



FACULTAD DE INGENIERIA

Universidad de Buenos Aires

Laboratorio de Microcontroladores - 86.07

Manejo de Puertos

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	

Índice

1. Objetivo	2
2. Descripción	2
3. Diagrama en bloques	2
4. Esquemático	2
5. Listado de componentes	2
6. Diagrama de Flujo	3
7. Código de programa	4
7.1. Código sin resistencia de pull-up	4
7.2. Código con resistencia de pull-up	4
8. Resultados	5
9. Conclusiones	5

1. Objetivo

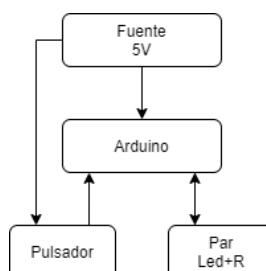
El objetivo de este trabajo es usar los puertos del microcontrolador para entrada y salida. Y entender la utilidad de la resistencia de pull-up interna.

2. Descripción

En este trabajo se programa un microcontrolador para recibir una señal de entrada (proveniente de un pulsador a un pin en particular) para repetirla en otro puerto/pin y de esa forma encender un LED; y se ve que modificaciones se deben hacer al circuito y al programa, al usar la resistencia de pull-up interna de los puertos.

Según la *datasheet* del microcontrolador, para habilitar las resistencias de pull-up de un pin, se debe poner en 1 lógico el bit de registro del PORTX al que corresponda el pin, cuando este está en modo entrada (DDRX a 0 lógico). Esto evita que quede un pin flotante, que se pueda controlar la corriente que entra al pin (controlado por V_{cc} y la resistencia interna) y que se disminuya la potencia del circuito (por lo menos en la parte del pulsador) al **no haber necesidad de conectar la fuente al pulsador**; ya que cuando hay resistencia de pull-up, se *setea* un 1 lógico a ese pin por default.¹ Además este último método hace que no sea necesario el resistor de $10\text{ k}\Omega$ (como se muestra en el esquemático del circuito sin R de pull-up).

3. Diagrama en bloques



4. Esquemático

5. Listado de componentes

- Microcontrolador *ATmega328p* y programador USBasp (Arduino UNO) [AR\$ 950]
- LED [AR\$ 10]
- Resistencia ($220\ \Omega$) [AR\$ 4]
- Pulsador [AR\$15]

¹Notar que será necesario cambiar el circuito para aprovechar la resistencia de pull-up como se muestra en la figura 2.

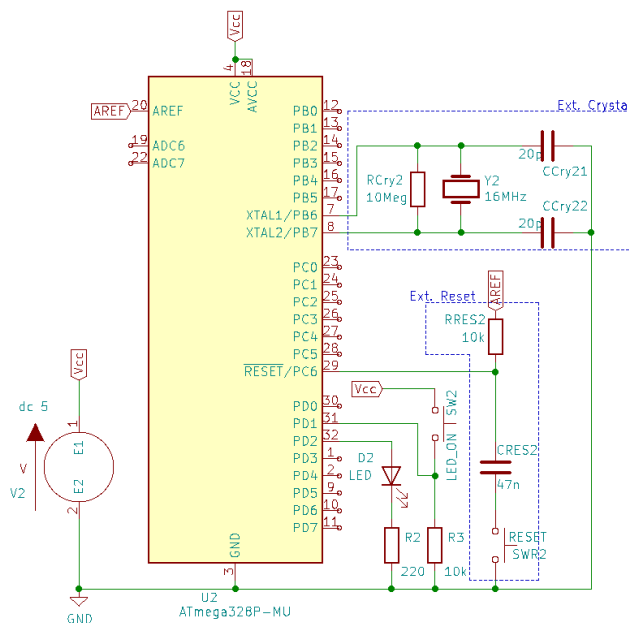


Figura 1: Circuito con R de pull-up deshabilitada.

