Magyar Agrár- és Élettudományi Egyetem
Károly Róbert Campus
Műszaki Intézet

PORTFÓLIÓ

Nagy Attila Sándor
(AIBWCZ)
programtervező informatikus felsőoktatási szakképzés

Dr. Pántya Róbert
egyetemi adjunktus

Gyöngyös
2023

Tartalomjegyzék

1. Bevezetés

Kisgyermekként kaptam a szüleimtől egy Commodore 64 típusú számítógépet, természetesen az akkoriban elengedhetetlen kellékekkel: Commodore Datasette magnóval és joystick-kel, azaz botkormánnyal. A botkormány bekerült valamelyik szekrény mélyére, mivel talán pár alkalommal használtam. Mindig az érdekelt, hogyan tudom programozni a számítógépem, elkezdtem tanulni a BASIC nyelvet. *Így kezdődött az egész!*

> *„Úgy vélem, semmi sem fogható ahhoz a lázas izgalomhoz, ami akkor tölti el a feltaláló szívét, amikor látja, hogy valamely agyszüleménye sikeresen megvalósul."*
> /Nikola Tesla (1856–1943) szerb-amerikai fizikus és feltaláló/ (Kurzweil 2014)

Sajnos nem vagyok – a hétköznapi értelemben vett – feltaláló, ennek ellenére informatikai szakemberként értem, érzem és rendszeresen megélem az fenti idézet mondanivalóját. Legyen az a legapróbb, pár sorból álló programkód, legyen az egy általam épített lokális hálózat. Ugyanakkor, ha „feltalálásról" beszélünk, szerintem az is egy ötlet kidolgozása, csiszolgatása, fejlesztése. Mindig alkotni, jobbá akartam tenni valamit. *Régebben is, most is, mindig azt vallom, hogy mi, a programozók vagyunk a látnokok. Mi vagyunk, akik megtervezik és megvalósítják a jövőt.* Gondoljunk csak bele: manapság már gyakorlatilag a kávéfőzőktől kezdve, a mosógépeken, gázkazánokon át, ... a legtávolabbi műholdakig és űrszondákig szinte majdnem minden eszközben szoftverek futnak. Mindig a szoftverek „vitalizálják", azaz keltik életre a mai eszközök nagy részét.

A programozással kapcsolatban – az egyetemi tanulmányaim megkezdéséig – szinte majdnem mindent autodidakta módon sajátítottam el. Mindössze a gimnáziumi éveim alatt tanultam Pascal nyelven programozni, nagyon rövid ideig. Az első gyakorlatban használt szoftvert Clipper-rel készítettem, mely egy egyszerű gyártási technológiai adatbázis volt az akkoriban még általánosan használt dBASE táblákkal. A szoftver feladata mindössze annyi volt, hogy a törzsadatként feltöltött anyag- és munkaidőnormák, továbbá a forgalmi adatként folyamatosan rögzített gyártási adatok alapján összesített anyagfelhasználást kalkuláljon az anyagkönyvelő részére, továbbá a szűkített önköltségszámításhoz szolgáltatott alapinformációkat a számvitelnek. A szoftvert mérlegképes könyvelő végzettségű, akkoriban főkönyvelőként dolgozó édesanyám és az anyagkönyvelést végző kolléganője használta, megkímélve őket a nehézkes kézi számításoktól, hiszen:

> *„A kőkorszak nem azért ért véget, mert elfogyott a kő."*
> /James Canton (1951–) amerikai jövőkutató/ (Kaku 2011)

Több helyen dolgoztam informatikai területen: múzeumban, cégeknél, közigazgatásban. A rendszerszemléletemből adódóan elég hamar megtalálom a folyamatok azon pontjait, melyek egyszerűsíthetők, automatizálhatók.

Tapasztalatomból adódóan nem kellett szakmai gyakorlaton részt vennem, így döntenem kellett hogy egy jelenleg általam – a vállalkozásom keretében – fejleszett saját projektről (magyarkézből.hu) írok, vagy egy másik, az egyetemi tanulmányaim során elkezdődött és azóta is futó komplexebb csapatprojektről. A saját projektem Ruby nyelven, Ruby on Rails keretrendszer alkalmazásával készülő online piactéri projekt. Igazából talán csak a Webprogramozás tantárgyak kötődnek hozzá. A csapatprojekt, – az ún. AgroDroneSystem – az egyetemen a Hungarian Startup University Program keretében indult, melynek programozójaként szinte majdnem teljes egészében *az egyetemen szerzett friss tudást és bátorságot* használtam fel. Úgy döntöttem, hogy ezen különleges projektet mutatom be a szakmai beszámolómban.

Jelen dolgozatom már a sokadik, többször átdolgozott verziója az eredetinek. Feltétlenül bízom benne, hogy mind a belső konzulensem: Dr. Pántya Róbert tanár úr, mind a dolgozatom bírálói, mind a záróvizsga bizottság tagjai, mindazon személyek, akiknek a későbbiekben kerül kezébe a dolgozatom, ugyanolyan élvezettel olvassák majd, mint amilyen élvezet volt a részemről elkészíteni, összeállítani.

2. Az AgroDroneSystem

A Pintér Bertold projektgazda által AgroDroneSystem-nek „elkeresztelt” projekt egy különleges projekt, mely a manapság egyre jobban előtérbe kerülő precíziós mezőgazdaság témaköréhez kapcsolódik. A projekt lényegében a permetező drónok minél automatizáltabb földi kiszolgálását végző egység kifejlesztéséről és megépítéséről szól. A végső cél az, hogy – az akkumulátorok cseréjének kivételével – a drónok teljesen önállóan végezzék el a területek permetezését, az operátor feladata csak a felügyeletre szorítkozzon.

2.1. Előzmények

A 2021/22-es tanévben csatlakozott a Magyar Agrár- és Élettudományi Egyetem a Hungarian Startup University Program-hoz, melynek *„célja a hazai egyetemisták megismertetése az innováció világával, a modern vállalkozói ismeretekkel és különösen a startupok működésével, mindezt egy új, közös oktatási platformon keresztül. A HSUP egy két féléves e-learning tárgy, amelynek I. félévében az innovatív gondolkodásmód és a startup világ megismerése kerül a fókuszba, míg a II. félévben egy vállalkozás felépítésével kapcsolatos gyakorlati tudást”* (Nemzeti Kutatási, Fejlesztési és Innovációs Hivatal 2022) ad át a tárgyakat felvevő hallgatók részére.

Nem volt kérdés, hogy én is felveszem a HSUP I. tantárgyat, mely teljesítéséhez el kellett készíteni egy ún. „one-pager”-t egy saját ötletről. A beküldött ötletekből a MATE-n alapított bizottság választott ki olyan projekteket, melyeket megvalósításra érdemesnek talált. Sajnos az én elképzelésem elhasalt. A HSUP II. tantárgy keretében van lehetőség a támogatásra érdemesnek talált projektek gazdájaként csapatot alapítani, továbbá – nem támogatott ötletek esetében – más projekthez csapattagként csatlakozni, melyek akár egyetemeken átívelők is lehetnek.

Mivel programtervező informatikus képzés a MATE-n belül csak a Károly Róbert Campus-on van – úgy döntöttem, hogy – erősítve ezzel Gyöngyös és a programtervező informatikus szak hírnevét – egyetemen belüli projekthez csatlakozok. Egymásra találtunk Pintér Bertolddal, a Szent István Campus mezőgazdasági mérnök BSc szakának hallgatójával, aki programozót keresett csapatába a fentebb említett eszköz megvalósításához.

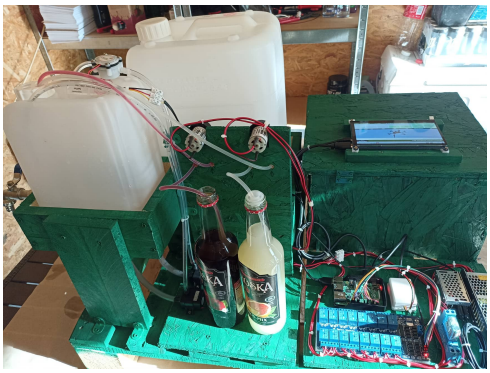
2.2. Megvalósítás

A HSUP keretében támogatásra érdemesnek tartott projektek esetében nem kötelező „fizikailag” bármit is megvalósítani. A projektet a program szabályai szerint ugyan tovább kell vinni, de ez dokumentálásból, számítások elvégzéséből, egy ún. „progress report” összeállításából és a projekt, továbbá a csapat bemutatásáról szóló videó elkészítéséből áll. Az anyagok összeállítását és a HSUP által biztosított felületre történő feltöltését követően azonban érdekes és számomra meglepő dolog történt: a csapatunkat – 19 másik csapattal

együtt – meghívták a *HSUP Demo Day 2022* elnevezésű rendezvényre, ahol két – részben befektetőkből álló – zsűri előtt mutatkozott be 10-10 projekt. A meghívó a HSUP Demo Day 2022 előtt két héttel érkezett meg.

Alapvető, hogy készülnie kellett egy diasornak a projektről, ugyanakkor a csapatunk azon gondolkodott, készíteni kellene egy ún. „deszkamodellt" mely a prototípus közel kicsinyített mása és bemutathatóvá teszi legalább az automatizált permetlékeverés folyamatát. Sokat beszélgettünk, majd megszületett – a szerintem nagyon jó – a döntés, így hatalmas nagy hajrával, gyakorlatilag egy hét alatt elkészült az 1. ábrán látható „deszkamodellünk", mely – a permetlé készítését látványosan bemutatva – kiváló málna és/vagy citromszörpöt tudott készíteni. Tudomásom szerint mi készítettünk egyedül a zsűrinek valamit a diasorokon kívül. A bemutatott eszköz egy 5 literes vegyszerkeverő „tartályba" adagolt egy 20 literes viztartályból vizet, beállítás szerint mellé adagolta a szörpö(ke)t és összekeverte. A zsűri – a mi águnkon – a három legjobb projekt közé választotta a projektünket.

1. ábra. Az elkészült „deszkamodell"



Forrás: saját

Az azóta eltelt időszakban – ahogy a 2. ábrán látható – természetesen elkezdtük a valódi prototípus készítését már 1 m^3-es térfogatú víztartállyal és 200 literes vegyszertartállyal, így abban a szerencsés helyzetben vagyok, hogy mind a „deszkamodell", mind a prototípus kapcsán írni tudok a projektről.

2.3. Hardver

Egyértelmű elvárás volt a csapatunk részéről már a „deszkamodellel" szemben is, hogy egyszerűen kezelhető, később – funkcionalitását tekitve – könnyen bővíthető eszköz készüljön el. Mivel a „deszkamodell" gyakorlatilag versenyre készült, lehetőleg látványos megoldást szerettünk volna bemutatni a zsűritagoknak. A hardvereszközök kiválasztásánál ugyanakkor – bár a csapatunk minden tagja kapott ösztöndíjat a HSUP-tól és az egyetem mellett mindannyian dolgozunk – figyelembe kellett venni bizonyos költséghatékonysági szempontokat is. Ezen elvek mentén döntöttem, hogy milyen eszközöket szerezzünk be már itt is maximálisan figyelembe véve az eszközök ár-érték arányát.

