



ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

**NEUMANN JÁNOS
INFORMATIKAI KAR**



SZAKDOLGOZAT

**OE-NIK
2021.**

Hallgató neve:
Hallgató törzskönyvi száma:

**Bánhidi Péter
T-005037/FI12904/N**

SZAKDOLGOZAT FELADATLAP

Hallgató neve: **Bánhidi Péter**
Törzskönyvi száma: T/005037/FI12904/N
Neptun kódja: 

A dolgozat címe:

IOT kisállat etető rendszer és android applikáció fejlesztése
IOT pet feeder system and android application development

Intézményi konzulens: Somlyai László
Külső konzulens:

Beadási határidő: 2021. december 15.

A záróvizsga tárgyai: Számítógép architektúrák
Beágyazott rendszerek és Mobil informatika
specializáció

A feladat

Napjainkban egyre elterjedtebb, hogy különböző szenzorokkal mérjük, valamint befolyásoljuk környezetünket. Legyen szó értékes növényekről, vagy nekünk kedves állatokról, amiknek szüksége van a rendszeres ellenőrzésre és valamiféle emberi beavatkozásra a túléléshez. Szinte minden háztartásra jut egy-egy háziállat. Ezeknek az állatoknak fontos lehet a rendszeres és precízebb adagolás etetés során, és ha távol maradunk otthonról (pl. nyaralás, hosszú műszakos munka) ez nehézkes feladattá válik, még akkor is, ha rövidebb időről van szó.

Tervezzon és valósítson meg egy egyszerűen kezelhető, költséghatékony és kompakt rendszert, ami erre a problémára kínál megoldást. Tervezzon egy mikrokontroller alapú beágyazott rendszert, amit egy Android alapú alkalmazással, interneten keresztül lehet elérni. Fontos, hogy bárholonnan, bármikor, a felhasználó biztonságosan hozzá tudjon férni a rendszerhez. A kapcsolat TCP/IP-n keresztül legyen megvalósítva, amit egy saját, helyi autentikációval tegyen biztonságosabbá. Annak érdekében, hogy minél szélesebb körben használható legyen az alkalmazás, Android operációs rendszerre tervezzon az alkalmazást. Részletesen dokumentálja a tervezés, fejlesztés, megvalósítás és tesztelés egyes lépéseit.

A dolgozatnak tartalmaznia kell:

- a megoldandó probléma pontos leírását és részletes elemzését,
- a témában megjelent projektek, irodalmak, valamint ezek vizsgálatát,
- a mikrokontroller által vezérelt szenzorok, modulok működését,
- az Android alkalmazás működését,
- a két eszköz közötti kommunikáció felépítését,
- a rendszertervet, valamint vázlatot,
- a megvalósítás menetét,
- a fejlesztés során felmerült problémákat, ezek megoldását,
- az elkészült rendszer és alkalmazás felhasználási-, és tovább fejlesztési lehetőségeit.



FC R

Dr. Fleiner Rita
intézetigazgató

A szakdolgozat elévülésének határideje: **2023. december 15.**
(OE TVSz 55.§ szerint)

A dolgozatot beadásra alkalmasnak tartom:

.....
külső konzulens

.....
intézményi konzulens



ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

Neumann János Informatikai Kar

HALLGATÓI NYILATKOZAT

Alulírott hallgató kijelentem, hogy a szakdolgozat / diplomamunka saját munkám eredménye, a felhasznált szakirodalmat és eszközöket azonosíthatóan közöltem. Az elkészült szakdolgozatomban / diplomamunkámban található eredményeket az egyetem és a feladatot kiíró intézmény saját céljára térítés nélkül felhasználhatja.

Budapest, 2021. 12. 15.

Bánkúdi Péter

.....
hallgató aláírása



KONZULTÁCIÓS NAPLÓ

Hallgató neve: Neptun kód: Tagozat:
Bánhidi Péter Nappali.....

Telefon: Levelezési cím (pl: lakcím):
.....

Szakdolgozat / Diplomamunka¹ címe magyarul:

IOT kisállat etető rendszer és android applikáció fejlesztése

Szakdolgozat / Diplomamunka² címe angolul:

IOT pet feeder system and android application development.....

Intézményi konzulens: Külső konzulens:
Somlyai László

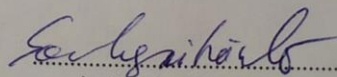
Kérjük, hogy az adatokat nyomtatott nagybetűkkel írja!

Alk.	Dátum	Tartalom	Aláírás
1.	2021.01.28.	Címmódosítás, kérdések átbeszélése	Somlyai László
2.	2021.02.24.	Állapotjelentés, kérdések átbeszélése	Somlyai László
3.	2021.03.25.	Feladatlap, kérdések átbeszélése	Somlyai László
4.	2021.05.12.	Tartalmi követelmények, formázás, kérdések	Somlyai László

A Konzultációs naplót összesen 4 alkalommal, az egyes konzultációk alkalmával kell láttamoztatni bármelyik konzulenssel.

A hallgató a Szakdolgozat I. / Szakdolgozat II. (BSc) vagy Diplomamunka 1 / Diplo-mamunka 2 / Diplomamunka 3 / Diplomamunka 4³ tantárgy követelményét teljesítette, beszámolóra / védésre⁴ bocsátható.

Budapest, 20. 21. 5. 14.


.....
intézményi konzulens



KONZULTÁCIÓS NAPLÓ

Hallgató neve: Bánhidi Péter Neptun kód: Tagozat: Nappali
Telefon: Levelezési cím (pl: lakcím):
.....

Szakdolgozat / Diplomamunka¹ címe magyarul:

IOT kisállat etető rendszer és Android applikáció fejlesztése

Szakdolgozat / Diplomamunka² címe angolul:

IOT pet feeder system and Android application development

Intézményi konzulens: Külső konzulens:

Somlyai László

Kérjük, hogy az adatokat nyomtatott nagybetűkkel írja!

Alk.	Dátum	Tartalom	Aláírás
1.	2021.09.25.	Kérdések - válaszok	Somlyai László
2.	2021.12.01.	Állapotjelentés, kérdések átbeszélése	Somlyai László
3.	2021.12.10.	Tartalmi követelmények	Somlyai László
4.	2021.12.14.	Tartalom, végső formázás, kérdések	Somlyai László

A Konzultációs naplót összesen 4 alkalommal, az egyes konzultációk alkalmával kell láttamoztatni bármelyik konzulenssel.

A hallgató a Szakdolgozat I. / Szakdolgozat II. (BSc) vagy Diplomamunka 1 / Diplo-mamunka 2 / Diplomamunka 3 / Diplomamunka 4³ tantárgy követelményét teljesítette, beszámolóra / védésre⁴ bocsátható.

Budapest, 20. 21. 12. 14.

Somlyai László
intézményi konzulens

Tartalomjegyzék

1	BEVEZETÉS.....	1
2	ALTERNATÍV MEGOLDÁSOK ÉS HASONLÓ RENDSZEREK	2
2.1	ALTERNATÍV MEGOLDÁSOK	2
2.1.1	Baráti / szomszédi segítség	2
2.1.2	Kisállat-szitter	2
2.1.3	Állatbarát szálláshellyel	3
2.1.4	Kisállatpanzió	3
2.1.5	Állatorvosi felvigyázás.....	3
2.1.6	Alternatív megoldások összefoglalása.....	3
2.2	KISÁLLAT ETETŐ GÉP ARDUINO ESZKÖZ FELHASZNÁLÁSÁVAL / (PET FOOD AUTOFEEDER BY USING ARDUINO).....	4
2.2.1	Automata kisállat etető Arduino használatával / (Automatic Pet Feeder using Arduino)...	5
2.3	AUTOMATA ARDUINOVAL MŰKÖDŐ KISÁLLAT ETETŐ / (AUTOMATIC ARDUINO POWERED PET FEEDER).....	6
2.4	OKOS KISÁLLAT GONDOZÓ RENDSZER / (SMART PET CARE SYSTEM USING INTERNET OF THINGS).....	8
2.5	OKOS KISÁLLAT ETETŐ / (SMART PET FEEDER).....	10
2.6	OKOS KUTYA ETETŐ, VEZETÉKNÉLKÜLI KOMMUNIKÁCIÓVAL, MQTT TECHNOLÓGIÁVAL ÉS ANDROID KLIENSSEL / (SMART DOG FEEDER DESIGN USING WIRELESS COMMUNICATION, MQTT AND ANDROID CLIENT).....	11
2.7	PROGRAMOZHATÓ KISÁLLAT ETETŐ / (PROGRAMMABLE PET FEEDER).....	12
2.8	AUTOMATA KISÁLLAT ETETŐ / (AUTOMATIC PET FEEDER).....	13
2.9	NURIPET: EGY OKOS KISÁLLAT ETETŐ GÉP, SNS-EL / (NURIPET: A SMART PET FEEDING MACHINE FOR SNS)	14
2.10	OKOS KISÁLLAT ETETŐ / (SMART PET FEEDER).....	16
2.11	PIACON MEGVÁSÁROLHATÓ TERMÉKEK	17
2.12	KONKLÚZIÓ	18
3	RENDSZER FUNKCIÓK ÉS ARCHITEKTÚRA	20
3.1	A RENDSZERARCHITEKTÚRA LEÍRÁSA	20
3.1.1	Otthoni Környezet	20
3.1.2	Távoli Környezet.....	20
3.1.3	Kommunikáció.....	20
3.2	A RENDSZER FUNKCIÓI	20
3.2.1	Automatikus Etetés	20
3.2.2	Itató rendszer	21
3.2.3	Lokális bemeneti interfész.....	22
3.2.4	Felhő adatbázis	22
3.2.5	Eledel mérleg.....	22
3.2.6	Felügyelet kamerával.....	23
3.2.7	Hő és páratartalom mérés	23
3.2.8	Mobil Applikáció.....	23
3.2.9	Rendelkezésre állás	24
4	ISMERT SEBEZHETŐSÉGEK	25
4.1	FIZIKAI (LOKÁLIS) JELLEGŰ	25
4.2	HÁLÓZATI (TÁVOLI) KOMMUNIKÁCIÓN KERESZTÜL	26
4.3	APPLIKÁCIÓ BIZTONSÁGA	26

5	RENDSZERTERV	27
5.1	KAMERA	28
5.2	VEZÉRLÉS, MIKROKONTROLLER	29
5.3	ETETŐ GÉP	32
5.4	ITATÓ GÉP.....	33
5.5	SZENZOROK.....	34
5.5.1	RTC	34
5.6	ULTRAHANGOS TÁVOLSÁGÉRZÉKELŐ	34
5.6.1	Hő és páratartalom mérő	35
5.7	APPLIKÁCIÓ	35
5.8	TÁPELLÁTÁS.....	36
5.9	FELHŐ, HÁLÓZATI KOMMUNIKÁCIÓ.....	36
5.10	LOKÁLIS INTERFÉSZ.....	36
5.11	TERVEZETT KÖLTSÉGVETÉS.....	37
6	MEGVALÓSÍTÁS.....	38
6.1	APPLIKÁCIÓ	38
6.1.1	Felépítés	38
6.1.2	Navigáció	38
6.1.3	Kezdőképernyő.....	39
6.1.4	Funkciók képernyő	40
6.1.5	Beállítások képernyő	42
6.1.6	A felhasználónak való visszajelzések.....	43
6.1.7	Egyéb technikai Információk.....	43
6.2	FELHŐ SZOLGÁLTATÁS.....	43
6.3	OKOS KISÁLLAT ETETŐ.....	45
6.3.1	Fő vezérlés.....	46
6.3.2	Konfiguráció feldolgozása	48
6.3.3	Etető Rendszer	48
6.3.4	Itató Rendszer	52
6.3.5	Szenzorok	55
6.3.6	Kamera.....	55
6.3.7	Lokális felhasználói interfész.....	57
6.3.8	Teljes Rendszer és Kapcsolási rajz	57
6.3.9	Költségvetés	60
7	TESZTELÉS.....	61
7.1	APPLIKÁCIÓ TESZT	61
7.2	ETETŐ RENDSZER TESZT	HIBA! A KÖNYVJELZŐ NEM LÉTEZIK.
7.3	ITATÓ RENDSZER TESZT	HIBA! A KÖNYVJELZŐ NEM LÉTEZIK.
7.4	FIZIKAI JELLEMZŐK	64
8	TOVÁBBFEJLESZTÉSI LEHETŐSÉGEK	66
9	ÖSSZEFOGLALÓ	68
10	SUMMARY.....	68
11	IRODALOMJEGYZÉK.....	69

12	ÁBRAJEGYZÉK.....	74
----	------------------	----

1 BEVEZETÉS

A probléma, amivel a szakdolgozatomban foglalkozni szeretnék az életem során már többször felmerült. Egész kicsi korom óta számos háziállatot tartunk itthon. Többnyire macskákat és kutyákat. Némelyek kint vannak a kertben egész nap, de akad egy-két kedvenc, amik bent laknak, szobatiszták és csak alkalom adtán keredzhetnek ki.

A kinti macskák tudnak gondoskodni magukról, hol itthon, hol a szomszédnál vacsoráznak. Ezzel ellentétben, a benti állataink igencsak igénylik a segítségünket, törődésünket.

Gyakran előfordul, hogy a családunk elutazik hosszabb időre, és nem marad itthon senki, aki etetni tudná őket. Ilyen esetekben a benti macskákat a rokonokhoz visszük.

Mivel kiszakítjuk őket a természetes élőhelyükből, idegen környezetbe kerülnek, nem érzik biztonságban magukat, rosszul viselkednek, kárt okozhatnak.

Ennek elkerülése érdekében, átvisszük a holmijaikat, ágyukat, eledelüket, kiskönyvüket. Sajnos ez mindig nagyon körülményes, ráadásul igen kellemetlenné tud válni, ha rendszeresen kell a rokonainkat megkérnünk erre a szívességre.

Ez a probléma a saját mindennapi életem során merült fel először, de kutatásaim során hamar meggyőződtem róla, hogy mivel több százmillió [1] kutyát és macskát tartanak az emberek világszerte, így számos másik háztartásban is okozhat ez gondot. Emiatt tehát érdemes foglalkozni a kérdéssel, hiszen ilyen nagy létszámú célcsoportnál könnyedén felmerülhet az igény ennek orvoslására.

Egy másik szempont, hogy vannak olyan kisállatok, fajtájukat vagy állapotukat illetően, amelyeknek nélkülözhetetlen az étel rendszeres, precíz adagolása, mind mennyiségben, mind pedig időpontban. Esetleg további környezeti körülmények is lényegesek lehetnek, mint például a hőmérséklet és páratartalom ellenőrzése.

Dolgozatomban ezekre a problémákra szeretnék egy alternatív megoldást találni, ami modern, költséghatékony és több szempontból is kedvezőbb lehet mind a gazdik, mind a kiskedvencek számára.

Ahogy fejlődik a világ, úgy az IoT (Internet of Things) eszközök is egyre nagyobb teret nyernek. Egyre több és újfajta szolgáltatást tudnak kiegészíteni, vagy akár helyettesíteni. Ez a projekt erre egy tökéletes példa.

Szerencsére az Óbudai Egyetemen a Beágyazott Rendszerek szakirányon lehetőségem volt megismerkedni különböző mikrokontrollerekkel és számos elektronikai modullal, mint például a DC (egyenáramú) motor, különféle érzékelők, kapcsolók és egyéb építőelemekkel, így ezt a tudást felhasználva fogok nekiállni a feladatnak.

2 ALTERNATÍV MEGOLDÁSOK ÉS HASONLÓ RENDSZEREK

Szakdolgozatom ezen fejezetében a már meglévő módszerekkel, illetve a saját megoldásomhoz hasonló rendszerekkel fogok foglalkozni. Először vizsgálom a természetes, mindennapi ötleteket, melyek nem igényelnek okos eszközt, majd rátérek az olyan projektmunkákra melyekben már megjelenik az IoT és egyéb technológiák alkalmazása is. Végezetül, áttekintem a napjainkban megvásárolható termékeket is.

2.1 ALTERNATÍV MEGOLDÁSOK

Állatokat, kisállatokat már évezredek óta tartanak az emberek. Ezek többnyire haszonállatok voltak. „A domesztikálás vagy magyarul a háziasítás folyamata megközelítő adatok szerint 11000 éve kezdődött és több meghatározó ok vezérelte. Elsődlegesen, mint táplálék (hús, tej, tojás) és eszközforrás (ruha és szerszám) alapanyag volt jelen, a másodlagosan a mezőgazdasági, őrző-védő, szállító és katonai célok szolgálása, a harmadlagos szerepe pedig a kialakult kötődés volt az állat társaságához. (Zeder, M. A. (2012))” [2] Azonban a modern életnek és az urbanizációnak köszönhetően mára már megváltozott a hétköznapi állattartás túlnyomó szerepe. Napjainkban az emberek több mint a fele tart hobbiból állatot. [1] Az ilyen fajta állattartás általában rendszeres figyelmet igényel a gazdától. Ez mindaddig nem okoz fennakadást amíg a gazdának nem kell elutaznia vagy hosszabb időt távol tölteni az otthonától, és így a kiskedvencétől is. A vizsgált probléma leginkább a szabadidő növekedésével együtt jelent meg, ugyanis az emberek egyre többet utaztak, illetve a munkába járás és az egyedi beosztások (pl. 24 és 48 órás műszak) is jelentős időt vesznek el az otthonlétből amikor a kisállat egyedül marad. Ezért különböző alternatív megoldásokat kellett alkalmazniuk a gazdiknak, hogy megfelelő módon tudjanak gondoskodni az állataikról. A következő alfejezetekben bemutatom az alternatív lehetőségeket, illetve megvizsgálom azok előnyeit és hátrányait.

2.1.1 BARÁTI / SZOMSZÉDI SEGÍTSÉG

Sok állattartó egyszerűen megkér valakit (pl. szomszédot, barátot, családtagot), hogy rendszeres időnként lássa el (etesse és itassa meg) kiskedvencét. Ha az állat nagy figyelmet igényel akkor arra az időre vagy odaköltözik, vagy magával viszi az ismerős a kiskedvencet. Ez a megoldási forma mindig baráti szívességen alapul, cserébe akár az állattartó is ugyanúgy vigyáz a másik kedvencére. Az ilyen baráti együttműködéseknek több formája van. Például lehet kölcsönös, amikor felváltva mennek el nyaralni és vigyáznak egymás állataira. Ez az alternatíva azonban nem működőképes, ha nincs olyan személy, akire rá merjük bízni kiskedvencünket vagy akikre igen, azok nem érnek éppen rá.

2.1.2 KISÁLLAT-SZITTER

Ez a módszer egy fokkal formálisabb az előzőnél, ugyanis valamilyen ellenszolgáltatással jár. Több változata is létezik. Legegyszerűbb az, amikor a kisállat-szitter jön a mi otthonunkba, így az állatnak nem kell utaznia. Ebben az esetben olykor az is lehetőség szokott lenni, hogy az ellenértéket nem pénzben, hanem szállásban nyújtják az illetőnek,

például nyaralás céljából. De van olyan is, aki önként, pusztán az állatbarát-természete miatt vállalja, hogy gondoskodik a kiskedvencünkről. Ennek ellenkező változata, hogyha mi visszük kiskedvencünket a kisállat-szitter házába. Ezek a változatok már olcsóbbak és természetesebb környezetet nyújtanak, mint egy kisállatpanzió. [3]

2.1.3 ÁLLATBARÁT SZÁLLÁSHELLYEL

Sokan úgy oldják meg ezt a kérdést, hogy ha elutaznak otthonról, magukkal viszik kedvenceiket. Ez több szempontból is bonyolult. Nem minden szálláshely fogad állatokat, de ha mégis sikerül egy ilyet találni, többnyire magas felárral kínálják. Emellett sok állat nem bírja az utazással járó megterheléseket. Tehát ez nem jelent tökéletes megoldást mindenki számára.

2.1.4 KISÁLLATPANZIÓ

Egy másik lehetőség a kisállatpanziók vagy kisállatnapközik attól függően, hogy mennyi időre szeretnénk kiskedvencünket elszállásoltatni. Az ilyen létesítményekben professzionális szolgáltatásra számíthatunk, de ez általában elvárás is az igénybe vevő gazdik részéről. Itt megtalálhatóak olyan eszközök, berendezések, amelyekkel jól gondját tudják viselni az állatoknak. Azonban megeshet, hogy kiskedvencünket hosszabb ideig is ketrecben tartják, mint amennyit szeretnénk. Azt is figyelembe kell venni, hogy kedvencünk egy ilyen helyen rengeteg másik állattal kapcsolatba léphet akár csak közvetve is, így előfordulhat a fertőzés veszélye. Ha emellett az opció mellett döntünk valószínűleg magasabb költségekre számíthatunk, mint a többi alternatívánál, illetve fontos, hogy informálódjunk a kiszemelt hely referenciáiról és először csak egy rövidebb próbaidőre adjuk be a kisállatunkat. [4]

2.1.5 ÁLLATORVOSI FELVIGYÁZÁS

Ez egy ideális megoldás lehet akkor, ha az állat nincs jól, egészségügyi problémái vannak vagy kezelés alatt áll. Így nem kell aggódnunk állatunkért, ha különleges figyelmet igényel. Ugyanakkor ez a választás stresszes környezetet jelent az állat számára, ezért csak végső lehetőségként ajánlják, és csak rövid időintervallumra. [4]

2.1.6 ALTERNATÍV MEGOLDÁSOK ÖSSZEFOGLALÁSA

Összegezve a fentebb felsorolt alternatív megoldásokat arra a következtetésre jutottam, hogy egyes opciók olykor nélkülözhetetlenek, de ha nem vagyunk bekorlátozva (pl. beteg a kisállat) akkor döntésünket megelőzően mérlegelnünk kell a lehetőségek előnyeit és hátrányait, ugyanis mindegyiknek van ez is, az is.

Az említett megoldások egy része nagyobb költségekkel jár, illetve a kisállatot kiszakítjuk a megszokott élőhelyéből, ami rossz hatással lehet rá. Ez különösen igaz a macskáknál. Rengeteg előnye van még annak, ha a kisállat otthon marad. Ilyen például, hogy a játékok, az udvar, a ketrec stb. már mind-mind adottak, rendelkezésre állnak, így az állat magabiztosan kiismeri magát, nyugodtabb marad. Egy másik előny, hogy nem tud kárt tenni idegen emberek holmijában, a már bevált eledelt kapja, továbbá nem áll fenn annak

a veszélye, hogy más állatokkal versengene, harcolna. Ezért, ha állatbarátabb opciót szeretnénk választani, ahol a kiskedvencnek nem kell elhagynia a természetes környezetét, otthonát, akkor egy lehetőség lehet, ha segítséget kérünk ismerőstől vagy valamilyen kisállat-szittert hívunk, aki házhoz jön. Ám ez szervezést igényel és egyéb kellemetlenségekkel, költségekkel járhat. [3][4]

Emiatt, és mert mint már említettem a saját életemben is tapasztaltam ezt a problémát, úgy gondolom, hogy szükség lenne egy olyan ellátási rendszerre, ami elhelyezhető a kisállat természetes élőhelyén, és amit távolról is lehet irányítani, így a gazdi is nyugodt lehet, hogy kiskedvence megkapta a megfelelő mennyiségű táplálékát.

A következő részben az ezen elven működő, már létező rendszereket tanulmányozom. Az alapvető funkciókat és az egyéb kulcsfontosságú jellemzőiket fontos, hogy részletesebb vizsgálat alá vessem, hogy így könnyebben, pontosabban tudjam meghozni a döntést arra vonatkozóan, hogy mely technikák, megoldások illeszkednek legjobban a saját rendszerem igényeihez.

2.2 KISÁLLAT ETETŐ GÉP ARDUINO ESZKÖZ FELHASZNÁLÁSÁVAL / (PET FOOD AUTOFEEDER BY USING ARDUINO)

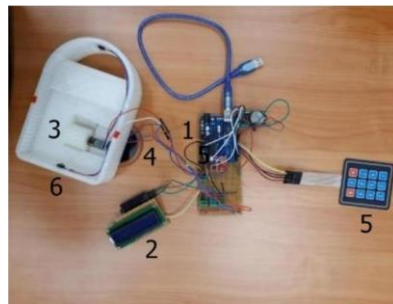
Ebben a cikkben [5] egy kisállat-etető gép koncepcióját mutatják be. Az eszköz fő célja, hogy helyettesítse az alapesetben ember által végzett etetési folyamatot. Ezt úgy teszi, hogy közben ügyel az etetések pontos, előre definiált időpontjaira, valamint az ételadag mennyiségekre. Ezeket az értékeket a felhasználó állíthatja be, a kisállat fajtájától, méretétől függően.

Az etetőgép megoldást nyújthat olyan állattulajdonosoknak, akik gyakran és/vagy sokat vannak távol otthonról és emiatt úgynevezett állathotelekbe, vagy ismerősökhöz kellene elhelyezniük kiskedvencüket. Emellett még segítséget kínál azoknak is, akinek nehézséget okoz a precíz adagok és rendszeres etetési időpontok betartása. Ez különösen fontos, ha a kisállat például orvosi diétán van, és emiatt fokozottan ügyelni kell az időzítésre és a pontos adagolásra, vagy akár, ha a gazda feledékeny lenne.

A projekt az etetőrendszer mechanizmusának tervezését és kifejlesztését is részletezi. Az alkatrészek közé tartozik egy mikrokontroller, egy óra, egy LCD kijelző, egy DC motor, egy hangszóró, egy 4x3-as billentyűzet az adatbevitelhez, illetve egy kapcsoló. Mikrokontrollernek az Arduino egy típusát alkalmazták, amely vezérli a rendszert, és eltárolja a használni kívánt időpontokat. Ezen időpontok felhasználásához egy Real Time Clock, azaz valós idejű órát használtak. Ez az óra gombjelekről kapja az energiát és első beüzemeléskor be kell rajta állítani a pontos időt.

A prototípus dizájnjáról egy részletes vázlatot is készítettek. Magára az etető mechanizmusra külön figyelmet szenteltek. Az eledel adagolásáért, egy fogaskerék által mozgatott, egyenes fogazású fogasléc felel, melynek végén egy kör alakú lyuk található. Ezen a lyukon esik le az eledel a tartályból. Ezeket az elemeket az ANSYS nevű szoftverben modellezték. A fogaskerék meghajtásáért a DC motor felel.

Számos tényezőt felhasználva, analizálták a várható strapabírást, amit a megtervezett konstrukciónak ki kell állnia. Különös hangsúlyt fektettek az ételadagoló nyílás mechanizmusának vizsgálatára. Ezen alkatrész megépítéséhez két anyagot is analizáltak deformáció, alakváltozás, illetve különböző mechanikai feszültségek szempontjából. Ennek eredményeként a HDPE műanyagra, azaz a nagy sűrűségű polietilénre esett a választás. Emellett prototípust is készítettek, melynek mérete 1:5 arányú. Ezen elméleti és gyakorlati tesztek is végrehajtottak, és ezek eredményeit kellőképpen kivizsgálták annak érdekében, hogy az etetési folyamat gond nélkül menjen végig. Ezen analízis egyfajta útmutatást nyújtott annak érdekében, hogy az etetőgép megfelelően, az előre eltervezett, végső formáját érje el. A megvalósításról képet (2-1. ábra) és kapcsolási rajzot is mellékeltek. A prototípus a



2-1. ábra: Kisállat etető gép Arduino eszköz felhasználásával

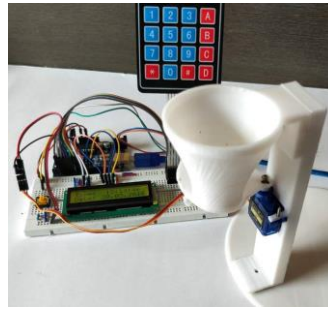
Felhasználói felület szempontjából egy, az etetőn megtalálható lokális interfészről van szó. Egy 12 gombos billentyűzet és egy LCD kijelző áll rendelkezésünkre. Az egyes gombok segítségével kell beállítanunk a pontos időt, majd a fixen kettő etetési alkalom időpontjait. A harmadik gomb funkciójával lehetőség van az adagok számát megnövelni, egészen 9-ig. Ezáltal az eledel mennyisége szabályozható. Ugyanakkor a fogasléc oda-vissza mozgatásával létrejött adagolás nem feltétlenül precíz. Emiatt a kiöntött táp körülbelüli mennyiségét, egy ötvenszer megismételt kísérlettel vizsgálták, melynek eredményeként kiderült, hogy az etető viszonylag konzisztensen tud 9.8g és 10.1g közötti értékeket adagolni, ami nagyon jó eredménynek számít.

