



SigmaStar Camera Timer 使用参考



© 2019 SigmaStar Technology Corp. All rights reserved.

SigmaStar Technology makes no representations or warranties including, for example but not limited to, warranties of merchantability, fitness for a particular purpose, non-infringement of any intellectual property right or the accuracy or completeness of this document, and reserves the right to make changes without further notice to any products herein to improve reliability, function or design. No responsibility is assumed by SigmaStar Technology arising out of the application or use of any product or circuit described herein; neither does it convey any license under its patent rights, nor the rights of others.

SigmaStar is a trademark of SigmaStar Technology Corp. Other trademarks or names herein are only for identification purposes only and owned by their respective owners.



REVISION HISTORY

Revision No.	Description	Date
{000001}	• {Initial release}	{05/06/2019}



TABLE OF CONTENTS

REVISION HISTORY	i
TABLE OF CONTENTS.....	ii
1. 使用说明	1
1.1. Enable Timer	1
1.2. Set Divide and Maximum.....	1
2. REGISTER TABLE	2
2.1. TIMER0 Register (Bank = 30).....	2
2.2. TIMER1 Register (Bank = 30).....	4
2.3. TIMER2 Register (Bank = 30).....	6

1. 使用说明

1.1. Enable Timer

1. Enable Timer engine – set "TIMER_EN" to 1
2. Trigger counter – set "TIMER_TRIG" to 1.
3. Read timer from register "TIMER_CAP[31:0]"

1.2. Set Divide and Maximum

1. Set the "divide" value to "TIMER_DIVIDE[7:0]" for timer counter frequency
2. Set the "max" value to "TIMER_MAX[31:0]".
3. Enable timer
4. If counter over the "TIMER_MAX[31:0]", then "TIMER_HIT" will assert.

2. REGISTER TABLE

2.1. TIMER0 Register (Bank = 30)

TIMER0 Register (Bank = 30)				
Index (Absolute)	Mnemonic	Bit	Description	
10h (3020h)	REG3020	7:0	Default : 0x00	Access : R/W
	-	7:2	Reserved.	
	TIMER_TRIG	1	set: Enable timer counting one time (from 0 to max, then stop). clear: By reset itself OR set reg_timer_en.	
	TIMER_EN	0	set: Enable timer counting rolled (from 0 to max, then rolled). clear: By reset itself OR set reg_timer_trig.	
10h (3021h)	REG3021	7:0	Default : 0x00	Access : R/W
	-	7:1	Reserved.	
	TIMER_INT_EN	0	set: Enable interrupt. clear: By reset itself.	
11h (3022h)	REG3022	7:0	Default : 0x00	Access : RO
	-	7:1	Reserved.	
	TIMER_HIT	0	assert: When counter enabled and matches reg_timer_max. deassert: By write 1 OR set reg_timer_en, reg_timer_once, reg_timer_max.	
12h (3024h)	REG3024	7:0	Default : 0xFF	Access : R/W
	TIMER_MAX[7:0]	7:0	Timer maximum value.	
12h (3025h)	REG3025	7:0	Default : 0xFF	Access : R/W
	TIMER_MAX[15:8]	7:0	See description of '3024h'.	
13h (3026h)	REG3026	7:0	Default : 0xFF	Access : R/W
	TIMER_MAX[23:16]	7:0	See description of '3024h'.	
13h (3027h)	REG3027	7:0	Default : 0xFF	Access : R/W
	TIMER_MAX[31:24]	7:0	See description of '3024h'.	
14h (3028h)	REG3028	7:0	Default : 0x00	Access : RO
	TIMER_CAP[7:0]	7:0	Timer current value.	

TIMER0 Register (Bank = 30)				
Index (Absolute)	Mnemonic	Bit	Description	
			Note: With non-32-bit-data system, please read from LSB.	
14h (3029h)	REG3029	7:0	Default : 0x00	Access : RO
	TIMER_CAP[15:8]	7:0	See description of '3028h'.	
15h (302Ah)	REG302A	7:0	Default : 0x00	Access : RO
	TIMER_CAP[23:16]	7:0	See description of '3028h'.	
15h (302Bh)	REG302B	7:0	Default : 0x00	Access : RO
	TIMER_CAP[31:24]	7:0	See description of '3028h'.	
16h (302Ch)	REG302C	7:0	Default : 0x00	Access : R/W
	TIMER_DIVIDE[7:0]	7:0	timer divide counter number 8'b0000: timer counter = clk_xiu/1 8'b0001: timer counter = clk_xiu/2 8'b0010: timer counter = clk_xiu/3 and so on	

