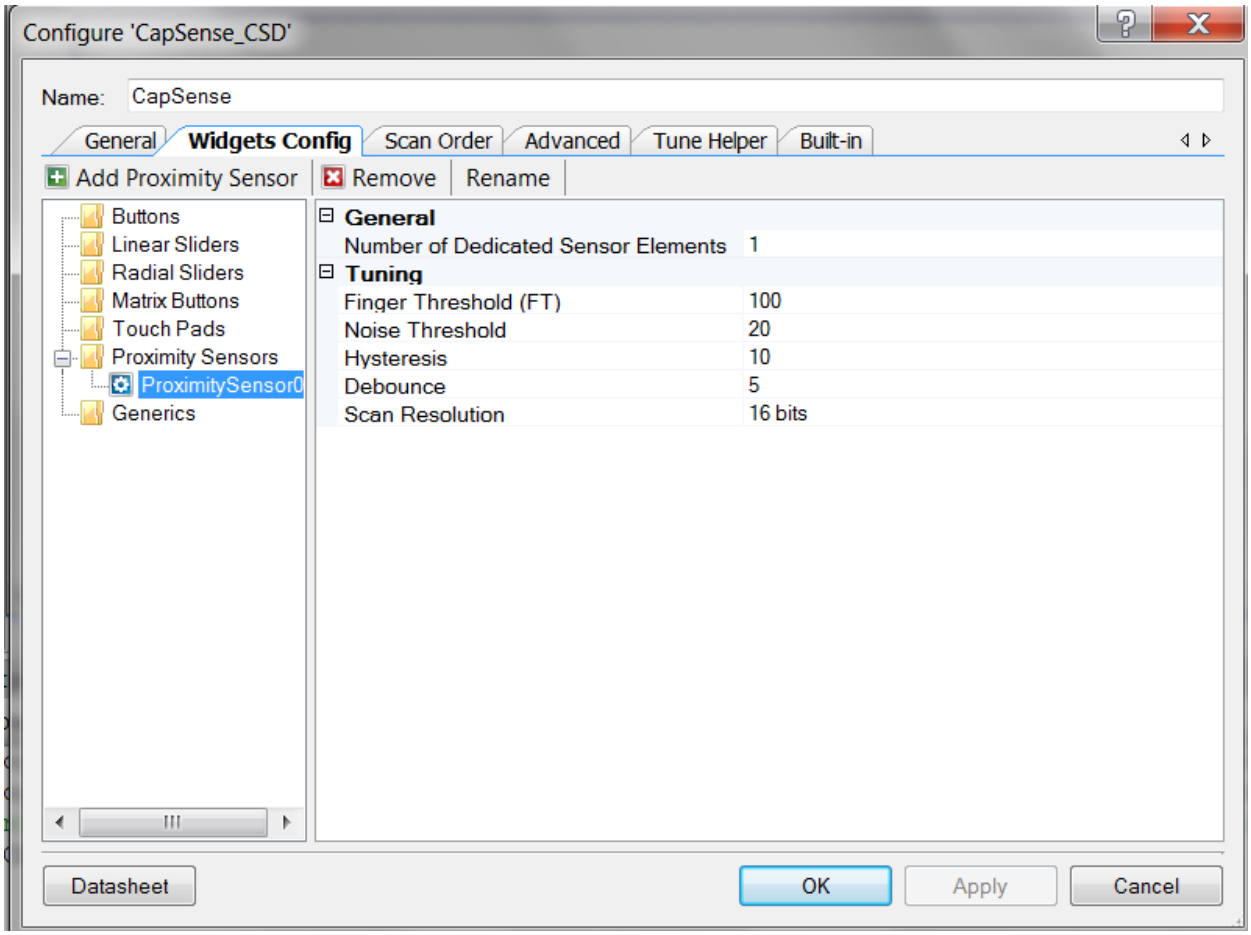
A simple Proximity Sensor using PSoC 4

Capsense CSD block supports Proximity Capacitive Sensors for proximity detection. A proximity sensor is optimized to detect the presence of finger, hand or other large object at a large distance from the sensor. This avoids the need for an actual touch. The Capsense CSD component can be configured to enable Proximity Detection.



Basic Programming Flow

The Proximity Sensor uses the following widget define, so that this define can be passed as the argument for the CapSense_EnableWidget API which would enable the Proximity Sensor:

CapSense_PROXIMITYSENSOR0__PROX

Here the instance name of the component is ‘CapSense’.

In the program, we define a state machine which can assume four States:

STATE_SCAN_CAPSENSE

STATE_WAIT_FOR_SCAN_COMPLETE

STATE_UPDATE_BASELINE

STATE_UPDATE_LED

State 1

The first state is to scan the Enabled Widget which is Proximity in our case. The process continues until the enabled widget is scanned. Proximity widgets must be manually enabled as their long scan time is incompatible with the fast response required of other widget types. That is why, before implementing the state machine itself, we call the following API:

```
CapSense_EnableWidget(CapSense_PROXIMITYSENSOR0__PROX);
```

After enabling the widget, it moves to the next state which checks if the scan is complete.

State 2

We check if the scan is completed or not using the API CapSense_IsBusy(). If the scanning is completed, it returns 0 or else it returns 1. If the scanning is complete, we move on to the next state which is STATE_UPDATE_BASELINE

State 3

This is done using the API CapSense_UpdateEnabledBaselines() which calls the Capsense_UpdateSensorBaselines() to update the baselines for the enabled sensor-proximity in this case. Once this is done, we move on to the next state which is STATE_UPDATE_LED.

State 4

Here we actually check if a finger has been brought near the sensor. In that case, we blink an LED . The API that is used for checking this is : CapSense_CheckIsWidgetActive(CapSense_PROXIMITYSENSOR0__PROX) This function basically compares the Proximity Sensor CapSense_Signal[ ] array value to its finger threshold. Hysteresis and debounce are also considered. If the sensor is active, the threshold is lowered by the hysteresis amount. If it is inactive, the threshold is increased by the hysteresis amount. If the active threshold is met, the debounce counter increments by one until reaching the sensor active transition, at which point this API sets the widget as active. This function also updates the sensor's bit in the CapSense_SensorOnMask[ ] array.

Default State

The Default State is STATE_SCAN_CAPSENSE.

Testing the Project

The project has been tested in CY8CKIT-042 PSoC 4 Pioneer kit. For testing, instead of connecting a straight wire to P0 [4], you could find better response if you make a loop out of it and connect it to P 0[4]. As you bring your hand close to this wire, you can see an LED blinking on the Kit .