2. ábra. A készülő prototípus

Forrás: Pintér Bertold

2.3.1. Raspberry Pi 4 model B

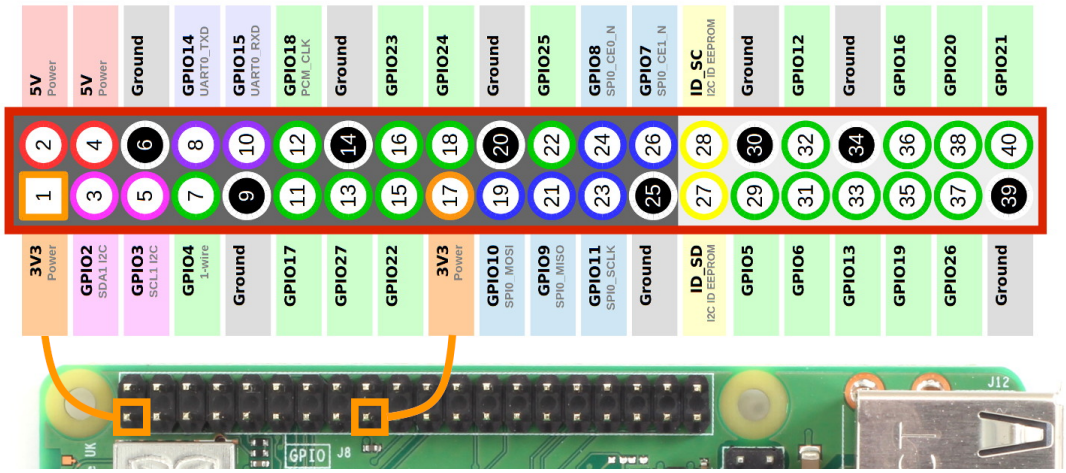
Az eszköz „agyának” a 3. ábrán látható Raspberry Pi 4 model B-t választottam. A Raspberry Pi egy kisméretű számítógép, melyet eredetileg olcsó, könnyen elérhető és használható, oktatási célú eszköznek szántak. A Raspberry Pi azonban széles körben használatossá vált a Do-It-Yourself (DIY, csináld magad) jellegű projektekben, továbbá a robotikai jellegű projektek és az Internet of Things (IoT, dolgok internete) területén. A Raspberry Pi 4 Model B a sorozat legújabb verziója. Az előző modellekhez képest új funkciókat és több javítást tartalmaz, amelyekkel még több feladat elvégzésére vált alkalmassá. A Raspberry Pi 4 Model B erősebb processzorral, nagyobb memóriakapacitással rendelkezik.

3. ábra. Raspberry Pi 4 Model B

Forrás: Wikipédia

A Raspberry Pi 4 Model B négyféle memóriamérettel rendelhető (1 GB, 2 GB, 4 GB és 8 GB). Az eszköz két micro-HDMI porttal, négy USB (két USB 2 és két USB 3) porttal, gigabites Ethernet porttal rendelkezik. Az alaplapon van WiFi modul, egy DISPLAY és CAMERA port, továbbá a távtáplálást lehetővé tévő tüskék. A Raspberry Pi-t gyártója ipari célokhoz is ajánlja. (Raspberry Pi Trading Ltd. 2023) Legfőbb érvként az eszköz teljesítményét, robosztusságát, alacsony energiaigényét és minőségét tudom említeni.

4. ábra. A GPIO tüskesor



Forrás: Raspberry Pi Spy

A Raspberry Pi egyik legjobb tulajdonsága, hogy a 4. ábrán látható – a külső, a „fizikai” világgal való kapcsolattartást egyszerűen lehetővé tévő – GPIO (General Purpose Input/Output – általános célú bemenet/kimenet) tüskesorral (header-rel) rendelkezik. (Hawkins 2012) A Raspberry Pi 4 model B modellen – a Pi 1-es sorozatot követően kiadott modellekkel egyezően – 40 tüske található. A 40 tüskéből elérhetők több tüskén keresztül a 3,3 V-os (3V3), az 5 V-os (5V) tápfeszültségek, a (logikai) földelés (GND), továbbá a GPIO 0–GPIO 26-ig számozott GPIO portok.

A GPIO portokról általánosságban elmondható, hogy tetszőleges céllal használhatók mind digitális bemenetként, mind digitális kimenetként. Fontos megemlíteni ugyanakkor, hogy vannak *kiemelt GPIO portok,* melyeken keresztül a Raspberry Pi PWM (Pulse width Modulation), I^2C (inter-IC), I^2S (inter-IC Sound), SPI (Serial Peripheral Interface, szinkron soros kommunikációs) és UART (Universal asynchronous receiver-transmitter, univerzális aszinkron adóvevő) interfészt biztosít. A felsorolt kommunikációs eszközökkel a Raspberry Pi gyakorlatilag kimeríthetetlen lehetőségeket biztosít az általunk csatlakoztatható eszközök (pl. pH-szenzor, folyadékszint-érzékelő stb.) terén. Sajátossága továbbá az eszköznek, hogy – a vezérlő szoftverből – bármelyik GPIO port megnyitható digitális bemenetként az alaplapra épített – gyakorlatilag „szoftveres” – lehúzó vagy felhúzó ellenállással is, így – általában – nem szükséges ezen eszközök beépítése.

Analóg bementekkel a Raspberry Pi alapból sajnos nem rendelkezik, de egy – a szakirodalom szerint általában leggyakrabban alkalmazott – *MCP3008* típusú analóg-digitális konverter IC-vel kiegészítve nyolc darab analóg bemenettel bővülnek a lehetőségeink. (Microsoft Learn 2023b) Az MCP3008 IC és a Raspberry Pi összekötését az 1. táblázatban foglaltam össze. A táblázatban nem szerepelnek az MCP3008 lábai 1–8-ig, mivel ezeken a lábakon vannak az analóg bemenetek rendre CH0–CH7-ig. Analóg bemenetre később lesz szükségünk, mivel a legtöbb szélsebességmérő (anemométer) analóg eszköz.

1. táblázat. Az MCP3008 és a Raspberry Pi összekötése

MCP3008 (láb)	Tüske (RPi)	Port (RPi)
V_{DD} (16)	2	3V3
V_{REF} (15)	2	3V3
AGND (14)	25	GND
CLK (13)	23	GPIO 11 (SPI0 SCLK)
D_{OUT} (12)	21	GPIO 9 (SPI0 MISO)
D_{IN} (11)	19	GPIO 10 (SPI0 MOSI)
CS/SHDN (10)	24	GPIO 8
DGND (9)	25	GND

Forrás: saját

Ahogy a táblázatból látható, a Raspberry Pi tüskesorából a Serial Peripheral Interfacehez (SPI) tartozó kiemelt GPIO portok használhatók az MCP3008-hoz történő csatlakozáshoz. Nagyon fontos, hogy az interfész használatát – a későbbiekben ismertetett I^2C interfészhez hasonlóan – a Raspberry Pi beállításainál engedélyezni kell.

Bár korábban sosem fejlesztettem Raspberry Pi-re alkalmazást, magát az eszközt is csak – mint érdekességet – kipróbáltam régebben, a fentebb felsorolt lehetőségei miatt egyértelmű volt hogy a „deszkamodellhez" és a későbbi prototípushoz Raspberry Pi kell! Már a legelején eldöntöttem, hogy .NET-es alkalmazást szeretnék fejleszteni C Sharp nyelven, melyek a Raspberry Pi-n – több operációs rendszerrel – futtathatók.

2.3.2. Érintőkijelző

Kijelzőkkel (legyenenek azok egyszerű LED-ek, LCD kijelző, monitor stb.) találkozunk minden alkalommal, amikor bármilyen elektronikus eszközt használunk, ezért kritikusak voltunk a kijelző megválasztásakor. Mivel a Raspberry Pi mellett döntöttem evidens döntés volt a részünkről, hogy érintőkijelzőt vásárolunk. Praktikussága mellett renkívüli módon megemelte a „deszkamodellünk" használati élményét, továbbá látványnak sem utolsó egy ilyen eszközön. Előny továbbá, hogy az érintőkijelző alkalmazásával javarészt szükségtelenné válnak a kezeléshez szükséges szervek (nyomógombok, kapcsolók stb.).

Az 5. ábrán látható 7"-es képátlójú IPS érintőkijelzőt vásároltuk meg. Ez volt a „deszkamodellen" és ezt szereljük át a prototípusunkra is. A Raspberry Pi 4 rendelkezik DISPLAY porttal, ahová a kompatibilis érintőkijelzők egyszerű szalagkábellel csatlakoztathatók, melyek – az általunk megismert modellek tekintetében – jóval kisebb felbontásúak, ugyanakkor sokkal költséghatékonyabbak, ezért a későbbiekben vizsgálni fogjuk ezen kijelzők alkalmazási lehetőségeit.

Az érintőkijelző két kábellel csatlakozik a Raspberry Pi-hez. Egyszerű HDMI–micro HDMI kábellel kötöttem össze a kijelző HDMI portját a Raspberry Pi HDMI0 portjával, míg táplálás és az érintőpanel jeleinek továbbítása micro USB–USB kábellel történik.

5. ábra. Kapacitív érintőkijelző 7" IPS, 1024×600, HDMI porttal

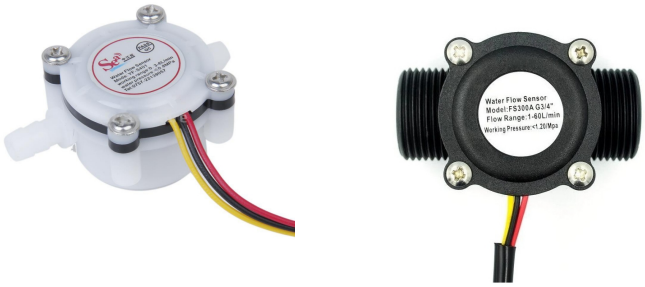
Forrás: rPI Bolt

2.3.3. Áramlásérzékelő

A permetlé keverésénél elengedhetetlenül fontos az „oldószer”, azaz a víz pontos mennyiségben történő adagolása is, ezért – a 6. ábrán látható – ún. Hall-szenzoros áramlásérzékelőt szereltünk fel. A „deszkamodellen” YF-S401-es típust alkalmaztunk, melynek teljesítménye 0,3–6 liter/perc. Ez a típus – tekintettel arra, hogy ott kis szivattyúk működtek – bőven megfelelt. A prototípusra FS300A G3/4" típusú áramlásérzékelőt szerelünk, melynek teljesítménye 1–60 liter/perc.

6. ábra. Áramlásérzékelők

(a) YF-S401 (b) FS300A G3/4"



Forrás: ElektROBOT.hu, mikroelektronik.hu

Az áramlásérzékelők a vízáramlás megindulását követően, az áramlás sebességével/-mennyiségével arányos számú impulzust generálnak a SIGNAL kimenetükön, mely számlálásával egyszerűen megállapítható az átáramlott víz mennyiége. Az érzékelőt közvetlenül a Raspberry Pi GPIO tüskesoráról táplálom, jelvezetéke bármelyik GPIO porthoz csatlakoztatható, melyen keresztül olvasható.

Az áramlásérzékelőből kivezetett hárompólusú kábelt RJ11-es fali aljzatba kötöttem, melynek párja – szintén RJ11-es aljzat – kapcsolódik a Raspberry Pi GPIO tüskéihez jumper vezetékek segítségével. A két fali aljzat közé a prototípuson előszerelt, négypó-

2. táblázat. Az áramlásérzékelők és a Raspberry Pi összekötése

YF-S401/FS300A G3/4"	Tüske (RPi)	Port (RPi)
VCC+ (piros)	2	5V
SIGNAL/DATA (sárga)	18	GPIO 24
GND (fekete)	20	GND

Forrás: saját

lusú telefonvezetéket, ún. „készülékzsinórt" csatlakoztatunk, mellyel egyszerűen bontható kapcsolatot alakítottam ki (20. ábra). A két eszköz összekötését a 2. táblázat szemlélteti. Az áramkörben nem szükséges „fizikai" felhúzó ellenállás beépítése, mivel a Raspberry Pi beépített, programból vezérelhető felhúzó ellenállását használom.

2.3.4. Hőmérséklet- és páratartalom-érzékelő

A „deszkamodellnek" nem volt része, de a prototípushoz csatlakoztatásra kerül egy, a 7. ábrán látható AM2315 típusú, tokozott hőmérséklet- és páratartalom-érzékelő egység, ugyanis a permetezésnél – különösen, ha az drónnal történik – természetesen a kezelt kultúrától és/vagy az alkalmazott peszticidtől függően sok esetben (felszívódás, hatékonyság stb.) komoly jelentősége lehet ezen időjárási körülményeknek is. (MagroPontHu 2020) Sajnos – egyszerűbb és olcsóbb – szélsebesség- és szélirányмérő egységet – mely a drónnal történő permetezés szempontjából a legfontosabb – egyelőre még nem tudtunk beszerezni, de úgy vélem, hogy belátható időn belül erre is sor kerül. Jelenleg az operátor – a permetezés megkezdésekor és azt követően folyamatosan – kézi szélsebességmérővel végez méréseket.

