Atmega16 寄存器

一. 引脚说明

表 1　引脚说明

引脚序号	引脚名称	引脚功能
1	PB5	8 位双向I/O 口，具有可编程的内部上拉电阻。其输出缓冲器具有对称的驱动特性，可以输出和吸收大电流。作为输入使用时，若内部上拉电阻使能，端口被外部电路拉低时将输出电流。在复位过程中，即使系统时钟还未起振，端口 A 处于高阻状态。
	MOSI	SPI 总线的主机输出/ 从机输入信号
2	PB6	8 位双向 I/O 口
	MISO	SPI 总线的主机输入/ 从机输出信号
3	PB7	8 位双向 I/O 口
	SCK	SPI 总线的串行时钟
4	RESET	复位输入引脚。持续时间超过最小门限时间的低电平将引起系统复位。
5	VCC	数字电路的电源
6	GND	地
7	XTAL2	反向振荡放大器的输出端
8	XTAL1	反向振荡放大器与片内时钟操作电路的输入端
9	PD0	8 位双向 I/O 口
	RXD	USART 输入引脚
10	PD1	8 位双向 I/O 口
	TXD	USART 输出引脚
11	PD2	8 位双向 I/O 口
	INT0	外部中断 0 的输入
12	PD3	8 位双向 I/O 口
	INT1	外部中断 1 的输入

13	PD4	8 位双向 I/O 口
13	OC1B	T/C1 输出比较 B 匹配输出
14	PD5	8 位双向 I/O 口
14	OC1A	T/C1 输出比较 A 匹配输出
15	PD6	8 位双向 I/O 口
15	ICP1	T/C1 输入捕捉引脚
16	PD7	8 位双向 I/O 口
16	OC2	T/C2 输出比较匹配输出
17	VCC	数字电路的电源
18	GND	地
19	PC0	8 位双向 I/O 口
19	SCL	两线串行总线时钟线
20	PC1	8 位双向 I/O 口
20	SDA	两线串行总线数据输入/ 输出线
21	PC2	8 位双向 I/O 口
21	TCK	JTAG 测试时钟
22	PC3	8 位双向 I/O 口
22	TMS	JTAG 测试模式选择
23	PC4	8 位双向 I/O 口
23	TDO	JTAG 测试数据输出
24	PC5	8 位双向 I/O 口
24	TDI	JTAG 测试数据输入
25	PC6	8 位双向 I/O 口
25	TOSC1	定时振荡器引脚 1
26	PC7	8 位双向 I/O 口
26	TOSC2	定时振荡器引脚 2
27	AVCC	端口 A 与 A/D 转换器的电源。。不使用 ADC 时，该引脚应直接与 VCC 连接。使用 ADC 时应通过一个低通滤波器与 VCC 连接。

28	AGND	A/D 的模拟地
29	AREF	A/D 的模拟基准输入引脚
30	PA7	8 位双向 I/O 口
	ADC7	ADC 输入通道 7
31	PA6	8 位双向 I/O 口
	ADC6	ADC 输入通道 6
32	PA5	8 位双向 I/O 口
	ADC5	ADC 输入通道 5
33	PA4	8 位双向 I/O 口
	ADC4	ADC 输入通道 4
34	PA3	8 位双向 I/O 口
	ADC3	ADC 输入通道 3
35	PA2	8 位双向 I/O 口
	ADC2	ADC 输入通道 2
36	PA1	8 位双向 I/O 口
	ADC1	ADC 输入通道 1
37	PA0	8 位双向 I/O 口
	ADC0	ADC 输入通道 0
38	VCC	数字电路的电源
39	GND	地
40	PB0	8 位双向 I/O 口
	T0	T/C0 外部计数器输入
	XCK	USART 外部时钟输入/ 输出
41	PB1	8 位双向 I/O 口
	T1	T/C1 外部计数器输入
42	PB2	8 位双向 I/O 口
	AIN0	模拟比较正输入
	INT2	外部中断 2 输入
43	PB3	8 位双向 I/O 口

	AIN1	模拟比较负输入
	OC0	T/C0 输出比较匹配输出
44	SS	SPI 从机选择引脚

二. I/O 端口

Atmega16单片机有32个通用I/O口,有PA~PD四组,每组都是8位。其主要的寄存器有DDRXn(X=A, B,C,D; n=0,2,…,7, 下同), PORTXn和PINXn。I/O组合设置见表1。

表1 I/O组合设置

DDRXn	PORTXn	I/O	上拉电阻	说明
0	0	输入	否	I/O三态输入
0	1	输入	是	I/O口带上拉电阻输入
1	0	输出	否	推免0输出
1	1	输出	否	推免1输出

1. PA 口寄存器

(1) DDRA

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
DDRA7	DDRA6	DDRA5	DDRA4	DDRA3	DDRA2	DDRA1	DDRA0

(2) PORTA

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PORTA7	PORTA6	PORTA5	PORTA4	PORTA3	PORTA2	PORTA1	PORTA0

(3) PINA

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PINA7	PINA6	PINA5	PINA4	PINA3	PINA2	PINA1	PINA0

2. PB 口寄存器

(1) DDRB

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
DDRB7	DDRB6	DDRB5	DDRB4	DDRB3	DDRB2	DDRB1	DDRB0

(2) PORTB

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

PORTB7	PORTB6	PORTB5	PORTB4	PORTB3	PORTB2	PORTB1	PORTB0

(3) PINB

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PINB7	PINB6	PINB5	PINB4	PINB3	PINB2	PINB1	PINB0

3. PC 口寄存器

(1) DDRC

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
DDRC7	DDRC6	DDRC5	DDRC4	DDRC3	DDRC2	DDRC1	DDRC0

(2) PORTC

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PORTC7	PORTC6	PORTC5	PORTC4	PORTC3	PORTC2	PORTC1	PORTC0

(3) PINC

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PINC7	PINC6	PINC5	PINC4	PINC3	PINC2	PINC1	PINC0

4. PD 口寄存器

(1) DDRD

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
DDRD7	DDRD6	DDRD5	DDRD4	DDRD3	DDRD2	DDRD1	DDRD0

(2) PORTD

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PORTD7	PORTD6	PORTD5	PORTD4	PORTD3	PORTD2	PORTD1	PORTD0

(3) PIND

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PIND7	PIND6	PIND5	PIND4	PIND3	PIND2	PIND1	PIND0

DDRX是方向寄存器，可读可写。在写操作时用于制定PX口是作为输入口还是输出口；在读操作时，从DDRX寄存器读出来的是端口的方向设定值。DDRX寄存器的初始值为0x00。

PORTX是数据寄存器，可读写。在写操作时，从PORTX写入的数据存入内部锁存器，以确定端口的工作状态或者将写入的数据送到外部数据总线。PORTX寄存器的初始值为0x00。

PINX用来访问端口X的逻辑值，且只允许读操作。从PINX读入的数据只是X口引脚的逻辑

状态。其初始值为高阻态。

三. 中断寄存器

中断是指微处理器在程序执行期间响应外部的事务请求，暂停当前的处理器工作转而处理外部事件，外部事件处理完成之后返回原来工作暂停的地方继续执行程序。

Atmega16具有21个中断源，每个中断源都有独立的中断向量作为中断服务程序的入口地址，而且所有的中断源都有自己独立的使能位。如果中断位I和中断向量使能位都置位，则中断发生，中断标志位置位，并执行中断服务程序。中断向量表见表2，处于越低地址的中断具有更高优先级。

表2 中断向量表

向量号	程序地址	中断源	描述
1	0x00	RESET	外部引脚电平引发的复位，上电复位，掉电检测复位，看门狗复位，以及JTAG AVR 复位
2	0x02	INT0	外部中断请求0
3	0x04	INT1	外部中断请求1
4	0x06	TIMER2 COMP	定时器/计数器2比较匹配
5	0x08	TIMER2 OVF	定时器/计数器2溢出
6	0x0a	TIMER1 CAPT	定时器/计数器1事件捕捉
7	0x0c	TIMER1 COMPA	定时器/计数器1比较匹配 A
8	0x0e	TIMER1 COMPB	定时器/计数器1比较匹配B
9	0x10	TIMER1 OVF	定时器/计数器1溢出
10	0x12	TIMER0 OVF	定时器/计数器0溢出
11	0x14	SPI, STC	SPI 串行传输结束
12	0x16	USART, RXC	USART, Rx 结束
13	0x18	USART, UDRE	USART数据寄存器空
14	0x1a	USART, TXC	USART, Tx 结束
15	0x1c	ADC	ADC 转换结束
16	0x1e	EE_RDY	EEPROM 就绪
17	0x20	ANA_COMP	模拟比较器
18	0x22	TWI	两线串行接口
19	0x24	INT2	外部中断请求2

20	0x26	TIMER0 COMP	定时器/ 计数器0 比较匹配
21	0x28	SPM_RDY	保存程序存储器内容就绪

1.状态寄存器 SREG

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
I	T	H	S	V	N	Z	C

I: 全局中断使能位。

在I置位后，单独的中断使能由不同的中断寄存器控制。若I为0，则禁止中断。

2.MCU 控制寄存器 MCUCR

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
SM2	SE	SM1	SM0	ISC11	ISC10	ISC01	ISC00

