



FACULTAD DE INGENIERIA

Universidad de Buenos Aires

Laboratorio de Microprocesadores - 86.07

Trabajo Práctico N°4: Interrupción Externa

Profesor:			Ing. Guillermo Campiglio							
Cuatrimestre/Año:			1°/2020							
Turno de las clases prácticas			Miércoles							
Jefe de trabajos prácticos:			Ing. Pedro Ignacio Martos							
Docente guía:			-							
Autor			Seguimiento del proyecto							
Francisco	Rossi	99540								

Observaciones:

.....

.....

.....

.....

.....

.....

.....

.....

Fecha de aprobación			Firma J.T.P		

Coloquio	
Nota final	
Firma profesor	

Índice

1. Introduccin	2
1.1. Objetivo	2
1.2. Descripcin	2
2. Materiales	2
3. Diagrama en Bloques	2
4. Esquemtico	3
5. Diagrama de flujo	4
6. Cdigos	5
7. Frecuencia de oscilacin del LED	7
8. Resultados	8
9. Conclusiones	8

1. Introduccin

En el siguiente informe se explica el diseo de un programa escrito en lenguaje Assembler para hacer parpadear un LED cinco veces a una velocidad aproximada de 1 Hz conectado al pin $PB1$ de un *ATMEGA328p* cuando se genera una interrupcin externa.

1.1. Objetivo

El objetivo es configurar el microcontrolador para que ante una interrupcin externa ejecute una subrutina que realice una accin sobre los puertos I/O del mismo.

1.2. Descripcin

Se conecta un LED con un resistor en serie al pin $PB0$, el mismo se apaga cuando se genera una interrupcin externa y se enciende cuando termina la subrutina asociada, al generarse la interrupcin adems de apagarse el LED en $PB0$ se hace titilar 5 veces a una frecuencia aproximada de 1 Hz a un LED conectado al pin $PB1$.

Para la interrupcin externa se utiliza un pulsador conectado a $INT0$ ($PD2$).

2. Materiales

Se utilizaron los siguiente materiales para el proyecto:

- 1 Pulsador (30\$ (Pesos Argentinos))
- 2 LEDs (40\$ (Pesos Argentinos))
- 2 Resistor de 220Ω (8\$ (Pesos Argentinos))
- 1 Microcontrolador ATmega328p (Utilizando el integrado con el Arduino Uno) (700\$ (Pesos Argentinos))

3. Diagrama en Bloques

En la **Fig. 1** se muestra un diagrama en bloques del circuito.

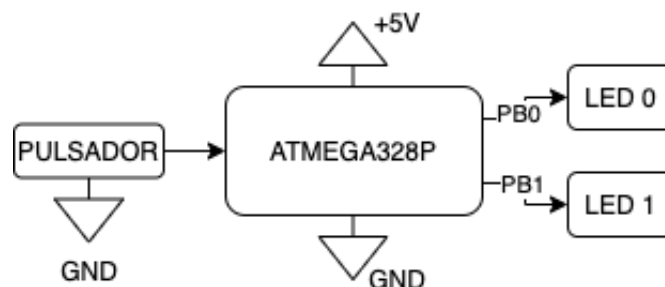


Figura 1: Diagrama en bloques.

4. Esquemtico

En las **Fig. 3** y **Fig. 2** se muestra como se conect el arduino con los LEDs, el pulsador y los resistores de $220\ \Omega$.

