



Kiss Zoltán - Export Igazgató - Head of R&D - Kocsis Csaba, hardver mérnök - Veresegyházy Zsolt, vezető szoftverfejlesztő - Kiss Bence Mérnök-Informatikus hallgató, gyakornok - Dankó Tibor villamosmérnök- gyártás

## E-Innováció 4. - Alkalmazási példa

**A** február havi számmal kezdődően a Magyar Elektronika hasábjain egy új cikksorozatot indítottunk az Endrich Bauelemente Vertriebs GmbH legújabb innovációiról, melyekkel a hazai és nemzetközi kiállításokon és konferenciákon találkozhatnak az érdeklődő szakemberek az év folyamán. Jelen számunkban egy olyan mintarendszert szeretnénk bemutatni, mely tervezésekor felvonnaltattuk az E-IoT koncepció jónéhány elemét, kezdve az előző lapszámunkban bemutatott szenzor és kommunikációs pajzsoktól a felhő alapú adatbázison keresztül a WEB és a mobiltelefonos applikáció alapú adatmegjelenítésről. Az utolsó hetekben a Budapesti Műszaki és Gazdaságtudományi Egyetem Villamosmérnöki és Informatikai Karán végzős gyakornokunk segítségével elkészült a Fibocom MA510 NB-IoT/LTE-M és 2G GSM modemjének C++ könyvtára, az objektumorientált érzékelő és adatbeküldő szoftver és megszületett a legújabb adatmegjelenítő eszközünk, az Android mobilapplikáció is.

*Az otthoni okoseszközök, mint például a rádiós klíma, árnyékoló roló és világítótestek Bluetooth-on keresztül vezérlésére pedig kidolgoztuk egy szimulátor alkalmazást. Jelen írásunkban ezt a komplex okosotthon koncepciót részletezzük.*

### **Az E-IOT platform segítségével kialakított okosotthon mintaalkalmazás**

A korábbi írásainkban részletesen tárgyaltuk az E-IoT platform hardver elemeit, melyek jellemzően valamilyen népszerű mikrovezérlő (GD32 ARM/Risc-V, Microchip SAMD21 és SAMD51 vagy RP2040) köré épült saját fejlesztésű miniszámítógépek IoT funkciókkal kiegészítve.

Ezek a 2in1 (szenzor és kommunikációs pajzs) vagy 3in1 megoldások (Single Board Computer) egy készülékben egyesítik az IoT végpontok számára nélkülözhetetlen **szenzor**, az ezek adatgyűjtését irányító **vezérlő**, és az adatok továbbítására szolgáló vezetékmentes (LPWA) **kommunikáció** funkciókat.

A mikrovezérlő kiegészíthető további vezetékmentes lokális WLAN hálózatot megvalósító eszközökkel is, melyek

felhasználhatók további okoszenzorok rendszerbe integrálására, ilyen volt például a korábban bemutatott Neo.Cortec sub-Gigahertz ISM frekvenciasávban működő MESH szenzorhálózat, melyet az E-IoT eszközök nagy része támogat.

Amennyiben a lakásban lévő általános célú okoseszközök (klíma, árnyékoló roló, világítás) vezérlését is el kell végezni, akkor WiFi vagy Bluetooth kapcsolatot is kell integrálni a központi vezérlőnkbe. Így a most megvalósított koncepcionális okosotthon rendszer is kapott egy Bluetooth 2.0 modult, ami egy okostelefonon futó BlueTooth Terminál program segítségével képes megjeleníteni a szenzorok által mért adatokon alapuló döntésekhez tartozó

