



Laboratorio de Microprocesadores - 86.07

Trabajo Práctico Obligatorio N°1

Profesor:			Ing. Guillermo Campiglio									
Cuatrimestre/Año:			1º/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									
Mariano Federico	Guglieri	99573										

Observaciones:

.....

.....

.....

.....

.....

.....

.....

.....

Fecha de aprobación			Firma J.T.P		

Coloquio	
Nota final	
Firma profesor	

Índice

1. Objetivos	2
2. Desarrollo	2
3. Diagrama en bloques	2
4. Esquemático y listado de componentes	3
5. Diagrama de flujos	4
6. Códigos	5
7. Resultados	5
8. Conclusiones	6

1. Objetivos

El objetivo de este trabajo consiste en implementar un programa que aumente y disminuya el brillo de un LED, utilizando el PWM, o modulación por ancho de pulso, del atmega328p. La intensidad del brillo será controlada mediante dos pulsadores, uno que aumente la intensidad y otro que la disminuya.

2. Desarrollo

Para la realización del trabajo se utilizó la plataforma de desarrollo de Atmel, Atmel Studio versión 7.0, en donde se implementó el Software. Para la parte física se requirió de un Arduino UNO, el cual sirvió como programador para el integrado ATMEGA328P, además de suplirlo con la energía necesaria para su funcionamiento. Se utilizó un resistor junto a un led de color verde conectado al puerto b, dos pulsadores conectados al pin de interrupción, cada uno con un resistor y varios cables para unir el circuito.

En la primera parte del código se declararon las direcciones de la sección de configuración, de las interrupciones y del programa principal. En la parte de configuración se configuró el Stack Pointer, los pines de entrada y salida, las interrupciones y el modo de trabajo del timer1. El programa principal sólo se encarga de limpiar el registro R16 para su uso posterior en las interrupciones.

Las interrupciones se configuraron para actuar por flanco descendente, lo que posibilita poder ver un aumento paulatino en la intensidad del brillo del led. Se usaron las interrupciones INT0 e INT1, las cuales corresponden al PIND2 y PIND3 respectivamente. Lo que hacen las interrupciones es aumentar o decrementar el valor del registro de comparación del timer1.

Para generar un pulso de ancho modulado utilizando el timer, se habilita el pin de OC1A. Este pin corresponde al PIN 1 del puerto B. Cuando se activa el modo OC1A el PINB1 se comporta como la salida de una señal cuadrada. Existen varios modos de PWM en el atmega328p. Para este trabajo se utilizó el modo fast-PMW de 8 bits sin ningún prescaler. Este era el modo más sencillo. El timer1 en este modo cuenta de 0 a 255 a la velocidad de frecuencia del clock. Una vez que llega a su valor máximo, comienza a contar desde 0. Cuando el registro de conteo (TCNT1L) esté por debajo del registro de comparación (OCR1AL), la señal estará en estado alto. Cuando el valor del registro de conteo sea superior, la señal de pulso cuadrado estará en estado bajo. De este modo se controla el ancho de pulso de la señal.

3. Diagrama en bloques

El diagrama de conexiones en bloque del trabajo se presenta a continuación:

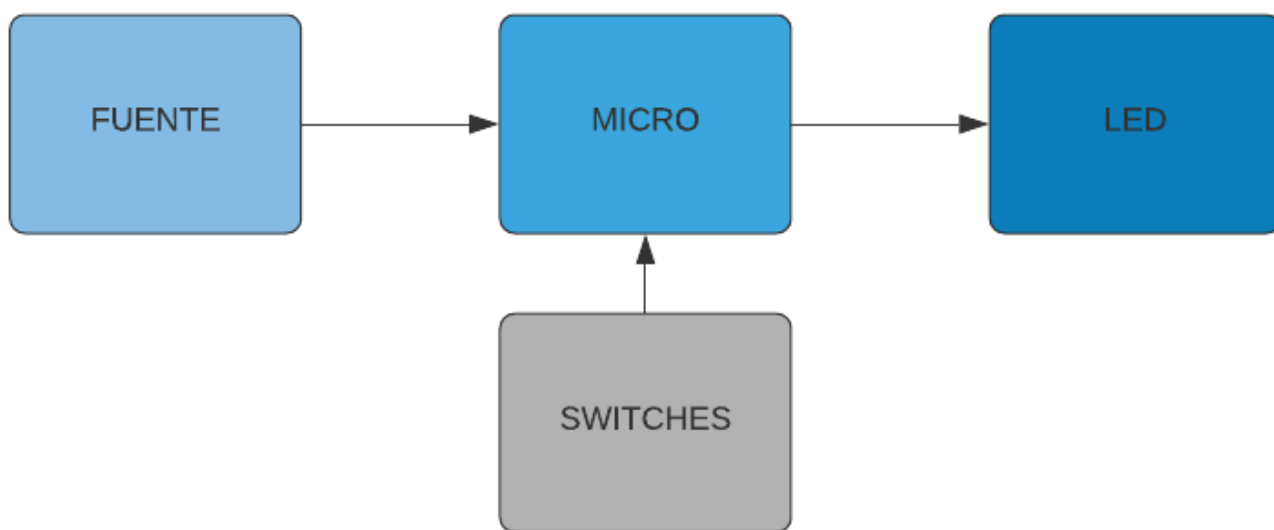


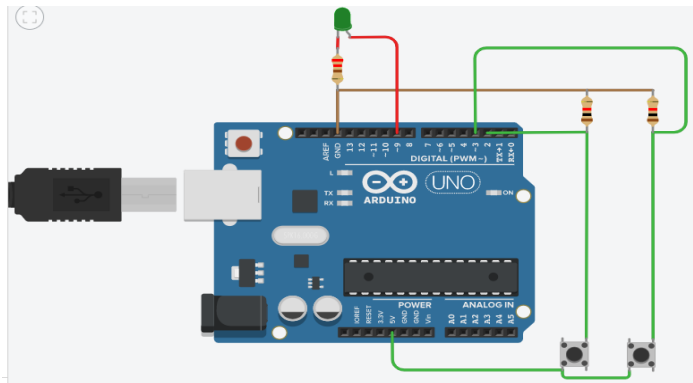
Figura 1: Diagrama de conexiones en bloque

Se observa que el funcionamiento surge primero de la pc donde se encuentra el código que será descargado en el micro y a su vez sirve como una fuente. El micro recibe información del estado de los botones. En base a si se pulsó uno o no, el micro ejecutará la rutina de interrupción correspondiente, aumentando o disminuyendo el brillo del LED. Esto es lo que se ve en el diagrama.

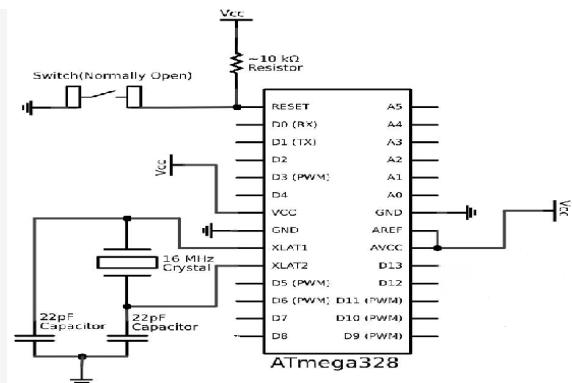
4. Esquemático y listado de componentes

Para el circuito se utilizaron los siguientes componentes:

- 2 resistores de $10K\Omega$
- 1 resistor de 220Ω
- 1 led de color verde
- 2 pulsadores
- 7 cables para protoboard
- 1 placa arduino con atmega328p
- 1 protoboard



(a) Esquemático del circuito implementado



(b) Circuito con cristal de 16MHz

Figura 2: Esquemático del circuito

En total se habrá gastado unos 150 pesos para el circuito. Sin contar la plataforma arduino.

La plataforma arduino le agrega al microcontrolador un cristal de 16Mhz y una resistencia de $10k\Omega$, como se observa en a figura 2b.

5. Diagrama de flujos

A continuación se presentan los diagramas de flujo. El primero consiste en el programa principal más la configuración previa. La configuración de puertos y de la interrupción se puso todo en un mismo bloque ya que esto ya se vió en los trabajos anteriores. Las rutinas de interrupción se encuentran al lado. La lógica es la misma para ambas, primero se verifica si el registro llegó a su valor límite. De ser así no se modifica el registro de comparación del timer1.

