Budapesti Műszaki és Gazdaságtudományi Egyetem
Gépészmérnöki Kar
Mechatronika, Optika és Gépészeti Informatika Tanszék

KLAFFL ATTILA ANDRÁS

HÁZI FELADAT

HŐMÉRSÉKLETMÉRÉS FUZZY-RENDSZERREL

BUDAPEST, 2023

TARTALOMJEGYZÉK

1. BEVEZETÉS

1.1. Célkitűzések

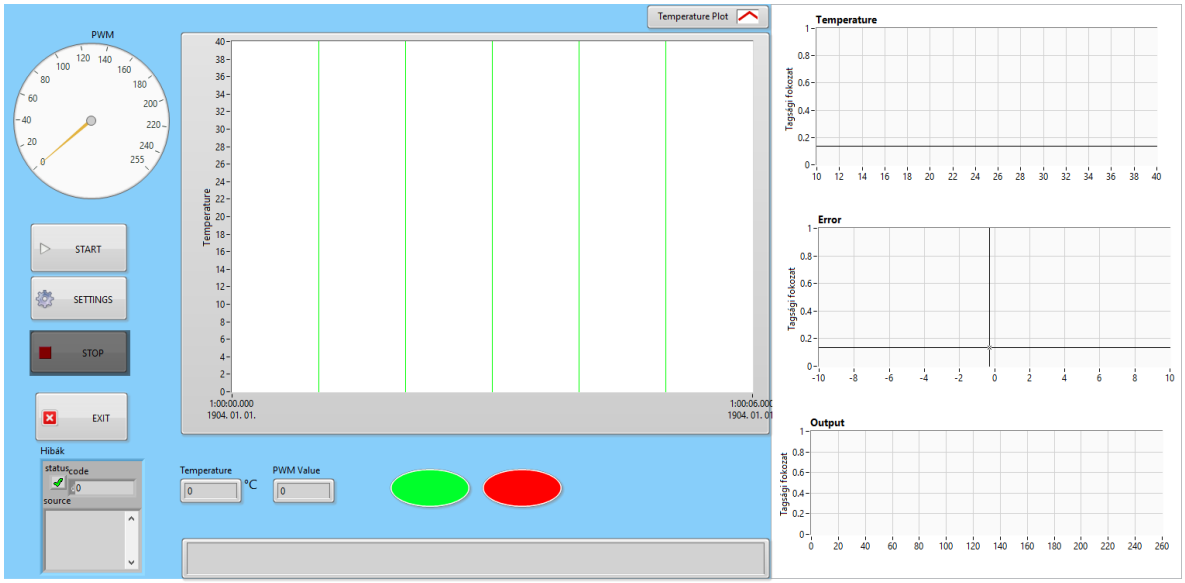
Házi feladatomban egy LM35 precíziós hőérzékelő segítségével mértem a hőmérsékletet, a mért adatokat Arduino segítségével gyűjtöttem, majd USB-n keresztül továbbítottam a számítógép felé. A LabVIEW-ban elkészített program a Fuzzy logika segítségével a bemenő hőmérsékletérték alapján és a felhasználó által beállított hőmérséklettől való eltérés meghatározása után kiszámítja a vezérléshez szükséges PWM (Pulse-Width Modulation) értéket, amelyet visszaküld a mikrokontroller felé. Az 1. ábra a mérőrendszer felépítésének vázlatát szemlélteti, a nyilak a kommunikáció irányát mutatják.

1.ábra Mérőrendszer sematikus ábra

2. FRONT PANEL

A Front Panel bal oldalán találhatóak a parancsgombok, melyekkel a felhasználó elindíthatja, leállíthatja a programot, kiléphet, valamint módosíthatja a konfigurációs beállításokat. Az alsó részen olvashatjuk le a hőmérsékletet, valamint a vezérléshez a PWM értékeket.

Középen nyomon követhetjük a hőmérséklet változását. A jobb oldalon a bementekhez és a kimenethez tartozó tagsági függvények láthatóak. Itt is lehetőségünk van leolvasni az értékeket, mivel a kurzor folyamatosan követi a hőmérséklet alakulását.



