Ayudantía Timers

Javier Díaz - Pablo Orellana

IEE2463 - Sistemas Electrónicos Programables

Mayo 2020

ATmega328P - Timer 0

- TCCR0A – Timer/Counter Control Register A:

Bit	7	6	5	4	3	2	1	0	
0x24 (0x44)	COM0A1	COM0A0	COM0B1	COM0B0	–	–	WGM01	WGM00	TCCR0A
Read/Write	R/W	R/W	R/W	R/W	R	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- TCCR0B – Timer/Counter Control Register B:

Bit	7	6	5	4	3	2	1	0	
0x25 (0x45)	FOC0A	FOC0B	–	–	WGM02	CS02	CS01	CS00	TCCR0B
Read/Write	W	W	R	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

ATmega328P - Timer 0

- Modos de Operación:

Mode	WGM02	WGM01	WGM00	Timer/Counter Mode of Operation	TOP	Update of OCRx at	TOV Flag Set on(1)(2)
0	0	0	0	Normal	0xFF	Immediate	MAX
1	0	0	1	PWM, phase correct	0xFF	TOP	BOTTOM
2	0	1	0	CTC	OCRA	Immediate	MAX
3	0	1	1	Fast PWM	0xFF	BOTTOM	MAX
4	1	0	0	Reserved	–	–	–
5	1	0	1	PWM, phase correct	OCRA	TOP	BOTTOM
6	1	1	0	Reserved	–	–	–
7	1	1	1	Fast PWM	OCRA	BOTTOM	TOP

Notes: 1. MAX = 0xFF
2. BOTTOM = 0x00

ATmega328P - Timer 0

- Prescaler:

CS02	CS01	CS00	Description
0	0	0	No clock source (Timer/Counter stopped)
0	0	1	$clk_{I/O}$/(no prescaling)
0	1	0	$clk_{I/O}$/8 (from prescaler)
0	1	1	$clk_{I/O}$/64 (from prescaler)
1	0	0	$clk_{I/O}$/256 (from prescaler)
1	0	1	$clk_{I/O}$/1024 (from prescaler)
1	1	0	External clock source on T0 pin. Clock on falling edge.
1	1	1	External clock source on T0 pin. Clock on rising edge.

ATmega328P - Timer 0

- TCNT0 – Timer/Counter Register
- OCR0A – Output Compare Register A
- OCR0B – Output Compare Register B
- TIMSK0 – Timer/Counter Interrupt Mask Register:

Bit	7	6	5	4	3	2	1	0	
(0x6E)	–	–	–	–	–	OCIE0B	OCIE0A	TOIE0	TIMSK0
Read/Write	R	R	R	R	R	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

MSP430F5529 - Timer A

- TAxCTL Register:

15	14	13	12	11	10	9	8
Reserved						TASSEL	
rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)
7	6	5	4	3	2	1	0
ID		MC		Reserved	TACLR	TAIE	TAIFG
rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	w-(0)	rw-(0)	rw-(0)

9-8	TASSEL	RW	0h	Timer_A clock source select 00b = TAxCLK 01b = ACLK 10b = SMCLK 11b = INCLK

5-4	MC	RW	0h	Mode control. Setting MC = 00h when Timer_A is not in use conserves power. 00b = Stop mode: Timer is halted 01b = Up mode: Timer counts up to TAxCCR0 10b = Continuous mode: Timer counts up to 0FFFFh 11b = Up/down mode: Timer counts up to TAxCCR0 then down to 0000h

MSP430F5529 - Timer A

- TAxCCTLn Register:

15	14	13	12	11	10	9	8
CM		CCIS		SCS	SCCI	Reserved	CAP
rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	r-(0)	r-(0)	rw-(0)
7	6	5	4	3	2	1	0
OUTMOD			CCIE	CCI	OUT	COV	CCIFG
rw-(0)	rw-(0)	rw-(0)	rw-(0)	r	rw-(0)	rw-(0)	rw-(0)

7-5	OUTMOD	RW	0h	Output mode. Modes 2, 3, 6, and 7 are not useful for TAxCCR0 because EQUx = EQU0. 000b = OUT bit value 001b = Set 010b = Toggle/reset 011b = Set/reset 100b = Toggle 101b = Reset 110b = Toggle/set 111b = Reset/set
4	CCIE	RW	0h	Capture/compare interrupt enable. This bit enables the interrupt request of the corresponding CCIFG flag. 0b = Interrupt disabled 1b = Interrupt enabled

- TAxCCRn: Contador Timer A
- TAxCCRn: Valor de comparación

Ejemplo Cronómetro - ATmega328P

```
// Enteros que permiten llevar cuenta del cronometro

volatile int unidad = 0;
volatile int decena = 0;
volatile int centena = 0;
volatile int display = 0;

// Determinar Velocidad de Cuenta Cronómetro

void Timer_1_Configuration(uint8_t freq){
    TCCR1B |= (1 << CS12) | (1 << WGM12); // Modo CTC, prescaler 256
    TIMSK1 |= (1 << OCIE1A); // Interrupcion de comparacion con registro OCR1A
    OCR1A |= (F_CPU/(freq*256)-1); //Definir frecuencia para velocidad del contador
}

// Determinar Frecuencia de Display

void Timer_0_Configuration(uint8_t freq){
    TCCR0A |= (1 << WGM01); // Modo CTC
    TCCR0B |= (1 << CS02); // prescaler 256
    TIMSK0 |= (1 << OCIE0A); // Interrupcion de comparacion con registro OCR1A
    OCR0A |= (F_CPU/(freq*256)-1); //Definir frecuencia para velocidad de pantalla display
}
```

