



# 1/6-Inch SOC VGA Digital Image Sensor Developer Guide

## MT9V112

### About This Document

This developer guide provides the information for applications which use Micron's MT9V112 as an integral part of the design. It is intended to supplement the MT9V112 data sheet and is provided exclusively for licensed developers.

### Applicable Parts

This document is applicable to the MT9V112, a VGA-format, single-chip camera with a 1/6-inch CMOS active-pixel digital image sensor. This device combines the MT9V012 image sensor core with Micron's fourth-generation digital image flow processor technology. It captures high-quality color images at VGA resolution.

### Conventions and Notations

This developer guide—like data sheets, errata, and technical notes—follows the conventions and notations described below.

- Register address convention:
  - Decimal notation: R35:1 is page 1, address 35
  - Hex notation: R0x123 is page 1, address 0x23
- Hexadecimal has 0x prefix
- Binary numbers have 0b prefix
  - Example: 0b1010 = 0xA
- Fixed point notation:
  - 0.8 (0.0 through 254/255)
  - 1.7 (0.0 through 1 + 127/128)
- I/O signals can be LOW (0 or DGND), HIGH (1 or VDDQ), or floating (high impedance or High-Z)
- Timing diagrams are not drawn to scale and do not illustrate the actual number of clocks necessary.

The following register address notations are used in this document:

- **R<decimal address>:<address page>** Example: R9:0—Shutter width register in sensor page (page 0). Used to uniquely specify a register.
- **R<decimal address>** Example: R240—Page address register. Used when the register address is global in all three pages or when by context the address page is understood.
- **R0x<2 digit hex address>** Example: R 0xF0—Page address register. Used when the register address is global in all three pages, or when by context the address page is understood.

See “Address Pages” on page 11.



## Table of Contents

About This Document .....	1
Applicable Parts .....	1
Conventions and Notations .....	1
Features .....	6
Feature Set Comparison .....	6
Functional Overview .....	8
Internal Architecture .....	9
Two-Wire Serial Interface .....	10
Serial Bus Description .....	10
Protocol .....	10
Address Pages .....	11
Page Location .....	11
Device ID .....	11
Contexts .....	11
Context Selection .....	12
Context Switching .....	14
Preview and Snapshot Modes .....	15
Preview Mode .....	15
Snapshot Mode .....	15
Switching Modes .....	15
Camera Control .....	15
Sequencer Operating Modes .....	18
Automatic Mode .....	18
Semiautomatic Mode .....	18
Manual Mode .....	18
Clocks and Blanking .....	19
Primary Operating Modes .....	19
Tuning Frame Rates .....	19
Border Pixels .....	20
Blanking Calculations .....	20
Exposure and Sensor Context Switching .....	21
Context Switching .....	21
Power-Saving Modes .....	22
Output Enable Control .....	23
Avoiding Floating Inputs .....	23
Dual Camera Systems .....	24
Power Sequencing .....	24
Reset .....	25
Timing Summary .....	26
Resize and Zoom .....	28
Aspect Ratios .....	29
Zoom .....	29
Advantages of Using the Zoom Window vs. the Sensor Window .....	29
Automatic Zoom .....	29
Pan .....	30
Snapshot Following Zoom and/or Pan .....	30
Sharpness .....	31
Aperture Correction .....	31
Auto Sharpen .....	31
2D Defect Correction .....	31
Flicker Avoidance .....	31



## MT9V112: Developer Guide Table of Contents

Auto Exposure (AE) . . . . .	.32
Controlling AE Speed and Sensitivity. . . . .	.33
AE Window and Backlight Compensation . . . . .	.33
Exposure Step Size . . . . .	.34
AE Zones. . . . .	.34
Target Luminance . . . . .	.38
Black Level . . . . .	.39
Exposure Control . . . . .	.39
Frame Rate and Day/Night Mode Exposure . . . . .	.39
Brightness Control . . . . .	.40
Scheme 1: Pre AE, Pre Gamma . . . . .	.40
Scheme 2: Post AE, Pre Gamma . . . . .	.40
Scheme 3: Post AE, Post AWB, Modify Gamma . . . . .	.41
Scheme 4: Post Gamma. . . . .	.41
Effects of the Various Brightness Schemes . . . . .	.41
Contrast Control. . . . .	.41
White Balance . . . . .	.42
AWB Overview. . . . .	.42
Color Correction Matrices . . . . .	.44
White Balance Settings . . . . .	.44
AWB Speed and Stability. . . . .	.46
Flash Color Correction . . . . .	.48
Effects Settings. . . . .	.48
Lens Shading Correction . . . . .	.48
Procedure for Calibrating LC . . . . .	.49
Gamma Correction . . . . .	.54
Selecting a Gamma Factor . . . . .	.54
Gamma Tables—Beyond Gamma Correction . . . . .	.55
Calculating Gamma Table . . . . .	.55
Gamma Tables and Luma Clip Range and Offset . . . . .	.56
Test Pattern Generation . . . . .	.57
Appendix A: Imaging Developer Guide FAQs . . . . .	.58
Appendix B: MT9V112 Image Resolution Settings . . . . .	.63
Appendix C: Glossary of Terms . . . . .	.64
Contacting Micron . . . . .	.66
Revision History. . . . .	.67



## List of Figures

Figure 1:	MT9V112 Functional Block Diagram . . . . .	8
Figure 2:	Internal Registers Grouping . . . . .	9
Figure 3:	16-bit MT9V112 Example: Read Timing from R0x009, Returned Value 0x0284 . . . . .	10
Figure 4:	16-bit MT9V112 Example: Write Timing to R0x009, Value Written 0x0284 . . . . .	11
Figure 5:	Nonflash Snapshot Program . . . . .	16
Figure 6:	LED Flash Snapshot Program . . . . .	16
Figure 7:	Xenon Flash Snapshot Program . . . . .	17
Figure 8:	Dual Camera System Output Enable Control. . . . .	24
Figure 9:	Power On/Off Sequence: MT9V112 . . . . .	25
Figure 10:	Hard Standby Operation Waveform . . . . .	26
Figure 11:	Soft Standby Operation Waveform . . . . .	27
Figure 12:	Aspect Ratios . . . . .	29
Figure 13:	AWB Sensor Controls . . . . .	42
Figure 14:	AWB Flow Processing . . . . .	43



## List of Tables

Table 1:	Feature Set Comparison: MT9V111, MT9M111, and MT9V112 .....	6
Table 2:	Page Location Conventions (Same Page, Different Notations) .....	11
Table 3:	Global Context Control Register Definitions .....	12
Table 4:	Typical Context Settings .....	19
Table 5:	Sensor-Timing-Dependent Registers .....	19
Table 6:	MT9V112 Output Enable Control .....	23
Table 7:	Window Setting Restrictions .....	28
Table 8:	Settings for Automatic Zoom .....	30
Table 9:	Pan and Zoom Registers for Snapshots from Preview Field-of-View .....	30
Table 10:	Aperture Correction Settings .....	31
Table 11:	Aperture Scale Values .....	31
Table 12:	Flicker Avoidance Registers .....	32
Table 13:	AE Zones .....	34
Table 14:	Abatement .....	37
Table 15:	Exposure Settings Values for the Exposure Target Register R46:2[7:0] (R0x22E[7:0]) .....	39
Table 16:	Scheme 4 Brightness Settings .....	41
Table 17:	Suggested Matrix Positions for Varying Conditions .....	45
Table 18:	Effects Settings .....	48
Table 19:	Output Register Values .....	56
Table 20:	Aspect Ratio Zoom Window Settings .....	63
Table 21:	Glossary of Terms .....	64



## Features

The MT9V112 provides sophisticated features and processing functions including:

- Color recovery
- Color correction matrix (CCM)
- Sharpening
- Programmable gamma correction
- Auto black reference clamping
- Auto exposure (AE)
- Automatic 50Hz/60Hz flicker avoidance
- Lens shading correction
- 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 MT9V112 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

## Feature Set Comparison

Table 1 compares the summarized feature sets of the MT9V111, the MT9M111, and the MT9V112.

**Table 1: Feature Set Comparison: MT9V111, MT9M111, and MT9V112**

Feature	MT9V111	MT9M111	MT9V112
Sensor resolutions	Single, 648 x 488	Dual: 1288 x 1032, 648 x 520	Single, 648 x 488
Clock speed	24 MHz (30 fps), 12 MHz (15 fps)	48–54 MHz (30 fps preview, 15 fps snapshot)	24–27 MHz (30 fps), 12–13.5 MHz (15 fps)
Clock parity	Launch data on falling edge, capture data on rising edge	Launch data on falling edge, capture data on rising edge	Launch data on falling edge, capture data on rising edge
Register address pages	2 pages, 1 for sensor and 1 for IFP	3 pages, 1 for sensor and 2 for IFP	3 pages, 1 for sensor and 2 for IFP
Register contexts	Single	Dual: preview and snapshot	Dual: preview and snapshot
Lens shading correction	5 x 4 zones	10 x 8 zones	10 x 8 zones



