



**FACULTAD
DE INGENIERIA**

Universidad de Buenos Aires

(6609) LABORATORIO DE MICROCOMPUTADORAS

Proyecto:

TP2 - Interrupciones externas y Memoria EEPROM

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

Autores			Seguimiento del proyecto									
Nombre	Apellido	Padrón										
Violeta	Perez Andrade	101456										

Observaciones:

Fecha de aprobación

Firma J.T.P.

COLOQUIO	
Nota final	
Firma Profesor	

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 manejo tanto de interrupciones externas como de la memoria EEPROM. Para esto se realizó un programa que lee una tabla de la EEPROM (y la carga en caso de no estar) y muestra el dato leído en un display de segmentos.

2. Descripción del proyecto

En este trabajo práctico se desarrolló un programa en lenguaje ensamblador. El mismo, se encarga de mostrar en un display los valores leídos en una tabla ubicada en la EEPROM, avanzando o retrocediendo a lo largo de la misma en base al pulsador que se presione. Para ello, a diferencia del TP pasado se utilizaron interrupciones en vez de estar constantemente chequeando el estado del pulsador. También, una vez que el microcontrolador se apaga (por el motivo que sea) al volver a encenderse, el mismo va a mostrar el valor que estaba mostrando al momento de apagarse.

3. Diagrama de conexiones en bloques

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

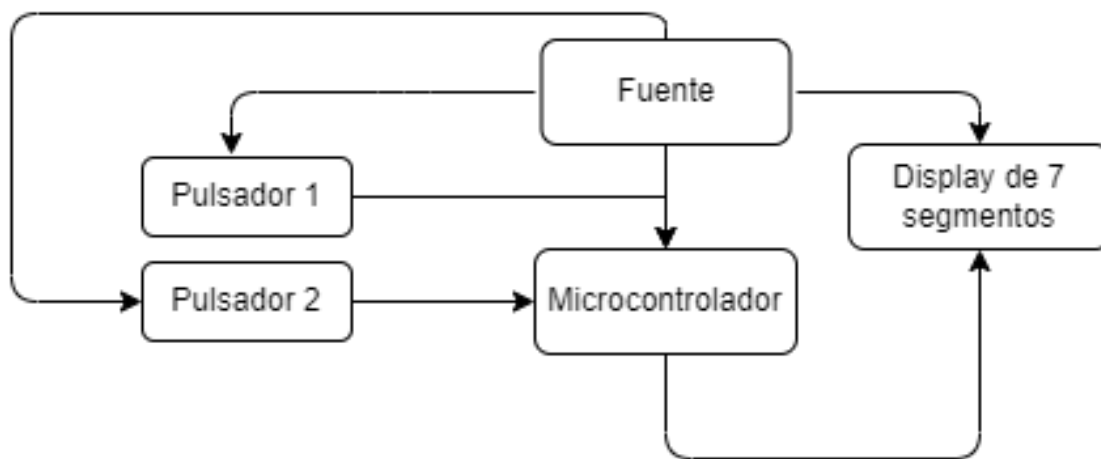


Figura 1: Diagrama de conexión en bloques

4. Circuito esquemático

Se incluye el diagrama esquemático del trabajo

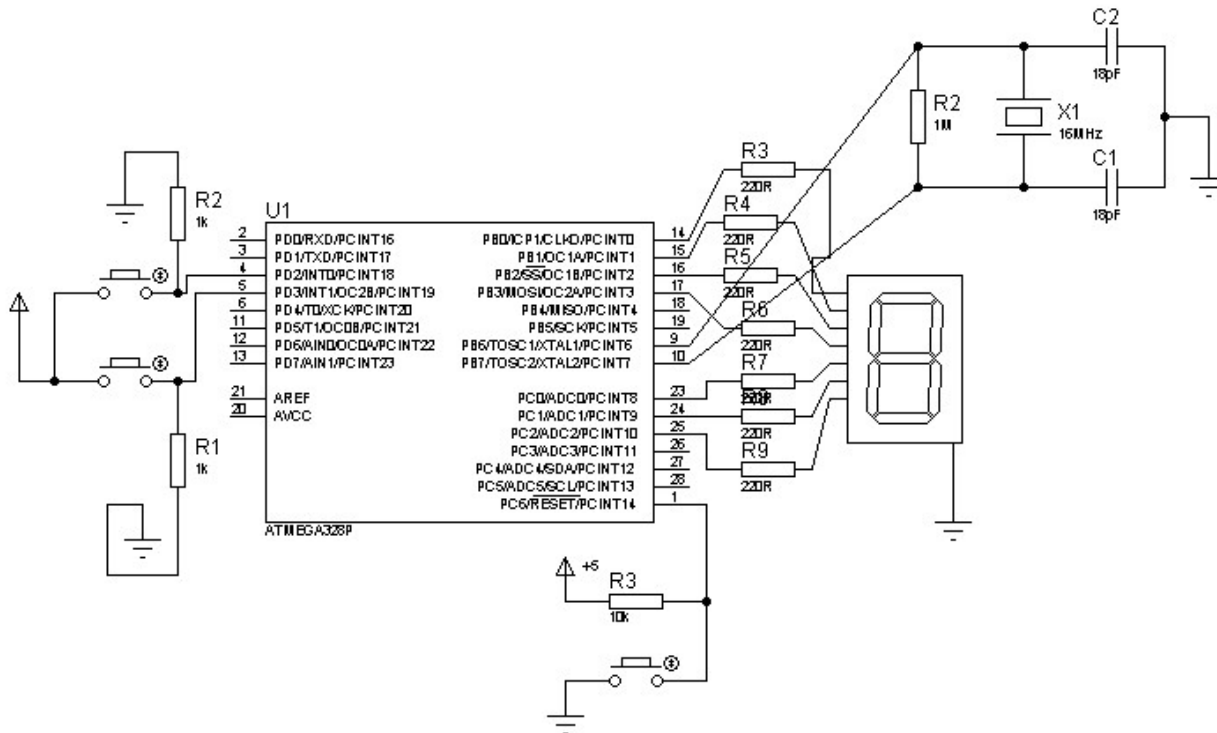


Figura 2: Circuito esquemático

5. Listado de componentes y tabla de gastos

Listado de componentes		
Componente	Cantidad	Precio uni.
Display 7 segemtnos	1	\$ 70,91
Touch Switch 5mm	2	\$ 22,28
Resistencia 220 Ω	7	\$ 35,52
Resistencia 10 Ω	2	\$ 16,30
Atmega 328P	1	\$ 878.90
Tira de 40 cables	1	\$ 727,91
Total		\$ 2003,52

6. Diagrama de flujo

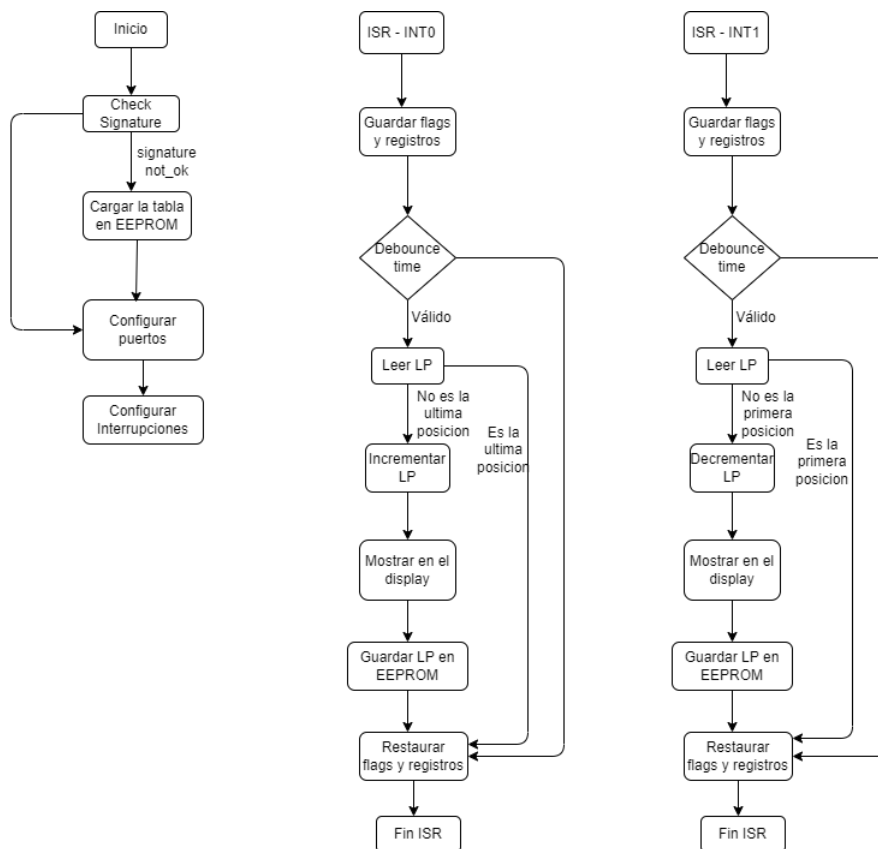


Figura 3: Diagrama de flujo

7. Resultados

Los resultados fueron los esperados, como se desarrolló en el código, se muestran correctamente los números leídos y al presionar los pulsadores sucede lo esperado

8. Conclusiones

En el presente trabajo se aprendió a utilizar las interrupciones del microcontrolador para detectar entradas sin tener que verificar manualmente. A su vez, se aprendió también sobre la memoria EEPROM, como funciona la misma y como es el mecanismo para leer y escribir datos desde y hacia la misma, que es diferente a la memoria RAM.

