



C3.8 Programación Microcontrolador NodeMCU ESP32

Arduino y sensor de tacto integrado al NodeMCU ESP32



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 **C3.8_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.

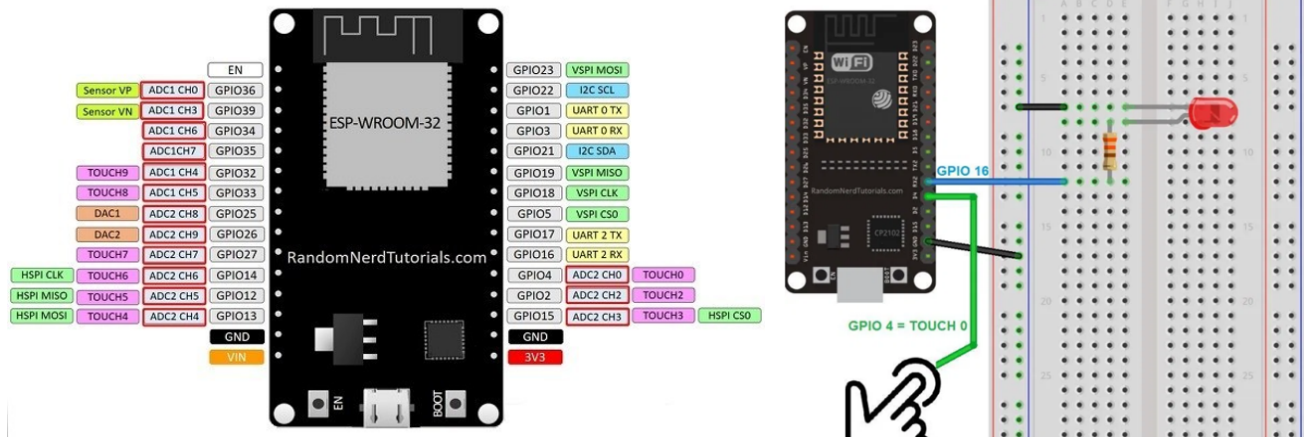
```
| readme.md
| | blog
| | | C3.1_TituloActividad.md
| | | C3.2_TituloActividad.md
| | | C3.3_TituloActividad.md
| | | C3.4_TituloActividad.md
| | | C3.5_TituloActividad.md
| | | C3.6_TituloActividad.md
| | | C3.7_TituloActividad.md
| | | C3.8_TituloActividad.md
| | img
| | docs
| | | A3.1_TituloActividad.md
| | | A3.2_TituloActividad.md
```



Desarrollo

1. Basado en el siguiente circuito y ensamblarlo, utilizando alguno de los simulados propuesto, explicando el resultado que se desea obtener del mismo.