## MT9V112: Developer Guide Feature Set Comparison

**Table 1: Feature Set Comparison: MT9V111, MT9M111, and MT9V112 (Continued)**

Feature	MT9V111	MT9M111	MT9V112
Demosaicing and interpolation	Classic	Bilinear or classic	Next generation  No user action needed to achieve highest quality interpolation at full resolution
Image resize	Decimation	Filtered, with subpixel resolution	Filtered, with subpixel resolution
Zoom window	1 set of zoom registers	2 sets of zoom (field of view) registers: context A and context B	1 set of zoom registers (common field of view for context A and context B)
Aperture correction clipping	–	Supports clipping of small aperture correction values, can help reduce noise under low light	No longer supported. User must set: R6:1[12] (R0x106[12]) = 0
Defect correction	2D and 3D correction; defect table inaccessible	2D and 3D correction; defect table readable and writable	2D correction
Color correction	3 x 3 matrix	3 x 3 matrix	3 x 3 matrix
Gamma correction	Single table	Dual table, context switchable	Dual table, context switchable
Output format context	Single context	Dual contexts	Dual contexts
Output format bypass mode	10-bit Bayer	10-bit Bayer, 8 + 2 Bayer	10-bit Bayer, 8 + 2 Bayer
Output format (YCbCr)	Supported	Supported	Supported
Output format (RGB)	565, 555, 444x, x444	Processed Bayer, 565, 555, 444x, x444	Processed Bayer, 565, 555, 444x, x444
Output pixel flow leveling	No output FIFO	Output FIFO	No output FIFO
Special Effects	Monochrome, sepia (using the CCM)	Monochrome, sepia, negative, solarize1, solarize2	Monochrome, sepia, negative, solarize1, solarize2
Auto Exposure	Smooth adaptation; pre- or post-gamma luma driven	Both fast and smooth adaptation; pre-gamma luma driven	Both fast and smooth adaptation; pre-gamma luma driven
Manual white balance	Supported through full custom CCM	Supported through full custom CCM	Supported through full custom CCM or through whitebalance position register
Auto white balance	No flash matrix support	Flash matrix support	Flash position support
Flash	No flash matrix support	Flash matrix CCM	Flash position register
Flicker	Detection and avoidance supported	Detection and avoidance supported	Detection and avoidance supported
Camera control	No sequencer; vertical blanking synchronized flash state	Snapshot (with or without flash) through parameterized sequencer; vertical blanking synchronized context change	Snapshot (with or without flash) through parameterized sequencer; vertical blanking synchronized context change
Sensor power mode	Full-power mode only	Low-power mode (for preview) and full-power mode (for snapshots)	Full-power mode only
Standby/output enable control	Separate STANDBY and OE# pins	Separate STANDBY and OE# pins. Serial interface control of both standby and output enable functions	Combined STANDBY/OE# pin. Serial interface control of both standby and output enable functions



# MT9V112: Developer Guide Functional Overview

**Table 1: Feature Set Comparison: MT9V111, MT9M111, and MT9V112 (Continued)**

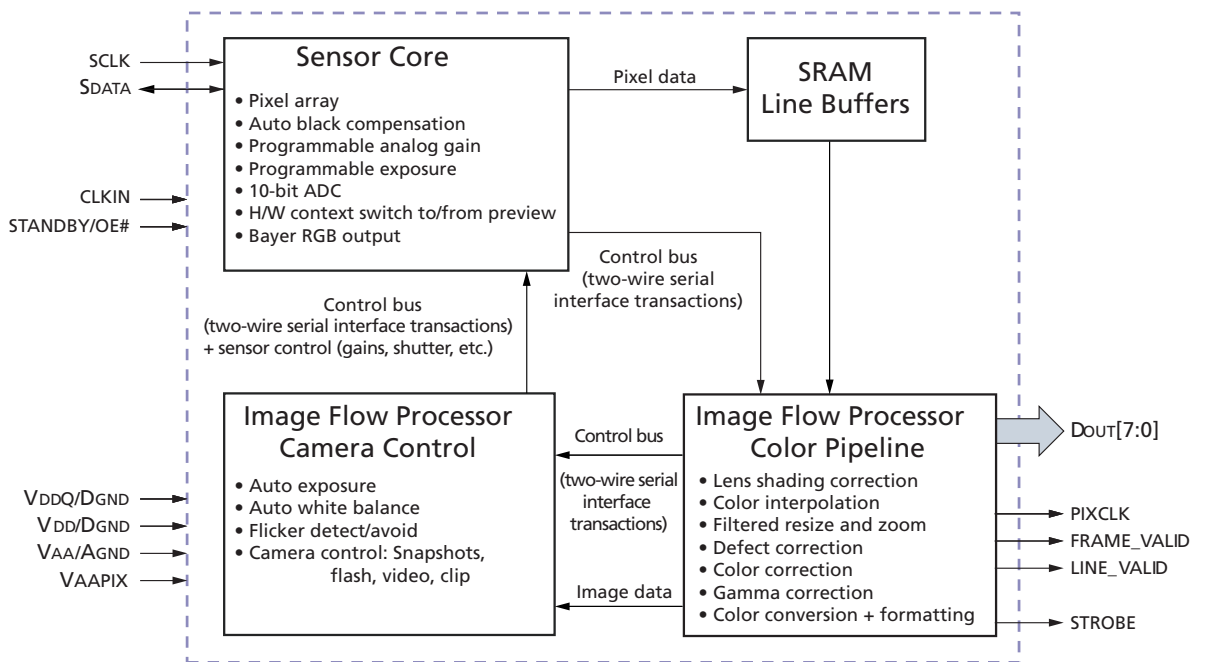
Feature	MT9V111	MT9M111	MT9V112
Standby latency	Following assertion of hard or soft standby wait two master clocks before turning master clock off	Following assertion of hard or soft standby wait 24 master clocks before turning master clock off	Following assertion of hard or wait a row time before turning master clock off
Upsize (interpolating) zoom	Zoom 2x only (for example, from QVGA zoom window to VGA output window)	Not supported	Not supported

## Functional Overview

The MT9V112 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, as shown in Figure 1.

**Figure 1: MT9V112 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 STANDBY 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.





## Internal Architecture

Internally, the MT9V112 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 2.

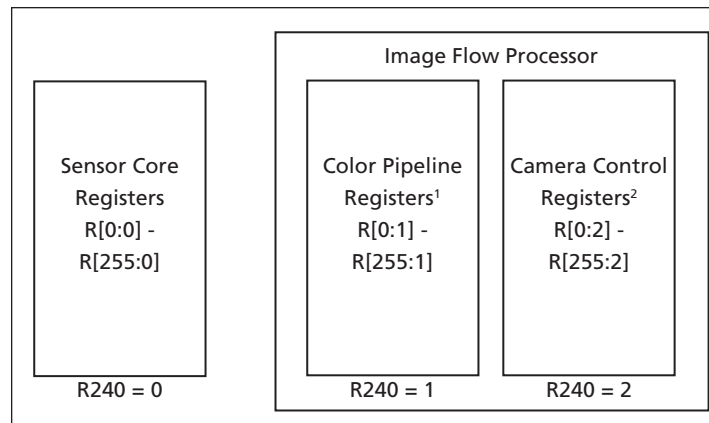
The MT9V112 accelerates mode-switching with hardware-assisted context switching, and supports taking snapshots, flash snapshots, and video clips using a configurable sequencer.

The MT9V112 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 standard FV/LV video interface with gated pixel clocks.

**Figure 2: Internal Registers Grouping**



- Notes:
1. Base configuration, lens shading, resize, output format
  2. AE, AWB, flicker, camera control sequencer



## Two-Wire Serial Interface

### Serial Bus Description

Registers are written to and read from the MT9V112 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 MT9V112 through the serial data (SDATA) line. The SDATA line is typically pulled up to VDDQ 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:

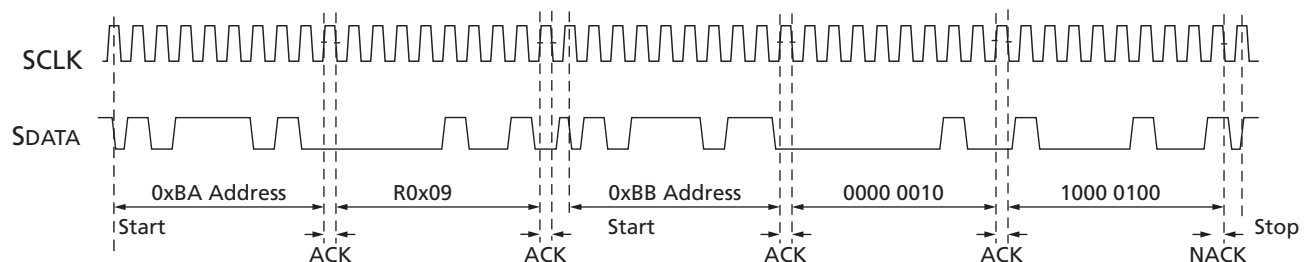
- Start bit
- (No) acknowledge bit
- 8-bit message
- Stop bit
- Slave device 8-bit address

SADDR and R13:0[10] (R0x00D[10]) are used to select between two different addresses in case of conflict with another device. If SADDR XOR R13:0[10] (R0x00D[10]) is LOW, the slave address is 0x90; if SADDR XOR R13:0[10] (R0x00D[10]) is HIGH, the slave address is 0xBA.

When accessing internal registers through the two-wire serial interface, select the desired address space by programming the R240 (R0x0F0) register with 0x90 or 0xBA.

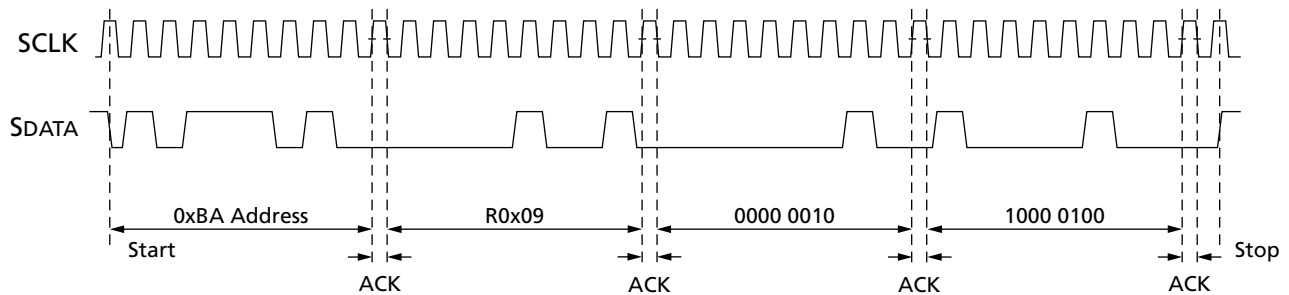
Refer to the data sheet for generic two-wire serial interface examples. Two specific examples are shown in Figure 3 and Figure 4 on page 11).

**Figure 3: 16-bit MT9V112 Example: Read Timing from R0x009, Returned Value 0x0284**





**Figure 4: 16-bit MT9V112 Example: Write Timing to R0x009, Value Written 0x0284**



## Address Pages

The SOC, sensor core, and camera control addresses are accessed through the two-wire serial interface, as shown in Table 2.

## Page Location

A single page register defines the currently open page for all accesses; this register is located at address R240 (R0x0F0, R0x1F0, and R0x2F0) in all three pages.

Throughout this document, registers are qualified by the page where they are located using the conventions shown in Table 2.

