

PRÁCTICA 3B: COMUNICACIÓN BLUETOOTH CON EL MOVIL

APARTADO 1

CÓDIGO

```
#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");
  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);
}
```

FUNCIONAMIENTO

El código comienza incluyendo la *library* BluetoothSerial :

```
#include "BluetoothSerial.h"
```

Después comprueba si el Bluetooth está habilitado con:

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

A continuación crea una instancia de `BluetoothSerial` llamada `SerialBT` :

```
BluetoothSerial SerialBT;
```

Después empieza el `void setup()` , este en la primera línea inicializa una comunicación en serie a una velocidad de 115200 bauds:

```
Serial.begin(115200);
```

En la siguiente línea inicializa el dispositivo serie Bluetooth y pasa como argumento el nombre del dispositivo, en este caso *ESP32test* (este nombre es el predeterminado):

```
SerialBT.begin("ESP32test");
```

Y para acabar el `setup()` escribes por pantalla que ya has inicializado el dispositivo:

```
Serial.println("The device started, now you can pair it with bluetooth!");
}
```

Para finalizar creamos el `void loop()` , en las primeras tres líneas creamos un *if*, donde verificamos si se están recibiendo bytes en el puerto serie, si es así que se envíe esa información a través de Bluetooth al dispositivo conectado.

```
if (Serial.available()) {
    SerialBT.write(Serial.read());
}
```

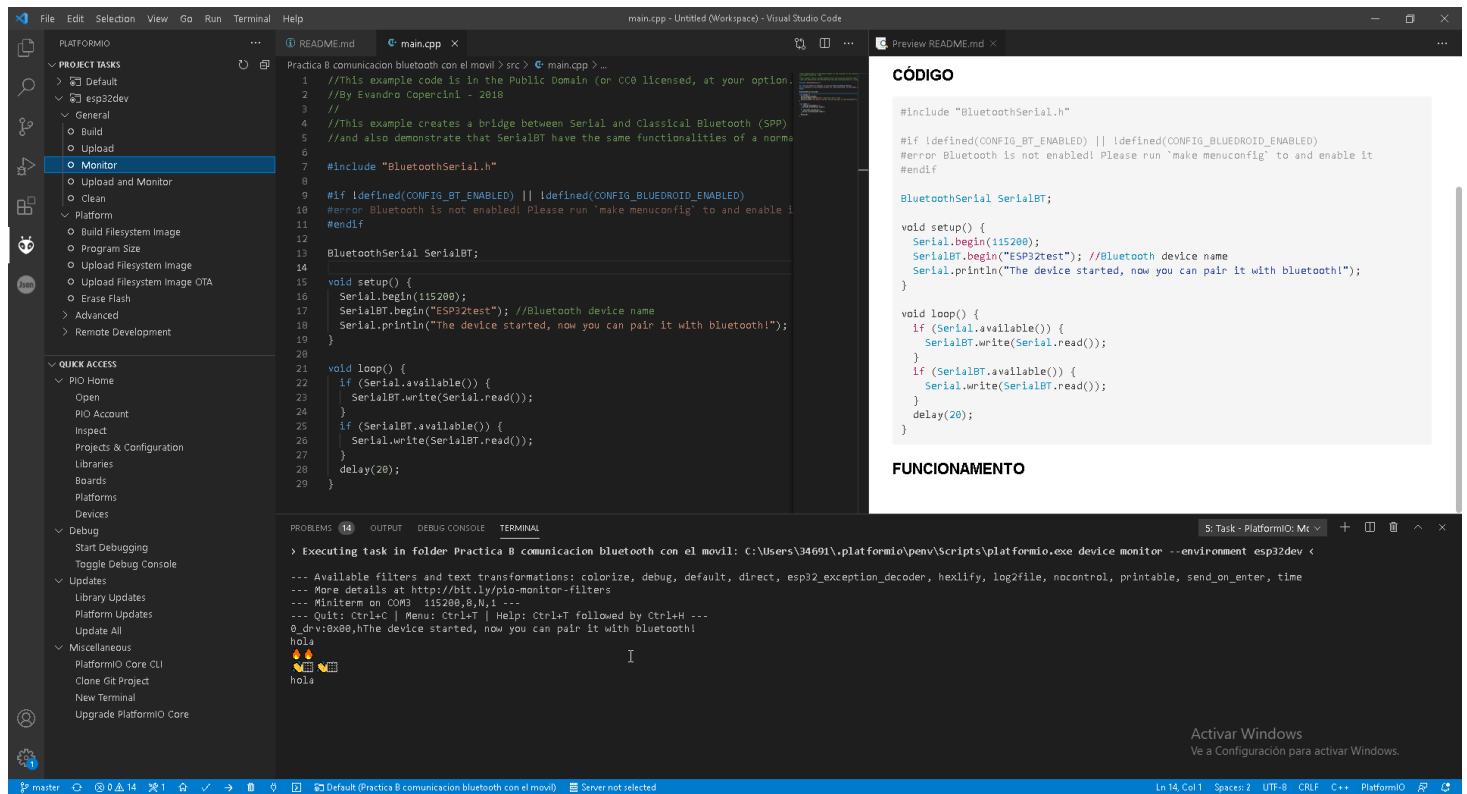
Y en el segundo *if* comprobamos si hay bytes disponibles para leer en el puerto serie Bluetooth, si es así escribiremos esos bytes en el monitor.

```
if (SerialBT.available()) {
    Serial.write(SerialBT.read());
}
```

Y para acabar el *loop* creamos un *delay*:

```
delay(20);
```

SALIDA POR EL TERMINAL



En este caso en el terminal se ven cuatro mensajes, que son los que he enviado desde otro dispositivo.