



Laboratorio de Microprocesadores - 86.07

Trabajo Práctico Obligatorio 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:			Ing. Fabricio Baglivo, Ing. Fernando Pucci									
Autores			Seguimiento del proyecto									
Nombre	Apellido	Padrón										
Santiago	López	100566										

Observaciones:

.....

.....

.....

.....

.....

.....

.....

.....

.....

Fecha de aprobación				Firma J.T.P

Coloquio	
Nota final	
Firma profesor	

Índice

1. Objetivo	2
2. Descripción	2
3. Diagrama de conexiones en bloque	2
4. Esquemático	2
5. Listado de componentes	3
6. Diagrama de flujo	3
7. Código fuente	5
8. Costos	8
9. Resultados	8
10. Conclusiones	8

1. Objetivo

Controlar la intensidad de un LED mediante la modulación por ancho de pulso (PWM) de una señal rectangular.

2. Descripción

El LED recibe una tensión de acuerdo al valor medio de una señal rectangular. A esta señal se le varirá el *duty cycle* a medida que se reciban pulsos externos para que el valor medio de la señal tanto suba como baje de acuerdo al pin que reciba el pulso.

Para generar el tren de pulsos que se conecta al LED se inicializa un valor de referencia entre 0 y 255, el cual se incrementa o decrementa de acuerdo a dos interrupciones externas.

El micro posee un registro el cual cuenta de forma ascendente de acuerdo a la frecuencia del clock¹, dividida por el prescale seteado. En cuanto el valor de este registro sea igual al del valor de referencia, el estado de la señal de salida cambia de 0 a 1, y luego volverá a cambiar de 1 a 0 cuando el registro pase del valor 0xFF a 0x00.

3. Diagrama de conexiones en bloque

Las conexiones siguieron el esquema de la Figura 1.

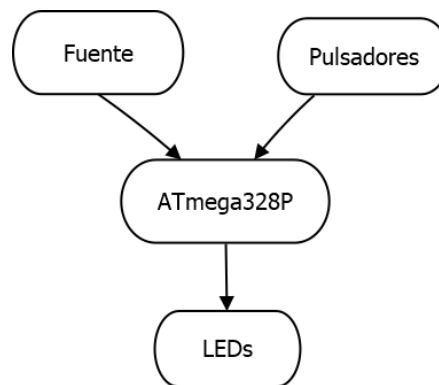


Figura 1: Diagrama de conexiones en bloque.

4. Esquemático

La Figura 2 muestra las conexiones eléctricas efectuadas en el práctico. Los componentes utilizados se encuentran en la sección siguiente. En este caso se decidió utilizar las resistencias de *pull-up* internas del micro y la salida en el pin PB3 por haberse utilizado el timer 2.

¹En este caso cuenta con un clock externo de 16MHz.

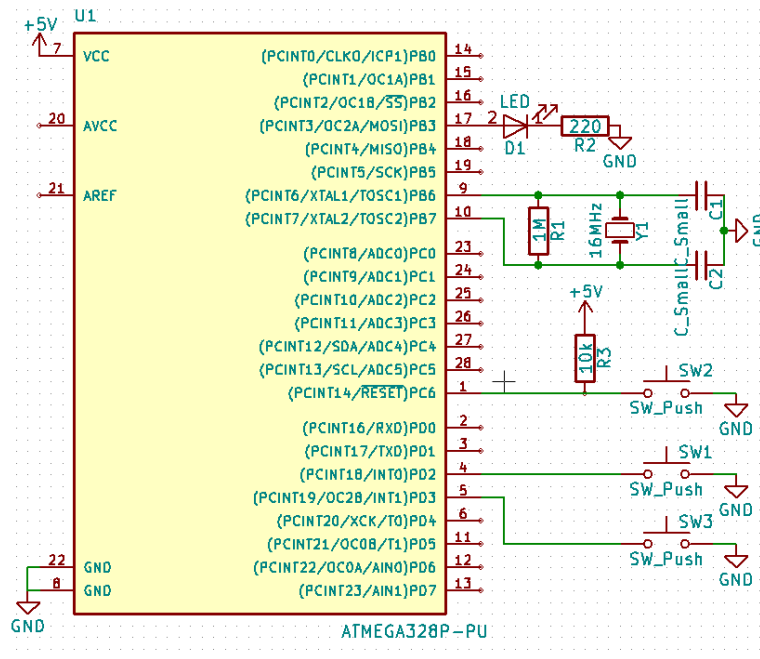


Figura 2: Circuito esquemático.

5. Listado de componentes

Los componentes utilizados fueron los listados a continuación:

- Placa de desarrollo Arduino UNO
- Resistores de 220Ω y 10kΩ
- LED de color rojo
- Pulsadores
- Protoboard
- Cables unipolares

6. Diagrama de flujo

En la Figura 3 se presentan los pasos a seguir por el programa. La primera rutina configura los puertos DDRB y DDRD para que el PWM pueda salir por el pin PB3 y los resistores de pull-up estén habilitados, luego llama a las rutinas de configuración del timer y de las interrupciones. La segunda rutina es la configuración del timer. Se encarga de setear los puertos del timer para el uso a lo largo de todo el programa. La tercer rutina habilita las interrupciones externas por flanco descendente.

