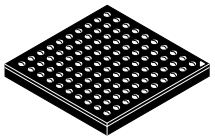
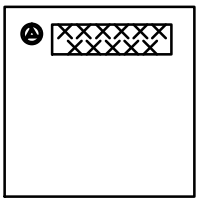
AP0201AT High-Dynamic Range (HDR) Image Signal Processor (ISP)

AP0201AT



VFBGA100, 7x7
CASE 138AH

MARKING DIAGRAM



XXXXXXXXXXX = Laser Marking

ORDERING INFORMATION

See detailed ordering and shipping information on page 2 of this data sheet.

GENERAL DESCRIPTION

onsemi's AP0201AT Image Signal Processor (ISP) is optimized for use with HDR (High Dynamic Range) sensors. The AP0201AT provides full auto−functions support (AWB and AE) and Adaptive Local Tone Mapping (ALTM) to enhance HDR images and advanced noise reduction which enables excellent low−light performance.

Table 1. KEY PERFORMANCE PARAMETERS

Parameter	Value	
Image Sensor Interfaces	Parallel and HiSPi	
Input Data Format	Parallel: 12−bit SDR (linear) or 12−bit HDR companded HiSPI: 12−bit SDR (linear) or 12/14−bit HDR companded	
Output Interface	Ethernet−MII, RMII, GMII	
Output Format	H.264, MJPEG	
Maximum Resolution	1920×1080 (2.0 Mp)	
Input Clock Range	10−29 MHz	
Maximum Frame Rate	1080p30, 960p45 and 720p60 assumes 1 clock per pixel mode. 2 clock per pixel reduces the frame rate	
Output Ethernet Data Rate	MII: 100 Mb/S RMII: 100 Mb/s GMII: 1 Gb/S at 2.5 V or higher IO voltage	
Supply Voltage	V_{DD}IO_S	1.8 or 2.8 V nominal
	V_{DD}IO_H	1.8 or 2.8 or 3.3 V nominal
	V_{DD}_REG	1.8 V nominal
	V_{DD}	1.2 V nominal
	V_{DD}_PLL	1.2 V nominal
	V_{DD}_PHY	2.8 V nominal
	V_{DD}IO_OTPM	2.5 to 3.3 V nominal
Operating Temperature	−40°C to +105°C (ambient) −40°C to +125°C (junction)	
Power Consumption	159 mW (Note 1)	

1. Refer to Tables 22 and 23 for operating currents.

Features

- Up to 2.0 Mp (1920 x 1080) **onsemi** Sensor Support
- 30 fps at 1080 p, 45 fps at 1.2 Mp, 60 fps at 720p (Optimized for Operation with HDR Sensors)
- Color and Gamma Correction
- Auto Exposure, Auto White Balance, 50/60 Hz Auto Flicker Detection and Avoidance
- Adaptive Local Tone Mapping (ALTM)
- Configurable through Low−cost SPI Flash and EEPROM Devices
- Up to 7 GPIO
- Fail−Safe IO
- Multi−Camera Synchronization Support
- MJPEG Encoding (8−bit)
- H.264 Encoding (8 and 10 bit intra−frame)
- Integrated Full−duplex Ethernet MAC
- Precise Timing Protocol (PTP): IEEE 802.1AS and 1588−2008
- IEEE 802.1Qav (Annex L Configurable Video Bandwidth)
- AVB (IEEE1722) and RTP Video Transport Protocol
- **onsemi** Custom UDP−based Protocol
- IPv4, IPv6 (specific usage), TCP, DHCP, QoS and ICMP4 Support
- Proxy Service for Customer Specific Protocol
- Metadata over Ethernet
- Hybrid Mode Operation: Configuration over Serial Interface and Video over Ethernet
- AEC−Q101 Qualified and PPAP Capable

Applications

- Surround, Rear and Front View Cameras
- Blind Spot/Side Mirror Replacement Cameras
- Automotive Viewing/Processing Fusion Cameras

ORDERING INFORMATION

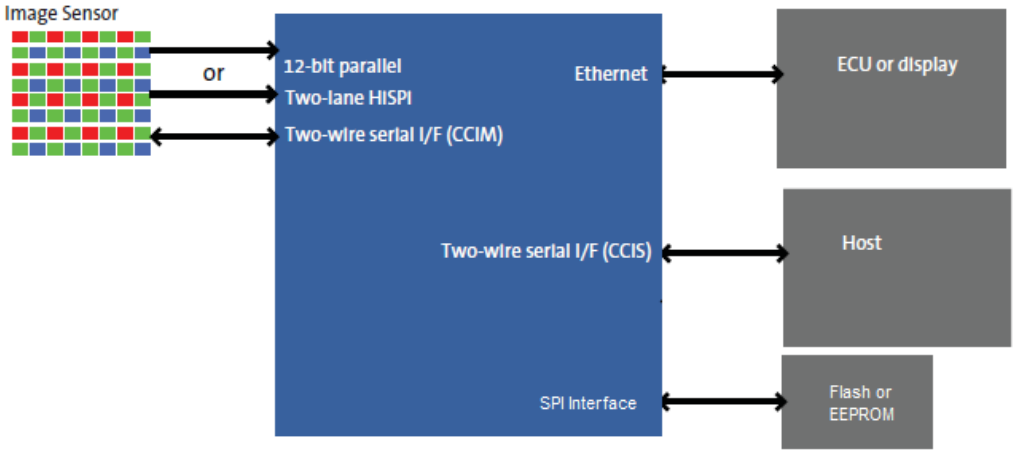
Table 2. AVAILABLE PART NUMBERS

Part Number	Product Description	Orderable Product Attribute Description
AP0201AT2L00XEGA0−DR	Ethernet Co−Processor, 100−ball VFBGA	Drypack
AP0201AT2L00XEGA0−TR	Ethernet Co−Processor, 100−ball VFBGA	Tape & Reel†
AP0201AT2L00XEGAD3−GEVK	AP0201AT Demo Kit	
AP0201AT2L00XEGAH3−GEVB	AP0201AT Head Board	

†For information on tape and reel specifications, including part orientation and tape sizes, please refer to our Tape and Reel Packaging Specifications Brochure, BRD8011/D.

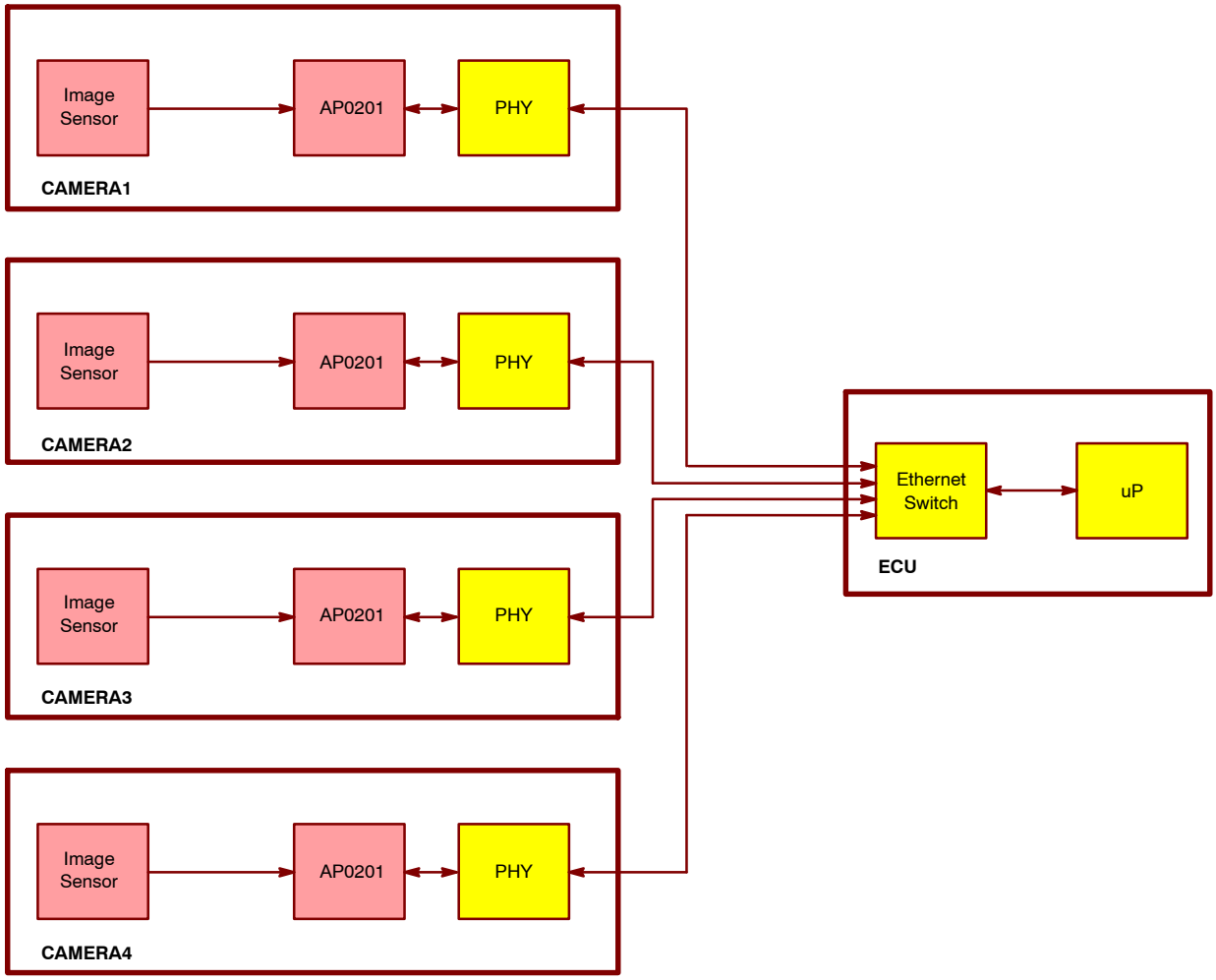
FUNCTIONAL OVERVIEW

Figure 1 shows the typical configuration of the AP0201AT in a camera system. On the host side, commands and image data are sent out over the Ethernet bus. The AP0201AT interface to the sensor supports a parallel interface or HiSPi interface.



NOTE: The Hybrid mode supports configuring the AP0201AT through the serial interface and streaming video over Ethernet. Hybrid Mode can be enabled by loading an available patch. Please contact **onsemi** support.

Figure 1. AP0201AT Connectivity



NOTE: The AP0201AT example above shows the PHY which is used between the Ethernet switch and the AP0201AT.

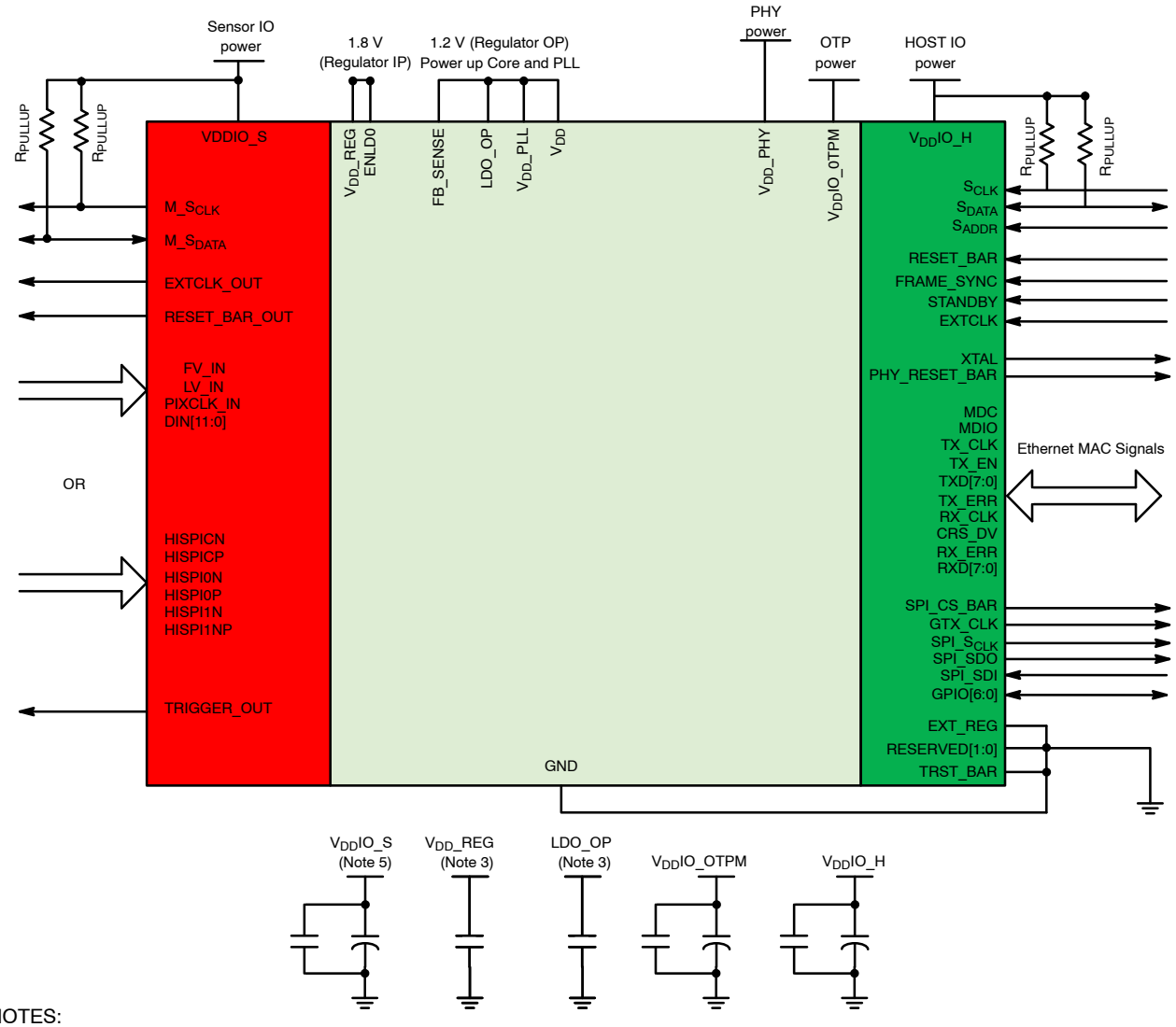
Figure 2. Example AP0201AT Connectivity

SYSTEM INTERFACES

Figure 3 shows typical AP0201AT device connections.

All power supply rails must be decoupled from ground using capacitors as close as possible to the package.

The AP0201AT signals to the sensor and host interfaces can be at different supply voltage levels to optimize power consumption and maximize flexibility. Table 3 provides the signal descriptions for the AP0201AT.



