Software Design Description (SDD)

Rev: A2

18 Oct 2013

ISL29038 sensor device driver For Intersil Corporation

VVDN Contact:

Bhupender Saharan

VVDN Technologies

+1 408 807 3951

bhupi@vvdntech.com

Revision History:

Date	Rev No.	Description	By
18th Sept. 2013	A1	VVDN_ISLU_SNSR_SDD_ISL29038_A1.pdf	VVDN

Table of Contents

Table of Figures

1 Introduction

This document describes software components of ISL29038 device driver for Android Jellybean (version 4.2) and Android Ice cream sandwich (version 4.0.4) which is to be developed by VVDN for Intersil Corporation.

This SDD is made for the reference of

1. Product managers and QAD at VVDN & Intersil to understand the software development phases.
2. Engineering Team at VVDN for System Architecture, Design and development of device driver.
3. System Integration and Verification teams at VVDN / Intersil for SW validation.

2 Scope of the document

This document describes the software design specification of ISL29038 device driver in detail.

3 Reference documents

S.NO	Description	Revision	Date
1	VVDN_ISLU_SNSR_SDD_ISL29038_A1.pdf	A1	18 Oct 2013

4 Overview of project

Intersil manufactures **ISL29038**, a Low power ambient light and Proximity sensor device with enhanced infrared rejection. It has a built-in IR-LED driver for proximity function. It is a low power, high sensitivity Ambient light sensor with an I2C (SM Bus compatible) interface. The scope of this project is to develop and test ISL29038 device driver on Android Operating system.

5 Software Requirement

5.1.1 Operating system requirement

The sensor device driver should be supported on following operating systems

1. Android Jelly Bean (version 4.2)
2. Android Ice cream sandwich (version 4.0.4)

5.1.2 Sensor Operations

The sensor device driver supports the following operations

1. Configure the ADC resolution separately for both ALS sensing and Proximity sensing
    a. A 12 - bit ADC is used to Digitize ALS output
    b. Proximity sensor uses an 8 – bit ADC which operates in same fashion as ALS do.

2. Configure device operation modes
    a. ALS Sensing
    b. Proximity Sensing

3. Configure the Separate High and Low interrupt thresholds for ALS and Proximity
4. Configure the Separate interrupt persistency for both ALS and Proximity
5. Configure the ALS range
    a. 125 Lux
    b. 250 Lux
    c. 2000 Lux
    d. 4000 Lux

6. Configure the Proximity sleep time
    a. 400 ms
    b. 100 ms
    c. 50 ms
    d. 25 ms
    e. 12.5ms
    f. 6.25 ms
    g. 3.125 ms
    h. 0 ms
7. Configure IR LED Current
    a. 31.25 mA
    b. 62.5 mA
    c. 125 mA
    d. 250 mA

8. Configure the Proximity offset compensation
9. Configure the ALSIR compensation

6 System Block Diagram

The Intersil ISL29038 sensor would be interfaced to a TI's OMAP4460 based Panda-board via an I2C and Interrupt line.

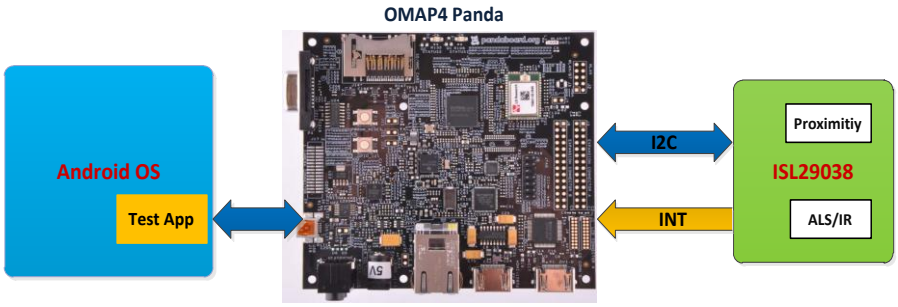


Figure 1 System block diagram

I2C interface would be used to

1. Configure the sensor device
2. Read sensor device (Light intensity or Proximity)

Sensor interrupt pin would be connected to an interruptible GPIO pin on Panda-board.

7 Software Architecture

The system software architecture diagram is shown below.