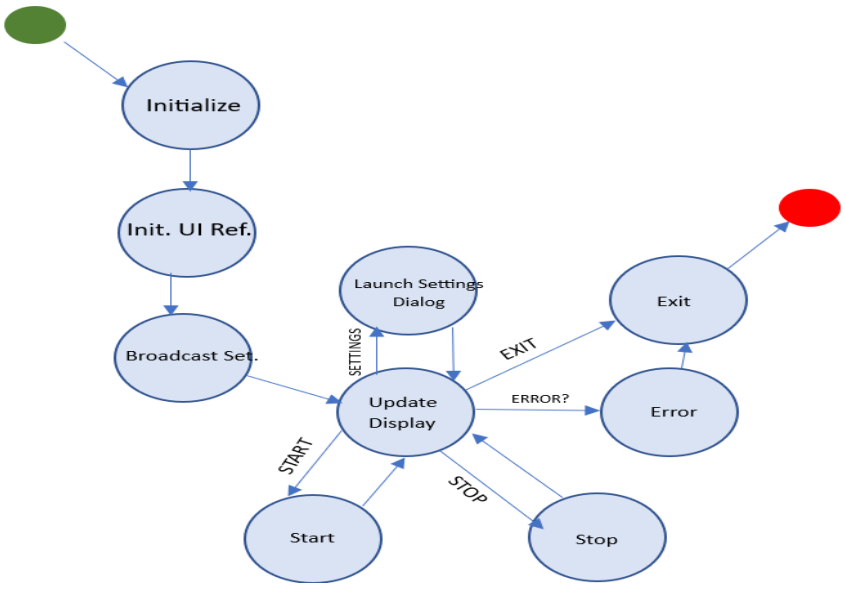
2.ábra Front Panel

3. PROGRAM ÁLLAPOTÁBRÁINAK BEMUTATÁSA

Az állapotábrák segítségével nyomon követhetjük a program belső működését. Ebben a fejezetben mind a három programrészlet – főprogram, mérésadatgyűjtés, fájlkezelés – állapotábrája bemutatásra kerül, megismerhetjük velük működésüket.

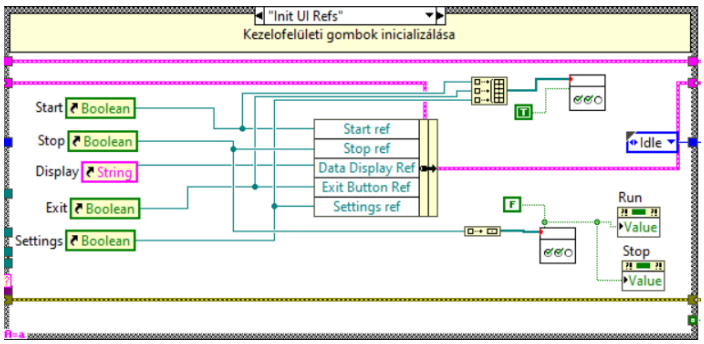
3.1. Főprogram állapotábrája

A program tervezése során az egyik legfontosabb lépés az állapotok definiálása, valamint az azok közötti átmenetek. A ábrán a fő VI (*Main.vi)* állapotábráját szemléltettem.



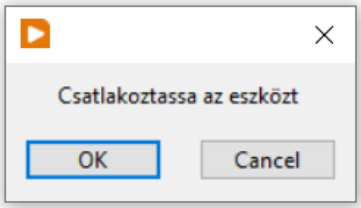
3.ábra Főprogram állapotábrája

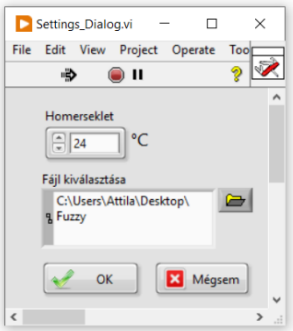
Indításkor a program automatikusan, az adott feladatok elvégzése után tovább lép a következő állapotba. Az inicializáló állapotban beolvassuk az előző futtatáskor elmentett konfigurációs fájlt, amely a beállított hőmérsékletértéket, valamint a fájl helyét tartalmazza. Ezután a gombok inicializálása következik. Az egyes állapotokban fontos, hogy melyik gombok megnyomását engedélyezzük a felhasználó számára, valamint, hogy melyeket nem. Amelyek nem elérhetők adott pillanatban, kiszürkítve jelennek meg a Front Panelen. (4. ábra)



