



(6609) LABORATORIO DE MICROCOMPUTADORAS

Proyecto:  
*TP1 - Manejo de puertos*

<b>Profesor:</b>	<b>Ing. Guillermo Campiglio</b>
<b>Cuatrimestre / Año:</b>	1er cuatrimestre - 2022
<b>Turno de clases prácticas:</b>	Jueves - 19 a 22
<b>Jefe de Trabajos Prácticos:</b>	Graciela Ratto
<b>Docente guía:</b>	Gavinowich Gabriel

Autores			Seguimiento del proyecto									
Nombre	Apellido	Padrón										
Violeta	Perez Andrade	101456										

Observaciones:

---

---

---

---

---

---

---

---

---

---

<b>Fecha de aprobación</b>

<b>Firma J.T.P.</b>

<b>COLOQUIO</b>	
<b>Nota final</b>	
<b>Firma Profesor</b>	

# Índice

1. Objetivo del proyecto	3
2. Descripción del proyecto	3
3. Diagrama de conexiones en bloques	3
4. Circuito esquemático	4
5. Listado de componentes y tabla de gastos	4
6. Diagrama de flujo	5
7. Resultados	5
8. Conclusiones	5
9. Apéndice: Código del programa	6

## 1. Objetivo del proyecto

El objetivo del presente trabajo es familiarizarse con el microcontrolador, mas precisamente con el manejo de puertos de entrada y salida. Para esto se realizó un programa que hace parpadear un LED y permita activar o desactivar el parpadeo con dos pulsadores.

## 2. Descripción del proyecto

En este trabajo práctico se desarrolló un programa en lenguaje ensamblador. El mismo, se encarga de hacer parpadear un LED permitiendo activar o desactivar el parpadeo presionando los pulsadores 1 y 2 respectivamente, uno conectado al PIN 13 y el otro conectado al PIN 14. El mismo se realizo utilizando una rutina de *delay* adaptada de una rutina provista por la cátedra.

## 3. Diagrama de conexiones en bloques

En el siguiente diagrama se puede observar cómo se conectaron los distintos bloques del trabajo

