

Sistemas Embebidos



Universitat d'Alacant
Universidad de Alicante



Practica 4

Francisco Joaquín Murcia Gómez

30 de abril de 2022

Índice

1. Funcionamiento del BLE	3
1.1. Prueba de la función BLE	3
1.1.1. Modo emisor	3
1.1.2. Modo monitor	4
1.2. Calculo de distancia mediante intensidad de señal	6
2. Funcionamiento combinado Wifi/BLE	8
2.1. Ping	8
2.2. Funcionamiento de Wifi y BLE simultaneo	11

1. Funcionamiento del BLE

1.1. Prueba de la función BLE

1.1.1. Modo emisor

Para este modo se ha usado el código de ejemplo para observar el nivel de batería del teléfono, así que se ha cargado el ejemplo “ArduinoBLE/Peripheral/BatteryMonitor“. Al ejecutarlo nos aparece en los ajustes de bluetooth la señal del arduino para conectarse:

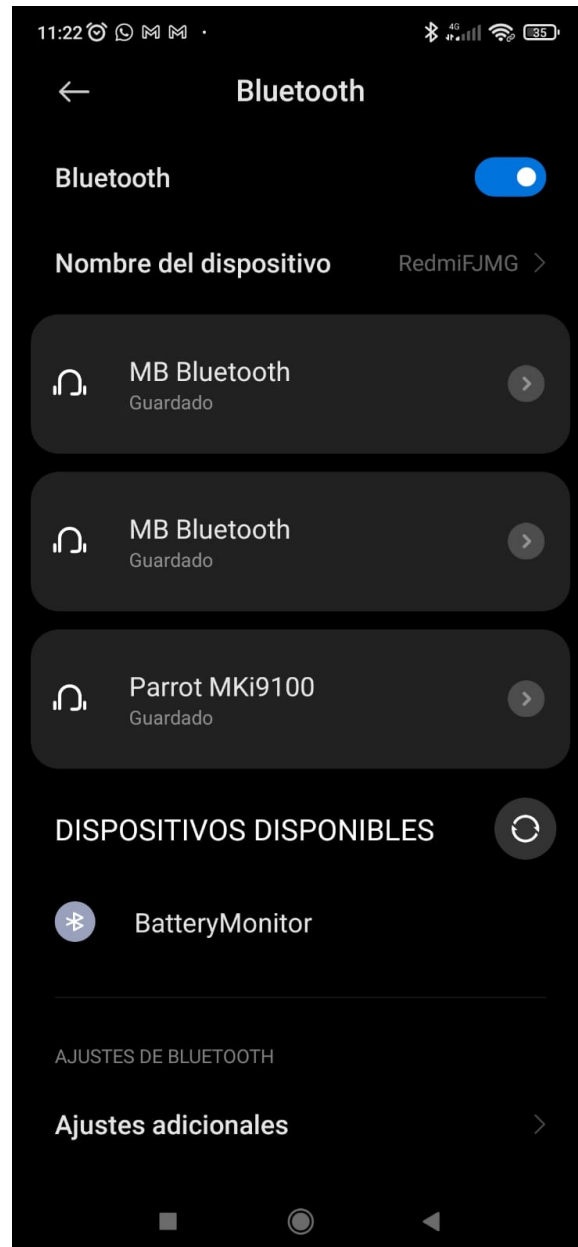


Figura 1: Dispositivos bluetooth disponibles

Cuando nos vinculamos el arduino muestra el nivel de batería del dispositivo

```
Bluetooth® device active, waiting for connections...
Connected to central: 49:6d:05:30:c3:91
Battery Level % is now: 47
Battery Level % is now: 45
Battery Level % is now: 44
Battery Level % is now: 43
Battery Level % is now: 44
Battery Level % is now: 43
Battery Level % is now: 44
Battery Level % is now: 43
Battery Level % is now: 44
Battery Level % is now: 45
Battery Level % is now: 44
Battery Level % is now: 43
Battery Level % is now: 44
Battery Level % is now: 43
Battery Level % is now: 44
Battery Level % is now: 42
Battery Level % is now: 43
Battery Level % is now: 44
Battery Level % is now: 43
Battery Level % is now: 44
Battery Level % is now: 43
Battery Level % is now: 44
Battery Level % is now: 45
Battery Level % is now: 44
Battery Level % is now: 45
Battery Level % is now: 44
Battery Level % is now: 45
Battery Level % is now: 43
Battery Level % is now: 44
Battery Level % is now: 43
Battery Level % is now: 44
Disconnected from central: 49:6d:05:30:c3:91
```