Ejemplo Cronómetro - ATmega328P

```
int main(void)
{

    // Puerto D - Leds de cada Display

    DDRD |= 0xFF;
    PORTD |= 0xFF;

    // Puerto B - Compuertas de cada Display

    DDRB |= 0xFF;
    PORTB |= COM_2;
    PORTB |= COM_3;
    PORTB |= COM_1;

    sei(); //activar interrupciones

    Timer_0_Configuration(500); // Display 500 Hz
    Timer_1_Configuration(10); // Cronómetro 10 Hz

    while (1){
    }
}
```

Ejemplo Cronómetro - ATmega328P

```
//Rutina de interrupcion para aumentar cuenta en cronómetro

ISR(TIMER1_COMPA_vect){

    unidad ++;
    if(unidad==10){
        unidad = 0;
        decena ++;
    }
    if (decena==10){
        unidad = 0;
        centena ++;
    }
    if (centena == 10){
        centena = 0;
    }

}
```

Ejemplo Cronómetro - ATmega328P

```
//Rutina de interrupcion para mostrar numero en pantalla

ISR(TIMER0_COMPA_vect){
    if (display == 0){
        PORTB = (COM_3);
        PORTD = ~(numeros[unidad]) | (DP);
    }

    else if (display == 1){
        PORTB = COM_2;
        PORTD = ~(numeros[decena]);
        PORTD &= ~(DP);
    }

    else if (display == 2){
        PORTB = COM_1;
        PORTD = ~(numeros[centena]) | (DP);
    }
    display++;
    if(display==3){
        display = 0;
    }
}
```

Ejemplo Cronómetro - MSP430F5529

```
void Pin_Configuration(void)
{
    P2DIR = ALL;
    P1DIR = b + d + e + f;
    P2OUT = ALL;
}

void Timer_A0_Configuration (void)
{
    TA0CCTL0 = CCIE;                          // Interrupción de comparación activada.
    TA0CCR0 = 200;                            // Valor al que debe llegar el contador para activar la interrupción.
    TA0CTL = TASSEL_1 + MC_1 + TACLR;

    /**
     * TASSEL_1 -> SMCLK - 2^20 Hz
     * MC_1 -> Up mode: Timer cuenta hasta TA0CCR0
     * TACLR -> Cuando este bit se declara en 1, el contador se resetea a 0. Este bit vuelve automáticamente a 0 después de esto.
    */
}

void Timer_A1_Configuration (void)
{
    TA1CCTL0 = CCIE;                          // Interrupción de comparación activada.
    TA1CCR0 = 1049;                           // Valor al que debe llegar el contador para activar la interrupción.
    TA1CTL = TASSEL_1 + MC_1 + TACLR;

    /**
     * TASSEL_1 -> SMCLK - 2^20 Hz
     * MC_1 -> Up mode: Timer cuenta hasta TA1CCR0
     * TACLR -> Cuando este bit se declara en 1, el contador se resetea a 0. Este bit vuelve automáticamente a 0 después de esto.
    */
}
```

Ejemplo Cronómetro - MSP430F5529

```
#pragma vector=TIMER0_A0_VECTOR
__interrupt void TIMER0_A0_ISR(void){
    switch(contador_2){
        case 2:
            centena = (contador/100)%10;
            P1OUT = e;
            P1OUT |= f;
            P2OUT |= ALL;
            P2OUT &= ~lista[centena];
            break;
        case 1:
            decena = (contador/10)%10;
            P1OUT = d + f ;
            P1OUT &= ~(f);
            P2OUT |= ALL;
            P2OUT &= ~lista[decena];
            break;
        case 0:
            unidad = contador%10;
            P1OUT = b;
            P1OUT |= f;
            P2OUT |= ALL;
            P2OUT &= ~lista[unidad];
            break;
    }
    contador_2++;
    if(contador_2 == 3){
        contador_2 = 0;
    }
}
```

Ejemplo Cronómetro - MSP430F5529

```
/**
 * Esta interrupción se encarga de llevar la cuenta de 0
 * a 999 con incrementos cada 1 ms aprox.
*/
#pragma vector=TIMER1_A0_VECTOR
__interrupt void TIMER1_A0_ISR(void){
    contador++;
    if (contador == 999){
        contador = 0;
    }
}
```

Ejemplo Cronómetro - MSP430F5529

```
unsigned int lista[10] = {ZERO, ONE, TWO, THREE, FOUR, FIVE, SIX, SEVEN, EIGHT, NINE};
unsigned int contador = 0;                    // Contador de 0 a 999
int          contador_2 = 0;                  // Contador de 0 a 2 que selecciona que digito vamos a prender
volatile int unidad;
volatile int decena;
volatile int centena;

void main(void)
{
    WDTCTL = WDTPW | WDTHOLD;                 // Parar el watchdog timer

    Timer_A0_Configuration();                 // Configuración funcionamiento Timer 0
    Timer_A1_Configuration();                 // Configuración funcionamiento Timer 1
    Pin_Configuration();                      // Configuración de los pines a utilizar

    __bis_SR_register(GIE);                   // Activa las interrupciones generales

    while(1){}
}
```