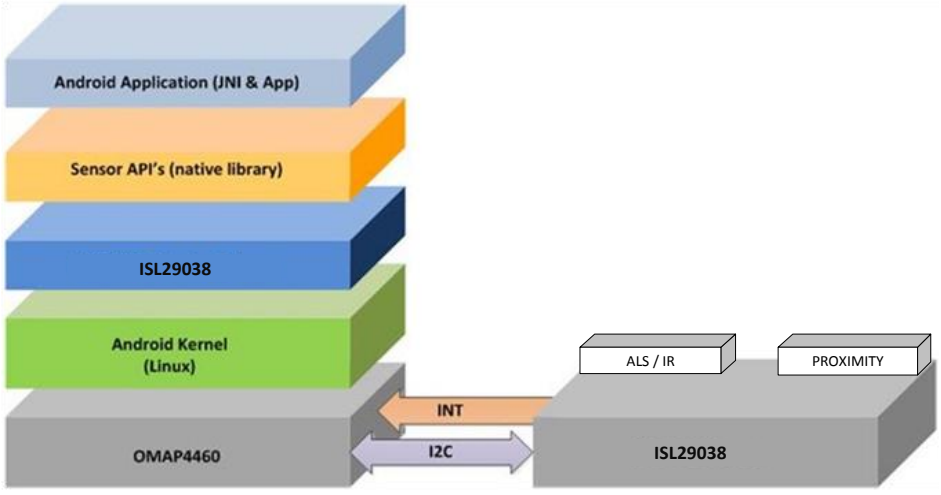


Figure 2 Software architecture diagram

8 Device driver use case model

The use case model for ISL29038 device driver is shown below. The functionalities provided by the device driver are as follows.

