INTELLIGENT OPTO SENSOR **Number 29A**

DESIGNER'S NOTEBOOK

TAOS TEXAS ADVANCED OPTOELECTRONIC SOLUTIONS®

Using the Lux Equation

by Kerry Glover

August 2011

Introduction

Basic implementation of an Ambient Light Sensing (ALS) lux equation is typically shown in the data sheets. However, there are many questions related to the implementation. What is the maximum lux level that can be measured? What is the accuracy of the lux calculations due to the digital nature of the measurements? Is the light source too bright resulting in saturating the device? What happens when the device is placed behind dark glass? These are a few questions that are answered in this issue of the designer's notebook as related to the TSL2571, ALS only, and the TSL2771, ALS + Proximity sensing, devices from TAOS.

Naming Convention

It is assumed that the reader is familiar with the TSL2x71 data sheets. The naming convention in this document follows the data sheet, and the examples follow C coding conventions where words in capital letters represent variable names.

```
ATIME is the actual value programmed into the device
ATIME_ms = (256 - ATIME) * 2.720; // Value converted to milliseconds
```

Likewise for the AGAIN:

```
AGAIN is the two bit value programmed into the device
AGAINx is the lookup table from AGAIN showing the amount of gain
AGAINx will be 1,8,16 or 120 for the TSL2x71
```

The first segment of the lux equation from the TSL2x71 data sheet is as follows:

```
CPL = (ATIME_ms * AGAINx) / (GA * 53);
Lux1 = (C0DATA – (2 * C1DATA)) / CPL;
```

Where, CPL is Counts per Lux. C0DATA and C1DATA are the 16 bit data values or counts read from the device. GA is the Glass Attenuation factor. The value "53" is referred to as the Device Factor (DF) which is experimentally derived and will vary depending upon device type. In some cases, such as when a custom lux equation is developed, the DF and GA will be combined into a single value of GDF (Glass and Device Factor). Lux1 is the first segment of the lux equation covering fluorescent and incandescent light.

Also for this document, fluorescent will refer to both fluorescent lighting and white LED lighting since they both have very little IR content. When used, the term LED will refer to a "White LED".

Glass Attenuation Factor

For many applications, the TAOS device may be placed behind a glass or plastic housing. Throughout this document, the term glass will refer to glass, plastic or other semitransparent material placed over the device. In many of these applications, a Glass Attenuation (GA) factor can be added to compensate for the lower light level at the device. If the glass or plastic has too much spectral distortion, a new lux equation may need to be generated. See DN28 Developing a Custom Lux Equation for details on developing new lux equations.

The GA factor is intended to compensate for reduced light conditions. The GA factor is inversely proportional to the glass transmissivity, T. So, GA is defined as follows:

GA = 1/T.

For example, if the light is attenuated by 50%, then the glass is 50% transmissive, and the GA factor is 2 (1/0.5). With light attenuated by 95%, the glass is 5% transmissive, and GA factor is 20 (1/0.05).

Device Saturation

Before calculating lux, it is important to understand device saturation. There are two conditions for device saturation: analog saturation and digital saturation. Analog saturation is when the analog input is greater than what can be accumulated with the light-to-frequency conversion. Digital saturation is when the digital accumulator is overflowing before the analog saturates.

The full scale value for analog saturation depends upon the integration time programmed into the device. In saturation, the device accumulates 1024 counts for each 2.72 ms of integration time up to a maximum of 65,535 counts. Analog saturation will occur up to an integration time of 172 ms.

If the ALS integration time is greater than 172 ms (ATIMEx ≤ 64), digital saturation will occur before analog saturation. Digital saturation is when the count is 65,535.

```
if ( (256 – ATIME)  >  63)          // if ATIME_ms > 172ms (Digital Saturation)
    SATURATION = 65535;
else                                // if ATIME_ms <= 172ms (Analog Saturation)
    SATURATION = 1024 * (256 – ATIME).
```

Ripple Rejection

One of the first factors impacting the integration time decision is 50/60Hz ripple rejection. If the programmed integration time is in multiples of 10 ms and 8.3 ms (the half cycle time), both frequencies are rejected. An integration time value of 50ms (ATIME=0xED) or multiples of 50ms are required to reject 50/60 Hz ripple. In cases requiring faster sampling time as in proximity detection, averaging over a 50ms period may be needed to reject the fluorescent light ripple.

