FACULTAD
DE INGENIERIA
Universidad de Buenos Aires

Laboratorio de Microprocesadores - 86.07

# Timers

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 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 manejar los registros de timers, generar interrupciones con eventos del timer, manejo de antirrebote de teclas.

# 2. Descripción

Se pide hacer un programa que haga parpadear el LED conectado al PB0, en 3 frecuencias distintas o que lo deje ENCENDIDO FIJO, según los valores que haya en las entradas PD0 PD1 según se indica en la siguiente tabla:

| PD0 | PD1 | Estado del LED |
|---|---|---|
| 0 | 0 | Encendido fijo |
| 0 | 1 | Parpadea con prescaler CLK/64 |
| 1 | 0 | Parpadea con prescaler CLK/256 |
| 1 | 1 | Parpadea con prescaler CLK/1024 |

Nota: si alguien presiona las teclas mientras está funcionando el micro, los leds deberán cambiar acorde a la tabla. Para resolver esta práctica usarán el Timer1, interrupción por OVERFLOW.

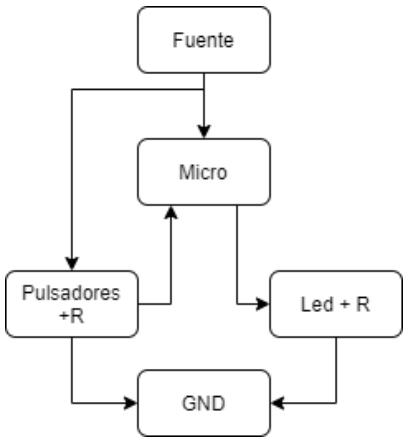
Cuando el LED esté ENCENDIDO FIJO el timer estará apagado.

En los otros 3 casos, el timer contará los pulsos de clock divididos por prescaler 64, 256 y 1024 respectivamente. Cuando se produce un overflow (desborde) deberán cambiar el estado del LED, es decir, si está prendido lo apagan y viceversa.

Calcular la frecuencia o periodo con que se prenderá el LED en los 3 casos, teniendo en cuenta que la frecuencia de un Arduino es de 16 MHz.

En las entradas PD0, PD1 están conectados 2 pulsadores, que como cualquier tecla produce rebotes al ser presionada. Implementar rutinas antirrebote para detectar correctamente las teclas.

# 3. Diagrama en bloques



# 4. Esquemático

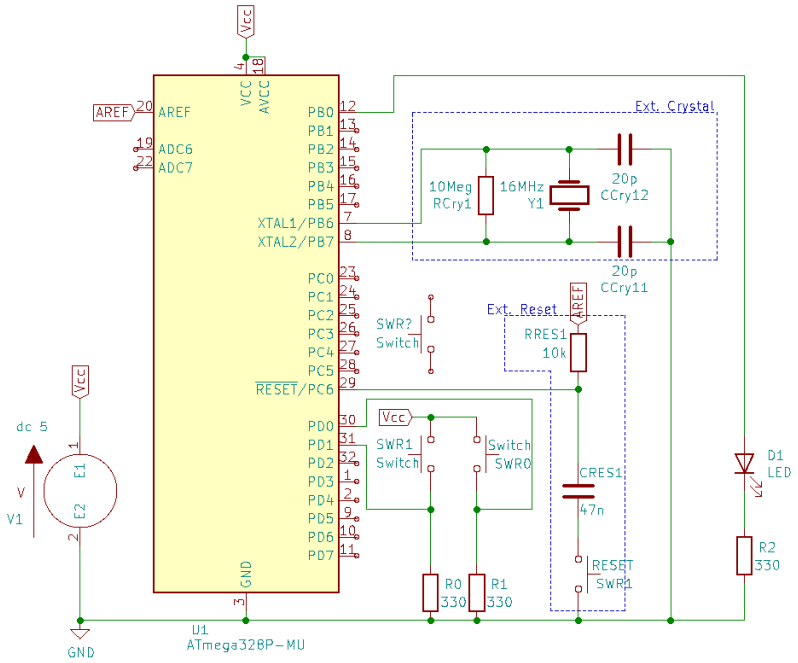
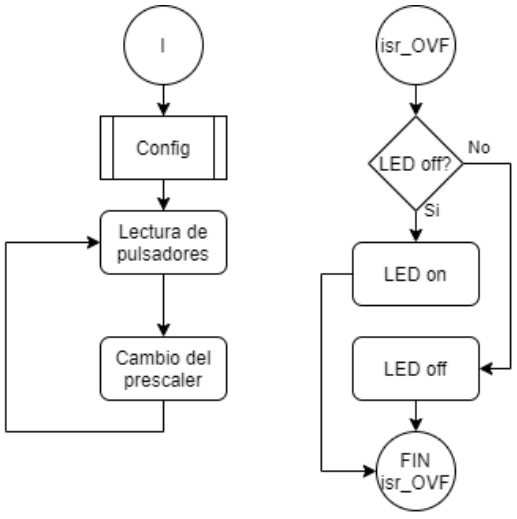



