# OB1203

## OB1203 Heart Rate, Blood Oxygen Concentration, Pulse Oximetry, Proximity, Light and Color Sensor: Proximity Sensing


### Abstract

This application note introduces the concepts and basic methods for proximity sensing with OB1203. The Renesas OB1203 all-in-one RGB color sensor/proximity sensor/PPG biosensor has an infrared (IR) proximity sensor (PS) for detection of nearby objects. The proximity sensor is designed for high performance mobile applications such as face proximity detection for in-call display on-off control.


## Contents



## Figures

# 1. Application Circuit

It is suggested to use low noise power supply sources, such low-noise low dropout (LDO) linear regulators. Independent supplies for analog and digital supply (VDD) and LED supply (LED_VDD or LVDD) are suggested. Pull-up resistors in the appropriate range to microcontroller logic voltage of 1.8–3.3V are necessary for I2C SDA, I2C SCL and INTB pins. A decoupling cap between VDD and analog ground (GND) of 0.1µF and optionally 1µF for longer leads is suggested. 4.7–10 µF capacitors are useful between LVDD and LED_GND (LGND). The analog ground (GND) and LED power ground (LGND) should be connected at a low impedance node to avoid ground bounce effects when the LED current is high. Trace lengths should be minimized and wider trace/via widths used for current carrying LVDD and LGND lines. All the LED pins should be connected to a low thermal impedance pad.

For bench test functionality purposes it is sufficient to connect LVDD and VDD to 3.3V source relative to LGND and GND, and connect the I2C and INTB lines to the logic level (typically 1.8V or 3.3V) via pull-up resistors in the appropriate range. Note that the internal digital and analog levels are set by an internal LDO regulator.

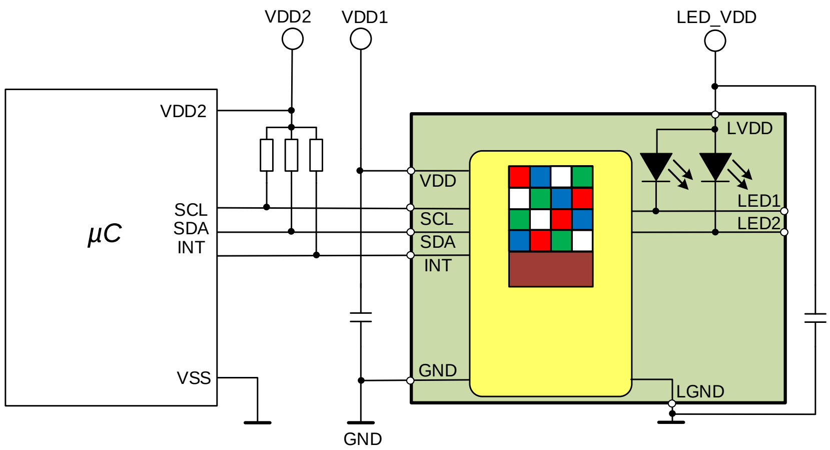


**Figure 1. Application Circuit**

# 2. Proximity Sensing with OB1203

The OB1203 proximity sensor utilizes a 940nm IR IED, a silicon photodiode and a high resolution, wide-range analog digital converter (ADC) with I2C readout to detect nearby objects, such as a user’s face. The 940nm LED is long enough in wavelength to ensure that no “red glow” is visible to users. Photodiode responsivity is limited to red and infrared wavelengths to reduce ambient light sensitivity while allowing response to the red and IR LEDs used in PPG-based heart rate measurements.