**Table 2: Page Location Conventions (Same Page, Different Notations)**

Convention	Example	Reference
R0x<hex_address>:<page #>	R0x1A:1	Hex address 0x1A on page 1
R<decimal_address>:<page #>	R26:1	Decimal address 26 on page 1

## Device ID

Device ID can be read from two addresses in the sensor core page: R0:0 (R0x000) and R255:0 (R0xFF).

## Contexts

This sensor 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 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)



## 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 R200:0 (R0x0C8) that are specific to the sensor core. Table 3 provides context definitions.

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

R200:2—0x2C8 - Global Context Control (R/W)	
Default	0x0000
Description	Defines sensor and color pipe contexts for current frame. Context A is typically used to define preview, while context B is typically used for snapshots. The bits of this register <i>directly</i> control the respective functions. Be careful when writing to this register to avoid a bad frame. See "Context Selection" on page 12.
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 R200 (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. Always write as 0.
Bit 13	Reserved. Always write as 0.
Bit 12	Reserved: Always write as 0.
Bit 11	Reserved: Always write as 0.
Bit 10	Resize/zoom context. Switch resize/zoom contexts: 0: Context A. 1: Context B.
Bit 9	Output format Control 2 context. See R58:1 (R0x13A) and R155:1 (R0x19B). 0: Context A. 1: Context B.
Bit 8	Gamma table context. R83:1 (R0x153)–R88:1 (R0x158) and R220:1 (R0x1DC)–R225:1 (R0x1E1). 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 R35:0 (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 ITU-R BT.656 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.


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

<b>R200:2—0x2C8 - Global Context Control (R/W)</b>	
Bit 3	Sensor read mode context. See R33:0 (R0x021) and R32:0 (R0x020). 0: Context A. 1: Context B.
Bit 2	LED flash: 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 R200:0 (R0x0C8). A WRITE to all bits of R200:0 (R0x0C8) will result in R200:1 (R0x1C8) and R200:2 (R0x2C8) being updated appropriately; a READ from R200:0 (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 R204:2 (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 R200:2 (R0x2C8) through R211:2 (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 R202:2 (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, R211:2 [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

The user has considerable freedom in the use of contexts, which allows a generic naming convention. As stated earlier, one typical usage model is to define context A as preview or 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 R200:0 (R0x0C8), R200:1 (R0x1C8), or R200:2 (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 Micron's local FAE for further information.



## Preview and Snapshot Modes

The sensor core is typically configured to generate VGA frames at 30 fps for both preview and snapshot applications, that is, the sensor read mode context is typically left at context A at all times.

### Preview Mode

QVGA (320 x 240) preview images are generated at up to 30 fps. The reduced-size images are generated by a scaling down operation. The sensor typically outputs a VGA-size image to the color pipe in both context A and context B.

### Snapshot Mode

VGA (640 x 480) images are generated at up to 30 fps. Snapshot mode is typically selected by setting R200:2[10] (R0x2C8[10]) = 1 and selecting resize/zoom context B.

### Switching Modes

The most direct method of switching to full resolution or snapshot mode is achieved by writing R200:2 (R0x2C8) = 0x870B. This restarts the sensor and configures most functions to context B state, leaving others in context A state. Bit 15 of R200:2 (R0x2C8) must be set to restart the sensor during switching.

Following this WRITE, a READ from R200:1 (R0x1C8) or R200:2 (R0x2C8) results in 0x070B being read. The most significant bit (MSB) is cleared automatically by the sensor.

Following a WRITE of 0x870B to R200:2 (R0x2C8), a READ from R200:0 (R0x0C8) results in 0x000B being read, as only the lower four bits of R200:n (R0xnC8) and the restart MSB are implemented in the sensor core.

To return to the default or preview mode, WRITE 0x8000 to R200:2 (R0xC8).

## Camera Control

With the camera control and built-in fast AE adaptation, the MT9V112 can deliver and fully automate the following functions:

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

Both semi-automatic 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 only sees the final snapshot frame.

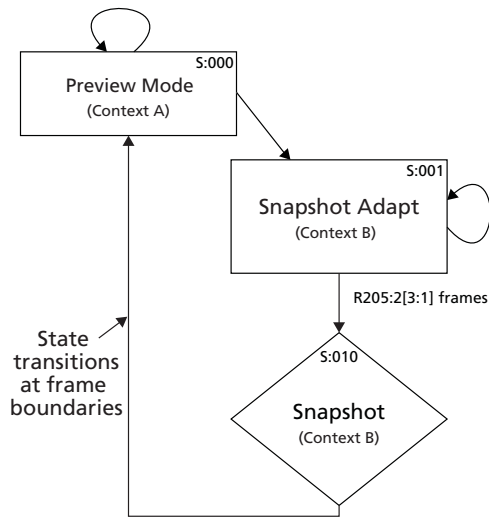
The snapshot picture size can be changed using the zoom control context setting. The snapshot output pixel format is also programmable using context control settings.

Flow charts for flash snapshot (LED and Xenon) and nonflash snapshot programming are provided in Figures 5, 6, and 7, starting on page 16.

For further information, contact Micron's local FAE.

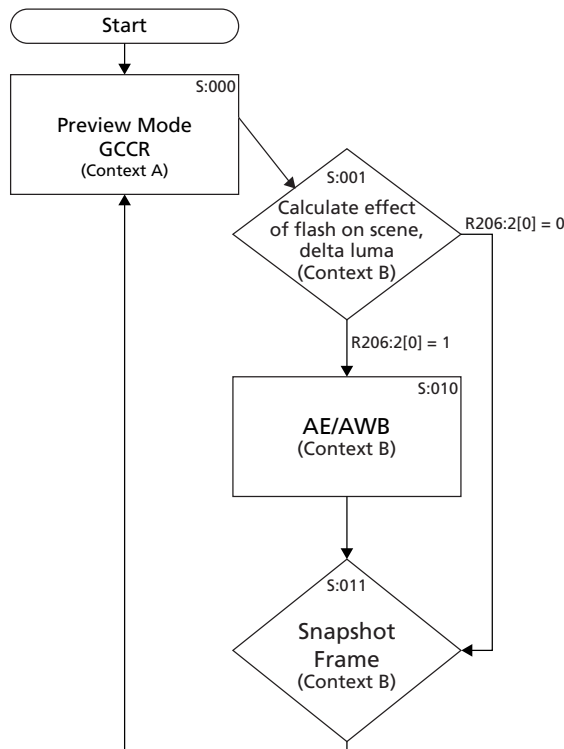


**Figure 5: Nonflash Snapshot Program**



Notes: 1. R204:2[3:0] (R0x2CC)[3:0] = "0000"; program advance = 1.

**Figure 6: LED Flash Snapshot Program**

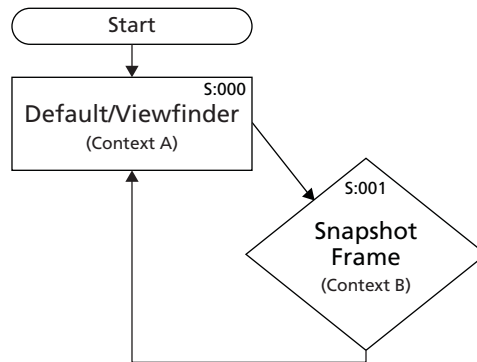


Notes: 1. R204:2[3:0] (R0x2CC)[3:0] = "0001"; program advance = 1.





**Figure 7: Xenon Flash Snapshot Program**



- Notes:
1. R204:2[3:0] (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 R204:2[3:0] (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 R205:2 (R0x2CD), then set R203:2[0] (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 R213:2[0] (R0x2D5[0]) (the user camera control context control register [UCCCR]) and R211:2 (R0x2D3) (UGCCR), then setting the R203:2[0] (R0x2CB[0]) to trigger the operation.

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

### 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 R213:2 (R0x2D5) (UCCCR) and R200:2 (R0x2C8) (GCCR). In this mode, the host must poll R202:2 (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.

To achieve VGA images at 30 fps, CLKIN must be in the 24–27 MHz range. The sensor divides CLKIN by two to generate PIXCLK. The device defaults assume a 24 MHz CLKIN and a 12 MHz PIXCLK. The minimum CLKIN frequency is 2 MHz.

## Primary Operating Modes

Other sensors can run in both full power and reduced power modes. There is no low-power mode in the MT9V112. For both preview and snapshot applications, the MT9V112 is typically run with the configurations shown in Table 4.

**Table 4: Typical Context Settings**

Context Function	Context A
Sensor read mode settings:	No skipping.
Sensor pixel clock:	12 MHz for 24 MHz master clock giving rise to a maximum pixel rate of 12 Mp/s (13.5 Mp/s if using the maximum master clock).
Maximum frame rate:	For 24–27 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, indicated by the “:0” after each register address. Sensor-timing-dependent registers are shown in Table 5.

**Table 5: Sensor-Timing-Dependent Registers**

Register	Maximum	Default	Minimum	Function
R4:0 (R0x004)	640	–	–	Pixel window width
R3:0 (R0x003)	480	–	–	Pixel window height
				<b>Context A</b>
R7:0 (R0x007)	–	203	132	Horizontal blanking (units of sensor clock)
R8:0 (R0x008)	–	11	R34:0[2:0]+1	Vertical blanking (number of rows, including black rows)
R33:0 (R0x021)	–	0x0400	–	Read mode control Oversize and show border controls are not context switchable.
				<b>Context B</b>
R5:0 (R0x005)	–	203	132	Horizontal blanking (units of sensor clock)
R6:0 (R0x006)	–	11	R34:0[2:0]+1 (R0x022[2:0]+1)	Vertical blanking (number of rows, including black rows)
R32:0 (R0x020)	–	0x0300	–	Read mode control [8]: Oversize [9]: Show border



## **Border Pixels**

In this family of sensors, the sensor core adds four border pixels all the way around the image. Active image size is 648 x 488 at full resolution.

## **Blanking Calculations**

Consult Micron's local FAE for a spreadsheet to assist with calculating line times. Refer to the MT9V112 data sheet for more information.



## 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 the line rate does not change when a mode changes, exposure time 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 registers such that the full-frame time is the same from mode to mode. For example, ensure that for 60Hz flicker avoidance, the full-frame time is  $1/30^{\text{th}}$  of a second for both contexts.

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

## Context Switching

The absence of multiple power modes means that the user is less likely to switch the sensor context when changing between preview and snapshot modes.

