

Kertészet klimatizációs rendszere

Tartalom

| | | |
|------|---|----|
| 1. | Bevezetés..... | 2 |
| 1.1. | Raspberry Pi 4..... | 2 |
| 1.2. | Sense HAT | 3 |
| 1.3. | Shelly 1 Wi-Fi-s okosrelé..... | 4 |
| 2. | A rendszer megvalósítása..... | 4 |
| 2.1. | Adatbázis | 10 |
| 2.2. | A hűtés megvalósítása..... | 7 |
| 2.3. | A fűtés megvalósítása..... | 7 |
| 2.4. | A levegő páratartalmának szabályozása | 8 |
| 2.5. | Üzenet küldése | 8 |
| 3. | Demo | 10 |

1. Bevezetés

Projektünk során egy képzeletbeli kertészet fóliasátrának a levegő hőmérsékletének, illetve a páratartalmának szabályozási rendszerét valósítottuk meg.

Megoldandó probléma: a fóliasátor temperálása – előre beállított minimálisnál alacsonyabb hőmérséklet esetén Shelly1 okosrelé kapcsolja a fűtés-t, ezen beállított érték +1Celsius (histerézis) elérése után lekapcsol. Előre beállított maximális hőmérséklet túllépése esetén a Shelly2 wifi-s okos relé bekapcsol, és elindítja a hűtőberendezést. Maximális hőmérséklet - 1Celsius (histerézis) elérése esetén a relé kikapcsol. Előre beállított páratartalom elérése esetén a Shelly 3 wifi-s okosrelé beindítja a szellőztető berendezést, amely a kívánt érték eléréséig működik.

Felhasznált anyagok:

- Raspberry Pi 4 8GB
- Sensehat
- 3db Shelly 1 wifi-s okosrelé
- A prezentációhoz szükséges: 12V-os tápegység, szerelődoboz, 3db Zöld LED, 3db 1kOhm-os ellenállás, sorkapocs, 1mm-es sodrott réz kábel,

1.1. Raspberry Pi 4

A Raspberry Pi a Raspberry Pi Foundation nevű jótékonyági szervezet által készített egylapos számítógépek sorozatának a neve, amelynek célja, hogy az embereket számítástechnikára oktassa, és megkönnyítse a számítástechnikai oktatáshoz való hozzáférést.

A Raspberry Pi már régóta az olcsó, hitelkártya méretű, ún. SoC (System on Chip) számítógépek etalonja, ami egyaránt alkalmas robotok, okosotthonok vezérlésétől különböző mérési feladatok ellátásán át szinte tetszőleges számítástechnikai feladat ellátására, amellett, hogy kiváló oktatási eszköz. A már régóta várt Raspberry Pi 4 egy következő szintre emelte a Pi-t, olyan teljesítménnyel, amely megfelelő ahhoz, hogy akár kisméretű asztali PC-ként használhassuk, kiegészítve azzal a képességgel, hogy 4K-s videókat 60 Hz-es sebességgel vagy kettős monitorral képes megjeleníteni.

Bármely verziójú Raspberry Pi egyik kiemelkedő lehetősége a 40 csatlakozóból álló GPIO (General Purpose Input / Output) port. A tűskék száma és elrendezése változatlan maradt a korábbi modellekhez képest, ami egységes a Raspberry Pi 2-es típus óta, így minden „HAT” panel, érzékelő vagy LCD kijelző, ami a Pi 2-höz vagy a 3-hoz csatlakoztatható volt, továbbra is kompatibilis marad.

A Raspberry Pi 4 azonban néhány új, bővített képességet adott a tűskesornak. Azoknak a fejlesztőknek, akik különféle új perifériákat terveznek fontos információ, hogy az új GPIO portkiosztás további három (azaz összesen négy) I2C, SPI és UART kapcsolatot tartalmaz. Tehát, ha érzékelői vagy perifériái megkövetelik ezen interfészek valamelyikét, akkor most nagymértékben bővültek a lehetőségek.

Az alábbiakban látható az új, kibővített GPIO tűskelosztás, külön jelölve a Pi 4-es verzió bővített lehetőségeit:



1.2. Sense HAT

A Sense HAT (Hardware Attached on Top) közvetlenül a Raspberry Pi tetejére illeszthető és a GPIO lábakon keresztül lesz összekötve a Pi-vel. A Sense HAT-ra különféle érzékelők lettek integrálva, amelyek alkalmasak kísérletek elvégzésére vagy tudományos kutatások folytatására (kertészet klimatizációs rendszer üzemeltetésére) de lehet akár játékokat is fejleszteni hozzá.

