



# C4.1 Programación Microcontrolador NodeMCU ESP32

---

Comunicación por medio del protocolo Bluetooth y el NodeMCU ESP32



## Instrucciones

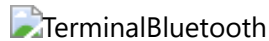
- De acuerdo con la información presentada por el asesor referente al tema, desarrollar lo que se indica dentro del apartado siguiente.
- Toda actividad o reto se deberá realizar utilizando el estilo **Markdown con extension .md** y el entorno de desarrollo VSCode, debiendo ser elaborado como un documento **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 **C4.1\_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
| | | C4.1_TituloActividad.md
| | | C4.2_TituloActividad.md
| | | C4.3_TituloActividad.md
| | | C4.4_TituloActividad.md
| | | C4.5_TituloActividad.md
| | img
| | docs
| | | A4.1_TituloActividad.md
| | | A4.2_TituloActividad.md
```



## Desarrollo

1. Busque en Play Store la app "Serial Bluetooth terminal" e instale a su dispositivo movil.



2. Apóyese del siguiente circuito, colocando una resistencia y un diodo led en serie a una de las terminales del Nodemcu ESP32 de tal forma que puede general una señal digital.



3. Analice y utilice el código que se indica a continuación para emparejar su dispositivo movil con el dispositivo ESP32 via bluetooth

```
//This example code is in the Public Domain (or CC0 licensed, at your option.)
//By Evandro Copercini - 2018
//
//This example creates a bridge between Serial and Classical Bluetooth (SPP)
//and also demonstrate that SerialBT have the same functionalities of a normal
Serial

#include "BluetoothSerial.h"

#if !defined(CONFIG_BT_ENABLED) || !defined(CONFIG_BLUEDROID_ENABLED)
#error Bluetooth is not enabled! Please run `make menuconfig` to and enable it
#endif

BluetoothSerial SerialBT;

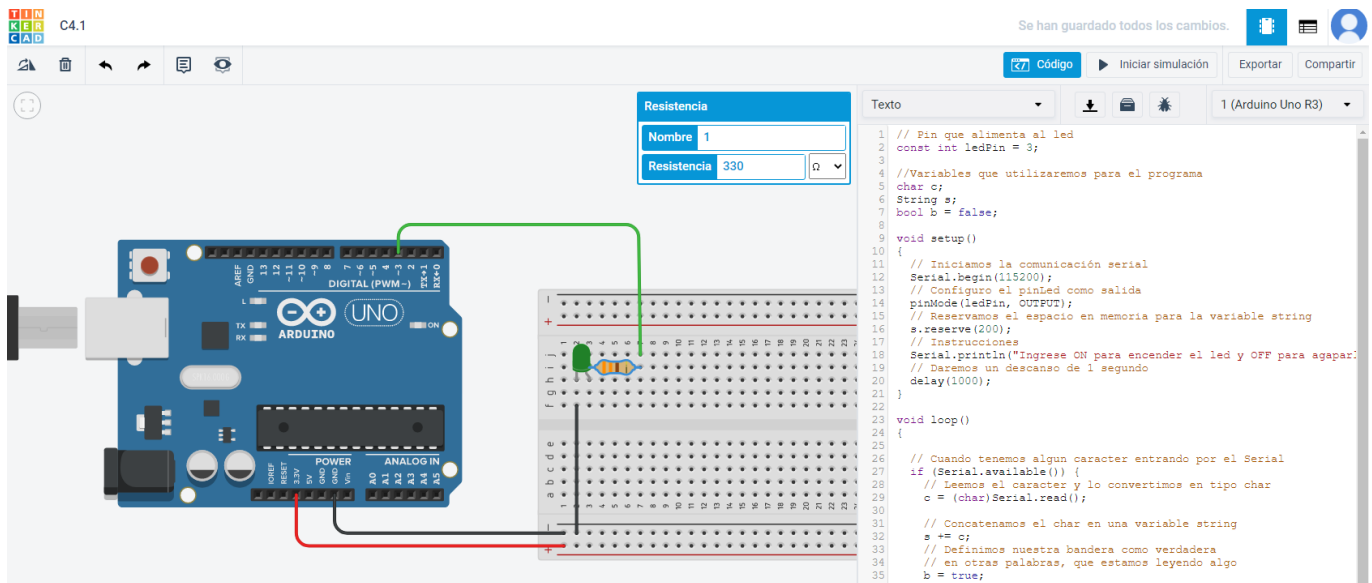
void setup() {
  Serial.begin(115200);
  SerialBT.begin("ESP32test"); //Bluetooth device name
  Serial.println("The device started, now you can pair it with bluetooth!");
}

