IOT ESZKÖZÖK ÉS EGY WAGO PFC200 KONTROLLER KOMMUNIKÁCIÓJÁNAK MEGVALÓSÍTÁSA

Simon Róbert[1], Trohák Attila[2]
[1]PhD hallgató, [2]egyetemi docens
[1,2]Automatizálási és Infokommunikációs Intézet, Gépészmérnöki és Informatikai Kar
[1]robert.simon@uni-miskolc.hu, [2]attila.trohak@uni-miskolc.hu

ABSZTRAKT

Az IoT eszközök és a Programozható Logikai Vezérlők (PLC-k) közötti kommunikáció olyan esetekben válik fontossá, amikor két, eltérő automatizálási megoldást akarunk ötvözni. Az IoT eszközök széles körű érzékelőkkel és adatgyűjtőkkel rendelkeznek, amelyek lehetővé teszik a környezeti változók monitorozását és a gyűjtött adatok feldolgozását. Ezek az adatok azonban csak akkor válnak hasznos információvá, ha a PLC-k képesek ezeket értelmezni és reagálni rájuk. A kommunikáció ezen eszközök között különböző protokollokon keresztül zajlik, mint például az MQTT (Message Queuing Telemetry Transport), az OPC UA (Open Platform Communications Unified Architecture) vagy a Modbus. Ezek a protokollok lehetővé teszik az adatok megbízható és hatékony átvitelét az IoT eszközök és a PLC-k között. Az integráció előnyei közé tartozik a folyamatos adatgyűjtés és monitorozás, ami lehetővé teszi a hatékonyabb döntéshozatalt és a trendelt értékek extrapolációját. Például egy gyártósor az IoT érzékelők által gyűjtött adatokat közvetlenül továbbíthatja a PLC-nek, amely így azonnal reagálhat a változásokra, optimalizálva a gyártási folyamatot és minimalizálva a leállási időket. Vagy egy nagy méretű létesítmény számos pontján helyezhetők el érzékelők, amelyek az épület környezeti paramétereit rögzítik és továbbítják azt a PLC felé, amely így képes a beavatkozóknak, a pillanatnyi állapotnak megfelelő parancsokat adni. Emellett az IoT eszközök és PLC-k közötti kommunikáció lehetővé teszi a távoli hozzáférést és irányítást, ami kritikus lehet bizonyos ipari környezetekben vagy ahol a távoli eléréssel hatékonyabbá válik a rendszer kezelése és karbantartása. Kutatásomban egy családi házat látok el érzékelőkkel, amelyek adatait egy WAGO PFC200-as PLC fogja feldolgozni.

1. BEVEZETÉS

A megvalósítás helyszíne sokban meghatározta a kutatás részleteit és megvalósíthatóságát, ezért a kutatás céljának meghatározása után mindenekelőtt annak kivitelezhetőségét és egyedi megoldás-igényét kellett megvizsgálnunk. Ilyen megoldás-igény a megvalósítás helyszíne, kiépítettsége, térbeli tagoltsága, bonthatósága/építhetősége. Ezen felül meg kell vizsgálni a kiépített kommunikációs médiumokat és az esetlegesen meglévő okos otthon-rendszereket, amelyekbe az újonnan telepítendő eszközök integrálhatók kell legyenek. Ez azért fontos, mert ezen ismeretek fényében szűkül a felhasználható eszközök köre, és adott esetben módosul az elképzelt projekt is.

1.1 A kutatás célja

A kutatás célja olyan, új IoT eszközök használata egy, alapvetően nem okos-otthonnak épített házban, amelynek már elkezdődött az okosítása. Így a házban van már meglévő vezérlő – egy, az 1. ábrán bemutatott, WAGO PFC200-as típusú PLC –, amely a ház redőnyeit és bizonyos világítótesteket vezérel.



