PSOC™ 4 CAPSENSE™ ultra-low-power capacitive sensing techniques

About this document

Scope and purpose

This application note describes how to use the PSOC™ 4 MCU family with fifth-generation CAPSENSE™ Multi-Sense Converter Low Power (MSCLP) devices to achieve a capacitive sensing application with the lowest power consumption and provides guidelines for firmware and device configuration and hardware design. This application note focuses on CAPSENSE™ specific guidelines; for PSOC™ 4 general guidelines, see AN86233 - PSOC™ 4 MCU low-power modes and power reduction techniques.

Intended audience

This document is intended for the firmware engineers who configure or use the device for low-power touch-sensing applications.

This document assumes that you are familiar with PSOC™, CAPSENSE™, and ModusToolbox™; see the relevant resources to get started.

If you are new to	See this
PSOC™ 4 MCU architecture	AN79953 – Getting started with PSOC™ 4 MCU
CAPSENSE™ technology	AN85951 – PSOC™ 4 and PSOC™ 6 MCU CAPSENSE™ design guide
Application development for PSOC™ 4 using ModusToolbox™ software	ModusToolbox™ software

Table of contents

About this document . . . 1
Table of contents . . . 2
1 Introduction . . . 4
2 Power consumption in low-power designs . . . 5
2.1 Power budgeting . . . 6
2.2 PSOC™ 4 device with MSCLP is ideal for low-power solutions . . . 7
2.3 Factors affecting low-power operation . . . 8
2.3.1 Scan duration . . . 8
2.3.2 Refresh rate . . . 8
2.3.3 Signal-to-noise ratio (SNR) . . . 8
2.3.4 Number of sensors in low-power mode . . . 8
2.4 Power estimation . . . 9
2.4.1 Current consumption at different scanning stages . . . 9
3 Firmware design considerations . . . 10
3.1 Power modes available in PSOC™ 4 device with MSCLP . . . 10
3.1.1 Power consumptions in different application states . . . 10
3.1.2 CAPSENSE™ hardware scanning modes . . . 11
3.2 Firmware techniques for low-power design . . . 12
3.2.1 Regular widget . . . 12
3.2.2 Low-power widget . . . 13
3.2.3 Refresh rate . . . 14
3.2.3.1 MSCLP timer . . . 15
3.2.4 Reducing the scan time in low-power mode . . . 16
3.2.4.1 Wakeup/power-on button . . . 16
3.2.4.2 Ganged sensors . . . 16
3.2.4.3 Proximity sensors . . . 17
3.2.4.4 Touchpad-specific techniques . . . 17
3.2.5 Shield design . . . 18
3.3 Low-power widget parameters . . . 19
3.3.1 Wake-on-touch scan interval . . . 19
3.3.2 Wake-on-Touch timeout . . . 19
3.3.3 Low-power IIR filter . . . 21
3.4 CIC2 filter . . . 22
3.5 Low-power CAPSENSE™ example . . . 23
4 Device configuration considerations . . . 25
4.1 CPU and system clocks . . . 25
4.2 CPU power modes . . . 25
4.3 Power state of other peripherals . . . 26

Table of contents

4.4 Unused pin states . . . 26
4.5 Debug pin state . . . 26

5 Application-specific considerations . . . 28
5.1 Liquid-tolerant applications . . . 28
5.1.1 Reducing refresh rate in Liquid Active mode . . . 29
5.2 Gesture applications . . . 29
5.2.1 Using touch button on WoT . . . 29
5.2.2 Using proximity on WoT . . . 29
5.3 Robustness to external noise . . . 29
5.3.1 Implementation of software filters . . . 29
5.3.2 Implementation of firmware logics . . . 29
5.3.3 Low-power EMC considerations . . . 30
5.3.3.1 Radiated immunity . . . 30
5.3.3.2 Radiated emission (RE) . . . 30

6 Hardware design considerations . . . 31
6.1 Shield regions . . . 31

References . . . 32

Revision history . . . 33

Trademarks . . . 35

Disclaimer . . . 36

1 Introduction

The wearable technology devices from fitness trackers to smart glasses and smart clothes are becoming increasingly popular. The capacitive sensing is one of the key human-machine interfaces (HMI) used in any wearable solution. The battery life is the major challenge in any wearable technology today; therefore, there is a constant need to lower the power consumption while still having the need for the devices to be responsive all the time.

PSOC™ 4 devices with CAPSENSE™ MSCLP addresses this challenge by introducing the new fifth-generation CAPSENSE™ technology, offering an ultra-low-power touch HMI solution. It enables scanning low-power buttons while the device is in DeepSleep and processing the results to wake the device in the event of a touch. This technology also has an inherent autonomous scanning capability, which does not need CPU intervention for scanning sensors. The device is kept in DeepSleep while scanning, therefore, reducing the power in active mode as well.

This application note explains how to design a low-power CAPSENSE™ application with PSOC™ 4 device with CAPSENSE™ MSCLP and discusses various factors affecting power consumption. It also covers the hardware, firmware, and system recommendations to achieve the lowest power consumption.

2 Power consumption in low-power designs

The wearable applications must have a long battery life (recharge cycle time). Currently, available fitness trackers and smart watches claim 5 to 15 days of battery life for normal usage. The battery life is calculated considering the active usage to be approximately 4 percent (~1 hour per day), which contributes to ~25 percent of total current consumption, and the remaining 96 percent time in a low-power mode taking ~75 percent of remaining current consumption. Therefore, the current usage in low-power mode has a major impact on having an extended battery life.

Most smart wearables use touch sensing as the trigger to transition from a low-power mode to an active mode. Touch-sensing solutions currently have different low-power modes to enable scanning in low-power state and active state.

In low-power mode, MCUs typically scan sensors with a low refresh rate to detect any user activity to transition to active mode with a higher refresh rate to meet the user experience requirement. The low-power mode is achieved by waking the CPU at a low frequency to scan and process the results. This increases the average low-power mode current and impacts the battery life. Figure 1 shows the instantaneous and average current waveform for a sensing MCU in different modes.

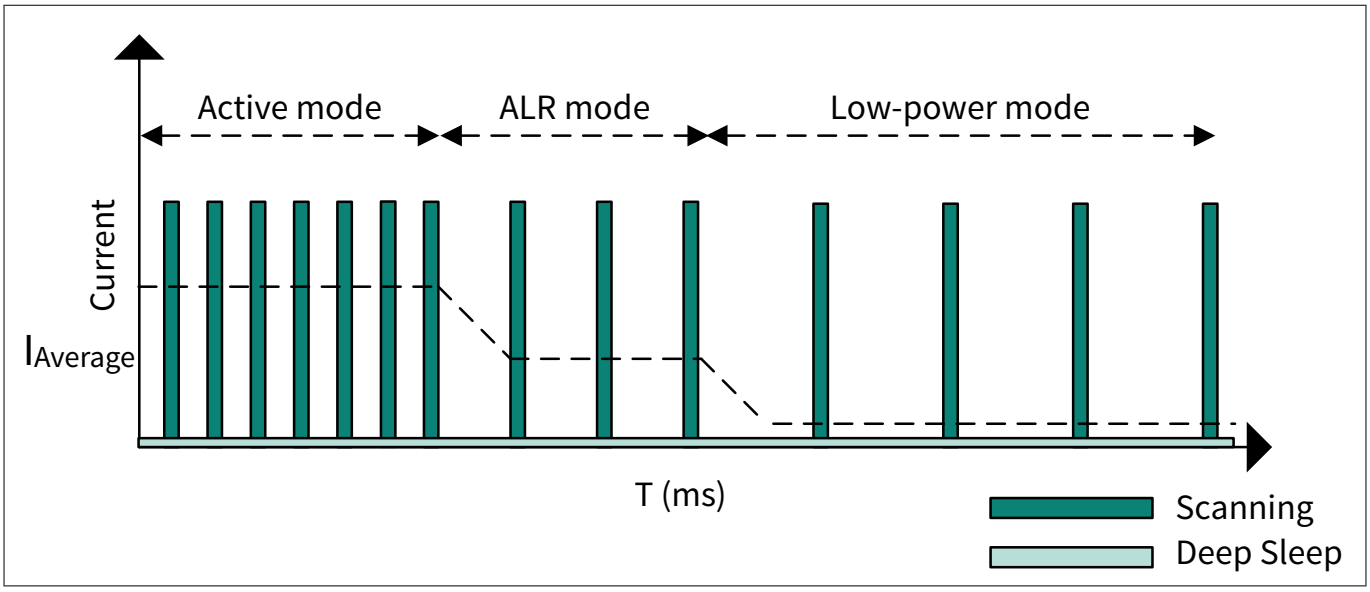


Figure 1 Current waveform in low-power mode

PSOC™ 4 introduces CAPSENSE™ MSCLP technology, an innovative way to decrease the current in low-power mode scanning by 5x to 10x (Figure 2) and improving the battery life of applications considerably. This technology has the wake-on-touch (WoT) mode that enables low-power touch detection without requiring the device to be active, i.e., to detect a touch while the device is in DeepSleep. MSCLP is also capable of scanning several sensors autonomously without any CPU intervention and does not need the device main clock for scanning; this reduces the current consumption in active mode as well.

Table 1 shows the maximum number of sensors supported for autonomous scanning. This varies based on the SRAM size of MSCLP and hardware IIR filter configurations.

Table 1 Maximum number of sensors - autonomous scan

Applies to devices	HW IIR filter state	MSCLP internal SRAM (bytes)	Max. sensors: Regular scan	Max. sensors: Low-power scan (WoT)
PSOC™ 4000T **PSOC™ 4100T Plus**	Disabled	1024	36	Not supported (N/A)

(table continues...)

Table 1 (continued) Maximum number of sensors - autonomous scan

Applies to devices	HW IIR filter state	MSCLP internal SRAM (bytes)	Max. sensors: Regular scan	Max. sensors: Low-power scan (WoT)
PSOC™ 4000T **PSOC™ 4100T Plus**	Enabled	1024	21	21

In regular scan (Active/ALR), the maximum number of sensors can be configured. The middleware will perform multiple autonomous scan frames one after another. Every autonomous frame scan completion will wake up the CPU from DeepSleep mode. Ensure that the application puts the CPU back into DeepSleep mode. Refer to Wake-on-Touch timeout for WoT frame size constraints.

In low-power scan (WoT), the number of sensors is limited to the maximum.

2.1 Power budgeting

The power budgeting is one of the main factors while designing low-power sensing solutions and is estimated using the required battery life and maximum battery capacity. The battery life means how long the end-product is required to operate without replacing batteries or recharging them; the maximum possible battery capacity is limited by form factor, size, and cost of the end-product.

The battery life is calculated using the average current for a typical use case using the following equation:

$$Battery\ Life = \frac{Maximum\ Battery\ capacity}{Average\ current\ consumption}$$

Equation 1 Battery life

The average current (power budget) for a battery-powered embedded system is calculated using the following equation:

$$Average\ current = \frac{(T_{active} * I_{active}) + (T_{low-power} * I_{low-power})}{(T_{active} + T_{low-power})}$$

Equation 2 Average current consumption

Where,

T_{active} – Total time in active power mode

I_{active} – Average current in active power mode

$T_{low\text{-}power}$ – Total time in low-power mode

$I_{low\text{-}power}$ – Average current in low-power mode

$I_{low\text{-}power}$ – Depends on DeepSleep current ($I_{deepsleep}$), low-power mode refresh rate, scan and processing current, and time. Consider an example when the device is in low-power mode state with the following parameters:

Low-power mode refresh rate = 16 Hz

Low-power mode period = 1/16 = 62.5 ms

Average scan current = 1 mA

Initialization and scan time = 120 µs

DeepSleep current = 1.65 µA

DeepSleep time = 62.5 ms – 0.12 ms = 62.38 ms

Then, low-power average current consumption is calculated using the following equation:

$$I_{\text{low-power}} = \frac{(62.38\ ms * 1.65\ \mu A) + (120\ \mu s * 1\ mA)}{(62.5\ ms)} = 3.57\ \mu A$$

Equation 3 Low-power average current consumption

2.2 PSOC™ 4 device with MSCLP is ideal for low-power solutions

PSOC™ 4 device with MSCLP provides 5-10x reduction in power consumption in low-power mode and 3-5x reduction in power consumption in active mode scanning; having the least power requirement in low-power mode is one of the major criteria for a wearable solution.

Here, comparing the power consumption of PSOC™ 4 with MSCLP devices with the previous CAPSENSE™ generation device (fourth-generation CAPSENSE™), Table 2 shows the exact improvement in different stages of application power modes.

Table 2 Power consumption comparison

Power mode	Power consumption			Condition	
	Fourth-generation CAPSENSE™	Fifth-generation MSCLP			
		PSOC™ 4000T	PSOC™ 4100T Plus	Fourth-generation CAPSENSE™ and PSOC™ 4000T	PSOC™ 4100T Plus
Wake-on-Touch (lowest power)	36 µA	3.9 µA[1)]	6 µA	10 buttons ganged to a total of 40 pF with shield enabled. Shield Cp = 39 pF Refresh rate = 16 Hz Sensitivity = 0.9 pF	10 buttons ganged to a total of 50 pF with shield enabled. Shield Cp = 67 pF Refresh rate = 16 Hz Sensitivity = 0.27 pF
Active – low refresh rate (intermediate power)	225 µA	55 µA	70 µA	13 buttons, each with electrode Cp = 4 pF with shield enabled. Shield Cp = 39 pF Refresh rate = 32 Hz Sensitivity = 0.4 pF	13 buttons, each with electrode Cp = 4 pF with shield enabled. Shield Cp = 67 pF Refresh rate = 32 Hz Sensitivity = 0.16 pF

(table continues...)