Előnye a strapabíró szerkezet és a kiszámítható tápmennyiség, illetve, hogy nincs hálózati sebezhetősége. Egyik hátránya az adagolási módból következően, hogy csak a fix adag többszöröseit képes produkálni, de ha nagyobb mennyiségről van szó ez nem feltétlenül lesz pontos. Emellett, a fogasléc mechanikájából kifolyólag, a megfelelő strapabíró anyaghasználatra ügyelni kell, illetve nincsen távoli elérés, így nem ellenőrizhető, konfigurálható, ha nem vagyunk a helyszínen.

2.2.1 AUTOMATA KISÁLLAT ETETŐ ARDUINO HASZNÁLATÁVAL / (AUTOMATIC PET FEEDER USING ARDUINO)

Az előző kiadványban szereplő rendszerhez hasonlót készítettek el ebben a házi projektben [6] is. A kétféle etető célja ugyanaz, ezért az sem meglepő, hogy a felhasznált alkatrészek némi különbséggel egyformák. Ugyanakkor az ételadagolók nyitó-záró mechanikája eltérő. Míg az előbbinél egy fogaskerekes fogaslécet, azaz időintervallumos megoldást alkalmaztak, addig ebben a projektben egy szervo-motor segítségével az

ételtartó alatt egy lyukas panelt mozgatnak, mely megoldásra inkább az ismétlésszám jellemző, mintsem az időtartam.



2-2. ábra: Automata Kisállat Etető Arduino Használatával

Egy apróbb különbség még az anyaghasználat, ugyanis amíg az előző kiadványban HDPE-t, itt PLA műanyagot használtak. Ez a két anyag leginkább hőtulajdonságaikban tér el egymástól. Az etető mind a négy részét 3D nyomtatóval készítették. A modelleket elérhetővé tették a weboldalon, így bárki letöltheti őket. A projekt megvalósítását képekkel, kapcsolási rajzzal és forráskóddal is dokumentálták lépésről, lépésre, illetve a már működő állapotáról videót is készítettek.

Az előző rendszerhez képest előnyt jelent, hogy itt nem DC motort, hanem szervomotort alkalmaztak, ami sokkal precízebb mozgásokra alkalmas. Ugyanakkor, a kiengedett eledel mennyiségéről csak becslést adhatunk, az előre lemerített átlagok statisztikája alapján, ugyanúgy, mint az előbbi etetőnél.

2.3 AUTOMATA ARDUINOVAL MŰKÖDŐ KISÁLLAT ETETŐ / (AUTOMATIC ARDUINO POWERED PET FEEDER)

A következő projekt, amit ebben a fejezetben vizsgálunk [7], szintén egy, az előzőekhez nagyon hasonló kisállatetetőgép. A gép megalkotóját a saját, mindennapi életének egy problémája ihlette meg. A macskája rendszerint felkeltette őt reggelente, étkezését követelve. Ennek a kellemetlenségnek a megszüntetése érdekében, egy automatikus állatetető rendszer kifejlesztése mellett döntött. Fontos volt, hogy az etető előre megadott időpontban, minden külső segítség nélkül, meg tudja etetni macskáját, így megkönnyítve gazdája reggeljeit. Kezdeként megvizsgálta a helyi, angliai kínálatot, de nem volt megelégedve vele. Állítása szerint az ezen termékek egyik felét nehéz és macerás beprogramozni, a másik felével pedig sajnos csak kevés, fix számú etetés lehetséges. Második opcióként az amerikai választékot is megtekintette, de a magas extra költségek és a körülményes szállítás miatt inkább úgy döntött, megalkotja a saját kisállatetetőjét. A projekt a probléma és a saját motiváció bemutatása után a gép felépítését és annak komponenseit ismerteti.

A főbb alkatrészek közé tartozik egy RC szervó, egy földvájóra hasonlító fúrószár és egy PVC cső, amik az eledel szállításáért felelnek. Az eledelt tároló tartályba, szemmel láthatólag keverési célból, egy másik RC szervót helyezett a készítő. A bemeneti interfész

egy inkrementális enkóderből, egy nyomógommból, illetve egy 16x2-es LCD képernyőből áll, amin egyszerű felhasználói felületet láthatunk. Szükséges még egy tartály az állateledel tárolásához, illetve egy farostlemez doboz, a készülék házához. Az irányítás egy Arduino Uno-val van megoldva, az idő ellenőrzését pedig egy RTC, azaz egy valós idejű óra végzi. A szervók egy 12V-os tápegységről üzemelnek. A készítő által készített fotók a 2-3. ábrán láthatóak.



2-3. ábra: Automata Arduinoval Működő Kisállat Etető

Érdekes kitérni az eledelt szállító mechanikára. A munkát egy folyamatos szervomotor végzi, amire egy csiga elvű, fűrészszerű forgórész lett rögzítve. Ennek a helye egy T idomú PVC csatlakozás, aminek a függőleges lyuka az eledel tartályához kapcsolódik, ahonnan a táp beleesik a gravitáció miatt a csőbe. Innentől pedig a fűrészszerű forgórész gondoskodik róla, hogy az eledel elinduljon a cső kimeneti nyílása felé. A fűrészszerű forgórész 3D nyomtatással készült, és úgy illeszkedik a csőben, hogy a fűrészszerű forgórész lapátjának távolsága a cső falától kisebb, mint a tápszemek nagysága, amiket eltol, így azok nem ragadnak be, nem maradnak a csőben. A fűrészszerű forgórész nyomtatáshoz szükséges háromdimenziós objektumot, illetve egyéb tervek fájljait és a projekt egész forráskódját is megosztotta. Ehhez magyarázatot is írt a szerző.

Ennek a megoldásnak az előnye egyértelműen az, hogy precízen, kontrolláltan tudjuk az eledelt adagolni. Míg az előző módszereknél a gravitáció segítségével és a nyílások lezárásával szabályozták a táplálék áramlását, addig itt a gravitáció csak a csigára, a csiga köré juttatja el a tápot. Miután a táp bekerült a csőbe, onnantól egyedül a csiga mozgása a mérvadó, és mivel szervót használtak a projektben, szinte tetszőleges sebességgel meghajthatják a csigát. Ezzel igen pontos mennyiségeket is produkálhat az etető, viszont hátránya, mint ahogyan az az előző projektekben, úgy itt sincsen semmiféle visszajelzés, az eddig kiadagolt eledel mennyiségéről. Emiatt az első beüzemeléskor, illetve, ha új mennyiséget szeretnénk beállítani, kísérleteznünk kell különböző időtartamokkal vagy sebességekkel, és ki kell tapasztalni, hogy mi a nekünk megfelelő mennyiségű egység. Egy másik sajátos szempontja, hogy a tápszemek ne akadjanak meg a csiga és a PVC cső között, emiatt gondos tervezést igényel a két alkatrész méretezésének összehangolása.

2.4 OKOS KISÁLLAT GONDOZÓ RENDSZER / (SMART PET CARE SYSTEM USING INTERNET OF THINGS)

Ebben a fejezetben egy harmadik kisállatetető projektet vizsgálok [8], a már említett rendszerekkel összehasonlítva. A projekt az előzőkhez képest egy másik főbb funkcióval is rendelkezik, ugyanis nemcsak az etetésről, hanem az alom tisztántartásáról is gondoskodik. A kész verzióról képeket is megosztottak (2-4. ábra)



2-4. ábra: Okos Kisállat Gondozó Rendszer

A szerző kifejti az IoT eszközök fejlődését és lehetőségeit. Ez is egy motivációja a projektjének. Azt kutatták, hogy milyen új szolgáltatásokat tudnának könnyedén a valós életükben felhasználni. Választásuk a célcsoportot illetően a kisállatos emberekre esett, ugyanis szerintük egyre több háztartásban van házikedvenc. Ugyanakkor elkerülhetetlen, hogy a gazdák olykor elhagyják otthonukat, így magukra hagyva kisállataikat. Ilyen esetben sokszor alternatív megoldásokhoz fordulnak, melyek költségesek és kiépített bizalmat igényelnek, mint ahogy azt már az Alternatív Megoldások című fejezetemben én is kifejtettem. Itt látták a lehetőséget egy IoT szolgáltatásban, amivel magabiztos megoldást tudnak kínálni az említett problémára. Továbbá a piacon akkortájt jelenlévő termékek zöme az időzített automata etetésen kívül, csak egy megfigyelő kamerával volt ellátva. Annak érdekében, hogy egy okos kisállatgondozó rendszert tudjanak nyújtani mind a kisállatnak mind a gazdájának, több dologra is gondolniuk kell. Gondoskodniuk kellett arról, hogy egy okostelefon segítségével a gazdi ellenőrizni tudja a jószág állapotát, és hogy a megfelelő ellátások a kisállat rendelkezésére álljanak. Ilyen szolgáltatás az alomtisztítás, ami a gazdi szemszögéből szintén egy fontos kérdés, illetve az etető tál újratöltése. Az architektúra az alábbiak mentén épül fel. A rendszerben megtalálható egy szerver, ami a különböző alkatrészekkel való kommunikációért felelős. Ezeket a komponenseket a következő bekezdésben részletezem. Magát a kiszolgáló gépet egy laptopal illusztrálták az ábrán, ami a helyi WIFI hálózatra csatlakozik, ezért a többi komponensnek is szükséges valamiféle WIFI/Ethernet modul a kommunikációs csatorna szerepére. Ugyanakkor megemlítik, hogy ez a kommunikációs csatorna nincs lekorlátozva a WIFI-re, felcserélhető más technológiákra, mint például Bluetooth vagy LTE hálózat. A szerver a helyi hálózatról elérhető, de router segítségével az internetre is csatlakozik, ahol egy okostelefon és a rajta levő applikációval akár manuálisan is

irányíthatjuk a komponenseket, valamint megtekinthetjük az általuk, a szerveren már korábban eltárolt adatokat.

Az említett komponensek közé tartozik az automatikus etető, az automata alom és a kamera. Az etetőben mérleget és időzítőt alkalmaznak. A mérleggel az eledel kívánt mennyiségét ellenőrizhetik, az időzítővel beállíthatják az etetések közti intervallumot. Kiegészítésként a mérleggel menthetik a kívánt mennyiséget egy esetleges kézi adagolást követően. A vezérlést a felhasználó okostelefonjáról lehet elvégezni. Az etető szervomotor segítségével működik, amit egy Arduino mikrokontroller irányít. Ugyanakkor az étel mozgatásának, adagolásának mechanikájára nem térnek ki. Egyedül a mérleget említik, amivel az adagok mennyisége bekalibrálható egy egyszeri mérés alapján, illetve az időzítőt, amivel megadható, milyen időközönként indítson etetést a rendszer.

A következő komponens az öntisztító alom. Manapság sok gazdi megtanítja kisállatát, hogy különféle almokba végezze dolgát. Azonban mikor a gazdi hosszabb időre elmegy, ezek az almok könnyedén eltelítődhetnek és tisztításuk, cseréjük nem megoldható. Erre a problémára próbáltak megoldást nyújtani az öntisztító alommal. Itt hőmérőt, nedvességszenzort és ultrahangos távolságmérőt is felhasználtak. A berendezés detektálja az állat jelenlétét, majd távozását, illetve ürülékét, és egy léptető motorral elfordul, így a rajta lévő salakanyag beesik egy speciális tároló dobozba. A motort itt is Arduino-val vezérik. Ennek az implementációja tulajdonképpen egy ürülék elszállító futószalag, aminek a végére valamilyen tárolót, például egy vödröt tehet a gazdi.

A harmadik alkatrész a kamera, amit már Raspberry Pi mikrokontrollerrel implementáltak. A kamera fő funkciója, hogy lássuk a kisállat állapotát és mozgását. Mellékfunkciójának tekinthetjük az etető és az alom monitorozását, segítségével ugyanis ezek állapotát is láthatjuk. Ezen három komponens irányítására és monitorozására az okostelefonos applikációt szánták. A felhasználó egy azonosító és jelszó párossal férhet hozzá az alkalmazás funkcióihoz. Az etetést és alom ürítését a kamera a szerveren rögzíti, így ezek utólag is megtekinthetők. Az eszközök hálózati beállításai csak rendszergazda jogosultsággal érhetők el.

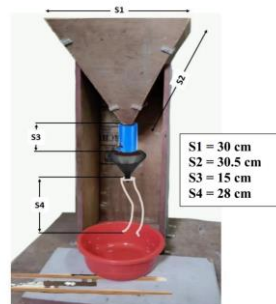
Az implementációt illetően, az etető házát 3D nyomtatóval hozták létre. Az öntisztító almot nehezen tudták volna kinyomtatni, ezért kartondobozból építették meg.

Mivel az etetési mechanikát, azon kívül, hogy szervomotorral működik, nem részletezték, így itt nem veszem az összehasonlítási szempontok közé. Amit előnyösnek gondolok ennél a rendszernél, az az, hogy etetéskor a mérleggel való visszacsatolásnak köszönhetően, pontosabban tudjuk megközelíteni a kívánt eledel mennyiségét. Ráadásul, a mérleget nemcsak az etetéskor, hanem a használat során, egy adag mennyiségének dinamikus beállítására is alkalmazhatjuk. A kamera alkalmazásával kényelmesen

monitorozhatjuk a különböző komponensek állapotát, ellenőrizhetjük, hogy azok gond nélkül, rendeltetésszerűen működnek. Ezáltal biztosak lehetünk benne, hogy az állat megkapta az eledelét, ami jelen esetben az egyik legfontosabb biztonsági szempont. A kamerának még egy előnye, hogy nem csak a rendszert, de a kisállatot is szemmel tarthatjuk, mikor étkezik, vagy a közelben tartózkodik. Hátránya, hogy nem mozgatható el, így fix a látókép. Emiatt az irányát kellő odafigyeléssel érdemes beállítani, hogy azzal a lehető legtöbb dolgot lássuk, amit szeretnénk. Az öntisztító macskaalmot jó ötletnek tartom, de ez a kivitelezés szerintem nem a leghatékonyabb. Az ürülék halmazállapotától, mennyiségétől függően, a kartonpapír könnyen átázhat és a futószalag folyamatosan használatban levő felülete nem higiénikus többszöri használat után.

2.5 OKOS KISÁLLAT ETETŐ / (SMART PET FEEDER)

Ennek a projektnek [9] az egyik érdekessége, hogy az etető prototípusának különböző komponenseit, újra felhasználható, maradék és hulladék anyagokból tervezték megépíteni. A 2-5. ábrán látható egy félkész változat.



2-5. ábra: Okos Kisállat Etető

A rendszer vezérlését egy ATMEGA32 mikrokontroller végzi. Az etetések közötti időintervallumot a felhasználó egy 4x4-es billentyűzeten tudja betáplálni. Ebben egy LCD képernyő segít, ami a jelenlegi és a következő etetésig szükséges időt, illetve a tára és mérlegre nehezedő súlyt jeleníti meg. Az etető tetején található egy 30 és 30.5 cm hosszú élekkel rendelkező kúp, amiben az eledelt tárolja. Közvetlenül alatta, egy PVC csövön halad tovább az eledel, ahol egy szervomotorra erősített lapos, kerek zár kontrollálja az áramlást egy tölcserbe, amiből az, egy 28 cm-es cső segítségével a tálba érkezik. Ez a tál egy mérlegre van ráerősítve, mely folyamatosan működik. A rendszer addig adagolja az eledelt a tálba, amíg a mért súly el nem éri az előre definiált értéket. A prototípust egy AVR fejlesztői áramkörrel építették, tesztelték. Az építő elemek egytől egyig az intézmény selejtezett tárgyaiból készültek.

A rendszer paramétereit csak helyileg, a lokális interfészen lehet beállítani. Az időzítés alapelve, hogy előre definiált időintervallumonként etessen a gép, ami nem a legfelhasználóbarátabb megoldás. A mérleg és a szervomotor segítségével egy korrekt,

aprólékos áramlást lehet elérni, ugyanakkor a tölcsér végén levő cső, 28 cm-es hosszúsága miatt, mire az eledel leér a tálba és a mérleg jelez a vezérlésnek, hogy a tálban levő táplálék elérte a kívánt mennyiséget, addigra, ez idő alatt, nagy mennyiségű eledelt átengedhet a szervó. Ezt kivédhetnénk azzal, hogy a csőben lévő csapot el-el zárjuk, és mindig megvárjuk amíg a mérleg frissít, ám a projekt folyamatábrája szerint, a szervomotor addig nyitja ki a csapot, amíg a mérleg a célmennyiséget el nem éri.

2.6 OKOS KUTYA ETETŐ, VEZETÉKNÉLKÜLI KOMMUNIKÁCIÓVAL, MQTT TECHNOLOGIÁVAL ÉS ANDROID KLIENSSEL / (SMART DOG FEEDER DESIGN USING WIRELESS COMMUNICATION, MQTT AND ANDROID CLIENT)

Ebben a projektben szereplő etető rendszer [10] elsősorban a kicsi vagy közepes méretű kutyák etetésére lett megtervezve, viszont, a szerzők által behatárolt 12-30 cm-es méretkülönböztetés miatt könnyedén beleférnek a macskák, illetve egyéb kisállatok is. A rendszer felépítése úgy néz ki, hogy egy Arduino Uno mikrokontroller felelős az etető különböző komponensek irányításáért, begyűjti az adatokat a szenzoroktól és működteti a különböző kapcsolókat, mozgó alkatrészeket. Az Arduino Uno soros portjára lett kapcsolva a vezeték nélküli kommunikációért felelős ESP8266, mely NodeMCU fejlesztői csomaggal van ellátva. A NodeMCU MQTT protokollal tartja fent a kapcsolatot. Ha kap egy üzenetet a szervertől, azt továbbítja az Arduino UNO számára soros kommunikáción keresztül, ami ennek következtében teljesíti a megfelelő funkciót. A szervert egy Node.js scripttel készítették, és egy MySQL adatbázisba menti az adatokat.



2-6. ábra: Okos Kutya Etető

A gépről készített fotók a 2-6. ábrán láthatóak. Az etetés paramétereit az okostelefonunkból állíthatjuk be, illetve az etető által küldött értesítéseket is itt tudjuk megtekinteni. Az applikáció, a szerver és az Okos Kutya Etető közötti kommunikáció WiFi-n keresztül történik, ahol az üzeneteket MQTT protokollal közlik és fogadják, és TLS/SSL protokollal tették biztonságosabbá. Az applikáció oldalán, ez a Paho Android Service segítségével történik.

Az Okos Kutya Etető másik feladata ügyelni arra, hogy idegen kutyáknak ne legyen hozzáférése az eledelhez. Ennek érdekében, az etetés előtt egy autentikációs folyamat zajlik le RFID technológia segítségével. Ezt úgy próbálta megoldani a szerző, hogy az RFID olvasót az etető rendszerre rögzítette, az RFID chipet pedig a kutya nyakörvébe illesztette. Az itt alkalmazott modul egy RFID RC522 volt.

Egy próbaetetést követően az eredmények azt mutatták, hogy a megadott időponton az RTC pontosan indukálta a folyamatot, a szervomotor működött és addig öntötte az eledelt, amíg a mérlegen a kívánt mennyiséget el nem érte. A mérlegnek kalibrációra volt szüksége, ezt megtették a MatLab szoftver segítségével, így annak a mérései pontosak voltak. A maximális távolság, amin az autentikáció hiba nélkül tudott üzemelni, körülbelül 2,5 cm volt, így az RFID olvasót kifejezetten pontosan kellett elhelyezni az etető tál oldalára, hogy az megfelelően tudjon működni. Az etető mechanizmust a szervomotor által meghajtott propeller végzi, ami vízkerék jelleggel engedi át az eledelt a csövön a gravitáció segítségével. Az eledeltároló edény kapacitása durván 1 kg. A használatos kutyaeledel száraz és kör alakú. Az elérhető adagok körülbelüli mennyisége 76 g, 150 g, 227 g.

A mérésekből az derült ki, hogy a propeller két lapátja közé, nagyjából 20g eledel fért, tehát amennyiben a kívánt mennyiség nem osztható ezzel az értékkel, a kiöntött eledel meg fogja haladni a célmennyiséget. Annak is fennáll az esélye, hogy a propeller lapátok közé beszorul egy táp szem, amitől az etetési folyamat inkonzisztenssé válik. Ezenkívül, ha az etetési cikluskor a szervomotor éppen hátrányos pozícióban áll meg, a kerék lapátjai részben nyitott pozícióban maradhatnak, így egy lapátnyi adaggal több tápot enged ki a tálba. Összegezve, ez a technikai megoldás igencsak hátrányos, ha a pontos, precíz adagolás is egy szempont.

2.7 PROGRAMOZHATÓ KISÁLLAT ETETŐ / (PROGRAMMABLE PET FEEDER)

Ebben a kutatási munkában [11], a fenti projektekhez képest, igencsak eltérő etetési megoldással találkozunk. Míg az előző konstrukciókban, egytől egyig, valamilyen adagolási mechanikát alkalmaztak, addig a Programozható Kisállat Etető más elven oldja meg ezt a kérdést. A fő alkatrész a 2-7. ábrán látható.



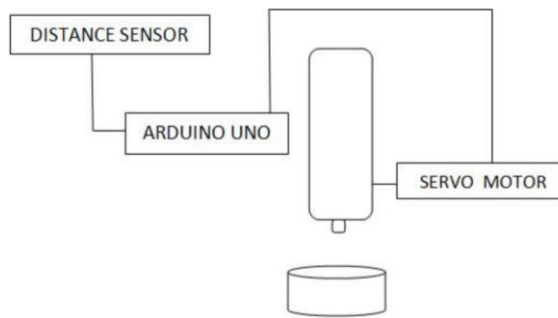
2-7. ábra: Programozható Kisállat Etető

A különbség abban rejlik, hogy a kisállat eledelét a felhasználó itt egy kör alakú forgóasztalnak a fix darabszámú, fiókszerű rendszerébe előre beadagolja. Az etető alakját kihasználva, egy léptető motor segítségével megoldható, hogy az állat előtt mindig csak egyetlen fiók legyen felfedve, így a többi eledelhez nem fér hozzá. Mikor eljön a következő etetés időpontja, a léptető motor forgat egyet az asztalon, így egy másik, még teli fiók kerül a nyitott pozícióba. Jelen esetben egy DC motort is alkalmaztak, ami, miután elfordult az asztal, kiadagolja az ételt a tálból. Ezt az eseményt a szerző nem részletezte, viszont kialakítástól függ, hogy kell-e. Ezekbe a fiókokba a felhasználónak előre kell bekészítenie az eledelt. Ebben a projektben az asztalt négy részre osztották, így ezzel 4 adag ételt tud biztosítani az állatnak. Az időzítést egy programmal tervezték megoldani. Miután elfordult az asztal, és az táplálék elérhetővé válik, az Etető egy buzzer segítségével jelez az állatnak, amit kifejezetten hasznosnak tartok. Egy pár alkalom és az állat megszokja a jelzést. Így mindig biztosak lehetünk benne, hogy az állat tud az etetésről, még akkor is, ha az eledel csörgését nem hallotta meg.

Előnye ennek a megoldásnak többek között, hogy nemcsak száraz, hanem nedves, illetve félszáraz táplálékot is rakhatunk az eledeltároló fiókokba. Persze ez a kialakításától is függ, illetve gondolni kell a nedves eledel szavatosságára, frissen tartására. Ami viszont kifejezetten hátrányos, hogy fix, és igen csekély alkalommal tudjuk vele kisállatunkat megetetni. Az adagokat nekünk kell kimérnünk kézzel, így azt az etetések során nem szabályozhatjuk. Ez a megoldás, ilyen méretben az adagoktól függően, maximum egy vagy két napra alkalmas. Ha az etetőt sikerülne nagyobb méretekben megvalósítani, elképzelhető lenne, hogy a kisállatnak naponta egy, nagyobb adagot tegyünk elérhetővé.

2.8 AUTOMATA KISÁLLAT ETETŐ / (AUTOMATIC PET FEEDER)

Ebben a projektben egy olyan etetőt mutatnak be [12], mely a kisállat detektálásakor automatikusan elindítja az etetést, és kiadagolja az előre beállított mennyiséget.

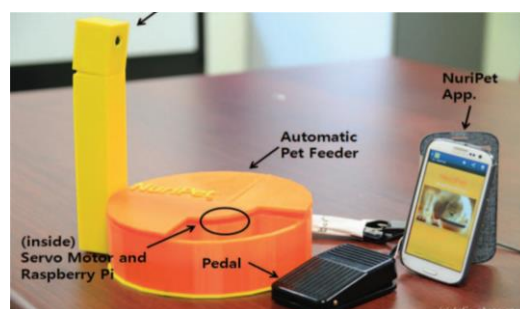


2-8. ábra: Automata Kisállat Etető Komponens Terv

Ezt egy távolságmérő vagy közelségérzékelő szenzorral kívánták megvalósítani. Ha az állat közeledik és egy bizonyos távolságba ér, az Arduino Uno elindítja a szervomotort, amely kiengedi az edelt a megfelelő időtartamig. Ez azt jelenti, hogy az állat annyiszor ehet egy nap, ahányszor csak szeretne. Erre úgy hivatkoznak, hogy „szabad” vagy „szabad döntésű” etetés. Ennek előnyeit, hátrányait egy későbbi fejezetben vizsgálom. Úgy gondolom, hogy a kamerát érdemes lenne elindítani akkor is, ha az állat nem étkezik, ehhez pedig felhasználható ez az ötlet is.

2.9 NURIPET: EGY OKOS KISÁLLAT ETETŐ GÉP, SNS-EL / (NURIPET: A SMART PET FEEDING MACHINE FOR SNS)

A projekt [13] koncepciója az, hogy az etetéshez a kisállatnak meg kell nyomnia egy pedált, mellyel nemcsak az edelt engedi ki a gép, de egy fényképet is készít az állatról. Ezt a folyamatot a gazdi is elindíthatja a mobil applikációból. A képet, melyet a kamera készít, a Raspberry Pi interneten keresztül továbbítja a NuriPet applikációba, ahol a gazdinak lehetősége van megosztani azt Facebookon sztoriként. Az applikáción a Facebookra ezt megelőzően be kell jelentkeznie a felhasználónak. A gép etető része hasonlít a 2.7-es fejezetben szereplő Programozható Kisállat Etető koncepciójára. A forgóasztalon egy nyílás van, az alatta található edel-tárolóban pedig négy fiók. Mikor a forgóasztalt a 360°-os szervomotor elfordítja, az állat hozzáfér egy másik fiókhoz, ahol a következő adag edel várja.



2-9. ábra: NuriPet: Egy Okos Kisállat Etető Gép, SNS-el

A NuriPet egyik motivációja többek között egy szívfacsaró történet volt, ami ráébreszti a gazdikat, hogy a kisállataik mindennapjairól minél több képet, videót érdemes készíteni, mert sosem lehet tudni, meddig tisztelnek még meg minket a jelenlétükkel. Az is köztudott, hogy rengeteg gazdi osztja meg nap mint nap a kisállataikról készített fotókat, videókat más gazdik, vagy állatbarát emberek között és ezt a folyamatot is szeretné automatizálni az elfoglaltabb gazdik számára.

A kutatás során a készítők meginterjúvoltak három állatorvost, akik az alábbiakat tanácsolták nekik. Először is, a fotók helyett inkább videókat készítsenek. Ebből több információ derül ki, mind a kisállat állapotáról, kedvéről, illetve biztonsági kameraként is funkcionálna. Egy másik ötlet volt, hogy az Okos Etetőhöz egyéb eszközöket, játékokat kellene kapcsolni valamilyen módon, így a kisállatok nem fogják magányosnak érezni magukat, illetve a gazdi is látja, hogy minden rendben. Azt is tanácsolták még, hogy az etető építőelemeit, főképp az étel tárolót, érdemes lenne valamilyen kitartóbb, masszívabb anyagból elkészíteni, ugyanis az állatok gyakran előre megpróbálják kiszedni ételüket saját maguk.

Az alap funkció az etetés és a képkészítés, viszont a koncepció alkalmas lehetne kisállat naplók, képkönyvek készítésére. Kereskedelmi szempontból a feltöltött képeket a gazdik megtudnák vásárolni kinyomtatva, illetve a különféle állateledel gyártóknak lehetőségük adódna reklámozni az eledeleiket a NuriApp-on keresztül, ahol a gazdik megrendelhetik azokat pár gombnyomással. Felmerültek egyéb fejlesztési ötletek is, mint például az élő videó hívás, hang üzenet, távolságérzékelőre aktiválódó fotó készítés.

A Kisállat Közösségi Média egy izgalmas ötlet, de a jelen feladatom szempontjából nem tűnik relevánsnak. Ezt a tovább fejlesztési ötletek között meg fogom említeni. Viszont egy ötletet adott arra a problémára, hogy ha a gazdi elutazott, és éppen nincs lehetősége megtekinteni az applikációt, lemarad a kisállat jelenlétéről, étkezéséről. Emiatt érdemes lehet felvenni a videót/képet későbbre a memóriában és akár elküldeni, a gazdi applikációjára, ahonnan visszanézhetővé válik.