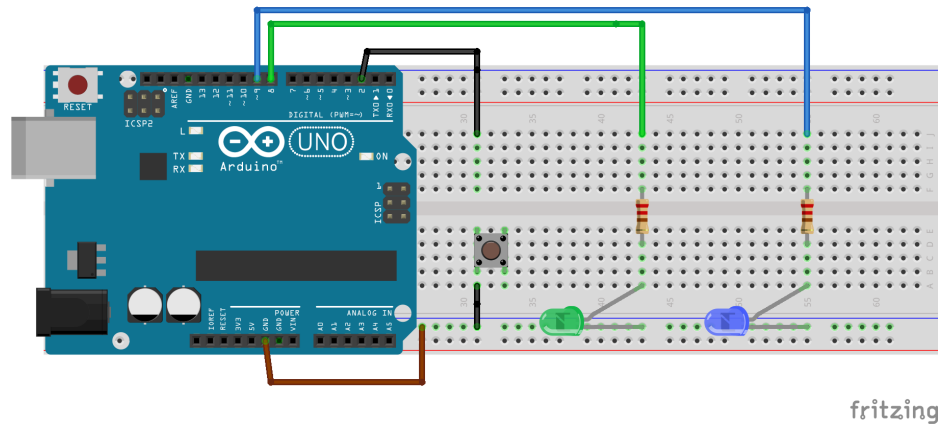


Figura 2: Coneccin entre el arduino, los LEDS y el pulsador

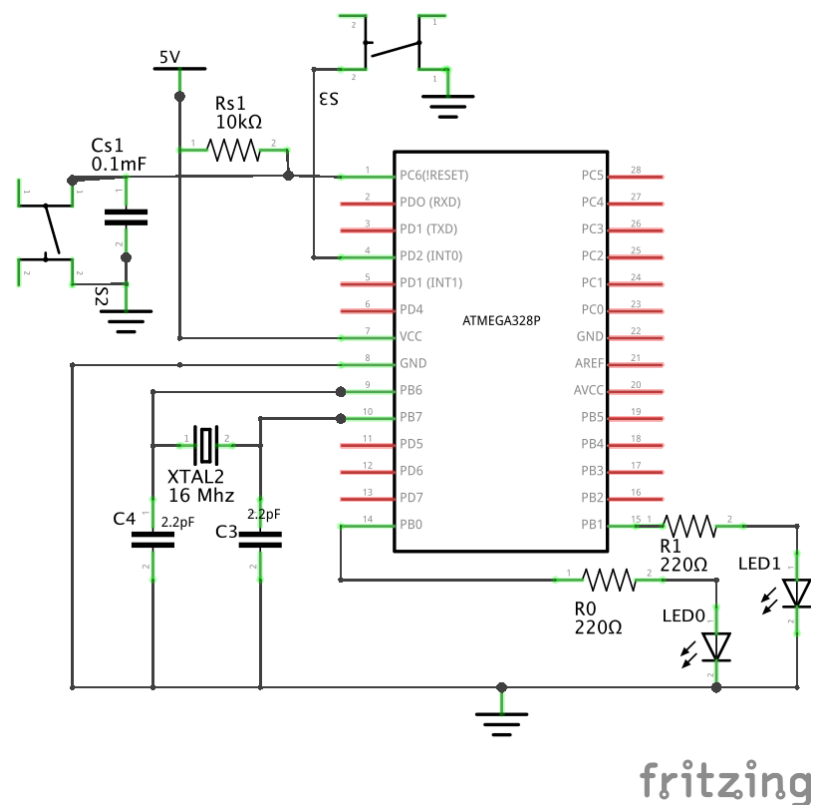


Figura 3: Esquemtico del circuito implementado.

5. Diagrama de flujo

En la **Fig. 4** se muestra el diagrama de flujo del programa.

