



# FACULTAD DE INGENIERIA

Universidad de Buenos Aires

Laboratorio de Microprocesadores - 86.07

## Trabajo Práctico N°6: Timers

Profesor:			Ing. Guillermo Campiglio									
Cuatrimestre/Año:			1°/2020									
Turno de las clases prácticas			Miércoles									
Jefe de trabajos prácticos:			Ing. Pedro Ignacio Martos									
Docente guía:			-									
Autor			Seguimiento del proyecto									
Francisco	Rossi	99540										

### Observaciones:

.....

.....

.....

.....

.....

.....

.....

.....

.....

Fecha de aprobación			Firma J.T.P		

Coloquio	
Nota final	
Firma profesor	

# Índice

<b>1. Introducción</b>	<b>2</b>
1.1. Objetivo . . . . .	2
1.2. Descripción . . . . .	2
<b>2. Materiales</b>	<b>2</b>
<b>3. Diagrama en Bloques</b>	<b>2</b>
<b>4. Esquemático</b>	<b>3</b>
<b>5. Diagrama de flujo</b>	<b>4</b>
<b>6. Código</b>	<b>5</b>
<b>7. Resultados</b>	<b>7</b>
<b>8. Conclusiones</b>	<b>7</b>

## 1. Introducción

En el siguiente informe se explica el diseño de un programa escrito en lenguaje Assembler con el cual a partir de las cuatro configuraciones posibles compuestas por dos switches se elije la frecuencia de oscilación de un LED conectado a PB0.

### 1.1. Objetivo

El objetivo es utilizar el **TIMER1** de 16 bits del arduino para hacer oscilar un LED a una frecuencia determinada.

### 1.2. Descripción

Se conectan dos switches, uno a PD0 y PD1 los cuales determinaran los valores de ambos pines, conectados como se indica en la **Fig. 3**. A partir de los cuales se determinará el encendido del LED según la **Tabla. 1**.

Tabla 1: Comportamiento del LED en función de las entradas PD0 yPD1.

PD0	PD1	Estado del LED
0	0	Encendido fijo
0	1	Parpadeo con prescaler clk/64
1	0	Parpadeo con prescaler clk/256
1	1	Parpadeo con prescaler clk/1024

## 2. Materiales

Se utilizaron los siguiente materiales para el proyecto:

- 1 LED (20\$ (Pesos Argentinos))
- 2 Resistores de  $220\ \Omega$  y 1 resistor de  $10\ k\Omega$  (12\$ (Pesos Argentinos))
- 1 Microcontrolador ATmega328p (Utilizando el integrado con el Arduino Uno) (700\$ (Pesos Argentinos))

## 3. Diagrama en Bloques

En la **Fig. 1** se muestra un diagrama en bloques del circuito.

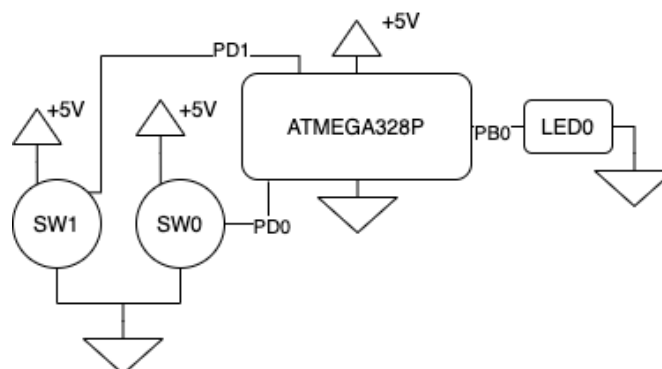


Figura 1: Diagrama en bloques.

## 4. Esquemático

En las **Fig. 3** y **Fig. 2** se muestra como se conectó el arduino con los LEDs, el potenciómetro y los resistores de  $220\ \Omega$ . Si bien se sugería utilizar dos resistores de  $10\ k\Omega$ , debido a la corriente de fuga en *PD0* se generaba una tensión en el pin de aprox.  $4,55\ V$  dando lugar a un falso valor positivo, se debió modificar el valor de resistencia a una más baja, reduciendo así el valor de tensión residual de este puerto.