Végül, ennek az okos etetőnek a mechanikájáról mondok véleményt. Legnagyobb előnye a dizájnnak, hogy nemcsak száraz, de nedves eledeleket is tárolhatunk benne kiskedvencünknek. A gazdi úgy adagolja, ahogyan neki tetszik, minden egyes napra egyedien. Precíz arányokat követve összekeverheti más tápokkal, vitaminokkal. Az egyenlő arányú elkeverést a gravitációs-adagolás etetők nem tudják biztosítani, csak akkor, ha a gazdi a tápot már a betöltés előtt összekeveri a hozzáadandó táppal, vitaminnal. A dizájn hátránya ugyanakkor, hogy a gazdinak saját magának kell kiadagolni, bekészíteni az eledelt, minden napra külön-külön. Ezek az etetők átlagosan 5-6 etetési alkalmat képesek kiszolgálni, így, ha a kisállat napi 2-3-szor étkezik, akkor ez a darabszám maximum 2-3 napra elegendő táplálékot jelent. Ráadásul, ha nedvesebb ételt szeretnénk benne tárolni, akkor érdemes olyat vásárolni, amelyik hűtő akkukkal van

felszerelve, különben az eledlek könnyen megromolhatnak. Ezek a hűtő akkuk is legfeljebb 1-2 napig tudják megfelelően hidegen tartani a táplálékot, a környezet hőmérsékletétől függően, ezáltal igencsak le vannak korlátozva az alkalmazható időtartamot illetően. Ha valakit érdekel egy ilyen etető, az értékelések alapján a „Cat Mate C500” egy jó választás lehet.

2.10 OKOS KISÁLLAT ETETŐ / (SMART PET FEEDER)

Ez az okos kisállat etető [14] nem csak eledelt, hanem a kisállat vizét is újra tölti. A mechanikát szervomotorok végzik. A víztank alatt található a szervomotor, mely egy kemény kifeszített dróttal zárja és nyitja a vízcsapot. A tápot szintén szervomotor adagolja a tálba, melynek egy mérleg adja a visszacsatolást, csakúgy, mint a vizesedénynél. Ha a tálak tele vannak, nem indít újra adagolást. Ahhoz, hogy érzékelje a rendszer, mennyi víz és táp maradt még a tartályokban, ultrahangos érzékelőket szerelt fel a készítő a tartályok tetejére, függőlegesen talajirányba, melyekkel viszonylag pontosan megmondhatja, hány százalék eledel és víz van még. Emellett a kisállat jelenlétét is ilyen módon detektálja, további kettő ultrahang szenzorral.



2-10. ábra: Okos Kisállat Etető

A megvalósításról megosztott kép a 2-10.-es ábrán látható. Az etetőhöz egy igen komoly applikációt is készített a szerző, melynek egy ESP8266 küldi az információkat Mosquitto és Node-RED segítségével. A kalkulációk logikáját teljes mértékben a Node-RED platformon készítette, így ezt dinamikusan lehet módosítani. Az applikáción belül megtekinthetők a maradék táp és vízszint, értesítéseket kaphatunk a rendszertől bizonyos eseményeknél, illetve beállíthatók az etetések időpontjai.

Előnyös, hogy az eledeltartályok szintjét ilyen pontosan meg tudja állapítani, ezért ezt a módszert valószínűleg felhasználom a saját rendszeremben is. A másik pozitívuma a rendszernek, hogy nem csak étellel, de folyadékkal is kiszolgálja az állatot. Ráadásul a víz kiengedésének a módja igen hatékony, ugyanakkor, ha a kisállat nem issza meg a teljes mennyiséget, nem fog teljes mértékben friss vizet kapni.

A maradék, több ideig állt vizet jobb lenne valahogyan kiönteni, mielőtt a friss vízzel feltöltjük a tálal. Az itatást is fontos kérdésnek tartom, ezért a saját rendszeremben is

implementálni szeretném valamilyen formában. Erre a 3-as fejezetben részletesebben kitérek.

2.11 PIACON MEGVÁSÁROLHATÓ TERMÉKEK

Mivel a megvalósítás során az egyik fontos szempontom a megfizethetőség, célszerű lenne a már működő, jelenleg is értékesített termékek funkcióit, árait szemügyre vennem, így lesz egy viszonyítási alapom. Ezek az etetők olyan további funkciókkal is rendelkezhetnek, melyek hasznosak lehetnének a saját projektben is.

Ebben a fejezetben az aktuális, piacon megvásárolható, az enyémhez hasonló felépítésű okos kisállat etető és itató kínálatról fogok egy rövid betekintést adni. Elsősorban az automatikus etetésen felüli, extra funkciókat emelem ki. Az árakat külföldi (amazon.co.uk, aliexpress.com) weboldalakról gyűjtöttem és szállítás nélkül értendőek, amik sokszor jelentős összegek. A magyar árakon kerekítettem, 2021.05.08.-ei valuta értékkel.

Több listát [15] [16] [17] is megtekintve, a legjobb értékeléseket ez a két okos etető kapta:

WOpet Sprite II - 140\$ (~ 41 300 Ft)

Jellemzői: WiFi, okostelefon applikáció, kamera, hangüzenet, 10 g lépésköz, maximum 15 etetés naponta, 7L-es tároló, tartalék akkumulátor

Richdog Automatic Pet Feeder, Food Dispenser for Cats and Dogs - 67\$ (~ 19600 Ft)

Jellemzői: Gombok és kijelző, hangüzenet, 4 etetés naponta, 6L-es tároló, tartalék akkumulátor

Valamivel gazdaságosabb opciók:

Le Bistro automata etető - 50\$ (~ 14600 Ft)

Jellemzői: Gombok és kijelző, 3 etetés, étkezések egyedi mennyiséggel, 2,25 kg eledel

PAKEWAY Cat Dog Self-Dispensing Pet Feeder - 21\$ (~ 6200 Ft)

Jellemzői: Gravitáció által szinten tartott eledel, kompakt kialakítás, egyéb funkció nincs

És végül néhány kisállat itató, melyeket a legjobbnak értékelték: [18] [19]

Drinkwell 360 Stainless Steel Pet Fountain - 70\$ (~ 23500 Ft)

Rozsdamentes acél, 3,8L víz, szivacs szűrő, állítható vízfolyam

Catit Flower Cat Water Fountain - 26\$ (~ 7600 Ft)

Műanyag, 3L víz, szén szűrő, halk működés

2.12 KONKLÚZIÓ

Ebben a fejezetben, a mostanáig felhalmozott ismeretek alapján, következtetéseket vonok le a megismert projektekről és termékekről. A fentebb vizsgált rendszerek mindegyike automatikus etetést biztosít valamilyen módon. Ez az elsődleges feladatuk, tehát ezt alapvető funkciónak tekintem. Az etetés során a táp mozgatásához, adagolásához különböző megoldásokat alkalmaznak. Ezek egytől-egyig működő koncepciók, de mindnek megvannak a sajátosságai, különféle előnyökkel és hátrányokkal.

A rendszerelemzések lezárásakor vizsgáltam a projektek különféle technikáit, megoldásait. A legfőbb tényező, számomra az etető mechanika volt, de szintén szempont volt, többek között a bemeneti interfész, és az új ötletek felhasználhatósága a saját projektemben.

Számtalan eledel adagoló módszert sikerült megismerni, így magabiztosan tudtam dönteni. Az etető mechanikát mindenképpen szervomotoros és fűrócsigás technikával szeretném megvalósítani, mivel úgy gondolom, hogy a megbízhatóságot, a hatékonyságot és a pontosságot szem előtt tartva, ez tűnik az optimális választásnak.

A 2.9-es fejezetben megismert megoldás jól bemutatta a kamerában rejlő lehetőségeket, így természetes, hogy a saját okos rendszeremnek is a része lesz egy kamera, elsősorban az élő streamelés érdekében.

A megvizsgált projektek egy részében, a bementi perifériát, csak a felületi interfész adta, azaz egy billentyűzet és egy apróbb kijelző, amiken a felhasználók beállíthatták a rendszer különféle paramétereit. Némelyik projektben már megjelentek az applikációk, amelyek WiFi-n keresztül kommunikáltak. Itt többek között egyéb technológiákat is alkalmaztak a kommunikáció folyamán. Ezek nincsenek feltétlenül lekorlátozva a lokális hálózatra, tehát az internetre is bekonfigurálhatóak. Az én projekttem egyik fő célja, hogy nyaraláskor, munka miatt és egyéb, az otthonról való távolmaradásoknál is gondoskodni tudjunk kisállatunkról. Emiatt olyan rendszert kell biztosítanom, amely távolról, az interneten, egy applikációból bárhonnan elérhető egy átlag felhasználó számára is.

A lokális interfészt azonban hasznosnak tartom, mert így nem vagyunk az applikációra korlátozva, illetve annak első beüzemelésekor is jól jöhet. Másrészt, ha probléma adódna a WiFi kapcsolattal, vagy az applikációval, ezen a felületen az alap funkciókat továbbra is üzemeltethetjük, a hibák okaihoz pedig segítséget nyújthat.

Az etetők mellett egy kisállat itatóval is találkoztunk. Ez egy remek ötlet, mindenképpen szeretném implementálni a rendszeremben valamilyen módon. Nagyon fontos, hogy az állatok megfelelő gyakorisággal kapjanak friss vizet.

Egyik célom, hogy a lehető legtöbb funkciót sikerüljön megvalósítani saját rendszeremben is, kedvezőbb áron.

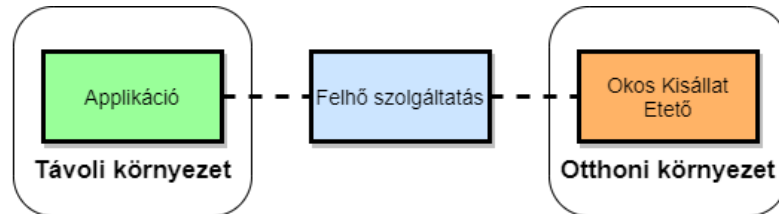
A 2.4-es fejezetben megismert automata macska vécé szintén jó ötlet, viszont számomra nem releváns, mert a macskánk már rászokott a macska alom használatára. Illetve praktikusabb lehetne egy olyan rendszer, ami megkeveri az almot olykor-olykor, mikor az nincsen használatban. Ha valaki ezzel szeretne kísérletezni, annak figyelmébe ajánlom a „Litter-Robot 3 Connect”-et, amely egy hatékony, de igencsak drága öntisztító macska alom.

Az RFID technológia hasznos lehet, ha több állat is van a házban, és azokat elszeretnénk szeparálni az etetés során. Ez megoldható az RFID technológiával működő mikrochip cédulával és kapuval. Egy lehetőség, hogy annyi etetőt szerzünk be, ahány különböző étrendű állat van a háztartásban. Az értékelések alapján erre a Surepetcare gyártó, Microchip Pet Feeder terméke a legjobb.

3 RENDSZER FUNKCIÓK ÉS ARCHITEKTÚRA

3.1 A RENDSZERARCHITEKTÚRA LEÍRÁSA

Egy tömör áttekintés az architektúra koncepciójáról a 3-1. ábrán:



3-1. ábra: Rendszer Architektúra

Ahogy az a 3-1. ábrán látszik, az architektúrát két részre lehet bontani, távoli és otthoni környezet. A köztes kommunikációs rétegben egy felhő alapú szolgáltatással, az interneten keresztül kapcsolódik a két környezet.

3.1.1 OTTHONI KÖRNYEZET

Az otthoni környezet részét képezi maga az Okos Kisállat Etető, mely tartalmazza a vezérlést biztosító mikrokontrollert, az etető gépet, az itató gépet, a szenzorokat, illetve egy lokális felhasználói interfészt. Ezt a bemeneti interfészt a felhasználó otthonról kezelheti.

3.1.2 TÁVOLI KÖRNYEZET

A távoli környezetben a felhasználó egy applikáció segítségével küldi és fogadja az adatokat. Ez az applikáció az interneten keresztül csatlakozik a felhő szolgáltatáshoz.

3.1.3 KOMMUNIKÁCIÓ

A két környezet közti kommunikációért egy felhő szolgáltatás felel majd. Ezt az interneten a router és az applikáció tudja csak elérni.

3.2 A RENDSZER FUNKCIÓI

A rendszer számos feladatot fog ellátni. Ebben a fejezetben egy absztrakt összefoglalót szeretnék nyújtani, a rendszerben megvalósítandó funkciókról. Ezekről csak magasszinten, a magvalósítást illető technikai részletek nélkül fogok beszélni.

3.2.1 AUTOMATIKUS ETETÉS

Az első és legfontosabb funkciót az etető rendszer végzi. A feladata, hogy a tápot egy nyíláson keresztül a tálba engedje, amihez az állat hozzáfér és biztonságban

elfogyaszthatja az eledelét. A felhasználandó eledel bármilyen száraztáp lehet. Az emberek nagy része, köztük állatorvosok is, a száraz, ropogós tápot ajánlják. [20]

Fontos, hogy a rendszer ügyeljen a precíz adagmennyiségekre. Egy alultáplált állatnál tápanyaghiány és súlyvesztés léphet fel, ugyanakkor a túletetés egy sokkal gyakoribb probléma. Kutatásokból kiderült, hogy a macskák, körülbelül 60 százaléka túlsúlyos vagy elhízott az Egyesült Államokban. A felesleges testzsír megnöveli a kockázatát több betegségnek, mint például a cukorbetegség, magas vérnyomás, mozgásszervi problémák, bőrbetegség, szívproblémák, egyes ráktípusok vagy akár a potenciálisan végzetes májbetegség, a máj lipidózis. [21] Emiatt az etető rendszerem adagolását, a lehetőségekhez képest, maximálisan pontosra szeretném megtervezni.

Az adagok mellett egy kulcsfontosságú szempont még a rendszeres időpontok betartása. [22] Az időpontokra a vezérlés, vagyis a mikrokontroller feladata ügyelni. Ehhez egy RTC, azaz Real Time Clock modul lesz segítségére. Kétféle módon aktiválódhat az etetés: automatikusan és kézzel. Az automata módban előre definiált mennyiségek és időpontok alapján adagol a rendszer. Ezek a paraméterek az adatbázisból kerülnek szinkronizálásra a készüléken. Ettől függetlenül, ha ad hoc módon egy véletlenszerű időpontban szeretnénk azon nyomban megetetni állatunkat, akkor manuálisan is elindíthatjuk az etetési folyamatot. Ez egyszeri alkalmat jelent, és a táp mennyiségét is meg kell adnunk hozzá. Ha a kisállat nem fogyasztotta el az összes eledelt, amit a gép kiszolgált, a gazdinak 3 opciója van. Az első, hogy a gép a következő etetéskor nem teljes adagot szervíroz, hanem kiegészíti az előzőleg megmaradt eledelt. A második, hogy a teljes adagot kiszolgálja ugyan úgy. A harmadik pedig, hogy addig nem indulnak el további etetések, amíg a kisállat a megmaradt tápot meg nem ette. Ezeket a beállításokat a gazdi az applikációban tudja majd bekonfigurálni.

Az etetési időpontok és mennyiségek bekalibrálásához elsősorban az okostelefonos applikáció használatos, de igénybe vehetjük a készüléken található, lokális felhasználói interfészt is. Az eledeltartály szintjének a monitorozására is lehetőség lesz az ultrahangos érzékelők segítségével.

3.2.2 ITATÓ RENDSZER

A folyadékpótlás elengedhetetlen a kisállatok életében, ugyanúgy, mint az emberekénél. [23] Sőt a kutyák és macskák testének vízszázaléka, meghaladja az emberét. Az emberek testének körülbelül 60, a kutyáknak 80, macskáknak akár 70 százaléka víz [24] [25] [26]. Ehhez képest a macskák gyakran nem isznak eleget, de ez a kutyáknál is előfordul. Emiatt a gazdiknak ajánlott valamilyen módszerrel csábítóbbá tenni az ivást kiskedvencük számára. Erre a célra az egyik legjobb módszer a víz frissen tartása. [25]

A friss víz kifejezetten fontos a macskáknak, ezért elengedhetetlen, hogy gyakran cseréljük a vizüket. Egy másik módszer, amittől több vizet ihat a kisállat, ha mozgó,

csöpögő vízfolyásként tálaljuk, ugyanis felizgatják őket a mozgó dolgok. Ezt a kérdést a kutyák és macskák a mindennapi életben slaggal, csappal, szökőkúttal [27] szokták megoldani. [25] Az én kisállat itatóm úgy lesz kialakítva, hogy a víz szintén folyjon/csöpögjön, ezzel is izgalmasabbá téve és ösztönözve a kisállatot az ivásra.

A víz adagolását vízpumpával szeretném megoldani, egy tárolóból, ami le van fedve, így ott frissen tud maradni a víz. A kiadagolt víz mennyiségét szintén a felhasználó állíthatja majd be, így ez a megoldás jól tud alkalmazkodni kisebb, nagyobb állatokhoz, miközben friss vizet biztosít a nap folyamán. Azt mondják [28] a hozzáértők, hogy legalább naponta egyszer cserélni kell a kisállatok vizét, hogy a tálban ne telepedjenek le bacilusok, így a kisállat egészségesen használhatja, akár több napon át. A víz cseréjét egy második vízpumpával oldanám meg, ami leereszti a tálban levő vizet, egy nagyobb víztárolóba vagy a lefolyóba, ha van rá lehetőség. Ezt naponta több alkalommal is elfogja tudni végezni a gép.

A folyó/mozgó vizet vagy szökőkutat nem minden macska preferálja [29]. Természetesen a lefolyó víz nyilván egy tálba fog érkezni, ahol nyugodtan ihatnak az állatok.

A tálról a felhasználó gazdiknak kell gondoskodnia, hogy az meg tudjon felelni a beállított mennyiségekhez. Azonban, a tálát illetően is fontos odafigyelnünk. Macskáknál lényeges, hogy a tál ne legyen túl szűk, ugyanis, ha a bajuszuk hozzá ér a tál oldalához, az irritálja őket, emiatt kevésbé fognak szimpatizálni az etetővel, ami miatt megnő az esélye, hogy kevesebbet esznek/isznak, mint amennyit szeretnének. [30]

3.2.3 LOKÁLIS BEMENETI INTERFÉSZ

A felhasználónak lehetősége van egy lokális interfészt használnia az alapvetőbb etetési konfigurációk elvégzésére. Szintén itt lesznek láthatóak különböző státuszjelzések a rendszerrel kapcsolatban, többek között a pontos idő, a következő etetés stb.

3.2.4 FELHŐ ADATBÁZIS

Az adatokat egy felhő alapú szolgáltatással fogom továbbítani az applikáció és a mikrokontroller között. Ez a szolgáltatás az adatok tárolására, szinkronizálásra is használható lesz. Az etető által küldött naplóbejegyzéseket szintén itt fogom megőrizni, amiket az applikációból később is elérhetünk.

3.2.5 ELEDEL MÉRLEG

A mérleget az etető rendszer, a pontos adagolás érdekében használja fel, visszacsatolással, de ad hoc módon a gazdi is mérhet vele eledelet. Ahhoz, hogy a felhasználó igénybe tudja venni ezt a funkciót, egy dedikált gombot kell megnyomnia a lokális interfészen.

3.2.6 FELÜGYELET KAMERÁVAL

A kamera segítségével az applikáción nyomon követhetjük kisállatunk étkezéseit. Ezzel biztosak lehetünk benne, hogy jó állapotban van, és hogy az evése, ivása nem maradt el. Többek között, magát a rendszer működését is ellenőrizhetjük.

3.2.7 HŐ ÉS PÁRATARTALOM MÉRÉS

Jól jöhet, ha meg tudjuk nézni kisállatunk környezetének hőmérsékletét, főleg, ha nincs a házban egy távolról is elérhető termosztát. Így megbizonyosodhatunk róla, hogy az állat megfelelő [31] hőmérsékletben vár minket otthon. Mivel nálunk nincsen termosztát, így előre beállítjuk a konvektorokat, hogy az a macskának kellemes legyen. Nem mindig van lehetőség megvárni, hogy ez pontosan hány fokra viszi fel vagy le a hőmérsékletet, ezért hasznos lenne, ha az applikáción keresztül, erről is megbizonyosodhatnánk.

Hasonlóan, mint az embereknél, a páratartalom az állatokra is hatással van. [32] Emiatt kihasználhatjuk, hogy a hőmérő modulok többnyire páratartalom mérővel párban is kaphatóak. A magas páratartalom csakugyan veszélyes az állatokra, [33] ezért érdemes ezt is monitorozni, hogy biztosak legyünk az állapotukban.

3.2.8 MOBIL APPLIKÁCIÓ

Ahhoz, hogy a felhasználó távolról is felügyelhesse az Okos Etető állapotát, kameraképet, egy applikációt fogok fejleszteni. Így a lokális interfész mellett a mobil applikáción is lehetőség lesz az adatok bevitelére. Ez sokkal felhasználóbarátabb felület. Fő funkciója az ételadagok és időpontok megadása, illetve az etetőtől kapott értesítések megjelenítése. Értesítést kaphatunk, például, hogy mennyi ételt sikerült adagolni, mennyi eledel maradt még a tárolóban, mikor lesz a következő etetés, mennyit evett a kisállat az utolsó etetéskor, illetve, ha kevés a táp vagy a víz a tartályokban. A kamerát szintén innen tudjuk majd elérni. A felhasználónak lehetősége lesz rendszeresen megadni a kisállat különböző paramétereit, ezzel naplózva azokat. Ezek az adatok feltöltődnek az adatbázisba, így az applikációban később is visszatekinthetők lesznek és leolvasható lesz az esetleges differencia.

Az etető gépnek nem csak a konfigurálását lehet majd elvégezni az applikációból. Képes lesz annak távoli irányítására, monitorozására. Indíthat azonnali etetést, megtekintheti a kamera képet, illetve, többek között az előző és a következő etetés időpontját. Az Okos Etető funkciónak indítására, szenzorok monitorozására is lehetőség lesz.

Kisállat tartási és etetési tippek adása több forrásból, kigyűjtve. Ezeket külön csoportosítva a kutyás gazdiknak és a macskás gazdiknak egyaránt.

Rendszeresen elmenthetjük a kisállat súlyát, ami tárolódik a naptárban, és később vissza lehet nézni, hogy mennyit evett és mennyit fogyott adott esetben.

Az egészséges táplálkozás érdekében okos táplálékkalkulátort is készítek, ami segít a gazdának megbecsülni az egészséges táplálék és folyadék mennyiségeket. Az adagok kalkulációjához a felhasználónak meg kell adnia kisállata paramétereit (súly, kor, nem), ezeket fogja figyelembe venni, és több forrásból összegyűjtött táblázat alapján behatárolni. Ez jól jöhet, ha az állat speciális diétán, például fogyókúrán van. Természetesen ezek csak ajánlásként lesznek a felületen.

3.2.9 RENDELKEZÉSRE ÁLLÁS

A rendelkezésre állás, vagyis, hogy a kisállat etető gép minden beállított etetés időpontjában üzemeljen, egy nagyon fontos kérdés a biztonság szempontjából. A rendelkezésre állás, általunk befolyásolható mértékének maximalizálása a cél. Ezt a szoftver hibamentesítésével és biztonsági mechanizmusokkal tudjuk elérni, mint például az időpontok biztonsági mentése, illetve az eledel és víz adagolásának, kiengedésének a biztosítása.

Itt még két külső akadály jöhet szóba. Az egyik az áramkimaradás. Erre az esetre egy akkumulátort lenne érdemes a rendszerbe implementálni, amely ki tudja szolgálni a legalapvetőbb funkciókat, ami az etetés és itatás. Az itatásnál a víz cseréje itt másodlagos prioritásúvá válik.

A másik akadály az internet kimaradás. Ugyebár ez áramkimaradás esetén is fennáll. Viszont, ha az internet kapcsolat megszakad, de a hálózati áram üzemel továbbra is, fontos, hogy a rendszer tovább tudjon működni. Emiatt életbevágó, hogy a rendszernek ne kelljen a hálózattól és a szerveren tárolt adatoktól függenie, az etetéshez. Ennek érdekében bizonyos időközönként biztonsági mentést hajt végre a szerveren tárolt adatokról (időpontok, mennyiségek) így, ha az interneten nem tudja elérni az erőforrásokat, ezeket is felhasználhatja a flash memóriából.

Ha nincsen hálózat, a felhasználónak akkor is lehetősége lesz új értékeket, időpontokat konfigurálni vagy etetést/itatást indítani a lokális interfész segítségével.

4 ISMERT SEBEZHETŐSÉGEK

Az IoT eszközök száma rohamosan nő, 2021 végére majdnem 14 milliárd eszköz fog üzemelni szerte a világon. [34]

Ez a megszámlálhatatlanul sok okos készülék széleskörű kényelmet kínál megfizethető áron, viszont másrésről új biztonsági sebezhetőségeket vonnak maguk után, melyek gyakran sértik, többek között a felhasználók magánéletét. [35]

Sajnos rengeteg IoT eszköz rendelkezik biztonsági résekkel, melyeket kihasználva, hackerek könnyedén kárt tehetnek az eszközeinkben. [36]

2012-ben az Internet Census támadásban/felmérésben egy hacker átvizsgálta az egész IPv4 címtartományt [37], 420 ezer routert felhasználva, melyeken még a gyári felhasználónév és jelszó páros volt beállítva. Ezeken keresztül, körülbelül 450 millió elérhető eszközt számolt, melyekhez hozzáfért és parancsokat tudott futtatni rajtuk. Ez megmutatta mennyire nem biztonságos az internet infrastruktúrája.

Ezekből a tanulmányokból is látszik, hogy mennyire fontos a biztonságra külön figyelmet fordítani és lépéseket tenni annak eléréséhez. Az saját IoT eszközömet 3 szemszögből vizsgálom. Fizikai, azaz lokális jellegű, a hálózati kommunikáció szempontjából, illetve a felhasználónál levő, telefonos applikációt illetően.

4.1 FIZIKAI (LOKÁLIS) JELLEGŰ

A forrás kód nem fejthető vissza, viszont a Flash memóriát kiolvashatják, ezért ezt valamilyen módon titkosítanunk kell. Lehetőség van az egész Flash Memória titkosítására, de ezt a mi esetünkben túlzásnak érzem, ugyanis csak pár darab adatot fogunk rajta tárolni folyamatosan. Amit realisabbnak látok, hogy a beleírt adatok értékét titkosítom, így, ha hozzá is fér valaki, nem fogja tudni értelmezni az adatokat.

A helyszínen hozzáférhető-e illetékteleneknek, például, ha egy idegen bejön a házba? Ha szétszedi az etetőt, hozzáférhet a mikrokontrollerhez, és átprogramozhatja az eredeti működését. Ez egy szélsőséges szituációnak tűnik, mivel az etető a házban található, és hogy valaki hozzáférjen a mikrokontrollerhez, először be kell jutnia illetéktelenül a házba, majd szét kell szednie az etető bizonyos részeit. Innentől egyéb problémák is felmerülnek, ezért erre a szituációra jelen feladatom szempontjából, nem fektetek nagyobb hangsúlyt.

Mivel egy IoT eszközről beszélünk értelemszerű, hogy a kommunikációhoz szüksége van valamilyen hálózatra/interfészre, amivel csatlakozni tud az internethez. Jelen esetben ez az otthoni router által szolgáltatott WiFi hálózat lesz. Ahhoz, hogy a routerhez csatlakozni tudjon, a felhasználónak meg kell adnia a WiFi használatához szükséges SSID-t és

jelszót. Ezeket a hitelesítő adatokat az első indításkor, Bluetooth segítségével, az applikációból tudja majd megadni a felhasználó.

Ezt követően a rendszer csatlakozik a WiFi-hez, és amennyiben a folyamat sikeresen lezajlott, az SSID és a jelszó titkosítva lesz, majd titkosítva mentésre kerül a flash memóriában. Emiatt érdemes olyan mikrokontrollert választani, mely rendelkezik Bluetooth technológiával, ellenkező esetben egy kiegészítő Bluetooth modullal kell megoldani. Az okostelefonokban pedig manapság mindig van Bluetooth-t.

Egy lehetséges harmadik opció még, hogy a mikrokontroller WiFi hoszt funkciójával (ha van annak ilyen) készíthetünk egy weboldalt, a mobilunkkal rácsatlakozhatunk a WiFijére, megnyitjuk az adott címet és beírjuk a rendes router WiFijének a hitelesítő adatait, majd konfirmáljuk. Ha nem találja meg a hálózatot, vagy helytelen a jelszó, akkor egy perc múlva újra elkezd a saját WiFijét hosztolni. Ha megtalálja a WiFit, akkor elmenti az SSID-t és a jelszót a flash memóriába titkosítva. Innen fogja tudni, hogy már nem kell a saját WiFijét hosztolni a következő indításnál, ha kikapcsolna vagy elmenne az áram.

