Once the electrical and mechanical design considerations of a proximity sensor are determined, you must test your sensor in your product scenario. If the design requires the proximity sensor to identify an approaching object, then testing will reveal the proximity output counts with respect to distance. The shape of the output depends on the composition and reflectivity of the object. The proximity response then allows the designer to select an appropriate detection scheme.

The output of the analog-to-digital converter (ADC count) has a typical response as shown in Figure 1.

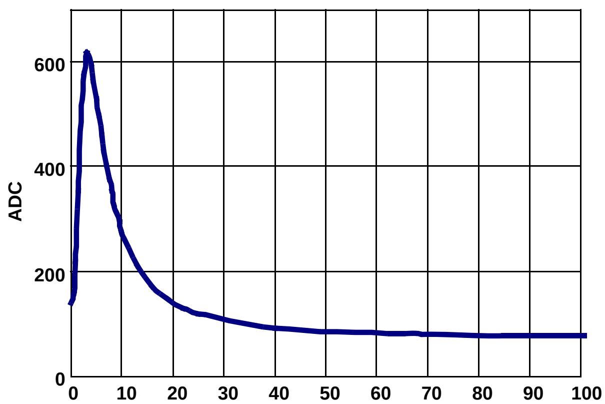


FIGURE 1. PROXIMITY RESPONSE CURVE vs DISTANCE FOR A TYPICAL OBJECT

The curve shows a peak ADC count at a distance just a few millimeters from the detector. The response drops at shorter distances because there is space between the emitter and detector. The closeness of the object reduces the amount of IR LED light that can reflect to the detector.

As the object moves far from the system, the ADC count flattens. This level is the noise floor set by the surrounding environment. The combination of the peak and noise floor is crucial in setting a trigger level.

Case 1 Algorithm

When the detected object is human skin, the design is somewhat simplified. Infrared light can penetrate human skin. The peak is shifted to a closer distance and the ADC count does not plummet at zero distance (see Figure 2). With optimum optical designs, it is possible to achieve Figure 2 proximity sensing performance with other non-human sensing objects.

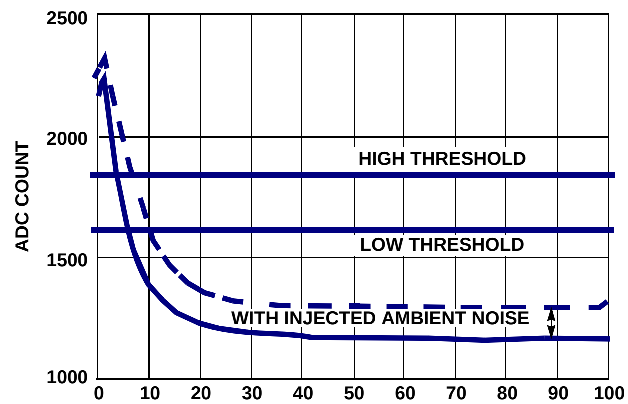
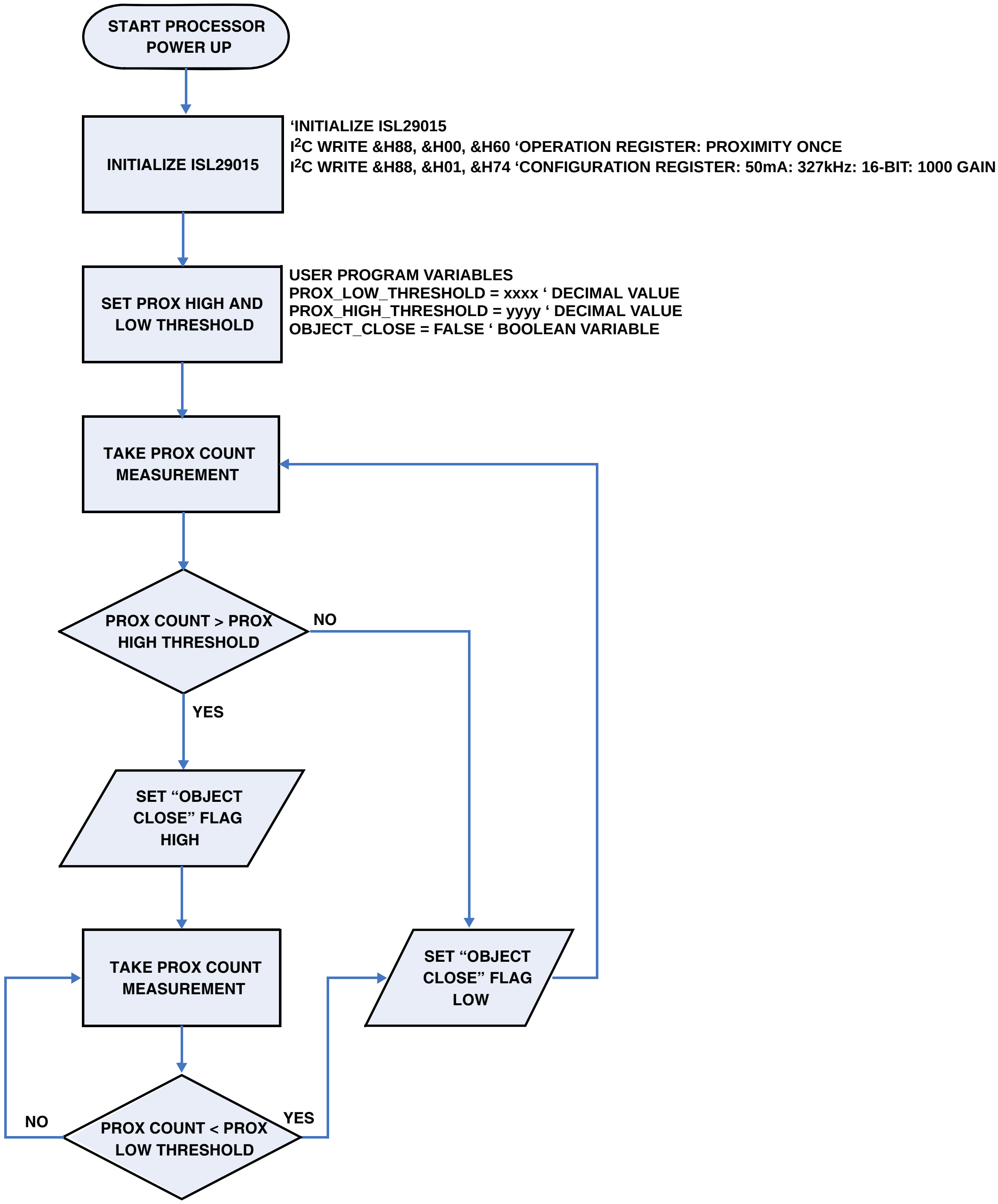