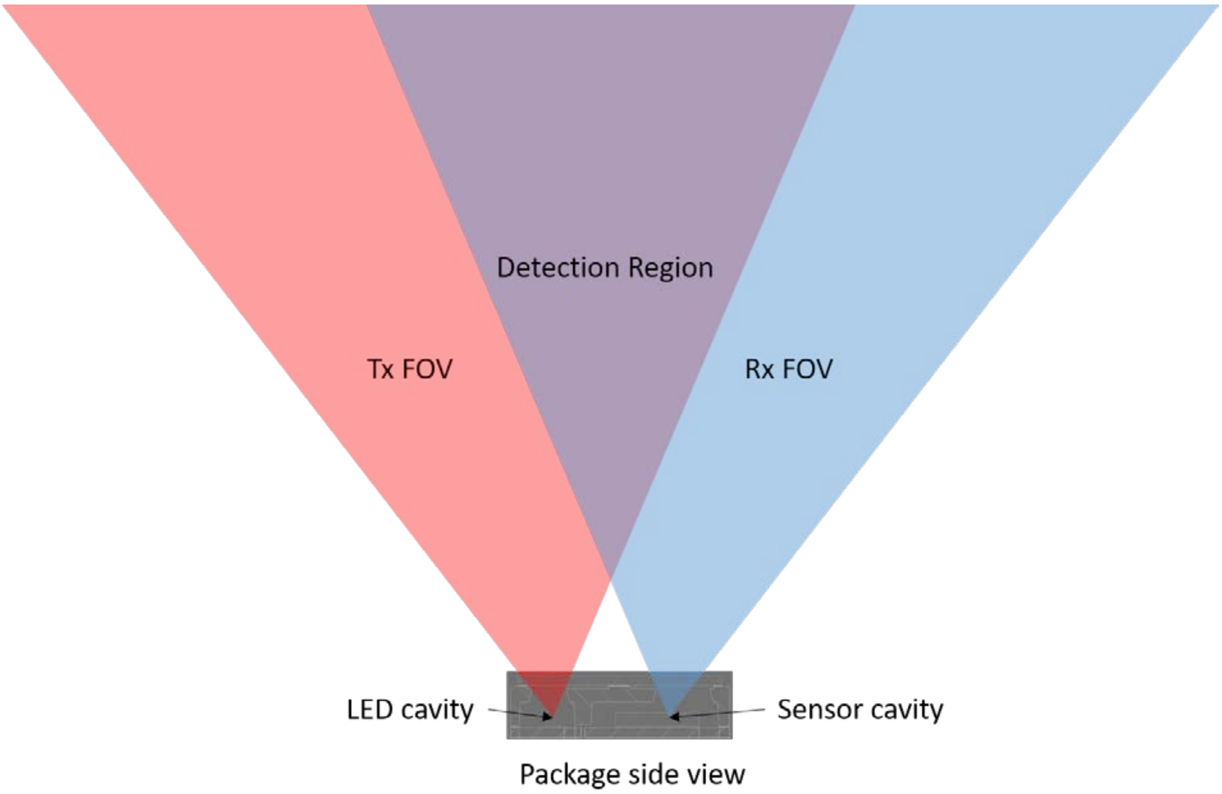


**Figure 2. Proximity Sensor Transmit (Tx), Receive (Rx) and Detection Fields of View (FOV)**

Referring to Figure 1, the LED (transmitter) emits light in a restricted Tx field of view (FOV). The photodiode (receiver) accepts light from the Rx FOV. Objects in the overlap region are detectable.

IR proximity sensors emit a train of short duration pulses of infrared light from the integrated IR-LED. Between the micropulses, the detector samples ambient light. During the micropulses the detector samples ambient light plus the return signal from the LED. The difference is accumulated and shown in the 16-bit wide (2x8 bit) proximity sensor register. The result is MSB justified.

Proximity detection occurs when the return signal exceeds a threshold count level. The signal from nearby objects scales with distance to the target (r) from 1/r2 for extended objects to 1/r4 for small objects. The return signal also depends on the target reflectivity, with darker objects returning less light. Dark (absorbing) targets are detected at a shorter distance than lighter targets, so a proximity sensor is not considered to be a distance sensor, per se. However, due to rapidly rising signal at short distances, proximity sensors are a robust way to detect objects in near proximity, whether light or dark colored.