4.2 HÁLÓZATI (TÁVOLI) KOMMUNIKÁCIÓN KERESZTÜL

Az okos otthonok esetében a biztonság az adatok és magánélet védelmét, elrejtését jelenti. Általánosságban két fajta fenyegetésnek van kitéve egy okos otthon, belső és külső módon. A belső fenyegetések a lokális hálózaton, míg a külsők az interneten való behatolást, adatlopást jelentik. Általános módszer az adatok védelmére a szimmetrikus és aszimmetrikus kulccsal való titkosítás, illetve a biometrikus azonosítás. [38]

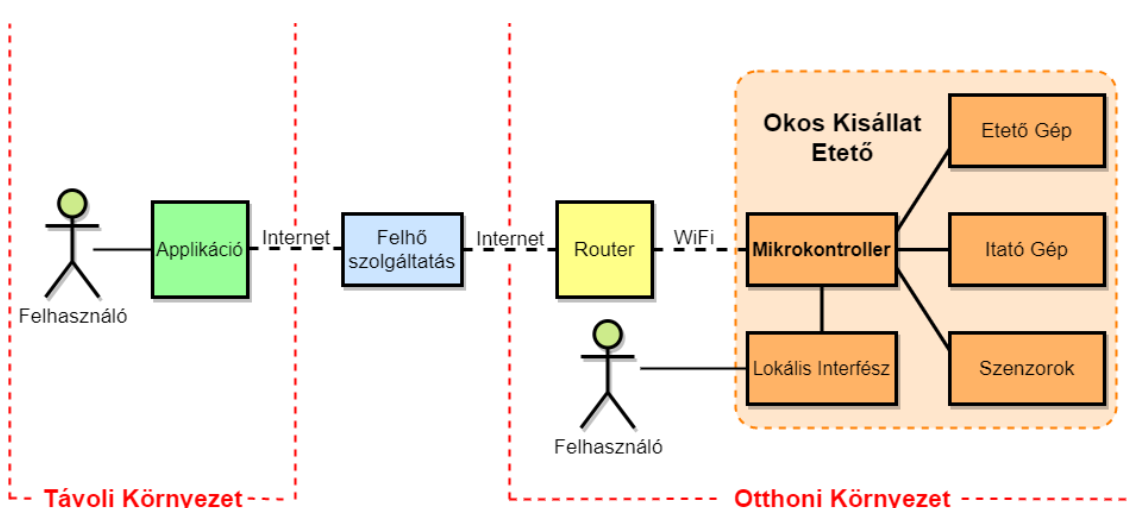
Mivel nem csak a helyi hálózatra (LAN) csatlakozik az eszköz, hanem az internetre is (WLAN), így fontos, hogy illetéktelenek ne tudjanak hozzáférni a rendszer kezelőfelületéhez. A másik dolog, amit meg fogok osztani az interneten az a kamera streamje. Ezt egy webszerverként fogom hosztolni. Ahhoz, hogy a web tartalmát jogtalanul ne tudják elérni, egy felhasználónév és jelszó kombinációs autentikációval védem meg. Illetve, a megvalósítás során alkalmazott hardverek lehetőségeit kihasználva fogom biztonságosabbá tenni a hálózati kommunikációt.

4.3 APPLIKÁCIÓ BIZTONSÁGA

Ha a felhasználó telefonja illetéktelenek kezébe kerül, fontos, hogy ne férjenek hozzá az applikáció funkcióihoz. Ennek érdekében az alkalmazásban a felhasználónak lehetősége lesz beállítani, hogy megnyitásakor egy felhasználónév és jelszó párost kérjen az app minden induláskor. Ezt érdemes lenne kiegészíteni biometrikus azonosítással is a kényelem érdekében.

5 RENDSZERTERV

Ahogy az már a 3. fejezetben szemléltettem, a rendszeremet két nagyobb részre lehet felbontani, otthoni és a távoli környezetre. Maga az okosetető az otthoni környezetben van jelen, de azt kontrollálni, monitorozni távolról is tudjuk majd az applikáción keresztül. A rendszerterv az alábbi, 5-1. ábrán látható.



5-1. ábra: Rendszerterv

Az Okos Kisállat Etető rendszer részét képezi egy etető gép, mely az eledelet adagolja megadott mennyiségben és egy itató gép, amely leereszti a régi vizet, és frisset önt helyette. Az időzítésre a vezérlés ügyel egy RTC-vel, azaz Valós Idejű Órával. A vezérlést egy mikrokontroller végzi, amely az okosetetőben foglal helyet. Ez amellet, hogy irányítja az etető és itató mechanikát, számos szenzorból gyűjti az adatokat, illetve kommunikál az applikációval és a lokális interfésszel. A felhasználónak lehetősége van ezt az interfészt bemeneti platformnak használnia az applikáció mellett. Ez korlátozottabb lehetőségekkel bír majd, mintha az applikációból csatlakozna a felhasználó, ugyanis az appot terveztem elsődleges bemenetnek, de az alapvető konfigurációkat végrehajthatjuk rajta. Ezeket az etetési konfigurációkat tetszés szerinti gyakorisággal végezhetjük el.

A szenzorok közül az egyik legfontosabb a mérleg, ugyanis ennek a visszajelzésével tud majd az etető gép precízen ételt adagolni. Egy másik jelentős szenzor még a kamera, amellyel élő videó stream-et tudunk megosztani és azt az applikációból nézni. Emellet kellene fog két ultrahangos távolságmérő az étel és víztároló készletfelméréséhez és végül egy hőmérséklet és páratartalom mérő. A mikrokontroller WiFin keresztül, egy router segítségével csatlakozik az internetre, ahonnan elérik egymást a Google Firebase felhő alapú szolgáltatással. Ez a szolgáltatás fog az applikációval is kommunikálni, ezzel megteremtve a csatornát a távoli környezethez. A következő fejezetekben megvizsgálom és összehasonlítom ezeknek a komponensek a megvalósításnál alkalmazható, különböző választási lehetőségeit, illetve kitérek a technikai részleteikre.

5.1 KAMERA

Mivel a kamera modell kiválasztása szintén befolyásolhatja a mikrokontroller típusának a kiválasztását, hatékonyabbnak tartom, először a különböző kamera alternatívákat megvizsgálni.

A kamera szerves funkcióját képezi a rendszernek, ezért fontos, hogy egy jó képfelbontású és lehetőleg magas képkockaszám/másodpercet biztosító modellt válasszunk. E két paraméter szorosan függ majd egymástól, így meg kell találni az arany középutat. Erre a kérdésre viszont ráér a megvalósítás során visszatérni, ugyanis a legtöbb kamerát szoftveres módon konfigurálhatjuk, és a képkockaszámot majd a felbontás nagyságának függvényében kapjuk meg. A modern telefonok magas felbontása miatt szintén érdemes törekedni a nagyobb felbontásra. Természetesen a költség is fontos tényező, ezért az eszközök árát is szemügyre véve, az OV2640 [39] és OV7670 [40] kamera modulok tűnnek optimális választásnak, ezek körülbelül 2-2,5\$ körül mozognak. Mindkét szenzorra igaz, hogy alacsony feszültségen üzemelő CMOS képérzékelő, ami teljeskörű funkcionalitással kínál, egy-chipes kamerát és képfeldolgozót egy kisterületű csomagban, illetve magas érzékenységgel operál gyenge fényviszonyok közben is. A két kamera paramétereit az 5.1-a. táblázatban tüntettem fel az összehasonlítás érdekében.

5-1. táblázat: Kamerák Összehasonlítása

Kamera	OV2640	OV7670
Maximum Felbontás	UXGA 1600x1200	VGA 640x480
Tápfeszültség	1.7V - 3.3V	1.7V - 3.0V
Energia szükséglet	600 μ A	<20 μ A
Üzemi Hőmérséklet (Fotó)	0°C - 50°C	0°C - 50°C
Képérzékelő Méret	1/4"	1/6"
Képkocka/másodperc (fps)	UXGA/SXGA:15 SVGA:30 CIF:60	VGA:30

A legfőbb különbség a kamerák felbontása, ugyanis az OV2640-nek ez 1600x1200, míg az OV7670-nek csak 640x480, így az OV2640-el sokkal szebb képet tudunk majd átadni, ami fontos a feladat szempontjából. Ezen a felbontáson, maximum 15 képkockát tud készíteni másodpercenként, ami megfelelő értéknek mondható, de SVGA felbontáson, (ami a VGA felbontásnak több, mint a duplája, 1024x768) a 30fps-t is elérheti. Amiben az OV7670 előnyösebb, az az elképesztően kicsi energiaszükséglete, viszont mivel a kamerák nem üzemelnek állandóan, hanem csak alkalomadtán egy bizonyos ideig, ez az érték elhanyagolható. Az én feladatomhoz az OV2640-et alkalmasabbnak látom, ezért azt fogom implementálni a rendszerbe.

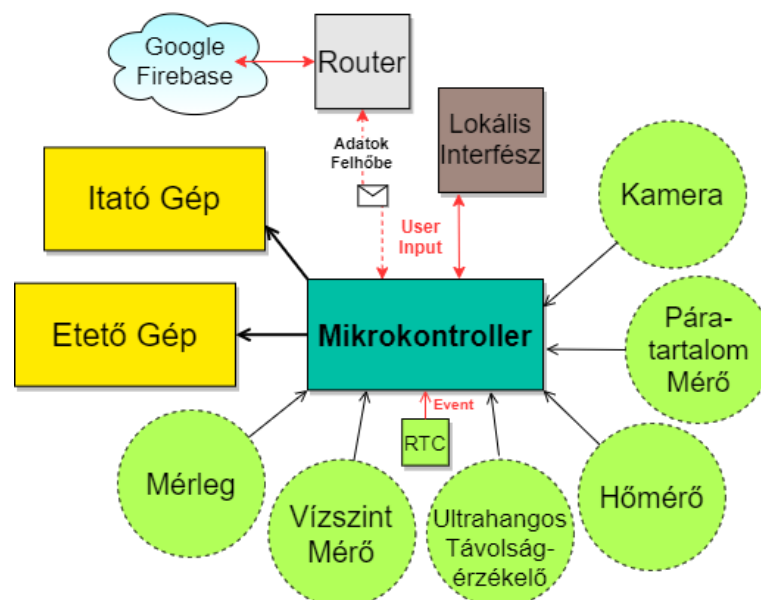
A Raspberry Pi kamera modulok között találhatunk 5, 8, de akár 12.3 megapixel-es felbontásokat, ami szemrevaló különbség az OV2640 2 megapixeléhez képest,

ugyanakkor ezek kiinduló ára körülbelül 25\$, így közelről sem férnek bele a keretbe, csakúgy ahogyan a Raspberry Pi mikrokontroller sem.

5.2 VEZÉRLÉS, MIKROKONTROLLER

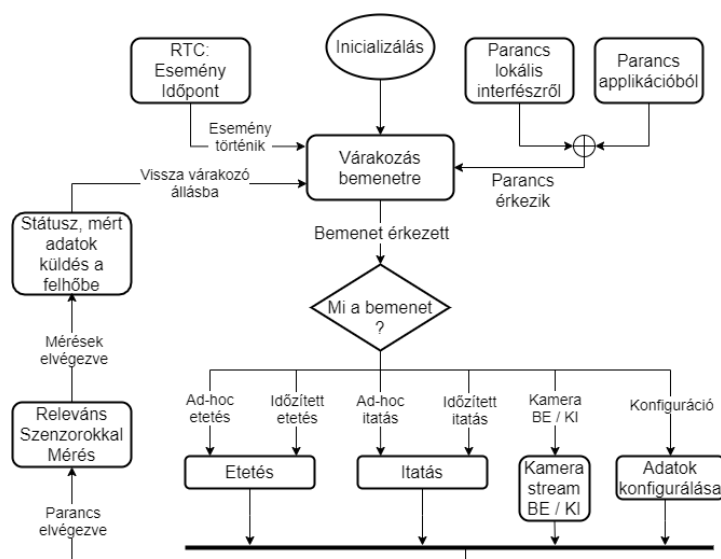
A rendszer alapvető vezérlője a mikrokontroller. Ez szabályozza az etetést, itatást, a méréseket és a kommunikációt is. A mikrokontroller egy lapkára integrált, vezérlési feladatokra optimalizált mikroszámítógép. Költséghatékonyan tud ellátni egyszerű, kis számítási teljesítményt igénylő műveleteket. Előnyük a csekély fogyasztás, apró méret és alacsony költség. [41] Ebben a fejezetben összegyűjtöm milyen feladatai lesznek a rendszer vezérlőjének, mely komponensekkel fog kapcsolódni és hogyan képzelem el a vezérlés folyamatát. Ezt követően megvizsgálom, mely mikrokontrollerek állnak a rendelkezésemre az adott keretek között. Összehasonlítom őket, majd választok egyet a feladatnak releváns szempontok szerint.

A mikrokontroller felel az Okos Etető összes komponensének vezérléséért. Az alap aktivitása, hogy vár amíg érkezik egy bemenet, majd elvégzi a kiszabott feladatot. Ez lehet User Input vagy esemény. User Inputnak számít, ha a felhasználó ad-hoc indít etetést vagy itatást, illetve, ha bekonfigurál egy új etetési/itatási időpontot a lokális interfészen vagy a felhőből. Eseménynek az időpontokhoz kötött etetés, itatás tartozik, amit az RTC aktivizál. A rendszer mikrokontrollerrel összekötött komponensei az 5-2. ábrán láthatóak. Az egyes elemek függetlenek egymástól, csak a mikrokontrollerhez csatlakoznak közvetlenül. Egyszerre csak egyetlen egy dolgot vezérel: etetést, itatást, hálózati kommunikációt vagy az egyik szenzort olvassa. Ezekre a következő fejezetekben térek ki. Egyetlen kivétel a mérleg, ugyanis azt etetés közben is használja, mint szabályozót.



5-2. ábra: Mikrokontroller Blokk Diagram

A fő vezérlést az 5-3. ábra alapján szeretném megvalósítani.



5-3. ábra: Vezérlés Folyamatábra

Ebben a részben végigmegyek a mikrokontroller kiválasztásának lépésein. A feladatom követelményeit szemmel tartva, egy pénztárcabarát konstrukcióhoz, optimális választás lenne az Arduino vagy az Espressif valamelyik mikrokontrollere. A lehetőségekhez képest ezek a panelek nagy teljesítményt nyújtanak kedvező árakon, és az én általam szándékozott feladatokra alkalmasak. Nagy fejlesztői közösséget tudhatnak maguk mögött, ezáltal rengeteg forrás könyvtár áll a rendelkezésünkre, ami nem hátrány. Az 5-2. táblázatban a különböző Arduino és ESP mikrokontrollerek jellemzőit hasonlítottam össze.

5-2. táblázat: Mikrokontrollerek Összehasonlítása

Név	Arduino Nano	Arduino Uno	Arduino Mega 2560	ESP8266	ESP32	ESP32-CAM
Processzor Típus	ATmega328	ATmega328	ATmega2560	Xtensa 32bit s-core L106	Xtensa 32bit s.d.-core LX6	Xtensa 32bit s.d.-core LX6
Processzor Frekvencia	16Mhz	16Mhz	16Mhz	80Mhz	160Mhz (240Mhz)	160Mhz (240Mhz)
Működési Feszültség	5V	5V	5V	3.3V	3.3V	3.3V
RAM	2Kb SRAM	2Kb SRAM	8Kb SRAM	512Kb, 4Mb Flash	520Kb SRAM	520Kb+4Mb PSRAM
I/O száma	22 Digital I/O pin	14 Digital	54 Digital	17 GPIO	34 GPIO	10 GPIO
Előnyös funkciók, jellemzők	USB csatl., 45x18mm	USB csatl., 68x53mm, 25g	USB csatl., I/O csatl. száma, külső tápellátás	USB csatl., WiFi	USB csatl., WiFi, Bluetooth, SD kártya foglalat, ULP processzor, titkosítás	

Az Arduino-k közül esősorban a Mega [42] jöhetne szóba, de viszonyítási alapnak a Nano [43] és Uno [44] változatokat is feltüntettem. Az ESP-k közül az ESP8266 [45], az ESP32 [46] és az ESP32-CAM [47] modelleket vettem fel.

Az Arduino panelek előnye, hogy számos I/O be és kimenettel rendelkeznek, így több modul esetén sem kell spórolnunk a hellyel, csakhogy számítási teljesítményük és memóriamennyiségük igen csekély az ESP alternatívákhoz képest. A feladat jellegéből adódóan szükség lesz internet hozzáférésre, hogy kommunikálni tudjunk a felhővel, és a kamera által felvett videót meg tudjuk osztani élőben az interneten. Emiatt a nagyobb sebesség és memória is indokolt lenne. Ráadásul az Arduino vezérlőkön nincs natív WiFi támogatás, így azok esetében úgynevezett bővítő paneleket szokás alkalmazni. Ebből levonva a következtetés, a mi feladatunk szempontjából az Arduino mikrokontrollerek nem ideálisak, így az ESP panelek közül kell választani egyet.

Az ESP8266-ban már beépített WiFi adóvevő áll a rendelkezésünkre, és a processzor órajele akár 80Mhz-en is tud üzemelni. Ez nagy előrelépés az Arduino alternatívákhoz képest. Ezt tovább erősítve, az ESP32 az ESP8266 továbbfejlesztett utódjának tekinthető. Az ESP32-ben a beépített WiFi kívül Bluetooth-t is kapunk, illetve egy kétmagos, akár 240Mhz-en operáló processzort. Különböző alvó módok állnak rendelkezésünkre mikor nem végzünk aktív feladatot, ezzel is minimalizálva az energiafelhasználást. Az extra funkciók miatt, drágábban kapható, mint nagytestvére, körülbelül 6\$ ameddig az ESP8266 csak 3\$.

Mivel kamerára is szükség lesz a projekt során, az ESP32 alapú ESP32-CAM panelt is érdemes szemügyre venni. A fő erénye az integrált kamera áramkör és a 4Mb PSRAM, melyet kihasználva, ideális választás internetet és kamerát igénylő projektekhez. A kamerához nagyon fényes beépített LEDet kapunk, amit vakuként használhatunk sötétebb fényviszonyoknál.

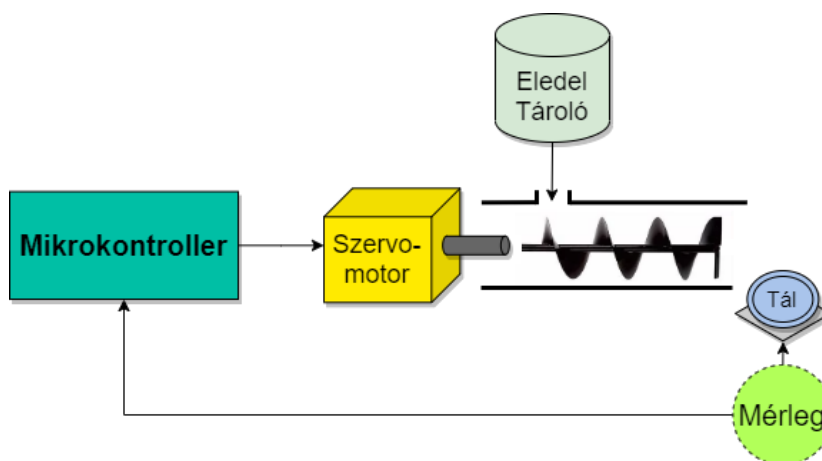
Ezeket a fejlesztői panelokat többnyire az OV2640-es kamerával találjuk meg, de más kamerákkal is kompatibilis, mint például az OV7670, illetve a kameralencsék betekintési szögéhez is több opciót kapunk. Hátránya az ESP32-CAM mikrokontrollernek, hogy kevesebb csatlakozót találunk rajta az integrált áramkörök miatt, ezért erre ügyelni kell a megvalósítás során.

Az ESP32 alapú integrált kamerás paneleknek több változata is van, mint például az ESP-Eye és több TTGO variáns. Mindegyik az OV2640-es kamerát használja, de a fejlesztői paneljeik eltérő sajátosságokkal rendelkeznek. Ilyenek például a PIR mozgás érzékelő szenzor, LCD kijelző, mikrofon, USB port. Ugyanakkor nem rendelkeznek kiegészítő I/O tűkkel, amire szükség van a többi komponens vezérléséhez, illetve az áruk is magasabb. [48]

5.3 ETETŐ GÉP

Az etető gép fő elektronikai részei a szervomotor és egy mérleg, amiket a mikrokontroller vezérel. Feladata az eledel egy beállított mennyiségének eljuttatása az ételtárolóból a kisállat táljába. A 3.2.1-es fejezetben taglaltak miatt, a megvalósítás során törekedni szeretnék a maximális pontosságra, ezért ehhez egy egyszerű szoftveres szabályozót készítek, amely a mérleg értékét dolgozza fel, és csatolja vissza a szervomotor hajtására.

A 2.3-as fejezetben megismert, szervomotor által meghajtott, csiga elvű fűrőszáras megoldást választottam etetési mechanikának. A pontosság és megbízhatóság szempontjából ez tűnik optimálisnak. A fűrőszár egy csővezetékben helyezkedik el, ami szorosan igazodik a fűrőszár keresztmetszetéhez, így az eltolt tápdarabok nem tudnak beszorulni. A fűrőt egy szervomotor hajtja, ami által az eledel kihullik a csatorna végén, bele az etető tálba. A csőnek a tetején egy nyílás található, erre a pontra csatlakozik az Eledel Tároló, ahonnan a gravitáció segítségével bejut a táp a rendszerbe. Ennek döntési szöge befolyásolja majd, mennyi eledelt mozgat egyszerre a fűrőszár. A megvalósítás során, ezt szükségszerű lesz finomhangolni, ezért kezdetben ezt egy 1,5-2L-es flakonnal fogom megoldani, aminek döntésszögét könnyen konfigurálhatom. A mérleg közvetlenül a tál alatt foglal helyet, így a tál súlyát is számításba kell venni. Az Etető gép blokk diagramja az 5-4. ábrán látható.



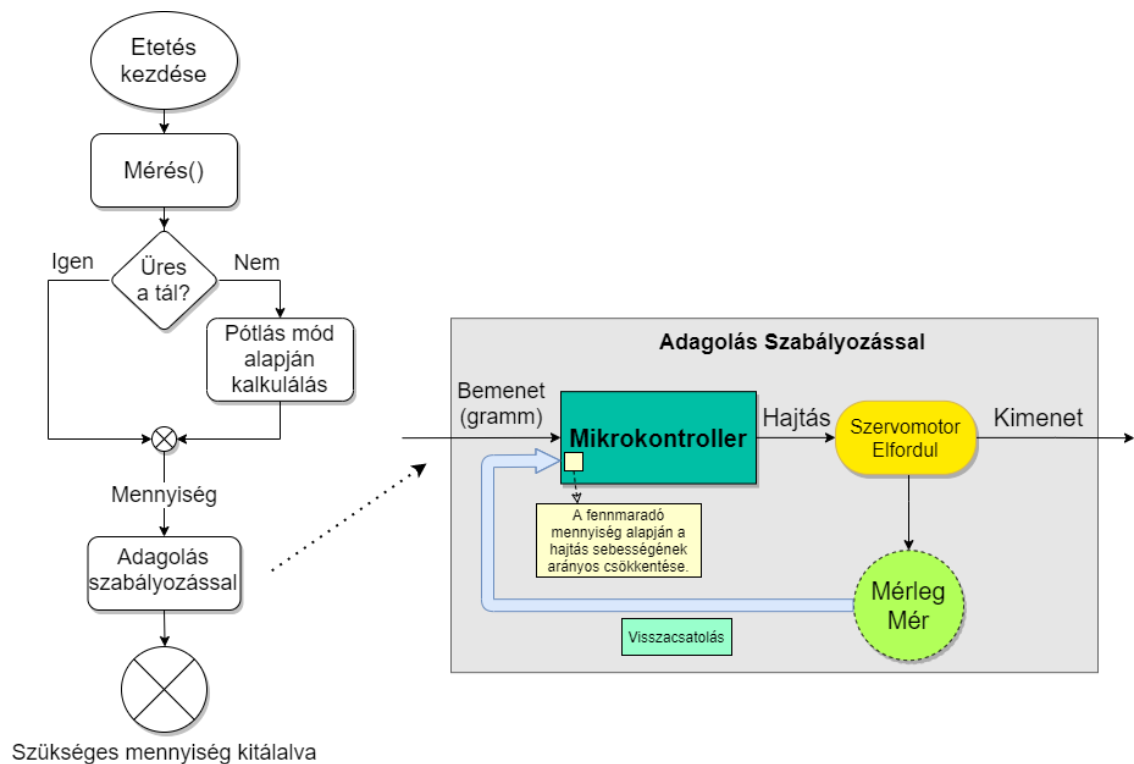
5-4. ábra: Etető Gép Blokk Diagram

A szervomotor helyett alkalmazhatunk DC motort vagy léptető motort is, viszont egy szervomotorral sokkal precízebben hajthatjuk meg a csigát, mint egy DC motorral, a beépített áttételek miatt, emellett pedig gyorsabbak, mint a léptető motorok. [49] Ugyanakkor a legtöbb szervomotor csak 0-180°-ban képes elfordulni, emiatt ügyelni kell rá, hogy folyamatos forgásút szerezzünk be, ami 360°-ban képest elforgatni a csigát. Emiatt a választásom egy MG995 fém fogaskerekes szervóra esett.

Kis testű állatoknál vagy speciális diétát, esetleg fogyókúrát követő állatkák esetén fontos, hogy a lehető legaprólékosabb, precíz adagolást tudjuk megvalósítani. Emiatt az eledel csatornát minél vékonyabbra fogom elkészíteni.

Mérleg modulnak egy HX711AD chipet alkalmazó egységet használok fel, aminek pontossága igen nagy, így kisebb adagokhoz is tökéletesen használható. A HX711 félvezető egy 24-bit pontos, súlyméréshez tervezett ADC, vagyis analóg-digitális átalakító. A kialakítás, amit felhasználok körülbelül 5 kg-ig operál, pontossága ± 1 gramm. Természetesen ehhez kalibrálni kell egy kezdeti méréssel, és a differenciát elmenteni. Ezt megtehetjük az EEPROM-ban, így a mérleget nem kell minden indításkor kalibrálni. [50]

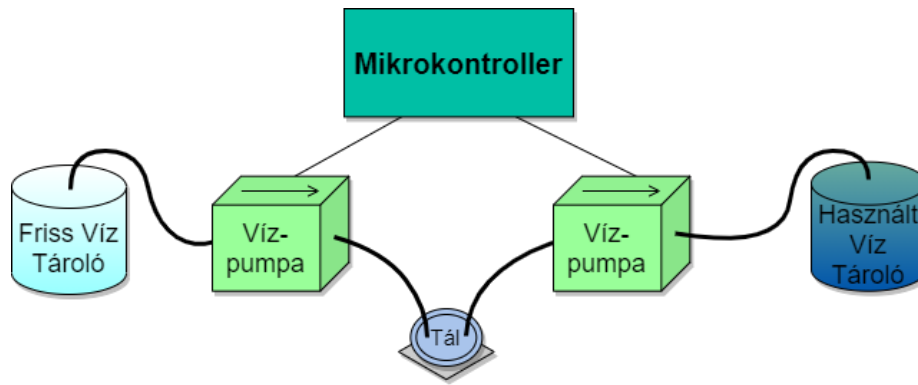
Az etetés folyamata az 5-5. ábrán látható. Ez a folyamat tulajdonképpen az 5-3. ábrán szereplő, Etetés aktivitást írja le. Az 5-5. ábrán az adagolás szabályozásának leegyszerűsített menetét is felvázoltam. Annak részleteire a megvalósításban térek ki.



5-5. ábra: Etető Gép Folyamatábra

5.4 ITATÓ GÉP

Az itató gépben, a 3.2.2-es fejezetben kifejtett okok miatt, vízcserélőre szeretném megépíteni. Ehhez két darab vízpumpát fogok felhasználni. Az egyik a Friss Víz Tárolóból pumpálja bele a tálba a vizet, a másik pedig a tálból ki egy másik tárolóba, esetleg lefolyóba. Így biztosak lehetünk benne, hogy nem marad használt víz a kisállat táljában, és mindig friss vizet ihat. Az itató gép blokk diagramja az 5-6. ábrán látható.