void loop() {
  if (Serial.available()) {
    SerialBT.write(Serial.read());
  }
  if (SerialBT.available()) {
    Serial.write(SerialBT.read());
  }
  delay(20);
}
```

4. Compile y cargue el programa a su dispositivo NodeMCU-ESP32
5. Empareje su dispositivo movil con el ESP32, y utilice la app bajada del play store para establecer la comunicación.
6. Abra la terminal serial del IDE de Arduino, y desde el movil envíe cualquier instrucción la cual se deberá observar en la terminal serial del IDE Arduino.
7. Una vez realizado lo anterior, agregue instrucciones al programa para que le indique al ESP32 que encienda o apague el LED que ha sido colocado en el circuito desde su dispositivo movil y la app

instalada.

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



Resistencia

Nombre 1

Resistencia 330  $\Omega$  ▼

Texto

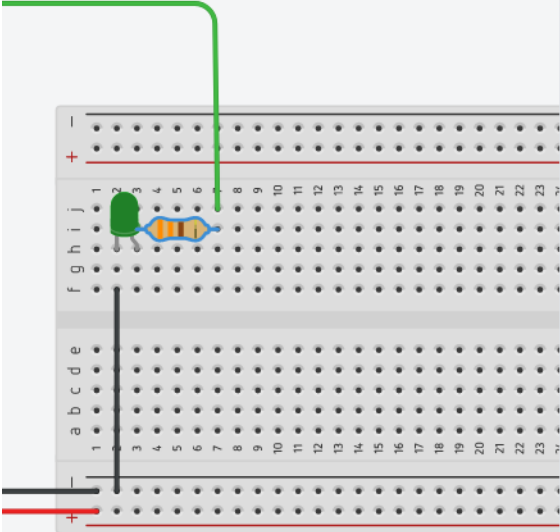
1 (Arduino Uno R3)

```
1 // Pin que alimenta al led
2 const int ledPin = 3;
3
4 //Variables que utilizaremos para el programa
5 char c;
6 String s;
7 bool b = false;
8
9 void setup()
10 {
11   // Iniciamos la comunicación serial
12   Serial.begin(115200);
13   // Configuro el pinLed como salida
14   pinMode(ledPin, OUTPUT);
15   // Reservamos el espacio en memoria para la variable string
16   s.reserve(200);
17   // Instrucciones
18   Serial.println("Ingrese ON para encender el led y OFF para apagarlo");
19   // Daremos un descanso de 1 segundo
20   delay(1000);
21 }
22
23 void loop()
24 {
25
26   // Cuando tenemos algun caracter entrando por el Serial
27   if (Serial.available()) {
28     // Leemos el caracter y lo convertimos en tipo char
29     c = (char)Serial.read();
30
31     // Concatenamos el char en una variable string
32     s += c;
33     // Definimos nuestra bandera como verdadera
34     // en otras palabras, que estamos leyendo algo
35     b = true;
36   }
37
38   // Cuando no estamos leyendo y nuestra bandera es verdadera
39   // significa que recién terminamos de leer todo lo del Serial
40   if(!Serial.available() && b == true){
41     // Imprimimos lo leído
42     Serial.println(s);
43     // Si lo leído es ON, encendemos el led
44     if(s == "ON"){
45       digitalWrite(ledPin, HIGH);
46     }
47     // Si lo leído es OFF, apagamos el led
48     if(s == "OFF"){
49       digitalWrite(ledPin, LOW);
50     }
51     // Declaramos la bandera false, pues terminamos de imprimir
52     b = false;
53     // Vaciamos la variable de lectura
54     s = "";
55   }
56   // Damos un descanso de 20 milisegundos
57   delay(20);
58 }
```

Resistencia

Nombre 1

Resistencia 330  $\Omega$



Código

Detener simulación

Exportar

Compartir

1 (Arduino Uno R3)

Texto