1. ábra
WAGO PFC 200

Az új IoT eszközök feladata elsődlegesen a ház fűtésének és újabb lámpa sorok vezérlésének megteremtése lesz. Ezzel együtt pedig egy olyan kommunikációs csatornának a kutatását is célul tűztem ki, amely két, egymástól jelentősen eltérő felhasználási területre tervezett eszköz közötti adatátvitelt képes megvalósítani. Ez azért fontos, mert a két eltérő terület általában más típusú terepi buszt, és kommunikációs protokollt alkalmaz. A PLC-k felhasználási területe elsődlegesen az ipari környezet, ahol jellemzően az erős külső behatások és zavarok miatt robosztus felépítés és szabványos – ám, az otthoni felhasználástól távol álló – kommunikációs protokollok segítik a biztos jelátvitelt, illetve a stabil működést. Az IoT eszközök elsődleges felhasználási területe a lakossági, otthoni környezet, amely az ipari környezet durva behatásai helyett, egy tiszta és zavarmentes környezetet jelent, ahol egyúttal az esztétikum is fontos szerephez jut. A robosztus kivitelezés nélkülözésével a méret jelentősen csökkenthető, és az energiafelhasználás optimalizálása is szemponttá válik. Kommunikáció tekintetében az IoT eszközök az iparban használt, vezetékes, terepi buszok helyett elsődlegesen vezetéknélküli adatátvitelt alkalmaznak.

2. A HELYSZÍN SAJÁTOSSÁGAI ÉS KIHÍVÁSAI

A kutatás helyszínéül egy jelenleg is lakott családi házat választottam, amely okosítása már elkezdődött ugyan, de még számos lehetőséget tartogat magában. Ennek első lépcsője a fűtés okosítása, mivel arra a korábbi kivitelezések hiányosságai miatt egyébként is szükség volt. A családi ház szerkezeti kialakítása kétszintes, hőforrása

gázkazán, fűtése tisztán padlófűtéses. A házat szintenként 4-4 fűtőkörrel látták el, két osztó-gyűjtőn keresztül. A hőforrás által előállított melegvizet a kazán saját, beépített szivattyúja volt hivatott mind a 8 fűtéskörre eljuttatni, ám ezt meglehetősen rossz hatásfokkal volt csupán képes megvalósítani, hiszen a rendszer rendkívül lassan reagált a fűtésigény megjelenésére – növelve a hőmérséklet volatilitását –, illetve a hőeloszlás sem volt egyenletes, akár 6-7 fokos eltérést is mutattak az egyes helyiségek. A problémát ráadásul tovább fokozta az, hogy csupán egyetlen termosztát volt az épületben, amely a nappali lakrészben volt elhelyezve. A ház maga egyébiránt rendkívül jó energiafelhasználású épület, hiszen egy téli, napsütéses napon a nappali akár 26 fokra is felmelegedhet, miközben kint jóval 0 fok alatti hőmérséklet van. Az emelet viszont kevesebb ablakkal rendelkezik, így annak felmelegedése sokkal kisebb mértékű. Ez a két szint hőmérsékleti diagramja közötti ollót tovább nyitotta.

Az első tél során megfogalmazódott tehát a lakók fejében a felújítás gondolata, amely második évre meg is valósult. Ennek során a szintek külön-külön kaptak megfelelő teljesítményű szivattyúkat, és beépítésre került egy hidraulikus váltó is, amely a kazán védőköreként és a fűtési rendszer osztóköreként is szerepet játszik. Természetesen a két külön fűtéskör (szint) külön hőmérsékleti szabályozást is igényelt, amely szabályozás megvalósítását már nem a szobai termosztát, hanem a PLC látja el.

2.1 A kivitelezés kihívásai

A vezetéknélküli szabályozás mindig tartogat magában veszélyt, hiszen a szabályozott jelről érkező visszacsatolás nem jut el teljes biztonsággal a szabályozó körhöz. Ezért elsődlegesen vezetékes megoldáson gondolkoztam, ám ennek kivitelezhetősége – a ház lakottsága okán – nehézkesen volt megvalósítható jelentős rombolás és koszképződés nélkül. Így végül maradt a vezetéknélküli kommunikáció, amelyet kiegészítettem egy vezetékes hőmérséklettávadóval, így jelvesztés esetén még mindig kapunk információt a ház állapotáról.