5-6. ábra: Ítató Gép Blokk Diagram

Egy alternatív opció lehetne még, hogy a Friss Víz Tárolóból nem pumpával, hanem egy szervomotorral kinyitott csapon keresztül eresszük ki a tálba a vizet. Ez 1-2 dollárral költségkímélőbb lenne a szervomotor és vízpumpa árkülönbsége miatt, azonban a csap által kiengedett víz mennyiségét nehezebb kiszámítani, illetve a csap fizikailag jobban elzárja a tálát a kisállattól. A csapok erre a célra viszonylag előnytelen helyéről nem is beszélve.

A pumpa csövét, a tál oldalán vezetve, az nemcsak a kisállat kényelmét biztosítja, de ösztönzi is az ivásra, a 3.2.2-ben tett megállapítások miatt. A víz folyásának sebességét a felhasználó fogja tudni szabályozni.

A vízpumpa, amit felhasználok az AD20P, 4.8W teljesítménnyel. Alternatívaként egy 6V-os pumpát fogok még tesztelni.

5.5 SZENZOROK

5.5.1 RTC

A mérleg és kamera mellett egyéb szenzorokat is használni fog a rendszer. Ilyen az RTC óra, ami nélkülözhetetlen a megbízható, pontos időzítések érdekében.

A DS3231 és AT24C32 modulok egy darab lapkán található RTC és EEPROM. Ezzel pontos időt tudunk mérni és az EEPROM-nak köszönhetően adatokat is naplózhatunk benne. A DS3231 alacsony feszültségen üzemelő, roppant pontos I2C valós idejű óra, integrált hőmérséklet-kompenzált kristály oszcillátorral. [51] [52]

5.6 ULTRAHANGOS TÁVOLSÁGÉRZÉKELŐ

Az Eledel Tároló szintjének mérésére egy ultrahangos távolságérzékelőt fogok használni. Ez a tároló tetejében lesz elhelyezve, vertikálisan a földszint felé irányítva. A felhasználónak be kell kalibrálnia az üres és teli állapotokat, így a rendszer tud majd számítani egy százalékos értéket.

Ahogy az Eledel Tárolónak úgy a Víz Tárolónak is mérni szeretném a tartalmát. Ehhez szintén alkalmazhatok egy ultrahangos távolságérzékelőt.

A modul, amit felhasználok a HC-SR04. Ez 2 - 400 cm-ig tud mérni, körülbelül 3-4 mm-es pontossággal, a hatásos szöge 15°. [53]

5.6.1 HŐ ÉS PÁRATARTALOMMÉRŐ

Egy hő és páratartalom mérő szenzort is adok a rendszerhez, azért, hogy a felhasználó biztos hagyja otthon kisállatát. A modul, amit felhasználok a DHT11 hő és páratartalom mérő. A hőmérője 0-50°C között, $\pm 2^\circ\text{C}$ pontosságú. A páratartalom mérője $\pm 5\%\text{RH}$ pontosságú. [54]

5.7 APPLIKÁCIÓ

Az applikáció segítségével távolról, az interneten keresztül is elérhetjük az Okos Kisállat Etetőt. Itt el kell dönteni, hogy Android vagy iOS készülékekre készítek-e az appot. 2021 áprilisában az okos telefon használók körülbelül 69%-a Androidos volt. [55] Annak érdekében, hogy minél szélesebb körben használható legyen, megvizsgálva a piaci eloszlásokat egyértelmű, hogy az Android operációs rendszer az elterjedtebb, így én is arra fogom az appot fejleszteni.

Felmerült a kérdés, hogy applikáció helyett, inkább weboldalt készítek erre a célra. A feladat szempontjából viszont előnyösebb az applikáció, mert könnyebben tudok értesítéseket küldeni rajta a felhasználónak, és gyorsabb, biztonságosabb, mint a webes felület.

Az applikáció funkciói közé tartozik az etetés/itálás indítása, időpontok, adagok konfigurálása (amire egy okos kalkulátor is segítségül lesz), kamera képének indítása és megjelenítése, kisállat tartási tippek adása, értesítések küldése a felhasználónak. Az adatokat a Google Firebase felhőszolgáltatás segítségével fogja közölni egymással a két rendszer. A kamera képét egy webszerver formájában fogom hosztolni, de úgy, hogy azt csak biztonságosan, az applikációból lehessen elérni.

Az okos táplálék kalkulátor és kisállat tippek funkciót már a 3.2.8-as fejezetben kifejtettem. Ezek csak az applikációból lesznek elérhetőek. A kalkulátornak szüksége lesz a kisállat adataira. A tippekhez és kalkulációhoz szükséges információkat több forrásból [21] [22] [56] [57] [58] [59] [60] [61] [62] [63] [64] [65] [66] fogom összegyűjteni, és szövegesen minden applikációindításkor megjeleníteni egyet az applikáció kezdőképernyőjén a gazdinak, de erre egy külön menüpont is található lesz.

Az appot megnyitva, az autentikációs folyamat után a kezdőképernyőt látja a felhasználó. Az 5-7.-es ábrán látható az app kinézetének vázlata. A színek, szövegek, méretek stb.

valószínűleg változnak majd. A 3 alsó navigációs gombbal válthatunk a menüablakok között. A Kezdőképernyőn statisztikai adatokat, tippeket láthatunk, mint egy dashboard-on. A Funkciók ablakban gombok találhatók, amik külön ablakokba navigálják a felhasználót, ahol a funkcióhoz szükséges bemeneti elemek jelennek meg. A Beállításokban konfigurálhatjuk az etetések, itatások időpontjait, mennyiségeit, esetleg az itatások időtartamát / napi számát.



5-7. ábra: Applikáció Wireframe

5.8 TÁPELLÁTÁS

Az alapértelmezett tápellátás a hálózati áramról fog történni. Egy 12V-os tápegységet fogok alkalmazni. Később ezt egy akkumulátorral is ki szeretném bővíteni a biztonság kedvéért. Erre a témára a megvalósítás során fogok nagyobb figyelmet fordítani.

5.9 FELHŐ, HÁLÓZATI KOMMUNIKÁCIÓ

Felhő szolgáltatásnak a Google Firebase-t választottam, mert ingyenes, egyszerű bekonfigurálni, támogatja az Androidot és webes platformot. Funkciókat tekintve, rendelkezik Real Time adatbázissal, amit jelen esetben használnék, autentikációval, szerver hosztolással.

5.10 LOKÁLIS INTERFÉSZ

A lokális interfész jól jöhet, ha a felhasználónak nem áll rendelkezésére az applikáció. Illetve a kezdeti konfigurációt, finomhangolásokat is innen végezhetjük el. A bemenet egy 4x4-es billentyűzetpanel. A kimenetért egy LCD1602 modul felelős, 16x2 karakterrel.

Az egyszerű menürendszerrel lehetőség lesz etetési/itatási időpontokat beállítani, mérleget nullázni, szenzorokat finomhangolni, ad-hoc etetést/itatást indítani. Az LCD képernyőn státuszjelzéseket fog megjeleníteni.

5.11 TERVEZETT KÖLTSÉGVETÉS

Ebben a fejezetben összegyűjtöm milyen költséggel számolhatok, mennyi pénzért tudok majd hozzájutni az egyes alkatrészekhez a megvalósítás során. A 2.11-es fejezetben megismert termékek árába nem számítottam postaköltséget, itt viszont azt is beleszámítom. Ehhez a megvalósítás során, természetesen még hozzáadódnak egyéb költségek, mint például kábelek, elektronikai alkatrészek stb.

5-3. táblázat: Tervezett Költségvetés

Alkatrész neve	Egységára
ESP32 CAM	5,5\$
Mérleg	8\$
Szervomotor	8\$
Ultrahangos távolságmérő (×2)	0,85\$
Hő és páratartalom mérő	2\$
Vízpumpa (×2)	4\$
RTC óra modul	4\$

6 MEGVALÓSÍTÁS

A következő alfejezetekben a rendszer főbb részeinek megépítéséről, implementációjáról írok. A sorrend a 3.1 -es fejezetben megismert rendszer architektúra mentén halad. Először a Távoli környezetben lévő applikációval foglalkozok, mely a felhasználó elsődleges ki és beviteli interfészét jelenti. Ezt követi a Felhőszolgáltatás konfigurálása, annak összekapcsolása az applikációval. Ezután az Okos Kisállat Etető megépítésével folytatom, mely magába foglalja az összes szenzort, elektronikus alkatrészt, amelyek az etetéshez, itatáshoz és egyéb környezeti értékek méréséhez szükségesek. Itt természetesen az Otthoni környezetben lévő routerrel való kommunikációt is biztosítani kell majd.

6.1 APPLIKÁCIÓ

Miután megrendeltem a rendszertervben felsorolt alkatrészeket, modulokat és a hozzájuk szükséges vezetékeket, az Android applikáció fejlesztéséhez fogtam hozzá. A cél a szenzor adatok monitorozása és utasítások küldése az interneten [67]. Az 5.7-es fejezetben már megtervezett felhasználói felület alapján az applikációt 3 főbb képernyővel készítettem el. Ezek a Kezdőképernyő, Funkciók és Beállítások címkével vannak ellátva.

6.1.1 FELEPÍTÉS

Az applikáció inicializálásakor egy autentikációs ablak fog megjelenni, ahol felhasználónév és jelszó párossal vagy újlenyomattal léphetünk be a rendszerbe. Ezután értelemszerűen a Kezdőképernyő nyílik meg. Itt láthatóak a különféle szenzorok által mért értékek, melyeket az Okos Kisállat Etető a felhőbe töltött fel, illetve néhány egyéb információt tudhatunk meg meg. A második képernyőn olyan funkciók találhatók, amiket a felhasználó kézzel aktiválhat. Ezek ad-hoc módon bármikor indíthatók. Az utolsó ablakban az etetések és az ivóvíz cserék időpontjait konfigurálhatjuk saját belátásunk szerint, ezeket ki- és bekapcsolhatjuk.

6.1.2 NAVIGÁCIÓ

A három főablak közötti váltást, az összes képernyő alatt megtalálható, fix navigációs sávban lévő gombokkal teheti meg a felhasználó. Itt egy kattintással bármelyik ablakra átugorhatunk. Az éppen aktív, megnyitott ablak gombját kiemelt színnel jelölöm, jelen esetben világos zölddel. A gombokra a képernyőneveken kívül szimbólumokat is elhelyeztem, amelyek a felhasználóbarát élményt javítják és intuitívabbá teszik az alkalmazás használatát. Ablakváltáshoz a jelenlegi, egyszeri kattintás mellett, lehetőség lesz az oldalirányú húzásra is, de ezt még nem implementáltam a jelenlegi verzióban. Az alkalmazás elhagyását a vissza gomb egyszeri lenyomásával tehetjük meg. Ezután, ha visszatérünk, a beírt adatok nem vesznek el. Ahhoz, hogy teljesen kiiktassuk a futó alkalmazást, a rendszer által szolgáltatott alkalmazás listában kell azt kilőni.

6.1.3 KEZDŐKÉPERNYŐ

Az applikáció megnyitásakor a felhasználót közvetlenül információkkal és a legfrissebb mért adatokkal szembesítjük. Ez az érték lista a következőkből áll. Az eddel tartályban jelenleg fennmaradó táp, arányos mennyisége százalékban, a víztartály hasonló jellemzője, az aktuális környezeti hőmérséklet, és a páratartalom. Ezek azok az értékek, amiket az Okos Kisállat Etető a legutóbbi ciklusában mért és feltöltött a szerverre. Ezen kívül az összes etetés és itatás időpontjai alapján kivannak számolva és feltüntetve az előző és következő etetések, itatások időpontjai is. Fontos, hogy ezek csak a bekonfigurált/naptári időpontokat veszik figyelembe, és az ad-hoc etetés, illetve vízcseré indításokat nem.



6-1. ábra Applikáció - Kezdőképernyő

A 6-1. ábrán szereplő képernyőfotón látható, hogy a színeknek hála igen figyelemfelkeltő az ablak megjelenése. A céljuk legalábbis ezt hivatott szolgálni. Az értékekkel kapcsolatban tovább szeretném fejleszteni az egyes háttér sávokat, úgy, hogy a százaléknak és egyéb értékeknek figyelembevételével, egyenesen arányosan, maga az érték mögötti sáv is akkora helyet foglaljon, olyan hosszú legyen, amekkora az adott érték.

Az időpontok alatt található Adatok Frissítése gombbal a felhőből lekérhetőek a friss adatok. Ezt ablakváltáskor is elvégzi az alkalmazás, de automatikusan még nem frissül. Ezért az automatikus frissítésért egy eseménykezelő lesz felelős, ami a felhőből kapott

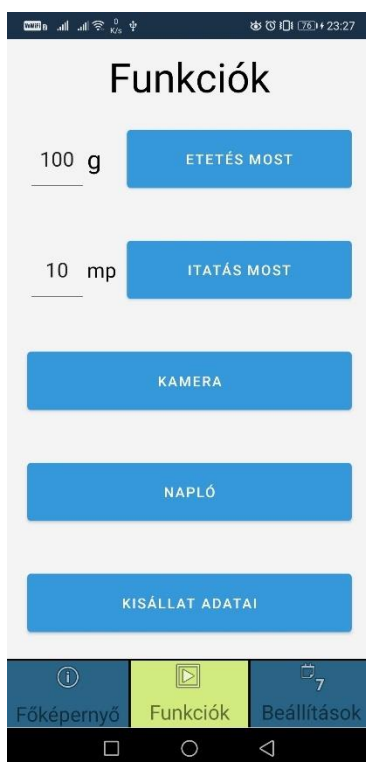
jelzésre lekéri az új, módosult adatokat, és befrissíti a képernyőn szereplő értékeket, még akkor is, ha a felhasználótól nem érkezik interakció.

Az utolsó elem, ami az ablakon található a tippek felület, amit a háziállatot tartó gazdiknak szánok. Itt minden alkalmazás-indításkor, véletlenszerűen kiválasztunk egy tippet a letöltött listából és megjelenítjük. A felhasználó két irányba is léptetheti a tippeket. A listát a felhőben tárolom, és ott lehet módosítani, kibővíteni azt. Amennyiben időközben hozzáadunk egy tippet a szerveren, az applikációban az Adatok Frissítése gomb megnyomása után már az aktuális listában lapozhatunk körbe.

6.1.4 FUNKCIÓK KÉPERNYŐ

A második képernyőn találhatóak az interaktív lehetőségek. Innen tud a felhasználó etetést és itatást indítani. Itatás alatt a tálban lévő víz teljes cseréje értendő. Az etetés indításához a felhasználónak meg kell adnia az adagolni kívánt mennyiséget gramm-ban. Az Etetés Most gombra kattintva az applikáció feltölti a szükséges paramétereket a felhőbe, aminek következtében maximum 5 másodpercen belül az Okos Kisállat Etető megkezdzi az etetési folyamatot a kívánt mennyiséggel.

Az itatáshoz, vagyis az ivóvíz frissítéséhez meg kell adnunk egy időtartamot másodpercben, amíg az egyik víz pumpa leszívja a használt ivóvizet, a másik víz pumpa pedig feltölti azt a friss víztárolóból. A Funkciók képernyő a 6-2. ábrán látható alább.



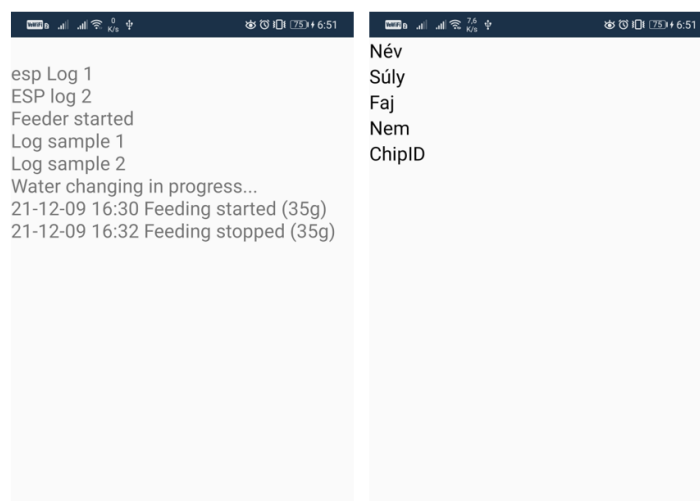
6-2. ábra Applikáció - Funkciók Képernyő

Ezen kívül itt található a Kamera indításáért szolgáló gomb. Ez egy új ablakban megnyitja a kamera menüt, ahogy az a 6-3. ábrán is látszik. Itt el tudjuk indítani a képet, illetve megállítani. Ebből az ablakból a Funkciók képernyőre térünk vissza.



6-3. ábra Applikáció - Kamera Ablak

A napló menüpontról megtekinthetjük a felhőben tárolt naplóbejegyzéseket, amelyeket az Okos Kisállat Etető töltött fel. Itt megtalálhatóak az etetések, itatások, illetve adatmódosítások naplózásai (6-4. ábra)



6-4. ábra Applikáció - Napló és Kisállat Adatai Ablakok

Végül pedig az utolsó gomb a kisállatunk adatainak megadásához ad felületet. Itt meg tudjuk adni a nevét, állatfaját, korát, nemét, chipid-t és esetleg tetszőleges adatokat is.

6.1.5 BEÁLLÍTÁSOK KÉPERNYŐ

Az utolsó képernyőn találjuk meg az etetések és ivóvízcseré időpont-beállításait. Erről képernyőfotó a 6-5. ábrán látható. Az időpontokat óra és percre pontosan lehet beállítani. Minden időpontot ki- és bekapcsolhatunk, hogyha nem szeretnénk, hogy lefusson az adott folyamat, mert például kevesebbszer kell az állatnak étkeznie egy napon. Ebben az esetben az etetés vagy vízcseré időpontja megmarad későbbre.

Az etetések egyes eledel mennyiségét itt is grammal, külön-külön adhatjuk meg, akár az összes alkalomra egyedi értékkel. Az itatásoknál megint csak másodpercben kell értéket adnunk, ami a vízcseré folyamatának hosszát állítja be. Ezeket egyelőre külön-külön lehet állítani mert ha az állatnak egyedi az étrendje ez fontos lehet, illetve tesztelési célokból is hasznos.

6-5. ábra Applikáció - Beállítások Képernyő

A Mentés Felhőbe gombbal tudjuk felülírni a szerver adatait az új, módosított értékeinkre. A Visszaállít gombbal pedig, ha volt elmentetlen érték, azt visszaállítja a szerveren szereplő, előző, eredeti értékre.

6.1.6 A FELHASZNÁLÓNAK VALÓ VISSZAJELZÉSEK

Fontos szempont, hogy a felhasználó tisztában legyen azzal, ha etetés, itatás történt. Erről valamilyen visszajelzést szeretnénk küldeni neki, így megbizonyosodhat, hogy az etetés vagy víz csere sikeresen elindult. Ezt akkor is meg kell oldania az applikációnak, amikor az nem fut a telefonon. Erre valók az értesítések. Értesítések küldésével el tudjuk érni, hogy üzenetet küldjünk az applikáción kívülre, így azt bezárás után is megkapja a felhasználó.

A jelenlegi verzióban az értesítések implementálására nem került sor, ugyanakkor toast-üzeneteket, azaz felugró ablakokat küldök az applikáción belül, melyek figyelemfelkeltőek. Sajnos ezeket csak akkor látja a felhasználó, ha az applikáció aktívan fut. Emiatt az értesítéseket is tervezem bevezetni.

6.1.7 EGYÉB TECHNIKAI INFORMÁCIÓK

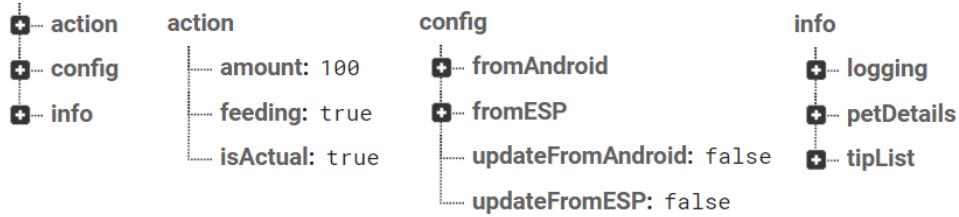
Magát az Android applikációt .NET-es Android Xamarin keretrendszerrel készítettem el. [68] Alapvetően az Android 11.0 (R) verziót veszi célba. A minimum Android verzió, amin futnia kell az Android 5.0 (API Level 21 - Lollipop). Az applikációnak szükséges engedélyek közül egyedül az internet elérést, illetve a hálózati állapot lekérdezését kellett bekapcsolni. (INTERNET, ACCESS_NETWORK_STATE)

Ennek a következménye, hogy az applikációnak jelenlegi formájában internet elérést kell biztosítanunk, azaz, ha a készülék nem kapcsolódik valamilyen interneteléréssel rendelkező hálózathoz, az applikáció nem fog elindulni. Ezt a későbbiekben kijavítom, ugyanakkor ez logikus is, mivel a legtöbb funkcióhoz feltétel, hogy elérjük a Google felhőszolgáltatását, amin keresztül küldjük és kapjuk az adatokat az Okos etetőtől. Ezen internetet igénylő funkciók alól kivételt képez, és logikus is lenne például a gazdik számára kínált állattartási tippek offline használata, illetve, az az eset, ha a felhasználó módosítana egy etetési/itatási időpontot, vagy ad-hoc etetést indítana azonnal, ahogy az internetkapcsolat újra elérhetővé válik. Ezen felhasználási esetekre szeretném kibővíteni az applikációt egy lokálisan lementett konfigurációs adathalmazzal, melyet magán a készüléken tárolunk, ezzel teret adva a hasonló jellegű igényeknek.

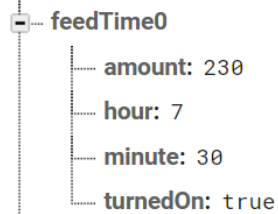
6.2 FELHŐ SZOLGÁLTATÁS

Minden adatot, konfigurációt és egyéb információt a Google Firebase felhőalapú, valós idejű adatbázisában tárolok. Az applikáció és az ESP32 között szükséges adatok itt mennek keresztül. Ezt alapvetően gyors, csekély mennyiségű adatküldésre, kommunikációra találták ki, így, ha hosszabb távra kell adatot eltárolnom a továbbiakban, például a naplózás vagy a tipp lista miatt, akkor ahhoz a Google Firestore adatbázis API felületét fogom felhasználni. A jelenlegi felhasználási célomra a Firebase az optimális felület. Magának a konfigurációnak a felépítése a 6-6. ábrán látható.

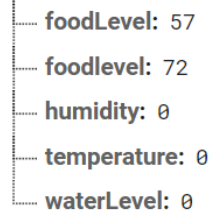
smartpetfeederfirebase-default-rtdb



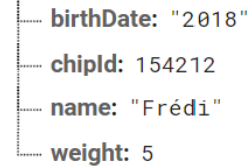
fromAndroid



fromESP



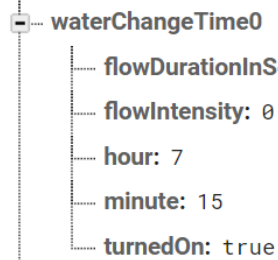
petDetails



feedTime1

feedTime2

feedTime3



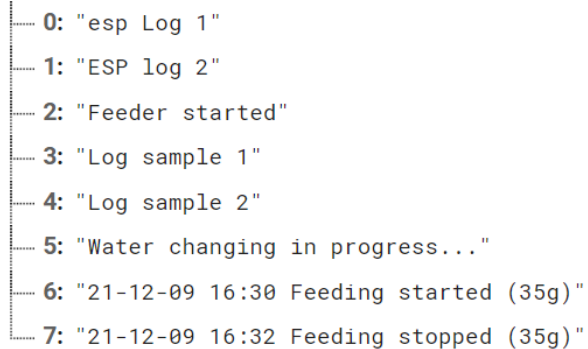
waterChangeTime1

waterChangeTime2

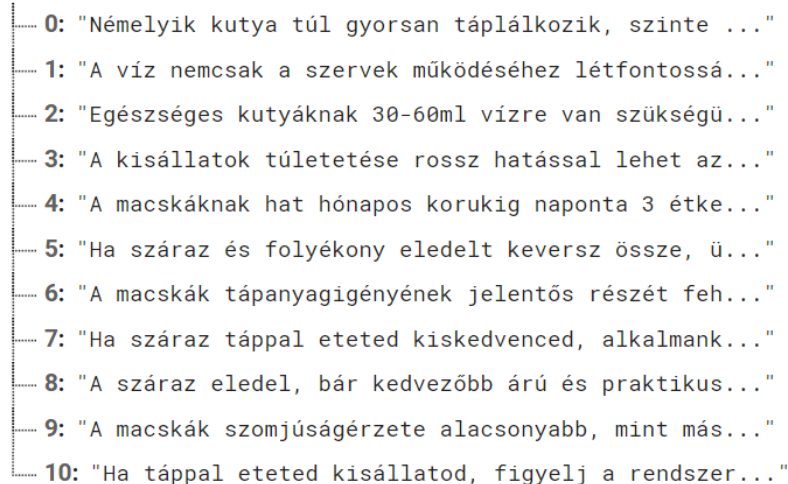
waterChangeTime3

logging

ESP32



tipList



6-6. ábra Felhőben Tárolt Adatbázis Struktúrája

6.3 OKOS KISÁLLAT ETETŐ

Ebben a fejezetben a rendszer Otthoni Környezetben lévő Okos Kisállat Etetőre vonatkozó komponenseket mutatom be. Azt mindenekelőtt tisztázni szeretném, hogy az alkatrészek megérkezése után fedeztem fel, hogy az ESP32-CAM nem fog elegendő GPIO pin-nel rendelkezni. Összesen 10 darab használható GPIO található rajta, ugyanakkor az felhasználni kívánt modulok és alkatrészek működtetéséhez körülbelül 15 darabra lesz szükség. Itt legjobb tudásom szerint kettő járható utat láttam megoldásként.

Az első, hogy valamilyen multiplexer modult alkalmazok az ESP32-CAM GPIO számainak kibővítéséhez. Ilyen például a HCF4051 vagy a 74HC4067. Ez minimális mértékben megnövelte volna a komplexitást, de olcsó megoldásnak mondható. A második, hogy egy az ESP32-CAM -tól független mikrokontrollerrel készítem el az Okos Kisállat Etető bázis részét vagy alapját, az ESP-CAM pedig csak egyetlen célt fog szolgálni, az pedig a kamerázás.

Ez igencsak előnyös lehetne, ugyanis így a kamera képét teljesen függetlenül használhatom az etető egyéb folyamataitól. Ugyan megnöveli a költségeket, de nem szignifikáns kiadás. Ráadásul lehetőséget ad arra, hogy az etető fizikai helyétől elszakadva, szabadon elhelyezhető egy független egységként. Ezáltal sokkal nagyobb és relevánsabb teret vehetek a középpontba. Természetesen ehhez szükség lesz egy hordozható áramforrásra vagy egy közeli hálózati csatlakozóra, illetve fontos, hogy a vezeték nélküli wifihálózat elégséges jelerősségű hatótávjában maradjunk.

Mivel az utóbbi megoldást választottam, megrendeltem egy ESP-32 mikrokontrollert is. A választás egyszerű volt, hiszen az 5.2-es fejezetben már összehasonlított értékeket megtekintve, ez tűnik ideálisnak a számos GPIO pin-je, vezeték nélküli, energiatakarékossági és biztonsági funkciói miatt.

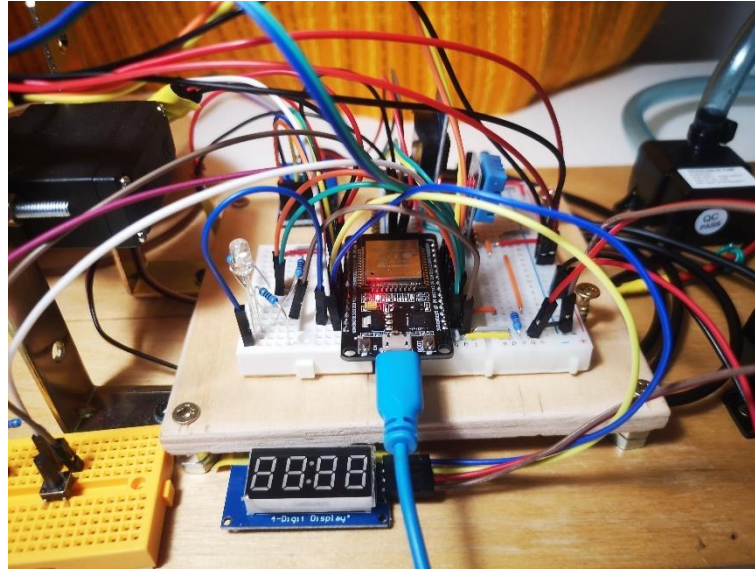
Ennek következményként a rendszerterv annyiban módosult, hogy a már meglévő koncepcióban szereplő mikrokontroller, az ESP32-CAM helyett egy ESP32 -vel fogom megépíteni az etetőt. Az ESP-CAM pedig egy ettől független egységként, de ugyanúgy a video stream-et fogja szolgáltatni.