Table 2 (continued) Power consumption comparison

Power mode	Power consumption			Condition	
	Fourth-generation CAPSENSE™	Fifth-generation MSCLP			
		PSOC™ 4000T	PSOC™ 4100T Plus	Fourth-generation CAPSENSE™ and PSOC™ 4000T	PSOC™ 4100T Plus
Active – high refresh rate (active power)	750 µA	206 µA	270 µA	13 buttons, each with electrode Cp = 4 pF with shield enabled. Shield Cp = 39 pF Refresh rate = 128 Hz Sensitivity = 0.4 pF	13 buttons, each with electrode Cp = 4 pF with shield enabled. Shield Cp = 67 pF Refresh rate = 128 Hz Sensitivity = 0.16 pF

1) Measured with the kit having DeepSleep current of 1.7 µA. Expected ~4.7 µA if the kit has typical DeepSleep current of 2.5 µA.

2.3 Factors affecting low-power operation

When designing a low-power application, you may need to consider some trade-offs such as scan duration, signal-to-noise ratio (SNR), refresh rate, and number of sensors are scanned while in DeepSleep. As shown in Figure 2, while in low-power mode, the scanning stage is the major contributor for power consumption. The scan duration and refresh rate directly impact the power consumption.

2.3.1 Scan duration

The scan duration depends on the parasitic capacitance (Cp) of the sensor because the maximum scan frequency is limited by Cp. A lower sensor Cp reduces the scan duration, which in turn lowers the power consumption. See Low-power IIR filter to understand techniques used to reduce the sensor Cp.

2.3.2 Refresh rate

The refresh rate should be decided based on the user experience and power consumption (see Section Refresh rate). Having a lower refresh rate reduces the power requirement but it will negatively impact the response performance. In most use cases, low-power sensors are used for waking up the device; a lower refresh rate means that the user needs to press the button a little longer to wake up the application.

2.3.3 Signal-to-noise ratio (SNR)

Another contributing factor for scan duration is the signal-to-noise ratio (SNR); noise can be reduced using built-in IIR filters (see Section Scan duration), therefore, increasing the SNR to the requirement without increasing the scan time.

2.3.4 Number of sensors in low-power mode

As mentioned in section Scan duration, having a low-Cp sensor in low-power mode reduces the power consumption; therefore, to have a dedicated wakeup button would be ideal. If there are multiple sensors that

need to be scanned for a specific application (such as a touchpad), ganging them all together will help in reducing the power consumption. See Table 1 for the maximum number of sensors supported.

For example, scanning five sensors together (ganged sensor) requires less power than scanning them separately even though the combined Cp scanned in low-power mode is the same. This is because separate scanning requires separate initialization.

2.4 Power estimation

To get a better estimation and identify areas to improve power consumption, it is important to know the power consumption at each stage of scanning such as – initialization, scanning, and processing.

2.4.1 Current consumption at different scanning stages

For each user power mode (see Section Power consumptions in different application states), based on the scanning stage, the current consumption varies:

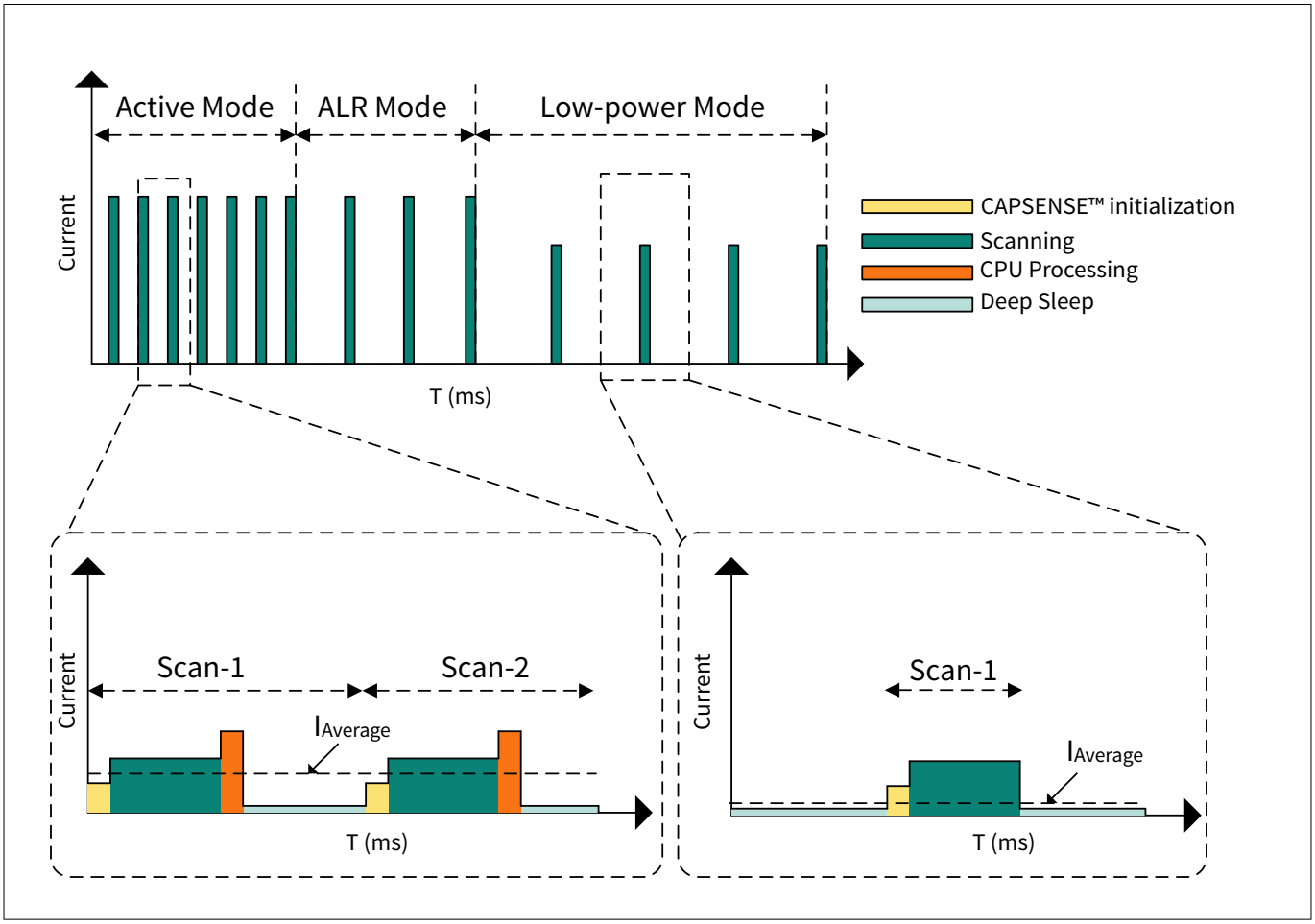


Figure 2 Active mode scanning stages

- **Active power mode:** Scanning stages consist of CAPSENSE™ hardware and firmware initialization, scanning, processing, and DeepSleep (Active mode in Figure 2). Note that the PSOC™ 4 device is in DeepSleep mode in all the stages except processing.
- **Low-power mode:** Scanning stages consist of CAPSENSE™ hardware initialization, scanning, and DeepSleep (Low-power mode in Figure 2)

Even if the scan duration is the same in active and DeepSleep modes, current consumption varies because of the low refresh rate and reduction in scanning stages.

3 Firmware design considerations

3.1 Power modes available in PSOC™ 4 device with MSCLP

The low-power modes are categorized as follows:

- Application (user) power states
- CAPSENSE™ (MSCLP) hardware-level power modes

3.1.1 Power consumptions in different application states

The application power modes define different states to scan the sensors based on user activity. You can define any number of states according to the application requirement to reduce the overall power consumption. Each state consumes only the minimum required power that is specific to that application; three modes are considered to cover most of the use cases for MSCLP solutions:

- **Active**: To have highest refresh rate while the user is actively using the solution.
- **Active low refresh rate (ALR)**: To have scanning at a low refresh rate on all sensors, this is an optional power mode and can also have multiple ALR mode based on application requirement. This acts as an intermediate state while transferring from the active to Wake-on-Touch state based on reduced user activity. This mode can also be used for periodically updating baselines of the sensor when there is no user activity for a long time.
- **Wake-on-Touch (WoT)**: The lowest-power mode, which scans a reduced number of sensors at a low refresh rate and processes the results without CPU intervention. The reduced list of sensors can be a button such as ‘wakeup or power-up’, proximity sensors, ganged sensors, etc. PSOC™ device can be kept at DeepSleep (both CPU and HFCLK is OFF) in this mode while MSCLP is scanning. A low-power interrupt is generated by the MSCLP hardware when there is a user-activity, or a timeout detected.

Figure 3 shows the different power modes and transition conditions for a typical use case. The section Firmware techniques for low-power design provides details on implementing different application power modes using CAPSENSE™ middleware and section Low-power CAPSENSE™ example provides a low-power code example and the firmware flowchart showing different user power modes and mode transition based on user activity.

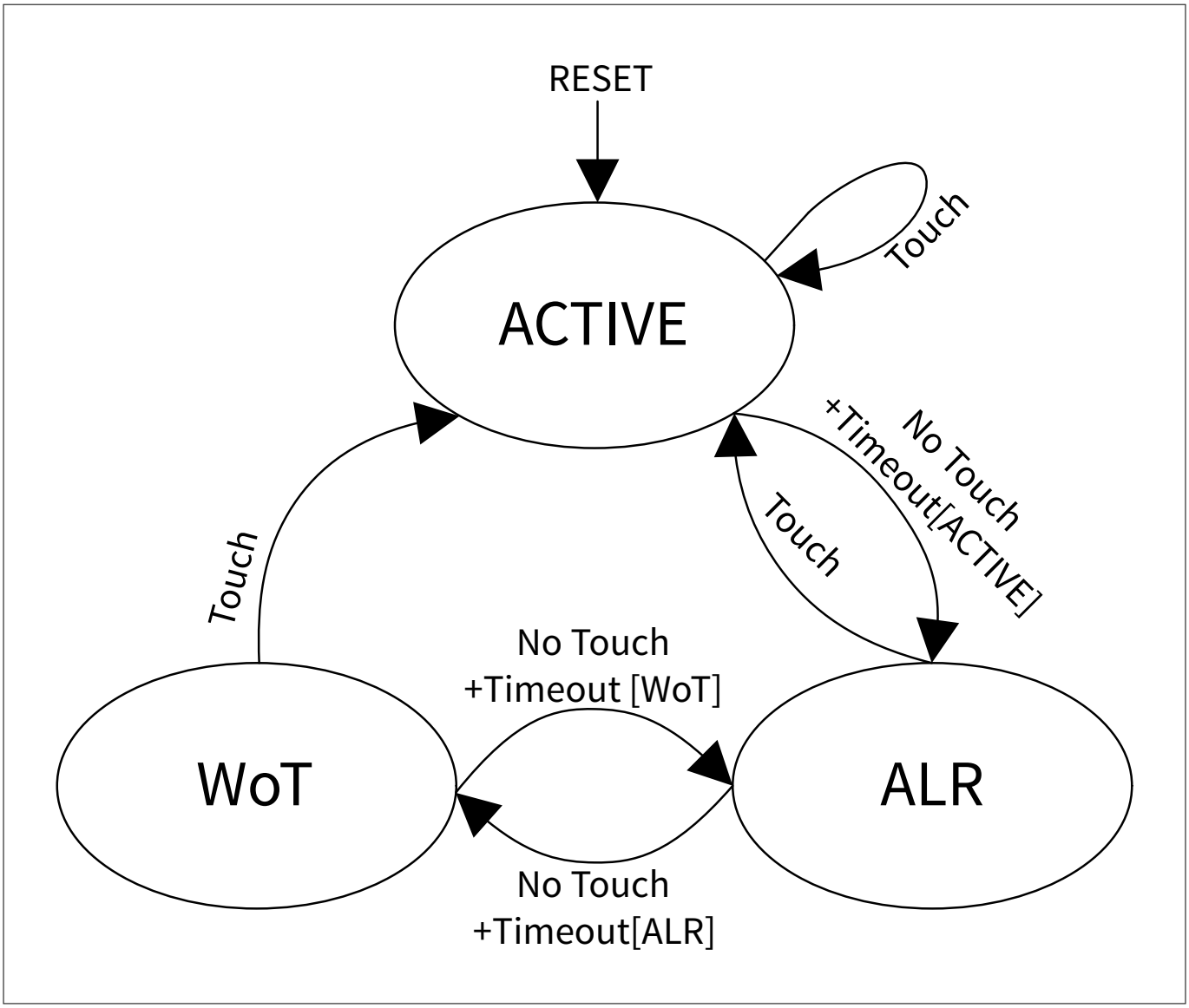


Figure 3 Application Power mode transition

3.1.2 CAPSENSE™ hardware scanning modes

The MSCLP CAPSENSE™ hardware supports two hardware scanning modes:

- CPU (legacy)
- Low-power, always-on sensing (LP-AoS)

CPU (legacy) mode: CPU is the only mode supported in the earlier generation CAPSENSE™ (fourth-generation and earlier), where each sensor needs to be configured by the CPU before each sensor scan. In this mode, after initiating the scan for the first sensor, at the end of each sensor scan completion, the CPU is interrupted to read the results, and to configure and initiate the next sensor scan. This mode is not used in PSOC™ 4 device with MSCLP for typical scans and only used for CAPSENSE™ initialization, auto-calibration, and built-in self-test(BIST) by CAPSENSE™ middleware.

Low-power always-on sensing (LP-AoS): Multiple sensors (a frame) can be scanned without the intervention of the CPU. In PSOC™ 4 device with MSCLP, a 1KB SRAM is available inside the MSCLP hardware; the middleware writes the configuration of multiple sensors to be scanned to the internal SRAM. Then, the MSCLP autonomously reads each sensor configuration, initiates scanning, and saves the result to the SRAM for the middleware to read and process after full-frame scanning is completed.