7. ábra. AM2315 hőmérséklet- és páratartalom-érzékelő

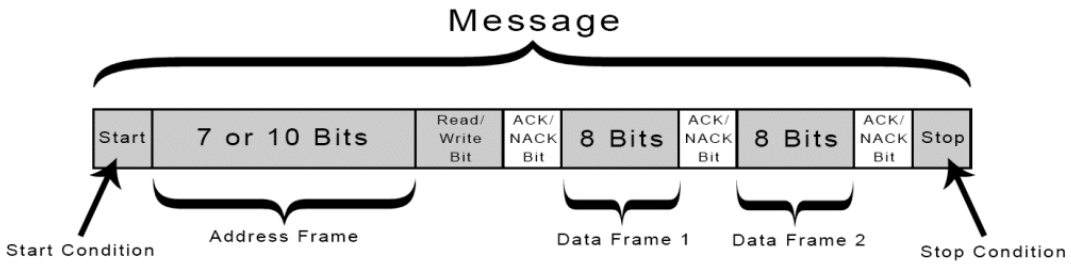
Forrás: Amazon.sa

Informatikai szempontból érdekessége ezen eszköznek, hogy az I²C (inter-Integrated Circuit, inter-IC) kommunikációs protokollt használja a „gazdagéppel", esetünkben a Raspberry Pi-vel történő adatcseréhez. Az I²C szabványt a Philips Semiconductor (ma NXP Semiconductors) fejlesztette ki kimondottan abból a célból, hogy kis távolságban lévő eszközök – akár egy eszközön belüli részegységek – számára biztosítson egy csomagkapcsolt

soros buszt a hozzá kapcsolódó protokollal együtt. (Wikimédia Alapítvány 2023a) Egy kiépített buszon több vezérlő és periféria is jelen lehet.

Az I^{2}C busz gyakorlatilag két jelből, így – a tápellátást végző és a földelő (GND) vezetéket nem számítva – két vezetékből áll. Az egyik a Serial Clock (SCL), az órajel, amit az éppen aktív vezérlő eszköz generál. A másik a Serial Data (SDA) az adatjel. Ha nincs semmilyen kommunikáció a buszon, akkor mindkét vezetéknek logikai magas jelszinten kell lennie ezért alapszabály, hogy gondoskodni kell az ún. felhúzó ellenállások beépítéséről. A szakirodalom szerint az I^{2}C esetében általában 4,7 kΩ értékű ellenállásokat kell használni mindkét vezeték és a tápvezeték között, de a busz méretének és eszközök számának függvényében ezen érték csökkentése válhat szükségessé. (SparkFun Learn 2023) A buszon aszinkron soros kommunikáció zajlik a vezérlő eszköz által generált SCL jel ütemében. Az üzenetek az SDA jelvezetéken továbbítódnak, melyek két részből, 7 vagy 10 bit-ből álló címzésből, egy vagy több 8 bites adatcsomagból és jelzőbitekből (ACK/NACK) állnak. Az I^{2}C üzenet felépítését a 8. ábra szemlélteti.

8. ábra. Az I^{2}C üzenet felépítése



Forrás: Circuit Basics

I^{2}C busszal, továbbá ahhoz kapcsolódó, azon keresztül kommunikáló eszközökkel korábban még nem találkoztam. Az eszköz, továbbá a Raspberry Pi GPIO portjának részletesebb dokumentációját áttekintve természetesen megtaláltam a megoldást, mely meglepően egyszerűnek bizonyult, mivel a GPIO 2 (SDA) és GPIO 3 (SCL) tüskékre követlenül csatlakoztatható az I^{2}C busz.

3. táblázat. Az AM2315 érzékelő és a Raspberry Pi összekötése

AM2315	Tüske (RPi)	Port (RPi)
VCC+ (piros)	1	3V3
SDA (sárga)	3	GPIO 2 (SDA)
SCL (fehér)	5	GPIO 3 (SCL)
GND (fekete)	9	GND

Forrás: saját

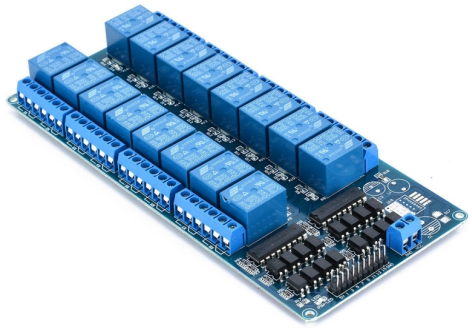
A hőmérséklet- és páratartalom-érzékelő egységből kivezetett négypólusú kábelt – az áramlásmérő egységhez hasonlóan – RJ11-es fali aljzatba kötöttem, melynek párja – szin-

tén RJ11-es aljzat – kapcsolódik a Raspberry Pi GPIO csatlakozójának tüskéihez jumper vezetékek segítségével. A két fali aljzat közé a prototípuson előszerelt, négypólusú telefonvezetéket, ún. „készülékzsinórt" csatlakoztatunk, mellyel egyszerűen bontható kapcsolatot alakítottam ki (19. ábra). A két eszköz összekötését a 3. táblázat szemlélteti. Az áramkörben nem szükséges az I^2C buszhoz a „fizikai" felhúzó ellenállások beépítése, mivel a Raspberry Pi beépített, programból vezérelhető felhúzó ellenállását használom.

2.3.5. Relémodul

A korábbiakban a Raspberry Pi-hez kapcsolódó érzékelőket mutattam be, most a szintén fontos vezérlési részre térek át. A rendszer mondhatni másik „lelke" a relémodul, melynek segítségével a Raspberry Pi-n futó szoftver egyszerűen be- és kikapcsolhatja (vezérelheti) a mikroszámítógép köré épített eszközöket. A relémodulok nem költségesek, ezért az ár-érték arányt, továbbá a későbbi fejlesztési lehetőségeket figyelembe véve minél több relével rendelkező modellt szerettem volna rendelni.

9. ábra. 16 csatornás, 5 V-os tápfeszültségű relémodul



Forrás: ElektROBOT.hu

A 9. ábrán látható 16 csatornás, 5 V-os tápfeszültségű relémodult választottam a „deszkamodellhez", mely eszközt szintén átszerelünk a prototípusra. Ez a relémodul ún. „low-level trigger" üzemmóddal rendelkezik, vagyis a csatlakoztatott Raspyberry PI GPIO portjának alacsony (LOW) jelszintje esetén húznak meg a relék. Pozitív tulajdonsága továbbá az eszköznek, hogy ún. „optocsatolt" elven működik, vagyis a Raspberry Pi-től érkező jelek egy LED-et kapcsolnak be, mely LED-ek fényét egy fototranzisztor érzékeli és így kapnak áramot a relék tekercsei. Mivel a LED-ek kizárólag a Raspberry Pi-hez csatlakoznak, a fototranzisztorok és tekercsek pedig elkülönített áramkörön vannak – külön tápegységek alkalmazásával – megvalósítható a teljes galvanikus leválasztás. A teljes galvanikus leválasztás esetén a Raspberry Pi-t a relémodul tökéletesen meg tudja védeni a köré épített eszközök körében fellépő elektromos zavarok esetén.

A relémodul és a Raspberry Pi összeköttetéseit, továbbá a vezérelt eszközök (funkciók) felsorolását a 4. táblázat szemlélteti. A relémodult – nyomtatott áramkörként – közvetlenül

4. táblázat. A relémodul és a Raspberry Pi összekötése, vezérelt eszközök

Relémodul	Tüske (RPi)	Port (RPi)	Vezérelt eszköz (funkció)
5V	4	5V	
GND	6	GND	
Relé 1. (1)	36	GPIO 16	Mágnesszelep 1.
Relé 2. (2)	37	GPIO 26	Mágnesszelep 2.
Relé 5. (5)	11	GPIO 17	Polaritásváltó (irányváltó) segédrelé
Relé 6. (6)	13	GPIO 27	Perisztaltikus (vegyszeradagoló-)szivattyú 1.
Relé 7. (7)	15	GPIO 22	Perisztaltikus (vegyszeradagoló-)szivattyú 2.
Relé 8. (8)	16	GPIO 23	Perisztaltikus (vegyszeradagoló-)szivattyú 3.
Relé 15. (15)	29	GPIO 5	Vízadagoló szivattyú
Relé 16. (16)	31	GPIO 6	Lékeringtető (keverő) szivattyú

Forrás: saját

a Raspberry Pi mellé szereltem. A relémodul és a GPIO tüskesor közötti kapcsolatot – mind a „deszkamodell", mind egyelőre a prototípus esetében – egyszerű jumper vezetékek segítségével oldottam meg.

A relémodulon bekötött 1., 2., 5., 6., 7. és 8. számú relék gyengeáramú – 12 V-os feszültségű – eszközöket vezérelnek. Talán egyetlen magyarázatra szorulú elem az 5. relé által vezérelt polaritásváltó (irányváltó) segédrelé, melynek feladata a perisztaltikus szivattyúkhoz szükséges polaritás-, így forgásirányváltás (töltés/ürítés) megvalósítása. A szintén bekötött 15. és 16. számú relék erősáramú – 230 V-os feszültségű – szivattyúkat vezérelnek. A relémodulon lévő relék elméletileg képesek lennének közvetlenül elvégezni a nagyobb feszültségű és áramfelvételű eszközök kapcsolását is, de biztonsági okok miatt 12V-os segédreléket fogunk alkalmazni már a prototípusnál is.

2.3.6. Egyéb, „passzív" elemek

Mind a „deszkamodellnél", mind a prototípusnál „passzív" elemekként kezelem azon eszközöket, melyek ugyan fontosak a berendezés működése szempontjából, de nem látnak el érzékelési-vezérlési feladatokat és/vagy nem csatlakoznak közvetlenül a mikroszámítógéphez. Ezen eszközökről csak érintőlegesen, felsorolásszerűen szeretnék írni pár dolgot:

– A berendezés villamos táplálását egyszerű, előregyártott *tápegységekkel* oldottam meg. A főtápegység egy PV-100-12 típusú 180-240 VAC bemeneti feszültségű, 12 VDC kimeneti feszültségű, 100 W-os teljesítményű stabilizált tápegység. A tápegység maximális terhelhetősége: 8,3 A. Ezen eszköz táplálja a berendezés kisfeszültséggel működő elemeit. Egy 12 VDC/5 VDC konverter gondoskosik a mikroszámítógép, a relémodul és közvetve az érintőkijelző táplálásáról (22. ábra).

– Bár a „deszkamodellhez" általam kifejlesztett szoftver összesen három *perisztaltikus (vegyszeradagoló-)szivattyú* vezérlését teszi lehetővé, arra csak kettőt szereltünk

fel, míg a prototípusnál egyelőre egy ilyen szivattyút alkalmazunk. Ezen szivattyúk feladata a permetlé előállításához szükséges, folyékony halmazállapotú peszticidek precíz adagolása. A szivattyúk „normál” üzemben a vegyszeredény irányából szívják fel a növényvédőszert, míg polaritásváltással „hátramenetben”, takarítás előtt a vegyszeradagoló cső gyakorlatilag tökéletesen és veszteségmentesen leüríthető. Ezen szivattyúk 12 V-os gépek, melyeket közvetlenül a relémodul vezérel, melyhez – a prototípuson – egyszerű UTP Cat5e kábelekkel csatlakoznak, a bontható csatlakozás érdekében RJ45-ös csatlakozókkal, aljzatokkal (21. ábra).