Por ultimo, se observó también que al utilizar interrupciones para detectar entradas provenientes de pulsadores todavía sigue siendo necesario contar con un retardo antirrebote dentro de la rutina de interrupción, aún cuando la rutina de interrupción se ejecute durante más tiempo que el rebote del pulsador. Esto se debe a que el microcontrolador es capaz de detectar una interrupción mientras otra está siendo atendida.

9. Apéndice: Código del programa

```
1  ;
2  ; TP2b.asm
3  ; Author : Violeta Perez Andrade 101456
4  ;
5
6
7  ; Replace with your application code
8  .include "m328Pdef.inc"
9  ;Inicio del codigo
10 .org 0x0000
11  rjmp inicio
12 .org INT0Addr
13  rjmp isr_int0
14 .org INT1Addr
15  rjmp isr_int1
16
17 .org INT_VECTORS_SIZE
18
19 inicio:
20
21 ; Se inicializa el Stack Pointer al final de la RAM utilizando la definicion global
22 ; RAMEND
23     ldi        r16,HIGH(RAMEND)
24     out        sph,r16
25     ldi        r16,LOW(RAMEND)
26     out        spl,r16.
27
28 main_loop:
29     rcall      check_signature
30     brtc      table_ok
31     rcall      load_table_in_eeprom
32 table_ok:
33     rcall      configure_ports
34     rcall      display_a_to_f
35     rcall      display_last
36     rcall      configurar_interrupciones
37     sei
38     here:     jmp here
39
40 ;*****
41 ; Subrutina que chequea la firma
42 ; para saber si la tabla esta cargada en la EEPROM
43 ; Registros utilizados: r17, r18, r19, r20
44 ;*****
45
46 check_signature:
47     push r17
48     push r18
49     push r19
50     ldi    r17, 0x20
51     ldi    r18, 0x21
52     ldi    xl, 0x0
53     ldi    xh, 0x0
54     rcall  read_eeprom ;X y r20
55     cpse  r20, r17
56     rjmp  fallo_signature
57     clr   r0
58     ldi   r19, 1
59     add   xl, r19
60     adc   xh, r0
61     rcall read_eeprom
62     cpse  r20, r18
63     rjmp  fallo_signature
64     rjmp  signature_ok
```

```

65     fallo_signature:
66         set
67         rjmp fin_ckeck_signature
68     signature_ok:
69         clt
70     fin_ckeck_signature:
71         pop r19
72         pop r18
73         pop r17
74         ret
75
76 ;*****
77 ; Subrutina para cargar la tabla de datos en
78 ; la memoria EEPROM
79 ; registros utilizados: r19, r20, r21
80 ;*****
81
82 load_table_in_eeprom:
83     push r21
84     push r20
85     push r19
86     ldi xl, 0x2
87     ldi xh, 0x0
88     ldi z1, LOW(TABLA_EEPROM<<1)
89     ldi zh, HIGH(TABLA_EEPROM<<1)
90     ldi r21, 22
91
92     loop_load_table:
93         lpm r20, z
94         rcall write_eeprom
95         clr r0
96         ldi r19, 1
97         add xl, r19
98         adc xh, r0
99         clr r0
100        add z1, r19
101        adc zh, r0
102        dec r21
103        brne loop_load_table
104        ldi xl, 0x0
105        ldi xh, 0x0
106        ldi r20, 0x20
107        rcall write_eeprom ;r20
108        clr r0
109        ldi r19, 1
110        add xl, r19
111        adc xh, r0
112        ldi r20, 0x21
113        rcall write_eeprom
114        clr r0
115        ldi r19, 1
116        add xl, r19
117        adc xh, r0
118        pop r19
119        pop r20
120        pop r21
121        ret
122
123 ;*****
124 ; Subrutina para configurar los puertos de los pulsadores
125 ; y del display
126 ; registros utilizados: r20, r21
127 ;*****
128
129 configure_ports:
130     push r20

