



C3.6 Programación Microcontrolador NodeMCU ESP32

Arduino y entrada analógica, utilizando un potenciómetro



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 **C3.6_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.

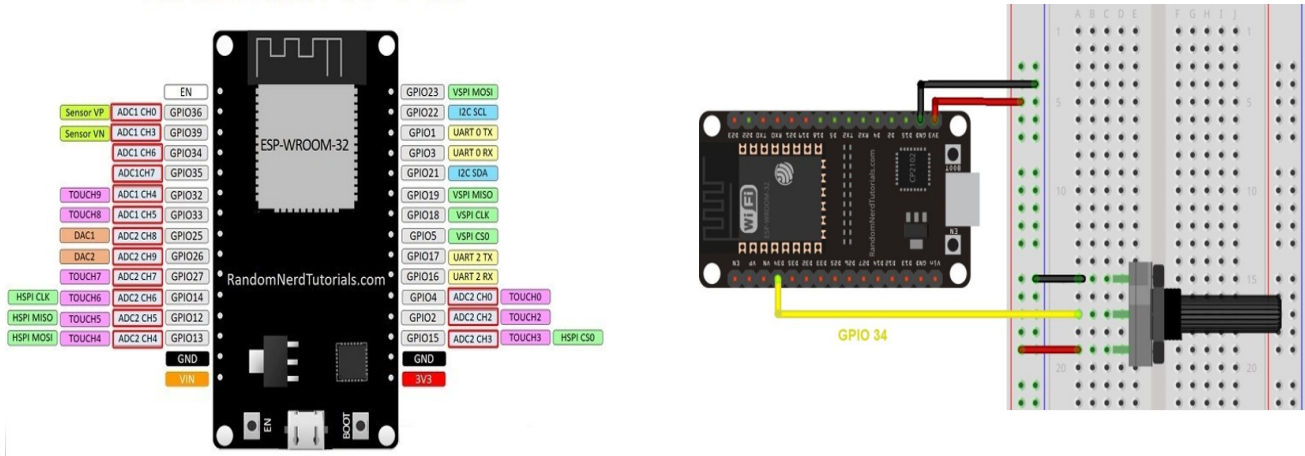
```
| readme.md
| | blog
| | | C3.1_TituloActividad.md
| | | C3.2_TituloActividad.md
| | | C3.3_TituloActividad.md
| | | C3.4_TituloActividad.md
| | | C3.5_TituloActividad.md
| | | C3.6_TituloActividad.md
| | img
| | docs
| | | A3.1_TituloActividad.md
| | | A3.2_TituloActividad.md
```



