



# FACULTAD DE INGENIERIA

Universidad de Buenos Aires

Laboratorio de Microprocesadores - 86.07

## Interrupción Externa

Profesor:			Ing. Guillermo Campiglio									
Cuatrimestre/Año:			1º/2020									
Turno de las clases prácticas			Miercoles 19 hs									
Jefe de trabajos prácticos:			Pedro Ignacio Martos									
Docente guía:			Pedro Martos, Fabricio Baglivo, Fernando Pucci									
Autores			Seguimiento del proyecto									
Nombre	Apellido	Padrón										
Leonel	Mendoza	101153										

### Observaciones:

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Fecha de aprobación			Firma J.T.P		

Coloquio	
Nota final	
Firma profesor	

## 1. Objetivo

El objetivo de este trabajo es usar rutinas de interrupciones, mediante una interrupción externa por INT0, configurar correctamente el comportamiento de interrupción, y comparar las implementaciones con resistencias de pull-up o pull-down.

## 2. Descripción

Como indica el esquema, hay un pulsador conectado a la INT0 (INTERRUPCION 0), de forma que al accionar el pulsador se produce una interrupción y 2 LEDs conectados al PORTB(0/1). LED0 y LED1.

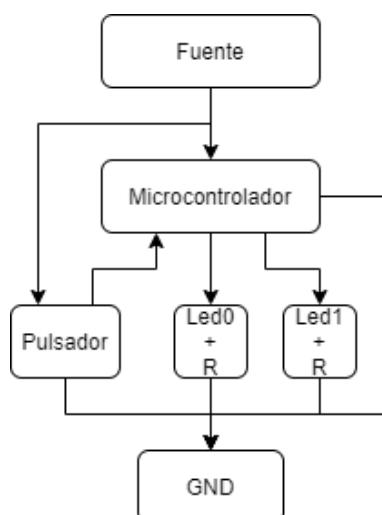
Se pide hacer un programa que prenda y apague LEDs siguiendo estas consignas:

- El programa principal va a ser responsable de encender el LED0
- Se detectará el accionar del pulsador por flanco, la rutina de interrupción debe hacer:
  - apagar el LED0
  - hacer parpadear al LED1 5 veces a una frecuencia aproximada de 1Hz
  - prender el LED0 antes de salir de la interrupción

Resumiendo: cuando resetee el micro prenderá el LED0, cada vez que oprima el pulsador apagará el LED0, parpadeará el LED1 5 veces y prenderá el LED0.

El circuito armado está configurado con un resistencia de pull-down, que modificaría para poner una resistencia de pull-up?

