

	UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA FACULTAD DE INGENIERÍA CAMPUS MEXICALI Formato de prácticas de laboratorio
---	---

ALUMNO - MATRÍCULA - GRUPO	LÓPEZ GARCÍA Kevin - 1155800 - 395		
CARRERA	PLAN DE ESTUDIOS	CLAVE DE ASIGNATURA	NOMBRE DE LA ASIGNATURA
ING. AEROESPACIAL	2009-2		INSTRUMENTACIÓN

LABORATORIO DE	NOMBRE DE LA PRÁCTICA	NO. PRÁCTICA	DURACIÓN
INSTRUMENTACIÓN	INTENSIDAD DE LUMINOSIDAD DE UN LED	4	2 HORAS

1. INTRODUCCIÓN

En este reporte de práctica se explicará el desarrollo de un circuito cuyo objetivo es regular la intensidad de un LED, por medio de la manipulación de las entradas analógicas del microcontrolador Arduino UNO.

2. OBJETIVO

Elaborar un circuito capaz de regular la intensidad de un LED por medio de un potenciómetro.

	UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA FACULTAD DE INGENIERIA MEXICALI INGENIERO AEROESPACIAL Formato de prácticas de laboratorio
---	---

3. MARCO TEÓRICO


Entrada analógica: Una señal analógica es una magnitud que puede tomar cualquier valor dentro de un intervalo $-V_{cc}$ a $+V_{cc}$. Una señal analógica de tensión entre 0V y 5V podría valer 2.72, o cualquier otro valor entre estos rangos; a diferencia de las entradas digitales, que sólo pueden leer valores de 0V o 5V (Llamas, s.f.).

analogRead(): Registra el valor especificado en el pin análogo, estos pines van del A0 al A5 en la mayoría de las placas de arduino. Los valores de voltaje de entrada son leídos entre 0 y 5V, estos valores se convierten en enteros entre 0 y 1023 (Arduino, 2020).

analogWrite(): Escribe un valor analógico (PWM wave) a un pin. Se puede utilizar para iluminar un LED y variar su brillo o manejar un motor a varias velocidades (Arduino, 2020).

Pulse Width Modulation: Es una técnica para obtener resultados analógicos por medios digitales. El control digital es utilizado para crear una onda cuadrada, una señal entre encendido y apagado (Arduino, 2020).

Potenciómetro: Es un dispositivo conformado por 2 resistencias en serie, las cuales poseen valores que pueden ser modificados por el usuario (González, 2016). Los potenciómetros utilizan tres terminales y suelen utilizar circuitos de poca corriente. El valor del potenciómetro se expresa en ohmios y el valor del potenciómetro es expresado por el valor máximo de la resistencia que puede tener (Mecafenix, 2017).

FORMULÓ	REVISÓ	APROBÓ	AUTORIZÓ
M.C. JOSÉ MANUEL RAMÍREZ ZARATE	DR. JUAN DE DIOS OCAMPO DÍAZ	M.C. JORGE MIRAMON	DR. DAVID ROSAS ALMEIDA
 UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA FACULTAD DE INGENIERIA MEXICALI INGENIERO AEROESPACIAL			
Formato de prácticas de laboratorio			

4. PROCEDIMIENTO (DESCRIPCIÓN)	
EQUIPO NECESARIO	MATERIAL DE APOYO
Computadora Software Proteus 8.1 Software Arduino 1 led 1 potenciómetro de 10kΩ Placa Arduino UNO	Internet

5. DESARROLLO DE LA PRÁCTICA

Comenzamos por el código de arduino, en donde se deben de leer entradas analógicas y exportar un valor analógico. Para esto se utilizan funciones como:

```
variable = analogRead("pin analógico");  
analogWrite("pin digital", "variable");
```

Con estas variables nos es posible controlar el valor de entrada del potenciómetro, de tal manera que esos valores leídos, no sólo resulten en un encendido y apagado o en una salida digital, sino que podremos aumentar y disminuir la intensidad de la luz, gracias a la variación de la resistencia del potenciómetro y el uso de estas líneas de código.