– A „deszkamodellen” nem voltak még, de a prototípusnál a víz és permetlé útjának irányítását alaphelyzetben zárt (NC), 12 V-os tekercsű *mágnesszelepekkel* oldjuk meg. A mágnesszelepeket közvetlenül a relémodul vezérli, melyek szintén egyszerű UTP Cat5e kábelekkel csatlakoznak, RJ45-ös csatlakozókkal, aljzatokkal (23. ábra).

– A „deszkamodellen” 12 V-os feszültségű *szivattyúk* végezték a vízadagolást, továbbá a permetlé keverését/keringtetését. Ezen szivattyúkat közvetlenül a relémodul vezérelte és a berendezés tápegységéről tápláltam őket. A prototípusra két 230 V-os üzemi feszültségű szivattyút szereltünk fel, melyek ugyanezen feladatokat látják el. A szivattyúk be- és kikapcsolását – villamos biztonsági okok miatt – a relémodulhoz kötött segédrelékkel valósítjuk meg.

– Logikai és egyes kapcsolási feladatokhoz *segédreléket* alkalmazok, melyek két váltókapcsolóval – ún. „morzeérintkezőkkel” – rendelkező, egyszerű 12 V-os tekercsű relék foglalattal. A perisztaltikus (vegyszeradagoló-)szivattyúkhoz szükséges polaritás-, azaz irányváltás logikai áramkörét, továbbá a minél biztonságosabb galvanikus leválasztás érdekében a prototípusra szerelt 230 V-os üzemi feszültségű szivattyúkat ezen relék kapcsolják, melyek tekercsei közvetlenül a relémodulhoz kapcsolódnak.

2.4. Szoftver

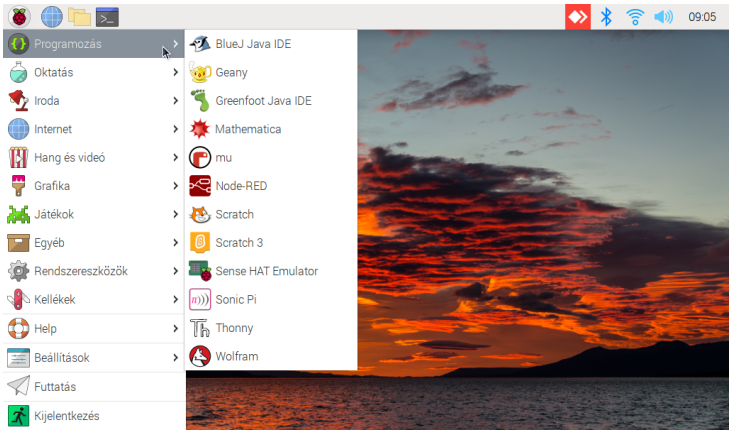
Ezen szakaszban szeretném részletesen bemutatni, hogy a Raspberry Pi-n milyen operációs rendszert használok, milyen eszközökkel, milyen nyelven és milyen minták alkalmazásával történik az AgroDroneSystem eszköz vezérlését biztosító szoftver fejlesztése. Az alkalmazott technológiák kiválasztásánál – az integrált fejlesztői környezetek kivételével – mindenképp szerettem volna olyan lehetőségeket alkalmazni, amik ingyenesek, lehetőleg nyílt forráskódúak és széleskörű közösségi támogatással rendelkeznek, ugyanakkor licenceik (pl. MIT) szempontjából alkalmazhatók kereskedelmi célú eszközök fejlesztéséhez is.

2.4.1. Raspberry Pi OS

Bár számtalan operációs rendszer készült már a Raspberry Pi platformra – többek között a Microsoft által fejlesztett Windows for IoT is – a Raspberry Pi OS-t (korábbi nevén

Raspbian-t) választottam a projekthez. A Raspberry Pi OS-t felfoghatjuk úgy, mint a Raspberry Pi „saját operációs rendszerét”, így egymással teljes mértékben kompatibilisek.

10. ábra. A Raspberry Pi OS felülete



Forrás: saját

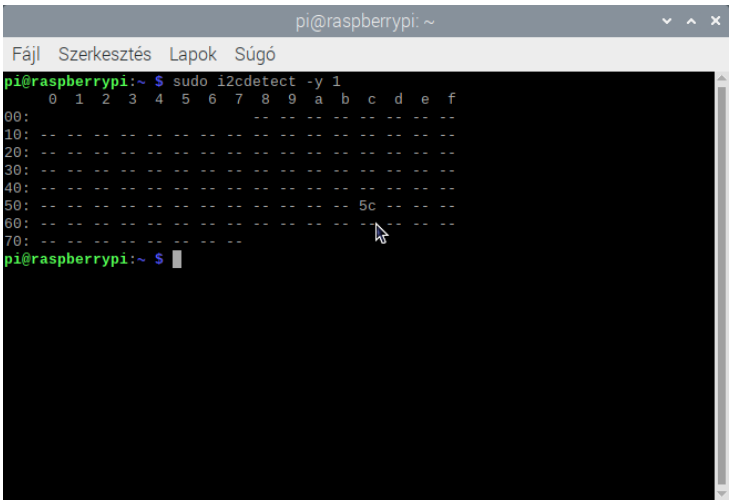
A Raspberry Pi OS a Debian Linux-on alapul, mely kimondottan a Raspberry Pi-hez készült. Az első verziókat Mike Thompson és Peter Green – Raspbian néven – fejlesztette ki, mely 2012-ben került először kiadásra. 2020-ban – már a Raspberry Pi Foundation által – került bejelentése a 64 bit-es verzió, mely hivatalosan 2022-ben jelent meg. 2013-tól a Raspberry Pi Foundation fejlesztei és terjeszti a Raspberry Pi-hez ajánlott elsődleges operációs rendszerként. A Raspberry Pi OS az ARM processzorokkal rendelkező Raspberry Pi-hez van optimalizálva, mely – a Pico kivételével – minden Raspberry Pi-n fut, mely egy – a 10. ábrán látható – PIXEL elnevezésű, LXDE alapú asztali környezetet kínál. A Raspberry Pi OS – a Debian Linux-hoz hasonlóan – az APT-t használja csomagkezelőnek, mellyel – egy saját APT repo létrehozásával – a közeljövőben meg kívánom oldani a vezérlő szoftver újabb kiadásainak terítését, frissítését. (Wikimédia Alapítvány 2023c)

Rendkívül fontos, hogy az I^2C interfész alkalmazásához fel kell telepíteni az *i2c-tools* csomagot. Engedélyezni kell továbbá a – akár terminálon keresztül is elérhető – *raspi-config* vagy grafikus felületen a *Raspberry Pi Configuration* eszközzel az I^2C interfész használatát. A 11. ábrán látható az *i2cdetect* utasítás válasza, mely szerint az I^2C interfészen keresztül a *0x5c* címen érhető el a csatlakoztatott, bővebben a 2.3.4 szakaszban bemutatott AM2315 típusú hőmérséklet- és páratartalom-érzékelő szenzor.

2.4.2. Fejlesztőeszközök

A szoftvert a Microsoft által fejleszett ingyenes, nyílt forráskódú, általános célú – Windows-on, MacOS-en és Linux-on is futó – fejlesztői platform, a *.NET Core 6.0 (LTS)* segítségével, *C Sharp* nyelven készítettem el. A fejlesztéshez elsősorban *JetBrains Rider*-t, továbbá – pár alkalommal – *Visual Studio*-t használtam. A prototípus adatkezeléséhez *Entity Framework*-öt használok, az eszközön lévő helyi adatok kezelését *SQLite3* végzi.

11. ábra. Az i2cdetect utasítás válasza



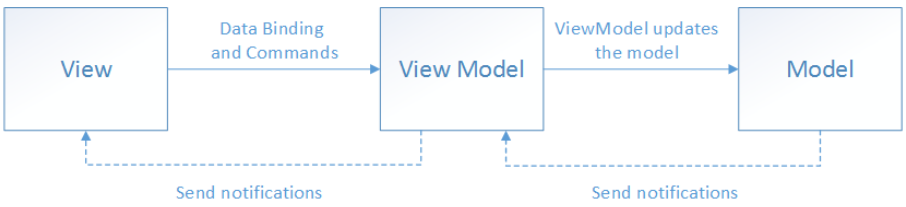
Forrás: saját

A Raspberry Pi esetén a legtöbb fellelhető forrás a *Python* nyelvet preferálja, de az egyetemi tanulmányaim során megismertem és megtapasztaltam a .NET-es alkalmazásfejlesztés előnyeit, továbbá a C Sharp nyelv hatékonyságát és erejét, ezért ragaszkodtam ezen frissen megismert technológiák alkalmazásához.

2.4.3. Avalonia UI, MVVM, ReactiveUI

A *Programozási környezetek* tantárgyunknál megismerkedtem a Windows Presentation Foundation (WPF) alkalmazások készítésével. A WPF rendkívül egyszerű és rugalmas lehetőséget biztosít a felhasználói felületek (nézetek) építéséhez az XML-en alapuló Extensible Application Markup Language (XAML) nyelv segítségével, ugyanakkor nem használható Linux környezetben futó szoftverek fejlesztéséhez. Alternatívaként a nyílt forráskódú *Avalonia UI* keretrendszert használom, mellyel – azonos kódbázissal – asztali, mobil, webes és beágyazott alkalmazások is készíthetők. Az Avalonia UI alapú fejlesztéshez mind a JetBrains Rider-hez, mind a Visual Studio-hoz bővítmények állnak rendelkezésre. Az Avalonia XAML fájlok – mivel a Visual Studio nem tolerálja a WPF-nél is alkalmazott .xaml kiterjesztés használatát – .axaml kiterjesztésűek.

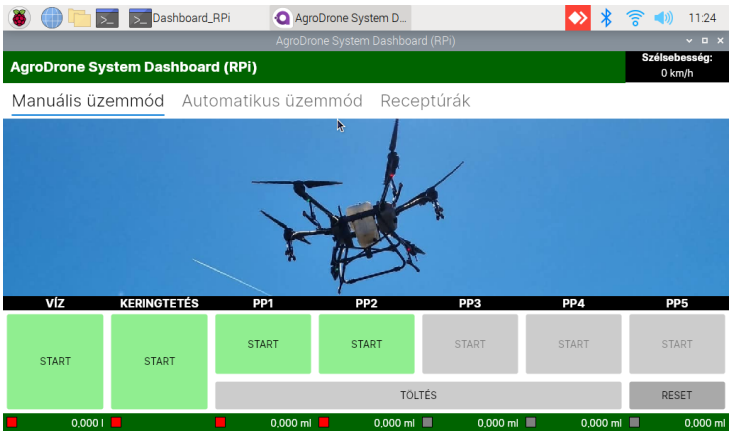
12. ábra. Az MVVM tervezési minta



Forrás: Microsoft Learn

Bár a „deszkamodellhez” készült szoftverben mindössze egy nézetet használok, már ezen alkalmazás felépítéséhez is az – 12. ábrán áttekinthető – *Model-View-ViewModel (MVVM)* tervezési mintát alkalmaztam. Az MVVM segítségével tökéletesen elválaszthatók egymástól a felhasználói felület megjelenítéséért felelős nézet (View), a nézet és a modell közötti interakciót biztosító nézetmodell (ViewModel) és az adatok kezelését/tárolását végző modellek (Model). A forráskód így könnyebben karbantartható és a legtöbb esetben újra felhasználható. A nézetekben szinte kizárólag a XAML nyelven kódolt felhasználói felület készül el. A nézeteken elhelyezett vezérlők – tartalmukat, funkcionalitásukat, pár esetben tulajdonságaikat illetően – *kötésekkel (Binding) csatlakoznak a ViewModel tulajdonságaihoz (property-jeihez)*, mely alapján folyamatosan frissülnek. A feladattól függően többféle mód (BindingMode Enum) (Default, OneTime, OneWay, OneWayToSource, TwoWay) létezik, mellyel az adatok mozgását szabályozni tudjuk.