At very short distances the overlap between Tx and Rx decreases, eventually cancelling out the increase in signal due to the short distance. At distances less than a few millimeters the signal begins to decrease, however the signal remains nonzero and larger than a typical proximity threshold.

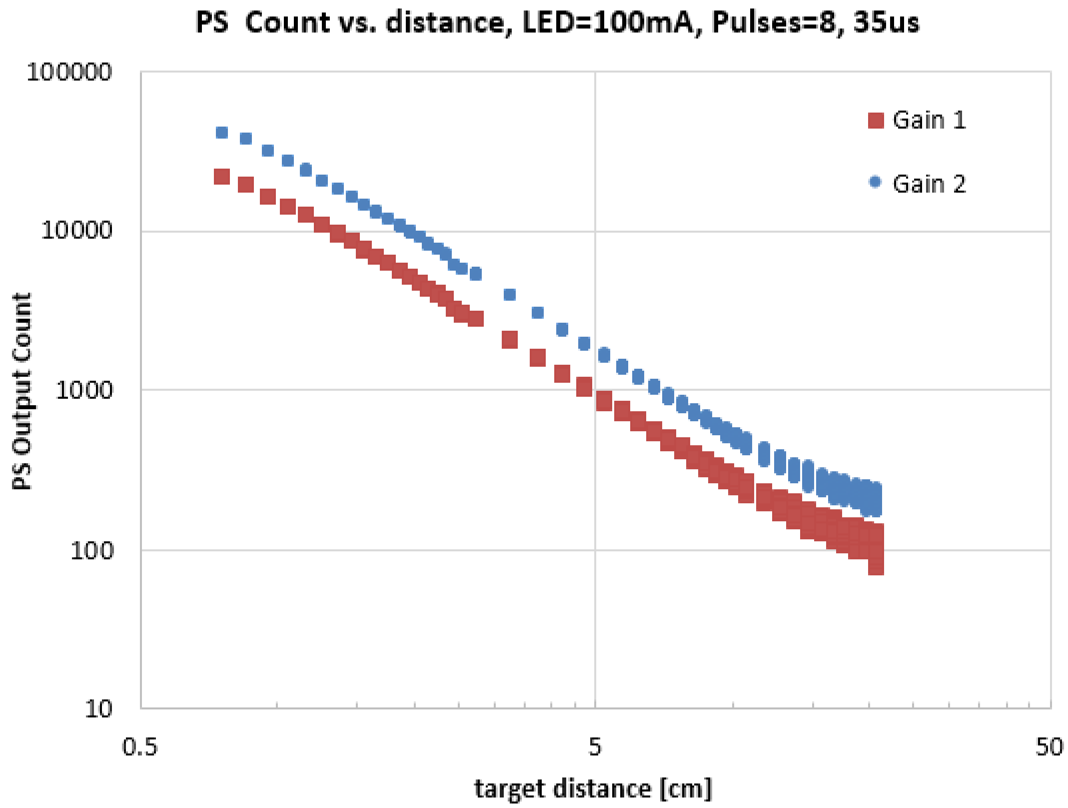


**Figure 3. Proximity Counts vs. Target Distance for an Extended (large) Target**

The highest LED current setting provides an optimal signal to noise ratio (SNR). By using a combination of short duration pulses or pulse trains and low sample rate duty cycles, the overall power consumption is low.

In typical applications, the presence of a cover glass such as a handset screen bezel in front of the sensor, or smudges on the cover glass due to makeup or fingerprints causes a static return signal called crosstalk (clutter in radar terminology). See Figure 3. Crosstalk typically arises from successive stray reflections and can reduce the dynamic range of the sensor.

The Tx FOV of OB1203 is intentionally wider than other proximity sensors. This results in a marginally more crosstalk than traditional proximity sensors. The benefits of the wider Tx FOV include a generous detection region with good performance down to zero distance, as well as spatial averaging of the signal, tolerance of small finger movements during bio PPG measurement, and necessary spatial overlap of red and IR LED beam patterns for SpO2 measurement.

