

Auto Brightness: How to Improve the Android Mobile Operating System in Handheld Consumer Devices



••••• QuickLogic® White Paper

Introduction

Display size on the average mobile phone has grown in the past few years; not coincidentally, this growth in size tracks directly with the explosion of smartphone sales. While the larger displays are ideal for the various functionalities smartphones offer (such as full internet browsing, video viewing, higher megapixel video capture, etc.), they do have the detraction of consuming a large percentage of available battery power. Display power consumption on a handheld device can approach 50% of available battery resources in some cases, regardless of whether the device is a smartphone, tablet, or notebook computer.

A quick and easy way to reduce display power consumption is to turn the liquid crystal display (LCD) backlight or organic light-emitting diode (OLED) display power down. While this can lead to measurable power savings, it also has the side effect of reducing the viewability of the display, especially when in non-ideal viewing conditions (e.g., direct sunlight). To illustrate this point, a smartphone set to 50% backlight is likely to be completely non-viewable when used outdoors during the day. Increasing the backlight to 100% improves the viewability, but reduces the battery life. Whereas reducing the backlight improves battery life, but sacrifices viewability. As smartphone displays become larger, and as tablets enter the everyday life of device users, display viewability and power savings become more important.

As single-charge battery life is a key selling point for original equipment manufacturers (OEMs), system designers must find a balance between ambient lighting, display power consumption, and display viewability. Often, that balance is achieved through non-optimal implementations.

Current Methods of Balancing Display Power and Viewability

To address the issue of balancing display power and viewability, the current solution most commonly used by system designers is ambient light correction (ALC), known to consumers as *Auto Brightness*. By adding an ambient light sensor (ALS) to the system, designers have a way to gauge the viewing conditions experienced by the system user. An ALS measures the ambient light level of the viewing environment and reports that value to the device CPU. Typically, the host system then employs a rudimentary algorithm (such as ALC) that scales display brightness to ambient lighting. A brighter viewing condition causes the system to increase display brightness, and vice versa.

In theory, this implementation should work well. However, industry experts have noted that even in devices considered to be the best, this implementation is done incorrectly. This incorrect implementation results in a viewing experience that is often compromised, with a much-shortened battery life. Displays are often needlessly bright in dark ambient light conditions (wasting power), and not bright enough in brighter ambient light conditions (degrading the user experience).

Dr. Raymond Soneira, of DisplayMate Technologies, details a number of these incorrect implementations in his paper “*BrightnessGate for Android/iPhone Smartphones and HDTVs: Why Existing Brightness Controls and Light Sensors are Effectively Useless*”¹. Dr. Soneira measures the effectiveness of the Auto Brightness algorithms in numerous Android OS smartphones by measuring display brightness versus ambient light levels. His findings support the argument that current Auto Brightness implementations used in popular consumer devices are at best suboptimal. He notes in the case of the Android OS:

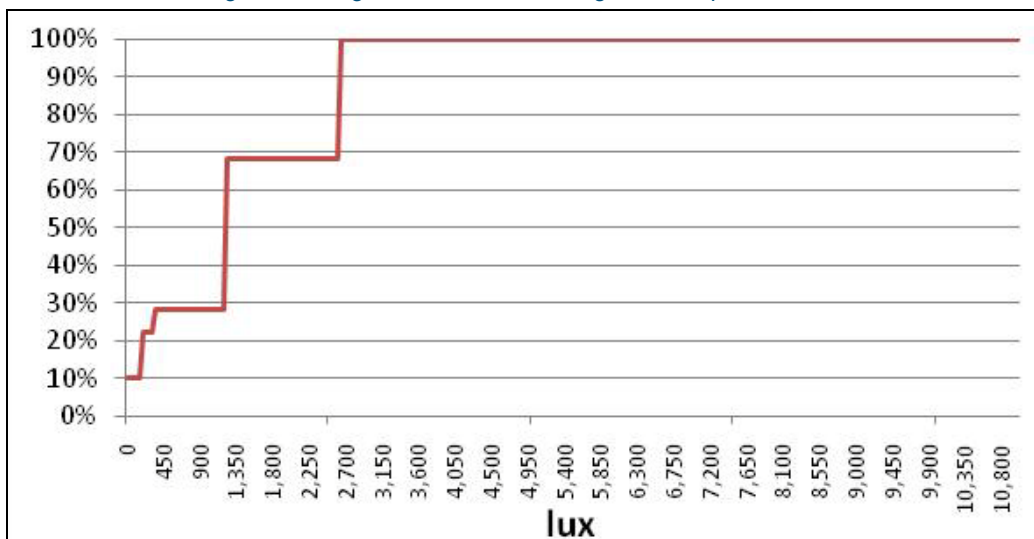
“Unfortunately, those Automatic Brightness settings are incredibly primitive and crude – on the Samsung Galaxy S and HTC Desire that we lab tested Automatic Brightness produces only four fixed screen brightness levels when the ambient lighting changes from pitch black all the way up to direct sunlight, with each manufacturer setting their own breakpoints. For this reason alone, Auto Brightness is effectively useless for Android.”

With those findings, it is fair to say that current implementations of Auto Brightness are not ideal, and in most cases not even appropriate for the users of consumer devices.

QuickLogic tested the Google NexusOne phone as an example of how Android handles backlighting. The testing corroborated the findings of Dr. Soneira, and it was discovered that the NexusOne essentially has five backlight levels (or steps).

The backlight settings are shown in **Figure 1**.

Figure 1: Google NexusOne Auto Brightness Implementation



1. Full paper available at http://www.displaymate.com/AutoBrightness_Controls_2.htm.

Additionally, it was noted that the reported ambient light values for NexusOne differ from actual measured values by a large amount. To prove this, QuickLogic placed a light meter next to the NexusOne, with the light sensitive areas at the same angle to ambient light to preserve an accurate comparison. In **Table 1**, the actual measured ambient light level shown is compared against the level that the NexusOne reports via Android utility.

Table 1: Google NexusOne Reported versus Actual Ambient Light Levels

Measured Ambient Light Level (Lux)	Reported Ambient Light Level (Lux)
3	10
4	160
6	160
6	225
16	225
16	320
38	320
38	640
39	640
41	1,280
98	1,280
98	2,600
128	2,600
130	10,240
3,000	10,240

Table 1 shows that there is a large discrepancy between actual and reported ambient light levels. The cause of this may be the location of the ALS inside of the NexusOne—it is located in the upper right above the display, under dark cover glass/plastic, somewhat obscuring incoming light.