NOTES:

1. This typical configuration shows only one scenario out of multiple possible variations for this device.
2. **onsemi** recommends a 1.5 kΩ resistor value for the two−wire serial interface $R_{PULL-UP}$.
3. The decoupling capacitors for the regulator input and output should have a value of 1.0 μF. The capacitors should be ceramic and need to have X5R or X7R dielectric.
4. TRST and RESERVED[0] connect to GND for normal operation, RESERVED[3:2] are floating and RESERVED[1] is connected to VDDIO_H for normal operation.
5. **onsemi** recommends that 0.1 μF and 1 μF decoupling capacitors for each power supply are mounted as close as possible to the pin. Actual values and numbers may vary depending on layout and design consideration.

Figure 3. Typical Ethernet Configuration

HiSPi and Parallel Connection

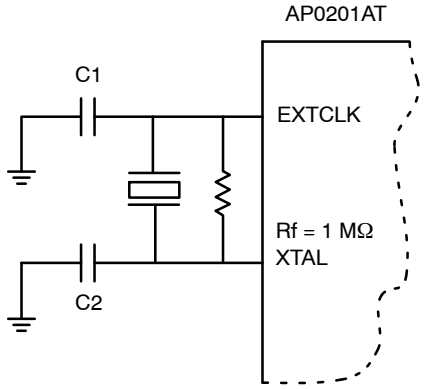
When using the HiSPi interface, connect the parallel interface to GND.

When using the parallel interface, it is recommended for the HiSPi interface to be connected to ground, and the power supply (VDD_PHY) to be connected to +2.8 V. Floating these pins is allowed as well.

Crystal Usage

As an alternative to using an external oscillator, a crystal may be connected between EXTCLK and XTAL. Two small loading capacitors and a feedback resistor should be added, as shown in Figure 4.

For applications above 85°C, **onsemi** does not recommend using the crystal option. A crystal oscillator with temperature compensation is recommended for applications that require this.



NOTE: Rf represents the feedback resistor, an Rf value of 1 MΩ is sufficient for AP0201AT. C1 and C2 are decided according to the crystal or resonator CL specification. In the steady state of oscillation, CL is defined as (C1xC2)/(C1+C2). In fact, the I/O ports, the bond pad, package pin and PCB traces all contribute the parasitic capacitance to C1 and C2. Therefore, CL can be rewritten to be (C1*xC2*)/(C1*+C2*), where C1* = (C1 +C_{IN}, STRAY) and C2* = (C2+$C_{OUT, STRAY}$). The stray capacitance for the IO ports, bond pad and package pin are known which means the formulas can be rewritten as C1* = (C1+1.5 pF+C_{IN}, PCB) and C2* = (C2+1.3 pF+C_{OUT}, PCB).

Figure 4. Using a Crystal Instead of External Oscillator

Table 3. PIN DESCRIPTIONS

Name	Type	Description
EXTCLK	Input	Master input clock, nominally 27 MHz. This can either be a square−wave generated from an oscillator (in which case the XTAL input must be left unconnected) or direct connection to a crystal.
XTAL	Output	If EXTCLK is connected to one pin of a crystal, this signal is connected to the other pin, otherwise this signal must be left unconnected.
RESET_BAR	Input/PU	Asynchronous active−low reset. When asserted, the device will return all interfaces to their reset state. When released, the device will initiate the boot sequence. This signal has an internal pull−up resistor.
FRAME_SYNC	Input	Pass through to TRIGGER_OUT. This signal should be connected to GND if not used. This pin has two modes of use for frame sync. One the pass through to TRIGGER_OUT. The other goes to the frame_sync_monitor block.
STANDBY	Input	Standby mode control, active HIGH.
EXT_REG	Input	Select external regulator if tied high.
ENLDO	Input	Regulator enable (V_{DD}_REG domain).
SPI_SCLK	Output	Clock output for interfacing to an external SPI flash or EEPROM memory.
SPI_SDI	Input	Data in from SPI flash or EEPROM memory. When no SPI device is fitted, this signal is used to determine whether the AP0201AT should auto−configure: 0: Do not auto−configure; Two−wire or ethernet interface will be used to configure the device (host−config mode). 1: Auto−configure. This signal has an internal pull−up resistor.
SPI_SDO	Output	Data out to SPI flash or EEPROM memory.
SPI_CS_BAR	Output	Chip select out to SPI flash or EEPROM memory.

Table 3. PIN DESCRIPTIONS (continued)

Name	Type	Description
EXTCLK_OUT	Output	Clock to external sensor.
RESET_BAR_OUT	Output	Reset signal to external sensor.
M_S_{CLK}	Output	Two−wire serial interface clock (Master).
M_S_{DATA}	I/O	Two−wire serial interface data (Master).
FV_IN	Input	Sensor frame valid input.
LV_IN	Input	Sensor line valid input.
PIXCLK_IN	Input	Sensor pixel clock input.
D_{IN}[11:0]	Input	Sensor pixel data input D_{IN}[11:0].
HiSPiCN	Input	Differential HiSPi clock (negative).
HiSPiCP	Input	Differential HiSPi clock (positive).
HiSPi0N	Input	Differential HiSPi data, lane 0 (negative).
HiSPi0P	Input	Differential HiSPi data, lane 0 (positive).
HiSPi1N	Input	Differential HiSPi data, lane 1 (negative).
HiSPi1P	Input	Differential HiSPi data, lane 1 (positive).
TRIGGER_OUT	Output	Trigger signal for external sensor.
PHY_RESET_BAR	Output	PHY_RESET_BAR Output.
TX_CLK	Output	Host frame valid output.
RX_CLK	Output	Host line valid output.
GTX_CLK	Output	Host pixel clock output. This signal is only used in GMII mode and may be left floating for all other modes.
TXD[7:4]	Output	Ethernet port. TXD[7:4] is only used for Gigabit Ethernet and should be tied to GND for 100 Mbit applications.
TXD[3:0]	Output	Ethernet port.
TX_ERR	Output	Ethernet port.
TX_EN	Output	Ethernet port.
RXD[7:4]	Input	Ethernet port. RXD[7:4] is only used for Gigabit Ethernet and should be tied to GND for 100 Mbit applications.
RXD[3:0]	Input	Ethernet port.
RX_ERR	Input	Ethernet port.
CRS_DV	Input	Ethernet port.
MDC	Output	Management Data clock for controlling the PHY.
MDIO	I/O	Management Data Input/Output for controlling the PHY.
GPIO_[6:1]	I/O	General purpose digital I/O.
TRST	Input	Must be tied to GND in normal operation.
Reserved[0]	Input	Must be tied to GND in normal operation.
V_{DD}IO_S	Supply	Sensor I/O power supply.
V_{DD}IO_H	Supply	Host I/O power supply.
V_{DD}_PLL	Supply	PLL supply.
V_{DD}	Supply	Core supply.
V_{DD}IO_OTPM	Supply	OTPM power supply.
V_{DD}_PHY	Supply	PHY IO voltage for HiSPi.
GND	Supply	Ground.
V_{DD}_REG	Supply	Input to on−chip 1.8 V to 1.2 V regulator.

Table 3. PIN DESCRIPTIONS (continued)

Name	Type	Description
LDO_OP	Output	Output from on chip 1.8 V to 1.2 V regulator.
FB_SENSE	Input	On−chip regulator sense signal.

Table 4. PACKAGE PINOUT

	1	2	3	4	5	6	7	8	9	10
A	Reserved[0]	V_{DD}IO_H	M_S_{DATA}	D_{IN}0	V_{DD}	D_{IN}5	D_{IN}10	LV_IN	V_{DD}IO_S	FV_IN
B	SCLK	GPIO_6	EXTCLK_OUT	M_S_{CLK}	D_{IN}1	D_{IN}4	D_{IN}9	D_{IN}11	HiSPi1N	HiSPi1P
C	SPI_SCLK	Reserved[1] (Note 3)	S_{DATA}	GPIO_5	TRIGGER_ OUT/ GPIO_0	D_{IN}3	D_{IN}8	PIXCLK_IN	HiSPiCN	HiSPiCP
D	SPI_SDO	SPI_SDI	SPI_CS_BAR	SADDR	RESET_ BAR_OUT	D_{IN}2	D_{IN}7	V_{DD}_PHY	HiSPi0N	HiSPi0P
E	V_{DD}	GPIO_1	STANDBY	GND	GND	GND	D_{IN}6	GND	GND	V_{DD}IO_H
F	V_{DD}IO_ OTMP	GPIO_2	GPIO_3	RESET_ BAR	GND	GND	GND	EXTCLK	XTAL	V_{DD}
G	TRST	GPIO_4	FRAME_SYNC	RX_CLK	RXD6	RXD2	TXD5	EXT_REG	ENLDO	V_{DD}_PLL
H	PHY_RESET_ BAR	TX_CLK	MDIO	RX_ERR	RXD4	TX_ERR	TXD6	TXD2	FB_SENSE	V_{DD}_REG
J	GTX_CLK	Reserved[3]	CRS_DV	RXD7	RXD3	RXD0	TXD7	TXD3	TXD0	LDO_OP
K	Reserved[2]	V_{DD}IO_H	MDC	RXD5	V_{DD}	RXD1	TX_EN	TXD4	TXD1	GND

2. Pin K1 and J2 should be left floating.
3. Pin C2 needs to be tied to VDDIO_H for normal Ethernet operation. If flash reprogramming is required, this pin should be tied to ground during reprogramming and then tied back to VDDIO_H for normal operation (NET_MODE).
4. A1 to be tied to ground.

ON−CHIP REGULATOR

The AP0201AT has an on−chip regulator, the output from the regulator is 1.2 V and should only be used to power up the AP0201AT. It is possible to bypass the regulator and provide power to the relevant pins that need 1.2 V. The following table summarizes the key signals when using/bypassing the regulator.

Table 5. KEY SIGNALS WHEN USING THE REGULATOR

Signal Name	Internal Regulator	External Regulator
V_{DD}_REG	1.8 V	Connect to V_{DD}IO_H
ENLDO	Connect to 1.8 V (V_{DD}_REG)	GND
FB_SENSE	1.2 V (input)	Float
LDO_OP	1.2 V (output)	Float
EXT_REG	GND	Connect to V_{DD}IO_H

POWER−UP SEQUENCE

Powering up the AP0201AT requires voltages to be applied in a particular order, as seen in Figure 5. The timing requirements are shown below.

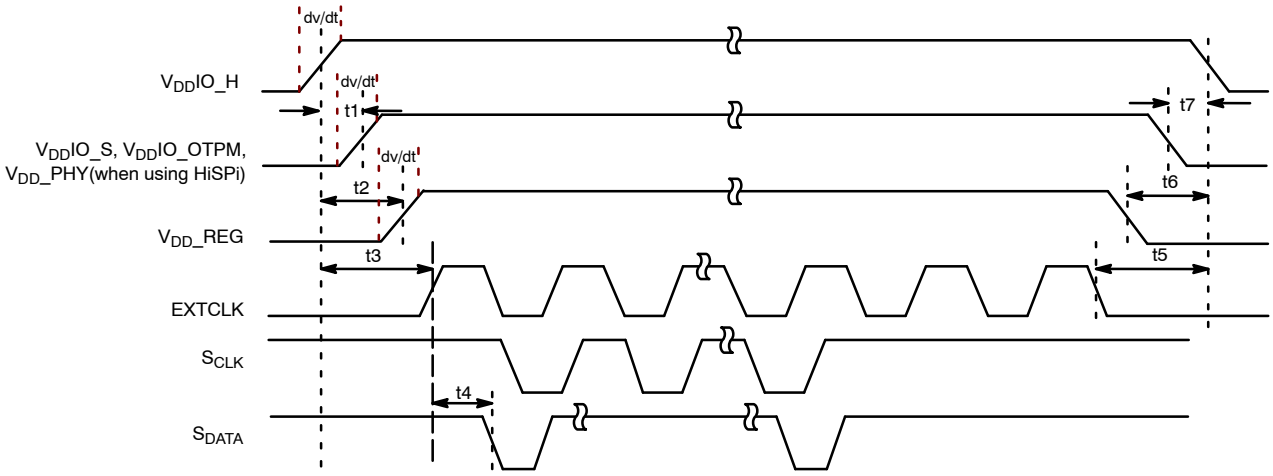


Figure 5. Power−Up and Power−Down Sequence

Table 6. POWER−UP AND POWER−DOWN SIGNAL TIMING

Symbol	Parameter	Min	Typ	Max	Unit
t1	Delay from V_{DD}IO_H to V_{DD}IO_S, V_{DD}IO_OTPM, V_{DD}_PHY (when using HiSPi)	0	−	50	ms
t2	Delay from V_{DD}IO_H to V_{DD}_REG	0	−	50	ms
t3	EXTCLK activation	t1 + t2 + 1	−	−	ms
t4	First serial command (Note 5)	100	−	−	EXTCLK cycles
t5	EXTCLK cutoff	t6	−	−	ms
t6	Delay from V_{DD}_REG to V_{DD}IO_H	0	−	50	ms
t7	Delay from V_{DD}IO_S, V_{DD}IO_OTPM, V_{DD}_PHY (when using HiSPi) to V_{DD}IO_H	0	−	50	ms
dv/dt	Power supply ramp time (slew rate)	−	−	0.1	V/μs

5. When using XTAL the settling time should be taken into account.
6. RESET_BAR can be either high or low at power−up.

RESET AND STANDBY MODES

Reset

The AP0201AT has two types of reset available:

- A hard reset is issued by toggling the RESET_BAR signal.
- A soft reset is issued by writing commands through the Ethernet interface.

Table 7 shows the output states when the part is in various states.