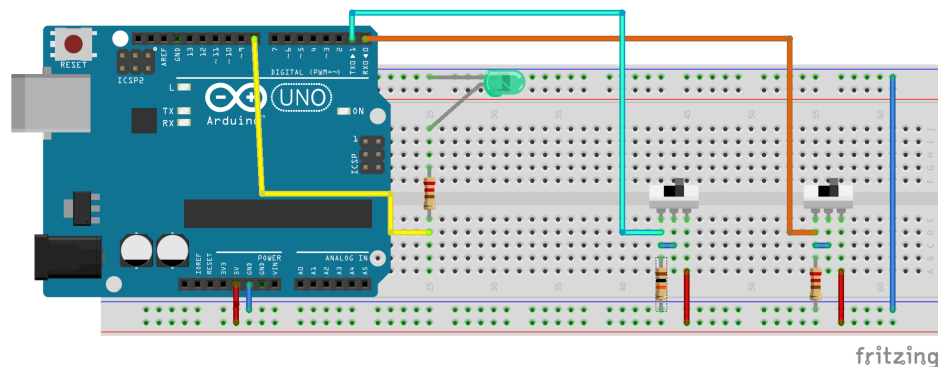


Figura 2: Conexión entre el arduino, los Switches y el LED

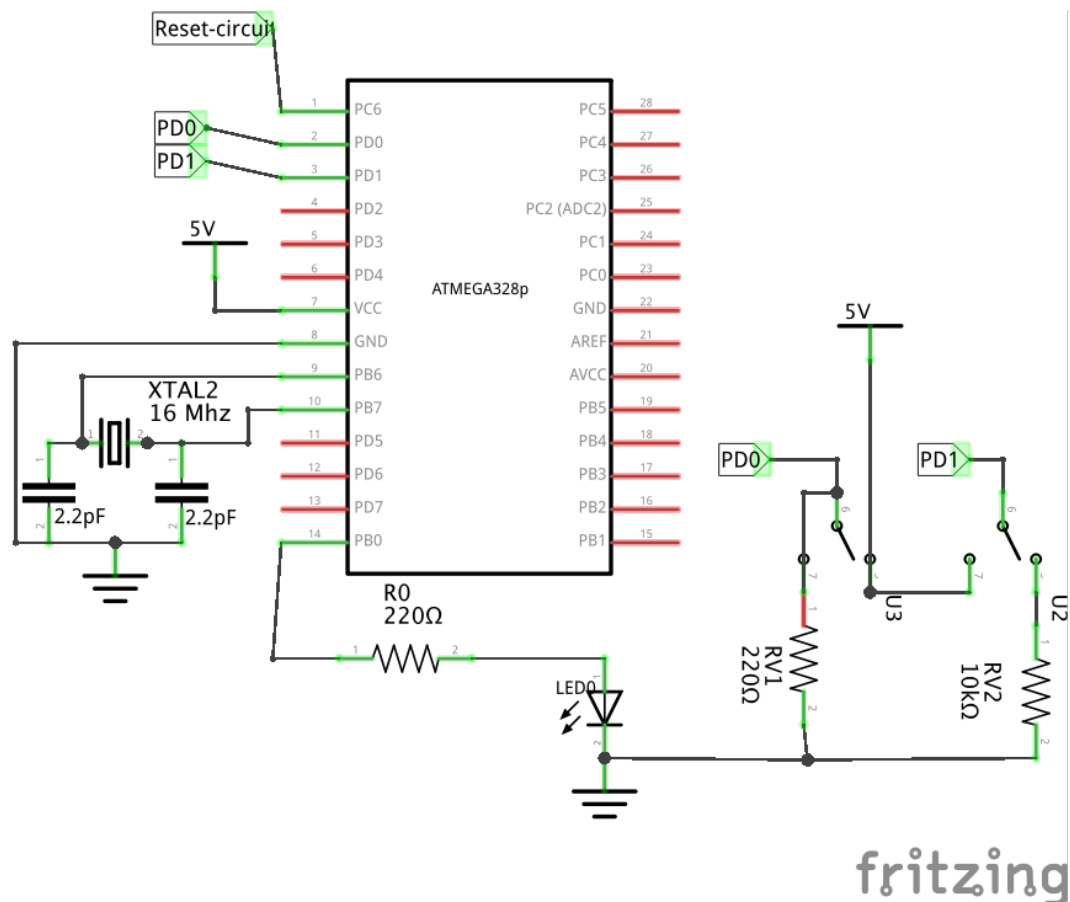


Figura 3: Esquemático del circuito implementado.