For dealing with crosstalk, OB1203 employs three strategies. First, the sensor is a wide range (> 11 effective bits) so low levels of crosstalk effect the dynamic range negligibly. Second, a programmable fixed number of counts may be digitally subtracted from the proximity sensor measurement (digital crosstalk cancellation) to digitally zero the crosstalk or set the counts to a target baseline level. Third, for applications with very high crosstalk such as quasi-reflective (e.g. white) cover glass, current subtraction of ¼ full scale may be applied at the analog front end (AFE), which ensures full dynamic range.

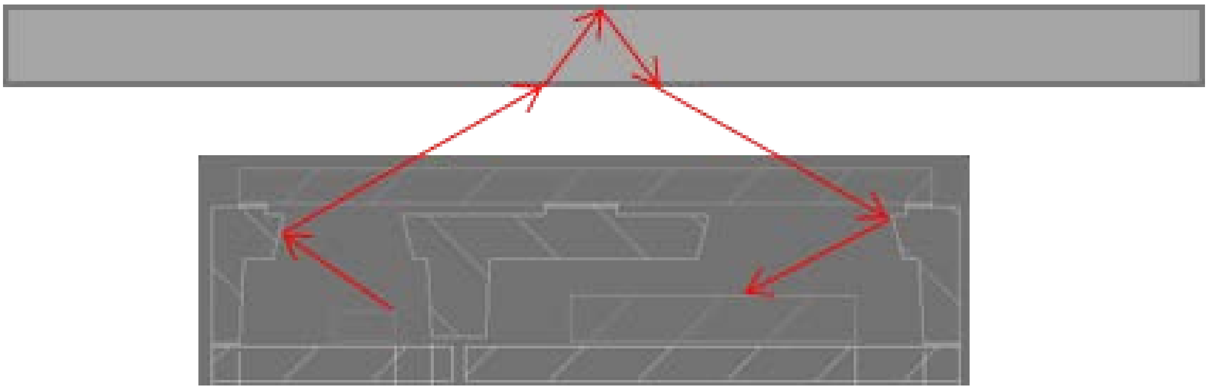

**Figure 4. Example Optical Crosstalk Light Path from Top Surface of Cover Glass**

# 3. Proximity Sensor (PS) Settings

OB1203 has configurable registers for PS settings including:

- Proximity measurement enable
- IR LED drive current
- ADC gain
- (micro) pulse width
- Number of (micro) pulses per measurement
- Measurement period
- Digital crosstalk subtraction
- Analog crosstalk cancellation
- Interrupt configuration
- Proximity Interrupt High threshold
- Proximity Interrupt Low threshold
- Moving average
- Hysteresis

After power on or assertion of the reset register, the device has no measurements enabled. Proximity measurements are enabled by asserting the PPG_PS_EN bit in the MAIN_CTRL_1 register (0x16), and setting the proximity sensor IR LED to current to a non-zero value.

*Note that the IR LED current setting for proximity (registers 0x17 and 0x18) are distinct from the register for setting the IR LED current for PPG heart rate measurements (registers 0x30 and 0x31).*

## 3.1 IR LED Current Setting

IR LED current is programmable by a 10-bit current DAC. The IR LED current is pulsed on only during the proximity sensor measurement micropulse duration and off otherwise. 0x3FF is the maximum register setting, nominally 250mA. The number of signal counts in the PS data register (0x02 and 0x03) scales with the LED current.

There lowest significant bits show some differential nonlinearity that results in occasional non-monotonic behavior (i.e. DAC is increased but the current is decreased slightly). For strictly monotonic behavior in the current DAC, use the upper 5 MSBs for current control. This is more than sufficient for proximity applications. The high resolution DAC is mainly intended for bio sensing applications.

