



SZAKDOLGOZAT FELADAT

Berta Máté (HPD5LB)

Villamosmérnök hallgató részére

Talaj nedvességtartalom mérés IoT eszközzel

Manapság népszerű kutatási, fejlesztési terület az IoT. Ezen belül is nagy népszerűségnek örvendenek a különféle otthon automatizálási megoldások. A hallgató feladata is ebbe a témakörbe illeszkedik egy automatizált kerti öntözőrendszerhez tartozó mérőegység megtervezésével. Az egységgel szemben további elvárás, hogy illeszkedjen a hallgató önálló laboratórium feladatában tervezett rendszerbe, a méréseket adott időközönként és módon végezze, a gyűjtött adatokat a rendszer központi egysége felé továbbítsa.

A feladat első része a mérőegység rendszertervének elkészítése, figyelembe véve a már meglévő rendszerbe történő illesztés támasztotta követelményeket. Ennek része a megfelelő perifériák (például rádiós kommunikációhoz, beavatkozó szervhez) kiválasztása, a mérési adatok megfelelő formátumra hozása, a mérőegység villamosenergia igényének figyelembevétele (becslés akkumulátoros üzemidőre), költségszámítás és az elosztott szenzorhálózatba építés realitásának mérlegelése.

A hallgató feladatának – a fentiek felül – a következőkre kell kiterjednie:

- Szakirodalmi kutatás a talaj nedvességtartalom kapacitív elven történő méréséről
- Kapacitív szenzor kiválasztása
- A választott szenzorhoz különböző vizsgálgó jeleket létrehozó áramkörök illesztése
- Kapcsolási rajz készítés
- NYÁK terv készítés
- Az elkészült áramkör bemérése, élesztése
- A mérőegység feldolgozó egységére beágyazott szoftver készítése
- Kalibrációs mérés a szakirodalomban is megtalálható módszerek segítségével
- A központi egység szoftverében módosítások végzése a mért adatok rendszerezett megtekinthetőségére elérése érdekében

Tanszéki konzulens: Naszály Gábor, mestertanár

Budapest, 2020.10.10.

.....
Dr. Dabóczi Tamás
tanszékvezető
egyetemi tanár, DSc

Budapesti Műszaki és Gazdaságtudományi Egyetem
Villamosmérnöki és Informatikai Kar
Méréstechnika és Információs Rendszerek Tanszék

Berta Máté

TALAJ NEDVESSÉGTARTALOM MÉRÉS IOT ESZKÖZZEL

KONZULENS

Naszály Gábor

BUDAPEST, 2020

Tartalomjegyzék

1	Bevezetés	8
1.1	Otthon automatizálás kertkapcsolatos házakban.....	8
1.2	Öntözőrendszer bemutatása	9
2	Kapacitív talaj nedvességtartalom mérés	11
2.1	Fizikai jelenségek.....	11
2.1.1	Definíció és validációs módszer.....	11
2.1.2	Kapacitív szenzor mérési elve.....	12
2.2	Szenzor jellemzése és kalibrációs mérés.....	12
2.2.1	Választott szenzor és mérési módszere	12
2.2.2	Mérőjel alakjának befolyásoló hatásának vizsgálata	13
2.2.3	Következtetés	19
3	Hardver tervezés	20
3.1	Elvi kapcsolási rajz	20
3.1.1	Mágnesszelep működése elve	20
3.1.2	Mágnesszelep vezérlése	22
3.1.3	Kliens tápellátása	24
3.1.4	Bemeneti feszültség védelmei.....	24
3.1.5	Akkumulátor töltő áramkör.....	25
3.1.6	Szabályozott tápfeszültség előállítása	26
3.1.7	Programozó interface	27
3.1.8	Méréshez szükséges csatlakozó	28
3.1.9	Mérőjel előállító áramkör.....	28
3.1.10	Hőmérséklet- és nyomásmérő áramkör	31
3.1.11	ESP32 WROOM-32 periféria áramkörei	32
3.1.12	Egyéb funkciókat ellátó áramkörök.....	33
3.2	Nyomtatott huzalozású áramköri lemez terv.....	33
4	Szoftver tervezés	35

4.1	Kliens beágyazott szoftvere	35
4.2	Központi egység szoftvere	36
4.2.1	SWAN szerver szoftveres környezete.....	36
4.2.2	MQTT broker	36
4.2.3	SQLite	37
4.2.4	Node-RED.....	37
4.2.5	SWAN SERVER flow	40
5	Konklúzió.....	48
5.1	Eredmények.....	48
5.2	Továbbfejleszthetőség.....	48
6	Irodalomjegyzék.....	49
7	Függelék.....	50

HALLGATÓI NYILATKOZAT

Alulírott Berta Máté, szigorló hallgató kijelentem, hogy ezt a szakdolgozatot meg nem engedett segítség nélkül, saját magam készítettem, csak a megadott forrásokat (szakirodalom, eszközök stb.) használtam fel. Minden olyan részt, melyet szó szerint, vagy azonos értelemben, de átfogalmazva más forrásból átvettem, egyértelműen, a forrás megadásával megjelöltem.

Hozzájárulok, hogy a jelen munkám alapadatait (szerző(k), cím, angol és magyar nyelvű tartalmi kivonat, készítés éve, konzulens(ek) neve) a BME VIK nyilvánosan hozzáférhető elektronikus formában, a munka teljes szövegét pedig az egyetem belső hálózatán keresztül (vagy hitelesített felhasználók számára) közzétegye. Kijelentem, hogy a benyújtott munka és annak elektronikus verziója megegyezik. Dékáni engedéllyel titkosított diplomatervek esetén a dolgozat szövege csak 3 év eltelte után válik hozzáférhetővé.

Kelt: Budapest, 2020. 12. 06.

.....
Berta Máté

Összefoglaló

Ide jön a ½-1 oldalas magyar nyelvű összefoglaló, melynek szövege a Diplomaterv Portálra külön is feltöltésre kerül.

Abstract

Ide jön a ½-1 oldalas angol nyelvű összefoglaló, amelynek szövege a Diplomaterv Portálra külön is feltöltésre kerül.

1 Bevezetés

1.1 Otthon automatizálás kertkapcsolatos házakban

Az otthonainkban egyre jobban betörő IoT eszközök megváltoztatják elvárásainkat háztartási eszközeinkkel szemben. Az „smart” jelzővel ellátott használati tárgyaink száma egyre nő, melyek gyakran, egyes kényelmi funkciókon túl, távoli elérést tesznek lehetővé. Olyan információkat gyűjthetünk otthonunkról, mint például napelem celláink teljesítménye, házunk áramfelvétele, háztartási eszközeink fogyasztása vagy akár bejárati ajtónkról élő kameraképet nézhetünk távolról.

Az otthon automatizálási feladatok ellátásával szemben állított jellemző műszaki követelmények közé tartozik az alacsony fogyasztás, integrálhatóság valamilyen IoT rendszerbe, könnyű kezelőfelület és relatív alacsony ár.

Hazánkban lakóövezet szerint a házak 62%-a családi ház, zártkerti vagy külterületi ingatlan. [\[1\]](#) Az ilyen típusú ingatlanok gyakran rendelkeznek kertkapcsolattal, ahol konyhakert vagy gyümölcsös található vagy telepíthető. Az ilyen ingatlanok esetén az otthon automatizálást kiterjeszthető a kerti feladatok automatizálására, melyek közül az egyik legkevesebb infrastrukturális változtatást megkövetelő az öntözőrendszer „okosítása”.

Sok kert rendelkezik kiépített öntözőrendszerrel, amely egy vagy több csap megnyitásával lehetővé teszi az öntözést. Ezek a rendszerek gyakran időzítőkkal vannak ellátva, mely bizonyos szintű automatizált megoldást nyújt, de ezek a rendszerek nem tudják figyelembe venni az időjárási körülményeket, gyakran sok emberi beavatkozást is igényelnek, illetve állapotuk nem ellenőrizhető távolról.

A kiépített öntözőrendszerek viszont lehetőséget adnak a könnyű automatizálásra, hiszen egy mágnesszelepes csap segítségével villamos jelekkel irányítható az öntözés.

1.2 Öntözőrendszer bemutatása

Jelen szakdolgozat témája egy öntözőrendszerbe építhető mérő- és beavatkozó egység. A rendszer követelményeit és a mérőegység alapelvének működőképességét önálló laboratórium feladatom definiálja és támasztja alá, az erről készült dokumentum elérhető nyilvánosan. [\[2\]](#)



1-1. ábra SWAN logó

Az öntözőrendszer a Smart Watering Automation System (S.W.A.N.) nevet kapta. A mérő- és beavatkozó egységre SWAN kliensként, a rendszerhez tartozó központi egységre SWAN szerverként hivatkozok a továbbiakban.

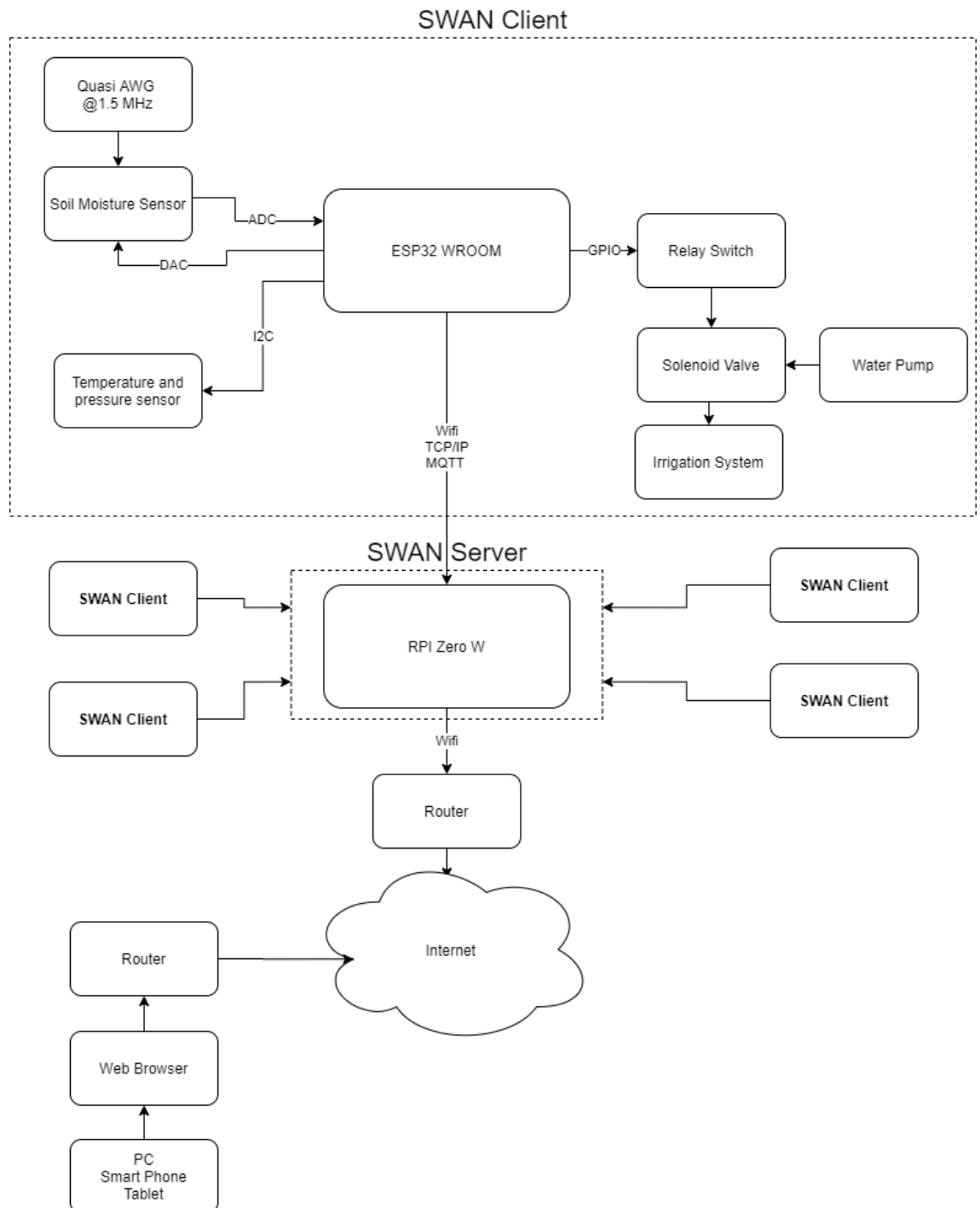
Minden kliens a talaj nedvességtartalmát, a levegő hőmérsékletét és nyomását méri. A mért adatok felhasználásával avatkozik be minden egység, amely ellátott az ehhez szükséges hardver elemekkel. Minden mért adatot batchenként közlik a kliensek a szerverrel, mely utasítást is adhat a klienseknek, amikor azok bejelentkeznek hozzá.

A szerver és a kliensek feltételezik, hogy saját rádiós perifériáiknak hatótávolságán belül vezeték nélküli kapcsolaton (WIFI) keresztül hozzáférésük van a helyi hálózathoz, melyen kommunikálnak egymással.