4.ábra Gombok megjelenésének beállítása

A felhasználó a START gomb megnyomásával indíthatja el az adatgyűjtést. Amennyiben nem találja az eszközt, felugró ablakban értesíti a felhasználót a hibáról. (5.ábra) A „Cancel" gombbal bezárhatjuk az ablakot, ekkor szünetel a program. (6.ábra)

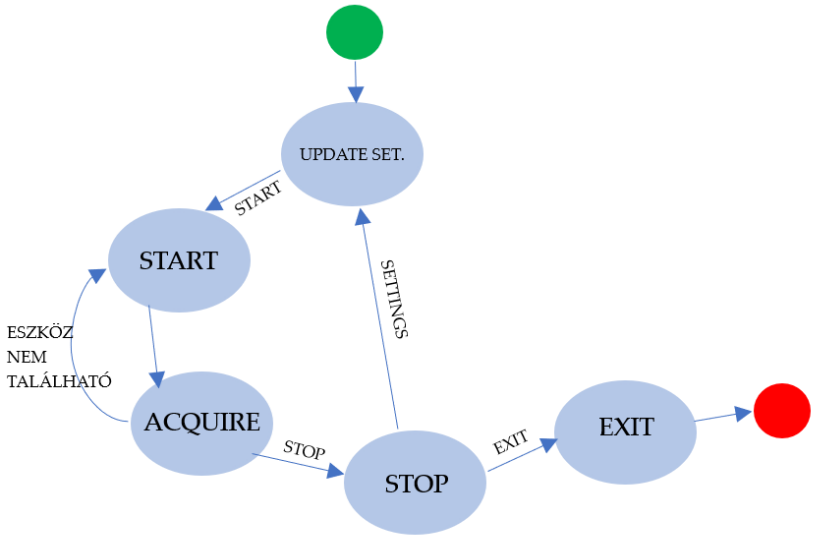




5. ábra Felugró ablak, ha az eszköz nem található 6. ábra Beállítások ablak

A STOP gombbal megállíthatjuk a mérést, itt lehetőségünk van a beállítások módosítására, melyeket jóváhagyva átírja a konfigurációs fájlban tárolt adatokat, így a következő futtatáskor ezeket olvassuk be inicializáláskor. Ebben az állapotban vagy újraindítjuk a programot vagy az EXIT gomb megnyomásával kilépünk.