In the NexusOne example, the implementation of Auto Brightness fails in at least two ways:

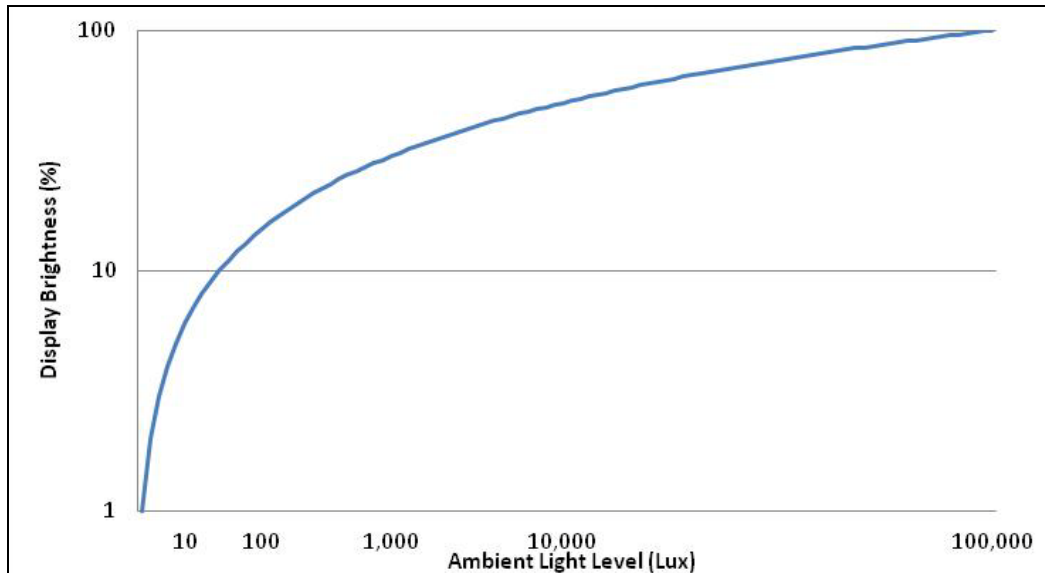
- The ambient light levels that the system reports are always at least 3X the actual light level of the user’s viewing environment.
- Due to incorrect reporting, the display brightness is set artificially high. QuickLogic’s testing showed that at a measured ambient light level of 96 lux (less than an average office environment), the system reported an ambient light level of 2600 lux, and the display was operating at 100% power.

Note that Android itself does not force the OEM to choose any particular ambient light/backlight step levels, nor does it force an incorrect corroboration of actual versus reported ambient lighting. These two activities/decisions are left solely to the system developer.

An Ideal Implementation of Auto Brightness

An ideal implementation of Auto Brightness is one with a linear scale of ambient light versus display brightness. 1 lux corresponding to the lowest brightness level that the display produces, and 100,000 lux (direct sunlight) corresponding to 100% display brightness. This is done on a logarithmic scale to compensate for human vision abilities in the lower light range (see **Figure 2**).

Figure 2: Display Viewability Ideal Implementation



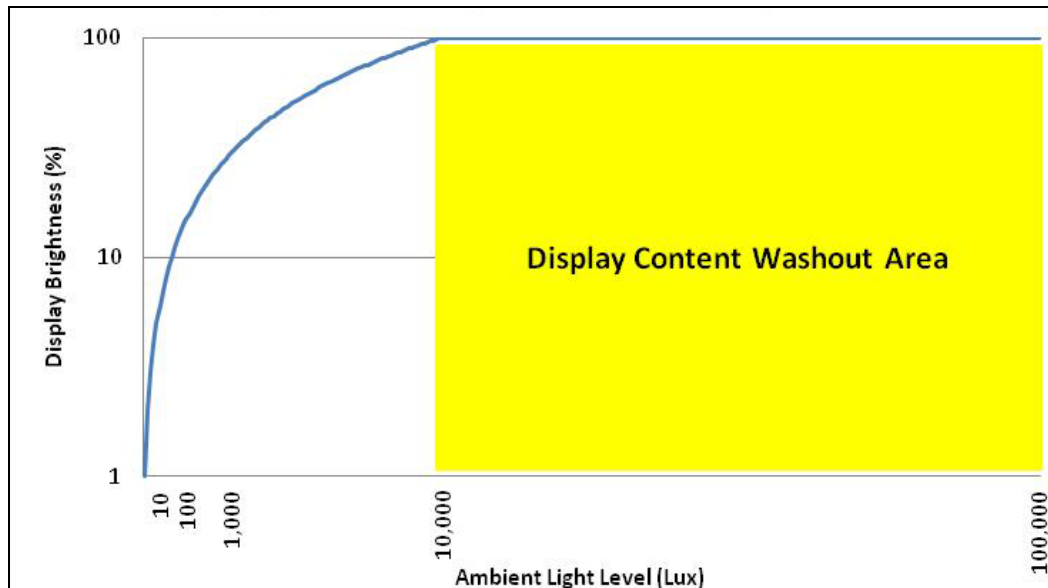
Initially, it seems easy to implement, but there two problems within this implementation:

- Display washout
- Constant ALS polling and backlight changes

The reality on most commercially-available displays is that the maximum brightness level is not enough to prevent display washout in brighter ambient lighting. Washout refers to a situation where the ambient light is sufficiently bright to render the contents of a display unviewable to the human eye. In lower lighting, washout is corrected in some cases by increasing the brightness of the display. As previously stated, this consumes additional power, but restores some viewing abilities to the user. However, every display eventually washes out in brighter ambient lighting, but this level changes with each display model.

To compensate for real-world washout, system designers could modify the display viewability ideal implementation shown in **Figure 2** to one that accounts for the actual display backlight capabilities versus ambient light. **Figure 3** shows a display backlight-realized implementation of Auto Brightness assuming that the maximum brightness of the display is washed out at 10,000 lux, which is much closer to the NexusOne example shown.

Figure 3: Backlight-Realized Auto Brightness Implementation



Additionally, constant ALS polling and the resulting backlight changes could become distracting to the user. Displays would constantly become brighter or darker with small ambient light changes distracting the viewer, and would have the side effect of increased load on the CPU and increased power consumption of display driver and ALS integrated circuits. For these reasons, the implementation of the ideal brightness curves shown in **Figure 2** on page 4 and **Figure 3** are not feasible in consumer devices.

The system designer must consider the best way to improve Auto Brightness without negatively affecting the user experience, and be within the practical limitations of handheld consumer devices.

How Different Implementations Can Improve Auto Brightness within Consumer System-Realizable Limitations

Properly Calibrated and Reported Ambient Light Data

The first step towards a better implementation is to properly calibrate the ALS within the application. In most systems, the ALS is covered by some sort of exterior glass or plastic that is part of the industrial design of the product. That cover reduces the amount of light that is captured by the ALS—knowing specifically how much the light is reduced allows the system designer to more accurately gauge appropriate backlight levels. Typically used ALSs can be calibrated to be as accurate as 1 lux to 8 lux, depending on integration time and ambient light level. An accuracy of 8 lux is more than sufficient for backlight settings.

TAOS Inc, a United States-based leading manufacturer of digital ALS for the handheld device market, recommends calibration to all manufacturers for all applications. Specifically, if the ALS device is behind spectrally distorting material or a bezel, the attenuation effects of the material must be characterized to determine how the sensor data translates to the actual external ambient light. The NexusOne data reported in **Table 1** on page 3 shows that not all OEMs follow this path.

Backlight Steps that Match Use Cases of the Device

The backlight step method of Android is a good start. However, the OEM-specific implementation of those steps almost never corresponds to specific device use cases. Adjusting the backlight steps to correspond to standard viewing environments/use cases also dramatically increase the user experience. QuickLogic calculated that on the NexusOne, the backlight level increases to the next step at the following ambient light levels: 225, 640, 1280, and 2600 lux. Unfortunately, those light levels do not correspond to a normal use case of a consumer device. If more appropriate light levels were used, the backlight settings could be tailored for the specific consumer viewing conditions, and result in a better user experience.

Table 2 shows use cases that are typically applicable to handheld consumer devices.

Table 2: Lux Values for Typical Consumer Viewing Conditions of Handheld Devices

Lux	Viewing Environment(s)
10	Dark room (no overhead light), (such as airplane or twilight)
50	Shaded area of indoor public space (such as airport or office)
150	Home
225	Office
500	Retail store (such grocery store or shopping mall)
1,250	Indirect sunlight
10,000	Direct sunlight (min.)
50,000-100,000	Direct sunlight (max.)

