



Laboratorio de Microprocesadores - 86.07

Trabajo Práctico Obligatorio N°1

| | | | | | | | | | | |
|-------------------------------|----------|---------|--|--|--|--|--|--|--|--|
| Profesor: | | | Ing. Guillermo Campiglio | | | | | | | |
| Cuatrimestre/Año: | | | 1º/2020 | | | | | | | |
| Turno de las clases prácticas | | | Miércoles | | | | | | | |
| Jefe de trabajos prácticos: | | | Ing. Pedro Ignacio Martos | | | | | | | |
| Docente guía: | | | Ing. Fabricio Baglivo, Ing. Fernando Pucci | | | | | | | |
| | | | | | | | | | | |
| Autores | | | Seguimiento del proyecto | | | | | | | |
| Nombre | Apellido | Padrón | | | | | | | | |
| Ivan Eric | Rubin | 100 577 | | | | | | | | |

Observaciones:

.....

.....

.....

.....

.....

.....

.....

.....

.....

| | | | | |
|---------------------|--|--|--|-------------|
| Fecha de aprobación | | | | Firma J.T.P |
| | | | | |

| | |
|----------------|--|
| Coloquio | |
| Nota final | |
| Firma profesor | |

1. Objetivo del trabajo

El objetivo del presente trabajo es hacer *titilar* un LED (Diodo Emisor de Luz) controlando la entrada y salida de un pin del microcontrolador. Se realizará de dos formas; una utilizando todo un puerto, y otra con un solo bit del puerto.

2. Descripción del trabajo

Se conectará un LED con una resistencia en serie a un pin del microprocesador para encenderlo y apagarlo utilizando instrucciones codificadas en el lenguaje Assembler. Para ello se deberá establecer determinado puerto como puerto de salida. Luego se utilizarán registros para encender el LED, generar un retardo, apagar el LED y generar el retardo nuevamente.

3. Diagrama de conexiones en bloques

En la figura 3.1 se puede ver el diagrama en bloques del sistema a realizar.