## 3. Diagrama en bloques



## 4. Esquemático

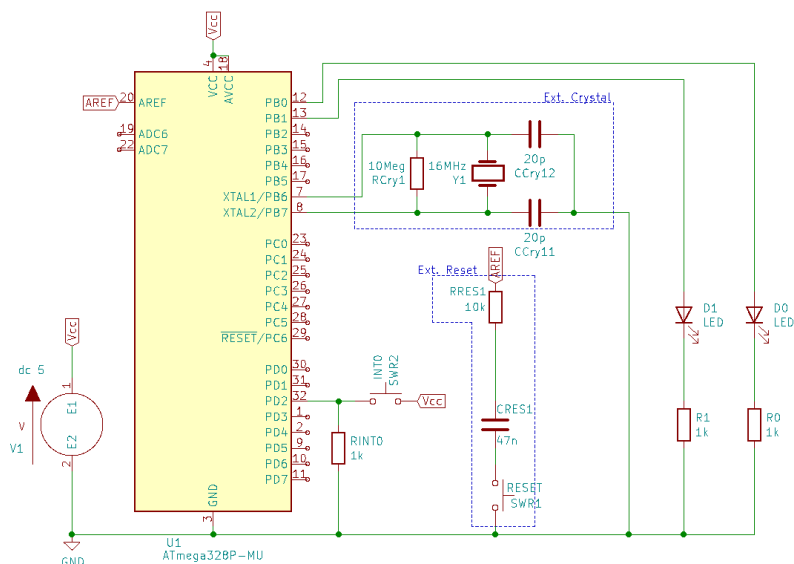


Figura 1: Esquemático con R pull-down

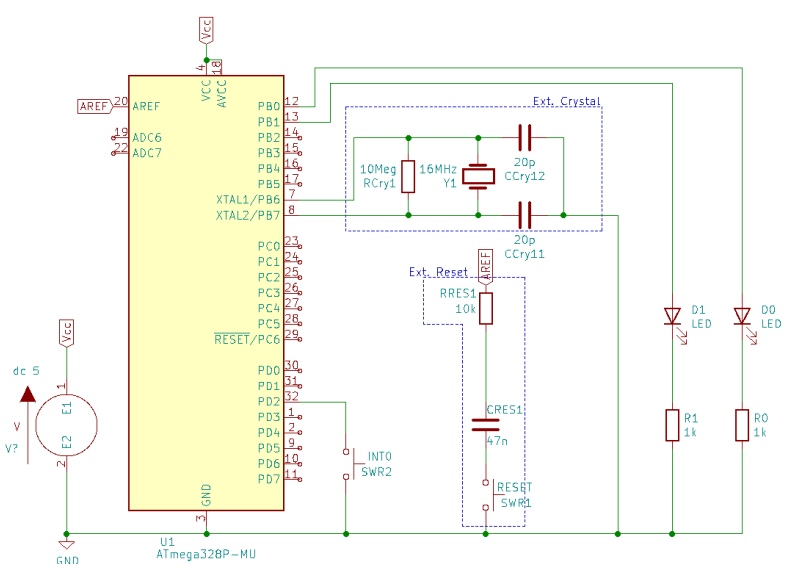
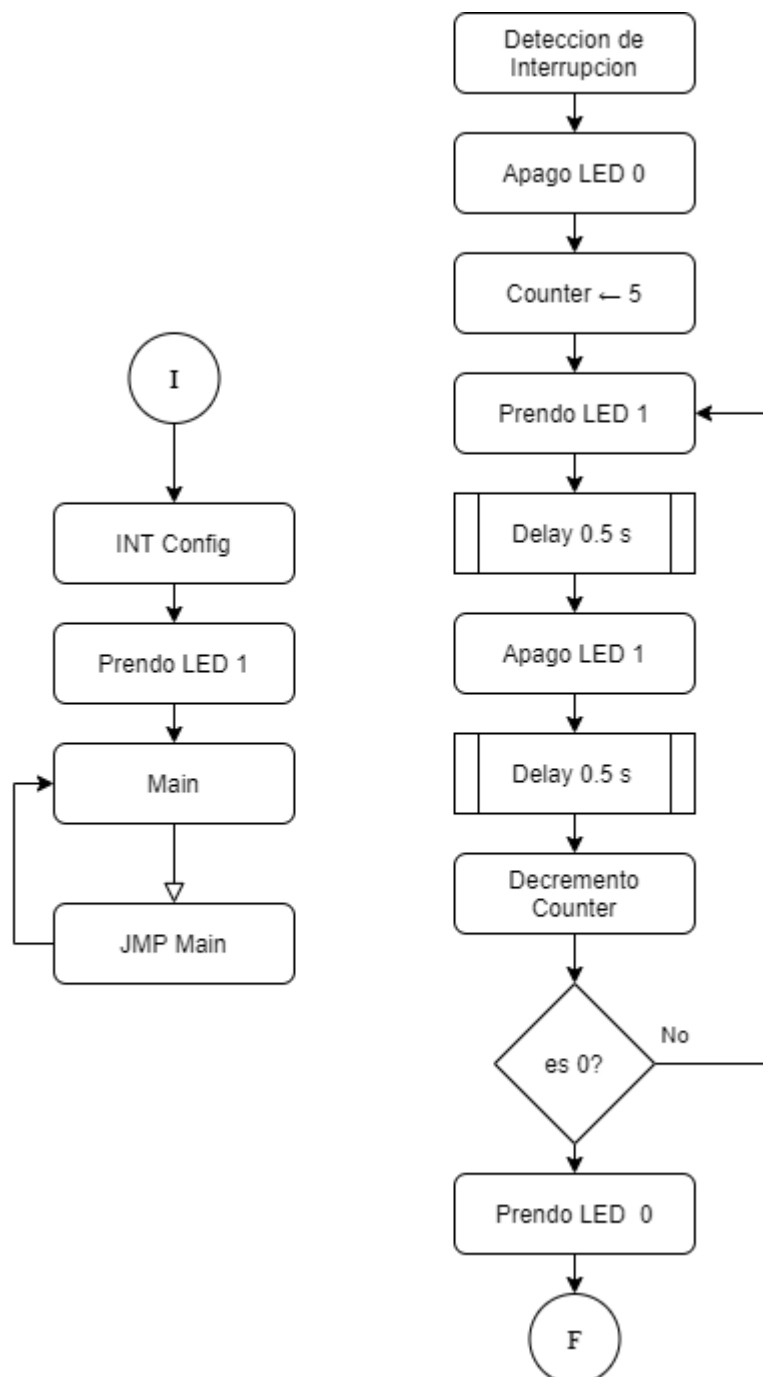


