

---

# 1/3-Inch 1.3Mp CMOS Active-Pixel Digital Image Sensor Developer Guide

## MT9M131

For the latest data sheet, refer to Aptina's web site: [www.apgina.com](http://www.apgina.com)

---

### Introduction

This developer guide provides the information needed for applications where the Aptina's MT9M131 is an integral part of the design. It is intended to supplement the MT9M131 data sheet and is provided exclusively for licensed developers.

### Applicable Part

This document is applicable to the MT9M131, an SXGA-format single-chip camera with a 1/3-inch CMOS active-pixel digital image sensor. This device combines the MT9M011 image sensor core with fourth-generation digital image flow processor technology from Aptina. It captures high-quality color images at SXGA resolution.





## Table of Contents

Introduction . . . . .	1
Applicable Part . . . . .	1
Register Operations . . . . .	2
Features . . . . .	8
Functional Overview . . . . .	9
Internal Architecture . . . . .	10
Two-Wire Serial Interface . . . . .	11
Serial Bus Description . . . . .	11
Protocol . . . . .	11
Maximum SCLK Frequency . . . . .	12
Address Pages . . . . .	12
Page Location . . . . .	12
DeviceID . . . . .	12
Contexts . . . . .	13
Context Selection . . . . .	13
Context Switching . . . . .	15
Preview and Snapshot Modes . . . . .	16
Low Power Preview Mode . . . . .	16
Full Resolution Snapshot Mode . . . . .	16
Default States . . . . .	16
Switching Modes . . . . .	16
Output Format Control . . . . .	17
Camera Control . . . . .	18
Sequencer Operating Modes . . . . .	20
Automatic Mode . . . . .	20
Semiautomatic Mode . . . . .	20
Fully Manual Mode . . . . .	20
Clocks and Blanking . . . . .	21
Primary Operating Modes . . . . .	22
Tuning Frame Rates . . . . .	23
Border Pixels . . . . .	24
Blanking Calculations . . . . .	24
Exposure and Sensor Context Switching . . . . .	25
Switching From Context A to Context B . . . . .	25
Horizontal Blanking . . . . .	25
Switching Frequency . . . . .	25
Simple Snapshots . . . . .	25
Power Saving Modes . . . . .	26
Entering Hard Standby . . . . .	26
Leaving Hard Standby . . . . .	26
Entering Soft Standby . . . . .	27
Leaving Soft Standby . . . . .	27
Output Enable Control . . . . .	27
Avoiding Floating Inputs . . . . .	27
Power Sequencing . . . . .	28
Reset . . . . .	29
Timing Summary . . . . .	29
Resize and Zoom . . . . .	31
Aspect Ratios . . . . .	32
Zoom . . . . .	32
Advantages of Using the Zoom Window Over the Sensor Window . . . . .	32



Manual Zoom .....	33
Automatic Zoom.....	33
Pan.....	34
Snapshot Following Zoom and / or Pan .....	35
Sensor Window 2x Zoom for Low Power 30 fps Viewfinder .....	35
Sharpness .....	37
Aperture Correction.....	37
Auto Sharpen.....	37
2D Defect Correction.....	38
Flicker Avoidance .....	38
Auto Exposure (AE) .....	39
Controlling AE Speed and Sensitivity.....	39
AE Window and Backlight Compensation .....	40
Exposure Step Size .....	40
AE Zones.....	41
Target Luminance .....	45
Black Level .....	45
Exposure Control .....	45
Frame Rate and Day/Night Mode Exposure .....	46
Brightness Control .....	47
Scheme 1: Pre AE, Pre Gamma .....	47
Scheme 2: Post AE, Pre Gamma .....	47
Scheme 3: Post AE, Post AWB, Modify Gamma .....	47
Scheme 4: Post Gamma.....	47
Effects of the Various Brightness Schemes .....	48
Contrast Control.....	48
White Balance .....	49
AWB Overview.....	49
Color Correction Matrices .....	51
White Balance Settings .....	51
AWB Speed and Stability.....	53
AWB Tuning.....	55
Sensor Setup.....	56
Procedure to Adjust Base Sensor Core Gain Ratio .....	57
Using the Color-Chart Overlay - AWB Error (RGB) page .....	58
Using Analysis Graph - RGB.....	59
Final Results .....	60
Color Correction Matrix Tuning .....	60
Initialization .....	60
Procedure.....	60
Using the Color-Chart Overlay - Color Error (RGB) and the Table pages .....	62
Using Analysis Graph - RGB.....	63
Final Results .....	64
Fast CCM.....	64
White Balance Loading Procedure .....	65
Flash Color Correction.....	65
Effects Settings.....	66
Lens Shading Correction .....	67
Procedure for Calibrating LC .....	68
Gamma Correction .....	72
Selecting a Gamma Factor .....	72
Gamma Tables—Beyond Gamma Correction .....	73
Calculating Gamma Table .....	74



---

Gamma Tables and Luma Clip Range and Offset .....	75
Test Pattern Generation .....	76
Appendix A: Imaging Developer Guide FAQs .....	77
Troubleshooting When LC is Turned Off .....	82
Troubleshooting When LC is Turned On .....	82
Miscellaneous FAQs .....	84
Appendix C: Glossary of Terms .....	85
Contacting Aptina .....	87
Appendix B: Typical Window Settings .....	88
Revision History .....	91



## List of Figures

Figure 1:	Register Legend	2
Figure 2:	MT9M131 Functional Block Diagram	9
Figure 3:	Internal Registers Grouping	10
Figure 4:	16-bit MT9M131 Example Device ID READ—Data Returned: 0x143A	11
Figure 5:	16-bit MT9M131 Example Device ID WRITE—Data Written	12
Figure 6:	Nonflash Snapshot Program	18
Figure 7:	LED Flash Snapshot Program	19
Figure 8:	Xenon Flash Snapshot Program	19
Figure 9:	Primary Sensor Core Clock Relationships	21
Figure 10:	Power On/Off Sequence	28
Figure 11:	Hard Standby Operation Waveform	29
Figure 12:	Soft Standby Operation Waveform	30
Figure 13:	Aspect Ratios	32
Figure 14:	AWB Sensor Controls	49
Figure 15:	AWB Flow Processing	50
Figure 16:	Color-Chart Overlay: Color Error (Lab) Page	55
Figure 17:	Color-Chart Overlay: Reference Squares Aligned with Macbeth Chart	56
Figure 18:	Color-Chart Overlay: AWB Error (RGB)	58
Figure 19:	Analysis Graph - RGB Window Showing R,G,B Components of the Macbeth Chart	59
Figure 20:	Matrix Buddy Page	61
Figure 21:	Color-Chart Overlay: Table Page	62
Figure 22:	Analysis Graph of R, G, B Macbeth Chart Squares	63

Table 1:	MT9M131 Two-Wire Interface ID Address Switching	11
Table 2:	Page Access	12
Table 3:	Global Context Control Register Definitions	13
Table 4:	Typical Context B Settings	22
Table 5:	Typical Context A Settings	22
Table 6:	Sensor-Timing-Dependent Registers	23
Table 7:	Blanking Parameter Calculations	24
Table 8:	Output Enable Control	27
Table 9:	Window Setting Restrictions	31
Table 10:	Manual Zoom Settings	33
Table 11:	Settings for Automatic Zoom	33
Table 12:	Pan and Zoom Registers for Snapshots from Preview Field-of-View	35
Table 13:	Sensor Window 2x Zoom Procedure	35
Table 14:	Aperture Correction Settings	37
Table 15:	Classic Interpolation	37
Table 16:	Aperture Scale Values	37
Table 17:	Flicker Avoidance Registers	38
Table 18:	AE Zones	41
Table 19:	Flicker Abatement	44
Table 20:	Exposure Settings Values for the Exposure Target Register R0x22E[7:0]	45
Table 21:	Scheme 4 Brightness Settings	48
Table 22:	Suggested Matrix Positions for Varying Conditions	52
Table 23:	Base and Delta Gain Ratios	60
Table 24:	Effects Settings	66
Table 25:	Output Register Values	74
Table 26:	Glossary of Terms	85
Table 27:	Aspect Ratio, Zoom, Viewfinder, and Snapshot Window Settings	88
Table 28:	Zoom Window Settings for Common Aspect Ratios	88
Table 29:	Manual Zoom Examples	89
Table 30:	Viewfinder Examples (R0x0C8 = 0, Context A)	89
Table 31:	Snapshot Examples (R0x0C8 = 0x90FB, Context B)	90

## Features

The MT9M131 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)
- Color pipeline (CP)

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 MT9M131 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:

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

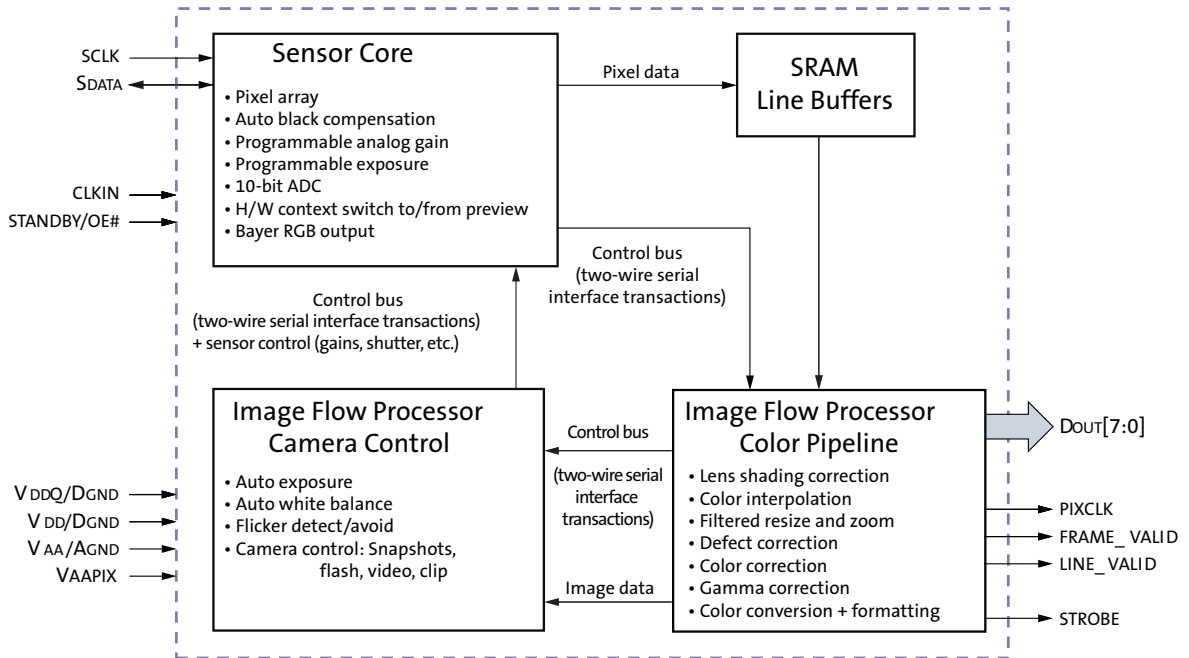


## Functional Overview

The MT9M131 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: MT9M131 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 MT9M131 accepts input clocks up to 54 MHz, delivering up to 15 fps for SXGA resolution images, and up to 30 fps for QSXGA (full field-of-view, sensor pixel skipping) images. The MT9M131 also supports a low-power preview configuration that delivers SXGA images at 7.5 fps and QSXGA images at 30 fps.

The MT9M131 can be programmed to slow the frame rate in low-light conditions to achieve longer exposures and better image quality.

## Internal Architecture

Internally, the MT9M131 consists of a sensor core and an image flow processor (IFP). The IFP is divided in two sections: the color pipeline (CP) and a measurement and control logic block, the camera controller (CC). The sensor core captures raw Bayer encoded images that are then input to the IFP.

The CP section of the IFP processes the incoming stream to create interpolated, color-corrected output, and the CC section controls the sensor core to maintain the desired exposure and color balance and to support snapshot modes.

The sensor core, CP, and CC registers are grouped in three separate address spaces, as shown in Figure 3.

The MT9M131 accelerates mode-switching with hardware-assisted context switching, and supports taking snapshots, flash snapshots, and video clips using a configurable sequencer.

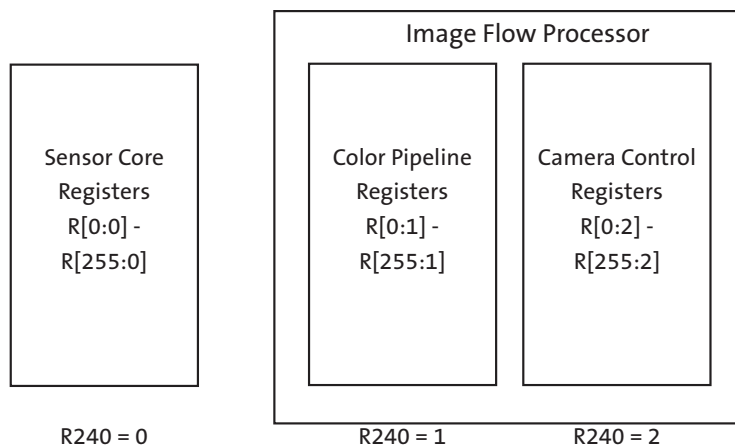
The MT9M131 supports a range of color formats derived from four primary color representations:

- 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 Registers Grouping**



## Two-Wire Serial Interface

### Serial Bus Description

Registers are written to and read from the MT9M131 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 MT9M131 through the serial data (SDATA) line. The SDATA line is typically pulled up to VDDIO 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.

### Protocol

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

- A start bit
- An acknowledge or a no acknowledge bit
- An 8-bit message
- A stop bit
- The slave device 8-bit address

The MT9M131 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: MT9M131 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.

**Figure 4: 16-bit MT9M131 Example Device ID READ—Data Returned: 0x143A**

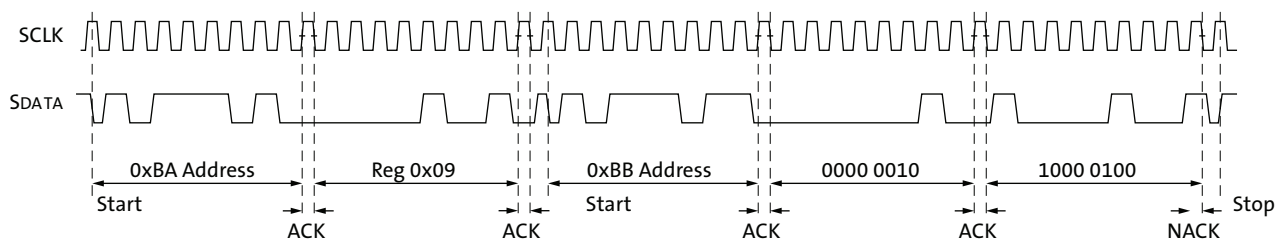
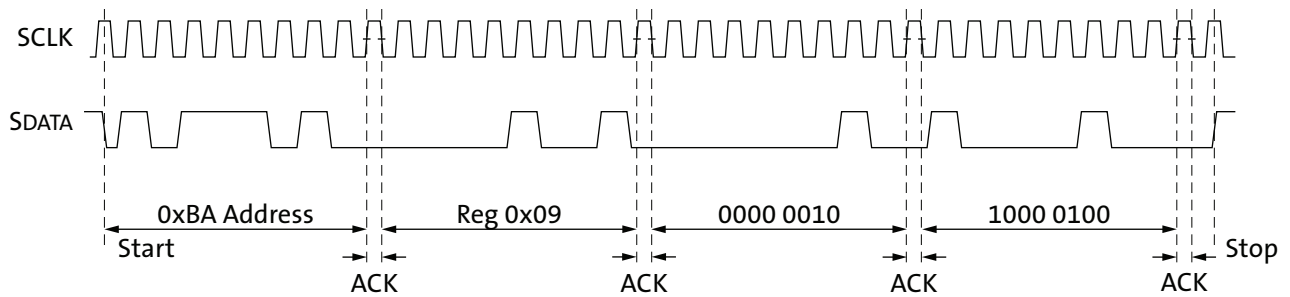


Figure 5: 16-bit MT9M131 Example Device ID WRITE—Data Written



## Maximum SCLK Frequency

The silicon supports  $\text{MAX SCLK} = \text{sensor core master clock}/4$ .

Sensor core master clock =  $\text{EXTCLK}$  (primary input clock)/2.

For example, with the MT9M131 and the recommended 54 MHz MAX clock frequency, and the sensor core master clock = 27 MHz, the MAX SCLK rate (digital theoretical) = 6.75 MHz).

Demonstrated MAX SCLK frequency is 1.8 MHz. For this SCLK frequency, an SDATA pull-up resistor of 910Ω must replace the default 1.5KΩ resistor. This is application-dependent, and a function of the load capacitance on the SDATA signal.