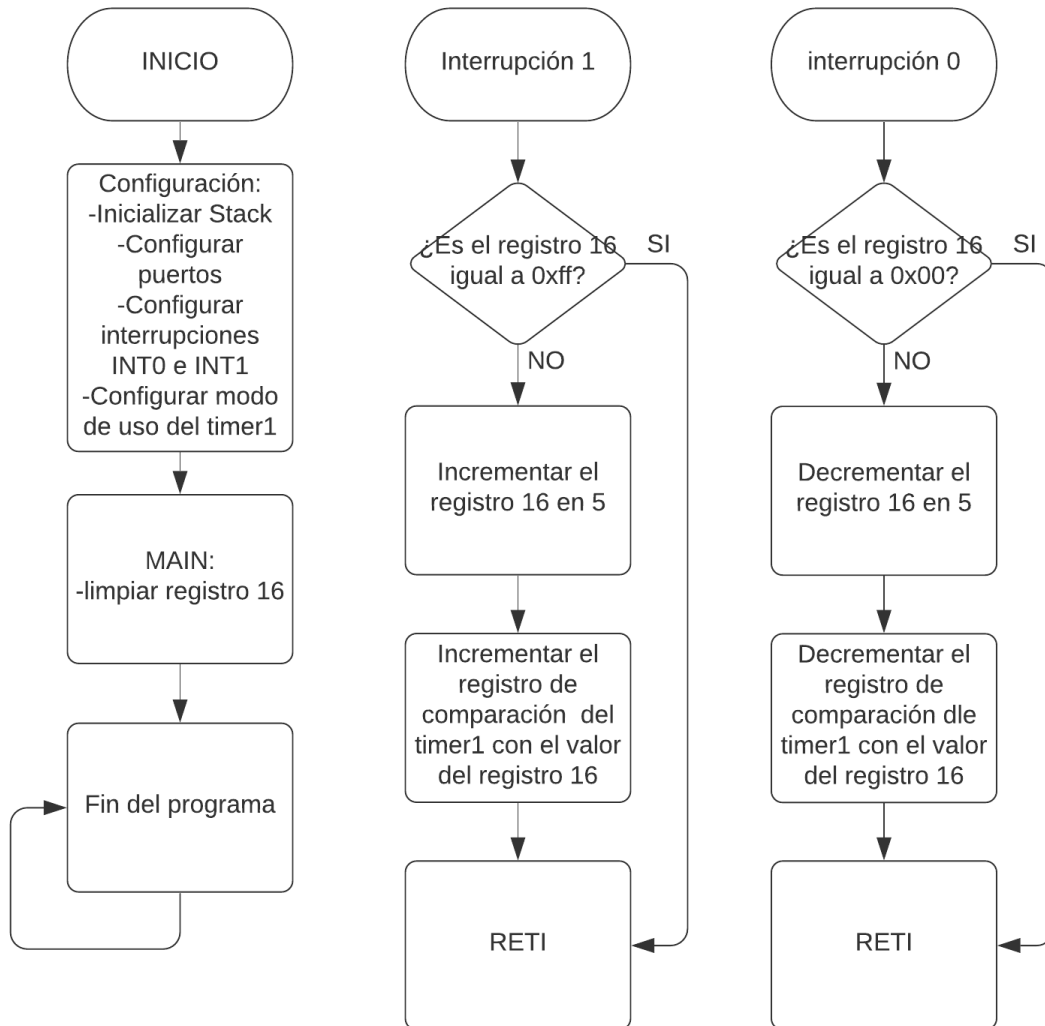


Figura 3: Diagrama de flujos

6. Códigos

A continuación se presenta el código:

```

1  .include "m328pdef.inc"
2
3  .cseg
4  ;direcciones de codigo
5  .org 0x0000
6  jmp     configuracion
7  .org INT0addr
8  jmp     interrupcion_0
9  .org INT1addr
10 jmp     interrupcion_1
11 configuracion:
12
13 ;inicializo el stack pointer
14 ldi     R16, HIGH(RAMEND)
15 out     SPH, R16
16 ldi     R16, LOW(RAMEND)
17 out     SPL, R16
18
19 ;declaro el PINB1 como salida
20 sbi     DDRB,1
21
22 ;declaro el PORTD como entrada
23 ldi     R16, 0xff
24 out     DDRD, R16
25
26 ;configuro el timer1 para que trabaje sin prescaler, en modo fast PWM de 8bits y sobre el pin OC1A
27 ldi     R16, ( 1<<CS10 | 1<<WGM12)
28 sts     TCCR1B, R16
29 ldi     R16, ( 1<<COM1A1 | 1<<WGM10)
30 sts     TCCR1A, R16
31
32 ;configuro la interrupcion 0 y 1 por flanco ascendente
33 ldi     R16,(1 << ISC11 | 0 << ISC10 | 1 << ISC01 | 0 << ISC00 )
34 sts     EICRA,R16
35 ldi     R16,(1 << INTO | 1 <<INT1)
36 out     EIMSK, R16
37
38 sei
39
40 main:
41     clc     R16     ;limpio el registro R16
42 fin:  rjmp  fin
43
44 interrupcion_1:                                ;si el registro es igual a 0xff, llego a su limite y no
45     se sigue incrementando
46     cpi     R16, 0xff
47     breq    nomas1
48     ldi     R17, 5
49     add     R16, R17
50     sts     OCR1AL, R16     ;el contenido del registro de comparacin se aumenta en
51     5, aumentando el ciclo de trabajo del pulso de la onda
52 nomas1:
53     reti
54
55 interrupcion_0:
56     cpi     R16, 0                                ;si el registro es igual a 0, llego a su limite y no se
57     sigue decrementando
58     breq    nomenos1
59     subi    R16, 5
60     sts     OCR1AL, R16     ;el contenido del registro de comparacin se disminuye
61     en 5, disminuyendo el ciclo de trabajo del pulso de la onda
62 nomenos1:
63     reti

```

7. Resultados

Los resultados fueron los esperados. No hubo mayor inconveniente en la implementación del problema.

8. Conclusiones

Este trabajo permite visualizar un importante atributo del microcontrolador, que es el control de la cantidad de energía hacia una fuente. Con esta técnica de modulación por ancho de pulso se puede controlar la intensidad de muchos parámetros que requieran del abastecimiento de corriente para su funcionamiento, como puede ser la velocidad de un motor eléctrico.

También, se observan las funciones secundarias de cada puerto, como las entradas de interrupción o los pines que habilitan las salidas del PWM. Esto genera una mayor comprensión de la estructura del micro.