

# TRABAJO PRÁCTICO N° 3

## Auto Fantástico

16/Jun/2020

---

98934 Vazquez, Rodrigo rvazquez@fi.uba.ar

### Docentes:

Campiglio, Guillermo Carlos

Stola, Gerardo Luis

Cofman, Fernando

Salaya Velazquez, Juan Guido

---

### Resumen

En el presente trabajo se desarrollará el diseño de programa que consiste en realizar una animación sobre 6 LED, dicha animacion debe emular el comportamiento que tenian las luces en el frente del auto de la serie televisiva *El Auto Fantástico*. Esta animacion consiste en un efecto de recorrido individual de las luces encendidas con un efecto de *rebote* en cada extremo. Para ellos se utilizo el microcontrolador *Atmel 328P* en la plataforma *Arduino*. Con posterior armado del circuito de testeo apropiado utilizando una placa *protoboard*

---

## Índice

1. Objetivo	2
2. Descripción del proyecto	2
3. Diagrama de conexión en bloques	2
4. Circuito esquemático	3
5. Listado de componentes y tabla de gastos	3
6. Software	3
7. Resultados	4
8. Conclusiones	5
9. Anexo - Código	6

## 1. Objetivo

Avanzar en el manejo de puertos del microcontrolador *Atmel 328P*, para analizar sus características de corriente continua y analizar la potencia y corriente entregada, comparando estos valores con la documentación del fabricante.

## 2. Descripción del proyecto

Se realizará un programa en lenguaje *Assembly* para la familia de microcontroladores *AVR*. La conexión de los perifericos estará dada en el puerto B del microcontrolador, ocupando desde el *pin 0* al *pin 5* con 6 diodos LED, utilizando al puerto como salida para mostrar una animacion de luces.