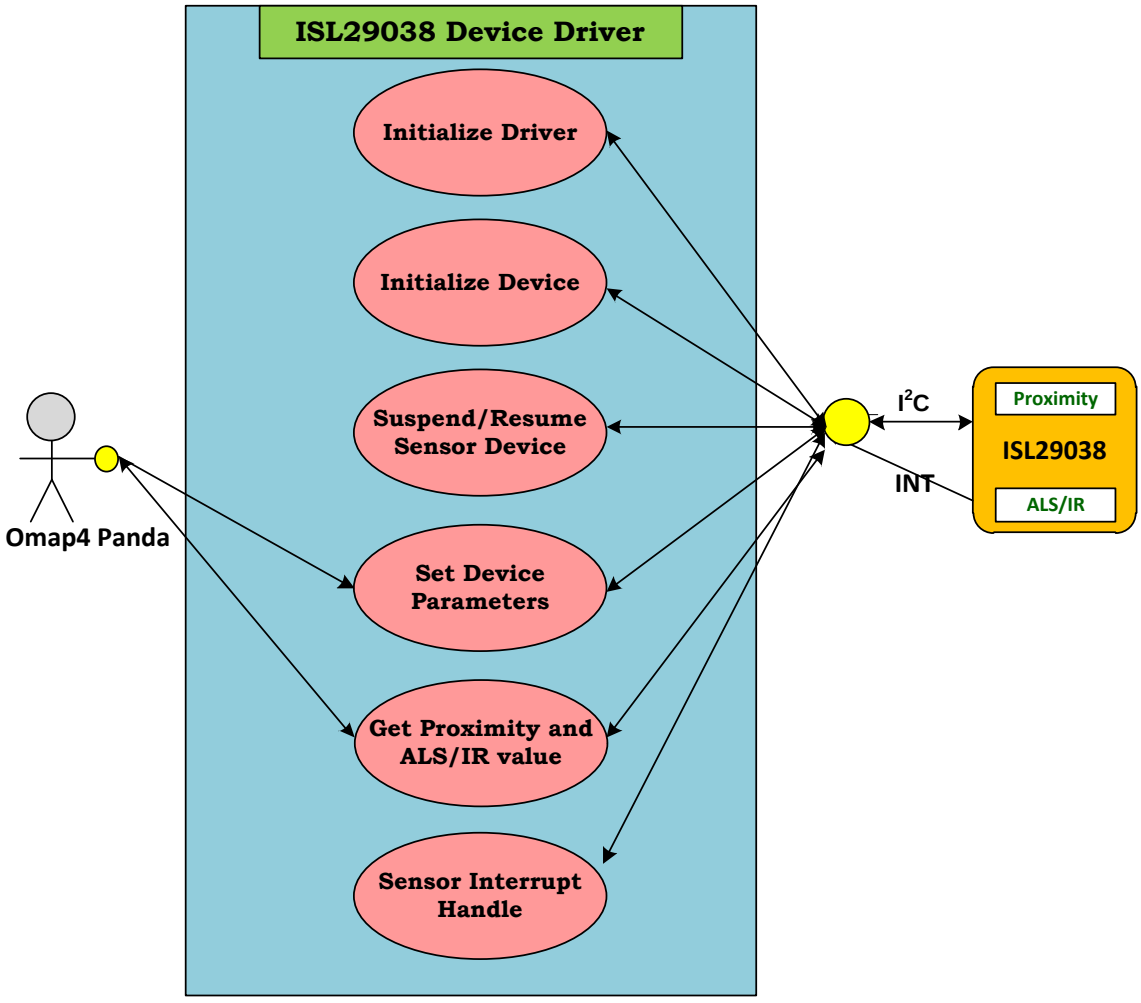


Figure 3 Sensor device driver use case diagram

8.1 Initialize the driver

The driver initialization code does the following

1. Registers the I2C device driver with the Linux I2C core framework.
2. Requests and set up the GPIO line (to be used as interrupt line) for handling the sensor interrupts.
3. Registers the Interrupt handlers for the IRQ number corresponding to the sensor interrupt line.

```
static int __init isl29038_init(void)
{
        return i2c_add_driver(&isl_sensor_driver);
}
```

8.2 Initialize the device

Once the I2C core framework recognizes the ISL29038 (slave address 1000100b or 0x44h) on the I2C bus. It calls the probe function of the device driver (that supports/services the ISL29038 device) which verifies the ALS/Proximity sensor device, and writes the initial configuration parameters to the device. At this stage the sensor device is ready to operate in the default operating conditions.

In the default configuration the device would work in ALS continuous mode and 16-bit ADC resolution. The Thresholds will be configured as follow

a. LOW threshold will be 5% of Maximum Range
b. HIGH threshold will be 20% of Minimum Range

8.3 Get Proximity and ALS/IR value

8.3.1 Proximity sensing mode

The Proximity sensing mode is enabled when ***PROX_EN = 1*** in CONFIGURATION register 0x01h. The Proximity value is read from the sensor device register PROX_DATA (Address 0x0Ah).

PROX_DATA is an 8-bit register; it stores the ADC conversion data for Proximity sensor. Reading PROX_DATA Register will give the 8-bit ADC code.

Add	Register	D7	D6	D 5	D 4	D3	D2	D1	D0
0x0Ah	PROX_DATA	0	0	0	0	0	0	0	0

8.3.2 ALS sensing mode

The Ambient light intensity (ALS) is read from the ISL29038 sensor device from the registers shown in the below table (Values are stored in 12-bit registers as MSB and LSB). The values are read when the user application sends an IOCTL request for reading ALS values.

REG Address	Register name	DATA
0x0Ch [7:0]	**ALS_DATA_LB**	**8-bit Data**
0x0Bh [3:0]	**ALS_DATA_HB**	**LSB Nibble**

Add	Register	D7	D6	D 5	D 4	D3	D2	D1	D0
0x0Ch	ALS_DATA_LB	0	0	0	0	0	0	0	0

Add	Register	D15	D14	D13	D12	D11	D10	D9	D8
0x0Bh	ALS_DATA_HB	0	0	0	0	0	0	0	0

|←----------MSB Nibble---------->|←----------LSB Nibble ---------->|

x – Reserved / Unused [0000]

The Lower Nibble of Address 0x0Bh is used as the upper 4-bits [11:8] of 12-bit ALS data. The Upper 4-bits of memory address 0x0Bh are unused and set to 0.

Conversion of Lux from ADC code :

E_{calc} = Range * ADC code / 4096

8.4 Set device parameters

The programming of device parameters is taken care by IOCTL implementation of the driver. Each parameter is assigned a unique identifier number based on which driver decides what action is to be taken.

The general sequence diagram for programming a device parameter from a test application is shown below.