En la Figura 4 se presentan las interrupciones del programa. La primera incrementa el valor medio de la tensión de salida, y la segunda lo decrementa, si es posible.

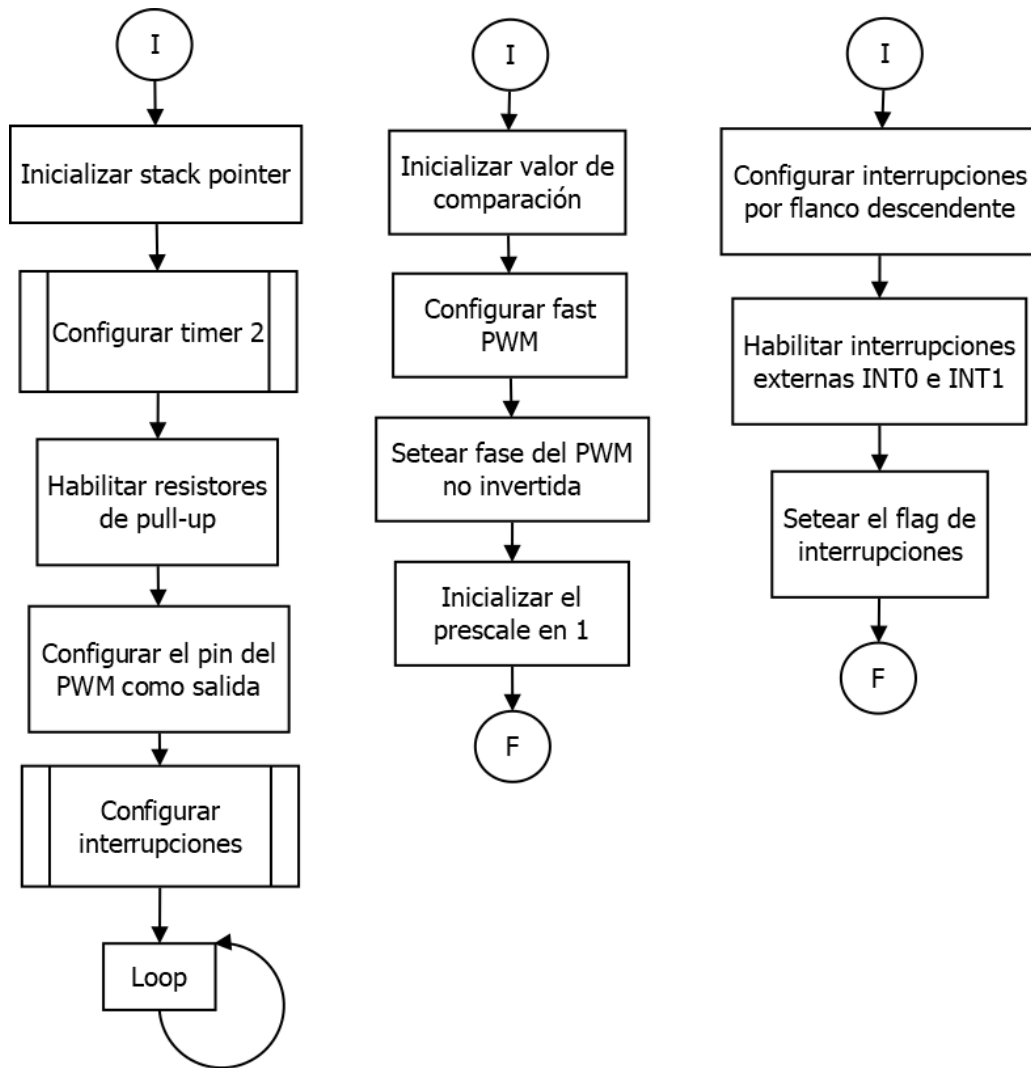


Figura 3: Diagrama de flujo.