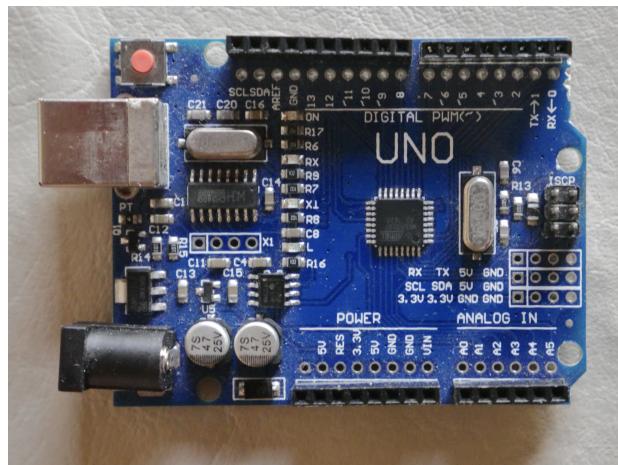


Figura 1: Placa Arduino UNO utilizada

## 3. Diagrama de conexión en bloques

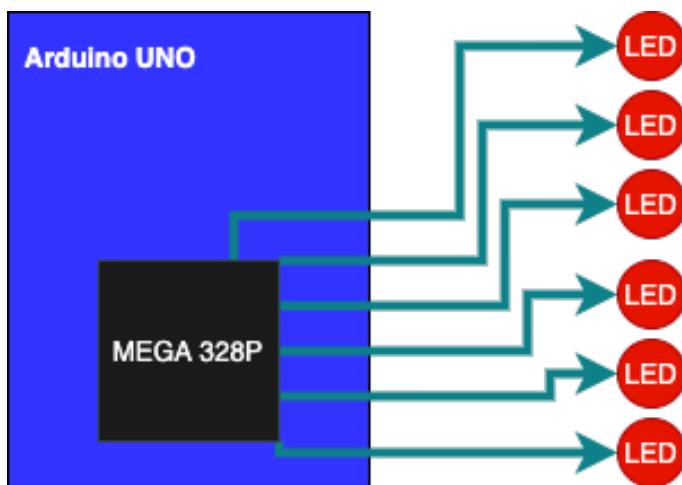


Figura 2: Diagrama de conexión del hardware

## 4. Circuito esquemático

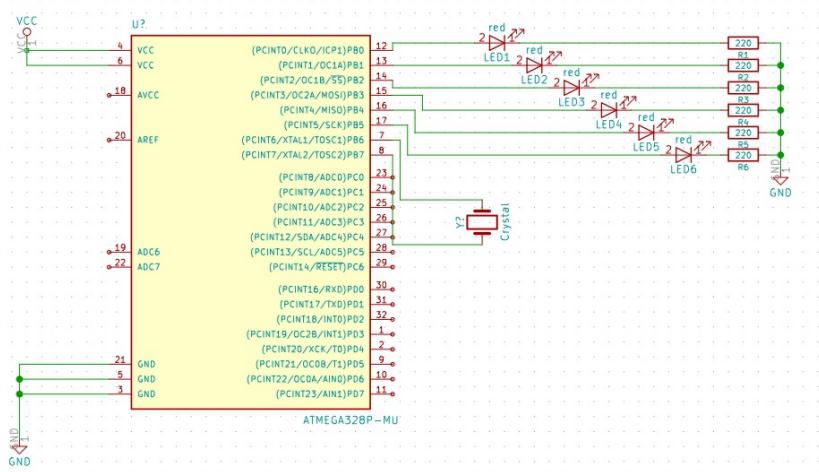


Figura 3: Circuito esquematico

## 5. Listado de componentes y tabla de gastos

Componente	Precio
Arduino UNO	\$750
6 LED	\$36
6 Resistencias	\$6
Total	\$ 792

Tabla 1: Tabla de componentes

## 6. Software

El programa comienza con la configuración del **Puerto B** como salida y encendiendo el primer LED ubicado en PB0. También se definieron dos máscaras que tendrán la función de indicar los pines de principio y fin de la animación. Luego se procede a utilizar la instrucción `lsl` para rotar a izquierda el puerto logrando que se de comienzo a la animación encendiéndolo el LED siguiente y apagando el actualmente encendido. Esta acción se encuentra en un *loop* hasta que el puerto se encuentre en el último estado de la *ida* de la animación. Cuando se llega a este estado, se pasa a hacer otra secuencia pero utilizando la instrucción `lsr` para rotar el puerto hacia derecha y así generando la *vuelta* de la animación. Esto se repite hasta volver al estado inicial, donde el primer LED estaba encendido, y allí se comienza de nuevo.

Se utilizaron las instrucciones de rotación lógica dado que esta no tiene el cuenta el flag de *Carry*, situación que afectaría la animación.

El software implementado también toma codificaciones hechas en ejercicios previos como la subrutina `DELAY`, esta actúa como espera de tiempo 25 ms con el fin de regular la velocidad de la animación.

