

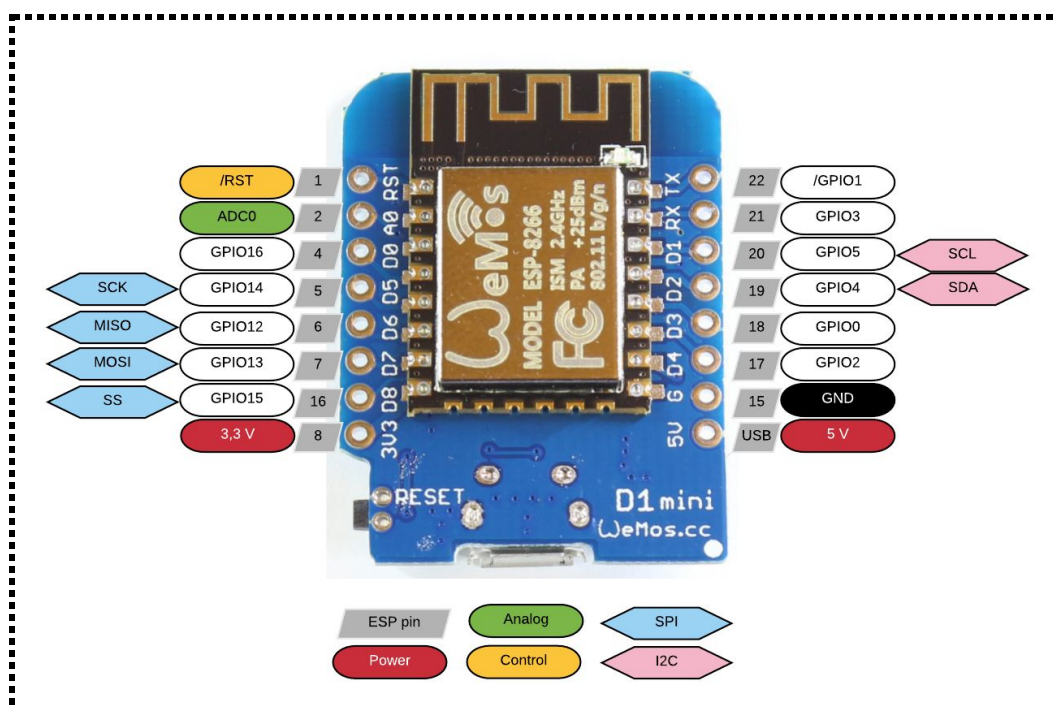
Az időjárásjelentők legszívesebben felgyűjtanák ezt az alkalmazást!

Dobai Botond, Kun Szabolcs

A feladat

A projektünkben egy Wemos D1 mini és egy bluetooth-modul segítségével készítettünk el egy időjárási adatokat lekérdező, egyszerű rendszert. Az eszközt és a telefont Bluetooth LE alapú kommunikációval kötöttük össze. A mobil egy androidos alkalmazások keresztül kérdezhet le különféle időjárási adatokat tőle (jelenleg: aktuális hőmérséklet és páratartalom).

Az eszköz egy ESP8266-os Wifi-modul segítségével lekérdezi [OpenWeatherMaps API](#) budapesti adatait, majd a lényegi információt a lekért json-fájlból kiszedve küldi tovább a telefonos alkalmazás számára.