The values in **Table 2** are median values of the particular viewing environment. In a correct implementation, a system designer would not want backlight levels to change at median, but at a point somewhere between the median values. This reduces potential usability issues such as flicker.

Figure 4 plots an implementation of the lux values from **Table 2**, with backlight levels changing between the environments, versus the NexusOne steps shown in **Figure 1** on page 2.

Figure 4: Consumer Use Cases versus NexusOne Presets in Lux

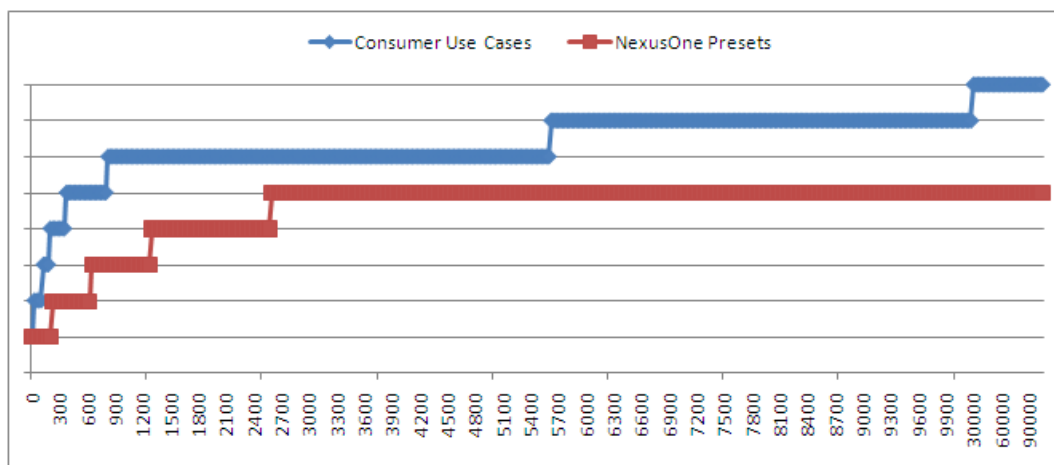


Figure 4 demonstrates that there exists many more consumer viewing use cases than the NexusOne accounts for, especially in sub-1500 lux environments. If the system is calibrated with respect for these use cases, the Auto Brightness function will be much more effective. The development costs for this are extremely small, and this improved approach adds no per-unit cost during mass production. Additionally, since lower ambient lighting requires less display brightness, adding more steps to the Auto Brightness function in the lower lux regions will extend battery life.

Backlight Curves and Brightness Sliders

Regardless of how scientific the approach a system designer uses to construct the Auto Brightness algorithm, users will still demand the option of disabling Auto Brightness, as well as manually adjusting display brightness while Auto Brightness is enabled.

In the case of a user who has disabled Auto Brightness, there is nothing that can be done to improve the viewability by modulating the display brightness.

A typical Android implementation in the case of the user who manually adjusts display brightness while Auto Brightness is enabled is to raise would be to simply raise the brightness and hold until lighting conditions change. A more correct approach to this is to continue to use the stepped approach of Figure 4, but to adjust the entire stepped curve up or down according to user inputs. Rather than have the backlight slider adjust the backlight once, a user-instituted manual backlight adjustment changes backlight at all ambient light levels, using the backlight versus ambient light step curve already established in Figure 4.

This allows for user-customization of the device, since each person has a backlight brightness level they are comfortable with. Being able to dial in personal preference once will lead to a better user experience.

Graphically, this approach is shown in Figure 5 (a 25% user-instituted increase of backlight with Auto Brightness enabled) and Figure 6 (a 40% user-instituted decrease of backlight with Auto Brightness enabled).

Figure 5: A 25% User-Instituted Increase of Backlight with Auto Brightness Enabled

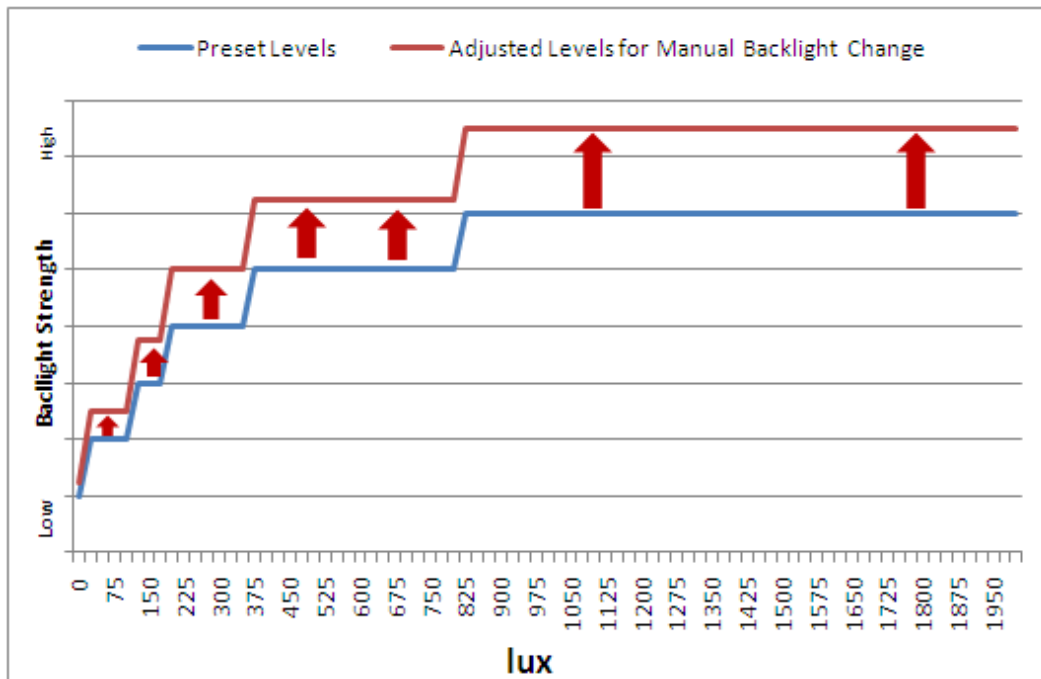
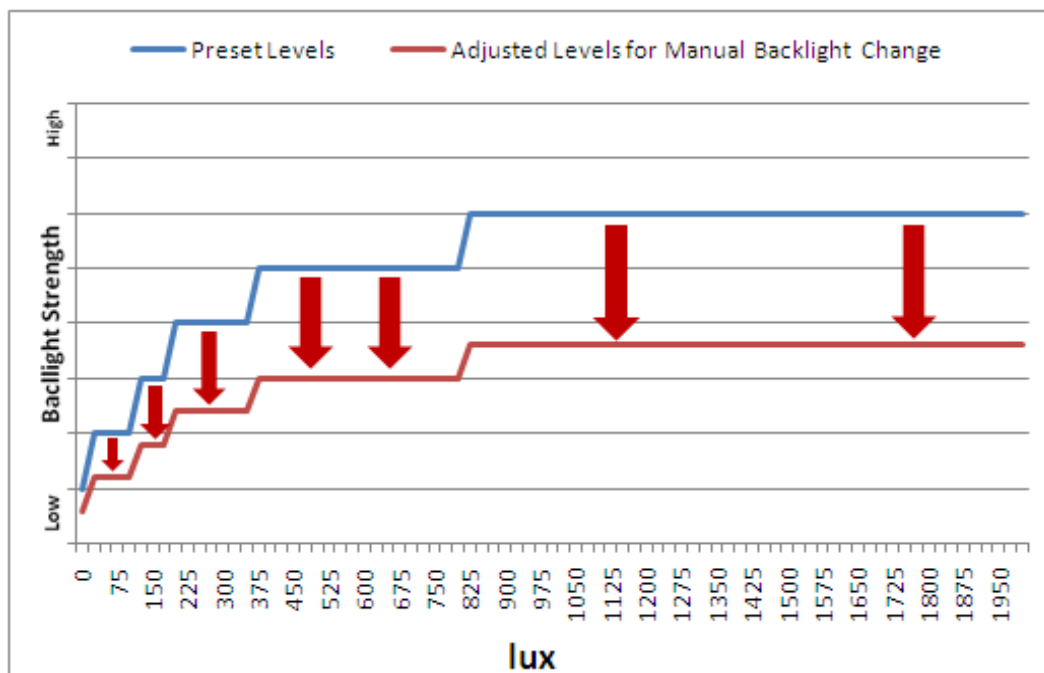


Figure 6: A 40% User-Instigated Decrease of Backlight with Auto Brightness Enabled



Once adjusted manually, Auto Brightness uses the new backlight curve until a further manual adjustment is made.