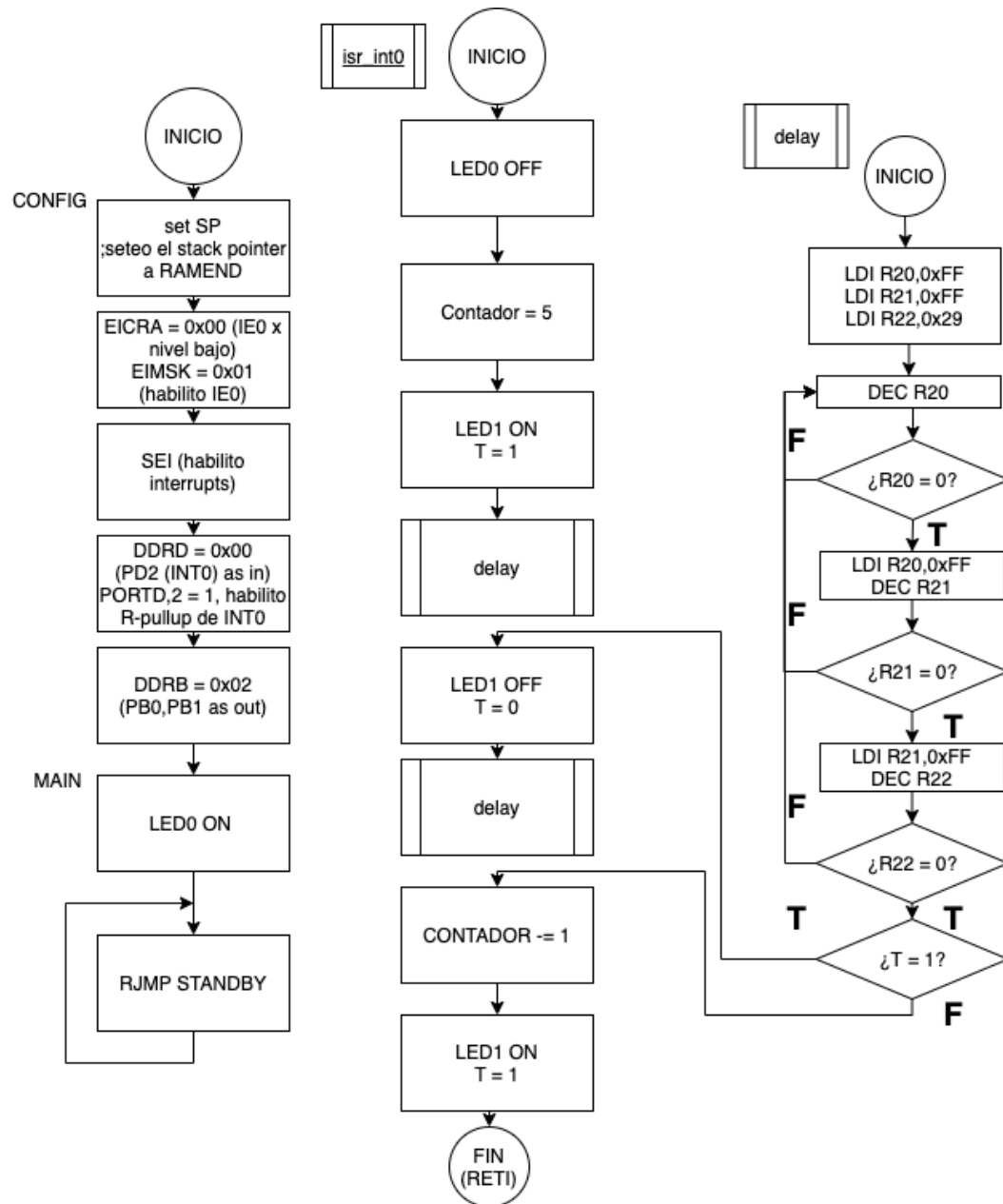


Figura 4: Diagrama de flujo.

6. Cdigos

El siguiente cdigo se puede dividir en tres partes, primero la configuracin **Config** donde se setea el stack pointer, las interrupciones y los puertos. Luego, el **main** donde se enciende el **LED0** y se deja el micro en espera a una interrupcin sin hacer nada. Este estado se modifica solo si se genera una interrupcin externa, la cul es posible generarla con un nivel bajo en **INT0**, dado por el pulsador correspondiente y **PD2** con R-pullup habilitada.

Por ltimo, cuando la interrupcin es generada se dispara la subrutina **isr_int0**, en la cul se apaga el **LED0** y se hace titilar 5 veces el **LED1** a una frecuencia aproximada de 1 Hz (Ver **Frecuencia de oscilacin del LED**), se apaga el **LED0** al salir de la interrupcin y el micro vuelve a la posicin de stand-by.

```
1 ; Autor: Francisco Rossi
2 ; Padron: 99540
3 ; 86.07 Laboratorio de Microprocesadores – FIUBA
4 ; Catedra: Miercoles
5 ; Fecha: 24 de junio de 2020
6
7 .include "m328pdef.inc"
8
9 .def contador = r16
10 .def dummy = r25
11
12 .macro set_sp
13     ldi dummy, low(RAMEND)
14     out spl, dummy
15     ldi dummy, high(RAMEND)
16     out sph, dummy
17 .endm
18
19 .cseg
20 .org 0x0000
21     jmp config
22
23 ; EXT INT
24
25 .org INT0addr
26     jmp isr_int0
27 .org INT.VECTORS_SIZE
28
29 config:
30     set_sp
31
32     ; configuro interrupcion externa INT0,INT1
33     ldi dummy,(0 << ISC01 | 0 << ISC00 ) ;0x03 ; IE0 por flanco descendente
34     sts EICRA,dummy ;(ISC01=1;ISC00=1)
35     ldi dummy,(1 << INT0) ;0x01 ; habilito IE0
36     out EIMSK,dummy
37
38     ; habilito interrupciones
39     sei ;(I en 1)
40
41     ; Configuracion de puertos
42     ; portd,2 como entrada
43     ldi dummy, (0 << 2)
44     out DDRD, dummy
45     ldi dummy, (1 << 2)
46     out PORTD,dummy ;(R pull up activa)
47
48     ; PB0 y PB1 como salidas
49     ldi dummy, (1 << 0 | 1 << 1)
50     out DDRB, dummy
51     clr dummy
52     out PORTB,dummy ; inicializo en cero
53
54 main:
55     sbi PORTB,0
56 standby:
```

```

57         rjmp    standby
58
59
60 isr_int0 :
61         cbi     PORTB,0
62 twink :
63         ldi     contador ,0x05
64 on :
65         sbi     PORTB,1
66         set
67         rjmp    delay
68 off :
69         cbi     PORTB,1
70         clt
71         rjmp    delay
72 rt :
73         dec     contador
74         brne    on
75         sbi     PORTB,0
76         reti
77
78 delay :
79         ldi     r20 ,0xff
80         ldi     r21 ,0xff
81         ldi     r22 ,0x29
82 cycle :
83         dec     r20
84         brne    cycle
85         ldi     r20 ,0xff
86         dec     r21
87         brne    cycle
88         ldi     r21 ,0xff
89         dec     r22 ; con este valor es facil variar de forma apreciable la frecuencia ya que es un
multiplicador de todos los incrementos anteriores
90         brne    cycle
91
92         brts    off
93         jmp     rt
    
```