SE: MCU休眠使能位

SM1~SM0: MCU休眠模式选择

SM2	SM1	SM0	休眠模式
0	0	0	空闲
0	0	1	ADC 噪声抑制模式
0	1	0	掉电模式
0	1	1	省电模式
1	0	0	保留
1	0	1	保留
1	1	0	Standby(1) 模式
1	1	1	扩展Standby(1) 模式

ISC11~ISC10: 外部中断1的中断检测方式

ISC11	ISC10	INT1中断
0	0	低电平中断
0	1	INT1 引脚上任意的逻辑电平变化都将引发中断
1	0	下降沿中断
1	1	上升沿中断

ISC01～ISC00: 外部中断0的中断检测方式

ISC01	ISC00	INT0中断
0	0	低电平中断
0	1	INT1 引脚上任意的逻辑电平变化都将引发中断
1	0	下降沿中断
1	1	上升沿中断

3.通用中断屏蔽寄存器 GICR

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
INT1	INT0	INT2	–	–	–	IVSEL	IVCE

INT1:使能外部中断请求1

INT0:使能外部中断请求0

INT2:使能外部中断请求2

4.通用中断标志寄存器 GIFR

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
INTF1	INTF0	INTF2	–	–	–	IVSEL	IVCE

INTF1: 外部中断标志

INT1引脚电平发生跳变时触发中断请求，并置位相应的中断标志INTF1。如果SREG的位I以及GICR寄存器相应的中断使能位INT1为”1”，MCU即跳转到相应的中断向量。进入中断服务程序之后该标志自动清零。此外，标志位也可以通过写入”1” 来清零。

INTF0: 外部中断标志

INT0引脚电平发生跳变时触发中断请求，并置位相应的中断标志INTF0。如果SREG的位I以及GICR寄存器相应的中断使能位INT0为”1”，MCU即跳转到相应的中断向量。进入中断服务程序之后该标志自动清零。此外，标志位也可以通过写入”1” 来清零。

INTF2: 外部中断标志

INT2引脚电平发生跳变时触发中断请求，并置位相应的中断标志INTF1。如果SREG的位I以及GICR寄存器相应的中断使能位INT2为”1”，MCU即跳转到相应的中断向量。进入中断服务程序之后该标志自动清零。此外，标志位也可以通过写入”1” 来清零。

5.T/C 中断屏蔽寄存器 TIMSK

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
OCIE2	TOIE2	TICIE1	OCIE1A	OCIE1B	TOIE1	OCIE0	TOIE0

TOIE0: T/C0 溢出中断使能

当TOIE0 和状态寄存器的全局中断使能位I 都为”1” 时，T/C0 的溢出中断使能。当T/C0发生溢出，即TIFR 中的TOV0 位置位时，中断服务程序得以执行。

OCIE0: T/C0 输出比较匹配中断使能

当OCIE0 和状态寄存器的全局中断使能位I 都为”1” 时，T/C0 的输出比较匹配中断使能。当T/C0 的比较匹配发生，即TIFR 中的OCF0 置位时，中断服务程序得以执行。

TOIE1: T/C1 溢出中断使能

当TOIE1 和状态寄存器的全局中断使能位I 都为”1” 时，T/C1 的溢出中断使能。当T/C1发生溢出，即TIFR 中的TOV1 位置位时，中断服务程序得以执行

OCIE1B: T/C1 输出比较 B 匹配中断使能

当该位被设为"1”，且状态寄存器中的I 位被设为"1” 时，使能T/C1 的输出比较B 匹配中断使能。一旦TIFR 上的OCF1B 置位，CPU 即开始执行T/C1 输出比较B 匹配中断服务程序。

OCIE1A: 输出比较 A 匹配中断使能

当该位被设为"1”，且状态寄存器中的I 位被设为"1” 时， T/C1 的输出比较A 匹配中断使能。一旦TIFR 上的OCF1A 置位，CPU 即开始执行T/C1 输出比较A 匹配中断服务程序。

TICIE1: T/C1 输入捕捉中断使能

当该位被设为"1”，且状态寄存器中的I 位被设为"1” 时， T/C1 的输入捕捉中断使能。一旦TIFR 的ICF1 置位，CPU 即开始执行T/C1 输入捕捉中断服务程序。

TOIE2: T/C2 溢出中断使能

当TOIE2 和状态寄存器的全局中断使能位I 都为”1” 时，T/C2 的溢出中断使能。当T/C2发生溢出，即TIFR 中的TOV2 位置位时，中断服务程序得以执行。

OCIE2: T/C2 输出比较匹配中断使能

当OCIE2 和状态寄存器的全局中断使能位I 都为”1” 时，T/C2 的输出比较匹配中断使能。当T/C2 的比较匹配发生，即TIFR 中的OCF2 置位时，中断服务程序得以执行。

6.T/C 中断标志寄存器 TIFR

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
OCF2	TOV2	ICF1	OCF1A	OCF1B	TOV1	OCF0	TOV0

OCF0: 输出比较标志0

当T/C0 与OCR0(输出比较寄存器0) 的值匹配时，OCF0 置位。此位在中断服务程序里硬件清零，也可以对其写1 来清零。当SREG 中的位I、OCIE0(T/C0 比较匹配中断使能) 和OCF0 都置位时，中断服务程序得到执行。

TOV0: T/C0 溢出标志

当T/C0 溢出时， TOV0 置位。执行相应的中断服务程序时此位硬件清零。此外， TOV0 也可以通过写1 来清零。当SREG 中的位I、TOIE0(T/C0 溢出中断使能) 和TOV0 都置位时，中断服务程序得到执行。在相位修正PWM 模式中，当T/C0 在0x00 改变记数方向时， TOV0 置位。

TOV1: T/C1 溢出标志

该位的设置与T/C1 的工作方式有关。工作于普通模式和CTC 模式时，T/C1 溢出时TOV1置位。对工作在其它模式下的TOV1 标志位置位

OCF1B: T/C1 输出比较 B 匹配标志位

当TCNT1 与OCR1B 匹配成功时，该位被设为"1”。强制输出比较(FOC1B) 不会置位OCF1B。执行强制输出比较匹配B 中断服务程序时OCF1B 自动清零。也可以对其写入逻辑"1” 来清除该标志位。

OCF1A: T/C1 输出比较 A 匹配标志位

当TCNT1 与OCR1A 匹配成功时，该位被设为"1”。强制输出比较(FOC1A) 不会置位OCF1A。执行强制输出比较匹配A 中断服务程序时OCF1A 自动清零。也可以对其写入逻辑"1” 来清除该标志位。

ICF1: T/C1 输入捕捉标志位

外部引脚ICP1 出现捕捉事件时ICF1 置位。此外，当ICR1 作为计数器的TOP 值时，一旦计数器值达到TOP， ICF1 也置位。执行输入捕捉中断服务程序时ICF1 自动清零。也可以对其写入逻辑"1” 来清除该标志位。

OCF2: 输出比较标志2

当T/C0 与OCR0(输出比较寄存器0) 的值匹配时，OCF0 置位。此位在中断服务程序里硬件清零，也可以对其写1 来清零。当SREG 中的位I、OCIE0(T/C0 比较匹配中断使能) 和OCF0 都置位时，中断服务程序得到执行。

TOV2: T/C2 溢出标志

当T/C2 溢出时， TOV2 置位。执行相应的中断服务程序时此位硬件清零。此外， TOV2

也可以通过写1 来清零。当SREG 中的位I、TOIE2(T/C2 溢出中断使能) 和TOV2 都置位时，中断服务程序得到执行。在相位修正PWM 模式中，当T/C2 在0x00 改变记数方向时， TOV2 置位。

四.定时器寄存器

Atmega16有2个8位定时器和一个16位定时器。

1. 定时器 0

（1）T/C 控制寄存器TCCR0

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
FOC0	WGM00	COM01	COM00	WGM01	CS02	CS01	CS00

FOC0: 强制输出比较位

FOC0仅在WGM00指明非PWM模式时才有效。但是，为了保证与未来器件的兼容性，在使用PWM 时，写TCCR0 要对其清零。对其写1 后，波形发生器将立即进行比较操作。比较匹配输出引脚 OC0 将按照COM01:0 的设置输出相应的电平。要注意FOC0 类似一个锁存信号，真正对强制输出比较起作用的是COM01:0 的设置。

FOC0不会引发任何中断，也不会在利用OCR0作为TOP的CTC模式下对定时器进行清零的操作。

读FOC0 的返回值永远为0。

WGM01:0: 波形产生模式

这几位控制计数器的计数序列，计数器的最大值TOP，以及产生何种波形。T/C 支持的模式有：普通模式，比较匹配发生时清除计数器模式(CTC)，以及两种PWM 模式。

波形产生模式的位定义

模式	WGM01	WGM00	T/C 的工作模式	TOP	OCR0的更新时间	TOV0 的置位时刻
0	0	0	普通	0xFF	立即更新	MAX
1	0	1	相位修正PWM	0xFF	TOP	BOTTOM
2	1	0	CTC	OCR0	立即更新	MAX
3	1	1	快速PWM	0xFF	TOP	MAX

COM01:0: 比较匹配输出模式

这些位决定了比较匹配发生时输出引脚OC0 的电平。如果COM01:0 中的一位或全部都置位， OC0 以比较匹配输出的方式进行工作。同时其方向控制位要设置为1

以使能输出驱动器。

当OC0 连接到物理引脚上时， COM01:0 的功能依赖于WGM01:0 的设置。下表给出了当WGM01:0 设置为普通模式或CTC 模式时COM01:0 的功能。