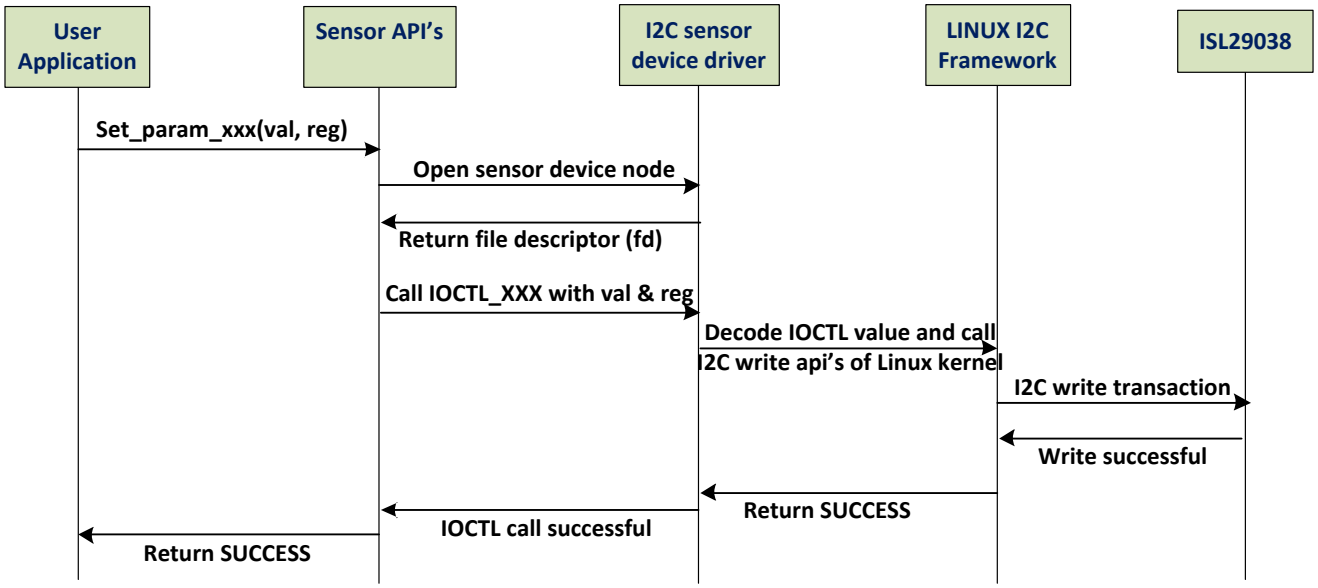


Figure 4 Sequence diagram for programming sensor device parameter

8.4.1 Device Operation mode

The device operation mode indicates the operating mode of the device. Sensor device supports the following operating modes.

- ALS sensing mode
- Proximity sensing mode

The device operating mode indicates the device state and which light's component is being converted by ADC. The device operation mode is programmed by writing CONFIGURATION Register (0x01h) or Register (0x02h) of sensor device.

a) ALS Sensing mode

The ISL29038 is set for ambient light sensing when Register bit ***ALS_EN = 1*** in register 0x02h.

Add	Register	D7	D6	D5	D4	D3	D2	D1	D0
0x02h	Config1	INT_ALG	PROX_OFFSET				ALS_EN	ALS_RANGE	

b) Proximity sensing mode

The Proximity sensing mode is enabled when ***PROX_EN = 1*** in Configuration register 0x00h. The Proximity value can be read from the Proximity sensor device register PROX_DATA (Address 0x0Ah).

User can put the sensor in Proximity mode by setting the PROX_EN bit.

Add	Register	D7	D6	D5	D4	D3	D2	D1	D0
0x00h	Config0	Reserved		PROX_EN	PROX_SLP			IRDR_DRV	

8.4.2 Range

The range of sensing defines the sensitivity range of the sensor device. There are four ranges for ALS sensing. The ALS measurement range is configured via Register 0x02[1:0].

1. ALS Sensing Ranges

1. 125 Lux [00]
2. 250 Lux [01]
3. 2000 Lux [10]
4. 4000 Lux [11]

This parameter is programmed by writing to the **ALS sensing range** [D1:D0] in Configuration register (0x02h) of sensor device.

