

ANDROID ALKALMAZÁSFEJLESZTÉS

Kommunikáció

GSM hívás

SMS küldés, fogadás

Bluetooth

NFC

USB



Sicz-Mesziár János

sicz-mesziar.janos@nik.uni-obuda.hu

Mezei József

mezei.jozsef@nik.uni-obuda.hu

2018. november 25.

GSM hívásindítás

Hívás kezdeményezése egyszerűen:

- Használjuk fel a beépített megoldást!

Jogosultság kérése:

- `android.permission.CALL_PHONE`
- `android.permission.CALL_PRIVILEGED`

Bármilyen szám hívása,
beleértve a
vészhívószámokat is.

Példakód 😊:

```
startActivity(Intent (
    Intent.ACTION_CALL, //Intent. ACTION_DIAL
    Uri.parse("tel:" + Uri.encode("+36307654321")))
))
```

MMI / USSD kódok miatt

Megoldás előnye, hogy már rendelkezik a híváshoz szükséges további gazdag funkciókkal.

URI séma használata:

- <http://tools.ietf.org/html/rfc3966>
- `tel:+3630...`

SMS küldése

Jogot kell kérni az AndroidManifest XML-ben:

- `android.permission.SEND_SMS`

SmsManager használata:

```
SmsManager sm = SmsManager.getDefault();
sm.sendTextMessage("+36307654321", null,
    "Hello world", null, null, null);
// sm.sendDataMessage(...); //data <- byte[]
```

Multipart SMS küldése, üzenet feldarabolásával:

```
val parts = sm.divideMessage("Nem rövid üzenet :-)");
sm.sendMultipartMessage(destAddress, null, parts,
    null, null);
```

Lásd még:

- [Telephony package](#), [PhoneNumberUtils](#)

SMS fogadása

Jogosultság kérése:

- `android.permission.RECEIVE_SMS`

BroadcastReceiver használatával:

```
fun onReceive(context: Context, intent: Intent) {  
    val messages = intent.extras?.get("pdus") as Array<*>  
    val sms = arrayOfNulls<SmsMessage>(messages.size)  
    for (i in messages.indices) {  
        sms[i] = SmsMessage.createFromPdu(  
            messages[i] as ByteArray  
        )  
    }  
    val firstMessage = sms[0]?.messageBody  
}
```

Akik szeretnek a felszín alá nézni → [PDU format](#)

NFC – Near Field Communication

Rövid hatótávú vezeték nélküli technológia (< 4cm)

13,56 Mhz (HF - High Frequency, rövidhullám)

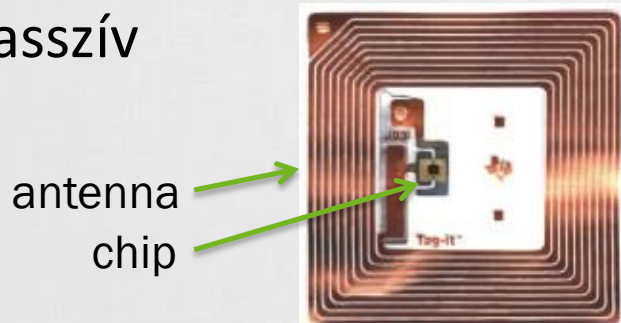
Mint az RFID, de célirányosan kidolgozott szabvány

Adatátviteli sebesség: 100-850 kbit/s

Rádiófrekvenciás tárgy-, és személyazonosítás

NFC címke fajtái:

- Aktív (saját áramforrás)
- Passzív



[Android 2.3 óta támogatott](#)

[How to NFC – Google I/O 2011](#)

[NFC kezelése Android alatt](#)



[Kép forrása](#)

[Android Beam !!!](#)

USB kezelése

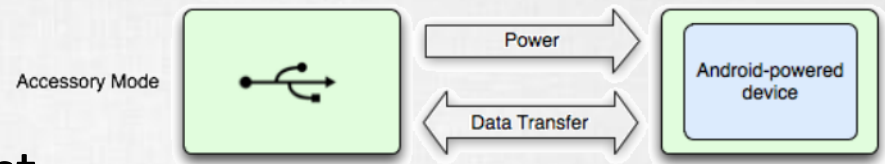
USB perifériák és USB eszközök támogatása.

- Android 3.1 (API 12) óta érhető el. ☹️
- Android 2.3.4-re (API 10) portolva, de csak az Accessory mód.

Két mód

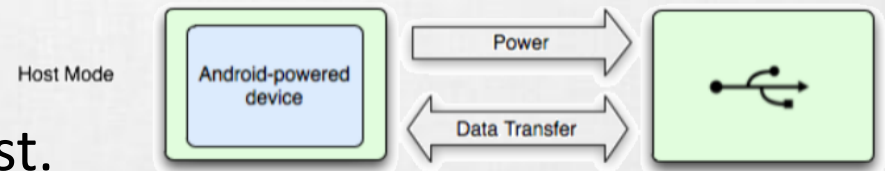
■ USB Accessory mode

- Külső USB hardver, mint USB Host.
- Az eszköznek Android Accessory protokolt be kell tartania.
- Példák: robotvezérlők, dokkolók, diagnosztikai és zenei eszközök, kártyaolvasók, stb...



