

UNIVERSIDAD DE COSTA RICA

IE-0624

LABORATORIO DE MICROCONTROLADORES

REPORTE #1

PROFESOR: FEDERICO RUÍZ UGALDE

---

# Reporte 1

---

*Estudiante:*

Andrés Alvarado Velázquez

*Carné:*

B30316

18-09-18

## Índice

<b>1. Bitacora de Trabajo</b>	<b>IV</b>
1.1. Parte I. Configuración básica del sistema y encendido . . . . .	IV
1.2. Parte II. Arduino Software. Uso Basico. Blink . . . . .	IV
1.3. Modificación Blink . . . . .	IV
<b>2. Resultados de la Actividad y Análisis de los resultados</b>	<b>VI</b>
<b>3. Cuestionario de Guía</b>	<b>IX</b>
<b>4. Lista de componentes</b>	<b>XI</b>
<b>5. Lista de Equipo</b>	<b>XII</b>
<b>6. Código</b>	<b>XIII</b>
<b>7. Referencias</b>	<b>XIV</b>

## Índice de figuras

1. LED integrado del arduino apagado . . . . .	VI
2. LED integrado del arduino encendido . . . . .	VI
3. LED conectado al pin 12 del arduino apagado . . . . .	VI
4. LED conectado al pin 12 del arduino encendido . . . . .	VII
5. Esquemático del circuito externo conectado al pin D12 . . . . .	VII
6. Señal de parpadeo del LED externo, transiciones de 1s . . . . .	VIII
7. Señal de parpadeo del LED externo, frecuencia minima visible por el ojo . . . . .	VIII
8. Señal de parpadeo del LED externo, frecuencia minima alcanzada . . . . .	VIII

**Resumen**

Este laboratorio tiene como fin introducir al uso del arduino, mediante un código sencillo que muestre uno de los uso que se le puede dar a este, así como el uso del osciloscopio como herramienta para ver el comportamiento de algo en específico, en este caso la señal dada por el encendido y apagado de un LED.

**Palabras Claves** Microcontrolador, Arduino, USB, LED, Frecuencia de muestreo.

# 1. Bitacora de Trabajo

## 1.1. Parte I. Configuración básica del sistema y encendido

Se inicia el laboratorio realizando la actualización de la distribución de Linux Debian a la última versión de Debian Unstable.

Después de eso se realizó la creación del directorio en el que se guardaría el archivo descargado de Arduino.

Se continua con la instalación del software de arduino, siguiendo los pasos indicados en la guía. Después de la descarga, fue necesario realizar un cambio en la instrucción de descompresión del archivo descargado, eliminando '-z' para que la instrucción funcione, obteniendo:

```
tar -xf Arduino-1.8.4-linux64.tar.xz
```

Después, de ejecutar arduino, desde el directorio en el que se ubica, se conectó el arduino a un puerto USB de la computadora y se ejecutó la siguiente instrucción para verificar que el sistema operativo detectó correctamente el Arduino:

```
sudo dmesg
```

Para la ejecución de Arduino en una forma más sencilla, se hizo un cambio en "PATH", escribiendo este cambio en .bashrc.

Finalmente se debe agregar el usuario que se esta utilizando a el grupo 'dialout', grupo al que también es asignado el arduino para poder enviarle el código que queremos utiliza mediante la siguiente instrucción sustituyendo "user" por el nombre del usuario que estamos utilizando.

```
sudo adduser [user] dialout
```

## 1.2. Parte II. Arduino Software. Uso Basico. Blink

Para esta sección, se realizó la ejecución de Arduino, y luego del programa ejemplo "Blink". Con el Arduino todavía conectado, se hace la compilación para verificar errores y se carga el programa al Arduino, por lo que la luz roja "L" del arduino destelló.

## 1.3. Modificación Blink

En esta sección se realizaron modificaciones en el programa, cambiando el tiempo de la instrucción "delay", de 1000 ms a 5000 ms. Después de realizar este cambio, se compiló y descargó el programa al Arduino de nuevo.

Como siguiente paso, se buscó el tiempo del "delay" hasta que no se pudiera observar la intermitencia del LED con los ojos, este tiempo fue de 12 ms.