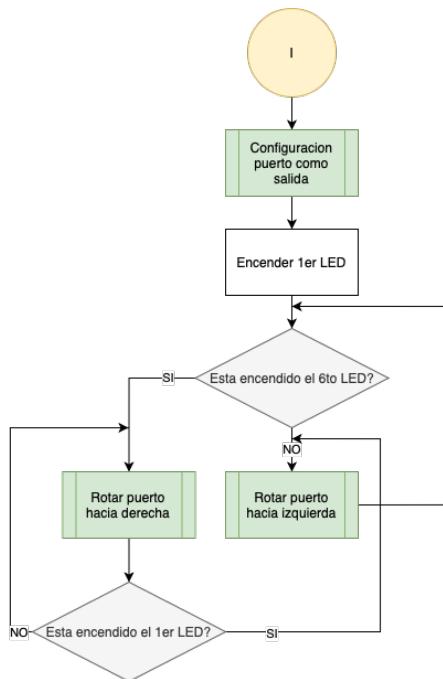


Figura 4: Diagrama de flujo del programa para encender un LED con botón pulsador

## 7. Resultados

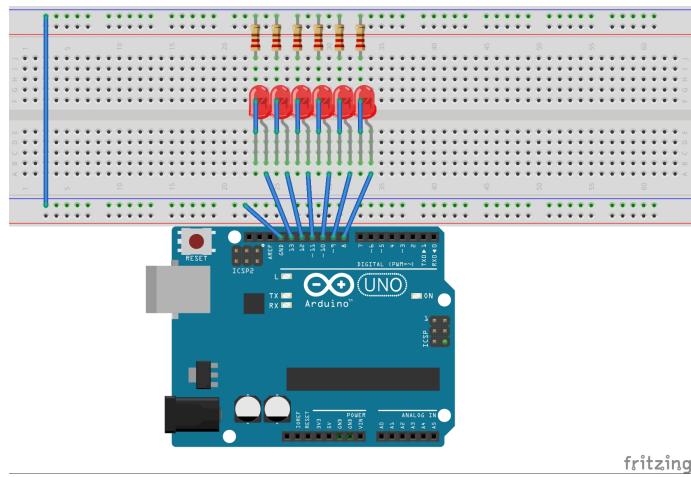


Figura 5: Conexionado utilizado

Se logró realizar la animación y visualizar correctamente, aunque hubo un pequeño paso iterativo para ajustar el *delay* y que la animación resulte clara (y lo más similar a la serie de TV posible). Al utilizar LED rojo, su tensión de directa es 1,7 V y se calcula que la corriente que pasa por el pin de salida es de 15 mA, siendo acorde a lo estipulado en la documentación del microcontrolador donde se indica que la corriente máxima que puede entregarse cuando la salida es un estado *Alto* es de **IOH= 20 mA**

El consumo del puerto entero es equivalente al consumo de un solo pin dado que en todo momento hay solo un pin con estado alto y entregando corriente. La potencia entregada por el **Puerto B** resulta  $P_{PORTB} = 5 \text{ V} \cdot 15 \text{ mA} = 75 \text{ mW}$  lo cual implica un 75% de la potencia maxima que puede entregar cada pin.

## 8. Conclusiones

Las principales conclusiones de este trabajo recaen en el manejo apropiado de los puertos de entrada y salida del microcontrolador Atmel Mega 328P vía el lenguaje de *Assembly* y que su comportamiento sea acorde a la documentación del fabricante. Siendo necesario un correcto conocimiento de los registros de estado, parámetros de las corrientes y tensiones de entrada/ salida.

## Bibliografía

MAZIDI, Muhammad Ali; NAIMI, Sarmad; NAIMI, Sepehr (2011). "The AVR microcontroller and embedded systems. Embedded system using Assembly and C". Pearson Education Inc. New Jersey.

## 9. Anexo - Código

```

1 .include "m328Pdef.inc"
2 ;#####
3 ;Seteo de hardware
4 .equ tope_ida = 0b00100000
5 .equ tope_vuelta = 0b00000001
6 .equ led_mask = 0b00111111
7 .def LED_reg = r20
8 .equ LED_PORT = PORTB
9 ldi r16, led_mask
10 out DDRB, r16
11 ;#####
12
13 inicio:
14   ldi LED_reg, tope_vuelta
15   call output
16 ida:
17   lsl LED_reg
18   call output
19   cpi LED_reg, tope_ida
20   brne ida
21 vuelta:
22   lsr LED_reg
23   call output
24   cpi LED_reg, tope_vuelta
25   brne vuelta
26   jmp ida
27 output:
28   out LED_PORT, LED_reg
29   ldi r16, SREG
30   call delay
31   out SREG, r16
32   ret
33 ;DELAY de 25ms
34 delay:
35   push r16
36   push r17
37   push r18
38   ldi r16, 8
39 L3:
40   ldi r18, 255
41 L2:
42   ldi r17, 255
43 L1:
44   dec r17
45   brne L1
46   dec r18
47   brne L2
48   dec r16
49   brne L3
50   pop r18
51   pop r17
52   pop r16
53   ret
54 ;#####
55

```