## 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	Colorpipe (base configuration, lens shading, resize, output format)
2	Camera control (auto exposure, AWB, flicker, camera control sequencer)

## 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 R0x0FF.

## Contexts

The MT9M131 supports hardware-accelerated context switching. A number of parameters have two copies of their setup registers; this allows two contexts to be loaded at any given time. These contexts are referred to as context A and context B.

These contexts enable the user to set up the camera for a number of different modes, then switch between modes with a single register write to the Global Context Control Register (GCCR).

A typical example is to use context A for viewfinder/preview settings and context B for snapshots.

Functions supporting context switching include:

- The resizer (output resolutions for preview and snapshot)
- Camera interface (for example, 565RGB for LCD preview and YCbCr for snapshots)
- Sensor core image array readout (for example, low-power QSXGA preview to/from full-power SXGA snapshot)

## Context Selection

Context selection for any single parameter is determined by the GCCR. There are copies of this register in each address page. A WRITE to any GCCR can control the context switching. However, a READ from address page 0 only returns the subset bits of R0x0C8 that are specific to the sensor core. Table 3 provides context definitions.

**Table 3: Global Context Control Register Definitions**

R0x2C8 – Global Context Control (R/W)	
Default	0x0000
Description	Defines sensor and colorpipe context for current frame. Context A is typically used to define preview or viewfinder mode, while Context B is typically used for snapshots. The bits of this register directly control the respective functions. Be careful when writing to this register to avoid a bad frame. “Context Selection” on page 13.
Bit 15	Controls assertion of sensor restart on update of GCCR. This helps ensure that the very next frame is generated with the new context. If sensor restart is NOT requested, then there is some uncertainty as to when the new context takes effect due to the rolling shutter. Typically this bit is always set when R0x2C8 is written. This bit is automatically cleared once the restart has occurred. 0: Do not restart sensor. 1: Restart sensor. Bit 15 can optionally be written “1” when updating the GCCR. This causes a sensor restart. If the GCCR update occurs during vertical blanking, the very next frame is guaranteed to reflect the new context (including any resulting changes to exposure). Bit 15 is automatically cleared by the sensor core when the restart has occurred, and will always read as “0” after restart.
Bit 14	Reserved.
Bit 13	Reserved.
Bit 12	Reserved.
Bit 11	PixClkDivN divide ratio context (R0x1B5). Selects clock divide ratio when using “uniform clocking” mode. 0: Context A. 1: Context B.
Bit 10	Resize/zoom context. Switch resize/zoom contexts: 0: Context A. 1: Context B.

**Table 3: Global Context Control Register Definitions (continued)**

R0x2C8 – Global Context Control (R/W)	
Bit 9	Output format Control 2 context. See R0x13A and R0x19B. 0: Context A. 1: Context B.
Bit 8	Gamma table context. See R0x153–R0x158 and R0x1DC–R0x2FF. 0: Context A. 1: Context B.
Bit 7	Arm xenon flash. 0: Do not toggle STROBE. 1: Toggle STROBE such that flash fires after all rows have been reset but before rows are read out. See R0x023.
Bit 6	Blanking control. This is primarily for use by the internal sequencer when taking automated (for example, flash) snapshots. Setting this bit stops frames from being sent over the BT656 external pixel interface. This is useful for ensuring that the desired frame during a snapshot sequence is the only frame captured by the host. 0: Do not blank frames to host. 1: Blank frames to host.
Bit 5	Reserved.
Bit 4	Reserved.
Bit 3	Sensor read mode context See R0x021 and R0x022. 0: Context A. 1: Context B.
Bit 2	LED flash ON: 0: Turn off LED Flash. 1: Turn on LED Flash.
Bit 1	Vertical blanking context: 0: Context A. 1: Context B.
Bit 0	Horizontal blanking context: 0: Context A. 1: Context B.

- Notes: 1. Not all GCCR bits are implemented in R0x0C8. A WRITE to all bits of R0x0C8 will result in R0x1C8 and R0x2C8 being updated appropriately; a READ from R0x0C8 will only return valid data for the bits implemented (those relevant to sensor core operation).

## Context Switching

There are a number of built-in mechanisms to assist in context switching:

- Automatic and single-step programs, which step through the camera control sequencer.  
The user runs a built-in program that automates the procedure for switching contexts; see R0x2CC for details.
- Semiautomatic mode.  
The user directly controls the contents of this register, but allows a built-in mechanism to time the update of this register to avoid a bad frame. This is achieved by writing the value intended from R0x2C8 through R0x2D3 instead.
- Manual mode.  
The user can write directly to this register. To avoid a bad frame, the WRITE must be timed to be in the vertical blanking time and *after* the end-of-frame (EOF) calculations have completed. See R0x2CA.

### Automatic Mode

The device also includes a sequencer that automates taking snapshots, particularly flash snapshots. A key function of the sequencer is to update the GCCR during each frame for a sequence of frames in the process of adapting and taking the snapshot. This is defined as fully automatic operation.

### Semiautomatic Mode

Using semiautomatic operation, the user updates a shadow register of the GCCR known as the user global context control register (UGCCR, R0x2D3).

After the EOF calculations, the contents of the UGCCR are transferred to the GCCR during vertical blanking. This helps ensure predictable timing following a context change.

### Manual Mode

As stated earlier, one typical usage model is to define context A as preview mode and context B as snapshot mode. The device defaults are configured based on this concept. This mechanism enables the user to have settings for preview and snapshot modes loaded at the same time, and then switch between them with a single WRITE to R0x0C8, R0x1C8, or R0x2C8. This is defined as manual operation.

## Camera Control Sequencer

To facilitate taking snapshots and flash snapshots, the IFP includes a camera control sequencer that automates the process of stepping through a number of preset configurable programs.

In addition to basic snapshots, there are programs for both xenon- and LED-assisted snapshots. A flash-triggering controller provides an appropriate timing strobe for synchronizing the flash illumination with the rolling shutter.

Contact Aptina's local FAE for further information.

## Preview and Snapshot Modes

The sensor core supports two primary readout modes: low power preview and full resolution snapshot.

### Low Power Preview Mode

QSXGA (640 x 512) images are generated at up to 30 fps. The reduced-size images are generated by skipping pixels in the sensor, that is, decimation.

The key sensor registers that define this mode are Read Mode Context A register R0x021 and Read Mode Context B register R0x020.

Only certain bits in these registers are context switchable; any bits that do not have multiple contexts are always defined by their values in R0x020. The active register is defined by the state of R0xnC8[3].

On reset, the global context control register read mode bit R200:n[3] (R0xnC8[3]) is set to "0". This will cause the read mode context A configuration, as defined in R33:0 (R0x021), to be loaded into the sensor.

### Full Resolution Snapshot Mode

SXGA (1280 x 1024) images are generated at up to 15 fps. Full resolution snapshot mode is typically selected by setting R200:n[3] = 1. This will cause the read mode context B configuration, as defined in R32:0 (R0x020), to be loaded into the sensor. This configuration displays an image in full resolution.

### Default States

The default state of R0xnC8 is "0." This selects context A for all device parameters and is suitable for preview or viewfinder operations. Contexts for each parameter are independent.

### Switching Modes

The most direct method of switching to full resolution or snapshot mode is achieved by writing R0x2C8 = 0x870B. This restarts the sensor and configures most functions to context B state, leaving others in context A state. Bit 15 of R0x2C8 must be set to restart the sensor during switching.

Following this WRITE, a READ from R0x1C8 or R0x2C8 results in 0x070B being read. The most significant bit (MSB) is automatically cleared by the sensor.

Following a WRITE of 0x870B to R0x2C8, a READ from R0x0C8 results in 0x000B being read, as only the lower four bits of R0xnC8 and the restart MSB are implemented in the sensor core.

To return to the default or preview mode, write 0x8000 to R0x2C8.



## Output Format Control

In the MT9M131, the default output format is the 4:2:2 YCbCr.

To bypass the IFP and get 10-bit raw Bayer output from the sensor:

1. Set R0x13A[10] for context A
2. Set R0x19B[10] for context B

After setting the sensor into bypass IFP mode, turn off the auto functions in R0x106:

1. Set R0x106[14] -- disable AE
2. Set R0x106[7] -- disable auto flicker control
3. Set R0x106[1] -- disable AWB

Another output format is RBG.

To enable RBG output:

1. Set R0x13A[8] for context A
2. Set R0x19B[8] for context B

Bit 7 and 6 specifies the output format:

- 00: 16-bit 565RGB
- 01: 15-bit 555RGB
- 10: 12-bit 444xRGB
- 11: 12-bit x444RGB

## Camera Control

With the camera control and built-in fast AE adaptation, the MT9M131 can deliver and fully automate the following functions:

- Context switching (for example, resolution/power mode switching)
- Flash (LED/xenon)
- Nonflash snapshot
- Video clip

Both semiautomatic and fully manual modes are also supported for flexibility in system implementation.

While the flash is active, the AWB flash color correction matrix is substituted for the adapted color correction matrix, to maintain white balance.

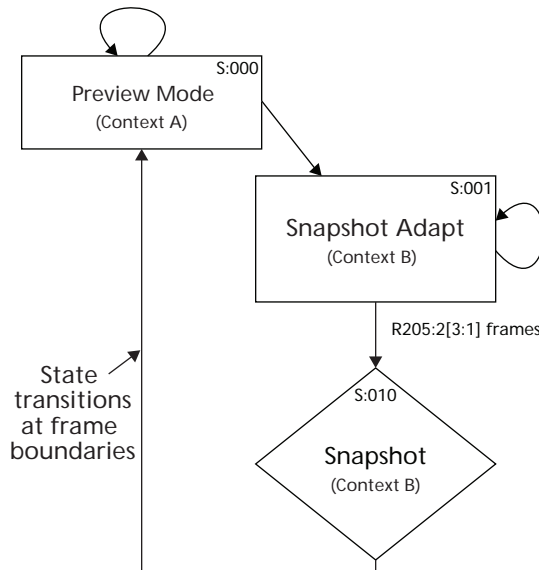
Blanking control is applied by masking out intermediate adaptation frames, so the host sees only the final snapshot frame.

The snapshot picture size can be changed using the zoom control context setting. The snapshot output pixel format is also programmable using context control settings.

Flow charts for flash snapshot (LED and xenon) and nonflash snapshot programming are provided in Figures 6, 7, and 8, starting on page 18.

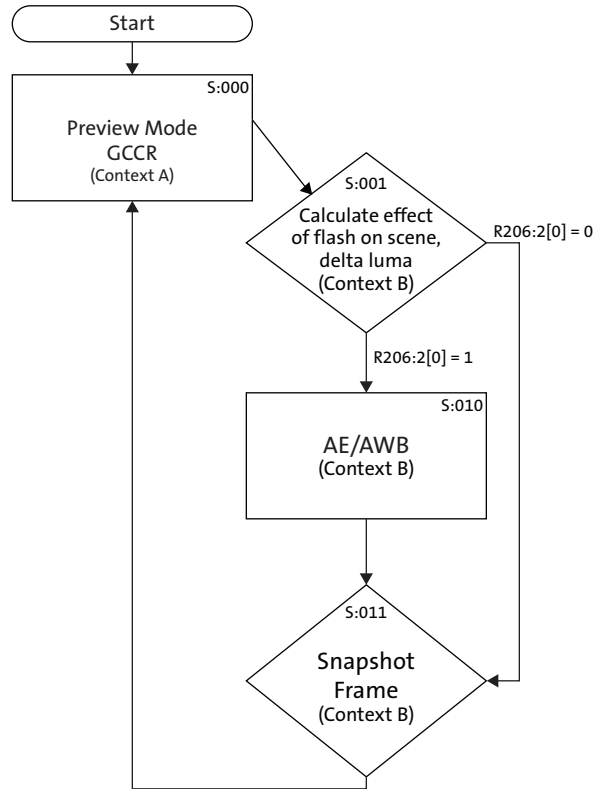
For further information, contact Aptina's local FAE.

**Figure 6: Nonflash Snapshot Program**



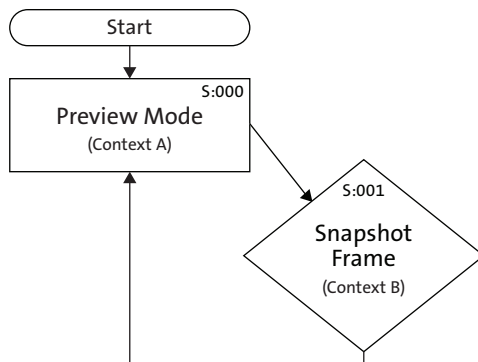
Notes: 1.  $R0x2CC[3:0] = 0000$ ; program advance = 1.

Figure 7: LED Flash Snapshot Program



Notes: 1. R0x2CC[3:0] = 0001; program advance = 1.

Figure 8: Xenon Flash Snapshot Program



Notes: 1. R0x2CC[3:0] = 0010; program advance = 1.  
2. Shutter width = one frame. Fire flash after rows have reset, but before READ.

## Sequencer Operating Modes

Nonflash snapshots are used as examples in each of the following descriptions. Nonflash snapshot mode is set up when 0000 is written to the R0x2CC[3:0].

The following sections provide guidelines for taking nonflash snapshots in each mode:

- Automatic mode
- Semiautomatic mode
- Manual mode

### Automatic Mode

Configure the nonflash program by first writing to R0x2CD, then set R0x2CB[0] to trigger program execution. The hardware sequencer delivers the desired snapshot output and switches back to preview mode when the task is complete.

All global context changes occur during the vertical blanking period.

### Semiautomatic Mode

The host controls AE/AWB commands and global context commands directly on a per-frame basis by programming R0x2D5[0] (the user camera control context control register [UCCCR]) and R0x2D3 (UGCCR), then setting the R0x2CB[0] to trigger the operation.

The hardware ensures that all global context switching occurs in the safe vertical blanking period.

### Fully Manual Mode

**Note:** Manual mode is not a recommended mode and is included here only for completeness. Use automatic mode or semiautomatic mode for normal camera applications.

AE/AWB commands, global context commands, and context switching timing are all controlled by the host writing to R0x2D5 (UCCCR) and R0x2C8 (GCCR). In this mode, the host must poll R0x2CA to detect the EOF calculation done counter increment.

The WRITE to GCCR should occur immediately after the increment is detected to ensure global context switching during the safe vertical blanking period.

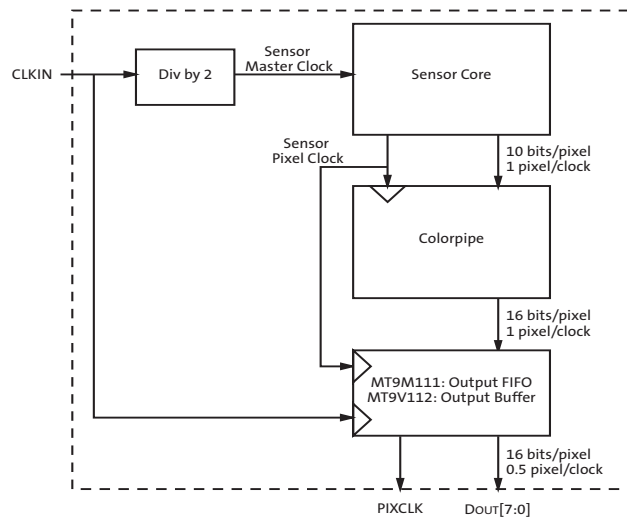
## Clocks and Blanking

The sensor core generates master timing in the system. 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.

The relationship of the primary clocks are depicted in Figure 9.

**Figure 9: Primary Sensor Core Clock Relationships**



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 SXGA images at 15 fps, the sensor core requires a clock in the 24–27 MHz range; this is also the pixel clock frequency (sensor pixel clock) for full power operation.

To achieve this pixel rate, EXTCLK must be in the 48–54 MHz range. The device defaults assume a 54 MHz clock.

The minimum EXTCLK frequency is 2 MHz.

## Primary Operating Modes

The MT9M131 supports two primary modes of operation with respect to the sensor core that affect pixel rate, frame rate, and blanking:

- Full power
- Low power

### Full Power

The sensor is in full resolution mode, generating 1.3 megapixels (SXGA = 1280 x 1024 + border) for interpolation.

The SXGA image fed from the sensor to the colorpipe can be resized in the colorpipe, but the frame rate is still defined by sensor core operation.

In full power mode, with full field of view, the frame rate is invariant with the final image size.

**Table 4: Typical Context B Settings**

Context: Functions	Context B
Sensor read mode settings:	No skipping. Full power, full data rate
Sensor pixel clock:	27 MHz for 54 MHz master clock results in a maximum pixel rate of 27 Mp/s
Maximum frame rate:	For 54 MHz master clock, 15 fps

### Low Power

For preview applications, the sensor is typically run in both low power and skip modes thus generating QSXGA frames (640x512 + border = 336960 pixels).

