

# CIRCUITOS DIGITALES Y MICROCONTROLADORES 2022

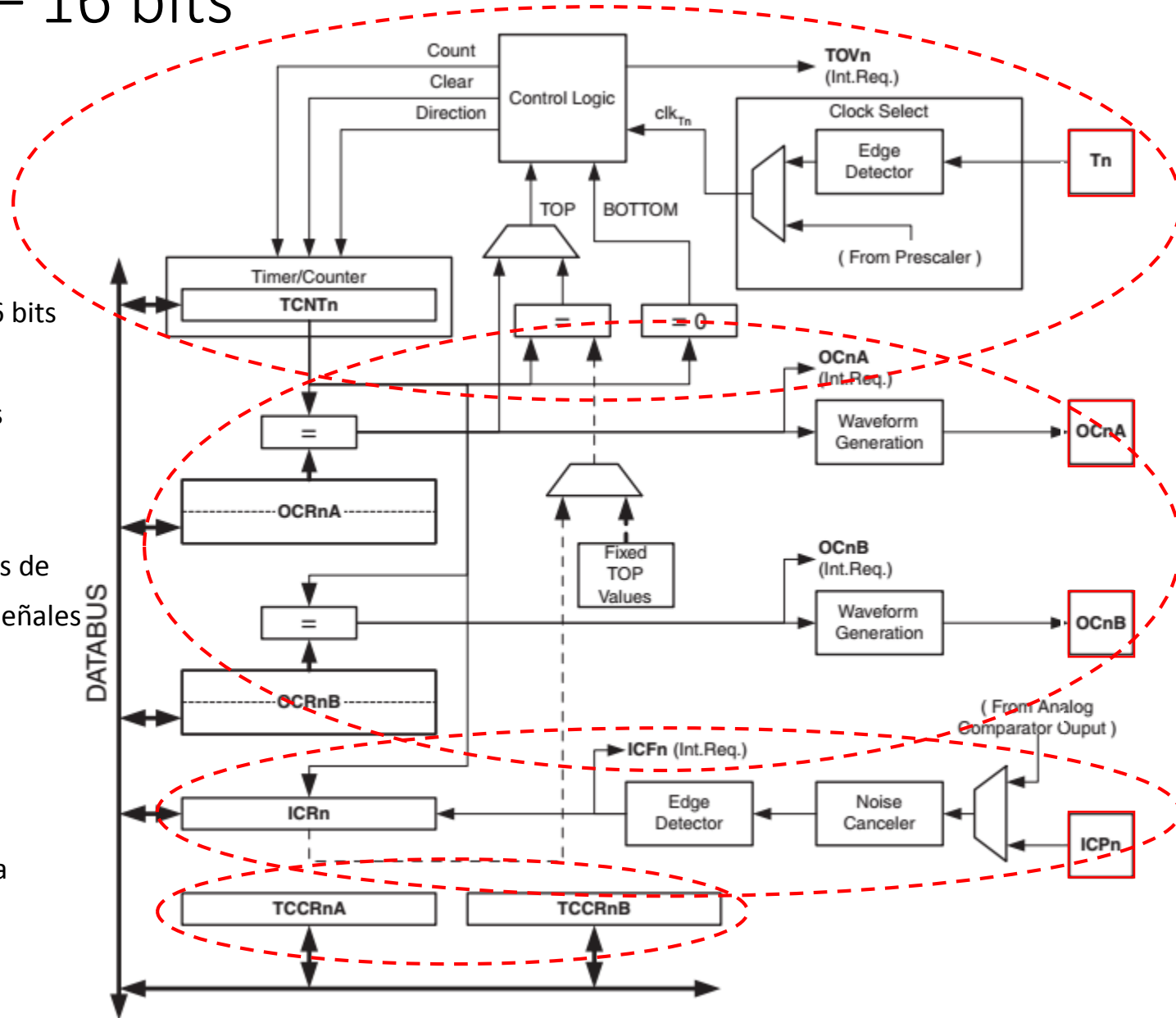
Facultad de Ingeniería  
UNLP

Periférico TIMER1 (continuación)

