



1/4-Inch SOC VGA CMOS Active-Pixel Digital Image Sensor Developer Guide

MT9V131

For the latest data sheet, refer to Aptina's web site: www.apgina.com

Introduction

This developer guide provides the information needed for applications where the Aptina's MT9V131 is an integral part of the design. It is intended to supplement the MT9V131 data sheet and is provided exclusively for licensed developers.

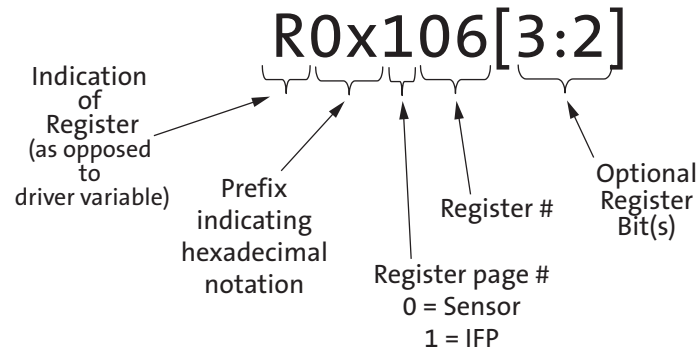
Applicable Part

This document is applicable to the MT9V131, an VGA-format single-chip camera with a 1/4-inch CMOS active-pixel digital image sensor. This device combines the MT9V011 image sensor core with fourth-generation digital image flow processor technology from Aptina Imaging. It captures high-quality color images at VGA resolution.

Register Operations

This developer guide refers to various registers that the user reads from or writes to for altering the MT9V131 operation. Hardware registers appear as follows and may be read from or written to by sending the address and data information over the two-wire serial interface.

Figure 1: Register Legend



The MT9V131 was designed to facilitate customizations to optimize image quality processing. Multiple parameters are allowed to be adjusted at various stages of the image processing pipeline to tune the quality of the output image.

The MT9V131 contains two register pages: sensor and IFP. The register page must be set prior to writing to a register in the page. Example: to write register R0x106 (register 6 in page 1):

- Write the value of “1” to the page map register (0xF0)
- Write the desired value to register R0x06

The sensor maintains the page number once set. The page map register is located at address 0xF0 for all register pages.

Table of Contents

Introduction	1
Applicable Part	1
Register Operations	2
Table of Contents	3
List of Figures	6
List of Tables	7
Features	8
Functional Overview	9
Internal Architecture	10
Two-Wire Serial Interface	11
Serial Bus Description	11
Protocol	11
Two-wire Serial Interface Sample Write and Read Sequences (with Saddr = 1)	12
16-Bit Write Sequence	12
16-Bit Read Sequence	12
Eight-Bit Write Sequence	13
Eight-Bit Read Sequence	13
Maximum SCLK Frequency	14
Address Pages	14
Page Location	14
DeviceID	14
Output Format Control	15
Raw Bayer Output	15
RGB Output	15
Test Pattern Generation	16
Clocks and Blanking	18
Tuning Frame Rates	19
Border Pixels	20
Blanking Calculations	20
Power Saving Modes	21
Entering Hard Standby	21
Leaving Hard Standby	21
Entering Soft Standby	22
Leaving Soft Standby	22
Avoiding Floating Inputs	22
Timing Summary	23
Power Sequencing	25
Power Up Sequence:	25
Power Down Sequence:	25
Reset	26
Resize and Zoom	27
Zoom	28
Pan	28
Advantages of Using the Zoom Window Over the Sensor Window	28
Sharpness	29
Aperture Correction	29
On-the-fly Defect Correction	29
Flicker Avoidance	29
Auto Exposure (AE)	30
Controlling AE Speed and Sensitivity	30



AE Window and Backlight Compensation	31
Exposure Step Size	31
AE Zones	32
Target Luminance	35
Brightness Control	36
Scheme 1: Pre AE, Pre Gamma	36
Scheme 2: Post AE, Post AWB, Modify Gamma	36
Scheme 3: Post Gamma	36
Effects of the Various Brightness Schemes	36
Auto White Balance (AWB)	37
AWB Overview	37
Color Correction Matrices	38
White Balance Settings	39
AWB Speed and Stability	41
AWB Tuning	42
Sensor Setup	43
Procedure to Adjust Base Sensor Core Gain Ratio	44
Using the Color-Chart Overlay - AWB Error (RGB) page	45
Using Analysis Graph - RGB	46
Final Results	47
Color Correction Matrix Tuning	47
Initialization	47
Procedure	47
Using the Color-Chart Overlay - Color Error (RGB) and the Table pages	49
Using Analysis Graph - RGB	50
Final Results	51
Fast CCM	51
White Balance Loading Procedure	52
Lens Shading Correction	53
Procedure for Calibrating LC	54
Gamma Correction	58
Selecting a Gamma Factor	58
Gamma Tables—Beyond Gamma Correction	59
Calculating Gamma Table	59
Gamma Tables and Luma Clip Range and Offset	60
Appendix A: Imaging Developer Guide FAQs	61
Optimal Settings for the MT9V131	61
Interpreting the Design ID	61
Enable Raw Bypass Mode	61
Default Mode for the MT9V131	61
Speed Up Auto Exposure	61
Speed up AWB	62
When to Speed up AE and AWB	62
How to Program the MT9V131 to Work off an Oscillator with a Given Frequency	62
When to Calibrate CCM and/or Lens Correction	62
How to Calibrate CCM	62
How to Calibrate Lens Correction	62
How to Fire Flash	63
How to Enable Built-in Test Patterns	63
How to Zoom and Pan Smoothly	63
How Fast does Two-Wire Serial Interface Run?	63
How to Calculate Gamma Table	64
Configuring a JPEG-friendly Output	64



How to Suppress Aliasing in Decimation65
How to Mirror the Image65
How to put Camera in Standby65
How to Load Defaults in DevWare65
How to Configure the MT9V131 for Video Capture at Constant 15 fps Rate65
How does Noise vs. Frame Rate Trade-off Work66
How to Reduce Banding66
How Soft Standby Compares to Hard Standby66
Where to get Software to Calculate the MT9V131 Timing66
Setting Exact Frame Rate66
Checking White and Black Levels Experimentally66
Gated Output Pixel Clock Mode67
Formulas used for Internal YCbCr Calculation67
Creating Special Effects67
How to Suppress Row Noise in Dark67
How to Choose IR-cut Filter68
How to Run the SOC Slower than 5 fps68
LPF in the MT9V13171
Color Suppression in Saturated Areas71
Color Suppression in Low Light Conditions71
Size of Interpolation, Defect Correction and Sharpening Kernels71
Defect Correction of the MT9V13171
How to Switch Master Clock On-the-Fly71
How to Reduce EMI72
How to Check AWB72
How to Operate White Balance Manually73
How to Achieve Red-Eye Reduction73
How to Increase Frame Start Blanking73
Appendix B: Glossary of Terms74
Contacting Aptina76
Revision History77

List of Figures

Figure 1:	Register Legend	2
Figure 2:	MT9V131 Functional Block Diagram	9
Figure 3:	Internal Register Grouping	10
Figure 4:	Timing Diagram Showing a Write to R0x09 with Value 0x0284	12
Figure 5:	Timing Diagram Showing a Read from R0x09; Returned Value 0x0284	12
Figure 6:	Timing Diagram Showing a Byte-wise Write to R0x09 with Value 0x0284	13
Figure 7:	Timing Diagram Showing a Byte-wise Read from R0x09; Returned Value 0x0284	13
Figure 8:	Sensor Core Clock	18
Figure 9:	Hard Standby Operation Waveform	23
Figure 10:	Soft Standby Operation Waveform	24
Figure 11:	Power-Up and Power-Down Sequences	25
Figure 12:	AWB Sensor Controls	37
Figure 13:	AWB Flow Processing	38
Figure 14:	Color-Chart Overlay: Color Error (Lab) Page	42
Figure 15:	Color-Chart Overlay: Reference Squares Aligned with Macbeth Chart	43
Figure 16:	Color-Chart Overlay: AWB Error (RGB)	45
Figure 17:	Analysis Graph - RGB Window Showing R,G,B Components of the Macbeth Chart	46
Figure 18:	Matrix Buddy Page	48
Figure 19:	Color-Chart Overlay: Table Page	49
Figure 20:	Analysis Graph of R, G, B Macbeth Chart Squares	50



List of Tables

Table 1:	MT9V131 Two-Wire Interface ID Address Switching	11
Table 2:	Page Access	14
Table 3:	Sensor-Timing-Dependent Registers.....	19
Table 4:	Blanking Parameter Calculations	20
Table 5:	Window Setting Restrictions	27
Table 6:	Aperture Correction Settings	29
Table 7:	AE Zones	32
Table 8:	Scheme 3 Brightness Settings	36
Table 9:	Suggested Matrix Positions for Varying Conditions	39
Table 10:	Base and Delta Gain Ratios.....	47
Table 11:	Output Register Values	60
Table 12:	Glossary of Terms	74

Features

The MT9V131 performs sophisticated processing functions including:

- Color recovery
- Color correction
- Sharpening
- Programmable gamma correction
- Auto black reference clamping
- Auto exposure (AE)
- Automatic 50Hz/60Hz flicker avoidance
- Lens shading correction (LC)
- Auto white balance (AWB)
- On-the-fly defect identification and correction
- Image flow processor (IFP)

Additional features include:

- Day/night mode configurations
- Special camera effects such as sepia tone and solarization
- Interpolation to arbitrary image size with continuous filtered zoom and pan
- Supports both xenon and LED-type flash light sources in several snapshot modes.
- The MT9V131 can be programmed to output progressive-scan images at up to 30 frames per second (fps).

The image data can be output in any one of six 8-bit formats:

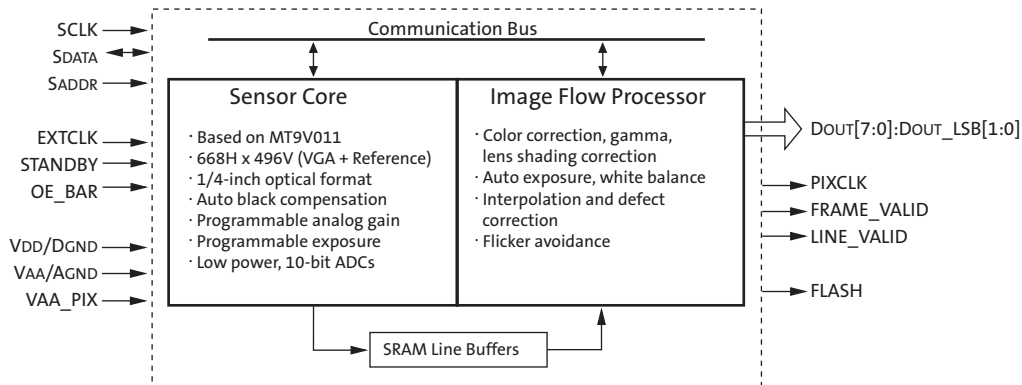
- Progressive ITU-R BT.656 (formerly CCIR656, progressive scan only) YCbCr
- 565RGB
- 555RGB
- 444RGB
- Raw Bayer
- Processed Bayer

Functional Overview

The MT9V131 is a fully automatic, single-chip camera, requiring only a power supply, lens, and clock source for basic operation.

Output video is streamed through a parallel 8-bit DOUT port, shown in the Figure 2.

Figure 2: MT9V131 Functional Block Diagram



The output pixel clock is used to latch data, while FRAME_VALID (FV) and LINE_VALID (LV) signals indicate the active video. Internal registers are configured using a two-wire serial interface.

The device can be put in a low-power sleep mode by asserting the standby pin and shutting down the clock.

Output pins enter High-Z mode when STANDBY is tied HIGH (STANDBY = 1) or the output disable register bit is set.

Putting output pins in High-Z mode and entry into standby mode can both be achieved through two-wire serial interface register writes.

The MT9V131 can accept an input clock of up to 27 MHz, delivering 30 fps. With power-on defaults, the camera is configured to deliver 15 fps at 12 MHz and automatically slows down the frame rate in low-light conditions to achieve longer exposures and better image quality.

Internal Architecture

Internally, the MT9V131 consists of a sensor core and an image flow processor (IFP). The sensor core functions to capture raw Bayer-encoded images that are input into the IFP as shown in Figure 2 on page 9. The IFP processes the incoming stream to create interpolated, color-corrected output and controls the sensor core to maintain the desirable exposure and color balance.

The sensor core and IFP registers are grouped into two separate address spaces, as shown in Figure 3. The internal registers can be accessed via the two-wire serial interface. Selecting the desired address space can be accomplished by programming register R1 which remains present in both register sets.

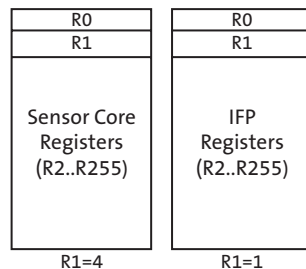
The MT9V131 supports a range of color formats derived from four primary color representations:

- YCbCr
- RGB
- Raw Bayer (unprocessed, directly from the sensor)
- Processed Bayer (Bayer format data regenerated from processed RGB)

The device also supports a variety of output signaling/timing options:

- Standard FV/LV video interface with gated pixel clocks
- Standard video interface with uniform clocking
- ITU-R BT.656 marker-embedded video interface with either gated or uniform pixel clocking

Figure 3: Internal Register Grouping



Notes: 1. Program R0x01 to select the desired space (0b0100 = sensor core registers, 0b0001 = IFP/SOC registers).

Two-Wire Serial Interface

Serial Bus Description

Registers are written to and read from the MT9V131 through the two-wire serial interface bus. The sensor is a serial interface slave controlled by the serial clock (SCLK), which is driven by the serial interface master. Data is transferred in and out of the MT9V131 through the serial data (SDATA) line. The SDATA line is typically pulled up to 2.8V offchip by a 1.5KΩ resistor (optimum value should be selected based on transmission speed and loading condition). Either the slave or the master device can pull the SDATA line down; the serial interface protocol determines which device is allowed to pull the SDATA line down at any given time. The registers are 16 bits wide and can be accessed through 16-bit or 8-bit two-wire serial bus sequences.

Protocol

The two-wire serial interface defines several different transmission codes, as follows:

- A start bit
- The slave device eight-bit address
SADDR is used to select between two different addresses in case of conflict with another device. If SADDR is LOW, the slave address is 0x90; if SADDR is HIGH, the slave address is 0xB8.
- An acknowledge or a no acknowledge bit
- An 8-bit message
- A stop bit

The MT9V131 can respond to one of two addresses on the two-wire serial interface as a function of the state of the serial address (SADDR) input pin. Refer to Table 1 for details.

When accessing internal registers through the two-wire serial interface, select the desired address space by programming the R240 (R0x0F0) register.

Table 1: MT9V131 Two-Wire Interface ID Address Switching

SADDR	Two-Wire Interface Address ID
0	0x90
1	0xBA

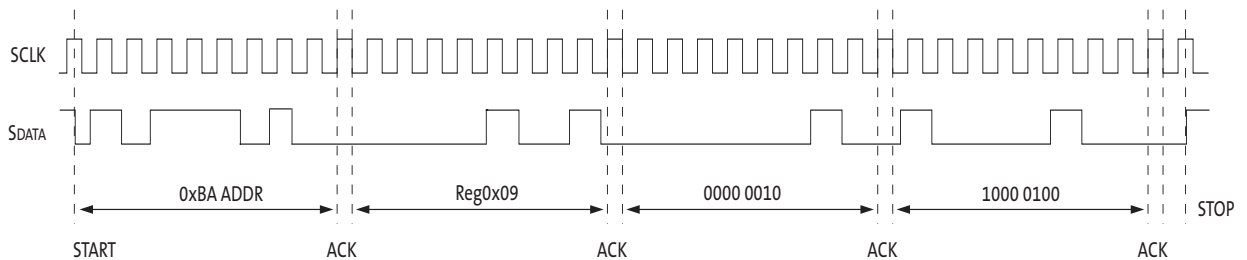
Refer to the data sheet for generic two-wire serial interface examples. Two specific examples are shown in Figure 4 and Figure 5 on page 12.

Two-wire Serial Interface Sample Write and Read Sequences (with SADDR = 1)

16-Bit Write Sequence

A typical write sequence for writing 16 bits to a register is shown in Figure 4. A start bit given by the master, followed by the write address, starts the sequence. The image sensor will then give an acknowledge bit and expects the register address to come first, followed by the 16-bit data. After each 8-bits, the image sensor will give an acknowledge bit. All 16 bits must be written before the register will be updated. After 16 bits are transferred, the register address is automatically incremented, so that the next 16 bits are written to the next register. The master stops writing by sending a start or stop bit.

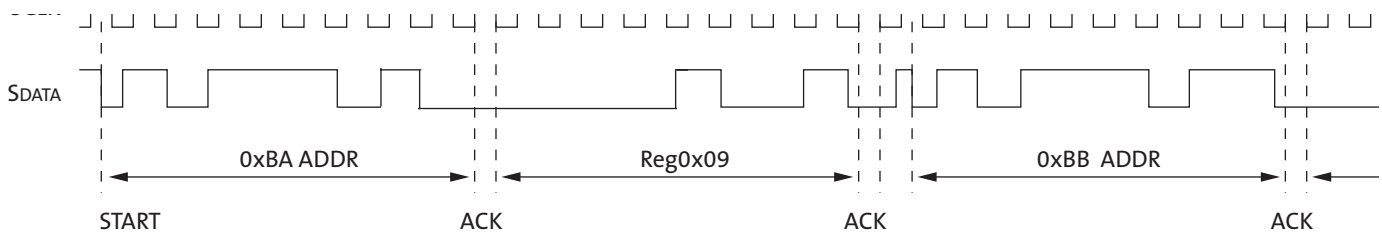
Figure 4: Timing Diagram Showing a Write to R0x09 with Value 0x0284



16-Bit Read Sequence

A typical read sequence is shown in Figure 5. First the master has to write the register address, as in a write sequence. Then a start bit and the read address specifies that a read is about to happen from the register. The master then clocks out the register data 8 bits at a time. The master sends an acknowledge bit after each 8-bit transfer. The register address is auto-incremented after every 16 bits is transferred. The data transfer is stopped when the master sends a no-acknowledge bit.