How a QuickLogic Visual Enhancement Engine (VEE) and Display Power Optimizer (DPO) Implementation Improves Auto Brightness

VEE and DPO Technology

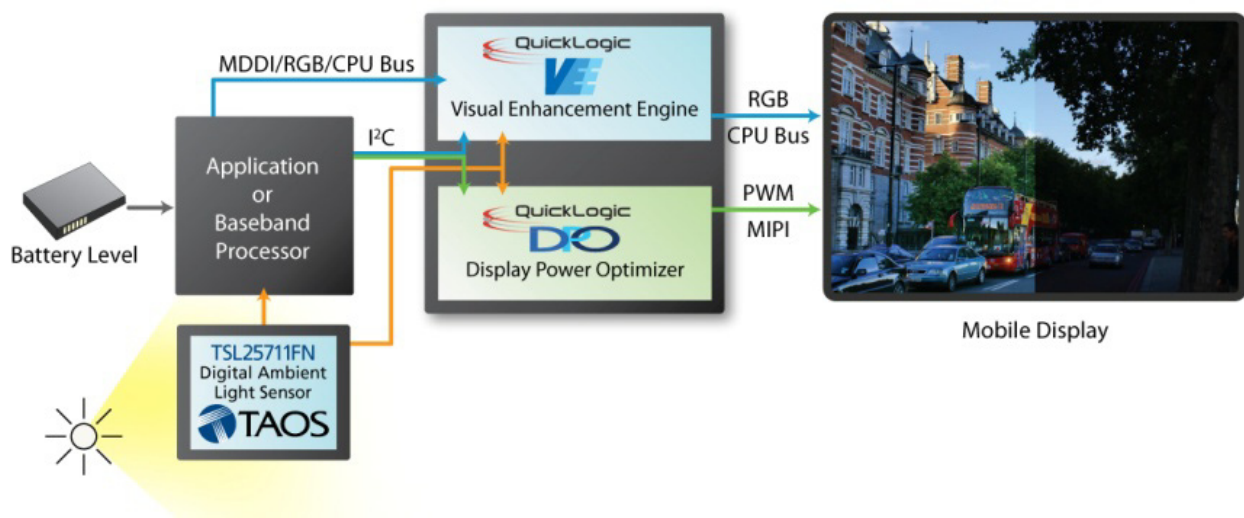
QuickLogic's VEE technology enables a television-quality visual experience on portable devices. This technology delivers the next generation of mobile entertainment experience by adapting display data, in real-time, to improve the user's viewing experience under difficult conditions such as low backlight or in bright ambient light. QuickLogic's proven VEE solution greatly enhances image and video quality for handset users by compressing the dynamic range to match the characteristics of the display, resulting in a substantially better viewing experience.

QuickLogic's VEE technology is based on the iridix® algorithm from Apical Limited. iridix is an implementation of a set of algorithms based on the orthogonal retina-morphic image transform (ORMIT), developed by Apical Limited and protected by multiple patents. It is a sophisticated method of dynamic range compression (DRC), which differs from conventional methods such as gamma correction in that it applies different tonal and color transformations to every pixel in an image. These algorithms implement a model of human perception, which results in a displayed image that retains detail, color and vitality even under difficult viewing conditions. VEE technology specifically addresses the problem of the low contrast ratio of mobile displays to bring a more television-like viewing experience to mobile devices.

While the VEE uses statistical information gathered pixel-by-pixel, frame-by-frame to adjust the value of individual pixels, the DPO uses that same information to adjust the backlight. The ability to provide a unique tone curve for each pixel, as well as have tight control over the display backlight, gives greater flexibility than the global adjustments of alternative implementations. The QuickLogic approach results in greater power savings and the entirely new capability of adapting to a bright environment. DPO seamlessly integrates with VEE, ensuring longer battery life and an excellent visual experience by coupling the pulse-width modulation (PWM) driving the display backlight with the display content processing parameters of the VEE technology.

VEE and DPO are hardware-based technologies, integrated into consumer devices via a Customer Specific Standard Product (CSSP) solution platform from QuickLogic. The CSSP is placed in the display path between the CPU/applications processor and the display as shown in **Figure 7**.

Figure 7: VEE and DPO in a System – Shown with a TAOS Ambient Light Sensor



Setting Up VEE and DPO Modes

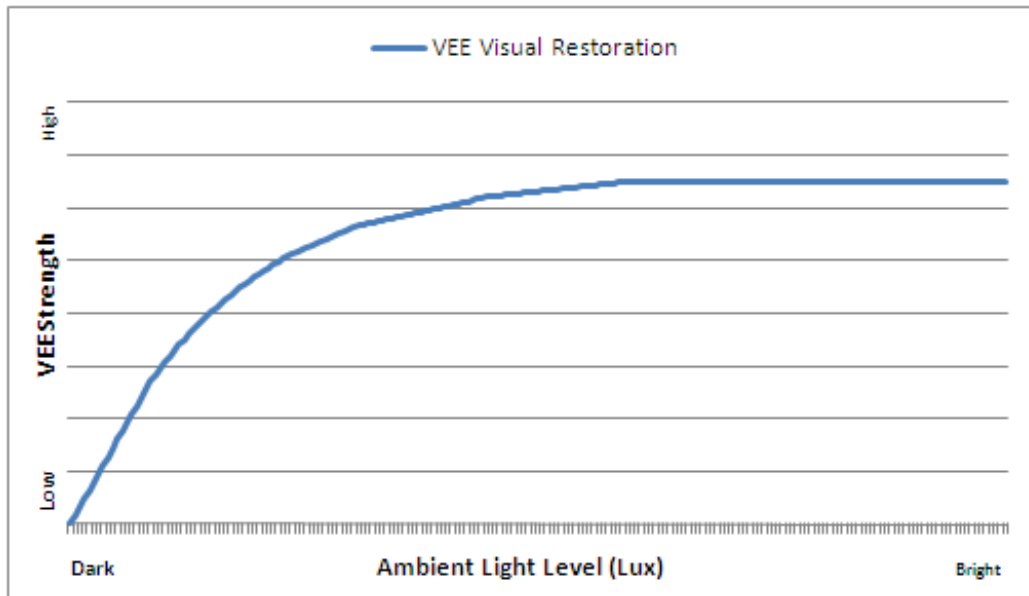
VEE and DPO modes can be set up in multiple ways:

- Visual restoration
- Visual restoration and power savings
- Visual improvement
- Visual improvement and power savings

Visual Restoration

VEE is set up in this mode to restore the viewing quality of the display to a level that is typically seen in a good viewing environment (lower ambient light with a bright display). DPO is not active in this mode, since backlight is never changed. **Figure 8** shows a graphical interpretation of visual improvement versus visual restoration.