FIGURE 2. PROXIMITY RESPONSE CURVE vs DISTANCE FOR HUMAN SKIN

The shape of this response allows for a simple threshold scheme to identify the proximity of a human (as in the case of answering a cell phone call). The low threshold must be high enough to ignore any changes in ambient noise. The high threshold must be low enough to withstand any changes in the peak output ADC count. The space between them must also be large enough to handle any ambient changes.

The following flow-chart reveals the design process for this type of system (see Figure 3).

CASE 1: ALGORITHM FOR PROX COUNT STAYS HIGH AS THE OBJECT IS CLOSE THE THE GLASS.



DETAIL A: TAKE PROXIMITY COUNT MEASUREMENT DETAIL

'ISL29015 PROX-IR MEASUREMENT SEQUENCE
I^2C WRITE &H88, &H00, &H60 'OPERATION REGISTER: PROXIMITY ONCE
DELAY 120ms
I^2C READ &H89, &H02, LSB_BYTE, MSB_BYTE 'READ AD REG: 2 BYTES
PROX_RAW = (MSB*256)+LSB
I^2C WRITE &H88, &H00, &H40 'OPERATION REGISTER: IR ONCE
DELAY 120ms
I^2C READ &H89, &H02, LSB_BYTE, MSB_BYTE 'READ AD REG: 2 BYTES
IR_RAW = (MSB*256)+LSB
PROX_COUNT = PROX_RAW-IR_RAW 'SAVE CALCULATED PROXIMITY IN VARIABLE