比较输出模式，非PWM 模式

COM01	COM00	说明
0	0	正常的端口操作，不与OC0 相连接
0	1	比较匹配发生时OC0 取反
1	0	比较匹配发生时OC0 清零
1	1	比较匹配发生时OC0 置位

比较输出模式，快速PWM 模式

COM01	COM00	说明
0	0	正常的端口操作，不与OC0 相连接
0	1	保留
1	0	比较匹配发生时OC0A 清零，计数到TOP 时OC0 置位
1	1	比较匹配发生时OC0A 置位，计数到TOP 时OC0 清零

注意:一个特殊情况是OCR0 等于TOP，且COM01 置位。此时比较匹配将被忽略，而计数到TOP 时OC0 的动作继续有效。

比较输出模式，相位修正PWM 模式

COM01	COM00	说明
0	0	正常的端口操作，不与OC0 相连接
0	1	保留
1	0	在升序计数时发生比较匹配将清零OC0 ，降序计数时发生比较匹配将置位OC0
1	1	在升序计数时发生比较匹配将置位OC0 ，降序计数时发生比较匹配将清零OC0

注意: 一个特殊情况是OCR0 等于TOP，且COM01 置位。此时比较匹配将被忽略，而计数到TOP 时OC0 的动作继续有效。

CS02:0: 时钟选择

用于选择T/C 的时钟源。

CS02	CS01	CS00	说 明
0	0	0	无时钟， T/C 不工作

☆萧联珠☆ ☆azhu_uestc@163.com

0	0	1	clkI/O/1（没有预分频）
0	1	0	clkI/O/8（来自预分频器）
0	1	1	clkI/O/64（来自预分频器）
1	0	0	clkI/O/256（来自预分频器）
1	0	1	clkI/O/1024（来自预分频器）
1	1	0	时钟由T0 引脚输入，下降沿触发
1	1	1	时钟由T0 引脚输入，上升沿触发

如果T/C0 使用外部时钟，即使T0 被配置为输出，其上的电平变化仍然会驱动记数器。利用这一特性可通过软件控制记数。

（2）T/C 寄存器TCNT0

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
TCNT0[7:0]							

通过T/C 寄存器可以直接对计数器的8 位数据进行读写访问。对TCNT0 寄存器的写访问将在下一个时钟阻止比较匹配。在计数器运行的过程中修改TCNT0 的数值有可能丢失一次TCNT0 和OCR0 的比较匹配。

（3）输出比较寄存器OCR0

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
OCR0[7:0]							

输出比较寄存器包含一个8 位的数据，不间断地与计数器数值TCNT0 进行比较。匹配事件可以用来产生输出比较中断，或者用来在OC0 引脚上产生波形。

2. 定时器1

（1）T/C1 控制寄存器A（TCCR1A）

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
COM1A1	COM1A0	COM1B1	COM1B0	FOC1A	FOC1B	WGM11	WGM10

COM1A1:0: 通道A 的比较输出模式

COM1B1:0: 通道B 的比较输出模式

COM1A1:0与COM1B1:0分别控制OC1A与OC1B状态。如果COM1A1:0(COM1B1:0)的一位或两位被写入"1"，OC1A(OC1B) 输出功能将取代I/O 端口功能。此时OC1A(OC1B)相应的输出引脚数据方向控制必须置位以使能输出驱动器。

OC1A(OC1B) 与物理引脚相连时，COM1x1:0 的功能由WGM13:0 的设置决定

比较输出模式，非PWM

COM1A1/COM1B1	COM1A0/COM1B0	说明
0	0	普通端口操作，非OC1A/OC1B 功能
0	1	比较匹配时OC1A/OC1B 电平取反
1	0	比较匹配时清零OC1A/OC1B(输出低电平)
1	1	比较匹配时置位 OC1A/OC1B （ 输出高电平)

比较输出模式，快速 PWM

COM1A1/COM1B1	COM1A0/COM1B0	说明
0	0	普通端口操作，非OC1A/OC1B 功能
0	1	WGM13:0 = 15: 比较匹配时OC1A 取反 ， OC1B 不占用物理引脚。WGM13:0 为其它值时为普通端口操作，非OC1A/OC1B 功能
1	0	比较匹配时清零OC1A/OC1B， OC1A/OC1B在TOP 时置位
1	1	比较匹配时置位 OC1A/OC1B， OC1A/OC1B在TOP 时清零

注意: 当OCR1A/OCR1B 等于TOP且COM1A1/COM1B1置位时，比较匹配被忽略，OC1A/OC1B 的置位/ 清零操作有效。

比较输出模式，相位修正及相频修正 PWM 模式

COM1A1/COM1B1	COM1A0/COM1B0	说明
0	0	普通端口操作，非OC1A/OC1B 功能
0	1	WGM13:0 = 9 或14: 比较匹配时OC1A 取反，OC1B 不占用物理引脚。 WGM13:0为其它值时为普通端口操作，非OC1A/OC1B 功能
1	0	升序记数时比较匹配将清零OC1A/OC1B，降序记数时比较匹配将置位OC1A/OC1B
1	1	升序记数时比较匹配将置位OC1A/OC1B，降序记数时比较匹配将清零OC1A/OC1B

注意: OCR1A/OCR1B 等于TOP 且COM1A1/COM1B1 置位是一个特殊情况。

FOC1A: 通道A 强制输出比较

FOC1B: 通道B 强制输出比较

☆萧联珠☆ ☆azhu_uestc@163.com

FOC1A/FOC1B只有当WGM13:0指定为非PWM模式时被激活。为与未来器件兼容，工作在PWM 模式下对TCCR1A 写入时，这两位必须清零。当FOC1A/FOC1B 位置1 ，立即强制波形产生单元进行比较匹配。COM1x1:0 的设置改变 OC1A/OC1B 的输出。注意 FOC1A/FOC1B 位作为选通信号。COM1x1:0 位的值决定强制比较的效果。

在CTC 模式下使用OCR1A 作为TOP 值，FOC1A/FOC1B 选通即不会产生中断也不好清除定时器。

FOC1A/FOC1B 位总是读为0

WGM11:0: 波形发生模式

这两位与位于TCCR1B 寄存器的WGM13:2 相结合，用于控制计数器的计数序列——计数器计数的上限值和确定波形发生器的工作模式。T/C 支持的工作模式有：普通模式(计数器)，比较匹配时清零定时器(CTC) 模式，及三种脉宽调制(PWM) 模式。

波形产生模式的位描述

模式	WGM13	WGM12	WGM11	WGM10	定时器/ 计数器工作模式	计数上限值 TOP	OCR1x 更新时刻	TOV1 置位时 TOV1 置位时刻
0	0	0	0	0	普通模式	0xFFFF	立即更新	MAX
1	0	0	0	1	8 位相位修正 PWM	0x00FF	TOP	BOTTOM
2	0	0	1	0	9 位相位修正 PWM	0x01FF	TOP	BOTTOM
3	0	0	1	1	10位相位修正 PWM	0x03FF	TOP	BOTTOM
4	0	1	0	0	CTC	OCR1A	立即更新 MAX	4
5	0	1	0	1	8位快速PWM	0x00FF	TOP	TOP
6	0	1	1	0	9位快速PWM	0x01FF	TOP	TOP
7	0	1	1	1	10位快速PWM	0x03FF	TOP	TOP
8	1	0	0	0	相位与频率修	ICR1	BOTTOM	BOTTOM

					正PWM			
9	1	0	0	1	相位与频率修正PWM	OCR1A	BOTTOM	BOTTOM
10	1	0	1	0	相位修正PWM	ICR1	TOP	BOTTOM
11	1	0	1	1	相位修正PWM	OCR1A	TOP	BOTTOM
12	1	1	0	0	CTC	ICR1	立即更新MAX	
13	1	1	0	1	保留 -	-	-	
14	1	1	1	0	快速PWM	ICR1	TOP	TOP
15	1	1	1	1	快速 PWM	OCR1A	TOP	TOP

（2）T/C1 控制寄存器B（TCCR1B）

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
ICNC1	ICES1	-	WGM13	WGM12	CS12	CS11	CS10

ICNC1: 入捕捉噪声抑制器

置位ICNC1 将使能输入捕捉噪声抑制功能。此时外部引脚ICP1 的输入被滤波。其作用是从ICP1 引脚连续进行4 次采样。如果4 个采样值都相等，那么信号送入边沿检测器。因此使能该功能使得输入捕捉被延迟了4 个时钟周期。