Ripple Saturation

Ripple Saturation is a second condition that impacts saturation. If there is ripple in the received signal, then the signal will fluctuate in and out of saturation, and the value read from CH0 will be less than the maximum but still have some effects of being saturated. Because of this, it is necessary to lower the gain if channel count values are above 75% of the saturated calculation. This is especially true in high gain mode. Under this condition, a channel reading may be slightly below the saturated calculation but in reality be saturated during the peaks resulting in a value less than the actual light level. At integration times > 240ms, digital saturation occurs before the analog saturation; therefore, this calculation would not be necessary.

```
SATURATION75 = SATURATION * 0.75; // if ATIME_ms < 240
```

The saturation check should be done before the lux calculation and return an overflow code if the device is saturated. Saturation is checked only against CH0 since the CH1 is always < CH0.

IR Adjusted Count and IR Factor

Recall from the TSL2x71 data sheet lux equation:

Lux = (C0DATA − 2 * C1DATA) / CPL;

where CPL is defined as Counts per Lux.

The equation can be rearranged as follows:

Lux = (C0DATA – 2 * C1DATA) / CPL;
Lux = (1 – 2 * C1DATA/C0DATA) * C0DATA / CPL;
Lux * CPL / C0DATA = 1 – 2 * C1DATA/C0DATA

By renaming C1DATA/C0DATA as RATIO and defining an Infrared Factor (IRF) as (1 – 2 * RATIO) and substituting, the lux equation above becomes the following:

RATIO = C1DATA/C0DATA;
Lux = (1 – 2 * RATIO) * C0DATA / CPL;
IRF = 1 – 2 * RATIO;
Lux = IRF * C0DATA / CPL;
IRF = Lux * CPL / C0DATA;

Light Sources

With the IRF, it is easy to see the reduction in count due to IR component of the light source. The RATIO varies depending upon the type of light present. For fluorescent light, the RATIO is around 0.1. For sunlight, the RATIO is around 0.25. For incandescent light, the RATIO is around 0.4. From this, the IRF for the different light types is listed as follows:

IRF ~ 1 – (2 * 0.1)	~ 0.8 Fluorescent/LED
IRF ~ 1 – (2 * 0.25)	~ 0.5 Sunlight
IRF ~ 1 – (2 * 0.4)	~ 0.2 Incandescent

From this, it can be seen that the CH0 count will be reduced to 20% for incandescent light, 50% in sunlight and 80% in fluorescent light. This is important in determining the maximum lux value.

Figure 1 shows data the ratio of data taken under different light sources. With a "single light" light sources, the ratio is a good indicator of light type and able to distinguish fluorescent from sunlight from incandescent. However, two light sources will combine linearly with a different ratio. For example, if a room has both incandescent and fluorescent light, they may produce the same ratio as sunlight.

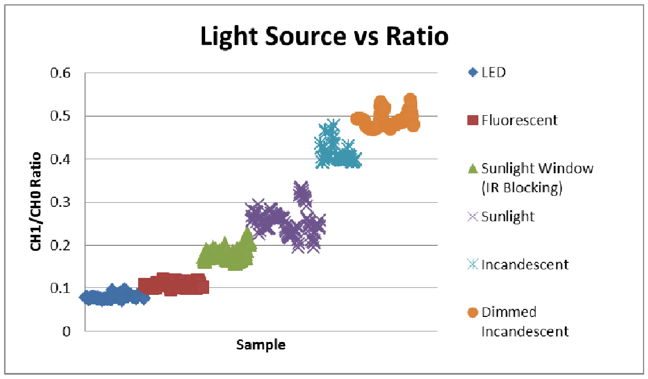

Figure 1: Light Source vs CH1/CH0 Ratio

Maximum Lux Value

Recall: Lux = IRF * C0DATA / CPL;

Replacing C0DATA with the SATURATION for the maximum count produces the following:

MaxLux = IRF * SATURATION / CPL;

For example assume an integration time of 200ms, a gain of 1, then:

CPL = 200 /53 = 3.25;

Under different lighting conditions, the maximum lux readings will be:

MaxLuxF = 0.8 * 65535 / 3.25; // ~ 14k lux in Fluorescent light
MaxLuxS = 0.5 * 65535 / 3.25; // ~ 8700 lux in Sunlight
MaxLuxI = 0.2 * 65535 / 3.25; // ~ 3500 lux in Incandescent light

Lux Accuracy

A simple formula can be used to determine the lux accuracy due to the Digitization Error (DER) present in the CH0 and CH1 increment by integer amounts. The following formula is used to calculate the lux accuracy:

DER = +/- 2 / CPL

For example, with an integration time of 200ms, a gain of 1 and GA of 1, the accuracy is:

CPL = 200 * 1 / 53 * 1 ~ 3.8;
DER = +/- 2 / 3.8 ~ +/- 0.53; // lux

It is recommended that CPL be greater that 5 to have a lux accuracy < +/- 0.5 lux.

