## Programación Microcontrolador Arduino



## Z C3.4 Entradas analógica y salida digital

Arduino, entrada análoga, salida digital y uso de función serial a través de un potenciómetro, una resistencia y un sensor LDR.



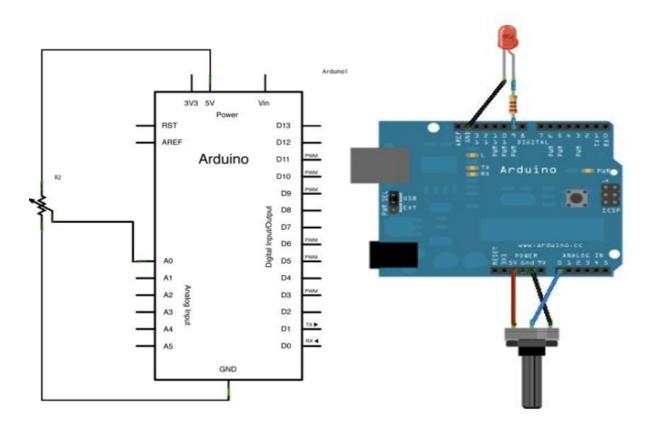
## Instrucciones

- De acuerdo con la información presentada por el asesor referente al tema, desarrollar lo que se indica dentro del apartado siguiente.
- Toda actividad o reto se deberá realizar utilizando el estilo MarkDown con extension .md y el entorno de desarrollo VSCode, debiendo ser elaborado como un documento single page, es decir si el documento cuanta con imágenes, enlaces o cualquier documento externo debe ser accedido desde etiquetas y enlaces.
- Es requisito que el archivo .md contenga una etiqueta del enlace al repositorio de su documento en Github, por ejemplo Enlace a mi GitHub
- Al concluir el reto el reto se deberá subir a github el archivo .md creado.
- Desde el archivo .md se debe exportar un archivo .pdf con la nomenclatura C2.3\_NombreAlumno\_Equipo.pdf, el cual deberá subirse a classroom dentro de su apartado correspondiente, para que sirva como evidencia de su entrega; siendo esta plataforma oficial aquí se recibirá la calificación de su actividad por individual.
- Considerando que el archivo .pdf, fue obtenido desde archivo .md, ambos deben ser idénticos y mostrar el mismo contenido.
- Su repositorio ademas de que debe contar con un archivo **readme**.md dentro de su directorio raíz, con la información como datos del estudiante, equipo de trabajo, materia, carrera, datos del asesor, e incluso logotipo o imágenes, debe tener un apartado de contenidos o indice, los cuales realmente son ligas o enlaces a sus documentos .md, evite utilizar texto para indicar enlaces internos o externo.
- Se propone una estructura tal como esta indicada abajo, sin embargo puede utilizarse cualquier otra que le apoye para organizar su repositorio.

```
readme.md
blog
| C3.1_TituloActividad.md
C3.2_TituloActividad.md
C3.3_TituloActividad.md
 C3.4 TituloActividad.md
| img
docs
| A3.1 TituloActividad.md
| A3.2 TituloActividad.md
```



1. Basado en el siguiente circuito, y utilizando uno de los simuladores propuestos, ensamblar lo que observa.

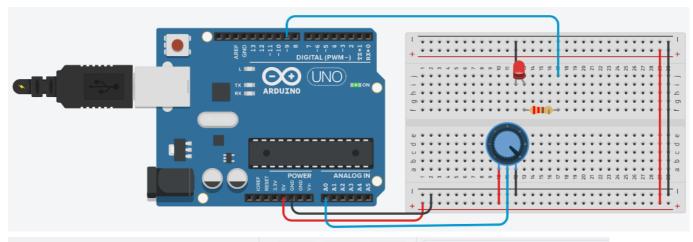


2. Analice y escriba el programa que se muestra a continuación.

```
Analoginput | Arduino 1.8.9
File Edit Sketch Tools Help
             1
  AnalogInput
22
     modified 30 Aug 2011
23
     By Tom Igoe
24
25
    This example code is in the public domain.
26
27
     http://www.arduino.cc/en/Tutorial/AnalogInput
28 */
29
30 int sensorPin = A0; // select the input pin for the potentiometer
31 int ledPin = 13;
                         // select the pin for the LED
32 int sensorValue = 0; // variable to store the value coming from the sensor
33
34 void setup() {
35
     // declare the ledPin as an OUTPUT:
36
    pinMode (ledPin, OUTPUT);
37 }
38
39 void loop() {
40
     // read the value from the sensor:
    sensorValue = analogRead(sensorPin);
41
     // turn the ledPin on
42
    digitalWrite(ledPin, HIGH);
43
44
    // stop the program for <sensorValue> milliseconds:
45
    delay(sensorValue);
46 // turn the ledPin off:
47
    digitalWrite (ledPin, LOW);
48
    // stop the program for for <sensorValue> milliseconds:
49
     delay (sensorValue);
50 }
```

3. Inserte aquí las imágenes que considere como evidencias para demostrar el resultado obtenido.

**R=** Se realizo un programa que encenderia y apagaria un led segun la entrada analogica del potenciometro. Se asigno una entrada analogica al Potenciometro y se asigno una salida al LED. Con esto, el Arduino detectaria la señal del potenciometro para variar el tiempo de encendido y apagado del LED, mientras mas bajo este el potenciometro mas lento sera el LED al prenderse y apagarse, y mientras mas alto este, mas rapido sera el LED en cambiar de estado.



```
Texto
                                                 4 (Arduino Uno R3)
                                <u>*</u>
   int sensorPin = A0;
   //Pin entrada - Potenciometro
   int ledPin = 9;
   //Pin del Led
   int sensorValue = 0;
   //Variable para calcular valor del sensor
 8
   void setup()
 9
10
    pinMode(ledPin, OUTPUT);
11
     //Declarar el pin del LED como salida
12
13
14 void loop()
15
    sensorValue = analogRead(sensorPin);
16
17
    //La variable sensorValue sera igual a lo que la entrada lea
    digitalWrite(ledPin, HIGH);
18
19
    //Entonces, el LED prendera
20
    delay(sensorValue);
21
     //Tardara el tiempo que el potenciometro marque
22
     digitalWrite(ledPin, LOW);
23
    //Despues de ese tiempo, el LED se apagara
24
    delay(sensorValue);
25
    //Y tardara el tiempo que el potenciometro marque apagado
```

## **S** Rubrica

Criterios	Descripción	Puntaje
Instrucciones	Se cumple con cada uno de los puntos indicados dentro del apartado Instrucciones?	20
Desarrollo	Se respondió a cada uno de los puntos solicitados dentro del desarrollo de la actividad?	80

