



UNIVERSIDAD DE BUENOS AIRES
FACULTAD DE INGENIERÍA

Laboratorio de Microprocesadores (86.07)

Trabajo Practico N°1

Integrantes:

Puy Gonzalo 99784 gpuy@fi.uba.ar

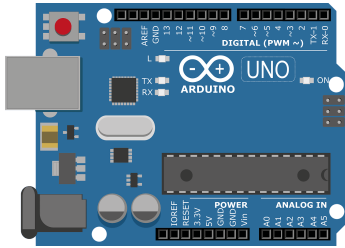
1. Introducción

El objetivo del presente trabajo es servir como introducción a la programación de microcontroladores. Esta centrado, especialmente, en el manejo de los puertos del microcontrolador y la utilidad de la resistencia interna de pull-up.

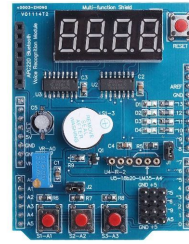
2. Desarrollo

Para la realización de este trabajo se utilizó el siguiente banco de mediciones:

- Placa de desarrollo Arduino Uno y su respectivo cable para conectar la placa a la PC.
- Shield Multifunción para la placa Arduino.



(a) Placa de desarrollo Arduino UNO.



(b) Shield multifunción.

Figura 1: Banco de mediciones.

2.1. Encendido del Led

En primer lugar, se realizó el encendido del Led “D4”, ubicado en el pin “~10” del shield multifunción. Este pin es equivalente en la placa Arduino y es el “PB2” del microcontrolador “ATMega 328P”. Primero, se configura el puerto B como salida colocando un “0” lógico en el bit deseado de DDRB para luego enviar un “0” lógico por medio de PORTB. Una vez encendido el Led, quedara titilando permanentemente con un delay de aproximadamente 1 segundo.

Cabe aclarar, que los Leds del shield multifunción, se encienden con un “0” lógico. Esto es debido a la conexión dada por el mismo shield. A continuación se muestra un esquema simplificado de la conexión.

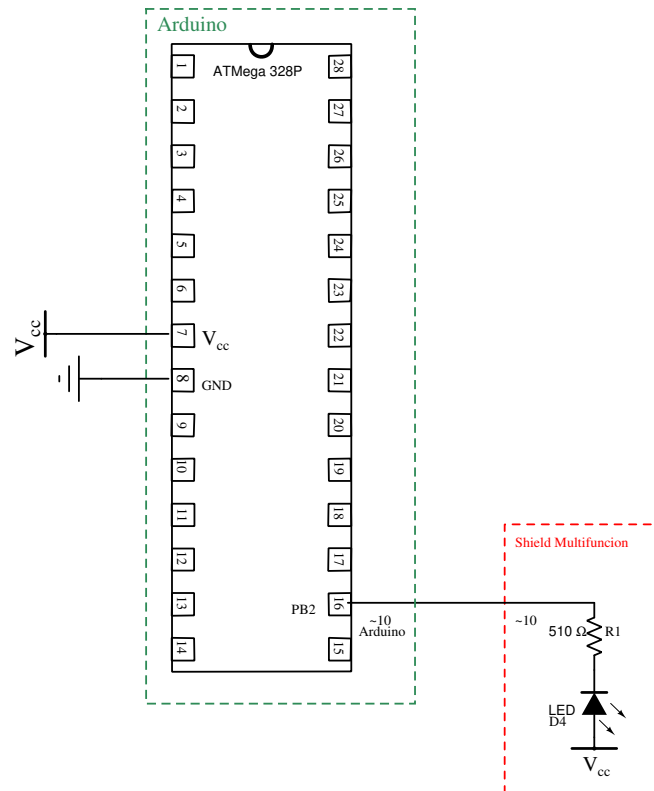


Figura 2: Diagrama simplificado de la conexión.

Al observar la figura 2, se nota que aplicando un “0” lógico en PORTB2, el Led queda en directa y por ende, se encenderá. Caso contrario, si se aplica un “1” lógico en PORTB2, el Led estará en inversa y no se encenderá.

