



Programozható elektronikák alkalmazásai

Készítette

Bagoly Gábor

programtervező informatikus

Témavezető

Dr. Geda Gábor

egyetemi docens

EGER, 2022

Tartalomjegyzék

1. Bevezetés	3
1.1. Az Arduino platform	3
1.2. Arduino UNO felépítése és tulajdonságai	3
2. IoT eszközök alkalmazása okos otthonban	6
2.1. IoT eszközök	6
2.2. Okos otthon	8
2.2.1. Okos otthon kiépítése	9
Irodalomjegyzék	11

1. fejezet

Bevezetés

A választott szakdolgozati témám a *Programozható elektronikák alkalmazásai*, azon belül a témám egy okos otthon tervezése, különböző okos eszközök irányításával egy lokális szerveren úgy, hogy azt egy *Raspberry Pi*, vagy *NodeMCU* szolgáltassa, és irányítaná az eszközöket. Az eszközökkel például világítást, fűtést is lehet vezérelni, akár automatikusan. Az eszközök különböző féle programozható elektronikák segítségével lesznek megvalósítva. Céлом az lenne ezzel, hogy belelássak az okos otthonok működésébe, különböző programozható eszközök alkalmazását jobban megismerjem, és, hogy egy olyan általános kezelőfelületet tudjak létrehozni, amit könnyen tud a felhasználó alkalmazni. Az okos otthon részről pedig majd a 2. fejezetnek 6. oldalától lehet majd többet megtudni, ahol megmutatom azt is, hogy C++ programozási nyelvet használva hogyan olvasom be és írom ki egy hőmérséklet és páratartalom szenzor által mért adatokat.

1.1. Az Arduino platform

Az Arduino egy széleskörűen elterjedt, nyílt forrású fejlesztőplatform. Az Arduino-t oktatási céllal hozták létre, de előszeretettel használják otthoni projektekhez, a kisebb automatizálási feladatoktól kezdve az okos otthonok kialakításáig. Ezek mellett ma már az ipari alkalmazások sem ritkák és rengeteg IoT¹ eszköznek is az alapja. Az Arduino platform része egy elektronikai áramkörü lap és a szoftveres környezet. Maga az áramkörü lap nagyon sokféle felépítésben megtalálható.

1.2. Arduino UNO felépítése és tulajdonságai

Az egyik legelső és leginkább elterjedt áramkör az *Arduino UNO*.

¹ A 2.1 szakaszon van kifejtve.



1.1. ábra. Az *Arduino UNO* áramkörtábla az egyik legelterjedtebb a sokféle kártya közül.

Az Arduino áramkörök leglényegesebb alkatrésze a mikrovezérlő. Az *UNO* esetében ez egy *ATmega328/P* típusú, 8 bites mikrovezérlő. A mikrovezérlő egy integrált áramkör (IC, chip), amiben a processzor és a memória mellett számos egyéb periféria is megtalálható. A legfontosabb perifériák egy általános mikrovezérlőben a következők lehetnek: időzítő áramkörök, analóg és digitális be- és kimenetek, kommunikációs perifériák és még sok egyéb. Ilyen mikrovezérlőkből egyre több található meg a mindennapi körülvéve hétköznapi használati tárgyainkban is, háztartási kis- és nagy gépekben például: mosógépben, digitális órában, távirányítóban, mobiltelefonban. Segítségükkel mérhetjük szenzorok jelét, beolvashatjuk nyomógombok állapotát, vezérelhetünk velük LED-eket, motorokat, kommunikálhatunk más áramkörökkel és még sok egyéb feladatra használhatók.

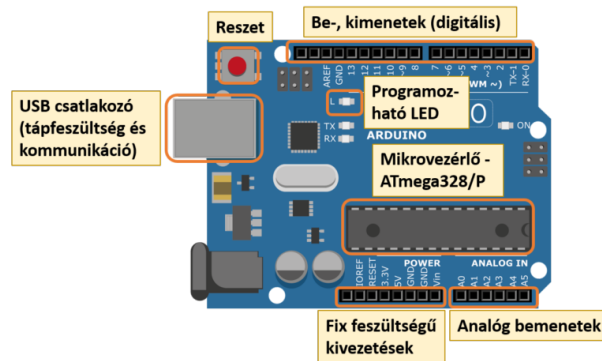
Az *Arduino UNO* áramkörtábla szerepe, hogy a rajta elhelyezett mikrovezérlőnek a lábai ki vannak vezetve egy kényelmesen hozzáférhető és könnyen használható csatlakozókra, valamint megtalálható rajta a mikrovezérlő működéséhez és programozásához szükséges egyéb alkatrészek.