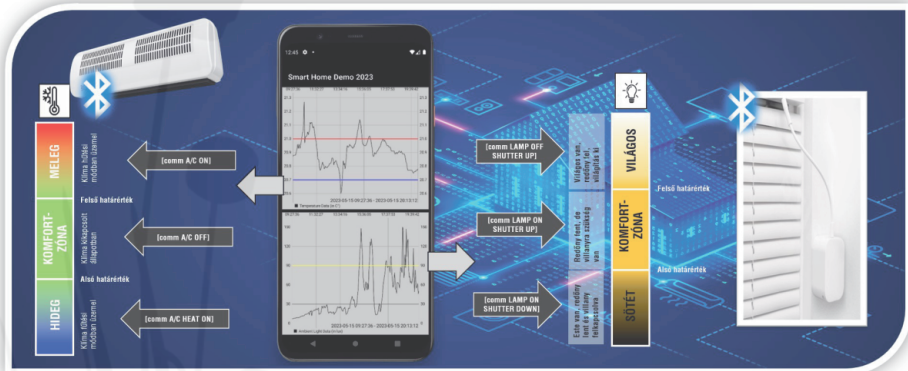
parancsokat, ezzel emulálva egy valós okosotthon rendszer lokális vezérlőhálózatát. A rendszer kihasználja az Endrich IoT szolgáltatásait, a mért szenzor adatokat felhőalapú adatbázisba juttatja a GSM hálózat segítségével (UDP), a mobil eszközökön valós idejű adatmegjelenítésre alkalmas és az MCU lokális vezetékmentes kapcsolaton keresztül parancsokat is képes adni az okoseszközöknek.

A megvalósított mintarendszer a következő funkciókkal rendelkezik:

- Hőmérséklet (és légnyomás) érzékelés integrált I<sup>2</sup>C digitális hőmérővel
- A látható fény intenzitásának mérése ALS szenzorral
- Előre beállított alsó és felső

## Klíma és redőny- illetve világításvezérlés

Érzékelés, vezérlés, adatok tárolása felhőben és megjelenítés



2 Endrich Bauelemente Vertriebs GmbH

www.endrich.com

1| Az okosotthon mintaal alkalmazás vezérlési struktúrája

határértékekből való kilépéskor BT 2.0 kommunikációval parancsokat küld az otthoni okoseszközök (klíma, a világítás és a redőny) számára (ezek emulációja BT terminálon)

- A mért szenzoradatok beküldése JSON táviratként a GSM modem segítségével az E-IOT CDB központi Cloud Szerverre

- A tárolt adatok lekérése az Endrich Szenzor API segítségével Android készülékre és grafikus ábrázolás (hasonló megoldás elérhető IoS készülékekre is)

- A mért adatok függvényében vezérlőparancsok küldése BT 2.0 kapcsolaton keresztül BT MASTER eszköz részére és onnan manuális beállítások fogadása

## A rendszer felépítése

Az alkalmazott központi hardver szinte bármelyik E-IOT eszköz, a RISC-V alapú E-IOT board, a tetszőleges SBC-hez UART-on keresztül illeszthető 3BIG-MOD és a hozzá tartozó kiértékelő lap, vagy az előző részben említett IoT Arduino-hoz vagy Raspberry Pi-hez fejlesztett IoT pajzs is lehet.

Közös elvárás, hogy szenzorokra van szükség, ahhoz pedig, hogy az általuk mért jellemzőket felhő alapú adatbázisba juttassuk, valamilyen kommunikációs eszközzel kell kiegészítenünk őket. Ahhoz, hogy az SBC IoT csomópontként

működhessen eddig is sok lehetősége volt a fejlesztőknek, hiszen a kereskedelmi forgalomban kapható miniszámítógépeken hozzáférhető I/O portok sokasága lehetővé teszi, hogy akár analóg, akár digitális (I<sup>2</sup>C) szenzorokat illesszenek a készülékhez, a UART porton keresztül pedig GSM modem képes az adatok felhőbe juttatására, azonban az E-IoT készülékek minden szükséges funkcióval rendelkeznek.

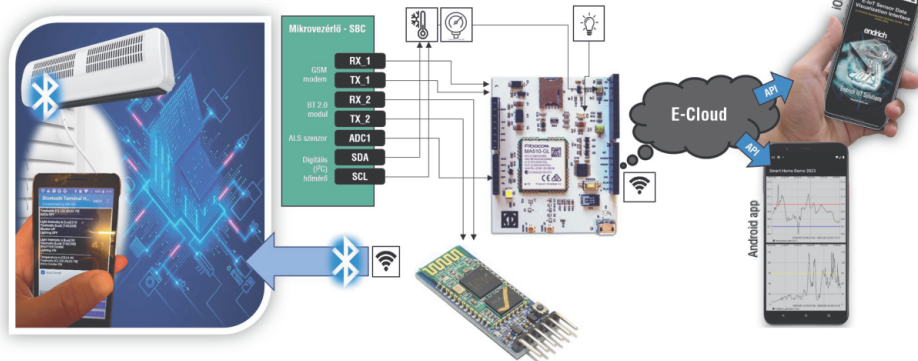
Bármelyik Endrich IoT pajzsra megtalálható a légnyomásszintet, a környezeti hőmérsékletet és a differenciális légnyomás alapján kalkulált magasságot érzékelő szenzor, és a látható tartományban, az emberi szem spektrális érzékenységet közelítően modellező ALS szenzor is.