## 5. Diagrama de flujo

En la **Fig. 4** se muestra el diagrama de flujo del programa.

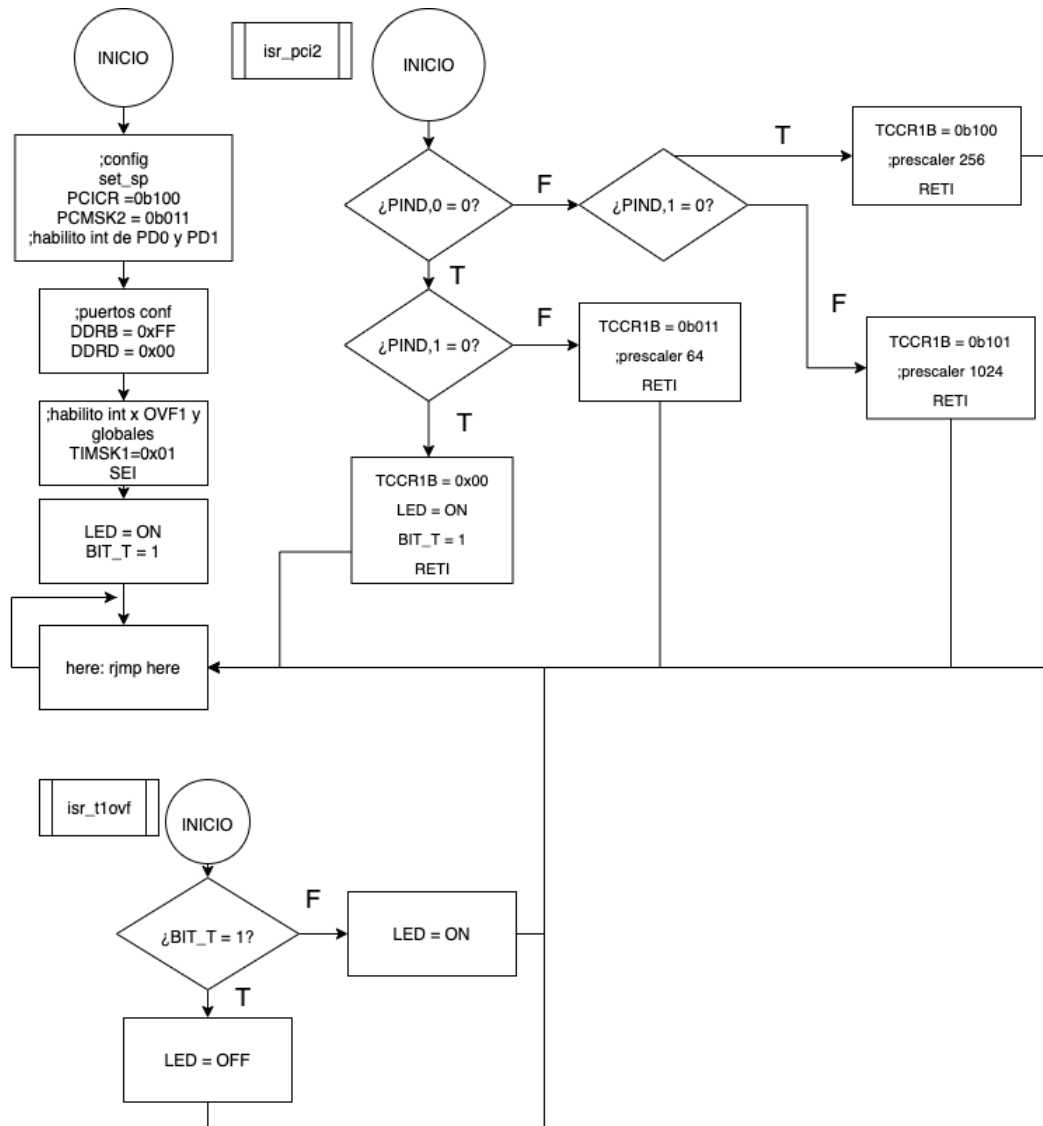


Figura 4: Diagrama de flujo.