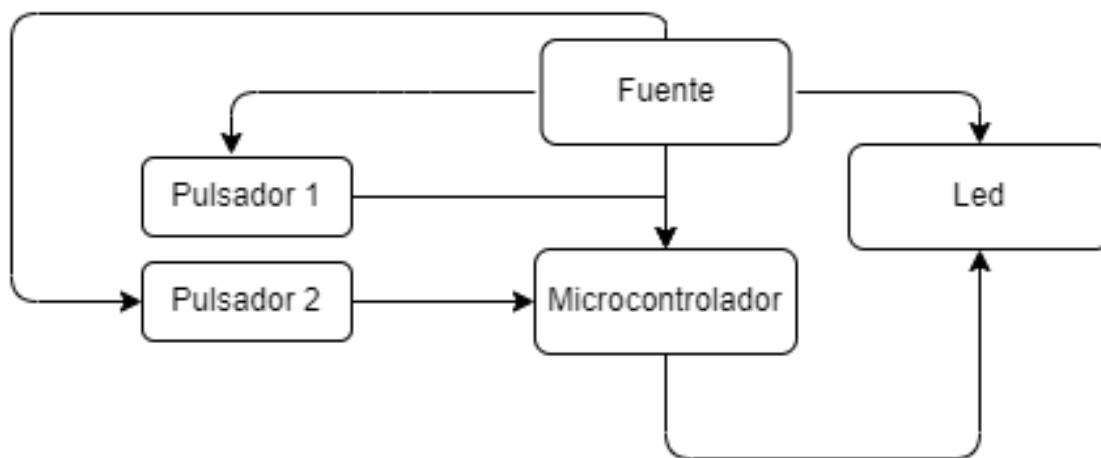


Figura 1: Diagrama de conexión en bloques

## 4. Circuito esquemático

Se incluye el diagrama esquemático del trabajo

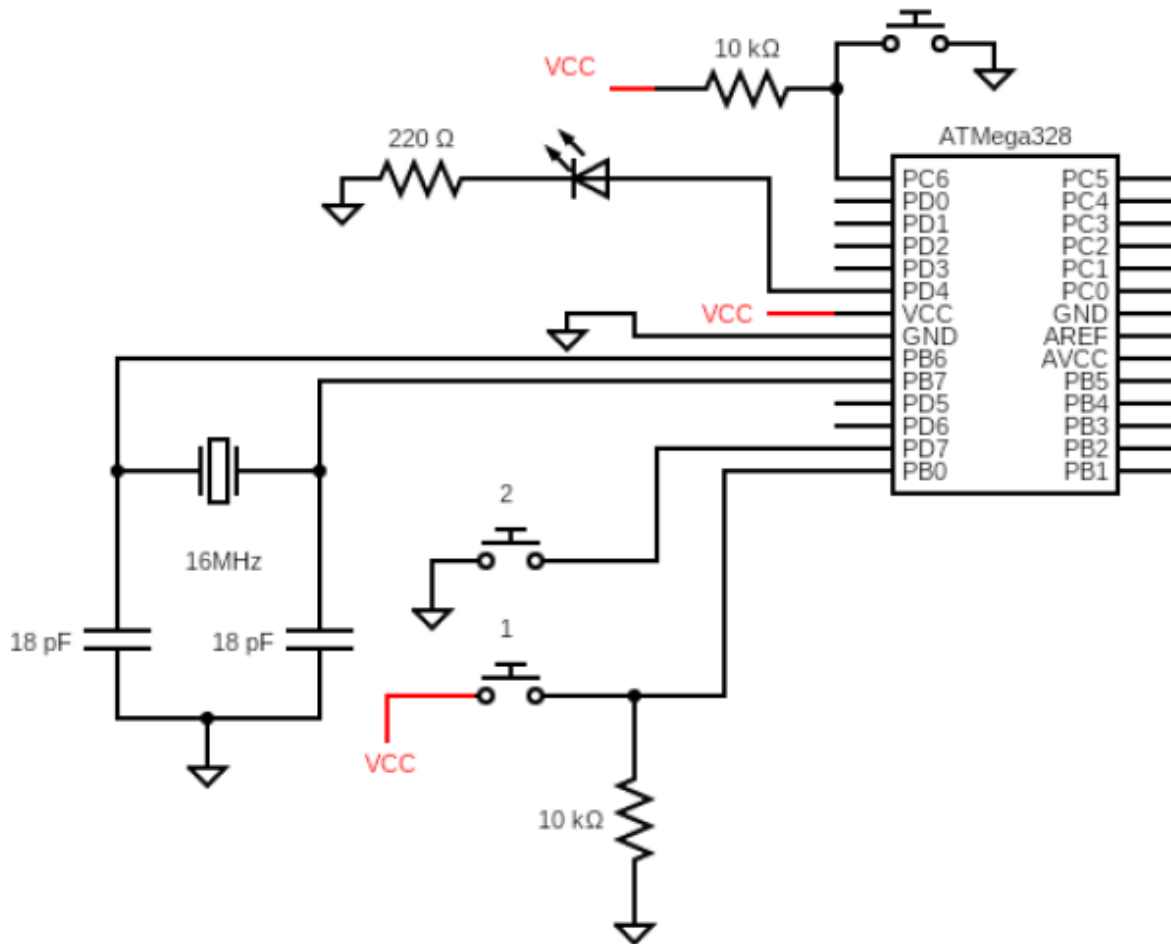


Figura 2: Circuito esquemático

## 5. Listado de componentes y tabla de gastos

Listado de componentes		
Componente	Cantidad	Precio uni.
LED Rojo 5mm	1	\$ 8,91
Touch Switch 5mm	2	\$ 22,28
Resistencia 220 $\Omega$	1	\$ 35,52
Resistencia 10 $\Omega$	1	\$ 16,30
Atmega 328P	1	\$ 878.90
Tira de 40 cables	1	\$ 727,91
Total		\$ 1706,1