El ciclo para realizar los retardos **Delay** esta basado en decrementar registros, dependiendo del valor del **bit T** del status register se determina si el **LED1** se debe apagar o prender. En la siguiente seccin se explicar en profundidad la relacin entre el decremento de los registros y los tiempos de encendido y apagado del LED. El cdigo fue pensado para lograr un tiempo en alto y en bajo similares es decir, que el *Duty Cycle* sea lo ms cercano posible al 50%.

7. Frecuencia de oscilacin del LED

Para generar una frecuencia de aproximadamente 1 Hz se comenz utilizando el mismo delay para cuando el LED1 este encendido y cuando este apagado, de manera de generar una cuadrada de un *Duty Cycle* del 50 %.

Dao que la frecuencia por default del clock del *Arduino UNO* es $f_{ck} = 16 \text{ MHz}$.

Luego, conociendo los ciclos de clock que toma cada operacin del ciclo podemos obtener los tiempos de encendido y apagado como $\Delta_t = \frac{\text{ciclos}_{\text{delay}}}{f_{ck}}$, y con esto podemos obtener la frecuencia de oscilacin del LED como $f = \frac{1}{2\Delta_t}$.

El Delay se basa en decrementar 3 registros desde 3 valores mximos, usandolos como un sistema *pesado*:

$$R22R21R20$$

donde por cada vez que se decrementa $R20$ desde $R20_{max}$ hasta cero se decremente una vez $R21$ y cada vez que se decrementa $R21$ desde $R21_{max}$ hasta cero se decrementa $R22$ una vez.

De manera que la cantidad de ciclos totales se podr calcular como la cantidad de ciclos que se toma en decrementar el registro $R20$ (CC_{R20}) sumado a la cantidad de ciclos que se toma en decrementar el registro $R21$ (CC_{R21}) sumado a la cantidad de ciclos que se toma en decrementar el registro $R22$ (CC_{R22}).

Podemos representar lo dicho anteriormente de manera matemtica como:

$$\begin{aligned} \text{ciclos}_{\text{delay}} &= 3 + CC_{R20} + CC_{R21} + CC_{R22} \\ CC_{R20} &= R22_{max} \cdot R21_{max} \cdot [R20_{max} (2 + 1)] - \overbrace{R21_{max} \cdot R22_{max}}^{\text{veces que el brne es falso}} \\ CC_{R21} &= R22_{max} \cdot [R21_{max} (2 + 1 + 1)] - \overbrace{R22_{max}}^{\text{veces que el brne es falso}} \\ CC_{R22} &= R22_{max} \cdot (1 + 1 + 2) - \underbrace{1}_{\text{veces que el brne es falso}} \end{aligned}$$

En este caso se utilizaron los siguientes valores:

$$R20_{max} = R21_{max} = 255$$

De manera que el valor ms cercano a $f = 1 \text{ Hz}$ se obtiene con:

$$R22_{max} = 41 = 0x29$$

Reemplazando por los valores son

Luego,

$$\text{ciclos}_{\text{delay}} = 3 + 7987620 + 41779 + 163 = 8029565 \implies \Delta_t = \frac{8029565}{16 \cdot 10^6} = 0,5018 \text{ seg}$$

Finalmente se obtiene la frecuencia de oscilacin del LED1:

$$f = \frac{1}{2 \cdot \Delta_t} = 0,9963 \text{ Hz}$$

Con estos resultados obtenemos la frecuencia de oscilacin del LED y el *Duty Cycle*:

$$\text{Duty Cycle} = 50 \%$$

$$f = 0,9963 \text{ Hz}$$

8. Resultados

Se logr diseñar un programa para el microcontrolador ATMEGA328p que haga titilar cinco veces un LED colocado en el puerto B cuando se genera una interrupcin externa por nivel bajo en *INT0* a una frecuencia de $f = 0,9963 \text{ Hz}$ y *Duty Cycle* $\approx 50 \%$.

9. Conclusiones

Se logr generar un programa que realice la tarea de configuracin y este listo ante una interrupcin externa a correr una subrutina la cual apague un LED en el *PB0* y haga titilar otro, en *PB1* cinco veces a una frecuencia aproximada de 1 Hz . Es posible tambien modificar la frecuencia de oscilacin y el duty cycle del mismo.