Table 7. OUTPUT STATES

Name	Hardware States		Firmware States				Notes
	Reset State	Default State	Hard Standby	Soft Standby	Streaming	Idle	
EXTCLK	(clock running or stopped)	(clock running)	(clock running or stopped)	(clock running)	(clock running)	(clock running)	Input
XTAL	n/a	n/a	n/a	n/a	n/a	n/a	Output

Table 7. OUTPUT STATES (continued)

Name	Hardware States		Firmware States				Notes
	Reset State	Default State	Hard Standby	Soft Standby	Streaming	Idle	
RESET_BAR	(asserted)	(negated)	(negated)	(negated)	(negated)	(negated)	Input
FRAME_ SYNC	n/a	n/a	n/a	n/a	n/a	n/a	Input. Must always be driven to a valid logic level.
STANDBY	n/a	(negated)	(negated)	(negated)	(negated)	(negated)	Input. Must always be driven to a valid logic level.
EXT_REG	n/a	n/a	n/a	n/a	n/a	n/a	Input. Must always be driven to a valid logic level.
ENLDO	n/a	n/a	n/a	n/a	n/a	n/a	Input. Must be tied to V_{DD}_REG or GND.
SPI_SCLK	High−impedance	driven, logic 0	driven, logic 0	driven, logic 0			Output
SPI_SDI	Internal pull−up enabled	Internal pull−up enabled	Internal pull−up enabled	Internal pull−up enabled			Input. Internal pull−up permanently enabled.
SPI_SDO	High−impedance	driven, logic 0	driven, logic 0	driven, logic 0			Output
SPI_CS_BAR	High−impedance	driven, logic 1	driven, logic 1	driven, logic 1			Output
EXT_CLK_ OUT	driven, logic 0	driven, logic 0	driven, logic 0	driven, logic 0	EXT_CLK_ OUT running in streaming state, stopped or running in idle state (depending on other FW sub−state). RESET_BAR _OUT low in streaming, low or high in idle state (depending on other FW sub−state).		Output
RESET_BAR_OUT	driven, logic 0	driven, logic 0	driven, logic 1	driven, logic 1			Output. Firmware will release sensor reset.
M_S_{CLK}	High−impedance	High−impedance	High−impedance	High−impedance			Input/Output. A valid logic level should be established by pull−up.
M_S_{DATA}	High−impedance	High−impedance	High−impedance	High−impedance			Input/Output. A valid logic level should be established by pull−up.
FV_IN LV_IN, PIXCLK_IN, DIN [11:0]	n/a	n/a	n/a	n/a	Dependent on interface used	n/a	Input. Must always be driven to a valid logical level.
HiSPi_CN	Disabled	Disabled	Dependent on interface used	Dependent on interface used	Dependent on interface used	Dependent on interface used	Input. Will be disabled and can be left floating.
HiSPi_CP							
HiSPi0_N							
HiSPi0_P							
HiSPi1_N							
HiSPi1_P							

Table 7. OUTPUT STATES (continued)

Name	Hardware States		Firmware States				Notes
	Reset State	Default State	Hard Standby	Soft Standby	Streaming	Idle	
TX_CLK, RX_CLK, GTX_CLK	High−impedance	Varied	Driven if used	Driven if used	Driven if used	Driven if used	Output. Default state dependent on configuration.
TXD[7:0], TX_ERR, TX_EN	Driven to '0'					Driven if used	Transmit data bits 7 to 0 for MII/RMII protocols (MAC to PHY). Transmit error. Transmit enable.
RXD[7:0], RX_ERR	High−impedance						Receive data bits 7 to 0 for MII/RMII protocols (PHY to MAC). Receive error.
MDC, MDIO	MDC: Driven to '0' MDIO: High Impedance						MDC: Management data clock line MDIO: Management data I/O line
GPIO[6:1]	High−impedance	Input, then high−impedance	Driven if used	Driven if used	Driven if used	Driven if used	Input/Output.
TRIGGER_ OUT	High−impedance	High−impedance	Driven if used	Driven if used	Driven if used	Driven if used	

Hard Reset

The AP0201AT enters the reset state when the external RESET_BAR signal is asserted LOW, as shown in Figure 6. Refer to Table 7 for details.

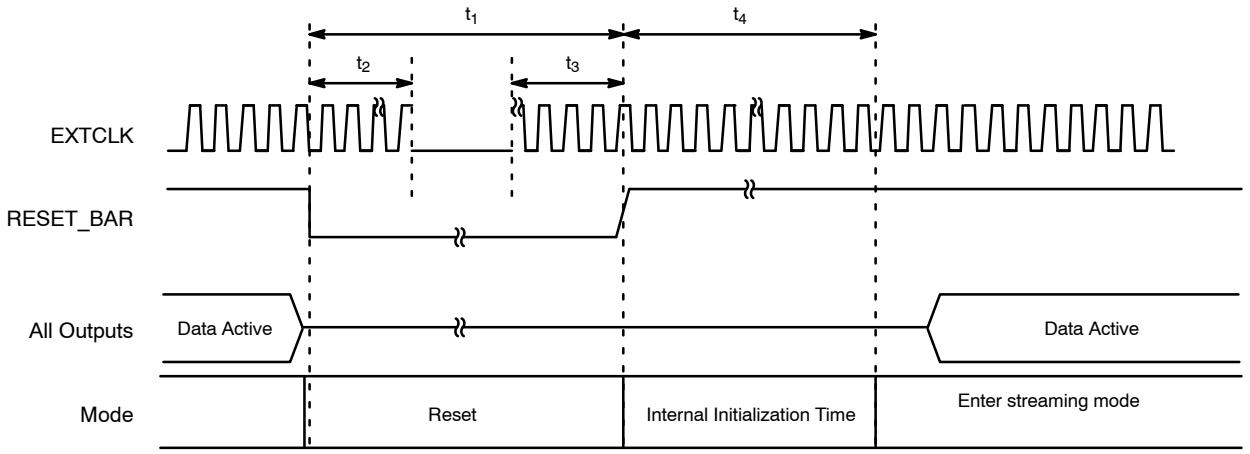


Figure 6. Hard Reset Operation

Table 8. HARD RESET

Symbol	Parameter	Min	Typ	Max	Unit
t_1	RESET_BAR pulse width	50	−	−	EXTCLK cycles
t_2	Active EXTCLK required after RESET_BAR asserted	10	−	−	
t_3	Active EXTCLK required after RESET_BAR de−asserted	10	−	−	
t_4	Internal initialization time after RESET is HIGH	100	−	−	

Soft Reset

A soft reset sequence to the AP0201AT can be activated by writing to a register through the Ethernet interface.

Hard Standby Mode

The AP0201AT can enter hard standby mode by using the external STANDBY signal, as shown in Figure 7. In hard standby mode, the total power consumption is reduced. In this mode, the AP0201AT is switched off. A further power reduction can be achieved by turning off the input clock, but this must be restored before de−asserting the STANDBY pin to LOW state to restart the device.

Entering Standby Mode

1. Assert STANDBY signal HIGH.

Existing Standby Mode

1. De−assert STANDBY signal LOW.

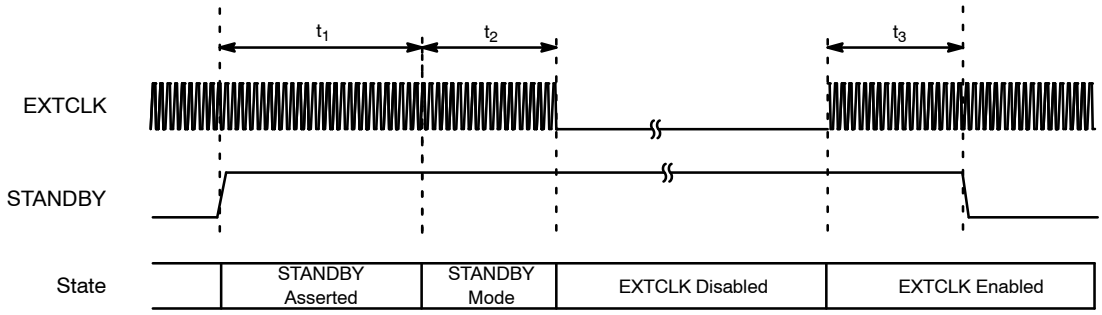


Figure 7. Hard Standby Operation

Table 9. HARD STANDBY SIGNAL TIMING

Symbol	Parameter	Min	Typ	Max	Unit
t_1	Standby entry complete	–	–	2	Frames
t_2	Active EXTCLK required after going into STANDBY mode	10	–	–	EXTCLKs
t_3	Active EXTCLK required before STANDBY de−asserted	10	–	–	EXTCLKs

DEVICE CONFIGURATION

After power is applied and the device is out of reset (either hard reset or soft reset), it will enter a boot sequence to configure its operating mode. There are essentially three configuration modes: Flash/EEPROM Config, Auto Config, and Host Config.

The AP0201AT firmware supports a System Configuration phase at start−up. This consists of three sub−phases of execution:

Flash detection, then one of:

a. Flash Config
b. Auto Config
c. Host Config

The System Configuration phase is entered immediately following power−up or reset. Then the firmware performs Flash Detection.

Flash Detection attempts to detect the presence of an SPI Flash or EEPROM device:

- If a device is detected, the firmware switches to the Flash−Config mode.
- If no device is detected, the firmware then samples the SPI_SDI pin state to determine the next mode:
  - ♦ If SPI_SDI is low, then it enters the Host−Config mode.
  - ♦ If SPI_SDI is high, then it enters the Auto−Config mode.

In the Flash−Config mode, the firmware interrogates the device to determine if it contains valid configuration records:

- If no records are detected, then the firmware enters the Host−Config mode.
- If records are detected, the firmware processes them. By default, when all Flash records are processed the firmware switches to the Host−Config mode. However, the records encoded into the Flash can optionally be used to instruct the firmware to proceed to auto−config, or to start streaming (via a Change−Config).

In the Host−Config mode, the firmware performs no configuration, and remains idle waiting for configuration and commands from the host. The System Configuration phase is effectively complete and the AP0201AT will take no actions until the host issues commands.

USAGE MODES

How a camera based on the AP0201AT will be configured depends on what features are used. In the simplest case, an AP0201AT operating in Auto−Config mode with no customized settings might be sufficient.

A back−up camera with dynamic input from the steering system will require a host system with Ethernet capability. Flash sizes vary depending on the register and firmware data being transferred−the AP0201AT supports devices up to 2 GB.

In the simplest case no EEPROM or Flash memory is required, as shown in Figure 8.

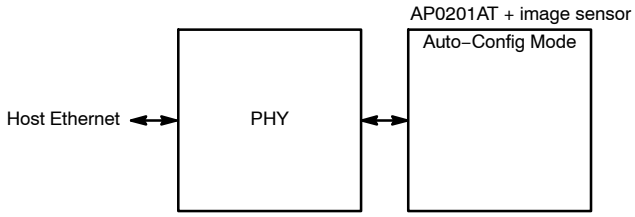
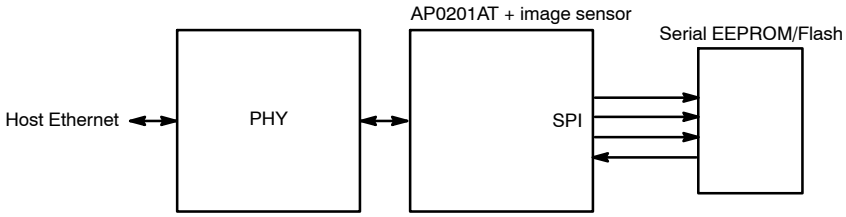
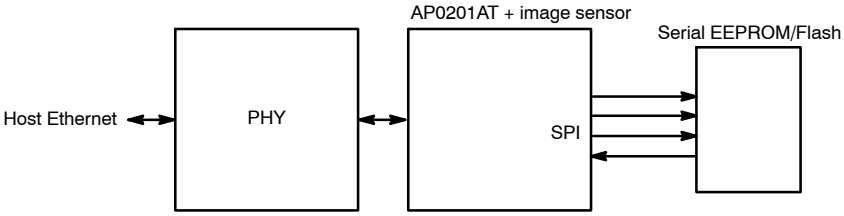


Figure 8. Auto−Config Mode



NOTE: The AP0201AT can be configured by a serial EEPROM or Flash through the SPI Interface.

Figure 9. Flash Mode



NOTE: In this configuration all settings are communicated to the AP0201AT and sensor through the host.

Figure 10. Host Mode with Flash

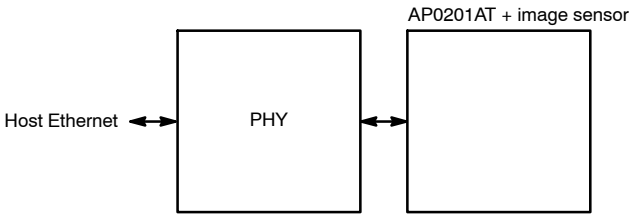


Figure 11. Host Mode

IMAGE FLOW PROCESSOR

Image and color processing in the AP0201AT is implemented as an image flow processor (IFP) coded in hardware logic. During normal operation, the embedded microcontroller will automatically adjust the operating parameters. For normal operation of the AP0201AT, streams of raw image data from the attached image sensor are fed into the color pipeline. The AP201AT also has the option to select a number of test patterns to be input instead of sensor data.

Defect Correction

Image stream processing commences with the defect correction function immediately after data decompanding.

To obtain defect free images, the pixels marked defective during sensor readout and the pixels determined defective by the defect correction algorithms are replaced with values derived from the non−defective neighboring pixels.

AdaCD (Adaptive Color Difference)

The next step in the image stream processing is noise reduction. The AP0201AT uses a noise reduction filter called AdaCD which focuses on removing color noise while preserving edge details. Automotive applications require good performance in extremely low light, even at high temperature conditions. In these stringent conditions the image sensor is prone to higher noise levels, and so efficient noise reduction techniques are required to circumvent this sensor limitation and deliver a high quality image to the user.

Black Level Substraction and Digital Gain

After noise reduction, the pixel data goes through black level subtraction and multiplication by a programmable digital gain. Independent color channel digital gain can be adjusted with registers. Black level subtraction (to compensate for sensor data pedestal) is a single value applied to all color channels. If the black level subtraction produces a negative result for a particular pixel, the value of this pixel is set to 0.

Positional Gain Adjustments (PGA)

Lenses tend to produce images whose brightness is significantly attenuated near the edges. There are also other factors causing fixed pattern signal gradients in images captured by image sensors. The cumulative result of all these factors is known as image shading. The AP0201AT has an embedded shading correction module that can be programmed to counter the shading effects on each individual R, Gb, Gr, and B color signal.

The Correction Function

The correction functions can then be applied to each pixel value to equalize the response across the image as follows:

$$P_{corrected}(row, col) = P_{sensor}(row, col) \times f(row, col) \quad \text{(eq. 1)}$$

where P are the pixel values and f is the color dependent correction functions for each color channel.

Adaptive Local Tone Mapping (ALTM)

Real world scenes often have very high dynamic range (HDR) that far exceeds the electrical dynamic range of the imager. Dynamic range is defined as the luminance ratio between the brightest and the darkest object in a scene. In recent years many technologies have been developed to capture the full dynamic range of real world scenes. For example, the multiple exposure method is widely adopted for capturing high dynamic range images, which combines a series of low dynamic range images of the same scene taken under different exposure times into a single HDR image.