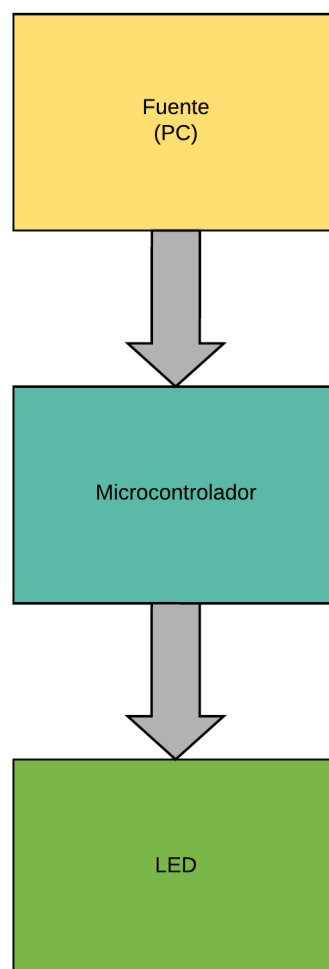


Figura 3.1: Diagrama en bloques

4. Circuito Esquemático

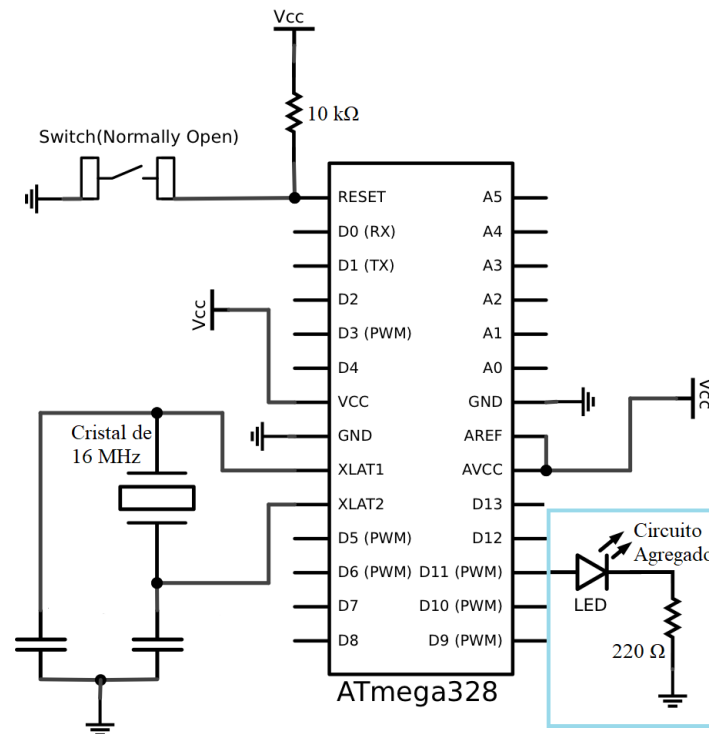


Figura 4.1: Circuito esquemático

En la figura 4.1 se puede ver un esquema que representa las conexiones realizadas para montar el circuito. Si bien el LED no está directamente conectado al pin del Atmega 328P se decidió representar la conexión de esta manera. En realidad el led se conecta al Arduino que se ocupa de la conexión con el microprocesador.

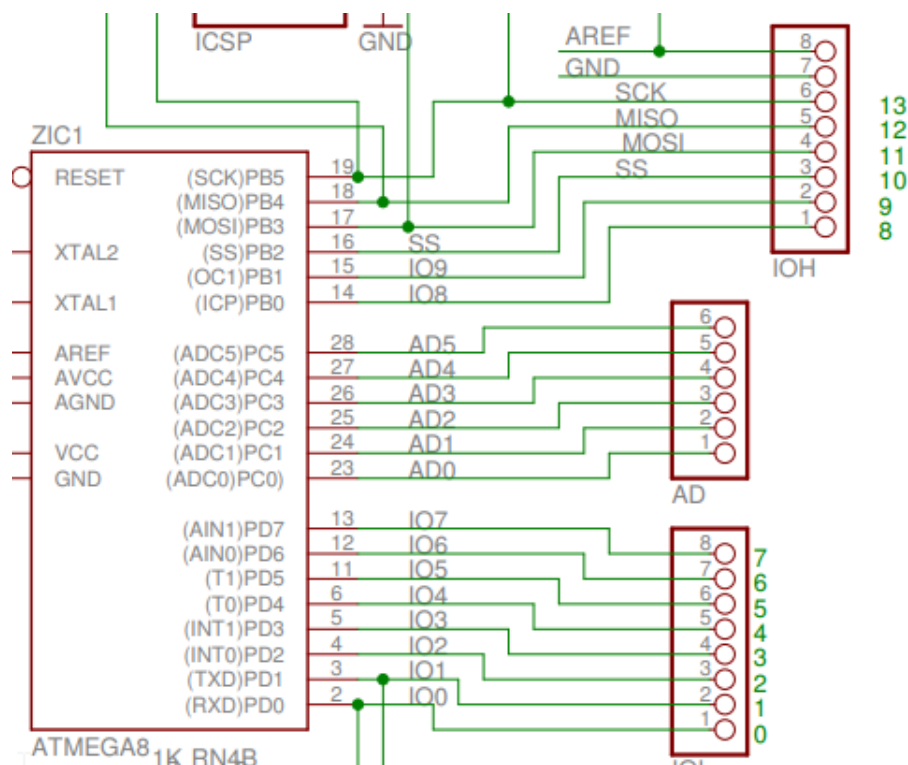


Figura 4.2: Esquemático Arduino

En la figura 4.2 se ve la conexión entre el puerto B del micro y los pines externos del Arduino Uno (Arriba a la derecha). Para el caso en que se utiliza todo el puerto se podrá conectar el LED a cualquiera de los pines 8 a 13.

5. Listado de componentes

- Arduino UNO.
- Microprocesador ATMEGA 328P (Integrado en Arduino).
- Resistencia de $220\Omega \pm 1\%$.
- Diodo Emisor de Luz (LED).
- Cables Macho-Macho.
- Protoboard.
- Conector Arduino-PC.

