PSOC™ Automotive 4: Capacitive liquid-level sensing

About this document

Scope and purpose

This document shows how to use PSOC™ Automotive 4 devices, CAPSENSE™ technology, and capacitive sensors to measure the depth or presence of water-based liquids in non-conductive containers. Sensors located on or near the container exterior provide real-time reporting of liquid height or percent remaining without liquid contact. Options exist to use low-cost sensor materials while still providing high-precision measurements.

Intended audience

This application note is intended for engineers who wants to work on capacitive-based liquid-level sensing and those with experience in working with PSOC™ microcontroller and ModusToolbox™ IDE.

Table of contents

About this document . . . 1
Table of contents . . . 2
1 Introduction . . . 3
1.1 Beginner's resources . . . 3
2 System overview . . . 4
2.1 Block diagram . . . 4
2.2 Hardware requirements . . . 4
3 System description . . . 6
3.1 CAPSENSE™ basics . . . 6
3.2 Environmental effect . . . 7
3.2.1 Temperature . . . 7
3.2.2 Parasitic capacitance . . . 8
3.2.3 Mechanical variations . . . 8
3.3 CAPSENSE™ Component setup . . . 8
3.4 Sensor pattern choice . . . 10
3.5 Segmented sensors . . . 10
3.5.1 General operation . . . 11
3.5.2 Liquid-level calculations . . . 12
3.5.3 Accuracy . . . 13
3.6 Differential sensors . . . 14
3.6.1 General operation . . . 15
3.6.2 Liquid-level calculations . . . 17
3.6.3 Accuracy . . . 18
4 Sensor layout . . . 19
4.1 General layout considerations . . . 19
4.2 Segmented sensor pattern considerations . . . 20
4.3 Differential sensor pattern consideration . . . 20
5 Code implementation with PSOC™ 4100S Max . . . 21
6 Summary . . . 24
References . . . 25
Glossary . . . 26
Revision history . . . 27
Trademarks . . . 28
Disclaimer . . . 29

1 Introduction

Liquid-level sensing (LLS) detects the presence and level of liquid in a container without any physical contact. There are various types of liquid-level sensors, such as capacitive, mechanical float, inductive, magnetic, Hall effect, optical, acoustic density, and ultrasonic; each has their own advantages and disadvantages. Capacitive liquid-level sensing has become popular because of its low cost, high reliability, low power, sleek aesthetics, and seamless integration with existing control architectures.

Infineon's PSOC™ Automotive 4 devices support liquid-level sensing with resolution down to 1 mm. Capacitive liquid-level sensing is provided through the use of the CAPSENSE™ Middleware available in the ModusToolbox™ Integrated Development Environment (IDE). The CAPSENSE™ Middleware APIs can be used to configure the on-chip CAPSENSE™ peripheral hardware and its operation on PSOC™ Automotive 4 devices. The following are the key liquid-level sensing benefits provided using CAPSENSE™:

- Non-contact measurement avoids contamination and cleaning problems
- Sensors located on the exterior of a non-conductive liquid container simplify industrial design and improve product user experience
- Optimized resolution and accuracy to support varying price points with a single, low-cost, base system
- Sensors may be constructed out of low-cost materials, such as plastic substrates and conductive ink

This application note provides details on the design of hardware, firmware, schematics, and bill of materials (BOM). See Code implementation with PSOC™ 4100S Max for the code snippet demonstrating the concepts of this application note. The code snippet provided in Code implementation with PSOC™ 4100S Max uses the combination of Arduino - compatible CY8CKIT-041S PSOC™ 4100S Max Pioneer Kit and CY8CKIT-022 Liquid Level Sensing Shield for this sensing purpose.

This application note is intended for advanced users of Infineon's CAPSENSE™ technology. If you are new to CAPSENSE™ technology, see the AN64846 – Getting started with CAPSENSE™ design guide. To learn about the implementation of CAPSENSE™ technology on PSOC™ 4 devices, see the AN85951 – PSOC™ 4 CAPSENSE™ design guide.

This application note focuses on areas that differ from traditional finger-touch applications using CAPSENSE™. Note that no two end designs are the same and that the design rules provided do not have hard limits unless specifically stated. Taking one or more design parameters near or past the advised limits may still work with reduced but acceptable performance, while the same performance may be unsuitable for another application. Use the provided design recommendations as guidance and perform validation under your design conditions.

1.1 Beginner's resources

See PSOC™ MCU webpage for an overview of PSOC™ devices. The web page includes a list of PSOC™ device families, IDEs, and associated development kits. In addition, see the following documents to get started with PSOC™ 4 devices and CAPSENSE™ technology:

- AN79953 – Getting started with PSOC™ 4 MCU
- AN64846 – Getting started with CAPSENSE™
- AN85951 – PSOC™ 4 CAPSENSE™ design guide
- PSOC™ 101 training series

2 System overview

A capacitive liquid-level sensing system comprises two key design elements:

- Capacitive sensor pattern to sense the liquid level
- PSOC™ 4 device with the CAPSENSE™ Component to measure the sensors and calculate the liquid level

This section provides a high-level overview of how they integrate into a larger system. Both design elements are described in detail in System description section.

2.1 Block diagram

Figure 1 shows the block diagram of the CY8CKIT-022 Liquid Level Sensing Shield showing the simplicity of a PSOC™ Automotive 4-based liquid-level sensing design. Most designs require only a PSOC™ Automotive 4 device and a CAPSENSE™ liquid sensor. Other system features can be integrated into the PSOC™ Automotive 4 device to further reduce system costs.

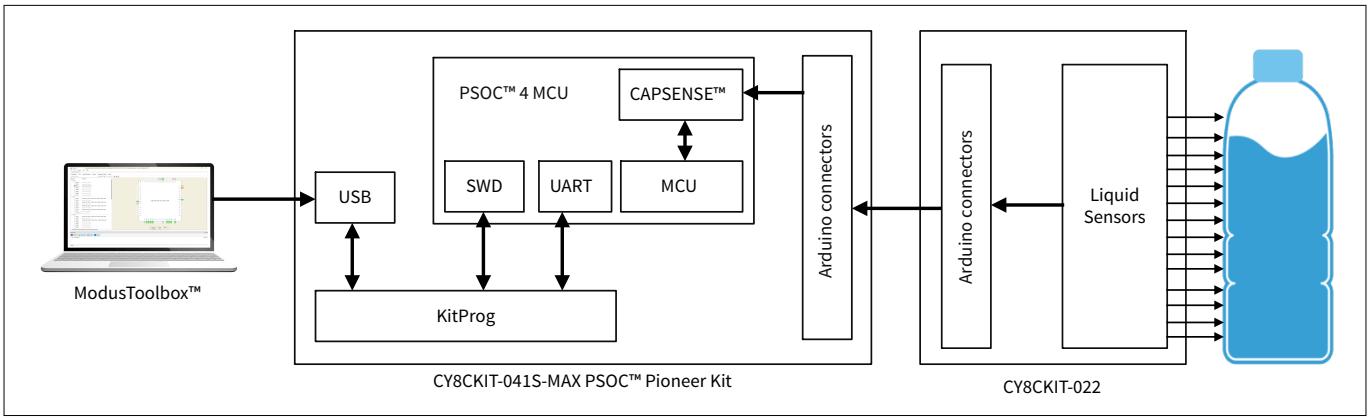


Figure 1 Liquid-level sensing block diagram

2.2 Hardware requirements

To implement CAPSENSE™ liquid-level sensing hardware, you need a PSOC™ Automotive 4 device and liquid-level sensors attached to a liquid container. Develop them as detailed in this application note or use those provided in the CY8CKIT-022 shield for an out-of-the-box, ready-to-run solution.

The CY8CKIT-022 kit plugs into the CY8CKIT-041S PSOC™ 4100-SMAX Pioneer Kit development platform from Infineon. In Code implementation with PSOC™ 4100S Max section, you can see the PSOC™ 4100S-MAX code and algorithm, which provides liquid-level sensing targeting the two kits. Figure 2 shows how the CY8CKIT-022 shield connects to the CY8CKIT-041S MAX kit.

2 System overview



Figure 2 CY8CKIT-022 shield connected to CY8CKIT-041S-MAX kit

3 System description

This section provides a high-level description of the principle of capacitive sensing. For more CAPSENSE™ operational and design details, see the AN64846 – Getting started with CAPSENSE™ and the AN85951 – PSOC™ 4 CAPSENSE™ design guide.

3.1 CAPSENSE™ basics

Capacitive liquid-level sensors are conductive pads or traces laid on a non-conductive material, such as PCB, plastic, or glass. The intrinsic capacitance of the PCB trace, pads, and other sensor connections is called the sensor parasitic capacitance (C_P), as Figure 3 shows.

When a target object such as water approaches the sensor, a small amount of liquid capacitance (C_L) is added to the C_P, as Figure 4 shows. Liquid-level sensing involves measuring the increased capacitance when water is near the sensor.

