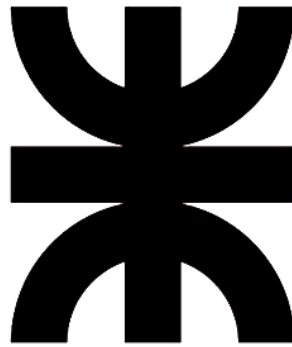


UNIVERSIDAD TECNOLÓGICA NACIONAL



FACULTAD REGIONAL PARANÁ

CARRERA: Ingeniería Electrónica

CÁTEDRA: Técnicas Digitales II

Trabajo Práctico N°9

Implementación de un dado electrónico

ALUMNOS:

Battaglia Carlo

Escobar Gabriel

Paraná, 16 de noviembre de 2022

1. Circuito implementado

Idem trabajo práctico N°8.

2. Código

```
1 ;
2 ; TP N°9.asm
3 ;
4 ; Authors : Battaglia Carlo y Escobar Gabriel
5 ;
6 .cseg
7 .org 0x00
8 rjmp start
9
10 .org PCIF1addr           ;dirección de memoria de Pin Change Interrupt
11     Request 1
12 rjmp isr_PCIF1_handler   ;ir a esta dirección si ocurre Pin Change Interrupt
13     1
14
15 .org OVIF1addr           ;dirección de memoria de Timer/Counter1 Overflow
16     Interrupt
17 rjmp isr_OVIF1_handler   ;ir a esta dirección si ocurre Timer/Counter1
18     Overflow Interrupt
19
20 .def OVFCOUNTERL = r20    ;registros que usaremos para contar interrupciones
21 .def OVFCOUNTERH = r21
22
23 start:
24 ldi r16, 0x80             ;Habilita interrupciones
25 out SREG, r16            ;Reset status del sistema
26
27 ldi r16, LOW(RAMEND)      ;Inicio del stack pointer
28 out SPL, r16
29 ldi r16, HIGH(RAMEND)
30 out SPH, r16
31
32 clr r16                  ;inicialización del Timer/Counter 1
33 sts TCCR1A, r16
34 clr r16
35 sts TCCR1B, r16
36 clr r16
37 sts TIMSK1, r16
38
39 clr r16
40 sts TCNT1L, r16
41 clr r16
42 sts TCNT1H, r16
43
44 clr r16
45 ldi r16, (1<<PCINT8)
46 sts PCMSK1, r16          ;Habilita pin pc0 para interrupciones
47 clr r16
48 ldi r16, (1<<PCIE1)
49 sts PCICR, r16           ;Habilita interrupcion 1 de cambio en el pin
50
```

```

47 clr r16
48 ldi r16, (1<<CS10)           ;Inicio del timer 1
49 sts TCCR1B, r16
50
51 clr r16
52 ldi r16, (1 << DDD7) | (1 << DDD6) | (1 << DDD5) | (1 << DDD4) | (1 << DDD3)
   | (1 << DDD2) | (1 << DDD1) | (1 << DDD0)
53 out DDRD, r16                ;PortD como salida
54
55 clr r16
56 clr r17
57 clr OVFCOUNTERL
58 clr OVFCOUNTERH
59
60 mainloop:
61 cpi OVFCOUNTERH, 0x02         ;corroborar si ocurrieron el equivalente a 10s en
   interrupciones
62 brcs mainloop                ;(teniendo en cuenta la frecuencia del clock)
63 cpi OVFCOUNTERL, 0x62
64 brcs mainloop
65
66 clr r17                       ;si pasaron 10 segundos resetea el contador y
   apaga el display
67 sts TIMSK1, r17
68 clr OVFCOUNTERL
69 clr OVFCOUNTERH
70 out PORTD, r17
71 rjmp mainloop
72
73 ;esta subrutina divide un número por 6 hasta obtener como resto un número
   entre 0 y 6
74 dividir:
75 cpi r16, 6
76 brcs resto
77 sbci r16, 6
78 rjmp dividir
79 resto:
80 inc r16
81 ret
82
83 ;si hubo un cambio en el PINCO (donde está el pulsador) toma el valor actual
   de TCNT1L y lo divide con la subrutina "dividir" para asignar un número
   entre 0 y 6.
84 isr_PIC1_handler:
85 push r17
86 sbis PINC, PINCO             ;chequea que el cambio en PORTD sea especí
   ficamente en PINCO
87 rjmp noHacerNada
88 lds r16, TCNT1L               ;toma el valor actual de TCNT1L
89 call dividir                  ;asigna un número entre 0 y 6
90 call DecodeNum                ;lo decodifica
91 out PORTD, r16                ;y lo expone en el display que se encuentra en
   el PORTD
92 clr OVFCOUNTERL
93 clr OVFCOUNTERH
94 clr r17
95 ldi r17, (1<<TOIE1)           ;Habilitamos interrupcion de OF
96 sts TIMSK1, r17

```

```

97     noHacerNada:
98     pop r17
99     reti ; Pin Change Interrupt 0
100
101 ;cada vez que ocurre una interrupción por overflow, incrementamos en 1 el
    valor del contador de 16 bits OVFCOUNTER.
102 isr_OVF1_handler:
103     mov r24, OVFCOUNTERL
104     mov r25, OVFCOUNTERH
105     adiw r24, 0x01
106     mov OVFCOUNTERL, r24
107     mov OVFCOUNTERH, r25
108     reti
109
110 ;decodificamos el número para mostrarlo en un display de 7 segmentos. Esta
    vez no utilizamos la tabla de conversión.
111 DecodeNum:
112     cpi r16, 6
113     breq num6
114     cpi r16, 5
115     breq num5
116     cpi r16, 4
117     breq num4
118     cpi r16, 3
119     breq num3
120     cpi r16, 2
121     breq num2
122 num1:
123     ldi r16, 0x06
124     RET
125 num2:
126     ldi r16, 0x5B
127     RET
128 num3:
129     ldi r16, 0x4F
130     RET
131 num4:
132     ldi r16, 0x66
133     RET
134 num5:
135     ldi r16, 0x6D
136     RET
137 num6:
138     ldi r16, 0x7D
139     RET

```