Figure 5: Timing Diagram Showing a Read from R0x09; Returned Value 0x0284

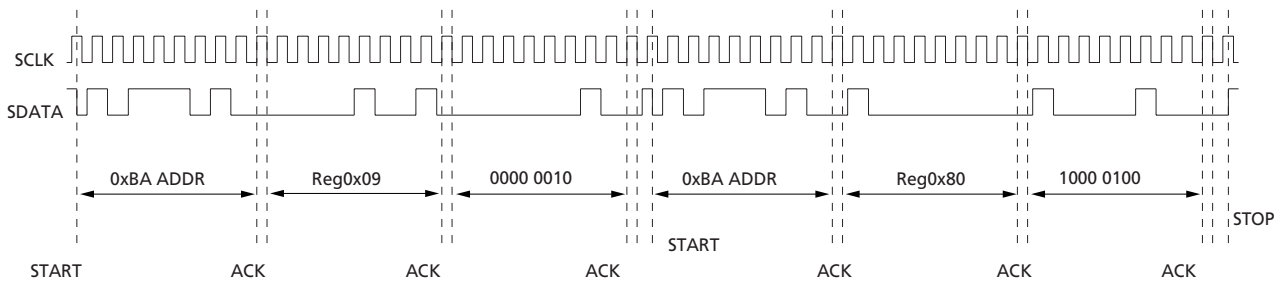


Eight-Bit Write Sequence

All registers in the camera are treated and accessed as 16-bit, even when some registers do not have all 16-bits used. However, certain hosts only support 8-bit serial communication access. The camera provides a special accommodation for these hosts.

To be able to write one byte at a time to the register a special register address is added. The 8-bit write is done by first writing the upper 8 bits to the desired register and then writing the lower 8 bits to the special register address (R0x17F). The register is not updated until all 16 bits have been written. It is not possible to just update half of a register. In Figure 6, a typical sequence for 8-bit writing is shown. The second byte is written to the special register (R0x17F).

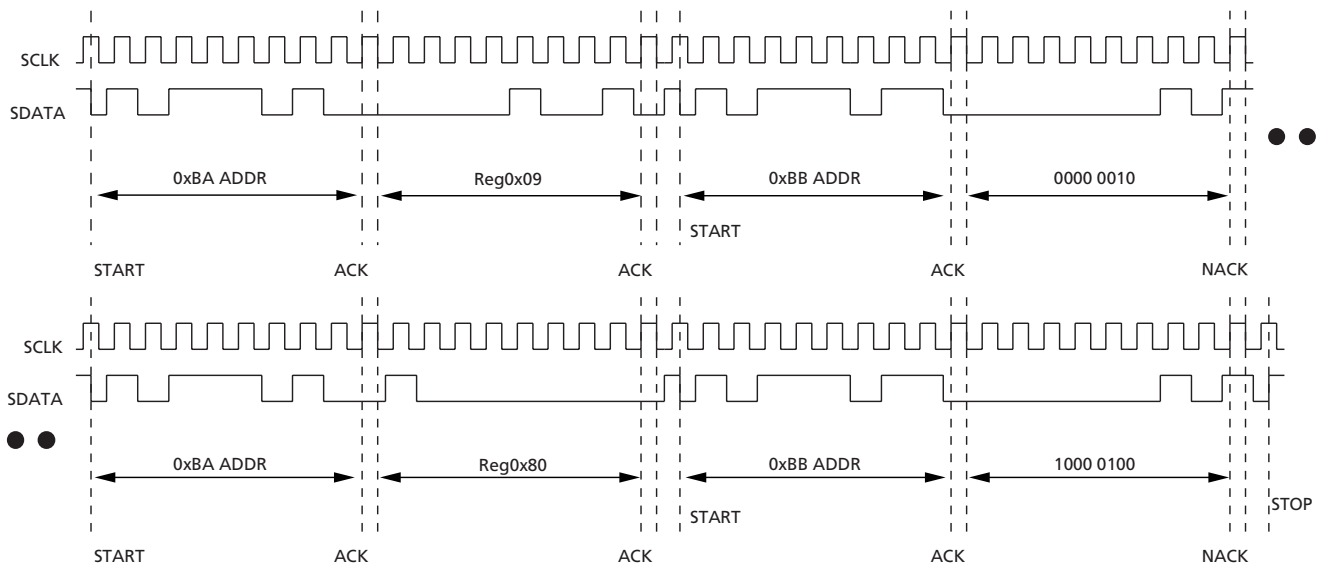
Figure 6: Timing Diagram Showing a Byte-wise Write to R0x09 with Value 0x0284



Eight-Bit Read Sequence

To read one byte at a time the same special register address is used for the lower byte. The upper 8 bits are read from the desired register. By following this with a read from the special register (R0x7F) the lower 8 bits are accessed, as shown in Figure 7. The master sets the no-acknowledge bits.

Figure 7: Timing Diagram Showing a Byte-wise Read from R0x09; Returned Value 0x0284





Maximum SCLK Frequency

The silicon supports $\text{MAX SCLK} = \text{EXTCLK} / 16$.

$\text{PIXCLK} = \text{EXTCLK}$ (or primary input clock)/2.

For example, with the MT9V131 and the recommended 27 MHz MAX clock frequency, and the PCLK = 13.5 MHz, the MAX SCLK rate (digital theoretical) = 1.68 MHz.

Address Pages

Three pages in the address space are accessed through the two-wire serial interface as shown in Table 2.

Table 2: Page Access

Page	Function
0 (default on reset)	Sensor core
1	IFP (Lens correction, color correction, gamma correction, AE, AWB, interpolation and defect correction, flicker avoidance)

Page Location

A single page register defines the currently open page for all accesses; this register is located at address 0xF0 in all three pages.

DeviceID

Device ID can be read from two addresses in the sensor core page: R0x000 and R0x0FE.



Output Format Control

In the MT9V131, the default output format is the 4:2:2 YCbCr.

Raw Bayer Output

To bypass the IFP and get 10-bit raw Bayer output from the sensor (LSB0 and LSB1 signals are available in the CLCC package pins 9 and 10 respectively):

1. Set R0x108[10] = 1.
2. To bypass the IFP and get 8+2 raw Bayer output (in two bytes) from the sensor
 - 2a. Set R0x108[7] = 1.

After setting the sensor into bypass IFP mode, turn off the auto functions:

1. Set R0x106[14] = 0 //disable AE
2. Set R0x106[1] = 0 // disable AWB
3. Set R0x108[11] = 0 // disable auto-flicker control

RGB Output

To enable RGB output:

1. Set R0x108[12] = 1. // This sets the output to RGB format.
2. R0x108[7:6] specify the output format:
 - 00: 16-bit 565RGB
 - 01: 15-bit 555RGB
 - 10: 12-bit 444xRGB
 - 11: 12-bit x444RGB

Test Pattern Generation

The MT9V131 can output various test patterns including a built-in color bar test pattern and a ramp test pattern.

Color Bar Test Pattern

The test pattern generator control register R0x148 can output various shades of grey and a color bar test pattern.

R0x148[2:0] = 000 // Normal operation
 R0x148[2:0] = 001 // Vertical monochrome gradient
 R0x148[2:0] = 111 // Colorbars test pattern

All other values: Flat fields of different luminances

To view the color bar test pattern without the effects of AWB and the color correction matrices, use the following procedure:

1. Set R0x148 = 0x0087 to turn on the color bars, (R0x148 = 0x0081 to turn on vertical monochrome gradient) and set WB digital gains to 1.0.
2. Set R0x106[1] = 0 to turn off AWB.
3. Set R0x106[4] = 1 to bypass the color correction matrix.
4. Set R0x108[11] = 0 to turn off auto flicker detection.
5. Set R0x108[8] = 0 to turn off lens-shading correction.
6. Set R0x106[14] = 0 to turn off AE.

Ramp Test Pattern

The output format control register can output ramp test patterns.

R0x13A[4:3] = 00 // Off
 R0x13A[4:3] = 01 // By column
 R0x13A[4:3] = 10 // By row
 R0x13A[4:3] = 11 // By frame

To test LCDs with low color depth, output RGB or YCbCr values can be shifted up 3 bits by setting R0x13A[5]=1.

The output for the test ramp is generated according to the following formula:

$$\text{If } R0x108[12] = 0, \quad (EQ 1)$$

$$Y = Cr = Cb = \text{pixel_position} [7:0]$$

$$\text{If } R0x108[12] = 1, \quad (EQ 2)$$

$$R = G = B = \text{pixel_position} [7:0]$$

Pixel values may be rounded, depending on the mode selected. See the example below. For YCrCb format, Cr or Cb will not increment the same way as Y due to the YUV422 format performing the UV subsampling.

For instance, *if $x = \text{pixel_position } 27$, then $Y = 27$, $Cb = Cr = 26$.*

Similarly, pixel values rounding can occur in 565RGB or 555RGB format change through the rounding operation on bit 1 or 2, respectively.



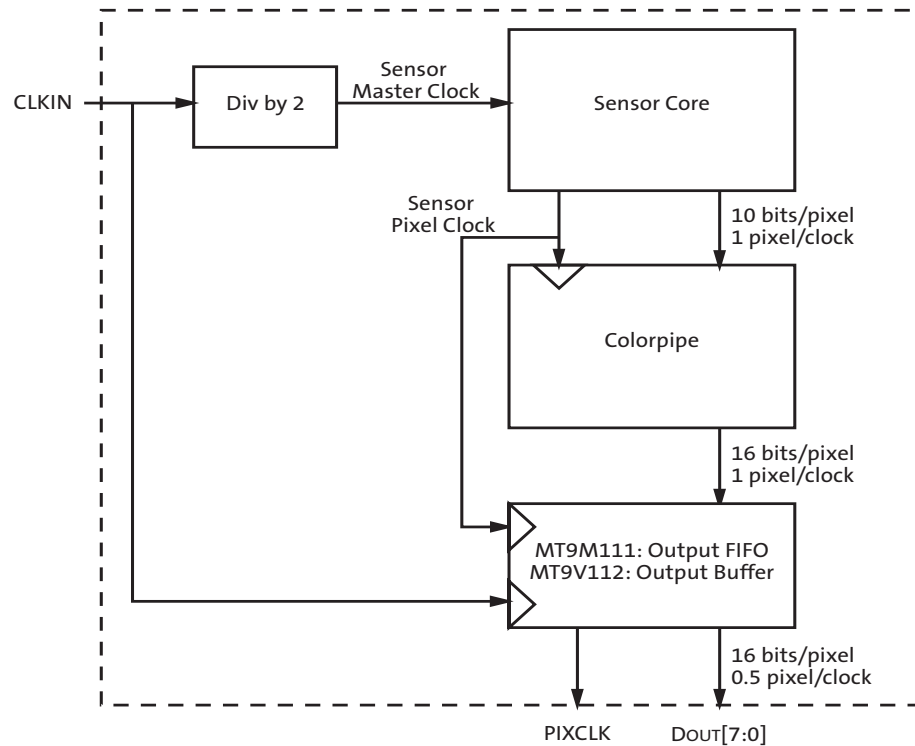
For instance, *if $x = \text{pixel_position } 28$: then $R = 32$, $G = 28$, $B = 32$* in 565RGB mode because *pixel_position 28* is equivalent to *b11100*; bit 2 is a 1, so R and B get rounded up to 32 (which is equivalent to *0b100000*); bit 1 is a 0, so G remains at or gets rounded down to 28 (which is equivalent to *0b11100*).

Clocks and Blanking

The sensor core generates master timing in the system as shown in Figure 8. The sensor core frame rate defines the overall image flow pipeline frame rate.

Horizontal blanking and vertical blanking are influenced by the sensor configuration and are also a function of certain image flow pipeline functions—particularly resize and use of the output FIFO to effect uniform clocking.

Figure 8: Sensor Core Clock



The image flow pipeline typically generates up to 16 bits per pixel, for example, YCbCr or 565RGB, but has only an 8-bit port through which to communicate this pixel data.

There is no phase-locked loop (PLL), so the primary input clock (EXTCLK) must be twice the pixel clock frequency.

To generate VGA images at 30 fps, the sensor core requires EXTCLK of 27 MHz. With power-on defaults, the camera is configured to deliver 15 fps at 12 MHz and automatically slows down the frame rate in low-light conditions to achieve longer exposures and better image quality.

The minimum EXTCLK frequency is 10 MHz.

Tuning Frame Rates

Actual frame rates are tuned by adjusting various sensor parameters. The sensor registers are in page 0. Sensor-timing-dependent registers are shown in Table 3.

Table 3: Sensor-Timing-Dependent Registers

Register	Default	Minimum	Function
R0x005	0x0026	0x0009	Horizontal blanking (units of sensor clock)
R0x006	0x0004	0x0003	Vertical blanking (number of rows, including black rows)

Border Pixels

The sensor core adds four border pixels all the way around the image.
The active image size is 648 x 488.

Blanking Calculations

Note: Consult Aptina's local FAE for a spreadsheet to assist with calculating line times.
The MT9V131 default blanking calculations are a function of context, as shown in Table 4.

Table 4: Blanking Parameter Calculations

Parameter	Calculation
PC_PERIOD: Sensor Pixel Clock Period	$PC_PERIOD = (2 / 27)\mu s = 0.074\mu s$
A: Active Data Time (per line): $R0x004 + 8 \text{ (border)} * PC_PERIOD$	$A = 648 * (2 / 27)\mu s = 48\mu s$
Q: Horizontal Blanking: $[R0x005] * PC_PERIOD$	$Q = 38 * (2 / 27)\mu s = 2.812\mu s$
Row Time: ¹ $Q + A$	50.764 μs
P1 Frame start blanking = $(R0x05 + 112) * 2$	= 300 pixel clocks = 300 master clocks = 25.0 μs
P2 Frame end blanking = 14 CLKS	= 14 pixel clocks = 14 master clocks = 1.17 μs
V Vertical blanking = $(R0x06 + 9) * (A + Q) + (Q - P1 - P2)$	= 20,778 pixel clocks = 20,778 master clocks = 1.73ms
F: Total Frame Time: $(R0x003 + [R0x006 + 2]) * (A + Q)$	$F = (487 + 4 + 2) * 50.764\mu s = 25026.652\mu s$

Notes: 1. The row or line rate is the same for both low power and full power configurations. This ensures that when switching modes, exposure time does not change; the pre-existing shutter width is valid.

Power Saving Modes

The MT9V131 supports a low power standby state. There are two methods for placing the devices in this state:

1. Assertion of the STANDBY signal (also known as “Hard Standby”)
2. Serial interface transactions to disable analog circuitry and gate clocks (also known as “Soft Standby”)

The two methods are equivalent and have the same effect:

- The analog signal chain and associated current and voltage sources are placed in a low-power state.
- While in standby, the state of the internal registers is maintained and the sensor continues to respond to accesses through its serial register interface.

There is a method to achieve even lower power consumption than the normal standby state. This state can be achieved by stopping the input clock (EXTCLK) while in normal standby. If the input clock is stopped, the sensor will not respond to accesses through the two-wire serial interface.

Low-power standby mode is exited through the same mechanism as entry into the mode. After the original standby source is de-asserted:

1. The internal clocks are restarted.
2. The analog circuitry is restored to its normal operating state.
3. The timing and control circuitry performs a restart, equivalent to writing `R0x00D[0] = 1`.
4. After this sequence has completed, normal operation is resumed.

Note: Power supplies must be maintained to the part during standby if any I/O will be toggled (for example, the two-wire serial interface). If supplies are removed and an I/O is driven HIGH, then the part is powered through the I/O pin and will consume significant current.

The following sections provide details on entering and leaving standby.

Entering Hard Standby

Assertion of the STANDBY and OE_BAR signal will always result in entry to the standby state. Refer to Figure 9, Hard Standby Operation Waveform for details. (“Resize and Zoom” on page 27.)

Following assertion of the STANDBY and OE_BAR signal, wait 24 master clocks before turning off the master clock.

If driven, device outputs will freeze in their pre-standby state. They should be considered to be in an unknown state.

The recommendation is to not assert the STANDBY signal during reset or for at least ten master clock cycles.

Leaving Hard Standby

De-assert the STANDBY and OE_BAR signal.

Ignore the first frame which was in-flight when the standby was entered.

Entering Soft Standby

To place the sensor core into standby assert OE_BAR, set R0x007[1] = 0, and R0x027[2] = 0.

When the serial interface transaction has completed, wait a further 24 master clocks before turning off the master clock.

Note: The two-wire serial interface receiver clocks are left running.

Leaving Soft Standby

To leave soft standby deassert OE_BAR, set R0x007[1] = 1 and R0x027[2] = 1.

Ignore the first frame which was in-flight when the standby was entered.

Avoiding Floating Inputs

Many MT9V131 signals use bidirectional pads. If these signals are not driven or pulled HIGH or LOW, they may float. Floating inputs must be avoided for two reasons.

- A floating input may cause unexpected results from the sensor.
- A floating input may cause excessive power consumption in the sensor.

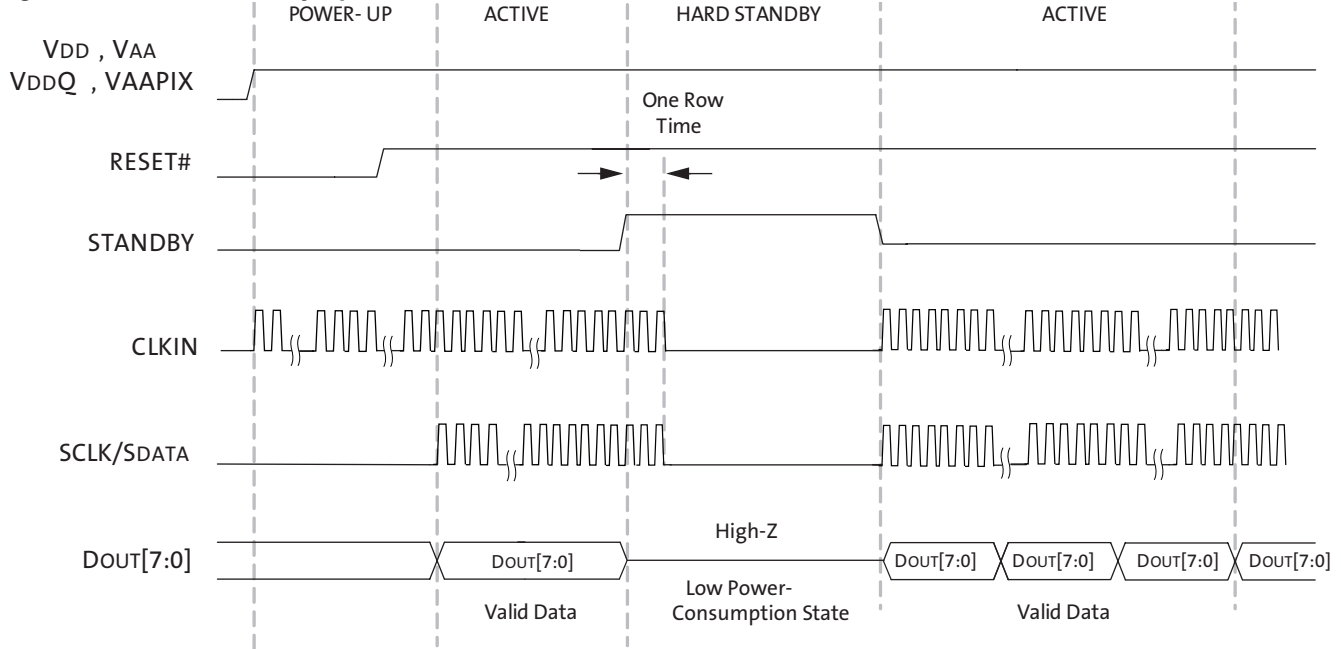
For these two reasons, standard design practice dictates that signal inputs should not be allowed to float for long periods of time.