Como en la práctica anterior, debemos de empezar por declarar las variables en donde los datos recibidos por el circuito serán almacenados.

```
int led = 9; //Se declara el led a utilizar  
float pot = 0; //variable del potenciómetro  
float potti = 0; //variable del potenciómetro en escala de 0 a 255
```

En esta ocasión se utilizó el LED 9 debido a que es un pin que nos permite visualizar la gráfica del osciloscopio...

Posteriormente se programa para el hardware:

```
void setup() {  
  pinMode(led, OUTPUT); //pin de salida  
  pinMode (A0, INPUT); //pin de entrada  
  Serial.begin(9600);  
}
```

Aquí utilizamos la línea Serial.begin(9600); para poder observar el cambio de los valores que recibiremos del potenciómetro, la parte complementaria de este código se escribe en el bloque del software:

```
void loop() {
```

```
pot = analogRead(A0); //lectura del pin análogo
potti = 255 * (pot / 1023); //conversión a escala de 0 a 255
Serial.println(potti); //impresión de valores en el monitor serie
analogWrite(led, potti); //encendido de led
}
```

La lógica aquí es que leemos los datos del pin análogo y se depositan en la variable `pot`. Después, se hace la conversión de escala de 0 – 1023 a 0 – 255, por medio de una regla de tres, y este valor se almacena en la variable `potti`. A continuación, está la línea para imprimir los valores leídos ya convertidos a la escala deseada, y se utiliza `ln` al final para hacer un salto de línea después de cada valor. Por último, se envía el valor ya convertido al LED con `analogWrite`, lo que regula su intensidad.

6. FÓRMULAS O BIBLIOGRAFÍA

Bibliografía

Arduino. (2020). Obtenido de <https://www.arduino.cc/reference/en/language/functions/time/delay/>
González, A. G. (16 de enero de 2016). *PANAMAHITEK*. Obtenido de <http://panamahitek.com/que-es-y-como-funciona-un-potenciometro/>
Llamas, L. (s.f.). *Luis Llamas*. Obtenido de <https://www.luisllamas.es/entradas-analogicas-en-arduino/>
Mecafenix, F. (21 de abril de 2017). *Ingeniería Mecafenix*. Obtenido de <https://www.ingmecafenix.com/electronica/potenciometro/>

7. RESULTADOS Y CONCLUSIONES

La utilización de las entradas analógicas y digitales nos permite controlar mejor los valores que se reciben en el LED, gracias a la línea de código `analogWrite`, es posible cambiar totalmente el valor de salida, y a comparación de la práctica pasada, nuestro LED ahora recibe un valor que puede llegar a ser fraccionario y no sólo 1 o 0, lo que provoca que su intensidad sea regulada. Además, al utilizar esta función, se reducen las líneas del código, pues no tenemos que estar encendiendo y apagando el LED con la función `delay`, y termina siendo un código más corto y que tiene más control sobre los componentes.

8. ANEXOS

Código utilizado:

```
int led = 9; //Se declara el led a utilizar
float pot = 0; //variable del potenciómetro
float potti = 0; //variable del potenciómetro en escala de 0 a 255
void setup() {
    pinMode(led, OUTPUT); //pin digital de salida
    pinMode(A0, INPUT); //pin analógico de entrada
    Serial.begin(9600);
}

void loop() {
    pot = analogRead(A0); //lectura del pin análogo
    potti = 255 * (pot / 1023); //conversión a escala de 0 a 255
    Serial.println(potti); //impresión de valores en el monitor serie
    analogWrite(led, potti); //voltaje enviado al LED
}
```