Después, se modificó el programa para que el pin de salida del Arduino fuera el 12, declarándolo como de salida. OUTPUT

```
pinmode (d12 , OUTPUT);
```

Después se desconectó el arduino. Una instrucción dada fue de nunca conectar el Arduino a la computadora cuando se están utilizando los pines externos antes de tener el visto bueno del profesor o asistente.

Se conectó el arduino a una protoboard y un LED en serie a una resistencia de 330 ohms, entre el pin

+5V y el pin D12. Se modificaron las instrucciones de "digitalWrite" del programa y luego se compiló y cargó el programa de nuevo al Arduino, para ver el LED parpadear a la velocidad previamente seleccionada.

Por medio del osciloscopio, se observó el comportamiento del circuito, como una onda cuadrada como se muestra en la sección de resultados y se realizaron mediciones de velocidad, tiempo en bajo y tiempo en alto, periodo y frecuencia.

## 2. Resultados de la Actividad y Análisis de los resultados

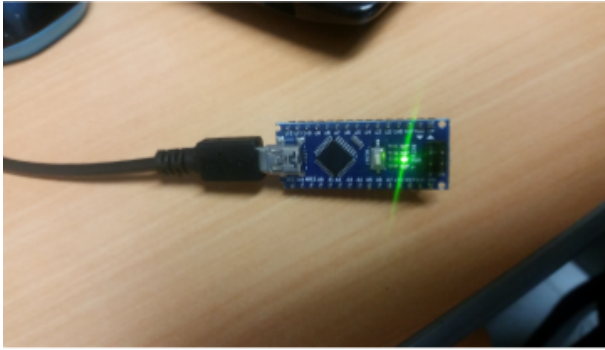


Figura 1: LED integrado del arduino apagado

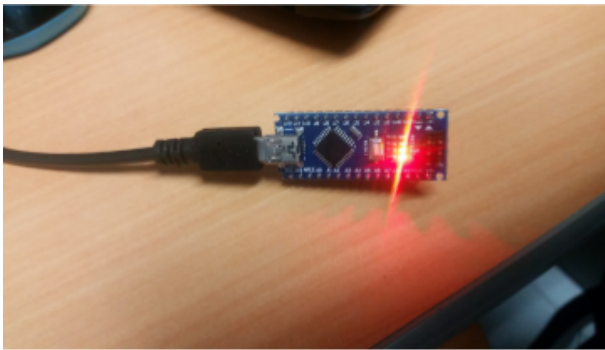


Figura 2: LED integrado del arduino encendido

En la figuras 1 y 2 se puede observar como el LED integrado del arduino se apaga y se enciendo, esto se logro mediante el código 6.