A következő szenzorok találhatók rajta:

- Gyorsulásmérő.
- 3D gyroscope.

(Pető-Sárai)

- Magnetométer.
- Hőmérsékletmérő.
- Légnedvességmérő.
- Légnomásmérő.

A Raspberry Pi és a Sense HAT összekapcsolása nagyon sok izgalmas lehetőséget nyújt. A szenzorok segítségével olyan helyeken is lehet kutatásokat végezni, ahova az ember nem, csak egy Raspberry fér be. Ott meg lehet mérni például, milyen gyorsan mozog a Pi (sebességmérés), mekkora a levegő hőmérséklete, páratartalma, mi a térbeli elhelyezkedése az eszköznek, stb.

A 8x8-as LED Mátrixon kijelvezhetők a különböző szenzorokról érkező adatok, egy iránytű programmal a mágneses északi irányt mutathatjuk vagy egyszerűen csak játszhatunk vele Tetrist, Pongot. Egy klassz kis joystick is helyet kapott a modulon. A joystickkal beavatkozhat a Raspberry PI Sense HAT futó programjaiba is. Egy izgalmas kiegészítő modul Raspberry Pi-hez, amely a Nemzetközi Űrállomás tudományos programjában is szerepel. Gyorsulásmérőjével meghatározhatók a térbeli mozgás paraméterei, ezentúl 3D gyroscope és magnetométer is található a modulon. A tudományos kísérletekhez elengedhetetlen hőmérséklet, páratartalom és légnyomás szenzorok is helyet kaptak rajta.

1.3. Shelly 1 Wi-Fi-s okosrelé

Ez az egyik legkisebb méretű, Wi-Fi-s eszköz kapcsolások okosvezérlésére. Apró mérete lehetővé teszi, hogy akár egy szabványos villanykapcsoló, konnektor mögé a kapcsolóaljzatba vagy egy villamossági kötődobozba beszereljük, akár utólag is.

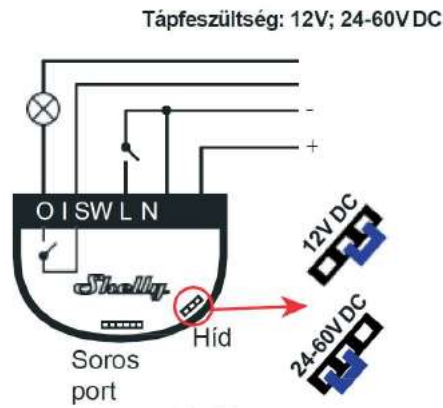
Meglévő kapcsolók vagy konnektorok mögé a fali aljzatba elhelyezett Shelly 1 segítségével a meglévő kapcsolóink, konnektoraink megtartásával tudjuk távvezérelni világításunkat vagy egyéb fogyasztókat, illetve követni ki/be kapcsolt státuszukat, így a termék ideális utólagos "okosításra" is. A Shelly 1-t elterjedten használják egy gombos villanykapcsolók és konnektorok mögé szerelve azok vezérlésére, 16A maximális teljesítménye konnektorok vezérlésére is alkalmassá teszi. A mi esetünkben a szellőztetést és fűtést fogjuk vezérelni vele.

2. A rendszer megvalósítása

Raspberry Pi4 – az eszközön Debian Linux operációs rendszer fut. A routerben MAC cím alapján foglaltunk IP címet, mely a következő lett: 192.168.1.102. Az összes komponens frissítése után a VNC szerver telepítés és engedélyezése következett, majd a Sensehat könyvtárt is külön installáltuk. Windows-os gépre a „PUTTY” alkalmazást telepítettük, melynek segítségével a Raspberry terminál ablakán keresztül telepítettük a Node-red platformot. Innentől a 192.168.1.102:1880-as porton keresztül a web browserünk segítségével tudtuk elkezdni a projektet. A dashboard a 192.168.1.102:1880/ui –n keresztül érhető el.