Figura 1: Esquemático del circuito

# 5. Listado de componentes

- Microcontrolador *ATmega328p* y programador USBasp (Arduino UNO) [AR$ 950]
- 1x LED [AR$ 10]
- 3x Resistencia (220 Ω) [AR$ 12]
- 1x Pulsador (10 $k\Omega$) [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

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



.DEF aux = R16

.EQU modeFIX = 0
.EQU mode64 = 1
.EQU mode256 = 2
.EQU mode1024 = 3

.CSEG

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

    .ORG OVF1addr
    JMP isr_toggle              ; interrupcion del overflow de timer

    .ORG INT_VECTORS_SIZE       ; Direccion donde escribir el codigo

conf:
```

```
    SET_SP  aux

    LDI     aux, 0x01
    OUT     DDRB, aux
    CLR     aux
    OUT     PORTB, aux
    OUT     DDRD, aux

    LDI     aux, 0      ; compare A & B disabled, WGM11/10 normal
    STS     TCCR1A, aux
    LDI     aux, 0      ; Input Capture Noise Cancel 0, Input Capture Edge 0,
    STS     TCCR1B, aux ; WGM13/12 normal, CS no clock source

    LDI     aux, 0x01   ; Input Capture Int disabled, Compares Int disabled,
    STS     TIMSK1, aux ; Overflow int enabled

    SEI

main:
    IN      aux, PIND
    ANDI    aux, 0x03
    CPI     aux, modeFIX
    BREQ    mfix
    CPI     aux, mode64
    BREQ    m64
    CPI     aux, mode256
    BREQ    m256
    CPI     aux, mode1024
    BREQ    m1024
    ;   Delay de 5 ms para evitar pulsos espurios del pulsador
    LDI     R18, 104
    LDI     R19, 229
L1: DEC     R19
    BRNE    L1
    DEC     R18
    BRNE    L1
    RJMP    main
mfix:
    LDS     aux, TCCR1B
    ANDI    aux, 0b11111000     ;mask TCCR1B
    STS     TCCR1B, aux
    SBI     PORTB, PB0
    RJMP    main
m64:
    LDS     aux, TCCR1B
    ANDI    aux, 0b11111000     ;mask TCCR1B
    ORI     aux, 0b00000011     ;CS presc 64 (011)
    STS     TCCR1B, aux
    RJMP    main
m256:
    LDS     aux, TCCR1B
    ANDI    aux, 0b11111000     ;mask TCCR1B
    ORI     aux, 0b00000100     ;CS presc 256 (100)
    STS     TCCR1B, aux
    RJMP    main
```

```
m1024:
    LDS     aux, TCCR1B
    ANDI    aux, 0b11111000     ;mask TCCR1B
    ORI     aux, 0b00000101     ;CS presc 1024 (101)
    STS     TCCR1B, aux
    RJMP    main

isr_toggle:
    SBIC    PORTB, PB0
    RJMP    led_off
    SBI     PORTB, PB0
end:
    RETI
led_off:
    CBI     PORTB, PB0
    RJMP    end
```

# 8. Resultados

Se logro controlar la frecuencia de parpadeo del LED mediante uso de timers, fue necesario cambiar las resistencias de 10 $k\Omega$ por otras de 220 $\Omega$, ya que presentaban una caída de tensión considerable y no se leían los valores de tensión esperados (`LOW`). En el caso de `PD0` (Rx), la tensión en el pin era de 4,45 $V$ cuando se esperaba una tensión de 0V, esto indica que la corriente que sale del pin es de $I_o = \frac{4,45}{10k}\ A = 445\ \mu A$. De la misma forma en `PD1` (Tx), la tensión medida fue de 1,04 $V$, con una corriente de $I_o = 104\ \mu A$. Una posible explicación luego de consultar con los profesores del curso es que **estaba siendo alimentado mediante el puerto serie** y este esta conectado al programador y directamente a Tx y Rx, y probablemente sea **necesario garantizar esa corriente en los pines**.

# 9. Conclusiones

Mediante la configuración de los timers, se pudo variar la frecuencia de parpadeo de un LED. Además se observó que al alimentar el arduino mediante USB, los pines Tx y Rx tienen comportamientos que dependen del programador (otro ATmega dentro del arduino que se encarga de programar el microcontrolador) y seria recomendable no usarlos si se quiere probar codigo mediante alimentacion USB.