

Programación Microcontrolador Arduino



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 led.



Instrucciones

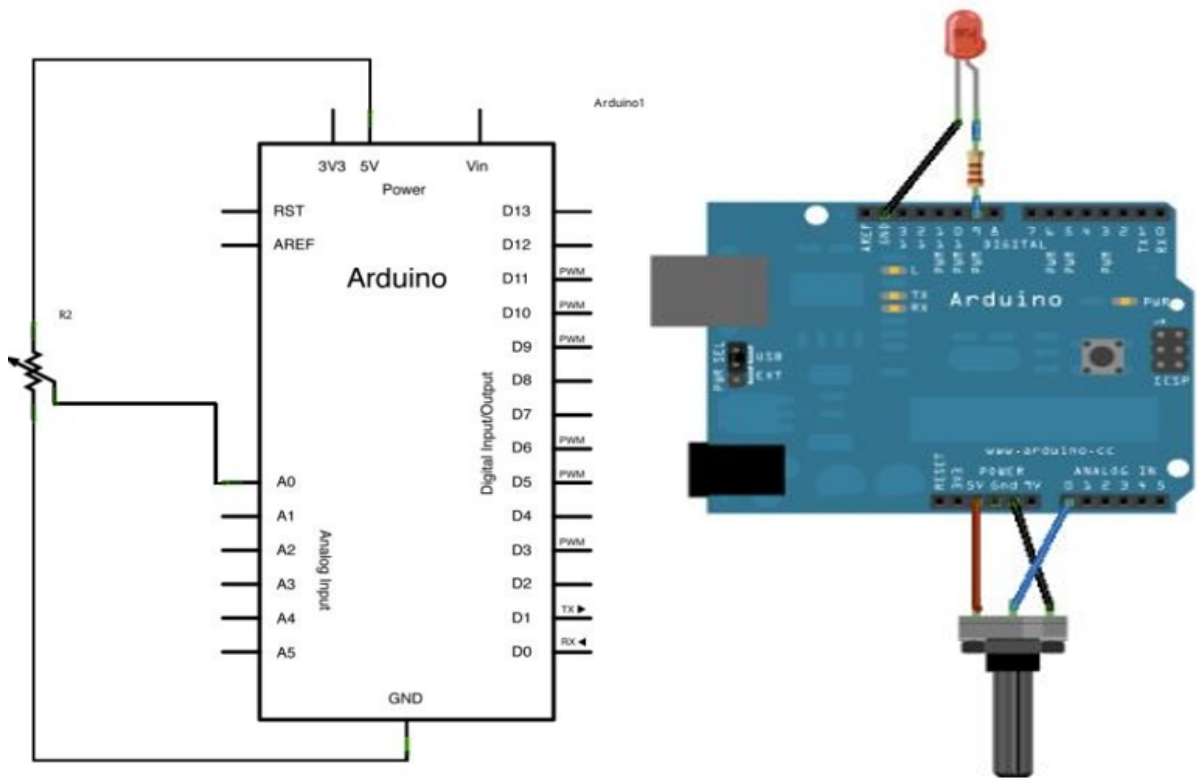
- De acuerdo con la información presentada por el asesor referente al tema, desarrollar lo que se indica dentro del apartado siguiente.