ICES1: 输入捕捉触发沿选择

该位选择使用ICP1 上的哪个边沿触发捕获事件。ICES 为"0” 选择的是下降沿触发输入捕捉； ICES1 为"1” 选择的是逻辑电平的上升沿触发输入捕捉。

按照ICES1 的设置捕获到一个事件后，计数器的数值被复制到ICR1 寄存器。捕获事件还会置为ICF1。如果此时中断使能，输入捕捉事件即被触发。

当ICR1 用作TOP 值（见TCCR1A 与TCCR1B 寄存器中WGM13:0 位的描述）时，ICP1 与输入捕捉功能脱开，从而输入捕捉功能被禁用。

Bit 5 - 保留位

该位保留。为保证与将来器件的兼容性，写TCCR1B 时，该位必须写入"0”。

WGM13:2: 波形发生模式

见TCCR1A 寄存器中的描述。

CS12:0: 时钟选择

这3 位用于选择T/C 的时钟源，

时钟选择位描述

CS02	CS01	CS00	说 明

0	0	0	无时钟源
0	0	1	clkI/0/1
0	1	0	clkI/0/8
0	1	1	clkI/0/64
1	0	0	clkI/0/256
1	0	1	clkI/0/1024
1	1	0	外部T1
1	1	1	外部 T1

选择使用外部时钟源后，即使T1 引脚被定义为输出，其1 引脚上的逻辑信号电平变化仍然会驱动T/C1 计数，这个特性允许用户通过软件来控制计数。

（3）T/C1（TCNT1H 与TCNT1L）

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
TCNT1[15:8]							
TCNT1[7:0]							

TCNT1H与TCNT1L组成了T/C1的数据寄存器TCNT1。通过它们可以直接对定时器/计数器单元的16 位计数器进行读写访问。为保证CPU 对高字节与低字节的同时读写，必须使用一个8 位临时高字节寄存器TEMP。TEMP 是所有的16 位寄存器共用的

在计数器运行期间修改TCNT1的内容有可能丢失一次TCNT1与OCR1x的比较匹配操作。写TCNT1 寄存器将在下一个定时器周期阻塞比较匹配。

（4）输出比较寄存器1A（OCR1AH与OCR1AL）

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
OCR1A[15:8]							
OCR1A[7:0]							

（5）输出比较寄存器1B（OCR1BH与OCR1BL）

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
OCR1B[15:8]							
OCR1B[7:0]							

该寄存器中的16 位数据与TCNT1 寄存器中的计数值进行连续的比较，一旦数据匹配，将产生一个输出比较中断，或改变OC1x 的输出逻辑电平。

输出比较寄存器长度为16 位。为保证CPU 对高字节与低字节的同时读写，必须使用一个8 位临时高字节寄存器TEMP。TEMP 是所有的16 位寄存器共用的，

（6）输入捕捉寄存器1（ICR1H 与ICR1L）

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
ICR1[15:8]							
ICR1[7:0]							

当外部引脚ICP1(或T/C1的模拟比较器)有输入捕捉触发信号产生时，计数器TCNT1中的值写入ICR1 中。ICR1 的设定值可作为计数器的TOP 值。

输入捕捉寄存器长度为16 位。为保证CPU 对高字节与低字节的同时读写，必须使用一个8 位临时高字节寄存器TEMP。TEMP 是所有的16 位寄存器共用的。

3. 定时器 2

（1）T/C 控制寄存器TCCR2

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
FOC2	WGM20	COM21	COM20	WGM21	CS22	CS21	CS20

FOC2: 强制输出比较位

FOC2仅在WGM指明非PWM模式时才有效。但是，为了保证与未来器件的兼容性，在使用PWM 时，写TCCR2要对其清零。对其写1 后，波形发生器将立即进行比较操作。比较匹配输出引脚 OC2 将按照COM21:0 的设置输出相应的电平。要注意FOC2类似一个锁存信号，真正对强制输出比较起作用的是COM21:0 的设置。

FOC2不会引发任何中断，也不会在利用OCR2作为TOP的CTC模式下对定时器进行清零的操作。

读FOC2 的返回值永远为0。

WGM21:0: 波形产生模式

这几位控制计数器的计数序列，计数器的最大值TOP，以及产生何种波形。T/C 支持的模式有：普通模式，比较匹配发生时清除计数器模式(CTC)，以及两种PWM 模式。

波形产生模式的位定义

模式	WGM21	WGM20	T/C 的工作模式	TOP	OCR2的更新时间	TOV2 的置位时刻
0	0	0	普通	0xFF	立即更新	MAX
1	0	1	相位修正PWM	0xFF	TOP	BOTTOM
2	1	0	CTC	OCR0	立即更新	MAX
3	1	1	快速PWM	0xFF	TOP	MAX

COM21:0: 比较匹配输出模式

☆萧联珠☆ ☆azhu_uestc@163.com

这些位决定了比较匹配发生时输出引脚OC2 的电平。如果COM21:0 中的一位或全部都置位， OC2 以比较匹配输出的方式进行工作。同时其方向控制位要设置为1以使能输出驱动器。

当OC2 连接到物理引脚上时， COM21:0 的功能依赖于WGM21:0 的设置。下表给出了当WGM21:0 设置为普通模式或CTC 模式时COM21:0 的功能。

比较输出模式，非PWM 模式

COM21	COM20	说明
0	0	正常的端口操作，不与OC2相连接
0	1	比较匹配发生时OC2取反
1	0	比较匹配发生时OC2清零
1	1	比较匹配发生时OC2置位

比较输出模式，快速PWM 模式

COM21	COM20	说明
0	0	正常的端口操作，不与OC2相连接
0	1	保留
1	0	比较匹配发生时OC2清零，计数到TOP 时OC2置位
1	1	比较匹配发生时OC2置位，计数到TOP 时OC2清零

注意:一个特殊情况是OCR2 等于TOP，且COM21 置位。此时比较匹配将被忽略，而计数到TOP时OC2 的动作继续有效。

比较输出模式，相位修正PWM 模式

COM21	COM20	说明
0	0	正常的端口操作，不与OC2相连接
0	1	保留
1	0	在升序计数时发生比较匹配将清零OC2 ，降序计数时发生比较匹配将置位OC2
1	1	在升序计数时发生比较匹配将置位OC2 ，降序计数时发生比较匹配将清零OC2

注意: 一个特殊情况是OCR2 等于TOP，且COM21 置位。此时比较匹配将被忽略，而计数到TOP时OC2的动作继续有效。

CS02:0: 时钟选择

用于选择T/C 的时钟源

☆萧联珠☆ ☆azhu_uestc@163.com

CS22	CS21	CS20	说　　明
0	0	0	无时钟， T/C 不工作
0	0	1	clkI/O/1 （没有预分频）
0	1	0	clkI/O/8 （来自预分频器）
0	1	1	clkI/O/64 （来自预分频器）
1	0	0	clkI/O/256 （来自预分频器）
1	0	1	clkI/O/1024 （来自预分频器）
1	1	0	时钟由T0 引脚输入，下降沿触发
1	1	1	时钟由T0 引脚输入，上升沿触发

如果T/C2 使用外部时钟，即使T2被配置为输出，其上的电平变化仍然会驱动记数器。利用这一特性可通过软件控制记数。

（2）T/C 寄存器TCNT2

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
TCNT2[7:0]							

通过T/C 寄存器可以直接对计数器的8 位数据进行读写访问。对TCNT2 寄存器的写访问将在下一个时钟阻止比较匹配。在计数器运行的过程中修改TCNT2的数值有可能丢失一次TCNT2 和OCR2 的比较匹配。

（3）输出比较寄存器OCR2

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
OCR2[7:0]							

输出比较寄存器包含一个8 位的数据，不间断地与计数器数值TCNT2 进行比较。匹配事件可以用来产生输出比较中断，或者用来在OC2 引脚上产生波形。

（4）异步状态寄存器ASSR

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
-	-	-	-	AS2	TCN2UB	OCR2UB	TCR2UB

AS2: 异步 T/C2

AS2为"0”时T/C2由I/O时钟clkI/O驱动；AS2为"1”时T/C2由连接到TOSC1引脚的晶体振荡器驱动。改变AS2 有可能破坏TCNT2、OCR2 与TCCR2 的内容。

TCN2UB: T/C2 更新中

T/C2工作于异步模式时，写TCNT2将引起TCN2UB置位。当TCNT2从暂存寄存器更新完毕后TCN2UB 由硬件清零。TCN2UB 为0表明TCNT2 可以写入新值了。

OCR2UB: 输出比较寄存器2 更新中

T/C2工作于异步模式时，写OCR2将引起OCR2UB置位。当OCR2从暂存寄存器更新完毕后OCR2UB 由硬件清零。OCR2UB 为0 表明OCR2 可以写入新值了。

TCR2UB: T/C2 控制寄存器更新中

T/C2工作于异步模式时，写TCCR2将引起TCR2UB置位。当TCCR2从暂存寄存器更新完毕后TCR2UB 由硬件清零。TCR2UB 为0 表明TCCR2 可以写入新值了。

如果在更新忙标志置位的时候写上述任何一个寄存器都将引起数据的破坏，并引发不必要的中断。

读取TCNT2、OCR2 和TCCR2 的机制是不同的。读取TCNT2 得到的是实际的值，而OCR2 和TCCR2 则是从暂存寄存器中读取的。

定时器/ 计数器2 的异步操作T/C2 工作于异步模式时要考虑如下几点:

警告: 在同步和异步模式之间的转换有可能造成TCNT2、OCR2 和TCCR2 数据的损毁。

安全的步骤应该是:

- 清零OCIE2 和TOIE2 以关闭T/C2 的中断
- 设置AS2 以选择合适的时钟源
- 对TCNT2、OCR2 和TCCR2 写入新的数据
- 切换到异步模式: 等待TCN2UB、OCR2UB 和TCR2UB 清零
- 清除T/C2 的中断标志
- 需要的话使能中断