3.2. *Mérésadatgyűjtés VI állapotábrája*

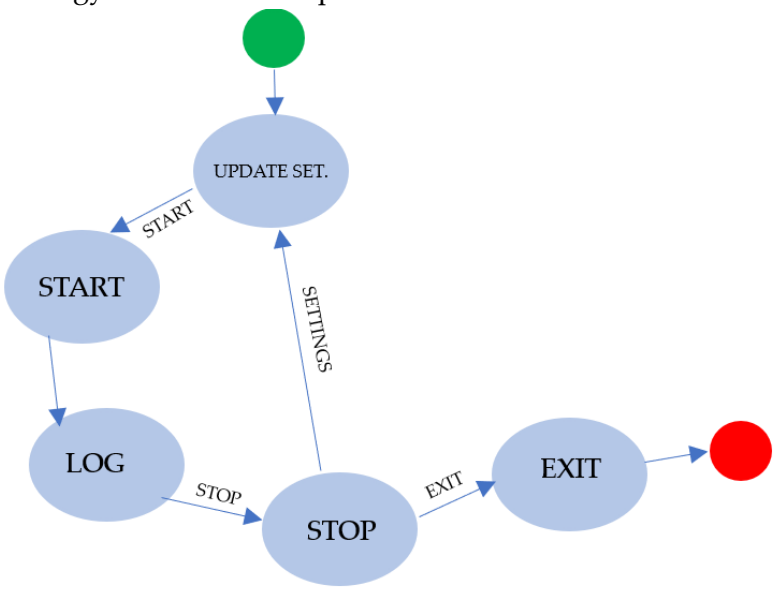


7. ábra Mérésadatgyűjtő VI állapotábrája

Inicializáláskor a konfigurációs fájl értékeit beolvassuk. A mérés elindításakor a program ellenőrzi, hogy az eszköz csatlakoztatva van e, amennyiben igen, a ciklus a következő állapotba lép és megkezdi a mérést. A mért értéket kiértékeljük és a kiszámított PWM értéket visszaküldjük az eszköznek. Ezeket egy queue-n keresztül továbbítjuk a fájlkezelő VI felé, hiszen itt fontos, hogy minden adat rögzítésre kerüljön, egy notifier-en keresztül pedig a megjelenítendő adatot küldjük. A Waveform Chart-on mindig a legfrissebb érték kerül megjelenítésre a felhasználó számára.

3.3. *Fájlkezelés VI állapotábrája*

A fájlkezeléshez is készíthetünk egy állapotábrát, amely sematikusan bemutatja a VI felépítését. A 8. ábra szemlélteti az egymást követő állapotokat.