Figura 2: Informacion del estado de la bateria

1.1.2. Modo monitor

Esta vez se ha importado el ejemplo "ArduinoBLE/Central/Scan" donde se imprime la intensidad de señal el UUID y la MAC

```
Discovered a peripheral
-----
Address: 6d:3b:be:52:fb:b2
Service UUIDs: fd6f
RSSI: -67
```

Figura 3: RSSI, Direccion MAC y UUID de mi dispositivo movil

Se le a añadido un servidor de tiempo para que cada 5 segundos reales haga el escaneo para ello se han añadido las líneas 6,7,27 donde esta ultima es la barrera de paso que se abre solo cuando pasan 5 segundos.

```
1  #include <ArduinoBLE.h>
2  #include <RTCZero.h>
3  RTCZero rtc;
4
5  void setup() {
6    Serial.begin(9600);
7    rtc.begin();
8
9    // begin initialization
10   if (!BLE.begin()) {
11     Serial.println("starting Bluetooth@ Low Energy module failed!");
12
13     while (1);
14   }
15
16   Serial.println("Bluetooth@ Low Energy Central scan");
17   // start scanning for peripheral
18   BLE.scan();
19 }
20
21 void loop() {
22   // check if a peripheral has been discovered
23   BLEDevice peripheral = BLE.available();
24
25   //creamos una barrera que mientras no sea un numero divisible entre 5 (0,5,10...)
26   //no se abre, de esta forma solo el programa contunudara cada 5 segundos
27   while(rtc.getSeconds()%5 != 0);
28
29   if (peripheral) {
30     // discovered a peripheral
31     Serial.println("Discovered a peripheral");
32     Serial.println("-----");
33
34     // print address
35     Serial.print("Address: ");
36     Serial.println(peripheral.address());
37
38     // print the local name, if present
```

```

39     if (peripheral.hasLocalName()) {
40         Serial.print("Local Name: ");
41         Serial.println(peripheral.localName());
42     }
43
44     // print the advertised service UUIDs, if present
45     if (peripheral.hasAdvertisedServiceUuid()) {
46         Serial.print("Service UUIDs: ");
47         for (int i = 0; i < peripheral.advertisedServiceUuidCount(); i++) {
48             Serial.print(peripheral.advertisedServiceUuid(i));
49             Serial.print(" ");
50         }
51         Serial.println();
52     }
53
54     // print the RSSI
55     Serial.print("RSSI: ");
56     Serial.println(peripheral.rssi());
57
58     Serial.println();
59
60 }
61 }

```

1.2. Calculo de distancia mediante intensidad de señal

Para este paso se ha estado tomando la intensidad de señal (RSSI) el beacon cada 10 centímetros.

Otros compañeros hicieron las medidas o en el aula de practicas o en sus casas y la señal tenia mucho error y era imprecisa, debido a esto he realizado las medidas en las afueras de mi pueblo en un huerto de mas de 50m de largo para así evitar cualquier interferencia.