A mikrovezérlők általában lényegesen kisebb számítási teljesítménnyel rendelkeznek egy hagyományos PC-hez képest, de ezáltal sokkal alacsonyabb a fogyasztásuk és méretük is jelentősen kisebb lehet. Mivel általában nem fut rajtuk operációs rendszer, ezért a rendelkezésre álló erőforrásnak megfelelő valós-idejű feladatokra is alkalmazhatók, azaz egy adott eseményre a mikrovezérlő gyorsan és egy garantált maximális idő alatt képes reagálni.² Az *ATmega328/P* típusú mikrovezérlő lényegesebb paraméterei:

- 8 bites architektúra
- 16 MHz-es órajel
- EEPROM (csak olvasható memória): 1 KB
- SRAM (változók tárolására): 2 KB

² Léteznek valós idejű operációs rendszerek, azonban ezek több erőforrást is igényelnek, mint a natív kód.

- A flash (programot tároló ROM): 32 KB



1.2. ábra. Az *Arduino UNO* áramköri lapon lévő legfontosabb alkatrészek

Ahhoz, hogy külső áramköröket is használjunk, tudnunk kell a lábkiosztást, azaz, hogy a kódból hogyan tudunk hivatkozni az adott kivezetésre. A *Arduino* áramköri lap két szélén lévő tűsoros aljzat melletti számokkal érjük el az adott kivezetést a kódból. Ezek a kivezetések lehetnek digitálisak (0-tól 13-ig jelölve) vagy analóg bemenetek (A0-tól A5-ig jelölve). Az A0-A5 kivezetések még digitális be- és kimenetek is lehetnek, ha a kódban úgy állítjuk be az adott kivezetést. Külső áramkörök esetén a fix feszültségű kivezetésekről kényelmesen tudunk tápfeszültséget adni az áramkör számára.

A csatlakozókhoz kényelmesen köthetünk szenzorokat, digitális IC-eket és egyéb külső áramköröket. Ahhoz, hogy ezeket biztonságosan tudjuk használni, előbb meg kell ismernünk a be- és kimenetek működésének korlátait. A témához szükséges ismerni a következő, legfontosabb szinteket és korlátokat:

- Működési feszültség: 5 V
- Maximális kivehető/befolyatható áram egy digitális kimenetből: 20 mA
- Maximális kivehető áram a 3,3 V-os kimenetből: 50 mA

Az feszültségszintekre és áramkorlátokra nagyon figyelni kell az *Arduino* használata során. Csak egyforma feszültségszinteken üzemelő áramköröket szabad összekapcsolni, az áramerősségeket pedig a számolás során kapott értéknek megfelelő, sorba kapcsolt ellenállással lehet a maximális megengedett érték alatt tartani.

Az *Arduino*-hoz számos „feltét” áramkör is kapható, amiket *Arduino* shield-eknek neveznek. Ezeknek az áramköri lapoknak az az előnye, hogy egyszerűen, egyetlen modulattal rá lehet helyezni az *Arduino*-ra és máris kiegészítettük az *Arduino*-t valamilyen hasznos funkcióval. A kompatibilitásra figyelni kell, mert léteznek 3,3 V tápfeszültséget igénylő shield-ek, amik tönkremehetnek az 5 V-os jelszintet használó *Arduino UNO*-tól. További *Arduino* eszközről az alábbi könyvben tudhat meg többet.[1, 8–12.oldal]

2. fejezet

IoT eszközök alkalmazása okos otthonban

2.1. IoT eszközök

Manapság számos kis és nagy háztartási készülék és gép érhető el okos kivitelben. Ezek úgynevezett IoT (Internet of Things) eszközök, amelyek az otthoni WiFi hálózaton keresztül egyszerűen párosíthatóak az adott gyártói applikációval. Erre azért van szükség, hogy a lehető legtöbb okos funkciót ki tudjuk használni, illetve esetleges firmware frissítéseket el tudjunk végezni. Ezzel a módszerrel azonban szinte minden márkához és eszközhöz külön applikációt kell használni, ezáltal egyfajta szigetüzem jön létre.