Shelly 1 wifi-s okosrelé: az eszközöket 12V tápellátással terveztük, ezért a jumpereket át kellett állítani, mivel azok alapértelmezetten 24-60DC feszültségre voltak jumper-relve.

A demonstrációhoz szükséges doboz, amely IP44 védettségű. A 12V-os tápellátást egy 18W-os tápegység biztosítja, amely a 3db LED meghajtásának elegendő. A relé bekötését a következő ábra mutatja:



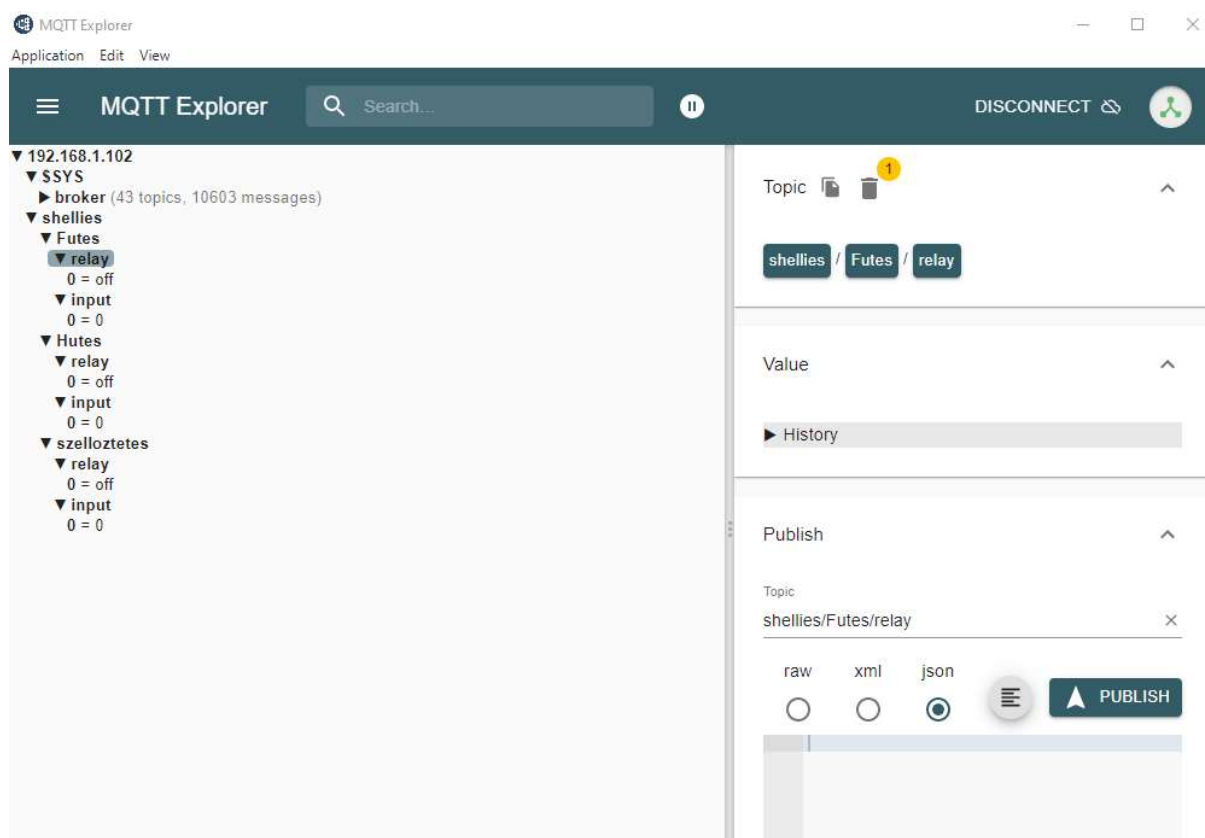
A 3 relét párhuzamosan kötöttünk. Mindegyik relé 1 áramkört képes vezérelni

A LED-eket a kapcsolóáramkörbe kötöttük, mindegyikkel sorosan 1-1db 1kOhm-os korlátozó ellenállást kapcsolunk.

A relék tápellátást követően alapértelmezetten saját host-ként üzemeltek. Felcsatlakoztunk az általuk szolgáltatott wifi-re, majd a 192.168.33.1-re belépve, módosítottuk a beállításait egyesével. Mindegyik a helyi wifi-re csatlakozik kliens módban. Statikus IP címük a következő:

- 192.168.1.105;
- 192.168.1.106;
- 192.168.1.107.