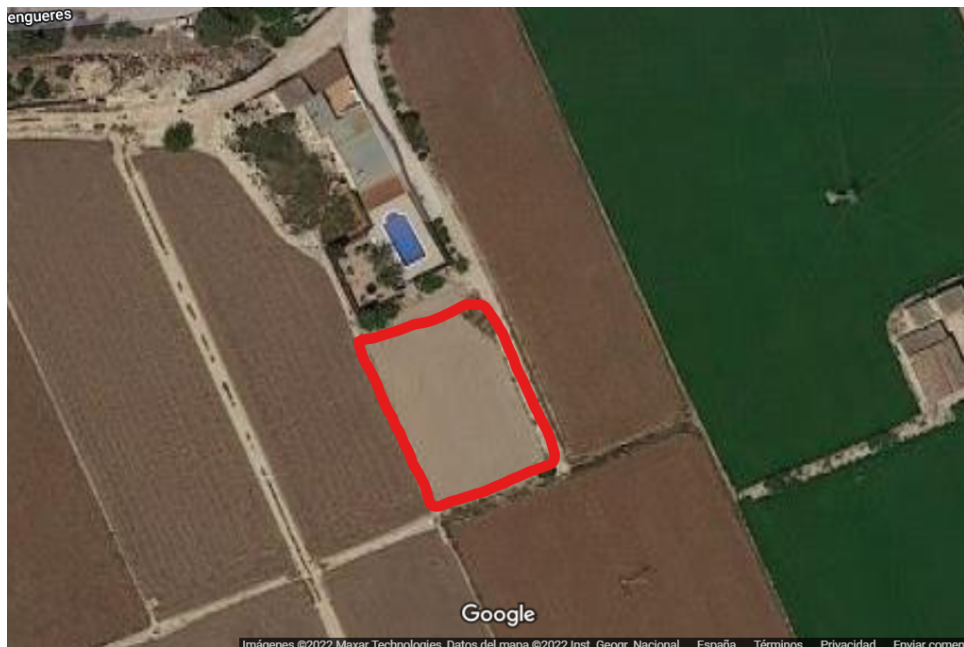


Figura 4: Ubicación elegida (rojo)

El código empleado es el mismo que el ejemplo de Scan que nos aporta la propia librería pero se ha cambiado la función Scan() por ScanForUuid() para solo escanear mi dispositivo (concretamente el "fd6f") para así seguir evitando interferencias.

En cada distancia se han tomado 10 medidas y se han eliminado las intensidades que sobrepasaban la desviación típica, también se ha anotado el error de la medida que en mi caso fue de un $\pm 5\%$

Distancia cm	RSSI	Error
0	-20	1
10	-30	1,5
20	-35	1,75
30	-43	2,15
40	-48	2,4
50	-56	2,8
60	-60	3
70	-70	3,5
80	-74	3,7
90	-76	3,8
100	-77	3,85
110	-80	4
120	-81	4,05
130	-83	4,15
140	-84	4,2
150	-85	4,25
160	-85	4,25
170	-87	4,35
180	-89	4,45
190	-88	4,4
200	-88	4,4
210	-88	4,4
220	-88	4,4
230	-93	4,65
240	-95	4,75
250	-97	4,85
260	-98	4,9
270	-98	4,9
280	-99	4,95
290	-99	4,95
300	-99	4,95