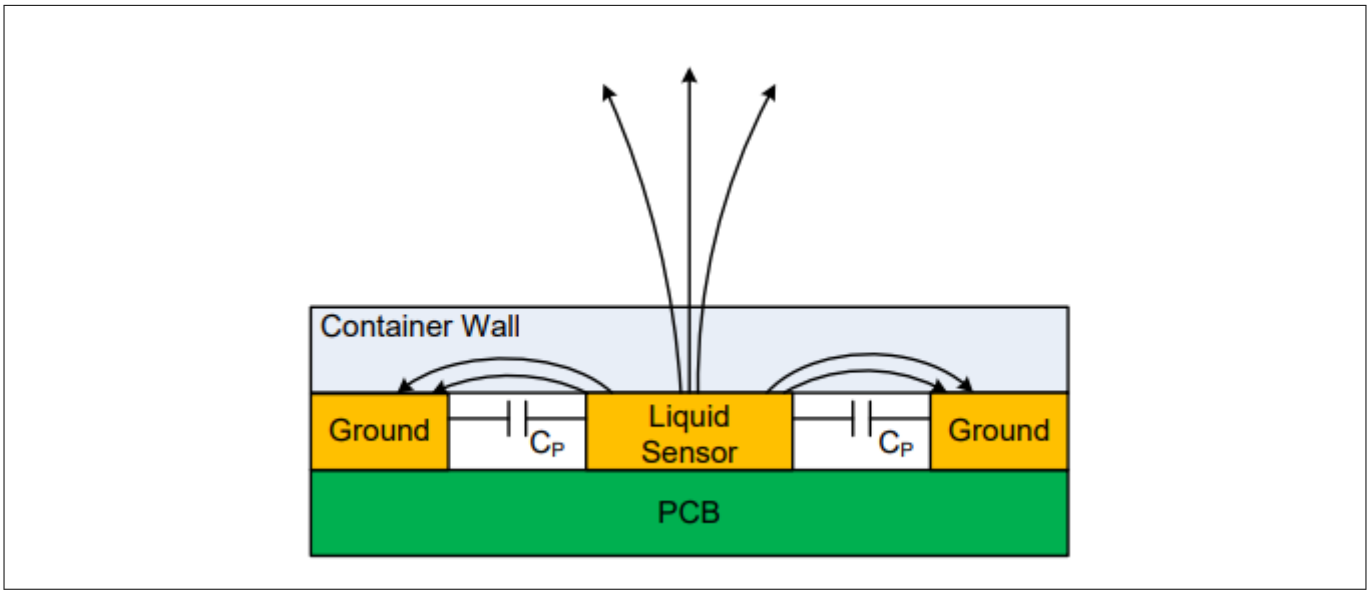


Figure 3 Capacitance and electric field of a capacitive liquid-level sensor

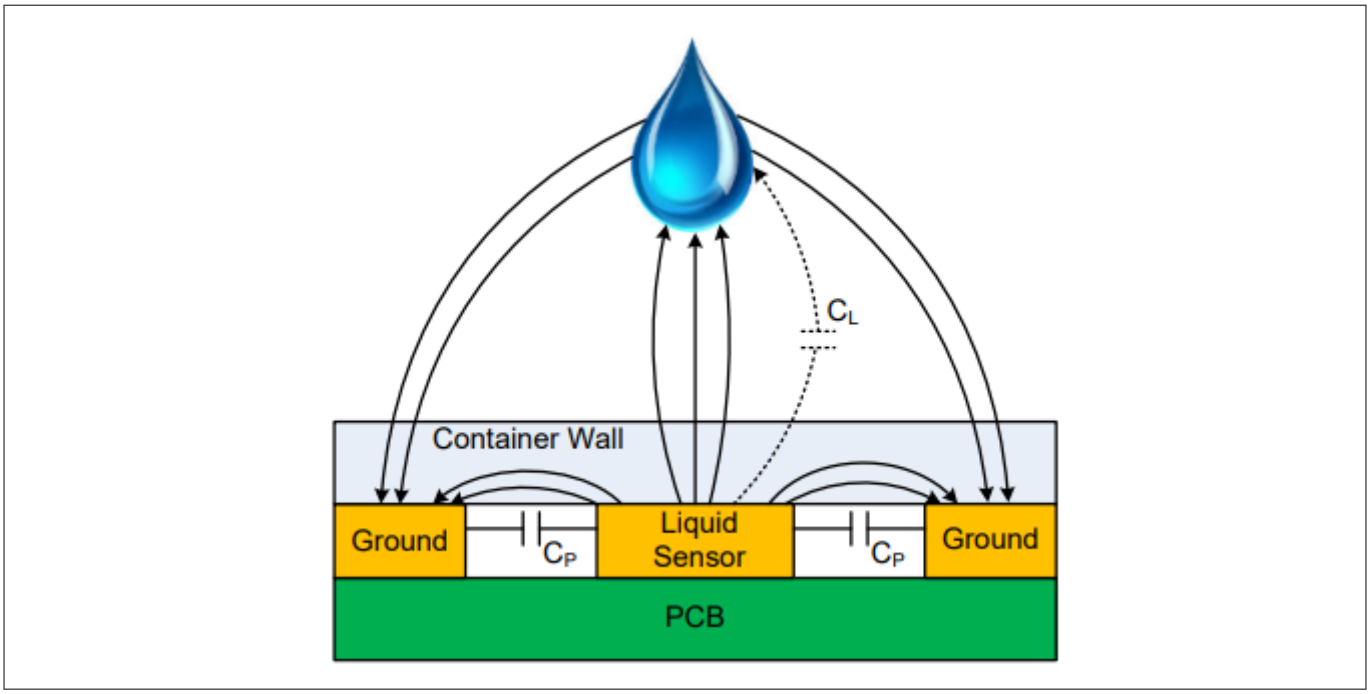


Figure 4 Added capacitance (C_L) when a liquid approaches a capacitive sensor

The CAPSENSE™ peripheral located inside the PSOC™ 4 MCU controller measures the capacitance by injecting a current into the sensor with a current digital-to-analog converter (IDAC). A timer measures how long it takes the IDAC to charge the sensor's voltage to a reference voltage using a comparator. When the conversion is complete, the timer count value that measured the IDAC charge time is used as the raw sensor value used in calculations and is commonly referred to as the sensor rawcount.

3.2 Environmental effect

There are several important factors that can affect both the resolution and accuracy of liquid-level sensing. These effects are caused by variations in the sensor capacitance resulting in inaccurate measurements.

3.2.1 Temperature

Temperature fluctuations during operation have the most significant impact on performance. With finger-activated CAPSENSE™ buttons (as compared to liquid-level sensing), the sensor value while not pressed is tracked over time to account for any ambient temperature-induced offset. This compensated non-pressed value is called the "baseline". In touch-sensing applications, this is possible because most of the time, the sensor is not touched. Additionally, the relatively short touch events provide a large instantaneous change in sensor values.

In liquid-level sensing applications, temperature compensation is more difficult. Unlike normally non-touched, finger-activated CAPSENSE™ buttons, cannot assume that the liquid sensor is uncovered with liquid as it can be covered to any level for any length of time. Therefore, compensate for temperature variation through the use of algorithms and optimized sensor designs. It is also important to accept that temperature has an impact on accuracy—design tradeoffs must be made to achieve an acceptable accuracy level. Specific temperature compensation methods are detailed in the sections describing each sensor pattern.

A secondary effect of temperature is condensation. A liquid that is significantly colder than the ambient air temperature can cause condensation to form on the sensor surface. Condensation can cause a higher capacitance, which in turn causes an increased error. Condensation during low-temperature testing can be reduced by insulating the surface of the sensor. Another technique is to provide a small insulating air gap

between the liquid container and the sensor substrate. The air gap should be no larger than 3 mm for best performance.

3.2.2 Parasitic capacitance

Parasitic capacitance (C_P) is the unwanted capacitance that exists between parts of an electronic circuit simply because they are located close to each other. The CAPSENSE™ Component measures the total capacitance, which is the total of parasitic and liquid capacitance (CTOTAL = C_P + C_L) within the limits of its dynamic range. In systems, where CP is larger, the parameter of interest, C_L, is the smaller of the total measured signal. This reduces the system sensitivity to liquid presence and therefore total system accuracy.

The main components of parasitic capacitance in CAPSENSE™ designs are trace capacitance and sensor capacitance. C_P is a nonlinear function of sensor size, trace length, trace width, and trace-to-trace spacing. There is no simple relation between C_P and PCB layout features, but general principles exist. Increases in sensor size, increases in trace length and width, and decreases in the annular gap all cause an increase in C_P. One way to reduce C_P is to increase the gap between the sensor and the ground. Unfortunately, widening the gap between sensor and ground decreases noise immunity. More detailed layout guidance is provided in Sensor layout section.

3.2.3 Mechanical variations

Mechanical variations within the system can take many forms but all result in a change to the sensors' C_P. There are two types of mechanical variations.