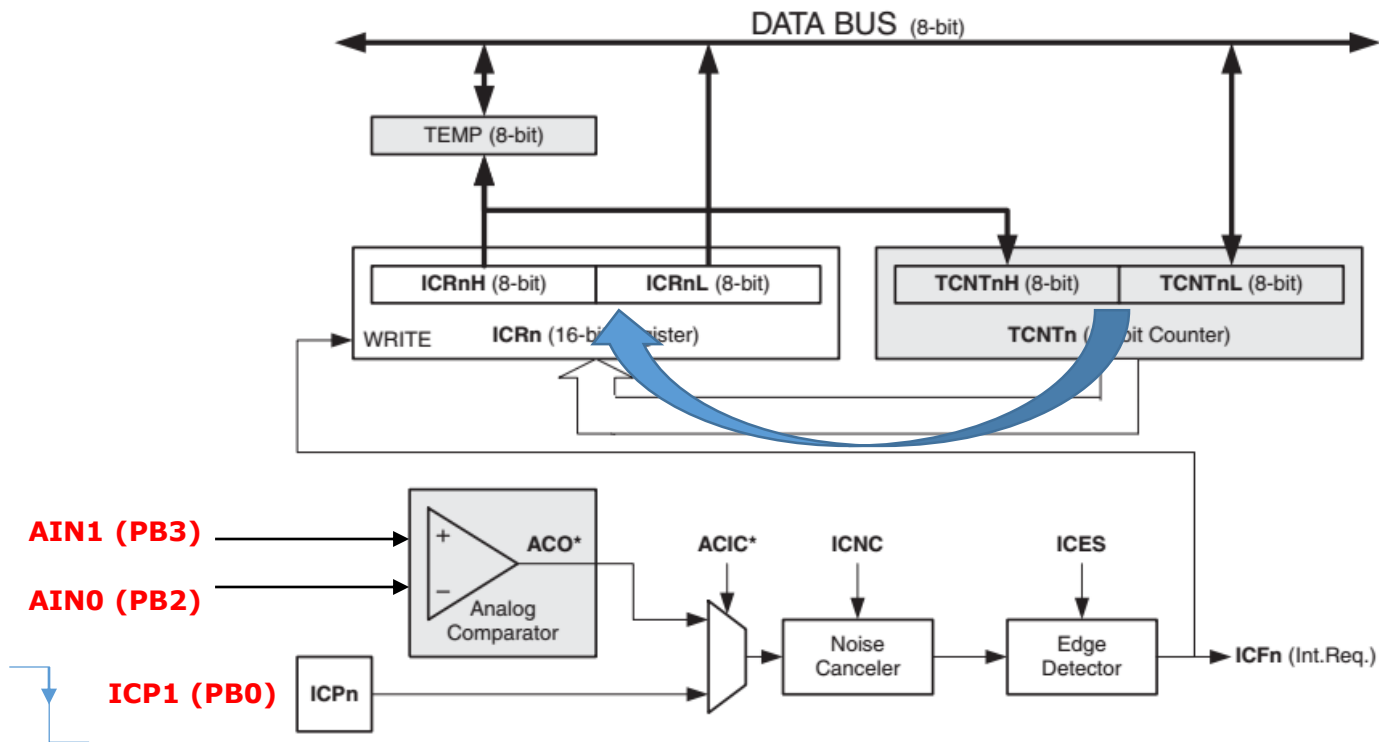
Ing. José Juárez

# 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: Capturador de entrada



- Cuando ocurre un flanco en el terminal ICP1 (PB0) se dispara una captura que copia el TCNT1 en el ICR1, se activa el flag ICF1 y si esta habilitada (TICIE1=1) se genera una interrupción de aviso de captura
- El comparador analógico, compara la tensión de AIN1(PB3) contra AIN0 (PB2) y si es mayor da una salida "1" (veremos este periférico en TP4)

