

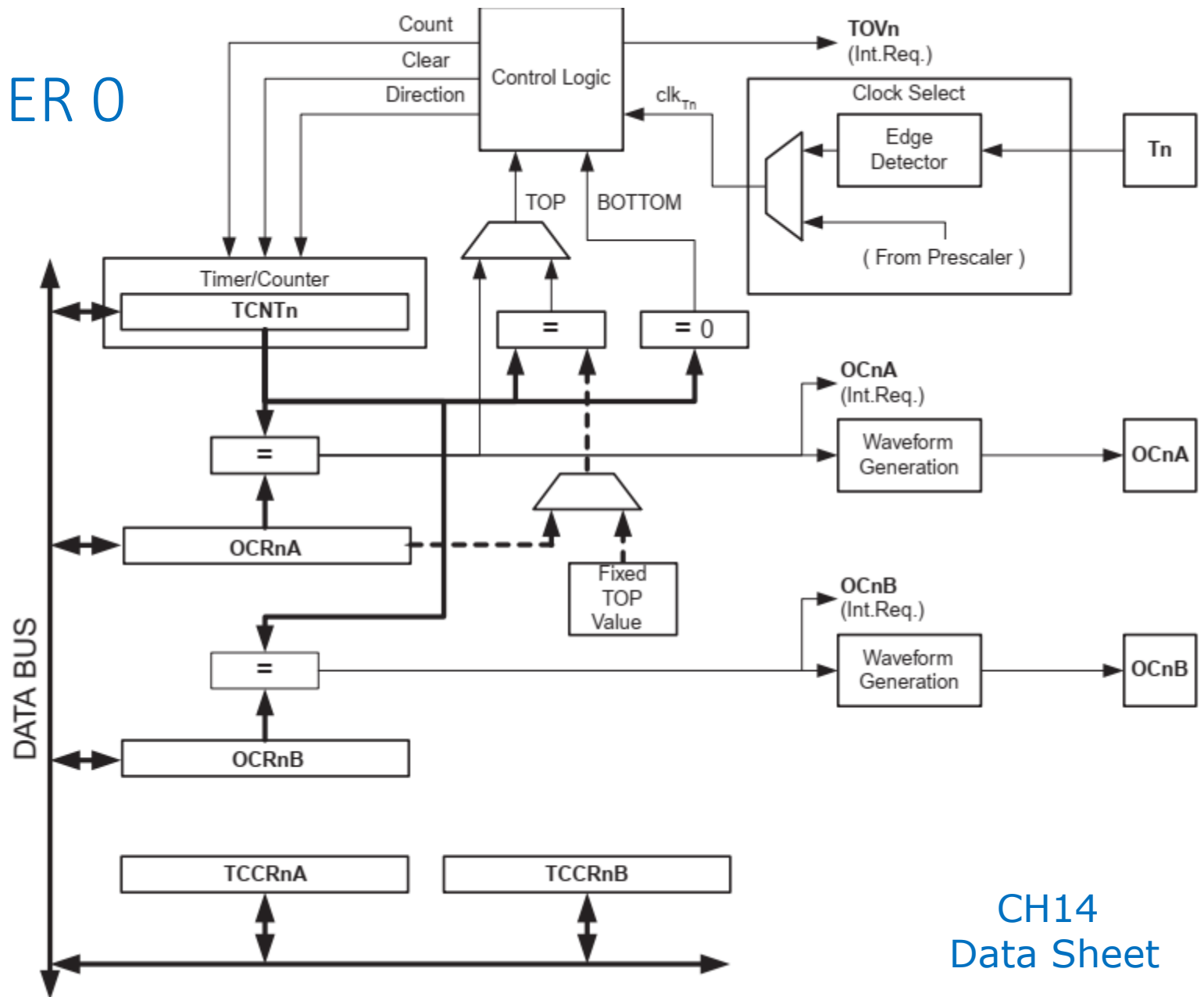
CIRCUITOS DIGITALES Y MICROCONTROLADORES 2022

Facultad de Ingeniería
UNLP

Periférico TIMER1

Ing. José Juárez

TIMER 0



TIMER 0

- Registros para su programación:

Registros de configuración

7	6	5	4	3	2	1	0	
COM0A1	COM0A0	COM0B1	COM0B0	–	–	WGM01	WGM00	TCCR0A
R/W	R/W	R/W	R/W	R	R	R/W	R/W	
7	6	5	4	3	2	1	0	
FOC0A	FOC0B	–	–	WGM02	CS02	CS01	CS00	TCCR0B
W	W	R	R	R/W	R/W	R/W	R/W	

Contador

7	6	5	4	3	2	1	0	
TCNT0[7:0]								TCNT0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Registros de comparación

7	6	5	4	3	2	1	0	
OCR0A[7:0]								OCR0A
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
7	6	5	4	3	2	1	0	
OCR0B[7:0]								OCR0B
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Conf. de interrupciones

