



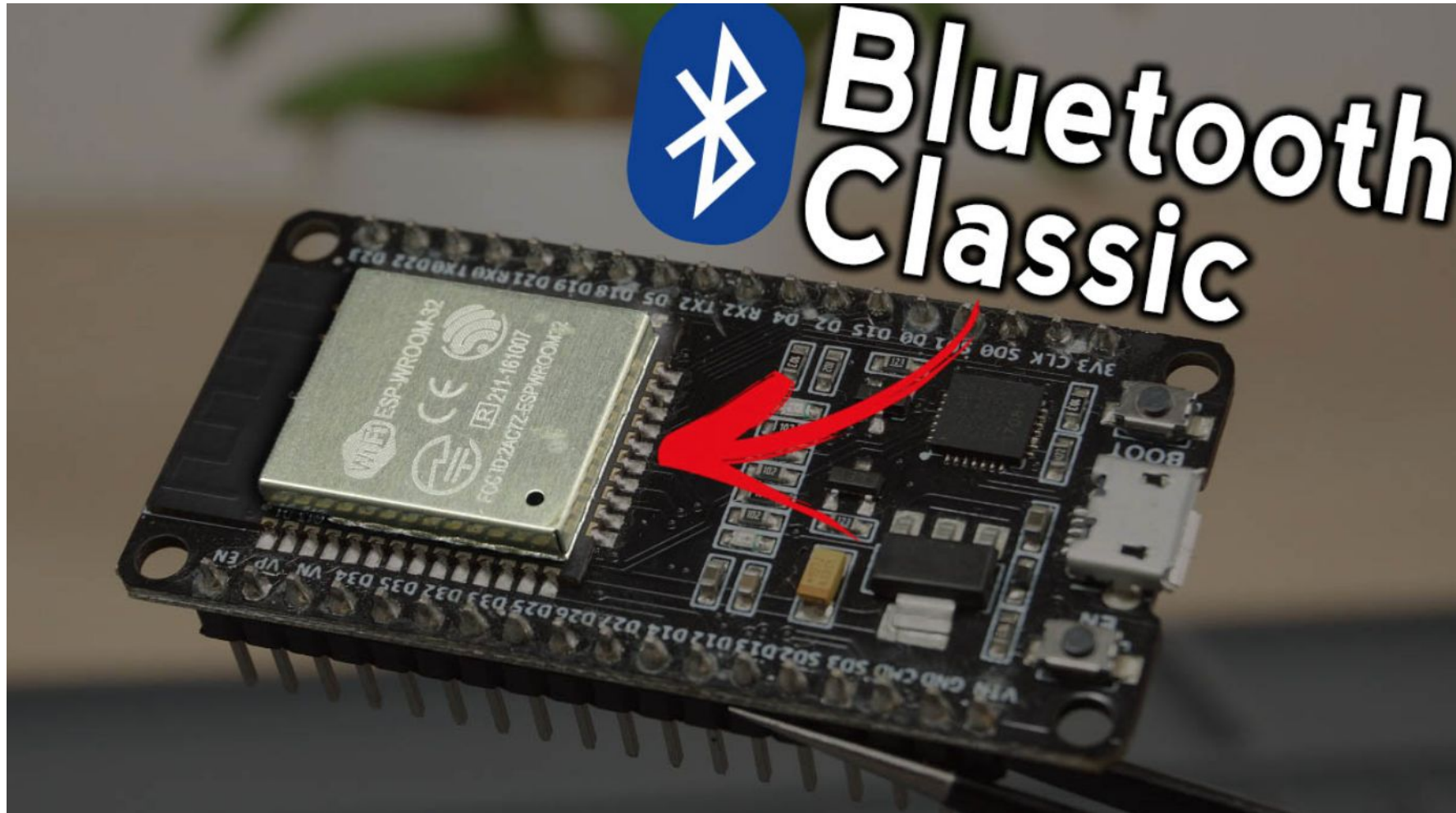
# Microcontroladores II

**Juan Esteban Giraldo Hoyos**

Ingeniero Electrónico

Magíster en Gestión de Ciencia, Tecnología e Innovación

# Microcontroladores II - Hoy trabajaremos



# Microcontroladores II - Comunicación Bluetooth

Versión Bluetooth	Fecha de Lanzamiento	Velocidad de Transferencia de Datos (Máx.)	Alcance (Máx.)	Mejoras Significativas
Bluetooth 1.0	1999	1 Mbps	10 metros	-
Bluetooth 2.0 + EDR	2004	3 Mbps	10 metros	EDR (Enhanced Data Rate)
Bluetooth 3.0 + HS	2009	24 Mbps (HS)	100 metros	HS (High Speed)
Bluetooth 4.0	2010	25 Mbps	60 metros	Bajo consumo de energía, BLE (Bluetooth Low Energy)
Bluetooth 4.1	2013	25 Mbps	100 metros	Mejoras en la privacidad y la conectividad
Bluetooth 4.2	2014	25 Mbps	100 metros	Mayor seguridad y eficiencia energética
Bluetooth 5.0	2016	50 Mbps	240 metros	Mayor velocidad, alcance y capacidad publicitaria
Bluetooth 5.1	2019	-	-	Mejoras en la localización y rastreo
Bluetooth 5.2	2020	50 Mbps ↓	240 metros	Mejoras en la seguridad y la eficiencia energética

