



FACULTAD DE INGENIERIA

Universidad de Buenos Aires

Laboratorio de Microprocesadores - 86.07

Trabajo Práctico N°7: PWM

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

1. Introducción	2
1.1. Objetivo	2
1.2. Descripción	2
2. Materiales	2
3. Diagrama en Bloques	2
4. Esquemático	3
5. Diagrama de flujo	4
6. Código	5
6.1. TCCR0A	5
6.2. TCCR0B	5
7. Resultados	7
8. Conclusiones	7

1. Introducción

En el siguiente informe se explica el diseño de un programa escrito en lenguaje Assembler el cual controla la intensidad lumínica de un LED (conectado a OC0A (PD6)) a partir de dos pulsadores (UP, DOWN).

1.1. Objetivo

El objetivo es utilizar el **TIMER0** de 8 bits del arduino para controlar el duty cycle de la señal de salida **OC0A** y así generar a partir de este **PWM** (modulación de ancho de pulso) sin modificar la frecuencia para modificar el valor medio de la señal de salida y así el brillo del LED.

1.2. Descripción

Se conectan dos switches, uno a **PD2 (INT0)** como *UP* y **PD3 (INT1)** como *DOWN*. A partir de los cuales se aumentará o disminuirá el brillo del LED respectivamente.

2. Materiales

Se utilizaron los siguiente materiales para el proyecto:

- 1 LED (20\$ (Pesos Argentinos))
- 1 Resistor de $220\ \Omega$ y 2 resistores 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.

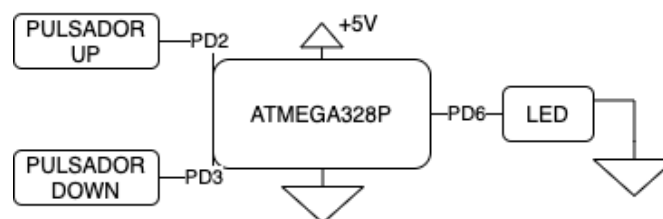


Figura 1: Diagrama en bloques.

4. Esquemático

En las **Fig. 3** y **Fig. 2** se muestra como se conectó el arduino con el LED, y los switches.