Note: When using the output-enable control, ensure that all MT9V131 bidirectional pads are driven to a valid logic level before they enter a High-Z state.

Timing Summary

The timing summary for the hard standby mode of the MT9V131 is shown in Figure 9.

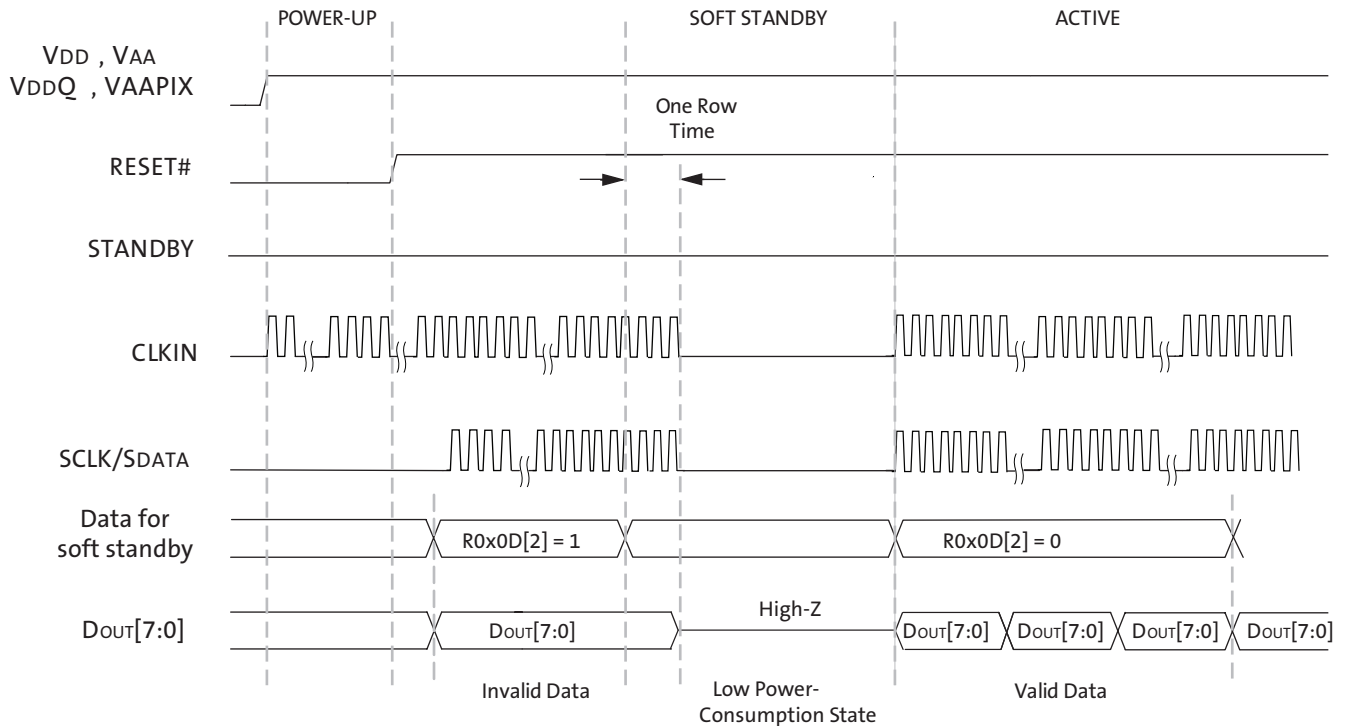
Figure 9: Hard Standby Operation Waveform



- Notes:
1. In the hard standby mode (RESET_BAR = 1, STANDBY = 1), the sensor outputs are tri-stated by default (FV = LV = PIXCLK = DOUT[7:0] = High-Z).
 2. When the sensor comes out of the hard standby mode, it maintains its previously stored register settings.
 3. The sensor draws the lowest current in the low power-consumption mode, when both EXTCLK and SCLK are off.
 4. If STROBE is not used it should not be tied HIGH or LOW, because it is an output-only signal.
 5. If the strobe is not used, it should be pulled down with a 10kΩ resistor so that it is not floating during standby.

The timing summary for the soft standby mode of the MT9V131 is shown in Figure 10.

Figure 10: Soft Standby Operation Waveform

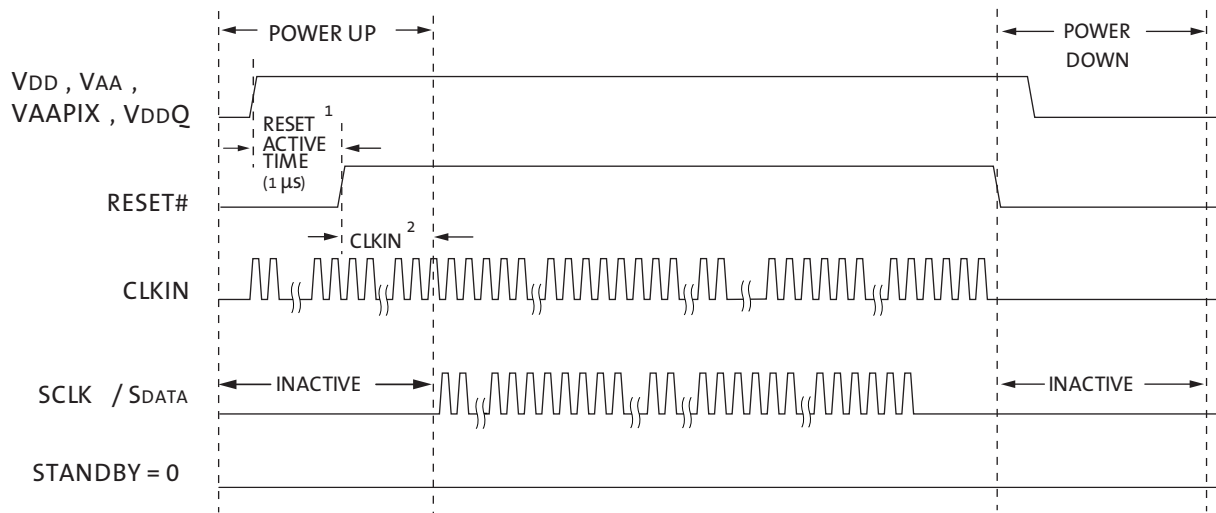


- Notes:
1. Soft standby is asserted/de-asserted by a two-wire serial interface to R0x0D[2]. In this mode, the analog block and the internal clocks are shut off. Also, the SOC outputs are NOT tri-stated by default. The leakage current consumed is lowest if the SCLK and the EXTCLK are turned off after one row time.
 2. When the sensor comes out of soft standby mode, it maintains its previous stored register settings.
 3. If STROBE is not used it should not be tied HIGH or LOW, because it is an output-only signal.

Power Sequencing

There are no constraints concerning the order in which the various power supplies are applied to the sensor provided the delay between the supplies is less than 1ms. It is recommended to power VDDIO and VDD before VAA and VAAPIX. As long as RESET_BAR is toggled following the stabilization of supplies, the sensor should function correctly. See Figure 11 for an example of both the power-up and power-down sequences.

Figure 11: Power-Up and Power-Down Sequences



- Notes:
1. For a safe reset to occur, EXTCLK should be running for 24 cycles, and it is recommended that STANDBY is LOW during the reset active time.
 2. After RESET_BAR is HIGH, the user must wait 24 EXTCLK cycles before initiating two-wire serial interface communication.

Power Up Sequence:

1. Turn-on the power supplies (VDD, VAA, and VAAPIX).
2. Start EXTCLK for more than 1 μ sec after the power exceeds 90% of VDD.
3. Set RESET_BAR to HIGH after 24 cycles of EXTCLK.
4. After RESET_BAR is HIGH, the user must wait 24 EXTCLK cycles before initiating two-wire serial interface transfers.

Power Down Sequence:

1. Stop EXTCLK and set it LOW.
2. Wait more than 1 μ s after setting RESET_BAR LOW.
3. Wait more than 1 μ s, then set all input signals LOW and then turn off VDD, VAA, VAAPIX.

Reset

The reset pin must be asserted for a minimum of 1μ EXTCLKs following stabilization of the power supplies. The master clock must be on.

Prior to the first clock edge following assertion of reset, the status of FV, LV, PIXCLK, and outputs DOUT[7:0] are undefined.

For MT9V131, with the first rising clock edge during reset, outputs settle to the following:

- FV = LV = PIXCLK = 0 (LOW)
- DOUT[7:5] = 0 (LOW)
- DOUT[4] = 1 (HIGH)
- DOUT[3:0] = 0 (LOW)

Following de-assertion of reset, wait 10 EXTCLK cycles before considering the device active and applying serial interface transactions or toggling STANDBY.

Resize and Zoom

Image reduction is controlled by zoom window size, output resolution, and pan offset registers compatible with the MT9V131. The reducer cannot expand pixels or lines, so each window setting must be the same size or smaller than the ones ahead of it. This restriction is expressed in Table 5.

Table 5: Window Setting Restrictions

Parameter	Notes	Constraints
Zoom width	Defines field of view. Always defined with respect to the full sensor window.	Sensor width \geq Zoom width.
Zoom height		Sensor height \geq Zoom height.
Output width	Output Resolution X Dimension	Zoom width \geq Output width. Sensor width \geq Output width.
Output height	Output Resolution Y Dimension	Zoom height \geq Output height. Sensor height \geq Output height.

Zoom width and height are always in sensor full resolution pixel units. The output width and height are always in output pixel units.

Zoom

Zoom is available when the output image is less than the sensor resolution. The interpolation mode needs to be “bilinear” for arbitrary zooming. If bilinear interpolation is enabled, the sensor automatically uses the correct interpolation algorithm to permit arbitrary zoom and resize. Row skip and column skip reduce sensor output by half in each dimension, which cuts the zoom range in half for a given output resolution. The zoom factor is the ratio of the current zoom window size to the full field of view.

Decimation control registers work to downsize the output image to any size. The output image size is specified in R0x1A7 and R0x1AA for horizontal and vertical sizes, respectively. For example, to downsize the VGA output to QQVGA set R0x1A7 = 160 and R0x1AA = 120. Whenever the output image is downsized, the zoom feature becomes available. To zoom in, program R0x1A6 and R0x1A9 with the size of the window to be decimated. For example, in QQVGA setting, R0x1A6 = 320 and R0x1A9 = 240 results in 2x zoom. Here the output image of 160 x 120 is created from a pre-decimation window of 320 x 240 instead of the full VGA 640 x 480.

Pan

Whenever the output image is zoomed, pan controls become available. To pan a zoomed image, program R0x1A5 and R0x1A8 to offset the pre-decimation window to the right and bottom respectively. When implementing a smooth zoom and pan, it is useful to synchronize the update of all decimation registers to avoid jerks in the output video. When writing a batch of decimation settings, set bit 15 of each datum to “1” to freeze the update. Set bit 15 of the last datum in the batch to “0” to enable normal operation. The entire batch of decimation settings will then be synchronously loaded on the next frame start.

Advantages of Using the Zoom Window Over the Sensor Window

It is possible to reduce the sensor window size as long as the zoom window is also reduced. Lens shading coefficients have to be recalculated or turned off if the sensor window is changed. It is much easier to use the zoom window.

Sharpness

Aperture Correction

Aperture correction has a much greater effect on sharpness than the interpolation algorithm option. The default aperture scale value is good for a wide variety of scenes. Increased aperture scale values produce sharper images but also amplify noise. See Table 6 for details.

Table 6: Aperture Correction Settings

Register	Value	Comment
R0x106	Bit 12 = 1	Enable aperture correction
R0x105	0x000F	Aperture scale maximum value.
R0x108	Bit 12 = 1	Output format control: Mode = RGB, which is slightly sharper than YCbCr.

On-the-fly Defect Correction

The MT9V131 has several defect correction techniques:

- 1D: Hot and dead pixels with known coordinates are corrected
- 2D: Hot and dead pixels are corrected on-the-fly
- 3D: Hot and dead pixel coordinates are determined on-the-fly, stored in a table and corrected

For the MT9V131, defect correction is enabled by setting R0x106[13] = 1.

Flicker Avoidance

To avoid flicker, the shutter width is made to be a function of the flicker frequency. The user must indicate to the AE unit how many line times constitute an integration time of 1/30 of a second to avoid 60Hz flicker, or 1/25 of a second to avoid 50Hz flicker.

The MT9V131 has automatic and manual flicker avoidance control (R0x15B[0]). If manual flicker avoidance is used, R0x15B[1] is used to specify 50Hz or 60Hz flicker.

Auto Exposure (AE)

Auto exposure provides each of the following functions:

- Target luminance maintenance adjusts a variety of controls within the sensor core and the IFP to maintain a target luminance (computed as a weighted average of pixels over the entire image) in response to changes in scene illumination.
- AWB change response modifies and updates target luminance settings in response to changes in AWB (color gains ratios) and flicker detection (shutter width factor).
- Illumination metrics management drives decreases in color saturation and sharpening for low illumination levels. (The target luminance control settings are used as an illumination level metric.)

AE is enabled by setting bit 14 in the IFP operating mode control register (R0x106[14]).

Controlling AE Speed and Sensitivity

AE speed and sensitivity are controlled by R0x12F.

Hysteresis Control

The first AE field R0x12F[2:0] determines how much to blend previous luminance measurements (R0x14D) with the current luminance measurement (R0x14C).

- A value of “0” results in using the current luminance as the criterion for determining whether auto exposure is within range of the target luminance.
- A value of “7” results in only 1/128 of the current luminance being blended into the previous luminance value.

Sample Rate Control

The second AE field R0x12F[5:3] determines how many frames to skip before sampling current luminance. Generally, this is set to “0”, as the AE incorporates the intelligence on how often to sample.

Adaptation Speed

R0x12F[7:6] controls the rate at which AE moves toward the target luminance, once it calculates a need to adjust the exposure.

For fast adaptation, set this control to “2”, which causes AE to adjust to the target luminance in two steps.

AE Window and Backlight Compensation

AE defines two windows within the image: the big window and the little window. The big window is defined by the AE horizontal boundaries register R0x126 and the AE vertical boundaries register R0x127. The little window is defined by the AE center horizontal boundaries register R0x12B and the AE center vertical boundaries register R0x12C. The region of the little window is usually within the region of the big window.

Backlight Compensation Control

The window that defines which pixels are used to compute the image average luminance is controlled by the AE backlight compensation control, in R0x106[3:2].

A value of zero selects the big window; a value of one selects the little window; and a value of two or three selects backlight compensation mode, where the pixels within the little window are weighted two times more heavily than the pixels in the region between the big and little windows.

Window Coordinates

Each of the window boundary registers is split into two bytes. The low byte of the horizontal registers controls the window left coordinate, and the high byte controls the window right coordinate. The low byte of the vertical registers controls the window top coordinate, and the high byte controls the window bottom coordinate.

The coordinates themselves represent a fraction of the width (for horizontal coordinates) and the height (for vertical coordinates) of the currently viewed image. Calculate this fraction by dividing the byte value by 255 (0xFF).

- A value of zero is interpreted as the left most (horizontal) or topmost (vertical) boundary of the image.
- A value of 255 (0xFF) is interpreted as the right most (horizontal) or bottommost (vertical) boundary of the image.

The top boundary of the big window is inset slightly from the topmost boundary of the image, to reduce the influence of the illumination source itself (ceiling lighting or sky) on AE.

Exposure Step Size

The exposure step size, which is 1/16 of the current exposure level, is fundamental to the operation of AE. All exposure adjustments are in step multiples of 1.0625. With default settings, this yields more than 220 exposure adjustment steps.

AE Zones

AE defines 25 exposure zones, 0 through 24, as described in Table 7.

Table 7: AE Zones

Zone	Description
Zone 0:	High illumination. Exposure controls are the sensor core shutter width R0x009 and shutter delay R0x00C.
Zones 1–23:	Moderate to low illumination. Exposure controls are the sensor core shutter width R0x009 and the color channel gains R0x02B, R0x02C, R0x02D, and R0x02E.
Zone 24:	Low to very low illumination. Exposure controls are the sensor core color channel gains R0x02B, R0x02C, R0x02D, and R0x02E, the analog-to-digital converter (ADC) voltage references R0x041, and the IFP digital gains R0x162.

- Notes:
1. When any combination of AE, AWB, or flicker detection is enabled, all of the above registers (as shown in Table 7) are volatile and can change value during the vertical blanking interval. To take full manual control of exposure: AE, AWB, and flicker detection must all be disabled (bits 1 and 11 of register R0x108 must be set to “0”).

The AE zone cannot be set directly, but it can be forced to a single value by manipulating the AE zone limits register R0x137 to bind it to a single value.

The shutter width in zones 1–24 are set to be a multiple of a base integration time corresponding to 1/120 of a second (for 60Hz flicker abatement), or 1/100 of a second (for 50Hz flicker abatement). The base integration times are defined in registers from R0x159 through R0x15A. The values of these registers are the shutter widths used for zone 4 in the following situations:

- R0x159: Shutter Width for One Frame Integration Time for 60Hz AC
- R0x15A: Shutter Width for One Frame Integration Time for 50Hz AC

The derived shutter widths for zones 1–3 are scaled down versions of the zone 4 value. The derived shutter widths for zones 5–24 are scaled up versions of the zone 4 value.

Zone 0 Exposure

In zone 0, the shutter width can assume any value between one and the shutter width value for zone 1. In general, flicker abatement is lost in zone 0, since the integration time is smaller than a cycle of illumination.

When the shutter width is less than 12, shutter delay smooths the steps between shutter width values, much as the sensor gains were used to smooth the steps between zone changes.

Shutter delay subtracts a fraction of a row of integration time from the total integration time. At the minimum exposure limit, the shutter width is one row, and the shutter delay is at its maximum value.

Shutter Delay Maximum Value

Overhead is associated with the beginning and end of row setups. The maximum shutter delay is equal to the sum of the number of row pixels and the horizontal blanking time, minus the overheads. Shutter delay can be between 75 and 80 percent of the total row time. The greater this percentage, the finer the exposure control is under extremely bright illumination conditions.

The AE shutter delay limit register R0x139 holds the maximum shutter delay value.

Contact your local Aptina FAE for a program that automates the calculation of appropriate values for the shutter delay limit register.

Minimum Analog Gain

The analog gain in Zone 0 is set by the lower byte of the AE gain limits register R0x136. The value in the register must reflect the gain value necessary to achieve the full range of codes from the ADC. Normally, this value is close to 1.0, but its nominal value is determined by sensor core characterization.

The numerical minimum gain value is computed by dividing this register value by 16.

