



Departamento de Electrónica
Laboratorio de Microprocesadores - 86.07

Trabajo Práctico Obligatorio N°6: Timers

Profesor:			Ing. Guillermo Campiglio									
Cuatrimestre/Año:			1°/2020									
Turno de las clases prácticas			Miércoles									
Jefe de trabajos prácticos:			Ing. Pedro Ignacio Martos									
Docente guía:			Ing. Fabricio Baglivo, Ing. Fernando Pucci									
Autor			Seguimiento del proyecto									
Maximiliano	Porta	98800										

Observaciones:

.....

.....

.....

.....

.....

.....

.....

Fecha de aprobación			Firma J.T.P		

2 de septiembre de 2020

Índice

1. Objetivo	2
2. Descripción	2
2.1. Introducción	2
2.2. Funcionamiento de los Timers	3
2.3. Diagrama de Conexión del Circuito Físico	4
2.4. Esquema en Bloques	4
2.5. Materiales Utilizados	4
2.6. Diagrama de Flujo	5
2.7. Lógica del Programa	5
2.8. Links de funcionamiento del circuito	5
3. Código	6
4. Resultados y Conclusiones	8
4.1. Resultados	8
4.2. Conclusiones	8

Mediante la programación se busca poder controlar mediante la utilización de dos pulsadores y el uso de la interrupción por overflow del timer la frecuencia de oscilación de un diodo led como indica la siguiente tabla:

PD0	PD1	Estado del Led
0	0	Encendido Fijo
0	1	Parpadea con Pre-escaler CLK/64
1	0	Parpadea con Pre-escaler CLK/256
1	1	Parpadea con Pre-escaler CLK/1024

Cuadro 1: Estado del Diodo led

2.2. Funcionamiento de los Timers

Temporizador o Timer, es básicamente un registro de n-bits que se incrementa de manera automática en cada ciclo de reloj, el ATmega328p incluye 3 temporizadores, 2 son de 8 bits y 1 de 16 bits (Timer 1), también se pueden utilizar por eventos, los eventos que se pueden generar por medio de los temporizadores son: Desbordamientos, coincidencias por comparación y captura de entrada.

En este trabajo practico utilizaremos el evento por desbordamiento (Overflow) este ocurre cuando alguno de los temporizadores (TCNTn) alcanza su valor máximo (MAXVAL) y se reinicia con 0, cuando ocurre con una transición de 1's a 0's, esta transición provoca que una bandera (TOVn) sea puesta en alto, se puede configurar al hardware para que el evento produzca una interrupción.

También para este trabajo se usara el pre-escalador que es básicamente es un divisor de frecuencia que se antepone a los registros de los temporizadores proporcionándoles la capacidad de alcanzar intervalos de tiempo mayores. Un pre-escalador incluye 2 componentes, un contador de n-bits y un multiplexor para seleccionar diferentes posiciones de bit en el contador. El contador se incrementa en cada ciclo de reloj y por lo tanto, con el multiplexor pueden seleccionarse diferentes frecuencias, como puede observarse en el siguiente gráfico:

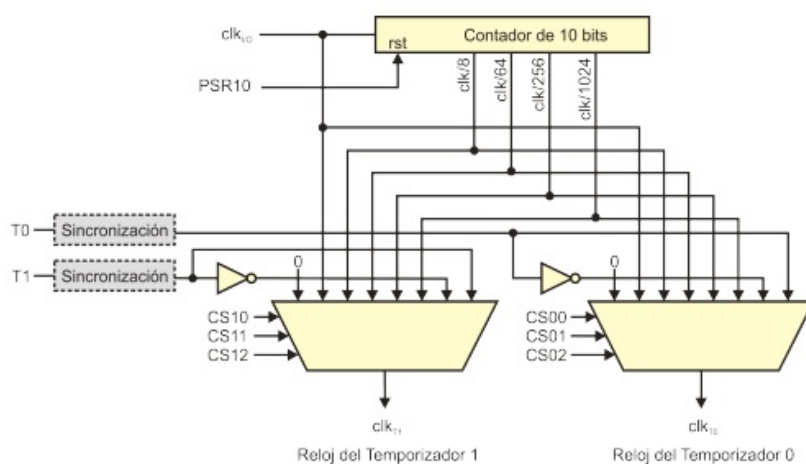


Figura 2: Diagrama del multiplexor

2.3. Diagrama de Conexión del Circuito Físico

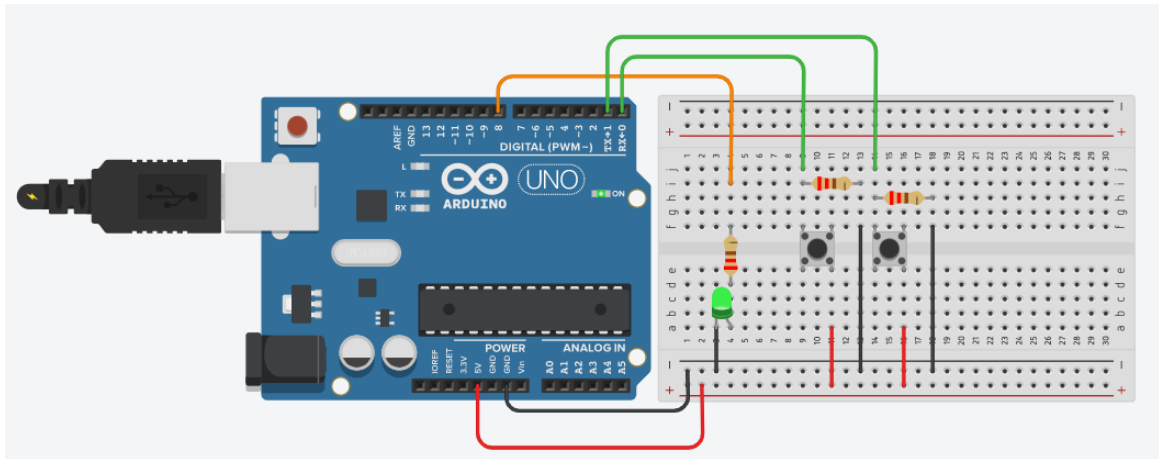


Figura 3: Diagrama del Circuito Físico

2.4. Esquema en Bloques

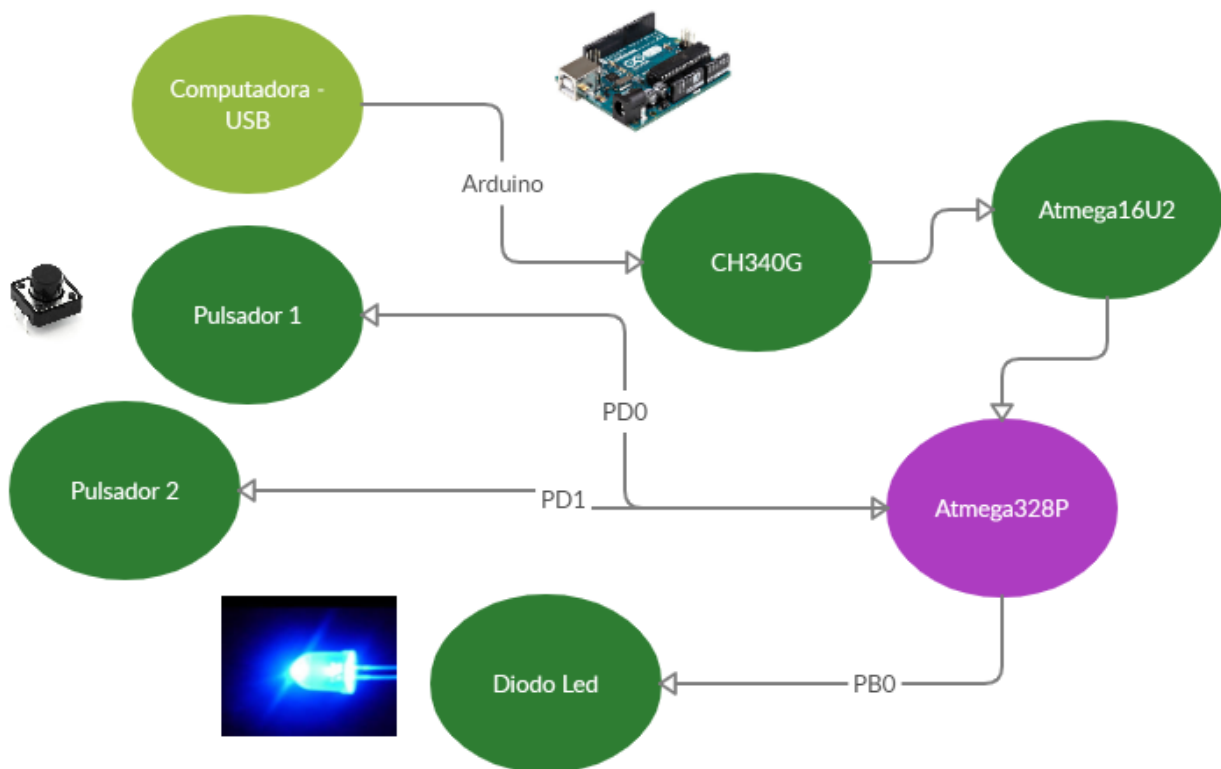


Figura 4: Diagrama en bloques

2.5. Materiales Utilizados

- x2(16 Ars)Pulsador
- x1(10 Ars)Diodo Led verde de 3mm (Alta Luminosidad).