Desarrollo

1. Ensamble el circuito mostrado en la figura siguiente.

ESP32 DEVKIT V1 - DOIT



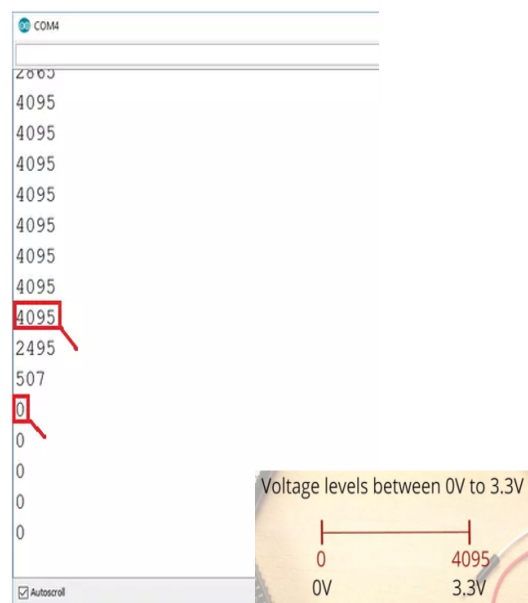
2. Analice y escriba el programa que se muestra a continuación.

```
// Potentiometer is connected to GPIO 34 (Analog ADC1_CH6)
const int potPin = 34;

// variable for storing the potentiometer value
int potValue = 0;

void setup() {
  Serial.begin(115200);
  delay(1000);
}

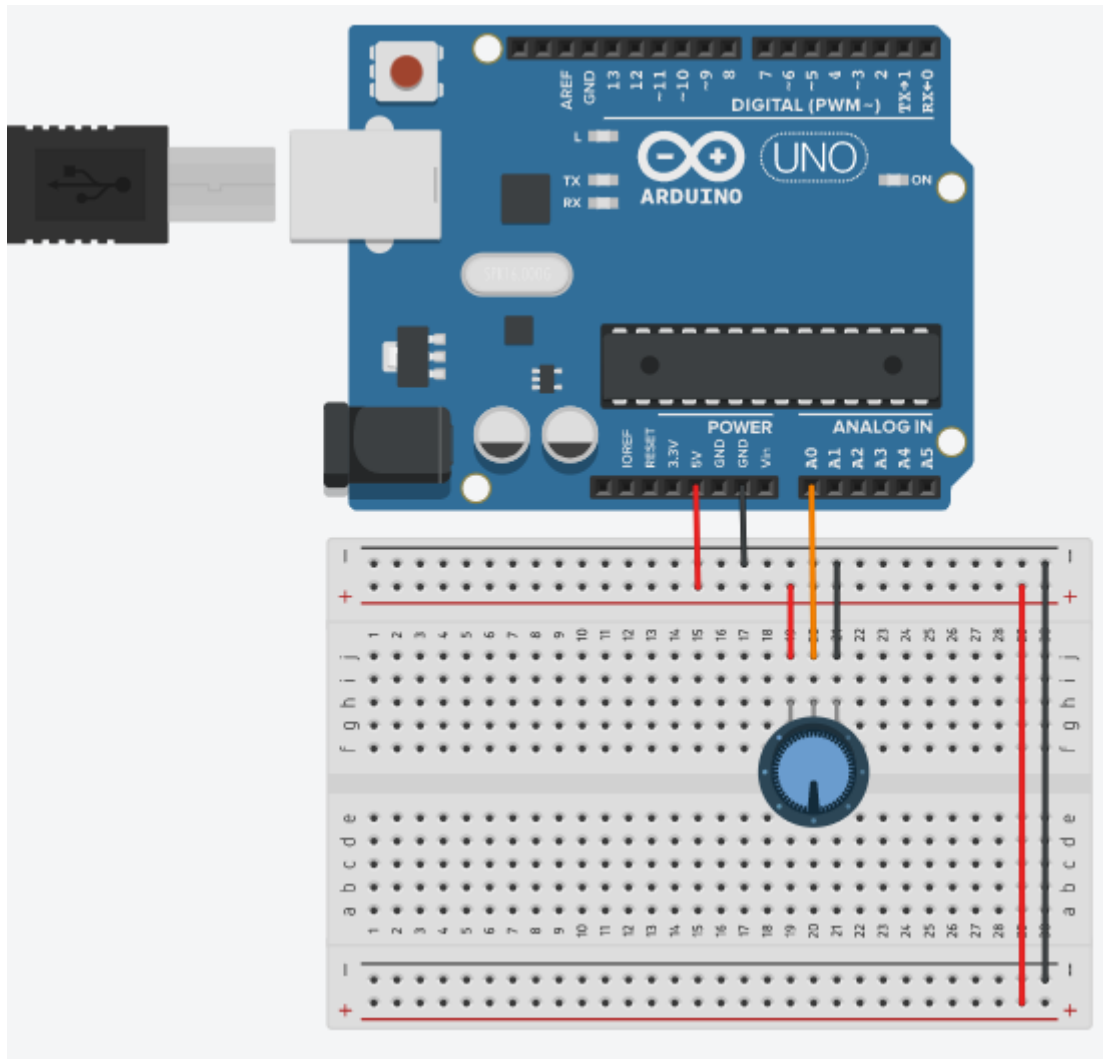
void loop() {
  // Reading potentiometer value
  potValue = analogRead(potPin);
  Serial.println(potValue);
  delay(500);
}
```



Fuente de consulta: [Random Nerd Tutorials](#)

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

Estas evidencias son hechas con el simulador.

