Arducam Mega SPI Camera Series

Mega SPI Camera Series Application Note

September, 2023

Catalog

1. Introduction

This application note describes the detail hardware messages of Arducam Mega SPI Camera Module.

2. SPI Slave Interface

Arducam Mega SPI slave interface is fixed SPI mode 0 with POL = 0 and PHA= 0. The recommended speed of SCLK is 8MHz. Also note that the performance may vary across different platforms. The SPI protocol is designed with a command phase with variable data phase. The chip select signal should always keep asserted during the SPI read or write bus cycle.

3. Arducam Mega Timing Diagram

3.1 Single Read Timing

The SPI bus single read timing is for read operation of Arducam Mega internal registers and single FIFO read function. It is composed of a command phase and a data phase during the assertion of chip select signal CSn. The first 8 bits is the command byte which is decoded as a register address. The second 8 bits is the dummy data, which is used to provide a delay area in a very short time to prepare data for the camera. The final 8 bits is written to the SPI bus MOSI signal, and the content read back from register is appeared on the SPI bus MISO signal.

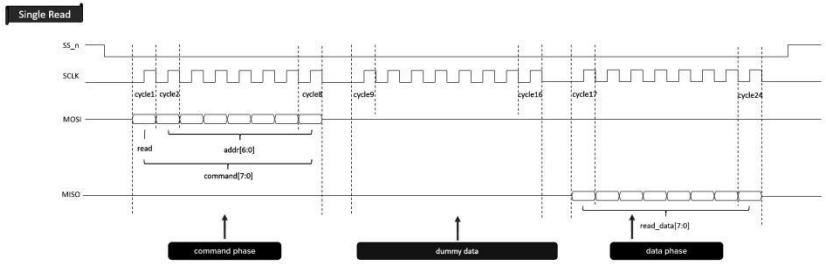

Figure1. Single Read Timing

3.2 Single Write Timing

The SPI bus write timing composed of a command phase and a data phase during the assertion of the chip select signal CSn. The first 8 bits is command byte which is decoded as a register address, and the second 8 bits is data byte to be written to the Arducam Mega internal registers.

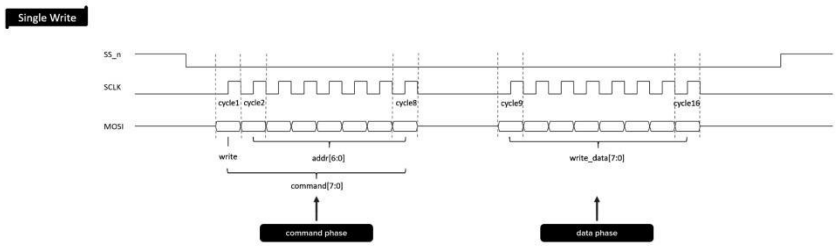

Figure2. Single Write Timing

3.3 First Time Burst Read Timing

SPI bus burst read timing only applies to burst FIFO read operations. It consists of a burst read command phase and multiple data phases to achieve multiple times throughput compared to a single FIFO read operation. Similar to a single read

mode, when formally burst reading data, the first byte is the dummy data(Allow enough time for data preparation) and the subsequent data is valid.

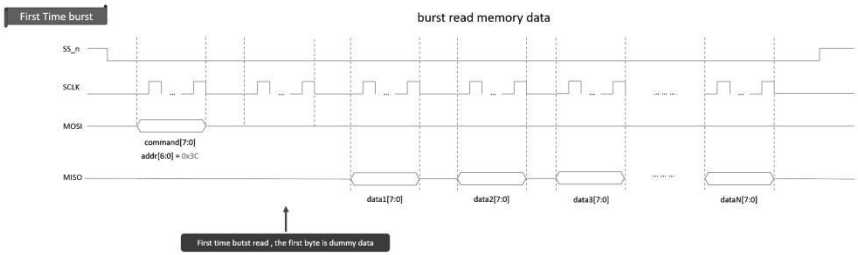

Figure3. First Time Burst Read Timing

3.4 Nth Burst Read Timing(N>=2)

Different from the First Time burst read mode, when the Nth burst read command is issued, the data will start to be prepared directly, and all subsequent data received will be valid data.