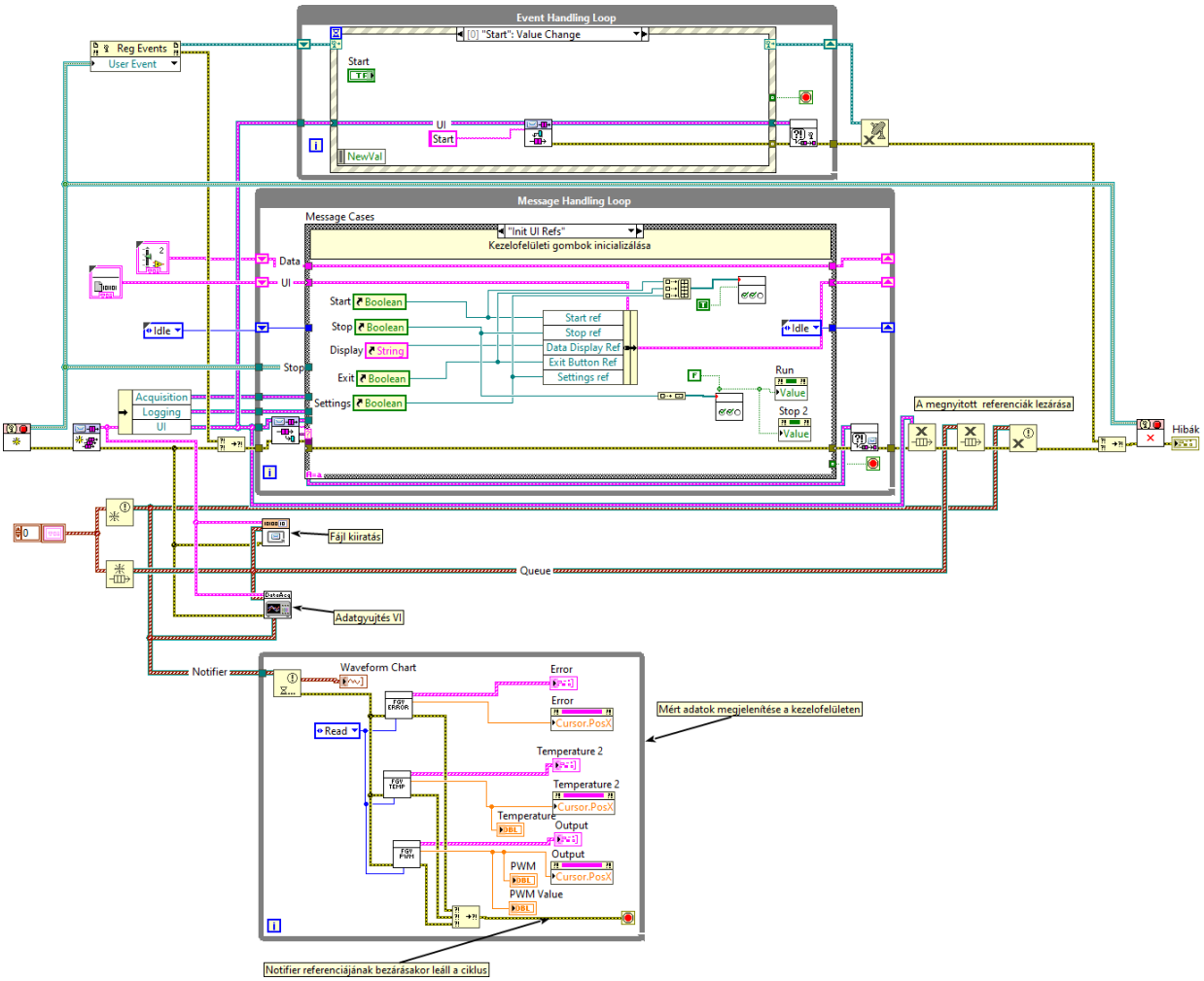


8. ábra Fájlkezelő VI állapotábrája

A felépítés az előző pontban felírt állapotábrához hasonló. A mérés elindítása után létrehozzuk a TDMS fájlt a megadott helyen, majd egy queue-n keresztül kapott mérésadatokat kiírjuk a fájlba. Beállításoknál megadhatunk másik elérési utat, hova szeretnénk elmenteni a fájlt. STOP gomb megnyomásakor a fájlt elmentjük és új mérés indításakor automatikusan új fájlt hozunk létre, így minden egyes mérés külön fájlban lesz tárolva.

4. A PROGRAM FELÉPÍTÉSE

A feladatban a LabVIEW-ban elérhető Queued Message Handler Template-et használtam, melyet *Mérés* és *Logging* VI-okkal egészítettem ki. Ezeket a Contiunous Measurement and Logging Template alapján készítettem el, a feladathoz szabva. A főprogram egymással párhuzamosan futó ciklusait a 9.ábra szemlélteti.

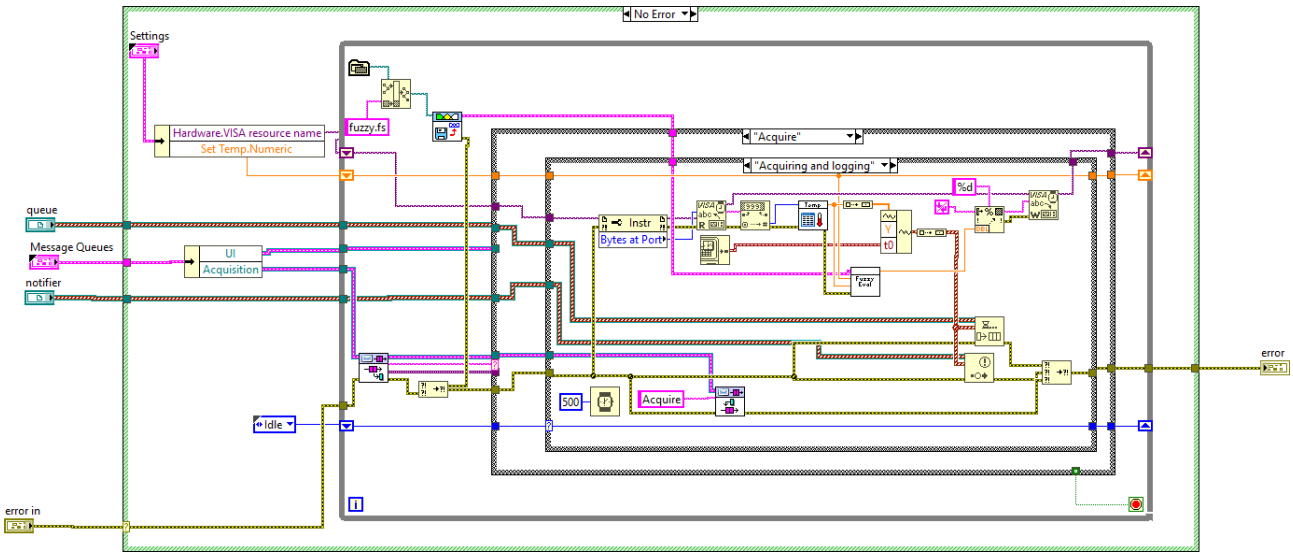


9. ábra Egymással párhuzamosan futó ciklusok

A felső ciklus az eseménykezelő ciklus. Lekezeli a felhasználó általi tevékenységeket a felületen. Felhasználó a gombok segítségével ad utasítást a programnak, melyet az eseménykezelő észlel, és a kért állapotba lépteti a többi ciklust. A legfontosabb események a program indítása, konfigurációs fájl beállításai, a szüneteltetés, valamint a leállítás.