A kliensek akkumulátoros üzemre optimalizáltak, viszont hálózatról (3,5V-6V DC) is meghajthatóak. Az akkumulátor töltése DC tápon keresztül történik, a szükséges töltések közötti idő minél hosszabbra növelése szempontja mind a hardveres, mind a szoftveres tervezésnek.

A SWAN kliens és szerver az alábbi rendszerterv szerint épül fel és csatlakozik egymáshoz.

Rendszerterv



1-2. ábra S.W.A.N rendszerterv

2 Kapacitív talaj nedvességtartalom mérés

2.1 Fizikai jelenségek

2.1.1 Definíció és validációs módszer

A talaj nedvességtartalma általában százalékban kifejezett mennyiség, amely arra megmutatja, hogy 1 gramm talaj hány g vizet tartalmaz. Mértéke tehát $\left[\frac{g}{g} \cdot 100\right] = [\%]$.

Egyszerű mérési módja, ha ismert tömegű, adott nedvességtartalmú talajt kiszárítunk és tömeg különbségét az elpárolgott víz tömegének vesszük. Ekkor a következőképpen számíthatunk:

$$S_{[\%]} = \frac{m_{H_2O}}{m_{talaj}} = \frac{m_{H_2O}}{m_{minta} - m_{H_2O}} = \frac{m_{minta} - m_{minta}'}{m_{minta}'}$$

Másik mérési módszer lehet, hogy ismert tömegű száraz, zérusnak tekinthető vizet mennyiségű tartalmazó talajhoz ismert tömegű vizet adunk. Ekkor különbség képzés nélkül megkaphatjuk ugyan azt az arányszámot:

$$S_{[\%]} = \frac{m_{H_2O}}{m_{talaj}}$$

Mindkét módszer tömegmérésre visszavezetett, amely jó pontosságú a jelenleg ismert technológiával. Viszont ezek a mérési módszerek nem alkalmasak gyakorlati helyzetekben, hiszen a vizsgálandó talaj általában nem eltávolítható és nem vonható ki belőle minden nedvesség. Emellett a talajba jutó vízmennyiség tömege gyakran nem ismert.

Az előzőekben ismertetett módszerek viszont alkalmasak más mérési módszerek ellenőrzésére, azoknak validációjára. Hiszen más fizikai jelenséget kihasználó mérési módszerrel mért értékekhez, ezekkel a módszerekkel lehetőség nyílik konkrét számértékeket rendelni.

Megjegyzendő, hogy a talaj nincs definiálva jelen dokumentumban, annak ellenére, hogy a talaj nedvességtartalom definícióját nem, de a továbbiakban bemutatott mérési módszert befolyásolja a talaj milyensége. Ennek vizsgálatára nem került sor, az elvégzett mérések kvarchomokban történtek, amely megfelelő tisztaságú és mennyiségű könnyen beszerezhető, illetve jól kiszárítható.

2.1.2 Kapacitív szenzor mérési elve

A talaj nedvességtartalmához valamilyen fizikai mennyiséget kell rendelnünk, annak érdekében, hogy mérni tudjuk a 2.1.1 fejezetben bemutatott módszerek alkalmazása nélkül.

Cél, hogy a mért fizikai mennyiséget feszültségmérésre vezessük vissza, mivel feszültséget kellő pontossággal és mikroprocesszorral tudunk mérni. Figyelembe vehetjük, hogy a talaj elektromos térben való viselkedése jellemezhető a komplex relatív permittivitásával (ϵ_r^*). **HIVATKOZÁS**

A következő egyenlet szerint változik ϵ_r^* :

$$\epsilon_r^* = \epsilon_r' - j \cdot \epsilon_r'' = \epsilon_r' - j \left(\epsilon_{relax}'' + \frac{\sigma_{dc}}{2\pi f \epsilon_0} \right)$$

Ahol σ_{dc} a talaj vezetőképessége, ϵ_{relax}'' molekulák nyugalmi hozzájárulása, ϵ_r' egy külső tér által az anyagban tárolt energiáját fejezi ki és j az imaginárius egység. A komplex permittivitás imaginárius része jellemzi, hogy az anyag mennyire „veszteséges” az elektromos térre nézve. A veszteség mértéke függ a frekvenciától, a nedvességtartalmától és az anyag só és ion tartalmától. Tehát a közeg nedvesség tartalmával összefüggésben lévő fizikai mennyiség a komplex relatív permittivitás.

Amennyiben G_0 ismert geometriájú szenzor segítségével villamos teret juttatunk a talajba a következő összefüggés szerint definiálhatjuk a kapacitást:

$$C = \epsilon_r^* \epsilon_0 G_0$$

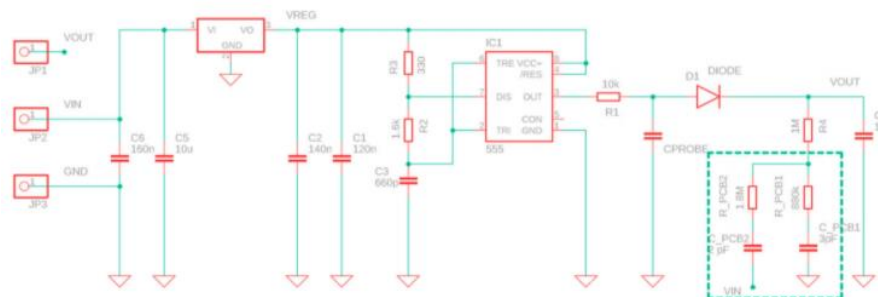
Az így bevezetett kapacitás lineárisan függ a komplex relatív permittivitástól, ráadásul villamosan mérhető mennyiség. Kapacitás mérésre több ismert módszer is létezik, melyek többsége feszültség mérésre vezeti vissza a problémát.

2.2 Szenzor jellemzése és kalibrációs mérés

2.2.1 Választott szenzor és mérési módszere

A választott szenzor egy márkajelzés nélküli különböző internetes áruházakban olcsón kapható és hobbi IoT alkalmazásokban gyakran használt mérőegység. A szenzor gyártója DFROBOT és SKU:SEN0193 néven érhető el. Specifikációja csak a tápfeszültség határait adja meg, működését és kapcsolási rajzát a hivatkozott cikk alapján ismertem meg.

A szenzor vizsgáló jelként 1.5 MHz frekvenciájú, egyenkomponenssel rendelkező négyszögjelet alkalmaz, mely előállítására bistabil multivibrátort alkalmaz 555 időzítő IC és passzív komponensek alkalmazásával.

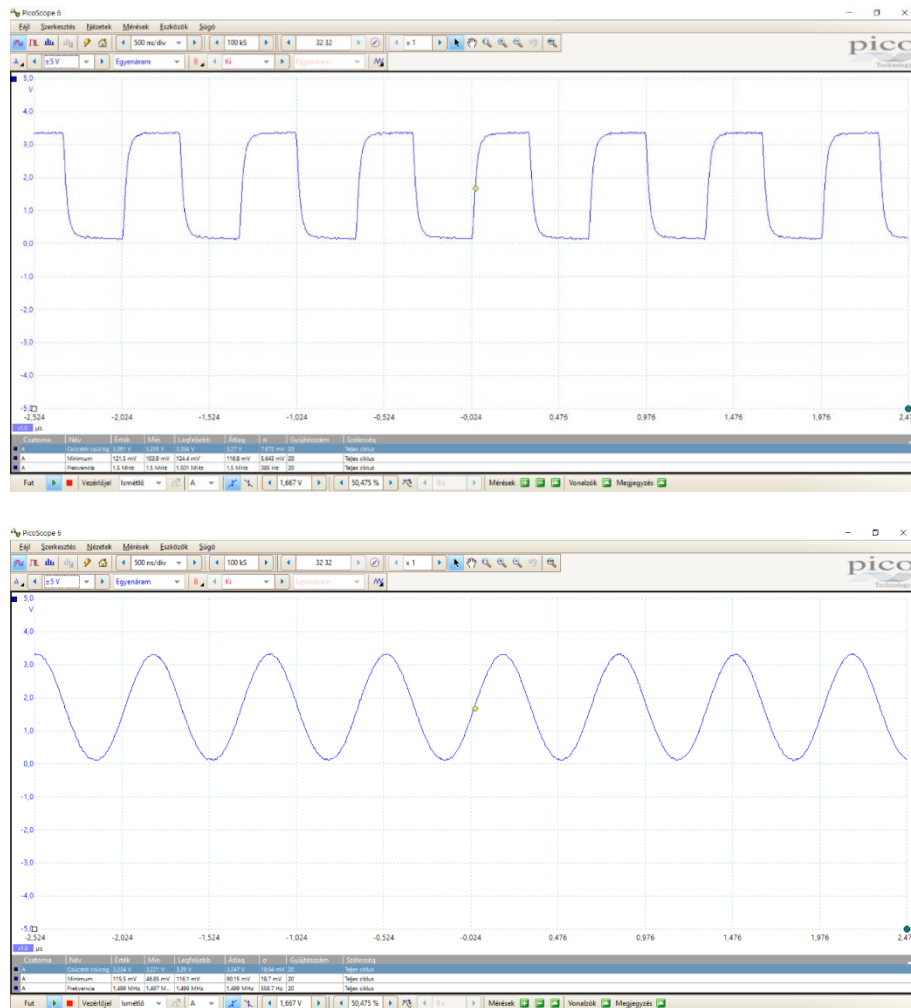


2.2.2 Mérőjel alakjának befolyásoló hatásának vizsgálata

2.2.2 Mérőjel alakjának befolyásoló hatásának vizsgálata

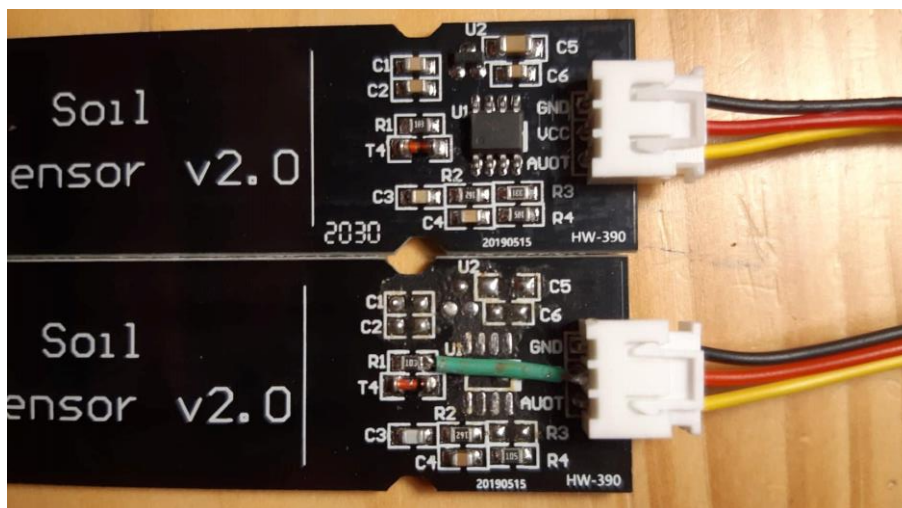
A vizsgálat továbbá lehetőséget nyújt, hogy a mért feszültségek és a talaj nedvességtartalma közötti kapcsolatot karakterizáljam, kalibrációs mérésenként felhasználjam, erre a 2.1.1 fejezetben említett második módszert alkalmaztam.

A vizsgálat a szenzor által előállított négyszögjelnél meredekebb élekkel rendelkező négyszögjellel és szinusz jellel történt. Minden jel 0 és 3.3 V közötti, egyenfeszültségű komponens is tartalmazó és 1.5 MHz frekvenciájú. A használt mérőjelek a 2-2. ábraán láthatóak.



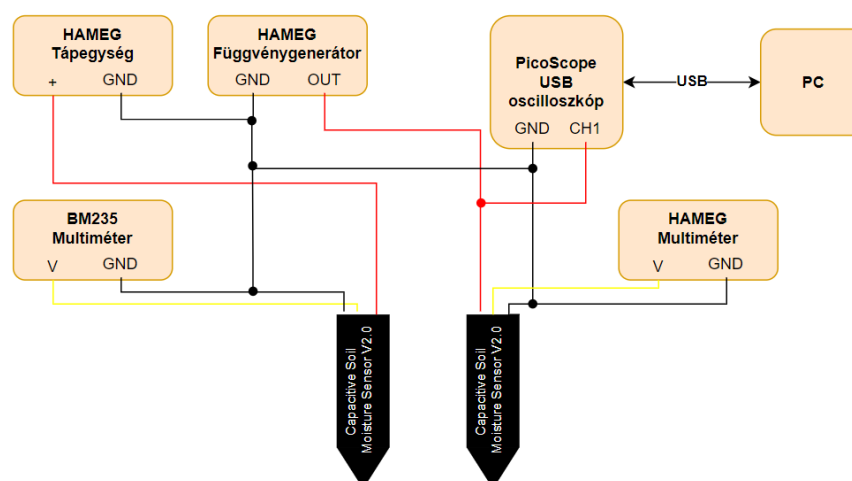
2-2. ábra Függvénygenerátorral előállított mérőjelek

A mérésem során két egy napon gyártott szenzor felhasználásával történt, melyek közül az egyiket módosítottam úgy, hogy külső mérőjel csatlakozhasson hozzá. A módosítás során az 555-ös IC és a hozzá tartozó áramkörü elemeket leforrasztottam és a eredeti kapcsolás R1 ellenállását közvetlenül a Vin bemenettel kapcsoltam össze. A két szenzor egymás mellett a 2-3. ábra látható.