Figura 5: Datos

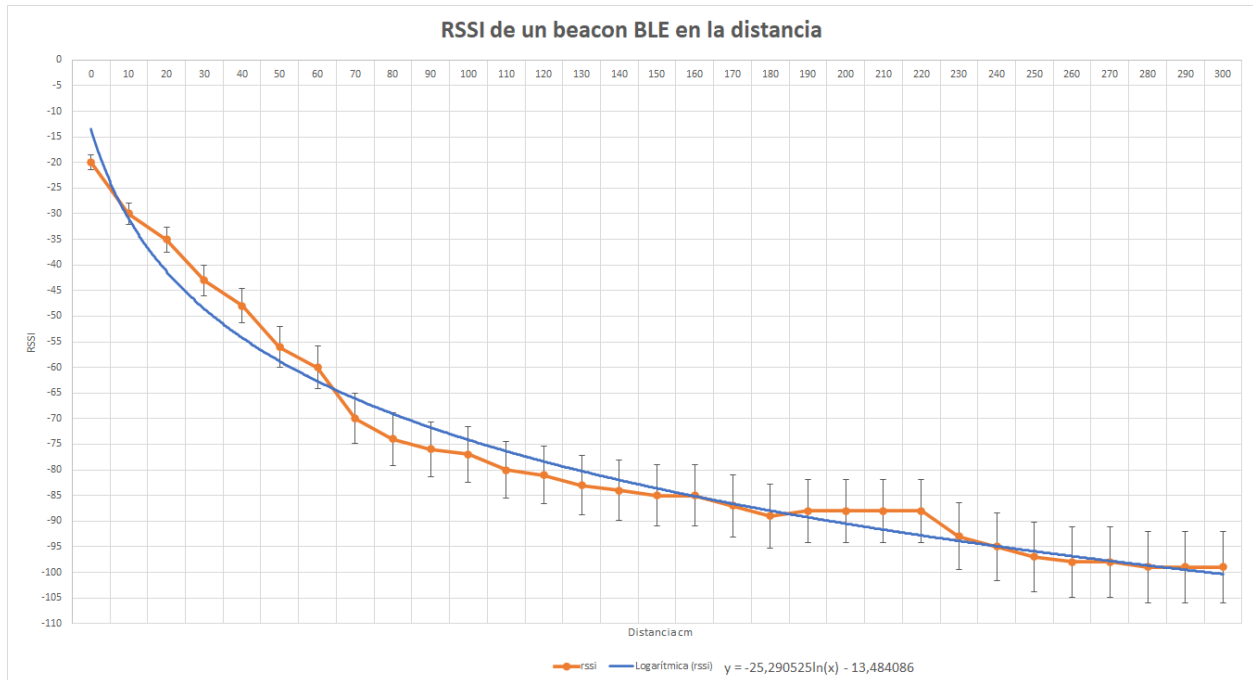


Figura 6: Comparación de la intensidad de señal con la distancia

Se han interpolado los puntos y se ha obtenido la siguiente ecuación logarítmica:

$$RSSI = -25,290525 \ln(DISTANCIA) - 13,484086$$

Podemos observar que al no haber interferencias podemos llegar hasta 3 metros, a partir de allí la señal se estanca. también las intensidades disminuyen mas orgánicamente con la distancia apreciándose mejor la curva logarítmica que tiene que dar al contrario que si se hiciera la prueba en la universidad como otros compañeros donde la cantidad de interferencias es tal que la intensidad disminuye y fluctúa mucho.

2. Funcionamiento combinado Wifi/BLE

2.1. Ping

Para este ejemplo hemos hecho ping a Google y se ha empleado el ejemplo de ping que nos aporta la librería WiFinINA

```

1  #include <SPI.h>
2  #include <WiFinINA.h>
3
4  #include "arduino_secrets.h"
5  //please enter your sensitive data in the Secret tab/arduino_secrets.h
6  char ssid[] = SECRET_SSID;        // your network SSID (name)
7  char pass[] = SECRET_PASS;        // your network password (use for WPA, or use as key for WEP)
8  int status = WL_IDLE_STATUS;      // the WiFi radio's status
9
10 // Specify IP address or hostname
11 String hostName = "www.google.com";
12 int pingResult;
13

```



```

14 void setup() {
15     // Setup leds
16     pinMode(LED_BUILTIN, OUTPUT);
17
18     // Initialize serial and wait for port to open:
19     Serial.begin(9600);
20
21
22     digitalWrite(LED_BUILTIN, HIGH);
23
24
25     // check for the WiFi module:
26     if (WiFi.status() == WL_NO_MODULE) {
27         Serial.println("Communication with WiFi module failed!");
28         // don't continue
29         while (true);
30     }
31
32     String fv = WiFi.firmwareVersion();
33     if (fv < WIFI_FIRMWARE_LATEST_VERSION) {
34         Serial.println("Please upgrade the firmware");
35     }
36
37     // attempt to connect to WiFi network:
38     while ( status != WL_CONNECTED) {
39         Serial.print("Attempting to connect to WPA SSID: ");
40         Serial.println(ssid);
41         // Connect to WPA/WPA2 network:
42         status = WiFi.begin(ssid, pass);
43
44         digitalWrite(LED_BUILTIN, LOW);
45         parpadeo(100, 700);
46     }
47
48     // you're connected now, so print out the data:
49     Serial.println("You're connected to the network");
50     digitalWrite(LED_BUILTIN, LOW);
51     printCurrentNet();
52     printWiFiData();
53 }
54
55 void loop() {
56     Serial.print("Pinging ");
57     Serial.print(hostName);
58     Serial.print(": ");
59
60     pingResult = WiFi.ping(hostName);
61
62     if (pingResult >= 0) {
63         Serial.print("SUCCESS! RTT = ");
64         Serial.print(pingResult);
65         Serial.println(" ms");
66     } else {
67         Serial.print("FAILED! Error code: ");
68         Serial.println(pingResult);
69     }
70