```
12 // Configuración de pines
13 // Configuro el pinLed como salida
14 pinMode(ledPin, OUTPUT);
15 // Reservamos el espacio en memoria para la variable string
16 s.reserve(200);
17 // Instrucciones
18 Serial.println("Ingrese ON para encender el led y OFF para apagarlo.");
19 // Daremos un descanso de 1 segundo
20 delay(1000);
21 }
22
23 void loop()
24 {
25
26 // Cuando tenemos algun caracter entrando por el Serial
27 if (Serial.available()) {
28 // Leemos el caracter y lo convertimos en tipo char
29 c = (char)Serial.read();
30
31 // Concatenamos el char en una variable string
32 s += c;
33 // Definimos nuestra bandera como verdadera
34 // en otras palabras, que estamos leyendo algo
35 b = true;
36 }
37
38 // Cuando no estamos leyendo y nuestra bandera es verdadera
39 // significa que recién terminamos de leer todo lo del Serial
40 if(!Serial.available() && b == true){
41 // Imprimimos lo leído
42 Serial.println(s);
43 // Si lo leído es ON, encendemos el led
44 if(s == "ON"){
45 digitalWrite(ledPin, HIGH);
46 }
47 // Si lo leído es OFF, apagamos el led
48 if(s == "OFF"){
49 
```

Monitor en serie

Ingrese ON para encender el led y OFF para apagarlo...

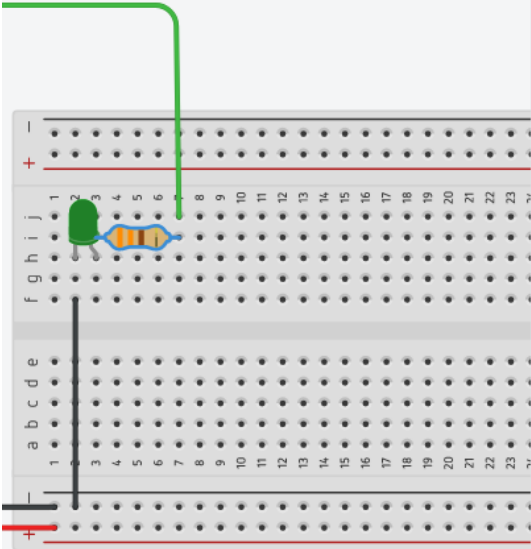
Hola

Env. Borrar

Resistencia

Nombre 1

Resistencia 330  $\Omega$



Código

Detener simulación

Exportar

Compartir

1 (Arduino Uno R3)

Texto