ESP32 DEVKIT V1 - DOIT



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

```
// set pin numbers
const int touchPin = 4;
const int ledPin = 16;

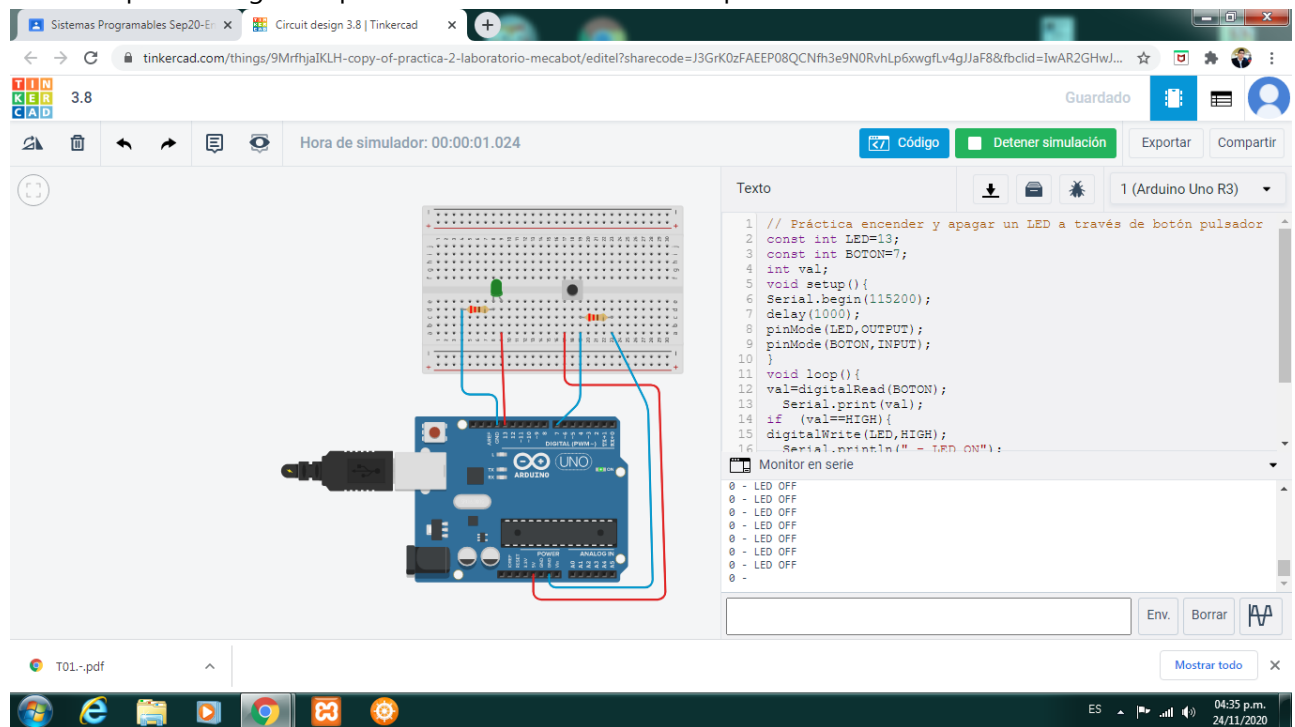
// change with your threshold value
const int threshold = 20;
// variable for storing the touch pin value
int touchValue;

void setup(){
  Serial.begin(115200);
  delay(1000); // give me time to bring up serial monitor
  // initialize the LED pin as an output:
  pinMode (ledPin, OUTPUT);
}

void loop(){
  // read the state of the pushbutton value:
  touchValue = touchRead(touchPin);
  Serial.print(touchValue);
  // check if the touchValue is below the threshold
  // if it is, set ledPin to HIGH
  if(touchValue < threshold){
    // turn LED on
    digitalWrite(ledPin, HIGH);
    Serial.println(" - LED on");
  }
  else{
    // turn LED off
    digitalWrite(ledPin, LOW);
    Serial.println(" - LED off");
  }
  delay(500);
}
```

Fuente de consulta: [Random Nerd Tutorials](https://randomnerdtutorials.com/)