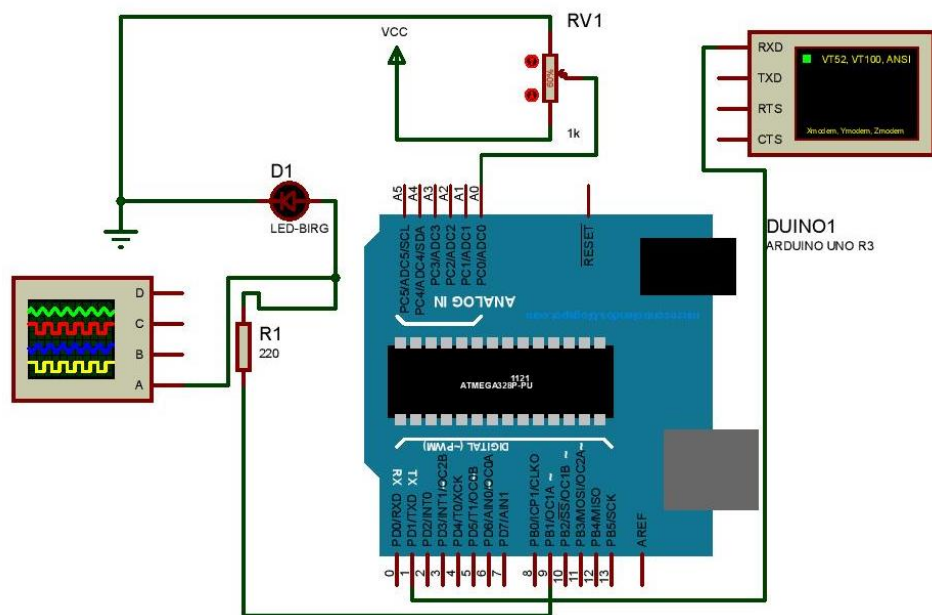
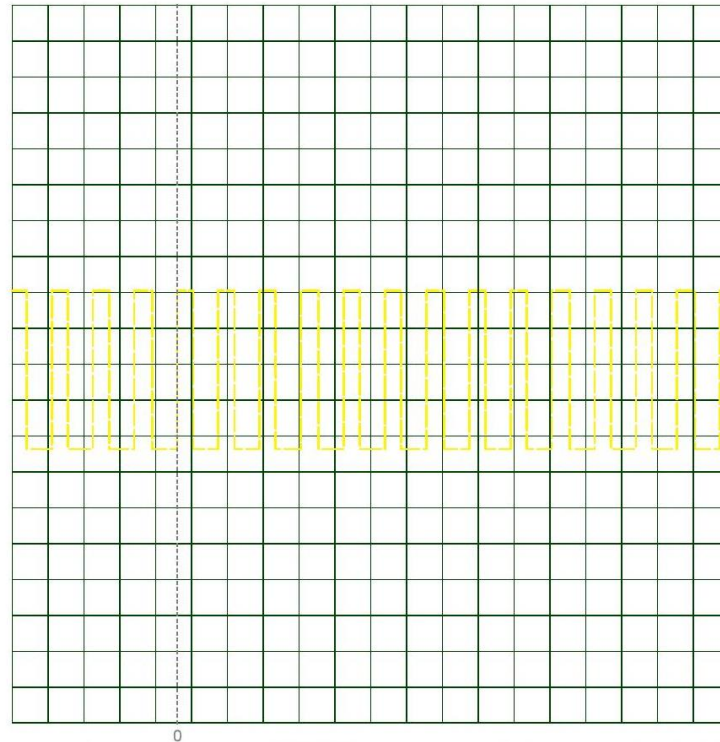


Fig. 1. Circuito en Proteus.

DSO Output



	Channel A	Channel B	Channel C	Channel D
V/Div	500.00 mV	5.00 V	5.00 V	5.00 V
Offset	-600.00 mV	20.00 V	-20.00 V	-60.00 V
Invert	Normal	Normal	Normal	Normal
Coupling	AC	Off	Off	Off
	Horizontal		Trigger	
Source	Trace	Source	Channel A	
Position	8.05 mS	Level	200.00 mV	
S/Div	1.75 mS	Coupling	DC	
		Edge	Rising	
		Mode	Auto	

Fig. 2. Gráfica del osciloscopio en Proteus.