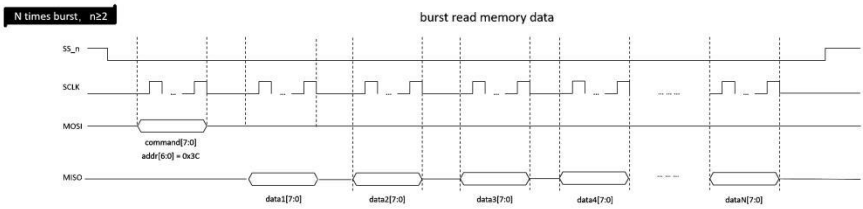

Figure4. Nth Burst Read Timing

4. Register Table

Table1. Arducam Mega SPI Camera Register Table

FPGA Register Address(8-bit)	Register Type	MEGA-5MP Camera	MEGA-3MP Camera	Default Value
0x00	RW	Test register	Test register	0x00
0x01	RW	Bit[7:0]: Number of shooting frames 0~254: Number of frames = value + 1 (unless full) 255: Indicates that the memory is full (8MB)		0x00
0x02	RW	Bit[2]:cam_power_en. 1: Normal. 0: Power off Bit[1]: cam_pwdn. 1: Sleep. 0: normal Bit[0]: cam_rst_n. 1: Normal. 0: reset		0x05
0x04	RW	memory control Bit[1]: Write 1 to start taking pictures Bit[0]: Write 1 to clear the write memory completion flag		Default
0x05	RW	Bit[7]: Select camera data or simulation data 0: camera data 1: Simulated data Bit[1]: In 16-bit mode, convert the input 10 bits into 16 bits. Choose whether to fill the high bit or the low bit with 0 0: Fill in the high 6 bits with 0 1: Fill in the lower 6 bits with 0		0x00

		Bit[0]: 8-bit mode or 16-bit mode 0: 8-bit mode 1: 10-bit mode		
0x06	RW	Bit[7]: test data (32-bit counter) 0: normal data 1: Test data Bit[0]: VSYN field signal level is valid high or low 0: Highly effective 1: low effective		0x01
0x07	RW	Bit[7]: Write 1, reset cache (SDRAM, 8M) Bit[6]: Write 1 to reset FPGA Bit[1]: Write 1 to reset I2C Bit[0]: Write 1 to initiate an I2C direct read camera operation.		Default
0x0A	RW	Directly write camera register, I2C device address		Default
0x0B	RW	Directly write camera register, the upper 8 bits of the I2C register address		Default
0x0C	RW	Directly write camera register, the lower 8 bits of the I2C register address		Default
0x20	WO	Bit[1:0]: write basic configuration 0: Write basic configuration 1: Write JPG basic configuration 2: Write RGB basic configuration 3: Write YUV basic configuration		Default
0x21	WO	Bit[6:0]: Resolution 1:320x240 2: 640x480	Bit[6:0]: Resolution 1:320x240 2: 640x480	Default

		4: 1280x720 6: 1600x1200 7: 1920x1080 9:2592x1944 10:96x96 11:128x128 12: 320x320	4: 1280x720 6: 1600x1200 7: 1920x1080 8:2048x1536 10:96x96 11:128x128 12: 320x320	
0x22	WO	Bit[3:0]: brightness adjustment 0: default 1: +1 2: -1 3: +2 4: -2 5: +3 6: -3 7: +4 8: -4		Default
0x23	WO	Bit[3:0]: Contrast adjustment 0: default 1: +1 2: -1 3: +2 4: -2 5: +3 6: -3		Default
0x24	WO	Bit[2:0]: Saturation adjustment 0: default 1: +1 2: -1 3: +2		Default

