

Capítulo 5

Bluetooth: Activar y Descubrir Redes

Dr. Raúl Marín Prades

Julio 2018

Tabla de Contenidos

Resumen

PASO 1: Estudiar clase «BluetoothAdapter»

PASO 2: Función «getDefaultAdapter()»

PASO 3: Importar «BluetoothAdapter»

PASO 4: Modificar «AndroidManifest.xml»

PASO 5: Comprobar si el Bluetooth está habilitado

PASO 6: Activar/Desactivar Bluetooth

PASO 7: Intent Activar Bluetooth

PASO 8: Función «onActivityResult»

PASO 9: Variable «bluetoothActive»

PASO 10: Lista dispositivos Bluetooth

PASO 11: Método «startDiscovery()»

PASO 12: Objeto «BroadcastReceiver»

Conclusión

Resumen

En este capítulo estudiaremos el primer paso para utilizar la red Bluetooth, el cual consiste en comprobar que el dispositivo Android dispone del adaptador físico necesario, activarlo en el caso en que no lo esté, solicitando permiso al usuario, y ponerlo en marcha en modo descubrimiento, con el objetivo de detectar redes bluetooth disponibles. Una vez detectada una red, es posible acceder a valores interesantes como el RSSI (Radio Signal Strength Indicator), el cual permite conocer la distancia estimada a un emisor bluetooth a partir de la intensidad de la señal. Este tipo de herramientas abre un abanico importante de aplicaciones de localización de sensores y dispositivo móviles.

En los siguientes capítulos trabajaremos también la comunicación a través del Bluetooth con un dispositivo Arduino.

PASO 1: Estudiar clase «BluetoothAdapter»

El punto de entrada para el estudio de las aplicaciones Android con Bluetooth es la clase «BluetoothAdapter». Esta clase proporciona acceso a las propiedades físicas y lógicas del adaptador, así como permite establecer comunicación a través de un conector (i.e. Socket).

Es bueno empezar accediendo a la siguiente página web con el objetivo de familiarizarse con el API de la clase «BluetoothAdapter»:

<http://developer.android.com/reference/android/bluetooth/BluetoothAdapter.html>

- Es importante destacar que a partir de la version 6 de Android es necesario tener en cuenta algunas instrucciones adicionales. También, el presente capítulo está centrado en el API «BluetoothAdapter», usada principalmente por los dispositivos en red basados en Arduino, sin entrar todavía en la versión 4.0 BLE.

PASO 2: Función «getDefaultAdapter()»

Siguiendo el código ejemplo del capítulo anterior, vamos a modificar la acción asociada al botón «Connect» (i.e. Conectar), con el objetivo de acceder el adaptador Bluetooth por defecto, añadiendo el siguiente código:

```
BluetoothAdapter bluetooth = BluetoothAdapter.getDefaultAdapter();
```

- ▶ Si la variable «bluetooth» devuelve el valor «null» significa que el dispositivo Android no tiene instalado un adaptador Bluetooth.

PASO 3: Importar «BluetoothAdapter»

Importar la clase Bluetooth Adapter, añadiendo el siguiente código Java en la clase «MainActivity.java»:

```
import android.bluetooth.BluetoothAdapter;
```

PASO 4: Modificar «AndroidManifest.xml»

Modificar el archivo «AndroidManifest.xml» con el objetivo de solicitar permiso al sistema operativo Android para utilizar el adaptador bluetooth. Añadir las siguientes líneas:

```
<uses-permission android:name="android.permission.BLUETOOTH_ADMIN" />  
<uses-permission android:name="android.permission.BLUETOOTH" />
```

- En dispositivos Android 6 y superiores es necesario añadir también los siguientes permisos al fichero «AndroidManifest.xml». También, es preciso asignar el valor 23 (o superior) al atributo «minSdkVersion».

```
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />  
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
```

PASO 4: Modificar «AndroidManifest.xml» (continuación)

Un ejemplo de archivo «AndroidManifest.xml» es el siguiente:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.raulmarin.myapplicationbluetooth">

    <uses-permission android:name="android.permission.BLUETOOTH_ADMIN" />
    <uses-permission android:name="android.permission.BLUETOOTH" />
    <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity android:name=".DiscoveredBluetoothDevicesActivity"></activity>
    </application>

</manifest>
```

PASO 5: Comprobar si el Bluetooth está habilitado

Continuando con el código que estamos preparando, asociado al evento del botón «Connect» (i.e. Conectar), una vez sabemos que el dispositivo Android dispone de adaptador bluetooth, usamos la función «isEnabled()» para comprobar si el usuario lo tiene activado desde la sección de «Preferencias» del dispositivo. En caso afirmativo, consultamos, a modo de ejemplo, la dirección MAC del adaptador bluetooth y el nombre, mostrándola en pantalla con el objeto «Toast».

- El objeto Toast permite visualizar un texto en la pantalla que desaparece automáticamente después de un tiempo configurable. Para mas información se puede consultar el API:
<https://developer.android.com/guide/topics/ui/notifiers/toasts.html>