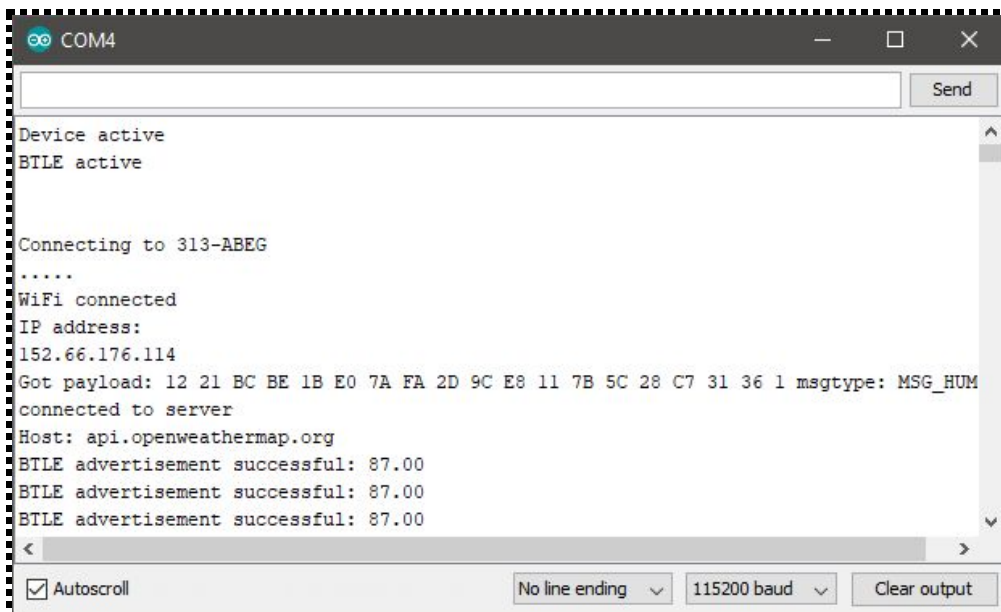
Nehézségek

A legnagyobb nehézséget talán a használati eset megtalálása okozza, mivel a használt környezetben alig pár bájt használható adatküldésre. Valószínűleg nem sikerült megtalálni a legértelmesebb felhasználást, de gyakorlásnak ez is megteszi. A hardver és a hozzá fellelhető könyvtárak alapvetően jól használhatóak, nagyon sokféle ötlet megvalósítható velük.

Megvalósítás I.

A Wemosra írt kódot [Arduino IDE](#) segítségével valósítottuk meg. A Bluetooth LE kommunikációhoz a BTLE, RF24 könyvtárakat, a Wifihez és a Http-kéréshez az [ESP8266WiFi](#)-at, a json-parsinghoz pedig a [ArduinoJson](#)-at használtuk fel (mind letölthető az IDE könyvtárkezelőjéből). A lábkiosztás megtalálható a forráskódban.

A program először inicializálja a BTLE-specifikus változókat a kapcsolathoz, illetve a megadott SSID és jelszó segítségével csatlakozik fel a wifihálózatra. Ezt követően a [BTLE üzenetfogadásra áll](#). Ha sikerül a jogosulttól (mobil app) csomagot fognia, akkor a kapott információval - milyen adatot adjon vissza - megkezdí a lekérdezést. Ehhez egy HTTP GET lekérdezést hajt végre az OWM API felé, és a válaszban kapott json-stringet pedig a parser segítségével kiszedi string formájában. A kapott eredményt közzé teszi a BTLE csatornán keresztül a mobil app számára.

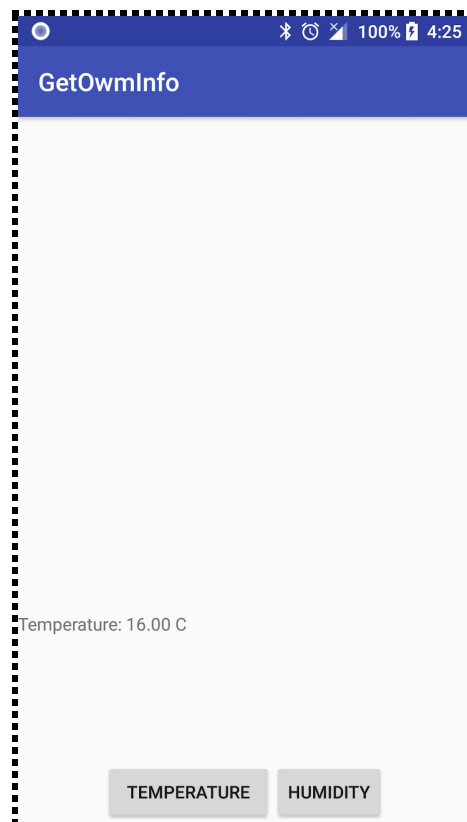


```
COM4
Device active
BTLE active

Connecting to 313-ABEG
.....
WiFi connected
IP address:
152.66.176.114
Got payload: 12 21 BC BE 1B E0 7A FA 2D 9C E8 11 7B 5C 28 C7 31 36 1 msgtype: MSG_HUM
connected to server
Host: api.openweathermap.org
BTLE advertisement successful: 87.00
BTLE advertisement successful: 87.00
BTLE advertisement successful: 87.00
```

Megvalósítás II.