# Microcontroladores II - Comunicación Bluetooth

Vamos a implementar un programa que se comuniqué por bluetooth con otro dispositivo y que muestre por el monitor serial los mensajes. También transmite lo que reciba por la comunicación serial hacia el dispositivo por bluetooth

## FREERTOS\_BLUETOOTH

```
#include <Arduino.h>
#include "BluetoothSerial.h"

// #define USE_PIN // Uncomment this to use PIN during pairing. The pin is specified on the line below
const char *pin = "1234"; // Change this to a more secure PIN.
String device_name = "ESP32-BT-Slave";

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

#if !defined(CONFIG_BT_SPP_ENABLED)
#error Serial Bluetooth not available or not enabled. It is only available for the ESP32 chip.
#endif

BluetoothSerial SerialBT;

void serialToBluetoothTask(void *pvParameters) {
    (void) pvParameters;
    while (1) {
        if (Serial.available()) {
            SerialBT.write(Serial.read());
        }
        vTaskDelay(pdMS_TO_TICKS(20));
    }
}
```



# Microcontroladores II - Comunicación Bluetooth

```
void bluetoothToSerialTask(void *pvParameters) {
    (void) pvParameters;
    while (1) {
        if (SerialBT.available()) {
            Serial.write(SerialBT.read());
        }
        vTaskDelay(pdMS_TO_TICKS(20));
    }
}

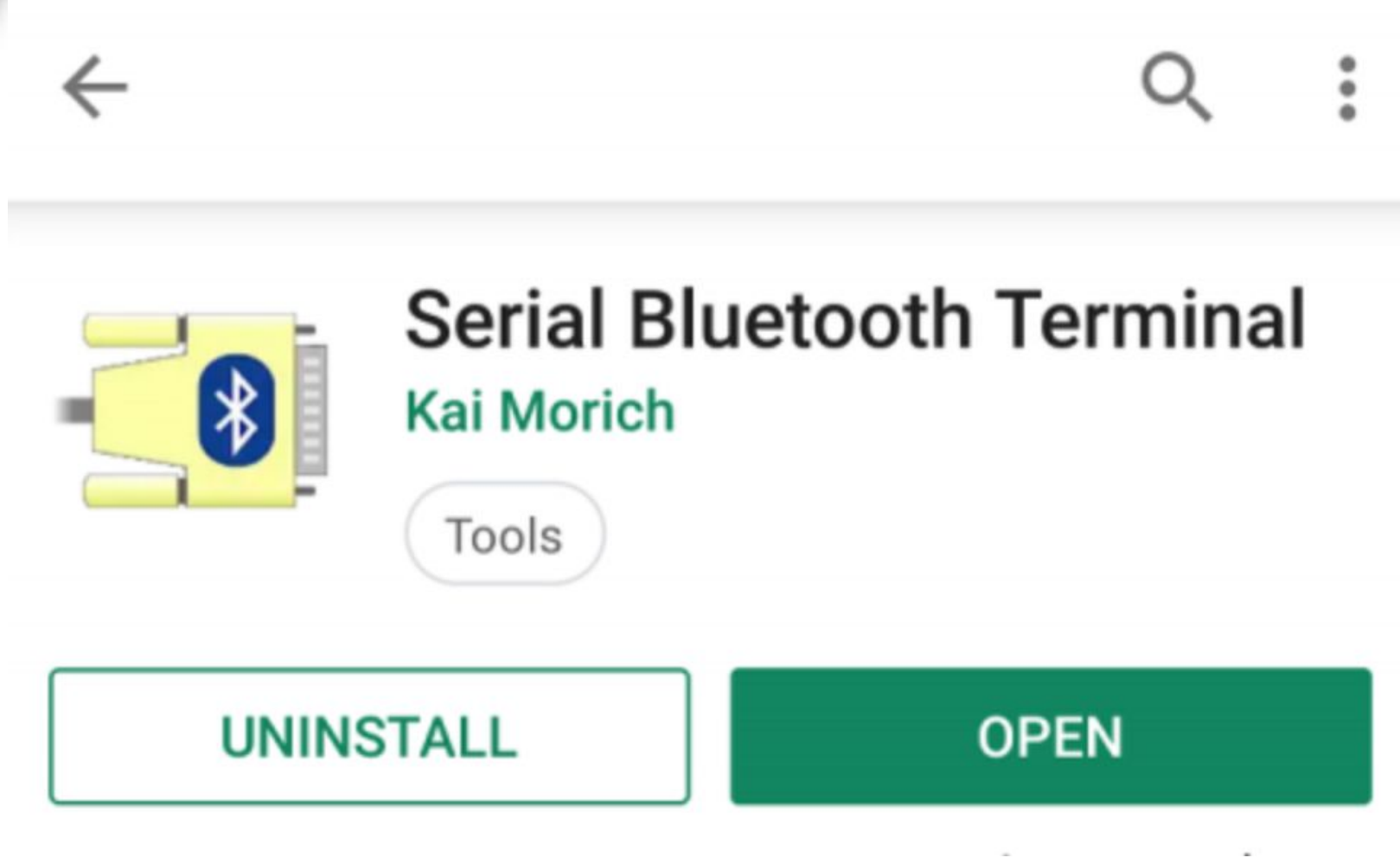
void setup() {
    Serial.begin(115200);
    SerialBT.begin(device_name); //Bluetooth device name
    Serial.printf("The device with name \"%s\" is started.\nNow you can pair it with Bluetooth!\n", device_name.c_str());
    #ifdef USE_PIN
        SerialBT.setPin(pin);
        Serial.println("Using PIN");
    #endif

    xTaskCreatePinnedToCore(serialToBluetoothTask, "SerialToBluetoothTask", 4096, NULL, 1, NULL, 1);
    xTaskCreatePinnedToCore(blueetoothToSerialTask, "BluetoothToSerialTask", 4096, NULL, 1, NULL, 1);
}

void loop() {
    // Empty, all tasks are managed in FreeRTOS tasks
}
```