Zones 1–23 Exposure

In zones 1–23, the color channel gains are used to generate intermediate exposure values between the relatively coarse jumps in exposure when moving from one AE zone to the next.

Zone 24 Exposure

The several factors that influence zone 24 exposure are detailed in the following sections.

Shutter Width

There is a final change in shutter width going from zone 23 to zone 24.

Maximum Analog Gain

Following the final change in shutter width, analog gains are allowed to increase up to the maximum value limit set by the upper byte of the AE gain limits register R0x136[15:8].

ADC Voltage References

After the maximum analog gain has been reached, the high and low ADC voltage references are manipulated (symmetrically) to achieve up to another factor of two increase in effective analog gain. The smaller the difference between the ADC voltage references, the greater the gain. The values of the ADC voltage references can be read from the V_{REF} DACs register R0x041. The low voltage reference (V_{REFLO}) is in R0x041[3:0], and the high voltage reference (V_{REFHI}) is in R0x041[7:4]. The AE ADC gain control register R0x13D controls the ADC. The minimum limit value for V_{REFLO} is R0x13E[11:8]. The minimum limit value for V_{REFHI} is R0x13E[3:0]; the maximum limit value for V_{REFHI} is R0x13E[7:4].

Caution Although this register is listed here for completeness, we do not recommend modifying the value of register R0x13D, both to ensure the stability of the AE algorithm and to prevent possible damage to the sensor.

After V_{REFHI} R0x041[7:4] is reduced to the value in R0x13D[3:0], digital gains in the IFP must be taken into account.

Digital Gain 1

Digital gain1 is applied before lens shading correction and scales the pixel values before entering the IFP. Its minimum value is 1.0; its maximum value is limited by the lower byte of the AE digital gain adjustment limits register R0x167[7:0]. The current value of digital gain1 can be found in the lower byte of the AE digital gain monitor register R0x162[7:0].

This is usually a read-only register. Disable AE to write to it. The numerical gain value represented by these registers can be calculated by dividing the values by 16.

Digital Gain 2

After digital gain1 has reached its maximum value, digital gain2 is employed to further boost the pixel values under extreme low-light conditions. Digital gain2 is unusual, in that its value is added to (not multiplied by) the per color gains computed in lens shading correction.

When digital gain2 has reached its maximum value—determined by the upper byte of AE digital gain adjustment limits register R0x167[15:8]—AE has reached its exposure adaptation limit. The current value of digital gain2 can be found in the upper byte of the AE digital gain monitor register R0x162[15:8].

Normally, this is a read-only register. Disable AE to write to it. The numerical gain value represented by these registers is calculated by dividing the values by 16.



Target Luminance

The target luminance is contained in the low byte of register R0x12E, and the accuracy of the target luminance is defined in the high byte of register R0x12E. As long as the time-averaged luminance low byte of R0x14D is in the range of $R0x12E[7:0] \pm R0x12E[15:8]$, AE is stable and the exposure criterion has been met.

Extreme Low Light Condition

If the scene illumination drops to a level that is beyond the range of AE to bring the average luminance into range, the sensor will go into a standby state. The sensor then waits for either the scene illumination to improve, or for the target luminance (under host control) to be lowered to a level that can be satisfied.

Flicker Abatement Dwell

The low byte of register R0x146 can be programmed with a luminance threshold to remain in zone 1 under high-illumination conditions (rather than transition to zone 0, where flicker abatement is impaired). If the average luminance is less than this threshold, the upper boundary on average luminance $R0x12E[7:0] + R0x12E[15:7]$ is overridden. The net effect is that the image is allowed to become brighter than normal to remain in zone 1 longer. By default, R0x146 is set to "0," disabling this feature.

Brightness Control

An offset can be added at four points in the color pipeline, as discussed in the following sections.

Scheme 1: Pre AE, Pre Gamma

This register R0x13C offset affects the luminance of the image, as well as the luminance being compared against the exposure target. Increasing this register yields a net effect of decreasing perceived contrast, without significantly increasing the output luminance level.

Scheme 2: Post AE, Post AWB, Modify Gamma

These registers R0x153–R0x158 shift the gamma curve upwards. In effect, add an offset to the knee points of the gamma curve and clamp them to the maximum output value. This happens after the measurement engines gather their data, and has a uniform effect on YCbCr and RGB output. It also copes with some of the problems that could arise from too large an offset applied elsewhere.

Scheme 3: Post Gamma

In R0x134 the lower 8 bits of this register are used to bias the luminance in YCbCr space, and the upper 8 bits are used to bias the RGB values (in 8 bits per component RGB space). The default values are 16 (decimal) for YCbCr space, and 0 for RGB space.

Effects of the Various Brightness Schemes

Scheme 1 creates an unintended effect on AE and AWB. Scheme 2 compresses the dynamic range, but leaves most of the image visible. Scheme 3, the easiest to implement with minimal adverse effect on AE and AWB, tends to shift the effective gamma exponent for the viewing display. Scheme 3 works like the brightness knob on a monitor that shifts the effective gamma; it is applied in the processing pipeline before the display device gamma takes effect.

In general, modifying the brightness reduces the dynamic range, and produces a “washed out” appearance in the image. Table 8 provides values for different brightness levels using scheme 3 by setting the low order 8 bits of R0x134. Table 8 lists both the low order 8-bit value and the entire range of the 16 bits of R0x134.

Table 8: Scheme 3 Brightness Settings

Brightness Level	R0x134 RGB (High 8 bits)	R0x134 YCbCr (Low 8 bits)	R0x134 (Full 16 bits)
0	0	161	16
1	8	24	2,072
2	16	32	4,128
3	24	40	6,184
4	32	48	8,240
5	40	56	10,296
6	48	64	12,352
7	56	72	14,408

Auto White Balance (AWB)

AWB Overview

The white balance module adjusts image colors when illumination changes. White balance contains a proprietary measurement engine and a compute engine. The measurement engine processes the image stream to detect the type of illumination. The compute engine derives values for sensor gains, CCM coefficients, and SOC digital gains from these statistics.

Figure 12 shows where the controls available to the white balance unit fit into the sensor (the sensor red and blue gains) and the color pipe (the CCM coefficients and the SOC digital gains). It also shows the place in the color pipe where the AWB measurement engine (ME) obtains color data for generating statistics.

Figure 12: AWB Sensor Controls

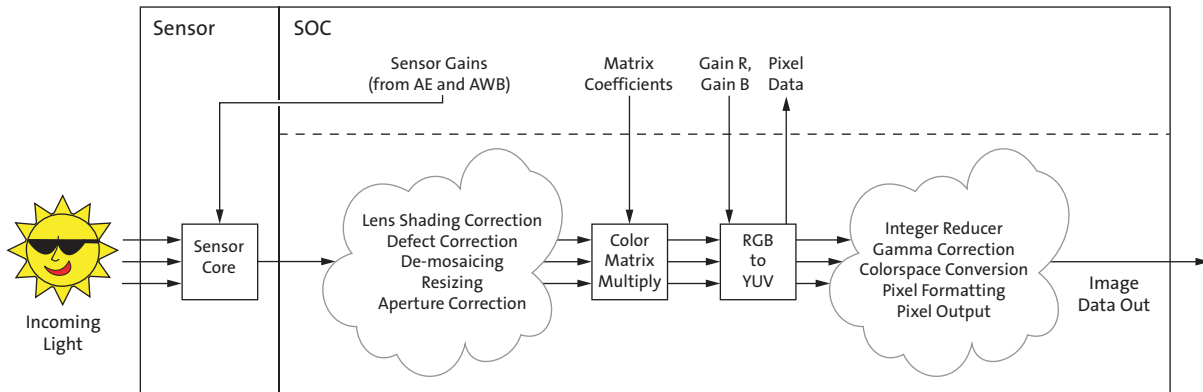
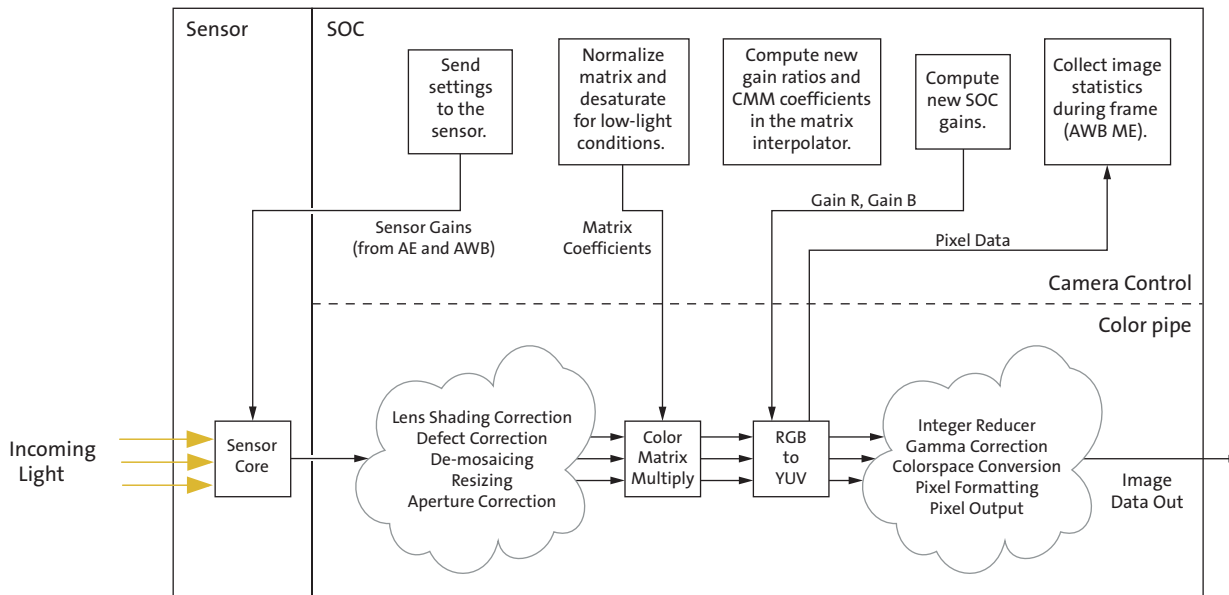


Figure 13 on page 38, expands upon this and adds the processing flow that occurs in the white balance unit between frames. Throughout a frame, the AWB ME collects data from the pixel stream and computes a set of statistics used by the other white balance computations. At the end of a frame, white balance begins by computing new SOC digital gains from the AWB ME statistics. It also applies a smoothing function to avoid abrupt changes in color.

Next, white balance determines if the CCM and sensor gain ratios need to change, and updates them accordingly. This is followed by matrix normalization (if enabled) and matrix desaturation in low-light conditions (if appropriate). Finally, white balance sends any new sensor gains to the AE unit, which writes them to the sensor before the next frame begins.

Figure 13: AWB Flow Processing



Color Correction Matrices

Color correction matrices consist of nine coefficients. Each coefficient is represented by a sign bit, a scale factor, and a scaled magnitude. The magnitudes are scaled to provide additional range for the coefficient values.

The coefficients are computed from the base and delta matrix values. In the abstract, the white balance unit interpolates coefficients from the red-rich and blue-rich matrices, which in turn are derived from calibration photos. In practice, registers store the base matrix coefficients (which are the averages of the corresponding red-rich and blue-rich coefficients) and the delta matrix coefficients (which are the differences between the corresponding red-rich and blue-rich coefficients). Scale factors are chosen for each of the nine coefficients to preserve as much precision as possible for the base and delta values without overflowing the representation.

In manual white balance mode, the white balance unit produces the base matrix coefficients. In AWB mode, the white balance unit incrementally adds or subtracts the delta matrix (divided by 128) from the current matrix to compute a new color correction matrix.

If the base matrix is considered to correspond to matrix position 64, then the red-rich calibration matrix lies at position 0, which is equivalent to base matrix - $64 \times$ delta matrix. The blue-rich matrix is at position 127, which is base matrix + $63 \times$ delta matrix.

White Balance Settings

There are two ways to approach preset white balance settings:

- Simple AWB
- Full-manual white balance

The following sections explain each approach.

Simple AWB

The easiest approach is to let the AWB algorithm adjust on its own. The primary downside of this approach is that, in the worst case, this takes several seconds to converge—128 frames for the matrix to move between extremes, plus a few more frames for the algorithm to fine tune the results.

The worst case for the algorithm is switching between predominantly red-rich light, such as incandescent light, and predominantly blue-rich light, such as sunlight.

White Balance Preset as a Matrix Position

One way to set a preset white balance matrix is to keep AWB turned on, but constrain the algorithm to a predetermined matrix position. It could take AWB a couple of seconds to adapt, during which time the CCM moves to the desired position.

To use this technique, set both fields in the color correction adjust limits register R0x124 to the desired matrix position. For example, if the chosen matrix position is 0x23, program R0x124 = 0x2323.

To determine the desired matrix position for various lighting conditions, follow these simple steps:

1. Set up AWB as normal.
2. Point the camera at a scene illuminated by the desired conditions. Choose colorful areas, or a MacBeth chart, if possible.
3. Let AWB settle.
4. Read the current matrix position from register R0x112. Table 9 provides useful matrix position examples.

Table 9: Suggested Matrix Positions for Varying Conditions

Lighting	Matrix Position Read from R0x112	Matrix Position Constraints R0x124
Cloudy	0x7F	0x7F7F
Compact White Fluorescent	0x2D	0x2D2D
Daylight	0x7F	0x7F7F
Incandescent	0x00	0x0000
U30 TL84 bulb	0x00	0x0000

Full Manual White Balance

A simple way to accelerate convergence to a desired mode is to reset the CCM to its central position in circumstances where this may be helpful. In a worst case scenario—transitioning between extreme lighting conditions—this can cut the convergence time in half.

To implement this technique:

1. Determine the current matrix position. If unknown, register R0x112 contains the position, represented as a 7-bit unsigned integer.
2. Determine the goal matrix position. Expose the camera to the target lighting and record the matrix position AWB finds after convergence. (Incandescent light produces matrix positions near “0”, while sunlight produces positions near 127.)
3. If the current and goal positions lie on opposite sides of position 64, or if the goal is closer to position 64 than the current position ($|P_{goal} - 64| < |P_{goal} - P_{current}|$), AWB can converge faster from the midpoint than from the current position. To jump to the midpoint:
 - 3a. Switch to manual white balance mode: R0x106[1] = 0 and R0x106[15] = 1.
 - 3b. Read the matrix position to check that the matrix has moved.
 - 3c. Return to AWB mode: R0x106[1] = 1 and R0x106[15] = 0.

See “AWB Speed and Stability” on page 41 for further information on accelerating CCM movement.

Optional Approach to Manual White Balance

After step 3 above, enabling both AWB and manual WB (that is, R0x106[1] = 1 and R0x106[15] = 1) causes the CCM to remain fixed (specifically, the base matrix), while IFP red and blue gains will be controlled by the AWB mechanism to fine tune the color balance.

Determining settings for a particular white balance preset requires these steps:

1. Set up known-good AWB matrices using previously calibrated values.
2. Let AWB adapt to the desired lighting condition.
3. Record the matrix position R0x112 and the red and blue IFP gains R0x113 and R0x114 determined by the AWB algorithm.
4. Compute signs and magnitudes of the preset matrix coefficients, and the gain ratios from the known good AWB matrices and the matrix position. (These were used in steps 1 and 2 of “Full Manual White Balance”.)
5. Use the recorded red and blue IFP gain values for the values used in step 3 of the preceding procedure.

AWB Speed and Stability

The speed at which AWB adjusts is controlled by two settings:

- R0x125[6:3], the frequency of matrix adjustments
- R0x125[2:0], the SOC digital gain smoothing control

In video or web camera applications, AWB should be relatively slow to avoid sharp changes in color due to slight scene changes, such as people walking by or camera panning.

The higher the frequency of matrix adjustments (that is, fewer frames between adjustments), the faster the matrix can move when the lighting changes. The larger the contribution of the newly computed SOC digital gain to the value actually used by the SOC (that is, smaller values for the control), the faster the SOC digital gains adapt.

With high matrix adjustment frequency, the SOC digital gain must change quickly to keep pace. Otherwise, the matrix adjustment can overshoot its target, and when the digital gains do catch up, the matrix will have to move in the opposite direction. This can lead to oscillation. For fast matrix adjustments, keep the gain smoothing control at small values.

If the AWB control rates are set too fast, small changes to an image can shift the color balance measurements and cause AWB to react prematurely. The net effect is that the color balance settings appear twitchy—little things cause sweeping change. This is especially true if the digital gains react too slowly (see above description), as small, transient scene changes can trigger oscillation.

Setting the controls to react too slowly can lead to lengthy convergence times. Ensure that a suitable balance is identified for your implementation.

The following process should be used to accelerate AWB convergence when using slower AWB settings. If the host can determine that a large AWB change is needed, reset the matrix to the midpoint (matrix position 64) by switching to manual white balance for one frame by setting R0x106[15] = 1. When manual white balance is subsequently turned off by clearing R0x106[15], AWB will resume from this central position. This is helpful for transitions between extreme color balance points, such as switching between incandescent lighting and sunlight.

Loading Color Correction Matrices

Loading color correction matrices involves two processes.

1. Load the CCM registers R0x109–R0x111 and R0x115–R0x11E that represent the matrices.
2. Force the new matrices into the matrix interpolator by toggling R0x106[15].

For the other matrices—the base matrix, base gain ratios, delta matrix, and delta gain ratios—two events must occur to affect the output of the SOC as described in the following sections.

To change the active CCM during auto and manual white balance modes:

1. Load new matrices into the CCM registers.
2. Toggle R0x106[15].

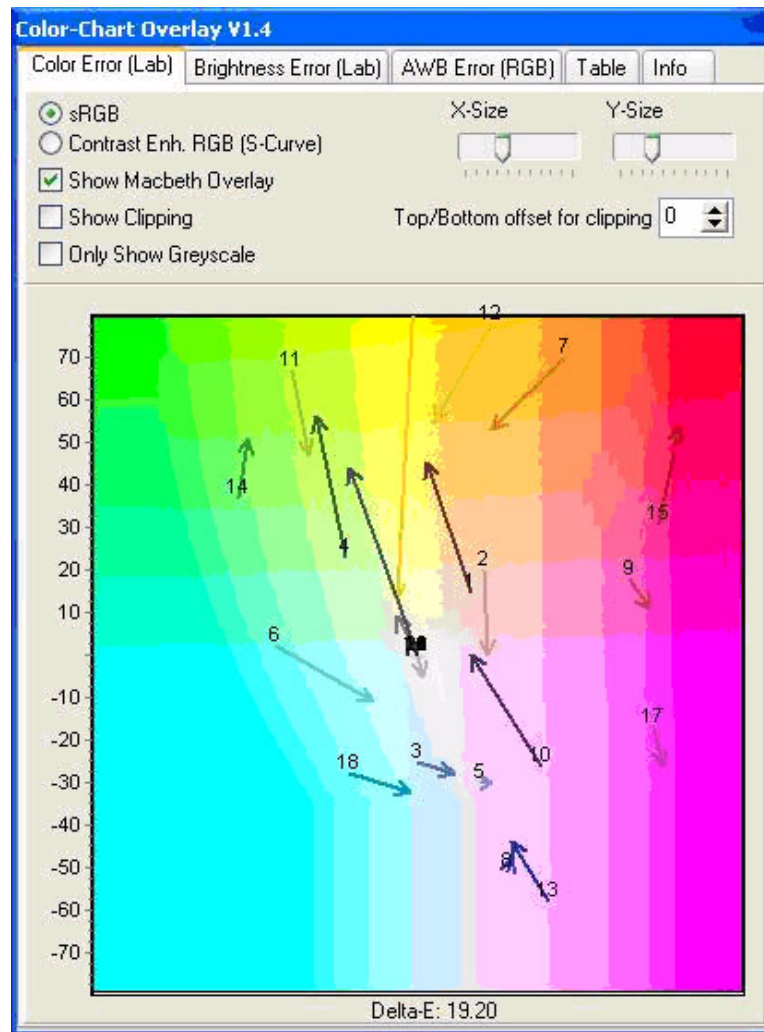
AWB Tuning

AWB is tuned first, followed by CCM.