Finalmente se muestra el diagrama de flujo que muestra la lógica del programa y se adjunta el código utilizado para esta etapa del trabajo.

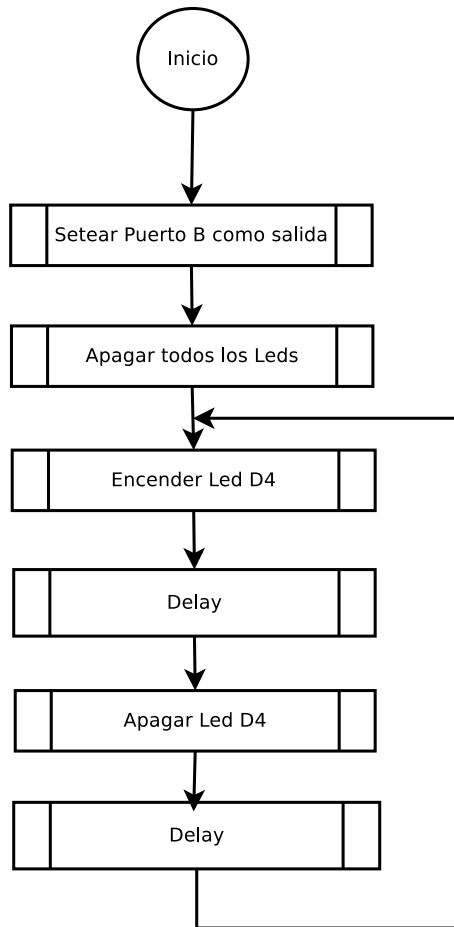


Figura 3: Diagrama de flujo para el encendido del led

```

1  ;
2  ; TP01 - Laboratorio de Microprocesadores (86.07)
3  ;
4  ; Created: 15/5/2021 17:13:43
5  ; Author : Puy Gonzalo
6  ; Padron : 99784
7
8  ;Codigo para punto 1) del TP - 01
9
10 .include "m328pdef.inc"
11
12 .def          aux1=r16
13 .def          aux2=r17
14
15 .cseg
16 .org 0x0000
17             jmp      main
18
19 .org INT_VECTORS_SIZE
20
21
22 main:
23

```

```

24 ; Configuro puerto
25
26         ldi        aux1,0xff
27         out        DDRB,aux1    ;Puerto B como salida
28
29 ;Seteo PORTB para que esten todos en 1.
30 ;De esta forma los leds empiezan apagados
31 ;Debo hacer esto debido al funcionamiento del shield.
32         out        PORTB,aux1
33
34 ciclo:
35         cbi        PORTB,2      ;encendio del led
36         call       delay_1s     ;llamo a delay
37         sbi        PORTB,2      ;apago led
38         call       delay_1s     ;llamo de nuevo a delay
39         jmp        ciclo        ;Comienzo el ciclo de nuevo
40
41 ; Subrutina de Delay
42
43 delay_1s:
44         ldi        r18, 102
45         ldi        r19, 118
46         ldi        r20, 194
47 L1:
48         dec        r20
49         brne       L1
50         dec        r19
51         brne       L1
52         dec        r18
53         brne       L1
54         ret

```

Listing 1: Código utilizado para encendido de Led

2.2. Encendido y apagado del Led mediante swithes

En esta etapa, se volvió a encender el Led “D4” del shield multifunción. Esta vez el Led no empezara a titilar instantáneamente, sino que se encenderá mediante la pulsación switch “S1”. Luego, se usara el switch “S2” para apagar “D4”. En esta ocasión, se utilizo un delay de 50 ms.

Para poder lograr esto, se configuro al puerto C como entrada, esto es, colocando un “0” lógico en los bits deseados de DDRC (si bien, los bits de DDRC comienzan por *default* en “0”, se configuro de todas formas en el código). Luego, se leerá el PINC para verificar si se tiene que prender (o no) el Led, donde la lógica para lograrlo es análoga a la etapa anterior.