A ház automatizálás fejlesztésének második lépése a világításvezérlés kiterjesztése volt további fényforrásokra is. Ehhez már mindenképpen vezetéknélküli eszközökre volt szükség, hiszen ez összehasonlíthatatlanul nagyobb bontással járt volna a fűtésszabályozás fejlesztésénél és már az sem volt elfogadható mértékű. Ez utóbbi esetben azonban a vezetéknélküli kommunikáció determinisztikus voltának hiánya nem jelent problémát, hiszen legrosszabb esetben a fényforrás nem kapcsol be, vagy nem kapcsol ki, amely bár kényelmetlen ugyan, de sem jelentős veszéllyel, sem jelentős többletköltséggel nem jár.

Egy lakott ház automatizálási fejlesztése azonban egyéb kihívásokat is tartogat magában, hiszen a fejlesztés során – a már automatizált eszközök – időlegesen működésképtelenné válhatnak. Egyúttal fel kell készülni biztonsági mentésekkel is, hiszen egy sikertelen, vagy nem megfelelően működő fejlesztés után vissza kell állni az eredeti állapotra, hogy a ház funkciói újra teljes értékűek legyenek.

3. AZ ALKALMAZOTT SZENZOROK ÉS AKTUÁTOROK

A lakossági célra készült, vezetéknélküli hőmérséklettávadókat célszerűen az IoT eszközök távadói közül érdemes kiválasztani. Előnyük ezen eszközöknek az alacsony energiafelhasználás mellett a stabil működés és az esztétikus megjelenés, amely fontos szempont a lakókörnyezeti alkalmazásnál. Ám sajátosságuk is, hogy az egyes IoT

eszközök általában gyártónként eltérően, jellemzően önmagukkal, a saját rendszerükkel kompatibilisek. A lakossági körben használt IoT eszközök jellemzően ZigBee, ZWave, Bluetooth (BLE) és WiFi protokollokat használnak, amelyek közül egyik sem támogatott a házban használt PLC-vel. Ezért olyan eszközt kellett keresni, amely által szolgáltatott adatok a saját rendszerén kívülről is elérhetők, működésük kívülről is vezérelhető, és képes olyan protokollt használni, amely közvetve – akár egy gateway-en, hub-on keresztül – képes a PLC által is feldolgozható módon továbbítani a mért paramétereket.

Ezért választásom végül a Shelly termékeire esett, mert WiFi-t használ, így a ház meglévő vezetéknélküli rendszere használható, és nincs szükség újabb hub-okra, esetleg gateway-ekre, amelyek képesek a távadók jelét transzformálni. Ezen felül mindegyik eszközük képes MQTT-t használva adatokat továbbítani, amely olyan univerzális üzenetbróker, amelyet natív támogat a PFC200-as PLC is.

A 2. ábrán látható, Shelly Plus H&T hőmérséklet távadó és páraérzékelő elemes és USB-C tápforrásokkal képes üzemelni, utóbbi alkalmazása esetén 5 percenként, elemes üzem esetén 2 óránként vagy 0,2°C vagy 3%-os páraváltozás esetén küld jelet. Az érzékelő pontossága hőmérsékletmérés esetén 1 tizedesjegy.



2. ábra
Shelly Plus H&T

A 3. ábrán látható, Shelly 1L egy olyan okosrelé, amely két kapcsolót képes kezelni egyetlen kimenettel, illetőleg képes nulla vezeték nélkül, pusztán csak fázissal működni. Ebben az eseteben a relé az output ágat használja nulla vezetékként, amely LED-es áramforrások esetén vibrálást, és sztochasztikus villanásokat idéz elő kikapcsolt állapotban.