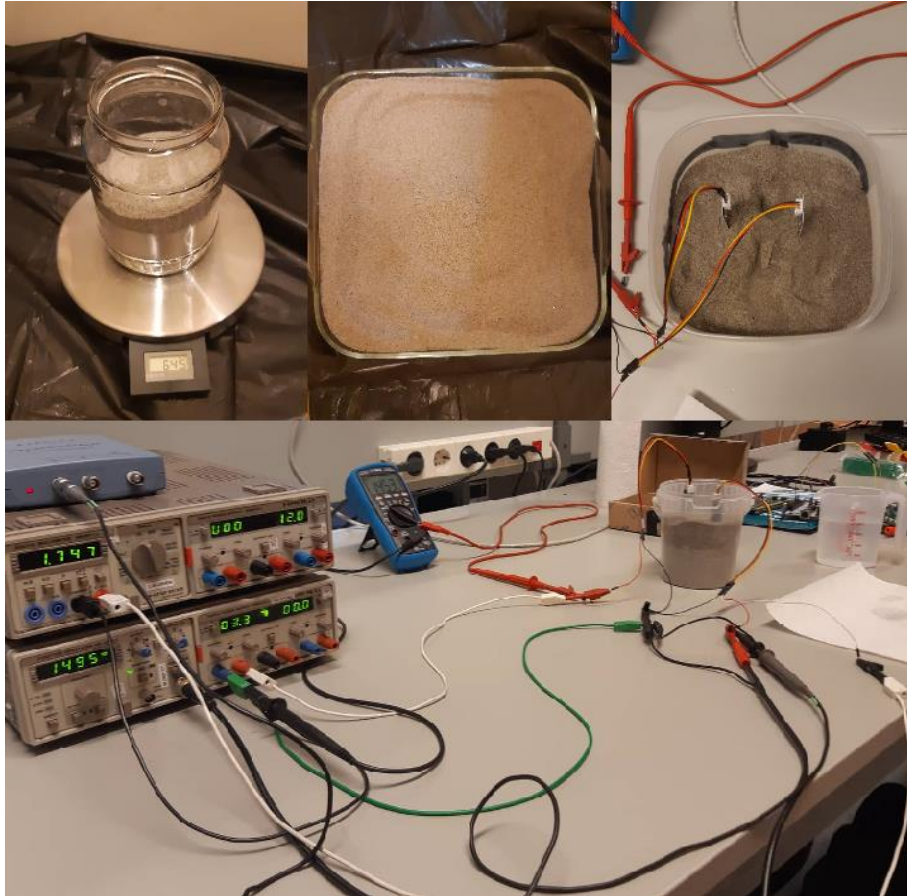
2-3. ábra Eredeti (felső) és módosított (alsó) szenzor

A kalibrációs mérések során a nem módosított és a módosított szenzor feszültségértékeit jegyzem fel és hasonlítom össze táblázatos és grafikus formában. A mérési elrendezés sematikus vázlata a 2-4. ábraán látható.



2-4. ábra Kalibrációs mérés

A mérés előkészületeként 5000g szárított kvarchomokot egy üvegedényben 1 órán keresztül 110 °C hőmérsékleten tovább szárítottam. Az így kapott homokot tekintem zérus nedvességtartalmú talajnak, számításaim során 0g víztartalmat tulajdonítok neki. A 5000g homokot 1:2:2 arányban felosztottam, egy teszt mérés és két éles mérés számára. A tesztmérés során megfigyeltem, hogy a homok milyen módon veszi fel a vizet, mennyire számít a szenzorok mozgatása és talajban lévő mélysége a mérés során. A 2-5. ábra a mérés előkészítése és a mérés elvégzése közben készített képeket tartalmaz.

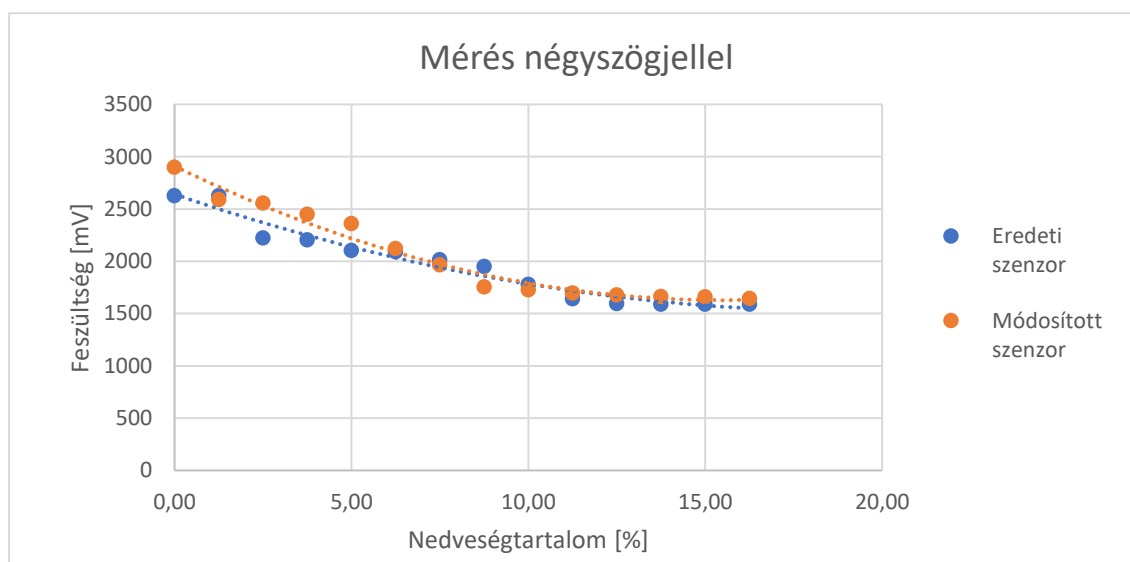


2-5. ábra Kalibrációs mérés fázisai

Az éles mérések során ügyeltem, hogy a szenzorok szimmetrikusan helyezkedjenek el a mérőedényben, teljes felületüket talaj fedje és a hozzáadott víz azonos mértékben jusson el hozzájuk. Víz hozzáadása előtt feljegyeztem a szenzorok által mutatott feszültséget, majd 25ml-es lépésenként adtam hozzá szobahőmérsékletű csapvizet. A víz hozzáadása után, megvártam amíg a multiméterek legalább két tizedes jegyig állandó értéket mutatnak és feljegyeztem a feszültséget egy táblázatba. A víz hozzáadását addig folytattam amíg legalább öt azonosnak tekinthető feszültségértéket nem olvastam le. A két éles mérés között a szenzorokat nedvszívó törlőkendővel tisztítottam meg, annak érdekében, hogy a felületen ne maradjon nedves homok.

Az első mérés négyszögjellel történt és a következő táblázatban rögzített adatokat kaptam, melyeket grafikonon is ábrázoltam.

Nedvességtartalom [%]	Feszültség ("kontroll" szenzor) [mV]	Feszültség (kísérleti szenzor) [mV]
0,00	2214	2901
1,25	1842	2592
2,50	1801	2556
3,75	1645	2451
5,00	1474	2360
6,25	1151	2121
7,50	1064	1965
8,75	979	1755
10,00	961	1728
11,25	945	1697
12,50	946	1678
13,75	947	1662
15,00	945	1659
16,25	939	1644

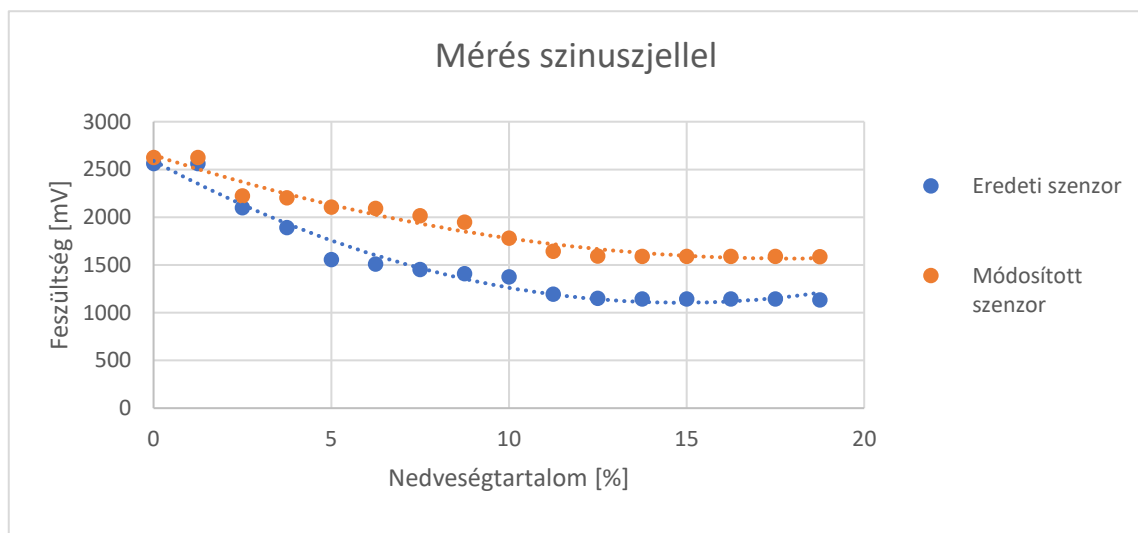


A grafikonon látható trendvonalak (szaggatott) másodfokú polinomiális görbék. Jól látható, hogy 5% felett a két görbe jól illeszkedik egymásra.

Az eredeti szenzor adatsorában található, 1,25%-hoz tartozó kiugró érték oka, hogy a kézi adagolás miatt nem egyenletesen oszlott el a folyadék a mérés ezen lépésénél.

Megállapítható, hogy a meredekebb éleket tartalmazó vizsgálojel szélesebb feszültségtartományt rendel a 0-16 % nedvességtartalomhoz, viszont a görbe jellegét tekintve nincs szignifikáns változás.

Nedvességtartalom [%]	Feszültség ("kontroll" szenzor) [mV]	Feszültség (kísérleti szenzor) [mV]
0	2563	2627
1,25	2562	2626
2,5	2100	2223
3,75	1893	2205
5	1555	2105
6,25	1510	2093
7,5	1452	2015
8,75	1410	1950
10	1375	1780
11,25	1195	1643
12,5	1152	1595
13,75	1144	1590
15	1145	1590
16,25	1144	1590
17,5	1143	1589
18,75	1134	1587



A grafikonon látható trendvonalak (szaggatott) másodfokú polinomiális görbék. A módosított szenzorhoz tartozó karakterisztika keskenyebb feszültség tartományt rendel a nedvességtartalomhoz és az első 0-10%-hoz tartozó „lineáris” szakaszának meredeksége is kisebb az eredeti szenzorhoz képest.

$$s_e = \frac{V_{e0} - V_{e9}}{S_{e0} - S_{e9}} = \frac{2563 - 1375}{0 - 10} = -118,8 \left[\frac{\text{V}}{\%} \right]$$

$$s_m = \frac{V_{m0} - V_{m9}}{S_{m0} - S_{m9}} = \frac{2627 - 1780}{0 - 10} = -84,7 \left[\frac{\text{V}}{\%} \right]$$

2.2.3 Következtetés

3 Hardver tervezés

3.1 Elvi kapcsolási rajz

3.1.1 Mágnesszelep működése elve

A kliensek kimenete egy az öntözőrendszerben meglévő beavatkozó szervhez kapcsolódik, mely működési elvének megértése szükséges a megfelelő vezérlőjel előállításához.