Add	Register	D7	D6	D5	D4	D3	D2	D1	D0
0x02h	Config1	INT_ALG	PROX_OFFSET				ALS_EN	ALS_RANGE	

8.4.3 Interrupt threshold

The interrupt threshold value defines a window for Lux values. Whenever the current ADC measurement reaches above or below this threshold window, the sensor device raises an interrupt to the processor. There are separate thresholds for both Proximity and ALS/IR sensor.

8.4.3.1 Proximity Thresholds:

1. PROX_INT_TL (0x05h)
2. PROX_INT_TH (0x06h)

The Proximity threshold register is a 16-bit register. The sensor device address 0x05 and 0x06 together constitute a 16-bit register and can be programmed separately. Proximity low Interrupt threshold can be programmed by writing address 0x05 of sensor. Proximity high Interrupt threshold can be programmed by writing address 0x06.

Add	Register	D7	D6	D 5	D 4	D3	D2	D1	D0
0x05h	PROX_INT_LT	0	0	0	0	0	0	0	0
Add	Register	D15	D14	D 13	D 12	D11	D10	D9	D8
0x06h	PROX_INT_HT	1	1	1	1	1	1	1	1

8.4.3.2 ALS Threshold

The ALS interrupt event ALS_FLAG (Reg. 0x04h [3]) is governed by Reg. 0x07 through 0x09. Two-12 bit high and low threshold values are written to these registers. The ISL29038 will set the ALS interrupt flag if the ADC conversion count in Registers 0x0B and 0x0C are outside the programmed thresholds. The ALS_FLAG is cleared by writing a '0' to Reg. 0x04 [3].

The ALS threshold consists of separate 12-bit low threshold and 12-bit high threshold.

1. ALS_INT_TL [ALS_INT_TL1 (7:0)]$_{0x07}$ + [ALS_INT_TL0 (7:4)]$_{0x08}$
2. ALS_INT_TH [ALS_INT_TH0 (7:0)]$_{0x09}$+ [ALS_INT_TH1 (3:0)]$_{0x08}$

These parameters are programmed by writing to the following registers

Add	Register	D7	D6	D 5	D 4	D3	D2	D1	D0
0x07h	ALS_INT_TL1	0	0	0	0	0	0	0	0

|←----------Upper threshold byte of ALS_INT_TL----------→|

Add	Register	D3	D2	D 1	D 0	D11	D10	D9	D8
0x08h	ALS_INT_TLH	0	0	0	0	1	1	1	1

|←---Lower byte of ALS_INT_TL---→|←--MSB nibble of ALS_INT_TH--→|

Add	Register	D7	D6	D 5	D 4	D3	D2	D1	D0
0x09h	ALS_INT_TH0	1	1	1	1	1	1	1	1

|←----------Lower threshold byte of ALS_INT_TH----------→|

The Low threshold consists of register ALS_INT_TL0 [7:4] + ALS_INT_TL1 [7:0] making it 12-bit threshold register. The default value of lower threshold register is 0.

1. The High threshold consists of register [ALS_INT_TH0 (7:0)]$_{0x09}$+ [ALS_INT_TH1 (3:0)]$_{0x08}$ making it 12-bit threshold register. The default value of high threshold register is 0xfff.

8.4.4 Interrupt persistency

Interrupt persistency is used to avoid false interrupt cases due to noise or sudden spikes in the ambient light conditions. This parameter defines the 'N' number of consecutive interrupt threshold events after which the interrupt will be raised to processor and appropriate flags will be set.

The ISL29038 supports separate interrupt persistency (1/4/8/16) for Proximity and ALS. Both ALS and PROX have their own independent interrupt persistency options. ALS_PRST and PROX_PRST configuration are controlled from Reg. 0x04h.

1. The Persistency for Proximity is programmed by writing bits [D6:D5] of address 0x04.
2. The Persistency for ALS is programmed by writing bits [D2:D1] of address 0x04.

The following table shows the bits alignment of 0x04.

Add	Register	D7	D6	D 5	D 4	D3	D2	D1	D0
0x04h	INTConfig	PROX_FLAG	PROX_PRST		PWR_FAIL	ALS_INT_FLG	ALS_INT_PRST		INT_CFG

Proximity persistency values are shown in following table.