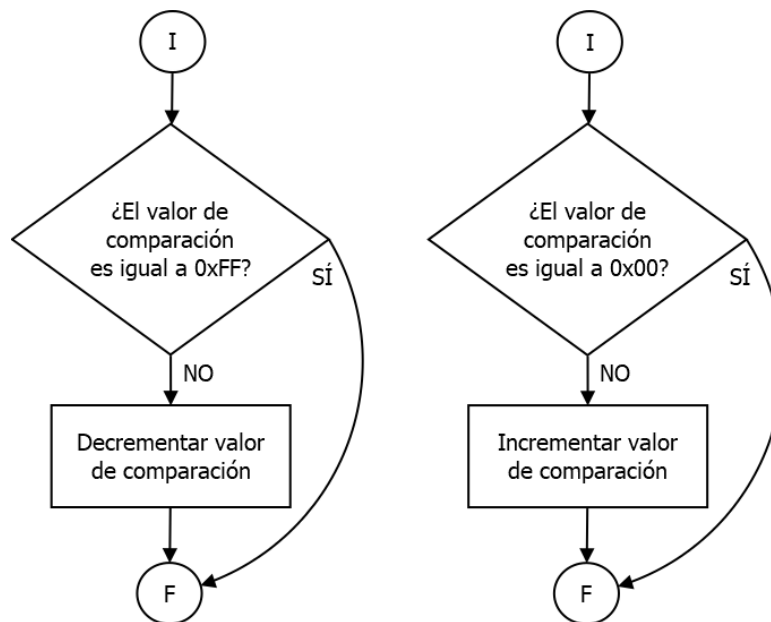


Figura 4: Diagrama de flujo - Interrupciones.

7. Código fuente

El programa consiste en una serie de rutinas de setup para luego esperar a las interrupciones provocadas al accionar los pulsadores conectados a los pines PD2 y PD3.

En las interrupciones se modifica el valor del registro OCR2A, el cual se compara constantemente contra el registro TCCR2 para así formar el PWM sobre el pin OC2A, el cual se visibiliza en el LED conectado al pin PB3.

En un comienzo se implementó el programa con el uso del timer 0, pero este trajo complicaciones las cuales no pudieron ser resueltas, por lo que se optó por el cambio al timer 2, con el cual los problemas no formaron parte del programa.

Se eligieron estos timers para trabajar con registros de 8 bits, ya que resultó suficiente para abarcar el uso del PWM.

```

1: .include "m328pdef.inc"
2:
3: .dseg
4:
5: .def conf = r16
6: .def intensity = r17
7: .equ vmed = OCR2A
8: .equ vinit = 0x0f
9: .equ led_port = ddrb
10: .equ led = 3
11:
12: .macro init_sp
13:     ldi conf, low (RAMEND)
14:     out spl, conf
15:     ldi conf, high(RAMEND)
16:     out sph, conf
17: .endmacro
18:
19: .macro enable_pull_up_resistor
20:     ldi conf, 0x0c ; pins pd2 y pd3 en salida
21:     out DDRD, conf
22:     ldi conf, 0x0c
23:     out PIND, conf
24: .endmacro
25:
26: .cseg
27:
28: .org 0x0000
29:     jmp main
30: .org INT0addr
31:     jmp isr_int0
32: .org INT1addr
33:     jmp isr_int1
34:
35: .org INT_VECTORS_SIZE
36: main:
37:
38:     init_sp
39:     call setup_timer
40:     enable_pull_up_resistor
41:     sbi led_port, led
42:     call setup_interruption
43:
44: loop:
45:     nop
46:     nop
47:     nop
48:     jmp loop
49:
50:
51: // subrutinas
52:
53: setup_timer:
54:     ldi intensity, vinit
55:     sts vmed, intensity
56: // wmg 011 -> modo de op fast PWM, tope en MAX
57: // com2A 10 -> clr OC2a en cmp, set OC2a en 0x00
58: // com2B 00 -> OC2b desconectado
59: // cs2 001 -> ck = 16MHz / 1
60:     ldi conf, 0x83
61:     sts TCCR2A, conf
62:     ldi conf, 0x01
63:     sts TCCR2B, conf
64: ret
65:
66: setup_interruption:
67:     ldi conf, 0x0a ; flanco descendente para int0 e int1
68:     sts eicra, conf
69:     ldi conf, 0x03 ; habilito interrupciones 0 y 1

```

```
70:    out eimsk, conf
71:    sei
72: ret
73:
74:
75: // interrupciones
76:
77: isr_int0:
78:    cpi intensity, 0xff
79:    breq end0
80:    inc intensity
81:    sts vmed, intensity
82: end0:
83: reti
84:
85: isr_int1:
86:    cpi intensity, 0x00
87:    breq end1
88:    dec intensity
89:    sts vmed, intensity
90: end1:
91: reti
```


8. Costos

A continuación se presenta un listado de los costos de los componentes utilizados en el práctico.

- Arduino UNO - \$850
- Resistores - \$50
- LED rojo - \$20
- Pulsadores - \$50
- Protoboard - \$240
- Cables unipolares - \$150

Sumando un costo total de \$1360.

9. Resultados

La corriente máxima que puede circular por un pin es de 20mA. Al estar en un estado alto constante se tienen 5V entregados por el micro, menos 1,5V que caen sobre el LED, dividido por los 220Ω del resistor, quedan unos 16mA, como máximo, circulando.

10. Conclusiones

El uso del PWM permite variar la tensión entregada por un pin, lo que es de utilidad si se requiere el uso de varios pines de un puerto, sin entregar mucha corriente, ya que permite entregar baja corriente si se setea un *duty cycle* bajo.

Sin embargo, el micro no cuenta con una gran cantidad de salidas con PWM, por lo que su uso debe ser utilizado a consciencia.