LED output power is monotonic with LED current. Due to well-known thermal effects, the LED light output power efficiency decreases marginally at the highest current settings. For example, the output power at 250mA is typically 80% higher than the light output power at 125mA, rather than 100% higher.

Where very high current levels are used, some slow thermal drift may be seen in the end application. For typical applications, the LED current would be set between 125mA and 175mA to balance SNR and excess heat generation.

In an application where the first few PS measurements are used to estimate crosstalk for a crosstalk cancellation algorithm, the LED current setting should be chosen to limit the thermal drift to acceptable levels.

## 3.2 ADC Gain

The overall sensitivity of the ADC is set by the ADC gain control in the register (0x2E). This also sets the gain in PPG1 and PPG2 biosensing mode.

Higher gain provides higher count values and lower range. Gain applies equally to signal and noise. Lower gain provides small count values and wider range, with proportionally smaller noise. The lowest useful gain should be chosen for the application. Typically, the gain 2 setting is sufficient for proximity applications.

## 3.3 Pulse Width

Each PS measurements consists of a series of micropulses (simply called pulses in the datasheet). The duration of each pulse (width) is set in register 0x1A. Shorter pulses lengths result in a lower signal to noise ratio, meaning more of the displayed data bits are noise bits. Longer micropulse widths result in a higher signal to noise but increased sensitivity to rapidly changing light sources and higher power consumption.

The number of proximity counts in the PS data registers does not scale with the pulse width. Longer pulses reduce the noise relative to the signal, with the signal to noise ratio improving roughly as the square root of the pulse length.

Compared to increasing pulse width, higher SNR would be achieved by using higher LED current, since SNR scales (roughly) linearly with LED current, but only improves as the square root of the LED pulse width.

## 3.4 Number of Pulses

The number of (micro) pulses per measurement is set in register 0x19. The number of signal counts scales with the number of micropulses per measurement and with the LED drive current (counts from each pulse are summed). The SNR improves as the square root of the number of pulses, similar to the pulse width. Typically, 4 to 16 pulses are sufficient for proximity applications.

## 3.5 Measurement Period

PS measurement period (1/measurement rate) is set with register 0x1A. Longer measurement periods produce a slower samples rate and lower power consumption. Choose the slowest useful measurement period to minimize power consumption. Typically 50ms or 100ms is sufficient for handset applications.

## 3.6 Digital Crosstalk Subtraction

The digital crosstalk subtraction register removes a fixed number of counts from the PS measurement prior to displaying in the PS_DATA registers. The value is set in registers 0x1B and 0x1C. See the Dynamic Cancellation of Crosstalk section for an example procedure.

## 3.7 Analog Crosstalk Subtraction

If the cover glass is highly reflective and/or scattering (e.g. white ink) such that the crosstalk level is greater than 50% of full scale, analog front end current cancellation may be enabled to introduce a new “zero” level. The maximum PS count value is 65,535. If the count level is somewhat greater than 33,000 counts, then analog crosstalk cancellation can be used.

Analog current subtraction method is subject to the typical process variation. Assume a few percent variance in analog current subtraction level. For example, some devices may subtract ~34k counts. Others may subtract ~31k counts. With analog current subtraction enabled signals of up to ~1.5x full scale can be measured, meaning that crosstalk levels of up to ~32,676 counts can be subtracted without loss in dynamic range.

A small increase in noise level is observed with analog cancellation enabled due to the additional of another current source.

## 3.8 Interrupt Configuration

With PS interrupts enabled (see register 0x2C in the datasheet) the INT pin (commonly referred to as $\overline{INT}$ or INTB) is pulsed low. In default operation, interrupts are release when the status registers are read by the host. If logic mode interrupt is enabled, the INT pin is low if the previous measurement was greater than the high threshold or less than the low threshold and the interrupt remains asserted even if the status register is read. (The only way to de-assert the interrupt in logic mode is set the high threshold to max or low threshold to 0)

Interrupt persistence (register 0x2D) is useful to reduce false interrupts. The number of successive interrupt conditions must equal the persistence in order to assert the interrupt. Persistence increases fidelity at the expense of introducing a lag in the system response.