2.2. TIMER1 Register (Bank = 30)

TIMER1 Register (Bank = 30)				
Index (Absolute)	Mnemonic	Bit	Description	
20h (3040h)	REG3040	7:0	Default : 0x00	Access : R/W
	-	7:2	Reserved.	
	TIMER_TRIG	1	set: Enable timer counting one time (from 0 to max, then stop). clear: By reset itself OR set reg_timer_en.	
	TIMER_EN	0	set: Enable timer counting rolled (from 0 to max, then rolled). clear: By reset itself OR set reg_timer_trig.	
20h (3041h)	REG3041	7:0	Default : 0x00	Access : R/W
	-	7:1	Reserved.	
	TIMER_INT_EN	0	set: Enable interrupt. clear: By reset itself.	
21h (3042h)	REG3042	7:0	Default : 0x00	Access : RO
	-	7:1	Reserved.	
	TIMER_HIT	0	assert: When counter enabled and matches reg_timer_max. deassert: By write 1 OR set reg_timer_en, reg_timer_once, reg_timer_max.	
	22h (3044h)	REG3044	7:0	Default : 0xFF
	TIMER_MAX[7:0]	7:0	Timer maximum value.	
22h (3045h)	REG3045	7:0	Default : 0xFF	Access : R/W
	TIMER_MAX[15:8]	7:0	See description of '3044h'.	
23h (3046h)	REG3046	7:0	Default : 0xFF	Access : R/W
	TIMER_MAX[23:16]	7:0	See description of '3044h'.	
23h (3047h)	REG3047	7:0	Default : 0xFF	Access : R/W
	TIMER_MAX[31:24]	7:0	See description of '3044h'.	
24h (3048h)	REG3048	7:0	Default : 0x00	Access : RO
	TIMER_CAP[7:0]	7:0	Timer current value. Note: With non-32-bit-data system, please read from LSB.	
24h (3049h)	REG3049	7:0	Default : 0x00	Access : RO
	TIMER_CAP[15:8]	7:0	See description of '3048h'.	
25h	REG304A	7:0	Default : 0x00	Access : RO



TIMER1 Register (Bank = 30)			
Index (Absolute)	Mnemonic	Bit	Description
(304Ah)	TIMER_CAP[23:16]	7:0	See description of '3048h'.
25h	REG304B	7:0	Default : 0x00 Access : RO
(304Bh)	TIMER_CAP[31:24]	7:0	See description of '3048h'.
26h	REG304C	7:0	Default : 0x00
(304Ch)	TIMER_DIVIDE[7:0]	7:0	timer divide counter number 8'b0000: timer counter = clk_xiu/1 8'b0001: timer counter = clk_xiu/2 8'b0010: timer counter = clk_xiu/3 and so on

2.3. TIMER2 Register (Bank = 30)

TIMER2 Register (Bank = 30)				
Index (Absolute)	Mnemonic	Bit	Description	
30h (3060h)	REG3060	7:0	Default : 0x00	Access : R/W
	-	7:2	Reserved.	
	TIMER_TRIG	1	set: Enable timer counting one time (from 0 to max, then stop). clear: By reset itself OR set reg_timer_en.	
	TIMER_EN	0	set: Enable timer counting rolled (from 0 to max, then rolled). clear: By reset itself OR set reg_timer_trig.	
30h (3061h)	REG3061	7:0	Default : 0x00	Access : R/W
	-	7:1	Reserved.	
	TIMER_INT_EN	0	set: Enable interrupt. clear: By reset itself.	
31h (3062h)	REG3062	7:0	Default : 0x00	Access : RO
	-	7:1	Reserved.	
	TIMER_HIT	0	assert: When counter enabled and matches reg_timer_max. deassert: By write 1 OR set reg_timer_en, reg_timer_once, reg_timer_max.	
32h (3064h)	REG3064	7:0	Default : 0xFF	Access : R/W
	TIMER_MAX[7:0]	7:0	Timer maximum value.	
32h (3065h)	REG3065	7:0	Default : 0xFF	Access : R/W
	TIMER_MAX[15:8]	7:0	See description of '3064h'.	
33h (3066h)	REG3066	7:0	Default : 0xFF	Access : R/W
	TIMER_MAX[23:16]	7:0	See description of '3064h'.	
33h (3067h)	REG3067	7:0	Default : 0xFF	Access : R/W
	TIMER_MAX[31:24]	7:0	See description of '3064h'.	
34h (3068h)	REG3068	7:0	Default : 0x00	Access : RO
	TIMER_CAP[7:0]	7:0	Timer current value. Note: With non-32-bit-data system, please read from LSB.	
34h (3069h)	REG3069	7:0	Default : 0x00	Access : RO
	TIMER_CAP[15:8]	7:0	See description of '3068h'.	
35h	REG306A	7:0	Default : 0x00	Access : RO



TIMER2 Register (Bank = 30)			
Index (Absolute)	Mnemonic	Bit	Description
(306Ah)	TIMER_CAP[23:16]	7:0	See description of '3068h'.
35h (306Bh)	REG306B TIMER_CAP[31:24]	7:0	Default : 0x00 Access : RO See description of '3068h'.
36h (306Ch)	REG306C TIMER_DIVIDE[7:0]	7:0	Default : 0x00 timer divide counter number 8'b0000: timer counter = clk_xiu/1 8'b0001: timer counter = clk_xiu/2 8'b0010: timer counter = clk_xiu/3 and so on