



Laboratorio de Microprocesadores - 86.07

Trabajo Práctico Obligatorio N°1

Profesor:	Ing. Guillermo Campiglio
Cuatrimestre/Año:	1º/2020
Turno de las clases prácticas	Miércoles
Jefe de trabajos prácticos:	Ing. Pedro Ignacio Martos
Docente guía:	Ing. Fabricio Baglivo, Ing. Fernando Pucci
Autores	
Mariano Federico	Guglieri 99573
Seguimiento del proyecto	

Observaciones:

.....

.....

.....

.....

.....

.....

.....

.....

.....

Fecha de aprobación		Firma J.T.P

Coloquio	
Nota final	
Firma profesor	

Índice

1. Objetivos	1
2. Desarrollo	1
2.1. Cálculo de la corriente entregada	1
3. Diagrama en bloques	1
4. Esquemático y listado de componentes	2
5. Diagrama de flujos	3
6. Códigos	3
7. Resultados	4
8. Conclusiones	4

1. Objetivos

El objetivo de este trabajo consta en generar una secuencia de iluminado idéntica a la del "auto fantástico" utilizando 6 leds y calcular la corriente que entrega el micro al realizar la tarea.

2. Desarrollo

La parte física del proyecto consta de 6 LEDS, 6 resistencias de 220Ω , una plataforma arduino, 7 cables para protoboard y una protoboard en donde se realizaron las conexiones. Es conveniente calcular la corriente y la potencia que entrega el micro, ya que en este caso se esta conectando una cantidad considerable de componentes.

2.1. Cálculo de la corriente entregada

Para el cálculo de la corriente, la tensión dada por el micro se estableció en 5V y la caída de tensión de los leds, al ser verdes, es de 2.4V. La resistencia utilizada para cada uno fue de 220Ω .

$$I = \frac{V - V_{LED}}{R} = \frac{5V - 2.4V}{220} = 11,82mA \quad (2.1)$$

Esta es la corriente aportada por el micro a cada uno de los leds. Como los leds se encienden de uno a la vez la corriente siempre serpa 11,82mA. La potencia del micro está dada por

$$P = vI = 5V \times 11,82mA = 59,1mW \quad (2.2)$$

Para la parte del desarrollo del programa, como novedad a los trabajos ya realizados, se hizo uso de subrutinas y de memoria RAM. Fuera de esto, el código presenta los usos que ya se venían viendo para declarar puertos y utilizar condicionales para generar loops.

3. Diagrama en bloques

El diagrama en bloques es similar al del trabajo práctico uno, sólo que en vez de estar compuesto por un led, el último bloque esta constituido por una linea de 6 leds.

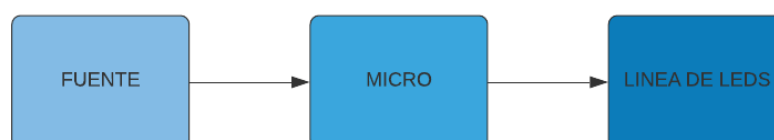


Figura 1: Diagrama en bloques del proyecto

4. Esquemático y listado de componentes

A continuación se presenta el esquemático.

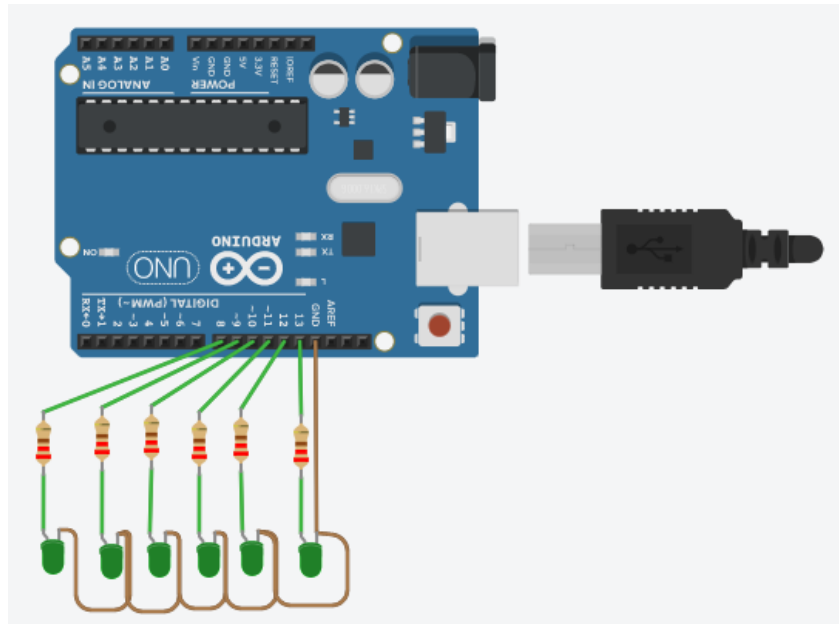


Figura 2: Esquemático del proyecto

La plataforma de arduino utiliza un cristal de 16MHz y una resistencia de reset de $10k\Omega$ de la siguiente manera:

