CMOS MT9V111 SOC Camera Module

1/4-Inch 0.3-Megapixel Module Datasheet

Rev 1.0, Nov. 2013

Table of Contents

1 Introduction

The Aptina® Imaging MT9V111 is a 1/4-inch VGA-format CMOS active-pixel digital image sensor, the result of combining the MT9V011 image sensor core with Micron Imaging's third-generation digital image flow processor technology. The MT9V111 has an active imaging pixel array of 649 x 489, capturing high-quality color images at VGA resolution. The sensor is a complete camera-on-a-chip solution and is designed specifically to meet the demands of battery-powered products such as cellular phones, PDAs, and toys. It incorporates sophisticated camera functions on-chip and is programmable through a simple two-wire serial interface.

2 Features

- Optical size 1/4 inch
- Resolution 640x480 VGA
- Onboard regulator, only single 3.3V supply needed
- Standard 0.1inch (2.54mm) pin pitch header connector
- Mounted with high quality F1.8 / 6mm lens
- Ultra low-power, low cost CMOS image sensor Superior low-light performance
- Up to 30 fps progressive scan at 27 MHz for high quality video at VGA resolution
- On-chip Image Flow Processor (IFP) performs sophisticated processing: color recovery and correction, sharpening, gamma, lens shading correction, on-the-fly defect correction, 2X fixed zoom
- Image decimation to arbitrary size with smooth, continuous zoom and pan
- Automatic exposure, white balance and black compensation, flicker avoidance, color saturation, and defect identification and correction, auto frame rate, back light compensation
- Two-wire serial programming interface
- ITU_R BT.656 (YCbCr), YUV, 565RGB, 555RGB, and 444RGB output data formats

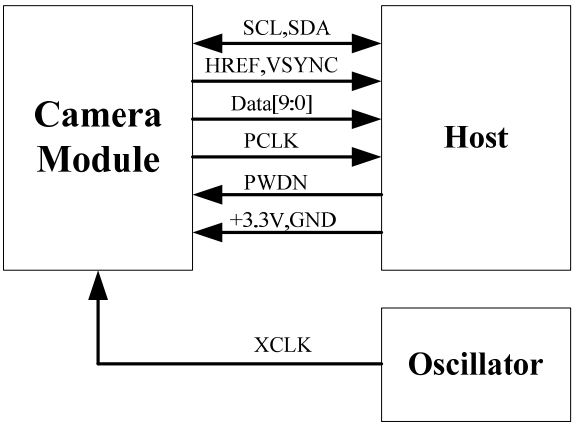
3 Key Specifications

Parameter		Typical Value
Optical Format		1/4-inch (4:3)
Active Imager Size		3.58mm(H) x 2.69mm(V) 4.48mm (Diagonal)
Active Pixels		640H x 480V (VGA)
Pixel Size		5.6um x 5.6um
Color Filter Array		RGB Bayer Pattern
Shutter Type		Electronic Rolling Shutter (ERS)
Maximum Data Rate/ Master Clock		12–13.5 MPS/24–27 MHz
Frame Rate	VGA (640 x 480)	15 fps at 12 MHz (default), programmable up to 30 fps at 27 MHz
	CIF (352 x 288)	Programmable up to 60 fps
	QVGA (320 x 240)	Programmable up to 90 fps
ADC Resolution		10-bit, on-chip
Responsivity		1.9 V/lux-sec (550nm)
Dynamic Range		60dB
SNR_{MAX}		45dB
Supply Voltage		2.8V ±0.25V
Power Consumption		<80mW at 2.8V, 15 fps at 12MHz
Operating Temperature		-20°C to +60°C
Packaging		44-Ball ICSP, wafer or die

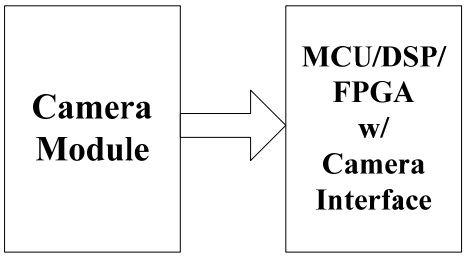
4 Application

- Cellular phones
- PDAs
- Toys
- Other battery-powered products
- Can be used in Arduino, Maple, ChipKit, STM32, ARM, DSP, FPGA platforms