This full field of view QSXGA image can be resized, but only to resolutions smaller than QSXGA.

The frame rate is defined by the operating mode of the sensor.

**Table 5: Typical Context A Settings**

Context: Functions	Context A
Sensor read mode settings:	Row skip 2x Column skip 2x Low power MAX data rate is half that of full power
Sensor pixel clock:	13.5 MHz for 54 MHz master clock results in a maximum pixel rate of 13.5 Mp/s
MAX frame rate:	For 54 MHz master clock, 30 fps

## 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 6.

**Table 6: Sensor-Timing-Dependent Registers**

Register	Default	Minimum	Function
<b>Context A (Low Power Mode)</b>			
R0x007	190	Full power: 202	Horizontal blanking (units of sensor clock)
		Low power: 114	
R0x008	17	R0x022[2:0]+1	Vertical blanking (number of rows, including black rows)
R0x021	0x040C	N/A	Read mode control Oversize and show border controls are not context switchable.
<b>Context B (Full Power Mode)</b>			
R0x005	388	Full power: 202	Horizontal blanking (units of sensor clock)
		Low power: 114	
R0x006	42	R0x022[2:0]+1	Vertical blanking (number of rows, including black rows)
R0x020	0x0300	N/A	Read mode control [8]: Oversize [9]: Show border

## Border Pixels

The sensor core adds four border pixels all the way around the image.

The active image size at full resolution (typically used in full power mode) is 1288 x 1032.

The active image size in row/col skip (typically used in low power mode) is 648 x 520.

## Blanking Calculations

**Note:** Consult Aptina's local FAE for a spreadsheet to assist with calculating line times.

The MT9M131 default blanking calculations are a function of context, as shown in Table 7.

**Table 7: Blanking Parameter Calculations**

Parameter	Calculation
PC_PERIOD: Sensor Pixel Clock Period	Full power: $(2/54)\mu s = 0.0370\mu s$ Low power: $(4/54)\mu s = 0.0185\mu s$
A: Active Data Time (per line): $R0x004 + 8 (\text{border}) * PC\_PERIOD$	Full power: $A = 1288 * (2/54)\mu s = 47.704\mu s$ Low power: $A = 648 * (4 / 54)\mu s = 48.000\mu s$
Q: Horizontal Blanking: $[R0x005   R00x007] * PC\_PERIOD$	Full power: $Q = 388 * (2 / 54)\mu s = 14.370\mu s$ Low power: $Q = 190 * (4 / 54)\mu s = 14.074\mu s$
Row Time: <sup>1</sup> $Q + A$	Full power: $62.074\mu s$ Low power: $62.074\mu s$
P: Frame Start / End Blanking: $4 * PC\_PERIOD$	Full power: $P = 4 * (2 / 54)\mu s = 0.148\mu s$ Low power: $P = 4 * (4 / 54)\mu s = 0.296\mu s$
V: Vertical Blanking: $[R0x006:0   R00x008] * (Q + A) + (Q - 2 * P)$	Full power: $V = (42 * 62.074) + (14.370 - 0.296) = 2621.333\mu s$ Low power: $V = (17 * 62.074) + (14.074 - 0.593) = 1068.740\mu s$
F: Total Frame Time: $(R0x003 + [R0x006   R00x008]) * (A+Q)$	Full power: $F = (1032 + 42) * 62.074\mu s = 66667.556 \mu s \geq 15 \text{ fps}$ Low power: $F = (520 + 17) * 62.074\mu s = 33333.778 \mu s \geq 30 \text{ fps}$

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.



## Exposure and Sensor Context Switching

Exposure time is defined by the shutter width. This is the number of lines to be reset before starting a frame READ.

If line rate does not change when a mode changes, exposure does not change.

If the sensor's line time changes when switching operation modes, a change in exposure time may occur.

There are two approaches to avoiding this problem:

1. Maintain line time between operation modes. For example, force the same line rate for both preview (typically context A settings) and snapshot (typically context B settings).
2. Set the AE / flicker full frame time registers such that the full-frame time is invariant with mode. For example, ensure that for 60Hz flicker avoidance, the full-frame time is 1/30 of a second for both contexts.

See your FAE for a spreadsheet to assist with calculating line times.

## Switching From Context A to Context B

Under typical or default settings in the MT9M131, the sensor pixel rate doubles when switching from preview (context A) to full resolution (context B). In addition, the number of pixels to be read per line nearly doubles. This keeps the line rates roughly equal. The difference occurs due to border pixels: for SOC operation, there are always eight border pixels regardless of context, therefore, the number of pixels in each line is not quite doubled since the border pixels are not doubled.

Compensate for this by context switching the horizontal blanking.

## Horizontal Blanking

Defined in terms of sensor pixel clocks. The sensor pixel clock rate doubles when switching from low power mode (preview, context A) to full power mode (full resolution, context B). So, to maintain the *same* horizontal blanking time, the value for horizontal blanking would *double*. This is handled by the dual, context switchable horizontal blanking registers.

**Note:** Consult Aptina's local FAE for a spreadsheet to assist with calculating horizontal blanking.

## Switching Frequency

The user can switch between sensor contexts as frequently as necessary without affecting exposure.

## Simple Snapshots

To take a snapshot, switch from context A to context B (with restart) for a few frames, then switch back again, capturing one of the Context B frames as the snapshot.

## Power Saving Modes

The MT9M131 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[1] = 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 pin will always result in entry to the standby state. Refer to Figure 11, Hard Standby Operation Waveform for details. (“Timing Summary” on page 29.)

Following assertion of the STANDBY pin, 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 pin during reset or for at least ten master clock cycles.

### Leaving Hard Standby

De-assert the STANDBY pin.

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

## Entering Soft Standby

To place the sensor core into standby set  $R0x00D[3:2] = 01$ .

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

If the master clock is to be left toggling during standby, set  $R179:1[0] = 1$  to turn off clocks internally.

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

## Leaving Soft Standby

Set  $R0x00D[3:2] = 10$ .

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

## Output Enable Control

The MT9M131 provides hardware and software means of tri-stating the following pads:

DOUT[7:0], FV, LV, PIXCLK, STROBE/FLASH. The ability to automatically tri-state the pads when entering standby is available.

Table 8 details output enable control.

**Table 8: Output Enable Control**

RESET_BAR	STANDBY (pin)	R0x01B3[1] (STBY_SIG_DIS)	OE# (pin)	R0x00D[4] (OUTPUT_DIS)	OUTPUT STATE
0	0	N/A	1	N/A	High-Z
0	0	N/A	0	N/A	Driven
1	1	1 (default)	Don't Care	Don't Care	High-Z
1	0	Don't Care	0	0 (default)	Driven
1	0	Don't Care	1	Don't Care	High-Z
1	0	Don't Care	Don't Care	1	High-Z
1	1	0	0	0 (default)	Driven
1	1	0	1	Don't Care	High-Z
1	1	0	Don't Care	1	High-Z

- Notes:
1. Soft standby register bits ( $R0x00D[3:2] = 01$ ) do not influence the output pin state.
  2. The soft reset register bit ( $R0x00D[5] = 1$ ) for the MT9M131 has the same effect on the output pin state as a hard reset.

## Avoiding Floating Inputs

Many MT9M131 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.

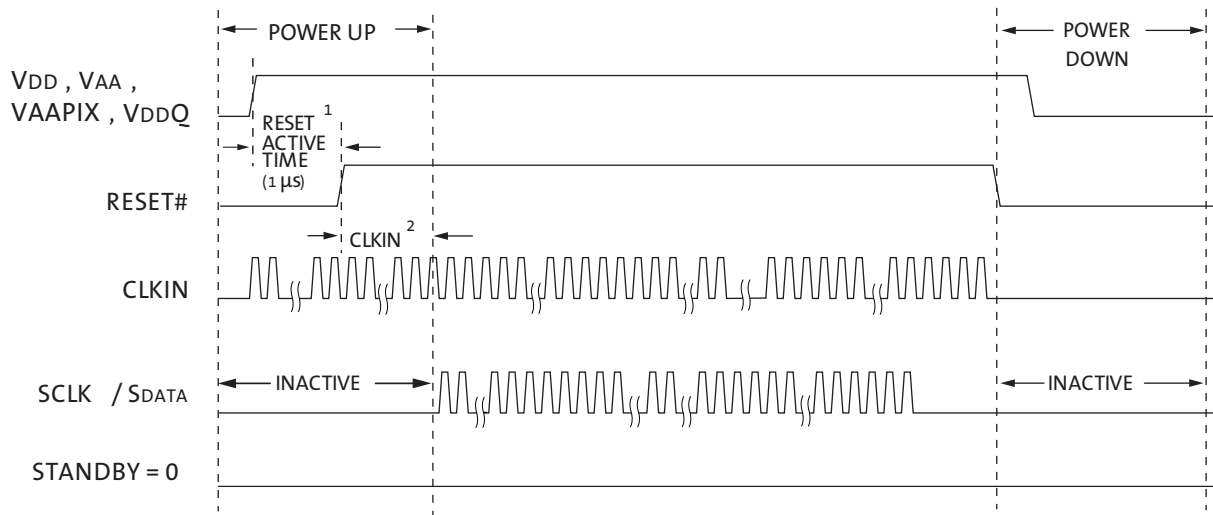
For these two reasons, standard design practice dictates that signal inputs should not be allowed to float for long periods of time.

**Note:** When using the output-enable control, ensure that all MT9M131 bidirectional pads are driven to a valid logic level before they enter a High-Z state.

## 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 10 for an example power-up sequence.

**Figure 10: Power On/Off Sequence**



- Notes:
1. For a safe reset to occur, EXTCLK should be running, and it is recommended that STANDBY is LOW during the reset active time as shown in the sequence.
  2. After RESET\_BAR is HIGH, the user must wait 10 EXTCLK rising edges before initiating two-wire serial interface communication.

## Reset

The reset pin must be asserted for a minimum of  $1\mu\text{s}$  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 MT9M131, with the first rising clock edge during reset, outputs settle to the following:

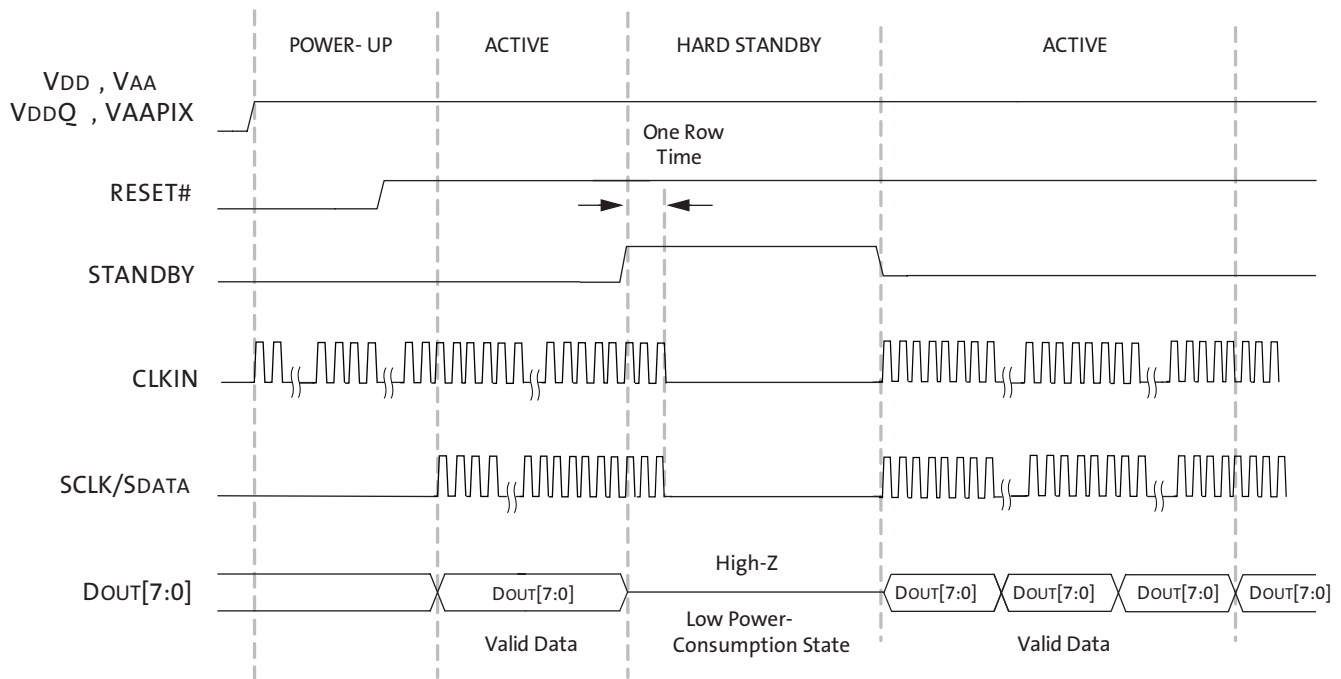
- FV = LV = PIXCLK = 0 (LOW)
- DOUT[7:5] = 0 (LOW)
- DOUT[4] = 1 (HIGH)
- DOUT[3:0] = 0 (LOW)

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

## Timing Summary

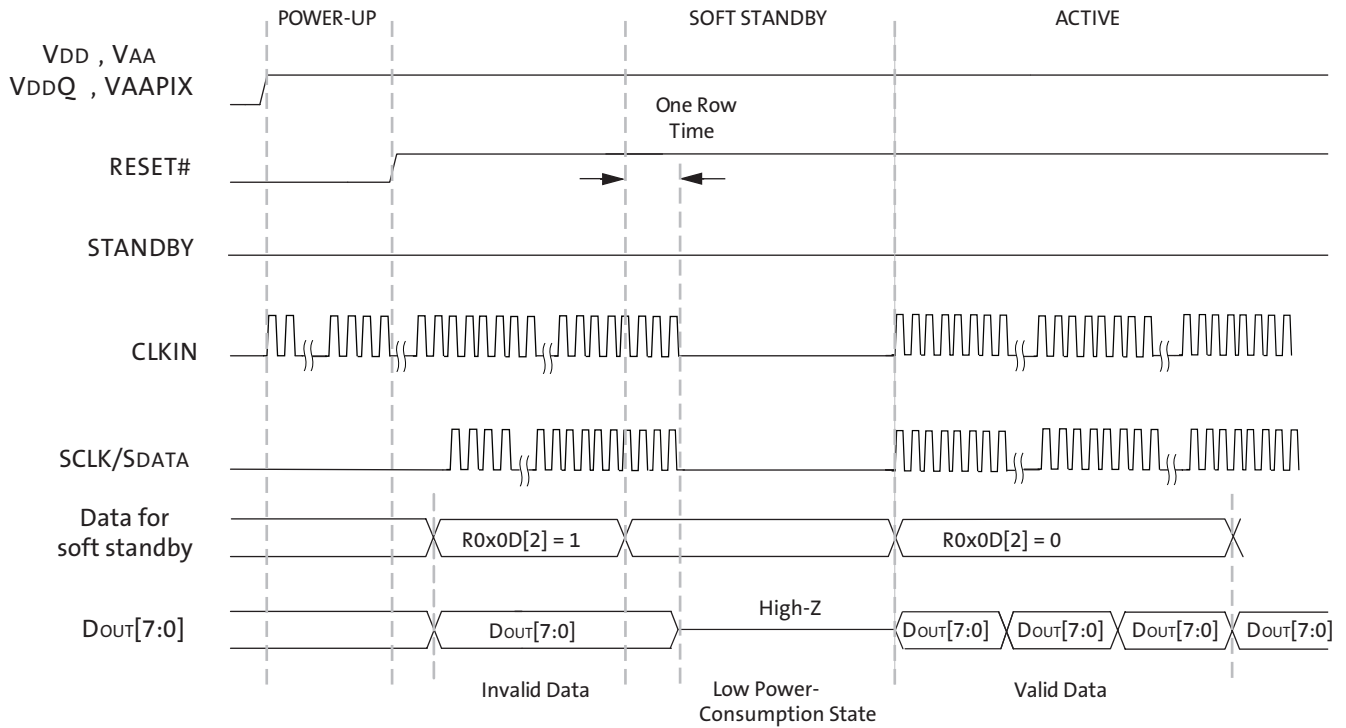
The timing summary for the hard standby mode of the MT9M131 is shown in Figure 11.

**Figure 11: 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  $10\text{k}\Omega$  resistor so that it is not floating during standby.

Figure 12: Soft Standby Operation Waveform



- Notes:
1. Soft standby is asserted/de-asserted by a two-wire serial interface to R0x00D[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.