Even though the new digital imaging technology enables the capture of the full dynamic range, low dynamic range display devices are the limiting factor. Today's typical LCD monitor has contrast ratio around 1,000:1; this contrast ratio is not enough for an HDR image (the contrast ratio for an HDR image is around 250,000:1). Therefore, in order to reproduce HDR images on a low dynamic range display device, the captured high dynamic range must be compressed to the available range of the display device. This is commonly called tone mapping.

Tone mapping methods can be classified into global tone mapping and local tone mapping. Global tone mapping methods apply the same mapping function to all pixels. While global tone mapping methods provide computationally simple and easy to use solutions, they often cause loss of contrast and detail. A local tone mapping is thus necessary in addition to global tone mapping for the reproduction of visually more appealing images that also reveal scene details that are important for automotive safety applications. Local tone mapping methods use a spatially variable mapping function determined by the neighborhood of a pixel, which allows it to increase the local contrast and the visibility of some details of the image. Local methods usually yield more pleasing results because they exploit the fact that human vision is more sensitive to local contrast.

onsemi's ALTM solution significantly improves the performance over global tone mapping. ALTM is directly applied to the Bayer domain to compress the dynamic range from 20−bit to 12−bit. This allows the regular color pipeline to be used for HDR image rendering.

Color Interpolation

In the raw data stream fed by the external sensor to the IFP, each pixel is represented by a 20− or 12−bit integer number, which can be considered proportional to the pixel's response to a one−color light stimulus, red, green, or blue, depending on the pixel's position under the color filter array. Initial data processing steps, up to and including ALTM, preserve the one−color−per−pixel nature of the data stream, but after ALTM it must be converted to a three−colors−per−pixel stream appropriate for standard color processing. The conversion is done by an edge−sensitive color interpolation module. The module pads the incomplete color information available for each pixel with information extracted from an

appropriate set of neighboring pixels. The algorithm used to select this set and extract the information seeks the best compromise between preserving edges and filtering out high frequency noise in flat field areas. The edge threshold can be set through register settings.

Color Correction and Aperture Correction

To achieve good color fidelity of the IFP output, interpolated RGB values of all pixels are subjected to color correction. The IFP multiplies each vector of three pixel colors by a 3 x 3 color correction matrix. The three components of the resulting color vector are all sums of three 10−bit numbers. The color correction matrix can be either programmed by the host or automatically selected by the auto white balance (AWB) algorithm implemented in the IFP. Color correction should ideally produce output colors that are corrected for the spectral sensitivity and color crosstalk characteristics of the image sensor. The optimal values of the color correction matrix elements depend on those sensor characteristics and on the spectrum of light incident on the sensor. The color correction variables can be adjusted through register settings.

The AP0201AT offers a three−CCM solution that will give the user improved color fidelity when under CWF type lighting.

To increase image sharpness, a programmable 2D aperture correction (sharpening filter) is applied to color−corrected image data. The gain and threshold for 2D correction can be defined through register settings

Gamma Correction

The gamma correction curve is implemented as a piecewise linear function with 33 knee points, taking 12−bit arguments and mapping them to 10−bit output. The abscissas of the knee points are fixed at 0, 8, 16, 24, 32, 40, 48, 56, 64, 80, 96, 112, 128, 160, 192, 224, 256, 320, 384, 448, 512, 640, 768, 896, 1024, 1280, 1536, 1792, 2048, 2560, 3072, 3584, and 4096. The 10−bit ordinates are programmable through variables.

The AP0201AT has the ability to calculate the 33−point knee points based on the tuning of cam_ll_gamma and cam_ll_contrast_gradient_bright. The other method is for the host to program the 33 knee point curve directly.

Also included in this block is a Fade−to Black curve which sets all knee points to zero and causes the image to go black in extreme low light conditions.

Color Kill

To remove high−or low−light color artifacts, a color kill circuit is included. It affects only pixels whose luminance exceeds a certain preprogrammed threshold. The U and V values of those pixels are attenuated proportionally to the difference between their luminance and the threshold.

YUV Color Filter

As an optional processing step, noise suppression by one−dimensional low−pass filtering of Y and/or UV signals is possible. A 3− or 5−tap filter can be selected for each signal.

CAMERA CONTROL AND AUTO FUNCTIONS

Auto Exposure

The auto exposure algorithm optimizes scene exposure to minimize clipping and saturation in critical areas of the image. This is achieved by controlling exposure time and analog gains of the external sensor as well as digital gains applied to the image.

Auto exposure is implemented by a firmware algorithm that is running on the embedded microcontroller that analyzes image statistics collected by the exposure measurement engine, makes a decision, and programs the sensor and color pipeline to achieve the desired exposure. The measurement engine subdivides the image into 25 windows organized as a 5 x 5 grid.

W 0,0	W 0,1	W 0,2	W 0,3	W 0,4
W 1,0	W 1,1	W 1,2	W 1,3	W 1,4
W 2,0	W 2,1	W 2,2	W 2,3	W 2,4
W 3,0	W 3,1	W 3,2	W 3,3	W 3,4
W 4,0	W 4,1	W 4,2	W 4,3	W 4,4

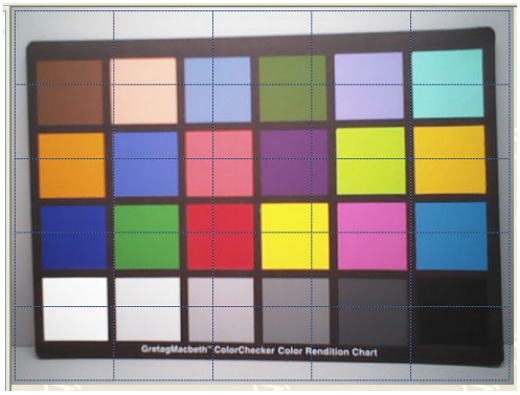


Figure 12. 5 x 5 Grid

The region of interest can be controlled through the combination of an inclusion window and an exclusion window.

AE Track Driver

Other algorithm features include the rejection of fast fluctuations in illumination (time averaging), control of speed of response, and control of the sensitivity to small changes. While the default settings are adequate in most situations, the user can program target brightness, measurement window, and other parameters described above.

The driver changes AE parameters (integration time, gains, and so on) to drive scene brightness to the programmable target.

To avoid unwanted reaction of AE on small fluctuations of scene brightness or momentary scene changes, the AE track driver uses a temporal filter for luma and a threshold around the AE luma target. The driver changes AE parameters only if the filtered luma is larger than the AE target step and pushes the luma beyond the threshold.

Auto White Balance

The AP0201AT has a built−in AWB algorithm designed to compensate for the effects of changing spectra of the scene illumination on the quality of the color rendition. The algorithm consists of two major parts: a measurement engine performing statistical analysis of the image and a driver performing the selection of the optimal color correction matrix and IFP digital gain. While default settings of these algorithms are adequate in most situations, the user can reprogram base color correction matrices, place limits on color channel gains, and control the speed of both matrix and gain adjustments. The AP0201AT AWB displays the current AWB position in color temperature, the range of which is defined by programmable settings.

The region of interest can be controlled through the combination of an inclusion window and an exclusion window.

Exposure and White Balance Control

The Sensor Manager firmware component is responsible for controlling the application of ’exposure’ and ’white balance’ within the system. This effectively means that all control of integration times and gains (whether for exposure or white balance) is delegated to the Sensor Manager. The Auto Exposure (AE) and Auto White Balance (AWB) algorithms use services provided by the Sensor Manager to apply exposure and/or white balance changes.

Dual Band IRCF

Manager firmware component is responsible for controlling the application of ’exposure’ and ’white balance’ within the system. This effectively means that all control of integration times and gains (whether for exposure or white balance) is delegated to the Sensor Manager. The Auto Exposure (AE) and Auto White Balance (AWB) algorithms use services provided by the Sensor Manager to apply exposure and/or white balance changes.

Exposure and White Balance Modes

The AP0201AT supports auto and manual exposure and white balance modes. In addition, it will operate within synchronized multi−camera systems. In this use case, one camera within the system will be the ’master’, and the others ’slaves’. The master is used to calculate the appropriate exposure and white balance. This is then applied to all slaves concurrently under host control.

Auto Mode

In Auto Exposure mode the AE algorithm is responsible for calculating the appropriate exposure to keep the desired scene brightness, and for applying the exposure to the underlying hardware. In Auto White Balance mode the AWB algorithm is responsible for calculating the color temperature of the scene and applying the appropriate red and blue gains to compensate.

Triggered Auto Mode

The Triggered Auto Exposure and Triggered Auto White Balance modes are intended for the multicamera use cases, where a host is controlling the exposure and white balance of a number of cameras. The idea is that one camera is in triggered−auto mode (the master), and the others in hostcontrolled mode (slaves). The master camera must calculate the exposure and gains, the host then copies this to the slaves, and all changes are then applied at the same time.

Manual Mode

Manual mode is intended to allow simple manual exposure and white balance control by the host. The host needs to set the CAM_AET_EXPOSURE_TIME_MS, CAM_AET_EXPOSURE_GAIN and CAM_AWB_COLOR_TEMPERATURE controls, the camera will calculate the appropriate integration times and gains.

Host Controlled

The Host Controlled mode is intended to give the host full control over exposure and gains to allow host to control desired output.

FLICKER AVOIDANCE

Flicker occurs when the integration time is not an integer multiple of the period of the light intensity. The AP0201AT can be programmed to avoid flicker for 50 or 60 Hertz. For integration times below the light intensity period (10ms for 50Hz environment, 8.33 ms for a 60 Hz environment), flicker cannot be avoided. The AP0201AT supports an indoor AE mode, that will ensure flicker−free operation.

FLICKER DETECTION

The AP0201AT supports flicker detection, the algorithm is designed only to detect a 50 Hz or 60 Hz flicker source.

OUTPUT FORMATTING

The pixel output data in AP0201AT will be transmitted as an 8−bit word over the Ethernet interface.

Output Video Formats

The AP0201AT conforms with the IEEE standard for both MJPEG and H.264 video outputs. For reference, the standard is “IEEE Standard for Layer 2 Transport Protocol for Time−Sensitive Applications in Bridged Local Area Networks” and can be obtained from IEEE.

H.264 Format

The AP0201AT is compliant with the ITU−T REC. H.264 standard published by the Telecommunication Standardization Sector of the International Telecommunication Union, which is equivalent to ISO/IEC 14496−10.

The AP0201AT supports the standard H.264 for video compression which is equivalent to MPEG−4 Part 10, also known as MPEG−4 Advanced Video Coding (AVC). The AP0201AT utilizes an advanced High profiles compliant encoder, constrained to the All−Intra encoding schemes. It supports real time encoding of 4:2:0 video streams, up to Level 5.2, in 8 and 10 bit sample depths. The core only needs to be programmed once per video sequence. Once programmed, the AP0201AT can encode an arbitrary number of video frames, without the need of any further intervention from the host system.

H.264 Features

The AP0201AT H.264 encoding includes the following features:

- High 10 intra profile encoding
- 8− and 10−bit sample depth encoding
- Level up to 5.2
- ITU−T H.264 Annex B compliant NAL byte stream output
- CQP − VBR encoding mode
  - Rate−Distortion optimized output
  - Up to 240MBits/s output (CAVLC)
- CBR encoding mode
  - HRD CPB compliant CBR NAL output
  - Sub−frame operation with tunable number of macroblocks basis
  - Further micro adjustment of quantization per macroblock maximizes the perceived video quality
  - Both Rate−Distortion metrics and perceived video quality optimized
  - On−the−fly rate changes are supported
  - Up to 240 Mbits/s output (CAVLC)
- Advanced Intra prediction
  - All four Intra 16x16 prediction modes
  - All four Intra Chroma prediction modes
  - All nine Intra 4x4 prediction modes
- Error resilience
  - Multiple slices per frame encoding
  - Deblocking filter in the decoder can be optionally constrained to operate within slice boundaries
- Optional advanced thresholding of quantized transform coefficients
  - Eliminates spares and insignificant transform coefficients
  - Improves the compression efficiency
  - Near−zero impact to the measured video quality
  - Zero impact to the perceived, subjective, video quality
- Run−time tunable operation enables decoder compatibility trade−offs
  - Full control of allowed Intra prediction modes
  - Single or multiple slices per frame encoding
  - Option and tunable deblocking filter operation
  - CAVLC coding

MJPEG FORMAT

JPEG Encoder

The JPEG compression engine in the AP0201AT is a highly integrated, high−performance solution that provides for low power consumption and programmability of JPEG compression parameters for image quality control.

The JPEG encoding block is designed for continuous image flow and is ideal for low power applications. After initial configuration for a target application, it can be controlled easily for instantaneous stop or restart. A flexible configuration and control interface allows for full programmability of various JPEG−specific parameters and tables.

JPEG Encoding Highlights

- Sequential DCT (baseline) ISO/IEC 10918−1 JPEG−compliant
- YCbCr 4:2:0 format compression
- JPEG capability at full resolution with JFIF− or RFC2435−compliant header
- Programmable automatic control of bit rate

Stream Breakdown

An MJPEG video stream consists of the following sequence of data sections. Each JPEG frame must have the following characteristics:

- Color Encoding is YcbCr
- 8 bits per color component, (24 bits/pixel before subsampling)
- 420 Subsampling
- Baseline sequential DCT (SOF0)

JPEG Header

The MJPEG stream can be output with 4 different header settings.

Three of them are similar to each other. The first is the standard JPEG header as defined in the original JPEG specification. The second is a JFIF header which is the standard header plus a JFIF segment. The third is a standard header minus the Huffman table. Since, for this design, the Huffman table is constant, some bandwidth can be saved by not including it.

The 4th header option is optimized for Ethernet and is referred to as RFC2435.

JFIF, Standard and Standard minus Huffman Headers

For these three header types, the header segments that will be included are listed below including examples. Note that data values in the examples are in hex. Comments are in decimal.