Az MCU lap szabad portjai továbbra is használhatóak külső perifériák illesztésére, így egy második UART porton keresztül csatlakoztatható valamilyen BT eszköz, például az általunk használt HC-05 BT2.0 modul.

A pajzsra található egy az MCU UART portjával összekapcsolt LPWA modem is, mely a soros porton az MCU felől érkező AT parancsok segítségével a celluláris hálózatot használva juttatja el az adatokat az Endrich felhőalapú adatbázisába, vagy egyéb publikus felhőszolgáltatókhoz.

## Okosotthon alkalmazás blokkvázlata

Érzékelés, vezérlés, adatok tárolása felhőben és megjelenítés



### 2) Az E-IOT okosotthon rendszer felépítése

Az elérhető szolgáltatások közül alacsony energiafogyasztás mellett igénybe vehetjük az NB-IoT vagy az LTE-M LPWA hálózatokat, azonban ezek hiányában sem kell lemondanunk az adatküldésről, mert a modem képes a 2G/GPRS hálózaton is kommunikálni.

### Az adatok felhőbe juttatása és megjelenítése

Az alkalmazott E-IoT eszközhöz fejlesztett C++ könyvtár lehetővé teszi az egyes érzékelők adatainak kiolvasását és kiértékelését is.

A mért adatok alapján azonnal lehetőség van a BT2.0 modulon keresztül parancsok kiadására a helyi BT eszközök felé.

Az MA510 GSM modemhez fejlesztett kommunikációs könyvtár minden a modemmel és a GSM hálózattal kapcsolatos szükséges funkciót támogat, így - a teljesség igénye nélkül felsorolva - egyszerűen lekérdezhető a Modem és a SIM kártya azonosítója, az Endrich felhőszerver DNS adatai, képes a szerver és a modem között UDP-csatorna létesítésére, a küldendő JSON távirat felépítésére, kódolására és feladására, elküldésére is.

Az E-IOT felhőalapú adatbázisa az adatokat egységes formában kéri ahhoz, hogy a Visualization Gateway és az applikációs megjelenítést támogató API-k számára egységes adatstruktúra álljon rendelkezésre.

Az így felépítendő JSON telegram a következő szerkezetű:



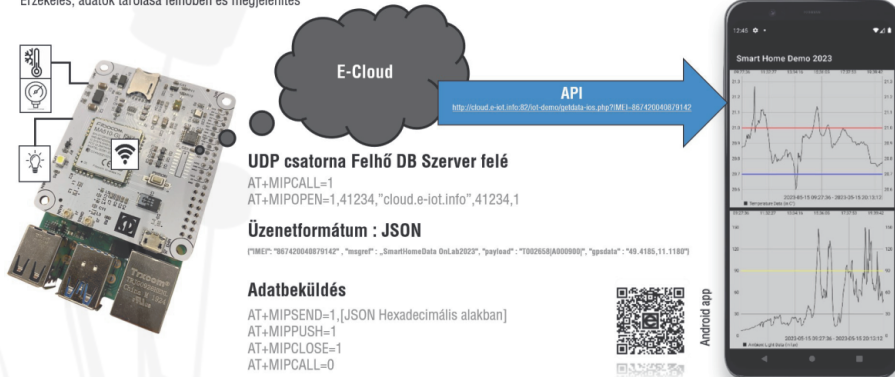
A létrehozott JSON távirat kódolásáról és annak feladásáról a C++ könyvtárfüggvények gondoskodnak, maguk a modem műveletek az MCU és a Modem UART adatkapcsolatán keresztül AT parancsok segítségével valósulnak meg.

Az E-Cloud relációs adatbázisába kerülő adatok hozzáféréséről az E-IoT rendszer többféleképpen gondoskodik.