## Resize and Zoom

Image reduction is controlled by zoom window size, output resolution, and pan offset registers compatible with the MT9M131. 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 9.

**Table 9: Window Setting Restrictions**

Parameter	Notes	Constraints	
Zoom width	Defines field of view. Always defined with respect to the full sensor window, that is, not a function of skip	Sensor width $\geq$ Zoom width.	
Zoom height		Sensor height $\geq$ Zoom height.	
Output width	Output Resolution X Dimension	Column skip disabled.	Column skip enabled.
		Zoom width $\geq$ Output width. Sensor width $\geq$ Output width.	(Zoom width)/2 $\geq$ Output width. (Sensor width)/2 $\geq$ Output width.
Output height	Output Resolution Y Dimension	Row skip disabled.	Row skip enabled.
		Zoom height $\geq$ Output height. Sensor height $\geq$ Output height.	(Zoom height)/2 $\geq$ Output height. (Sensor height)/2 $\geq$ 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.

Placing the sensor in row and column skip mode reduces the number of generated pixels by a factor of four, (a factor of two for row skip and a factor of two for column skip).

If the output size and zoom windows are such that the sensor would generate insufficient pixels in a particular skip mode, then the MT9M131 stops generating frames; that is, it appears to lock up. If this happens, program in register settings that comply with the restrictions detailed above (Table 9) and the sensors will once again generate pixel data.

In typical usage, context A is used for viewfinder modes and context B for snapshots. Context B registers are programmed the same way as context A registers. The MT9M131 supports two contexts for zoom, allowing a different field of view to be configured for each context.

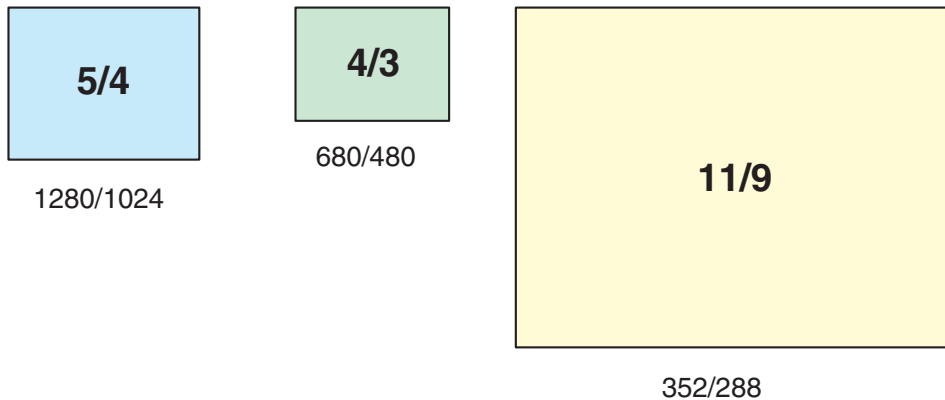
## Aspect Ratios

Maintaining the same aspect ratios in snapshot and viewfinder output sizes and zoom windows helps to avoid distorting the image.

The MT9M131 sensor has three aspect ratios to choose from, as shown in Figure 13:

- 5/4
- 4/3
- 11/9

**Figure 13: Aspect Ratios**



The recommended zoom window settings for these aspect ratios are provided in "Appendix B: Typical Window Settings" on page 88.

## 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.

All examples in the aspect ratio tables in "Appendix B: Typical Window Settings" on page 88 have 1X zoom even though they have different output sizes. The output size must be less than or equal to the zoom window size. The smaller the viewfinder image, the greater the zoom range. A 12 x 10-pixel output image supports greater than 100X zoom.

## 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.

The MT9M131 adds a centered origin option to each of the pan registers. Setting bit 14 of each pan register moves the zoom window origin to the center, so the pan registers can be ignored until the user wants to pan. See "Pan" on page 34 for additional details.



## Manual Zoom

The zoom window can be programmed to jump directly to a particular field of view. Settings are in Table 10 and in “Appendix B: Typical Window Settings” on page 88

**Table 10: Manual Zoom Settings**

Register	Value	Comment
R0x19F	0x4000	Horizontal pan register B with centered origin.
R0x1A2	0x4000	Vertical pan register B with centered origin.
R0x1A5	0x4000	Horizontal pan register A with centered origin.
R0x1A8	0x4000	Vertical pan register A with centered origin.
R0x1AF	0x10	Bilinear (default).
R0x1AF	0x18	Classic interpolation at 1280 x 1024 output, else bilinear (recommended).
R0x020; R0x021	Bits 3:2 = 1,1	Skip mode: allows 30 fps MAX, but limits output size to MIN 640x512 or smaller.
	Bits 3:2 = 0,0	No skip: twice the zoom range, better image, 15 fps MAX.

## Automatic Zoom

The typical zoom usage model uses the manual zoom settings described in the previous section. However, the MT9M131 sensor also provides automatic zoom controls where the zoom window is automatically adjusted in steps between full field of view and the maximum degree of zoom supported for the current output resolution. These settings are shown in Table 11.

The R0x1AF zoom in and zoom out bits behave like optical zoom controls. When the zoom-in bit is set, the internal zoom window is decremented by the values in R0x1AE between each frame. This provides a smooth, continuous zoom with minimal host intervention. In the same manner, the zoom out bit increments the internal zoom window by the values in R0x1AE between each frame. Clearing the zoom bits stops the zoom.

Automatic zoom stops zoom in before the field of view shrinks below the current output resolution as defined in the sensor. It stops zoom out before it exceeds 1280 x 1024.

Automatic zoom does not change the values in zoom window registers. The current internal zoom window can be read from R0x1AB and R0x1AC.

See "Snapshot Following Zoom and / or Pan" on page 35 section for information.

**Table 11: Settings for Automatic Zoom**

Register	Value	Register	Comment
R0x19F	0x4000	Reducer Horizontal Pan—Context B (R/W)	Horizontal pan register B with centered origin.
R0x1A2	0x4000	Reducer Vertical Pan—Context B (R/W)	Vertical pan register B with centered origin.
R0x1A5	0x4000	Reducer Horizontal Pan—Context A (R/W)	Horizontal pan register A with centered origin.
R0x1A8	0x4000	Reducer Vertical Pan—Context A (R/W)	Vertical pan register A with centered origin.
R0x1AF	0x0010	Reducer Zoom Control	Bilinear (default).
	0x0018		Classic interpolation at 1280x1024 output, else bilinear (recommended).
	0x0118		Zoom in.
	0x0218		Zoom out.
	0x0318		Zoom in and out loop for debug and demonstrations.

## Pan

Pan is available when the output image is zoomed in. For applications that do not require pan (such as handheld devices), set all four pan registers to 0x4000 to automatically center images. Stationary applications should also start with 0x4000 in all four pan registers to take advantage of the automatic centering feature.

In the MT9M131, the offset from the center is an 11-bit two's complement number, so its range is from -1024 to +1023, which is sufficient for the smallest output image size. The hardware keeps the image from going off the left or top edges of the sensor while zooming out. The right edge is not checked well and the bottom edge is not checked at all. The driver has to make sure the right and bottom edges of the image stay within the sensor boundaries while panning or zooming out a panned image.

## Snapshot Following Zoom and / or Pan

The MT9M131 has two sets of zoom registers, one set for context A, the other for context B. The easiest way to take a snapshot with the same field of view as the preview is to use the register copies provided in Table 12. If it is known that the pan registers are the same, there is no need to copy them. The output size can be reduced if a smaller snapshot is desired.

**Table 12: Pan and Zoom Registers for Snapshots from Preview Field-of-View**

Copy This Register	Masked With	Comment	To This Register	Comment
R0x1A5	0xFFFF	Horizontal pan in resize A	R0x19F	Horizontal pan in resize B
R0x1A8	0xFFFF	Vertical pan in resize A	R0x1A2	Vertical pan in resize B
R0x1AB	0x07FF	Current zoom in X	R0x1A0	Horizontal zoom in resize B
R0x1AB	0x07FF	Current zoom in X	R0x1A1	Horizontal output size B
R0x1AC	0x07FF	Current zoom in Y	R0x1A3	Vertical zoom in resize B
R0x1AC	0x07FF	Current zoom in Y	R0x1A4	Vertical output size B

## Sensor Window 2x Zoom for Low Power 30 fps Viewfinder

When using skip mode for preview, there are situations where a particular zoom step is not possible due to insufficient pixels being generated by the sensor. To maintain frame rate and achieve the desired zoom, it may be possible to switch the sensor out of skip mode and reduce the sensor window. A detailed example follows.

The sensor window can be used for 2x zoom in low power mode, at 30 fps, with a 48 MHz master clock. This complements the skip mode by providing the rest of the zoom range without giving up low power or high frame rates.

A 640 x 512 sensor window is small enough to provide the same frame rate as skip mode. This limits the zoom window to 640 x 512 or less. The output size must always be less than or equal to the zoom window.

The largest snapshot possible with a 640 x 512 sensor window is 640 x 512. Smaller output sizes, such as QCIF (176 x 144), make room for both manual and automatic zoom.

The lens shading should be turned off or given new coefficients for the 640 x 512 sensor window.

**Table 13: Sensor Window 2x Zoom Procedure**

Register	Value	Comment
R0x1A1	640	X size B: no larger than sensor window.
R0x1A4	512	Y size B: no larger than sensor window.
R0x1A0	640	X zoom B: no larger than sensor window.
R0x1A3	512	Y zoom B: no larger than sensor window.
R0x19F	0x46C0	X pan B: center origin at -320 for automatic zoom.
R0x1A2	0x4700	Y pan B: center origin at -256 for automatic zoom.
R0x1A6	640	X zoom A: no larger than sensor window.
R0x1A9	512	Y zoom A: no larger than sensor window.
R0x1A5	0x46C0	X pan A: center origin at -320 for automatic zoom.
R0x1A8	0x4700	Y pan A: center origin at -256 for automatic zoom.
R0x106	0x4000	Operating mode: bit 10 = 0. Disable lens shading instead of loading coefficient.

**Table 13: Sensor Window 2x Zoom Procedure (continued)**

Register	Value	Comment
R0x0D	0x8008	Sync changes: suspend changes.
R0x020	0x0700	READ_MODE_B low power, oversized; show border, but no skip.
R0x021	0x0400	READ_MODE_A low power but no skip.
R0x003	512	Row width (window height).
R0x004	640	Column width.
R0x001	268	Row start: 12 + 256.
R0x002	348	Column start: 28 + 320.
R0x005	97	Horizontal blanking for 48 MHz Osc.
R0x006	17	Vertical blanking B.
R0x007	97	Horizontal blanking A for 48 MHz Osc.
R0x008	17	Vertical blanking A.
R0x00D	0x0008	Sync changes: apply changes now.

## 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 14 for details.

**Table 14: Aperture Correction Settings**

Register	Value	Comment
R0x105	0x0007	Aperture scale maximum value.
R0x106	Bit 12 = 0	Operating mode control bit 12. Clip small aperture corrections.
R0x1AF	0x0048 0x0000	Classic interpolation at 1280 x 1024 output, else bilinear. Safe for all modes.
R0x13A	Bit 8 = 1	Output format 2A: Mode = RGB, which is slightly sharper than YCbCr.
R0x19B	Bit 8 = 1	Output format 2B: Mode = RGB, which is slightly sharper than YCbCr.

Developers can achieve greater sharpness by disabling bilinear interpolation. This is a highly restrictive mode, using no zoom and limited to the output resolutions shown in Table 15.

Setting R0x1AF to 0x0048 provides the sharpest full resolution image without sacrificing continuous zoom.

**Table 15: Classic Interpolation**

Zoom Control <sup>1</sup>	Output X	Output Y	Read Mode B	Comment
R0x1AF	R0x1A1	R0x1A4	R0x020[3:2]	
0	1280	1024	0,0	SXGA 1280 x 1024, 15 fps max
0	640	512	1,1	QXGA, low power
0	320	256	1,1	QQXGA low power
0	160	128	1,1	160 x 128 low power
0	80	64	1,1	80 x 64 low power

Notes: 1. Bilinear disabled; limited to these output resolutions and no zoom.

### Auto Sharpen

Auto sharpen R0x105[3] generates its own aperture scale value based on the current analog and digital gains to avoid amplifying noise at low light levels. The AE sharpness and saturation control register R0x233 has two gain thresholds as shown in Table 16. One threshold limits aperture scale to 75 percent of its value; the other limits it to 50 percent of its value. The threshold settings are highly subjective and must be tuned to the target application.

**Table 16: Aperture Scale Values**

Gain Zone	Equivalent Aperture Scale Value	Aperture Scale Factor	Comment
0	4	100%	Gain less than R0x233[7:0]
1	3	75%	Else if gain < R0x233[15:8])
2	2	50%	Else

## 2D Defect Correction

The MT9M131 contains a robust defect correction feature. A local 2D spatial kernel detects pixels that differ substantially from their neighbors and then computes a replacement value for the pixel, using values from the neighboring kernels. This has the effect of slightly softening the image.

For the MT9M131, 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 default values for these registers are calculated based on the default line time achieved with a 54 MHz clock. See “Clocks and Blanking” on page 21 for more details.

**Table 17: Flicker Avoidance Registers**

Register	Function	MT9M131 Default
R0x257	Auto exposure frame size for 60Hz Context A	(1/30) / 62.074?s = 537 (0x219)
R0x258	Auto exposure frame size for 50Hz Context A	(1/25) / 62.074?s = 644 (0x284)
R0x259	Auto exposure frame size for 60Hz Context B	(1/30) / 62.074?s = 537 (0x219)
R0x25A	Auto exposure frame size for 50Hz Context B	(1/25) / 62.074?s = 644 (0x284)

Either 50Hz or 60Hz flicker avoidance can be forced (regardless of the state of flicker detection) by programming both frame sizes for a given context to the same value, consistent with the frequency of the flicker to be rejected.

Recalculating these parameters is a straightforward function of line rate, and is largely determined by sensor image width and horizontal blanking.

## 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.)
- Context switching management, by default, maintains a constant exposure when switching between context A and context B.

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

## Controlling AE Speed and Sensitivity

AE speed and sensitivity are controlled by R0x22F in context A, and by R0x29C in context B. As the format of these two registers is the same, the R0x22F register is referenced for convenience.

### Hysteresis Control

The first AE field R0x22F[2:0] determines how much to blend previous luminance measurements with the current luminance measurement.

- 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.

Current luminance can be read as a 16-bit quantity in register R0x24C[15:0]. The blended or time-averaged luminance can be read as an 8-bit quantity in register R0x24D[7:0]. This value is set to zero for fast adaptation.

Use a non-zero value when a slower response to small fluctuations in scene illumination is desired, for instance, in video capture mode.

### Sample Rate Control

The second AE field R0x22F[4:3] 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

R0x22F[8:6] 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 four, which causes AE to adjust to the target luminance in one step. For a smoother response, set the value to two, which causes AE to adjust to a luminance point halfway between its current position and the target luminance. This results in an exponentially asymptotic response.

## Saturation Boost

R0x22F[14] enables a boost to the degree of adaptation if a significant portion of the pixels in an image are luminance saturated, that is, overexposed to white.

White overexposure is common for LED flash snapshots. AE boost adaptation is set through bits R0x227. A value of three produces the greatest boost; a value of zero produces the lowest boost.

## 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 R0x226 and the AE vertical boundaries register R0x227. The little window is defined by the AE center horizontal boundaries register R0x22B and the AE center vertical boundaries register R0x22C. 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 four 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 128 (0x80).

- A value of zero is interpreted as the left most (horizontal) or topmost (vertical) boundary of the image.
- A value of 128 (0x80) 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 18.

**Table 18: 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 R0x262.

- Notes:
1. When any combination of AE, AWB, or flicker detection is enabled, all of the above registers 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, 7, and 14 of register R0x106 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 R0x237 to bind it to a single value. The current value of the AE zone is read from R0x23F.

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 established in registers from R0x257 through R0x5A. The values of these registers are the shutter widths used for zone 4 in the following four situations:

- Context A, 60Hz
- Context A, 50Hz
- Context B, 60Hz
- Context B, 50Hz

Under normal circumstances, the values for contexts A and B are identical, that is, care is taken that switching from preview to snapshot context (or back) does not change the exposure.

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 algorithm must receive the total line size data, and the maximum size of the shutter delay; however, there is no direct method of accessing or computing these values. These values must be input into a set of IFP registers.

The AE line size A register R0x239[10:0] holds the row size (height in pixels) for context A. The AE line size B register R0x23A[10:0] holds the row size for context B.

The AE shutter delay limit A register R0x23B[10:0] holds the maximum shutter delay value for context A. Likewise, the AE shutter delay limit B register R0x23C[10:0] holds the maximum shutter delay value for context B.