Bits 6:5	**Action**
00	INT after **1** Proximity Flag Event
01	INT after **2** Consecutive Proximity Flag Event
10	INT after **4** Consecutive Proximity Flag Event
11	INT after **8** Consecutive Proximity Flag Event

A proximity interrupt event (PROX_FLAG) is governed by the high and low thresholds in Reg. 0x05 and 0x06 (PROX_LT and PROX_HT) and is indicated by Reg. 0x04[7]. PROX_FLAG is set when the measured proximity data is more than the higher threshold. The proximity interrupt flag is cleared when the proximity data is lower than the low proximity threshold or by writing a ‘0’ to Reg. 0x04[7].

The Proximity interrupt generation can be selected between ‘out-of-window’ threshold and hysteresis schemes. When the PROX_INT_ALG register (Reg. 0x02, Bit 7) is set to 0, proximity uses a window comparator scheme; when set to 1, proximity uses a hysteresis scheme. In hysteresis mode, the interrupt event is generated if the proximity ADC count is higher than the PROX_HT threshold and the interrupt event is cleared when the proximity ADC count is less than the PROX_LT threshold. The interrupt event flag can also be cleared by writing a ‘0’ to Reg. 0x04[7].

ALS/IR persistency value is shown in following table.

Bits 6:5	**Action**
00	INT after **1** ALS Flag Event
01	INT after **2** Consecutive ALS Flag Event
10	INT after **4** Consecutive ALS Flag Event
11	INT after **8** Consecutive ALS Flag Event

The ALS interrupt event ALS_FLAG (Reg. 0x04[3]) is governed by Reg. 0x07 through 0x09. Two-12 bit high and low threshold values are written to these registers. The ISL29038 will set the ALS interrupt flag if the ADC conversion count in Registers 0x0B and 0x0C are outside the programmed thresholds. The ALS_FLAG is cleared by writing a '0' to Reg. 0x04[3].

8.5 Sensor device suspend/resume

The drivers suspend and resume routines does the following operation.

1. Put the sensor in standby mode when android OS sleeps
2. Wake the sensor from standby when android OS wakes up

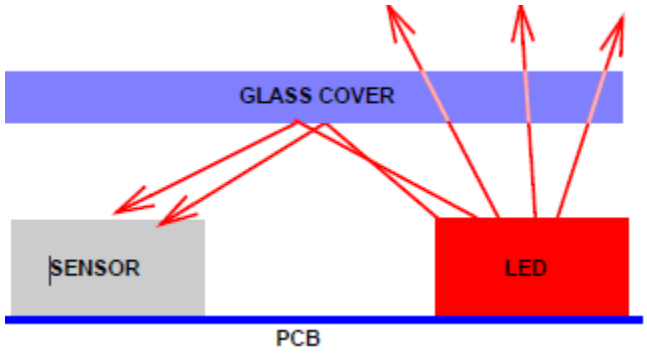
Both these events are taken care in the device driver by use of driver's suspend and resume callback functions.

8.6 ALS IR Compensation

The ISL29038 is designed for operation under dark glass cover. Glass or plastic covers can significantly attenuate visible light and pass the Infrared light without much attenuation. Consequently, the ISL29038 under a glass cover experiences an IR rich environment. The on-chip ALS passive optical filter on the ISL29038 is designed to block most of the IR incident on the ALS photo diodes. In addition, the ISL29038 provides a programmable active IR compensation that subtracts residual IR still reaching the sensor. The ALS_IR_COMP register (Reg. 0x03[4:0]) allows fine tuning of the residual infrared component from the ALS output.

8.7 Proximity Offset

Systems built with a protective glass cover over the ISL29038 can provide light 'leakage' or 'crosstalk' from the IR LED by reflection from the glass saturating the proximity sensor measurement system (Figure 11). Saturation can occur when the reflection from the glass with no object in the proximity detection space exceeds the full scale of the measurement system. The ISL29038 proximity system provides a user programmable proximity offset correction to compensate for this reflection.



The PROX_IR_COMP register (Reg. 0x02[6:3]) applies a corrective offset to the received signal prior to ADC conversion, which allows the signal to be brought within the usable range of the proximity measurement system.