13. ábra. Az AgroDroneSystem Dashboard felhasználói felülete a „deszkamodellen”



Forrás: saját

Az MVVM-es alkalmazások helyes működéséhez, vagyis a nézetek-nézetmodellek-modellek közötti kommunikáció megvalósításához szükséges implementálni a nézethez kapcsolódó nézetmodellekben, modellekben a .NET Core által definiált *INotifyPropertyChanged* interfészt. (Microsoft Learn 2022b) Ezen interfész implementációja a NuGet Gallery-ből letölthető *ReactiveUI* és *ReactiveUI.Fody* csomagok betöltésével, továbbá a nézetmodell vagy modell tulajdonságainak (property-jeinek) definiciója elé írt *[Reactive]* annotációval kiküszöbölhető, a forráskód egyszerűbb lesz. (ReactiveUI Association Inc. 2023)

2.4.4. GPIO

A 2.3.1. szakaszban írtam a Raspberry Pi GPIO tüskesoráról, továbbá a GPIO portokról. Első alkalommal – programozási szempontból – érdekes feladat volt számomra a GPIO portok elérése és manipulálása, ezért pár megjegyzés erejéig szeretnék rá kitérni, melyek elérését (írását, olvasását) a NuGet Gallery-ből betölthető *System.Device.Gpio* csomaggal lehet egyszerűen megvalósítani. (Microsoft Learn 2023a)

1. kód. Példakód a GPIO portok írásáról

```
1  using System;
2  using System.Device.Gpio;
3
4  GpioController gpioController = new GpioController();
5
6  gpioController.OpenPin(10, PinMode.Output);
7  gpioController.Write(10, PinValue.High);
8  gpioController.Write(10, PinValue.Low);
9
10 gpioController.ClosePin(10);
```

A GPIO portok írásához – ahogy az 1. kódnál látható – példányosítanunk kell a *GpioController* osztályt, melynek metódusain keresztül nyithatók, írhatók és zárhatók a GPIO portok. A GpioController objektumból – a legtöbb esetben – alkalmazásonként elegendő egyet létrehozni, több GpioController objektum használata esetén az alkalmazás instabillá válhat, amennyiben nem gondoskodunk a portműveletek előtti állapotellenőrzésekről, melyekhez külön metódusok állnak rendelkezésre. A GpioController osztály – bár több eszközzel kompatibilis – kiváló lehetőségeket biztosít a Raspberry Pi GPIO portján keresztüli kommunikációhoz. Fontos megjegyezni, hogy a dokumentációban „pin"-nek nevezett paramétereknél minden esetben a GPIO port (logikai) sorszámát és nem a tüske „fizikai" sorszámát kell átadni a metódushívásoknál.

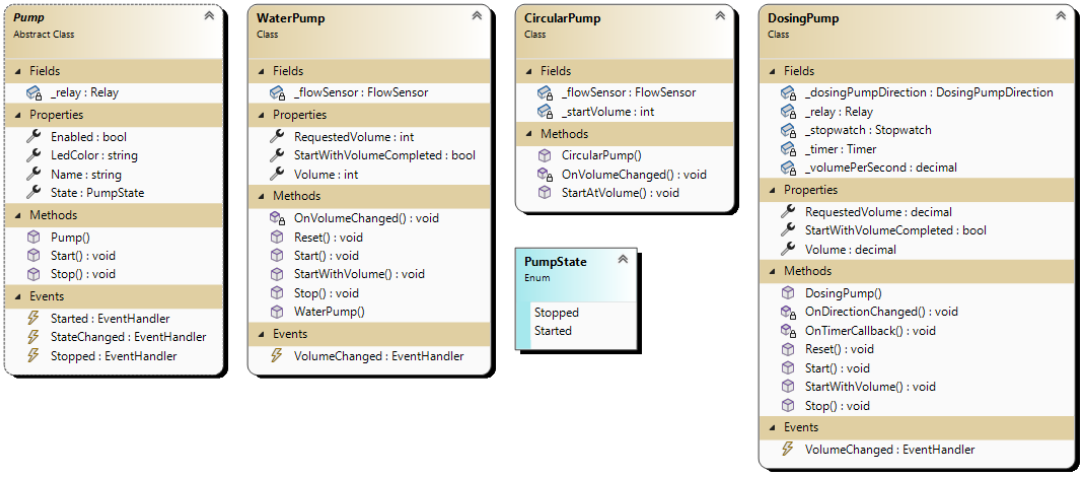
A GpioController objektum egyik leghasznosabb metódusa a *RegisterCallbackForPinValueChangedEvent()* metódus, mely delegate segítségével lehetővé teszi, hogy a paraméterként megadott GPIO port állapotváltozásainál meghívásra kerüljön a – szintén paraméterként átadott – saját eljárásunk. (Microsoft Learn 2022a) A szoftverben – bővebben a 2.3.3. szakaszban bemutatott – áramlásérzékelők által küldött impulzusokat ezen megoldás segítségével olvasom és számlálom.

2.4.5. Osztályok, objektumok

A szoftver tervezése és megvalósítása során a – Raspberry Pi köré épített, kézzel fogható – hardverelemeket „építőkockáknak" tekintve minden eszközhöz külön osztályt terveztem és készítettem, természetesen – a legtöbb esetben – alkalmaztam az objektumorientált programozás alapelveit. (Microsoft Learn 2022c) Ezen osztályok segítségével – már tetszőleges, könnyen módosítható működési logikával – egyszerűbb, könnyebben karbantartható és részben újrahasznosítható forráskóddal készült el az AgroDroneSystem Dashboard.

Az osztályok tervezésekor és megvalósításakor figyelembe kellett vennem, hogy az MVVM tervezési mintát alkalmazom. A Pump „ősosztály" például ennek megfelelően, a felhasználói felületen történő egyszerű „állapotkijelzés" lehetőségével készült. Minden osztálynál definiáltam event-eket, melyekkel egyszerűen jelezni tudom a szoftverben használt más objektumoknak, ha az adott objektumnál bármilyen változás áll be.

14. ábra. A Pump osztályok öröklődésének bemutatása osztálydiagrammal



Forrás: saját

Az objektumorientált programozási alapelvek (egységbezárás, öröklődés, többalakúság) legszebb példáját mutatja a 14. ábrán látható osztálydiagram. A vízszivattyú (WaterPump), a lékeringtető (keverő) szivattyú (CircularPump) és a perisztaltikus (vegyszeradagoló-)szivattyú (DosingPump) vezérléséhez készült osztályok mind egy őstől, a Pump absztrakt osztályból származnak. Az egyes szivattyúknál eltérő működési és vezérlési módokat polimorfizmussal, továbbá újabb metódusok implementálásával oldottam meg. A mellékletben található 15. ábrán látható osztálydiagramon tekinthetők át a szoftverben létrehozott releváns osztályok, a fontosabbak feladatairól és működéséről felsorolásszerűen írni szeretnék pár gondolatot, információt:

- A *Relay* osztály a legegyszerűbb, melynek feladata a relémodulon lévő relék vezérlése. A konstruktor részére a már létrehozott GpioController objektumot, továbbá a GPIO port számát kell átadni. A létrejött objektum segítségével a relék egyszerűen az *On()/Off()* metódusokkal vezérelhetők, az aktuális státusz a *State* tulajdonsággal ellenőrizhető. Az állapotváltozások – szükség esetén – a *TurnedOn, TurnedOff* és *StateChanged* eseményekre történő feliratkozásokkal követhetők.

- A *Pump* egy *absztrakt osztály,* mely önállóan nem példányosítható. Alapfunkciókat valósít meg *virtuális* metódusokkal, melyek: *Start()/Stop().* Az állapotváltozások követéséhez létrehozott események: *Started, Stopped, StateChanged.* Az újabb forráskódban már létrehoztam egy *LedColor* tulajdonságot, mely már közvetlenül – az MVVM tervezési mintának köszönhetően – a felhasználói felülethez köthető.

- A *WaterPump* osztály – őse a *Pump* – feladata a permetlé készítéséhez szükséges víz szivattyújának vezérlése a víz kimérésével. A konstruktor a kapcsolódó *Relay* objektumon kívül elvár egy *FlowSensor* objektumot is paraméterként. Legfontosabb metódusa a *StartWithVolume()*, mely paraméterként a kimérendő víz mennyiségét

várja el. A metódus hatására a szivattyú elindul és a – *FlowSensor* objektum által mért – szükséges mennyiségnél leáll. A töltési folyamat a *VolumeChanged* eseményre történő feliratkozással folyamatosan követhető.

– A *DosingPump* osztály a *WaterPump* osztályhoz hasonló feladatokat lát el, de a peszticidek adagolását végzi perisztaltikus (vegyszeradagoló-)szivattyú vezérlésével. A konstruktornak a *Relay* mellett át kell adni egy – az irányváltást szolgáló – *DosingPumpDirection* objektumot, továbbá – műszaki/technikai információként – a *volumePerSecond* paraméterben a szivattyú által másodpercenként kiadagolt térfogat mértékét. A szivattyúzott mennyiség követését az osztály — tekintettel a perisztaltikus szivattyúk pontos működésére — időalapon valósítja meg. Az adagolás folyamata a *VolumeChanged* eseményen keresztül követhető.

– A *CircularPump* osztály – kapcsolódó szivattyú vezérlésével – lékeverési feladatot lát el. A konstruktor *Relay* objektumot vár el paraméterként. A *Pump* ősosztály által biztosított metódusok mellé implementált metódus a *StartAtVolume()*, mellyel a keverési folyamat a víztöltési folyamathoz köthető. A metódus – hívását követően – a paraméterként kapott *FlowSensor* objektumot figyelve a *startVolume* paraméterben megadott mennyiség elérésekor indítja el a lékeverési folyamatot.

– A *Valve* osztály – a *Relay* osztályhoz hasonlóan – rendkívül egyszerű, feladata mindössze a folyadékok áramlását szabályozó mágnesszelepek vezérlése egyszerű *Open()/Close()* metódusokkal. A konstruktor *Relay* paramétert vár el.

– A *FlowSensor* osztály feladata a 2.3.3. szakaszban bemutatott áramlásérzékelők jeleinek feldolgozása, a konstruktornak GpioController objektumot, a GPIO port számát, továbbá a – *pulsePerLiter* paraméterben, műszaki/technikai információként – az áramlásérzékelő által literenként küldött impulzusok számát kell átadni. A *VolumeChanged* eseményre történő feliratkozással követhető folyamatosan az átáramlott mennyiség. A *Reset()* metódussal a számlálóállás nullázható.

– Az *AM2315* osztály a Raspberry Pi I^2C interfészére csatlakoztatott, 2.3.4. szakaszban bemutatott hőmérséklet- és páratartalom-érzékelő szenzor adatainak olvasását végzi. A konstruktor a GpioController objektumot, továbbá az eszköz címét várja el paraméterként. A *Read()* metódus hatására az objektum *Temperature* és *Humidity* tulajdonságai frissülnek, a *Readed* esemény a sikeres olvasást követően, a *Changed* esemény bármelyik tulajdonság változásakor hívódik meg.

A fenti – gyakorlatilag teljesen újrahasznosítható – osztályokkal, mint „építőkockákkal" tetszőleges logikával felépíthetők a lékeverési folyamatok.

2.5. Eredmények

A „deszkamodell” bemutatásra került – ahogy korábban említettem – szakmai zsűri előtt a *HSUP Demo Day 2022* rendezvényen, továbbá a projektünk részt vett a Magyar Agrár- és Élettudományi Egyetem saját projektversenyén, az *Innovázión* is. Mind a zsűriknek, mind az azóta – leendő szakmai partnereknek – történt bemutatók alkalmával elismerték a projektet. Szakmai partnereink további ötletekkel és gazdaságosabb anyagbeszerzési források biztosításával támogatják a prototípus megvalósítását.

A Raspberry Pi mikroszámítógép alkalmazásával – GPIO tüskesorából adódóan – gyakorlatilag korlátlan lehetőségek nyíltak előttem. A szoftverfejlesztéshez használt .NET Core fejlesztői platform kellő stabilitást és biztonságot nyújt, az – általam az egyetemen megismert – kiváló C Sharp nyelv egyszerűvé teszi a stabil és hibatűrő multiplatformos alkalmazások fejlesztését. Úgy gondolom, hogy jól döntöttem!