3. ábra
Shelly 1L

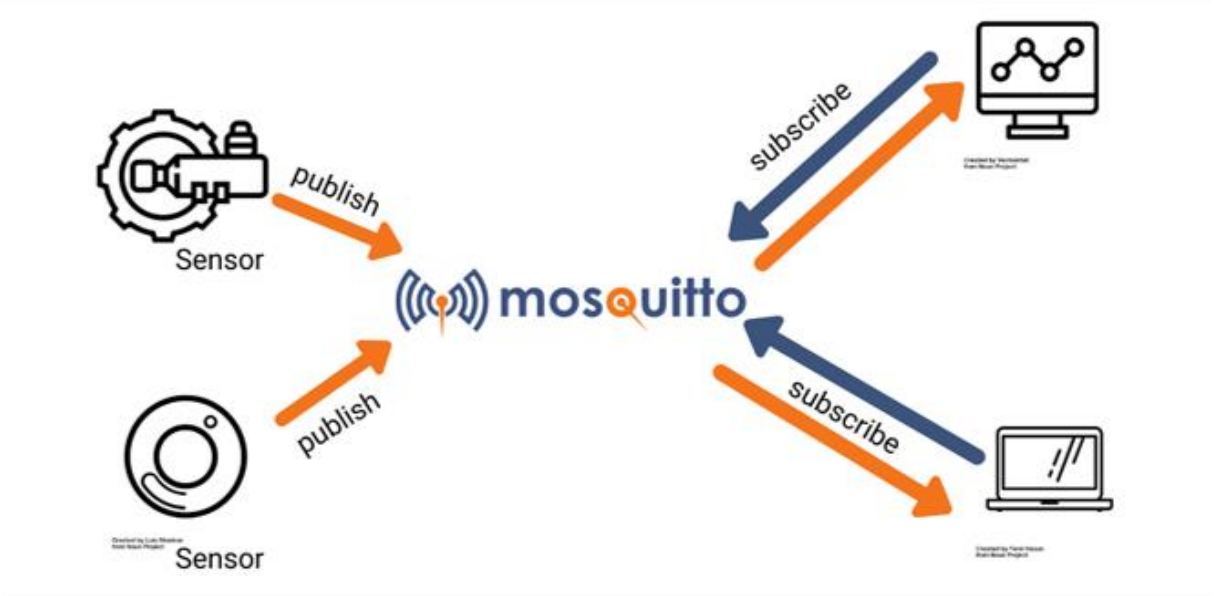
Ez kiküszöbölhető egy, a fényforrással párhuzamosan kötött kondenzátorral, ami olcsó és észrevétlen megoldás, hiszen így a relé mindenféle további vezetékezés nélkül a meglévő kapcsolók mögé beépíthető.

4. MQTT

Az MQTT egy mozaikszó, amely a Message Queue Telemetry Transport szavakból tevődik össze, és a ISO/IEC PRF 20922 definiálja a működését. Működése publish-subscribe alapú, ami azt jelenti, hogy bizonyos eszközök adatokat publikálnak, míg más eszközök a publikált üzeneteket megkapják. Az MQTT protokoll egy alacsony sávszélességű, alacsony erőforrásigényű szolgáltatás, amely szerverként, illetve ún. brókerként fut. Ezen bróker fogadja az adatokat és továbbítja a kliensek felé, akik feliratkoztak az adott adattípusra, vagy másnéven témára. A kliensek több témára is feliratkozhatnak, illetve egyetlen témára több kliens is feliratkozhat.

Az MQTT további előnye, hogy ingyenes, nyílt forráskódú szoftverként könnyen hozzáférhető, és lokálisan is telepíthető, így helyi környezetben, internetszolgáltatás kiesése esetén is akadálytalanul képes működni. Ezzel együtt pedig képes kétirányú kommunikációra is, így egy eszköz nem csupán publikálni képes az általa mért, szolgáltatott adatokat, hanem fogadni is tud a működését befolyásoló parancsokat. Ekképpen a fenti, 3. ábrán látható okosrelé nem csupán a kimenet állapotáról szolgáltat információkat, hanem egyúttal képes vezérlő parancsokat is fogadni, így távolról manuálisan, vagy esetünkben a PLC által vezérelt módon képes ki és bekapcsolni a kimenetét.

Számos MQTT verzió elérhető, én a széleskörűen használt, nyílt forráskódú, 4. ábrán látható, mosquitto distribúciót használtam a megvalósításnál.