FIGURE 3. FLOW-CHART FOR PROXIMITY SYSTEM DETECTING HUMAN SKIN

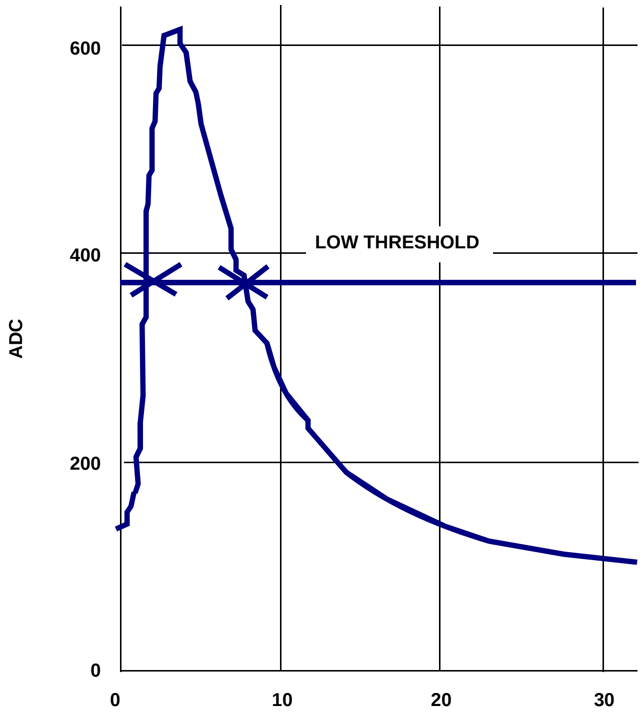


FIGURE 4. PROXIMITY RESPONSE CURVE vs DISTANCE WITH LOW THRESHOLD CROSSING

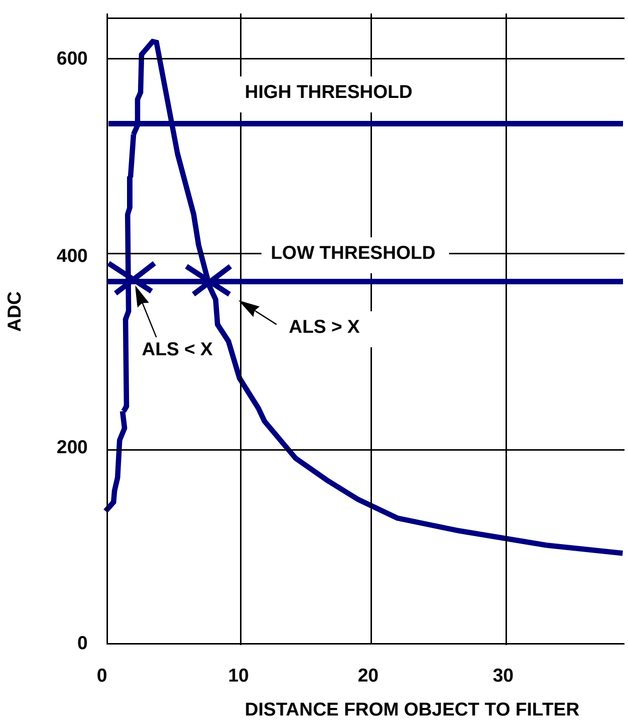


FIGURE 5. PROXIMITY RESPONSE CURVE vs DISTANCE WITH LOW AND HIGH THRESHOLDS CROSSING

Case 2 Algorithm

The case 2 algorithm is for a system with a response similar to the curve in Figure 1. Again we would like to set a low and high threshold. However, the selection is more complicated since the low threshold selection will, most likely, cross the proximity response curve twice (see Figure 4).