A „deszkamodellhez” kiválasztott hardvereszközök, továbbá a szoftver kifejlesztéséhez alkalmazott technológiák beváltak. A „deszkamodell” minden bemutatásakor probléma nélkül működött, sem a hardverek, sem a szoftver működésében nem lépett fel zavar. Így úgy gondolom, hogy a tervezés és műszaki megvalósítás első szakasza sikerrel zárult.

2.6. Hogyan tovább?

Ahogy azt fényképekkel is bemutatom – a C. mellékletben – elkezdődött a prototípus felépítése már 1 m^3-es térfogatú víztartállyal és 200 literes térfogatú vegyszerkeverő tartállyal, dróntároló dobozzal, akkumulátortöltővel. A prototípus vezérlését biztosító eszközök „asztali” élesztése megtörtént (16. ábra), a szoftver fejlesztése – a folyamatosan felmerülő újabb igények és ötletek beépítése, a berendezés folyadékköreit érintő esetleges módosulások követése – folyamatban van.

Az *eszköz vezérlését biztosító áramkörökkel* kapcsolatban – a prototípus későbbi módosításával – szeretném megvalósítani a kötések stabilizálását. A jumper kábeleket szeretném kiváltani szalagkábellel és kimondottan az eszközhöz tervezett nyomtatott áramkör segítségével megvalósítani a Raspberry Pi GPIO tüskesorával történő kapcsolattartást. Szükség továbbá – az erősáramú áramkörök további módosításával, bővítésével (áramvédő relé, kismegszakítók stb.) – a villamos biztonsági feltételeknek történő megfelelés, melyet megfelelő végezttségű szakemberrel ellenőriztetnünk kell.

A *vezérlő szoftver* tekintetében szeretném, ha minél ergonómikusabb és tetszetősebb felület állna az operátor rendelkezésére, mindemellett – a vezérlő- és érzékelőosztályok refaktorálásával – programozástechnikai megoldásokkal szeretném stabilizálni, biztonságosabbá tenni a szoftver működését.

A projektgazda webes felülettel történő elérést is szorgalmaz, saját ötletem – elsősorban minőségbiztosítási, élelmiszerbiztonsági okok miatt – a permetezési folyamatok részletes dokumentálásának megvalósítása. Fontos lesz az interneten kesztüli adatközpon-

ti szinkronizáció, mellyel például léreceptek töltődnek le, továbbá feltöltésre kerülhetnek a permetezési adatok, időjárásinformációk, technikai/műszaki naplók. Minél hamarabb szeretném, hogy megvalósuljon a saját APT repo-n keresztüli szoftverfrissítés legfőképp azért, mert a prototípus tőlem távol, Békés vármegyében található, hamarosan induló tesztüzemeire is ott kerül sor.

A prototípus mindenképp megépül. Végső célunk az, hogy a permetező drón automatikusan magára az eszközre vagy az eszközhöz kapcsolt „dróntérre" szálljon le, ahol önállóan feltölti saját 10 literes tartályát, majd visszarepül és folytatja a permetezést. A piac részletesebb igényeinek felmérését követően folytatjuk a tervezést és fejlesztést, továbbá megkezdjük az eszköz sorozatgyártását.

3. Köszönetnyilvánítás

Sokat gondokodtam, hogy illik-e beletenni egy felsőoktatási szakképzés végén készülő portfólióba a könyvek végén szokásos köszönetnyilvánítást. Természetesen – mivel Google a barátunk – indítottam egy gyors keresést. A mobilom képernyőjét nézve, a tartalmat hirtelen mozdulatokkal húzva futottam bele az alábbi mondatokba:

> *„Ebben a rövid részben az író emberközelivé válik. Akár bármelyik ismerősöd, akivel nap, mint nap találkozol. Itt megismerheted, hogy olyan dolgokon megy át, mint Te vagy én. Ők sem tudnak mindent, számtalan ember segítette őket odáig, hogy megírták a könyvet. Kiadók, szerkesztők, grafikusok, kutatók, egyéb szakemberek, akik nélkül valóban nem írta volna meg a könyvet."*
>
> (Szlafkai 2017)

Fentieket olvasva – bár korántsem egy nagy terjedelmű, hónapokig/évekig készülő tanulmányt írtam – úgy érzem, köszönetet illik mondanom a hozzám régóta közel álló, továbbá a MATE Károly Róbert Campus-án megismert embereknek, akik támogattak a tanulmányaim, a munkám, továbbá a képzés éveinek időszakában az életem során.

Köszönettel tartozom annak, akit az élet nekem küldött. Aki szeretetével, erejével és bölcsességével támogat és véd mindentől, mindenkitől. Aki a sziget az élet tomboló viharában, aki mellett mindig menedékre találok: a páromnak, a „Kiflentyűmnek"...

Köszönettel tartozom a egyetemi tanáraimnak, akik hol szigorral, hol elnézéssel, kiváló tanárokként mindent elkövettek annak érdekében, hogy piacképes tudást szerezzek. Akik sikerrel elérték azt is, hogy az általam fölöslegesnek vélt tantárgyak és témakörök értelmére leljek, a megszerzett tudást gyakorlatba ültessem.

Köszönettel tartozom az évfolyamtársaimnak, a „tankörtársaimnak", akikket együtt tanultunk, vizsgáztunk, nevettünk, sírtunk. Köszönöm a felejthetetlen, soha véget nem érő csoportos videochat-eket. Köszönöm, hogy segíthettelek/tanítgathattalak benneteket. Minket már összeköt az élmény! Köszönöm az új barátságokat! Köszönöm a „keménymagnak" (Nagy Krisztián, Mészáros Árpi), hogy azóta is velem vagytok.

Köszönettel tartozom a szüleimnek, a barátaimnak és más ismerőseimnek aki bíztattak, türelmükkel és bátorításukkal támogattak.

4. Jegyzékek, hivatkozások

4.1. Ábrák jegyzéke

4.2. Táblázatok jegyzéke

4.3. Kódok jegyzéke

4.4. Irodalomjegyzék

Kaku, Michio (2011). *A jövő fizikája*. Akkord. ISBN: 9789632520681.

Kurzweil, Ray (2014). *A szingularitás küszöbén*. Ad Astra, 21. old. ISBN: 9786155229251.

4.5. Hivatkozások (internetes irodalom)

AvaloniaUI OÜ (2023). *Documentation for Avalonia UI*. URL: https://docs.avaloniaui.net/ (elérés dátuma 2023. 04. 02.).

Campbell, Scott (2016). *Basics of the I2C Communication Protocol*. URL: https://www.circuitbasics.com/basics-of-the-i2c-communication-protocol/ (elérés dátuma 2023. 04. 11.).

Glover, Dave (2023). *Build Raspberry Pi .NET Core IoT Applications running on Raspberry Pi OS or Ubuntu 20.04*. URL: https://github.com/gloveboxes/Create-RaspberryPi-dotNET-Core-C-Sharp-IoT-Applications (elérés dátuma 2023. 03. 10.).

Hawkins, Matt (2012). *Simple Guide to the Raspberry Pi GPIO Header*. URL: https://www.raspberrypi-spy.co.uk/2012/06/simple-guide-to-the-rpi-gpio-header-and-pins/ (elérés dátuma 2022. 10. 09.).

J., Byron és Shawn Hymel (2023). *Raspberry Pi SPI and I2C Tutorial*. URL: https://learn.sparkfun.com/tutorials/raspberry-pi-spi-and-i2c-tutorial/all#i2c-on-pi (elérés dátuma 2023. 04. 11.).

MagroPontHu (2020). *Tényezők, amelyek befolyásolhatják a növényvédő szerek hatékonyságát*. URL: https://www.magro.hu/agrarhirek/tenyezok-amelyek-befolyasolhatjak-a-novenyvedo-szerek-hatekonysagat/ (elérés dátuma 2023. 03. 03.).

Microsoft Learn (2022a). *Delegates (C Programming Guide)*. URL: https://learn.microsoft.com/en-us/dotnet/csharp/programming-guide/delegates/ (elérés dátuma 2023. 04. 19.).

— (2022b). *Model-View-ViewModel (MVVM)*. URL: https://learn.microsoft.com/hu-hu/dotnet/architecture/maui/mvvm (elérés dátuma 2023. 04. 19.).

— (2022c). *Object-Oriented programozás (C)*. URL: https://learn.microsoft.com/hu-hu/dotnet/csharp/fundamentals/tutorials/oop (elérés dátuma 2023. 04. 21.).

— (2023a). *Blink an LED*. URL: https://learn.microsoft.com/en-us/dotnet/iot/tutorials/blink-led (elérés dátuma 2023. 03. 01.).

— (2023b). *Read values from an analog-to-digital converter*. URL: https://learn.microsoft.com/en-us/dotnet/iot/tutorials/adc (elérés dátuma 2023. 03. 21.).

Monk, Simon (2023). *Configuring I2C - Adafruit's Raspberry Pi Lesson 4. GPIO Setup*. URL: https://learn.adafruit.com/adafruits-raspberry-pi-lesson-4-gpio-setup/configuring-i2c (elérés dátuma 2023. 04. 11.).

Nemzeti Kutatási, Fejlesztési és Innovációs Hivatal (2022). *Hungarian Startup University Program*. URL: https://hsup.nkfih.gov.hu/ (elérés dátuma 2022. 10. 09.).

Raspberry Pi Trading Ltd. (2019). *Raspberry Pi 4 model B Datasheet*. URL: https://datasheets.raspberrypi.com/rpi4/raspberry-pi-4-product-brief.pdf (elérés dátuma 2023. 04. 09.).

— (2021). *Raspberry Pi 4 model B Product Brief*. URL: https://datasheets.raspberrypi.com/rpi4/raspberry-pi-4-product-brief.pdf (elérés dátuma 2023. 04. 02.).

— (2023). *Raspberry Pi for industry*. URL: https://www.raspberrypi.com/for-industry/ (elérés dátuma 2023. 04. 11.).

ReactiveUI Association Inc. (2023). *ReactiveUI - View Models*. URL: https://www.reactiveui.net/docs/handbook/view-models/ (elérés dátuma 2023. 04. 21.).

SparkFun Learn (2023). *I2C*. URL: https://learn.sparkfun.com/tutorials/i2c/all (elérés dátuma 2023. 04. 11.).

Szlafkai, Éva (2017). *Mostantól mindig el fogod olvasni, mert a könyv lelke – Köszönetnyilvánítás*. URL: https://365letszikra.hu/koszonetnyilvanitas-iro-tamogatas/ (elérés dátuma 2023. 04. 25.).

Wikimédia Alapítvány (2023a). *I^2C (szócikk)*. URL: https://hu.wikipedia.org/wiki/I%C2%B2C (elérés dátuma 2023. 04. 11.).