- Static variation is generally caused by manufacturing tolerances in the sensor assembly, PCB, and sensor alignment to the liquid container. Static variations, if understood and controlled, are compensated for during manufacturing with the baseline calibration operation described in Segmented sensors and Differential sensors sections
- Dynamic variation is caused by changes during operation. These changes often manifest themselves as changes to C_P by changing the sensor's capacitor dimensions. Similar to temperature changes, dynamic variation is difficult to compensate for. It is best to design the system to minimize dynamic mechanical variation effects on the sensors

The most common mechanical variation encountered is a change in the distance between capacitor plates, where the first plate is the sensor and the second plate is the liquid surface. This can be caused by air bubbles in the adhesive used to attach the sensors to the container—the bubbles grow and shrink with air pressure. Another cause occurs when the sensors are not directly attached to the liquid container, allowing a changing air gap between the container wall and the sensor substrate. To reduce these factors, ensure that there are no air bubbles present and the mechanical design is sufficient to maintain accurate sensor alignment.

3.3 CAPSENSE™ Component setup

Liquid level sensing uses the standard ModusToolbox™ IDE's CAPSENSE™ Middleware APIs to scan the liquid level sensors. The CAPSENSE™ Configurator in ModusToolbox™ also supports SmartSense Auto-Tune capability, allowing either auto-tuning or manual tuning for liquid level sensing. If manual tuning is used, the parameters discussed in this section must be properly tuned for correct results. Parameters not discussed in this section either have no effect on liquid level sensing or should remain at their default values. For more information on the Component parameters and their effects on tuning, see the PSOC™ 4 CAPSENSE™ design guide and PSOC™ 4 CAPSENSE™ Middleware APIs list.

Begin the CAPSENSE™ setup by selecting **Manual** tuning as the tuning mode in the CAPSENSE™ Configurator's **Basic** tab, as Figure 5 shows (either Manual Tuning or Auto-Tuning).

In most designs, the Compensation IDAC can be left disabled. It is acceptable to enable it to explore additional tuning options.

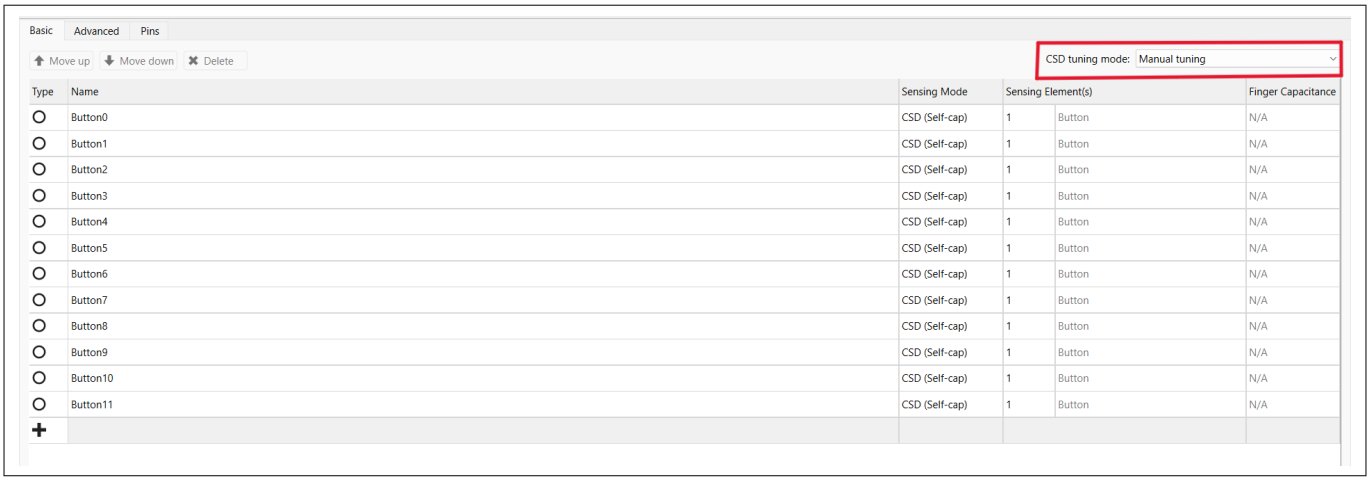


Figure 5 CAPSENSE™ Configurator - Basic tab

On the **Advanced** > **CSD Settings** tab as shown in Figure 6, set Modulator clock frequency, Rawcount calibration level, Compensation IDAC, it is recommended to set the Rawcount calibration level to 85% but can also be changeable based on the environmental conditions. The Inactive sensor connection and Shield must all remain in their default settings as their purpose is to reduce the impact of water in touch designs. In liquid-level sense designs, maximize the effect of water on the sensors. Also, it is always better to keep the IDAC Auto Calibration Enabled.

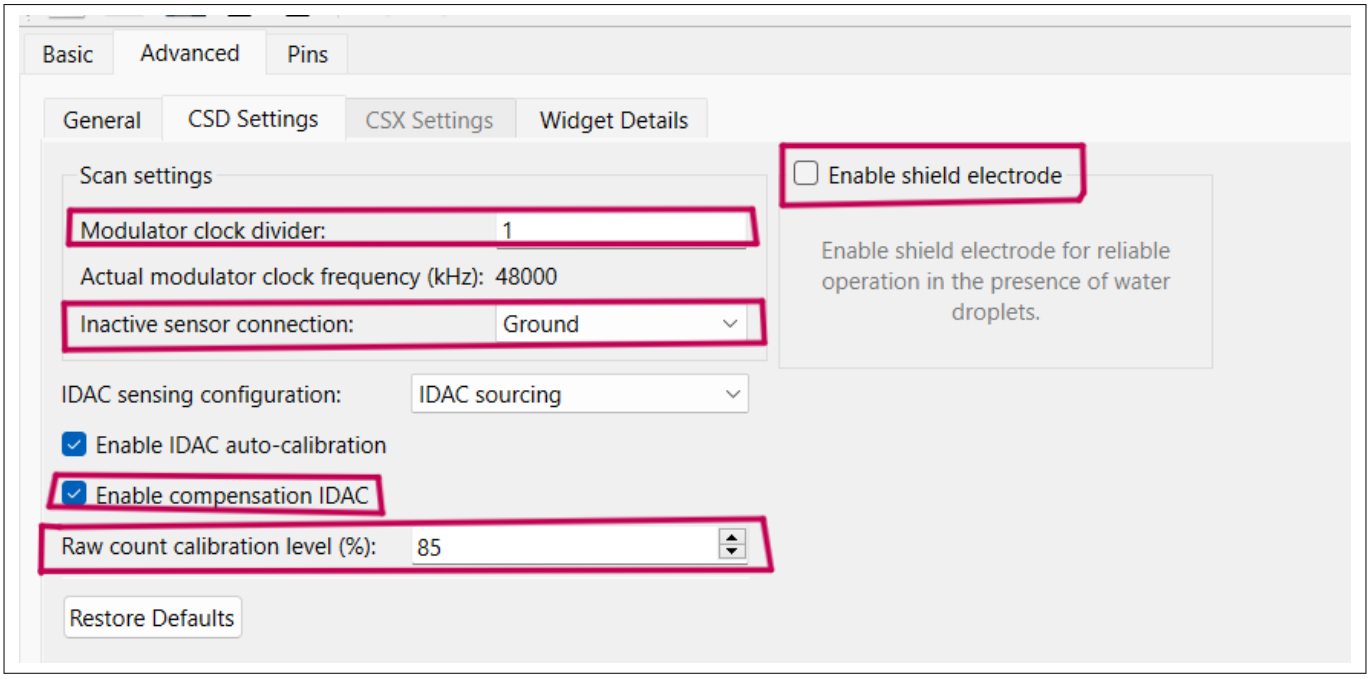


Figure 6 CAPSENSE™ Configurator - Advanced tab

Set all the widget-related parameters at the **Advanced** > **Widget details** tab as shown in Figure 7. Set all the essential tuning parameters, such as Sense clock frequency, Scan Resolution, IDAC value, Threshold values. If SmartSense (Auto-Tune) is used, then all these parameters are computed and set automatically during runtime.