✧ 振荡器最好使用32.768 kHz手表晶振。给TOSC1 提供外部时钟，可能会造成T/C2 工作错误。系统主时钟必须比晶振高4 倍以上。

✧ 写TCNT2，OCR2和TCCR2时数据首先送入暂存器，两个TOSC1时钟正跳变后才锁存到对应到的寄存器。在数据从暂存器写入目的寄存器之前不能执行新的数据写入操作。3 个寄存器具有各自独立的暂存器，因此写TCNT2 并不会干扰OCR2 的写操作。异步状态寄存器ASSR 用来检查数据是否已经写入到目的寄存器。为0则MCU 就进入了休眠模式，那么比较匹配中断永远不会发生， MCU 也永远无法唤醒了。

✧ 如果要用T/C2作为省电模式或扩展Standby模式的唤醒条件，必须注意重新进入这些休眠模式的过程。中断逻辑需要一个TOSC1 周期进行复位。如果从唤醒到重新进入休眠的时间小于一个TOSC1 周期，中断将不再发生，器件也无法唤醒。如果用户怀疑自己程序是否满足这一条件，可以采取如下方法:

- 对TCCR2、TCNT2 或OCR2 写入合适的数据
- 等待ASSR 相应的更新忙标志清零
- 进入省电模式或扩展Standby 模式

✧ 若选择了异步工作模式，T/C2 的 32.768 kHz 振荡器将一直工作，除非进入掉电模式

或 Standby 模式。用户应该注意，此振荡器的稳定时间可能长达1 秒钟。因此，建议用户在器件上电复位，或从掉电/Standby 模式唤醒时至少等待1 秒钟后再使用T/C2。同时，由于启动过程时钟的不稳定性，唤醒时所有的T/C2 寄存器的内容都可能不正确，不论使用的是晶体还是外部时钟信号。用户必须重新给这些寄存器赋值。

✧ 使用异步时钟时省电模式或扩展Standby 模式的唤醒过程：中断条件满足后，在下一个定时器时钟唤醒过程启动。也就是说，在处理器可以读取计数器的数值之前计数器至少又累加了一个时钟。唤醒后MCU 停止4 个时钟，接着执行中断服务程序。中断服务程序结束之后开始执行SLEEP 语句之后的程序。

✧ 从省电模式唤醒之后的短时间内读取TCNT2 可能返回不正确的数据。因为TCNT2 是由异步的TOSC 时钟驱动的，而读取TCNT2 必须通过一个与内部I/O 时钟同步的寄存器来完成。同步发生于每个TOSC1 的上升沿。从省电模式唤醒后I/O 时钟重新激活，而读到的TCNT2 数值为进入休眠模式前的值，直到下一个TOSC1 上升沿的到来。从省电模式唤醒时TOSC1 的相位是完全不可预测的，而且与唤醒时间有关。因此，读取TCNT2 的推荐序列为：

- 写一个任意数值到OCR2 或TCCR2
- 等待相应的更新忙标志清零
- 读TCNT2

✧ 在异步模式下，中断标志的同步需要3 个处理器周期加一个定时器周期。在处理器可以读取引起中断标志置位的计数器数值之前计数器至少又累加了一个时钟。输出比较引脚的变化与定时器时钟同步，而不是处理器时钟。

（5）特殊功能IO寄存器SFIOR

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
ADTS2	ADTS1	ADTS0	-	ACME	PUD	PSR2	PSR10

PSR2: 预分频复位T/C2

当该位置1， T/C2 预分频器复位。操作完成后，该位被硬件清零。该位写0 无效。若内部CPU 时钟作为T/C2 时钟，该位读为0。当T/C2 工作在异步模式时，直到预分频器复位该位保持为1。

三. 串行外设接口 SPI

1. SPI 控制寄存器 SPCR

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
SPIE	SPE	DORD	MSTR	CPOL	CPHA	SPR1	SPR0

SPIE: 使能SPI 中断

置位后，只要SPSR 寄存器的SPIF 和SREG 寄存器的全局中断使能位置位，就会引发SPI 中断。

SPE: 使能SPI

SPE 置位将使能SPI。进行任何SPI 操作之前必须置位SPE。

DORD: 数据次序

DORD 置位时数据的LSB 首先发送；否则数据的MSB 首先发送。

MSTR: 主/ 从选择

MSTR置位时选择主机模式，否则为从机。如果MSTR为"1”，SS配置为输入，但被拉低，则MSTR 被清零，寄存器SPSR 的SPIF 置位。用户必须重新设置MSTR 进入主机模式。

CPOL: 时钟极性

CPOL 置位表示空闲时SCK 为高电平；否则空闲时SCK 为低电平。

CPOL 功能

CPOL	起始沿结束沿	CPOL
0	上升沿下降沿	0
1	下降沿上升沿	1

SPR1, SPR0: SPI 时钟速率选择1 与0

确定主机的SCK 速率。SPR1 和SPR0 对从机没有影响。SCK 和振荡器的时钟频率fosc关系如下表所示:

SCK 和振荡器频率的关系

SPI2X	SPR1	SPR0	SCK
0	0	0	fosc/4
0	0	1	fosc/16
0	1	0	fosc/64
0	1	1	fosc/128
1	0	0	fosc/2
1	0	1	fosc/8
1	1	0	fosc/32
1	1	1	fosc/64

2. SPI 状态寄存器 SPSR

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0

SPIF	WCOL	-	-	-	-	-	SPI2X

SPIF: SPI 中断标志

串行发送结束后，SPIF 置位。若此时寄存器SPCR 的SPIE 和全局中断使能位置位，SPI中断即产生。如果SPI 为主机， SS 配置为输入，且被拉低， SPIF 也将置位。进入中断服务程序后SPIF自动清零。或者可以通过先读SPSR，紧接着访问SPDR来对SPIF清零。

WCOL: 写碰撞标志

在发送当中对SPI 数据寄存器SPDR写数据将置位WCOL。WCOL可以通过先读SPSR，紧接着访问SPDR 来清零。

Bit 5..1 - Res: 保留

保留位，读操作返回值为零。

SPI2X: SPI 倍速

置位后SPI 的速度加倍。若为主机，则SCK 频率可达CPU 频率的一半。若为从机，只能保证fosc /4。

ATmega16的SPI接口同时还用来实现程序和EEPROM的下载和上载。

3. SPI 数据寄存器 SPDR

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
MSB							LSB

SPI 数据寄存器为读/写寄存器，用来在寄存器文件和SPI移位寄存器之间传输数据。写寄存器将启动数据传输，读寄存器将读取寄存器的接收缓冲器。

4.数据模式

相对于串行数据， SCK 的相位和极性有4 种组合。CPHA 和CPOL 控制组合的方式。SPI 数据传输格式。每一位数据的移出和移入发生于SCK不同的信号跳变沿，以保证有足够的时间使数据稳定。

CPOL 与CPHA 功能

	起始沿结束沿 SPI	模式	起始沿结束沿 SPI
CPOL = 0	CPHA = 0 采样 （ 上升沿） 采样 （ 下降沿） 0	CPOL = 0	CPHA = 0 采样 （ 上升沿） 采样 （ 下降沿） 0
CPOL = 0	CPHA = 1 设置 （ 上升沿） 采样 （ 下降沿） 1	CPOL = 0	CPHA = 1 设置 （ 上升沿） 采样 （ 下降沿） 1
CPOL = 1	CPHA = 0 采样 （ 下降沿） 采	CPOL = 1	CPHA = 0 采样 （ 下降沿） 采样

	样（上升沿）2		（上升沿）2
CPOL = 1	CPHA = 1 采样（下降沿）采样（上升沿）3	CPOL = 1	CPHA = 1 采样（下降沿）采样（上升沿）3

五. USART

通用同步和异步串行接收器和转发器(USART) 是一个高度灵活的串行通讯设备。主要特点为:

- 全双工操作(独立的串行接收和发送寄存器)
- 异步或同步操作
- 主机或从机提供时钟的同步操作
- 高精度的波特率发生器
- 支持5，6，7，8，或9 个数据位和1 个或2 个停止位
- 硬件支持的奇偶校验操作
- 数据过速检测
- 帧错误检测
- 噪声滤波，包括错误的起始位检测，以及数字低通滤波器
- 三个独立的中断：发送结束中断，发送数据寄存器空中断，以及接收结束中断
- 多处理器通讯模式
- 倍速异步通讯模式

1. USART I/O 数据寄存器 UDR

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
RXB[7:0]							
TXB[7:0]							

USART 发送数据缓冲寄存器和USART 接收数据缓冲寄存器共享相同的I/O 地址，称为USART 数据寄存器或UDR。将数据写入UDR 时实际操作的是发送数据缓冲器存器 (TXB)，读UDR 时实际返回的是接收数据缓冲寄存器(RXB) 的内容。

在5、6、7 比特字长模式下，未使用的高位被发送器忽略，而接收器则将它们设置为0。只有当UCSRA寄存器的UDRE标志置位后才可以对发送缓冲器进行写操作。如果UDRE没有置位，那么写入UDR 的数据会被USART 发送器忽略。当数据写入发送缓冲器后，若移位寄存器为空，发送器将把数据加载到发送移位寄存器。然后数据串行地从TxD 引脚输出。