```
if (bluetooth.isEnabled()){  
    String address = bluetooth.getAddress();  
    String name = bluetooth.getName();  
    //Mostramos la datos en pantalla (The information is shown in the screen)  
    Toast.makeText(getApplicationContext(), "Bluetooth ENABLED:"+name+": "+address,  
        Toast.LENGTH_SHORT).show();  
}else{  
    Toast.makeText(getApplicationContext(), "Bluetooth NOT enabled",  
        Toast.LENGTH_SHORT).show();  
}
```

PASO 6: Activar/Desactivar Bluetooth

Acceder a la ventana de preferencias del dispositivo móvil (ver Figura 1), y desactivar el Bluetooth. Ejecutar de nuevo la aplicación para ver si el estado es reconocido por el código.

Activar de nuevo el Bluetooth en Preferencias, y ejecutar de nuevo la aplicación, examinando el resultado.

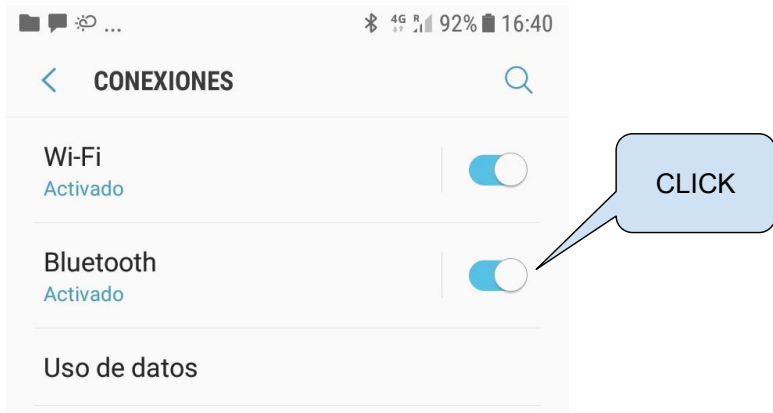


Figura 1: Activar/Desactivar el Bluetooth para comprobar el programa

PASO 7: Intent Activar Bluetooth

Un Intent es un objeto que permite invocar acciones del sistema operativo, como por ejemplo:

- ▶ Activar Bluetooth
- ▶ Activar Wifi
- ▶ Mostrar una nueva ventana

Añade el siguiente código al ejemplo del paso anterior, de forma que, cuando el Bluetooth esté desactivado, aparezca una ventana del sistema Android (i.e. Intent), solicitando confirmación al usuario para activar el Bluetooth.

```
startActivityForResult(new Intent(BluetoothAdapter.ACTION_REQUEST_ENABLE),1);
```

- ▶ El parámetro con valor «1» sirve para poder comprobar, dependiendo del Intent, que hemos lanzado, el botón seleccionado por el usuario.

PASO 7: Intent Activar Bluetooth (continuación)

Como resultado, si el Bluetooth está desactivado, aparecerá la siguiente ventana (ver Figura 2)).

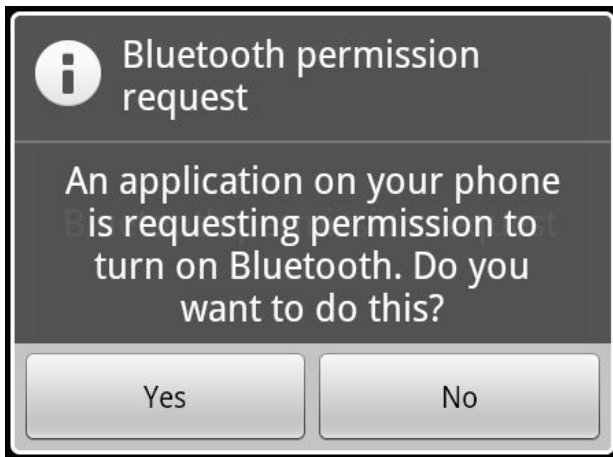


Figura 2: Resultado de ejecutar el Intent `BluetoothAdapter.ACTION_REQUEST_ENABLE`

PASO 8: Función «onActivityResult»

Es necesario añadir la función «onActivityResult» a la clase «MainActivity.java», con el objetivo de conocer la opción elegida por el usuario en la ventana del Intent. Añadir el siguiente código a la clase «MainActivity.java»:

```
protected void onActivityResult(int requestCode, int resultCode, Intent data) {  
  
    if (requestCode == 1) //Bluetooth permission request code  
        if (resultCode == RESULT_OK){  
            Toast.makeText(getApplicationContext(),"User Enabled Bluetooth",  
                Toast.LENGTH_SHORT).show();  
        }else{  
            Toast.makeText(getApplicationContext(),"User Did not enable Bluetooth",  
                Toast.LENGTH_SHORT).show();  
        }  
}
```

El parámetro de entrada «requestCode» informa del identificador dado al Intent en el momento de su ejecución. En nuestro caso es el mismo especificado en el paso 7 (i.e. "1").

PASO 9: Variable «bluetoothActive»

Añadir una variable global a la clase «MainActivity.java», llamada «bluetoothActive». Esta variable guardará si el adaptador bluetooth está disponible, con el objetivo de proceder con la conexión.

