

# Android alkalmazásfejlesztés

Kommunikáció Androidon

GSM hívás

SMS küldés, fogadás

Bluetooth

NFC

USB

OE-NIK

2012. november 6.

**Sicz-Mesziár János**

sicz-mesziar.janos@

nik.uni-obuda.hu



# Kommunikációs eszközök Androidon

## ◎ Wi-Fi

- Ma már minimum: a / b / g / n szabványok

## ◎ Mobilinternet

- EDGE, UMTS, HSPDA, LTE, ...
- Rosszabb lefedettség, mint GSM esetén

## ◎ GSM / SMS / GPRS

- Jó lefedettség, klasszikus, már megszokott
- Elsődlegesen beszédátvitel volt a cél

## ◎ Bluetooth

- Egy hálózatban 1 masterhez max. 7 másik eszköz csatlakozhat

## ◎ NFC

- Nagyon rövid hatótávú kommunikáció
- Még kevés készülékben található meg

## ◎ USB

- Vezetékes adatátvitel

## ◎ ~~Infra~~

- ~~Már nem jellemző, 1-2 készülék, set-top boxok~~

# 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  
vész hívószámokat is.

## ◎ Példakód 😊:

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

**v. ACTION\_DIAL**

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:

```
ArrayList<String> 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:

```
void onReceive(Context context, Intent intent) {  
    Bundle bundle = intent.getExtras();  
    Object[] messages = (Object[])bundle.get("pdus");  
    SmsMessage sms[] = new  
        SmsMessage[messages.length];  
    for (int i=0; i<messages.length; i++){  
        sms[i] = SmsMessage.createFromPdu  
            (byte[])messages[i]);  
    }  
    String elsoUzi = sms[0].getMessageBody();  
}
```

## ⊙ 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



Android Beam !!!

# USB

- ◉ 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

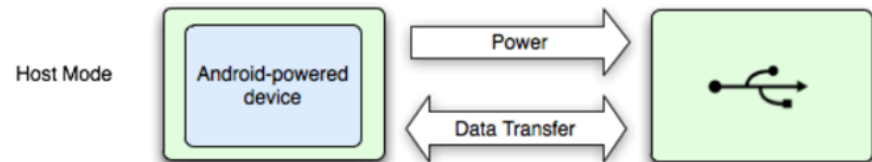
- **USB Accessory mode**

- ◉ 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<sup>®</sup>**





# Bluetooth Android alatt

## ☉ Támogatásának fejlődése

- API level 1 (1.0): interfész, nincs konkrét implementáció, lásd.: [backport API level 1.5-ig](#)
- 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

*Görbe tükör*

WiFi

Tethering

NFC

USB

WiFi Direct  
Android Beam

## ☉ API használatának főbb lépései

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

**Bluetooth emulátoron  
még nem támogatott!**

## ☉ Jogosultság kérése:

- android.permission.BLUETOOTH
- android.permission.BLUETOOTH\_ADMIN

# 1. Bluetooth elérhetősége

## ◎ BluetoothAdapter megszerzése

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

## ◎ Elérhetőség ellenőrzése

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

## 2. Párosított és új eszközök felderítése

### ◎ Felderítés aszinkron módon, BroadcastReceiver-rel

```
void onReceive(Context context, Intent intent) {
    String action = intent.getAction();
    if(action.equals(BluetoothDevice.ACTION_FOUND)) {
        BluetoothDevice bd =
            intent.getParcelableExtra(
                BluetoothDevice.EXTRA_DEVICE);
this.registerReceiver(btReceiver, new
    IntentFilter(BluetoothDevice.ACTION_FOUND));
    }
}
```

### ◎ Párosított eszközök lekérése (szinkron)

```
adapter.startDiscovery();
Set<BluetoothDevice> devices =
    adapter.getBondedDevices();
for(BluetoothDevice bd : devices)
    bd.getName();
```

# 3. Csatlakozás az eszközhöz

- ◎ 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

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

```
// Kliensre vár...
```

```
BluetoothSocket socket = bss.accept();
```

```
// Socket lekezelése
```

thread

```
socket.close();
```

## 4. Adatcsere

### ◎ Bluetooth kliens

```
// Pl. listából kiválasztva, felderítés után
BluetoothDevice device = ...;
// Ugyanaz az UUID, mint szerVERNél
BluetoothSocket socket =
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