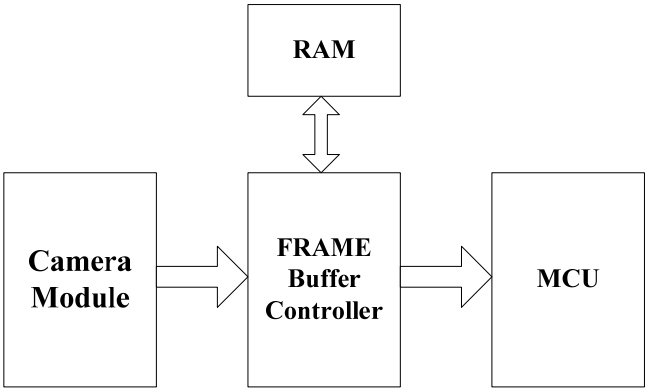
The following schematic diagram show a basic camera based system. The camera module is powered from a single +3.3V power supply. An external oscillator provide the clock source for camera module XCLK pin. With proper configuration to the camera internal registers via I2C bus, then the camera supply pixel clock (PCLK) and camera data (Data[9:0]) back to the host with synchronize signal like HREF and VSYNC.



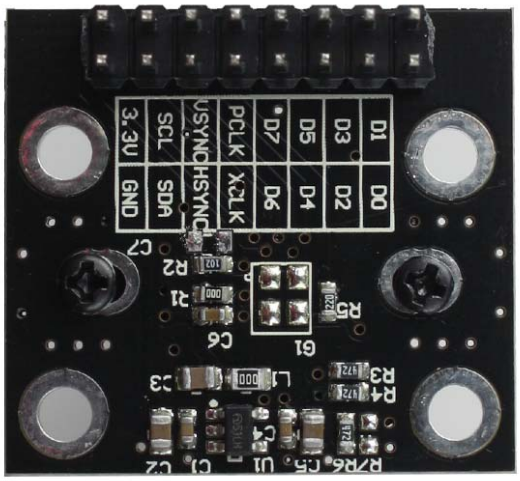
The host may have integrate camera interface like STM32F2 or STM32F4 series MCUs, or ARM9/11 which has dedicate camera port, and DPS like TI TMS320DM series, as well as FPGAs that user can design special logic for camera application. The typical connection between these system and camera module would show like following diagram.



For the host that doesn't have a dedicate camera interface, additional hardware is needed. User need to buffer a entire frame before read them out with low speed MCUs. For example ArduCAM shield is a additional hardware that can be connected to Arduino UNO/Mega board, user can take a photo or something like that easily. The following diagram show the system without dedicate camera interface.



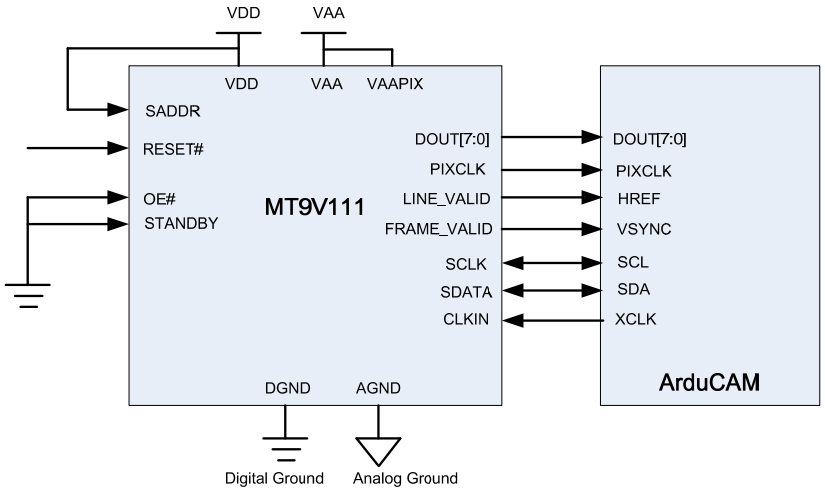
5 Pin Definition



Pin No.	PIN NAME	TYPE	DESCRIPTION
1	VCC	POWER	3.3v Power supply
2	GND	Ground	Power ground
3	SCL	Input	Two-Wire Serial Interface Clock
4	SDATA	Bi-directional	Two-Wire Serial Interface Data I/O
5	VSYNC	Output	Active High: Frame Valid; indicates active frame
6	HSYNC	Output	Active High: Line/Data Valid; indicates active pixels
7	PCLK	Output	Pixel Clock output from sensor
8	XCLK	Input	Master Clock into Sensor
9	DOUT7	Output	Pixel Data Output 7 (MSB)
10	DOUT6	Output	Pixel Data Output 6
11	DOUT5	Output	Pixel Data Output 5
12	DOUT4	Output	Pixel Data Output 4
13	DOUT3	Output	Pixel Data Output 3
14	DOUT2	Output	Pixel Data Output 2
15	DOUT1	Output	Pixel Data Output 1
16	DOUT0	Output	Pixel Data Output 0 (LSB)

6 Typical Application

This section describes how to connect MT9V111 module with ArduCAM shield, the connection diagram list as follows.



