# Flexible Flash Current Programming with AS2Cwire- and I2C-Enabled Flash LED Drivers

## Introduction

Traditional photo flash LED applications used in cellular phone handsets, smartphones and digital camera modules are typically set with a high current flash amplitude that is fixed by the circuit hardware configuration and controlled via a GPIO flash enable signal. Many LED flash applications can benefit from the ability to adjust the flash LED current as system conditions change. This application note describes dynamically programmable flash current control, implemented in several Skyworks flash LED driver devices, and some of the benefits this brings to the end products as highlighted below:

1. **Current drain reduction on the battery during low battery voltage operating conditions.**

   Smartphone application processors continuously monitor the battery input supply voltage. Should the voltage drop below a predetermined threshold during flash LED operation, the system, through the AS2Cwire or I2C data bus, can re-program the flash driver to a lower current level to decrease the demand on the failing voltage supply. Without doing so, the system would be in danger of a voltage drop-out condition which would shut down the device.

2. **LED temperature control for safety in over-temperature operating conditions or to adjust the flash LED light output characteristics.**

   LED power dissipation in high current flash applications is often a concern since an LED can be damaged if allowed to overheat during a flash event. To prevent overheating, a thermistor or other temperature sensing device is monitored by the application processor allowing it to shut down the flash LED if an over-temperature condition occurs. An over-temperature flash LED shutdown can be avoided by the application processor by dynamically adjusting the flash LED current to a reduced level via the AS2Cwire or I2C interface.

   Flash LED color temperature is directly related to the forward current through the flash LED. Different camera sensors under certain conditions can benefit from the ability to adjust the LED forward current to affect a change in LED color temperature, which is made possible by reprogramming the driver constant current sink.

3. **Provide exposure compensation for the camera sensor**

   Varying ambient light conditions place different demands on the camera sensor and subsequently the LED flash. Many systems require the ability to add daylight fill-in flash during a picture exposure. With varying ambient light conditions, the LED flash brightness must be adjusted to compensate. AS2Cwire or I2C interface programming provides such flash LED flexibility, once the flash event is executed via the hardware GPIO enable function.

4. **Adjust flash LED brightness for other accessory functions such as operating flash light applications, emergency strobe beacons, etc.**

   System application software may utilize the flash LED hardware for functions not related to the camera sensor. Such application could include flash light features, emergency strobe lights and fashion lighting. These functions may trigger flash LED operation through the hardware enable pin, but must reduce the flash LED current to a lower level than what may be present by the flash driver hardware. By reprogramming the flash LED to a lower current level, software applications can dynamically adjust the flash LED current, through AS2C or I2C programming, to an optimal level for the implemented feature.

Many Skyworks flash LED driver ICs are capable of dynamic current adjustment after a flash enable signal is asserted though their AS2Cwire or I2C interfaces. Such devices include, but are not limited to the AAT1270, AAT1271, AAT1271A, AAT1272, AAT1274, AAT1278, AAT1281, AAT1282, AAT1290, AAT1290A. This application note will describe Skyworks flash driver IC operation and programming with application examples.
Flexible Flash Current Programming
with AS²Cwire- and I²C-Enabled Flash LED Drivers

Operation
Flash LED drivers listed in this application note are comprised of three basic operational blocks: the DC/DC switching boost converter, a control block that incorporates data registers with discrete logic inputs as well as either an AS²Cwire or I²C interface, and a constant current sink(s) to regulate the LED drive current. These flash LED drivers permit two modes of LED operation and are controlled with a fixed functional hierarchy: high current flash mode and a movie/torch mode. Movie mode operation is programmed and enabled exclusively through the AS²Cwire or I²C interface. In contrast, the high current flash mode is triggered by an external flash enable signal to the flash enable (FLEN) pin. Asserting the FLEN signal will force the LED current sink to operate at the maximum current level as determined by the R_SET resistor. When the FLEN signal is applied, any previously programmed movie mode operation is halted giving priority to the high current flash function.

Flash LED constant current sink current levels are determined by two variables for high current flash operation and three variables for movie mode operation.