Please contact your FAE for a spreadsheet that automates the calculation of appropriate values for AE line size and shutter delay limit registers.

### Minimum Analog Gain

The analog gain in Zone 0 is set by the lower byte of the AE gain limits register R54:2[7:0]. 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.

The AE gains table R0x282–R0x295 encodes the gain settings necessary to step smoothly (in exposure step sizes) through the AE zones.

The AE gains table, along with the AE gain limits register R0x236, the AE zone limits register R0x237, and the AE table gain range control register R0x238 are the parameters that drive the values of the sensor core green channel gains. The AWB color gains ratio drives the ratio of the sensor red and blue gains relative to the sensor green gain.

### 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 R0x236[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 VREF DACs register R0x041. The low voltage reference (VREFLO) is in R0x041[3:0], and the high voltage reference (VREFHI) is in R0x041[7:4]. The AE ADC gain control register R0x23D controls the ADC. The minimum limit value for VREFLO is R0x23E[11:8]. The minimum limit value for VREFHI is R0x23E[3:0]; the maximum limit value for VREFHI is R0x23E[7:4].

**Caution** Although this register is listed here for completeness, we do not recommend modifying the value of register R0x23D, both to ensure the stability of the AE algorithm and to prevent possible damage to the sensor.

After VREFHI R0x041[7:4] is reduced to the value in R0x23D[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 R0x267[7:0]. The current value of digital gain1 can be found in the lower byte of the AE digital gain monitor register R0x262[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 R0x267[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 R0x62[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.

## Zones and Frame Rate

When the value of the shutter width R0x009 exceeds the number of lines of the image R0x003 plus border rows R0x029[8] plus vertical blanking R0x008 and R0x006, the frame rate is decreased. This is due to additional blanking rows being added to meet the integration time specified by the shutter width. For power on default settings (which assumes a 54 MHz master clock frequency), this point occurs when making the exposure gain Zone transitions shown in Table 19 on page 44.

**Table 19: Flicker Abatement**

Preview		Snapshot	
60Hz Flicker Abatement	50Hz Flicker Abatement	60Hz Flicker Abatement	50Hz Flicker Abatement
Zone 4 ⇨ Zone 5	Zone 3 ⇨ Zone 4	Zone 8 ⇨ Zone 9	Zone 6 ⇨ Zone 7

- Notes: 1. These transition points will vary as a function of changes in master clock frequency, horizontal blanking, vertical blanking, and maximum target frame rate. All of these combine to affect the value of the shutter width defined in the AE shutter width basis registers R0x257–R0x25A, which defines the shutter width value for exposure gain zone 4 as a function of flicker abatement frequency and context.

### Maximum Frame Rate Dwell

As the frame rate starts to decrease after zone 4 (after zone 3 for 50Hz flicker abatement), the AE dwell feature enables you to dwell at the maximum frame rate longer by changing the policy on the analog gains.

The range in analog gains for a particular zone is designed to transition smoothly from one zone to another with a single exposure step size of 1/16 of the current value. When going from one zone to the next higher zone (greater exposure), the analog gain must be decreased to compensate for the increase in shutter width. It is undesirable for the analog gain decrease to exactly cancel the increase in exposure caused by the increase in shutter width, since a step increase of 1/16 in the total exposure is still needed to satisfy the AE step size requirement.

For the purposes of discussion, in the following paragraphs assume that the exposure gain zone 4 to zone 5 transition is where the frame rate falls below the maximum.

For zone increase transitions within the range of zones 1 through 4; the value of the analog gain decreases to nearly the minimum gain R0x237[7:0]. The additional AE steps increase the gain with the new zone, and the behavior continues until zone 4.

Within zone 4 a new policy takes effect, and the analog gain is allowed to ramp up to a larger limit value set by the low byte of the AE gain zone range control register R0x238[7:0]. This allows a greater number of AE steps within zone 4, and delays the transition from zone 4 to zone 5. Under default conditions for preview resolution, this causes the AE algorithm to maintain the maximum possible frame rate for a wider range of exposures.

The upper bits of the AE gain zone range control register R0x238[11:8] determine the zone where the new policy takes effect. By default this is zone 4; however, as mentioned in the note in “Zones and Frame Rate” on page 43, the zone transition where the frame rate is decreased is a function of the number of frame timing settings. R0x238[11:8] should be set to the zone just before frame rate would be decreased.

## Target Luminance

The target luminance is contained in the low byte of register R0x22E, and the half range of the target luminance is in the high byte of register R0x22E. As long as the time-averaged luminance low byte of R0x24D is in the range of R0x22E[7:0]± R0x22E[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 R0x246 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 R0x22E[7:0]+ R0x22E[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, R0x246 is set to “0,” disabling this feature.

## AE Gains Table

The AE gains table is encoded for compression and is not easily generated. An application program that calculates the gains table based upon a set of dependent variables has been generated. Check with your FAE for program availability.

## Black Level

The black level, which provides a margin for noise undershoot, is set in the sensor core register field R0x030[9:0] (row noise). In general, the value of this register field should not be changed. If it is changed, a corresponding register field in the color pipeline, R0x13B (black level subtraction), should be changed so that the difference between R0x030[9:0] and R0x13B[9:0] is maintained.

In normal operation, this difference is zero; the values are identical.

**Note:** Modifying the black level from its default values can have a negative effect on the downstream elements of the color pipeline, in some cases reducing the maximum pixel values that will be output. It can also skew the operation of some range thresholding metrics within the pipeline itself.

## Exposure Control

Because the target luminance is linear with respect to exposure, an arbitrary linear relationship defines the exposure setting offsets. Alternatively, if it is assumed that these values are relative f-stops, in increments of a half stop, the values listed in Table 20 for the exposure target register R0x22E[7:0] would be used.

**Table 20: Exposure Settings Values for the Exposure Target Register R0x22E[7:0]**

F#	Calculation	Result (Decimal)	Result (Hex)	Setting
f/0.7	$74 * 2$	148	0x94	+2
f/0.8	$74 * \text{sqrt}(2)$	105	0x69	+1

**Table 20: Exposure Settings Values for the Exposure Target Register R0x22E[7:0] (continued)**

F#	Calculation	Result (Decimal)	Result (Hex)	Setting
f/1	$74 * 1$	74	0x4a	0
f/1.2	$74 / \text{sqrt}(2)$	52	0x34	-1
f/1.4	$74 / 2$	37	0x25	-2

## Frame Rate and Day/Night Mode Exposure

By default, at 60Hz flicker abatement, the MT9M131 is in “night mode,” with a minimum preview frame rate of 5 fps, and a maximum preview frame rate of 30 fps.

The maximum frame rate for snapshot resolution (full SXGA resolution) is half that of preview resolution.

Changing the zone limit register R0x237 can bind the frame rate to a higher minimum value. AE currently enables full range down through zone 24 (default “night mode” at 5 fps), or a limit stop at some intermediate point. There is only one preprogrammed intermediate point per gain table (currently the default is zone 8). If a different intermediate zone limit is required, then the AE gains table must be modified. Some examples are listed in the following sections.

When switching between day mode and night mode, AE must be forced to move into a new zone if it is at the current zone limit. Procedures for switching smoothly between day and night mode are described in the following sections. These procedures assume a clock frequency and horizontal and vertical blanking settings consistent with a maximum frame rate of 30 fps in AE zones 0 through 4.

For assistance with settings for other maximum frame rates, please consult your Aptina FAE.

### Initial Setting or Transition from Night Mode to Day Mode (15 fps)

To change to day mode, where the minimum preview frame rate is 15 fps, use the following:

1. Set R0x237 to 0x100 (AE zone 8).
2. Set R0x282 to 0x03FC.
3. Read R0x22E to save to local variable.
4. Set R0x22E to 0x00 (forcing AE to move).
5. Read R0x23F. If the result is greater than eight (AE has not moved into the legal range), wait 50ms (or one frame) and repeat.
6. Restore R0x22E.

### Transition from Day Mode (15 fps) to Night Mode (5 fps)

To change to night mode, where the minimum frame rate is 5 fps, use the following:

1. Set R0x237 to 0x300 (AE zone 24).
2. Set R0x282 to 0x03F9.
3. Read R0x22E to save to local variable.
4. Set R0x22E to 0x00 (forcing AE to move).
5. Read R0x23F. If the result is equal to eight (AE has not moved higher), wait 50ms (or one frame) and repeat.
6. Restore R0x22E.

### Transition from Day Mode (15 fps) to Alternate Night Mode (7.5 fps)

To change to night mode, where the minimum frame rate is 7.5 fps, use the following:

1. Set R0x237 to 0x200 (AE zone 16).
2. Set R0x282 to 0x03FC.
3. Read R0x22) to save to local variable.
4. Set R0x22E to 0x00 (forcing AE to move).
5. Read R0x23F. If the result is equal to eight (AE has not moved higher), wait 50ms (or one frame) and repeat.
6. Restore R0x22E.

## 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, Pre Gamma

If R0x13C and R0x265 are both increased, the luminance of the image is offset, and the increase is subtracted from the luminance value AE uses for comparison against the target luminance.

### Scheme 3: Post AE, Post AWB, Modify Gamma

These registers R0x153–R0x158 and R0x1DC–R0x1E1 shift the gamma curve upwards. In effect, add the offset into 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 4: 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.

The ITU-R BT.656 YCbCr space requires an offset of 16 for luminance values (to provide for undershoot and coded values). The maximum value for luma with the default settings is 240. For an RGB offset of X, an equivalent offset of  $(224/255) \times X + 16$  is necessary.

## Effects of the Various Brightness Schemes

Scheme 1 creates an unintended effect on AE and AWB. Scheme 2 provides the desired brightness, but could adversely affect AWB. Scheme 3 compresses the dynamic range, but leaves most of the image visible. Scheme 4, the easiest to implement with minimal adverse effect on AE and AWB, tends to shift the effective gamma exponent for the viewing display. Scheme 4 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 21 on page 48 provides values for different brightness levels using scheme 4 by setting the low order 8 bits of R0x134. Table 21 on page 48 lists both the low order 8-bit value and the entire range of the 16 bits of R0x134.

**Table 21:** Scheme 4 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

## Contrast Control

Contrast is typically defined as a gain or scale factor, usually preceding brightness control in the IFP. For the MT9M131, contrast control is achieved through adjustment of exposure control. See “Exposure Control” on page 45 for more details.



## White Balance

### 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 14 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 14: AWB Sensor Controls

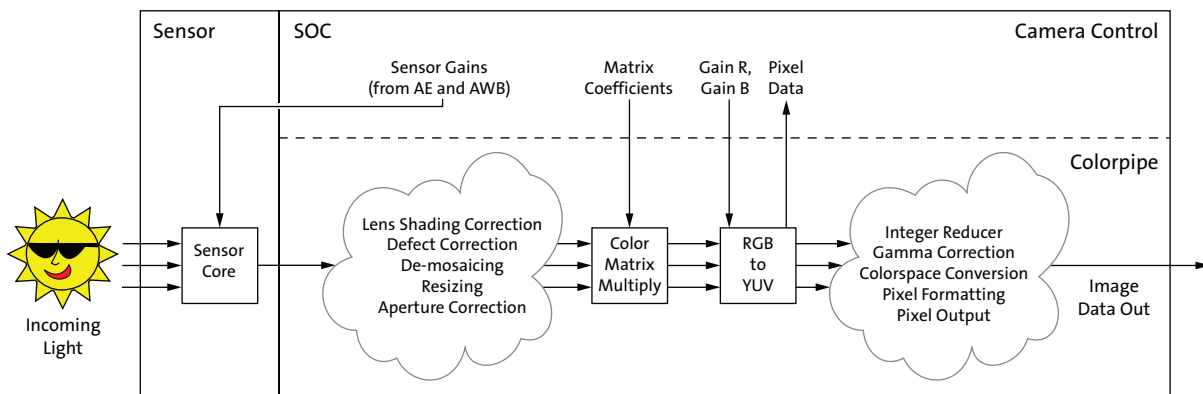
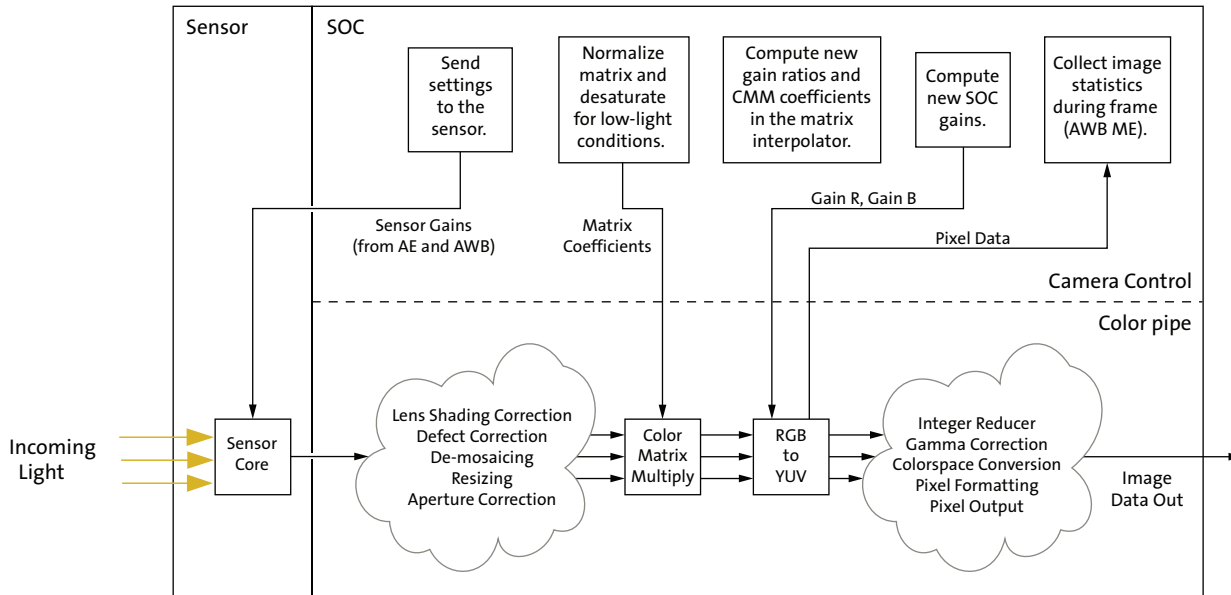


Figure 15 on page 50, expands upon this and adds the processing flow that occurs in the white balance unit between frames. Throughout a frame, the AWB ME collects data from the pixel stream and computes a set of statistics used by the other white balance computations. At the end of a frame, white balance 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 15: AWB Flow Processing



The white balance unit can be configured to work in two modes:

- Gray world mode R0x22A = 0x0000, R0x21F = 0x3F, R0x220 = 0xFF00
- Real-life mode (power-up default plus set R0x22A = 0x0080)

In the gray world mode, AWB makes the picture gray on average. This is a standard mode for many cameras. This mode is recommended for use with Dai Nippon Printing (DNP) color bar and the skin tone (girl with rose) test charts.

The real-life mode typically works best with real-life scenes. For example, gray world AWB has problems when taking close-up photos of people's faces—faces will be gray. AWB attempts to detect such situations and correct the colors. However, color bar and skin tone charts may not look correct in this mode.

To understand the differences between these modes, think of the AWB as a gray-world model with enhancements designed to produce better results in the real world. The gray world mode disables many of these enhancements and works better in certain laboratory test environments.

The benefits of the real-life mode become apparent when operating outside the laboratory environment and in the real world, where natural lighting can be far less than ideal.

## 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 × delta matrix. The blue-rich matrix is at position 127, which is base matrix + 63 × delta matrix.

The flash matrix is discussed under “Flash Color Correction” on page 65. The flash matrix coefficients use the same scale factors as the base and delta matrices. The sensor gain ratios are interpolated along with the color correction matrix coefficients, but without applied scale factors.

## White Balance Settings

There are three ways to approach preset white balance settings:

- Simple AWB
- Assisted 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 R0x224 to the desired matrix position. For example, if the chosen matrix position is 0x23, program R0x224 = 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 R0x212. Table 22 provides useful matrix position examples.

**Table 22: Suggested Matrix Positions for Varying Conditions**

Lighting	Matrix Position Read from R0x212	Matrix Position Constraints R0x224
Cloudy	0x7F	0x7F7F
Compact White Fluorescent	0x2D	0x2D2D
Daylight	0x7F	0x7F7F
Incandescent	0x00	0x0000
U30 TL84 bulb	0x00	0x0000

### Assisted AWB

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 R0x212 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: Set R0x101[1] = 0 and R0x106[15] = 1.
  - 3b. Read the matrix position to check that the matrix has moved.
  - 3c. Return to AWB mode: Set R0x101[1] = 1 and R0x106[15] = 0.