I2C Slave Address: 0xB8 (SADDR=1), 0x90 (SADDR=0)

Due to the LCD screen on the ArduCAM shield is limited to 320x240, user have to configure the module to output 320x240 resolution.

Please note that the MT9V111 pin SADDR is connected to VCC inside the module, so the slave I2C address is 0xB8. The initialized setting for 320x240 RGB565 output format is listed in the following table.

```
const struct sensor_reg MT9V111_QVGA_30fps[] PROGMEM =
{

    {0x01,   0x04},        //REG=4
    {64,     488 },        // anti-eclipse Vref
    {47,     63408},       //set for Fmclk>13.5MHz
    {2,  11},
    {3,      487},
    {4,  646},
    {0x06,   70 },         //vertical blank
    {32,     0x4000},

    {0x01,   1},       //REG=1
    {0x08,   0xd802}, //RGB mode
    {58,     0x0001},
    {165,    0x8000},      // [QVGA 1:1 zoom]
    {166,    0x8280},      //
    {167,    0x8140},      //
    {168,    0x8000},      //
    {169,    0x81e0},      //
```

```
    {170,    0x00f0},    //

    //[Auto Lens Correction Setup]
    {0x2B, 0x0022},   //(36) GREEN1_GAIN_REG
    {0x2C, 0x003E},   //(48) BLUE_GAIN_REG
    {0x2D, 0x0020},   //(53) RED_GAIN_REG
    {0x2E, 0x0022},   //(36) GREEN2_GAIN_REG
    {0x02, 0x0000},   //(18) BASE_MATRIX_SIGNS
    {0x03, 0x3923},   //(11) BASE_MATRIX_SCALE_K1_K5
    {0x04, 0x0724},   //(10) BASE_MATRIX_SCALE_K6_K9
    {0x06, 0xB00C},   //(14) MODE_CONTROL
    {0x09, 0x0080},   //(3) BASE_MATRIX_COEF_K1
    {0x0A, 0x0000},   //(2) BASE_MATRIX_COEF_K2
    {0x0B, 0x0000},   //(2) BASE_MATRIX_COEF_K3
    {0x0C, 0x0000},   //(1) BASE_MATRIX_COEF_K4
    {0x0D, 0x0080},   //(3) BASE_MATRIX_COEF_K5
    {0x0E, 0x0000},   //(2) BASE_MATRIX_COEF_K6
    {0x0F, 0x0000},   //(2) BASE_MATRIX_COEF_K7
    {0x10, 0x0000},   //(2) BASE_MATRIX_COEF_K8
    {0x11, 0x0080},   //(3) BASE_MATRIX_COEF_K9
    {0x15, 0x0000},   //(21) DELTA_COEFS_SIGNS
    {0x16, 0x0000},   //(3) DELTA_MATRIX_COEF_D1
    {0x17, 0x0000},   //(2) DELTA_MATRIX_COEF_D2
    {0x18, 0x0000},   //(2) DELTA_MATRIX_COEF_D3
    {0x19, 0x0000},   //(1) DELTA_MATRIX_COEF_D4
    {0x1A, 0x0000},   //(3) DELTA_MATRIX_COEF_D5
    {0x1B, 0x0000},   //(2) DELTA_MATRIX_COEF_D6
    {0x1C, 0x0000},   //(2) DELTA_MATRIX_COEF_D7
    {0x1D, 0x0000},   //(2) DELTA_MATRIX_COEF_D8
    {0x1E, 0x0000},   //(4) DELTA_MATRIX_COEF_D9
    {0x25, 0x0514},   //(1) AWB_SPEED_SATURATION
    {0x34, 0x0010},   //(5) LUMA_OFFSET
    {0x35, 0xF010},   //(10) CLIPPING_LIM_OUT_LUMA
    {0x48, 0x0080},   //(1) TEST_PATTERN_GEN
    {0x53, 0x0804},   //(10) GAMMA_Y1_Y2
    {0x54, 0x2010},   //(10) GAMMA_Y3_Y4
    {0x55, 0x6040},   //(10) GAMMA_Y5_Y6
    {0x56, 0xA080},   //(10) GAMMA_Y7_Y8
    {0x57, 0xE0C0},   //(10) GAMMA_Y9_Y10
    {0xff, 0xffff },
};
```