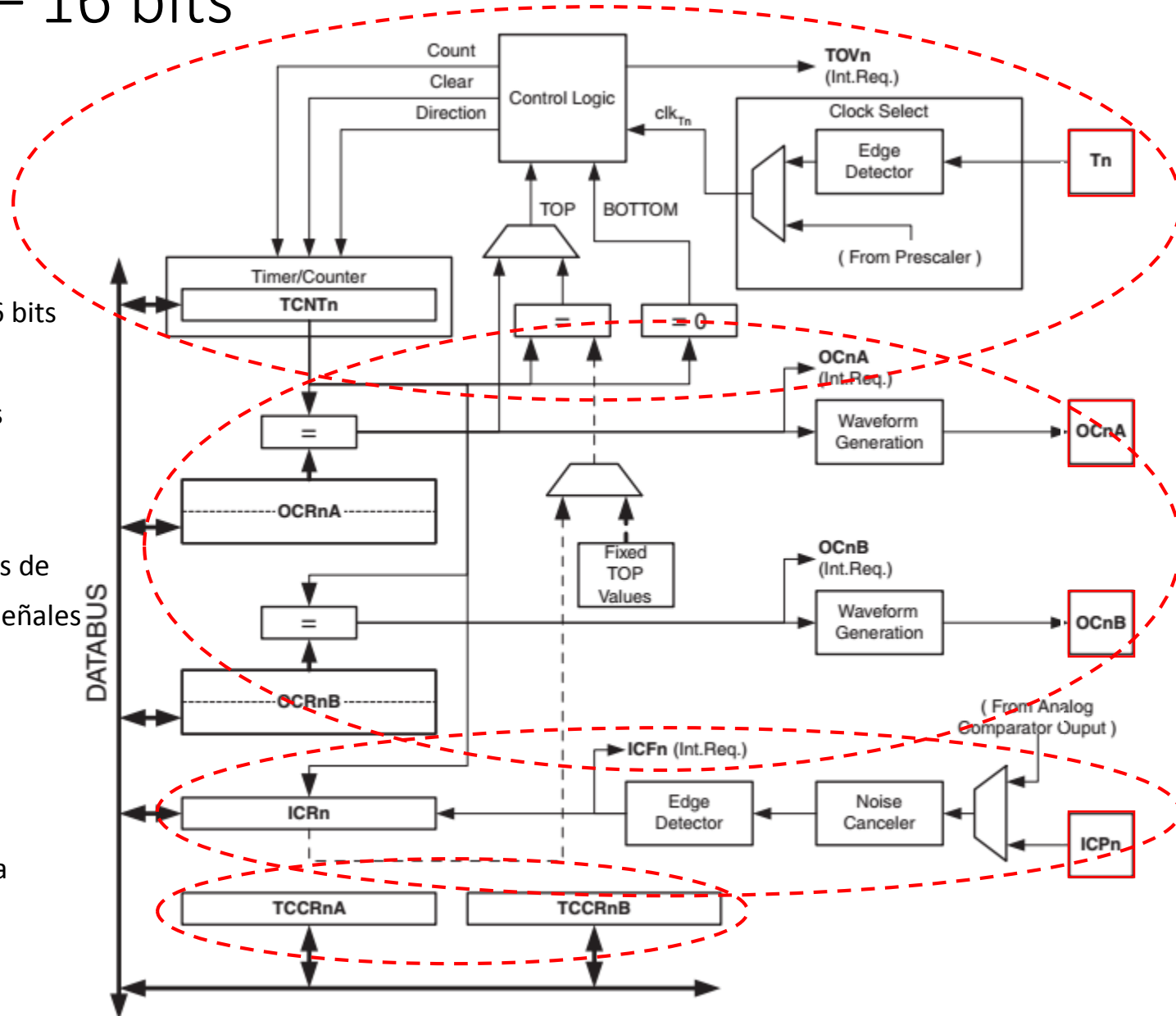
7	6	5	4	3	2	1	0	
–	–	–	–	–	OCIE0B	OCIE0A	TOIE0	TIMSK0
R	R	R	R	R	R/W	R/W	R/W	

Banderas de notificación

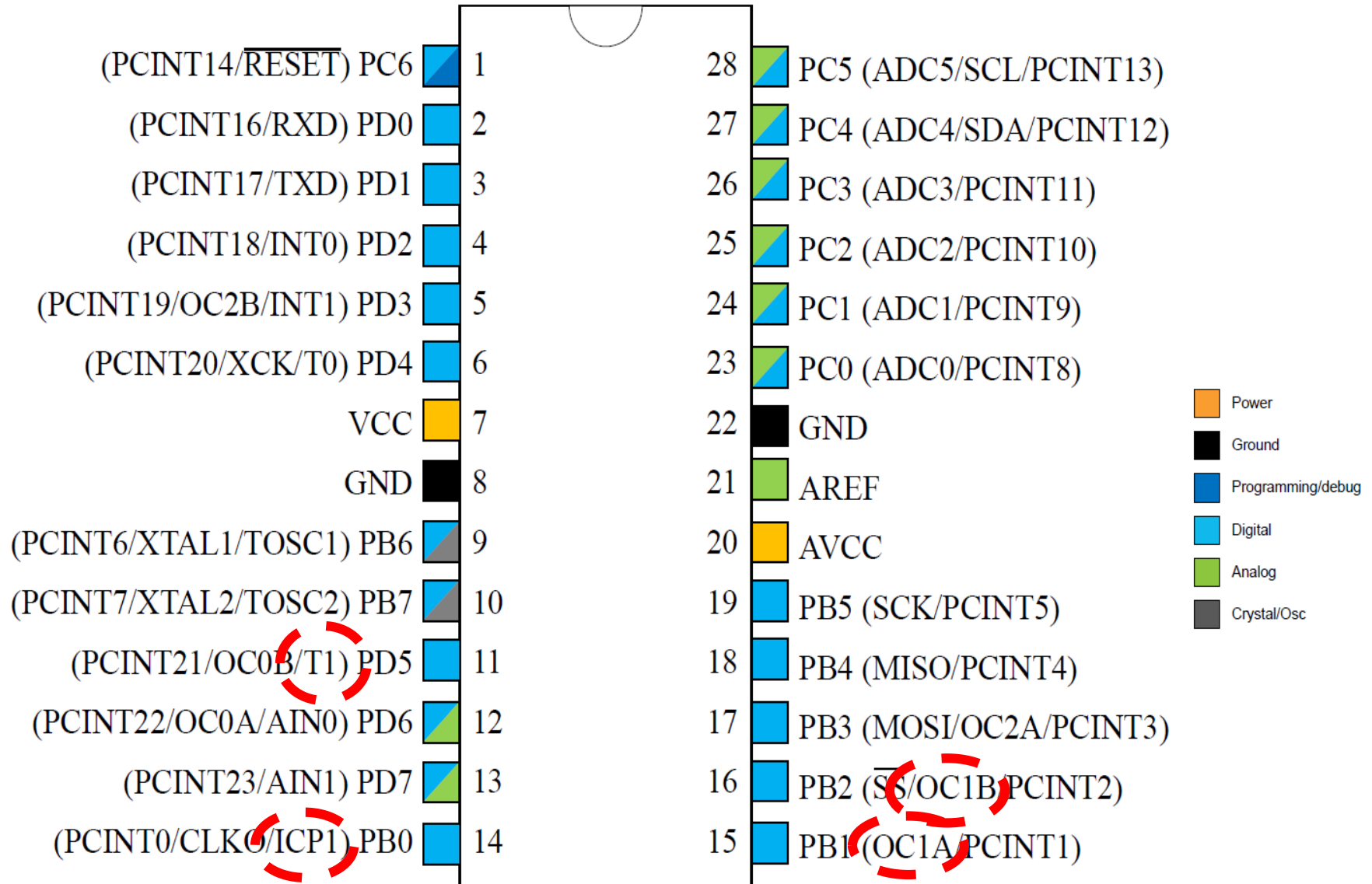
7	6	5	4	3	2	1	0	
–	–	–	–	–	OCF0B	OCF0A	TOV0	TIFR0
R	R	R	R	R	R/W	R/W	R/W	