- Toda actividad o reto se deberá realizar utilizando el estilo **Markdown con extension .md** y el entorno de desarrollo VSCode, debiendo ser elaborado como un documento **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
```

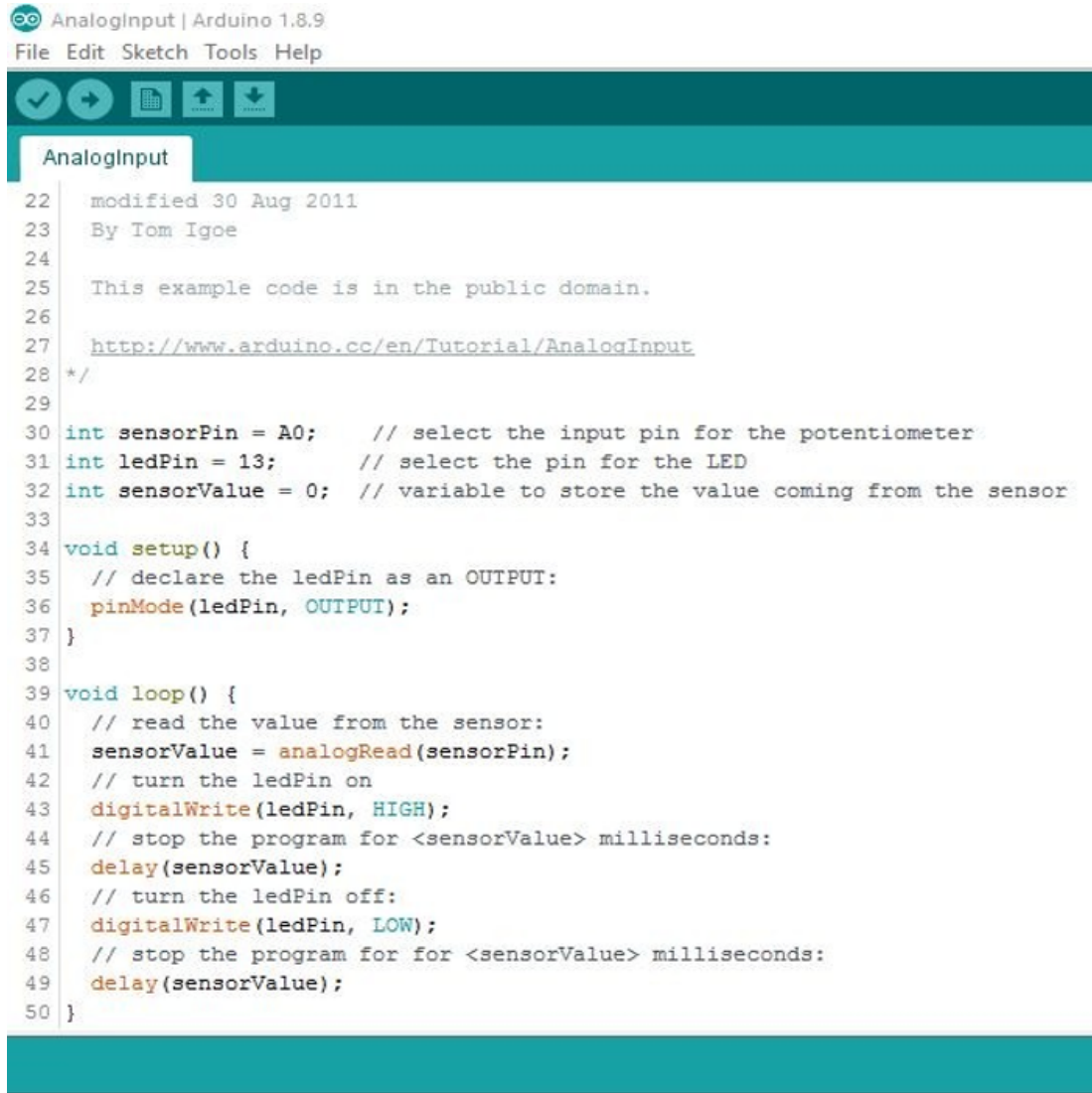


Desarrollo

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.