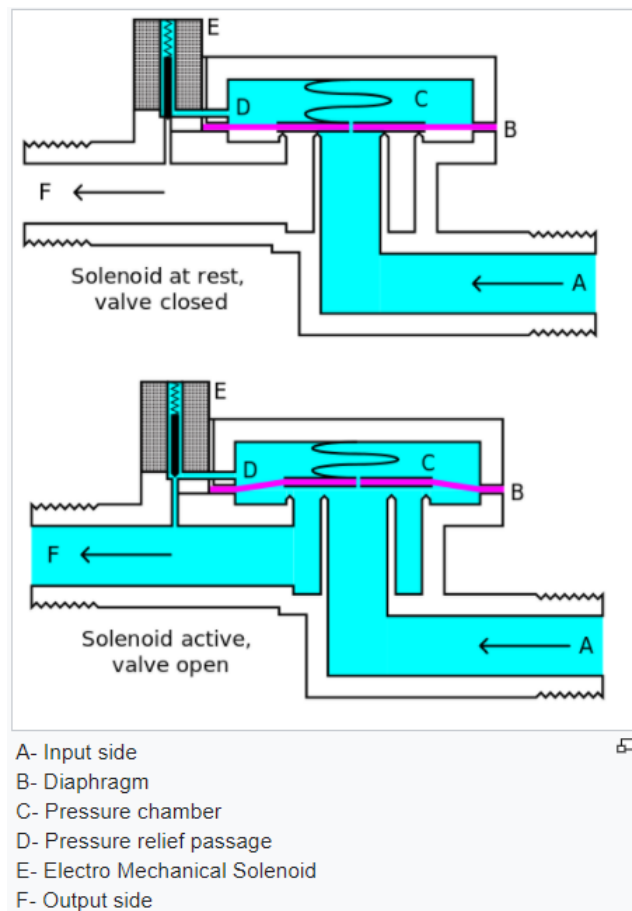
Ez a beavatkozószer egy mágnesszelep, amely elektromechanikusan működtetett szelep. A mágnesszelepek a folyadékokat szállító rendszerekben leggyakrabban használt vezérlőelemek. Számos alkalmazási területen megtalálhatók. Feladatuk a folyadékok útjának elzárása, felszabadítása, a folyadék adagolása, elosztása vagy keverése. A mágnesszelepek gyorsak és biztonságosak, nagy megbízhatóságúak, hosszú élettartammal rendelkeznek.

A 3-1. ábra egy általános szelep kialakítását mutatja, amely a víz áramlását szabályozza ebben a példában. A legfelső ábrán a szelep zárt állapotban van. A nyomás alatt levő víz A pontba kerül. B egy rugalmas membrán, fölötte pedig egy gyenge rugó található, amely lenyomja a membránt. A membrán középpontjában lyuk található, amely lehetővé teszi, hogy nagyon kis mennyiségű víz áramoljon rajta keresztül. Ez a víz úgy tölti ki a membrán másik oldalán lévő C üreget, hogy a nyomás a membrán mindkét oldalán egyenlő legyen, azonban az összenyomott rugó összességében lefelé irányuló erőt szolgáltat. A rugó gyenge, és csak azért képes bezárni a beömlőnyílást, mert a membrán mindkét oldalán kiegyenlítődik a víznyomás.

Amint a membrán bezárja a szelepet, az alja kimeneti oldalán lévő nyomás csökken, és a fenti nagyobb nyomás még szilárdabban zárja. Így a rugónak nincs jelentősége a szelep zárva tartása szempontjából.

Mindez azért működik, mert a kis D lefolyójáratot egy csap rögzíti, amely az E mágnesszelep armatúrája, amelyet egy rugó nyom le. Ha az áram áthalad a mágnesszelepen, a csapot mágneses erővel visszahúzza, ekkor a C kamrában lévő víz gyorsabban üríti ki a D járatot, mint amennyit a lyuk képes feltölteni. A C kamrában a nyomás csökken, a beérkező nyomás megemeli a membránt, ezáltal kinyitva a fő szelepet. A víz a mágnesszelep inaktiválódásáig közvetlenül A-ból F-be áramlik.

Amikor a mágnesszelep inaktíválódik, a D járat ismét záródik, ekkor a rugónak nagyon kevés erőre van szüksége a membrán újbóli lenyomásához, a főszelep bezárul. A gyakorlatban gyakran nincs külön rugó; a membrán úgy van kialakítva, hogy rugóként is funkcionáljon. [3]



3-1. ábra Mágnesszelep működése

Ez a kialakítás normál állapotában zárt (NC), amely ebben az alkalmazásban előnyös, hiszen energiát, csak nyitott állapot közben fogyaszt. Egy öntözőrendszer egy nap során legtöbbet zárt állapotban van, mivel nem öntözünk folyamatosan.

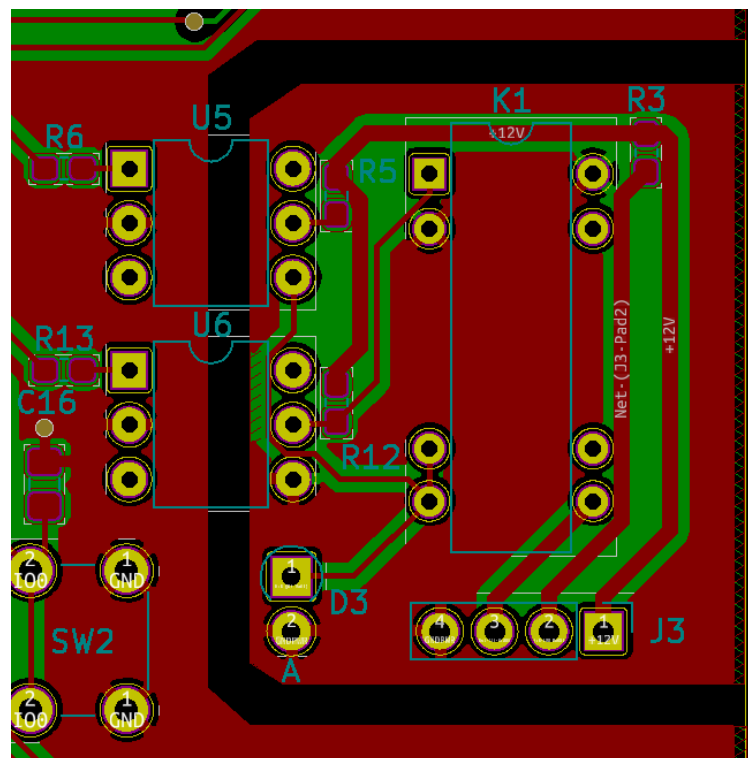
3.1.2 Mágnesszelep vezérlése

A mágnesszelep működési elvéből kitalálhatjuk, hogy villamos vezérlés szempontjából egy tekercset (vagy gondolkozhatunk úgy is, hogy egy elektromágnest) kell energizált állapotba hozni, majd kikapcsolni. A rendszer olyan szelepeket tud kezelni, melyeket elegendő DC feszültséggel ellátni, így a tekercs energizálásához nem szükséges szinuszos vezérlő jelet előállítani. A méretezés ebben az esetben 12 V-ról működő és maximum 6 W-ot fogyasztó szelep alapján történt.

A kliensek alacsony fogyasztású mérőeszközök, nem rendelkeznek 12 V-os tápfeszültséggel, viszont 12 V-os táp csatlakoztatásával vezérelhető lesz a mágnesszelep.

A 12 V-os táp galvanikusan leválasztott a kliensek akkumulátorról előállított feszültségeitől. A leválasztást optikai csatoló áramkörrel és külön PWRGND rézréteg segítségével realizáltam, az IPC2221 szabvány által definiált minimumtávolság többszörös betartásával. (3-2. ábra)

(HIVATKOZÁS) <https://www.smpspowersupply.com/ipc2221pcbclearance.html>



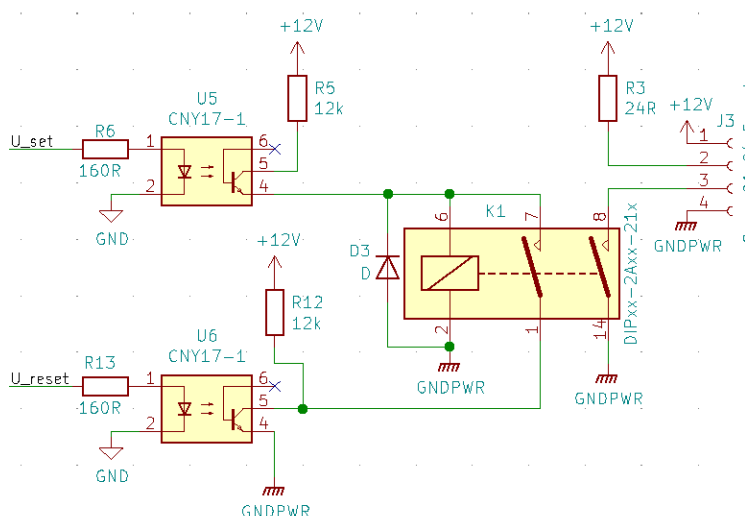
3-2. ábra Galvanikusan leválasztott öntartó relé kapcsolás realizációja

A 3-3. ábra tartalmazza az elvi kapcsolási rajzt. Az áramkörnek két bemenete van U_{set} és U_{reset} , melyek a be- és kikapcsolásért felelnek. A be- és kikapcsoló vezérlőjelek azonosak, legalább 6 μs széles 3.3 V feszültségű impulzusok. A R_6 , R_{13} soros ellenállások áramkorlátozó szereppel rendelkeznek, 2 mA a maximális áram.

A bekapcsoló impulzus hatására U5 kapcsolón keresztül 1 mA-es áram folyik K1 relé tekercsén, mely hatására K1 NC kontaktusai bezárnak és J3 csatlakozón keresztül maximum 500 mA áram mellett a mágnesszelep tápot kap.

A bekapcsoló impulzus után az áramkör öntartásban marad K1 relé 1-7 kontaktusán keresztül, mindaddig amíg a relé tekercsének potenciálját a kikapcsoló impulzus hatására U6 kapcsoló 0 V-ra nem húzza.

A K1 relé tekercsének kikapcsolásakor figyelembe kell venni, hogy a tekercs mágneses terében tárolt energia kontrollált módon disszipálódjon el. D3 teljesítménydióda ezt a feladatot látja el, kikapcsolás esetén a tekercs a diódán keresztül disszipálja el energiáját.



3-3. ábra Galvanikusan leválasztott öntartó relé kapcsolás

A 12V-os külső táp J3 csatlakozón keresztül kerül a NYÁK-ra, így szükség esetén más DC feszültséget is kapcsolathatunk a kliens eszközökkel, például 24 V-ot. Ekkor az ellenállás értékek változtatása szükséges lehet. A huzalok szélessége és egymástól való távolsága megengedi 3.3 V-tól 24 V-ig terjedő külső feszültség használatát.

3.1.3 Kliens tápellátása

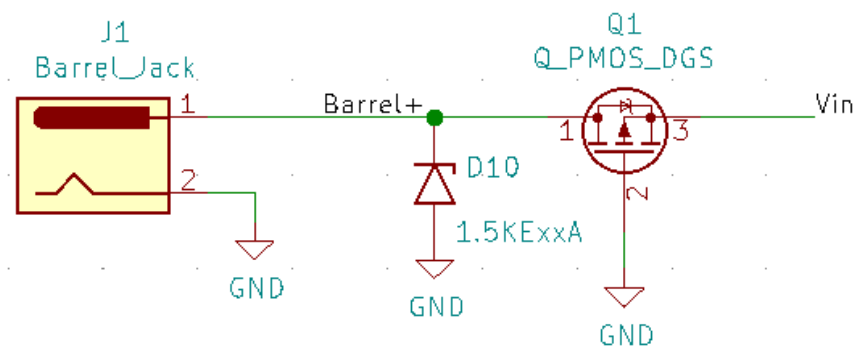
A kliensek mérési helyszíne nem teszi lehetővé a kényelmes csatlakozást a villamos hálózathoz. Mindenképpen telepes vagy akkumulátoros tápellátásra van szükség.

Alacsony fogyasztású alkalmazásoknál a telepes tápellátás nagy előnyökkel bír, hiszen technológiai előnye az akkumulátorokkal szemben, hogy elhanyagolható az önkisülési jelenség. Viszont egy tipikus CR2032 telep 100 mA-es csúcsnál nagyobb áram leadására nem képes, emiatt nagy méretű hidegítő kondenzátorokra (akár szuper kondenzátorokra) lenne szükség, hogy az ESP32 vezeték nélküli kommunikációja közben elegendő áramot tudjon szolgáltatni egy ilyen cella. Mindemellett 200-300 mAh-s tárolókapacitása, sem lenne elég a kliens ellátására.

Az viszonylag magas áramleadási szükséglet és nagy 2000-3000 mAh-s tárolókapacitás érdekében Li-ion akkumulátor alkalmazása mellett döntöttem.

A választás egy 18650-es és 3.7 V nominális feszültséggel rendelkező akkumulátorra esett, melynek a kapacitása 2600 mAh és maximális áramterhelhetősége 10A. A 18650 elnevezés az akkumulátor felépítésének dimenzióit adja meg, emiatt legalább 6,5 cm (praktikusan 8 cm) helyre van szükség a NYÁK-on az akkumulátornak. A helyigény mellett az akkumulátor beszerzési ára a legnagyobb hátránya ennek a megoldásnak, hiszen az körülbelül az alkatrészek árának $\frac{1}{4}$ része. (A hardveres költségek összegzését külön alfejezet tárgyalja.)