- SOI, Start of Image. 2 bytes.
  ff d8
- APP0, Application Segment 0. N bytes (only included in JFIF headers):
  Example JFIF marker: ff e0 00 10
  4a 46 49 46 00 01 02 00 00 01 00 01 00 00
- DHT, Define Huffman Tables, 420 bytes (Not included in standard header without Huffman table)
  Example: ff c4 01 a2

#DC Table 0

00																
00	01	05	01	01	01	01	01	01	00	00	00	00	00	00	00	#12 codes
00																
00	01	02	03	04	05	06	07	08	09	0a	0b					

#AC Table 0

10															
00	02	01	03	03	02	04	03	05	05	04	04	00	00	01	7d	#162 codes
01	02	03	00	04	11	05	12	21	31	41	06	13	51	61	07	
22	71	14	32	81	91	a1	08	23	42	b1	c1	15	52	d1	f0	
24	33	62	72	82	09	0a	16	17	18	19	1a	25	26	27	28	
29	2a	34	35	36	37	38	39	3a	43	44	45	46	47	48	49	
4a	53	54	55	56	57	58	59	5a	63	64	65	66	67	68	69	
6a	73	74	75	76	77	78	79	7a	83	84	85	86	87	88	89	
8a	92	93	94	95	96	97	98	99	9a	a2	a3	a4	a5	a6	a7	
a8	a9	aa	b2	b3	b4	b5	b6	b7	b8	b9	ba	c2	c3	c4	c5	
c6	c7	c8	c9	ca	d2	d3	d4	d5	d6	d7	d8	d9	da	e1	e2	
e3	e4	e5	e6	e7	e8	e9	ea	f1	f2	f3	f4	f5	f6	f7	f8	
f9	fa															

#DC Table 1

01																
	00	03	01	01	01	01	01	01	01	01	01	00	00	00	00	00	#12 codes
	00	01	02	03	04	05	06	07	08	09	0a	0b					

#AC Table 1

11															
00	02	01	02	04	04	03	04	07	05	04	04	00	01	02	77	#162 codes
00	01	02	03	11	04	05	21	31	06	12	41	51	07	61	71	
13	22	32	81	08	14	42	91	a1	b1	c1	09	23	33	52	f0	
15	62	72	d1	0a	16	24	34	e1	25	f1	17	18	19	1a	26	
27	28	29	2a	35	36	37	38	39	3a	43	44	45	46	47	59	
49	4a	53	54	55	56	57	58	59	5a	63	64	65	66	67	68	
69	6a	73	74	75	76	77	78	79	7a	82	83	84	85	86	87	
88	89	8a	92	93	94	95	96	97	98	99	9a	a2	a3	a4	a5	
a6	a7	a8	a9	aa	b2	b3	b4	b5	b6	b7	b8	b9	ba	c2	c3	

#AC Table 1

c4	c5	c6	c7	c8	c9	ca	d2	d3	d4	d5	d6	d7	d8	d9	da
e2	e3	e4	e5	e6	e7	e8	e9	ea	f2	f3	f4	d5	f6	f7	f8
f9	fa														

- DQT, Define Quantization Tables. 134 bytes.
  Example: ff db 00 84

#8−bit, Table 0

00															
10	0b	0c	0e	0c	0a	10	0e	0d	0e	12	11	10	13	18	28
1a	18	16	16	18	31	23	25	1d	28	3a	33	3d	3c	39	33
38	37	40	48	5c	4e	40	44	57	45	37	38	50	6d	51	57
5f	62	67	68	67	3e	4d	71	79	70	64	78	5c	65	67	63

#8−bit, Table 1

01															
11	12	12	18	15	18	2f	1a	1a	2f	63	42	38	42	63	63
63	63	63	63	63	63	63	63	63	63	63	63	63	63	63	63
63	63	63	63	63	63	63	63	63	63	63	63	63	63	63	63
63	63	63	63	63	63	63	63	63	63	63	63	63	63	63	63

The quantization table can be adjusted for each frame for more or less compression.

- DRI, Define Restart Interval. 6 bytes.
  Example: ff db 00 04 00 78

The segment is optional. The host will determine whether to include Restart markers and at what interval.

- SOF0, Start of Frame 0. 19 bytes.
  Example: ff c0 00 11

08			#Sample precision
04	38		#Number of rows = 1080
07	80		#Number of columns = 1920
03			#Number of components
01	21	00	#Component 1: HSF = 2, VSF = 1, Q Table = 0
02	11	01	#Component 2: HSF = 1, VSF = 1, Q Table = 1
03	11	01	#Component 3: HSF = 1, VSF = 1, Q Table = 1

- SOS, Start of Scan. 14 bytes.
  Example: ff da 00 0c

03		#Number of components
01	00	#Component 1: DC table 0, AC table 0
02	11	#Component 2: DC table 1, AC table 1
03	11	#Component 3: DC table 1, AC table 1
00		#Start of spectral selection
3f		#End of spectral selection
00		#Successive approximation high/low

Compressed Data With or Without Restart Markers

This is compressed binary data of the frame which can be decoded to display the captured image. The JPEG compression engine can be configured to insert restart marker at programmable intervals.

EOI

This is the End of Image code. It is only 2 bytes long.
ff d9

Huffman Table

When standard JPEG headers without Huffman tables is selected, the Huffman table is not included in the data stream. However, the decoder will need to know what the table is to perform the decode.

The required Huffman table is:

```
/* Default DHT Segment */
MJPGHDTSEG_STORAGE BYTE MUPGDHTSeg[0x1A0] = {
/*JPEG DHT Segment for YCrCb omitted from MJPG data*/
0xFF  0xC4  0x01  0xA2
0x00  0x00  0x01  0x05  0x01  0x01  0x01  0x01  0x01  0x01  0x00  0x00  0x00  0x00  0x00
0x00  0x00  0x00  0x01  0x02  0x03  0x04  0x05  0x06  0x07  0x08  0x09  0x0A  0x0B  0x01
0x00  0x03  0x01  0x01  0x01  0x01  0x01  0x01  0x01  0x01  0x01  0x00  0x00  0x00  0x00

0x00  0x00  0x01  0x02  0x03  0x04  0x05  0x06  0x07  0x08  0x09  0x0A  0x0B  0x10  0x00
0x02  0x01  0x03  0x03  0x02  0x04  0x03  0x05  0x05  0x04  0x04  0x00  0x00  0x01  0x7D
0x01  0x02  0x03  0x00  0x04  0x11  0x05  0x12  0x21  0x31  0x41  0x06  0x13  0x51  0x61
0x07  0x22  0x71  0x14  0x32  0x81  0x91  0xA1  0x08  0x23  0x42  0xB1  0xC1  0x15  0x52
0xD1  0xF0  0x24  0x33  0x62  0x72  0x82  0x09  0x0A  0x16  0x17  0x18  0x19  0x1A  0x25
0x26  0x27  0x28  0x29  0x2A  0x34  0x35  0x36  0x37  0x38  0x39  0x3A  0x43  0x44  0x45
0x46  0x47  0x48  0x49  0x4A  0x53  0x54  0x55  0x56  0x57  0x58  0x59  0x5A  0x63  0x64

0x65  0x66  0x67  0x68  0x69  0x6A  0x73  0x74  0x75  0x76  0x77  0x78  0x79  0x7A  0x83
0x84  0x85  0x86  0x87  0x88  0x89  0x8A  0x92  0x93  0x94  0x95  0x96  0x97  0x98  0x99
0x9A  0xA2  0xA3  0xA4  0xA5  0xA6  0xA7  0xA8  0xA9  0xAA  0xB2  0xB3  0xB4  0xB5  0xB6
0xB7  0xB8  0xB9  0xBA  0xC2  0xC3  0xC4  0xC5  0xC6  0xC7  0xC8  0xC9  0xCA  0xD2  0xD3
0xD4  0xD5  0xD6  0xD7  0xD8  0xD9  0xDA  0xE1  0xE2  0xE3  0xE4  0xE5  0xE6  0xE7  0xE8
0xE9  0xEA  0xF1  0xF2  0xF3  0xF4  0xF5  0xF6  0xF7  0xF8  0xF9  0xFA  0x11  0x00  0x02
0x01  0x02  0x04  0x04  0x03  0x04  0x07  0x05  0x04  0x04  0x00  0x01  0x02  0x77  0x00

0x01  0x02  0x03  0x11  0x04  0x05  0x21  0x31  0x06  0x23  0x41  0x51  0x07  0x61  0x71
0x13  0x22  0x32  0x81  0x08  0x14  0x42  0x91  0xA1  0xB1  0xC1  0x09  0x23  0x33  0x52
0xF0  0x15  0x62  0x72  0xD1  0x0A  0x16  0x24  0x34  0xE1  0x25  0xF1  0x17  0x18  0x19
0x1A  0x25  0x27  0x28  0x29  0x2A  0x35  0x36  0x37  0x38  0x39  0x3A  0x43  0x44  0x45
0x46  0x47  0x48  0x49  0x4A  0x53  0x54  0x55  0x56  0x57  0x58  0x59  0x5A  0x63  0x64
0x65  0x66  0x67  0x68  0x69  0x6A  0x73  0x74  0x75  0x76  0x77  0x78  0x79  0x7A  0x82
0x83  0x84  0x85  0x86  0x87  0x88  0x89  0x8A  0x92  0x93  0x94  0x95  0x96  0x97  0x98

0x99  0x9A  0xA2  0xA3  0xA4  0xA5  0xA6  0xA7  0xA8  0xA9  0xAA  0xB2  0xB3  0xB4  0xB5

0xB6  0xB7  0xB8  0xB9  0xBA  0xC2  0xC3  0xC4  0xC5  0xC6  0xC7  0xC8  0xC9  0xCA  0xD2
0xD3  0xD4  0xD5  0xD6  0xD7  0xD8  0xD9  0xDA  0xE2  0xE3  0xE4  0xE5  0xE6  0xE7  0xE8
0xE9  0xEA  0xF2  0xF3  0xF4  0xF5  0xF6  0xF7  0xF8  0xF9  0xFA
};
```

Embedded Data and Statistics

Some **onsemi** sensor's support a feature that, if enabled, inserts two extra lines at the beginning and end of each frame which contain information about that frame. The first two lines contain specific register values that were used to capture that frame. These values allow the host to know certain important things about how the sensor was configured for that frame, e.g. exposure, gain, image size, etc. The last two lines contain statistics about the image that was captured, e.g. mean values, intensity histograms, etc.

In Ethernet mode AP0201AT sends out four lines of Sensor Embedded data over Ethernet for every frame. Each line is carried in a separate UDP packet. The destination port and IP address can be specified. By default the destination IP address and port are set to 255.255.255.255 and 50010.

UDP packet's payload consists of following data bytes:

1. Four byte of Timestamp to match with the frame.
2. Four byte of line number. AP0201AT sends line numbers zero and one at the beginning of the frame. Line numbers two and three are sent at the end of the frame.
3. Embedded Data.

Note that embedded data is limited to 256 outputs per line, so does not support sensors with additional embedded data beyond that number as the additional data will be truncated.

SUPPORTRED SPI DEVICES

The supported devices are those that conform to the JEDEC−compliant programming interface. Please contact **onsemi** for specific design criteria and requirements. The maximum size supported in 2 GB.

SLAVE TWO−WIRE SERIAL INFTERFACE (CCIS)

The two−wire slave serial interface bus enables read/write access to control and status registers within the AP0201AT.

The interface protocol uses a master/slave model in which a master controls one or more slave devices.

Protocol

Data transfers on the two−wire serial interface bus are performed by a sequence of low−level protocol elements, as follows:

- a start or restart condition
- a slave address/data direction byte
- a 16−bit register address
- an acknowledge or a no−acknowledge bit
- data bytes
- a stop condition

The bus is idle when both SCLK and SDATA are HIGH. Control of the bus is initiated with a start condition, and the bus is released with a stop condition. Only the master can generate the start and stop conditions.

The SADDR pin is used to select between two different addresses in case of conflict with another device. If SADDR is LOW, the slave address is 0x90; if SADDR is HIGH, the slave address is 0xBA. See Table 10. The user can change the slave address by changing a register value.

Table 10. TWO−WIRE INTERFACE ID ADDRESS SWITCHING

S_{ADDR}	Two−Wire Interface Address ID
0	0x90
1	0xBA

Start Condition

A start condition is defined as a HIGH−to−LOW transition on SDATA while SCLK is HIGH.

At the end of a transfer, the master can generate a start condition without previously generating a stop condition; this is known as a "repeated start" or "restart" condition.

Data Transfer

Data is transferred serially, 8 bits at a time, with the MSB transmitted first. Each byte of data is followed by an acknowledge bit or a no−acknowledge bit. This data transfer mechanism is used for the slave address/data direction byte and for message bytes. One data bit is transferred during each SCLK clock period. SDATA can change when SCLK is low and must be stable while SCLK is HIGH.

Slave Address/Data Direction Byte

Bits [7:1] of this byte represent the device slave address and bit [0] indicates the data transfer direction. A "0" in bit [0] indicates a write, and a "1" indicates a read. The default slave addresses used by the AP0201AT are 0x90 (write address) and 0x91 (read address). Alternate slave addresses of 0xBA (write address) and 0xBB (read address) can be selected by asserting the SADDR input signal.

Message Byte

Message bytes are used for sending register addresses and register write data to the slave device and for retrieving register read data. The protocol used is outside the scope of the two−wire serial interface specification.

Acknowledge Bit

Each 8−bit data transfer is followed by an acknowledge bit or a no−acknowledge bit in the SCLK clock period following the data transfer. The transmitter (which is the master when writing, or the slave when reading) releases SDATA. The receiver indicates an acknowledge bit by driving SDATA LOW. As for data transfers, SDATA can change when SCLK is LOW and must be stable while SCLK is HIGH.

No−Acknowledge Bit

The no−acknowledge bit is generated when the receiver does not drive SDATA low during the SCLK clock period following a data transfer. A no−acknowledge bit is used to terminate a read sequence.

Stop Condition

A stop condition is defined as a LOW−to−HIGH transition on SDATA while SCLK is HIGH.

Typical Operation

typical READ or WRITE sequence begins by the master generating a start condition on the bus. After the start condition, the master sends the 8−bit slave address/data direction byte. The last bit indicates whether the request is for a READ or a WRITE, where a “0” indicates a WRITE and a “1” indicates a READ. If the address matches the address of the slave device, the slave device acknowledges receipt of the address by generating an acknowledge bit on the bus.

If the request was a WRITE, the master then transfers the 16−bit register address to which a WRITE will take place. This transfer takes place as two 8−bit sequences and the slave sends an acknowledge bit after each sequence to indicate that the byte has been received. The master will then transfer the 16−bit data, as two 8−bit sequences and the slave sends an acknowledge bit after each sequence to indicate that the byte has been received. The master stops writing by generating a (re)start or stop condition. If the request was a READ, the master sends the 8−bit write slave address/data direction byte and 16−bit register address, just as in the write request. The master then generates a (re)start condition and the 8−bit read slave address/data direction byte, and clocks out the register data, 8 bits at a time. The master generates an acknowledge bit after each 8−bit transfer. The data transfer is stopped when the master sends a no−acknowledge bit.

