid levels, and understanding the principles behind the sensor's operation and response. Also, the benefit of using this manual is that it will guide step-by-step procedures in detail so that the user understands the significance of every step while performing them.

## References

### References

- [1] Infineon Technologies AG : AN239805: Liquid-level sensing with PSOC™ 4 CAPSENSE™ ;  
[Available online](#)

## Glossary

### Glossary

#### CSD

CAPSENSE™ Sigma Delta is a patented method of performing self-capacitance (also called self-cap) measurements for capacitive sensing applications. In CSD mode, the sensing system measures the self-capacitance of an electrode, and a change in the self capacitance is detected to identify the presence or absence of a finger.

#### MSCLP

Multi sense converter with Low-Power: The multi sense converter is the analog to digital converter used in Fifth-Generation CAPSENSE™ technology also known as Ratiometric sensing technology.

#### SNR

Signal-to-noise ratio: The ratio of the sensor signal, when touched, to the noise signal of an untouched sensor.

#### Widget

A user-interface element in the CAPSENSE™ Component that consists of one sensor or a group of similar sensors. Button, proximity sensor, linear slider, radial slider, matrix buttons, and touchpad are the supported widgets.

## Revision history

### Revision history

Document revision	Date	Description of changes
**	2025-12-12	Initial document

## **Trademarks**

All referenced product or service names and trademarks are the property of their respective owners.

**Edition 2025-12-12**

**Published by**

**Infineon Technologies AG  
81726 Munich, Germany**

**© 2025 Infineon Technologies AG.  
All Rights Reserved.**

**Do you have a question about this  
document?**

**Email:**

[erratum@infineon.com](mailto:erratum@infineon.com)

**Document reference**

**User guide number**

## **Important notice**

The information given in this document shall in no event be regarded as a guarantee of conditions or characteristics ("Beschaffenheitsgarantie").

With respect to any examples, hints or any typical values stated herein and/or any information regarding the application of the product, Infineon Technologies hereby disclaims any and all warranties and liabilities of any kind, including without limitation warranties of non-infringement of intellectual property rights of any third party.

In addition, any information given in this document is subject to customer's compliance with its obligations stated in this document and any applicable legal requirements, norms and standards concerning customer's products and any use of the product of Infineon Technologies in customer's applications.

The data contained in this document is exclusively intended for technically trained staff. It is the responsibility of customer's technical departments to evaluate the suitability of the product for the intended application and the completeness of the product information given in this document with respect to such application.

## **Warnings**

Due to technical requirements products may contain dangerous substances. For information on the types in question please contact your nearest Infineon Technologies office.

Except as otherwise explicitly approved by Infineon Technologies in a written document signed by authorized representatives of Infineon Technologies, Infineon Technologies' products may not be used in any applications where a failure of the product or any consequences of the use thereof can reasonably be expected to result in personal injury.