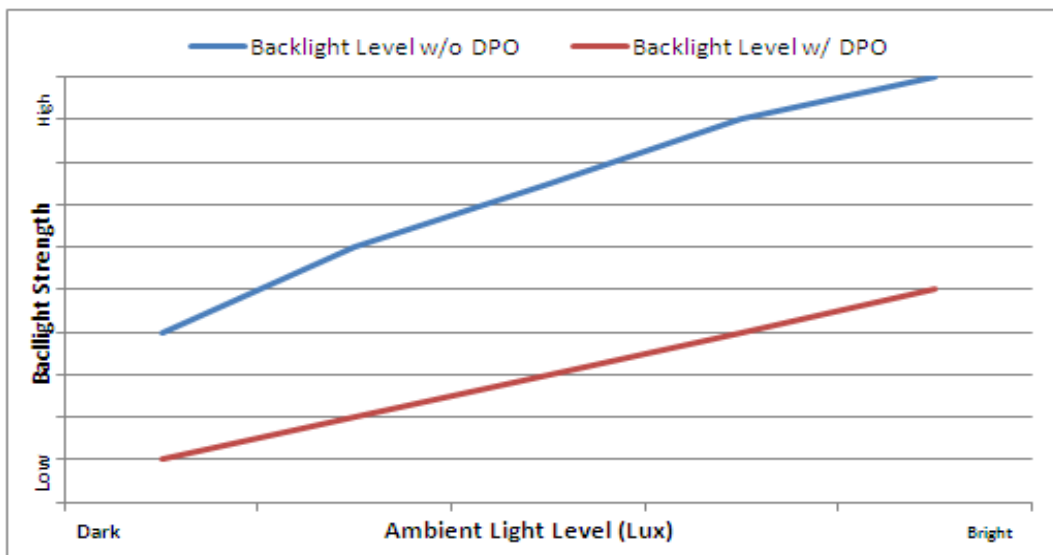
Figure 8: VEE Visual Restoration



Visual Restoration and Power Savings

VEE is set up in this mode to restore the viewing quality of the display to a level that is seen in a good viewing environment (lower ambient light with a bright display). Additionally, DPO works in concert with VEE to actively adjust the backlight, based on current ambient light conditions as measured by an ALS, such as the TAOS TSL2571, to the lowest level possible without negatively affecting the viewing experience. While actual power savings depends on the specific display and system characteristics, typical system battery savings that OEMs have achieved in tablet and smartphone designs average 25%, with some designs showing system savings as much as 36%. **Figure 9** represents the DPO power savings by comparing the backlight level with and without DPO.

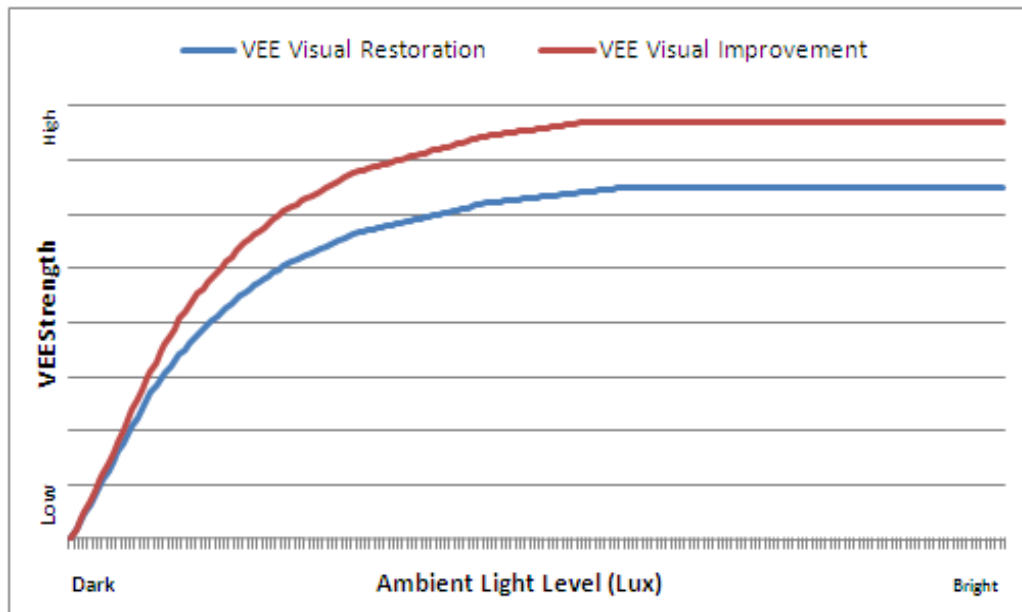
Figure 9: DPO Power Savings



Visual Improvement

VEE is set up in this mode to improve the viewing quality of the display to the best level possible by applying VEE strength as strong as possible. While VEE does not create artifacts in the display content, artifacts or noise present in the source display content will be amplified along with the 'good' content. DPO is not active in this mode, as backlight is never changed. **Figure 10** shows a graphical interpretation of visual improvement versus visual restoration.

Figure 10: VEE Visual Restoration vs. Visual Improvement



Visual Improvement and Power Savings

VEE is set up in this mode to improve the viewing quality of the display to the best level possible by applying VEE strength as strong as possible. While VEE does not create artifacts in the display content, artifacts or noise present in the source display content will be amplified along with the 'good' content. Additionally, DPO works in concert with VEE to actively adjust the backlight, based on the current ambient lighting conditions as measured by the ALS, to the lowest level possible without negatively affecting the viewing experience. Changing VEE strength does not increase device power consumption, thus there is no difference between VEE power consumption when used in restoration or improvement modes.

In actual application, most OEMs choose an implementation of VEE that falls between the restoration and improvement curves. A curve closer to restoration is generally applied to low ambient light levels, where image details (and thus artifacts) are much more noticeable. An improvement curve tends to be used in the higher ambient light levels, where sheer viewability is more important than absolute image quality.

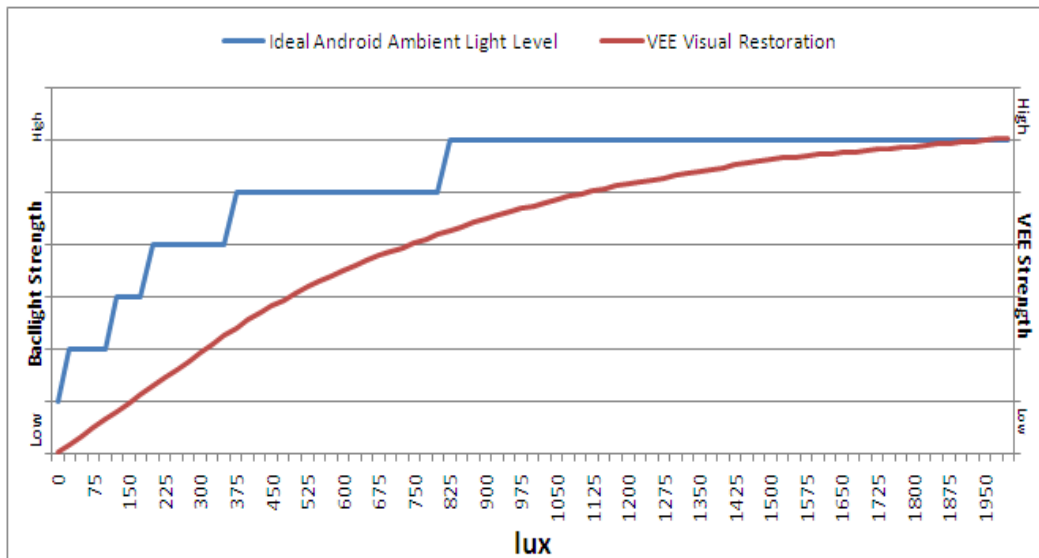
VEE and DPO Applied to the Standard Android Stepped Backlight Implementation

As discussed earlier, the existing Android backlighting implementation can be improved by changing the step locations to match use cases. All of the VEE setup modes can be applied to this improved implementation, as well as the existing implementation.

