



Laboratorio de Microprocesadores - 86.07

Trabajo Práctico N°2:

Manejo de Puertos

Profesor:			Ing. Guillermo Campiglio									
Cuatrimestre/Año:			1º/2020									
Turno de las clases prácticas			Miércoles									
Jefe de trabajos prácticos:			Ing. Pedro Ignacio Martos									
Docentes guía:			Ing. Fabricio Baglivo, Ing. Fernando Pucci									
Fecha de presentación:			27/05/2020									
Alumno			Seguimiento del proyecto									
Nombre	Apellido	Padrón										
Maximiliano Daniel	Reigada	100565										

Observaciones:

.....

.....

.....

.....

.....

.....

.....

.....

Fecha de aprobación			Firma J.T.P

Índice

1. Objetivo	1
2. Descripción	1
3. Diagrama de conexiones en bloques	1
4. Circuito esquemático	2
5. Listado de componentes	2
6. Diagrama de flujo del Software	3
7. Código de programa	4
8. Resultados	4
9. Conclusiones	5

1. Objetivo

El presente trabajo práctico tiene como objetivo poner en práctica el uso de los registros de los puertos de entrada/salida en un microcontrolador, y ver la utilidad de las resistencias pull-up internas que se encuentran conectadas a los pines de este.

2. Descripción

Mediante la implementación de una placa de microcontrolador de código abierto basado en el microchip ATmega328P, denominada Arduino Uno, se busca controlar el encendido/apagado de un LED a partir del uso de un pulsador.

Se procede a conectar el LED al pin 2 del puerto B del microcontrolador, colocando un resistor en serie para limitar la corriente de este. A su vez, se conecta el pulsador entre V_{CC} y el pin 0 de dicho puerto, empleando un resistor a modo de pull-down, para establecer un estado lógico bajo en el pin cuando este se encuentre en reposo.

La idea principal de este trabajo radica en que el LED se mantenga encendido mientras el pulsador se encuentra presionado, y permanezca apagado cuando esto no suceda. Como práctica adicional, se pretende también analizar las modificaciones a implementar tanto en programación como en circuito, para hacer uso de la resistencia pull-up interna de los puertos del microcontrolador.

3. Diagrama de conexiones en bloques

A continuación, puede verse el diagrama de conexiones en bloques correspondiente a este trabajo práctico:

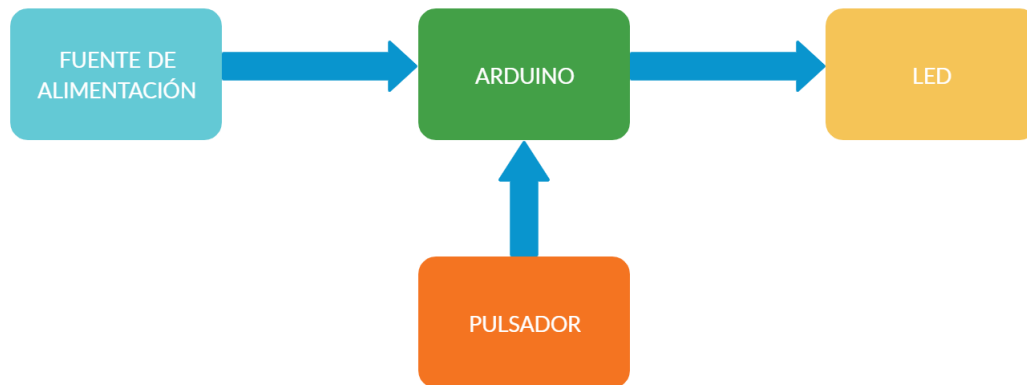


Figura 1: Diagrama de conexiones en bloques.

