

## I. 288.: Utcai WiFi térkép

Azért ezt választottam, mert az iskolánkban viszonylag kevés WiFi hálózat található.

Az adatok gyűjtéséhez saját programot készítettem androidos mobilomra a <http://marakana.com/forums/android/examples/40.html> weboldalon található példát alapul véve (lásd a dokumentum végén).

Az adatok gyűjtését gyalog végeztem egy kb. 1 km-es szakaszon a lakhelyemen, Nagytarcsán.

Az adatok grafikus szemléltetéséhez a [http://www.alyrice.net/wifi\\_mapping](http://www.alyrice.net/wifi_mapping) weboldalon található perl szkriptet használtam.

Az adók nevének, csatornájának, biztonságának elemzését egy Visual Basic programmal oldottam meg (lásd a dokumentum végén).

A térképen ábrázoláshoz a Google Earth szoftvert használtam.

---

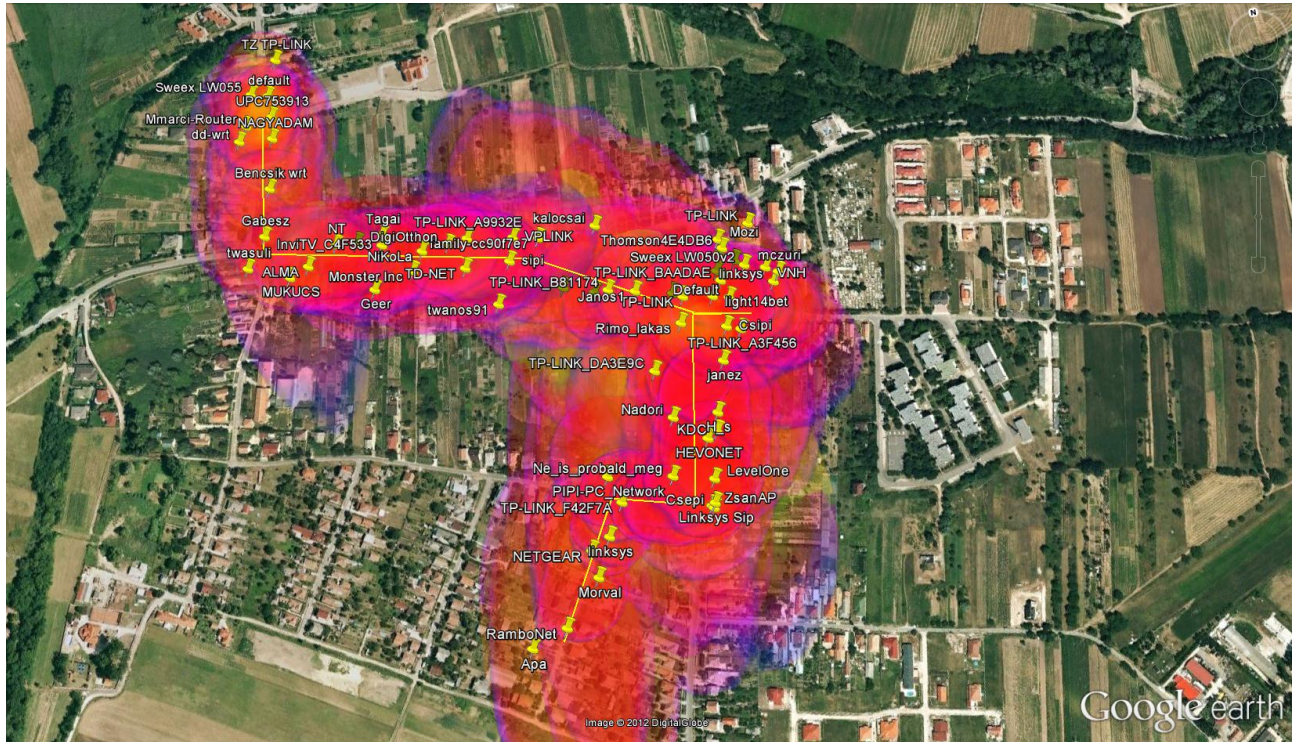
Egy-egy WiFi AP (Access Point) ilyesmi jelerősség-képet adott:



A jelerősségtérképen a zöld a legjobb jelerősséget jelenti, a sárga a jót, a lila a gyengét... .

Az adók helyét megpróbáltam a legjobb jelerősségű pontokba rakni.