Single READ from Random Location

Figure 13 shows the typical READ cycle of the host to the AP0201AT. The first two bytes sent by the host are an internal 16−bit register address. The following 2−byte READ cycle sends the contents of the registers to host.

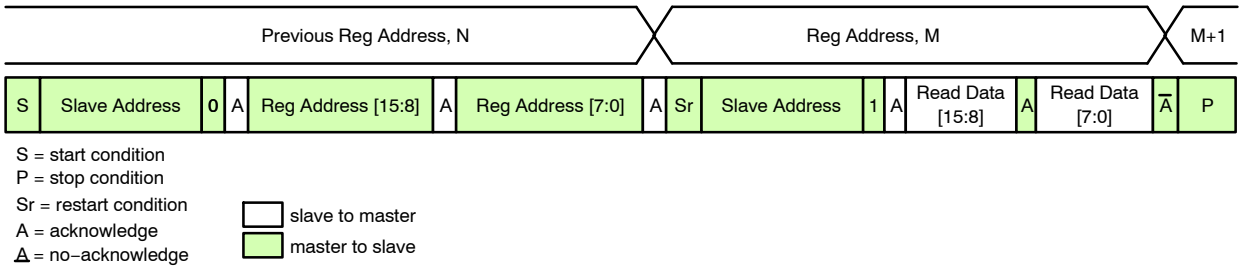


Figure 13. Single READ from Random Location

Single READ from Current Location

Figure 14 shows the single READ cycle without writing the address. The internal address will use the previous address value written to the register.

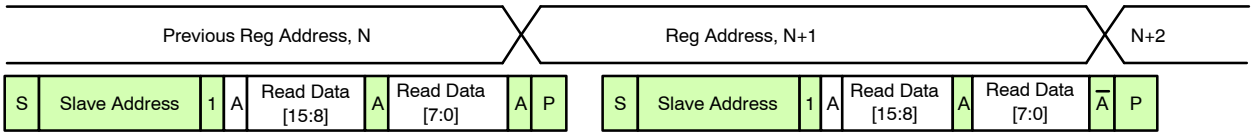


Figure 14. Single Read from Current Location

Sequential READ, Start from Random Location

This sequence (Figure 15) starts in the same way as the single READ from random location (Figure 13). Instead of generating a no−acknowledge bit after the first byte of data has been transferred, the master generates an acknowledge bit and continues to perform byte READs until “L” bytes have been read.

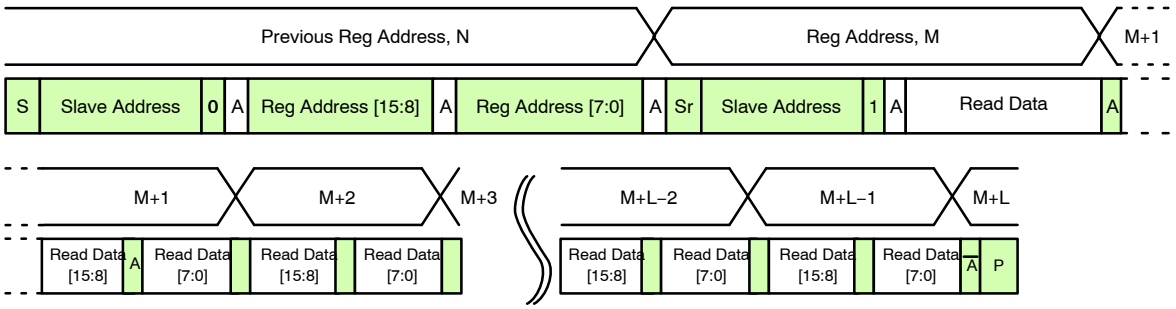


Figure 15. Sequential READ, Start from Random Location

Sequential READ, Start from Current Location

This sequence (Figure 16) starts in the same way as the single READ from current location (Figure 14). Instead of generating a no−acknowledge bit after the first byte of data has been transferred, the master generates an acknowledge bit and continues to perform byte reads until “L” bytes have been read.

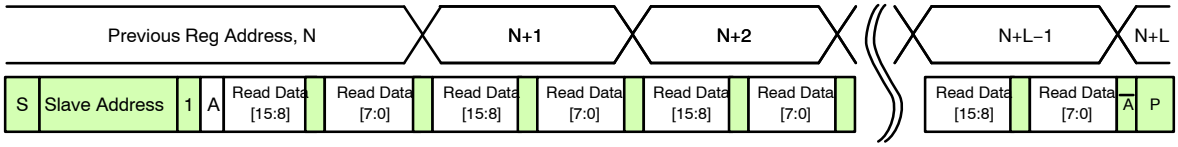


Figure 16. Sequential READ, Start from Current Location

Single Write to Random Location

Figure 17 shows the typical WRITE cycle from the host to the AP0201AT. The first 2 bytes indicate a 16−bit address of the internal registers with most−significant byte first. The following 2 bytes indicate the 16−bit data.

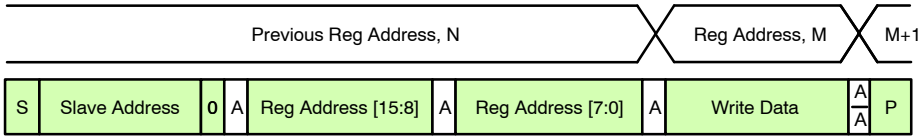


Figure 17. Sequential READ, Start from Current Location

Sequential WRITE, Start at Random Location

This sequence (Figure 18) starts in the same way as the single WRITE to random location (Figure 17). Instead of generating a no−acknowledge bit after the first byte of data has been transferred, the master generates an acknowledge bit and continues to perform byte writes until “L” bytes have been written. The WRITE is terminated by the master generating a stop condition.

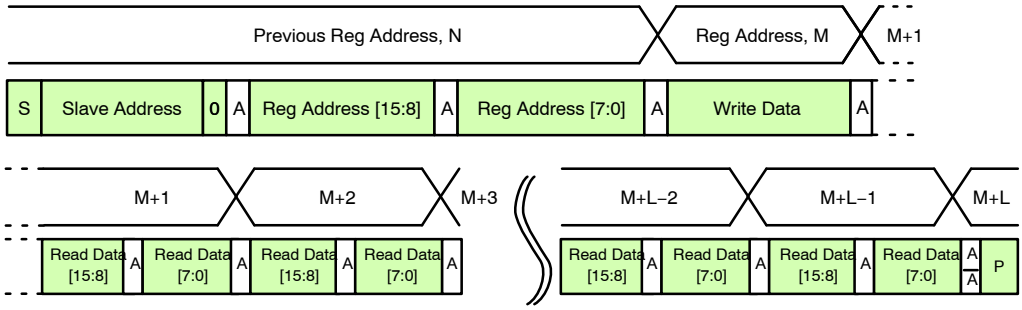


Figure 18. Sequential WRITE, Start from Current Location

ETHERNET INTERFACE

Overview

AP0201AT's Ethernet mode has complete support for configuring the part and streaming video out. By default IPv4 and IPv6 are enabled. The default MAC address and IP address are in the table below. For a complete set of default settings and configurable variables refer to the AP0201AT Register Reference. AP0201AT supports time precision protocols to synchronize several cameras.

Type	Default Address
MAC	2:00:00:00:00:01
IPv4	192.168.1.5
IPv6	fe80::ff:fe00:0

AP0201AT also supports a UDP−based protocol with **onsemi** host commands. Using this protocol all registers and variables are accessible.

AP0201AT MAC supports MII, GMII and RMII protocols. Video can be streamed in MJPEG or H.264 format over RTP/RTSP or AVB (IEEE 1722a) protocols.

Registers and variables can be modified using HCI access commands over Ethernet or through the serial interface. For initial settings of these variables, firmware will check NVM before booting up. (Please refer to the network section of the HCI document for a discussion of all Ethernet firmware and hardware variables.)

Features

- Ethernet hardware
  - Supports 100 Mb/s and 1000 Mb/s data rates
  - Supporting MII/RMII/GMII interfaces for PHYs
- UDP−based command protocol
- Patching firmware and updating the settings over Ethernet and I2C
- VLAN support for Video packets
- Muti−camera synchronization using time precision protocol

Protocol

The AP0201AT supports the following Ethernet protocols.

- TCP/IP Stack
  - IPv4 − Fully supported. ICMP, IGMP, ARP, UDP and TCP
  - IPv6 − The AP0200AT supports IPv6 with H.264 using Annex−B, but not RFC−6184 headers.
- Command Protocols
  - onsemi UDP−based Command Protocol
  - Real Time Streaming Protocol (RTSP)
- Streaming Protocol:
  - AVB (IEEE 1722a−2011)
  - Real−time Transport Protocol (RTP)
- Video Compression Protocol:
  - H.264 − profiles supported: constrained baseline (intra frames only) and high10intra level required: up to 5.0 Both Annex B and RFC6184 formatting are supported.
  - MJPEG − RFC2435
- Precision Time Protocol:
  - EEE 1588−2008 (PTPv2): Messages transport over UDP.
    - Delay Mechanism
      End−to−End (Multicast)
      Peer−to−Peer (Multicast)
    - PTP Supported Feature
      Master Mode for Test only
      Slave (default)
      Management message
      Best Master Clock (BMC) Algorithm
  - IEEE 802.1as (gPTP): Messages transport in layer−2
- IP Configuration Protocols
  - DHCP4 − IP address, Subnet Mask, Domain Name, and Host Name can be updated via DHCP4
  - DHCP6
- Custom Diagnostic Protocol − **onsemi** command protocol can be used to retrieve diagnostic information from the hardware. Refer to the AP0201AT Technical Note that describes the protocol in detail.
- Hybrid operation mode: While the AP201 is in Ethernet mode, all the configurations including patches can be applied using the serial connection. Nevertheless, the video will stream out over Ethernet. This feature is useful for customer who use a micro−controller and does not want to connect a NVM (flash/eeprom) to the part.
- Metadata UDP packets: AP201AT sends out four extra UDP packets for every frame. These packets include the frame's state and embedded rows.
- Proxy Service: AP0201AT can open up to 4 UDP sockets over the serial interface. The AP0201AT can then send or receive packets over a user−specified port using these sockets. If proxy service is enabled, AP0201AT does not manipulate packet payload for these sockets and is transparent for these sockets. There are special set of host commands that customer can use over the serial interface to open a socket, send data, receive data and close sockets.

For additional information, please refer to the technical note TN−09−333: AP0200AT Ethernet Quick Start Guide.

Metadata

In Ethernet mode AP0201AT sends out four lines of metadata over Ethernet for every frame. Each line is carried in a separate UDP packet. Costumer can specify the destination port and IP address. By default, the destination IP address and port are set to 255.255.255.255 and 50010.

The UDP packet's payload consists of following data bytes:

1. Four byte of Timestamp to match the frame.
2. Four byte of line number. AP0201AT sends line numbers zero and one at the beginning of frame. Line numbers two and three are sent at the end of frame.
3. Metadata itself.

Supported PHYs

The AP0201AT is compatible with most 100 Mbs MII PHYs. We have specifically tested with the following automotive−qualified PHYs:

- BroadR Reach BCM89810, BCM89811
- Micrel KMZ8051MN, KSZ8081MNXIA
- DP83848C
- TJA1100

The AP0201AT also supports 1Gb Ethernet operation and has been tested with the Micrel KSZ9031MNX PHY.

The AP0201AT is also compatible with most RMII PHYs by using the Universal PHY patch. For all RMII, MII, and GMII PHYs not listed here, please check with your support team for details.

HOST COMMAND INTERFACE

The AP0201AT has a mechanism to write higher level commands, the Host Command Interface (HCI). Once a command has been written through the HCI, it will be executed by on chip firmware and the results are reported back. EEPROM or Flash memory is also available to store commands for later execution.

Full details of the Host Command Interface can be found in the AP0201AT Host Command Interface (HCI) Specification document.

OPENS SOURCE SOFTWARE INFORMATION

Portions of the 1588 support code are licensed from a third party that requires the following notice:

Copyright (c) 2011−2012 George V. Neville−Neil,
Steven Kreuzer,
Martin Burnicki,
Jan Breuer,
Gael Mace,
Alexandre Van Kempen,
Inaqui Delgado,
Rick Ratzel,
National Instruments.

Copyright (c) 2009−2010 George V. Neville−Neil,
Steven Kreuzer,
Martin Burnicki,
Jan Breuer,
Gael Mace,
Alexandre Van Kempen

Copyright (c) 2005−2008 Kencall Correll, Aidan Williams

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS ”AS IS” AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;

LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The views and conclusions contained in the software and documentation are those of the authors and should not be interpreted as representing official policies, either expressed or implied, of the FreeBSD Project.

SPECIFICATIONS

Caution: Stresses greater than those listed in Table 11 may cause permanent damage to the device. This is a stress rating only, and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied.

Exposure to absolute maximum rating conditions for extended periods may affect reliability.

Table 11. ABSOLUTE MAXIMUM RATINGS

Parameter	Rating		Unit
	Min	Max	
Digital power (V_{DD}_REG)	−0.3	4.95	V
Host I/O power (V_{DD}IO_H)	−0.3	4.95	V
Sensor I/O power (V_{DD}IO_S)	−0.3	4.2	V
PLL power (V_{DD}_PLL)	−0.3	1.8	V
Digital core power (V_{DD})	−0.3	1.8	V
OTPM power (V_{DD}IO_OTPM)	−0.3	4.95	V
HiSPi power (V_{DD}IO_PHY)	−0.3	4.2	V
DC Input Voltage	−0.3	V_{DD}IO_*+0.3	V
DC Output Voltage	−0.3	V_{DD}IO_*+0.3	V
Storage Temperature	−50	150	°C

Stresses exceeding those listed in the Maximum Ratings table may damage the device. If any of these limits are exceeded, device functionality should not be assumed, damage may occur and reliability may be affected.

Table 12. ELECTRICAL CHARACTERISTICS AND OPERATING CONDITIONS

Parameter	Min	Typ	Max	Unit
Supply input to on−chip regulator (V_{DD}_REG)	1.71	1.8	1.89	V
Host I/O voltage (V_{DD}IO_H)	1.71	1.8/2.8/3.3	3.46	V
Sensor I/O voltage (V_{DD}IO_S)	1.71	1.8/2.8	2.94	V
Core voltage (V_{DD})	1.14	1.2	1.26	V
PLL voltage (V_{DD}_PLL)	1.14	1.2	1.26	V
HiSPi PHY voltage (V_{DD}_PHY)	2.3	2.8	2.94	V
OTPM power supply (V_{DD}IO_OTPM)	2.38	2.5/3.3	3.47	V
Functional operating temperature (ambient – T_A)	−40		105	°C

Product parametric performance is indicated in the Electrical Characteristics for the listed test conditions, unless otherwise noted. Product performance may not be indicated by the Electrical Characteristics if operated under different conditions.