接收缓冲器包括一个两级FIFO，一旦接收缓冲器被寻址FIFO 就会改变它的状态。因此不要对这一存储单元使用读- 修改- 写指令(SBI 和CBI)。使用位查询指令(SBIC 和SBIS)

时也要小心，因为这也有可能改变FIFO 的状态。

2. USART 控制和状态寄存器 A（UCSRA）

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
RXC	TXC	UDRE	FE	DOR	PE	U2X	MPCM

RXC: USART 接收结束

接收缓冲器中有未读出的数据时RXC 置位，否则清零。接收器禁止时，接收缓冲器被刷新，导致RXC 清零。RXC 标志可用来产生接收结束中断（见对RXCIE 位的描述）。

TXC: USART 发送结束

发送移位缓冲器中的数据被送出，且当发送缓冲器（UDR）为空时TXC 置位。执行发送结束中断时TXC 标志自动清零，也可以通过写1 进行清除操作。TXC 标志可用来产生发送结束中断（见对TXCIE 位的描述）。

UDRE: USART 数据寄存器空

UDRE标志指出发送缓冲器(UDR)是否准备好接收新数据。UDRE为1说明缓冲器为空，已准备好进行数据接收。UDRE标志可用来产生数据寄存器空中断(见对UDRIE位的描述)。复位后UDRE 置位，表明发送器已经就绪。

FE: 帧错误

如果接收缓冲器接收到的下一个字符有帧错误，即接收缓冲器中的下一个字符的第一个停止位为0，那么FE 置位。这一位一直有效直到接收缓冲器(UDR) 被读取。当接收到的停止位为1 时， FE 标志为0。对UCSRA 进行写入时，这一位要写0。

DOR: 数据溢出

数据溢出时DOR 置位。当接收缓冲器满（包含了两个数据），接收移位寄存器又有数据，若此时检测到一个新的起始位，数据溢出就产生了。这一位一直有效直到接收缓冲器(UDR) 被读取。对UCSRA 进行写入时，这一位要写0。

PE: 奇偶校验错误

当奇偶校验使能(UPM1 = 1)，且接收缓冲器中所接收到的下一个字符有奇偶校验错误时UPE 置位。这一位一直有效直到接收缓冲器（UDR）被读取。对UCSRA 进行写入时，这一位要写0。

U2X: 倍速发送

这一位仅对异步操作有影响。使用同步操作时将此位清零。此位置1可将波特率分频因子从16降到8，从而有效的将异步通信模式的传输速率加倍。

MPCM: 多处理器通信模式

设置此位将启动多处理器通信模式。MPCM置位后，USART 接收器接收到的那些不包含地址信息的输入帧都将被忽略。发送器不受MPCM设置的影响。

3. USART 控制和状态寄存器 B（UCSRB）

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
RXCIE	TXCIE	UDRIE	RXEN	TXEN	UCSZ2	RXB8	TXB8

RXCIE: 接收结束中断使能

置位后使能RXC中断。当RXCIE为1，全局中断标志位SREG置位，UCSRA寄存器的RXC亦为1时可以产生USART接收结束中断。

TXCIE: 发送结束中断使能

置位后使能TXC中断。当TXCIE为1，全局中断标志位SREG置位，UCSRA寄存器的TXC亦为1时可以产生USART发送结束中断。

UDRIE: USART 数据寄存器空中断使能

置位后使能UDRE中断。当UDRIE 为1，全局中断标志位SREG置位，UCSRA寄存器的UDRE亦为1时可以产生USART数据寄存器空中断。

RXEN: 接收使能

置位后将启动USART接收器。RxD 引脚的通用端口功能被USART功能所取代。禁止接收器将刷新接收缓冲器，并使 FE、DOR及PE标志无效。

TXEN: 发送使能

置位后将启动将启动USART发送器。TxD引脚的通用端口功能被USART功能所取代。TXEN 清零后，只有等到所有的数据发送完成后发送器才能够真正禁止，即发送移位寄存器与发送缓冲寄存器中没有要传送的数据。发送器禁止后，TxD引脚恢复其通用I/O功能。

UCSZ2: 字符长度

UCSZ2与UCSRC寄存器的UCSZ1:0结合在一起可以设置数据帧所包含的数据位数(字符长度)。

RXB8: 接收数据位8

对9位串行帧进行操作时，RXB8 是第9个数据位。读取UDR包含的低位数据之前首先要读取RXB8。

TXB8: 发送数据位8

对9位串行帧进行操作时，TXB8是第9个数据位。写UDR之前首先要对它进行写操作。

4. USART 控制和状态寄存器 C（UCSRC）

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
URSEL	UMSEL	UPM1	UPM0	USBS	UCSZ1	UCSZ0	UCPOL

UCSRC寄存器与UBRRH寄存器共用相同的I/O地址。对该寄存器的访问。

URSEL: 寄存器选择

通过该位选择访问UCSRC寄存器或UBRRH寄存器。当读UCSRC时，该位为1 ；当写UCSRC时，URSEL为1。

UMSEL: USART 模式选择

通过这一位来选择同步或异步工作模式。

UMSEL设置

UMSEL	模式
0	异步操作
1	同步操作

UPM1: 0: 奇偶校验模式

这两位设置奇偶校验的模式并使能奇偶校验。如果使能了奇偶校验，那么在发送数据，发送器都会自动产生并发送奇偶校验位。对每一个接收到的数据，接收器都会产生一奇偶值，并与UPM0 所设置的值进行比较。如果不匹配，那么就将UCSRA 中的PE 置位。

UPM 设置

UPM1	UPM0	奇偶模式
0	0	禁止
0	1	保留
1	0	偶校验
1	1	奇校验

USBS: 停止位选择

通过这一位可以设置停止位的位数。接收器忽略这一位的设置。

USBS 设置

USBS	停止位位数
0	1
1	2

UCSZ1: 0: 字符长度

UCSZ1: 0与UCSRB寄存器的 UCSZ2结合在一起可以设置数据帧包含的数据位数(字符长度)。

UCSZ 设置

UCSZ2	UCSZ1	UCSZ0	字符长度

0	0	0	5
0	0	1	6
0	1	0	7
0	1	1	8
1	0	0	保留
1	0	1	保留
1	1	0	保留
1	1	1	9

UCPOL: 时钟极性

这一位仅用于同步工作模式。使用异步模式时，将这一位清零。UCPOL 设置了输出数据的改变和输入数据采样，以及同步时钟XCK 之间的关系。

UCPOL 设置

UCPOL	发送数据的改变(TxD引脚的输出)	接收数据的采样(RxD 引脚的输入)
0	XCK上升沿XCK	下降沿
1	XCK 下降沿 XCK	上升沿

5. USART 波特率寄存器（UBRRL 和 UBRRH）

bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
URSEL	-	-	-	UBRR[11:8]			
UBRR[7:0]							

UCSRC寄存器与UBRRH寄存器共用相同的I/O地址。对该寄存器的访问。

URSEL: 寄存器选择

通过该位选择访问UCSRC 寄存器或UBRRH 寄存器。当读UBRRH时，该位为0；当写UBRRH时， URSEL为0。

Bit 14:12 - 保留位

这些位是为以后的使用而保留的。为了与以后的器件兼容，写UBRRH时将这些位清零。

UBRR11:0: USART 波特率寄存器

这个12位的寄存器包含了USART的波特率信息。其中UBRRH包含了USART波特率高4位，UBRRL包含了低8位。波特率的改变将造成正在进行的数据传输受到破坏。写UBRRL将立即更新波特率分频器。

六. 两线串行接口 TWI

- 简单，但是强大而灵活的通讯接口，只需要两根线
- 支持主机和从机操作
- 器件可以工作于发送器模式或接收器模式
- 7位地址空间允许有128个从机
- 支持多主机仲裁
- 高达400kHz 的数据传输率
- 斜率受控的输出驱动器
- 可以抑制总线尖峰的噪声抑制器
- 完全可编程的从机地址以及公共地址
- 睡眠时地址匹配可以唤醒AVR

两线接口TWI很适合于典型的处理器应用。TWI协议允许系统设计者只用两根双向传输线就可以将128个不同的设备互连到一起。这两根线一是时钟SCL，一是数据SDA。外部硬件只需要两个上拉电阻，每根线上一个。所有连接到总线上的设备都有自己的地址。TWI协议解决了总线仲裁的问题。

1. TWI 比特率寄存器 TWBR

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
TWBR7	TWBR6	TWBR5	TWBR4	TWBR3	TWBR2	TWBR1	TWBR0

TWBR为比特率发生器分频因子。比特率发生器是一个分频器，在主机模式下产生SCL时钟频率。

2. TWI 控制寄存器 TWCR

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	-	TWIE

TWCR用来控制TWI操作。它用来使能TWI，通过施加START到总线上来启动主机访问，产生接收器应答，产生STOP状态，以及在写入数据到TWDR 寄存器时控制总线的暂停等。这个寄存器还可以给出在TWDR无法访问期间，试图将数据写入到TWDR而引起的写入冲突信息。

TWINT: TWI 中断标志

当TWI完成当前工作，希望应用程序介入时TWINT置位。若SREG的I标志以及TWCR寄存器的TWIE标志也置位，则MCU执行TWI中断例程。当TWINT置位时， SCL信号的低电平被延长。TWINT标志的清零必须通过软件写"1”来完成。执行中断时硬件不会自动将其改写为"0”。要注意的是，只要这一位被清零，TWI立即开始工作。因此，在清零

