Laboratorio de Microprocesadores - 86.07

Trabajo Práctico Obligatorio N°7

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									
Ivan Eric	Rubin	100 577									

Observaciones:

.....
.....
.....
.....
.....
.....
.....
.....
.....

Fecha de aprobación				Firma J.T.P

Coloquio	
Nota final	
Firma profesor	

1. Objetivo del trabajo

El objetivo de este trabajo es controlar la intensidad de brillo de un LED utilizando dos pulsadores que modifican el ciclo de trabajo de la señal utilizada por este. Se utilizara la modulación por ancho de pulso del ATMega328p.

2. Descripción del trabajo

El trabajo consiste en conectar un LED al puerto B del microprocesador ATMega328p y dos botones con configuración pull-down al puerto D. La idea es que con un pulsador aumente el brillo del LED y con el otro disminuya. Se utilizará el modo PWM (Modulación por ancho de pulso) del timer que posee el microprocesador. Consiste en modificar el ancho de la señal de pulsos que llega al LED aumentando la energía que este utiliza, por ende aumentando la intensidad del brillo. Todo esto sin cambiar la frecuencia de la señal.

3. Diagrama de conexiones en bloques

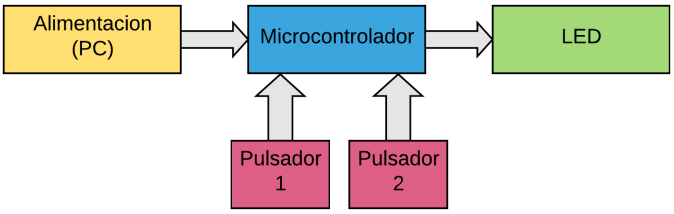

Figura 3.1: Diagrama de bloques.

4. Circuito Esquemático

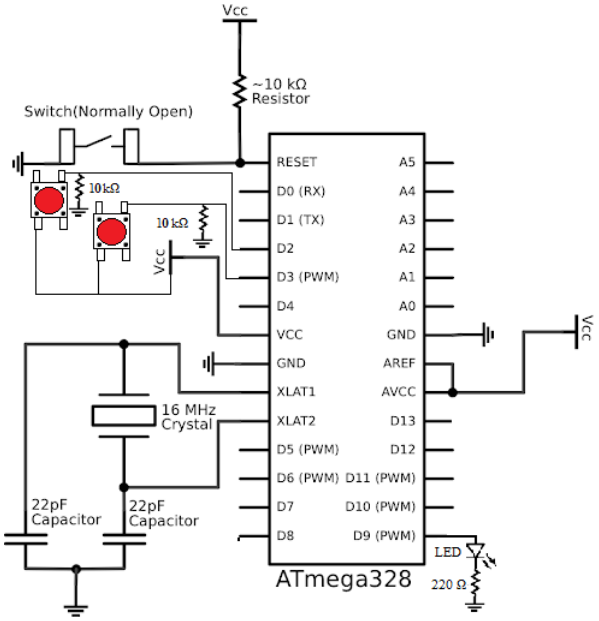

Figura 4.1: Circuito Esquemático.

5. Listado de componentes

- Arduino UNO + Conector PC (864$).
- Microprocesador ATMEGA 328P (Integrado en Arduino).
- Resistencia de $220\Omega \pm 1\,\%$ (5$).
- Diodo Emisor de Luz (LED) (1 x 10$).
- Resistencia de $10\text{k}\Omega \pm 1\,\%$ (5$).
- Pulsadores (2 x 40$).
- Cables Macho-Macho (40 cables x 100$).
- Protoboard (200$).