TIMER 1 – 16 bits

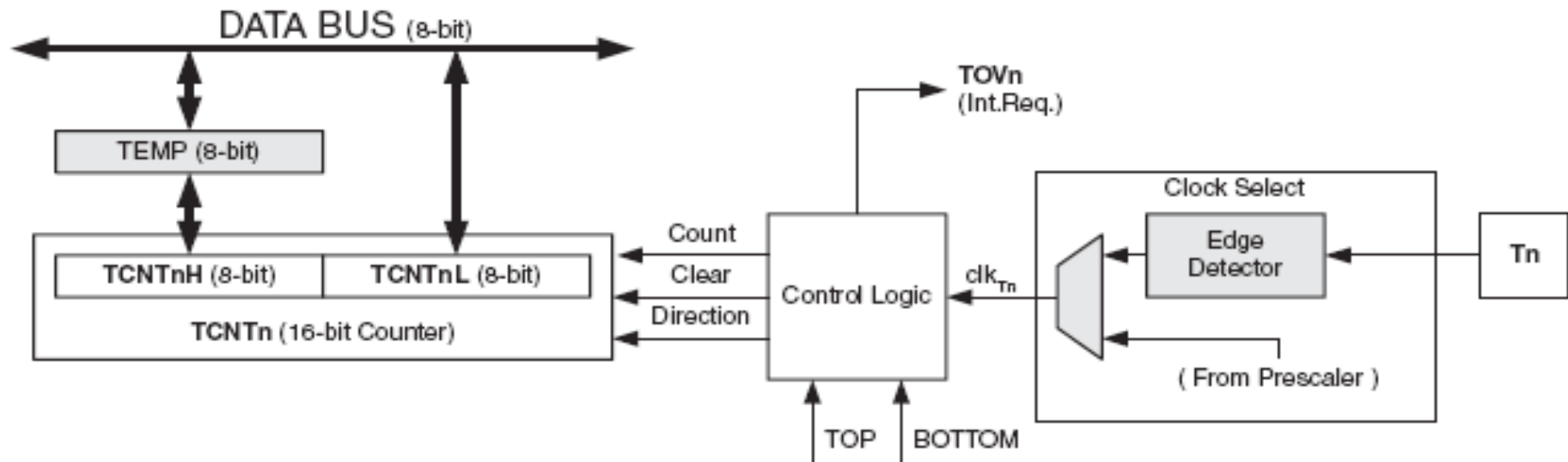
- T1 es la entrada externa de reloj
- TCNT1 es el contador de 16 bits
- OCR1A y B son los registros de comparación
- OC1A y B son los terminales de salida para generación de señales
- OCF1 A y B son los flags de comparación
- ICR1 capturador de entrada
- TCCR1A y B registros de configuración



TIMER 1 en ATMEGA328

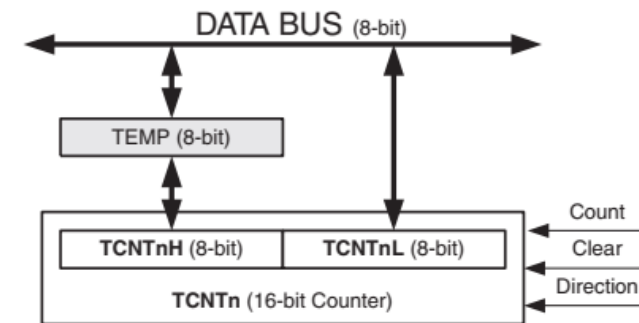


TIMER 1



- Count: incrementa o decrementa TCNT1 en 1.
- Direction: selecciona entre incremento o decremento.
- Clear: borra el TCNT1.
- clk_{T1}: señal de Reloj del Timer/Contador 1.
- TOP: valor máximo que alcanza TCNT1.
- BOTTOM: valor mínimo que alcanza TCNT1.
- TOV1: flag de overflow (0xFFFF -> 0x0000)

TIMER 1



- Lectura y escritura secuencial de los registros de 16 bits
¿Como accedemos a un registro de 16bits?
 - Se dispone de un registro **TEMP** de 8bits para el almacenamiento del byte alto
 - Cuando se lee o escribe el byte mas bajo, el registro TEMP también, formando 16 bits.