■ USB Host mode

- Android készülék lesz az USB Host.
- Példák: digitális fényképezők, kamerák, billentyűzetek, egerek, játékvezérlők, stb...



USB programozásáról bővebben

- <http://developer.android.com/guide/topics/connectivity/usb/index.html>



Bluetooth



Bluetooth Android alatt

Támogatásának fejlődése

- API level 1 (1.0): interfész, nincs konkrét implementáció
- **API level 5 (2.0): első hivatalos implementáció** elérhető
- API level 8 (2.2): voice dialing, car & desk dock támogatás
- API level 10 (2.3.3): nonsecure socket támogatás
- **API level 11 (3.0): A2DP és HSP profil** támogatás
- API level 14 (4.0): Health Device Profile (HDP) támogatás
- **API level 18 (4.3):** BLE támogatás
- API level 21 (5.0): BLE *wearable* és *beacon* támogatás
- API level 28 (P): Bluetooth HID API

API használatának főbb lépései

- Bluetooth elérhetőségének ellenőrzése
- Párosított eszközök elérése, újak felderítése
- Kiválasztott eszközhöz csatlakozás
- Adatcsere két eszköz között

Görbe tükör

WiFi

Tethering

NFC

USB

WiFi Direct, Beam

```
android.permission.BLUETOOTH  
android.permission.BLUETOOTH_ADMIN
```


1. Bluetooth elérhetősége

BluetoothAdapter megszerzése

```
var adapter = BluetoothAdapter.getDefaultAdapter()
```

Elérhetőség ellenőrzése

```
if (adapter == null) {  
    // Bluetooth nem támogatott  
} else if (!adapter.isEnabled) {  
    // BT nincs engedélyezve, küldünk egy kérést  
    startActivity(Intent(  
        BluetoothAdapter.ACTION_REQUEST_ENABLE  
    ))  
} else {  
    // munka Bluetooth-al  
    // ...  
}
```

2. Párosított és új eszközök felderítése

Felderítés aszinkron módon, BroadcastReceiver-rel

```
fun onReceive(context: Context, intent: Intent) {  
    if (intent.action == BluetoothDevice.ACTION_FOUND) {  
        val bd = intent.getParcelableExtra<BluetoothDevice>(BluetoothDevice.EXTRA_DEVICE)  
    }  
    // TODO with bd  
}  
  
registerReceiver(receiver,  
    IntentFilter(BluetoothDevice.ACTION_FOUND))
```

Párosított eszközök lekérése (szinkron)

```
adapter.startDiscovery();  
  
val devices = adapter.getBondedDevices();  
  
for(BluetoothDevice bd in devices)  
    bd.getName();
```

3. Csatlakozás eszközökhöz

RFCOMM szabványt használnak a készülékek

UUID használata, generálható is: UUID.randomUUID();

- „If you are connecting to a Bluetooth serial board then try using the well-known SPP UUID 00001101-0000-1000-8000-00805F9B34FB. However if you are connecting to an Android peer then **please generate your own unique UUID.** „

Bluetooth szerver

```
val bss : BluetoothServerSocket =  
    adapter.listenUsingRfcommWithServiceRecord(  
        BT_NAME, BT_UUID);
```

```
// Kliensre vár...
```

```
val socket : BluetoothSocket = bss.accept();  
// Socket lekezelése thread  
socket.close();
```

4. Adatcsere

Bluetooth kliens

```
// Pl. listából kiválasztva, felderítés után
val device : BluetoothDevice = ...;
// Ugyanaz az UUID, mint szervernél

val socket : BluetoothSocket =
device.createRfcommSocketToServiceRecord(uuid);
// Kapcsolódás
socket.connect();
// Socket kezelése, mint szervernél
```