## 6. Código

El siguiente código consta de tres partes, en primer lugar la configuración donde se inicializa el **stack pointer**, se configuran los puertos como entradas y salidas y los registros que afectan el comportamiento del **TIMER1**. Para setear el **TIMER** se habilitaron las interrupciones por overflow de este timer, y en diferentes condiciones se setean distintos valores del registro **TCCR1B** según el comportamiento deseado.

Cada overflow del **TIMER1** genera un cambio de valor lógico en el pin **PB0**, el cual enciende o apaga el **LED**.

Cada cambio de valor lógico en alguna de las entradas **PD0** o **PD1** generara una interrupción y se cargara el valor correspondiente según esas entradas al registro **TCCR1B**.

```
1 ; Autor: Francisco Rossi
2 ; Padron: 99540
3 ; 86.07 Laboratorio de Microprocesadores – FIUBA
4 ; Catedra: Miercoles
5 ; Fecha: 29 de julio de 2020
6 ; TP6 – Timers
7
8 .include "m328pdef.inc"
9
10 ; ETIQUETAS
11 .equ NOCLOCK = 0x00
12 .equ PRE64 = 0x03
13 .equ PRE256 = 0x04
14 .equ PRE1024 = 0x05
15
16 .def dummy = r25
17
18 ;MACROS
19 .macro set_sp
20     ldi dummy, low(RAMEND)
21     out spl, dummy
22     ldi dummy, high(RAMEND)
23     out sph, dummy
24 .endm
25
26
27 .macro set_port_as_out
28     ldi dummy, 0xFF
29     out @0, dummy
30 .endm
31
32 .macro set_port_as_in
33     ldi dummy, 0x00
34     out @0, dummy
35 .endm
36
37 .cseg
38 .org 0x0000
39     jmp config
40
41 ; PIN CHANGES
42 .org PCIE2addr
43     jmp isr_pci2
44 ; INT x OVF del timer 1
45 .org OVFladdr
46     jmp isr_t1lovf
47 .org INT.VECTORS_SIZE
48
49 config:
50     set_sp
51
52 ; pin change config para PD0 y PD1 (PCINT16 y PCIN17)
53     ldi dummy, (1<<PCIE2) ; PCIE2 enable pinchange
54     sts PCICR, dummy
```

```

55
56     ldi    dummy, (1<<1 | 1<<0)
57     sts    PCMSK2, dummy
58
59     ; config de puertos
60     set_port_as_in DDRD
61     set_port_as_out DDRB
62
63     ; config inicial del timer default habilito int x overflow y globales
64     ldi    dummy, 0x01
65     sts    TMSK1, dummy
66
67     sei
68
69 main:
70     ; led encendido
71     sbi    PORTB, 0
72     set
73
74 here:
75     jmp    here
76
77 isr_pci2:
78     sbis    PIND, 0
79     rjmp    low_es_cero
80
81 ; x1
82 low_es_uno:
83     sbis    PIND, 1
84     rjmp    high_es_cero
85 ; caso 11
86     ldi    dummy, PRE1024
87     sts    TCCR1B, dummy
88     reti
89
90 ; caso 01
91 high_es_cero:
92     ldi    dummy, PRE64
93     sts    TCCR1B, dummy
94     reti
95
96 low_es_cero:
97     sbis    PIND, 1
98     rjmp    input_es_cero
99 ; caso 10
100    ldi    dummy, PRE256
101    sts    TCCR1B, dummy
102    reti
103 ; caso 00
104 input_es_cero:
105    ldi    dummy, NOCLOCK
106    sts    TCCR1B, dummy
107    sbi    PORTB, 0
108    set
109    reti
110
111 ; RUTINA DE INT X OVF DEL TIMER 1
112 isr_tlovf:
113     brts    apagar_led
114
115 encender_led:
116     sbi    PORTB, 0
117     set
118     reti
119
120 apagar_led:
121     cbi    PORTB, 0
122     clt
123     reti

```

## 7. Resultados

Se logró diseñar un programa para el microcontrolador ATMEGA328p con el cual se utiliza el TIMER1 para hacer parpadear un LED a frecuencia determinada por los valores de los primeros dos pines del puerto D (PD0 y PD1).

## 8. Conclusiones

Se logró generar un programa que realicé la tarea de configuración del TIMER1 de 16bits y a partir del mismo hacer oscilar un LED conectado a PB0 a diferentes frecuencias.