Additionally, this mode has the capability of scanning the same frame periodically and processing the result to detect touch. Therefore, the CPU is not required to process or initiate scanning; this allows the PSOC™ MSCLP device to be kept in DeepSleep for a longer time and to have the sensor touch detection as a wakeup interrupt. Therefore, the application can get the lowest power consumption with touch-sensing capability.

The device can be kept in any device power mode such as Active, Sleep, or DeepSleep while MSCLP is scanning multiple sensors. MSCLP has all the system resources such as high-frequency clock and reference generator internal to the MSCLP hardware to enable scanning in device low-power state (i.e., DeepSleep).

Table 3 shows the user power modes for a typical use case.

Table 3 Recommended MSCLP power modes based on user application

#	Application power modes	Scope	Refresh rate	Power consumption	Optimum device power mode
1	Active (max refresh rate)	All sensors	High	High	**Active mode:** To trigger scan and process **DeepSleep mode:**[1)] For frame scan and waiting (to achieve required refresh rate).
2	Active - low refresh rate (ALR)	All sensors	Low	Medium	**Active mode:** To trigger scan and process **DeepSleep mode:** For frame scan and waiting (to achieve required refresh rate).
3	Wake on touch (WoT)	A few sensors or ganged sensors	Very low	Low	**DeepSleep mode:** To scan, process, and wait.

1) Refer to the CPU power modes section for more details on different PSOC™ 4 MCU power modes and resource availabilities.

CPU mode is not required for sensor scanning; it is used for CAPSENSE™ initialization, auto-calibration, and built-in self-test (BIST) by CAPSENSE™ middleware. To get the least power consumption on all application power modes, LP-AoS is best suited and is the default. However, LP-AoS mode always power cycles (switches ON and OFF) the MSCLP system resource between each frame scan; this can reduce the maximum refresh rate.

3.2 Firmware techniques for low-power design

The main objective of firmware techniques is to stay in a low-power mode as long as possible by reducing the scan time and refresh rate, while still ensuring a reliable operation and the best user experience.

3.2.1 Regular widget

As mentioned in the Low-power always-on sensing (LP-AoS) section, MSCLP enables scanning while the device is in DeepSleep mode. This can reduce the power consumption while the device is in Active mode.

Code Listing 1 shows how to scan all the regular widgets while the device is in DeepSleep mode.

Code Listing 1

```
Cy_CapSense_ScanAllSlots(&cy_capsense_context);
while (Cy_CapSense_IsBusy(&cy_capsense_context))
{
    Cy_SysPm_CpuEnterDeepSleep();
}
```

3.2.2 Low-power widget

MSCLP has a new widget called "Low-Power Widget" to scan sensors in wake-on-touch (WoT) mode. A low-power sensor that is required to be scanned while in low-power mode must be configured as the low-power widget (see Figure 4). These widgets are scanned and processed without any CPU intervention; they can act as a wakeup source for the device.

***Note**: For the normal operation of these buttons in active mode, these sensors need to be additionally configured as normal widgets such as button, proximity, slider, and touchpad. Buttons such as power-on/wakeup, ganged sensors, and proximity sensors are examples of a low-power widget.*

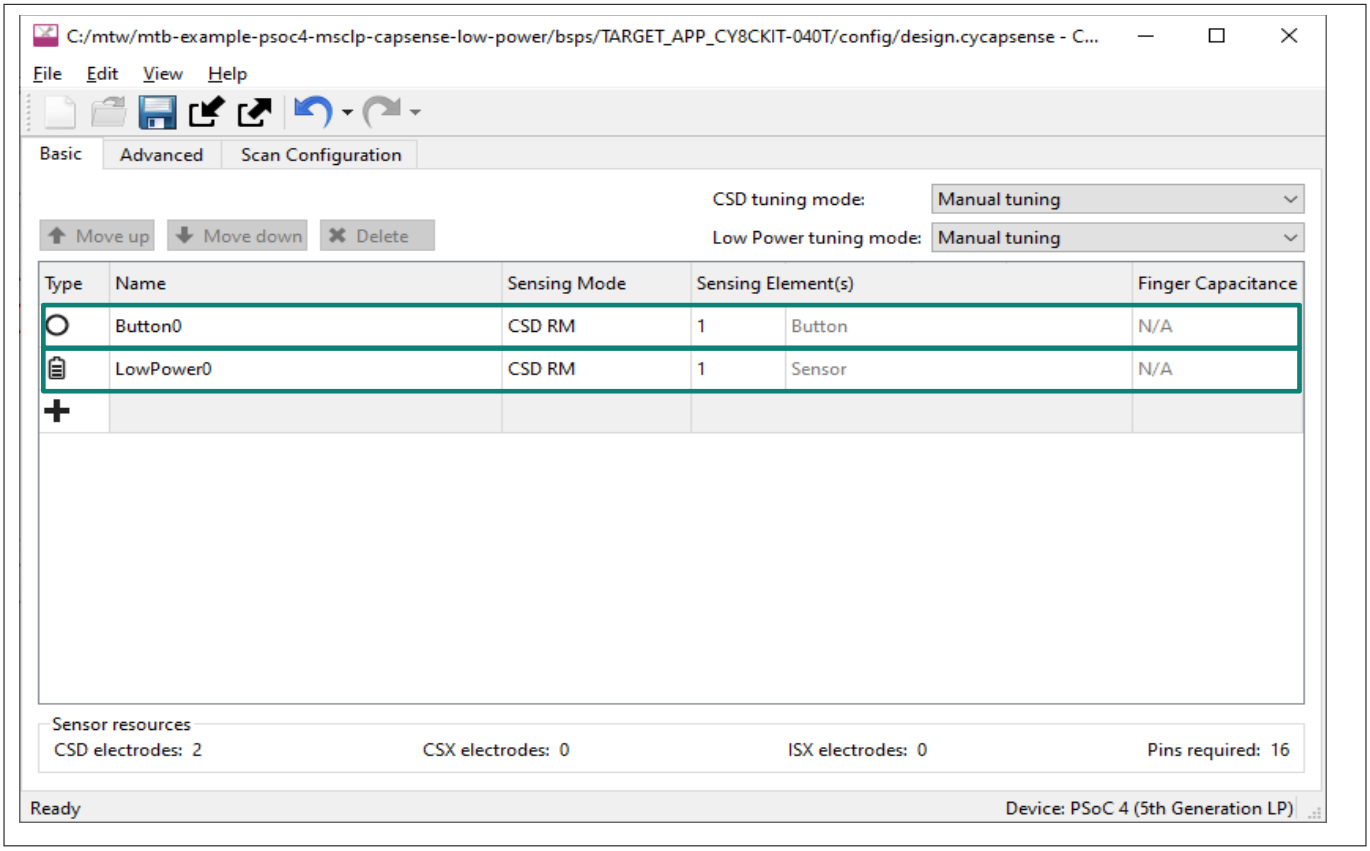


Figure 4 Add low-power widget

To scan an LP widget in Wake-on-Touch (WoT) mode, there are new APIs added to CAPSENSE™ middleware; see Code Listing 2.

Code Listing 2

```
Cy_CapSense_ScanAllLpSlots(&cy_capsense_context);
while (Cy_CapSense_IsBusy(&cy_capsense_context))
{
    Cy_SysPm_CpuEnterDeepSleep();
}
```

The `ScanAllLpSlots` API scans all the configured low-power widgets based on Low-power widget parameters. See Section CIC2 filter for a complete low-power code example.

3.2.3 Refresh rate

As shown in Figure 2, there are many stages in different scan modes; the time spent in scanning has the maximum impact on power consumption. The amount of time for scanning sensors depends on the refresh rate and sensor frame scan time, as shown in Figure 5. As the refresh rate reduces, power consumption decreases but increases the response time. Therefore, the refresh rate should be decided based on the user requirement and power consumption.

For active mode, users expect to have an immediate response (maximum refresh rate). With a refresh rate of 128 Hz, the sensors are scanned every 7.8 ms; this can detect a touch in approximately 25 ms (considering debounce = 3 and no software filter; see AN85851 - PSOC™ 4 and PSOC™ 6 MCU CAPSENSE™ design guide for more details), which is in typical range of human response time. Therefore, any refresh rate above 128 Hz will provide a good user experience.

For low-power mode (WoT), the objective is to wake up the complete system on a touch event with the lowest power consumption possible. Any typical low-power solution tends to have a minimum of 500 ms long press to wake up the system. But MSCLP scanning at 16-Hz refresh rate can wake the system and report a touch event in less than 200 ms (considering debounce = 1 and default hardware filter settings).

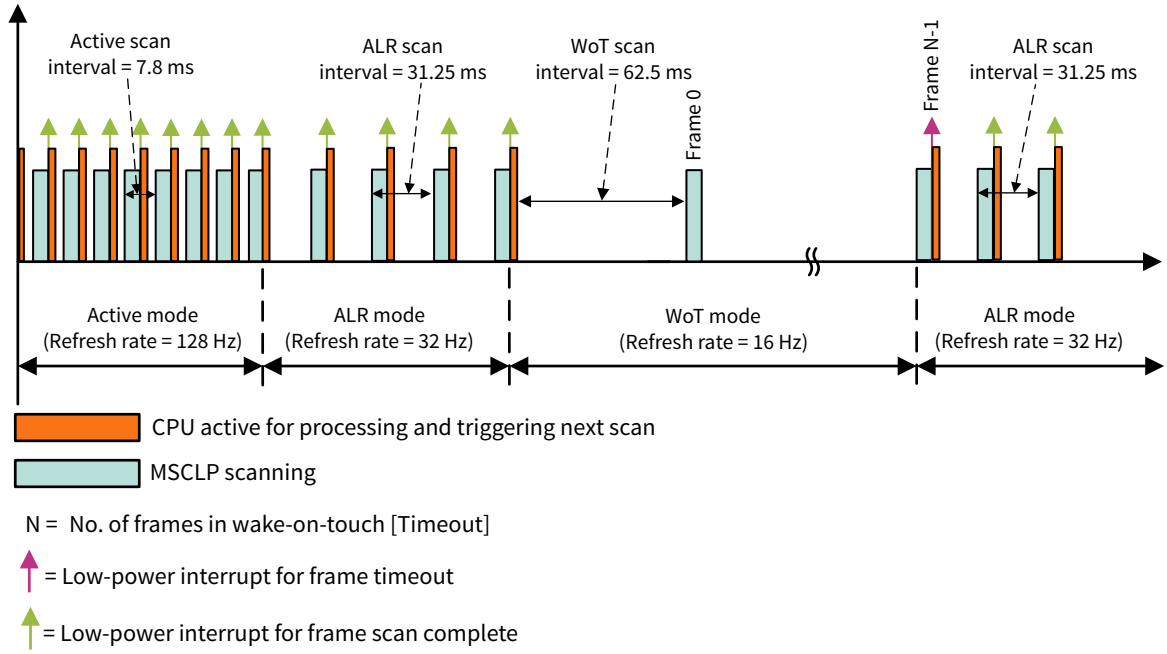


Figure 5 Refresh rates in different power modes

3.2.3.1 MSCLP timer

The CAPSENSE™ MSCLP hardware has a built-in low-power timer (MSCLP timer), which uses the internal ILO as a clock source. This timer is used to control the refresh rate in different power modes by introducing a delay at the start of the single-frame scan.

The usage of the MSCLP wakeup timer is shown in the following code snippet:

Code Listing 3

```
for (; ;)
{
Cy_CapSense_ConfigureMsclpTimer(wakeupTimer, &cy_capsense_context);
Cy_CapSense_ScanAllSlots(&cy_capsense_context);
interruptStatus = Cy_SysLib_EnterCriticalSection();
while (Cy_CapSense_IsBusy(&cy_capsense_context))
{
Cy_SysPm_CpuEnterDeepSleep();
Cy_SysLib_ExitCriticalSection(interruptStatus);

/* This is a place where all interrupt handlers will be executed */
interruptStatus = Cy_SysLib_EnterCriticalSection();
}
Cy_SysLib_ExitCriticalSection(interruptStatus);

Cy_CapSense_ProcessAllWidgets(&cy_capsense_context);
}
```

The MSCLP block starts the scan of all slots (one frame) after the delay (wakeupTimer) set by the MSCLP timer configuration API. See Figure 6.

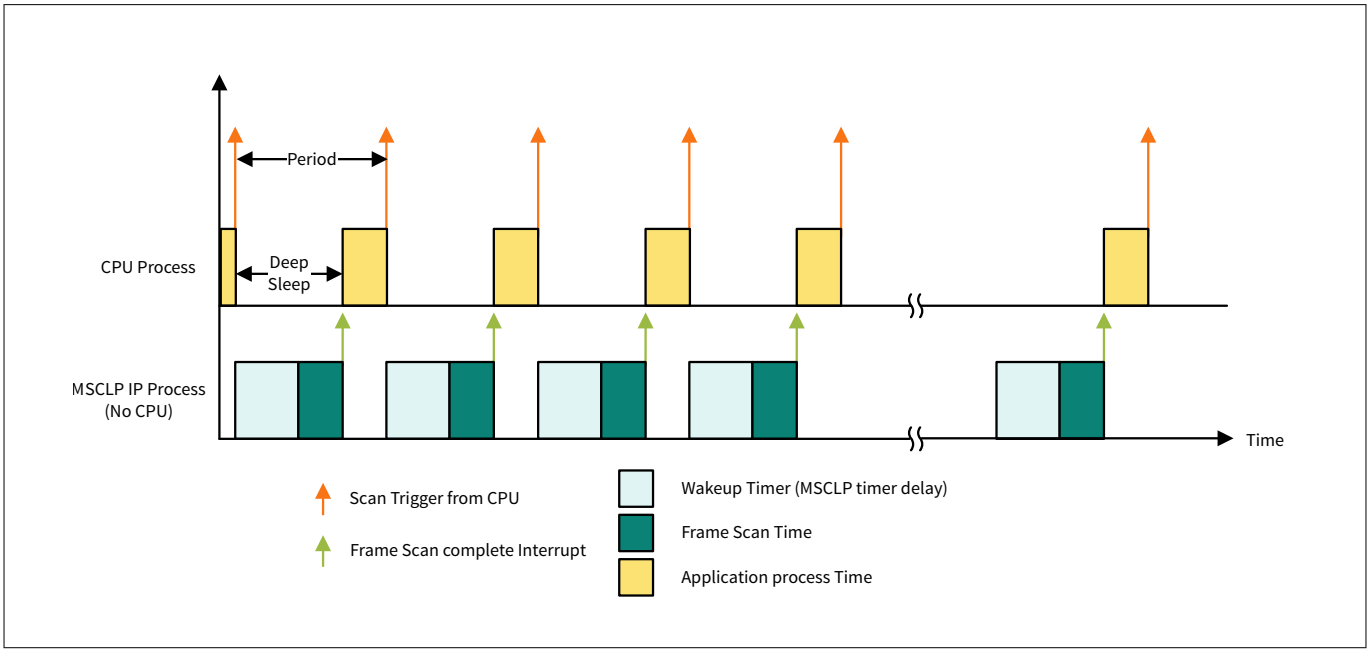


Figure 6 MSCLP timer and refresh rate

Therefore, refresh rate is calculated using the following equation:

$$Period = \text{wakeupTimer} + Scan\ time + Application\ process\ time$$

$$Refresh\ rate = \frac{1}{Period}$$

Equation 4 Refresh rate

3.2.4 Reducing the scan time in low-power mode

As mentioned in the previous section, the total scan time and refresh rate contribute to the power consumption. To reduce the total scan time, one way is to reduce the number of sensors, thereby reducing the Cp.