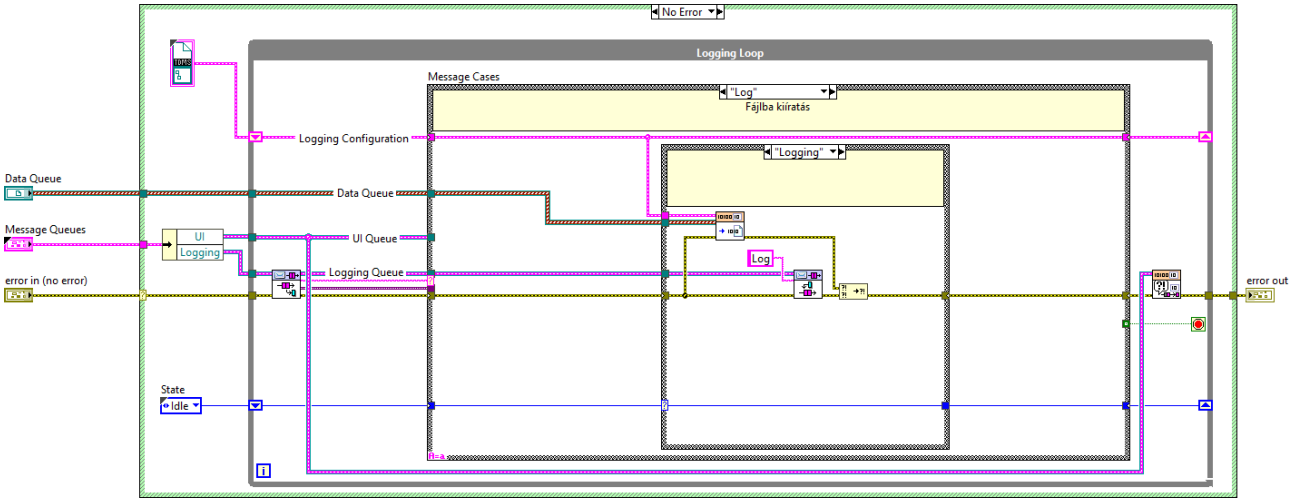
Ezután az úgynevezett *Message Handler* ciklus következik. Itt küldjük tovább állapotonként a megfelelő üzenetet a többi ciklus felé. Például elindítjuk a mérésért felelős ciklust, leállítjuk azt. Adatcsomag küldésére is van lehetőség

A főprogramban subVI-ként vannak elhelyezve a mérést, valamint a fájlkezelést végző ciklusok. Az adatgyűjtés Block Diagrammját a 10. ábrán láthatjuk. Megtörténik az eszköz inicializálása és a kapcsolat ellenőrzése, ezután pedig létrejön a folyamatos kommunikáció. Adatokat olvas be az eszköztől, illetve továbbít felé.



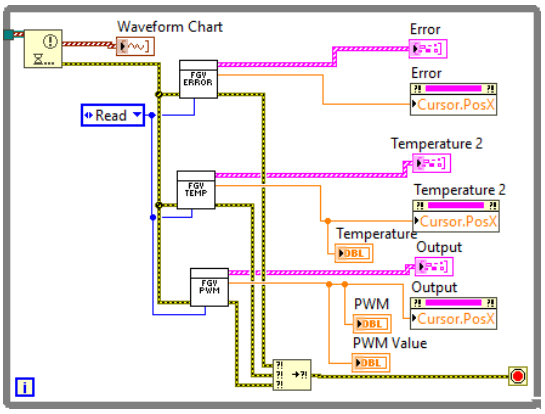
10. ábra Mérésadatgyűjtés

A fájlkezelés Block Diagrammjának egy részletét a 11. ábra mutatja. Az adott állapotban folyamatosan történik az adatmentés egy TDMS fájlba.