See “AWB Speed and Stability” on page 53 for further information on accelerating CCM movement.

## 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 R0x212 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: R0x101[1] = 0 and R0x106[15] = 1.
  - 3b. Read the matrix position to check that the matrix has moved.
  - 3c. Return to AWB mode: R0x101[1] = 1 and R0x106[15] = 0.

See “AWB Speed and Stability” 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 R0x212 and the red and blue IFP gains R0x213 and R0x214 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 the preceding procedure.)
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:

- R0x228[6:3], the frequency of matrix adjustments
- R0x228[2:0], the SOC digital gain smoothing control

AWB works fast in preview mode and in still-capture applications. It may be acceptable for AWB to oscillate slightly to achieve faster convergence.

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  $R0x006[15] = 1$ . When manual white balance is subsequently turned off by clearing  $R0x006[15]$ , 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  $R0x202$ – $R0x211$  and  $R0x215$ – $R0x21E$  that represent the matrices.
2. Force the new matrices into the matrix interpolator.

For the flash matrix, this is straightforward: the matrix interpolator sees the CCM register values whenever the sequencer switches to the flash matrix. This forces writing sensor values (gains and exposure settings) to the sensor.

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.

Accelerated support for changing the current CCM is built into the MT9M131. A new control bit has been added,  $R0x228[12]$ . Assertion of this bit forces the current CCM to be loaded from the CCM registers during the next vertical blank without the need to enter manual white balance mode.  $R0x228[12]$  is auto-cleared once the CCM has taken effect.

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

1. Load new matrices into the CCM registers.
2. Assert  $R0x228[12]$ .

**Note:** The CCM scale factors  $R0x203$  and  $R0x204$  are not double-buffered in the same way that the CCM coefficients, signs, and analog gain ratios are. The scale factors will take effect immediately as they are written. This can result in a period of time when the CCM scale factors and coefficients are not consistent, potentially resulting in illegal values. The result can be a corrupted frame or partial frame. One workaround is to

keep the scale factors invariant with CCM coefficients; that is, keep one set of scale factors for all CCMs. Your Aptina FAE can provide assistance with this.

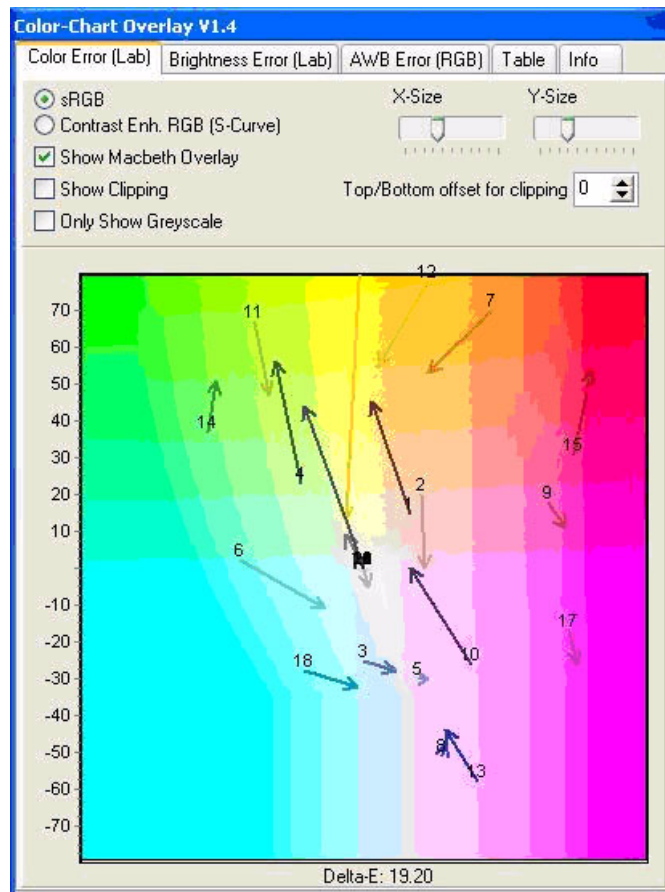
## AWB Tuning

AWB is tuned first, followed by CCM.

### Initialization

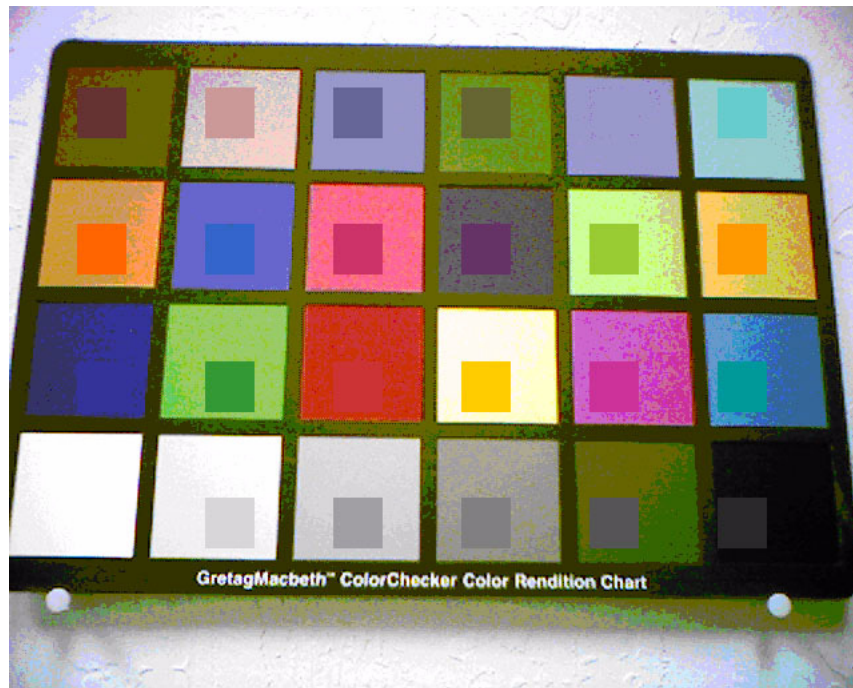
1. Set up the sensor in a light box to view a Macbeth chart. The chart should be centered within the sensors field of view. Ideally, the light source should be behind the imager, not on top (that is, for a Judge II light box, remove the light source from the box top and stand it up behind the sensor 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 16.

Figure 16: 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 17 on page 56.

Figure 17: 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->Gamma, Saturation dialog box. Set the Saturation slider to 100 percent ( $R0x125[5:3] = 0$ ). 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.
4. Set Gamma to 0.45. In the Sensor Control dialog box, Gamma, Saturation page, move the Gamma Correction Table slider to 0.45.
5. 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.
6. Set the Black Correct to "8." In the Sensor Control dialog box, Gamma, Saturation page, move the Black Correct slider to "8."
7. Load lens correction values using the Presets dialog box.
8. Set the digital gains to 1.0. In the Register panel, set  $R0x148[7]$  to "1."
9. 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.
10. 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  $R0x22E[7:0]$  to  $0x44$ .
  - b. Add the Auto Exposure Current Luma Monitor ( $R0x24C$ ) to the watch page by clicking the right mouse button on  $0x24C$  (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 R0x24C is not 0x44.

11. Monitor register R0x212, **Current Color Correction Matrix Position**. Add this register to the watch page by right-clicking on the register in the Register panel.
12. Verify that R0x106 is set to default (AWB, AE enabled, CCM bypassed, LC enabled).
13. Set the **Delta Sensor Core Gain Ratio** register (R0x25F) equal to 0.

### Procedure to Adjust Base Sensor Core Gain Ratio

The goal is to adjust the Base Sensor Core Gain Ratio register (R0x25E) to minimize the AWB error. The procedure is done twice for 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 R0x224 = 0000.
  - 1b. For Daylight, set R0x224 = 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.

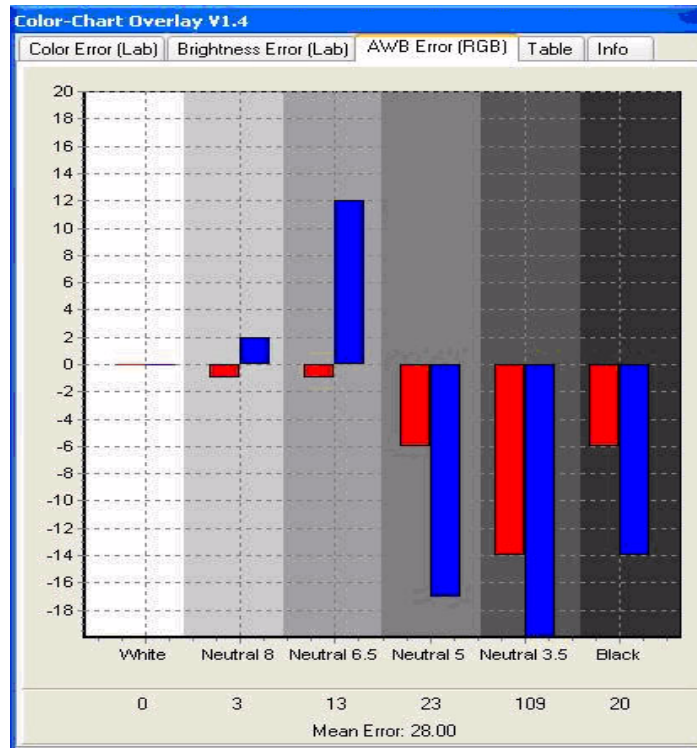
## Using the Color-Chart Overlay - AWB Error (RGB) page

1. In the Color-Chart Overlay dialog box, click on the AWB Error (RGB) page. See Figure 18 below.
2. In the Register panel, adjust R0x25E to center the red and blue color bars as close to zero as possible. When a new value for R0x25E 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.

**Note:** R0x228[12] can also be used to load the new AWB (or CCM) values (this will reload the matrix based on the position in R0x2F5).

3. Wait for the AWB to settle before adjusting R0x25E again; monitor register R0x276 for settled values.
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 18: Color-Chart Overlay: AWB Error (RGB)



## Using Analysis Graph - RGB

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

Figure 19: 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 R0x25E to adjust the red and blue color lines as close as possible to the green line. When a new value for R0x25E 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.

**Note:** R0x228[12] can also be used to load the new AWB (or CCM) values. This will reload the matrix based on the position in R0x2F5.

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 R0x212 for settled values before doing another adjustment.
7. Once the R0x25E value for incandescent is determined, record that number to a safe place and switch lighting to Daylight (switch R0x224 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 (R94:2) and Delta Matrix Gain Ratio (R0x25F) values:

**Table 23: Base and Delta Gain Ratios**

Base Gain Ratios			
Blue	$(0x48 + 0x68)/2$	= 0x58	R0x25E Result: 0x5852
Red	$(0x6E + 0x36)/2$	= 0x52	
Delta Gain Ratios			
Blue	$ABS(0x48 - 0x68)$	= 0x20	R0x25F Result: 0x2038
Red	$ABS(0x6E - 0x36)$	= 0x38	

Note: ABS = Absolute Value.

2. Save the resulting R0x25E and R0x25F values to an INI file. They will be used in the CCM adjustment.

To check the results, perform the following steps:

1. Set R0x224 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 63.

## 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 55. Then follow steps 1 through 8 in “Sensor Setup” on page 56.
2. Load the Base and Delta Sensor Core Gain Ratios (R0x25E, R0x25F) that were determined in AWB tuning.

### Procedure

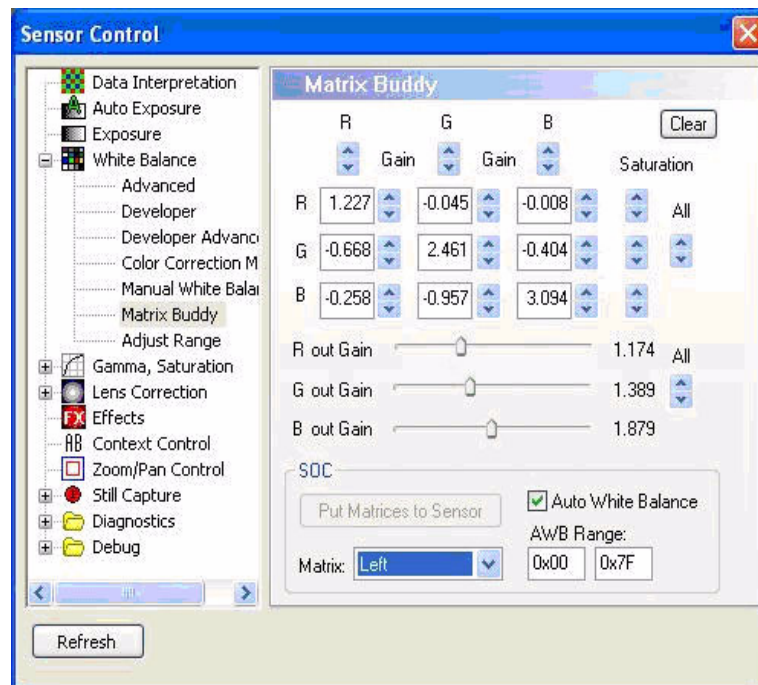
The goal of this procedure is to adjust the Color Correction Matrix so that Macbeth chart squares 13, 14 and 15 (Blue, Green, Red) are as close as possible to reference values. If possible, square 2 (Light Skin) should also be close to reference.

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 R0x224 = 0000.  
Incandescent uses the left Color Correction Matrix.
    - 1b. For Daylight, set R0x224 = 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 R0x212, Current Color Correction Matrix Position, for AWB settling.

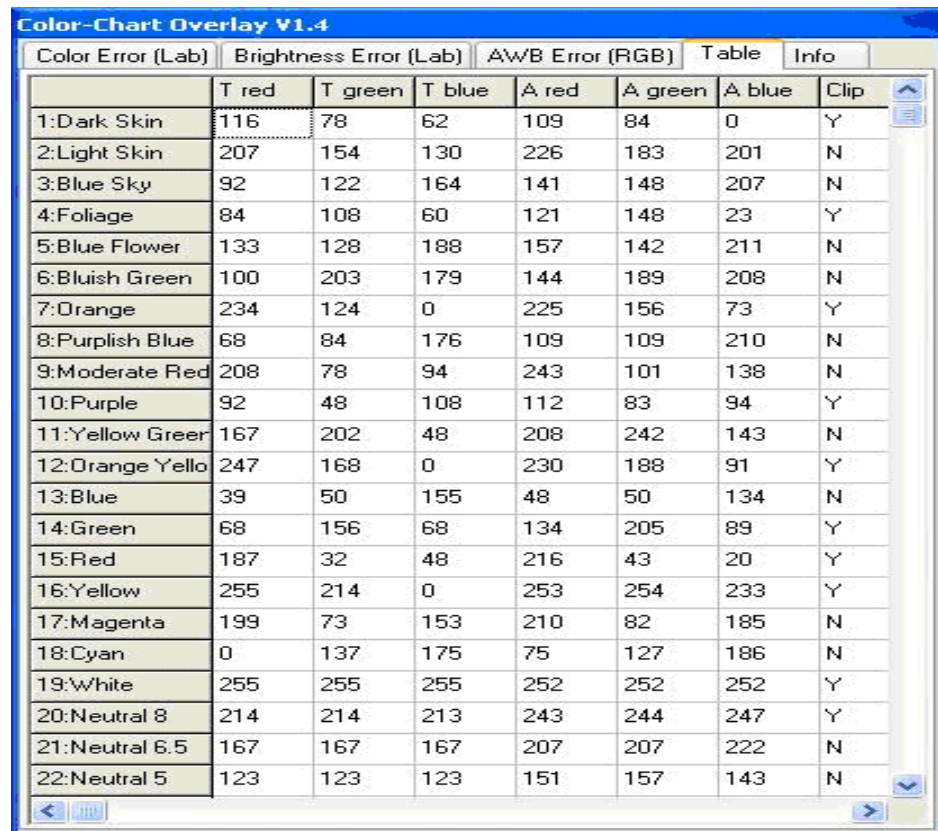
Figure 20: Matrix Buddy Page



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

- In the Color-Chart Overlay dialog box, click on the Color Error (RGB) page. Figure 16 on page 55.  
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).
- 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.
- Concentrate only on the red, green, and blue squares.
- Monitor register R0x212 for settled values before doing another adjustment.
- To view the actual RGB levels versus the reference values, look at the Tables page (See Figure 21 below).  
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 21: Color-Chart Overlay: Table Page