However, if skip mode is used for preview (for example, to increase frame rate), then the line rate for preview mode is approximately half that of snapshot (full resolution) mode. To avoid exposure changes, either ensure common line rate through appropriate horizontal blanking settings or adjust full-frame time registers as mentioned above.

## Switching Modes

Initiate mode switches from preview (context A) to snapshot (context B) during vertical blanking; switching should be accompanied by a sensor restart. R200:2[15] (R0x2C8) must be written as "1" when changing contexts.

## 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.

Alternate methods are supported by an internal sequencer. These additional methods are particularly useful for taking flash snapshots.



### Power-Saving Modes

The MT9V112 supports a low power consumption standby state. There are two methods for placing the device in this state:

1. Assert STANDBY (typically known as “hard standby”).
2. Use the two-wire serial interface to disable analog circuitry and gate clocks (typically 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 (CLKIN) 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 R13:0[1] (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 STANDBY results in entry to the standby state only if R13:0[7] (R0x00D[7]) = 0 (default). Refer to Figure 10 on page 26 for details.

Following assertion of STANDBY, wait an entire row time before turning off the master clock. Calculation of row time is part of setting up the sensor’s timing (blanking registers).

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

Do not assert STANDBY during reset until at least 10 master clock cycles have passed.

#### Leaving Hard Standby

De-assert STANDBY.

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

#### Soft Standby

To place the sensor core into soft standby, set R13:0[3:2] (R0x00D[3:2]) = 1.



## MT9V112: Developer Guide Exposure and Sensor Context Switching

When the two-wire serial interface transaction has completed, wait one more row time before turning off the master clock. Calculation of row time is part of setting up the sensor's timing (blanking registers).

### Leaving Soft Standby

Set R13:0[3:2] (R0x00D[3:2]) = 10.

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

### Output Enable Control

The MT9V112 provides both hardware and software methods for putting the following pads into the High-Z state: DOUT[7:0], FV, LV, PIXCLK, and STROBE. The ability to automatically put the pads in High-Z state when entering standby is also available.

Table 6 details the output enable control signals.

**Table 6: MT9V112 Output Enable Control**

RESET#	Standby (Pin)	R13:0[4] (R0x00D[4]) (OUTPUT_DIS)	R13:0[6] (R0x00D[6]) (DRIVE_SIGNALS)	Output State
0	0	N/A	N/A	Driven
0	1	N/A	N/A	High-Z
1	0	0 (default)	0 (default)	Driven
1	1	0 (default)	0 (default)	High-Z
1	0	1	0 (default)	High-Z
1	1	1	0 (default)	High-Z
1	Don't care	0 (default)	1	Driven
1	Don't care	1	1	High-Z

### Avoiding Floating Inputs

Many MT9V112 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.

1. A floating input may cause unexpected results from the sensor.
2. 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.

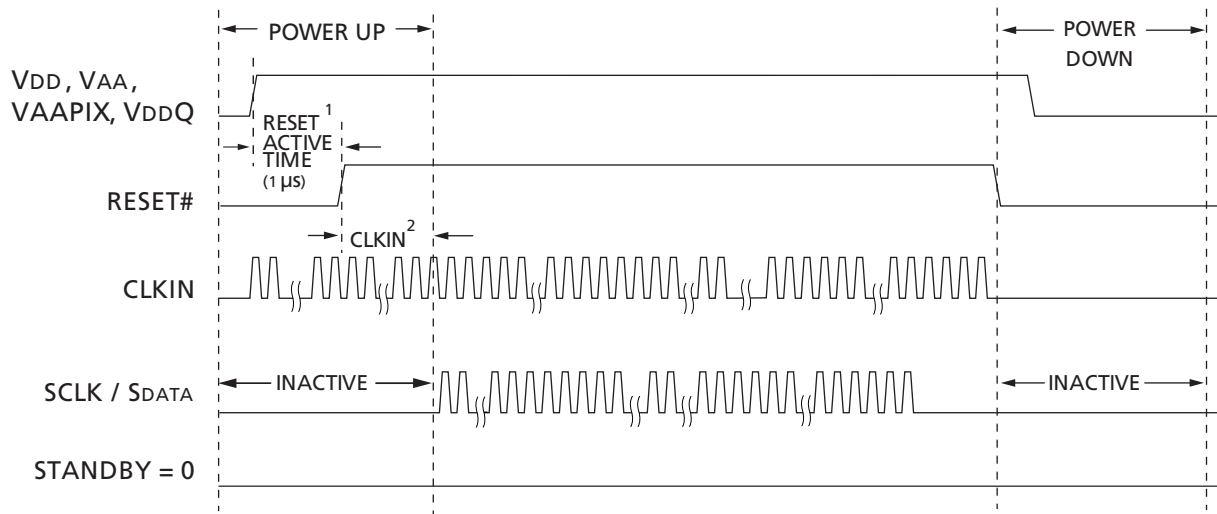
**Note:** When using the output-enable control, ensure that all MT9V112 bidirectional pads are driven to a valid logic level before they enter a High-Z state.

This leads to the following recommended usage models:

- Single camera systems  
Do not place the output pin in High-Z during standby mode. There is no power saving benefit and it may lead to floating device input buffers.
- Dual camera systems  
Ensure that one camera is driving the shared bus at all times. If both cameras are in standby, one camera must still be driving the bus.






**Figure 9: Power On/Off Sequence: MT9V112**


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

## Reset

The reset pin must be asserted for a minimum of 1 $\mu$ 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 the MT9V112, with the first rising clock edge during reset, outputs settle to the following:

- FV = LV = PIXCLK = 0 (LOW)
- DOUT[7:0] = random (each output pin retains the state prior to reset)

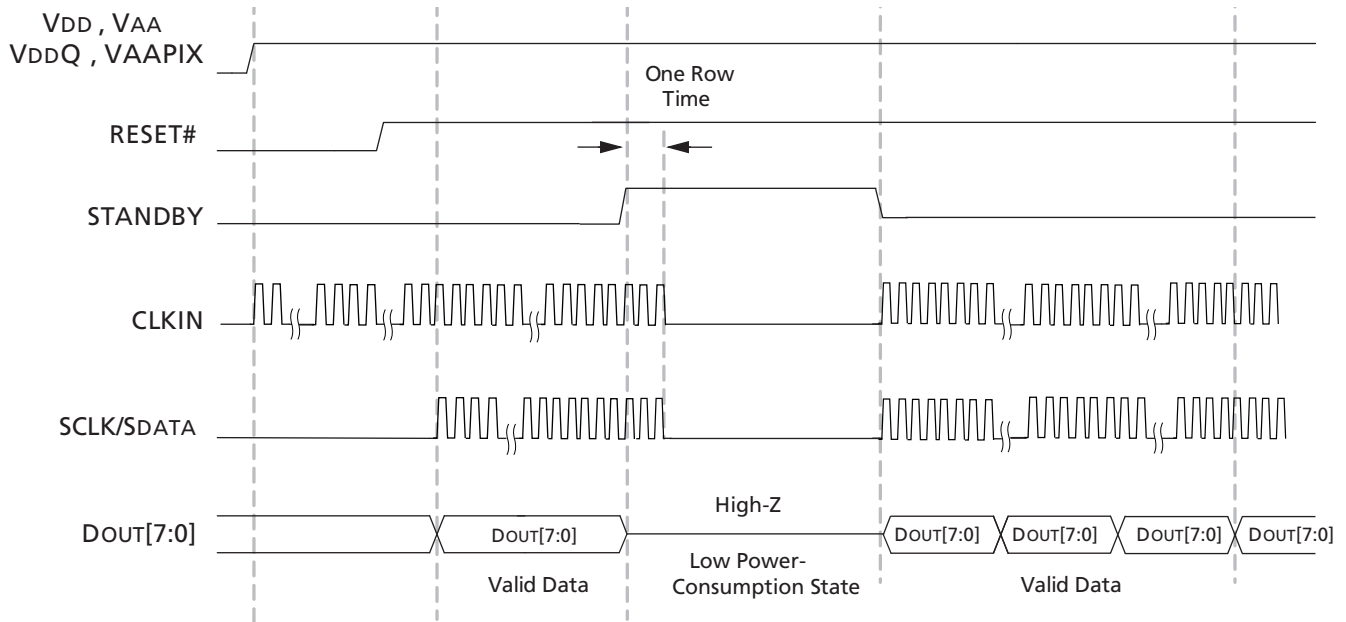
Following de-assertion of reset, wait 10 CLKIN cycles before considering the device active and applying serial interface transactions or toggling STANDBY.



## Timing Summary

The timing summary for the MT9V112 is shown in Figure 10 and in Figure 11 on page 27.