```
12 // Configuración de pines
13 // Configuro el pinLed como salida
14 pinMode(ledPin, OUTPUT);
15 // Reservamos el espacio en memoria para la variable string
16 s.reserve(200);
17 // Instrucciones
18 Serial.println("Ingrese ON para encender el led y OFF para apagarlo");
19 // Daremos un descanso de 1 segundo
20 delay(1000);
21 }
22
23 void loop()
24 {
25
26 // Cuando tenemos algun caracter entrando por el Serial
27 if (Serial.available()) {
28 // Leemos el caracter y lo convertimos en tipo char
29 c = (char)Serial.read();
30
31 // Concatenamos el char en una variable string
32 s += c;
33 // Definimos nuestra bandera como verdadera
34 // en otras palabras, que estamos leyendo algo
35 b = true;
36 }
37
38 // Cuando no estamos leyendo y nuestra bandera es verdadera
39 // significa que recién terminamos de leer todo lo del Serial
40 if(!Serial.available() && b == true){
41 // Imprimimos lo leído
42 Serial.println(s);
43 // Si lo leído es ON, encendemos el led
44 if(s == "ON"){
45 digitalWrite(ledPin, HIGH);
46 }
47 // Si lo leído es OFF, apagamos el led
48 if(s == "OFF"){
49 }
```

Monitor en serie

Ingrese ON para encender el led y OFF para apagarlo...

Hola

Env. Borrar

Resistencia

Nombre 1

Resistencia 330  $\Omega$

Código

Detener simulación

Exportar

Compartir

1 (Arduino Uno R3)

Texto

```
13 // Configuro el pinLed como salida
14 pinMode(ledPin, OUTPUT);
15 // Reservamos el espacio en memoria para la variable string
16 s.reserve(200);
17 // Instrucciones
18 Serial.println("Ingrese ON para encender el led y OFF para agapar!
19 // Daremos un descanso de 1 segundo
20 delay(1000);
21 }
22
23 void loop()
24 {
25
26 // Cuando tenemos algun caracter entrando por el Serial
27 if (Serial.available()) {
28 // Leemos el caracter y lo convertimos en tipo char
29 c = (char)Serial.read();
30
31 // Concatenamos el char en una variable string
32 s += c;
33 // Definimos nuestra bandera como verdadera
34 // en otras palabras, que estamos leyendo algo
35 b = true;
36 }
37
38 // Cuando no estamos leyendo y nuestra bandera es verdadera
39 // significa que recién terminamos de leer todo lo del Serial
40 if(!Serial.available() && b == true){
41 // Imprimimos lo leído
42 Serial.println(s);
43 // Si lo leído es ON, encendemos el led
44 if(s == "ON"){
45 digitalWrite(ledPin, HIGH);
46 }
47 // Si lo leído es OFF, apagamos el led
48 if(s == "OFF"){
49
```

Monitor en serie

Ingrese ON para encender el led y OFF para agaparlo...  
Hola  
ON

Env.

Borrar

Resistencia

Nombre 1

Resistencia 330  $\Omega$

Texto

1 (Arduino Uno R3)

```
13 // Configuro el pinLed como salida
14 pinMode(ledPin, OUTPUT);
15 // Reservamos el espacio en memoria para la variable string
16 s.reserve(200);
17 // Instrucciones
18 Serial.println("Ingrese ON para encender el led y OFF para agapar:");
19 // Daremos un descanso de 1 segundo
20 delay(1000);
21 }
22
23 void loop()
24 {
25
26 // Cuando tenemos algun caracter entrando por el Serial
27 if (Serial.available()) {
28 // Leemos el caracter y lo convertimos en tipo char
29 c = (char)Serial.read();
30
31 // Concatenamos el char en una variable string
32 s += c;
33 // Definimos nuestra bandera como verdadera
34 // en otras palabras, que estamos leyendo algo
35 b = true;
36 }
37
38 // Cuando no estamos leyendo y nuestra bandera es verdadera
39 // significa que recién terminamos de leer todo lo del Serial
40 if(!Serial.available() && b == true){
41 // Imprimimos lo leído
42 Serial.println(s);
43 // Si lo leído es ON, encendemos el led
44 if(s == "ON"){
45 digitalWrite(ledPin, HIGH);
46 }
47 // Si lo leído es OFF, apagamos el led
48 if(s == "OFF"){
49 }
```

Monitor en serie

Ingrese ON para encender el led y OFF para agaparlo...  
Hola  
ON  
OFF

Env. Borrar

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

[Ir a readme](#)

[Ver en repositorio](#)