High Current Flash Mode:
1. The maximum flash current level ($I_{FL(MAX)}$) is fixed by the value of the external $R_{SET}$ resistor
2. The default current setting register is set to 100%
   a. This value may be changed through the AS²Cwire or the I²C interface.

Movie Mode:
1. The maximum current level is fixed by the value of the external $R_{SET}$ resistor ($I_{FL(MAX)}$)
2. Flash to movie mode ratio setting, as programmed by the Flash/Movie Mode Current Ratio register programming bits
3. Movie mode current, as programmed by the Movie Mode Current register programming bits
   a. Items (2) and (3) are values programmed through the AS²Cwire or I²C interface.

Movie Mode Operation:
To enable movie mode, the user must program and enable the flash LED through the AS²Cwire or the I²C interface by writing to the movie mode current ratio and movie mode current registers, followed by writing to the current sink enable register. Movie mode operation will remain on until the current sink enable bits are cleared. The default movie mode current ratio register setting is 1:7.3, with the default movie mode current register set at 100%. Refer to the respective product datasheet for additional movie mode programming information.

\[ I_{MOVIE_MODE} = \frac{I_{FL(MAX)}}{Movie \ Mode \ Ratio} \cdot \text{Movie Mode Current Data} \]

Where,

\[ I_{MOVIE_MODE} \] is the Movie Mode current and Movie Mode Ratio and Movie Mode Current Data are values in the registers.

Using the default register values, the default movie mode current is given by:

\[ I_{MOVIE_MODE \ (Default)} = \frac{I_{FL(MAX)}}{7.3} \cdot 100\% \]
**Flexible Flash Current Programming**
**with AS2Cwire- and I2C-Enabled Flash LED Drivers**

![Diagram](image)

**Figure 1: Movie Mode Current Programming Timing Diagram.**

**Normal High Current Flash LED Operation:**

A high current flash event is executed by initiating a flash enable signal logic high that is applied to the flash enable pin (FLEN). Regardless of preprogrammed movie mode operation or IC standby state, applying the FLEN signal will force the internal controller to bypass the movie mode ratio register and set the movie mode current register to 100%, permitting the maximum flash current level, as fixed by the $R_{SET}$ resistor, to be applied to the flash LED current sink.

\[
I_{FL(MAX)} = \frac{I_{FL(MAX)}}{100} \cdot \text{Current Data}
\]

Where,

$I_{FL MODE}$ is the Flash Mode current. Again, using the default register value, the default Flash Mode maximum current is given by:

\[
I_{FL MODE} (Default) = I_{FL(MAX)} \cdot 100\%
\]

Clearly, the default flash mode current equals the maximum flash current.

![Diagram](image)

**Figure 2: Timing Diagram For Normal High Current Flash Mode Operation.**
Adjustable High Current Flash LED Operation:

The high current flash LED level may be changed by writing to the Movie Mode Current register after the FLEN signal has been asserted. This is made possible by the fact that the movie mode ratio register is removed (ratio fixed at 1:1) from the current sink control when the FLEN signal is asserted. At this time, the movie mode current register is reset to 100% to permit the full \( R_{SET} \) programmed flash current. The high current flash control response time is 1µs. Once the flash function is initialized, the system controller can write to the movie mode current register to select one of 16 levels, to reduce the real-time flash LED current.

New data to change the Flash LED current via the Movie Mode Current register from 100% to a reduced level may be written \( \geq 1\mu s \) after the FLEN signal is asserted. A new flash mode current level is determined by applying the new movie mode current register data value to equation 2. When the flash event is terminated by transitioning the FLEN signal to a logic low level, all register data programming is reset to their respective default states.

Flash Current Programming

AS\(^2\)Cwire Interface:

Writing movie mode current data through the AS\(^2\)Cwire interface \( \geq 1\mu s \) after FLEN goes is asserted will change the flash LED current during the flash mode period from \( I_{FL(MAX)} \) at a 100% multiple to \( I_{FL(MAX)} \) at the reduced multiple.