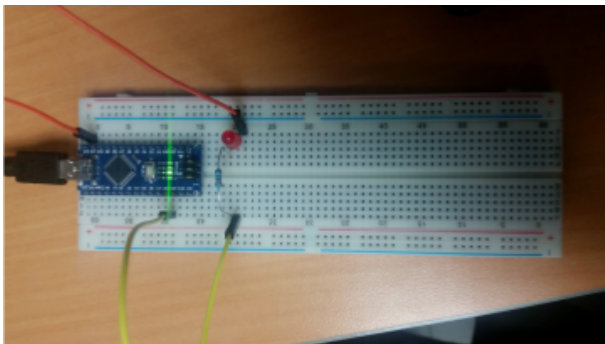


Figura 3: LED conectado al pin 12 del arduino apagado

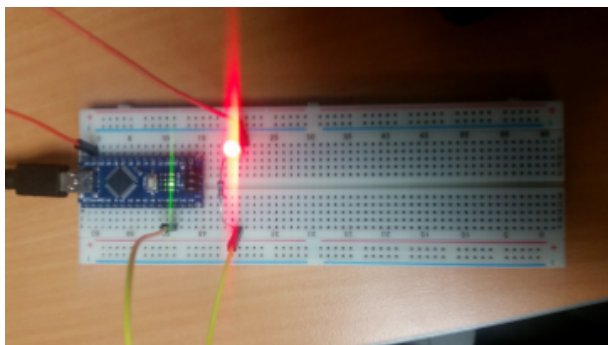


Figura 4: LED conectado al pin 12 del arduino encendido

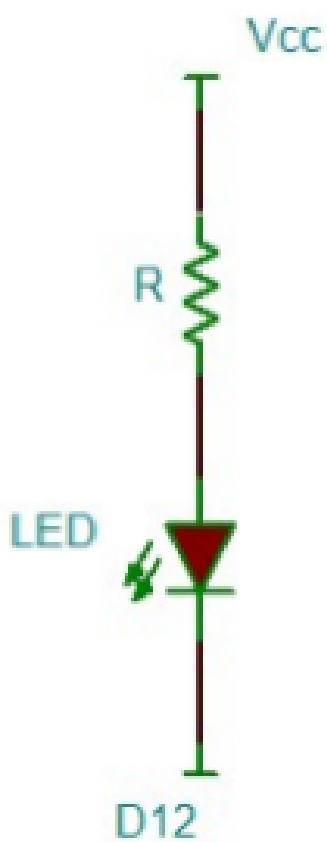


Figura 5: Esquemático del circuito externo conectado al pin D12

Se conecta al arduino un LED de la forma mostrada en la figura 5 el pin D12, mediante el código 6 que se encarga de encender el LED cuando dicho pin se encuentre en estado bajo y apagarlo cuando esta en estado alto. Estos cambios de puede observar en las figuras 3 y 4.



Figura 6: Señal de parpadeo del LED externo, transiciones de 1s

La figura anterior muestra los cambios de estado del D12 que a su vez corresponde a la señal de parpadeo del LED. Como se puede observar corresponde a una señal cuadrada con periodo de 2s, dado que como se aprecia en 6 se poseen 2 instrucciones de 'delay(1000)'.

A continuacion se puede ver la señal del LED, esta corresponde a la mínima visible por el ojo humano, esto se logro modificando el parámetro de la instrucción 'delay()' buscando el valor en la cual ya no se pudiera distinguir cuando el LED se apagaba y encendía.

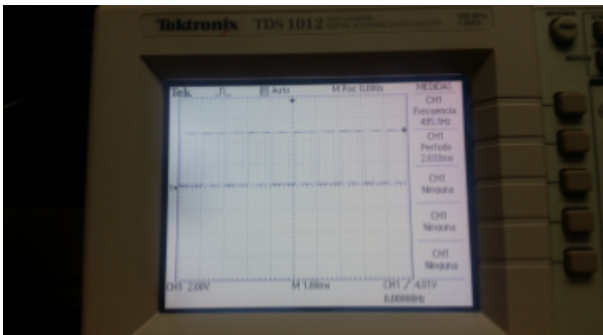


Figura 7: Señal de parpadeo del LED externo, frecuencia minima visible por el ojo



Figura 8: Señal de parpadeo del LED externo, frecuencia minima alcanzada

En la figura anterior se observa la señal del LED minima alcanzada, se buscaba encontrar la frecuencia a la cual el osciloscopio ya no pudiera detectar la señal del LED. Dado a las limitaciones de la instrucción 'delay' no se pudo llegar a dicha frecuencia.



### 3. Cuestionario de Guía

1. Anote el puerto "virtual" serial asignado por el Kernel a la conexión USB del arduino

El puerto virtual asignado fue ttyUSB0

**Presione el botón Upload. Este botón descarga "flashea" el programa compilado al microcontrolador.**

2. Describa lo que sucede durante este proceso.

Cuando se presiona el botón Upload, el programa se carga al Arduino y ambos LEDs del Arduino parpadean.

**La instrucción "delay", realiza una espera sobre el programa, esto causa que el LED se mantenga en el último estado seleccionado por el tiempo determinado por "delay". Modifique "delay" en todos los lugares del programa, cámbielo de 1000 a 5000. Compile y descargue el programa en el arduino. (de ahora en adelante "ejecute" el programa)**

3. ¿Cuánto tiempo dura el LED encendido y apagado?

