Application Programmer's Guide

Rev: A2

25 Sept. 2013

Android Application Programmer's Guide for

ISL29125 RGB sensor

VVDN Contact:

Bhupender Saharan
VVDN Technologies
+1 408 807 3951

bhupi@vvdntech.com

Revision History:

Date	Rev No.	Description	By
25 June 2013	A1	Android application programmer's guide_A1	VVDN
25 Sept 2013	A2	Android application programmer's guide_A2	VVDN

Table of Contents

1 Introduction

This document describes Application software development guide for "ISL29125 RGB sensor Driver" which is developed by VVDN for Intersil Corporation.

This Application Programmer's Guide is made for the reference of

- Product managers and QAD at VVDN & Intersil to understand the interface between android application and the sensor device driver.
- Engineering Team at VVDN for System Architecture, Design and development of android application software.
- System Integration and Verification teams at VVDN / Intersil for SW validation.

2 Scope of the document

This document describes the methods by which the android application can interact with the ISL29125 sensor device driver.

3 Android application interface to ISL29125 RGB Sensor Driver

The android application interacts with the ISL29125 device driver in the Linux kernel using the ***sysfs*** files exported by the device driver.

The android application reads / writes a string from /to the sysfs file to interact with the ISL29125 RGB sensor. The below table lists the sysfs files exported by the ISL29125 driver and possible input / output values.

Path of sysfs files in Linux kernel depends upon the following parameters

1. I2C Bus number in the target board
2. I2C Slave address of RGB sensor device

Sysfs path: /`sys/class/i2c-dev/i2c-<Bus_num>/device/<Bus_num>-<Slave addr>/`

Example: The ISL29125 RGB sensor is connected to I2C bus number 4 with slave address 0x44h (1000100b) to the pandaboard

Sysfs path: `/sys/class/i2c-dev/i2c-4/device/4-0044/`

SL	SYSFS FILE	R/W	DESCRIPTION	VALID R/W VALUES
1.	mode	R/W	RGB sensor mode control file used to set/get the modes of operation of sensor device	red green blue green.red.blue green.red green.blue pwdn standby
2.	red	R	Get red value	0 – 65535
3.	green	R	Get green value	0 – 65535
4.	blue	R	Get blue value	0 – 65535
5.	adc_resolution_bits	R/W	Set or get the ADC resolution bits	12 /16
6.	optical_range	R	Get current optical range of sensor device	330 / 4000
7.	active_ir_comp	R/W	Set or get active IR compensation	0 - 63
8.	adc_start_sync	R/W	Set or get conversion control status	risingIntb i2cwrite
9.	intr_persistency	R/W	Set or get interrupt persistency	1 / 2 / 4 / 8
10.	intr_threshold_assign	R/W	Assign interrupt threshold for particular color light	none green red blue
11.	intr_threshold_low	R/W	Assign Low interrupt threshold	0 – 65535

12.	intr_threshold_high	R/W	Assign high interrupt threshold	0 – 65535
13.	rgb_conv_intr	R/W	Enable or disable the RGB conversion interrupt	enable disable

Important Note:

1. Open the above sysfs files in the android application and read/write valid values as shown in above table for interaction with ISL29125 RGB sensor device driver.
2. Reading from a sysfs file returns number of bytes read in case of successful read else returns -1.
3. Writing to a sysfs file returns number of bytes written else returns -1.

3.1 CCT calculation used in the test android application

CCT can be calculated from the following values (values subject to change according to sensor).

Default value of the coefficients used to calculate CCT for ISL29125 are shown belows

alpha = 0.660723

beta = 1.156519

gamma = 2.049847

krg = 0.5254

krb = -0.08255

kgr = 0.127203

kgb = -0.51737

kbr = -0.1022

kbg = -0.28453

Formula used for CCT computation is shown below

Rp = alpha * (Red_Reading + krg * Green_Reading + krb * Blue_Reading)

Gp = beta * (kgr * Red_Reading + Green_Reading + kgb * Blue_Reading)

Bp = gamma * (kbr * Red_Reading + kbg * Green_Reading + Blue_Reading)

X = Rp / (Rp + Gp + Bp)

Y = Gp / (Rp + Gp + Bp)

n = (X - 0.332) / (Y - 0.1858)

CCT = n * (n * (- 449 * n + 3525) - 6823.3) + 5520.33 K