```

```

71     delay(5000);
72 }
73
74 void printWiFiData() {
75     // print your board's IP address:
76     IPAddress ip = WiFi.localIP();
77     Serial.print("IP address : ");
78     Serial.println(ip);
79
80     Serial.print("Subnet mask: ");
81     Serial.println((IPAddress)WiFi.subnetMask());
82
83     Serial.print("Gateway IP : ");
84     Serial.println((IPAddress)WiFi.gatewayIP());
85
86     // print your MAC address:
87     byte mac[6];
88     WiFi.macAddress(mac);
89     Serial.print("MAC address: ");
90     printMacAddress(mac);
91 }
92
93 void printCurrentNet() {
94     // print the SSID of the network you're attached to:
95     Serial.print("SSID: ");
96     Serial.println(WiFi.SSID());
97
98     // print the MAC address of the router you're attached to:
99     byte bssid[6];
100     WiFi.BSSID(bssid);
101     Serial.print("BSSID: ");
102     printMacAddress(bssid);
103     // print the received signal strength:
104     long rssi = WiFi.RSSI();
105     Serial.print("signal strength (RSSI): ");
106     Serial.println(rssi);
107
108     // print the encryption type:
109     byte encryption = WiFi.encryptionType();
110     Serial.print("Encryption Type: ");
111     Serial.println(encryption, HEX);
112     Serial.println();
113 }
114
115 void printMacAddress(byte mac[]) {
116     for (int i = 5; i >= 0; i--) {
117         if (mac[i] < 16) {
118             Serial.print("0");
119         }
120         Serial.print(mac[i], HEX);
121         if (i > 0) {
122             Serial.print(":");
123         }
124     }
125     Serial.println();
126 }
127

```

```

128 void parpadeo(int duracion, int frecuencia) {
129     for (int i = 1; i <= 5; i++) {
130         for (int j = 1; j <= i; j++ ) {
131             digitalWrite(LED_BUILTIN, HIGH);
132             delay(duracion);
133             digitalWrite(LED_BUILTIN, LOW);
134             delay(duracion);
135         }
136         delay(frecuencia);
137     }
138 }

```

```

You're connected to the network
SSID: Redmi
BSSID: C2:25:14:1A:E0:0D
signal strength (RSSI): -20
Encryption Type: 4

IP address : 192.168.47.144
Subnet mask: 255.255.255.0
Gateway IP : 192.168.47.126
MAC address: C8:C9:A3:E6:7E:F8
Pinging www.google.com: SUCCESS! RTT = 210 ms
Pinging www.google.com: SUCCESS! RTT = 110 ms
Pinging www.google.com: SUCCESS! RTT = 110 ms
Pinging www.google.com: SUCCESS! RTT = 70 ms
Pinging www.google.com: SUCCESS! RTT = 70 ms

```

Figura 7: Ping a Google

2.2. Funcionamiento de Wifi y BLE simultaneo

Este funcionamiento a priori seria imposible ya que Arduino nano solo dispone de una antena

```

1  #include <ArduinoBLE.h>
2  #include <WiFiNINA.h>
3
4  // SSID and PASSWORD
5  #define SSID      "Redmi"
6  #define PASSWORD  "987654321"
7  int status = WL_IDLE_STATUS;
8
9  void setup() {