Initialization

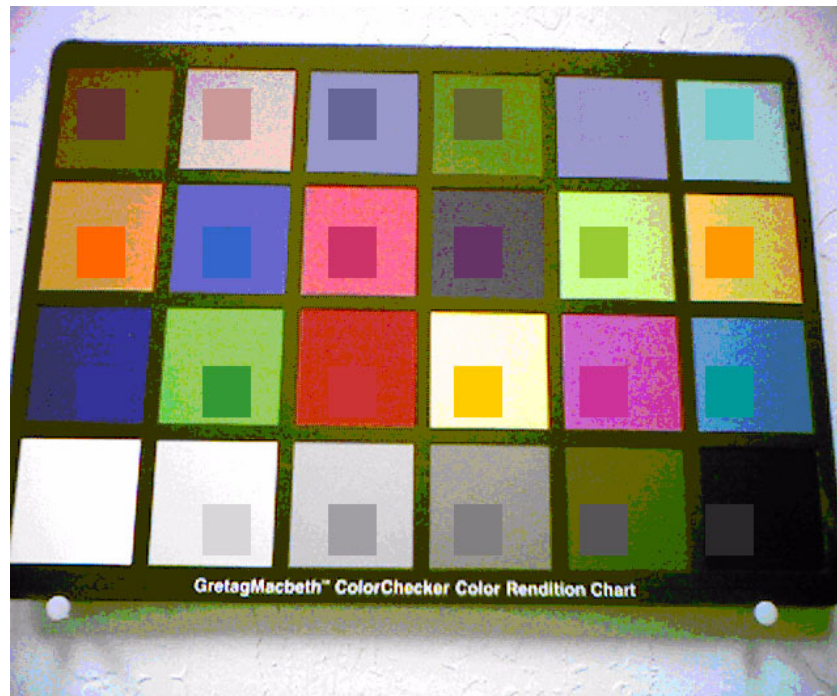
1. Set up the sensor in a light box to view a Macbeth chart. The chart should be centered within the sensors field of view. Ideally, the light source should be behind the imager, not on top (that is, for a Judge II light box, remove the light source from the box top and stand it up behind the sensor to be tuned).
2. Launch DevWare and reset the sensor.
3. Click on the Plug-Ins --> Color-Chart Overlay menu item; this brings up the Color-Chart Overlay dialog box. On the Color Error (Lab) page, verify that sRGB is checked and that Show Macbeth Overlay is enabled. See Figure 14.

Figure 14: Color-Chart Overlay: Color Error (Lab) Page



4. Align the reference color squares in the DevWare field of view with the Macbeth chart squares; adjust the X-Size and Y-Size so the reference squares do not touch the black sides of the Macbeth chart. See Figure 15 on page 43.

Figure 15: Color-Chart Overlay: Reference Squares Aligned with Macbeth Chart



Sensor Setup

1. Switch the sensor to the snapshot context, Context B.
2. Disable scaling/zooming. The image should be full size. Set the mirror/flip bits if needed.
3. Set saturation to 100 percent. Bring up the Sensor Control --> Enhancement Saturation dialog box. Set the Saturation slider to 100 percent.
4. If 150% saturation is used, the Contrast Enhanced RGB (S-Curve) setting can be used in the Color-Chart Overlay --> Color Error (Lab) dialog box.
5. Set Gamma to 0.45. In the Sensor Control dialog box, --> Gamma, move the Gamma Correction Table slider to 0.45.
6. The initial tuning should be done with the S/W Gamma radio button checked. This allows AWB/CCM targets to be reached faster. After the initial tuning is complete, the H/W Gamma radio button is checked. H/W Gamma is used to verify final results and make adjustments.
7. Set the Black Correct to "8." In the Sensor Control dialog box, --> Gamma, move the Black Correct slider to "8."
8. Load lens correction values using the Presets dialog box.
9. Set the digital gains to 1.0. In the Register panel, set R0x148[7] to "1."
10. Using the DevWare presets dialog box, load the default Color Correction Matrix for the sensor. Do not load the fast CCM setting that is present in some DevWare INI files.
11. Set the Auto Exposure Luma target. It is important to have the correct exposure when doing these procedures.
 - a. In the Register panel set Auto Exposure Target and Precision Control R0x12E[7:0] to 0x64.
 - b. Add the Auto Exposure Current Luma Monitor (R0x14C) to the watch page by clicking the right mouse button on 0x14C (and clicking on **Add to Watch** menu

- item). Monitor this register for the correct Luma value. Wave a hand in front of the sensor if R0x14C is not 0x44.
12. Monitor register R0x112, **Current Color Correction Matrix Position**. Add this register to the watch page by right-clicking on the register in the Register panel.
 13. Verify that R0x106 is set to default (AWB, AE enabled, CCM bypassed, LC enabled).
 14. Set the **Delta Sensor Core Gain Ratio** register (R0x15F) equal to 0.

Procedure to Adjust Base Sensor Core Gain Ratio

The goal is to adjust the Base Sensor Core Gain Ratio register (R0x15E) to minimize the AWB error. The procedure is done twice under two light temperatures: Daylight (D65, 6500 degrees Kelvin) and Incandescent (A, 2850 degrees Kelvin).

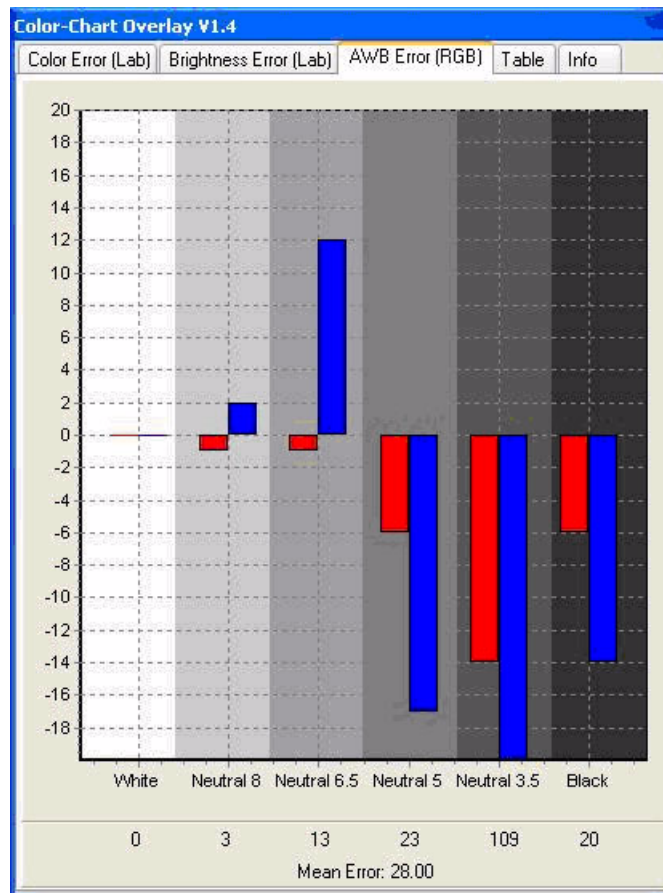
1. Start the procedure with Incandescent light.
 - 1a. For Incandescent, set R0x124 = 0000.
 - 1b. For Daylight, set R0x124 = 0x7F7F.
 - 1c. Using the Color-Chart Overlay - AWB Error (RGB) page.
 - 1d. Using the Analysis Graph-RGB chart.

Use the Color-Chart Overlay method to perform the AWB tuning procedure; this method gives a numerical value (that is, the Mean Error on the AWB Error (RGB) page) for tuning precision. The Analysis Graph can later be used to check results.

Using the Color-Chart Overlay - AWB Error (RGB) page

1. In the Color-Chart Overlay dialog box, click on the AWB Error (RGB) page. See Figure 16 below.
2. In the Register panel, adjust R0x15E to center the red and blue color bars as close to zero as possible. When a new value for R0x15E is entered, toggle R0x106[15] to 1, then 0 to load the AWB values (a checkbox to do this is located on the Sensor Control --> White Balance --> Color Correction Matrix page --> Manual WB). R0x106[15] must be toggled; otherwise register values are not loaded into the SOC.
3. Wait for the AWB to settle before adjusting R0x15E again.
4. Focus adjustments on the middle 3 gray Macbeth chart squares (that is, Neutral 8, Neutral 6.5 and Neutral 5) while tuning. The outer squares may be thrown off by lens roll-off.
5. The Mean Error value (located at the bottom of the AWB Error (RGB) page), should be as low as possible (values of 4 to 6).

Figure 16: Color-Chart Overlay: AWB Error (RGB)



Using Analysis Graph - RGB

1. Click the Graphs button to bring up the Analysis Graph - RGB window.

Figure 17: Analysis Graph - RGB Window Showing R,G,B Components of the Macbeth Chart



2. Click on the Intensity tab.
3. In the Info panel, click on **Mouse Selection** --> **Row** to display a horizontal line on the image. Position the line through the Macbeth chart's lowest color row (the white, neutral, ... black squares). The line should intersect the reference patches displayed by the Color-Chart Overlay.
4. In the Registers page, alter R0x15E to adjust the red and blue color lines as close as possible to the green line. When a new value for R0x15E is entered, toggle R0x106[15] to 1, then 0 to load the AWB values (a checkbox to do this is located on the Sensor Control --> White Balance --> Manual WB). R0x106[15] must be toggled otherwise register values are not loaded into the SOC.
5. The AWB setting is best when:
 - 5a. The R,G,B values are at the same level and
 - 5b. The R,G,B values for each square line up with the reference patch values (that is, when the RGB value on the side of the reference patch line up with the reference patch R,G,B values).
6. Monitor register R0x112 for settled values before doing another adjustment.
7. Once the R0x15E value for incandescent is determined, record that number to a safe place and switch lighting to Daylight (switch R0x124 to 0x7F7F). Do the same procedure using Daylight and record that number.

Final Results

Two values are recorded, for example:

- Daylight: D65: 0x486E
- Incandescent: A: 0x6836

1. Compute the resulting Base Matrix Gain Ratio (R0x15E) and Delta Matrix Gain Ratio (R0x15F) values:

Table 10: Base and Delta Gain Ratios

Base Gain Ratios			
Blue	$(0x48 + 0x68)/2$	= 0x58	R0x15E Result: 0x5852
Red	$(0x6E + 0x36)/2$	= 0x52	
Delta Gain Ratios			
Blue	$ABS(0x48 - 0x68)$	= 0x20	R0x15F Result: 0x2038
Red	$ABS(0x6E - 0x36)$	= 0x38	

Note: ABS = Absolute Value.

2. Save the resulting R0x15E and R0x15F values to an INI file. They will be used in the CCM adjustment.

To check the results, perform the following steps:

1. Set R0x124 to 0x7F00 (to allow full AWB adjustment).
2. Enable Digital Gains (R0x148[7] = 0).
3. Write the Base and Delta values to their respective registers. Toggle the MWB matrix using R0x106[15].
4. S/W Gamma can remain enabled throughout this process. H/W Gamma can be used to check the final results for both AWB and CCM.
5. Using the Color-Chart Overlay, check the AWB Error (RGB) page for both Daylight and Incandescent. A Mean Error value of 6 to 7 is considered good.
 - a. Adjust the Blue and Red values as necessary using the procedure in “Using Analysis Graph - RGB” on page 50.

Color Correction Matrix Tuning

Initialization

1. For CCM tuning, follow the steps 1 through 4 in the section on Initialization of “AWB Tuning” on page 42. Then follow steps 1 through 8 in “Sensor Setup” on page 43.
2. Load the Base and Delta Sensor Core Gain Ratios (R0x15E, R0x15F) that were determined in AWB tuning.

Procedure

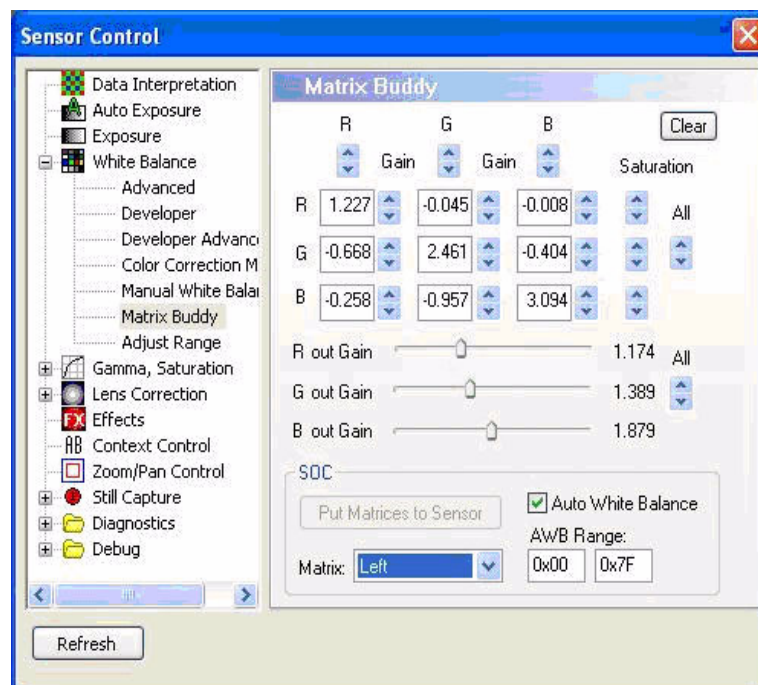
The goal of this procedure is to adjust the Color Correction Matrix so that Macbeth chart squares 13, 14 and 15 (Blue, Green, Red) are as close as possible to reference values. If possible, square 2 (Light Skin) should also be close to reference.

The procedure is duplicated for two light temperatures:

- Incandescent (A, 2850 degrees Kelvin).

- Daylight (D65, 6500 degrees Kelvin)
 1. Start the procedure with Incandescent light.
 - 1a. For Incandescent, set R0x124 = 0000.
Incandescent uses the left Color Correction Matrix.
 - 1b. For Daylight, set R0x124 = 0x7F7F.
Daylight uses the right Color Correction Matrix.
 2. Adjust the CCM values using one of the two methods given below.
 - 2a. Using the Color-Chart Overlay - Color Error (RGB) and the Table pages
 - 2b. Use the Color-Chart Overlay method to perform the CCM tuning procedure; this method gives a numerical value (that is, the Delta-E value on the Color Error (Lab) page) for tuning precision.
 - 2c. Using the Analysis Graph-RGB chart
 - 2d. If desired, the Analysis Graph can later be used to check results.
 3. Bring up the Sensor Control Panel --> White Balance --> Matrix Buddy page. AWB should be enabled with the AWB range 0 to 0x7F. See Figure 5 below.
 4. For the Incandescent light source, set the Matrix to left. For Daylight light source, set the Matrix to right.
 5. Using either of the measurement methods below, adjust the matrix. After a single adjustment, press the Put Matrices to Sensor button to load the SOC matrices. Monitor register R0x112, Current Color Correction Matrix Position, for AWB settling.

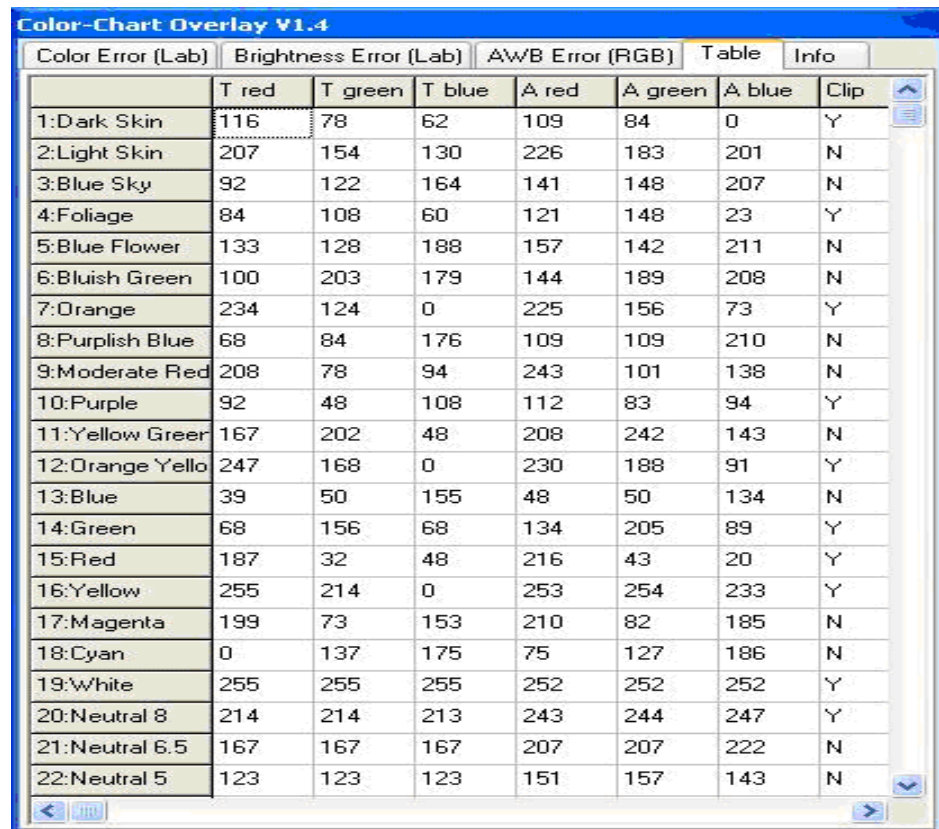
Figure 18: Matrix Buddy Page



Using the Color-Chart Overlay - Color Error (RGB) and the Table pages

1. In the Color-Chart Overlay dialog box, click on the Color Error (RGB) page. Figure 14 on page 42.
1. Each of the Macbeth chart colors is represented by an arrow and a number. Numbers 13, 14 and 15 correspond to red, green and blue. Important considerations for this page are the length of the arrow and the direction of the arrow. The shorter the arrow, the better. Ideally, the arrow direction should correspond to the color of the patch (that is, blue arrow pointing to blue).
2. Using the Sensor Control --> White Balance --> Matrix Buddy page, adjust the matrix for each color component (R,G,B) of each square (R,G,B and others). Click the Put Matrices to Sensor button to write new values to the sensor.
3. Concentrate only on the red, green, and blue squares.
4. Monitor register R0x112 for settled values before doing another adjustment.
5. To view the actual RGB levels versus the reference values, look at the Tables page (See Figure 19 below).
6. The T values are the theoretical (reference) values, the A values are the actual values. Ideally, the reference and actual values should match. More realistically, the differences between the reference and actual values for the three color component should match.