# TIMER 1

- Registros de configuración y control:

7	6	5	4	3	2	1	0	
<b>COM1A1</b>	<b>COM1A0</b>	<b>COM1B1</b>	<b>COM1B0</b>	–	–	<b>WGM11</b>	<b>WGM10</b>	<b>TCCR1A</b>
R/W	R/W	R/W	R/W	R	R	R/W	R/W	
0	0	0	0	0	0	0	0	

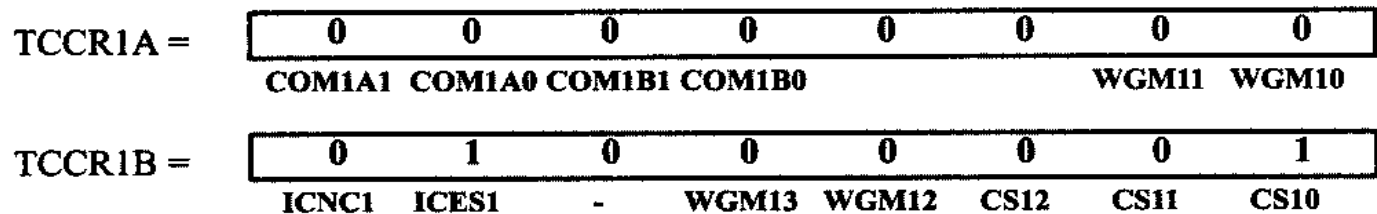
7	6	5	4	3	2	1	0	
<b>ICNC1</b>	<b>ICES1</b>	–	<b>WGM13</b>	<b>WGM12</b>	<b>CS12</b>	<b>CS11</b>	<b>CS10</b>	<b>TCCR1B</b>
R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	
0	0	0	0	0	0	0	0	

- 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.

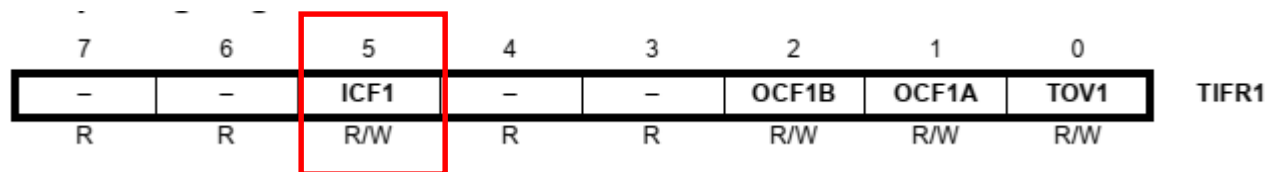
# TIMER 1

- Configuración del Capturador:



Modo Normal, sin prescaler, sin cancelador de ruido, con captura en flanco +

Registro de Flags TIFR1:



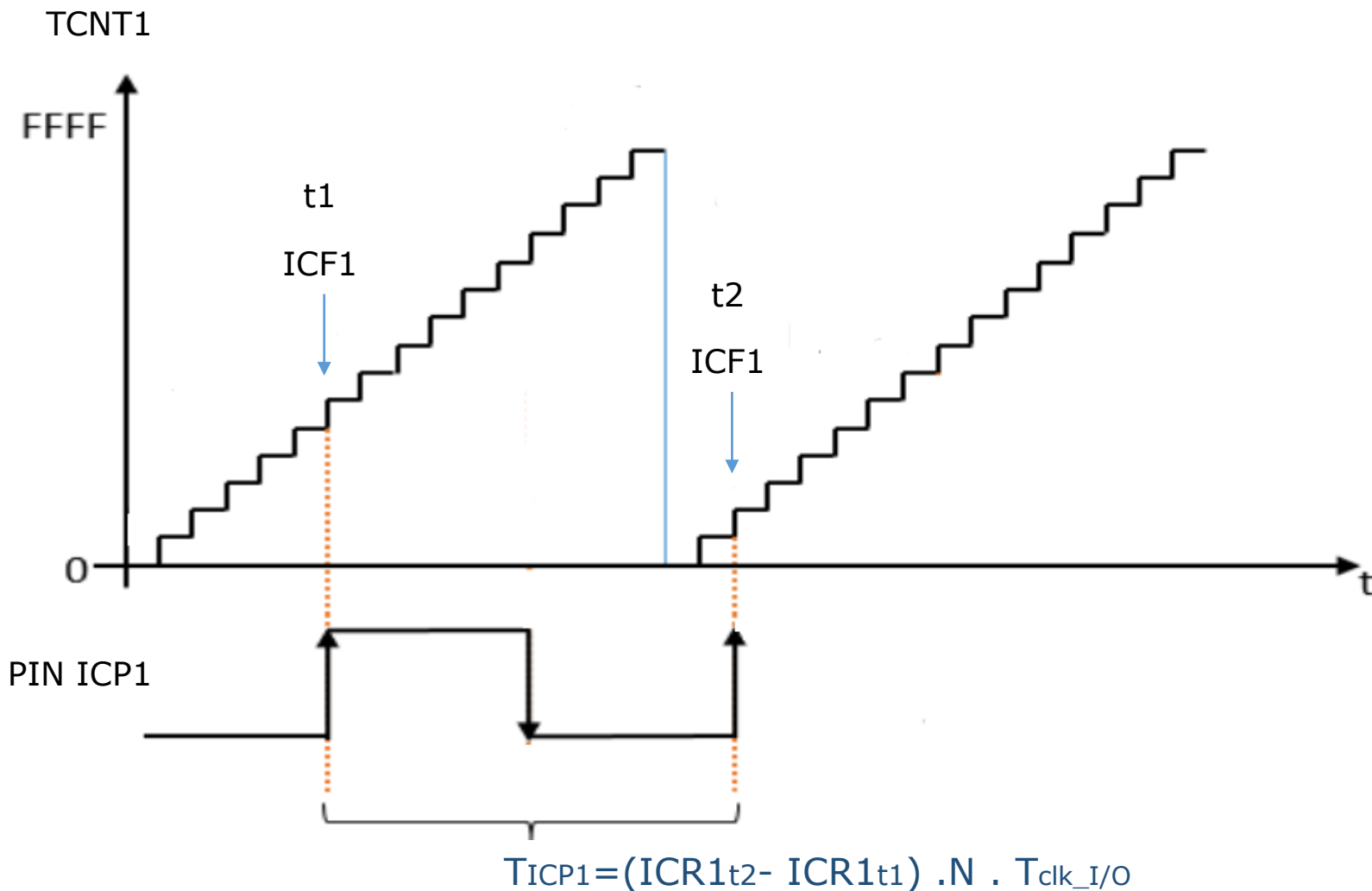
El Flag ICF1 se activará cuando ocurra un flanc + en terminal ICP1

Luego se debe borrar escribiendo un "1":  $TIFR1 \mid= (1 \ll ICF1)$  ;

O se borra automáticamente cuando se ejecuta el handler de interrupción

# TIMER 1      Modo Normal -Input Capture

Ejemplo: medición de periodo de una señal digital incógnita



# TIMER 1

$$T_{ICP1} = (ICR1_{t2} - ICR1_{t1}) \cdot N \cdot T_{clk\_I/O}$$

- ¿Cuántos valores distintos puedo medir?

Para  $f_{clk\_I/O}$  y  $N$  fijos  $\Rightarrow$   $ICR1$  variable (16bits)  
por lo tanto 65535 valores diferentes de  $T_{ICP1}$  o  $f_{ICP1}$

- Cuál es el mínimo periodo?

$$\Rightarrow |ICR1_{t2} - ICR1_{t1}| = 1 \Rightarrow T_{ICP1\_Min} = 1 \cdot N \cdot T_{clk\_I/O}$$

- Cuál es el máximo periodo?

$$\Rightarrow |ICR1_{t2} - ICR1_{t1}| = 65535 \Rightarrow T_{ICP1\_Max} = 65535 \cdot N \cdot T_{clk\_I/O}$$

- Cuál es la resolución del método?