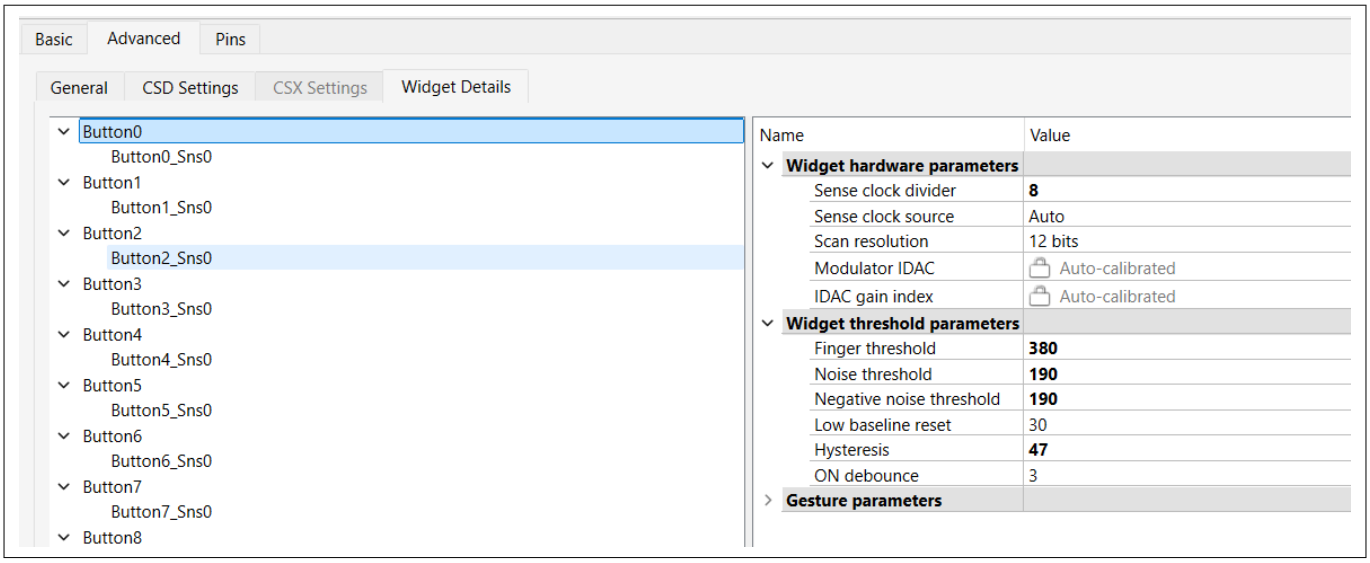


Figure 7 CAPSENSE™ Configurator - Widget Details tab

3.4 Sensor pattern choice

This document supports two sensor patterns. See Table 1 to determine which sensor pattern is best suited to a given set of design constraints.

Table 1 Sensor design attributes

Sensor design attribute	Differential sensor	Segmented sensor
Number of required CAPSENSE™ sensors	2	1 or more (typically, 10 to 20)
Sensor pattern constraints	Both sensors must be the same size and specific triangular shape	Each sensor can be a different size and shape
Resolution	Linear, ~1 mm	Determined by the total height of the container divided by the number of sensors. ≥ 1 mm
Accuracy	Typically, 20% accurate near empty improving to 5% accurate near full	Maximum accuracy is equal to the largest sensor's height and may be improved at critical levels by using shorter sensors (typically, each sensor height is 5-10% of the total sensor)

3.5 Segmented sensors

The most accurate and flexible sensor pattern is an array of sensor segments equally spaced from the top to the bottom of the liquid container. Each segment of the example sensor pattern #1 shown in Figure 8 provides an incremental portion of the total liquid level. While the 'segmented' pattern requires more sensor elements and pins compared to the 2-sensor differential version, it allows liquid-level resolution to be customized for each design.

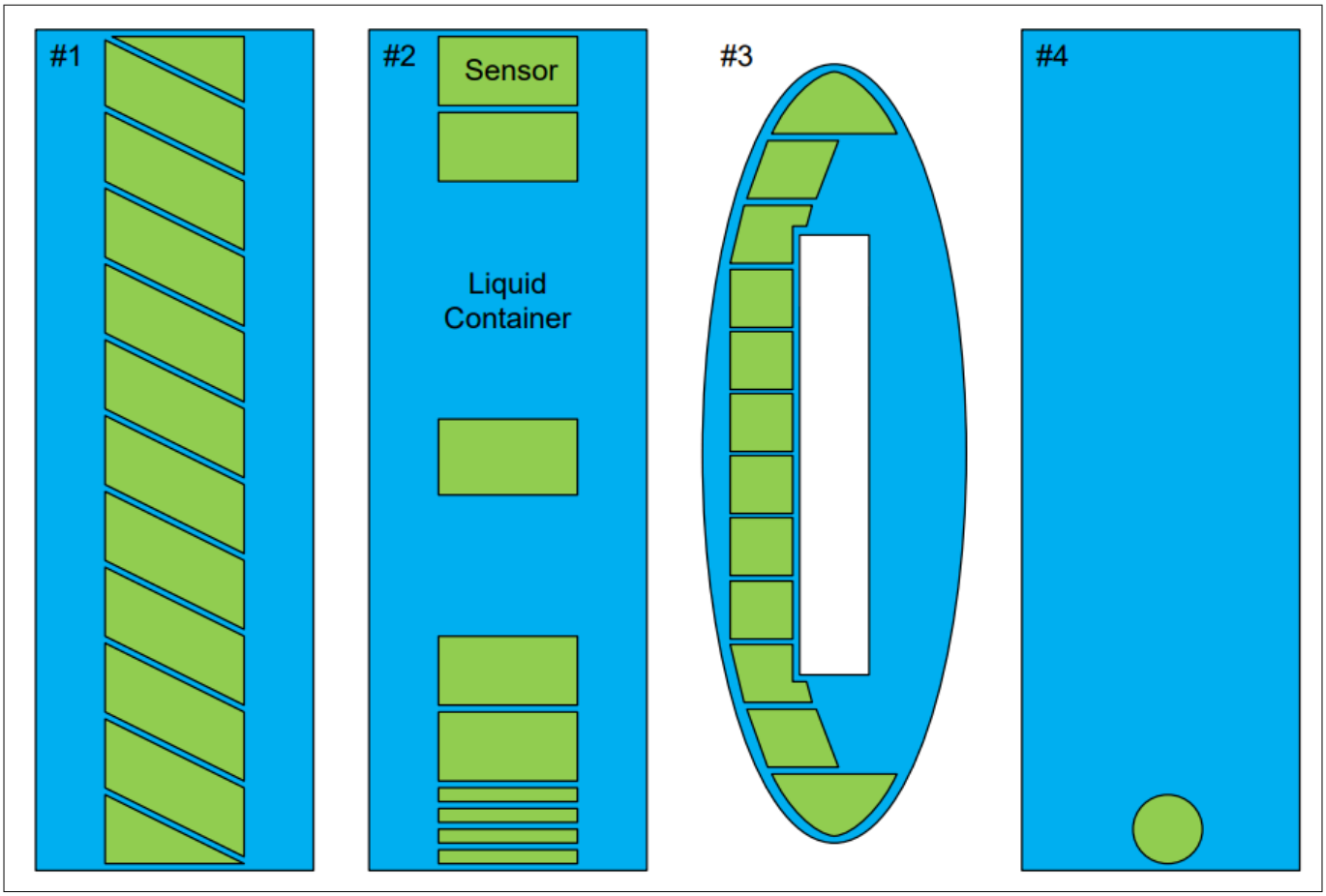


Figure 8 Segmented sensor examples

Sensor design and placement is very flexible, allowing the sensor shape to fit within a given industrial design and provide resolution tailored to specific portions of the container.

- **Sensor height:** Each sensor's height is independent of the other sensors in the array. Changing the sensor height allows a small number of sensors to provide increased accuracy at critical levels, such as empty and full, while providing decreased resolution in the middle of the container. The example sensor #1 shown in Figure 8 uses a single height for most of the sensors and ½ height sensors at the top and bottom to increase the accuracy at empty and full levels
- **Sensor shape:** Individual sensor shapes can be any geometric solid to assist in arranging the sensor array on complex-shaped liquid containers or to account for industrial design constraints. Typical constraints include custom blow-moulded containers and pour spouts or mounting flanges that would otherwise block sensor placement as shown in example #3 of Figure 8
- **Sensor Placement:** Sensors are evenly spaced along the container's height as shown in example #1 of Figure 8. If the application only needs liquid-level information at specific levels, sensors can be clustered at some levels and completely absent at other levels as shown in example #2 of Figure 8. At the lowest limit of resolution, a single sensor can indicate a single critical liquid level, such as when the container is empty as shown in example #4 of Figure 8

3.5.1 General operation

The segmented sensor pattern provides the highest accuracy across all operating conditions and is the best choice for most designs. The sensor pattern is segmented, allowing each sensor element to accurately measure the liquid level in its limited range. Each sensor generates a binary output when it is half covered with liquid, by comparing the sensor counts with a reference value. The reported liquid level is the height of the highest sensor

covered with liquid; if using an equally sized sensor array, it is the sum of the covered sensors multiplied by the height of each sensor.

In the example #1 sensor pattern shown in Figure 9, the top and bottom sensors are each half the height and area of the other sensors, thereby increasing the measurement resolution and accuracy at the empty and full limits. The two half-height sensors combined have the same height as one full-size sensor, thereby reducing the equivalent number of full-height sensors by one. Maximum liquid-level error is equal to 100% divided by the number of equally spaced sensors. The maximum error for the 12-sensor example is 100% divided by 11 (11 = 10 full height + ½ height + ½ height) which equals 9% for the full-height sensors. The typical error is half of the maximum error, which is 4.6% for the sensor shown in example #1. Due to the binary output of each sensor, the reported liquid level of the segmented sensors is stair-step compared to the actual liquid level as seen in Figure 9.