Visual Restoration

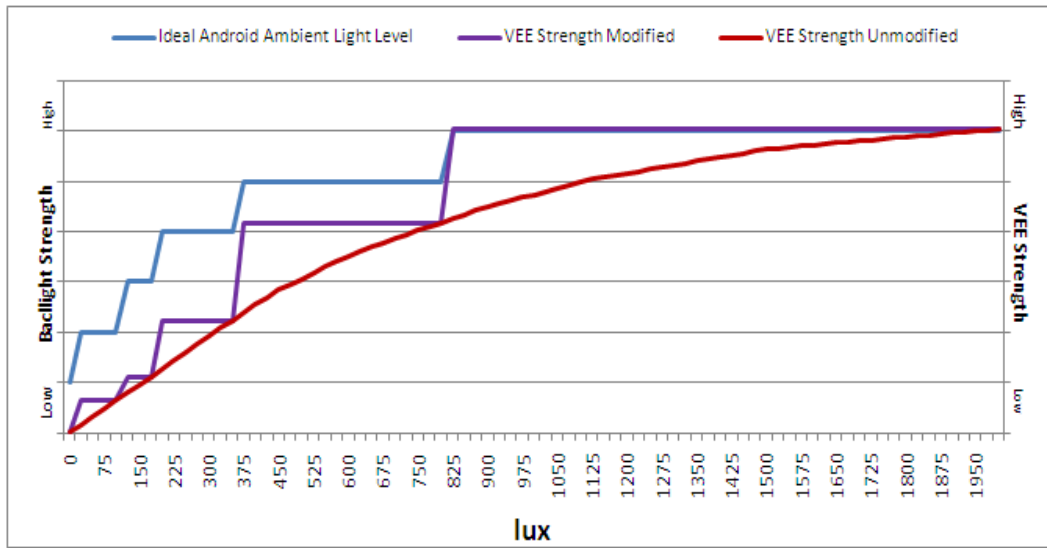
To demonstrate how VEE can best be applied to the Android stepped backlight implementation, the improved step choices from **Figure 4** on page 6 are used. First, a VEE curve is applied to the display, independent of the Android stepped backlight methodology (this is compensated for later). Then the example restoration curve from **Figure 8** on page 10 is applied as shown in **Figure 11** (the 0 to 2000 lux graph is used for ease of viewing).

Figure 11: VEE Strength versus Ideal Android Implementation



To follow the Android stepped methodology, VEE strength remains the same at each step. Using the data from **Figure 11**, the VEE curve is modified by selecting the highest VEE strength seen within a given step and applying it to every point within that step. This change is shown in **Figure 12** as the purple line.

Figure 12: VEE Strength Modified for Ideal Android Implementation



What results from **Figure 11** on page 12 is a VEE strength chart that matches the Android stepped ambient light versus backlight level algorithm, best matched to the typical uses of handheld mobile devices. Throughout each one of these properly matched and calibrated viewing environment, the consumer can expect to have a viewing experience unaffected by ambient light and amplified artifacts.

Power Savings

To demonstrate DPO, the backlight data from **Figure 12** on page 13 is used (blue line). By applying this data with the backlight level savings from **Figure 9** on page 10, **Figure 13** is created.

Figure 13: DPO Modified for Ideal Android Implementation

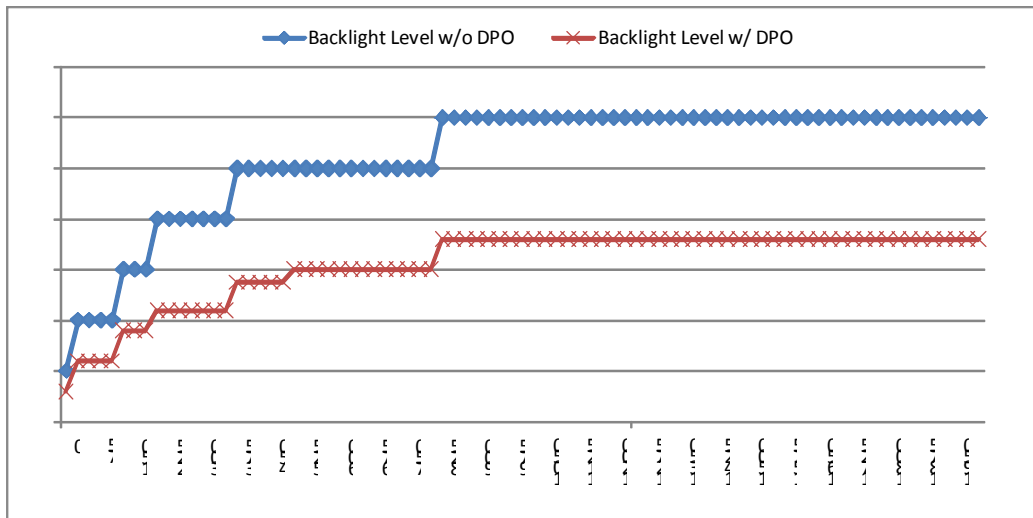
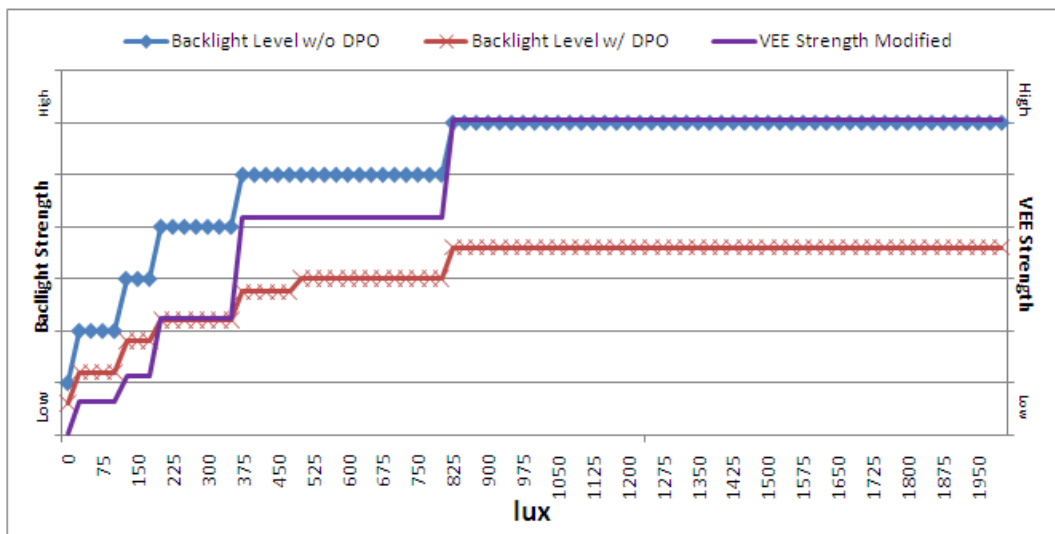


Figure 13 demonstrates the power savings that can be achieved using QuickLogic’s DPO technology. Applying hysteresis to this example eliminates flicker and constant changes when the light level is on the edge of a transition. Additionally, instituting a ‘breathing effect’, a gradual rising and lowering of backlight, eliminates the jarring of a suddenly-changed backlight. While actual power savings depends on the specific display and system characteristics, typical system battery savings that OEMs have achieved in tablet and smartphone designs average 25%, with some designs showing system savings as much as 36%.