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

$$T_{ICP1\_x} = DIF\_x \cdot N \cdot T_{clk\_I/O}$$

$$T_{ICP1\_y} = DIF\_y \cdot N \cdot T_{clk\_I/O}$$

$$\Rightarrow \Delta T = T_{ICP1\_y} - T_{ICP1\_x} = N \cdot T_{clk\_I/O} \cdot (DIF\_y - DIF\_x)$$

La mínima diferencia es  $\cdot (DIF\_y - DIF\_x) = \pm 1$

$$\Rightarrow \text{La resolución es } \Delta T_{min} = \pm N \cdot T_{clk\_I/O}$$

$$\text{Y la resolución relativa en \% es } \Delta T_{min} / T_{ICP1} = \pm 100 / |ICR1_{t2} - ICR1_{t1}|$$

# TIMER 1

$$T_{ICP1} = (ICR1_{t2} - ICR1_{t1}) \cdot N \cdot T_{clk\_I/O}$$

- Ejemplo: **Medición de período** y mostrar en PORTC y D

```
#include <avr/io.h>
int main(void)
{
    uint16_t t, periodo;
    DDRB &= ~(1<<PB0);           //ICP1 =entrada
    DDRC = 0xFF;                  //PORTC =salida
    DDRD = 0xFF;                  //PORTD =salida

    TCCR1A = 0;                   //Modo normal
    TCCR1B = (1<<ICES1) | (1<<CS10); //ICP1 flaco + No prescaler

    while(TIFR1 & (1<<ICF1) == 0);
    t = ICR1;
    TIFR1 = (1<<ICF1);
    while(TIFR1 & (1<<ICF1) == 0);
    periodo = ICR1 - t;
    TIFR1 = (1<<ICF1);

    PORTD = periodo;              // Low byte
    PORTC = periodo >> 8;         //High byte

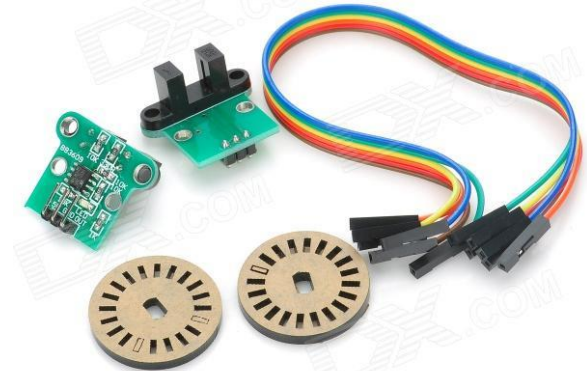
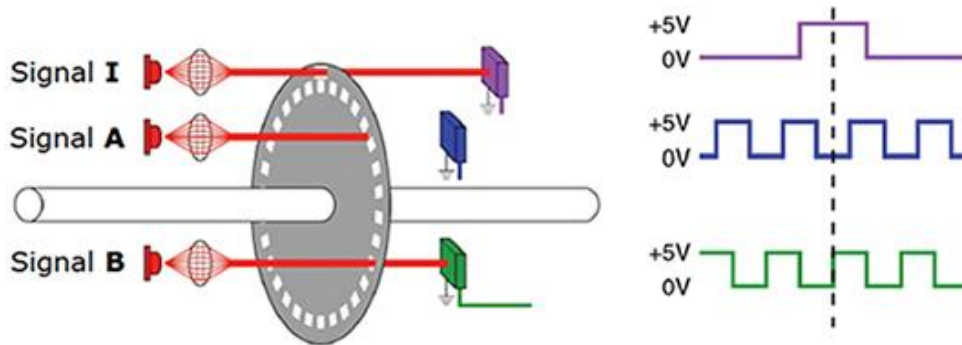
    while(1);
    return 0;
}
```