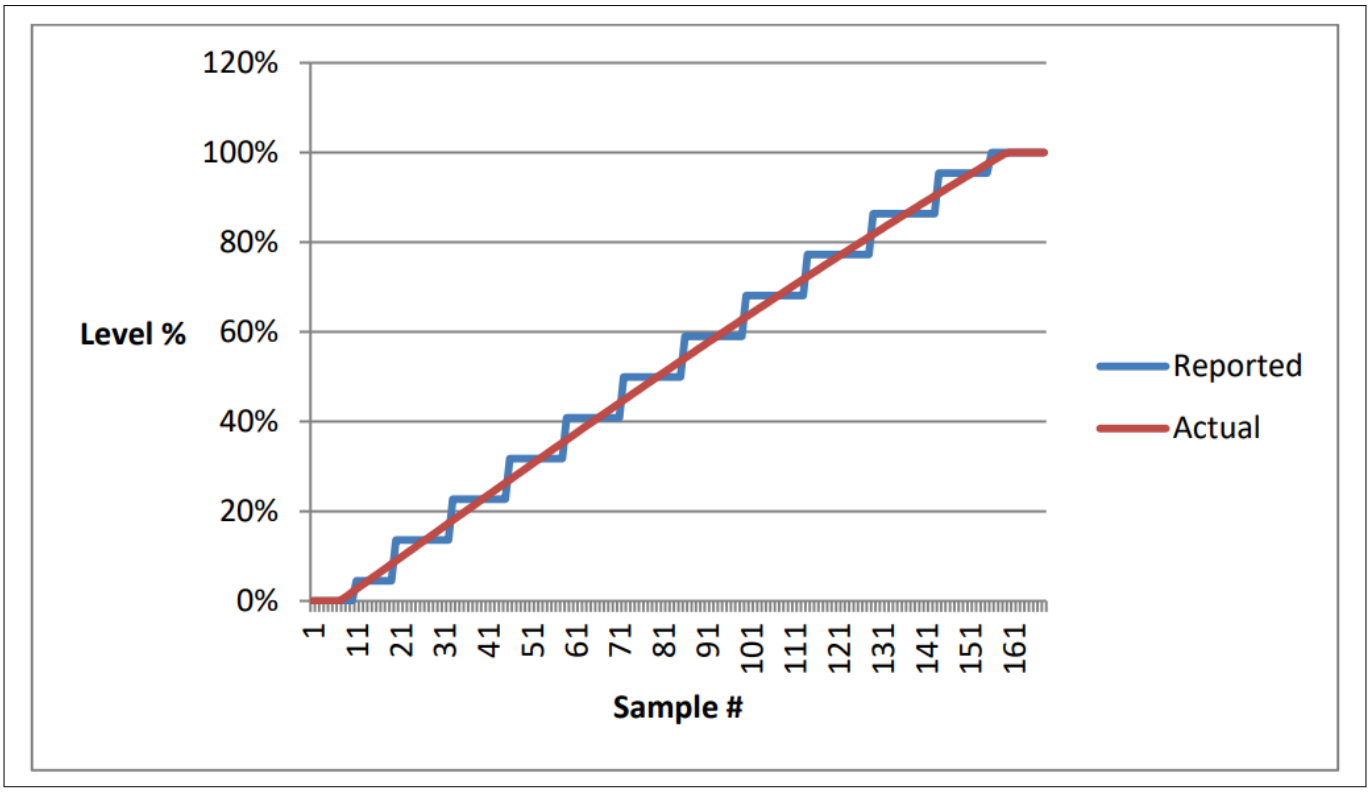


Figure 9 12-sensor segmented response

3.5.2 Liquid-level calculations

The calculation of liquid level from segmented sensors is very simple and comprises the following four main steps:

1. Measure static baseline of sensors:

The absolute value of each sensor is not important for determining whether a liquid is present because the parasitic capacitance of the sensor and device-to-device variations can greatly impact the sensor's base value relative to other sensors and finished units. Prior to use, determine the raw value baseline of each sensor in an empty liquid container. Measure the baseline values for each unit during manufacturing test and stored in nonvolatile memory. On subsequent bootups, the baseline values are recalled for use. Liquid level is the most accurate when the operating conditions and temperature are most similar to the baseline measurement conditions. It is recommended to measure the baseline value at a temperature near the center of the operating temperature range for improved accuracy. An optional method to increase the accuracy uses a temperature sensor to measure the temperature of the liquid or container and modify the baseline value within an experimentally derived range

2. Remove static baseline from sensors counts

   During normal operation, each sensor's baseline value is subtracted from its raw value to find the difference counts. Difference counts are then limited to positive values and optionally scaled to a normalized value to account for sensors of different sizes and capacitance values

3. Determine which sensor segments are covered with liquid

   To determine whether a sensor is covered with liquid, its difference counts are compared to a threshold value as shown in Figure 10. Figure 10 shows the difference counts from each sensor as the container is filled. The data samples shown are recorded at one-second intervals while filling the container at a calibrated 1 mm per second rate. If the difference counts are above the threshold, it indicates that the sensor is covered and the binary state recorded. Hysteresis can optionally be added to the threshold to reduce the sensor state oscillation due to noise near the threshold value

   Set the threshold value at ½ the sensor difference counts when fully covered with liquid. The ½ maximum threshold value ensures good level-sensing performance across a wide operating temperature range

   Sensor arrays with equally sized and spaced sensors can record the total number of sensors covered. Sensor arrays with irregularly sized or positioned sensors should record which sensor is the highest active sensor

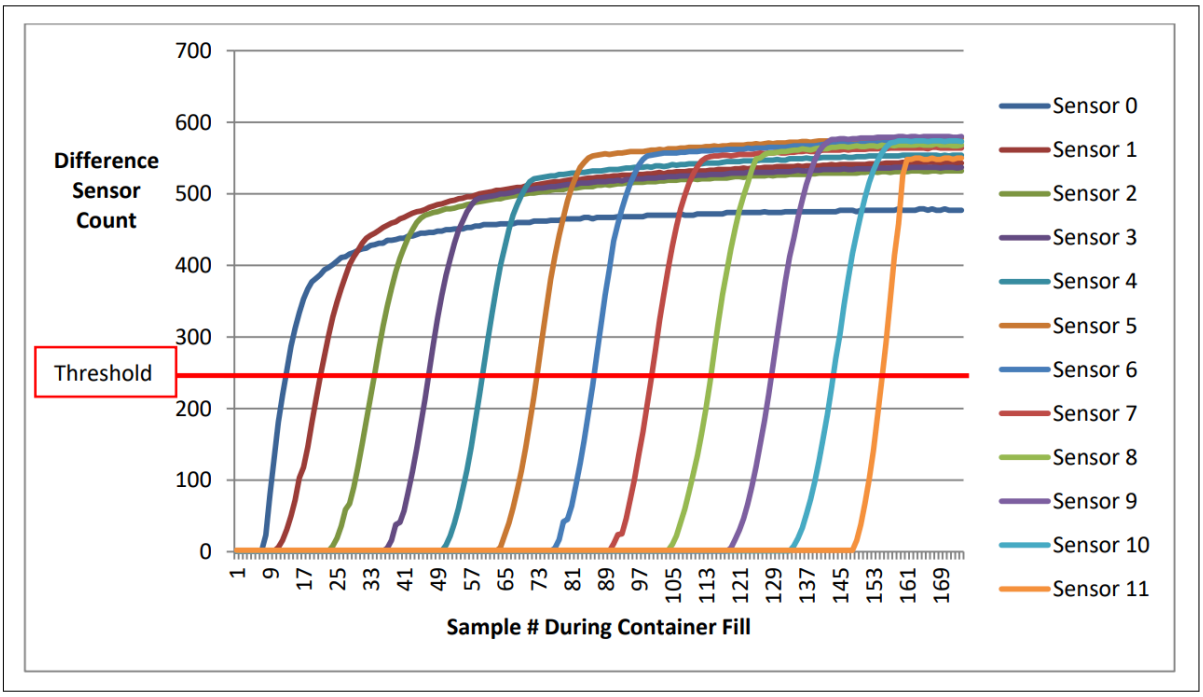


Figure 10 Segmented 12-sensor difference count versus sample number during fill

4. Calculate the liquid height:

   If the total number of equally sized sensors covered is known, then multiply the number of covered sensors by the standard sensor height. Standard sensor height equals the total sensor pattern height divided by the number of sensors. If the highest sensor covered is known, look up the height of that sensor from an array that contains the height of each sensor. Both methods result in a stair-stepped response as shown in Figure 12

   The calculated liquid height can optionally be filtered as required by the system response and converted to percent full by dividing the reported height by the total sensor pattern height