3.2.4.1 Wakeup/power-on button

Many low-power solutions are designed to have a wakeup or power-on button. Only this sensor needs to be scanned in low-power mode; a touch input from any other sensor is ignored. Limiting the number of low-power sensors will reduce the total Cp and the scan time, this in turn reduces the power consumption.

3.2.4.2 Ganged sensors

Ganging multiple sensors help to reduce the power consumption if the requirement is to have the complete touch area as a wake-up source while in a low-power mode, for applications such as smart watches. To cover the complete touch area, all sensor electrodes should be scanned. Even though the total Cp that should be scanned for all the sensors is the same, it is beneficial to gang all sensors as one and perform a single scan because each individual sensor scan needs additional time for initialization, configuration, and processing.

3.2.4.3 Proximity sensors

Instead of ganged sensors, which require the full sensors to be scanned in low-power mode, proximity sensors can be used. Here, the device can wake up while the hand is approaching to the touch region; this can have a better user experience. As the approach of the hand itself wakes the system and the first touch will be considered for normal operation, the customer experience is identical to that of the device always being in active mode.

3.2.4.4 Touchpad-specific techniques

Solutions with touchpad which need the complete touchpad to be responding to touch while in DeepSleep pose additional challenges because the total Cp of the complete touchpad is high.

1. **Using self-capacitance scanning method for mutual-capacitance-based touchpad**

   A mutual capacitance (CSX) touchpad has several sensors compared to a self-capacitance (CSD) touchpad with the same dimensions and number of electrodes. For example, a 5 x 4 CSX touchpad (Figure 7) has 20 individual sensors to be scanned whereas in the CSD method, it is just 9 sensor scans. The number of sensors to be scanned in low-power mode can be further reduced by sensing only the row or column sensors as highlighted in Figure 8.

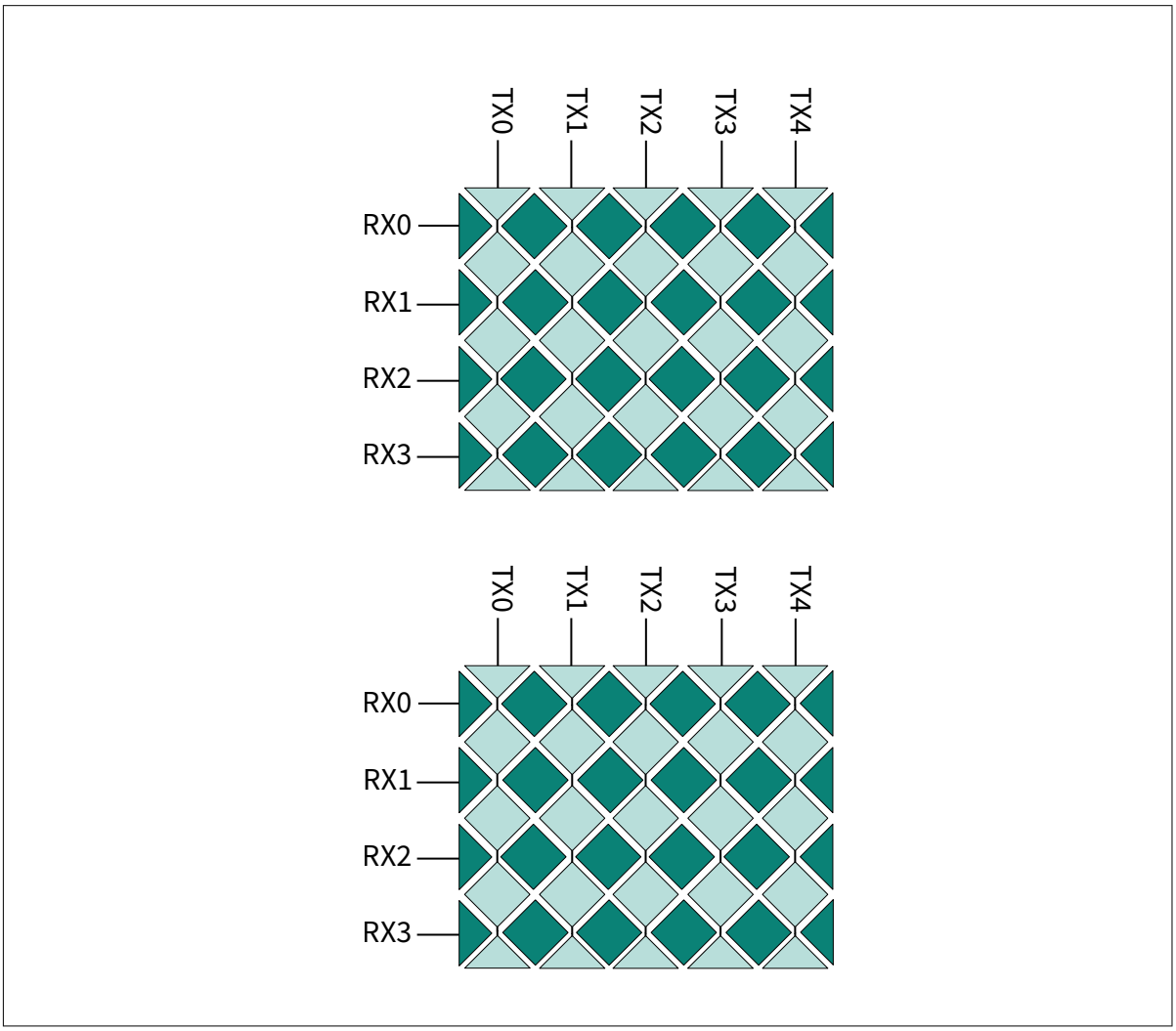


Figure 7 5 x 4 CSX touchpad

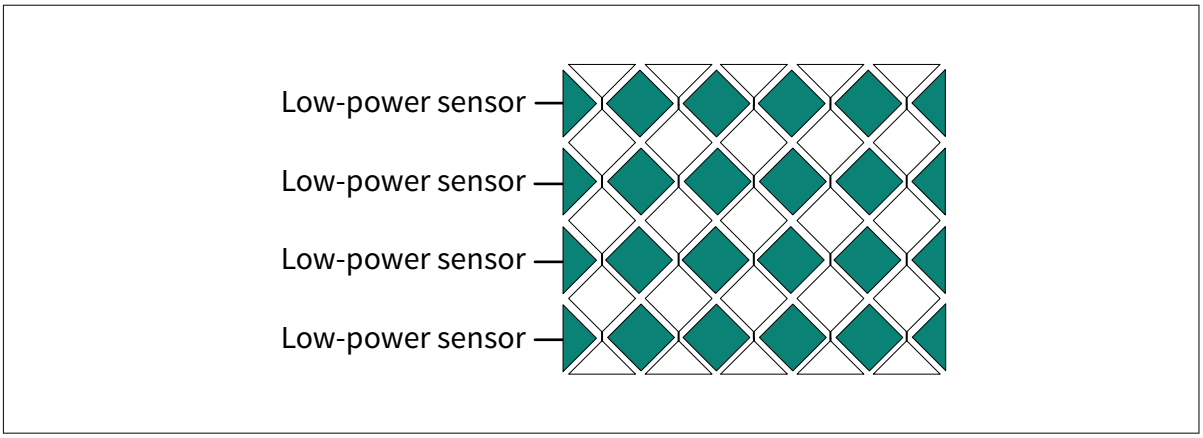


Figure 8 CSX touchpad configuration in low-power mode

By sensing the row or column sensors, the total Cp to be scanned in low-power mode can be further reduced. In addition, these sensors can be ganged to a single low-power widget to get the least power consumption.

2. Configure outer electrodes as a proximity loop

A touchpad with many electrodes will still have a high Cp to be scanned in low-power mode. In this case, configuring the outer electrodes as a single proximity sensor, reduces the total Cp scanned in low-power mode and provides the additional feature of waking up on proximity detection. Figure 9 shows the configuration of a proximity loop using outer electrodes in a 6x8 touchpad.

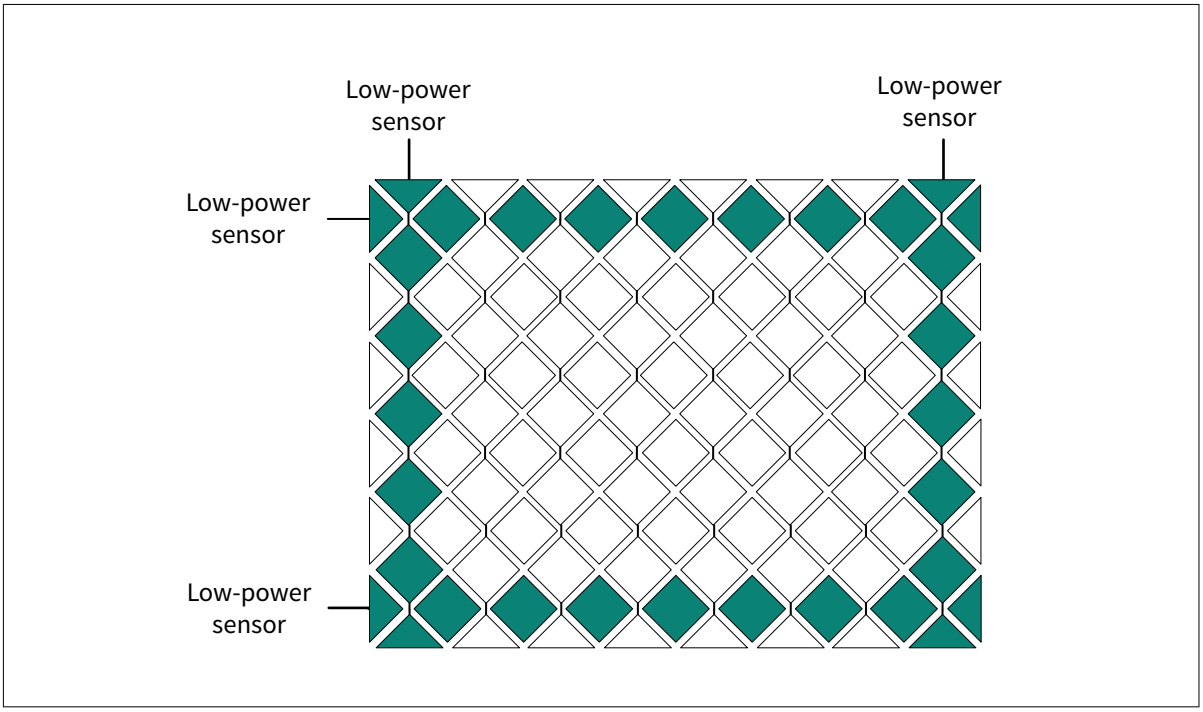


Figure 9 6 x 8 touchpad configured as a proximity loop

3.2.5 Shield design

For the CSD sensing method, a driven shield electrode can provide benefits such as liquid tolerance, and reduction in sensor Cp. Shield electrodes can be used in low-power design to reduce the power consumption because they decrease the sensor Cp.

A typical shield design consists of hatch patterns around the sensors in the top layer and directly beneath the sensors on bottom layers. For the low-power design, a driven shield can reduce the power consumption by

reducing the sensor Cp but can increase the power consumption because the shield electrodes need to be additionally driven. In addition, a higher shield Cp can adversely affect power consumption as the maximum possible sense clock frequency will be limited by a large shield Cp. This increases the power consumption due to additional scan time requirement.

To avoid this, split the shield for Low-power and Active modes by configuring multiple pins as the shield, which is connected to independent shield regions. Design one of the shield regions specific to the low-power sensor (see Hardware design considerations). While in Low-power mode, configure this shield region/pin to be driven with the shield signal and connect the remaining shield to ground. This can reduce the sensor Cp, therefore reducing the overall power requirement in Low-power mode.

3.3 Low-power widget parameters

As mentioned in section Regular widget, low-power widgets need to be used to design low-power designs, because they can provide advanced low-power features such as scan and process sensors while the CPU in DeepSleep. This section explains different parameters specific to low-power widgets.

3.3.1 Wake-on-touch scan interval

The scan interval is the time between two subsequent frame scans. The CAPSENSE™ MSCLP hardware contains a built-in low-power timer (MSCLP timer) which triggers the next frame scan while in WoT mode. This can be used to control the scan refresh rate while the device in WoT mode by appropriately providing the scan interval in microseconds (see Figure 12). This timer uses the ILO clock; you can configure the scan interval in microseconds for the required refresh rate.