8.8 Power-Up and 'Brown-Out' Reset

The ISL29038 has an enhanced power-on-reset system. A 'Brown-Out' detector flag in Reg. 0x04[4] informs the system that the device has powered-up properly. This flag should be reset as part of the initialization sequence. A 'Brown-Out' condition is defined as an operating condition when the power supply voltage is not within the specified limits. During the brown-out period at power-up, the I2C interface and the IR LED driver are inactive. Following brown-out, the I2C interface is re-initialized and the configuration registers are set to power-up default values. After power-up and during device initialization, host should examine that the PWR_FAIL flag (Reg. 0x04[4]) is set and then clear the flag by writing '0' to Reg. 0x04[4].

8.9 Power-Down

Setting ALS_EN (Reg. 0x02[2]) and PROX_EN (Reg. 0x01[5]) to '0' puts the ISL29038 into a power-down state with power supply current dropping to less than 1µA. All configuration registers are maintained in power-down mode.

8.10 Soft Reset

A Software reset to ISL29038 can be initiated by writing 0x38 to Reg. 0x0E. Following reset, all configuration registers are set to their default power-up state. After soft reset, the ISL29038 defaults to the power-down configuration.

8.11 Handle sensor interrupts

The ALS sensor device has the following sources of interrupt

1. ALS value exceeds the lower or upper threshold
2. Proximity value exceeds the lower or upper interrupt
3. ADC completed the conversion

The interrupt handling scheme is explained in the below flowchart.

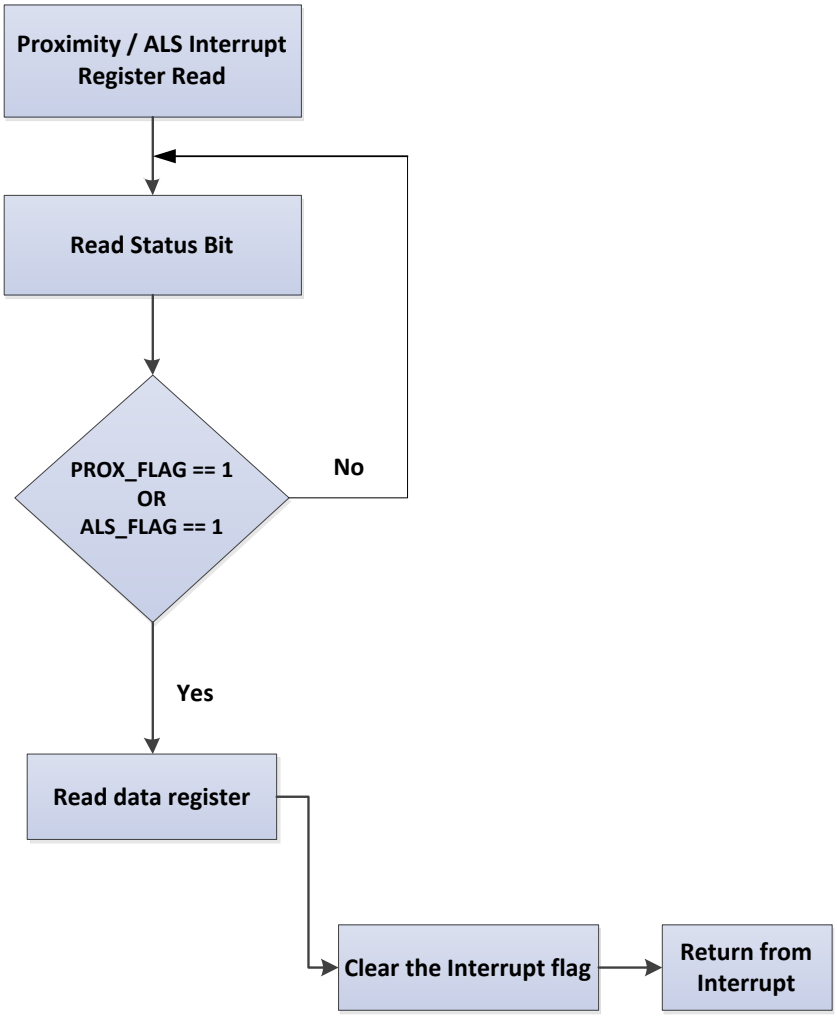


Figure 5 Sensor device interrupt handling

9 ALS sensor API's

Note: Detailed programmer API documentation for ISL29038 ALS sensor access would be provided along with the device driver and library code.