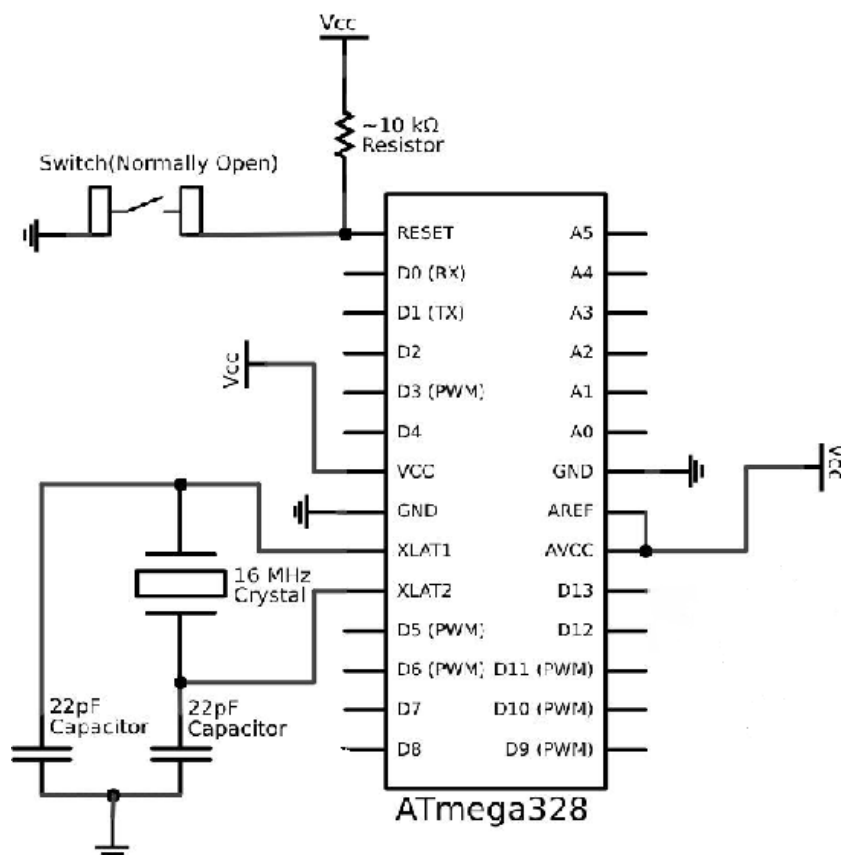


Figura 3: Microcontrolador con cristal y resistencia de reset

5. Diagrama de flujos

El diagrama de flujos presente indica la lógica del programa. La implementación de los pasos a seguir están detallados en el código.

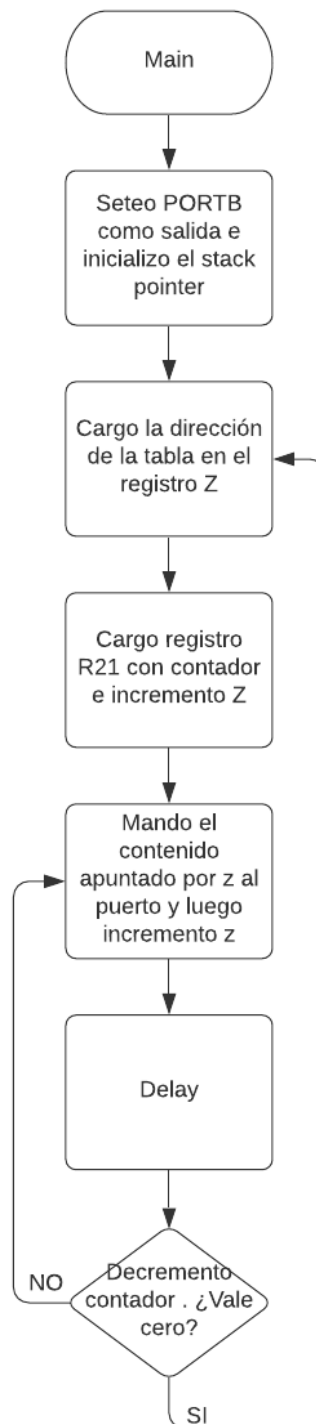


Figura 4: Diagrama de flujo del programa

6. Códigos

A continuación se presenta el código. El archivo assembler subido contiene los detalles de cada sección.

```
.include "m328pdef.inc"
```

```
.cseg
```

```
.org      0x0000
                jmp                main

.org INT_VECTORS_SIZE

main:
    ldi                R20, 0xff
    out                DDRB, R20

    ldi                R16, HIGH(RAMEND)
    ldi                R16, LOW(RAMEND)
    out                SPL, R16

loop:
    ldi                ZH, HIGH(secuencial <<1)
    ldi                ZL, LOW ((secuencial <<1)+1)
    lpm                R21, Z+

siguiente:      lpm                R20, Z+
                out                PORTB, R20
                call                delay
                dec                R21
                brne             siguiente

                jmp                loop

.org      0x0500
secuencial: .DB 0,10,1,2,4,8,16,32,16,8,4,2
.org      0x0550
secuencia2: .DB 0,2,21,42
.org      RAMEND
delay:
    ldi r23, 32
    loop3: ldi r24, 255
    loop2: ldi r22, 255
    loop1: dec r22
    brne loop1
    dec r24
    brne loop2
    dec r23
    brne loop3
    ret
```

7. Resultados

Los resultados fueron los esperados. Se pudo generar la secuencia de leds sin inconvenientes.

8. Conclusiones

Como conclusión se puede observar que el uso de subrutinas facilita la lectura del código, haciendolo más legible y permitiendo visulizar la lógica central del programa.

Si bien este trabajo se hubiera podido realizar sin recurrir a la memoria RAM, a partir de la implementación de esta se pudo observar los beneficios que conlleva utilizarla en el código. Utilizando memoria RAM, se puede dejar a los registros de propósito general exclusivamente para la lógica del programa. Es decir que si quisieramos implementar otra secuencia, la lógica no se modificaría. Sólo se tendría que cargar otro vector con la secuencia deseada.