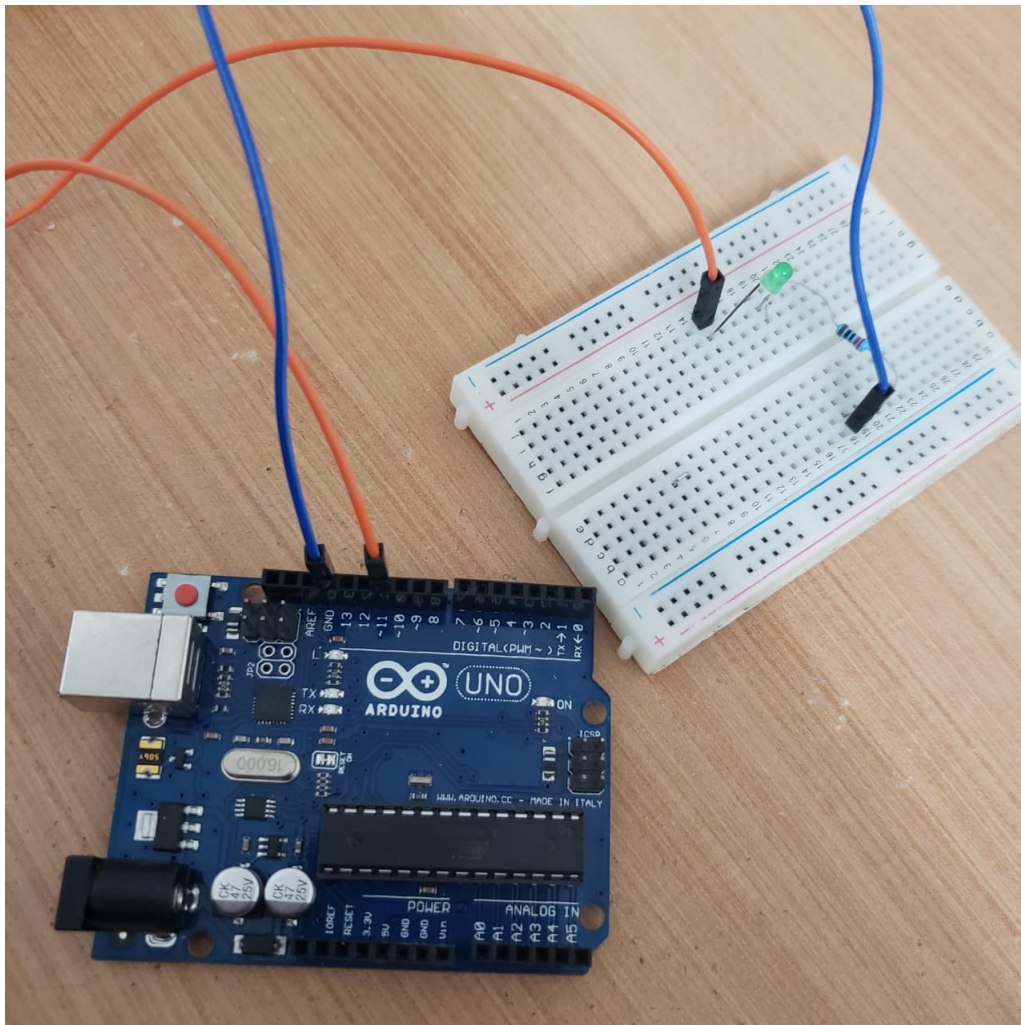


Figura 5.1: Componentes

6. Diagrama de flujo del software

Se presenta el diagrama de flujo del trabajo en la figura 6.1.