I/O TIMING

MII I/O Timing

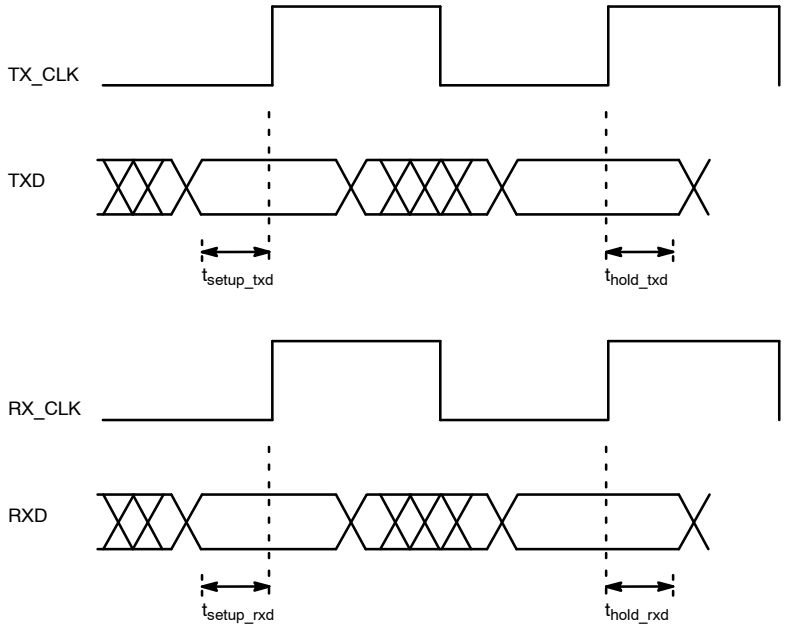


Figure 19. MII I/O Timing Diagram

Table 13. MII I/O TIMING CHARACTERISTICS

Symbol	Parameter	Conditions	Min	Max	Unit
Freq	TX_CLK and RX_CLK Input clocks	1.8 V to 3.3 V, 25 pF load	25–110 ppm	25+110 ppm	MHz
t_{setup_txd}	TXD[3:0], TX_ERR, TX_EN, setup to TX_CLK input rise	1.8 V to 3.3 V, 25 pF load	15.0		ns
t_{hold_txd}	TXD[3:0], TX_ERR, TX_EN hold from TX_CLK input rise	1.8 V to 3.3 V, 25 pF load	2.0		ns
t_{setup_rxd}	RXD[3:0], RX_ERR, CRS_DV setup to RX_CLK input rise	1.8 V to 3.3 V	8.0		ns
t_{hold_rxd}	RXD[3:0], RX_ERR, CRS_DV hold from RX_CLK input rise	1.8 V to 3.3 V	8.0		ns

RMII I/O Timing

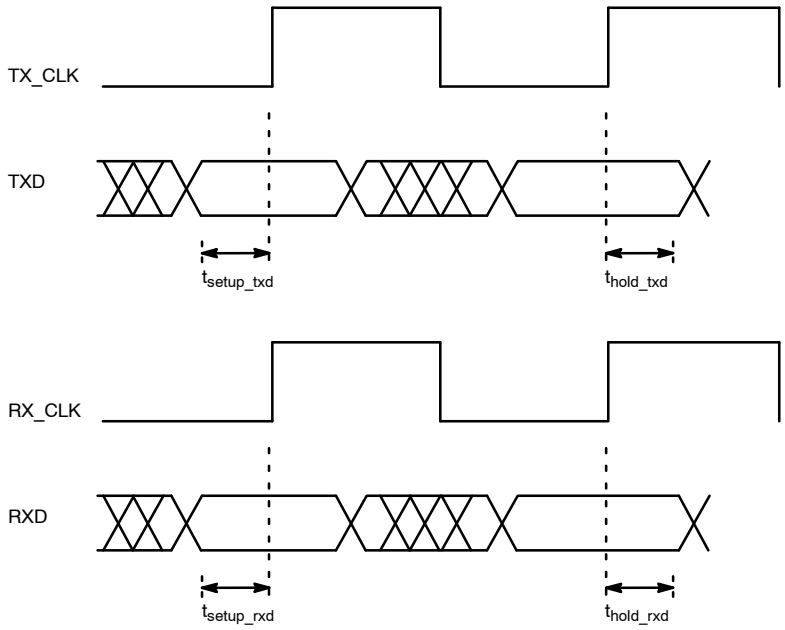


Figure 20. RMII I/O Timing Diagram

Table 14. RMII I/O TIMING CHARACTERISTICS

Symbol	Parameter	Conditions	Min	Max	Unit
Freq	TX_CLK and RX_CLK Input clocks	1.8 V to 3.3 V, 25 pF load	50 – 110 ppm	50 + 110 ppm	MHz
t_{setup_txd}	TXD[1:0], TX_ERR, TX_EN setup to TX_CLK input rise	1.8 V to 3.3 V, 25 pF load	4.0		ns
t_{hold_txd}	TXD[1:0], TX_ERR, TX_EN hold from TX_CLK input rise	1.8 V to 3.3 V, 25 pF load	2.0		ns
t_{setup_rxd}	RXD[1:0], RX_ERR, CRS_DV setup to RX_CLK input rise	1.8 V to 3.3 V	4.0		ns
t_{hold_rxd}	RXD[1:0], RX_ERR, CRS_DV hold from RX_CLK input rise	1.8 V to 3.3 V	2.0		ns

GMII I/O Timing

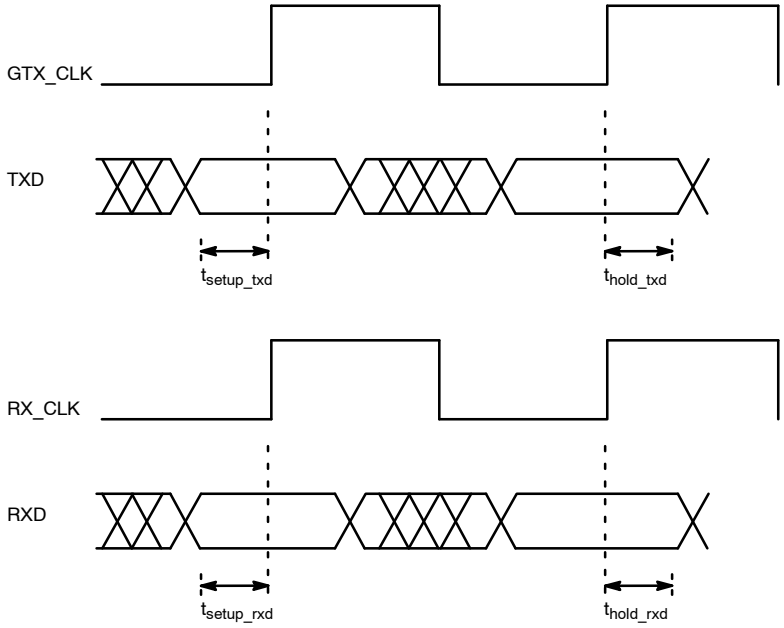


Figure 21. GMII I/O Timing Diagram

Table 15. GMII I/O TIMING CHARACTERISTICS

Symbol	Parameter	Conditions	Min	Max	Unit
Freq	GTX_CLK output frequency	2.5 V to 3.3 V, 10 pF load	125 – 110 ppm	125 + 110 ppm	MHz
Freq	TXD[7:0], TX_EN, TX_ERR setup to GTX_CLK input rise	2.5 V to 3.3 V, 10 pF load	2.5		ns
t_{setup_txd}	TXD[7:0], TX_EN, TX_ERR hold from GTX_CLK output rise	2.5 V to 3.3 V, 10 pF load	0.5		ns
t_{hold_txd}	RXD[7:0], RX_ERR, CRS_DV setup to RX_CLK input rise	2.5 V to 3.3 V	2.3		ns
t_{setup_rxd}	RXD[7:0], RX_ERR, CRS_DV hold from RX_CLK input rise	2.5 V to 3.3 V	0.3		ns

ELECTRICAL CHARACTERISTICS

Table 16. DC ELECTRICAL CHARACTERISTICS

Symbol	Parameter	Condition	Min	Max	Unit	Notes
V_{IH}	Input HIGH voltage		$V_{DD}IO\_H$ or $V_{DD}IO\_S$*0.8	–	V	7
V_{IL}	Input LOW voltage		–	$V_{DD}IO\_H$ or $V_{DD}IO\_S$*0.2	V	7
I_{IN}	Input leakage current	V_{IN} = 0 V or V_{IN} = $V_{DD}IO\_H$ or $V_{DD}IO\_S$*		10	μA	8
V_{OH}	Output HIGH voltage		$V_{DD}IO\_H$ or $V_{DD}IO\_S$*0.80	–	V	
V_{OL}	Output LOW voltage		–	$V_{DD}IO\_H$ or $V_{DD}IO\_S$*0.2	V	

7. V_{IL} and V_{IH} have min/max limitations specified by absolute ratings.
8. Excludes pins that have internal PU resistors.

Table 17. INPUT CLOCKS

Clock	Min (MHz)	Typical (MHz)	Max (MHz)	Description
EXTCLK	10 – osc 20 – xtal	27	29	Primary system clock. Drives PLLs. Crystal frequency range is 20 – 29 MHz, otherwise 10 – 29 MHz.
PIXCLK_IN	10	74.25	80	Clock for parallel input bus (from sensor).
HISPI_CLK	30		300	Clock for HISPI image data receiver.

Table 18. OUTPUT CLOCKS

Clock	Min (MHz)	Typical (MHz)	Max (MHz)	Description
MCLK	10	27	29	Primary clock to sensor. Equals EXTCLK.
GTX_CLK	18	74.25/25	80/125	Clock of parallel output bus. If pad voltage is 1.8 V nominal, then max frequency is 80 MHz. If 2.5 V or 3.3 V, then 125 MHz. See electrical specs.
SPI_CLK	1		20	SPI clock to nonvolatile external memory.

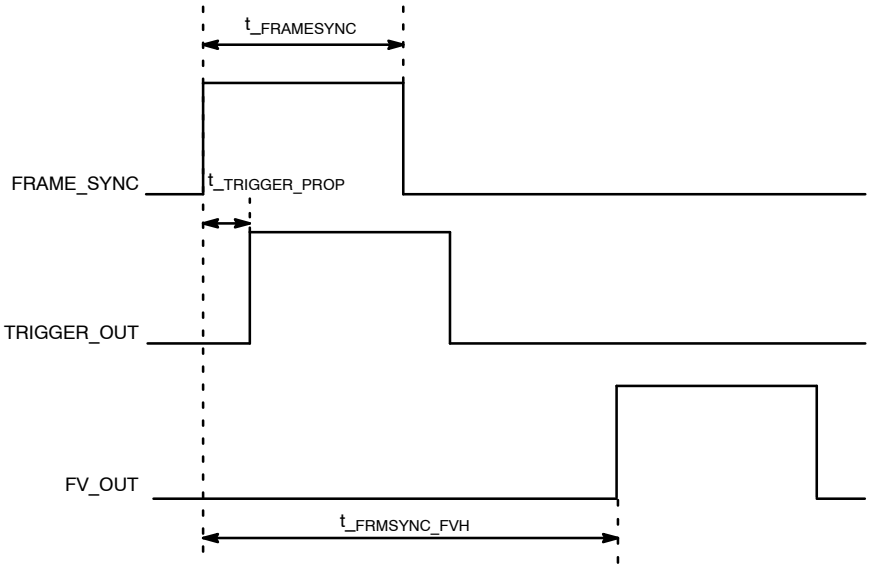


Figure 22. Frame_Sync Diagram

Table 19. TRIGGER TIMING

Parameter	Name	Conditions	Min	Typ	Max	Unit
FRAME_SYNC to FV_OUT	$t_{FRMSYNC\_FVH}$		8 lines + exposure time + sensor delay	–	–	Lines
FRAME_SYNC to TRIGGER_OUT	$t_{TRIGGER\_PROP}$		–	–	30	ns
t_{FRAME_SYNC}	$t_{FRAMESYNC}$		3	–	–	EXTCLK cycles

Table 20. STANDBY CURRENT CONSUMPTION

(Default Setup Conditions: f_{EXTCLK} = 27 MHz, V_{DD}_REG = 1.8 V, V_{DD}IO_H not included in measurement, V_{DD}IO_S = 1.8 V, V_{DD}IO_OTPM = 2.8 V, V_{DD}_PHY = 2.8 V, T_A = 105°C unless otherwise stated)

Parameter	Condition	Typ	Max	Unit
IDD_REG		1.50	1.91	mA
IDDIO_S		0.19	0.24	mA

Table 20. STANDBY CURRENT CONSUMPTION
(Default Setup Conditions: f_{EXTCLK} = 27 MHz, V_{DD}_REG = 1.8 V, V_{DD}IO_H not included in measurement, V_{DD}IO_S = 1.8 V, V_{DD}IO_OTPM = 2.8 V, V_{DD}_PHY = 2.8 V, T_A = 105°C unless otherwise stated)

Parameter	Condition	Typ	Max	Unit
IDDIO_H		1.20	1.52	mA
IDDIO_OTPM		0.18	0.23	mA
IDD_PHY		0.00	0.1	mA
Total Standby Power		7.46	94.7	mW
Total Standby Power	t_{EXTCLK} = 27 MHz	3.50		mA

Table 21. INRUSH CURRENT

Supply	Voltage	Typ	Max	Unit
V_{DD}IO_H	2.8/3.3	140	190	mA
V_{DD}IO_S	1.8	90	105	mA
V_{DD}IO_REG	1.8	130	180	mA
V_{DD}IO_OTPM	2.8/3.3	140	160	mA
V_{DD}_PHY	2.8	180	180	mA

Table 22. OPERATING CURRENT CONSUMPTION – SENSOR PARALLEL OUTPUT
(Default Setup Conditions: f_{EXTCLK} = 27 MHz, V_{DD}_REG = 1.8 V, V_{DD}IO_H not included in measurement, V_{DD}IO_S = 1.8 V, V_{DD}IO_OTPM = 2.5 V, V_{DD}_PHY = 2.5 V, T_A = 105°C unless otherwise noted)

Symbol	Conditions	Min	Typ	Max	Unit
V_{DD}_REG	MJPEG		85	108	mA
	H.264		110	140	mA
V_{DD}IO_S	MJPEG		3	4	mA
	H.264		3	4	mA
V_{DD}IO_H	MJPEG		35		mA
	H.264		35		mA
V_{DD}IO_OTPM	MJPEG		0.2	0.3	mA
	H.264		0.2	0.3	mA
V_{DD}_PHY	MJPEG		0	0.1	mA
	H.264		0	0.1	mA
Total power consumption	MJPEG		159	202	mW
	H.264		204	259	mW

Table 23. OPERATING CURRENT CONSUMPTION – SENSOR HISPI OUTPUT
(Default Setup Conditions: f_{EXTCLK} = 27 MHz, V_{DD}_REG = 1.8 V, V_{DD}IO_H not included in measurement, V_{DD}IO_S = 1.8 V, V_{DD}IO_OTPM = 2.5 V, V_{DD}_PHY = 2.5 V, T_A = 105°C unless otherwise stated)

Symbol	Conditions	Min	Typ	Max	Unit
I_{DD}IO_REG	H.264		110	140	mA
I_{DD}IO_S	H.264		3	4	mA
I_{DD}IO_H	H.264		35		mA
I_{DD}IO_OTPM	H.264		0.2	0.3	mA
I_{DD}_PHY	H.264		0.3	0.4	mA
Total power consumption	H.264		212	270	mW

TWO−WIRE SERIAL REGISTER INTERFACE

The electrical characteristics of the two−wire serial register interface (SCLK, SDATA) are shown in Figure 23 and Table 24.