![Flash LED Current Programming Timing](image)

Figure 3: Flash LED Current Programming Timing.

Figure 3 shows the timing plot of AS\(^2\)Cwire flash current re-programming during a high current flash on-time interval. 40µs after the FLEN signal is asserted, the flash current sink is enabled and begins rising to the \( I_{FL(MAX)} \) level. When 1µs has elapsed after the application of FLEN, a new movie mode current register setting may be programmed into the device. After the data write and corresponding data latch time (typically \( \leq 500\mu s \)), the flash LED current sink will be set to the new reduced current level as shown in equation 2.

The time from the last rising edge of the clocked data to the execution of the current level update is bounded by the AS\(^2\)Cwire data latch time. Figure 4 shows the \( t_{EN/SET(LAT)} \) variation over input voltage range. The latch time can vary over input voltage and the operating temperature range for a given device.
Flexible Flash Current Programming with AS²Cwire- and I²C-Enabled Flash LED Drivers

EN/SET Latch Timeout vs Input Voltage

**Figure 4:** $t_{\text{EN/SET(LAT)}}$ vs Input Voltage*.

* Note: The typical data characteristic curve from the AAT1274 is shown. Refer to the specific product datasheet for the relevant latch time characteristics.

**I²C Interface:**

Some devices in this product family employ an industry standard the I²C interface for function programming. When the FLEN signal is applied, the flash current sink will begin to turn on within 40µs. Writing movie mode current data through the I²C interface ≥1µs after FLEN is asserted will change the flash LED current during the flash mode period from $I_{FL(\text{MAX})}$ at a 100% multiple to $I_{FL(\text{MAX})}$ at the reduced multiple. For the plot shown in figure 5, the default flash current of 750mA is re-programmed to 400mA. The I²C data is clocked in 1µs after the rising edge of FLEN. The resulting flash LED current change is executed 150µs following the movie mode data register update.

**Figure 5:** Normal High Current Flash Enable And Reprogrammed Flash Current Via I²C Programming.
Reprogrammed Reduced Current Level Example (AAT1274)

Table 1 shows the specific AS^2 Cable Movie Mode Current Data register for the AAT1274. Depending on the number of rising edges clocked to the EN/SET pin 1µs after the rising edge of FLEN, the flash LED current will be adjusted to the percentage value of I_{FL(MAX)} based on the written code.

<table>
<thead>
<tr>
<th>Data (Number of EN/SET Edges)</th>
<th>Percentage of Maximum Current</th>
<th>Data (Number of EN/SET Edges)</th>
<th>Percentage of Maximum Current</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>100%</td>
<td>9</td>
<td>40%</td>
</tr>
<tr>
<td>2</td>
<td>89%</td>
<td>10</td>
<td>36%</td>
</tr>
<tr>
<td>3</td>
<td>79%</td>
<td>11</td>
<td>32%</td>
</tr>
<tr>
<td>4</td>
<td>71%</td>
<td>12</td>
<td>28%</td>
</tr>
<tr>
<td>5</td>
<td>63%</td>
<td>13</td>
<td>25%</td>
</tr>
<tr>
<td>6</td>
<td>56%</td>
<td>14</td>
<td>22%</td>
</tr>
<tr>
<td>7</td>
<td>50%</td>
<td>15</td>
<td>20%</td>
</tr>
<tr>
<td>8</td>
<td>45%</td>
<td>16</td>
<td>0%</td>
</tr>
</tbody>
</table>

Table 1: AAT1274 Address 0 Movie Mode Current Data Register Values.*

* Note: The data interpretation can vary by device. Please refer to the relevant product datasheet movie mode current data table for the specific device under consideration.