4. Circuito esquemático

Se procede a realizar el esquemático del circuito a implementar. Cabe destacar que, si bien en la realidad los componentes se encuentran conectado al Arduino Uno y este a su vez a una fuente de tensión externa, resulta más útil representar en el esquema solo los componentes electrónicos esenciales del sistema, conectados a referencias de tensión de un determinado valor:

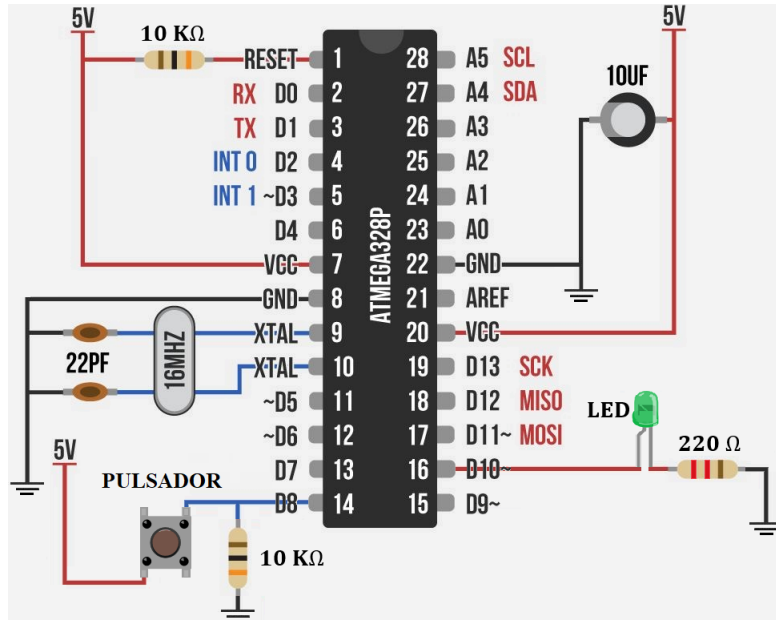


Figura 2: Circuito esquemático.

5. Listado de componentes

En el desarrollo de este trabajo práctico se utilizan los siguientes componentes:

- 1 LED verde de 5mm
- 1 Pulsador Dip Tact Switch 6x6x5mm
- 1 Resistor de 220 Ω
- 1 Resistor de 10 $k\Omega$
- 4 Cables de conexión
- 1 Protoboard
- 1 Arduino Uno (con microcontrolador ATmega328P incluido)

6. Diagrama de flujo del Software

A continuación, se explicita el diagrama de flujo correspondiente al código desarrollado:

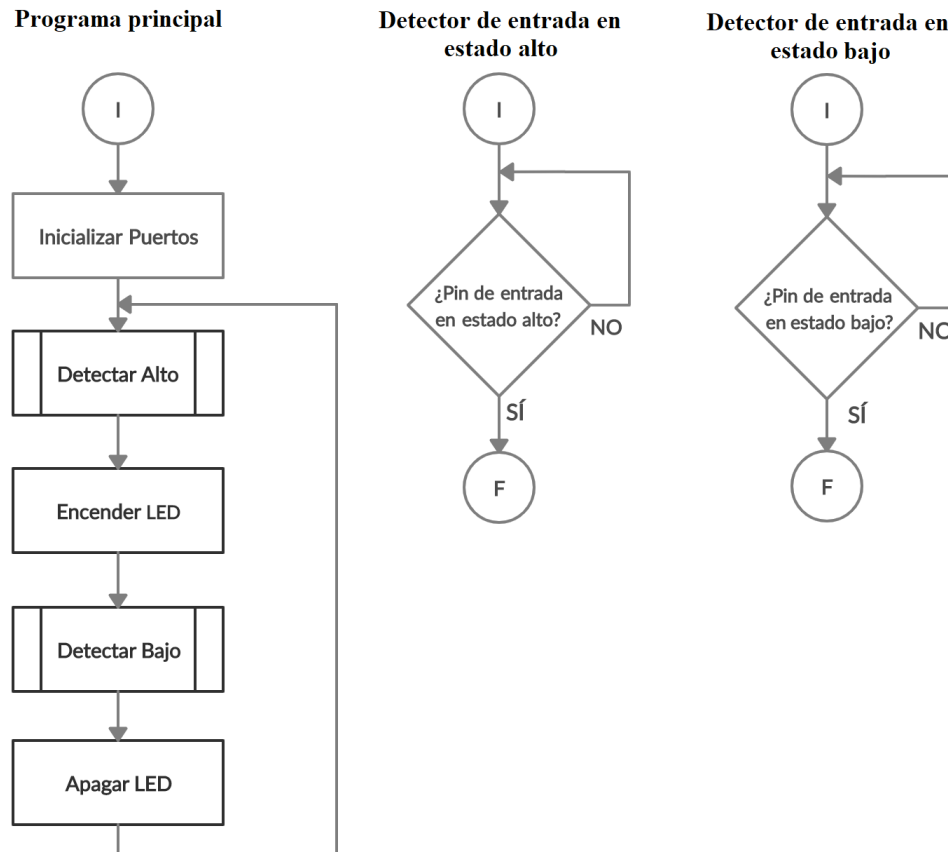


Figura 3: Diagrama de flujo.