Figure 19: Color-Chart Overlay: Table Page

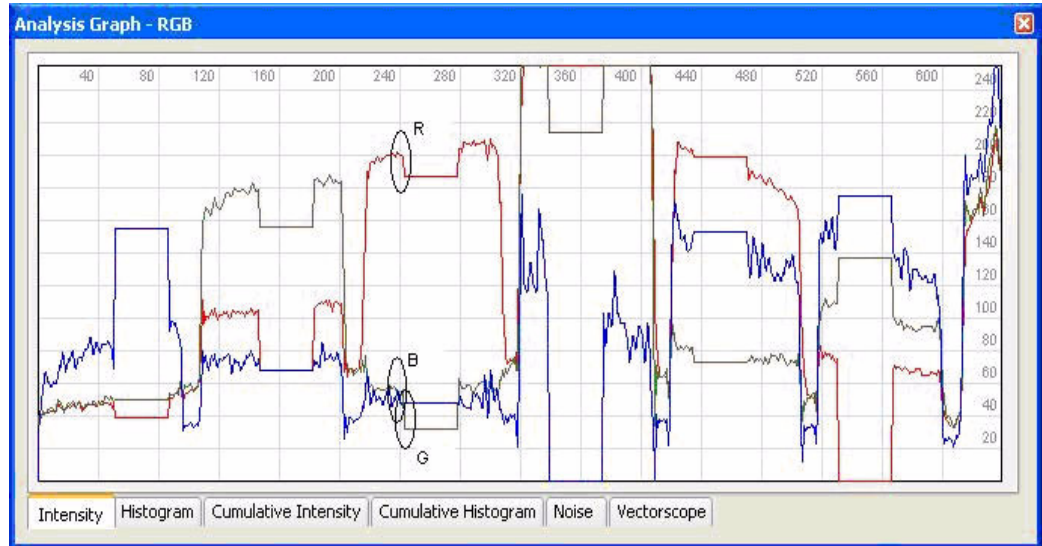


	T red	T green	T blue	A red	A green	A blue	Clip
1:Dark Skin	116	78	62	109	84	0	Y
2:Light Skin	207	154	130	226	183	201	N
3:Blue Sky	92	122	164	141	148	207	N
4:Foliage	84	108	60	121	148	23	Y
5:Blue Flower	133	128	188	157	142	211	N
6:Bluish Green	100	203	179	144	189	208	N
7:Orange	234	124	0	225	156	73	Y
8:Purplish Blue	68	84	176	109	109	210	N
9:Moderate Red	208	78	94	243	101	138	N
10:Purple	92	48	108	112	83	94	Y
11:Yellow Greer	167	202	48	208	242	143	N
12:Orange Yello	247	168	0	230	188	91	Y
13:Blue	39	50	155	48	50	134	N
14:Green	68	156	68	134	205	89	Y
15:Red	187	32	48	216	43	20	Y
16:Yellow	255	214	0	253	254	233	Y
17:Magenta	199	73	153	210	82	185	N
18:Cyan	0	137	175	75	127	186	N
19:White	255	255	255	252	252	252	Y
20:Neutral 8	214	214	213	243	244	247	Y
21:Neutral 6.5	167	167	167	207	207	222	N
22:Neutral 5	123	123	123	151	157	143	N

Using Analysis Graph - RGB

1. Click the **Graphs** button to bring up the Analysis Graph - RGB window.

Figure 20: Analysis Graph of R, G, B Macbeth Chart Squares



2. Click on the Intensity tab.
3. In the Info dialog box, click on **Mouse Selection>Row** to display a horizontal line on the image. Position the line through the Macbeth chart's third row (blue, green, red, ... cyan).
4. The goal is to have the similar relative values for R, G, B (that is, the offset from reference to actual is the same for R,G, B values). In Figure 20, the reference offset is the flat R, G, or B line. The actual value is the distance between reference and the high/low measurement. For the third square, the R offset (that is, distance from reference to actual) is larger than the B offset and smaller than the G offset.
5. Ideally, the actual R,G,B values should match the reference values but this is not always possible.
6. Using the **Sensor Control --> White Balance --> Matrix Buddy** page, adjust the matrix for each color component (R,G,B) of each square (R,G,B and others). Click the **Put Matrices to Sensor** button to write new values to the sensor.
7. Concentrate only on the red, green, and blue squares.
8. Monitor register R0x112 for settled values before doing another adjustment.
9. Once the CCM values for Incandescent are determined, save the current matrix values (that is, **Put Matrices to Sensor**). Save the CCM to an INI file by going to the **White Balance -->** page and clicking the **Save As** button.
10. Continue the CCM tuning by switching the light source to Daylight. Switch R0x124 to 0x7F7F and to switch the Matrix to Left. Perform the same procedure using Daylight. Then view results with Gamma set to H/W. Continue tuning for final results.

Final Results

To check the results:

1. Set R0x124 to 0x7F00 (to allow full AWB adjustment).
2. Enable Digital Gains (R0x148[7] = 0).
3. Enable H/W Gamma.
4. Load the resulting CCM matrix and AWB settings (CCM and AWB settings are automatically saved when the **Save As** button is clicked). Use the **Presets** dialog box to load the saved settings.
5. Test the settings using Incandescent, CWF, TL84, and Daylight light sources. Check the Macbeth chart versus the reference overlay using the Color-Chart Overlay and the Analysis Graph.

Fast CCM

Fast CCM improves the response time for AWB and CCM to settle. It reduces the number of steps from 128 to 64.

To create Fast CCM matrices:

1. Load the Base and Delta matrices that were derived from the CCM procedure above.
2. Go to the **Sensor Control --> White Balance --> Adjust Range** page.
3. Click the **Load from Sensor** buttons for both left and right matrices.
4. Set the AWB position variables (this will center the matrix offsets):
 - 4a. Change 0x00 to 0x20
 - 4b. Change 0x7F to 0x5F
5. Click the **Recompute Matrices** button
6. Click the **Put Matrices to Sensor** button

Save the resulting register settings using the Sensor Control --> White Balance --> **Save As** button.

White Balance Loading Procedure

1. The matrix interpolator must experience at least one EOF update cycle in manual white balance mode. This loads the base and delta matrices into the matrix interpolator, along with the base and delta gain ratios.
2. The matrix interpolator must actively produce a new set of outputs, which ensures that the AE unit writes new gain settings to the sensor during EOF processing. Any of the following triggers the sensor update:
 - 2a. AWB moves the matrix.
 - 2b. The interpolator switches into, or out of, manual white balance mode.
 - 2c. The interpolator switches into, or out of, flash matrix mode.
 - 2d. The load CCM bit is set $R0x106[11] = 1$.

The following sequence satisfies these requirements for AWB mode:

1. Load new matrices into the CCM registers.
2. Switch to manual white balance mode $R0x106[15] = 1$.
3. Wait for one frame.
4. Switch off manual white balance mode $R0x106[15] = 0$.

This sequence works for manual white balance mode:

1. Load new matrices into the CCM registers.
2. Turn off manual white balance $R0x106[15] = 0$.
3. Wait for one frame.
4. Switch on manual white balance $R0x106[15] = 1$.
5. Another option is enabling the load CCM bit for at least one frame.

Lens Shading Correction

The MT9V131 incorporates a lens-shading correction function (LC). Ideally, a camera must have a uniform sensor response, producing a uniform image of a uniform scene. However, inexpensive lenses make an image darker near its edges. Other factors affecting sensor response include:

- **Microlens positions:** The angle of incidence of light hitting the microlens varies with the distance of the microlens from the optical center of the sensor. A different angle of incidence causes a different area below the microlens to be illuminated. To address this effect, microlenses are shifted. They are not uniformly positioned over pixels.
- **Filter characteristics:** An IR cut filter (used to eliminate unwanted infrared light) can be angle dependent.
- **Lens position:** The lens needs to be positioned at the optical center of the sensor.

In addition to lenses, a variety of factors can contribute to smooth, location-dependent variation in sensor response. This document refers to all such shading, regardless of origin, as lens shading. Lens shading is different in each color channel (R, G, and B), producing coloration shifts from one region of the image to another. By performing LC, a more uniform signal response across the image is obtained by compensating each channel.

Enable LC with R0x106[10]. The LC registers are from R0x180 through R0x195. These registers must be programmed so that each color channel delivers a flat, uniform response throughout the image. The settings of the LC registers are called LC curves or LC parameters. They must be calibrated for each channel.

When the sensor displays a uniformly illuminated gray card, white card, or other uniform target of constant color and brightness, a given channel is calibrated to maximize satisfaction of the calibration criterion listed in the next section.

Calibration Criterion

Every pixel must have value P , where P is the maximum value of the uncorrected pixels in the channel. This requirement can be split into two component subcriteria:

- **Criterion I:** All pixels in a given channel must have the same value.
- **Criterion II:** In a given channel, the same maximum value is observed, whether LC is on or off, that is, the maximum corrected value is the same as the maximum uncorrected value.

If the Calibration Criterion is satisfied, so are Criteria I and II. If Criteria I and II are met, so is the Calibration Criterion.

When judging whether these criteria are violated, ignore violations due to noise.

See the “Procedure for Calibrating LC” on page 54 to find instructions for creating LC curves meeting these criteria as nearly as possible.

Procedure for Calibrating LC

For a specific module, lens, and filter, calibrating LC consists of finding LC curves satisfying criteria I and II, for a column and a row passing through the optical center. The correction for other pixels is automatically derived from the corrections for this column and this row, where LC satisfies criteria I and II best.

The procedure below is a guide through this process. Customers must develop lens shading parameters appropriate to their own modules and/or applications.

Note: Highlighted words refer to icons or data from the DevWare GUI.

Begin

1. To begin, reset the sensor, and load **DevWare**. Once **DevWare** is running perform the following steps.
2. This ensures the sensor is at it's highest resolution (640 x 480). This can be verified by checking the **Properties/Values** columns (**Width** and **Height**) on the left side of the main screen.

Setup

For each combination of lens, lens position, module, IR cut filter, mirror columns, mirror rows, and so on, to be used, a separate set of LC curves must be developed. Be sure the lens is centered over the optical center of the die.

1. Point the module at a uniformly pigmented, uniformly illuminated, flat, matte surface of uniform reflectance. A tint is acceptable, but not so much of a tint that R, G, or B contain excessive noise. Best results are achieved with white or light gray. If possible, use equipment specifically designed to provide a uniform calibration target. Otherwise, verify uniformity by throwing the lens as far out of focus as possible, and moving the module up, down, right and left, and rotating it, looking for any variations in the image.

Note: Once calibration is achieved, the module, light source, and target must not be moved throughout the rest of the procedure.

2. Use Notepad (or some other simple word-processing tool) to create a starting **.ini** file, and save it with a name corresponding to the module being calibrated, the date, name, and so on. This is where the defined LC curves are to be saved.

Presets

To open the Preset window, click the **Preset** button at the top of the screen. In the window:

1. Scroll down through the list and select **Lens Correction: Calibration-Set Up**. Click **Load**. Pause while the file is loaded, then click **Load** again. This may darken the image, which is acceptable.
2. Load **Lens Correction: Calibration-Flat Curves**.
3. Click on **Options** and ensure that only **All Writable Registers** is selected. Deselect all other radio buttons and close the **Options** dialog box.
4. Click **Browse**, and browse to the starting **.ini** file. Open it, click **New**, and enter a name for the section where the curves will be saved. Click **OK** and then **Save**.

While developing curves, periodically open the **.ini** file, highlight the new section and click **Save**. This saves the curves to allow backtracking on adjustments. The LC curves will be registers from R0x180 through R0x195 and from R0x1B6 through R0x1C4.

When the curves are saved, all writable register values will be saved, not just the LC curves. Take great care that a needed section is not overwritten. If a section is highlighted for another module, that section will be overwritten.

Begin Vertical Calibration

1. Click the **control** icon at the top of the screen to open the Sensor Control window.
2. In the window, click on **Lens Correction**.
3. In the **Lens Correction** window, select **Enable Lens Correction**, set Scale = 3 Low, and set Corners = 0 Off.
This sets $R0x180 = 3$. **Scale** is kd or Kx . **Corners** is kp or Kxy . See the MT9V131 data sheet for details on register descriptions.
4. If another scale setting is to be used, restart the LC calibration procedure from the beginning.
5. Click on the **graphs** icon at the top of the main screen (the Analysis Graph window will open).
6. Select the **Intensity** tab on the Analysis Graph window.
7. Click on the **info** icon at the top of the main screen.
8. At the bottom left of the main screen, under **Mouse selection**, select **Column**, then move the cursor to column 320 (half of the snapshot horizontal resolution) in the image and click on it. The cursor position can be read from the **Property/Value** area of the main screen.
The values for column 320 are now plotted in the Analysis Graph.
9. Select one color channel—R, G, or B—to calibrate. The other channels must be deselected. Selecting and deselecting is done by clicking in the appropriate **Red**, **Green**, or **Blue** box in the Lens Correction/Sensor Control window.
10. Select **Vertical** to complete vertical calibration for that channel before doing anything with another channel.
11. Click the **Auto-Adjust** box and watch the sliders for the knees change and the display in the Analysis Graph flatten out across the window.
12. Repeat step 9 through step 11 for each of the other channels.

Note: Do not select **Horizontal** until all channels have been calibrated under **Vertical**.

13. If only the selected channel to be visible in the Analysis Graph is desired, use **Data Interpretation** in the Sensor Control window.

Vertical Calibration—Initial Value

1. Deselect **Enable Lens Correction** on the Lens Correction window.
2. On the Analysis Graph, note the selected channel's peak value when LC is turned off. Call this value P .
3. In this section, set the Initial Value for the selected channel so that with LC on, the first pixel in the column has value P .
4. Right-click on the Analysis Graph and click **Lens Correction Guides**.
A vertical guide line now indicates the location of each knee. Knees are the adjustable points on the selected line profile.
5. Turn off LC. Modify $R0x009$ so the channel is very high on the graph, and maximize the vertical height of the Analysis Graph window to magnify the plot.
6. Inside the Analysis Graph window, click on P , the peak value of the channel being calibrated, to make the horizontal indicator line pass through the highest point on the plot.
For this step only, use the **Cumulative Intensity** tab on the Analysis Graph. Visualize where the peak of the plot would be if the plot were smoothed; use this value as P .



7. Select **Enable Lens Correction**, then use the **Initial** slider and up and down buttons to match the value of the first pixel to the indicator level P , without going over. The arrow keys on the keyboard can also be used to change the value.

Caution Do not proceed until this is complete; later, it may not be possible to change this value without extensive rework.

Vertical Calibration—Knee Values

In this section, to the extent possible (and ignoring noise), give the entire column the same value as the first pixel. Visualize a smoothed plot. To make the plot flat, bend or tilt it at several points. These points are labeled on the Analysis Graph as *Knee 0*, *Knee 1*, and so on.

Using the Intensity tab on the Analysis Graph is a faster approach. Once the plot is nearly flat, use Cumulative Intensity for fine tuning.

1. First, make the plot flat at *Knee 0*.
To tilt the plot up where a knee line crosses it, raise the slider for that knee. To tilt the plot down, lower the slider for that knee. For fine adjustments, use the up and down buttons ( and ). It may be necessary to click several times to arrive at a value that induces the plot to tilt. Tilt to flatten the plot between *Knee 0* and *Knee 1*.
2. Proceed from one knee to the next, adjusting each knee value to flatten the plot at that knee, the next knee, the previous knee, and between that knee and the next knee. To be sure the plot cannot be flattened further, it is necessary to tilt further and further, until the tilt has gone too far in the wrong direction, and then back up. The last pixels in an interval between knees are sometimes the last to fall into line when the tilt is adjusted.
3. Periodically backtrack to earlier knees to see if further flattening is possible. For example, after adjusting *Knee 2*, readjust *Knee 0* and *Knee 1* if further flattening is possible, to improve the Initial Value setting.
Backtracking can be useful when the plot between two knees is too low. If adjusting these knees can only make the plot between them too high, try adjusting one or more previous knees to push the plot lower. Now, the adjustments that earlier would only push the plot too high may no longer do so.
If two successive knees have the same value, changing the value of one knee by one may cause a large, undesirable tilt. With no change, the tilt may also be undesirable—in the other direction—with no intermediate flatter tilt available.

Note: It may be necessary to go back several knees and make slight changes in previously set knees, so that the two knees that were of equal value no longer have equal values. By making these adjustments, intermediate tilts can become available.

As the plot is flattened, visualize that it is smoothed (with no noise) and prevent it from exceeding P and the horizontal indicator line.

To ensure criterion II is met, periodically toggle **Enable Lens Correction**. Adjust knee values as necessary so that the plot achieves the same maximum whether LC is on or off. Imagine that the plot is smoothed regardless of whether LC is on or off. The smoothed plot should have the same maximum in either case. When finished, click the **Histogram** tab on the Analysis Graph. Ensure that the same global maximum is obtained whether LC is on or off; the peak of the histogram should occur at the same value.

Be sure that the value at the center pixel is no lower with LC on than it is with LC off. If it must be either higher or lower with LC on, chose higher.

When LC is off, the plot must peak at the center; no other pixel can be higher. If another pixel must be either higher or lower, choose lower.

To make the plot as flat as possible, backtrack, experimenting with every knee. If no knee can be tilted further up or down to produce a flatter plot, then the plot is flat enough.

When criteria I and II are satisfied to the extent possible, the channel has been vertically calibrated. Save the curves. Select another channel with the radio buttons. Be sure to select only one channel at a time. Turn off the radio buttons for the other channels. Perform vertical calibration for the newly selected color channel, then repeat until the R, G, and B channels have all been vertically calibrated.

Horizontal Calibration

After completing all vertical calibrations, perform horizontal calibration in a similar manner, using row 256, and selecting the **Horizontal** radio button. Work with only one channel at a time until the R, G, and B channels have all been horizontally calibrated. Once horizontal calibration has begun, you cannot return to vertical calibration unless **Lens Correction: Calibration-Flat Horizontal Curves** is loaded or unless the horizontal curves have already been developed to the point that the same peak value is obtained whether LC is on or off.