A másik tisztázni való kérdés pedig a Lokális Felhasználói Interfész. Bemeneti perifériának eredetileg egy 4x4-es mátrixú, membrán billentyűzetet terveztem, kimenetként pedig egy 16x2 karakteres LCD kijelzőt. A rendszerem jelenlegi felépítése mellett, ezek alkalmazásához szükségem lenne demultiplexerekre, viszont ezek beszerzéséhez már kicsúsztam az időből.

A jelenlegi változatban egy 4 számjegyű, 7 szegmenses kijelzőt alkalmazok, egy RGB LED-et és egy nyomógombot erre a célra. Viszont a későbbiekben a fent említett alkatrészekre szeretném cserélni őket.

6.3.1 FŐ VEZÉRLÉS

Az Okos Kisállat Etető vezérlését az ESP32 mikrokontroller végzi. Ennek a rendszerben való, fizikai elhelyezkedését az alábbi, 6-7. ábrán láthatjuk.



6-7. ábra ESP32 Mikrokontroller és bekötése

A vezérlés fő lépéseiről az 5-3. ábrán már felvázoltam egy tervet. Az ábra jelenleg is megállja a helyét, ugyanakkor a vezérlés fő szálának egyes lépéseit ebben az alfejezetben most szeretném kibővíteni és részletezni azokat, a pontos, valós sorrend igénye nélkül.

A rendszer inicializálásakor először a modulok, illetve segédmodulok adott értékeit állítjuk be. Ide tartoznak a modulok ki- és bemeneti pontjai, a hozzájuk tartozó változók és konstansok kezdeti paraméterei, a konfiguráció struktúrájának deklarációja, valamint a felhő kapcsolatnál autentikáláshoz szükséges hitelesítő adatok. Ezt követi az adatok felhőből való lekérése, amikor a szerveren tárolt, legfrissebb adatokat letöltjük. Utána ezeket az értékeket beállítjuk a mikrokontroller memóriájában létrehozott konfigurációs objektum típusváltozóinak.

Következő lépésként az összes szenzorral elvégzünk egy kezdeti mérést és feltöltjük a friss értékeket a felhőbe. Így az applikáció rögtön a friss adatokat kapja, ha azt megnyitják vagy a felhasználó ráfrissít. A valós idejű óra értékét kiírjuk a kijelzőre, óra és perc pontossággal. Végül pedig a státusz LED-et a jelenlegi Inicializálás színről a készenléti állapotnak megfelelő színre állítjuk.

Az ezt követő vezérlési szakaszban a rendszer valamilyen bemenetre vár. Ez már maga a „loop” metódusa a mikrokontrollernek, így folyamatosan ismétlődik. Viszont az nem teljesen igaz, hogy vár, mert folyamatosan aktívan keres, de mindenképpen valahány lehetőség, előre meghatározott esemény bekövetkezését figyeli.

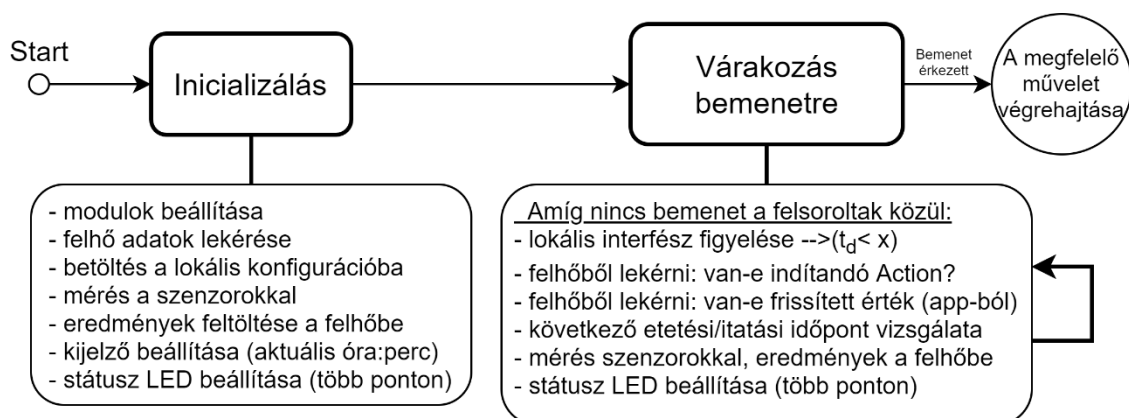
Először a lokális felhasználói interfész bementi értékét figyeli, amely a nyomógomb lenyomásával aktiválódik. A gazdinak ezen a ponton, x másodpercig, lehetősége van megnyomni a gombot. Ha ezt időben megteszi, aktiválja magát a lokális felhasználói interfészt. Ennek a procedúráját egy későbbi fejezetben részletezem.

Ha a felhasználó nem él ezzel a lehetőséggel, a rendszer lekérdez egy kijelölt értéket a felhőből, ami alapján eldönti, hogy van-e aktuálisan kijelölt parancs, amit végre kell hajtania. A lehetséges parancsok az azonnali etetés vagy vízcsere indítása. Amennyiben van, végre hajtja azt, majd átbillenti a felhőben a megfelelő értéket hamisra, így azt nem fogja a következő ciklus futásában, tévesen, újra végrehajtani.

Ezt követően azonos módon lekérdez egy másik értéket, mely azt hivatott jelölni, hogy van-e módosult adat, amit le kell, hogy töltsön a szerverről. Ez akkor lesz igaz, ha a felhasználó az applikációból módosított egy értéket, és elmentette azt. Itt az etetések és vízcserek időpontjairól, illetve a hozzájuk tartozó mennyiségekről van szó. Amennyiben van módosítás, az összes etetési és itatási beállítást lehúzza és felfrissíti a konfigurációs objektumban.

Ezután lekéri az RTC órával a pontos időt és megvizsgálja az objektumban szereplő időpontokat. Végigmegy rajtuk, és ha az egyik időszerű, végrehajtja azt, majd folytatja a lista átnézését másik esedékes feladatért. Azt, hogy esedékes-e egy időpont, egy megadott küszöbérték segítségével tudja eldönteni. Ha a vizsgált időpont és az aktuális idő különbsége kisebb, mint ez a megadott küszöbérték, akkor indítja az akciót.

Ezek után mérést végez a szenzorokkal. Abban az esetben, ha etetés vagy itatás is történt, a víz és az eledel tartó aktuális töltöttségi szintjét is megméri, különben nem. A friss eredményeket a felhőbe küldi. Végül a státusz LED színét állítjuk a megfelelő értékre, és kezdődik a loop metódus előlről, megismételve a lépéseket. A folyamat egyes lépéseit a 6-8. ábrán szemléltetem.



6-8. ábra A Vezérlés Fő Szála

6.3.2 KONFIGURÁCIÓ FELDOLGOZÁSA

A konfiguráció értékei a felhőben vannak tárolva. Amikor se az applikáció sem a mikrokontroller nem üzemel, az értékei nem módosulnak, nem vesznek el. Az applikációból feltöltendő értékeket a „/config/fromAndroid” elérési úton tároltam. A mikrokontroller által módosított értékek pedig a „/config/fromESP” útvonalon találhatók. A két kliens, az applikáció és a mikrokontroller, egymás útvonalán található értékeket nem módosíthatja, így ezt az esetet nem kell lefednem. Az egyedüli értékek, amiket mindketten módosítanak az az „/action/isActual”, „/config/updateFromAndroid” és „/config/updateFromESP”. A kliensek a saját adataik felhőben való módosításakor átbillentik a hozzájuk tartozó, „/config/from...” kezdetű logikai értéket igazra. Majd a másik fél a friss adatok lementését követően, visszabillenti azt hamis logikai értékre.

Mindkét kliensről igaz, hogy inicializáláskor mentik le a számukra releváns adatokat. Erre a célra konfigurációs objektumokat hoztam létre. Az applikáció esetében a teljes konfigurációt lementjük, mivel a felhasználó innen tudja azokat megvizsgálni. A mikrokontroller ezzel ellentétben csak a rá ténylegesen releváns információkat kéri le, ezzel is optimalizálva a kritikus folyamatot.

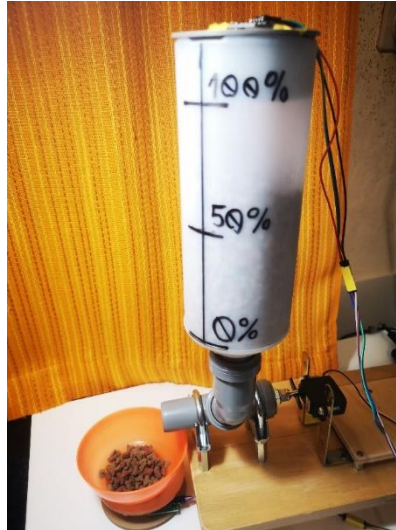
6.3.3 ETETŐ RENDSZER

Az Okos Kisállat Etetőm egyik, ha nem a legfontosabb, komponense maga az Etető Rendszer. Ebben az alfejezetben először ismertetni fogom milyen részekből épül fel, majd rátérek, hogy hogyan működik pontosan, valamint bemutatom a prototípusom sajátos megoldásait ábrákkal.

A megvalósított etetési mechanika kiválasztásának okait már a 2.12, a 3.2.1, és az 5.3-as fejezetekben részletesen leírtam, így ezt most itt nem tárgyalom ismételtén. Amit viszont fontosnak tartok az maga a megvalósítás, illetve annak részletei.

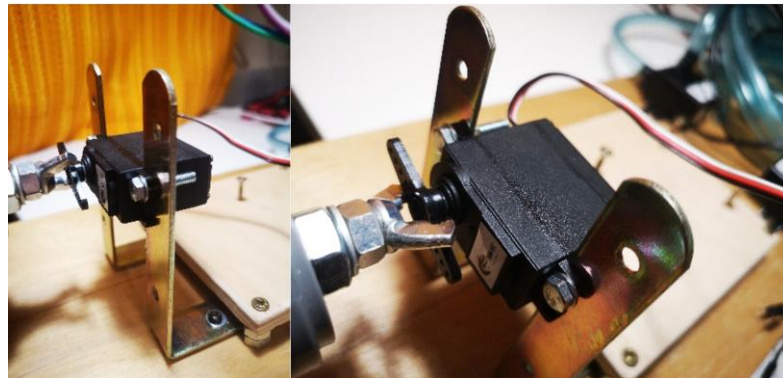
Nem véletlenül találkozhattunk annyiféle megoldással az irodalomkutatás során, hiszen nincsen tökéletes megoldás. Az összes megvalósítási módszer jó valamilyen szempontból, de mindegyiknél a részleteken múlik a győzelem. Ugyanis egy ilyen apró volumenű és viszonylag precíz rendszerrel nem mindig egyértelmű a sikeres kivitelezés.

Magát a hardvert logikailag négy jól elszeparálható részre lehet bontani. A szervomotor és az általa meghajtott csiga. A csőrendszer, amelyben a csiga forog és a táp behullik. Az edetel tartály, melybe a gazdi betöltheti a tápot. Illetve az edetel végállomása, a mérleg és valamilyen erre a célra alkalmas tá. Az elkészült Etető Rendszer a 6-9. ábrán látható.



6-9. ábra Etető Rendszer

A hajtást egy DS04-NFC típusú szervomotor végzi, amely 360°-os folyamatos, állandó elfordulásra képes. Ezáltal tetszőleges ideig forgathatom a lapátját a kiválasztott irányban. A szervomotort két derékszögű bútorvasalatra csavaroztam. A motor felfüggesztésének módja, illetve a tengellyel való kapcsolata a 6-10. ábrán látható. Egy adatkábelrel rendelkezik. Az 5V-os áramot pedig ajánlottan, külső tápegységről biztosítom.



6-10. ábra Etető Rendszer - Szervomotor

A szervoval hajtott csigát 3D nyomtatóval PLA anyagból készítettem, amely növényi cukorból készül és viszonylag környezetbarát. A modellt a 2.3-as fejezetben bemutatott projekt készítője bocsátotta közhasználatra. Annyi módosítás történt, hogy átméreteztem. Az új méret 27mm-es keresztmetszettel rendelkezik és körülbelül 80mm hosszú a csiga része, amely a 6-11. ábrán jobb oldalt látható. Ezt a lecsökkentett méretet a precízebb adagolás érdekében választottam, illetve a rendelkezésre álló PVC idom méretei alapján.

A végső verzióban a bal oldali képen látható 3D-nyomtatott alap és szélesített tengely tartó már nincsen a csigán, ugyanis letört. Erre a megoldás az volt, hogy a csiga tengely végére egy külső borítást húztam, amit átfúrtam, és ebbe a tartóba megy bele a főtengely

csavar, amit már a szervomotor hajt. A csavar végére egy szárnyas anyát helyeztem, amellyel egy körmös kapcsolóhoz hasonló hajtást érek el a szervomotor és a tengely között, így tehermentesítem a szervot és elkerülöm a fixálás egyéb hátrányait.



6-11. ábra Etető Rendszer - Csiga

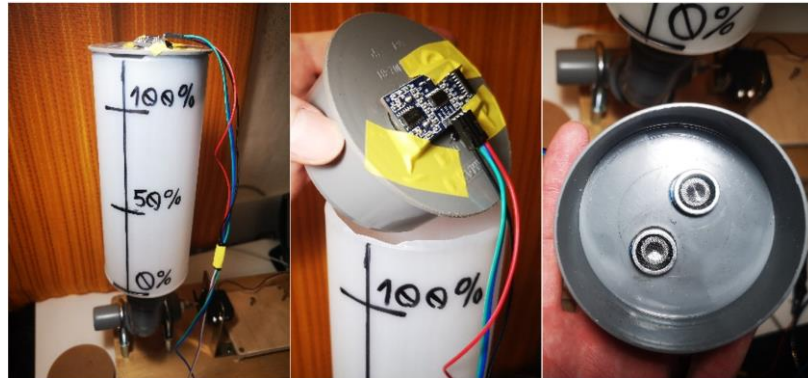
A csőrendszert egy PVC lefolyó T-idommal oldottam meg, amit kettő darab csőbilinccsel és egy menetes távtartóval rögzítettem az alaphoz, a 6-12. ábrán látható módon. A T-idom tágabb oldalán foglal helyet a szervomotor, a tetején pedig egy másik PVC cső toldás, amibe azeledeltároló edény vége kerül. Ezt a toldást azeledel hatékony befolyási szögének azelérése érdekében tettem a T-idomra, ugyanis függőleges helyzetben azeledel befolyásának a mértéke túl nagy volt, amit a szervomotorom elbírt volna, és mivel a tároló edényben fontos a gravitáció iránya a távolságmérés miatt, ezért ezt a 45°-os toldalékot helyeztem rá.



6-12. ábra Etető Rendszer - Cső Rendszer

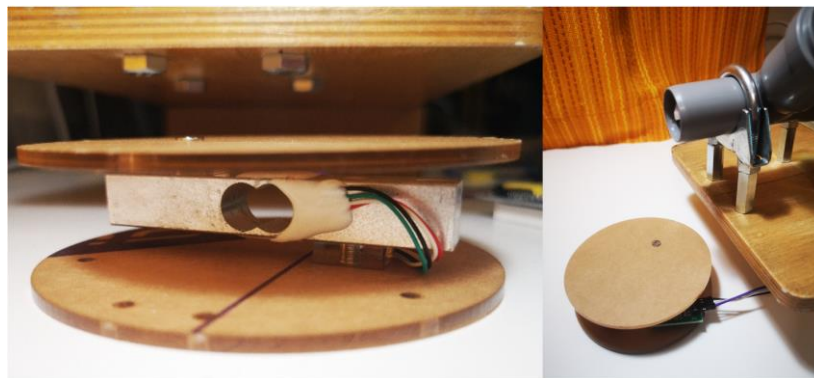
Azeledeltároló szerepét egy műanyag, 1 literes flakon látja el, melynek a tetejére egy arra illeszkedő tetőt vágtam ki, így azeledel kevesebb nedvesség éri. Magára a tetőre is fűrtam kettő lyukat, ahova a HC-SR04 ultrahangos távolságérzékelő került. Ezek az alkatrészek a 6-13. ábrán láthatóak. Az adatbázisban szereplőeledeltároló telítettségét jelző százalékos értéket ezzel a távolságérzékelővel számítom ki. Először teljesen üres tartállyal mértem, ami 21cm volt, majd feltöltöttem táppal a tárolót, amíg a szenzor még

helyes távolságokat mért, ami körülbelül 4-5 cm volt, és bejelöltem a tároló oldalán. A százalékszámításhoz a mért távolságot arányosítom úgy, hogy a 21cm jelenti a 0%-ot, az 5cm pedig a 100% -ot, így ez az érték mindig 0 és 100 között lesz. A csőben fennmaradó tápdarabokat figyelmen kívül hagyom, ugyanis azokat nem tudom pontosan megmérni, és a jelölésen is jól látható, hogy ez az érték csak a tároló töltöttségét hivatott mutatni.



6-13. ábra Etető Rendszer - Eledel Tároló

Az Etető rendszer utolsó eleme a mérleg, mellyel a kiadagolt eledel mennyiségét tudjuk ellenőrizni. A mérleg amelyet alkalmaztam egy HX711AD súly szenzor és mérőcella párosítása. A maximális terhelhetősége 5 kg, pontossága +/- 1mm. A mérleget egyelőre különálló egységként lehet mozgatni, az nincs rögzítve az alaphoz. Az első beüzemeléskor a szükséges kalibrációkat elvégeztem, így kijött egy szorzási érték, amellyel a mérleg pontosan tud mérni. A mérleg a 6-14. ábrán látható.



6-14. ábra Etető Rendszer - Mérleg

Az Etető Rendszer működési elve igen egyszerű. Ha etetési folyamatot indít a vezérlés, a szervó teljes sebességgel kezdi forgatni a csigát. Ilyenkor a táp automatikusan folyik befele az adagoló csőbe, ahol a csiga kitolja azt a nyíláson. Ez az a kritikus pont, ami miatt a részletek jelentős szerepet játszanak.

A befolyó nyílás nagyságán és döntési szögén múlik, hogy egységnyi idő alatt mennyi tápot kell a szervomotornak kitolnia. Itt lehet szabályozni az adagolást a mérleg

viSSZACsAtolt értéKének segítségével, viszont mivel egy viszonylag kis teljesítményű szervomotort alkalmazok, így az szervo forgási sebességét nem érdemes csökkentenem, különben erőlködik. Egy nagyobb és erősebb szervoval ez lehetséges lenne. A jelenlegi vezérlési kódom szerint a szervo addig forog, amíg el nem éri a kívánt értéket. A későbbiekben tesztelni fogom, hogy ezzel a szervoval milyen mértékben tudok szabályozni.

Etetési folyamatot kétféleképpen indíthat a rendszer. Az első, ha a felhasználó ad-hoc módon indítja azt. Ez történhet az applikációból vagy a lokális felhasználó interfészeiről. Az applikációból indítva szükséges megadni a kiadagolandó mennyiséget grammal. Ezt az értéket az applikáció feltölti a felhőbe és a mikrokontroller a következő vezérlési ciklusban végrehajtja azt a kívánt mennyiséggel.

Ha viszont a lokális felhasználói interfészeiről kezdeményezzük az etetést, egy fix, előre bekonfigurált mennyiséget fog a rendszer kiadagolni. Ez jelen esetben a forráskódban lett definiálva 200 grammal. Ezt az értéket a későbbiekben szintén konfigurálhatóvá szeretném tenni oly módon, hogy azt a mikrokontroller lementse az EEPROM memóriájába, így nem lesz szükség az adat lekérésére a felhőből, azaz internetkapcsolatra sem. Tulajdonképpen a lokális interfésznek ez lenne az egyik fő szerepe. Amennyiben nincs internetkapcsolat, az alapműveleteket a felhasználó helyben, kézikönyv is el tudja majd indítani.

6.3.4 ITATÓ RENDSZER

Az Itató Rendszer az Etető Rendszer mellett szintén kiemelt prioritással rendelkezik. A víz bizonyítottan fontos az állatoknak, csakúgy, mint az embereknek. Emiatt érdemes ösztönözni őket az ivásra. És ahhoz, hogy ösztönözni tudjuk az állatokat az ivásra nem elég, ha kiteszünk nekik egy tál álló vizet, még akkor sem féltlenül, ha az friss. Az állatok tipikusan kevesebbet isznak a kelleténél, ezért minden olyan lehetőséget kikell használnunk, amikor megitathatjuk kiskedvenceinket. Az én itató rendszeremben az itatás alatt nem szimplán a víz kipumpálását értem, hanem az elhasznált, állott víz megelőző kiszívását, majd a friss víz bepumpálását. Végeredményben lecseréljük a kisállat vizét, attól függetlenül, hogy mennyit hagyott az ivó tájában. Ezáltal az állatnak szinte egész nap friss vize lesz, hiszen naponta többször is lecserélhetjük azt a rendszerrel.

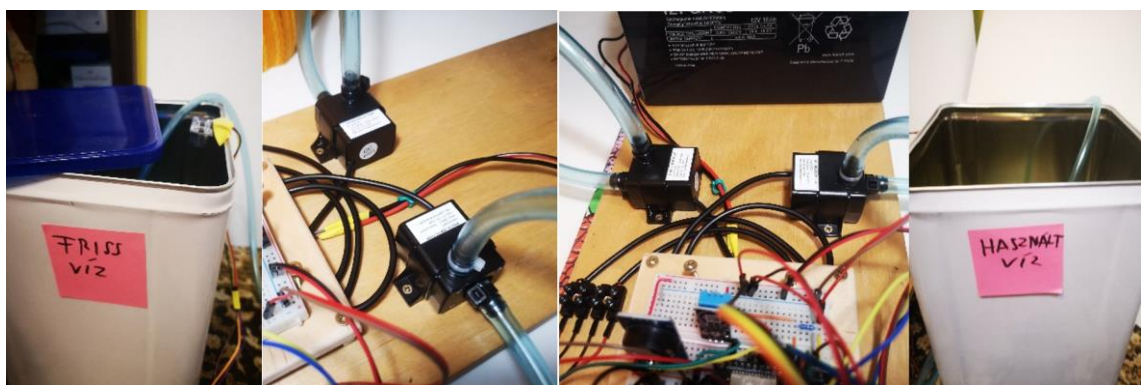
Eredetileg csak etetőt szerettem volna építeni, de a kutatások során hamar felismertem, hogy az állatoknak az ivás ugyanúgy fontosabb az evésnél, mint ahogy az az embereknek. Sajnos a piacon levő automatikus okos etetők és itatók erre a megoldást macskaszökőkúttal és a gravitáció által tálba eresztett vízszintkülönbséggel oldják meg. A vízszintkülönbséges módszer nagy hátránya, hogy a víz folyamatosan kivan téve a környezetnek, amittől koszos és higiéniatlanná válhat a kisállat ivóvize. A baktériumok elkerülése érdekében ez nem ajánlott módszer. Az ivószökőkutaknál már nem áll

folyamatosan a víz fedetlenül, hanem a rendszer forgatja azt. A forgatás során a szűrőberendezésnek köszönhetően a viszonylag nagyobb szennyeződések, port, szöszöket el tud különíteni, de a baktériumok ugyanúgy benne maradnak a körforgásban, amíg azt le nem cserélik. Úgy gondolom, hogy az én rendszerem ivóvizet cserélő megoldása hatalmas előnyt jelenthet a konkurens itatókkal szemben.

A működtetés a következő lépésekből áll. Először is szükség van egy frissvíz-tárolóra, ami lehetőleg fedett legyen, illetve egy használtvíz-tárolóra is, amit már nem feltétlenül kell lefedni. Első lépésben a vízpumpa a fennmaradt vizet a kisállat ivó táljából a használtvíz-tárolóba szivattyúzza a csövön keresztül egy megadott ideig. Ezután a másik pumpa a frissvíz-tárolóból pumpálja bele a tálba a vizet, szintén a vezérléstől megkapott időtartamig. Ezt az értéket a felhasználó adhatja meg, aki így könnyen kitapasztalhatja, hogy mennyi víz szükséges a kisállat tájának teletöltéséhez.

Magát a használtvíz-tárolót érdemes lehet más szempontból is megközelítenünk. Mivel a pumpával egy bizonyos távolságon belül bárhova el tudjuk juttatni a használt vizet, ezért érdemes lenne neki valamilyen alternatív hasznosítást találni. Például lehetne ezzel a vízzel növényt locsolni, vagy olyan helyre gyűjteni, ahonnan még hasznosíthatjuk azt, de semmiféleképpen nem a lefolyóba engedni. Természetesen ez csak egy ötlet a hasznosítására, de korlátlan lehetőség van még. A prototípusomhoz a tesztelés érdekében kettő darab fém dobozt használtam, mivel ezek lefedhetőek és viszonylag nagy tárolókapacitással bírnak. A tisztavíz-tárolóra kerül felhelyezésre még a második HC-SR04 típusú, ultrahangos távolságérzékelő modul. Ezt a tároló falának a peremére rögzítettem, ideiglenes megoldásként.

A két vízpumpa, amit alkalmazok erre a feladatra, 12V-on üzemel és kefe nélküli DC motorral rendelkezik. Percenként 4 liter kiszívására képes. El lehet méríteni, ugyanis vízálló, és 3 m magasságig képes felpumpálni a vizet. A víztárolók és pumpák a 6-15. ábrán tekinthetők meg.

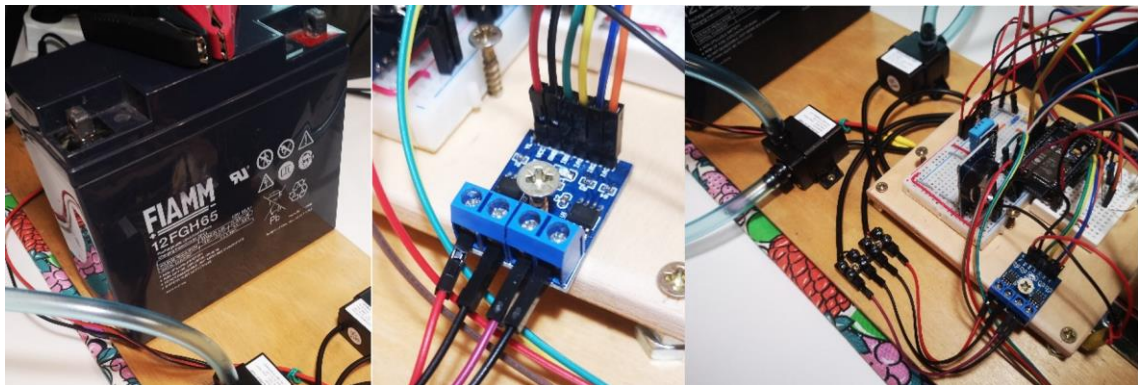


6-15. ábra Itató Rendszer - Víztárolók és Pumpák

Magukat a pumpákat a hozzájuk kapott műanyag bilincsekkel, lecsavarozva rögzítettem az alaphoz, így azok nem tudnak elmozdulni működés közben. Az összesen 4 méternyi, 8mm-es belső keresztmetszetű mipolán csöveket gyorskötöző segítségével szorítottam rá a pumpák szívó és nyomó ágaira, a légmentes működés biztosításához.

Mivel az áramkörben minden egyéb modul 5V, illetve 3,3V feszültséggel üzemel, szükség van egy külső, 12V-os áramforrásra is a működtetéshez. Én erre a célra egy 12V 18Ah-s akkumulátort használok fel, mivel ez állt rendelkezésemre, de a hálózatra köthető tápegységet ajánlom, ugyanis az kedvezőbb áron beszerezhető. Az akkumulátor és az általam megvalósított megoldás a 6-16. ábrán látható.

Mindazonáltal, amennyiben az akkumulátor mellett döntünk, úgy azt a teljes áramkörben hasznosíthatjuk feszültségszabályozóval, így teljesen függetlenül a hálózati áramtól tudnánk működtetni a Kisállat Etető Rendszert, ezzel elhárítva az esetleges áramszünetek következményeit.



6-16. ábra Itató Rendszer - Akkumulátor és Motorvezérlő

Az áramforrás mellett még szükségünk van egy motor vezérlőre, amivel a pumpákat be- és kikapcsolhatjuk a mikrokontrollerről irányítva. Ezt relével [69] is megoldhatjuk, de a motorvezérlővel nemcsak zárásra és nyitásra van lehetőségünk, hanem a motorok irányának váltására és sebességük szabályozására, valamint az ára is kedvezőbb.