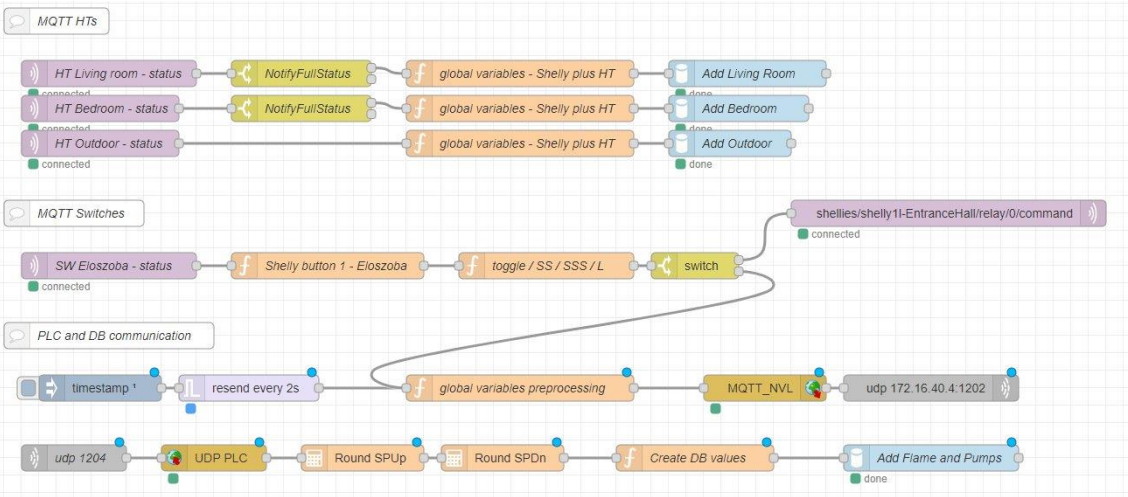


4. ábra
Az MQTT működési elve a mosquitto nevű brókerrel bemutatva

Működése stabil, minimális erőforrással beéri, gyors és hatékony megoldás. Alapvetően linux környezetre lett kifejlesztve, ám létezik Windows-os fordítás is, amit én magam is használok. A használata jól dokumentált, ám használat előtt mindenképpen be kell konfigurálni, kiváltképp a biztonságos kommunikáció, illetve az autentikációs adatok definiálásához.

5. NODE-RED

Az 5. ábrán bemutatott Node-RED egy olyan, flow-alapú, eseményvezérelt programozási nyelv, amely vizuálisan képes megjeleníteni az egyes eseményeket és azokat transzformálva, műveleteket végez az eseményhez tartozó, vagy annak pillanatában elérhető környezeti adatokkal. Eredetileg az IBM fejlesztette ki hardvereszközök és API-k közötti információcseréhez, amely képessége az IoT világában is rendkívül kedveltté tette.



5. ábra
Az MQTT üzeneteket feldolgozó Node-RED program

A házban meglévő PLC képes kezelni az MQTT üzeneteket, ám ettől függetlenül én mégis Node-RED-ben dolgoztam fel ezeket a típusú üzeneteket. A PLC programok, funkcióblokkok ugyanis jellemzően ciklikusan, nagy frekvenciával futnak le, ezzel szemben az MQTT üzenetek stochasztikusan érkeznek, nagy időközökkel. Ezért egy olyan eseményvezérelt környezet, mint a Node-RED ezt sokkal kényelmesebben és áttekinthetőbben tudja kezelni. A PLC és Node-RED közti kommunikációt a Codesys környezet NVL – Network Variable List (Hálózati változó lista) megoldásával végeztem el. Ez a megoldás azért jó, mert a változók frissítése automatikusan, a háttérben történik meg, így PLC oldalon, a ciklikus funkcióblokkokban elegendő csupán a változók módosulását figyelnem, így a programozás PLC oldalon is sokkal egyszerűbbé és kezelhetőbbé vált.

6. HMI

A lakás hőmérsékletének módosításához szükséges egy olyan felület, ahol a kívánt értékeket meg tudom adni. Ehhez a PLC belső HMI vezérlőjét használtam, és az alábbi, 6. ábrán látható felületet hoztam létre. Ezen a felületen olvasni tudom a hőmérséklet távadók által szolgáltatott információkat – hőmérséklet, páratartalom és telep állapota – , illetve meg tudom jeleníteni azt is, hogyha a távadó hosszabb ideje nem szolgáltat adatot, vagy egyéb hiba lépett fel a használat során.