```

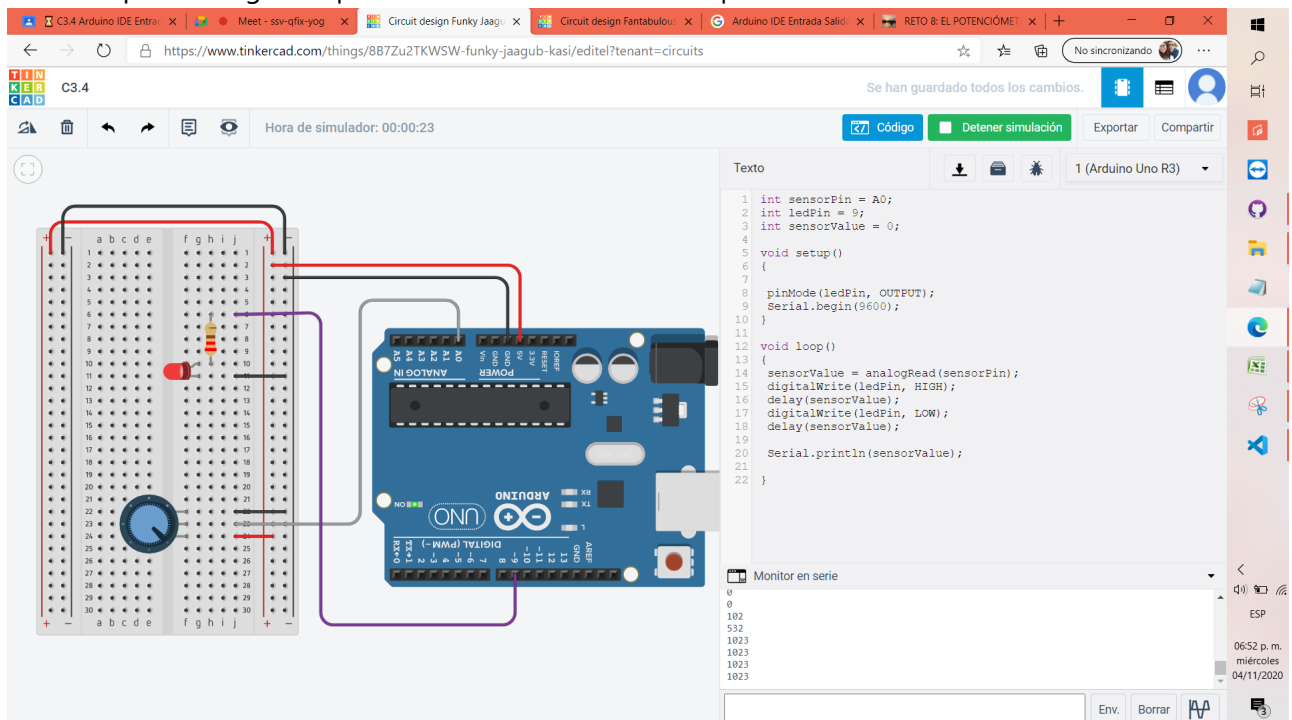
AnalogInput | Arduino 1.8.9
File Edit Sketch Tools Help

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:
41   sensorValue = analogRead(sensorPin);
42   // turn the ledPin on
43   digitalWrite(ledPin, HIGH);
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.



Se han guardado todos los cambios.

Hora de simulador: 00:00:23

Código Detener simulación Exportar Compartir

Texto

```

1 int sensorPin = A0;
2 int ledPin = 9;
3 int sensorValue = 0;
4
5 void setup()
6 {
7   pinMode(ledPin, OUTPUT);
8   Serial.begin(9600);
9 }
10
11
12 void loop()
13 {
14   sensorValue = analogRead(sensorPin);
15   digitalWrite(ledPin, HIGH);
16   delay(sensorValue);
17   digitalWrite(ledPin, LOW);
18   delay(sensorValue);
19
20   Serial.println(sensorValue);
21
22 }

```

Monitor en serie

```

0
102
532
1023
1023
1023
1023

```

Env. Borrar

ESP 06:52 p. m. miércoles 04/11/2020

The screenshot shows the Tinkercad web interface. On the left, an Arduino Uno R3 is connected to a breadboard. A potentiometer is connected to the breadboard, with its wiper connected to an LED. The code on the right is as follows:

```
1 int sensorPin = A0;
2 int ledPin = 9;
3 int sensorValue = 0;
4
5 void setup()
6 {
7   pinMode(ledPin, OUTPUT);
8   Serial.begin(9600);
9 }
10
11 void loop()
12 {
13   sensorValue = analogRead(sensorPin);
14   digitalWrite(ledPin, HIGH);
15   delay(sensorValue);
16   digitalWrite(ledPin, LOW);
17   delay(sensorValue);
18   Serial.println(sensorValue);
19 }
20
21
22
```

The serial monitor shows the following output:

```
1023
1023
1083
511
511
511
511
```

The screenshot shows the Tinkercad web interface with a potentiometer control panel added. The panel has the following settings:

- Nombre: 1
- Resistencia: 50 kΩ

The code on the right is the same as in the first screenshot. The serial monitor shows the following output:

```
20
20
20
20
20
20
20
```



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



[Link a GitHub](#)