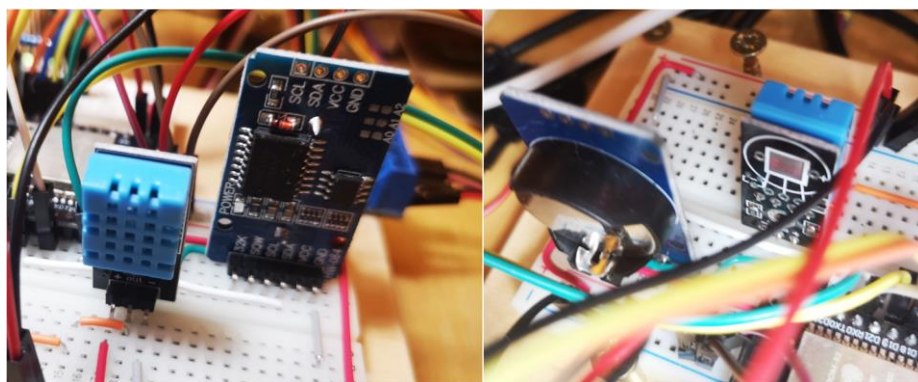
Az irány jelen esetben adott, ezek a vízpumpák nem működnek visszafelé és sebességüket nem fogom szabályozni, mivel visszacsatolás hiányában sokkal kiszámíthatóbb, ha teljes sebességgel, inkább az időtartam alapján vezérlem őket.

Motorvezérlőnek az L9110 típusú, két csatornás H híd modult választottam. Ez alkalmas 12V feszültségen, maximum 800mA áramot átengedni. Összesen négy darab bemeneti adat kiállása van, amiket a mikrokontrollerhez csatlakoztattam. A modul szintén az 6-15. ábrán látható.

6.3.5 SZENZOROK

A kettő darab ultrahangos távolságérzékelő mellett egy DHT11-M0 hőmérséklet és páratartalom mérőt alkalmazok. A DHT11 gyengébb specifikációkkal rendelkezik, mint fejlettebb verziója, a DHT22. [70] Ezek tökéletesen megfelelnek hobbi szintű felhasználásra, mivel nem túl gyorsak és pontosak. Maga a szenzor két részből áll: egy kapacitív páratartalom szenzorból és egy termisztorból. Egy általános chipet alkalmaz, amely analógból digitálisra konvertálja a szenzorok értékeit és kiküldi azokat. [69] Ezzel a hőmérsékletet és páratartalmat minden egyes ciklusban meg tudom mérni és feltöltöm a felhőbe, hogy az applikáció mindig a legfrissebb értékekkel rendelkezessen.

Ezenkívül a rendszer alkalmaz még egy DS3231+AT24C32 valós idejű órát, aminek a memóriáját egy gombelem táplálja, így az a rendszer kikapcsolásánál és újraindításánál nem veszti el értékeit. Ezáltal folyamatosan rendelkezésre áll egy igen pontos óra, ami körülbelül 2 évig üzemel a hozzátartozó CR2032 -es gombelemről. Az RTC modultól lekérve a pontos időt, a rendszer vizsgálja, hogy van-e esedékes etetés vagy itatás, illetve a lokális felhasználói interfész, 7 szegmenses kijelzőjén jeleníti meg a pontos időt. A két modul a 6-17. ábrán található.



6-17. ábra DHT és RTC modulok

6.3.6 KAMERA

Egy másik, a gazdik számára hasznos funkciója a rendszeremnek, az élő kamerakép megtekinthetősége az applikációból, interneten keresztül. Ehhez az ESP32-CAM mikrokontrollert használtam fel, ami eredetileg az Okos Kisállat Etetőben foglalt volna helyet, így fix pozícióból, viszonylag csak közelről tudott volna kamerázni, ám a 6.3-as fejezetben taglaltak miatt, különálló egységként fog üzemelni.

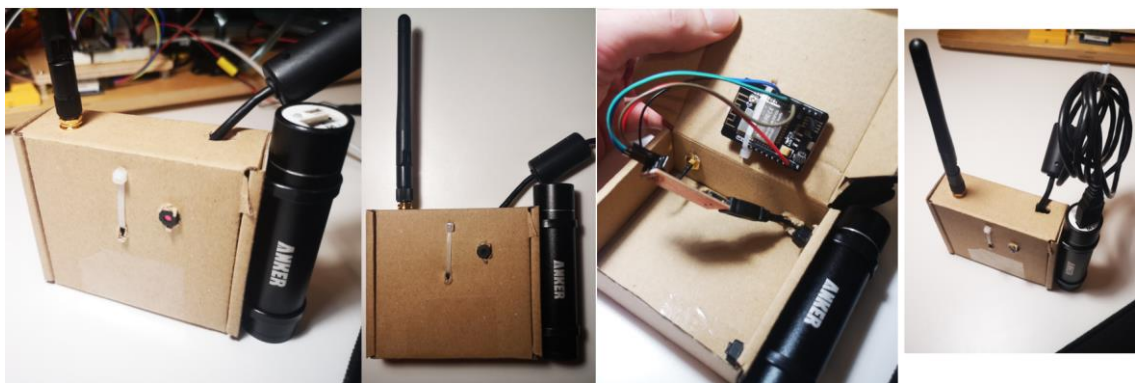
Magát a mikrokontrollert már az 5.2-es fejezetben kielemeztem. Jelenleg úgy van beprogramozva, hogy miután a WiFi-re felsatlakozott, a kiosztott IP címen egy megadott elérési úton elkezd a videóképet streamelni. Ez tulajdonképpen egy példa kód a mikrokontroller gyártóitól és minimális mértékben módosítottam csak. Viszont maga a

webszerver http protokoll alatt végzi a kommunikációt, ezért nem biztonságos. A jövőben szeretnék certificate-ekkel tanúsítványkezelést megvalósítani, így https-en is kommunikálhat majd a kamera az applikációval.

Mivel ebben a verzióban az adatokat titkosítatlan formában küldi a kamera szervere, úgy próbáltam meg biztonságosabbá tenni az elérését, hogy a stream útvonalába, vagyis az URL-be egy előre bekonfigurált, random generált számot tettem, hogy azt bárki más ne érje el olyan könnyen. Ez a random szám egy GUID, azaz egy Globális Egyedi Azonosító, aminek egyedisége matematikailag garantált. A kulcsok száma 2^{128} , így ahhoz, hogy valaki elérje a stream-et, ezt a 16 bájttal hosszú számot kell jól megtippelnie, aminek az esélye körülbelül 0. Ezt az egyedi azonosítót csak az android applikációm ismeri, így az eléri a stream-et a megadott URL-n és megkapja a képet az ESP32-CAM-tól.

Elsősorban lokális hálózaton próbáltam a stream-et, de port forwarding-gal az interneten is leteszteltem. A kamera webszerverén a portot is konfigurálhatjuk, így ki tudtam választani egy olyat, amit csak a kamera használ, más nem, így minden egyéb eszköz biztonságban maradt. Azonban indokolt lehet aggódni, ha a felhasználó kétes WiFi hálózatról lép be az applikációba, ahol egy „Man-in-the-middle” támadással a GUID kiszivároghat, és azt könnyedén eléri majd más is. Emiatt lesz fontos az adatok titkosítása.

A kamerának egy egyszerű védőházat is készítettem. Egy, a mikrokontrollerhez méretben hozzá illő kartondobozból vágtam ki a szükséges méreteket. A 6-18. -ábrán látható a végeredmény. Az ESP32-CAM beépített antennája jól működik általános felhasználáskor, viszont egy videó stream-elése igen nagy sávszélességet igényel. A jobb felbontásért gyorsabb WiFi-kapcsolatra van szükség. Ahhoz, hogy a sávszélességet meg tudjuk növelni az ESP32-CAM esetében, szükségünk van egy külső antennára, amit egy klipsz-szerű csatlakozóval illeszthetünk a lapkára. Ezzel körülbelül kétszer olyan nagy sávszélességet érhetünk el, mint a beépített antennával. [72]



6-18. ábra ESP32-CAM Független, Mobil Kamera Egység

Az említett külső antennát is beszereztem, viszont, ahhoz, hogy alkalmazni tudjam, először át kell forrasztanom egy apró ellenállást a belső és külső antenna áramköre között,

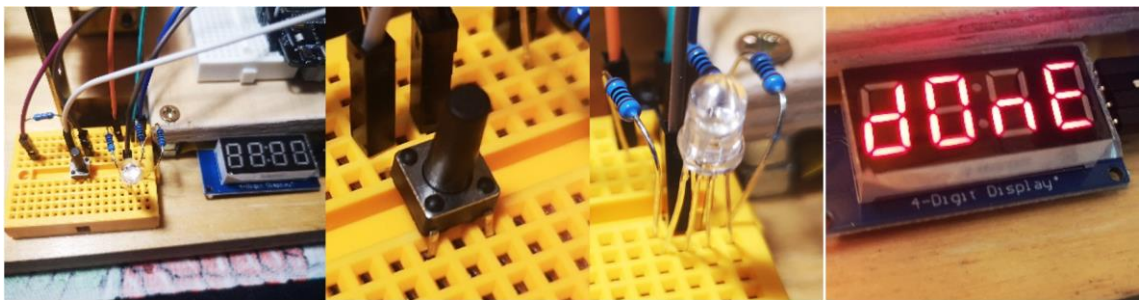
ám ehhez nem voltak meg a szükséges eszközeim. Az antenna helyét azonban kivágtam és beépítettem a prototipizálás kedvéért, és látható, hogy komfortos külsőt ad a doboznak.

6.3.7 LOKÁLIS FELHASZNÁLÓI INTERFÉSZ

A jelenlegi verzióban a lokális felhasználói interfészen egy nyomógomb, egy RGB LED és egy 4x7 szegmenses kijelző foglal helyet. Az interfészen keresztül a gazdi alap funkciókat indíthat, mint például etetés vagy itatás, és itt kap visszajelzést helyileg is az eseményekről.

Az egyes funkciókhoz különböző értékek tartoznak majd, és a rendszer az aktuális állapota alapján, egy státusz jelzést tud mutatni a LED-el a felhasználó részére.

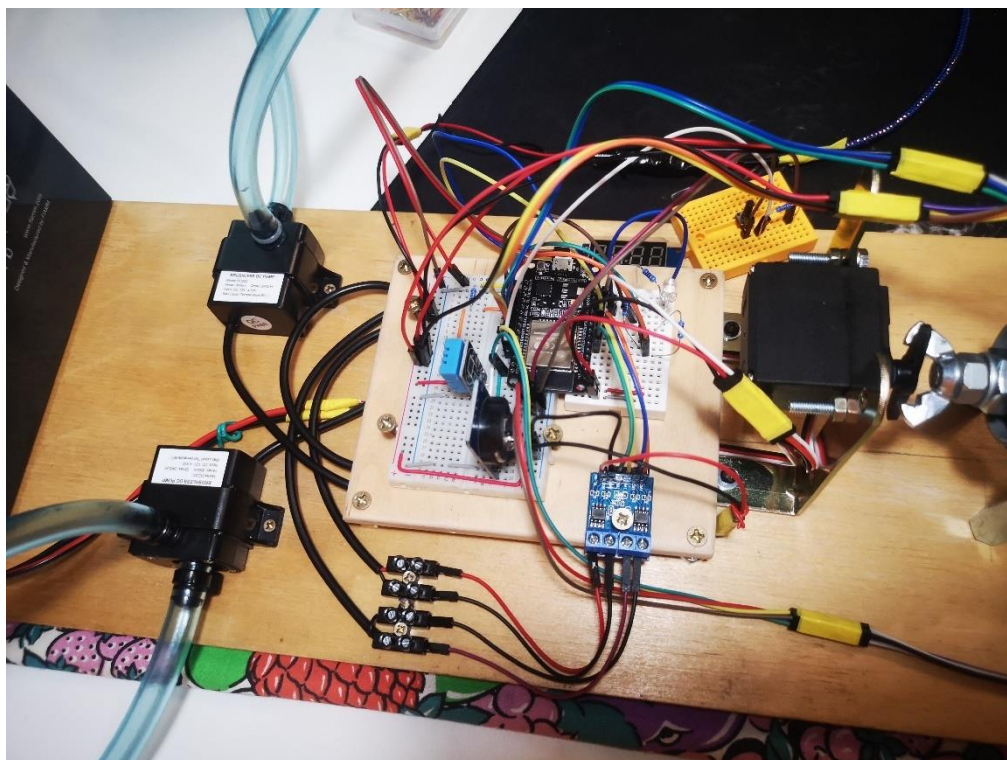
A nyomógomb pedig az egyetlen bemeneti perifériát jelenti jelenleg, így azt időzítéssel tudjuk helyesen használni. Az erre vonatkozó jövőbeni továbbfejlesztéseket a 6.3-as fejezetben már részletesebben leírtam. A Lokális Felhasználói Interfészt a 6-19. ábrán láthatjuk.



6-19. ábra Lokális Felhasználói Interfész

6.3.8 TELJES RENDSZER ÉS KAPCSOLÁSI RAJZ

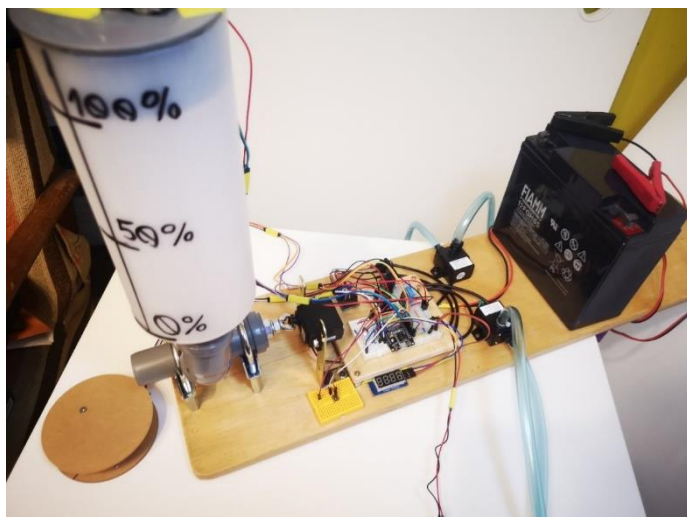
A teljes rendszerről átfogó képek, az összes alkatrészszel, modullal együtt, a 6-20., 6-21. és 6-22. ábrákon tekinthetők meg.



6-20. ábra Teljes Rendszer - Elektronika felülről

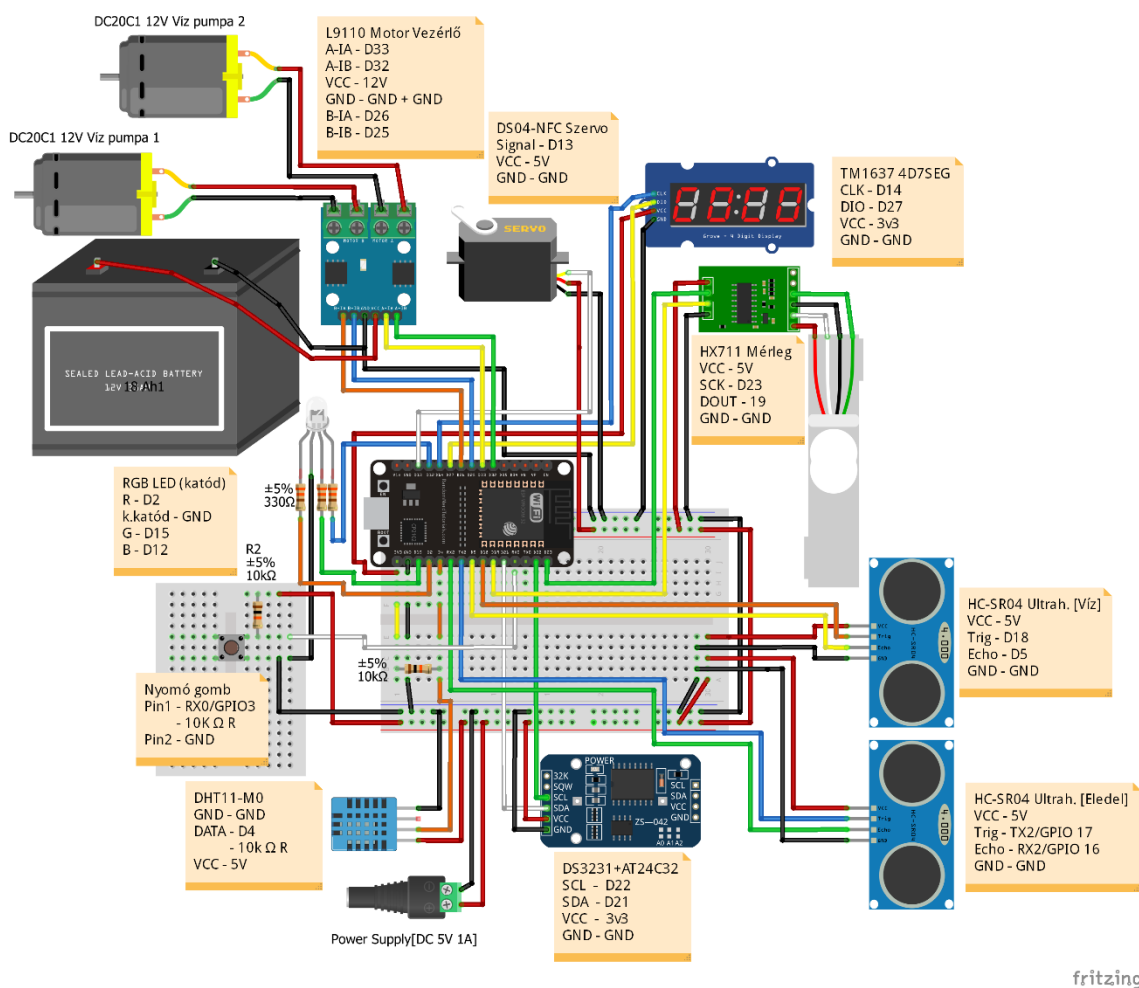


6-21. ábra Teljes Rendszer - Összes elem, kiegészítővel



6-22. ábra Teljes Rendszer - Oldalról

Az Okos Kisállat Etető rendszer teljes kapcsolási rajza a 6-23. ábrán látható.



6-23. ábra Teljes Rendszer - Kapcsolási rajz

6.3.9 KÖLTSÉGVETÉS

Az 5.11-es fejezetben már ismertettem a tervezett költségvetést, ám a felhasznált alkatrészek száma azóta kiegészült pár extra, be nem tervezett elemmel. Itt most az összes, végül felhasznált alkatrészt, modult, anyagokat felsorolom, amelyek a megvalósításhoz kellettek. Az árakat USD-ban adtam meg, és azok a szállítási költségeket is tartalmazzák a külföldről beszerzett alkatrészek esetében. Az időtállóság kedvéért, a jelenlegi árfolyam: 1 USD = 326.42 HUF.

Az Okos Kisállat Etető elektronikai költségei a 6-1. táblázatban szerepelnek.

6-1. táblázat Végző költségvetés - Elektronika

Alkatrész neve	Egységára
ESP32 CAM	5,5\$
ESP32	6\$
HX711 Mérleg	8\$
Szervomotor	8\$
Ultrahangos távolságmérő (×2)	0,85\$
Hő és páratartalom mérő	2\$
L9110 Motorvezérlő	1\$
Vízpumpa (×2)	4\$
RTC modul	4\$
4 Számjegyű 7 Szegmenses kijelző	2\$
RGB LED	0,3\$
Nyomógomb	0,05\$
Próbapanelek	3\$
12V (5A) Tápegység a vízpumpákhoz	10\$
Opcionális / Egyéb:	
Kábelek	2,2\$
ESP32-CAM Külső Antenna	2\$

Az Okos Kisállat Etető fizikai, anyag költségei a 6-2. táblázatban szerepelnek.

6-2. táblázat Végző költségvetés - Anyagok

Elem neve	Ára
Préselt falemez (ebből kb. a fele kell)	1\$
PVC mipolán cső (4m)	1\$/m
PVC T idom és 45°-os toldás	2,3\$
3D nyomtatás (csak anyag)	0,5\$
Egyéb: fém, csavarok, fa, bilincs	1\$

7 TESZTELÉS

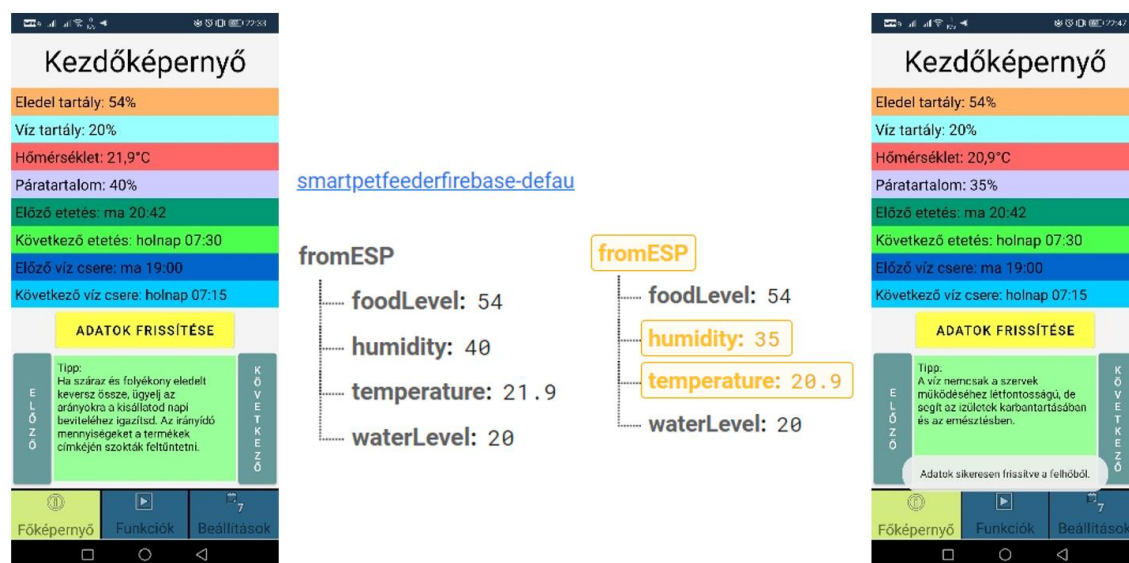
Ebben a fejezetben letesztelem az Android applikációt és az Okos Kisállat Etető legtöbb funkcióját. A kettőt szétválasztottam, mivel egymástól függetlenül is működnek, és ez megkönnyítette a tesztelést is.

7.1 APPLIKÁCIÓ TESZT

Az applikáció használata során azt állapítottam meg, hogy jól kezelhető, viszonylag kevés funkcióval rendelkezik és ezért könnyen átlátható, így gyorsan lehet navigálni benne.

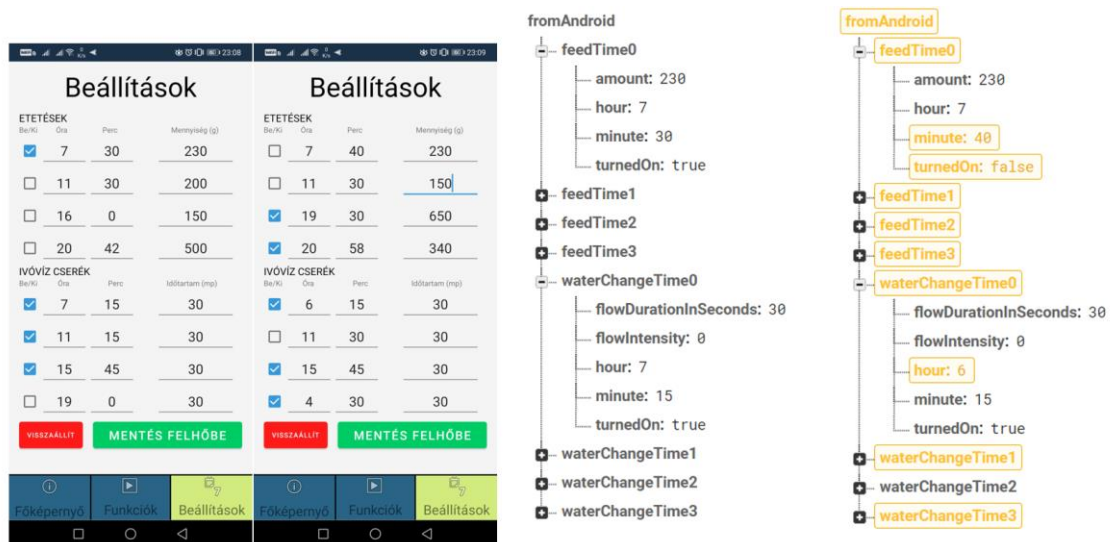
Az első teszt az applikáció futtatása és összehasonlítása volt két különböző eszközön. A fizikai eszköz egy Huawei P20 Pro, Android 10.0 API 29 frissítéssel, a virtuális eszköz pedig direkt egy régebbi (2015) modell, a Nexus 6P mely Android 5.0 API 21-es verzióval rendelkezik. A funkciók gond nélkül működtek mind a két eszközön.

A következő teszt, amit végrehajtottam, az az adatok lekérése és frissítése a felhőből. A 7-1 ábra bal oldali képernyőjén a kezdőértékek szerepelnek. A felhőben lévő értékek módosulása után, az Adatok Frissítése gombra nyomva, az új értékek jelennek meg, és megjelenik egy toast üzenet a képernyő alsó részén ezzel kapcsolatban.



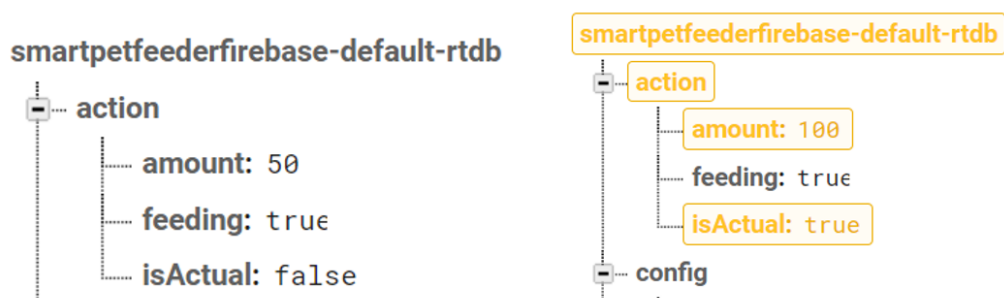
7-1. ábra Applikáció teszt - Adatok frissítése a felhőből

A harmadik tesztben az applikációban szereplő értékeket módosítom, majd mentek. Az értékeknek a felhőben is frissülniük kell. Ez a 7-2. ábrán látható.



7-2. ábra Applikáció teszt - Adatok mentése a felhőre

A következő tesztben egy etetést indítok a telefonról. A felhőben frissül a megfelelő érték.



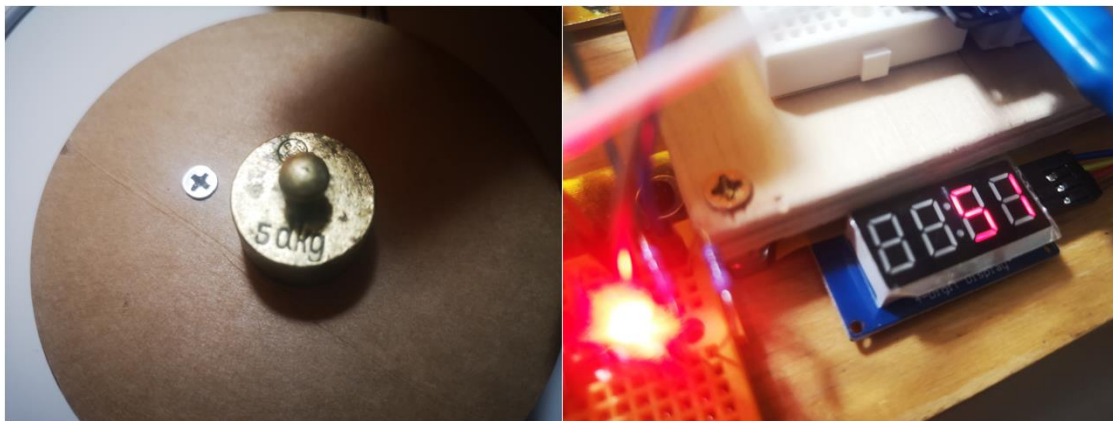
7-3. ábra Applikáció teszt - Etetés indítása

Az utolsó teszt, amit a telefonról végeztem az a kamera tesztelése volt. Körülbelül 2-3 méterre kell letenni az etetőtől, hogy éles képet kapjunk. Ez az antennával javítható lesz.