# TIMER 1

$$T_{ICP1} = (ICR1_{t2} - ICR1_{t1}) \cdot N \cdot T_{clk\_I/O}$$

- **Ejemplo:** Medición de RPM de un motor con un encoder



A y B desfasados 1/4 de vuelta para detectar el sentido de giro  
Supongamos que el encoder tiene 25 pulsos/vuelta, entonces

$$1 \text{ vuelta tarda } 25 \cdot T_{pulso} [\text{seg}]$$

por lo tanto en 1 minuto tendremos:

$$RPM = 60 / (25 \cdot T_{pulso})$$

Si utilizamos la entrada de captura para medir entonces  $T_{pulso} = T_{ICP1}$

Por lo tanto:

$$RPM = 60 / (25 \cdot DIF_{ICR1} \cdot N \cdot T_{clk\_I/O})$$

# TIMER 1

- calcular el rango y la resolución del valor en RPM

$$RPM[rpm] = \frac{60 fclk_{IO}}{25 DIF N}$$

$$RPM_{min}[rpm] = \frac{60 fclk_{IO}}{25 DIF N} = \frac{60 fclk_{IO}}{25.65535 N}$$

$$RPM_{max}[rpm] = \frac{60 fclk_{IO}}{25 DIF N} = \frac{60 fclk_{IO}}{25 N}$$

$$DIF' = DIF + 1LSB$$

$$\Delta RPM_{min}[rpm] = \frac{60 fclk_{IO}}{25 DIF' N} - \frac{60 fclk_{IO}}{25 DIF N} \cong \pm \frac{60 fclk_{IO}}{25 DIF^2 N}$$

$$\frac{\Delta RPM_{min}}{RPM} [\%] \cong \pm \frac{100}{DIF}$$

*Lo que significa que debo seleccionar el prescaldor N para diferencias grandes . Por ejemplo si quiero medir con resolución  $< \pm 0,1\%$  entonces  $|ICR1t2 - ICR1t1| > 1000$*

# TIMER 1

Ejemplo: **Medición de ancho de pulso** (asumiendo que el valor a medir cabe en 8 bits)