TWINT之前一定要首先完成对地址寄存器TWAR，状态寄存器TWSR，以及数据寄存器TWDR 的访问。

TWEA: 使能TWI 应答

TWEA标志控制应答脉冲的产生。若TWEA置位，出现如下条件时接口发出ACK脉冲：

- ✧ 器件的从机地址与主机发出的地址相符合
- ✧ TWAR的TWGCE置位时接收到广播呼叫
- ✧ 在主机/从机接收模式下接收到一个字节的数据

将TWEA清零可以使器件暂时脱离总线。置位后器件重新恢复地址识别。

TWSTA: TWI START 状态标志

当CPU希望自己成为总线上的主机时需要置位TWSTA。TWI硬件检测总线是否可用。若总线空闲，接口就在总线上产生START状态。若总线忙，接口就一直等待，直到检测到一个STOP状态 ，然后产生START以声明自己希望成为主机。发送START之后软件必须清零TWSTA。

TWSTO: TWI STOP 状态标志

在主机模式下，如果置位TWSTO，TWI 接口将在总线上产生STOP状态，然后TWSTO自动清零。在从机模式下，置位TWSTO可以使接口从错误状态恢复到未被寻址的状态。此时总线上不会有STOP状态产生，但TWI返回一个定义好的未被寻址的从机模式且释放SCL与SDA为高阻态.

TWWC: TWI 写碰撞标志

当TWINT为低时写数据寄存器TWDR将置位TWWC。当TWINT为高时，每一次对TWDR的写访问都将更新此标志。

TWEN: TWI使能

TWEN位用于使能TWI操作与激活TWI接口。当TWEN位被写为"1”时，TWI引脚将I/O引脚切换到SCL与SDA引脚，使能波形斜率限制器与尖峰滤波器。如果该位清零， TWI接口模块将被关闭，所有TWI传输将被终止。

Res: 保留

保留，读返回值为”0”。

TWIE: 使能TWI中断

当SREG的I以及TWIE置位时，只要TWINT 为"1”， TWI中断就激活。

3. TWI 状态寄存器 TWSR

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
TWS7	TWS6	TWS5	TWS4	TWS3	-	TWPS1	TWPS0

TWS: TWI状态

这5位用来反映TWI逻辑和总线的状态。不同的状态代码将会在后面的部分描述。注意从TWSR读出的值包括5位状态值与2位预分频值。检测状态位时设计者应屏蔽预分频位为"0”。这使状态检测独立于预分频器设置。在无特殊声明的情况下，在手册中使用该方法。

Res: 保留

保留，读返回值为"0”。

TWPS: TWI预分频位

这两位可读/写，用于控制比特率预分频因子。

TWI 比特率预分频器

TWPS1	TWPS0	预分频器值
0	0	1
0	1	4
1	0	16
1	1	64

4. TWI 数据寄存器 TWDR

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
TWD7	TWD6	TWD5	TWD4	TWD3	TWD2	TWD1	TWD0

在发送模式， TWDR包含了要发送的字节；在接收模式， TWDR包含了接收到的数据。当TWI 接口没有进行移位工作(TWINT 置位) 时这个寄存器是可写的。在第一次中断发生之前用户不能够初始化数据寄存器。只要TWINT置位，TWDR的数据就是稳定的。在数据移出时，总线上的数据同时移入寄存器。TWDR总是包含了总线上出现的最后一个字节，除非MCU是从掉电或省电模式被TWI中断唤醒。此时TWDR的内容没有定义。总线仲裁失败时，主机将切换为从机，但总线上出现的数据不会丢失。ACK的处理由TWI逻辑自动管理， CPU不能直接访问ACK。

TWD: TWI 数据寄存器

根据状态的不同，其内容为要发送的下一个字节，或是接收到的数据。

5. TWI(从机) 地址寄存器 TWAR

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
TWA6	TWA5	TWA4	TWA3	TWA2	TWA1	TWA0	TWGCE

TWAR的高7位为从机地址。工作于从机模式时，TWI将根据这个地址进行响应。主机模式

不需要此地址。在多主机系统中，TWAR 需要进行设置以便其他主机访问自己。TWAR的LSB 用于识别广播地址 (0x00)。器件内有一个地址比较器。一旦接收到的地址和本机地址一致，芯片就请求中断。

TWA: TWI 从机地址寄存器

其值为从机地址。

TWGCE: 使能TWI 广播识别

置位后MCU 可以识别TWI 总线广播。

七. 模数转换器

1. ADC 多工选择寄存器 ADMUX

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
REFS1	REFS0	ADLAR	MUX4	MUX3	MUX2	MUX1	MUX0

REFS1: 0: 参考电压选择

通过这几位可以选择参考电压。如果在转换过程中改变了它们的设置，只有等到当前转换结束(ADCSRA寄存器的ADIF置位)之后改变才会起作用。如果在AREF引脚上施加了外部参考电压，内部参考电压就不能被选用了。

ADC 参考电压选择

REFS1	REFS0	参考电压选择
0	0	AREF,
0	1	AVCC,
1	0	保留
1	1	2.56V

ADLAR: ADC 转换结果左对齐

ADLAR影响ADC转换结果在ADC数据寄存器中的存放形式。ADLAR置位时转换结果为左对齐，否则为右对齐。ADLAR 的改变将立即影响ADC 数据寄存器的内容，不论是否有转换正在进行。

MUX4: 0: 模拟通道与增益选择位

通过这几位的设置，可以对连接到ADC 的模拟输入进行选择。也可对差分通道增益进行选择。如果在转换过程中改变这几位的值，那么只有到转换结束 (ADCSRA寄存器的ADIF 置位) 后新的设置才有效。

输入通道与增益选择

MUX4..0	单端输入正差	MUX4..0	单端输入正差	MUX4..0

☆萧联珠☆ ☆azhu_uestc@163.com

	分输入负差分输入增益		分输入负差分输入增益	
00000	ADC0			
00001	ADC1			
00010	ADC2			
00011	ADC3			
00100	ADC4			
00101	ADC5			
00110	ADC6			
00111	ADC7			
ADC0	N/A	ADC0	ADC0	10x
01001		ADC1	ADC0	10x
01010(1)		ADC0	ADC0	200x
01011(1)		ADC1	ADC0	200x
01100		ADC2	ADC2	10x
01101		ADC3	ADC2	10x
01110(1)		ADC2	ADC2	200x
01111(1)		ADC3	ADC2	200x
10000		ADC0	ADC1	1x
10001		ADC1	ADC1	1x
10010		ADC2	ADC2	ADC1
10011		ADC3	ADC1	1x
10100		ADC4	ADC1	1x
10101		ADC5	ADC1	1x
10110		ADC6	ADC1	1x
10111		ADC7	ADC1	1x
11000		ADC0	ADC2	1x
11001		ADC1	ADC2	1x
11010		ADC2	ADC2	1x

11011		ADC3	ADC2	1x
11100		ADC4	ADC2	1x
11101	ADC5	ADC2	ADC2	1x
11110	1.22V	N/A		
11111	0V			

2. ADC 控制和状态寄存器 A(ADCSRA)

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
ADEN	ADSC	ADATE	ADIF	ADIE	ADPS2	ADPS1	ADPS0

ADEN: ADC使能

ADEN置位即启动ADC，否则ADC功能关闭。在转换过程中关闭ADC将立即中止正在进行的转换。

ADSC: ADC开始转换

在单次转换模式下，ADSC置位将启动一次ADC转换。在连续转换模式下，ADSC置位将启动首次转换。第一次转换(在ADC启动之后置位ADSC，或者在使能ADC的同时置位ADSC)需要25个ADC 时钟周期，而不是正常情况下的13 个。第一次转换执行ADC初始化的工作。在转换进行过程中读取ADSC的返回值为"1”，直到转换结束。ADSC清零不产生任何动作。

ADATE: ADC自动触发使能

ADATE置位将启动ADC自动触发功能。触发信号的上跳沿启动ADC转换。触发信号源通过SFIOR寄存器的ADC触发信号源选择位ADTS设置。

ADIF: ADC中断标志

在ADC转换结束，且数据寄存器被更新后， ADIF置位。如果ADIE及SREG中的全局中断使能位I也置位，ADC 转换结束中断服务程序即得以执行，同时ADIF硬件清零。此外，还可以通过向此标志写1来清ADIF。要注意的是，如果对ADCSRA进行读－修改－写操作，那么待处理的中断会被禁止。这也适用于SBI及CBI指令。

ADIE: ADC 中断使能

若ADIE及SREG 的位I置位， ADC转换结束中断即被使能。

ADPS2: 0: ADC 预分频器选择位

由这几位来确定XTAL 与ADC 输入时钟之间的分频因子。

ADC 预分频选择

ADPS2	ADPS1	ADPS0	分频因子

0	0	0	2
0	0	1	2
0	1	0	4
0	1	1	8
1	0	0	16
1	0	1	32
1	1	0	64
1	1	1	128

3. ADC 数据寄存器(ADCL 及 ADCH)

ADLAR = 0

bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
–	–	–	–	–	–	ADC9	ADC8
ADC7	ADC6	ADC5	ADC4	ADC3	ADC2	ADC1	ADC0