A hőmérsékleti adatok mellett olvashatók és módosíthatók az egyes emeletek célhőmérséklet értékei, illetve állítható a fűtésszabályozó hiszterézis nagysága is. Ezen felül néhány visszajelző lámpa is van, amelyik a fűtés és szivattyúk állapotát mutatja, a fűtés ki-be kapcsolására szolgáló gombok között.



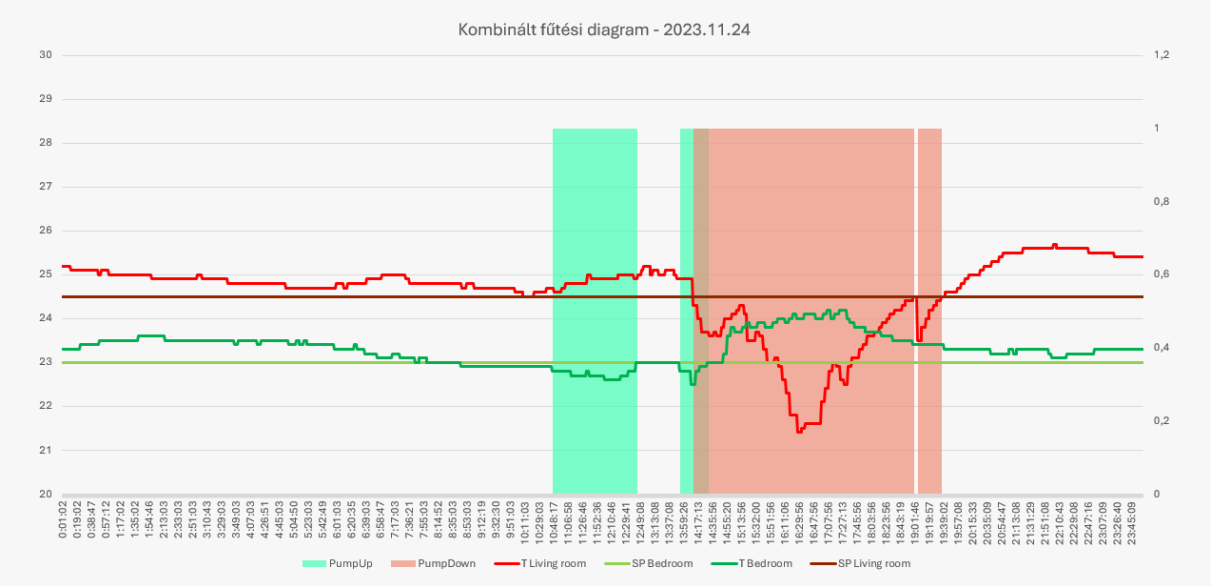
6. ábra
A fűtésszabályozó HMI felülete

7. A MŰKÖDÉS

A fűtésszabályozás működése alapjául egy egyszerű hiszterézises algoritmust használtam, amely az alábbi szerint dolgozik.

$$Fűtés \begin{cases} 1: & Fűtés = 0 \; AND \; (Hőmérséklet < Célérték - 0{,}2) \\ 0: & Fűtés = 1 \; AND \; (Hőmérséklet \geq Célérték) \end{cases}$$

A fenti algoritmus működését az alábbi, 7. ábra mutatja be. A zöld szín árnyalatai a felső emelettel kapcsolatos adatokat jelenítik meg, mint célérték, aktuális hőmérséklet, illetve az oszlopdiagram a fűtés bekapcsolt állapotát mutatja. A piros szín árnyalatai az alsó szinttel kapcsolatos adatokat jelenítik meg. Látható, hogy a két szint fűtése eltérő időpontokban kapcsol be és ki, függően a pillanatnyi hőmérséklettől, illetve az is látható, hogy délután 14:17 – 14:35között – egy rövid időre – mindkét szinten egyszerre ment a fűtés.



7. ábra
Kombinált fűtési diagram

8. HIBAKEZELÉS

A fűtési rendszer átépítésével és automatizálásával kapcsolatban néhány lehetséges hibát is kiküszöböltem, ami a tartós üzem mellett előfordulhat. Ezeket az alábbi pontokban foglaltam össze:

Mi történik hálózati hiba esetén, ha pl. a WiFi nem elérhető?