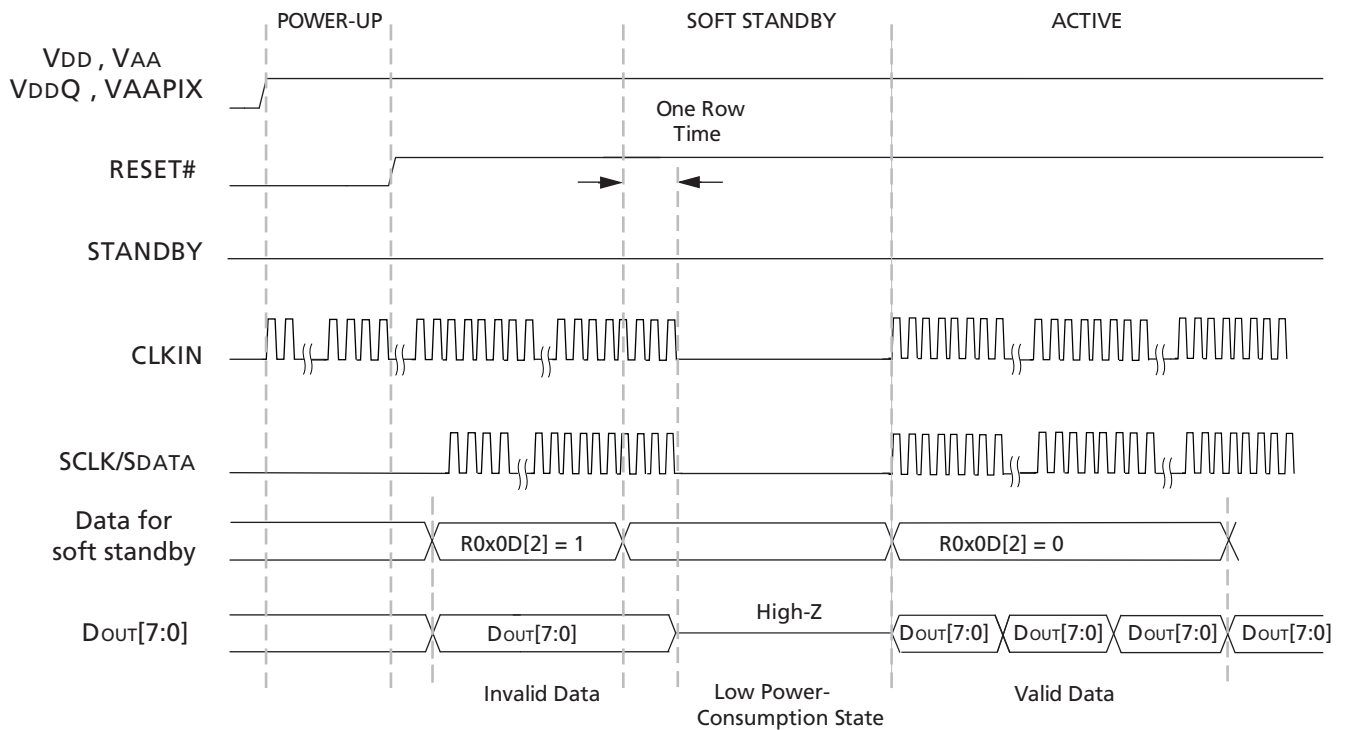
**Figure 10: Hard Standby Operation Waveform**



- Notes:
1. In the hard standby mode (RESET# = 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 CLKIN 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.



Figure 11: Soft Standby Operation Waveform



- Notes:
1. Soft standby is asserted/de-asserted by a two-wire serial interface to R13:0[2] (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 CLKIN 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 MT9V112. 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 7.

**Table 7: 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. Zoom width $\geq$ output width Sensor width $\geq$ output width	Column skip enabled. (Zoom width)/2 $\geq$ output width (Sensor width)/2 $\geq$ output width
Output height	Output resolution Y dimension	Row skip disabled. Zoom height $\geq$ output height Sensor height $\geq$ output height	Row skip enabled. (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 MT9V112 stops generating frames; that is, the sensor appears to lock up. If this happens, program register settings that comply with the restrictions detailed in Table 7 and the sensors will once again generate pixel data.

Context A is typically used for preview mode and context B for snapshots. Context B registers are programmed the same way as context A registers.

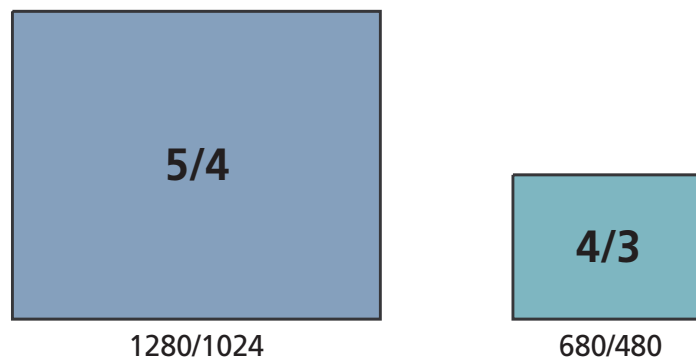
There is just one set of zoom registers in the MT9V112, which results in a common field of view between contexts. This is the more common usage model.



## Aspect Ratios

Maintaining the same aspect ratios in snapshot and preview output sizes and zoom windows helps to avoid squeezing the image more in one dimension than the other (see Figure 12 for more details).

**Figure 12: Aspect Ratios**



We recommend a 4/3 aspect ratio image size for the preview display. More details are available in “Appendix B: MT9V112 Image Resolution Settings” on page 63.

## Zoom

Zoom is available when the output image is less than the sensor resolution. 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: MT9V112 Image Resolution Settings” on page 63, 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 preview image, the greater the zoom range. A 12 x 10-pixel output image supports greater than 100X zoom.

## Advantages of Using the Zoom Window vs. the Sensor Window

The sensor window size can be reduced, as long as the zoom window is also reduced. Lens shading coefficients must be recalculated or turned off if the sensor window is changed. It is much easier to use the zoom window.

The MT9V112 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 30 for additional details.

## Automatic Zoom

The typical zoom usage model uses the manual zoom settings described in the previous section. However, the MT9V112 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 8.



The R175:1 (R0x1AF[1]) zoom in and zoom out bits behave like optical zoom controls. Setting the zoom-in decrements the internal zoom window by the values in R174:1 (R0x1AE[1]) 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 R174:1 (R0x1AE[1]) 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. It stops zoom-out before it exceeds 640 x 512.

Automatic zoom does not change the values in zoom window registers. The current internal zoom window can be read from R171:1 (R0x1AB[1]) and R172:1 (R0x1AC[1]).

See “Snapshot Following Zoom and/or Pan” on page 30 for detailed information.

**Table 8: Settings for Automatic Zoom**

Register	Value	Register	Comment
R165:1 (R0x1A5)	0x4000	Reducer horizontal pan—context A (R/W)	Horizontal pan register A with centered origin
R168:1 (R0x1A8)	0x4000	Reducer vertical pan—context A (R/W)	Vertical pan register A with centered origin
R175:1 (R0x1AF)	0x0100		Zoom in
	0x0200		Zoom out
	0x0300		Zoom in and out (see above)

## 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.

## Snapshot Following Zoom and/or Pan

The MT9V112 has just one set of zoom registers. The device automatically uses the same field of view for snapshots and preview, as shown in Table 9.

**Table 9: Pan and Zoom Registers for Snapshots from Preview Field-of-View**

Copy This Register	Masked With	Comment	To This Register	Comment
R165:1 (R0x1A5)	0xffff	Horizontal pan in resize A	R159:1 (R0x19F)	Horizontal pan in resize B
R168:1 (R0x1A8)	0xffff	Vertical pan in resize A	R162:1 (R0x1A2)	Vertical pan in resize B
R171:1 (R0x1AB)	0x07ff	Current zoom in X	R160:1 (R0x1A0)	Horizontal zoom in resize B
R171:1 (R0x1AB)	0x07ff	Current zoom in X	R161:1 (R0x1A1)	Horizontal output size B
R172:1 (R0x1AC)	0x07ff	Current zoom in Y	R163:1 (R0x1A3)	Vertical zoom in resize B
R172:1 (R0x1AC)	0x07ff	Current zoom in Y	R164:1 (R0x1A4)	Vertical output size B



## 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 10 for details).

**Table 10: Aperture Correction Settings**

Register	Value	Comment
R05:1 (R0x105)	0x0007	Aperture scale maximum value.
R06:1 (R0x106)	Bit 12 = 0	Aperture correction algorithm has been revised and no longer supports clipping of small aperture corrections. Ensure R06:1[12] = 0 at all times.
R175:1 (R0x1AF)	0x0000	Optimal interpolation used at all times. No user intervention necessary.
R58:1 (R0x3A)	Bit 8 = 1	Output format 2A: Mode = RGB, which is slightly sharper than YCbCr.
R155:1 (R0x19B)	Bit 8 = 1	Output format 2B: Mode = RGB, which is slightly sharper than YCbCr.

### Auto Sharpen

Auto sharpen R5:1[3] (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 R51:2 (R0x233) has two gain thresholds, as shown in Table 11. 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 11: Aperture Scale Values**

Gain Zone	Equivalent Aperture Scale Value	Aperture Scale Factor	Comment
0	4	100%	Gain < R51:2 [7:0] (R0x233[7:0])
1	3	75%	Else if gain < R51:2 [15:8] (R0x233[15:8])
2	2	50%	Else

### 2D Defect Correction

The MT9V112 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 correction softens the image slightly.

For the MT9V112, 2D defect correction is enabled by setting R6:1[13] (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<sup>th</sup> of a second for 60Hz flicker avoidance, or 1/25<sup>th</sup> of a second for 50Hz flicker



avoidance. The default values for these registers are calculated based on the default line time achieved with a 27 MHz clock. See “Clocks and Blanking” on page 19 for more details.

**Table 12: Flicker Avoidance Registers**

Register	Function	MT9V112 Default
R87:2 (R0x257)	AE frame size for 60Hz Context A	(1/30) / 66.83 $\mu$ s = 499 (0x1F3)
R88:2 (R0x258)	AE frame size for 50Hz Context A	(1/25) / 66.83 $\mu$ s = 599 (0x257)
R89:2 (R0x259)	AE frame size for 60Hz Context B	(1/30) / 66.83 $\mu$ s = 499 (0x1F3)
R90:2 (R0x25A)	AE frame size for 50Hz Context B	(1/25) / 66.83 $\mu$ s = 599 (0x257)

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)

AE 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 R6:1[14] (R0x106[14]).





## Controlling AE Speed and Sensitivity

AE speed and sensitivity are controlled by R47:2 (R0x22F) in context A, and by R156:2 (R0x29C) in context B. As the format of these two registers is the same, the R47:2 (R0x22F) register is referenced for convenience.

### Hysteresis Control

The first AE field R47:2[2:0] (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 AE is within range of the target luminance.
- A value of “7” results in only 1/128th of the current luminance being blended into the previous luminance value.

Current luminance can be read as a 16-bit quantity in register R76:2[15:0] (R0x24C)[15:0]). The blended or time-averaged luminance can be read as an 8-bit quantity in register R77:2[7:0] (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, R47:2[4:3] (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

R47:2[8:6] (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

R47:2[14] (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 R47:2[13:12] (R0x22F[13:12]). 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 R38:2 (R0x226) and the AE vertical boundaries register R39:2 (R0x227). The little window is defined by the AE center horizontal boundaries register R43:2 (R0x22B) and the AE center vertical boundaries register R44:2 (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 R6:1[3:2] (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 leftmost (horizontal) or topmost (vertical) boundary of the image.
- A value of 128 (0x80) is interpreted as the rightmost (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  $1/16^{\text{th}}$  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 13.