11. ábra Fájlkezelés

Végül egy külön ciklusba került a felhasználó számára megjelenített tartalom (12. ábra). A legfrissebb hőmérsékletérték jelenik meg a diagrammon, valamint a kurzor értéke is az aktuális Fuzzy bementen és kimeneten jelenik meg. A Fuzzy változók - a hőmérséklet, a kimenet, valamint az error - esetében globális változót használtam.

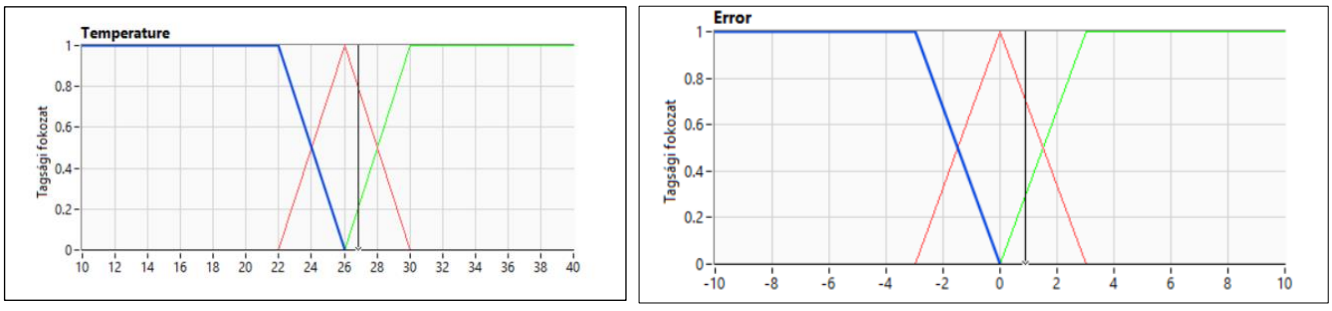


12. ábra Megjelenítés

5. FUZZY LOGIKA MEGVALÓSÍTÁSA

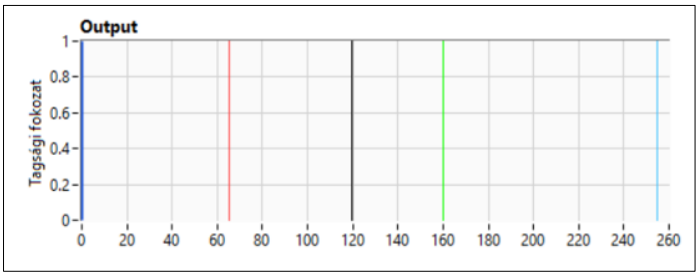
5.1. Be-, illetve kimenetek

A Fuzzy-rendszer kettő bemenettel rendelkezik, ezek a mért hőmérséklet, valamint a beállított értéktől való eltérés. Ezek függvényében számítja ki a kimeneti értéket 0-255 sávban.
A bemeneteknek standard partíciót kell alkotniuk. Ez azt jelenti, hogy egy bemeneti változó (pl. Hőmérséklet) tagsági függvényeinek összege az értelmezési tartomány minden pontjában 1-et kell adjon. Emiatt a széleken trapéz alakú tagsági függvényt kell alkalmazni, mind a két bemenet esetén.
Az alábbi két ábra a bemenetekhez tartozó tagsági függvényeket ábrázolja. A hőmérséklethez tartozó tagsági függvényeket a beállításoknak megfelelően toljuk el, így a kiválasztott értékhez mindig a középső függvény tartozik.