```
public static Boolean bluetoothActive = false;
```

Modificar el código para que el valor de esta variable esté actualizado, tanto en el constructor como durante la ejecución del método «onActivityResult». Puedes visualizar el valor de la variable con la clase «Toast».

PASO 10: Lista dispositivos Bluetooth

Declarar una variable global de tipo «ArrayList», que almacene la lista de dispositivos Bluetooth descubiertos:

```
private ArrayList<BluetoothDevice> deviceList = new ArrayList<BluetoothDevice>();
```

PASO 11: Método «startDiscovery()»

Crear un método de la clase «MainActivity.java» que incluya todo el código Java necesario para descubrir los dispositivos bluetooth remotos alcanzables:

```
private void startDiscovery(){
    if (bluetoothActive){
        //Borramos la lista de dispositivos anterior
        deviceList.clear();
        //Activamos un Intent Android que avise cuando se encuentre un dispositivo
        //NOTA: <<discoveryResult>> es una clase <<callback>> que describiremos en
            el siguiente paso
        registerReceiver(discoveryResult, new
            IntentFilter(BluetoothDevice.ACTION_FOUND));
        //Ponemos el adaptador bluetooth en modo <<Discovery>>
        bluetooth.startDiscovery();
    }
}
```

- Teniendo en cuenta que la acción «bluetooth.startDiscovery();» es asíncrona, necesitamos un objeto de tipo «BroadcastReceiver» (i.e. variable discoveryResult), que escuche los eventos «bluetooth discovery» y los gestione de manera adecuada. Para ello, se crea un objeto de tipo «IntentFilter», asociado al evento «BluetoothDevice.ACTION_FOUND».

PASO 12: Objeto «BroadcastReceiver»

Crear el objeto «BroadcastReceiver» para mostrar la información de los dispositivos bluetooth que se descubran:

```
BroadcastReceiver discoveryResult = new BroadcastReceiver() {
    @Override
    public void onReceive(Context context, Intent intent) {
        //Guardamos el nombre del dispositivo descubierto
        String remoteDeviceName =
            intent.getStringExtra(BluetoothDevice.EXTRA_NAME);
        //Guardamos el objeto Java del dispositivo descubierto, para poder
            conectar.
        BluetoothDevice remoteDevice =
            intent.getParcelableExtra(BluetoothDevice.EXTRA_DEVICE);
        //Leemos la intensidad de la radio con respecto a este dispositivo
            bluetooth
        int rssi =
            intent.getShortExtra(BluetoothDevice.EXTRA_RSSI, Short.MIN_VALUE);
        //Guardamos el dispositivo encontrado en la lista
        deviceList.add(remoteDevice);
        //Mostramos el evento en el Log.
        Log.d("MyFirstApp", "Discovered "+ remoteDeviceName);
        Log.d("MyFirstApp", "RSSI "+ rssi + "dBm");
    }
};
```

PASO 12: Objeto «BroadcastReceiver» (continuación)

- ▶ NOTA: En los sistemas Android 6 y superiores, los permisos «ACCESS_COARSE_LOCATION» y «ACCESS_FINE_LOCATION» se deben solicitar también a través del programa Java, por medio de la ejecución del siguiente código (i.e. método «checkBTPermissions()»), justo antes de invocar el método «startDiscovery()».

```
public void checkBTPermissions(){  
  
    if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {  
        switch (ContextCompat.checkSelfPermission(getBaseContext(),  
            Manifest.permission.ACCESS_COARSE_LOCATION)) {  
            case PackageManager.PERMISSION_DENIED:  
                if (ContextCompat.checkSelfPermission(getBaseContext(),  
                    Manifest.permission.ACCESS_COARSE_LOCATION) !=  
                    PackageManager.PERMISSION_GRANTED) {  
                    this.requestPermissions(new  
                        String[]{Manifest.permission.ACCESS_FINE_LOCATION,  
                            Manifest.permission.ACCESS_COARSE_LOCATION}, 1001);  
                }  
                break;  
            case PackageManager.PERMISSION_GRANTED:  
                break;  
        }  
    }  
}
```

Conclusión

En este capítulo hemos estudiado el primer paso para trabajar con el sistema Bluetooth, el cual nos ha permitido descubrir los dispositivos remotos disponibles. Hemos visto el concepto de Intent, para activar acciones sobre el adaptador Bluetooth, gestionadas a través del sistema operativo. También, hemos estudiado los permisos necesarios, a ser especificados de forma estática en el fichero «AndroidManifest.xml» y, a partir de la versión 6 de Android, también en el propio programa.