Dark Glass Example

As an example, assume the system will operate behind 95% attenuating glass with the GA of 20. To compensate for this attenuation, AGAINx would need to be increase to 16x. With an integration time greater than 200ms, the SATURATION is 65k counts. From this, the count per lux, maximum lux, and lux accuracy under fluorescent light is calculated as follows:

CPL = 400 * 16 / (53 * 20) ~ 6;
MaxLuxF = 0.8 * 65k / 6 ~ 8700; // lux
DER = +/- 2 / 6 ~ +/- 0.3; // lux

With the implementation of an algorithm that automatically adjusted the gain, the maximum lux can be increase when the AGAINx is reduced to the 1x as follows:

CPL = 400 * 1 / (53 * 20) ~ 0.4;
MaxLuxF = 0.8 * 65k / 0.4 ~ 130,000; // lux
DER = +/- 2 / 0.4 = +/- 5; // lux

Efficient Lux Calculation

Many of the factors involved with the lux equation deal with making the calculations efficient. Eliminating floating point representation is the first goal. Since the lux value is calculated every time the device data value is sampled, minimizing the number of multiplies and divides is also important.

By multiplying the coefficients by 1000 and using the integration time in microseconds (ATIME_us = (256 - ATIME) * 2720;) floating point calculations are eliminated. The lux equation then becomes:

Lux1 = ((1000 * C0DATA) – (2000 * C1DATA)) / ((ATIME_us * AGAINx) / (GA * 53));

In the lux equation, only C0DATA and C1DATA variables change with every reading. The ATIME_us and AGAINx may seldom or never change; therefore, the denominator is calculated only when changes occur. The denominator value calculated is denoted as Counts per Lux (CPL). Using ATIME_us in microseconds, results in using counts per kiloLux (CPkL), this ensures integer value calculations. Therefore, CPkL and Lux1 equations are represented as follows:

CPkL = (ATIME_us * AGAINx) / (GA * 53);
Lux1 = (1000 * C0DATA − 2000 * C1DATA) / CPkL;

Note the (GA * 53) term should be pre-calculated as a single value to retain precision in the integer domain.

Extending the Lux Range Using CH1 Only

In bright light conditions, it may be beneficial to extend the lux measuring capability for the device. This occurs when the bright light condition causes CH0 saturate before the lux measurement is complete. The lux measurement range is extended by using CH1 counts in conjunction with a previous RATIO.

Recall the lux equation from the TSL2x71 data sheet lux equation section. The lux equation can be rearranged as follows:

Lux = (C0DATA/C1DATA -2) * C1DATA / CPL;
Lux = (1/RATIO – 2) * C1DATA / CPL;

The lux equation is a function of the RATIO and C1DATA. The algorithm is to recall the RATIO from the previous lux equation calculation when the current C0DATA is saturated. The previous RATIO is used with the current C1DATA to calculate the lux value. The assumption is that the RATIO will not change quickly and will remain the same until the light intensity decreases.

Recall, the RATIO will vary depending upon the type of light present. The RATIO for fluorescent light, sunlight, and incandescent light is typically 0.1, 0.25, and 0.4, respectively.

For maximum lux measurement, assume an integration time of 200ms and a gain of 1x; then the CPL = 3.25. Under different lighting types, the maximum lux calculations for fluorescent light, sunlight, and incandescent light, respectively, are as follows:

```
MaxLuxF = (1 / 0.1   – 2) * 65535 / 3.25; // ~ 160k lux in Fluorescent light using CH1 only
MaxLuxS = (1 / 0.25 – 2) * 65535 / 3.25; // ~ 40k lux in Sunlight using CH1 only
MaxLuxI  = (1 / 0.4   – 2) * 65535 / 3.25; // ~ 10k lux in Incandescent light using CH1 only
```

Calculating Lux for the TSL2581

The lux equation for other TAOS products may be presented in a slightly different form. For example, the first segment of the lux equation for the TSL2581 (ODFN package) is as follows:

Lux = 0.130 * CH0 − 0.240 * CH1; // Valid for an integration time of 400ms and a gain of 1x.