Operation	1st Access	2nd Access
Read	Low Byte	High Byte
Write	High Byte	Low Byte

En Asm:

```
...
; Set TCNT1 to 0x01FF
ldi r17,0x01
ldi r16,0xFF
out TCNT1H,r17
out TCNT1L,r16
; Read TCNT1 into r17:r16
in r16,TCNT1L
in r17,TCNT1H
...
```

En C:

```
unsigned int i;
:.
/* Set TCNT1 to 0x01FF */
TCNT1 = 0x1FF;
/* Read TCNT1 into i */
i = TCNT1;
```

TIMER 1

- 16 Modos de funcionamiento: Normal, CTC, y PWM (según 4 bits WGM1[0:3])
- TOP: es el valor mas alto que cuenta el TCNT
 - En modo Normal es 0xFFFF
 - En algunos modos puede ser 0xFF, 0x1FF, 0x3FF
 - En otros modos se especifica en el OCR1A o en ICR1

• MAX= 0xFFFF

• BOTTOM=0

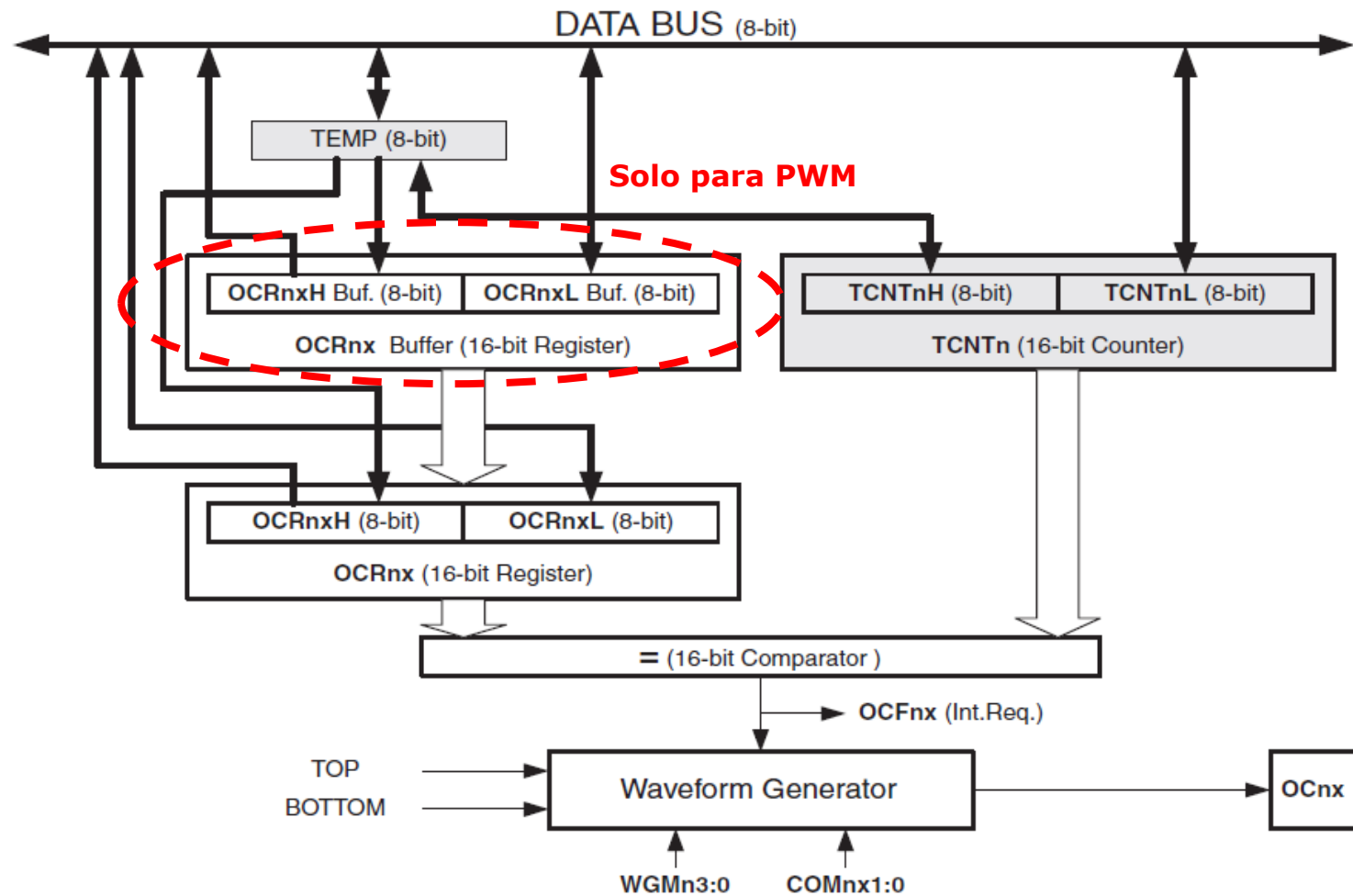
Mode	WGM13	WGM12	WGM11	WGM10	Timer/Counter Mode of Operation	Top	Update of OCR1x	TOV1 Flag Set on
0	0	0	0	0	Normal	0xFFFF	Immediate	MAX
1	0	0	0	1	PWM, Phase Correct, 8-bit	0x00FF	TOP	BOTTOM
2	0	0	1	0	PWM, Phase Correct, 9-bit	0x01FF	TOP	BOTTOM
3	0	0	1	1	PWM, Phase Correct, 10-bit	0x03FF	TOP	BOTTOM
4	0	1	0	0	CTC	OCR1A	Immediate	MAX
5	0	1	0	1	Fast PWM, 8-bit	0x00FF	TOP	TOP
6	0	1	1	0	Fast PWM, 9-bit	0x01FF	TOP	TOP
7	0	1	1	1	Fast PWM, 10-bit	0x03FF	TOP	TOP
8	1	0	0	0	PWM, Phase and Frequency Correct	ICR1	BOTTOM	BOTTOM
9	1	0	0	1	PWM, Phase and Frequency Correct	OCR1A	BOTTOM	BOTTOM
10	1	0	1	0	PWM, Phase Correct	ICR1	TOP	BOTTOM
11	1	0	1	1	PWM, Phase Correct	OCR1A	TOP	BOTTOM
12	1	1	0	0	CTC	ICR1	Immediate	MAX
13	1	1	0	1	Reserved	-	-	-
14	1	1	1	0	Fast PWM	ICR1	TOP	TOP
15	1	1	1	1	Fast PWM	OCR1A	TOP	TOP