***Note**: If the frame scan time is bigger than the set scan interval, the next frame scan will start after the completion of the current frame.*

3.3.2 Wake-on-Touch timeout

While the device in WoT mode, the MSCLP hardware scans the low-power widget at set refresh rate and wakes up the device if there is a touch (see Figure 10) or if there is a timeout (see Figure 11). A timeout denotes the maximum number of frame scans while the device is in WoT mode before waking the device and moving to active mode.

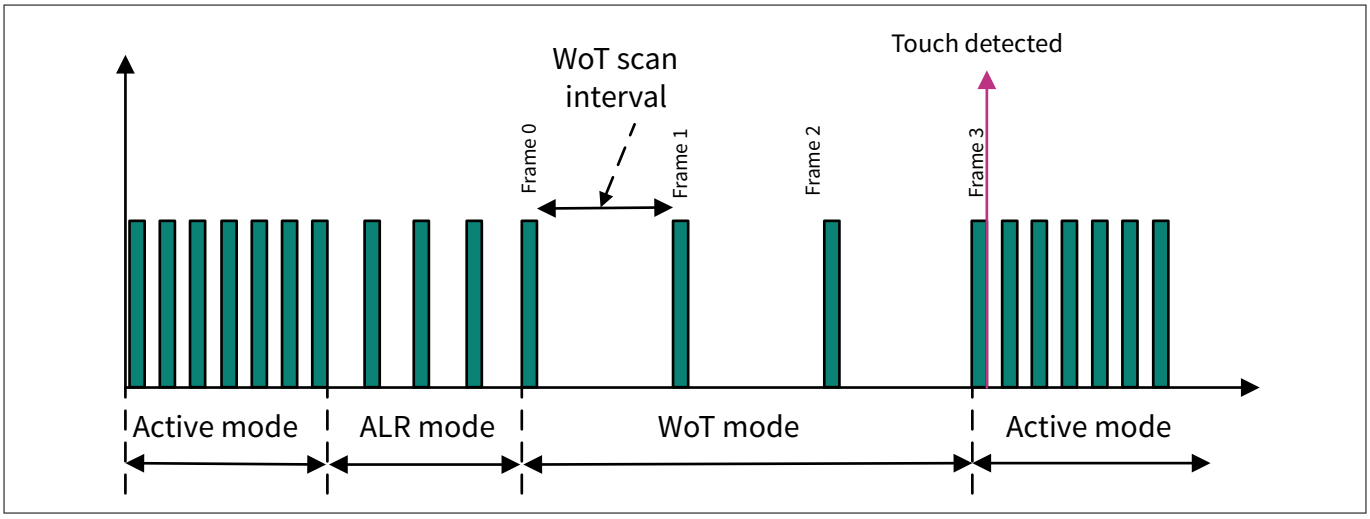


Figure 10 WoT wakeup due to touch detection

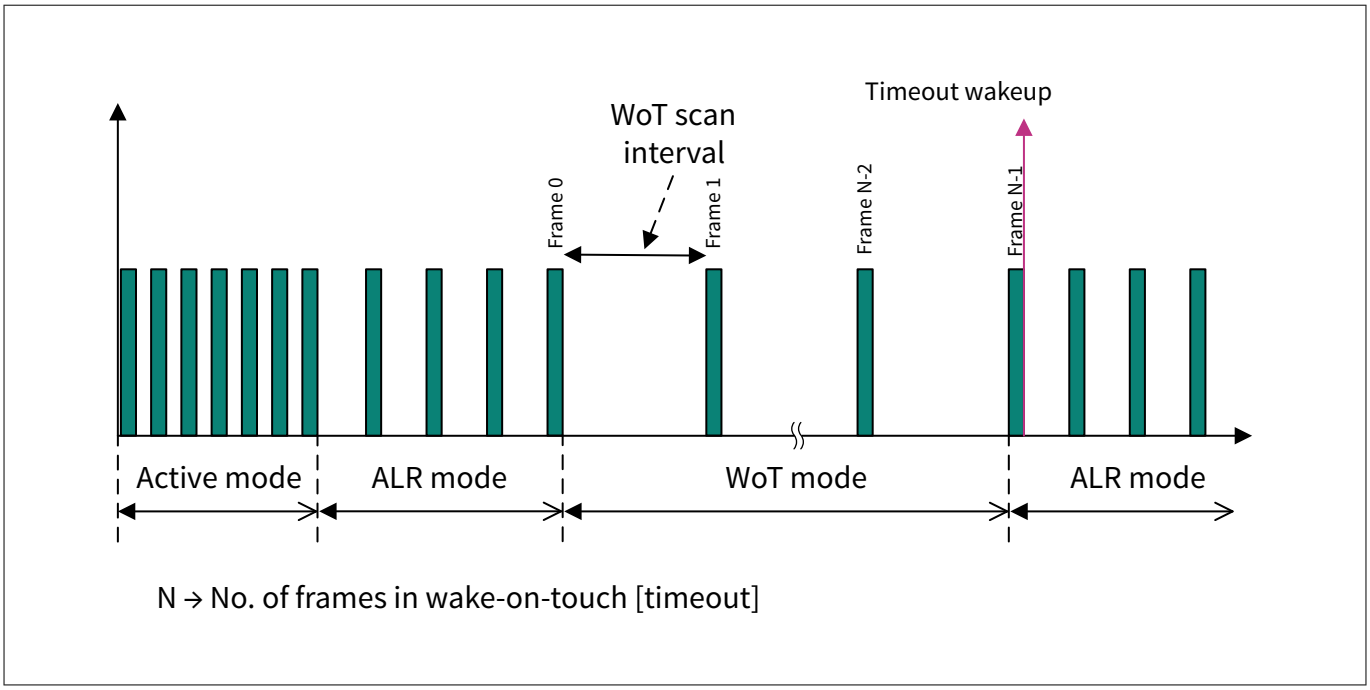


Figure 11 WoT wakeup due to timeout

The timeout is required because the regular widget needs to be scanned and the baseline needs to be updated periodically to avoid drifting of the baseline because of environment variations such as temperature. The maximum timeout can be calculated using the following equation:

$$WoT\ Timeout = WoT\ scan\ interval * Number\ of\ frames\ in\ WoT$$

Equation 5 WoT timeout

The internal SRAM of the MSCLP hardware acts as a FIFO to store the scan results (raw count). These stored raw counts are useful while tuning the low-power widgets or can also be used for any custom processing. The maximum number of raw count results which can be stored depends on the maximum internal memory size and number of low-power sensors; see Table 4.

Table 4 Maximum number of raw count results stored in MSCLP SRAM

Number of low-power sensors	Maximum number of raw count results in SRAM
1	245
2	117
3	74
4	53
5	40
6	31
7	25
8	21

The FIFO logic has a limitation. It is not a circular buffer, stops storing the raw count results of the scan frames once it is full.

If the number of frames in WoT mode exceeds the maximum number of raw count results as shown in Table 4, the SRAM can hold only the first set of scan results, which is up to the maximum number of raw count results because of the limitation of SRAM storage. Any subsequent raw counts beyond this limit are ignored. However, it does not impact the touch detection and wake-up functionality of the WoT mode.

The CAPSENSE™ tuner can calculate the baseline using the MSCLP algorithm to recreate the baseline while the device is scanning in DeepSleep mode. This can only work correctly if the **number of frames in WoT** is less than or equal to **maximum number of raw count results in SRAM**. Ensure that the application must be configuring the low-power widgets.

3.3.3 Low-power IIR filter

To detect a touch in WoT mode, MSCLP contains a hardware engine which runs the baseline update and touch detection algorithm. This works like CAPSENSE™ middleware processing, which applies an IIR filter to the raw count and baseline, do signal calculation, and touch detection.

An IIR filter acts as a low-pass filter applied to the raw count passing the low-frequency signal (finger touch responses) and blocking high-frequency noise signal. Therefore, SNR will be improved for a given scan configuration; in other words, the same SNR can be achieved with a lower scan time or lower power consumption.

$$RawCount = \frac{1}{2^{iirRCcoef}} RawCount_{New} + \left(1 - \frac{1}{2^{iirRCcoef}}\right) RawCount_{Previous}$$

Equation 6 IIR filter applied on raw count

Where,

iirRCcoef – IIR filter raw count coefficient; valid range: 1 to 8. A lower coefficient means lower filtering; a higher coefficient means a higher response time.

The baseline IIR filter uses two different coefficients ($iirBLcoef_{fast}$ and $iirBLcoef_{slow}$) for low-power widgets, based on the signal level. When the raw count starts to increase, the baseline value is updated quickly to attempt to track the raw count using $iirBLcoef_{fast}$. However, once the noise threshold is exceeded, the baseline is updated slowly (using $iirBLcoef_{slow}$) if the raw count is increasing because of a touch or signal event. The baseline can be calculated using the following equation:

$$Baseline = \frac{1}{2^{iirBLcoef}} RawCount_{New} + \left(1 - \frac{1}{2^{iirBLcoef}}\right) Baseline_{Previous}$$

Equation 7 IIR filter applied on baseline

Where,

$iirBLcoef_{fast}$ – IIR filter baseline coefficient for fast update; valid range: 4 to 8.

$iirBLcoef_{slow}$ – IIR filter baseline coefficient for slow update; valid range: 4 to 8.

Figure 12 highlights the low-power widget-specific parameters in CAPSENSE™ Configurator.

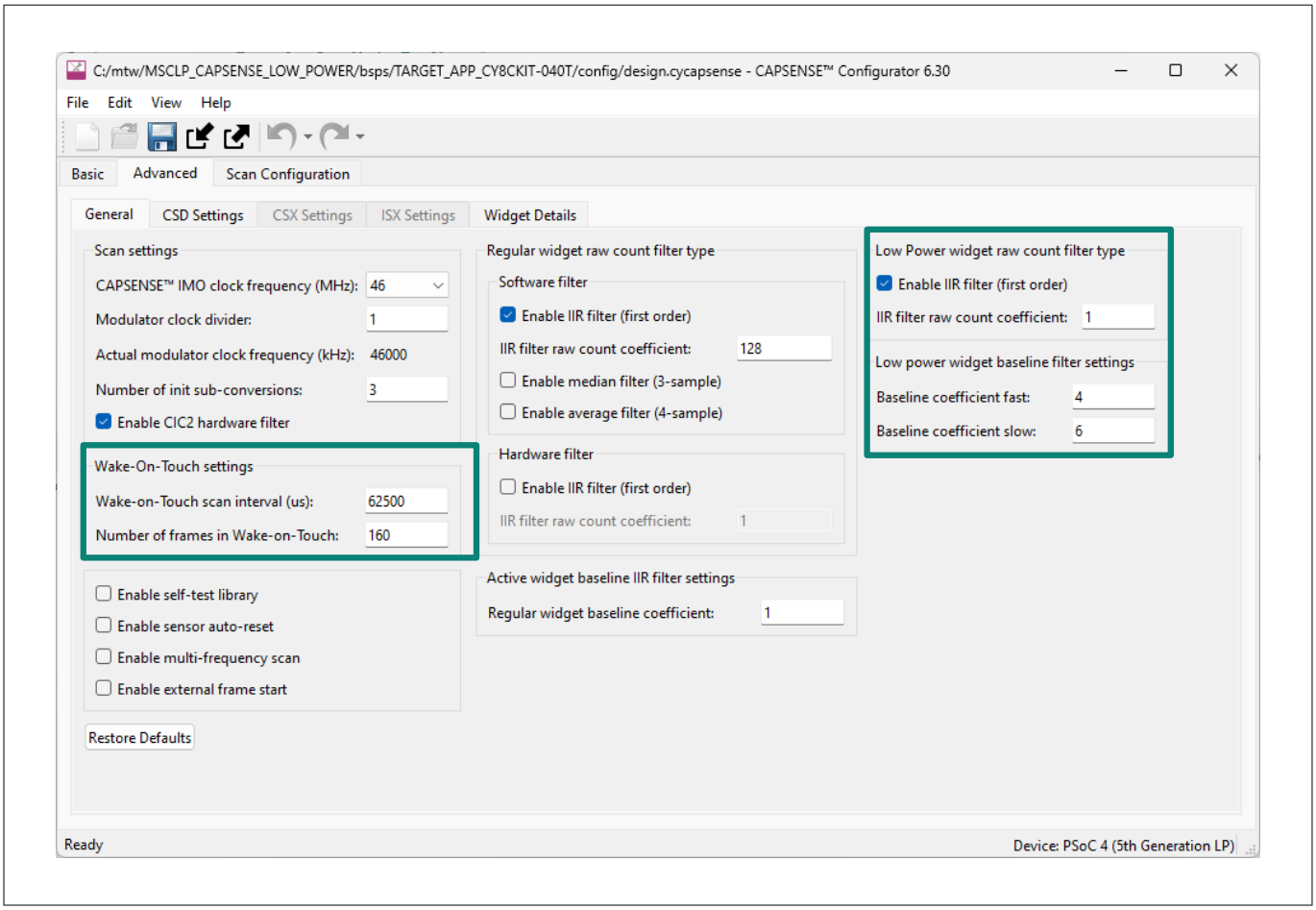


Figure 12 Low-power widget parameters

3.4 CIC2 filter

The CAPSENSE™ MSCLP has a built-in cascaded integrator-comb 2 (CIC2) digital filter which improves the effective resolution, and thereby the SNR, for a given scan period. CIC2 filter is a 2nd order digital low-pass (decimation) filter used to filter delta-sigma converters. Figure 13 shows the representation of a CIC2 filter made up of a cascade of two integrators and two comb filters.

CIC2 filter receives the output of the MSCLP analog front-end which is a delta-sigma convertor. This converter generates a bitstream of 1s and 0s representing its input and moves the quantization noise to high frequencies. This high-frequency noise is filtered out by a digital low-pass filter; the down-sampler converts the input to a single digital word representing the measured signal. This combination of a low-pass filter (A0, A1) and a down-sampler (D0, D1) is known as a decimator (CIC2 filter) as shown in Figure 13.