```

```

10
11
12 // setup serial output
13 Serial.begin(9600);
14
15 // si se puede conectar conectamos a wifi
16 while ( status != WL_CONNECTED) {
17     Serial.print("conectando...");
18     Serial.println(SSID);
19     // conectando:
20     status = WiFi.begin(SSID, PASSWORD);
21     delay(5000);
22 }
23
24 // Once connected, notify via serial monitor, and try BLE scanning.
25     Serial.print("conectado a:");
26     Serial.println(SSID);
27 Serial.println("Encendiendo BLE");
28
29 if (!BLE.begin()) {
30     Serial.println("Fallo al encender BLE");
31
32     while (1);
33 }
34
35 Serial.println("Starting BLE Scanning...");
36 BLE.scan();
37 BLEDevice peripheral = BLE.available();
38
39 if (peripheral) {
40     // discovered a peripheral
41     Serial.println("Discovered a peripheral");
42     Serial.println("-----");
43
44     // print address
45     Serial.print("Address: ");
46     Serial.println(peripheral.address());
47
48     // print the local name, if present
49     if (peripheral.hasLocalName()) {
50         Serial.print("Local Name: ");
51         Serial.println(peripheral.localName());
52     }
53
54     // print the advertised service UUIDs, if present
55     if (peripheral.hasAdvertisedServiceUuid()) {
56         Serial.print("Service UUIDs: ");
57         for (int i = 0; i < peripheral.advertisedServiceUuidCount(); i++) {
58             Serial.print(peripheral.advertisedServiceUuid(i));
59             Serial.print(" ");
60         }
61         Serial.println();
62     }
63
64     // print the RSSI
65     Serial.print("RSSI: ");
66     Serial.println(peripheral.rssi());

```

```
67 }  
68 }
```

Cuando ejecutamos este código se bloquea en la línea 36 ya que la antena está ocupada por la función WIFI

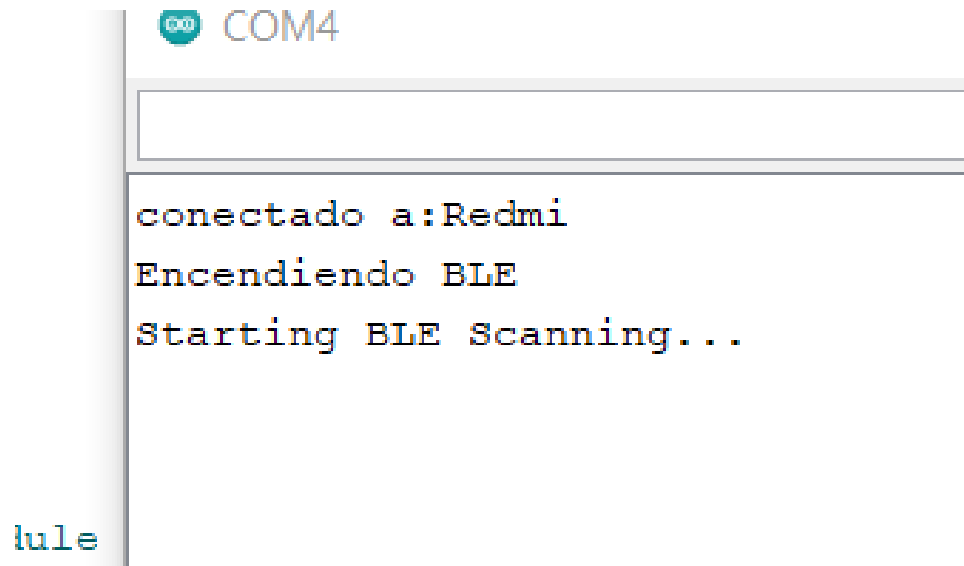


Figura 8: BLE WIFI combinado

Esto se podría solucionar apagando la antena y enchufándola cada vez que se cambie de funcionalidad.