6. Diagrama de flujo del software

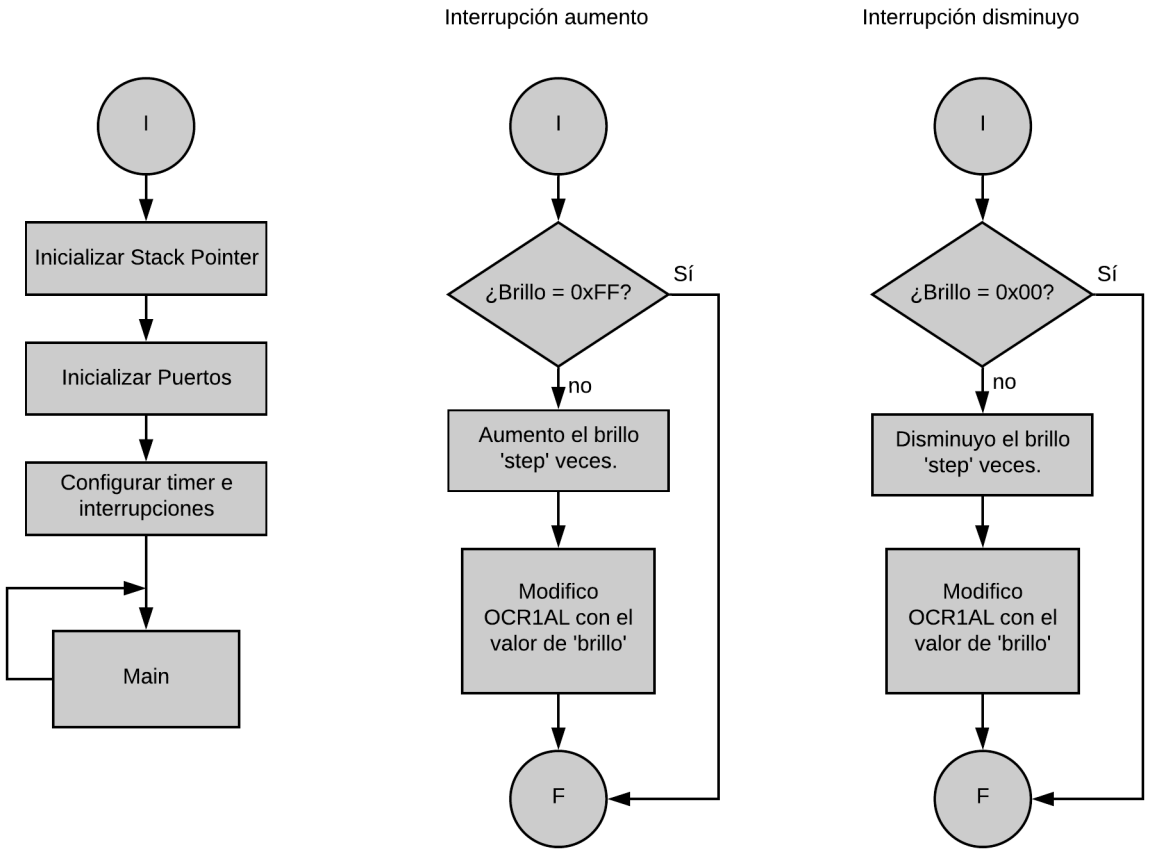

Figura 6.1: Diagrama de flujo.

7. Código de programa

```
/*
 *
   Trabajo Práctico Obligatorio 7
   Autor: Ivan Eric Rubin
 *
 */
.include "m328pdef.inc"

.def dummyr=R16
.def brillo=R17

.equ step=51    ; con 51 tengo 5 niveles de brillo (5x51=255)

.cseg

.org 0x0000
   rjmp  config

.org INT0addr
   rjmp  disminuyo
.org INT1addr
   rjmp  aumento

.org INT_VECTORS_SIZE

config:
   ; Inicializo el Stack Pointer.
   ldi       dummyr, HIGH(RAMEND)
   out       SPH, dummyr
   ldi       dummyr, LOW(RAMEND)
   out       SPL, dummyr

   ; Puerto B como salida.
   ldi       dummyr, 0xFF
   out       DDRB, dummyr

   ;Puerto D como entrada.
   ldi       dummyr, 0x00
   out       DDRD, dummyr

   ; Configuro timer en el modo PWM rápido (8 bits).
   ldi       dummyr, ( 1<<CS10 | 1<<WGM12)
   sts       TCCR1B, dummyr
   ldi       dummyr, ( 1<<COM1A1 |1<<WGM10)
   sts       TCCR1A, dummyr

   ; Configuro las interrupciones por flanco ascendente.
   ldi       dummyr,(1 << ISC11 | 0 << ISC10 | 1 << ISC01 | 0 << ISC00  )
   sts       EICRA,dummyr
   ldi       dummyr,(1 << INT0 | 1 <<INT1)
   out       EIMSK, dummyr

   ; Habilito  las  interrupciones  globales.
   sei

   ; Inicializo el brillo en 0.
   ldi       brillo, 0x00

main:
```

```
60      rjmp      main
61
62  ; Rutina de interrupcion de aumento de PWM.
63  aumento:
64      cpi       brillo, 0xFF
65      breq      limite            ; Si se llega al límite superior, termina.
66      ldi       dummyr, step
67      add       brillo, dummyr    ; Aumenta el brillo 'step' veces.
68      sts       OCR1AL, brillo    ; Modifico el ancho de pulso.
69      reti
70
71  ; Rutina de interrupción de disminución de PWM.
72  disminuyo:
73      cpi       brillo, 0x00
74      breq      limite            ; Si se llega al límite inferior, termina.
75      subi      brillo, step      ; Disminuye el brillo 'step' veces.
76      sts       OCR1AL, brillo    ; Modifico el ancho de pulso.
77      reti
78
79  limite:
80      reti
```

8. Resultados

Se pudo lograr todo lo pedido en el trabajo. Se diseño un codigo con el que se puede tener distintas cantidades de niveles de brillo. Esto depende de la variable step que debe ser un valor divisor de 255 ya que el valor del brillo se define con un registro de 8 bits que modifica el ancho del pulso de la señal que recibe el LED.

9. Conclusiones

Como conclusión se destaca la importancia de lo aprendido en este trabajo con respecto al manejo de la energia utilizada. Se pueden diseñar sistemas en los cuales no haya malgasto de energía controlando el ancho del pulso tal como se hizo en este trabajo. También se profundizo el conocimiento de las interrupciones y las distintas configuraciones que posee el timer del ATMega328p.