Az Android alkalmazás jelenlegi állapotából egy nagyon egyszerű felületből, és a lekérdezési logikából áll. Az Activity-nk onCreate(..) metódusában kérjük le a felületi elemek referenciáit, illetve adjuk meg a gombokhoz tartozó eseménykezelőket. Az eseménykezelők lényegében csak letiltják ideiglenesen a gombokat, majd elindítják a lekérési logikát.



A kommunikációhoz az android.bluetooth.le könyvtár lehetőségeit tudjuk kihasználni. Ehhez egy [nagyon hasznos tutorialt](#) használtunk fel, a két linkelt oldalon megtalálhatóak a főbb technikai részletek. Lényegében létre kell hozni az adatstruktúrát, illetve be kell állítanunk az 'adapterünket' a csomagok küldésére/fogadására. (Fontos a források autentikációja is a helyes működéshez!)

Elindítjuk egy-egy aszinkron szálon a kérést tartalmazó csomagok küldését, illetve a végeredményt tartalmazó csomagok fogadását. A további logika (mikor álljon le a kommunikáció, felület állapotának változtatása, lekérdezett adatok megjelenítése) a fogadó szolgáltatás callback-függvényében lett megvalósítva. További nélkülözhetetlen elem egy timeout indítása időzített módon, hogy ne

The screenshot displays the Android Studio interface with the following components:

- Top Bar:** Contains icons for Run, Debug, and other development tools. The status bar at the bottom indicates the current mode is 'Run' and the device name is 'COM4'.
- Console Output:** Shows the following logs:
 - `I/InstantRun: starting instant run server: is main process`
 - `E/ANDR-PERF-JNI: IPerf::tryGetService failed!`
 - `D/OpenGLESRenderer: HWUI GL Pipeline`
 - `I/Adreno: QUALCOMM build : 7f08991, I8a9bdcf8d3`
 - `Build Date : 12/21/17`
 - `OpenGL ES Shader Compiler Version: EV031.20.00.04`
 - `Local Branch :`
 - `Remote Branch : refs/tags/AU_LINUX_ANDROID_LA.UM.6.5.R`
 - `Remote Branch : NONE`
 - `Reconstruct Branch : NOTHING`
 - `I/vndksupport: sphal namespace is not configured for this process. Loading /vendor/`
 - `I/Adreno: PFP: 0x005ff087, ME: 0x005ff063`
 - `I/OpenGLESRenderer: Initialized EGL, version 1.4`
 - `D/OpenGLESRenderer: Swap behavior 2`
 - `I/vndksupport: sphal namespace is not configured for this process. Loading /vendor/`
 - `D/BluetoothAdapter: isLeEnabled(): ON`
 - `D/BluetoothAdapter: isLeEnabled(): ON`
 - `D/BluetoothAdapter: isLeEnabled(): ON`
 - `D/BluetoothLeScanner: onScannerRegistered() - status=0 scannerId=6 mScannerId=0`
 - `D/BLE: Advertising onStartSuccess`
 - `D/BLE: Scanning onScanResult`
 - `D/BLE: resultString: 16.00`
 - `Scanning onScanResult: Temperature: 16.00 C`
 - `D/BluetoothAdapter: isLeEnabled(): ON`
 - `I/zygote64: Do partial code cache collection, code=30KB, data=26KB`
 - `After code cache collection, code=30KB, data=26KB`
 - `Increasing code cache capacity to 128KB`
- Logcat View:** Displays a series of BTLE advertisements, all with a status of 'successful' and a value of '16.00'. The log shows a sequence of advertisements from a device with address '12 21 BC BE 1B E0 7A FA 2D 9C E8 11 7B 5C 28 C7 31 36 0'.