# Microcontroladores II - Comunicación Bluetooth

Descargar una app cómo la siguiente para comunicarnos desde un smartphone con nuestro ESP32 por bluetooth



# Microcontroladores II - Comunicación Bluetooth

/dev/cu.usbserial-0001

ets Jun 8 2016 00:22:57

rst:0x1 (POWERON\_RESET),boot:0x13 (SPI\_FAST\_FLASH\_BOOT)

ets Jun 8 2016 00:22:57

rst:0x10 (RTCWDT\_RTC\_RESET),boot:0x13 (SPI\_FAST\_FLASH\_BOOT)

config: 0, SPIWP:0xee

clk\_drv:0x00,q\_drv:0x00,d\_drv:0x00,cs0\_drv:0x00,hd\_drv:0x00,wp\_drv:0x00

mode:DI0, clock div:1

load:0x3fff0030,len:1344

load:0x40078000,len:13964

load:0x40080400,len:3600

entry 0x400805f0

The device with name "ESP32-BT-Slave" is started.

Now you can pair it with Bluetooth!

☒ Autoscroll ☐ Mostrar marca temporal

Retorno de carro



115200 baudio



Escuela de  
Ingenierías

IUSH  
Institución Universitaria  
SALAZAR Y HERRERA

SER PERSONA DESDE EL SABER Y EL HACER

# Microcontroladores II - Comunicación Bluetooth

```
Terminal
Gracias por la bienvenida

16:50:40.194 Connecting to ESP32-BT-Slave ...
16:50:40.630 Connected
16:50:59.671 Bienvenid@ ingenier@
16:51:16.686 Gracias por la bienvenida

ON OFF M3 M4 M5 M6
Bienvenid@ ingenier@
```

```
/dev/cu.usbseri

Gracias por la bienvenida

rst:0x1 (POWERON_RESET),boot:0x13 (SPI_FAST_FLASH_BOOT)
ets Jun 8 2016 00:22:57

rst:0x10 (RTCWDT_RTC_RESET),boot:0x13 (SPI_FAST_FLASH_BOOT)
config: 0, SPIWP:0xee
clk_drv:0x00,q_drv:0x00,d_drv:0x00,cs0_drv:0x00,hd_drv:0x00,wp_drv:0x00
mode:DIO, clock div:1
load:0x3fff0030,len:1344
load:0x40078000,len:13964
load:0x40080400,len:3600
entry 0x400805f0
The device with name "ESP32-BT-Slave" is started.
Now you can pair it with Bluetooth!
Bienvenid@ ingenier@

☒ Autoscroll ☐ Mostrar marca temporal
```



# FREERTOS Y ESP32 BLUETOOTH 15%

Al anterior programa agregar lo siguiente:

Al recibir un mensaje de ON desde la comunicación Bluetooth encender un led.

Al recibir un mensaje de OFF desde la comunicación Bluetooth apagar un led.

Estudiar la implementación de las otras versiones de Bluetooth (BLE) les recomiendo:

<https://randomnerdtutorials.com/esp32-bluetooth-guide/>

# Microcontroladores II

PROYECTOS DE AULA  
(OBLIGATORIA LA PARTICIPACIÓN)

ESPACIO PARA PREPARACIÓN

PRESENTACIÓN DE PROYECTOS  
24 DE MAYO

Escuela de  
Ingenierías

IUSH  
Institución Universitaria  
SALAZAR Y HERRERA

SER PERSONA DESDE EL SABER Y EL HACER