After finishing vertical calibration, begin horizontal calibration again from the beginning, or reload the saved horizontal curves. Be sure to save the horizontal calibrations before loading the flat horizontal curves or they will be lost.

Save the LC curves. Reset. Load the settings defining normal use, not the calibration settings. Load the saved LC curves. If any regions of color shifts exist, note exactly where they begin horizontally and/or vertically. Find the corresponding locations on the plots for the center row and column. Departures from flatness in the plot should be visible. Correct these starting calibration again and loading the saved curves. Adjust the curves. Shifts in corners or near edges may not correspond to non-flat areas in the plots.

Corners

After completing all of the above procedures, save the LC curves. Reset. Load the settings defining normal use, not the calibration settings. Load the saved LC curves. Check to see if the corner areas are too dark.

If desired, adjust **Corners** kp or Kxy in R0x180 to increase values near image edges and corners. This affects all channels. Setting **Corners** too high can cause visible banding.

Excessive Fall Off

If the plot cannot be brought up to the level of P , decrement **Scale** by one and begin the LC calibration procedure from the beginning.

Cleaning Up the File

After the final results have been saved, use Notepad to edit the saved file. Remove all settings except the LC registers. This file can now be used to load the LC curves without disturbing other register settings.

The result is a file with the necessary parameters to program LC for the module.

Gamma Correction

Selecting a Gamma Factor

While the following is not intended to be a complete reference on the subject of gamma correction, it provides helpful background information.

Gamma correction in a display system compensates for power transfer functions, such as those that occur when a CRT converts signal voltage into output intensity. Assuming a signal range of 0.0–1.0, the transfer function of a color channel is:

$$C_{out} = (C_{in})^y \quad (\text{EQ 3})$$

If the gamma function is applied several times with different factors, the net effect on the system (precision issues notwithstanding) is as though a single gamma function is applied with a gamma that is the product of the different factors:

$$C_{out} = \left(\left(\left(\left(C_{in} \right)^{y_1} \right)^{y_2} \right)^{y_3} \right)^{y_4} = (C_{in})^{y_1 y_2 y_3 y_4} \quad (\text{EQ 4})$$

Consider the following simple example: An image captured by a camera with output gamma correction capability is displayed on a CRT with the standard 2.2 gamma transfer function. The goal is to display a faithful rendition of the image on the CRT.

To reproduce the image correctly, the camera must pre-correct the image so that its colors are displayed correctly on the monitor. This requires selecting a gamma factor for the gamma correction function of the camera that counteracts the effects of the CRT gamma correction. For the output image to be the same as the input image, the product of the gamma of the camera and the gamma of the CRT must be 1.0. This in turn requires that the gamma factor of the camera be set to approximately 0.45 (1/2.2).

Gamma correction factors can occur in a variety of places:

- CRT electronics correct gamma for colors. Standard gamma for a CRT is 2.2, although larger gammas can be used. The PAL TV standard uses a standard gamma of 2.8, and gamma correction ranging from 2.35 to 2.55 may be appropriate for some CRTs.
- The graphics hardware in a system can apply a gamma correction curve.
- The standards for representing images or video can require specific gamma correction. For example, Rec. ITU-R BT.709 requires image data pre-correction with a gamma of 0.45.
- Additional gamma factors can be applied to make images more pleasing. For example, the 2.2 gamma of the NTSC TV standard is lower than the 2.35 to 2.55 gamma of actual monitors. This improves viewing in dark environments. The same data viewed in a bright environment may require an additional gamma correction of 1.1 to 1.2.
- The camera itself may apply a gamma factor.

Before selecting the gamma factor to be applied in the MT9V131 gamma correction unit, consider the role of the sensor in the image processing chain. Two gamma tables are provided in the MT9V131 to support two different roles. For example, one gamma table is used when previewing images on a cell phone display, the other is used to comply with an encoding standard when taking snapshots.

The MT9V131 default gamma tables correspond to a gamma of approximately 0.6—a setting demonstrated to produce perceptually pleasing images when the graphics hardware performs pre-correction for the display device.

Gamma Tables—Beyond Gamma Correction

The implementation of gamma correction in the MT9V131 is as a piecewise linear approximation using 11 points at specific, predetermined locations. These points work well for gamma functions. For any 10-bit input luma, an output luma will be interpolated between the two nearest points in the table.

The default tables assume a gamma factor of 0.6. They also assume that the largest input value will be 896 (4×224). By the time 10-bit lumens in the 0–896 range are output in YCbCr format, their value has increased to around 16,235. Data in RGB format is scaled up to provide full range color in each component.

The table elements themselves are simply 8-bit luma values. Any desired function in the gamma tables must be monotonically increasing. The output of the gamma correction unit is clamped to the value of the 11th knee point in the gamma table (Y10). Lowering Y10 below the other knee points is not recommended.

The sensor also incorporates an effects unit capable of implementing several algorithms that might otherwise be implemented with nonstandard gamma tables.

Formulas Used for Internal YCbCr Calculation

$$Y = (0.299R + 0.587G + 0.114B) + 16 \quad (\text{EQ 5})$$

$$Cr = 0.713(R - (Y - 16)) + 128 \quad (\text{EQ 6})$$

$$Cb = 0.564(B - (Y - 16)) + 128 \quad (\text{EQ 7})$$

The +16 is added to Y through the luma offset register, R0x134[15:8].

Calculating Gamma Table

By default, the MT9V131's YCbCr output is configured to comply with Rec. ITU-R BT.656 standards (formerly known as CCIR 656). If the output is used for JPEG compression, the SOC output should be reconfigured to use the full range of Y.

Use the following formula to calculate a gamma table:

$$Y_i = Y_{max} \left(\frac{X_i}{896} \right)^y \quad (\text{EQ 8})$$

where

$$X_{0...10} = \{0, 16, 32, 64, 128, 256, 384, 512, 640, 768, 896\} \quad (\text{EQ 9})$$

$$Y_{max} = \begin{cases} 240 & \text{ITU601/656YCbCr or RGB output} \\ 255 & \text{JPEG-friendly YCbCr output} \end{cases} \quad (\text{EQ 10})$$

A typical value of y is $y = 0.45$, which is $1/2.2$ (the reciprocal of a standard monitor gamma). The default uses a gamma of 0.6.

Calculated values are programmed into R0x153–R0x157.

The luma offset register R0x134 and the luma clip register R0x135 must remain consistent with the outputs from the gamma table. The luma offset value must be appropriate for the output format, and the luma clip register must be set to pass all the legal luma values for that output format. Table 11 illustrates appropriate values for these registers.

Table 11: Output Register Values

Output Format	Luma Offset R0x134	Luma Clip R0x135
RGB	[7:0] = 0	n/a
ITU-601/656 YCbCr	[15:8] = 16	0xF010
JPEG-friendly YCbCr	[15:8] = 0	0xFF01
Processed Bayer	[7:0] = 0	n/a
Bayer bypass	n/a	n/a

For convenience, DevWare can calculate the appropriate register settings on the gamma table page. Simply choose ITU Rec. BT-601 or JPEG-friendly forms, and enter the desired gamma factor.

Gamma Tables and Luma Clip Range and Offset

The luma clip range and luma offset depend on the range of the gamma table output, so separate luma clip and offset values are needed for each gamma table context.

In a situation where YCbCr output is desired from both contexts with different gamma tables, there are two possible options:

1. Adjust the gamma table values to incorporate different offsets. For example, set the luma offset value to “0” and add 16 to each entry in the ITU-BT. Rec. 601/656 gamma table.
2. Reprogram the luma clip range and luma offset registers just prior to taking the snapshot.



Appendix A: Imaging Developer Guide FAQs

Optimal Settings for the MT9V131

Please find the optimal settings in the latest MT9V131 image sensor data sheet. Also, start DevWare, click on “Presets” toolbar button and select defaults to load from the list of presets.

Interpreting the Design ID

The design ID can be read out in core R0x0FF.

Bits 15-12:	internal project ID number, 1st character
Bits 11-8:	internal project ID number, 2nd character
Bits 7-4:	analog core revision
Bit 3:	1=SOC, 0=standalone
Bits 2-0:	digital block revision

For example, the MT9V131 has the design ID: 0x823A. “82” is the internal project ID number; the analog revision is “3”. “A” means this is an SOC (bit 3 is set) and the digital block revision is 2.

Enable Raw Bypass Mode

The MT9V131 can output raw data (Bayer pattern) directly from the sensor. Two modes are available:

1. 10-bit bypass
2. 8+2 bypass

In 10-bit bypass mode the MT9V131 outputs 10-bit data just like it is the MT9V011. However, this mode is not available in the PLCC package as it requires additional pad bonding. To enable this mode set IFP R0x108[10] = 1.

In the 8+2 mode the MT9V131 outputs 10-bit raw data in the PLCC package. The 10 bits are output as 2 bytes—the first byte containing bits R0x108[9:2]; the second byte is “0 0 0 0 D1 D0”. To enable this mode set IFP R0x108[7] = 1.

Note: The values in the outgoing defect specification of the MT9V131 are different from that of the MT9V011.

Default Mode for the MT9V131

The default power-on mode is VGA YCbCr 15 frames per second (fps) using a 12 MHz input clock.

Speed Up Auto Exposure

Use IFP R0x12F to speed up auto exposure, for example, R0x12F = 0x84. For the fastest auto exposure set R0x12F = 0x80 and IFP R0x13F = 1.

Note: A fast AE may overshoot and oscillate on approach.

In dark conditions the frame rate can drop which naturally slows down auto exposure. To disable frame rate slowing down auto exposure program IFP R0x137.

Speed up AWB

Use IFP R0x125[2:0] and R0x125[6:3] to speed up auto white balance.

Note: Faster adjustment may result in the white balance oscillating.
 Example: R0x125 = 0x4514.

When to Speed up AE and AWB

Generally, you need to speed up AE and AWB for digital still photography. The preview mode for digital stills should adjust very quickly. It may even be acceptable that AWB and AE oscillate slightly as they try to settle as fast as possible.

For video capture keep AWB and AE slower because it may be undesirable for image brightness and color to change because of slight scene changes, such as people walking by or the camera panning over a room.

IFP R0x12A = 0 configures the SOC to make picture gray at all times. If this is insufficient, try IFP R0x11F = 0 and R0x120 = 0xC800. Switch AWB to “gray world” mode, see “Auto White Balance (AWB)” on page 37.

How to Program the MT9V131 to Work off an Oscillator with a Given Frequency

The MT9V131 can work with master clock frequencies from 12 to 27 MHz:

- If EXTCLK is close to 27 MHz, set core R0x007[4]=0.
- Use Aptina software to calculate horizontal and vertical blanking for the sensor core to output desired fps—core R0x005 and R0x006.
- Set shutter width numbers in IFP from R0x159 through R0x160.
- Set line length in IFP R0x139.
- Use the Aptina Imaging software to calculate the SOC flicker avoidance parameters, IFP from R0x15C through R0x15D.

Contact your local Aptina Imaging FAE to obtain a copy of this software.

When to Calibrate CCM and/or Lens Correction

The color correction matrix and lens correction must be calibrated for each combination of lens and infrared-cut filter. Each camera module manufacture needs its own set of optimized defaults.

How to Calibrate CCM

The following procedure is recommended: ask a Aptina Imaging FAE to supply you with the setting for the module of interest and supply the FAE with several modules.

How to Calibrate Lens Correction

Image a uniformly illuminated surface. Adjust the lens shading correction registers IFP from R0x180 through R0x195 to obtain a flat image. The color correction matrix must be disabled during capture.

In DevWare, click on “Controls” button, select the “Lens Shading” tab and check “Enable lens shading”. Now you can adjust lens shading using the curve tool. To have DevWare calibrate lens shading for you automatically, point the camera at a uniformly illuminated surface and click “Detect Lens Shading” button. DevWare will adjust registers to get a flat image. Registers from R0x180 through R0x195 will contain settings you are looking for.

How to Fire Flash

The MT9V131 supports both xenon and LED light sources using a dedicated output signal. For xenon flashes the signal generates a strobe to fire when the imager’s shutter is fully open. For an LED flash the signal can be asserted or de-asserted asynchronously. To turn the LED off and on program R0x198[8]. To fire a xenon flash, arm the strobe trigger by setting R0x198[13] = 1. The strobe will appear when the shutter fully opens. Strobe length is set by R0x198[7:0]. Other available modes include continuous versus single firing and skipping a programmable number of frames after arming and before firing.

How to Enable Built-in Test Patterns

Set IFP R0x148[7] = 1 and R0x148[2:0] = 1..7 to enable test pattern at the IFP input. R0x148 test patterns include color bars.

IFP R0x13A generates another set of test patterns at the IFP output. Set IFP R0x13A[5:3] to obtain ramps at output. You may want to selectively disable color or luma channels using IFP R108[4:2].

How to Zoom and Pan Smoothly

Decimation control registers work to downsize output image to any size. The output image size is specified in IFP R0x1A7 and R0x1AA for horizontal and vertical directions respectively. For example, to downsize the VGA output to QQVGA set R0x1A7 = 0xA0 and R0x1AA = 0x78. Whenever output image is downsized, the zoom feature becomes available. To zoom in, program R0x1A6 and R0x1A9 with the size of window to be decimated. For example, in QQVGA setting R0x1A6 = 0x140 and R0x1A9 = 0x0F0 results in 2x zoom. Here the output image of 160 x 120 is created from a pre-decimation window of 320 x 240 instead of the full VGA 640 x 480. Whenever the output image is zoomed, pan controls become available. To pan a zoomed image program R0x1A5 and R0x1A8 to offset the pre-decimation window in to the right and bottom respectively. When implementing a smooth zoom and pan, it is useful to synchronize the update of all decimation registers to avoid jerks in the output video. When writing a batch of decimation settings, set bit 15 of each datum to “1” to freeze the update. Set bit 15 of the last datum in the batch to “0” to enable normal operation. The entire batch of decimation settings will then be synchronously loaded on the next frame start.

How Fast does Two-Wire Serial Interface Run?

In a dual sensor application the SOC two-wire serial interface runs slower than in a standalone sensor. This is due to access to the shared bus. A ballpark number for maximum speed is 500 KHz at 30 fps.

How to Calculate Gamma Table

Use the following formula to calculate a YCbCr gamma table:

$$Y_i = 224 * (X_i/896)^{\gamma} \quad (\text{EQ 11})$$

where $X_{0..10} = \{0, 16, 32, 64, 128, 256, 384, 512, 640, 768, 896\}$.

Here $\gamma = 0.45$ typically. Calculated values are programmed in the IFP from R0x153 through R0x158.

For YUV output set IFP R0x134 = 0x0000, R0x135 = 0xFF00 and use the following formula:

$$Y_i = 255 * (X_i/896)^{\gamma} \quad (\text{EQ 12})$$

where $X_{0..10} = \{0, 16, 32, 64, 128, 256, 384, 512, 640, 768, 896\}$.

Configuring a JPEG-friendly Output

By default the YCbCr output of the MT9V131 is configured to comply with CCIR 601 and 656 standards. CCIR 601 limits Y to range from 16 to 235. However, JPEG accepts Y ranging from 0 to 255. Therefore, if the output is used for JPEG compression the SOC output should be reconfigured to use the full range of Y.

The SOC can be configured to output Y ranging from 0 to 255. First, program the appropriate clipping and offset by setting IFP R0x135 = 0xFF00 and R0x134 = 0x0000. Secondly, disable CCIR codes by setting R0x106[7]=0 (optional). Thirdly, reprogram the gamma table from R0x153 through R0x157 according to this formula:

$$Y_i = 255 * (X_i/896)^{\gamma} \quad (\text{EQ 13})$$

where $X_{0..10} = \{0, 16, 32, 64, 128, 256, 384, 512, 640, 768, 896\}$.

For the MT9V131, set IFP R0x13C[9:0] = 0 and use the formula above.

For convenience, please use the "gamma_calculator.xls" Excel spreadsheet to calculate register settings. This file is available from Aptina Imaging FAEs. Below are several pre-calculated sets for the MT9V131, CCIR codes from the IFP R0x103C[7] = 0 and black read zero IFP R0x13C[9:0] = 0.

Register Values for Gamma 0.6..1 and Output Luma Range of 0...255 for use with JPEG are shown below:

	0.6	0.7	0.8	1.0
R0x153	0x2317	0x190F	0x120A	0x0905
R0x154	0x4734	0x4128	0x361F	0x2412
R0x155	0x9978	0x8D6A	0x815E	0x6D49
R0x156	0xD0B6	0xC9AC	0xC3A3	0xB692
R0x157	0xFFE8	0xFFE5	0xFFE1	0xFFD8
R0x158	0x0000	0x0000	0x0000	0x0000
R0x134	0x0000	0x0000	0x0000	0x0000
R0x135	0xFF00	0xFF00	0xFF00	0xFF00



How to Suppress Aliasing in Decimation

Aliasing in decimation modes can be suppressed by forcing the interpolation to do more averaging: $R0x147 = 0x7F$. Also, you may want to turn off aperture correction.

Note: These settings will result in a picture that is considerably less sharp.

How to Mirror the Image

To mirror the image vertically, set sensor $R0x020[15] = 1$ and IFP $R0x108[1] = 1$. To mirror the image horizontally, set sensor $R0x020[14] = 1$, $R0x108[0] = 1$ and core $R0x02 = 0xF401$.

How to put Camera in Standby

The sequence is:

1. Drive STANDBY pin HIGH.
2. Set core $R0x007[1] = 0$.
3. Stop master clock (optional, but recommended).
4. If peripheral logic has a finite input impedance, tri-state outputs.

Your standby current depends on the voltage of your interface logic. Standby power is smallest when logic connected to the camera uses 2.8V signaling. If your logic runs on a different power supply voltage, for example, 3.3V, there will be pad leakage.

How to Load Defaults in DevWare

Click on the "Presets" button in the toolbar and choose defaults to load.

How to Configure the MT9V131 for Video Capture at Constant 15 fps Rate

Limit shutter width in IFP $R0x137$. For 15 fps, 12 MHz

```
R0x137 = x100 // constant 15 fps
```

How does Noise vs. Frame Rate Trade-off Work

The MT9V131 can slow down its frame rate in the dark. Before it slows down the frame rate the MT9V131 will try to use higher analog gains. This keeps the camera running at full speed, but reduces the signal-to-noise ratio. IFP registers from R0x140 through R0x145 specify a threshold for the analog gain when the MT9V131 will start slowing down. This scheme is similar to choosing the ISO film speed setting on a digital still camera.

The MT9V131 can be configured to either:

1. Keep noise low and start slowing fps at relatively high illumination, or
2. Allow more noise in the picture, but let the camera run at full speed up to a relatively low illumination. For further assistance, contact your local Aptina FAE.