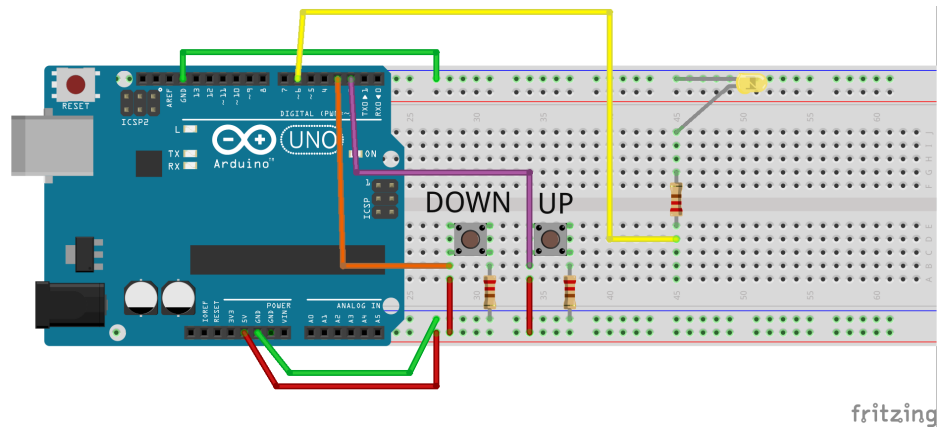


Figura 2: Conexión entre el arduino, los Switches y el LED

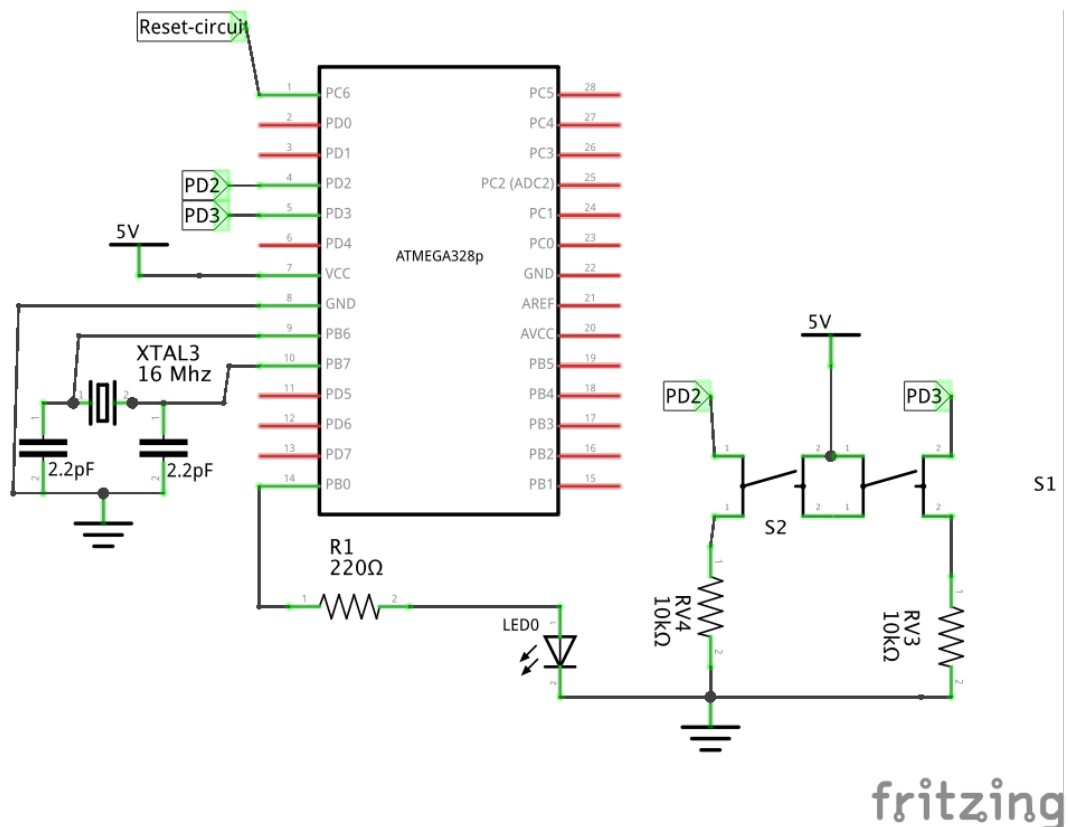


Figura 3: Esquemático del circuito implementado.

5. Diagrama de flujo

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

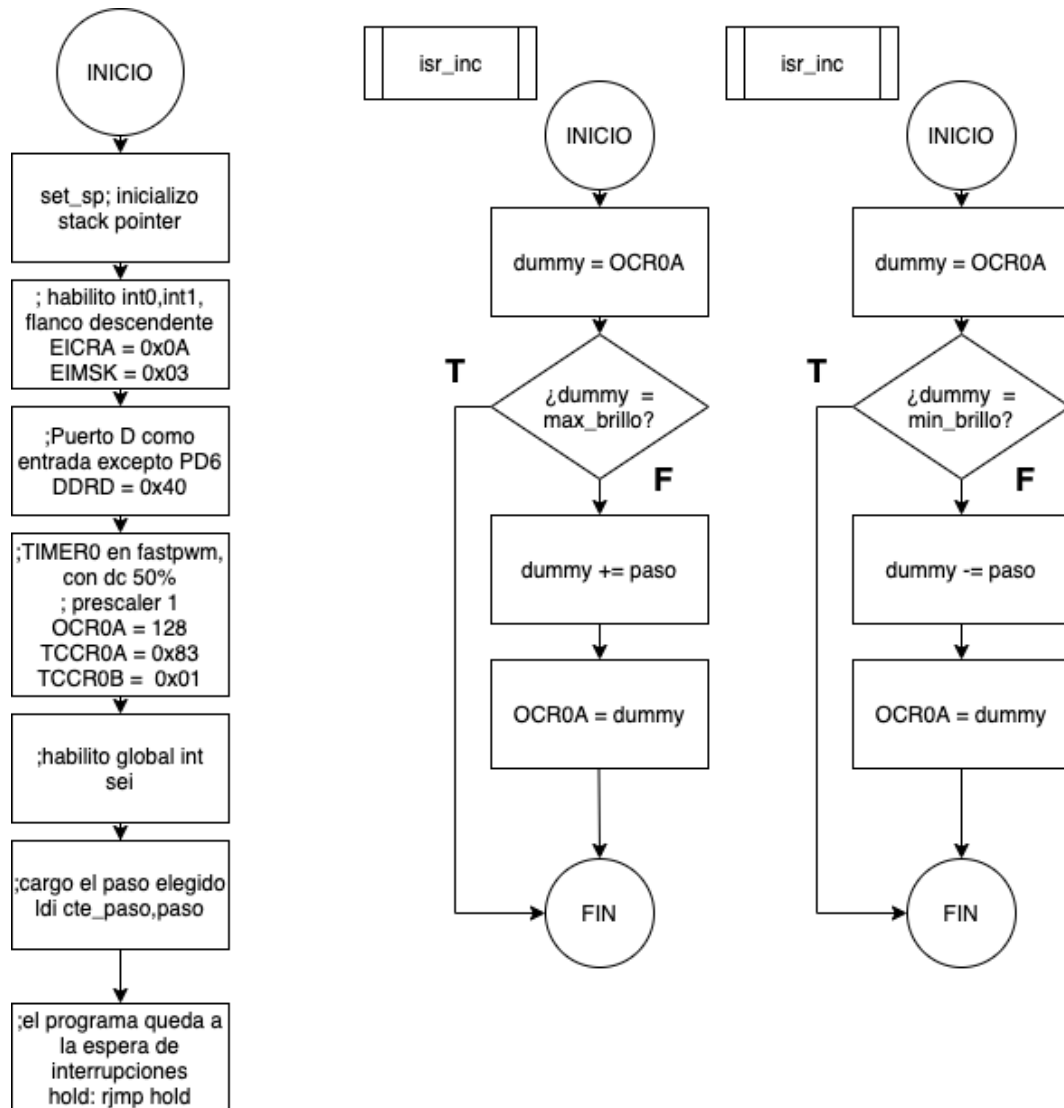


Figura 4: Diagrama de flujo.