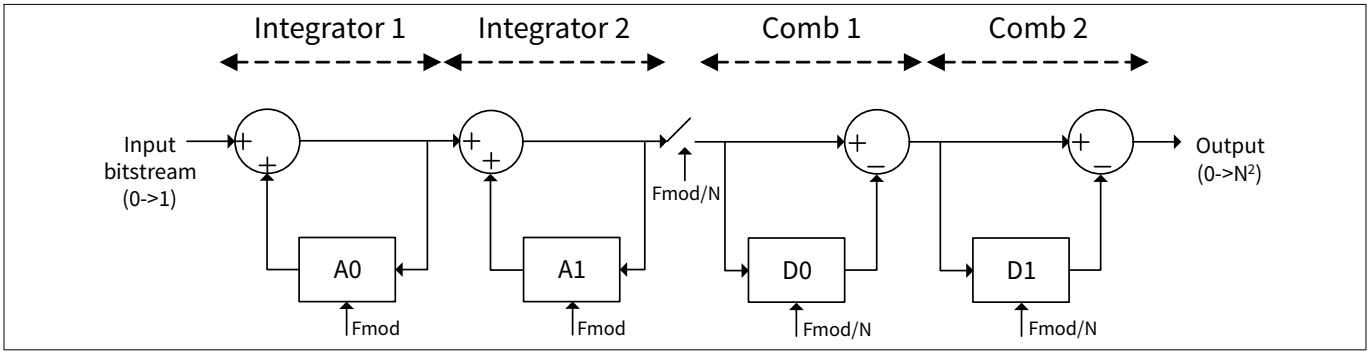


Figure 13 CIC2 filter block diagram

The raw count gets accumulated at the end of each valid sample.

A minimum of two valid CIC2 samples are required for proper CIC2 filtering. Considering this the decimation(down-sampling) rate is calculated using the following equation:

$$Decimation\ rate\ (N) = \frac{Sns_Clk_Div\ *Nsub}{3}$$

Equation 8 Decimation rate

Where,

Nsub – Number of Sub conversions

Sns_Clk_Div – Sense Clock Divider

Configure the CIC2 Shift parameter as "Auto" in CAPSENSE™ configurator. This automatically selects the appropriate hardware shift (hardware divider).

The max raw count, when CIC2 is enabled, is calculated using the following equation:

$$Rawcount_{max} = Decimation\ rate^2$$

Equation 9 Max raw count

When the Decimation rate reaches its maximum of 255, the following condition needs to be ensured to avoid CIC2 accumulator (25 bit) overflow.

$$CIC2_Acc_Size_{Max} \geq TRUNC\left(\frac{Nsub*Sns_Clk_Div}{Decimation} - 1\right)*Decimation^2$$

Equation 10 CIC2 accumulator overflow condition

Where,

$CIC2_Acc_Size_{Max}$ – Maximum CIC2 accumulator size

Maximum CIC2 accumulator size for direct clock is $(2^{25} - 1)$ and for PRS clock is $\frac{(2^{25} - 1)}{2}$.

3.5 Low-power CAPSENSE™ example

The code example CE235111 - PSOC™ 4: MSCLP CAPSENSE™ low-power demonstrates how to configure and manually tune the low-power widgets. In this code example, a single CSD button is scanned as a regular widget (for Active and ALR modes) and low-power widget for WoT mode. The firmware flowchart (Figure 14) shows different user power modes and mode transition based on user activity.

3 Firmware design considerations

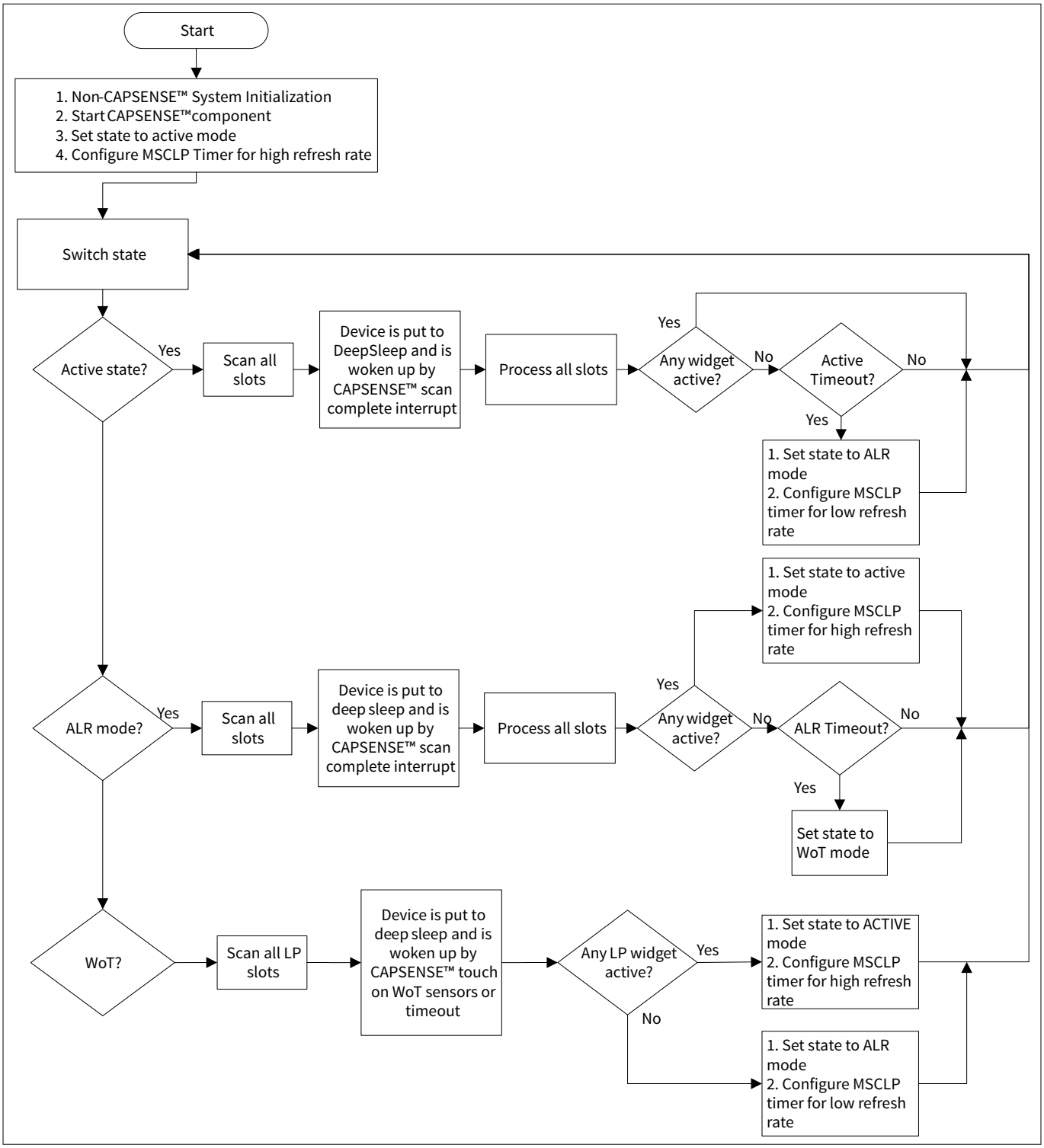


Figure 14 Low-power design flowchart

4 Device configuration considerations

Firmware techniques for low-power design section discusses firmware techniques for the CAPSENSE™ component to reduce the power consumption. In a user application, it is important to have the PSOC™ device and other peripherals to be configured to reduce the power consumption.

4.1 CPU and system clocks

In some cases, running the CPU clock faster can reduce the average current consumption. Having a faster clock can reduce the processing time. As shown in Figure 2, the processing stage has the highest current consumption. Reducing the processing time reduces the overall current consumption.

For example, in PSOC™ 4000T, the typical current consumption is 3 mA when the CPU clock is at 24 MHz and 5.4 mA when CPU clock is at 48 MHz. At a 48-MHz clock, even though the instantaneous current while in active mode is higher by ~80 percent, the average current for the same refresh rate is reduced by 10 percent because the processing time is halved (see AN86233: PSOC™ 4 MCU low-power modes and power reduction techniquesfor a detailed explanation).

4.2 CPU power modes

PSOC™ 4 device with MSCLP devices has three different power modes - Active, Sleep, and DeepSleep.

- **Active mode:** Normal operating power mode in which all peripherals are available, and the CPU is active. For a low-power CAPSENSE™ application, the sensor configuration, trigger scanning, and raw data processing occur in this mode
- **Sleep mode:** CPU does not run any instruction and it waits for an interrupt to occur; all peripherals remain active. This mode is not recommended, if it is viable by an application to use DeepSleep. Sleep mode is less efficient than DeepSleep in terms of power savings
- **DeepSleep mode:** High-frequency clocks and peripherals that require high-frequency clocks are disabled, except the MSCLP block. For achieving the lowest current consumption in low-power design, the device should be kept in this DeepSleep as long as possible

The WoT mode of CAPSENSE™ enables low-power scanning while the device is in DeepSleep mode; this is enabled by the internal system resources and MSCLP timer. MSCLP can generate local system resources (such as high-frequency clock) internal to the MSCLP block for scanning low-power sensors. The MSCLP timer can act as a periodic interrupt to start a complete frame scan according to the requirement.

The system power management (SysPm) API provides functions to change power modes. The API can also register callback functions to execute a peripheral function before or after power mode transitions. See Low-power CAPSENSE™ example section for low-power code example, which shows callback function usage and see the code example: Peripheral driver library documents for more details.

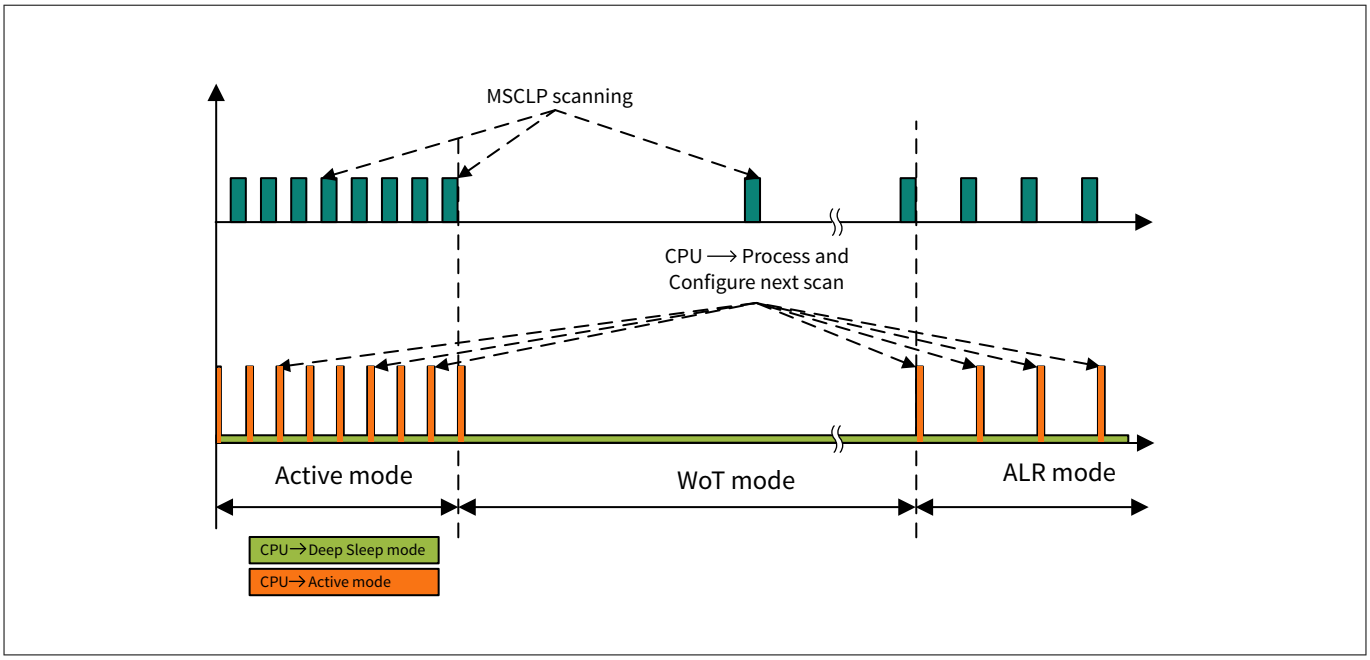


Figure 15 Recommended device power modes for different user modes

4.3 Power state of other peripherals

Each peripheral component configured in the device increases the current consumption; it is recommended to turn off or disable every component which are not in use.

The SysPm driver provides three functions for callback: registration, deregistration, and execution. These functions not only help in power optimization, but also in preventing an abnormal peripheral state after mode transition. Most peripheral drivers have predefined callbacks associated with each power mode. For more information on callback registration and implementation, see AN86233 - PSOC™ 4 MCU low-power modes and power reduction techniques.

4.4 Unused pin states

Configure all unused pins that are connected to shield with drive mode to Analog High-Z. The drive mode can be set in the ModusToolbox™ Device Configurator or by using the following API function:

Code Listing 4

```
/* Set MyPin to Analog HI-Z for low-power. */
Cy_GPIO_SetDrivemode(MYPIN_0_PORT, MYPIN_0_NUM, CY_GPIO_DM_HIGHZ);
```

4.5 Debug pin state

By default, the programming and debugging interface in PSOC™ 4 is disabled in all the power modes. Enabling this causes high power consumption because the SWD pins will be in STRONG drive mode. To achieve minimum power consumption, set “Debug mode = NONE” in the Device Configurator; see Figure 16. This reconfigures the SWD pins to the drive mode selected by the user (Analog High-Z) after a delay from the device reset.

