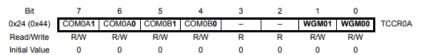
## Ayudantía Timers

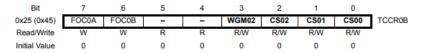
Javier Díaz - Pablo Orellana IEE2463 - Sistemas Electrónicos Programables

Mayo 2020

• TCCR0A - Timer/Counter Control Register A:



• TCCR0B – Timer/Counter Control Register B:



### Modos de Operación:

Mode	WGM02	WGM01	WGM00	Timer/Counter Mode of Operation	ТОР	Update of OCRx at	TOV Flag Set on <sup>(1)(2)</sup>
0	0	0	0	Normal	0xFF	Immediate	MAX
1	0	0	1	PWM, phase correct	0xFF	TOP	BOTTOM
2	0	1	0	СТС	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	воттом	TOP

Notes: 1. MAX = 0xFF

2. BOTTOM = 0x00

#### • Prescaler:

CS02	CS01	CS00	Description
0	0	0	No clock source (Timer/Counter stopped)
0	0	1	clk <sub>I/O</sub> /(no prescaling)
0	1	0	clk <sub>I/O</sub> /8 (from prescaler)
0	1	1	clk <sub>I/O</sub> /64 (from prescaler)
1	0	0	clk <sub>I/O</sub> /256 (from prescaler)
1	0	1	clk <sub>I/O</sub> /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.

- 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
		Rese	erved		TAS	SEL	
rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)
7	6	5	4	3	2	1	0
ll ll	D	M	C	Reserved	TACLR	TAIE	TAIFG
rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	w-(0)	rw-(0)	rw-(0)

9-8	TASSEL	RW	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	CC	CIS	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	Oh	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

7 / 15

```
volatile int unidad = 0:
volatile int decena = 0;
volatile int centena = 0;
volatile int display = 0;
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
void Timer 0 Configuration(uint8 t freq){
    TCCR0A |= (1 << WGM01); // Modo CTC
    TCCR0B |= (1 << CS02); // prescaler 256
   TIMSK0 |= (1 << OCIEOA): // Interrupcion de comparacion con registro OCR1A
   OCROA |= (F CPU/(freg*256)-1): //Definir frecuencia para velocidad de pantalla display
```

```
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){
```

```
//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;
```

```
/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:
```

```
void Pin Configuration(void)
    P2DIR = ALL:
   P1DIR = b + d + e + f;
    P2OUT = ALL:
void Timer A0 Configuration (void)
    TA0CCTL0 = CCIE:
   TA0CCR0 = 200;
   TAOCTL = TASSEL 1 + MC 1 + TACLR;
void Timer A1 Configuration (void)
    TA1CCTL0 = CCIE:
   TA1CCR0 = 1049;
   TA1CTL = TASSEL 1 + MC 1 + TACLR:
```

```
#pragma vector=TIMER0 A0 VECTOR
 interrupt void TIMER0 A0 ISR(void){
    switch(contador 2){
        case 2:
            centena = (contador/100)%10;
           P10UT = e:
           P10UT |= f;
           P20UT |= ALL;
           P2OUT &= ~lista[centena];
           break:
        case 1:
            decena = (contador/10)%10;
           P10UT = d + f;
           P10UT &= ~(f):
           P2OUT |= ALL;
           P2OUT &= ~lista[decena];
           break:
        case 0:
           unidad = contador%10;
           P10UT = b;
           P10UT |= f;
           P20UT |= ALL:
           P2OUT &= ~lista[unidad];
   contador_2++;
   if(contador 2 == 3){
        contador 2 = 0;
```

```
unsigned int lista[10] = {ZERO, ONE, TWO, THREE, FOUR, FIVE, SIX, SEVEN, EIGHT, NINE};
unsigned int contador = 0;
           contador_2 = 0;
volatile int unidad:
volatile int decena:
volatile int centena;
void main(void)
   WDTCTL = WDTPW | WDTHOLD;
   Timer A0 Configuration():
   Timer A1 Configuration();
   Pin Configuration():
   bis SR register(GIE);
```