Ekkor a másodlagosan bekötött, direkt vezetékezésű hőmérséklet távadó értékére fog szabályozni a PLC, aminek célértéke jelenleg 24°C-ban van rögzítve. Később ez módosítható az alsó szint célértékére, vagy annak valamilyen származtatott értékére – így az esetleges hosszabb távollét idejére visszaszabályozott fűtési célérték esetén sem lesz túlfűtés. További lényeges pont, hogy ekkor a két fűtési kör egyszerre dolgozik, így megkerülhető az emeleti vezetékes távadó kiépítésének szükségessége.

Hogyan detektálható, ha egy szenzor nem kommunikál?

Ebben az esetben detektálni kell, hogy nem érkezik jel a távadó felől. Hiszen a hálózati hiba a PLC felől nem látszik hibának, mivel a hőmérsékleti adatok érkezése nem folyamatos, hanem ciklikus. Így én a hibás működést 20 perc, vagy annál nagyobb jelszünetre definiáltam. Vagyis minden közléshez időbélyeget rögzítek, és ezen időbélyegek életkorát ciklikusan kiértékelem az aktuális időponthoz képest.

Hogyan kezelhető a Node-RED szolgáltatás leállása, vagy az UDP kommunikáció megszakadása?

Nincs különbség az előző pontokhoz képest, a hibák minden esetben a közölt jel hiányaként manifesztálódnak, így kezelésük is ugyanúgy történik.

Van-e beépített hardveres védelem szoftverhiba esetére?

Természetesen igen, beépítésre került egy hidraulikus váltó, amely a fűtéskörök osztóköreként is szolgál, ám emellett kellő méretű puffert is ad a kazánnak túlfűtés esetére, így a kazánon beállított maximális előremenő vízhőmérséklet felé nem emelkedik a váltóban a hőmérséklet.

9. ÖSSZEFOGLALÁS

Az eszközölt fejlesztés a fűtési körön – a tapasztalatok alapján – jól sikerült. Megszűntek a korábbi jelentős hőmérséklet eltérések az egyes helyiségek között, és hideg padlójú szobák sincsenek már. Érezhetően javult a rendszer reakció ideje, bár padlófűtés révén azért elég jelentős holtidővel tud csupán beavatkozni ténylegesen a szabályozókör a szoba hőmérsékletébe. Ez idő alatt akár egy-két tized fokot tovább tud esni a hőmérséklet a helyiségben, mielőtt stabilizálódik és megfordul a görbe, ezért a túl nagy hiszterézis a komfort érzés rovására menne csak. További előnye a mostani szabályozásnak, hogy a két szint egymástól eltérően tud dolgozni, így az alsó szinten történő tevékenységek hőtermelése nem zavar be a felső szint klímájába, és nem hűl túl főzés, szárítás, vasalás – vagy akárcsak jelentős napsütés esetén sem. Összességében tehát megállapítható, hogy a tervezett kutatás elérte eredményét, és beváltotta a hozzá fűzött reményeket. Az algoritmus jól működik – a lakók legnagyobb örömére.

További fejlesztési potenciál lehet még egy olyan prediktív hőmérsékletszabályozás, amelyik a holtidővel számol, tehát még azelőtt beavatkozik, hogy a hőmérséklet ténylegesen eléri a hőmérsékleti minimumot, és azelőtt kikapcsol, hogy túllőne a célértéken. Ehhez azonban legalább század pontosságú hőmérséklettávadókra van szükség, és a jelenlegitől lényegesen nagyobb frekvenciájú mintavételezésre, hogy a hőmérsékleti adatok trendelhetők, illetve az apróbb hőmérséklet ingadozások is extrapolálhatóak legyenek.

IRODALOMJEGYZÉK

[1] Eclipse Foundation, Inc. (2024. 01 24). eclipse.org. Forrás: MQTT man page: https://mosquitto.org/man/mqtt-7.html

[2] iso.org. (2024. 01 24). Forrás: International Organization for Standardization: https://www.iso.org/standard/69466.html

[3] Shelly. (2024. 01 24). Shelly.com. Forrás: Shelly knowledge base: https://kb.shelly.cloud/knowledge-base/shelly-plus-h-t