***Note**: When the debug interface is disabled by setting SWD pins to Analog High-Z, a reset must occur to access the debug controller inside the PSOC™ device.*

4 Device configuration considerations

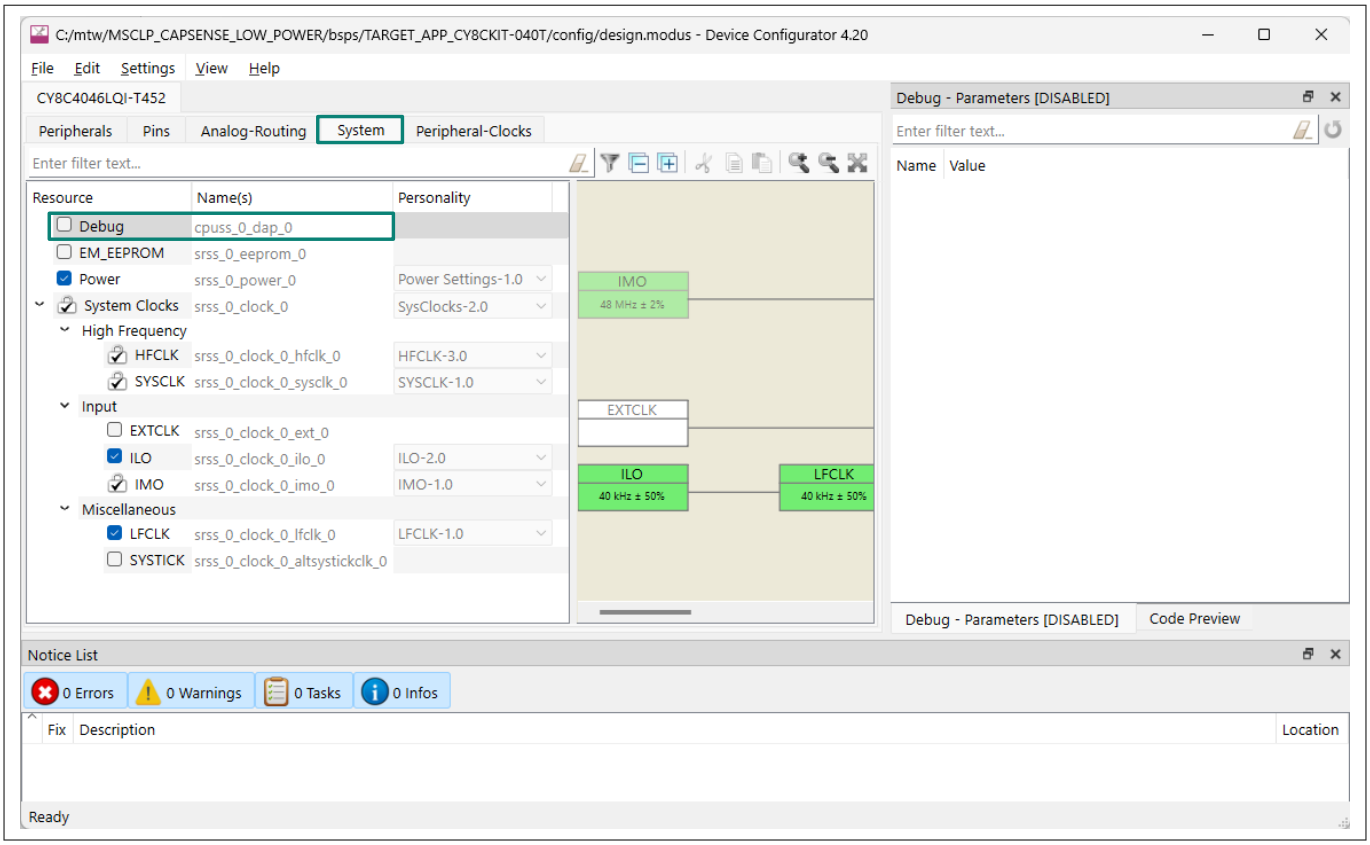


Figure 16 Disabling the debug mode

5 Application-specific considerations

This section explains the possible cases where it is difficult to achieve low power. The two main scenarios that are possible are as follows:

- **Liquid tolerance:** Size of shield electrode, scanning of guard sensors, and implementation of additional firmware logics.
- **Robustness to noise:** Design needed to avoid false trigger of sensors. Possible cases are applications with extra firmware logics, EMC design specifications, and logics like reference sensor scan are influencing this.

5.1 Liquid-tolerant applications

Figure 17 shows the different power modes and transition conditions for a typical liquid-tolerant application. When the firmware detects liquid over the sensors, the application moves to the **Liquid Active** mode. A special liquid detection algorithm is executed in the firmware using a certain set of sensors to detect liquid on the surface. While in **Active** mode, the liquid detection algorithm needs to be executed every time before processing the sensors, as the liquid on the sensor can cause false detection. Once the device is in **Liquid Active** mode, the liquid detection algorithm is executed periodically to check if the liquid is removed from the touch surface, and the device can be safely moved to **Active** mode. The code example CE234752 - PSOC™ 4: MSCLP robust low-power liquid-tolerant CAPSENSE™demonstrates how to implement a low-power, liquid-tolerant, and robust capacitive sensing solution.

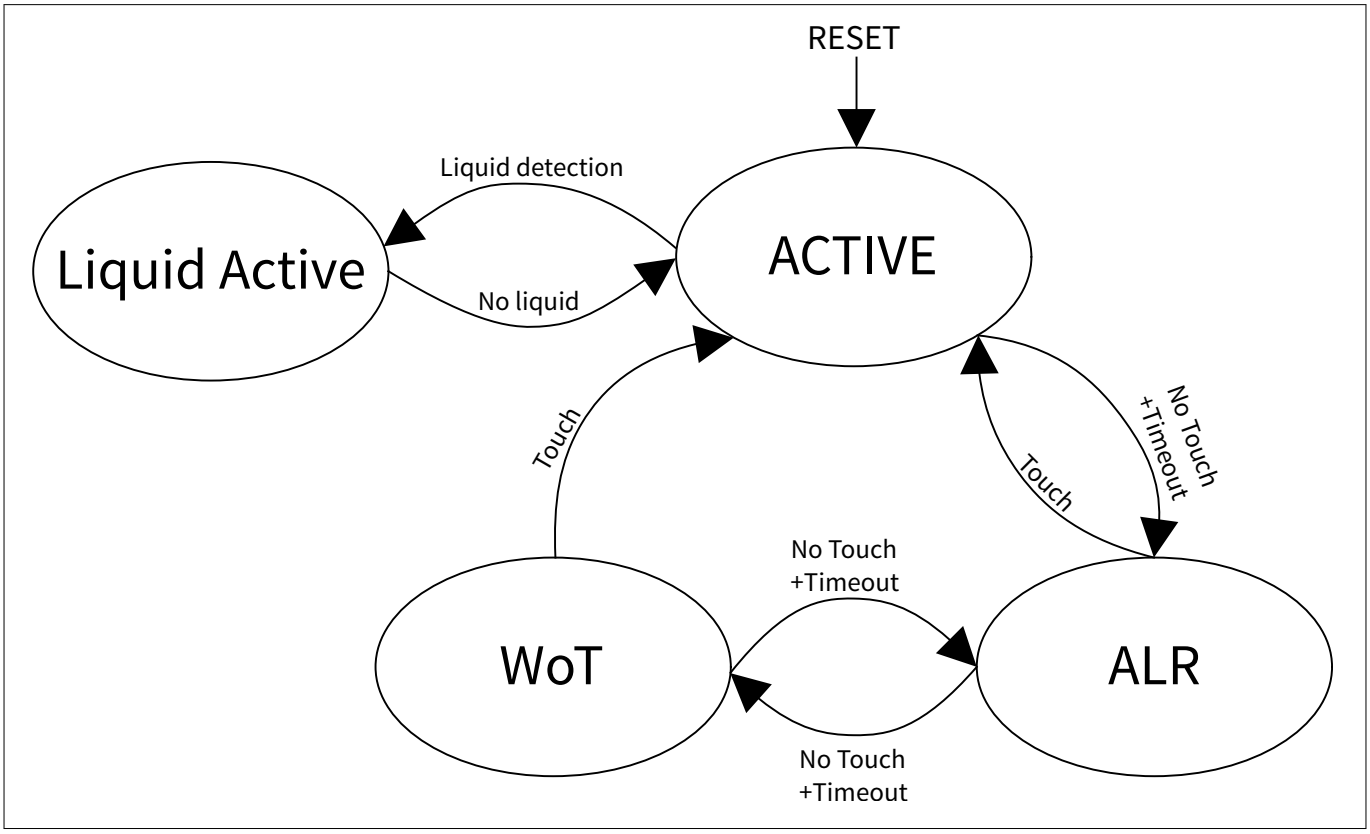


Figure 17 Power mode transition

In wearable applications, exposure to liquids is common (exposure to rain, swimming, physical activities, and so on) and is likely to occur for extended periods. Therefore, optimizing the power consumption in **Liquid Active** mode is critical. Reducing the refresh rate is one of the approaches used for reducing the power consumption in this mode.

5.1.1 Reducing refresh rate in Liquid Active mode

Even though there are sensors reporting signals in Liquid Active mode, scanning the sensors at high refresh rate is not required. In this mode, all the touch reporting is disabled and the condition to bring back the device to Active mode is slow (removal or dry up of liquid). Therefore, using a lower refresh rate such as 16 or 32 Hz decreases the power consumption while the solution is in contact with liquid.

5.2 Gesture applications

For low-power gesture applications, the wake-up method is particularly important to achieve best power optimization results.

5.2.1 Using touch button on WoT

This is a simple and efficient method to achieve best scanning time and power utilization.

- It is a two-stage approach: wake-up by touch and gesture detection
- It does not allow direct gesture on WoT mode as it must wait until wake-up by touch and then scan for gesture

5.2.2 Using proximity on WoT

Use proximity sensors around the track/touch pad, which can detect the presence of hand in the Wake-on-Touch mode to put the system in Active mode for gesture detection.

Implement a proximity sensor by ganging other sensors together as shown in Figure 9. This is accomplished by combining multiple sensor pads into one large sensor using the firmware.

However, the application must consider the following points if opting for this solution:

- Application can do gesture detection directly after wake-up, when the wake-up sequence is invisible to the user. This is the time to reach the trackpad after proximity wake-up
- It depends on the state transition time from WoT to Active mode to get the system ready for scanning after proximity detection
- Grouping more sensors increases the scanning time and power consumption. Therefore, it is preferred to go with minimum number of sensors. This can be achieved by configuring proximity sensing with specific region/sides of the tracked pad
- It introduces high parasitic capacitance

Refer to the following documents for more information on proximity sensing:

- Proximity sensing section in AN64846 - Getting started with CAPSENSE™
- Gesture Section in CAPSENSE™ in AN85951 - PSOC™ 4 and PSOC™ 6 MCU CAPSENSE™ design guide

5.3 Robustness to external noise

5.3.1 Implementation of software filters

The proper software filter selection helps reduce external noise; see the “Software filtering” section in the AN64846 - Getting started with CAPSENSE™ application note for more details. It is recommended to consider the increase in power consumption due to increased processing time while configuring software filters.

5.3.2 Implementation of firmware logics

Adding additional customized firmware logics increases the processing time. Also, scanning logics like the reference sensor scan may lead to longer scanning time. In both cases, achieving low power is challenging.

5.3.3 Low-power EMC considerations

See “ESD protection” and “Electromagnetic compatibility (EMC) considerations” sections in the AN64846 - Getting started with CAPSENSE™ application note for general guidelines.

Radiated Immunity (RI) and Radiated Emission (RE) are the major points that need to be considered.

5.3.3.1 Radiated immunity

1. Multi-frequency scanning (MFS) can prevent false touch detection in the presence of external noise at a particular frequency. However, MFS increases scanning time by 3x and increases power consumption as well
2. Floating pins and traces can receive external noise and impact CAPSENSE™ performance. When such traces and pins are not in use, configuring them as shield can improve immunity
3. Using a shield can improve immunity as well as increase the signal but it reduces the maximum possible scan frequency due to an increase in shield capacitance, this increases scan time as well as power consumption
4. Increasing external series resistance (see Figure 18) decreases noise from input.

   However, this will have the following impacts:

   a. Decreases sense frequency (1/20 RC) and increase scan time and power
   b. Reduces the signal and SNR values

   This is a trade-off between noise immunity and power that the design should account for
5. Eliminating high-transient voltages entering the system with proper design with decoupling capacitors, ferrite bead, and TVS diodes will improve noise immunity

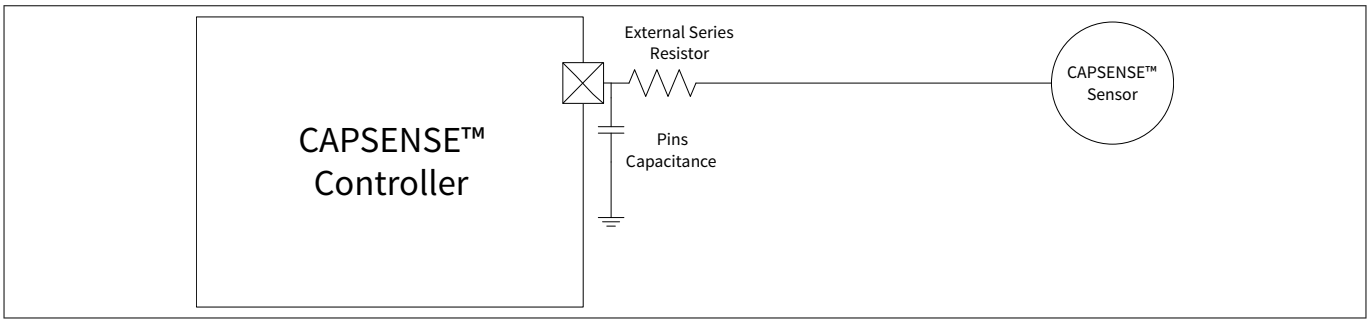


Figure 18 RC filter

5.3.3.2 Radiated emission (RE)

1. The RE can be reduced by shielding adjacent sensors (see section Shield regions)
   a. When inactive sensor connections are shielded, it covers a large area, emitting more and consuming more power
   b. Instead of shielding all the sensors, shield only the adjacent ones
2. Any design configuration that has a high scanning time leads to more emissions (see the "Radiated emissions"section in the AN64846 - Getting started with CAPSENSE™ application note)
3. To avoid the antenna effect of cables carrying a signal outside, use properly shielded cables and as few sensors traces as possible

The sensors planned for low-power WoT operation can be configured in such a way to meet RI and RE considerations during the initial design phase itself.