## 6. Diagrama de flujo

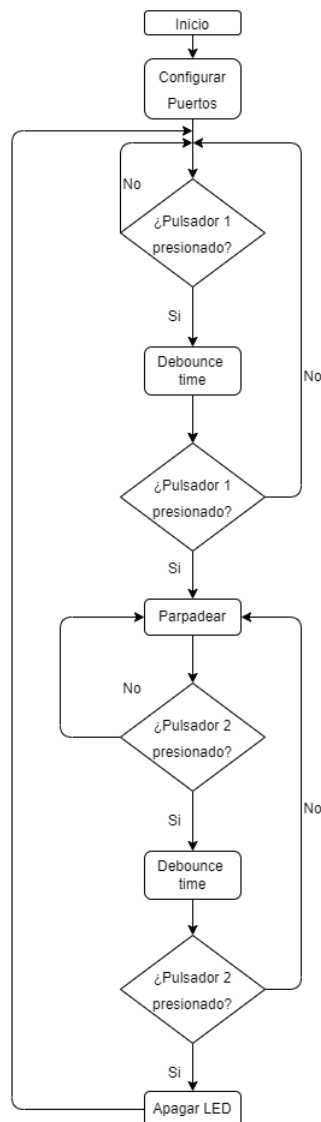


Figura 3: Diagrama de flujo

## 7. Resultados

Los resultados fueron los esperados, como se desarrolló en el código, el delay del parpadeo del LED efectivamente fue de 1s. Y al presionar los pulsadores sucede lo esperado

## 8. Conclusiones

En el presente trabajo se aprendió a utilizar los puertos del microcontrolador como entradas y salidas digitales. A su vez, se evidenció que para poder hacer un retardo al encender y apagar un LED se debe realizar una rutina que ocupe el tiempo en ciclos de máquina, realizando la cuenta de cuántos ciclos de máquina son requeridos en base a las instrucciones que se ejecutarán.

Por otra parte, también se observó que bajo ciertas condiciones es necesario insertar un retardo antirrebote al leer la entrada de un pulsador ya que el mismo no produce una señal escalón ideal, sino que la misma tiene un cierto ruido debido a diferentes factores físicos.

## 9. Apéndice: Código del programa

```
1  ;
2  ; TP1v2.asm
3  ;
4  ; Rutina que hace titilar un led al apretar el pulsador uno
5  ; y apagarlo al presionar el pulsador 2
6  ; Author : Violeta Perez Andrade 101456
7  ;
8
9
10 ; Replace with your application code
11 .include "m328Pdef.inc"
12
13 .equ    LED_PORT_DIR =  DDRD
14 .equ    LED_PORT     =  PORTD
15 .equ    LED_PIN      =  2
16 .equ    PUL1_PORT_DIR =  DDRB
17 .equ    PUL1_PORT_IN  =  PINB
18 .equ    PUL1_PIN      =  0
19 .equ    PUL2_PORT_DIR =  DDRD
20 .equ    PUL2_PORT_IN  =  PIND
21 .equ    PUL2_PIN      =  7
22 .equ    PUL2_PIN_PULL_DOWN = PORTD
23
24 ;Inicio del codigo
25 .org 0x0000
26     rjmp inicio;
27
28 inicio:
29
30 ; Se inicializa el Stack Pointer al final de la RAM utilizando la definicion global
31 ; RAMEND
32     ldi        r16,HIGH(RAMEND)
33     out        sph,r16
34     ldi        r16,LOW(RAMEND)
35     out        spl,r16.
36
37     rcall      configure_ports
38
39 main_loop:
40
41 wait_asc_flank:
42     sbic LED_PORT, LED_PIN
43     rjmp wait_desc_flank
44     sbis    PUL1_PORT_IN, PUL1_PIN
45     rjmp wait_asc_flank
46     rcall   debounce_time ; Se detecto el flanco, espera el tiempo de debounce
47     sbis    PUL1_PORT_IN, PUL1_PIN ;Rechequear el valor para evitar falsos positivos
48     rjmp wait_asc_flank
49     rcall   turn_led_on ; prendo el led
50
51 wait_desc_flank:
52     sbis    LED_PORT, LED_PIN
53     rjmp wait_asc_flank
54     rjmp blink_led
55
56 check_desc_flank:
57     sbic    PUL2_PORT_IN, PUL2_PIN
58     rjmp wait_desc_flank
59     rcall   debounce_time ; Se detecto el flanco, espera el tiempo de debounce
60     sbic    PUL2_PORT_IN, PUL2_PIN ;Rechequear el valor para evitar falsos positivos
61     rjmp wait_desc_flank
62     rcall   turn_led_off
63     rjmp main_loop
64
```

```

65 blink_led:
66     sbis      LED_PORT, LED_PIN
67     rjmp     wait_asc_flank
68     rcall    delay_1s
69
70     rcall    turn_led_off
71     rcall    delay_1s
72
73     rcall    turn_led_on
74     rjmp     check_desc_flank
75
76 ; sbic      PUL1_PORT_IN, PUL1_PIN
77 ; sbi       LED_PORT,LED_PIN //si el pulsador uno esta en 1, quiero encender el led
78 ; sbic      PUL2_PORT_IN, PUL2_PIN //si el pulsador uno esta en 0,
79 ; salteo la instruccion y chequeo el 2
80 ; cbi       LED_PORT,LED_PIN //si esta en uno, no skipeo y apago el led
81 ; rjmp     main_loop
82
83
84 ;*****
85 ; Se configuran los puertos del microcontrolador como entrada/salida/otra funcion
86 ;
87 ; En este caso, los puertos de los pulsadores se configuran como input
88 ; y el led como output
89 ;
90 ;Entrada:
91 ;Salida:
92 ;Registros utilizados: R20
93 ;*****
94
95 configure_ports:
96     ldi       r20,0xFF
97     out       LED_PORT_DIR,r20
98     out       LED_PORT,r20
99     sbi       PUL2_PIN_PULL_DOWN, PUL2_PIN ; Prendo la resistencia de pull-down
100    cbi       PUL2_PORT_DIR,PUL2_PIN
101    cbi       PUL1_PORT_DIR,PUL1_PIN
102    ret
103
104 ;*****
105 ; Subrutina para prender el led
106 ;*****
107
108 turn_led_on:
109     sbi       LED_PORT, LED_PIN
110     ret
111
112 ;*****
113 ; Subrutina para apagar el led
114 ;*****
115
116 turn_led_off:
117     cbi       LED_PORT, LED_PIN
118     ret
119
120 ;*****
121 ; Retardo de 5ms (Calculado con un cristal de 16MHz)
122 ; Si ocurren x ciclos de clock, el tiempo que transcurre es:
123 ;  $t = x/F$  siendo F la frecuencia
124 ; del clock, en este caso 16MHz. Entonces  $x = t * F$ 
125 ; y, particularmente con  $t=5ms$ , entonces  $x = 5e-3 * 16e6 = 80000$ 
126 ; Entonces, necesito 80k clocks
127 ; Para eso, planteo dos ciclos.
128 ; En el loop_2(interno) realizo un dec de un clock y un brne de 2
129 ; En el loop_1 realizo: ldi de un clock, dec de un clock y brne de dos clocks
130 ; (con la salvedad de que el ultimo es de uno)