**Table 13: AE Zones**

Zone	Description
Zone 0:	High illumination. Exposure controls are the sensor core shutter width R9:0 (R0x009) and shutter delay R12:0 (R0x00C).
Zones 1–23:	Moderate to low illumination. Exposure controls are the sensor core shutter width R9:0 (R0x009) and the color channel gains R43:0 (R0x02B), R44:0 (R0x02C), R45:0 (R0x02D), and R46:0 (R0x02E).
Zone 24:	Low to very low illumination. Exposure controls are the sensor core color channel gains R43:0 (R0x02B), R44:0 (R0x02C), R45:0 (R0x02D), and R46:0 (R0x02E), the analog-to-digital converter (ADC) voltage references R65:0 (R0x041), and the IFP digital gains R98:2 (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 R6:1 (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 R55:2 (R0x237) to bind it to a single value. The current value of the AE zone is read from R63:2 (R0x23F).

The shutter width in zones 1–24 are set to be a multiple of a base integration time corresponding to  $1/120^{\text{th}}$  of a second (for 60Hz flicker abatement), or  $1/100^{\text{th}}$  of a second (for 50Hz flicker abatement). The base integration times are established in registers R87:2 (R0x257) through R90:2 (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 R57:2[10:0] (R0x239[10:0]) holds the row size (height in pixels) for context A. The AE line size B register R58:2[10:0] (R0x23A[10:0]) holds the row size for context B.

The AE shutter delay limit A register R59:2[10:0] (R0x23B[10:0]) holds the maximum shutter delay value for context A. Likewise, the AE shutter delay limit B register R60:2[10:0] (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] (R0x236[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 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 (R130:2 (R0x282)–R149:2 (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 R54:2 (R0x236), the AE zone limits register R55:2 (R0x237), and the AE table gain range control register R56:2 (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 R54:2[15:8] (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 V<sub>REF</sub> DACs register R65:0 (R0x041). The low voltage reference (V<sub>REFLO</sub>) is in R65:0[3:0] (R0x041[3:0]), and the high voltage reference (V<sub>REFHI</sub>) is in R65:0[7:4] (R0x041[7:4]). The AE ADC gain control register R61:2 (R0x23D) controls the ADC. The minimum limit value for V<sub>REFLO</sub> is R61:2[11:8] (R0x23E[11:8]). The minimum limit value for V<sub>REFHI</sub> is R61:2[3:0] (R0x23E[3:0]); the maximum limit value for V<sub>REFHI</sub> is R61:2[7:4] (R0x23E[7:4]).

#### Caution

Although this register is listed here for completeness, we do not recommend modifying the value of register R61:2 (R0x23D), both to ensure the stability of the AE algorithm and to prevent possible damage to the sensor.

After V<sub>REFHI</sub> R65:0[7:4] (R0x041[7:4]) is reduced to the value in R61:2[3:0] (R0x23D)[3:0], digital gains in the IFP must be taken into account.



### Digital Gain1

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 R103:2[7:0] (R0x267[7:0]). The current value of digital gain1 can be found in the lower byte of the AE digital gain monitor register R98:2[7:0] (R0x 262[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 Gain2

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 R103:2[15:8] (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 R98:2[15:8] (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 R9:0 (R0x009) exceeds the number of lines of the image R3:0 (R0x003) plus border rows R32:0[8] (R0x029[8]) plus vertical blanking R8:0 (R0x008) and R6:0 (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.

**Table 14: 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 R87:2 (R0x257)–R90:2 (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  $1/16^{\text{th}}$  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  $1/16^{\text{th}}$  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 R54:2[7:0] (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 R56:2[7:0] (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 R56:2[11:8] (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 37, the zone transition where the frame rate is decreased is a function of the number of frame timing settings. R56:2[11:8] (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 R46:2 (R0x22E), and the half range of the target luminance is in the high byte of register R46:2 (R0x22E). As long as the time-averaged luminance low byte of R77:2 (R0x24D) is in the range of R46:2[7:0] (R0x22E[7:0]) ± R46:2[15:8] (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 R70:2 (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 R46:2[7:0] (R0x22E[7:0]) + R46:2[15:8] (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, R70:2[7:0] (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 R48:0[9:0] (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, R59:1[9:0] (R0x13B0 (black level subtraction), should be changed so that the difference between R48:0[9:0] (R0x030[9:0]) and R59:1[9:0] (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 15 for the exposure target register R46:2[7:0] (R0x22E[7:0]) would be used.

**Table 15: Exposure Settings Values for the Exposure Target Register R46:2[7:0] (R0x22E[7:0])**

F#	Calculation	Result (Decimal)	Result (Hex)	Setting
f/0.7	$74 \times 2$	148	0x94	+2
f/0.8	$74 \times \text{sqrt}(2)$	105	0x69	+1
f/1	$74 \times 1$	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 MT9V112 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 R55:2 (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 Micron 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:

- Set R55:2 (R0x237) to 0x100 (AE zone 8)
- Set R130:2 (R0x282) to 0x03FC
- Read R46:2 (R0x22E) to save to local variable
- Set R46:2 (R0x22E) to 0x00 (forcing AE to move)
- Read R63:2 (R0x23F). If the result is greater than eight (AE has not moved into the legal range), wait 50ms (or one frame) and repeat
- Restore R46:2 (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:

- Set R55:2 (R0x237) to 0x300 (AE zone 24)
- Set R130:2 (R0x282) to 0x03F9
- Read R46:2 (R0x22E) to save to local variable
- Set R46:2 (R0x22E) to 0x00 (forcing AE to move)
- Read R63:2 (R0x23F). If the result is equal to eight (AE has not moved higher), wait 50ms (or one frame) and repeat
- Restore R46:2 (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:

- Set R55:2 (R0x237) to 0x200 (AE zone 16)
- Set R130:2 (R0x282) to 0x03FC
- Read R46:2 (R0x22E) to save to local variable
- Set R46:2 (R0x22E) to 0x00 (forcing AE to move)
- Read R63:2 (R0x23F). If the result is equal to eight (AE has not moved higher), wait 50ms (or one frame) and repeat
- Restore R46:2 (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 R60:1 (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 R60:1 (R0x13C) and R101:2 (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 R83:1 (R0x153)–R88:1 (R0x158) and R220:1 (R0x1DC)–R225:1 (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 R52:1 (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 16 provides values for different brightness levels using scheme 4 by setting the low order 8 bits of R52:1 (R0x134). Table 16 lists both the low order 8-bit value and the entire range of the 16 bits of R52:1 (R0x134).

**Table 16: Scheme 4 Brightness Settings**

Brightness Level	R52:1 (R0x134) RGB (High 8 bits)	R52:1 (R0x134) YCbCr (Low 8 bits)	R52:1 (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 MT9V112, contrast control is achieved through adjustment of exposure control. See “Exposure Control” on page 39 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 13 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 13: AWB Sensor Controls**

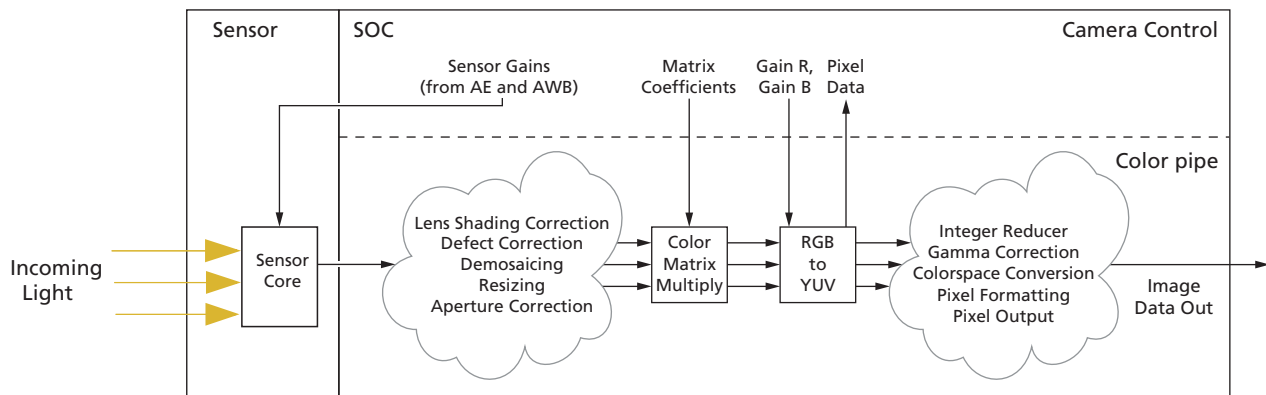
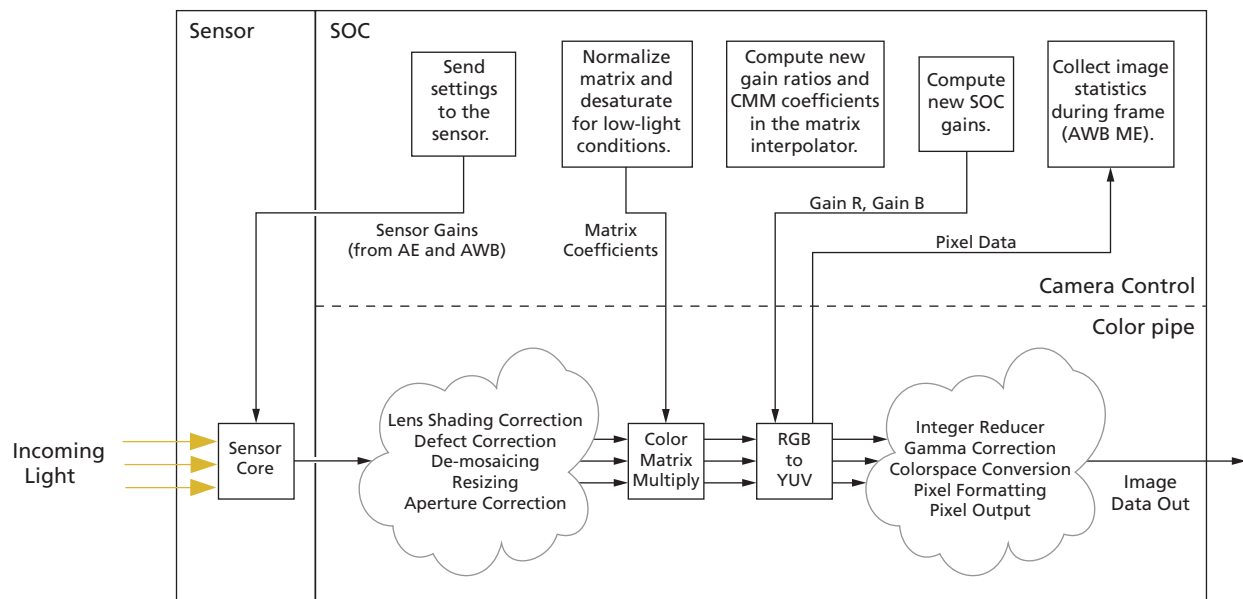


Figure 14 on page 43, expands upon this and adds the processing flow that occurs in the white balance unit between frames. Throughout a frame, the AWB ME collects data from the pixel stream and computes a set of statistics used by the other white balance computations. At the end of a frame, white balance 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 14: AWB Flow Processing**


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

- Gray world mode R42:2 (R0x22A) = 0x0000, R31:2 (R0x21F) = 0x3f, R32:2 (R0x220) = 0xff00
- Real-life mode (power-up default plus set R42:2(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 48. 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 R36:2 (R0x224) to the desired matrix position. For example, if the chosen matrix position is 0x23, program R36:2 (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 R18:2 (R0x212). Table 17 provides useful matrix position examples.