13. ábra Fuzzy bemenetek

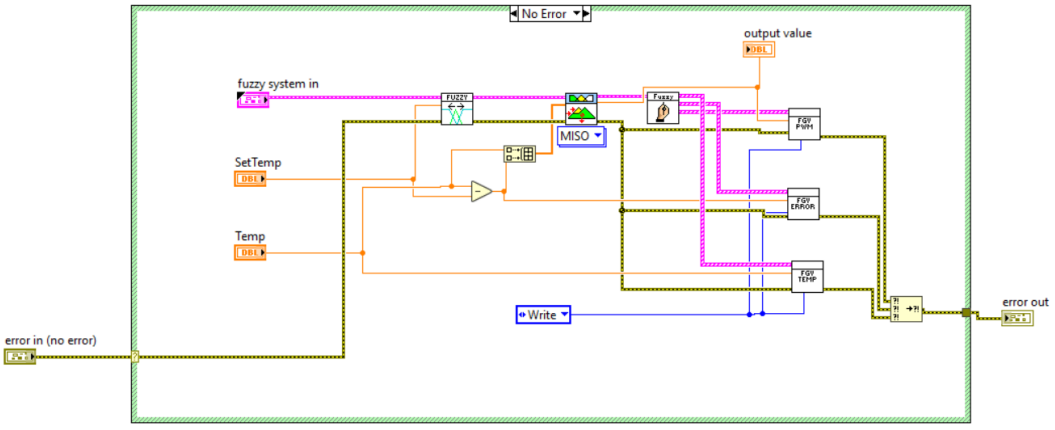
A kimenetre ez a szabály nem vonatkozik, így ide az úgynevezett *Singleton* függvényt választottam. (14. ábra)



14. ábra Kimenet tagsági függvényei

5.2. *Programban megvalósítva*

A realizációt a *Meres* VI-ban található *FuzzyEv* nevű subVI valósítja meg. (15. ábra) A bemenetekből a definiált szabályoknak megfelelően kiszámítja a kimeneti értéket. A kettő bemenetet és kimenetet egy-egy globális változóban tárolja, így a Front Panelen megjeleníthetőek lesznek a függvények, valamint egy kurzor segítségével nyomon követhetjük, éppen milyen értékeket vesznek fel a változók.



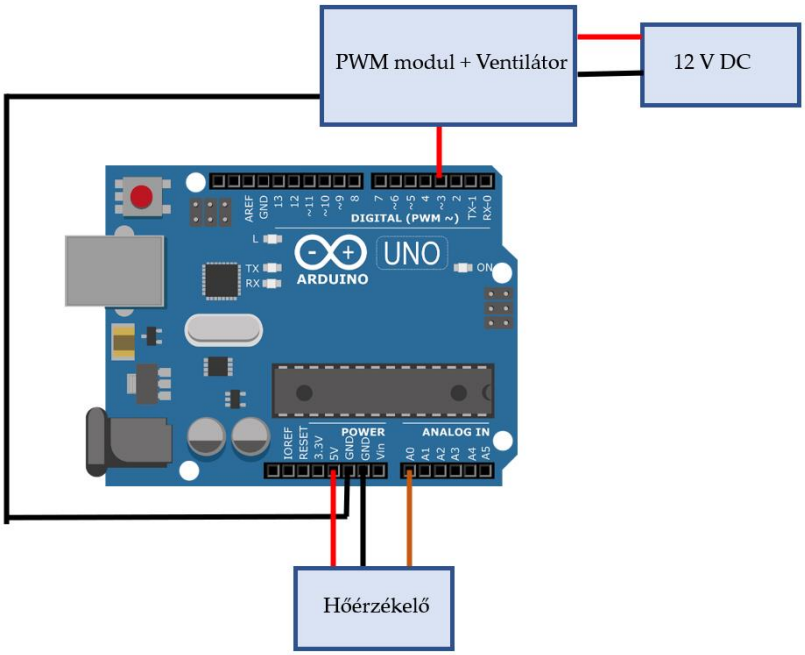
15. ábra Programbeli megvalósítás

6. HARDVER ESZKÖZÖK

A projektben felhasznált hardver eszközök:

- Arduino UNO
- LM35 precíziós hőérzékelő
- 12 V DC ventilátor
- D4184 PWM modul

A sematikus vázlatot a 16. ábra mutatja.



16. ábra Hardver eszközök kapcsolása

FELHASZNÁLT IRODALOM

National Instruments: Queued Message Handler dokumentáció
Link: https://www.ni.com/hu-hu/support/documentation/supplemental/21/using-a-queued-message-handler-in-labview.html
Utolsó megtekintés dátuma: 2023. május 30.