A continuación se muestra un esquema simplificado de la conexión utilizada:

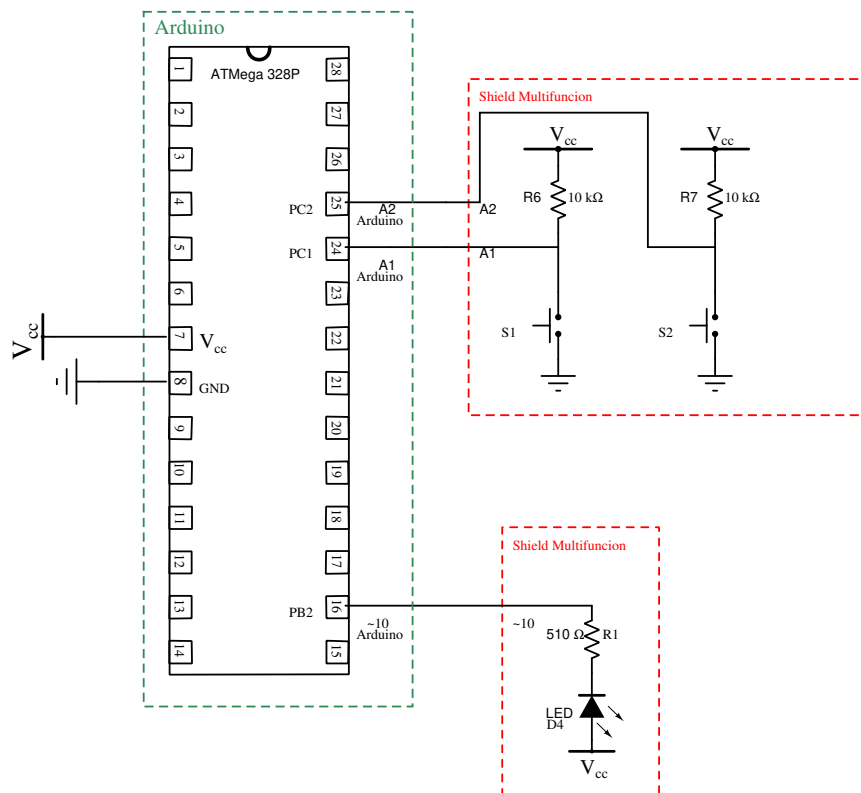


Figura 4: Diagrama simplificado de la conexión

Como se puede observar, el shield multifunción tiene una conexión de tipo pull-up con el switch. Esto significa que mientras el switch no este pulsado, se forzara un “1” lógico en el puerto. Sucediendo lo inverso cuando se presiona el switch.

Por ultimo, se adjunta el correspondiente diagrama de flujo y el código utilizado