**Table 17: Suggested Matrix Positions for Varying Conditions**

Lighting	Matrix Position Read From R18:2 (R0x212)	Matrix Position Constraints R36:2 (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 R18:2 (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 R6:1[1] (R0x101[1]) = 0 and R6:1[15] (R0x106[15]) = 1.
  - 3b. Read the matrix position to check that the matrix has moved.
  - 3c. Return to AWB mode: Set R6:1[1] (R0x101[1]) = 1 and R6:1[15] (R0x106[15]) = 0.

See “AWB Speed and Stability” on page 46 for further information on accelerating CCM movement.

### Full Manual White Balance

To set all the white balance color pipe controls for a specific preset lighting condition:

1. Load the base matrix with the desired white-balance matrix. This consists of coefficient signs in register R2:2 (R0x202), coefficient scale factors in R3:2 (R0x203) and R4:2 (R0x204), and scaled coefficients from registers R9:2 (R0x209) through R17:2 (R0x211). The scale factors do not need to be reloaded at each matrix change if they remain consistent across all matrices in use.



2. Load the base sensor gain ratios R94:2 (R0x25E) with the desired ratios for this white balance mode.
3. Load the manual red/blue gain register, R34:2 (R0x222), with the desired gains for the red and blue channels in the IFP. The value in each field is an 8-bit number in 1.7 fixed point representation; 0x80 represents unity gain.
4. Switch to manual white balance mode: Set R6:1[1] (R0x106)[1] = 0 to disable AWB, and R6:1[15] (R0x106[15]) = 1 to enable manual WB and load the matrices into the hardware.

### Optional Approach to Manual White Balance

After step 3 above, enabling both AWB and manual WB (that is, R6:1[1] (R0x106[1]) = 1 and R6:1[15] (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 R18:2 (R0x212) and the red and blue IFP gains R19:2 R0x213) and R20:2 (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:

- R40:2[6:3] (R0x228[6:3]), the frequency of matrix adjustments
- R40:2[2:0] (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 R6[15] (R0x006[15]) = 1. When manual white balance is subsequently turned off by clearing R6[15] (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 R2:2 (R0x202)–R17:2 (R0x211) and R21:2 (R0x215)–R30:2 (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 MT9V112. A new control bit has been added, R40:2[12] (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. R40:2[12] (R0x228[12]) is auto-cleared once the CCM has taken effect.

To change the active CCM during auto and manual white balance modes for MT9V112:

1. Load new matrices into the CCM registers.
2. Assert R40:2[12] (R0x228[12]).

**Note:** The CCM scale factors R3:2 (R0x203) and R4:2 (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 Micron FAE can provide assistance with this.



## 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 R206:2[13:12] (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, R246:2 (R0x2F6).

## Effects Settings

To achieve different effects, use the settings for the effects hardware provided in Table 18.

**Table 18: Effects Settings**

Effect	R226:1[2:0] (R0x1E2[2:0])	R227:1[15:0] (R0x1E3[15:0])
Black and 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 R226:1[15:8] (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 MT9V112 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 R6:1[10] (R0x106[10]). The LC registers are from R128:1 (R0x180) through R149:1 (R0x195) and from R182:1 (R0x1B6) through R196:1 (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 Criteria

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. Ignore violations due to noise when judging whether these criteria are met.

### 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.
4. Click **Browse**, and browse to the starting **.ini** file. Open it, click **New**, and enter a name for the section where the curves will be saved. Click **OK** and then **Save**.

While developing curves, periodically open the **.ini** file, highlight the new section and click **Save**. This saves the curves to allow backtracking on adjustments. The LC curves will be registers from R128:1 (R0x180) through R149:1 (R0x195) and from R182:1 (R0x1B6) through R196:1 (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 high-lighted for another module, that section will be overwritten.

## Begin Vertical Calibration

1. Click the **control** icon at the top of the screen to open the Sensor Control window.
2. In the window, click on **Lens Correction**.
3. In the **Lens Correction** window, select **Enable Lens Correction**, set Scale = 3 Low, and set Corners = 0 Off.  
This sets R128:1 (R0x180) = 3. **Scale** is *kd* or *Kx*. **Corners** is *kp* or *Kxy*. See the MT9V112 data sheet for details on register descriptions.
4. If another scale setting is to be used, restart the LC calibration procedure from the beginning.
5. Click on the **graphs** icon at the top of the main screen (the Analysis Graph window will open).
6. Select the **Intensity** tab on the Analysis Graph window.
7. Click on the **info** icon at the top of the main screen.



8. At the bottom left of the main screen, under **Mouse selection**, select **Column**, then move the cursor to column 320 (half of the snapshot horizontal resolution) in the image and click on it. The cursor position can be read from the **Property/Value** area of the main screen.  
The values for column 320 are now plotted in the Analysis Graph.
9. Select one color channel—R, G, or B—to calibrate. The other channels must be deselected. Selecting and deselecting is done by clicking in the appropriate **Red**, **Green**, or **Blue** box in the Lens Correction/Sensor Control window.
10. Select **Vertical** to complete vertical calibration for that channel before doing anything with another channel.
11. Click the **Auto-Adjust** box and watch the sliders for the knees change and the display in the Analysis Graph flatten out across the window.
12. Repeat step 9 through step 11 for each of the other channels.

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

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

### Vertical Calibration – Initial Value



1. Deselect **Enable Lens Correction** on the Lens Correction window.
2. On the Analysis Graph, note the selected channel's peak value when LC is turned off. Call this value *P*.
3. In this section, set the Initial Value for the selected channel so that with LC on, the first pixel in the column has value *P*.
4. Right-click on the Analysis Graph and click **Lens Correction Guides**.  
A vertical guide line now indicates the location of each knee. Knees are locations where the slope of the selected column plot on the Analysis Graph is altered. Because the image is resized to 640 x 480, *Knee 0* is correctly labeled, but *Knee 2* is labeled *Knee 1*. Even-numbered knees are labeled *Knee n* where the knee is actually *Knee 2n*. Odd-numbered knees are not shown, but lie halfway between consecutive even-numbered knees.
5. Turn off LC. Modify R9:0 (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. Do not proceed until this is complete; later, it may not be possible to change this value without extensive rework.

### Vertical Calibration – Knee Values

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



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

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

**Note:** It may be necessary to back up 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 R128:1 (R0x180) to increase values near image edges and corners. This affects all channels. Setting **Corners** too high can cause visible banding.

## Excessive Fall-Off

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

## Cleaning Up the File

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

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



## Gamma Correction

### Selecting a Gamma Factor

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

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

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

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:

$$C_{out} = \left( \left( \left( \left( C_{in} \right)^{y_1} \right)^{y_2} \right)^{y_3} \right)^{y_4} = (C_{in})^{y_1 y_2 y_3 y_4} \quad (\text{EQ 2})$$

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 MT9V112 gamma correction unit, consider the role of the sensor in the image processing chain. Two gamma tables are provided in the MT9V112 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 MT9V112 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 MT9V112 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

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

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

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

The +16 is added to Y through the luma offset register, R52:1[15:8] (R0x134).

## Calculating Gamma Table

By default, the MT9V112'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

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

$$Y_{max} = \begin{cases} 224 & \text{ITU601/656YCbCr or RGB output} \\ 255 & \text{JPEG-friendly YCbCr output} \end{cases} \quad (\text{EQ 8})$$

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

Calculated values are programmed into R83:1 (R0x153)–R88:1 (R0x158) for the context A gamma table, and into R220:1 (R0x1DC)–R225:1 (R0x1E1) for the context B gamma table.



## MT9V112: Developer Guide Gamma Correction

The luma offset register R52:1 (R0x134) and the luma clip register R53:1 (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 19 illustrates appropriate values for these registers.

**Table 19: Output Register Values**

Output Format	Luma Offset R52:1 (R0x134)	Luma Clip R53:1 (R0x134)
RGB	[15:8] = 0	n/a
ITU-601/656 YCbCr	[7:0] = 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 MT9V112 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 MT9V112 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 shade 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:

- Set R72:1 (R0x148) = 0x0087 to turn on the color bars, and set WB digital gains to 1.0
- Set R6:1[1] (R0x106[1]) = 0 to turn off AWB
- Set R6:1[4] (R0x106[4]) = 1 to bypass the color correction matrix
- Set R6:1[7] (R0x106[7]) = 0 to turn off auto flicker detection
- Set R6:1[10] (R0x106[10]) = 0 to turn off lens-shading correction
- Set R6:1[14] (R0x106[14]) = 0 to turn off AE

### Ramp Test Pattern

The output format register for each context can output a ramp test pattern. R58:1[5:4] (R0x131[5:4]) controls the ramp test output for context A while R155:1[5:4] (R0x19B[5:4]) controls the ramp test output for context B.

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

$$\text{If } R58:1[8] \text{ (R0x13A[8]) or } R155:1[8] \text{ (R0x19B[8]) for context B} = 0, \quad (\text{EQ 9})$$

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

$$\text{If } R58:1[8] \text{ (R0x13A[8]) or } R155:1[8] \text{ (R0x19B[8]) for context B} = 1,$$

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

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 RGB565 mode because pixel\_position 28 is equivalent to b11100; bit 2 is a 1, so R and B get rounded up to 32 (which is equivalent to b100000); bit 1 is a 0, so G remains at or gets rounded down to 28 (which is equivalent to b11100).