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

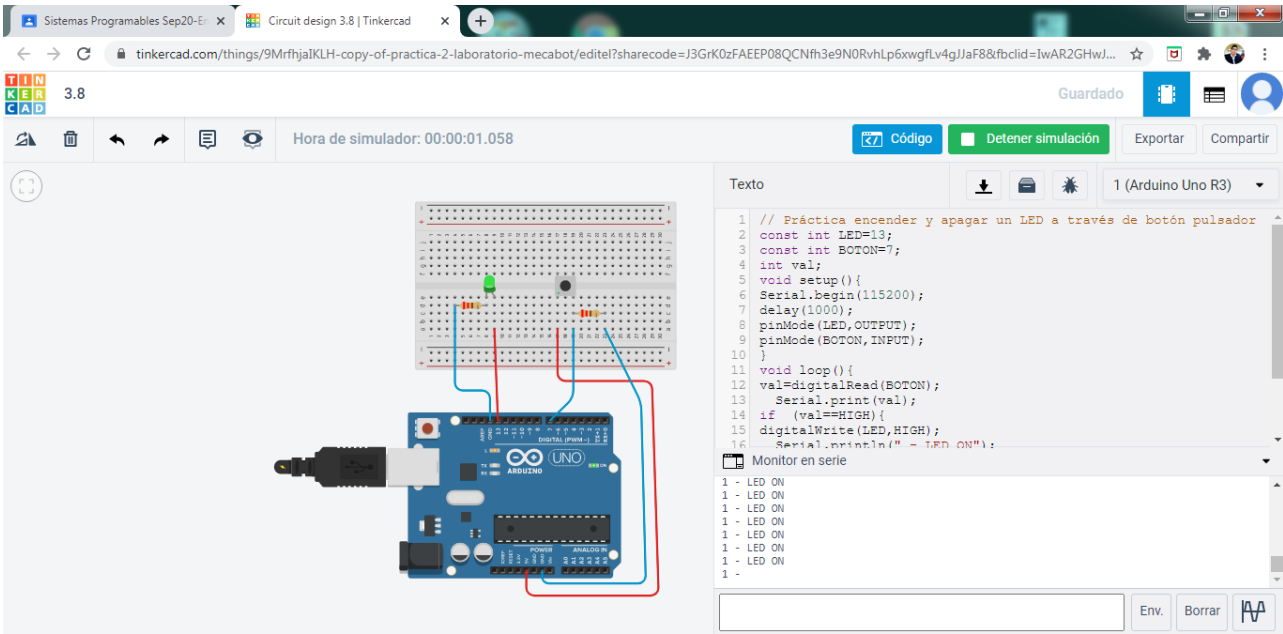


The screenshot displays the Tinkercad web interface. On the left, an Arduino Uno R3 is connected to a breadboard. A push button is connected to the breadboard, and an LED is also connected. The code on the right is as follows:

```
1 // Práctica encender y apagar un LED a través de botón pulsador
2 const int LED=13;
3 const int BOTON=7;
4 int val;
5 void setup() {
6   Serial.begin(115200);
7   delay(1000);
8   pinMode(LED,OUTPUT);
9   pinMode(BOTON,INPUT);
10 }
11 void loop() {
12   val=digitalRead(BOTON);
13   Serial.print(val);
14   if (val==HIGH){
15     digitalWrite(LED,HIGH);
16     Serial.println(" - LED ON");
17   }
18 }
```

The serial monitor shows the following output:

```
0 - LED OFF
0 - LED OFF
0 - LED OFF
0 - LED OFF
0 - LED OFF
0 - LED OFF
0 - LED OFF
0 -
```