7.2 OKOS KISÁLLAT ETETŐ

Etetéskor a kívánt mennyiségtől maximum 10%-ban tért el a végső érték. Ez annak köszönhető, hogy szabályozást nem készítettem a rendszerbe, a szervomotor kímélése miatt. Ez a 10% egy erősebb szervomotorral és szabályozott meghajtással csökkenthető lenne.

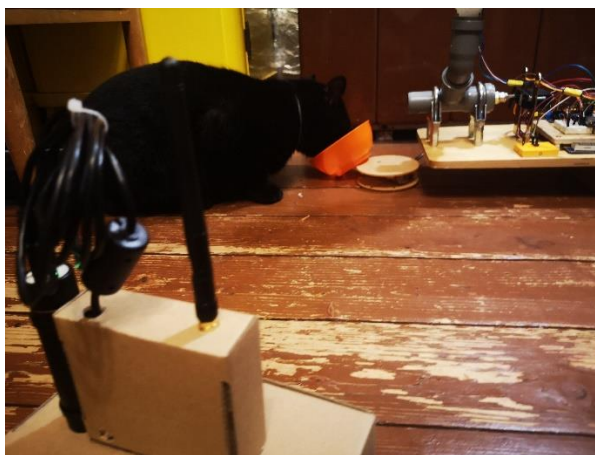
A mérleg teszteléséhez egy 5dkg -os súlyt használtam. A mérleg 51grammot mért, ami igen pontosnak mondható.



7-4. ábra Mérleg teszt

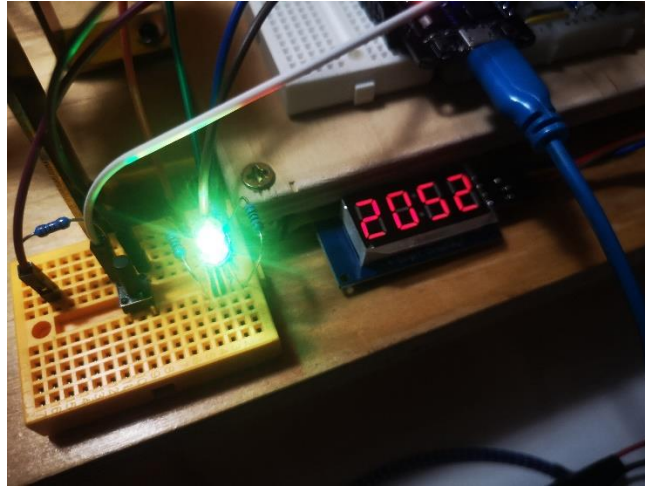
Az adatok lekérése és felhőbe való töltése gond nélkül zajlott.

Egy éles tesztet is végrehajtottam a macskáink segítségével. A mérleg magassága miatt a tálat lehúzta a földre, ezért az alacsonyabban kell megoldanom. 7-5. ábra.



7-5. ábra Éles teszt

A lokális interfészen minden látható volt, amit elterveztem: 7-6. ábra.



7-6. ábra Lokális interfész teszt

7.3 FIZIKAI JELLEMZŐK

Ahogy azt a 6-22. ábrán láthatjuk, a Kisállat Etető Rendszert egy rétegelt falemezre erősítettem. Ennek az volt a célja, hogy fix és jól hozzáférhető helyen tudjam a tesztek elvégezni. Ezzel a megoldással viszonylag gyorsan egy strapabíró alapra tudtam az alkatrészeket erősíteni. Természetesen ez csak egy prototípus. A cél a koncepció bemutatása, tesztelése volt. Helytakarékoság szempontjából a mikrokontroller alapját a cső mellett akár függőlegesen állítva, vagy akár a cső fölött tudnám elképzelni. Ezáltal hosszúságban körülbelül a felére lehetne csökkenteni a terméket, amelyre egy téglalap alakú védődobozt vagy keretet helyezhetünk, egy kör alakú nyílással, ahol az eledel cső irányul kifelé.

Konklúzió:

Az applikáció megfelelően működik, egyszerűen megtalálhatóak az egyes menüpontok és opciók. Az ablakok intenzív váltogatása során viszont felfedezhetőek minimális késések. Ez amiatt lehet, hogy a felhőből frissítem a szenzorok értékét. A víz beállításoknál nem biztos, hogy kell minden időpontra egyedi időtartam, ezt még kikell tapasztalni, ugyanis viszonylag egyedi megoldásról beszélünk, nincsenek best practice-ek.

A mikrokontroller vezérlési ciklusait feltétlenül optimalizálni kell. Feleslegesnek éreztem az adatok frissítését akkor, ha a felhasználó nem nézi meg azokat. Ebből arra következtetek, hogy az állandó értékfrissítés a felhőben nem feltétlenül hatékony megoldás. Sokkal inkább egy igényküldés alapján volna értelme az adatok mérésének és felhőbe töltésének. Erre szeretnék valamilyen megoldást találni a jövőben.

A másik gond, hogy a rendszeres mérésekkel és státuszváltozásokkal szintén az energiát pazaroljuk. Nem beszélve a ciklusonkénti óraellenőrzésről. Ezt ritkábban is elég volna megtenni, vagy az RTC könyvtárban lévő értesítési funkciókat alkalmazni. Mindenesetre amíg nem aktív az eszköz, addig célszerű lenne alvó állapotba küldeni azt. Az ESP32 és ESP32-CAM által is támogatott deep sleep, azaz mély alvás mód segítségével az eszközt olyan állapotba helyezhetjük amikor az szinte nem fogyaszt semennyi áramot. Ezáltal az akkumulátorról működtetett változatnak is sokkal kedvezőbb működési ideje lenne. Ha a felhasználó dönthetné el, hogy milyen gyakran szeretne frissített adatokat látni az applikáción akkor szintén hatékonyabban működne a rendszer.

8 TOVÁBBFEJLESZTÉSI LEHETŐSÉGEK

Az irodalomkutatás alatt több olyan ötletet láttam, amik hasznosnak és érdekesnek tűntek. Ebben a fejezetben ezeket az újításokat, illetve a megvalósítás és tesztelés fejezetek készítése során felmerült tovább fejlesztési ötleteket fogom áttekinteni.

Az első és legfontosabb a biztonság, így ezt mind az applikációnál, mind a kamera és a központi egység által szolgáltatott tartalomnál szeretném fejleszteni. Olyan biztonsági tényezőket lenne szükséges alkalmaznom, mint a jelszó, illetve, amennyiben rendelkezésre áll, úgy az ujjlenyomat azonosítást is. A kamera stream-jét az egyedi, véletlenszerűen generált URL -en kívül, titkosított kommunikációval, HTTPS protokoll alatt szeretném hosztolni az mobil applikáció felé, így a felhasználó biztos lehet benne, hogy az adatai, a beltéri kamera egység stream-je nem kerül illetéktelenek birtokába. Az ESP32 esetében pedig a már lekért adatokat az EEPROM memóriába mentenénk, így az internet kapcsolat hiányakor is tovább tudnánk üzemeltetni a rendszert.

Magát az Android applikációt illetően, szükségesnek tartom az értesítések küldését, mivel nem várhatjuk el a felhasználtól, hogy csak a mi applikációnkat figyelje és az legyen megnyitva egész álló nap. Ehelyett értesítéseket fogok küldeni, amikor valamilyen esemény történt az Okos Etetővel, így a gazdi rögtön tudomásul veheti azt és reagálhat. Egy másik ötlet még az offline mód bevezetése. Ennek megvannak az előnyei és hátrányai is, de azt mindenképpen implementálnom kell, hogy az app ne dobjon ki az internetkapcsolat hiányában. Az elvárt viselkedés valahogy úgy fog kinézni, hogy az alkalmazásban lépkedhetünk és a felhasználó tudtára adjuk, hogy nincsen kapcsolat. Az is megeshet, hogy csak elfelejtette bekapcsolni azt. Emellett az applikáció teljesítményét is szeretném növelni, a felhővel való kommunikáció és konfiguráció frissítés optimalizálásával.

Jogos kérdés lehet az is, hogy miért csak Android platformra készült el az applikáció, és iOS-re miért nem, hiszen nem használok semmilyen Android-specifikus funkciót, ami megkövetelné ezt a platformot. Emiatt a Xamarin helyett a továbbfejlesztett változatot Xamarin Forms keretrendszerben szeretném implementálni, ugyanis a Xamarin Forms Android, iOS és Windows App-ként is futtatható.

A felhasználó által elindítható, energiatakarékos módnak az implementálása. Itt Deep Sleep módra váltunk be az ESP32-vel a meghatározott időre, ezután fogjuk lekérni a felhőt frissítésért, megnézni mikor van a következő Action és addig Deep Sleepben maradni, vagy csak percenként például egyszer megnézni a felhőt.

A hasonló rendszereknél már volt szó képfelismerő technikák alkalmazásáról. Ezt hasznosnak gondolnám, ha mondjuk kint van a teraszon az okos kisállat etető, és szeretnénk elkerülni, hogy a macskánk eledeléhez más állat is hozzáférjen. Ezáltal nem időponthoz kötött etetéseket kapna, hanem alkalomhoz kötöttet. Azaz csak akkor etetne

a rendszer, ha a mi macskánk van a képen. Ezt tovább lehet finomítani, hogy csak 8-12 óra között egyszer, 12-16 óra között még egyszer, majd 16-20 óra között utoljára etet, ha látja a képen az állatot, ha éppen arra jár.

Ez a már megismert RFID chip-es megoldással egyszerűbben megvalósítható, viszont a nyakörv és egy érzékelő pontos elhelyezése után, szükség van az eledel mechanikai elzárására is. Ez akkor is életszerű megoldás lenne, ha több állatunk van, és van köztük sajátos étrendű is, amelyik, pl. kevesebbet ehet. Erre olyan ötletem támadt, hogy ezesetben RFID Chip kell az összes állat nyakába, amit a gép érzékel amikor az állat közel megy hozzá. Az elv az lehetne, hogy kitöltjük a legkevesebbet evő állat adagját, azaz a minimális adagot, amit mindegyik állat megenne, és ha odamegy az egyik állat, megnézzük, hogy ő az az állat-e, akinek a legkevesebbet kell adni. Ha igen, akkor nem töltünk tovább eledelt. Ha nem akkor rátöltünk még annyit amennyivel neki többet kell ennie a minimális mennyiségnél.

Amennyiben sokáig vagyunk távol kiskedvencünktől, az biztosan örülne a hangunknak. Ehhez egy hangszórót is beépíthetnék a rendszerbe, amivel beszélhetnénk hozzá, vagy akár előre felvett médiát játszana le SD kártyáról. Még egy interaktív ötlet, hogy egy lézer pointer és egy vagy két darab szervomotor segítségével játszhatnánk is a macskánkkal.

Ha változó a heti beosztásunk akkor az RTC modul segítségével lekérhetjük a dátumot is, így külön napokra is beállíthatná a felhasználó, hogy mikor legyenek vagy ne legyenek etetések és itatások. Például, ha egy nap sokáig dolgozik akkor aznapra állít csak...

Okos termosztáttal való összehangolás is jól jöhetne, mert akkor nem nekünk kell az értékeket mérni, illetve bizonyosan pontosabb értékeket kapnánk. Ezt M2M azaz Machine-To-Machine kommunikációval is véghez lehet vinni, amiben a felhasználónak semmi dolga nincs.

Jó ötlet lenne még egy dedikált funkció a mérleg szabad felhasználásához. Mikor a gazdi nem használhatja az etetőt, mert például nedves tápot önt kisállatának, a mérleget és tálat ugyanúgy felhasználhatná. Miközben rakja az eledelt a gazdi, a 4x7 szegmenses kijelzőn ugyanúgy látja, hogy hány grammot szedett ki eddig és amikor a felhasználó megnyomja a gombot, végzünk a procedúrával.

9 ÖSSZEFOGLALÓ

A célokat úgy gondolom elértem. Egy távolról irányítható és monitorozható okos rendszert készítettem, amivel etetni, itatni és környezeti értékeket is mérhetünk. Emellett a rendszer kedvező alkatrész költségekkel rendelkezik, ami a konkurenciával szemben előnyt jelent. Az kutatás nagy részét a hasonló rendszerek vizsgálatával töltöttem, így sokféle megoldást megismertem. Összeszedtem több olyan funkciót, amit más rendszerekben csak külön-külön lehetett megtalálni.

Ugyanakkor az etető rendszer és annak mechanikája további optimalizálásra szorul, az alapkoncepciót működőképesnek gondolom, de itt a részleteken múlik minden. Az értékeket helyesen tárolja és módosítja a felhőben. Az etetéseknel viszonylag pontos értékeket produkál.

Végösszegzésben, az elvárt funkciókat sikerül megvalósítani és a projekt költségeit alacsony szinten tartani.

Kijelenthető, hogy bármilyen száraz tápot fogyasztó állatnak alkalmas lehet. A táp legrövidebb dimenziójának az átmérője alapján 2-3mm-nél nagyobb méretű táp-szerű edelt kifog tudni adagolni gond nélkül.

10 SUMMARY

I think my goal has been accomplished. I created a remotely controlled smart system, that can feed pets, change their water and measure the environmental factors. One of the biggest goals was to create the whole system as cheap as possible, under 65\$. In the first few sections I could learn numerous ideas and good solutions for pet care systems. Using this knowledge I tried to put as many features as I could into my prototype. Most of these features are not present in one system, only in separate systems.

However, the feeder and its mechanism needs to be optimised. The basic concept is working, but it all depends on the details.

11 IRODALOMJEGYZÉK

- [1] Emma Bedford: Number of dogs and cats kept as pets worldwide in 2018, <https://www.statista.com/statistics/1044386/dog-and-cat-pet-population-worldwide/>, utoljára megtekintve: 2021.05.14.
- [2] K-Fitness: Az állatok és a házasítás története, 2019, <https://kfitness.hu/az-allatok-es-a-haziasitas-tortenete/>, utoljára megtekintve: 2021.05.14.
- [3] Charles Kosman: What To Do With Your Pets When You Go On Vacation, <https://www.thebarefootnomad.com/travel-tips/what-to-do-with-your-pets-when-you-go-on-vacation/>, utoljára megtekintve: 2021.05.14.
- [4] Ismeretlen, What to do with your pets when you go travelling, <https://littlegreybox.net/2017/03/28/what-to-do-with-your-pets-when-you-go-travelling/>, utoljára megtekintve: 2021.05.14.
- [5] M Ibrahim, H Zakaria, EE Wei Xian: Pet food autofeeder by using Arduino. IOP Conference Series: Materials Science and Engineering, Vol. 670. (012069), Malajzia, Penang-sziget, 2019
- [6] Pankaj Khatri: Automatic Pet Feeder using Arduino, 2018, <https://circuitdigest.com/microcontroller-projects/automatic-pet-feeder-using-arduino>, utoljára megtekintve: 2021.05.14.
- [7] Roj Don: Automatic Arduino Powered Pet Feeder, 2015, <https://www.instructables.com/Automatic-Arduino-Powered-Pet-Feeder/>, utoljára megtekintve: 2021.05.14.
- [8] Seungcheon Kim: Smart Pet Care System using Internet of Things. International Journal of Smart Home, Vol. 10. No. 3. 2016, 211-218. old.
- [9] S. Koley, S. Srimani, D. Nandy, P. Pall, S. Biswas és Dr. I. Sarkar: Smart Pet Feeder. Journal of Physics: Conference Series, Vol. 1797. (012018), Kalyani, Nyugat-Bengál, India, 2021
- [10] Vania, K. Karyono és I. H. T. Nugroho: Smart dog feeder design using wireless communication, MQTT and Android client, 2016 International Conference on Computer, Control, Informatics and its Applications (IC3INA), Jakarta, Indonézia, 2016
- [11] T. G. Berhan, W. T. Ahemed, T. Z. Birhan: Programmable Pet Feeder. International Journal of Scientific Engineering and Research (IJSER), Vol. 3. Issue 11. (online), 2015, 99-104. old.
- [12] K. Aasavari és G. (M.) Vaishali, Automatic Pet Feeder. 2018, <https://ssrn.com/abstract=3274472>, utoljára megtekintve: 2021.05.14.
- [13] Ji-Yong Jung, Chul-Min Ji, Joo-Rak Sohn, Hong-Jae Meng, Byung-Sun Hwang: NuriPet: A smart pet feeding machine for SNS. IEEE, Las Vegas, Nevada, USA, 2016, ISBN: 978-1-4673-8364-6, ISSN: 2158-4001

- [14] Ismeretlen, Smart Pet Feeder, <https://www.instructables.com/Smart-Pet-Feeder-1/>, 2020, utoljára megtekintve: 2021.05.14.
- [15] Madeleine Burry: The 8 Best Automatic Cat Feeders of 2021, 2021, <https://www.thesprucepets.com/best-automatic-cat-feeders-4175145>, utoljára megtekintve: 2021.05.14.
- [16] Sarah Zorn: The Best Automatic Pet Feeders of 2021, 2020, <https://www.reviewed.com/home-outdoors/best-right-now/the-best-automatic-pet-feeders>
- [17] Ismeretlen, Best Automatic Cat Feeders – 2021 Edition, <https://animalcorner.org/best-automatic-cat-feeders/>, utoljára megtekintve: 2021.05.14.
- [18] Rebekah Valentine: The Best Water Fountain for Cats and Dogs, 2021, <https://www.nytimes.com/wirecutter/reviews/best-water-fountain-for-cats-and-dogs/>, utoljára megtekintve: 2021.05.14.
- [19] Theresa Holland, The 8 Best Cat Water Fountains of 2021, 2021, <https://www.thesprucepets.com/top-automatic-water-dispensers-554933>, utoljára megtekintve: 2021.05.14.
- [20] Dr. Ernie Ward: 2018 Pet Obesity Survey Results - U.S. Pet Obesity Rates Plateau and Nutritional Confusion Grows, 2019 <https://petobesityprevention.org/2018>, utoljára megtekintve: 2021.05.14.
- [21] Dr. Jennifer Coates: Do You Know How Much to Feed a Cat? The Definitive Guide to Cat Food Portions, 2020, <https://be.chewy.com/how-much-should-i-feed-my-cat/>, utoljára megtekintve: 2021.05.14.
- [22] Ryan Llera, Robin Downing: Feeding Times and Frequency for your Dog, <https://vcahospitals.com/know-your-pet/feeding-times-and-frequency-for-your-dog>, utoljára megtekintve: 2021.05.14.
- [23] Ismeretlen, How often should you change your pet's water?, 2020, <https://www.wellnesspetfood.com/our-community/wellness-blog/health-nutrition/general-care/how-often-should-you-change-your-pets>, utoljára megtekintve: 2021.05.14.
- [24] Ismeretlen, Hydration for Pets: How Much is Enough?, 2014, <https://hastingsvet.com/hydration-much-enough/>, utoljára megtekintve: 2021.05.14.
- [25] Jessie Isbell: Is Your Dog Getting Enough Water?, 2016, <https://hfu.hollywoodfeed.com/92354246-7-easy-ways-to-keep-your-pets-hydrated/>, utoljára megtekintve: 2021.05.14.
- [26] Dr. Sarah Wooten: How Much Water Should Dogs & Cats Drink Each Day?, 2020, <https://www.hillspet.com/pet-care/nutrition-feeding/how-much-water-should-dogs-and-cats-drink>, utoljára megtekintve: 2021.05.14.

- [27] Pierre: How Often Should You Change Your Cat's Water?
<https://purrcaze.com/how-often-should-you-change-your-cats-water>, utoljára megtekintve: 2021.05.14.
- [28] Franny Syufy: How Much Water Does an Adult Cat Need to Drink?, 2020,
<https://www.thesprucepets.com/how-much-water-555599>,
utoljára megtekintve: 2021.05.14.
- [29] S. Handl és J. Fritz: The water requirements and drinking habits of cats, 2018,
<https://vetfocus.royalcanin.com/en/scientific/the-water-requirements-and-drinking-habits-of-cats>, utoljára megtekintve: 2021.05.14.
- [30] Ismeretlen, Do Cats Drink Water? Cat Hydration & Dehydration Prevention,
<https://www.purina.com/articles/cat/health/guide-to-cat-hydration>, utoljára megtekintve: 2021.05.14.
- [31] Ismeretlen, The Best House Temperature For Pets, 2018,
<https://paylesspower.com/blog/best-house-temperature-for-pets/>, utoljára megtekintve: 2021.05.14.
- [32] Ismeretlen, Why is humidity important?,
<https://www.metoffice.gov.uk/weather/learn-about/weather/types-of-weather/humidity/effects>, utoljára megtekintve: 2021.05.14.
- [33] Alja Isakovic, Track your home's temperature and humidity to keep your pets safe in the summer heat, 2015, <http://blog.cubesensors.com/2015/07/track-your-homes-temperature-and-humidity-to-keep-your-pets-safe-in-the-summer-heat>, utoljára megtekintve: 2021.05.14.
- [34] Lionel Sujay Vailshery: Internet of Things (IoT) and non-IoT active device connections worldwide from 2010 to 2025, 2021,
<https://www.statista.com/statistics/1101442/iot-number-of-connected-devices-worldwide/>, utoljára megtekintve: 2021.05.14.
- [35] R. Williams, E. McMahon, S. Samtani, M. Patton, H. Chen: Identifying Vulnerabilities of Consumer Internet of Things (IoT) Devices: A Scalable Approach. IEEE, Peking, Kína, 2017, ISBN: 978-1-5090-6728-2, Electronic ISBN: 978-1-5090-6727-5
- [36] Z. Zhang, M. C. Y. Cho, C. Wang, C. Hsu, C. Chen and S. Shieh: IoT Security: Ongoing Challenges and Research Opportunities, 2014 IEEE 7th International Conference on Service-Oriented Computing and Applications, Macue, Japán, 2014, 230-234. old.
- [37] Ismeretlen, Internet Census 2012,
<http://census2012.sourceforge.net/paper.html>, utoljára megtekintve: 2021.05.14.
- [38] D. Mocrii a, Y. Chenb, P. Musilek: IoT-based smart homes: A review of system architecture, software, communications, privacy and security, Internet of Things 1–2, Elsevier, 2018, vol. 1-2., 81-98. old.
- [39] OV2640 Datasheet, 2006

- [40] OV7670 Datasheet, 2006
- [41] Ismeretlen, <https://hu.wikipedia.org/wiki/Mikrovezérlő>, utoljára megtekintve: 2021.05.14.
- [42] Arduino Mega Datasheet, 2010
- [43] Arduino Nano Datasheet, 2008
- [44] Arduino Uno Datasheet, 2010
- [45] ESP8266 Datasheet, 2014
- [46] ESP32 Datasheet, 2016
- [47] ESP32-CAM Datasheet, 2017
- [48] Sara Santos: ESP32 Camera Dev Boards Review and Comparison (Best ESP32-CAM), 2020, <https://makeradvisor.com/esp32-camera-cam-boards-review-comparison/>, utoljára megtekintve: 2021.05.14.
- [49] Ismeretlen: Electric Motor Comparison (DV vs Servo vs Stepper), 2018, <https://totemmaker.net/blog/electric-motor-comparison-stepper-vs-servo-vs-dc/>, utoljára megtekintve: 2021.05.14.
- [50] HX711 Datasheet, 2014
- [51] DS3231 Datasheet, 2005
- [52] AT24C32 Datasheet, 2003
- [53] HC-SR04 Datasheet, 2010
- [54] DHT11 Datasheet, 2012
- [55] Ismeretlen, Mobile Operating System Market Share Europe, 2021, <https://gs.statcounter.com/os-market-share/mobile/europe>, utoljára megtekintve: 2021.05.14.
- [56] Dr. Jennifer Coates, Are You Feeding Your Dog the Right Amount?, 2020, <https://www.petmd.com/blogs/nutritionnuggets/dr-coates/2015/july/are-you-feeding-your-dog-right-amount-32905>, utoljára megtekintve: 2021.05.14.
- [57] Brittany: How Much Should My Dog Eat?, 2019, <https://www.prudentpet.com/how-much-should-dog-eat/>, utoljára megtekintve: 2021.05.14.
- [58] Stanton P. Allen: Your Cat's Nutritional Needs: A Science - Based Guide For Pet Owners, https://www.nap.edu/resource/10668/cat_nutrition_final.pdf, utoljára megtekintve: 2021.05.14.
- [59] Stanton P. Allen: Your Dog's Nutritional Needs: A Science-Based Guide For Pet Owners, https://www.nap.edu/resource/10668/dog_nutrition_final_fix.pdf, utoljára megtekintve: 2021.05.14.
- [60] Marion Nestle, Malden Nesheim: Feed Your Pet Right: The Authoritative Guide to Feeding Your Dog and Cat, Simon & Schuster, 2010, utoljára megtekintve: 2021.05.14.

- [61] Dr. Lisa A. Pierson, Feeding Your Cat: Know the Basics of Feline Nutrition, <https://catinfo.org/>, utoljára megtekintve: 2021.05.14.
- [62] Ismeretlen, How Often Should You Feed Your Cat?, <https://www.vet.cornell.edu/departments-centers-and-institutes/cornell-feline-health-center/health-information/feline-health-topics/how-often-should-you-feed-your-cat>, utoljára megtekintve: 2021.05.14.
- [63] Ismeretlen, Your dog's diet, <https://www.pdsa.org.uk/taking-care-of-your-pet/looking-after-your-pet/puppies-dogs/your-dogs-diet>, utoljára megtekintve: 2021.05.14.
- [64] Ismeretlen, Feeding Your Cat, 2017, <https://www.vet.cornell.edu/departments-centers-and-institutes/cornell-feline-health-center/health-information/feline-health-topics/feeding-your-cat>, utoljára megtekintve: 2021.05.14.
- [65] Susan Jenks, Our Fat Pets, 2018, <https://www.nytimes.com/2018/08/02/well/fat-pets-dog-cat-health.html>, utoljára megtekintve: 2021.05.14.
- [66] Michael Babins, How Often Should You Change Your Dog's Water?, <https://wagwatchers.com/tips-tricks/how-often-should-you-change-your-dogs-water>, utoljára megtekintve: 2021.05.14.
- [67] Tanish Sehgal, Shubham More:
Home Automation using IOT and Mobile App. IRJET, Maharashtra, India, 2017. e-ISSN: 2395 -0056, p-ISSN: 2395-0072
- [68] Microsoft, Learn Xamarin.
<https://dotnet.microsoft.com/en-us/learn/xamarin>, utoljára megtekintve: 2021.12.09.
- [69] Mr. Sachin M. Karmuse, Mr. Saurabh R. Prasad:
Android based Home Automation System with Power Optimization Modes. IRJET, Ichalkaranji, India, 2019. e-ISSN: 2395-0056, p-ISSN: 2395-0072
- [70] Dejan Nedelkovski, DHT11 & DHT22 Sensors Temperature and Humidity Tutorial using Arduino, utoljára megtekintve: 2021.12.09.
- [71] Ismeretlen: Universally unique identifier,
https://en.wikipedia.org/wiki/Universally_unique_identifier, utoljára megtekintve: 2021.12.09.
- [72] Sara Santos, ESP32-CAM Connect External Antenna (Extend Wi-Fi Coverage), <https://randomnerdtutorials.com/esp32-cam-connect-external-antenna/>, utoljára megtekintve: 2021.12.15.

12 ÁBRAJEGYZÉK

2-1. ábra: Kisállat etető gép Arduino eszköz felhasználásával.....	5
2-2. ábra: Automata Kisállat Etető Arduino Használatával.....	6
2-3. ábra: Automata Arduinoval Működő Kisállat Etető.....	7
2-4. ábra: Okos Kisállat Gondozó Rendszer.....	8
2-5. ábra: Okos Kisállat Etető.....	10
2-6. ábra: Okos Kutya Etető	11
2-7. ábra: Programozható Kisállat Etető.....	13
2-8. ábra: Automata Kisállat Etető Komponens Terv.....	14
2-9. ábra: NuriPet: Egy Okos Kisállat Etető Gép, SNS-el.....	14
2-10. ábra: Okos Kisállat Etető.....	16
3-1. ábra: Rendszer Architektúra	20
5-1. ábra: Rendszerterv	27
5-2. ábra: Mikrokontroller Blokk Diagram.....	29
5-3. ábra: Vezérlés Folyamatábra	30
5-4. ábra: Etető Gép Blokk Diagram	32
5-5. ábra: Etető Gép Folyamatábra	33
5-6. ábra: Itató Gép Blokk Diagram	34
5-7. ábra: Applikáció Wireframe	36
5-1. táblázat: Kamerák Összehasonlítása.....	28
5-2. táblázat: Mikrokontrollerek Összehasonlítása.....	30
5-3. táblázat: Tervezett Költségvetés.....	37