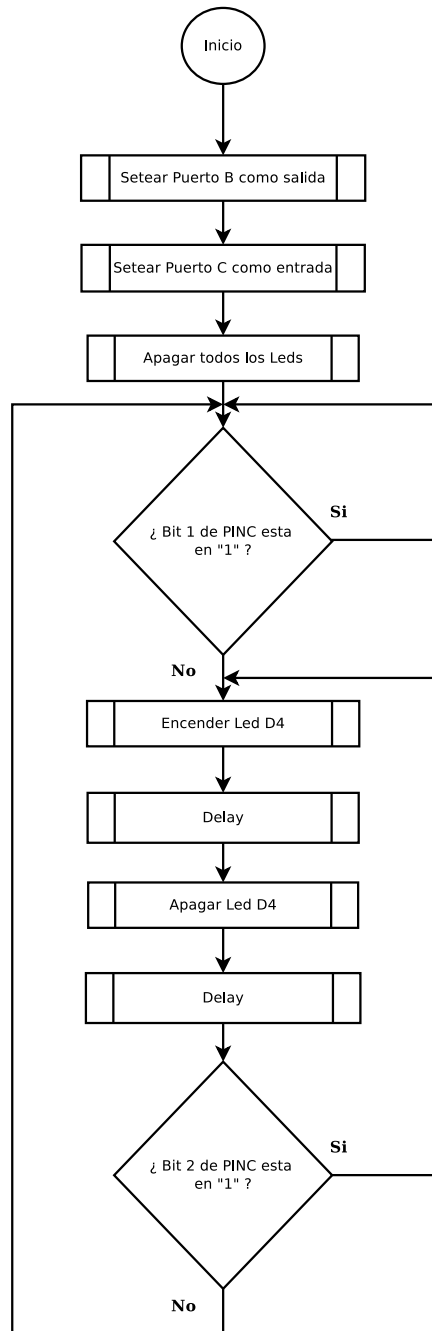


Figura 5: Diagrama de flujo para el encendido y apagado del led mediante pulsadores

```

1 ;
2 ; TP01 - Laboratorio de Microprocesadores (86.07)
3 ;
4 ; Created: 15/5/2021 17:13:43
5 ; Author : Puy Gonzalo
6 ; Padron : 99784
7
8 ;Codigo para punto 2) del TP - 01
9
10 .include "m328pdef.inc"
11
12 .def          aux1=r16

```



```

13 .def          aux2=r17
14
15 .cseg
16 .org 0x0000
17             jmp          main
18
19 .org INT_VECTORS_SIZE
20
21
22 main:
23
24 ; Configuro puertos
25
26             ldi          aux1,0xff
27             out          DDRB,aux1    ;Puerto B como salida
28             ldi          aux2,0x00    ;
29             out          DDRC,aux2    ;Puerto C como entrada
30
31 ;Seteo PORTB para que esten todos en 1.
32 ;De esta forma los leds empiezan apagados
33 ;Debo hacer esto debido al funcionamiento del shield.
34             out          PORTB,aux1
35
36 ; Subrutina de Led apagado
37 loopapagado:
38             sbis         PINC,1        ;Skipto la siguiente linea si el
39             bit 1 de PINC esta en "1"
40             jmp          ciclo
41             jmp          loopapagado
42
43 ; Subrutina de parpadeo del led
44 ciclo:
45             cbi          PORTB,2      ;encendio del led
46             call         delay_50ms   ;llamo a delay
47             sbi          PORTB,2      ;apago led
48             call         delay_50ms   ;llamo de nuevo a delay
49             sbis         PINC,2      ;Skipto la siguiente linea si el
50             bit 2 de PINC esta en "1"
51             jmp          loopapagado  ;Vuelvo al loop del Led apagado
52             jmp          ciclo        ;Vuelvo a comenzar el ciclo
53
54 ; Subrutina de Delay
55
56 delay_50ms:
57             ldi          r20, 5
58             ldi          r21, 15
59             ldi          r22, 242
60 L1: dec      r22
61             brne         L1
62             dec          r21
63             brne         L1
64             dec          r20

```

```
63  brne L1
64  ret
```

Listing 2: Código utilizado para el encendido y apagado del led mediante pulsadores

2.3. Resistencia interna de pull-up

En esta sección se analizará el comportamiento del circuito cuando se conecta la resistencia interna de pull-up.

Al haber colocado el puerto C como entrada en la sección anterior, se tiene la opción de activar la resistencia de pull-up interna colocando un “1” lógico en el bit correspondiente de PORTC. Esto se puede lograr añadiendo las siguientes líneas al código utilizado en la etapa anterior

- `ldi r16,0xff`
- `out PORTC,r16`

Una vez activada la resistencia de pull-up, la conexión vista en la figura 4 cambiará. En este simple esquema mostrado a continuación, se puede observar el cambio en la conexión de un solo switch, la otra conexión es análoga.

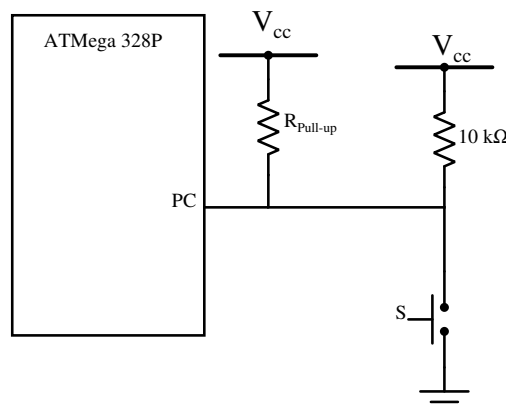


Figura 6: Esquema de conexión con resistencia de Pull-up activada

Como se puede observar en el esquema, el circuito queda equivalente al circuito externo (con el switch). Esto demuestra que al activar la resistencia interna de pull-up, esta queda en paralelo con la resistencia de $10\text{ k}\Omega$. Esto hace que la resistencia equivalente sea menor que $10\text{ k}\Omega$, ya que según la [hoja de datos](#) del ATmega 328P, esta resistencia tiene un valor de $20\text{ k}\Omega < R_{PU} < 50\text{ k}\Omega$.