Referring to Equation 2, a range of reduced flash LED current levels can be determined based on the hardware set I_{FL(MAX)} value. The following assumptions are used for this example:

- R_SET = 107kΩ
- I_{FL(MAX)} = 1.5A
- Desired reduced flash current level = 500mA

Flash Mode Current = \( \frac{I_{FL(MAX)}}{1} \cdot \text{Movie Mode Current Data} \)

Initial Flash Current = \( \frac{1.5A}{1} \cdot 100\% = 1.5A \)

Reprogrammed flash current using data code 11 for a reduction to 32% of I_{FL(MAX)} (1.5A) = 480mA

Based on this example, the flash LED current can be adjusted over a range from the initial 1.5A maximum level to 20% of I_{FL(MAX)}, or 300mA, in 15 steps when referencing Table 1.
AAT1274 Bench Measurements

The following figures 6, 7 and 8 show AAT1274 measurements of the AS²Cwire update to the movie mode register and the resulting change in the ILED current. Figure 6 shows 4 rising edges clocked to the EN/SET pin after the FLEN signal has been applied. The resulting flash current is changed from 1A to approximately 710mA (71% of max current) as indicated by Table 1.

![Figure 6: Flash LED Current Reprogramming Example (AAT1274, RSET = 162kΩ).](image1)

The flash LED current may be programmed or reprogrammed at any time after the rising edge of the FLEN signal as long as the FLEN is held at a logic high state and the flash safety timer has not expired. Figure 7 shows an example of multiple programming data writes to the movie mode current register during a single flash event.

![Figure 7: Flash LED Current Programming Example.](image2)
Flexible Flash Current Programming
with AS3Cwire- and I2C-Enabled Flash LED Drivers

Figure 8 shows an example of the LED current being controlled by EN/SET with the addition of the flash inhibit (FLINH) signal during a flash event. Under normal operation, the AAT1274 detects logic high signal when applied to the FLEN pin. Within 1μs, the device disconnects the flash/movie mode current ratio register and resets the movie mode current data to 100% to generate a default 1A LED current as set by external RSET resistor (RSET = 162kΩ).

For this example, the AAT1274 is enabled for a 1A flash event, then reprogrammed to 500mA when the device detects 7 rising edges on the EN/SET signal and adjusts the flash LED current accordingly.

When a flash inhibit signal (FLINH) is applied to the device by the system controller, the LED current is reduced to the true movie mode level by re-applying the movie mode ratio divisor of 1/7.3 to the previous movie mode register setting. In this case, the effective FLINH flash LED current will be changed to ILED = (1A/7.3) · 0.5 ≈ 68mA. As soon as the FLINH pin is released to a logic low level, the LED current returns to the previously programmed 500mA level, provided that the flash event timer has not expired.

![Figure 8: An Example Of A Re-Programed Flash Led Current With The Flash Inhibit (FLINH) Function Asserted During A Flash Event.](image)

Reprogrammed LED Flash Current for Thermal Management

During a high current flash event, the flash LED power dissipation can exceed safe operating conditions for the package specifications if the applied flash current is:

a) Set for a level above that rated for the flash LED
b) The duration of the flash event becomes extended (flash on time)
c) High ambient operating conditions reduce the thermal headroom for the default flash operation

For any of the cases, if the system can monitor the flash LED temperature, it may take action to reduce the LED temperature by reprogramming the flash current to a reduced level. Figure 9 shows an example of a flash LED temperature versus forward current characteristic. As the applied forward current rises, the LED temperature will increase accordingly. Excessive thermal stress to the flash LED can reduce performance and in extreme cases permanently damage the flash LED.
Flexible Flash Current Programming with AS^2Cwire- and I^2C-Enabled Flash LED Drivers

For systems that monitor flash LED temperature, the flash LED current can be adjusted during a flash event to a lower level to decrease the LED power dissipation and resulting thermal effects. Figure 10 shows an example of a default flash LED operated at 500mA and resulting LED temperature of 149°C. The Flash LED current is then reprogrammed during the flash event for 300mA and 100mA as shown in Figures 11 and 12. The flash LED temperature is reduced from 149°C to 84°C and 42°C respectively. From these pictures, it is clear that reducing the LED current through the AS^2Cwire or I^2C interface as described earlier, can allow system designers to control the operating temperature of the flash LED and limit a potential degradation in system performance.
Flexible Flash Current Programming with AS²Cwire- and I²C-Enabled Flash LED Drivers