Por los tiempos de "delay" asignados, el LED dura 5000 ms encendido y 5000 ms apagado.

4. ¿Cuál es la unidad utilizada en la instrucción "delay"?

El "delay" se mide en milisegundos.

**Siga modificando los valores de "Delay" hasta que no logre observar con los ojos la intermitencia**

5. ¿Cuál es el valor más bajo de "delay" que puede usted observar con sus ojos?

Por medio de la realización de la prueba, se detectó que el valor más bajo de "delay" que los ojos pueden detectar es de 12 ms.

6. Convierta dicho valor al periodo de la señal y su frecuencia.

Haciendo la conversión, se tiene que el periodo de la señal es de 24 ms, con la señal obtenida siendo cuadrada, encendido = 1 y apagado = 0. A partir del periodo, se sabe también que la frecuencia de la señal es de 84 Hz.

**Conecte un osciloscopio al pin D12. Para esto espere las instrucciones del profesor.**

7. Encuentre la velocidad mínima de parpadeo del LED visible por el osciloscopio.

La función "delay" no acepta cantidades de punto flotante y hace redondeo. Por esto el menor "delay" que se puede poner es de cero. Técnicamente, no se puede encontrar la velocidad mínima de parpadeo del LED, visible por el osciloscopio, ya que el osciloscopio es más rápido que el microcontrolador, por lo que la velocidad mínima se consigue quitando la instrucción "delay" de la programación.

8. Mida con el osciloscopio el tiempo en bajo y el tiempo en alto. Observer si dichos tiempos varían o no son estables.

De la imagen, se puede observar que el tiempo en alto es igual al tiempo en bajo, por lo que cada uno equivale a la mitad del periodo de la señal.

9. Calcule el periodo y la frecuencia de la señal.

De la figura que muestra este procedimiento, se muestra que por medio del osciloscopio se sabe que el periodo es de  $7,489 \mu s$ , mientras que la frecuencia de la señal es de  $133,5 \text{ kHz}$ .

10. ¿Qué tipo de señal es esta?

La señal obtenida, como se muestra en la última figura de la sección de resultados, es cuadrada.

11. Por qué se conecta el LED a la señal de  $+5V$  en lugar de GND o tierra?

Aunque se puede hacer de las dos formas, la cantidad de corriente que puede entregar el pin es menor a la corriente que puede drenar.

## 4. Lista de componentes

En la tabla 1 se muestra la lista de componentes que se van a utilizar en el experimento.

Componente	Sigla	Valor nominal	Valor medido	Tolerancia	Potencia
Diodo LED	-	-	-	-	-
Resistencia	R5	330 $\Omega$	328 $\Omega$	5 %	0.25W

## 5. Lista de Equipo

En la tabla 2 se muestra la lista de equipo utilizado en el experimento.

Equipo	Marca	Modelo	Placa
Osciloscopio digital	Tektronix	TDS1012	235546

## 6. Código

A continuación se presenta el código que se utilizó en el arduino nano.

```
void setup(){  
    //initialize digital pin LED_BUILTIN as an output  
    pinMode(LED_BUILTIN, OUTPUT);  
}  
void loop() {  
    digitalWrite(LED_BUILTIN, HIGH); //Turns on the led  
    delay(1000) //wait for a second  
    digitalWrite(LED_BUILTIN, LOW); //turns off the led  
    delay(1000); //wait for a second  
}
```

El código anterior es un ejemplo que se encuentra en los “Ejemplos” del programa de arduino que se encarga de encender o apagar el LED integrado que posee este.

```
int d12 = 12;  
void setup() {  
    pinMode(d12, OUTPUT); //inicializa el pin d12 como un output  
}  
void loop() {  
    digitalWrite(d12, HIGH); //encender el led  
    //delay(0);  
    digitalWrite(d12, LOW); //apagar el led  
    //delay(1);  
}
```

Este código se encarga de cambiar de estado(Alto o Bajo) un pin, en este caso el D12, del arduino para encender o apagar un LED que se encuentra conectado externamente a este.

## 7. Referencias

- [1] Página oficial de Arduino. <https://www.arduino.cc/en/Tutorial/DigitalPins>