6. Código

El siguiente código da la posibilidad de elegir la cantidad de pasos entre encendido y apagado del LED, cambiando el valor de la etiqueta `cant_pasos` entre valores potencias de dos hasta 256.

En cada interrupción se modifica el valor de `OCR0A` el cual determina el duty cycle de la señal de salida, de manera que determina la intensidad de brillo del LED como el PWM esta en modo no inversor a mayor `OCR0A` mayor duty cycle y mayor brillo.

Se elige un valor máximo de brillo de manera que la salida llegue hasta ese valor y si se trata de aumentar el brillo ya estando en este valor no habrá modificaciones, de igual manera un brillo mínimo.

La frecuencia de la cuadrada se eligio igual que el clock default del arduino: $f_{ck} = 16 \text{ MHz}$.

Los registros se configuran de la siguiente forma:

6.1. TCCR0A

En la **Tabla 1** se muestra la configuración del registro TCCR0A.

Tabla 1: Configuración del registro TCCR0A.

COM0A1	COM0A0	COM0B1	COM0B0	-	-	WGM01	WGM00
1	0	0	0	x	x	1	1

Donde se configura para que `OC0A` se ponga en 0 cuando el valor del contador alcanza el valor de `OCR0A`, y `WGM01`, `WGM00` son parte de la configuración para que funcione en Fast PWM.

6.2. TCCR0B

En la **Tabla 2** se muestra la configuración del registro TCCR0B.

Tabla 2: Configuración del registro TCCR0B.

FOC0A	FOC0B	-	-	WGM02	CS02	CS01	CS00
0	0	x	0	0	x	0	1

De donde se configura el prescaler en 1 y se adopta `WGM02` para el tipo de funcionamiento del timer.

```
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
12 .equ cant_pasos = 16 ;(Solo usar potencias de dos hasta 256)
13 .equ paso = (256/cant_pasos)
14 .equ brillo_minimo = 0x00
15 .equ brillo_maximo = (256 - paso)
16
17 .def cte_paso = r24
18 .def dummy = r25
19
20 ;MACROS
21 .macro set_sp
22     ldi dummy, low(RAMEND)
23     out spl, dummy
24     ldi dummy, high(RAMEND)
25     out sph, dummy
26 .endm
27
28
29 .cseg
30 .org 0x0000
31     jmp config
32
33 ; Interrupciones
34 .org INT0addr
35     jmp isr_inc
36 .org INT1addr
37     jmp isr_dec
38
39 .org INT.VECTORS_SIZE
40
41 config:
42     set_sp
43
44     ; config de interrupciones HABILITO INT0 INT1 POR FLANCO DESCENDENTE
45     ldi dummy, (1 << 3 | 1 << 1)
46     sts EICRA, dummy
47
48     ldi dummy, (1 << 1 | 1 << 0)
49     out EIMSK, dummy
50
51     ; config de puertos pd6 como salida y pd2 y pd3 como entrada
52     ldi dummy, (1 << 6); pd6 como salida y demas entrada
53     out DDRD, dummy
54
55     ; config inicial del timer0
56     ldi dummy, 128
57     out OCR0A, dummy ; inicio a aprox 50%
58     ldi dummy, (1 << 7| 1 << 1 | 1 << 0) ;toggle OCR0A +fastpwm mode3
59     out TCCR0A, dummy
60     ldi dummy, (1 << 0) ;no prescaler , max freq y wgm2=0
61     out TCCR0B, dummy
62
63     ; habilito interrupciones globales
64     sei
65
66 main:
67     ldi cte_paso, paso
68
69 hold:
```

```
70  rjmp  hold
71
72  isr_inc :
73      in    dummy, OCR0A
74
75      cpi    dummy, brillo_maximo
76      breq   max_brillo
77
78      add    dummy, cte_paso
79      out    OCR0A, dummy
80
81  max_brillo :
82      reti
83
84  isr_dec :
85      in    dummy, OCR0A
86      cpi    dummy, brillo_minimo
87      breq   min_brillo
88
89      sub    dummy, cte_paso
90      out    OCR0A, dummy
91
92  min_brillo :
93      reti
```

7. Resultados

Se logró diseñar un programa para el microcontrolador ATMEGA328p con el cual se utiliza el TIMER0 en modo **fast PWM** para controlar el brillo de un LED a partir de dos pulsadores conectados en los pines PD2 y PD3.

8. Conclusiones

Se logró generar un programa que realicé la tarea de configuración del TIMER0 de 8 bits y a partir del mismo controlar el brillo de un LED con el modo **PWM** de una señal cuadrada de 16 *MHz*. Fue posible configurar la cantidad de pasos entre encendido y apagado del mismo de manera que depende el uso que se necesite poder tener mas o menos pasos.