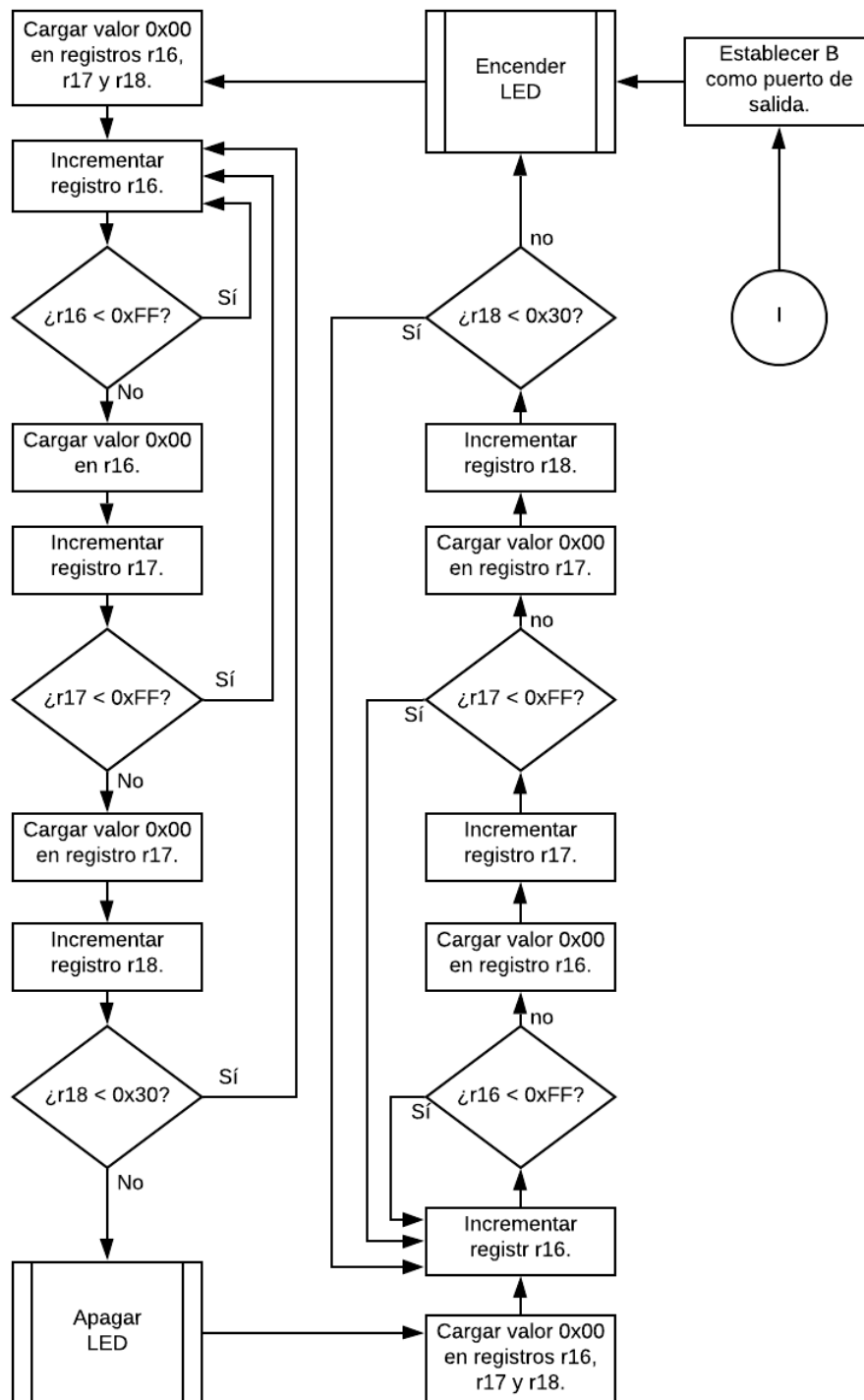


Figura 6.1: Diagrama de flujo.

7. Código de programa

```

1  .include "m328pdef.inc"
2
3  .cseg ;A partir de aquí hay código.
4  .org 0x0000 ;Voy a la posición de memoria 0x0000 (Donde se encuentra RESET).
5      jmp      main    ;Sobre-escribo RESET con un salto a main.
6
7  .org INT_VECTORS_SIZE ;Salteo la memoria reservada para periféricos.
8  main:
9
10 ; Conecto LED en el pin 11 de ARDUINO UNO (Pin B3).
11 ; Configuro puerto B.
12     ldi      r16,0xff ;(PORTB como salida).
13     out      DDRB,r16
14
15 ; Rutina de encendido y apagado.
16
17 prendo:      sbi      PORTB,3    ; Se enciende el LED.
18                ; Poniendo un 1 lógico (Que se traduce a 5 Volts) en el Pin
19                3 del puerto B.
20
21 demora1:
22     ldi      r16,0x00
23     ldi      r17,0x00
24     ldi      r18,0x00 ;Se setean los registros en cero.
25 ciclo1:      inc      r16          ;Se incrementa uno de ellos.
26     cpi      r16,0xff ;Se lo compara con el valor 0xff (255 en decimal).
27     brlo     ciclo1      ;Si la comparación es verdadera, vuelve a realizar el
28     ciclo.
29     ldi      r16,0x00 ;Si es falsa, setea el registro incrementado a 0x00 y
30     realiza el ciclo1
31     inc      r17          ;varias veces hasta que se cumpla la siguiente
32     condición y asi.
33     cpi      r17,0xff
34     brlo     ciclo1
35     ldi      r17,0x00
36     inc      r18
37     cpi      r18,0x30
38     brlo     ciclo1
39
40     cbi      PORTB,3      ; Se apaga el LED.
41
42 demora2:
43     ldi      r16,0x00 ;Se realiza otro delay con la lógica anterior
44     ldi      r17,0x00
45     ldi      r18,0x00
46 ciclo2:      inc      r16
47     cpi      r16,0xff
48     brlo     ciclo2
49     ldi      r16,0x00
50     inc      r17
51     cpi      r17,0xff
52     brlo     ciclo2
53     ldi      r17,0x00
54     inc      r18
55     cpi      r18,0x30
56     brlo     ciclo2
57
58     RJMP     prendo      ; Se reinicia el ciclo.