TIMER 1

- En resumen:
 - Los modos de operación básicos son los mismos del Timer 0 y 2:
 - Normal, CTC, Fast PWM y Phase Correct
 - Posee 2 modos adicionales de PWM
 - El valor TOP es programable con el OCR1A y el ICR1 (modos CTC)
 - Existe un orden específico para la lectura y escritura de los reg. de 16bits (se las arregla el compilador si utilizamos variables de 16bits)
 - Posee 2 generadores de señales A Y B y un capturador de entrada
- Veremos estos subsistemas a continuación...

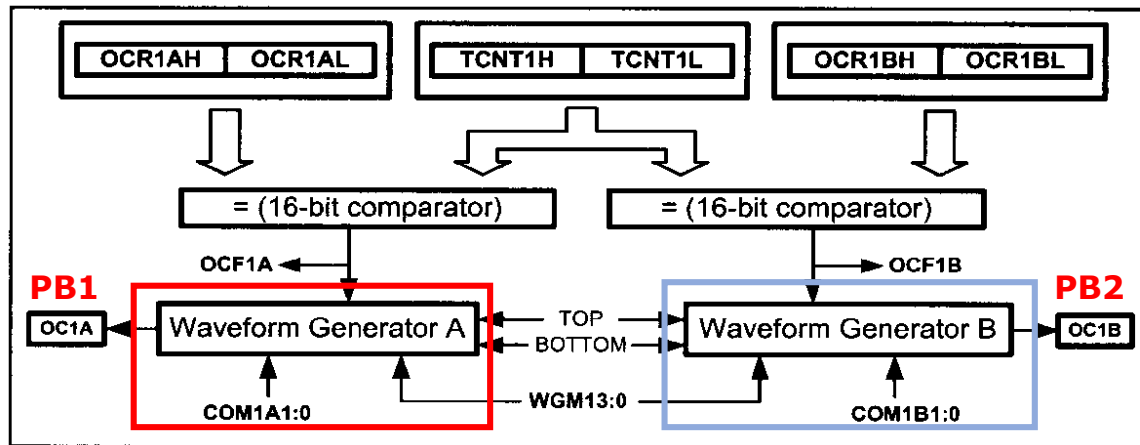
TIMER 1

- Generador de señales

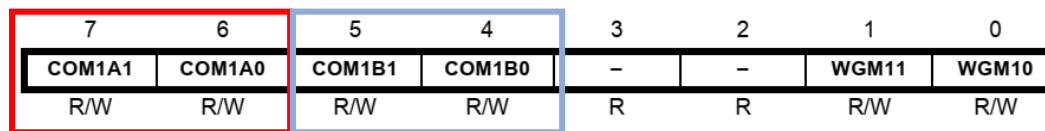


TIMER 1

- Generadores de señales Atmega328p: 2 canales independientes



- TCCR1A:



COM1A1:COM1A0 D7 D6 Compare Output Mode for Channel A

COM1A1	COM1A0	Description
0	0	Normal port operation, OC1A disconnected
0	1	Toggle OC1A on compare match
1	0	Clear OC1A on compare match
1	1	Set OC1A on compare match

Tabla para cualquiera de los modos No PWM

Hay otra tabla para el canal B

TIMER 1

- TCCR1B:

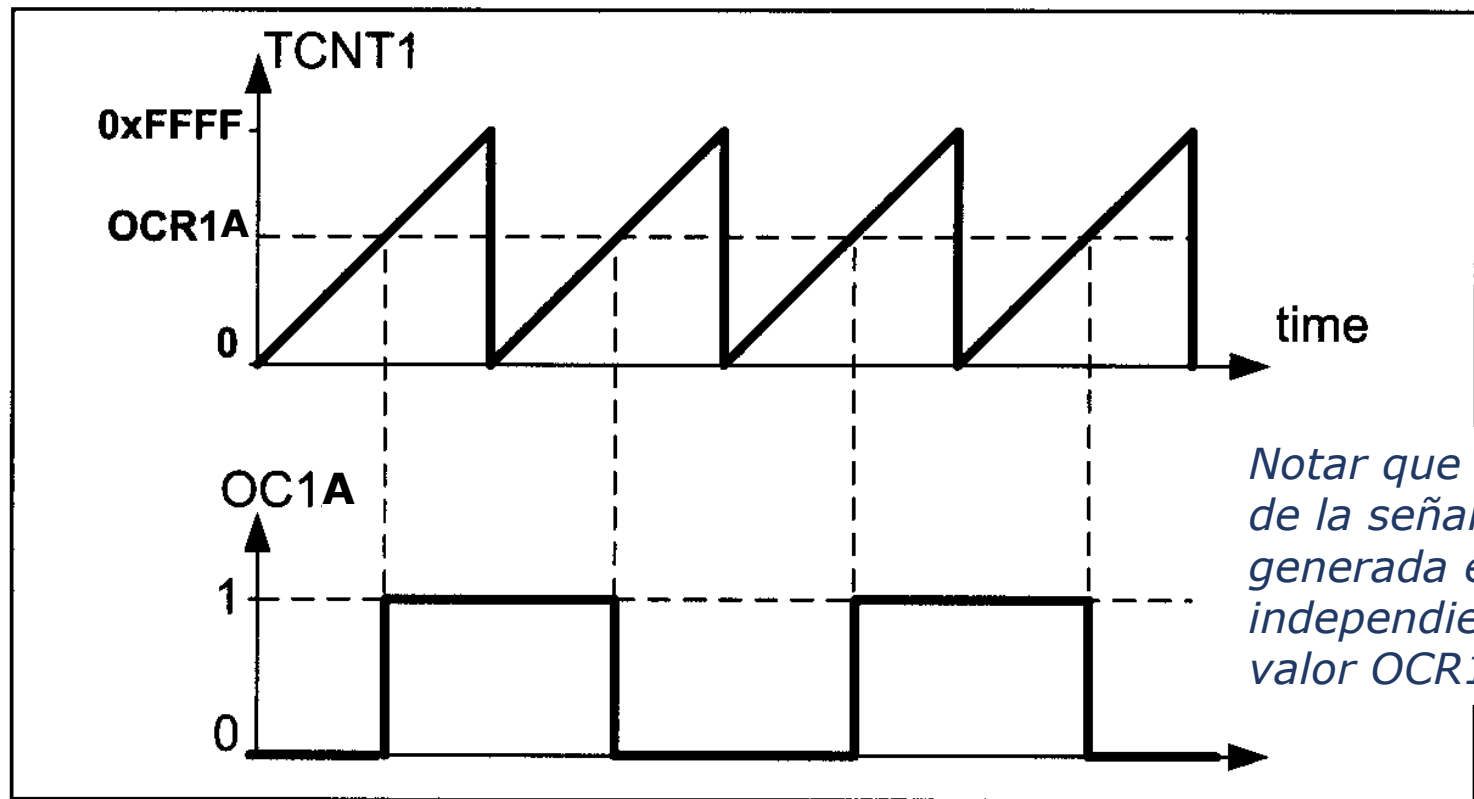
Bit	7	6	5	4	3	2	1	0	
	ICNC1	ICES1	–	WGM13	WGM12	CS12	CS11	CS10	TCCR1B
Read/Write	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- ICNC1: Noise canceler enable (4 samples @fcpu)
- ICES1: selección de flanco 0: bajada, 1: subida
- WGM1x: ver modos de funcionamientos en tabla de Modos.
- CS1x: clock select

CS12	CS11	CS10	Description
0	0	0	No clock source (Timer/Counter stopped).
0	0	1	$\text{clk}_{\text{IO}}/1$ (No prescaling)
0	1	0	$\text{clk}_{\text{IO}}/8$ (From prescaler)
0	1	1	$\text{clk}_{\text{IO}}/64$ (From prescaler)
1	0	0	$\text{clk}_{\text{IO}}/256$ (From prescaler)
1	0	1	$\text{clk}_{\text{IO}}/1024$ (From prescaler)
1	1	0	External clock source on T1 pin. Clock on falling edge.
1	1	1	External clock source on T1 pin. Clock on rising edge.

TIMER 1

- Ejemplo: Generación de señal cuadrada con modo Normal
=>WGM1[0:3] en 0
=>COM1A1=0 COM1A0=1 (toggle)



Notar que el periodo de la señal generada es independiente del valor OCR1A

Figure 15-15. Generating Square Wave Using Normal Mode and Toggle Mode

TIMER 1

- Ejemplo: generación de señal cuadrada en modo 4 CTC

```
#include <avr/io.h>

int main(void)
{
    DDRB = (1<<PB1);           //PB1 =salida
    TCCR1A = (1<<COM1A0);       //COM1A =Toggle
    TCCR1B = (1<<WGM12) | (1<<CS10); //Modo CTC No prescaler
    OCR1A = 512;                //TOP=512
    while(1);
    return 0;
}
```

- Frecuencia de la señal generada:

$512 + 1 = 513$ clocks and

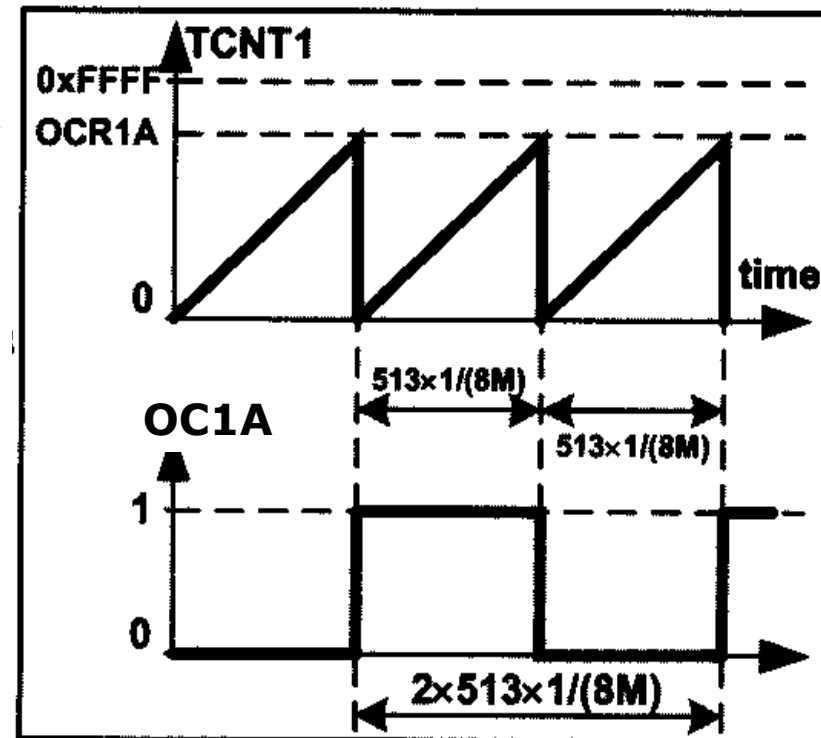
$T_{\text{timer clock}} = 1 / 8 \text{ MHz} = 0.125 \mu\text{s}$