Visual Restoration and Power Savings

Figure 14 is the combination of **Figure 12** on page 13 and **Figure 13** on page 13. This illustrates what an ideal VEE and DPO implementation looks like under the Android Auto Brightness implementation correctly purposed for handheld consumer devices.

Figure 14: VEE and DPO Shown for Ideal Android Implementation



Software Implementation in Android

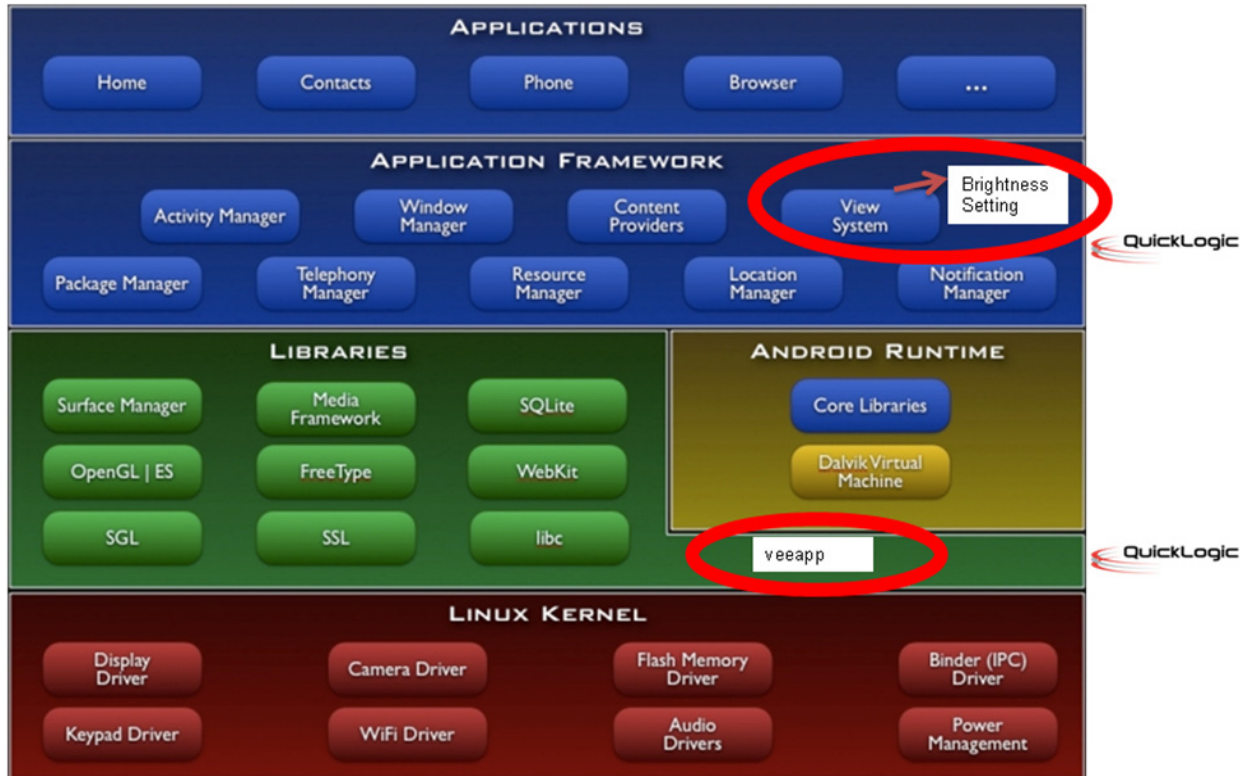
QuickLogic has developed three distinct software pieces to implement these recommendations in the Android operating system:

- veeapp.apk
- settings.apk
- QlAutoAl.apk — Mobile Device Display Optimizer

VEEApp.apk

VEEApp is a Linux application with two modes: command line and daemon. It sits within the library level of Android, above the kernel level (see **Figure 15**). VEEApp communicates with the Android Power Manager to get updated ambient light and display backlight levels and with the ArcticLink II VX solution platform to correctly operate VEE and DPO. VEEApp is programmed to the customer's unique system by QuickLogic, and requires no development work on the part of the system designer.

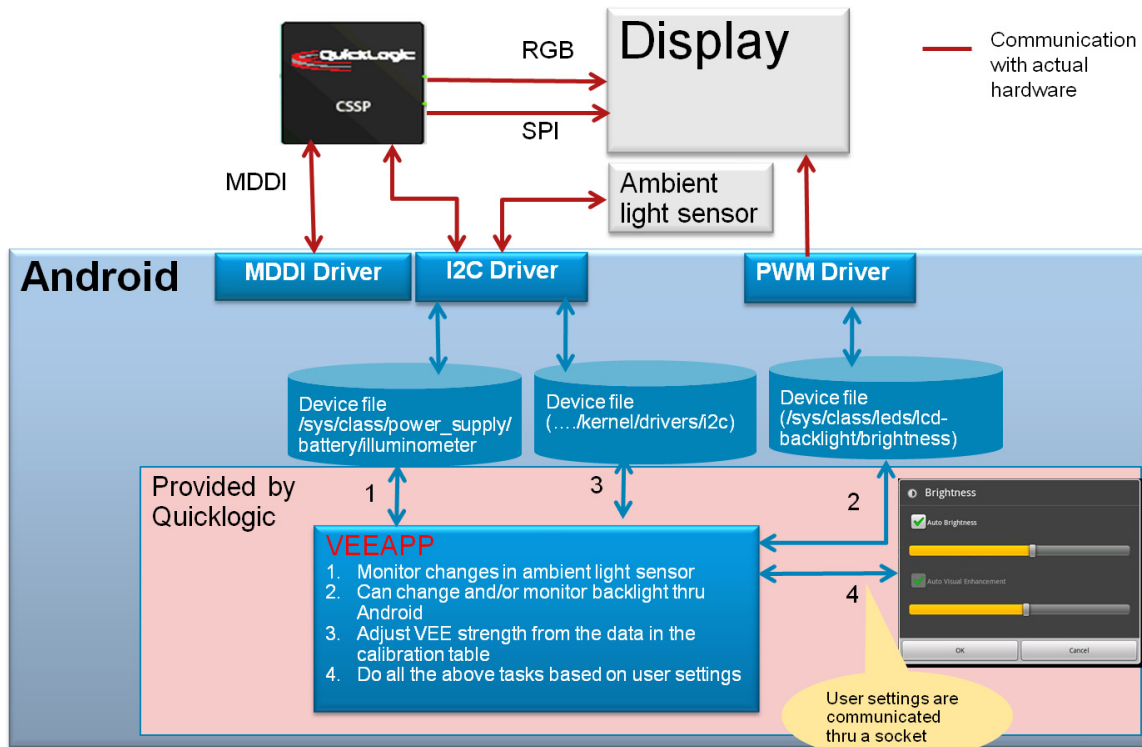
Figure 15: Android Operating System



The VEEApp application:

- Communicates with the ArcticLink II VX solution platform through I²C drivers in the system (see **Figure 16**).
- Communicates with the Android light sensor and backlight control services to:
 - Receive events from Android regarding ambient light and display brightness level changes from Android and adjusts VEE operation.
 - Change the display brightness level through Android.
 - Poll the ambient light level at a defined interval through I²C drivers independent of Android light sensor service.
 - Talk to any Android application through a socket client interface.
- Provides the capability to read/write ArcticLink II VX solution platform registers in command line mode, which is a good hardware debugging tool and useful during the calibration process.