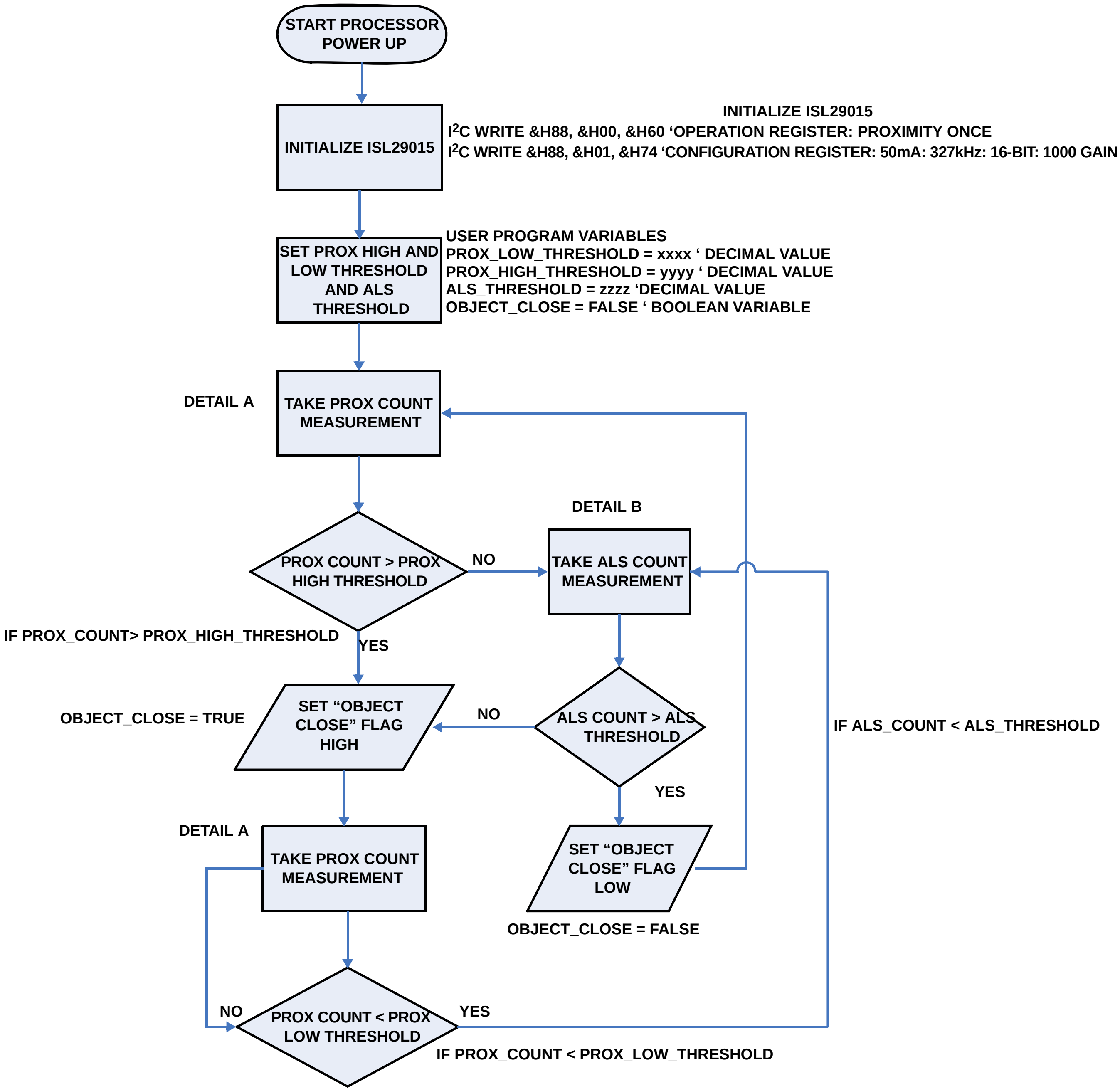
The lower threshold is used to identify the evacuation of an object from the area directly above the proximity sensor. Only one of the 2 crosspoints will correctly identify that situation, the second one on the right side of the peak.

To determine which of the object distance has caused the threshold to be crossed, we sense the amount of ambient light (see Figure 5). Ambient light can be used because the object will naturally obstruct ambient light as it approaches the sensor. Since the object blocks virtually no ambient light at large distances and blocks a continuously increasing amount of light as it approaches the sensor, the amount of ambient light gives us another measurement we can use to separate the two lower threshold crossings from each other.

The ambient light (by the Ambient Light Sensor) is sampled any time the high threshold is crossed. That value is stored in a register. When the low threshold is crossed, the ambient light is sampled again. If the ambient light reading is less than the value taken at the high threshold, then we are on the left side of the curve—the low threshold crossing we want to ignore. However, if the ambient light reading is greater than the value taken at the high threshold, the object is further away from both sensors and we are on the right side of the curve. A microcontroller can be programmed to identify the real low threshold crossings.

A flow-chart describing the design methodology of case 2 is given in Figure 6.

CASE 2: ALGORITHM FOR PROX COUNT DECREASES AS THE OBJECT IS CLOSE THE THE GLASS.



DETAIL A :TAKE PROXIMITY COUNT MEASUREMENT DETAIL
'ISL29015 PROX-IR MEASUREMENT SEQUENCE
I^2C WRITE &H88, &H00, &H60 'OPERATION REGISTER: PROXIMITY ONCE
DELAY 120ms
I^2C READ &H89, &H02, LSB_BYTE, MSB_BYTE 'READ AD REG: 2 BYTES
PROX_RAW = (MSB*256)+LSB
I^2C WRITE &H88, &H00, &H40 'OPERATION REGISTER: IR ONCE
DELAY 120ms
I^2C READ &H89, &H02, LSB_BYTE, MSB_BYTE' READ AD REG: 2 BYTES
IR_RAW = (MSB*256)+LSB
PROX_COUNT = PROX_RAW-IR_RAW 'SAVE CALCULATED PROXIMITY IN VARIABLE

DETAIL B: TAKE ALS COUNT MEASUREMENT
'ISL29015 LIGHT MEASUREMENT SEQUENCE
I^2C WRITE &H88, &H00, &H20 'OPERATION REGISTER: LIGHT ONCE
DELAY 120ms
I^2C READ &H89,&H02,LSB_BYTE,MSB_BYTE 'READ AD REG: 2 BYTES
ALS_COUNT = (MSB*256)+LSB

FIGURE 6. FLOW-CHART FOR PROXIMITY SYSTEM DETECTING GENERIC OBJECT