A végső térkép:



Az AP-k szétválogatása aszerint, hogy milyen nevük (SSID-jük) van:

MAC Address	Name	Channel	Security
<input type="checkbox"/> 00:02:6f:22:c6:66	twanos91	2457	
<input type="checkbox"/> 00:30:4f:59:dc:e2	Geer	2437	[WPA2-PSK-CCMP-preauth]
<input checked="" type="checkbox"/> b0:48:7a:ca:db:24	DigiOthton	2427	[WPA2-PSK-TKIP+CCMP][WPS]
<input type="checkbox"/> 08:86:3b:1e:31:bc	sipi	2457	[WPA2-PSK-CCMP][WPS]
<input checked="" type="checkbox"/> 08:18:1a:c4:f5:33	InviTV_C4F533	2412	[WPA-PSK-TKIP][WPA2-PSK-TKIP]
<input type="checkbox"/> 74:ea:3a:f4:3a:0a	Tagai	2412	[WPA-PSK-CCMP][WPA2-PSK-CCMP][WPS]
<input type="checkbox"/> 00:1b:fc:3a:db:9b	MUKUCS	2412	[WPA-PSK-TKIP]
<input type="checkbox"/> 00:19:db:01:b5:40	Monsters Inc	2412	[WEP]
<input checked="" type="checkbox"/> 74:ea:3a:da:3e:9c	TP-LINK_DA3E9C	2412	
<input type="checkbox"/> d8:5d:4c:85:cf:4c	NT	2412	[WPA-PSK-CCMP][WPA2-PSK-CCMP][WPS]
<input type="checkbox"/> 54:e6:fc:ce:94:6c	GABESZ	2412	[WPA-PSK-TKIP+CCMP][WPA2-PSK-TKIP+CCMP][WPS]
<input checked="" type="checkbox"/> 38:60:77:d3:59:ad	UPC753913	2412	[WPA2-PSK-TKIP+CCMP][WPS]
<input type="checkbox"/> 00:1e:8c:7e:ef:70	default	2437	[WPA2-PSK-TKIP+CCMP]
<input type="checkbox"/> 00:17:b7:20:22:0c	twasuli	2462	
<input type="checkbox"/> b0:48:7a:b0:02:c8	dd-wrt	2462	[WPA2-PSK-TKIP]
<input type="checkbox"/> 00:1e:8c:70:e0:2b	NAGYADAM	2412	[WPA-PSK-TKIP+CCMP]
<input type="checkbox"/> 00:23:69:f4:6f:7c	Mmarci-Router	2437	[WPA-PSK-TKIP]
<input checked="" type="checkbox"/> 00:18:4d:4a:ff:74	wireless	2462	[WPA-PSK-TKIP]
<input type="checkbox"/> 00:1d:7e:e3:23:e3	Bencsik wrt	2452	[WPA-PSK-TKIP]
<input checked="" type="checkbox"/> 00:25:86:b3:bd:50	TD-NET	2437	[WPA-PSK-TKIP+CCMP][WPA2-PSK-TKIP+CCMP-preauth]
<input type="checkbox"/> 00:0c:42:1f:02:be	geri	2427	[WPA2-PSK-TKIP]
<input type="checkbox"/> 00:0c:76:a9:a5:fb	Othton	2442	[WEP]
<input checked="" type="checkbox"/> 00:0e:2e:af:b7:ff	default	2462	
<input checked="" type="checkbox"/> 00:23:cd:15:2c:4a	TP-LINK	2412	[WPA2-PSK-TKIP+CCMP][WPS]
<input checked="" type="checkbox"/> 00:16:0a:11:26:58	Sweex LW055	2437	[WPA-PSK-TKIP]
<input checked="" type="checkbox"/> f8:d1:11:4a:14:f4	TZ TP-LINK	2412	[WPA-PSK-TKIP+CCMP][WPA2-PSK-TKIP+CCMP][WPS]

Összen	76	100%
gyári (pl. TP-LINK)	33	43%
személynév (pl. geri)	21	28%
egyéb (pl. RamboNet)	22	29%



A használt biztonság típusa:

Összesen	76	100%
Nincs/Nyílt	12	16%
WEP	7	9%
WPS	1	1%
WPA	9	12%
WPA2	47	62%

A használt csatorna (ez az AP frekvenciájából számítható ki (5 MHz-enként vannak a csatornák, kivéve a 14.-et, amit Japánban használnak), <http://www.moonblinkwifi.com/2point4freq.cfm>):

Összesen	76	100%
1	24	32%
2	0	0%
3	2	3%
4	5	7%
5	1	1%
6	18	24%
7	2	3%
8	1	1%
9	3	4%
10	3	4%
11	16	21%
12	1	1%
13	0	0%
14	0	0%

Az AP-k számlálásához használt program kódja:

```
Public Class Form1
    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles MyBase.Load
        Dim s$() = My.Computer.FileSystem.ReadAllText("C:\wifi2.txt").Split(New
String() {vbLf}, StringSplitOptions.RemoveEmptyEntries)
        Dim a As List(Of String) = New List(Of String)
        For i% = s.Length - 1 To 0 Step -1
            If Not a.Contains(s.ElementAt(i)) Then
                a.Add(s.ElementAt(i))
            End If
        Next
        For i% = a.Count - 1 To 0 Step -1
            CheckedListBox1.Items.Add(a.ElementAt(i))
        Next
    End Sub
    Private Sub CheckedListBox1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles CheckedListBox1.Click
        Text = CheckedListBox1.CheckedItems.Count
    End Sub
End Class
```

