

(6609) Laboratorio de Microcomputadoras

Proyecto:  
*( tp3 luces de kitt)*

<b>Profesor:</b>	<b>Ing. Jorge A. Alberto</b>
<b>Cuatrimestre / Año:</b>	1ro/2020
<b>Turno de clases prácticas:</b>	Miércoles
<b>Jefe de Trabajos Prácticos:</b>	Pedro Martos
<b>Docente guía:</b>	

Autores			Seguimiento del proyecto									
Nombre	Apellido	Padrón										
Cristian	Simonelli	87879										

**Observaciones:**

---



---



---



---



---



---



---



---

<b>Fecha de aprobación</b>

<b>Firma J.T.P.</b>

<b>COLOQUIO</b>	
<b>Nota final</b>	
<b>Firma Profesor</b>	

<b>Objetivo:</b>	<b>2</b>
<b>Desarrollo.</b>	<b>2</b>
Características eléctricas.	2
Consumo de corriente.	2
Valores típicos.	2
Consumos de corriente según color.	2
Secuencia de encendido.	3
<b>Listado de componentes:</b>	<b>4</b>
<b>Diagrama en bloques:</b>	<b>5</b>
<b>Circuito esquemático:</b>	<b>5</b>
<b>Diagrama de flujos:</b>	<b>6</b>
<b>Código:</b>	<b>7</b>
<b>Resultado:</b>	<b>8</b>
<b>Conclusiones:</b>	<b>8</b>

## Objetivo:

El objetivo del trabajo práctico es prender un array de 6 leds de uno en vez de forma similar a la que lo hacía el auto fantástico, rotando bits.

## Desarrollo.

Se utilizarán los 6 bits disponibles del puerto B del Atmega328p (pines del 8 al 13 del arduino uno).

Se encenderán de a un led a la vez, para encender es necesario poner el pin correspondiente en estado alto. Desde ese pin se conecta un led y una resistencia. La resistencia es necesaria porque el led es un diodo y por lo tanto la corriente es exponencial, suponemos que estamos dentro del rango en el cual el led se comporta como una caída de tensión fija, por lo tanto controlamos la corriente con una resistencia. Dependiendo del valor de la resistencia, y por lo tanto la corriente, se varía el brillo del led.

## Características eléctricas.

### Consumo de corriente.

El pack de leds que compre tiene leds de distintos colores.

A saber rojo, blanco, amarillo y azul.

Cada color tiene su caída de tensión y por supuesto no tienen hoja de datos.

Valores típicos según wikipedia:

### Valores típicos.

Tipo de diodo	Diferencia de potencial típica (voltios)
Rojo de bajo brillo	1.7 voltios
Rojo de alto brillo, alta eficiencia y baja corriente	1.9 voltios
Naranja y amarillo	2 voltios
Verde	2.1 voltios
Blanco brillante, verde brillante y azul	3.4 voltios
Azul brillante y LED especializados	4.6

Con estos valores las corrientes según color (ya que las resistencias son fijas) serán (aproximadamente).

$$i = (5v - v_d)/220$$

### Consumos de corriente según color.

Color	Corriente
Blanco	7,7mA
Rojo	14mA

Amarillo	13,6mA
Azul	7,7mA

Según datasheet se pueden controlar hasta 40mA por pin y hasta 200mA en total.

Parameters	Min.	Typ.	Max.	Unit
Operating temperature	−55		+125	°C
Storage temperature	−65		+150	°C
Voltage on any pin except RESET with respect to ground	−0.5		$V_{CC} + 0.5$	V
Voltage on RESET with respect to ground	−0.5		+13.0	V
Maximum operating voltage		6.0		V
DC current per I/O pin		40.0		mA
DC current $V_{CC}$ and GND pins		200.0		mA
Injection current at $V_{CC} = 0V$		$\pm 5.0^{(1)}$		mA
Injection current at $V_{CC} = 5V$		$\pm 1.0$		mA

Note: 1. Maximum current per port = ±20mA

## Secuencia de encendido.

Para poder manejar el orden se utilizará un registro (R20) el cual se rotará de derecha a izquierda entre 2 valores borde.

izquierdo : 01000000

derecho: 00000001

El valor de borde derecho si se llega a copiar y el izquierdo no. Eso quiere decir 2 cosas.