3.1.4 Bemeneti feszültség védelmei



3-4. ábra ESD és fordított polaritás védelem

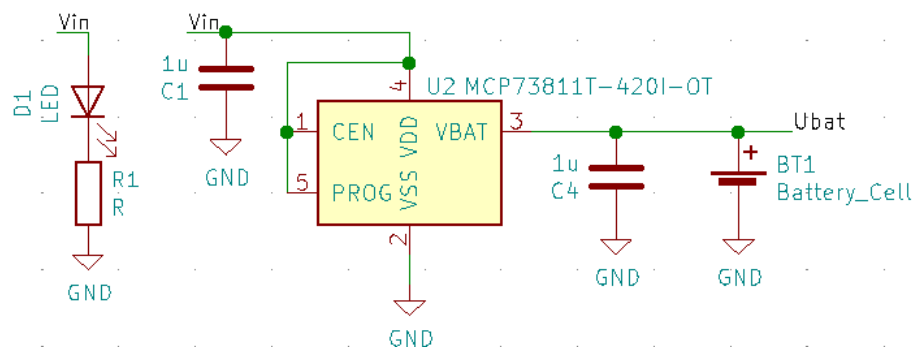
A 3-4. ábra a bemeneti feszültségre alkalmazott ESD és fordított polaritás védelem áramkörének kapcsolási rajzát tartalmazza. Az ESD védelmet TVS dióda látja

el. Amennyiben J1 csatlakozón fordított polaritású feszültség jelenik meg Q1 P csatornás MOS-FET szakadásként viselkedik, ezáltal védve az áramkört.

Ennek bizonyítására végezzünk gondolatkísérletet. Tegyük fel, hogy Q1 kapcsoló zárt állapotban van, ekkor V_{in} GND-hez képest negatív feszültség. Egy P csatornás FET csak negatív gate-source feszültség (U_{GS}) mellett van zárt állapotban. Viszont V_{in} GND-hez viszonyítva pontosan $-U_{GS}$, tehát negatív V_{in} feszültség esetén U_{GS} pozitív, melyből következik, hogy a tranzisztor nyitott állapotban kell legyen.

Összegezve Q1 tranzisztor csak helyes bemeneti feszültség mellett lehet nyitott állapotban, U_{DS} maximális feszültségig védi az áramkört a fordított polaritás ellen.

3.1.5 Akkumulátor töltő áramkör



3-5. ábra Li-ion akkumulátor töltő áramkör

Az akkumulátor töltését egy single-cell (egy cellás) Li-ion akkumulátor töltő integrált áramkör látja el. Az akkumulátort konstans áramú töltés mellett 4.2 V feszültségig tölti az áramkör. Az IC PROG lábának logikai magas feszültségre kötésével 500 mA-es töltőáram lett beállítva. Így az akkumulátor $\frac{2600 \text{ mAh}}{500 \text{ mA}} = 5.2h \sim 5h$ alatt tölthető fel. Azáltal, hogy töltésre ritkán van szükség és az akkumulátor kevesebb, mint egy éjszaka alatt feltölthető, így elfogadható a töltési idő.

C_1 és C_4 kondenzátorok a be- és kimeneti feszültségek hullámosságának csökkentésére és impulzuszerű áramfelvételek töltés szükségletének ellátására szolgálnak. Más szempontból nézve passzív RC szűrőként funkcionálnak a tápvonalakon.

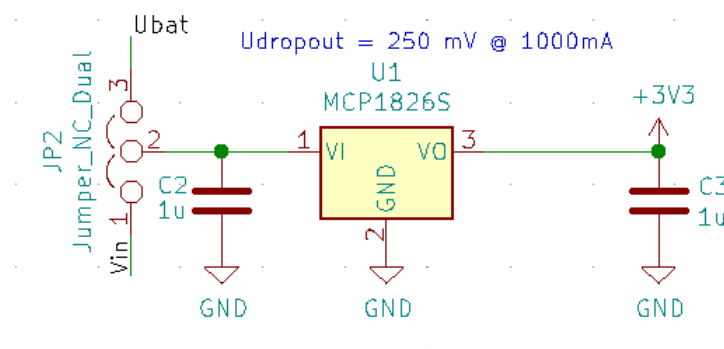
3.1.6 Szabályozott tápfeszültség előállítása

Az ESP32 és egyéb a kliens részét képező IC-k számára a tápfeszültséget egy 1000 mA-t leadni képes Low Dropout Voltage lineáris ún. áteresztő tranzisztoros szabályozó látja el. LDO-ra szükség az akkumulátor feszültség nominális értékének és 3.3 V-os előállítandó feszültségnek közelsége és a felesleges hőfejlesztés elkerülése miatt volt szükség.

C₂ és C₃ kondenzátorok szerepe megegyezik a 3.1.5 pontban szereplő C₁ és C₄ kondenzátorokkal.

A feszültségszabályzó bemenetére jumper segítségével külső tápfeszültséget is lehet csatlakoztatni, de ennek szerepe szigorúan fejlesztési célú.

A lineáris feszültségszabályozókat nagyfrekvenciás zajok ellen védeni kell, ennek gyakori megoldása egy a szabályzóval sorba kötött L-C szűrő. A szűrésre, azért van szükség, mivel a szabályzóban lévő tranzisztor nem képes végtelen sávszélességgel kapcsolni és a többletfeszültséget eldisszipálni. Azonban a Li-ion akkumulátorok feszültsége kevés zajt tartalmaznak, eleve jól szűrt tápforrásnak tekinthetők. Emellett még passzív LC szűrő alkalmazásához is szükség van egy viszonylag drága áramköri elemre, a tekercsre. A felsorolt érvek miatt (kis zajtartalmú forrás, költségminimalizálás) nem került szűrő az LDO bemenetére.



3-6. ábra LDO fix 3.3 V kimeneti feszültséggel

A kimeneten található 1µF-os kerámia kondenzátor mellett minden IC táplábaihoz közel 100 nF-os kondenzátor van párhuzamosan kötve C₃ kapacitással. Ennek segítségével növelve a kondenzátorok effektív sávszélességét.

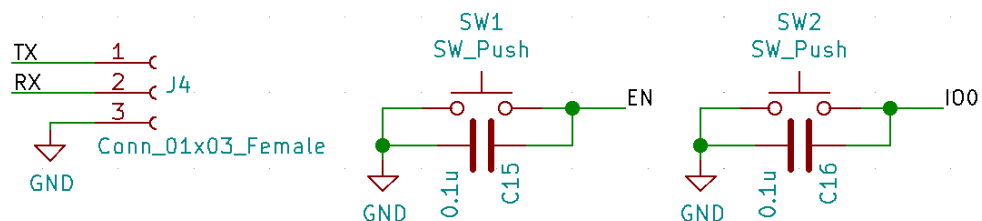
3.1.7 Programozó interface

Az ESP32 programozása UART periférián keresztül történik. Ez az UART periféria nem csak a programozhatóságot teszi lehetővé, hanem debug portként is használható fejlesztés közben, vagy javítások elvégzésénél. (Hibaüzenetek küldhetők rajta.)

Az ESP32 modul programozható állapotba hozásához szükséges a processzor IO0 lábát alacsony logikai szinten tartani, erre nyomógomb segítségével van lehetőség. Programozás után hardveres reset szükséges, melyet EN jel logikai alacsony szintre húzásával érhetünk el, nyomógomb segítségével.

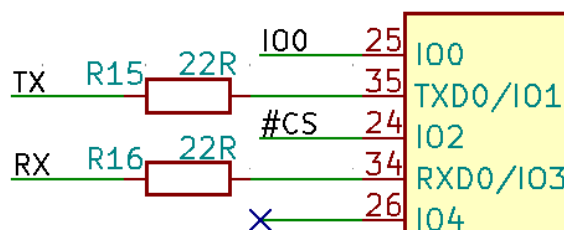
A nyomógombok pergésmentesítésére az ESP32 Dev Kit C adatlapja által javasolt megoldást alkalmaztam. **(HIVATKOZÁS)**

A programozó interface és a pergésmentesített nyomógombok kapcsolási rajzát a 3-7. ábra tartalmazza.



3-7. ábra Programozó interface

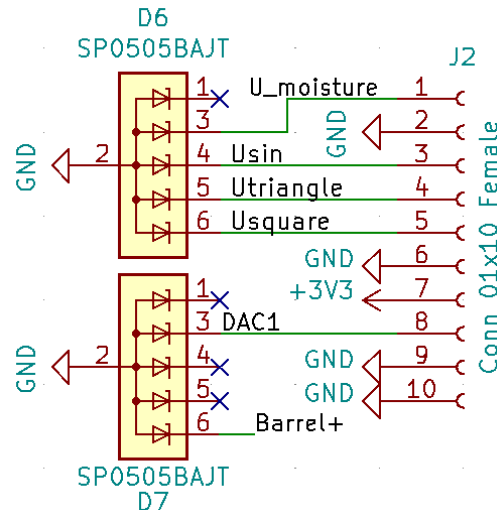
Kihasználva a NYÁK-on megjelenő parazitakapacitásokat, kis értékű $22\ \Omega$ értékű ellenállások felhasználásával TX és RX jelek meredek éleit szűröm, ezzel csökkentve az általuk a NYÁK-on keltett zajt.



3-8. ábra Digitális jel szűrőáramkör

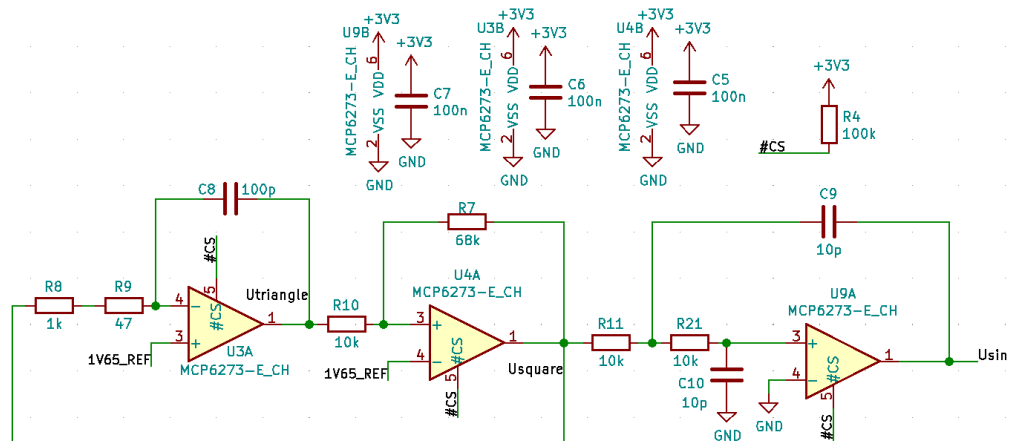
3.1.8 Méréshez szükséges csatlakozó

A méréshez szükséges gerjesztőjelek és a mért jel, 3.3 V és GND jelekkel kiegészítve tüskesor csatlakozón található a NYÁK-on. A mérővezetékek hossza, illetve a külvilággal való rendeltetésszerű kapcsolata miatt ESD védelemmel láttam el a tüskesor csatlakozót, TVS dióda tömb alkalmazásával.



3-9. ábra Tüskesor csatlakozó ESD védelme

3.1.9 Mérőjel előállító áramkör



3-10. ábra Mérőjel előállító áramkör

A 3-10. ábra tartalmazza a méréshez szükséges 1.5 MHz-es jeleket előállító áramkört. A méréshez DC komponens is tartalmazó, négyszögjelet és szinuszjelet állítunk elő.

Az áramkör három alapkapcsolásból, egy invertáló integrátorból, egy Schmitt-triggerből és egy Sallen-Key szűrőből áll, melyek együttesen hozzák létre a kívánt jeleket.

A 3-10. ábra által ábrázolt kapcsolásban a létrehozott négyszögjel kerül szűrésre, egy Sallen-Key kapcsolás másodfokú aluláteresztő kapcsolás alkalmazásával.

Mindhárom alapkapcsoláshoz lineáris műveleti erősítőre van szükség, mellyel szemben a következő feltételeket határoztam meg:

- legalább 1.5 MHz működési sávszélesség
- aszimmetrikus táp lehetősége
- rail-to-rail működés
- alacsony fogyasztás

A választás a MCP6273 gyártói jelű integrált áramkörre esett, mely egy 170 μ A fogyasztási móddal és aszimmetrikus táppal rendelkező 2 MHz sávszélességű rail-to-rail műveleti erősítőt tartalmaz. **(HIVATKOZÁS)**

A méretezés során figyelembe vettem, hogy 0603 és 0508 méretekkkel rendelkező SMD alkatrészekkel történik a megvalósítás, illetve az E24-es szabványsor szerinti értékek érhetők el.

Az erősítő engedélyezéséhez egy aktív alacsony jelet kell a mikrovezérlőnek előállítani. A \overline{CS} lába az IC-nek 100 k Ω ellenálláson keresztül 3.3 V feszültségre van kötve. A mérőjel frekvenciáját meghatározó RC-tag a 3-10. ábra C8, R8 és R9 elemeiből tevődik össze.