Az adatok gyűjtéséhez használt program kódja:

```
package hu.fbalazs.wifi;import java.io.FileWriter;import java.io.IOException;import
java.util.List;import android.app.Activity;import android.content.BroadcastReceiver;import
android.content.Context;import android.content.Intent;import android.content.IntentFilter;import
android.location.Location;import android.location.LocationListener;import
android.location.LocationManager;import android.net.wifi.ScanResult;import
android.net.wifi.WifiManager;import android.os.Bundle;import android.os.Handler;import
android.os.Vibrator;import android.view.View;import android.view.View.OnClickListener;import
android.widget.Button;import android.widget.TextView;import android.widget.Toast;
public class WifiDemo extends Activity implements OnClickListener {
    WifiManager wifi;
    BroadcastReceiver receiver;
    TextView textStatus;
    Button buttonScan;
    Vibrator v;
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        v = (Vibrator) this.getSystemService(Context.VIBRATOR_SERVICE);
        textStatus = (TextView) findViewById(R.id.textStatus);
        buttonScan = (Button) findViewById(R.id.buttonScan);
        buttonScan.setOnClickListener(this);
        wifi = (WifiManager) getSystemService(Context.WIFI_SERVICE);
        if (receiver == null)
            receiver = new WifiScanReceiver(this);
        registerReceiver(receiver, new IntentFilter(
            WifiManager.SCAN_RESULTS_AVAILABLE_ACTION));
    }
    @Override
    public void onStop() {unregisterReceiver(receiver); }
    Handler handler = new Handler();
    Runnable runnable = new Runnable() {public void run() {wifi.startScan();}};
    public void onClick(View arg0) {
        if (buttonScan.getText().equals("Scan")) {
            handler.post(runnable);
            buttonScan.setText("Stop");
        } else {
            buttonScan.setText("Scan");
        }
    }
}
class WifiScanReceiver extends BroadcastReceiver {
    WifiDemo wifiDemo;
    StringBuilder str, str2;
    FileWriter f;
    LocationManager gps;
    GPSTListener gpsl;
    public boolean willstop = false;

    public WifiScanReceiver(WifiDemo wifiDemo) {
        super();
        this.wifiDemo = wifiDemo;
        str = new StringBuilder();
        str2 = new StringBuilder();
        gps = (LocationManager) wifiDemo
            .getSystemService(Context.LOCATION_SERVICE);
        gpsl = new GPSTListener();
        gps.requestLocationUpdates(LocationManager.GPS_PROVIDER, 0, 0, gpsl);
    }
    @Override
    public void onReceive(Context c, Intent intent) {
        ((Vibrator)WifiDemo.getSystemService(Context.VIBRATOR_SERVICE)).vibrate(100);
        List<ScanResult> results = wifiDemo.wifi.getScanResults();
        if (results == null)return;
        for (ScanResult res : results) {
            str.append(String
                .format("N %f\t E %f\t( %s ) BSS\t( %s ) 00:00:00 (GMT)\t[ 10
%d 149 ] # ( )\t0\t0\t0\t0\t0\n",
                    gpsl.latitude, gpsl.longitude, res.SSID,
                    res.BSSID,
                    res.level + 249).replace(',', '.'));
            str2.append(res.BSSID + "\t" + res.SSID + "\t" + res.frequency
```

```

        + "\t" + res.capabilities + "\n");
    }
    wifiDemo textStatus.setText(str.toString());
    try {
        f = new FileWriter("/sdcard/wifi.txt", true);
        f.append(str.toString());
        f.flush();
        f = new FileWriter("/sdcard/wifi2.txt", true);
        f.append(str2.toString());
        f.flush();
        Toast.makeText(wifiDemo, "log on sdcard", Toast.LENGTH_SHORT);
    } catch (IOException e) {}
    str = new StringBuilder();
    str2 = new StringBuilder();
    if (wifiDemo.buttonScan.getText().equals("Stop"))
        wifiDemo.handler.postDelayed(wifiDemo.runnable, 250);
}
class GPSListener implements LocationListener {
    public double longitude = 0, latitude = 0;
    public float accuary = 0;
    public void onLocationChanged(Location arg0) {
        longitude = arg0.getLongitude();
        latitude = arg0.getLatitude();
        accuary = arg0.getAccuracy();
    }
    public void onProviderDisabled(String arg0) {accuary = 0; }
    public void onProviderEnabled(String arg0) {}
    public void onStatusChanged(String arg0, int arg1, Bundle arg2) {}
}
}

```

---

**Fényes Balázs** 9. o. t.  
 Budapest, Szerb A. Gimn.  
 feny.es.balazs@freemail.hu