- 1- El diagrama de flujos no es simétrico.
- 2- Nunca están todos los leds apagados.

Se seguirá la siguiente secuencia.

```
00100000
00010000
00001000
00000100
00000010
00000001
00000010
00000100
00001000
00010000
00100000
```

Se reutiliza la función de retardo del tp1, para que se aprecie el encendido y el apagado de los leds. (los valores se fueron ajustando arbitrariamente hasta que el efecto fuera vistoso)

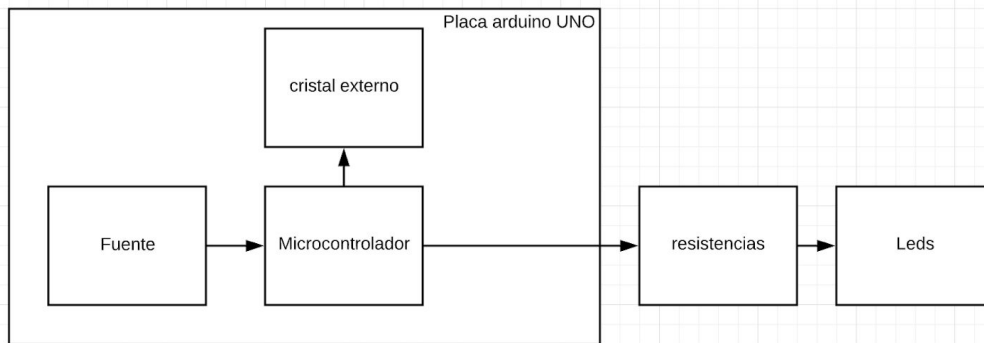
## **Listado de componentes:**

Placa arduino UNO Atmega 328p \$659 aprox 10 usd.