ADLAR = 1

bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
ADC9	ADC8	ADC7	ADC6	ADC5	ADC4	ADC3	ADC2
ADC1	ADC0	–	–	–	–	–	–

ADC转换结束后，转换结果存于这两个寄存器之中。如果采用差分通道，结果由2的补码形式表示。

读取ADCL之后，ADC数据寄存器一直要等到ADCH也被读出才可以进行数据更新。因此，如果转换结果为左对齐，且要求的精度不高于8比特，那么仅需读取ADCH就足够了。否则必须先读出ADCL再读ADCH。

ADMUX寄存器的ADLAR及MUXn 会影响转换结果在数据寄存器中的表示方式。如果ADLAR为1，那么结果为左对齐；反之(系统缺省设置)，结果为右对齐。

ADC9:0: ADC转换结果

4.特殊功能 IO 寄存器 SFIOR

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
ADTS2	ADTS1	ADTS0	–	ACME	PUD	PSR2	PSR10

ADTS2:0: ADC 自动触发源

若ADCSRA寄存器的ADATE置位，ADTS的值将确定触发ADC转换的触发源；否则，ADTS的设置没有意义。被选中的中断标志在其上升沿触发ADC转换。从一个中断标志清零的触发源切换到中断标志置位的触发源会使触发信号产生一个上升沿。如果此时ADCSRA寄存器的ADEN为1，ADC转换即被启动。切换到连续运行模式(ADTS[2:0]=0)时，即使ADC中断标志已经置位也不会产生触发事件。

ADC 自动触发源选择

ADTS2	ADTS1	ADTS0	触发源
0	0	0	连续转换模式
0	0	1	模拟比较器
0	1	0	外部中断请求0
0	1	1	定时器/
1	0	0	定时器/
1	0	1	定时器/
1	1	0	定时器/
1	1	1	定时器/

Res: 保留位

这一位保留。为了与以后的器件相兼容，在写SFIOR时这一位应写0。

八. 模拟比较器

1. 特殊功能 IO 寄存器 SFIOR

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
ADTS2	ADTS1	ADTS0	-	ACME	PUD	PSR2	PSR10

ACME: 模拟比较器多路复用器使能

当此位为逻辑"1”，且ADC处于关闭状态(ADCSRA 寄存器的ADEN为"0”) 时，ADC多路复用器为模拟比较器选择负极输入。当此位为"0”时， AIN1连接到比较器的负极输入端。更详细描述的请参见 P191 “ 模拟比较器多工输入” 。

2. 模拟比较器控制和状态寄存器 ACSR

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
ACD	ACBG	ACO	ACI	ACIE	ACIC	ACIS1	ACIS0

ACD: 模拟比较器禁用

ACD置位时，模拟比较器的电源被切断。可以在任何时候设置此位来关掉模拟比较器。这可以减少器件工作模式及空闲模式下的功耗。改变ACD位时，必须清零ACSR寄存器的ACIE位来禁止模拟比较器中断。否则ACD改变时可能会产生中断。

ACBG: 选择模拟比较器的能隙基准源

ACBG置位后，模拟比较器的正极输入由能隙基准源所取代。否则，AIN0连接到模拟比较器的正极输入。

ACO: 模拟比较器输出

模拟比较器的输出经过同步后直接连到ACO。同步机制引入了1-2 个时钟周期的延时。

ACI: 模拟比较器中断标志

当比较器的输出事件触发了由ACIS1及ACIS0定义的中断模式时，ACI置位。如果ACIE和SREG寄存器的全局中断标志I也置位，那么模拟比较器中断服务程序即得以执行，同时ACI被硬件清零。ACI也可以通过写"1”来清零。

ACIE: 模拟比较器中断使能

当ACIE位被置"1”且状态寄存器中的全局中断标志I也被置位时，模拟比较器中断被激活。否则中断被禁止。

ACIC: 模拟比较器输入捕捉使能

ACIC置位后允许通过模拟比较器来触发T/C1的输入捕捉功能。此时比较器的输出被直接连接到输入捕捉的前端逻辑，从而使得比较器可以利用T/C1输入捕捉中断逻辑的噪声抑制器及触发沿选择功能。ACIC为"0”时模拟比较器及输入捕捉功能之间没有任何联系。为了使比较器可以触发T/C1 的输入捕捉中断，定时器中断屏蔽寄存器TIMSK的TICIE1必须置位。

ACIS1, ACIS0: 模拟比较器中断模式选择

这两位确定触发模拟比较器中断的事件。Table79给出了不同的设置。

ACIS1/ACIS0 设置

ACIS1	ACIS0	中断模式
0	0	比较器输出变化即可触发中断
0	1	保留
1	0	比较器输出的下降沿产生中断
1	1	比较器输出的上升沿产生中断

需要改变ACIS1/ACIS0时，必须清零ACSR寄存器的中断使能位来禁止模拟比较器中断。否则有可能在改变这两位时产生中断。

3. 模拟比较器多工输入

可以选择ADC7..0之中的任意一个来代替模拟比较器的负极输入端。ADC复用器可用来完成这个功能。当然，为了使用这个功能首先必须关掉ADC。如果模拟比较器复用器使能位(SFIOR 中的ACME)被置位，且ADC也已经关掉(ADCSRA 寄存器的ADEN为0)，则可以通过ADMUX寄存器的MUX2..0来选择替代模拟比较器负极输入的管脚。如果ACME清零或ADEN置位，则模拟比较器的负极输入为AIN1。

模拟比较器复用输入

ACME	ADEN	MUX2..0	模拟比较器负极输入
0	x	xxx	AIN1
1	1	xxx	AIN1
1	0	000	ADC0
1	0	001	ADC1
1	0	010	ADC2
1	0	011	ADC3
1	0	100	ADC4
1	0	101	ADC5
1	0	110	ADC6
1	0	111	ADC7

九. EEPROM 数据存储器

1. EEPROM 地址寄存器(EEARH 和 EEARL)

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
EEAR7	EEAR6	EEAR5	EEAR4	EEAR3	EEAR2	EEAR1	EEAR0

Res: 保留

保留位，读操作返回值为零。

EEAR8..0: EEPROM 地址

EEPROM地址寄存器 – EEARH和EEARL指定了512字节的EEPROM空间。EEPROM地址是线性的，从0 到511。EEAR的初始值没有定义。在访问EEPROM之前必须为其赋予正确的数据。

2. EEPROM 数据寄存器 EEDR

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0

MSB							LSB

EEDR7.0: EEPROM 数据

对于EEPROM写操作， EEDR是需要写到EEAR单元的数据；对于读操作，EEDR 是从地址EEAR读取的数据。

3. EEPROM 控制寄存器 EECR

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
–	–	–	–	EERIE	EEMWE	EEWE	EERE

Res: 保留

保留位，读操作返回值为零。

EERIE: 使能EEPROM 准备好中断

若SREG 的I 为"1"，则置位EERIE 将使能EEPROM 准备好中断。清零EERIE 则禁止此中断。当EEWE 清零时EEPROM 准备好中断即可发生。

EEMWE: EEPROM 主机写使能

EEMWE决定了EEWE置位是否可以启动EEPROM写操作。当EEMWE为"1"时，在4个时钟周期内置位EEWE将把数据写入EEPROM的指定地址；若EEMWE为"0 “，则操作EEWE不起作用。EEMWE置位后4个周期，硬件对其清零。见EEPROM 写过程中对EEWE 位的描述。

EEWE: EEPROM写使能

EEWE为EEPROM写操作的使能信号。当EEPROM数据和地址设置好之后，需置位EEWE以便将数据写入EEPROM。此时EEMWE必须置位，否则EEPROM写操作将不会发生。写时序如下(第3步和第4步的次序并不重要)：

- ✧ 等待EEWE位变为零
- ✧ 等待SPMCSR中的SPMEN位变为零
- ✧ 将新的EEPROM地址写入EEAR(可选)
- ✧ 将新的EEPROM数据写入EEDR(可选)
- ✧ 对EECR寄存器的EEMWE写"1"，同时清零EEWE
- ✧ 在置位EEMWE的4个周期内，置位EEWE

在CPU写Flash存储器的时候不能对EEPROM进行编程。在启动EEPROM写操作之前软件必须检查Flash写操作是否已经完成。仅在软件包含引导程序并允许CPU对Flash进行编程时才有用。

注意：如果一个操作EEPROM的中断打断了另一个EEPROM操作，EEAR或EEDR寄存器可能被修改，引起EEPROM 操作失败。建议此时关闭全局中断标志I。经过写访问时间之后，EEWE 硬件清零。用户可以凭借这一位判断写时序是否已经完成。EEWE 置位后，CPU 要停止两个时钟周期才会运行下一条指令。

EERE: EEPROM 读使能

EERE为EEPROM读操作的使能信号。当EEPROM地址设置好之后，需置位EERE以便将数据读入EEAR。EEPROM 数据的读取只需要一条指令，且无需等待。读取EEPROM后CPU要停止4个时钟周期才可以执行下一条指令。

用户在读取EEPROM时应该检测EEWE。如果一个写操作正在进行，就无法读取EEPROM，也无法改变寄存器EEAR。

☆萧联珠☆ ☆azhu_uestc@163.com