thread

Sikeres kapcsolódást után Socket kezelése

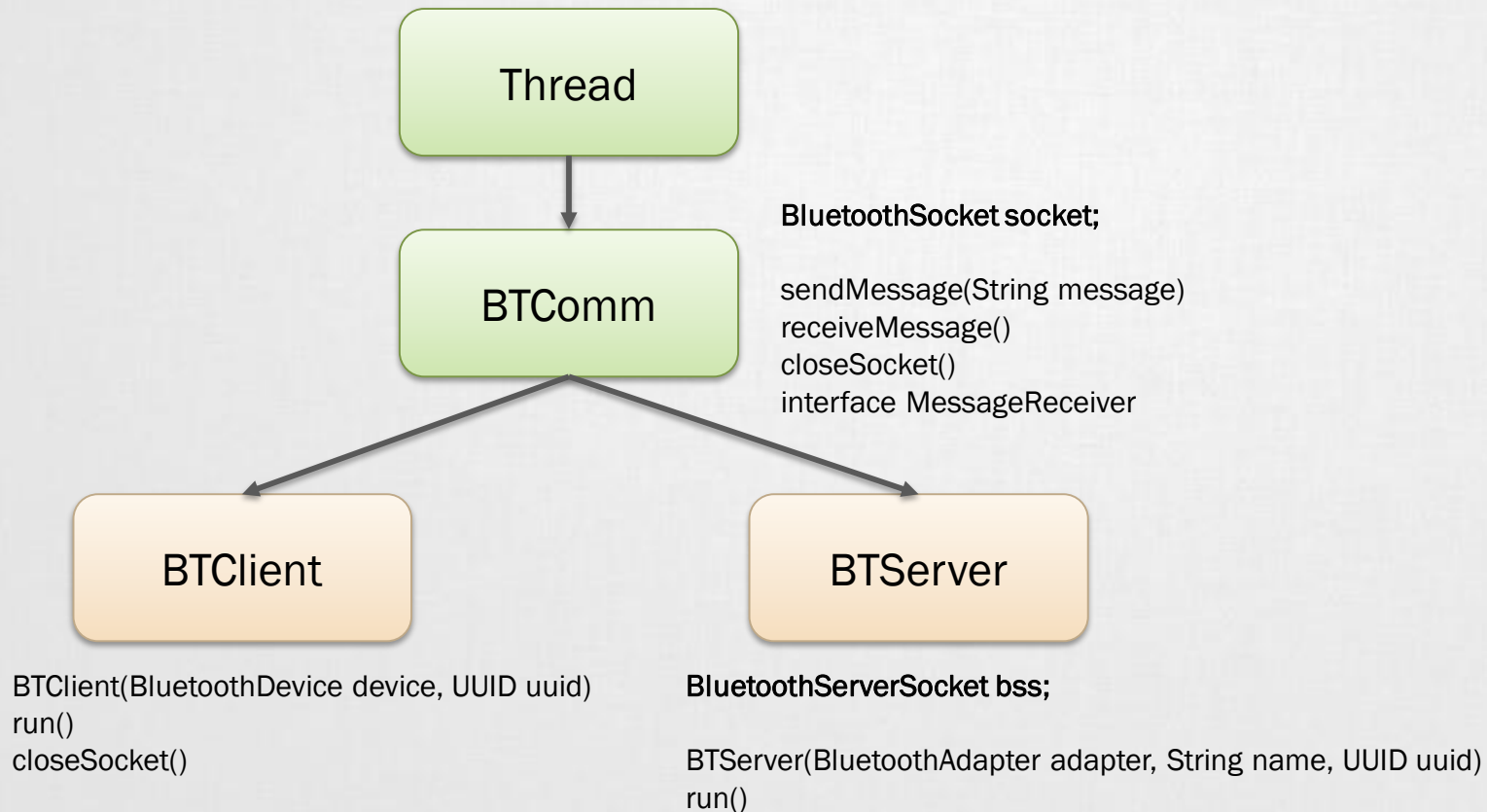
- socket.getInputStream();
- socket.getOutputStream();
- socket.getRemoteDevice();

Példa program

Bluetooth Chat:

- [http://nik.uni-obuda.hu/malk/android/ea_2012_osz/sources/08 - BluetoothChatDemo.zip](http://nik.uni-obuda.hu/malk/android/ea_2012_osz/sources/08_-_BluetoothChatDemo.zip)

Felépítése



BLE : Bluetooth Low Energy

- Android 4.3+ (API level 18)
- Eszközök felfedezése, szolgáltatások lekérdezése
- Minimális mennyiségű adatforgalom eszközök között
- Proximity szenzorok, Google Beacons, heart rate monitoring, ...
- Szignifikánsan alacsony energia felhasználás

Jogosultság kérése

```
<uses-permission android:name="android.permission.BLUETOOTH"/>  
<uses-permission android:name="android.permission.BLUETOOTH_ADMIN"/>  
<uses-feature android:name="android.hardware.bluetooth_le" android:required="true"/>
```

Támogatottság ellenőrzése

```
if (!getPackageManager().hasSystemFeature(PackageManager.FEATURE_BLUETOOTH_LE)){ ... }
```

Fel és leiratkozás

```
mBluetoothAdapter.startLeScan(mLeScanCallback);  
mBluetoothAdapter.stopLeScan(mLeScanCallback);
```

Bluetooth Beacon Specifications



[EddyStone](#)



[AltBeacon](#)



[iBeacon](#)

Beacon készülékgyártók:

https://developers.google.com/beacons/eddystone#beacon_manufacturers