Manapság már beszerezhető például okos WiFi-s mosógép, ahol külön alkalmazáson keresztül nyomon követhetjük, mikor jár le a mosás. De automatizációval előre meghatározott időpontban is elindíthatjuk a mosógépet. Ha nem vagyunk otthon a vízérzékelők figyelik, hogy nincs-e probléma, és szükség esetén gondoskodnak a víz elzárásáról.

IoT eszközök beszerzésével tehát elérjük, hogy a különböző gépek wifin vagy más protokollokon keresztül képesek egymással kommunikálni. De ha nem időben gondolkodunk rendszerben, akkor könnyen eljuthatunk oda, hogy minden gyártóhoz egy külön alkalmazást rendelünk, ami egyre bonyolultabbá teszi az egyes eszközök kezelhetőségét és ellehetetleníti az automatizációt. IoT eszközök közé tartozik az 1.2. szakaszban említett *Arduino UNO* is.

Ellentétben a „kész” rendszerekkel vagy DIY¹ megoldásokkal az egész háztartást összeköthetjük egy rendszer égisze alatt. Az okos otthon megalkotása során elindulhatunk nagy gyártók² szisztémái mentén, de nyílt forráskódú rendszerek mellett is dönthetünk.

Az otthonok automatizálására egyszerű “barkácskörülmények” között is lehetőség

¹ Do It Yourself, azaz magyarul Csináld Magad módszer.

² Az alábbiak például: Chameleon Smart Home, Zipato vagy Loxone.

van, így egyre többen használnak Raspberry Pi-t például FHEM, Domoticz vagy Node RED szoftverrel, amihez külön szerzik be az egyes érzékelőket, reléket és a különböző bővítő modulokat, a kontrollálni kívánt eszközöket, a vezérlő szoftvert, illetve a helyi vagy távoli vezérlőegységet. A beszerzés után ezeket manuálisan kell egy rendszerbe kötni és beprogramozni az automatizációkat.

Az IoT eszközök a felhasználási terület alapján az módon kategorizálhatóak:

– Industrial Internet Of Things³

- Az eszközök alkalmazása ipari, szállítási, energetikai, vagy egészségügyi területen.
- Az adatok mennyisége és sebessége a tartóstól a viszonylag magasig terjed.
- Az alkalmazások biztonság-kritikusak, például egy intelligens közlekedési rendszer hibás működése veszélyeztetheti az emberek életét.
- Az IIoT alkalmazások rendszerközpontúak.

– Consumer Internet Of Things⁴

- Fogyasztói eszközök, például mobil, hűtőszekrény, szemüveg stb.
- Az adatok mennyisége és sebessége viszonylag alacsony.
- Az alkalmazások nem túl kritikusak, például a fitness modulok meghibásodása nem okoz kárt.
- A CIoT alkalmazások fogyasztóközpontúak.
- CIoT még ezen belül az alábbi kategóriákra bontható:
 - Személyes IoT.
 - Csoportos IoT
 - Közösségi IoT

Programkód 2.1. Hőmérséklet és páratartalom szenzor által mért értékek mérése és kiírása

```
1 #include "DHT.h"
2 #define DHTPIN 23
3 #define DHITYPE DHT22
4
5 DHT dht(DHTPIN, DHITYPE);
6
7 void setup() {
8     Serial.begin(9600);
```

³ Magyarul: Ipari Dolgok Internetje.