How to Reduce Banding

Banding occurs if:

- There is resistance in the VAAPIX power supply (for example, in a module assembly or power connector)
- There is no decoupling capacitor right next to VAAPIX

If you see banding, please check the module design and put a decoupling cap immediately next to VAAPIX and widen the VAAPIX trace. Banding can be greatly reduced by setting core R0x007[4]=0 when your master clock is ≤ 13.5 MHz. Also, set core R0x02F = 0xF7B6 and IFP R0x139 = 0x122, using R5R6.exe script.

How Soft Standby Compares to Hard Standby

Hard standby (driving STANDBY pin HIGH) powers down analog circuits and turns off clocks. Soft standby only powers down analog circuits—the digital block continues to run. Therefore, in soft standby externally stop the master clock to achieve same standby current as in hard standby.

Where to get Software to Calculate the MT9V131 Timing

This software is available from the Aptina Imaging FAE.

Setting Exact Frame Rate

In the MT9V131 frame rate by default is set at not exactly 15 fps to allow for flicker detection. For reasons concerning a relation between fps and integration time, when the frame rate is set to exactly 15 fps, flicker detection does not work. Therefore, when having an exact frame rate is important, such as for capturing video clips, disable the flicker avoidance (freeze the current detected 50/60Hz frequency) and reprogram timing defaults.

Checking White and Black Levels Experimentally

The correctness of the white and black levels can be checked as follows.

1. Check if output range is 0..0xFF
 - a. In room-like conditions freeze auto exposure and white balance (IFP R0x106[14] = R0x106[1] = 0).
 - b. Turn on more light and check that histogram has 0xFF LSB codes.
 - c. Turn off light and put camera in complete darkness; check that the majority of codes is 0 LSB.
2. Check if levels are not clipped.



- a. In (1c) the histogram should look like a Gaussian distribution with its peak at 0 LSB. Not seeing the peak means the black level is clipped. To see the distribution better, you can let a bit of light into the camera and see the distribution move.
- b. Clipping of the white level can be deduced from register settings.

Gated Output Pixel Clock Mode

The MT9V131 has a special output mode when output pixel clock is gated by LINE_VALID. To enable this mode, set core R0x028[0] = 1. This mode works for both 8-bit and 10-bit output. In 10-bit bypass mode (IFP R0x108[10] = 1) you can capture raw, 10-bit sensor data and adjust output pixel clock phase using core R0x007[15,11:8].

Formulas used for Internal YCbCr Calculation

$$Y = (0.299 * R + 0.587 * G + 0.114 * B) + 16 \quad (\text{EQ 14})$$

$$Cr = (R - (Y - 16)) * 0.713 + 128 \quad (\text{EQ 15})$$

$$Cb = (B - (Y - 16)) * 0.564 + 128 \quad (\text{EQ 16})$$

Creating Special Effects

The MT9V131 supports monochrome output. Set IFP R0x108[5] = 1.

With some extra programming it is possible to produce sepia. Just before capturing a snapshot, reprogram the color correction matrix in IFP as follows:

1. Force AWB digital gains to 1: R0x148[7] = 1.
2. Load sepia CCM:
 - 2a. R0x102 = 0x0024; R0x103 = 0x2914; R0x104 = 0x8E4; R0x109 = 0x0009;
R0x10A = 0x008E; R0x10B = 0x0011; R0x10C = 0x0018; R0x10D = 0x0086;
R0x10E = 0x002B; R0x10F = 0x000A; R0x110 = 0x0093; R0x111 = 0x0011;
R0x115 = 0x0000; R0x116 = 0x0000; R0x117 = 0x0000; R0x118 = 0x0000;
R0x119 = 0x0000; R0x11A = 0x0000; R0x11B = 0x0000; R0x11C = 0x0000;
R0x11D = 0x0000; R0x11E = 0x0000; R0x15E = 0x5050; R0x15F = 0x0000;
R0x160 = 0x0000
3. Toggle R0x106[15] to new CCMs take effect: R0x106[15]=1; R0x106[15]=0

Note: Negative effect cannot be done in the MT9V131.

How to Suppress Row Noise in Dark

Under conditions with very low illumination the MT9V131 exhibits some row noise. In general, the noise is reduced:

1. By enabling saturation reduction in low light earlier.
For example, IFP R0x133 = 0x116E will cut saturation and sharpness to 50% when the lens shading digital gain is >1.
2. By limiting maximum digital gain (the picture will look darker).
For example, IFP R0x167 = 0x1910 enables lens correction ("second") digital gain to go up to 1.5x. By default this gain is allowed to go up to 4x.
3. By allowing fps to drop below 15 fps.
For example, IFP R0x137 = 0x300 (this is the default setting).
4. By increasing gamma.
For example, by increasing the gamma from 0.45 to 0.6.
5. By programming a knee in the gamma curve.
For example, for a gamma of 0.45 for CCIR YCbCr output and a knee can be
R0x153 = 0x2B18; R0x154 = 0x5D44; R0x155 = 0x987F; R0x156 = 0xC0AE;
R0x157 = 0xE0D0; R0x158 = 0x0000; R0x135 = 0xF010.



How to Choose IR-cut Filter

IR-cut filter spectral characteristics are very important, as shown below.

1. IR-cut transmission or infrared at ~1μm can create blooming into dark pixels and blooming in active pixels in very bright conditions (90K Lx).
2. IR-cut cut-off can affect color quality, especially for red colors. We recommend calibrating color matrices for your IR-cut.

Aptina Demo system uses this filter:

<http://www.optics-online.com/IRC.asp/PN=IRC30-15x15>

How to Run the SOC Slower than 5 fps

In low light conditions the MT9V131 can automatically increase integration time beyond one frame, thus decreasing output frame rate. The minimum supported frame rate is 5 fps. The minimum rate is 5 fps regardless of chosen maximum fps—15 fps or 30 fps. The following explains how the MT9V131 can be reconfigured to run below 5 fps.

Two solutions are available for reconfiguring the MT9V131 to run below 5 fps.

- Solution A is simple, but has a small trade-off.
- Solution B is more complicated and requires an external application processor help.

Before describing the solutions, here is an explanation of how the MT9V131 handles integration time. The MT9V131 calculates integration time and writes it into core R9. This number, expressed as lines of integration time, almost always equals to

$$\text{Core R0x009} = \text{IFP R0x159} * k \text{ or}$$

$$\text{Core R0x0099} = \text{IFP R0x160} * k, \text{ where } k=0,1,2\dots24.$$

The SOC programs integration time as an integer multiple of the flicker period, T_{FLICKER} . The flicker period is the period of oscillation of indoor illumination, which is typically 2*60Hz in Europe and Japan and 2*50Hz in USA and Japan. IFP R0x159 specifies T_{FLICKER} in lines of integration for 60Hz environment, while IFP R0x160 contains T_{FLICKER} for 50Hz. The MT9V131 can automatically detect the flicker period in the scene and select IFP R0x159 or RR0x160. In the MT9V131 data sheet one T_{FLICKER} is also called “zone”. Thus, the SOC integration time can be in “zones 0,1,2...24”. The “24” is a hard limit in the SOC. This number limits the smallest fps: maximum integration time is $24 * T_{\text{FLICKER}}$, which works out to 5 fps in the 50Hz environment.

When illumination is too high and the SOC is operating in “zone 0” integration time must be reduced and is not an integer multiple of T_{FLICKER} anymore. Since R0x109 is not an integer multiple of T_{FLICKER} , in indoor conditions in “zone 0” you will see rolling bands due to scene flicker aliasing with the camera frame rate.

Solution A

To allow fps to drop 2x more (to 2.5 fps) double the value in IFP R0x159 and IFP R0x160. To allow fps to drop 4x (to 1.25 fps) quadruple the value in IFP R0x159 and IFP R0x160, and so forth.

The drawback of this solution can be seen in indoor conditions. In indoor conditions having light sources with flicker, the MT9V131 will start showing rolling flicker bands at a certain level of illumination and above, for example 1000Lx. However, if R0x159 and R0x160 are doubled, flicker will start appearing at one-half the illumination, which would be $1000Lx * 0.5 = 500Lx$ in our example.

Solution B

If an external application processor is available to monitor the SOC registers, that application processor can dynamically reconfigure the SOC to allow <5 fps frame rates without the flicker trade-off. The idea is to monitor the MT9V131 digital gains. In low light conditions when the digital gains increase to 2x or more, the application processor must quickly reprogram a number of the SOC registers. In particular, the processor must freeze the auto exposure, AWB and flicker avoidance, double IFP R0x159 and R0x160, double core R0x009 and halve IFP R0x162 containing the current digital gains. Then the processor un-freezes AE, AWB and flicker avoidance. The SOC then can decrease fps further. Please see sample code below. To avoid corrupt frames during programming the processor should discard several frames.

This routine automatically drops down framerate by increasing integration time (in 2 times) in the dark.

```

bool NMode = false;
WORDreg89 = GetIFPR006Register(89); // default values for reg89
WORDreg90 = GetIFPR006Register(90); // default values for reg90
WORD reg6 = GetIFPR006Register(6);
WORD IT;

void NightMode()
{
  // read current digital AE gains register
  WORD reg98 = FMain->GetR006Register(98);
  WORD DGain = reg98 >> 8; // calc. current digital AE gain

  if (NMode)
  {
    if (DGain==16) // turn off Night mode
    {
      IT = GetSensorRegister(9); // read current integration time
      SetIFPRegister(6, (reg6 & ~(1<<14))); // turn AE Off

      // return minimal Integration time step to default
      SetIFPRegister(89, reg89);
      SetIFPRegister(90, reg90);

      // double digital AE gain
      SetIFPRegister(98, DGain*256*2 + (reg98&255));
      SetSensorRegister(9, IT/2); // reduce IT

      SetIFPRegister(6, reg6); // turn AE On
      NMode = false;
    }
  }
  else
  {

```



```
if (DGain>=32) // turn on Night mode
{
    IT = GetSensorRegister(9); // read current integration time
    SetIFPRegister(6, (reg6 & ~(1<<14))); // turn AE Off

    // double minimal Integration time step
    SetIFPRegister(89, reg89*2);
    // to double maximal integration time
    SetIFPRegister(90, reg90*2);
    SetSensorRegister(9, IT*2); // double IT and
    // reduce digital AE gain to compensate it
    SetIFPRegister(98, DGain*256/2 + (reg98&255));

    SetIFPRegister(6, reg6); // turn AE On
    NMode = true;
}
}
```



LPF in the MT9V131

The MT9V131 contains an adaptive filter that suppresses noise in flat, texture-free image areas. IFP R0x147[7:1] = 0 disables this filter.

Color Suppression in Saturated Areas

The MT9V131 decreases color saturation for overexposed pixels. Color suppression is activated when pixel values exceed a threshold. IFP R0x125[10:8] specifies this threshold.

Color Suppression in Low Light Conditions

The MT9V131 automatically decreases sharpness and color saturation in low light conditions. IFP R0x125[14] = 1 enables automatic saturation reduction (default). IFP R0x105[3] = 511 enables automatic decrease in sharpening (default).

IFP R0x133 and R0x13E specify thresholds when saturation and sharpness are decreased. The reduction is performed in 2 steps: by decreasing values to 75% and 50%. The two thresholds can be specified with respect to current analog gain, ADC gain and either of the digital gains. IFP R0x133[7:0] and R0x13E[9:8] specify the 75% setting threshold. IFP R0x133[15:8] and R0x13E[11:10] specify the 50% setting threshold.

Size of Interpolation, Defect Correction and Sharpening Kernels

Interpolation, defect correction and sharpening kernels are up to 5 pixels on the side.

Defect Correction of the MT9V131

The MT9V131 has several defect correction techniques:

- 1D: Hot and dead pixels with known coordinates are corrected
- 2D: Hot and dead pixels are corrected on-the-fly
- 3D: Hot and dead pixel coordinates are determined on-the-fly, stored in a table and corrected

How to Switch Master Clock On-the-Fly

Some applications require special accommodations where master clock speed is halved during still capture. It can be done in this way:

1. Set the sensor up for 24 MHz.
 - a. Program 1 zone to be 2 * Tint.
2. For still capture:
 - a. Freeze AE/AWB/flicker (IFP R0x106 and R0x108).
 - b. HALVE the integration time: core R0x009 = 0.5 * R0x009.
 - c. Switch the clock to 12 MHz; skip a bad frame.
 - d. Capture a frame/save as snapshot.
 - e. Switch the clock back to 24 MHz.
 - f. Restore core R0x009.
 - g. Enable AE/AWB/flicker.
 - h. Drop a bad frame.

How to Reduce EMI

EMI can be reduced by:

- 1.Limiting MCLK slew—put a resistor on the PCB in series with MCLK.
- 2.Limiting output pad slew—put resistors on the PCB in series.
- 3.Shielding the module.

How to Check AWB

- 1.Put a Macbeth chart in the scene, see the illustration below.
- 2.Capture the scene.
- 3.Send this image to the Aptina Imaging FAE.
- 4.When using the Aptina Demo, send both image (captureXXXX.bmp) and register dump (captureXXXX.txt)





How to Operate White Balance Manually

Input and output for color correction is R, G, B data.

$$\begin{array}{l} |R| \quad |c11 \ c12 \ c13| \quad |r| \\ |G| = |c21 \ c22 \ c23| * |g| \\ |B| \quad |c31 \ c32 \ c33| \quad |b| \end{array}$$

To change CCM in manual mode:

- Set digital AWB gains to 1 (IFP R0x148[7] = 1 // That will prevent CCM position from being changed).
- Change base CCM (IFP from R0x102 through R0x104, and from R0x109 through R0x111).
- Change ratios of imager analog gains for “Base” matrix (IFP R0x15E).
- Toggle bit 15 of IFP R0x106[15] to set new values.

How to Achieve Red-Eye Reduction

Red-eye reduction can be achieved by adjusting flash.

1.Xenon—fire xenon twice in a row, if possible.

The eye will adapt after 1st flash.

2.LED—turn on LED for a longer time.

The eye will have time to adapt when LED is on.

There is no hardware red-eye reduction otherwise in the MT9V131.

How to Increase Frame Start Blanking

Reconfigure image size to be 1 line larger ($488 + 1 = 489$) and use “windowing” at the SOC output to crop image back to VGA size.

Appendix B: Glossary of Terms

Table 12: Glossary of Terms

Term	Definition
ADC	Analog-to-digital converter
AWB	Auto white balance
Bayer color filter array	Color space jointly developed by MicroSoft and Hewlett Packard as a color standard. Refer to www.w3.org/Graphics/Color/sRGB .
Camera control sequencer	Automates video clips and snapshots with or without flash
CP	Color pipe; color processing pipeline
CCD	Charge-coupled device: one of the two main types of image sensors used in digital cameras
CCM	Color correction module or color correction matrix
CMOS	Complementary metal oxide semiconductor
D65	An approximation for daylight (D65 is the color produced by a black body radiating at 6,500 Kelvin).
ERS	Electronic rolling shutter
FIFO	First in, first out; refers to a stack
fps	Frames per second
Frame rate	VGA(1,280 x 1,024) – 15 fps at 54 MHz QVGA(640 x 512) – 30 fps at 54 MHz
Gamma characteristic	A gamma characteristic is a power-law relationship that approximates the relationship between the encoded luminance in a television system and the actual desired image brightness. With this nonlinear relationship, equal steps in encoded luminance correspond subjectively to approximately equal steps in brightness.
Gamma correction	The gamma correction is the name of a nonlinear operation used to code and decode luminance or tristimulus values in video or still image systems.
Hysteresis	Hysteresis is a property of the system that does not instantly react to the changes applied to it, but reacts slowly, or does not return completely to its original state.
I/F	Interface
IFP	Image flow processor. Performs color recovery and correction, sharpening, gamma correction, lens shading correction, and on-the-fly defect correction
LC	Lens shading correction
LCD	Liquid crystal display. A type of display, usually numerical, used in electronic equipment. Dark characters are formed on a lighter background. Requires external back-lighting to be visible under low-light conditions. A type of display used especially in small portable electronic devices (digital watches and clocks, cell phones, PDAs, and so on). The type of display found on digital watches and lap-top computers. Their major advantage is low power consumption, leading to long battery life for portable devices.
LED	Light emitting diode
LSB	Least significant bit
Luma	Intensity
Luminance	The quality of being luminous; emitting or reflecting light; “its luminosity is measured relative to that of our sun” [syn: luminosity, brightness, brightness level, luminousness, light]
MSB	Most significant bit
Output resolutions	Includes, but is not limited to, VGA, QVGA, CIF, and QCIF
Pixels	A short name for pixel element. The smallest part that can be displayed on a monitor.
Progressive scan	An image sensor that gathers its data and processes each scan line one after another in sequence. Compare to interlaced.
Resolution	A measure of how much information is stored in an image.
RGB	Red, green, and blue. A way to represent color and a data monitor.
SADDR	Sensor address for two-wire serial interface



Table 12: Glossary of Terms (Continued)

Term	Definition
SCLK	Serial clock for two-wire serial interface
Sensor core	In the MT9V131, it consists of a pixel array of 1316 x 1048 total, an analog readout chain, 10-bit ADC with programmable gain and black offset, and timing and control circuitry.
SOC	System on a chip
SRAM	Static random access memory. SRAM is a type of memory that is faster and more reliable than the more common DRAM (dynamic RAM). The term static is derived from the fact that it doesn't need to be refreshed like DRAM. Because it's expensive, SRAM is often used only as a memory cache.
VGA	Refers to a resolution standard of 640 by 480 pixels.

Contacting Aptina

When contacting Aptina, please provide the following:

- Urgency: customer production stoppage, code freeze tomorrow, need information for purchasing decision, etc.
- Sample images (BMP or PNG lossless compression, not JPEG or PDF with compression)
- Conditions (register dump, VDD, frequency, other settings)
- Failure rate (all parts failing, percentage failing, number of failing parts)
- Reproducible or one time only?
- Revision of chip
 - Provide fuse ID using DevWare (Open Control Panel in DevWare. Select Diagnostics and Fuse ID Reader.)
- Provide sufficient information or material to reproduce the problem at Aptina. If you want us to tune color (LC, CCM), provide:
 - The best three modules (minimum three units; better results can be achieved if more units are provided).
 - Two head boards with sockets for sensor modules that interfaces to DEMO2 board. Any adapter boards used to connect the head board and DEMO2 board must also be included.
 - Schematics of module and head board; this is mandatory—we will not power up the sensor until we verify all supply voltages are within specification. The lens must be centered.



Revision History

Rev. B	9/10
• Updated to Aptina template	
Rev. A	1/08
• Initial release	

10 Eunos Road 8 13-40, Singapore Post Center, Singapore 408600 prodmktg@aptina.com www.apina.com
Aptina, Aptina Imaging, and the Aptina logo are the property of Aptina Imaging Corporation
All other trademarks are the property of their respective owners.
This data sheet contains minimum and maximum limits specified over the power supply and temperature range set forth herein. Although considered final, these specifications are subject to change, as further product development and data characterization sometimes occur.