3.5.3 Accuracy

Because each sensor provides a binary contribution to the liquid level, the maximum error is limited to the highest partially covered sensor. The 12-sensor example design described in this section has a typical error of

±7 mm (± 4.6%) because the sensors are 14 mm high. The error has the following three components that combine for the total error:

1. Quantized error is formed by the resolution of the sensor's physical size and is ± half of the sensor's height. For example, a 10 mm tall sensor has a quantized error of ± 5 mm. Decreasing the sensor height improves the accuracy. The sensor height should always be greater than or equal to the container wall thickness to ensure sufficient signal level
2. Temperature error is caused by temperature variations from the baseline temperature that change material permittivities. Temperature deviation in liquid-level sensing applications is typically less than 25% of the full-scale sensor value. Increasing the number of sensors decreases the temperature component of error
3. Random errors are caused by parametric deviations and electronic interference in the system and difficult to calculate. Random errors can never be eliminated but can be minimized by following the design and layout recommendations from the AN85951 – PSOC™ 4 CAPSENSE™ design guide

3.6 Differential sensors

The differential sensor method uses the ratio value of two sensors with triangular patterns and provides a low-cost method of determining the liquid level. The pattern shown in Figure 11 looks similar to a backgammon game board and is therefore commonly referred to as the 'backgammon' pattern. The differential method provides a reduced accuracy compared to the segmented sensor but requires only two sensors for reduced sensor cost. The sensor pattern is composed of two sensors shaped so that the ratio of their values is equal to the percent value of the liquid level.

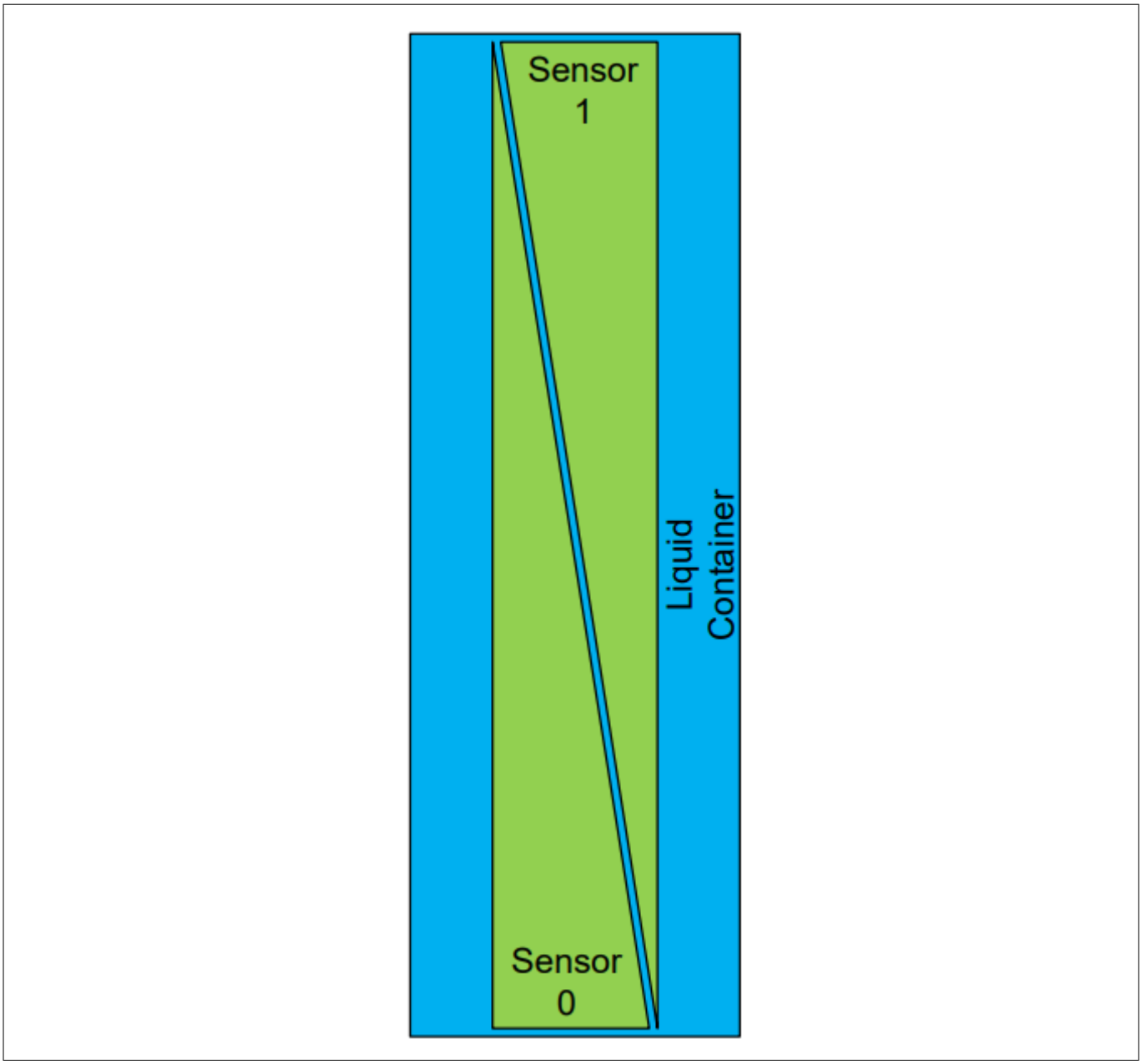


Figure 11 Differential sensor example

3.6.1 General operation

The simplest sensor pattern is the differential pattern as only two sensors are required. The two key differences from the segmented pattern are reduced accuracy near the empty level and strict layout requirements for the two sensors.

The sensors are required to be right triangles with the bottom edge of Sensor0 and bottom point of Sensor1 at the empty liquid level while the opposite top edge and point must be at the full liquid level. While both sensors are covered to the same level with liquid, the total surface area covered by each sensor is very different. The sensor area covered by liquid directly relates to the total capacitance seen by each sensor. Observe Figure 12 that Sensor0's value is increasing faster than Sensor1's value because it has more area in contact with the liquid just above the empty level than the small point of Sensor1. As it approach the full liquid level, see that the Sensor0's rate of change decreases approaching the point while Sensor1's remains constant right up to the 100% level.

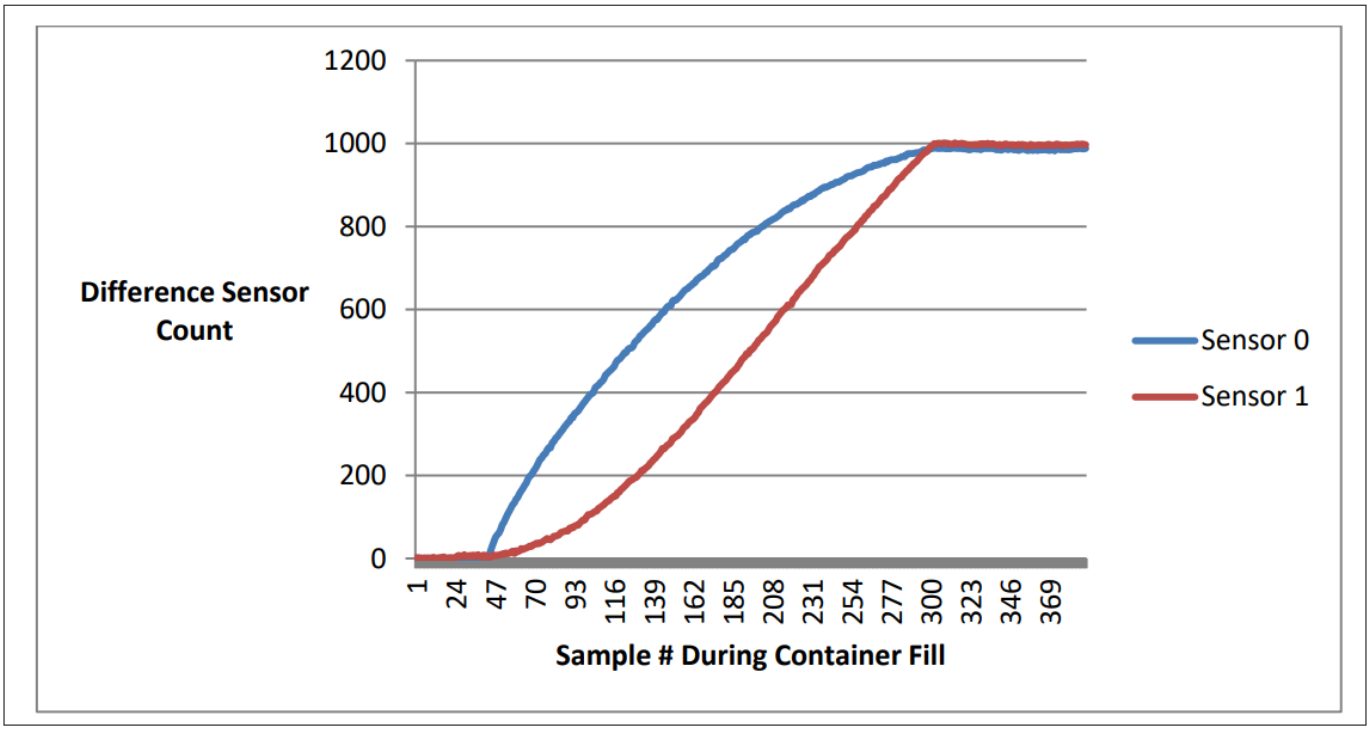


Figure 12 Two-sensor differential values

Because sensor count values are small near the empty level, any sensor error is magnified into a large error in liquid level. As the liquid level increases, the same error value becomes a smaller portion of the total value; therefore, the accuracy is best at the full level. So, the differential sensor should not be used in designs that require accurate determination of an empty container.