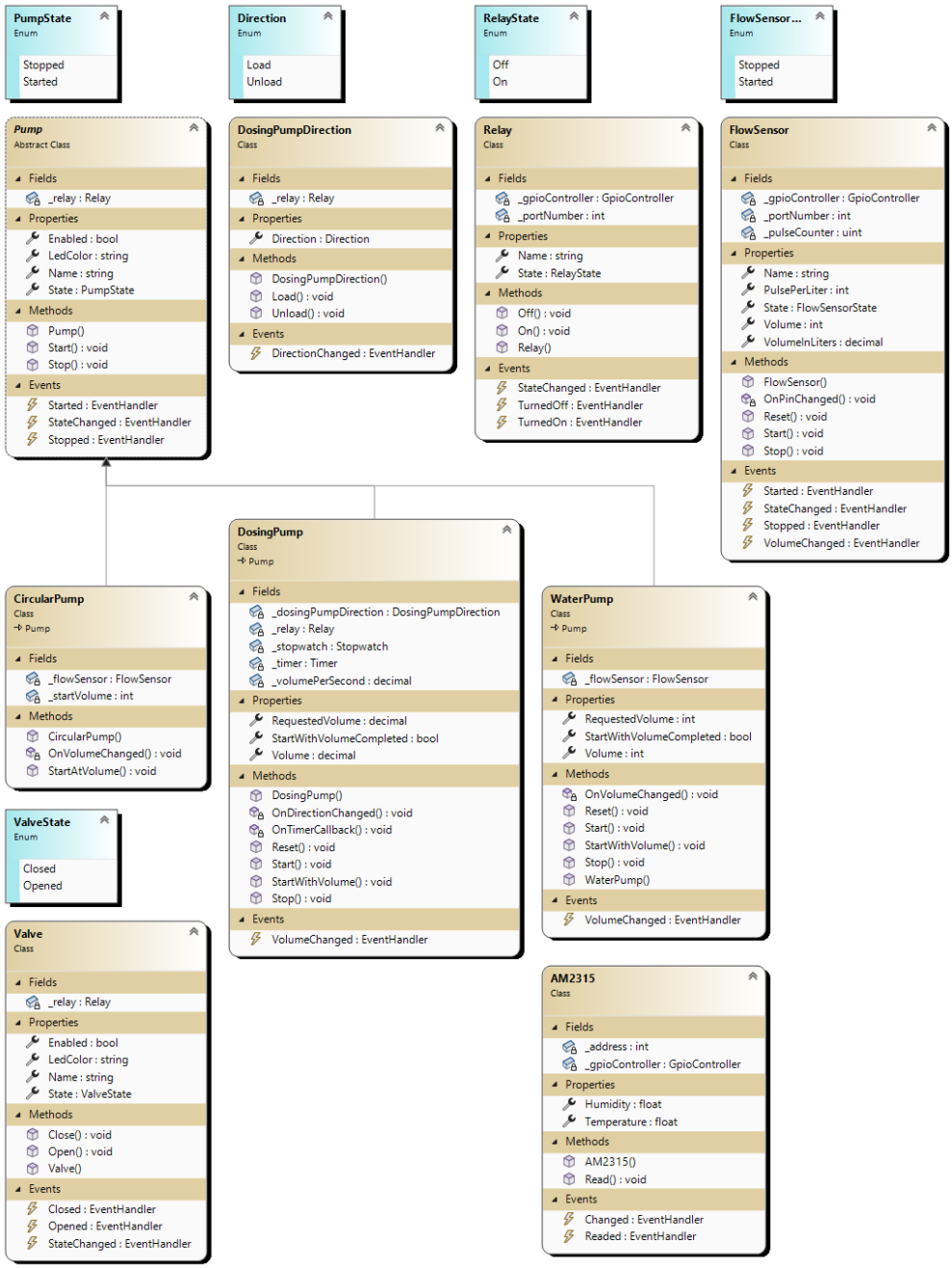
— (2023b). *Raspberry Pi (szócikk)*. URL: https://hu.wikipedia.org/wiki/Raspberry_Pi (elérés dátuma 2023. 03. 01.).

— (2023c). *Raspberry Pi OS (szócikk)*. URL: https://en.wikipedia.org/wiki/Raspberry_Pi_OS (elérés dátuma 2023. 03. 15.).

Mellékletek

A. Diagram az AgroDroneSystem Dashboard releváns osztályairól

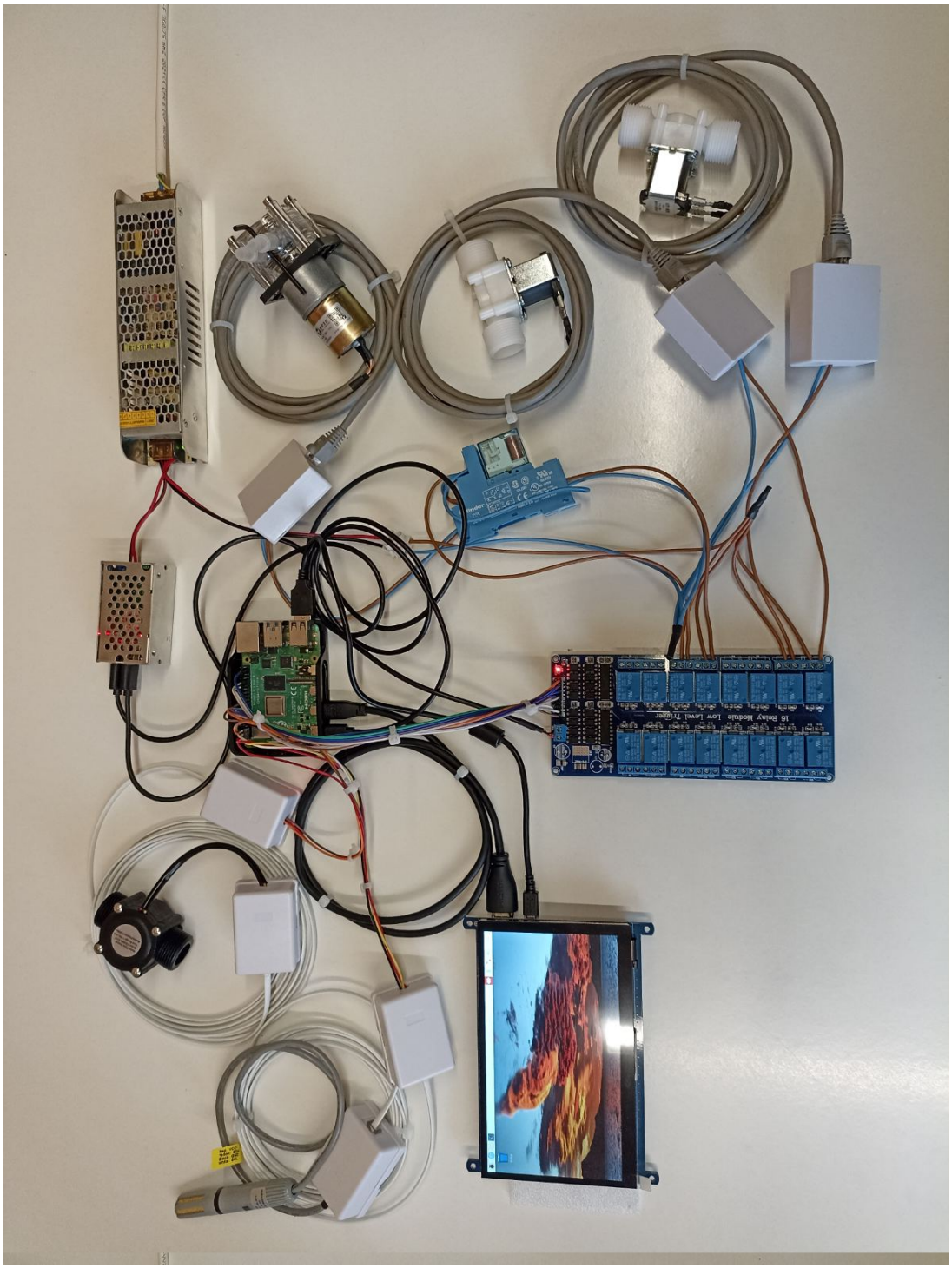
15. ábra. Az AgroDroneSystem Dashboard releváns osztályai