Az MQTT eszközök menedzselését az MQTT Explorer segítségével tudjuk elvégezni:



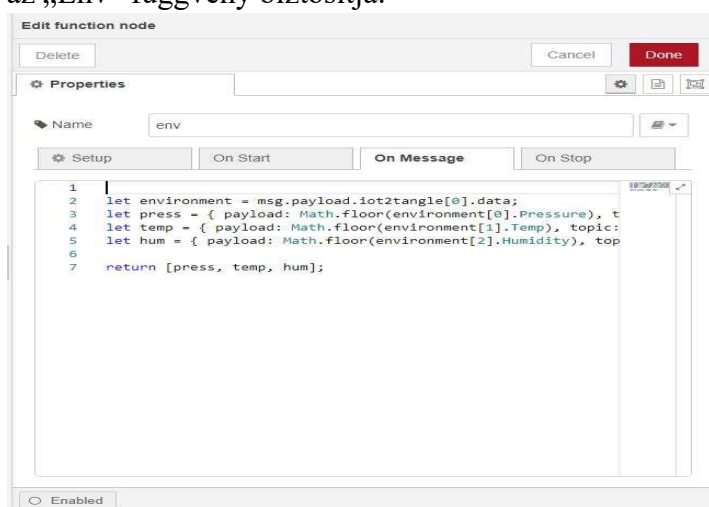
Végül az ADVANCED – DEVELOPER SETTINGS-ben bekapcsoltuk az MQTT protokollt. Szerverként a Raspberry ip címét adtuk meg, az 1883-as port számmal: 192.168.1.102:1883 és prefix-ként a futes, hutes, szelloztetes nevet adtuk nekik.

Így már készen álltak az eszközök a kommunikációra.

A mért adatokat a Raspberry Pi **Sense HAT** paneljének a hőmérséklet és páratartalom érzékelői szolgáltatják. Ezen panel további 4 érzékelője (légnyomás, gyorsulás, giroszkóp, és magnetométer) a projektünk során nem került felhasználásra.

A mért adatokat MQTT protokoll-on keresztül kapjuk meg. A tervezés során kommunikációs problémát a node-red és a Sense HAT között egy python script segítségével oldottuk meg. A script a data kapcsoló segítségével aktiválható a Dashboard-on.

A Sense HAT-ból érkező jeleket javascript nyelven írt kóddal tároljuk. Az eredményeket az „Env” függvény biztosítja.



A nyomás (Pressure) adatokat a projekt során nem használtuk fel.

A hőmérséklet (temp) és a páratartalom (hum) adatokat közvetlenül a Dashboardon megjelenítjük, Gauge kijelzőn, illetve grafikonon is. A grafikon X tengelyén 1 órás intervalumot jelenít meg. A hőmérséklet esetében a Z tengely 0-50 Celsius fok között, a páratartalom esetén pedig 0-100% között változhat.



A beérkező hőmérsékleti adatokat két irányba vezetjük. A fűtés, illetve a hűtés vezérléséhez.

2.1. A hűtés megvalósítása

A jel egy switch-be érkezik, amely eldönti, hogy a payload-ban beérkező érték, magasabb-e, mint a maxho változóban előre beállított érték. A mi esetünkben ez 40C. Amennyiben magasabb, az egyes kimenet engedi tovább. Különben a kettesen.

- Az egyes kimenet egy changenode-ra van kötve, amely beérkező jel esetén, a kimenetének a payload-jába on üzenetet küld, mellyek a shelly-nek a bekapcsolás parancs. Ugyanez a kimenetet egy másik changenode-ra is kötötök, mellyel a dashboardon a hűtés állapotot jelző LED-et kapcsolja zöldre az 1-es üzenet segítségével.
- A kettes kimenet szintén két különböző changenode-ra van kötve. az első, amely a beérkező jelet OFF üzenetre alakítja, a shelly okosrelének ad parancsot. Az így lekapcsolt. A másik changenode a dashboardon lévő hűtést jelző LED-nek küld 0-s üzenetet. Az így pirosra vált.

2.2. A fűtés megvalósítása

A jel egy switch-be érkezik, amely eldönti, hogy a payload-ban beérkező érték, alacsonyabb-e, mint a minho változóban előre beállított érték. A mi esetünkben ez 25C. Amennyiben alacsonyabb, az egyes kimenet engedi tovább. Különben a kettesen.