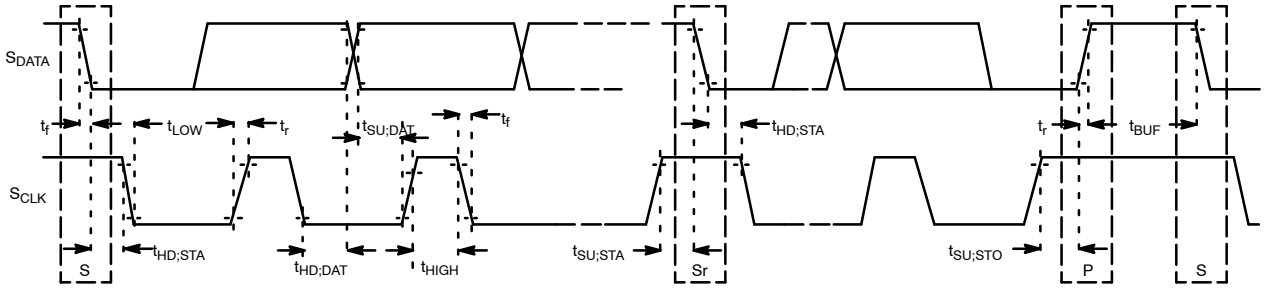


Figure 23. Slave Two Wire Serial Bus Timing Parameters (CCIS)

Table 24. SLAVE TWO−WIRE SERIAL BUS CHARACTERISTICS (CCIS)

(Default Setup Conditions: f_{EXTCLK} = 27 MHz, $V_{DD}IO\_H = V_{DD}\_OTPM$ = 2.8 V, $V_{DD}\_REG = V_{DD}IO\_S$ = 1.8 V, T_A = 25°C unless otherwise stated)

Parameter	Symbol	Standard Mode		Fast Mode		Fast Mode Plus		Unit
		Min	Max	Min	Max	Min	Max	
SCLK Clock Frequency	fSCL	0	100	0	400	0	1000	kHz
Hold time (repeated) START condition: After this period, the first clock pulse is generated	tHD;STA	4.0	–	0.6	–	0.26	–	μs
LOW period of the SCLK clock	tLOW	4.7	–	1.3	–	0.5	–	μs
HIGH period of the SCLK clock	tHIGH	4.0	–	0.6	–	0.26	–	μs
Set−up time for a repeated START condition	tSU;STA	4.7	–	0.6	–	0.26	–	μs
Data hold time	tHD;DAT	0 (Note 10)	3.45 (Note 11)	0	0.9 (Note 11)	0	–	μs
Data set−up time	tSU;DAT	250	–	100	–	50	–	ns
Rise time of both SDATA and SCLK signals (10−90%)	tr	–	1000	20 + 0.1Cb (Note 12)	300	20 + 0.1Cb (Note 12)	300	ns
Fall time of both SDATA and SCLK signals (10−90%)	tf	–	300	20 + 0.1Cb (Note 12)	300	20 + 0.1Cb (Note 12)	300	ns
Set−up time for STOP condition	tSU;STO	4.0	–	0.6	–	0.26	–	μs
Bus free time between a STOP and START condition	tBUF	4.7	–	1.3	–	0.5	–	μs
Capacitive load for each bus line	Cb	–	400	–	400	–	550	pF
Serial interface input pin capacitance	CIN_SI	–	3.3	–	3.3	–	3.3	pF
SDATA max load capacitance	CLOAD_SD	–	30	–	30	–	30	pF
SDATA pull−up resistor	RSD	1.5	4.7	1.5	4.7	1.5	4.7	kΩ

9. All values referred to VIHmin = 0.9 $V_{DD}IO$ and VILmax = 0.1 $V_{DD}IO$ levels. EXCLK = 27 MHz.
10. A device must internally provide a hold time of at least 300 ns for the S_{DATA} signal to bridge the undefined region of the falling edge of S_{CLK}.
11. The maximum $t_{HD;DAT}$ has only to be met if the device does not stretch the LOW period (t_{LOW}) of the S_{CLK} signal.
12. Cb = total capacitance of one bus line in pF.
13. 1 MHz is supported as well, but requires a 25 MHz EXTCLK.

The electrical characteristics of the master two−wire serial interface (M_SCLK, M_SDATA) are shown in Figure 24 and Table 25.

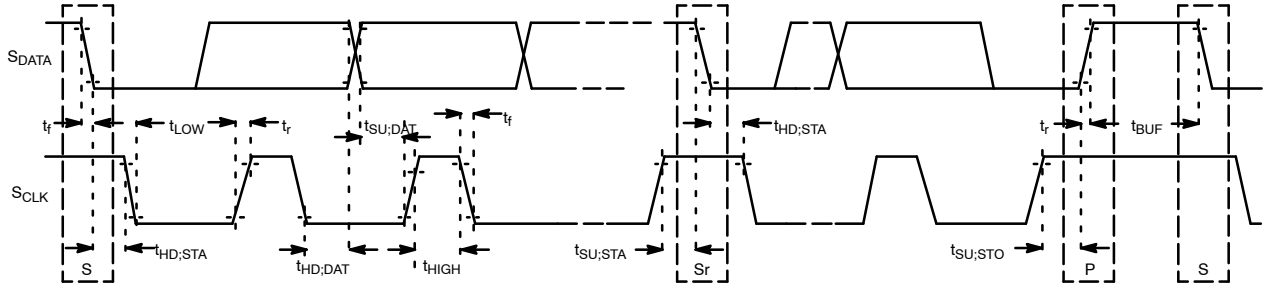


Figure 24. Master Two Wire Serial Bus Timing Parameters (CCIM)

Table 25. MASTER TWO−WIRE SERIAL BUS CHARACTERISTICS (CCIM)
(Default Setup Conditions: f_{EXTCLK} = 27 MHz, $V_{DD}IO\_H = V_{DD}\_OTPM$ = 2.8 V, $V_{DD}\_REG = V_{DD}IO\_S$ = 1.8 V, T_A = 25°C unless otherwise stated)

Parameter	Symbol	Standard Mode		Fast Mode		Unit
		Min	Max	Min	Max	
M_S_{CLK} Clock Frequency	f_{SCL}	0	100	0	400	KHz
Hold time (repeated) START condition: After this period, the first clock pulse is generated.	$t_{HD;STA}$	4.0		0.6		μs
LOW period of the M_S_{CLK} clock	t_{LOW}	4.7		1.2		μs
HIGH period of the M_S_{CLK} clock	t_{HIGH}	4.0		0.6		μs
Set−up time for a repeated START condition	$t_{SU;STA}$	4.7		0.6		μs
Data hold time	$t_{HD;DAT}$	0 (Note 15)	3.45 (Note 16)	0	0.9 (Note 16)	μs
Data set−up time	$t_{SU;DAT}$	250		100		ns
Rise time of both M_S_{DATA} and M_S_{CLK} signals (10−90%)	t_r		1000	20+0.1Cb (Note 17)	300	ns
Fall time of both M_S_{DATA} and M_S_{CLK} signals (10−90%)	t_f		300	20+0.1Cb (Note 17)	300	ns
Set−up time for STOP condition	$t_{SU;STO}$	4.0		0.6		μs
Bus free time between a STOP and START condition	t_{BUF}	4.7		1.3		μs
Capacitive load for each bus line	Cb		400		400	pF
Serial interface input pin capacitance	C_{IN_SI}		3.3		3.3	pF
M_S_{DATA} max load capacitance	C_{LOAD_SD}		30		30	pF
M_S_{DATA} pull−up resistor	R_{SD}	1.5	4.7	1.5	4.7	KΩ

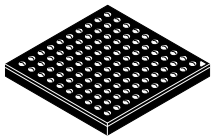
14. All values referred to VIHmin = 0.9 V_{DD}IO and VILmax = 0.1 V_{DD}IO levels. EXCLK = 27 MHz.
15. A device must internally provide a hold time of at least 300 ns for the M_S_{DATA} signal to bridge the undefined region of the falling edge of M_S_{CLK}.
16. The maximum $t_{HD;DAT}$ has only to be met if the device does not stretch the LOW period (t_{LOW}) of the M_S_{CLK} signal.
17. Cb = total capacitance of one bus line in pF.

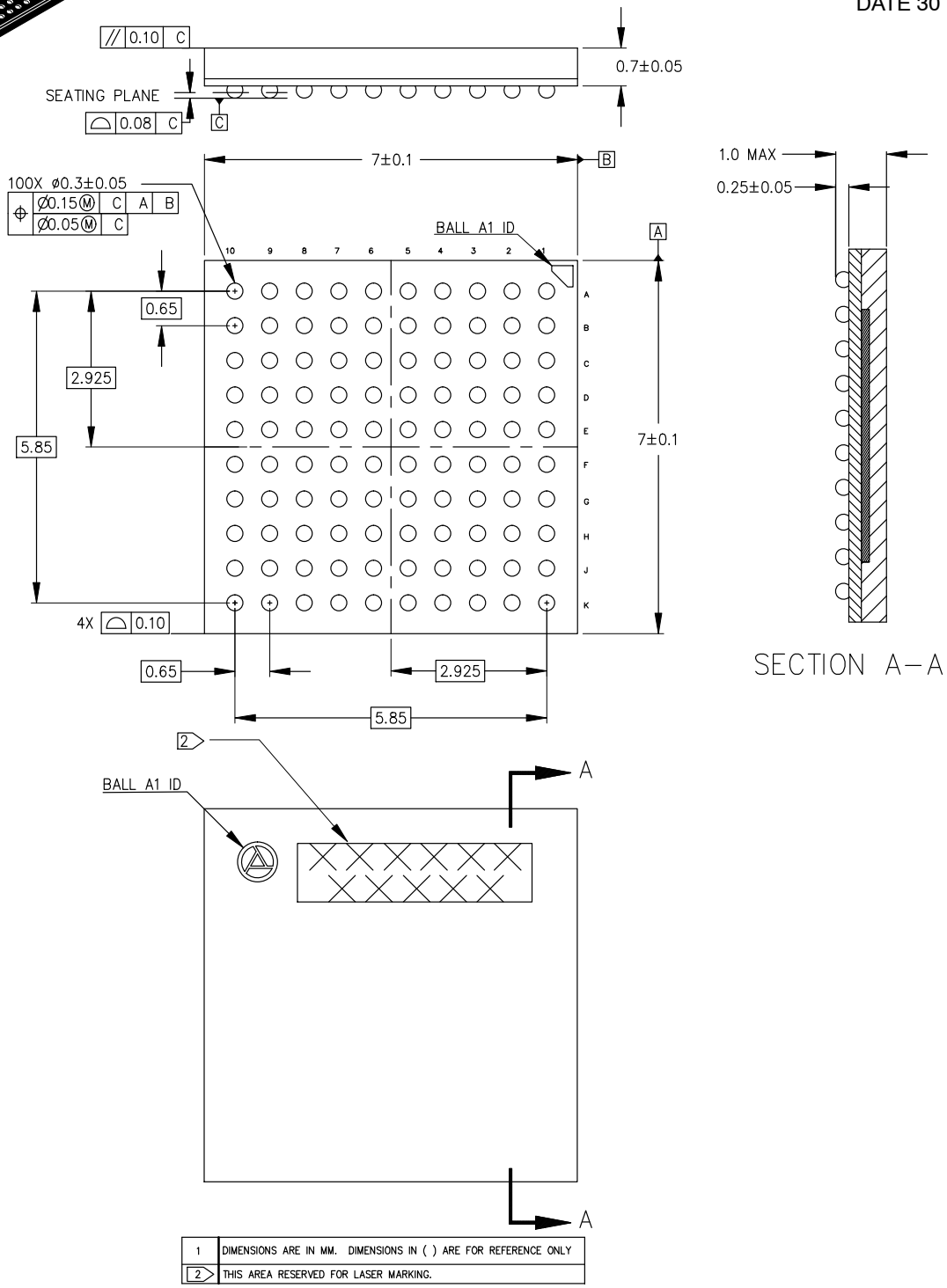
MECHANICAL CASE OUTLINE
PACKAGE DIMENSIONS

ON Semiconductor®

VFBGA100 7x7
CASE 138AH
ISSUE O

DATE 30 DEC 2014





1	DIMENSIONS ARE IN MM. DIMENSIONS IN ( ) ARE FOR REFERENCE ONLY
2	THIS AREA RESERVED FOR LASER MARKING.

DOCUMENT NUMBER:	**98AON93700F**	Electronic versions are uncontrolled except when accessed directly from the Document Repository. Printed versions are uncontrolled except when stamped "CONTROLLED COPY" in red.
DESCRIPTION:	**VFBGA100 7X7**	**PAGE 1 OF 1**

PUBLICATION ORDERING INFORMATION

LITERATURE FULFILLMENT:
Email Requests to: orderlit@onsemi.com

onsemi Website: www.onsemi.com

TECHNICAL SUPPORT
North American Technical Support:
Voice Mail: 1 800−282−9855 Toll Free USA/Canada
Phone: 011 421 33 790 2910

Europe, Middle East and Africa Technical Support:
Phone: 00421 33 790 2910
For additional information, please contact your local Sales Representative

◊