A frekvencia a következő egyenlet szerint számítható:

$$f = \frac{1}{2\pi} \cdot \frac{1}{(R_8 + R_9) \cdot C_8} \cong 1520104.52 \text{ Hz} \approx 1.5 \text{ MHz}$$

A komparátor (Schmitt-trigger) billenési feszültségét a következő módon származtathatjuk: **(HIVATKOZÁS)**

$$IN_+ = U_{ref}$$

$$IN_+ = U_{triangle} \cdot \frac{R_7}{R_{10} + R_7} + U_{square} \cdot \frac{R_{10}}{R_{10} + R_7}$$

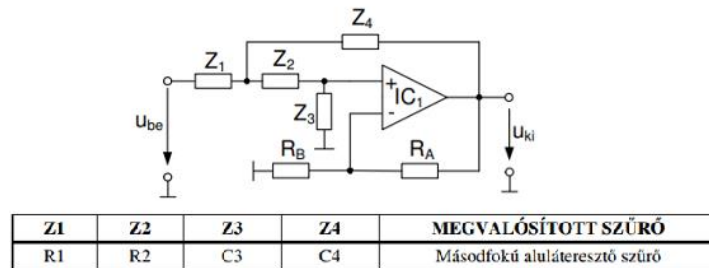
$$U_{billenési} = \left(1 + \frac{R_{10}}{R_7}\right) \cdot U_{ref} - \frac{R_{10}}{R_7} \cdot U_{square}$$

,ahol $U_{ref} = 1.65 \text{ V}$, $R_7 = 68 \text{ k}\Omega$ és $R_{10} = 10 \text{ k}\Omega$

Ezek alapján a billenési feszültségek: $U_1 \cong 1.41 \text{ V}$ $U_2 \cong 1.89 \text{ V}$

A bevezetett hiszterézis ugyan a háromszögjel szimmetriáján ront, viszont biztosítja, hogy nem történhet oszcillálás a komparátor kimenetén.

A szinuszos mérőjel előállításához Sallen-Key alaptagot alkalmaztam, mellyel másodfokú aluláteresztő szűrőt valósítok meg. **(HIVATKOZÁS)**

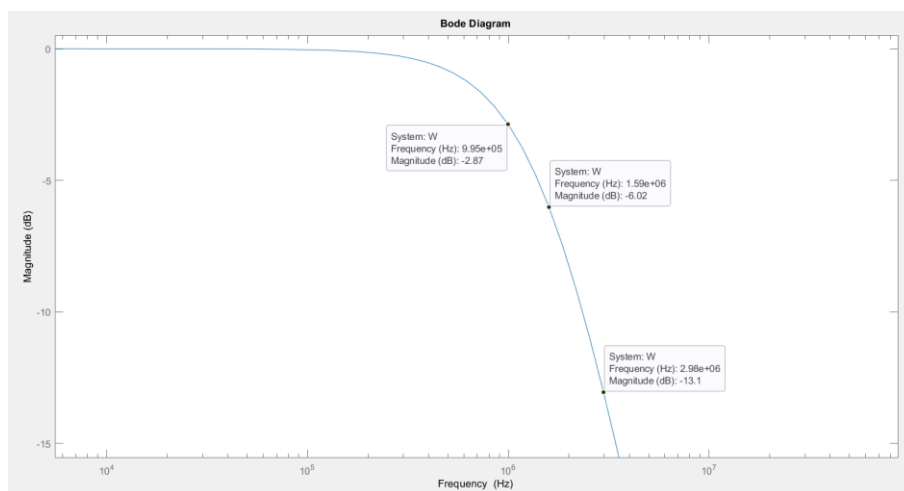


3-11. ábra Sallen-Key alaptag

A 3-10. ábra szerinti kapcsolásban R₁ szerepét R₁₁, R₂-t R₂₁, C₃-t C₁₀ és C₄-t C₉ veszi fel.

A kapcsolat átvitele a következő:

$$\frac{U_{ki}(s)}{U_{be}(s)} = \frac{1}{s^2 + s\left(\frac{1}{R_{21}C_9} + \frac{1}{R_{10}C_9}\right) + \frac{1}{R_{11}C_{10}R_{21}C_9}}$$



3-12. ábra Sallen-Key szűrő Bode-diagramja

Az előzőekben ismertetett tervezési folyamat a műveleti erősítő aszimmetrikus táplására miatt hibás. Erre a hibára a NYÁK bemérése során derült fény, melynek okát a **(HIVATKOZÁS)** fejezetben tárgyalom.

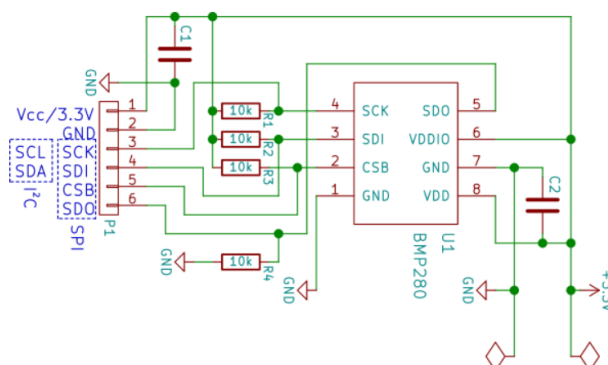
3.1.10 Hőmérséklet- és nyomásmérő áramkör

Az öntözőrendszer számára fontos, hogy a környezeti hőmérsékletet (elsősorban talajközeli levegőréteg hőmérsékletét) legalább 1 °C pontossággal mérje, annak érdekében, hogy fagy vagy fagyközeli levegőhőmérséklet esetén, ne kapcsolja be az öntözést, még a talaj nedvességtartalmának csökkenése esetén sem.

A levegő nyomásának mérésével következtethetünk az időjárás változására például a nyári záporok előtt nyomásváltozás történik. **(HIVATKOZÁS)**

Erre a feladatra a BMP280 elnevezésű szenzort használtam. Amellett, hogy megfelelő pontossággal mér hőmérsékletet és nyomást, kapható ún. breakout boardon is, így külön kis NYÁK-ra került a szenzor és a nagyáramot is szállító NYÁK hőmérséklete kevésbé befolyásolja a hőmérséklet mérést.

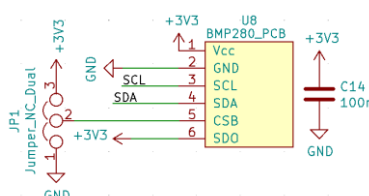
A szenzort I2C (opcionálisan SPI) buszon lehet elérni 0x76 és 0x77 címen, melyen hardveresen lehet állítani, egy jumper segítségével. A kapcsolás belső felépítését a 3-13. ábra, a kliens NYÁK-hoz csatlakozását a 3-14. ábra mutatja.



3-13. ábra BMP280 PCB belső felépítése

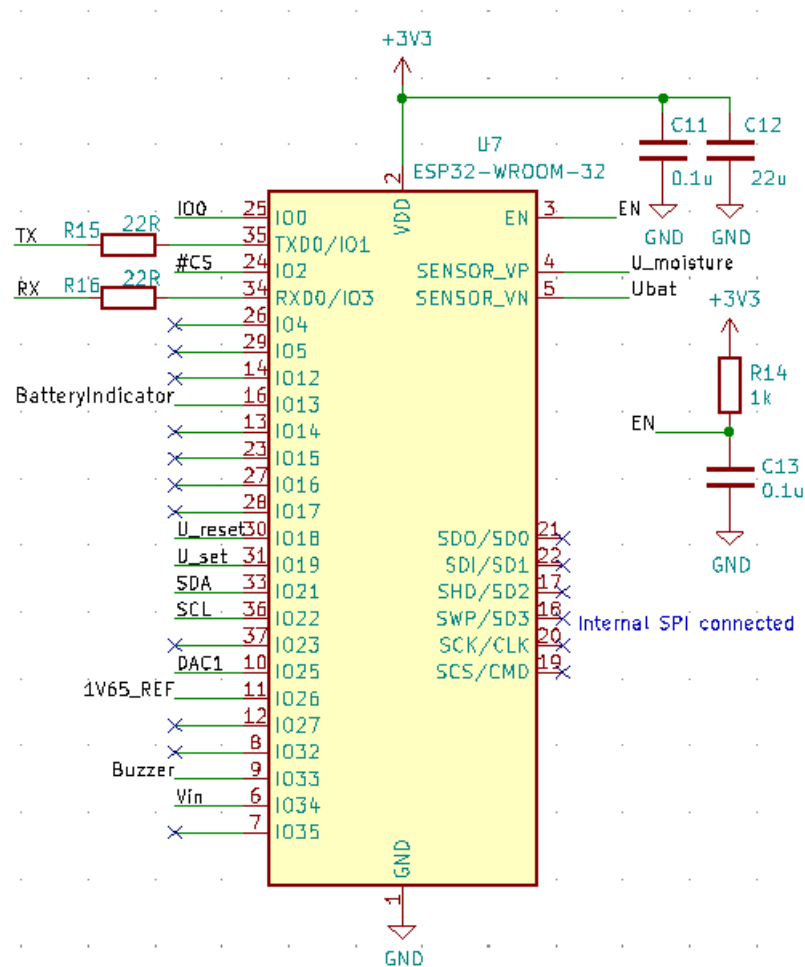
<https://startingelectronics.org/pinout/GY-BMP280-pressure-sensor-module/>

(HIVATKOZÁS)



3-14. ábra BMP280 PCB külső áramköre

3.1.11 ESP32 WROOM-32 periféria áramkörei



3-15. ábra ESP32 WROOM-32

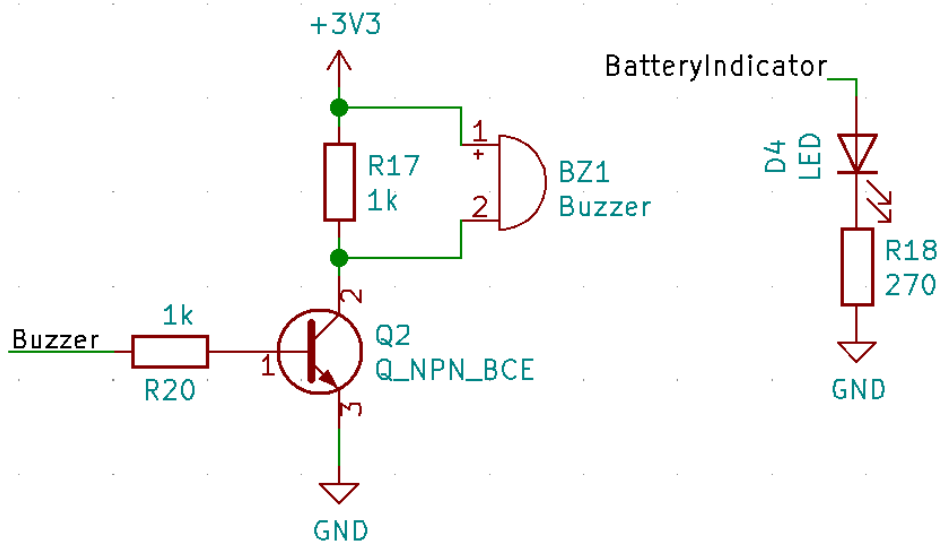
Az ESP32 modul ezen alkalmazáshoz szükséges periféria áramkörök nagy részét tartalmazza, így nem volt szükség sok külső alkatrész csatlakoztatásához.

A 3-15. ábra C₁₁ és C₁₂ hidegítő kondenzátorok a gyártó által javasolt méretekkel rendelkeznek.

R₁₄ és C₁₃ az engedélyező jel (EN) szűrését látják el.

Az általam használt WROOM-32 modul tartalmaz 4 MB flash memóriát, amely SPI buszon keresztül összeköttetésben áll a processzorral. Emiatt a modul 17-22 lábait nem szabad semmilyen külső egységhez kötni.

3.1.12 Egyéb funkciókat ellátó áramkörök



3-16. ábra Buzzer és LED meghajtó áramkörök

A 3-16. ábra tartalmazza a keresési funcióhoz szükséges buzzer meghajtó áramkört, illetve az akkumulátor indikátor áramkört. A LED 5 mA-es árammal és 2 V nyitófeszültséggel lett méretezve. A buzzer áramkört 4 kHz közötti négyszögjellel kell meghajtani.

3.2 Nyomtatott huzalozású áramköri lemez terv

3.2.1 Prototipizálás

A klienst alkotó áramkört nem prototipizáló panelen, hanem nyomtatott huzalozású lemezen készítettem el, annak érdekében, hogy a feszültség mérés pontossága minél nagyobb és a méréshez szükséges „nagy” frekvenciás jelek zajterhelése minél kisebb legyen.

A tervezés közben prototípusként kategorizáltam az elkészítendő NYÁK-ot, ezért kivétel nélkül minden csatlakozó tűske- és hüvelysor, melyek az olcsó és gyors fejlesztést könnyítették meg. A szakdolgozatban összefoglalt tanulságok alapján ennek a NYÁK-nak második verziója elkészítésekor már mindenképpen figyelembe kell venni a külvilág fel való csatlakozás gyakorlati kényszereit is.

A NYÁK terv az UniPCB kft. legolcsóbb gyártási technológiájának megfelelő méreteket tartalmazza, melyek betartásához két huzal osztály rendeltem ().