- Az egyes kimenet egy changenode-ra van kötve, amely beérkező jel esetén, a kimenetének a payload-jába on üzenetet küld, mellyek a shelly-nek a bekapcsolás parancs. Ugyanez a

kimenetet egy másik changenode-ra is kötöttük, mellyel a dashboardon a fűtés állapotot jelző LED-et kapcsolja zöld állapotúra a az 1-es üzenet segítségével.

- A kettes kimenet szintén két különböző changenode-ra van kötve. az első, amely a beérkező jelet OFF üzenetre alakítja, a shelly okosrelének ad parancsot. Az így lekapcsolt. A másik changenode a dashboardon lévő fűtést jelző LED-nek küld 0-s üzenetet. Az így pirosra vált.

2.3. A levegő páratartalmának szabályozása

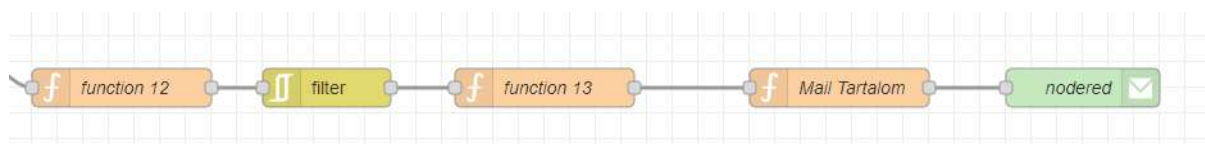
Páratartalom esetében csak a megnövekedett értékek esetében kell beavatkozni. Amennyiben 80% fölé emelkedik, a szellőztetőrendszert elindítjuk. A szoftveres része hasonló képpen működik, mint a hőmérséklet setében. Az ideális páratartalmat „para” változó néven tároljuk. A sensor-ból érkező jel egy switch-be érkezik, ami eldönti, hogy az magasabb-e, mint a meghatározott maximum. Amennyiben magasabb, akkor az 1. kimeneten engedi tovább a jelet, ha kisebb, akkor a másodikon.

Az 1. kimenetet két külön changenode-ra van kötve. 1. „paratartalom-shelly-be” az on üzenetet küldi a „szellozteto shelly”-nek, ami így bekapcsol. A másik „Szellozteto-LED-be” changenod a beérkező jelet 1 értékű számmá változtatja, és a dashboardon található szellőztetést jelző LED-et zöldre állítja.

Kisebb páratartalom esetén a második kimeneten küldi a jelet. Amely szintén két changenode-ra van vezetve. Az egy a „paratartalom-shelly-ki”, ami az OFF üzenetet küldi tovább a relének, ami így kikapcsol. A másik a „Szellozteto-LED-ki”, ami a jelet 0-ra állítja és a dashboardon a szelloztetés LED-et pirosra állítja.

2.4. Üzenet küldése

A rendszer figyelmeztetést küld – email formájában – a felhasználónak amennyiben a páratartalom túl magas. Az üzenetet a Gmail rendszerén keresztül valósítjuk meg.



Function tartalma:

```
let allapot = false;
if(msg.payload >= 85){
  allapot = true;
}else{
  allapot = false;
}
msg.payload = allapot;
return msg;
```

Alapból a „Function 12”-t, false értékre állítjuk. Abban az esetben, ha 85 feletti értéket kap, akkor átvált true-ra, különben marad false.

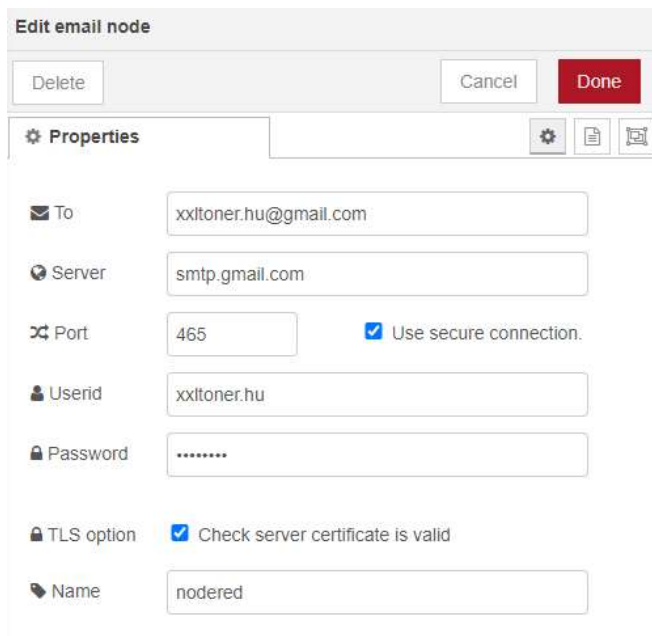
A kimenetére adja az aktuális állapot értékét.

