

Android alkalmazásfejlesztés

Kommunikáció

- Wi-Fi, Mobil internet
- Bluetooth
- GSM / GPRS és SMS
- NFC

Sicz-Mesziár János

sicz.mj@gmail.com

2011. április 21.

OE-NIK



Kommunikációs eszközök Androidon

◎ Wi-Fi

- Ma már minimum: a / b / g szabványok, újabban már n szabvány is

◎ Bluetooth

- 2.4 GHz-es frekvenciasáv
- Egy hálózatban 1 masterhez max 7 másik eszköz csatlakozhat

Osztály	Teljesítmény	Hatótáv
1	100 mW (20 dBm)	100 méter
2	2,5 mW (4 dBm)	10 méter
3	1 mW (0 dBm)	1 méter

◎ GSM / GPRS / SMS

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

◎ Mobil internet

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

◎ NFC

- Nagyon rövid hatótávú kommunikáció

GSM	2G
HSCSD	
GPRS	2.5G
EDGE	
UMTS	3G
HSDPA	

Internet és hálózati kommunikáció

- ◎ Wi-Fi kezelése a Wi-Fi API-kon keresztül:
 - elérhető Access Pointok, csatlakozott hálózat sebessége
 - IP-cím, autentikáció
 - kapcsolódás, kapcsolatbontás, ...
- ◎ Wi-Fi Tethering üzemmód 2.2-es Android óta
- ◎ Adatátvitel módja detektálható (Wi-Fi vs. Mobilnet)
- ◎ IP szabvány szállítási rétegében:
 - **TCP** : kapcsolat orientált, csomag megérkezést megerősíti, és a csomagok sorrendjéről is gondoskodik
 - **UDP** : gyors, apró üzenetváltások, de nem sorrendtartó, valamint a csomagok megérkezése nem garantált
- ◎ Internet használathoz jogot kell kérnünk:
 - android.permission.INTERNET

UDP adatátvitel

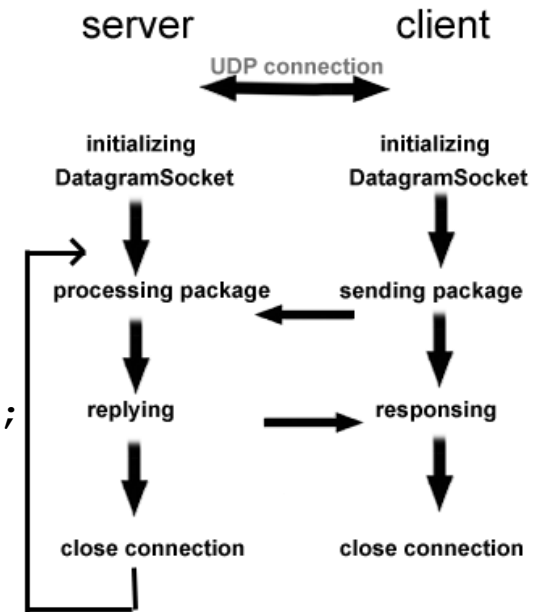
◎ Jogosultság: `android.permission.INTERNET`

◎ Szerver oldal

```
int serverPort = 50004;
InetAddress ip =
    InetAddress.getByName("192.168.1.1");
byte[] buf = new byte[1024 * 65];
DatagramPacket receivePacket =
    new DatagramPacket(buf, buf.length);
DatagramSocket socket = new
    DatagramSocket(serverPort);
socket.receive(receivePacket);      thread
receivePacket.getData();
```

◎ Kliens oldal

```
String msg = "Hello UDP Package";
byte[] msgByte = msg.getBytes();
DatagramSocket socket = new DatagramSocket();
InetAddress serverIP = InetAddress.getByName("192.168.1.1");
socket.connect(serverIP, 50004);
socket.send(new DatagramPacket(msgByte, msgByte.length));
```



[Kép forrása](#)

Android 1.5 alatt még bug-os volt.

TCP adatátvitelt

◎ TCP client:

```
Socket socket = new Socket();
socket.connect(new InetSocketAddress(/* Cím */,
                                   /* port */));

byte[] buffer = new byte[1024];
InputStream in = socket.getInputStream();
int len = 0;
while((len = in.read(buffer)) != -1) {           thread
    /* valamit csinálunk */
}
in.close();
```

◎ TCP server:

```
ServerSocket server = new ServerSocket(/* port */);
while(true) {
    Socket client = server.accept();
    /* Klient szokás új szálon kezelni a      thread
       továbbiakban */
}
```

Fájl letöltése URL alapján

◎ Adott URL tartalmának letöltése egy fájlba:

```
URL url = new URL("http://nik.uni-obuda.hu/malk/");
File malkFile = new File("/sdcard/malk.html");
URLConnection ucon = url.openConnection();
InputStream is = ucon.getInputStream();
FileOutputStream fos = new
    FileOutputStream(malkFile);
byte[] buffer = new byte[1024];
int len = 0;
while((len = is.read()) != -1)
    fos.write(buffer, 0, len);
fos.close();
is.close();
```

Ha SD-kártyára írjuk ne felejtünk el jogot kérni hozzá:
android.permission.WRITE_EXTERNAL_STORAGE

Bluetooth

◎ Android 2.0 óta támogatott (API level: 5)

◎ Jogosultság kérése:

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

◎ Kell egy Bluetooth adapter

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

◎ Elérhetőség ellenőrzése

```
if(adapter == null){  
    /* Bluetooth nem támogatott */  
}else if(!adapter.isEnabled()){  
    /* BT nincs engedélyezve */  
}else{ /* munka Bluetooth-al */ }
```

Bluetooth eszközök felderítése

1. Ismert v. korábban párosított eszközök lekérése

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

2. Új eszközök felderítése: egy BroadcastReceiver-rel értesülünk az új eszközökről

```
String action = intent.getAction();  
if(action.equals(BluetoothDevice.ACTION_FOUND)) {  
    BluetoothDevice bd =  
        intent.getParcelableExtra(  
            BluetoothDevice.EXTRA_DEVICE);  
  
    if(bd.getBondState() !=  
        BluetoothDevice.BOND_BONDED)  
        bd.getName();
```

```
this.registerReceiver(btReceiver, new  
    IntentFilter(BluetoothDevice.ACTION_FOUND));
```


Bluetooth adatcsere

⊙ Az alábbiak csak iránymutatás

⊙ Kell egy UUID: Universally unique identifier

```
UUID myUUID = UUID.randomUUID();
```

⊙ Socket-ek használata RFCOMM használattal

■ BluetoothServerSocket a szerverhez

```
adapter.listenUsingRfcommWithServiceRecord(String,  
                                             UUID);
```

■ BluetoothSocket a kliensekhez

```
bd.createRfcommSocketToServiceRecord(myUUID);
```

■ Socket-en InputStream/OutputStream-ek írás olvasása, a megszokott módon

⊙ Konkrét példa:

■ Hivatalos minta: Bluetooth Chat

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"))  
));
```

MMI / USSD kódok miatt

⊙ A megoldás előnye, hogy direkt erre a célra kialakított megoldást használjuk.



⊙ Jó, mert a kész megoldás, már rendelkezik a híváshoz szükséges további gazdag funkciókkal.

SMS küldés és fogadás

◎ SMS küldéshez jogot kell kérni a Manifest XML-ben:

- `android.permission.SEND_SMS`

◎ SMS küldés egyszerűen az SmsManager-rel:

```
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);
```

◎ SMS fogadás:

- BroadcastReceiver használatával. [Jó kis leírás itt.](#)

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 érhető el támogatás NFC-hez