```

```

131     push r21
132     ldi r21, 0x0
133     ldi r20, 0xFF
134     out DDRC, r20
135     out DDRB, r20
136     out DDRD, r21
137     pop r21
138     pop r20
139     ret
140
141 ;*****
142 ; Subrutina para mostrar, al comienzo del programa
143 ; la secuencia de la A a la F
144 ; registros utilizados: r17, r18, r19
145 ;*****
146
147 display_a_to_f:
148     push r17
149     push r18
150     push r19
151     ldi z1, LOW(DIC<<1)
152     ldi zh, HIGH(DIC<<1)
153     ldi r18, 6
154
155     loop_display_af:
156     lpm r17, z
157     rcall display_number
158     rcall delay_1s
159     clr r0
160     ldi r19, 1
161     add z1, r19
162     adc zh, r0
163     dec r18
164     brne loop_display_af
165
166     pop r19
167     pop r18
168     pop r17
169     ret
170     //ldi r17, 0xA
171     //rcall display_number
172     //rcall delay_1s
173     //ldi r17, 0xB
174     //rcall display_number
175     //rcall delay_1s
176     //ldi r17, 0xC
177     //rcall display_number
178     //rcall delay_1s
179     //ldi r17, 0xC
180     //rcall display_number
181     //rcall delay_1s
182     //ldi r17, 0xD
183     //rcall display_number
184     //rcall delay_1s
185     //ldi r17, 0xE
186     //rcall display_number
187     //rcall delay_1s
188     //ldi r17, 0xF
189     //rcall display_number
190     //rcall delay_1s
191
192 ;*****
193 ; Subrutina para mostrar en el display el ultimo numero
194 ; que fue mostrado antes de que se apague el micro
195 ; registros utilizados: r17, r20
196 ;*****

```



```

197
198 display_last:
199     ldi     x1, 0x17 ;pos de LP
200     ldi     xh, 0x0
201     rcall  read_eeprom
202     mov     x1, r20 ;carga en x1 el indice del dato que quiero dibujar
203     rcall  read_eeprom
204     mov     r17, r20
205     rcall  display_number
206     ret
207
208 ;*****
209 ; Subrutina para configurar las interrupciones
210 ; registros utilizados: r16
211 ;*****
212
213 configurar_interrupciones:
214     ; INT0 e INT1 responden al flanco ascendente
215     ldi     r16, (1 << ISC11) | (1 << ISC10) | (1 << ISC01) | (1 << ISC00)
216     sts     EICRA, r16
217     ; Activar interrupciones para INT0 e INT1
218     ldi     r16, (1 << INT1) | (1 << INT0)
219     out     EIMSK, r16
220     ret
221
222 ;*****
223 ; Subrutina para, dado un numero, mostrar
224 ; este en el display de 7 segmentos
225 ; registros utilizados: r17, r18, r4
226 ;*****
227
228 display_number:
229     push    r4
230     push    r18
231     push    r17
232     push    z1
233     push    zh
234
235     ldi     z1, LOW(TABLA<<1) ;ZL = 0x00 (low byte of address)
236     ldi     zh, HIGH(TABLA<<1) ;ZH = 0x05 (high byte of address)
237
238     clr     r0
239     add     z1, r17
240     adc     zh, r0
241
242     lpm     r17, z
243     mov     r4, r17
244     mov     r18, r17
245     andi    r18, 0b00001111
246
247     swap    r4
248
249     out     PORTB, r4
250     out     PORTC, r18
251
252     pop     zh
253     pop     z1
254     pop     r17
255     pop     r18
256     pop     r4
257     ret
258
259 ;*****
260 ; Subrutina para poder leer un dato de una posicion
261 ; dada de la memoria EEPROM
262 ; registros utilizados: r20