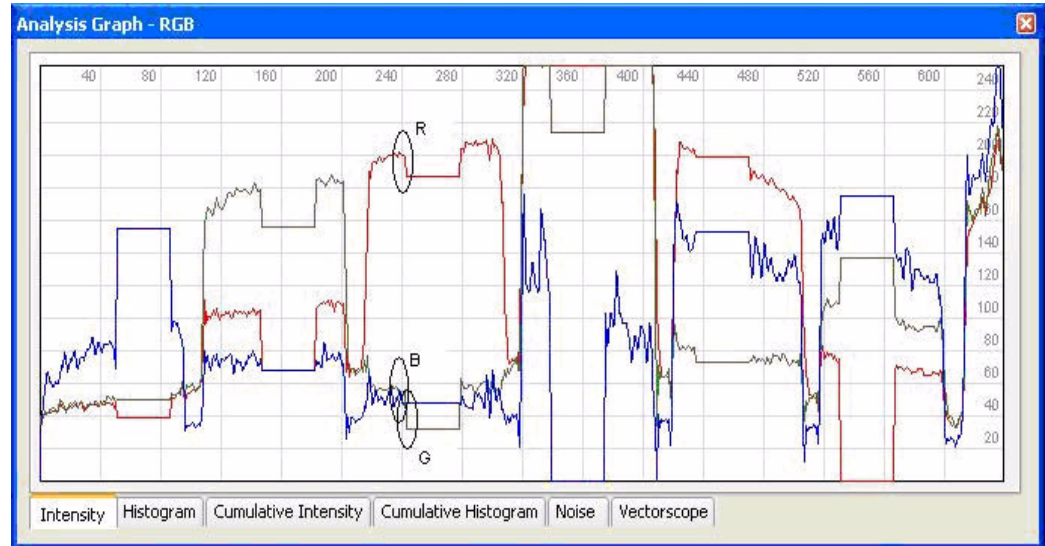


	T red	T green	T blue	A red	A green	A blue	Clip
1:Dark Skin	116	78	62	109	84	0	Y
2:Light Skin	207	154	130	226	183	201	N
3:Blue Sky	92	122	164	141	148	207	N
4:Foliage	84	108	60	121	148	23	Y
5:Blue Flower	133	128	188	157	142	211	N
6:Bluish Green	100	203	179	144	189	208	N
7:Orange	234	124	0	225	156	73	Y
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 Green	167	202	48	208	242	143	N
12:Orange Yellow	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 22: 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).  
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 22, 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. Ideally, the actual R,G,B values should match the reference values but this is not always possible.
4. 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.
5. Concentrate only on the red, green, and blue squares.
6. Monitor register R0x212 for settled values before doing another adjustment.
7. 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>Color Correction Matrix** page and clicking the **Save As** button.
8. Continue the CCM tuning by switching the light source to Daylight. Switch R0x224 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 R0x224 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, CWE, 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>Color Correction Matrix 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.

## Flash Color Correction

The flash CCM provides another matrix for balancing color in flash photography. Based on the settings in the camera control LED flash snapshot program configuration register  $R0x206[13:12]$ , the flash matrix is automatically applied under one of the following conditions:

- Always
- When there is a sufficient change in illumination after the LED turns on
- Never

Instead of defining the full CCM to be used during a flash shot, the user simply defines a flash matrix position,  $R0x2F6$ .

## Effects Settings

To achieve different effects, use the settings for the effects hardware provided in Table 24.

**Table 24: Effects Settings**

Effect	R0x1E2[2:0]	R0x1E3[15:0]
Black & White	1	n/a
Sepia (Antique)	2	0xB023
Reddish	2	0xA040
Greenish	2	0xC0C0
Bluish	2	0x40A0
Negative	3	n/a
Solarization1	4	n/a
Solarization2	5	n/a

- Notes: 1. Most of these effects can also be implemented by loading a specially constructed base CCM and, if desired, flash CCM, and switching to manual white balance mode.

Solarization effects have an additional parameter, which is the threshold specified in R0x1E2[15:8]. Larger threshold values cause the inversions to happen in brighter pixels only, and in the limiting case, nothing is inverted. Smaller threshold values bring the transition to inversion to dimmer pixels, thus affecting more of the image. At a value of zero, all pixels are modified by the effect.

## Lens Shading Correction

The MT9M131 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:

- **Micro lens 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 and from R0x1B6 through R0x1C4. 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 68 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. Click **Control** at the top of the screen.
3. In the **Sensor Control** window, click **Context Control**.
4. To the right of **context**, click **column B** and then **Program Advance**.
5. Ensure that **Program Select** (at the bottom of the Context Control window) has **Snapshot** selected. This ensures the sensor is at its 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.

**Note:** The full resolution image is now averaging the two green channels, for correction by the one green LC correction curve.

### Setup

For each combination of lens, lens position, module, IR cut filter, 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.

- 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

- Click the **control** icon at the top of the screen to open the Sensor Control window.
- In the window, click on **Lens Correction**.
- 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 MT9M131 data sheet for details on register descriptions.
- If another scale setting is to be used, restart the LC calibration procedure from the beginning.
- Click on the **graphs** icon at the top of the main screen (the Analysis Graph window will open).
- Select the **Intensity** tab on the Analysis Graph window.
- Click on the **info** icon at the top of the main screen.
- 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.
- 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.
- Select **Vertical** to complete vertical calibration for that channel before doing anything with another channel.
- 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.
- 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**.

- 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

- Deselect **Enable Lens Correction** on the Lens Correction window.
- On the Analysis Graph, note the selected channel's peak value when LC is turned off. Call this value  $P$ .
- In this section, set the Initial Value for the selected channel so that with LC on, the first pixel in the column has value  $P$ .
- 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. 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:

$$C_{out} = (C_{in})^{\gamma} \quad (\text{EQ 1})$$

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:

$$C_{out} = \left( \left( \left( \left( C_{in} \right)^{\gamma_1} \right)^{\gamma_2} \right)^{\gamma_3} \right)^{\gamma_4} = (C_{in})^{\gamma_1 \gamma_2 \gamma_3 \gamma_4} \quad (\text{EQ 2})$$

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 MT9M131 gamma correction unit, consider the role of the sensor in the image processing chain. Two gamma tables are provided in the MT9M131 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 MT9M131 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 MT9M131 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 ( $4 \times 224$ ). By the time 10-bit lumas 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 11<sup>th</sup> 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

$$Y = (0.299R + 0.587G + 0.114B) + 16 \quad (\text{EQ 3})$$

$$Cr = 0.713(R - (Y - 16)) + 128 \quad (\text{EQ 4})$$

$$Cb = 0.564(B - (Y - 16)) + 128 \quad (\text{EQ 5})$$

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

## Calculating Gamma Table

By default, the MT9M131'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:

$$Y_i = Y_{max} \left( \frac{X_i}{896} \right)^y \quad (\text{EQ 6})$$

where

$$X_{0...10} = \{0, 16, 32, 64, 128, 384, 512, 640, 768, 896\} \quad (\text{EQ 7})$$

$$Y_{max} = \begin{cases} 240 & \text{ITU601/656YCbCr or RGB output} \\ 255 & \text{JPEG-friendly YCbCr output} \end{cases} \quad (\text{EQ 8})$$

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–R0x158 for the context A gamma table, and into R0x1DC–R0x1E1 for the context B gamma table.

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 25 illustrates appropriate values for these registers.

**Table 25: 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 select the appropriate table (context A or context B), 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 practice, most applications can tolerate a common luma offset/clip setting and code for the gamma table in both context A and context B. If different values are required, then new values for luma offset and clip must be programmed when changing context.

The primary applications for the MT9M131 are cellular telephones or PDAs running in one of the RGB modes during preview (for direct display on LCD screens) and in JPEG for taking snapshots. In this case, the only YCbCr output data is for the snapshot, and the luma clip range and offset can be set based on the snapshot settings (for JPEG-friendly output, this will be full range and 0, respectively). The luma clip range only affects YCbCr output data, not RGB data; the offsets are separate for YCbCr and RGB output.

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.

## Test Pattern Generation

The MT9M131 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 R72:1 (R0x148) can output various shades of grey and a color bar test pattern. 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, 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 R0x106[7] = 0 to turn off auto flicker detection.
5. Set R0x106[10] = 0 to turn off lens-shading correction.
6. Set R0x106[14] = 0 to turn off AE.

### Ramp Test Pattern

The output format register for each context can output a ramp test pattern. R0x131[5:4] controls the ramp test output for context A while R0x19B[5:4] controls the ramp test output for context B.

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

$$\text{If } R0x13A[8] \text{ or } R0x19B[8] \text{ for context B) = 0,} \quad (EQ\ 9)$$

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

$$\text{If } R0x13A[8] \text{ or } R0x19B[8] \text{ for context B) = 1,} \quad (EQ\ 10)$$

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

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  $\text{pixel\_position } 28$  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  $0b11100$ ).

## Appendix A: Imaging Developer Guide FAQs

This appendix answers the following frequently asked questions:

### Auto Exposure (AE)

1. How is the speed of AE adaptation and recovery increased? (p. 78)
2. Can adaptation speed be slowed to match the device response speed? (p. 78)
3. Can adaptation speed be slowed further? (p. 78)
4. How can the AWB decisions be biased to produce warmer or cooler images? (p. 78)

### Auto White Balance (AWB)

5. AWB occasionally gets stuck. What should be done when this happens? (p. 78)
6. When should the color correction matrix be calibrated? (p. 78)
7. How should color correction matrices be calibrated? (p. 79)
8. What can be done about pictures that come out with the wrong colors? (p. 79)
9. What can be done when the white balance is incorrect for a test color chart? (p. 79)
10. Is there a difference between AWB windows in preview and snapshot modes? (p. 79)

### Troubleshooting When LC is Turned Off

11. The image has darker corners and a brighter center, or changing tint across the image. How can this be corrected? (p. 79)
12. The sensor response looks flat when LC is off. Can a design be based on this sensor? (p. 79)

### Troubleshooting when LC is Turned On

13. Why does the image have darker or brighter corners and a brighter or darker center, or changing tint across the image? (p. 79)
14. Why does the sensor response look flat with LC on? (p. 79)
15. Are there special requirements for changing to and from preview mode? (p. 80)
16. Why does a change of mirror mode cause lens shading? (p. 80)
17. What timing constraints apply when the host or application changes LC curves? (p. 80)
18. Coloration or shading changes as the multi-position lens moves. Is something wrong? (p. 80)
19. Are manufacturing tolerances of concern? (p. 80)

### Miscellaneous FAQs

20. Is there color suppression in saturated areas? (p. 80)
21. Is there color suppression in low-light conditions? (p. 81)
22. I've attempted to change sensor gains, but there is no change. (p. 81)
23. How is contrast enhanced or improved? (p. 81)

## Auto Exposure (AE)

### 1. How is the speed of AE adaptation and recovery increased?

The speed of AE adaptation is controlled by registers R0x22F context A and R0x29C context B. By default, it is set close to the fastest adaptation speed (0xDF20). The speed at which AE recovers from an overwhelming increase in illumination can be augmented by increasing the value in field R0x22F[13:12] or R0x29C[13:12] from 1 to 2 or 3. The risk of overshoot increases when increasing this field. When switching from one context to another, the speed of adaptation might be decreased.

### 2. Can adaptation speed be slowed to match the device response speed?

To reduce adaptation speed, the value of R0x22F or R0x29C can be set to 0x1444. If any oscillation or stuttering is experienced in AE adaptation with this setting, try 0x9444 as an alternative setting.

### 3. Can adaptation speed be slowed further?

To slow adaptation speed further:

- Increase the hysteresis control field, R0x22F[2:0] or R0x29C[2:0] up to a maximum value of 7
- Increase the frame interval during which AE attempts a change, adjust R0x22F[4:3] or R0x29C[4:3] up to a maximum of 3 (this maximum results in four frames between changes)
- Slow the rate at which AE tries to adjust the current image luminance R0x24C towards the target luminance R0x22E[7:0] by setting the value of R0x22E[8:6] or R0x29C[863] to "0"

**Note:** While it might be tempting to set the width of the AE target luminance window R0x22E[15:8] to "0" to try to guarantee that AE will always converge to the same exposure under identical lighting conditions, doing so will result in instability and oscillation in the AE algorithm. As a general rule, the ratio of R0x22E[7:0] divided by R0x22E[15:8] should be maintained when adjusting R0x22E[7:0] to push image exposure up or down.

## Auto White Balance (AWB)

### 4. How can the AWB decisions be biased to produce warmer or cooler images?

The R0x2F3 register can skew the measured statistics to make the image look cooler or warmer. This feature is useful for adjusting the tint to customer preference and compensating for display white point.

### 5. AWB occasionally gets stuck. What should be done when this happens?

AWB ignores tinted areas. Therefore, when illumination color temperature changes drastically or the scene has a colored background, the camera will see the scene heavily tinted and white balance cannot compensate.

To activate AWB, move the camera around. If this does not activate AWB, relax the settings in R0x22A. R0x22A = 0 configures the SOC to make the picture gray at all times. If this produces insufficient results, change the settings to R0x1F = 0 and R0x21F = 0xC800. Switch AWB to gray-world mode. See "AWB Overview" on page 49 for details.

### 6. When should the color correction matrix be calibrated?

The color correction matrix must be calibrated for each combination of lens and infrared (IR) cut filter. Each camera module manufacturer requires its own set of optimized defaults.

### 7. How should color correction matrices be calibrated?

Contact your Aptina FAE to obtain camera module settings. You will need to supply your FAE with several sample modules.

### 8. What can be done about pictures that come out with the wrong colors?

Make sure the camera defaults are loaded. In DevWare, click on the **Defaults** button in the toolbar. Click the **Presets** button and choose from a list of available preset defaults for various modes.

Check the IR cut filter. A poor IR cut filter can ruin colors, especially when used with incandescent illumination.

### 9. What can be done when the white balance is incorrect for a test color chart?

In the MT9M131, AWB is configured by default to work in the real-life mode, which does not perform well on certain test charts. Change the AWB gray-world mode. See “AWB Overview” on page 49 for details.

### 10. Is there a difference between AWB windows in preview and snapshot modes?

There is no real difference, since the AWB windows specification is very coarse and is a function of screen size. There will be slight differences between them when the preview and snapshot resolutions differ by factors other than powers of two.

Overall, the AWB window specification is independent of image resolution.

## Troubleshooting When LC is Turned Off

### 11. The image has darker corners and a brighter center, or changing tint across the image. How can this be corrected?

Calibrate LC for the lens and module. Turn on LC and load the LC registers with LC curves for the lens. Check the “Procedure for Calibrating LC” on page 68 for a discussion on calibrating parameters for the lens and module.

Lens shading is to be expected, even with a good quality lens. The MT9M131 chip has a shifted microlens array, designed for a small, inexpensive lens and module. When used with a higher quality lens such as a 12mm or similar C-mount lens (as provided with a Demo2 board) the microlens shift itself produces lens shading.

### 12. The sensor response looks flat when LC is off. Can a design be based on this sensor?

To achieve a flat uniform sensor response with LC turned off, a high-quality lens/module having little or no lens shading must be used. To achieve similar results without a high-quality lens, the die must have a shifted microlens array and LC must be turned on. A die without a shifted microlens array is not designed for use in the small modules typically employed in cell phones. Contact your Aptina representative for more details.

## Troubleshooting when LC is Turned On

### 13. Why does the image have darker or brighter corners and a brighter or darker center, or changing tint across the image?

The default values for the LC parameters are not suitable for the module/lens. Load lens-shading parameters appropriate to the module and camera. Check “Procedure for Calibrating LC” on page 68 for instruction on defining these parameters.

### 14. Why does the sensor response look flat with LC on?

Parameters appropriate to the module and lens will be used if criteria I and II are met.

If a module/lens with no lens shading is being used, criteria I and II are satisfied by register settings that provide no correction. Such a set of curves is found in the DevWare .ini file, under **Lens Correction: Calibration—Flat Curves**.

#### 15. Are there special requirements for changing to and from preview mode?

LC provides automatic support for toggling between snapshot and preview modes. No change in register values is necessary. Toggling does not change the appearance of the image. The parameter set is interpreted appropriately, without host intervention.

#### 16. Why does a change of mirror mode cause lens shading?

There is no automatic adjustment for mirror modes. When changing mirror modes, the host must load new LC parameters.

- A separate set of lens-shading parameters must be loaded when entering, exiting, or altering mirror mode.
- A separate set of parameters must be developed for each combination of horizontal and vertical mirroring used.

#### 17. What timing constraints apply when the host or application changes LC curves?

There are times when the application must load a new set of lens shading parameters, different from the set already in the registers (typically during context switching). New sets of lens shading parameters must be loaded during the vertical blanking interval.

#### 18. Coloration or shading changes as the multi-position lens moves. Is something wrong?

A set of LC curves is designed to correct for the lens shading caused by a specific combination of module, lens, and lens position. A new set must be loaded into the LC registers whenever a different lens or lens position introduces different lens-shading characteristics.