6 Hardware design considerations

In a CAPSENSE™ application, capacitive sensors are formed by the traces and pads of a printed circuit board (PCB) or flex circuit. A good hardware design ensures that the design is robust and helps in achieving low average power consumption. Poorly designed CAPSENSE™ hardware can lead to increased power consumption that cannot be fully compensated by firmware algorithm or sensor tuning. See AN85851 - PSOC™ 4 and PSOC™ 6 MCU CAPSENSE™ design guide for hardware considerations such as sensor construction, overlay selection, and PCB layout guidelines in the “Design considerations” section.

The following sections explain additional hardware design considerations specific to PSOC™ 4 with MSCLP devices to achieve low average power.

6.1 Shield regions

While in low-power mode - WoT, the recommendation is to have the low-power widget sensors with the lowest Cp possible. This can be achieved by reducing number of sensors as mentioned in section Reducing the scan time in low-power mode. Even after reducing the number of sensor or sensor pad area, Cp can be further reduced by having an active shield. Shield is a technique used by CAPSENSE™ to enable liquid tolerance and to reduce sensor Cp, in which the shield electrode is driven by a signal that is equal to the sensor switching signal in phase and amplitude. Shield electrodes are hatch patterns surrounding the sensors. The hatch pattern around the sensor can be driven as a shield for further reduction of Cp, therefore decreasing the current consumption. However, a larger shield having higher shield capacitance (Csh) limits the maximum frequency and therefore increases the scan time and current consumption.

The solution here is to have different isolated shield regions connected to PSOC™ through different pins. One shield electrode region surrounding just the low-power widgets. In low-power mode, the pin connected to this shield electrode region will be driven, therefore reducing the shield electrode Cp, and also reducing the low-power sensor Cp in low-power mode. The other shield electrode region covers all the remaining sensors to be scanned in active mode.

In Figure 19, Shield Electrode 0 is driven with the shield signal while in WoT mode and both the shield electrodes (Shield Electrode 0 and Shield Electrode 1) are driven with the shield signal in Active mode. The application must handle the runtime reconfiguration of shield electrodes using the middleware API `Cy_CapSense_SlotPinState()`. See CAPSENSE™ Middleware API Reference Guide for more details.

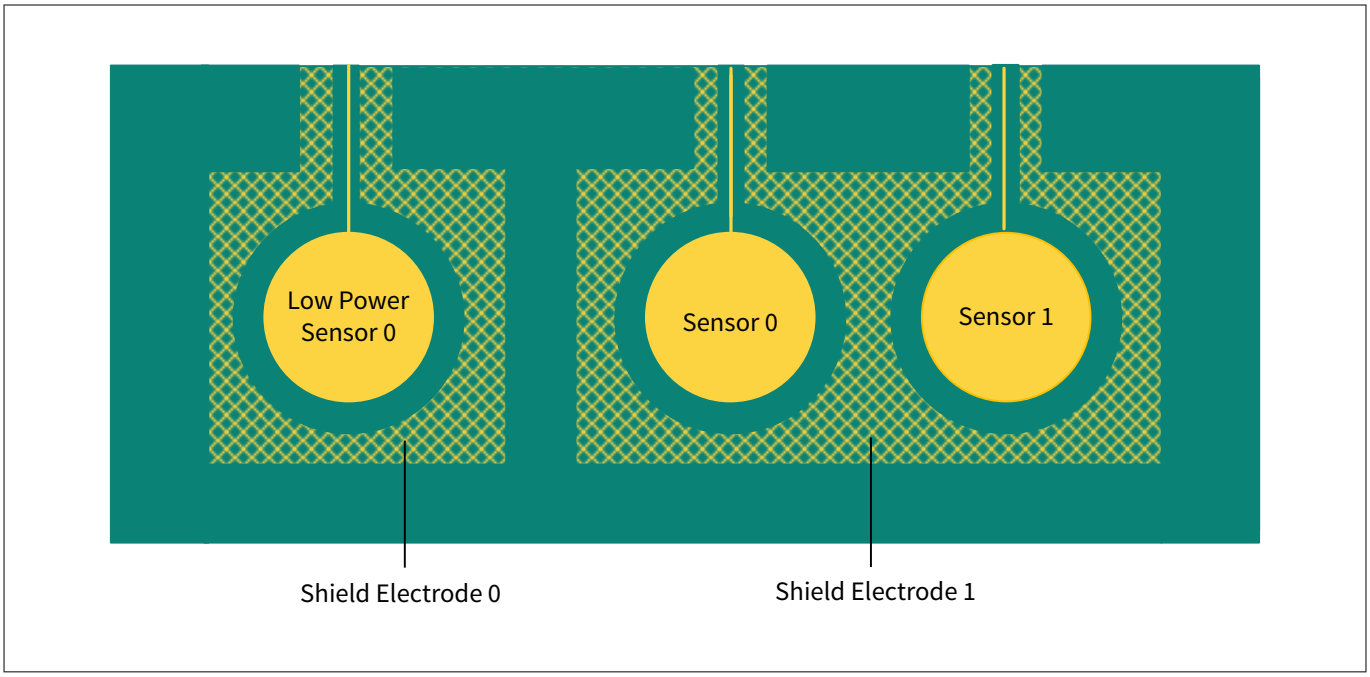


Figure 19 Independent shield electrodes for low-power sensor

References

Product page

[1] Infineon Technologies AG: *PSOC™ 4000T*; Available online
[2] Infineon Technologies AG: *PSOC™ 4100T Plus*; Available online

Device datasheet

[3] Infineon Technologies AG: *PSOC™ 4: PSOC™ 4000T datasheet*; Available online
[4] Infineon Technologies AG: *PSOC™ 4: PSOC™ 4100T Plus datasheet*; Available online

Application notes

[5] Infineon Technologies AG: *AN85951 - PSOC™ 4 and PSOC™ 6 CAPSENSE™MCU design guide*; Available online
[6] Infineon Technologies AG: *AN79953 - Getting started with PSOC™ 4 MCU*; Available online

Development kits

[7] Infineon Technologies AG: *PSOC™ 4000T CAPSENSE™ Evaluation Kit (CY8CKIT-040T)*; Available online
[8] Infineon Technologies AG: *PSOC™ 4000T CAPSENSE™ Prototyping Kit (CY8CPROTO-040T)*; Available online
[9] Infineon Technologies AG: *PSOC™ 4000T Multi-Sense Prototyping Kit (CY8CPROTO-040T-MS)*; Available online
[10] Infineon Technologies AG: *PSOC™ 4100T Plus CAPSENSE™ Prototyping Kit (CY8CPROTO-041TP)*; Available online

Component datasheet/middleware documentation

[11] Infineon Technologies AG: *CAPSENSE™ middleware library*; Available online
[12] Infineon Technologies AG: *CAPSENSE™ Middleware API Reference Guide*; Available online
[13] Infineon Technologies AG: *ModusToolbox™ CAPSENSE™ Configurator guide*; Available online

ModusToolbox™

[14] Infineon Technologies AG: *ModusToolbox™ release notes*; Available online
[15] Infineon Technologies AG: *ModusToolbox™ install guide*; Available online
[16] Infineon Technologies AG: *ModusToolbox™ user guide* Available online
[17] Infineon Technologies AG: *ModusToolbox™ quick start guide* Available online
[18] Infineon Technologies AG: *ModusToolbox™ CAPSENSE™ Configurator* Available online
[19] Infineon Technologies AG: *ModusToolbox™ CAPSENSE™ Tuner* Available online
[20] Infineon Technologies AG: *ModusToolbox™ Device Configurator* Available online
[21] Infineon Technologies AG: *ModusToolbox™ tools package user guide* Available online

Design support

[22] Infineon Technologies AG: *Infineon Developer Community*; Available online
[23] Infineon Technologies AG: *Technical support*; Available online

Revision history

Document revision	Date	Description of changes
**	2022-06-14	Initial release
*A	2023-02-24	Updated Power budgeting in Section Power budgeting Updated power mode content in Section Power consumptions in different application states Updated scanning mode content in Section CAPSENSE™ hardware scanning modes Added section Regular widget Regular Widget Updated shield design in Section Shield design Updated current measurements in Table 2. Added "CIC2 shift value" column in Table 4. Updated Figure 1, Figure 2, Figure 5, Figure 10, and Figure 11: Added ALR mode after Active mode. Updated Figure 12: updated with IMO clock of 46 MHz. Added section Application-specific considerations
*B	2023-03-02	Corrected typo in Section Scan duration.
*C	2023-07-11	Updated Equation-9 in section CIC2 filter Updated Table 4. Updated Figure 12: updated latest GUI. Added section Gesture applications Removed Section 5.1.2.
*D	2023-07-28	Removed the "restricted" tag in the header.
*E	2023-09-07	Updated Section CIC2 filter Removed Number of CIC2 samples equation. Updated Decimation rate equation. Removed CIC2 look-up table. Updated all the URLs to vanity URLs.
*F	2024-03-12	Fixed broken links. Updated References section.
*G	2024-03-22	Updated section 3.4 CIC2 filter. Added Equation 10: CIC2 accumulator overflow condition. Added Figure 6: MSCLP timer and Refresh rate. Updated Code Listing 3. Updated section 4.5 Debug pin state. Updated Figure 16: Disabling the debug mode.

Revision history

Document revision	Date	Description of changes
*H	2024-09-19	Changed the title of the document. Updated the applicable occurrences of PSOC™ 4000T to PSOC™ 4 device with MSCLP. Updated Section Power consumption in low-power designs Added Table 1 Updated Table 2 to include the measurements of PSOC™ 4100T Plus and power consumption Updated Section MSCLP timer Updated Section Wakeup/power-on button Updated Section Wake-on-touch scan interval Updated Section Wake-on-Touch timeouts Updated Section Shield regions Updated References section
*I	2025-03-14	Updated the document with the latest branding guidelines
*J	2025-05-23	Template update; no content change
*K	2025-12-04	Updated graphics for latest branding.
*L	2026-09-01	Updated Table 1

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 2026-09-01

Published by

Infineon Technologies AG

81726 Munich, Germany

© 2026 Infineon Technologies AG

All Rights Reserved.

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

Email: erratum@infineon.com

Document reference
IFX-qrr1744301683609

Important notice

Products which may also include samples and may be comprised of hardware or software or both ("Product(s)") are sold or provided and delivered by Infineon Technologies AG and its affiliates ("Infineon") subject to the terms and conditions of the frame supply contract or other written agreement(s) executed by a customer and Infineon or, in the absence of the foregoing, the applicable Sales Conditions of Infineon. General terms and conditions of a customer or deviations from applicable Sales Conditions of Infineon shall only be binding for Infineon if and to the extent Infineon has given its express written consent.

For the avoidance of doubt, Infineon disclaims all warranties of non-infringement of third-party rights and implied warranties such as warranties of fitness for a specific use/purpose or merchantability. Infineon shall not be responsible for any information with respect to samples, the application or customer's specific use of any Product or for any examples or typical values given in this document.

If this document is marked as 'Preliminary', 'Target', 'Draft' or 'Proposal', Infineon reserves the right to change all information given in this document at any time without notice. Subject to the development and release of a Product for series supply by Infineon, the technical specifications of the Product are set forth in the relevant datasheet provided by Infineon without such marking.

The data contained in this document is exclusively intended for technically qualified and skilled customer representatives. It is the responsibility of the customer to evaluate the suitability of the Product for the intended application and the customer's specific use and to verify all relevant technical data contained in this document in the intended application and the customer's specific use. The customer is responsible for properly designing, programming, and testing the functionality and safety of the intended application, as well as complying with any legal requirements related to its use.

Unless otherwise explicitly approved by Infineon, Products may not be used in any application where a failure of the Products or any consequences of the use thereof can reasonably be expected to result in personal injury.

However, the foregoing shall not prevent the customer from using any Product in such fields of use that Infineon has explicitly designed and sold it for, provided that the overall responsibility for the application lies with the customer. Infineon expressly reserves the right to use its content for commercial text and data mining (TDM) according to applicable laws, e.g. Section 44b of the German Copyright Act (UrhG). If the Product includes security features:

Because no computing device can be absolutely secure, and despite security measures implemented in the Product, Infineon does not guarantee that the Product will be free from intrusion, data theft or loss, or other breaches ("Security Breaches"), and Infineon shall have no liability arising out of any Security Breaches.

If this document includes or references software:

The software is owned by Infineon under the intellectual property laws and treaties of the United States, Germany, and other countries worldwide. All rights reserved. Therefore, you may use the software only as provided in the software license agreement accompanying the software.

If no software license agreement applies, Infineon hereby grants you a personal, non-exclusive, non-transferable license (without the right to sublicense) under its intellectual property rights in the software (a) for software provided in source code form, to modify and reproduce the software solely for use with Infineon hardware products, only internally within your organization, and (b) to distribute the software in binary code form externally to end users, solely for use on Infineon hardware products. Any other use, reproduction, modification, translation, or compilation of the software is prohibited.

For further information on the Product, technology, delivery terms and conditions, and prices, please contact your nearest Infineon office or visit www.infineon.com.