A felhasználóknak lehetősége van – felhasználói bejelentkezés után – a WEB-es ügyfélporthálon kerülni letölténi, az E-Visualization Gateway Webes műszerfal segítségével megjeleníteni, vagy a rendelkezésre álló API-k segítségével saját applikációba integrálni. Ez utóbbi módszert használta gyakoronok kollégánk is a frissen kifejlesztett Android okostelefonos applikáció megvalósításához.

### Szenzoradatok felhőbe juttatása

Érzékelés, adatok tárolása felhőben és megjelenítés



#### UDP csatorna Felhő DB Szerver felé

AT+MIPCALL=1  
AT+MIPOPEN=1,41234,"cloud.e-iot.info",41234,1

#### Üzenetformátum : JSON

["IMEI": "867420040879142", "msgref": "...SmartHomeData OnLab2023", "payload": "T002658|A000900", "gpsdata": "49.4185,11.1180"]

#### Adatbeküldés

AT+MIPSEND=1,[JSON Hexadecimális alakban]  
AT+MIPPUSH=1  
AT+MIPCLOSE=1  
AT+MIPCALL=0

3| Az okosotthon mintarendszer távoli szerverrel (E-IoT Cloud-al) való kommunikációja

Az API meghívásakor a lekért eszköz adatai szintén JSON szerkezetben állnak rendelkezésre, melyek feldolgozásával grafikus megjelenítésre van lehetőség.

## Lokális okoseszköz-vezérlés emulációja Bluetooth-on keresztül

Az E-IOT SBC hőmérséklet és látható fény intenzitás szenzorai által érzékelt adatok alapján az MCU egyszerű döntéseket képes hozni az 1. ábrán vázolt rendszerben.

Ahhoz, hogy ezen egyszerű döntések alapján az otthoni okoseszközöket vezérelni tudjuk, valamilyen támogatott vezetékmentes lokális rádiós kapcsolat létesítésre van szükség, ilyen lehet a

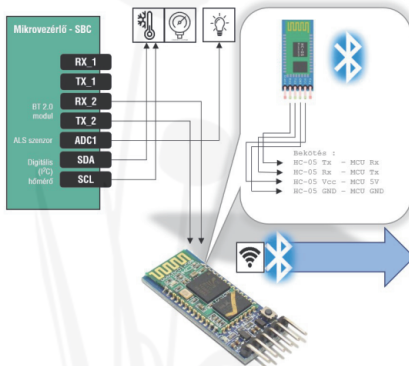
WiFi, vagy a Bluetooth technológia integrálása.

Mintarendszerünkben egy Bluetooth 2.0 modult használunk egy szabad UART Tx és Rx lábainak csatlakoztatásával a HC-05 Rx és Tx lábaihoz, ezek után már csak 5V tápfeszültséget kell biztosítani a modul számára és a soros porton való kommunikáció útján parancsokat küldhetünk.

Mivel általános koncepcionális megoldásban gondolkodunk, valós eszközök helyett egy mobiltelefont és egy azon futó Bluetooth terminált használunk a parancsok megjelenítésére, valamint az ott begépelte manuális adatok fogadására az E-IoT eszközön. A HC-05 modul és a mobiltelefon párosítása után

### Klíma- és redőnyvezérlés (BlueTooth) emulálása

Érzékelés, vezérlés – BT 2.0 kapcsolattal



4) Az okosotthon mintarendszer helyi Bluetooth alapú kommunikációja

az BT terminál emulátor elindítása után már használható is a rendszer.

## Az alkotókról

A cikksorozat befejezésével engedje meg az olvasó, hogy bemutassam a rendszer megalkotóit és beszéljek feladataikról.

Kocsis Csaba fejlesztőmérnök, a hardver rendszerek tervezője és megalkotója

Veresegyházy Zsolt okl. villamosmérnök, a Webes háttérrendszerek, az ECDB, az ügyfélkiszolgáló rendszer, a Visualization Gateway és az API-k fejlesztője

Kiss Bence. BME VIK harmadéves hallgató, gyakornok, az MA510 és szenzor C++ könyvtár fejlesztő, az Android App és a Bluetooth vezérlés-emulátor program készítője

Dankó Tibor, a hardverek gyártója és rengeteg jó ötlet adója

Jómagam pedig a HW&SW rendszerek alapjainak kidolgozásáért felelek, valamint a beágyazott szoftverek (C/C++) területén, és az IoS alapú programozásban (Swift) tevékenykedem.