Conclusion

By utilizing the ability to dynamically reprogram high current flash LED functions, system designers can overcome issues created by flash LED current levels that are fixed by hardware implementations. By reprogramming preset flash LED current levels, system controllers used in Smartphones, camera modules and other camera enabled portable devices can:

a) Reduce current drain on the battery during low battery voltage operating conditions.

b) Control LED temperature for safety in over-temperature operating conditions or adjust the flash LED light output characteristics.

c) Provide exposure compensation for camera sensors

d) Adjust flash LED brightness for software driven accessory functions such as operating flash light applications, emergency strobe beacons, etc.

Many Skyworks flash LED driver ICs are capable of dynamic current adjustment after a flash enable signal has been asserted though their integrated AS²Cwire or I²C interfaces. Such devices include, but are not limited to the AAT1270, AAT1271, AAT1271A, AAT1272, AAT1274, AAT1278, AAT1281, AAT1282, AAT1290, AAT1290A. For all AS²Cwire enabled flash LED drivers, the flash LED current can be updated in the time bounded by the data write and data latch time 1µs after the manual flash enable has been asserted. For devices utilizing an I²C interface, current changes can be affected less than 250µs after the data write is initiated. These fast current adjustment times allow flexible programing and reprogramming flash LED current from hardware fixed levels as needed during flash events. For LED current programming details and data timing specifications, refer to the product datasheet for the relevant device.
Flexible Flash Current Programming
with AS²Cwire- and I²C-Enabled Flash LED Drivers

Copyright © 2012 Skyworks Solutions, Inc. All Rights Reserved.

Information in this document is provided in connection with Skyworks Solutions, Inc. (“Skyworks”) products or services. These materials, including the information contained herein, are provided by Skyworks as a service to its customers and may be used for informational purposes only by the customer. Skyworks assumes no responsibility for errors or omissions in these materials or the information contained herein. Skyworks may change its documentation, products, services, specifications or product descriptions at any time, without notice. Skyworks makes no commitment to update the materials or information and shall have no responsibility whatsoever for conflicts, incompatibilities, or other difficulties arising from any future changes.

No license, whether express, implied, by estoppel or otherwise, is granted to any intellectual property rights by this document. Skyworks assumes no liability for any materials, products or information provided hereunder, including the sale, distribution, reproduction or use of Skyworks products, information or materials, except as may be provided in Skyworks Terms and Conditions of Sale.

The materials, products and information are provided “AS IS” without warranty of any kind, whether express, implied, statutory, or otherwise, including fitness for a particular purpose or use, merchantability, performance, quality or non-infringement of any intellectual property right; all such warranties are hereby expressly disclaimed. Skyworks does not warrant the accuracy or completeness of the information, text, graphics or other items contained within these materials. Skyworks shall not be liable for any damages, including but not limited to any special, indirect, incidental, statutory, or consequential damages, including without limitation, lost revenues or lost profits that may result from the use of the materials or information, whether or not the recipient of materials has been advised of the possibility of such damage.

Skyworks products are not intended for use in medical, lifesaving or life-sustaining applications, or other equipment in which the failure of the Skyworks products could lead to personal injury, death, physical or environmental damage. Skyworks customers using or selling Skyworks products for use in such applications do so at their own risk and agree to fully indemnify Skyworks for any damages resulting from such improper use or sale.

Customers are responsible for their products and applications using Skyworks products, which may deviate from published specifications as a result of design defects, errors, or operation of products outside of published parameters or design specifications. Customers should include design and operating safeguards to minimize these and other risks. Skyworks assumes no liability for applications assistance, customer product design, or damage to any equipment resulting from the use of Skyworks products outside of stated published specifications or parameters.

Skyworks, the Skyworks symbol, and “Breakthrough Simplicity” are trademarks or registered trademarks of Skyworks Solutions, Inc., in the United States and other countries. Third-party brands and names are for identification purposes only, and are the property of their respective owners. Additional information, including relevant terms and conditions, posted at www.skyworksinc.com, are incorporated by reference.