---

```
#include <avr/io.h>
int main(void)
{
    uint16_t t,apulso;
    DDRB &= ~(1<<PB0);           //ICP1 =entrada
    DDRC = 0xFF;                  //PORTC =salida
    DDRD = 0xFF;                  //PORTD =salida

    TCCR1A = 0;                   //Modo normal
    TCCR1B = (1<<ICES1) | (1<<CS10); //ICP1 flaco + No prescaler

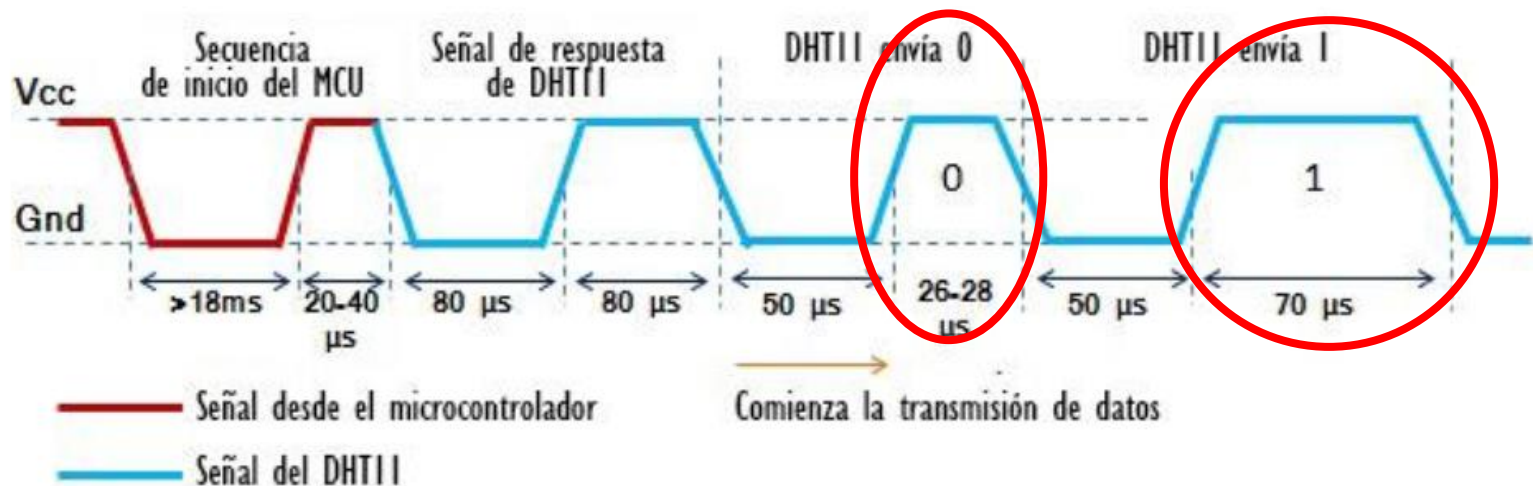
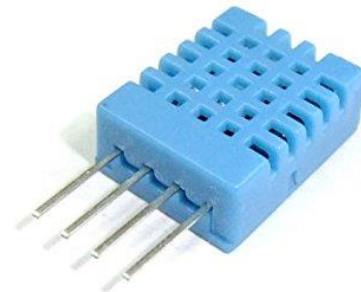
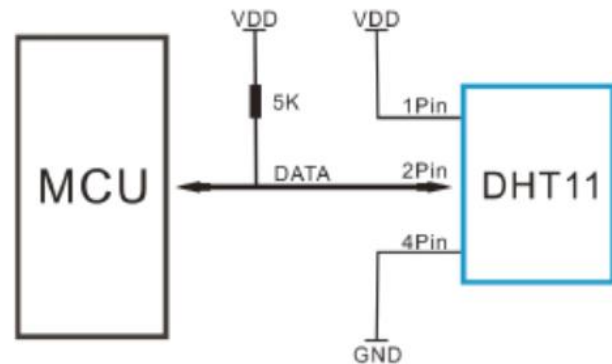
    while (TIFR1 & (1<<ICF1) == 0);
    t=ICR1;
    TIFR1=(1<<ICF1);
    TCCR1B &= ~(1<<ICES1); //ICP1 flaco -
    while (TIFR1 & (1<<ICF1) == 0);
    apulso=ICR1-t;
    TIFR1=(1<<ICF1);

    PORTD= apulso;                // Low byte
    PORTC= apulso>>>8;           //High byte

    while(1);
    return 0;
}
```

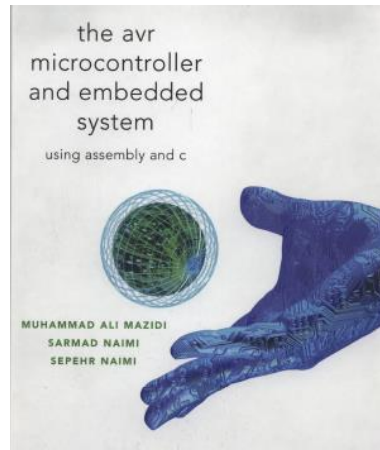
# TIMER 1

- Ejemplo: Medición de Temperatura y Humedad relativa con sensor digital

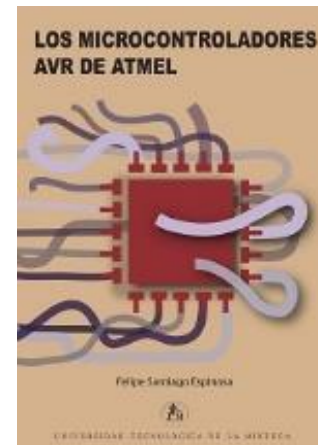


# Bibliografía:

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



- Libro Digital de Felipe Espinosa, CH4



- Hojas de datos ATMEGA328p, 2560