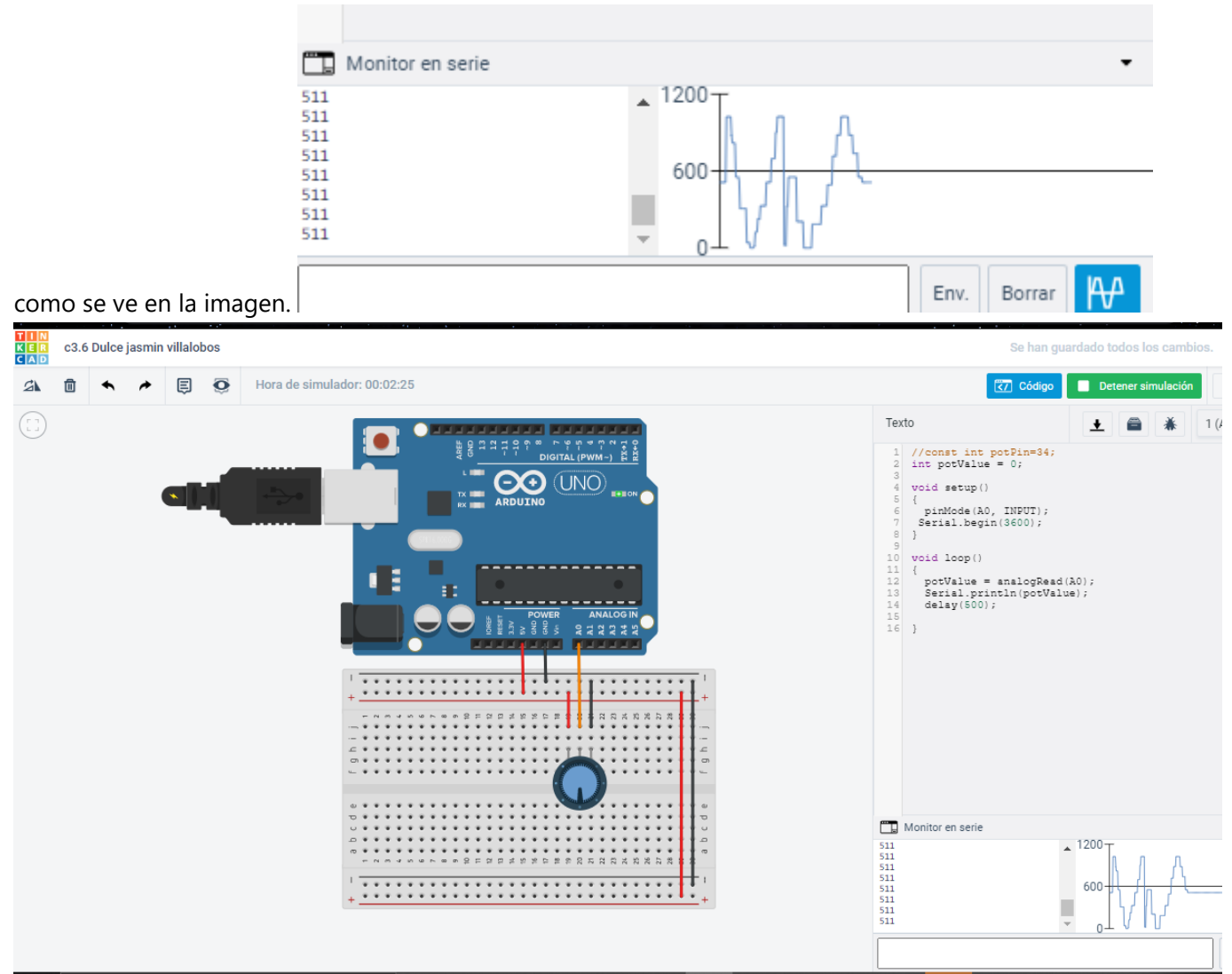


```

2
3 //Declaro variable del valor del ponteciometro
4 int potValue = 0;
5
6 void setup()
7 {
8   //En la instruccion PinMode configuramos el modo de trabajo
9   pinMode(A0, INPUT);
10  // El NO. Pin y el tipo de senal
11  Serial.begin(3600);
12  //Abrimos el puerto serial y
13  //fijamos la velocidad con esta instruccion
14 }
15
16 void loop()
17 {
18   //Asignamos el valor a nuestra variable
19   potValue = analogRead(A0);
20   //Con esta instruccion imprimimos los datos del puerto serial
21   Serial.println(potValue);
22   delay(500);
23 }
24

```

Con el ponteciometro de va cambiando el valor que se imprime en la grafica , el nivel del voltaje varia, y se imprime la señal analogiaca



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

 [ENLACE - MI GITHUB](#)