```

```

263 ;*****
264
265 read_eeprom:
266     ; Wait for completion of previous write
267     sbic EECR,EEPE
268     rjmp read_eeprom
269     ; Set up address X in address register
270     out EEARH, XH
271     out EEARL, XL
272     ; Start eeprom read by writing EERE
273     sbi EECR,EERE
274     ; Read data from Data Register
275     in r20,EEDR
276     ret
277
278 ;*****
279 ; Subrutina para poder escribir un dato de una posicion
280 ; dada en la memoria EEPROM
281 ; registros utilizados: r20
282 ;*****
283
284 write_eeprom:
285     ; Wait for completion of previous write
286     sbic EECR,EEPE
287     rjmp write_eeprom
288     ; Set up address X in address register
289     out EEARH, XH
290     out EEARL, XL
291     ; Write data (r16) to Data Register
292     out EEDR,r20
293     ; Write logical one to EEMPE
294     sbi EECR,EEMPE
295     ; Start eeprom write by setting EEPE
296     sbi EECR,EEPE
297     ret
298
299 ;*****
300 ; Subrutina para que al presionar el pulsador
301 ; se avance en la tabla
302 ; registros utilizados: r17, r21, r22, r24,
303 ;*****
304
305 isr_int0: ; Pulsador 1
306     push r24
307     push r22
308     push r21
309     push r17
310     in r24, SREG ; Guardar registro de estado
311     call delay50ms
312     sbis PIND, 2 ; antirebote
313     rjmp isr_int0_fin
314     ldi xl, 0x17 ;pos de LP
315     ldi xh, 0x0
316     rcall read_eeprom
317     cpi r20, 0x16 ;si ya llegue al fin de la tabla salgo
318     brge isr_int0_fin
319     inc r20
320 draw_isr_int0:
321     mov r21, r20 ; me guardo lo que voy a dejar en lp
322     mov xl, r20 ;carga en xl el indice del prox dato(que quiero dibujar)
323     rcall read_eeprom
324     mov r17, r20 ;dejo en r17 lo que quiero dibujar para llamar a display number
325     rcall display_number
326     ;ahora quiero dejar en lp el indice del dato dibujado (r21)
327     ldi xl, 0x17
328     mov r20, r21

```

```

329     rcall write_eeprom
330
331 ;*****
332 ; Subrutina para que al presionar el pulsador
333 ; se retroceda en la tabla
334 ; registros utilizados: r17, r21, r22, r24,
335 ;*****
336
337 isr_int0_fin:
338     out SREG, r24 ; Restaurar registro de estado
339     pop r17
340     pop r21
341     pop r22
342     pop r24
343
344     reti
345
346 isr_int1: ; Pulsador 2
347     push r24
348     push r22
349     push r21
350     push r17
351     in r24, SREG ; Guardar registro de estado
352     call delay50ms
353     sbis PIND, 3 ; antirebote
354     rjmp isr_int1_fin
355     ldi xl, 0x17 ;pos de LP
356     ldi xh, 0x0
357     rcall read_eeprom
358     cpi r20, 0x3 ;si estoy al comienzo de la tabla salgo
359     brlt isr_int1_fin
360     dec r20
361 draw_isr_int1:
362     mov r21, r20 ; me guardo lo que voy a dejar en lp
363     mov xl, r20 ;carga en xl el indice del prox dato(que quiero dibujar)
364     rcall read_eeprom
365     mov r17, r20 ;dejo en r17 lo que quiero dibujar para llamar a display number
366     rcall display_number
367     ;ahora quiero dejar en lp el indice del dato dibujado (r21)
368     ldi xl, 0x17
369     mov r20, r21
370     rcall write_eeprom
371
372 isr_int1_fin:
373     out SREG, r24 ; Restaurar registro de estado
374     pop r17
375     pop r21
376     pop r22
377     pop r24
378
379     reti
380
381 delay50ms:
382     push r19
383     push r18
384     push r17
385     ldi r19, 4
386     loop0:
387     ldi r18, 201
388     loop1:
389     ldi r17, 248
390     loop2:
391     nop
392     dec r17
393     brne loop2
394     dec r18

```

```

395 | brne loop1
396 | dec r19
397 | brne loop0
398 | pop r17
399 | pop r18
400 | pop r19
401 | ret
402 |
403 | delay_1s:
404 |     ldi r20, 50
405 | outer_loop:
406 |     ldi r21, 255
407 | middle_loop:
408 |     ldi r22, 255
409 | inner_loop:
410 |     dec r22
411 |     brne inner_loop
412 |
413 |     dec r21
414 |     brne middle_loop
415 |
416 |     dec r20
417 |     brne outer_loop
418 |     ret
419 |
420 | TABLA:      .DB 243, \
421 |              96, \
422 |              181, \
423 |              244, \
424 |              102, \
425 |              214, \
426 |              215, \
427 |              112, \
428 |              247, \
429 |              246, \
430 |              119, \
431 |              199, \
432 |              147, \
433 |              229, \
434 |              151, \
435 |              23
436 |
437 | TABLA_EEPROM: .DB 0,2,4,6,8,0xA,0xC,0xE,0xF,0xD,0xB,9,7,5,3,1,0,0xF,0,0xA,8,2,20,21
438 | DIC: .DB 0xA,0xB,0xC,0xD,0xE,0xF

```