Net Classes								
Name	Clearance	Track Width	Via Size	Via Drill	μVia Size	μVia Drill	dPair Width	dPair Gap
Default	0.25 mm	0.25 mm	0.9 mm	0.7 mm	0.3 mm	0.1 mm	0.25 mm	0.25 mm
Power	0.25 mm	0.3048 mm	1.5 mm	1 mm	0.3 mm	0.1 mm	0.25 mm	0.25 mm

3-17. ábra Huzal osztályok

4 Szoftver tervezés

4.1 Kliens beágyazott szoftvere

4.2 Központi egység szoftvere

4.2.1 SWAN szerver szoftveres környezete

A SWAN szerver egy Raspberry Pie Zero W single board computeren fut. A RPI hivatalos operációs rendszerét DietPi szoftverre cseréltem, annak érdekében, hogy minél kevesebb erőforrást használjon el az operációs rendszer és több maradjon az MQTT broker futtatására és a webes felület kiszolgálására.

<https://www.raspberrypi.org/products/raspberry-pi-zero-w/?resellerType=home>

<https://dietpi.com/dietpi-software.html>

A teljes rendszerhez való hozzáférés SSH-n keresztül vagy monitor és USB billentyűzet csatlakoztatásával lehetséges. Ezen felül az eszköznek kiosztott IP címen keresztül különböző telepített szoftverek kezelőfelületei is elérhetőek lokális hálózatról.

A teljes szoftveres felépítéshez egy publikusan elérhető SD kártya image-t használtam fel, amely több használt alkalmazás szoftverét előre telepített és „élesztett” állapotban tartalmazta.

<https://www.sensorsiot.org/diet-pi-supporting-material-videos-126-and-128/>

4.2.2 MQTT broker

Az MQTT protokoll ideális IoT rendszerek adatainak közvetítésére, mivel kis erőforrás és kód igényű, megbízható és kétirányú kommunikációt valósít meg.

Az MQTT Publish/Subscribe architektúrát leíró protokoll, amelyben alap esetben egy broker és több kliens közötti kommunikációt tesz lehetővé.

Minden üzenet küldés akár egy kliens, akár a bróker küldi egy adott témába (topic) kerül publikálásra. A kliensek feliratkoznak egyes témákra, amelyet a bróker jegyez és abban az esetben, ha arra üzenet érkezik minden feliratkozott kliensnek közvetíti. A brókeren keresztül megy minden üzenet, nincs lehetőség kliens-kliens közvetlen kapcsolatra.

A választásom a brókert megvalósító szoftverre, a mosquitto mqtt broker, amely egy nyílt forráskódú lightweight megoldás.

<https://mosquitto.org/>

4.2.3 SQLite

A kliensek felől érkezett adatokat tárolni kell a szerveren, hogy azokat feldolgozhassuk, illetve megjelenítsük a felhasználó számára. Nagy mennyiségű adat rendszerezett tárolására az egyik legelterjedtebb módszer a relációs adatbázisok alkalmazása. Az adatbázisokhoz való hozzáférés szabványosan Structured Query Language (SQL) nyelv segítségével tehetjük meg.

A SWAN szerver SQLite adatbázis motort alkalmazza, mely nyílt forráskódú lightweight megoldást nyújt. Lekérdezések mellett böngészőben megnyitott felületen is van lehetőségünk elérnünk az adatbázis kezelő motort. Ilyen módon került létrehozásra az a két tábla (SWAN,serialnumbers), amelybe adatokat szűr be a szerver működése során.

```
CREATE TABLE 'SERIALNUMBERS' ('key' INTEGER PRIMARY KEY NOT NULL, 'timestamp' DATETIME DEFAULT CURRENT_TIMESTAMP)
```

```
CREATE TABLE 'SWAN' ('primkey' INTEGER PRIMARY KEY NOT NULL, 'id' INTEGER, 'moisture' REAL, 'temperature' REAL, 'pressure' REAL, 'status' INTEGER, 'battery' REAL, 'timestamp' DATETIME DEFAULT CURRENT_TIMESTAMP)
```

<https://www.sqlite.org/index.html>

https://vik.wiki/images/f/f0/Info2_Adatb%C3%A1zisok_jegyzet_%28r%C3%A9gi%29.pdf

4.2.4 Node-RED

A SWAN szerver három fő feladatot lát el, felhasználói felületet biztosít a mért adatok megtekintéséhez és a kliensekhez való hozzáférésnek, MQTT brókert futtat és adatbázisban tárolja a mért adatokat.

A három fő funkció összefésülésére egy IoT rendszerekben gyakran használt programozási eszközt használtam, a Node-RED alkalmazást.

<https://nodered.org/>

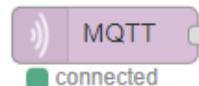
Az alkalmazás lehetővé teszi, hogy úgynevezett node-k összekötésével egyszerű és átlátható módon készítsünk backend-et és frontend-et összekötő szoftvert. Lehetővé teszi, hogy adatáramlást figyelve programozzunk, magas absztrakciós „nyelven”. Node.js alapon működik, amely egy esemény vezérelt modellű lightweight szoftverrendszer.

<https://nodejs.org/en/>

Az alkalmazást, úgynevezett flow-t, böngészőben elérhető grafikus felületen érhetjük el. A következőben ismertetem a SWAN szerver elkészült szoftverében szereplő node-okat általánosan, majd külön alfejezetben a konkrét programot.

MQTT input node:

Feladata a MQTT brókeren keresztül érkező üzenet fogadása egy megadott témában. Tekinthejtük úgy, hogy a node-red flow így iratkozik fel egy témára. Legfontosabb kimenete, msg.payload amely az üzenetet tartalmazó karakterlánc.



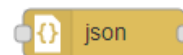
MQTT output node:

Kimenetként az MQTT bróker felé továbbít egy üzenetet egy megadott témával. Tekinthejtük úgy, hogy a node-red flow így publikál egy adott témában. Legfontosabb kimenetei; msg.payload és msg.topic. A node beállításakor a Quality of Service és Retain flag beállítására is ügyelnünk kell.



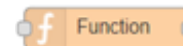
JSON node:

JSON objektumot karakterláncá vagy karakterláncot JSON objektummá alakít. msg.payload attribútumon kívül mást nem módosít.



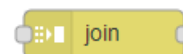
Function node:

Bármilyen javascript programot tartalmazhat ez a node. Kimenetét return kulcsszóval vagy node.send() függvénnyel küldheti tovább a következő node-nak. A kimeneti üzenet a node-red konvenciónak megfelel, amennyiben a függvény blokk nem állítja a kapott msg attribútumait azokat tovább adja a következő node számára.

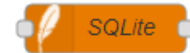


Join node:

Több node által küldött üzeneteket egyesíthetünk ezzel a node-dal. Beállítása szerint az üzenetek száma, időkorlát vagy msg.complete attribútum bebillentése okozza az addigi üzenetek összekapcsolását. Szintén beállítható az egyesítés módja (tömb, kulcs-érték objektum vagy objektum összefonás).

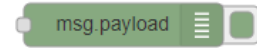


SQLite node:



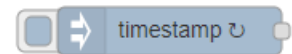
Az adatbázissal kapcsolatot teremtő node. A bemenetére érkezett üzenet témáját (msg.topic) kérdezi le az adatbázistól. Az adatbázis válaszát a kimenetre üzenetként továbbítja (msg.payload).

Debug node:



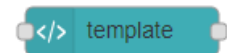
Fejlesztéshez elengedhetetlen, hogy a debug fülön nyomonkövethető legyen az adatáramlás. Ez a node a bemenetét a debug fülre továbbítja, illetve időbélyeggel is ellátja azt. A jobb oldalán található zöld gombbal ki/be kapcsolható hatása, a könnyebb fejlesztés érdekében.

Inject node:



Beállításai szerint induláskor vagy periodikusan vagy a fejlesztő felületen megnyomásakor az aktuális időbélyeget küldi ki üzenetként. Debugolásra vagy timer megvalósításra alkalmas node.

UI template node:



A felhasználói felületet UI node-k reprezentálják. A template node bármilyen HTML állomány fogadására képes. A HTML állományban leírt objektum a node-hoz rendelt dashboard elemen jelenik meg.

Chart node:

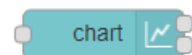
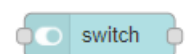


Chart node segítségével egyszerűen hozhatunk létre dinamikusan formálódó kétdimenziós ábrát. A bemenetén kapott Y értékeket automatikusan ábrázolja az aktuális időbélyeg szerint egy kétdimenziós ábrán. A bejövő üzenet témája (msg.topic) külön ábrázolja az adatokat. Állapotát nem őrzi meg, minden újraindítás után üresen indul az ábra. Lehetőség van a Chart.js könyvtár dokumentációja szerinti paraméterezésre, amellyel egyszerre több X-Y pár megadása is lehetséges (pl. régebbi adatok betöltése érdekében). Ezt a lehetőséget a SWAN serveren futó node-red verzióval viszont nem sikerült kihasználni.

<https://www.chartjs.org/>

Dashboard Switch node:

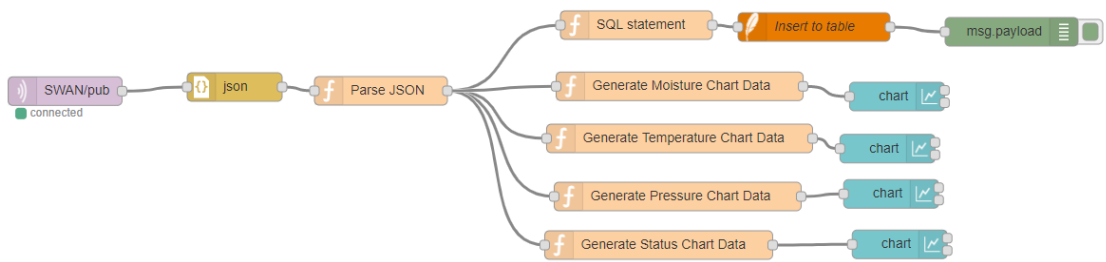


A felhasználói felületen egy kapcsoló jelenik a beállított dashboard elemen. A kapcsolóra való kattintás során a node előre beállított üzenetet továbbít.

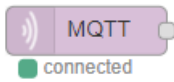
4.2.5 SWAN SERVER flow

A szervert megvalósító node-red flow négy különálló részre bontható. Az adatokat fogadó és ábrázoló részre, új kliensek számára sorszámot szolgáltató részre, a felhasználói felületet kezelő részre és az adatbázist karbantartó/lekérdező részre.

A 4-1. ábra az adatokat fogadó és ábrázoló részt megvalósító node-k közötti kapcsolatok tartalmazza.



4-1. ábra Mérési adatok mentése és ábrázolása



MQTT

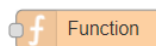
Feladat:	SWAN/pub témájú üzenetek fogadása.
Beállítás:	server: localhost:1883, QoS2 annak érdekében, hogy legfeljebb egyszer fogadjunk minden üzenetet.



Parse JSON

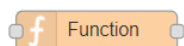
Feladat:	A hálózaton érkező üzenet JSON objektumot karakterlánc formátumra hozta az előző JSON node. A Parse JSON függvény feladata külön változókba való adatmentés, illetve a bizonyos adatok átlagolása és két számjegyre kerekítése.
Kód:	<pre>1 var avgtemp; 2 var avgmoist; 3 var avgpres; 4 var status= msg.payload.status; 5 var id=msg.payload.Client;</pre>

	<pre> 6 var battery = 7 msg.payload.Battery_Voltage; 8 9 avgtemp = (msg.payload.temp[0]+ msg.payload.temp[1]+ msg.payload.temp[2]+ msg.payload.temp[3]+ msg.payload.temp[4])/5; avgpres = (msg.payload.pressure[0]+ msg.payload.pressure[1]+ msg.payload.pressure[2]+ 10 msg.payload.pressure[3]+ msg.payload.pressure[4])/5; avgmoist = (msg.payload.U_moisture[0]+ msg.payload.U_moisture[1]+ msg.payload.U_moisture[2]+ msg.payload.U_moisture[3]+ msg.payload.U_moisture[4])/5; 11 12 13 temp = avgtemp.toFixed(2); 14 pres = avgpres.toFixed(2); 15 moist = avgmoist.toFixed(2); 16 17 var data = {payload: [id,moist,temp,pres,status,battery]}; 18 19 return data; </pre>
--	--



SQL statement (insert payload)

Feladat:	A beérkező adatokból SQL állítás összeállítása.
Kód:	<pre> 1 var topic = " INSERT INTO \"SWAN\" (\"ID\", \"moisture\", \"temperature \", \"pressure\", \"status\", \"battery\") VALUES ('"+msg.payload[0]+"', '"+msg.payload[1] +"', '"+msg.payload[2]+"', '"+msg.payload[3]+ +"', '"+msg.payload[4]+"', '"+msg.payload[5]+"'"); 2 var newmsg={} 3 newmsg.topic = topic; 4 return newmsg; </pre>



Generate Moisture Chart Data

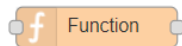
Feladat:	Chart node számára szükséges formátum előállítás.
Kód:	<pre> 1 var msg1 = { payload:msg.payload[1], topic:msg.payload[0] }; </pre>