⁴ Magyarul: Fogyasztói Dolgok Internetje.

```

9   dht.begin();
10  }
11
12  void loop() {
13      delay(2000);
14
15      float h = dht.readHumidity();
16      float t = dht.readTemperature();
17
18      if (isnan(h) || isnan(t)) {
19          Serial.println("Failed to read from DHT sensor!");
20          return;
21      }
22
23      Serial.print("Humidity: ");
24      Serial.print(h);
25      Serial.print(" %\t");
26      Serial.print("Temperature: ");
27      Serial.print(t);
28      Serial.println(" *C ");
29
30  }

```

2.2. Okos otthon

Egy tudományos cikk alapján az okos otthon az alábbi módon definiálható:

2.1. Definíció. „az okos otthon egy olyan (a lakhelyünkön kiépített) platform, ami IoT-s, kommunikációs és különböző számítógépes technológiákat felhasználva eszközöket köt össze egymással egy hálózaton keresztül úgy, hogy ezek automatizálhatóvá és könnyebben irányíthatóvá és váljanak, ezzel javítva az egész rendszer (ház) kezelhetőségét, kényelmét és biztonságát.”[2]

Sok ember megpróbált részletesebb kritériumot is adni az okos otthonokra. MICHAEL C. MOZER szerint egy épület csak akkor tekinthető intelligensnek, ha képes irányítani a világítást, a fűtést, hűtést és a redőnyöket ⁵, habár ez a megközelítés pontatlan. Szerintem nem kell ennyire szigorúan megszabni, a ház mely részeit kell irányítania egy ilyen rendszernek, csak arra kell törekednie, hogy a felhasználó életét megkönnyítse és segítse anélkül, hogy zavaróvá válna.

⁵ Továbbiakban olvashat erről MICHAEL C. MOZER: Lessons from an Adaptive Home című írásában.[3]

2.2.1. Okos otthon kiépítése

Az okosotthon⁶ kiépítése során az első és legfontosabb kérdés, hogy a különböző protokollon kommunikáló eszközöket hogyan fogjuk össze és vezéreljük. Ehhez meg kell vizsgálni, hogy az okoseszközök milyen kompatibilitással rendelkeznek.

A nyílt forráskódú, otthonautomatizálási platform, az OpenHab segítségével számos eszközt és rendszert integrálhatunk. Ezáltal létrejön az egységes felhasználói felület, viszont mindez időigényes folyamat, amely komoly precizitást és hozzáértést igényel. A lépéseket akár az interneten keresztül is nyomon követhetjük, YouTube videók által is megtanulható, konfigurálható.

A Home Assistant open-source⁷ szoftver szintén népszerű Magyarországon, ami a vezérlés mellett az adatvédelmet is előtérbe helyezi. Sok angol és több magyar nyelvű fórumon tájékozódhatunk, kérdezhetünk másoktól, amire általában szükség is van, mert könnyű elakadni a beállítási lehetőségek rengetegében.

Az okosotthon házilag való telepítése ugyanakkor kockázatokkal is jár. Amennyiben problémák lépnek fel, megfelelő szaktudás nélkül nehezebb helyrehozni azokat, például szoftverfrissítések esetén vagy az eszközök meghibásodásánál. Gondoljunk csak bele, hogy egy rendszerhibánál működésképtelenné válik az otthonunk. Ha csak a világítás-vezérlés nem működik a probléma orvoslásáig, talán nem okoz életbevágó problémát, de a fűtésvezérlés egy hideg téli estén már annál inkább.

⁶ Melyet a 2.1. definíció kifejt.

⁷ Magyarul: nyílt forrású.

Összegzés

A (2.1) képleten pedig látható az, hogy a \LaTeX mennyire alkalmas matematikai képletek megjelenítésére.

$$g: \mathbb{R} \setminus \left\{ \frac{k\pi}{2} : k \in \mathbb{Z} \right\} \rightarrow \mathbb{R}, \quad g(x) := \begin{cases} \frac{\arctg^2(x+\pi)}{\sin(2x)}, & \text{ha } x \geq \frac{2}{3}, \\ \cos^4(5x), & \text{különben.} \end{cases} \quad (2.1)$$

Irodalomjegyzék

- [1] JEREMY BLUM: *Exploring Arduino: Tools and Techniques for Engineering Wizardry*, 2019.
- [2] SMART HOME: *Architecture, Technologies and Systems*, URL: <https://tinyurl.com/smarthomeBook>
- [3] MICHAEL C. MOZER: *Lessons from an Adaptive Home*, University of Colorado, URL: <https://tinyurl.com/MozerBook>