The differential nature of the sensors also requires that both sensors be of exactly the same size, have the same exposure to the liquid, and the same parasitic capacitance. Even an optimally constructed sensor pattern has errors impacting the calculated liquid level; therefore, a method of compensating the reported level is provided.

See Figure 13 that the calculated liquid level output is more linear than the segmented sensor pattern and there is a large error ‘spike’ near empty on the ‘Ratio Basic’ graph.

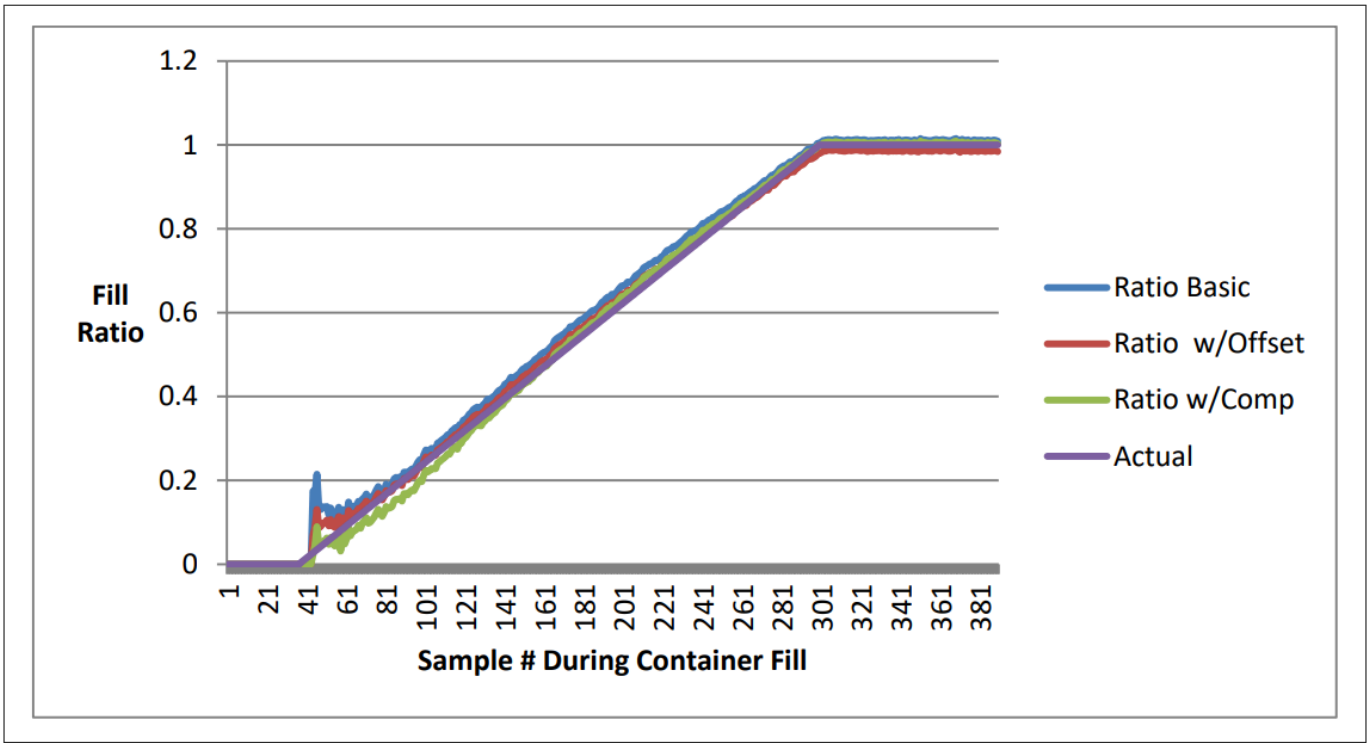


Figure 13 Two-sensor differential response

3.6.2 Liquid-level calculations

The calculation of liquid level from differential sensors is a simple ratio calculation. There are four steps:

1. Measure the static baseline of sensors:

   The baseline storage process and reason are the same as for the segmented sensor pattern
2. Remove the static baseline from sensors counts:

   During normal operation, each sensor's baseline value is subtracted from its raw value to find the difference counts. The difference counts are then limited to positive values
3. Calculate the sensor ratio:

   The sensor ratio is calculated as Sensor1 divided by Sensor0 when using the sensor pattern shown in Figure 11. The resulting ratio is shown as the 'Ratio Basic' graph in Figure 9

   Due to the small values on Sensor0, which is the denominator of our ratio equation, any error or offset on Sensor1 is magnified as can be seen with the 0.2 spike (20% error) in the ratio near the empty level. To reduce this error, add a small positive offset to Sensor0's value to minimize the effect of the error. By adding an offset of +25 counts to Sensor0's difference counts in the example data, the error is cut in half as 'Ratio w/Offset' shows in Figure 9. The greater the offset value, the smaller the empty error becomes, but the greater the full error. The offset of +25 counts decreases the maximum ratio possible to 0.98 of the full liquid level, and is never reported as 100% full

   One additional special case checked for are divide by zero errors if Sensor0 = 0. To avoid this, if Sensor0 = 0, then skip the ratio calculation and set the liquid level to 0%
4. Compensate the sensor ratio:

   As seen in Step 3 how the ratio error can be reduced, but there can still be original or offset-induced linearity issues with the reported liquid level. The 'Ratio w/Comp' graph in Figure 9 shows the improvement in accuracy that can be attained by performing a slope and offset modification to the 'Ratio w/Offset' value. The combination of a small slope adjustment and offset further reduces the error near empty and allows the full value to reach a ratio of 1.0 (100%). The compensated ratio is calculated

by multiplying the offset ratio with an experimentally derived constant and then adding an offset constant. In most designs, these constants will both be near 1

The calculated liquid height can optionally be filtered as required by the system response and converted to height in millimeters by multiplying the reported ratio by the total sensor-pattern height

3.6.3 Accuracy

Because of the small numbers used in the ratio calculation near the empty level, the calculated liquid level is least accurate near empty. The two-sensor example design described in this section has a typical error of ±15% (± 23 mm) near empty. The error has the following three components that combine for the total error:

1. Small number errors are caused by the large error produced when small numbers are used to calculate a ratio. For example, an error of 1 divided by 100 is a 1% error, while an error of 1 divided by 2 is a 50% error. This error is intrinsic to the differential design but can be reduced by adding a positive offset to the denominator sensor value. As the added offset increases, the error at the full level increases, which can be partially compensated for. The optimal offset and compensation slope equation constants must be experimentally determined for each design
2. Temperature error is created by the sensor value offset caused by temperature deviation from the baseline temperature. Temperature deviation in liquid-level sensing applications is typically less than 25% of the full-scale sensor value. Temperature affects both sensors simultaneously; therefore, the temperature effect is less than 25% as the offsets partially cancel out
3. Random errors are caused by parametric deviations and electronic interference in the system and difficult to calculate. Random errors can never be eliminated but can be minimized by following the design and layout recommendations provided in the AN85951 – PSOC™ 4 CAPSENSE™ design guide

The ratiometric calculations require fraction values to maintain accuracy. While floating-point numbers could be used, they result in larger and slower code. To provide the required accuracy and avoid floating-point numbers and calculations, fixed precision values are used instead. If you are not familiar with fixed precision numbers and calculations, see articles on the subject at Fixed-point arithmetic and Binary scaling.

4 Sensor layout

4.1 General layout considerations

Liquid-level sensor layout guidelines are nearly identical to the CAPSENSE™ touch layout guidelines in the AN85951 – PSOC™ 4 CAPSENSE™ design guide. Therefore, only key highlights and differences are presented in this application note.