```

```

1  .include "m328pdef.inc"
2
3  .cseg ;A partir de aquí hay código.
4  .org 0x0000 ;Voy a la posición de memoria 0x0000 (Donde se encuentra RESET).
5      jmp      main ;Sobre-escribo RESET con un salto a main.
6
7  .org INT_VECTORS_SIZE ;Salteo la memoria reservada para periféricos.
8  main:
9
10 ; Conecto LED en el pin 8, 9, 10, 11, 12 o 13 de ARDUINO UNO.
11 ; Configuro puerto B.
12     ldi      r19,0xff ;Cargo un registro con unos para prender el LED en cada
        ciclo.
13     ldi      r16,0xff ;(PORTB como salida).
14     out      DDRB,r16
15
16 ; Rutina de encendido y apagado.
17
18 prendo: out      PORTB,r19 ; Se enciende el LED.
19             ; Poniendo un 1 lógico (Que se traduce a 5 Volts) en todo el
        puerto B (pins 8 a 13 del ARDUINO).
20
21 demora1:
22     ldi      r16,0x00
23     ldi      r17,0x00
24     ldi      r18,0x00 ;Se setean los registros en cero.
25 ciclo1: inc      r16 ;Se incrementa uno de ellos.
26     cpi      r16,0xff ;Se lo compara con el valor 0xff (255 en decimal).
27     brlo     ciclo1 ;Si la comparación es verdadera, vuelve a realizar el
        ciclo.
28     ldi      r16,0x00 ;Si es falsa, setea el registro incrementado a 0x00 y
        realiza el ciclo1
29     inc      r17 ;varias veces hasta que se cumpla la siguiente
        condición y así.
30     cpi      r17,0xff
31     brlo     ciclo1
32     ldi      r17,0x00
33     inc      r18
34     cpi      r18,0x30
35     brlo     ciclo1
36
37     out      PORTB,r16 ; Se apaga el LED.
38
39 demora2:
40     ldi      r16,0x00 ;Se realiza otro delay con la lógica anterior
41     ldi      r17,0x00
42     ldi      r18,0x00
43 ciclo2: inc      r16
44     cpi      r16,0xff
45     brlo     ciclo2
46     ldi      r16,0x00
47     inc      r17
48     cpi      r17,0xff
49     brlo     ciclo2
50     ldi      r17,0x00
51     inc      r18
52     cpi      r18,0x30
53     brlo     ciclo2
54
55
56     RJMP     prendo ; Se reinicia el ciclo.

```

En la página 5 del presente informe está el código utilizado para prender y apagar el LED utilizando un solo pin del puerto B. En la página 6 se muestra el código que utiliza todo el puerto. Adjunto al trabajo también estarán los archivos .asm de los códigos mostrados.

8. Resultados

Se logró controlar un LED manejando los pines presentes en uno de los puertos del microprocesador. También se calculó el tiempo de encendido contando la cantidad de ciclos y teniendo en cuenta la frecuencia que presentá el cristal interior del Arduino UNO que utiliza el Atmega 328P.

La cantidad de ciclos se obtuvo utilizando la herramienta de *debuggeo* del programa AVR Studio. Se obtuvo que el *delay* aplicado entre que se prende el led y se apaga es de 12533956 ciclos de programa. Como la frecuencia del microprocesador es de 16MHz el tiempo de *delay* en segundos es de aproximadamente 0,78s.

9. Conclusiones

Como conclusión del trabajo es importante destacar que el tipo de *delay* implementado no es el mas óptimo ya que se necesitan muchas líneas de código, uso de registros e instrucciones para lograr el cometido. Sin embargo, la realizado sirvió como una buena introducción al manejo de microprocesadores y al lenguaje Assembler.