Forrás: saját

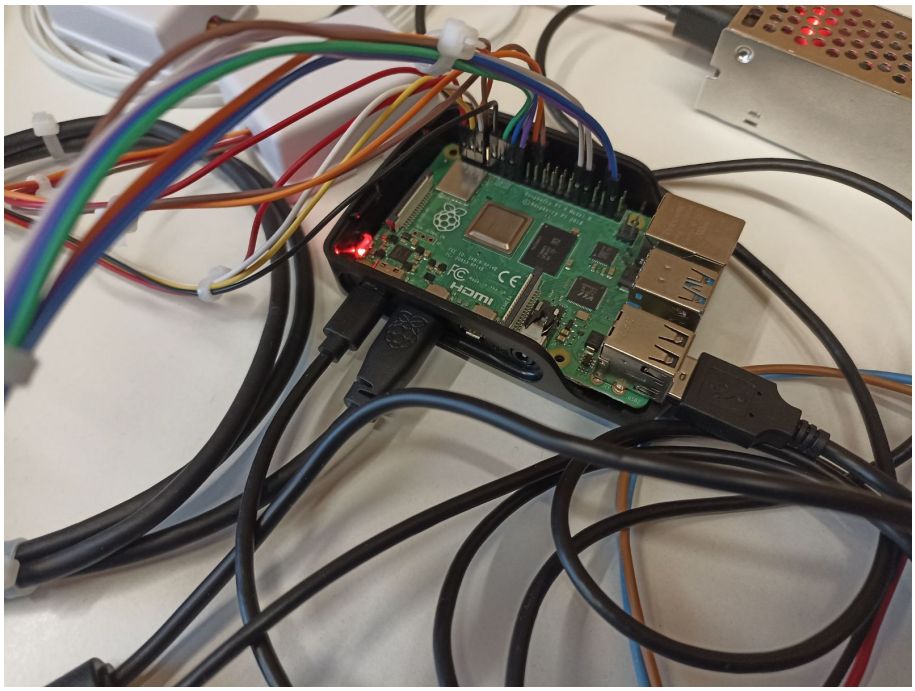
B. Fotók a hardverelemekről

16. ábra. A prototípus élesztése az „asztalon”, gyengeáramú elemekkel



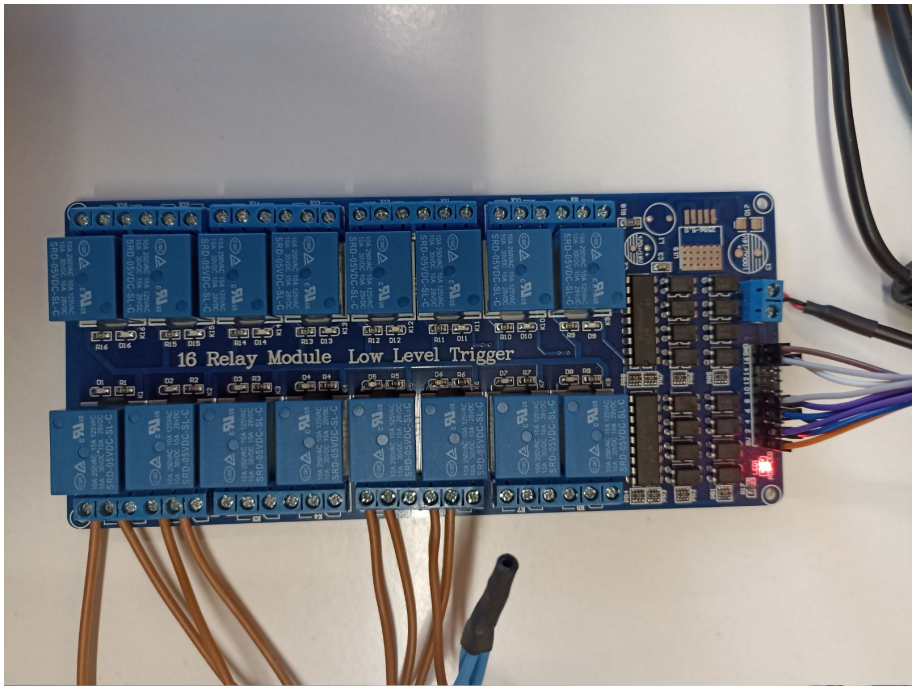
Forrás: saját

17. ábra. A Raspberry Pi jumper vezetékekkel



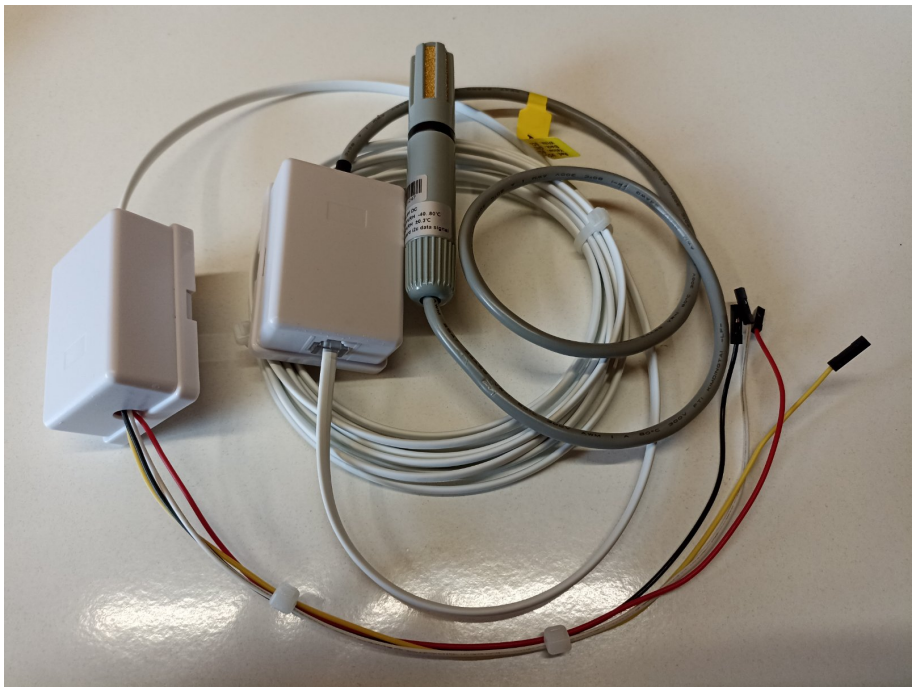
Forrás: saját

18. ábra. A relémodul



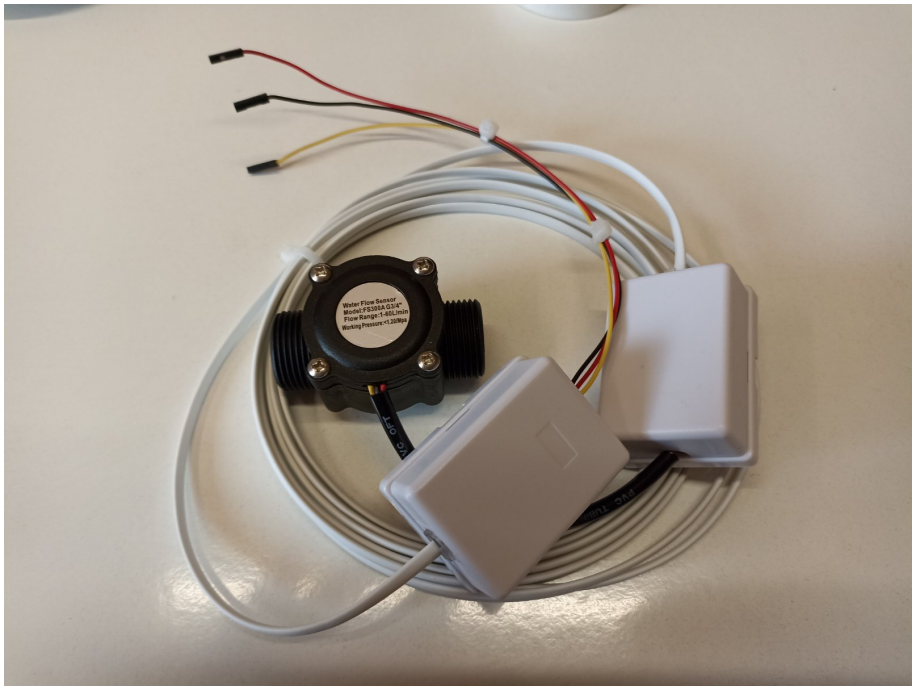
Forrás: saját

19. ábra. A „felvezetékelt” AM2315 érzékelő



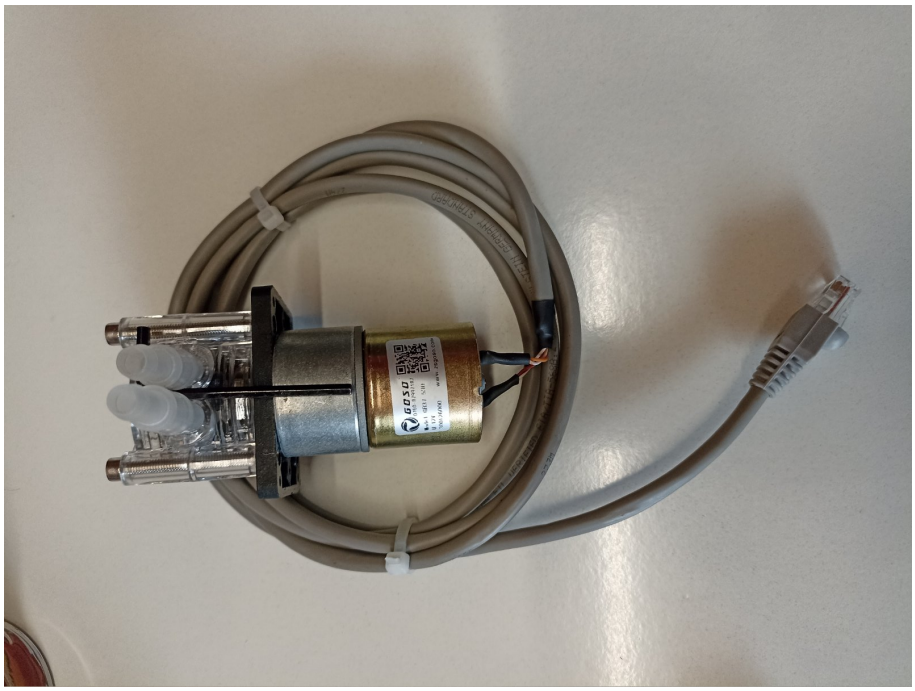
Forrás: saját

20. ábra. A „felvezetékelt” áramlásérzékelő



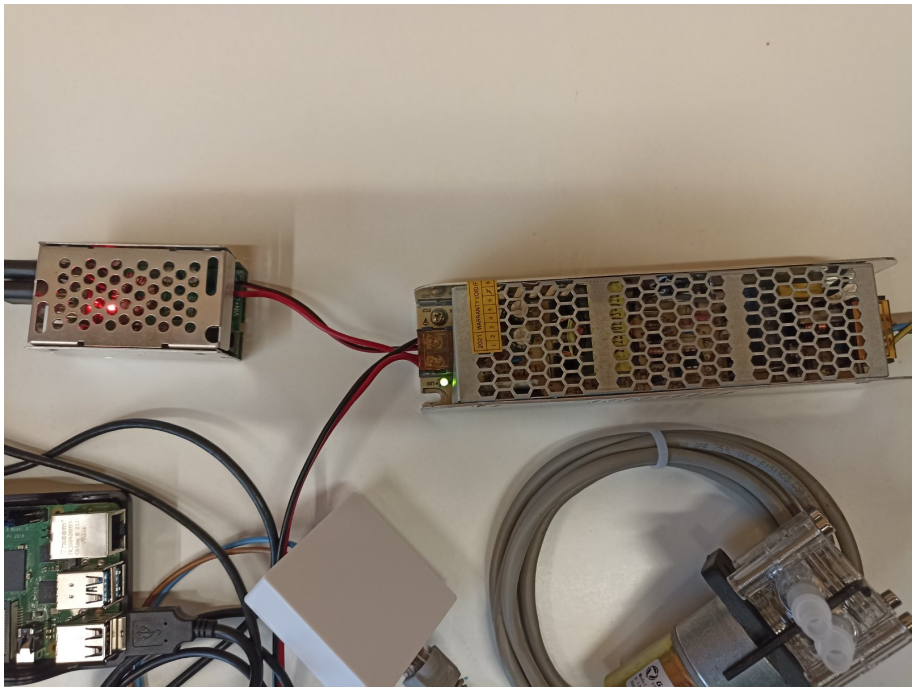
Forrás: saját

21. ábra. A „felvezetékelt” perisztaltikus (vegyszeradagoló-)szivattyú



Forrás: saját

22. ábra. A tápegységek



Forrás: saját

23. ábra. A „felvezetékelt” mágnesszelepek

Forrás: saját

C. Fotók a készülő prototípusról

24. ábra. A készülő prototípus (1)

Forrás: Pintér Bertold

25. ábra. A készülő prototípus (2)

Forrás: Pintér Bertold

26. ábra. A készülő prototípus (3)

Forrás: Pintér Bertold

27. ábra. A készülő prototípus (4)

Forrás: Pintér Bertold

D. Feltöltött önálló munkák (mellékletek)

A portfóliómhoz csatolt önálló munkák:

1. `Kő-papír-olló.sb3` – a Programozás módszertani alapjai tárgyunk keretében készült Scratch játék, mely a https://scratch.mit.edu/projects/460008583/ URL-en online is elérhető Kő-papír-olló játék.

2. `Nagy Attila Sándor (AIBWCZ) - Beadando_28.tar.gz` – a Magasszintű programozási nyelvek I. tárgyunk során, C Sharp nyelven készült beadandó feladat.

3. `ko-papir-ollo-master.tar.gz` – a Magasszintű programozási nyelvek II. tárgyunkhoz C Sharp nyelven, Windows Forms segítségével készült beadandó feladat.

4. `ads-eredeti-master.tar.gz` – a Hungarian Startup University Program II. tárgy keretében készült AgroDroneSystem Dashboard „korai” verziójának forráskódja, mely azóta már több alkalommal refaktorálásra került.

NYILATKOZAT
a portfólió[1] nyilvános hozzáféréséről és eredetiségéről

A hallgató neve:	**Nagy Attila Sándor**
A hallgató Neptun azonosítója:	**AIBWCZ**
A dolgozat címe:	**Portfólió**
A megjelenés éve:	**2023**
A konzulens tanszék neve:	**Alkalmazott Informatika Tanszék**

Kijelentem, hogy az általam benyújtott portfólió[2] egyéni, eredeti jellegű, saját szellemi alkotásom. Azon részeket, melyeket más szerzők munkájából vettem át, egyértelműen megjelöltem, s az irodalomjegyzékben szerepeltettem.

Ha a fenti nyilatkozattal valótlant állítottam, tudomásul veszem, hogy a záróvizsga bizottság a záróvizsgából kizár és a záróvizsgát csak új dolgozat készítése után tehetek.

A leadott dolgozat, mely PDF dokumentum, szerkesztését nem, megtekintését és nyomtatását engedélyezem.

Tudomásul veszem, hogy az általam készített dolgozatra, mint szellemi alkotás felhasználására, hasznosítására a Magyar Agrár- és Élettudományi Egyetem mindenkori szellemitulajdon-kezelési szabályzatában megfogalmazottak érvényesek.

Tudomásul veszem, hogy dolgozatom elektronikus változata feltöltésre kerül a Magyar Agrár- és Élettudományi Egyetem könyvtári repozitori rendszerébe.

Kelt: **Boconád, 2023. május 3.**

Nagy Attila Sándor
egyetemi hallgató

[1] A megfelelő dolgozattípus meghagyása mellett a többi típus törlendő.
[2] A megfelelő dolgozattípus meghagyása mellett a többi típus törlendő.

KONZULTÁCIÓS
NYILATKOZAT

A **Nagy Attila Sándor** (név) (hallgató Neptun azonosítója: **AIBWCZ**) konzulenseként nyilatkozom arról, hogy a portfóliót[1] áttekintettem, a hallgatót az irodalmi források korrekt kezelésének követelményeiről, jogi és etikai szabályairól tájékoztattam.

A portfóliót a záróvizsgán történő védésre javaslom / nem javaslom[2].

A dolgozat állam- vagy szolgálati titkot tartalmaz: igen nem*[3]

Kelt: **2022.** év **október** hó **26.** nap

Belső konzulens

[1] A megfelelő dolgozattípus meghagyása mellett a többi típus törlendő.
[2] A megfelelő aláhúzandó.
[3] A megfelelő aláhúzandó.