$T_{\text{wave}} = 2 \times 513 \times 0.125 \mu\text{s} = 128.25 \mu\text{s}$

$F_{\text{wave}} = 1 / 128.25 \mu\text{s} = 7797 \text{ Hz} = 7.797 \text{ kHz}$

- Expresión general:

$$f_{OCnA} = \frac{f_{\text{clk_I/O}}}{2 \cdot N \cdot (1 + OCRnA)}$$



TIMER 1

$$f_{OCnA} = \frac{f_{clk_I/O}}{2 \cdot N \cdot (1 + OCRnA)}$$

Ahora el periodo de la señal generada depende del valor OCR1A

- El modo CTC es preferible al Normal porque permite ajustar el período a generar

Si:

$$f_{clk_I/O} = 16\text{MHz}$$

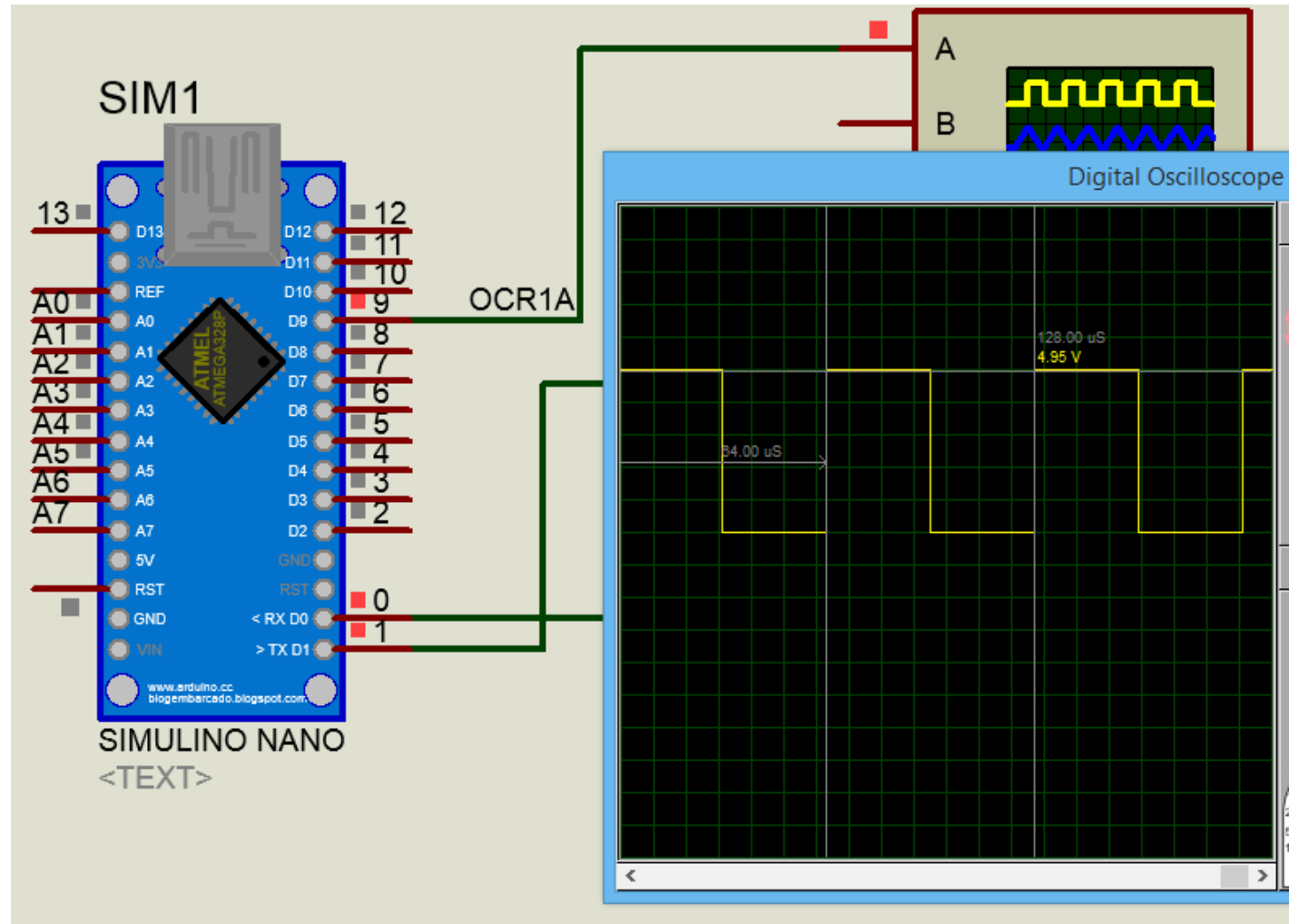
$$N=1 \text{ y}$$

$$OCR1A=512$$

Entonces:

$$f_{OC1A} = 15,59\text{KHz}$$

$$T_{OC1A} = 64,1\mu\text{s}$$



TIMER 1

$$f_{OCnA} = \frac{f_{clk_I/O}}{2 \cdot N \cdot (1 + OCRnA)}$$

- ¿Cuántas señales de distinto periodo puedo generar?

