

Programación Bluetooth en Java

1. OBJETIVO

Programación en Java de aplicaciones sobre Bluetooth utilizando la API BlueCove. Se utilizarán las aplicaciones desarrolladas para implementar diversas aplicaciones cuya comunicación entre los clientes y el servidor se base en Bluetooth.

En aquellos equipos en los que no haya un dispositivo Bluetooth disponible, se hará uso del emulador de la pila de protocolos proporcionado también por la distribución de Bluecove (bluecove-emu.jar).

<http://www.bluecove.org>

2. DISPOSITIVO DE COMUNICACIÓN BLUETOOTH

1. Utiliza la documentación de la API y los métodos de la clase `LocalDevice` para obtener información acerca de la configuración de tu dispositivo Bluetooth local ¿Qué información puedes obtener?

NOTA : utilizad, entre otros, el método estático `getProperty()` de la clase `LocalDevice` (las propiedades están definidas en la interfaz `BlueCoveLocalDeviceProperties`)

<http://bluecove.org/bluecove/apidocs/index.html>

3. DESCUBRIMIENTO DE DISPOSITIVOS REMOTOS

2. Implementación de una aplicación de escritorio que permita descubrir todos los dispositivos bluetooth visibles cercanos.
3. Implementación de una aplicación que permita descubrir un dispositivo concreto (por dirección Bluetooth y/o *friendly name*).

4. DESCUBRIMIENTO DE SERVICIOS

4. Implementación de una aplicación que permita descubrir todos los servicios de un dispositivo concreto.
5. Implementación de una aplicación que permita descubrir todos los servicios de todos los dispositivos cercanos.
6. Implementación de una aplicación que permita descubrir un servicio de clase público y de clase puerto serie (serial port) en todos los dispositivos cercanos.
7. Implementación de una aplicación que permita descubrir un servicio con un nombre concreto en un dispositivo concreto.

En todas las aplicaciones desarrolladas se deberán gestionar errores, y se deberá solicitar al usuario la información pertinente y mostrar los mensajes oportunos al usuario.

Haz uso de los siguientes UUIDs en la búsqueda de servicios:

```
static int SERVICE_NAME_ATTRID = 0x0100;
...
// service search parameters
UUID uuids[] = new UUID[1];
uuids[0] = new UUID(0x1002); //PublicBrowseGroup
uuids[1] = new UUID(0x1101); //SerialPort

int attridset[] = new int[1];
attridset[0] = SERVICE_NAME_ATTRID;
```

Puedes consultar la especificación de los UUIDs en Bluetooth:

<https://www.bluetooth.org/en-us/specification/assigned-numbers>

Para servicios:

<https://www.bluetooth.org/en-us/specification/assigned-numbers/service-discovery>

5. COMUNICACIÓN A TRAVÉS DE RFCOMM DE BLUETOOTH

1. Implementa una aplicación cliente-servidor que permita el envío de mensajes de texto entre ellos (como un chat textual). Desde el cliente, tras encontrar al dispositivo servidor, encuentra el servicio de chat (de clase puerto-serie) a través del nombre que el servidor le ha asignado (en el ejemplo a continuación, "chat"). Finalmente, usa su URL de conexión en el cliente para poder enviar y recibir datos.

Puedes reutilizar en parte el código desarrollado en los apartados anteriores (para la búsqueda del servidor y del servicio).

Podemos asumir que el servidor publica el servicio con la siguiente URL (y atributos):

```
String url = "btspp://localhost:" + new UUID(0x1101).toString() +
    ";name=chat";
```

6. REFERENCIAS

- [1] Bluecove: Java library for Bluetooth (JSR-82 implementation) <http://bluecove.org/>