„Filter” (szűrő) segítségével tudjuk elérni, hogy csak akkor kapjunk üzeneteket, amikor történik változás. Így tudjuk kiküszöbölni a sorozatos email-ek érkezését.

A „function 13” gyakorlatilag egy típuskonverziót hajt végre. True érték esetén 1-es üzenetet küld:

```
if(msg.payload == true){  
  msg.payload = 1;  
}else{  
  return  
}  
  
return msg;
```

A „function 13” a „Mail tartalom” blokkjának a tartalmát küldi a „nodered” nevű email blokkra, ami magát az emailt küldi.



The screenshot shows the 'Edit email node' configuration window. At the top, there are three buttons: 'Delete', 'Cancel', and 'Done'. Below this is a 'Properties' section with several fields and checkboxes. The fields are: 'To' (xxltoner.hu@gmail.com), 'Server' (smtp.gmail.com), 'Port' (465), 'Userid' (xxltoner.hu), 'Password' (masked with dots), and 'Name' (nodered). There are also checkboxes for 'Use secure connection.' and 'Check server certificate is valid', both of which are checked.

A Gmail beállítása közben szembesültünk a kétlépcsős azonosítás problémájával. A kétlépcsős azonosítást az email node nem támogatja amit az „Alkalmazásjelszavak” Google fiók beállítással sikerült orvosolni:

← Alkalmazásjelszavak

Az alkalmazásjelszavak segítségével olyan eszközök alkalmazásaiban is beléphet Google-fiókjába, amelyek nem támogatják a kétlépcsős azonosítást. Csak egyszer kell majd megadnia az alkalmazásjelszót, így nem kell rá emlékeznie. [További információ](#)

Saját alkalmazásjelszavak

| Név | Létrehozva | Utolsó használat ideje | |
|---------|------------|------------------------|--|
| NodeRed | dec. 15. | 9:20 | |

Válassza ki azt az alkalmazást és eszközt, amelyhez alkalmazásjelszót szeretne generálni.

Válassza ki az alkalmazást ▼ Válassza ki az eszközt ▼

LÉTREHOZÁS

A beállítások során elég egyszer megadni a felhasználói jelszót és a rendszer biztosítja az email küldés lehetőségét a Gmail smtp (smtp.gmail.com) szerverén keresztül úgy, hogy nincs szükség kétlépcsős azonosításra.

2.5. Adatbázis

A Sense HAT által mért értékeket egy adatbázisban tároljuk, melynek neve „SensehatDB”.



A tárolás érdekében létrehoztunk egy „Envinsert” nevű node-t, mely a Sense HAT-ből érkező adatokat az adatbázisba menti:

```
let temp, hum;
let tempData = msg.payload.iot2tangle[0].data[1].Temp;
let humData = msg.payload.iot2tangle[0].data[2].Humidity;
/*
node.warn(pressureData);
node.warn(tempData);
node.warn(humData);
*/
let sqlTemp = "INSERT INTO temp (data) VALUES ('"+tempData+"')";
let sqlHum = "INSERT INTO hum (data) VALUES ('"+humData+"')";
```

3. Demo

Mivel a valóságban nehezen tudunk gyorsan változó hőmérsékletet, és páratartalmat előidézni, ezért dashboard-ra elhelyeztünk két csúszkát, amellyel tudjuk a rendszerbe küldeni a valós értékekhez hasonlókat, és meg tudjuk figyelni a rendszer viselkedését.

A dashboard-ra még elhelyeztünk két kapcsolót, melynek a segítségével a Sense HAT 8x8-as LED matrix-ra ki tudjuk futófény formájában iratni az aktuális hőmérsékletet.



(Pető-Sárai)