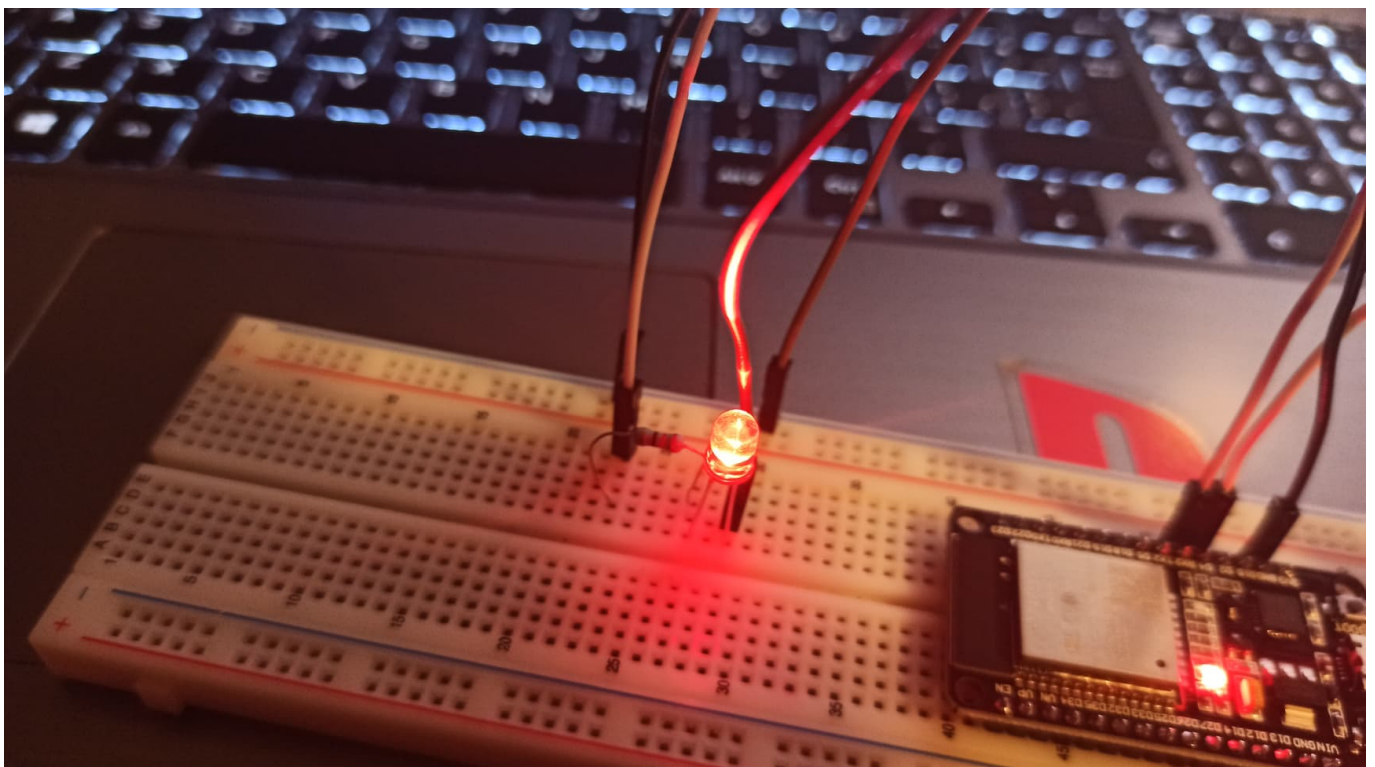
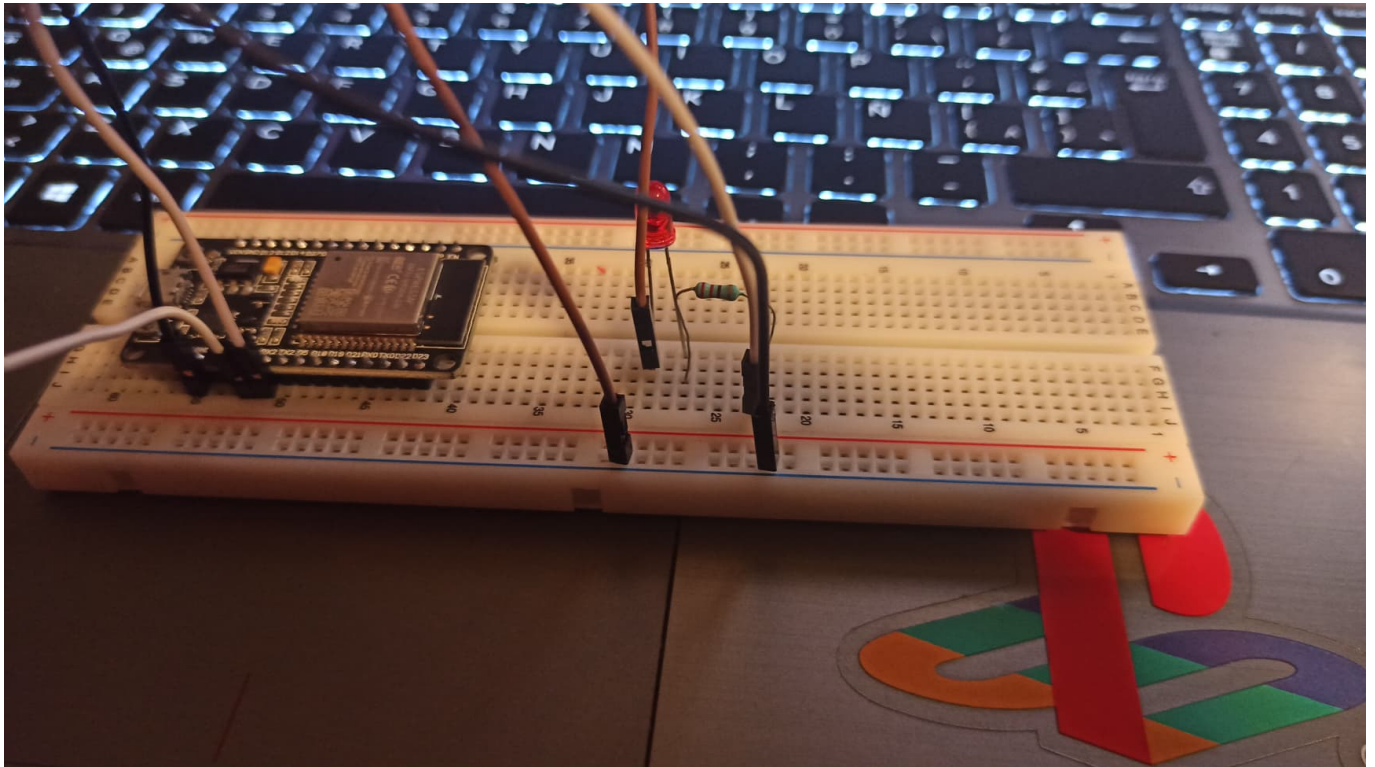
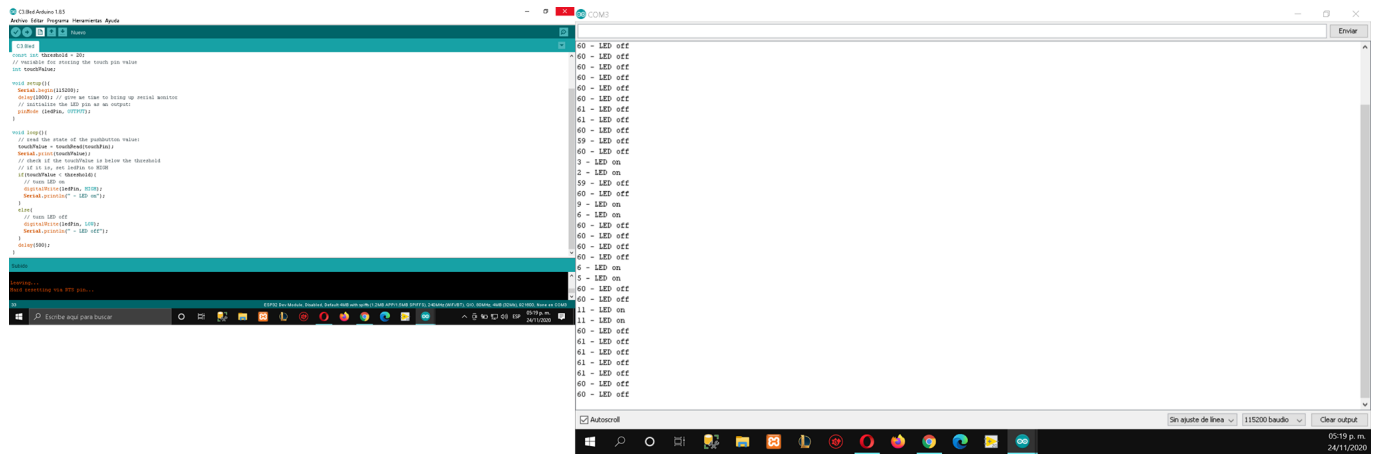


```
1 // Práctica encender y apagar un LED a través de botón pulsador
2 const int LED=13;
3 const int BOTON=7;
4 int val;
5 void setup() {
6   Serial.begin(115200);
7   delay(1000);
8   pinMode(LED, OUTPUT);
9   pinMode(BOTON, INPUT);
10 }
11 void loop() {
12   val=digitalRead(BOTON);
13   Serial.print(val);
14   if (val==HIGH) {
15     digitalWrite(LED,HIGH);
16     Serial.println(" - LED ON");
17   }
18   else { digitalWrite(LED,LOW);
19     Serial.println(" - LED OFF");
20   }
21 }
```

Monitor en serie

```
1 - LED ON
1 - LED ON
1 - LED ON
1 - LED ON
1 - LED ON
1 - LED ON
1 - LED ON
1 - LED ON
```

Evidencia de la Practica con el ESP32 realizada por mi compañero Alejandro Diaz Navarro



[Video Demostrativo](#)

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 a mi GitHub](#)