- Ensure that the industrial design for sensors and traces are isolated from conductive materials or sources of interference. Air gaps provide the best isolation from conductive objects as the relative permittivity of air is the lowest available at 1.0. An air gap of at least 5 mm provides excellent isolation when used with a hatched ground plane. As conductive objects approach closer to the sensors, the effectiveness of a hatched ground decreases as more of the sensors' field escapes between the hatched lines to the conductive object
- The liquid container can be any non-conductive material. Materials with higher relative permittivity perform better by increasing the signal level of sensors. Material guidelines are the same as for touch application overlay materials shown in Table 2

Table 2 Relative permittivity of container materials

Material	ε_r
Air	1.0
Glass (standard)	7.6 to 8.0
Glass (ceramic)	6.0
Polycarbonate (Lexan)	2.9 to 3.0
Acrylic (Plexiglass)	2.8
ABS	2.4 to 4.1

- Minimize the container wall thickness because thinner walls increase the sensor signal level. Container walls thinner than 5 mm give the best performance. Use thicker walls with increased scan times and possible decreases in accuracy
- The sensor-to-container adhesive must be non-conductive and should have a high permittivity. 3M 467MP and 468MP transfer tapes provide excellent performance
- Minimize the parasitic capacitance (C_P) to maximize the sensors' response to liquids. The area of each sensor in the sensor pattern should be less than 3,000 mm^2. The space between sensors should be equal to the container wall thickness or 1 mm, whichever is wider. The minimum sensors area is 25 mm^2 while optimal performance is typically found near 150 mm^2
- Sensor and PCB board thickness should be greater than 0.25 mm to maximize the gap between the sensors and the ground plane
- Sensor trace width should not be wider than 7 mil (0.18 mm) with a 10 mil to 20 mil (0.25 mm to 0.51 mm) clearance. Ideally, minimize the trace length to less than 30 cm
- Route the sensor traces on the PCB side opposite of the liquid surface to reduce interaction. Route the traces at least 0.25 mm from switching signals and should have a ground between them. If sensor traces must cross switching signals, they should cross at right angles
- Use a ground plane on the top and bottom of the PCB, wherever sensor traces are present to increase sensor immunity to electronic interference. To limit sensor C_P, the ground planes over and under the sensors, and traces must be hatched with a 17% fill (7-mil line, 70-mil spacing). Use all other areas a solid ground plane optimized for precision analog circuits

- Provide a power supply decoupling capacitors as recommended in the device datasheet to minimize noise in the high-sensitivity analog system of PSOC™ CAPSENSE™
- GPIO pins used for sensors should have 560 Ω resistors in series with the sensors, placed as close to the pin as practical for increased EMI immunity

4.2 Segmented sensor pattern considerations

There are no additional requirements for segmented sensor patterns, as the sensors can be of any practical shape.

4.3 Differential sensor pattern consideration

The differential sensor pattern has the following additional unique constraints to ensure performance:

- Both sensors must be as close in size as possible to provide matched C_P
- Both sensor traces should have similar routing to provide matched C_P
- Both sensors must be precisely aligned and mirrored right triangles, as Figure 11 shows
- The optimal height-to-width aspect ratio of each sensor is 8:1. Sensor aspect ratios should fall within the range of 16:1 to 1:1 with lower ratios providing greater accuracy

5 Code implementation with PSOC™ 4100S Max

The following code snippet for capacitive liquid level sensing using PSOC™ 4100S Max is provided along with the algorithm, it is just a sample code, you can modify the algorithm as per your need.

```
initCapsense();             /* Capsense Initialization */
initCapsenseTuner();        /* Tuner initialization */
for (;;)
    {
        /* Check for CapSense scan complete*/
        if(CY_CAPSENSE_NOT_BUSY == Cy_CapSense_IsBusy(&cy_capsense_context))
        {
            /* Process all widgets */
            if (CY_CAPSENSE_STATUS_SUCCESS !=
Cy_CapSense_ProcessAllWidgets(&cy_capsense_context))
            {
                Cy_SCB_UART_PutString(scb_1_HW, "Error in processing widgets\r\n");
                CY_ASSERT(0u);
            }
            /* Delay to control data logging rate */
            Cy_SysLib_Delay(UART_DELAY);

            /* Read and store new sensor raw counts, remove empty offset
             * calibration from sensor raw counts.
             */
            for(uint8_t i = 0; i < NUMSENSORS; i++)
            {
                sensorRaw[i] = cy_capsense_tuner.sensorContext[i].raw;
                sensorDiff[i] = sensorRaw[i];

            }

            /* Start scan for next iteration */
            if (CY_CAPSENSE_STATUS_SUCCESS != Cy_CapSense_ScanAllWidgets(&cy_capsense_context))
            {
                Cy_SCB_UART_PutString(scb_1_HW, "Error in scanning widgets\r\n");
                CY_ASSERT(0u);
            }
            if(cal_flag == TRUE)
            {
                cal_flag = FALSE;
                store_calibration();
            }

            /* Remove empty offset calibration from sensor raw counts
             * and normalize sensor full count values.
             */
            sensorActiveCount = 0;
            for(uint8_t i = 0; i < NUMSENSORS; i++)
            {
                sensorDiff[i] -= sensorEmptyOffset[i];
                if (sensorDiff[i] < 0)
                {
```

```
                    sensorDiff[i] = 0;
                }
                sensorProcessed[i] = (sensorDiff[i] * sensorScale[i]) >> 8;
                /* Find the number of submerged sensors */
                if(sensorProcessed[i] > sensorLimits[i]/2)
                {
                    /* First and last sensor are half the height of middle sensors. */
                    if((i == 0) || (i == NUMSENSORS - 1))
                    {
                        sensorActiveCount += 1;
                    }
                    /* Middle sensors are twice the height of 1st and last sensors */
                    else
                    {
                        sensorActiveCount += 2;
                    }
                }

            }
        }

        /* Calculate liquid level height in mm */
        levelMm = sensorActiveCount * (sensorHeight >> 1);
        /* If level is near full value then round to full.
         * Avoids fixed precision rounding errors.
         */
        if(levelMm > ((int32_t)LEVELMM_MAX << 8) - (sensorHeight >> 2))
        {
            levelMm = LEVELMM_MAX << 8;
        }

        /* Calculate level percent. Stored in fixed precision
         * 24.8 format to hold fractional percent.
         */
        levelPercent = (levelMm * 100) / LEVELMM_MAX;
        /* Report level and process UART interfaces */
        display_cur_liquid_level();
#if CAPSENSE_TUNER_EN
        /* Establishes synchronized communication with the CapSense Tuner tool */
        Cy_CapSense_RunTuner(&cy_capsense_context);
#endif
    }
```

The code determines the liquid level based on the number of submerged sensors. There are totally 12 sensors, R0 to R11.

The algorithm for the above code is as follows:

1. Initialize CAPSENSE™ peripheral and CAPSENSE™ Tuner using the CAPSENSE™ Middleware APIs
2. Read the new Rawcounts and store it in an array called **sensorRaw[]**
3. Final processed Rawcounts will be stored in the **sensorProcessed[]** array
4. Track the submerged sensors by checking whether the Rawcount of each sensor has crossed the threshold limit or not with the help of the `sensorActiveCount` variable

5. Each sensor have fixed predefined height, using that find that up to what level water is filled (levelMm = sensorActiveCount × sensorHeight)
6. Finally, the percentage of water filled will be calculated and sends via the UART interface using the `display_cur_liquid_level()` API

6 Summary

This application note provided guidance on the design and implementation of capacitive liquid-level sensing. Two sensor patterns and algorithms were presented to allow optimization of the design to meet requirements.

References

[1] Infineon Technologies AG: *AN64846 – Getting started with CAPSENSE™*; Available online
[2] Infineon Technologies AG: *AN85951 – PSOC™ 4 CAPSENSE™ design guide*; Available online
[3] Infineon Technologies AG: *AN79953 – Getting started with PSOC™ 4 MCU*; Available online
[4] Infineon Technologies AG: *CY8CKIT-041S-MAX PSOC™ 4100S Max Pioneer Kit*; Available online
[5] Infineon Technologies AG: *CY8CKIT-022 CAPSENSE™ Liquid Level Sensing Shield*; Available online
[6] Infineon Technologies AG: *ModusToolbox™ software*; Available online
[7] Infineon Technologies AG: *ModusToolbox™ documentation*; Available online
[8] Infineon Technologies AG: *CAPSENSE™ controllers*; Available online

Glossary

CL
liquid capacitance

CP
parasitic capacitance

CSD
capacitive sigma delta

IDAC
current digital-to-analog converter

LLS
liquid-level sensing

Revision history

Document revision	Date	Description of changes
**	2025-05-06	Initial release

Trademarks

PSOC™, formerly known as PSoC™, is a trademark of Infineon Technologies. Any references to PSoC™ in this document or others shall be deemed to refer to PSOC™.

Trademarks

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

Edition 2025-05-06

Published by

Infineon Technologies AG

81726 Munich, Germany

© 2025 Infineon Technologies AG

All Rights Reserved.

Do you have a question about any aspect of this document?

Email: erratum@infineon.com

Document reference
IFX-cyz1742475567135

Important notice

The information contained in this application note is given as a hint for the implementation of the product only and shall in no event be regarded as a description or warranty of a certain functionality, condition or quality of the product. Before implementation of the product, the recipient of this application note must verify any function and other technical information given herein in the real application. 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) with respect to any and all information given in this application note.

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.