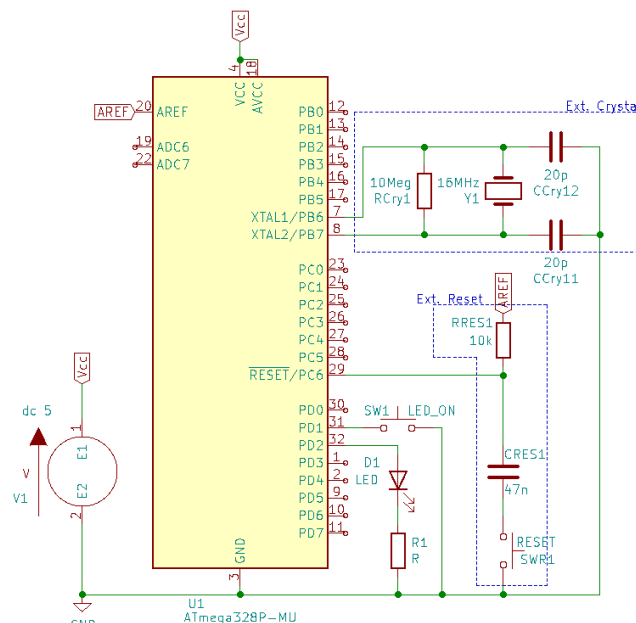
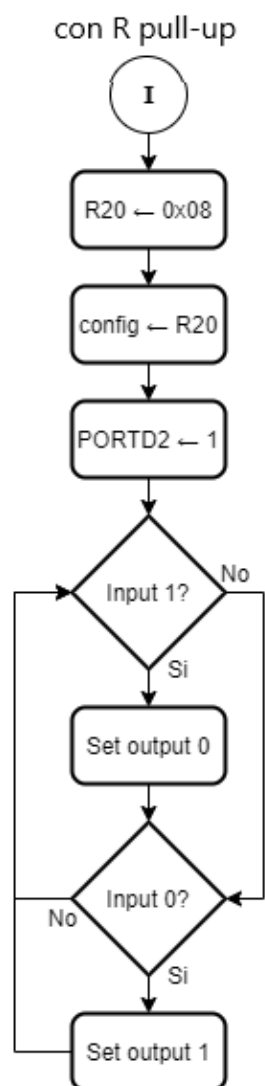
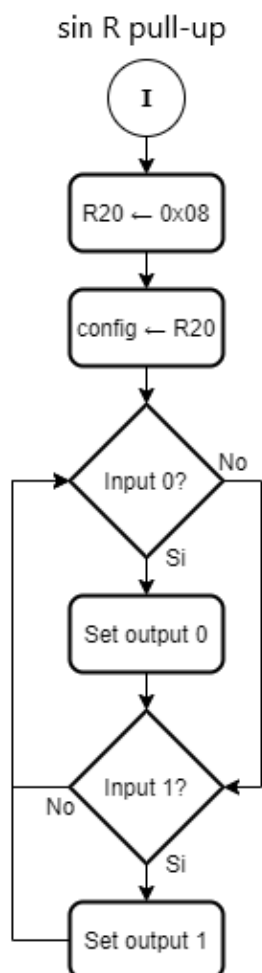


Figura 2: Circuito con R de pull-up habilitada.

6. Diagrama de Flujo



7. Código de programa

7.1. Código sin resistencia de pull-up

```
.include "m328pdef.inc"

; Defino mis puertos de salida/entrada ,
; y su configuracion .

.EQU PORT.IN = PIND
.EQU PORT.OUT = PORTD
.EQU PORT.CONF = DDRD
.EQU PIN.IN = PIND2
.EQU PIN.OUT = PORTD3

.CSEG

JMP MAIN
.ORG INT_VECTORS_SIZE

main:
; Inicializo los puertos
    LDI R20, (0<<0 | 0<<1 | 0<<2 | 1<<3)
    OUT PORT.CONF, R20

start:
; Pregunto LOW
    SBIS PORT.IN, PIN.IN ; Si esta en 0, sigo con la instruccion (apago led)
    CBI PORT.OUT, PIN.OUT ; Si esta en 1, skipeo la instruccion (Pregunto HIGH)
; Pregunto HIGH
    SBIC PORT.IN, PIN.IN ; Si esta en 1, sigo con la instruccion (prendo led)
    SBI PORT.OUT, PIN.OUT ; Si esta en 0, skipeo (Pregunto LOW)
    JMP start
```

7.2. Código con resistencia de pull-up

```
[...]

main:
; Inicializo los puertos
    LDI R20, (0<<0 | 0<<1 | 0<<2 | 1<<3)
    OUT PORT.CONF, R20
    SBI PORTD, PIN.IN ; Habilito R pull-up

start:
; Pregunto HIGH
    SBIC PORT.IN, PIN.IN ; Si esta en 1, sigo con la instruccion (apago led)
    CBI PORT.OUT, PIN.OUT ; Si esta en 0, skipeo la instruccion (Pregunto HIGH)
; Pregunto LOW
    SBIS PORT.IN, PIN.IN ; Si esta en 0, sigo con la instruccion (prendo led)
    SBI PORT.OUT, PIN.OUT ; Si esta en 1, skipeo (Pregunto LOW)
    JMP start
```

8. Resultados

Se logro encender un LED usando un pulsador para recibir una señal de entrada y replicarla en un pin de salida del microcontrolador con los dos métodos.

9. Conclusiones

Mediante el manejo de puertos de entrada y de salida, se lograron dos métodos para encender un LED en un pin del microcontrolador controlado por un pulsador en otro pin de entrada. Para uno de ellos se activo la resistencia de pull-up, así es posible controlar la corriente que entra al pin cuando hay un 1 o un 0 lógico. Para poder aprovechar también la ventaja de consumo que trae, también fue necesario cambiar levemente el circuito, colocando el pulsador directamente en el pin, de esta forma **evitamos** tener un pin flotante cuando esta en 0 lógico, y no es necesario tener el pulsador conectado a V_{cc} .