## 3.9 Thresholds

PS interrupt high and low thresholds are set in registers 0x25–0x26 and 0x28–0x29. The default values are 0xFFFF and 0x0000. With the default values, the interrupt will not assert. (The PS data value has to be greater than the high threshold or lower than the low threshold.)

Since the IR LED current, ADC gain and number of pulses all scale the output signal count value. The PS thresholds should be scaled accordingly when configurations are changed. PS_AVG and PS_WIDTH settings do not change the count value.

## 3.10 Moving Average

The moving average feature (register 0x1D) simply adds the current the previous PS samples and bit shifts once to the right. This reduces noise by $\sqrt{2}$ at the expense of a slight increase in response time.

## 3.11 Hysteresis

The hysteresis interrupt feature (0x1D) is enabled by setting its register bits to a nonzero count value. When enabled it tracks the maximum count value and sets the low threshold to the max count value minus the hysteresis register value. Once the count level drops below the low threshold the low level, the minimum count value it tracked and the high threshold is slaved to the minimum count value plus the hysteresis register value. This mode is useful for alerting on changes of target direction (in/out) to capture user behavioral dynamics with the need to constantly monitor the PS counts.

# 4. Basic Threshold Operation

Basic proximity sensor operation uses the following procedure:

1. Set IR current, pulse width, number of pulses, measurement period, PS averaging (if desired).
2. Set high threshold to 2000 counts (example).
3. Set low threshold to 0 counts (example).
4. Enable PS interrupts and PS measurement.
5. Host sleep.
6. If threshold interrupt asserts, wake host and read PS data.
   a. If PS data > PS high threshold, then a proximity event is detected. Set low threshold to 1400 counts (70% of prox engage counts), set high threshold to maximum (xFFFF). Turn on the screen.
   b. If PS data < PS low threshold, then a proximity event is over. Set low threshold to 0 counts. Set high threshold to 2000 counts. Turn off screen. Sleep host.

# 5. Dynamic (use case-based) Compensation of Crosstalk

Follow the example below to set the initial crosstalk level.

1. Set IR current, pulse width, number of pulses, PS averaging (if desired) and enable PS measurement.
2. Set measurement period to as low as possible.
3. Ignore the first few samples (LED stabilization time).
4. Average a few tens of ms of PS samples to obtain ps_avg.
5. Choose a target count level ps_target, such 100 counts. This value should be higher than the PS data noise level.
6. Calculate digital cancellation crosstalk level: ps_can = ps_avg-ps_target.
7. Write ps_target to the digital cancellation registers. The new count level should be close to ps_target.
8. Set the measurement period to the desired operational measurement period such as 50ms or 100ms.
9. Set PS high interrupt threshold at the level for detecting the approach of a user, such as 2000 counts.
10. Set the low PS interrupt threshold at a value less than the target minus the noise level. For example, if the noise is 25 counts and the target is 100, the threshold could be set at 50 counts.
11. Enable PS interrupts.

If the PS count level falls below the low threshold, then the crosstalk has reduced by a temperature decrease or mechanical change. Go to step 1 and proceed to set a new digital cancellation crosstalk level. This is necessary to prevent the digital cancellation from subtracting too many counts such that the proximity signal would be negative and not visible in the unsigned integer data register.

This can only be done when not in a proximity event.

As usual, when a proximity event is detected, the PS low threshold should be set to the desired proximity release threshold. This is typically 60–80% of the proximity detect threshold. Then the low interrupt asserts, we exit proximity mode and go to step 10.

A proximity event that creates a large smudge which causes the proximity signal to not be able to reach the low threshold must be detected algorithmically. One method is to detect attempted waking of the phone (home button or power button press) during a proximity event. This tells the host that the user is no longer in proximity. Go to step 1 to reset the crosstalk digital cancellation.