Figura 2: Esquemático con R pull-up

## 5. Listado de componentes

- Microcontrolador *ATmega328p* y programador USBasp (Arduino UNO) [AR\$ 950]
- 2x LED [AR\$ 20]
- 2x Resistencia ( $220\ \Omega$ ) [AR\$ 8]
- Resistencia ( $1\ k\Omega$ ) [AR\$ 4]
- Pulsador [AR\$ 15]

## 6. Diagrama de Flujo



## 7. Código de programa

```
.include "m328pdef.inc"

; * * * * *
;   START MACROS   ;
; * * * * *

.MACRO SET_SP ;[auxGPR]
    LDI @0, low(RAMEND)
    OUT SPL, @0
    LDI @0, high(RAMEND)
    OUT SPH, @0
.ENDM

.MACRO SET_X ;[LABEL to data memory]
    LDI XL, low(@0)
    LDI XH, high(@0)
.ENDM

.MACRO SET_Y ;[LABEL to data memory]
    LDI YL, low(@0)
    LDI YH, high(@0)
.ENDM

.MACRO SET_Z ;[LABEL to prog memory]
    LDI ZL, low(@0 << 1)
    LDI ZH, high(@0 << 1)
.ENDM

.MACRO D10ms ; [reg_cycle], [reg_aux1], [reg_aux2]          9.98 ms
delay:
    clr @2
loop1:
    ldi @1,207
loop2:
    DEC @1                ;cuento 256 veces con @1
    BRNE loop2
    DEC @2                ;cuento 256 veces el conteo de @1 (256*256)
    BRNE loop1
    DEC @0                ;cuento "@0" veces el conteo de @2 (@0*256*256)
    BRNE delay
.ENDM

; * * * * *
;   END MACROS   ;
; * * * * *

.DEF aux = R16
.DEF delay_cycle = R17
```

.CSEG

```
.ORG 0X0000      ; En esta direccion escribo la instruccion JMP conf
JMP conf

.ORG INT0addr    ; Direccion donde escribir el JMP a las rutinas de interrupcion
JMP isr_int0

.ORG INT_VECTORS_SIZE    ; Direccion donde escribir el codigo

conf:
    SET_SP    aux

    LDI        aux, (0 < PD2 || 0 < PD3)
    OUT        DDRD, aux
;    LDI        aux, (1 < PD2 || 1 < PD3)    ; En caso de usar R de pull-up/down descomentar
;    OUT        PORTD, aux

    LDI        aux, (1 << PB0 || 1 << PB1) ; Configuracion de puertos para leds
    OUT        DDRB, aux
    LDI        aux, (1 << PB0 || 0 << PB1) ; Inicializo led_0 on y led_1 off
    OUT        PORTB, aux

    ; Configuro interrupciones

    LDI        aux, (1 << ISC01 || 0 << ISC00)
    STS        EICRA, aux
    LDI        aux, (1 << INT0)
    OUT        EIMSK, aux

    SEI        ; Habilito Interrupciones

main:
    NOP
    RJMP main

isr_int0:
    CBI        PORTB, PB0    ; Apago primer LED
    LDI        aux, 5        ; 5 ciclos de 1 Hz
loop_leds:
    SBI        PORTB, PB1
    LDI        delay_cycle, 50    ; Uso 50 ciclos de delay 10ms (.5 Hz en alto)
    D10ms delay_cycle, R18, R19
    CBI        PORTB, PB1
    LDI        delay_cycle, 50    ; Uso 50 ciclos de delay 10ms (.5 Hz en bajo)
    D10ms delay_cycle, R18, R19
    DEC        aux
    BRNE       loop_leds
    SBI        PORTB, PB0    ; Cuando termino prendo el primer LED nuevamente
    RETI
```

## 8. Resultados

Se logró ejecutar la rutina de interrupción por flanco, el único inconveniente es que de vez en cuando se ejecute 2 veces la interrupción (LED parpadea 10 veces), se supone que esto se debe a ruidos del pulsador al no tener ningún sistema de filtrado de ruidos (como un Schmitt Trigger) y esto ejecuta la rutina 2 veces.

## 9. Conclusiones

Mediante el manejo de interrupciones externas, se logro configurar e implementar correctamente la rutina de interrupción deseada por medio de INT0 con flanco descendente, configurando los registros EICRA y EIMSK para cambiar las condiciones de detección de interrupción, y uso de pines como interruptores externos respectivamente.