	2
	3 <code>return msg1;</code>



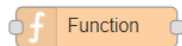
Generate Temperature Chart Data

Feladat:	Chart node számára szükséges formátum előállítása.
Kód:	<pre> 1 var msg1 = { payload:msg.payload[2], topic:msg.payload[0] }; 2 3 return msg1;</pre>



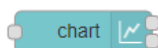
Generate Pressure Chart Data

Feladat:	Chart node számára szükséges formátum előállítása.
Kód:	<pre> 1 var msg1 = { payload:msg.payload[3], topic:msg.payload[0] }; 2 3 return msg1;</pre>



Generate Status Chart Data

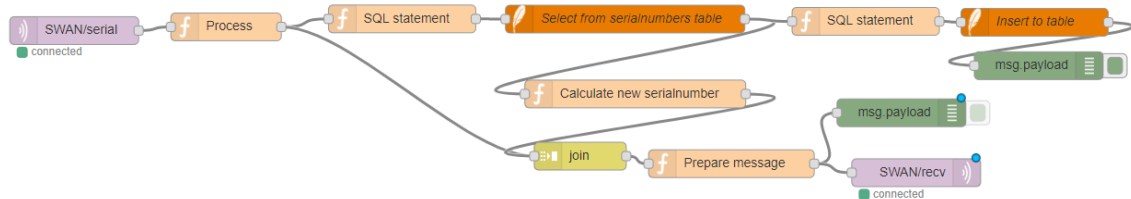
Feladat:	Chart node számára szükséges formátum előállítása.
Kód:	<pre> 1 var msg1 = { payload:msg.payload[4], topic:msg.payload[0] }; 2 3 return msg1;</pre>



Chart

Feladat:	A felhasználói felületen kétdimenziós ábra megjelenítése.
Beállítás:	X-tengely HH:mm:ss formátummal definiált, az elmúlt egy napban érkezett adatokat, de legfeljebb 1000 pontot ábrázolunk.

A 4-2. ábra a kliensek sorszámát generáló részt megvalósító node-k közötti kapcsolatok tartalmazza.

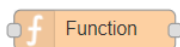


4-2. ábra Kliensek sorszám generálása



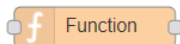
MQTT

Feladat:	SWAN/serial témájú üzenetek fogadása.
Beállítás:	server: localhost:1883, QoS2 annak érdekében, hogy legfeljebb egyszer fogadjunk minden üzenetet.



Process

Feladat:	A beérkezett karakterlánc egy egész számot tartalmaz. Ez a függvény a karakterlánc és egész szám közötti konverziót végzi.
Kód:	<pre> 1 msg.payload = parseInt(msg.payload); 2 3 return msg; </pre>



SQL statement (Select maxkey)

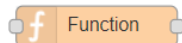
Feladat:	<p>A függvény feladata, hogy SQL node számára előállítsa a SQL lekérést.</p> <p>newmsg.topic tartalmazza az érvényes SQL állítást. A serialnumbers táblából SELECT utasítással a key oszlop értékeit kérjük le. MAX() függvénnyel módosítjuk a lekérést, ezáltal csak az oszlop legnagyobb értékét kapjuk vissza.</p> <p>A létrejövő MAX(key) javascript objektumot hibás szintaxisú, ezért AS kulcsszóval új nevet adunk neki, maxkey, amely már javascript kompatibilis.</p>
-----------------	--

Kód:	<pre> 1 var topic = "SELECT MAX(key) AS maxkey FROM serialnumbers"; 2 3 var newmsg={} 4 newmsg.topic = topic; 5 return newmsg;</pre>
-------------	--



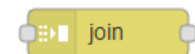
SQL statement (inject newkey)

Feladat:	A beérkező adatokból SQL állítás összeállítása.
Kód:	<pre> 1 var tempnum = msg.payload[0].maxkey+1; 2 serialnumber = tempnum.toString(); 3 4 5 var topic = " INSERT INTO \"SERIALNUMBERS\" (\"key\") VALUES ('"+serialnumber+"')"; 6 var newmsg={} 7 newmsg.topic = topic; 8 return newmsg;</pre>



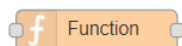
Calculate new serialnumber

Feladat:	A lekérdezés a jelenlegi legnagyobb sorszámot adja vissza. Az új sorszám eggyel nagyobb lesz.
Kód:	<pre> 1 if (msg.payload[0].maxkey === null){ 2 msg.payload = 1;} 3 else{ 4 msg.payload = msg.payload[0].maxkey + 1 5 }; 6 return msg;</pre>



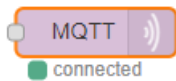
Join

Feladat:	Process és Calculate new serialnumber függvények üzeneteit tömbbe tömöríti.
Beállítás:	Két üzenet beérkezése után üzenet küldése. Üzenet formátuma tömb.



Prepare message

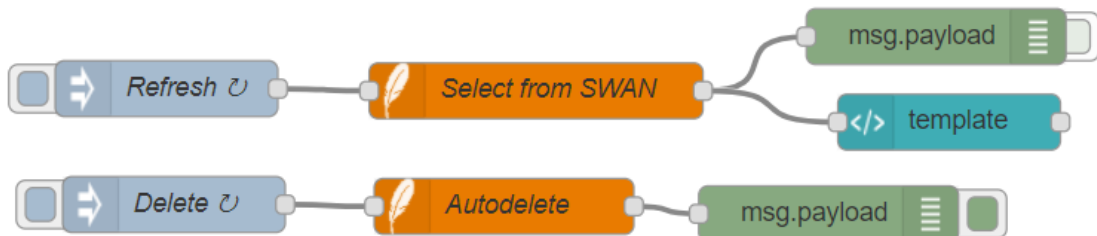
Feladat:	A beérkező tömb két elemét , karakterrel elválasztva karakterláncként tovább küldeni .
Kód:	<pre> 1 msg.payload = msg.payload[0].toString() + "\", " + msg.payload[1].toString(); 2 3 return msg; </pre>



MQTT

Feladat:	SWAN/recv témájó üzenet publikálása. A SWAN SERVER flow-ban több helyen is.
Kód:	server: localhost:1883, QoS1 annak érdekében, hogy legalább egyszer elküldjünk minden üzenetet. Retain:false, mivel fontos elkerülni, hogy egy kliens az energiacsökkentett állapotából visszatérve téves üzenetet kapjon, mert a bróker új feliratkozóként regisztrálja.

A 4-3. ábra az adatbázist karbantartó részt megvalósító node-k közötti kapcsolatok tartalmazza.



4-3. ábra Adatbázis karbantartása



Refresh

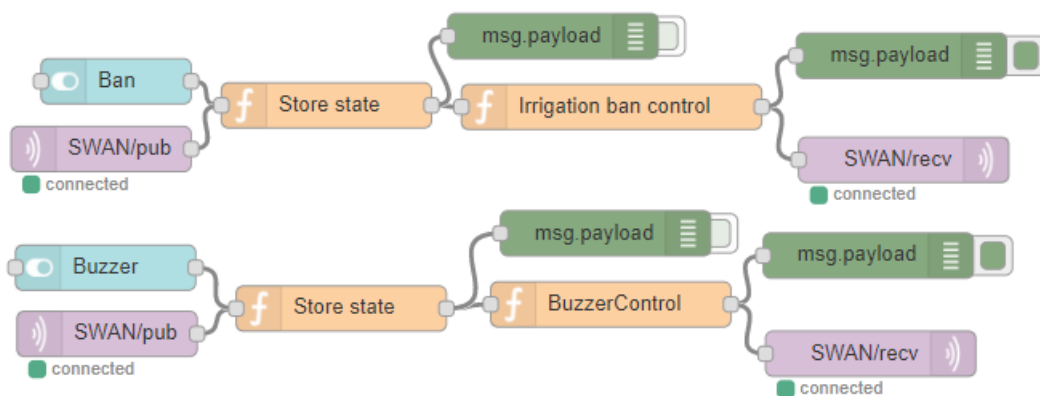
Feladat:	Friss adatok lekérdezése percenként.
Beállítás:	<p>Inject at start once? és perces ciklusidő beállítva.</p> <p>Topic: <i>SELECT * FROM SWAN ORDER BY primkey DESC</i></p>



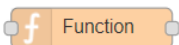
Delete

Feladat:	Óránként a 30 napnál régebbi adatok törlése az adatbázisból.
Kód:	Egy órás ciklusidő beállítva. Topic: <i>DELETE FROM SWAN WHERE TIMESTAMP <= strftime('%s','now', '-30 days')*1000</i>

A 4-4. ábra az felhasználó felületet kezelő részt megvalósító node-k közötti kapcsolatok tartalmazza. Az azonos nevű node-k azonos funkciót látnak el, ezért a már korábban bemutatott node-okat nem kerülnek újra bemutatásra.

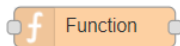


4-4. ábra Felhasználói felület kezelése



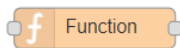
Store state

Feladat:	A switch node állapota nem lekérdezhető, ezért valamilyen javascript nyelvi eszközzel minden állapot változást követni kell. Erre a kontextus a megfelelő eszköz. Ez a függvény elmenti a switch állapotát, majd minden futása során a kimeneten továbbítja azt.
Kód:	<pre>1 var state = context.get('state') false; 2 3 if (msg.topic == 'switch') { 4 state = msg.payload; 5 context.set('state', state); 6 } 7 8 return {payload: state};</pre>



Irrigation ban control

Feladat:	A felhasználói felületen található kapcsoló állapotáról az éppen a hálózatra jelentkező kliensnek tudnia kell. Ez a függvény a kapcsoló állapota szerinti karakterláncot továbbítja az MQTT node-nak.
Kód:	<pre>1 if (msg.payload === true) { 2 msg.payload = "banned"; 3 } 4 else { 5 msg.payload = "enabled"; 6 } 7 8 return msg;</pre>



Buzzer Control

Feladat:	A felhasználói felületen található kapcsoló állapotáról az éppen a hálózatra jelentkező kliensnek tudnia kell. Ez a függvény a kapcsoló állapota szerinti karakterláncot továbbítja az MQTT node-nak.
Kód:	<pre>1 if (msg.payload === true) { 2 msg.payload = "lost"; 3 } 4 else { 5 msg.payload = "found"; 6 } 7 return msg;</pre>

Table

ID	Moisture Voltage [V]	Temperature [°C]	Pressure [hPa]	Status	Timestamp
1 1	23.27	1010.05	0	2020-11-29 21:07:38	
1 1	22.6	1010	0	2020-11-29 20:51:53	
1 1	21.85	1010.11	0	2020-11-29 20:36:10	
1 1	22.56	1010.09	0	2020-11-29 20:20:27	
1 1	23.18	1010	0	2020-11-29 20:04:45	
1 1	23.73	1009.92	0	2020-11-29 19:49:03	
1 1	24.13	1009.78	0	2020-11-29 19:33:21	
1 1	24.09	1009.67	0	2020-11-29 19:17:37	
1 1	24.04	1009.58	0	2020-11-29 19:01:53	
1 1	23.98	1009.63	0	2020-11-29 18:46:09	
1 1	23.89	1009.59	0	2020-11-29 18:30:25	
1 1	23.92	1009.43	0	2020-11-29 18:14:42	
1 1	24.02	1009.46	0	2020-11-29 17:58:58	
1 1	24.17	1009.41	0	2020-11-29 17:43:15	
1 1	24.15	1009.38	0	2020-11-29 17:27:31	
1 1	23.81	1009.26	0	2020-11-29 17:11:47	
1 1	23.47	1009.27	0	2020-11-29 16:56:05	
1 1	23.43	1009.19	0	2020-11-29 16:40:26	
1 1	23.43	1009.17	0	2020-11-29 16:38:36	
1 1	23.42	1009.13	0	2020-11-29 16:36:20	
1 1	23.37	1009.06	0	2020-11-29 16:30:34	
1 1	23.36	1009.05	0	2020-11-29 16:29:15	
1 1	23.36	1009.04	0	2020-11-29 16:28:26	
1 1	23.3	1008.94	0	2020-11-29 16:17:43	
1 1	23.29	1008.91	0	2020-11-29 16:15:25	
1 1	23.28	1008.94	0	2020-11-29 16:13:07	
1 1	23.27	1008.88	0	2020-11-29 16:10:40	

5 Konklúzió

5.1 Eredmények

5.2 Továbbfejleszthetőség

6 Irodalomjegyzék

[1] KSH, A lakott lakások lakóövezeti jelleg szerint (2001)

http://www.ksh.hu/nepszamlalas/docs/tablak/lakasviszonyok/12_02_28.xls

[2] Önálló laboratórium projekt

<https://github.com/Zaion-BM/SWAN>

[3] Mágnesszelep működése

https://en.wikipedia.org/wiki/Solenoid_valve (2020.10.15)

[4]

[5]

[6]

[7]

[8]

[9]

[10]

[11]

[12]

[13]

[14]

[15]

7 Függelék