The above method also covers the use case of having the ps sensor covered when the crosstalk is calculated. As soon as the user removes the obstruction, the algorithm immediately adjusts the cancellation down to the new value.

# 6. Power Saving Strategies

A useful power saving strategy is to dynamically increase the power level when it is believed that a target may be present. This is a Bayesian technique that allows the system to take advantage of what is known to avoid having to make high fidelity measurements when the confidence in getting a positive result is low.

Persistence is intended to reduce false positives and is useful in situations where the noise level is small relative to the threshold. In this scenario, we are aggressively choosing low power consumption settings such that the relatively simplistic method of persistence is no longer a reliable algorithm for proximity detection.

The Bayesian method effectively boosts SNR without a significant increase in power consumption.

As an example the PS register settings are

- Measurement rate: 100ms
- IR LED current: 75mA
- LED pulse width: 22µs
- # of LED pulses: 4
- ADC gain: 4
- PS interrupt: normal threshold interrupts enabled
- Moving average: enabled
- No persistence
- Low threshold: 0 counts
- High threshold: 1000 counts

These settings condition results in a comparatively low measurement fidelity due to the low LED current, short LED pulse width and few number of pulses. As a result, the signal is noisy. However, the odds of the noise signal exceeding the high threshold with no object in proximity are still low.

When the high threshold asserts, the host immediately changes the settings to the following in order to make a rapid high fidelity measurement.

- Measurement rate: 3.125ms
- IR LED current: 150mA
- LED pulse width: 35µs
- # of LED pulses: 16
- ADC gain: 2
- PS interrupt: disabled
- Moving average: disabled

The gain scaled by ½, the number of LED pulses scaled by 4 and the current scaled by 2, so the count level should increase by $\frac{1}{2} \times 4 \times 2 = 4$. The LED pulse duration also increased which decreases noise without changing the signal count.

The host polls and averages a few (e.g. 4) measurements as needed to make a high fidelity measurement and compares with the scaled threshold (4 × 1000 = 4000 counts). If the new measurements exceed the scale threshold, the proximity event is confirmed and the system is restored to low power mode, with a low threshold of 500 counts and maximized high threshold counts. When the low interrupt asserts, we repeat the high fidelity measurement, compare with a scaled low count of 2000, and so forth.

Some tuning is needed to avoid too frequent assertion of the interrupt.

In this example, the high power setting consumes 8x the power of the low power setting. If we choose 4 measurements for our high fidelity comparison the overall power consumed is equivalent to 32 low power measurements. If a false interrupt asserts once every ten seconds we simply add 32 measurements to the 100 measurements made every 10 seconds. The power savings then is 1.32/8 or about 1/6$^{th}$ of the LED power consumption versus a high power setting at 100ms sample period. This is obviously a very beneficial method, but it neglects the processor power consumption. Including the processor power consumption typically shows that settings asserting a false interrupt every few minutes will still achieve significant power savings over a basic persistence-based method.

A driver employing this relatively simple Bayesian detection strategies can achieve power savings commensurate with higher performance “dumb” hardware. Further situational optimization is possible by enabling the driver to dynamically optimize the thresholds based on the ratio of false and true positives using a Kalman approach.

# 7. Revision History

| Revision | Date | Description |
|---|---|---|
| 1.1 | Apr.30.20 | Redefined document title. |
| 1.0 | Apr.16.20 | Initial release. |

## IMPORTANT NOTICE AND DISCLAIMER



### Corporate Headquarters

TOYOSU FORESIA, 3-2-24 Toyosu,
Koto-ku, Tokyo 135-0061, Japan
www.renesas.com

### Trademarks



### Contact Information

For further information on a product, technology, the most up-to-date version of a document, or your nearest sales office, please visit www.renesas.com/contact-us/.