If multiple sets are required, for example, for multi-position lenses (focusable lenses), the host must know which position is currently being used and must load a new, appropriate set whenever necessary. If this is impossible, then one set of lens-shading parameters must be used for multiple lens positions. In this case, the different lens positions must exhibit substantially similar lens-shading characteristics, or significant lens shading artifacts will be visible.

#### 19. Are manufacturing tolerances of concern?

Yes, all modules to be used with a single set of LC parameters must be manufactured to a tolerance that ensures that the lens shading characteristics of all the modules, including lenses, filters, and so forth, are sufficiently similar so that all modules are visually acceptable. For example, the positions of the centers of the lenses must not vary so much that multiple sets of correction parameters are required.

#### Miscellaneous FAQs

#### 20. Is there color suppression in saturated areas?

The MT9M131 decreases color saturation for very bright and overexposed pixels. Color suppression is activated when pixel values exceed a relative threshold. R0x125[2:0] specifies the point at which color suppression starts.

The amount of color reduction increases with pixel brightness until 100 percent attenuation is reached at maximum luma.



## 21. Is there color suppression in low-light conditions?

The MT9M131 automatically decreases color saturation in low-light conditions. R23E[12] = 1 enables automatic saturation reduction (default).

Registers R0x133 and R0x23E specify the thresholds at which saturation decreases.

The reduction is performed in two steps. First, decreasing values to 75 percent, and second, decreasing values to 50 percent. The two thresholds can be specified with respect to current analog gain, ADC gain, and either of the digital gains.

R0x133[7:0] and R0x23E[9:8] specify the 75-percent setting threshold; R0x133[15:8] and R0x23E[11:10] specify the 50-percent setting threshold.

## 22. I've attempted to change sensor gains, but there is no change.

In general, the SOC writes gains R0x02B–R0x02E, integration time R0x009, R0x00C, and ADC voltage reference R0x029 settings to the sensor core. If an attempt is made to change sensor gains manually, the manual values will be overwritten.

To set gains manually, disable AE (R0x106[14] = 0, AWB R0x106[14] = 0), force matrix loading R0x106[11] = 0, and flicker detection R0x106[7] = 0.

The following actions can trigger writes to the these sensor core registers:

- Switching to and from the flash matrix
- Switching to and from manual white balance

## 23. How is contrast enhanced or improved?

In the MT9M131, contrast enhancement should be dealt with by the host.

While the MT9M131 can provide contrast control, contrast enhancement is an image-processing function that is not implemented in this device. Methods of contrast enhancement can range from various forms of local or global histogram equalization, to luminance transfer functions that resemble a sigmoid curve, with the inflection point centered on the mean luminance value.

Although the gamma curve could be reprogrammed to be a composition of the contrast-enhancing transfer function and the gamma curve, the ability to shift the contrast curve around is limited by the sparse placement of knee values.

The color correction matrix must be calibrated for each combination of lens and infrared (IR) cut filter. Each camera module manufacturer requires its own set of optimized defaults.

## 24. How should color correction matrices be calibrated?

Contact your Aptina FAE to obtain camera module settings. You will need to supply your FAE with several sample modules.

## 25. Is there a difference between AWB windows in preview and snapshot modes?

There is no real difference because the AWB windows specification is very coarse and is a function of screen size, there will be slight differences between them when the preview and snapshot resolutions differ by factors other than powers of two.

Overall, the AWB window specification is independent of image resolution.

## Troubleshooting When LC is Turned Off

### 26. The image has darker corners and a brighter center, or changing tint across the image. How to correct this?

Calibrate LC for the lens and module. Turn on LC and load the LC registers with LC curves for the lens. See “Procedure for Calibrating LC” on page 68 for a discussion on calibrating parameters for the lens and module.

Lens shading is to be expected, even with a good quality lens. The MT9M131 chip has a shifted microlens array, designed for a small, inexpensive lens and module. When used with a higher quality lens such as a 12mm or similar C-mount lens, as provided with a Demo2 board, the microlens shift itself produces lens shading.

### 27. The sensor response looks flat when LC is off. Can a design be based on this sensor?

A high quality lens/module having little or no lens shading is being used, with a die that does not have a shifted microlens array. Check with your FAE. Such a die is not designed for use in the small modules typically employed in cell phones. Do not design using this die.

If you want to design a module using this sensor, contact your Aptina representative.

## Troubleshooting When LC is Turned On

### 28. Why does the image have darker (brighter) corners and a brighter (darker) center, or changing tint across the image?

The default values for the LC parameters are not suitable for the module/lens. Load lens shading parameters appropriate to the module/camera. “Procedure for Calibrating LC” on page 68 for instruction on defining these parameters.

### 29. Why does the sensor response look flat with LC on?

Parameters appropriate to the module/lens will be used if Criteria I and II are met.

If a module/lens with no lens shading is being used, Criteria I and II are satisfied by register settings that provide no correction. Such a set of curves is found in the DevWare .ini file, under Lens Correction: Calibration—Flat Curves.

### 30. Are there special requirements for changing to/from preview mode?

LC provides automatic support for toggling between full resolution and preview modes. No change in register values is necessary. Toggling does not change the appearance of the image. The parameter set is interpreted appropriately, without host intervention.

### 31. Why does a change of mirror mode cause lens shading?

There is no automatic adjustment for mirror modes. When changing mirror modes, the host must load new LC parameters.

- A separate set of lens shading parameters must be loaded when entering, exiting, or altering mirror mode
- A separate set of parameters must be developed for each combination of horizontal and vertical mirroring used

### 32. What timing constraints apply when the host/application changes LC curves?

There are times when the application must load a new set of lens shading parameters, different from the set already in the registers. To load a new set of lens shading parameters, do the following:

- When changing mirror mode, load new curves during the vertical blanking interval that immediately precedes the first frame affected by the mode.

---

**Caution** If new LC curves are loaded at any other time, bad frames may result.

### 33. Coloration or shading changes as the multi-position lens moves. Is something wrong?

A set of LC curves is designed to correct for the lens shading imparted by a specific combination of module, lens, and lens position. A new set must be loaded into the LC registers whenever a different lens or lens position introduces different lens shading characteristics.

If multiple sets are required, for example, for multi-position lenses such as focusable lenses, the host must know which position prevails, and must load a new, appropriate set whenever necessary. If this is impossible, then one set of lens shading parameters must be used for multiple lens positions. In this case, the different lens positions must exhibit substantially the same lens shading characteristics, or significant lens shading artifacts will be visible.

### 34. Are manufacturing tolerances of concern?

All modules to be used with a single set of LC parameters must be manufactured to a tolerance that ensures that the lens shading characteristics of all the modules, including lenses, filters, and so forth, are sufficiently similar that all modules are visually acceptable.

For example, the positions of the centers of the lenses must not vary so much that multiple sets of correction parameters are required.

### 35. What factors contribute to lens shading?

The following factors contribute to lens shading and should be taken into account as the camera module is adjusted.

## Miscellaneous FAQs

### 36. Is there color suppression in saturated areas?

These devices decrease color saturation for very bright and overexposed pixels. Color suppression is activated when pixel values exceed a relative threshold. R0x125[2:0] specifies the point at which color suppression starts.

The amount of color reduction increases with pixel brightness until 100% attenuation is reached at maximum luma.

### 37. Is there color suppression in low light conditions?

These devices automatically decrease color saturation in low light conditions. R0x23E[12] = 1 enables automatic saturation reduction (default).

Registers R0x233 and R0x23E specify the thresholds at which saturation decreases.

The reduction is performed in two steps. First, decreasing values to seventy-five percent, and second, decreasing values to fifty percent.

R0x233[7:0] and R0x23E[9:8] specify the seventy-five percent setting threshold; R0x233[15:8] and R0x23E[11:10] specify the fifty percent setting threshold.

The two thresholds can be specified with respect to current analog gain, ADC gain, and either of the digital gains.

### 38. When trying to change sensor gains, there is no change.

In general, the SOC writes gains (R0x02B–R0x02E), integration time (R0x009, R0x00C), and ADC voltage reference (R0x029) settings to the sensor core.

If an attempt is made to change sensor gains manually, the manual values will be overwritten.

To set gains manually, disable auto exposure (R0x106[14] = 0), AWB (R0x106[1] = 0), forced matrix loading (R0x106[11] = 0), and flicker detection (R0x106[7] = 0).

The following actions can trigger writes to these sensor core registers:

- Switching to/from the flash matrix
- Switching to/from manual white balance

### 39. How is contrast enhanced or improved?

For these devices, contrast enhancement should be dealt with by the host.

While these devices provide contrast control, contrast enhancement is an image processing function that is not implemented in this device. Methods of contrast enhancement can range from various forms of local or global histogram equalization, to luminance transfer functions that resemble a sigmoid curve, with the inflection point centered on the mean luminance value.

Although the gamma curve could be reprogrammed to be a composition of the contrast enhancing transfer function and the gamma curve, the ability to shift the contrast curve around is limited by the sparse placement of knee values.

### 40. How can pixel-wise fixed-pattern noise be minimized?

Register R0x034 should be set to 0xC039 to minimize the fixed-pattern noise.

## Appendix C: Glossary of Terms

Table 26: Glossary of Terms

Term	Definition
1.3Mp resolution	1280 x 1024 pixels
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 <a href="http://www.w3.org/Graphics/Color/sRGB">www.w3.org/Graphics/Color/sRGB</a> .
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	SXGA(1,280 x 1,024) – 15 fps at 54 MHz QSXGA(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.
GCCR	Global context control register
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
PDA	Personal digital assistant
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.



Table 26: Glossary of Terms (Continued)

Term	Definition
QSXGA	High definition super extended graphics array (2560 × 2048 = 1280 × 1024 × 4 face) maximum
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
SCLK	Serial clock for two-wire serial interface
Sensor core	In the MT9M131, 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.
SXGA	Super extended graphics array; refers to a resolution standard of 1280 by 1024 pixels.
UGCCR	User global context control register

## 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.



## Appendix B: Typical Window Settings

The following tables summarize aspect ratio, zoom, viewfinder, and snapshot settings.

**Table 27: Aspect Ratio, Zoom, Viewfinder, and Snapshot Window Settings**

View Finder Output	Aspect Ratio	Full Field of View	Zoom Range		Zoom Window		Output Size		Zoom Step
			Skip Mode	No Skip	X <sup>1</sup>	Y	X	Y	
					R0x1A6, R0x1A0	R0x1A9, R0x1A3	R0x1A7, R0x1A1	R0x1AA, R0x1A4	R0x1AE, R0x1AE
1280 x 1024	5/4	1280 x 1024	No	1x	1280	1024	1280	1024	0x0A08
640 x 512	5/4	1280 x 1024	1x	2x	1280	1024	640	512	0x0A08
320 x 256	5/4	1280 x 1024	2x	4x	1280	1024	320	256	0x0A08
300 x 240	5/4	1280 x 1024	2.13x	4.26x	1280	1024	300	240	0x0A08
208 x 166	5/4	1280 x 1024	3.07x	6.15x	1280	1024	208	166	0x0A08
176 x 141	5/4	1280 x 1024	3.63x	7.27x	1280	1024	176	141	0x0A08
160 x 128	5/4	1280 x 1024	4x	8x	1280	1024	160	128	0x0A08
150 x 120	5/4	1280 x 1024	4.26x	8.53x	1280	1024	150	120	0x0A08
120 x 96	5/4	1280 x 1024	5.3x	10.6x	1280	1024	120	96	0x0A08
1280 x 960	4/3	1280 x 960	No	1x	1280	960	1280	960	0x0C09
640 x 480	4/3	1280 x 960	1x	2x	1280	960	640	480	0x0C09
320 x 240	4/3	1280 x 960	2x	4x	1280	960	320	240	0x0C09
160 x 120	4/3	1280 x 960	4x	8x	1280	960	160	120	0x0C09
120 x 90	4/3	1280 x 960	5.3x	10.6x	1280	960	120	90	0x0C09
704 x 576	11/9	1251 x 1024	No	1.7x	1251	1024	704	576	0x0B09
352 x 288	11/9	1251 x 1024	1.7x	3.5x	1251	1024	352	288	0x0B09
176 x 144	11/9	1251 x 1024	3.5x	7.1x	1251	1024	176	144	0x0B09
120 x 98	11/9	1251 x 1024	5.2x	10.4x	1251	1024	120	98	0x0B09

Notes: 1. X and Y reduction are independent, making it is possible to squeeze one aspect ratio to another, however, this causes circles to appear as ovals.

**Table 28: Zoom Window Settings for Common Aspect Ratios**

Aspect Ratio	Field of View	Zoom Window A		Zoom Window B		Zoom Step	Comment
		X	Y	X	Y		
		R0x1A6	R0x1A9	R0x1A0	R0x1A0	R0x1AE	
5/4	1280 x 1024	1280	1024	1280	1024	0x0A08	x = 10 y = 8
4/3	1280 x 960	1280	960	1280	960	0x0C09	x = 12 y = 9
11/9	1251 x 1024	1251	1024	1251	1024	0x0B09	x = 11 y = 9





**Table 29: Manual Zoom Examples**

Aspect Ratio	Field of View	Zoom	Zoom Window A		Zoom Window B	
			X	Y	X	Y
			R0x1A6	R0x1A9	R0x1A0	R0x1A3
5/4	1280 x 1024	1x	1280	1024	1280	1024
5/4	1275 x 1020	1.004x	1275	1020	1275	1020
5/4	640 x 512	2x	640	512	640	512
5/4	960 x 768	3x	960	768	960	768
5/4	320 x 256	4x	320	256	320	256
5/4	300 x 240	4.26x	300	240	300	240
5/4	208 x 166	6.15x	208	166	208	166
5/4	176 x 141	7.27x	176	141	176	141
5/4	160 x 128	8x	160	128	160	128
5/4	120 x 96	10.6x	120	96	120	96
5/4	40 x 32	32x	40	32	40	32
4/3	1280 x 960	1x	1280	960	1280	960
4/3	640 x 480	2x	640	480	640	480
4/3	960 x 720	3x	960	720	960	720
4/3	320 x 240	4x	320	240	320	240
4/3	160 x 120	8x	160	120	160	120
4/3	120 x 90	10.6x	120	90	120	90
4/3	40 x 30	32x	40	30	40	30
11/9	1251 x 1024	1x	1251	1024	1251	1024
11/9	625 x 512	2x	625	512	625	512
11/9	417 x 341	3x	417	341	417	341
11/9	352 x 288	3.55x	352	288	352	288
11/9	313 x 256	4x	313	256	313	256
11/9	250 x 205	5x	250	205	250	205
11/9	176 x 144	7.11x	176	144	176	144

**Table 30: Viewfinder Examples (R0x0C8 = 0, Context A)**

Zoom	Field of View	Output	Zoom X	Zoom Y	Output X	Output Y
			R0x1A6	R0x1A9	R0x1A7	R0x1AA
1x	1251 x 1024	176x144	1251	1024	176	144
2x <sup>1</sup>	704 x 576	176x144	704	576	176	144
4x <sup>2</sup>	352 x 288	176x144	352	288	176	144
8x	176 x 144	176x144	176	144	176	144
1x	1280 x 1024	176x144	1280	1024	176	144
2x	640 x 480	176x144	640	480	176	144
4x	320 x 240	176x144	320	240	176	144

- Notes: 1. Actual zoom is  $1024/576 = 1.78x$ ,  
2. Actual zoom is  $1024/288 = 3.56x$



Table 31: Snapshot Examples (R0x0C8 = 0x90FB, Context B)

Zoom	Field of View	Output	Zoom X	Zoom Y	Output X	Output Y
			R0x1A0	R0x1A3	R0x1A1	R0x1A4
1x	1280 x 1024	SXGA 1280x1024	1280	1024	1280	1024
1x	1280 x 1024	VGA 640x480	1280	1024	640	480
2x	640 x 480	VGA 640x480	640	480	640	480
1x	1280 x 1024	QVGA 320x240	1280	1024	320	240
2x	640 x 480	QVGA 320x240	640	480	320	240
4x	320 x 240	QVGA 320x240	320	240	320	240
1x	1280 x 1024	QQVGA 160x120	1280	1024	160	120
2x	640 x 480	QQVGA 160x120	640	480	160	120
4x	320 x 240	QQVGA 160x120	320	240	160	120
8x	160 x 120	QQVGA 160x120	160	120	160	120

Note: R0x0C8 will read back as 0x000B, R0x1C8 will read back as 0x10FB.



---

## Revision History

Rev. B .....	2/10
• Updated to Aptina template	
Rev. A .....	9/07
• Preliminary release	

10 Eunos Road 8 13-40, Singapore Post Center, Singapore 408600 prodmktg@aptina.com www.apina.com  
Aptina, Aptina Imaging, DigitalClarity, 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.