This can be translated into a format similar to the TSL2571 as follows:

Lux = (0.130 * CH0 − 0.240 * CH1) * 400 / (ATIME_ms * AGAINx);
Lux = (CH0 – 1.85 * CH1) / (ATIME_ms * AGAINx / 52);
Lux = (CH0 – 1.85 * CH1) / CPL;
CPL = (ATIME_ms * AGAINx / 52);

Other segments can be normalized multiplying them by CPL at 400ms and 1x gain:

CPL = 400 * 1 / 52 = 7.7

For CH1/CH0 = 0.00 to 0.30	Lux = (CH0 – 1.85 CH1) / CPL;
For CH1/CH0 = 0.30 to 0.38	Lux = (1.27 * CH0 – 2.74 * CH1) / CPL;
For CH1/CH0 = 0.38 to 0.45	Lux = (0.75 * CH0 – 1.37 * CH1) / CPL;
For CH1/CH0 = 0.45 to 0.54	Lux = (0.477 * CH0 − 0.769 * CH1) / CPL;
For CH1/CH0 > 0.54	Lux = 0

Calculating Lux for the TSL2561

The following shows an example of calculating the maximum lux for the TSL2561. The following is a summary of the multi-segment lux equation for the TSL2561 (FN package) using an integration time of 400ms and a 16x gain.

For CH1/CH0 = 0.00 to 0.125	Lux = 0.0304 * CH0 − 0.0272 * CH1;
For CH1/CH0 = 0.125 to 0.25	Lux = 0.0325 * CH0 − 0.0440 * CH1;
For CH1/CH0 = 0.25 to 0.375	Lux = 0.0351 * CH0 − 0.0544 * CH1;
For CH1/CH0 = 0.375 to 0.5	Lux = 0.0375 * CH0 − 0.0624 * CH1;
For CH1/CH0 = 0.5 to 0.61	Lux = 0.0224 * CH0 − 0.031 * CH1;
For CH1/CH0 = 0.61 to 0.8	Lux = 0.0128 * CH0 − 0.0153 * CH1;
For CH1/CH0 = 0.8 to 1.3	Lux = 0.00146 * CH0 − 0.00112 * CH1;
For CH1/CH0 > 1.3	Lux = 0;

The first segment can be translated into a common format as follows:

Lux = (0.0304 * CH0 – 0.0272 * CH1) * 400 * 16 / (ATIME_ms * AGAINx);
Lux = (CH0 – 0.895 * CH1) / (ATIME_ms * AGAINx / 194);
Lux = (CH0 – 0.895 * CH1) / CPL;
CPL = (ATIME_ms * AGAINx / 194);

Multiplying all of the original equations by CPL at 400ms and 16x gain gives:

CPL = 400 * 16 / 194 = 33

For CH1/CH0 = 0.00 to 0.125	Lux = (CH0 – 0.895 * CH1) / CPL;
For CH1/CH0 = 0.125 to 0.25	Lux = (1.07 * CH0 – 1.45 * CH1) / CPL;
For CH1/CH0 = 0.25 to 0.375	Lux = (1.15 * CH0 – 1.79 * CH1) / CPL;
For CH1/CH0 = 0.375 to 0.5	Lux = (1.26 * CH0 – 2.05 * CH1) / CPL;
For CH1/CH0 = 0.5 to 0.61	Lux = (0.74 * CH0 – 1.02 * CH1) / CPL;
For CH1/CH0 = 0.61 to 0.8	Lux = (0.42 * CH0 – 0.5 * CH1) / CPL;
For CH1/CH0 = 0.8 to 1.3	Lux = (0.048 * CH0 – 0.037 * CH1) / CPL;
For CH1/CH0 > 1.3	Lux = 0;

Two Segment Lux Equations for the TSL2561

The following is a simplified two segment lux equation for the TSL2561:

CPL = (ATIME_ms * AGAINx / 200);
Lux1 = (CH0 – 1.5 * CH1) / CPL
Lux2 = (0.40 * CH0 − 0.48 * CH1) / CPL;
Lux = Max(Lux1,Lux2,0);