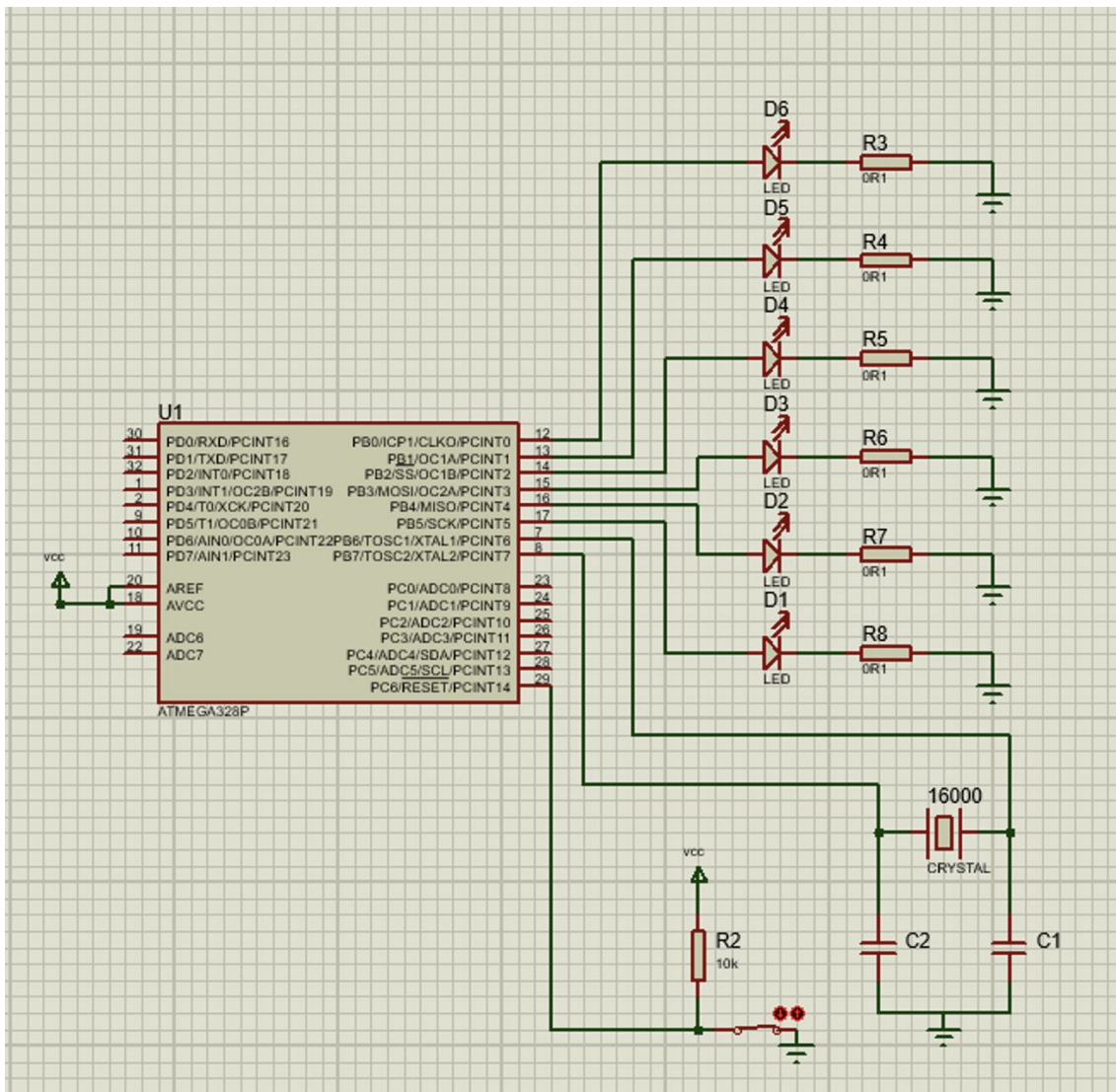
6 led (pack de 10) \$70.

6 resistencias 220 ohm 1/8w 1% \$50.

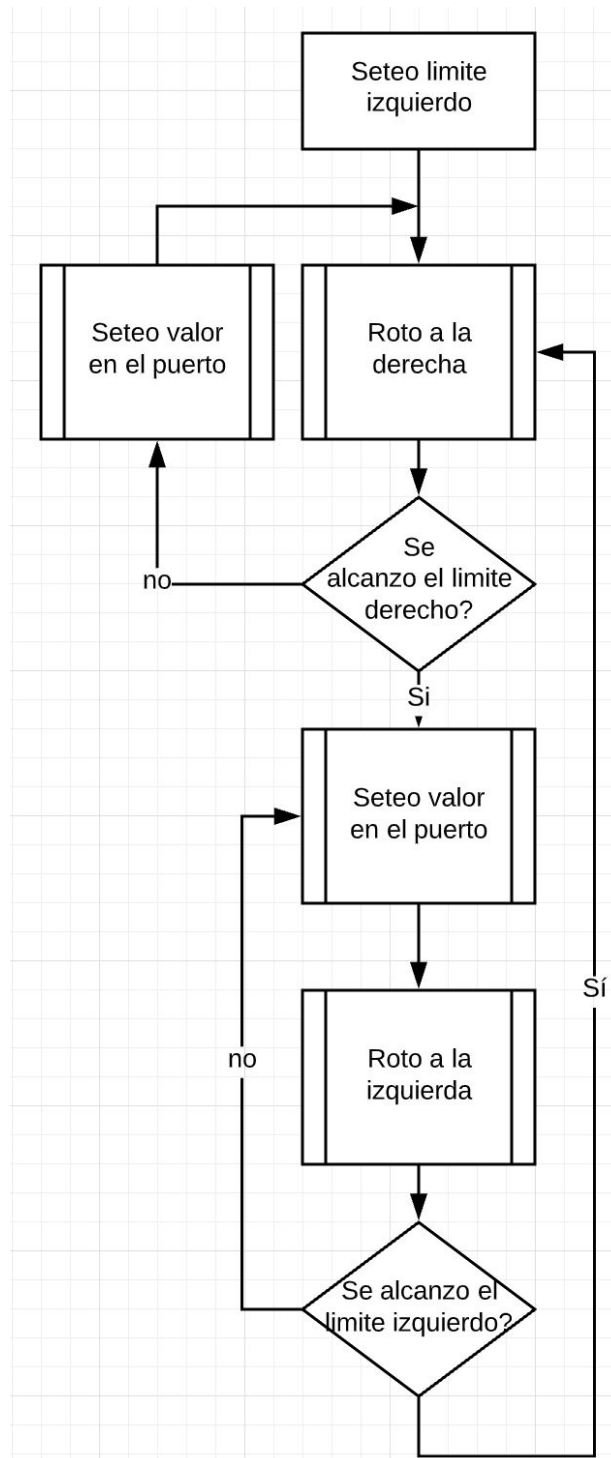
## Diagrama en bloques:



## Circuito esquemático:



## Diagrama de flujos:



## Código:

```
; knight rider lights
.include "m328pdef.inc" ; Valid definitions to 238p

.equ init_mask = 0x40 ; pin connected to the input switch
.equ end_mask = 0x01 ; pin connected to the input switch

.org 0x000 ; The next instruction has to be
written to add 0x0000

        rjmp     main ; Relative jump to main
.org INT_VECTORS_SIZE ; inter vector

main:
        ldi      r20, HIGH(RAMEND) ; Load r20 with the last ram address
higher byte
        out      sph, r20 ; Load higher byte in sp with r20
        ldi      r20, LOW(RAMEND) ; Load r20 with the last ram address
lower byte
        out      spl, r20 ; Load lower byte in sp with r20

        ldi      r20, init_mask ; Set b port with the bound value
right:   lsr      r20 ; right shift ... 010000 -> 001000
        out      PORTB, r20 ; Set the b port pin
        out      DDRB, r20 ; Set the same port pin as output and
input all others

        cpi      r20, end_mask ; Compare with the bound value
        breq     left ; If bound value has been reached
start to shift to the other side
        call     delay ; Delay
        jmp      right ; Go to right
left:    lsl      r20 ; left shift... 000001 -> 000010
        cpi      r20, init_mask ; Compare to the left bound value
        breq     right ; If bound value has been reached
start to shift to the other side
        call     delay ; Delay
        out      PORTB, r20 ; Set the b port pin
        out      DDRB, r20 ; Set the same port pin as output and
input all others

        jmp      left ; Goto left

delay: ; Delay procedure
        push     r20 ; Save the r20 value in the stack
        push     r21 ; Save the r21 value in the stack
        push     r22 ; Save the r22 value in the stack
        ldi      r22, 50 ;
loop1:   ldi      r21, 50 ;
```



```

loop2:      ldi      r20, 100      ;
loop3:      dec      r20          ; decrement r20 by 1
           brne     loop3        ; If r20 had reached 0, z flag would
have been seted                ; and we will jump to loop 3
           dec      r21          ; The same as above
           brne     loop2
           dec      r22
           brne     loop1
           pop      r22          ; Set r22, r21, r20 to the same value
that                            ; it had before entere this proc
           pop      r21
           pop      r20
           ret                ; go back to main

```

## Resultado:

Se logró un efecto similar al del auto fantástico.

## Conclusiones:

El microcontrolador avr 328p no tiene todos los bits de los puertos habilitados en cualquier condición. Solo el puerto D tiene los 8 bits.

El puerto utilizado (B) tiene 6 bits habilitados, del 0x00 and 0x20.

Segun la configuracion por default en arduino uno, esos 2 bits se utilizan para el clock externo.

Hay que modificar los fusibles para poder utilizarlos.