- x3(10 Ars)Resistor de 220Ω de carbón 1/4 Watt.
- 1x(725 Ars)Placa Arduino Uno con USB.
- 1x(96.5 Ars)Placa Experimental.
- 1x(10 Ars)Cables de conexión.
- Costo del Proyecto : 870 Ars

2.6. Diagrama de Flujo

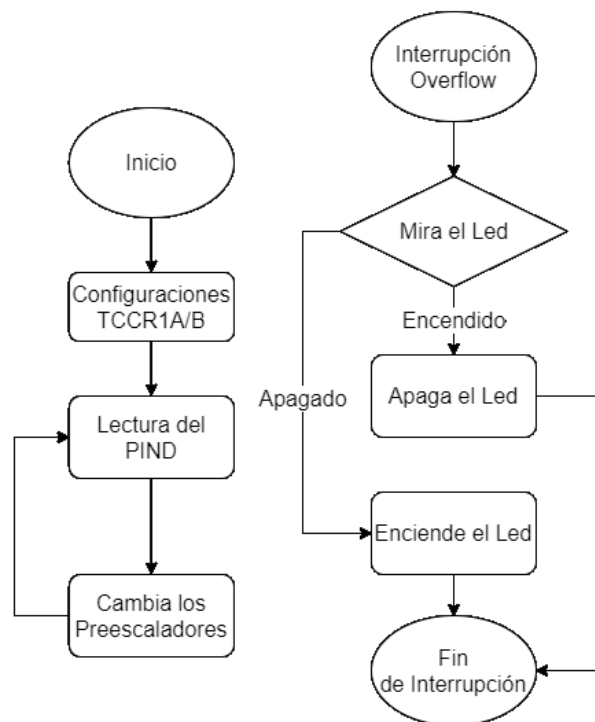


Figura 5: Diagrama de flujo

2.7. Lógica del Programa

Primero realizo todas las configuraciones de los puertos, configuro en modo normal el timer 1 poniendo en TCCR1A/B todo en 0, luego activo las interrupciones por overflow poniendo en TIMSK, luego leo el PIND y en función de la lectura ajusto el pre-escalador. En la interrupción controlo el encendido y apagado del led.

2.8. Links de funcionamiento del circuito

Timers:https://youtu.be/N_flHvqfMQM

3. Código

```
.INCLUDE "M328PDEF.INC"
.DEF aux = R16
```

```
.CSEG ;COMIENZO ESCRITURA EN ROM
.ORG 0X0000 ;INICIO DE LA ROM
    JMP MAIN
```

```
.ORG OVFladdr
    JMP INTERRUPCION
```

```
.ORG INT_VECTORS_SIZE ;52 WORDS EN ATMEGA328P
```

MAIN:

```
    LDI R16,0XFF
    OUT DDRB,R16
    LDI R16,0X00
    OUT DDRD,R16
    SBI PORTB,0 ;ARRANCA CON LED OFF
    LDI R16,0X00
    STS TCCR1A,R16 ;CONFIGURO EN MODO NORMAL
    STS TCCR1B,R16
    LDI R16,0X01
    STS TIMSK1,R16 ;ACTIVO INTERRUPCION X OVERFLOW
    LDI R16,0X00
    SEI ;ACTIVO INTERRUPCIONES
```

LOOP:

```
    IN R17,PIND
    MOV R22,R17
    CALL DELAY ;EVITA REBOTES MECANICOS (DELAY 5mS)
    IN R17,PIND ;LEO DE NUEVO LA ENTRADA
    CP R22,R17 ;COMPARA SI SE MANTIENE IGUAL
    BREQ SETEO ;SE MANTUVO IGUAL ENTONCES SETEO
    JMP LOOP
    ; SI NO SON IGUALES HUBO UN RUIDO Y VUELVO A LEER
```

SETEO:

```
    CPI R17,0X00
    BREQ CLK_OFF
    CPI R17,0X01
    BREQ CLK_64
    CPI R17,0X02
    BREQ CLK_256
    CPI R17,0X03
    BREQ CLK_1024
    JMP LOOP ; SI SE INGRESA ALGO NO VALIDO VUELVE A LEER
```

CLK_OFF:

```
LDI R16, 0X00
STS TCCR1B, R16      ;APAGO EL CLOCK
SBI PORTB, 0         ;ENCIENDE EL LED
JMP LOOP
```

CLK_64:

```
LDI R16, 0X03
STS TCCR1B, R20
JMP LOOP
```

CLK_256:

```
LDI R20, 0X04
STS TCCR1B, R20
JMP LOOP
```

CLK_1024:

```
LDI R20, 0X05
STS TCCR1B, R20
JMP LOOP
```

INTERRUPCION:

```
SBIC PORTB, 0      ;MIRO EL ESTADO DEL LED
JMP APAGO
SBI PORTB, 0
RETI
```

APAGO:

```
CBI PORTB, 0
RETI
```

DELAY: ;CONFIGURO EL TIEMPO DE ESPERA

```
LDI R20, 1
t3: LDI R19, 80
t2: LDI R18, 250
t1: NOP
```

```
DEC R18      ; 250 x 250ns = 62.5us
BRNE t1
```

```
DEC R19      ; 80 x 62.5us = 5ms
BRNE t2
```

```
DEC R20      ; 1 x 5ms = 5ms
BRNE t3
```

```
RET
```


4. Resultados y Conclusiones

4.1. Resultados

Se logro controlar el parpadeo del diodo led conectado al puerto b del microcontrolador, en función de las entradas en el puerto d 0 y 1.

4.2. Conclusiones

Aprendí como funcionan los timers del Atmega328p, también a revisar los valores de las resistencias, por que no funcionaba en un principio por que con 10K no se podían leer bien los valores por que caía mucha tensión así que use resistores de 1k.