Figure 16: VEEApp Interaction



Settings.apk

The `settings.apk` file provides reference Java code for the brightness settings dialog, which can replace the original Android Backlight Settings menu.

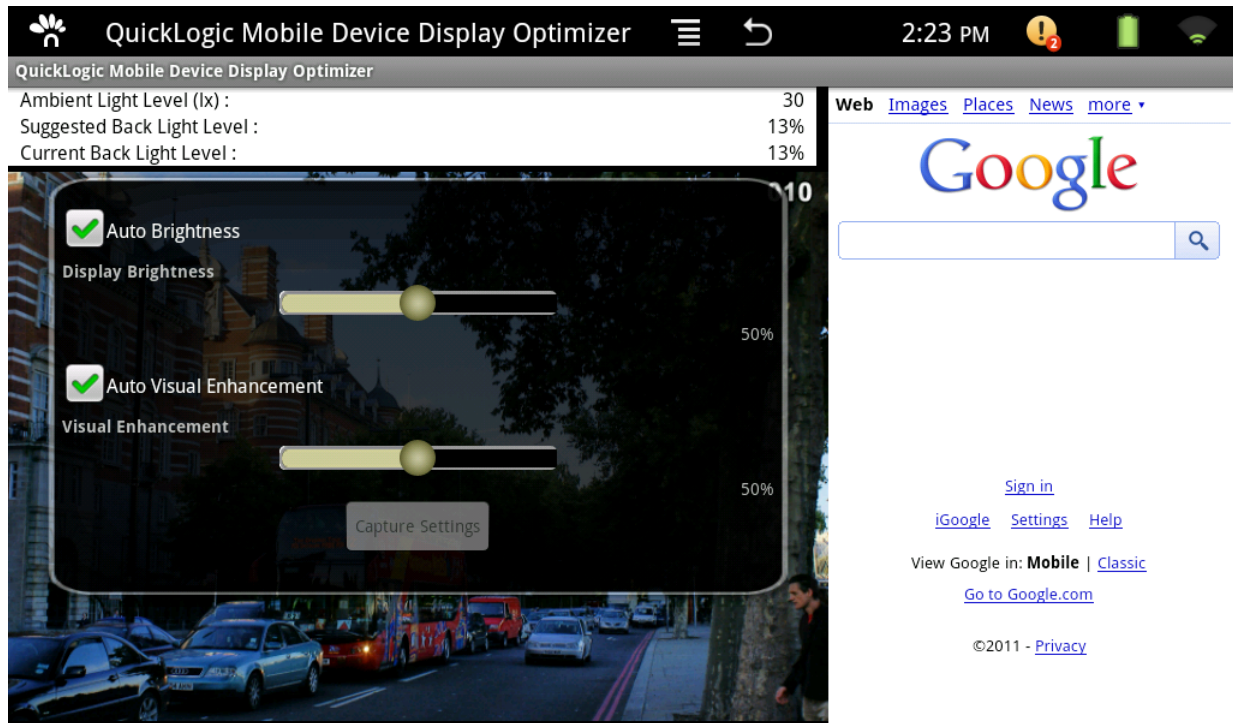
GUI implementation includes:

- Auto Brightness – Automatically adjusts backlight based on the current ambient light.
- Individual user preference setting for brightness – Adjusts brightness even with Auto Brightness enabled.
- Auto Visual Enhancement – Automatically adjusts VEE compensation based on current ambient light and backlight level.
- Individual user preference setting for VEE compensation – Adjusts VEE compensation even with Auto Visual Enhancement enabled.
- Auto Visual Enhancement with manual backlight control (Auto Brightness deselected) – Adjusts VEE compensation based on the current ambient light level and the user’s manual backlight setting.
- Implements a socket client to communicate user selections with VEEApp daemon.

QIAutoAI.apk — Mobile Device Display Optimizer

QuickLogic's MDDO (QIAutoAI.apk) is an Android application designed to replace the standard Auto Brightness window within Android in systems with a QuickLogic CSSP featuring VEE and DPO (see [Figure 17](#)).

Figure 17: MDDO Settings



MDDO allows the user to:

- Select/deselect Auto Brightness
- Manually adjust brightness even with Auto Brightness enabled (using the methods described earlier)
- Select/deselect VEE
- Manually adjust VEE strength

MDDO assumes that the system is using standard Android APIs for communicating with and controlling ambient light information and display backlight. If non-standard methods are used, the software must be modified to work within the system framework (contact QuickLogic).

This application is optional and not required to receive the benefits of VEE and DPO, but is strongly recommended for ideal implementation of the technologies to best improve user experience.

For information on the coding within the program, contact QuickLogic.

Additionally, this application can be customized in appearance to suit a particular OEM or system designer need.

Conclusion

Current Auto Brightness algorithms on the Android OS are not ideally implemented for the usage models of the devices they inhabit. Further, the actual information the algorithms use to process may be suspect as well.

With a more scientific and dedicated approach to Auto Brightness, system designers can create Auto Brightness algorithms that are much more functional and usage-friendly.

Additionally, with the addition of QuickLogic's VEE and DPO technology into a system, device users will experience superior viewability and longer battery life.

Contact Information

Phone: (408) 990-4000 (US)
(647) 367-1014 (Canada)
+(44) 1932-21-3160 (Europe)
+(886) 2-6603-9848 (Taiwan)
+(86) 21-5116-0532 (China)

E-mail: info@quicklogic.com

Sales: America-sales@quicklogic.com

Europe-sales@quicklogic.com

Asia-sales@quicklogic.com

Japan-sales@quicklogic.com

Support: www.quicklogic.com/support

Internet: www.quicklogic.com

Revision History

Revision	Date	Originator and Comments
A	July 2011	Paul Karazuba and Kathleen Bylsma

Notice of Disclaimer

QuickLogic is providing this design, product or intellectual property "as is." By providing the design, product or intellectual property as one possible implementation of your desired system-level feature, application, or standard, QuickLogic makes no representation that this implementation is free from any claims of infringement and any implied warranties of merchantability or fitness for a particular purpose. You are responsible for obtaining any rights you may require for your system implementation. QuickLogic shall not be liable for any damages arising out of or in connection with the use of the design, product or intellectual property including liability for lost profit, business interruption, or any other damages whatsoever. QuickLogic products are not designed for use in life-support equipment or applications that would cause a life-threatening situation if any such products failed. Do not use QuickLogic products in these types of equipment or applications.

QuickLogic does not assume any liability for errors which may appear in this document. However, QuickLogic attempts to notify customers of such errors. QuickLogic retains the right to make changes to either the documentation, specification, or product without notice. Verify with QuickLogic that you have the latest specifications before finalizing a product design.

Copyright and Trademark Information

Copyright © 2011 QuickLogic Corporation. All Rights Reserved.

The information contained in this document is protected by copyright. All rights are reserved by QuickLogic Corporation. QuickLogic Corporation reserves the right to modify this document without any obligation to notify any person or entity of such revision. Copying, duplicating, selling, or otherwise distributing any part of this product without the prior written consent of an authorized representative of QuickLogic is prohibited.

QuickLogic is a registered trademark, and the QuickLogic logo is a trademark of QuickLogic. Other trademarks are the property of their respective companies.