Para $f_{clk_I/O}$ y N fijos \Rightarrow OCR1A variable(16bits)
por lo tanto 65536 valores diferentes de TOC1A o fOC1A

- Cuál es el mínimo periodo?

Para $f_{clk_I/O}$ y N fijos, OCR1A=0 \Rightarrow $T_{OC1A_Min} = 2N \cdot 1 \cdot T_{clk_I/O}$

- Cuál es el máximo periodo?

Para $f_{clk_I/O}$ y N fijos, OCR1A=65535 \Rightarrow $T_{OC1A_Max} = 2N \cdot 65536 \cdot T_{clk_I/O}$

- Cuál es la resolución del periodo según definimos en la clase09 ?

Para $f_{clk_I/O}$ y N fijos \Rightarrow diferencia entre dos periodos

$$T_{OC1A_x} = 2N \cdot (1 + OCR1Ax) \cdot T_{clk_I/O}$$

$$T_{OC1A_y} = 2N \cdot (1 + OCR1Ay) \cdot T_{clk_I/O}$$

$$\Rightarrow \Delta T_{OC1A} = T_{OC1A_x} - T_{OC1A_y} = 2N \cdot T_{clk_I/O} \cdot (OCR1Ax - OCR1Ay)$$

La mínima diferencia es $(OCR1Ax - OCR1Ay) = \pm 1$ (1LSB)

\Rightarrow La resolución es $\Delta T_{OC1A \min} = \pm 2N \cdot T_{clk_I/O}$

Y la resolución relativa en % es $\Delta T_{OC1A \min} / T_{OC1A} = \pm 100 / (1 + OCR1A)$

TIMER 1

$$f_{OCnA} = \frac{f_{clk_I/O}}{2 \cdot N \cdot (1 + OCRnA)}$$

- Por ejemplo:

Para $f_{clk_I/O} = 8\text{MHz}$ y $N = 64$

- $OCR1A = 453$

$$T_{OC1A} = 2 \cdot 64 \cdot 454 / 8\text{MHz} = 7,264\text{ms}$$

- $OCR1A = 14578$

$$T_{OC1A} = 2 \cdot 64 \cdot 14579 / 8\text{MHz} = 233,264\text{ms}$$

- Resolución del método:

$$\Delta T_{OC1A \text{ min}} = \pm 2 \cdot 64 / 8\text{MHz} = \pm 0,016\text{ms}$$

- Resolución relativa en cada caso:

$$\Delta T_{OC1A \text{ min}} / T_{OC1A} = \pm 100 / (1 + 453) = \pm 0,22\%$$

$$\Delta T_{OC1A \text{ min}} / T_{OC1A} = \pm 100 / (1 + 14579) = \pm 0,07\%$$

Notar que asumimos $f_{clk_I/O} = 8.000.000 \text{ Hz} \Rightarrow$ debemos sumar al error relativo la tolerancia del reloj

TIMER 1

- Práctica 3

EJERCICIO PARA REALIZAR CON SIMULADOR

Ejercicio No 2: TIMER1

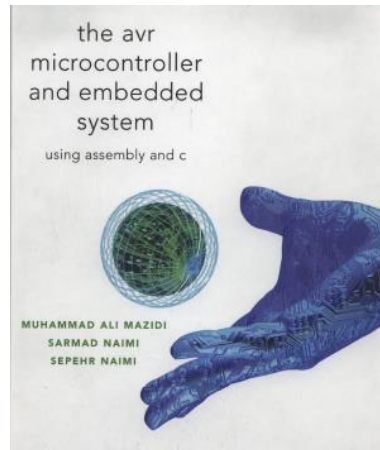
a) Los generadores de señal del TIMER 1, pueden configurarse en distintos modos que permiten generar eventos de tiempo precisos y reducir la latencia que tendrían si el software interviene. Por ejemplo, un sintetizador de sonidos requiere generar los tonos musicales de acuerdo a la frecuencia de las notas seleccionadas. Muestre como utilizar un canal en modo Output Compare para generar señales digitales periódicas con la frecuencia de las notas musicales.

```
// The following array stores the frequencies for the musical notes [Hz]
uint16_t note[12] =
{
//C      C#    D      D#    E      F      F#    G      G#    A      A#    B
 262, 277, 294, 311, 330, 349, 370, 392, 415, 440, 466, 494    //4th octave
};
```

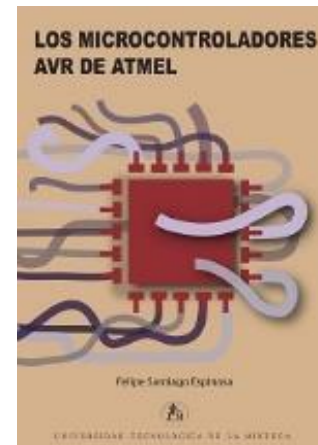
Simular en Proteus la generación de las distintas señales con el TIMER1 modo 2 CTC. Utilice los instrumentos del simulador para verificar la frecuencia de la señal generada. También puede colocar un speaker y escuchar las notas.

Bibliografía:

- *The AVR microcontroller & Embedded Systems*. Mazidi,Naimis, CH15



- Libro Digital de Felipe Espinosa, CH4



- Hojas de datos ATMEGA328p, 2560