También, al quedar el mismo circuito, seguiremos teniendo el caso de la sección anterior, donde se forzara un “1” lógico cuando el switch este sin presionar y un “0” lógico cuando se presiona. Esto nos dice que en teoría, se puede despreciar la resistencia externa de $10\text{ k}\Omega$ (claramente no es posible hacer esto, ya que la resistencia se encuentra integrada en el shield multifunción).

De todas formas, el shield multifunción cuenta con un jumper, con el cual se puede desconectar las resistencias conectadas a los pulsadores. De esta forma, cuando se activan las resistencias internas de pull-up la conexión queda como la vista en la figura 4. Por lo tanto el programa funciona correctamente y el comportamiento del circuito es el esperado.

Como se menciono anteriormente, solo hay que hacer un pequeño cambio en el código de la sección anterior para utilizar la resistencia de pull-up en los puertos. Se adjunta el código resultante a continuación.

```

1 ;
2 ; TP01 - Laboratorio de Microprocesadores (86.07)
3 ;
4 ; Created: 15/5/2021 17:13:43
5 ; Author : Puy Gonzalo
6 ; Padron : 99784
7
8 ;Codigo para punto 3) del TP - 01
9
10 .include "m328pdef.inc"
11
12 .def          aux1=r16
13 .def          aux2=r17
14
15 .cseg
16 .org 0x0000
17             jmp      main
18
19 .org INT_VECTORS_SIZE
20
21
22 main:
23
24 ; Configuro puertos
25
26             ldi      aux1,0xff
27             out      DDRB,aux1 ;Puerto B como salida
28             ldi      aux2,0x00 ;
29             out      DDRC,aux2 ;Puerto C como entrada
30
31 ;Seteo PORTB para que esten todos en 1.
32 ;De esta forma los leds empiezan apagados
33 ;Debo hacer esto debido al funcionamiento del shield.
34             out      PORTB,aux1
35             out      PORTC,aux1 ;Resistencia de Pull-up activada
36
37 ; Subrutina de Led apagado
38 loopapagado:
39             sbis     PINC,1      ;Skipto 1 siguiente linea si el
              bit 1 de PINC esta en "1"
40             jmp      ciclo
41             jmp      loopapagado
42
43 ; Subrutina de parpadeo del led
44 ciclo:
45             cbi      PORTB,2     ;encendio del led
46             call     delay_50ms  ;llamo a delay
47             sbi      PORTB,2     ;apago led

```

```

48         call    delay_50ms    ;llamo de nuevo a delay
49         sbis    PINC,2        ;Skipeo la siguiente linea si el
        bit 2 de PINC esta en "1"
50         jmp     loopapagado    ;Vuelvo al loop del Led apagado
51         jmp     ciclo          ;Vuelvo a comenzar el ciclo
52
53 ; Subrutina de Delay
54
55 delay_50ms:
56     ldi    r20, 5
57     ldi    r21, 15
58     ldi    r22, 242
59 L1:    dec    r22
60         brne   L1
61         dec    r21
62         brne   L1
63         dec    r20
64         brne   L1
65     ret

```

Listing 3: Código modificado para la Resistencia de pull-up

3. Conclusiones

En conclusión, se puede decir que esta experiencia resulto muy positiva como introducción a la programación de microcontroladores. Además, se obtuvo un conocimiento básico sobre el manejo de puertos y la parte digital del circuito interno del microcontrolador. Por ultimo se pudo ver la utilidad de la resistencia interna de pull-up que permite, por ejemplo en este caso, reducir componentes del sistema de trabajo.