```

```

131 ; Entonces, siendo m la cantidad de veces que se realiza el loop 2
132 ; y n la cantidad de veces que se realiza el loop 1, queda:
133 ;  $80000 = 3*m*n + 4*m$ 
134 ; Entonces, eligiendo algun valor para m, por ejemplo, 256
135 ; el valor de n sera  $80000 = 3*256*m*n + 4*256 \Leftrightarrow n = 104$ 
136 ;Registros utilizados: r20, r21
137 ;*****
138
139 debounce_time:
140     ldi    r20, 103
141 loop_1:
142     ldi    r21, 255
143 loop_2:
144     dec    r21
145     brne   loop_2
146     dec    r20
147     brne   loop_1
148     ret
149
150 ;*****
151 ; Retardo de 1s (Calculado con un cristal de 16MHz)
152 ;
153 ; Como se explico en el caso anterior para dos ciclos, el tiempo es:
154 ;  $t = (3m+4)*n/16\text{MHz}$  con m y n la cantidad de loops,
155 ; el maximo tiempo que se puede conseguir es:
156 ;  $t = (3*256+4)*256/16\text{MHz} = 12,3 \text{ ms}$ , como se necesita mas tiempo(1s)
157 ; se agrega un tercer ciclo, y siguiendo la misma logica:
158 ;  $t*f = [(3*m + 4)*n + 1+2+1]*p$ ,
159 ; siendo m la cantidad de veces que se ejecuta el ciclo mas interno, n el intermedio,
160 ; y p el mas externo,
161 ;asi:  $p = t*f/[5*m*n + 4*n + 4]$ 
162 ; y dandole a m y n los valores 256, p debe valer 49
163 ;Registros utilizados: r20, r21, r22
164 ;*****
165
166 delay_1s:
167     ldi    r20, 48
168 outer_loop:
169     ldi    r21, 255
170 middle_loop:
171     ldi    r22, 255
172 inner_loop:
173     sbis   PUL2_PORT_IN, PUL2_PIN ; Si a la mitad se aprieta el boton salgo del ciclo
174     ret
175     dec    r22
176     brne   inner_loop
177
178     dec    r21
179     brne   middle_loop
180
181     dec    r20
182     brne   outer_loop
183     ret

```