7. Código de programa

A continuación, puede verse el código desarrollado para este proyecto:

```

1  .INCLUDE "M328PDEF.INC"
2
3  .EQU PUERTO_SALIDA = PORTB
4  .EQU PUERTO_ENTRADA = PINB
5  .EQU CONF_PUERTO = DDRB
6  .EQU BOTON = 0
7  .EQU LED = 2
8
9  .CSEG
10 .ORG 0x0000
11     JMP MAIN
12
13 .ORG INT_VECTORS_SIZE
14
15 MAIN:
16
17     LDI R18, 0x24
18     OUT CONF_PUERTO, R18
19
20     ; Se inicializan puertos
21
22     CONTROLAR_LED:
23
24     CALL DETECTAR_ALTO
25     SBI PUERTO_SALIDA, LED
26     CALL DETECTAR_BAJA
27     CBI PUERTO_SALIDA, LED
28     JMP CONTROLAR_LED
29
30     ; Se espera hasta que el botón en el puerto de entrada sea presionado
31     ; Se espera hasta que el botón en el puerto de entrada deje de estar presionado
32     ; Se reinicia el ciclo de control
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99

```

Cabe destacar, que si bien en este informe no se adjunta el programa implementado para el uso de las resistencias pull-up internas del microcontrolador, el mismo se encuentra entre los archivos con los que se entrega este trabajo.

8. Resultados

Luego de desarrollar el código descrito en las secciones anteriores y armar el circuito dispuesto, se logra controlar el encendido y apagado del led mediante el accionar y reposo del pulsador dispuesto en uno de los puertos de entrada del microcontrolador.

Por otra parte, se consigue analizar que para utilizar la resistencia pull-up interna del puerto de entrada, se debe modificar el circuito de la siguiente manera:

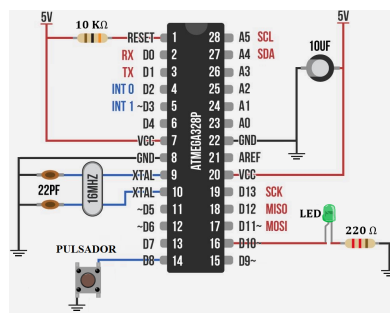


Figura 4: Circuito esquemático usando resistencia interna pull-up.

De esta forma ya no es necesario utilizar el resistor de $10k\Omega$ externo, sin embargo, se debe

modificar el programa. Para empezar, luego de indicar que pines se utilizan como entradas y salidas, se debe poner un 1 en el bit 0 del registro PORTB para así poder habilitar la resistencia interna. Por otro lado, como ahora se trata con una conexión pull-up, se debe invertir la lógica utilizada en el ejercicio anterior, ya que cuando se presiona el botón, en el pin de entrada se tiene un nivel lógico bajo, y de lo contrario se tiene un nivel lógico alto.

Finalmente, se procede a realizar un listado de los componentes utilizados y sus costos:

Componentes	Costos
Led verde de 5mm	\$7,00
Pulsador Dip Tact Switch	\$22,00
Resistor de 220 Ω	\$5,50
Resistor de 10 $k\Omega$	\$5,50
Cables de conexión	\$8,00
Protoboard	\$232,00
Arduino Uno	\$720,00
TOTAL	\$1000,00

9. Conclusiones

Tras haber realizado todos los pasos pedidos en el enunciado de este trabajo práctico, resta destacar las conclusiones que la experiencia ha aportado.

Si bien en este caso de aplicación, se utilizan solo los pines de un determinado puerto, tanto para encender el led, como para detectar el estado del pulsador, el desarrollo del programa da una idea bastante completa de que cosas deberían modificarse en este si fuera necesario utilizar varios puertos. Por otra parte, hacer uso de las resistencias pull-up internas es una práctica cuya utilidad se extiende más allá de este trabajo, y puede ser útil a futuro.