## 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. 59)
2. Can adaptation speed be slowed to match the device response speed? (p. 59)
3. Can adaptation speed be slowed further? (p. 59)
4. How can the AWB decisions be biased to produce warmer or cooler images? (p. 59)

### Auto White Balance (AWB)

5. AWB occasionally gets stuck. What should be done when this happens? (p. 59)
6. When should the color correction matrix be calibrated? (p. 60)
7. How should color correction matrices be calibrated? (p. 60)
8. What can be done about pictures that come out with the wrong colors? (p. 60)
9. What can be done when the white balance is incorrect for a test color chart? (p. 60)
10. Is there a difference between AWB windows in preview and snapshot modes? (p. 60)

### 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. 60)
12. The sensor response looks flat when LC is off. Can a design be based on this sensor? (p. 60)

### 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. 61)
14. Why does the sensor response look flat with LC on? (p. 61)
15. Are there special requirements for changing to and from preview mode? (p. 61)
16. Why does a change of mirror mode cause lens shading? (p. 61)
17. What timing constraints apply when the host or application changes LC curves? (p. 61)
18. Coloration or shading changes as the multi-position lens moves. Is something wrong? (p. 61)
19. Are manufacturing tolerances of concern? (p. 61)

### Miscellaneous FAQs

20. Is there color suppression in saturated areas? (p. 62)
21. Is there color suppression in low-light conditions? (p. 62)
22. I've attempted to change sensor gains, but there is no change. (p. 62)
23. How is contrast enhanced or improved? (p. 62)



## Auto Exposure (AE)

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

The speed of AE adaptation is controlled by registers R47:2 (R0x22F) context A and R156:2 (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 R47:2[13:12] (R0x22F[13:12]) or R156:2[13:12] (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 R47:2 (R0x22F) or R156:2 (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, R47:2[2:0] (R0x22F[2:0]) or R156:2[2:0] (R0x29C[2:0]) up to a maximum value of 7
- Increase the frame interval during which AE attempts a change, adjust R47:2[4:3] (R0x22F[4:3]) or R156:2[4:3] (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 R76:2 (R0x24C) towards the target luminance R46:2[7:0] (R0x22E[7:0]) by setting the value of R47:2[8:6] (R0x22E[8:6]) or R156:2[8:6] (R0x29C[863]) to "0"

**Note:** While it might be tempting to set the width of the AE target luminance window R46:2[15:8] (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 R46:2[7:0] (R0x22E[7:0]) divided by R46:2[15:8] (R0x22E[15:8]) should be maintained when adjusting R46:2[7:0] (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 R243:2 (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 R42:2 (R0x22A). R42:2 (R0x22A) = 0 configures the SOC to make the picture gray at all times. If this produces insufficient results, change the settings to R31:2 (R0x1F) = 0 and R32:2 (R0x21F) = 51200 (0xC800). Switch AWB to gray-world mode. See "AWB Overview" on page 42 for details.



## MT9V112: Developer Guide

### Appendix A: Imaging Developer Guide FAQs

#### 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 Micron 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 MT9V112, 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 42 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 49 for a discussion on calibrating parameters for the lens and module.

Lens shading is to be expected, even with a good quality lens. The MT9V112 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 Micron 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 49 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 MT9V112 decreases color saturation for very bright and overexposed pixels. Color suppression is activated when pixel values exceed a relative threshold. R37:1[2:0] (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 MT9V112 automatically decreases color saturation in low-light conditions. R62:2[12] (R23E[12]) = 1 enables automatic saturation reduction (default).

Registers R51:2 (R0x133) and R62:2 (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.

R51:2[7:0] (R0x133[7:0]) and R62:2[9:8] (R0x23E[9:8]) specify the 75-percent setting threshold; R51:2[15:8] (R0x133[15:8]) and R62:2[11:10] (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 R43:0 (R0x02B)–R46:0 (R0x02E), integration time R9:0 (R0x009), R12:0 (R0x00C), and ADC voltage reference R41:0 (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 R6:1[14] (R0x106[14]) = 0, AWB R6:1[1] (R0x106[14]) = 0, forced matrix loading R6:1[11] (R0x106[11]) = 0, and flicker detection R6:1[7] (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 MT9V112, contrast enhancement should be dealt with by the host.

While the MT9V112 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.



## MT9V112: Developer Guide Appendix B: MT9V112 Image Resolution Settings

### Appendix B: MT9V112 Image Resolution Settings

**Table 20: Aspect Ratio Zoom Window Settings**

View Finder Output	Aspect Ratio	Full Field of View	Zoom Range	Zoom Window		Output Size A		Output Size B		Zoom Step
				X (max) R166:1 (R0x1A6)	Y (max) R169:1 (R0x1A9)	X R167:1 (R0x1A7)	Y R170:1 (R0x1AA)	X (max) R161:1 (R0x1A1)	Y (max) R164:1 (R0x1A4)	R174:1 (R0x1AE)
640 x 480	4:3	640 x 480	1.00	640	480	640	480	640	480	0x0C09
352 x 264	4:3	640 x 480	1.82	640	480	352	264	640	480	0x0C09
320 x 240	4:3	640 x 480	2.00	640	480	320	240	640	480	0x0C09
240 x 180	4:3	640 x 480	2.67	640	480	240	180	640	480	0x0C09
208 x 156	4:3	640 x 480	3.08	640	480	208	156	640	480	0x0C09
176 x 132	4:3	640 x 480	3.64	640	480	176	132	640	480	0x0C09
160 x 120	4:3	640 x 480	4.00	640	480	160	120	640	480	0x0C09
120 x 90	4:3	640 x 480	5.33	640	480	120	90	640	480	0x0C09
88 x 66	4:3	640 x 480	7.27	640	480	88	66	640	480	0x0C09
80 x 60	4:3	640 x 480	8.00	640	480	80	60	640	480	0x0C09
40 x 30	4:3	640 x 480	16.00	640	480	40	30	640	480	0x0C09
586 x 480	11:9	586 x 480	1.00	586	480	586	480	586	480	0x0B09
352 x 288	11:9	586 x 480	1.66	586	480	352	288	586	480	0x0B09
320 x 261	11:9	586 x 480	1.83	586	480	320	261	586	480	0x0B09
240 x 196	11:9	586 x 480	2.44	586	480	240	196	586	480	0x0B09
208 x 170	11:9	586 x 480	2.82	586	480	208	170	586	480	0x0B09
176 x 144	11:9	586 x 480	3.33	586	480	176	144	586	480	0x0B09
160 x 130	11:9	586 x 480	3.66	586	480	160	130	586	480	0x0B09
120 x 98	11:9	586 x 480	4.88	586	480	120	98	586	480	0x0B09
88 x 72	11:9	586 x 480	6.66	586	480	88	72	586	480	0x0B09
80 x 65	11:9	586 x 480	7.33	586	480	80	65	586	480	0x0B09
40 x 32	11:9	586 x 480	14.653	586	480	40	32	586	480	0x0B09



## Appendix C: Glossary of Terms

**Table 21: Glossary of Terms**

Term	Definition
1.3-megapixel resolution	1280 x 1024
ADC	Analog-to-digital converter
Asymptotic	Approaching a value or curve arbitrarily closely
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> .
Bitmap	An image containing only raster information
BMP	A bitmap format
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).
DNP	Dai Nippon printing. Woman with flower is image #5, ITE flesh tones chart <a href="http://www.dnp.co.jp/semi/e/test/index.html">www.dnp.co.jp/semi/e/test/index.html</a>
DRAM	Dynamic random access memory
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 a way to change how an image will look on different displays.
GCCR	Global context control register
GIF	Graphics interchange format
Hysteresis	Hysteresis describes a system's tendency to not react instantly to adjustments made to a given value. Rather, the system reacts only when it has achieved a minimum threshold set by the user. For example, luma target.
I/F	Interface
IFP	Image flow processor. Performs color recovery and correction, sharpening, gamma correction, lens shading correction, and on-the-fly defect correction
Interlaced	The graphic data is split (usually into two parts), and is displayed alternately line by line.
Interlacing	Also known as progressive display. GIF, JPG and TIFF have supported this feature since about 1990.
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.





## MT9V112: Developer Guide Appendix C: Glossary of Terms

**Table 21: Glossary of Terms (Continued)**

Term	Definition
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
NTSC	National television standards committee. The North American standard (525-line interlaced raster-scanned video) for the generation, transmission, and reception of television signals
Output resolutions	Includes, but is not limited to, VGA, QVGA, CIF, and QCIF
PAL	The mainly European standard of displaying analog television signals. It consists of 625 horizontal lines of resolution at 50Hz. See also NTSC. Acronym stands for phase alternating line, phase alternation by line or for phase alternation line
PDA	Personal digital assistant
Pixels	A short name for pixel element. The smallest part that can be displayed on a monitor.
PNG	Portable network graphics. A file format.
Progressive scan	An image sensor that gathers its data and processes each scan line one after another in sequence. Compare to interlaced.
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.
RST	Reset
SADDR	Sensor address for two-wire serial interface
SCLK	Serial clock for two-wire serial interface
Sensor core	In the MT9V112, it consists of a pixel array of 648 x 488 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.
TIFF	Tagged image file format
True Color	24-bit color, 16,777,216 colors
UGCCR	User global context control register
VGA resolution	Video graphics array; Supports a maximum resolution of 640 x 480 pixels in 16 colors out of a palette of 262,144 colors.
VREF	Voltage reference



## Contacting Micron

When contacting Micron, 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 Micron. 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.



8000 S. Federal Way, P.O. Box 6, Boise, ID 83707-0006, Tel: 208-368-3900

prodmktg@micron.com www.micron.com Customer Comment Line: 800-932-4992

**Micron, the M logo, and the Micron logo are trademarks of Micron Technology, Inc. All other trademarks are the property of their respective owners.**

**Advance: This document contains initial descriptions of products still under development.**



## MT9V112: Developer Guide Revision History

---

### Revision History

Rev. A .....	11/17/2006
<ul style="list-style-type: none"><li>• Initial release</li></ul>	