		4: -2 5: +3 6: -3		
0x25	WO	Bit[2:0]: Exposure compensation adjustment 0: default 1: +1 2: -1 3: +2 4: -2 5: +3 6: -3		Default
0x26	WO	Bit[2:0]: White balance mode 0: automatic 1: Daylight 2: Office 3: cloudy day 4: Indoor		Default
0x27	WO	Bit[3:0]: Special effect selection 0: No special effects 1: cool color 2: Warm color 3: black and white 4: yellowing 5: reverse color 6: Greenish	Bit[3:0]: Special effect selection 0: No special effects 1: cool color 2: Warm color 3: black and white 4: yellowing 5: reverse color 6: Greenish 9: light yellow	Default
0x28	WO	-	Bit[2:0]: Sharpness adjustment	Default

			0: automatic 1: Sharpness 1 2: Sharpness 2 3: Sharpness 3 4: Sharpness 4 5: Sharpness 5 6: Sharpness 6 7: Sharpness 7 8: Sharpness 8	
0x29	WO	Bit[1:0]: Autofocus 0: Turn on the autofocus base configuration (the configuration is very long; Default single focus) 1: Single autofocus 2: Continuous autofocus 3: Pause autofocus 4: Turn off autofocus	-	Default
0x2A	WO	Bit[1:0]: JPG mode image quality 0: High quality 1: Medium quality 2: Low quality		Default
0x30	WO	Bit[7]: Turn on/off automatic mode 1: Turn on Automatic 0: Turn off automatic Bit[1:0]:		Default

		0: Automatic gain 1: Auto exposure 2: Auto white balance	
0x31	WO	Bit[1:0]: Manual gain [9:8]	Default
0x32	WO	Bit[7:0]: Manual gain [7:0]	Default
0x33	WO	Bit[3:0]: Manual exposure [19:16]	Default
0x34	WO	Bit[7:0]: Manual exposure [15:8]	Default
0x35	WO	Bit[7:0]: Manual exposure [7:0]	Default

5. Brief of Mega SDK

Arducam Mega SDK is a C and C++ package, containing convenience classes and functions that help in most common tasks while using Arducam Mega API. We support both C API and C++ API. The SDK contains HAL layer and Arducam Mega Cam protocol layer and API.

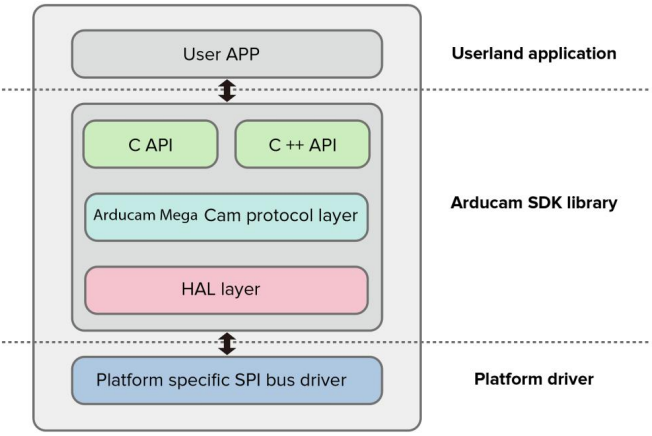

Figure5. Arducam Mega SDK Framework

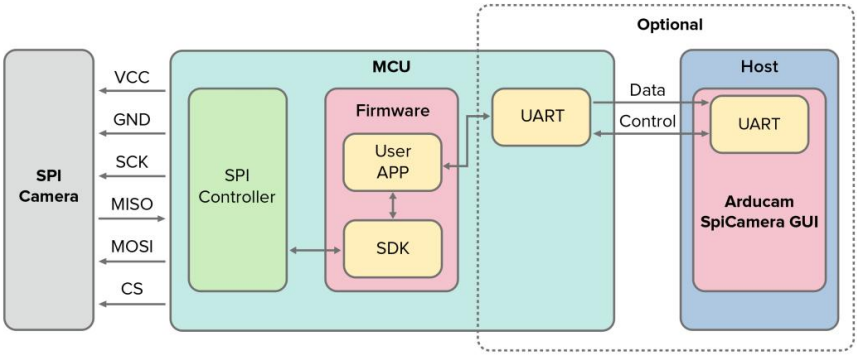

Figure6. Typical System Block

For more information about Arducam Mega SDK, API, and Arducam Mega GUI, please refer to the following link:
https://www.arducam.com/docs/arducam-mega/arducam-mega-getting-started/index.html